



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Nogales



SEP

SECRETARÍA DE  
EDUCACIÓN PÚBLICA

IMPLEMENTACIÓN DE ARQUITECTURA MVC EN EL SITIO WEB  
DEL TecNM/ITN DEL LABORATORIO DE MATEMÁTICAS

**TESIS**

QUE PARA OBTENER EL GRADO DE

**MAESTRO EN SISTEMAS COMPUTACIONALES**

PRESENTA

**JORGE IGNACIO MARTÍNEZ VALENZUELA**

DIRECTOR

**M.C. ZINDI SÁNCHEZ HERNÁNDEZ**

**H. NOGALES, SONORA, MÉXICO.**

**OCTUBRE DE 2019.**



"2019, Año del Caudillo del Sur, Emiliano Zapata"

H. Nogales, Sonora, 21/octubre/2019

Oficio No. DEPI/364/2019

**REYNALDO GUTIÉRREZ GUTIÉRREZ**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**  
**PRESENTE**

Por este medio le comunicamos a Usted que el trabajo de Tesis denominado: **"IMPLEMENTACIÓN DE ARQUITECTURA MVC EN EL SITIO WEB DEL TecNM DE LABORATORIO DE MATEMÁTICAS"**, que presentó el alumno **JORGE IGNACIO MARTÍNEZ VALENZUELA**, con número de control 17341004, candidato a obtener el grado de **Maestro en Sistemas Computacionales**, ha sido revisado por los miembros del Comité Tutorial y cubiertas las observaciones realizadas, se Autoriza su Impresión y se Acepta para su Evaluación en la presentación del Examen de Grado.

Agradeciendo de antemano el apoyo brindado al presente, le reitero mi consideración distinguida.

**ATENTAMENTE**

*Excelencia en Educación Tecnológica -  
La Ciencia y la Tecnología para la Liberación del Hombre*

**GUILLERMINA MUÑOZ ZAMORA**  
**REVISORA**

**SIGIFREDO GARCÍA ALVA**  
**REVISOR**

  
**JESÚS RAÚL CRUZ RENTERÍA**  
**REVISOR**



"2019, Año del Caudillo del Sur, Emiliano Zapata"

H. Nogales, Sonora, 21/octubre/2019

Oficio No. DEPI/365/2019

**JORGE IGNACIO MARTÍNEZ VALENZUELA**  
**CANDIDATO A OBTENER EL GRADO DE MAESTRO EN SISTEMAS COMPUTACIONALES**  
**PRESENTE**

De acuerdo con el Reglamento de Titulación del Sistema Nacional de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que el Comité Tutorial realizó, con respecto a su Tesis titulada: **"IMPLEMENTACIÓN DE ARQUITECTURA MVC EN EL SITIO WEB DEL TecNM DE LABORATORIO DE MATEMÁTICAS"**, la División de Estudios de Posgrado e Investigación Autoriza su Impresión.

Agradeciendo de antemano el apoyo brindado al presente, le reitero mi consideración distinguida.

**ATENTAMENTE**

*Excelencia en Educación Tecnológica*  
*La Ciencia y la Tecnología para la Liberación del Hombre*



**SEP**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

**INSTITUTO TECNOLÓGICO DE NOGALES**  
**DIVISIÓN DE ESTUDIOS DE,**  
**POSGRADO E INVESTIGACIÓN**

**REYNALDO GUTIÉRREZ GUTIÉRREZ**  
**JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

RGG/cmmj\*

C.p. Archivo



**IMPLEMENTACIÓN DE ARQUITECTURA MVC EN EL  
SITIO WEB DEL TecNM DEL LABORATORIO DE  
MATEMÁTICAS**

---

## RESUMEN

En los últimos años, las instituciones han estado utilizando plataformas de apoyo como Ambientes Virtuales de Aprendizaje (AVA's) con el propósito de mejorar la calidad de la educación e impactar directamente en el aprendizaje de los alumnos, quienes cada vez más optan por utilizar las Tecnologías Informáticas como un apoyo para su formación académica. Sin embargo, también los jóvenes hoy en día, utilizan tutoriales en YouTube y documentos en internet con solución de ejercicios, siendo otra manera práctica de mejorar su aprendizaje, por lo que la problemática se centra en desarrollar una plataforma que contenga esas herramientas y estén disponibles al alumnado.

Con esta tesis se plantea la implementación de un sitio web el cual sirva de apoyo a los maestros y mejorar el aprendizaje del alumnado en la solución de ejercicios de cálculo diferencial del TecNM/Instituto Tecnológico de Nogales y que pueda servir a todo el sistema del Tecnológico Nacional de México (TecNM), con esta herramienta se podrán crear ejercicios y realizar prácticas de acuerdo a los programas de estudio vigentes que se tengan registradas en el sitio.

Lo que se verá en este documento es el proceso que se realizó para llegar al diseño y aplicación del prototipo que cumple con los requerimientos que se plantearon, donde al aplicar la arquitectura MVC, proporciona una plataforma con ambiente moderno, que incorpora las Tecnologías Informáticas que se utilizan hoy en día en la formación académica.

Una vez que ha sido utilizado en la práctica, se encontraron resultados favorables en el aprendizaje de los alumnos, dando pie a que en una futura implementación, los maestros puedan enriquecer el contenido creando sus propios temarios, ejercicios y video-tutoriales para que sea más adecuado al alumnado del Instituto Tecnológico de Nogales.

---

## ABSTRACT

In recent years, the institutions have been using support platforms for Virtual Learning Environments (AVAs), with the purpose of improving the quality of education and impact directly on the students learning, who increasingly choose to use Information Technologies as a support for their education. However, also young people today, use video tutorials on YouTube and online documents with exercises solutions, being another practical way to improve their learning, so the problem focuses on developing a platform that consolidates those tools and are available to the students.

With this thesis the implementation of a website is proposed, which serves as support to the teachers and improve student learning in the solution of differential calculus exercises of the TecNM / Instituto Tecnológico de Nogales and that can serve the entire system of the Tecnológico Nacional de México (TecNM), with this tool they will be able to create exercises and perform practices according to current study programs that are registered on the site.

What will be seen in this document is the process that was carried out to arrive at the design and application of the prototype that meets the requirements that were raised, where when applying the MVC architecture, it provides a platform with a modern environment, which incorporates the Information Technologies that they are used today in academic training.

Once it was used in practice, favorable results were found in the students' learning, leading to a future implementation, teachers can enrich the content by creating their own guides, exercises and video-tutorials to make it more suitable for students of the Instituto Tecnológico de Nogales.

---

## **AGRADECIMIENTOS**

El logro de haber concluido este proyecto se lo debo al gran apoyo de mi esposa e hijas, es por ello el agradecimiento infinito hacia ellos.

A mi Director Zindi Sánchez Hernández, que me supo guiar por el camino del conocimiento y darme aliento en cada obstáculo a superar.

A mis maestros, M.C. Jesús Raúl Cruz Rentería, M.C. Guillermina Muñoz Zamora, M.C. Sigifredo García Alva y Dr. Samuel González López por ser mis guías en esta fascinante etapa.

A mis compañeros, Alejandra de la Rosa, Roberto León, Aarón Cortez y Humberto Pedraza, que a pesar de encontrarnos con difíciles momentos, los pudimos superar.

A mis padres, quienes desde temprana edad a mis hermanos y a mí, nos fomentaron a seguir adelante en nuestras metas y no parar hasta conseguirlas.

Por último, al TecNM/Instituto Tecnológico de Nogales, quien sin su apoyo, esto no hubiera sido posible, muchas gracias.

---

## DEDICATORIA

Con infinito amor a mi esposa e hijas,  
quienes con su aliento en el día a día, me  
impulsaron a continuar en esta senda del  
conocimiento, haciéndome sentir pleno  
de dicha, satisfacción y agradecimiento.

---

# ÍNDICE

CAPÍTULO 1. ANTECEDENTES .....	1
1.1 Estado del Arte.....	1
1.2 Planteamiento del problema .....	5
1.3 Justificación.....	6
1.4 Objetivos .....	7
1.4.1 Objetivo General .....	7
1.4.2 Objetivos Específicos.....	8
1.5 Hipótesis .....	8
1.6 Alcances y limitaciones del proyecto. ....	8
CAPÍTULO 2. MARCO TEÓRICO.....	10
2.1 Arquitectura MVC .....	10
2.1.1 Introducción .....	10
2.1.2 Por qué utilizar MVC .....	11
2.1.3 Uso en PHP .....	11
2.1.4 Uso en Framework Yii2.....	12
2.2 Modelos.....	13
2.3 Vistas .....	13
2.4 Controladores.....	14
2.5 Arquitectura de aplicaciones MVC.....	14
2.6 Lenguaje PHP .....	15
2.6.1 Introducción .....	15
2.6.2 Visión general .....	15
2.6.3 Características de PHP.....	15
2.7 Características de las interfaces.....	17
2.8 SQL y bases de datos relacionales .....	18
2.8.1 Introducción .....	18
2.8.2 Ventajas de SQL.....	18
2.8.3 Estructuras de datos relacionales .....	18
2.8.4 Normalización .....	20
2.9 Sublime Text 3 Editor .....	21

---

2.10 HTML .....	22
2.10.1 Hojas de Estilo (CSS) .....	22
2.11 Yii2 Codeception .....	23
2.11.1 Tests.....	23
2.11.2 Desarrollo con tests .....	23
2.11.3 Cómo probar y qué probar. ....	24
2.11.4 Preparación del entorno de pruebas .....	25
2.11.5 Modelo CUEVA.....	26
CAPÍTULO 3. ANÁLISIS.....	28
3.1 Método utilizado en la investigación .....	28
3.2 Requerimientos Funcionales .....	29
3.4 Perfil de los actores .....	31
3.4.1 Administrador .....	31
3.4.2 Alumno .....	31
3.5 Casos de uso de alto nivel.....	31
3.5.1 Casos de uso de alto nivel para el módulo “Registro” .....	31
3.5.2 Casos de uso de alto nivel para el módulo “Control de Sesión” .....	32
3.5.3 Casos de uso de alto nivel para el módulo “Opciones del Administrador” .....	33
3.5.4 Casos de uso de alto nivel para el módulo “Opciones del Alumno” .....	34
3.6 Diagramas de casos de uso .....	35
3.7 Diagramas de secuencia .....	37
3.8 Modelo Conceptual de la aplicación .....	42
3.9 Arquitectura de la aplicación.....	43
CAPÍTULO 4. DISEÑO DE LA APLICACIÓN.....	44
4.1 Introducción.....	44
4.2 Diseño de la interfaz del usuario .....	45
4.3 Diseño de la base de Datos.....	47
4.4 Utilización de la herramienta Yii code Generator.....	50
4.4.1 Model Generator .....	50
4.4.2 CRUD Generator .....	52
4.4.3 Formas Dinámicas .....	60
CAPÍTULO 5. PRUEBAS Y RESULTADOS.....	63

---

---

5.1 Usabilidad .....	63
5.2 Pruebas del proyecto en relación a los requerimientos funcionales.....	67
5.3 Pruebas del sitio en diferentes navegadores. ....	69
5.4 Caso de Estudio .....	69
CAPÍTULO 6. CONCLUSIONES Y COMENTARIOS FINALES .....	87
6.1 Análisis de resultados.....	87
6.2 Conclusiones.....	88
6.3 Trabajos futuros y mejoras .....	88
ANEXO.- Código de formas dinámicas Wbraganca. ....	90
REFERENCIAS .....	91

---

## ÍNDICE DE TABLAS

Tabla 2.1 Factores del modelo CUEVA.....	26
Tabla 2.2 Criterio de Juicio de Valor para modelo CUEVA.....	27
Tabla 3.1 Requerimiento funcional del módulo de Registro. ....	29
Tabla 3.2 Requerimientos funcionales del módulo de Control de Sesión.....	29
Tabla 3.3 Requerimientos funcionales del módulo de Opciones del Administrador.....	30
Tabla 3.4 Requerimientos funcionales del módulo de Opciones del Alumno. ....	30
Tabla 3.5 Caso de uso de alto nivel para Registro en Sitio. ....	31
Tabla 3.6 Caso de uso de alto nivel para Inicio de sesión.....	32
Tabla 3.7 Caso de uso de alto nivel para Cierre de Sesión.....	32
Tabla 3.8 Caso de uso de alto nivel para Crear una Materia.....	33
Tabla 3.9 Caso de uso de alto nivel para Crear Unidades. ....	33
Tabla 3.10 Caso de uso de alto nivel para Crear Temas. ....	33
Tabla 3.11 Caso de uso de alto nivel para Crear Ejercicios. ....	34
Tabla 3.12 Caso de uso de alto nivel para Realizar Prácticas.....	34
Tabla 3.13 Caso de uso de alto nivel para Ver mis Calificaciones. ....	34
Tabla 5.1 Evaluación de usuario sin experiencia 1. ....	63
Tabla 5.2 Evaluación de usuario sin experiencia 2. ....	64
Tabla 5.3 Evaluación de usuario con experiencia. ....	64
Tabla 5.4 Evaluación de usuario experto. ....	65
Tabla 5.5 Cumplimiento de Requerimientos Funcionales (parte 1). ....	67
Tabla 5.6 Cumplimiento de Requerimientos Funcionales (parte 2). ....	68

---

## ÍNDICE DE FIGURAS

Figura 2.1 Diagrama de capas MVC .....	10
Figura 2.2 Estructura estática de una aplicación Yii2. ....	12
Figura 2.3 Diagrama MVC. ....	14
Figura 2.4 Menú de inicio de XAMPP. ....	16
Figura 2.5 Interfaz Sublime Text 3 Editor. ....	21
Figura 3.1 Diagrama de caso de uso "Registro" .....	35
Figura 3.2 Diagrama de caso de uso "Control de Sesión" .....	35
Figura 3.3 Diagrama de caso de uso "Opciones del Administrador" .....	36
Figura 3.4 Diagrama de caso de uso "Opciones del Alumno". ....	36
Figura 3.5 Diagrama de secuencia "Registro en Sitio". ....	37
Figura 3.6 Diagrama de secuencia "Inicio de Sesión". ....	38
Figura 3.7 Diagrama de secuencia "Cierre de Sesión" .....	38
Figura 3.8 Diagrama de secuencia "Crear Materia". ....	39
Figura 3.9 Diagrama de secuencia "Crear Unidades". ....	39
Figura 3.10 Diagrama de secuencia "Crear Temas". ....	40
Figura 3.11 Diagrama de secuencia "Crear Ejercicios". ....	40
Figura 3.12 Diagrama de secuencia "Resolver Prácticas" .....	41
Figura 3.13 Diagrama de secuencia "Ver mis Calificaciones". ....	41
Figura 3.14 Mapa Conceptual de la Aplicación. ....	42
Figura 3.15 Arquitectura del patrón MVC de la Aplicación. ....	43
Figura 4.1 Captura de pantalla de menú Inicio. ....	46
Figura 4.2 Captura de pantalla de Inicio de Sesión .....	46
Figura 4.3 Captura de Pantalla de selección de Ejercicios .....	47
Figura 4.4 Tablas con relación directa con Users. ....	48
Figura 4.5 Tablas con relación directa con Ejercicios. ....	49
Figura 4.6 Tablas con relación directa con Prácticas. ....	50
Figura 4.7 Utilización de GII para creación de Modelos. ....	51
Figura 4.8 Código del modelo generado. ....	51
Figura 4.9 Utilización de GII para creación de CRUD. ....	53
Figura 4.10 Pre visualización de los archivos CRUD. ....	53
Figura 4.11 Código de vista form.php. ....	54
Figura 4.12 Código de vista create.php. ....	55
Figura 4.13 Código de vista update.php. ....	56
Figura 4.14 Código de vista index.php. ....	56
Figura 4.15 Código de vista view.php. ....	57
Figura 4.16 Código de vista search.php. ....	58
Figura 4.17 Código de vista MateriaController.php. ....	59
Figura 4.18 Forma Dinámica Wbraganca. ....	60
Figura 4.19 Incisos en forma dinámica con Ejercicios. ....	61
Figura 4.20 Tutoriales en forma dinámica con Ejercicios. ....	61

