

Manual de instalación:

Prerrequisitos:

Lyx

Para instalar Lyx:

```
sudo add-apt-repository ppa:lyx-devel/daily
sudo apt-get update
sudo apt-get install lyx2.0
```

Maxima

Para instalar Maxima:

```
sudo apt-get install maxima
```

Para instalar Java JRE:

```
sudo apt-get install openjdk-6-jre
```

Instalar MySQL y entorno grafico para MySQL (MySQL Administrator):

```
sudo aptitude install mysql-server-5.0 mysql-client
sudo apt-get install mysql-admin
```

Nota:El usuario debe ser: root y no debe tener password.

Instalación del software:

- 1.- Copiar la del Cd, las carpetas: "SGdEdIP" y "dist" en el escritorio.
- 2.- Importar la Base de Datos "SGdEdIP.sql" mediante MySQL Administrator.
- 3.- Para ejecutar el software, localizar el archivo "SGdEdIP.rar" dentro de: /Escritorio/dist
dar permisos de ejecución al archivo y ejecutar con OpenJDK Java 6 Runtime.

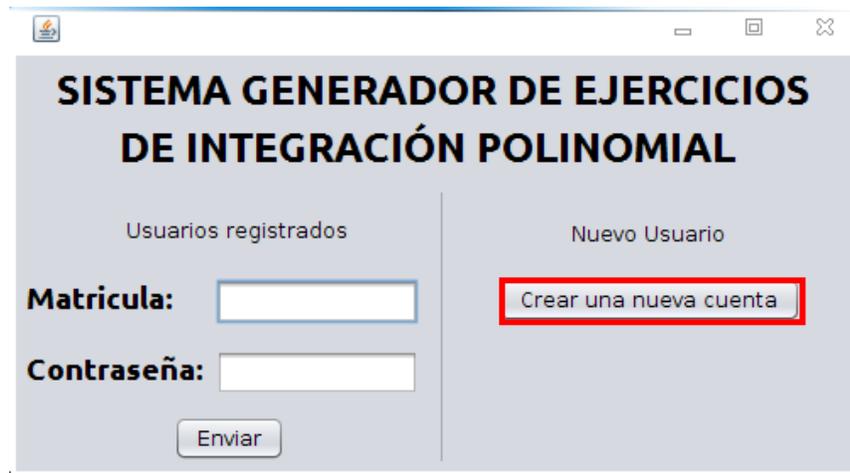
Nota: Consultar manual de usuario.

SISTEMA GENERADOR DE EJERCICIOS DE INTEGRACIÓN POLINOMIAL

Manual de usuario

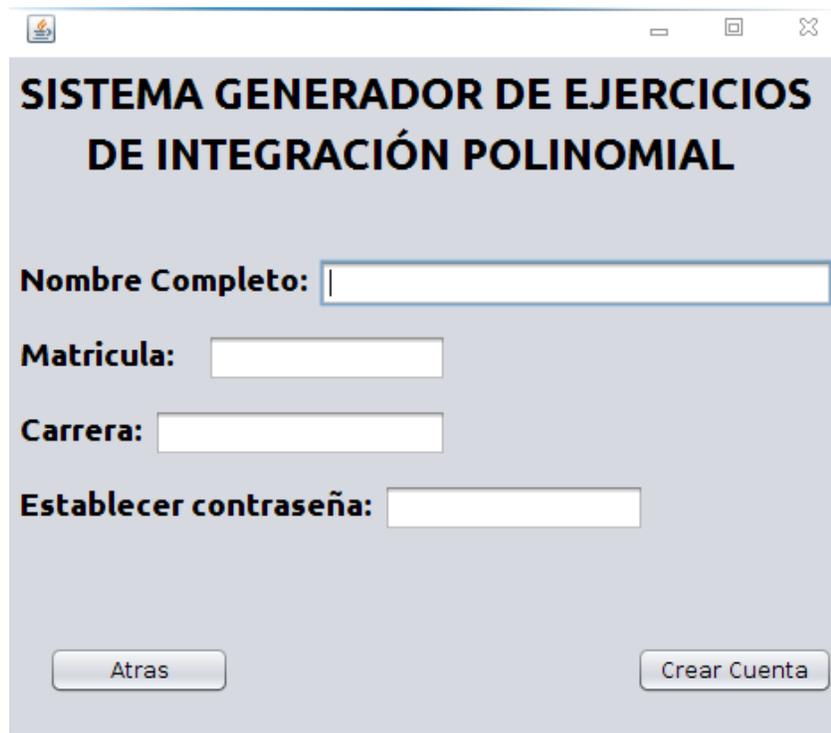
Perfil de Alumno:

Para utilizar el sistema, es necesario crear una cuenta, si es la primera vez que lo utilizamos, para lo cual deberemos seleccionar la opción de “Crear nueva cuenta en el menú principal”.



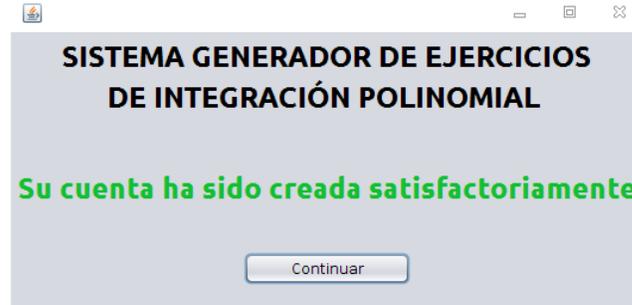
The screenshot shows the main interface of the system. At the top, the title "SISTEMA GENERADOR DE EJERCICIOS DE INTEGRACIÓN POLINOMIAL" is displayed. Below the title, there are two columns. The left column is titled "Usuarios registrados" and contains two input fields: "Matricula:" and "Contraseña:". Below these fields is a button labeled "Enviar". The right column is titled "Nuevo Usuario" and contains a button labeled "Crear una nueva cuenta", which is highlighted with a red rectangular border.

Para crear nuestra cuenta, es necesario llenar el siguiente formulario con nuestros datos personales:



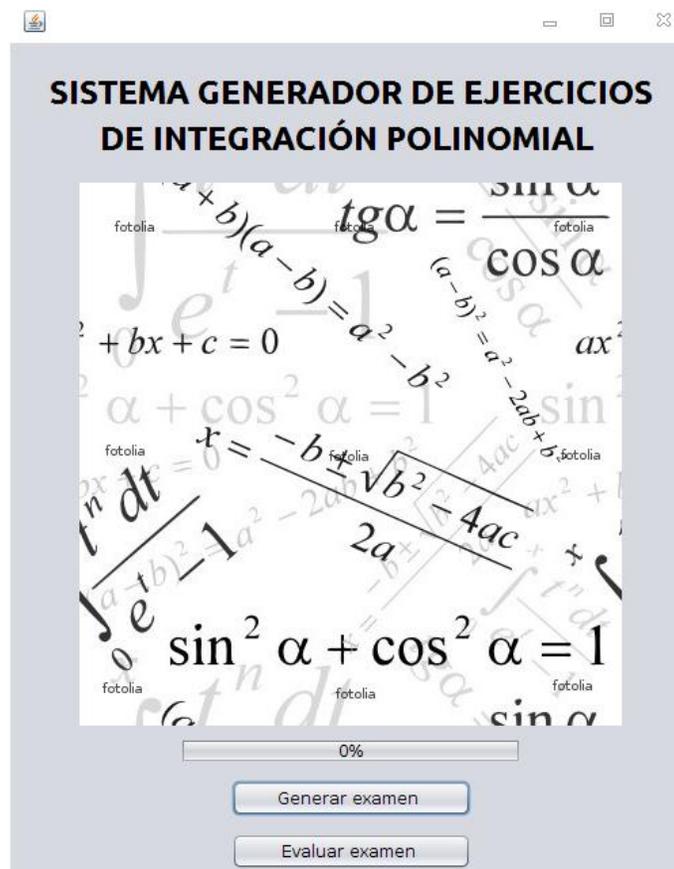
The screenshot shows the registration form. At the top, the title "SISTEMA GENERADOR DE EJERCICIOS DE INTEGRACIÓN POLINOMIAL" is displayed. Below the title, there are four input fields: "Nombre Completo:", "Matricula:", "Carrera:", and "Establecer contraseña:". At the bottom of the form, there are two buttons: "Atras" on the left and "Crear Cuenta" on the right.

Una vez finalizado el proceso de creación de cuenta, se recibirá una confirmación de dicho proceso.



Posteriormente seremos re direccionados al menú principal, mediante el cual, con nuestra cuenta, se podrá ingresar al sistema.

Después de autenticarse en el sistema, al usuario le será presentado el siguiente menú:



Cuando un usuario, seleccione la opción “Generar un examen” se generara un examen aleatorio, el cual será presentado en procesador de textos Lyx, como se muestra a continuación:

UNIVERSIDAD AUTONOMA METROPOLITANA
CALCULO INTEGRAL
EJERCICIOS DE POLINOMIOS

Escribe tu reultado despues del simbolo **R=** , para ingresar al modo matemático, presiona:
Ctrl + m
Matricula: 207200857

1.- Evaluar el siguiente polinomio con $x= 7$

$$3x^4 + 3x + 2$$

R=

2.- Evaluar la siguiente integral definida

$$\int_{-2}^1 2x^5 + 4x + 3dx$$

Mediante este editor, el usuario podrá resolver el examen, para su posterior evaluación.

