



**BUAP**



Facultad de Ciencias  
de la Computación

**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA  
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

**“SIMULADOR WEB DE EXÁMENES DE LA MATERIA  
DE INGENIERÍA DE SOFTWARE PARA EL  
ACOMPañAMIENTO EN EL APRENDIZAJE DEL  
ALUMNO”**

**TESIS**

**PARA OBTENER EL GRADO DE:  
LICENCIATURA EN INGENIERÍA EN CIENCIAS DE LA  
COMPUTACIÓN**

**PRESENTA:  
JOSÉ MIGUEL LÓPEZ AGUILERA**

**DIRECTOR Y ASESOR DE TESIS:  
DR. MARIO ROSSAINZ LÓPEZ**

**PUEBLA, PUE.**

**AGOSTO, 2022**

## Agradecimientos

A mi tutor y asesor el Dr. Mario Rossainz López por su ayuda, paciencia y dedicación, su conocimiento y experiencia me ha servido de gran ayuda durante este proceso, mi total admiración y respeto. Asimismo, expreso mi gratitud hacia la facultad de ciencias de la computación por brindarme las enseñanzas durante este largo camino y a los docentes por compartir su sabiduría y conocimiento.

A mis amigos de la facultad (Bernardo Pérez López, José Carlos Flores Rivera, Abimael Rueda Galindo, Luis Eduardo Ríos Jaen, Fernando López Martínez, entre otros), por su apoyo y amistad que me han mostrado desde el primer día. Gracias, por esas horas de estudio y diversión, donde nos acompañamos en nuestra formación; tareas, proyectos, exámenes, toda reunión fue emocionante.

A mis perritos (Yeikob, Dingo, Gobi) y gatitos (Mio, Toby) los que están y los que se adelantaron; sé que no pueden leer esto, pero estoy contento por tenerlos y darle felicidad a mi vida en esos días que regresaba de la universidad y ellos me recibían con cariño.

A mis tíos/tías (Sofía Del Carmen Aguilera Matías, Iracema Aguilera Matías, Rosalinda Aguilera Matías y Felipe de Jesús Aguilera Matías), por apoyarme moral, anímica e incluso económicamente, estudiar fuera y buscar mejores oportunidades no hubiera sido posible sin los recursos y apoyos que me brindaron estos años.

A mis abuelos (Felipe Aguilera Viveros, Gudelia Matías Bautista, Rosa Aguilar García y José Gaudencio López Ronquillo), por estar ahí en todo momento viéndome crecer y estando presentes en cada etapa de mi vida, me llena de alegría que puedan verme terminar mis estudios, esto no lo pude lograr sin ustedes, sin sus consejos y apoyo incondicional.

Finalmente, a mis padres Thelma Aguilera Matías y Miguel Ángel López Aguilar, mi hermana Amayrani López Aguilera, gracias por su confianza, por los valores y principios que me han inculcado, sin mencionar los grandes sacrificios que hicieron para hacer realidad esto. Ustedes siempre son el motor que impulsan mis sueños y esperanzas, fueron muchas las veces que el camino se me complicaba, pero siempre han sido mi razón para seguir adelante, les dedico a ustedes este gran logro en mi vida.

# Índice

Agradecimientos .....	2
Introducción .....	5
Capítulo 1: Marco Teórico .....	8
1.1 Dominio del problema .....	8
1.2 Dominio computacional.....	9
Capítulo 2: Estado del Arte .....	11
2.1 UNITIPS .....	11
2.2 AppMisión .....	13
2.3 QuizPM.....	15
Capítulo 3: Planeación.....	17
3.1 Antecedentes .....	17
3.2 Estrategia de solución .....	21
3.3 Objetivos .....	22
3.4 Metodología .....	23
3.4.1 Análisis de requerimientos .....	23
3.4.2 Diseño web .....	26
3.4.3 Implementación .....	29
3.4.4 Ejecución y pruebas .....	29
Capítulo 4: Análisis de Requerimientos .....	30
4.1 Identificación de usuarios finales y casos de uso .....	30
4.2 Diagrama de casos de uso .....	33
4.3 Diagrama de actividades .....	34
Capítulo 5: Diseño .....	40
5.1 Diagrama de contenido .....	40
5.2 Diagrama de navegación.....	41
5.3 Diagrama de presentación.....	43
Capítulo 6: Estructura y Creación de la Base de Datos .....	52
6.1 Diseño conceptual (Entidad-Relación) .....	52
6.2 Diseño lógico (Modelo Relacional) .....	53
6.3 Diseño físico (Motor de Base de Datos).....	54

Capítulo 7: Implementación y Pruebas.....	57
7.1 Frontend con ReactJS .....	57
7.2 Backend con Python Flask.....	58
7.3 Pruebas.....	60
Conclusiones.....	66
Trabajos futuros .....	67
Bibliografía.....	68

## Introducción

El presente trabajo de tesis tiene como objetivo principal desarrollar una aplicación web destinada a la simulación de exámenes de la materia de Ingeniería de Software de los programas educativos de la Lic. e Ing. en Ciencias de la Computación de la Facultad de Ciencias de la Computación de la BUAP; con la finalidad de servir como acompañamiento en el aprendizaje que el alumno adquiere en esta asignatura. La motivación del presente trabajo tiene su origen en la problemática que se vive por el alumno en cuanto a su rendimiento académico, pues se estima que un porcentaje considerable de alumnos presenta problemas para acreditar dicha materia y esto además de afectar directamente su historia académica, también retrasa la finalización de su carrera. Una alternativa para disminuir dichos problemas es la propuesta presentada en esta tesis de desarrollar un simulador web que acompañe al alumno en su proceso de aprendizaje de la asignatura de Ingeniería de Software de manera que le permita mejorar su aprovechamiento y aprendizaje y sea una forma de preparación para futuros exámenes reales de la materia y que garantice una buena nota en ellos.

Se decidió desarrollar un simulador web porque hoy en día, las aplicaciones web cada vez toman mayor importancia en la vida cotidiana y debe de tener una presencia relevante en la educación, existen diversas plataformas de aprendizaje las cuales ofrecen una gran variedad de temas relacionados a ciencia, tecnología, entre otras, algunas de uso comercial y otras de forma libre pero todas ellas tienen el mismo propósito que es preparar a las personas en una área o tema específico y enriquecer sus conocimientos. El software que se desarrolla puede llegar a generar un gran impacto en la comunidad de la facultad, alumnos dispuestos a mejorar sus notas teniendo un nuevo recurso de aprendizaje en sus manos.

Este documento de tesis se forma de los siguientes capítulos:

En el capítulo 1 se presenta un marco teórico que permite conocer dos conceptos básicos para el entendimiento de esta aplicación web y proporciona al lector una idea más clara acerca de este tema.

**Dominio del problema:** Enfocado en un conjunto de conceptos interrelacionados que es necesario conocer para entender lo que se va a resolver. Menciona a grandes rasgos, ¿Cuál es el origen del problema?, ¿Cómo va manifestándose? y las posibles consecuencias en particular, además de ver los beneficios de un simulador.

**Dominio computacional:** Que son todos aquellos términos relacionados a la tecnología que se implementará, conocer un poco sobre que hace cada una y como podrían ayudarnos.

En el capítulo 2 se aborda una investigación rigurosa para el estado del arte de simuladores de exámenes, cada uno con un enfoque diferente, algunos con el fin de enseñar y promover el aprendizaje en aplicaciones web y otros enfocados a cursos o certificaciones bajo un tema en específico.

Asimismo, se presentan aquellas funciones relevantes y lo que las distingue del resto de aplicaciones, es decir, si tiene un formulario de registro, la estructura de los exámenes y contenido adicional tales como: videos, asesorías, etc.

En el capítulo 3 está centrado en la fase de planeación, que comienza con una recopilación de antecedentes analizando los diversos planes de estudios que maneja la FCC (Facultad de Ciencias de la Computación), a lo largo de este estudio nos daremos cuenta como es que la materia de Ingeniería de Software juega un papel importante en todos los planes, se profundiza en considerar los factores de riesgo al momento de desacreditar la asignatura.

La solución que se plantea en este capítulo es la que se ha citado con anterioridad, desarrollar un simulador web de calidad que sea un simulador de exámenes de la materia de Ingeniería de Software para el acompañamiento del alumno en su aprendizaje. Para lograr esto es necesario contar con ciertas habilidades técnicas, dominio de un lenguaje de programación orientado a la web y gestión de proyectos. Este último es importante ya que, si planteamos una estrategia y coordinamos la dependencia de tareas, el producto final (la aplicación web) será de calidad.

En este apartado se define el objetivo general y los objetivos específicos, que son metas más concretas en un plazo determinado de tiempo en cuanto a la realización de la aplicación, podemos decir que son la consecución del objetivo general.

Al finalizar este capítulo se aborda la metodología UWE (UML-Based Web Engineering) como la idónea para el modelado, análisis y diseño de la aplicación web que se propone, UWE utiliza UML como el lenguaje de modelado y que mediante la incorporación de estereotipos se proporciona una semántica distinta a la semántica de los diagramas UML en la Orientación a Objetos, es decir, se utiliza UML para el análisis y diseño formal de la aplicación web: mapas del sitio, storyboards, formularios, diseños de caja, etc., a través de diagramas UML.

El capítulo 4 marca el inicio de la metodología con un análisis de requerimientos donde se identifican todas aquellas funcionalidades que dispondrá la aplicación y quienes pueden llevar a cabo dicha función, para posteriormente llevar todos estos datos a un diagrama de casos de uso y sea mucho más intuitivo de comprender de manera más gráfica. Seguido de esto se explicará de forma breve los diagramas de actividades, que son la continuación de los casos de uso en su realización, en estos diagramas lo que encontraremos es una secuencia de pasos para finalizar una tarea, compuestos por un inicio y un fin, podremos visualizar datos de entrada y salida, condicionales, formularios, validaciones, etc. Adicionalmente, notaremos el uso de ciertos componentes que nos ofrece la metodología UWE.

En el capítulo 5 incluye la siguiente fase de diseño, por diseño hacemos referencia a la arquitectura de la aplicación web y esta contendrá un diagrama de contenido, que maneja un esquema compuesto por atributos y métodos, bastante similar a un diagrama de clases. Posteriormente, los diagramas de navegación que como su nombre lo indica presenta la forma en que un usuario puede navegar por las diversas interfaces y con qué tipo de contenido se

interactúa, estos enlaces pueden ser simples o con algún proceso asociado. Finalmente, los diagramas de presentación que sirven para indicar una forma clara y gráfica lo que se quiere hacer, donde cada diagrama es literalmente una página web, podemos colocar imágenes, texto, botones, listas, un menú de navegación, entre otros componentes y claramente haciendo uso de estereotipos que nos ofrece UWE.

El capítulo 6 está dedicado para la estructuración e implementación de la base de datos, se hace uso de herramientas online para la generación de diagramas. Es importante mencionar que se trata de una base de datos relacional, por lo tanto, primero se parte por un diseño conceptual que nos ayude a identificar cuáles serán nuestras entidades, atributos y llaves primarias, este diseño también puede ser llamado como entidad-relación. El diseño lógico se basa en el diagrama entidad-relación y está centrado en un modelo relacional, se destaca por representar datos en tablas haciendo uso de la cardinalidad y aplicando la normalización, de igual forma nos ayuda a prevenir errores y duplicidad de datos, mejorando la eficiencia de las consultas. Por último, el diseño físico trata de pasar las tablas a un sistema gestor de base de datos a través de un lenguaje de consulta como lo es SQL, asimismo presento las operaciones básicas que suelen manejarse en una base de datos, todo esto a través de WampServer que actúa como servidor virtual y nos ayudará en la ejecución de la aplicación.

Finalmente, en el capítulo 7 se entra en un proceso de implementación y pruebas, que consiste en el desarrollo de la aplicación dividido en dos partes como lo es el frontend y el backend, siendo algo muy similar al modelo cliente-servidor. Del lado del frontend se utiliza el framework ReactJS que aborda tecnología como diseño en CSS, programación con JavaScript y etiquetas HTML la importancia que tiene utilizar este marco de trabajo es que podemos renderizar componentes mezclando estas 3 tecnologías. Para el backend lo tenemos orientado al framework Python Flask, se menciona la sencilla forma que es su implementación y todas aquellas librerías que nos permiten tener comunicación con el frontend, para esto adjunto un breve ejemplo de una petición a una función. Para el apartado de pruebas está enfocado en una recopilación de las páginas más relevantes en la aplicación con una breve explicación de lo que consiste, se menciona las posibles limitantes a considerar para su uso.

De esta forma podemos garantizar un simulador de exámenes sofisticado y eficiente, aplicando gran parte de mis conocimientos que he adquirido en esta carrera y recurriendo a recursos adicionales que complementen tanto la escritura de la tesis como la programación. A lo largo del presente trabajo de tesis para cada uno de los capítulos se utilizan diversas herramientas que nos facilitan la carga de trabajo las cuales se hacen mención en su correspondiente capítulo.

## Capítulo 1: Marco Teórico

El marco teórico que se desarrolla a continuación permite conocer los conceptos básicos necesarios para el entendimiento de la problemática a resolver y del mismo modo definir una estrategia que nos permita aplicar el dominio computacional en la práctica. En este capítulo se presenta una consulta teórica que alimentara de información cada fase de desarrollo tanto en la investigación como en la implementación.

### 1.1 Dominio del problema

Actualmente el sistema educativo nivel licenciatura ofrece programas educativos que incluyen actividades planificadas semánticamente por medio de contenidos temáticos, se explican cuáles son los objetivos de aprendizaje, se menciona la metodología de enseñanza y los modos de evaluación. En estos últimos la acreditación es un tema para destacar, dado que muchas materias están estrechamente relacionadas entre sí, donde no tratarse en tiempo y forma puede ocasionar un recursamiento.

Un recursamiento es equivalente a desaprobar una materia o módulo, es no haber cumplido con las calificaciones mínimas necesarias para regularizar la materia, lo que significa que el alumno la debe de volver a cursar en el siguiente periodo. Esto a mayor frecuencia puede perjudicar el desempeño académico de un alumno e incluso truncar una carrera universitaria.

El desempeño académico generalmente hace referencia a la evaluación del conocimiento adquirido en el ámbito universitario, puede verse reflejado obteniendo buenas calificaciones en exámenes. Sin embargo, llevar un mal ritmo de aprendizaje puede generar que el desempeño vaya disminuyendo, llegando incluso a perder la materia.

Rezago educativo es una condición de atraso en donde las personas no han alcanzado el nivel educativo que esperan, por lo que, tener una ruta de aprendizaje optima mejoraría dicha situación. Generalmente disponer de una herramienta que nos acompañe durante el aprendizaje es muy recomendado para circunstancias donde se requiera mejorar el rendimiento académico.

Una ruta de aprendizaje es un camino por seguir para que una persona obtenga aprendizaje necesario para alcanzar con éxito un resultado, puede ser considerado como estímulo para evitar deserción y reforzar el compromiso en la medida en que se avanza.

Existen simuladores basados en tecnología web que aprovechan los servicios de internet para la simulación de entornos virtuales a través de un navegador web. En la educación los simuladores proveen material didáctico y audiovisual con el cual una persona puede hacer uso de este contenido y estimular su interés en aprender, tales como simuladores de examen. Estos simuladores son muy importantes hoy en día, nos ofrecen muchos beneficios pueden

mejorar el rendimiento académico, evitar recursamiento y disminuir el rezago educativo, de esta forma generan un impacto significativo en la sociedad, personas autodidactas que fortalecen sus conocimientos en un área o disciplina. Es dar un paso más en la educación adaptando nuevas tecnologías, una herramienta que este a disposición de los alumnos y que con entrenamiento puedan abrir muchas puertas en su vida profesional.

## 1.2 Dominio computacional

Aplicaciones web: Aplicación cliente-servidor, donde tanto el cliente (el navegador) como el servidor (servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados e intercambian información. El cliente web es un programa que interactúa con el usuario para solicitar a un servidor web el envío de recursos. El servidor web es un programa que está esperando las solicitudes por parte de un cliente web [1].

Modelo Cliente-Servidor: En las aplicaciones web, este modelo las tareas se reparten entre dos roles: un proveedor que proporciona recursos o servicios (llamado Servidor) y un consumidor (llamado Cliente) que contacta al servidor con el objetivo de hacer uso de los recursos que este provee.

Base de datos: Colección organizada de información almacenada para posteriormente acceder a esos datos de forma rápida y estructurada. El modelo relacional es un ejemplo de un modelo basado en registros. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro [2].

Python: Lenguaje de programación interpretado de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

Framework web: Herramienta de desarrollo web que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías o funcionalidades ya creadas.

HTML5: Lenguaje de marcado donde se estructura la información de una aplicación web, proporciona una manera simple de manipular gráficos en un navegador web sin recurrir a complementos como Flash, ofrece métodos para insertar texto, audio y video en páginas web [3].

CSS (Cascading Style Sheets): Es un lenguaje para la composición y estructuración de páginas web (HTML o XML). Contiene elementos de codificación y se compone de estas “hojas de estilo en cascada” que también se llaman archivos CSS (.css). Puede diseñar cualquier elemento HTML para cambiar sus dimensiones, colores, bordes, espaciado, etc., sino que ahora también puede agregar transiciones y transformaciones animadas a sus páginas web, usando solo unas pocas líneas de CSS [3].

**PHP:** Tecnología de lado de servidor con la que se han implementado más servidores en Internet, es multiplataforma y se integra normalmente con Apache y MySQL. Tiene control ilimitado sobre su servidor web, ya sea que necesite modificar HTML realizar sesiones o agregar detalles de usuario a una base de datos, puede hacerlo todo desde los mismos archivos PHP en los que reside el HTML [3].

**JavaScript:** Lenguaje de programación interpretado que habitualmente se utiliza en sitios web para ejecutar acciones del lado del cliente. Proporciona un medio para la interacción dinámica, recoge información de entrada por el usuario y la manda al servidor para ser procesada, como verificar la validez de un correo electrónico en los formularios [3].

**API:** Conjunto de reglas que determinan como las aplicaciones o los dispositivos pueden conectarse y comunicarse entre sí. Mientras que una API REST como el método más común para conectar componentes en la arquitectura de microservicios, proporciona una forma flexible y ligera de integrar aplicaciones [4].

**Frontend:** Es la parte del desarrollo web dedicada a la parte frontal de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños y efectos. Además de ser la parte de la página con la que interactúan los usuarios, es todo el código que se ejecuta en el navegador. Trabaja con lenguajes como: HTML, CSS, JavaScript [5].

**Backend:** Procesa información que alimentara al frontend de datos, es la lógica tecnológica que hace que una página web funcione. Exige el dominio de otros términos de programación, lenguajes que requieren de lógica ya que esta área se encarga de optimizar recursos, de la seguridad y otros factores [5].

## Capítulo 2: Estado del Arte

En este capítulo se aborda una investigación a diversas herramientas que facilitan el aprendizaje de las personas por medio de una aplicación web, siendo el objetivo principal: poder acreditar un examen para un área en específico. Las presentes aplicaciones generalmente son comerciales, por lo que el simulador de exámenes a desarrollar es de uso libre y está disponible para la comunidad de la facultad.

**Tema:** Simulación de exámenes en diversas áreas de conocimiento.

**Estado del Arte:** Recopilación de simuladores de exámenes en línea comerciales o de uso libre en internet.

### 2.1 UNITIPS

Nombre	Autores	Año	Descripción	Referencia
UNITIPS	Empresa UNITIPS	2015	Plataforma de preparación para el ingreso a la universidad. Cuenta con exámenes de simulacro para la UNAM, IPN, Ceneval EXANI II, entre otros más.	<a href="https://www.unitips.mx/">https://www.unitips.mx/</a>

Para acceder es necesario realizar un registro e indicar que curso nos interesa. En esta vista podemos observar algunos de los beneficios de la prueba gratis, existen diferentes planes que son comerciales, donde el contenido y asesoramiento es aún mayor. Generalmente aplica para exámenes de admisión, sin embargo, también va dirigido a personas que deseen reforzar sus conocimientos.

The image shows a registration form on the left and a benefits list on the right. The form is titled "Crea una cuenta" and includes fields for "Nombre Completo", "Correo Electrónico", "Contraseña", "Celular (10 dígitos)", and a dropdown for "Elige tu Curso". Below the form is a disclaimer: "Al registrarte estás aceptando los **Terminos y Condiciones** y el **Aviso de privacidad**". A blue "Crear Cuenta" button is at the bottom. The right side features a testimonial from a user: "Unitips me ayudó a pasar mi examen en el primer intento." Below it, another testimonial: "Me hizo más fácil el estudio ya que tenía todos los recursos en la plataforma." The main heading is "Crea tu cuenta y accede sin costo a:", followed by a list of benefits with green checkmarks: "Guía UNAM, CENEVAL, IPN, UAM y mucho más", "3 lecciones de prueba", "Ejercicios de práctica", "Examen diagnóstico", and "Becas y beneficios exclusivos".

Ilustración 2.1 UNITIPS - Formulario de registro (<https://www.unitips.mx/accounts/signup/>)

Una vez indicado el curso a realizar, nos ofrece una interfaz muy amplia en el que destacan: lecciones, tutorías, puntuaciones, porcentaje de avances por cada examen/materia. Otro rasgo por destacar son los Lives, que son videos grabados por los mejores maestros de México, suelen subir nuevos videos por cada semana, los meses de vigencia depende mucho del plan comercial.

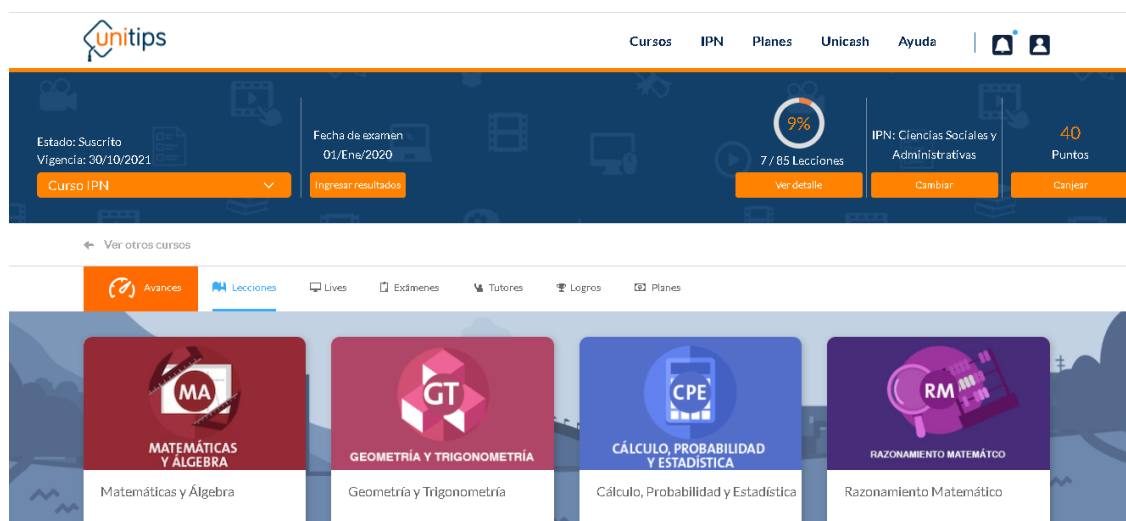


Ilustración 2.2 UNITIPS - Elección de cursos (<https://ayuda.unitips.mx/hubfs/image-png-Aug-13-2020-03-32-13-80-PM.png>)

Dentro del curso que estemos tomando también podemos tener acceso a videotutoriales de algún tema en específico, son videos de no más de 10 minutos, son breves y claramente explicados. Al final nos ofrece comunicación con el profesor en caso de dudas.



Ilustración 2.3 UNITIPS - Contenido multimedia (<https://www.unitips.mx/>)

## 2.2 AppMisión


Nombre	Autores	Año	Descripción	Referencia
appmision	Empresa appmision	2018	Simulador de exámenes comercial enfocado en admisión a la universidad para diferentes áreas de conocimiento.	<a href="https://appmision.com/">https://appmision.com/</a>

Para hacer uso de esta aplicación web es necesario realizar un registro. Los privilegios de esta herramienta son acceso de por vida, exámenes exclusivos, es decir, exámenes con un propósito en específico por ejemplo acreditar un examen TOEFL. El costo depende del tipo de examen que desees practicar.

