
**Universidad Autónoma Metropolitana
Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

Programa de Desarrollo Profesional en Automatización

Reporte del Proyecto Terminal

“Tutor inteligente para la enseñanza-aprendizaje de matemáticas a nivel básico”

Autora:

Gabriela Ramírez Monroy

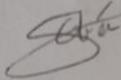
Asesor:

M. en C. Gerardo Aragón González

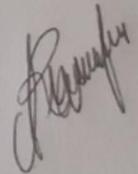
Asesor externo:

Dr. Marco Antonio Rodríguez Andrade

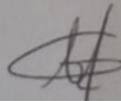
Yo, M. en C. Gerardo Aragón González, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Yo, Dr. Marco Antonio Rodríguez Andrade, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Yo, Gabriela Ramírez Monroy, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



CONTENIDO

RESUMEN	1
INTRODUCCIÓN	1
CAPÍTULO 1	2
¿Por qué IDE Eclipse?	2
PARTE 1 "Proyecto Eclipse"	3
PARTE 2 "Herramientas de proyecto Eclipse"	3
PARTE 3 "Proyecto tecnológico Eclipse"	3
Arquitectura de Eclipse:	4
Diferencias entre AWT Y SWT para desarrollo de una GUI	5
Java Swing	6
Uso de Swing/AWT para crear una GUI	8
Manejador MySQL	8
CAPÍTULO 2	9
¿Cuál es la función de un tutor?	9
Objetivo general	10
Objetivos específicos	10
Descripción del tutor	10
Diagrama de módulos	11
Ejemplo de un problema en el tutor	12
Análisis de sistema	13
Requisitos funcionales	13
Diagrama de actividades	13
Diagrama de actividades del tutor	14
Casos de uso	22
Casos de uso del tutor	24
Diagramas de secuencia	28
Diagrama de secuencia del tutor	29
Identificación de requerimientos funcionales	32
Requerimientos no funcionales	36

CAPÍTULO 3	38
Manual de usuario	38
CONCLUSIONES	55
APÉNDICE	56
BIBLIOGRAFÍA	82

RESUMEN

En el desarrollo de la enseñanza-aprendizaje, han aparecido nuevas herramientas y métodos para mejorar la práctica educativa [1]. Esto debido al avance tecnológico que permite crear dichas herramientas para ayudar al profesor, este proyecto se enfoca en la enseñanza de matemáticas a nivel básico.

La implementación de un tutor inteligente implica todo lo que un maestro humano podría hacer: selecciona el material apropiado, prepara el ejercicio o evaluación con base a una metodología, su la retroalimentación necesaria y proporciona estrategias de resolución de problemas.

Se desarrolla como proyecto terminal un tutor inteligente que cumpla con las características anteriores, utilizando el entorno de desarrollo Eclipse, con el lenguaje de programación JAVA, en conjunto con el manejador de base de datos MySQL¹ y utilizando las bibliotecas² AWT y Swing, las cuales son utilizadas para el desarrollo de un entorno gráfico para el usuario, que en específico será el alumno desde primer hasta tercer año de secundaria en el área de las matemáticas.

INTRODUCCIÓN

El **Programa de Desarrollo Profesional en Automatización** (PDPA) se fundó en 1992, impulsado por tres profesores Titulares del departamento de Energía, enmarcado en un acuerdo de colaboración signado por la Rectoría de la Unidad Azcapotzalco y la empresa Parker Hannifin de México. Entre las actividades primordiales del PDPA se cuentan la producción de prototipos y soluciones tecnológicas para el sector productivo, desarrollados con base en acuerdos de colaboración técnica con diferentes actores económicos de nuestro país.

Las actividades de desarrollo tecnológico incluyen la participación de alumnos de las licenciaturas de ingeniería, con el propósito de ponerlos en contacto directo con las aplicaciones de la potencia fluida, el control de movimiento y la automatización de procesos de manufactura.

¹ MySQL: Es un sistema administrativo relacional de bases de datos (RDBMS por sus siglas en inglés Relational Database Management System).

² Biblioteca: Ofrecen al programador un conjunto bien definido de funciones para realizar tareas comunes.

Al conjunto con estas actividades, se añade el interés de los profesores por motivar y reforzar el aprendizaje de los alumnos, siendo la un claro ejemplo de ese interés el reporte de Metodologías para la construcción de evaluaciones [2], donde abordan los problemas que la educación plantea, particularmente aquellos en la enseñanza matemática.

Este reporte se compone por tres capítulos: el primero hace una descripción de las herramientas utilizadas para el desarrollo del tutor. El segundo aborda la parte de la ingeniería de software del sistema, la cual involucra el análisis previo de la situación, el diseño del proyecto, el desarrollo del tutor y la elaboración de las pruebas necesarias para confirmar su correcto funcionamiento.

Cabe destacar que en el de desarrollo del software está implicado el ciclo de vida del software, que está formado por cuatro etapas: concepción, elaboración, construcción y transición.

También implica fijar el alcance de la aplicación, definir el plan de desarrollo, detallar las características y fundamentar la arquitectura, la construcción y el resultado final del tutor.

El tercer capítulo involucra el manual de uso del tutor, siendo un documento de comunicación técnica con el propósito de brindar asistencia al estudiante que lo utiliza. Finalmente en el Apéndice se incluye el código completo en lenguaje JAVA del tutor realizado en este proyecto terminal.

CAPITULO 1

¿POR QUÉ IDE³ ECLIPSE?

Eclipse es un entorno de desarrollo integrado de Código abierto y Multiplataforma, esto lo hace una completa y avanzada plataforma de programación, desarrollo y compilación de diversos elementos como es el caso de aplicaciones JAVA [3]. Se divide en tres partes, que a su vez también son divididas:

³ IDE: entorno de desarrollo integrado, plataforma completa con herramientas y funciones necesarias, además de una interfaz atractiva para el desarrollo de programas.

PARTE 1 "PROYECTO ECLIPSE"

Es un proyecto de desarrollo de software con la ventaja de que es libre, el cual tiene la finalidad de proporcionar una plataforma de desarrollo de herramientas integradas y completas. Se puede dividir en tres partes:

1. La plataforma, que contiene las herramientas necesarias para facilitar el desarrollo.
2. JDT⁴: añade a la plataforma un IDE de JAVA completamente equipado, contiene un editor de texto con resaltado de sintaxis donde se puede ver el contenido del fichero en el que se está trabajando, todo esto con una compilación en tiempo real y un depurador.
3. PDE⁵: conjunto de herramientas de ayuda en las tareas de desarrollo, prueba, depuración, construcción y distribución de plug-ins⁶.

PARTE 2 "HERRAMIENTAS DE PROYECTO ECLIPSE"

Proporciona una variedad de herramientas para que el desarrollador minimice la duplicación y fomente la interoperabilidad entre todos los tipos de herramientas.

PARTE 3 "PROYECTO TECNOLÓGICO ECLIPSE"

Tiene como objetivo brindar nuevas y mejores vías de comunicación a estudiantes, desarrolladores, maestros, etc. para que se involucren en la evolución y mejora del proyecto.

Al realizar la descarga del SDK ⁷ de Eclipse, se incluye las herramientas anteriormente descritas. Cabe señalar que para que Eclipse pueda ejecutarse, debido a que está escrito en JAVA debe tener instalado el JDK⁸ o el JRE⁹.

⁴ JDT: Java Development Toolkit, herramientas de desarrollo de Java.

⁵ PDE: Plug-in Development Environment, entorno de desarrollo de plug-in.

⁶ Plug-in: es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

⁷ SDK: Software Development Kit, es un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto.

⁸ JDK: Java Development Kit, kit de desarrollo de JAVA.

⁹ JRE: Java Runtime Environment, entorno en tiempo de ejecución de Java

ARQUITECTURA DE ECLIPSE:

Debido a su detallado diseño y su funcionalidad la Plataforma Eclipse nos ofrece, es una opción muy utilizada por los desarrolladores, sin embargo, por sí sola no funcionaría, lo que le da un verdadero valor y desempeño es el uso de unidades: los plug-ins.

Por esto, Eclipse se encuentra bien estructurado por un conjunto subsistemas los cuales son implementados en uno o más plug-ins que corren sobre una pequeña plataforma de ejecución Dichos subsistemas definen puntos de extensión para permitir agregar funcionalidad a la plataforma[3].

La estructura está definida en la siguiente figura:



Plataforma de tiempo de ejecución: es el único componente de Eclipse que no es un plug-in. Cuando una herramienta agrega una implementación para determinado punto de extensión se dice que añade una extensión a la plataforma. A su vez, cada plug-in puede definir sus propios puntos de extensión.

Espacio de trabajo: es el bloque central para los archivos regulares que son específicos de cada usuario, y sobre los que actúan las diferentes herramientas instaladas en la plataforma. Consta de uno o más proyectos donde cada uno se mapea en un directorio especificado por el usuario en el sistema de archivos.

Banco de trabajo: es el aspecto visual que permite al usuario navegar por los recursos y utilizar las herramientas. Se presentan las partes visuales, los cuales se dividen en editores y vistas.

SWING¹⁰: es una biblioteca gráfica para desarrollo de aplicaciones java, que permite herramientas para el usuario, tales como cajas de texto, botones, desplegados y tablas.

JFace: es un interfaz gráfico implementado sobre SWT y Swing.

DIFERENCIAS ENTRE AWT Y SWT PARA DESARROLLO DE UNA GUI

GUI (Interfaz Gráfica de Usuario) es una interfaz de usuario en la que una persona interactúa con la información digital a través de un entorno gráfico de simulación.

Para diseñar una interfaz es necesario pasar por cuatro etapas:

1. Análisis de requerimientos del sistema, se analizan las tareas que va a realizar.
2. Generación de prototipos para investigar desde lo general hasta el detalle y desarrollo de la aplicación.
3. Pruebas de la aplicación.
4. Conclusión, análisis de los datos, elaboración del informe, resultados y recomendaciones.

Al ser Java un lenguaje multiplataforma, se descartó en un principio el uso de toolkits nativos o específicos de una plataforma. Fue entonces cuando se creó un toolkit

¹⁰ SWT: Standard Widget Toolkit, es un conjunto de componentes de interacción desarrollados por el proyecto Eclipse.

propio de la máquina virtual de JAVA y que formara parte de la biblioteca de clases. A este toolkit se le llamó AWT¹¹.

A diferencia de AWT, SWT proporciona todos los controles que se suponen imprescindibles en cualquier interfaz gráfica moderna. Casi todo el código de las clases SWT se limita a hacer de puente entre Java y la plataforma, excepto en el caso de los componentes SWT sin equivalente en la plataforma. SWT usa, siempre que puede, la información que proporciona la API gráfica de cada plataforma.

Cualquier aplicación escrita en Java podría utilizar la JNI¹² para establecer correspondencia entre los métodos Java y las capacidades gráficas propias de la plataforma usada. La ventaja de utilizar SWT es que evita a los programadores escribir su código JNI[4].

JAVA SWING

El paquete Swing es parte de la JFC¹³ en la plataforma Java. La JFC provee facilidades para ayudar a la gente a construir GUIs. Swing abarca componentes como botones, tablas, marcos, etc.

Las componentes Swing se identifican porque pertenecen al paquete *javax.swing*. Swing existe desde la JDK 1.1. Antes de la existencia de Swing, las interfaces gráficas con el usuario se realizaban a través de AWT, de quien Swing hereda todo el manejo de eventos [5].

Usualmente, para toda componente AWT existe una componente Swing que la reemplaza, por ejemplo, la clase Button de AWT es reemplazada por la clase JButton de Swing (el nombre de todas las componentes Swing comienza con "J").

¹¹ AWT: Abstract WindowToolkit, API de Java que permite a los programadores hacer aplicaciones JAVA con componentes GUI.

¹² JNI: Java Native Interface, es un marco de trabajo de programación que permite que un programa escrito en Java ejecutado en la máquina virtual java (JVM) pueda interactuar con programas escritos en otros lenguajes como C, C++ y ensamblador.

¹³ JFC: Java Foundation Classes, es una serie de funciones ya creadas para que se puedan utilizar al crear programas en lenguaje Java.

Componentes de java Swing:

Swing consta de nueve paquetes y cientos de clases e interfaces. No obstante, la clase `JComponent` de `java.awt.swing` es la clase superior de la jerarquía de componentes Swing, es decir, es prácticamente la base en el diseño de las ventanas de aplicación.

La clase `JComponent` es una subclase de la clase `java.awt.container` y, por lo tanto, es a la vez un componente y un contenedor en el sentido del AWT. Dado que

`JComponent` es la superclase de todos los componentes Swing, todos ellos descienden de `java.awt.Container` y `java.awt.Component`. Ver diagrama 1.

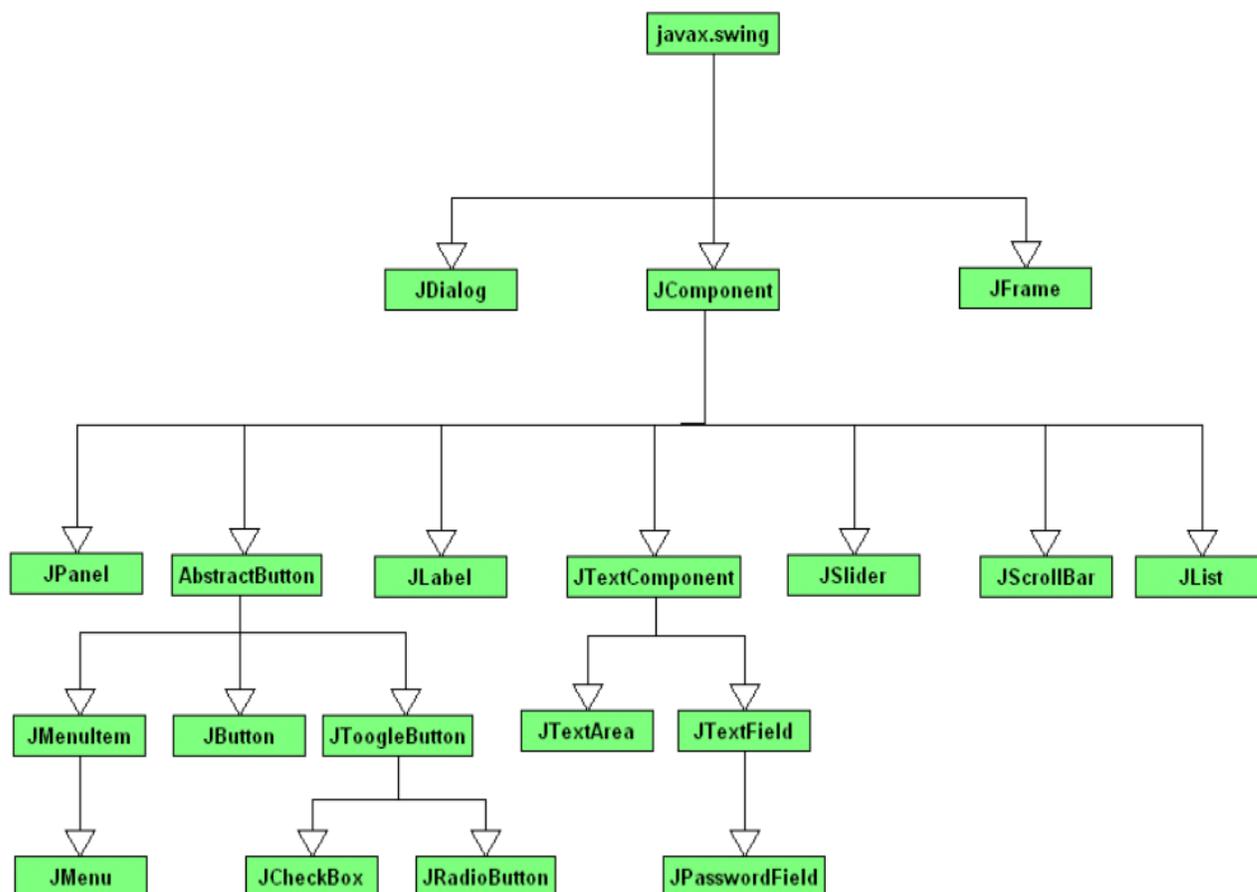


Diagrama 1

Características de los componentes:

- Todos los componentes heredan de `javax.swing.JComponent`
- `JFrame` será la base para la aplicación principal.
- `JDialog` construirá los diálogos (ventanas).
- El resto de clases serán componentes simples.
- Usar en todas las clases `import javax.swing.*` e `import java.awt.*`.

Las ventajas de Swing son:

- El diseño en Java puro posee menos limitaciones de plataforma.
- El desarrollo de componentes Swing es más activo.
- Los componentes de Swing soportan más características.

USO DE SWING/AWT PARA CREAR UNA GUI

Swing y AWT muchas veces son vistos como tecnologías competidoras, cuestionándonos cuál de las dos es preferible utilizar. Algunas personas tienen opiniones fuertes sobre cual resulta preferible para utilizar en el desarrollo de una interfaz de usuario. Sin embargo, en el mundo real, los extremos son a menudo poco práctico.

Algunos casos de uso válidos requieren ambas tecnologías para coexistir en una sola aplicación. Mientras que la mezcla de los dos juegos de herramientas no es una tarea sencilla, se puede hacer, y se puede hacer de tal manera que los dos conjuntos de herramientas se integran sin problemas.

MANEJADOR MYSQL

MySQL es un sistema de administración de bases de datos para relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo,

permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl, JAVA, y su integración en distintos sistemas operativos [6].

CAPITULO 2

¿CUÁL ES LA FUNCIÓN DE UN TUTOR?

La función de un tutor es optimizar el proceso de enseñanza-aprendizaje, donde se debe tomar en cuenta la capacidad y el potencial desarrollado en cada alumno, como un proceso de la orientación escolar, realizada en paralelo o conjunto con el trabajo docente, permitiendo a estudiantes lograr un mejor desempeño a lo largo de su vida académica.

Los sistemas de tutores inteligentes (ITS) buscan imitar los métodos y diálogo de tutores humanos naturales. La implementación de ITS incorpora mecanismos computacionales y representaciones del conocimiento de los campos de inteligencia artificial, lingüística computacional, y de ciencias cognoscitivas [7].

La funcionalidad de un ITS incluye todo lo que un maestro humano podría hacer: selecciona el material apropiado, "prepara" el ejercicio, monitorea la actividad del estudiante, proporciona sugerencias durante los ejercicios y la retroalimentación necesaria, personaliza el estilo de la presentación al estilo del estudiante, pregunta y contesta a las preguntas.

Aprender y enseñar con tutores humanos e ITS se caracteriza por la mezcla de diálogo y, a menudo, el uso de lenguaje natural. Para lograr tales interacciones, los ITS incluyen mecanismos de tutores humanos con algoritmos avanzados y arquitecturas computacionales, como los sistemas de reglas de producción o sistemas expertos, las gramáticas generadoras, redes Bayesianas¹⁴, espacios semánticos de alto nivel, sistemas de control difuso y sistemas dinámicos no-lineales. Con estos métodos los ITS usan diálogo tutelar en lenguaje natural para responder adaptándose a las emociones del estudiante y los estados motivacionales, y eficazmente enlazar la interfaz hombre-máquina, mientras se adaptan al perfil multidimensional del estudiante.

¹⁴ Redes Bayesianas: gráfico acíclico dirigido es un modelo gráfico probabilístico (un tipo de modelo estadístico) que representa un conjunto de variables aleatorias y sus dependencias condicionales a través de un gráfico acíclico dirigido.

OBJETIVO GENERAL

Diseñar y desarrollar un ambiente visual computacional en la enseñanza-aprendizaje de las matemáticas para alumnos de Educación Básica (Secundaria). La finalidad es de motivar y reforzar el aprendizaje de matemáticas en los alumnos.

OBJETIVOS ESPECÍFICOS

- Proporcionar a estudiantes material de apoyo, de utilidad para adquirir y reafirmar conocimientos matemáticos correspondientes a la Educación Básica, enfocándose exclusivamente en los tres grados de nivel secundaria.
- Brindar a docentes que imparten esta materia un apoyo didáctico de enseñanza.
- Colaborar en el desarrollo de la capacidad del estudiante para resolver problemas matemáticos, mostrándoles lo útil que resulta para aspectos de la vida cotidiana.