---

Figura 4.21 Ayuda PDF en forma dinámica con Ejercicios.....	62
Figura 5.1 Grafica comparativa de los resultados. ....	65
Figura 5.2 Prueba unitaria para Login.....	66
Figura 5.3 Captura de pantalla para selección de registro de nuevo usuario. ....	69
Figura 5.4 Captura de pantalla de datos de nuevo usuario. ....	70
Figura 5.5 Captura de pantalla de selección de inicio de sesión. ....	70
Figura 5.6 Captura de pantalla de inicio de sesión.....	71
Figura 5.7 Captura de pantalla de usuario con inicio de sesión. ....	71
Figura 5.8 Captura de Pantalla de selección de Materias, Unidades, Temas o Subtemas. ...	72
Figura 5.9 Captura de pantalla de Alta de Materia. ....	72
Figura 5.10 Captura de pantalla de View después de alta de Materia.....	73
Figura 5.11 Captura de pantalla de Alta de Unidad.....	73
Figura 5.12 Captura de pantalla de View con información de Unidad guardado. ....	74
Figura 5.13 Captura de pantalla de Alta de Tema.....	74
Figura 5.14 Captura de pantalla de View con información de Tema guardado.....	75
Figura 5.15 Captura de pantalla de Alta de Subtema.....	75
Figura 5.16 Captura de pantalla de View con información de Subtema guardado. ....	76
Figura 5.17 Captura de pantalla de entrada de Respuesta. ....	77
Figura 5.18 Captura de pantalla de selección de imagen de respuesta. ....	78
Figura 5.19 Captura de pantalla de View con información de respuesta guardada. ....	78
Figura 5.20 Captura de pantalla de Alta de Ejercicio (sección 1). ....	79
Figura 5.21 Captura de pantalla de ingreso de incisos a un ejercicio.....	80
Figura 5.22 Captura de pantalla de selección de enlaces de apoyo.....	80
Figura 5.23 Captura de pantalla de sección final del ingreso del Ejercicio. ....	81
Figura 5.24 Captura de pantalla de View con información del ejercicio guardado. ....	82
Figura 5.25 Captura de pantalla de menú de Ejercicios Alumno. ....	82
Figura 5.26 Captura de pantalla de selección de ejercicios para una práctica. ....	83
Figura 5.27 Captura de pantalla de ejercicios seleccionados.....	83
Figura 5.28 Pantalla de resolución de ejercicios. ....	84
Figura 5.29 Captura de pantalla de selección de respuestas. ....	84
Figura 5.30 Captura de pantalla de Resultados de Ejercicios. ....	85
Figura 5.31 Captura de pantalla de consulta de prácticas.....	85
Figura 5.32 Captura de pantalla de resultados de una Práctica. ....	86

---

## INTRODUCCIÓN

Hoy en día, el uso de las plataformas web es muy común para el reforzamiento del conocimiento que se imparte en el aula, donde tutoriales en video, dan ejemplos de solución práctica para mejorar la habilidad y entendimiento de los diferentes temas relacionados.

Es por ello que este proyecto de tesis plantea una propuesta para desarrollar un sitio web en la que se tenga la herramienta para gestionar el curso de Cálculo Diferencial del TecNM que se imparte en las carreras que oferta, utilizando un estándar de programación web con relación de bases de datos para el manejo de la información requerida y que contenga herramientas de apoyo como lo son tutoriales en documento y video.

A lo largo de este documento se verá como la arquitectura MVC (Modelo Vista Controlador) es de gran ayuda para la creación de la plataforma, se describirán y pondrán en práctica los puntos necesarios para el desarrollo del software y cómo el proceso en cada uno de los capítulos dio como resultado una plataforma web sólida y amigable que servirá como herramienta de aprendizaje al alumnado.

En el primer capítulo se revisó cómo es el funcionamiento de las plataformas anteriores y algunas que están actualmente en uso para conocer las mejores opciones de aprendizaje y que el sitio contará con las más posibles. En el segundo capítulo se explican los elementos que forman parte de la arquitectura MVC, y cómo con el lenguaje PHP y HTML en conjunto pueden crear un entorno web eficaz y amigable, así como su relación con las bases de datos con SQL. En el tercer capítulo se explica el análisis del sitio, el método utilizado, la revisión de los casos de uso y el modelo conceptual de la aplicación que servirá para su desarrollo. El cuarto capítulo muestra todos los pasos del diseño, tanto su interfaz como la base de datos utilizada.

---

En el capítulo cinco se mencionan las pruebas que existen y cuáles se aplicaron al sitio para comprobar su funcionamiento y revisar su nivel de usabilidad.

En el último capítulo se revisan las conclusiones obtenidas en base a las pruebas realizadas, se proponen trabajos futuros y mejoras para las siguientes versiones de la plataforma.

---

## **CAPÍTULO 1. ANTECEDENTES**

En este capítulo se mencionan las principales plataformas que se utilizan actualmente como herramientas de apoyo para el aprendizaje, de las cuales el alumnado cada vez encuentra sitios más completos en lo que se refiere a contenido y diversidad de los principales temas que cursan.

### **1.1 Estado del Arte**

Para el desarrollo de esta plataforma, se revisaron trabajos anteriores tanto nacionales como internacionales, a continuación se mencionan algunas de las principales.

Un software muy utilizado es ALEKS (Assessment and Learning in Knowledge Spaces), el cual, fue creado por la Universidad de California en 1994 como apoyo a la Fundación de Ciencia Nacional. ALEKS es un programa en línea que maneja tutorías y asesoramiento de Matemáticas, Química, Estadística y Negocios, basado en la teoría de espacios de conocimiento para desarrollar comprensión combinatoria de los principales tópicos que los estudiantes comprenden o no. De esta manera, determina que temas el estudiante está listo para aprender y le permite escogerlos desde módulos interactivos de aprendizaje. En 2013 ALEKS fue adquirido por la compañía de ciencia McGraw-Hill Education (McGraw-Hill Education, 2013).

Como lo menciona un artículo de Scotty, interactuar con ALEKS después de las clases normales, es tan efectivo como tomar clases con maestros expertos en la materia, por lo que su comportamiento y desempeño es equivalente al de las clases que normalmente asisten. La efectividad de utilizar la plataforma ALEKS, definido como un Sistema de Tutoría Inteligente para matemáticas, es un método que se utiliza para estudiar después de clase, como intervención estratégica, para mejorar las habilidades (Scotty D. Craig, 2013).

También se encuentra Khan Academy, el cual fue creado en 2006 por una organización educativa no lucrativa, por el educador Salman Khan, cuyo objetivo fue el de crear una serie de herramientas en línea que ayudara a educar a los estudiantes. La organización produce algunos tutoriales cortos en la plataforma de videos YOUTUBE. Pero además, en su sitio web incluye ejercicios prácticos complementarios y material para docentes.

El idioma que maneja es principalmente en Inglés, pero además se puede encontrar en Español, Portugués, Italiano, Ruso, Turco, Francés, Hindú y Alemán (Khan, 2006).

Derivado de la plataforma Khan Academy, aparece ALAS-KA, que es una herramienta que extiende el análisis de enseñanza, incluyendo nuevas visualizaciones de toda la clase e individualmente a cada estudiante, con más de 20 nuevos indicadores de enseñanza. ALAS-KA ayuda a los maestros para tomar decisiones basadas en los más altos niveles de información disponible, ayudando a estudiantes a ser conscientes de su proceso de aprendizaje por autorreflexión, también permite a los maestros detectar las tendencias de las clases y utilizar modelos de aprendizajes adecuados. Estos cursos (Física, Química y Matemáticas) para alumnos de nuevo ingreso fueron desarrollados en la Universidad Carlos III de Madrid, España (UC3M) y fueron utilizados por más de 300 estudiantes (José A. Ruipérez-Valiente, 2015).

Otro de los más utilizados es Moodle (modular object-oriented dynamic learning environment), el cual es de categoría de código abierto y libre al público, orientado a la administración de sistemas de aprendizajes. Desarrollado en un inicio con fines pedagógicos en 2002, por Martin Dougiamas, para ayudar a docentes donde ellos crearan cursos en línea con la intención de manejar interacción y colaboración de contenido. Hoy en día es usado para cursos a distancia, clase interactiva compartida, aprendizaje electrónico en escuelas, universidades, centros de trabajo, entre otros (Dougiamas, 2002).

En un artículo de Science Direct (Princely Ifinedo, 2017), los factores de uso son positivos relacionado a los estudiantes que utilizan Moodle, es un modelo tecnológico aceptado, con un sistema de información continuo, con una gestión de aprendizaje gratuito, que permite crear formaciones en línea avanzadas, flexibles y atractivas, donde el papel que actúa el profesor y el alumno es más activo y de interés para ambos.

Algunas de las ventajas que presenta esta plataforma:

- Las páginas Web pueden explorarse en cualquier orden.
- Los cursos contienen chats en vivo entre los estudiantes y los profesores.
- Existen foros en los que los usuarios pueden puntuar los mensajes por su relevancia o conocimiento.
- Hay talleres en línea que permiten a los estudiantes evaluar el trabajo de los demás.
- Hay encuestas espontáneas que permiten al profesor evaluar la opinión de los estudiantes durante el progreso de un curso.
- Existen directorios configurados de forma individual para que los profesores puedan subir y compartir sus archivos.

Todas estas características crean un entorno activo de formación, lleno de distintos tipos de interacciones estudiante-estudiante y estudiante-profesor.

En Medellín, en la universidad de San Buenaventura surge la necesidad de investigar el por qué había tantos alumnos reprobados en las asignaturas de matemáticas, la cual resultó en que los alumnos estaban fuertemente ligados a la cultura digital. Un análisis arrojó como resultado que una base sólida para el aprendizaje es la motivación y que la herramienta didáctica impacte realmente a los alumnos, eso mismo ocasionó que se desarrollara la utilización de Ambientes Virtuales de Aprendizaje (AVA's).

Los cuales, con ciertos protocolos en su diseño que permitieran obtener interesantes resultados en el enfoque de pensamiento matemático y administración del conocimiento para la solución de problemas, en este caso se utilizó la plataforma MyMathlab (Bravo, 2011).

Oscar Mora publicó en el 2012 lo siguiente: “la terminología laboratorio de matemáticas, no es más que mediar el proceso de enseñanza-aprendizaje utilizando las herramientas que ofrece el software de matemáticas, generando así, una serie de aplicativos interactivos con la posibilidad de interactuar y experimentar con las figuras” (Mora Arroyo, 2012).

El proyecto de Oscar Mora tuvo como objetivo planear herramientas didácticas para ser utilizadas en AVA's a través de un Sistema de Gestión de aprendizaje para agregarlas al programa de matemáticas básicas de los grupos de educación media y superior.

Anteriormente el Instituto Tecnológico de Nogales, para sus cursos propedéuticos utilizó una aplicación llamada *MyMathLab* de *Pearson* por la cual la institución pagó una cuota por el derecho de utilizar dicha aplicación.

La plataforma *MyMathLab* le permite al profesor de la asignatura programar ejercicios propuestos, actividades y exámenes los cuales la misma aplicación se encarga de evaluar. También cuenta con la capacidad de relacionar el contenido de la materia con libros de *Pearson*.

Es la colección de productos para tareas, tutoriales y evaluaciones en línea más grande del mundo, y está diseñada con un sólo propósito en mente: mejorar los resultados de cada uno de los alumnos de educación superior.

Con la retroalimentación de más de 11 millones de alumnos anualmente, MyMathLab y Mastering crea experiencias de aprendizaje que en verdad son personalizadas y que se adaptan continuamente.

MyMathLab y Mastering reaccionan según el desempeño real de cada alumno, ofreciendo ayuda específica que les permite comprender mejor el material del curso y entender los conceptos difíciles. (Pearson, 2018).

Un sistema realizado por Jorge Vega, alumno del ITN, fue desarrollado con el objetivo de atender las necesidades académicas de esta institución, el sitio web funciona de manera que se dan de alta instructores, alumnos y también un catálogo de libros con sus respectivos capítulos y temas, por lo que cada instructor puede asignar tareas y ejercicios a resolver por ciertos alumnos y una vez completado, aplicar exámenes. Sin embargo la interfaz no contiene aspectos gráficos que despierten o aumenten el interés del alumno para que continúe con su aprendizaje (Vega, 2015)

## **1.2 Planteamiento del problema**

Anteriormente, el TecNM/Instituto Tecnológico de Nogales trabajaba con la plataforma MyMathlab, donde se pagaba una licencia por cada estudiante de \$700 pesos anuales, y se les proporcionaba acceso a cada uno mediante un usuario y contraseña, así como también un libro de ejercicios de la editorial Pearson, sin embargo, no todos los docentes y alumnos la utilizaban porque se les hacía tediosa y sin resultados reflejados en su aprendizaje.

En la actualidad no se cuenta con una plataforma propia, la cual pueda utilizarse para llevar a cabo las tareas requeridas por los docentes, no se cuenta con un sitio web donde el alumnado pueda reforzar los conocimientos iniciados en clase, aunque existen otras plataformas donde es necesario pagar una suscripción y no todas ellas siguen el mismo lineamiento de contenido que sea de completa utilidad para el plan de estudios de las principales materias de matemáticas, físicas y química.

Para que una plataforma sea de utilidad para el alumnado de los Institutos Tecnológicos, hay que manejar un estándar en los ejercicios, donde la mayoría de las plataformas gratuitas no cuentan con esa unificación de ejercicios y en ocasiones, es posible confundir al alumnado con el amplio contenido que manejan.

Por ello, al no contar con una plataforma personalizada y de uso gratuito, el alumnado busca otras fuentes de apoyo y en ocasiones optan por realizar pago por tutorías o libros, así mismo, algunos docentes buscan opciones como documentales en línea o video-tutoriales para reforzar los temas vistos en clase.

### **1.3 Justificación**

Las tecnologías de la información y la comunicación tienen un gran potencial para divulgar el conocimiento, fomentar un aprendizaje más efectivo y desarrollar servicios educativos más eficientes.

Hoy en día el proceso enseñanza-aprendizaje ha superado la barrera de las aulas escolares y utilizando la tecnología al alcance, se tiene la facilidad de obtener información de temas de interés en sitios web de organizaciones privadas y públicas. Es por ello que son cada vez más las instituciones dedicadas a desarrollar software para ayudar a mejorar el aprendizaje de los estudiantes (UNESCO, 2005).

Al utilizar la arquitectura MVC en el sitio web, permitirá que las aplicaciones se comuniquen en una forma que no depende de la plataforma ni del lenguaje de programación, teniendo una ventaja de manejar datos con mucho más facilidad y permitir una comunicación más libre entre los software, donde cada módulo se puede ver como una unidad de trabajo y es donde cada uno maneja una tarea funcional específica.

