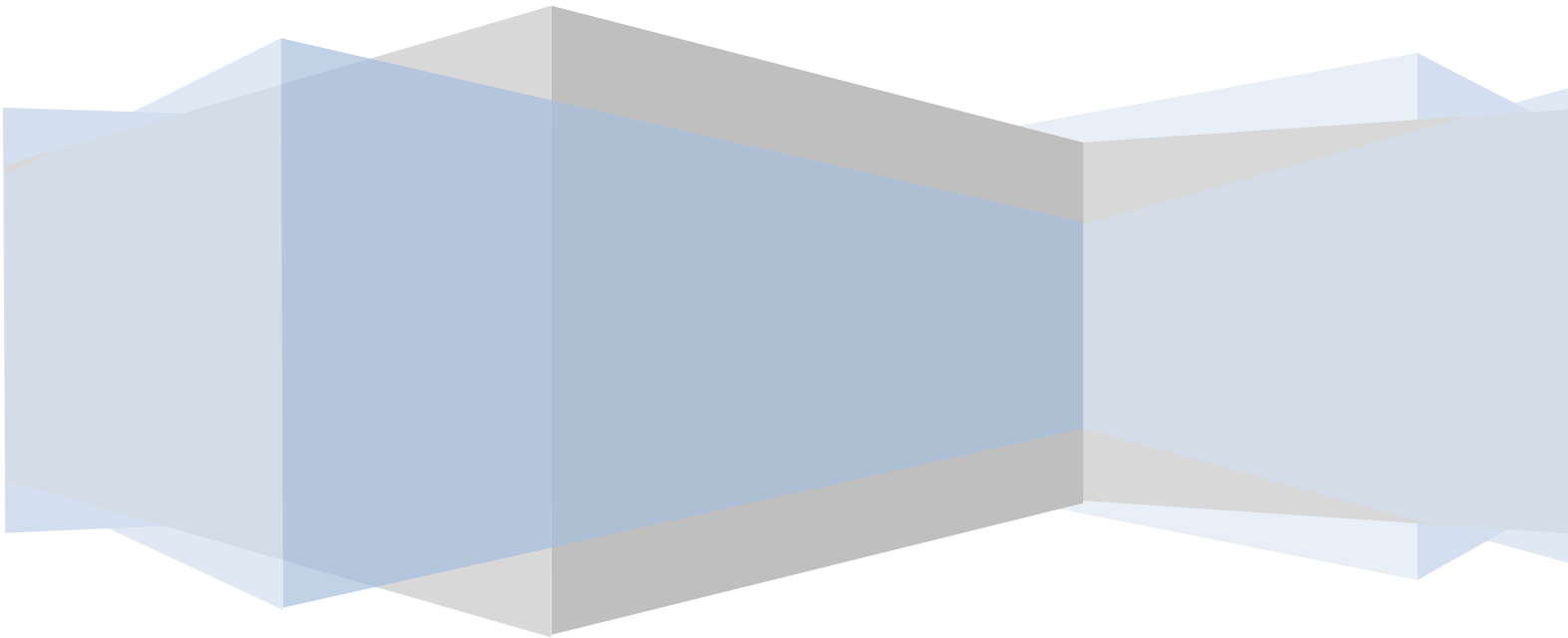
CAPÍTULO 5

Desarrollo



CAPÍTULO 5. DESARROLLO

5.1 Planteamiento

En la actualidad los estudiantes de la asignatura de Álgebra necesitan “aprender-aprender” tópicos de mayor reprobación como son: sistemas de ecuaciones lineales y polinomios, etc. El aprender es apropiarse de un conocimiento de tal forma que el esfuerzo por aprenderlo gira en torno a elementos como son:

El profesor como agente que contribuye a la formación integral (conocimientos, habilidades y actitudes) y el alumno como responsable de su aprendizaje.

La ingeniería como encargada de aplicar conocimientos científicos y empíricos para la creación e innovación de sistemas y procesos.

Para el desarrollo de conocimientos científicos la base fundamental la constituyen las matemáticas y en particular la asignatura de álgebra que busca formar alumnos que sean capaces de analizar y manejar los conceptos básicos y aplicarlos al estudio de la física, la matemática aplicada y la ingeniería.

Dentro de la asignatura se trabaja la resolución de problemas, en el tema de ecuaciones lineales. Al resolver los problemas no hay una metodología, técnica o estrategia, por lo que se sugiere el uso del ABP.

5.2 Objetivo

Diseñar, desarrollar e implementar un sistema de micromundos para la asignatura de Álgebra en particular de los temas de sistemas de ecuaciones lineales y polinomios. Estos conocimientos serán integrados en una base de datos. El profesor podrá crear, editar, eliminar ver los problemas que el alumno realizará. Por otro lado, el alumno resolverá el problema de acuerdo a la simulación realizada con la información proporcionada por el profesor.

5.3 Metodología

En el desarrollo de software, el ciclo de vida es el conjunto de fases por las que pasa un sistema que se está desarrollando desde que se tiene la idea inicial hasta que el software es retirado o cambiado (muere).

Según el Instituto Nacional de Tecnologías de la Información entre las funciones que puede tener un ciclo de vida se pueden destacar:

- Determinar el orden de las fases del proceso de software
- Establecer los criterios de transición para pasar de una fase a la siguiente
- Definir las entradas y salidas de cada fase
- Describir los estados por los que pasa el producto
- Describir las actividades a realizar para transformar el producto
- Definir un esquema que sirve como base para planificar, organizar, coordinar, desarrollar

Al realizar el desarrollo de un sistema de software es importante analizar los distintos ciclos de vida a los que se puede someter dicho desarrollo.

Modelo	Definición	Ventajas	Desventajas
Cascada	Es un proceso de desarrollo secuencial, en el que el desarrollo se ve fluyendo hacia abajo (como una cascada) sobre las fases que componen el ciclo de vida	- Puede ser apropiado, en general, para proyectos estables, donde es posible y probable que los diseñadores predigan totalmente áreas de problema del sistema y produzcan un diseño correcto antes de que empiece la implementación - Es un modelo en el que no se mezclan las fases, es simple y fácil de usar - Es fácil de gestionar ya que cada fase tiene entregables específicos y un proceso de revisión - Las fases son procesadas y completadas de una vez	- En la vida real un proyecto rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lo lleve al fracaso - Trabajar con este modelo provoca un gran atraso, ya que es muy restrictivo y no permite movilizarse entre fases - Los resultados y mejoras no son visibles progresivamente, el producto se ve cuando ya está finalizado - Es un modelo pobre para proyectos largos, complejos, orientados a objetos y por su puesto en aquellos en los que los registros tengan un riesgo de moderado a alto de cambiar. Genera altas cantidades de riesgos e incertidumbres

V	Es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Describe las actividades y resultados que han de ser producidos durante el desarrollo del producto. La parte izquierda de la V representa la descomposición de los requisitos y la creación de las especificaciones del sistema. El lado derecho de la V representa la integración de partes y su verificación. V significa "Validación y Verificación"	- En este modelo no hace falta que los requisitos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir refinando en cada una de las iteraciones - Tiene las ventajas propias de realizar el desarrollo en pequeños ciclos, lo que permite gestionar mejor los riesgos y gestionar mejor las entregas	- El hecho de no tener definidos los requisitos desde el principio también puede ser desventaja, ya que pueden surgir problemas relacionados con la arquitectura
Desarrollo incremental	Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.	- Mediante este modelo se genera software operativo de forma rápida y en etapas tempranas del ciclo de vida del software - Es un modelo más flexible, por lo que se reduce el coste en el cambio de alcance y requisitos - Es más fácil probar y depurar en una iteración más pequeña - Es más fácil gestionar riesgos	- Cada fase de una iteración es rígida y no se superponen con otras - Pueden surgir problemas referidos a la arquitectura del sistema porque no todos los requisitos se han reunido, ya que se supone que todos ellos se han definido al inicio

