

2011

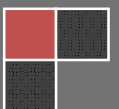
# Universidad Autónoma Metropolitana - Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

“Desarrollo de una aplicación de software para la Implementación de Estrategias de Muestreo en Poblaciones de Datos.”

Ximena Cerna Landa  
México, D.F.  
Trimestre 011-I



**Para mi mamá y mi  
hermano, ellos son el  
viento bajo mis alas.**

**Para mi asesor Héctor  
Javier, quien además de  
ser un gran profesor es  
un gran amigo.**

# Contenido

<b>1. OBJETIVO DEL PROYECTO</b>	<b>5</b>
<b>2. OBJETIVOS PARTICULARES</b>	<b>5</b>
<b>3. BASES TEÓRICAS</b>	<b>5</b>
3.1 TERMINOLOGÍA BÁSICA	6
3.2 MÉTODOS DE MUESTREO	6
<b>4. DESCRIPCIÓN DE LA APLICACIÓN</b>	<b>7</b>
4.1 ANÁLISIS DE REQUERIMIENTOS	7
4.2 JUSTIFICACIÓN	8
4.2.1 ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN	8
4.2.2 DISEÑO DE LA INTERFAZ GRÁFICA	9
<b>5. IMPLEMENTACIÓN (MEMORIA DE DISEÑO)</b>	<b>10</b>
5.1 ESTRUCTURA GENERAL	11
5.2 ARQUITECTURA DE LA APLICACIÓN	11
5.2.1 INTERACCIÓN ENTRE CLASES	15
5.3 DESCRIPCIÓN TÉCNICA	21
5.3.1 MÉTODO ALEATORIO SIMPLE NO PONDERADO	22
5.3.1.1 Algoritmo 0. Permutación Aleatoria	22
5.3.1.2 Algoritmo 1. Hipergeométrico	23
5.3.1.3 Algoritmo 2. Substitución	25
5.3.1.4 Algoritmo 3. Saltos Aleatorios A	26
5.3.1.5 Algoritmo 4. Saltos Aleatorios B	27
5.3.2 MÉTODO SISTEMÁTICO	28
5.3.3 MÉTODO ESTRATIFICADO	30
5.3.4 MÉTODO ALEATORIO SIMPLE PONDERADO	34
5.4 RECOMENDACIONES	37
<b>6. PRUEBAS</b>	<b>39</b>
6.1 MÉTODO ALEATORIO SIMPLE NO PONDERADO	39
6.2 ALEATORIO SIMPLE PONDERADO	42
6.3 MÉTODO SISTEMÁTICO	43

<b>6.4</b>	<b>MÉTODO ESTRATIFICADO .....</b>	<b>46</b>
<b>7</b>	<b><u>CONCLUSIONES .....</u></b>	<b>48</b>
<b>8</b>	<b><u>BIBLIOGRAFÍA.....</u></b>	<b>49</b>
<b>9</b>	<b><u>LISTADO DEL PROGRAMA.....</u></b>	<b>50</b>
<b>9.1</b>	<b>CLASE PRINCIPAL .....</b>	<b>50</b>
9.1.1	PTO2APP.JAVA .....	50
<b>9.2</b>	<b>PAQUETE VISTAS .....</b>	<b>51</b>
9.2.1	INICIO.JAVA.....	51
9.2.2	ASIMPLEPONDERADO.JAVA .....	60
9.2.3	ASIMPLENP.JAVA.....	76
9.2.4	SISTEMATICO.JAVA.....	90
9.2.5	CODIFICACION.JAVA .....	107
9.2.6	ESTRATIFICADO.JAVA.....	134
9.2.7	TOMAMUESTRAS.JAVA .....	148
9.2.8	FINAL.JAVA .....	160
9.2.9	FINALESTRATA.JAVA.....	185
9.2.10	FINALESTRATIFICADO.JAVA .....	204
9.2.11	POLITAPICO.JAVA.....	226
9.2.12	AYUDA.JAVA.....	238
<b>9.3</b>	<b>PAQUETE ARCHIVOS.....</b>	<b>241</b>
9.3.1	ARCHIVOS.JAVA.....	241
<b>9.4</b>	<b>PAQUETE KERNEL.....</b>	<b>253</b>
9.4.1	MUESTREO.JAVA .....	253
9.4.2	RESULTADOS.JAVA .....	260
<b>9.5</b>	<b>PAQUETE ESTADÍSTICA.....</b>	<b>284</b>
9.5.1	OPERACIONES.JAVA .....	284
<b>9.6</b>	<b>PAQUETE MODELOSTABLAS.....</b>	<b>291</b>
9.6.1	TABLAINICIAL.JAVA .....	291
9.6.2	TABLAMUESTRA.JAVA .....	293
9.6.3	TABLARESPOB.JAVA .....	295
9.6.4	TABLAFINAL.JAVA.....	297
9.6.5	TABLALIMITES.JAVA .....	298
9.6.6	TABLAESTRATOS.JAVA .....	301

## 1. Objetivo del Proyecto

Diseñar, elaborar y evaluar una aplicación de software para asistir en el diseño y análisis de procedimientos de muestreo estadístico.

## 2. Objetivos Particulares

- Definir los requerimientos de la aplicación.
- Diseñar la arquitectura de la aplicación.
- Planificar los distintos módulos de la aplicación (Interfaz, Esquemas, Archivos, Generador de números aleatorios).
- Programar los distintos módulos e integrarlos a la aplicación.
- Realizar el plan de pruebas.
- Redactar la documentación técnica y la documentación de usuario

## 3. Bases Teóricas

El muestreo es una herramienta de la investigación científica. Su función básica es determinar qué parte de una realidad en estudio (población o universo) debe examinarse con la finalidad de hacer inferencias sobre dicha población. Obtener una muestra adecuada significa lograr una versión representativa y simplificada de la población que reproduzca de algún modo sus rasgos básicos.

### *Las ventajas del muestreo*

- a) Costos reducidos.
- b) Mayor rapidez para obtener resultados.
- c) Mayor exactitud o mejor calidad de la información debido a los siguientes factores:
  - Volumen de trabajo reducido.
  - Puede existir mayor supervisión en el trabajo.
  - Se puede dar más entrenamiento al personal.
- d) Factibilidad de hacer el estudio cuando la toma de datos implica técnicas destructivas, por ejemplo:
  - Pruebas de germinación.
  - Análisis de sangre.
  - Control de calidad.

Con el fin de que la muestra sea lo más representativa posible de la población, es necesario tomar en cuenta sus características o cualquier información que ayude a mejorar la calidad y representatividad de la muestra. Si no se observan características que diferencien a los elementos de la población, lo más adecuado es elegir la muestra mediante un proceso aleatorio. Si cada uno de los individuos tiene la misma representatividad, se puede asignar a cada elemento la misma probabilidad, sino será necesario obtener mediante un método

estadístico o empírico las probabilidades para elegir cada elemento. Por otra parte, los elementos de la muestra se pueden elegir con o sin remplazo.

Si se observan ciertas características que identifiquen ciertos elementos de la población o uno o varios patrones que distingan algunos elementos de otros, será de gran utilidad agrupar los datos y seleccionar elementos dentro de cada grupo, no necesariamente del mismo tamaño.

### 3.1 Terminología básica

**Población:** conjunto de todos los individuos que son objeto del estudio.

**Muestra:** parte de la población en la que miden las características estudiadas.

**Muestreo:** proceso seguido para la extracción de una muestra.

**Encuesta:** proceso de obtener información de la muestra.

### 3.2 Métodos de Muestreo

A continuación se listan los métodos de muestreo revisados para la elaboración de este proyecto:

El **Muestreo Aleatorio Simple** consiste en seleccionar  $n$  individuos de una población al azar, suponiendo que todos tienen la misma importancia o probabilidad de ser elegidos. La muestra puede tomarse con o sin reemplazo según se requiera. Si la muestra se toma con reemplazo el procedimiento es relativamente simple pues no hay que preocuparse de elegir más de una vez los mismos elementos. Sin embargo si se realiza se desea realizar un muestreo sin reemplazo existen varios algoritmos que evitan o reducen la probabilidad de tomar más de una vez al mismo elemento.

El **Muestreo Sistemático** consiste en seleccionar individuos de una población, saltando  $k$  posiciones, hasta completar los  $n$  individuos deseados.

El **Muestreo Aleatorio Simple Ponderado** toma en cuenta la importancia de los individuos en la lista de tamaño  $N$ . Se toma la muestra considerando la importancia (ponderación).

El **Muestreo Estratificado** consiste en separar la lista inicial de tamaño  $N$  en  $H$  estratos y seleccionar  $n_i$  individuos a cada estrato  $H$  aplicando el método aleatorio simple, con la restricción de que  $n = n_1 + n_2 + n_3 + \dots + n_h$ .

El **Muestreo por Conglomerados** consiste en seleccionar el mismo número de individuos dividido en  $K$  grupos sin importar su tamaño.

El **Muestreo de Razón** consiste tomar una muestra de  $n$  individuos considerando el radio de dos características.

El **Muestreo Politápico** consiste en tomar muestras en forma secuencial. Por ejemplo la aplicación de este algoritmo para tomar una muestra de tamaño  $n$  ( $n = n_1 + n_2$ ) en dos etapas

consiste en tomar  $n_1$  elementos de la lista inicial de tamaño  $N$  y luego tomar  $n_2$  elementos de la sub lista de tamaño  $n_1$ .

## 4. Descripción de la Aplicación

### 4.1 Análisis de Requerimientos

Actualmente, las aplicaciones computacionales dedicadas al tema del muestreo presentan carencias en ámbitos como interfaz gráfica, manejo de formatos de archivos, falta de diversidad en cuanto a métodos de muestreo a utilizar por el usuario, etc.

Ante la necesidad del usuario de contar con una aplicación que sea capaz de asistirlo en la selección de muestras de una manera óptima, el desarrollo de este proyecto se basa en el diseño e implementación de una aplicación capaz de cubrir éstos ámbitos.

Para el diseño, se tomaron en cuenta los siguientes requerimientos que para su mejor desarrollo se plantean en forma de preguntas:

- ¿Quiénes son los usuarios, cuáles sus conocimientos, y qué pueden aprender?

Esta aplicación está dedicada a usuarios que requieran aplicar algún método de muestreo, ya sea para investigación académica o profesional. Aunque es deseable que se tenga conocimiento previo del tema del muestreo, no es necesario tener conocimientos especializados sobre los algoritmos ya que la aplicación a través de sus módulos, guía al usuario con sugerencias para el cálculo de la muestra. La aplicación deberá ser muy intuitiva gracias a la interfaz gráfica diseñada, por lo que no tendrá dificultad para manejar la aplicación.

- ¿Qué quieren o necesitan hacer los usuarios?

El usuario necesita tener un archivo en formato .txt o .dat con los datos de la población a muestrear, así como una propuesta inicial del tamaño de la muestra. Posteriormente el programa le sugerirá otro tamaño de ser necesario.

- ¿Cuál es la formación general de los usuarios?

La aplicación de muestreo la utilizarán básicamente usuarios dedicados a la investigación.

- ¿Qué debe dejarse a la máquina? ¿Qué al usuario?

El usuario es responsable de que sus datos estén correctos, de elegir la variable y el método deseados. Será decisión del mismo, seguir las sugerencias que la aplicación haga respecto al tamaño de la muestra así como elegir el algoritmo de selección de muestra que más le convenga. La máquina, por su parte, es responsable de hacer los cálculos correspondientes de una manera eficaz, veraz y confiable.

A continuación se presenta el diagrama de flujo con la arquitectura lógica de la aplicación propuesta tomando en cuenta lo anterior:

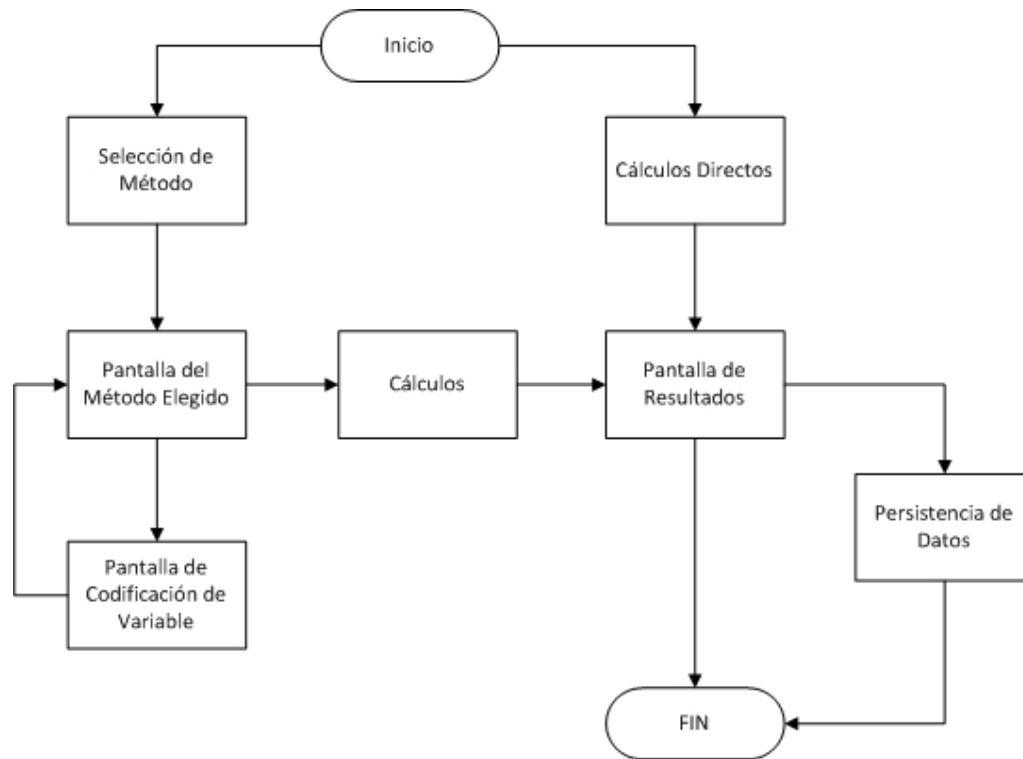


Figura 1. Diagrama de Flujo de la Lógica de la Aplicación

## 4.2 Justificación

### 4.2.1 Elección del Lenguaje de Programación

Para el desarrollo de esta aplicación se eligió el lenguaje de programación Java por ser un lenguaje independiente de la plataforma y tener un entorno de ejecución ligero y gratuito. Además, Java es un lenguaje orientado a objetos, lo que facilita diseñar el software de forma modular. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos.

Un módulo es un archivo Java que contiene clases escritas para interactuar con las APIs (interfaz de programación de aplicaciones) de los entornos de desarrollo (IDE) y un archivo especial (manifest file) que lo identifica como módulo.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

El entorno de desarrollo utilizado es Netbeans, un producto libre y gratuito que se puede obtener desde la página de internet: [www.netbeans.org](http://www.netbeans.org).

Cabe mencionar que existen otros entornos de desarrollo similares al Netbeans como lo es Eclipse, que es una herramienta de desarrollo más completa en cuanto a que se extiende para lenguajes de programación que Netbeans no contempla, sin embargo para el desarrollo de esta aplicación se tiene más experiencia trabajando bajo Netbeans facilitando la construcción de aplicaciones basadas en SWT (entorno gráfico).



Independientemente del IDE con el que se trabaje, es necesario bajar de internet el JRE (Java Runtime Environment, o Entorno en Tiempo de Ejecución de Java) que es el software necesario para ejecutar cualquier aplicación desarrollada para la plataforma Java. Puede obtenerse de la siguiente dirección:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html>

#### 4.2.2 Diseño de la Interfaz Gráfica

La interfaz gráfica es la parte principal de la aplicación, ya que es a través de ella que el usuario interactúa con los diferentes métodos de muestreo implementados.

Consiste en un conjunto de pantallas presentadas en secuencia, que depende del método de muestreo seleccionado.

El diseño es simple, pero a la vez cuenta con los elementos requeridos por cada método y permite al usuario navegar por toda la aplicación de manera eficiente y sencilla.

Se compone por cajas de texto para introducir el tamaño de la muestra, botones para cargar el archivo que contiene la población y navegar entre pantallas, cajas de selección para la variable que se va a utilizar, tablas para desplegar resultados del muestreo y etiquetas con instrucciones. Todas utilizan el tipo de letra Trebuchet MS ya que se considera una fuente que brinda estética a la aplicación.

Algunos métodos requieren de más entradas de datos que otros, pudiendo resultar en vistas dispares en cuanto a la distribución de los componentes, por lo que se diseñó una plantilla base para darle un aspecto homogéneo a la aplicación y sobre ella agregar los componentes requeridos de cada método. (Ver Figura 2).

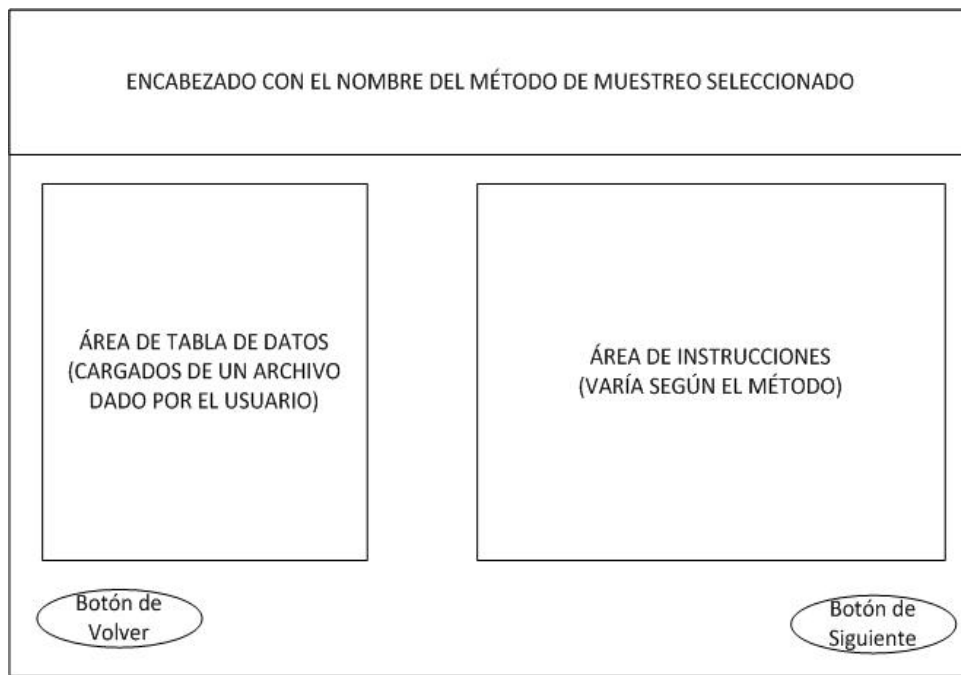


Figura 2. Plantilla de diseño de la Interfaz Gráfica de la aplicación.

Adicionalmente, está diseñada e implementada para cubrir los siguientes requisitos básicos de la usabilidad:

- **Facilidad de Aprendizaje:** Para que el usuario se familiarice y distinga los diferentes métodos utilizados a lo largo de esta aplicación, se asignó a la interfaz gráfica un color específico para cada uno como se muestra en la siguiente tabla:

Método de Muestreo / Pantalla	Color Asociado
Aleatorio Simple no Ponderado	Morado
Aleatorio Simple Ponderado	Gris
Sistemático	Azul
Estratificado	Amarillo
Politápico	Verde
Módulo de Codificación	Naranja
Resultados	Rosa

- **Facilidad de Uso:** Tiene botones y etiquetas con instrucciones claras y visibles, con un tamaño de letra adecuado para la fácil comprensión del usuario. Además se incluyen menús de ayuda en cada método de muestreo.
- **Flexibilidad:** Se diseñó de tal manera que se puede utilizar en cualquier circunstancia que se requiera de muestreo estadístico y además es una aplicación portable, ya que, al estar escrita en lenguaje Java, puede utilizarse bajo más de una plataforma.
- **Robustez:** Cuenta con la opción de almacenar la muestra generada en un archivo junto con los cálculos para usos posteriores. Además se incluyen las siguientes funciones para que el usuario obtenga una muestra óptima o compare el desempeño de los algoritmos implementados: cálculo de tamaño de la muestra, toma de muestras con o sin reemplazo.

## 5 Implementación (Memoria de Diseño)

El modelo elegido para diseñar la aplicación de estrategias de muestreo está basado en el modelo-vista-controlador, que se describe a continuación:

**Modelo Vista Controlador (MVC)** es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, sin embargo este modelo es útil en la implementación de esta aplicación ya que las clases que funcionan como vistas permanecen ligeras y sólo sirven de presentación, mientras que otras solo se dedican a cálculos y seguir algoritmos.

- **Modelo:** Representación específica de la información con la cual el sistema opera.
- **Vista:** Interfaz de usuario.
- **Controlador:** Responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y a la vista.

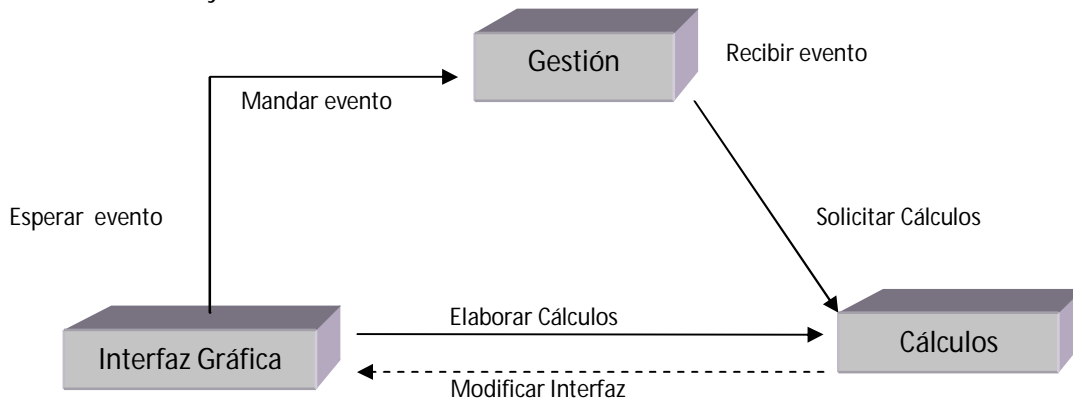


Figura 3. Diagrama Conceptual del Sistema de Muestreo

## 5.1 Estructura General

La aplicación está formada de 2 partes principales. La primera se encarga de hacer el muestreo correspondiente utilizando como datos de entrada el archivo que introdujo el usuario y se hace el muestreo según el usuario lo haya indicado.

Una vez tomada la muestra, la segunda parte de la aplicación se encargará de calcular los intervalos de confianza, varianza, desviación estándar y error estimado de la muestra y de la población a partir de una confianza dada, para posteriormente mostrar en pantalla los resultados.

El usuario tiene entonces la opción de guardar la muestra en un archivo por medio de la clase dedicada a la persistencia de datos, aplicar otro método de muestreo ó salir de la aplicación.

## 5.2 Arquitectura de la Aplicación

La aplicación desarrollada se compone de 6 paquetes como lo muestra la Figura 5. Esto es con la finalidad de tener una estructura más comprensible y ordenada de la aplicación, facilitando su mantenimiento.

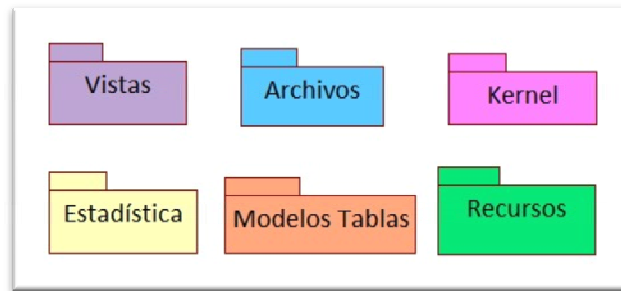


Figura 5. Diagrama de Paquetes de la aplicación.

En el paquete Vistas se encuentran 10 clases que corresponden a la interfaz gráfica de la aplicación.

A continuación se muestran las clases que componen a este paquete con sus atributos y métodos. Posteriormente se hará más énfasis a los métodos de cada clase.

El paquete Archivos (ver Figura 6) contiene una clase llamada Archivos, que gestiona las operaciones de apertura, cerrado y escritura de archivos generados por el sistema y por el usuario.



Figura 6. Diagrama de Clase del Paquete Archivos.

El paquete Vistas (ver Figura 7) tiene las clases que corresponden a la interfaz gráfica de la aplicación. Estas clases interactúan de diferentes maneras con las demás clases contenidas en los paquetes. Más adelante se detallan las interacciones.

## Vistas

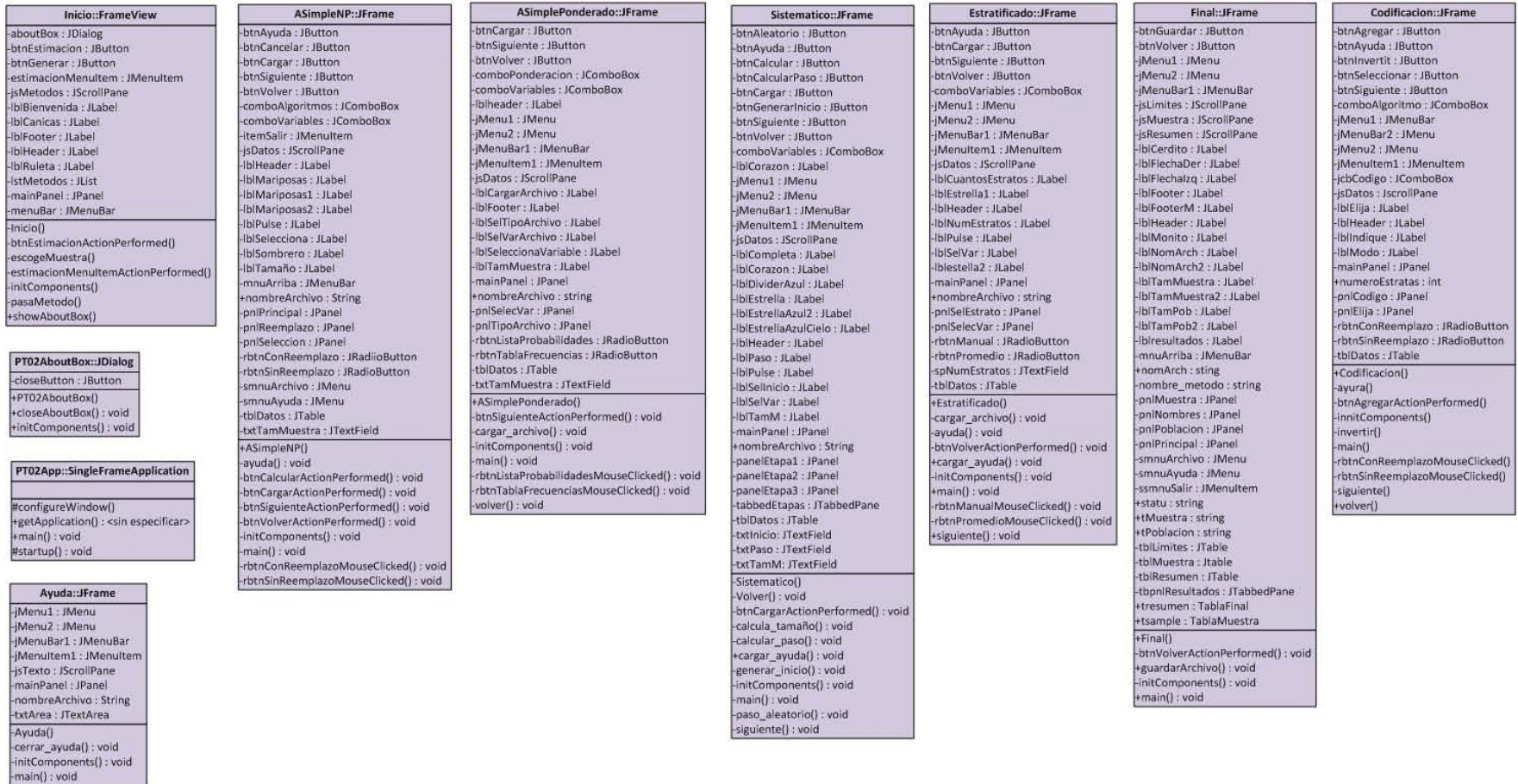


Figura 7. Diagrama de Clases del Paquete Vistas

El paquete Estadística (ver Figura 8) contiene la clase Operaciones que se encarga de calcular la varianza, promedio, intervalos de confianza, desviación estándar, etc. para la población y la muestra. Adicionalmente contiene algoritmos de ordenamiento para números enteros y flotantes.

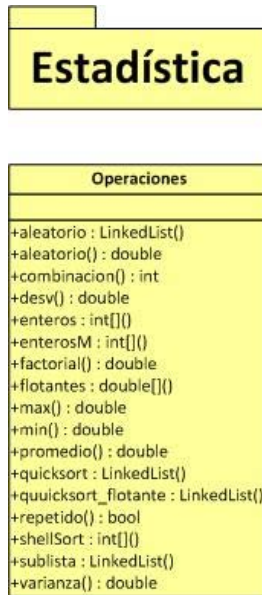


Figura 8. Diagrama de la Clase Operaciones del Paquete Estadística.

El paquete Modelos Tablas (ver Figura 9) contiene clases que arman los modelos de las tablas que se despliegan en la interfaz gráfica, tanto en los resultados como en los diferentes métodos.

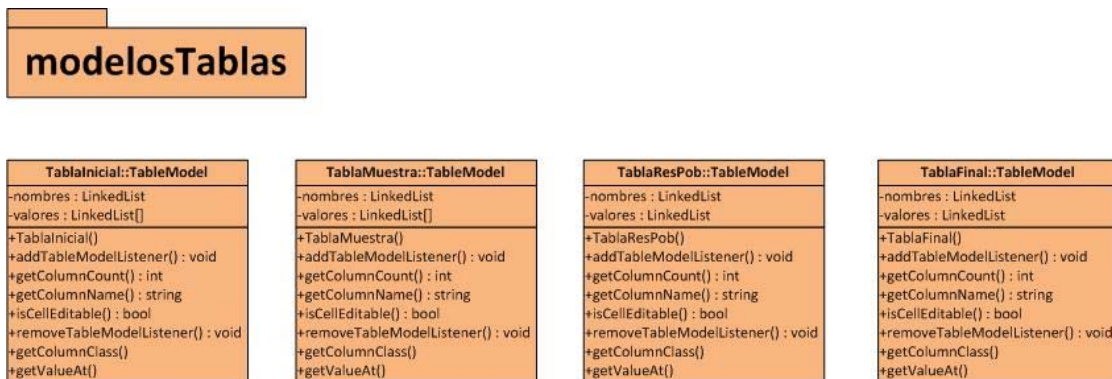


Figura 9. Diagrama de Clases del Paquete ModelosTablas.



El paquete Kernel (ver Figura 10) tiene 2 clases que son el corazón de la aplicación: Resultados y Muestreo. La clase Muestreo es la encargada de controlar a los métodos de muestreo llamando a la clase Resultados con los argumentos que según le correspondan.

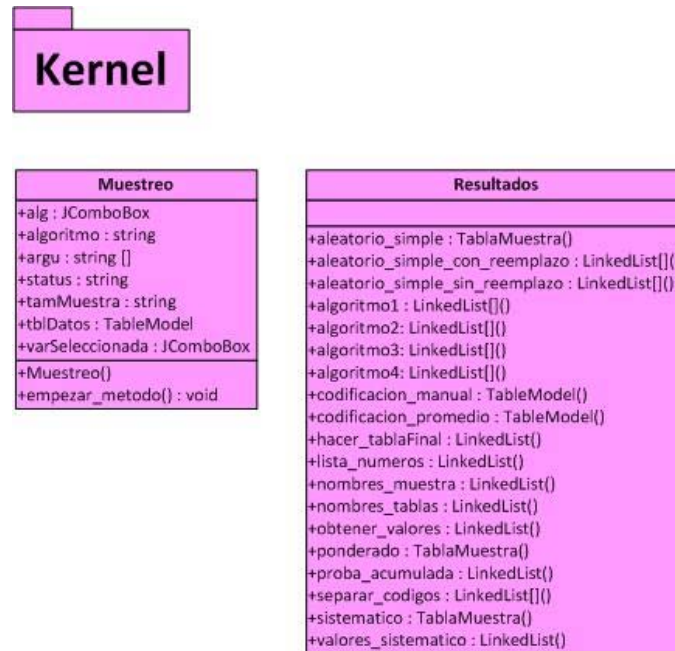


Figura 10. Diagrama de Clases del Paquete Kernel.

Finalmente, el paquete recursos contiene archivos de imágenes necesarias para darle una apariencia agradable al sistema así como archivos de configuración de las vistas para cambio de apariencia del sistema.

### 5.2.1 Interacción entre clases

Las clases de la Aplicación están diseñadas para interactuar entre ellas por medio de paso de argumentos, es decir, al momento de su instanciación, se les pasa por medio de argumentos los datos a desplegar, en el caso de la interfaz gráfica ó el identificador de la función que deben utilizar para el cálculo o toma de muestras.

Por ejemplo, para la clase Inicio que contiene la primera pantalla que se despliega cuando se arranca la aplicación, se implementó el método llamado pasaMetodo que se manda llamar al momento de dar click con el mouse sobre el nombre del método seleccionado desplegado en una lista.

Éste es encargado de mandar llamar las clases que pertenecen a la interfaz gráfica según corresponda por medio de identificadores. En el siguiente código perteneciente al método que se mencionó anteriormente, se ilustran los identificadores utilizados para llamar a cada método:

```

private void pasaMetodo(java.awt.event.MouseEvent evt)
{
    String[] args={};
    switch(lstMetodos.getSelectedIndex())
  
```

```

{
  case 0:
    ASimpleNP.main(args);
    break;
  case 1:
    ASimplePonderado.main(args);
    break;
  case 2:
    Sistematico.main(args);
    break;
  case 3:
    Estratificado.main(args);
    break;
  case 4:
    Politapico.main(args);
    break;
  default:
    JOptionPane.showMessageDialog(null, lstMetodos.getSelectedValue().toString()+"
    Próximamente Codificado!!!...");
    break;
}
}

```

Al momento de ser llamadas, las clases de la interfaz gráfica, pasan argumentos a las clases de gestión (*Archivos*, *Métodos* y *Resultados*) y ellas a través de sus constructores despliegan los componentes correspondientes.

Una vez desplegados los componentes del método correspondiente, se le pide al usuario que cargue el archivo que contiene los datos que se van a muestrear y por medio del evento que genera el botón "Cargar" se manda llamar a la clase *Archivos*. Esta clase por medio de su método *manejar\_archivo*, recibe la tabla donde desplegará los datos introducidos y el comboBox que contendrá las variables que el usuario deberá seleccionar para llevar a cabo el muestreo. Una vez cumplido lo anterior, cuando el usuario da click en el botón "Siguiete", se genera el evento que manda llamara a la clase *Muestreo* que, para saber qué tipo de muestreo es se le mandan los argumentos a su constructor: el comboBox que contiene las variables, el modelo de tabla que contiene los datos a utilizar y cadenas de caracteres que le indican si se hará el muestreo con o sin reemplazo y que algoritmo de selección de muestra seleccionó el usuario. Posteriormente por medio del método *empezar\_metodo*, se mandan argumentos que le indican a la clase de qué método se trata por medio de un número identificador, tal como en la función de ejemplo anterior.

Finalmente, utiliza los métodos de la clase *Resultados* que a su vez manda llamar a las clases de *Modelos de Tablas* y *Operaciones* para armar la información que será utilizada para enviar como argumento para el constructor de la clase de la interfaz gráfica *Resultados* quien desplegará los resultados en sus componentes correspondientes.



A continuación se presenta el siguiente diagrama de clases que ilustra interacción entre las clases que componen la Aplicación completa de manera general, posteriormente se presentan diagramas relacionales para cada método de muestreo.

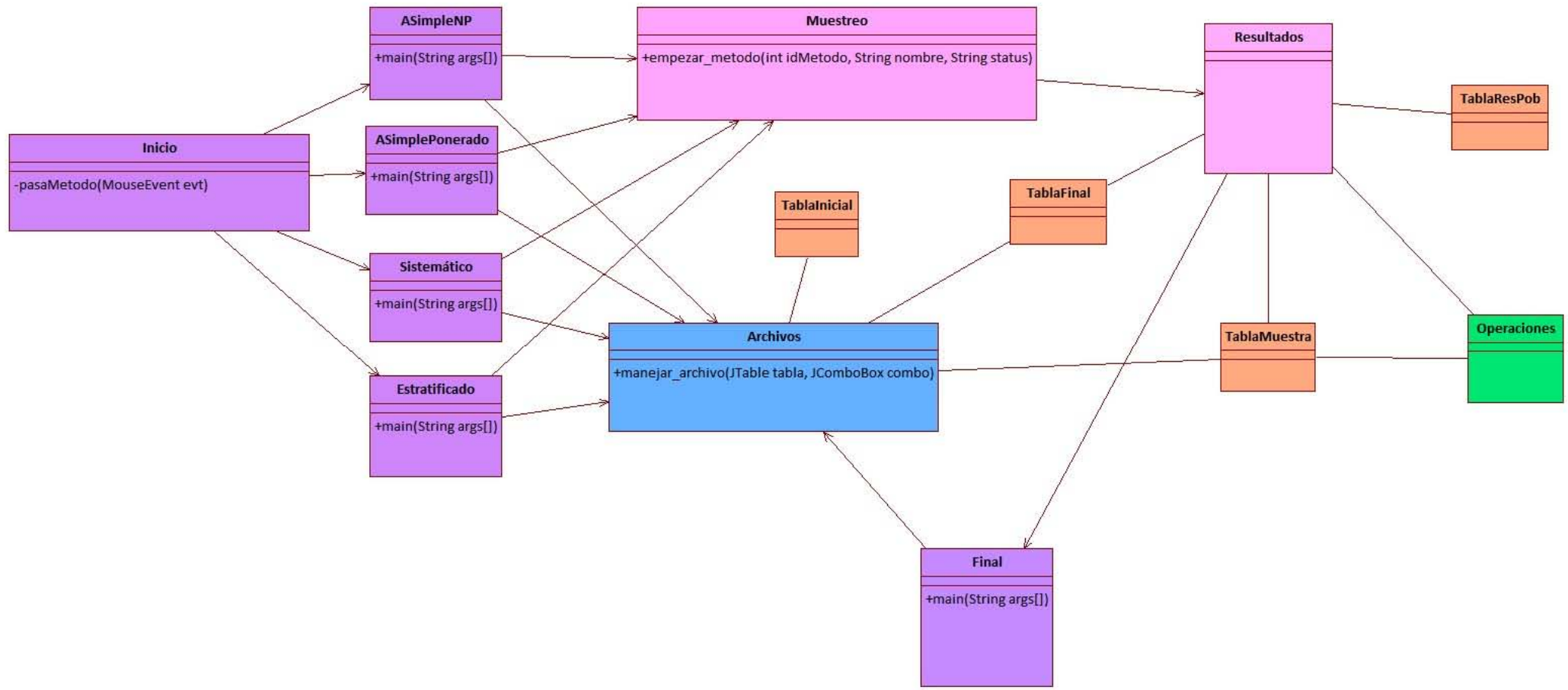


Figura 11. Diagrama de clases de la Aplicación

El siguiente diagrama representa la interacción de la clase Inicio, que es la que arranca la aplicación con las demás clases del proyecto. Se muestra como dependiendo del índice de la lista IstMetodo se manda llamar al método main del método de muestreo asignado.

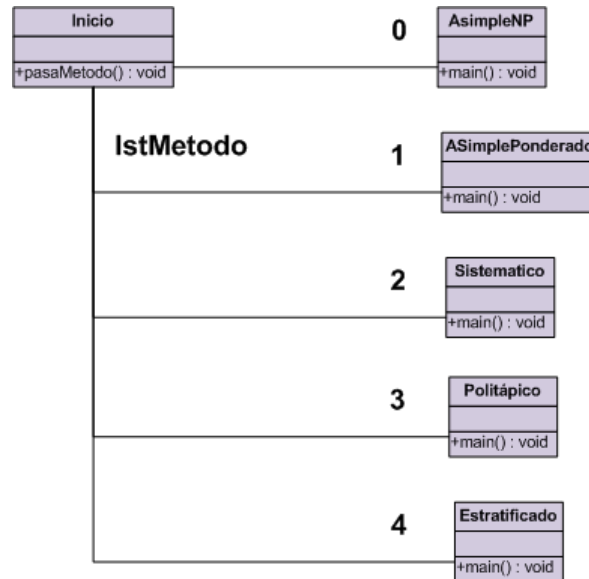


Diagrama 1. Interacción de la Clase Inicio

A continuación se muestra el diagrama de interacción de la clase ASimpleNP del paquete interfaz gráfica, con las demás clases de la aplicación.

Es importante señalar que todas las clases de la interfaz gráfica interactúan de la misma manera con las clases de los paquetes Archivos, Kernel, ModelosTablas y Estadística.

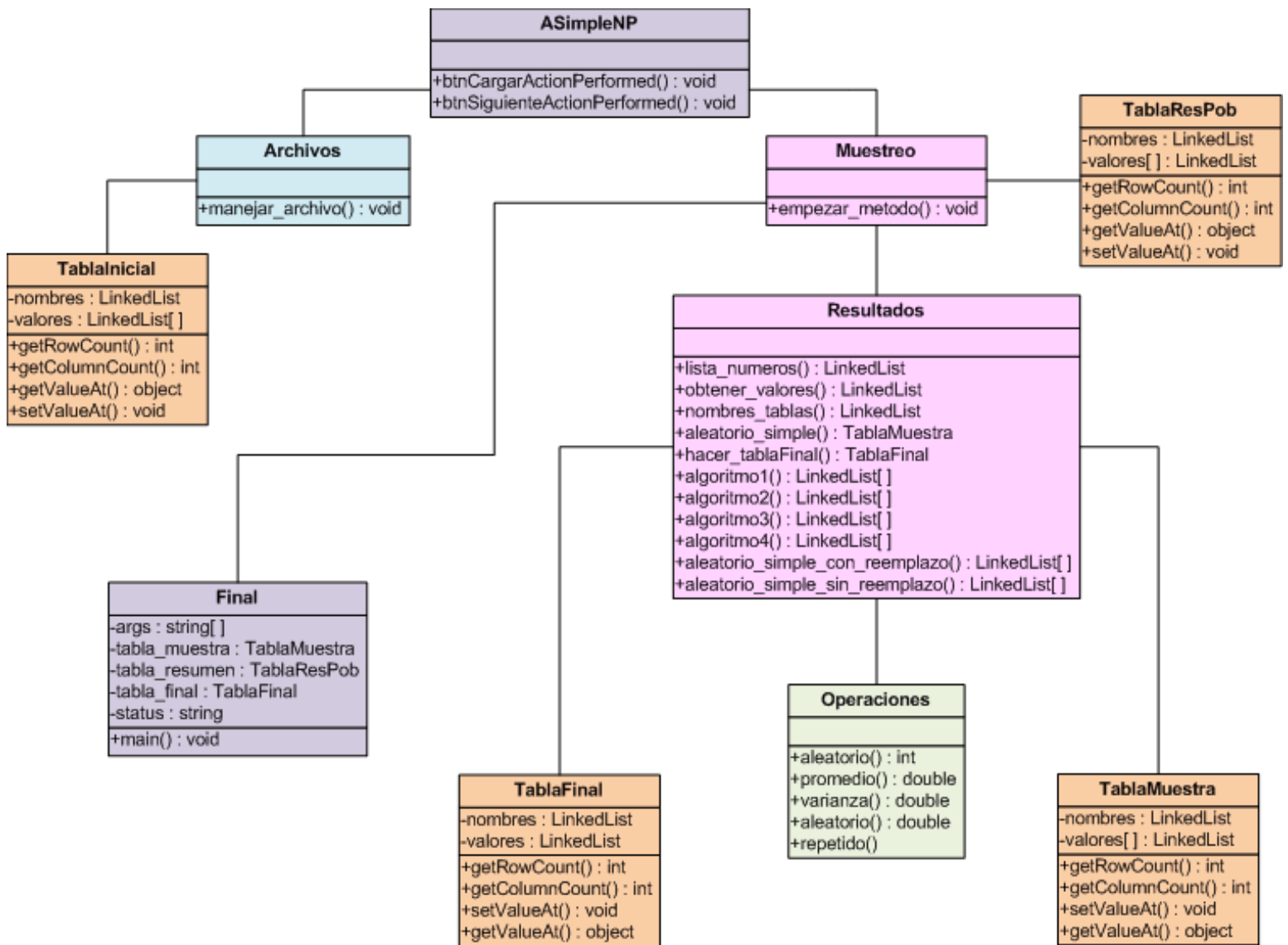


Diagrama 2. Interacción entre las clases del módulo Aleatorio Simple No Ponderado

### 5.3 Descripción Técnica

Esta aplicación se apoya en una técnica de programación orientada a objetos utilizando el lenguaje de programación Java. Se tiene poca experiencia con el lenguaje, pero se investigó que posee una amplia variedad de bibliotecas que facilitan la apertura, escritura y manipulación de archivos, así como para manejar cadenas de entrada a partir de los mismos.

- `java.io.BufferedReader;`
- `java.io.File;`
- `java.io.FileReader;`
- `java.io.FileWriter;`
- `java.io.PrintWriter;`
- `java.util.Scanner;`

Se decidió utilizar estructuras de datos simples para almacenar los datos obtenidos del archivo inicial que proporciona el usuario, tales como listas ligadas, modelos de tablas, arreglos y cadenas.

Las clases utilizan variables globales para no perder referencias a sus miembros que se mandan como parámetro para la comunicación con otras clases.

Lo primero que se diseñó, fue la interfaz gráfica. Se modelaron las pantallas de la aplicación y posteriormente se les dio funcionalidad a los botones.

La nomenclatura utilizada para objetos, métodos y variables es relativa a la función que desempeña. Es decir, para las clases de la interfaz gráfica, el formato de la nomenclatura es el siguiente:

Para las etiquetas, el prefijo `lbl` seguido de un nombre que indica el texto que contiene. Por ejemplo `lblHeader` se refiere a la etiqueta del título del método, `lblTamaño` es la etiqueta que contiene texto que pide al usuario introducir el tamaño de la muestra.

Para las tablas, el prefijo `tbl` seguido de un nombre que indica el contenido de la tabla. Por ejemplo `tblDatos` es el nombre de la tabla que contiene la población inicial, `tblMuestra` es la tabla que contiene los datos de la muestra tomada por la aplicación.

Para los botones, el prefijo `btn` seguido del nombre del botón. Por ejemplo `btnCargar` se refiere al botón que el usuario debe apretar para cargar su archivo en el sistema, `btnGuardar` es el botón para guardar la muestra generada por el sistema en un archivo.

Para los paneles contenedores, el prefijo `pnl` seguido de los elementos que contiene, por ejemplo `pnlPrincipal` contiene a todos los elementos de la pantalla, `pnlSelVar` contiene los elementos donde el usuario selecciona la variable que se oculta o se muestra según la opción seleccionada. Igualmente se tienen prefijos para los menús (`mnu`), submenús (`smnu`), elementos de menú, radio botones (`rbtn`) y checkbox (`chkbx`).

La descripción de cada método de muestreo se describe a continuación. Estos algoritmos han sido publicados en diferentes revistas y libros de texto (consultar las referencias). En caso de que alguien decida usar los algoritmos como han sido programados en este reporte se considerará necesario solicitar el permiso correspondiente a los autores tanto alumno como al profesor responsable de este reporte. El código en Java está en el registro de la propiedad industrial.

### 5.3.1 Método Aleatorio Simple No Ponderado

Para la programación de éste módulo se tomó en cuenta que una muestra aleatoria es aquella sacada de una población de unidades, de manera que todo elemento de la población tenga la misma probabilidad de selección.

Para esto se tomaron 5 algoritmos de selección sin reemplazo y 1 algoritmo para selección de muestra con reemplazo.

Los algoritmos mencionados se describen a continuación.

#### 5.3.1.1 Algoritmo 0. Permutación Aleatoria

El algoritmo 0 ó XCL, como su nombre lo indica, es una propuesta de algoritmo de selección creado por la autora de este proyecto y trabaja de la siguiente manera:

Se tiene almacenado en una lista ligada cada uno de los elementos de la población, por lo que cada elemento tiene un índice.

Se genera una lista de  $N$  números aleatorios en un rango de 0 a  $N$ , siendo  $N$  el tamaño de la población.

Se toman los primeros  $n$  números de la lista, siendo  $n$  el tamaño de la muestra elegido por el usuario. Esta lista representa los índices de la lista ligada donde se tienen almacenados los elementos de la población.

Así se toman los elementos de la población según lo indica esa lista de  $n$  números aleatorios.

El diagrama de flujo del algoritmo se muestra en la figura 5.3.1.1.1

DIAGRAMA DE FLUJO - ALEATORIO SIMPLE NO PONDERADO - ALGORITMO 0

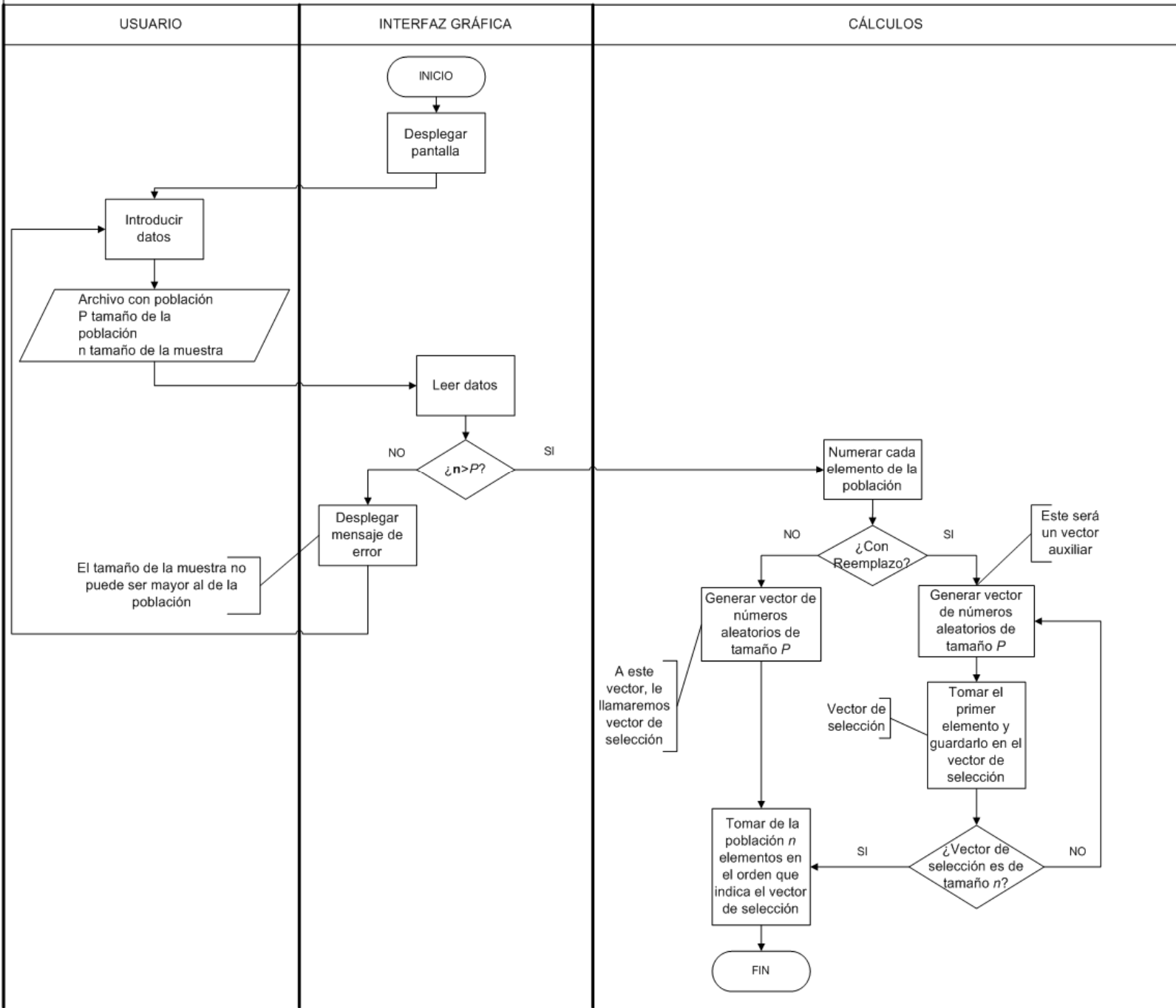


Figura 5.3.1.1.1. Diagrama de flujo del Algoritmo 0 ó XCL

**5.3.1.2 Algoritmo 1. Hipergeométrico**

El algoritmo 1 trabaja de la siguiente manera:

Se tiene una lista de datos que representan una población de tamaño N y el usuario define el tamaño  $n$  para la muestra. Se tiene también una lista ligada que representa el vector con la muestra obtenida.

- Se genera un número aleatorio  $U$ .
- Si  $U > \frac{N}{n}$  entonces se disminuye el tamaño de la población ( $N=N-1$ )
- Si no, entonces se toma el dato y se agrega al vector de la muestra.
- Se repite el primer paso hasta obtener un vector de tamaño  $n$ .

### DIAGRAMA DE FLUJO - ALEATORIO SIMPLE NO PONDERADO - ALGORITMO 1

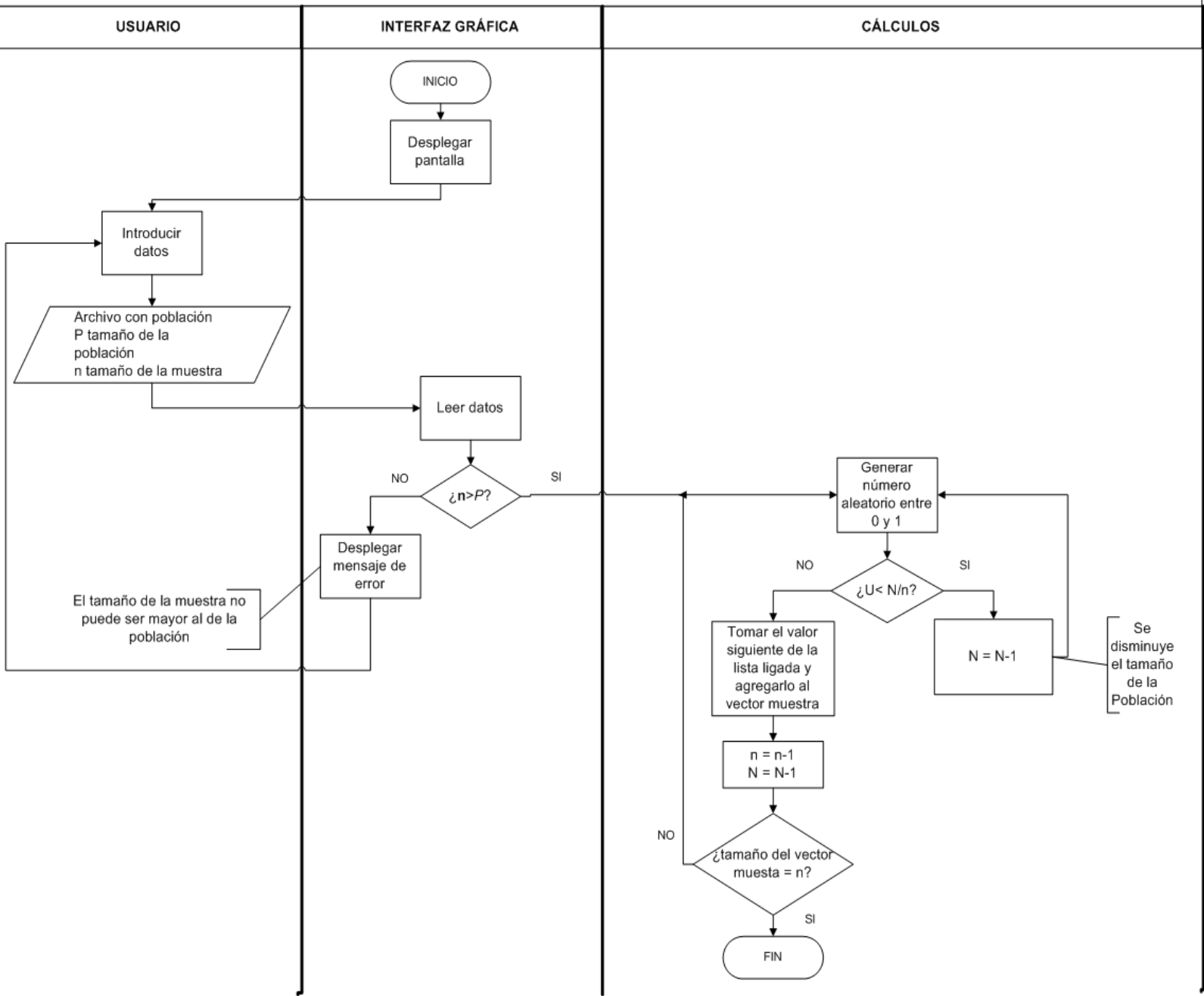


Figura 5.3.1.2.1. Diagrama de Flujo del Algoritmo 1



### 5.3.1.3 Algoritmo 2. Substitución

El algoritmo 2 trabaja de la siguiente manera:

Se tienen 2 listas ligadas. Una lista contiene los elementos de la población y la otra contendrá la muestra elegida.

- Se elige un número  $k$  arbitrario. Para esta implementación, el número  $k = \text{tamaño de la muestra} + 1$
- Se genera un número aleatorio  $U$ .
- Si  $U > \frac{n}{K}$  entonces  $k = k + 1$  hasta que se cumpla la condición.
- Si no, entonces se toma de la lista de la población el elemento  $k$ .
- Se itera hasta obtener  $n$  elementos en la lista ligada que contiene la muestra elegida.

El diagrama de flujo correspondiente se muestra a continuación:

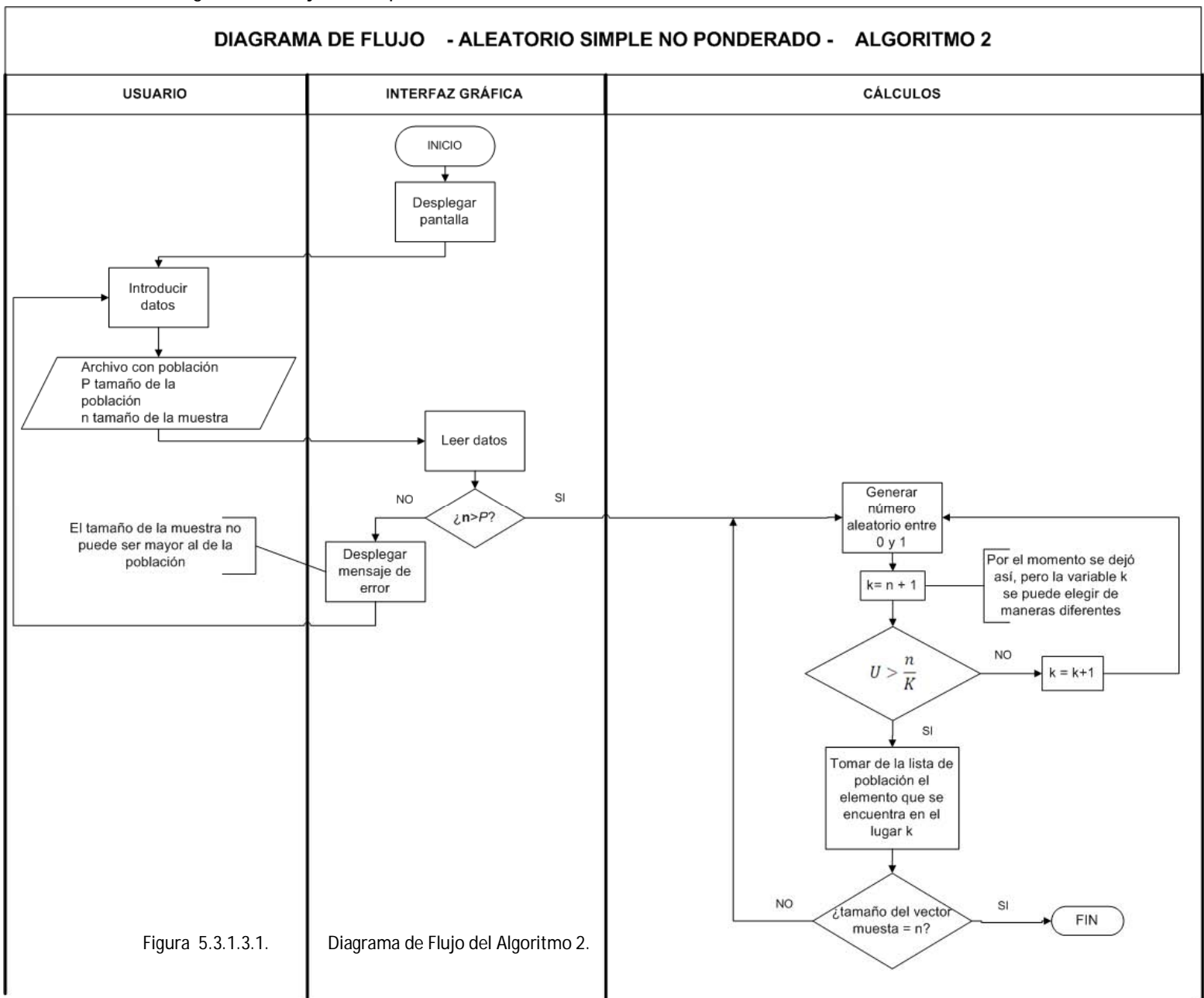


Figura 5.3.1.3.1.

Diagrama de Flujo del Algoritmo 2.

### 5.3.1.4 Algoritmo 3. Saltos Aleatorios A

El algoritmo 3 trabaja con combinaciones para encontrar el número  $S$  más pequeño que cumpla

con la siguiente condición:  $1 - \frac{N-S-1C_n}{NC_n}$ .

Recordemos que  $NC_n = \frac{N!}{n!(N-n)!}$

Donde en este caso,  $N$  es el tamaño de la población y  $n$  el tamaño de la muestra.

Se tiene una lista ligada que contiene los datos de la población y otra lista ligada que contiene la muestra elegida. Para este caso, el número  $S$  representa el índice que se tomará de la lista ligada de población para la muestra. El algoritmo se detalla en el diagrama 5.3.1.4.1.

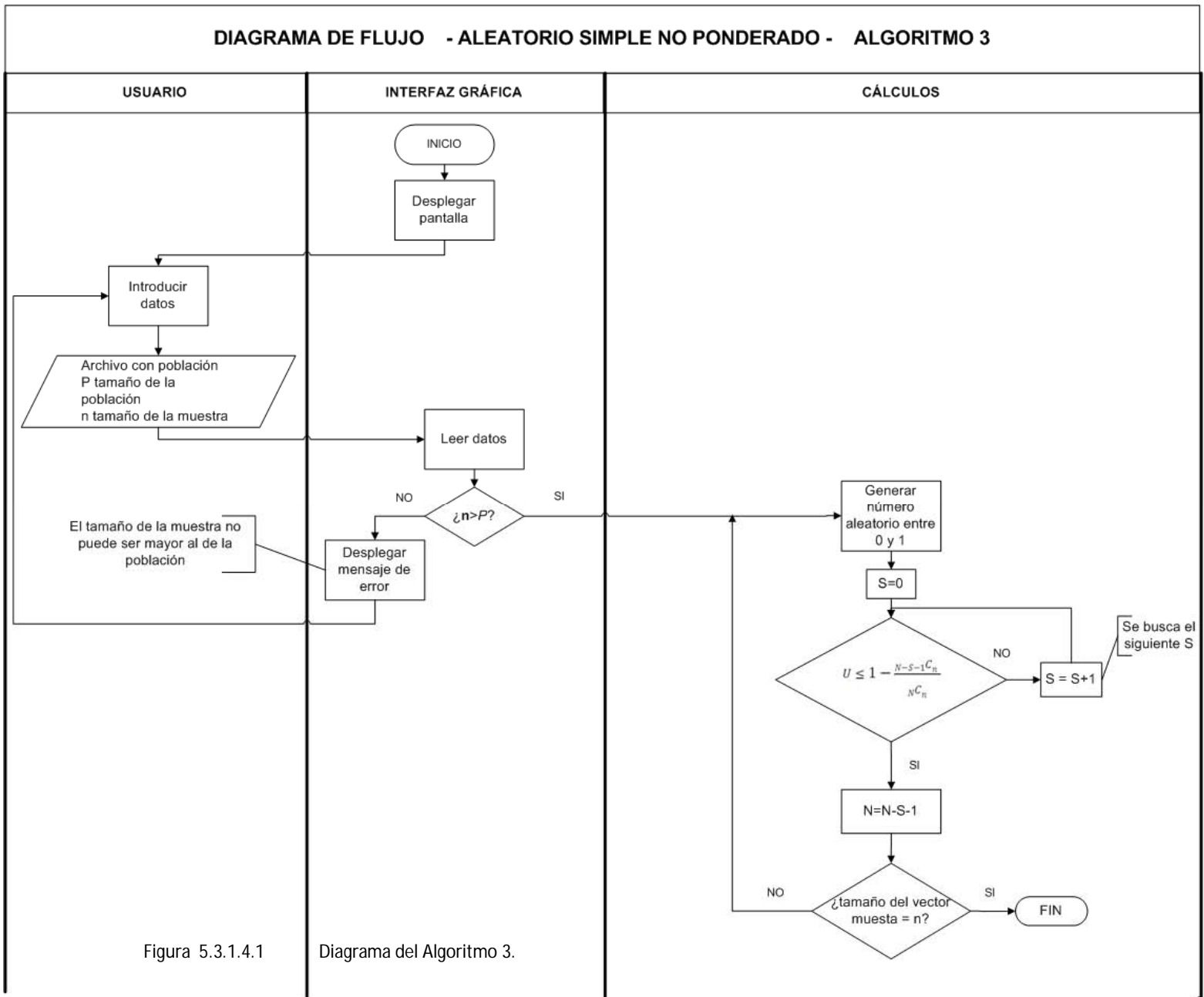


Figura 5.3.1.4.1

Diagrama del Algoritmo 3.