### Regístrate

Únete al movimiento: [#ceroExámenesReprobados](#)

0



Las contraseña debe tener por lo menos 6 caracteres




Ilustración 2.4 AppMisión - Formulario de registro (<https://appmision.com/register>)

Posteriormente podemos hacer uso de un catálogo de cursos con un costo asociado. Y para garantizar una mejor calidad, cada examen cuenta con estrellas de valoración, las cuales indican si la comunidad lo recomienda y el número de personas que han utilizado dicho examen, así como los comentarios.

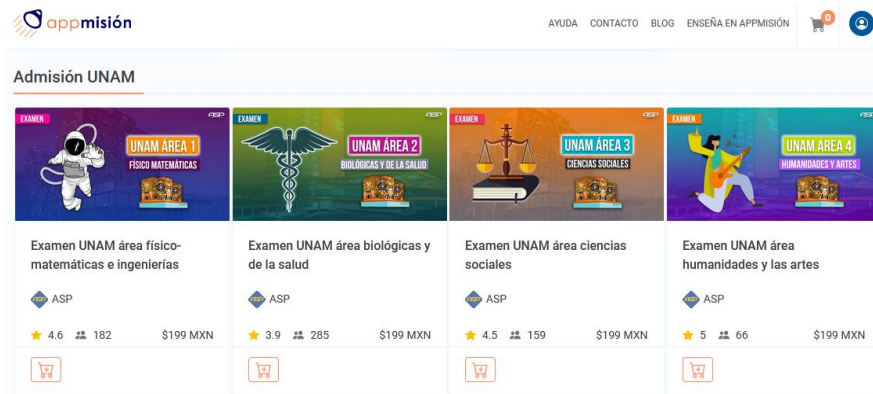


Ilustración 2.5 AppMisión - Elección de examen como usuario registrado (<https://appmision.com/login>)

Realizar un examen es como siempre esperamos que sea, enunciado y sus respuestas de opción múltiple, puede incluir gráficos y una barra de avance, esta barra nos muestra el progreso en el examen, señalando aquellas preguntas respondidas y las que faltan por responder, además de disponer de un cronómetro. Una vez realizado un examen, muestra un resumen estadístico de cada tema, marcando aquellas asignaturas donde el alumno tiene dificultades o donde obtuvo un resultado sobresaliente. La siguiente Ilustración 2.6 muestra la simulación vista desde un dispositivo móvil, la experiencia de usuario es un requisito hoy en día para cada simulador.

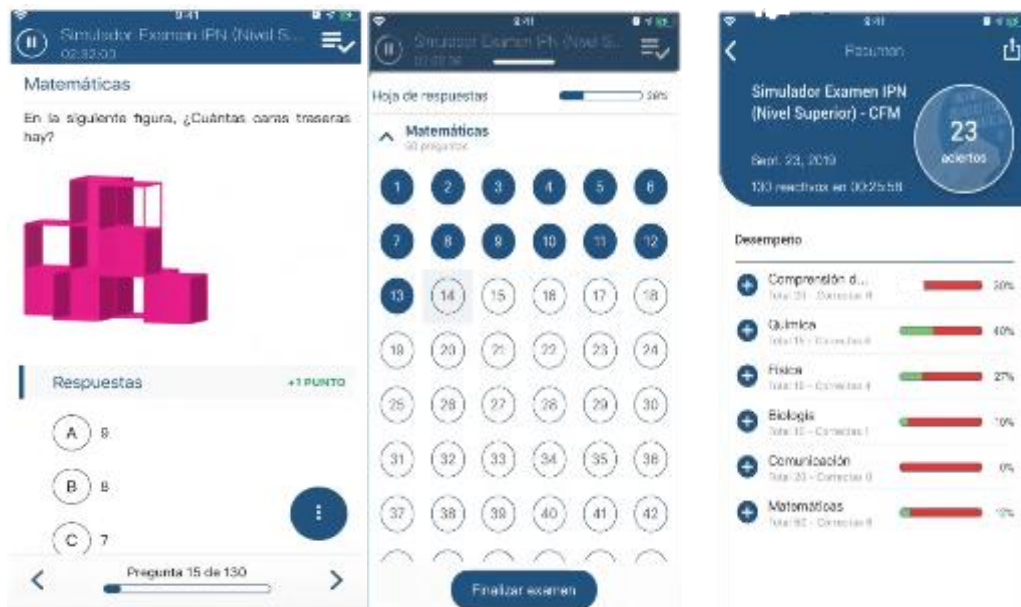


Ilustración 2.6 AppMisión - Examen en ejecución desde Google play ([https://play.google.com/store/apps/details?id=mx.aspit.appmision&hl=es\\_US&gl=US](https://play.google.com/store/apps/details?id=mx.aspit.appmision&hl=es_US&gl=US))

## 2.3 QuizPM

Nombre	Autores	Año	Descripción	Referencia
QuizPM	Empresa QUIZPM	2015	Simulador de examen PMP adaptado a la guía PMBOK 6. Ofrece 3 modos de estudio, control de progreso y mejora la experiencia del usuario.	<a href="https://quizpm.com/">https://quizpm.com/</a>

Para hacer uso del simulador es necesario tener un registro, y estos son algunos de los servicios que ofrece una prueba gratuita. Sus principales ventajas de utilizar esta herramienta son: aprendizaje a medida, control de progreso, experiencia de usuario. Incluye preguntas en español e inglés, elaboradas y revisadas por profesionales certificados. En sus planes comerciales de suscripción cuentan con soporte online con 3 exclusivas modos de estudio.

The image shows two side-by-side screenshots from the QuizPM website. The left screenshot is a registration form titled 'Regístrate gratis' (Register for free). It asks for an email address, name, surnames, password, and password confirmation. There is a 'REGISTRARME' button and a link for existing users. The right screenshot shows a 'TRIAL' offer for 0€. It lists benefits: 1 test de prueba, 50 preguntas, and estadísticas personalizadas. There is a 'PROBAR' button.

Ilustración 2.7 QuizPM - Formulario de registro y beneficios  
(<https://app.quizpm.com/account/register#>)

La siguiente estructura representada en la Ilustración 2.8 es como se forman los exámenes dentro del simulador, podemos observar que en la barra superior cuenta con una barra de navegación que hace referencia a nuestra cuenta. La barra lateral muestra opciones como el cronómetro, el modo de estudio, fijar preguntas, salir del examen. Finalmente, las preguntas con opción múltiple cubren el resto de la pantalla que de igual forma pueden ser de forma gráfica, se puede desplazar de pregunta.

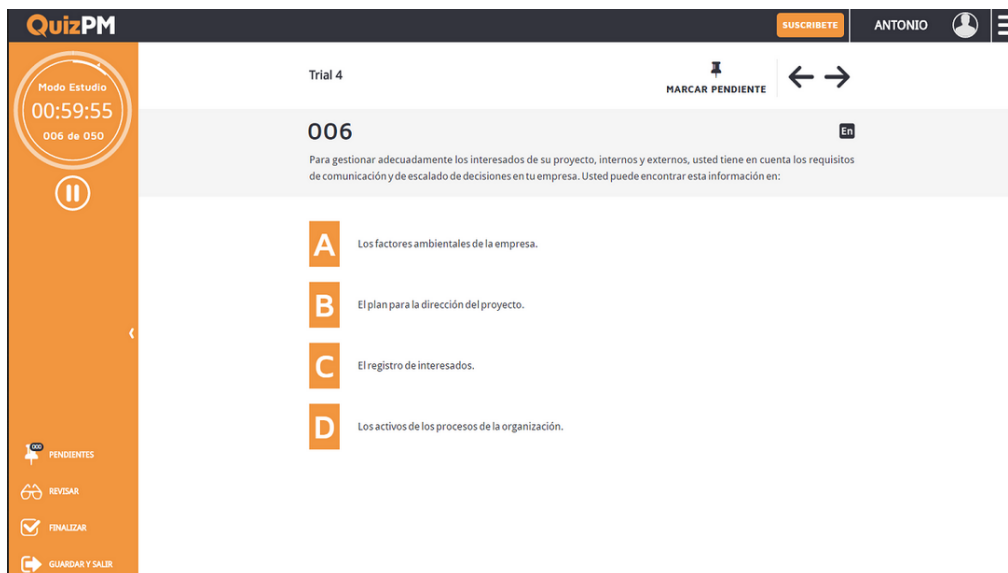


Ilustración 2.8 QuizPM - Examen en ejecución  
(<https://www.antoniomartel.com/tag/pmp>)

Una vez realizado un examen muestra un resumen estadístico de preguntas acertadas, la duración y un promedio aritmético. Cada pregunta puede estar catalogada y de esta forma tenemos una mejor gestión de nuestros resultados, otro dato a señalar es que cada examen se va guardando en un historial, con el fin de que se vean las mejoras por el usuario.



Ilustración 2.9 QuizPM - Resultado estadístico al final de cada examen  
(<https://www.antoniomartel.com/tag/pmp>)

Finalmente, las principales diferencias entre estos simuladores y la propuesta de tesis son: como mejora, el alumno cuenta con un apartado de retroalimentación posterior al examen para que estudie sus posibles errores, el profesor cuenta con una búsqueda más ágil al momento de consultar los alumnos. Cambia la forma de ejecución de un examen al no contar con un cronómetro o un tiempo límite para su finalización (por ser herramienta de acompañamiento), esto normalmente ocurre en certificaciones y evaluaciones.

## Capítulo 3: Planeación

Planificar es una actividad importante en cuanto al proceso de toma de decisiones para alcanzar metas en un futuro deseado, teniendo en cuenta la situación actual y todos aquellos factores que pueden influir en alcanzar los objetivos. En el presente capítulo partimos del origen de la problemática en términos generales, mostrando la necesidad de diseñar una herramienta que ayude a progresar el desempeño académico, posteriormente planteamos una estrategia de solución y definimos nuestros objetivos que serán el reflejo del producto final. Al término del capítulo se detalla la forma en que se aplicará la metodología UWE basada en UML para tener un análisis gráfico y mejor detallado de lo que se pretende diseñar, al igual que el modo de implementación.

### 3.1 Antecedentes

Uno de los principales motivos por el cual se plantea este proyecto es debido al gran número de aspirantes que han recurrido la materia de Ingeniería de Software y el inminente rezago educativo que esto puede generar. Por lo que, acreditar esta materia puede ser de vital importancia si el alumno requiere terminar la carrera en tiempo y forma.

Como se sabe, la facultad consta de 3 programas educativos los cuales tienen materias afines a la Ingeniería de Software e incluso con nombres iguales y con contenidos temáticos muy similares pues se adecúan al perfil de egreso del alumno, haciendo de esta una de las más importantes en el plan de estudio. Lo que nos lleva a imaginarnos ¿Cuál es el riesgo de recurrir la materia? Muchas veces nosotros los alumnos pensamos que retomar la materia el siguiente periodo pueda ser la solución idónea, lo cierto es que para cada plan de estudio la problemática se presenta de diversas formas con rutas críticas distintas. Para este análisis hago referencia a la página de secretaría académica FCC [6].

En Ingeniería en Ciencias de la Computación, la materia se ubica en el quinto periodo en su correspondiente mapa gráfico como se muestra en la Ilustración. 3.1. A simple vista pareciera que no tiene una ruta de aprendizaje, pero si profundizamos un poco más podemos concluir que los fundamentos que provee la materia son aplicados en cualquier área de las ciencias de la computación, donde su función principal es crear software de calidad que es implementada a través de una metodología. De esta forma, es fundamental acreditar la materia para así posteriormente desarrollar algún proyecto y plantearle una solución.

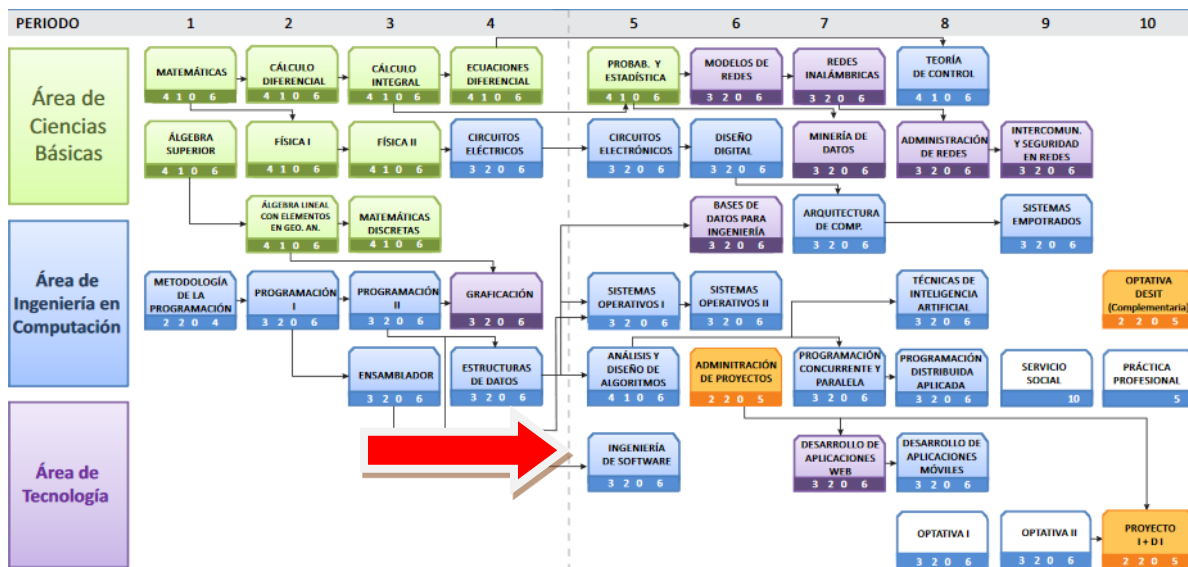


Ilustración 3.1 Programa educativo de Ingeniería en Ciencias de la Computación.  
[https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapa\\_ICC%202017\\_071217.pdf](https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapa_ICC%202017_071217.pdf)

En cuanto a las optativas en el correspondiente programa educativo dicho anteriormente, podemos observar en la Ilustración. 3.2 que hay una dependencia que hace referencia a esta materia de Ingeniería de Software. De este modo, acreditarla se vuelve cada vez más una necesidad, dado que lo aprendido sirve como sustento o base para las materias que están próximas.

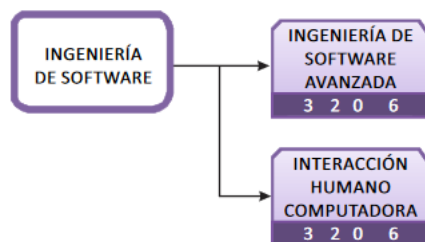


Ilustración 3.2 Optativas de Ingeniería en Ciencias de la Computación.  
[https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapa\\_ICC%202017\\_071217.pdf](https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapa_ICC%202017_071217.pdf)

En Licenciatura en Ciencias de la Computación, de igual forma se presenta en el quinto periodo acorde al mapa gráfico como se marca en la Ilustración. 3.3. Y se suscita el mismo caso, donde la materia más allá de no tener una secuencia, su importancia genera un gran impacto en diversas áreas de las ciencias de la computación. Se vuelve necesario el aplicar lo aprendido en la materia para así desarrollar proyectos de mayor calidad y eficiencia.

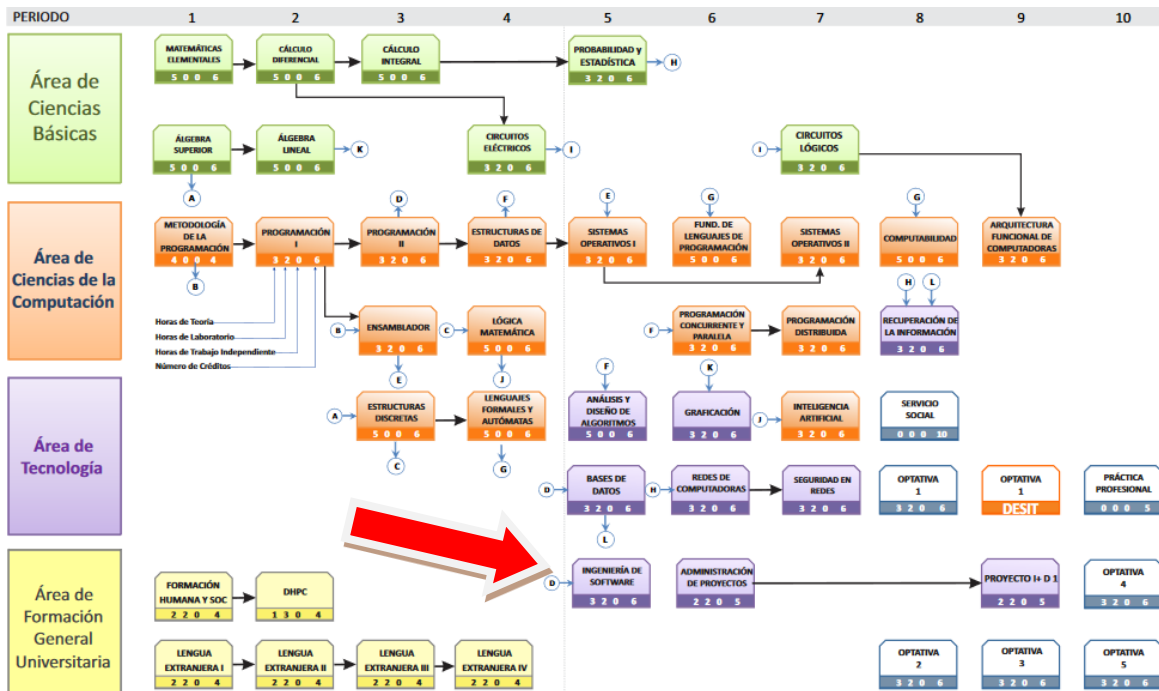


Ilustración 3.3 Programa educativo de Licenciatura en Ciencias de la Computación.  
 ([https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapas\\_Carreras\\_LCC\\_040821.pdf](https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapas_Carreras_LCC_040821.pdf))

En el caso de las optativas tenemos la siguiente Ilustración. 3.4, en este programa educativo únicamente cuatro materias son las que le dan un seguimiento, volviéndose una materia de mayor relevancia. Cabe señalar que a medida que el alumno se acerca a la recta final de su carrera deberá elegir un modo de titulación y en el mayor de los casos está presente la aplicación de ingeniería de software.

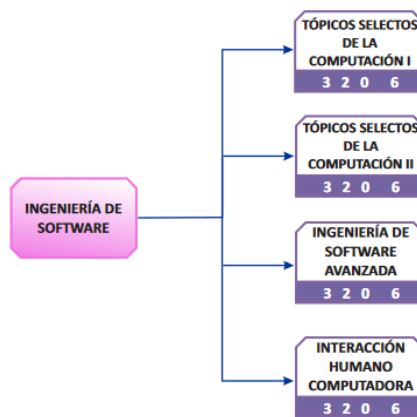


Ilustración 3.4 Optativas de Licenciatura en Ciencias de la Computación.  
 ([https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapas\\_Carreras\\_LCC\\_040821.pdf](https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapas_Carreras_LCC_040821.pdf))

En Ingeniería en Tecnologías de la Información, la materia se presenta en el cuarto periodo, su mapa gráfico se señala en la Ilustración. 3.5; podemos observar que, en este programa educativo, ingeniería de software da pie a una gran cantidad de materias y el simple hecho de recurrir a la materia retrasa enormemente el desempeño académico del alumno.

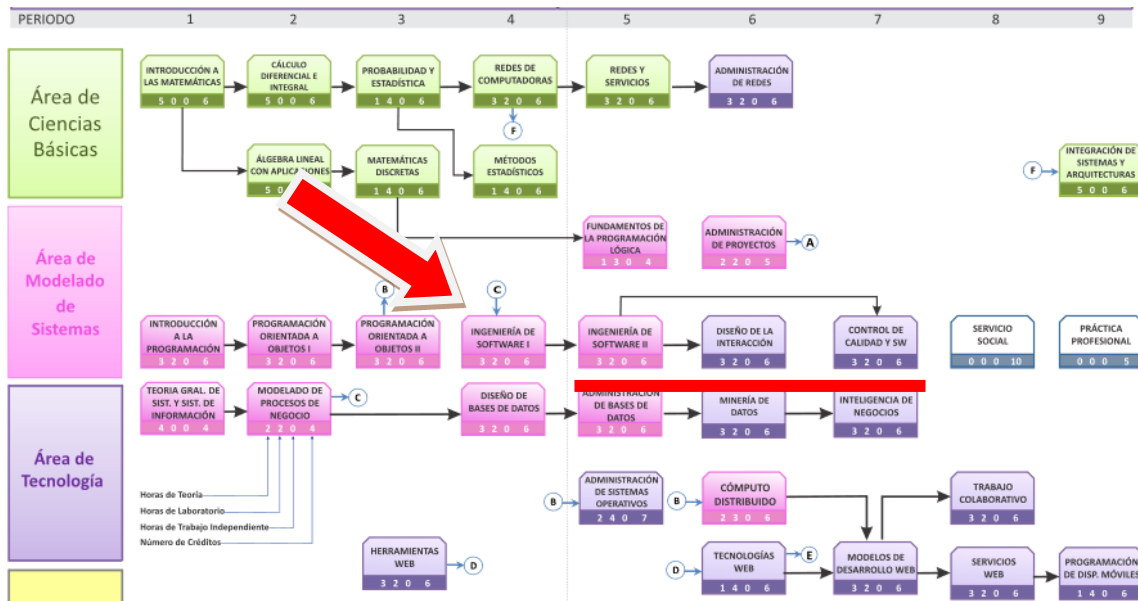


Ilustración 3.5 Programa educativo de Ingeniería en Tecnologías de la Información.  
([https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapas\\_Carreras\\_ITI\\_v2802.pdf](https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapas_Carreras_ITI_v2802.pdf))

Para sus optativas en este programa educativo tenemos como referencia la Ilustración 3.6. En este caso estas dos materias son las que dependen de Ingeniería de software II, pero en si Ingeniería de software I es quien da el inicio a esta secuencia. Por este modo, es indispensable acreditarla en tiempo y forma.



Ilustración 3.6 Optativas de Ingeniería en Tecnologías de la Información.  
([https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapas\\_Carreras\\_ITI\\_v2802.pdf](https://secreacademica.cs.buap.mx/MumMapas/Semestres/Mapas_Carreras_ITI_v2802.pdf))

Por otro lado, para el perfil de egreso de un licenciado o ingeniero es tan importante la ingeniería de software porque los conocimientos y habilidades que ofrece dicha materia ayudan a formar a una profesionista capaz de trabajar en equipo, diseñar soluciones, administrar proyecto, metodología y técnicas para el desarrollo del software. Además de generar actitudes como: liderazgo, perseverancia en solución de problemas, afán de superación, responsabilidad, conducta emprendedora e innovadora, entre otros.

Por último, otra problemática por mencionar es la falta de interés, los alumnos buscan un aprendizaje más práctico actualmente y este simulador puede agilizar el proceso de enseñanza, mezclando la teoría vista en clase con lo práctico que es su uso. Se prevé que, con un uso adecuado de esta propuesta, el alumno que utiliza el simulador para ejercitarse en los

conocimientos teóricos y prácticos del área de la ingeniería de software, incremente sus probabilidades de acreditar las materias que se relacionan con dicha área.

### **3.2 Estrategia de solución**

La idea general de este proyecto es desarrollar una herramienta de aprendizaje que sea capaz de solucionar un problema que cada vez se ve más reflejado en la comunidad de la Facultad de Ciencias de la Computación, como lo es el recursamiento en la materia de Ingeniería de Software y el posible rezago académico que puede tener un alumno.

Un simulador de exámenes para dicha materia donde el alumno tendrá la oportunidad de retroalimentarse, practicar para futuros exámenes departamentales y ser más autodidactas, que con un uso adecuado puedan garantizar una acreditación satisfactoria y enriquecer su perfil de egreso como ingenieros o licenciados. Además, el profesor que imparte la materia puede utilizar el simulador como un complemento a sus clases y promover un uso más frecuente a esta tecnología, le puede ayudar a observar que alumnos requieren de una mejor atención y ofrecer asesoría. Adicional a esto podemos asociar resultados estadísticos en cuanto al rendimiento del alumno, ya que cada examen está catalogado por un temario y preguntas formuladas de manera aleatoria.