Espiral	Las actividades de este modelo se conforman en una espiral, cada bucle representa un conjunto de actividades. Las actividades no están fijadas a priori, sino que las siguientes se eligen en función del análisis de riesgos, comenzando por el bucle anterior	- Reduce riesgos del proyecto - Incorpora objetivos de calidad - Integra el desarrollo con el mantenimiento - Es posible tener en cuenta mejoras y nuevos requerimientos sin romper con el modelo, ya que el ciclo de vida no es rígido ni estático - Mediante este modelo se produce software en etapas tempranas del ciclo de vida y suele ser adecuado para proyectos largos de misión crítica	- Es un modelo que genera mucho trabajo adicional. Al ser el análisis de riesgos una de las tareas principales exige un alto nivel de experiencia y cierta habilidad en los analistas de riesgos (es bastante difícil) - Es un modelo costoso - No es un modelo que funcione bien para proyectos pequeños

Tabla 12. Tipos de ciclos de vida

Una combinación de los ciclos antes expuestos, forma la base de una metodología, ya que una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo.

La metodología para el desarrollo de software en un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado.

En el ámbito de ingeniería del software, podemos destacar que una **metodología**:

- Optimiza el proceso y el producto de software.
- Métodos que guían en la planificación y en el desarrollo del software.
- Define qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto.

Una metodología define una estrategia global para enfrentarse con el proyecto. Entre los elementos que forman parte de una metodología se pueden destacar:

- Fases: tareas a realizar en cada fase.
- Productos: E/S de cada fase, documentos.
- Procedimientos y herramientas: apoyo a la realización de cada tarea.
- Criterios de evaluación: del proceso y del producto. Saber si se han logrado los objetivos.

Existen diversos tipos de metodologías, de las más destacadas se presenta a continuación una tabla comparativa.

Metodología	*Definición*	*Características*
EXTREME PROGRAMMING (XP)	Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.	Elementos - Las historias de usuario - Roles XP - Proceso XP Principios - Simplicidad - Comunicación - Realimentación - Coraje o valentía Fases - Codificación - Pruebas - Escuchar - Diseño

	XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. A Kent Beck se le considera el padre de XP	
RATIONAL UNIFIED PROCESS (RUP)	Es un marco de trabajo de proceso de desarrollo de software iterativo. RUP no es un proceso preceptivo concreto individual, sino un marco de trabajo de proceso adaptable, con la idea de ser adaptado por las organizaciones de desarrollo y los equipos de proyecto de software que seleccionarán los elementos del proceso que sean apropiados para sus necesidades.	Elementos - Roles - Productos de trabajo - Tareas Principios - Adaptación del proceso - Equilibrio de prioridades - Demostración - Colaboración - Abstracción - Calidad Fases - Iniciación - Elaboración - Construcción - Transición
MICROSOFT SOLUTION FRAMEWROK (MSF)	Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.	Elementos -Modelo de Arquitectura del Proyecto - Modelo de Equipo - Modelo de Proceso - Modelo de Gestión del Riesgo - Modelo de Diseño del Proceso - Modelo de Aplicación Principios - Adaptabilidad - Escalabilidad - Flexibilidad - Tecnología agnóstica Fases -Estrategia y alcance

	-Planificación y prueba de concepto - Estabilización

Tabla 13. Metodologías

5.4 Análisis

La metodología que se utilizará en el sistema para generar micromundos para la materia de álgebra será Extreme Programming (XP), debido a que es una de las metodologías pertenecientes a los métodos ágiles, mismos que ponen énfasis en la adaptabilidad y no en la previsibilidad. XP permitirá los cambios de requisitos en cualquier punto de vida del proyecto, lo que se ha considerado un excelente beneficio para poder contar con una mejor y más realista aproximación de lo que se desea desarrollar, y así mismo esto permitirá intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. De manera general se considera a XP como una buena opción de metodología para llevar a cabo el proyecto y así mismo poder aplicarlo de manera dinámica durante el ciclo de vida del sistema.

XP, es una metodología ágil centrada en cuatro principios básicos y de fundamental importancia en el desarrollo de un proyecto, tales como simplificar un diseño para agilizar el desarrollo y facilitar el mantenimiento (simplicidad), la comunicación es otro de los principios de esta metodología y el cual tiene la finalidad de entablar una excelente relación entre el usuario y el desarrollador, y por ende llevar al éxito el proyecto cumpliendo de manera satisfactoria los requisitos establecidos por el usuario. La Realimentación es la continuación del principio anterior, y gracias a este principio se pueden realizar prácticas mediante ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos del usuario y ayudará a centrarse en los requisitos de mayor importancia, de manera general con el principio de realimentación se ejecutan pruebas unitarias frecuentemente que permiten descubrir fallos. Y por último el principio de valentía con el que deberá contar todo aquel que desee desarrollar un proyecto de cualquier índole, ya que se debe tener la valentía para enfrentar los requerimientos del usuario y a su vez implementar las características que esté mismo desee modificar.

Se trabajará bajo esta metodología, debido a que así se podrán implementar prácticas cortas que brindarán la posibilidad de disminuir la mítica curva exponencial del costo y del tiempo a lo largo del proyecto, por lo que se realizarán las prácticas suficientes para que el diseño evolutivo funcione.

Es necesario destacar que las prácticas se realizarán a lo largo del proyecto en todas sus fases, mismas ya establecidas en la metodología XP, tales como exploración, planificación de la entrega, iteraciones, producción, mantenimiento y muerte del proyecto. Una de las ventajas de utilizar XP, es que el bucle que está forma es eficaz y sencillo, si se aplican los principios de la presente metodología, ya que solo consisten en comunicarse con el usuario, estimar lo que se necesita para cumplir el requerimiento del usuario, construir o desarrollar lo que el usuario desea, y nuevamente regresar a mostrarlo al usuario teniendo la capacidad de entender que puede o no lograrse la satisfacción total del usuario y así mismo es necesario contar con la valentía para actuar de manera habilidosa ante las peticiones del usuario. Lo anterior respalda la decisión del uso de la metodología XP, ya que intenta reducir la complejidad del software o aplicación a desarrollar, por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad y la valentía de reacción.

El diseño evolutivo que representa la metodología XP colabora a que no se le dé demasiada importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento.

- Herramientas de software.

Se hará uso de diversas herramientas de software libre, debido a las ventajas que estos nos ofrecen.

Tabla comparativa		
	Software Libre	Software Propietario
Costo Adquisición	Ninguno	Costoso
Uso	Libre	Restringida
Innovación tecnológica	Rápida	Lenta
Requisitos del Hardware	Pocos	Muchos
Soporte del Hardware	Muy alto	Bajo en equipos antiguos
Soporte Técnico	Medio	Bueno
Independencia del proveedor	Si	No
Adaptación del Software	Muy alta	Media
Idiomas, dialectos, etc.	Fácil de implementar	Los más usados.
Curva Aprendizaje	Baja	Alta
Costo de aprendizaje	Ninguno	Alto
Garantía	Ninguna	Si
Reparación de errores	Se puede hacer en el momento	Hasta otra versión
Nivel de programación	Alto	Poco
Control de Calidad	Medio	Alto
Recursos de Investigación	Bajo	Alto
Difusión del Software	Bajo	Alto

Tabla 14. Diferencias entre software libre y propietario.