### 5.3.1.5 Algoritmo 4. Saltos Aleatorios B

Este método de selección de muestra consiste en encontrar el mínimo número  $S$  tal que  $S$  cumple con la siguiente condición:

$$S = \text{Min}[(N - i + 1)(U)],$$

Donde  $N$  es el tamaño de la población,  $i$  es el iterador y  $U$  es un número aleatorio.

Se escoge el número más pequeño calculado y ese valor se toma de la lista de población, posteriormente

Se tienen 2 listas ligadas. La primera representa la población y la segunda la muestra. El diagrama de flujo del algoritmo se ilustra con la siguiente figura:

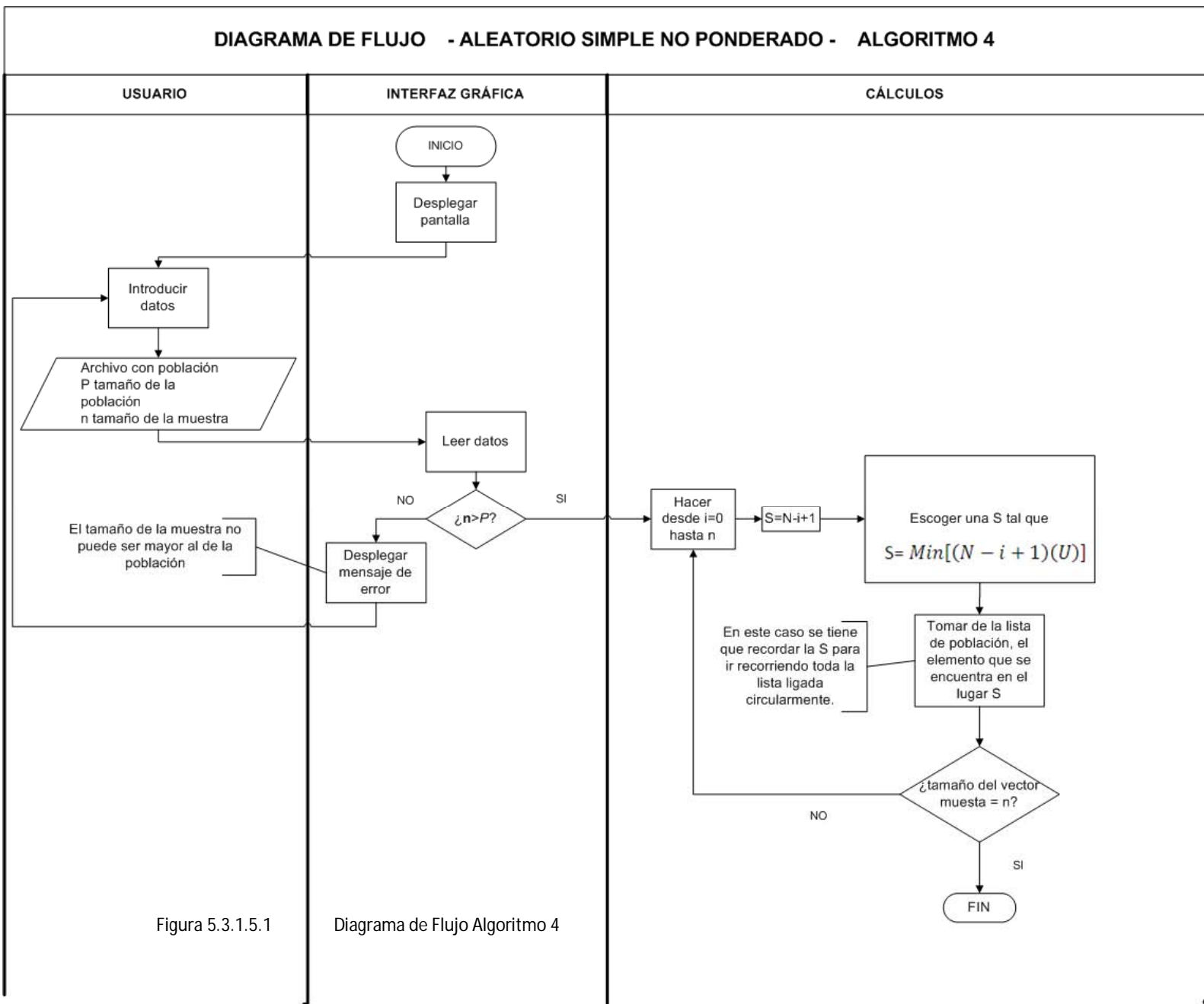


Figura 5.3.1.5.1

Diagrama de Flujo Algoritmo 4

### 5.3.2 Método Sistemático

De manera general, el método Sistemático consiste en seleccionar los individuos, saltando  $k$  individuos, hasta completar  $n$  individuos deseados. Hay que calcular una constante, que se denomina coeficiente de elevación  $k = N/n$ ; donde  $N$  es el tamaño de la población y  $n$  el tamaño de la muestra. Para la primera extracción, hay que elegir al azar un número entre 1 y  $k$ ; de ahí en adelante tomar uno de cada  $k$  a intervalos regulares.

Esto quiere decir que si tenemos un determinado número de personas que es la población ( $N$ ) y queremos escoger de esa población un número más pequeño el cual es la muestra ( $n$ ), dividimos el número de la población por el número de la muestra que queremos tomar y el resultado de esta operación será el intervalo, entonces escogemos un número al azar desde uno hasta el número del intervalo, y a partir de este número escogemos los demás siguiendo el orden del intervalo.

Para el desarrollo del módulo método sistemático, se realizó el siguiente procedimiento:

Se tiene una lista ligada circular de  $N$  individuos, un tamaño de muestra  $n$ , un intervalo de salto  $k$  y un inicio  $p$  dentro de un intervalo entre 1 y  $k$ . Los valores de  $k$  y  $p$  pueden ser calculados, aleatorios o definidos previamente por el usuario. Posteriormente se seleccionan los individuos de la lista hasta completar  $n$ .

El algoritmo se programó de la siguiente manera:

Se tienen 2 listas ligadas. Una lista contiene los datos de la población y otra lista representa a la muestra seleccionada.

1. Numerar cada elemento de la población
2. Tomar de la lista de población el elemento que ocupa el lugar  $k$
3. Guardar el elemento seleccionado en la lista de la muestra
4. Tomar el elemento  $k+p$  de la lista de población hasta que la lista de la muestra tenga longitud  $n$

El diagrama 5.3.4.1 ilustra el algoritmo anterior

## DIAGRAMA DE FLUJO - SISTEMÁTICO -

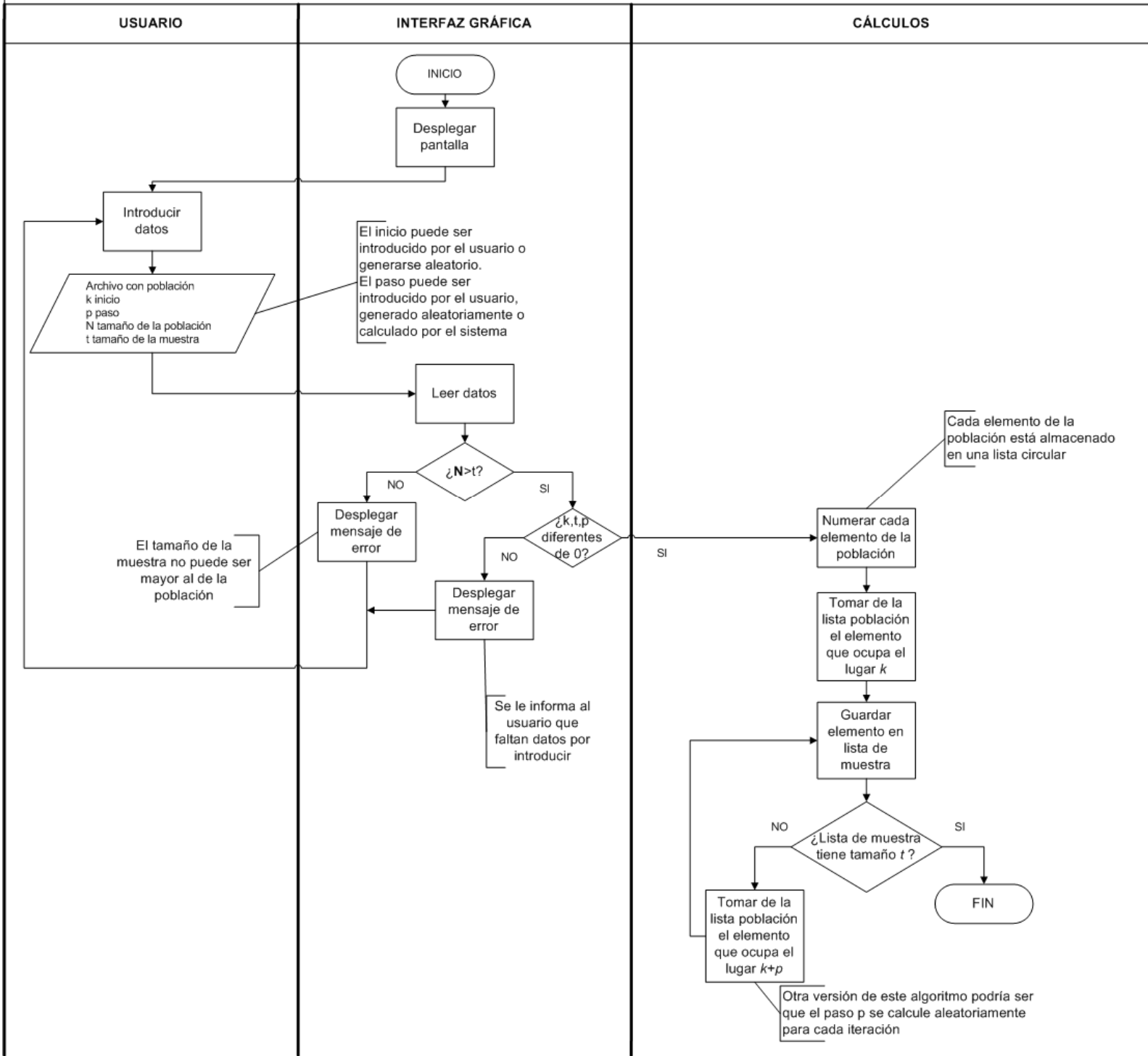


Figura 5.3.4.1. Diagrama Método Sistemático

### 5.3.3 Método Estratificado

La idea general del método consiste en la división previa de la población en grupos o clases que se suponen homogéneos con respecto a alguna característica de las que se van a estudiar. A estos grupos se les denomina estratos. A cada uno de estos se le asigna un código que determinará el número de miembros del mismo que compondrán la muestra.

Dentro de cada estrato se usa algún algoritmo del método de muestreo aleatorio simple no ponderado para obtener la muestra, por lo que este método es el más elaborado.

Cuenta con 2 módulos, uno para codificar la variable en caso de que el usuario así lo requiera, el módulo que separa en estratos y el módulo de selección del tamaño de la muestra.

Según la cantidad de elementos de la muestra que se han de elegir de cada uno de los estratos, existen tres opciones para el muestreo estratificado:

- **Asignación proporcional:** el tamaño de la muestra dentro de cada estrato es proporcional al tamaño del estrato dentro de la población.
- **Asignación óptima:** la muestra recogerá más individuos de aquellos estratos que tengan más variabilidad. Para ello es necesario un conocimiento previo de la población.
- **Asignación manual:** el usuario es responsable de elegir el tamaño de la muestra para cada estrato.

Este módulo tiene contemplados 3 casos de uso: el usuario quiere codificar sus variables en tiempo real, quiere que sus variables sean codificadas por promedio o bien, ya tiene un archivo que contiene los códigos junto con la población.

El primer caso de uso se ilustra en la figura 5.3.3.1, donde se le muestra al usuario una tabla donde están colocados sus datos iniciales y una columna en blanco para que introduzca manualmente los códigos de su elección. Cabe mencionar que el sistema en este caso acepta letras u otros símbolos para la división de los estratos.

Para el segundo caso, se calcula el promedio de la población y se codifica a los elementos con un '0' para aquellos valores que se encuentran por debajo del promedio calculado y con un '1' los valores que se encuentran por arriba. Se separan en una tabla, el usuario tiene la opción de invertir dicha codificación y posteriormente escoger el tamaño para las muestras y el algoritmo de selección. La figura 5.3.3.2 ilustra el caso promedio.

En el tercer caso, donde el usuario tiene sus estratos definidos en el mismo archivo inicial, se leen los datos, se le pregunta al usuario acerca del tamaño de las muestras como en el caso manual y se separan los grupos para posteriormente llamar al método de selección correspondiente. La figura 5.3.3.3 ilustra el caso archivo.

**DIAGRAMA DE FLUJO - ESTRATIFICADO MANUAL-**

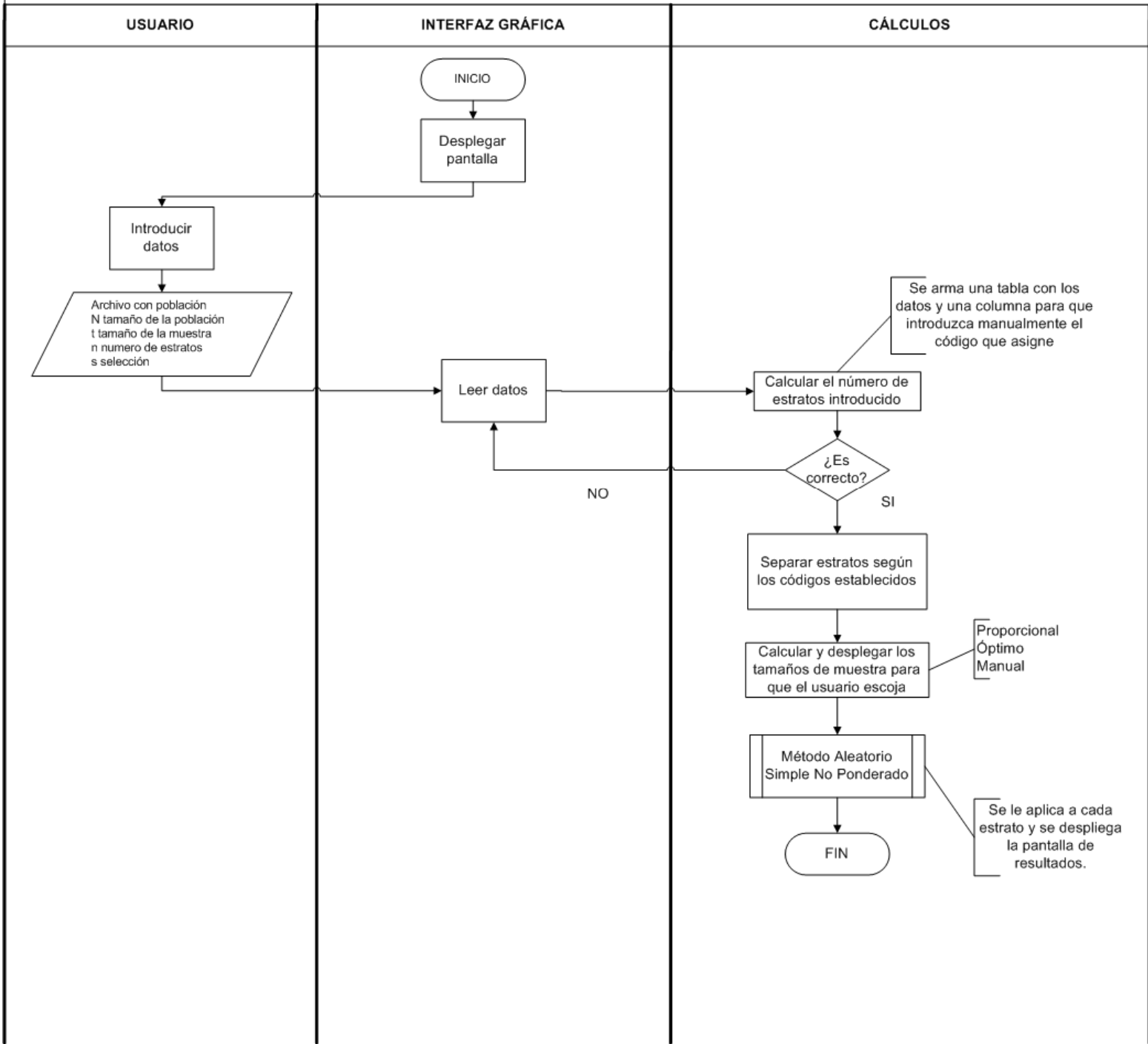


Figura 5.3.3.1. Diagrama de flujo. Método Estratificado opción Manual

## DIAGRAMA DE FLUJO - ESTRATIFICADO PROMEDIO-

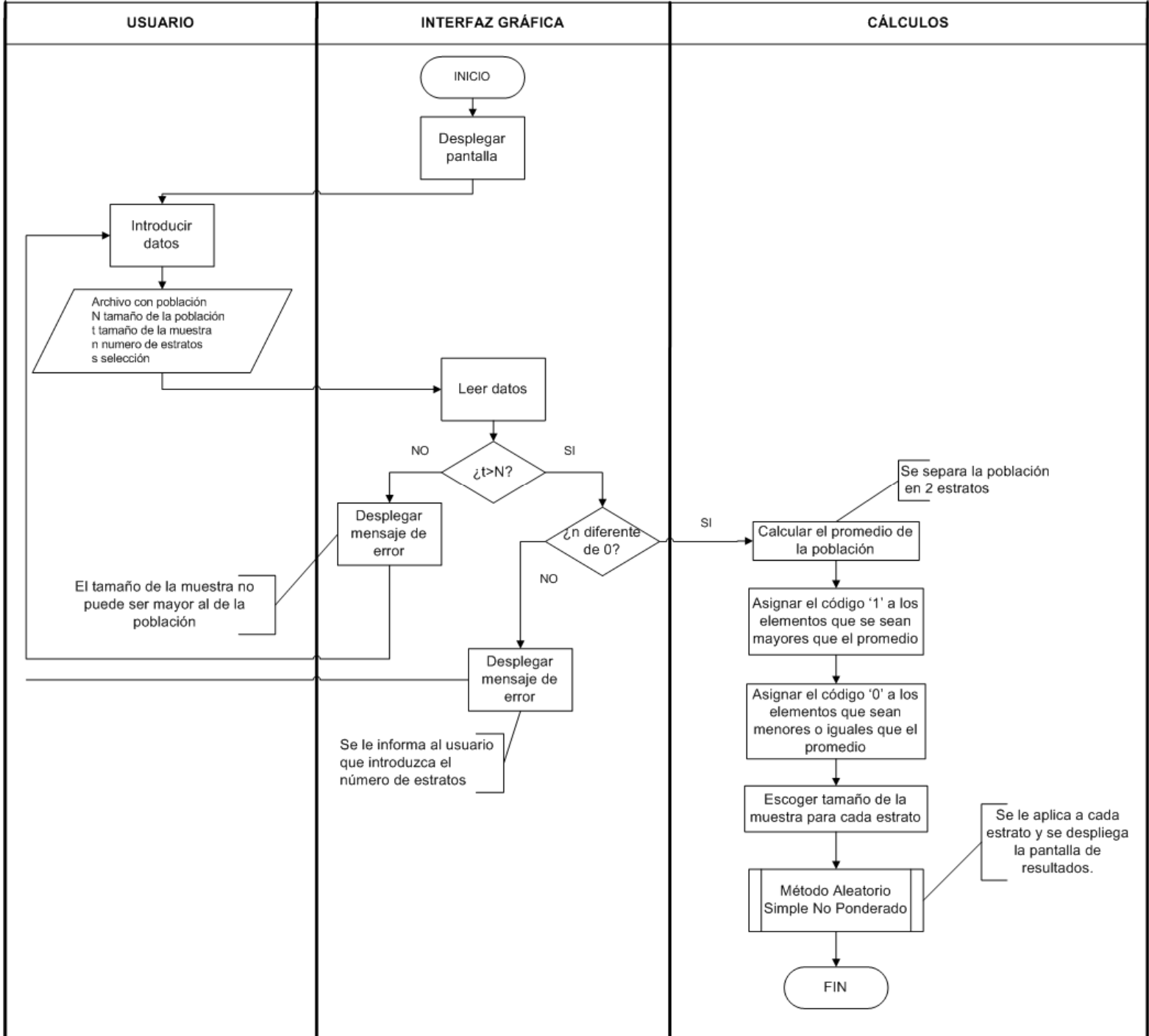


Figura 5.3.3.2 Diagrama de flujo Método Estratificado opción Promedio



## DIAGRAMA DE FLUJO - ESTRATIFICADO ARCHIVO-

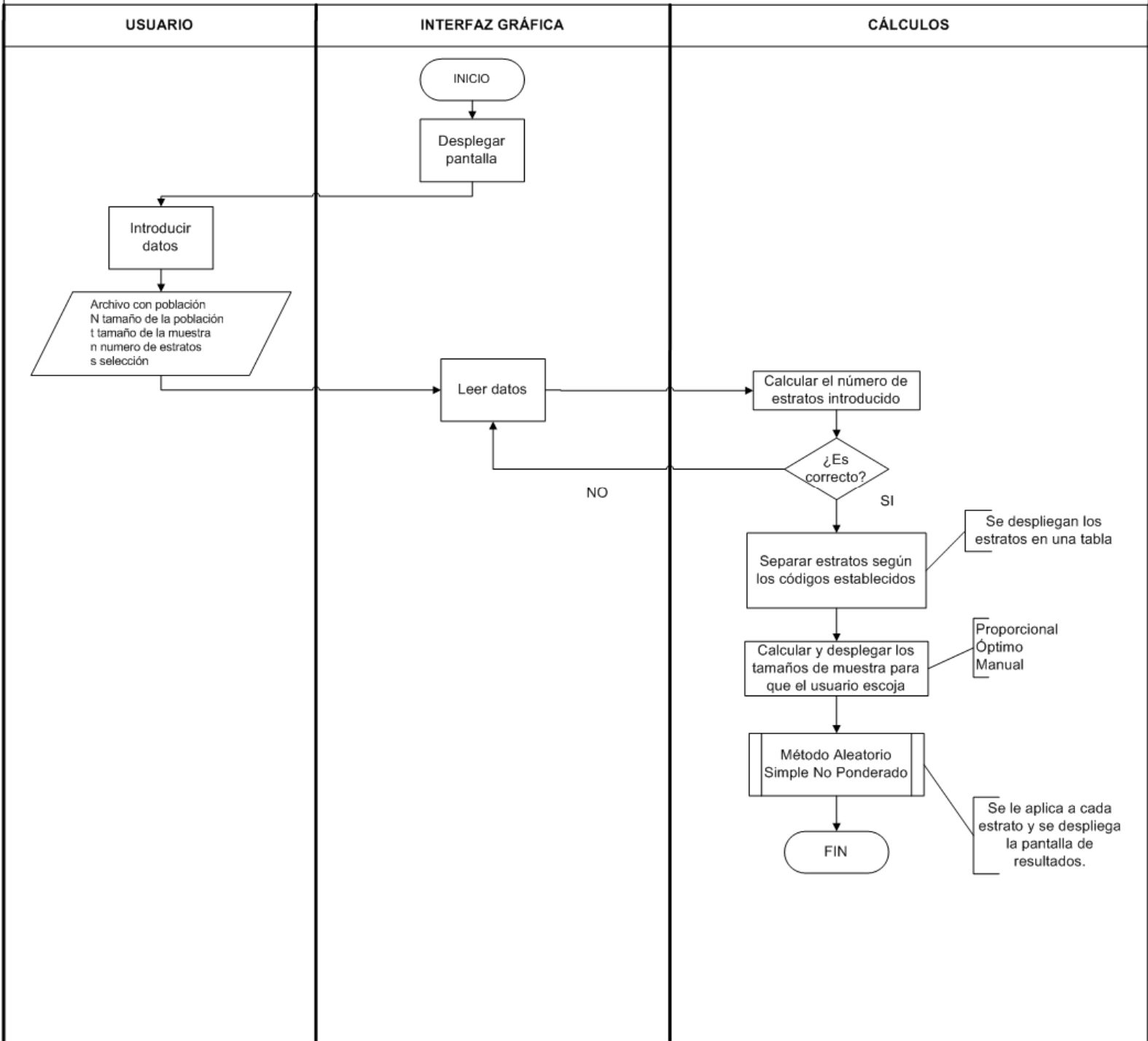


Figura 5.3.3.3. Diagrama de Flujo. Método Estratificado archivo.

### 5.3.4 Método Aleatorio Simple Ponderado

Este método consiste en elegir aleatoriamente a los elementos de una población, donde cada elemento tiene probabilidades diferentes de ser elegido. El elemento con una probabilidad más alta, será elegido más veces.

Este módulo presenta 2 casos de uso: cuando el usuario introduce los datos junto con sus probabilidades o tiene una tabla de frecuencias.

Para el caso de la lista de probabilidades, se ordenan de menor a mayor las probabilidades y se genera una ponderación de las mismas. Posteriormente se genera un número aleatorio y se compara con el vector de ponderaciones. Se elige el dato cuyo entre en el rango correspondiente hasta obtener  $n$  datos, siendo  $n$  el tamaño de la muestra.

Cabe mencionar que, para este método en particular se encuentra programado en la clase Operaciones, el algoritmo de ordenamiento quicksort que, por ser un algoritmo basado en la técnica de divide y vencerás, permite, en promedio, ordenar  $n$  elementos de una lista en un tiempo proporcional a  $n \log n$ , lo que garantiza que el tiempo de ejecución del método no se vea afectado en caso de tener una lista con datos mayores a 10000. En este caso se permite que el tamaño de la muestra sea mayor al tamaño de la población.

La figura 5.3.4.1 ilustra el caso de uso de lista de probabilidades.

Para el caso de tabla de frecuencias, se tiene el dato y la frecuencia con la que se repite, por lo que primero se tiene que desglosar la población, para después hacer la selección. En este caso, después de haber desglosado la tabla, se le da a escoger al usuario entre los 5 algoritmos de selección del método aleatorio simple no ponderado.

La figura 5.3.4.2 ilustra el caso de uso de tabla de frecuencias.

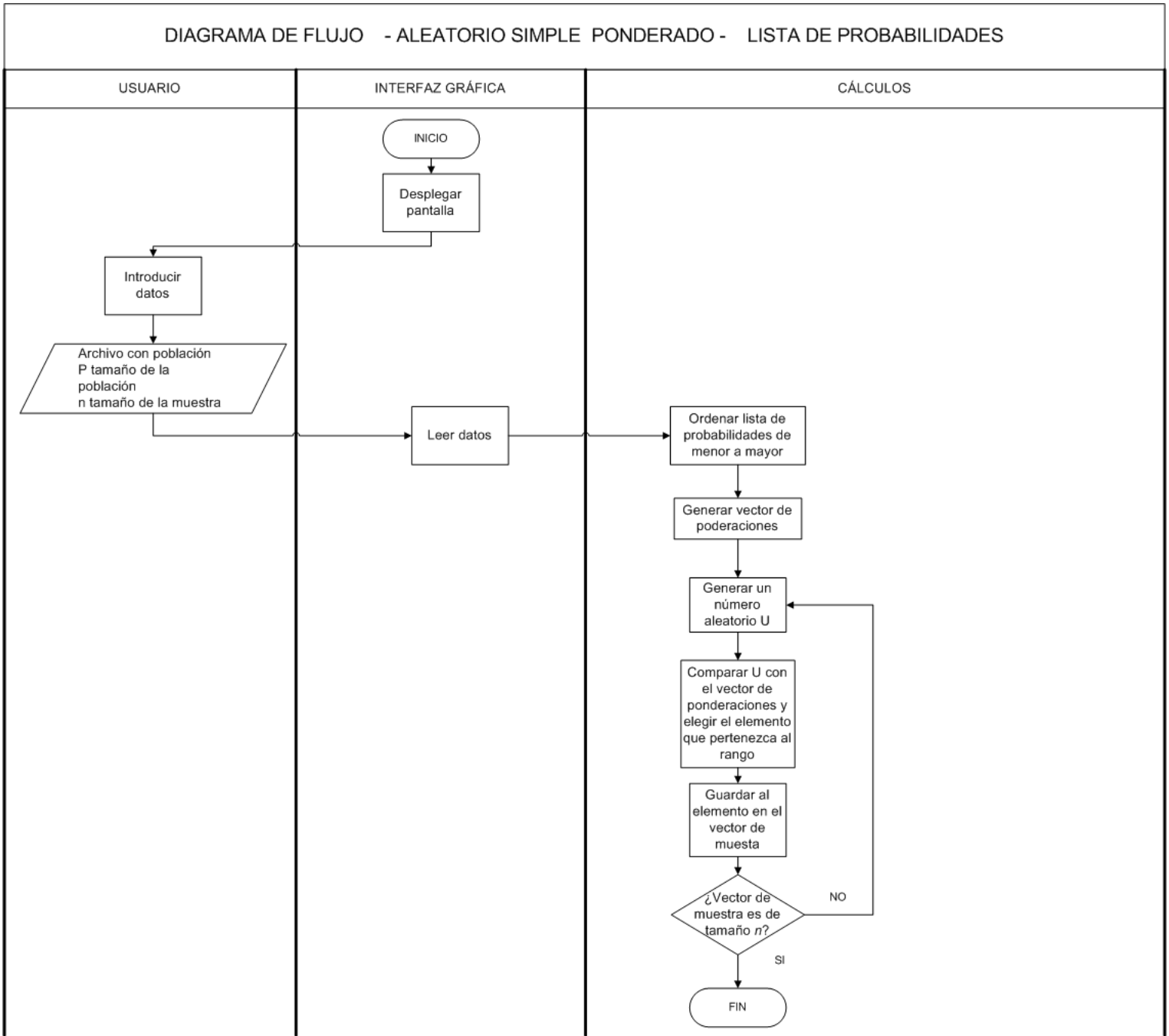


Figura 5.3.4.1. Diagrama de Flujo. Aleatorio Simple Ponderado. Lista de Probabilidades.

DIAGRAMA DE FLUJO - ALEATORIO SIMPLE PONDERADO - TABLA DE FRECUENCIAS

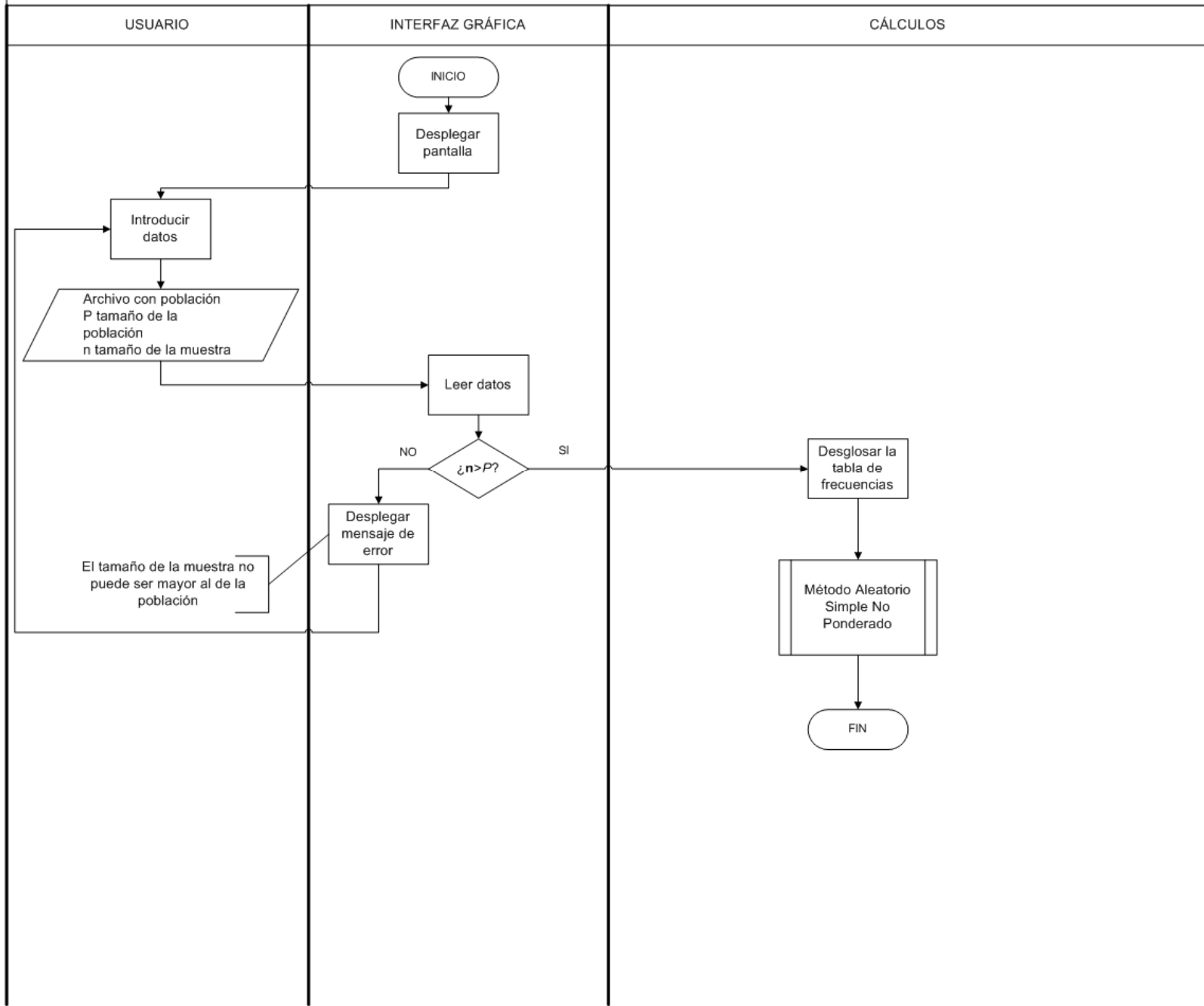


Figura 5.3.4.2. Diagrama de Flujo. Método Simple Ponderado. Tabla de Frecuencias.

## 5.4 Recomendaciones

- Para el uso del Algoritmo 3, que usa combinaciones, se recomienda que el tamaño de la muestra no contenga más de 20 individuos, ya que, al utilizar la función factorial, se puede saturar la memoria y causar un error del tipo *stack-overflow* por la cantidad de multiplicaciones y sumas utilizadas.
- Para el caso de los archivos de ayuda, es importante detectar la ruta en donde se instala la aplicación, ya que el programa abre una ruta específica, pero debe tener la cadena de la ruta verdadera o completa desde la raíz. Es decir, se ha dejado en `\NetBeansProjects\PT02\src\pt02\archivos`, los archivos que contienen el contenido de la ayuda numerados, así el desarrollador podrá cambiar el texto del mismo, ya que lee los archivos en tiempo real. Solamente falta completar la ruta verdadera para que el programa pueda abrirlos.
- Los archivos que se encuentran en la carpeta `resources`, pueden modificarse para cambiar apariencia, imágenes, etiquetas, tipo de letras, tamaños, etc, para el caso que se requiera cambiar el diseño y colores de la aplicación. En el método  `initComponents()` se indica de donde se tomará dicho archivo.
- Para agregar un algoritmo al método aleatorio simple no ponderado, primero debe agregarse en el combobox *ComboAlgoritmos*, el nombre del nuevo método. Este combobox se encuentra en el formulario *ASimpleNP*.

Posteriormente, en el método `empezar_método` de la clase `Muestreo`, se modificará la siguiente función:

```
public void empezar_metodo(int idMetodo,String nombre,String confianza)
```

El método `Aleatorio Simple No Ponderado` tiene el `idMetodo=0`, por lo que dentro del `switch` se buscará el caso 0. Dentro de ese caso hay una variable que se llama *reemplazo*. Dicha variable es la que controla los índices de los algoritmos del 0 al 4. Para agregar otro algoritmo, se debe iniciar con el número 5 y agregarlo a ese `switch` (línea 106, clase `Muestreo`).

```

if(reemplazo==0)
{
    switch(alg.getSelectedIndex())
    {
        case 0: reemplazo=0; //algoritmo xcl
            break;
        case 1: reemplazo=1; //algoritmo 1
            break;
        case 2: reemplazo=2; // algoritmo 2
            break;
        case 3: reemplazo=3; //algoritmo 3
            break;
        case 4: reemplazo=4; //algoritmo 4
            break;
        default:break;
    }
}

```

Figura 7.1 Agregar un nuevo algoritmo al método aleatorio simple no ponderado

Una vez agregado, en la línea 124 se manda llamar al método *aleatorio\_simple* de la clase Resultados, donde el parámetro *reemplazo* también será comparado en un *switch* para realizar la acción correspondiente. Es ahí donde debemos agregar el código que utilizará el nuevo algoritmo.

```

public TablaMuestra aleatorio_simple(String nombre,int tamP,LinkedList datos,String tamM,int
status)

```

En la línea 115 de la clase Resultados, se agregará un nuevo case al *switch* el código del nuevo algoritmo, en este caso el case 6.

```

switch(status)
{
    case 0: //SIN REEMPLAZO
        //GENERAR P NÚMEROS ALEATORIOS ENTEROS
        mst[0]=operaciones.aleatorio(tamP); //genero los n número aleatorios (índices)
        mst[1]=datos; //datos a muestrear
        //ASIGNAR A CADA DATO DE LA POBLACIÓN UN NÚMERO
        TablaMuestra tablamuestra=new TablaMuestra(names,mst);
        //TOMAR LOS PRIMEROS N DATOS DE LA POBLACIÓN
        buena=aleatorio_simple_sin_reemplazo(tablamuestra,tamM,mst[0]);
        TablaMuestra nice=new TablaMuestra(names,buena);
        return nice;

    case 1: //SIN REEMPLAZO utilizando el algoritmo 1
        buena=algoritmo1(datos, Integer.parseInt(tamM.trim()), tamP-1);
        TablaMuestra resalg1=new TablaMuestra(names,buena);
        return resalg1;

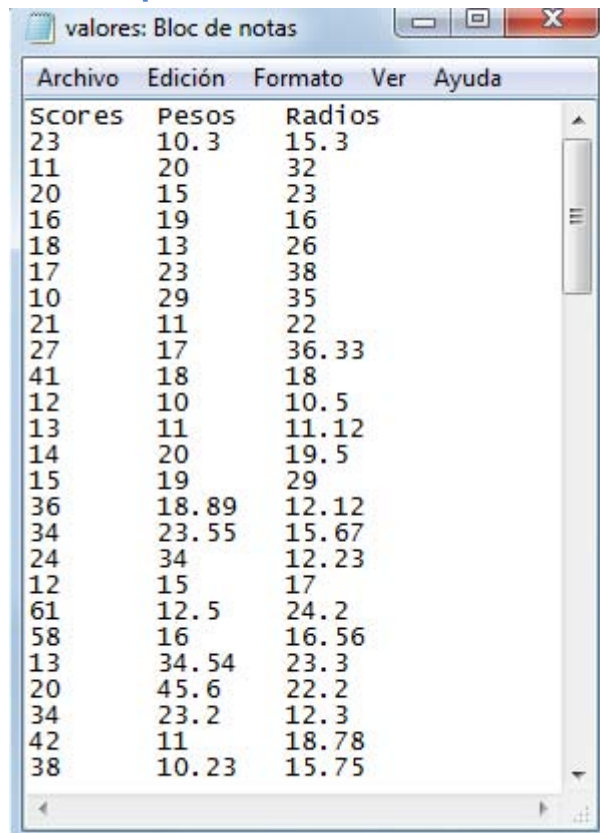
    case 2: //SIN REEMPLAZO utilizando el algoritmo 2
        buena=algoritmo2(datos, Integer.parseInt(tamM.trim()), tamP-1);
        TablaMuestra resalg2=new TablaMuestra(names,buena);
        return resalg2;

```

## 6 Pruebas

Para las pruebas se utilizaron diferentes archivos iniciales dependiendo del método de muestreo. En el caso de la población, los cálculos de promedio y varianza se realizaron en Excel con el fin de comprobar que sean correctos los calculados por el sistema. Cabe mencionar que, dado que no podemos replicar los números aleatorios obtenidos en cada corrida del programa, para el muestreo solo se tomará en cuenta que el sistema tome los índices correctos en las listas ligadas.

### 6.1 Método Aleatorio Simple No Ponderado



Scores	Pesos	Radios
23	10.3	15.3
11	20	32
20	15	23
16	19	16
18	13	26
17	23	38
10	29	35
21	11	22
27	17	36.33
41	18	18
12	10	10.5
13	11	11.12
14	20	19.5
15	19	29
36	18.89	12.12
34	23.55	15.67
24	34	12.23
12	15	17
61	12.5	24.2
58	16	16.56
13	34.54	23.3
20	45.6	22.2
34	23.2	12.3
42	11	18.78
38	10.23	15.75

Figura 6.1.1. Archivo Inicial para el método Simple No Ponderado

Para esta prueba se tomará en cuenta la variable *pesos*. Se tomará una muestra de 15 individuos utilizando el algoritmo 2 del método de selección sin reemplazo. Los resultados son los siguientes:



	A	B	C	D
1		Scores	Pesos	Radios
2		23	10.3	15.3
3		11	20	32
4		20	15	23
5		16	19	16
6		18	13	26
7		17	23	38
8		10	29	35
9		21	11	22
10		27	17	36.33
11		41	18	18
12		12	10	10.5
13		13	11	11.12
101		38	10.23	15.75
102	promedio	25.2	19.1924	20.8744
103	varianza	198.707071	75.932673	62.8791764
104				

Figura 6.1.2. Resultados de los cálculos de la población





Figura 6.1.3. Resultados del muestreo.

Y finalmente, el archivo creado de la muestra es el siguiente:

```

ooooooooooooooooooooo: Bloc de notas
Archivo Edición Formato Ver Ayuda
Archivo Creado el Tue May 03 02:26:36 CDT 2011

++++Aleatorio Simple No Ponderado++++

Sin Reemplazo: Algoritmo 2

Nombre del Archivo: valores.txt
Tamaño de la Población: 100

****MUESTRA****

Tamaño de la Muestra: 15

    Pesos
    34.0
    12.5
    16.0
    34.54
    45.6
    23.2
    20.0
    17.0
    18.89
    45.6
    29.0
    20.0
    34.0
    20.0
    19.0
####ESTIMACIONES DE LA MUESTRA####

Límite Inferior:      642.7757488217762
Límite Superior:     748.3626262339918
Error Estimación:    5.1820341857139605
Promedio:            25.955333333333332
Varianza:            103.79091822222223

Nota: Este es el archivo de prueba aleatorio simple

```

Figura 6.1.4. Archivo Creado

## 6.2 Aleatorio Simple Ponderado

Para esta prueba utilizaremos un archivo inicial que contenga las probabilidades de cada individuo. Se tomará como correcto, si el sistema elige más veces al individuo que tiene la probabilidad mayor. Se tomará una muestra de 15 individuos. En este ejemplo, el número con más probabilidad de elección es el número 16.

Alturas	Probabilidades
12.3	0.01
11.2	0.07
13.45	0.02
16	0.5
10.7	0.4
13.16	0.1

Figura 6.2.1. Archivo Inicial. Simple Ponderado

ID	Alturas
2	13.45
4	10.7
2	13.45
2	13.45
3	16.0
2	13.45
3	16.0
3	16.0
3	16.0
3	16.0
2	13.45
3	16.0
3	16.0
3	16.0
3	16.0
3	16.0
3	16.0

Figura 6.2.2. Resultados muestreo.

### 6.3 Método Sistemático

Para la prueba de este método se utilizará el mismo archivo del método simple no ponderado. La variable utilizada será radios. Se escogerá un tamaño de muestra de 20 con un inicio en 10 tomando el salto de 4 en 4.

valores: Bloc de notas

Archivo Edición Formato Ver Ayuda

Scores	Pesos	Radios
23	10.3	15.3
11	20	32
20	15	23
16	19	16
18	13	26
17	23	38
10	29	35
21	11	22
27	17	36.33
41	18	18
12	10	10.5
13	11	11.12
14	20	19.5
15	19	29
36	18.89	12.12
34	23.55	15.67
24	34	12.23
12	15	17
61	12.5	24.2
58	16	16.56
13	34.54	23.3
20	45.6	22.2
34	23.2	12.3
42	11	18.78
38	10.23	15.75

Figura 6.3.1. Archivo de Inicio sistemático

Población Muestra

Población:

Variable Elegida	Tipo	Mínimo	Máximo	Promedio	Varianza	Desviación Std
Radios	numérico	10.5	38.0	20.874399999...	62.250384639..	7.8898912945...

Figura 6.3.2. Pantalla de resultados del sistema

	A	B	C	D
1		Scores	Pesos	Radios
2		23	10.3	15.3
3		11	20	32
4		20	15	23
5		16	19	16
6		18	13	26
7		17	23	38
8		10	29	35
9		21	11	22
10		27	17	36.33
11		41	18	18
12		12	10	10.5
13		13	11	11.12
101		38	10.23	15.75
102	promedio	25.2	19.1924	20.8744
103	varianza	198.707071	75.932673	62.8791764
104				

Figura 6.3.3 Cálculos hechos en excel

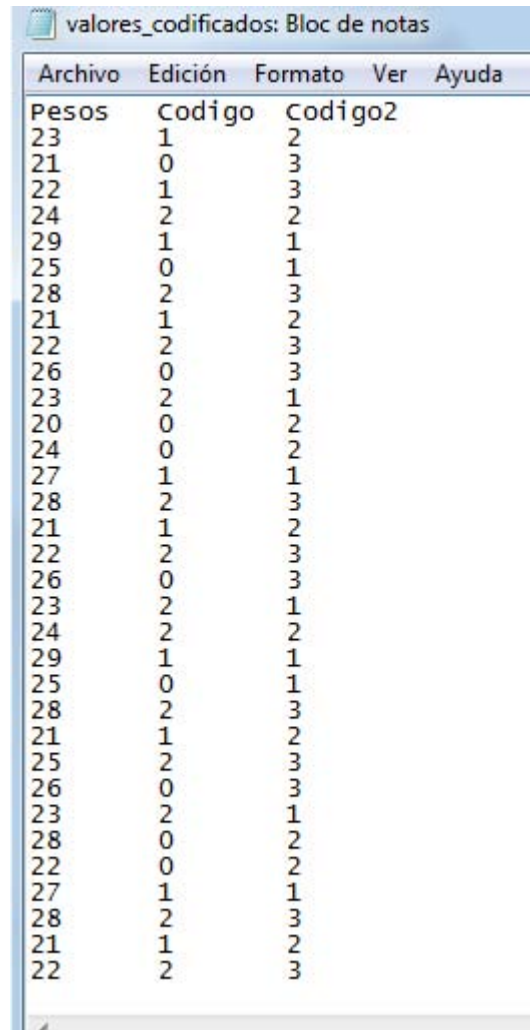
The image shows a software interface with two main panels. On the left, a list titled 'Radios' contains various numerical values. On the right, a pink panel titled 'Muestra' (Sample) contains a table with two columns: 'ID' and 'Radios'. The 'Muestra' table lists several rows of data. In both panels, several values are circled in red to highlight specific data points.

ID	Radios
10	10.5
14	12.12
18	24.2
22	12.3
26	32.0
30	38.0
34	18.0
38	29.0
42	17.0
46	22.2
50	15.3
54	26.0
58	36.33
62	19.5
66	12.23
70	23.3
74	15.75
78	16.0

Figura 6.3.4. Muestra

## 6.4 Método Estratificado

Para este método se utilizó un archivo inicial que contiene la codificación de los estratos. Son 3 con códigos 0, 1 y 2. Las muestras se tomarán con el algoritmo 0 del método aleatorio simple sin reemplazo.



Pesos	Codigo	Codigo2
23	1	2
21	0	3
22	1	3
24	2	2
29	1	1
25	0	1
28	2	3
21	1	2
22	2	3
26	0	3
23	2	1
20	0	2
24	0	2
27	1	1
28	2	3
21	1	2
22	2	3
26	0	3
23	2	1
24	2	2
29	1	1
25	0	1
28	2	3
21	1	2
25	2	3
26	0	3
23	2	1
28	0	2
22	0	2
27	1	1
28	2	3
21	1	2
22	2	3

Figura 6.4.1. Archivo Inicial. Método Estratificado.

La muestra será de tamaño 20 y los tamaños de muestra de cada estrato se escogerán proporcionalmente.

**Datos:**

1	0	2
23.0	21.0	24.0
22.0	25.0	28.0
29.0	26.0	22.0
21.0	20.0	23.0
27.0	24.0	28.0
21.0	26.0	22.0
29.0	25.0	23.0
21.0	26.0	24.0
27.0	28.0	28.0
21.0	22.0	25.0
		23.0
		28.0
		22.0

Figura 6.4.2. Separación de estratos.

**ESTRATOS**

Estrato 1	Estrato 2	Estrato 3
21	21	24
27	25	23
29	24	22
29	25	24
21	28	23
23	26	23
		22
		28

Figura 6.4.3. Resultados del muestreo.

## 7 Conclusiones

La aplicación computacional para la toma de muestras de una población, es una aplicación útil, sencilla y amigable para el usuario. Utiliza los métodos de muestreo más utilizados con una amplia variedad de algoritmos de selección de muestra con y sin reemplazo y ofrece la posibilidad de trabajar con variables continuas o con proporciones.

Tomando en cuenta la poca experiencia que se tiene en el lenguaje de programación Java para el diseño y codificación de la interfaz gráfica, se considera que el desarrollo del proyecto ayudó a reforzar conocimientos previos de programación orientada a objetos, programación orientada a eventos y modelado de clases, además de aportar conocimientos nuevos en aspectos como sobrecarga de métodos y paso de parámetros.

Se considera que Java fue un lenguaje adecuado para el desarrollo del proyecto ya que facilita la creación y ensamblado de módulos y el IDE Netbeans fue muy útil para el modelado y programación de la interfaz gráfica.

Se considera también que la arquitectura de la aplicación no es la óptima, pues hay clases y métodos que pudieron organizarse de manera más general, por ejemplo en la parte de despliegue de resultados, se tienen 3 clases de interfaz gráfica, ya que en cada método se utilizan parámetros diferentes del despliegue de tablas, paneles y etiquetas, que por falta de dominio del lenguaje no fue posible juntar en una sola pantalla.

Algunas posibles mejoras u oportunidades de desarrollo que existen para esta aplicación son las siguientes:

- Se puede trabajar en el módulo de lectura de archivos, para que la aplicación acepte más formatos de archivo inicial, ya sean .csv, .dat, .xls o cualquier otro formato deseado.
- Se puede trabajar en un módulo para idiomas, es decir, que las etiquetas utilizadas en la interfaz gráfica tengan traducción para diferentes idiomas. Se puede realizar mediante los archivos que se encuentran en la carpeta *resources*.
- Se puede implementar el uso interno de la tabla Z, ya que actualmente es el usuario quien introduce el valor de los cuantiles. En este caso se recomienda utilizar tablas hash como estructura de datos para almacenar los valores de dicha tabla Z y/o t de student.
- Se puede trabajar en mejorar la interfaz gráfica por medio de *skins* que permitan el cambio de apariencia del mismo, en caso de que el desarrollador lo considere necesario.



## 8 Bibliografía

- [1] Ceballos, Javier, 2006, "Java 2. Interfaces Gráficas y Aplicaciones para Internet", Primera Edición, Alfaomega Grupo Editor, ISBN: 9701511808.
- [2] Ceballos, Javier, 2007, "Java 2. Lenguaje y Aplicaciones", Primera Edición, Alfaomega Grupo Editor, ISBN: 9789701512494
- [3] Devore., J., 2005, "Probabilidad y Estadística para Ingeniería y Ciencias", Thomson Learning,
- [4] GrosBras Jean-Marie, "Methodes Statistiques des Sondages", Editorial Economica, 1987.
- [5] Knuth, Donald (1997), «Volume 2: Seminumerical Algorithms», *The Art of Computer Programming* (3rd edición), Addison-Wesley, pp. 65–66, ISBN 0-201-89684-2.
- [6] Martin Sierra, Antonio J., "Programador Certificado Java 2. Curso Práctico 3ra. Ed." Alfaomega Grupo Editor.
- [7] Scheaffer, R.L., W. Mendenhall, and L. Ott. 1996. *Elementary Survey Sampling*, fifth edition. Duxbury Press, Boston, 464pp.
- [8] P. Wright, R. Fields, M. Harrison: "Analysing Human-Computer Interaction as Distributed Cognition". *Human Computer Interaction Journal*, 51(1). 2000.
- [9] Weiss, Mark Allen, "Estructura de datos en Java", Editorial Pearson, ISBN: 8478290354

## 9 Listado del Programa

### 9.1 Clase Principal

#### 9.1.1 PT02App.java

```
/*  
 * PT02App.java es la clase principal de la aplicación, es la que manda llamar a la interfaz gráfica y  
 * maneja las ventanas de la aplicación  
 */
```

```
package pt02.vistas;  
import org.jdesktop.application.Application;  
import org.jdesktop.application.SingleFrameApplication;  
public class PT02App extends SingleFrameApplication {  
  
    /**  
     * Al empezar, muestra el formulario principal de la aplicación.  
     */  
    @Override protected void startup() {  
        show(new Inicio(this));  
    }  
    /**  
     * Este método inicializa la ventana especificada inyectando recursos.  
     * Las ventanas mostradas en la aplicación se inicializan desde el GUI  
     * builder, por lo que no es necesario alguna configuración adicional.  
     */  
    @Override protected void configureWindow(java.awt.Window root) {  
    }  
    public static PT02App getApplication() {  
        return Application.getInstance(PT02App.class);  
    }  
  
    /**  
     * Método Main.  
     */  
    public static void main(String[] args) {  
        launch(PT02App.class, args);  
    }  
}
```

## 9.2 Paquete Vistas

### 9.2.1 Inicio.java

```
/*
 * Inicio.java
 * Este es el formulario principal, la pantalla de inicio de la aplicación
 */

package pt02.vistas;
import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Timer;
import javax.swing.Icon;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class Inicio extends FrameView {

    public Inicio(SingleFrameApplication app) {
        super(app);

        initComponents();
        jsMetodos.setVisible(false);
        ResourceMap resourceMap = getResourceMap();
        int messageTimeout = resourceMap.getInteger("StatusBar.messageTimeout");
        messageTimer = new Timer(messageTimeout, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                statusMessageLabel.setText("");
            }
        });
        messageTimer.setRepeats(false);
        int busyAnimationRate = resourceMap.getInteger("StatusBar.busyAnimationRate");
        for (int i = 0; i < busyIcons.length; i++) {
            busyIcons[i] = resourceMap.getIcon("StatusBar.busyIcons[" + i + "]");
        }
        busyIconTimer = new Timer(busyAnimationRate, new ActionListener() {
```

```

public void actionPerformed(ActionEvent e) {
    busylconIndex = (busylconIndex + 1) % busylcons.length;
    statusAnimationLabel.setIcon(busylcons[busylconIndex]);
}
});
idlelcon = resourceMap.getIcon("StatusBar.idlelcon");
statusAnimationLabel.setIcon(idlelcon);
progressBar.setVisible(false);
TaskMonitor taskMonitor = new TaskMonitor(getApplication().getContext());
taskMonitor.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        String propertyName = evt.getPropertyName();
        if ("started".equals(propertyName)) {
            if (!busylconTimer.isRunning()) {
                statusAnimationLabel.setIcon(busylcons[0]);
                busylconIndex = 0;
                busylconTimer.start();
            }
            progressBar.setVisible(true);
            progressBar.setIndeterminate(true);
        } else if ("done".equals(propertyName)) {
            busylconTimer.stop();
            statusAnimationLabel.setIcon(idlelcon);
            progressBar.setVisible(false);
            progressBar.setValue(0);
        } else if ("message".equals(propertyName)) {
            String text = (String)(evt.getNewValue());
            statusMessageLabel.setText((text == null) ? "" : text);
            messageTimer.restart();
        } else if ("progress".equals(propertyName)) {
            int value = (Integer)(evt.getNewValue());
            progressBar.setVisible(true);
            progressBar.setIndeterminate(false);
            progressBar.setValue(value);
        }
    }
});
}
}

```

```

@Action
public void showAboutBox() {
    if (aboutBox == null) {

```

```

    JFrame mainFrame = PT02App.getApplication().getMainFrame();
    aboutBox = new PT02AboutBox(mainFrame);
    aboutBox.setLocationRelativeTo(mainFrame);
}
PT02App.getApplication().show(aboutBox);
}
private void initComponents() {
    mainPanel = new javax.swing.JPanel();
    lblHeader = new javax.swing.JLabel();
    jsMetodos = new javax.swing.JScrollPane();
    lstMetodos = new javax.swing.JList();
    btnGenerar = new javax.swing.JToggleButton();
    jLabel2 = new javax.swing.JLabel();
    btnEstimacion = new javax.swing.JToggleButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    menuBar = new javax.swing.JMenuBar();
    javax.swing.JMenu fileMenu = new javax.swing.JMenu();
    javax.swing.JMenuItem exitMenuItem = new javax.swing.JMenuItem();
    estimacionMenuItem = new javax.swing.JMenuItem();
    javax.swing.JMenu helpMenu = new javax.swing.JMenu();
    javax.swing.JMenuItem aboutMenuItem = new javax.swing.JMenuItem();
    statusPanel = new javax.swing.JPanel();
    statusMessageLabel = new javax.swing.JLabel();
    statusAnimationLabel = new javax.swing.JLabel();
    progressBar = new javax.swing.JProgressBar();
    jLabel3 = new javax.swing.JLabel();

    org.jdesktop.application.ResourceMap resourceMap =
    org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getR
    esourceMap(Inicio.class);
    mainPanel.setBackground(resourceMap.getColor("mainPanel.background"));
    mainPanel.setName("mainPanel");
    mainPanel.setPreferredSize(new java.awt.Dimension(800, 511));
    lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon"));
    lblHeader.setText(resourceMap.getString("lblHeader.text"));
    lblHeader.setName("lblHeader");
    jsMetodos.setBackground(resourceMap.getColor("jsMetodos.background"));
    jsMetodos.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
    resourceMap.getString("jsMetodos.border.title"), javax.swing.border.TitledBorder.CENTER,
    javax.swing.border.TitledBorder.DEFAULT_POSITION,

```

```

resourceMap.getFont("jsMetodos.border.titleFont"),
resourceMap.getColor("jsMetodos.border.titleColor"));
jsMetodos.setName("jsMetodos");
lstMetodos.setBackground(resourceMap.getColor("lstMetodos.background"));
lstMetodos.setFont(resourceMap.getFont("lstMetodos.font"));
lstMetodos.setForeground(resourceMap.getColor("lstMetodos.foreground"));
lstMetodos.setModel(new javax.swing.AbstractListModel() {
String[] strings = { "Aleatorio Simple No Ponderado", "Aleatorio Simple Ponderado",
"Sistemático", "Estratificado", "Politápico" };
public int getSize() { return strings.length; }
public Object getElementAt(int i) { return strings[i]; }
});
lstMetodos.setSelectionMode(javax.swing.ListSelectionModel.SINGLE_SELECTION);
lstMetodos.setName("lstMetodos");
lstMetodos.setSelectionBackground(resourceMap.getColor("lstMetodos.selectionBackground
"));
lstMetodos.setSelectionForeground(resourceMap.getColor("lstMetodos.selectionForeground
"));
lstMetodos.addMouseListener(new java.awt.event.MouseAdapter() {
public void mouseClicked(java.awt.event.MouseEvent evt) {
pasaMetodo(evt);
}
});
jsMetodos.setViewportView(lstMetodos);
btnGenerar.setFont(resourceMap.getFont("btnGenerar.font"));
btnGenerar.setForeground(resourceMap.getColor("btnGenerar.foreground"));
btnGenerar.setIcon(resourceMap.getIcon("btnGenerar.icon"));
btnGenerar.setText(resourceMap.getString("btnGenerar.text"));
btnGenerar.setToolTipText(resourceMap.getString("btnGenerar.toolTipText"));
btnGenerar.setBorderPainted(false);
btnGenerar.setFocusPainted(false);
btnGenerar.setName("btnGenerar");
btnGenerar.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
escogeMuestra(evt);
}
});
jLabel2.setFont(resourceMap.getFont("jLabel2.font"));
jLabel2.setForeground(resourceMap.getColor("jLabel2.foreground"));
jLabel2.setText(resourceMap.getString("jLabel2.text"));
jLabel2.setName("jLabel2");
btnEstimacion.setFont(resourceMap.getFont("btnEstimacion.font"));

```

```

btnEstimacion.setForeground(resourceMap.getColor("btnEstimacion.foreground"));
btnEstimacion.setIcon(resourceMap.getIcon("btnEstimacion.icon"));
btnEstimacion.setText(resourceMap.getString("btnEstimacion.text"));
btnEstimacion.setToolTipText(resourceMap.getString("btnEstimacion.toolTipText"));
btnEstimacion.setBorderPainted(false);
btnEstimacion.setFocusPainted(false);
btnEstimacion.setName("btnEstimacion");
btnEstimacion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEstimacionActionPerformed(evt);
    }
});
jLabel1.setFont(resourceMap.getFont("jLabel1.font"));
jLabel1.setForeground(resourceMap.getColor("jLabel1.foreground"));
jLabel1.setText(resourceMap.getString("jLabel1.text"));
jLabel1.setName("jLabel1");
jLabel4.setFont(resourceMap.getFont("jLabel4.font"));
jLabel4.setText(resourceMap.getString("jLabel4.text"));
jLabel4.setName("jLabel4");
javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblHeader, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addGap(562, 562, 562)
            .addComponent(jLabel4)
            .addGap(562, 562, 562)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addGap(87, 87, 87)
                .addComponent(btnGenerar, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(87, 87, 87)
                .addComponent(jLabel2)
                .addGap(87, 87, 87)
                .addComponent(btnEstimacion, javax.swing.GroupLayout.PREFERRED_SIZE, 55,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(28, 28, 28)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(33, 33, 33)
            )
        )
);

```

```

        .addGroup(mainPanelLayout.createSequentialGroup())
        .addContainerGap()
            .addComponent(jsMetodos, javax.swing.GroupLayout.DEFAULT_SIZE, 294,
Short.MAX_VALUE)
        .addGap(496, 496, 496))
);
mainPanelLayout.setVerticalGroup(
mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(mainPanelLayout.createSequentialGroup())
    .addComponent(lblHeader, javax.swing.GroupLayout.PREFERRED_SIZE, 221,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jLabel4)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(mainPanelLayout.createSequentialGroup())
        .addGap(26, 26, 26)
        .addComponent(jLabel2))
        .addGroup(mainPanelLayout.createSequentialGroup())
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addComponent(btnEstimacion)
            .addComponent(btnGenerar)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
        .addGap(18, 18, 18)
        .addComponent(jsMetodos, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(300, 300, 300))
);
menuBar.setName("menuBar");
fileMenu.setText(resourceMap.getString("fileMenu.text"));
fileMenu.setName("fileMenu");

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActio
nMap(Inicio.class, this);
exitMenuItem.setAction(actionMap.get("quit"));