DESCRIPCIÓN DEL TUTOR

El tutor inteligente está dividido en tres módulos que corresponden a cada uno de los grados de educación secundaria. Los temas considerados en cada uno están delineados con base en la serie de libros "Matemáticas para la vida" [8].

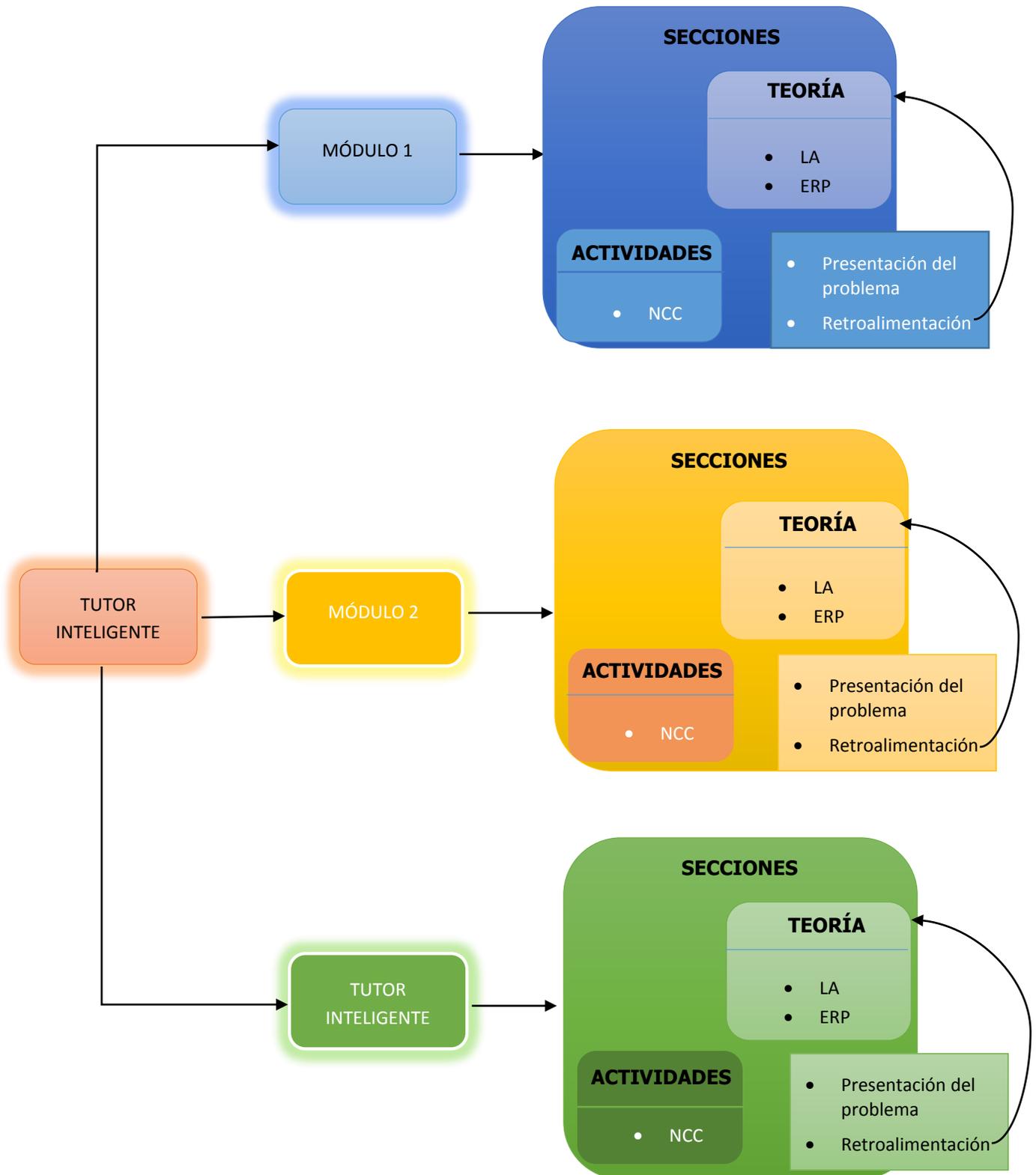
Cada uno de los tres módulos cuenta con tres secciones:

- Lección de Aprendizaje (LA)
- Estrategias de resolución de problemas (ERP)
- Planteamiento de Problemas, narración de un cuento clásico (NCC)

Las tres secciones anteriores están distribuidas en:

- Teoría: proporciona al estudiante las herramientas teóricas y métodos de resolución de problemas dependiendo su nivel matemático.
- Actividades: el estudiante resuelve un conjunto de problemas relacionados con la teoría aunado con la narración de un cuento clásico.

DIAGRAMA DE MÓDULOS

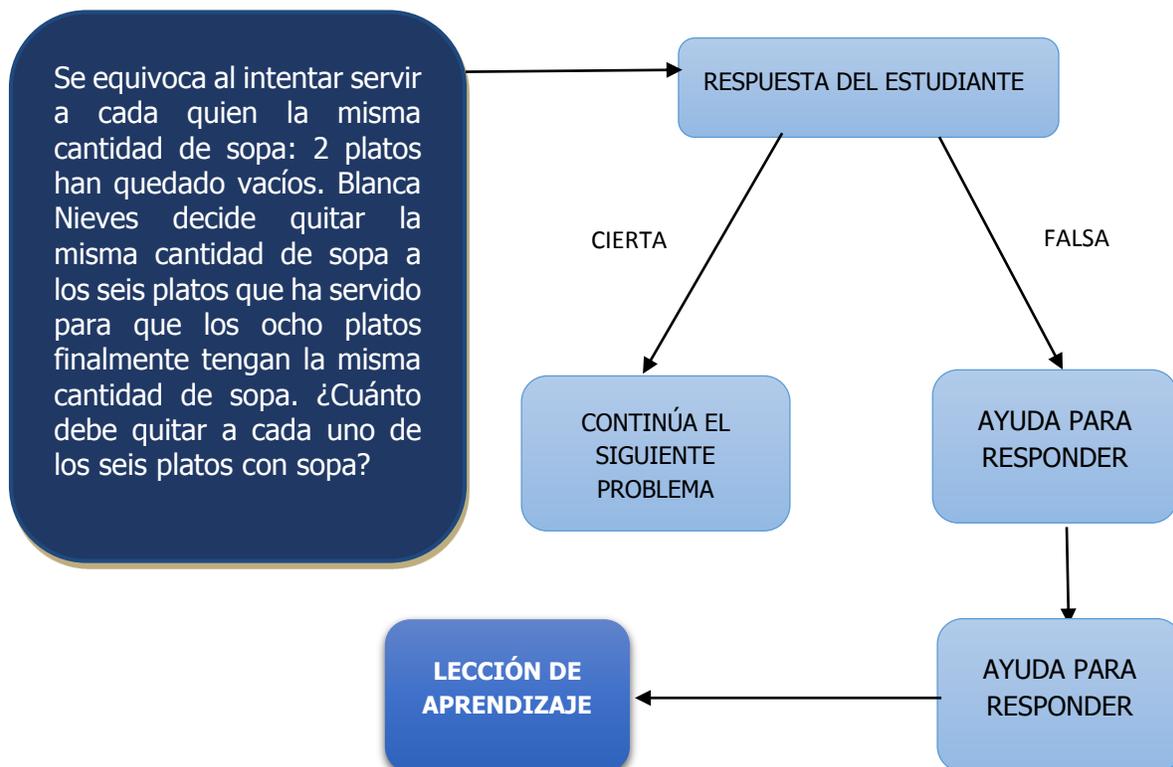


EJEMPLO DE UN PROBLEMA EN EL TUTOR

En la sección de actividades se lleva a cabo la narración de un cuento y en cada situación de este se plantea un problema a resolver, se analizan las respuestas del alumno y se continúa la historia, en caso contrario se le proporciona ayuda para resolverlo o se repasa el tema (lección de aprendizaje).

Como ejemplo de cómo será la forma en que el alumno solucionará el problema se presenta una breve descripción de una situación dentro del cuento y el problema matemático a resolver (módulo 2).

Blanca Nieves tiene que organizar los deberes de la casa mientras ellos trabajan. Ha decidido preparar sopa para los enanos quienes regresan cansados...



ANÁLISIS DE SISTEMA

El análisis de sistema, trata básicamente de determinar los objetivos y límites del tutor, el cual es el sistema objeto de análisis, caracterizar su estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos y evaluar sus consecuencias.

Para realizar el análisis del tutor, se deben identificar los requisitos funcionales, los cuales se obtienen de un conjunto de diagramas y descripciones que se mostrarán en este reporte.

REQUISITOS FUNCIONALES

Describen la interacción entre el tutor y su ambiente independientemente de su forma en que fue implementado [9]. El ambiente incluye al estudiante y cualquier otro sistema externo que podría interactuar con el tutor.

Para detallar estos requerimientos es necesario elaborar un conjunto de diagramas donde se detallan los escenarios, actores, actividades, entre otros, que intervienen en el tutor [10].

DIAGRAMA DE ACTIVIDADES

Un diagrama de actividades se usa para mostrar la secuencia de actividades. Los diagramas de actividades muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad.

Una actividad es la especificación de una secuencia de comportamiento. Muestra un rectángulo con las puntas redondeadas adjuntando todas las acciones, flujos de control y otros elementos que constituyen la actividad.

Una actividad se representa dentro de:



El flujo de ejecución debe tener inicio y fin y son marcados por las señales:

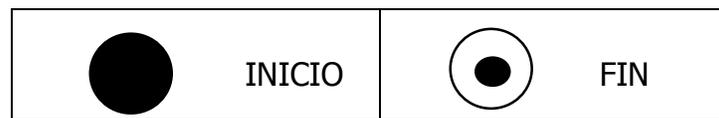
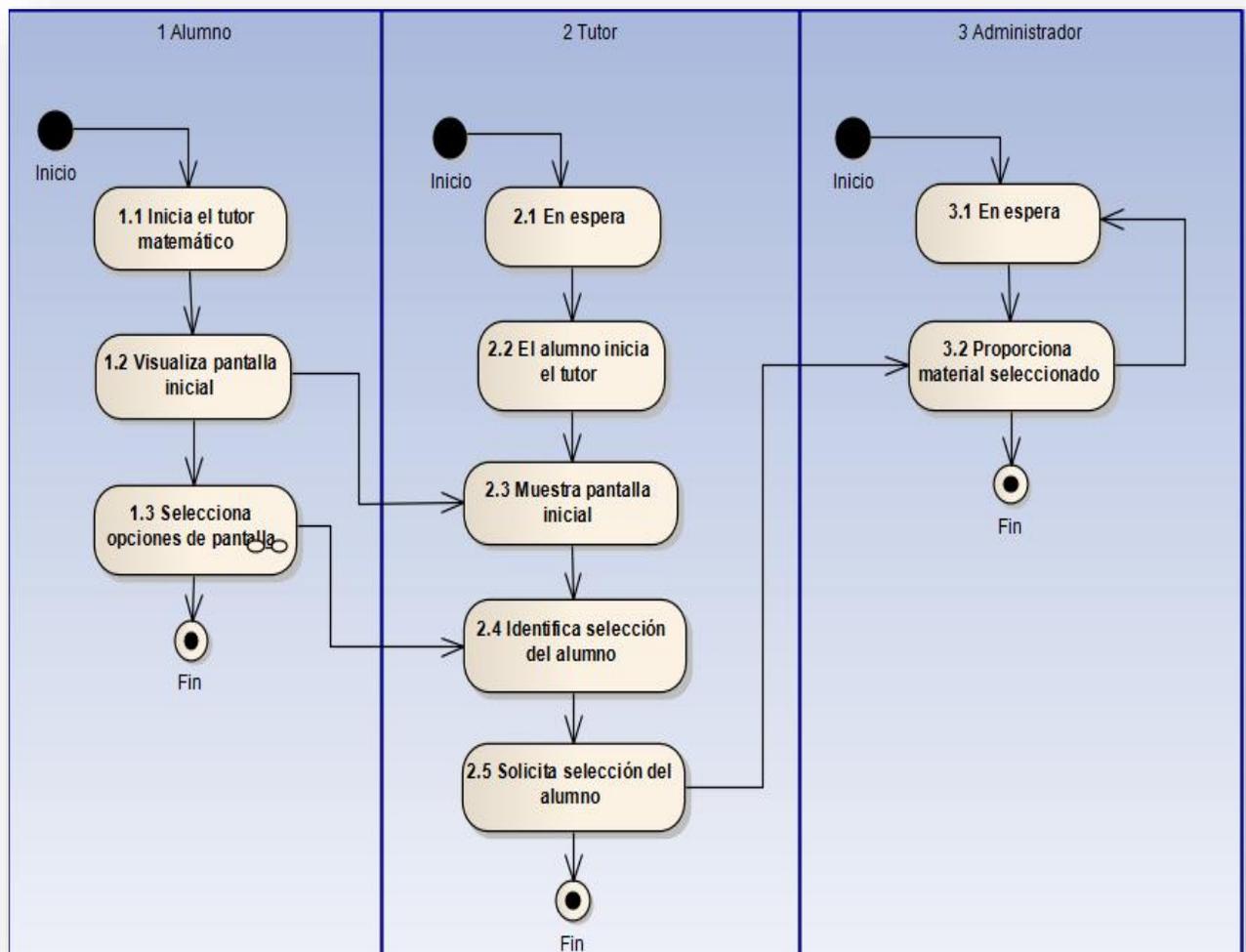


DIAGRAMA DE ACTIVIDADES DEL TUTOR

Pantalla Inicial

Es la primera pantalla que el alumno visualiza al iniciar el tutor



1. Alumno

ACTIVIDAD	PRECONDICIÓN	DESCRIPCIÓN
Inicia tutor matemático	<ul style="list-style-type: none"> • Ejecutable del tutor en la computadora • Ícono del tutor 	El alumno inicia el tutor de igual forma que una aplicación en la computadora, contando con un ícono de acceso directo.
Visualiza pantalla inicial	<ul style="list-style-type: none"> • Correcta ejecución del tutor 	El alumno visualiza la pantalla inicial del tutor, después de ejecutar la aplicación.
Selecciona opciones de pantalla	<ul style="list-style-type: none"> • Tutor muestra las opciones iniciales en la pantalla 	Después de visualizar la pantalla inicial, el alumno selecciona la opción que requiere.

1. Tutor

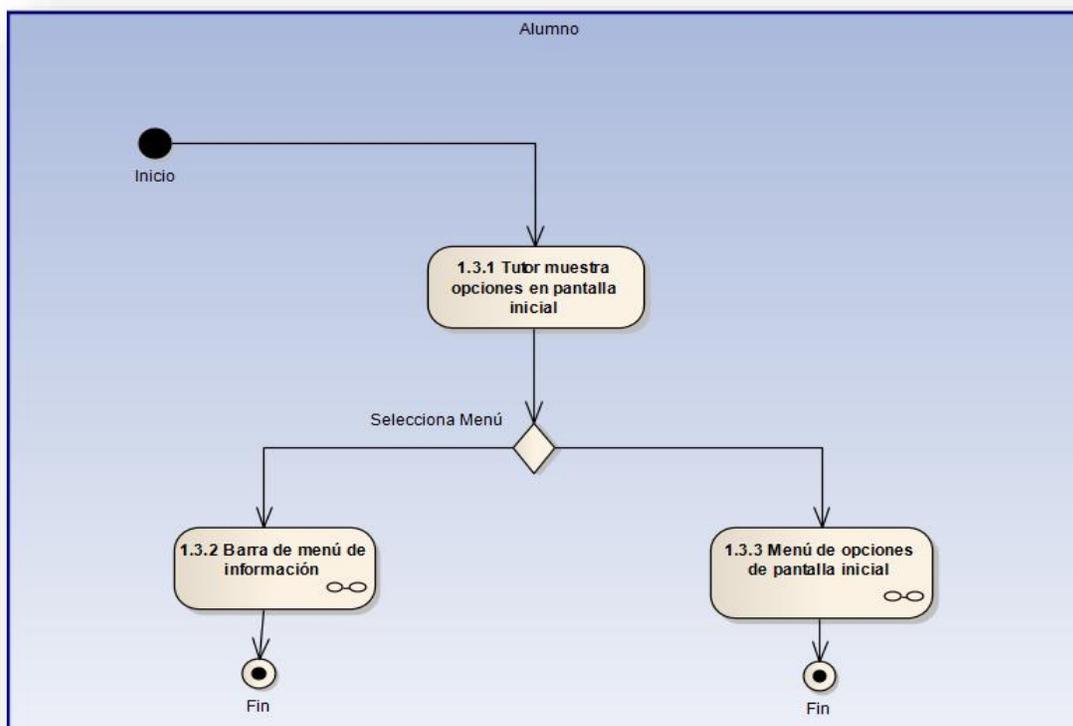
ACTIVIDAD	PRECONDICIÓN	DESCRIPCIÓN
En espera	<ul style="list-style-type: none"> • Tutor instalado en la computadora para uso del alumno 	Se deben identificar las condiciones técnicas necesarias para que el tutor funcione en el equipo. Una vez validados los requerimientos, el tutor se encuentra como cualquier aplicación dentro del equipo.
Alumno inicia tutor	<ul style="list-style-type: none"> • Alumno requiere utilizar el tutor matemático en el ordenador 	El alumno inicia el tutor para obtener recursos teóricos o desarrollar actividades de retroalimentación.
Muestra pantalla inicial	<ul style="list-style-type: none"> • Tener validadas las opciones que son proporcionadas en la pantalla inicial 	El tutor muestra las opciones correspondientes a la pantalla inicial.
Identifica selección de alumno	<ul style="list-style-type: none"> • El alumno selecciona una opción de la pantalla inicial 	El tutor identifica la selección que el alumno realizó al visualizar la pantalla inicial.

Solicita selección de alumno	<ul style="list-style-type: none"> Tutor registra la selección del alumno dentro de la pantalla inicial 	El tutor solicita al administrador la opción que el alumno eligió en la pantalla inicial.
-------------------------------------	--	---

2. Administrador

ACTIVIDAD	PRECONDICIÓN	DESCRIPCIÓN
En espera	<ul style="list-style-type: none"> Administrador debe estar conectado al el tutor 	El Administrador se encuentra en espera para recibir solicitudes del tutor, dada la selección del alumno en la pantalla inicial.
Proporciona material seleccionado	<ul style="list-style-type: none"> Tutor solicita la opción señalada por el alumno en la pantalla inicial 	Proporciona al tutor el recurso solicitado por el alumno para que lo visualice en pantalla

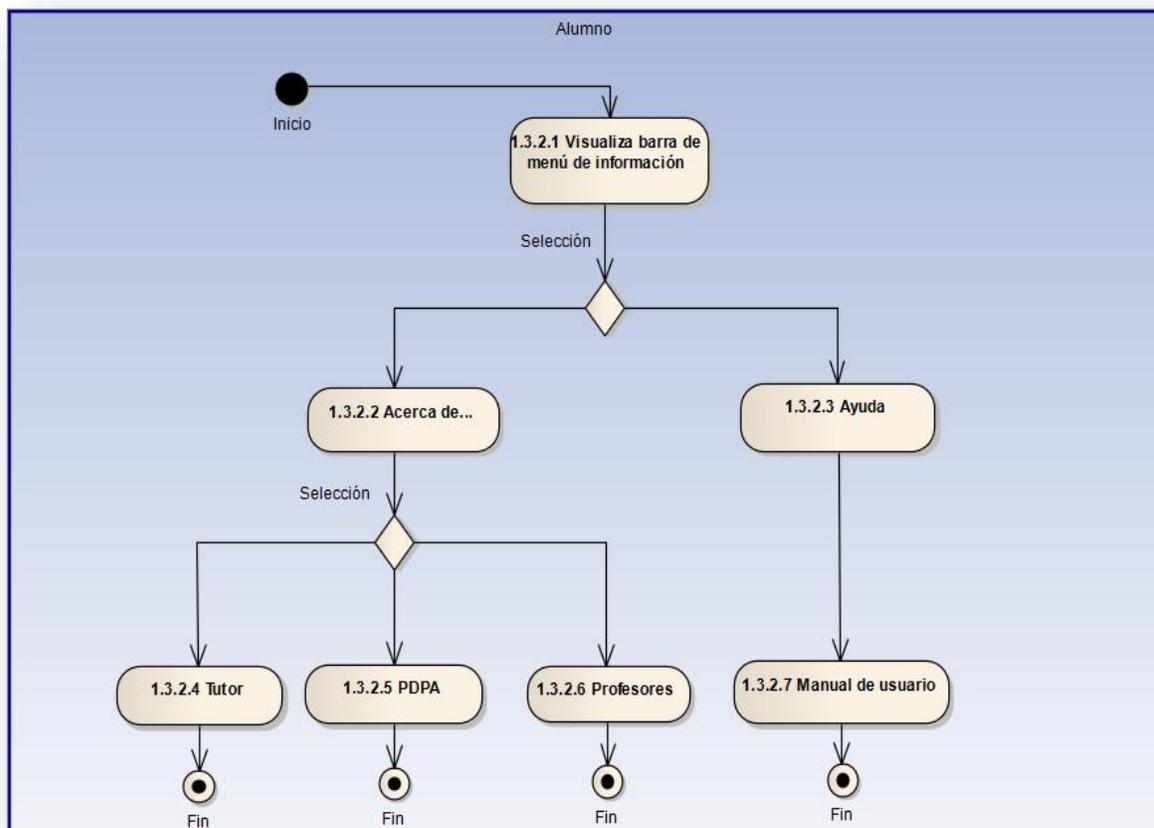
Pantalla Inicial (Selección de opciones)



1. Alumno

ACTIVIDAD	PRECONDICIÓN	DESCRIPCIÓN
Tutor muestra opciones en pantalla inicial	<ul style="list-style-type: none"> El alumno ejecuta el tutor en la computadora 	En la pantalla inicial del tutor se muestra la barra de menú de información y el menú de opciones.
Barra de menú de información	<ul style="list-style-type: none"> El alumno requiere información del tutor 	Se muestra al alumno las opciones referentes a las características del tutor.
Menú de opciones de pantalla inicial	<ul style="list-style-type: none"> El alumno requiere realizar acción sobre el tutor 	Se muestra al alumno las opciones referentes a las acciones en el tutor.