Para el desarrollo del presente trabajo se ha decidido implementar el sistema propuesto utilizando la metodología de diseño de reactivos, estructura y formato que utiliza el Centro Nacional de Evaluación para la Educación Superior (CENEVAL). Este organismo es una institución encargada de elaborar exámenes y pruebas de conocimiento en México para distintas disciplinas, entre ellas la de Computación. Los exámenes que diseña preparan, desarrolla y aplica el CENEVAL tienen validez oficial en todo el país y con ellos universidades públicas y privadas adoptan este proceso de evaluación como parte del abanico de formas de titulación que ofrecen a sus egresados, entre ellas se encuentra nuestra Benemérita Institución y la Facultad de Ciencias de la Computación ofrece esta alternativa de titulación a sus egresados y pasantes. Esta adopción hará que nuestra propuesta esté muy cercana a la manera en que un egresado de la facultad que opta por presentar el examen de CENEVAL encontrará al realizarlo. Si el sistema que se propone es utilizado por dicho egresado como entrenamiento en sus materias de ingeniería de software que cursó en la licenciatura o Ingeniería en Ciencias de la Computación, seguramente incrementará las posibilidades de obtener el puntaje deseado para su titulación pues los exámenes que aplica CENEVAL son en el área de Sistemas Computacionales que tienen una alta carga de reactivos relacionados con la Ingeniería de Software.

En el presente trabajo de tesis resalto las partes más esenciales en cuanto al análisis y desarrollo del proyecto, tales como: profundizar un poco más en la problemática considerando los posibles factores de riesgo en cuanto a los planes de estudios que ofrece la facultad, adecuarnos a una metodología enfocada al desarrollo de aplicaciones web con el uso de tecnologías que pueden proporcionarnos una mejor experiencia de usuario, modelado

apropiado a los diagramas a implementar para que sea sencillo interpretar la forma en la que funciona el simulador, así como los roles que tienen los usuarios al disponer de la herramienta y la definición de nuestros objetivos que establecen una meta a alcanzar. Por otro lado, se menciona los beneficios y el impacto social que puede generar la realización de este simulador de exámenes, sus aportaciones que pueden ser de utilidad no solo para la comunidad de la facultad, sino que para otras áreas de conocimiento a través de alguna adaptación y posibles cambios.

### 3.3 Objetivos

#### Generales

Implementar una aplicación web que sea capaz de generar exámenes de la materia de “Ingeniería de Software” para que los alumnos que estudian dicha asignatura se ejerciten con la aplicación.

#### Particulares:

- El sistema deberá generar exámenes de la materia de Ingeniería de Software por temas.
- Cada examen contendrá una serie de preguntas asociadas a los temas a evaluar.
- Los exámenes generados por la aplicación estarán constituidos de una serie de preguntas elegidas al azar de un banco almacenado en una base de datos.
- Todas las preguntas tendrán respuestas de opción múltiple. Cada pregunta tendrá 4 posibles respuestas donde una de ellas será la correcta y el resto serán respuestas incorrectas.
- El diseño de las preguntas se hará conforme al formato y estructura que utiliza el CENEVAL para la creación de sus reactivos.
- El alumno se deberá registrar y autenticar, una vez esto, podrá seleccionar el tema en el que se quiere evaluar.
- El sistema deberá mostrarle al alumno un resumen estadístico de los exámenes que ha realizado con el sistema y que consiste en la siguiente información:
  - El número de respuestas correctas que tuvo en el examen que realizó.
  - El número de respuestas incorrectas que tuvo en el examen que realizó.
  - Si reprobó o no el examen.
  - La fecha de realización del examen.
  - El número de veces que ha realizado dicho examen.
- La aplicación tendrá también un administrador que será el profesor de la materia quien podrá actualizar los temas, preguntas y respuestas (agregar, eliminar o modificar), así como tener el control de los alumnos registrados en el sistema. El profesor también deberá registrarse y autenticarse para usar la aplicación.
- Toda la información con la que trabaje la aplicación tanto del administrador como de los usuarios deberá estar en una base de datos relacional.

### 3.4 Metodología

Para el desarrollo e implementación de este simulador de exámenes se pone a disposición el uso de la metodología UWE (UML-Based Web Engineering) que permite especificar de mejor manera una aplicación web en su proceso de creación, mantiene una notación estándar basada en el uso de UML (Lenguaje Unificado de Modelado) para sus modelos y sus métodos, lo que facilita la transición.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario [7].

Con respecto a la metodología me enfocare en 4 fases primordiales para llevar a cabo la implementación del simulador de exámenes.

1. Análisis de requerimientos
2. Diseño Web.
3. Implementación.
4. Ejecución y pruebas.

#### 3.4.1 Análisis de requerimientos

En esta primera fase, una de las primeras actividades en la construcción de aplicaciones web es la identificación de los requisitos, y para ellos en UWE se especifican mediante el modelo de requerimientos, donde involucran el diagrama de casos de uso y el diagrama de actividades en base a UML.

##### *1. Diagrama de casos de uso*

El diagrama de casos de uso está conformado por los elementos actor y casos de uso. Los actores se utilizan para modelar los distintos usuarios de la aplicación web, por ejemplo: empleado, alumno, instructor, cliente... que pueden interactuar con el mismo. Los casos de uso se utilizan para visualizar las diversas funcionalidades que la aplicación proporciona, como lo es: crear nuevo usuario, realizar pedido, actualiza cuenta, borrar producto, entro otros más [8].

En la siguiente Ilustración. 3.7 se muestra un ejemplo del diagrama de casos de uso para una agenda de contactos. Cabe señalar que, para cada fase de la metodología, UWE provee de diferentes estereotipos, que son una forma de representar los procesos a través de iconos.

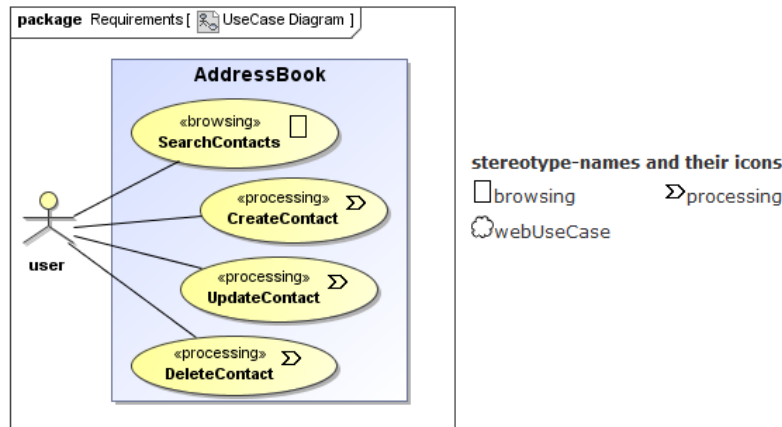


Ilustración 3.7 Ejemplo diagrama de casos de uso y sus estereotipos [9].  
 (<https://uwe.pst.ifi.lmu.de/teachingTutorialRequirementsSpanish.html>)

En la anterior Ilustración 3.7 se observa al actor “user” quien puede realizar operaciones como “buscar contacto”, “crear contacto”, “actualizar contacto” y “borrar contacto” dentro de la agenda. Cada caso de uso trae consigo un icono como anteriormente se mencionó y un encabezado donde nos indica si los datos de la aplicación son modificados o no.

Significado de los estereotipos en base al diagrama anterior en Ilustración 3.7:

- <<browsing>> representa que los datos son leídos y presentados al usuario. En el caso de uso “buscar contacto” modela la búsqueda de contactos y son presentados.
- <<processing>> representa los cambios de los datos, por ejemplo, en los casos de uso “crear contacto”, “borrar contacto” y “actualizar contacto”, realizan cambios en el sistema, algunos son ejecutados, creados, modificados o eliminados valga la redundancia.

## 2. Diagrama de actividades

En algunas ocasiones no es posible capturar toda la información de una aplicación presentada a través de un diagrama, por ello cada caso de uso planteado debe ser tratado de una forma más detallada para así ofrecer un mejor panorama del proceso por el cual pasan los datos.

El diagrama de actividades nos va a mostrar que es lo que sucede durante un caso de uso, tiene una secuencia similar a los diagramas de flujo. Las actividades son parte de un caso de uso y forman parte del modelado, algunos de los componentes básicos que conforman este diagrama son: nodo inicial, nodo final, acciones, flujo de control, entre otros.

La siguiente Ilustración. 3.8 podemos observar la interpretación de un caso de uso llamado “Crear contacto”. El diagrama comienza en el nodo inicial y se dirige a la acción “Crear formulario” indicando que es de tipo formulario, marcado con el estereotipo <<displayAction>> el cual modela el contenido visto al usuario. Este formulario presenta los datos que son requeridos tales como: nombre, email, dirección, teléfono.

La siguiente acción “Ingreso de datos” es la secuencia del formulario, marcado con el estereotipo <<userAction>> que indica la interacción usuario-aplicación web, en este caso, el usuario proporcionando los datos que son procesados y validados en este paso.

Los datos cuando son validados por el sistema pasan a la acción “Guardar contacto” que es la confirmación del contacto, marcado con el estereotipo <<systemAction>> que consta de la ejecución por parte del sistema. En este punto procede a ser almacenado y con ello termina la actividad “Crear contacto”.

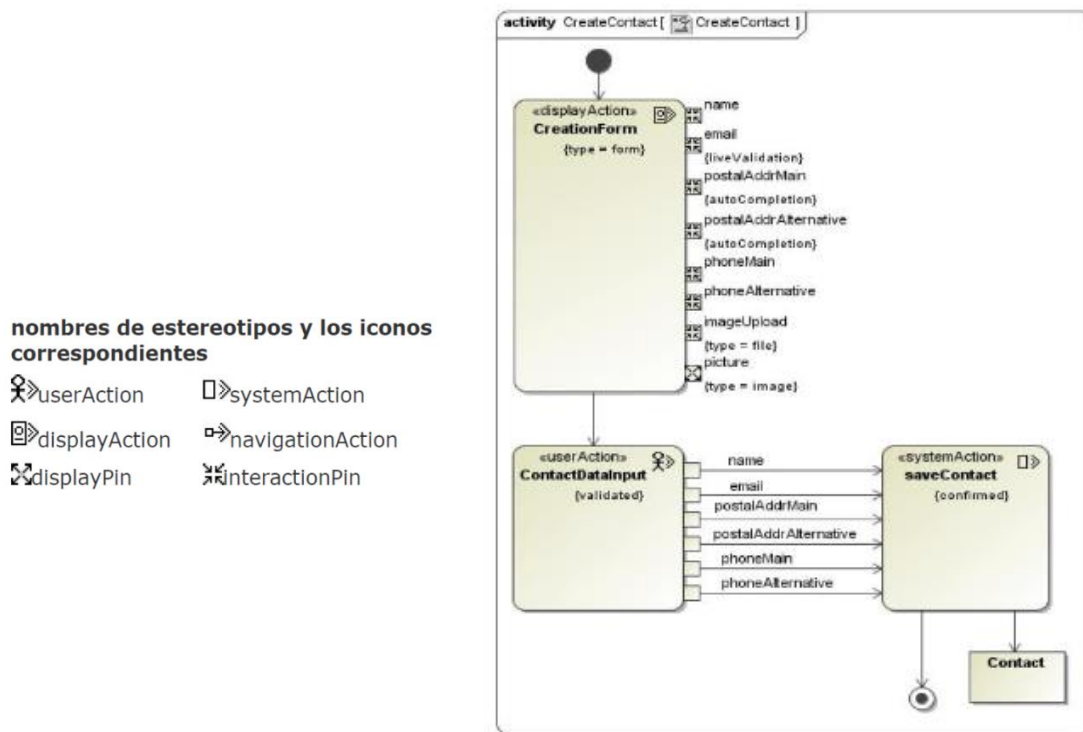


Ilustración 3.8 Ejemplo diagrama de actividades y sus estereotipos [9].  
<https://uwe.pst.ifi.lmu.de/teachingTutorialRequirementsSpanish.html>

Significado de los estereotipos en base al diagrama anterior de la Ilustración 3.8:

- <<displayAction>> determina cuales son los datos que son presentados al usuario, por ejemplo, se presentan los campos: nombre, email, dirección, etc.
- <<userAction>> indica las interacciones del usuario en la página, por ejemplo, la actividad “ContactDataInput” recopila la información que proporciona el usuario.
- <<systemAction>> describe las acciones que son ejecutados por el sistema, podemos verlo reflejado en la actividad “saveContact” donde crear y almacena el contacto.
  - <<interactionPin>> modela la entrada de datos.
  - <<displayPin>> modela la salida de datos.
  - <<navigationAction>> modela operaciones de navegación y elementos de presentación.

En resumen, en esto consiste la fase 1 de la implementación de esta metodología; análisis, datos, restricciones, operaciones, son algunos de los conceptos que mayor resaltan en este nivel. Se pretende explorar todo lo que nos ofrece UWE en esta fase para hacerlo más entendible hacia el usuario final.

### 3.4.2 Diseño web

Para esta segunda fase una vez determinada el análisis de requerimientos se proceder a la fase de diseño siendo dividida en dos partes, ambas enfocadas al ambiente gráfico. Se crea un modelo de contenido (basado en UWE) donde el objetivo es proporcionar una especificación visual de la información en el dominio relevante de la aplicación web, dicho de otra forma, se modela el mapa del sitio web que se diseña utilizando diagramas de clases en UML junto con los estereotipos que el UWE proporciona en esta fase para cambiar la semántica o significado del elemento gráfico de una clase y que ésta represente una página WEB que forma parte del mapa de sitio que se modela, para ello se presenta el siguiente ejemplo en la Ilustración. 3.9.

Muestra el modelo de contenido del ejemplo de la libreta de direcciones simple, con las clases definidas para Libreta de direcciones, Contacto, Dirección y Teléfono. Los atributos representan los datos asociados a los objetos instanciados por esa clase.

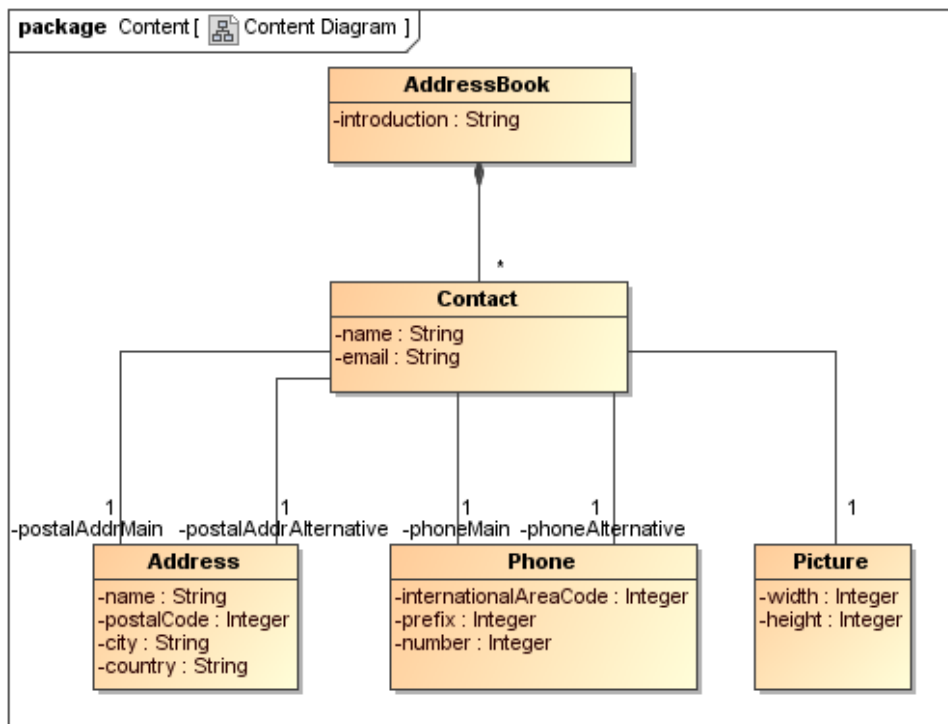


Ilustración 3.9 Ejemplo diagrama de clases [9].  
<https://uwe.pst.ifi.lmu.de/teachingTutorialContentSpanish.html>

## 1. Diagrama de navegación

Este diagrama indica como el sistema de páginas web se encuentra relacionado internamente. Es decir, muestra cómo se enlazan los elementos de navegación también se le puede llamar como *mapa de sitio web*, en base al diagrama de clases donde dichas clases representan una página web. Como ejemplos tenemos la siguiente Ilustración. 3.10.

El ejemplo muestra como interactúan la página “Libreta de direcciones” con la página “Contacto” representado con el estereotipo <<navigationClass>> donde para llegar de una a la otra, pasan por un índice, en este caso “Lista de contacto”. Los enlaces de navegación <<navigationLink>> y <<processLink>> muestran vínculos directos entre las clases de navegación y representan posibles pasos a seguir por el usuario, por lo tanto, estos vínculos tienen que ser dirigidos.

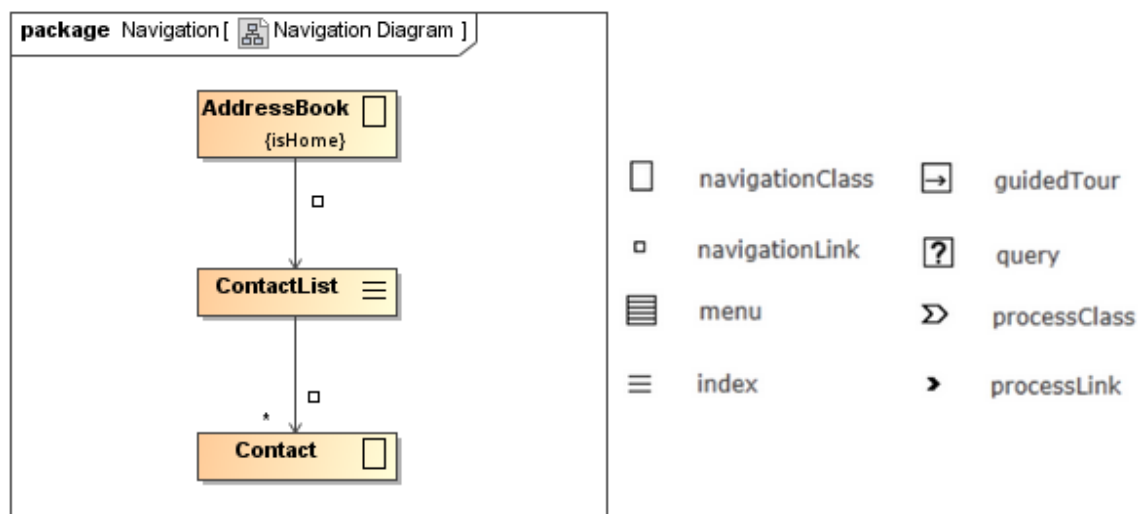


Ilustración 3.10 Ejemplo diagrama de navegación [9].  
(<https://uwe.pst.ifi.lmu.de/teachingTutorialNavigation.html>)

Significado de los estereotipos en base al diagrama anterior de la Ilustración 3.10 [8]:

- <<navigationClass>> representa los nodos navegables de la textura de hipertexto. Por ejemplo, “Libreta de direcciones” debe conectarse con “Contacto”.
- <<navigationLink>> muestra los vínculos directos entre las clases de navegación.
  - <<menu>> son las rutas alternativas de navegación manejadas por un menú.
  - <<index>> y <<guidedTour>> indica los accesos que se utilizan para llegar a múltiples instancias de una clase de navegación. Por ejemplo, “Lista de contacto” redirige a la página “Contacto” según sea el indicado.
  - <<query>> selecciona elementos.
  - <<processClass>> forman los puntos de entrada y salida de los procesos de negocio en este modelado.
  - <<processLink>> modela por enlaces de procesos a las clases de navegación.

## 2. Diagrama de presentación

El diagrama de presentación ofrece un modelo de la interfaz de usuario de una página web. Se basa en el modelo de navegación y en los aspectos concretos de la interfaz de usuario, donde destacan elementos como: formularios, imágenes, texto, enlaces, etc., todos ellos representados por estereotipos de UWE que proporcionan la semántica correcta al diseño de cada una de las páginas web que integran una aplicación web.

En el siguiente esquema de la Ilustración. 3.11 se muestra la representación del diagrama de presentación para una “Libreta de direcciones”. Está conformada por un encabezado llamado “introducción” marcado con el estereotipo texto, indicando que es el título de esta página. Otro elemento que destaca es “Contacto” siendo indicado como un grupo de presentación y se compone de otros elementos que pueden ser texto, botones, imágenes, entre muchos más.

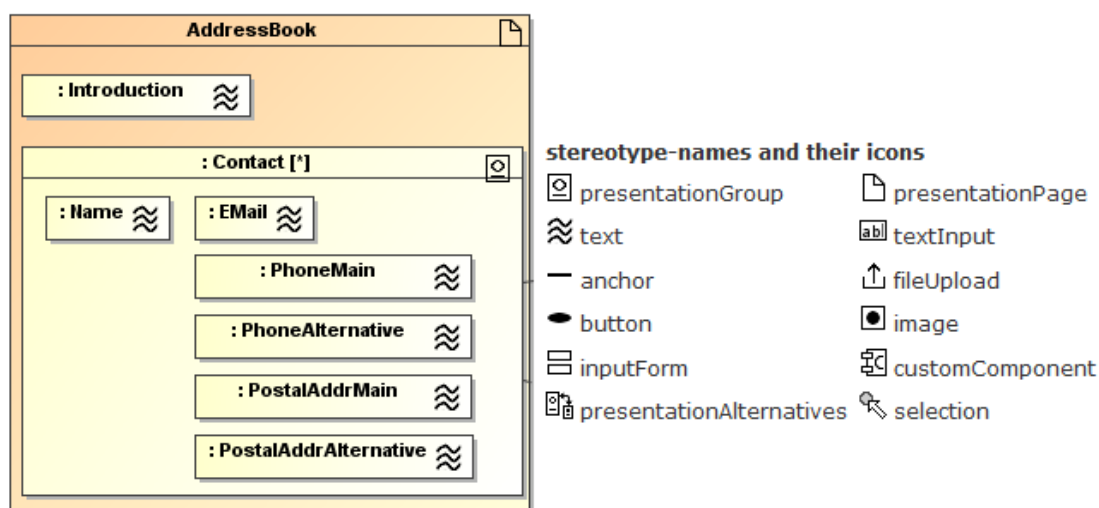


Ilustración 3.11 Ejemplo diagrama de presentación [9].  
(<https://uwe.pst.ifi.lmu.de/teachingTutorialPresentation.html>)

De esta forma concluye esta fase de diseño, hasta este punto se cubren las partes más esenciales de esta metodología donde ya teniendo todos estos modelos podemos dar comienzo a la implementación.

Significado de los estereotipos en base al diagrama anterior de la Ilustración 3.11 [8]:

- <<presentationGroup>> indica un conjunto de componentes en uno solo.
- <<text>> indica un párrafo u oración.
- <<anchor>> indica los enlaces.
- <<button>> indica la ejecución de un proceso.
- <<inputForm>> indica la recolección de información proporcionada por el usuario.
- <<presentationPage>> puede contener grupos de presentación, alternativas de presentación y grupos de presentación iterativos.

- <<textInput>> indica los datos de entrada.
- <<fileUpload>> indica la subida de archivos.
- <<image>> indica la colocación de una imagen.
- <<selection>> manipulación de objetos o componentes por parte del usuario.

### 3.4.3 Implementación

Para las fases anteriormente estipuladas se hace uso de MagicDraw la herramienta CASE (Ingeniería de Software Asistida por Computadora) para el modelado de diagramas, bajo los estereotipos definidos por UWE y hacer más fácil la interpretación de la funcionalidad del simulador de exámenes.

Este es el proceso de ejecución y programación de la aplicación web, donde tenemos como base el modelado de presentación y a través de este prototipo o boceto comenzamos a darle un diseño y funcionalidad. En esta implementación tenemos como base la estructura que maneja CENEVAL para la elaboración de reactivos en los exámenes EGEL que aplica.