```



```

        exitMenuItem.setBackground(resourceMap.getColor("exitMenuItem.background"));
exitMenuItem.setText(resourceMap.getString("exitMenuItem.text"));
        exitMenuItem.setToolTipText(resourceMap.getString("exitMenuItem.toolTipText"));
exitMenuItem.setName("exitMenuItem");
        fileMenu.add(exitMenuItem);

estimacionMenuItem.setBackground(resourceMap.getColor("estimacionMenuItem.background"))
;
        estimacionMenuItem.setText(resourceMap.getString("estimacionMenuItem.text"));
estimacionMenuItem.setName("estimacionMenuItem");
        estimacionMenuItem.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                estimacionMenuItemActionPerformed(evt);
            }
        });
fileMenu.add(estimacionMenuItem);
menuBar.add(fileMenu);
helpMenu.setText(resourceMap.getString("helpMenu.text"));
helpMenu.setName("helpMenu");
aboutMenuItem.setAction(actionMap.get("showAboutBox"));
aboutMenuItem.setText(resourceMap.getString("aboutMenuItem.text"));
aboutMenuItem.setName("aboutMenuItem");
helpMenu.add(aboutMenuItem);
menuBar.add(helpMenu);
statusPanel.setBackground(resourceMap.getColor("statusPanel.background"));
statusPanel.setName("statusPanel");
statusMessageLabel.setName("statusMessageLabel");
statusAnimationLabel.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
statusAnimationLabel.setName("statusAnimationLabel");
progressBar.setName("progressBar");
jLabel3.setIcon(resourceMap.getIcon("jLabel3.icon"));
jLabel3.setText(resourceMap.getString("jLabel3.text"));
jLabel3.setName("jLabel3");
javax.swing.GroupLayout statusPanelLayout = new javax.swing.GroupLayout(statusPanel);
statusPanel.setLayout(statusPanelLayout);
statusPanelLayout.setHorizontalGroup(
    statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(statusPanelLayout.createSequentialGroup()
            .addGap(32, 32, 32)
            .addComponent(jLabel3)

```

```

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(statusPanelLayout.createSequentialGroup()
        .addGap(162, 162, 162)
        .addComponent(statusMessageLabel)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 146,
Short.MAX_VALUE)
        .addComponent(statusAnimationLabel))
    .addGroup(statusPanelLayout.createSequentialGroup()
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addContainerGap()
);
statusPanelLayout.setVerticalGroup(
    statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
statusPanelLayout.createSequentialGroup()
        .addContainerGap(34, Short.MAX_VALUE)

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel3)

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(statusMessageLabel)
    .addComponent(statusAnimationLabel)))
    .addContainerGap()
    .addGroup(statusPanelLayout.createSequentialGroup()
        .addGap(52, 52, 52)
        .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(29, Short.MAX_VALUE))
);

setComponent(mainPanel);
setMenuBar(menuBar);
setStatusBar(statusPanel);
}

private void btnEstimacionActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null, "Próximamente codificado!!!");
}

```

```

private void escogeMuestra(java.awt.event.ActionEvent evt) {
    btnEstimacion.setVisible(false);
    jLabel1.setVisible(false);
    jsMetodos.setVisible(true);
}

private void pasaMetodo(java.awt.event.MouseEvent evt) {
    String[] args={};
    switch(IstMetodos.getSelectedIndex())
    {
        case 0:
            ASimpleNP.main(args);
            break;
        case 1:
            ASimplePonderado.main(args);
            break;
        case 2:
            Sistematico.main(args);
            break;
        case 3:
            Estratificado.main(args);
            break;
        case 4:
            Politapico.main(args);
            break;
        default:
            JOptionPane.showMessageDialog(null,IstMetodos.getSelectedValue().toString()+"
Próximamente Codificado!!!...");
            break;
    }
}

private void estimacionMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null, "Próximamente Codificado");
}

// Declaración de Variables
private javax.swing.JToggleButton btnEstimacion;
private javax.swing.JToggleButton btnGenerar;
private javax.swing.JMenuItem estimacionMenuItem;
private javax.swing.JLabel jLabel1;

```

```

private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JScrollPane jsMetodos;
private javax.swing.JLabel lblHeader;
private javax.swing.JList lstMetodos;
private javax.swing.JPanel mainPanel;
private javax.swing.JMenuBar menuBar;
private javax.swing.JProgressBar progressBar;
private javax.swing.JLabel statusAnimationLabel;
private javax.swing.JLabel statusMessageLabel;
private javax.swing.JPanel statusPanel;
private final Timer messageTimer;
private final Timer busyIconTimer;
private final Icon idleIcon;
private final Icon[] busyIcons = new Icon[15];
private int busyIconIndex = 0;
private JDialog aboutBox;
}

```

### 9.2.2 ASimplePonderado.java

```

/*
 * ASimplePonderado.java
 *
 * Created on 06-dic-2010, 16:28:13
 */
package pt02.vistas;
import javax.swing.JOptionPane;
import javax.swing.table.TableModel;
import org.jdesktop.application.Action;
import pt02.archivos.Archivos;
import pt02.kernel.Muestreo;
public class ASimplePonderado extends javax.swing.JFrame
{
    public String nombreArchivo;
    /** Crea un nuevo formulario ASimplePonderado */
    public ASimplePonderado()
    {
        initComponents();
        jsDatos.setVisible(false);
        pnlTipoArchivo.setVisible(false);
    }
}

```

```

    pnlSelecVar.setVisible(false);
    pnlSelReemplazo.setVisible(false);
}
@SuppressWarnings("unchecked")
private void initComponents() {
    mainPanel = new javax.swing.JPanel();
    header = new javax.swing.JLabel();
    jsDatos = new javax.swing.JScrollPane();
    tblDatos = new javax.swing.JTable();
    btnVolver = new javax.swing.JButton();
    pnlTipoArchivo = new javax.swing.JPanel();
    rbtnTablaFrecuencias = new javax.swing.JRadioButton();
    lblSelTipoArchivo = new javax.swing.JLabel();
    rbtnListaProbabilidades = new javax.swing.JRadioButton();
    btnSiguiete = new javax.swing.JButton();
    btnCargar = new javax.swing.JButton();
    lblCargarArchivo = new javax.swing.JLabel();
    pnlSelecVar = new javax.swing.JPanel();
    comboVariables = new javax.swing.JComboBox();
    lblSeleccionaVariable = new javax.swing.JLabel();
    txtTamMuestra = new javax.swing.JTextField();
    lblTamMuestra = new javax.swing.JLabel();
    lblSelVarArchivo = new javax.swing.JLabel();
    comboPonderacion = new javax.swing.JComboBox();
    lblFooter = new javax.swing.JLabel();
    pnlSelReemplazo = new javax.swing.JPanel();
    rbtnConReemplazo = new javax.swing.JRadioButton();
    rbtnSinReemplazo = new javax.swing.JRadioButton();
    comboReemplazo = new javax.swing.JComboBox();
    btnAyuda = new javax.swing.JButton();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenuItem2 = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(ASimplePonderado.class);
    setTitle(resourceMap.getString("FrmSimplePonderado.title"));
    setName("FrmSimplePonderado");
}

```

```
mainPanel.setBackground(resourceMap.getColor("mainPanel.background"));
mainPanel.setName("mainPanel");
```

```
header.setIcon(resourceMap.getIcon("header.icon"));
header.setText(resourceMap.getString("header.text"));
header.setName("header");
```

```
jsDatos.setBackground(resourceMap.getColor("jsDatos.background"));
```

```
jsDatos.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.create
LineBorder(resourceMap.getColor("jsDatos.border.border.lineColor")),
resourceMap.getString("jsDatos.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsDatos.border.titleFont"),
resourceMap.getColor("jsDatos.border.titleColor")));
jsDatos.setName("jsDatos");
```

```
tblDatos.setFont(resourceMap.getFont("tblDatos.font"));
tblDatos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
tblDatos.setName("tblDatos");
tblDatos.setSelectionBackground(resourceMap.getColor("tblDatos.selectionBackground"));
jsDatos.setViewportViewView(tblDatos);
```

```
btnVolver.setFont(resourceMap.getFont("btnVolver.font"));
btnVolver.setText(resourceMap.getString("btnVolver.text"));
btnVolver.setName("btnVolver");
btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        volver(evt);
    }
});
```

```

    pnlTipoArchivo.setBackground(resourceMap.getColor("pnlTipoArchivo.background"));
    pnlTipoArchivo.setBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("
    pnlTipoArchivo.border.highlightColor"),
    resourceMap.getColor("pnlTipoArchivo.border.shadowColor")));
        pnlTipoArchivo.setName("pnlTipoArchivo");

    rbtnTablaFrecuencias.setBackground(resourceMap.getColor("rbtnTablaFrecuencias.background"))
;
    rbtnTablaFrecuencias.setFont(resourceMap.getFont("rbtnTablaFrecuencias.font"));
    rbtnTablaFrecuencias.setText(resourceMap.getString("rbtnTablaFrecuencias.text"));
    rbtnTablaFrecuencias.setName("rbtnTablaFrecuencias");
    rbtnTablaFrecuencias.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnTablaFrecuenciasMouseClicked(evt);
    }
    });

    lblSelTipoArchivo.setFont(resourceMap.getFont("lblSelTipoArchivo.font"));
    lblSelTipoArchivo.setForeground(resourceMap.getColor("lblSelTipoArchivo.foreground")); //
    lblSelTipoArchivo.setText(resourceMap.getString("lblSelTipoArchivo.text"));
    lblSelTipoArchivo.setName("lblSelTipoArchivo");

    rbtnListaProbabilidades.setBackground(resourceMap.getColor("rbtnListaProbabilidades.backgroun
nd"));
    rbtnListaProbabilidades.setFont(resourceMap.getFont("rbtnListaProbabilidades.font"));
    rbtnListaProbabilidades.setForeground(resourceMap.getColor("rbtnListaProbabilidades.foregroun
d"));
    rbtnListaProbabilidades.setSelected(true);
    rbtnListaProbabilidades.setText(resourceMap.getString("rbtnListaProbabilidades.text"));
    rbtnListaProbabilidades.setName("rbtnListaProbabilidades");
    rbtnListaProbabilidades.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnListaProbabilidadesMouseClicked(evt);
    }
    });

    javax.swing.GroupLayout pnlTipoArchivoLayout = new
    javax.swing.GroupLayout(pnlTipoArchivo);
    pnlTipoArchivo.setLayout(pnlTipoArchivoLayout);
    pnlTipoArchivoLayout.setHorizontalGroup(

```

```

        pnlTipoArchivoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlTipoArchivoLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(lblSelTipoArchivo)
            .addContainerGap(18, Short.MAX_VALUE)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlTipoArchivoLayout.createSequentialGroup()
                .addContainerGap(59, Short.MAX_VALUE)

        .addGroup(pnlTipoArchivoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        G)
            .addComponent(rbtnListaProbabilidades)
            .addComponent(rbtnTablaFrecuencias))
            .addGap(40, 40, 40)
    );
    pnlTipoArchivoLayout.setVerticalGroup(
        pnlTipoArchivoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlTipoArchivoLayout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(lblSelTipoArchivo)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(rbtnListaProbabilidades)
            .addGap(2, 2, 2)
            .addComponent(rbtnTablaFrecuencias)
            .addGap(77, 77, 77)
    );

    btnSiguiete.setFont(resourceMap.getFont("btnSiguiete.font"));
    btnSiguiete.setText(resourceMap.getString("btnSiguiete.text"));
    btnSiguiete.setEnabled(false);
    btnSiguiete.setName("btnSiguiete");
    btnSiguiete.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnSiguieteActionPerformed(evt);
        }
    });

    btnCargar.setFont(resourceMap.getFont("btnCargar.font"));
    btnCargar.setText(resourceMap.getString("btnCargar.text"));
    btnCargar.setName("btnCargar");
    btnCargar.addActionListener(new java.awt.event.ActionListener() {

```



```

    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cargar_archivo(evt);
    }
});

lblCargarArchivo.setFont(resourceMap.getFont("lblCargarArchivo.font"));
lblCargarArchivo.setText(resourceMap.getString("lblCargarArchivo.text"));
lblCargarArchivo.setName("lblCargarArchivo");

pnlSelecVar.setBackground(resourceMap.getColor("pnlSelecVar.background"));
pnlSelecVar.setName("pnlSelecVar");

comboVariables.setBackground(resourceMap.getColor("comboVariables.background"));
comboVariables.setFont(resourceMap.getFont("comboVariables.font"));
comboVariables.setForeground(resourceMap.getColor("comboVariables.foreground"));
comboVariables.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1",
"Item 2", "Item 3", "Item 4" }));
comboVariables.setName("comboVariables");

lblSeleccionaVariable.setFont(resourceMap.getFont("lblSeleccionaVariable.font"));
lblSeleccionaVariable.setForeground(resourceMap.getColor("lblSeleccionaVariable.foreground"));
lblSeleccionaVariable.setText(resourceMap.getString("lblSeleccionaVariable.text"));
lblSeleccionaVariable.setName("lblSeleccionaVariable");

txtTamMuestra.setFont(resourceMap.getFont("txtTamMuestra.font"));
txtTamMuestra.setText(resourceMap.getString("txtTamMuestra.text"));
txtTamMuestra.setName("txtTamMuestra");

lblTamMuestra.setFont(resourceMap.getFont("lblTamMuestra.font"));
lblTamMuestra.setForeground(resourceMap.getColor("lblTamMuestra.foreground"));
lblTamMuestra.setText(resourceMap.getString("lblTamMuestra.text"));
lblTamMuestra.setName("lblTamMuestra");

lblSelVarArchivo.setFont(resourceMap.getFont("lblSelVarArchivo.font"));
lblSelVarArchivo.setForeground(resourceMap.getColor("lblSelVarArchivo.foreground"));
lblSelVarArchivo.setText(resourceMap.getString("lblSelVarArchivo.text"));
lblSelVarArchivo.setName("lblSelVarArchivo");

comboPonderacion.setBackground(resourceMap.getColor("comboPonderacion.background"));
comboPonderacion.setFont(resourceMap.getFont("comboPonderacion.font"));

```



```

        .addGap(18, 18, 18)
        .addComponent(lblSelVarArchivo)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(comboPonderacion, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(18, Short.MAX_VALUE)
    );

    lblFooter.setIcon(resourceMap.getIcon("lblFooter.icon"));
    lblFooter.setText(resourceMap.getString("lblFooter.text"));
    lblFooter.setName("lblFooter");

    pnlSelReemplazo.setName("pnlSelReemplazo");

    rbtnConReemplazo.setFont(resourceMap.getFont("rbtnConReemplazo.font"));
    rbtnConReemplazo.setSelected(true);
    rbtnConReemplazo.setText(resourceMap.getString("rbtnConReemplazo.text"));
    rbtnConReemplazo.setName("rbtnConReemplazo");
    rbtnConReemplazo.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            rbtnConReemplazoMouseClicked(evt);
        }
    });

    rbtnSinReemplazo.setFont(resourceMap.getFont("rbtnSinReemplazo.font"));
    rbtnSinReemplazo.setForeground(resourceMap.getColor("rbtnSinReemplazo.foreground"));
    rbtnSinReemplazo.setText(resourceMap.getString("rbtnSinReemplazo.text"));
    rbtnSinReemplazo.setName("rbtnSinReemplazo");
    rbtnSinReemplazo.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            rbtnSinReemplazoMouseClicked(evt);
        }
    });

    comboReemplazo.setFont(resourceMap.getFont("comboReemplazo.font"));
    comboReemplazo.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "XCL",
"Algoritmo 1", "Algoritmo 2", "Algoritmo 3", "Algoritmo 4" }));
    comboReemplazo.setName("comboReemplazo");

    javax.swing.GroupLayout pnlSelReemplazoLayout = new
javax.swing.GroupLayout(pnlSelReemplazo);
    pnlSelReemplazo.setLayout(pnlSelReemplazoLayout);

```

```

pnlSelReemplazoLayout.setHorizontalGroup(
    pnlSelReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSelReemplazoLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(rbbtnConReemplazo)
            .addContainerGap(90, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlSelReemplazoLayout.createSequentialGroup()
            .addContainerGap(62, Short.MAX_VALUE)
            .addComponent(comboReemplazo, javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(42, 42, 42))
        .addGroup(pnlSelReemplazoLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(rbbtnSinReemplazo)
            .addContainerGap(94, Short.MAX_VALUE))
);
pnlSelReemplazoLayout.setVerticalGroup(
    pnlSelReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSelReemplazoLayout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(rbbtnConReemplazo)
            .addGap(5, 5, 5)
            .addComponent(rbbtnSinReemplazo)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(comboReemplazo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

btnAyuda.setIcon(resourceMap.getIcon("btnAyuda.icon"));
btnAyuda.setText(resourceMap.getString("btnAyuda.text"));
btnAyuda.setName("btnAyuda");

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(header)

```

```

.addGroup(mainPanelLayout.createSequentialGroup())
    .addContainerGap()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addGroup(mainPanelLayout.createSequentialGroup())

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(lblCargarArchivo, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jsDatos, javax.swing.GroupLayout.DEFAULT_SIZE, 314,
Short.MAX_VALUE))
    .addComponent(btnCargar))

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup())
        .addGap(73, 73, 73)
        .addComponent(pnlSelecVar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup())
        .addComponent(pnlSelReemplazo,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(39, 39, 39))
    .addGroup(mainPanelLayout.createSequentialGroup())
        .addGap(65, 65, 65)
        .addComponent(pnlTipoArchivo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
47, Short.MAX_VALUE)
        .addComponent(btnAyuda, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGap(117, 117, 117))
.addGroup(mainPanelLayout.createSequentialGroup())
    .addComponent(btnVolver)
    .addGap(99, 99, 99)
    .addComponent(lblFooter)
    .addGap(99, 99, 99)

```

```

        .addComponent(btnSiguiente))))))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
mainPanelLayout.setVerticalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addComponent(header)
        .addGap(18, 18, 18)

        .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 313,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(lblCargarArchivo)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnCargar))
            .addGroup(mainPanelLayout.createSequentialGroup()

            .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(mainPanelLayout.createSequentialGroup()
                    .addGap(9, 9, 9)
                    .addComponent(pnlTipoArchivo, javax.swing.GroupLayout.PREFERRED_SIZE,
103, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(mainPanelLayout.createSequentialGroup()
                    .addGap(17, 17, 17)
                    .addComponent(btnAyuda)))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(pnlSelReemplazo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(11, 11, 11)
                .addComponent(pnlSelecVar, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 15,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(btnVolver)
            .addComponent(btnSiguiente)
            .addComponent(lblFooter, javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap())

```

```

);

jMenuBar1.setName("jMenuBar1");
jMenu1.setText(resourceMap.getString("jMenu1.text"));
jMenu1.setName("jMenu1");

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActionMap(ASimplePonderado.class, this);
jMenuItem1.setAction(actionMap.get("quit"));
jMenuItem1.setText(resourceMap.getString("jMenuItem1.text"));
jMenuItem1.setName("jMenuItem1");
jMenu1.add(jMenuItem1);
jMenuBar1.add(jMenu1);
jMenu2.setText(resourceMap.getString("jMenu2.text"));
jMenu2.setName("jMenu2");
jMenuBar1.add(jMenu2);
setJMenuBar(jMenuBar1);
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE, 801,
javax.swing.GroupLayout.PREFERRED_SIZE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
pack();
}

private void volver(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void cargar_archivo(java.awt.event.ActionEvent evt) {
    Archivos archivo=new Archivos();
    nombreArchivo=archivo.manejar_archivo(tblDatos, comboVariables,comboPonderacion);
    jsDatos.setVisible(true);
    pnlTipoArchivo.setVisible(true);
}

```

```

        pnlSelecVar.setVisible(true);
        lblCargarArchivo.setVisible(false);
        btnSiguiete.setEnabled(true);
    }

    private void rbtnTablaFrecuenciasMouseClicked(java.awt.event.MouseEvent evt) {
        if(rbtnTablaFrecuencias.isSelected())
            rbtnListaProbabilidades.setSelected(false);
            rbtnTablaFrecuencias.setSelected(true);
            pnlSelReemplazo.setVisible(true);
    }

    private void rbtnListaProbabilidadesMouseClicked(java.awt.event.MouseEvent evt) {
        if(rbtnListaProbabilidades.isSelected())
            rbtnTablaFrecuencias.setSelected(false);
            rbtnListaProbabilidades.setSelected(true);
            pnlSelReemplazo.setVisible(false);
    }

    private void btnSiguieteActionPerformed(java.awt.event.ActionEvent evt) {
        // 1) ver si la variable y frecuencia seleccionados no son las mismas
        String uno=comboVariables.getSelectedItem().toString();
        String dos=comboPonderacion.getSelectedItem().toString();
        String tipo=null;
        int tamMuestra=Integer.parseInt(txtTamMuestra.getText().trim());
        if(uno.equals(dos))
            JOptionPane.showMessageDialog(null,"Sus variables deben ser diferentes.....");
        else
        {
            // 2) ver si se trata de lista de frecuencias o de probabilidades
            if(rbtnListaProbabilidades.isSelected())
                tipo="Lista de Probabilidades";
            else
                tipo="Tabla de Frecuencias";
            // 3) Verificar que el tamaño de la muestra no sea 0
            if(tamMuestra==0)
                JOptionPane.showMessageDialog(null, "Por favor selecciona el tamaño de la
muestra.....");
            else
                if(tipo.contains("Tabla")) //tabla de frecuencias
                {

```



```

        int tamP=cuenta_pob(tblDatos.getModel(),comboPonderacion.getSelectedIndex());
        if(tamMuestra>tamP)
            JOptionPane.showMessageDialog(null, "El tamaño de la muestra no puede ser mayor al
de la población! (" +tamP+"");

        else
        {
            // 4) todo listo
            JOptionPane.showMessageDialog(null, "Seleccionó la Variable: " +
comboVariables.getSelectedItem().toString() + "\n Tamaño de la Muestra: " + tamMuestra
+ "\n Tipo de Archivo: "+tipo+"");
            // 5) Llamo a la clase muestreo para empezar el método
            if(tipo.contains("Lista")) //lista de probabilidades
            {
                Muestreo metodo=new
Muestreo(comboVariables,tblDatos.getModel(),txtTamMuestra.getText().trim(),comboReemplazo.
getSelectedItem().toString(),comboPonderacion);
                metodo.empezar_metodo(3, nombreArchivo, tipo);
            }
            else //tabla de frecuencias
            {
                String num_algo=null;
                if(rbbtnConReemplazo.isSelected())
                {
                    num_algo="0";
                }
                else
                if(rbbtnSinReemplazo.isSelected())
                {
                    Object na=comboReemplazo.getSelectedIndex();
                    num_algo=na.toString();
                }
                Muestreo metodo=new
Muestreo(comboVariables,tblDatos.getModel(),txtTamMuestra.getText().trim(),tipo,num_algo,co
mboPonderacion);
                metodo.empezar_metodo(3, nombreArchivo, tipo);
            }
        }
    }
}
else
{

```

```

//lista de probas
// 4) todo listo
JOptionPane.showMessageDialog(null, "Seleccionó la Variable: " +
comboVariables.getSelectedItem().toString() + "\n Tamaño de la Muestra: " + tamMuestra
+ "\n Tipo de Archivo: "+tipo+""");
// 5) Llamar a la clase muestreo para empezar el método
if(tipo.contains("Lista")) //lista de probabilidades
{
Muestreo metodo=new
Muestreo(comboVariables,tblDatos.getModel(),txtTamMuestra.getText().trim(),tipo,comboPonderacion);
metodo.empezar_metodo(3, nombreArchivo,
comboPonderacion.getSelectedItem().toString());
}
else //tabla de frecuencias
{
String num_algo=null;
if(rbbtnConReemplazo.isSelected())
{
num_algo="0";
}
else
if(rbbtnSinReemplazo.isSelected())
{
Object na=comboReemplazo.getSelectedIndex();
num_algo=na.toString();
}
Muestreo metodo=new
Muestreo(comboVariables,tblDatos.getModel(),txtTamMuestra.getText().trim(),tipo,num_algo,comboPonderacion);
metodo.empezar_metodo(3, nombreArchivo, tipo);
}
}
}
}

private void rbbtnConReemplazoMouseClicked(java.awt.event.MouseEvent evt) {
if(rbbtnConReemplazo.isSelected())
rbtnSinReemplazo.setSelected(false);
rbtnConReemplazo.setSelected(true);
}
}
}

```

```

}

private void rbtnSinReemplazoMouseClicked(java.awt.event.MouseEvent evt) {
    if(rbbtnSinReemplazo.isSelected())
        rbbtnConReemplazo.setSelected(false);
        rbbtnSinReemplazo.setSelected(true);
}

private int cuenta_pob(TableModel tabla, int columna)
{
    int contador=0;
    for(int y=0;y<tabla.getRowCount();y++)
    {
        String c=tabla.getValueAt(y, columna).toString();
        if(c.contains(".0"))
            c=c.replace(".0", "");
        contador=contador+Integer.parseInt(c);
    }
    return contador;
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ASimplePonderado().setVisible(true);
        }
    });
}

// Declaración de Variables
private javax.swing.JButton btnAyuda;
private javax.swing.JButton btnCargar;
private javax.swing.JButton btnSiguiente;
private javax.swing.JButton btnVolver;
private javax.swing.JComboBox comboPonderacion;
private javax.swing.JComboBox comboReemplazo;
private javax.swing.JComboBox comboVariables;
private javax.swing.JLabel header;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;

```

```

private javax.swing.JScrollPane jsDatos;
private javax.swing.JLabel lblCargarArchivo;
private javax.swing.JLabel lblFooter;
private javax.swing.JLabel lblSelTipoArchivo;
private javax.swing.JLabel lblSelVarArchivo;
private javax.swing.JLabel lblSeleccionaVariable;
private javax.swing.JLabel lblTamMuestra;
private javax.swing.JPanel mainPanel;
private javax.swing.JPanel pnlSelReemplazo;
private javax.swing.JPanel pnlSelecVar;
private javax.swing.JPanel pnlTipoArchivo;
private javax.swing.JRadioButton rbtnConReemplazo;
private javax.swing.JRadioButton rbtnListaProbabilidades;
private javax.swing.JRadioButton rbtnSinReemplazo;
private javax.swing.JRadioButton rbtnTablaFrecuencias;
private javax.swing.JTable tblDatos;
private javax.swing.JTextField txtTamMuestra;
}

```

### 9.2.3 ASimpleNP.java

```

/*
 * Esta clase es el formulario del método de muestreo aleatorio simple no ponderado
 * ASimpleNP.java
 * Created on 9/09/2010, 01:11:07 AM
 */
package pt02.vistas;
import javax.swing.JOptionPane;
import pt02.archivos.Archivos;
import pt02.kernel.Muestreo;
/**
 *
 * @author XiMe
 */
public class ASimpleNP extends javax.swing.JFrame
{
    public String nombreArchivo;

    /** Crea un nuevo formulario ASimpleNP */
    public ASimpleNP() {
        initComponents();
        comboAlgoritmos.setVisible(false);
        pnlSeleccion.setVisible(false);
    }
}

```

```

    pnlReemplazo.setVisible(false);
}

@SuppressWarnings("unchecked")
private void initComponents() {
    pnlPrincipal = new javax.swing.JPanel();
    jsDatos = new javax.swing.JScrollPane();
    tblDatos = new javax.swing.JTable();
    pnlSeleccion = new javax.swing.JPanel();
    lblSelecciona = new javax.swing.JLabel();
    txtTamMuestra = new javax.swing.JTextField();
    comboVariables = new javax.swing.JComboBox();
    lblTamaño = new javax.swing.JLabel();
    btnCalcular = new javax.swing.JButton();
    btnVolver = new javax.swing.JButton();
    btnSiguiente = new javax.swing.JButton();
    lblPulse = new javax.swing.JLabel();
    pnlReemplazo = new javax.swing.JPanel();
    rbtnConReemplazo = new javax.swing.JRadioButton();
    rbtnSinReemplazo = new javax.swing.JRadioButton();
    comboAlgoritmos = new javax.swing.JComboBox();
    btnAyuda = new javax.swing.JButton();
    btnCargar = new javax.swing.JButton();
    lblMariposas = new javax.swing.JLabel();
    lblMariposas1 = new javax.swing.JLabel();
    lblMariposas2 = new javax.swing.JLabel();
    lblSombrero = new javax.swing.JLabel();
    lblHeader = new javax.swing.JLabel();
    mnuArriba = new javax.swing.JMenuBar();
    smnuArchivo = new javax.swing.JMenu();
    itemSalir = new javax.swing.JMenuItem();
    smnuAyuda = new javax.swing.JMenu();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    org.jdesktop.application.ResourceMap resourceMap =
    org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getR
    esourceMap(ASimpleNP.class);
    setTitle(resourceMap.getString("Aleatorio Simple No Ponderado.title"));
    setAlwaysOnTop(true);
    setBackground(resourceMap.getColor("Aleatorio Simple No Ponderado.background"));
    setBounds(new java.awt.Rectangle(0, 0, 0, 0));
    setName("Aleatorio Simple No Ponderado");
}

```

```

setResizable(false);

pnlPrincipal.setBackground(resourceMap.getColor("pnlPrincipal.background"));
pnlPrincipal.setName("pnlPrincipal");

jsDatos.setBackground(resourceMap.getColor("jsDatos.background"));
jsDatos.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
resourceMap.getString("jsDatos.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsDatos.border.titleFont"),
resourceMap.getColor("jsDatos.border.titleColor")));
jsDatos.setName("jsDatos");

tblDatos.setFont(resourceMap.getFont("tblDatos.font"));
tblDatos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null},
        {null, null, null},
        {null, null, null},
        {null, null, null}
    },
    new String [] {
        "-", "-", "-"
    }
));
tblDatos.setGridColor(resourceMap.getColor("tblDatos.gridColor"));
tblDatos.setName("tblDatos");
tblDatos.setSelectionBackground(resourceMap.getColor("tblDatos.selectionBackground"));
jsDatos.setViewportView(tblDatos);

pnlSeleccion.setBackground(resourceMap.getColor("pnlSeleccion.background"));
pnlSeleccion.setBorder(new javax.swing.border.MatteBorder(new
javax.swing.ImageIcon(getClass().getResource("/pt02/vistas/resources/marcoMorado.jpg"))));
;
pnlSeleccion.setName("pnlSeleccion");

lblSelecciona.setFont(resourceMap.getFont("lblSelecciona.font")); // NOI18N
lblSelecciona.setForeground(resourceMap.getColor("lblSelecciona.foreground")); // NOI18N
lblSelecciona.setText(resourceMap.getString("lblSelecciona.text")); // NOI18N
lblSelecciona.setName("lblSelecciona"); // NOI18N

```



```

        .addContainerGap()
        .addComponent(lblSelecciona))
    .addGroup(pnlSeleccionLayout.createSequentialGroup())
        .addGap(64, 64, 64)
        .addComponent(comboVariables, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(pnlSeleccionLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(lblTamaño)))
    .addContainerGap(38, Short.MAX_VALUE)
);
pnlSeleccionLayout.setVerticalGroup(
    pnlSeleccionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlSeleccionLayout.createSequentialGroup())
        .addComponent(lblSelecciona)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(comboVariables, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(lblTamaño)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(pnlSeleccionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
        .addComponent(btnCalcular)
        .addComponent(txtTamMuestra, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap()
);

btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
btnVolver.setToolTipText(resourceMap.getString("btnVolver.toolTipText")); // NOI18N
btnVolver.setName("btnVolver"); // NOI18N
btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnVolverActionPerformed(evt);
    }
});

```



```

btnSiguiente.setFont(resourceMap.getFont("btnSiguiente.font"));
btnSiguiente.setForeground(resourceMap.getColor("btnSiguiente.foreground"));
btnSiguiente.setText(resourceMap.getString("btnSiguiente.text"));
btnSiguiente.setToolTipText(resourceMap.getString("btnSiguiente.toolTipText"));
btnSiguiente.setEnabled(false);
btnSiguiente.setName("btnSiguiente");
btnSiguiente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSiguienteActionPerformed(evt);
    }
});

```

```

lblPulse.setFont(resourceMap.getFont("lblPulse.font"));
lblPulse.setForeground(resourceMap.getColor("lblPulse.foreground"));
lblPulse.setText(resourceMap.getString("lblPulse.text"));
lblPulse.setName("lblPulse");

```

```

pnlReemplazo.setBackground(resourceMap.getColor("pnlReemplazo.background"));
pnlReemplazo.setBorder(new javax.swing.border.MatteBorder(new
javax.swing.Imgelcon(getClass().getResource("/pt02/vistas/resources/marcoMorado.jpg"))));
pnlReemplazo.setName("pnlReemplazo"); // NOI18N

```

```

rbtnConReemplazo.setBackground(resourceMap.getColor("rbtnConReemplazo.background"));
rbtnConReemplazo.setFont(resourceMap.getFont("rbtnConReemplazo.font")); // NOI18N
rbtnConReemplazo.setForeground(resourceMap.getColor("rbtnConReemplazo.foreground"));
rbtnConReemplazo.setSelected(true);
rbtnConReemplazo.setText(resourceMap.getString("rbtnConReemplazo.text")); // NOI18N
rbtnConReemplazo.setName("rbtnConReemplazo"); // NOI18N
rbtnConReemplazo.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnConReemplazoMouseClicked(evt);
    }
});

```

```

rbtnSinReemplazo.setBackground(resourceMap.getColor("rbtnSinReemplazo.background"));
rbtnSinReemplazo.setFont(resourceMap.getFont("rbtnSinReemplazo.font")); // NOI18N
rbtnSinReemplazo.setForeground(resourceMap.getColor("rbtnSinReemplazo.foreground"));
rbtnSinReemplazo.setText(resourceMap.getString("rbtnSinReemplazo.text"));
rbtnSinReemplazo.setName("rbtnSinReemplazo");
rbtnSinReemplazo.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnSinReemplazoMouseClicked(evt);
    }
});

```

```

    }
    });

    comboAlgoritmos.setBackground(resourceMap.getColor("comboAlgoritmos.background"));
    comboAlgoritmos.setFont(resourceMap.getFont("comboAlgoritmos.font")); // NOI18N
    comboAlgoritmos.setForeground(resourceMap.getColor("comboAlgoritmos.foreground"));
    comboAlgoritmos.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Algoritmo XCL", "Algoritmo 1", "Algoritmo 2", "Algoritmo 3", "Algoritmo 4" }));
    comboAlgoritmos.setName("comboAlgoritmos");

    btnAyuda.setIcon(resourceMap.getIcon("btnAyuda.icon"));
    btnAyuda.setText(resourceMap.getString("btnAyuda.text"));
    btnAyuda.setToolTipText(resourceMap.getString("btnAyuda.toolTipText"));
    btnAyuda.setEnabled(false);
    btnAyuda.setName("btnAyuda");
    btnAyuda.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ayuda(evt);
        }
    });

    javax.swing.GroupLayout pnlReemplazoLayout = new
javax.swing.GroupLayout(pnlReemplazo);
    pnlReemplazo.setLayout(pnlReemplazoLayout);
    pnlReemplazoLayout.setHorizontalGroup(
        pnlReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlReemplazoLayout.createSequentialGroup()
                .addGroup(pnlReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(pnlReemplazoLayout.createSequentialGroup()
                        .addGroup(pnlReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(pnlReemplazoLayout.createSequentialGroup()
                                .addComponent(rbtnConReemplazo)
                                .addComponent(rbtnSinReemplazo))
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 40,
Short.MAX_VALUE)
                                .addComponent(btnAyuda, javax.swing.GroupLayout.PREFERRED_SIZE, 46,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(pnlReemplazoLayout.createSequentialGroup()
                            .addGroup(pnlReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(pnlReemplazoLayout.createSequentialGroup()
                                    .addGroup(pnlReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(pnlReemplazoLayout.createSequentialGroup()
                                            .addGap(21, 21, 21)

```

```

        .addComponent(comboAlgoritmos, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap()
    );
    pnlReemplazoLayout.setVerticalGroup(
        pnlReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlReemplazoLayout.createSequentialGroup()
            .addContainerGap()

.addGroup(pnlReemplazoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    G)
        .addComponent(btnAyuda)
        .addGroup(pnlReemplazoLayout.createSequentialGroup()
            .addComponent(rbbtnConReemplazo)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(rbbtnSinReemplazo))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(comboAlgoritmos, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    btnCargar.setFont(resourceMap.getFont("btnCargar.font")); // NOI18N
    btnCargar.setForeground(resourceMap.getColor("btnCargar.foreground")); // NOI18N
    btnCargar.setText(resourceMap.getString("btnCargar.text")); // NOI18N
    btnCargar.setToolTipText(resourceMap.getString("btnCargar.toolTipText")); // NOI18N
    btnCargar.setName("btnCargar"); // NOI18N
    btnCargar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnCargarActionPerformed(evt);
        }
    });

    lblMariposas.setIcon(resourceMap.getIcon("lblMariposas.icon")); // NOI18N
    lblMariposas.setText(resourceMap.getString("lblMariposas.text")); // NOI18N
    lblMariposas.setName("lblMariposas"); // NOI18N

    lblMariposas1.setIcon(resourceMap.getIcon("lblMariposas1.icon"));
    lblMariposas1.setName("lblMariposas1");

    lblMariposas2.setIcon(resourceMap.getIcon("lblMariposas2.icon"));
    lblMariposas2.setName("lblMariposas2");

```

```

lblSombrero.setIcon(resourceMap.getIcon("lblSombrero.icon"));
lblSombrero.setText(resourceMap.getString("lblSombrero.text"));
lblSombrero.setName("lblSombrero");

lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon"));
lblHeader.setText(resourceMap.getString("lblHeader.text"));
lblHeader.setName("lblHeader");

javax.swing.GroupLayout pnlPrincipalLayout = new javax.swing.GroupLayout(pnlPrincipal);
pnlPrincipal.setLayout(pnlPrincipalLayout);
pnlPrincipalLayout.setHorizontalGroup(
    pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPrincipalLayout.createSequentialGroup()
            .addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlPrincipalLayout.createSequentialGroup()
                    .addGroup(pnlPrincipalLayout.createSequentialGroup()
                        .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 317,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(pnlPrincipalLayout.createSequentialGroup()
                                .addGroup(pnlPrincipalLayout.createSequentialGroup()
                                    .addComponent(btnCargar)
                                    .addComponent(lblPulse)))
                                .addGroup(pnlPrincipalLayout.createSequentialGroup()
                                    .addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
                                        false)
                                        .addComponent(pnlReemplazo,
                                            javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE,
                                            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                        .addComponent(pnlSeleccion, javax.swing.GroupLayout.Alignment.TRAILING,
                                            javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                                            Short.MAX_VALUE)
                                        .addGap(38, 38, 38)
                                        .addComponent(lblSombrero))))))
                            .addComponent(lblHeader, javax.swing.GroupLayout.DEFAULT_SIZE,
                                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(76, 76, 76)
            )
        )
    );

```

```

        .addContainerGap()
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPrincipalLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(btnVolver)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 54,
Short.MAX_VALUE)
        .addComponent(lblMariposas)
        .addGap(18, 18, 18)
        .addComponent(lblMariposas1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(lblMariposas2)
        .addGap(28, 28, 28)
        .addComponent(btnSiguiete)
        .addGap(28, 28, 28)
    );
    pnlPrincipalLayout.setVerticalGroup(
        pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPrincipalLayout.createSequentialGroup()
            .addComponent(lblHeader)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
            .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 458,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(pnlPrincipalLayout.createSequentialGroup()

        .addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlPrincipalLayout.createSequentialGroup()
                .addComponent(lblSombrero, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(37, 37, 37))
            .addGroup(pnlPrincipalLayout.createSequentialGroup()
                .addGap(16, 16, 16)
                .addComponent(pnlSeleccion, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
            .addComponent(pnlReemplazo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(41, 41, 41)
            .addComponent(lblPulse)

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btnCargar, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(18, 18, 18)

.addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(lblMariposas2)
        .addComponent(lblMariposas1)
        .addComponent(lblMariposas)
        .addComponent(btnVolver)
        .addComponent(btnSiguiente))
        .addContainerGap()
);

mnuArriba.setFont(resourceMap.getFont("mnuArriba.font")); // NOI18N
mnuArriba.setName("mnuArriba"); // NOI18N

smnuArchivo.setText(resourceMap.getString("smnuArchivo.text")); // NOI18N
smnuArchivo.setName("smnuArchivo"); // NOI18N

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActionMap(ASimpleNP.class, this);
itemSalir.setAction(actionMap.get("quit")); // NOI18N
itemSalir.setText(resourceMap.getString("itemSalir.text")); // NOI18N
itemSalir.setName("itemSalir"); // NOI18N
smnuArchivo.add(itemSalir);

mnuArriba.add(smnuArchivo);

smnuAyuda.setText(resourceMap.getString("smnuAyuda.text")); // NOI18N
smnuAyuda.setName("smnuAyuda"); // NOI18N
mnuArriba.add(smnuAyuda);

setJMenuBar(mnuArriba);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(pnlPrincipal, javax.swing.GroupLayout.PREFERRED_SIZE, 798,
javax.swing.GroupLayout.PREFERRED_SIZE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlPrincipal, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    );

```

```

    pack();
} // </editor-fold>

```

```

private void btnCargarActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    Archivos inicio=new Archivos();
    nombreArchivo=inicio.manejar_archivo(tblDatos, comboVariables);
    //mostrar los demás componentes
    lblPulse.setVisible(false);
    btnCalcular.setEnabled(true);
    btnSiguiente.setEnabled(true);
    btnAyuda.setEnabled(true);
    pnlSeleccion.setVisible(true);
    pnlReemplazo.setVisible(true);
}

```

```

private void btnVolverActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    this.dispose();
}

```

```

private void btnSiguienteActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    // String confianza=JOptionPane.showInputDialog("Indica el nivel de confiabilidad alfa (%");
    String confianza="95";
    String texto=txtTamMuestra.getText();
    String status=null;
    if(rbtnConReemplazo.isSelected())
        status="Con Reemplazo";
    else status="Sin Reemplazo: "+comboAlgoritmos.getSelectedItem().toString();

```

```

    int tam=Integer.parseInt(texto);
    if(tam!=0)

```

```

{
    if(tam>tblDatos.getRowCount())
        JOptionPane.showMessageDialog(null, "El tamaño de la muestra debe ser menor que el
tamaño de la población!! (" +tblDatos.getRowCount()+")");
    else
    {
        JOptionPane.showMessageDialog(null, "Seleccionó la Variable: " +
comboVariables.getSelectedItem().toString() + "\n Tamaño de la Muestra: " + tam
        + "\n Confiabilidad: "+confianza+"%\n"+status);
        Muestreo muestreo=new
Muestreo(comboVariables,tblDatos.getModel(),texto,status,comboAlgoritmos);
        muestreo.empezar_metodo(0,nombreArchivo,confianza); //0 es para el aleatorio simple no
ponderado
    }
}
else
    JOptionPane.showMessageDialog(null,"Porfavor elige un tamaño para la muestra....");
}

private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null, "Próximamente Codificado!!!!");
}

private void rbtnConReemplazoMouseClicked(java.awt.event.MouseEvent evt) {
    if(rbtnSinReemplazo.isSelected())
        rbtnSinReemplazo.setSelected(false);
    comboAlgoritmos.setVisible(false);
}

private void rbtnSinReemplazoMouseClicked(java.awt.event.MouseEvent evt) {
    if(rbtnConReemplazo.isSelected())
        rbtnConReemplazo.setSelected(false);
    comboAlgoritmos.setVisible(true);
}

private void ayuda(java.awt.event.ActionEvent evt) {
    String hola[]={ "C:\\Users\\USUARIO\\Documents\\h.txt",this.getName()};
    //String hola[]={ "D:\\Documents and Settings\\xcerna\\Mis
Documentos\\vals.txt",this.getName()};
}

```



```

// String hola[]={"C:\\Documents and Settings\\PATRICIA\\Mis
Documentos\\prueba.txt",this.getName()};
Ayuda.main(hola);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ASimpleNP().setVisible(true);
        }
    });
}

```

```

// Declaración de variables
private javax.swing.JButton btnAyuda;
private javax.swing.JButton btnCalcular;
private javax.swing.JButton btnCargar;
private javax.swing.JButton btnSiguiente;
private javax.swing.JButton btnVolver;
private javax.swing.JComboBox comboAlgoritmos;
private javax.swing.JComboBox comboVariables;
private javax.swing.JMenuItem itemSalir;
private javax.swing.JScrollPane jsDatos;
private javax.swing.JLabel lblHeader;
private javax.swing.JLabel lblMariposas;
private javax.swing.JLabel lblMariposas1;
private javax.swing.JLabel lblMariposas2;
private javax.swing.JLabel lblPulse;
private javax.swing.JLabel lblSelecciona;
private javax.swing.JLabel lblSombrero;
private javax.swing.JLabel lblTamaño;
private javax.swing.JMenuBar mnuArriba;
private javax.swing.JPanel pnlPrincipal;
private javax.swing.JPanel pnlReemplazo;
private javax.swing.JPanel pnlSeleccion;
private javax.swing.JRadioButton rbtnConReemplazo;
private javax.swing.JRadioButton rbtnSinReemplazo;
private javax.swing.JMenu smnuArchivo;
private javax.swing.JMenu smnuAyuda;
private javax.swing.JTable tblDatos;
private javax.swing.JTextField txtTamMuestra;
}

```

#### 9.2.4 Sistematico.java

```
/*
 * Sistematico.java
 * Created on 22/09/2010, 10:50:31 PM
 */
package pt02.vistas;
import java.util.LinkedList;
import javax.swing.JOptionPane;
import org.jdesktop.application.Action;
import pt02.archivos.Archivos;
import pt02.estadistica.Operaciones;
import pt02.kernel.Muestreo;
/**
 *
 * @author Xime
 */
public class Sistematico extends javax.swing.JFrame {
    public String nombreArchivo;
    public Sistematico()
    {
        initComponents();
        lblCompleta.setVisible(false);
        tabbedEtapas.setVisible(false);
        btnAyuda.setVisible(false);
    }
    @SuppressWarnings("unchecked")
    private void initComponents()
    {
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextPane1 = new javax.swing.JTextPane();
        mainPanel = new javax.swing.JPanel();
        lblHeader = new javax.swing.JLabel();
        jsDatos = new javax.swing.JScrollPane();
        tblDatos = new javax.swing.JTable();
        lblPulse = new javax.swing.JLabel();
        btnCargar = new javax.swing.JButton();
        btnVolver = new javax.swing.JButton();
        btnSiguiente = new javax.swing.JButton();
        tabbedEtapas = new javax.swing.JTabbedPane();
        panelEtapa1 = new javax.swing.JPanel();
        lblSellInicio = new javax.swing.JLabel();
    }
}
```

```

btnGenerarInicio = new javax.swing.JToggleButton();
txtInicio = new javax.swing.JTextField();
lblEstrella = new javax.swing.JLabel();
panelEtapa2 = new javax.swing.JPanel();
lblPaso = new javax.swing.JLabel();
btnCalcularPaso = new javax.swing.JButton();
btnAleatorio = new javax.swing.JButton();
txtPaso = new javax.swing.JTextField();
lblEstrellaAzul2 = new javax.swing.JLabel();
panelEtapa3 = new javax.swing.JPanel();
lblSelVar = new javax.swing.JLabel();
lblTamM = new javax.swing.JLabel();
txtTamM = new javax.swing.JTextField();
btnCalcular = new javax.swing.JToggleButton();
comboVariables = new javax.swing.JComboBox();
lblEstrellaAzulCielo = new javax.swing.JLabel();
lblCompleta = new javax.swing.JLabel();
lblDividerAzul = new javax.swing.JLabel();
btnAyuda = new javax.swing.JButton();
lblCorazon = new javax.swing.JLabel();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();

jScrollPane1.setName("jScrollPane1"); // NOI18N

jTextPane1.setName("jTextPane1"); // NOI18N
jScrollPane1.setViewportView(jTextPane1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(Sistematico.class);
setTitle(resourceMap.getString("Form.title")); // NOI18N
setBackground(resourceMap.getColor("Form.background")); // NOI18N
setName("Form"); // NOI18N

mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
mainPanel.setBorder(new javax.swing.border.MatteBorder(new
javax.swing.Imgelcon(getClass().getResource("/pt02/vistas/resources/marcoAzul.jpg"))));
mainPanel.setName("mainPanel"); // NOI18N

```

```
lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
lblHeader.setName("lblHeader"); // NOI18N
```

```
jsDatos.setBackground(resourceMap.getColor("jsDatos.background")); // NOI18N
jsDatos.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
resourceMap.getString("jsDatos.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsDatos.border.titleFont"),
resourceMap.getColor("jsDatos.border.titleColor"))); // NOI18N
jsDatos.setName("jsDatos"); // NOI18N
```

```
tblDatos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null},
        {null, null, null},
        {null, null, null}
    },
    new String [] {
        "-", "-", "-"
    }
));
tblDatos.setName("tblDatos"); // NOI18N
jsDatos.setViewportView(tblDatos);
```

```
lblPulse.setFont(resourceMap.getFont("lblPulse.font")); // NOI18N
lblPulse.setForeground(resourceMap.getColor("lblPulse.foreground")); // NOI18N
lblPulse.setText(resourceMap.getString("lblPulse.text")); // NOI18N
lblPulse.setName("lblPulse"); // NOI18N
```

```
btnCargar.setFont(resourceMap.getFont("btnCargar.font")); // NOI18N
btnCargar.setForeground(resourceMap.getColor("btnCargar.foreground")); // NOI18N
btnCargar.setText(resourceMap.getString("btnCargar.text")); // NOI18N
btnCargar.setName("btnCargar"); // NOI18N
btnCargar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCargarActionPerformed(evt);
    }
});
```

```
btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
btnVolver.setName("btnVolver"); // NOI18N
btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        Volver(evt);
    }
});
```

```
btnSiguiente.setFont(resourceMap.getFont("btnSiguiente.font")); // NOI18N
btnSiguiente.setForeground(resourceMap.getColor("btnSiguiente.foreground")); // NOI18N
btnSiguiente.setText(resourceMap.getString("btnSiguiente.text")); // NOI18N
btnSiguiente.setEnabled(false);
btnSiguiente.setName("btnSiguiente"); // NOI18N
btnSiguiente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        siguiente(evt);
    }
});
```

```
tabbedEtapas.setBackground(resourceMap.getColor("tabbedEtapas.background"));
tabbedEtapas.setForeground(resourceMap.getColor("tabbedEtapas.foreground")); // NOI18N
tabbedEtapas.setFont(resourceMap.getFont("tabbedEtapas.font")); // NOI18N
tabbedEtapas.setName("tabbedEtapas"); // NOI18N
```

```
panelEtapa1.setBackground(resourceMap.getColor("panelEtapa1.background")); // NOI18N
panelEtapa1.setName("panelEtapa1"); // NOI18N
```

```
lblSellInicio.setFont(resourceMap.getFont("lblSellInicio.font")); // NOI18N
lblSellInicio.setForeground(resourceMap.getColor("lblSellInicio.foreground")); // NOI18N
lblSellInicio.setText(resourceMap.getString("lblSellInicio.text")); // NOI18N
lblSellInicio.setName("lblSellInicio"); // NOI18N
```

```
btnGenerarInicio.setFont(resourceMap.getFont("btnGenerarInicio.font")); // NOI18N
btnGenerarInicio.setForeground(resourceMap.getColor("btnGenerarInicio.foreground"));
btnGenerarInicio.setIcon(resourceMap.getIcon("btnGenerarInicio.icon")); // NOI18N
btnGenerarInicio.setText(resourceMap.getString("btnGenerarInicio.text")); // NOI18N
btnGenerarInicio.setEnabled(false);
btnGenerarInicio.setName("btnGenerarInicio"); // NOI18N
btnGenerarInicio.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```

        generar_inicio(evt);
    }
});

txtInicio.setFont(resourceMap.getFont("txtInicio.font")); // NOI18N
txtInicio.setText(resourceMap.getString("txtInicio.text")); // NOI18N
txtInicio.setName("txtInicio"); // NOI18N

lblEstrella.setIcon(resourceMap.getIcon("lblEstrella.icon")); // NOI18N
lblEstrella.setText(resourceMap.getString("lblEstrella.text")); // NOI18N
lblEstrella.setName("lblEstrella"); // NOI18N

javax.swing.GroupLayout panelEtapa1Layout = new javax.swing.GroupLayout(panelEtapa1);
panelEtapa1.setLayout(panelEtapa1Layout);
panelEtapa1Layout.setHorizontalGroup(
    panelEtapa1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelEtapa1Layout.createSequentialGroup()
            .add(panelEtapa1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(panelEtapa1Layout.createSequentialGroup()
                    .addComponent(lblSellInicio)
                    .addContainerGap(26, Short.MAX_VALUE))
                .addGroup(panelEtapa1Layout.createSequentialGroup()
                    .addComponent(lblEstrella)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(txtInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(31, 31, 31)
                    .addComponent(btnGenerarInicio)
                    .addGap(121, 121, 121))))
            .addContainerGap())
);
panelEtapa1Layout.setVerticalGroup(
    panelEtapa1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelEtapa1Layout.createSequentialGroup()
            .addComponent(lblSellInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(26, Short.MAX_VALUE)
            .addComponent(lblEstrella, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(btnGenerarInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(26, Short.MAX_VALUE)
            .addComponent(lblEstrella, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
);
panelEtapa1Layout.setHorizontalGroup(
    panelEtapa1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelEtapa1Layout.createSequentialGroup()
            .addComponent(lblSellInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(26, Short.MAX_VALUE)
            .addComponent(lblEstrella, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(btnGenerarInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(26, Short.MAX_VALUE)
            .addComponent(lblEstrella, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
);
panelEtapa1Layout.setVerticalGroup(
    panelEtapa1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelEtapa1Layout.createSequentialGroup()
            .addComponent(lblSellInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(26, Short.MAX_VALUE)
            .addComponent(lblEstrella, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(btnGenerarInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(26, Short.MAX_VALUE)
            .addComponent(lblEstrella, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
);

```

```

        .addGroup(panelEtapa1Layout.createSequentialGroup()
            .addGap(21, 21, 21)

        .addGroup(panelEtapa1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
        )
            .addComponent(btnGenerarInicio)
            .addComponent(txtInicio, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGroup(panelEtapa1Layout.createSequentialGroup()
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(lblEstrella)))
            .addContainerGap(18, Short.MAX_VALUE)
        );

        tabbedEtapas.addTab(resourceMap.getString("panelEtapa1.TabConstraints.tabTitle"),
panelEtapa1); // NOI18N

        panelEtapa2.setBackground(resourceMap.getColor("panelEtapa2.background")); // NOI18N
        panelEtapa2.setName("panelEtapa2"); // NOI18N

        lblPaso.setFont(resourceMap.getFont("lblPaso.font")); // NOI18N
        lblPaso.setForeground(resourceMap.getColor("lblPaso.foreground")); // NOI18N
        lblPaso.setText(resourceMap.getString("lblPaso.text")); // NOI18N
        lblPaso.setName("lblPaso"); // NOI18N

        btnCalcularPaso.setFont(resourceMap.getFont("btnCalcularPaso.font")); // NOI18N
        btnCalcularPaso.setForeground(resourceMap.getColor("btnCalcularPaso.foreground"));
        btnCalcularPaso.setText(resourceMap.getString("btnCalcularPaso.text")); // NOI18N
        btnCalcularPaso.setEnabled(false);
        btnCalcularPaso.setName("btnCalcularPaso"); // NOI18N
        btnCalcularPaso.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                calcular_paso(evt);
            }
        });

        btnAleatorio.setFont(resourceMap.getFont("btnAleatorio.font")); // NOI18N
        btnAleatorio.setForeground(resourceMap.getColor("btnAleatorio.foreground")); // NOI18N
        btnAleatorio.setText(resourceMap.getString("btnAleatorio.text")); // NOI18N
        btnAleatorio.setEnabled(false);
        btnAleatorio.setName("btnAleatorio"); // NOI18N
        btnAleatorio.addActionListener(new java.awt.event.ActionListener() {

```

```

    public void actionPerformed(java.awt.event.ActionEvent evt) {
        paso_aleatorio(evt);
    }
});

txtPaso.setFont(resourceMap.getFont("txtPaso.font")); // NOI18N
txtPaso.setText(resourceMap.getString("txtPaso.text")); // NOI18N
txtPaso.setName("txtPaso"); // NOI18N

lblEstrellaAzul2.setIcon(resourceMap.getIcon("lblEstrellaAzul2.icon")); // NOI18N
lblEstrellaAzul2.setText(resourceMap.getString("lblEstrellaAzul2.text")); // NOI18N
lblEstrellaAzul2.setName("lblEstrellaAzul2"); // NOI18N

javax.swing.GroupLayout panelEtapa2Layout = new javax.swing.GroupLayout(panelEtapa2);
panelEtapa2.setLayout(panelEtapa2Layout);
panelEtapa2Layout.setHorizontalGroup(
    panelEtapa2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelEtapa2Layout.createSequentialGroup())
        .addContainerGap()

    .addGroup(panelEtapa2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelEtapa2Layout.createSequentialGroup())
        .addComponent(lblPaso)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(panelEtapa2Layout.createSequentialGroup())

    .addGroup(panelEtapa2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
        .addComponent(btnCalcularPaso, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btnAleatorio, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGap(48, 48, 48)
        .addComponent(txtPaso, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 47,
Short.MAX_VALUE)
        .addComponent(lblEstrellaAzul2)
        .addGap(45, 45, 45))))
);
panelEtapa2Layout.setVerticalGroup(
    panelEtapa2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```



```

        .addGroup(panelEtapa2Layout.createSequentialGroup())
        .addContainerGap()
        .addComponent(lblPaso)

    .addGroup(panelEtapa2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelEtapa2Layout.createSequentialGroup())
        .addGap(18, 18, 18)
        .addComponent(btnCalcularPaso)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btnAleatorio)
        .addGroup(panelEtapa2Layout.createSequentialGroup())
        .addGap(31, 31, 31)
        .addComponent(txtPaso, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(panelEtapa2Layout.createSequentialGroup())
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(lblEstrellaAzul2)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

```

```

    tabbedEtapas.addTab(resourceMap.getString("panelEtapa2.TabConstraints.tabTitle"),
panelEtapa2); // NOI18N

```

```

panelEtapa3.setBackground(resourceMap.getColor("panelEtapa3.background")); // NOI18N
panelEtapa3.setName("panelEtapa3"); // NOI18N

```

```

lblSelVar.setFont(resourceMap.getFont("lblSelVar.font")); // NOI18N
lblSelVar.setForeground(resourceMap.getColor("lblSelVar.foreground")); // NOI18N
lblSelVar.setText(resourceMap.getString("lblSelVar.text")); // NOI18N
lblSelVar.setName("lblSelVar"); // NOI18N

```

```

lblTamM.setFont(resourceMap.getFont("lblTamM.font")); // NOI18N
lblTamM.setForeground(resourceMap.getColor("lblTamM.foreground")); // NOI18N
lblTamM.setText(resourceMap.getString("lblTamM.text")); // NOI18N
lblTamM.setName("lblTamM"); // NOI18N

```

```

txtTamM.setFont(resourceMap.getFont("txtTamM.font")); // NOI18N
txtTamM.setText(resourceMap.getString("txtTamM.text")); // NOI18N
txtTamM.setName("txtTamM"); // NOI18N

```

```

btnCalcular.setFont(resourceMap.getFont("btnCalcular.font")); // NOI18N
btnCalcular.setForeground(resourceMap.getColor("btnCalcular.foreground")); // NOI18N

```

```

btnCalcular.setText(resourceMap.getString("btnCalcular.text")); // NOI18N
btnCalcular.setEnabled(false);
btnCalcular.setName("btnCalcular"); // NOI18N
btnCalcular.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        calcula_tamaño(evt);
    }
});

comboVariables.setFont(resourceMap.getFont("comboVariables.font")); // NOI18N
comboVariables.setForeground(resourceMap.getColor("comboVariables.foreground"));
comboVariables.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1",
"Item 2", "Item 3", "Item 4" }));
comboVariables.setName("comboVariables"); // NOI18N

lblEstrellaAzulCielo.setIcon(resourceMap.getIcon("lblEstrellaAzulCielo.icon")); // NOI18N
lblEstrellaAzulCielo.setText(resourceMap.getString("lblEstrellaAzulCielo.text")); // NOI18N
lblEstrellaAzulCielo.setName("lblEstrellaAzulCielo"); // NOI18N

javax.swing.GroupLayout panelEtapa3Layout = new javax.swing.GroupLayout(panelEtapa3);
panelEtapa3.setLayout(panelEtapa3Layout);
panelEtapa3Layout.setHorizontalGroup(
    panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(panelEtapa3Layout.createSequentialGroup()
            .add(panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(panelEtapa3Layout.createSequentialGroup()
                    .addGap(20, 20, 20)
                )
            )
        )

.addGroup(panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelEtapa3Layout.createSequentialGroup()
        .addGroup(panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .add(panelEtapa3Layout.createSequentialGroup()
                .addComponent(txtTamM, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(btnCalcular)
                .addContainerGap(192, Short.MAX_VALUE)
            )
            .addGroup(panelEtapa3Layout.createSequentialGroup()
                .addGroup(panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(panelEtapa3Layout.createSequentialGroup()
                        .addGroup(panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(panelEtapa3Layout.createSequentialGroup()
                                .addComponent(comboVariables, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addComponent(lblTamM)
                                .addComponent(lblSelVar)
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 66,
Short.MAX_VALUE)
                            )
                        )
                    )
                )
            )
        )
    )
)

```

```

        .addComponent(lblEstrellaAzulCielo)
        .addGap(37, 37, 37)))
);
panelEtapa3Layout.setVerticalGroup(
    panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(panelEtapa3Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING
)
        .addComponent(lblEstrellaAzulCielo)
        .addGroup(panelEtapa3Layout.createSequentialGroup()
            .addComponent(lblSelVar)
            .addGap(3, 3, 3)
            .addComponent(comboVariables, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(lblTamM)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(panelEtapa3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
        .addComponent(txtTamM, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnCalcular))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

tabbedEtapas.addTab(resourceMap.getString("panelEtapa3.TabConstraints.tabTitle"),
panelEtapa3); // NOI18N

lblCompleta.setFont(resourceMap.getFont("lblCompleta.font")); // NOI18N
lblCompleta.setForeground(resourceMap.getColor("lblCompleta.foreground")); // NOI18N
lblCompleta.setText(resourceMap.getString("lblCompleta.text")); // NOI18N
lblCompleta.setName("lblCompleta"); // NOI18N

lblDividerAzul.setIcon(resourceMap.getIcon("lblDividerAzul.icon")); // NOI18N
lblDividerAzul.setText(resourceMap.getString("lblDividerAzul.text")); // NOI18N
lblDividerAzul.setName("lblDividerAzul"); // NOI18N

```



```

        .addGroup(mainPanelLayout.createSequentialGroup())
        .addGap(47, 47, 47)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
        .addComponent(tabbedEtapas, javax.swing.GroupLayout.Alignment.LEADING,
0, 0, Short.MAX_VALUE)
        .addComponent(lblCompleta, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGap(18, 18, 18)
        .addComponent(btnAyuda, javax.swing.GroupLayout.PREFERRED_SIZE, 60,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(btnCargar)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
        .addComponent(lblCorazon)
        .addGap(27, 27, 27))))
    .addContainerGap()
);
mainPanelLayout.setVerticalGroup(
mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addComponent(lblHeader)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 449,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(mainPanelLayout.createSequentialGroup()

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addGap(29, 29, 29)
            .addComponent(lblCompleta)
            .addGap(18, 18, 18)
            .addComponent(tabbedEtapas, javax.swing.GroupLayout.PREFERRED_SIZE, 147,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addGap(118, 118, 118)
            .addComponent(btnAyuda)))

```

```

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(lblPulse)
    .addGap(47, 47, 47)
    .addComponent(btnCargar)
    .addGap(23, 23, 23))
    .addGroup(mainPanelLayout.createSequentialGroup()
    .addGap(44, 44, 44)
    .addComponent(lblCorazon))))
    .addGap(14, 14, 14)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(btnVolver)
    .addComponent(btnSiguiete, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblDividerAzul))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

jMenuBar1.setName("jMenuBar1"); // NOI18N

jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

jMenuItem1.setAction(actionMap.get("quit")); // NOI18N
jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
jMenuItem1.setName("jMenuItem1"); // NOI18N
jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);

```