El sitio desarrollado sirve para evitar pagar rentas por plataformas similares, ahorrar en el gasto de libros que no se encuentren disponibles en biblioteca, y utilizar esta herramienta para su aprendizaje en su continua preparación para mejorar en sus exámenes y evitar el pago de asesorías externas, ya que en ocasiones, no es posible que los alumnos asistan a las que se otorgan gratuitamente en la institución, ya sea por su horario o alguna otra actividad.

El presente proyecto tiene como objetivo utilizar técnicas que se manejan hoy en día, mejorando la arquitectura del sitio web, dándole mayor eficiencia y a la vez, crear una interfaz que sea más atractiva tanto para el personal docente, como al alumnado, agregando aspectos gráficos que despierten el interés del alumnado y que su aprendizaje crezca y así, se refleje directamente en las calificaciones de las materias relacionadas con matemáticas.

La viabilidad para el desarrollo y cumplimiento de las expectativas de este proyecto fueron revisadas y partiendo de que hay antecedente de un proyecto anterior donde se puede mejorar la interfaz, los aspectos técnicos y cambiar su arquitectura, se concluyó que sí es factible su realización e implementación.

Los usuarios potenciales en la propia institución son más de 3,000 estudiantes y de todo el sistema del Tecnológico Nacional de México son más de 500,000.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Desarrollar un sitio web para un Laboratorio de Matemáticas, utilizando arquitectura MVC con programación PHP, con soporte de bases de datos para poder administrar ejercicios, prácticas y almacenar los resultados para su uso posterior, con la finalidad de ayudar al rendimiento del alumnado del TecNM/Instituto Tecnológico de Nogales.

### **1.4.2 Objetivos Específicos**

1. Utilizar arquitectura MVC.
2. Diseñar e implementar el módulo de administración de ejercicios, en el cual se añadirán materias, unidades, temas y subtemas.
3. Diseñar e implementar un módulo para que el alumno pueda realizar prácticas y consultar el resultado de las mismas.
4. Desarrollar la base de datos correspondiente utilizando los criterios de normalización necesarios.
5. Publicar el sitio web de manera local y en la red “Depi” para realizar pruebas y validaciones.

### **1.5 Hipótesis**

Es posible implementar un ambiente virtual para la administración y uso de ejercicios de Cálculo Diferencial utilizando la Arquitectura MVC y almacenar sus resultados en una base de datos para su administración y control.

### **1.6 Alcances y limitaciones del proyecto.**

Herramientas utilizadas:

- Se utilizó la arquitectura MVC para el desarrollo del sitio web.
- Se utilizó la plataforma XAMPP v3.2.2.
- Como sistema gestor de base de datos se utilizó MYSQL y el tipo de base de datos será con el modelo relacional.
- Se utilizó lenguaje PHP versión 5.36.
- Se utilizó el framework Yii2 para describir, producir, consumir y visualizar los servicios.

El sitio no realiza exámenes, ya que esta versión del sistema se enfoca en la práctica de ejercicios para que su aprendizaje se refleje en los exámenes de su asignatura, tampoco lleva seguimiento de calificaciones por alumno en sus diferentes prácticas.

No cuenta con la opción de inscribirse a cursos de algún docente en particular, pero que en versiones posteriores, estas opciones pueden ser agregadas para que sea un sistema más completo.

La aplicación cuenta con una interfaz web desde donde se puede hacer el inicio de sesión, los usuarios con rol de maestro tendrán a su disposición la posibilidad de dar de alta materias, unidades, registrar temas y subtemas, agregar ejercicios a los temas disponibles, los alumnos podrán seleccionar los ejercicios a responder, revisar su calificación y guardar la práctica correspondiente.

La aplicación no evalúa los procedimientos utilizados por los alumnos para la resolución de problemas, solo se evalúa la respuesta final.

---

## CAPÍTULO 2. MARCO TEÓRICO

En este capítulo se muestra cómo es la arquitectura en capas del Modelo-Vista-Controlador para aplicarse al diseño del proyecto y el desarrollo de cada una de ellas y que en conjunto con los lenguajes PHP y HTML crean un entorno web sólido con una interfaz amigable.

### 2.1 Arquitectura MVC

#### 2.1.1 Introducción

MVC es un patrón de diseño utilizado para el desarrollo de software con interfaces de usuario, creado a partir de la necesidad de tener sistemas más robustos, donde el mantenimiento, separación de funciones y reutilización de código sea más flexible.

En su diseño se utilizan tres capas diferentes, separadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores o lo que es lo mismo, *Model*, *Views* & *Controllers*, como se muestra en la figura 2.1.

#### Arquitectura y diseño: Patrón MVC

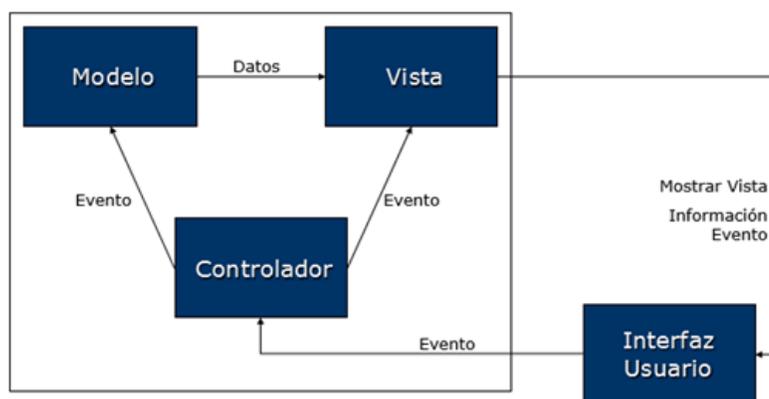


Figura 2.1 Diagrama de capas MVC

MVC fue una idea creada ya hace tiempo, incluso antes de la web, pero en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de una variedad de *frameworks* de desarrollo web que utilizan el MVC como modelo para la arquitectura de las aplicaciones web (Alvarez, 1999).

### **2.1.2 Por qué utilizar MVC**

En ingeniería del software se crean sistemas de calidad donde se incluyen factores como estructuración de los programas o reutilización de código, lo que ocasiona que el desarrollo y mantenimiento sea de mayor facilidad.

Una de las soluciones que concluyen los ingenieros de software con el paso del tiempo respecto a los procesos de desarrollo de sistemas, es la utilización de la arquitectura basada en capas que separan el código en función de sus responsabilidades, por lo que MVC está ahí para facilitar el desarrollo de aplicaciones con mayor calidad en menor tiempo (Alvarez, 1999).

### **2.1.3 Uso en PHP**

Cuando se programa en PHP, cualquier persona combina tanto el código PHP como el código HTML (e incluso el Javascript) en el mismo documento. Esto produce lo que se denomina el "Código Espagueti". Si se desea cambiar la presentación del contenido, se está obligado a repasar todas y cada una de las páginas que tiene el proyecto. Sería mucho más útil que el HTML estuviera separado del PHP (Alvarez, 1999).

### 2.1.4 Uso en Framework Yii2

Yii2 maneja el diseño de patrón modelo-vista-controlador (MVC), el cual se utiliza fuertemente en la programación web en los últimos años. MVC separa la lógica del negocio de las consideraciones de la interfaz de usuario para que los programadores puedan modificar cada parte fácilmente sin afectar a la otra. En MVC el modelo representa los datos y las reglas del negocio; la vista contiene elementos de la interfaz de usuario como textos, formularios de entrada y el controlador gestiona la comunicación entre la vista y el modelo.

Yii2 maneja un *front-controller* conocido también como aplicación, el cual representa el contexto de ejecución del procesamiento del pedido. La aplicación procesa el pedido del usuario y la regresa al controlador apropiado para tratamiento futuro (Alvarez, 1999).

En la figura 2.2 se muestra el diagrama de la estructura estática de una aplicación Yii2.

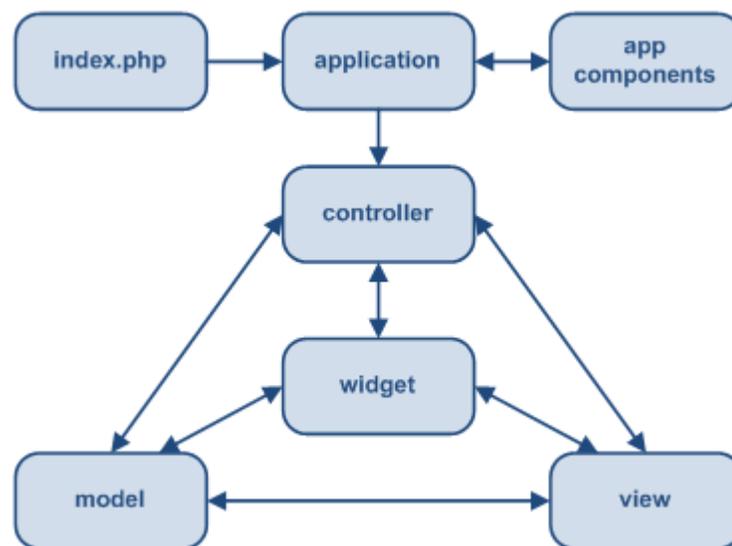


Figura 2.2 Estructura estática de una aplicación Yii2.

Como se puede observar en la figura 2.2, el inicio es mediante la página *Index.php*, de ahí nos envía a la aplicación y carga los componentes. Cualquier petición de la aplicación es directamente enviada al controlador y éste a su vez, administra la interacción entre las otras partes como lo son el modelo, la vista y componentes auxiliares.

## **2.2 Modelos**

Es la capa que procesa los datos, contiene los mecanismos para el ingreso de la información y también para actualizar su estado. Los datos se tienen almacenados en una base de datos, por lo que en los modelos se tendrán todas las funciones que acceden a las tablas y harán las correspondientes llamadas *selects*, *updates*, *inserts*, entre otras.

Al utilizar MVC lo normal también es manejar otras librerías como PDO o algún ORM como Doctrine, donde se puede trabajar con abstracción de bases de datos y persistencia en objetos. Por ello, en vez de usar directamente sentencias SQL, que suelen depender del motor de base de datos con el que se esté trabajando, se utiliza un dialecto de acceso a datos basado en clases y objetos (Alvarez, 1999).

## **2.3 Vistas**

Las vistas, contienen el código que despliega la visualización de las interfaces de usuario, es decir, el código que permitirá conocer los estados de la aplicación en HTML. En las vistas nada más se tienen los códigos HTML y PHP que permite mostrar la salida.

En la vista se trabaja con los datos, sin embargo, no se accede directo a éstos. Las vistas solicitarán los datos a los modelos y ellas generarán la salida, tal como la aplicación lo requiera.

## 2.4 Controladores

Gestionan el código para responder a las acciones que solicita el usuario en la aplicación, como visualizar un elemento, realizar una compra, requerir una búsqueda de información, entre otros.

Es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan necesitarse para implementar las necesidades de la aplicación. Sin embargo, su responsabilidad no es manipular directamente los datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo (Alvarez, 1999).

## 2.5 Arquitectura de aplicaciones MVC

En la figura 2.3 se muestra un diagrama que ayuda a entender un poco mejor cómo colaboran las distintas capas que componen la arquitectura de desarrollo de software en el patrón MVC (Alvarez, 1999).

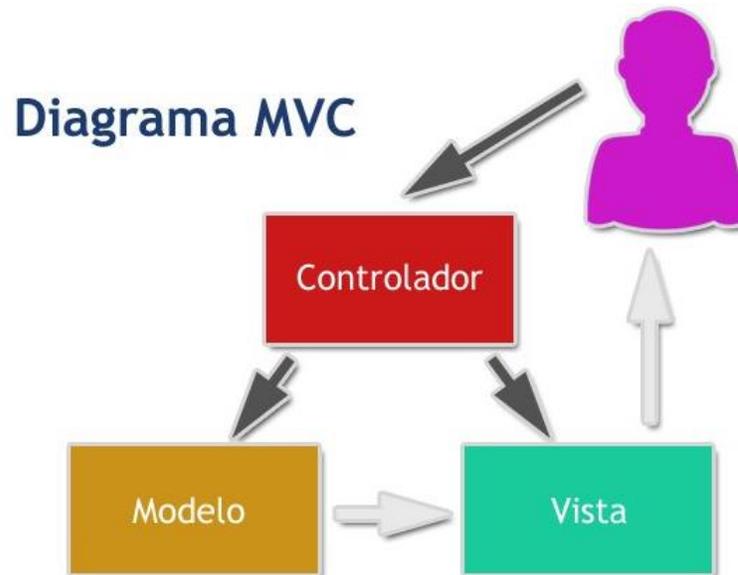


Figura 2.3 Diagrama MVC.

## 2.6 Lenguaje PHP

### 2.6.1 Introducción

PHP, que significa *Hypertext Preprocessor* (procesador de hipertexto), es un lenguaje de programación diseñado para el desarrollo web de contenido dinámico. Fue de los primeros lenguajes donde se podían incorporar directamente en un archivo HTML en lugar hacerlo en un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.

PHP se ha fortalecido agregando ahora una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas. Es utilizado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas de forma gratuita.

### 2.6.2 Visión general

El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. El número de sitios basados en PHP se ha visto reducido progresivamente en los últimos años, con la aparición de nuevas tecnologías como Node.JS, Golang, ASP.NET, entre otros.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache e ISAPI (PHP GROUP, 2001).

### 2.6.3 Características de PHP

#### 2.6.3.1 Características

- Se enfoca al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

- Es fácil de aprender, ya que se simplificaron algunas especificaciones.
- El código es invisible al navegador web y al cliente.
- Tiene capacidad de conexión con la mayoría de los motores de base de datos como con *SQL*, *MySQL* y *PostgreSQL*.
- No es necesario definir tipos de variables, se evalúan en tiempo de ejecución (PHP GROUP, 2001).

### 2.6.3.2 Desventajas

- Al interpretarse en ejecución, puede ser un inconveniente que el código fuente no pueda ser ocultado.
- Al ser un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel.

### 2.6.3.3 XAMPP

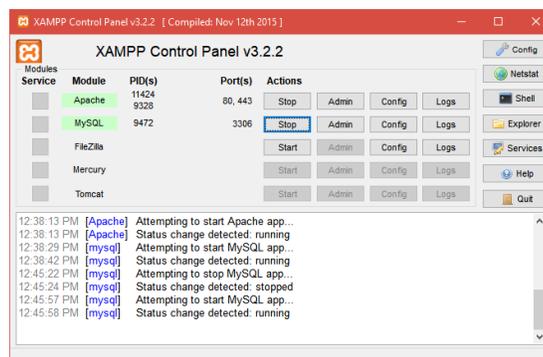


Figura 2.4 Menú de inicio de XAMPP.

XAMPP es un servidor independiente de plataforma de software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas (PHP GROUP, 2001).

XAMPP se utiliza para el diseño local de la plataforma, donde simula las mismas características de un servidor web y poder ejecutar cada una de las páginas para poder realizar pruebas y corregir errores.

## **2.7 Características de las interfaces**

De acuerdo a los autores Weinschenk, Jamar y Yeo en su libro *GUI Design Essencials*, algunas de las características y principios sobre los cuales se deben basar las páginas Web son las siguientes:

- Mostrar contenidos significativos, donde se muestre lo que el usuario quiere saber, no lo que el desarrollador quiere decir.
- Priorizar la usabilidad.
- Manejar un diseño unificado utilizando el mismo tipo de imágenes y colores en todo el sitio.
- Utilizar una jerarquía para facilitar la comprensión de la estructura del sitio.
- Tener la página principal como el menú para el resto del sitio.
- Manejar los archivos pequeños para que el tiempo de carga de las páginas sea rápido.
- Proveer señales al usuario para no perderse en el sitio.
- Manejar la regla de los tres clics para llegar a la información buscada.
- Utilizar links con etiqueta reducida y clara.
- Evitar manejar colores que causen fatiga o que distraigan al usuario (Weinschenk, 1997).