Reglas para resolver el examen.

Para que la revisión del examen sea correcta, se deben tener en cuenta las siguientes consideraciones:

Las respuestas a los reactivos, pueden ser de dos tipos: Números enteros, y fracciones tanto positivas como negativas.

Para que una respuesta sea considerada por el sistema, deberá ser colocada exactamente entre el símbolo “R=”, y el símbolo “ \leftarrow ”.

Para escribir una respuesta entera, bastara con colocarse en la posición mencionada anteriormente, y escribir el resultado.

En caso, de que la respuesta sea una fracción, será necesario ingresar en modo matemático, para esto, bastara con presionar la combinación de teclas Ctrl + M, lo cual nos desplegara una barra, la cual nos permite ingresar fracciones seleccionando el icono que se menciona a continuación.



Una vez que el usuario a finalizado el examen bastara con guardar el documento, presionando Ctrl + G, y será re direccionado al menú de examen.

Para verificar sus resultados, el alumno, deberá seleccionar la opción de “evaluar examen” y esperar a que la barra llegue hasta el 100% (2 segundos aproximadamente), y que el botón “evaluar examen” cambie a “CALIFICACION”.



Una vez que esto ha sucedido, el alumno, deberá hacer clic sobre el botón “CALIFICACION”, el cual le abrirá nuevamente el editor Lyx con el examen ya calificando, indicando si su respuesta fue correcta o incorrecta.

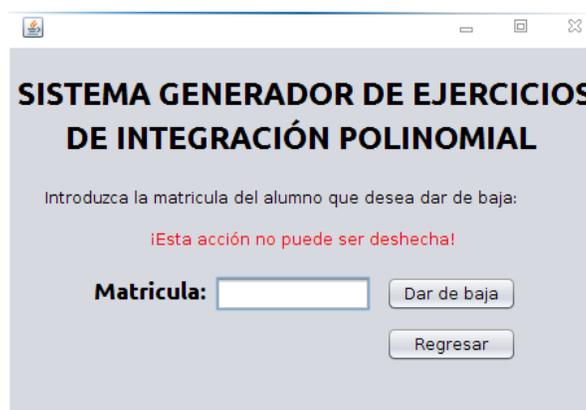
Perfil Administrador:

En este perfil, será posible dar de baja alumnos del sistema, los datos del usuario administrador, son los siguientes:

Matricula: 123456789

Contraseña: admin

Bastara con ingresar la matricula del alumno que se desea dar de baja, para que este sea eliminado del sistema.



Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

SISTEMA GENERADOR DE EJERCICIOS DE INTEGRACIÓN POLINOMIAL

Ludwig Villarreal Guzmán 20700857

Trimestre 12-O

Dr. Carlos Barrón Romero Profesor-Investigador Titular C

Índice

Agradecimientos	3
Introducción.....	5
CAPITULO 1 – Antecedentes.....	6
1.1 Justificación:	6
1.2 Referencias y antecedentes	6
1.2.1 Referencias internas:	7
1.2.2 Referencias Externas:.....	7
1.3 Objetivos	8
1.3.1 Objetivo general:	8
1.3.2 Objetivos específicos:	8
CAPITULO 2 - Descripción técnica:.....	9
2.1 Descripción de Funcionamiento	9
2.2 Descripción de módulos:.....	11
2.2.1 Interface principal.....	11
2.2.2 Generador aleatorio de expresiones algebraicas.....	11
2.2.3 Módulo editor	11
2.2.4 Módulo de verificación de resultados	11
2.3 Especificación Técnica:	12
CAPITULO 3 – Desarrollo	13
3.1 Interfaz Grafica	13
3.2 Modulo generador de expresiones matemáticas	17
3.3 Modulo de integración con un editor LaTeX	19
3.3 Modulo de integración con un sistema de álgebra computacional	20
3.4 Modulo verificador de resultados	22
CAPITULO 4 – Resultados y Conclusiones.....	31
Posibles continuaciones:.....	32
Bibliografía:.....	32

Agradecimientos

Antes de comenzar con las menciones independientes, quiero agradecer a todas aquellas personas que se vieron involucradas de manera directa e indirecta en mi formación universitaria desde la persona que me vendió algún material en la papelería, hasta los profesores que más me influenciaron durante mi estadía en la universidad, ya que cada uno de ellos en cierta forma aportaron algo para mi formación como estudiante universitario de ingeniería en computación.

En primer lugar quiero agradecer a los miembros de mi familia, la cual, para mí es considerada bajo el esquema de lo que se conoce como una familia nuclear; quiero agradecer a mi padre, José Villarreal Bastamente, que no solo fue uno de los que me ofrecieron la posibilidad de tener estudios universitarios, sino que también estuvo para mí como un ejemplo quien me motivó a terminar mi carrera y obtener la mejor herencia que me pudieron dejar; mis estudios.

A mi madre Julieta Guzmán Rosales que siempre estaba para mí en cualquier momento, desde en la mañana para enviarme el desayuno, hasta en la noche en la que me ofrecía con gusto un café, para poder aguantar la larga noche, que ella sabía, me esperaba, pero todo eso queda de lado ya que uno de los mayores aportes de su parte, fue la motivación y una sonrisa que me decía con una voz muda “Tranquilo hijo tu puedes”.

Mi hermano Paris Villarreal Guzmán, que siempre estuvo ahí para mí en todo momento, y literalmente, en todo momento, ya que hemos sido compañeros de toda la vida y no podía faltarme en la etapa más importante de mi vida, tolerando el no poder tener la libertad, de no poder ser libre, en el único lugar en el que él debería de haber podido hacer lo que él quisiese, sin embargo, cuando alguien le pedía guardar silencio porque “yo, estaba estudiando” él lo hacía sin chistar, y aunque parece algo muy simple, todos sabemos que esas noches de estudio son imposibles de desaprovechar.

A mi novia, compañera, amiga y consejera; Fabiola Piedra Ruiz que todo el tiempo fue lo que me inspiraba a dar lo mejor de mí, ella que estuvo a mi lado; incluso desde antes de siquiera intentar ser un ingeniero, y que durante todo el camino soportó las cosas buenas y las malas que pasan durante los años universitarios, y que, a pesar de que muchas personas dijeran, “yo no hubiera aguantado tanto”, ella siempre estuvo, no solo ahí, siempre es tuvo conmigo y para mí.

Y finalmente, quiero hacer un agradecimiento a todos los profesores con lo que tuve el honor de encontrarme dentro de las aulas y laboratorios de la UAM; Todos ustedes tuvieron una gran influencia en mi formación, y se los agradezco, pero estoy obligado a hacer algunas menciones a aquellos profesores con los que tuve la oportunidad de tener una relación un poco más allá de la relación profesor alumno que se forma en un salón de clases.

Profesor Hugo Pablo Leyva, ya que sus consejos me ayudaron mucho en los inicios de mi vida laboral; Profesor Felipe Monroy Pérez, agradezco las pláticas y reflexiones que me ofrecía, las cuales me presentaron el panorama académico actual en México, lo cual me sirvió mucho para tomar algunas decisiones sobre mi futuro profesional.

Profesor Risto Fermín Rangel Kuoppa, agradezco todas esas charlas, y vaya que fueron muchas; sobre problemas existenciales que alguna vez atravesaron por mi mente, ya que no sabía en que área trabajar si en la electrónica, o en la computación; pero ahora puedo decirle que tome la mejor decisión; nuevamente, agradezco los consejos recibidos.

Profesor Francisco Javier Zaragoza Martínez agradezco mucho todo el tiempo que me dedico como alumno para la resolución de dudas desde las dudas que surgían en clase hasta las dudas de procedimientos o tramites propios de la universidad; y le agradezco que sea uno de esos profesores que nos inspiran a los alumnos a ir mas allá de la UAM y demostrar de que estamos hechos.

Finalmente quiero hacer un agradecimiento muy especial a mi asesor; el profesor Carlos Barrón Romero que ha tenido la paciencia y la tolerancia de trabajar con una persona que por motivos laborales fue necesario buscar una forma diferente de trabajo y poder adaptar la manera de trabajar juntos; pero sobre todo, quiero agradecer todos sus consejos, y anécdotas que siempre me llevaron a grandes reflexiones; además, quiero agradecerle que sea un profesor exigente, que nos motiva a hacer las cosas de la mejor manera posible ya que en este mundo las exigencias son las que llevan al éxito.

“Agradezco a todos los que creyeron en mí, pero sobre todo a los que dudaron, porque en lo personal, encuentro en los retos la mayor fuente de motivación de para mi trabajo”.

Una vez más... ¡Gracias!

Ludwig Villarreal Guzmán

Introducción

Durante el periodo de formación de ingenieros, se van adquiriendo una gran cantidad de conocimientos matemáticos, para lo cual, es primordial contar con alguna herramienta que facilite la realización de cálculos matemáticos, ya sea software para álgebra computacional, o software de graficación.

A su vez, una parte muy importante durante la formación de ingenieros, es brindarle herramientas (Software) que le permita, no solo la resolución de problemas sino también el apoyo necesario para la reafirmación de los conocimientos obtenidos, siendo este último el tema central de este trabajo.

Mi proyecto, es un sistema que permita a los alumnos realizar autoevaluaciones de los conocimientos matemáticos obtenidos, en particular es solo para exámenes de evaluaciones de polinomios y sus integrales definidas. Además con la correcta utilización de la notación matemática, ya que esa es una de las principales carencias de la mayoría de las herramientas gratuitas que existen en el mercado.