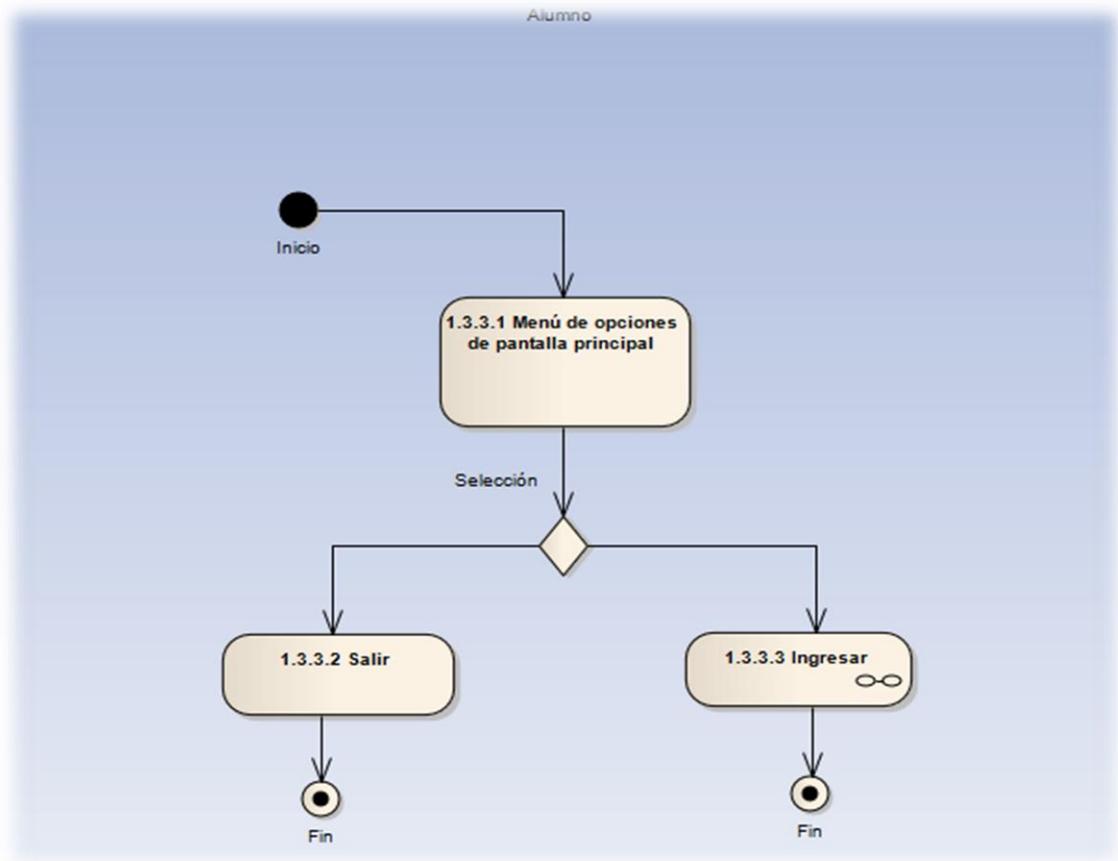
Pantalla Inicial (Barra de menú de información)



1. Alumno

ACTIVIDAD	PRECONDICIÓN	DESCRIPCIÓN
Visualiza barra de menú de información	<ul style="list-style-type: none"> Alumno requiere algún tipo de información 	El alumno selecciona la barra de menú ubicada en la parte superior de la pantalla inicial.
Acerca de...	<ul style="list-style-type: none"> Alumno requiere información respecto al entorno de desarrollo del tutor 	La opción de acerca de... informa al alumno sobre el entorno involucrado en el desarrollo del tutor matemático.
Tutor	<ul style="list-style-type: none"> Alumno requiere información sobre los antecedentes del tutor 	El tutor proporciona información sobre su definición y el aporte que hace al estudio de las matemáticas a nivel secundaria.
PDPA	<ul style="list-style-type: none"> Alumno requiere información respecto al área donde fue desarrollado el tutor 	El tutor proporciona información sobre el área en la que fue desarrollado y cuáles fueron los factores que impulsaron su diseño.
Profesores	<ul style="list-style-type: none"> Alumno requiere información respecto a los profesores involucrados en el diseño y desarrollo del tutor 	El tutor proporciona información sobre los profesores que diseñaron y planearon el tutor y los problemas que se presentan en los módulos.
Ayuda	<ul style="list-style-type: none"> Alumno requiere información del manejo del tutor 	El tutor proporciona información al usuario sobre cómo manejar el tutor y su funcionamiento.
Manual de usuario	<ul style="list-style-type: none"> El alumno requiere instrucciones de uso del tutor 	El tutor proporciona un manual de usuario para que el alumno identifique las partes y funcionamiento de éste.

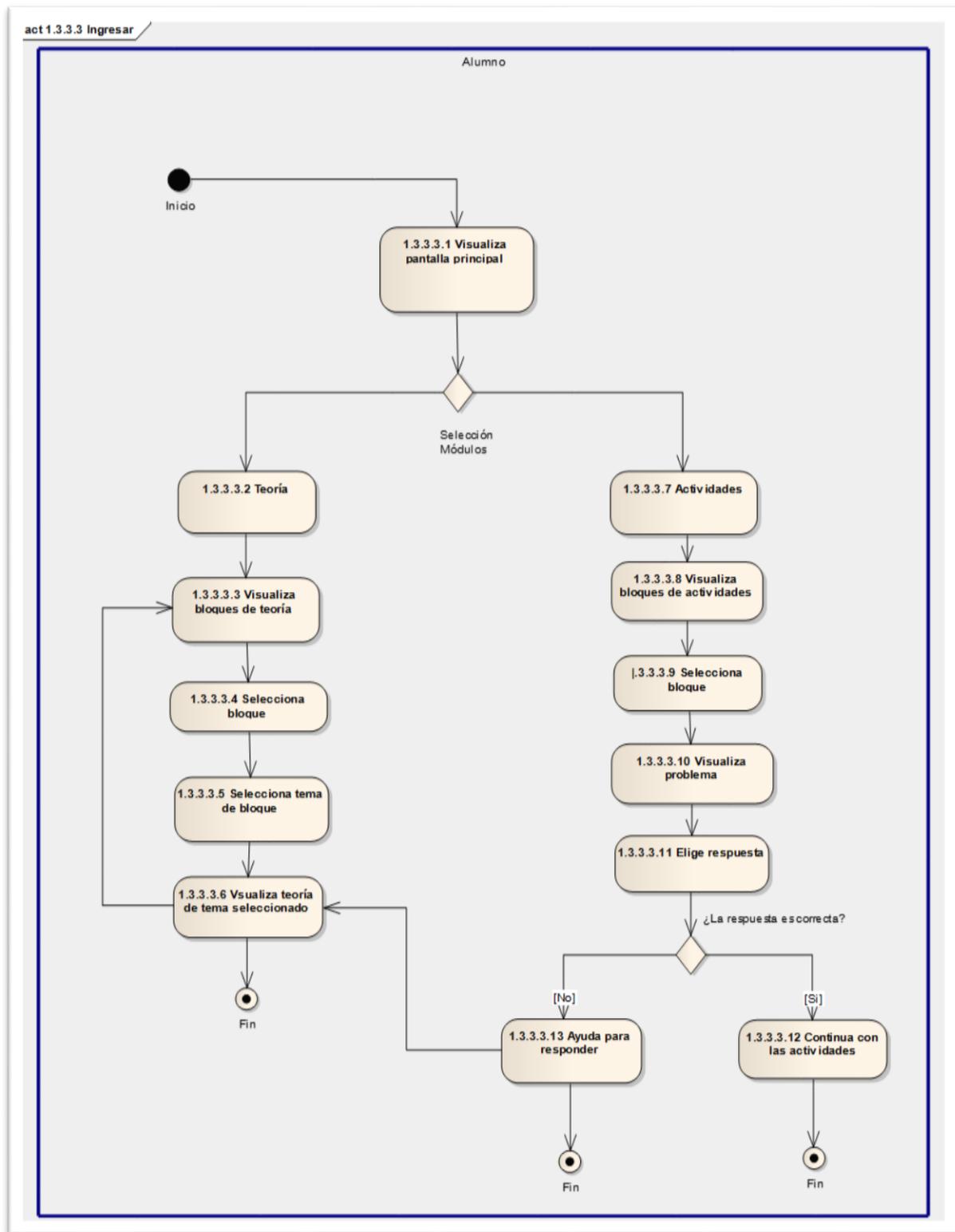
Pantalla Inicial (Opciones de pantalla inicial)



1. Alumno

ACTIVIDAD	PRECONDICIÓN	DESCRIPCIÓN
Menú de opciones de pantalla inicial	<ul style="list-style-type: none"> El alumno requiere realizar una acción sobre el tutor 	Se visualizan en la pantalla inicial dos botones: <ol style="list-style-type: none"> Salir Ingresar
Salir	<ul style="list-style-type: none"> El alumno requiere salir del tutor 	Se visualiza en la pantalla inicial del tutor el botón de salida.
Ingresar	<ul style="list-style-type: none"> El alumno requiere ingresar al menú principal del tutor 	Se visualiza en la pantalla inicial del tutor el botón para ingresar a la pantalla inicial.

Pantalla principal del tutor



1. Alumno

ACTIVIDAD	PRECONDICIÓN	DESCRIPCIÓN
Visualiza pantalla principal	<ul style="list-style-type: none"> Alumno requiere ingresar a la pantalla principal del tutor 	El alumno ingresa a la pantalla principal del tutor donde encontrará tres módulos diferentes, uno por cada año de secundaria. Podrá seleccionar teoría o actividades en cada uno de ellos.
Teoría	<ul style="list-style-type: none"> Alumno requiere ingresar a la sección de teoría del tutor 	El alumno selecciona ingresar a la sección de teoría de cualquier de los tres módulos del tutor.
Visualiza bloques de teoría	<ul style="list-style-type: none"> Alumno requiere visualizar los bloques de temario del módulo elegido 	El alumno visualiza la pantalla de bloques de teoría del módulo seleccionado.
Selecciona bloque	<ul style="list-style-type: none"> Alumno requiere visualizar teoría de un bloque de temario 	El alumno selecciona un bloque de teoría para revisar el material teórico correspondiente.
Selecciona tema de bloque	<ul style="list-style-type: none"> Alumno requiere seleccionar un tema específico del bloque de teoría 	El tutor selecciona un tema en específico del bloque de teoría que seleccionó.
Visualiza teoría de tema seleccionado	<ul style="list-style-type: none"> Alumno requiere visualizar la teoría de un tema específico del bloque de teoría 	El tutor visualiza la teoría del tema que seleccionó del bloque de teoría.
Actividades	<ul style="list-style-type: none"> Alumno requiere ingresar a la sección de actividades 	El alumno selecciona ingresar a la sección de actividades de cualquier de los tres módulos del tutor.
Visualiza bloque de actividades	<ul style="list-style-type: none"> Alumno requiere visualizar los bloques de actividades del módulo seleccionado 	El alumno visualiza la pantalla de bloques de actividades

Visualiza problema	<ul style="list-style-type: none"> Alumno requiere realizar actividades correspondientes al bloque seleccionado 	El alumno visualiza la historia relacionada con la actividad y las opciones de respuesta que tiene el problema presentado.
Elige respuesta	<ul style="list-style-type: none"> Alumno visualiza las posibles respuestas al problema presentado 	El alumno selecciona una respuesta de las que son presentadas en la pantalla de la actividad.
Ayuda para responder	<ul style="list-style-type: none"> Alumno falla en respuesta elegida 	El alumno visualiza la pantalla de ayuda para poder seleccionar la respuesta correcta. El alumno tiene también la opción de revisar la teoría correspondiente.
Continua con las actividades	<ul style="list-style-type: none"> Alumno acierta en la respuesta elegida 	El alumno continúa con la siguiente actividad.

CASOS DE USO

Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. Un caso de uso es iniciado por un actor. A partir de ese momento, ese actor, junto con otros actores, intercambian datos o control con el sistema, participando de ese caso de uso [11].

El nombre de un caso de uso se expresa con un verbo en gerundio, seguido generalmente por el principal objeto o entidad del sistema que es afectado por el caso. Gráficamente, los casos de uso se representan con un óvalo, con el nombre del caso en su interior.

Los casos de uso tienen las siguientes características:

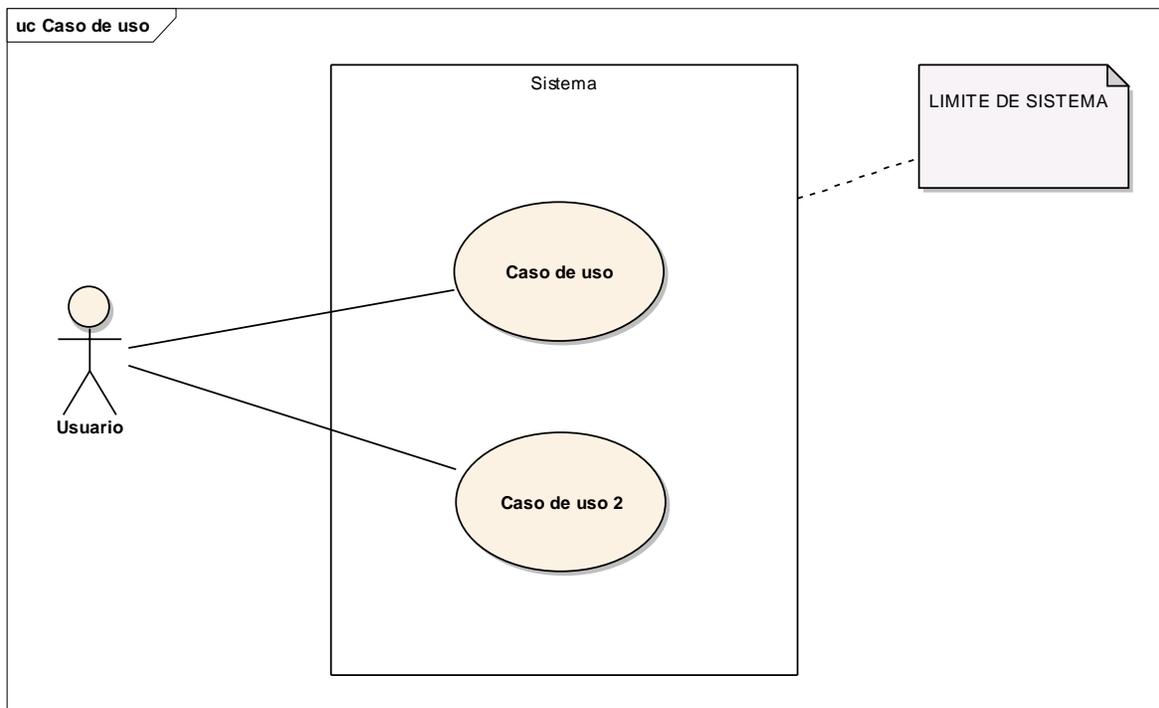
1. Son expresados desde el punto de vista del actor.
2. Se documentan con texto informal.
3. Describen tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él.

4. Son iniciados por un único actor.
5. Están acotados al uso de una determinada funcionalidad.

Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que estamos construyendo de la misma forma. Por ejemplo, para una empresa que recibe pedidos en forma telefónica, todos los operadores que reciban pedidos y los ingresen en un sistema de ventas, si pueden hacer las mismas cosas con el sistema, son considerados un único actor.

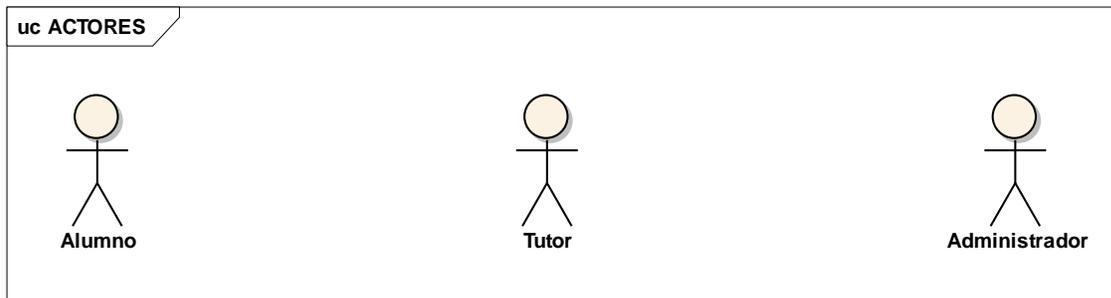
Un escenario es el ambiente en que el actor interactúa con el sistema y se define con un conjunto de pasos o instrucciones para cumplir con el caso de uso.