## 2.8 SQL y bases de datos relacionales

### 2.8.1 Introducción

*SQL (Structured Query Language)* es un lenguaje de alto nivel para la gestión de bases de datos basado en el modelo relacional (Davidson, Pro Sql Server , 2008).

### 2.8.2 Ventajas de SQL

Una simple instrucción en *SQL*, representa cientos y miles de líneas de código, donde la mayor parte se ejecuta físicamente accediendo a datos en los discos de almacenamiento, convirtiéndolos en registros y realizando operaciones en el CPU.

Un objetivo fundamental es utilizar un set basado en operaciones. Las tablas de *SQL* son construidas para trabajar eficientemente en grandes cantidades de datos a la vez (Davidson, Pro Sql Server , 2008)

### 2.8.3 Estructuras de datos relacionales

Esta sección presenta las siguientes estructuras de bases de datos relacionales básicos y conceptos:

- **Base de datos y el esquema**

Una base de datos es una colección estructurada de hechos. En *SQL* la base de datos es el contenedor de nivel más alto que se va utilizar para agrupar los objetos y el código que sirven a un propósito en común.

Utiliza esquemas para agrupar objetos en la base de datos con los temas comunes y se pueden acceder conociendo la base de datos en que residen y el esquema, por ejemplo: `databaseName.schemaName.objectName`.

- **Las tablas, filas y columnas**

En *SQL*, las tablas son una representación de los datos de los que se han eliminado todos los aspectos de la aplicación. El objetivo de la teoría relacional es liberarte de las limitaciones de las rígidas estructuras de una aplicación como una hoja de cálculo de Excel, aunque estos términos pueden llegar a ser muy confusos.

Una entidad se suele implementar como una tabla con los atributos de la definición de las columnas de esa tabla y cada instancia como una fila:

- La columna representa los datos a su nivel más bajo que se necesitan para trabajar en *SQL*.
- Así, que teniendo una columna que se define como dos valores independientes, dígame una *Columna.X* y *Columna.Y*, es perfectamente aceptable, mientras que definiendo una columna para lidiar con valores como '1,1' no lo sería, debido a que el valor debe ser descompuesto para ser útil.

- **El principio de la Información**

Toda la información en una base de datos relacional se representa explícitamente en el nivel lógico exactamente de una manera por los valores de las tablas, lo que significa que hay sólo una forma de acceder a los datos en una base de datos relacional y que es mediante la comparación de los valores en las columnas.

Lo que es importante es que el código rara vez se necesita preocuparse dónde están los datos. Sólo tiene que abordar los datos por su nombre y éstos serán recuperados.

- **Llaves**

En teoría relacional, a una relación no se le permiten tener llaves duplicadas, por lo que las tablas deben tener al menos una llave definida para asegurarse de que todas las filas sean únicas.

#### **2.8.4 Normalización**

La normalización es el proceso en una base de datos relacional que va dirigido a convertir los esquemas relacionales en una forma más deseable, donde el objetivo es remover la redundancia en las relaciones y los problemas presentados a raíz de esto.

El proceso de normalización con cada paso transforma a los esquemas relacionales a una forma normal más alta. Cada forma normal contiene la misma o inclusive mayor optimización de las formas normales de más bajo nivel.

Las formas normales de mayor uso son:

- Primera Forma Normal (1NF)
- Segunda Forma Normal (2NF)
- Tercera Forma Normal (3NF)

Una relación es considerada que está en primera forma normal si todos sus atributos tienen dominios que son indivisibles o atómicos. Es decir, que en ninguno de sus atributos o columnas se tengan datos o información que sea posible dividir, como por ejemplo, el caso de la información de un contacto que tiene más de un número de teléfono.

Se dice que una relación está en segunda forma normal, cuando está en 1NF y no tienen algún atributo que no sea llave que dependa en parte de la llave candidata, sino que dependa completamente de la llave candidata.

También se puede partir de esa definición que, una relación que tiene un solo atributo como su llave candidata está siempre en 2NF, y se considera en tercera forma normal (3NF) si está en 2NF y no hay atributo no-llave que dependa transitivamente de una llave candidata, esto es, que cada atributo depende directamente de una llave primaria y no a través de una relación transitiva. Por ello, una relación en 3NF es aquella que sus atributos no-llave son mutuamente independientes (Sharma, y otros, 2010).

## 2.9 Sublime Text 3 Editor

Sublime Text es un editor de código multiplataforma ligero, creado para programar sin distracciones, donde su interfaz de color oscuro y la riqueza de coloreado de la sintaxis, centra la atención completamente.

Se pueden tener varias pestañas abiertas, e incluso emplear varios paneles para utilizar más de un monitor. Dispone de modo de pantalla completa, para aprovechar al máximo el espacio visual disponible de la pantalla y para navegar por el código cuenta con un *Minimap*, un panel que permite moverse por el código de forma rápida, en la figura 2.5 se muestra un ejemplo de la interfaz principal del editor Sublime Text 3 (Skinner, 2008) .

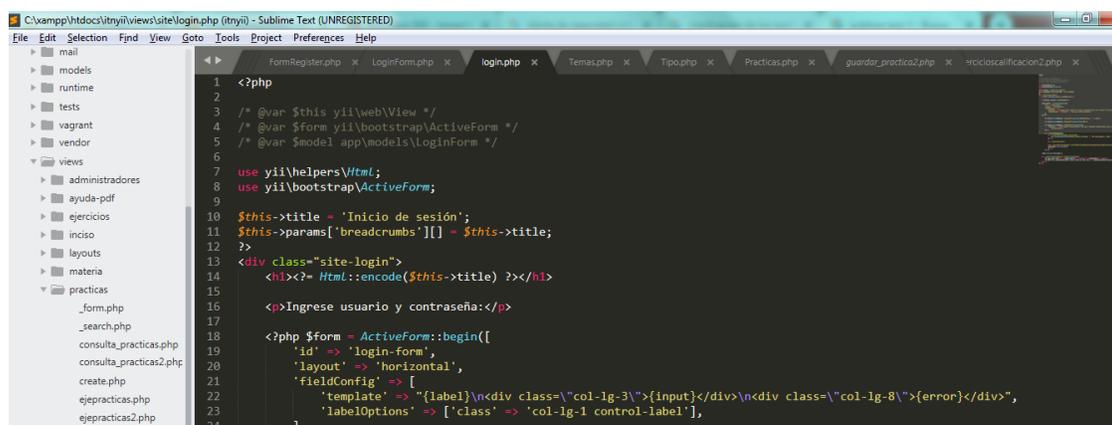


Figura 2.5 Interfaz Sublime Text 3 Editor.

Dispone de auto-guardado, opciones de personalización, cuenta con un buen número de herramientas para la edición del código y automatización de tareas. Soporta macros, *Snippets* y auto completar, entre otras funcionalidades. Algunas de sus características son ampliables mediante plugins.

Sublime Text es gratuito en su versión de prueba, plenamente funcional y sin limitación de tiempo o si se requiere la licencia individual cuesta 59 dólares, la aplicación está disponible para OS X, Linux y Windows (Skinner, 2008).

## **2.10 HTML**

Es el lenguaje más utilizado en páginas web y es comprendido por todos los navegadores web que existen hoy en día. Los documentos HTML son simples archivos de texto que incluyen marcas, texto y datos adicionales que influyen en el texto. Actualmente la versión más reciente de HTML es HTML5, aunque no todos los navegadores son completamente compatibles (Puncey & York, 2011).

### **2.10.1 Hojas de Estilo (CSS)**

CSS por su acrónimo en inglés (Cascade Style Sheets), es un lenguaje designado para describir la apariencia de documentos escritos en un lenguaje de marcas como HTML, se puede controlar el color del texto, los tipos de fuentes, el espacio entre párrafos, cómo las columnas son dimensionadas, qué imágenes o colores de fondo son utilizados, así como una gran variedad de efectos visuales.

Uno de los mayores beneficios de CSS es que una simple Hoja de Estilo puede ser utilizada para más de una sola página, es decir, que el estilo de un sitio web completo puede ser ajustado sin tener que cambiar cada página individualmente (Puncey & York, 2011).

## 2.11 Yii2 Codeception

Es un *framework* especializado en pruebas para desarrollos web, donde mediante sus pruebas unitarias, funcionales y de aceptación, ayudan a que los sistemas sean más sólidos y con menos errores.

### 2.11.1 Tests

Una parte importante del desarrollo de software, son las pruebas. Por lo que se realizan pruebas continuamente, por ejemplo, cuando se escribe una clase en PHP, se puede depurar paso a paso o simplemente usar declaraciones *echo* o *die* para verificar que la implementación funciona conforme al plan inicial. En el caso de una aplicación web, se introducen algunos datos de prueba en los formularios para asegurar de que la página interactúa con el usuario como se esperaba.

Todo el proceso de pruebas se puede realizar de manera automático, solo se invoca el código que verifica que el resultado coincide con lo que se ha planeado y se le conoce como *test* y el proceso de su creación y posterior ejecución es conocido como prueba automatizada (Khilkov, 2001).

### 2.11.2 Desarrollo con tests

El Desarrollo Dirigido por Pruebas (*Test-Driven Development* o TDD) y el Desarrollo Dirigido por Comportamientos (*Behavior-Driven Development* o BDD) son enfoques para desarrollar software, en los que se describe el comportamiento de un trozo de código o de toda la funcionalidad como un conjunto de escenarios o pruebas antes de escribir el código real y sólo entonces crear la implementación que permite pasar esos *tests* verificando que se ha logrado el comportamiento pretendido.

El proceso de desarrollo de una funcionalidad es el siguiente:

- Crear un nuevo test que describe una funcionalidad a implementar.

- Ejecutar el nuevo *test* y asegurarse de qué falla.
- Escribir un código sencillo para superar el nuevo *test*.
- Ejecutar todos los *tests* y asegurarse de que se pasan todos.
- Mejorar el código y asegurarse de que los *tests* siguen superándose.

Una vez hecho, se repite el proceso de nuevo para otra funcionalidad o mejora. Si se va a cambiar la funcionalidad existente, también hay que cambiar los *tests*.

La razón para crear los *tests* antes de hacer ninguna implementación es que eso permite centrarse en lo que se quiere alcanzar y sumergirse totalmente en «cómo hacerlo» después. Normalmente conduce a mejores abstracciones y a un más fácil mantenimiento de los *tests* cuando toque hacer ajustes a las funcionalidades o componentes menos acoplados.

Para resumir, las ventajas de este enfoque son las siguientes:

- Le mantiene centrado en una sola cosa en cada momento, lo que resulta en una mejor planificación e implementación.
- Resulta en más funcionalidades cubiertas por *tests* y en mayor detalle. Es decir, si se superan los *tests*, lo más probable es que no haya nada roto.

A largo plazo normalmente tiene como efecto un buen ahorro de tiempo (Khilkov, 2001).

### **2.11.3 Cómo probar y qué probar.**

Cuando el proyecto es simple en ocasiones no es necesario desarrollar los *tests* descritos anteriormente.

A continuación se menciona cuándo es apropiado:

- El proyecto ya es grande y complejo.
- Los requisitos del proyecto están empezando a hacerse complejos.  
El proyecto crece constantemente.
- El proyecto pretende ser a largo plazo.
- El coste de fallar es demasiado alto.

No hay nada malo en crear *tests* que cubran el comportamiento de una implementación existente.

- Es un proyecto llegado que se va a renovar gradualmente.
- Le han dado un proyecto sobre el que trabajar y no tiene *tests*.

En algunos casos cualquier forma de prueba automatizada sería exagerada:

- El proyecto es sencillo y no se va a volver más complejo.
- Es un proyecto puntual en el que no se seguirá trabajando.

Si en su proyecto se dispone de tiempo, lo mejor es automatizar las pruebas en esos casos (Khilkov, 2001).

#### **2.11.4 Preparación del entorno de pruebas**

Yii2 utiliza el *framework* de pruebas *Codeception*, que le permite crear los siguientes tipos de *tests*.

Unitario. Verifica que una unidad simple de código funciona como se espera.

Funcional. Verifica escenarios cuando el usuario utiliza un navegador para su emulación.

De aceptación. Verifica escenarios desde la perspectiva de un usuario en un navegador.

### 2.11.5 Modelo CUEVA

Una evaluación de usabilidad es el modelo de calificación de la usabilidad de entornos virtuales de aprendizaje (CUEVA), el cual da a conocer el grado de usabilidad (Alarcón-Aldana, 2014), en la tabla 2.1 se muestran los factores que se evaluaron.

Factor	Abreviatura	Ponderación
Comunicación y Comprensibilidad	CC	12%
Facilidad Estética	CFE	13%
Operatividad	CO	13.5%
Facilidad de uso	CFU	15%
Aprendizaje- Factores Humanos	CA	12.5%
Facilidad de Comprensión	CFC	12.5%
Entrenamiento	CE	11.5%
Documentación	CD	10.5%

**Tabla 2.1 Factores del modelo CUEVA.**

El modelo CUEVA se compone de 8 factores, se describen a continuación:

- Factores Estéticos (CFE). Mide el entorno artístico de la interfaz.
- Comunicación (CC). Mide el grado de interacción entre el software y el usuario.
- Operatividad (CO). Mide la capacidad del sitio para cumplir las funciones establecidas por el usuario.
- Facilidad de Uso (CFU). Mide el grado de satisfacción del usuario frente al producto.

- Aprendizaje- Factores Humanos (CA). Mide el grado de asimilación de conocimientos de los usuarios al interactuar con el sistema.
- Facilidad de Comprensión (CFC). Mide la capacidad que ofrece el sitio para usar sus recursos en tareas y condiciones de uso individual.
- Entrenamiento (CE). Mide la capacidad del sitio para obtener conocimientos, desarrollar habilidades y capacidades al interactuar.
- Documentación (CD). Mide la incidencia de los documentos que ofrece el sitio para la interacción (Alarcón-Aldana, 2014).

Cada uno de los factores cuenta con su respectivo juicio de valor, donde basado en la percepción del usuario, se otorga su calificación correspondiente, como se muestra en la tabla 2.2 .

Criterio de Juicio de Valor	Valor
Se cumple plenamente	5
Se cumple en alto grado	4 a 4.9
Se cumple aceptablemente	3 a 3.9
Se cumple insatisfactoriamente	1 a 2.9
No se cumple	0.0 a .9

**Tabla 2.2 Criterio de Juicio de Valor para modelo CUEVA.**

---

## CAPÍTULO 3. ANÁLISIS

En este capítulo se abordarán los primeros cuatro conceptos que forman parte del método utilizado, como lo son los requerimientos funcionales del sistema, así como los casos de uso y sus diagramas, todo esto para conocer el modelo conceptual de la aplicación y su arquitectura.