En este trabajo se presentan las ideas, los problemas y las soluciones que surgieron durante la realización del mismo; además de utilizar fragmentos del código fuente para ejemplificar de una manera muy visual, la metodología que se empleó en cada uno de los diferentes problemas, y la manera en que fueron resueltos en cada una de las etapas del proyecto, los errores que impedían la finalización de una etapa para poder continuar con la siguiente; y así poder cumplir con los objetivos establecidos para este proyecto.

Finalmente se presentan las conclusiones, a las que se llegó después de finalizado el presente proyecto; y algunas de las posibles aplicaciones y extensiones de áreas temáticas que se podrían abarcar en el futuro con herramientas de software como las utilizadas aquí..

CAPITULO 1 – Antecedentes

1.1 Justificación:

Durante la formación de ingenieros o cualquier área que tenga una estrecha relación con las matemáticas, se adquieren conocimientos primordiales, que se van correlacionando entre sí para generar así un marco teórico con el cual, serán capaces de desarrollarse profesionalmente.

Los sistemas de práctica de ejercicios de matemáticas, en especial para integración de polinomios, no tienen una interface adecuada en notación matemática, generalmente ofrecen soluciones en opción múltiple, que no permiten la verificación de la correcta escritura matemática.

Esta propuesta ofrece práctica de la correcta escritura en notación matemática, a problemas pertenecientes a la evaluación de polinomios y de sus integrales definidas.

Si bien, la utilización de software que permite al estudiante saltar u obtener atajos a ciertos pasos de determinados procesos matemáticos; lo cual desvirtúa el conocimiento ya que convierte al estudiante en un experto en algún software específico para la resolución de determinados problemas como lo son: Matlab, Maple etc.

Lo cual lo lleva a ser un experto en herramientas como las antes mencionadas, perdiendo la noción y práctica de las expresiones matemáticas que conllevan.

Este proyecto, es una propuesta de un prototipo para paliar solucionar problemáticas con las que se encuentran los estudiantes de ingeniería; por ejemplo, el repetir exámenes o tareas ocasiona que los alumnos hagan trampa en lugar de aprender. Otra es tener a la mano, una herramienta capaz de entender el lenguaje matemático tal cual es enseñado en las aulas de clase, es decir todo el tiempo estar trabajando con notación matemática en lugar de tener que aprender un lenguaje ad-hoc para determinado software y así brindar apoyo tanto a profesores como a alumnos, mediante una herramienta que pueda, por el lado de los alumnos; apoyarlos en la práctica y reafirmación de los conocimientos adquiridos, mediante la realización de ejercicios, y por parte de los profesores, agilizar el proceso de retroalimentación del desempeño de sus alumnos con la generación y evaluación de exámenes y tareas distintas.

1.2.1 Referencias internas:

El primer proyecto relacionado, consta de un sistema para el desarrollo de reactivos y sus respectivas soluciones, para la elaboración de exámenes a partir de bases de datos, propuesto en el trimestre 10-P [1]. La diferencia principal radica, en el método de obtención de la soluciones.

Una segunda referencia, es un proyecto en el cual, se desarrollo una herramienta para crear exámenes con distintos formatos de pregunta respuesta, desarrollado durante el trimestre 09-P [2]. La diferencia con este proyecto, consiste en la especialización en el área de las matemáticas con lo cual serán omitidos diferentes tipos de respuesta centrándose únicamente en soluciones matemáticas.

1.2.2 Referencias Externas:

Scientific Workplace [3] es un paquete de software para el procesamiento de textos científicos, que está estructurado bajo la filosofía WYSIWYG¹ (what you see is what you get), lo cual significa que cualquier cambio realizado es visualizado de forma instantánea en la pantalla, Microsoft Word es probablemente el más difundido y conocido.

Este software, es capaz de proporcionar una plataforma en la cual se trabaja únicamente con notación matemática, evitando la necesidad de estudiar un lenguaje de programación para la creación de textos científicos proporcionando además la posibilidad de realizar cálculos matemáticos sobre el documento.

Moodle [4] es un software, capaz de ofrecer diversos tipos de exámenes en diferentes formatos de respuestas, desde respuestas con opción múltiple, hasta solución de problemas matemáticos, estos últimos teniendo una gran deficiencia, ya que no son capaces de reconocer expresiones matemáticas, limitándose a expresiones alfanuméricas bastante confusas para los alumnos; en específico no es capaz de presentar integrales definidas de la siguiente forma, ya que no puede integrar ecuaciones como la siguiente en lenguaje matemático, al no soportar lenguajes matemáticos al ser una plataforma web.

$$\int_1^a x^2 + x + 2 dx$$

Siendo uno de los objetos de este proyecto, el fomentar en el alumno la correcta utilización de la notación matemática para evitar expresiones” del tipo: `integrate(x^2+x+2,x ,1 ,3)`

¹Es el acrónimo de What You See Is What You Get (en inglés, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

1.3 Objetivos

1.3.1 Objetivo general:

Diseñar e implementar un sistema para la generación y evaluación de ejercicios de integración polinomial.

1.3.2 Objetivos específicos:

- Implementa la generación de expresiones matemáticas usando funciones aleatorias.
- Diseñar e implementar una interfaz gráfica para la administración de la aplicación.
- Diseñar e implementar un módulo verificador de resultados en notación matemática.
- Implementar la integración de la aplicación con un editor LaTeX².
- Implementar la integración de la aplicación con un sistema de álgebra computacional³ que permitirá a la aplicación la evaluación de los ejercicios.
- Implementar la comunicación entre todos los módulos anteriores.

²Es un sistema de composición de textos, orientado especialmente a la creación de libros, documentos científicos y técnicos que contengan fórmulas matemáticas.

³Un sistema algebraico computacional o sistema de álgebra computacional SAC (CAS, del inglés computer algebra system) es un programa de ordenador o calculadora avanzada que facilita el cálculo simbólico.

CAPITULO 2 - Descripción técnica:

2.1 Descripción de Funcionamiento

Las funciones que el sistema puede realizar se presentan en el diagrama de casos de uso mostrado en la Figura 1.

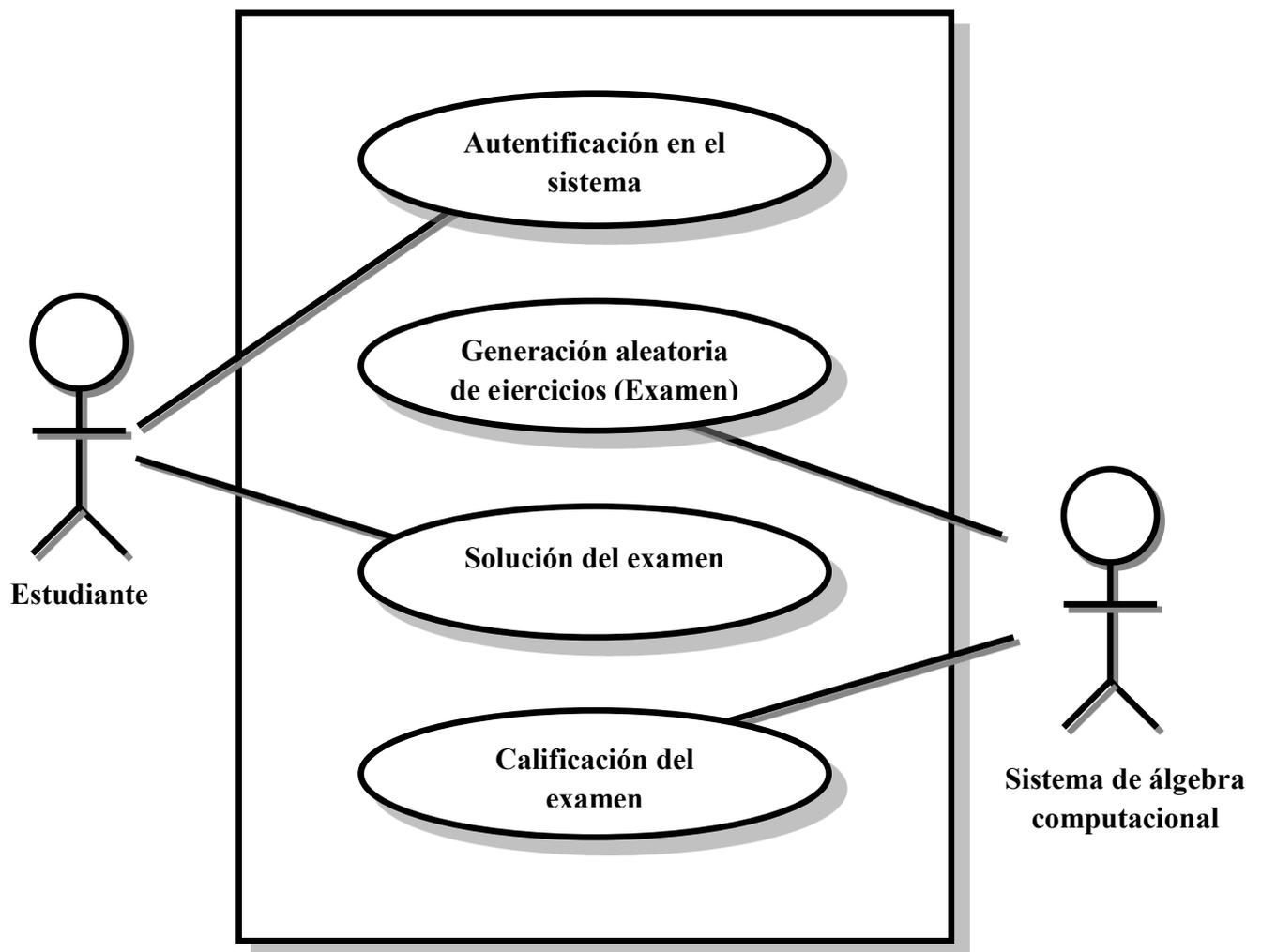


Figura 1. Funcionamiento del sistema.