```

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    );
pack();
}

```

```

private void Volver(java.awt.event.ActionEvent evt) {
    this.dispose();
}

```

```

private void btnCargarActionPerformed(java.awt.event.ActionEvent evt) {
    Archivos inicio=new Archivos();
    nombreArchivo=inicio.manejar_archivo(tblDatos, comboVariables);
    lblPulse.setVisible(false);
    lblCompleta.setVisible(true);
    tabbedEtapas.setVisible(true);
    btnGenerarInicio.setEnabled(true);
    btnAyuda.setVisible(true);
    btnSiguiente.setEnabled(true);
}

```

```

private void generar_inicio(java.awt.event.ActionEvent evt) {
    Object ka=null;
    String kaaa=null;
    Operaciones op=new Operaciones();
    LinkedList kaa=new LinkedList();
    btnCalcularPaso.setEnabled(true);
    btnAleatorio.setEnabled(true);
    kaa=op.aleatorio(tblDatos.getRowCount());
    ka=kaa.get(0);
    kaaa=ka.toString();
    kaaa=kaaa.replace(".0", " ");
    txtInicio.setText(kaaa);
}

```

```

private void calcular_paso(java.awt.event.ActionEvent evt) {
    double div=0;
    double ka=0;
    String kaa=null;
    String jaa=null;
    Object wee=new Object();
    kaa=txtInicio.getText();
    kaa=kaa.trim();
    ka=Double.parseDouble(kaa);
    div=tblDatos.getRowCount()/ka;
    div=Math.ceil(div);
    wee=div;
    jaa=wee.toString();
    jaa=jaa.replace(".0","");
    jaa=jaa.trim();
    txtPaso.setText(jaa);
    btnCalcular.setEnabled(true);
}

private void paso_aleatorio(java.awt.event.ActionEvent evt) {
    Operaciones op=new Operaciones();
    Object n=new Object();
    String nn=null;
    LinkedList numeros=op.aleatorio(tblDatos.getRowCount());
    n=numeros.get(0);
    nn=n.toString();
    if(nn.equals("-0.0"))
        nn=nn.replace("-0.0", "0");
    else
        nn=nn.replace(".0","");
    txtPaso.setText(nn);
    btnSiguiente.setEnabled(true);
}

private void calcula_tamaño(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null, "Próximamente codificado!!!");
}

private void siguiente(java.awt.event.ActionEvent evt) {
    String tam=null;
    String seleccionado=comboVariables.getSelectedItem().toString();
    int tamPoblacion=tblDatos.getRowCount();
}

```



```

String tampob=null;
String kinicio=null;
String npaso=null;
kinicio=txtInicio.getText();
kinicio=kinicio.trim();
npaso=txtPaso.getText();
npaso=npaso.trim();
Object y=new Object();
y=tblDatos.getRowCount();
tampob=y.toString();
int t=0;
tam=txtTamM.getText();
if(tam.equals("0"))
    JOptionPane.showMessageDialog(null, "Porfavor elige el tamaño de la Muestra...");
else
{
    t=Integer.parseInt(tam);
    if(t>tblDatos.getRowCount())
        JOptionPane.showMessageDialog(null, "El tamaño de la muestra debe ser menor al de la
población (" +tamPoblacion+"...)");
    else
    {
        JOptionPane.showMessageDialog(null, "Seleccionó la variable:
"+seleccionado+"\nTamaño de la muestra: " +tam
        +"\nInicio: "+txtInicio.getText()+"\nPaso: "+txtPaso.getText());
        String[]
hola={"Sistemático",nombreArchivo,tam,tampob,txtInicio.getText(),txtPaso.getText(),seleccionado
};
        Muestreo muestreo=new Muestreo(comboVariables,tblDatos.getModel(),tam,hola);
        muestreo.empezar_metodo(1, nombreArchivo,"95");
    }
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Sistematico().setVisible(true);
        }
    });
}

```

```

@Action
public void cargar_ayuda() {
    String hola[]={ "C:\\Users\\USUARIO\\Documents\\h.txt",this.getName()};
    //String hola[]={ "D:\\Documents and Settings\\xcerna\\Mis
Documentos\\vals.txt",this.getName()};
    //String hola[]={ "C:\\Documents and Settings\\PATRICIA\\Mis
Documentos\\prueba.txt",this.getName()};
    Ayuda.main(hola);
}

```

```

// Declaración de Variables
private javax.swing.JButton btnAleatorio;
private javax.swing.JButton btnAyuda;
private javax.swing.JToggleButton btnCalcular;
private javax.swing.JButton btnCalcularPaso;
private javax.swing.JButton btnCargar;
private javax.swing.JToggleButton btnGenerarInicio;
private javax.swing.JButton btnSiguiente;
private javax.swing.JButton btnVolver;
private javax.swing.JComboBox comboVariables;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextPane jTextPane1;
private javax.swing.JScrollPane jsDatos;
private javax.swing.JLabel lblCompleta;
private javax.swing.JLabel lblCorazon;
private javax.swing.JLabel lblDividerAzul;
private javax.swing.JLabel lblEstrella;
private javax.swing.JLabel lblEstrellaAzul2;
private javax.swing.JLabel lblEstrellaAzulCielo;
private javax.swing.JLabel lblHeader;
private javax.swing.JLabel lblPaso;
private javax.swing.JLabel lblPulse;
private javax.swing.JLabel lblSelInicio;
private javax.swing.JLabel lblSelVar;
private javax.swing.JLabel lblTamM;
private javax.swing.JPanel mainPanel;
private javax.swing.JPanel panelEtapa1;
private javax.swing.JPanel panelEtapa2;

```

```

private javax.swing.JPanel panelEtapa3;
private javax.swing.JTabbedPane tabbedEtapas;
private javax.swing.JTable tblDatos;
private javax.swing.JTextField txtInicio;
private javax.swing.JTextField txtPaso;
private javax.swing.JTextField txtTamM;
}

```

### 9.2.5 Codificacion.java

```

/*
 * Pantalla de codifiación.
 * Tiene la lista de la población
 * Created on 21/10/2010, 09:16:13 PM
 */
package pt02.vistas;
import java.awt.Component;
import java.util.EventObject;
import java.util.LinkedList;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.event.CellEditorListener;
import javax.swing.table.TableColumn;
import javax.swing.table.TableColumnModel;
import javax.swing.table.TableModel;
import org.jdesktop.application.Action;
import pt02.kernel.Resultados;
import pt02.modelosTablas.TablaEstratos;
import pt02.modelosTablas.TablaFinal;
import pt02.modelosTablas.TablaInicial;
import pt02.modelosTablas.TablaLimites;
import pt02.modelosTablas.TablaMuestra;
import pt02.modelosTablas.TablaResPob;
/**
 * @author XiMe
 */
public class Codificacion extends javax.swing.JFrame
{
    public int numeroEstratas;
    public TableModel invertido;
    public TableModel presentacion;
    public String nombreArchivo;
    public String modoCodif;

```

```

public String variable;
public LinkedList datos;
public String muestras_t;
public LinkedList[] tMs; //lista ligada con los tamaños de las muestras
/** Constructor Estratificado Promedio */
public Codificacion(TableModel codif,String modo,TableModel codifInv,String nomA,LinkedList
dat,String v)
{
    this.invertido=codifInv;
    this.nombreArchivo=nomA;
    this.datos=dat;
    this.modosCodif=modo;
    this.variable=v;
    initComponents(codif,modo,nomA);
    comboAlgoritmo.setVisible(false);
    //if(modo.equals("Promedio") || modo.equals("Archivo"))
        btnAgregar.setVisible(false);
    // if(modo.equals("Archivo"))
        pnlCodigo.setVisible(false);
}
/**Constructor Estratificado Manual y Archivo */
public Codificacion(TableModel codifP,TableModel codif,String modo,String nomA,LinkedList
dat,String v,int id,LinkedList[] tms,String toma_muestras)
{
    this.nombreArchivo=nomA;
    this.datos=dat;
    this.modosCodif=modo;
    this.variable=v;
    this.numeroEstratas=id;
    this.tMs=tms;
    this.invertido=codif;
    this.presentacion=codifP;
    this.muestras_t=toma_muestras;
    initComponents(codifP,modo,nomA);
}

@SuppressWarnings("unchecked")
private void initComponents() {
    mainPanel = new javax.swing.JPanel();
    lblHeader = new javax.swing.JLabel();
    jsDatos = new javax.swing.JScrollPane();
    tblDatos = new javax.swing.JTable();
}

```

```

btnAgregar = new javax.swing.JButton();
btnSeleccionar = new javax.swing.JButton();
btnSiguiente = new javax.swing.JButton();
pnlCodigo = new javax.swing.JPanel();
jcbCodigo = new javax.swing.JComboBox();
lblIndique = new javax.swing.JLabel();
btnInvertir = new javax.swing.JButton();
pnlElija = new javax.swing.JPanel();
rbtnConReemplazo = new javax.swing.JRadioButton();
rbtnSinReemplazo = new javax.swing.JRadioButton();
lblElija = new javax.swing.JLabel();
btnAyuda = new javax.swing.JButton();
comboAlgoritmo = new javax.swing.JComboBox();
lblModo = new javax.swing.JLabel();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResour
ceMap(Codificacion.class);
setTitle(resourceMap.getString("Form.title")); // NOI18N

mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
mainPanel.setForeground(resourceMap.getColor("mainPanel.foreground")); // NOI18N
mainPanel.setName("mainPanel"); // NOI18N

lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
lblHeader.setName("lblHeader"); // NOI18N

jsDatos.setBackground(resourceMap.getColor("jsDatos.background")); // NOI18N
jsDatos.setBorder(javax.swing.BorderFactory.createTitledBorder(new
javax.swing.border.LineBorder(resourceMap.getColor("jsDatos.border.border.lineColor"), 1, true),
resourceMap.getString("jsDatos.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsDatos.border.titleFont"),
resourceMap.getColor("jsDatos.border.titleColor"))); // NOI18N
jsDatos.setName("jsDatos"); // NOI18N

```

```

tblDatos.setFont(resourceMap.getFont("tblDatos.font")); // NOI18N
tblDatos.setForeground(resourceMap.getColor("tblDatos.foreground")); // NOI18N
tblDatos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null},
        {null, null},
        {null, null},
        {null, null}
    },
    new String [] {
        "Title 1", "Title 2"
    }
));
tblDatos.setGridColor(resourceMap.getColor("tblDatos.gridColor")); // NOI18N
tblDatos.setName("tblDatos"); // NOI18N
tblDatos.setSelectionBackground(resourceMap.getColor("tblDatos.selectionBackground"));
tblDatos.setSelectionForeground(resourceMap.getColor("tblDatos.selectionForeground"));
jsDatos.setViewportView(tblDatos);

```

```

btnAgregar.setFont(resourceMap.getFont("btnAgregar.font")); // NOI18N
btnAgregar.setForeground(resourceMap.getColor("btnAgregar.foreground")); // NOI18N
btnAgregar.setText(resourceMap.getString("btnAgregar.text")); // NOI18N
btnAgregar.setName("btnAgregar"); // NOI18N
btnAgregar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAgregarActionPerformed(evt);
    }
});

```

```

btnSeleccionar.setFont(resourceMap.getFont("btnSeleccionar.font")); // NOI18N
btnSeleccionar.setForeground(resourceMap.getColor("btnSeleccionar.foreground"));
btnSeleccionar.setText(resourceMap.getString("btnSeleccionar.text")); // NOI18N

```

```

btnSeleccionar.setActionCommand(resourceMap.getString("btnSeleccionar.actionCommand"));
btnSeleccionar.setName("btnSeleccionar"); // NOI18N
btnSeleccionar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        volver(evt);
    }
});

```

```

    javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActio
nMap(Codificacion.class, this);
    btnSiguiente.setAction(actionMap.get("siguiente")); // NOI18N
    btnSiguiente.setFont(resourceMap.getFont("btnSiguiente.font")); // NOI18N
    btnSiguiente.setForeground(resourceMap.getColor("btnSiguiente.foreground")); // NOI18N
    btnSiguiente.setText(resourceMap.getString("btnSiguiente.text")); // NOI18N
    btnSiguiente.setName("btnSiguiente"); // NOI18N

    pnlCodigo.setBackground(resourceMap.getColor("pnlCodigo.background")); // NOI18N

pnlCodigo.setBorder(javax.swing.BorderFactory.createLineBorder(resourceMap.getColor("pnlCodi
go.border.lineColor"))); // NOI18N
    pnlCodigo.setName("pnlCodigo"); // NOI18N

    jcbCodigo.setFont(resourceMap.getFont("jcbCodigo.font")); // NOI18N
    jcbCodigo.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "-", "-", "-" }));
    jcbCodigo.setName("jcbCodigo"); // NOI18N

    lblIndique.setFont(resourceMap.getFont("lblIndique.font")); // NOI18N
    lblIndique.setForeground(resourceMap.getColor("lblIndique.foreground")); // NOI18N
    lblIndique.setText(resourceMap.getString("lblIndique.text")); // NOI18N
    lblIndique.setName("lblIndique"); // NOI18N

    javax.swing.GroupLayout pnlCodigoLayout = new javax.swing.GroupLayout(pnlCodigo);
    pnlCodigo.setLayout(pnlCodigoLayout);
    pnlCodigoLayout.setHorizontalGroup(
        pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlCodigoLayout.createSequentialGroup()
                .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblIndique))
            .addGap(33, Short.MAX_VALUE))
    );
    pnlCodigoLayout.setVerticalGroup(
        pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlCodigoLayout.createSequentialGroup()
                .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblIndique))
            .addGap(33, Short.MAX_VALUE))
    );
    pnlCodigoLayout.setHorizontalGroup(
        pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlCodigoLayout.createSequentialGroup()
                .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblIndique))
            .addGap(33, Short.MAX_VALUE))
    );
    pnlCodigoLayout.setVerticalGroup(
        pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlCodigoLayout.createSequentialGroup()
                .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblIndique))
            .addGap(33, Short.MAX_VALUE))
    );
    pnlCodigoLayout.setHorizontalGroup(
        pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlCodigoLayout.createSequentialGroup()
                .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblIndique))
            .addGap(33, Short.MAX_VALUE))
    );
    pnlCodigoLayout.setVerticalGroup(
        pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlCodigoLayout.createSequentialGroup()
                .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(lblIndique))
            .addGap(33, Short.MAX_VALUE))
    );

```

```

        .addComponent(lblIndique)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 16,
Short.MAX_VALUE)
        .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(39, 39, 39)
    );

    btnInvertir.setFont(resourceMap.getFont("btnInvertir.font")); // NOI18N
    btnInvertir.setForeground(resourceMap.getColor("btnInvertir.foreground")); // NOI18N
    btnInvertir.setText(resourceMap.getString("btnInvertir.text")); // NOI18N
    btnInvertir.setName("btnInvertir"); // NOI18N
    btnInvertir.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            invertir(evt);
        }
    });

    pnlElija.setBackground(resourceMap.getColor("pnlElija.background")); // NOI18N

    pnlElija.setBorder(javax.swing.BorderFactory.createLineBorder(resourceMap.getColor("pnlElija.bo
rder.lineColor"))); // NOI18N
    pnlElija.setName("pnlElija"); // NOI18N

    rbtnConReemplazo.setBackground(resourceMap.getColor("rbtnConReemplazo.background"));
    rbtnConReemplazo.setFont(resourceMap.getFont("rbtnConReemplazo.font")); // NOI18N
    rbtnConReemplazo.setForeground(resourceMap.getColor("rbtnConReemplazo.foreground"));
    rbtnConReemplazo.setText(resourceMap.getString("rbtnConReemplazo.text")); // NOI18N
    rbtnConReemplazo.setName("rbtnConReemplazo"); // NOI18N
    rbtnConReemplazo.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            rbtnConReemplazoMouseClicked(evt);
        }
    });

    rbtnSinReemplazo.setBackground(resourceMap.getColor("rbtnSinReemplazo.background"));
    rbtnSinReemplazo.setFont(resourceMap.getFont("rbtnSinReemplazo.font")); // NOI18N
    rbtnSinReemplazo.setForeground(resourceMap.getColor("rbtnSinReemplazo.foreground"));
    rbtnSinReemplazo.setSelected(true);
    rbtnSinReemplazo.setText(resourceMap.getString("rbtnSinReemplazo.text")); // NOI18N
    rbtnSinReemplazo.setName("rbtnSinReemplazo"); // NOI18N

```





```

        .addComponent(btnAyuda, javax.swing.GroupLayout.PREFERRED_SIZE, 59,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlElijaLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(lblElija))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
pnlElijaLayout.setVerticalGroup(
    pnlElijaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlElijaLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(lblElija)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(pnlElijaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlElijaLayout.createSequentialGroup())
        .addComponent(rbtnConReemplazo)
        .addGap(18, 18, 18)
        .addComponent(rbtnSinReemplazo))
        .addComponent(btnAyuda))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(comboAlgoritmo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(16, Short.MAX_VALUE))
);

lblModo.setBackground(resourceMap.getColor("lblModo.background")); // NOI18N
lblModo.setFont(resourceMap.getFont("lblModo.font")); // NOI18N
lblModo.setForeground(resourceMap.getColor("lblModo.foreground")); // NOI18N
lblModo.setText(resourceMap.getString("lblModo.text")); // NOI18N
lblModo.setName("lblModo"); // NOI18N

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblHeader)
        .addGroup(mainPanelLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 371,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
    .addComponent(btnAgregar, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addComponent(btnInvertir, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addGap(77, 77, 77)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(lblModo, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(pnlCodigo, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(pnlElija, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
    .addContainerGap(104, Short.MAX_VALUE)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(btnSeleccionar)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 447,
Short.MAX_VALUE)
        .addComponent(btnSiguiente)
        .addGap(19, 19, 19))
);
mainPanelLayout.setVerticalGroup(
mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addComponent(lblHeader)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addGap(18, 18, 18)

```

```

        .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 436,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(mainPanelLayout.createSequentialGroup())
        .addGap(29, 29, 29)
        .addComponent(lblModo)
        .addGap(18, 18, 18)
        .addComponent(pnlCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(pnlElija, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btnAgregar)
        .addGap(18, 18, 18)
        .addComponent(btnInvertir)))
        .addGap(18, 18, 18)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(btnSeleccionar)
        .addComponent(btnSiguiente))
        .addContainerGap()
    );

jMenuBar1.setName("jMenuBar1"); // NOI18N
jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

jMenuItem1.setAction(actionMap.get("quit")); // NOI18N
jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
jMenuItem1.setName("jMenuItem1"); // NOI18N
jMenu1.add(jMenuItem1);
jMenuBar1.add(jMenu1);
jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);
setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE, 798,
Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE, 608,
javax.swing.GroupLayout.PREFERRED_SIZE)
    );
    pack();
}
private void volver(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void invertir(java.awt.event.ActionEvent evt) {
    tblDatos.setModel(invertido);
    btnInvertir.setEnabled(false);
}

private void btnAgregarActionPerformed(java.awt.event.ActionEvent evt) {
    // actualizar tabla de datos
    int num_columnas=tblDatos.getColumnCount();
    LinkedList nombres=new LinkedList();
    LinkedList[] valores=new LinkedList[num_columnas+1];

    //actualizar combo box
    String[] n=new String[tblDatos.getColumnCount()];
    for(int i=1;i<tblDatos.getColumnCount();i++)
        n[i-1]="Código "+i;
    Object ff=n.clone();
    jcbCodigo.setModel(new javax.swing.DefaultComboBoxModel(n));
}

private void rbtnConReemplazoMouseClicked(java.awt.event.MouseEvent evt) {
    if(rbtnSinReemplazo.isSelected())
        rbtnSinReemplazo.setSelected(false);
    comboAlgoritmo.setVisible(false);
}

private void rbtnSinReemplazoMouseClicked(java.awt.event.MouseEvent evt) {
    if(rbtnConReemplazo.isSelected())

```

```

        rbtnConReemplazo.setSelected(false);
        comboAlgoritmo.setVisible(true);
    }

    private void ayuda(java.awt.event.ActionEvent evt) {
        String hola[]{"C:\\Users\\USUARIO\\Documents\\h.txt",this.getName()};
        //String hola[]{"D:\\Documents and Settings\\xcerna\\Mis
Documentos\\vals.txt",this.getName()};
        // String hola[]{"C:\\Documents and Settings\\PATRICIA\\Mis
Documentos\\prueba.txt",this.getName()};
        Ayuda.main(hola);
    }

    private void initComponents(TableModel codificacion,String modo,String archo)
    {
        mainPanel = new javax.swing.JPanel();
        lblHeader = new javax.swing.JLabel();
        jsDatos = new javax.swing.JScrollPane();
        tblDatos = new javax.swing.JTable();
        btnAgregar = new javax.swing.JButton();
        btnSeleccionar = new javax.swing.JButton();
        btnSiguiente = new javax.swing.JButton();
        pnlCodigo = new javax.swing.JPanel();
        jcbCodigo = new javax.swing.JComboBox();
        lblIndique = new javax.swing.JLabel();
        btnInvertir = new javax.swing.JButton();
        pnlElija = new javax.swing.JPanel();
        rbtnConReemplazo = new javax.swing.JRadioButton();
        rbtnSinReemplazo = new javax.swing.JRadioButton();
        lblElija = new javax.swing.JLabel();
        btnAyuda = new javax.swing.JButton();
        comboAlgoritmo = new javax.swing.JComboBox();
        lblModo = new javax.swing.JLabel();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
        jMenuItem1 = new javax.swing.JMenuItem();
        jMenu2 = new javax.swing.JMenu();
        nombreArchivo=archo;
        modoCodif=modo;

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

    org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(Codificacion.class);
    setTitle(resourceMap.getString("Form.title")); // NOI18N

    mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
    mainPanel.setForeground(resourceMap.getColor("mainPanel.foreground")); // NOI18N
    mainPanel.setName("mainPanel"); // NOI18N

    lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
    lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
    lblHeader.setName("lblHeader"); // NOI18N

    jsDatos.setBackground(resourceMap.getColor("jsDatos.background")); // NOI18N
    jsDatos.setBorder(javax.swing.BorderFactory.createTitledBorder(new
javax.swing.border.LineBorder(resourceMap.getColor("jsDatos.border.border.lineColor"), 1, true),
resourceMap.getString("jsDatos.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsDatos.border.titleFont"),
resourceMap.getColor("jsDatos.border.titleColor"))); // NOI18N
    jsDatos.setName("jsDatos"); // NOI18N

    tblDatos.setFont(resourceMap.getFont("tblDatos.font")); // NOI18N
    tblDatos.setForeground(resourceMap.getColor("tblDatos.foreground")); // NOI18N
    tblDatos.setGridColor(resourceMap.getColor("tblDatos.gridColor")); // NOI18N
    tblDatos.setName("tblDatos"); // NOI18N
    tblDatos.setSelectionBackground(resourceMap.getColor("tblDatos.selectionBackground"));
    tblDatos.setSelectionForeground(resourceMap.getColor("tblDatos.selectionForeground"));
    tblDatos.setModel(codificacion);//*****
    jsDatos.setViewPortView(tblDatos);

    btnAgregar.setFont(resourceMap.getFont("btnAgregar.font")); // NOI18N
    btnAgregar.setForeground(resourceMap.getColor("btnAgregar.foreground")); // NOI18N
    btnAgregar.setText(resourceMap.getString("btnAgregar.text")); // NOI18N
    btnAgregar.setName("btnAgregar"); // NOI18N
    btnAgregar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnAgregarActionPerformed(evt);
        }
    });

```

```

btnSeleccionar.setFont(resourceMap.getFont("btnSeleccionar.font")); // NOI18N
btnSeleccionar.setForeground(resourceMap.getColor("btnSeleccionar.foreground"));
btnSeleccionar.setText(resourceMap.getString("btnSeleccionar.text")); // NOI18N
btnSeleccionar.setName("btnSeleccionar"); // NOI18N
btnSeleccionar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        volver(evt);
    }
});

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActionMap(Codificacion.class, this);
btnSiguiente.setAction(actionMap.get("siguiente")); // NOI18N
btnSiguiente.setFont(resourceMap.getFont("btnSiguiente.font")); // NOI18N
btnSiguiente.setForeground(resourceMap.getColor("btnSiguiente.foreground")); // NOI18N
btnSiguiente.setText(resourceMap.getString("btnSiguiente.text")); // NOI18N
btnSiguiente.setEnabled(true);
btnSiguiente.setName("btnSiguiente"); // NOI18N

pnlCodigo.setBackground(resourceMap.getColor("pnlCodigo.background")); // NOI18N
pnlCodigo.setBorder(javax.swing.BorderFactory.createLineBorder(resourceMap.getColor("pnlCodigo.border.lineColor"))); // NOI18N
pnlCodigo.setName("pnlCodigo"); // NOI18N

jcbCodigo.setFont(resourceMap.getFont("jcbCodigo.font")); // NOI18N
jcbCodigo.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Código 1" }));
//*****
jcbCodigo.setName("jcbCodigo"); // NOI18N

lblIndique.setFont(resourceMap.getFont("lblIndique.font")); // NOI18N
lblIndique.setForeground(resourceMap.getColor("lblIndique.foreground")); // NOI18N
lblIndique.setText(resourceMap.getString("lblIndique.text")); // NOI18N
lblIndique.setName("lblIndique"); // NOI18N

javax.swing.GroupLayout pnlCodigoLayout = new javax.swing.GroupLayout(pnlCodigo);
pnlCodigo.setLayout(pnlCodigoLayout);
pnlCodigoLayout.setHorizontalGroup(
    pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlCodigoLayout.createSequentialGroup()
            .addContainerGap()

```



```

.addGroup(pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(lblIndique))
    .addContainerGap(33, Short.MAX_VALUE)
);
pnlCodigoLayout.setVerticalGroup(
    pnlCodigoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlCodigoLayout.createSequentialGroup()
    .addContainerGap()
    .addComponent(lblIndique)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 16,
Short.MAX_VALUE)
    .addComponent(jcbCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(39, 39, 39))
);

```

```

btnInvertir.setFont(resourceMap.getFont("btnInvertir.font")); // NOI18N
btnInvertir.setForeground(resourceMap.getColor("btnInvertir.foreground")); // NOI18N
btnInvertir.setText(resourceMap.getString("btnInvertir.text")); // NOI18N
btnInvertir.setName("btnInvertir"); // NOI18N
btnInvertir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        invertir(evt);
    }
});

```

```

pnlEliza.setBackground(resourceMap.getColor("pnlEliza.background")); // NOI18N

```

```

pnlEliza.setBorder(javax.swing.BorderFactory.createLineBorder(resourceMap.getColor("pnlEliza.bo
rder.lineColor"))); // NOI18N
pnlEliza.setName("pnlEliza"); // NOI18N

```

```

rbtnConReemplazo.setBackground(resourceMap.getColor("rbtnConReemplazo.background"));
rbtnConReemplazo.setFont(resourceMap.getFont("rbtnConReemplazo.font")); // NOI18N
rbtnConReemplazo.setForeground(resourceMap.getColor("rbtnConReemplazo.foreground"));
rbtnConReemplazo.setText(resourceMap.getString("rbtnConReemplazo.text")); // NOI18N
rbtnConReemplazo.setName("rbtnConReemplazo"); // NOI18N

```

```

rbtnConReemplazo.setSelected(true);
rbtnConReemplazo.addMouseListener(new java.awt.event.MouseAdapter() {
    @Override
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnConReemplazoMouseClicked(evt);
    }
});

```

```

rbtnSinReemplazo.setBackground(resourceMap.getColor("rbtnSinReemplazo.background"));
rbtnSinReemplazo.setFont(resourceMap.getFont("rbtnSinReemplazo.font")); // NOI18N
rbtnSinReemplazo.setForeground(resourceMap.getColor("rbtnSinReemplazo.foreground"));
rbtnSinReemplazo.setText(resourceMap.getString("rbtnSinReemplazo.text")); // NOI18N
rbtnSinReemplazo.setName("rbtnSinReemplazo"); // NOI18N
rbtnSinReemplazo.addMouseListener(new java.awt.event.MouseAdapter() {
    @Override
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnSinReemplazoMouseClicked(evt);
    }
});

```

```

lblElija.setFont(resourceMap.getFont("lblElija.font")); // NOI18N
lblElija.setForeground(resourceMap.getColor("lblElija.foreground")); // NOI18N
lblElija.setText(resourceMap.getString("lblElija.text")); // NOI18N
lblElija.setName("lblElija"); // NOI18N

```

```

btnAyuda.setIcon(resourceMap.getIcon("btnAyuda.icon")); // NOI18N
btnAyuda.setText(resourceMap.getString("btnAyuda.text")); // NOI18N
btnAyuda.setName("btnAyuda"); // NOI18N
btnAyuda.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ayuda(evt);
    }
});

```

```

comboAlgoritmo.setForeground(resourceMap.getColor("comboAlgoritmo.foreground"));
comboAlgoritmo.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Algoritmo XCL", "Algoritmo 1", "Algoritmo 2", "Algoritmo 3", "Algoritmo 4" }));
comboAlgoritmo.setName("comboAlgoritmo"); // NOI18N

```

```

javax.swing.GroupLayout pnlElijaLayout = new javax.swing.GroupLayout(pnlElija);
pnlElija.setLayout(pnlElijaLayout);
pnlElijaLayout.setHorizontalGroup(

```

```

        pnlElijaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlElijaLayout.createSequentialGroup()

.addGroup(pnlElijaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlElijaLayout.createSequentialGroup()
                .addGap(28, 28, 28)

.addGroup(pnlElijaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(rbbtnConReemplazo)
            .addComponent(rbbtnSinReemplazo)
            .addGroup(pnlElijaLayout.createSequentialGroup()
                .addGap(21, 21, 21)
                .addComponent(comboAlgoritmo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(18, 18, 18)
            .addComponent(btnAyuda, javax.swing.GroupLayout.PREFERRED_SIZE, 59,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(pnlElijaLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(lblElija))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    pnlElijaLayout.setVerticalGroup(
        pnlElijaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlElijaLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(lblElija)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(pnlElijaLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlElijaLayout.createSequentialGroup()
                .addComponent(rbbtnConReemplazo)
                .addGap(18, 18, 18)
                .addComponent(rbbtnSinReemplazo))
            .addComponent(btnAyuda))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(comboAlgoritmo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(16, Short.MAX_VALUE))
    );

    lblModo.setBackground(resourceMap.getColor("lblModo.background")); // NOI18N

```

```

lblModo.setFont(resourceMap.getFont("lblModo.font")); // NOI18N
lblModo.setForeground(resourceMap.getColor("lblModo.foreground")); // NOI18N
lblModo.setText(modos); // NOI18N *****
lblModo.setName(modos); // NOI18N *****

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblHeader)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addComponent(lblModo)
            .addGap(10, 10, 10)
            .addComponent(btnAgregar, javax.swing.GroupLayout.PREFERRED_SIZE, 77,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(10, 10, 10)
            .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(mainPanelLayout.createSequentialGroup()
                    .addComponent(btnInvertir, javax.swing.GroupLayout.PREFERRED_SIZE, 77,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addGroup(mainPanelLayout.createSequentialGroup()
                        .addComponent(lblModo)
                        .addGap(10, 10, 10)
                        .addComponent(pnlCodigo, javax.swing.GroupLayout.PREFERRED_SIZE, 104,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(10, 10, 10)
                        .addComponent(pnlElija, javax.swing.GroupLayout.PREFERRED_SIZE, 77,
                            javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(10, 10, 10)
                        .addComponent(btnSeleccionar)
                    )
                )
            )
        )
);

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 447,
Short.MAX_VALUE)
        .addComponent(btnSiguiete)
        .addGap(19, 19, 19))
    );
    mainPanelLayout.setVerticalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
    .addComponent(lblHeader)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addGroup(mainPanelLayout.createSequentialGroup()
    .addGap(18, 18, 18)
    .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 436,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(mainPanelLayout.createSequentialGroup()
    .addGap(29, 29, 29)
    .addComponent(lblModo)
    .addGap(18, 18, 18)
    .addComponent(pnlCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addComponent(pnlElija, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(btnAgregar)
    .addGap(18, 18, 18)
    .addComponent(btnInvertir)))
    .addGap(18, 18, 18)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnSeleccionar)
    .addComponent(btnSiguiete))
    .addContainerGap()
    );

jMenuBar1.setName("jMenuBar1"); // NOI18N

jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

```

```

jMenuItem1.setAction(actionMap.get("quit")); // NOI18N
jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
jMenuItem1.setName("jMenuItem1"); // NOI18N
jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE, 798,
Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE, 608,
javax.swing.GroupLayout.PREFERRED_SIZE)
);

if(modo.equals("Manual"))
{
    btnInvertir.setVisible(false);
    btnSiguiente.setEnabled(true);
}
else if (modo.equals("Archivo"))
{
    pnlCodigo.setVisible(false);
    btnInvertir.setVisible(false);
    btnSiguiente.setEnabled(true);
    btnAgregar.setVisible(false);
}
pack();
}

public static void main(String args[],TableModel estratos,TableModel estratosInv,LinkedList datos)

```

```

{
    final TableModel codigos=estratos;
    final TableModel codigosInv=estratosInv;
    final String modo=args[0];
    final String nombre=args[1];
    final String var=args[2];
    final LinkedList numeros=datos;
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Codificacion(codigos,modo,codigosInv,nombre,numeros,var).setVisible(true);
        }
    });
}
public static void main(String args[],TableModel presentacion,TableModel estratos,LinkedList
datos,LinkedList[] tMs)
{
    final TableModel codigos=estratos;
    final TableModel p=presentacion;
    final String modo=args[0];
    final String nombre=args[1];
    final String var=args[2];
    final String muestra_tomada=args[4];
    final int id=Integer.parseInt(args[3]);
    final LinkedList numeros=datos;
    final LinkedList[] tMS=tMs;
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new
Codificacion(p,codigos,modo,nombre,numeros,var,id,tMS,muestra_tomada).setVisible(true);
        }
    });
}


```

```

        tipo="Con Reemplazo";
    }
    else
    switch(comboAlgoritmo.getSelectedIndex())
    {
        case 0: reemplazo=0; //algoritmo xcl
            tipo="Algoritmo XCL";
            break;
        case 1: reemplazo=1; //algoritmo 1
            tipo="Algoritmo 1";
            break;
        case 2: reemplazo=2; // algoritmo 2
            tipo="Algoritmo 2";
            break;
        case 3: reemplazo=3; //algoritmo 3
            tipo="Algoritmo 3";
            break;
        case 4: reemplazo=4; //algoritmo 4
            tipo="Algoritmo 4";
            break;
        default:
            break;
    }
    Resultados resultados=new Resultados();
    LinkedList resumen=resultados.obtener_valores(datos, variable, "Numérico");
    LinkedList nt=resultados.nombres_tablas(0); //resumen población
    TablaResPob tabla_resumen=new TablaResPob(nt,resumen);
    int modo=0;
    String mode=lblModo.getText().toString();
    String argumentosMain[]={nombreArchivo,mode,tipo,variable,muestras_t};
    //nombre del archivo, si fue manual,archivo,promedio,con o sin reemplazo,tipo de algo y
    nombre variable
    if(mode.equals("Promedio"))
        modo=1;
    else
        if(mode.equals("Manual"))
            modo=2;
        else
            modo=3;//archivo
    switch(modo)
    {
        case 1: //el usuario seleccionó promedio

```



```

LinkedList[] valoresSeparados=new LinkedList[2];
LinkedList ids=resultados.codificacion_promedio(datos, modo);
valoresSeparados=resultados.separar_codigos(tblDatos.getModel(),ids);
String tamMuestra1=JOptionPane.showInputDialog("Indica el Tamaño de la Muestra
para el Estrato 1\nTamaño Población: "+valoresSeparados[0].size()).trim();
if(Integer.parseInt(tamMuestra1)>valoresSeparados[0].size())
{ JOptionPane.showMessageDialog(null,"El Tamaño de la Muestra no puede ser mayor a
"+valoresSeparados[0].size());
break;
}
String tamMuestra2=JOptionPane.showInputDialog("Indica el Tamaño de la Muestra
para el Estrato 2\nTamaño Población: "+valoresSeparados[1].size()).trim();
if(Integer.parseInt(tamMuestra2)>valoresSeparados[1].size())
{ JOptionPane.showMessageDialog(null,"El Tamaño de la Muestra no puede ser
mayor a "+valoresSeparados[1].size());
break;
}
LinkedList nombres_final=resultados.nombres_tablas(3);
TablaMuestra tabla_muestra1=resultados.aleatorio_simple("Código
1's",valoresSeparados[0].size(),valoresSeparados[0],tamMuestra1,reemplazo);
TablaMuestra tabla_muestra2=resultados.aleatorio_simple("Código
0's",valoresSeparados[1].size(),valoresSeparados[1],tamMuestra2,reemplazo);
LinkedList vfinal1=resultados.hacer_tablaFinal(tabla_muestra1);
LinkedList vfinal2=resultados.hacer_tablaFinal(tabla_muestra2);
TablaFinal fin1=new TablaFinal(nombres_final,vfinal1);
TablaFinal fin2=new TablaFinal(nombres_final,vfinal2);
String a[]={nombreArchivo,tamMuestra1,tamMuestra2,tipo,modoCodif};

```

```

FinalEstratificado.main(a,tabla_muestra1.getRowCount(),tabla_muestra2.getRowCount(),tabla_m
uestra1,tabla_muestra2,fin1,fin2,null);
break;

```

```

case 2: //el usuario seleccionó manual
//politápico
if(numeroEstratas==500)
{
boolean indi=false;
LinkedList[] d=new LinkedList[3];
LinkedList da=new LinkedList();
LinkedList codigos_etapa1=new LinkedList();
LinkedList codigos_etapa2=new LinkedList();

```

```

LinkedList aux1=new LinkedList();
LinkedList aux2=new LinkedList();
//inicia for para llenar las listas con los códigos del politápico
for(int i=0;i<tblDatos.getRowCount();i++)
{
    if(tblDatos.getValueAt(i, 1)==null || tblDatos.getValueAt(i,2)==null)
    {
        JOptionPane.showMessageDialog(null,"No deje espacios en blanco porfavor...");
        indi=true;
        break;
    }
    else
    {
        da.add(tblDatos.getValueAt(i,0));
        codigos_etapa1.add(tblDatos.getValueAt(i, 1));
        aux1.add(tblDatos.getValueAt(i, 1));
        codigos_etapa2.add(tblDatos.getValueAt(i,2));
        aux2.add(tblDatos.getValueAt(i,2));
    }
}
//fin del for que llena las listas de códigos del politápico
if(!indi)
{

    LinkedList tokens_codigo1=resultados.contar_estratos(codigos_etapa1);
    LinkedList tokens_codigo2=resultados.contar_estratos(codigos_etapa2);
    LinkedList names=new LinkedList();
    LinkedList[] tokens=new LinkedList[2];
    String[] ar={"",""};
    names.add(variable);
    names.add("Etapa 1");
    names.add("Etapa 2");
    d[0]=da;
    d[1]=aux1;
    d[2]=aux2;
    TablaLimites ti=new TablaLimites(names,d);
    tokens[0]=tokens_codigo1;
    tokens[1]=tokens_codigo2;
    Politapico.main(ar, ti,tokens);
    break;
}
else
    break;

```

```

}
else //estratificado
{
    //lista ligada de los codigos y tokens
    LinkedList codigos=new LinkedList();
    LinkedList tamMuestras=new LinkedList();
    int correcto=-1;
    boolean indi=false;
    LinkedList qqq=new LinkedList(); //contiene el tamaño de la poblacion de cada estrato
    for(int i=0;i<tblDatos.getRowCount();i++)
    {
        if(tblDatos.getValueAt(i, 1)==null)
        {
            JOptionPane.showMessageDialog(null,"No deje espacios en blanco porfavor...");
            correcto=4;
            indi=true;
            break;
        }
        else
            codigos.add(tblDatos.getValueAt(i, 1));
    }
    if(!indi)
    {
        LinkedList tokens=resultados.contar_estratos(codigos);
        correcto=JOptionPane.showConfirmDialog(null, "Usted Introdujo "+tokens.size()+"
Estratos, es correcto?");
        if(correcto==0)
        {
            for(int i=0;i<tblDatos.getRowCount();i++)
                codigos.add(tblDatos.getValueAt(i, 1)); //se tuvo que repetir pq se perdía la
referencia
            TableModel rr=resultados.codificacion_archivo(datos,
codigos,tokens,tblDatos.getColumnName(0).toString());
            qqq=resultados.tamPob_estrato(rr);//*****
            for(int j=0;j<qqq.size();j++)
                tamMuestras.add(JOptionPane.showInputDialog("Indica el tamaño de la Muestra
para el Estrato "+j
                +"\nTamaño de la Población: "+qqq.get(j)));
            TablaMuestra[] tm=new TablaMuestra[qqq.size()]; //TABLAS CON EL MUESTREO DE
CADA ESTRATO
            TablaFinal[] fl=new TablaFinal[qqq.size()];//TABLAS CON LOS RESULTADOS DEL
MUESTREO

```

```

LinkedList nfl=resultados.nombres_tablas(3); //tabla final
LinkedList[] calculos=new LinkedList[qqq.size()];
int acum=0;
for(int y=0;y<qqq.size();y++)
{
    if(y==0)
        acum=0;
    else
        acum=acum+Integer.parseInt(qqq.get(y-1).toString());
    tm[y]=resultados.aleatorio_simple("Estrato "+y,
Integer.parseInt(qqq.get(y).toString()),resultados.lista_estratos(rr,qqq,y,acum),tamMuestras.get(y
).toString(), reemplazo);
        calculos[y]=resultados.hacer_tablaFinal(tm[y]);
        fl[y]=new TablaFinal(nfl,calculos[y]);
    }
    TableModel
tabl_muestra=resultados.mergeTables(tm,datos.size(),qqq.size(),tamMuestras);
    TableModel tabl_fin=resultados.mergeTables(fl,qqq.size());
    //llamamos a la vista de resultados
    FinalEstrata.main(argumentosMain, tabl_muestra,
tabl_fin,tabla_resumen,tamMuestras);
    break;
}
else
    if(correcto==1)
    {
        JOptionPane.showMessageDialog(null,"Codifique de nuevo....");
        break;
    }
    else
        break;
}
else
    break;
}

```

case 3: //el usuario seleccionó archivo

```

LinkedList qqqq=resultados.tamPob_estrato(tblDatos.getModel());
LinkedList nf=resultados.nombres_tablas(3); //tabla final
LinkedList tmues=tMs[0];
/*for(int j=0;j<qqqq.size();j++)

```

```

        tmues.add(JOptionPane.showInputDialog("Indica el tamaño de la Muestra para el
Estrato "+j
        +"\nTamaño de la Población: "+qqqq.get(j));*/
        TablaMuestra[] tmArch=new TablaMuestra[qqqq.size()]; //TABLAS CON EL MUESTREO
DE CADA ESTRATO
        TablaFinal[] tfn=new TablaFinal[qqqq.size()]; //TABLAS CON LOS CALCULOS DE LAS
MUESTRAS
        LinkedList[] tff=new LinkedList[qqqq.size()]; //Lista de resultados para la tabla final
        int acumulado=0;
        for(int y=0;y<qqqq.size();y++)
        {
            if(y==0)
                acumulado=0;
            else
                acumulado=acumulado+Integer.parseInt(qqqq.get(y-1).toString());
            tmArch[y]=resultados.aleatorio_simple("Estrato "+y,
Integer.parseInt(qqqq.get(y).toString()),resultados.lista_estratos(tblDatos.getModel(),qqqq,y,acu
mulado),tmues.get(y).toString(),reemplazo);
            tff[y]=resultados.hacer_tablaFinal(tmArch[y]);
            tfn[y]=new TablaFinal(nf,tff[y]);
        }
        TableModel
tabla_muestra=resultados.mergeTables(tmArch,datos.size(),qqqq.size(),tmues);
        TablaEstratos tabla_fin=resultados.mergeTables(tfn,qqqq.size());
        //llamamos a la vista de resultados
        tblDatos.setModel(presentacion);
        FinalEstrata.main(argumentosMain, tabla_muestra, tabla_fin,tabla_resumen,tmues);
        break;
    }
}

// Declaración de variables
private javax.swing.JButton btnAgregar;
private javax.swing.JButton btnAyuda;
private javax.swing.JButton btnInvertir;
private javax.swing.JButton btnSeleccionar;
private javax.swing.JButton btnSiguiente;
private javax.swing.JComboBox comboAlgoritmo;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;

```

```

private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JComboBox jcbCodigo;
private javax.swing.JScrollPane jsDatos;
private javax.swing.JLabel lblElija;
private javax.swing.JLabel lblHeader;
private javax.swing.JLabel lblIndique;
private javax.swing.JLabel lblModo;
private javax.swing.JPanel mainPanel;
private javax.swing.JPanel pnlCodigo;
private javax.swing.JPanel pnlElija;
private javax.swing.JRadioButton rbtnConReemplazo;
private javax.swing.JRadioButton rbtnSinReemplazo;
private javax.swing.JTable tblDatos;
}

```

### 9.2.6 Estratificado.java

```

/*
 * Esta es la interfaz gráfica del muestreo estratificado
 * Created on 26/09/2010, 09:32:47 PM
 */

```

```

package pt02.vistas;
import java.util.LinkedList;
import javax.swing.JComboBox;
import javax.swing.JOptionPane;
import javax.swing.table.TableModel;
import org.jdesktop.application.Action;
import pt02.archivos.Archivos;
import pt02.kernel.Muestreo;
import pt02.kernel.Resultados;

public class Estratificado extends javax.swing.JFrame {
    public String nombreArchivo;

    /** Crea un Nuevo formulario Estratificado */
    public Estratificado()
    {
        initComponents();
        pnlSelecVar.setVisible(false);
        pnlArchivo.setVisible(false);
    }
}

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    mainPanel = new javax.swing.JPanel();
    pnlSelecVar = new javax.swing.JPanel();
    comboVariables = new javax.swing.JComboBox();
    lblSelVar = new javax.swing.JLabel();
    pnlSelEstrato = new javax.swing.JPanel();
    lblCuantosEstratos = new javax.swing.JLabel();
    rbtnManual = new javax.swing.JRadioButton();
    rbtnPromedio = new javax.swing.JRadioButton();
    rbtnArchivo = new javax.swing.JRadioButton();
    btnCargar = new javax.swing.JButton();
    pnlArchivo = new javax.swing.JPanel();
    comboColumna = new javax.swing.JComboBox();
    lblSelecciona = new javax.swing.JLabel();
    lblSelecciona2 = new javax.swing.JLabel();
    btnVolver = new javax.swing.JButton();
    btnSiguiete = new javax.swing.JButton();
    lblHeader = new javax.swing.JLabel();
    lblEstrella2 = new javax.swing.JLabel();
    jsDatos = new javax.swing.JScrollPane();
    tblDatos = new javax.swing.JTable();
    btnAyuda = new javax.swing.JButton();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenuItem2 = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(Estratificado.class);
    setTitle(resourceMap.getString("Form.title")); // NOI18N
    setAlwaysOnTop(true);
    setName("Form"); // NOI18N
    setResizable(false);

    mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
    mainPanel.setName("mainPanel"); // NOI18N

```

```

pnISelecVar.setBackground(resourceMap.getColor("pnISelecVar.background")); // NOI18N

pnISelecVar.setBorder(javax.swing.BorderFactory.createLineBorder(resourceMap.getColor("pnISelecVar.border.lineColor"))); // NOI18N
pnISelecVar.setName("pnISelecVar"); // NOI18N

comboVariables.setFont(resourceMap.getFont("comboVariables.font")); // NOI18N
comboVariables.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "-", "-", "-"
}));
comboVariables.setName("comboVariables"); // NOI18N

lblSelVar.setFont(resourceMap.getFont("lblSelVar.font")); // NOI18N
lblSelVar.setForeground(resourceMap.getColor("lblSelVar.foreground")); // NOI18N
lblSelVar.setText(resourceMap.getString("lblSelVar.text")); // NOI18N
lblSelVar.setName("lblSelVar"); // NOI18N

javax.swing.GroupLayout pnISelecVarLayout = new javax.swing.GroupLayout(pnISelecVar);
pnISelecVar.setLayout(pnISelecVarLayout);
pnISelecVarLayout.setHorizontalGroup(
    pnISelecVarLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnISelecVarLayout.createSequentialGroup()
            .addGroup(pnISelecVarLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnISelecVarLayout.createSequentialGroup()
                    .addComponent(lblSelVar)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(comboVariables, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(pnISelecVarLayout.createSequentialGroup()
                    .addGap(18, Short.MAX_VALUE)
                    .addGap(18, Short.MAX_VALUE)))
            .addContainerGap())
);
pnISelecVarLayout.setVerticalGroup(
    pnISelecVarLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnISelecVarLayout.createSequentialGroup()
            .addGroup(pnISelecVarLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnISelecVarLayout.createSequentialGroup()
                    .addComponent(lblSelVar)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(comboVariables, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(pnISelecVarLayout.createSequentialGroup()
                    .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
            .addContainerGap())
);

pnISelestrato.setBackground(resourceMap.getColor("pnISelestrato.background")); // NOI18N

```



```
pnlSelEstrato.setBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("pnlSelEstrato.border.highlightColor"), null)); // NOI18N
pnlSelEstrato.setName("pnlSelEstrato"); // NOI18N
```

```
lblCuantosEstratos.setFont(resourceMap.getFont("lbl.font")); // NOI18N
lblCuantosEstratos.setForeground(resourceMap.getColor("lbl.foreground")); // NOI18N
lblCuantosEstratos.setText(resourceMap.getString("lbl.text")); // NOI18N
lblCuantosEstratos.setName("lbl"); // NOI18N
```

```
rbtnManual.setBackground(resourceMap.getColor("rbtnManual.background")); // NOI18N
rbtnManual.setFont(resourceMap.getFont("rbtnManual.font")); // NOI18N
rbtnManual.setForeground(resourceMap.getColor("rbtnManual.foreground")); // NOI18N
rbtnManual.setSelected(true);
rbtnManual.setText(resourceMap.getString("rbtnManual.text")); // NOI18N
rbtnManual.setName("rbtnManual"); // NOI18N
rbtnManual.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnManualMouseClicked(evt);
    }
});
```

```
rbtnPromedio.setBackground(resourceMap.getColor("rbtnPromedio.background"));
rbtnPromedio.setFont(resourceMap.getFont("rbtnPromedio.font")); // NOI18N
rbtnPromedio.setForeground(resourceMap.getColor("rbtnPromedio.foreground")); // NOI18N
rbtnPromedio.setText(resourceMap.getString("rbtnPromedio.text")); // NOI18N
rbtnPromedio.setName("rbtnPromedio"); // NOI18N
rbtnPromedio.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnPromedioMouseClicked(evt);
    }
});
```

```
rbtnArchivo.setBackground(resourceMap.getColor("rbtnArchivo.background")); // NOI18N
rbtnArchivo.setFont(resourceMap.getFont("rbtnArchivo.font")); // NOI18N
rbtnArchivo.setForeground(resourceMap.getColor("rbtnArchivo.foreground")); // NOI18N
rbtnArchivo.setText(resourceMap.getString("rbtnArchivo.text")); // NOI18N
rbtnArchivo.setName("rbtnArchivo"); // NOI18N
rbtnArchivo.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        rbtnArchivoMouseClicked(evt);
    }
});
```



```

        .addComponent(comboColumna, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap()
    );
    pnlArchivoLayout.setVerticalGroup(
        pnlArchivoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlArchivoLayout.createSequentialGroup())
        .addComponent(lblSelecciona)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 10,
Short.MAX_VALUE)
        .addComponent(lblSelecciona2))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlArchivoLayout.createSequentialGroup())
        .addContainerGap(15, Short.MAX_VALUE)
        .addComponent(comboColumna, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap()
    );

    javax.swing.GroupLayout pnlSelEstratoLayout = new javax.swing.GroupLayout(pnlSelEstrato);
    pnlSelEstrato.setLayout(pnlSelEstratoLayout);
    pnlSelEstratoLayout.setHorizontalGroup(
        pnlSelEstratoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSelEstratoLayout.createSequentialGroup())
        .addContainerGap()

        .addGroup(pnlSelEstratoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
        )
            .addGroup(pnlSelEstratoLayout.createSequentialGroup())

        .addGroup(pnlSelEstratoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
        )
            .addComponent(rbbtnManual)
            .addComponent(lblCuantosEstratos)
            .addComponent(rbbtnArchivo)
            .addGroup(pnlSelEstratoLayout.createSequentialGroup())
                .addComponent(rbbtnPromedio)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 47,
Short.MAX_VALUE)
                .addComponent(btnCargar)))
            .addContainerGap(25, Short.MAX_VALUE))

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlSelEstratoLayout.createSequentialGroup()
        .addComponent(pnlArchivo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap()))
);
pnlSelEstratoLayout.setVerticalGroup(
pnlSelEstratoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(pnlSelEstratoLayout.createSequentialGroup()
.addContainerGap()
.addComponent(lblCuantosEstratos)
.addGap(18, 18, 18)
.addComponent(rbtnManual)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(pnlSelEstratoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(rbtnPromedio)
.addComponent(btnCargar)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(rbtnArchivo)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addComponent(pnlArchivo, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addContainerGap())
);

btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
btnVolver.setName("btnVolver"); // NOI18N
btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnVolverActionPerformed(evt);
    }
});

btnSiguiente.setFont(resourceMap.getFont("btnSiguiente.font")); // NOI18N
btnSiguiente.setForeground(resourceMap.getColor("btnSiguiente.foreground")); // NOI18N
btnSiguiente.setEnabled(false);
btnSiguiente.setLabel(resourceMap.getString("btnSiguiente.label")); // NOI18N
btnSiguiente.setName("btnSiguiente"); // NOI18N

```

```

btnSiguiente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        siguiente(evt);
    }
});

lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
lblHeader.setName("lblHeader"); // NOI18N

lblEstrella2.setIcon(resourceMap.getIcon("lblEstrella2.icon")); // NOI18N
lblEstrella2.setText(resourceMap.getString("lblEstrella2.text")); // NOI18N
lblEstrella2.setName("lblEstrella2"); // NOI18N

jsDatos.setBackground(resourceMap.getColor("jsDatos.background")); // NOI18N

jsDatos.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.create
EtchedBorder(resourceMap.getColor("jsDatos.border.border.highlightColor"),
resourceMap.getColor("jsDatos.border.border.shadowColor")),
resourceMap.getString("jsDatos.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsDatos.border.titleFont"),
resourceMap.getColor("jsDatos.border.titleColor"))); // NOI18N
jsDatos.setName("jsDatos"); // NOI18N

tblDatos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
tblDatos.setName("tblDatos"); // NOI18N
jsDatos.setViewportView(tblDatos);

```



```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
        .addComponent(pnlSelEstrato, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(31, 31, 31)
        .addComponent(btnAyuda, javax.swing.GroupLayout.PREFERRED_SIZE, 57,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGap(103, 103, 103))
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 369,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap()
    );
    mainPanelLayout.setVerticalGroup(
        mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addComponent(lblHeader)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(mainPanelLayout.createSequentialGroup()
                    .addGap(54, 54, 54)
                    .addComponent(btnAyuda))
                .addGroup(mainPanelLayout.createSequentialGroup()
                    .addGap(32, 32, 32)
                    .addComponent(pnlSelEstrato, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 87,
Short.MAX_VALUE)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
                    .addComponent(pnlSelecVar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(90, 90, 90))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
                    .addComponent(lblEstrella2)

```

```

        .addGap(71, 71, 71)))
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 477,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnVolver)
    .addComponent(btnSiguiente))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

jMenuBar1.setName("jMenuBar1"); // NOI18N

jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

jMenuItem1.setAction(actionMap.get("quit")); // NOI18N
jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
jMenuItem1.setName("jMenuItem1"); // NOI18N
jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE, 800,
javax.swing.GroupLayout.PREFERRED_SIZE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
);

```



```

);

pack();
} // </editor-fold>

private void CargarArchivo(java.awt.event.ActionEvent evt) {
    Archivos inicio=new Archivos();
    nombreArchivo=inicio.manejar_archivo(tblDatos, comboVariables,comboColumna);
    pnlSelecVar.setVisible(true);
    btnSiguiente.setEnabled(true);
    if(rbtnArchivo.isSelected())
        pnlArchivo.setVisible(true);
}

private void siguiente(java.awt.event.ActionEvent evt) {
    //INICIAN VALIDACIONES
    String estrata=null;
    Resultados result=new Resultados();
    LinkedList a=new LinkedList();
    if(rbtnManual.isSelected())
        estrata="Manual";
    else
        if(rbtnPromedio.isSelected())
            estrata="Promedio";
        else
            estrata="Archivo";

    a=result.cuenta_estratos(tblDatos.getModel(),comboColumna.getSelectedIndex()); //lista
de tokens estratos
    Object nestrata=a.size();

    if(estrata.equals("Archivo"))
        JOptionPane.showMessageDialog(null, "Seleccionó la Variable: " +
comboVariables.getSelectedItem().toString()
        + "\nElección de Estratos: "+estrata+"\nNúmero de Estratos: "+nestrata.toString()+
        "\nCódigo: "+comboColumna.getSelectedItem().toString());
    else
        JOptionPane.showMessageDialog(null, "Seleccionó la Variable: " +
comboVariables.getSelectedItem().toString()
        + "\nElección de Estratos: "+estrata);
}

```

```

String[] argumentos={nombreArchivo,estrata,nestrata.toString()};
if(estrata.equals("Promedio") || estrata.equals("Manual"))
{
    Muestreo muestreo=new
Muestreo(comboVariables,tblDatos.getModel(),comboVariables.getSelectedItem().toString(),estra
ta,comboColumna,a,null);
    muestreo.empezar_metodo(2,nombreArchivo, estrata); //2 es para estratificado
}
else
{
    Resultados estratos=new Resultados();
    LinkedList data=estratos.lista_numeros(tblDatos.getModel(),
comboVariables.getSelectedIndex());

TomaMuestras.main(argumentos,comboVariables,tblDatos.getModel(),comboColumna,data);
}
}

private void rbtnManualMouseClicked(java.awt.event.MouseEvent evt) {
    if(rbtnManual.isSelected())
    {
        rbtnPromedio.setSelected(false);
        rbtnArchivo.setSelected(false);
        pnlArchivo.setVisible(false);
    }
}

private void rbtnPromedioMouseClicked(java.awt.event.MouseEvent evt) {
    if(rbtnPromedio.isSelected())
    {
        rbtnManual.setSelected(false);
        rbtnArchivo.setSelected(false);
        pnlArchivo.setVisible(false);
    }
}

private void btnVolverActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
}

private void ayuda(java.awt.event.ActionEvent evt) {

```

```

// TODO add your handling code here:
}

private void rbtnArchivoMouseClicked(java.awt.event.MouseEvent evt) {
    if(rbtnArchivo.isSelected())
    {
        rbtnManual.setSelected(false);
        rbtnPromedio.setSelected(false);
    }
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Estratificado().setVisible(true);
        }
    });
}

@Action
public void cargar_ayuda() {
    String hola[]={"C:\\Users\\USUARIO\\Documents\\h.txt",this.getName()};
    //String hola[]={"D:\\Documents and Settings\\xcerna\\Mis
Documentos\\vals.txt",this.getName()};
    //String hola[]={"C:\\Documents and Settings\\PATRICIA\\Mis
Documentos\\prueba.txt",this.getName()};
    Ayuda.main(hola);
}

// Declaración de variables
private javax.swing.JButton btnAyuda;
private javax.swing.JButton btnCargar;
private javax.swing.JButton btnSiguiente;
private javax.swing.JButton btnVolver;
private javax.swing.JComboBox comboColumna;
private javax.swing.JComboBox comboVariables;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JScrollPane jsDatos;
private javax.swing.JLabel lblCuantosEstratos;

```

```

private javax.swing.JLabel lblHeader;
private javax.swing.JLabel lblSelVar;
private javax.swing.JLabel lblSelecciona;
private javax.swing.JLabel lblSelecciona2;
private javax.swing.JLabel lblEstrella2;
private javax.swing.JPanel mainPanel;
private javax.swing.JPanel pnlArchivo;
private javax.swing.JPanel pnlSelEstrato;
private javax.swing.JPanel pnlSelecVar;
private javax.swing.JRadioButton rbtnArchivo;
private javax.swing.JRadioButton rbtnManual;
private javax.swing.JRadioButton rbtnPromedio;
private javax.swing.JTable tblDatos;
}

```

### 9.2.7 TomaMuestras.java

```

/*
 * TomaMuestras.java
 *
 * Created on 1/02/2011, 10:40:12 PM
 */
package pt02.vistas;
import java.util.LinkedList;
import javax.swing.JOptionPane;
import javax.swing.table.TableModel;
import pt02.kernel.Muestreo;
import pt02.kernel.Resultados;
/**
 *
 * @author Xime!!!!
 */
public class TomaMuestras extends javax.swing.JFrame
{
public javax.swing.JComboBox vari;
public javax.swing.JComboBox colu;
public String nombreA;
public String mod;
public String nums;
public TableModel dats;
public LinkedList dT;

```

```

public TomaMuestras() {
    initComponents();
}
public TomaMuestras(javax.swing.JComboBox var,TableModel dat,javax.swing.JComboBox
col,String nomA,String modo,String numE,LinkedList d) {
    this.nombreA=nomA;
    this.mod=modo;
    this.nums=numE;
    this.dats=dat;
    this.dT=d;
    this.vari=var;
    this.colu=col;
    initComponents();
    lblreparte.setText("Introduzca la Repartición de Estratos (Tamaño Población: "+ dT.size()+")");
    pnlToma.setVisible(false);
    tblTamMuestras.setVisible(false);
    btnSiguiete.setVisible(false);

}
@SuppressWarnings("unchecked")
private void initComponents() {

    mainPanel = new javax.swing.JPanel();
    lblHeader = new javax.swing.JLabel();
    jsMuestras = new javax.swing.JScrollPane();
    tblTamMuestras = new javax.swing.JTable();
    pnlToma = new javax.swing.JPanel();
    lblTam = new javax.swing.JLabel();
    btnAyuda = new javax.swing.JButton();
    comboMuestras = new javax.swing.JComboBox();
    lblfooter = new javax.swing.JLabel();
    btnVolver = new javax.swing.JButton();
    btnSiguiete = new javax.swing.JButton();
    lblstrella = new javax.swing.JLabel();
    txtReparte = new javax.swing.JTextField();
    lblreparte = new javax.swing.JLabel();
    btnOK = new javax.swing.JButton();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenuItem2 = new javax.swing.JMenuItem();
}

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(TomaMuestras.class);
setTitle(resourceMap.getString("Form.title")); // NOI18N
setName("Form"); // NOI18N

mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
mainPanel.setName("mainPanel"); // NOI18N

lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
lblHeader.setName("lblHeader"); // NOI18N

jsMuestras.setBackground(resourceMap.getColor("jsMuestras.background")); // NOI18N

jsMuestras.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cr
eateEtchedBorder(resourceMap.getColor("jsMuestras.border.border.highlightColor"),
resourceMap.getColor("jsMuestras.border.border.shadowColor")),
resourceMap.getString("jsMuestras.border.title"), javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsMuestras.border.titleFont"),
resourceMap.getColor("jsMuestras.border.titleColor"))); // NOI18N
jsMuestras.setName("jsMuestras"); // NOI18N

tblTamMuestras.setFont(resourceMap.getFont("tblTamMuestras.font")); // NOI18N
tblTamMuestras.setForeground(resourceMap.getColor("tblTamMuestras.foreground"));
tblTamMuestras.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Tamaño", "Manual", "Proporcional", "Óptima"
    }
));
tblTamMuestras.setGridColor(resourceMap.getColor("tblTamMuestras.gridColor"));
tblTamMuestras.setName("tblTamMuestras"); // NOI18N

```

```
tblTamMuestras.setSelectionBackground(resourceMap.getColor("tblTamMuestras.selectionBackground")); // NOI18N
```

```
tblTamMuestras.setSelectionForeground(resourceMap.getColor("tblTamMuestras.selectionForeground")); // NOI18N
```

```
    jsMuestras.setViewPortView(tblTamMuestras);
```

```
tblTamMuestras.getColumnModel().getColumn(0).setHeaderValue(resourceMap.getString("tblTamMuestras.columnModel.title0")); // NOI18N
```

```
tblTamMuestras.getColumnModel().getColumn(1).setHeaderValue(resourceMap.getString("tblTamMuestras.columnModel.title1")); // NOI18N
```

```
tblTamMuestras.getColumnModel().getColumn(2).setHeaderValue(resourceMap.getString("tblTamMuestras.columnModel.title2")); // NOI18N
```