Un caso de uso se representa:



CASOS DE USO DEL TUTOR

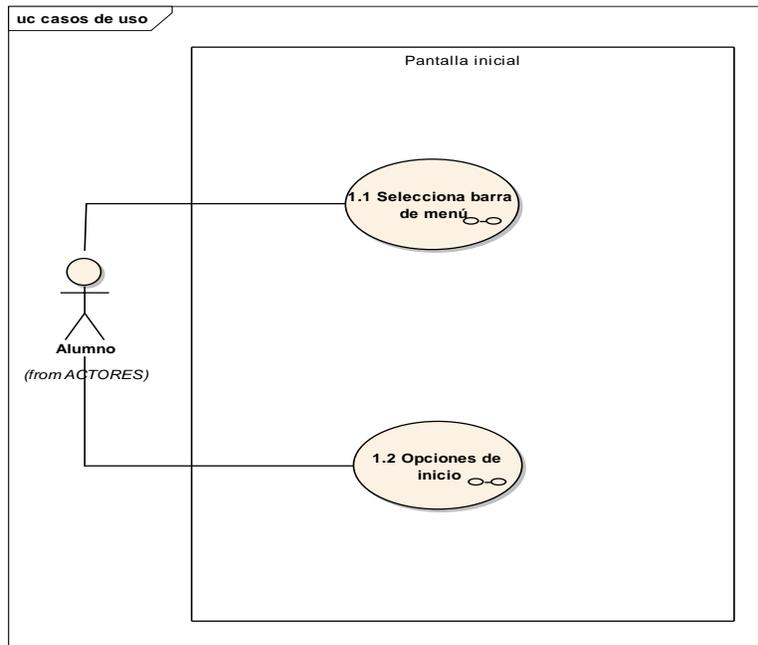
ACTORES



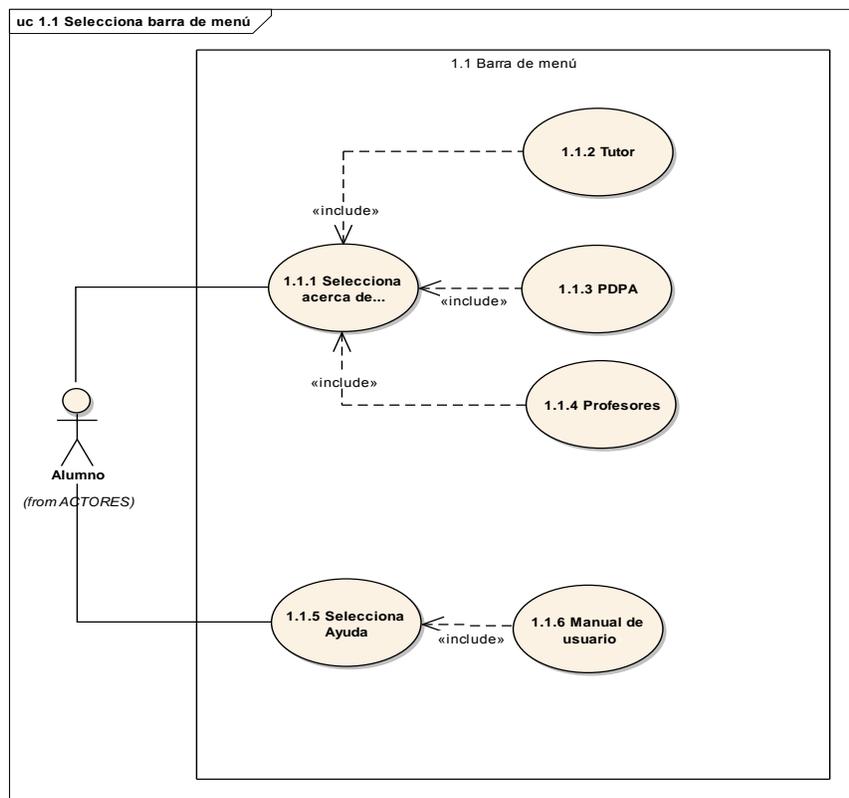
ALUMNO	TUTOR	ADMINISTRADOR
<p>Puede utilizar los recursos que proporciona el tutor.</p> <p>Solicita recursos teóricos dependiendo la selección de módulos y de temas.</p>	<p>Muestra al alumno los recursos en pantalla.</p> <p>Dichos recursos pueden ser de teoría o desarrollo de actividades, que involucran planteamiento de problema y su retroalimentación.</p>	<p>Proporciona al tutor los recursos teóricos que son necesarios para mostrar al alumno en la pantalla de teoría.</p> <p>Proporciona los recursos para el desarrollo de las actividades.</p>

CASOS DE USO

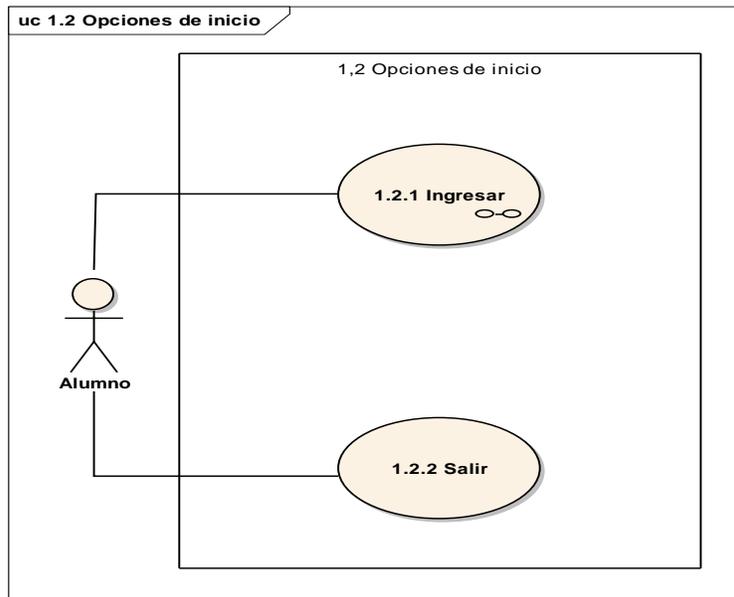
Pantalla Inicial



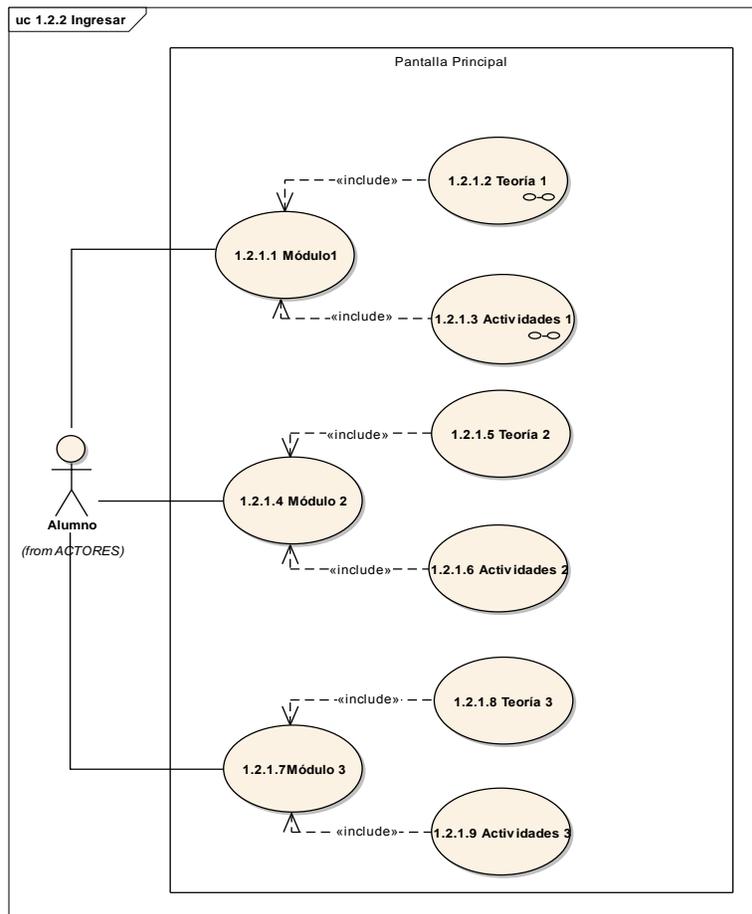
Selecciona barra de menú



Opciones de inicio

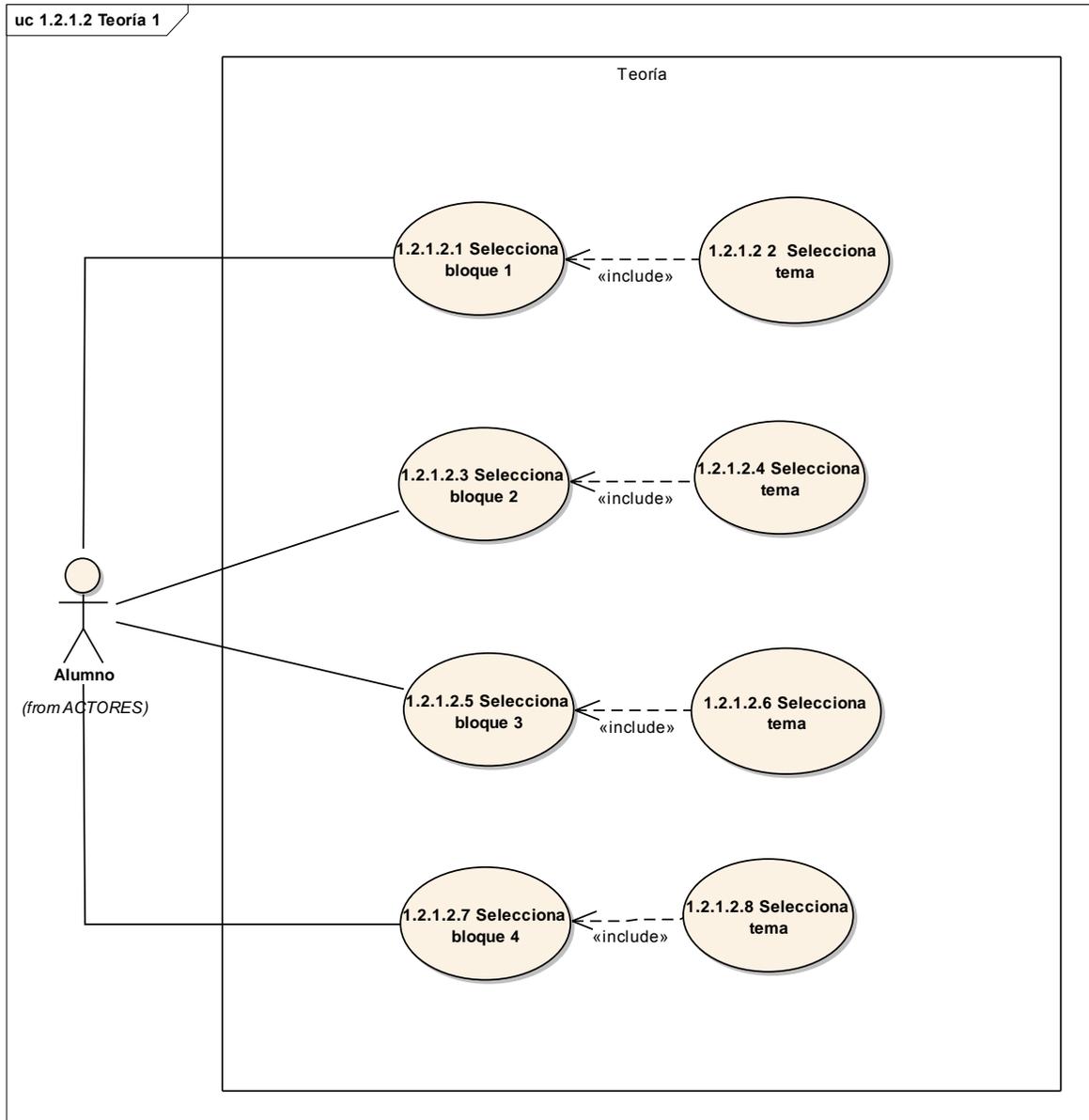


Pantalla Principal

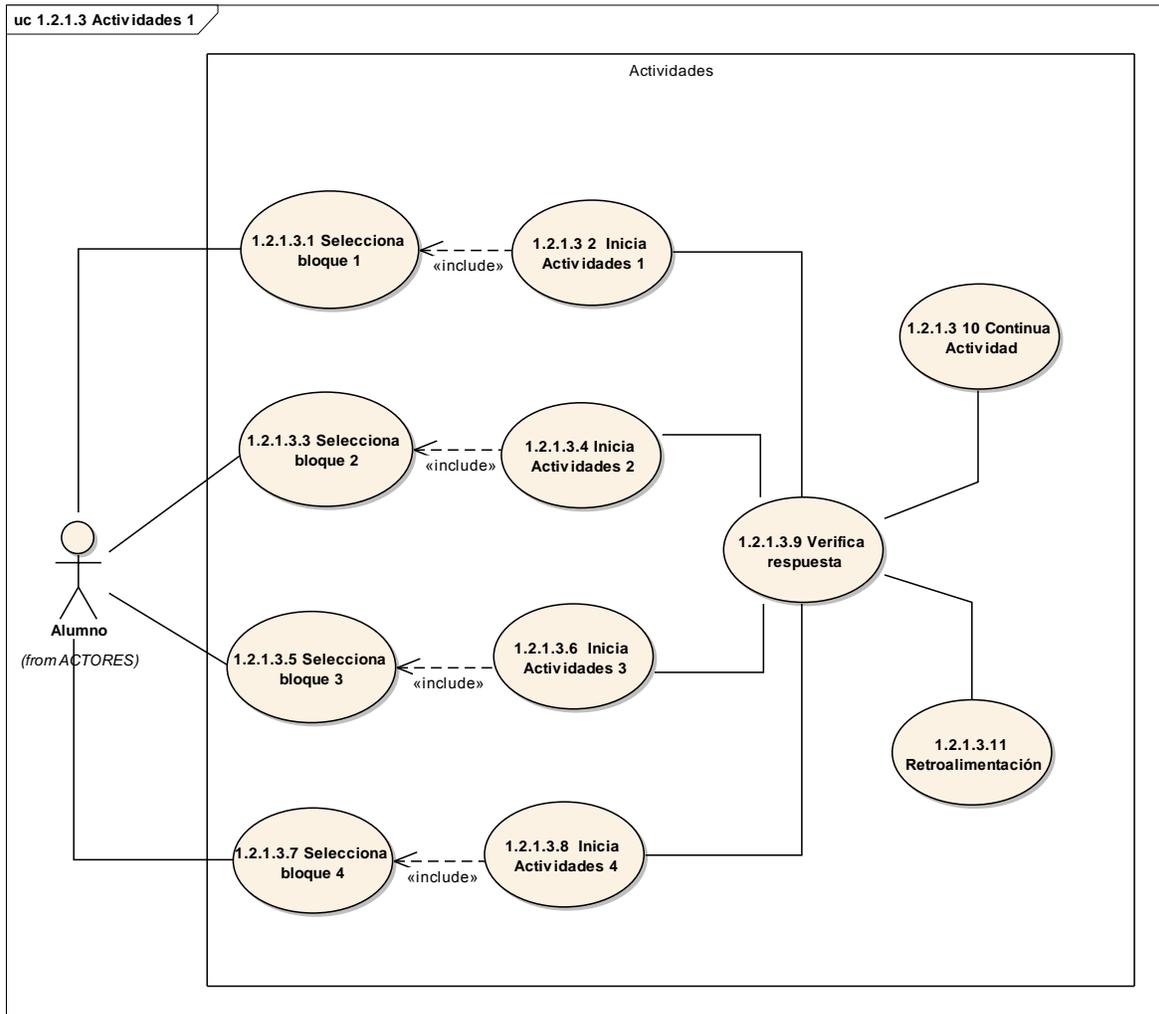


La sección de teoría y actividades se divide de la misma forma en los tres módulos. Se muestra el ejemplo del Módulo 1:

Teoría



Actividades

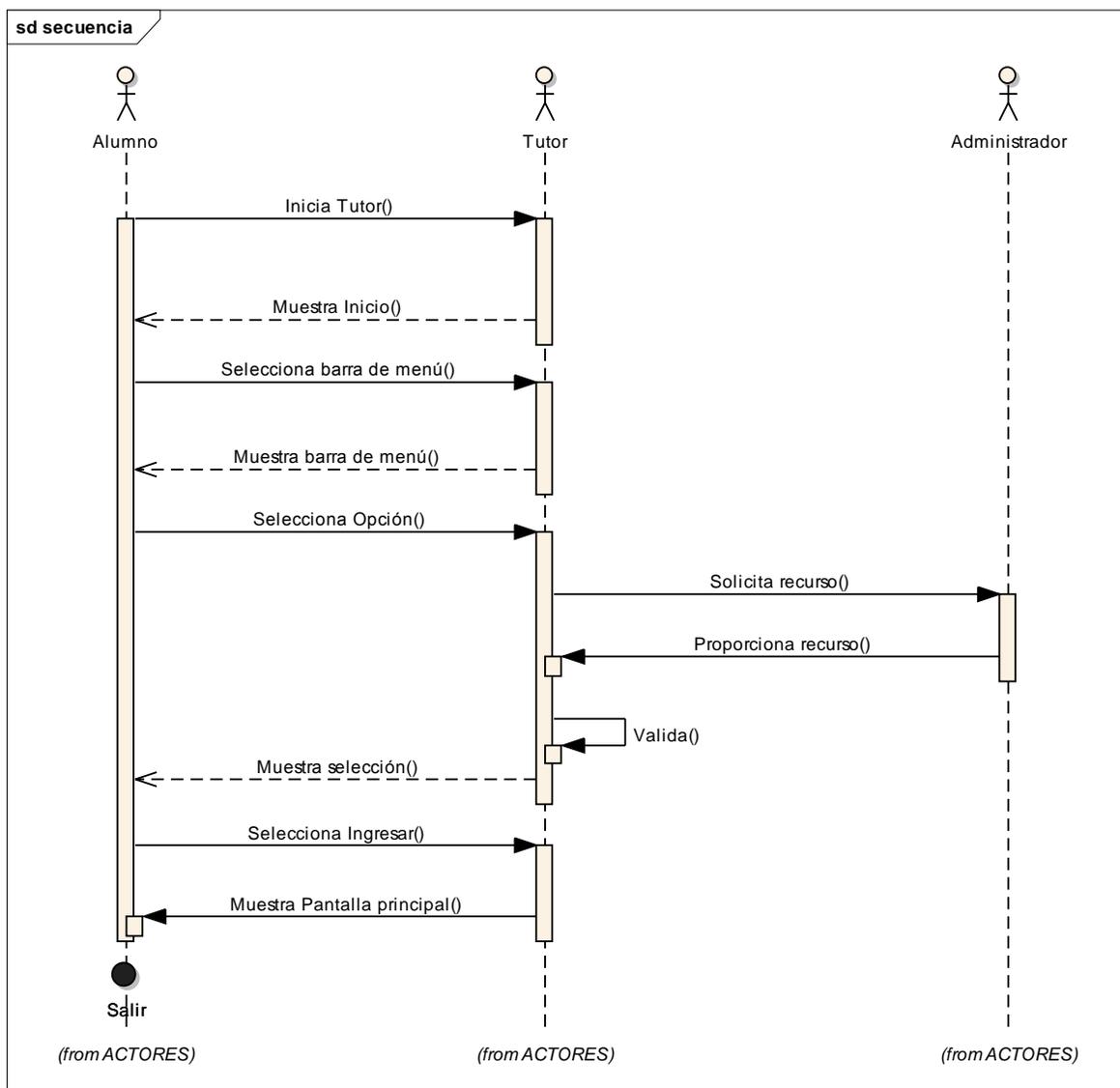


DIAGRAMAS DE SECUENCIA

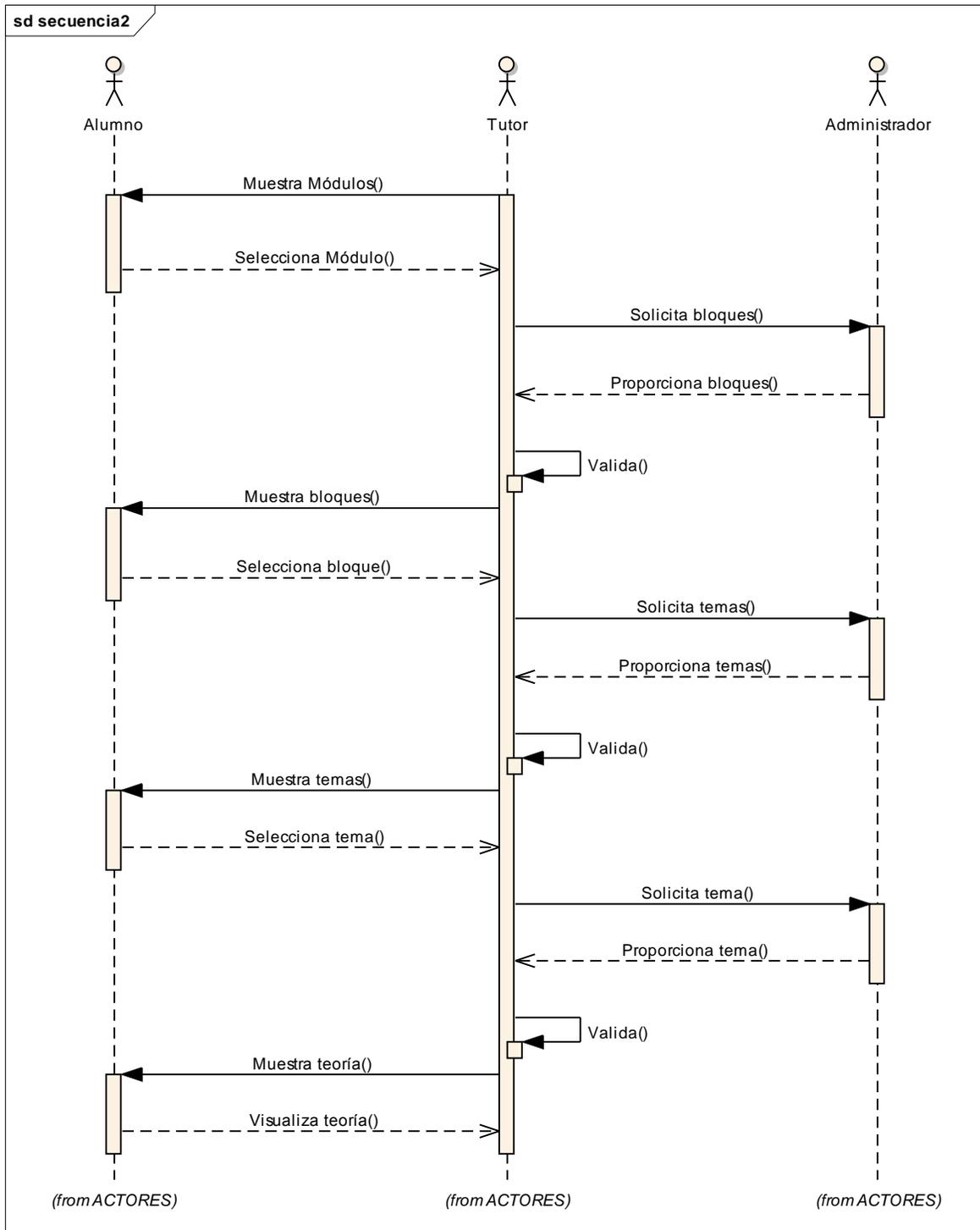
En un diagrama de secuencia se indicarán los módulos o clases que forman parte del programa y las llamadas que se hacen en cada uno de ellos para realizar una tarea determinada. Se realizan diagramas de secuencia para definir acciones que se pueden realizar en la aplicación en cuestión.