La interacción de los módulos se realizara como se muestra en la Figura 2.

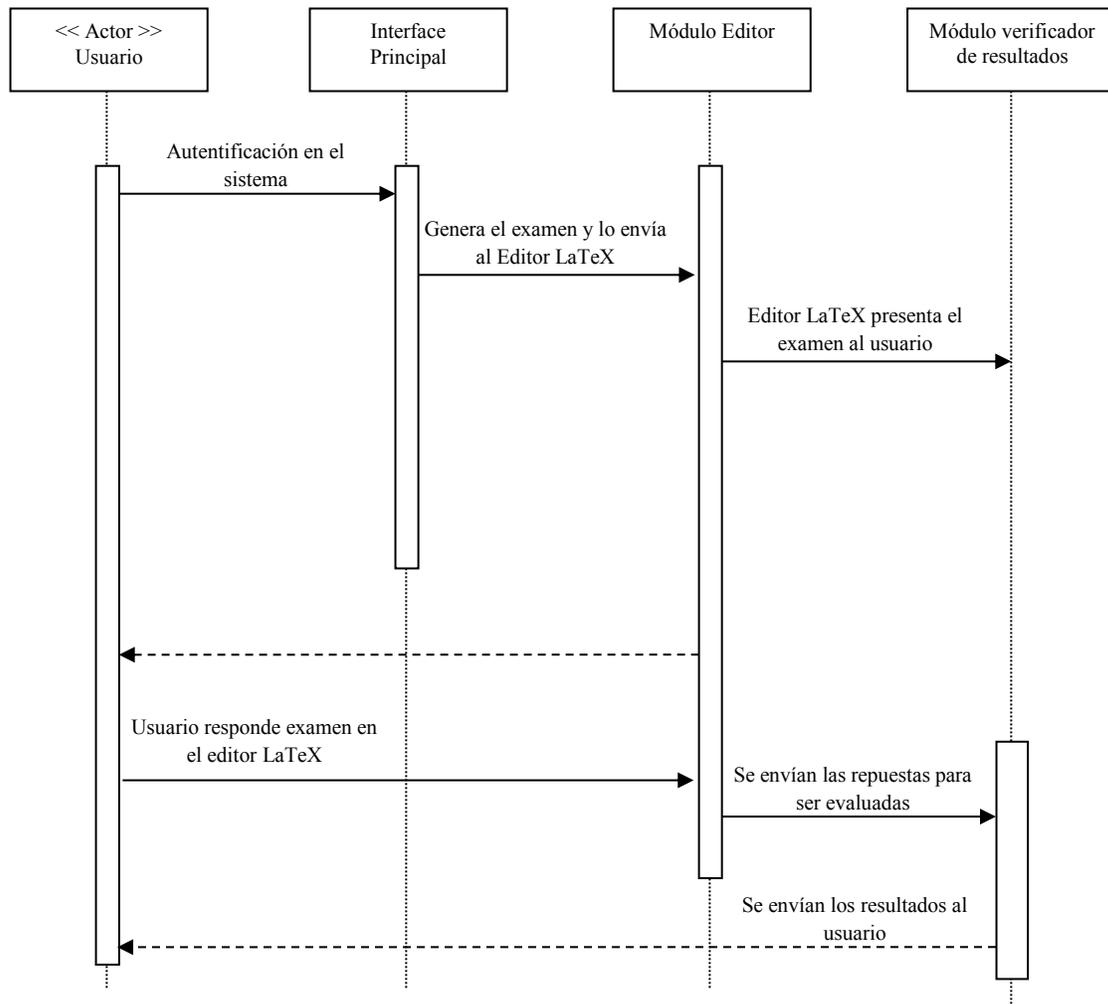


Figura 2.

2.2 Descripción de módulos:

2.2.1 Interface principal

Esta encargada, de presentar al usuario las diferentes pantallas que le permitirán interactuar con la aplicación, como los son la autenticación en el sistema, generación del examen y la calificación del mismo, así como para el administrador del sistema, permitirá la administración de los usuarios y sus roles dentro del sistema.

2.2.2 Generador aleatorio de expresiones algebraicas

Esta encargado de la generación de expresiones matemáticas, tanto la generación de los polinomios, como la generación de las integrales definidas que serán parte del examen, cabe destacar que estas expresiones serán generadas para que sean expresiones en notación matemática, pero también se crearan expresiones entendibles para un sistema de algebra computacional.

2.2.3 Módulo editor

Este módulo, es el que le presenta al alumno la serie de ejercicios a resolver (Examen) empleando Lyx, un editor de textos que utiliza la tecnología LaTeX para la captura de sus respuestas única y estrictamente en notación matemática, las cuales serán enviadas al módulo de verificación de resultados para su evaluación.

2.2.4 Módulo de verificación de resultados

Este módulo es el encargado de la revisión de los resultados obtenidos por el alumno, los cuales fueron capturados en el módulo editor, serán verificados mediante la utilización de Máxima, un software de álgebra computacional, los cuales se verificaran de la siguiente forma:

Integrales definidas y evaluación de polinomios:

Se enviaran los ejercicios que resolvió el alumno al software de álgebra computacional para ser resueltos, y finalmente se compararan los valores de la respuesta con los obtenidos por el alumno.

2.3 Especificación Técnica:

Debido a la complejidad del proyecto, este se delimita y solución única y exclusivamente de problemas de integrales definidas, y la evaluación de polinomios.

Para que este proyecto se diera por terminado, se tendría que presentar una aplicación, capaz de presentar al alumno, un examen el cual, podrá contestar en un editor LaTeX, para su posterior procesamiento y revisión, una vez que se verifica los resultados, se entrega una calificación; llegando hasta este punto se daría por completa la elaboración del proyecto.

CAPITULO 3 – Desarrollo

El Desarrollo del Sistema será planteado de acuerdo a la creación de los diferentes módulos que conforma el sistema, en su mayoría, los módulos se comunican entre sí mediante métodos de clase, pero también mediante la escritura y lectura de archivos, que son generados por el método anterior, y que genera un archivo para el método siguiente.

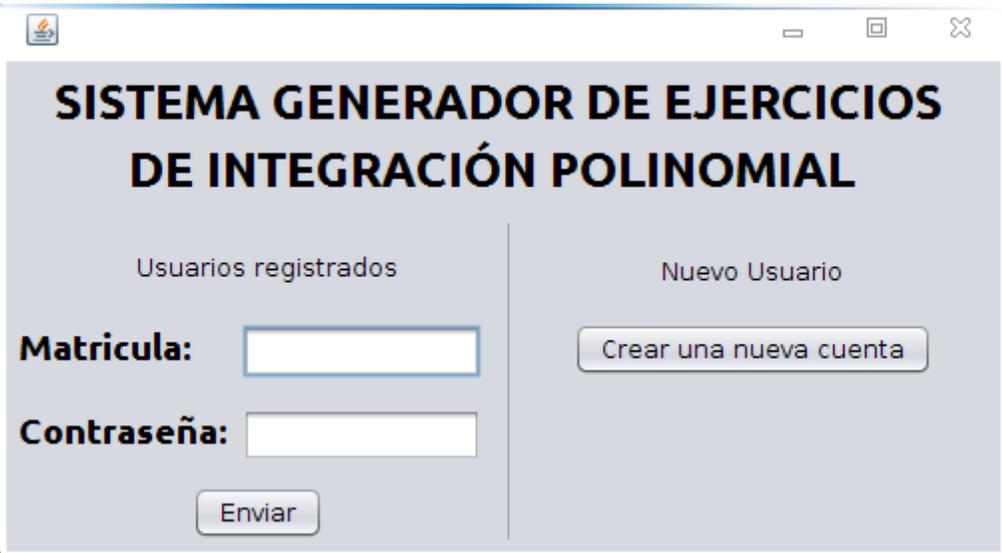
3.1 Interfaz Grafica

La interfaz gráfica se implementó mediante la utilización de Jpanel, que un objeto contenedor, que es llamado así porque sirve para contener otros objetos en Java, se seleccionó Jpanel, ya que proporciona una manera muy amigable para generar interfaces, las cuales pueden ser creadas de una forma visual simplemente creando una ventana base, y arrastrando y soltando los elementos extra, para dar formato simplemente se modificaron las propiedades de los objetos, ya sea desde el ID de desarrollo o modificando las propiedades de los objetos desde alguno de los métodos de nuestras clases, como lo veremos en el módulo de verificación.

A continuación se presentan las diferentes ventanas para los dos perfiles que soporta el sistema, Administrador y alumno, además se describirá la funcionalidad de cada una de las pantallas, que a su vez me permitirá usar para un mejor entendimiento del flujo del sistema.