```
    pnlToma.setBackground(resourceMap.getColor("pnlToma.background")); // NOI18N
```

```
pnlToma.setBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("pnlToma.border.highlightColor"), resourceMap.getColor("pnlToma.border.shadowColor"))); // NOI18N
```

```
    pnlToma.setName("pnlToma"); // NOI18N
```

```
    lblTam.setFont(resourceMap.getFont("lblTam.font")); // NOI18N
```

```
    lblTam.setForeground(resourceMap.getColor("lblTam.foreground")); // NOI18N
```

```
    lblTam.setText(resourceMap.getString("lblTam.text")); // NOI18N
```

```
    lblTam.setName("lblTam"); // NOI18N
```

```
    btnAyuda.setIcon(resourceMap.getIcon("btnAyuda.icon")); // NOI18N
```

```
    btnAyuda.setText(resourceMap.getString("btnAyuda.text")); // NOI18N
```

```
    btnAyuda.setName("btnAyuda"); // NOI18N
```

```
    btnAyuda.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            ayuda(evt);
```

```
        }
```

```
    });
```

```
    comboMuestras.setFont(resourceMap.getFont("comboMuestras.font")); // NOI18N
```

```
    comboMuestras.setForeground(resourceMap.getColor("comboMuestras.foreground"));
```

```
    comboMuestras.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Manual", "Proporcional", "Costo", "Varianza" }));
```

```
    comboMuestras.setName("comboMuestras"); // NOI18N
```





```

lblfooter.setName("lblfooter"); // NOI18N

btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
btnVolver.setName("btnVolver"); // NOI18N
btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        volver(evt);
    }
});

btnSiguiente.setFont(resourceMap.getFont("btnSiguiente.font")); // NOI18N
btnSiguiente.setForeground(resourceMap.getColor("btnSiguiente.foreground")); // NOI18N
btnSiguiente.setText(resourceMap.getString("btnSiguiente.text")); // NOI18N
btnSiguiente.setName("btnSiguiente"); // NOI18N
btnSiguiente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        correr(evt);
    }
});

lblestrella.setIcon(resourceMap.getIcon("lblestrella.icon")); // NOI18N
lblestrella.setText(resourceMap.getString("lblestrella.text")); // NOI18N
lblestrella.setName("lblestrella"); // NOI18N

txtReparte.setFont(resourceMap.getFont("txtReparte.font")); // NOI18N
txtReparte.setForeground(resourceMap.getColor("txtReparte.foreground")); // NOI18N
txtReparte.setText(resourceMap.getString("txtReparte.text")); // NOI18N
txtReparte.setName("txtReparte"); // NOI18N

lblreparte.setFont(resourceMap.getFont("lblreparte.font")); // NOI18N
lblreparte.setForeground(resourceMap.getColor("lblreparte.foreground")); // NOI18N
lblreparte.setText(resourceMap.getString("lblreparte.text")); // NOI18N
lblreparte.setName("lblreparte"); // NOI18N

btnOK.setFont(resourceMap.getFont("btnOK.font")); // NOI18N
btnOK.setForeground(resourceMap.getColor("btnOK.foreground")); // NOI18N
btnOK.setText(resourceMap.getString("btnOK.text")); // NOI18N
btnOK.setName("btnOK"); // NOI18N
btnOK.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        cambiar_tabla(evt);
    }
});

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblHeader, javax.swing.GroupLayout.PREFERRED_SIZE, 780,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                .addGroup(mainPanelLayout.createSequentialGroup()

                    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(mainPanelLayout.createSequentialGroup()
                            .addComponent(lblreparte)
                            .addGroup(mainPanelLayout.createSequentialGroup()
                                .addComponent(txtReparte, javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(30, 30, 30)
                                .addComponent(btnOK))
                            .addComponent(jsMuestras, javax.swing.GroupLayout.PREFERRED_SIZE, 512,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGroup(mainPanelLayout.createSequentialGroup()
                            .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(mainPanelLayout.createSequentialGroup()
                                    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(mainPanelLayout.createSequentialGroup()
                                            .addGap(62, 62, 62)
                                            .addComponent(lblestrella))
                                        .addGroup(mainPanelLayout.createSequentialGroup()
                                            .addGap(18, 18, 18)
                                            .addComponent(pnlToma, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
                                    .addGap(196, 196, 196))
                                .addGroup(mainPanelLayout.createSequentialGroup()
                                    .addComponent(btnVolver)
                                    .addGap(82, 82, 82)
                                    .addComponent(lblfooter)

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btnSiguiente)
        .addGap(182, 182, 182))))
);
mainPanelLayout.setVerticalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addComponent(lblHeader)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addGap(30, 30, 30)
        .addComponent(pnlToma, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(127, 127, 127)
        .addComponent(lblestrella))
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(jsMuestras, javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(lblreparte)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(txtReparte, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(btnOK))))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 69,
Short.MAX_VALUE)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(btnVolver)
    .addComponent(btnSiguiente))
    .addComponent(lblfooter)
    .addContainerGap()
);

jMenuBar1.setName("jMenuBar1"); // NOI18N

```

```

jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

    javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActionMap(TomaMuestras.class, this);
    jMenuItem1.setAction(actionMap.get("quit")); // NOI18N
    jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
    jMenuItem1.setName("jMenuItem1"); // NOI18N
    jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE, 780,
javax.swing.GroupLayout.PREFERRED_SIZE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
pack();
}

private void volver(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void correr(java.awt.event.ActionEvent evt) {

    LinkedList[] tam_muestras=new LinkedList[4];

```

```

LinkedList aux=new LinkedList();
LinkedList codigos=new LinkedList();
int indice=comboMuestras.getSelectedIndex()+3;
//si escogió manual, checar que hay distribuido bien y que no haya celdas vacías
if(comboMuestras.getSelectedIndex()==0)
{
    int suma=0;
    for(int j=0;j<tblTamMuestras.getRowCount();j++)
    {
        suma=suma+Integer.parseInt(tblTamMuestras.getValueAt(j,3).toString());
    }
    if (suma!=Integer.parseInt(txtReparte.getText()))
        JOptionPane.showMessageDialog(null,"La suma de los estratos no es
"+txtReparte.getText()+", porfavor distribuya de nuevo....");
    else
    {
        for(int i=0;i<tblTamMuestras.getRowCount();i++)
        {
            aux.add(tblTamMuestras.getValueAt(i, indice));
            codigos.add(tblTamMuestras.getValueAt(i, 1));
        }
        tam_muestras[0]=aux;
        Resultados resultados=new Resultados();
        LinkedList tokens=resultados.contar_estratos(codigos);
        Muestreo estratificado=new Muestreo(vari,dats,nums,mod,colu,tokens,tam_muestras);
        estratificado.empezar_metodo(2,
nombreA,comboMuestras.getSelectedItem().toString());//*****
    }
}
else
{
    for(int i=0;i<tblTamMuestras.getRowCount();i++)
    {
        aux.add(tblTamMuestras.getValueAt(i, indice));
        codigos.add(tblTamMuestras.getValueAt(i, 1));
    }
    tam_muestras[0]=aux;
    Resultados resultados=new Resultados();
    LinkedList tokens=resultados.contar_estratos(codigos);
    Muestreo estratificado=new Muestreo(vari,dats,nums,mod,colu,tokens,tam_muestras);
    estratificado.empezar_metodo(2,
nombreA,comboMuestras.getSelectedItem().toString());//*****
}
}

```

```

    }
}

private void ayuda(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cambiar_tabla(java.awt.event.ActionEvent evt) {
    pnlToma.setVisible(true);
    tblTamMuestras.setVisible(true);
    btnSiguiete.setVisible(true);
    changeComponents(nums,dT,dats);
}

private void changeComponents(String num_estratos,LinkedList datos,TableModel tblDatos){
    String[] noms={"Estrato","Código","N","Manual","Proporcional","Costo","Varianza"};
    int n=Integer.parseInt(num_estratos);
    //PARA SABER EL TAMAÑO N DE CADA ESTRATO
    LinkedList codigos=new LinkedList();
    LinkedList copia=new LinkedList();
    LinkedList tokens=new LinkedList();
    LinkedList tam_muestras=new LinkedList();
    Resultados resultados=new Resultados();
    for(int i=0;i<tblDatos.getRowCount();i++)
        codigos.add(tblDatos.getValueAt(i, colu.getSelectedIndex()));
    tokens=resultados.contar_estratos(codigos);
    for(int i=0;i<tblDatos.getRowCount();i++)
        copia.add(tblDatos.getValueAt(i, colu.getSelectedIndex()));
    TableModel rr=resultados.codificacion_archivo(datos,
copia,tokens,tblDatos.getColumnName(0).toString());
    tam_muestras=resultados.tamPob_estrato(rr);
    //*****
    int tamPob=datos.size();
    int strato=Integer.parseInt(txtReparte.getText().trim());
    if(strato==0)
        JOptionPane.showMessageDialog(null,"Porfavor introduzca un valor para la repartición.....");
    else
        if(strato>tamPob)
            JOptionPane.showMessageDialog(null,"No debe de excederse de "+tamPob+"!!!");
        else
        {
            int num=0;
            double num2=0.0;

```

```

double mult=0;
LinkedList proporcional=new LinkedList();
for(int y=0;y<n;y++)
{
    num=Integer.parseInt(tam_muestras.get(y).toString());
    num2=(double)num/(double)tamPob;
    mult=num2*strato;
    mult=Math.round(mult);
    proporcional.add(mult);
}
for(int i=0;i<tblDatos.getRowCount();i++)
    codigos.add(tblDatos.getValueAt(i, colu.getSelectedIndex()));
tokens=resultados.contar_estratos(codigos);
Object[][]vals=new Object[n][7];
for(int i=0;i<n;i++)
{
    vals[i][0]=i+1; //INDICE DE LOS ESTRATOS
    vals[i][1]=tokens.get(i); //CÓDIGO DE CADA ESTRATO
    vals[i][2]=tam_muestras.get(i); //TAMAÑO DE CADA ESTRATO
    vals[i][4]=proporcional.get(i).toString().replace(".0", "").trim(); //TAMAÑO DE MUESTRAS
    PROPORCIONAL
}
TableModel tm=new javax.swing.table.DefaultTableModel(vals,noms);
tblTamMuestras.setModel(tm);
}
}
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new TomaMuestras().setVisible(true);
        }
    });
}
public static void main(String args[],javax.swing.JComboBox combo_variables,TableModel
datos,javax.swing.JComboBox columna,LinkedList data) {
    final javax.swing.JComboBox cv=combo_variables;
    final TableModel tblD=datos;
    final javax.swing.JComboBox c=columna;
    final String na=args[0];

```

```

    final String md=args[1];
    final String ne=args[2];
    final LinkedList dd=data;
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new TomaMuestras(cv,tblD,c,na,md,ne,dd).setVisible(true);
        }
    });
}
// Declaración de variables
private javax.swing.JButton btnAyuda;
private javax.swing.JButton btnOK;
private javax.swing.JButton btnSiguiente;
private javax.swing.JButton btnVolver;
private javax.swing.JComboBox comboMuestras;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JScrollPane jsMuestras;
private javax.swing.JLabel lblHeader;
private javax.swing.JLabel lblTam;
private javax.swing.JLabel lblEstrella;
private javax.swing.JLabel lblFooter;
private javax.swing.JLabel lblReparte;
private javax.swing.JPanel mainPanel;
private javax.swing.JPanel pnlToma;
private javax.swing.JTable tblTamMuestras;
private javax.swing.JTextField txtReparte;
}

```

### 9.2.8 Final.java

```

/*
 * Esta clase es la que tiene los resultados del muestreo y de los límites
 * Created on 21/09/2010, 11:44:55 AM
 */

package pt02.vistas;
import javax.swing.JOptionPane;
import org.jdesktop.application.Action;
import pt02.archivos.Archivos;

```



```

import pt02.modelosTablas.TablaFinal;
import pt02.modelosTablas.TablaMuestra;
import pt02.modelosTablas.TablaResPob;
/**
 *
 * @author XiMe!!
 */
public class Final extends javax.swing.JFrame
{
    public String nombre_metodo;
    public TablaMuestra tsample;
    public TablaFinal tresumen;
    public String statu;
    public String tMuestra;
    public String nomArch;
    public String tPoblacion;
    public Final(String nom_arch,String tamM,String tamP,String metodo,TablaResPob
tr,TablaMuestra tm,TablaFinal tf,String stat)
    {
        nombre_metodo=metodo;
        tsample=tm;
        tresumen=tf;
        statu=stat;
        tMuestra=tamM;
        nomArch=nom_arch;
        tPoblacion=tamP;
        initComponents(nom_arch,tamM,tamP,tr,tm,tf,stat);
    }
    @SuppressWarnings("unchecked")

    private void initComponents() {

        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
        jMenu2 = new javax.swing.JMenu();
        pnlPrincipal = new javax.swing.JPanel();
        tbpnlResultados = new javax.swing.JTabbedPane();
        pnlPoblacion = new javax.swing.JPanel();
        jsResumen = new javax.swing.JScrollPane();
        tblResumen = new javax.swing.JTable();
        pnlNombres = new javax.swing.JPanel();
        lblNomArch = new javax.swing.JLabel();

```

```
lbtamPob = new javax.swing.JLabel();
lbtamMuestra = new javax.swing.JLabel();
lbnomArch2 = new javax.swing.JLabel();
lbtamPob2 = new javax.swing.JLabel();
lbtamMuestra2 = new javax.swing.JLabel();
lbmonito = new javax.swing.JLabel();
lbfooter = new javax.swing.JLabel();
pnlmuestra = new javax.swing.JPanel();
jsmuestra = new javax.swing.JScrollPane();
tblmuestra = new javax.swing.JTable();
jslimites = new javax.swing.JScrollPane();
tblimites = new javax.swing.JTable();
lbrresultados = new javax.swing.JLabel();
lbcerdito = new javax.swing.JLabel();
lbfooterM = new javax.swing.JLabel();
btngurdar = new javax.swing.JButton();
lbflechaDer = new javax.swing.JLabel();
btnvolver = new javax.swing.JButton();
lbflechalzq = new javax.swing.JLabel();
lbheader = new javax.swing.JLabel();
mnuArriba = new javax.swing.JMenuBar();
smnuArchivo = new javax.swing.JMenu();
ssmnuSalir = new javax.swing.JMenuItem();
smnuAyuda = new javax.swing.JMenu();
```

```
jMenuBar1.setName("jMenuBar1"); // NOI18N
```

```
org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResourceMap(Final.class);
```

```
jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N
jMenuBar1.add(jMenu1);
```

```
jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle(resourceMap.getString("Form.title")); // NOI18N
setAlwaysOnTop(true);
setName("Form"); // NOI18N
```

```
pnlPrincipal.setBackground(resourceMap.getColor("pnlPrincipal.background")); // NOI18N
pnlPrincipal.setName("pnlPrincipal"); // NOI18N
```

```
tbpnlResultados.setForeground(resourceMap.getColor("tbpnlResultados.foreground"));
tbpnlResultados.setFont(resourceMap.getFont("tbpnlResultados.font")); // NOI18N
tbpnlResultados.setName("tbpnlResultados"); // NOI18N
```

```
pnlPoblacion.setBackground(resourceMap.getColor("pnlPoblacion.background")); // NOI18N
pnlPoblacion.setFont(resourceMap.getFont("pnlPoblacion.font")); // NOI18N
pnlPoblacion.setName("pnlPoblacion"); // NOI18N
```

```
jsResumen.setBackground(resourceMap.getColor("jsResumen.background")); // NOI18N
```

```
jsResumen.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cr
eateEtchedBorder(resourceMap.getColor("jsResumen.border.border.highlightColor"), null),
resourceMap.getString("jsResumen.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsResumen.border.titleFont"),
resourceMap.getColor("jsResumen.border.titleColor"))); // NOI18N
jsResumen.setName("jsResumen"); // NOI18N
```

```
tblResumen.setFont(resourceMap.getFont("tblResumen.font")); // NOI18N
tblResumen.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4", "Title 5", "Title 6", "Title 7"
    }
));
tblResumen.setGridColor(resourceMap.getColor("tblResumen.gridColor")); // NOI18N
tblResumen.setName("tblResumen"); // NOI18N
```

```
tblResumen.setSelectionBackground(resourceMap.getColor("tblResumen.selectionBackground"));
tblResumen.setSelectionForeground(resourceMap.getColor("tblResumen.selectionForeground"));
jsResumen.setViewportViewView(tblResumen);
```

```
pnlNombres.setBackground(resourceMap.getColor("pnlNombres.background")); // NOI18N
```

```
pnlNombres.setBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("pnlNombres.border.highlightColor"), null)); // NOI18N
pnlNombres.setName("pnlNombres"); // NOI18N
```

```
lblNomArch.setBackground(resourceMap.getColor("lblNomArch.background")); // NOI18N
lblNomArch.setFont(resourceMap.getFont("lblNomArch.font")); // NOI18N
lblNomArch.setForeground(resourceMap.getColor("lblNomArch.foreground")); // NOI18N
lblNomArch.setText(resourceMap.getString("lblNomArch.text")); // NOI18N
lblNomArch.setName("lblNomArch"); // NOI18N
```

```
lblTamPob.setBackground(resourceMap.getColor("lblTamPob.background")); // NOI18N
lblTamPob.setFont(resourceMap.getFont("lblTamPob.font")); // NOI18N
lblTamPob.setForeground(resourceMap.getColor("lblTamPob.foreground")); // NOI18N
lblTamPob.setText(resourceMap.getString("lblTamPob.text")); // NOI18N
lblTamPob.setName("lblTamPob"); // NOI18N
```

```
lblTamMuestra.setBackground(resourceMap.getColor("lblTamMuestra.background"));
lblTamMuestra.setFont(resourceMap.getFont("lblTamMuestra.font")); // NOI18N
lblTamMuestra.setForeground(resourceMap.getColor("lblTamMuestra.foreground"));
lblTamMuestra.setText(resourceMap.getString("lblTamMuestra.text")); // NOI18N
lblTamMuestra.setName("lblTamMuestra"); // NOI18N
```

```
lblNomArch2.setBackground(resourceMap.getColor("lblNomArch2.background")); // NOI18N
lblNomArch2.setFont(resourceMap.getFont("lblNomArch2.font")); // NOI18N
lblNomArch2.setForeground(resourceMap.getColor("lblNomArch2.foreground")); // NOI18N
lblNomArch2.setText(resourceMap.getString("lblNomArch2.text")); // NOI18N
lblNomArch2.setName("lblNomArch2"); // NOI18N
```

```
lblTamPob2.setBackground(resourceMap.getColor("lblTamPob2.background")); // NOI18N
lblTamPob2.setFont(resourceMap.getFont("lblTamPob2.font")); // NOI18N
lblTamPob2.setForeground(resourceMap.getColor("lblTamPob2.foreground")); // NOI18N
lblTamPob2.setText(resourceMap.getString("lblTamPob2.text")); // NOI18N
lblTamPob2.setName("lblTamPob2"); // NOI18N
```

```
lblTamMuestra2.setBackground(resourceMap.getColor("lblTamMuestra2.background"));
lblTamMuestra2.setFont(resourceMap.getFont("lblTamMuestra2.font")); // NOI18N
lblTamMuestra2.setForeground(resourceMap.getColor("lblTamMuestra2.foreground"));
lblTamMuestra2.setText(resourceMap.getString("lblTamMuestra2.text")); // NOI18N
lblTamMuestra2.setName("lblTamMuestra2"); // NOI18N
```

```
javax.swing.GroupLayout pnlNombresLayout = new javax.swing.GroupLayout(pnlNombres);
```

```

pnINombres.setLayout(pnINombresLayout);
pnINombresLayout.setHorizontalGroup(
    pnINombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnINombresLayout.createSequentialGroup()
        .addContainerGap()

.addGroup(pnINombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addGroup(pnINombresLayout.createSequentialGroup()
        .addComponent(lblTamMuestra)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(lblTamMuestra2))
    .addGroup(pnINombresLayout.createSequentialGroup()
        .addComponent(lblTamPob)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(lblTamPob2))
    .addGroup(pnINombresLayout.createSequentialGroup()
        .addComponent(lblNomArch)
        .addGap(49, 49, 49)
        .addComponent(lblNomArch2)))
    .addContainerGap(93, Short.MAX_VALUE))
);
pnINombresLayout.setVerticalGroup(
    pnINombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnINombresLayout.createSequentialGroup()
        .addGap(21, 21, 21)

.addGroup(pnINombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
    .addComponent(lblNomArch)
    .addComponent(lblNomArch2))
    .addGap(31, 31, 31)

.addGroup(pnINombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
    .addComponent(lblTamPob)
    .addComponent(lblTamPob2))
    .addGap(29, 29, 29)

```

```

.addGroup(pnlNombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
)
    .addComponent(lblTamMuestra)
    .addComponent(lblTamMuestra2)
    .addContainerGap(30, Short.MAX_VALUE)
);

lblMonito.setIcon(resourceMap.getIcon("lblMonito.icon")); // NOI18N
lblMonito.setText(resourceMap.getString("lblMonito.text")); // NOI18N
lblMonito.setName("lblMonito"); // NOI18N

lblFooter.setIcon(resourceMap.getIcon("lblFooter.icon")); // NOI18N
lblFooter.setText(resourceMap.getString("lblFooter.text")); // NOI18N
lblFooter.setName("lblFooter"); // NOI18N

javax.swing.GroupLayout pnlPoblacionLayout = new javax.swing.GroupLayout(pnlPoblacion);
pnlPoblacion.setLayout(pnlPoblacionLayout);
pnlPoblacionLayout.setHorizontalGroup(
    pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlPoblacionLayout.createSequentialGroup())

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
    .addComponent(jsResumen, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 656, Short.MAX_VALUE)
    .addGroup(pnlPoblacionLayout.createSequentialGroup()
        .addGap(115, 115, 115)
        .addComponent(lblMonito)
        .addGap(18, 18, 18)

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
    .addComponent(lblFooter)
    .addComponent(pnlNombres, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))))
    .addContainerGap())
);
pnlPoblacionLayout.setVerticalGroup(
    pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()

```

```

        .addGap(37, 37, 37)
        .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 79,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(46, 46, 46)

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addComponent(pnlNombres, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(lblMonito))
        .addGap(18, 18, 18)
        .addComponent(lblFooter)
        .addGap(27, 27, 27))
);

tblpnlResultados.addTab(resourceMap.getString("pnlPoblacion.TabConstraints.tabTitle"),
pnlPoblacion); // NOI18N

pnlMuestra.setBackground(resourceMap.getColor("pnlMuestra.background")); // NOI18N
pnlMuestra.setName("pnlMuestra"); // NOI18N

jsMuestra.setBackground(resourceMap.getColor("jsMuestra.background")); // NOI18N

jsMuestra.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cre
ateEtchedBorder(resourceMap.getColor("jsMuestra.border.border.highlightColor"), null),
resourceMap.getString("jsMuestra.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsMuestra.border.titleFont"),
resourceMap.getColor("jsMuestra.border.titleColor"))); // NOI18N
jsMuestra.setForeground(resourceMap.getColor("jsMuestra.foreground")); // NOI18N
jsMuestra.setName("jsMuestra"); // NOI18N

tblMuestra.setBackground(resourceMap.getColor("tblMuestra.background")); // NOI18N
tblMuestra.setFont(resourceMap.getFont("tblMuestra.font")); // NOI18N
tblMuestra.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null},
        {null, null},
        {null, null},
        {null, null},
        {null, null}

```

```

    },
    new String [] {
        "Title 1", "Title 2"
    }
));
tblMuestra.setGridColor(resourceMap.getColor("tblMuestra.gridColor")); // NOI18N
tblMuestra.setName("tblMuestra"); // NOI18N

```

```
tblMuestra.setSelectionBackground(resourceMap.getColor("tblMuestra.selectionBackground"));
```

```
tblMuestra.setSelectionForeground(resourceMap.getColor("tblMuestra.selectionForeground"));
jsMuestra.setViewportView(tblMuestra);
```

```

jsLimites.setBackground(resourceMap.getColor("jsLimites.background")); // NOI18N
jsLimites.setName("jsLimites"); // NOI18N

```

```
tblLimites.setFont(resourceMap.getFont("tblLimites.font")); // NOI18N
```

```
tblLimites.setModel(new javax.swing.table.DefaultTableModel(
```

```

    new Object [][] {
        {null, null, null, null, null}
    },

```

```

    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4", "Title 5"
    }
});

```

```
tblLimites.setGridColor(resourceMap.getColor("tblLimites.gridColor")); // NOI18N
tblLimites.setName("tblLimites"); // NOI18N
```

```
tblLimites.setSelectionBackground(resourceMap.getColor("tblLimites.selectionBackground"));
```

```
tblLimites.setSelectionForeground(resourceMap.getColor("tblLimites.selectionForeground"));
```

```
jsLimites.setViewportView(tblLimites);
```

```
lblresultados.setBackground(resourceMap.getColor("lblresultados.background")); // NOI18N
```

```
lblresultados.setFont(resourceMap.getFont("lblresultados.font")); // NOI18N
```

```
lblresultados.setForeground(resourceMap.getColor("lblresultados.foreground")); // NOI18N
```

```
lblresultados.setText(resourceMap.getString("lblresultados.text")); // NOI18N
```

```
lblresultados.setName("lblresultados"); // NOI18N
```

```
lblCredito.setIcon(resourceMap.getIcon("lblCredito.icon")); // NOI18N
```

```
lblCredito.setText(resourceMap.getString("lblCredito.text")); // NOI18N
```

```
lblCredito.setName("lblCredito"); // NOI18N
```

```
lblFooterM.setIcon(resourceMap.getIcon("lblFooterM.icon")); // NOI18N
```



```

lblFooterM.setText(resourceMap.getString("lblFooterM.text")); // NOI18N
lblFooterM.setName("lblFooterM"); // NOI18N

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActionMap(Final.class, this);
    btnGurdar.setAction(actionMap.get("guardarArchivo")); // NOI18N
    btnGurdar.setFont(resourceMap.getFont("btnGurdar.font")); // NOI18N
    btnGurdar.setForeground(resourceMap.getColor("btnGurdar.foreground")); // NOI18N
    btnGurdar.setText(resourceMap.getString("btnGurdar.text")); // NOI18N
    btnGurdar.setName("btnGurdar"); // NOI18N

lblFlechaDer.setIcon(resourceMap.getIcon("lblFlechaDer.icon")); // NOI18N
lblFlechaDer.setText(resourceMap.getString("lblFlechaDer.text")); // NOI18N
lblFlechaDer.setName("lblFlechaDer"); // NOI18N

javax.swing.GroupLayout pnlMuestraLayout = new javax.swing.GroupLayout(pnlMuestra);
pnlMuestra.setLayout(pnlMuestraLayout);
pnlMuestraLayout.setHorizontalGroup(
    pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlMuestraLayout.createSequentialGroup()
            .addGap(14, 14, 14)
            .addComponent(jsMuestra, javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
            .addGroup(pnlMuestraLayout.createSequentialGroup()
                .addGap(184, 184, 184)
                .addComponent(lblresultados)
                .addGap(18, 18, 18)
                .addComponent(lblFlechaDer)
                .addGap(18, 18, 18)
                .addComponent(btnGurdar))
            .addGroup(pnlMuestraLayout.createSequentialGroup()
                .addGap(184, 184, 184)
                .addComponent(lblFooterM)
                .addGap(18, 18, 18)
                .addComponent(btnGurdar))
        );

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnIMuestraLayout.createSequentialGroup()
        .addGap(401, 401, 401)
        .addComponent(lblCredito)))
        .addContainerGap())
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnIMuestraLayout.createSequentialGroup()
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jsLimites, javax.swing.GroupLayout.PREFERRED_SIZE, 453,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(27, 27, 27)))
    );
    pnIMuestraLayout.setVerticalGroup(
        pnIMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnIMuestraLayout.createSequentialGroup()
        .addGap(11, 11, 11)

.addGroup(pnIMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnIMuestraLayout.createSequentialGroup()
        .addComponent(jsMuestra, javax.swing.GroupLayout.DEFAULT_SIZE, 355,
Short.MAX_VALUE)
        .addContainerGap())
        .addGroup(pnIMuestraLayout.createSequentialGroup()
        .addComponent(lblCredito)
        .addGap(7, 7, 7)
        .addComponent(lblresultados)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jsLimites, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(pnIMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
        .addGroup(pnIMuestraLayout.createSequentialGroup()
        .addGap(124, 124, 124)

.addGroup(pnIMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(btnGurdar)
        .addComponent(lblFlechaDer))
        .addGap(22, 22, 22)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnIMuestraLayout.createSequentialGroup()

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(lblFooterM)
        .addGap(31, 31, 31))))))
);

    tbpnlResultados.addTab(resourceMap.getString("pnlMuestra.TabConstraints.tabTitle"),
pnlMuestra); // NOI18N

    btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
    btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
    btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
    btnVolver.setName("btnVolver"); // NOI18N
    btnVolver.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnVolverActionPerformed(evt);
        }
    });

    lblFlechalzq.setIcon(resourceMap.getIcon("lblFlechalzq.icon")); // NOI18N
    lblFlechalzq.setText(resourceMap.getString("lblFlechalzq.text")); // NOI18N
    lblFlechalzq.setName("lblFlechalzq"); // NOI18N

    lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
    lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
    lblHeader.setName("lblHeader"); // NOI18N

    javax.swing.GroupLayout pnlPrincipalLayout = new javax.swing.GroupLayout(pnlPrincipal);
    pnlPrincipal.setLayout(pnlPrincipalLayout);
    pnlPrincipalLayout.setHorizontalGroup(
        pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(lblHeader)
            .addGroup(pnlPrincipalLayout.createSequentialGroup()
                .addComponent(lblFlechalzq)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnVolver))
            .addGroup(pnlPrincipalLayout.createSequentialGroup()
                .addGap(64, 64, 64)
                .addComponent(tbpnlResultados, javax.swing.GroupLayout.PREFERRED_SIZE, 671,
javax.swing.GroupLayout.PREFERRED_SIZE))
    );

```

```

pnlPrincipalLayout.setVerticalGroup(
    pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPrincipalLayout.createSequentialGroup()
            .addComponent(lblHeader)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(tbpnlResultados, javax.swing.GroupLayout.DEFAULT_SIZE, 410,
Short.MAX_VALUE)
            .addGap(16, 16, 16)

        .addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(lblFlechalzq)
            .addComponent(btnVolver)
            .addContainerGap()
        );

mnuArriba.setName("mnuArriba"); // NOI18N

smnuArchivo.setText(resourceMap.getString("smnuArchivo.text")); // NOI18N
smnuArchivo.setName("smnuArchivo"); // NOI18N

ssmnuSalir.setAction(actionMap.get("quit")); // NOI18N
ssmnuSalir.setText(resourceMap.getString("ssmnuSalir.text")); // NOI18N
ssmnuSalir.setName("ssmnuSalir"); // NOI18N
smnuArchivo.add(ssmnuSalir);

mnuArriba.add(smnuArchivo);

smnuAyuda.setText(resourceMap.getString("smnuAyuda.text")); // NOI18N
smnuAyuda.setName("smnuAyuda"); // NOI18N
mnuArriba.add(smnuAyuda);

setJMenuBar(mnuArriba);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlPrincipal, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(pnlPrincipal, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
}

private void initComponents(String nom_arch,String tamM,String tamP,TablaResPob
tr,TablaMuestra tm,TablaFinal tf,String stat)
{
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenu2 = new javax.swing.JMenu();
    pnlPrincipal = new javax.swing.JPanel();
    tbpnlResultados = new javax.swing.JTabbedPane();
    pnlPoblacion = new javax.swing.JPanel();
    jsResumen = new javax.swing.JScrollPane();
    tblResumen = new javax.swing.JTable(tr); //+++++++
    pnlNombres = new javax.swing.JPanel();
    lblNomArch = new javax.swing.JLabel();
    lblTamPob = new javax.swing.JLabel();
    lblTamMuestra = new javax.swing.JLabel();
    lblNomArch2 = new javax.swing.JLabel();
    lblTamPob2 = new javax.swing.JLabel();
    lblTamMuestra2 = new javax.swing.JLabel();
    lblMonito = new javax.swing.JLabel();
    lblFooter = new javax.swing.JLabel();
    pnlMuestra = new javax.swing.JPanel();
    jsMuestra = new javax.swing.JScrollPane();
    tblMuestra = new javax.swing.JTable(tm); //+++++++
    jsLimites = new javax.swing.JScrollPane();
    tblLimites = new javax.swing.JTable(tf); //+++++++
    lblresultados = new javax.swing.JLabel();
    lblCredito = new javax.swing.JLabel();
    lblFooterM = new javax.swing.JLabel();
    btnGurdar = new javax.swing.JButton();
    lblFlechaDer = new javax.swing.JLabel();
    btnVolver = new javax.swing.JButton();
    lblFlechalzq = new javax.swing.JLabel();
    lblHeader = new javax.swing.JLabel();
    mnuArriba = new javax.swing.JMenuBar();
    smnuArchivo = new javax.swing.JMenu();
}

```

```

ssmnuSalir = new javax.swing.JMenuItem();
smnuAyuda = new javax.swing.JMenu();

jMenuBar1.setName("jMenuBar1"); // NOI18N

org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(Final.class);
jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N
jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle(resourceMap.getString("Form.title")); // NOI18N
setName("Form"); // NOI18N

pnlPrincipal.setBackground(resourceMap.getColor("pnlPrincipal.background")); // NOI18N
pnlPrincipal.setName("pnlPrincipal"); // NOI18N

tbpnlResultados.setForeground(resourceMap.getColor("tbpnlResultados.foreground"));
tbpnlResultados.setFont(resourceMap.getFont("tbpnlResultados.font")); // NOI18N
tbpnlResultados.setName("tbpnlResultados"); // NOI18N

pnlPoblacion.setBackground(resourceMap.getColor("pnlPoblacion.background")); // NOI18N
pnlPoblacion.setFont(resourceMap.getFont("pnlPoblacion.font")); // NOI18N
pnlPoblacion.setName("pnlPoblacion"); // NOI18N

jsResumen.setBackground(resourceMap.getColor("jsResumen.background")); // NOI18N

jsResumen.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cr
eateEtchedBorder(resourceMap.getColor("jsResumen.border.border.highlightColor"), null),
resourceMap.getString("jsResumen.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsResumen.border.titleFont"),
resourceMap.getColor("jsResumen.border.titleColor"))); // NOI18N
jsResumen.setName("jsResumen"); // NOI18N

```

```

tblResumen.setFont(resourceMap.getFont("tblResumen.font")); // NOI18N
tblResumen.setGridColor(resourceMap.getColor("tblResumen.gridColor")); // NOI18N
tblResumen.setName("tblResumen"); // NOI18N

tblResumen.setSelectionBackground(resourceMap.getColor("tblResumen.selectionBackground"));
tblResumen.setSelectionForeground(resourceMap.getColor("tblResumen.selectionForeground"));
jsResumen.setViewPortView(tblResumen);

pnlNombres.setBackground(resourceMap.getColor("pnlNombres.background")); // NOI18N

pnlNombres.setBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("pnl
Nombres.border.highlightColor"), null)); // NOI18N
pnlNombres.setName("pnlNombres"); // NOI18N

lblNomArch.setBackground(resourceMap.getColor("lblNomArch.background")); // NOI18N
lblNomArch.setFont(resourceMap.getFont("lblNomArch.font")); // NOI18N
lblNomArch.setForeground(resourceMap.getColor("lblNomArch.foreground")); // NOI18N
lblNomArch.setText(resourceMap.getString("lblNomArch.text")); // NOI18N
lblNomArch.setName("lblNomArch"); // NOI18N

lblTamPob.setBackground(resourceMap.getColor("lblTamPob.background")); // NOI18N
lblTamPob.setFont(resourceMap.getFont("lblTamPob.font")); // NOI18N
lblTamPob.setForeground(resourceMap.getColor("lblTamPob.foreground")); // NOI18N
lblTamPob.setText(resourceMap.getString("lblTamPob.text")); // NOI18N // NOI18N
lblTamPob.setName("lblTamPob"); // NOI18N

lblTamMuestra.setBackground(resourceMap.getColor("lblTamMuestra.background"));
lblTamMuestra.setFont(resourceMap.getFont("lblTamMuestra.font")); // NOI18N
lblTamMuestra.setForeground(resourceMap.getColor("lblTamMuestra.foreground"));
lblTamMuestra.setText(resourceMap.getString("lblTamMuestra.text")); // NOI18N// NOI18N
lblTamMuestra.setName("lblTamMuestra"); // NOI18N

lblNomArch2.setBackground(resourceMap.getColor("lblNomArch2.background")); // NOI18N
lblNomArch2.setFont(resourceMap.getFont("lblNomArch2.font")); // NOI18N
lblNomArch2.setForeground(resourceMap.getColor("lblNomArch2.foreground")); // NOI18N
lblNomArch2.setText(nom_arch); // NOI18N
lblNomArch2.setName("lblNomArch2"); // NOI18N

lblTamPob2.setBackground(resourceMap.getColor("lblTamPob2.background")); // NOI18N
lblTamPob2.setFont(resourceMap.getFont("lblTamPob2.font")); // NOI18N
lblTamPob2.setForeground(resourceMap.getColor("lblTamPob2.foreground")); // NOI18N
lblTamPob2.setText(tamP); // NOI18N

```

```

lbtamPob2.setName("lbtamPob2"); // NOI18N

lbtamMuestra2.setBackground(resourceMap.getColor("lbtamMuestra2.background"));
lbtamMuestra2.setFont(resourceMap.getFont("lbtamMuestra2.font")); // NOI18N
lbtamMuestra2.setForeground(resourceMap.getColor("lbtamMuestra2.foreground"));
lbtamMuestra2.setText(tamM); // NOI18N
lbtamMuestra2.setName("lbtamMuestra2"); // NOI18N

javax.swing.GroupLayout pnlNombresLayout = new javax.swing.GroupLayout(pnlNombres);
pnlNombres.setLayout(pnlNombresLayout);
pnlNombresLayout.setHorizontalGroup(
    pnlNombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlNombresLayout.createSequentialGroup()
            .addComponent(lbtamMuestra)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(lbtamMuestra2))
        .addGroup(pnlNombresLayout.createSequentialGroup()
            .addComponent(lbtamPob)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(lbtamPob2))
        .addGroup(pnlNombresLayout.createSequentialGroup()
            .addComponent(lbtamNomArch)
            .addGap(49, 49, 49)
            .addComponent(lbtamNomArch2)))
        .addGap(93, Short.MAX_VALUE));
pnlNombresLayout.setVerticalGroup(
    pnlNombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlNombresLayout.createSequentialGroup()
            .addComponent(lbtamMuestra)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(lbtamMuestra2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(lbtamPob)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(lbtamPob2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(lbtamNomArch)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(lbtamNomArch2)
            .addContainerGap())
        .addGroup(pnlNombresLayout.createSequentialGroup()
            .addComponent(lbtamNomArch)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(lbtamNomArch2)
            .addContainerGap())
);

```



```

        .addGap(31, 31, 31)

    .addGroup(pnlNombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
    )
        .addComponent(lblTamPob)
        .addComponent(lblTamPob2))
        .addGap(29, 29, 29)

    .addGroup(pnlNombresLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE
    )
        .addComponent(lblTamMuestra)
        .addComponent(lblTamMuestra2))
        .addContainerGap(30, Short.MAX_VALUE)
    );

    lblMonito.setIcon(resourceMap.getIcon("jLabel3.icon")); // NOI18N
    lblMonito.setText(resourceMap.getString("jLabel3.text")); // NOI18N
    lblMonito.setName("jLabel3"); // NOI18N

    lblFooter.setIcon(resourceMap.getIcon("jLabel4.icon")); // NOI18N
    lblFooter.setText(resourceMap.getString("jLabel4.text")); // NOI18N
    lblFooter.setName("jLabel4"); // NOI18N

    javax.swing.GroupLayout pnlPoblacionLayout = new javax.swing.GroupLayout(pnlPoblacion);
    pnlPoblacion.setLayout(pnlPoblacionLayout);
    pnlPoblacionLayout.setHorizontalGroup(
        pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPoblacionLayout.createSequentialGroup())

    .addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
    )
        .addComponent(jsResumen, javax.swing.GroupLayout.Alignment.TRAILING,
    javax.swing.GroupLayout.DEFAULT_SIZE, 656, Short.MAX_VALUE)
        .addGroup(pnlPoblacionLayout.createSequentialGroup())
            .addGap(115, 115, 115)
            .addComponent(lblMonito)
            .addGap(18, 18, 18)

    .addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
    )
        .addComponent(lblFooter)

```

```

        .addComponent(pnlNombres, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap()
    );
    pnlPoblacionLayout.setVerticalGroup(
        pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
            .addGap(37, 37, 37)
            .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 79,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(46, 46, 46)

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addComponent(pnlNombres, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(lblMonito))
            .addGap(18, 18, 18)
            .addComponent(lblFooter)
            .addGap(27, 27, 27))
    );

    tbpnlResultados.addTab(resourceMap.getString("pnlPoblacion.TabConstraints.tabTitle"),
pnlPoblacion); // NOI18N

    pnlMuestra.setBackground(resourceMap.getColor("pnlMuestra.background")); // NOI18N
    pnlMuestra.setName("pnlMuestra"); // NOI18N

    jsMuestra.setBackground(resourceMap.getColor("jsMuestra.background")); // NOI18N

jsMuestra.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cre
ateEtchedBorder(resourceMap.getColor("jsMuestra.border.border.highlightColor"), null),
resourceMap.getString("jsMuestra.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsMuestra.border.titleFont"),
resourceMap.getColor("jsMuestra.border.titleColor"))); // NOI18N
    jsMuestra.setForeground(resourceMap.getColor("jsMuestra.foreground")); // NOI18N
    jsMuestra.setName("jsMuestra"); // NOI18N

    tblMuestra.setBackground(resourceMap.getColor("tblMuestra.background")); // NOI18N

```

```

tblMuestra.setFont(resourceMap.getFont("tblMuestra.font")); // NOI18N
tblMuestra.setGridColor(resourceMap.getColor("tblMuestra.gridColor")); // NOI18N
tblMuestra.setName("tblMuestra"); // NOI18N

tblMuestra.setSelectionBackground(resourceMap.getColor("tblMuestra.selectionBackground"));
tblMuestra.setSelectionForeground(resourceMap.getColor("tblMuestra.selectionForeground"));
jsMuestra.setViewportView(tblMuestra);

jsLimites.setBackground(resourceMap.getColor("jsLimites.background")); // NOI18N
jsLimites.setName("jsLimites"); // NOI18N

tblLimites.setFont(resourceMap.getFont("tblLimites.font")); // NOI18N
tblLimites.setGridColor(resourceMap.getColor("tblLimites.gridColor")); // NOI18N
tblLimites.setName("tblLimites"); // NOI18N
tblLimites.setSelectionBackground(resourceMap.getColor("tblLimites.selectionBackground"));
tblLimites.setSelectionForeground(resourceMap.getColor("tblLimites.selectionForeground"));
jsLimites.setViewportView(tblLimites);

lblresultados.setBackground(resourceMap.getColor("lblresultados.background")); // NOI18N
lblresultados.setFont(resourceMap.getFont("lblresultados.font")); // NOI18N
lblresultados.setForeground(resourceMap.getColor("lblresultados.foreground")); // NOI18N
lblresultados.setText(resourceMap.getString("lblresultados.text")); // NOI18N
lblresultados.setName("lblresultados"); // NOI18N

lblCredito.setIcon(resourceMap.getIcon("lblCredito.icon")); // NOI18N
lblCredito.setText(resourceMap.getString("lblCredito.text")); // NOI18N
lblCredito.setName("lblCredito"); // NOI18N

lblFooterM.setIcon(resourceMap.getIcon("jLabel1.icon")); // NOI18N
lblFooterM.setText(resourceMap.getString("jLabel1.text")); // NOI18N
lblFooterM.setName("jLabel1"); // NOI18N

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActionMap(Final.class, this);
btnGurdar.setAction(actionMap.get("guardarArchivo")); // NOI18N
btnGurdar.setFont(resourceMap.getFont("btnGurdar.font")); // NOI18N
btnGurdar.setForeground(resourceMap.getColor("btnGurdar.foreground")); // NOI18N
btnGurdar.setText(resourceMap.getString("btnGurdar.text")); // NOI18N
btnGurdar.setName("btnGurdar"); // NOI18N

lblFlechaDer.setIcon(resourceMap.getIcon("jLabel2.icon")); // NOI18N

```



```

pnIMuestraLayout.setVerticalGroup(
    pnIMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnIMuestraLayout.createSequentialGroup()
            .addGap(11, 11, 11)

.addGroup(pnIMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnIMuestraLayout.createSequentialGroup()
        .addComponent(jsMuestra, javax.swing.GroupLayout.DEFAULT_SIZE, 355,
Short.MAX_VALUE)
        .addContainerGap())
        .addGroup(pnIMuestraLayout.createSequentialGroup()
            .addComponent(lblCredito)
            .addGap(7, 7, 7)
            .addComponent(lblresultados)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jsLimites, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(pnIMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addGroup(pnIMuestraLayout.createSequentialGroup()
        .addGap(124, 124, 124)

.addGroup(pnIMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(btnGurdar)
    .addComponent(lblFlechaDer))
    .addGap(22, 22, 22))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnIMuestraLayout.createSequentialGroup()
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(lblFooterM)
        .addGap(31, 31, 31))))))
);

tbpnlResultados.addTab(resourceMap.getString("pnIMuestra.TabConstraints.tabTitle"),
pnIMuestra); // NOI18N

btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
btnVolver.setName("btnVolver"); // NOI18N

```

```

btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnVolverActionPerformed(evt);
    }
});

lblFlechalzq.setIcon(resourceMap.getIcon("lblFlecha.icon")); // NOI18N
lblFlechalzq.setText(resourceMap.getString("lblFlecha.text")); // NOI18N
lblFlechalzq.setName("lblFlecha"); // NOI18N

lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
lblHeader.setName("lblHeader"); // NOI18N

javax.swing.GroupLayout pnlPrincipalLayout = new javax.swing.GroupLayout(pnlPrincipal);
pnlPrincipal.setLayout(pnlPrincipalLayout);
pnlPrincipalLayout.setHorizontalGroup(
    pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblHeader)
        .addGroup(pnlPrincipalLayout.createSequentialGroup()
            .addGap(64, 64, 64)
            .addComponent(lblFlechalzq)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(btnVolver))
        .addGroup(pnlPrincipalLayout.createSequentialGroup()
            .addGap(64, 64, 64)
            .addComponent(tbPnlResultados, javax.swing.GroupLayout.PREFERRED_SIZE, 671,
javax.swing.GroupLayout.PREFERRED_SIZE))
);
pnlPrincipalLayout.setVerticalGroup(
    pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPrincipalLayout.createSequentialGroup()
            .addComponent(lblHeader)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tbPnlResultados, javax.swing.GroupLayout.DEFAULT_SIZE, 410,
Short.MAX_VALUE)
            .addGap(16, 16, 16)
        )
        .addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(lblFlechalzq)
            .addComponent(btnVolver)
            .addGap(16, 16, 16)
        )
);

```

```

);

mnuArriba.setName("mnuArriba"); // NOI18N
smnuArchivo.setText(resourceMap.getString("smnuArchivo.text")); // NOI18N
smnuArchivo.setName("smnuArchivo"); // NOI18N
ssmnuSalir.setAction(actionMap.get("quit")); // NOI18N
ssmnuSalir.setText(resourceMap.getString("ssmnuSalir.text")); // NOI18N
ssmnuSalir.setName("ssmnuSalir"); // NOI18N
smnuArchivo.add(ssmnuSalir);
mnuArriba.add(smnuArchivo);
smnuAyuda.setText(resourceMap.getString("smnuAyuda.text")); // NOI18N
smnuAyuda.setName("smnuAyuda"); // NOI18N
mnuArriba.add(smnuAyuda);
setJMenuBar(mnuArriba);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlPrincipal, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlPrincipal, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
pack();
}
private void btnVolverActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

public static void main(String args[], TablaResPob tr, TablaMuestra tm, TablaFinal tf, String stat) {
    final String metodo=args[0];
    final String nombre_archivo=args[1];
    final String tam_muestra=args[2];
    final String tam_poblacion=args[3];
    final TablaResPob tres=tr;
    final TablaMuestra tmu=tm;
    final TablaFinal tfi=tf;
    final String st=stat;

```

```

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new
Final(nombre_archivo,tam_muestra,tam_poblacion,metodo,tres,tmu,tfi,st).setVisible(true);
            }
        });
    }
}

```

```

@Action
public void guardarArchivo() {
    Archivos archivo=new Archivos();
    String argumentos[]={nombre_metodo,statu,nomArch,tMuestra,tPoblacion};
    archivo.guardar(tsample, tresumen, argumentos,0);
}

```

```

// Declaración de Variables
private javax.swing.JButton btnGurdar;
private javax.swing.JButton btnVolver;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JScrollPane jsLimites;
private javax.swing.JScrollPane jsMuestra;
private javax.swing.JScrollPane jsResumen;
private javax.swing.JLabel lblCerdito;
private javax.swing.JLabel lblFlechaDer;
private javax.swing.JLabel lblFlechalzq;
private javax.swing.JLabel lblFooter;
private javax.swing.JLabel lblFooterM;
private javax.swing.JLabel lblHeader;
private javax.swing.JLabel lblMonito;
private javax.swing.JLabel lblNomArch;
private javax.swing.JLabel lblNomArch2;
private javax.swing.JLabel lblTamMuestra;
private javax.swing.JLabel lblTamMuestra2;
private javax.swing.JLabel lblTamPob;
private javax.swing.JLabel lblTamPob2;
private javax.swing.JLabel lblresultados;
private javax.swing.JMenuBar mnuArriba;
private javax.swing.JPanel pnlMuestra;
private javax.swing.JPanel pnlNombres;

```



```

private javax.swing.JPanel pnlPoblacion;
private javax.swing.JPanel pnlPrincipal;
private javax.swing.JMenu smnuArchivo;
private javax.swing.JMenu smnuAyuda;
private javax.swing.JMenuItem ssmnuSalir;
private javax.swing.JTable tblLimites;
private javax.swing.JTable tblMuestra;
private javax.swing.JTable tblResumen;
private javax.swing.JTabbedPane tbpnlResultados;
}

```

### 9.2.9 FinalEstrata.java

```

/*
 * FinalEstrata.java
 * Created on 20/01/2011, 07:49:15 PM
 */

package pt02.vistas;
import java.util.LinkedList;
import javax.swing.table.TableModel;
import pt02.archivos.Archivos;
import pt02.modelosTablas.TablaResPob;

public class FinalEstrata extends javax.swing.JFrame
{
public String nombreArchivo;
public String modoArchivo;
public String tipoAlgoritmo;
public String variable;
public LinkedList tamEstratos;
public String toma_muestra;
public TableModel informacion;
public int indicador;

public FinalEstrata(String nom_arch,String modo,String tipo,String nom_var,TableModel
m,TableModel c,TablaResPob rp,LinkedList tamStratos,String tm)
{
this.nombreArchivo=nom_arch;
this.modoArchivo=modo;
this.tipoAlgoritmo=tipo;
this.variable=nom_var;

```

```

    this.tamEstratos=tamStratos;
    this.toma_muestra=tm;
    initComponents(m,c,rp,0);
}

    public FinalEstrata(String nom_arch,String tipo,String varselecc,String
tammuestra,TableModel m,TableModel c,TablaResPob rp,TableModel info)
{
    this.nombreArchivo=nom_arch; //nombre del archivo
    this.modaArchivo=tipo; //si fue frecuencias o probabilidades
    this.tipoAlgoritmo=tammuestra; //variable seleccionada
    this.variable=varselecc;
    this.informacion=info;
    initComponents(m,c,rp,1);
}

@SuppressWarnings("unchecked")
private void initComponents() {

    mainPanel = new javax.swing.JPanel();
    lblHeader = new javax.swing.JLabel();
    btnGuardar = new javax.swing.JButton();
    btnVolver = new javax.swing.JButton();
    lblFooter = new javax.swing.JLabel();
    tabbedResultados = new javax.swing.JTabbedPane();
    pnlPoblacion = new javax.swing.JPanel();
    jsResumen = new javax.swing.JScrollPane();
    tblResumenPoblacion = new javax.swing.JTable();
    jsInfo = new javax.swing.JScrollPane();
    tblInfo = new javax.swing.JTable();
    lblHappyDance = new javax.swing.JLabel();
    pnlMuestra = new javax.swing.JPanel();
    jsEstratos = new javax.swing.JScrollPane();
    tblEstratos = new javax.swing.JTable();
    pnlCalculos = new javax.swing.JPanel();
    jsCalculos = new javax.swing.JScrollPane();
    tblCalculos = new javax.swing.JTable();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenuItem2 = new javax.swing.JMenuItem();
}

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(FinalEstrata.class);
setTitle(resourceMap.getString("Form.title")); // NOI18N
setName("Form"); // NOI18N

mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
mainPanel.setName("mainPanel"); // NOI18N

lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
lblHeader.setName("lblHeader"); // NOI18N

btnGuardar.setFont(resourceMap.getFont("btnGuardar.font")); // NOI18N
btnGuardar.setForeground(resourceMap.getColor("btnGuardar.foreground")); // NOI18N
btnGuardar.setText(resourceMap.getString("btnGuardar.text")); // NOI18N
btnGuardar.setName("btnGuardar"); // NOI18N
btnGuardar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        guardar(evt);
    }
});

btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
btnVolver.setName("btnVolver"); // NOI18N
btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        volver(evt);
    }
});

lblFooter.setIcon(resourceMap.getIcon("lblFooter.icon")); // NOI18N
lblFooter.setText(resourceMap.getString("lblFooter.text")); // NOI18N
lblFooter.setName("lblFooter"); // NOI18N

tabbedResultados.setBackground(resourceMap.getColor("tabbedResultados.background"));
tabbedResultados.setForeground(resourceMap.getColor("tabbedResultados.foreground"));
tabbedResultados.setFont(resourceMap.getFont("tabbedResultados.font")); // NOI18N
tabbedResultados.setName("tabbedResultados"); // NOI18N

```

```
pnlPoblacion.setBackground(resourceMap.getColor("pnlPoblacion.background")); // NOI18N
pnlPoblacion.setName("pnlPoblacion"); // NOI18N
```

```
jsResumen.setBackground(resourceMap.getColor("jsResumen.background")); // NOI18N
```

```
jsResumen.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(null, resourceMap.getColor("jsResumen.border.border.shadowColor")),
resourceMap.getString("jsResumen.border.title"), javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsResumen.border.titleFont"),
resourceMap.getColor("jsResumen.border.titleColor"))); // NOI18N
jsResumen.setName("jsResumen");
```

```
tblResumenPoblacion.setForeground(resourceMap.getColor("tblResumenPoblacion.foreground"));
;
```

```
tblResumenPoblacion.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
tblResumenPoblacion.setName("tblResumenPoblacion"); // NOI18N
jsResumen.setViewportView(tblResumenPoblacion);
```

```
jsInfo.setBackground(resourceMap.getColor("jsInfo.background")); // NOI18N
```

```
jsInfo.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("jsInfo.border.border.highlightColor"),
resourceMap.getColor("jsInfo.border.border.shadowColor")),
resourceMap.getString("jsInfo.border.title"), javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsInfo.border.titleFont"),
resourceMap.getColor("jsInfo.border.titleColor"))); // NOI18N
jsInfo.setName("jsInfo"); // NOI18N
```

```
tblInfo.setModel(new javax.swing.table.DefaultTableModel(
```



```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnIPoblacionLayout.createSequentialGroup()
        .addGap(37, 37, 37)
        .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(pnIPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addGroup(pnIPoblacionLayout.createSequentialGroup()
        .addGap(58, 58, 58)
        .addComponent(jsInfo, javax.swing.GroupLayout.PREFERRED_SIZE, 177,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnIPoblacionLayout.createSequentialGroup()
        .addGap(78, 78, 78)
        .addComponent(lblHappyDance)))
        .addGap(44, 44, 44)
);

tabbedResultados.addTab(resourceMap.getString("pnIPoblacion.TabConstraints.tabTitle"),
pnIPoblacion); // NOI18N

pnIMuestra.setBackground(resourceMap.getColor("pnIMuestra.background")); // NOI18N
pnIMuestra.setName("pnIMuestra"); // NOI18N

jsEstratos.setBackground(resourceMap.getColor("jsEstratos.background")); // NOI18N

jsEstratos.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cre
ateEtchedBorder(resourceMap.getColor("jsEstratos.border.border.highlightColor"),
resourceMap.getColor("jsEstratos.border.border.shadowColor")),
resourceMap.getString("jsEstratos.border.title"), javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsEstratos.border.titleFont"),
resourceMap.getColor("jsEstratos.border.titleColor"))); // NOI18N
jsEstratos.setName("jsEstratos"); // NOI18N

tblEstratos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
);

```

```

    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
tblEstratos.setName("tblEstratos"); // NOI18N
jsEstratos.setViewportView(tblEstratos);

javax.swing.GroupLayout pnlMuestraLayout = new javax.swing.GroupLayout(pnlMuestra);
pnlMuestra.setLayout(pnlMuestraLayout);
pnlMuestraLayout.setHorizontalGroup(
    pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlMuestraLayout.createSequentialGroup()
            .addGap(75, 75, 75)
            .addComponent(jsEstratos, javax.swing.GroupLayout.PREFERRED_SIZE, 608,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(79, Short.MAX_VALUE))
        );
pnlMuestraLayout.setVerticalGroup(
    pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlMuestraLayout.createSequentialGroup()
            .addGap(28, 28, 28)
            .addComponent(jsEstratos, javax.swing.GroupLayout.DEFAULT_SIZE, 352,
Short.MAX_VALUE)
            .addGap(79, Short.MAX_VALUE))
        );

tabbedResultados.addTab(resourceMap.getString("pnlMuestra.TabConstraints.tabTitle"),
pnlMuestra); // NOI18N

pnlCalculos.setBackground(resourceMap.getColor("pnlCalculos.background")); // NOI18N
pnlCalculos.setName("pnlCalculos"); // NOI18N

jsCalculos.setBackground(resourceMap.getColor("jsCalculos.background")); // NOI18N

jsCalculos.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cre
ateEtchedBorder(resourceMap.getColor("jsCalculos.border.border.highlightColor"),
resourceMap.getColor("jsCalculos.border.border.shadowColor")),
resourceMap.getString("jsCalculos.border.title"), javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsCalculos.border.titleFont"),
resourceMap.getColor("jsCalculos.border.titleColor"))); // NOI18N

```

```

jsCalculos.setName("jsCalculos"); // NOI18N

tblCalculos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
tblCalculos.setName("tblCalculos"); // NOI18N
jsCalculos.setViewportView(tblCalculos);

javax.swing.GroupLayout pnlCalculosLayout = new javax.swing.GroupLayout(pnlCalculos);
pnlCalculos.setLayout(pnlCalculosLayout);
pnlCalculosLayout.setHorizontalGroup(
    pnlCalculosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlCalculosLayout.createSequentialGroup()
            .addGap(44, 44, 44)
            .addComponent(jsCalculos, javax.swing.GroupLayout.PREFERRED_SIZE, 655,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(63, Short.MAX_VALUE))
        );
pnlCalculosLayout.setVerticalGroup(
    pnlCalculosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlCalculosLayout.createSequentialGroup()
            .addGap(59, 59, 59)
            .addComponent(jsCalculos, javax.swing.GroupLayout.PREFERRED_SIZE, 260,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(72, Short.MAX_VALUE))
        );

tabbedResultados.addTab(resourceMap.getString("pnlCalculos.TabConstraints.tabTitle"),
    pnlCalculos); // NOI18N

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```



```

        .addComponent(lblHeader)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addGap(13, 13, 13)
            .addComponent(btnVolver)
            .addGap(18, 18, 18)
            .addComponent(lblFooter)
            .addGap(18, 18, 18)
            .addComponent(btnGuardar)
            .addContainerGap(30, Short.MAX_VALUE))
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(tabbedResultados, javax.swing.GroupLayout.DEFAULT_SIZE, 767,
Short.MAX_VALUE)
            .addGap(23, 23, 23))
    );
    mainPanelLayout.setVerticalGroup(
        mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addComponent(lblHeader)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tabbedResultados, javax.swing.GroupLayout.PREFERRED_SIZE, 423,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(37, 37, 37)

        .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(btnGuardar)
            .addComponent(lblFooter)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(btnVolver)))
            .addContainerGap()
    );

    jMenuItem1.setName("jMenuBar1"); // NOI18N

    jMenuItem1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
    jMenuItem1.setName("jMenu1"); // NOI18N

    javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActio
nMap(FinalEstrata.class, this);
    jMenuItem1.setAction(actionMap.get("quit")); // NOI18N

```

```

jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
jMenuItem1.setName("jMenuItem1"); // NOI18N
jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);

pack();
}

private void initComponents(TableModel muestras, TableModel calculos, TablaResPob
resPob,int id)
{

    mainPanel = new javax.swing.JPanel();
    lblHeader = new javax.swing.JLabel();
    btnGuardar = new javax.swing.JButton();
    btnVolver = new javax.swing.JButton();
    lblFooter = new javax.swing.JLabel();
    tabbedResultados = new javax.swing.JTabbedPane();
    pnlPoblacion = new javax.swing.JPanel();
    jsResumen = new javax.swing.JScrollPane();
    tblResumenPoblacion = new javax.swing.JTable();
    jsInfo = new javax.swing.JScrollPane();

```

```

tblInfo = new javax.swing.JTable();
lblHappyDance = new javax.swing.JLabel();
pnlMuestra = new javax.swing.JPanel();
jsEstratos = new javax.swing.JScrollPane();
tblEstratos = new javax.swing.JTable();
pnlCalculos = new javax.swing.JPanel();
jsCalculos = new javax.swing.JScrollPane();
tblCalculos = new javax.swing.JTable();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
indicador=id;

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setName("Resultados Método Estratificado"); // NOI18N

```

```

org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(FinalEstrata.class);

```

```

setTitle(resourceMap.getString("Form.title")); // NOI18N
mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
mainPanel.setName("mainPanel"); // NOI18N

```

```

lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
lblHeader.setName("lblHeader"); // NOI18N

```

```

btnGuardar.setFont(resourceMap.getFont("btnGuardar.font")); // NOI18N
btnGuardar.setForeground(resourceMap.getColor("btnGuardar.foreground")); // NOI18N
btnGuardar.setText(resourceMap.getString("btnGuardar.text")); // NOI18N
btnGuardar.setName("btnGuardar"); // NOI18N
btnGuardar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        guardar(evt);
    }
});

```

```

btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
btnVolver.setName("btnVolver"); // NOI18N