Un diagrama de secuencia se utiliza para analizar, documentar o validar la lógica en el sistema que está siendo analizado. Este modelo visual utiliza cuadros y líneas para detallar la secuencia de eventos de un sistema. Varios diagramas de secuencia se pueden utilizar si el proceso es complejo.

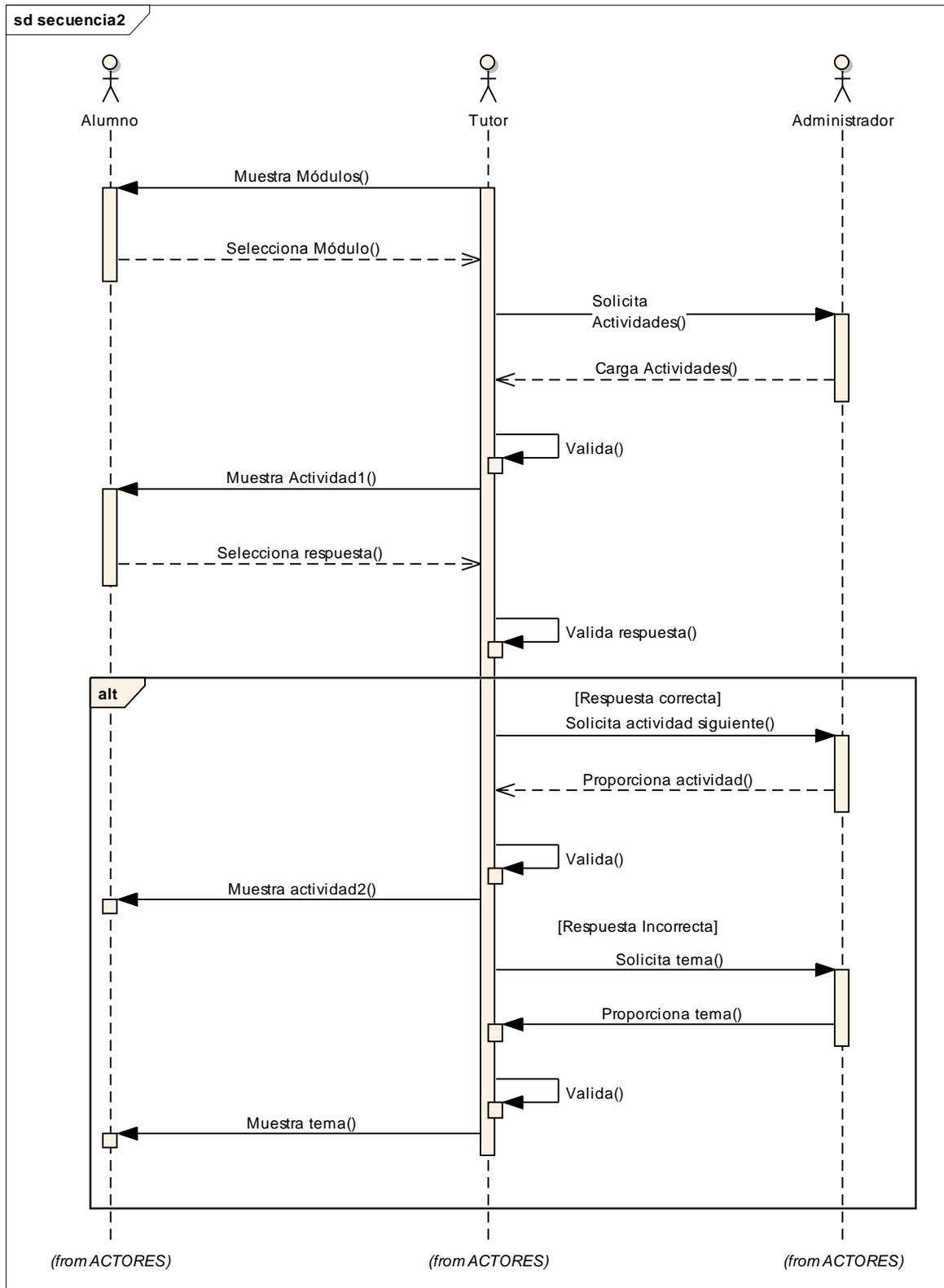
DIAGRAMA DE SECUENCIA DEL TUTOR



Teoría



Actividades



IDENTIFICACIÓN DE REQUERIMIENTOS FUNCIONALES

Gracias a los diagramas UML¹⁵ realizados en el análisis del sistema se pueden identificar los requerimientos funcionales [9]. Los cuales involucran los roles y descripciones de los escenarios de cada uno de los casos de uso identificados anteriormente.

1. ROLES

Son representados por los actores, que pueden ser representados por una persona, un dispositivo de hardware o incluso otro sistema.

Los actores identificados en el proyecto son: alumno, tutor y administrador.

2. ESCENARIOS

Un escenario es una descripción narrativa de cómo las personas hacen las cosas y muestran como tratarían de hacer uso del sistema.

Para describir los escenarios es necesario identificar cada uno de los casos de uso realizados en los diagramas:

Pantalla inicial

CASO DE USO	ESCENARIO
1.1 Selecciona barra de menú	<ol style="list-style-type: none"> 1. Alumno: inicia Tutor Matemático 2. Alumno: identifica barra de menú 3. Alumno: selecciona barra de menú
1.1.1 Selecciona Acerca de...	<ol style="list-style-type: none"> 1. Alumno: selecciona barra de menú 2. Alumno: selecciona Acerca de... 3. Tutor: muestra opciones: Tutor, PDPA y Profesores.
1.1.2 Tutor	<ol style="list-style-type: none"> 1. Alumno: selecciona Acerca de... 2. Alumno: selecciona opción Tutor 3. Tutor: solicita recurso a administrador 4. Administrador: proporciona recurso "Tutor" 5. Tutor: muestra recurso al Alumno

¹⁵ UML: Lenguaje de Modelo Estructurado, es un lenguaje de modelado de sistemas de software.

1.1.3 PDPA	<ol style="list-style-type: none"> 1. Alumno: selecciona Acerca de... 2. Alumno: selecciona opción PDPA 3. Tutor: solicita recurso a administrador 4. Administrador: proporciona recurso "PDPA" 5. Tutor: muestra recurso al Alumno
1.1.4 Profesores	<ol style="list-style-type: none"> 1. Alumno: selecciona Acerca de... 2. Alumno: selecciona opción Profesores 3. Tutor: solicita recurso a administrador 4. Administrador: proporciona recurso "Profesores" 5. Tutor: muestra recurso al Alumno
1.1.5 Selecciona Ayuda	<ol style="list-style-type: none"> 1. Alumno: selecciona barra de menú 2. Alumno: selecciona Ayuda 3. Tutor: muestra opción: Ayuda
1.1.6 Manual de usuario	<ol style="list-style-type: none"> 1. Alumno: selecciona Ayuda 2. Alumno: selecciona opción Manual de Usuario 3. Tutor: solicita recurso a administrador 4. Administrador: proporciona recurso "Manual de usuario" 5. Tutor: muestra recurso al Alumno

Pantalla inicial (Opciones de inicio)

CASO DE USO	ESCENARIO
1.2 Opciones de inicio	<ol style="list-style-type: none"> 1. Alumno: inicia Tutor Matemático 2. Alumno: identifica opciones en pantalla
1.2.1 Ingresar	<ol style="list-style-type: none"> 1. Alumno: inicia Tutor Matemático 2. Alumno: identifica opciones en pantalla 3. Alumno: selecciona Ingresar 4. Tutor: carga Pantalla Principal
1.2.2 Salir	<ol style="list-style-type: none"> 1. Alumno: inicia Tutor Matemático 2. Alumno: identifica opciones en pantalla 3. Alumno: selecciona Salir 4. Tutor: cierra aplicación

Pantalla Principal

CASOS DE USO	DESCRIPCIÓN
Selecciona: 1.2.1.1 Módulo1 1.2.1.4 Módulo 2 1.2.1.7 Módulo 3	1. Alumno: selecciona ingresar 2. Tutor: muestra Pantalla Principal donde muestra tres módulos 3. Tutor: muestra opciones Teoría y Actividades
Selecciona: 1.2.1.2 Teoría 1 1.2.1.5 Teoría 2 1.2.1.8 Teoría 3	1. Tutor: muestra opciones Teoría y Actividades 2. Alumno: selecciona Teoría (Cualquier módulo) 3. Tutor: muestra pantalla de Teoría (Según el módulo seleccionado)
Selecciona: 1.2.1.2 Actividades 1 1.2.1.5 Actividades 2 1.2.1.8 Actividades 3	1. Tutor: muestra opciones Teoría y Actividades 2. Alumno: selecciona Actividades (Cualquier módulo) 3. Tutor: muestra pantalla de Actividades (Según el módulo seleccionado)

Teoría

CASO DE USO	ESCENARIO
Selecciona 1.2.1.2.1 Bloque 1 1.2.1.2.3 Bloque 2 1.2.1.2.5 Bloque 3 1.2.1.2.7 Bloque 4	1. Tutor: muestra pantalla de Teoría (Según el módulo seleccionado) 2. Alumno: selecciona bloque de teoría 3. Tutor: muestra teoría de bloque seleccionado
Selecciona 1.2.1.2.2 Temas 1 1.2.1.2.4 Temas 2 1.2.1.2.6 Temas 3 1.2.1.2.8 Temas 4	1. Alumno: visualiza temas del bloque seleccionado 2. Tutor: solicita tema seleccionado a administrador 3. Tutor: Valida tema 4. Tutor: muestra tema al alumno 3. Alumno: visualiza teoría seleccionada

Actividades

CASO DE USO	ESCENARIO
Selecciona 1.2.1.3.1 Bloque 1 1.2.1.3.3 Bloque 2 1.2.1.3.5 Bloque 3 1.2.1.3.7 Bloque 4	1. Tutor: muestra pantalla de Actividades (Según el módulo seleccionado) 2. Alumno: selecciona bloque de actividades
Inicia 1.2.1.3.2 Actividad 1 1.2.1.3.4 Actividad 2 1.2.1.3.6 Actividad 3 1.2.1.3.8 Actividad 4	1. Alumno selecciona bloque de actividades 2. Tutor: solicita actividad al administrador 3. Tutor: valida Actividad 4. Alumno: visualiza problema 5. Alumno: selecciona respuesta
1.2.1.3.9 Verifica respuesta	1. Alumno: selecciona una respuesta 2. Tutor: recibe respuesta 3. Tutor: verifica respuesta
1.2.1.3.10 Continúa actividad	1. Tutor: identifica respuesta correcta 2. Tutor: solicita siguiente actividad a administrador 3. Tutor: valida actividad 4. Tutor: muestra actividad al alumno
1.2.1.3.11 Retroalimentación	11. Tutor: identifica respuesta incorrecta 2. Tutor: solicita retroalimentación a administrador 3. Tutor: valida retroalimentación 4. Tutor: muestra ayuda al alumno 5. Alumno: selecciona repetir actividad

REQUERIMIENTOS NO FUNCIONALES

Describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema.

Los requerimientos no funcionales identificados son:

TIEMPO DE EJECUCIÓN:

<u>Actividad</u>	<u>Tiempo de ejecución</u>	<u>Equipo</u>
Inicio del tutor	5 segundos aprox.	Equipo 1
	7 segundos aprox.	Equipo 2
Transición entre ventanas	3 segundos aprox.	Equipo 1
	5 segundos aprox.	Equipo 2
Tiempo en cargar teoría	2.5 segundos aprox.	Equipo 1
	3.5 segundos aprox.	Equipo 2
Tiempo en cargar actividades	4 segundos aprox.	Equipo 1
	6.2 segundos aprox.	Equipo 2

Descripción de equipos:

Equipo 1	Equipo 2
Asus Procesador Intel i5, 1.80 GHz, 6 Gb RAM Sistema Operativo Windows 8 64 bits	Sony VAIO Procesador Intel Core 2Duo, 1.6 GHz, 2 Gb RAM Sistema Operativo Windows Vista 64 bits

Recursos consumidos: 120 MB aproximadamente

Lenguaje de implementación: Java

Requerimientos de ejecución:

1. Instalar Java: es gratuito y utilizado para la ejecución de diversos programas en Recursos para diferentes sistemas operativos.

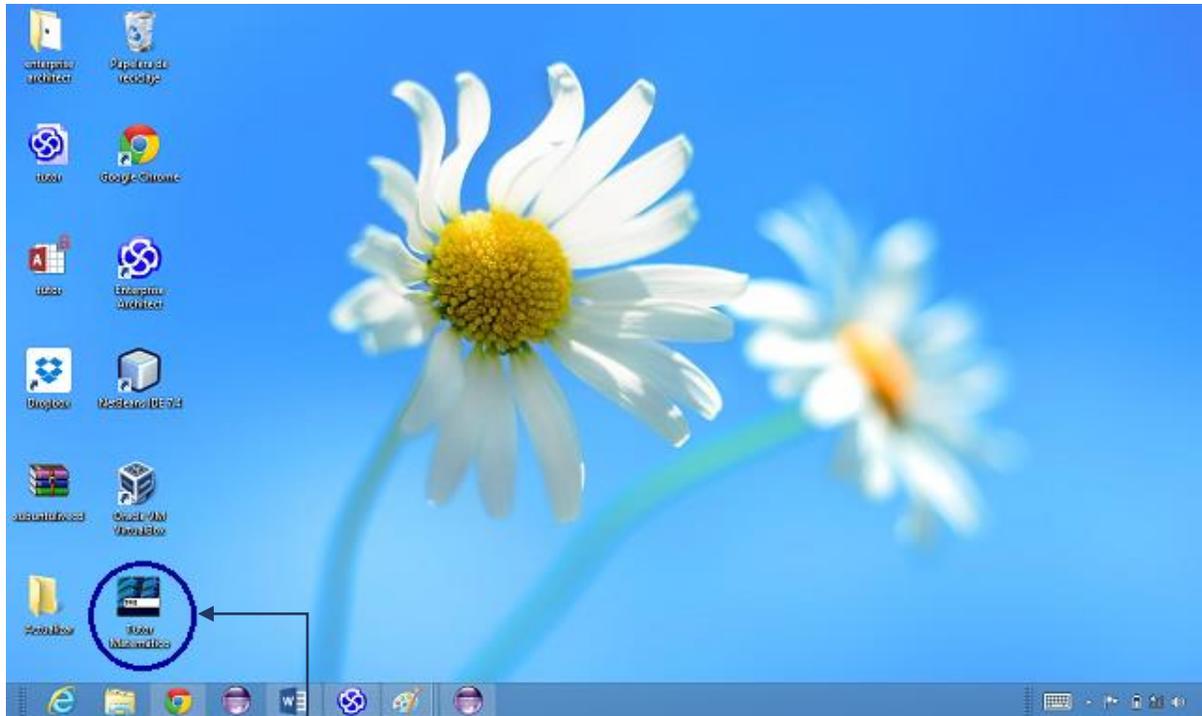
Parte del proceso de instalación de java en algunos sistemas operativos incluyen la asociación de los archivos JAR con el programa interpretador de Java, es decir, que se ejecutará en un ambiente gráfico.

Si el archivo JAR será ejecutado en un ambiente gráfico, con doble clic bastará para abrir el tutor.

CAPITULO 3

MANUAL DE USUARIO

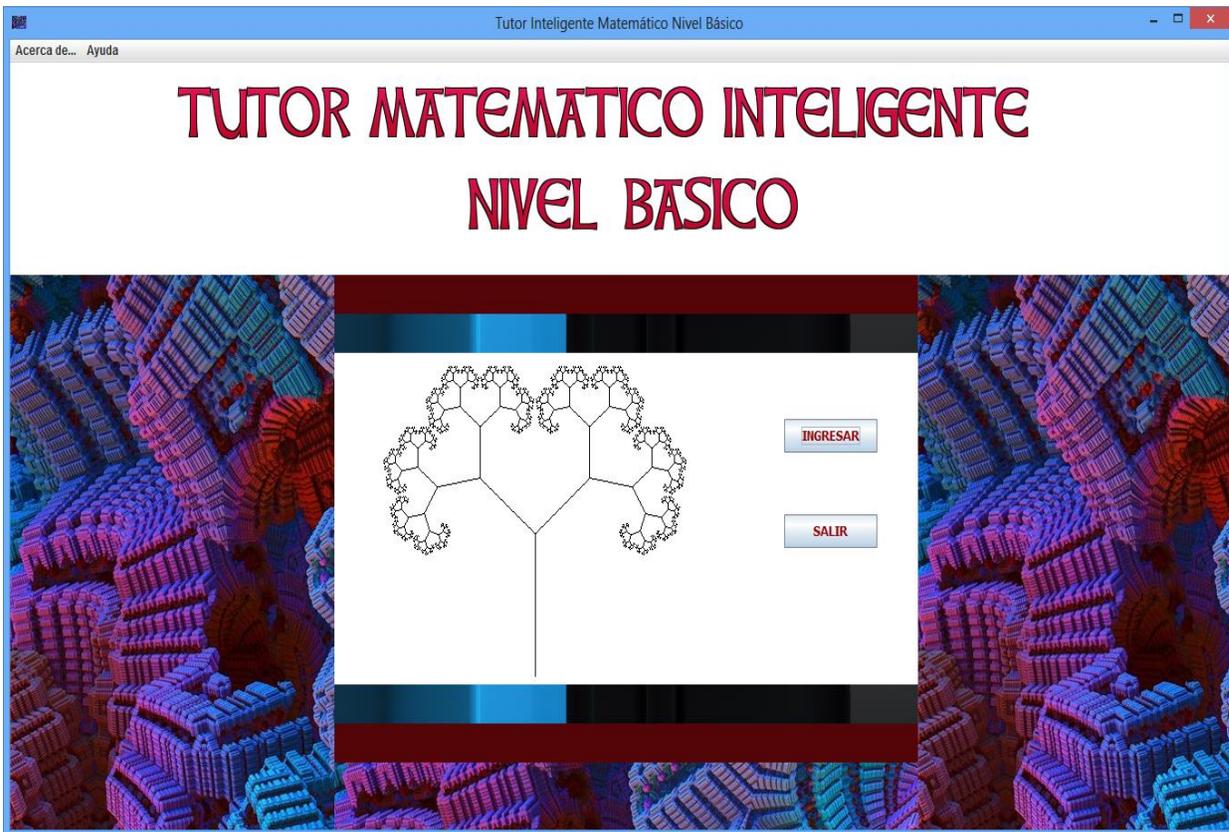
Para iniciar el tutor selecciona el ícono de acceso el cual se identifica por la imagen y el nombre Tutor inteligente.