### 3.1 Método utilizado en la investigación

El método empleado para el desarrollo de este proyecto fue el siguiente:

1. Definir los requerimientos funcionales del sitio web.
2. Obtener los diagramas de casos de uso, casos de uso de alto nivel, diagramas de secuencia y modelo conceptual.
3. Adentrarse en los detalles de la arquitectura MVC.
4. Realizar una interfaz en la cual se basará todo el sitio web.
5. Desarrollar la base de datos con todas las tablas, relaciones y normalizar a la mayor forma normal posible con el objetivo de eliminar la redundancia de datos utilizando el manejador de bases de datos MYSQL.
6. Elaborar la aplicación web base, programar en el lenguaje web PHP, los servicios se elaboraron con el editor Sublime Text 3, se agregaron métodos y funciones necesarias para el registro de materias, unidades, temas y ejercicios, de tal manera que se adecúe a la base de datos, lo anterior haciendo uso de los procedimientos respectivos en cada capa de la Arquitectura MVC.
7. Realizar pruebas de usabilidad cargando la base de datos desde la aplicación para ver su comportamiento e identificar posibles errores, este paso se repite hasta tener un sitio consistente y sin errores.
8. Cargar el sitio de manera local y en la red “Depi” de la escuela para colocar el proyecto en la web.

9. Publicar el proyecto en el sitio web creado y volver a hacer pruebas de la aplicación para evaluar los resultados y comportamiento de la aplicación, encontrar posibles errores y corregirlos.
10. Obtener resultados y emitir conclusiones.

### 3.2 Requerimientos Funcionales

A continuación se definen cuáles son los requerimientos funcionales del sistema, es decir, aquellas tareas que son fundamentales para el funcionamiento del sitio, según las necesidades del usuario final y se definió el tipo de categoría a la que pertenecen.

Los requerimientos fueron separados por módulos, primeramente tenemos el requerimiento del módulo Registro mostrado en la Tabla 3.1.

**Tabla 3.1 Requerimiento funcional del módulo de Registro.**

Referencia	Función	Categoría
<b>REG01</b>	El sistema permite al usuario registrarse en el sitio y crear una cuenta.	Evidente

Los requerimientos del módulo Control de Sesión se muestran en la Tabla 3.2.

**Tabla 3.2 Requerimientos funcionales del módulo de Control de Sesión.**

Referencia	Función	Categoría
<b>LOG01</b>	El sistema permite identificarse en el sitio para poder acceder a las opciones disponibles.	Evidente
<b>LOG02-</b>	El sistema permite al usuario con sesión iniciada cerrar la misma.	Evidente

Los siguientes dos módulos son de las opciones que tiene cada uno de los tipos de usuario que se tendrán disponibles en el sitio. En la Tabla 3.3 se tienen los requerimientos del módulo “Opciones del Administrador”, después en la Tabla 3.4 están los requerimientos del módulo “Opciones del Alumno”.

**Tabla 3.3 Requerimientos funcionales del módulo de Opciones del Administrador.**

Referencia	Función	Categoría
ADM01	El sistema permite a los usuarios con privilegios de Administrador dar de alta materias.	Evidente
ADM02	El sistema permite a los usuarios con privilegios de Administrador registrar unidades de las materias disponibles.	Evidente
ADM03	El sistema permite a los usuarios con privilegios de Administrador registrar temas en las unidades de las materias disponibles.	Evidente
ADM04	El sistema permite a los usuarios con privilegios de Administrador agregar ejercicios a los temas disponibles.	Evidente

**Tabla 3.4 Requerimientos funcionales del módulo de Opciones del Alumno.**

Referencia	Función	Categoría
ALM01	El sistema permite a los usuarios con privilegios de Alumno registrarse en el sistema.	Evidente
ALM02	El sistema permite a los usuarios con privilegios de Alumno resolver ejercicios seleccionados en cada práctica.	Evidente
ALM03	El sistema permite a los usuarios con privilegios de Alumno ver las calificaciones obtenidas en las prácticas realizadas.	Evidente

### 3.4 Perfil de los actores

A continuación se menciona el tipo de usuario que podrá utilizar la plataforma desarrollada.

#### 3.4.1 Administrador

Es el encargado de dar de alta las materias, unidades, temas y ejercicios en los que se basarán las prácticas que posteriormente podrán utilizarse, así mismo, un docente también puede fungir como administrador.

#### 3.4.2 Alumno

Este tipo de usuarios se registran en el sistema y se encargan de realizar las prácticas respectivas con cada grupo de ejercicios.

### 3.5 Casos de uso de alto nivel

Los casos de uso son documentos narrativos que describen la secuencia de eventos de un actor que utiliza un sistema para completar un proceso, esta definición se maneja en el libro UML y patrones (Larman, 2004).

A continuación se muestran los casos de uso de alto nivel para cada uno de los requerimientos funcionales de los distintos módulos que se presentaron en el punto 3.3.

#### 3.5.1 Casos de uso de alto nivel para el módulo “Registro”

La Tabla 3.5 muestra el caso de uso de alto nivel para el registro en el sitio.

**Tabla 3.5 Caso de uso de alto nivel para Registro en Sitio.**

<b>Caso de Uso:</b>	Registro en Sitio
<b>Actores:</b>	Usuario, sistema.
<b>Tipo:</b>	Primario.

<b>Descripción:</b>	Un usuario entra a la pantalla de registro, llena los datos requeridos por el sistema incluyendo su correo electrónico, da clic en el botón registrar, si esta todo correcto, el sistema registra al usuario.
---------------------	---

### 3.5.2 Casos de uso de alto nivel para el módulo “Control de Sesión”

La Tabla 3.6 muestra el caso de uso de alto nivel para el inicio de sesión.

**Tabla 3.6 Caso de uso de alto nivel para Inicio de sesión.**

<b>Caso de Uso:</b>	Inicio de Sesión
<b>Actores:</b>	Usuario, sistema.
<b>Tipo:</b>	Primario.
<b>Descripción:</b>	Un usuario entra a la pantalla de Inicio de Sesión, ingresa su nombre de usuario y contraseña, el sistema se encarga de realizar la autenticación y verifica el rol de usuario al que pertenece para redirigir a la pantalla de inicio correspondiente y mostrará el nombre de usuario en la esquina superior derecha.

La tabla 3.7 muestra el caso de uso de alto nivel para el cierre de sesión.

**Tabla 3.7 Caso de uso de alto nivel para Cierre de Sesión.**

<b>Caso de Uso:</b>	Cierre de Sesión
<b>Actores:</b>	Usuario registrado, sistema.
<b>Tipo:</b>	Primario.
<b>Descripción:</b>	En cualquier pantalla debajo del nombre del usuario se podrá seleccionar la opción “Cerrar sesión”, una vez hecho esto el sistema cierra la sesión y dirige a pantalla de inicio de sesión.

### 3.5.3 Casos de uso de alto nivel para el módulo “Opciones del Administrador”

La Tabla 3.8 muestra el caso de uso de alto nivel para crear una Materia.

**Tabla 3.8 Caso de uso de alto nivel para Crear una Materia.**

<b>Caso de Uso:</b>	Crear una Materia
<b>Actores:</b>	Administrador, sistema.
<b>Tipo:</b>	Primario.
<b>Descripción:</b>	Un administrador ingresa a la pantalla para registro de Materia, ingresa campos requeridos y da clic en botón crear, el sistema informa si se logró el registro de la materia y hace los cambios necesarios en la Base de Datos.

La Tabla 3.9 muestra el caso de uso de alto nivel para agregar Unidades.

**Tabla 3.9 Caso de uso de alto nivel para Crear Unidades.**

<b>Caso de Uso:</b>	Crear Unidades
<b>Actores:</b>	Administrador, sistema.
<b>Tipo:</b>	Primario.
<b>Descripción:</b>	Un administrador ingresa a la pantalla para registro de unidades, selecciona la materia a generarle unidades, ingresa los campos necesarios y al finalizar el sistema confirma que se ha creado la unidad mostrándolo en la lista de unidades de la materia seleccionada y hace los cambios necesarios en la Base de Datos.

La Tabla 3.10 muestra el caso de uso de alto nivel para agregar temas.

**Tabla 3.10 Caso de uso de alto nivel para Crear Temas.**

<b>Caso de Uso:</b>	Crear Temas
<b>Actores:</b>	Administrador, sistema.
<b>Tipo:</b>	Primario.
<b>Descripción:</b>	Un administrador ingresa a la pantalla para registro de temas, selecciona una materia y unidad a generarle temas, ingresa los campos necesarios y al finalizar el sistema confirma que se ha creado el tema mostrándolo en la lista de temas y hace los cambios necesarios en la Base de Datos.

La Tabla 3.11 muestra el caso de uso de alto nivel para agregar ejercicios.

**Tabla 3.11 Caso de uso de alto nivel para Crear Ejercicios.**

<b>Caso de Uso:</b>	Crear Ejercicios
<b>Actores:</b>	Administrador, sistema.
<b>Tipo:</b>	Primario.
<b>Descripción:</b>	Un administrador ingresa a la pantalla para creación de ejercicios, elige la materia, unidad y tema al que se le agregará el ejercicio e ingresa toda la información necesaria, el sistema confirma el resultado, la creación del nuevo ejercicio y hace los cambios necesarios en la Base de Datos.

### 3.5.4 Casos de uso de alto nivel para el módulo “Opciones del Alumno”

La Tabla 3.12 muestra el caso de uso de alto nivel para resolver prácticas.

**Tabla 3.12 Caso de uso de alto nivel para Realizar Prácticas.**

<b>Caso de Uso:</b>	Resolver Prácticas
<b>Actores:</b>	Alumno, sistema.
<b>Tipo:</b>	Primario.
<b>Descripción:</b>	Un alumno ingresa a la página de Ejercicios, selecciona la materia, unidad, tema, subtema de los ejercicios a realizar, el sistema lo redirige a la página de respuestas con ayuda de tutoriales y documentos PDF, al finalizar el sistema lo redirige a la página de resultados, le muestra la calificación obtenida y le da la opción de almacenar la práctica en la Base de Datos.

La Tabla 3.13 muestra el caso de uso de alto nivel para la opción “ver mis calificaciones”.

**Tabla 3.13 Caso de uso de alto nivel para Ver mis Calificaciones.**

<b>Caso de Uso:</b>	Ver mis Calificaciones
<b>Actores:</b>	Alumno, sistema.
<b>Tipo:</b>	Primario.
<b>Descripción:</b>	Un alumno ingresa a la página de prácticas y el sistema le muestra en un Dropdownlist sus prácticas, selecciona que calificación requiere consultar, y el sistema le muestra los ejercicios realizados y su calificación.

### 3.6 Diagramas de casos de uso

En base a los requerimientos funcionales se realizaron los diagramas de caso de uso, que son representaciones gráficas de un conjunto de casos de uso, actores y la relación entre ellos. En los diagramas los casos de uso se representan con óvalos y los actores como figuras estilizadas. Las relaciones entre ellos son representadas por líneas (Larman, 2004).

La Figura 3.1 muestra el diagrama de caso de uso para el módulo Registro.



Figura 3.1 Diagrama de caso de uso "Registro".

La Figura 3.2 muestra el diagrama de caso de uso para el módulo Control de Sesión.

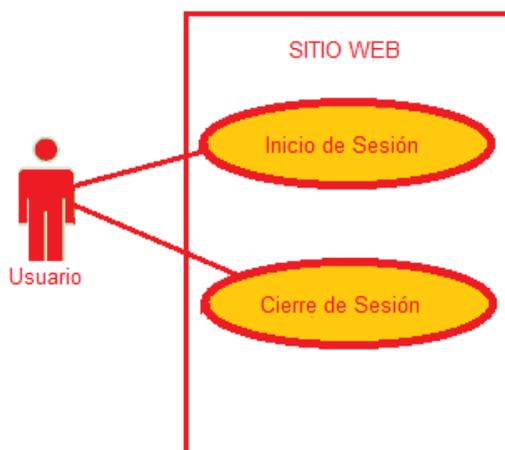
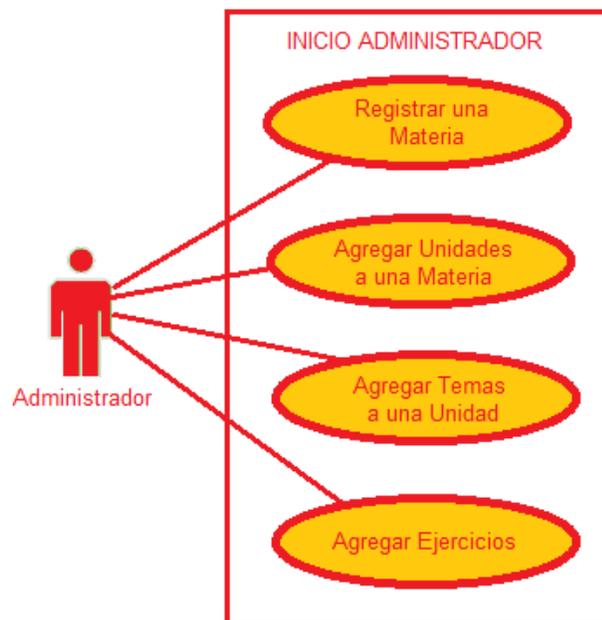


Figura 3.2 Diagrama de caso de uso "Control de Sesión".

En la Figura 3.3 se muestra el diagrama de caso de uso para el módulo Opciones del Administrador.



**Figura 3.3 Diagrama de caso de uso “Opciones del Administrador”.**

En la Figura 3.4 se muestra el diagrama de caso de uso para el módulo Opciones del Alumno.



**Figura 3.4 Diagrama de caso de uso “Opciones del Alumno”.**

### 3.7 Diagramas de secuencia

Una vez elaborados los casos de uso, lo siguiente es realizar los diagramas de secuencia de cada uno de ellos, siguiendo el orden de los artefactos a utilizar.

Los diagramas de secuencia son utilizados para representar las interacciones y eventos de los actores externos con el sistema, estos son representados en base al escenario de un caso de uso en específico y dentro del dominio del curso normal de los eventos (Larman, 2004).

El siguiente diagrama de secuencia mostrado en la Figura 3.5 es del caso de uso Registro en Sitio.

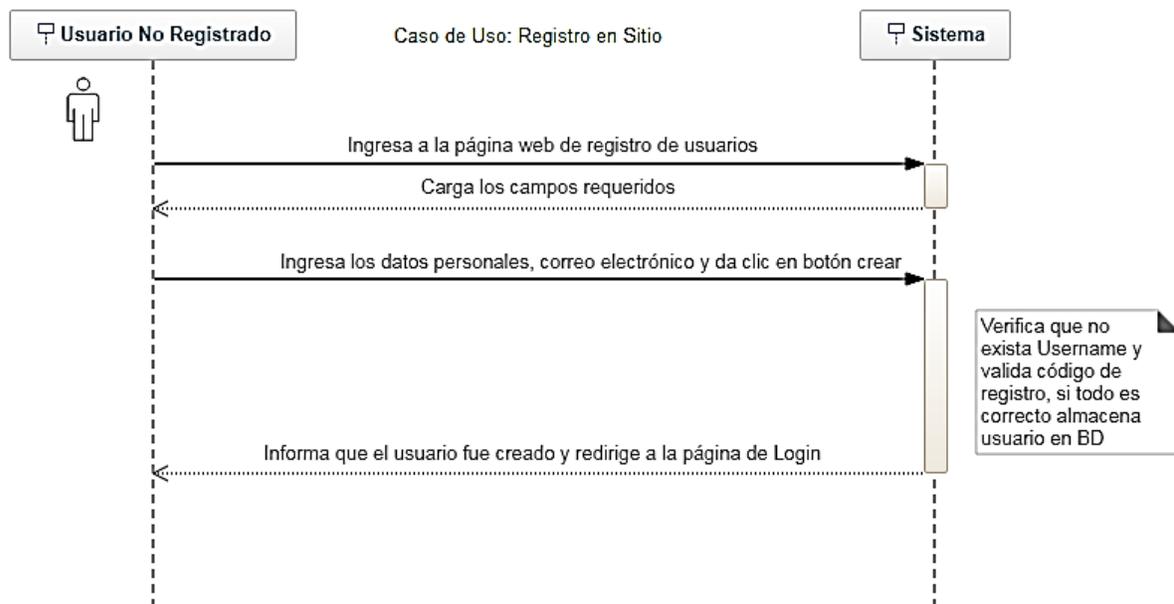


Figura 3.5 Diagrama de secuencia "Registro en Sitio".