En esta fase hacemos uso de diversas herramientas software como lo son editores de código, además de técnicas y algoritmos de búsqueda, consulta, a través de lenguajes de programación tales como: PHP, JavaScript e incluso Python con la utilización de algún Framework.

El modelo cliente-servidor podemos tomarlo de referencia para interpretar la forma en el que el usuario realiza las peticiones con el sistema, podemos verlo reflejado en nuestro simulador cuando un usuario solicita ver la lista de temas, la proporción de preguntas a contestar, validación de usuarios, entre otros.

### 3.4.4 Ejecución y pruebas

En esta fase final se aborda la corrección de errores que se vayan manifestando durante la implementación y en el producto final, pueden suscitarse errores sintácticos y lógicos. Algunas de las pruebas a realizar son:

- Realizar consultas a base de datos de manera correcta.
- Conexión a servidor web.
- Adaptación a dispositivos.
- Prevenir colisiones.
- Registro y validación de usuarios.
- Transición correcta del examen.

## Capítulo 4: Análisis de Requerimientos

En esta etapa es importante identificar las necesidades de los usuarios finales para llegar a una definición de requisitos del sistema que es fundamental en el proceso de desarrollo de software. Identificar usuarios finales es una forma de conocer los roles que asume una persona al momento de interactuar con la aplicación, de esta manera, se conocen los casos de uso que son operaciones o tareas que pueden realizar dentro de la aplicación.

### 4.1 Identificación de usuarios finales y casos de uso

Identificación de usuarios finales:

<i>Actores</i>	<i>Descripción</i>
Internauta (usuario final)	Es quien no dispone de un registro previo en el sistema y entra en un modo de visita en el sitio web. Puede visualizar que temas son para evaluar, pero sin tener el derecho de realizar uno y también dispone de poder realizar su registro a través de su matrícula y sus datos personales.
Alumno (usuario final)	Es quien ya paso por un registro y un logeo anteriormente. Puede elegir sobre qué tema realizar su examen y consultar su historial de resultados.
Profesor (administrador y usuario final)	Es quien ya cuenta con un registro y un logeo. Puede visualizar el avance de sus alumnos y llevar a cabo acciones como: agregar, modificar y eliminar preguntas o temas.

*Tabla 1 Descripción usuarios finales.*

Identificación de casos de uso para usuario final: Internauta

<i>Caso de Uso</i>	<i>Descripción</i>
Registro	Es la sección de registro para nuevos usuarios en el sistema donde se solicitan datos como: nombre, matrícula, contraseña, email, etc.... Al final de este proceso, los datos son guardados en el sistema de base de datos.
En registro: Verificar matrícula	Se hace una consulta en la base de datos sobre la matrícula que se desea registrar, evitando así una duplicidad de datos.

En inicio de sesión: Verificar matrícula y contraseña	Se realiza una consulta para determinar si la matrícula y contraseña coincide con las que fueron registradas anteriormente.
Inicio de sesión	Proceso por el cual debe pasar un internauta si quiere disponer del servicio que se ofrece, pidiendo datos como: matrícula y contraseña. Cabe destacar que debe pasar por un proceso de registro previo. Una vez concluido esto, el usuario pasa a ser un alumno o un profesor.
Consultar temas	Es un listado de temas que son dados de alta por el profesor que consta entre 10 y 15 preguntas. Dichas preguntas no se pueden mostrar en este listado hasta que se inicie sesión y se realice un examen.

*Tabla 2 Descripción casos de uso: Internauta.*

Identificación de casos de uso para usuario final: Alumno

<i>Caso de Uso</i>	<i>Descripción</i>
En inicio de sesión: Verificar matrícula y contraseña	Se realiza una consulta para determinar si la matrícula y contraseña coincide con las que fueron registradas anteriormente.
Inicio de sesión	Proceso por el cual debe pasar un alumno si quiere disponer del servicio que se ofrece, pidiendo datos como: matrícula y contraseña. Cabe destacar que debe pasar por un proceso de registro previo. Una vez concluido esto, el alumno podrá realizar exámenes y consultar sus resultados.
Elegir tema	Es un proceso de selección donde un alumno dispone de un temario para así poder elegir entre esta lista solo un tema y pasar al proceso de “realizar examen”.
Realizar examen	Es una serie de preguntas lanzadas de forma aleatoria conformadas por 4 posibles respuestas donde solo una es posible seleccionar.
Ver calificación	Proceso final de la realización de un examen, muestra la calificación final con su número de respuestas correctas e incorrectas.

Historial académico Es una sección donde se pueden visualizar el número de exámenes realizados con su respectiva calificación, un número de intentos, entre otros detalles más.

Tabla 3 Descripción casos de uso: Alumno

Identificación de casos de uso para administrador: Profesor

<i>Caso de Uso</i>	<i>Descripción</i>
En inicio de sesión: Verificar matrícula y contraseña	Se realiza una consulta para determinar si la matrícula y contraseña coincide con las que fueron registradas anteriormente.
Inicio de sesión	Proceso por el cual debe pasar un profesor si quiere disponer del servicio que se ofrece, pidiendo datos como: matrícula y contraseña. Cabe destacar que debe pasar por un proceso de registro previo. Una vez concluido esto, el profesor será capaz de modificar el temario y visualizar el progreso de sus alumnos.
Progreso del alumno	Es una sección similar a “Historial académico” pero esta es vista por el profesor, en ella él puede ver calificaciones por cada alumno en cada examen realizado. Además de llevar el control de sus alumnos.
Alta de temas	Proceso que únicamente el profesor es capaz de realizar, agrega los temas nuevos a realizar, guardándolos en el sistema de base de datos.
Baja de temas	Proceso que elimina temas que ya no son requeridos para los alumnos, los borrará de forma definitiva de la base de datos.
Alta de preguntas	Proceso que es manipulado por el profesor donde se agregan preguntas nuevas, para ello se debe elegir primero a que tema corresponde, con sus respectivas respuestas y son guardadas en la base de datos.
Baja de preguntas	Proceso que elimina preguntas cuando estas ya no son requeridas, son borradas de forma definitiva de la base de datos.

Modificar preguntas

Proceso que altera las preguntas cuando las respuestas suelen cambiar, se puede modificar la pregunta o únicamente las respuestas.

Tabla 4 Descripción casos de uso: Profesor

## 4.2 Diagrama de casos de uso

El siguiente diagrama mostrado en Ilustración 4.1 es la representación gráfica de como los usuarios finales anteriormente establecidos tiene interacción con el sistema en relación con las funciones que pueden realizar dependiendo su rol. Cabe señalar que los estereotipos inmersos en el diagrama fueron presentados en el capítulo anterior, *Capítulo 3.4: Metodología*.

Las relaciones denominadas con <<include>> nos indican que hay una dependencia entre las actividades, un claro ejemplo puede ser; la operación registro no puede llevarse a cabo si antes no se verifica la matrícula proporcionada, es decir, son dos operaciones en una. Otro ejemplo es: no se puede iniciar un examen si antes no se selecciona un tema para evaluar.

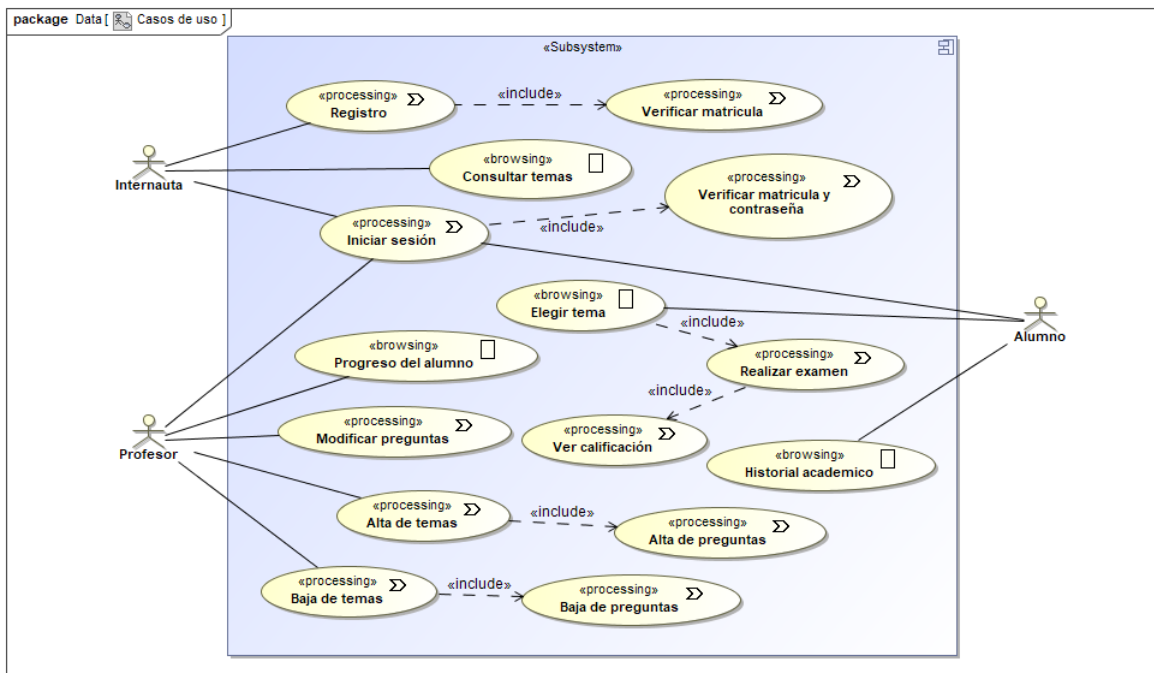


Ilustración 4.1 Diagrama de casos de uso

### 4.3 Diagrama de actividades

Los siguientes diagramas representan a detalle cada uno de los casos de uso visto en el diagrama anterior en Ilustración 4.1, desde la presentación de los componentes, la interacción del usuario hasta el comportamiento interno en el sistema. En este tipo de diagramas nos podemos encontrar con una toma de decisiones (representada con una forma de rombo) que nos pueden llevar a realizar diferentes procesos dependiendo del comportamiento del usuario con la aplicación.

#### Caso de uso Internauta: Registro / Verificar matrícula

El registro tiene una serie de actividades que nos van a permitir, identificar, buscar y validar los datos proporcionados por el usuario, con una base de datos de por medio para realizar peticiones. La siguiente Ilustración. 4.2 representa lo anterior dicho.

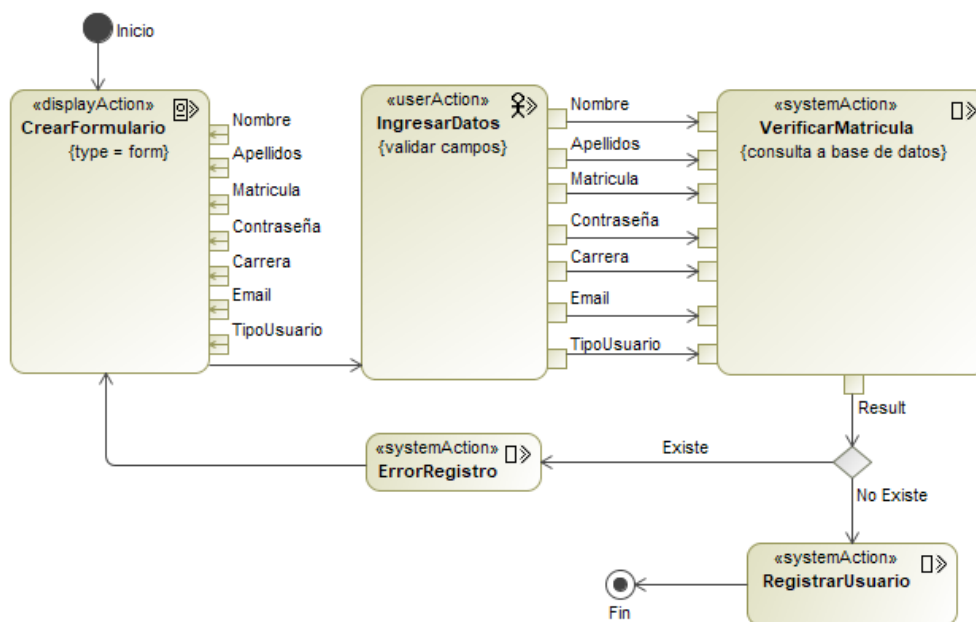


Ilustración 4.2 Diagrama actividad: Registro - Verificar matrícula

## Caso de uso Internauta: Consultar temas

La consulta de temas es una actividad que no requiere de muchos recursos, únicamente se solicita un listado de todos los temas que están disponibles y se le presentan al usuario de forma ordenada. La siguiente Ilustración. 4.3 muestra el proceso.

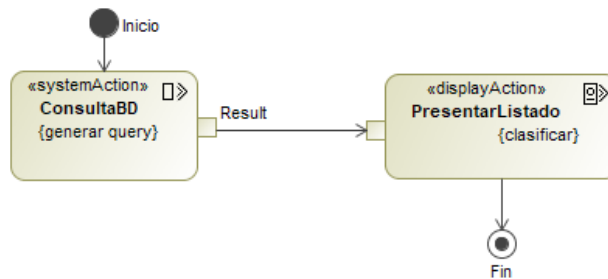


Ilustración 4.3 Diagrama actividad: Consultar temas

## Caso de uso Internauta, Alumno y Profesor: Iniciar sesión / Verificar matrícula y contraseña

El procedimiento de inicio de sesión es un proceso de autorización que consiste en mostrar los recursos adecuados dependiendo del tipo de usuario, únicamente se requiere de la matrícula y contraseña. Internamente se realiza una búsqueda que determina si el usuario está registrado o no en el sistema; de no estar registrado se genera una alerta indicando el motivo por el cual no inicio sesión y en caso contrario, se le da acceso; como se muestra en la Ilustración 4.4.

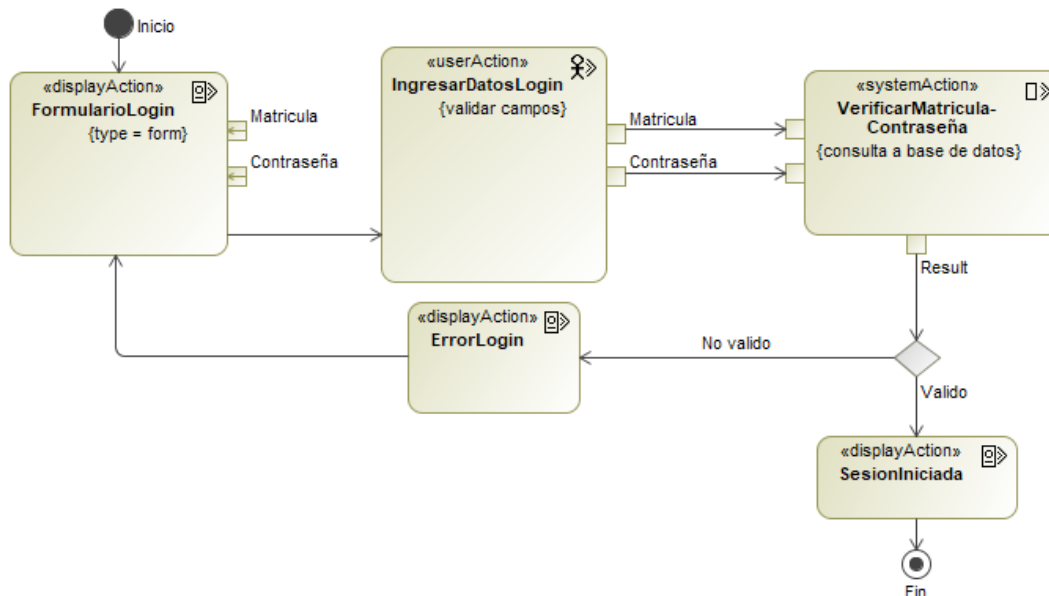


Ilustración 4.4 Diagrama actividad: Iniciar sesión – Verificar matrícula y contraseña

## Caso de uso Alumno: Elegir tema / Realizar examen / Ver calificación

Realizar un examen es un procedimiento que involucra utilizar diversos recursos como: peticiones, vistas, manipulación de datos, desestructuración, entre otros. La idea principal de este diagrama es mostrarle al alumno una serie de preguntas que están asociadas al tema seleccionado. Mientras el examen este en curso el alumno no podrá retornar al menú hasta concluirlo, si desea cancelarlo sin completar esto genera que las preguntas sin contestar se manden vacías, por lo cual, de este modo la calificación se ve afectada. La forma en que se realiza la actividad se representa en la siguiente Ilustración 4.5:

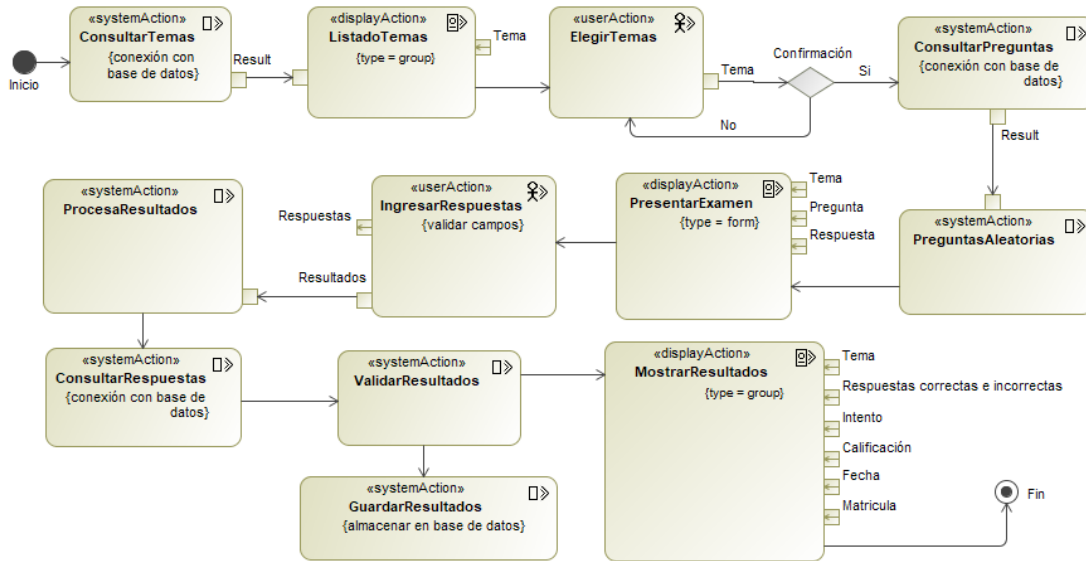


Ilustración 4.5 Diagrama actividad: Elegir tema – Realizar examen – Ver calificación

## Caso de uso Alumno: Historial académico

Este es un procedimiento de consulta, la petición hace referencia al tema que el alumno desea visualizar y como resultado es un promedio de todos los intentos realizados para ese examen en específico, como se puede ver en la Ilustración 4.6.

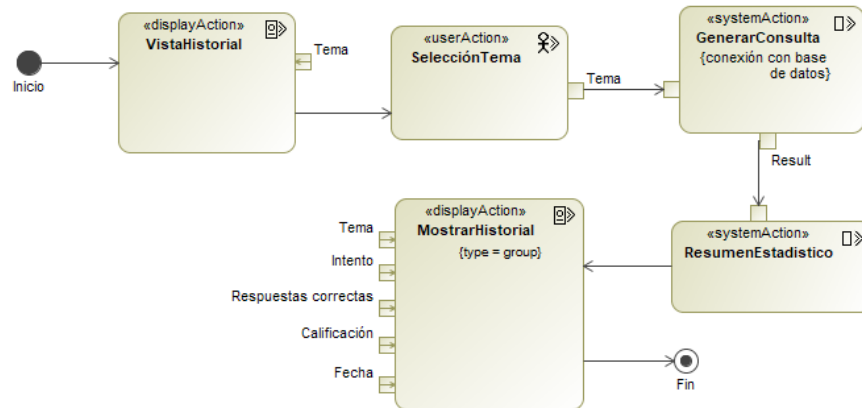


Ilustración 4.6 Diagrama actividad: Historial académico

## Caso de uso Profesor: Progreso del alumno

El siguiente diagrama de la Ilustración 4.7 tiene un enfoque muy parecido al “Historial Académico” del alumno, la diferencia es que esta labor de “Progreso del Alumno” es una tarea del profesor que tiene asociada una lista de todos los alumnos registrados donde puede ver el rendimiento de cada uno.

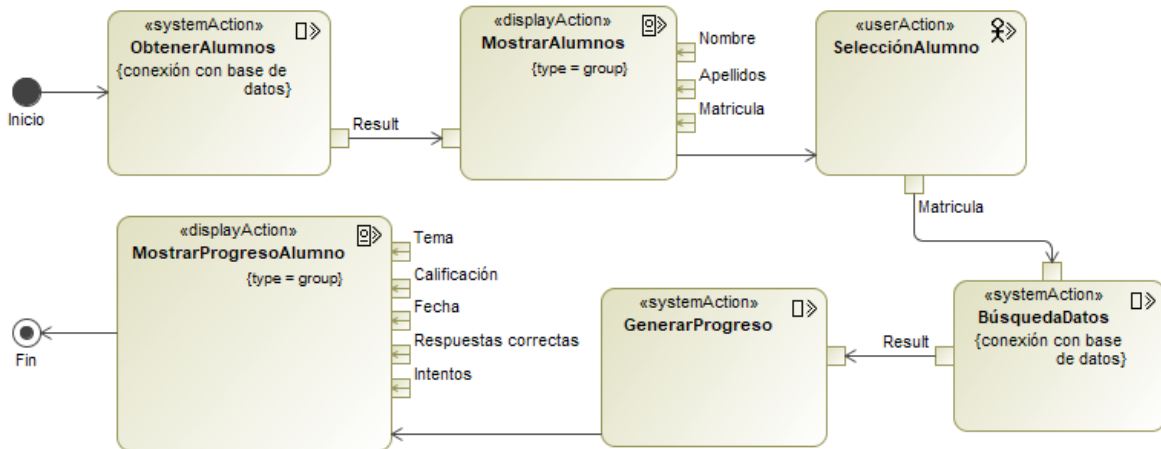


Ilustración 4.7 Diagrama actividad: Progreso del alumno

## Caso de uso Profesor: Modificar preguntas

La Ilustración 4.8 representa un apartado donde el profesor puede consultar para reformular o editar sus preguntas, para ello es necesario primero indicar el tema del cual se trata y es un procedimiento iterativo, concluye cuando el profesor para de modificar.

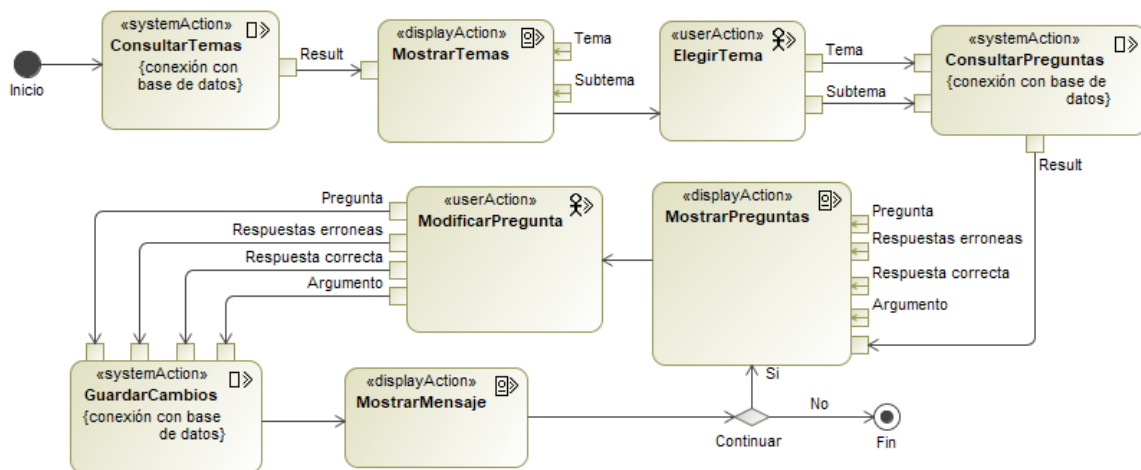


Ilustración 4.8 Diagrama actividad: Modificar preguntas

## Caso de uso Profesor: Alta de temas / Alta de preguntas

Alta de temas tiene dos opciones principales. Nuevo tema, ocurre cuando el tema a ingresar es nuevo y no forma parte del listado de temas actual. Tema existente, ocurre cuando ya está dado de alta un tema en concreto, una vez aclarando este filtro lo siguiente es la “Alta de preguntas” que influye de la misma forma en ambas opciones como se puede observar en la siguiente Ilustración 4.9. Podemos notar que al final de la actividad se lleva de manera iterativa el registro de las preguntas y concluye cuando el profesor lo indica.