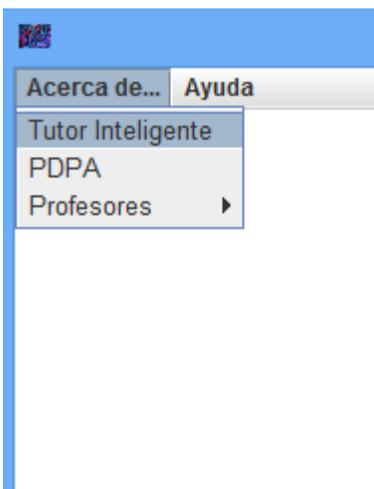
La identificación del tutor será con el ícono mostrado en la imagen y el nombre que se muestra es Tutor Matemático.

Matemático.
por 26.11.2020 a las 12:00

Pantalla inicial:



En la pantalla inicial se encuentran las siguientes opciones:



En la parte superior izquierda se encuentra la barra de menú

Al seleccionar Acerca de... Tutor Inteligente, visualizas la pantalla:

Se muestra una breve descripción de las funciones que realiza un Tutor y su definición

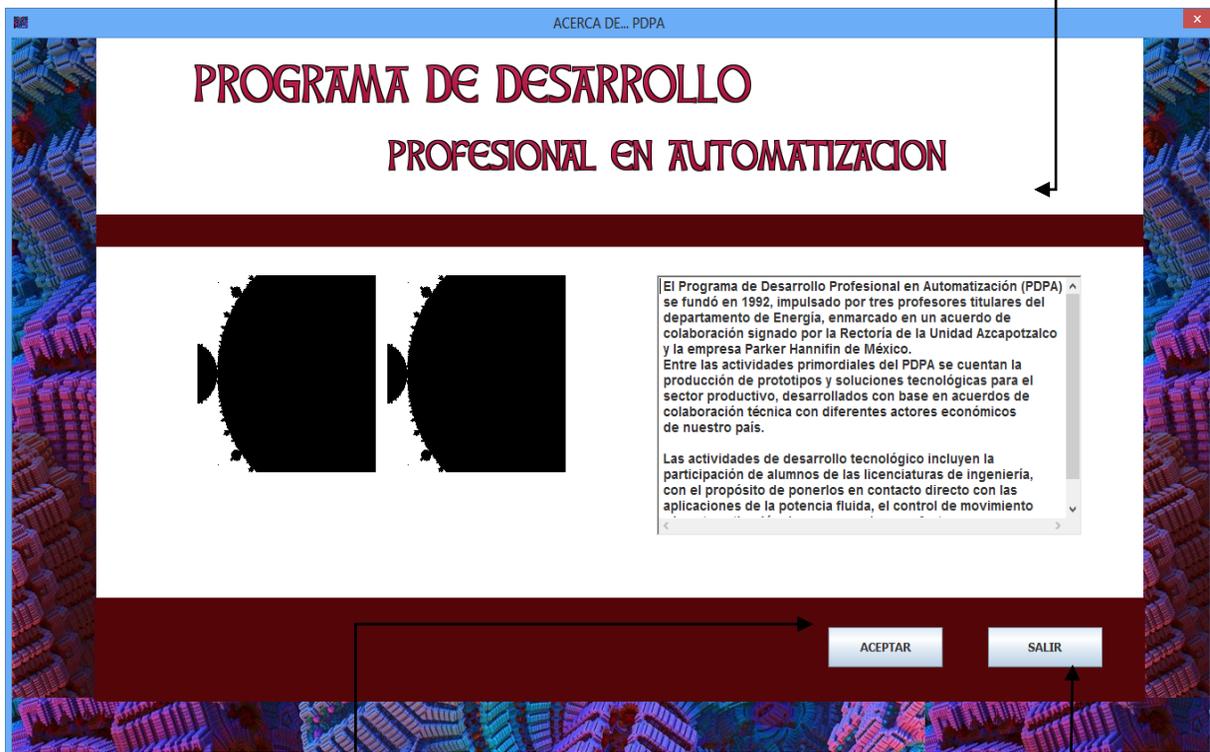


Para regresar a la pantalla inicial, presiona el botón Aceptar, localizado en la parte inferior de la pantalla.

Para salir del tutor presiona el botón Salir, localizado en la parte inferior de la pantalla.

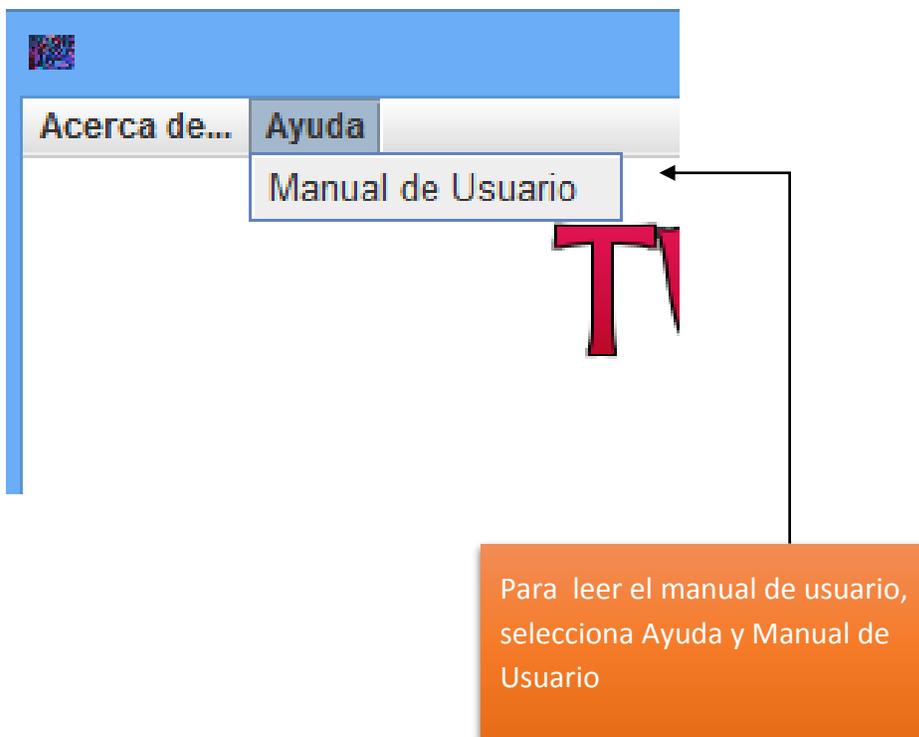
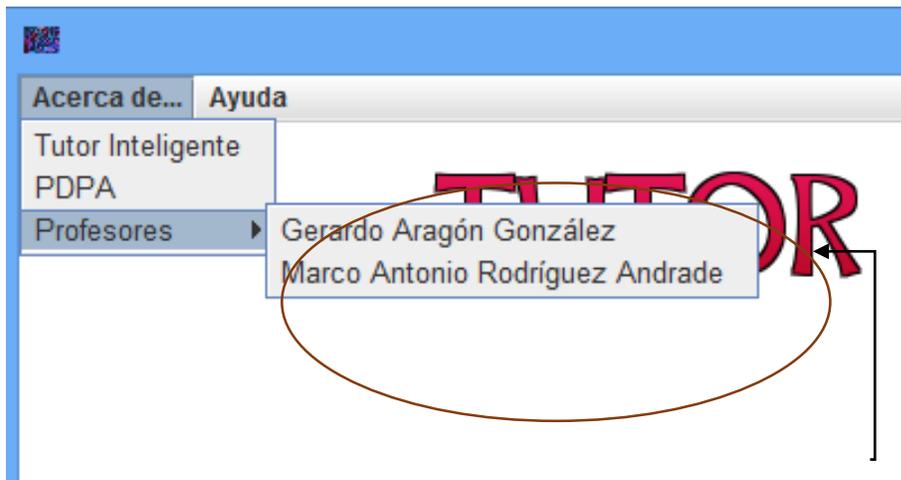
Al seleccionar Acerca de... PDPA, visualizas la pantalla:

Se muestra una descripción del PDPA, donde fue desarrollado el Tutor



Para regresar a la pantalla inicial, presiona el botón Aceptar, localizado en la parte inferior de la pantalla.

Para salir del tutor presiona el botón Salir, localizado en la parte inferior de la pantalla.



Los dos botones restantes de la pantalla inicial son:

	Para abrir la pantalla principal del tutor
---	--

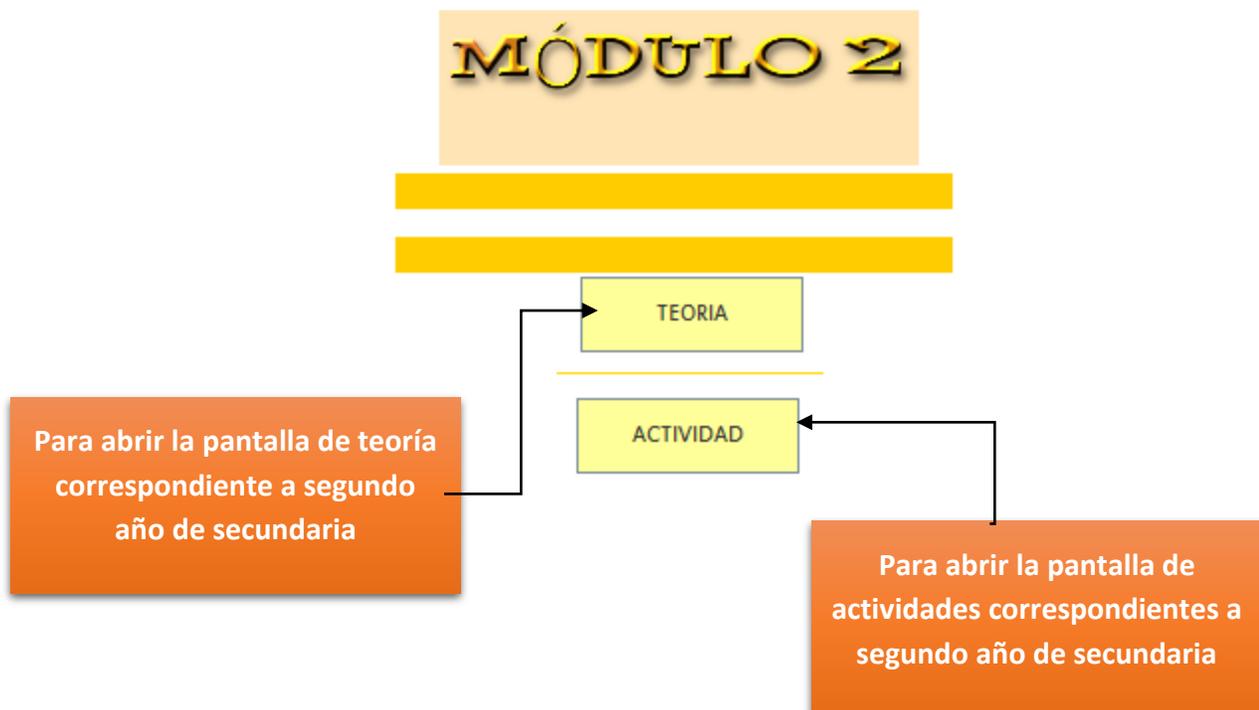
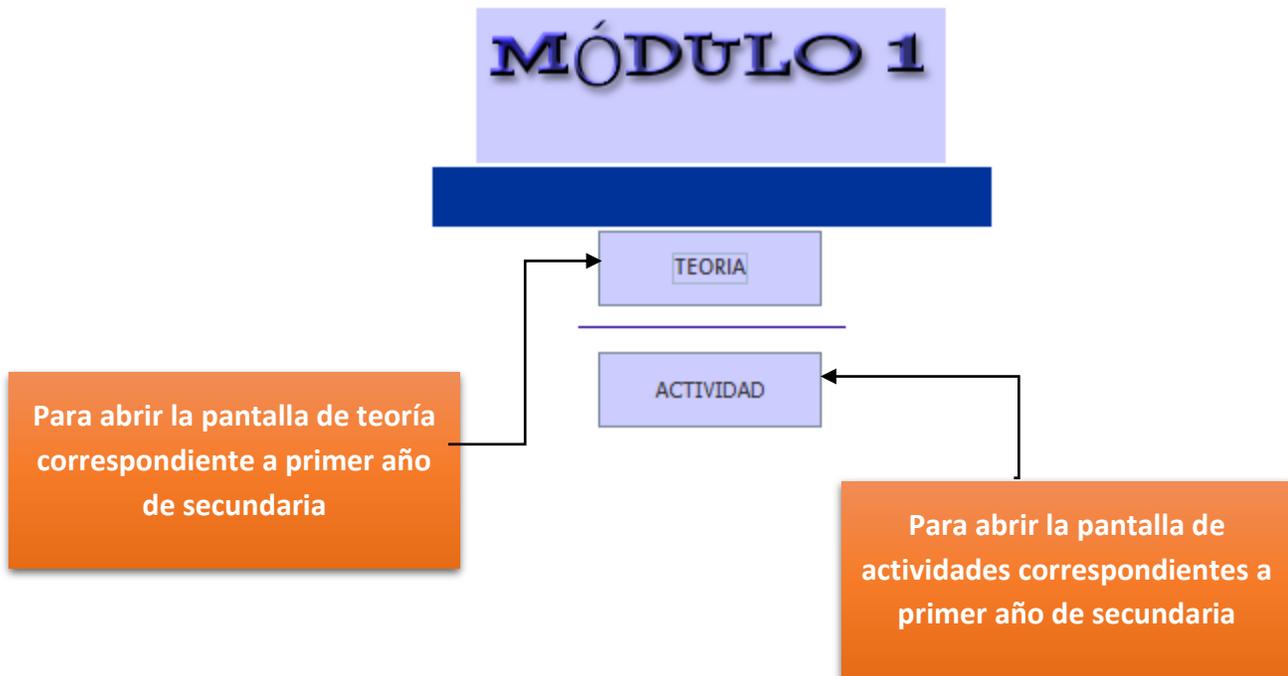
	Para cerrar el tutor
---	----------------------

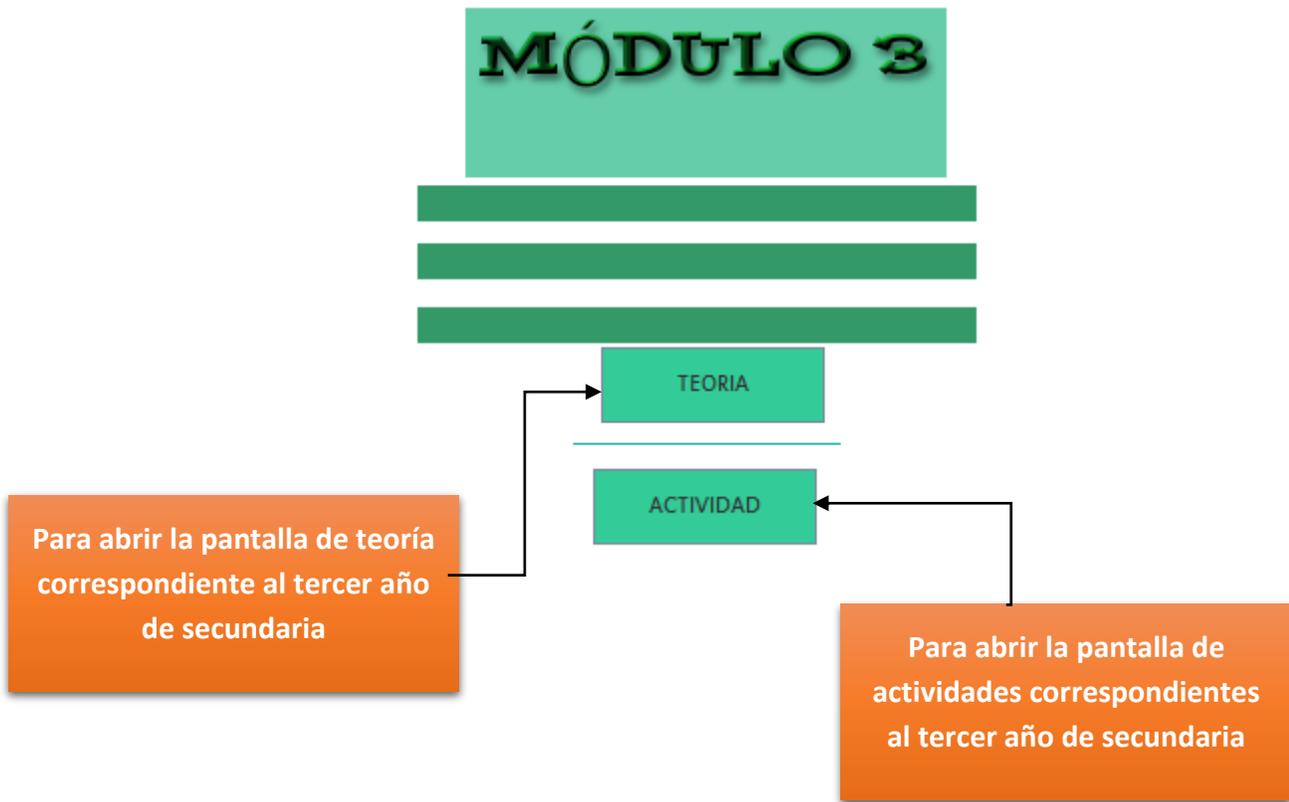
Opciones mostradas en la parte inferior de la pantalla inicial

Pantalla principal:



Se muestran tres diferentes módulos:





Los dos botones restantes de la pantalla principal son:

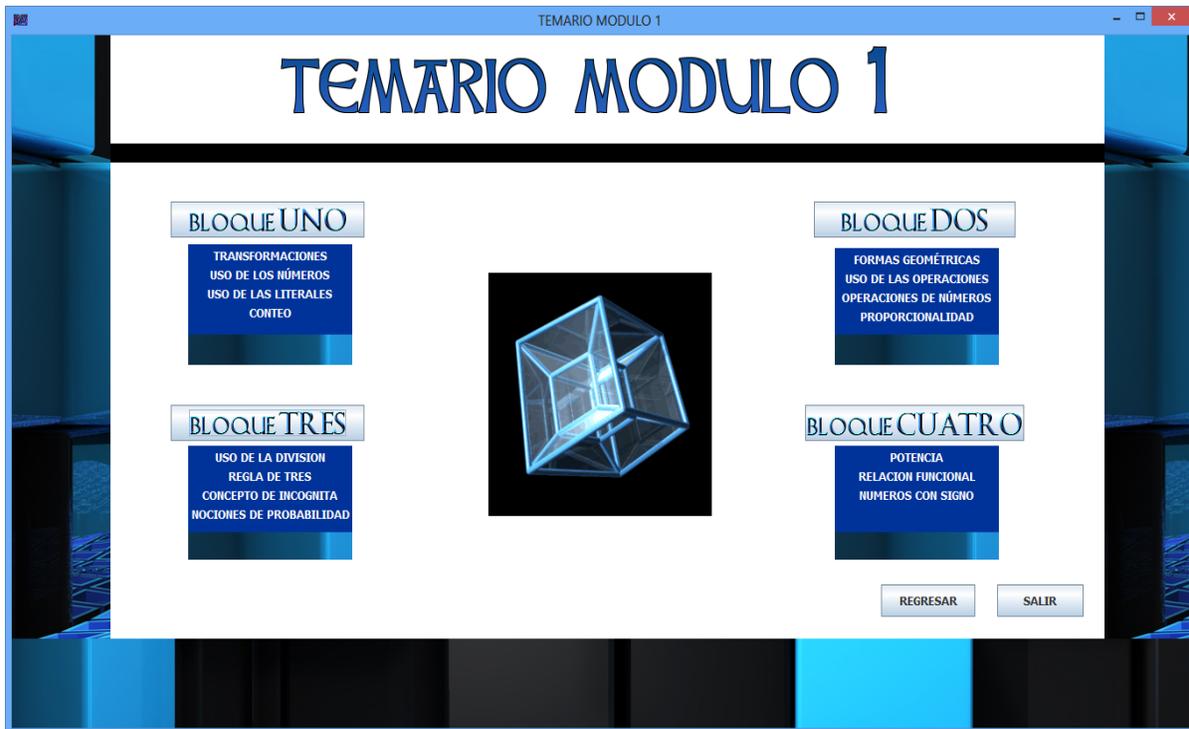
REGRESAR	Para volver a la pantalla inicial del tutor
----------	---