- PHP

Ventajas

- ✓ Es un lenguaje multiplataforma.
- ✓ Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- ✓ El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- ✓ La programación en PHP sea segura y confiable.
- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad (MySQL, PostgreSQL, etc.)
- ✓ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- ✓ Posee una amplia documentación en su página oficial.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos.
- ✓ Biblioteca nativa de funciones sumamente amplia e incluida.
- ✓ No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- ✓ Tiene manejo de excepciones (desde PHP5).
- ✓ Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Inconvenientes

- ✓ La ofuscación de código es la única forma de ocultar las fuentes.

  - Mysql

  Esta herramienta se analizó en el capítulo 3.4

  - Javascript

- ✓ Es un software orientado al usuario.

- ✓ Permite identificar ciertas acciones del usuario con el sistema.

  - Apache

  Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos .

  Ventajas

- ✓ Multiplataforma.
- ✓ Es un servidor de web conforme al protocolo HTTP.
- ✓ Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- ✓ Basado en hebras en la versión 2.0
- ✓ Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- ✓ Se desarrolla de forma abierta.
- ✓ Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.
- ✓

Por lo que se utilizara el software PHP, Apache y Javascript para la implementación del proyecto

5.5 Diseño

A continuación se presentan los diagramas de flujo y el diagrama Entidad-Relación de la base de datos.