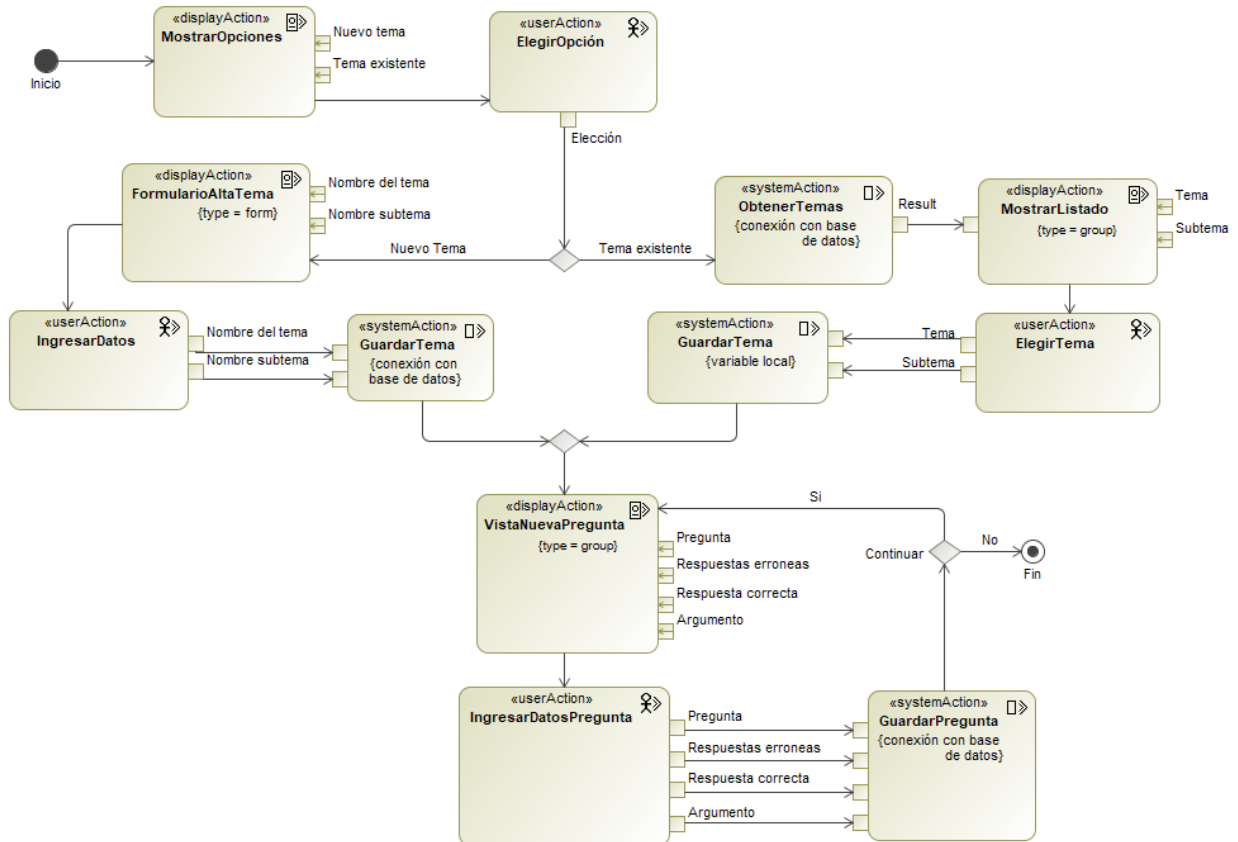


Ilustración 4.9 Diagrama actividad: Alta de temas - Alta de preguntas

## Caso de uso Profesor: Baja de temas / Baja de preguntas

Baja de temas es una operación rápida de llevar a cabo pero que puede ser riesgosa, es decir, indicar la baja de un tema involucra borrar todo lo relacionado con este tema, esto incluye todas las preguntas y registros donde se haga referencia al tema. Baja de preguntas borra de manera única una pregunta indicada y asociada a un tema con sus respectivos reactivos y argumento, no se puede borrar en grupos de preguntas como se puede ver en la Ilustración 4.10.

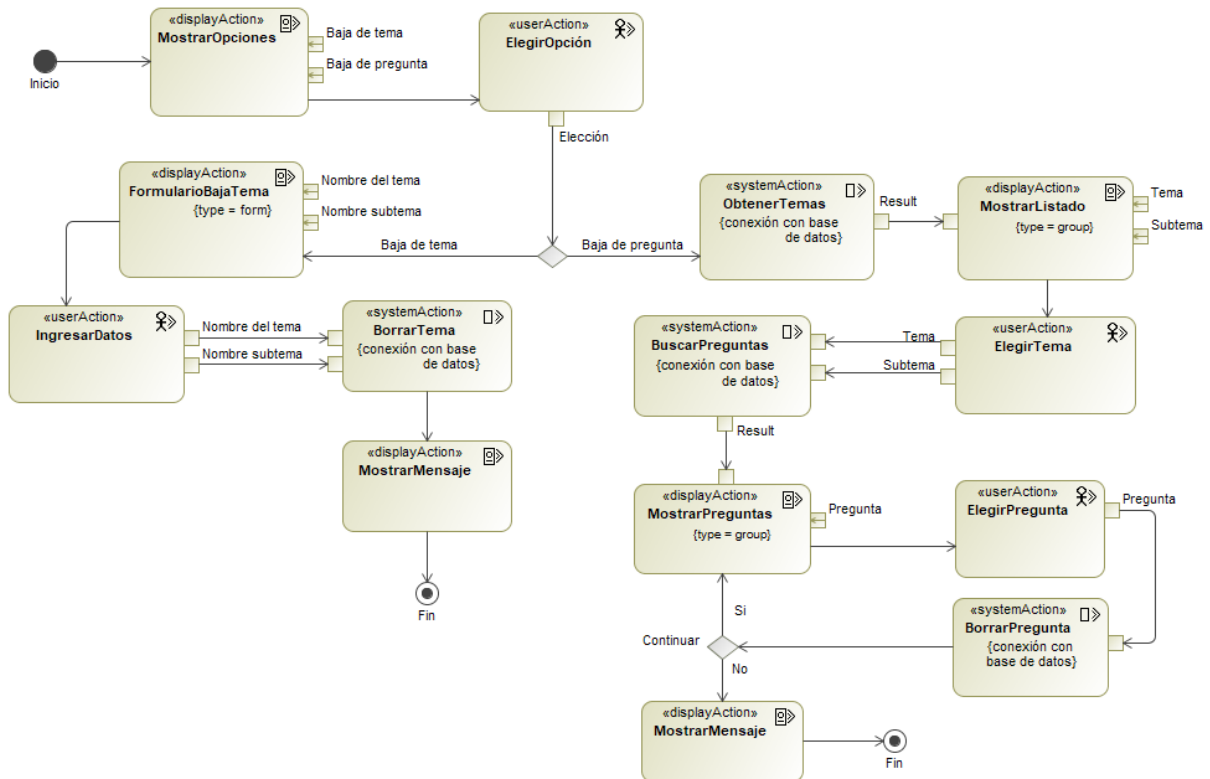


Ilustración 4.10 Diagrama de actividad: Baja de temas - Baja de preguntas

## Capítulo 5: Diseño

El diseño de software es un proceso que nos permite crear varios modelos del sistema que se desea construir, con el fin de tener un producto final sofisticado y de alta calidad. En este capítulo se abordan conceptos y diagramas relacionadas a la fase de diseño, partiendo del diagrama de contenido o también llamado diagrama de clases, para posteriormente realizar diagramas de navegación y el respectivo diagrama de presentación, este último es la base para la estructura de la aplicación web.

### 5.1 Diagrama de contenido

Recordemos que un diagrama de clases es una estructura estática que describe la estructura de un sistema mostrando las clases, sus atributos, operaciones, y las relaciones entre los objetos. En la siguiente Ilustración 5.1 se muestra la interacción entre dichos objetos.

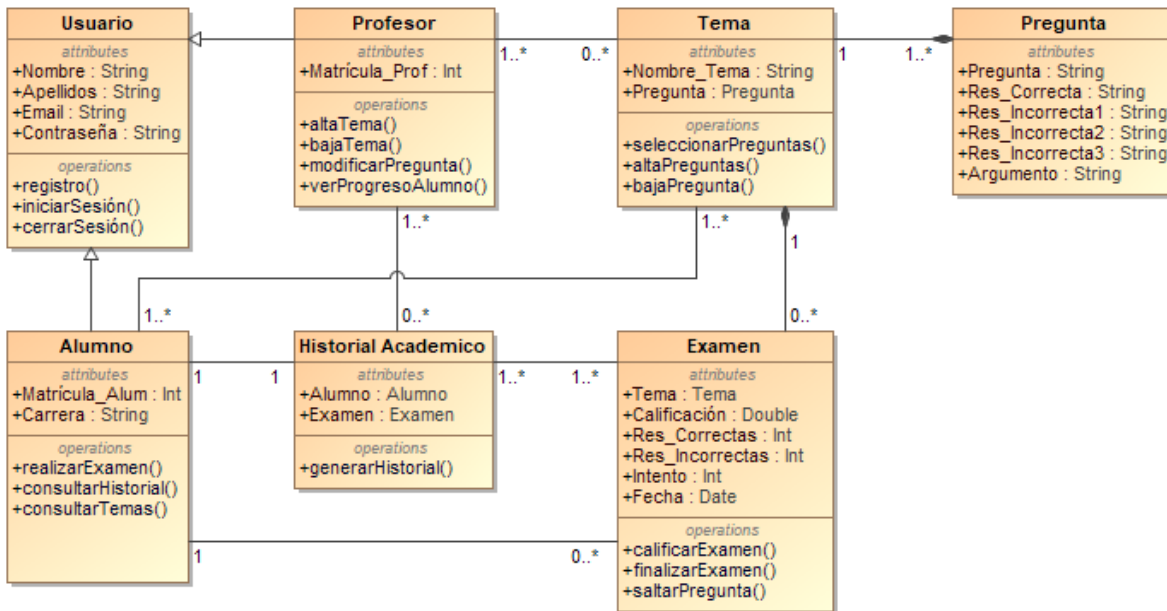


Ilustración 5.1 Diagrama de contenido

Una interpretación breve del diagrama anterior de la Ilustración 5.1 es el siguiente:

- El alumno y profesor son usuarios finales en el sistema que heredan atributos de la clase Usuario.
- El profesor puede agregar, eliminar y modificar temas y sus respectivas preguntas. Asimismo, puede consultar el historial académico de sus alumnos.

- El alumno puede consultar los temas disponibles, realizar exámenes sobre un tema en específico y a su vez puede consultar su historial académico.
- Cada examen corresponde a un único tema.
- Cada tema tiene su propio banco de preguntas.

## 5.2 Diagrama de navegación

A continuación, los diagramas que se presentan corresponden a la forma en que se navega internamente en la aplicación web, esto conlleva a redireccionamientos, cambios de dominio en la URL y el contenido presentado en cada interfaz.

Como se sabe toda página, aplicación o sitio web tiene un inicio que comúnmente lo llamamos Home o Index. La Ilustración 5.2 representa las vistas y redireccionamientos entre las páginas desde una perspectiva de internauta, en el que podemos observar está compuesta por un menú que contiene páginas como: Registro, Inicio de Sesión y la Consulta de Temas. Visitar una página puede llegar a requerir de una búsqueda y cabe señalar que los enlaces entre estas páginas UWE la denomina como <<navigationLink>> y <<processLink>> en el *Capítulo 3.4.2 Diseño web* se hace referencia a estos estereotipos.

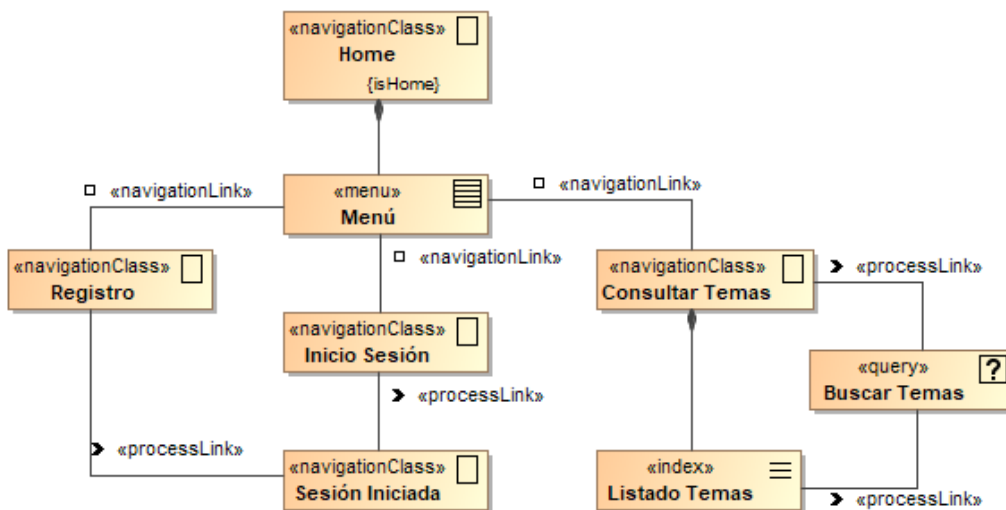


Ilustración 5.2 Diagrama de navegación – Internauta

Para la navegación desde la perspectiva del alumno se identifica cuáles serán las opciones para presentar tales como: Consultar Temas, Historial Académico y Realizar Examen. Es importante mencionar que cada página requiere de una búsqueda porque es necesario indicar que datos quiere visualizar el alumno. En el caso de Cerrar Sesión no es como tal una página que se puede acceder, sino que por medio del estereotipo <<processClass>> indicamos que es un punto de salida dentro del modelado. Todo lo mencionado se puede visualizar en la Ilustración 5.3.

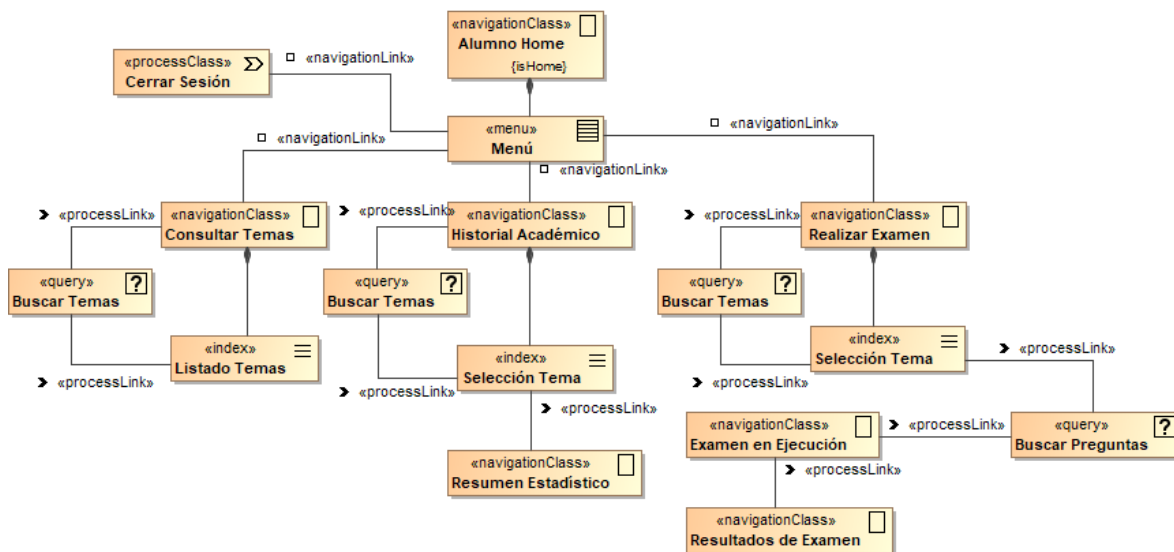


Ilustración 5.3 Diagrama de navegación - Alumno

Ilustración 5.4 representa la navegación del profesor que es muy similar al alumno.

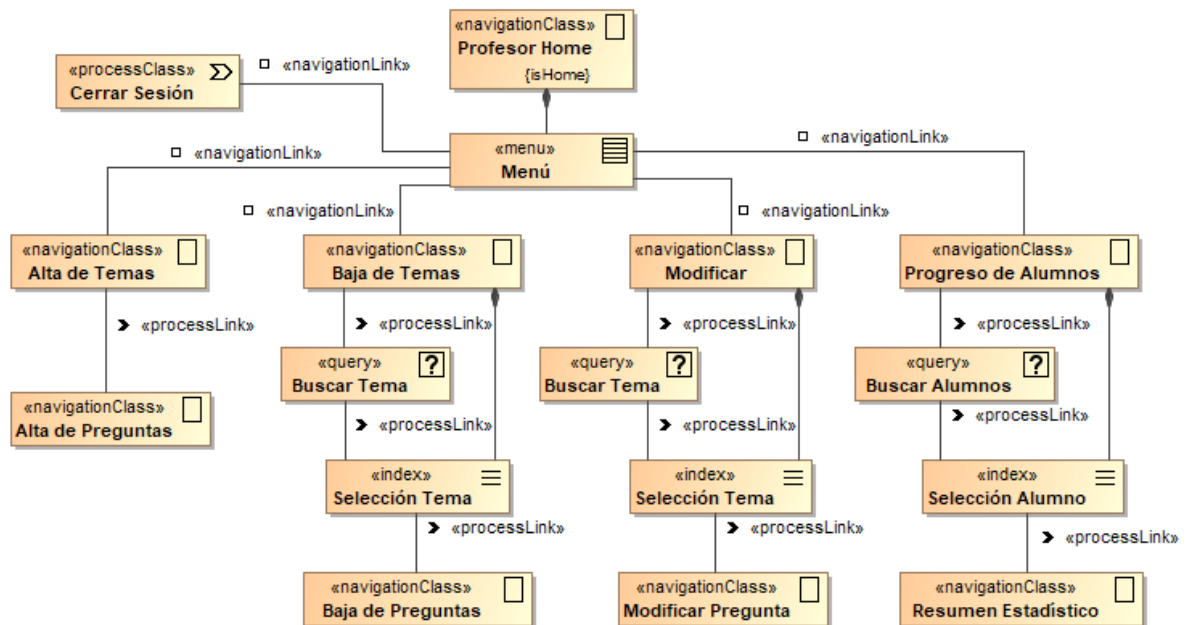


Ilustración 5.4 Diagrama de navegación – Profesor

### 5.3 Diagrama de presentación

El diseño de la interfaz involucra aspectos importantes como: color, tipografía, experiencia de usuario y contenido multimedia, todo lo necesario para captar la atención del usuario. Se propone tener una vista amigable, cómoda y dinámica para que el usuario se sienta seguro y disfrute de los beneficios de la aplicación, pero ¿Cómo podemos lograr esto? Los estilos en CSS nos ofrecen una forma de hacer más llamativa la aplicación por medio de diseños responsivos, animación, además que nos aportan un mejor control en la presentación de un sitio web.

El objetivo principal es proporcionarle al usuario la información que desea ver o manipular sin requerir tanta labor de su parte, es decir, tener a disposición los recursos. Por ello, ninguna información a mostrar debe sobrepasar de dos a tres clics ya que con esto se espera tener mayor fluidez al momento de navegar. La siguiente Ilustración 5.5 corresponde a la página de inicio visto desde un internauta como se observará es una bienvenida al sitio en el que se describirá en lo que consta la aplicación.

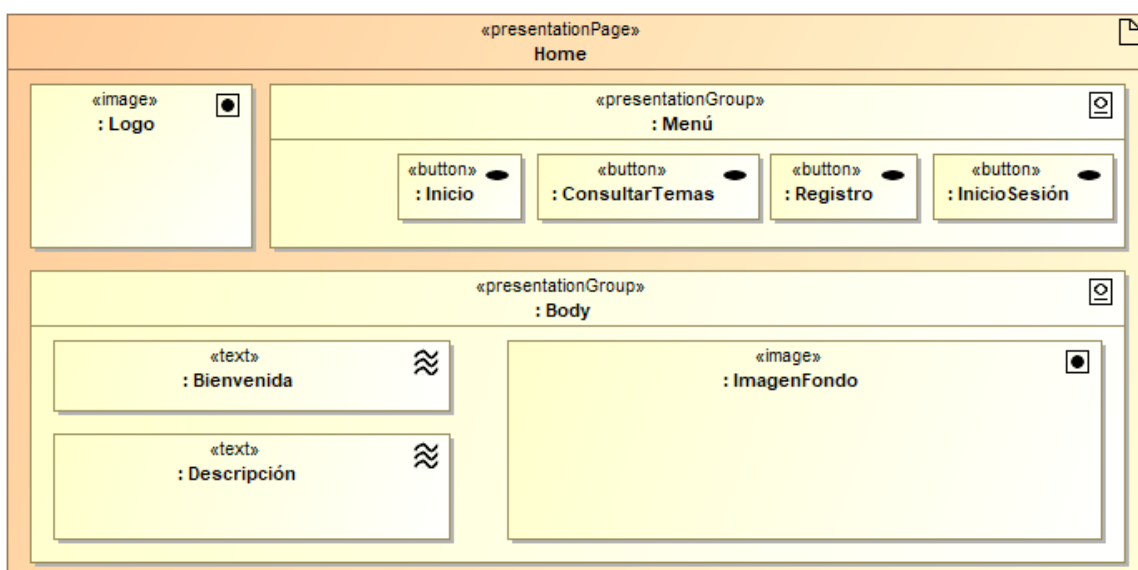


Ilustración 5.5 Diagrama de presentación - Internauta Home

Consultar Temas se compone de un listado dinámico, donde el usuario podrá seleccionar un tema de interés y observar los subtemas, de este modo evitamos presentar una larga lista. Dicho diseño podemos visualizarlo en la Ilustración 5.6.

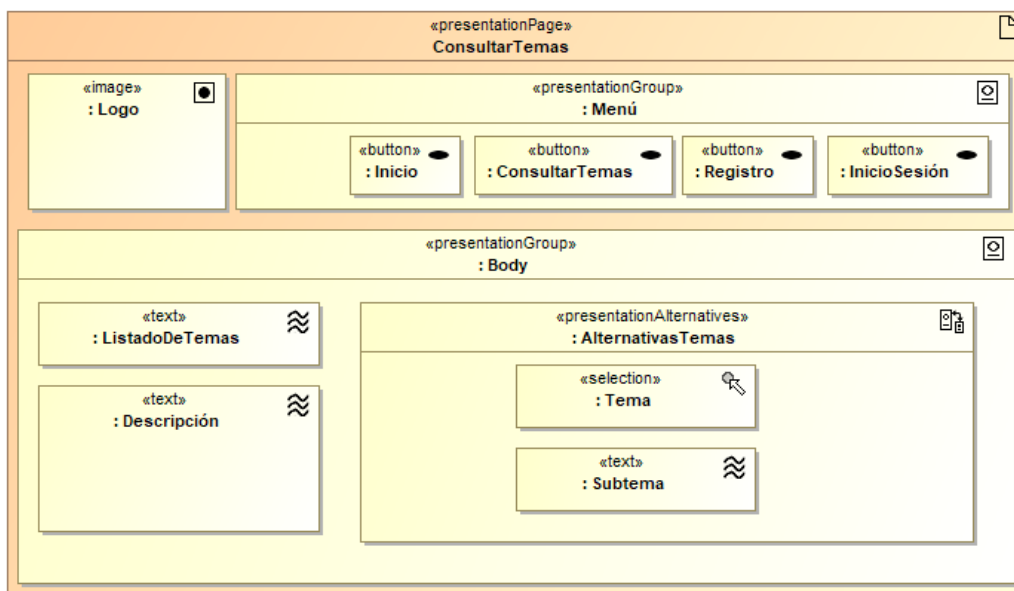


Ilustración 5.6 Diagrama de presentación - Internauta Consultar temas

Registro se conforma de un formulario que reúne los datos esenciales de los usuarios, en la parte izquierda el usuario podrá asignarse un rol como “Alumno” o “Profesor” y de acuerdo con este rol el formulario presentará los campos requeridos para su registro como se indica en la Ilustración 5.7.