SALIR	Para cerrar el tutor
-------	----------------------

Opciones mostradas en la parte inferior de la pantalla principal

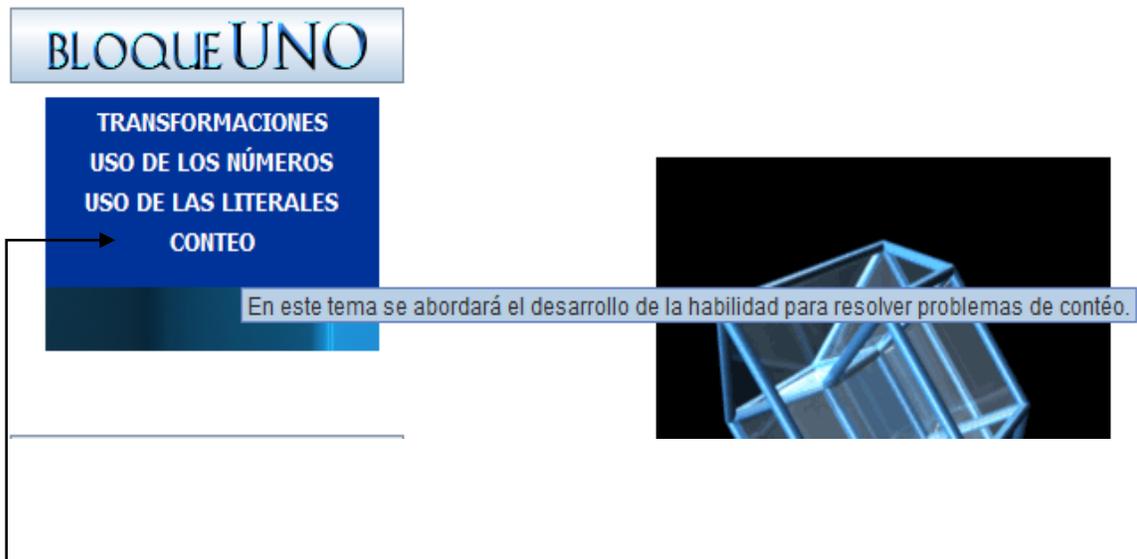
Pantalla de teoría:

(Módulo 1 – Primer año de secundaria)



Descripción del bloque





Para obtener una breve descripción del tema, poner el cursor sobre el título del tema.

Los dos botones restantes de la pantalla de teoría de bloque son:

REGRESAR	Para volver a la pantalla de teoría del tutor
----------	---

SALIR	Para cerrar el tutor
-------	----------------------

Opciones mostradas en la parte inferior de la pantalla de teoría de bloque

Pantalla de teoría de bloque:

The screenshot shows a software interface titled "BLOQUE 1". At the top, there are four navigation tabs: "TRANSFORMACIONES", "USO DE NUMEROS", "USO DE LITERALES", and "CONTEO". The "TRANSFORMACIONES" tab is selected and highlighted in blue. Below the tabs, the content is organized into four panels:

- Top Left Panel:**

Se llama **eje de simetría** a una línea recta con respecto a la cual dos figuras geométricas son simétricas.

Se llama **simetría axial** a la que presentan las figuras en un plano con respecto a una línea recta.

El símbolo \overline{AB} representa al pedazo de línea recta que une el punto A con el punto B, al cual llamamos segmento de recta.
Por ejemplo: Los lados de un triángulo y de un rombo son segmentos de recta.
- Top Right Panel:**

Dos rectas se dicen **perpendiculares** si al cortarse forman cuatro ángulos de igual medida.

Diagram showing two lines intersecting at a right angle, labeled "Perpendicular" and "90°".
- Bottom Left Panel:**

Diagram showing two triangles, ABC and DEF, with a vertical line of symmetry between them.

Para verificar que E es el simétrico de A, no basta con comprobar que la recta que los une es perpendicular al eje de simetría, también hay que medir la longitud del segmento \overline{AE} y comprobar que es igual a la longitud del segmento \overline{OE} .
- Bottom Right Panel:**

Si se mide la longitud de uno de los lados de una figura, su simétrico mide lo mismo; si se mide un ángulo de una figura, su simétrico mide lo mismo; si dos lados de una figura son paralelos, sus simétricos son paralelos; y si dos lados de una figura son perpendiculares, sus simétricos son perpendiculares.

Diagram showing a right-angled triangle and its mirror image across a vertical line.

On the right side of the screen, there are two buttons: "REGRESAR" and "SIGUIENTE".

BLOQUE 1

TRANSFORMACIONES | USO DE NUMEROS | USO DE LITERALES | CONTEO

Cada tema del bloque se puede abrir al seleccionar la pestaña con su nombre

Los dos botones restantes de la pantalla de temas de bloque son:

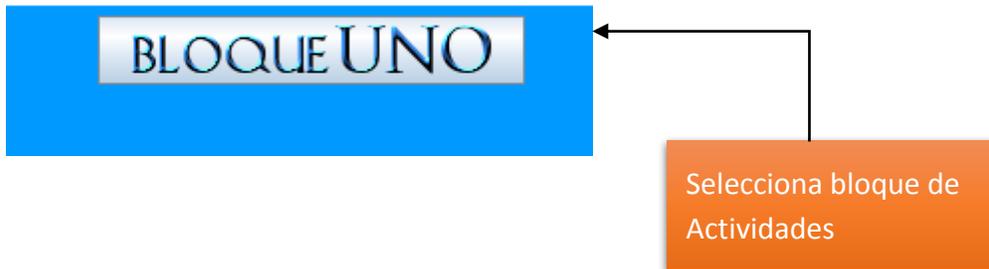
	Para volver a la pantalla de teoría del tutor
---	---

	Para cerrar el tutor
---	----------------------

Opciones mostradas en la parte inferior de la pantalla de teoría

Pantalla de actividades:





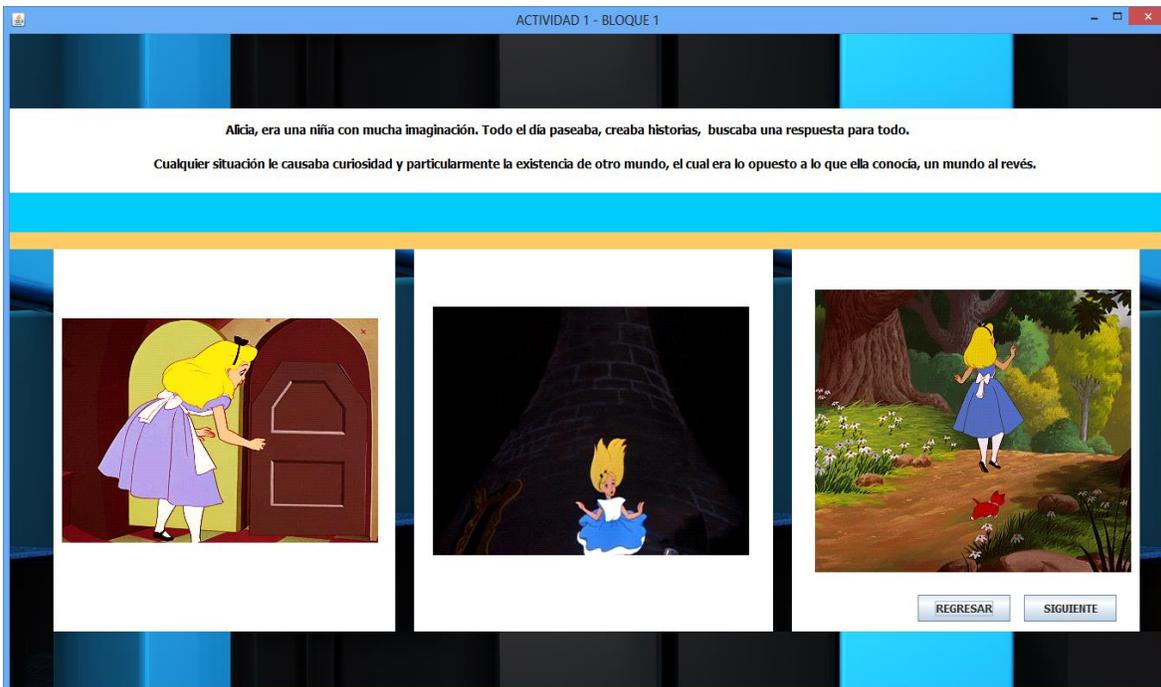
Los dos botones restantes de la pantalla de actividades son:

	Para volver a la pantalla principal del tutor
---	---

	Para cerrar el tutor
---	----------------------

Opciones mostradas en la parte inferior de la pantalla de actividades

Introducción de actividad:



Los dos botones de la pantalla de introducción de actividades son:

	Para volver a la pantalla de actividades del tutor
---	--

	Para cerrar el tutor
---	----------------------

Opciones mostradas en la parte inferior de la pantalla de introducción de actividades

Actividad:

Planteamiento del problema

The screenshot shows a web application window titled "ACTIVIDAD 2 - BLOQUE 1". The interface is divided into several sections:

- Header:** "Pasaba horas viéndose al espejo, imaginando un mundo al revés"
- Problem Statement:** "¿ En donde debe poner Alicia su espejo para que su reflejo sea como en la figura 2?"
- Figure 2:** A central image of Alice in a blue dress, labeled "FIGURA 2".
- Options Panel:** Titled "OPCIONES:", it contains three buttons: "Lado izquierdo", "Lado derecho", and "Lado inferior". Below these are "REGRESAR" and "INICIO" buttons.
- Image:** A small image on the left shows Alice looking into a mirror, with a reflection of her.

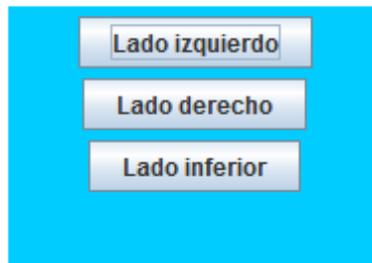
Introducción de problema

Opciones de respuesta

Tipo de respuesta:

Opción múltiple

OPCIONES:



Los dos botones de la pantalla actividad son:

	Para volver a la pantalla de actividades del tutor
---	--

	Para volver al inicio de tutor
---	--------------------------------

Opciones mostradas en la parte inferior de la pantalla de actividad

Si la respuesta del alumno es incorrecta, se visualiza la pantalla de ayuda para solucionar el problema:

AYUDA

TU RESPUESTA ES INCORRECTA

Un ejemplo de como visualizar las imágenes a través de un espejo:

Al colocar un espejo debajo de la letra M, queda de la siguiente manera:

Al colocar un espejo de lado izquierdo de la letra n, queda de la siguiente manera:

REINTENTAR

TEORIA

Permite reintentar la actividad

Visualiza la teoría de la actividad

CONCLUSIONES

Finalizar este Tutor Inteligente enfocado en la educación matemática, permite aportar a nuestro entorno nuevas herramientas de enseñanza-aprendizaje, para el nivel de estudios de educación básica. Como es conocido las matemáticas son base fundamental de la mayoría de las carreras, desde el actuario, economista, químico, ingeniería y más, debido a que son empleadas a diario, como uno de los pilares en el desarrollo escolar, en mayor o menor medida, incluso con diferente enfoque, las matemáticas están presentes y en específico de los estudiantes de ingeniera. No obstante, tutores similares pueden ser útiles para otras fuentes de conocimiento y experiencias para aportar al entorno de novedosas herramientas de aprendizaje, para cualquier nivel de estudios.

Por otra parte, el enfoque e interés fundamental de este proyecto estuvo centrado en una disciplina que sin duda causa problema a cualquier estudiante, sin distinción de grado académico o de la escuela en la que estudian: Matemáticas. Por lo que es indispensable que los estudiantes desde nivel primario, no vean a las Matemáticas como una asignatura aburrida e innecesaria; al contrario, que descubran lo interesante y amplio campo que abarcan, la capacidad de análisis y de razonamiento que puede desarrollar en cada uno y que puede facilitarles la solución de muchos problemas cotidianos. Este proyecto contribuye a ésta problemática.

También intenta contribuir a los problemas que causaba estudiar Matemáticas, por ejemplo, aleja a muchos estudiantes, incluso a nivel superior, de carreras interesantes, que incluso pueden marcar un factor determinante en sus vidas, por el simple hecho de tener esta materia dentro de su plan de estudios.

Existen diversas formas en las que un estudiante de ingeniería puede colaborar en el desarrollo de la educación, más específicamente como estudiante de ingeniería en computación- La realización de este Tutor es una de ellas y fue motivada por el enorme interés y compromiso en la enseñanza-aprendizaje mostrada por los profesores involucrados en la asesoría de este proyecto.

Finalmente, este trabajo puede considerarse como el principio de muchos otros proyectos relacionados con la Educación en Matemáticas, y/o en otras Disciplinas, con el objeto y compromiso de proporcionar y mejorar instrumentos para la enseñanza matemática, que sin duda deja una amplia lista de proyectos por realizar.

APÉNDICE

```

//Pantalla inicial

package Ventanas; //paquete para poder invocar las ventanas de la pantalla

import java.awt.*;

import javax.swing.*;

//paquete de biblioteca java swing

public

class V_INICIO extends JFrame { //comienzo de la clase Inicio de la cual se
extiende el JFrame de swing

private static final long serialVersionUID = 1L;

private JPanel contentPane; // se declara el panel de contenido de la pantalla
Inicio

public static void main(String[] args) //clase principal de Inicio, donde comienza a
ejecutarse el programa

    {

        EventQueue.invokeLater(new Runnable() //para invocar el comienzo de la
clase Inicio

            {

                public void run() {

                    try {

                        V_INICIO Marco = new V_INICIO(); //Se crea el objeto de Inicio

                        Marco.setVisible(true); //Hace visible el objeto

                        GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();

                        grafica.setFullScreenWindow(Marco); //Hace pantalla completa

                    } catch (Exception e) {

```

```

        e.printStackTrace();
    }
}
});}

```

```

public V_INICIO() { //Formato de pantalla

setTitle("Tutor Inteligente Matem\u00E1tico Nivel B\u00E1sico\r\n");//Inserta el
titulo del tutor

setIconImage(Toolkit.getDefaultToolkit().getImage(V_INICIO.class.getResource("/
Img/fondo.jpg")));//Inserta ícono de la aplicacion

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setBounds(100, 100, 1330, 900);//Alineación del título

    contentPane = new JPanel();

    contentPane.setBackground(Color.WHITE); //Color de fondo de la aplicación

    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));//Define borde de la
aplicación

    setContentPane(contentPane);

    contentPane.setLayout(null);

    JMenuBar barra_de_menu = new JMenuBar();//Crea la barra de menú

    barra_de_menu.setBounds(0, 0, 1378, 21);

    contentPane.add(barra_de_menu);//Agrega la barra de menu al panel

        JMenu Acerca_de = new JMenu("Acerca de...");//Título de componente de
barra de Menú

        barra_de_menu.add(Acerca_de);//Agrega el componente a la barra de
Menú

        JMenuItem Tutor_Inteligente = new JMenuItem("Tutor
Inteligente\r\n");//Subtítulo de componente Acerca de... de barra de menú

```

```

Tutor_Inteligente.setFont(new Font("Arial", Font.PLAIN, 13));//Tipo de letra
Tutor_Inteligente.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
V_TUTOR estancia1 = new V_TUTOR();//Crea objeto de la ventana tutor
GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();
grafica.setFullScreenWindow(estancia1);
estancia1.setVisible(true);
estancia1.setLocationRelativeTo(null);
dispose();
;}});
Acerca_de.add(Tutor_Inteligente);
JMenuItem Pdpa = new JMenuItem("PDPA");//Subtítulo de componente Acerca
de... de barra de menú
Pdpa.setFont(new Font("Arial", Font.PLAIN, 13));//Tipo de letra
Pdpa.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
V_PDPA estancia2 = new V_PDPA();//Crea objeto de la ventana PDPA
GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();
grafica.setFullScreenWindow(estancia2);
estancia2.setVisible(true);//Hace visible la ventana PDPA
estancia2.setLocationRelativeTo(null);

```

```

        dispose();}

    });

    Acerca_de.add(Pdpa);

    JMenu mnNewMenu = new JMenu("Profesores");//Subtítulo de componente Acerca
de... de barra de menú

    mnNewMenu.setFont(new Font("Arial", Font.PLAIN, 13));//Tipo de letra

    Acerca_de.add(mnNewMenu);

    JMenuItem GerardoAragnGonzlez = new JMenuItem("Gerardo Arag\u00F3n
Gonz\u00E9lez");//Titulo de nombre de profesor

    GerardoAragnGonzlez.setFont(new Font("Arial", Font.PLAIN, 13));//Tipo de letra

    mnNewMenu.add(GerardoAragnGonzlez);//Agrega titulo a la barra de menu

    JMenuItem MarcoAntonioAndrade = new JMenuItem("Marco Antonio
Andrade");//Titulo de nombre de profesor

    MarcoAntonioAndrade.setFont(new Font("Arial", Font.PLAIN, 13));//Tipo de letra

    mnNewMenu.add(MarcoAntonioAndrade);

    JMenu Ayuda = new JMenu("Ayuda\r\n");//Titulo de componente de barra de
menu

    barra_de_menu.add(Ayuda);

    JMenuItem mntmManualDeUsuario = new JMenuItem("Manual de
Usuario\r\n");//Subtitulo de componente Ayuda

    mntmManualDeUsuario.setFont(new Font("Arial", Font.PLAIN, 13));//Tipo de letra

    Ayuda.add(mntmManualDeUsuario);

    JLabel imagen = new JLabel("");//Etiqueta contenedora de imagen

        imagen.setIcon(new
ImageIcon(V_INICIO.class.getResource("/Img/secuence.GIF")));//Inserta imagen

    imagen.setBounds(381, 289, 471, 306);//Define tamaño de etiqueta

    contentPane.add(imagen);