5.5.1 Diagramas de flujo

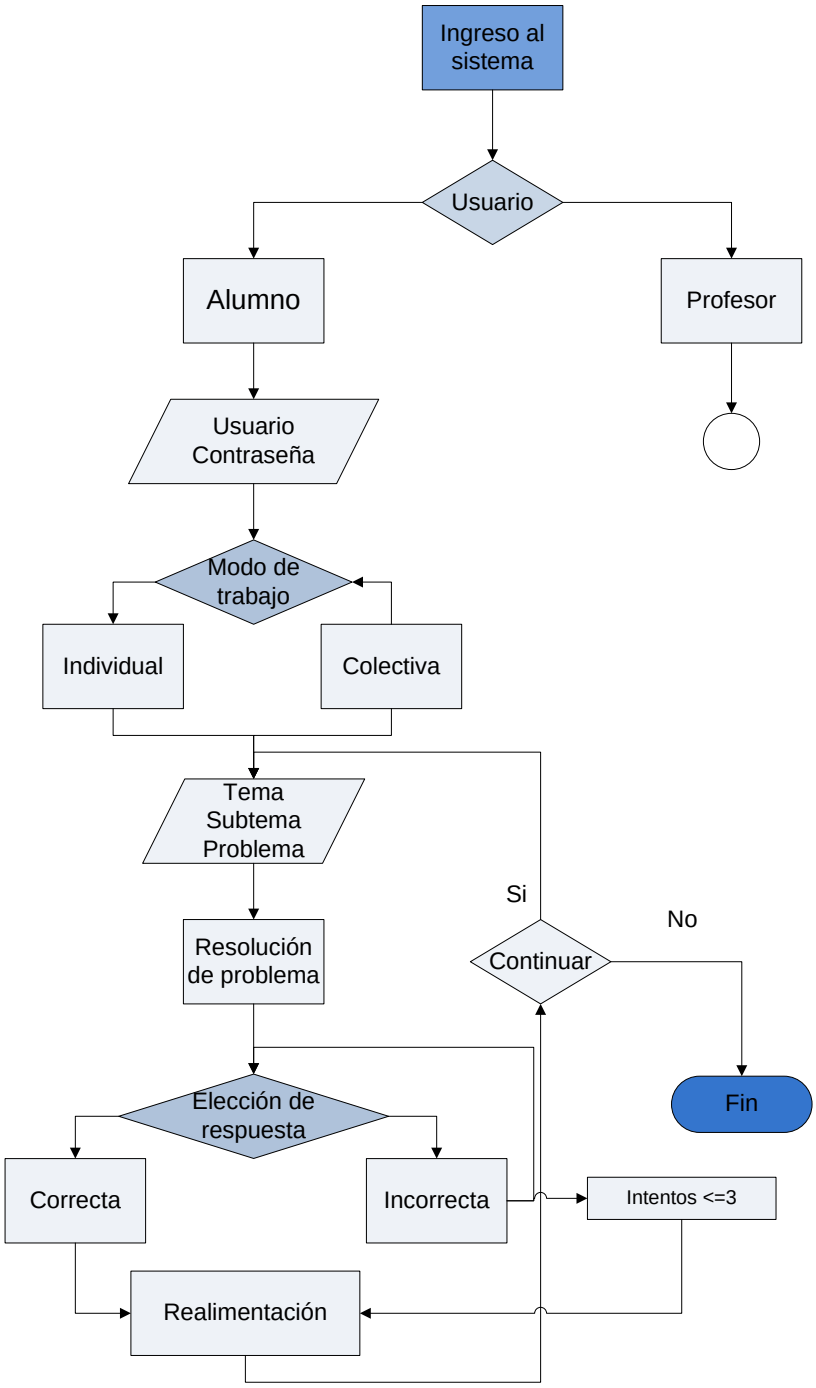


Figura 26 Diagrama de flujo del alumno

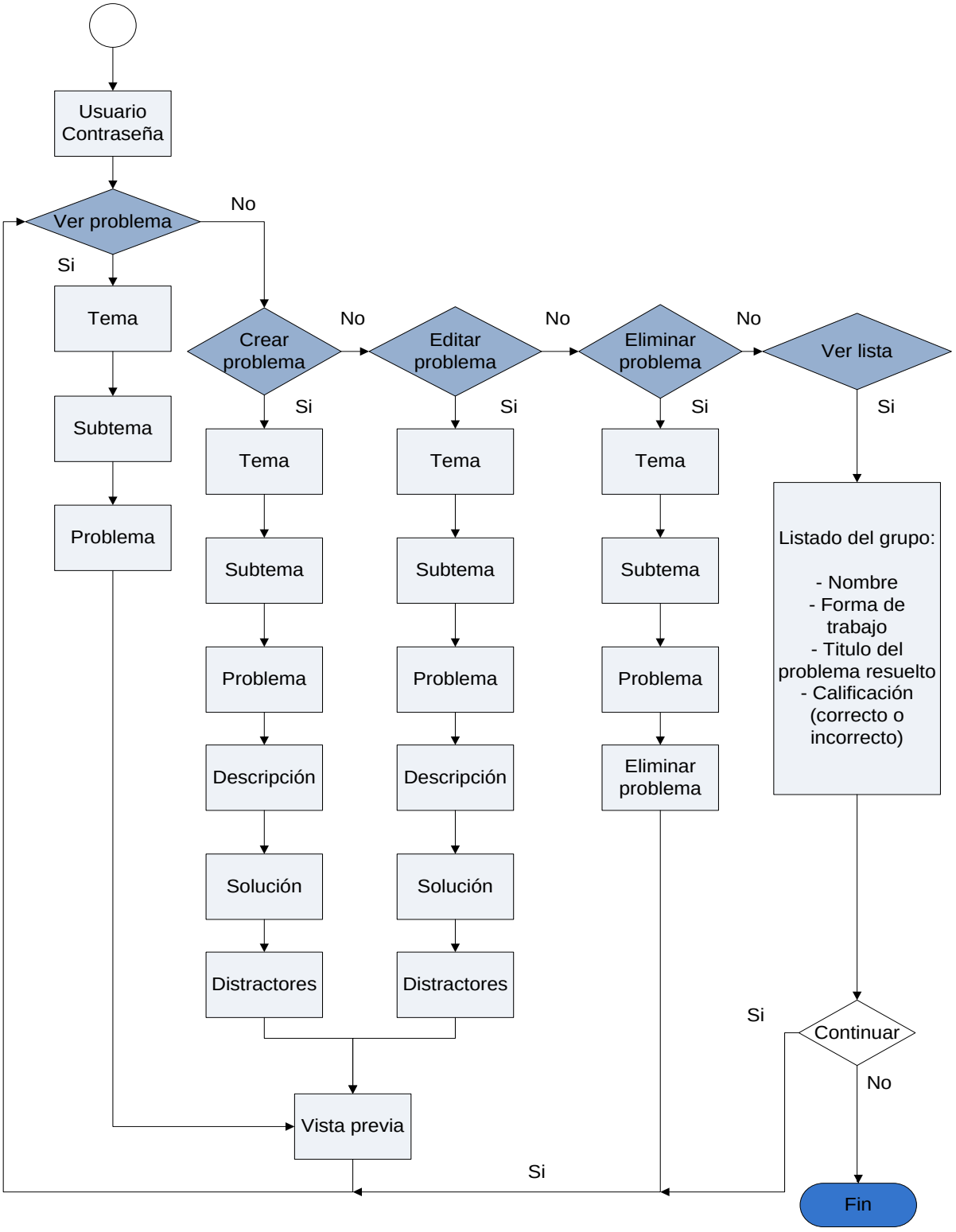


Figura 27 Diagrama de flujo del profesor

5.5.2 Diseño de la Base de datos

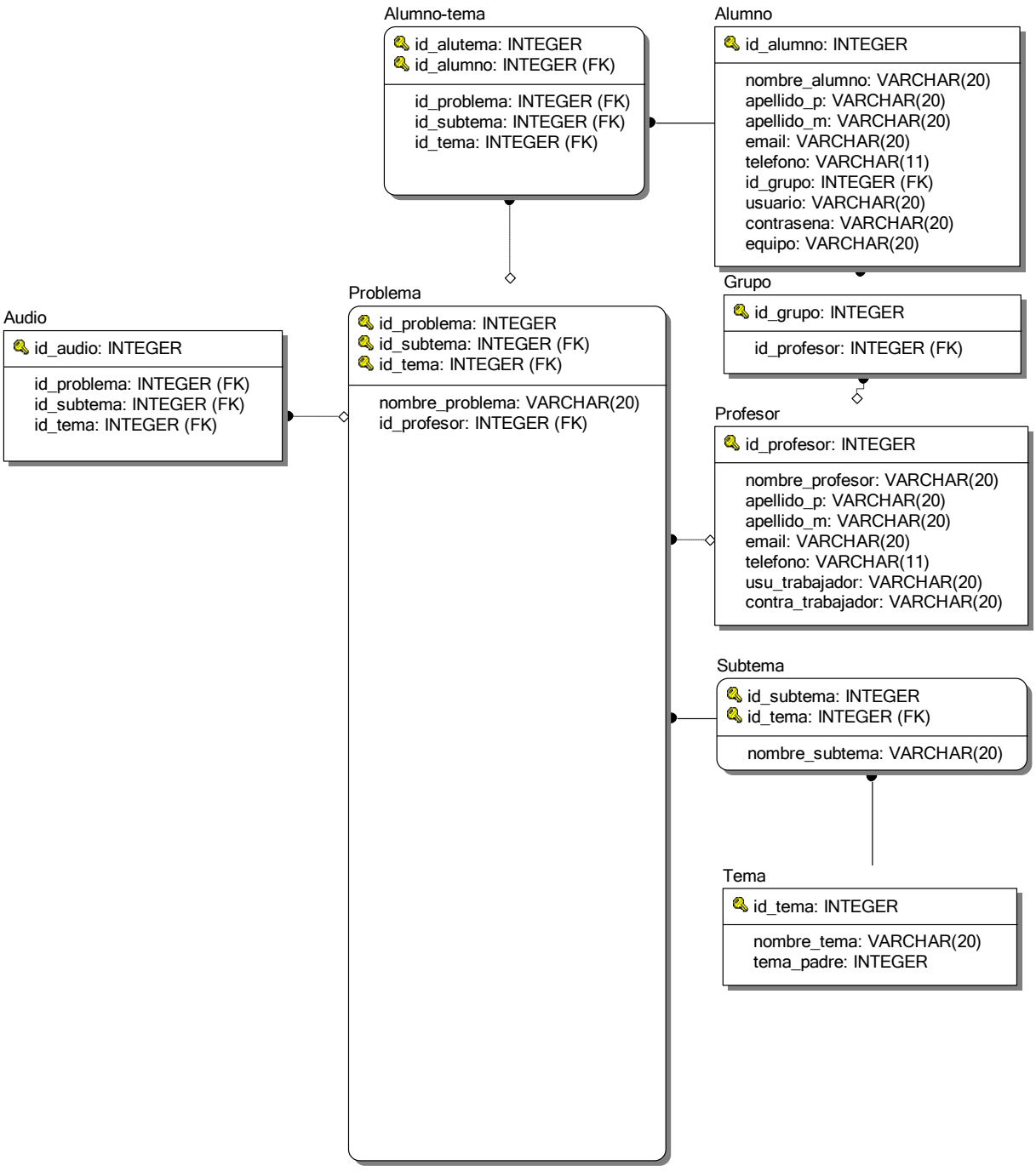


Figura 28 Diagrama Entidad-Relación

5.6 Implementación

Utilizando la metodología Extreme Programming (XP), y con ayuda las herramientas de software mencionadas se diseño y desarrollo de manera satisfactoria el Sistema para Generar Micromundos para la asignatura de álgebra. El sistema cuenta con tres módulos: el profesor, el alumno y el administrador.

- Profesor

Figura 29 Profesor

Para tener acceso al sistema, el profesor debe validarse, ingresando usuario y contraseña, para fines prácticos es el número de trabajador para ambos casos.

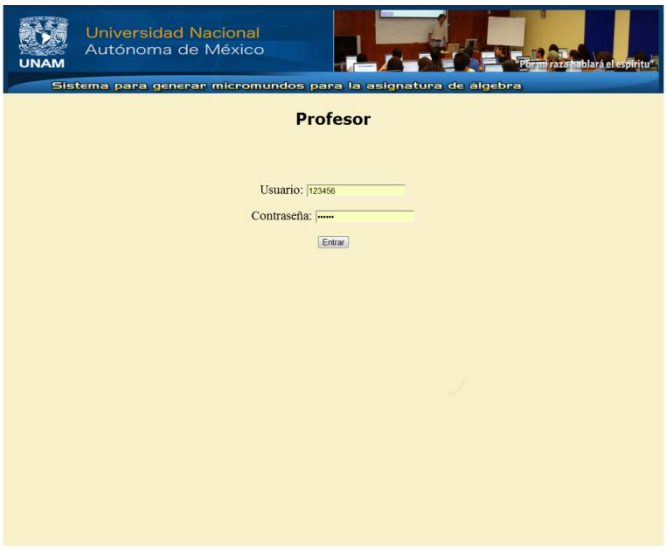


Figura 30 Iniciando sesión

Una vez que se valida la cuenta del profesor, este puede elegir una de cinco opciones a realizar:

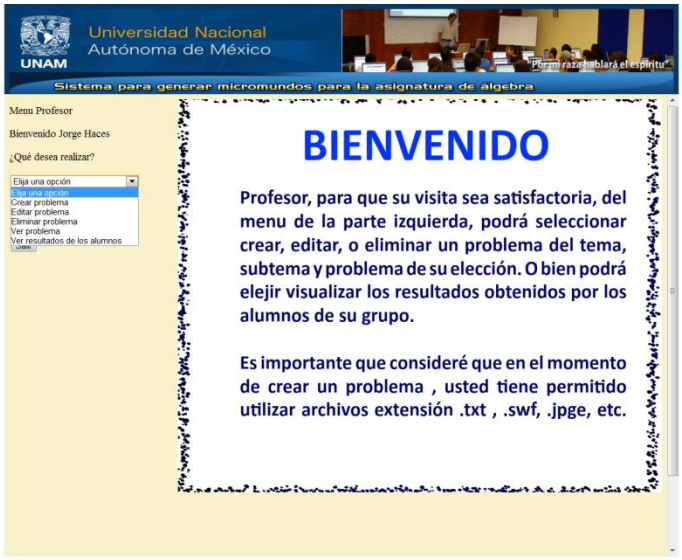


Figura 31 Menú del profesor

- *Crear problema*

  - ✓ Aparece un formulario en que el profesor elije el tema, subtema y descripción del problema. El usuario también puede elegir el número de distractores con su respectiva realimentación.

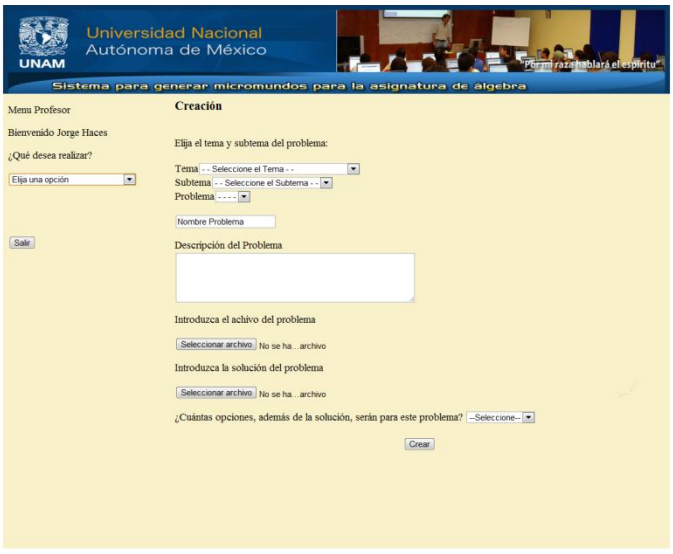


Figura 32 Creación de un problema

- ✓ Ya seleccionado el botón crear, se muestra la vista previa del problema, con sus distractores y su respectiva realimentación.

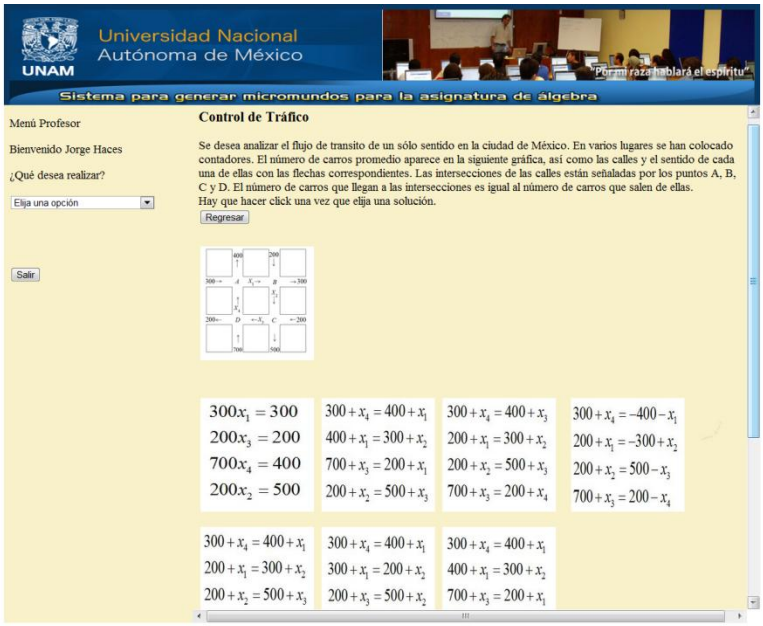


Figura 33 Vista previa de un problema

- ➢ *Editar problema*
  - ✓ El profesor elige el tema, subtema y nombre del problema a editar.

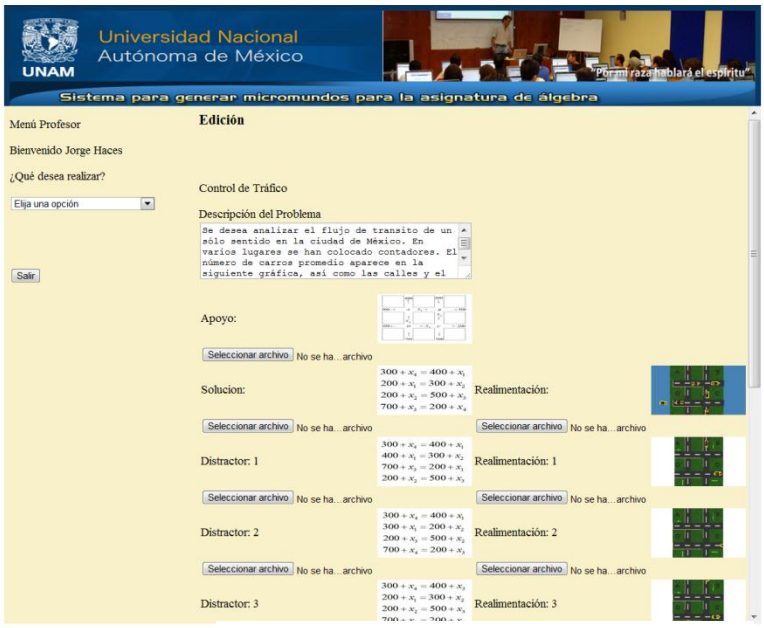


Figura 34 Edición de un problema

- ✓ Se hacen los cambios de archivos correspondientes
- ✓ Se genera la vista previa del problema editado.

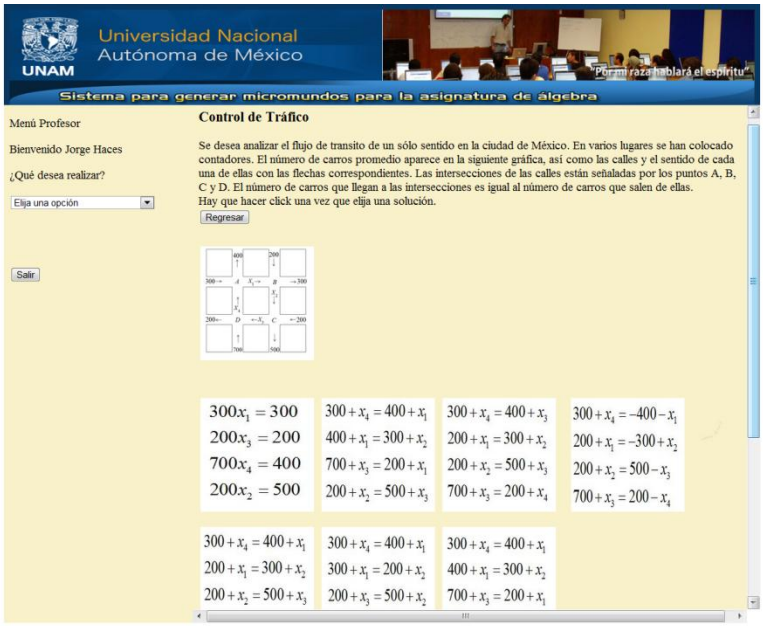


Figura 35 Vista previa de un problema

- ➢ *Eliminar problema*

Figura 36 Eliminación de un problema

- ✓ El profesor elige el tema, subtema y nombre del problema que eliminará.