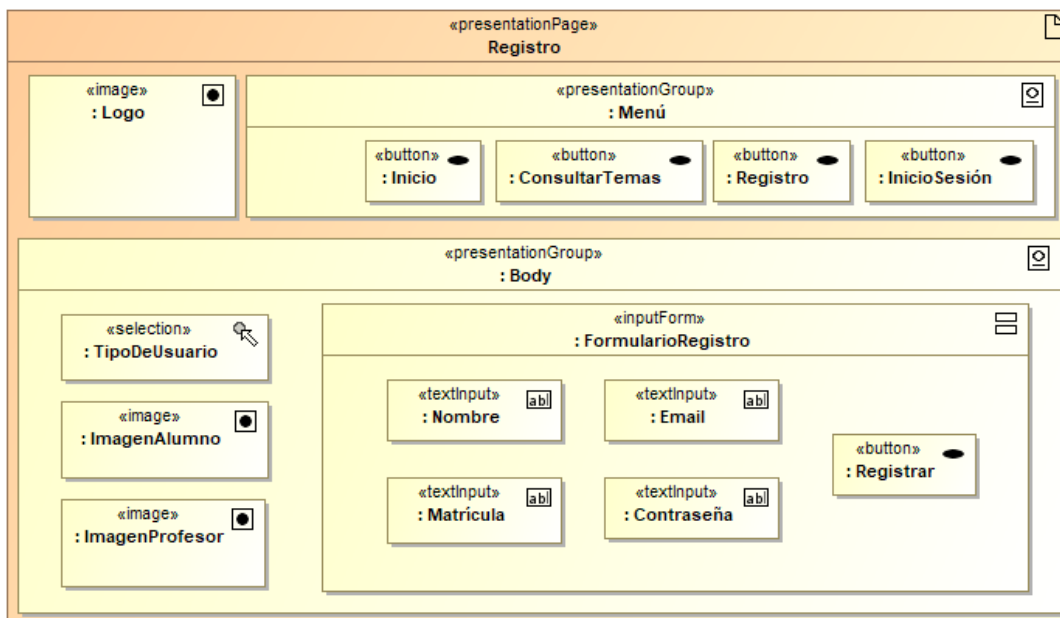


Ilustración 5.7 Diagrama de presentación - Internauta Registro

Inicio de sesión es el formulario básico que requiere de una matrícula y contraseña para tener acceso a la plataforma, la Ilustración 5.8 es prueba de ello.

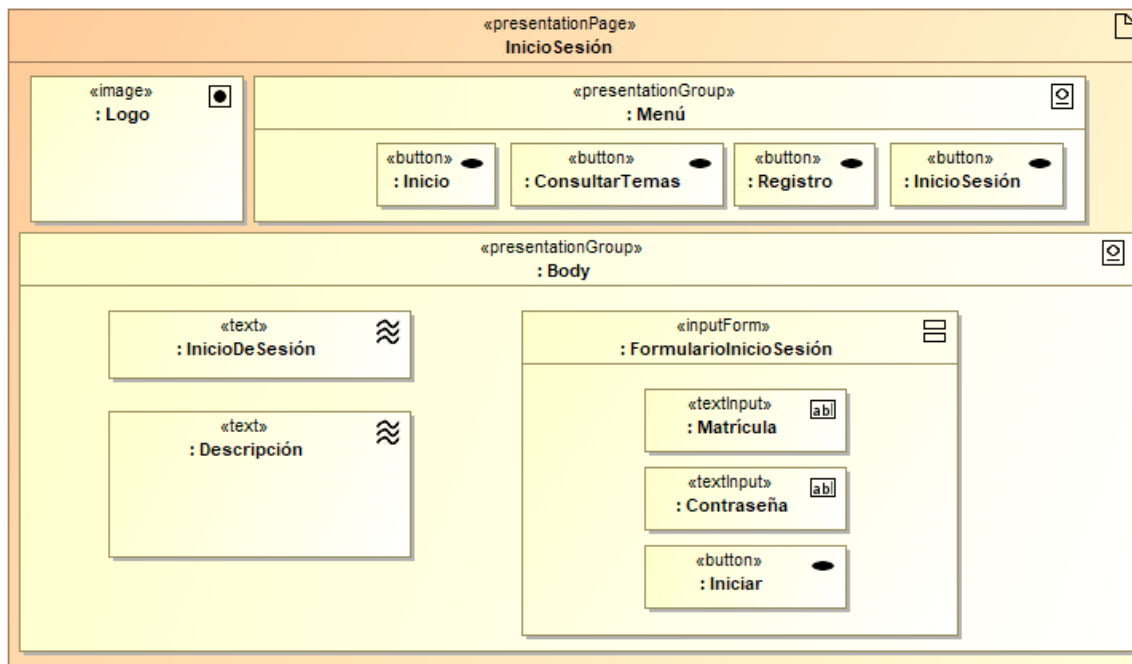


Ilustración 5.8 Diagrama de presentación - Internauta Inicio de sesión

Una vez se lleva a cabo un registro o inicio de sesión, la siguiente página mostrará el contenido del Home para un alumno como lo indica la Ilustración 5.9. En el menú se presentan atajos por medio de botones que redireccionan a otras páginas.

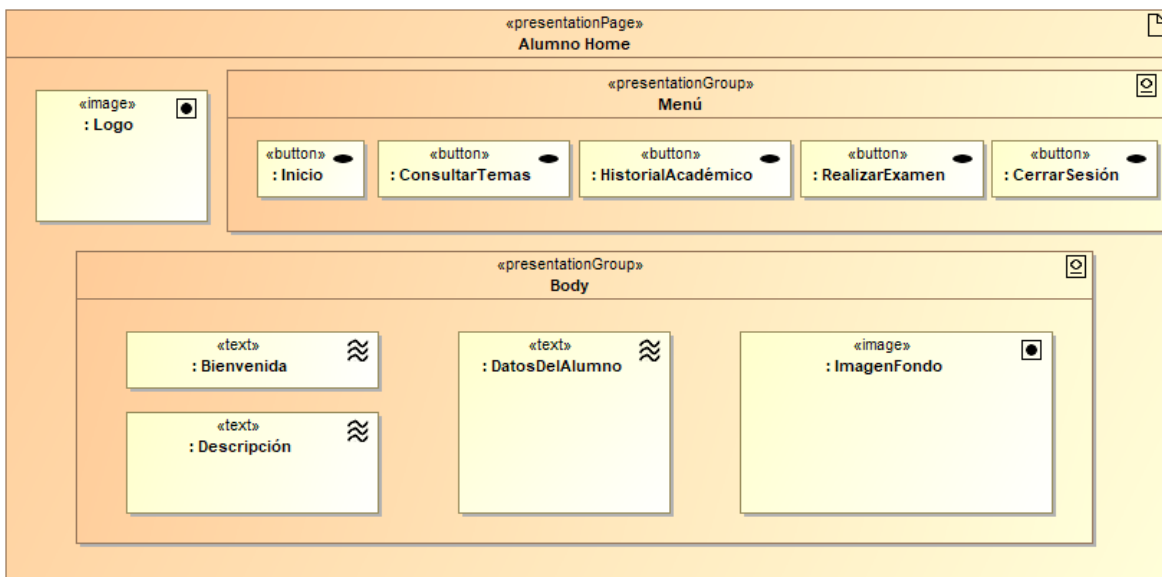


Ilustración 5.9 Diagrama de presentación - Alumno Home

Consultar temas es muy similar a la página que se le presenta al Internauta, a diferencia de esta se le proporciona al alumno un botón a “Realizar Examen” y se puede considerar como un atajo rápido que con lleva a la ejecución del examen, la Ilustración 5.10 hace alusión a lo anterior mencionado.

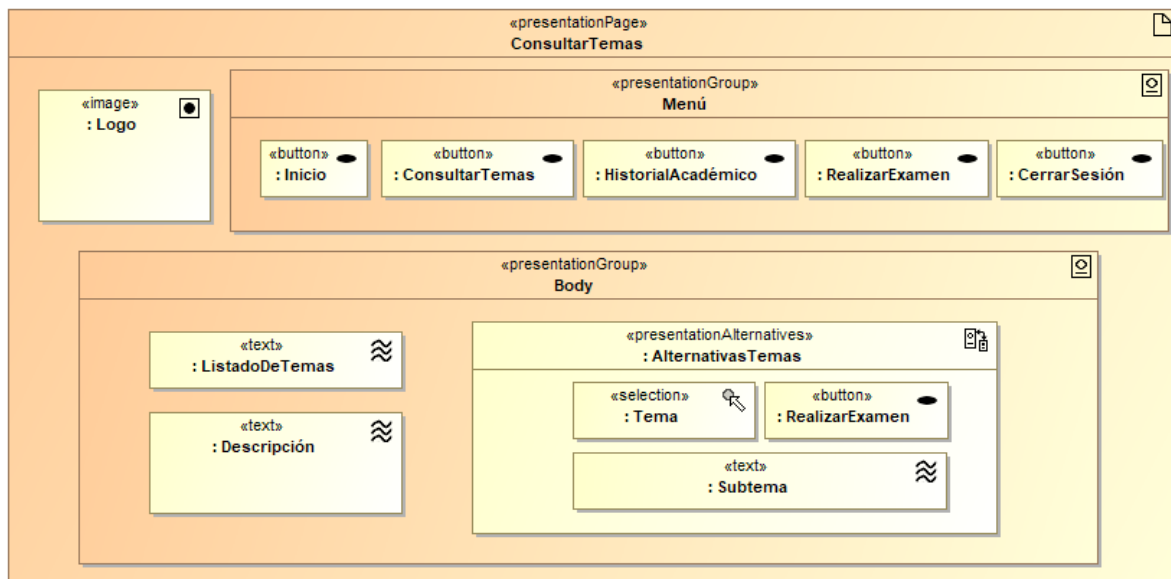


Ilustración 5.10 Diagrama de presentación - Alumno Consultar temas

El historial académico tiene su similitud a vistas anteriormente establecidas como se ve en la Ilustración 5.11, donde la diferencia radica en solo seleccionar el nombre de un tema para generar el resumen estadístico de cada intento asociado a este, como se puede apreciar en la Ilustración 5.12.

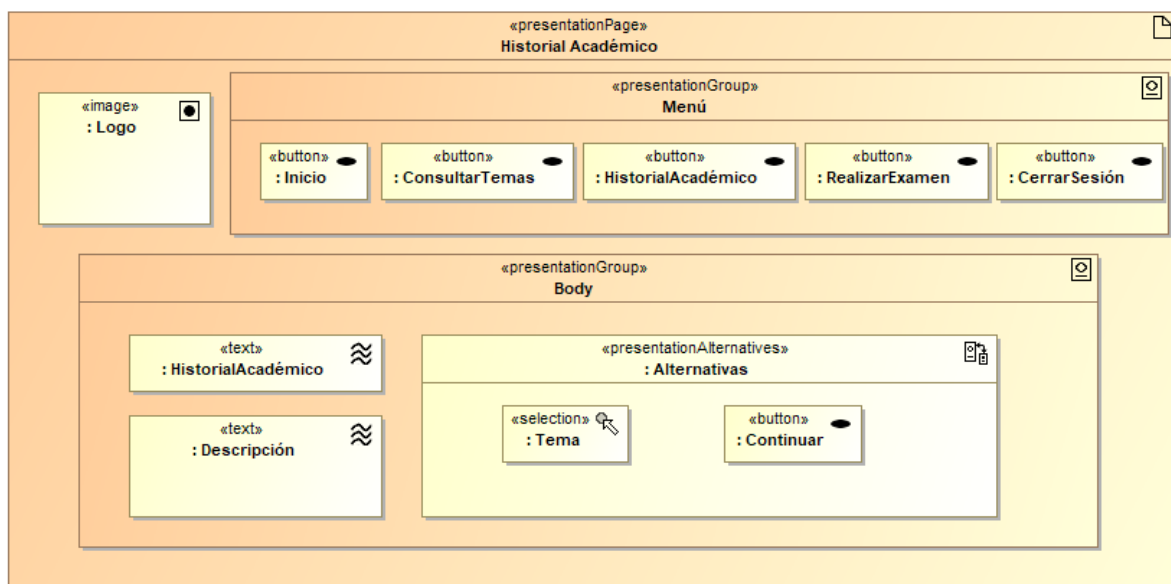


Ilustración 5.11 Diagrama de presentación - Alumno Historial académico

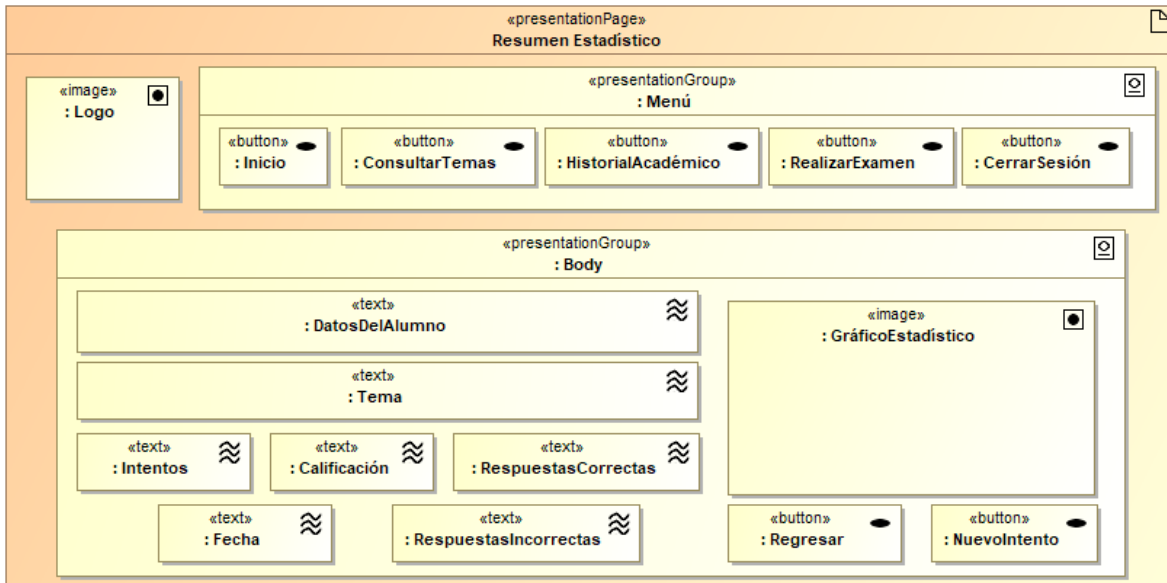


Ilustración 5.12 Diagrama de presentación - Alumno Resumen estadístico

Para realizar un examen como anteriormente se ha señalado, es necesario indicar cual es el tema por evaluar, por ello utilizo el estereotipo de <<presentationAlternatives>> porque las opciones cambian dependiendo del tema a seleccionar, siendo la Ilustración 5.13 la imagen que lo representa.

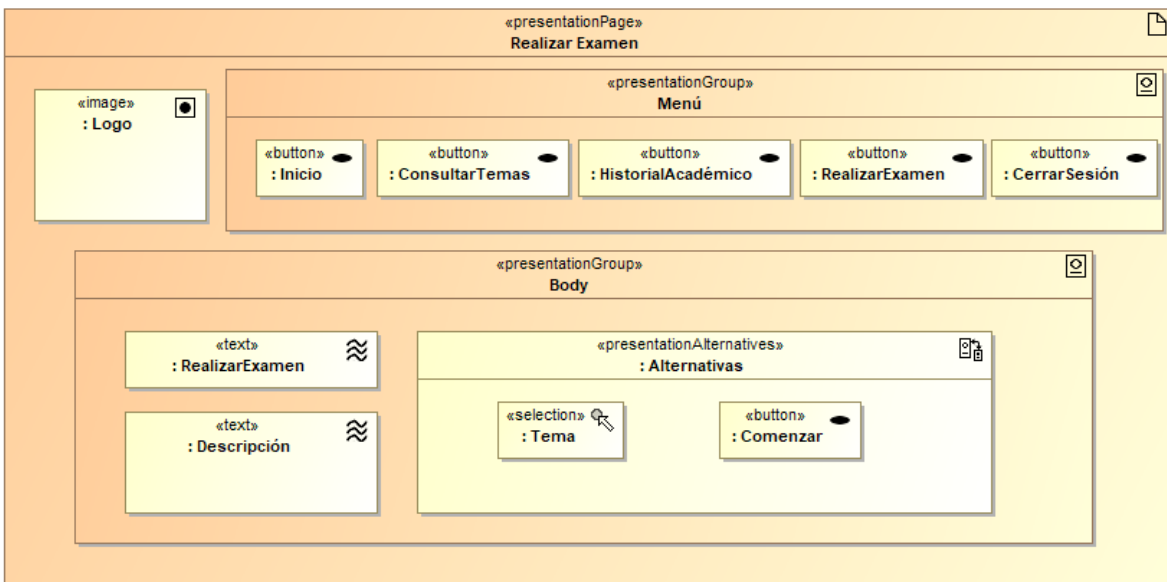


Ilustración 5.13 Diagrama de presentación - Alumno Realizar examen

Una vez confirmado el tema que desea evaluarse, se inicia el examen, la siguiente Ilustración 5.14 corresponde a la página.

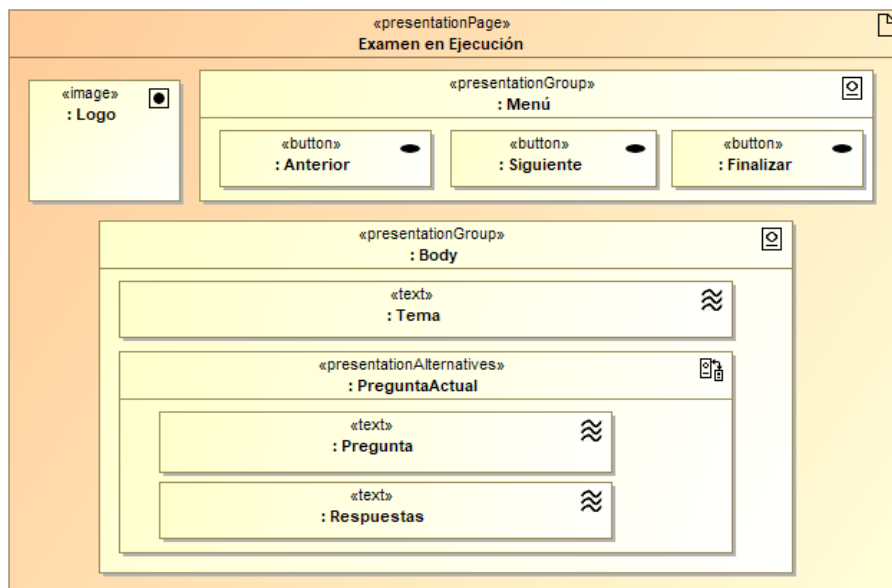


Ilustración 5.14 Diagrama de presentación - Alumno Examen en ejecución

Finalmente, cuando concluye el examen se procede a mostrar los resultados del examen, el cual contiene datos como la calificación, el número de intentos, entre otros. Además, incluye un apartado llamado “PreguntasRespondidas” que consiste en mostrarle al alumno cuales fueron sus errores o aciertos y darle una razón por medio de un argumento, como lo indica la Ilustración 5.15.

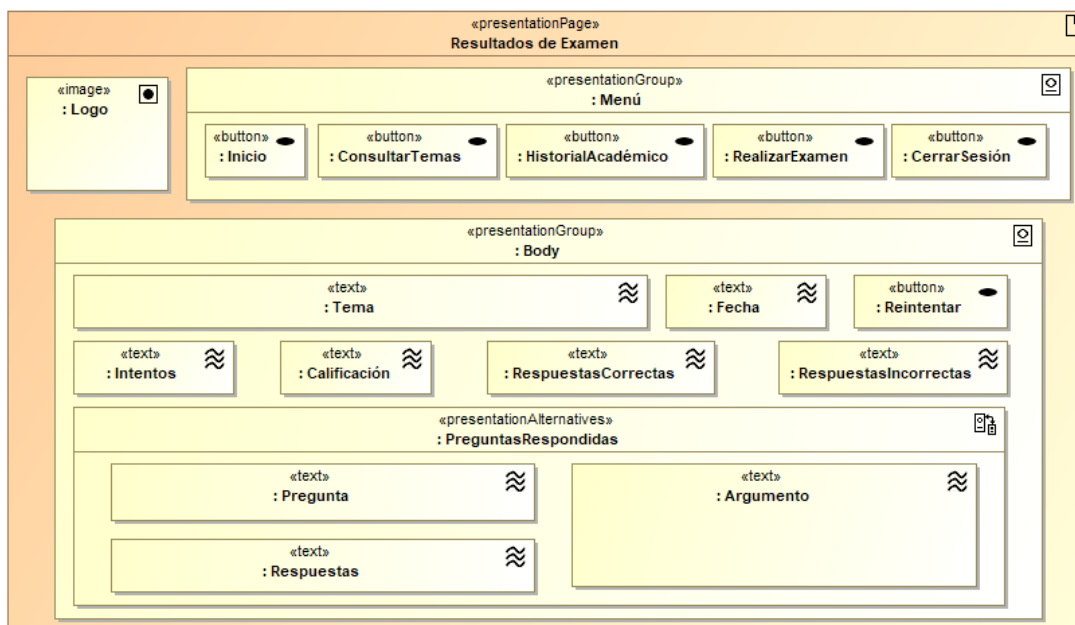


Ilustración 5.15 Diagrama de presentación - Alumno Resultados de examen

La página inicial del profesor (home) de la Ilustración 5.16 mantiene la misma estructura que el inicio de un alumno, siendo el menú el cambio entre estos dos.

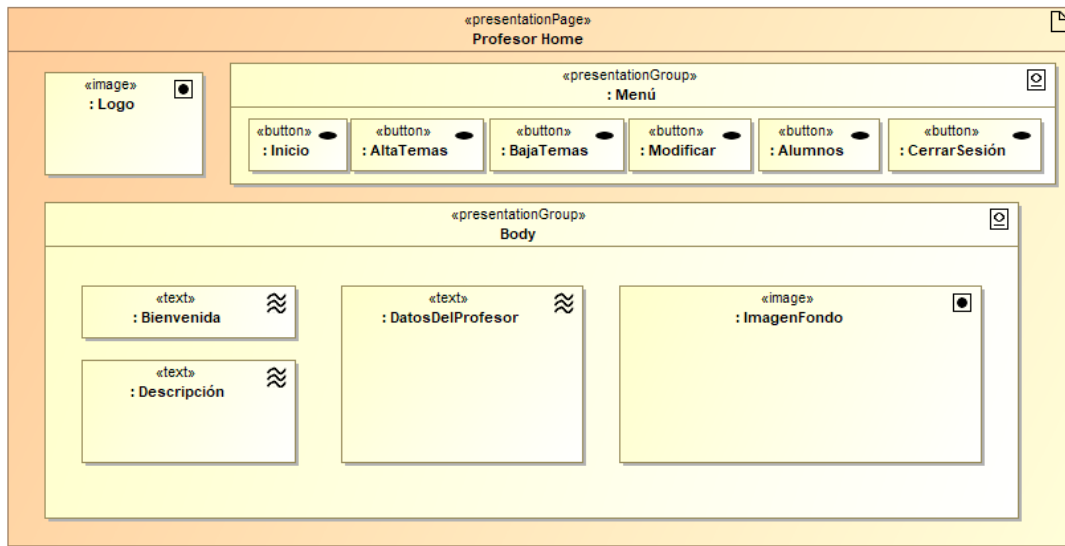


Ilustración 5.16 Diagrama de presentación – Profesor Home

La vista de alta de temas recordemos que está compuesta de dos diferentes elecciones, cuando se desea agregar un nuevo tema se presenta un apartado para indicar el nombre, en caso contrario se indica el tema y se procede a llenar el formulario. Ilustración 5.17 hace referencia a dicha vista o página.