Pantalla de Autenticación (Perfil de Alumno)

En esta ventana, se podrá ingresar si el usuario ya tiene una cuenta existente, de lo contrario, nos da la opción de crear una cuenta nueva.

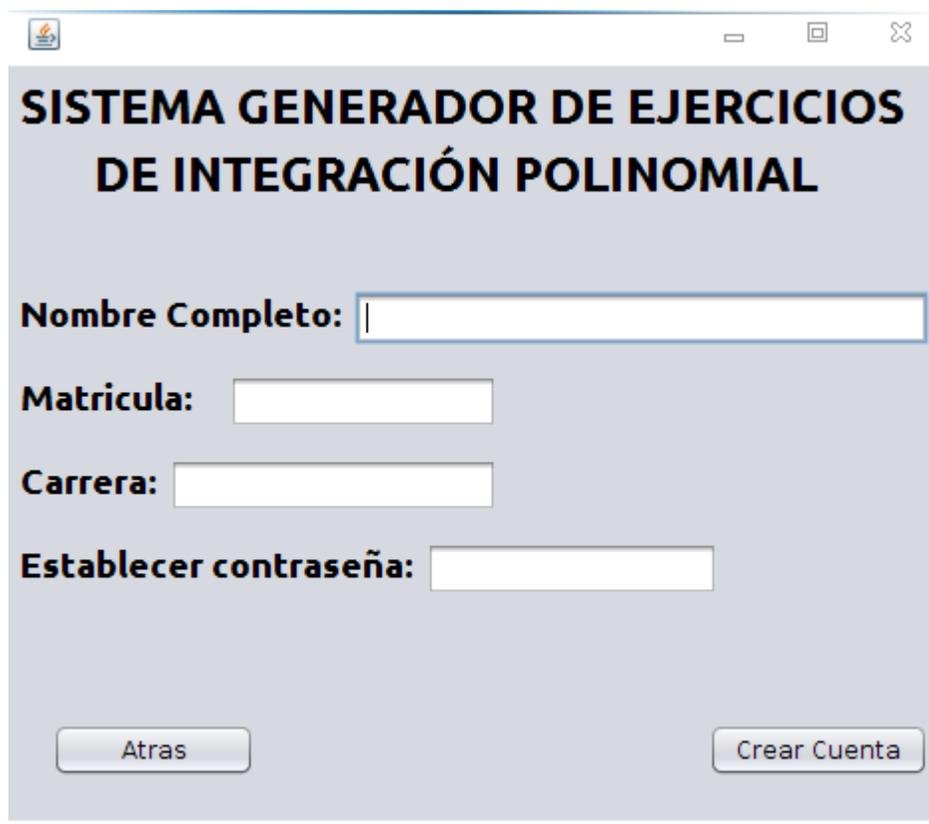


The screenshot shows a Java Swing window titled "SISTEMA GENERADOR DE EJERCICIOS DE INTEGRACIÓN POLINOMIAL". The window is divided into two main sections by a vertical line. The left section is titled "Usuarios registrados" and contains two text input fields: "Matricula:" and "Contraseña:". Below these fields is an "Enviar" button. The right section is titled "Nuevo Usuario" and contains a "Crear una nueva cuenta" button. The window has a standard Windows-style title bar with minimize, maximize, and close buttons.

Formulario Usuario Nuevo (Perfil de Alumno)

Este formulario nos permite dar de alta una cuenta en el sistema, solo es necesario ingresar nuestros datos.

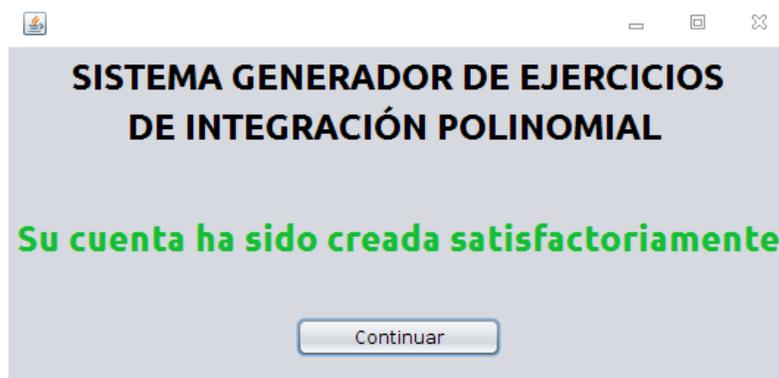
Nota: La base de datos, usa como llave primaria la matricula de los alumnos, esto implica que la cuenta no será creada debido a que ya existe.



The screenshot shows a web browser window with the title "SISTEMA GENERADOR DE EJERCICIOS DE INTEGRACIÓN POLINOMIAL". The page contains a registration form with the following fields and buttons:

- Nombre Completo:** A text input field.
- Matricula:** A text input field.
- Carrera:** A text input field.
- Establecer contraseña:** A text input field.
- Atras:** A button located at the bottom left.
- Crear Cuenta:** A button located at the bottom right.

En caso de que la creación de la cuenta sea exitosa, se mostrara la siguiente pantalla:



The screenshot shows a web browser window with the title "SISTEMA GENERADOR DE EJERCICIOS DE INTEGRACIÓN POLINOMIAL". The page displays a success message:

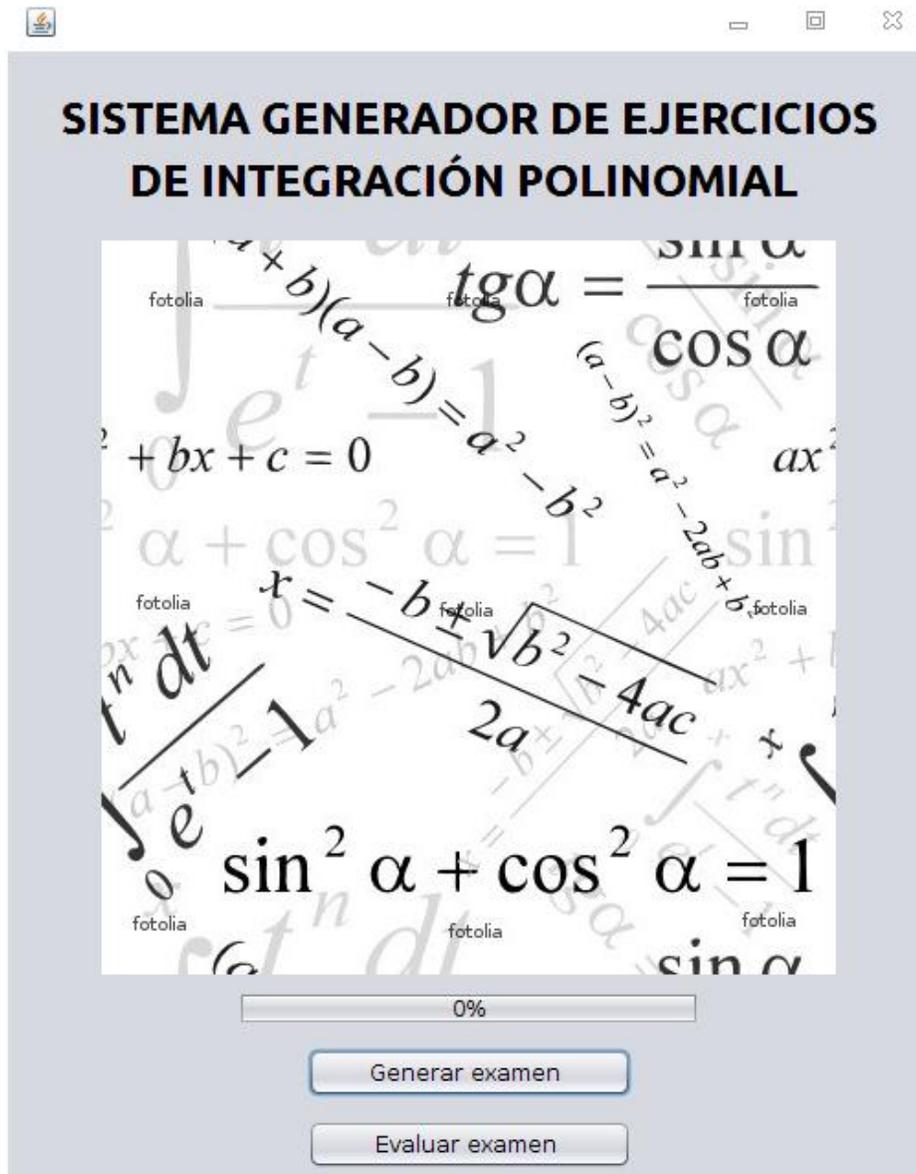
Su cuenta ha sido creada satisfactoriamente

Below the message is a button labeled "Continuar".

Nota: Decidí, integrar mensajes de confirmación porque proporcionan información importante de lo que hizo el usuario.

Finalmente cuando se ha creado la cuenta, podemos acreditararnos como usuario en la ventana de Autenticación, que se presenta en cuanto se hace clic en el botón continuar del mensaje de confirmación anterior.

Cuando nos hemos autenticado en el sistema, se presentara la siguiente ventana, en la que tendremos la opción de generar un examen, después de responderlo se deberá guardar el archivo, para su posterior procesamiento y calificación⁴.



⁴Para mayor información, favor de revisar el manual de usuario, documento anexo.