```

```
btnVolver.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        volver(evt);  
    }  
});
```

```
lblFooter.setIcon(resourceMap.getIcon("lblFooter.icon")); // NOI18N  
lblFooter.setText(resourceMap.getString("lblFooter.text")); // NOI18N  
lblFooter.setName("lblFooter"); // NOI18N
```

```
tabbedResultados.setBackground(resourceMap.getColor("tabbedResultados.background"));  
tabbedResultados.setForeground(resourceMap.getColor("tabbedResultados.foreground"));  
tabbedResultados.setFont(resourceMap.getFont("tabbedResultados.font")); // NOI18N  
tabbedResultados.setName("tabbedResultados"); // NOI18N
```

```
pnlPoblacion.setBackground(resourceMap.getColor("pnlPoblacion.background")); // NOI18N  
pnlPoblacion.setName("pnlPoblacion"); // NOI18N
```

```
jsResumen.setBackground(resourceMap.getColor("jsResumen.background")); // NOI18N
```

```
jsResumen.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cr  
eateEtchedBorder(null, resourceMap.getColor("jsResumen.border.border.shadowColor"),  
resourceMap.getString("jsResumen.border.title"), javax.swing.border.TitledBorder.CENTER,  
javax.swing.border.TitledBorder.DEFAULT_POSITION,  
resourceMap.getFont("jsResumen.border.titleFont"),  
resourceMap.getColor("jsResumen.border.titleColor"))); // NOI18N  
jsResumen.setName("jsResumen"); // NOI18N
```

```
tblResumenPoblacion.setForeground(resourceMap.getColor("tblResumenPoblacion.foreground"))  
;  
tblResumenPoblacion.setModel(resPob);  
tblResumenPoblacion.setName("tblResumenPoblacion"); // NOI18N  
jsResumen.setViewportView(tblResumenPoblacion);
```

```
jsInfo.setBackground(resourceMap.getColor("jsInfo.background")); // NOI18N
```

```
jsInfo.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEt  
chedBorder(resourceMap.getColor("jsInfo.border.border.highlightColor"),  
resourceMap.getColor("jsInfo.border.border.shadowColor"),  
resourceMap.getString("jsInfo.border.title"), javax.swing.border.TitledBorder.CENTER,  
javax.swing.border.TitledBorder.DEFAULT_POSITION,
```

```

resourceMap.getFont("jsInfo.border.titleFont"),
resourceMap.getColor("jsInfo.border.titleColor")); // NOI18N
    jsInfo.setName("jsInfo"); // NOI18N

String tamaños = new String();
if(id==0) //estratificado
{

    for(int i=0;i<tamEstratos.size();i++)
    tamaños=tamaños.concat(tamEstratos.get(i).toString()+",");
    int resta=tamaños.length();
    resta=resta-1;
    tamaños=tamaños.substring(0,resta);
    tblInfo.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {"Nombre Archivo", nombreArchivo},
        {"Tamaño Estratos", tamaños},
        {"Método de Selección", tipoAlgoritmo},
        {"Variable Seleccionada", variable},
        {"Toma de Muestra", toma_muestra},
    },
    new String [] {
        "", ""
    }
    ));
}
else //aleatorio simple ponderado
{
    tblInfo.setModel(informacion);
}
tblInfo.setName("tblInfo"); // NOI18N
jsInfo.setViewPortView(tblInfo);

lblHappyDance.setIcon(resourceMap.getIcon("lblHappyDance.icon")); // NOI18N
lblHappyDance.setText(resourceMap.getString("lblHappyDance.text")); // NOI18N
lblHappyDance.setName("lblHappyDance"); // NOI18N

javax.swing.GroupLayout pnlPoblacionLayout = new javax.swing.GroupLayout(pnlPoblacion);
pnlPoblacion.setLayout(pnlPoblacionLayout);
pnlPoblacionLayout.setHorizontalGroup(
    pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlPoblacionLayout.createSequentialGroup()

```

```

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING
, false)
    .addGroup(pnlPoblacionLayout.createSequentialGroup()
        .addGap(70, 70, 70)
        .addComponent(lblHappyDance)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jsInfo, javax.swing.GroupLayout.PREFERRED_SIZE, 353,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlPoblacionLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 734,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(18, Short.MAX_VALUE))
);
pnlPoblacionLayout.setVerticalGroup(
    pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
            .addGap(37, 37, 37)
            .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
    .addGroup(pnlPoblacionLayout.createSequentialGroup()
        .addGap(58, 58, 58)
        .addComponent(jsInfo, javax.swing.GroupLayout.PREFERRED_SIZE, 177,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(pnlPoblacionLayout.createSequentialGroup()
            .addGap(78, 78, 78)
            .addComponent(lblHappyDance)))
        .addGap(44, 44, 44))
);

tabbedResultados.addTab(resourceMap.getString("pnlPoblacion.TabConstraints.tabTitle"),
pnlPoblacion); // NOI18N

pnlMuestra.setBackground(resourceMap.getColor("pnlMuestra.background")); // NOI18N
pnlMuestra.setName("pnlMuestra"); // NOI18N

```

```

jsEstratos.setBackground(resourceMap.getColor("jsEstratos.background")); // NOI18N
if(id==0)//estratificado

jsEstratos.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cre
ateEtchedBorder(resourceMap.getColor("jsEstratos.border.border.highlightColor"),
resourceMap.getColor("jsEstratos.border.border.shadowColor")),
resourceMap.getString("jsEstratos.border.title"), javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsEstratos.border.titleFont"),
resourceMap.getColor("jsEstratos.border.titleColor"))); // NOI18N
    jsEstratos.setName("jsEstratos"); // NOI18N

    tblEstratos.setModel(muestras);
    tblEstratos.setName("tblEstratos"); // NOI18N
    jsEstratos.setViewportView(tblEstratos);

    javax.swing.GroupLayout pnlMuestraLayout = new javax.swing.GroupLayout(pnlMuestra);
    pnlMuestra.setLayout(pnlMuestraLayout);
    pnlMuestraLayout.setHorizontalGroup(
        pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlMuestraLayout.createSequentialGroup()
                .addGap(75, 75, 75)
                .addComponent(jsEstratos, javax.swing.GroupLayout.PREFERRED_SIZE, 608,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(79, Short.MAX_VALUE))
            );
    pnlMuestraLayout.setVerticalGroup(
        pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(pnlMuestraLayout.createSequentialGroup()
                .addGap(28, 28, 28)
                .addComponent(jsEstratos, javax.swing.GroupLayout.DEFAULT_SIZE, 352,
Short.MAX_VALUE)
                .addGap(79, Short.MAX_VALUE))
            );

    tabbedResultados.addTab(resourceMap.getString("pnlMuestra.TabConstraints.tabTitle"),
pnlMuestra); // NOI18N

    pnlCalculos.setBackground(resourceMap.getColor("pnlCalculos.background")); // NOI18N
    pnlCalculos.setName("pnlCalculos"); // NOI18N

```

```

jsCalculos.setBackground(resourceMap.getColor("jsCalculos.background")); // NOI18N

jsCalculos.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("jsCalculos.border.border.highlightColor"), resourceMap.getColor("jsCalculos.border.border.shadowColor")), resourceMap.getString("jsCalculos.border.title"), javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT_POSITION, resourceMap.getFont("jsCalculos.border.titleFont"), resourceMap.getColor("jsCalculos.border.titleColor"))); // NOI18N
jsCalculos.setName("jsCalculos"); // NOI18N

tblCalculos.setModel(calculos);
tblCalculos.setName("tblCalculos"); // NOI18N
jsCalculos.setViewportView(tblCalculos);

javax.swing.GroupLayout pnlCalculosLayout = new javax.swing.GroupLayout(pnlCalculos);
pnlCalculos.setLayout(pnlCalculosLayout);
pnlCalculosLayout.setHorizontalGroup(
    pnlCalculosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlCalculosLayout.createSequentialGroup()
            .addGap(44, 44, 44)
            .addComponent(jsCalculos, javax.swing.GroupLayout.PREFERRED_SIZE, 655, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(63, Short.MAX_VALUE))
        );
pnlCalculosLayout.setVerticalGroup(
    pnlCalculosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlCalculosLayout.createSequentialGroup()
            .addGap(59, 59, 59)
            .addComponent(jsCalculos, javax.swing.GroupLayout.PREFERRED_SIZE, 260, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(72, Short.MAX_VALUE))
        );

tabbedResultados.addTab(resourceMap.getString("pnlCalculos.TabConstraints.tabTitle"), pnlCalculos); // NOI18N

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```



```

        .addComponent(lblHeader)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addGap(13, 13, 13)
            .addComponent(btnVolver)
            .addGap(18, 18, 18)
            .addComponent(lblFooter)
            .addGap(18, 18, 18)
            .addComponent(btnGuardar)
            .addContainerGap(30, Short.MAX_VALUE))
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(tabbedResultados, javax.swing.GroupLayout.DEFAULT_SIZE, 767,
Short.MAX_VALUE)
            .addGap(23, 23, 23))
    );
    mainPanelLayout.setVerticalGroup(
        mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addComponent(lblHeader)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(tabbedResultados, javax.swing.GroupLayout.PREFERRED_SIZE, 423,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(37, 37, 37)

        .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(btnGuardar)
            .addComponent(lblFooter)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(btnVolver)))
            .addContainerGap()
    );

    jMenuItem1.setName("jMenuBar1"); // NOI18N

    jMenuItem1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
    jMenuItem1.setName("jMenu1"); // NOI18N

    javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActio
nMap(FinalEstrata.class, this);
    jMenuItem1.setAction(actionMap.get("quit")); // NOI18N

```

```

jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
jMenuItem1.setName("jMenuItem1"); // NOI18N
jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);

pack();
}
private void volver(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void guardar(java.awt.event.ActionEvent evt) {
    Archivos archivo=new Archivos();
    if(indicador==0)//estratificado
    {

    }
    else //ponderado simple
    {
        String tres=tblInfo.getValueAt(4, 1).toString();
        String [] argumentos={"Ponderado Simple",tblResumenPoblacion.getValueAt(0,
1).toString(),nombreArchivo,"TamPob",tblInfo.getValueAt(1,1).toString(),

```

```

        tres};
        archivo.guardar_ponderado(argumentos, tblEstratos.getModel(), tblCalculos.getModel());
    }
}

public static void main(String args[],TableModel muestras, TableModel calculos,TablaResPob
resumenPob,LinkedList te) {
    final String nom_arch=args[0];
    final String modo=args[1];
    final String tipo=args[2];
    final String variable=args[3];
    final String Tm=args[4];
    final TableModel m=muestras;
    final TableModel c=calculos;
    final TablaResPob rp=resumenPob;
    final LinkedList tamEs=te;
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FinalEstrata(nom_arch,modo,tipo,variable,m,c,rp,tamEs,Tm).setVisible(true);
        }
    });
}

public static void main(String args[],TableModel muestras, TableModel calculos,TablaResPob
resumenPob,TableModel info) {
    final String nom_arch=args[0];
    final String modo=args[1];
    final String tipo=args[2];
    final String variable=args[3];
    final TableModel m=muestras;
    final TableModel c=calculos;
    final TablaResPob rp=resumenPob;
    final TableModel info=info;
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new FinalEstrata(nom_arch,modo,tipo,variable,m,c,rp,info).setVisible(true);
        }
    });
}

// Declaración de variables
private javax.swing.JButton btnGuardar;
private javax.swing.JButton btnVolver;

```

```

private javax.swing.JMenu jMenuItem1;
private javax.swing.JMenu jMenuItem2;
private javax.swing.JMenuBar jMenuItemBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JScrollPane jsCalculos;
private javax.swing.JScrollPane jsEstratos;
private javax.swing.JScrollPane jsInfo;
private javax.swing.JScrollPane jsResumen;
private javax.swing.JLabel lblFooter;
private javax.swing.JLabel lblHappyDance;
private javax.swing.JLabel lblHeader;
private javax.swing.JPanel mainPanel;
private javax.swing.JPanel pnlCalculos;
private javax.swing.JPanel pnlMuestra;
private javax.swing.JPanel pnlPoblacion;
private javax.swing.JTabbedPane tabbedResultados;
private javax.swing.JTable tblCalculos;
private javax.swing.JTable tblEstratos;
private javax.swing.JTable tblInfo;
private javax.swing.JTable tblResumenPoblacion;
}

```

### 9.2.10 FinalEstratificado.java

```

/*
 * FinalEstratificado.java
 * Created on 5/01/2011, 11:11:05 AM
 */
package pt02.vistas;
import pt02.archivos.Archivos;
import pt02.modelosTablas.TablaFinal;
import pt02.modelosTablas.TablaMuestra;
import pt02.modelosTablas.TablaResPob;
/**
 *
 * @author XIME
 */
public class FinalEstratificado extends javax.swing.JFrame
{

    public FinalEstratificado(String nomA,String tm,String tp, String met,TablaResPob tr,
TablaMuestra t1, TablaMuestra t2, TablaFinal tf1, TablaFinal tf2, String st)
    {

```

```

initComponents(tr,t1,t2,tf1,tf2,st);
tblResumen.setValueAt(nomA, 0, 1);
tblResumen.setValueAt(tm, 1, 1);
tblResumen.setValueAt(tp, 2, 1);
tblResumen.setValueAt(met, 3, 1);
}

```

```

@SuppressWarnings("unchecked")
private void initComponents() {

```

```

    mainPanel = new javax.swing.JPanel();
    lblHeader = new javax.swing.JLabel();
    tabbedResultados = new javax.swing.JTabbedPane();
    pnlPoblacion = new javax.swing.JPanel();
    lblCorazon = new javax.swing.JLabel();
    jsResumen = new javax.swing.JScrollPane();
    tblResumen = new javax.swing.JTable();
    jsEstrato1 = new javax.swing.JScrollPane();
    tblEstrato1 = new javax.swing.JTable();
    jsEstrato2 = new javax.swing.JScrollPane();
    tblEstrato2 = new javax.swing.JTable();
    lblTipoCodif = new javax.swing.JLabel();
    pnlMuestra = new javax.swing.JPanel();
    jsCalculos = new javax.swing.JScrollPane();
    tblCalculos = new javax.swing.JTable();
    lblCalculos = new javax.swing.JLabel();
    jsCalculos2 = new javax.swing.JScrollPane();
    tblCalculos2 = new javax.swing.JTable();
    lblCalculos2 = new javax.swing.JLabel();
    lblCredito = new javax.swing.JLabel();
    lblRosita = new javax.swing.JLabel();
    btnGuardar = new javax.swing.JButton();
    btnVolver = new javax.swing.JButton();
    lblFooter = new javax.swing.JLabel();
    lblflechalzq = new javax.swing.JLabel();
    lblflechaDer = new javax.swing.JLabel();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenu2 = new javax.swing.JMenu();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

    org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(FinalEstratificado.class);
    setTitle(resourceMap.getString("Form.title")); // NOI18N
    setName("Form"); // NOI18N

    mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
    mainPanel.setName("mainPanel"); // NOI18N

    lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
    lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
    lblHeader.setName("lblHeader"); // NOI18N

    tabbedResultados.setForeground(resourceMap.getColor("tabbedResultados.foreground"));
    tabbedResultados.setFont(resourceMap.getFont("tabbedResultados.font")); // NOI18N
    tabbedResultados.setName("tabbedResultados"); // NOI18N

    pnlPoblacion.setBackground(resourceMap.getColor("pnlPoblacion.background")); // NOI18N
    pnlPoblacion.setName("pnlPoblacion"); // NOI18N

    lblCorazon.setIcon(resourceMap.getIcon("lblCorazon.icon")); // NOI18N
    lblCorazon.setText(resourceMap.getString("lblCorazon.text")); // NOI18N
    lblCorazon.setName("lblCorazon"); // NOI18N

    jsResumen.setBackground(resourceMap.getColor("jsResumen.background")); // NOI18N
    jsResumen.setName("jsResumen"); // NOI18N

    tblResumen.setFont(resourceMap.getFont("tblResumen.font")); // NOI18N
    tblResumen.setForeground(resourceMap.getColor("tblResumen.foreground")); // NOI18N
    tblResumen.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {"Nombre del Archivo", null},
            {"Tamaño Estrata 1", null},
            {"Tamaño Estrata 2", null},
            {"Método Selección", null}
        },
        new String [] {
            "", ""
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false

```

```

};

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}
});
tblResumen.setName("tblResumen"); // NOI18N
jsResumen.setViewportView(tblResumen);

jsEstrato1.setBackground(resourceMap.getColor("jsEstrato1.background")); // NOI18N

jsEstrato1.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cre
ateEtchedBorder(resourceMap.getColor("jsEstrato1.border.border.highlightColor"),
resourceMap.getColor("jsEstrato1.border.border.shadowColor")),
resourceMap.getString("jsEstrato1.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsEstrato1.border.titleFont"),
resourceMap.getColor("jsEstrato1.border.titleColor"))); // NOI18N
jsEstrato1.setName("jsEstrato1"); // NOI18N

tblEstrato1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null},
        {null, null},
        {null, null},
        {null, null}
    },
    new String [] {
        "Title 1", "Title 2"
    }
));
tblEstrato1.setName("tblEstrato1"); // NOI18N
jsEstrato1.setViewportView(tblEstrato1);

jsEstrato2.setBackground(resourceMap.getColor("jsEstrato2.background")); // NOI18N

jsEstrato2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.cre
ateEtchedBorder(resourceMap.getColor("jsEstrato2.border.border.highlightColor"),
resourceMap.getColor("jsEstrato2.border.border.shadowColor")),
resourceMap.getString("jsEstrato2.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,

```

```

javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsEstrato2.border.titleFont"),
resourceMap.getColor("jsEstrato2.border.titleColor")); // NOI18N
jsEstrato2.setName("jsEstrato2"); // NOI18N

tblEstrato2.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null},
        {null, null},
        {null, null},
        {null, null}
    },
    new String [] {
        "Title 1", "Title 2"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblEstrato2.setName("tblEstrato2"); // NOI18N
jsEstrato2.setViewportView(tblEstrato2);

lblTipoCodif.setFont(resourceMap.getFont("lblTipoCodif.font")); // NOI18N
lblTipoCodif.setForeground(resourceMap.getColor("lblTipoCodif.foreground")); // NOI18N
lblTipoCodif.setText(resourceMap.getString("lblTipoCodif.text")); // NOI18N
lblTipoCodif.setName("lblTipoCodif"); // NOI18N

javax.swing.GroupLayout pnlPoblacionLayout = new javax.swing.GroupLayout(pnlPoblacion);
pnlPoblacion.setLayout(pnlPoblacionLayout);
pnlPoblacionLayout.setHorizontalGroup(
    pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPoblacionLayout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(jsEstrato1, javax.swing.GroupLayout.PREFERRED_SIZE, 178,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

```



```

        .addComponent(jsEstrato2, javax.swing.GroupLayout.PREFERRED_SIZE, 180,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 46,
Short.MAX_VALUE)

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
            .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 293,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
            .addComponent(lblCorazon)
            .addGap(45, 45, 45))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
            .addComponent(lblTipoCodif)
            .addGap(145, 145, 145))))
);
pnlPoblacionLayout.setVerticalGroup(
    pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPoblacionLayout.createSequentialGroup()
            .addContainerGap()

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
            .addComponent(lblCorazon)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 52,
Short.MAX_VALUE)
            .addComponent(lblTipoCodif)
            .addGap(18, 18, 18)
            .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(69, 69, 69))
        .addGroup(pnlPoblacionLayout.createSequentialGroup()

.addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)

```

```

        .addComponent(jsEstrato1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 389, Short.MAX_VALUE)
        .addComponent(jsEstrato2, javax.swing.GroupLayout.DEFAULT_SIZE, 389,
Short.MAX_VALUE))
        .addContainerGap()))
);

tabbedResultados.addTab(resourceMap.getString("pnlPoblacion.TabConstraints.tabTitle"),
pnlPoblacion); // NOI18N

pnlMuestra.setBackground(resourceMap.getColor("pnlMuestra.background")); // NOI18N
pnlMuestra.setName("pnlMuestra"); // NOI18N

jsCalculos.setName("jsCalculos"); // NOI18N

tblCalculos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
tblCalculos.setName("tblCalculos"); // NOI18N
jsCalculos.setViewportView(tblCalculos);

lblCalculos.setFont(resourceMap.getFont("lblCalculos.font")); // NOI18N
lblCalculos.setForeground(resourceMap.getColor("lblCalculos.foreground")); // NOI18N
lblCalculos.setText(resourceMap.getString("lblCalculos.text")); // NOI18N
lblCalculos.setName("lblCalculos"); // NOI18N

jsCalculos2.setName("jsCalculos2"); // NOI18N

tblCalculos2.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    }

```



```

        .addGroup(pnlMuestraLayout.createSequentialGroup())
        .addGap(228, 228, 228)
        .addComponent(lblCalculos2))
        .addGroup(pnlMuestraLayout.createSequentialGroup())
        .addContainerGap()
        .addComponent(lblRosita)
        .addContainerGap(660, Short.MAX_VALUE)
    );
    pnlMuestraLayout.setVerticalGroup(
        pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlMuestraLayout.createSequentialGroup())

        .addGroup(pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlMuestraLayout.createSequentialGroup())
        .addGap(28, 28, 28)
        .addComponent(lblCalculos)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jsCalculos, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(55, 55, 55)
        .addComponent(lblCalculos2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jsCalculos2, javax.swing.GroupLayout.PREFERRED_SIZE, 44,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(lblCredito))
        .addGap(69, 69, 69)
        .addComponent(lblRosita)
        .addContainerGap(32, Short.MAX_VALUE)
    );

    tabbedResultados.addTab(resourceMap.getString("pnlMuestra.TabConstraints.tabTitle"),
pnlMuestra); // NOI18N

    btnGuardar.setFont(resourceMap.getFont("btnGuardar.font")); // NOI18N
    btnGuardar.setForeground(resourceMap.getColor("btnGuardar.foreground")); // NOI18N
    btnGuardar.setText(resourceMap.getString("btnGuardar.text")); // NOI18N
    btnGuardar.setName("btnGuardar"); // NOI18N
    btnGuardar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            guardar(evt);
        }
    });

```



```

        .addComponent(lblflechalzq)
        .addGap(13, 13, 13)
        .addComponent(btnVolver)
        .addGap(109, 109, 109)
        .addComponent(lblFooter)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 121,
Short.MAX_VALUE)
        .addComponent(lblflechaDer)
        .addGap(18, 18, 18)
        .addComponent(btnGuardar)))
        .addGap(30, 30, 30)))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
mainPanelLayout.setVerticalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addComponent(lblHeader)
        .addGap(18, 18, 18)
        .addComponent(tabbedResultados, javax.swing.GroupLayout.PREFERRED_SIZE, 440,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(lblflechalzq)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(btnGuardar)
        .addComponent(lblflechaDer))

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblFooter)
        .addComponent(btnVolver))))
);

jMenuBar1.setName("jMenuBar1"); // NOI18N

jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActio
nMap(FinalEstratificado.class, this);

```

```

jMenuItem1.setAction(actionMap.get("quit")); // NOI18N
jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
jMenuItem1.setName("jMenuItem1"); // NOI18N
jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(layout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE, 800,
javax.swing.GroupLayout.PREFERRED_SIZE)
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(layout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        );
pack();
}

private void btnVolverActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void guardar(java.awt.event.ActionEvent evt) {

String arg[]=new String [6];
arg[0]="Estratificado";
arg[1]="Por Promedio";
arg[2]=tblResumen.getValueAt(0,1).toString();

```

```

    arg[3]=tblResumen.getValueAt(1,1).toString();
    arg[4]=tblResumen.getValueAt(2,1).toString();
    arg[5]=tblResumen.getValueAt(3,1).toString();
    Archivos archivo=new Archivos();
    archivo.guardar(arg, tblEstrato1.getModel(), tblEstrato2.getModel(), tblCalculos.getModel(),
tblCalculos2.getModel());
}
private void initComponents(TablaResPob tr, TablaMuestra t1, TablaMuestra t2, TablaFinal tf1,
TablaFinal tf2, String st)
{
    mainPanel = new javax.swing.JPanel();
    lblHeader = new javax.swing.JLabel();
    tabbedResultados = new javax.swing.JTabbedPane();
    pnlPoblacion = new javax.swing.JPanel();
    lblCorazon = new javax.swing.JLabel();
    jsResumen = new javax.swing.JScrollPane();
    tblResumen = new javax.swing.JTable();
    jsEstrato1 = new javax.swing.JScrollPane();
    tblEstrato1 = new javax.swing.JTable();
    jsEstrato2 = new javax.swing.JScrollPane();
    tblEstrato2 = new javax.swing.JTable();
    lblTipoCodif = new javax.swing.JLabel();
    pnlMuestra = new javax.swing.JPanel();
    jsCalculos = new javax.swing.JScrollPane();
    tblCalculos = new javax.swing.JTable();
    lblCalculos = new javax.swing.JLabel();
    jsCalculos2 = new javax.swing.JScrollPane();
    tblCalculos2 = new javax.swing.JTable();
    lblCalculos2 = new javax.swing.JLabel();
    lblCredito = new javax.swing.JLabel();
    lblRosita = new javax.swing.JLabel();
    btnGuardar = new javax.swing.JButton();
    btnVolver = new javax.swing.JButton();
    lblFooter = new javax.swing.JLabel();
    lblflechalzq = new javax.swing.JLabel();
    lblflechaDer = new javax.swing.JLabel();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenu2 = new javax.swing.JMenu();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```



```

    org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(FinalEstratificado.class);
    setTitle(resourceMap.getString("Form.title")); // NOI18N
    setName("Form"); // NOI18N

    mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
    mainPanel.setName("mainPanel"); // NOI18N

    lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
    lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
    lblHeader.setName("lblHeader"); // NOI18N

    tabbedResultados.setForeground(resourceMap.getColor("tabbedResultados.foreground"));
    tabbedResultados.setFont(resourceMap.getFont("tabbedResultados.font")); // NOI18N
    tabbedResultados.setName("tabbedResultados"); // NOI18N

    pnlPoblacion.setBackground(resourceMap.getColor("pnlPoblacion.background")); // NOI18N
    pnlPoblacion.setName("pnlPoblacion"); // NOI18N

    lblCorazon.setIcon(resourceMap.getIcon("lblCorazon.icon")); // NOI18N
    lblCorazon.setText(resourceMap.getString("lblCorazon.text")); // NOI18N
    lblCorazon.setName("lblCorazon"); // NOI18N

    jsResumen.setBackground(resourceMap.getColor("jsResumen.background")); // NOI18N
    jsResumen.setName("jsResumen"); // NOI18N

    tblResumen.setFont(resourceMap.getFont("tblResumen.font")); // NOI18N
    tblResumen.setForeground(resourceMap.getColor("tblResumen.foreground")); // NOI18N
    tblResumen.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {"Nombre del Archivo", null},
            {"Tamaño Estrata 1", null},
            {"Tamaño Estrata 2", null},
            {"Método Selección", null}
        },
        new String [] {
            "", ""
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false

```

```

};

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}
});
tblResumen.setName("tblResumen"); // NOI18N
jsResumen.setViewportView(tblResumen);

jsEstrato1.setBackground(resourceMap.getColor("jsEstrato1.background")); // NOI18N

jsEstrato1.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("jsEstrato1.border.border.highlightColor"), resourceMap.getColor("jsEstrato1.border.border.shadowColor")), resourceMap.getString("jsEstrato1.border.title"), javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT_POSITION, resourceMap.getFont("jsEstrato1.border.titleFont"), resourceMap.getColor("jsEstrato1.border.titleColor"))); // NOI18N
jsEstrato1.setName("jsEstrato1"); // NOI18N

tblEstrato1.setModel(t1);
tblEstrato1.setName("tblEstrato1"); // NOI18N
jsEstrato1.setViewportView(tblEstrato1);

jsEstrato2.setBackground(resourceMap.getColor("jsEstrato2.background")); // NOI18N

jsEstrato2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(resourceMap.getColor("jsEstrato2.border.border.highlightColor"), resourceMap.getColor("jsEstrato2.border.border.shadowColor")), resourceMap.getString("jsEstrato2.border.title"), javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT_POSITION, resourceMap.getFont("jsEstrato2.border.titleFont"), resourceMap.getColor("jsEstrato2.border.titleColor"))); // NOI18N
jsEstrato2.setName("jsEstrato2"); // NOI18N

tblEstrato2.setModel(t2);
tblEstrato2.setName("tblEstrato2"); // NOI18N
jsEstrato2.setViewportView(tblEstrato2);

lblTipoCodif.setFont(resourceMap.getFont("lblTipoCodif.font")); // NOI18N

```

```

lblTipoCodif.setForeground(resourceMap.getColor("lblTipoCodif.foreground")); // NOI18N
lblTipoCodif.setText(st); // NOI18N
lblTipoCodif.setName("lblTipoCodif"); // NOI18N

javax.swing.GroupLayout pnlPoblacionLayout = new javax.swing.GroupLayout(pnlPoblacion);
pnlPoblacion.setLayout(pnlPoblacionLayout);
pnlPoblacionLayout.setHorizontalGroup(
    pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPoblacionLayout.createSequentialGroup()
            .addComponent(jsEstrato1, javax.swing.GroupLayout.PREFERRED_SIZE, 178,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jsEstrato2, javax.swing.GroupLayout.PREFERRED_SIZE, 180,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 46,
Short.MAX_VALUE)
        .addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
                .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 293,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
                    .addComponent(lblCorazon)
                    .addGap(45, 45, 45))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnlPoblacionLayout.createSequentialGroup()
                    .addComponent(lblTipoCodif)
                    .addGap(145, 145, 145))))
        );
pnlPoblacionLayout.setVerticalGroup(
    pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlPoblacionLayout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addGroup(pnlPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jsEstrato1)
                .addComponent(jsEstrato2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 46,
Short.MAX_VALUE)
                .addComponent(jsResumen)
                .addComponent(lblCorazon)
                .addComponent(lblTipoCodif)
            )
        );

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnIPoblacionLayout.createSequentialGroup()
        .addComponent(lblCorazon)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 52,
Short.MAX_VALUE)
        .addComponent(lblTipoCodif)
        .addGap(18, 18, 18)
        .addComponent(jsResumen, javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(69, 69, 69))
        .addGroup(pnIPoblacionLayout.createSequentialGroup()

.addGroup(pnIPoblacionLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addComponent(jsEstrato1, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 389, Short.MAX_VALUE)
        .addComponent(jsEstrato2, javax.swing.GroupLayout.DEFAULT_SIZE, 389,
Short.MAX_VALUE))
        .addContainerGap()))
);

tabbedResultados.addTab(resourceMap.getString("pnIPoblacion.TabConstraints.tabTitle"),
pnIPoblacion); // NOI18N

pnIMuestra.setBackground(resourceMap.getColor("pnIMuestra.background")); // NOI18N
pnIMuestra.setName("pnIMuestra"); // NOI18N

jsCalculos.setName("jsCalculos"); // NOI18N

tblCalculos.setModel(tf1);
tblCalculos.setName("tblCalculos"); // NOI18N
jsCalculos.setViewportViewView(tblCalculos);

lblCalculos.setFont(resourceMap.getFont("lblCalculos.font")); // NOI18N
lblCalculos.setForeground(resourceMap.getColor("lblCalculos.foreground")); // NOI18N
lblCalculos.setText(resourceMap.getString("lblCalculos.text")); // NOI18N
lblCalculos.setName("lblCalculos"); // NOI18N

jsCalculos2.setName("jsCalculos2"); // NOI18N

tblCalculos2.setModel(tf2);
tblCalculos2.setName("tblCalculos2"); // NOI18N

```



```

        .addContainerGap(660, Short.MAX_VALUE)
    );
    pnlMuestraLayout.setVerticalGroup(
        pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlMuestraLayout.createSequentialGroup())

    .addGroup(pnlMuestraLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlMuestraLayout.createSequentialGroup()
            .addGap(28, 28, 28)
            .addComponent(lblCalculos)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jsCalculos, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(55, 55, 55)
            .addComponent(lblCalculos2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jsCalculos2, javax.swing.GroupLayout.PREFERRED_SIZE, 44,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(lblCredito)
            .addGap(69, 69, 69)
            .addComponent(lblRosita)
            .addContainerGap(32, Short.MAX_VALUE)
        );

    tabbedResultados.addTab(resourceMap.getString("pnlMuestra.TabConstraints.tabTitle"),
pnlMuestra); // NOI18N

    btnGuardar.setFont(resourceMap.getFont("btnGuardar.font")); // NOI18N
    btnGuardar.setForeground(resourceMap.getColor("btnGuardar.foreground")); // NOI18N
    btnGuardar.setText(resourceMap.getString("btnGuardar.text")); // NOI18N
    btnGuardar.setName("btnGuardar"); // NOI18N
    btnGuardar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            guardar(evt);
        }
    });

    btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
    btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
    btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
    btnVolver.setName("btnVolver"); // NOI18N
    btnVolver.addActionListener(new java.awt.event.ActionListener() {

```

```

    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnVolverActionPerformed(evt);
    }
});

lblFooter.setIcon(resourceMap.getIcon("lblFooter.icon")); // NOI18N
lblFooter.setText(resourceMap.getString("lblFooter.text")); // NOI18N
lblFooter.setName("lblFooter"); // NOI18N

lblflechalzq.setIcon(resourceMap.getIcon("lblflechalzq.icon")); // NOI18N
lblflechalzq.setText(resourceMap.getString("lblflechalzq.text")); // NOI18N
lblflechalzq.setName("lblflechalzq"); // NOI18N

lblflechaDer.setIcon(resourceMap.getIcon("lblflechaDer.icon")); // NOI18N
lblflechaDer.setText(resourceMap.getString("lblflechaDer.text")); // NOI18N
lblflechaDer.setName("lblflechaDer"); // NOI18N

javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(mainPanelLayout.createSequentialGroup()
            .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
                false)
                .addComponent(lblHeader)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                    mainPanelLayout.createSequentialGroup()
                        .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                            .addGroup(mainPanelLayout.createSequentialGroup()
                                .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
                                        mainPanelLayout.createSequentialGroup()
                                            .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                                .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
                                                    mainPanelLayout.createSequentialGroup()
                                                        .addGap(30, 30, 30)
                                                        .addComponent(tabbedResultados, javax.swing.GroupLayout.PREFERRED_SIZE,
                                                            740, javax.swing.GroupLayout.PREFERRED_SIZE))
                                                    .addGroup(mainPanelLayout.createSequentialGroup()
                                                        .addComponent(lblflechalzq)
                                                        .addGap(13, 13, 13)
                                                        .addComponent(btnVolver)
                                                        .addGap(109, 109, 109)
                                                        .addComponent(lblFooter)

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 121,
Short.MAX_VALUE)
        .addComponent(lblflechaDer)
        .addGap(18, 18, 18)
        .addComponent(btnGuardar)))
    .addGap(30, 30, 30)))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
mainPanelLayout.setVerticalGroup(
    mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(mainPanelLayout.createSequentialGroup()
        .addComponent(lblHeader)
        .addGap(18, 18, 18)
        .addComponent(tabbedResultados, javax.swing.GroupLayout.PREFERRED_SIZE, 440,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(lblflechalzq)

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(btnGuardar)
        .addComponent(lblflechaDer))

    .addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(lblFooter)
        .addComponent(btnVolver))))
);

jMenuBar1.setName("jMenuBar1"); // NOI18N

jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActio
nMap(FinalEstratificado.class, this);
jMenuItem1.setAction(actionMap.get("quit")); // NOI18N
jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
jMenuItem1.setName("jMenuItem1"); // NOI18N
jMenu1.add(jMenuItem1);

```



```

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE, 800,
                    javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap())
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(mainPanel, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap())
);
pack();
}

public static void main(String args[],int tam1,int tam2, TablaMuestra muestra1,TablaMuestra
muestra2,TablaFinal f1,TablaFinal f2,TablaResPob trp) {
    final String nomArch=args[0];
    final String tamMues1=args[1];
    final String tamMues2=args[2];
    final String modo=args[3];
    final String tipoCodif=args[4];
    final TablaResPob tr=trp;
    final TablaMuestra tm1=muestra1;
    final TablaMuestra tm2=muestra2;
    final TablaFinal tf1=f1;
    final TablaFinal tf2=f2;
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

```

```

        new
FinalEstratificado(nomArch,tamMues1,tamMues2,modo,tr,tm1,tm2,tf1,tf2,tipocodif).setVisible(tr
ue);
    }
    });
}

```

```

// Declaración de Variables
private javax.swing.JButton btnGuardar;
private javax.swing.JButton btnVolver;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JScrollPane jsCalculos;
private javax.swing.JScrollPane jsCalculos2;
private javax.swing.JScrollPane jsEstrato1;
private javax.swing.JScrollPane jsEstrato2;
private javax.swing.JScrollPane jsResumen;
private javax.swing.JLabel lblCalculos;
private javax.swing.JLabel lblCalculos2;
private javax.swing.JLabel lblCredito;
private javax.swing.JLabel lblCorazon;
private javax.swing.JLabel lblFooter;
private javax.swing.JLabel lblHeader;
private javax.swing.JLabel lblRosita;
private javax.swing.JLabel lblTipoCodif;
private javax.swing.JLabel lblflechaDer;
private javax.swing.JLabel lblflechalzq;
private javax.swing.JPanel mainPanel;
private javax.swing.JPanel pnlMuestra;
private javax.swing.JPanel pnlPoblacion;
private javax.swing.JTabbedPane tabbedResultados;
private javax.swing.JTable tblCalculos;
private javax.swing.JTable tblCalculos2;
private javax.swing.JTable tblEstrato1;
private javax.swing.JTable tblEstrato2;
private javax.swing.JTable tblResumen;
}

```

### 9.2.11 Politapico.java

```
/*
```

```

* Politapico.java
* Created on 22/01/2011, 03:37:54 PM
*/
package pt02.vistas;
import java.util.LinkedList;
import javax.swing.JOptionPane;
import javax.swing.table.TableModel;
import pt02.archivos.Archivos;
import pt02.kernel.Muestreo;
import pt02.kernel.Resultados;
import pt02.modelosTablas.TablaLimites;
/*
* @author XiMe!!!
*/
public class Politapico extends javax.swing.JFrame {
    public String nombreArchivo;
    public int id=-1;
    public LinkedList tokensEtapa1;
    public LinkedList tokensEtapa2;

    /** Constructor de Politapico cuando carga la primera vez */
    public Politapico() {
        initComponents();
        pnlSelecciones.setVisible(false);
        jsDatos.setVisible(false);
    }
    /**Constructor de politápico cuando regresa de codificarse*/
    public Politapico(TableModel dts,LinkedList E1,LinkedList E2) {
        initComponents();
        this.tokensEtapa1=E1;
        this.tokensEtapa2=E2;
        tblDatos.setModel(dts);
        Object[] noms=new Object[tblDatos.getColumnCount()];
        for(int i=0;i<tblDatos.getColumnCount();i++)
        {
            noms[i]=tblDatos.getColumnName(i);
        }
        comboSelCod1.setModel(new javax.swing.DefaultComboBoxModel(noms));
        comboSelCod2.setModel(new javax.swing.DefaultComboBoxModel(noms));
        comboSelecVar.setModel(new javax.swing.DefaultComboBoxModel(noms));
        lblCargar.setVisible(false);
        btnCargar.setVisible(false);
    }
}

```

```

    btnSiguiete.setEnabled(true);
}
@SuppressWarnings("unchecked")
private void initComponents() {

    pnlPrincipal = new javax.swing.JPanel();
    lblHeader = new javax.swing.JLabel();
    jsDatos = new javax.swing.JScrollPane();
    tblDatos = new javax.swing.JTable();
    btnVolver = new javax.swing.JButton();
    lblFooter = new javax.swing.JLabel();
    btnSiguiete = new javax.swing.JButton();
    pnlSelecciones = new javax.swing.JPanel();
    lblSelVar = new javax.swing.JLabel();
    comboSelecVar = new javax.swing.JComboBox();
    lblSelCodigo1 = new javax.swing.JLabel();
    comboSelCod1 = new javax.swing.JComboBox();
    lblSelCod2 = new javax.swing.JLabel();
    comboSelCod2 = new javax.swing.JComboBox();
    btnCargar = new javax.swing.JButton();
    lblCargar = new javax.swing.JLabel();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu1 = new javax.swing.JMenu();
    jMenuItem1 = new javax.swing.JMenuItem();
    jMenuItem2 = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(Politapico.class);
    setTitle(resourceMap.getString("Form.title")); // NOI18N
    setName("Form"); // NOI18N

    pnlPrincipal.setBackground(resourceMap.getColor("pnlPrincipal.background")); // NOI18N
    pnlPrincipal.setName("pnlPrincipal"); // NOI18N

    lblHeader.setIcon(resourceMap.getIcon("lblHeader.icon")); // NOI18N
    lblHeader.setText(resourceMap.getString("lblHeader.text")); // NOI18N
    lblHeader.setName("lblHeader"); // NOI18N

    jsDatos.setBackground(resourceMap.getColor("jsDatos.background")); // NOI18N

```

```

jsDatos.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.create
EtchedBorder(resourceMap.getColor("jsDatos.border.border.highlightColor"),
resourceMap.getColor("jsDatos.border.border.shadowColor")),
resourceMap.getString("jsDatos.border.title"),
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION,
resourceMap.getFont("jsDatos.border.titleFont"),
resourceMap.getColor("jsDatos.border.titleColor"))); // NOI18N
    jsDatos.setName("jsDatos"); // NOI18N

```

```

tblDatos.setFont(resourceMap.getFont("tblDatos.font")); // NOI18N
tblDatos.setForeground(resourceMap.getColor("tblDatos.foreground")); // NOI18N
tblDatos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
tblDatos.setGridColor(resourceMap.getColor("tblDatos.gridColor")); // NOI18N
tblDatos.setName("tblDatos"); // NOI18N
tblDatos.setSelectionBackground(resourceMap.getColor("tblDatos.selectionBackground"));
jsDatos.setViewPortView(tblDatos);

```

```

btnVolver.setFont(resourceMap.getFont("btnVolver.font")); // NOI18N
btnVolver.setForeground(resourceMap.getColor("btnVolver.foreground")); // NOI18N
btnVolver.setText(resourceMap.getString("btnVolver.text")); // NOI18N
btnVolver.setName("btnVolver"); // NOI18N
btnVolver.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        volver(evt);
    }
});

```

```

lblFooter.setIcon(resourceMap.getIcon("lblFooter.icon")); // NOI18N
lblFooter.setText(resourceMap.getString("lblFooter.text")); // NOI18N
lblFooter.setName("lblFooter"); // NOI18N

```

```

btnSiguiente.setFont(resourceMap.getFont("btnSiguiente.font")); // NOI18N
btnSiguiente.setForeground(resourceMap.getColor("btnSiguiente.foreground")); // NOI18N
btnSiguiente.setText(resourceMap.getString("btnSiguiente.text")); // NOI18N
btnSiguiente.setEnabled(false);
btnSiguiente.setName("btnSiguiente"); // NOI18N
btnSiguiente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        siguiente(evt);
    }
});

pnlSelecciones.setBackground(resourceMap.getColor("pnlSelecciones.background"));
pnlSelecciones.setName("pnlSelecciones"); // NOI18N

lblSelVar.setBackground(resourceMap.getColor("lblSelVar.background")); // NOI18N
lblSelVar.setFont(resourceMap.getFont("lblSelVar.font")); // NOI18N
lblSelVar.setForeground(resourceMap.getColor("lblSelVar.foreground")); // NOI18N
lblSelVar.setText(resourceMap.getString("lblSelVar.text")); // NOI18N
lblSelVar.setName("lblSelVar"); // NOI18N

comboSelecVar.setFont(resourceMap.getFont("comboSelecVar.font")); // NOI18N
comboSelecVar.setForeground(resourceMap.getColor("comboSelecVar.foreground"));
comboSelecVar.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1",
"Item 2", "Item 3", "Item 4" }));
comboSelecVar.setName("comboSelecVar"); // NOI18N

lblSelCodigo1.setBackground(resourceMap.getColor("lblSelCodigo1.background")); // NOI18N
lblSelCodigo1.setFont(resourceMap.getFont("lblSelCodigo1.font")); // NOI18N
lblSelCodigo1.setForeground(resourceMap.getColor("lblSelCodigo1.foreground")); // NOI18N
lblSelCodigo1.setText(resourceMap.getString("lblSelCodigo1.text")); // NOI18N
lblSelCodigo1.setName("lblSelCodigo1"); // NOI18N

comboSelCod1.setFont(resourceMap.getFont("comboSelCod1.font")); // NOI18N
comboSelCod1.setForeground(resourceMap.getColor("comboSelCod1.foreground"));
comboSelCod1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1",
"Item 2", "Item 3", "Item 4" }));
comboSelCod1.setName("comboSelCod1"); // NOI18N

lblSelCod2.setBackground(resourceMap.getColor("lblSelCod2.background")); // NOI18N
lblSelCod2.setFont(resourceMap.getFont("lblSelCod2.font")); // NOI18N
lblSelCod2.setForeground(resourceMap.getColor("lblSelCod2.foreground")); // NOI18N

```

```

lblSelCod2.setText(resourceMap.getString("lblSelCod2.text")); // NOI18N
lblSelCod2.setName("lblSelCod2"); // NOI18N

comboSelCod2.setFont(resourceMap.getFont("comboSelCod2.font")); // NOI18N
comboSelCod2.setForeground(resourceMap.getColor("comboSelCod2.foreground"));
comboSelCod2.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1",
"Item 2", "Item 3", "Item 4" }));
comboSelCod2.setName("comboSelCod2"); // NOI18N

javax.swing.GroupLayout pnlSeleccionesLayout = new
javax.swing.GroupLayout(pnlSelecciones);
pnlSelecciones.setLayout(pnlSeleccionesLayout);
pnlSeleccionesLayout.setHorizontalGroup(
    pnlSeleccionesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSeleccionesLayout.createSequentialGroup()
            .addGap(19, 19, 19)
            .addGroup(pnlSeleccionesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(lblSelVar)
                .addComponent(lblSelCodigo1)
                .addComponent(lblSelCod2)
                .addComponent(comboSelCod2, javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(comboSelCod1, javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(comboSelecVar, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(19, Short.MAX_VALUE))
        );
pnlSeleccionesLayout.setVerticalGroup(
    pnlSeleccionesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(pnlSeleccionesLayout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addGroup(pnlSeleccionesLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(lblSelVar)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(comboSelecVar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(lblSelCodigo1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            )
        )
    );

```

```

        .addComponent(comboSelCod1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(lblSelCod2)
        .addGap(18, 18, 18)
        .addComponent(comboSelCod2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    btnCargar.setFont(resourceMap.getFont("btnCargar.font")); // NOI18N
    btnCargar.setForeground(resourceMap.getColor("btnCargar.foreground")); // NOI18N
    btnCargar.setText(resourceMap.getString("btnCargar.text")); // NOI18N
    btnCargar.setName("btnCargar"); // NOI18N
    btnCargar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            cargar_archivo(evt);
        }
    });

    lblCargar.setFont(resourceMap.getFont("lblCargar.font")); // NOI18N
    lblCargar.setForeground(resourceMap.getColor("lblCargar.foreground")); // NOI18N
    lblCargar.setText(resourceMap.getString("lblCargar.text")); // NOI18N
    lblCargar.setName("lblCargar"); // NOI18N

    javax.swing.GroupLayout pnlPrincipalLayout = new javax.swing.GroupLayout(pnlPrincipal);
    pnlPrincipal.setLayout(pnlPrincipalLayout);
    pnlPrincipalLayout.setHorizontalGroup(
        pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(pnlPrincipalLayout.createSequentialGroup()
            .addGroup(pnlPrincipalLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(pnlPrincipalLayout.createSequentialGroup()
                    .addGroup(pnlPrincipalLayout.createSequentialGroup()
                        .addComponent(btnVolver)
                        .addGap(47, 47, 47)
                        .addComponent(lblFooter))
                    .addGroup(pnlPrincipalLayout.createSequentialGroup()
                        .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 444,
javax.swing.GroupLayout.PREFERRED_SIZE)

```



```

.addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlPrincipalLayout.createSequentialGroup()
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(btnCargar)
    .addComponent(lblCargar)))
.addGroup(pnlPrincipalLayout.createSequentialGroup()
    .addGap(47, 47, 47)

.addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(btnSiguiete)
    .addComponent(pnlSelecciones, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))))))
    .addGap(221, 221, 221))
    .addComponent(lblHeader, javax.swing.GroupLayout.Alignment.LEADING)
);
pnlPrincipalLayout.setVerticalGroup(
    pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(pnlPrincipalLayout.createSequentialGroup()
        .addComponent(lblHeader)

.addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
pnlPrincipalLayout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(jsDatos, javax.swing.GroupLayout.PREFERRED_SIZE, 444,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(pnlPrincipalLayout.createSequentialGroup()
        .addGap(33, 33, 33)
        .addComponent(pnlSelecciones, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 156,
Short.MAX_VALUE)
        .addComponent(lblCargar)
        .addGap(18, 18, 18)
        .addComponent(btnCargar)))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 18,
Short.MAX_VALUE)

.addGroup(pnlPrincipalLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

```

```

        .addComponent(btnSiguiente)
        .addComponent(lblFooter)
        .addComponent(btnVolver)
        .addGap(21, 21, 21))
);

jMenuBar1.setName("jMenuBar1"); // NOI18N

jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

    javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActionMap(Politapico.class, this);
    jMenuItem1.setAction(actionMap.get("quit")); // NOI18N
    jMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
    jMenuItem1.setName("jMenuItem1"); // NOI18N
    jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlPrincipal, javax.swing.GroupLayout.PREFERRED_SIZE, 799,
javax.swing.GroupLayout.PREFERRED_SIZE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(pnlPrincipal, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    pack();
}

```

```

private void cargar_archivo(java.awt.event.ActionEvent evt) {
    Archivos archivo=new Archivos();
    Object[] parametros=new Object[2];
    parametros[0]="Cargar de Archivo";
    parametros[1]="Cargar de Archivo y Codificar";
    Object sel="Seleccione una Opción";
    int opcion=JOptionPane.showOptionDialog(null,sel, "Indique si ya tiene sus variables
codificadas",JOptionPane.YES_NO_CANCEL_OPTION,JOptionPane.QUESTION_MESSAGE,null,para
metros,parametros[0]);
    if(opcion==0)
    {
        nombreArchivo=archivo.manejar_archivo(tblDatos, comboSelecVar, comboSelCod1,
comboSelCod2);
        lblCargar.setVisible(false);
        jsDatos.setVisible(true);
        pnlSelecciones.setVisible(true);
        id=1;
        btnSiguiente.setEnabled(true);
    }
    else
    {
        nombreArchivo=archivo.manejar_archivo(tblDatos, comboSelecVar);
        lblCargar.setVisible(false);
        jsDatos.setVisible(true);
        pnlSelecciones.setVisible(true);
        lblSelCodigo1.setVisible(false);
        comboSelCod1.setVisible(false);
        lblSelCod2.setVisible(false);
        comboSelCod2.setVisible(false);
        lblSelVar.setText("2) Seleccione la variable a codificar");
        id=0;
        btnSiguiente.setEnabled(true);
    }
}
}

```

```

private void siguiente(java.awt.event.ActionEvent evt) {
    switch(id)
    {
        case 0:
            Resultados res=new Resultados();

```

```

        String[]
        argumentos={"Manual",nombreArchivo,comboSelecVar.getSelectedItem().toString(),"500"};
        LinkedList
        datos=res.lista_numeros(tblDatos.getModel(),comboSelecVar.getSelectedIndex());
        LinkedList[] codif_manual=new LinkedList[3];
        LinkedList nombres=new LinkedList();
        LinkedList aux=new LinkedList();
        for(int r=0;r<datos.size();r++)
            aux.add(null);
        nombres.add(comboSelecVar.getSelectedItem().toString());
        nombres.add("Etapa 1");
        nombres.add("Etapa 2");
        codif_manual[0]=datos;
        codif_manual[1]=aux;
        codif_manual[2]=aux;
        TablaLimites ti=new TablaLimites(nombres,codif_manual);
        Codificacion.main(argumentos,ti,null,datos);
        this.dispose();
        break;

    case 1:
        //empezar validaciones

        if(comboSelecVar.getSelectedItem().toString().equals(comboSelCod1.getSelectedItem().toString())
        ||
        comboSelecVar.getSelectedItem().toString().equals(comboSelCod2.getSelectedItem().toString()))
            {
                JOptionPane.showMessageDialog(null, "Los códigos no pueden ser iguales");
                break;
            }
            else
            {
                Muestreo politapico=new
                Muestreo(comboSelecVar,comboSelCod1,comboSelCod2,tblDatos.getModel());
                politapico.empezar_metodo(4, nombreArchivo, nombreArchivo);
                break;
            }

        default:break;
    }
}

```

```

private void volver(java.awt.event.ActionEvent evt) {
    this.dispose();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Politapico().setVisible(true);
        }
    });
}

public static void main(String args[],TableModel datos,LinkedList[]t) {
    final TableModel d=datos;
    final LinkedList etapa1=t[0];
    final LinkedList etapa2=t[1];
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Politapico(d,etapa1,etapa2).setVisible(true);
        }
    });
}

// Declaración de variables
private javax.swing.JButton btnCargar;
private javax.swing.JButton btnSiguiente;
private javax.swing.JButton btnVolver;
private javax.swing.JComboBox comboSelCod1;
private javax.swing.JComboBox comboSelCod2;
private javax.swing.JComboBox comboSelecVar;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JScrollPane jsDatos;
private javax.swing.JLabel lblCargar;
private javax.swing.JLabel lblFooter;
private javax.swing.JLabel lblHeader;
private javax.swing.JLabel lblSelCod2;
private javax.swing.JLabel lblSelCodigo1;
private javax.swing.JLabel lblSelVar;
private javax.swing.JPanel pnlPrincipal;

```

```

private javax.swing.JPanel pnlSelecciones;
private javax.swing.JTable tblDatos;
}

```

### 9.2.12 Ayuda.java

```

/*
 * Ayuda.java
 * Created on 17/11/2010, 09:18:35 PM
 */

package pt02.vistas;
import org.jdesktop.application.Action;
import pt02.archivos.Archivos;
/**
 *
 * @author Xime
 */
public class Ayuda extends javax.swing.JFrame {
    private String nombreArchivo;

    public Ayuda(String nombre)
    {
        this.nombreArchivo=nombre;
        initComponents();
        Archivos archivo=new Archivos();
        archivo.abrir(nombreArchivo,txtArea);
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {

        mainPanel = new javax.swing.JPanel();
        jsTexto = new javax.swing.JScrollPane();
        txtArea = new javax.swing.JTextArea();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenuItem1 = new javax.swing.JMenuItem();
        jMenuItem2 = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

```

```

    org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getResou
rceMap(Ayuda.class);
    setTitle(resourceMap.getString("Form.title")); // NOI18N
    setLocationByPlatform(true);
    setName("Form"); // NOI18N

    mainPanel.setBackground(resourceMap.getColor("mainPanel.background")); // NOI18N
    mainPanel.setName("mainPanel"); // NOI18N

    jsTexto.setName("jsTexto"); // NOI18N

    txtArea.setColumns(20);
    txtArea.setEditable(false);
    txtArea.setFont(resourceMap.getFont("txtArea.font")); // NOI18N
    txtArea.setRows(5);

txtArea.setBorder(javax.swing.BorderFactory.createLineBorder(resourceMap.getColor("txtArea.bo
rder.lineColor"), 2)); // NOI18N
    txtArea.setName("txtArea"); // NOI18N
    jsTexto.setViewportView(txtArea);

    javax.swing.GroupLayout mainPanelLayout = new javax.swing.GroupLayout(mainPanel);
    mainPanel.setLayout(mainPanelLayout);
    mainPanelLayout.setHorizontalGroup(
        mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addGap(127, 127, 127)
                .addComponent(jsTexto, javax.swing.GroupLayout.PREFERRED_SIZE, 387,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, Short.MAX_VALUE))
            );
    mainPanelLayout.setVerticalGroup(
        mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(mainPanelLayout.createSequentialGroup()
                .addGap(44, 44, 44)
                .addComponent(jsTexto, javax.swing.GroupLayout.PREFERRED_SIZE, 323,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(39, Short.MAX_VALUE))
            );

    jMenuBar1.setName("jMenuBar1"); // NOI18N

```

```

jMenu1.setText(resourceMap.getString("jMenu1.text")); // NOI18N
jMenu1.setName("jMenu1"); // NOI18N

    javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(pt02.vistas.PT02App.class).getContext().getActionMap(Ayuda.class, this);
    JMenuItem1.setAction(actionMap.get("cerrar_ayuda")); // NOI18N
    JMenuItem1.setText(resourceMap.getString("jMenuItem1.text")); // NOI18N
    JMenuItem1.setName("jMenuItem1"); // NOI18N
    jMenu1.add(jMenuItem1);

jMenuBar1.add(jMenu1);

jMenu2.setText(resourceMap.getString("jMenu2.text")); // NOI18N
jMenu2.setName("jMenu2"); // NOI18N
jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(mainPanel, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
);
pack();
}

public static void main(String args[])
{
    final String nomb;
    nomb=args[0];
    java.awt.EventQueue.invokeLater(new Runnable()
    {

```



```

        public void run()
        {
            new Ayuda(nomb).setVisible(true);
        }
    });
}

@Action
public void cerrar_ayuda() {
    this.dispose();
}

// Declaración de variables
private javax.swing.JMenu jMenuItem1;
private javax.swing.JMenu jMenuItem2;
private javax.swing.JMenuBar jMenuItemBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JScrollPane jsTexto;
private javax.swing.JPanel mainPanel;
private javax.swing.JTextArea txtArea;
}

```

## 9.3 Paquete Archivos

### 9.3.1 Archivos.java

```

/*
 * Esta clase contiene métodos para apertura y lectura del archivo de datos proporcionado por el
 usuario
 * y para guardar la muestra generada por el sistema
 */
package pt02.archivos;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.util.Date;
import java.util.LinkedList;
import java.util.Scanner;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;

```

```

import javax.swing.JTextArea;
import javax.swing.table.TableModel;
import pt02.modelosTablas.TablaFinal;
import pt02.modelosTablas.TablaInicial;
import pt02.modelosTablas.TablaMuestra;
/**
 *
 * @author XiMe!!!
 */
//IMPORTANTE: LOS ARCHIVOS NO DEBEN CONTENER PALABRAS CON ACENTOS YA QUE LOS
CARACTERES NO LOS RECONCE
//PODRÍA INLUIRSE EN EL APARTADO DE MEJORAS AL SISTEMA EN EL REPORTE FINAL
public class Archivos
{

public LinkedList valores(File selfFile)
{
String a=null;
int palabras=0;
LinkedList arreglo=new LinkedList();
File archivo=selfFile;
a=obtener_linea(archivo);
if(a!=null){
palabras=cuenta_palabras(a);
arreglo.addFirst(palabras);
}
else{
JOptionPane.showMessageDialog(null,"No se pudo leer la línea");
return null;
}
try
{
Scanner sc= new Scanner(archivo);
while (sc.hasNext())
{
arreglo.add(sc.next());
}
sc.close();
return arreglo;
}
catch(Exception e)
{

```

```

        e.printStackTrace();
    }
    return null;
}

```

```

public String obtener_linea(File f) //obtiene la primer línea del archivo
{
    String linea=null;
    File archivo=null;
    FileReader fr = null;
    BufferedReader br = null;
    archivo=f;
    try{
        fr = new FileReader (archivo);
        br = new BufferedReader(fr);
        linea=br.readLine();
        linea=linea.replace('\u0009',' ');
        linea=linea.replace(' ', ',');
        fr.close();
        return linea;
    }
    catch(Exception e){
        e.printStackTrace();
    }
    return null;
}

```

```

public int cuenta_palabras(String linea)//cuenta las comas para saber cuantas palabras hay
{
    char c[]=null;
    int indice=0;
    c=linea.toCharArray();

    for(int i=0;i<c.length;i++)
    {
        if(c[i]==',')
        {
            indice=indice+1;
        }
    }
    return indice+1;
}

```

```

public String nombre_archivo()
{
    JFileChooser fc = new JFileChooser();
    String nombre;
    fc.setName("- Abrir Archivo de Población -");
    fc.showOpenDialog(null); // Mostramos el file chooser
    File selFile = fc.getSelectedFile();
    nombre=selFile.getName();
    return nombre;
}

public String manejar_archivo(javax.swing.JTable tabla,javax.swing.JComboBox combo)
{

    JOptionPane.showMessageDialog(null,"Porfavor seleccione su archivo...");
    JFileChooser fc = new JFileChooser();
    fc.setName("- Abrir Archivo de Poblacion -");
    fc.showOpenDialog(null); // Mostramos el file chooser
    File selFile = fc.getSelectedFile();
    String nombre_arch=selFile.getName();
    LinkedList lista_valores=new LinkedList();
    TablaInicial tablaInicial=null; //modelo de tabla
    lista_valores=valores(selFile);//lista de los datos de la población
    if(lista_valores!=null)
        tablaInicial=arma_datos(lista_valores);
    else
        JOptionPane.showMessageDialog(null,"No se pueden convertir los datos a tabla");
    Object[] noms=new Object[tablaInicial.getColumnCount()];
    for(int i=0;i<tablaInicial.getColumnCount();i++)
    {
        noms[i]=tablaInicial.getColumnName(i);
    }
    tabla.setModel(tablaInicial);
    combo.setModel(new javax.swing.DefaultComboBoxModel(noms));
    return nombre_arch;
}

public String manejar_archivo(javax.swing.JTable tabla,javax.swing.JComboBox
combo,javax.swing.JComboBox combo2)
{

    JOptionPane.showMessageDialog(null,"Porfavor seleccione su archivo...");

```

```

JFileChooser fc = new JFileChooser();
fc.setName("- Abrir Archivo de Poblacion -");
fc.showOpenDialog(null); // Mostramos el file chooser
File selFile = fc.getSelectedFile();
String nombre_arch=selFile.getName();
LinkedList lista_valores=new LinkedList();
TablaInicial tablaInicial=null; //modelo de tabla
lista_valores=valores(selFile);//lista de los datos de la población
if(lista_valores!=null)
    tablaInicial=arma_datos(lista_valores);
else
    JOptionPane.showMessageDialog(null,"No se pueden convertir los datos a tabla");
Object[] noms=new Object[tablaInicial.getColumnCount()];
for(int i=0;i<tablaInicial.getColumnCount();i++)
{
    noms[i]=tablaInicial.getColumnName(i);
}
tabla.setModel(tablaInicial);
combo.setModel(new javax.swing.DefaultComboBoxModel(noms));
combo2.setModel(new javax.swing.DefaultComboBoxModel(noms));
return nombre_arch;
}

public TablaInicial arma_datos(LinkedList datos) //contiene las listas de los nombres de las
variables y los numeros
{
    Object p=datos.get(0);
    String pal=p.toString();
    LinkedList var=new LinkedList();
    int palabras=Integer.parseInt(pal);
    LinkedList[] valores=new LinkedList[palabras];
    for(int i=1;i<=palabras;i++)
    {
        var.add(datos.get(i));
    }
    for(int j=0;j<palabras;j++)
    {
        valores[j]=arma_valores(datos,palabras,j);
    }
    TablaInicial tablaInicial=new TablaInicial(var,valores); //este es el modelo de tabla
    return tablaInicial;
}

```

```
}
```

public LinkedList arma\_valores(LinkedList datos,int palabras,int indice) //arma la lista que contiene los datos a analizar

```
{
    LinkedList temp=new LinkedList();
    LinkedList ret=new LinkedList();
    for(int i=palabras+1;i<datos.size();i++)
    {
        temp.add(datos.get(i));
    }
    for(int j=0;j<temp.size();j++)
    {
        if((j*palabras)+indice<temp.size())
            ret.add(temp.get((j*palabras)+indice));
    }
    return ret;
}
```

public void guardar(TablaMuestra muestra,TablaFinal resumen, String[] metodo,int id) //esta funcion escribe la muestra y los resultados en un archivo

```
{
    FileWriter fichero = null;
    PrintWriter pw = null;
    try
    {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setName("-Guardar Como -");
        int seleccion = fileChooser.showSaveDialog(null);
        if (seleccion == JFileChooser.APPROVE_OPTION)
        {
            File arch = fileChooser.getSelectedFile();
            fichero = new FileWriter(arch+".txt");
        }
        pw = new PrintWriter(fichero);
        Date fecha = new Date();
        String fech=fecha.toString();
        switch(id)
        {
            case 0:
                pw.println("Archivo Creado el "+fech);
                pw.println("\n");
        }
    }
}
```

```

        pw.println("\t\t\t\t++++"+metodo[0]+"++++");
        pw.println("\n");
        pw.println(metodo[1]);
        pw.println("\n");
        pw.println("Nombre del Archivo: "+metodo[2]);
        pw.println("Tamaño de la Población: "+metodo[4]);
        pw.println("\n");
        pw.println("****MUESTRA****");
        pw.println("\n");
        pw.println("Tamaño de la Muestra: "+metodo[3]);
        pw.println("\n");
        pw.println("\t"+muestra.getColumnModel(1)+"\n");
        for (int i = 0; i < muestra.getRowCount(); i++)
            pw.println("\t"+muestra.getValueAt(i, 1)+"\n");
        pw.println("####ESTIMACIONES DE LA MUESTRA####");
        pw.println("\n");
        for(int k=0;k<resumen.getRowCount();k++)
            pw.println(resumen.getColumnModel(k)+": "+"\t"+resumen.getValueAt(0, k));
        JOptionPane.showMessageDialog(null, "Archivo Guardado Correctamente");
        break;
    default:
        break;
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        // Nuevamente aprovechamos el finally para
        // asegurarnos que se cierra el archivo.
        if (null != fichero)
            fichero.close();
    } catch (Exception e2) {
        e2.printStackTrace();
    }
}
}
}

public void abrir(String nombre,JTextArea texto) //abre el archivo de ayuda y lo despliega en
textarea
{
    File archivo = new File (nombre);
    try