En las figuras Figura 3.6 y Figura 3.7 se muestran los diagramas de secuencia del módulo Control de sesión, siendo la primera para el caso de uso Inicio de Sesión y la segunda para el caso de uso Cierre de Sesión.

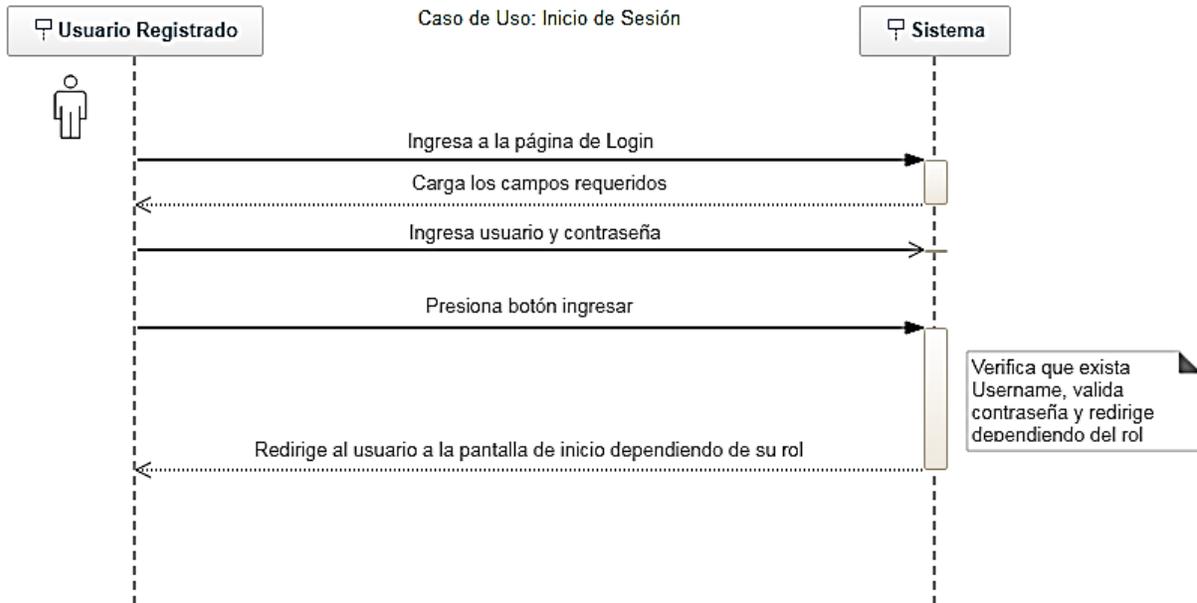


Figura 3.6 Diagrama de secuencia "Inicio de Sesión".

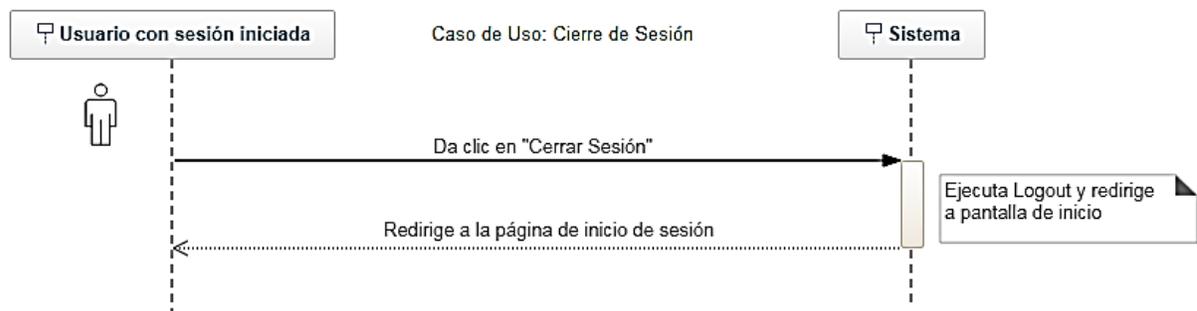


Figura 3.7 Diagrama de secuencia "Cierre de Sesión".

En las figuras Figura 3.8, Figura 3.9, Figura 3.10 y Figura 3.11 se muestran los diagramas de secuencia correspondientes al módulo Opciones del Administrador.

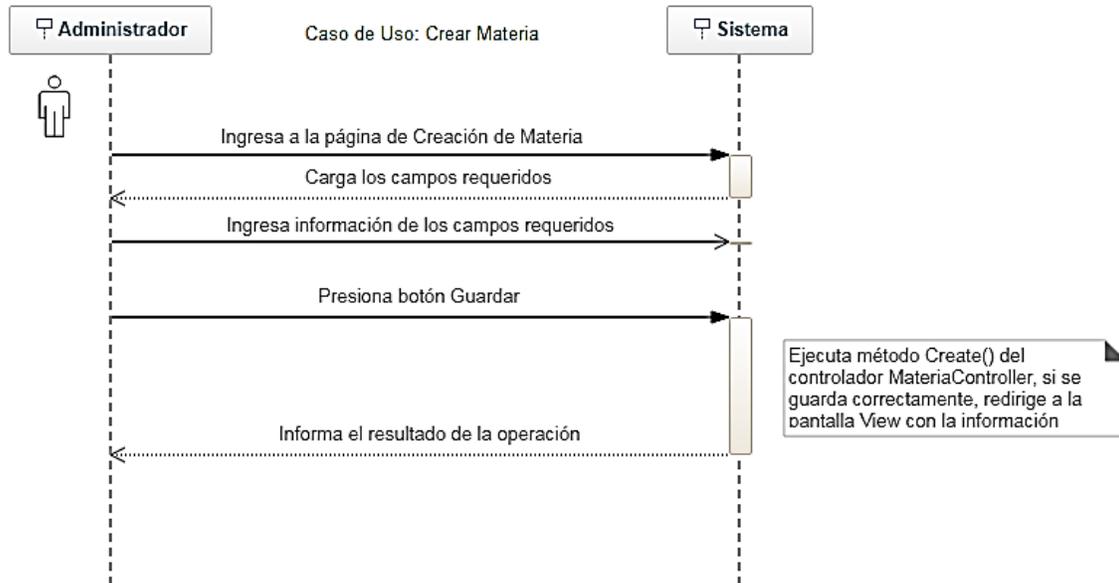


Figura 3.8 Diagrama de secuencia "Crear Materia".

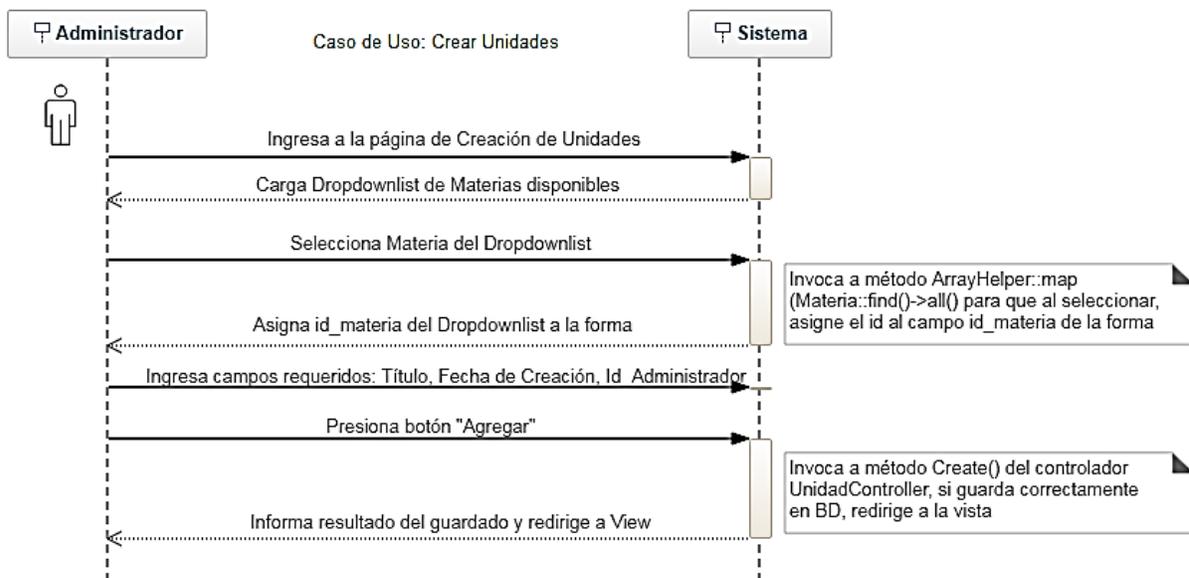


Figura 3.9 Diagrama de secuencia "Crear Unidades".

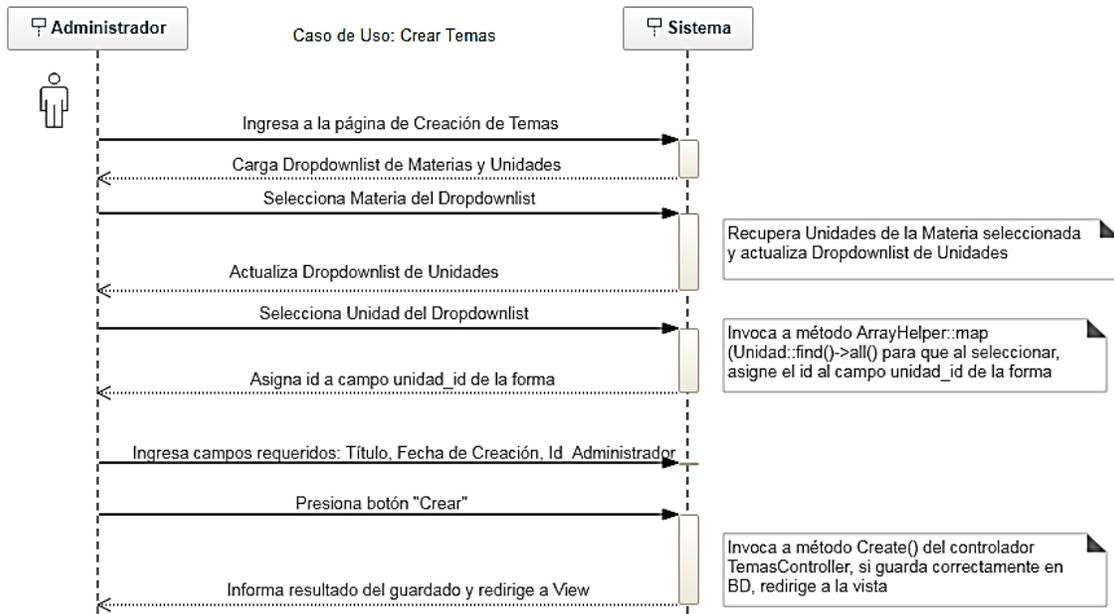


Figura 3.10 Diagrama de secuencia "Crear Temas".



Figura 3.11 Diagrama de secuencia "Crear Ejercicios".

Por último se tienen las figuras Figura 3.12 y Figura 3.13 donde se muestran los diagramas de secuencia correspondientes para el módulo Opciones del Alumno.



Figura 3.12 Diagrama de secuencia "Resolver Prácticas".

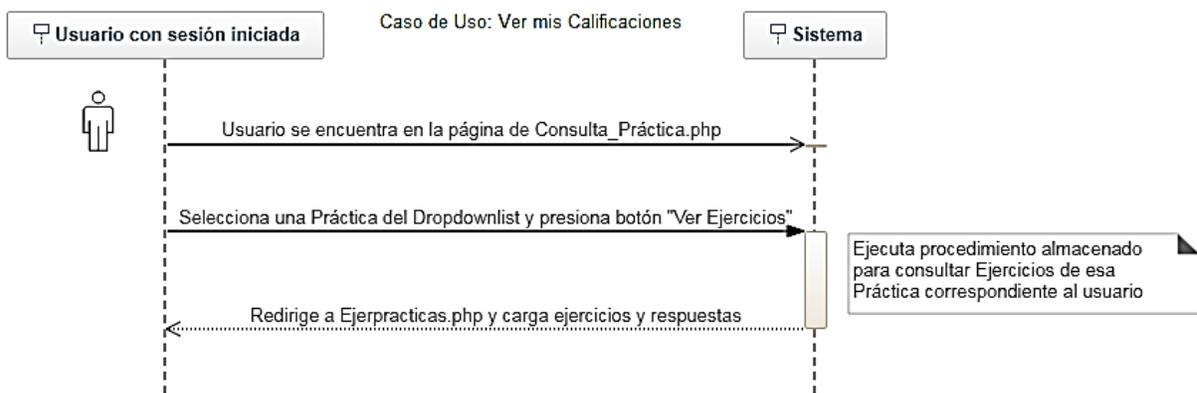


Figura 3.13 Diagrama de secuencia "Ver mis Calificaciones".

### 3.8 Modelo Conceptual de la aplicación

Después de realizar los casos de uso y desarrollar los diagramas de secuencia, el artefacto más importante a crear durante la fase de análisis orientado a objetos es el mapa conceptual. En él se explica a sus creadores los conceptos con mayor importancia en un dominio del problema (Larman, 2004).

A continuación se muestra en la Figura 3.14 el mapa conceptual del prototipo de esta tesis.

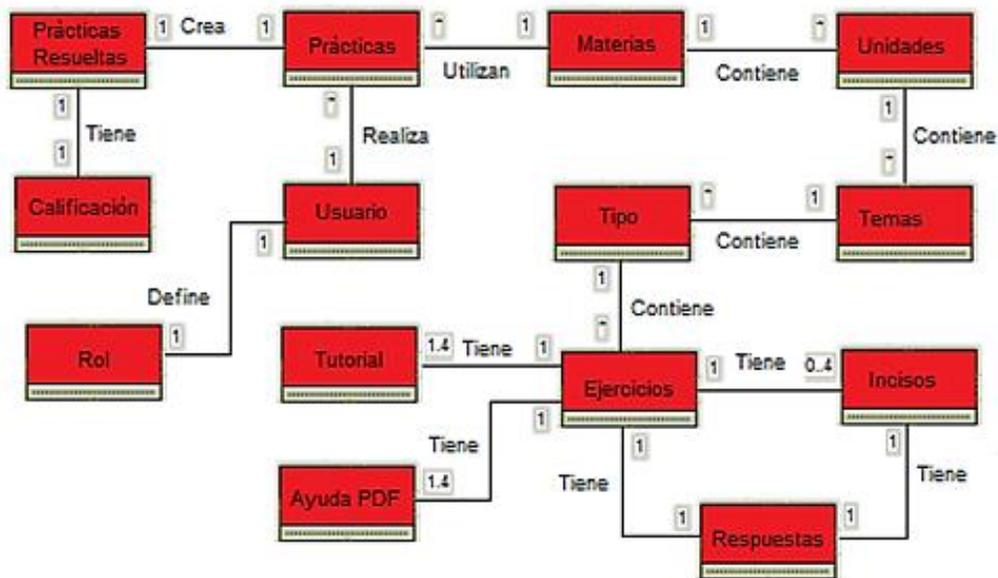


Figura 3.14 Mapa Conceptual de la Aplicación.