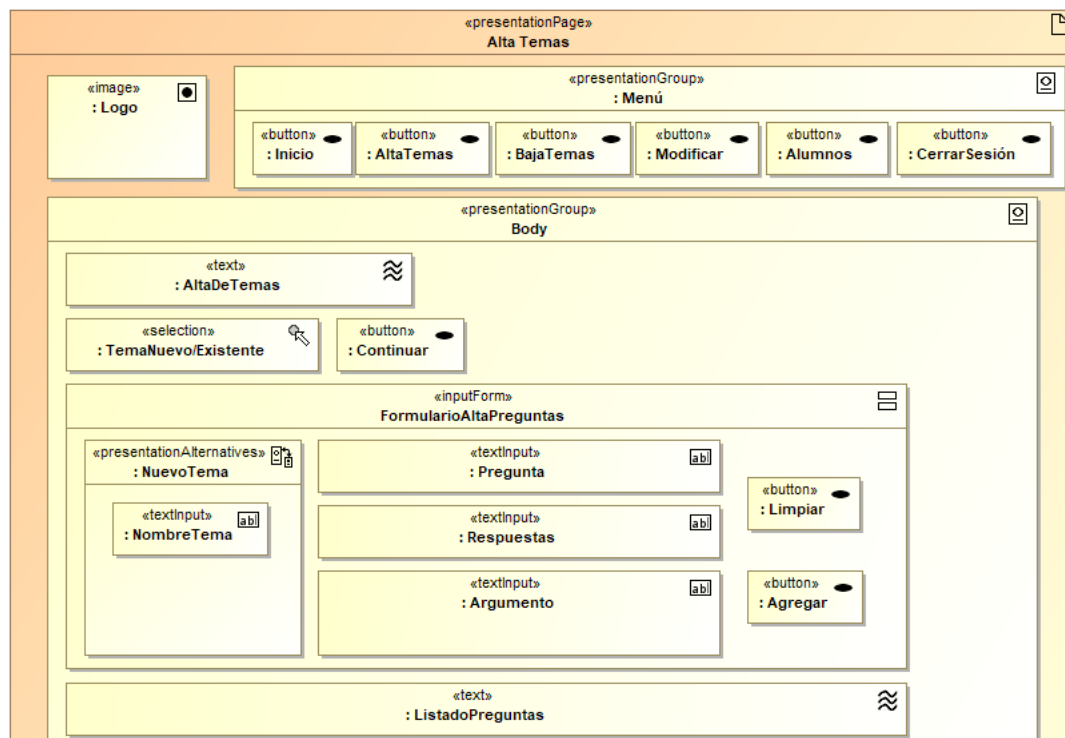


Ilustración 5.17 Diagrama de presentación – Profesor Alta temas

Dar de baja temas es una interfaz con un renderizado condicional, es decir, en base a la opción que elija el usuario se va a presentar un apartado que puede ser: dar de baja un tema en específico o dar de baja una pregunta, como se puede observar en Ilustración 5.18.

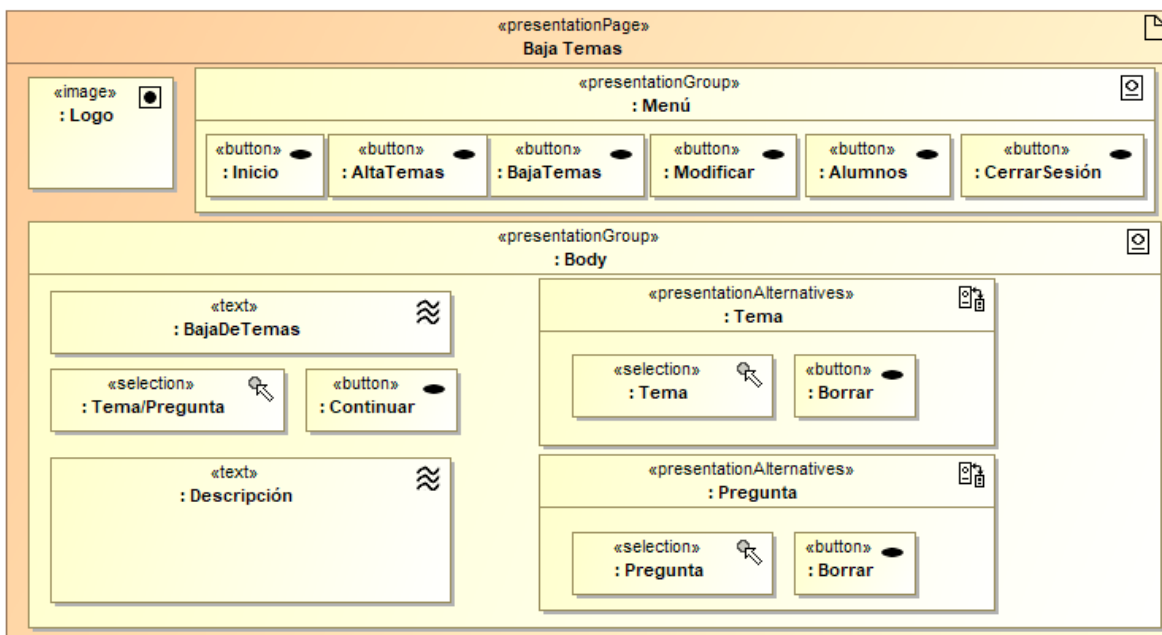


Ilustración 5.18 Diagrama de presentación – Profesor Baja temas

La siguiente página en la Ilustración 5.19 corresponde a modificar preguntas, incluye de un lado la selección del tema y del otro lado un formulario que contiene los datos de la pregunta a actualizar.

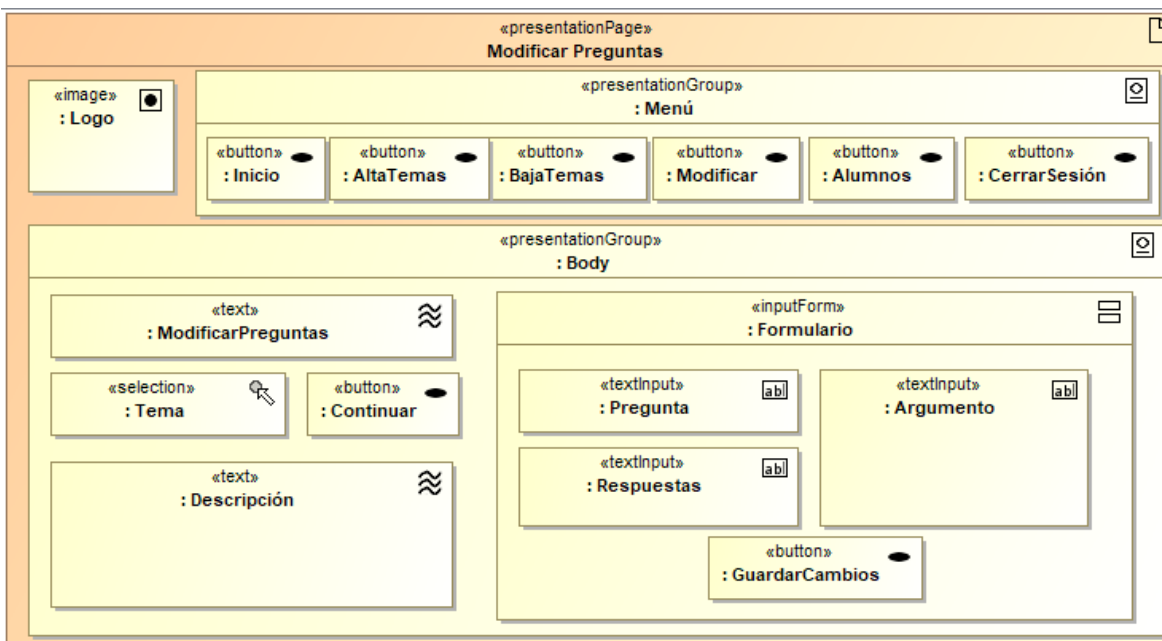


Ilustración 5.19 Diagrama de presentación – Profesor Modificar preguntas

Finalmente, la página de un progreso del alumno consiste en un apartado con dos filtros (nombre del alumno y tema) que nos ayudaran a encontrar con más precisión los datos de un examen, como se visualiza en la Ilustración 5.20.

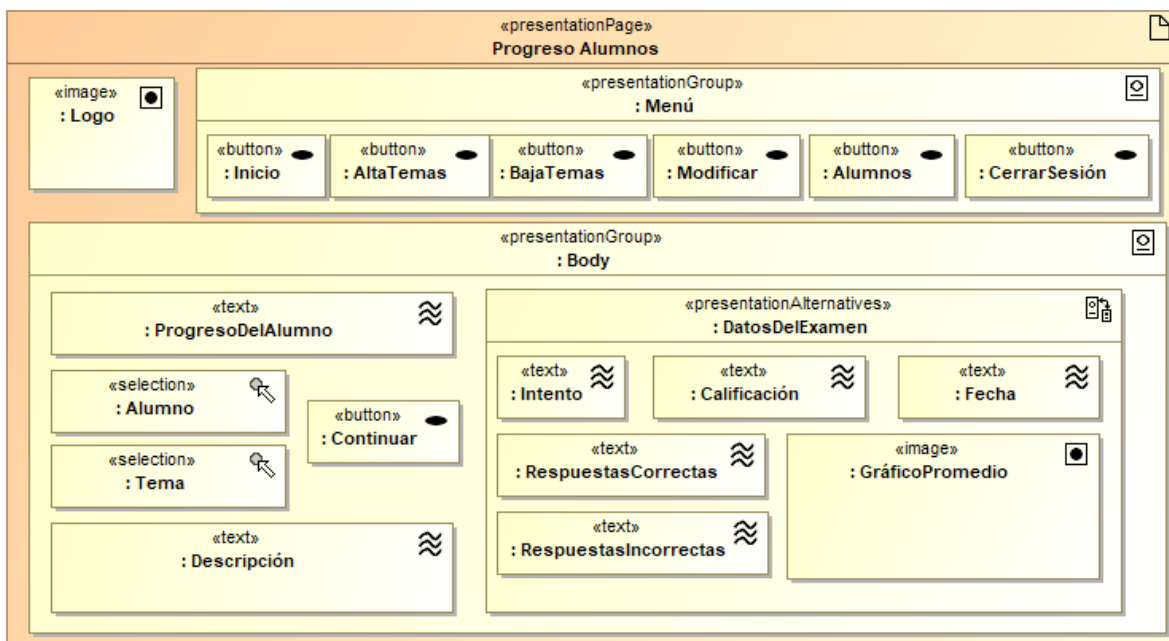


Ilustración 5.20 Diagrama de presentación – Profesor Progreso de alumnos

## Capítulo 6: Estructura y Creación de la Base de Datos

Como definición de base de datos entendemos que es un conjunto estructurado de datos almacenados, nos puede ofrecer recuperación, agregación, modificación y eliminación de los datos. Hoy en día el mercado actual del desarrollo web implementa al menos una base de datos, ya sea estructurada, no estructurada o semiestructurada, por consiguiente, toda aplicación necesita de un recurso para alojar ahí los datos de sus usuarios. En este capítulo se presentan las etapas del diseño de nuestra base de datos, comenzando por el diseño e identificando entidades, para después llevarlo a un diseño lógico y físico respectivamente [10].

### 6.1 Diseño conceptual (Entidad-Relación)

En esta primera etapa está enfocada al diseño de un diagrama entidad-relación, el cual nos permitirá visualizar de manera más fácil la representación de entidades de una base de datos, además de ofrecernos mayor simplicidad de los componentes que participan en un proceso de negocio y el modo en que estos se relacionan entre sí.

El diagrama está conformado por entidades representadas por rectángulos, cada entidad tiene asociada una serie de atributos representados por un círculo donde el atributo principal queda resaltado ya que será el que identificará cada registro y por último se definen las relaciones entre dichas entidades representadas por un rombo.

Se identificaron 5 entidades: usuario, tema, subtema, pregunta e historial.

Relaciones entre entidades:

1. Un usuario puede consultar muchos historiales y evaluarse en muchos temas.
2. Un tema puede ser evaluado por ninguno o muchos usuarios, del mismo modo puede estar presente en ninguno o muchos historiales, cada tema tiene asociado de uno a muchos subtemas y estas disponen de una a muchas preguntas.
3. Cada pregunta está asociada a un único subtema.
4. El historial puede ser consultado por muchos usuarios y puede componerse de uno a muchos temas.

El diagrama presentado en la Ilustración 6.1 es el resultado de este análisis; no se desarrolló en la herramienta MagicDraw debido a que no cuenta con los componentes necesarios para realizarse, en su lugar se recurrió a herramienta online **Moqups** que es una aplicación web simplificada e intuitiva que ayuda a crear y colaborar en esquemas, maquetas, diagramas y prototipos [11].

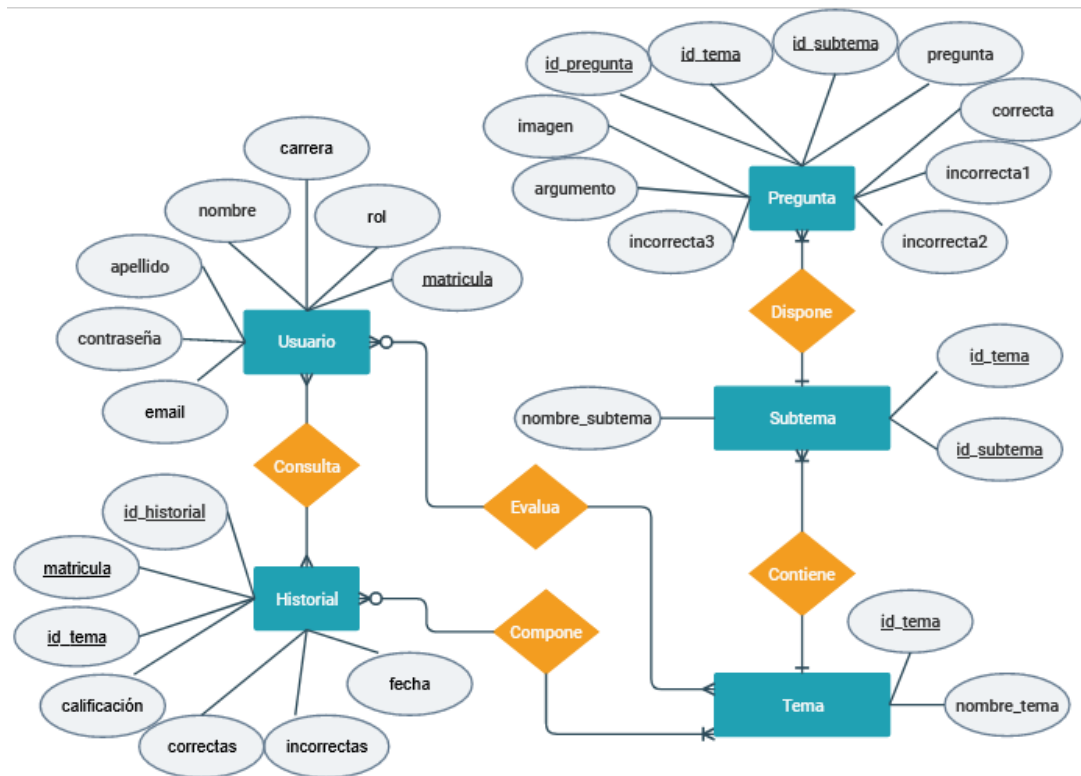


Ilustración 6.1 Diagrama Entidad-Relación  
Realizado en: <https://moqups.com/es/>

## 6.2 Diseño lógico (Modelo Relacional)

En esta segunda etapa se normalizaron las tablas hasta la tercera forma normal, ya que los atributos son atómicos, dependen únicamente de la clave primaria y se eliminan las dependencias transitivas con esto evitamos duplicidad de información y nos ahorramos grandes cantidades de almacenamiento.

En la terminología del modelo relacional formal, una fila se denomina tupla, el encabezado de una columna se denomina atributo y la tabla se denomina relación o entidad. Cada tabla como se mencionó anteriormente tiene un atributo que actúa como identificador puede ser independiente (clave primaria) o compuesto (claves foráneas), nos servirá para tener control de las filas y este valor debe ser único e irrepetible [12].

En cuanto a los tipos de datos a utilizar, en su mayoría son de tipo cadena (text y varchar) donde algunos atributos ya tienen una longitud predefinida para ahorrar espacio en memoria y se implementa “text” para aquellos campos donde la longitud puede variar. Las claves primarias y foráneas generalmente son enteros, también se utilizará tipos de datos como double, date y blob, este último es un objeto binario grande y nos permitirá poder encriptar datos sensibles, en este caso, la contraseña. Es importante mencionar que el atributo de imagen en la tabla pregunta es de tipo text el cual almacenara únicamente el nombre de la

imagen; las imágenes existirán dentro de una carpeta independiente y se consultaran por medio del nombre.

Finalmente, el siguiente diagrama de la Ilustración 6.2 corresponde al modelado final, donde se diseñó en un software de diagramación online: **Lucidchart** que ayuda a individuos y equipos a visualizar procesos, sistemas y estructuras organizacionales [13].

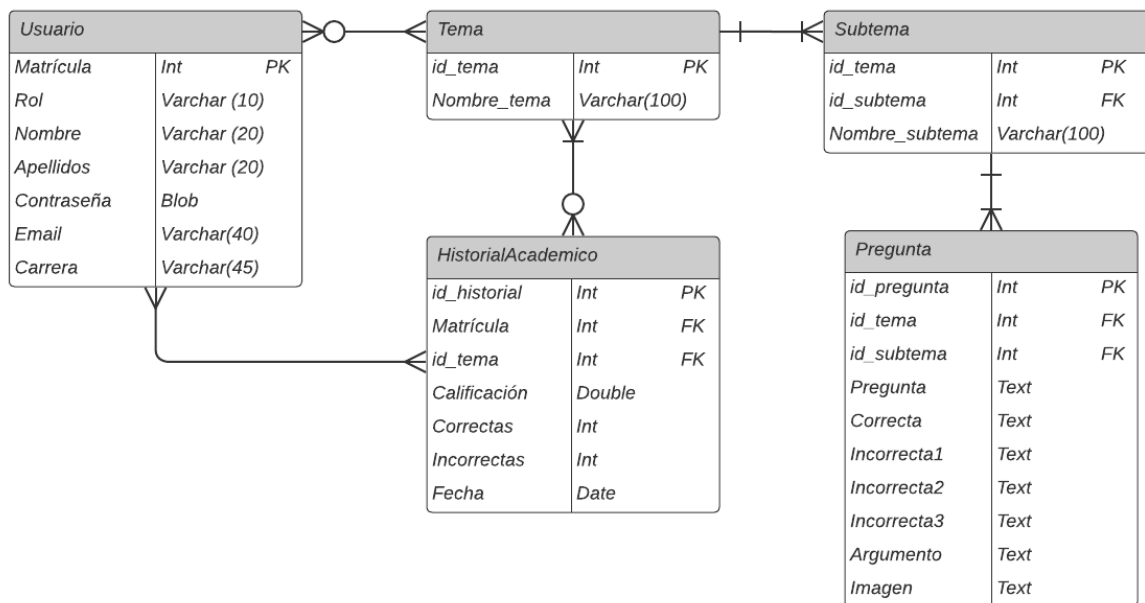


Ilustración 6.2 Modelo relacional  
Realizado en: <https://www.lucidchart.com/pages/es>

### 6.3 Diseño físico (Motor de Base de Datos)

Por último, en esta etapa es importante tener presente cual será el motor de base de datos que actuará como intermediario entre el almacenamiento y nuestra aplicación, además de conocer de cierto modo el lenguaje de consulta. Por ello MySQL siendo un sistema de base de datos relacional que nos ofrece mayor velocidad y flexibilidad, este sistema de gestión funciona bajo el modelo cliente-servidor, cabe señalar que nuestra aplicación web funciona bajo esta arquitectura y su implementación nos brindará una mejor forma de consumir los recursos.

Generalmente las aplicaciones de hoy en día manejan el acrónimo CRUD, que está estrechamente vinculado a la gestión de datos digitales y que resumen las funciones requeridas por el usuario para la gestión de los datos, donde cada una de las letras corresponde a una acción en particular: Create (crear), Read (leer), Update (actualizar) y Delete (borrar) [14].

En base a lo anterior mencionado, utilizo esta forma de manipulación de información aplicada a nuestra base de datos relacional siendo los siguientes ejemplos una muestra del lenguaje SQL:

Create: Creación de un usuario.

```
INSERT INTO usuario
VALUES (201772797, "Alumno", "José", "López", "Buap1234",
"alumno@alumno.buap.mx");
```

Read: Consulta de una pregunta respecto a un tema.

```
SELECT id_pregunta, pregunta, correcta
FROM pregunta
WHERE id_tema=10;
```

Update: Actualización del nombre de un tema.

```
UPDATE tema
SET nombre = "Metodologías Ágiles"
WHERE id_tema=10;
```

Delete: Eliminación de un usuario.

```
DELETE FROM usuario
WHERE matrícula = 201772797;
```

Adicionalmente, se programan procedimientos almacenados que es un conjunto de instrucciones SQL que se almacena asociado una base de datos. Se crea con la sentencia CREATE PROCEDURE mientras que la instrucción CALL invoca al procedimiento, los parámetros se pueden pasar de un lado a otro, omitirse o enumerarse en cualquier orden. Los procedimientos se pueden llamar localmente (en el mismo sistema donde se ejecuta la aplicación) o de forma remota en un sistema diferente [15].

Un claro ejemplo de un procedimiento almacenado es cuando insertamos datos de forma anidada y se requiere de la clave foránea para insertar datos en la siguiente tabla.

Como bien se mencionó con anterioridad, una de las principales herramientas de las cuales se hacen uso a lo largo del presente trabajo de tesis es WAMPSEVER que ofrece tecnologías como:

1. Apache: servidor web.
2. MySQL: gestión de base de datos.
3. PHP: lenguaje de programación.

Este paquete de software nos provee phpMyAdmin con el fin de apoyar en la administración de MySQL a través de una página web. Utilizo este recurso para visualizar de manera más precisa los cambios en las tablas, además de ofrecernos un entorno de desarrollo con interfaz gráfica que se presenta en la Ilustración 6.3 [16].

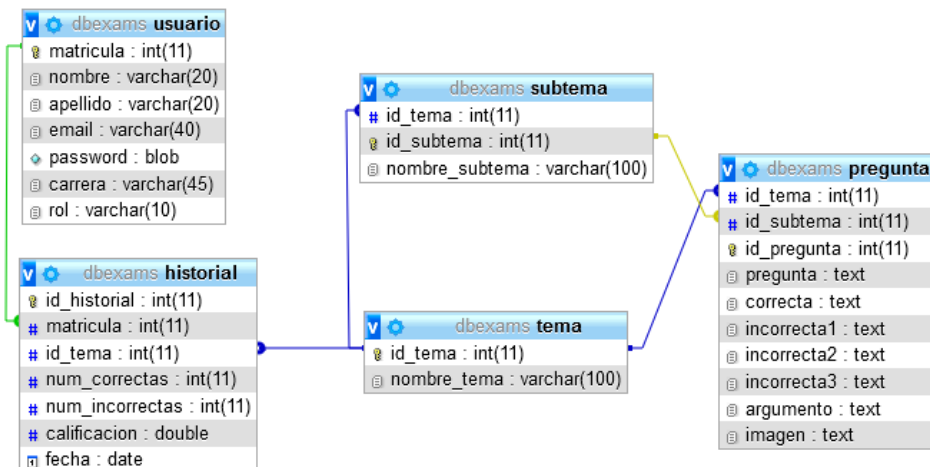


Ilustración 6.3 Interfaz gráfica de las tablas en phpMyAdmin

Para complementar esta sección, en el siguiente capítulo de implementación hablo de las API y como se relacionan con las bases de datos; de esta manera va tomando forma nuestro backend.