```

```

{
FileReader fr = new FileReader (archivo);
Scanner sc= new Scanner(archivo);

while (sc.hasNext())
{
    texto.append(sc.nextLine().toString()+"\n");
}
sc.close();
fr.close();

}
catch(Exception e){
    e.printStackTrace();
}
}

public void guardar(String [] argumentos, TableModel t1, TableModel t2, TableModel tf1,
TableModel tf2)
{
    FileWriter fichero = null;
    PrintWriter pw = null;
    try
    {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setName("-Guardar Como -");
        int seleccion = fileChooser.showSaveDialog(null);
        if (seleccion == JFileChooser.APPROVE_OPTION)
        {
            File arch = fileChooser.getSelectedFile();
            fichero = new FileWriter(arch+".txt");
        }
        pw = new PrintWriter(fichero);
        Date fecha = new Date();
        String fech=fecha.toString();

        pw.println("Archivo Creado el "+fech);
        pw.println("\n");
        pw.println("\t\t\t\t\t++++"+argumentos[0]+"++++");
        pw.println("\n");
        pw.println(argumentos[1]);
        pw.println("\n");
    }
}

```



```

        pw.println("Nombre del Archivo: "+argumentos[2]);
        pw.println("Tamaño del Estrato 0: "+argumentos[3]);
        pw.println("Tamaño del Estrato 1: "+argumentos[4]);
        pw.println("Método de Selección de Muestra: "+argumentos[5]);
        pw.println("\n");
        pw.println("****MUESTRA Estrato 0****");
        pw.println("\n");
        pw.println("\t"+t1.getColumnName(1)+"\n");
        for (int i = 0; i < t1.getRowCount(); i++)
            pw.println("\t"+t1.getValueAt(i, 1)+"\n");
        pw.println("####ESTIMACIONES DE LA MUESTRA####");
        pw.println("\n");
        for(int k=0;k<tf1.getRowCount();k++)
            pw.println(tf1.getColumnName(k)+": "+\t"+tf1.getValueAt(0, k));
        pw.println("****MUESTRA Estrato 1****");
        pw.println("\n");
        pw.println("\t"+t2.getColumnName(1)+"\n");
        for (int i = 0; i < t2.getRowCount(); i++)
            pw.println("\t"+t2.getValueAt(i, 1)+"\n");
        pw.println("####ESTIMACIONES DE LA MUESTRA####");
        pw.println("\n");
        for(int k=0;k<tf2.getRowCount();k++)
            pw.println(tf2.getColumnName(k)+": "+\t"+tf2.getValueAt(0, k));
        JOptionPane.showMessageDialog(null, "Archivo Guardado Correctamente");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            // Nuevamente aprovechamos el finally para
            // asegurarnos que se cierra el archivo.
            if (null != fichero)
                fichero.close();
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }
}

public void guardar_estratificado(String argumentos[], TableModel poblacion, TableModel
info, TableModel muestras, TableModel calculos)
{
    FileWriter fichero = null;
    PrintWriter pw = null;

```

```

try
{
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setName("-Guardar Como -");
    int seleccion = fileChooser.showSaveDialog(null);
    if (seleccion == JFileChooser.APPROVE_OPTION)
    {
        File arch = fileChooser.getSelectedFile();
        fichero = new FileWriter(arch+".txt");
    }
    pw = new PrintWriter(fichero);
    Date fecha = new Date();
    String fech=fecha.toString();
    pw.println("Archivo Creado el "+fech);
    pw.println("\n");
    pw.println("\t\t\t\t++++"+argumentos[0]+"++++");
    pw.println("\n");
    pw.println(argumentos[1]);
    pw.println("\n");
    pw.println("Nombre del Archivo: "+argumentos[2]);
    pw.println("Tamaño de los Estratos: "+argumentos[3]);
    pw.println("Método de Selección de Muestra: "+argumentos[4]);
    pw.println("\n");

    pw.println("*****MUESTRAS*****");
    pw.println("\n");
    pw.println("\t"+muestras.getColumnName(1)+"\n");
    for (int i = 0; i < muestras.getRowCount(); i++)
    pw.println("\t"+muestras.getValueAt(i, 1)+"\n");
    pw.println("####ESTIMACIONES DE LA MUESTRA####");
    pw.println("\n");
    for(int k=0;k<calculos.getRowCount();k++)
    pw.println(calculos.getColumnName(k)+" "+"\t"+calculos.getValueAt(0, k));
    pw.println("*****MUESTRA Estrato 1*****");
    pw.println("\n");
    pw.println("\t"+muestras.getColumnName(1)+"\n");
    for (int i = 0; i < muestras.getRowCount(); i++)
    pw.println("\t"+muestras.getValueAt(i, 1)+"\n");
    pw.println("####ESTIMACIONES DE LA MUESTRA####");
    pw.println("\n");
    for(int k=0;k<muestras.getRowCount();k++)
    pw.println(muestras.getColumnName(k)+" "+"\t"+muestras.getValueAt(0, k));

```

```

        JOptionPane.showMessageDialog(null, "Archivo Guardado Correctamente");
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
    finally
    {
        try
        {
            // Nuevamente aprovechamos el finally para
            // asegurarnos que se cierra el archivo.
            if (null != fichero)
                fichero.close();
        }
        catch (Exception e2)
        {
            e2.printStackTrace();
        }
    }
}

public String manejar_archivo(javax.swing.JTable tabla, javax.swing.JComboBox
variable, javax.swing.JComboBox codigo1, javax.swing.JComboBox codigo2)
{

    JOptionPane.showMessageDialog(null, "Porfavor seleccione su archivo....");
    JFileChooser fc = new JFileChooser();
    fc.setName("- Abrir Archivo de Poblacion -");
    fc.showOpenDialog(null); // Mostramos el file chooser
    File selFile = fc.getSelectedFile();
    String nombre_arch=selFile.getName();
    LinkedList lista_valores=new LinkedList();
    TablaInicial tablaInicial=null; //modelo de tabla
    lista_valores=valores(selFile); //lista de los datos de la población
    if(lista_valores!=null)
        tablaInicial=arma_datos(lista_valores);
    else
        JOptionPane.showMessageDialog(null, "No se pueden convertir los datos a tabla");
    Object[] noms=new Object[tablaInicial.getColumnCount()];
    for(int i=0;i<tablaInicial.getColumnCount();i++)
    {
        noms[i]=tablaInicial.getColumn(i);
    }
}

```

```

    }
    tabla.setModel(tablaInicial);
    variable.setModel(new javax.swing.DefaultComboBoxModel(noms));
    codigo1.setModel(new javax.swing.DefaultComboBoxModel(noms));
    codigo2.setModel(new javax.swing.DefaultComboBoxModel(noms));
    return nombre_arch;
}

public void guardar_ponderado(String [] argumentos, TableModel t1, TableModel tf2)
{
    FileWriter fichero = null;
    PrintWriter pw = null;
    try
    {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setName("-Guardar Como -");
        int seleccion = fileChooser.showSaveDialog(null);
        if (seleccion == JFileChooser.APPROVE_OPTION)
        {
            File arch = fileChooser.getSelectedFile();
            fichero = new FileWriter(arch+".txt");
        }
        pw = new PrintWriter(fichero);
        Date fecha = new Date();
        String fech=fecha.toString();

        pw.println("Archivo Creado el "+fech);
        pw.println("\n");
        pw.println("\t\t\t\t++++"+argumentos[0]+"++++");
        pw.println("\n");
        pw.println(argumentos[1]);
        pw.println("\n");
        pw.println("Nombre del Archivo: "+argumentos[2]);
        pw.println("Tamaño de la Población: "+argumentos[3]);
        pw.println("Tamaño de la Muestra: "+argumentos[4]);
        pw.println("Método de Selección: "+argumentos[5]);
        pw.println("\n");
        pw.println("****MUESTRA ****");
        pw.println("\n");
        pw.println("\t"+t1.getColumnName(1)+"\n");
        for (int i = 0; i < t1.getRowCount(); i++)

```

```

        pw.println("\t"+t1.getValueAt(i, 1)+"\n");
        pw.println("###ESTIMACIONES DE LA MUESTRA###");
        pw.println("\n");
        for(int k=0;k<tf2.getRowCount();k++)
            pw.println(tf2.getColumnName(k)+" "+"\\t"+tf2.getValueAt(0, k));
        JOptionPane.showMessageDialog(null, "Archivo Guardado Correctamente");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            // Nuevamente aprovechamos el finally para
            // asegurarnos que se cierra el archivo.
            if (null != fichero)
                fichero.close();
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }
}
}
}

```

## 9.4 Paquete Kernel

### 9.4.1 Muestreo.java

```

/*
 * Esta clase contiene la gestión de los métodos. Tienen un id para saber cual método
 * se va a realizar.
 */
package pt02.kernel;
import java.util.LinkedList;
import javax.swing.JOptionPane;
import javax.swing.table.TableModel;
import pt02.modelosTablas.TablaFinal;
import pt02.modelosTablas.TablaMuestra;
import pt02.modelosTablas.TablaResPob;
import pt02.vistas.Codificacion;
import pt02.vistas.Final;
import pt02.vistas.FinalEstrata;
/**
 * @author XiMe
 */

```

```

public class Muestreo
{
public javax.swing.JComboBox varSeleccionada; //variable a muestrear
public javax.swing.JComboBox alg;      // numero del algoritmo a utilizar o etapa 1 del politapico
public javax.swing.JComboBox etapa2;    //etapa 2 del politápico
public TableModel tblDatos;           //la tabla de datos
public String tamMuestra;              //tamaño de la muestra
public String status;                  // con o sin reemplazo
public String algoritmo;                // tipo de algoritmo
public String [] argu;                  //argumentos para el estratificado
public LinkedList tkns;                 //lista de tokens del estratificado
public LinkedList[] tms;                //lista con los tamaños de las muestras elegidos

public Muestreo(javax.swing.JComboBox v, TableModel t, String tm,String
st,javax.swing.JComboBox a) //constructor aleatorio simple
{
this.varSeleccionada=v;
this.tblDatos=t;
this.tamMuestra=tm;
this.status=st;
this.alg=a;
}

public Muestreo(javax.swing.JComboBox v, TableModel t, String tm,String st,String
num_alg,javax.swing.JComboBox a) //constructor aleatorio simple ponderado frec
{
this.varSeleccionada=v;
this.tblDatos=t;
this.tamMuestra=tm;
this.status=st;
this.algoritmo=num_alg;
this.alg=a;
}

public Muestreo(javax.swing.JComboBox v, TableModel t, String tm,String [] st) //constructor
sistemático
{
this.varSeleccionada=v;
this.tblDatos=t;
this.tamMuestra=tm;
this.argu=st;
}
}

```

```

public Muestreo(javax.swing.JComboBox v, TableModel t, String tm,String
st,javax.swing.JComboBox a,LinkedList tokens,LinkedList[] tms) //constructor estratificado
{
this.varSeleccionada=v; //variable a muestrear
this.tblDatos=t; //tabla de los datos
this.tamMuestra=tm; //numero de estratos
this.status=st; //manual, promedio o archivo
this.alg=a; //algoritmo a utilizar en el muestreo
this.tkns=tokens; //lista de simbolos de los estratos
this.tms=tms; //arreglo de listas de los tamaños de las muestras
}

public Muestreo(javax.swing.JComboBox variable,javax.swing.JComboBox
etapa1,javax.swing.JComboBox etap2,TableModel tbl) //constructor politápico
{
this.varSeleccionada=variable; //variable a muestrear
this.alg=etapa1; //columna de la etapa1
this.etapa2=etap2; //columna de la etapa2
this.tblDatos=tbl; //tabla de los datos
}

public void empezar_metodo(int idMetodo,String nombre,String confianza)
{
switch (idMetodo)
{
case 0://ALEATORIO SIMPLE
Resultados res=new Resultados();
int reemplazo=-1;
//INICIA TABLA RESUMEN
LinkedList a=res.lista_numeros(tblDatos, varSeleccionada.getSelectedIndex()); //datos
de la variable
LinkedList
valores=res.obtener_valores(a,varSeleccionada.getSelectedItem().toString(),status); //cálculos
para tabla resumen
LinkedList nombres=res.nombres_tablas(0);
LinkedList nombres_final=res.nombres_tablas(3);
TablaResPob tabla_res=new TablaResPob(nombres,valores);
//FIN DE TABLA RESUMEN
}
}

```

```

Object an=tblDatos.getRowCount();
String tpob=an.toString();
if(status.startsWith("Con"))
    reemplazo=5; //con reemplazo
else reemplazo=0; //sin reemplazo
String[] args={"Aleatorio Simple No Ponderado",nombre,tamMuestra,tpob};
//INICIA TABLA CON MUESTRA
if(reemplazo==0)
{
    switch(alg.getSelectedIndex())
    {
        case 0: reemplazo=0; //algoritmo xcl
            break;
        case 1: reemplazo=1; //algoritmo 1
            break;
        case 2: reemplazo=2; // algoritmo 2
            break;
        case 3: reemplazo=3; //algoritmo 3
            break;
        case 4: reemplazo=4; //algoritmo 4
            break;
        default:break;
    }
}

```

TablaMuestra

```

tabla_muestra=res.aleatorio_simple(varSeleccionada.getSelectedItem().toString(),tblDatos.getRo
wCount(),a,tamMuestra,reemplazo);
LinkedList vfinal=res.hacer_tablaFinal(tabla_muestra);
TablaFinal fin=new TablaFinal(nombres_final,vfinal);
//FIN TABLA CON MUESTRA
Final.main(args, tabla_res, tabla_muestra, fin,status);
break;

```

case 1: //SISTEMATICO

//String[]

```

hola={"Sistemático",nombreArchivo,tam,tampob,txtInicio.getText(),txtPaso.getText(),seleccionado
};

```

```

Resultados sistematico=new Resultados();
LinkedList listaNumeros=new LinkedList();
LinkedList val=new LinkedList();
LinkedList nombresfinal=sistematico.nombres_tablas(3);

```



```

int ts=0,ini=0,paso=0,tpo=0;
ts=Integer.parseInt(argu[2].trim());
ini=Integer.parseInt(argu[4].trim());
paso=Integer.parseInt(argu[5].trim());
tpo=Integer.parseInt(argu[3].trim());
int sel=varSeleccionada.getSelectedIndex();
listaNumeros=sistemico.lista_numeros(tblDatos, sel);
val=sistemico.obtener_valores(listaNumeros, argu[6], "numérico");
LinkedList nomb=sistemico.nombres_tablas(0);
TablaResPob tablares=new TablaResPob(nomb,val);
TablaMuestra muestra=sistemico.sistemico(argu[6],tpo, ts,listaNumeros,ini,paso);
LinkedList vfin=sistemico.hacer_tablaFinal(muestra);
TablaFinal f=new TablaFinal(nombresfinal,vfin);
Final.main(argu,tablares, muestra, f, "Inicio: "+ini+"\n"+" Paso: "+paso);
break;

```

case 2://ESTRATIFICADO

```

Resultados estratos=new Resultados();
//lista de los datos a muestrear, depende de la variable seleccionada
LinkedList data=estratos.lista_numeros(tblDatos, varSeleccionada.getSelectedIndex());
//comparamos para ver que selecciono el usuario, fuente de los datos
if(status.equals("Archivo"))
{
String []
argumentos={status,nombre,varSeleccionada.getSelectedItem().toString(),tamMuestra,confianza};
//seleccionó codigos de un archivo, nombre del archivo
LinkedList datos2=estratos.lista_numeros(tblDatos,alg.getSelectedIndex());
//lista de los códigos asignados a los datos sirve para presentación solamente
TableModel
archivo=estratos.codificacion_archivo(data,datos2,tkns,varSeleccionada.getSelectedItem().toString());
//lista que contiene los códigos, se usa para los cálculos
LinkedList indices=estratos.tama;
TableModel archivo2=estratos.pulir_tabla(archivo, tkns,indices);
//CON LA FUNCION LISTA ESTRATOS SE PUEDE PULIR PARA PRESENTACION LA TABLA
DE DATOS
Codificacion.main(argumentos,archivo2,archivo,data,tmS);
}
else
if(status.equals("Manual"))
{

```

```

        //hacemos la tabla con los datos
        TableModel
manual=estratos.codificacion_manual(data,varSeleccionada.getSelectedItemAt().toString());
        String[] argumentos={status,nombre,varSeleccionada.getSelectedItemAt().toString()};
        //le mandamos el modelo de tabla al módulo de codificación
        Codificacion.main(argumentos>manual>manual,data);
        break;
    }
    else
        if(status.equals("Promedio"))
        {
            //hacemos la tabla con los códigos por promedio
            TableModel
promedios=estratos.codificacion_promedio(data,varSeleccionada.getSelectedItemAt().toString(),0);
            TableModel
promediosInvertido=estratos.codificacion_promedio(data,varSeleccionada.getSelectedItemAt().toString(),1);
            String[] argumentos={status,nombre,varSeleccionada.getSelectedItemAt().toString()};
            //le mandamos el modelo de tabla al módulo de codificación
            Codificacion.main(argumentos,promedios,promediosInvertido,data);
            break;
        }
        break;
    case 3://ALEATORIO SIMPLE NO PONDERADO
        Resultados ponderado=new Resultados();
        //HACEMOS LA TABLA RESUMEN DE LA POBLACIÓN
        LinkedList datosVar=ponderado.lista_numeros(tblDatos,
varSeleccionada.getSelectedIndex()); //datos de la variable
        LinkedList datosPF=ponderado.lista_numeros(tblDatos,
alg.getSelectedIndex());//ponderaciones o frecuencias
        LinkedList
calculosResumen=ponderado.obtener_valores(datosVar,varSeleccionada.getSelectedItemAt().toString(),status); //cálculos para tabla resumen
        LinkedList nombresTablas=ponderado.nombres_tablas(0); //nombres de la tabla
resumen
        LinkedList nombresFinal=ponderado.nombres_tablas(3); //nombres de la tabla final
        TablaResPob tablaResumen=new TablaResPob(nombresTablas,calculosResumen);
        //FIN DE TABLA RESUMEN DE LA POBLACIÓN
        String
cadena[]={nombre,confianza,varSeleccionada.getSelectedItemAt().toString(),tamMuestra};
        //EMPEZAR MUESTREO
        if(status.contains("Lista"))

```

```

    {
        TablaMuestra
tablaPonderado=ponderado.ponderado_probados(status,datosVar,datosPF);
        LinkedList ponderaciones=ponderado.lista_numeros(tablaPonderado, 2);
        TablaMuestra tp=ponderado.ponderadoSinReemplazo(datosVar, datosPF,
ponderaciones, tamMuestra,cadena[2]);
        /////
        LinkedList tablaFinal=ponderado.hacer_tablaFinal(tablaPonderado);
        TablaFinal finnal=new TablaFinal(nombresFinal,tablaFinal);
        FinalEstrata.main(cadena,tp,finnal,tablaResumen,tablaPonderado);
    }
    else
    {
        TablaMuestra
tablaPonderado=ponderado.ponderado_frecuencias(algoritmo,datosVar,datosPF,tamMuestra,cad
ena[2]);
        LinkedList tablaFinal=ponderado.hacer_tablaFinal(tablaPonderado);
        TablaFinal finnal=new TablaFinal(nombresFinal,tablaFinal);
        TableModel tblInfo;
        tblInfo=new javax.swing.table.DefaultTableModel(
        new Object [][] {
        {"Nombre Archivo", cadena[0]},
        {"Tamaño de la Muestra", tamMuestra,},
        {"Método de Selección", cadena[1]},
        {"Variable Seleccionada", cadena[2]},,
        {"Algoritmo Seleccionado","Algoritmo "+algoritmo}
        },
        new String [] {
        "" , ""
        }
        );
        FinalEstrata.main(cadena,tablaPonderado,finnal,tablaResumen,tblInfo);
    }
    //el string status indica si fue tabla de frecuencias o lista de probas
    break;
    case 4: //POLITAPICO
        Resultados politapico=new Resultados();
        LinkedList
datos=politapico.lista_numeros(tblDatos,varSeleccionada.getSelectedIndex());
        LinkedList etapa1=politapico.lista_numeros(tblDatos, alg.getSelectedIndex());
        LinkedList etapa2=politapico.lista_numeros(tblDatos,etapa2.getSelectedIndex());
        tkns=politapico.contar_estratos(etapa1);

```

```

        etapa1=politapico.lista_numeros(tblDatos, alg.getSelectedIndex()); //porq se pierde la
referencia
        TableModel
archivo=politapico.codificacion_archivo(datos,etapa1,tkns,varSeleccionada.getSelectedItem().toSt
ring());
        break;
        default: JOptionPane.showMessageDialog(null,"Method Unavailable");
    }
}
}

```

#### 9.4.2 Resultados.java

```

/*
 *Esta clase hace los cálculos y genera los modelos de tablas para desplegar en el JFrame Final
 */
package pt02.kernel;
import pt02.estadistica.Operaciones;
import java.util.LinkedList;
import javax.swing.table.TableModel;
import pt02.modelosTablas.TablaEstratos;
import pt02.modelosTablas.TablaLimites;
import pt02.modelosTablas.TablaMuestra;
/**
 * @author XiMe
 */
public class Resultados
{
    public LinkedList tama;
    public LinkedList obtener_valores(LinkedList numeros,String sel,String tipo) //hace los cálculos
para mostrar en la tabla resumen
    {
        LinkedList calculos=new LinkedList();
        Operaciones operaciones=new Operaciones();
        double promedio=0;
        double max=0;
        double min=0;
        double varianza=0;
        double desv=0;

        promedio=operaciones.promedio(numeros);
        max=operaciones.max(numeros);

```

```

min=operaciones.min(numeros);
varianza=operaciones.varianza(numeros,promedio);
desv=operaciones.desv(varianza);
calculos.add(sel);
calculos.add(tipo);
calculos.add(min);
calculos.add(max);
calculos.add(promedio);
calculos.add(varianza);
calculos.add(desv);

return calculos;
}

```

```

public LinkedList lista_numeros(TableModel tbl,int sel) //datos de la variable
{
    LinkedList a=new LinkedList();
    int n=tbl.getRowCount();
    Object[] num=new Object[n];
    double[] aux=new double[n];
    String en=null;
    for(int i=0;i<n;i++)
    {
        num[i]=tbl.getValueAt(i, sel);
        en=num[i].toString().trim();
        aux[i]=Double.parseDouble(en);
        a.add(aux[i]);
    }
    return a;
}

```

```

public LinkedList nombres_muestra(String variable) //lista de nombres para la tabla muestra
{
    LinkedList muestra=new LinkedList();
    muestra.add("ID");
    muestra.add(variable);
    return muestra;
}

```

```

public LinkedList[] aleatorio_simple_sin_reemplazo(TableModel tm,String
tamMuestra,LinkedList aleatorios) //para el aleatorio simple
{

```

```

int tam=0;
tam=Integer.parseInt(tamMuestra); //sacar tamaño de la muestra
int[] rndm=new int[tam];
Operaciones operaciones=new Operaciones();
LinkedList[] modelo=new LinkedList[2];
LinkedList datos=new LinkedList();

rndm=operaciones.enteros(aleatorios); //arreglo de numeros aleatorios enteros (poblacion)
datos=operaciones.sublista(rndm, tam); //sublista de numeros aleatorios para la muestra
LinkedList numeros=new LinkedList();

for(int i=0;i<tam;i++)
{
    int id=Integer.parseInt(datos.get(i).toString().trim());
    Object objeto=tm.getValueAt(id,1);
    numeros.add(objeto);
}

    modelo[0]=datos;
    modelo[1]=numeros;

return modelo;
}

public LinkedList valores_sistematico(int tam)
{
    LinkedList res=new LinkedList();
    for(int i=0;i<tam;i++)
        res.add(i);
    return res;
}

public TablaMuestra aleatorio_simple(String nombre,int tamP,LinkedList datos,String tamM,int
status)
{
    Operaciones operaciones=new Operaciones();
    LinkedList names=nombres_muestra(nombre); //nombres para el modelo de tabla de la
muestra
    LinkedList mst[]=new LinkedList[2]; //valores para el modelo de tabla de la muestra
    LinkedList buena[]=new LinkedList[2];
    LinkedList sr=new LinkedList();
    LinkedList auz=new LinkedList();

```

```

int muestra=Integer.parseInt(tamM);
switch(status)
{
case 0: //SIN REEMPLAZO
    //GENERAR P NÚMEROS ALEATORIOS ENTEROS
    mst[0]=operaciones.aleatorio(tamP);//genero los n número aleatorios (índices)
    mst[1]=datos; //datos a muestrear
    //ASIGNAR A CADA DATO DE LA POBLACIÓN UN NÚMERO
    TablaMuestra tablamuestra=new TablaMuestra(names,mst);
    //TOMAR LOS PRIMEROS N DATOS DE LA POBLACIÓN
    buena=aleatorio_simple_sin_reemplazo(tablamuestra,tamM,mst[0]);
    TablaMuestra nice=new TablaMuestra(names,buena);
    return nice;

case 1: //SIN REEMPLAZO utilizando el algoritmo 1
    buena=algoritmo1(datos, Integer.parseInt(tamM.trim()), tamP-1);
    TablaMuestra resalg1=new TablaMuestra(names,buena);
    return resalg1;

case 2: //SIN REEMPLAZO utilizando el algoritmo 2
    buena=algoritmo2(datos,Integer.parseInt(tamM.trim()),tamP-1);
    TablaMuestra resalg2=new TablaMuestra(names,buena);
    return resalg2;

case 3: //SIN REEMPLAZO utilizando el algoritmo 3
    buena=algoritmo3(datos,Integer.parseInt(tamM.trim()),tamP-1);
    TablaMuestra resalg3=new TablaMuestra(names,buena);
    return resalg3;

case 4: //SIN REEMPLAZO utilizando el algoritmo 4
    buena=algoritmo4(datos,Integer.parseInt(tamM.trim()),tamP-1);
    TablaMuestra resalg4=new TablaMuestra(names,buena);
    return resalg4;

case 5: //CON REEMPLAZO
    //TOMAR EL PRIMER DATO DE LA LISTA N VECES HASTA COMPLETAR LA MUESTRA
    for(int i=0;i<muestra;i++)
    {
        sr=operaciones.aleatorio(tamP);//genero los n número aleatorios (índices)
        auz.add(sr.get(0));
    }
    mst[1]=datos; //datos a muestrear

```

```

        mst[0]=auz;
        buena=aleatorio_simple_con_reemplazo(mst[1],mst[0],muestra);
        TablaMuestra ok=new TablaMuestra(names,buena);
        return ok;

    default:
        return null;
    }
}

public LinkedList[] aleatorio_simple_con_reemplazo(LinkedList datos,LinkedList numeros,int
tam)
{
    //llega la lista están los datos
    //llega la lista de los aleatorios que indica q valores tomar del modelo de tabla
    //llega el tamaño de la muestra
    LinkedList [] simple=new LinkedList[tam];
    LinkedList valores_muestra=new LinkedList();
    LinkedList id=new LinkedList();
    int aux=0;
    for(int i=0;i<tam;i++)
    {
        aux=Integer.parseInt(numeros.get(i).toString().replace(".0", "").trim());
        valores_muestra.add(datos.get(aux));
    }
    for(int j=0;j<tam;j++)
    {
        id.add(Integer.parseInt(numeros.get(j).toString().replace(".0", "").trim()));
    }
    simple[0]=id; //id
    simple[1]=valores_muestra;//datos
    return simple;
}

public TablaMuestra sistematico(String nombre,int tamP,int tamM,LinkedList valores,int k,int n)
{
    //k es el inicio
    //n es el paso
    LinkedList names=nombres_muestra(nombre); //nombres para el modelo de tabla
    LinkedList[] vals=new LinkedList[2];
    LinkedList w=new LinkedList();
    LinkedList y=new LinkedList();
}

```



```

w.add(valores.get(k));
y.add(k);
int kactual=k;
for(int i=1;i<tamM;i++)
{
    kactual=(kactual+n)%tamP;
    w.add(valores.get(kactual));
    y.add(kactual);
}
vals[1]=w;
vals[0]=y;
TablaMuestra res=new TablaMuestra(names,vals);
return res;
}

public LinkedList hacer_tablaFinal(TablaMuestra tabla)
{
    Operaciones operaciones=new Operaciones();
    LinkedList numeros=new LinkedList();
    LinkedList valores=new LinkedList();

    double promedio=0;
    double varianza=0;
    double zeta=1.97; //al 95%
    double error=0;

    for(int a=0;a<tabla.getRowCount();a++)
        valores.add(tabla.getValueAt(a, 1));

    promedio=operaciones.promedio(valores);
    varianza=operaciones.varianza(valores,promedio);

    double liminf=(promedio-zeta)*(varianza/Math.sqrt(valores.size()));
    double limsup=(promedio+zeta)*(varianza/Math.sqrt(valores.size()));

    numeros.add(liminf);
    numeros.add(limsup);
    numeros.add(error);
    numeros.add(promedio);
    numeros.add(varianza);

    return numeros;
}

```

```
}
```

```
public LinkedList nombres_tablas(int id)
{
    LinkedList aux=new LinkedList();
    switch (id)
    {
        case 0: //TABLA RESUMEN
            aux.add("Variable Elegida");
            aux.add("Tipo");
            aux.add("Mínimo");
            aux.add("Máximo");
            aux.add("Promedio");
            aux.add("Varianza");
            aux.add("Desviación Std");
            return aux;

        case 1: //TABLA CODIFICACION MANUAL
            aux.add("Códigos");
            aux.add("Estratos");
            aux.add("n");
            return aux;

        case 2: //TABLA CODIFICACION PROMEDIO
            aux.add("Códigos");
            aux.add("Estratos");
            aux.add("n");
            aux.add("Límite Inferior");
            aux.add("Límite Superior");
            return aux;

        case 3: //TABLA FINAL
            aux.add("Límite Inferior");
            aux.add("Límite Superior");
            aux.add("Error Estimación");
            aux.add("Promedio");
            aux.add("Varianza");
            return aux;

        case 4: //TABLA ESTRATIFICADO
            aux.add("Estrato");
            aux.add("N");
    }
}
```

```

        aux.add("n");
        aux.add("Límite Inferior");
        aux.add("Límite Superior");
        aux.add("Error Estimación");
        aux.add("Promedio");
        aux.add("Varianza");
        return aux;

    default: return null;
}
}

public TableModel codificacion_promedio(LinkedList datos,String variable,int id)
{
    Operaciones op=new Operaciones();
    LinkedList[] numeros=new LinkedList[4];
    LinkedList nombres=new LinkedList();
    LinkedList aux0=new LinkedList();
    LinkedList aux1=new LinkedList();
    LinkedList datos0=new LinkedList();
    LinkedList datos1=new LinkedList();
    double comparador;
    nombres.add("Código 0");
    nombres.add(variable);
    nombres.add("Código 1");
    nombres.add(variable);
    double promedio=op.promedio(datos);

    switch(id)
    {
        case 0:
            for(int i=0;i<datos.size();i++)
            {
                comparador=Double.parseDouble(datos.get(i).toString());
                if(comparador>=promedio)
                {
                    aux1.add("1");
                    datos1.add(datos.get(i).toString());
                }
                else
                {
                    aux0.add("0");
                }
            }
        }
    }
}

```

```

        datos0.add(datos.get(i).toString());
    }

}
aux0=rellenar(aux0,datos.size());
datos0=rellenar(datos0,datos.size());
aux1=rellenar(aux1,datos.size());
datos1=rellenar(datos1,datos.size());
numeros[0]=aux0;
numeros[1]=datos0;
numeros[2]=aux1;
numeros[3]=datos1;
TablaLimites tabla=new TablaLimites(nombres,numeros);
return tabla;
case 1:
for(int i=0;i<datos.size();i++)
{
    comparador=Double.parseDouble(datos.get(i).toString());
    if(comparador>=promedio)
    {
        aux0.add("0");
        datos0.add(datos.get(i).toString());
    }
    else
    {
        aux1.add("1");
        datos1.add(datos.get(i).toString());
    }

}
aux0=rellenar(aux0,datos.size());
datos0=rellenar(datos0,datos.size());
aux1=rellenar(aux1,datos.size());
datos1=rellenar(datos1,datos.size());
numeros[0]=aux0;
numeros[1]=datos0;
numeros[2]=aux1;
numeros[3]=datos1;
TablaLimites tbl=new TablaLimites(nombres,numeros);
return tbl;
}

```

```

return null;
}

public TableModel codificacion_manual(LinkedList datos,String variable)
{
    LinkedList nombres=new LinkedList();
    LinkedList nulos=new LinkedList();
    nombres.add(variable);
    nombres.add("Código 1");
    LinkedList[] names=new LinkedList[2];
    names[0]=datos;
    for(int i=0;i<names[0].size();i++)
        nulos.add(null);
    names[1]=nulos;
    TablaLimites codigos=new TablaLimites(nombres,names);

    return codigos;
}

```

```

public LinkedList[] separar_codigos(TableModel tabla,LinkedList ids) //separa en valores con 1's
y 0's
{
    LinkedList unos=new LinkedList();
    LinkedList ceros=new LinkedList();
    int uno=Integer.parseInt(ids.get(1).toString().trim());
    int dos=Integer.parseInt(ids.get(0).toString().trim());
    LinkedList[] codigosSeparados=new LinkedList[2];
    for(int i=0;i<uno;i++)
        ceros.add(tabla.getValueAt(i, 1));
    for(int j=0;j<dos;j++)
        unos.add(tabla.getValueAt(j, 3));
    codigosSeparados[0]=unos;
    codigosSeparados[1]=ceros;
    return codigosSeparados;
}

```

```

public LinkedList[] algoritmo1(LinkedList datos,int tamMuestra,int tamPob) //sin reemplazo
{
    Operaciones operaciones=new Operaciones();
    double aleatorio=0;
    int referencia=tamPob;
    LinkedList[] resAlgoritmo1=new LinkedList[tamMuestra];
}

```

```

LinkedList a=new LinkedList(); //id's de datos
LinkedList b=new LinkedList(); //datos elegidos
//empezar ciclo
while(tamMuestra!=0)
{
    //generar numero aleatorio flotante
    aleatorio=operaciones.aleatorio();
    double div=(double)tamMuestra/(double)tamPob;
    tamPob=tamPob-1;
    else
    {
        //armar los valores
        if(tamPob+1>referencia)
        {
            if(a.isEmpty())
            {
                a.add(tamPob);
                b.add(datos.get(tamPob));
            }
            else
            {
                if(operaciones.repetido(a, tamPob)) //PARA QUE NO META REPETIDOS
                {
                    tamPob=tamPob+1;
                    tamMuestra=tamMuestra+1;
                }
                else
                {
                    a.add(tamPob);
                    b.add(datos.get(tamPob));
                }
            }
            tamPob=tamPob-1;
            tamMuestra=tamMuestra-1;
        }
    }
    else
    {
        if(a.isEmpty())
        {
            a.add(tamPob);
            b.add(datos.get(tamPob));
        }
    }
}

```

```

else
{
    if(operaciones.repetido(a, tamPob+1))
    {
        tamPob=tamPob+1;
        tamMuestra=tamMuestra+1;
    }
    else
    {
        a.add(tamPob+1);
        b.add(datos.get(tamPob+1));
    }
}
tamPob=tamPob-1;
tamMuestra=tamMuestra-1;
}
}
} //FIN DEL CICLO PRINCIPAL
resAlgoritmo1[0]=a;
resAlgoritmo1[1]=b;
return resAlgoritmo1;
}

```

```

public LinkedList[] algoritmo2(LinkedList datos,int tamMuestra,int tamPob) //sin reemplazo
{
    Operaciones operaciones= new Operaciones();
    LinkedList[] ok=new LinkedList[2];
    int k=tamMuestra+1; //por el momento
    LinkedList a=new LinkedList();
    LinkedList b=new LinkedList();
    //empieza ciclo
    while(tamMuestra!=0)
    {
        //generar aleatorio
        double aleatorio=operaciones.aleatorio();
        //comparar
        double div=(double)tamMuestra/(double)k;
        if(aleatorio>div)
            k=k+1;
        else
        {
            else

```

```

    {
        if(operaciones.repetido(a, k%datos.size()))
        {
            tamMuestra=tamMuestra+1;
        }
        else
        {
            a.add(k%datos.size());
            b.add(datos.get(k%datos.size()));
        }
    }
    tamMuestra=tamMuestra-1;
    k=k+1;
}
}
ok[0]=a;
ok[1]=b;
return ok;
}

```

```

public LinkedList[] algoritmo3(LinkedList datos, int tamMuestra,int tamPob) //sin reemplazo
{
    Operaciones operaciones=new Operaciones();
    int tp=tamPob;
    int indice=0;
    LinkedList a=new LinkedList();//indices
    LinkedList b=new LinkedList();//valores
    LinkedList[] fin=new LinkedList[2];
    //INICIA CICLO WHILE PRINCIPAL
    while(tamMuestra!=0)
    {
        boolean indicador=false;
        int S=1;
        //generar numero aleatorio
        double aleatorio=operaciones.aleatorio();
        int denominador=operaciones.combinacion(tamPob,tamMuestra);
        //INICIA CICLO PARA ENCONTRAR S
        while(indicador==false)
        {
            int numerador=operaciones.combinacion(tamPob-S-1, tamMuestra);
            double div=(double)numerador/(double)denominador;
            double molote=1-div;

```



```

//comparar con molote
if(aleatorio<=molote)

    {
        indice=S+Integer.parseInt(a.getLast().toString()+1;
        if(!operaciones.repetido(a,indice))
            {
                a.add(indice%tp);
                b.add(datos.get(indice%tp));
            }
        }
    if(S>=tamPob)
        indicador=true;
    else
        if(tamPob-S-1<=tamMuestra) //*****CHECAR
            tamPob=datos.size()-1;

        tamPob=tamPob-S-1;
        tamMuestra=tamMuestra-1;
        indicador=true;
    }
    else
    {
        if(S>=tamPob)
            indicador=true;
        else
            S=S+1; //FALTA PONER CUANDO NO ENCUENTRA S
    }

} //FIN DEL CICLO WHILE PARA ENCONTRAR S
} //FIN DEL CICLO WHILE PRINCIPAL
fin[0]=a;
fin[1]=b;
return fin;
}

public LinkedList[] algoritmo4(LinkedList datos, int tamMuestra,int tamPob) //sin reemplazo
{
    Operaciones operaciones=new Operaciones();
    // int tp=tamPob;
    //int indice=0;
    int S=0;

```

```

LinkedList a=new LinkedList();//indices
LinkedList b=new LinkedList();//valores
LinkedList aleatorios=new LinkedList();
LinkedList[] fin=new LinkedList[2];
double ran=0;
//CICLO PRINCIPAL
while(tamMuestra!=0)
{
    aleatorios.clear(); //limpiar lista de aleatorios
    //generar n numeros aleatorios
    for(int i=0;i<tamMuestra;i++)
    {
        ran=operaciones.aleatorio();
        aleatorios.add((tamPob-i+1)*ran);
    }
    double minimo=operaciones.min(aleatorios);
    S=(int)minimo;
    if(a.isEmpty())
    {
        a.add(S);
        b.add(datos.get(S));
    }
    else
    {
        if(operaciones.repetido(a,S) //si esta repetido, debe tomar otro
            tamMuestra=tamMuestra+1;
        }
        tamMuestra=tamMuestra-1;
    }
}
//FIN DEL CICLO PRINCIPAL

fin[0]=a;
fin[1]=b;
return fin;
}

public TablaMuestra ponderado_probos(String tipo,LinkedList valores,LinkedList probas)
{
    //Lista de Probabilidades
    //ordenar probabilidades de menor a mayor
    Operaciones op=new Operaciones();

```

```

LinkedList nombres=new LinkedList();
LinkedList[] res=new LinkedList[3];
nombres.add("Valores");
nombres.add("Probabilidades");
nombres.add("Acumulado");
LinkedList ordenados=new LinkedList();
LinkedList acumulados=new LinkedList();
double[] numeros=op.flotantes(probas);
ordenados=op.quicksort_flotante(numeros, 0, numeros.length-1);
    //hacer las probabilidades acumuladas
acumulados=proba_acumulada(ordenados);
res[0]=valores;
res[1]=probas;
res[2]=acumulados;
TablaMuestra tabla=new TablaMuestra(nombres,res);
return tabla;
}

```

```

public TablaMuestra ponderado_frecuencias(String tipo_alg,LinkedList valores,LinkedList
probas,String tm,String v)
{
    //lista de frecuencias
    int st=Integer.parseInt(tipo_alg);
    LinkedList poblacion=new LinkedList();
    //armar la lista de toda la poblacion desglosada
    int indice=0;
    String c=null;
    for(int j=0;j<probas.size();j++)
    {
        c=probas.get(j).toString();
        if(c.contains(".0"))
            c=c.replace(".0", "");
        indice=Integer.parseInt(c.trim());
        for(int i=0;i<indice;i++)
        {
            poblacion.add(valores.get(j));
        }
    }

    TablaMuestra tabla=aleatorio_simple(v,poblacion.size(),poblacion,tm,st);
    return tabla;
}

```

```

}
public LinkedList proba_acumulada(LinkedList nums)
{
    LinkedList resultados=new LinkedList();
    double siguiente=0;
    double numero=0;
    for(int i=0;i<nums.size()-1;i++)
    {
        numero=Double.parseDouble(nums.get(i).toString().trim());
        siguiente=Double.parseDouble(nums.get(i+1).toString().trim());
        resultados.add(numero+siguiente);
    }
    resultados.add("1");
    return resultados;
}

public LinkedList nombres_estratos(int numero)
{
    LinkedList nombres=new LinkedList();
    for(int i=1;i<=numero;i++)
        nombres.add("Estrato "+i);
    return nombres;
}

public TableModel codificacion_archivo(LinkedList datos, LinkedList codigos,LinkedList
tokens,String variable)
{
    int numeroEstratas=tokens.size(); //cuantos estratos tenemos
    LinkedList nombres=new LinkedList();
    LinkedList n=new LinkedList();
    LinkedList[] coso=new LinkedList[2];
    //LLENADO DE LA LISTA DE NOMBRES PARA LA TABLA
    for(int i=0;i<numeroEstratas;i++)
    {
        nombres.add(variable);
        nombres.add(tokens.get(i).toString());
    }
    n.add("Datos");
    n.add("Códigos");
    coso[0]=datos;
    coso[1]=codigos;
    TablaLimites wawa=new TablaLimites(n,coso);
}

```

```

TablaEstratos contenido=separar_estratos(wawa,tokens,nombres);
//TablaLimites res=new TablaLimites(nombres,contenido);
return contenido;
}

```

```

public LinkedList rellenar(LinkedList datos, int numero)
{
    int falta=numero-datos.size();
    for(int i=0;i<falta;i++)
        datos.add(null);
    return datos;
}

```

```

public LinkedList codificacion_promedio(LinkedList datos,int id)
{
    Operaciones op=new Operaciones();
    LinkedList aux0=new LinkedList();
    LinkedList aux1=new LinkedList();
    LinkedList datos0=new LinkedList();
    LinkedList datos1=new LinkedList();
    double comparador;
    double promedio=op.promedio(datos);
    LinkedList ok=new LinkedList();

    switch(id)
    {
        case 0:
            for(int i=0;i<datos.size();i++)
            {
                comparador=Double.parseDouble(datos.get(i).toString());
                if(comparador>=promedio)
                {
                    aux1.add("1");
                    datos1.add(datos.get(i).toString());
                }
                else
                {
                    aux0.add("0");
                    datos0.add(datos.get(i).toString());
                }
            }
    }
}

```

```

        ok.add(datos0.size());
        ok.add(datos1.size());
        return ok;
    case 1:
        for(int i=0;i<datos.size();i++)
        {
            comparador=Double.parseDouble(datos.get(i).toString());
            if(comparador>=promedio)
            {
                aux0.add("0");
                datos0.add(datos.get(i).toString());
            }
            else
            {
                aux1.add("1");
                datos1.add(datos.get(i).toString());
            }

        }
        ok.add(datos0.size());
        ok.add(datos1.size());
        return ok;
    }
    return null;
}

```

```

public LinkedList contar_estratos(LinkedList datos)
{
    LinkedList tokens=new LinkedList();
    LinkedList frecuencia=new LinkedList();
    LinkedList vals=new LinkedList();
    while(!datos.isEmpty())
    {
        int contador=0;
        String numero=null;
        String comparador=null;
        int tam=datos.size();
        numero=datos.getFirst().toString();
        for(int i=0;i<tam;i++)
        {
            comparador=datos.get(i).toString();

```

```

    if(comparador.equals(numero))
    {
        contador=contador+1;
        vals.add(i); //indices para quitar en datos
    }
}
tokens.add(numero);
frecuencia.add(contador);
for(int jj=0;jj<vals.size();jj++)
{
    int n=Integer.parseInt(vals.get(jj).toString());
    datos.remove(n-jj);
}
vals.clear();
} //FIN DEL CICLO PRINCIPAL
return tokens;
}

```

```

public TablaEstratos separar_estratos(TableModel tabla, LinkedList tokens,LinkedList n)
{
    String[] nombres=new String[tokens.size()];
    for(int ii=0;ii<tokens.size();ii++)
        nombres[ii]=tokens.get(ii).toString();
    int tam=tabla.getRowCount();
    Double [][] strata=new Double[tam][tam];
    Double [][] aux=new Double[tam][tam];
    Double [][]ok=new Double[tam][tam];
    LinkedList indices=new LinkedList();
    LinkedList indices2=new LinkedList();
    LinkedList tamaños=new LinkedList();
    //ARMAR EL ARREGLO DE DATOS
    for(int j=0;j<tokens.size();j++)
    {
        String tok1=tokens.get(j).toString();
        if(!tok1.contains(".0"))
            tok1=tok1.concat(".0");
        for(int i=0;i<tam;i++)
        {
            String comp=tabla.getValueAt(i, 1).toString();
            if(!comp.contains(".0"))
                comp=comp.concat(".0");
            if(comp.equals(tok1))

```

```

        {
            String s=tabla.getValueAt(i, 0).toString();
            strata[j][i]=Double.parseDouble(s);
            indices.add(i);
            indices2.add(i);
        }
    }
    tamaños.add(indices2.size());
    indices2.clear();
}
for(int t=0;t<tokens.size();t++)
    for (int tt=0;tt<tam;tt++)
    {
        aux[t][tt]=strata[t][Integer.parseInt(indices.get(tt).toString())];
    }
this.tama=tamaños;
TablaEstratos ojala=new TablaEstratos(aux,nombres);
return ojala;
}
public LinkedList cuenta_estratos(TableModel datos, int indice)
{

    Resultados res=new Resultados();
    LinkedList estratos=new LinkedList();
    for(int i=0;i<datos.getRowCount();i++)
        estratos.add(datos.getValueAt(i, indice));
    LinkedList numEstratos=res.contar_estratos(estratos);
    return numEstratos;
}
public LinkedList tamPob_estrato(TableModel cod) //Lista ligada que contiene el tamaño de
cada estrato
{
    LinkedList hh=new LinkedList();
    int contador=0;
    //PRIMERO SACAMOS EL TAMAÑO DE LA POBLACION DE CADA ESTRATO
    for(int j=0;j<cod.getColumnCount();j++)
    {
        for(int i=0;i<cod.getRowCount();i++)
        {
            Object oka=cod.getValueAt(i, j);
            if(oka!=null)
                contador=contador+1;
        }
    }
}

```



```

    }
    hh.add(contador);
    contador=0;
}
return hh;
}
public LinkedList lista_estratos(TableModel tm,LinkedList l,int indice,int jmp)
{

    LinkedList estrato=new LinkedList();
    for(int i=0;i<Integer.parseInt(l.get(indice).toString());i++)
        estrato.add(tm.getValueAt(i+jmp,indice));

    return estrato;
}
public TablaEstratos mergeTables(TableModel[] tm,int tamPob,int tamEstrato,LinkedList tam)
//juntar las tablas de muestras
{
    //creamos la lista de nombres de la tabla
    LinkedList nombres=new LinkedList();
    for(int w=1;w<=tamEstrato;w++)
    {
        //nombres.add("ID");
        nombres.add("Estrato "+w);
    }
    String[]n=new String[nombres.size()];
    for(int e=0;e<nombres.size();e++)
        n[e]=nombres.get(e).toString();
    //tamaño total de la tabla de muestras
    int contador=0;
    for(int q=0;q<tam.size();q++)
        contador=contador+tm[q].getRowCount();
    //arreglos para juntar las muestras en una sola tabla
    Double[][]arreglo=new Double[nombres.size()][contador];
    TablaEstratos te=new TablaEstratos(arreglo,n);
    for(int j=0;j<nombres.size();j++) //controla las columnas
        for(int i=0;i<tm[j].getRowCount();i++)
            te.setValueAt(tm[j].getValueAt(i,1), i, j);
    return te;
}
public TablaEstratos mergeTables(TableModel[]tm,int tamEstrato) //juntar las tablas de calculos

```

```

{
    LinkedList nombres=nombres_tablas(3);
    nombres.addFirst("Estrato");
    String[]n=new String[nombres.size()];
    for(int a=0;a<nombres.size();a++)
        n[a]=nombres.get(a).toString();
    Double[][]arreglo=new Double[tamEstrato][nombres.size()];
    Double[][]arregloT=new Double[nombres.size()][tamEstrato];
    for(int i=0;i<tamEstrato;i++)
    {
        arreglo[i][0]=(double)i+1;
        for(int j=1;j<6;j++)
        {
            arreglo[i][j]=Double.parseDouble(tm[i].getValueAt(0, j-1).toString());
        }
    }
    for(int t=0;t<nombres.size();t++)
    {
        for(int r=0;r<tamEstrato;r++)
        {
            arregloT[t][r]=arreglo[r][t];
        }
    }
    TablaEstratos te=new TablaEstratos(arregloT,n);
    return te;
}

```

```

public TableModel pulir_tabla(TableModel tabla, LinkedList nombres, LinkedList indices)
{
    Object[] n=new Object[nombres.size()];
    Object[][]nn=new Object[tabla.getRowCount()][tabla.getRowCount()];
    for(int ii=0;ii<nombres.size();ii++)
        n[ii]=nombres.get(ii).toString(); //encabezado de la tabla
    //llenado de la tabla
    int jmp=0;
    for(int i=0;i<nombres.size();i++)
    {
        for(int j=0;j<Integer.parseInt(indices.get(i).toString());j++)
        {
            nn[j][i]=tabla.getValueAt(j+jmp,i);
        }
        jmp=jmp+Integer.parseInt(indices.get(i).toString());
    }
}

```

```

    }
    TableModel tm=new javax.swing.table.DefaultTableModel(nn,n);
    return tm;
}

public TablaMuestra ponderadoSinReemplazo(LinkedList datos, LinkedList
ponderaciones,LinkedList acumulada,String tamM,String var)
{
    int n=Integer.parseInt(tamM);
    LinkedList muestra=new LinkedList();
    LinkedList nombres=new LinkedList();
    LinkedList ids=new LinkedList();
    LinkedList[] mm=new LinkedList[2];
    nombres.add("ID");
    nombres.add(var);
    Operaciones op=new Operaciones();
    int y=0;
    //muestra
    while(y<n)
    {
        double num=op.aleatorio();
        for(int i=0;i<n;i++)
        {
            if(num<Double.parseDouble(acumulada.get(i).toString()) || num<Double.parseDouble(acumulada.g
            et(i+1).toString()))
            {
                muestra.add(datos.get(i));
                ids.add(i);
                y=y+1;
                break;
            }
        }
    }
    //fin de la muestra
    //armar la tabla
    mm[0]=ids;
    mm[1]=muestra;
    TablaMuestra mu=new TablaMuestra(nombres,mm);
    return mu;
}
}

```

## 9.5 Paquete Estadística

### 9.5.1 Operaciones.java

```
/*
 * Esta clase contiene todas las operaciones necesarias para los cálculos estadísticos de los
muestreos
 */
package pt02.estadistica;
import java.util.LinkedList;
import java.util.List;
/**
 * @author XiMe
 */
public class Operaciones
{
    public double promedio(LinkedList lista) //devuelve el promedio de una lista de datos
    {
        int tamaño=lista.size();
        double suma=0,doble=0;
        String ay=null;
        if(tamaño==0)
            return 0;
        else
        {
            for(int i=0;i<tamaño;i++)
            {
                Object aux=lista.get(i);
                ay=aux.toString();
                doble=Double.parseDouble(ay);
                suma=suma+doble;
            }
            return suma/tamaño;
        }
    }

    public double min(LinkedList lista) //devuelve el mínimo elemento de una lista
    {
        double[] ui=new double[lista.size()];
        Object r=null;
        String a=null;
    }
}
```

```

for(int i=0;i<lista.size();i++)
{
    r=lista.get(i);
    a=r.toString();
    ui[i]=Double.parseDouble(a);
}
double min=ui[0];
for(int q=1;q<lista.size();q++)
{
    min=Math.min(min,ui[q]);
}
return min;
}

```

```

public double max(LinkedList lista) //devuelve el máximo elemento de una lista
{
    double[] ui=new double[lista.size()];
    Object r=null;
    String a=null;
    for(int i=0;i<lista.size();i++)
    {
        r=lista.get(i);
        a=r.toString();
        ui[i]=Double.parseDouble(a);
    }
    double max=ui[0];
    for(int q=1;q<lista.size();q++)
    {
        max=Math.max(max,ui[q]);
    }
    return max;
}

```

```

public double varianza(LinkedList lista,double prom) //varianza
{
    double res=0,y=0,cuad=0;
    double[] ui=new double[lista.size()];
    double pi=lista.size();
    pi=1/pi;
    Object r=null;
    String a=null;
    for(int i=0;i<lista.size();i++)

```

```

    {
        r=lista.get(i);
        a=r.toString();
        ui[i]=Double.parseDouble(a);
    }

    for(int i=0;i<lista.size();i++)
    {
        y=ui[i]-prom;
        cuad=Math.pow(y,2);
        res=res+cuad;
    }
    return res*pi;
}

public double desv(double var) //desviación estándar
{
    double res=0;
    res=Math.sqrt(var);
    return res;
}

public LinkedList aleatorio(int tamaño) //aleatorio entero
{
    LinkedList aleatorio=new LinkedList();
    double aux=0;
    for(int i=0;i<tamaño*2;i++)
    {
        double rndm=Math.random()*tamaño;
        aux=Math.ceil(rndm-1);
        if(aux>0)
            aleatorio.add(aux);
        else
            aleatorio.add((double)0);
    }
    return aleatorio;
}

public double aleatorio() //aleatorio punto flotante
{

```

```

double aleatorio=0;
aleatorio=Math.random();
return aleatorio;
}

public LinkedList quicksort(int[] vector, int primero, int ultimo)
{
    LinkedList ok=new LinkedList();
    ultimo=ultimo-1;
    int i=primero, j=ultimo;
    int pivote=vector[(primero + ultimo) / 2];
    int auxiliar;

    do{
        while(vector[i]<pivote) i++;
        while(vector[j]>pivote) j--;

        if (i<=j){
            auxiliar=vector[j];
            vector[j]=vector[i];
            vector[i]=auxiliar;
            i++;
            j--;
        }

    } while (i<=j);

    if(primero<j) quicksort(vector,primero, j);
    if(ultimo>i) quicksort(vector,i, ultimo);

    for(int g=0;g<vector.length;g++)
        ok.add(vector[g]);
    return ok;
}

public int[] enteros(LinkedList datos)
{
    int[] resultado=new int[datos.size()];
    for(int i=0;i<datos.size();i++)
    {
        Object a=datos.get(i);
        String h=a.toString();
    }
}

```

```

        if(h.equals("-0.0"))
            h=h.replace("-0.0","0");
        else
            h=h.replace(".0","");
        h=h.trim();
        resultado[i]=Integer.parseInt(h);
    }
    return resultado;
}

```

```

public LinkedList sublista(int[] num,int tam)
{
    LinkedList metido=new LinkedList();
    List res=null;
    LinkedList ret=new LinkedList();

    metido.add(num[0]);
    for(int i=1;i<num.length;i++)
    {
        if(!metido.contains(num[i]))
            metido.add(num[i]);
    }

    if(metido.size()>=tam)
    {
        res=metido.subList(0, tam);
        for(int r=0;r<res.size();r++)
            ret.add(res.get(r));
        return ret;
    }
    return null;
}

```

```

public int[] enterosM(int[] numeros)
{
    int[] num=new int[numeros.length];
    for(int i=0;i<numeros.length;i++)
        num[i]=numeros[i]-1;
    return num;
}

```



```

public double factorial(int n)
{
    if ((n==0) || (n==1))
        return 1;
    else
        return n*factorial(n-1);
}

```

```

public int combinacion(int n,int r)
{
    double N=factorial(n);
    double NmenosR=factorial(Math.abs(n-r)); //no se si esté bien CHECAR
    double R=factorial(r);
    double div=N/(R*NmenosR);
    return (int)div;
}

```

public boolean repetido(LinkedList datos, int numero) //busca en una lista ligada si un elemento a introducir esta repetido

```

{
    boolean indicador=false;
    int comp=0;
    for(int i=0;i<datos.size();i++)
    {
        comp=Integer.parseInt(datos.get(i).toString().trim());
        if(numero==comp)
        {
            indicador=true;
            break;
        }
    }
    return indicador;
}

```

```

public double[] flotantes(LinkedList datos)
{
    double[] resultado=new double[datos.size()];
    for(int i=0;i<datos.size();i++)
    {
        Object a=datos.get(i);
        String h=a.toString().trim();
        resultado[i]=Double.parseDouble(h);
    }
}

```

```

    }
    return resultado;
}

public int[] shellSort(int[] a)
{
    for ( int increment = a.length / 2;
        increment > 0;
        increment = (increment == 2 ? 1 : (int) Math.round(increment / 2.2)))
    {
        for (int i = increment; i < a.length; i++)
        {
            for (int j = i; j >= increment && a[j - increment] > a[j]; j -= increment)
            {
                int temp = a[j];
                a[j] = a[j - increment];
                a[j - increment] = temp;
            }
        }
    }
    return a;
}

```

```

public LinkedList quicksort_flotante(double[] vector, int primero, int ultimo)
{
    int i=primero, j=ultimo;
    double pivote=vector[(primero + ultimo) / 2];
    double auxiliar;
    LinkedList ordenados=new LinkedList();

    do{
        while(vector[i]<pivote) i++;
        while(vector[j]>pivote) j--;

        if (i<=j){
            auxiliar=vector[j];
            vector[j]=vector[i];
            vector[i]=auxiliar;
            i++;
            j--;
        }
    }
}

```

```

        } while (i<=j);

        if(primero<j)
        quicksort_flotante(vector,primero,j);
        if(ultimo>i)
        quicksort_flotante(vector,i, ultimo);

    for(int k=0;k<vector.length;k++)
        ordenados.add(vector[k]);

    return ordenados;
}
}

```

## 9.6 Paquete ModelosTablas

### 9.6.1 TablaInicial.java

```

/*
 * Esta clase crea un modelo de tabla para poder crear el JTable de
 * los datos iniciales
 */

package pt02.modelosTablas;

import java.util.LinkedList;
import javax.swing.event.TableModelListener;
import javax.swing.table.TableModel;
/**
 * @author XiMe
 */
public class TablaInicial implements TableModel
{
    private LinkedList nombres;
    private LinkedList[] valores;

    public TablaInicial(LinkedList nom,LinkedList[] vals)
    {
        this.nombres=nom;
        this.valores=vals;
    }
}

```

```

public int getRowCount() {
    return valores[0].size();
}

public int getColumnCount() {
    return nombres.size();
}

public String getColumnName(int columnIndex) {
    Object a=nombres.get(columnIndex);
    String aux=a.toString();
    return aux;
}

public Class<?> getColumnClass(int columnIndex) {
    return String.class;
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return true;
}

public Object getValueAt(int rowIndex, int columnIndex)
{
    switch (columnIndex)
    {
        case 0:return valores[0].get(rowIndex);
        case 1:return valores[1].get(rowIndex);
        case 2:return valores[2].get(rowIndex);
        case 3:return valores[3].get(rowIndex);
        case 4:return valores[4].get(rowIndex);
        case 5:return valores[5].get(rowIndex);
        case 6:return valores[6].get(rowIndex);
        case 7:return valores[7].get(rowIndex);
        default: return null;
    }
}

public void setValueAt(Object aValue, int rowIndex, int columnIndex)
{

```

```

        valores[columnIndex].set(rowIndex, aValue);
    }

    public void addTableModelListener(TableModelListener l) {
        //System.out.println(l.toString());
    }

    public void removeTableModelListener(TableModelListener l) {
        //System.out.println(l.toString());
    }
}

```

### 9.6.2 TablaMuestra.java

```

/*
 * Éste es el modelo de la tabla donde se despliegan los resultados de haber aplicado
 * algún método de muestreo.
 */
package pt02.modelosTablas;
import java.util.LinkedList;
import javax.swing.event.TableModelListener;
import javax.swing.table.TableModel;

/**
 *
 * @author XiMe
 */
public class TablaMuestra implements TableModel
{
    private LinkedList nombres;
    private LinkedList[] valores;

    public TablaMuestra(LinkedList nom, LinkedList[] val)
    {
        this.nombres=nom;
        this.valores=val;
    }

    public int getRowCount() {
        return valores[0].size();
    }
}

```

```

public int getColumnCount() {
    return nombres.size();
}

public String getColumnName(int columnIndex) {
    Object a=nombres.get(columnIndex);
    String aux=a.toString();
    return aux;
}

public Class<?> getColumnClass(int columnIndex) {
    return Object.class;
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return false;
}

public Object getValueAt(int rowIndex, int columnIndex)
{
    switch (columnIndex)
    {
        case 0: return valores[0].get(rowIndex);
        case 1: return valores[1].get(rowIndex);
        default: return valores[columnIndex].get(rowIndex);
    }
}

public void setValueAt(Object aValue, int rowIndex, int columnIndex)
{
    //esta tabla no es editable
}

public void addTableModelListener(TableModelListener l) {
    //System.out.println(l.toString());
}

public void removeTableModelListener(TableModelListener l) {
    //System.out.println(l.toString());
}

```

```
}
```

### 9.6.3 TablaResPob.java

```
/*  
 * Esta clase representa el modelo de tabla para el resumen de la población en el frame de  
 resultados  
 * de la población  
 */  
package pt02.modelosTablas;  
import java.util.LinkedList;  
import javax.swing.event.TableModelListener;  
import javax.swing.table.TableModel;  
/**  
 * @author XiMe  
 */  
public class TablaResPob implements TableModel  
{  
    private LinkedList nombres;  
    private LinkedList valores;  
  
    public TablaResPob(LinkedList nom, LinkedList val)  
    {  
        this.nombres=nom;  
        this.valores=val;  
    }  
  
    public int getRowCount() {  
        return valores.size();  
    }  
  
    public int getColumnCount() {  
        return nombres.size();  
    }  
  
    public String getColumnName(int columnIndex) {  
        Object a=nombres.get(columnIndex);  
        String aux=a.toString();  
        return aux;  
    }  
  
    public Class<?> getColumnClass(int columnIndex) {
```

```

        return String.class;
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return false;
    }

    public Object getValueAt(int rowIndex, int columnIndex)
    {

        if(rowIndex==0)
        {
            switch(columnIndex)
            {
                case 0: return valores.get(0);
                case 1: return valores.get(1);
                case 2: return valores.get(2);
                case 3: return valores.get(3);
                case 4: return valores.get(4);
                case 5: return valores.get(5);
                case 6: return valores.get(6);
                default: return null;
            }
        }
        else return null;
    }

    public void setValueAt(Object aValue, int rowIndex, int columnIndex)
    {
        //esta tabla no es editable
    }

    public void addTableModelListener(TableModelListener l) {
        //System.out.println(l.toString());
    }

    public void removeTableModelListener(TableModelListener l) {
        //System.out.println(l.toString());
    }
}

```



#### 9.6.4 TablaFinal.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package pt02.modelosTablas;
import java.util.LinkedList;
import javax.swing.event.TableModelListener;
import javax.swing.table.TableModel;
/**
 *
 * @author XIME
 */
public class TablaFinal implements TableModel {
    private LinkedList nombres;
    private LinkedList valores;

    public TablaFinal(LinkedList nom, LinkedList val)
    {
        this.nombres=nom;
        this.valores=val;
    }
    public int getRowCount() {
        return valores.size();
    }

    public int getColumnCount() {
        return nombres.size();
    }

    public String getColumnName(int columnIndex) {
        Object a=nombres.get(columnIndex);
        String aux=a.toString();
        return aux;
    }

    public Class<?> getColumnClass(int columnIndex) {
        return Object.class;
    }
}
```

```

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return false;
}

public Object getValueAt(int rowIndex, int columnIndex)
{
    if(rowIndex==0)
    {
        switch(columnIndex)
        {
            case 0: return valores.get(0);
            case 1: return valores.get(1);
            case 2: return valores.get(2);
            case 3: return valores.get(3);
            case 4: return valores.get(4);
            default: return null;
        }
    }
    else return null;
}

public void setValueAt(Object aValue, int rowIndex, int columnIndex)
{
    //no es editable
}

public void addTableModelListener(TableModelListener l) {
    //System.out.println(l.toString());
}

public void removeTableModelListener(TableModelListener l) {
    //System.out.println(l.toString());
}
}

```

### 9.6.5 TablaLimites.java

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

```

```

package pt02.modelosTablas;
import java.util.LinkedList;
import javax.swing.event.TableModelListener;
import javax.swing.table.TableModel;
/**
 * @author XiMe
 */
public class TablaLimites implements TableModel
{
    private LinkedList nombres;
    private LinkedList[] valores;

    public TablaLimites(LinkedList nom, LinkedList[] val)
    {
        this.nombres=nom;
        this.valores=val;
    }
    public int getRowCount() {
        return valores[0].size();
    }

    public int getColumnCount() {
        return nombres.size();
    }

    public String getColumnName(int columnIndex) {
        Object a=nombres.get(columnIndex);
        String aux=a.toString();
        return aux;
    }

    public Class<?> getColumnClass(int columnIndex) {
        return Object.class;
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {

        return true;
    }

    public Object getValueAt(int rowIndex, int columnIndex)
    {

```

```

switch (columnIndex)
{
    case 0: return valores[0].get(rowIndex);
    case 1: return valores[1].get(rowIndex);
    case 2: return valores[2].get(rowIndex);
    case 3: return valores[3].get(rowIndex);
    case 4: return valores[4].get(rowIndex);
    case 5: return valores[5].get(rowIndex);
    case 6: return valores[6].get(rowIndex);
    case 7: return valores[7].get(rowIndex);
    case 8: return valores[8].get(rowIndex);
    case 9: return valores[9].get(rowIndex);
    case 10: return valores[10].get(rowIndex);
    case 11: return valores[11].get(rowIndex);
    case 12: return valores[12].get(rowIndex);
    case 13: return valores[13].get(rowIndex);
    case 14: return valores[14].get(rowIndex);
    default: return valores[columnIndex].get(rowIndex);
}
}

public void setValueAt(Object aValue, int rowIndex, int columnIndex)
{
    switch(columnIndex)
    {
        case 1:
            valores[1].set(rowIndex, aValue);
            break;
        case 2:
            valores[2].set(rowIndex, aValue);
            break;
        default: valores[columnIndex].set(rowIndex, aValue);
            break;
    }
}

public void addTableModelListener(TableModelListener l) {
    //System.out.println(l.toString());
}

```

```

public void removeTableModelListener(TableModelListener l) {
    //System.out.println(l.toString());
}
}

```

### 9.6.6 TablaEstratos.java

```

package pt02.modelosTablas;
import javax.swing.event.TableModelListener;
import javax.swing.table.TableModel;
/**
 * @author XIME
 */
public class TablaEstratos implements TableModel
{
    private String[] nombres;
    private Double[][] valores;

    public TablaEstratos(Double[][] val,String[] nom)
    {
        this.nombres=nom;
        this.valores=val;
    }
    public int getRowCount() {
        return valores[0].length;
    }

    public int getColumnCount() {
        return nombres.length;
    }

    public String getColumnName(int columnIndex) {
        return nombres[columnIndex];
    }

    public Class<?> getColumnClass(int columnIndex) {
        return Double.class;
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return false;
    }
}

```

```
}

public Object getValueAt(int rowIndex, int columnIndex)
{

    return valores[columnIndex][rowIndex];

}

public void setValueAt(Object aValue, int rowIndex, int columnIndex)
{
    valores[columnIndex][rowIndex]=Double.parseDouble(aValue.toString());
}

public void addTableModelListener(TableModelListener l) {
    //System.out.println(l.toString());
}

public void removeTableModelListener(TableModelListener l) {
    //System.out.println(l.toString());
}

}
```

# Manual de Usuario



2011

# TABLA DE CONTENIDO

**DESCRIPCIÓN FUNCIONAL ..... 3**

**DESCRIPCIÓN DE LA APLICACIÓN ..... 4**

PANTALLA DE INICIO ..... 4

FORMATO DEL ARCHIVO INICIAL ..... 5

ALEATORIO SIMPLE NO PONDERADO ..... 7

MÉTODO SISTEMÁTICO ..... 13

MÉTODO DE MUESTREO ESTRATIFICADO ..... 19

CODIFICACIÓN MANUAL ..... 20

CODIFICACIÓN PROMEDIO ..... 26

CODIFICACIÓN ARCHIVO..... 30

**MÉTODO ALEATORIO SIMPLE PONDERADO ..... 37**

TABLA DE FRECUENCIAS ..... 38

LISTA DE PROBABILIDADES ..... 42

**INSTALACIÓN Y EJECUCIÓN DE LA APLICACIÓN ..... 46**

REQUERIMIENTOS PREVIOS ..... 46

ESTRUCTURA DE CARPETAS ..... 46



## Descripción Funcional

La aplicación computacional para la toma de muestras de una población, es un software que permite al usuario tomar muestras a partir de una población a través de diferentes métodos de muestreo.