A pesar de que el modelo conceptual es una abstracción de un concepto del mundo real y no una definición del software es una herramienta que nos lleva a darnos una idea de cómo es el diagrama de clases, incluso el diseño de la base de datos, de aquí la gran importancia que este artefacto tiene.

### 3.9 Arquitectura de la aplicación

Después de conocer el mapa conceptual de la aplicación, siguiendo con la secuencia de los artefactos a realizar, necesitamos diseñar su respectiva arquitectura.

La selección de la arquitectura del software tiene el efecto más crucial en la implementación del sistema y es casi imposible cambiarla en etapas avanzadas del diseño (Qin, Xing, & Zheng, 2008).

La arquitectura es una buena manera de identificar a nivel muy general el comportamiento de la aplicación como estructuras y mecanismos de comunicación, sin entrar en detalles, como lenguaje de programación a utilizar, gestor de base de datos, etc. En proyectos muy grandes ayuda a organizar a los miembros del equipo con responsabilidades y tareas específicas.

En la figura 3.15 se muestra la descripción de la arquitectura a utilizar en el prototipo de esta tesis.

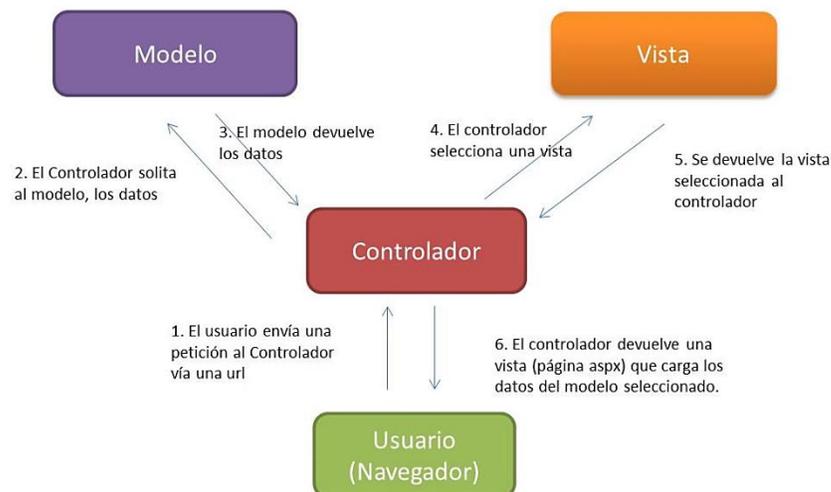


Figura 3.15 Arquitectura del patrón MVC de la Aplicación.

---

## CAPÍTULO 4. DISEÑO DE LA APLICACIÓN

En este capítulo se explican los puntos cinco y seis del método utilizado, como lo es el funcionamiento del *framework* Yii2 y como fue el desarrollo de la interfaz de usuario con apoyo de la herramienta *Yii2 code generator*, así como el diseño de la base de datos.

### 4.1 Introducción

Para la elaboración del prototipo de esta tesis se partió del hecho de que se realizaría una aplicación que sirva de apoyo para gestionar los cursos de cálculo diferencial del sistema probado en el ITN y se pueda generalizar al TecNM.

Por tratarse de una aplicación que gestiona las prácticas de cada persona por separado es un hecho que se necesitan controles de acceso, es decir usuarios y sus respectivas contraseñas, además de gestionar los roles de acceso o privilegios de cada usuario (alumno o administrador).

Los usuarios se registran a la página utilizando su correo electrónico y creando un usuario y contraseña, el cual, tendrá que ser verificado en su correo mediante un link para validar su registro, con el cual, se identificarán para iniciar sesión en el sitio web.

Los usuarios con privilegios de administrador pueden dar de alta materias, unidades, temas y subtemas, así como los ejercicios de cada uno, para que estos formen parte de las prácticas a realizar.

Los usuarios con privilegios de alumno tienen la capacidad de seleccionar los ejercicios a realizar, dependiendo de la selección previa de materia, unidad, tema y subtema, donde al comenzar la práctica, le aparecen respuestas aleatorias, así como también el orden aleatorio de los problemas a resolver.

## 4.2 Diseño de la interfaz del usuario

El desarrollo de esta interfaz se basó en las indicaciones y recomendaciones del libro de Weinschenk, Jamar y Yeo llamado *GUI design essentials*.

En él se hablan de los aspectos a tomar en cuenta en las fases de análisis, diseño y las correspondientes evaluaciones para cada uno de ellos (Weinschenk, 1997).

Utilizando el *framework* Yii2, haciendo mención a la parte de la vista del patrón MVC, se maneja una barra de navegación donde los accesos a las opciones es mediante su menú que se encuentra en la parte superior, y en cada selección se mantiene la misma estructura, mostrando el contenido en la parte central, que varía dependiendo de la selección de opción.

Como principal objetivo para la creación de las páginas fue mantenerlas sin saturarlas de información o controles que pudieran hacer muy incómodo a la vista. Para el caso de tareas que requieren mucha información por parte del usuario se dividió el proceso en varias páginas, como es el caso de la creación de prácticas.

A continuación en las figuras Figura 4.1, Figura 4.2 y Figura 4.3 se muestran unas capturas de pantalla de distintas páginas del sitio donde se puede apreciar la plantilla que se utilizó para el prototipo de la aplicación, donde se observan características mencionadas en el punto 2.7 relacionado con las interfaces como lo son el diseño unificado, manejo de página principal como menú, mismo tipo de imágenes y colores para todo el sitio, entre otros.



Figura 4.1 Captura de pantalla de menú Inicio.

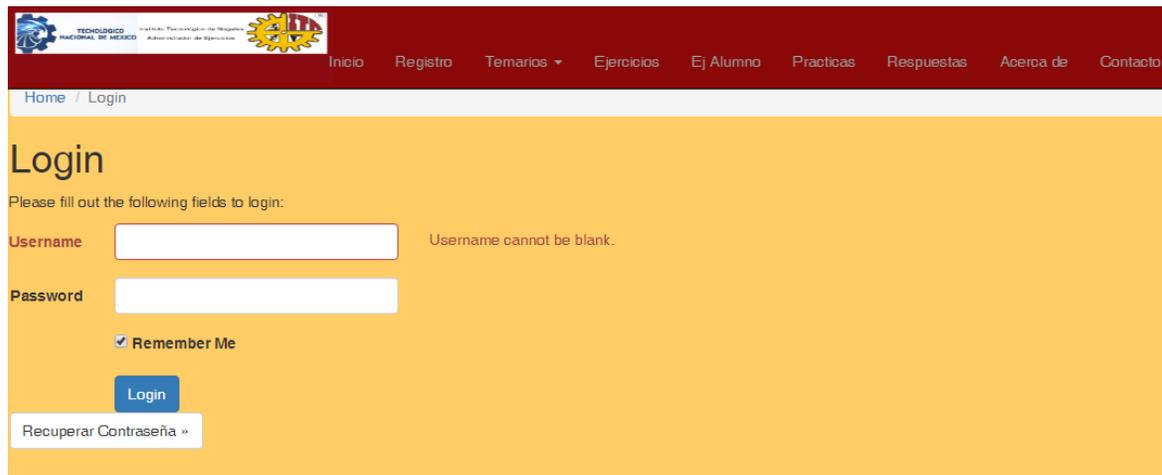


Figura 4.2 Captura de pantalla de Inicio de Sesión

Esta pantalla corresponde al caso de uso de inicio de sesión relacionado con la tabla 3.6 y con el diagrama de secuencia de la figura 3.6.



**Figura 4.3 Captura de Pantalla de selección de Ejercicios.**

Esta pantalla corresponde al caso de uso de realizar prácticas, relacionado con la tabla 3.12 y con el diagrama de secuencia de la figura 3.12.

### 4.3 Diseño de la base de Datos

Para la creación de las tablas de la base de datos de la aplicación, se utilizó como referencia el mapa conceptual mostrado en la Figura 3.14, ya que en él se tienen las relaciones entre conceptos, así como su multiplicidad, de esa manera se fueron creando una a una las tablas con campos necesarios, intentando utilizar la mayor forma normal posible, que en este proyecto fue la 3fn, donde utilizando llaves foráneas en las tablas relacionadas, evitamos redundancia de datos.

La 3fn se utilizó en las tablas correspondientes al temario, como lo son materia, unidad, temas y tipo, donde en cada nivel se va agregando la llave foránea para ir enlazando a que materia pertenece la unidad, a que unidad pertenece el tema y a que tema pertenece el subtema.

Así mismo, también se aplicó en la tabla ejercicios, donde se agregaron las llaves foráneas del subtema, usuario y respuesta para identificarlo de los demás.

Como resultado se obtuvo una estructura de base de datos que se muestra a continuación, separada por secciones para poder apreciarla mejor, primeramente la tabla de usuario y sus relaciones directas se muestran en la Figura 4.4.

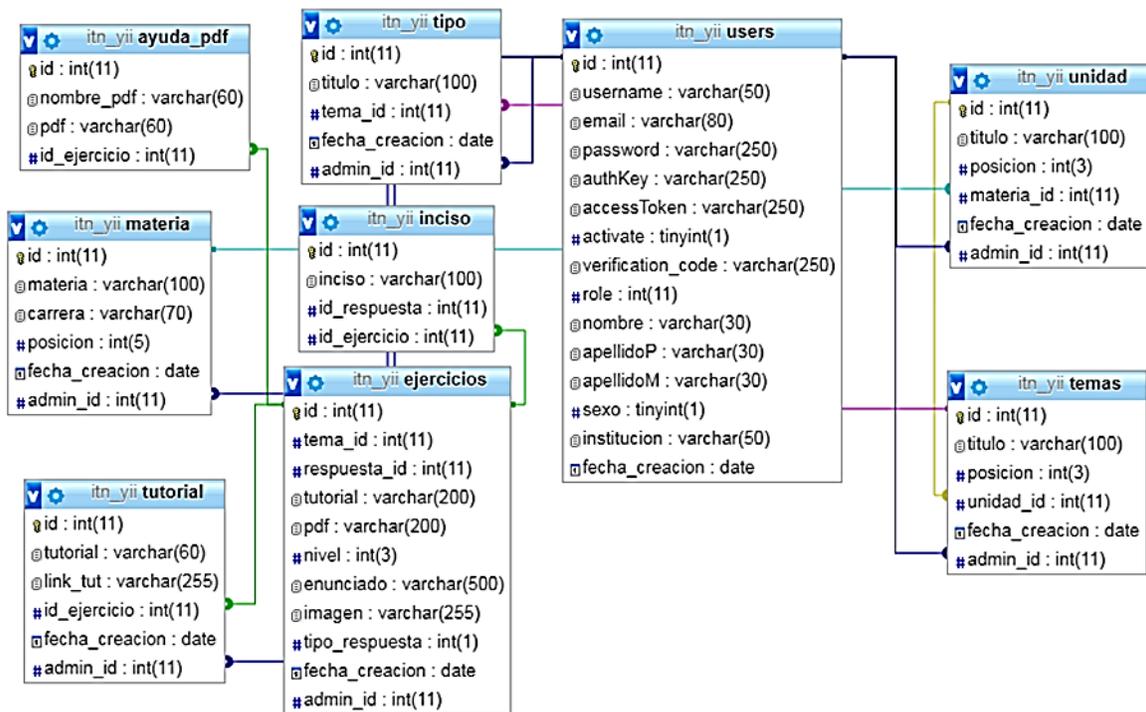


Figura 4.4 Tablas con relación directa con Users.

En la figura 4.4 se muestra como la tabla Users interactúa con la mayoría de las otras tablas, ya que al momento de dar de alta materias, unidades, temas, subtemas, ejercicios y sus tablas derivadas, es necesario se agregue el registro de que usuario fue el que realizó la operación.

En la figura 4.5, se muestra como interactúa la tabla Ejercicios con las demás, ya que esta es la base para el desarrollo de las prácticas que realiza el alumno, por lo que su diseño involucra las tablas de usuario, subtemas, tutorial, ayuda PDF, respuestas e inciso.

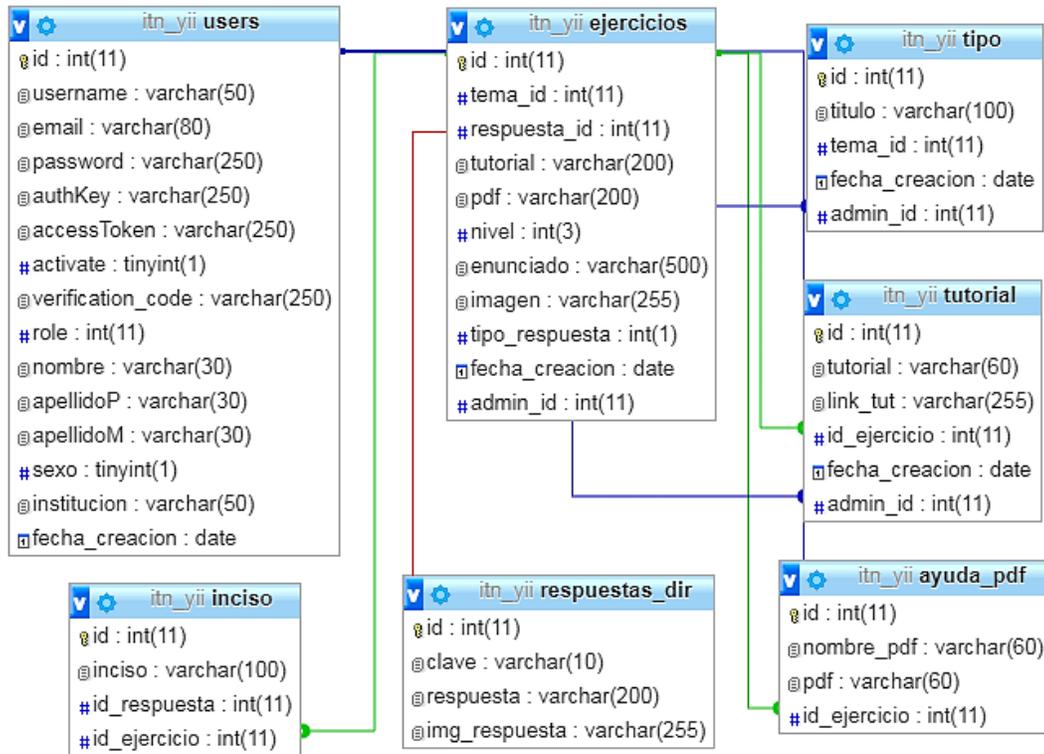


Figura 4.5 Tablas con relación directa con Ejercicios.

En la figura 4.5 se puede ver como la tabla Ejercicios se compone de las principales llaves foráneas de tema\_id, para conocer a que tema pertenece el ejercicio, respuesta\_id para enlazarlo con su respuesta correspondiente y admin\_id para saber que Administrador fue el creador.

En la figura 4.6 se muestra como la tabla Prácticas almacena el nombre de la Práctica, el id del usuario que la realizó y la calificación obtenida, así también, el cómo interactúa con la tabla Prácticas resueltas, donde en ella se guarda el detalle de los ejercicios realizados en cada práctica, así como la respuesta hecha en cada uno de los ejercicios.