## Capítulo 7: Implementación y Pruebas

La implementación aborda el manejo de todo un conjunto de tecnologías que están relacionadas entre sí; diseño, funcionalidad, optimización son algunos de los conceptos a tomar en cuenta al momento de programar.

A nivel de arquitectura la implementación está inspirada en el diseño Clean Architecture que está formada por capas para la administración del proyecto, donde divido por entidades toda la aplicación, cada entidad tiene capas de aplicación donde recae toda la lógica de negocio, capa de infraestructura que contiene todos los servicios de la API REST y la capa de interfaz de usuario que es todo lo que el usuario puede ver y manipular.

A lo largo de este capítulo se resaltarán las partes más esenciales que conforman la aplicación, todo esto con el fin de hacer más legible, eficiente y escalable el proyecto.

### 7.1 Frontend con ReactJS

El diseño es sin duda la parte más atractiva de una aplicación web es la experiencia de usuario e interfaz gráfica, hacer que el usuario se sienta cómodo al momento de navegar y de este modo, hacer más simples las tareas a realizar. Por lo que opte por un diseño sin mucha complejidad poniendo como principal prioridad la función de ejecutar exámenes de una manera fluida y dinámica.

Para garantizar la adaptación a dispositivos móviles o tabletas se elaboró un diseño responsivo a cada página, que acorde a la resolución de la pantalla del dispositivo actual se proporcionara un diseño apto al cliente.

Las validaciones de este lado de la aplicación son para datos proporcionados por el usuario desde los formularios antes de ser procesados y enviados al backend, un claro ejemplo es la validación de la contraseña que debe cumplir ciertas características para dar paso a un registro o inicio de sesión.

Optimizar código es una tarea requerida cuando se programan aplicaciones grandes, en este caso creamos componentes con el fin de mantener el código lo más legible posible. Cada componente tiene sus propias variables, funciones y estilos, las cuales tiene un propósito específico y pueden ser llamadas en todo momento.

Técnicamente hablando se abordaron librerías como:

- Material UI: Ofrece una amplia gama de componentes que pueden ser personalizables, tales como botones, cajas de texto, selectores, barra de navegación, iconos, entre otros.
- React: Uso de hooks que permiten usar el estado y otras características de React sin escribir una clase.

- **React-Router-Dom:** Implementación de rutas dinámicas que permiten a los usuarios navegar entre páginas.
- **React-Redux:** Maneja el estado de las aplicaciones, es decir, es un conjunto de variables globales almacenadas dentro de la aplicación que pueden ser llamadas en todo momento durante la ejecución.
- **Axios:** API HTTP del cliente que permite realizar peticiones a un servidor de forma asíncrona.
- **Chart.js:** Para la generación de gráficos estadísticos.

Se hace uso del `localStorage` y `sessionStorage` que son propiedades del navegador que nos permite almacenar datos de manera local, la diferencia entre estas dos es que `localStorage` almacena información de forma indefinida o hasta que se limpie manualmente y `sessionStorage` almacena información mientras que la pestaña del navegador siga abierta, una vez cerrada, la información se elimina. Estos dos objetos me permiten guardar datos como: identificadores (matrículas) y tokens de acceso [17].

Por otro lado, las imágenes que son parte de la aplicación ya tienen un estilo definido pues son estáticas y no requieren de animación, mientras que las imágenes que son parte de los exámenes y de los formularios si tienen un diseño más acorde a sus dimensiones con el fin de no distorsionarlas.

Con esto garantizamos que la aplicación sea intuitiva y fácil de usar tanto para un alumno como para un profesor.

## 7.2 Backend con Python Flask

El backend es la parte invisible pero fundamental de todo proyecto web, tiene una importancia tan relevante a un punto que la aplicación web no tendría ninguna funcionalidad si este no existiera. Se encarga de todos los procesos y operaciones lógicas necesarias para que funcione la web correctamente y nos ofrece intercambio de información de forma segura. Basado en el framework Python Flask, que nos proporciona una manera fácil de levantar un servidor web compuesto en un entorno virtual, dentro se realizan tareas como validación, gestión de APIS REST y peticiones a la base de datos.

Dependencias / Librerías:

- **Flask:** Contiene todos los módulos requeridos para que el servidor funcione.
- **Flask-CORS:** Permite comunicar el frontend con el backend cuando estos son de orígenes diferentes.
- **Pymysql:** Paquete que permite la interacción con la base de datos MySQL. Conexión, petición y ejecución de consultas.
- **Dotenv:** Para manipulación de variables de entorno.
- **Flas-jwt-extended:** Provee métodos para la generación de tokens.

Una parte fundamental e importante es la ciberseguridad en la aplicación para ello se implementa el uso de los JSON Web Tokens (JWT) que permiten el intercambio de datos en sesión de manera segura entre el cliente y el servidor. El access token es emitido por el servidor y permite el acceso a un recurso protegido en nombre del usuario, este token se usa como método de autenticación y autorización por parte de la aplicación cliente frente al servidor que aloja el recurso [18].

¿Cómo funciona el Access token? Cuando un usuario inicia sesión de forma correcta se le provee un token de acceso bajo las credenciales que proporciona el usuario (matrícula) por un tiempo definido donde podrá utilizar la aplicación con total libertad, pasado este tiempo el token expira y se requerirá iniciar sesión nuevamente, alguno de los recursos protegidos son: la ejecución de un examen, consulta de alumnos, alta y baja de temas, entre otros, en términos generales, los recursos protegidos son nuestras API REST con el fin de proteger la integridad de la aplicación.

La estructura principal que manejo en las API REST está dividida en dos partes, la primera son las rutas estáticas que no requieren de autenticación para ser ejecutadas como lo es el registro de usuarios o ver el temario principal y las rutas protegidas que requieren de una autenticación por parte del usuario. Cada API es una función compuesta por un método http que indica la acción a realizar (método GET, POST, PUT o DELETE), dentro de dicha función se hace una validación de parámetros, se obtiene el body de la petición donde este incluye datos que proporcionó el usuario, se hace una conexión a la base de datos y se ejecuta la consulta a realizar, para finalmente retornar un objeto JSON.

A continuación, un pequeño ejemplo de una función que retorna una lista de subtemas en pseudocódigo:

```
@app.route('dirección-api-url', método<-método-HTTP)
FUNCIÓN obtenerSubtemas (numero: id_tema):
VAR
  objeto:   base_de_datos
  cadena:   consulta <- "SELECT ... FROM ... WHERE ..."
  lista:    resultado
  JSON:     lista_temas
INICIO
  SI método = 'GET' ENTONCES
    conexión <- base_de_datos()
    conexión.ejecutar(consulta, id_tema)
    resultado <- conexión.traerTodo()
    lista_temas <- JSON.convertirJSON(resultado)
    devolver lista_temas
FIN-FUNCIÓN
```

Es importante siempre indicar de que tipo de petición se trata, debido a que se llevan a cabo procedimientos diferentes algunos contienen un body (contiene información adicional), la forma que se estructuran los datos pasan por ciclos, validaciones y desestructuración.

Por otro lado, las imágenes se administran de diferente modo, cuando se hace la petición para cargar la imagen únicamente almacenamos en la base de datos el nombre de la imagen y para evitar duplicidad está compuesta por un identificador seguido del nombre original. Después se procede a almacenar la imagen en una carpeta específica, es muy importante no borrar su contenido porque puede afectar directamente la ejecución de un examen.

### 7.3 Pruebas

El proceso de someter a pruebas la aplicación web es una suma de actividades relacionadas con descubrir errores de contenido, funcionalidad, facilidad de uso, la navegabilidad, desempeño y seguridad. Este proceso comienza sobre aquellos aspectos que son visibles para el usuario y procede a probar dicha tecnología.

A continuación, se presentan las pruebas más importantes, ya que representan las vistas con mayor interacción de datos y conexión entre cliente-servidor.

Registro: Formado por un formulario donde primero se debe indicar que tipo de usuario se hace referencia; datos como la matrícula, correo institucional y contraseña, deben de cumplir con ciertas características antes de llevar a cabo el registro, en caso contrario se le indicará al usuario que errores se presentan en los datos proporcionados, como son: matrícula existente, formato de correo invalido, contraseña mayor a 8 dígitos considerando mayúscula, minúscula y un número o caracteres especiales. Ilustración 7.1 representa el registro.



Ilustración 7.1 Registro de usuarios

Examen (realizado por el alumno): Una vez elegido el tema que se desea evaluar, se presenta la siguiente página en la Ilustración 7.2 que simula la ejecución del examen, con preguntas y respuestas aleatorias. Se pueden destacar los siguientes puntos:

- En la parte superior se muestra el tema y subtema al que pertenece la pregunta, asimismo un conteo de puntos por cada acierto.
- Se puede visualizar un índice que indica en que pregunta va.
- El botón continuar evalúa la pregunta actual y pasa a la siguiente.
- El botón finalizar concluye el examen con los datos obtenidos hasta ese momento.
- Si no se contesta una pregunta se tomará como incorrecta.
- En caso de cerrar sesión o cerrar página a mitad de un examen, se obtendrán resultados negativos que afectarán su rendimiento, por lo que es importante finalizarlo.

Tema: Modelos de Procesos de Desarrollo de Software  
Subtema: Modelos de Ciclo de Vida

Puntos: 1/15

3

Pregunta

Que modelo representa el siguiente diagrama:

Preanálisis → Análisis → Diseño → Desarrollo → Implantación

Validar (¿Se está haciendo el producto?)

Verificar (¿Se está haciendo correctamente?)

Pruebas de unidad → Pruebas de integración → Operación → Mantenimiento

Modelo XP  
 Modelo en V  
 Cascada  
 Desarrollo Incremental

CONTINUAR FINALIZAR

Ilustración 7.2 Ejecución de un examen

Posteriormente se presenta la retroalimentación con las preguntas contestadas y sus respectivos argumentos, con el fin de proporcionarle al alumno una justificación a sus posibles fallas y mejorar su entendimiento a la pregunta. En esta página mostrada en la Ilustración 7.3 podemos ver los datos generales del alumno y datos obtenidos del examen, el botón salir se regresa al menú principal mientras que el nuevo intento conlleva a la ejecución de una nueva simulación.

## Resultados Finales

Alumno(a): Samantha Martínez González	Matricula: 201815141	
Excelente / 93.3	Preguntas correctas: 14	Preguntas incorrectas: 1
<input type="button" value="NUEVO INTENTO"/> <input type="button" value="SALIR"/>		

### Retroalimentación: Introducción a Ingeniería de Software

Subtema: Conceptos Básicos

Pregunta

1.- Se desarrolla para un cliente en particular o para un mercado en general.

- Software
- Hardware
- Documentacion
- Herramienta CASE

#### Argumentación

En ingeniería de software generalmente los productos finales entregado a los clientes son programas (software).

Ilustración 7.3 Resultados de examen y retroalimentación

Alta de temas (realizado por el profesor): Esta sección se realiza por medio de una petición POST al servidor indicando el tema a introducir y bajo que subtema será el primero en formar parte. Los datos que se introduzcan son totalmente válidos, ya que se toman como variables de cadenas, su única restricción es no dejar el campo vacío. La siguiente Ilustración 7.4 representa todo lo anterior mencionado.

### Alta de Tema

Tema  
Introducción a Ingeniería de Software

---

Subtema  
Conceptos Básicos

---

Ilustración 7.4 Formulario para el registro de temas

Después de registrar el tema, se viene el apartado de agregar preguntas, conformadas por el siguiente formulario mostrado en Ilustración 7.5, donde los datos a ingresar deben ser llenados y sin dejar campos vacíos, puede o no existir una imagen.

Se proporciona un botón para limpiar todos los campos, un botón de actualizar por si la pregunta necesita una corrección y un botón de finalizar para volver al menú principal.

Formulario para el registro de preguntas. El formulario contiene los siguientes campos:

- Pregunta
- Correcta
- Incorrecta
- Incorrecta
- Incorrecta
- Argumento

Debajo de los campos, hay un botón "Examinar..." y el texto "Ningún archivo seleccionado." En la parte inferior del formulario, hay cuatro botones: "AGREGAR", "LIMPIAR", "ACTUALIZAR" y "FINALIZAR".

Ilustración 7.5 Formulario para el registro de preguntas

Baja de temas (realizado por el profesor): Aquí únicamente se indica que se desea dar de baja, tema, subtema o preguntas. No contiene restricciones esta sección como se puede observar en la Ilustración 7.6.

Formulario para la baja de temas o preguntas. El formulario está dividido en dos secciones:

- Baja de Tema:** Incluye un campo de selección "Selecciona un Tema" y un botón "CONFIRMAR".
- Baja de Subtema y Preguntas:** Incluye dos campos de selección: "Selecciona un Tema" y "Selecciona un Subtema", y un botón "CONTINUAR".

Ilustración 7.6 Formulario para la baja de temas o preguntas

Rendimiento académico (realizado por el profesor): Este apartado recopila todos los resultados estadísticos seleccionando un tema en específico en relación con un alumno, además de proporcionar datos de contacto como se ve en Ilustración 7.7. El botón editar nos lleva a una página donde se puede reestablecer la contraseña o dar de baja al alumno.

## Rendimiento Académico



Nombre: Samantha Martínez González

Matrícula: 201815141

Carrera: Ingeniería en Tecnologías de la Información

Contacto: [samantha.mtz@alumno.buap.mx](mailto:samantha.mtz@alumno.buap.mx)
[EDITAR](#)

**Selecciona el Tema a Consultar**

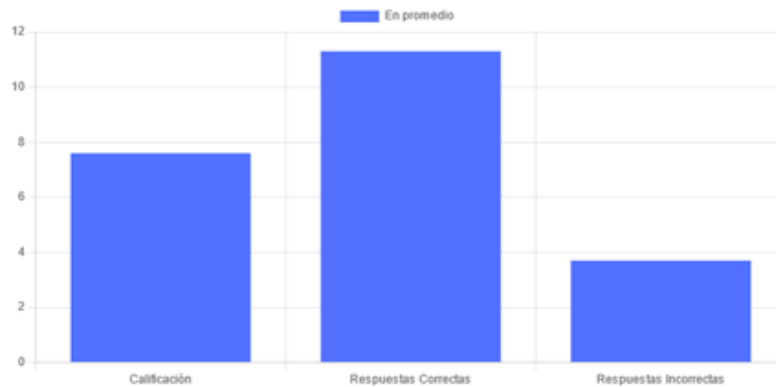
Tema

[CONTINUAR](#) [REGRESAR](#)

Ilustración 7.7 Menú de opciones en rendimiento académico

Indicando el tema a consultar la siguiente Ilustración 7.8 se proporciona un gráfico en promedio al número de intentos realizados, aquí no hay restricciones de por medio.

**Tema: Introducción a Ingeniería de Software**  
Ha realizado 3 intentos


[CONSULTAR OTRO TEMA](#)

Intento	Calificación	Correctas	Incorrectas	Fecha
1	93.3	14	1	01/11/2022
2	60	9	6	13/04/2022
3	73.3	11	4	13/04/2022

Ilustración 7.8 Gráfico estadístico e historial

Entre otras pruebas que se realizan implícitamente se encuentran:

Conexión al servidor: como anteriormente se mencionó es la parte más importante de la aplicación y debe estar ejecutada en todo momento para que la información fluya de manera correcta.

Consultas a la base de datos: están testeadas previamente antes de ser puestas en el backend, solamente se envían los parámetros necesarios para ser ejecutada o mandando a llamar algún procedimiento almacenado.

Registro y validación de usuarios: existen filtros ya mencionados antes de registrar usuarios.

Transición correcta del examen: los datos se mapean del lado del frontend para solamente desplazarse de una pregunta a otra, las únicas formas de tronar un examen son que el token de acceso expire y se tenga que iniciar sesión nuevamente o que el usuario cierre la página manualmente.

## Conclusiones

Este trabajo de tesis tuvo como objetivo diseñar y programar un simulador de exámenes que acompañe al alumno en el aprendizaje de la materia de ingeniería de software, diseñada para los alumnos de la facultad de ciencias de la computación en sus diversos planes de estudio de nivel licenciatura: La ingeniería y la Licenciatura en Ciencias de la Computación. Con base en los antecedentes estipulados en capítulos previos de este documento y en los referentes al análisis, diseño e implementación de la propuesta, se puede concluir que el simulador cumple con los lineamientos mencionados en el protocolo y satisface los requerimientos solicitados; la herramienta de aprendizaje funciona como un recurso adicional para incentivar a los alumnos a mejorar su desempeño académico.

A pesar de existir problemas en el aprendizaje en el alumno para acreditar las materias de Ingeniería de Software se llegó a la conclusión de que con un uso prolongado un alumno puede generar un impacto positivo en sus notas finales, pues es un mecanismo de entrenamiento y de retroalimentación.

Asimismo, la metodología aplicada en este trabajo es un gran acierto pues me brindó un mayor entendimiento al momento de diseñar la aplicación, de modo que cuando comienza la implementación se vuelve más sencillo identificar las prioridades del sistema y con ello, la estructuración de la base de datos se puede derivar del diagrama de clases ya que presentan cierta similitud. Considero que con los estereotipos que maneja la metodología UWE son suficientemente intuitivos y claros para un lector que no cuente con conocimientos UML y tenga una idea sobre los diagramas.

Para la implementación la parte más desafiante sin duda alguna fue la adaptación a la tecnología ReactJS y Python Flask, debido a que no contaba con los conocimientos previos a dichas tecnologías; sin embargo, con los fundamentos de programación y el inglés técnico en la documentación me permitieron darle un mejor sentido a mi aprendizaje y en poco tiempo la programación fue más fluida. Durante todo este proceso he aprendido más sobre el desarrollo web, programación asíncrona e ingeniería de software, reforcé mucho lo aprendido en bases de datos para las consultas, pero sobre todo me quedo con lo nuevo que aprendí, con ganas de poder aplicarlo en el mundo laboral y seguir aprendiendo aún más.

Finalmente, tengo mucha esperanza que mi trabajo sirva para inspirar al desarrollo de simuladores dentro de la facultad o de la misma universidad, ya que considero que el material que proporcionen los profesores son de fuentes aprobadas, más apegadas al plan de estudios e incluso contenido que presenta CENEVAL; por lo tanto, la comunidad estudiantil puede verse beneficiada de este modo.

## Trabajos futuros

Como continuación de este trabajo de tesis y como muchos otros proyectos web, existen diversas formas de mejorar el simulador, las cuales abren las posibilidades de continuar con el trabajo e incluir nuevos aspectos, ya sea en cuanto a diseño o funcionamiento. Cabe mencionar que mis consideraciones no son objeto de esta tesis, su implementación no es requerida en la actual propuesta de tesis; por lo tanto, pueden servir para retomarlas posteriormente por otro desarrollador motivado a mejorar este simulador.

Entre los posibles trabajos futuros considero los siguiente:

1. Valoraciones en las preguntas, en casos donde la pregunta no esté bien formulada o sea difícil de comprender por parte del alumno, ofreciendo la opción de calificar el nivel de entendimiento.
2. Considerar un buzón de sugerencias tanto para el temario como para la misma aplicación y conocer la opinión pública de los alumnos e incluso de los profesores.
3. Incluir animaciones en inicio de sesión, al momento de iniciar o finalizar un examen y ofrecer una mejor experiencia de usuario.
4. Ofrecer la posibilidad de pausar la simulación y reanudarla para más tarde cuando el alumno esté listo.
5. Implementar un apartado de recursos donde los alumnos puedan consultar sitios web o libros recomendados por los profesores, ya sean enlaces o documentos PDF.
6. Con ajustes necesarios el sistema puede escalar y no solo estar enfocada a una sola materia, sino a muchas y con ello poco a poco crear una plataforma de aprendizaje.

## Bibliografía

[1] Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante, España. Editorial Club Universitario.

[2] Silberschatz, A., Korth, H.F. y Sudarshan, S. (2007). *Fundamentos de Bases de Datos*. 5ªed. McGraw-Hill.

[3] Nixon, R. (2014). *Learning PHP, MySQL, JavaScript, CSS & HTML5*. 3ªed. O'Reilly.

[4] IBM Cloud Education. *API REST - ¿Qué es una API REST?* Abril 2021.

Recuperado de: <https://www.ibm.com/mx-es/cloud/learn/rest-apis>

Accedido el 15 de abril del 2022

[5] Iván Jahel Bautista Garcia por Servnet. *Backend y Frontend ¿Qué es y cómo funcionan en la programación?* Ciudad de México, México. Marzo 2021

Recuperado de: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programaci%C3%B3n-de-una-aplicaci%C3%B3n-web>

Accedido el 15 de abril del 2022

[6] Secretaría Académica. *Plan de estudios – Semestral*. Puebla, México. 2016.

Recuperado de: <https://secreacademica.cs.buap.mx/Servicios-Escolares.html>

Accedido el 27 de febrero del 2021

[7] Álvarez Carrión G. *UWE – Definición de metodología UWE*. Academia. 2019.

Recuperado de: <https://www.academia.edu/4493506/UWE1>

Accedido el 07 de marzo del 2021

[8] Nieves Guerrero Citlali, Ucán Pech Juan, Menéndez Domínguez Victor. *UWE en Sistemas de Recomendación de Aprendizaje. Aplicando Ingeniería Web: Un Método en Caso de Estudio*. Universidad Autónoma de Yucatán. Mérida, Yucatán. México. 2014.

Recuperado de:

[https://www.researchgate.net/publication/280580830\\_UWE\\_en\\_Sistema\\_de\\_Recomendacion\\_de\\_Objetos\\_de\\_Aprendizaje\\_Aplicando\\_Ingenieria\\_Web\\_Un\\_Metodo\\_en\\_Caso\\_de\\_Estudio](https://www.researchgate.net/publication/280580830_UWE_en_Sistema_de_Recomendacion_de_Objetos_de_Aprendizaje_Aplicando_Ingenieria_Web_Un_Metodo_en_Caso_de_Estudio)

Accedido el 07 de marzo del 2021

[9] LMU – Ludwig-Maximilians-Universität München. *UWE (UML-Based Web Engineering)*. *Model Examples*. Institute for Informatics. 2021.

[10] Alexys Lozada por EDteam. *Las etapas del diseño de una correcta base de datos relacional*. Lima, Perú. 2018.

Recuperado de: <https://ed.team/blog/las-etapas-del-diseno-de-una-correcta-base-de-datos-relacional>

Accedido el 10 de noviembre del 2021

- [11] Moqups. *Plantillas de diagramas & diagramas de flujo en línea*. 2018.  
Recuperado de: <https://moqups.com/es/>  
Accedido el 15 de noviembre del 2021
- [12] Ramez, E., Shamkant, N. (2010). *Fundamentals of Database Systems*. 6a ed. Addison Wesley.
- [13] Lucidchart. *Programa para Hacer Diagrama de Entidad Relación Online*. Utah. 2008.  
Recuperado de: <https://www.lucidchart.com/pages/es/ejemplos/herramienta-ERD>  
Accedido el 15 de noviembre del 2021
- [14] IONOS. *CRUD: la base de la gestión de datos*. Septiembre 2019.  
Recuperado de: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/crud-las-principales-operaciones-de-bases-de-datos/>  
Accedido el 25 de noviembre del 2021
- [15] Brainbridge, J., Bedoya, H., Bestgen, R., Cain, M., Cruikshank, D., Denton, J. (2016). *SQL Procedures, Triggers, and Functions on IBM DB2 for i*. First Edition. Redbooks. U.S.A.
- [16] Bourdon Romain. *WAMPSEVER*. 2016.  
Recuperado de: <https://www.wampserver.com/en/>  
Accedido el 04 de enero del 2022
- [17] Mora Alexis por EDteam. *¿Qué es y cómo utilizar localStorage y sessionStorage?*  
Lima, Perú. 2019.  
Recuperado de: <https://ed.team/blog/que-es-y-como-utilizar-localstorage-y-sessionstorage>  
Accedido el 13 de mayo del 2022
- [18] Hurtado Ruesga Hector por BBVA. *JSON WEB tokens: claves para usarlos de manera segura*. 2020.  
Recuperado de: <https://www.bbva.com/es/json-web-tokens-jwt-claves-para-usarlos-de-manera-segura/>  
Accedido el 13 mayo del 2022