Figura 37 Selección de un problema

- ✓ El profesor confirma su decisión de eliminar el problema.

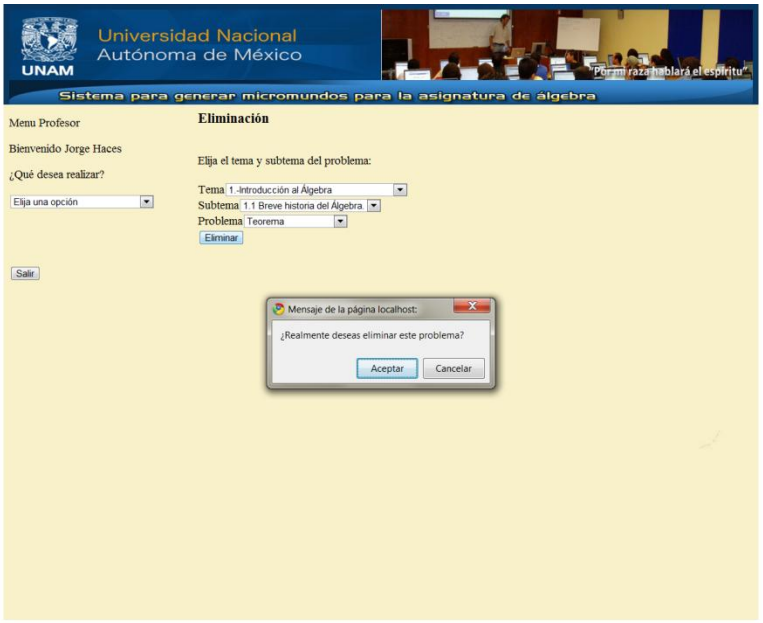


Figura 38 Confirmar la eliminación de un problema

- *Ver problema*, puede visualizar el problema de su elección, seleccionando su ubicación correcta, es decir, tema, subtema y nombre.

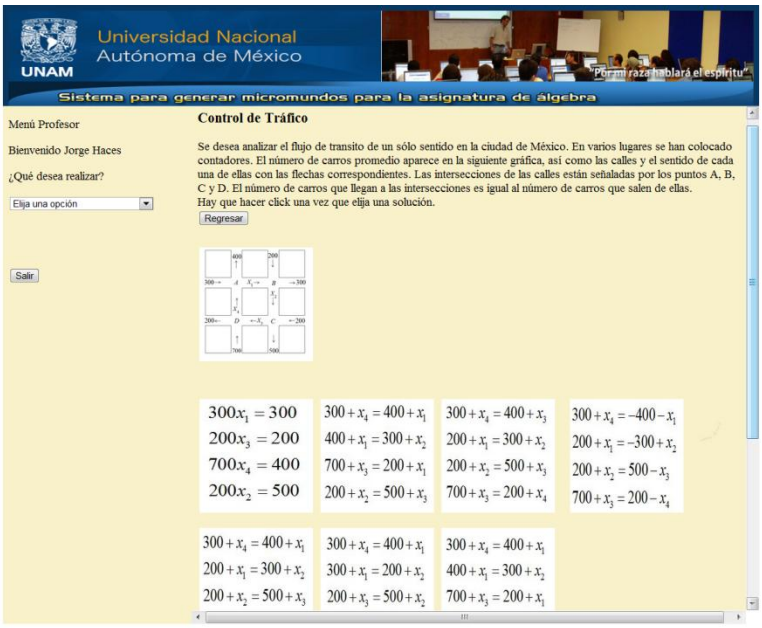
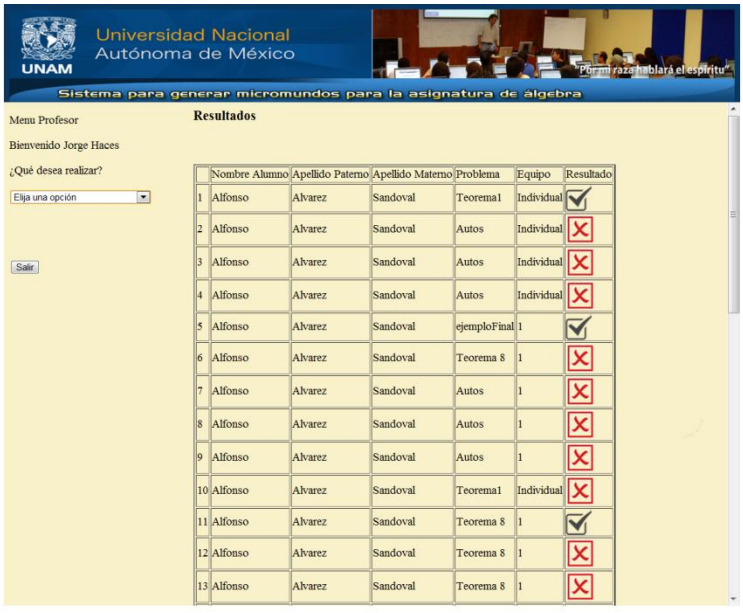


Figura 39 Ver un problema

- *Ver resultados de los alumnos*, el profesor puede visualizar la lista de su grupo, misma que contiene nombre del alumno, nombre del o los ejercicios resueltos, así como sí trabajo de forma individual o en su defecto el número de equipo.



	Nombre Alumno	Apellido Paterno	Apellido Materno	Problema	Equipo	Resultado
1	Alfonso	Alvarez	Sandoval	Teorema1	Individual	✓
2	Alfonso	Alvarez	Sandoval	Autos	Individual	✗
3	Alfonso	Alvarez	Sandoval	Autos	Individual	✗
4	Alfonso	Alvarez	Sandoval	Autos	Individual	✗
5	Alfonso	Alvarez	Sandoval	ejemploFinal	1	✓
6	Alfonso	Alvarez	Sandoval	Teorema 8	1	✗
7	Alfonso	Alvarez	Sandoval	Autos	1	✗
8	Alfonso	Alvarez	Sandoval	Autos	1	✗
9	Alfonso	Alvarez	Sandoval	Autos	1	✗
10	Alfonso	Alvarez	Sandoval	Teorema1	Individual	✗
11	Alfonso	Alvarez	Sandoval	Teorema 8	1	✓
12	Alfonso	Alvarez	Sandoval	Teorema 8	1	✗
13	Alfonso	Alvarez	Sandoval	Teorema 8	1	✗

Figura 40 Resultados de los alumnos

Al finalizar cada una de las opciones que el usuario puede elegir, se muestra la pantalla del menú inicial, en la cual el usuario elije salir o en su defecto realizar otra acción.

- *Alumno*

Para tener acceso al sistema, el alumno debe validarse, ingresando usuario y contraseña, para fines prácticos es el número de cuenta, también debe seleccionar su forma de trabajo (individual o colectiva).

➢ *Trabajo individual*

El alumno elije el tema, subtema y problema que desea resolver, o bien el que su profesor le ha indicado.