El usuario elige el método que más le acomode y debe cargar un archivo en formato .txt que contiene la población a la cual se le va a aplicar el muestreo.

El sistema está compuesto por 5 módulos que corresponden a los diferentes métodos de muestreo que maneja:

- Aleatorio Simple No Ponderado
- Aleatorio Simple Ponderado
- Sistemático
- Estratificado
- Politápico

El funcionamiento general de la aplicación se ilustra con el siguiente diagrama de flujo:

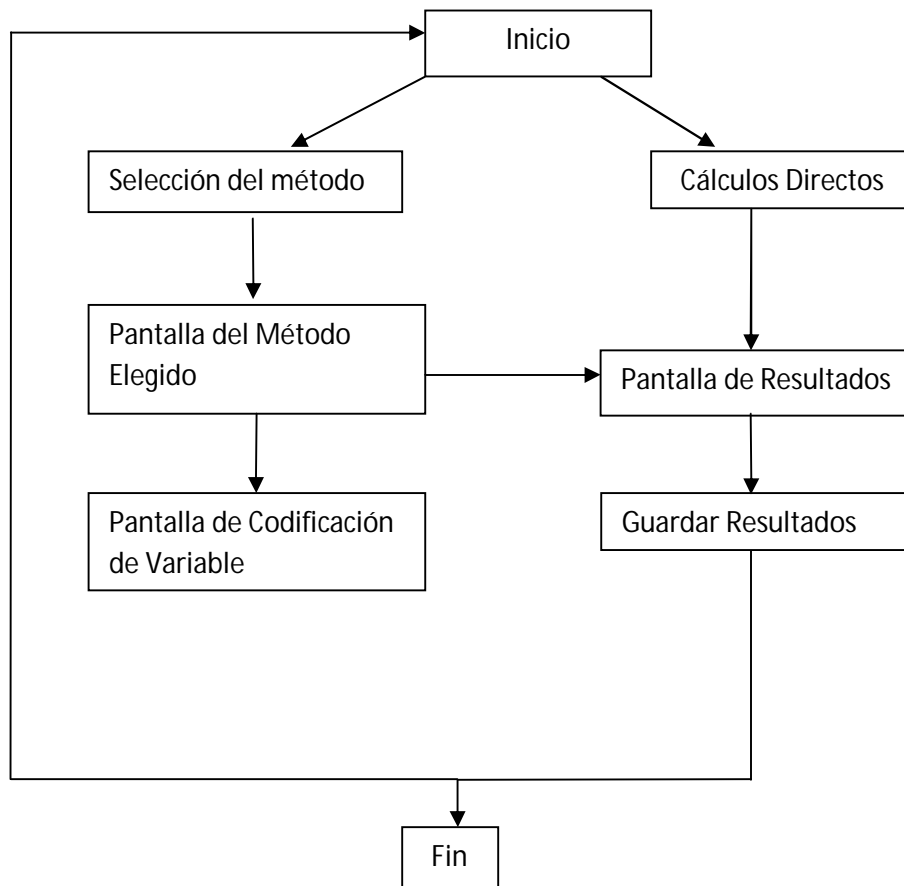


Figura 1. Diagrama de Flujo de la Aplicación

## Descripción de la Aplicación

### *Pantalla de Inicio*

Al iniciar la aplicación nos mostrará la siguiente pantalla, donde el usuario deberá dar click en el botón "Método de Muestreo" para tomar una muestra utilizando alguno de los 5 diferentes métodos, o si ya se tiene una, simplemente dar click en el botón "Estimación de Parámetros a Partir de una Muestra" para hacer los cálculos correspondientes (límite superior, límite inferior, varianza, promedio, error de estimación) a partir de un archivo que contenga una muestra.

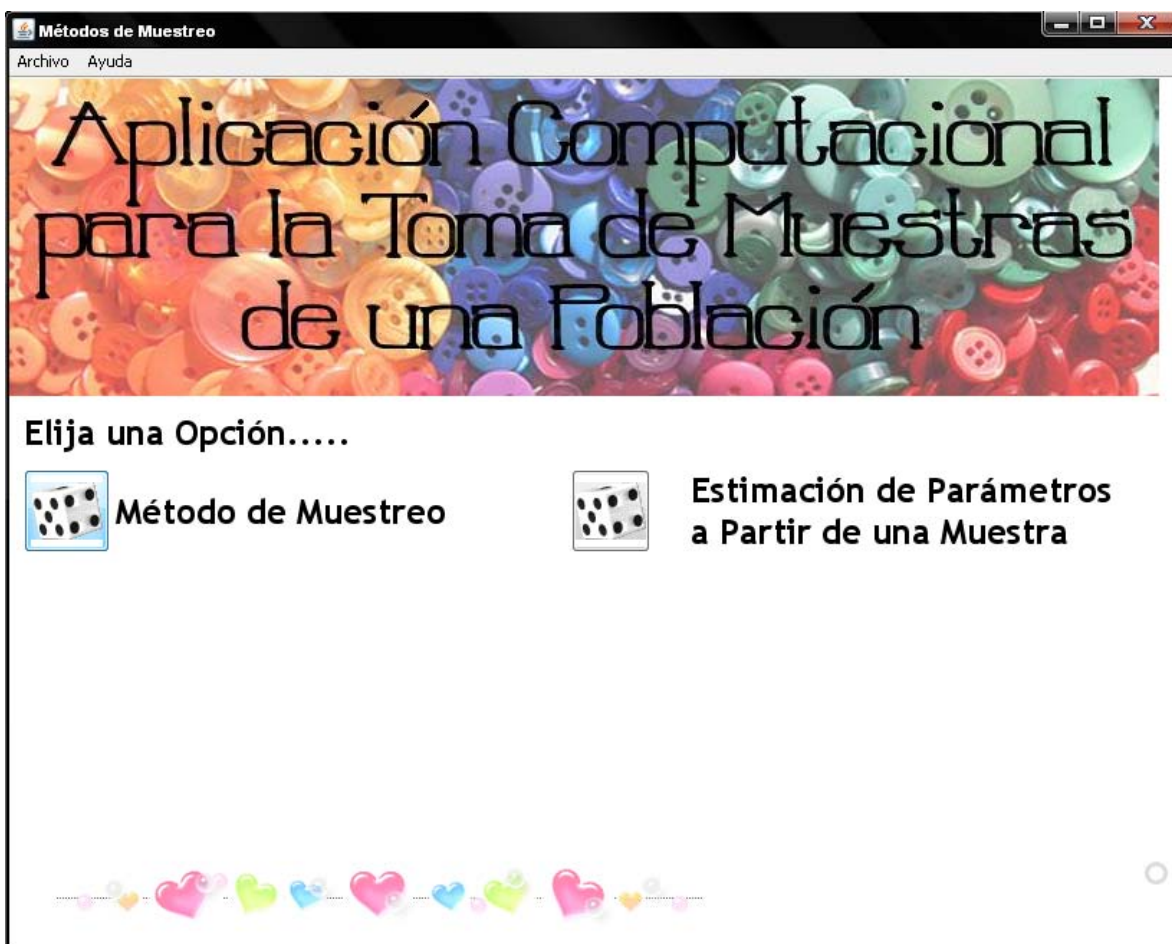


Figura 2. Pantalla de inicio de la Aplicación

Si el usuario da click en el botón "Método de Muestreo", aparecerá una lista para seleccionar el método, como se muestra en la figura 3:



Figura 3. Pantalla de Selección de Método

### ***Formato del Archivo Inicial***

La aplicación de muestreo lee un archivo que contiene la población de los datos que serán muestreados. Este archivo debe tener extensión .txt y los datos separados ya sea por espacios o por tabulaciones, en caso de tener más de 1 variable o un dato en cada línea en caso de que se tenga 1 sola variable.

Es importante que el archivo no contenga huecos entre los datos y que contenga solamente números, a excepción de la primera línea, que es la que indica el nombre de las variables que se utilizarán para el muestreo, de lo contrario la aplicación no podrá interpretar los datos correctamente.

En la figura 4 se muestra un ejemplo de archivo correcto que contiene 4 variables con los valores correspondientes a cada una.

nodos	radios	alturas	salones	
4	5	6	6	7
7	8	2	9	2
1	2	5	8	4
2	4	7	7	3
8	6	6	6	8
7	6	5	5	4
5	6	8	7	5
1	1	8	7	6
4	5	6	6	7
7	8	2	9	2
1	2	5	8	4
2	4	7	7	3
8	6	6	6	8
7	6	5	5	4
5	6	8	7	5
1	1	8	7	6
4	5	6	6	7
7	8	2	9	2
1	2	5	8	4
2	4	7	7	3
8	6	6	6	8
7	6	5	5	4

Figura 4. Ejemplo de Archivo con formato Correcto

Para el método Estratificado, el sistema interpretará el formato de archivo de 3 maneras diferentes dependiendo si el usuario definirá los estratos de manera manual, si quiere que la aplicación los divida por promedio ó si él ya tiene sus estratos definidos en el mismo archivo.

En todos los casos el archivo debe contener en la primera línea el nombre de la o las variables separadas por espacio o tabulación, seguido de un retorno de carro (enter) y las demás líneas indicarán los datos que se relacionarán a esa variable, como se muestra en la figura anterior.

Para el método Aleatorio Simple Ponderado, el sistema interpretará el formato del archivo de 2 maneras diferentes dependiendo si el usuario tiene una Tabla de Frecuencias o Probabilidades. En este caso la primera línea será interpretada como en los métodos anteriores, así que debe contener el nombre de la o las variables involucradas separadas por espacios o tabulaciones y en los siguientes renglones vendrá el dato seguido de su frecuencia o probabilidad.

Cabe mencionar que se puede tener más de una columna de frecuencias o probabilidades y datos. Lo anterior se definirá más adelante en la explicación de cada método.

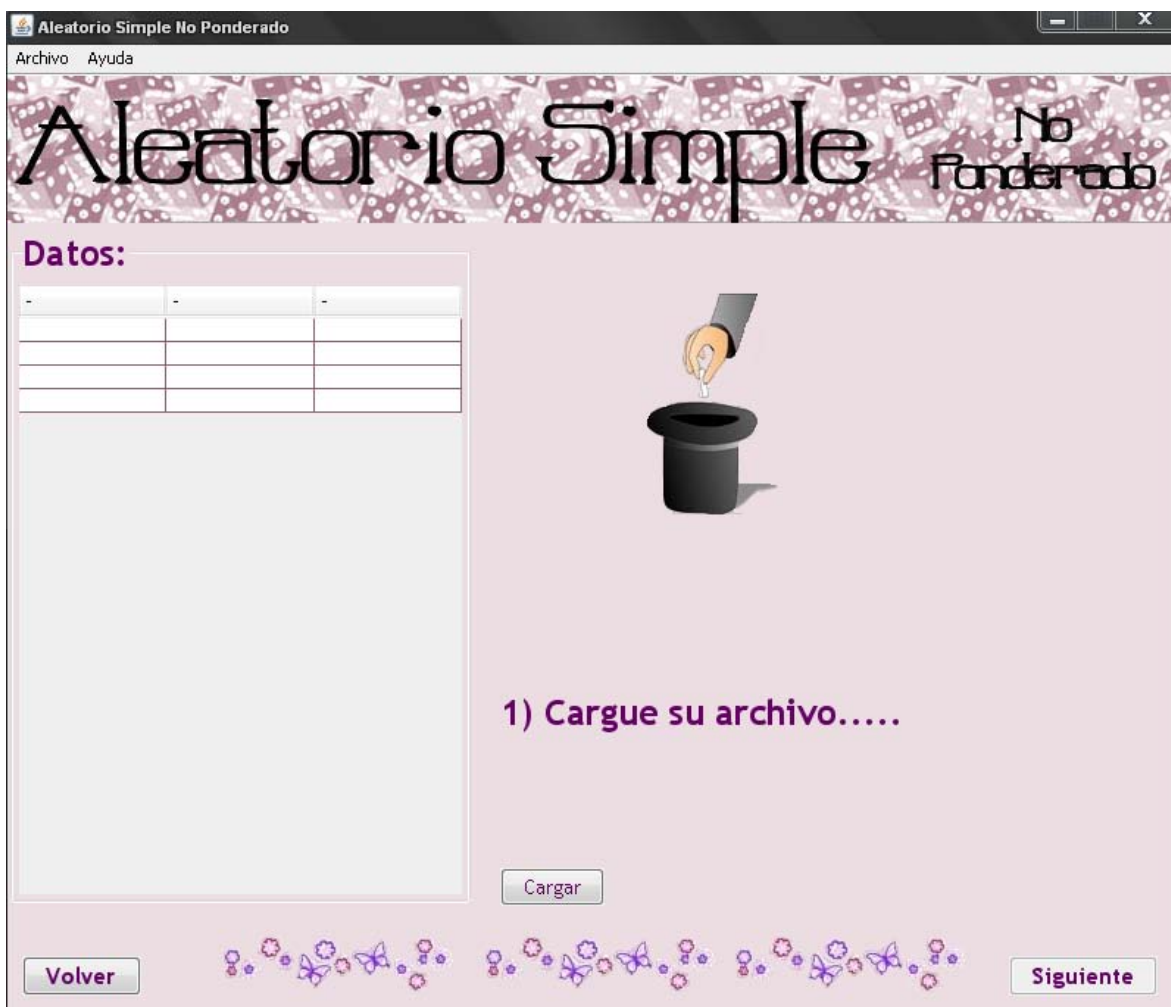
## ***Aleatorio Simple No Ponderado***

El método de muestreo aleatorio simple No Ponderado consiste en seleccionar  $n$  individuos al azar suponiendo que todos tienen la misma importancia.

En este módulo de la aplicación, el usuario debe seleccionar el archivo desde el cual se tomarán los datos, el tamaño de la muestra, la variable a muestrear y el algoritmo de selección deseado.

Una vez seleccionado, debe dar click en el botón siguiente y si no se detecta algún error, el sistema mostrará la pantalla de resultados.

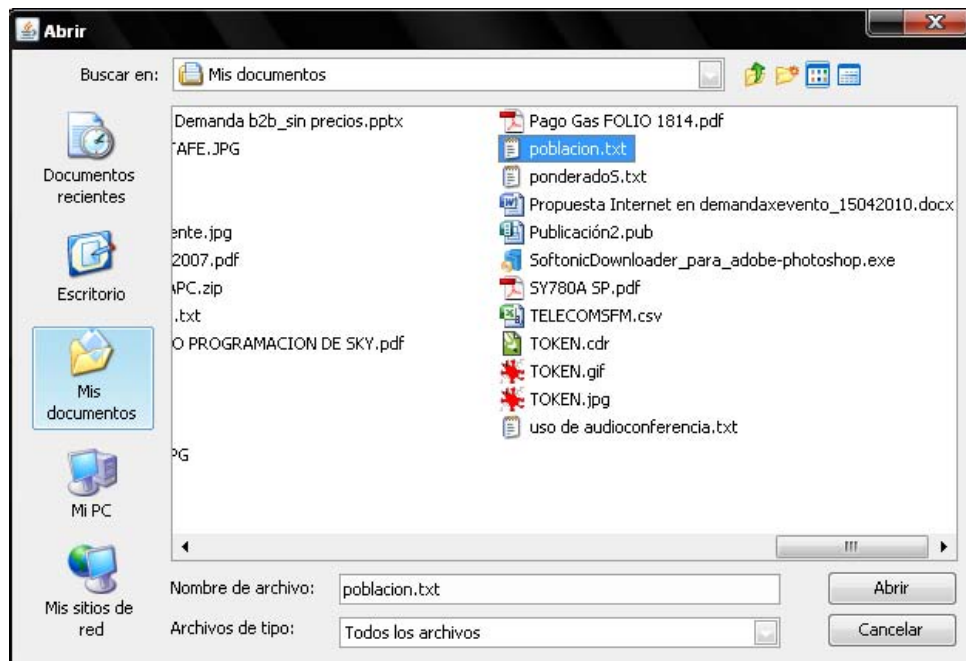
La figura 5 es una muestra de la pantalla del método aleatorio simple ponderado, después de haber elegido método en el menú.



**Figura 5. Pantalla Inicial del Método Aleatorio Simple No Ponderado**

Dando click en el botón "Cargar", aparecerá un cuadro de diálogo para que pueda seleccionar el archivo con los datos que se cargarán en el sistema. En este caso el archivo elegido se llama población.txt.

Damos click en el botón aceptar y aparecerá otra pantalla con los datos correspondientes.



**Figura 6. Cuadro de Diálogo de Elección de Archivo**



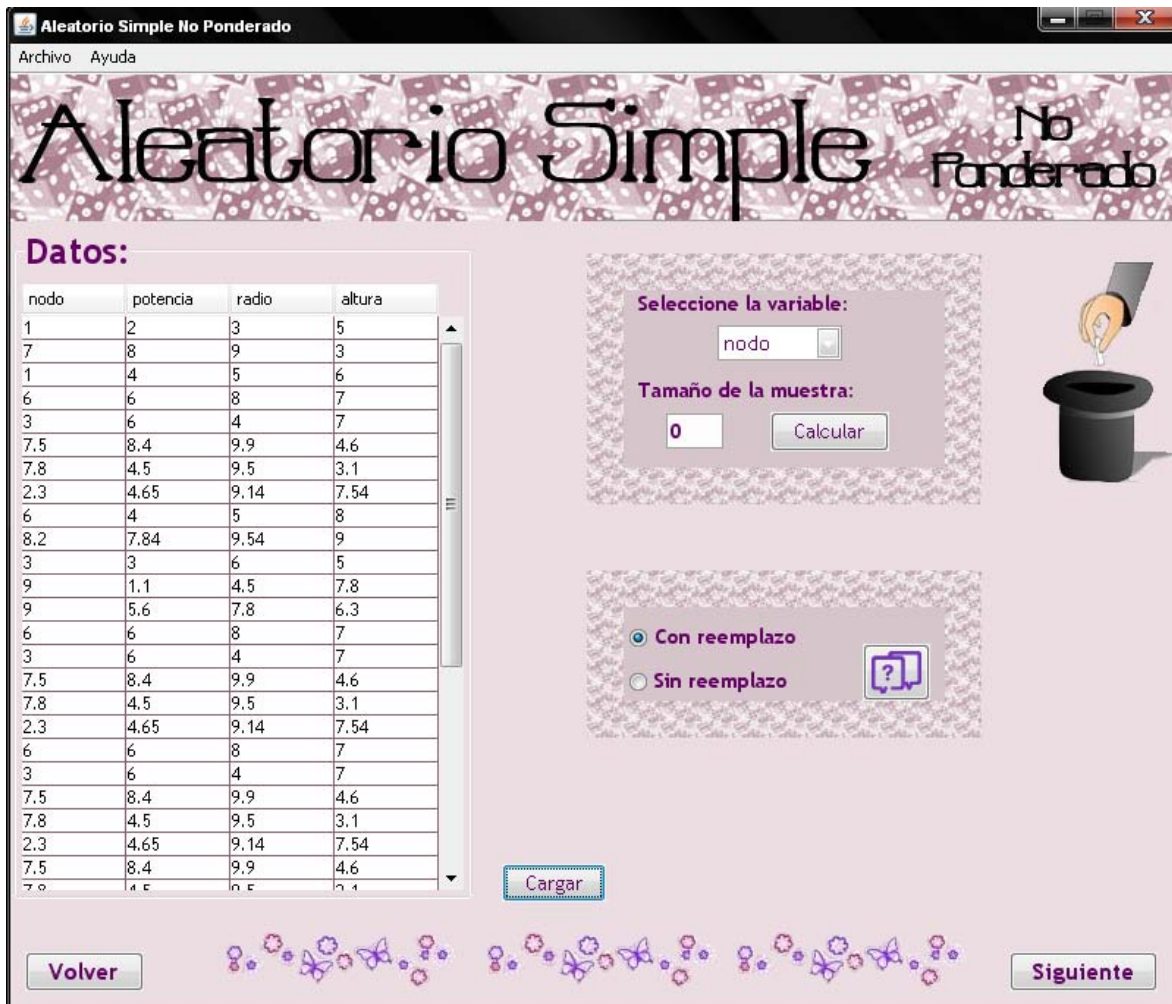


Figura 7. Pantalla del Método Aleatorio Simple No Ponderado con los datos cargados correctamente

Una vez cargados los datos, seleccionamos la variable correspondiente, en este caso NODOS, debemos elegir el tamaño de la muestra, así como el algoritmo con el que queremos se elabore la misma y dar click en el botón "Siguiente".

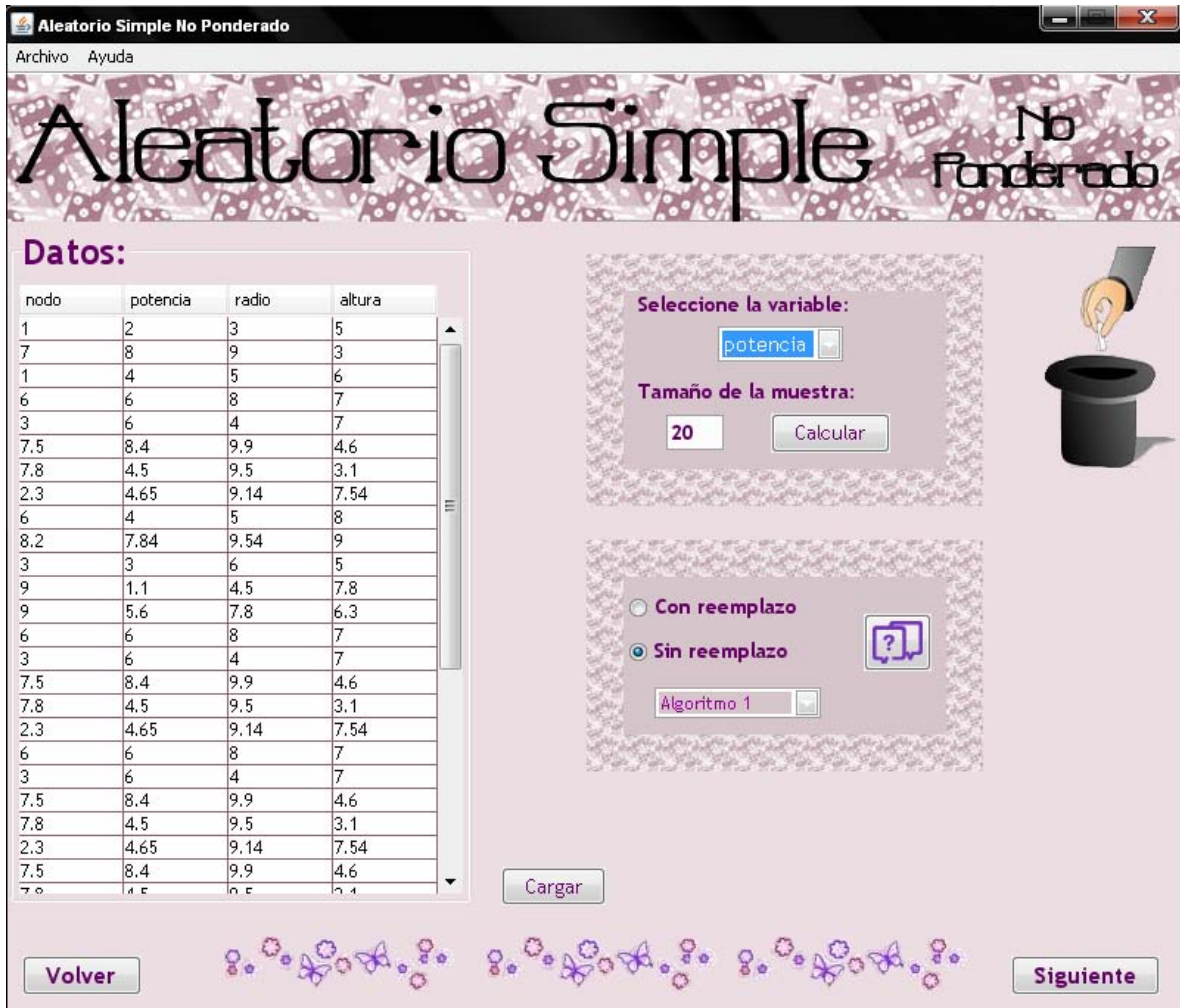


Figura 8. Pantalla del Método Aleatorio Simple No Ponderado con elección de variable

En este caso, se ha seleccionado muestrear la variable POTENCIA, el tamaño de la muestra será de 20 utilizando el algoritmo 1 sin reemplazo. Posteriormente saldrá un mensaje de confirmación y se desplegará la pantalla de resultados.





Figura 9. Pantalla de Resultados con la pestaña de Población activada.

En la figura 9 se encuentran los resultados del muestreo. En una pestaña se presentan los datos generales de la población resumidos en una tabla, los tamaños de la población y la muestra y el nombre del archivo donde se encuentra la población.

En la segunda pestaña se encuentra una tabla que contiene la muestra tomada con su valor y el id asignado. En la parte central se encuentra una tabla con los cálculos de límite inferior, límite superior, promedio, varianza y error de estimación de la muestra.

En la parte inferior derecha se encuentra el botón "Guardar" que le permitirá al usuario guardar la información calculada para posterior consulta.

**Resultados**

**Muestra:**

ID	potencia
37	1.1
36	3.0
35	7.84
34	4.0
32	4.5
30	5.6
29	1.1
27	7.84
23	8.4
21	4.5
18	6.0
17	4.65
15	8.4
13	6.0
12	5.6
8	4.0
7	4.65
5	8.4

**Resultados**

Límite Inferior	Límite Superior	Error Estimación	Promedio	Varianza
3.3717692334...	7.3865080608...	0.0	5.2790000000...	4.556969

**VOLVER** **GUARDAR**

Figura 10. Pantalla de Resultados con la pestaña de Muestra activada.

## Método Sistemático

El método Sistemático consiste en seleccionar los individuos, saltando  $k$  individuos, hasta completar los  $n$  individuos deseados.

Como en el método anterior, después de seleccionar el método en el menú, nos aparecerá una pantalla para que carguemos el archivo inicial. La pantalla se muestra en la figura 11.

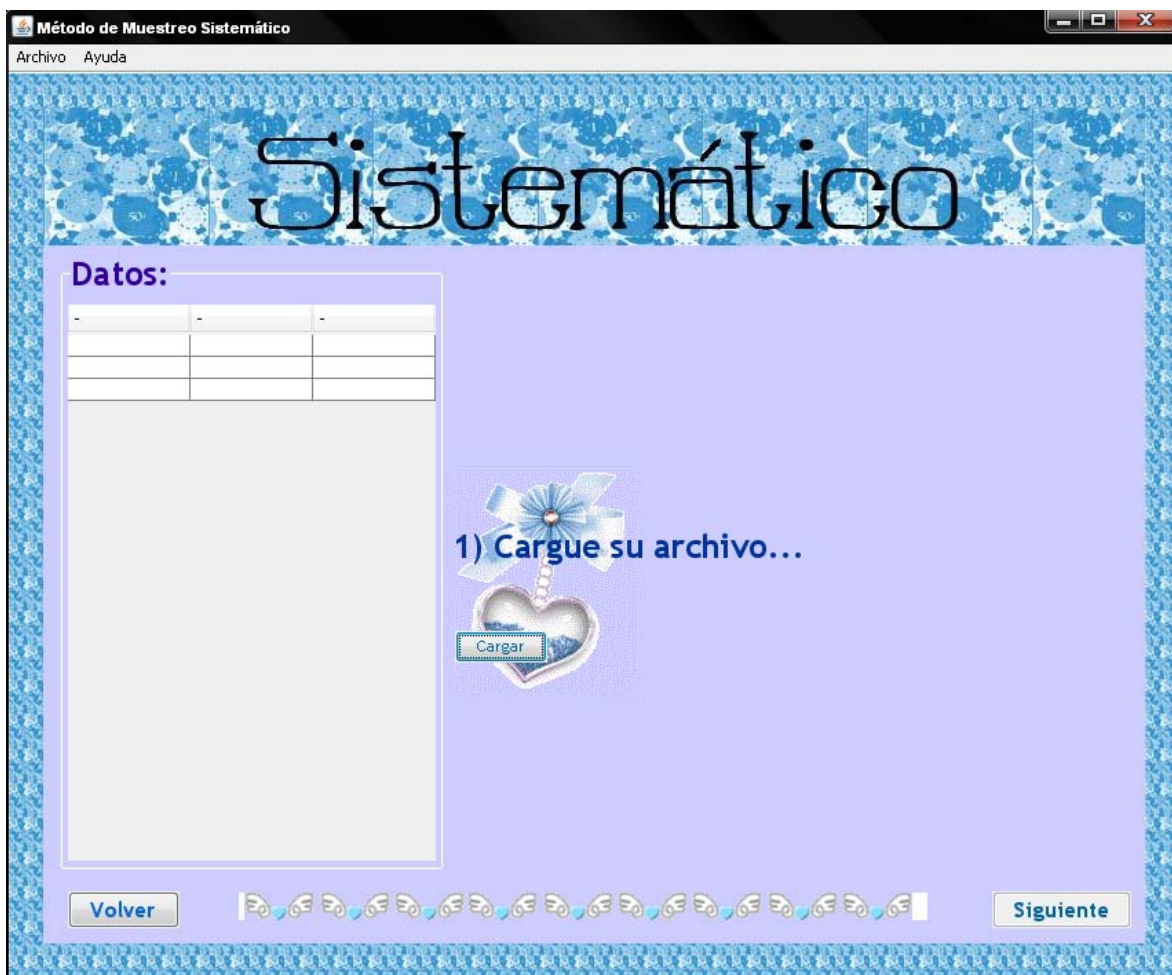


Figura 11. Pantalla Inicial del Método Sistemático

Una vez cargados los datos, se tienen 3 etapas a seguir. Primero debemos indicar el lugar de inicio del muestreo, es decir, un número que nos indique el lugar de partida. Si el usuario no sabe de dónde partir, puede elegir el 0 para partir desde el primer dato de su lista o bien dar click en el botón "Generar" para que el sistema aleatoriamente seleccione un punto de partida. Esta pantalla se muestra en la figura 12.

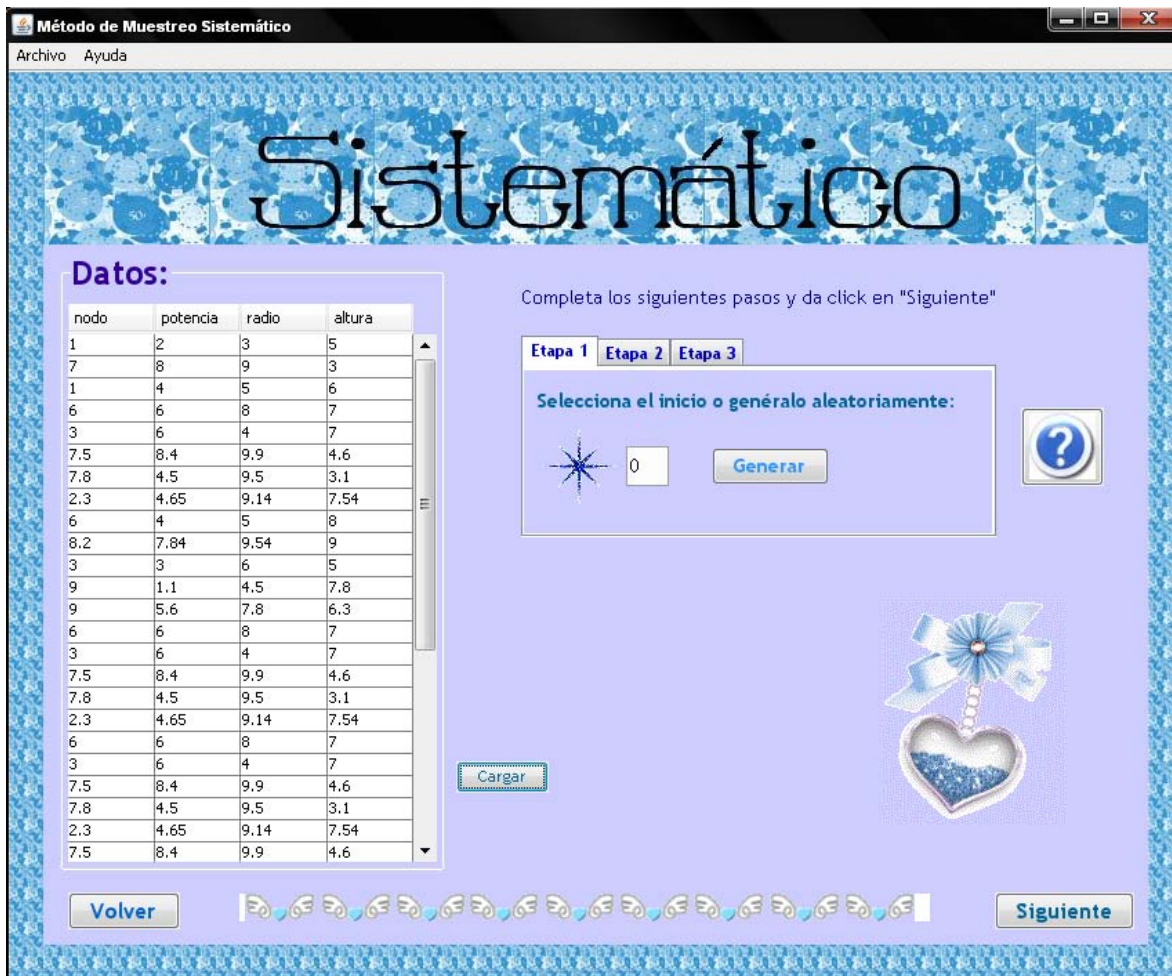


Figura 12. Método Sistemático en su primera etapa.

En la segunda etapa el usuario debe elegir el salto, es decir, cuántos lugares se va a saltar entre cada dato. Puede ser introducido manualmente por el usuario, calculado por el sistema o aleatorio.

La pantalla de la segunda etapa se muestra en la figura 13.





Figura 13. Método Sistemático en su segunda etapa.

Finalmente, la etapa 3 consiste en seleccionar la variable a muestrear y el tamaño de la muestra, como se indica en la figura 14.



Figura 14. Método Sistemático en su etapa final.

Para obtener los resultados se da click en el botón "Siguiente".

La pantalla de resultados para este ejemplo se muestra en la figura 15.



Figura 15. Pantalla de Resultados con la pestaña de Población activada.

En la pantalla de resultados se muestra una tabla correspondiente al resumen de la población. En la siguiente pestaña, la que dice “Muestra” se encuentra la muestra seleccionada, junto con un botón de “Guardar” para que el usuario pueda guardarla en un archivo de texto y usarla para estudios posteriores.

En la figura 16 se muestra la pantalla de resultados con la pestaña de muestra activada.



Figura 16. Pantalla de Resultados con la pestaña de Muestra activada.



## Método de Muestreo Estratificado

El método de muestreo estratificado consiste en separar la lista inicial de tamaño  $N$  en  $H$  estratos y seleccionar  $n_i$  individuos a cada estrato  $H$  aplicando algún algoritmo del método aleatorio simple, con la restricción de que  $n = n_1 + n_2 + n_3 + \dots + n_h$ .

Al iniciar el método lo primero que debemos hacer es definir como serán interpretados nuestros datos iniciales. Recordemos que este método interpreta el archivo inicial de 3 maneras diferentes (ver la sección Formato de Archivo Inicial). Una vez elegida la manera de escoger los estratos debemos dar click en el botón "Cargar" como se muestra en la figura 17.



Figura 17. Pantalla de Inicio del Método Estratificado

## Codificación Manual

Si vamos a codificar manualmente las variables nos aparecerá la figura 18, donde debemos seleccionar la variable para codificar y dar click en el botón "Siguiente".



**Estratificado**

Archivo Ayuda

# Estratificado

**Datos:**

voltaje	c1	c2	c3
5	1	5	4
6	2	6	4
7	0	7	a
4	1	6	a
3	0	5	a
2	0	5	1
3	1	6	1
2	0	6	1
4	0	5	4
4	2	7	4
3	2	7	1
5	1	5	1
3	2	6	a
4	1	7	a
5	1	7	a
6	2	5	4
5	2	6	a
6	1	6	1
5	0	5	a
7	0	5	a
7	1	7	4
4	1	5	4
5	2	6	1
2	1	7	a
3	1	5	a
4	0	6	4

¿Cómo escogerá sus estratos?

Manual

Promedio

Archivo

Selecione su variable:

voltaje

Figura 18. Pantalla del Método Estratificado con selección Manual

Después de recibir un mensaje de confirmación nos aparecerá la figura 19 que corresponde al módulo de codificación de variables. En este módulo debemos llenar la tabla correspondiente con los códigos que le vamos a asignar a cada estrato y posteriormente seleccionar el método de selección de la muestra ya sea con reemplazo o alguno de los algoritmos implementados sin reemplazo (ver Método Aleatorio Simple). Para continuar debe dar click en el botón "Siguiente".

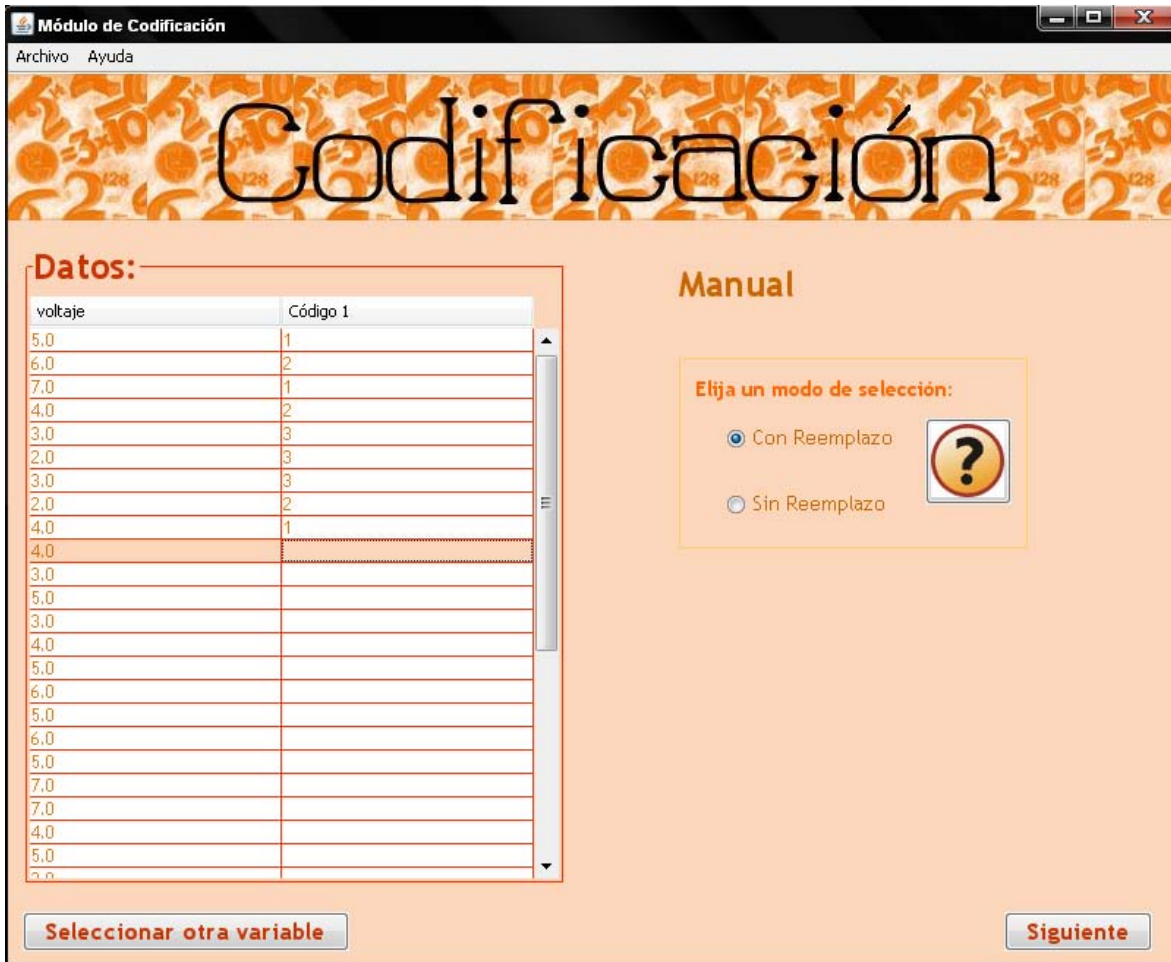


Figura 19. Pantalla de Codificación de Variable, elección de estratos manual

Si llegara a cometer el error de dejar espacios en blanco en el llenado, el sistema se lo indicará para que llene correctamente la tabla. Posteriormente el sistema contará los estratos y le preguntará si es correcto el conteo, como se muestra en la figura 20.

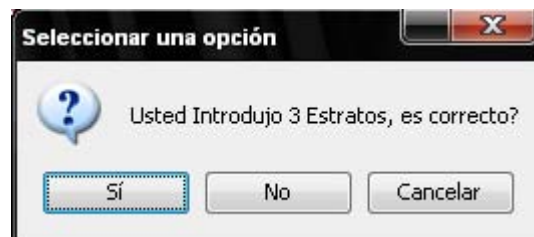
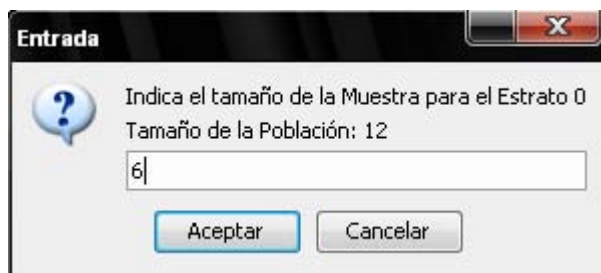


Figura 20. Conteo de Estratos

En caso de contestar que si, el sistema le pedirá introducir el tamaño de la muestra para cada estrato, indicándole en todo momento el tamaño de cada estrato para que el usuario no exceda el número pedido. En caso de que el tamaño de la muestra sea mayor al de la población, el sistema se lo hará saber al usuario para que corrija su entrada de datos.



**Figura 21. Entrada de tamaño de la muestra para los estratos.**

Una vez indicado el tamaño de la muestra para cada estrato, aparecerá la pantalla de resultados.

En la primera pestaña se encuentran las estimaciones de la población completa y una tabla resumen que contiene información acerca del nombre del archivo que utilizó el usuario, cuántos estratos eligió, la variable seleccionada y el tamaño de la población de cada estrato.

En la segunda pestaña se encuentra una tabla con la muestra elegida para cada estrato con su respectivo valor y en la tercera pestaña se despliega una tabla con las estimaciones calculadas para cada estrato.

En la parte inferior de la pantalla se encuentran los botones "Volver" para retroceder en los pasos anteriores y el botón "Guardar" para registrar toda la información desplegada en un archivo de texto para usos posteriores.

Las figuras 22,23 y 24 ejemplifican lo anterior.



Figura 22. Pantalla de Resultados con la primera pestaña activada.







Figura 24. Pantalla de Resultados con la tercera pestaña activada.

## Codificación Promedio

The screenshot shows the 'Estratificado' software interface. The title bar includes 'Estratificado', 'Archivo', and 'Ayuda'. The main title 'Estratificado' is displayed in a large, stylized font. Below the title, there is a section labeled 'Datos:' containing a table with four columns: 'voltaje', 'c1', 'c2', and 'c3'. The table contains 20 rows of data. To the right of the table, there is a section titled '¿Cómo escogerá sus estratos?' with three radio button options: 'Manual', 'Promedio' (which is selected), and 'Archivo'. A 'Cargar' button is located next to the 'Promedio' option. Below this section is a yellow star icon. To the right of the star is a section titled 'Seleccione su variable:' with a dropdown menu showing 'voltaje'. At the bottom left, there is a 'Volver' button, and at the bottom right, there is a 'Siguiente' button.

voltaje	c1	c2	c3
5	1	5	4
6	2	6	4
7	0	7	a
4	1	6	a
3	0	5	a
2	0	5	1
3	1	6	1
2	0	6	1
4	0	5	4
4	2	7	4
3	2	7	1
5	1	5	1
3	2	6	a
4	1	7	a
5	1	7	a
6	2	5	4
5	2	6	a
6	1	6	1
5	0	5	a
7	0	5	a
7	1	7	4
4	1	5	4
5	2	6	1
2	1	7	a
3	1	5	a
4	0	6	4

Figura 25. Método Estratificado, elección por promedio

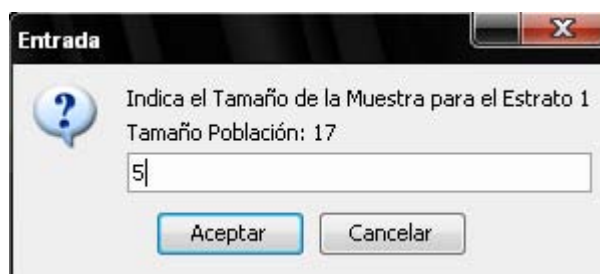
Al elegir la codificación promedio, una vez que hemos cargado el archivo inicial, daremos click en el botón "Siguiente" y nos aparecerá la figura 26, donde la población será dividida en 2 estratos. El primer estrato será codificado con un "0". Éste indica que esos valores están por debajo del promedio total de la población, y los que tengan el código "1", estarán por arriba del promedio de la población. El sistema entonces desplegará una tabla con los estratos correspondientes y el usuario tendrá la opción de invertir los códigos y de elegir el método de toma de muestra ya sea con reemplazo o alguno de los algoritmos implementados sin reemplazo (ver Método Aleatorio Simple). Para continuar debe dar click en el botón "Siguiente".





Figura 26. Módulo de Codificación por promedio.

Una vez elegido el modo de seleccionar la muestra, el sistema le pedirá introducir el tamaño de la muestra para cada estrato, indicándole en todo momento el tamaño de cada estrato para que el usuario no exceda el número pedido. En caso de que el tamaño de la muestra sea mayor al de la población, el sistema se lo hará saber al usuario para que corrija su entrada de datos.



Una vez indicado el tamaño de la muestra para cada estrato, aparecerá la pantalla de resultados.

En la primera pestaña se encuentran las tablas con la muestra elegida para ambos estratos con su respectivo valor y una tabla resumen que contiene el nombre del archivo seleccionado, el tamaño elegido de la muestra y el método de selección de la misma

En la segunda pestaña se despliega una tabla con las estimaciones calculadas para cada estrato.

En la parte inferior de la pantalla se encuentran los botones "Volver" para retroceder en los pasos anteriores y el botón "Guardar" para registrar toda la información desplegada en un archivo de texto para usos posteriores.

Las figuras 27 y 28 ejemplifican lo anterior.



Figura 27. Pantalla de Resultados con la primera pestaña activada.



Figura 28. Pantalla de Resultados con la segunda pestaña activada.

## Codificación Archivo

Para los casos en los que los códigos asignados a los estratos ya se encuentren en un archivo, se deberá de elegir esta opción. Se desplegará una pantalla donde el usuario debe indicar al sistema que columna corresponde al código y que columna corresponde a la variable como se muestra en la figura 29.

The screenshot shows the 'Estratificado' software interface. The title bar includes 'Estratificado', 'Archivo', and 'Ayuda'. The main header features the word 'Estratificado' in a large, stylized font. Below the header, there is a 'Datos:' section containing a table with four columns: 'voltaje', 'c1', 'c2', and 'c3'. The table contains 20 rows of data. To the right of the table, there is a configuration panel titled '¿Cómo escogerá sus estratos?' with three radio button options: 'Manual', 'Promedio', and 'Archivo'. The 'Archivo' option is selected. Below these options is a 'Cargar' button and a 'Selección columna de Código' dropdown menu with 'c1' selected. A question mark icon is also present. Below the configuration panel is a large gold star graphic and a 'Selección su variable:' dropdown menu with 'voltaje' selected. At the bottom of the interface are 'Volver' and 'Siguiente' buttons.

voltaje	c1	c2	c3
5	1	5	4
6	2	6	4
7	0	7	a
4	1	6	a
3	0	5	a
2	0	5	1
3	1	6	1
2	0	6	1
4	0	5	4
4	2	7	4
3	2	7	1
5	1	5	1
3	2	6	a
4	1	7	a
5	1	7	a
6	2	5	4
5	2	6	a
6	1	6	1
5	0	5	a
7	0	5	a
7	1	7	4
4	1	5	4
5	2	6	1
2	1	7	a
3	1	5	a
4	0	6	4

Figura 29. Pantalla método Estratificado con estratos en un archivo inicial.

En este ejemplo se ha elegido como variable de codificación la que lleva por nombre C1 y la variable que contiene los datos se llama voltaje. Al dar click en "Siguiente" aparecerá un diálogo de confirmación y la figura 30, donde el usuario debe indicar el tamaño total de la muestra sin tomar en cuenta cómo repartirla en cada estrato, ya que el sistema hará esos cálculos posteriormente y le indicará el usuario el número de estratos y el tamaño recomendado de cada estrato.



Figura 30. Método Estratificado, repartición de estratos.

Al dar click en el botón "OK" el sistema automáticamente calculará el tamaño de la población para cada uno de los estratos de 3 formas diferentes: proporcional, costo y varianza. El usuario tiene la opción también de elegir el tamaño de la población de los estratos elegidos e introducirlos en la tabla. Posteriormente elegirá el tamaño según su conveniencia. Un ejemplo de lo anterior se muestra en la figura 31.





Figura 31. Tabla de elección de tamaño de muestra de cada estrato.

Para este ejemplo, se eligió los tamaños de la población de manera proporcional. Si el usuario quisiera elegir los tamaños manualmente, es responsable de hacer la repartición de manera correcta, ya que si los tamaños elegidos no suman en este caso 20, el sistema le marcará un error que deberá corregir para continuar.

En esta tabla se presenta también el código asignado a cada estrato, así como el número de estratos calculados y el tamaño total de cada estrato. La interpretación de la tabla, al haber elegido el tamaño de muestra proporcional, en este ejemplo, debe ser de la siguiente manera: hay 3 estratos, para el primero de tamaño 18 se elegirán 9 individuos, para el segundo de tamaño 8 se elegirán 4 individuos y para el tercero de tamaño 13 se elegirán 7 individuos.

Al dar click en el botón "Siguiente" aparecerá la figura 32. Esta pantalla contiene los datos separados en cada estrato respectivamente según la elección anterior y el usuario tendrá la opción de elegir el método de toma de muestra ya sea con reemplazo o alguno de los algoritmos



En la parte inferior de la pantalla se encuentran los botones “Volver” para retroceder en los pasos anteriores y el botón “Guardar” para registrar toda la información desplegada en un archivo de texto para usos posteriores.

Las figuras 33, 34 y 35 ejemplifican lo anterior.



Figura 33. Pantalla de Resultados del método Estratificado Archivo. Pestaña Población







Figura 35. Pantalla de Resultados del método Estratificado Archivo. Pestaña Cálculos

## Método Aleatorio Simple Ponderado

Este método consiste en seleccionar  $n$  individuos al azar, tomando en cuenta la importancia de los individuos en la lista de tamaño  $N$ . Se toma la muestra considerando la importancia (ponderación).



Figura 36. Método Aleatorio Simple Ponderado

Al iniciar este método, se debe cargar el archivo que contenga la población. Como se vio en la sección Formato de Archivo Inicial, con este método se pueden introducir datos en forma de tabla de frecuencias o probabilidades.

## Tabla de Frecuencias

prueba	frecuencias
10.1	4
12	3
12.98	5
13.3	2
10.5	6
9.2	2
8.5	3
8.88	1

Figura 37. Método Aleatorio Simple Ponderado. Tabla de Frecuencias.

Si el usuario quiere utilizar una tabla de frecuencias para elegir su muestra, deberá indicarlo en la selección del tipo de archivo. También debe seleccionar si la muestra será elegida con o sin reemplazo y en su caso el algoritmo a utilizar para tal selección. Posteriormente, deberá seleccionar el nombre de la variable a muestrear, el tamaño de la muestra y el nombre de la variable que corresponde a las frecuencias.

Es importante mencionar que si el usuario introduce un tamaño de muestra mayor al de la población, el sistema le pedirá corregir el dato, ya que se calcula automáticamente.

En este ejemplo, solo tenemos una variable, pero cabe mencionar que el usuario en su archivo puede manejar más variables y frecuencias, sin embargo para el muestreo solo se podrá seleccionar una de cada una.

Al dar click en el botón "Siguiente", se mostrará la figura 38 que contiene los resultados obtenidos después del muestreo.

En la primera pestaña se encuentran las estimaciones de la población completa y una tabla resumen que contiene información acerca del nombre del archivo que utilizó el usuario, tamaño de la muestra, la variable seleccionada, método de selección y algoritmo elegido.

En la segunda pestaña se encuentra una tabla con la muestra y en la tercera pestaña se despliega una tabla con las estimaciones calculadas de la muestra.

En la parte inferior de la pantalla se encuentran los botones “Volver” para retroceder en los pasos anteriores y el botón “Guardar” para registrar toda la información desplegada en un archivo de texto para usos posteriores.

Las figuras 38, 39 y 40 ejemplifican lo anterior.



Figura 38. Resultados Método Aleatorio Simple Ponderado. Pestaña Población



Figura 39. Resultados Método Aleatorio Simple Ponderado. Pestaña Muestra



Figura 40. Resultados Método Aleatorio Simple Ponderado. Pestaña Cálculos



## Lista de Probabilidades

valores	frecuencia
9	0.05
7	0.2
8	0.02
6	0.03
4	0.5
2	0.1
3	0.1

Figura 41. Método Aleatorio Simple Ponderado. Lista de Probabilidades.

Si el usuario quiere utilizar una lista de probabilidades para elegir su muestra, deberá indicarlo en la selección del tipo de archivo. Posteriormente, deberá seleccionar el nombre de la variable a muestrear, el tamaño de la muestra y el nombre de la variable que corresponde a las probabilidades.

Cabe destacar que el usuario es responsable de asignar dichas probabilidades ya que al momento de sacar los acumulados, la última suma siempre debe ser 1. De no ser así, el sistema no aceptará las probabilidades propuestas.

Es importante mencionar que en este caso, el usuario puede introducir un número mayor al tamaño de la población, ya que por el método, algunos individuos tienen más probabilidad de ser elegidos que otros.



En este ejemplo, solo tenemos una variable, pero el usuario en su archivo inicial puede manejar más variables y probabilidades, aunque para el muestreo sólo se podrá seleccionar una de cada una.

Al dar click en el botón “Siguiente”, se mostrará la pantalla de resultados.

En la primera pestaña se encuentran las estimaciones de la población completa y una tabla resumen que contiene los cálculos del acumulado de las probabilidades indicadas en el archivo inicial.

En la segunda pestaña se encuentra una tabla con la muestra y en la tercera pestaña se despliega una tabla con las estimaciones calculadas de la muestra.

En la parte inferior de la pantalla se encuentran los botones “Volver” para retroceder en los pasos anteriores y el botón “Guardar” para registrar toda la información desplegada en un archivo de texto para usos posteriores.

Las figuras 42, 43 y 44 ejemplifican lo anterior.



Figura 42. Resultados Método Aleatorio Simple Ponderado. Pestaña Población



Figura 43. Resultados Método Aleatorio Simple Ponderado. Pestaña Muestra.



Figura 44. Resultados Método Aleatorio Simple Ponderado. Pestaña Cálculos.

# Instalación y Ejecución de la Aplicación

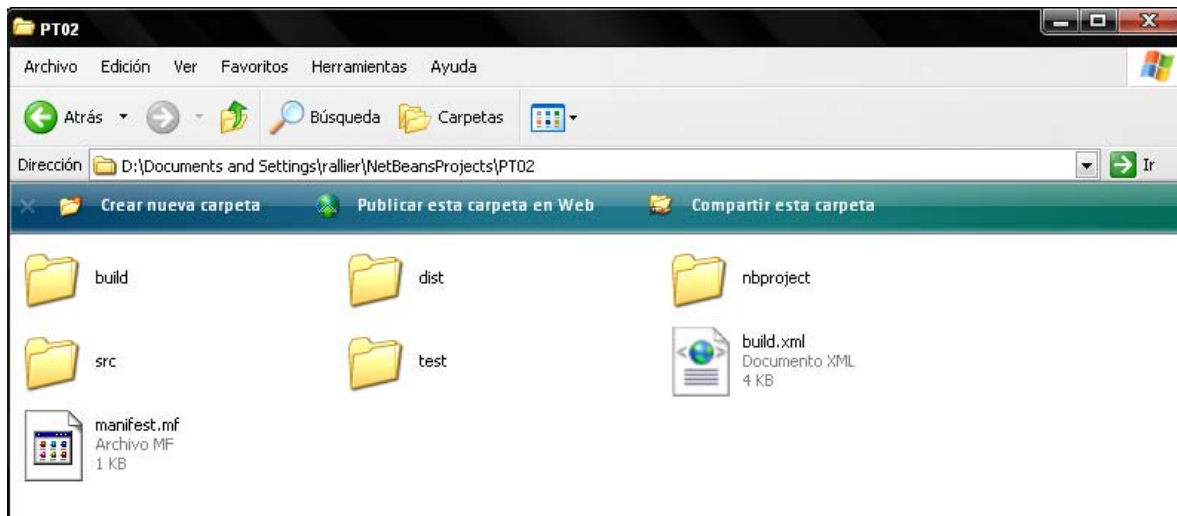
## Requerimientos Previos

Para la correcta ejecución de la aplicación la computadora donde se va a ejecutar el programa debe tener instalado el JRE (Java Runtime Environment) de preferencia la versión 6 en adelante para que pueda interpretar todas las librerías que utiliza esta aplicación, especialmente aquellas librerías que utiliza la interfaz gráfica. Se puede obtener de la siguiente liga:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

## Estructura de Carpetas

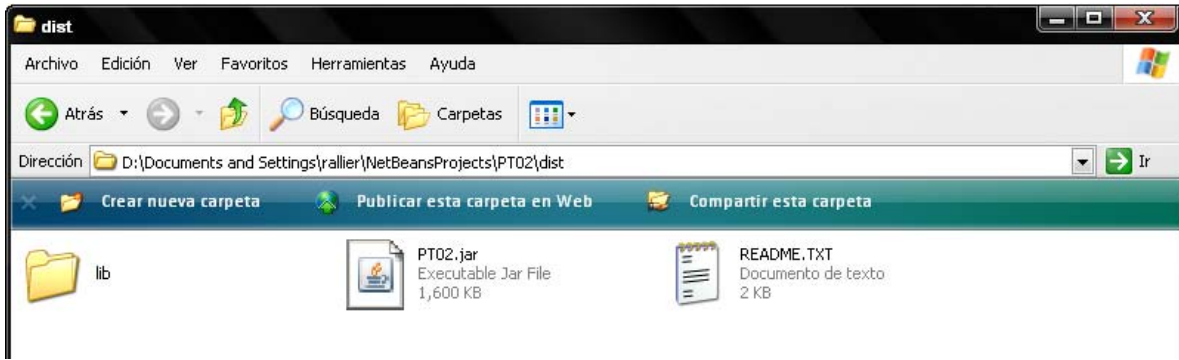
Se tiene una carpeta llamada PT02 que contiene la aplicación completa.



Dentro de ella, se encuentran 5 carpetas y dos archivos auxiliares para la ejecución de la aplicación.

La carpeta PT02 puede ser colocada en cualquier parte de la computadora, se recomienda para Windows colocarla en la carpeta de Mis Documentos.

Las carpetas *build*, *test* y *nbproject* contienen información de compilación de clases para la máquina virtual de Java. La carpeta *src* contiene el código fuente de la aplicación, y la carpeta *dist* es la que contiene el archivo ejecutable.



Dentro de esta carpeta se encuentra el archivo ejecutable PT02.jar. Al hacer doble click en él se ejecutará la aplicación.