Una vez que hacemos clic en Generar examen, se abrirá el Editor de textos Lyx, con un examen único y diferente cada vez, como el que se muestra a continuación.

UNIVERSIDAD AUTONOMA METROPOLITANA
CALCULO INTEGRAL
EJERCICIOS DE POLINOMIOS

Escribe tu resultado despues del simbolo **R=** , para ingresar al modo matemático, presiona:
Ctrl + m
Matricula: 207200857

1.- Evaluar el siguiente polinomio con $x= 7$

$$3x^4 + 3x + 2$$

R=

2.- Evaluar la siguiente integral definida

$$\int_{-2}^1 2x^5 + 4x + 3dx$$

Se decidió que el examen debería de ser personalizado, por lo que incluye la matricula del usuario que está realizando el examen, el examen que se genera contiene 10 reactivos todos diferentes.

Una vez que el examen es resuelto, debe guardarse y posteriormente hacer clic en el botón evaluar examen de la ventana donde dimos clic para generar el examen.

UNIVERSIDAD AUTÓNOMA METROPOLITANA
CALCULO INTEGRAL
EJERCICIOS DE POLINOMIOS

Escribe tu resultado despues del simbolo **R=** , para ingresar al modo matemático, presiona:
Ctrl + m
Matricula: 207200857

1.- Evaluar el siguiente polinomio con $x= 7$

$$3x^4 + 3x + 2$$

R=900[Incorrecto]

Y nos regresa un examen calificado, indicándonos si nuestras respuestas son correctas o incorrectas.

Se ha presentado la interfaz de usuario, y se ha contextualizado el flujo del sistema.

Nota: La información expresada en este documento, no es un manual de usuario, por lo que podrían faltar detalles de su utilización, por favor revisar manual de usuario para aclaraciones del uso.

3.2 Modulo generador de expresiones matemáticas

Este modulo tiene la finalidad de generar los ejercicios que serán resueltos por los alumnos por lo cual se implementaron varias funciones aleatorias, para generar el tipo de preguntas que contendrá el examen, consecuentemente, también se crearon funciones para crear los polinomios y las integrales, variando los índices de integración, coeficientes y exponentes que conforman las diferentes expresiones matemáticas que pueden ser creadas para generar exámenes completamente diferentes.

A continuación se presentan algunos fragmentos de código, los cuales servirán para ejemplificar la manera en que trabajan estas funciones para crear las expresiones matemáticas⁵.

Estas funciones aleatorias, escriben directamente en un archivo que es creado a través de Java utilizando I/O estándar mediante las clases File, FileWriter, PrintWriter, FileReader además de sus diferentes métodos.

El código que genera las expresiones matemáticas es el siguiente:

```
// Generamos aleatoriamente de que tipo de reactivo sera el que se insertara en el examen
// Tipo 1 Evaluar la integral definida
// Tipo 2 Evaluar un polinomio en un X determinado

int tipo = (int) (Math.random()*2+1);

// Aleatoriamente, generamos los indices superior e inferior para las integrales definidas

int expo1 = -2 + (int) (Math.random()*5);
int expo2 = -2 + (int) (Math.random()*5);

//Aseguramos de el limite superior sea siempre mayor al limite inferior
if(expo1>expo2){
    int temp;
    temp = expo2;
    expo2 = expo1;
    expo1 = temp;
}
if(expo1 == expo2)
    expo2 ++;
```

⁵Si es necesario más detalle sobre el código, este puede ser analizado por completo, ya que también será entregado en el CD.

Algunos otros de los componentes de las expresiones matemáticas, como los son los exponentes, los índices también son generados aleatoriamente de la siguiente forma.

```
// Generamos indices y exponentes aleatoriamente
int na = (int) (Math.random()*4+1)+1;
int ind1 = (int) (Math.random()*4+1)+1;
int ind2 = (int) (Math.random()*4+1)+1;
int ind3 = (int) (Math.random()*4+1)+1;
int ind4 = (int) (Math.random()*4+1)+1;
int ind5 = (int) (Math.random()*4+1)+1;
int ind6 = (int) (Math.random()*4+1)+1;
int na2 = (int) (Math.random()*4+1)+1;
int exis = (int) (Math.random()*9+1)+1;
```

Y mediante un switch, seleccionamos aleatoriamente, cual es el tipo de ejercicio que insertaremos dentro del examen, ya sea evaluar un polinomio para cierto valor de x, o resolver una integral definida.

```
// Construimos las expresiones matematicas aleatoriamente
// A su vez, las escribimos aleatoriamente en el examen
```

```
switch(tipo){
    case 1:
        pw.println("\\begin layout Standard");
        pw.println("\\align left");
        pw.println((i+1)+".- Evaluar la siguiente integral definida");
        pw.println("\\begin inset Newline newline");
        pw.println("\\end inset");
        pw.println("\\begin layout Standard");
        pw.println("\\align left");
        pw.println("\\begin inset Formula $\\int_{"+expo1+"}^{"+expo2+"}"+ind1+" x^{"+na+"}"+ind2+"x"+na2+" dx$");
        pw.println("\\end inset");
        String cad1 = "\\begin inset Formula $\\int_{"+expo1+"}^{"+expo2+"}"+ind1+" x^{"+na+"}"+ind2+"x"+na2+" dx$";
        pw2.println("integrate("+ind1+"*x^{"+na+"}"+ind2+"*x"+na2+", x, "+expo1+", "+expo2+"");");
        break;
    case 2:
        pw.println("\\begin layout Standard");
        pw.println("\\align left");
        pw.println((i+1)+".- Evaluar el siguiente polinomio con x= "+exis);
        pw.println("\\begin inset Newline newline");
        pw.println("\\end inset");
        pw.println("\\begin layout Standard");
        pw.println("\\align left");
        pw.println("\\begin inset Formula $" +ind5+"x^{"+na+"}"+ind6+"x"+na2+"$");
        pw.println("\\end inset");
        String cad3 = "\\begin inset Formula $" +ind5+"x^{"+na+"}"+ind6+"x"+na2+"$";
        pw2.println(ind5+"*x^{"+na+"}"+ind6+"*x"+na2+", x: "+exis+"");
        break;
}
```

3.3 Modulo de integración con un editor LaTeX

Para crear un archivo de Lyx, fue necesario utilizar los comandos propios del lenguaje LaTeX.

```
// Lineas que se escribieran en el Archivo .lyx, con un formato
// que maneja el lenguaje LaTeX

pw.println("\\begin_layout Standard");
pw.println("\\align center");
pw.println("Escribe tu resultado despues del simbolo");
pw.println("\\series bold");
pw.println(" R= ");
pw.println("\\series default");
pw.println(", para ingresar al modo matemático, presiona:");
pw.println("\\end_layout");
```

Las cuales generan la salida:

Escribe tu resultado despues del simbolo **R=** , para ingresar al modo matemático, presiona:

De la misma manera, para escribir una expresión matemática, escribimos en el archivo lo siguiente:

```
\begin_layout Standard
\align left
\begin_inset Formula $\int_{-2}^1 2x^5 + 4x + 3dx$
\end_inset
```

Lo cual nos produce como resultado:

$$\int_{-2}^1 2x^5 + 4x + 3dx$$

3.3 Modulo de integración con un sistema de álgebra computacional

Una vez que el archivo .lyx fue creado, el usuario puede escribir sus respuestas, posteriormente deberá guardarlo; a continuación ese archivo es procesado para extraer sus respuestas.

Al mismo tiempo que se generan las funciones matemáticas entendibles por Lyx, se generan las mismas funciones matemáticas en un archivo diferente, que contiene comandos propios de Lyx.

```
// Lineas que generan los ejercicios para entendibles para Maxima
// Linea que genera polinomios
pw2.println("integrate("+ind1+"*x^"+na+"+"+ind2+"*x^"+na2+",x,"+expol+"+"+expo2+");");

// Linea que genera integrales de polinomio
pw2.println(ind5+"*x^"+na+"+"+ind6+"*x^"+na2+",x:"+exis+");");
```

Una vez que termina, nos entrega un archivo llamado xxxxxxxres.txt, donde xxxxxxx es la matrícula del alumno que está resolviendo el examen; el siguiente, es un ejemplo del archivo generado en esta etapa:

```
2*x^5+2*x+3,x:5;
2*x^5+2*x+4,x:4;
5*x^4+3*x+2,x:5;
4*x^3+3*x+2,x:8;
integrate(2*x^2+3*x+2,x,1,2);
4*x^2+2*x+5,x:7;
2*x^3+2*x+5,x:2;
integrate(3*x^2+4*x+3,x,-1,1);
integrate(4*x^5+4*x+3,x,0,1);
```

El cual será procesado por Máxima mediante un pipe que será llamado desde el siguiente script:

```
#!/bin/sh
exec maxima</home/ludwig/Escritorio/SGdEdIP/207200857/207200857res.txt>
/home/ludwig/Escritorio/SGdEdIP/207200857/respuestas.txt
```

El cual toma el archivo creado en el paso anterior xxxxxxxres.txt y genera uno llamado respuestas.txt, el cual contendrá las respuestas correctas a los ejercicios que serán utilizadas posteriormente por el modulo verificador, el siguiente es un ejemplo de respuestas.txt:

Maxima 5.22.1 <http://maxima.sourceforge.net>
using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (a.k.a. GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.