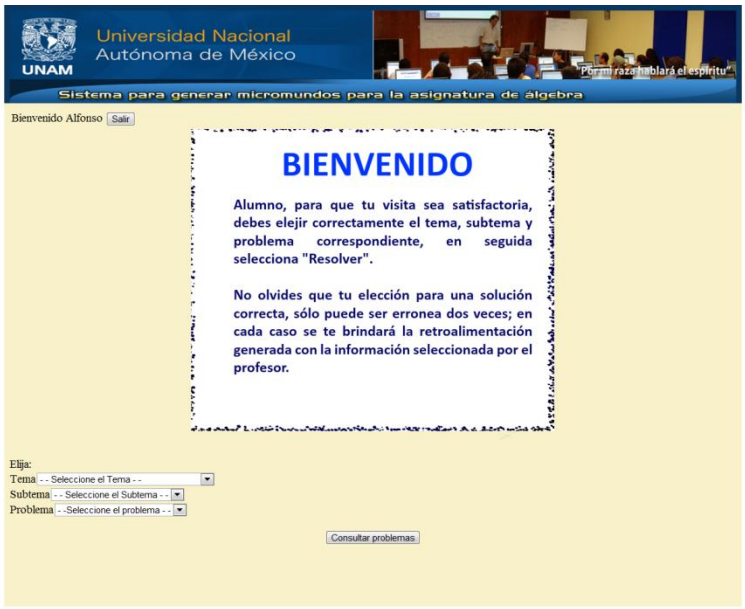


Figura 41 Menú del alumno(individual)

El alumno visualiza la descripción del problema, y las posibles soluciones.

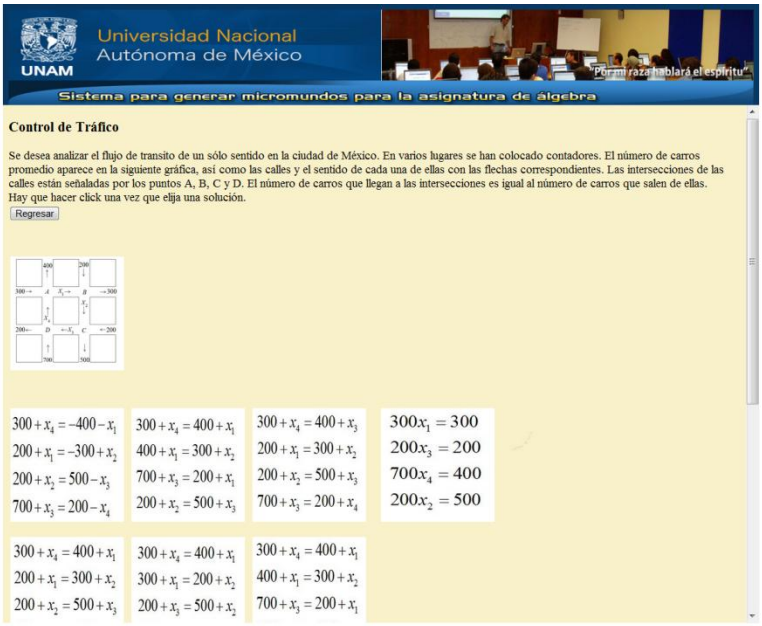


Figura 41 Problema

Si la respuesta elegida es incorrecta, visualizará la realimentación correspondiente al distractor que ha seleccionado.

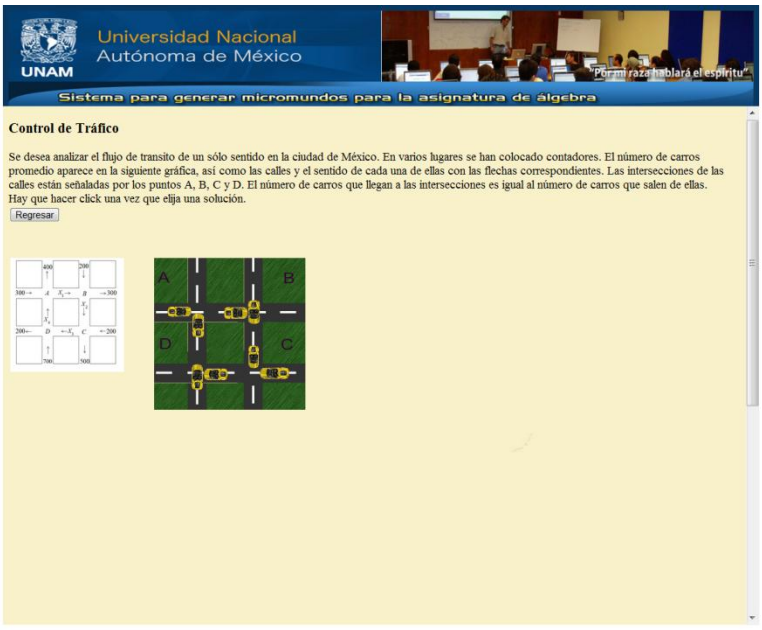


Figura 42 Respuesta incorrecta

Si la respuesta elegida es correcta, visualizará una notificación, o bien el contenido del archivo que el profesor eligió para la solución.

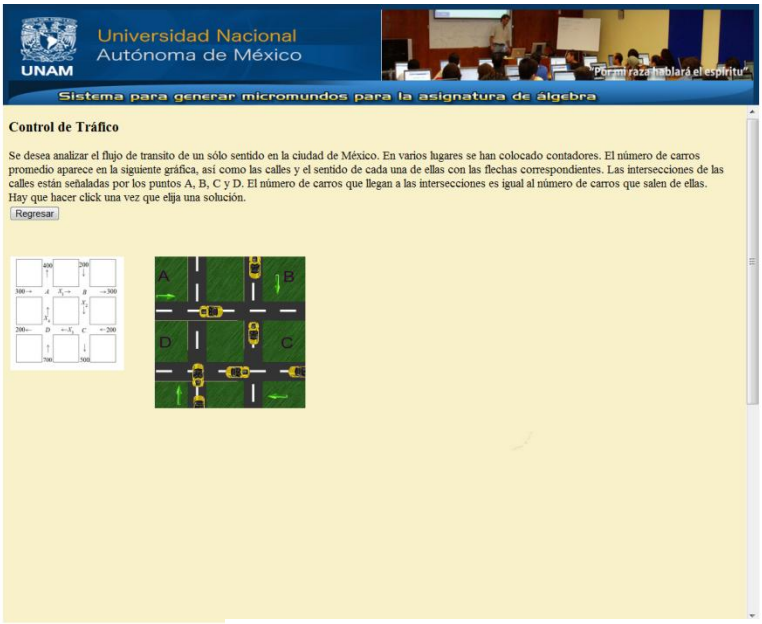


Figura 43 Respuesta Correcta

- Trabajo colectivo

El alumno elije el tema, subtema y problema que desea resolver, o bien el indicado por el profesor. Se puede visualizar en la parte izquierda de la página, los nombres de los integrantes del equipo.

Figura 44 Menú del alumno(Colectivo)

El alumno visualiza la descripción del problema, y las posibles soluciones.