```

```
JLabel imagen1 = new JLabel(""); //Etiqueta contenedora de imagen

imagen1.setIcon(new
ImageIcon(V_INICIO.class.getResource("/Img/fondo.jpg"))); //Inserta imagen

imagen1.setBounds(0, 217, 359, 565); //Define tamaño de
etiquetacontentPane.add(imagen1);

JLabel imagen2 = new JLabel(""); //Etiqueta contenedora de imagen

        imagen2.setIcon(new
ImageIcon(V_INICIO.class.getResource("/Img/fondo.jpg")));

        imagen2.setBounds(1005, 217, 353, 565); //Define tamaño de etiqueta
contentPane.add(imagen2);

JLabel titulo = new JLabel("");

titulo.setIcon(new ImageIcon(V_INICIO.class.getResource("/Img/titulo_tutor
matematico inteligente.png")));

titulo.setBounds(185, 32, 957, 71);

contentPane.add(titulo);

JLabel titulo1 = new JLabel("");

titulo1.setIcon(new ImageIcon(V_INICIO.class.getResource("/Img/titulo_nivel
basico.png")));

titulo1.setBounds(507, 125, 383, 53);

contentPane.add(titulo1);

JLabel fondo = new JLabel("");

fondo.setIcon(new ImageIcon(V_INICIO.class.getResource("/Img/barra.jpg")));

fondo.setBackground(new Color(128, 0, 0));

fondo.setBounds(358, 217, 652, 36);

contentPane.add(fondo);
```

```
JLabel fondo1 = new JLabel("");
fondo1.setIcon(new ImageIcon(V_INICIO.class.getResource("/Img/fondo.jpg")));
fondo1.setBackground(new Color(128, 0, 0));
fondo1.setBounds(358, 667, 652, 82);
        contentPane.add(fondo1);

JButton Ingresar = new JButton("INGRESAR");
Ingresar.setForeground(new Color(153, 0, 0));
Ingresar.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent arg0) {
        V_MENU ingresa= new V_MENU();
                GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();
        grafica.setFullScreenWindow(ingresa);
        ingresa.setVisible(true);
        ingresa.setLocationRelativeTo(null);
                dispose();
        }
});

btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 12));
btnNewButton.setBounds(857, 350, 103, 31);
contentPane.add(btnNewButton);
JButton Salir = new JButton ("SALIR\r\n");
```

```
Salir.setForeground(newColor(153, 0, 0));
Salir.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});
Salir.setFont(new Font("Tahoma", Font.BOLD, 12));
Salir.setBounds(857, 438, 103, 31);
contentPane.add(Salir);

JLabel fondo3 = new JLabel("");
fondo3.setIcon(new ImageIcon(V_INICIO.class.getResource("/Img/barra2.jpg")));
fondo3.setBackground(new Color(128, 0, 0));
fondo3.setBounds(358, 253, 652, 36);
contentPane.add(fondo3);

JLabel fondo4 = new JLabel("");
fondo4.setIcon(new ImageIcon(V_INICIO.class.getResource("/Img/barra.jpg")));
fondo4.setBackground(new Color(128, 0, 0));
fondo4.setBounds(358, 631, 652, 36);
contentPane.add(fondo4);

JLabel fondo5 = new JLabel("");
fondo5.setIcon(new ImageIcon(V_INICIO.class.getResource("/Img/barra2.jpg")));
fondo5.setBackground(new Color(128, 0, 0));
fondo5.setBounds(358, 595, 652, 36);
```

```
contentPane.add(fondo5);}}  
  
//Principal  
  
package Ventanas; //paquete para poder invocar las ventanas de la pantalla  
  
import java.awt.*;  
import javax.swing.*;  
  
//paquete de biblioteca java swing  
  
public class V_MENU extends JFrame {  
  
    private JPanel contentPane;  
  
    public static void main(String[] args) {  
        EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                try {  
                    V_MENU marco = new V_MENU();  
                    marco.setVisible(true);  
  
                    GraphicsDevice  
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev  
ice();  
  
                    grafica.setFullScreenWindow(marco);  
  
                } catch (Exception e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
    }  
}
```

```
        }  
    });  
}  
  
public V_MENU() {  
  
    setIconImage(Toolkit.getDefaultToolkit().getImage(V_MENU.class.getResource("/Img/fondo.jpg")));  
  
    setTitle("MEN\u00DA DE M\u00D3DULOS");  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setBounds(100, 100, 1334, 835);  
    contentPane = new JPanel();  
    contentPane.setBackground(new Color(255, 255, 255));  
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  
    setContentPane(contentPane);  
    contentPane.setLayout(null);  
  
    JPanel panel_1 = new JPanel();  
    panel_1.setBackground(new Color(255, 228, 181));  
    panel_1.setBounds(534, 326, 241, 78);  
    contentPane.add(panel_1);  
  
    JLabel imagen = new JLabel("");  
    imagen.setIcon(new  
    ImageIcon(V_MENU.class.getResource("/Img/modulo2.png")));  
    panel_1.add(imagen);
```

```
JPanel fondo = new JPanel();
fondo.setBackground(new Color(102, 205, 170));
fondo.setBounds(858, 290, 241, 85);
contentPane.add(fondo);

JLabel imagen1= new JLabel("");
imagen1.setIcon(new
ImageIcon(V_MENU.class.getResource("/Img/modulo3.png")));
imagen1.add(imagen1);

JLabel imagen2 = new JLabel("");
panel.add(imagen2);
imagen2.setIcon(new
ImageIcon(V_MENU.class.getResource("/Img/modulo1.png")));

JLabel espacio = new JLabel("");
panel_2.add(espacio);

JLabel espacio1 = new JLabel("");
panel_2.add(espacio1);

JLabel espacio2 = new JLabel("");
panel_2.add(espacio2);

JLabel espacio3 = new JLabel("");
```

```
panel_3.add(espacio3);  
JPanel panel = new JPanel();  
panel.setBackground(new Color(204, 204, 255));  
panel.setBounds(228, 348, 235, 78);  
contentPane.add(panel);
```

```
JSeparator separador1 = new JSeparator();  
separador1.setForeground(new Color(255, 215, 0));  
separador1.setBounds(593, 508, 134, 38);  
contentPane.add(separador1);
```

```
JSeparator separador2 = new JSeparator();  
separador2.setForeground(new Color(32, 178, 170));  
separador2.setBounds(912, 508, 134, 38);  
contentPane.add(separador2);
```

```
JSeparator separador = new JSeparator();  
separador.setForeground(new Color(51, 0, 153));  
separador.setBounds(279, 508, 134, 38);  
contentPane.add(separador);
```

```
JPanel panel_3 = new JPanel();  
panel_3.setBackground(new Color(0, 51, 153));  
panel_3.setBounds(206, 428, 280, 30);
```

```

contentPane.add(panel_3);

JButton btnTemario = new JButton("ACTIVIDAD");
btnTemario.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        V_ACTIVIDADES1 llamar= new V_ACTIVIDADES1();
        llamar.setVisible(true);
        llamar.setLocationRelativeTo(null);
        GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();

        grafica.setFullScreenWindow(llamar);
        dispose();
    }
});
btnTemario.setBounds(289, 521, 112, 38);
contentPane.add(btnTemario);

btnTemario.setToolTipText("VISUALIZA TEMARIO DE PRIMER
A\u00D1O DE SECUNDARIA");

btnTemario.setFont(new Font("Tahoma", Font.PLAIN, 11));
btnTemario.setBackground(new Color(204, 204, 255));

JButton btnActividad = new JButton("TEORIA");
btnActividad.setBounds(289, 460, 112, 38);
contentPane.add(btnActividad);

```

```
12));  
  
btnActividad.setFont(new Font("Estrangelo Edessa", Font.PLAIN,  
  
btnActividad.setBackground(new Color(204, 204, 255));  
  
JLabel imagen3= new JLabel("");  
  
imagen3.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/fondo.jpg")));  
  
imagen3.setBounds(169, 694, 1042, 48);  
  
contentPane.add(imagen3);  
  
  
JPanel panel_4 = new JPanel();  
  
panel_4.setBackground(new Color(255, 204, 0));  
  
panel_4.setBounds(512, 440, 280, 18);  
  
contentPane.add(panel_4);  
  
  
JLabel espacio3 = new JLabel("");  
  
panel_4.add(espacio3);  
  
  
JPanel panel_5 = new JPanel();  
  
panel_5.setBackground(new Color(255, 204, 0));  
  
panel_5.setBounds(512, 408, 280, 18);  
  
contentPane.add(panel_5);  
  
  
JLabel espacio4= new JLabel("");  
  
panel_5.add(espacio4);
```

```

JButton btnTeoria = new JButton("TEORIA\r\n");
btnTeoria.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
    }
});
btnTeoria.setFont(new Font("Estrangelo Edessa", Font.PLAIN, 12));
btnTeoria.setBackground(new Color(255, 255, 153));
btnTeoria.setBounds(605, 460, 112, 38);
contentPane.add(btnTeoria);

JButton button = new JButton("ACTIVIDAD");
button.setToolTipText("VISUALIZA TEMARIO DE PRIMER A\u00D1O
DE SECUNDARIA");
button.setFont(new Font("Estrangelo Edessa", Font.PLAIN, 12));
button.setBackground(new Color(255, 255, 153));
button.setBounds(603, 521, 112, 38);
contentPane.add(button);

JPanel panel_6 = new JPanel();
panel_6.setBackground(new Color(51, 153, 102));
panel_6.setBounds(834, 379, 280, 18);
contentPane.add(panel_6);

JLabel espacio5 = new JLabel("");
panel_6.add(espacio5);

```

```
JPanel panel_7 = new JPanel();  
panel_7.setBackground(new Color(51, 153, 102));  
panel_7.setBounds(834, 408, 280, 18);  
contentPane.add(panel_7);
```

```
JLabel espacio6 = new JLabel("");  
panel_7.add(espacio6);
```

```
JPanel panel_8 = new JPanel();  
panel_8.setBackground(new Color(51, 153, 102));  
panel_8.setBounds(834, 440, 280, 18);  
contentPane.add(panel_8);
```

```
JLabel espacio7 = new JLabel("");  
panel_8.add(espacio7);
```

```
JButton btnTeoria_1 = new JButton("TEORIA");  
btnTeoria_1.setFont(new Font("Estrangelo Edessa", Font.PLAIN,  
12));
```

```
btnTeoria_1.setBackground(new Color(51, 204, 153));  
btnTeoria_1.setBounds(926, 460, 112, 38);  
contentPane.add(btnTeoria_1);
```

```
JButton button_2 = new JButton("ACTIVIDAD");
```

```
        button_2.setToolTipText("VISUALIZA TEMARIO DE PRIMER  
A\u00D1O DE SECUNDARIA");  
  
        button_2.setFont(new Font("Estrangelo Edessa", Font.PLAIN, 12));  
        button_2.setBackground(new Color(51, 204, 153));  
        button_2.setBounds(922, 521, 112, 38);  
        contentPane.add(button_2);  
  
        JLabel imagen4 = new JLabel("");  
        imagen4.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/titulo_seleccion.png")));  
        imagen4.setBounds(417, 24, 516, 78);  
        contentPane.add(imagen4);  
  
        JLabel imagen5 = new JLabel("");  
        imagen5.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/titulo_modulos de tutor.png")));  
        imagen5.setBounds(537, 113, 604, 60);  
        contentPane.add(imagen5);  
  
        JLabel imagen6 = new JLabel("");  
        imagen6.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/fondo.jpg")));  
        imagen6.setBounds(0, 196, 196, 546);  
        contentPane.add(imagen6);  
  
        JLabel imagen7 = new JLabel("");
```

```
    imagen7.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/menu_fondo.gif")));
```

```
    imagen7.setBounds(-14, -87, 335, 279);
```

```
    contentPane.add(imagen7);
```

```
    JLabel imagen8 = new JLabel("");
```

```
    imagen8.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/fondo.jpg")));
```

```
    imagen8.setBounds(1140, 196, 226, 546);
```

```
    contentPane.add(imagen7);
```

```
    JLabel imagen9 = new JLabel("");
```

```
    imagen9.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/barra.jpg")));
```

```
    imagen9.setBounds(190, 196, 951, 30);
```

```
    contentPane.add(imagen9);
```

```
    JLabel imagen10 = new JLabel("");
```

```
    imagen10.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/barra.jpg")));
```

```
    imagen10.setBounds(190, 664, 951, 30);
```

```
    contentPane.add(imagen10);
```

```
    JLabel imagen11 = new JLabel("");
```

```
    imagen11.setIcon(new  
ImageIcon(V_MENU.class.getResource("/Img/barra2.jpg")));
```

```
imagen11.setBounds(190, 634, 951, 30);
contentPane.add(imagen11);

JButton btnRegresar = new JButton("REGRESAR");
btnRegresar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        V_INICIO llamar= new V_INICIO();
        llamar.setVisible(true);
        GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();

        grafica.setFullScreenWindow(llamar);
        llamar.setLocationRelativeTo(null);
        dispose();
    }
});

btnRegresar.setFont(new Font("Tahoma", Font.BOLD, 12));
btnRegresar.setBounds(847, 599, 104, 30);
contentPane.add(btnRegresar);

JButton btnSalir = new JButton("SALIR\r\n");
btnSalir.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
}
```

```

});

btnSalir.setFont(new Font("Tahoma", Font.BOLD, 12));

btnSalir.setBounds(995, 599, 104, 30);

contentPane.add(btnSalir);

JLabel label_20 = new JLabel("");

label_20.setIcon(new
ImageIcon(V_MENU.class.getResource("/Img/barra2.jpg")));

label_20.setBounds(187, 225, 957, 30);

contentPane.add(label_20);

btnActividad.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        V_TEMARIO1 llamar= new V_TEMARIO1();

        llamar.setVisible(true);

        llamar.setLocationRelativeTo(null);

        GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();

        grafica.setFullScreenWindow(llamar);

        dispose();

    }

});

}}

//Actividades

```

```

package Ventanas; //paquete para poder invocar las ventanas de la pantalla

import java.awt.*;

import javax.swing.*;

//paquete de biblioteca java swing

public class V_ACTIVIDADES1 extends JFrame {

    private JPanel contentPane;

    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable() {

            public void run() {

                try {

                    V_ACTIVIDADES1 frame = new V_ACTIVIDADES1 ();

                    frame.setVisible(true);

                    GraphicsDevice
                    grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
                    ice();

                                grafica.setFullScreenWindow(frame);

                                } catch (Exception e) {

                                    e.printStackTrace();

                                }

                            }

                    });

    }

    public V_ACTIVIDADES1 () {

```

```
setIconImage(Toolkit.getDefaultToolkit().getImage(V_TEMARIO1.class.getResource("/Img/fondo.jpg")));
```

```
setTitle("ACTIVIDADES MODULO 1\r\n");  
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
setBounds(100, 100, 1414, 731);  
contentPane = new JPanel();  
contentPane.setBackground(Color.WHITE);  
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  
setContentPane(contentPane);  
contentPane.setLayout(null);
```

```
JLabel fondo = new JLabel("");  
fondo.setBackground(new Color(51, 153, 255));  
fondo.setIcon(null);  
fondo.setBounds(662, 643, 46, 38);  
contentPane.add(fondo);
```

```
JLabel imagen = new JLabel("");  
imagen.setIcon(new  
ImageIcon(V_TEMARIO1.class.getResource("/Img/barra2.jpg")));  
imagen.setBounds(-71, 635, 1452, 133);  
contentPane.add(imagen);  
JLabel imagen1 = new JLabel("");  
imagen1.setIcon(new  
ImageIcon(V_ACTIVIDADES1.class.getResource("/Img/titulo_actividades1.png")));
```

```
imagen1.setBounds(332, 5, 842, 73);
contentPane.add(imagen1);

JLabel fondo1 = new JLabel("");

fondo1.setIcon(new
ImageIcon(V_TEMARIO1.class.getResource("/Img/barra2.jpg")));

fondo1.setBounds(0, 0, 113, 637);

contentPane.add(fondo1);

JLabel fondo2 = new JLabel("");

fondo2.setIcon(new
ImageIcon(V_TEMARIO1.class.getResource("/Img/barra2.jpg")));

fondo2.setBounds(1252, 0, 108, 637);

contentPane.add(fondo2);

JLabel imagen2 = new JLabel("");

imagen2.setIcon(new
ImageIcon(V_ACTIVIDADES1.class.getResource("/Img/puerta_alicia.gif")));

imagen2.setBounds(232, 234, 360, 261);

contentPane.add(imagen2);

JPanel panel = new JPanel();

panel.setBackground(new Color(0, 153, 255));

panel.setBounds(232, 154, 360, 90);

contentPane.add(panel);

JButton boton = new JButton("");

panel.add(boton);

boton.addActionListener(new ActionListener() {
```

```

public void actionPerformed(ActionEvent e) {
    V_ACT1_BLOQUE1 llamar= new V_ACT1_BLOQUE1();
    llamar.setVisible(true);
    llamar.setLocationRelativeTo(null);
    GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();

    grafica.setFullScreenWindow(llamar);
    dispose();}

});

boton.setIcon(new
ImageIcon(V_TEMARIO1.class.getResource("/Img/BLO1.png")));

JPanel panel_1 = new JPanel();
panel_1.setBackground(new Color(0, 153, 255));
panel_1.setBounds(232, 489, 360, 90);
contentPane.add(panel_1);

JButton imagen3 = new JButton("");
panel_1.add(imagen3);

imagen3.setIcon(new
ImageIcon(V_TEMARIO1.class.getResource("/Img/BLO3.png")));

JLabel imagen4 = new JLabel("");
imagen4.setIcon(new
ImageIcon(V_ACTIVIDADES1.class.getResource("/Img/enano_bn.gif")));
imagen4.setBounds(771, 234, 360, 261);

```

```
contentPane.add(imagen4);
```

```
JPanel panel_2 = new JPanel();
```

```
panel_2.setBackground(new Color(0, 153, 255));
```

```
panel_2.setBounds(771, 154, 360, 90);
```

```
contentPane.add(panel_2);
```

```
JButton boton3 = new JButton("");
```

```
panel_2.add(boton3);
```

```
boton3.setIcon(new  
ImageIcon(V_TEMARIO1.class.getResource("/Img/BLO2.png")));
```

```
JPanel panel_3 = new JPanel();
```

```
panel_3.setBackground(new Color(0, 153, 255));
```

```
panel_3.setBounds(771, 489, 360, 90);
```

```
contentPane.add(panel_3);
```

```
JButton boton5 = new JButton("");
```

```
panel_3.add(boton5);
```

```
boton5.setIcon(new  
ImageIcon(V_TEMARIO1.class.getResource("/Img/BLO4.png")));
```

```
JButton Regresar = new JButton("REGRESAR");
```

```
Regresar.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        V_MENU llamar= new V_MENU();
        llamar.setVisible(true);

        GraphicsDevice
grafica=GraphicsEnvironment.getLocalGraphicsEnvironment().getDefaultScreenDev
ice();

        grafica.setFullScreenWindow(llamar);

        llamar.setLocationRelativeTo(null);

        dispose();
    }
});

Regresar.setFont(new Font("Tahoma", Font.BOLD, 12));
Regresar.setBounds(1017, 594, 114, 30);
contentPane.add(Regresar);

JButton Salir = new JButton("SALIR");
Salir.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});

Salir.setFont(new Font("Tahoma", Font.BOLD, 12));
Salir.setBounds(1141, 594, 99, 29);
contentPane.add(btnSalir);

JLabel imagen5 = new JLabel("");
```

```
        imagen5.setIcon(new
ImageIcon(V_ACTIVIDADES1.class.getResource("/Img/barra2.jpg")));
        imagen5.setBounds(111, 89, 1141, 21);
        contentPane.add(imagen5);
    }

private class SwingAction extends AbstractAction {
    public SwingAction() {
        putValue(NAME, "Cambiar");
    }
}
}
}
```

BIBLIOGRAFÍA

[1] H. Torres Maldonado y D.A. Girón Padilla, *Didáctica General*, Argentina, Coordinación Educativa y Cultural Centroamericana, CEEC/SICA, 2009.

[2] G. Aragón González et al., "*Metodología para la construcción de evaluaciones*", Universidad Autónoma Metropolitana, Ciencias básicas e ingeniería, Reporte de investigación, 31.

[3] <https://www.eclipse.org/>

[4] Deitel, *Como programar en Java*, Editorial Pearson, Quinta edición, 2010.

[5] <http://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/edavis/swing.html>

[6] <http://www.mysql.com/>

[7] G. Aragón González et al., "*Tutores inteligentes en educación básica, media y superior*", Sociedad Cooperativa STG Consulting, Proyecto e-ducación, agosto de 2007.

[8] Rocha, González, Rodríguez, Rosainz. *Matemáticas para la vida 1*, Editorial Pearson Educación de México, 1ra Edición, 2007.

Rocha, González, Rodríguez, Rosainz. *Matemáticas para la vida 2*, Editorial Pearson Educación de México, 2da Edición, 2008.

Rocha, González, Rodríguez, Rosainz. *Matemáticas para la vida 3*, Editorial Pearson Educación de México, 1ra Edición, 2007.

[9] <http://sistemas.uniandes.edu.co/~csof5101/dokuwiki/lib/exe/fetch.php?media=principal:csof5101-requerimientos.pdf>

[10] http://www.sparxsystems.com.ar/download/ayuda/erica_content_static.html

[11] Craig Larman, *UML y patrones*, Prentice Hall, Pearson Education, 2da Edición, 2009.