The function `bug_report()` provides bug reporting information.

```
(%i1) (%o1) 6263
(%i2) (%o2) 2060
(%i3) (%o3) 3142
(%i4) (%o4) 2074
(%i5) (%o5) 67
(%i6) (%o6) 6
(%i7) (%o7) 215
(%i8) (%o8) 25
(%i9) (%o9) 8
(%i10) (%o10) 17
(%i11) (%o11) 3
```

Este archivo será utilizado en pasos siguientes por el modulo verificador como lo veremos más adelante.

3.4 Modulo verificador de resultados

Este modulo, se encarga de verificar si las respuestas generadas por el usuario son las correctas, para esto, utilizaremos el archivo respuestas.txt generado en el modulo anterior el cual deberá ser procesado de nuevo, ya que su actual formato no es el óptimo para poder trabajar con él; y un archivo llamado respusuario.txt, que es generado a partir, de leer el examen (archivo .lyx) y extraer las respuestas que el usuario escribió en el examen.

De la siguiente manera se genera respusuario.txt:

```
public void resultadosusuario () throws IOException{
>
>     FileWriter fichero = null;
>     PrintWriter pw = null;
>     try
>     {
>
>         fichero = new FileWriter(d1+"respusuario.txt");
>         pw = new PrintWriter(fichero);
>
>
>     String linea;
>     //Leemos linea por linea, los datos del archivo .lyx
>
>     BufferedReader bf = new BufferedReader(new FileReader(d2+".lyx"));
>
>
>     while ((linea = bf.readLine())!=null) {
>     //Identificamos si es una respuesta de tipo numerica y la escribimos en el nuevo archivo
>     if(linea.startsWith("R=1")||linea.startsWith("R=2")||linea.startsWith("R=3")||
>     linea.startsWith("R=4")||linea.startsWith("R=5")||linea.startsWith("R=6")||linea.startsWith("R=7")||
>     |linea.startsWith("R=8")||linea.startsWith("R=9")||linea.startsWith("R=-")){
>         pw.println(linea.substring(2));
>
>     }
>
>
>     // Identificamos si es una respuesta de tipo fracion positiva
>     // La procesamos y la escribimos en el nuevo archivo
>     else if(linea.contains("$\\frac")){
>         String sub = linea.substring(28);
>         int ind = sub.indexOf("}");
>         String cad1 = sub.substring(0,ind);
>         String cad2 = sub.substring(ind + 2);
>         cad2 = cad2.replace('}', ' ');
>         cad2 = cad2.replace('$', ' ');
>         cad2 = cad2.trim();
>
>         pw.println(cad1+"/"+cad2 );
>     }
>
> }
```

```
// Identificamos si es una respuesta de tipo fracion positiva
// La procesamos y la escribimos en el nuevo archivo
else if(linea.contains("-\\frac")){
```

```
String sub = linea.substring(29);
int ind = sub.indexOf("}");
String cad1 = sub.substring(0,ind);
String cad2 = sub.substring(ind + 2);
cad2 = cad2.replace('}', ' ');
cad2 = cad2.replace('$', ' ');
cad2 = cad2.trim();

pw.println("-"+cad1+"/"+cad2 );

}
}
}
```

Una vez que este proceso finaliza, se genera el archivo respusuario.txt, este es un ejemplo de ese archivo:

```
6263
2060
3142
2074
67/6
214
1
1
1
```



```

        if(linea.charAt(i)=='x'){
            if(linea.charAt(i-1)=='/' && linea.charAt(i-2)=='\\'){
                coef[l] = linea.charAt(i-3);
                l++;
            }
            else{
                coef[l] = ' ';
                l++;
            }
        }

        if(linea.charAt(i)=='+'){
            indice[m] = i;
            m++;
        }
    }

String cadena1 = linea.substring(0, indice[0]);
String cadena2 = linea.substring(indice[0]+1, indice[1] );

if(cadena1.contains("r")){
    div[0] = cadena1.charAt(cadena1.indexOf('r')+2);
}
else
    div[0] = ' ';

if(cadena2.contains("r")){
    div[1] = cadena2.charAt(cadena2.indexOf('r')+2);
}
else
    div[1] = ' ';

String cadena = (" +coef[0]+*x^"+exp[0]+)"/"+div[0]+(" +coef[1]+*x^"+exp[1]+)"/"+div[1]+ " +coef[2]+*x";
String p1 = cadena.substring(0, 9);
String p2 = cadena.substring(10, 19);
String p3 = cadena.substring(20);
String r1=" ", r2 = " ";

if((p1.charAt(1)!=' ')&(p1.charAt(5)!=' ')&(p1.charAt(8)!=' ')){
    r1 = (" +coef[0]+*x^"+exp[0]+)"/"+div[0]+";
}

else if((p1.charAt(1)!=' ')&(p1.charAt(5)!=' ')&(p1.charAt(8)==' ')){
    r1 = (" +coef[0]+*x^"+exp[0]+);
}

else if((p1.charAt(1)!=' ')&(p1.charAt(5)==' ')&(p1.charAt(8)!=' ')){
    r1 = (" +coef[0]+*x)/"+div[0]+";
}

else if((p1.charAt(1)!=' ')&(p1.charAt(5)==' ')&(p1.charAt(8)==' ')){
    r1 = (" +coef[0]+*x";
}

```

```

else if((p1.charAt(1)==' ')&(p1.charAt(5)!=' ')&(p1.charAt(8)!=' ')){
    r1 = "(x^"+exp[0]+" )/"+div[0]+" ";
}

else if((p1.charAt(1)==' ')&(p1.charAt(5)!=' ')&(p1.charAt(8)==' ')){
    r1 = "(x^"+exp[0]+" )";
}

else if((p1.charAt(1)==' ')&(p1.charAt(5)==' ')&(p1.charAt(8)!=' ')){
    r1 = "x/"+div[0]+" ";
}

if((p2.charAt(1)!=' ')&(p2.charAt(5)!=' ')&(p2.charAt(8)!=' ')){
    r2 = "( "+coef[1]+"*x^"+exp[1]+" )/"+div[1]+" ";
}

else if((p2.charAt(1)!=' ')&(p2.charAt(5)!=' ')&(p2.charAt(8)==' ')){
    r2 = "( "+coef[1]+"*x^"+exp[1]+" )";
}

else if((p2.charAt(1)!=' ')&(p2.charAt(5)==' ')&(p2.charAt(8)!=' ')){
    r2 = "( "+coef[1]+"*x )/"+div[1]+" ";
}

else if((p2.charAt(1)!=' ')&(p2.charAt(5)==' ')&(p2.charAt(8)==' ')){
    r2 = "( "+coef[1]+"*x )";
}

else if((p2.charAt(1)==' ')&(p2.charAt(5)!=' ')&(p2.charAt(8)!=' ')){
    r2 = "(x^"+exp[1]+" )/"+div[1]+" ";
}

else if((p2.charAt(1)==' ')&(p2.charAt(5)!=' ')&(p2.charAt(8)==' ')){
    r2 = "(x^"+exp[1]+" )";
}

else if((p2.charAt(1)==' ')&(p2.charAt(5)==' ')&(p2.charAt(8)!=' ')){
    r2 = "x/"+div[1]+" ";
}

else if((p2.charAt(1)==' ')&(p2.charAt(5)==' ')&(p2.charAt(8)==' ')){
    r2 = "x";
}

pw.println(r1+r2+p3);
}

else if(linea.startsWith("(") && l2.startsWith("(") && l3.startsWith(" ")){
    String resultado2 = linea.substring(5).trim()+"/"+l3.trim();
    pw.println(resultado2);
}

}

bf.reset();
}

catch(NullPointerException e) {

}

}

catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        // Nuevamente aprovechamos el finally para
        // asegurarnos que se cierra el fichero.
        if (null != fichero)
            fichero.close();
    } catch (Exception e2) {
        e2.printStackTrace();
    }
}

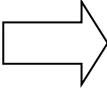
}

//System.out.println("CREAMOS respcorrect.txt");
}

```

Una vez terminado este proceso, obtendremos los siguiente:

```
Maxima 5.22.1 http://maxima.sourceforge.net
using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (a.k.a. GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) (%o1) 6263
(%i2) (%o2) 2060
(%i3) (%o3) 3142
(%i4) (%o4) 2074
(%i5) (%o5) 67
--
6
(%i6) (%o6) 215
(%i7) (%o7) 25
(%i8) (%o8) 8
(%i9) (%o9) 17
--
3
(%i10)
```



```
6263
2060
3142
2074
67/6
215
25
8
17/3
```

Respcorrect.txt

Respuestas.txt

Como podemos observar, respusuario.txt y respcorrect.txt tienen el mismo formato, por lo cual pueden ser comparadas para la verificación de resultados.