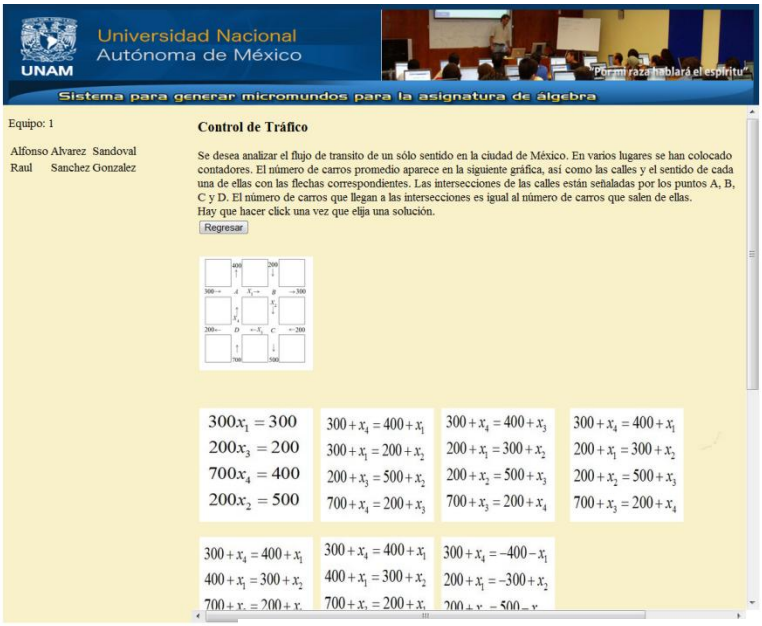


Figura 45 Problema

Si la respuesta elegida es incorrecta, visualizará la realimentación correspondiente al distractor que ha seleccionado.

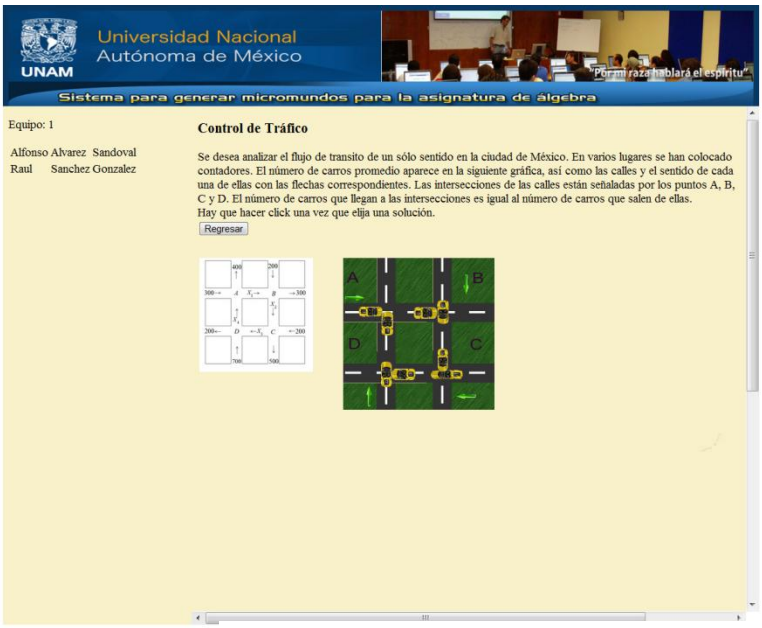


Figura 46 Respuesta Incorrecta

Si la respuesta elegida es correcta, visualizará una notificación, o bien el contenido del archivo que el profesor eligió para la solución.

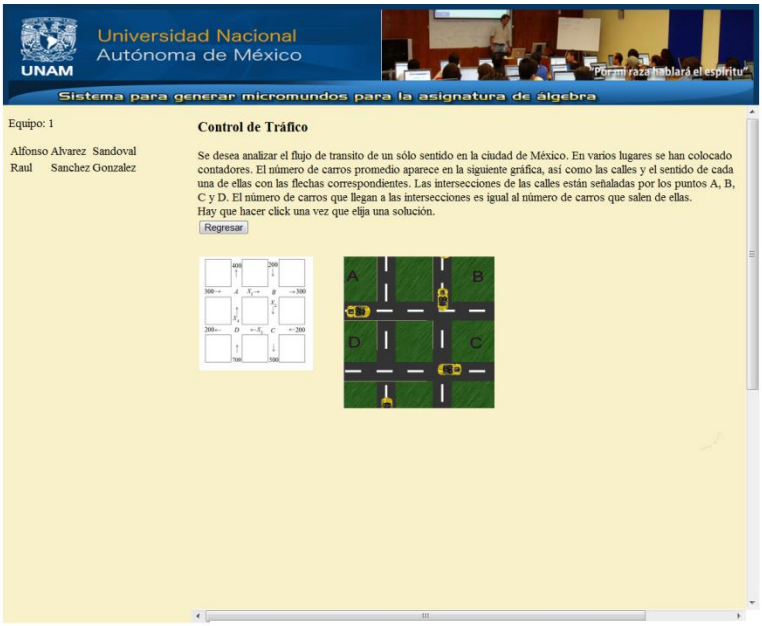


Figura 47 Respuesta Correcta

- Administrador

El admistrador tiene la capacidad de dar de alta o baja a los usuarios, puede realizar cambios pertinentes del sistema. En el caso de los alumnos, este asigna el número de equipo aleatoriamente para cada uno.

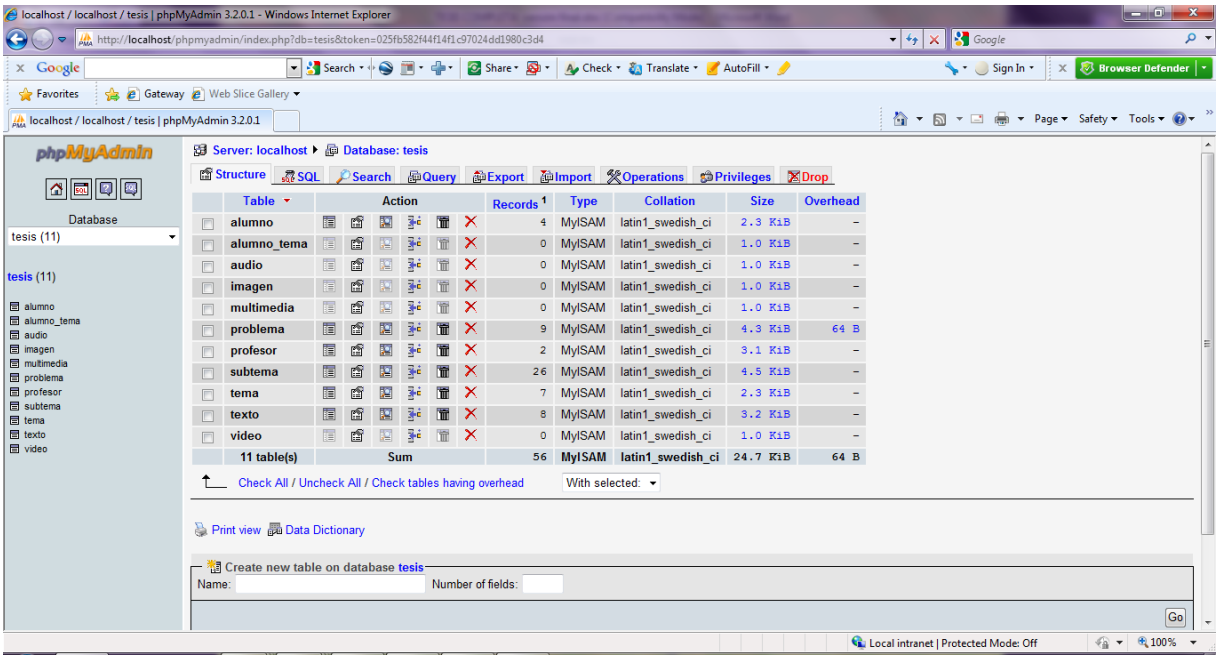


Figura 48 Administrador