Para verificar los resultados, se leerán tanto respusuario.txt como respcorrect.txt y con ellos se formará una sentencia de máxima que nos dirá si son iguales o si son diferentes, a este archivo, lo llamaremos final.txt.

```
// Leemos ambos archivos línea por línea
String linea, linea2;
BufferedReader bf = new BufferedReader(new FileReader(dl+"respcorrect.txt"));
BufferedReader bf2 = new BufferedReader(new FileReader(dl+"respusuario.txt"));

while (((linea = bf.readLine())!=null) && ((linea2 = bf2.readLine())!=null)) {
>
// Sentencia de Maxima, que regresa si son iguales, o diferentes
pw.println("if((" + linea2 + ") - (" + linea + ")) = 0 then CORRECTO else INCORRECTO;");
}
}
```

A partir de este proceso, se genera final.txt cuyo contenido es el siguiente:

```
if((6263)-(6263))= 0 then CORRECTO else INCORRECTO;  
if((2060)-(2060))= 0 then CORRECTO else INCORRECTO;  
if((3142)-(3142))= 0 then CORRECTO else INCORRECTO;  
if((2074)-(2074))= 0 then CORRECTO else INCORRECTO;  
if((67/6)-(67/6))= 0 then CORRECTO else INCORRECTO;  
if((214)-(215))= 0 then CORRECTO else INCORRECTO;  
if((1)-(25))= 0 then CORRECTO else INCORRECTO;  
if((1)-(8))= 0 then CORRECTO else INCORRECTO;  
if((1)-(17/3))= 0 then CORRECTO else INCORRECTO;
```

Mediante un pipe que llama a máxima, el cual se encuentra dentro de el siguiente script, procesamos final.txt, para verificar si los resultados son iguales.

```
#!/bin/sh  
exec maxima</home/ludwig/Escritorio/SGdEdIP/207200857/final.txt>  
/home/ludwig/Escritorio/SGdEdIP/207200857/resultados.txt
```

El cual nos genera un archivo llamado resultados.txt, el cual contiene una precalificación al examen, este es un ejemplo de resultados.txt:

```
Maxima 5.22.1 http://maxima.sourceforge.net  
using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (a.k.a. GCL)  
Distributed under the GNU Public License. See the file COPYING.  
Dedicated to the memory of William Schelter.  
The function bug_report() provides bug reporting information.  
(%i1) (%o1) CORRECTO  
(%i2) (%o2) CORRECTO  
(%i3) (%o3) CORRECTO  
(%i4) (%o4) CORRECTO  
(%i5) (%o5) CORRECTO  
(%i6) (%o6) INCORRECTO  
(%i7) (%o7) INCORRECTO  
(%i8) (%o8) INCORRECTO  
(%i9) (%o9) INCORRECTO  
(%i10)
```

El sistema es capaz de presentarle al usuario, el mismo examen que el respondió, e indicándole si su respuesta fue correcta o no, para ello, se realizan los siguientes procesos:

Leemos el archivo resultado.txt línea por línea y guardamos en un arreglo de tipo String, ya sea la palabra correcto o incorrecto, una vez que tenemos ese arreglo, se lo pasamos al siguiente método:

```
public void test(String [] calificaciones) throws IOException{

    FileWriter fichero = null;
    PrintWriter pw = null;
    int i=0;

    try
    {
        fichero = new FileWriter(d1+"Calificacion.lyx");
        pw = new PrintWriter(fichero);

        String linea;
        //Leemos linea por linea, los datos del archivo .lyx original, con las respuestas del usuario
        BufferedReader bf = new BufferedReader(new FileReader(d2+".lyx"));

        while ((linea = bf.readLine())!=null) {
            // Verificamos si esa linea es una respuesta numerica, si lo es, verificamos en el arreglo cual fue su resultado
            if (linea.startsWith("R=1")||linea.startsWith("R=1")||linea.startsWith("R=2")||linea.startsWith("R=3")||
                linea.startsWith("R=4")||linea.startsWith("R=5")||linea.startsWith("R=6")||linea.startsWith("R=7")||
                linea.startsWith("R=8")||linea.startsWith("R=9")||linea.startsWith("R=-")){
                pw.println(linea);
                // Dependiendo de la respuesta coloreamos el texto, ya sea rojo
                if(calificaciones[i].contains("INCORRECTO")){
                    pw.println("\\color red");
                    pw.println("[Incorrecto]");
                    pw.println("\\color inherit");
                }
                else{
                    // 0 verde
                    pw.println("\\color green");
                    pw.println("[Correcto]");
                    pw.println("\\color inherit");
                }
                i++;
            }

            // Verificamos si esa linea es una respuesta fraccionaria positiva, si lo es, verificamos en el arreglo cual fue su resultado
            else if (linea.contains("$\\frac")){

                pw.println(linea);
                // Dependiendo de la respuesta coloreamos el texto, ya sea rojo
                if(calificaciones[i].contains("INCORRECTO")){
                    pw.println("\\color red");
                    pw.println("[Incorrecto]");
                    pw.println("\\color inherit");
                }
                else{
                    // 0 verde
                    pw.println("\\end_inset");
                    pw.println("\\color green");
                    pw.println("[Correcto]");
                    pw.println("\\color inherit");
                    pw.println("\\begin_inset");
                }
                i++;
            }

            // Verificamos si esa linea es una respuesta fraccionaria negativa, si lo es, verificamos en el arreglo cual fue su resultado
            else if (linea.contains("-\\frac")){

                pw.println(linea);
                // Dependiendo de la respuesta coloreamos el texto, ya sea rojo
                if(calificaciones[i].contains("INCORRECTO")){
                    pw.println("\\color red");
                    pw.println("[Incorrecto]");
                    pw.println("\\color inherit");
                }
                else{
                    // 0 verde
                    pw.println("\\end_inset");
                    pw.println("\\color green");
                    pw.println("[Correcto]");
                    pw.println("\\color inherit");
                    pw.println("\\begin_inset");
                }
            }
        }
    }
}
```

Lo cual nos dará como resultado, un examen calificado, como el que se muestra a continuación:

4.- Evaluar el siguiente polinomio con $x= 8$

$$4x^3 + 3x + 2$$

↵

$$R=2074[\text{Correcto}]$$

↵

↵

5.- Evaluar la siguiente integral definida

$$\int_1^2 2x^2 + 3x + 2 dx$$

↵

$$R= \frac{67}{6}[\text{Correcto}]$$

↵

↵

6.- Evaluar el siguiente polinomio con $x= 7$

$$4x^2 + 2x + 5$$

↵

$$R=214[\text{Incorrecto}]$$

↵

CAPITULO 4 – Resultados y Conclusiones

Una vez finalizado el proceso de desarrollo, pruebas y depuración del proyecto, se ha cumplido con el objetivo del proyecto, que es:

Diseñar e implementar un sistema para la generación y evaluación de ejercicios de integración polinomial.

El cual se verifica satisfactoriamente, ya que el sistema presentado en este trabajo es capaz de generar ejercicios no solo de integrales definidas de polinomios, sino que también genera la evaluación de polinomios para una x dada.

Consecuentemente, se cumplieron los objetivos específicos, generando una interfaz de usuario amigable y fácil de entender, que permitirá la fácil utilización del sistema, así como la administración del mismo.

El sistema es capaz de generar un examen distinto cada vez, gracias al módulo generador de funciones matemáticas, el cual trabaja con funciones aleatorias lo que permite que los usuarios tengan a su disposición ejercicios nuevos, lo que propicia el mejoramiento de sus habilidades.

Todos los ejercicios que el usuario realice son verificados por el módulo verificador, lo cual es de mucha ayuda, porque siempre es muy importante tener una retroalimentación de los ejercicios realizados, lo cual le reforzará al alumno, si está utilizando el método correcto, o algo está fallando.

El sistema implementa un editor de texto basado en LaTeX, para presentar y responder el examen, pero, el alumno jamás tendrá que escribir con notación de este lenguaje, las respuestas que el dará serán completamente en notación matemática, incluso si son fracciones, el no escribirá la fracción con el símbolo “/” lo escribirá como si lo escribiera en una hoja de papel, lo cual cumple con una de las primicias de este proyecto, impulsar a que los alumnos trabajen con notación matemática.

Posibles continuaciones:

Este proyecto, puede tener dos continuaciones inmediatas, se podría ampliar los métodos generadores de ejercicios, lo cual daría posibilidad para abarcar una amplia gama de temas de matemáticas; inclusive podría extenderse a otras materias de educación.

Y segundo, generar un modo extra al modo “Evaluación”, este modo extra podría ser llamado “Aprendizaje” con el cual, se podría llevar al alumno paso a paso hasta llegar a la solución de diversos tipos de problemas matemáticos mediante pequeños consejos que aparecerán en la pantalla, así mismo evaluar el procedimiento realizado por el alumno.

Bibliografía:

[1] Jonathan Mendoza Barrón, “Desarrollo de reactivos y sus respectivas soluciones, para la elaboración de exámenes, a partir de una base de datos”, Proyecto terminal de Ingeniería en Computación, Departamento de sistemas, Universidad Autónoma Metropolitana Azcapotzalco, D.F, México, 2010.

[2] Luis Oswaldo Vázquez Mondragón, “Herramienta para crear exámenes con distintos formatos de pregunta/respuesta”, Proyecto terminal de Ingeniería en Computación, Departamento de sistemas, Universidad Autónoma Metropolitana Azcapotzalco, D.F, México, 2009.

[3] Mackichan [En línea]. Disponible: <http://www.mackichan.com>

[4] Moodle [En línea]. Disponible: http://docs.moodle.org/20/en/About_Moodle

[5] Java Plataform [En línea]. Disponible: <http://docs.oracle.com/javase/6/docs/api>

[6] Manual de Maxima [En línea]. Disponible:
<http://maxima.sourceforge.net/docs/manual/es/maxima.html>

[7] Lyx [En línea]. Disponible: <http://www.lyx.org/Development>

[8] Craig Larman, UML y Patrones, Introducción al Análisis y Diseño Orientado a Objetos, Primera Edición.