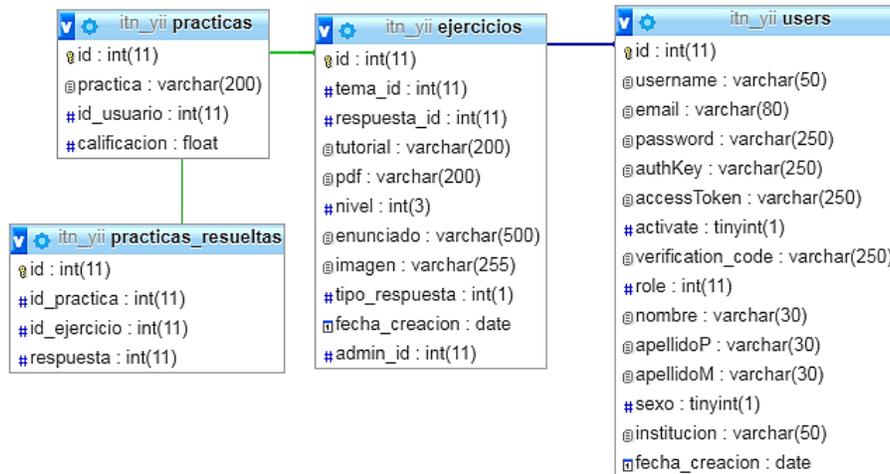


Figura 4.6 Tablas con relación directa con Prácticas.

## 4.4 Utilización de la herramienta Yii code Generator

### 4.4.1 Model Generator

Una de las herramientas principales del Framework Yii2 es GII, donde una vez realizada de manera correcta la base de datos, se comienza a desarrollar el sitio con el generador de código, haciendo más sólido el entorno con opciones en común y enfocarse en las opciones que ocupan desarrollo personalizado.

Es por ello que se comienza realizando el “*Model*” de cada una de las tablas de la base de datos, siendo este una representación de todas las operaciones que se realizan con el CRUD, el procedimiento principal se muestra en la figura 4.7.

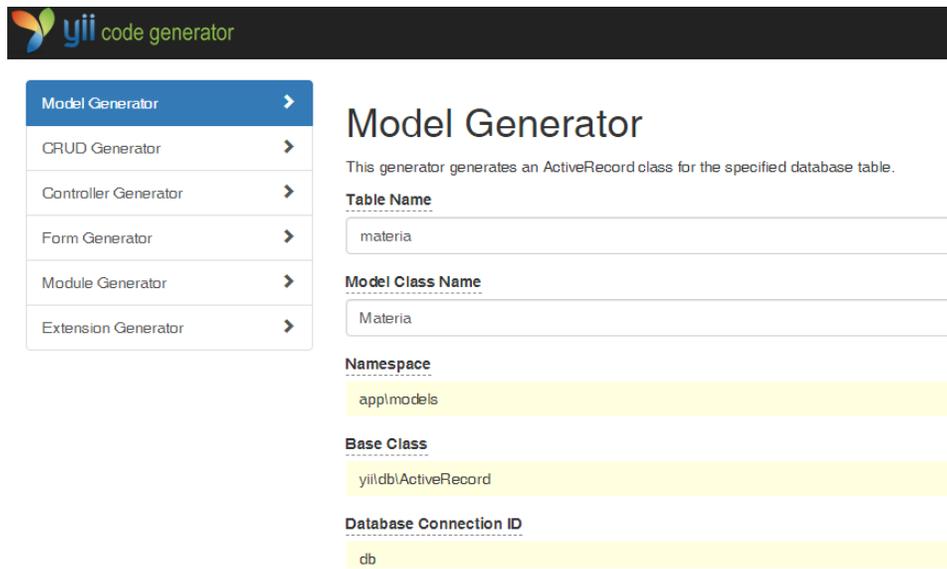


Figura 4.7 Utilización de GII para creación de Modelos.

Una vez creado el modelo, se puede ver su código, en él se definen las reglas que deben coincidir con las establecidas en la tabla, así como las etiquetas que se utilizarán en la interfaz, como se muestra en la figura 4.8.

```

public function rules()
{
    return [
        [['materia', 'carrera', 'fecha_creacion', 'admin_id'], 'required'],
        [['posicion', 'admin_id'], 'integer'],
        [['fecha_creacion'], 'safe'],
        [['materia'], 'string', 'max' => 100],
        [['carrera'], 'string', 'max' => 70],
        [['admin_id'], 'exist', 'skipOnError' => true, 'targetClass' => Users::className(), 'targetAttribute' =>
            ['admin_id' => 'id']],
    ];
}

/**
 * {@inheritdoc}
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'materia' => 'Materia',
        'carrera' => 'Carrera',
        'posicion' => 'Posición',
        'fecha_creacion' => 'Fecha de Creación',
        'admin_id' => 'Admin ID',
    ];
}

```

Figura 4.8 Código del modelo generado.

En la función *rules()* de la figura 4.8 podemos ver los campos “materia, carrera, posición, fecha\_creación, admin\_id”, que están señalados como *required*, esto significa que al momento de ingresar un registro en esta tabla no pueden contener nulos, tiene que ingresarse información, de lo contrario marcará un error y no se podrá continuar.

También se puede observar que los campos “posición y admin\_id” son de tipo *integer* (entero), los campos “materia y carrera” son de tipo *string* (cadena), con longitud máxima de caracteres de 100 y 70 respectivamente.

Al final de la función, se observa la palabra *exist* refiriéndose al campo “admin\_id”, donde señala que el campo debe de existir con relación a la tabla “Users” y su campo *Id*. En otras palabras, no podemos agregar un registro en la tabla Materia sin que el campo ingresado en “admin\_id” exista en la tabla *Users*.

Continuando con la siguiente función *attributeLabels()* se observa como los campos anteriormente mencionados, van acompañados de las etiquetas que se utilizarán al ser llamados en la interfaz de usuario, como es el caso del campo “fecha\_creacion”, que se mostrará en la interfaz como “Fecha de Creación”.

Todo esto concuerda con el diseño utilizado al momento de crear la tabla y sus relaciones con las demás en la base de datos, por eso es muy importante que el análisis al inicio sea lo más acertado posible, para evitar ajustes posteriores.

#### **4.4.2 CRUD Generator**

Otra herramienta importante después de realizar los modelos de cada una de las tablas de la base de datos, es el generador de CRUD (*Create, Read, Update, Delete*) donde se pueden generar las opciones básicas del sitio e ir las adecuando a necesidades específicas del sistema.

Para realizar el CRUD de cada una de las tablas que interactúan en las operaciones del sistema, es necesario seleccionar el modelo, nombrar el controlador y el controlador de búsqueda, como se muestra en la figura 4.9.

**Figura 4.9 Utilización de GII para creación de CRUD.**

Una vez ingresados esos datos, se presiona el botón *Preview* donde se muestran los archivos a crear, que serán 7 para la carpeta *views*, que es donde el usuario interactúa entre las opciones del CRUD, otro para el modelo de búsqueda, y otro para el controlador, que será el que administre las peticiones del usuario y las respuestas del sistema, como se muestra en la figura 4.10.

Code File	Action	
controllers\MateriaController.php	unchanged	<input type="checkbox"/>
models\MateriaSearch.php	unchanged	<input type="checkbox"/>
views\material_form.php <span>diff</span>	overwrite	<input type="checkbox"/>
views\material_search.php	unchanged	<input type="checkbox"/>
views\materialcreate.php	unchanged	<input type="checkbox"/>
views\materialindex.php <span>diff</span>	overwrite	<input type="checkbox"/>
views\materialupdate.php	unchanged	<input type="checkbox"/>
views\materialview.php	unchanged	<input type="checkbox"/>

**Figura 4.10 Pre visualización de los archivos CRUD.**

Una vez revisado el *Preview*, se presiona el botón *Generate* y ahí se crearán los archivos mencionados en el punto anterior. A continuación se menciona el funcionamiento de cada uno de los archivos.

## Form.php

Este archivo es el que organiza la presentación de los campos en la forma y el tipo de ingreso por parte del usuario, es utilizada por *create.php* (insertar registros) y *update.php* (actualizar registros), como se muestra en la figura 4.11.

```
1 <?php
2
3 use yii\helpers\Html;
4 use yii\widgets\ActiveForm;
5 use yii\helpers\ArrayHelper;
6 use app\models\Users;
7
8 /* @var $this yii\web\View */
9 /* @var $model app\models\Materia */
10 /* @var $form yii\widgets\ActiveForm */
11 ?>
12
13 <div class="materia-form">
14
15     <?php $form = ActiveForm::begin(); ?>
16
17     <?= $form->field($model, 'materia')->textInput(['maxlength' => true]) ?>
18
19     <?= $form->field($model, 'carrera')->textInput(['maxlength' => true]) ?>
20
21     <?= $form->field($model, 'posicion')->textInput() ?>
22
23     <?= $form->field($model, 'fecha_creacion')->input("date") ?>
24
25     <?= $form->field($model, 'admin_id')->dropDownList(ArrayHelper::map(Users::find()->all(), 'id', 'username'), [
26         'class' => 'form-control inline-block']) ?>
27     <div class="form-group">
28         <?= Html::submitButton('Save', ['class' => 'btn btn-success']) ?>
29     </div>
```

Figura 4.11 Código de vista form.php.

En la figura 4.11 se muestran los campos que están dentro de la forma activa *ActiveForm* y se le agrega la manera en que se capturarán los datos, por lo que “materia, carrera y posición” serán de tipo *textInput* (entrada de texto), en el caso de la fecha desplegará un objeto de tipo *Date*, donde se seleccionará la fecha de un calendario, y por último el campo “admin\_id” desplegará un *DropDownList* con los usuarios que existan dentro de la tabla *Users* utilizando la función *ArrayHelper::map()*.

## Create.php

Este archivo agrega opciones de etiquetas y reenvía el modelo a utilizar por la forma en la creación de nuevos registros, se muestra en la figura 4.12.

```
1 |<?php
2
3 | use yii\helpers\Html;
4
5
6 | /* @var $this yii\web\View */
7 | /* @var $model app\models\Materia */
8
9 | $this->title = 'Create Materia';
10 | $this->params['breadcrumbs'][] = ['label' => 'Materias', 'url' => ['index']];
11 | $this->params['breadcrumbs'][] = $this->title;
12 | ?>
13 | <div class="materia-create">
14 |
15 |     <h1><?= Html::encode($this->title) ?></h1>
16 |
17 |     <?= $this->render('_form', [
18 |         'model' => $model,
19 |     ]) ?>
20 |
21 | </div>
```

Figura 4.12 Código de vista create.php.

La figura 4.12 muestra las etiquetas que pertenecen a la opción crear nuevo registro, y cuando es llamada, redirige a la página *Index* enviando la forma con los campos definidos en *form.php* y la variable tipo *model* que será donde se guardará la información.

## Update.php

Este archivo agrega opciones de etiquetas y reenvía el modelo a utilizar por la forma en la actualización de registros existentes, se muestra en la figura 4.13.

```

1 |<?php
2
3 | use yii\helpers\Html;
4
5 | /* @var $this yii\web\View */
6 | /* @var $model app\models\Materia */
7
8 | $this->title = 'Update Materia: ' . $model->id;
9 | $this->params['breadcrumbs'][] = ['label' => 'Materias', 'url' => ['index']];
10 | $this->params['breadcrumbs'][] = ['label' => $model->id, 'url' => ['view', 'id' => $model->id]];
11 | $this->params['breadcrumbs'][] = 'Update';
12 | ?>
13 | <div class="materia-update">
14
15 |     <h1><?= Html::encode($this->title) ?></h1>
16
17 |     <?= $this->render('_form', [
18 |         'model' => $model,
19 |     ]) ?>
20
21 | </div>

```

Figura 4.13 Código de vista update.php.

En la figura 4.13 se muestran las etiquetas que pertenecen a la opción actualizar registro y al igual que la opción *Create* reenvía la forma y el modelo, solo que agrega la variable id, con la información del registro que se va actualizar.

## Index.php

En este archivo, se despliegan los registros existentes, se configuran los campos a mostrar y de qué manera se indexan, se muestra en la figura 4.14.

```

10 | $this->title = 'Materias';
11 | $this->params['breadcrumbs'][] = $this->title;
12 | ?>
13 | <div class="materia-index">
14
15 |     <h1><?= Html::encode($this->title) ?></h1>
16 |     <?php // echo $this->render('_search', ['model' => $searchModel]); ?>
17
18 |     <p>
19 |         <?= Html::a('Create Materia', ['create'], ['class' => 'btn btn-success']) ?>
20 |     </p>
21
22 |     <?= GridView::widget([
23 |         'dataProvider' => $dataProvider,
24 |         'filterModel' => $searchModel,
25 |         'columns' => [
26 |             ['class' => 'yii\grid\SerialColumn'],
27
28 |             'id',
29 |             'materia',
30 |             'carrera',
31 |             'posición',
32 |             'fecha_creacion',
33 |             // 'admin_id',
34 |             // ['attribute' => 'admin_id',
35 |             // 'value' => 'users.username',
36 |             // ],
37 |             // ],
38 |             ['class' => 'yii\grid\ActionColumn'],
39 |         ],
40 |     ); ?>
41 | </div>

```

Figura 4.14 Código de vista index.php.

En la figura 4.14 se muestra el código de la página *Index*, donde en la función *GridView::widget()* se despliegan todos los registros existentes, teniendo como variable de datos *\$dataProvider* y como variable de búsqueda *\$searchModel*, donde al momento de querer realizar una consulta se podrá hacer por id, materia, carrera o fecha de creación.

## View.php

En este archivo se muestra la información detallada y se le agregan los links para borrar (*Delete*) o para modificar el registro (*Update*), así como los campos que se presentan, se muestra en la figura 4.15.

```
9  $this->title = $model->id;
10 $this->params['breadcrumbs'][] = ['label' => 'Materias', 'url' => ['index']];
11 $this->params['breadcrumbs'][] = $this->title;
12 ?>
13 <div class="materia-view">
14
15     <h1><?= Html::encode($this->title) ?></h1>
16
17     <p>
18         <?= Html::a('Update', ['update', 'id' => $model->id], ['class' => 'btn btn-primary']) ?>
19         <?= Html::a('Delete', ['delete', 'id' => $model->id], [
20             'class' => 'btn btn-danger',
21             'data' => [
22                 'confirm' => 'Are you sure you want to delete this item?',
23                 'method' => 'post',
24             ],
25         ]) ?>
26     </p>
27
28     <?= DetailView::widget([
29         'model' => $model,
30         'attributes' => [
31             'id',
32             'materia',
33             'carrera',
34             'posicion',
35             'fecha_creacion',
36             'admin_id',
37         ],
38     ]) ?>
39
40 </div>
```

Figura 4.15 Código de vista view.php.

En la figura 4.15 se muestra el código que se ejecuta cuando se selecciona ver un registro, en la función *DetailView::widget()* se detallan los campos a desplegar en pantalla, agregados por dos links, uno para actualizar *Html::a("Update")* el otro para borrar *Html::a("Delete")*, donde este último incluye un mensaje en forma de pregunta para confirmar si quiere borrar ese registro.

## Search.php

En este archivo se llevan a cabo las búsquedas por los campos definidos, ya sea por Id o nombre, se configuran los campos a mostrar se muestra en la figura 4.16.

```
4 use yii\widgets\ActiveForm;
5
6 /* @var $this yii\web\View */
7 /* @var $model app\models\MateriaSearch */
8 /* @var $form yii\widgets\ActiveForm */
9 >>
10
11 <div class="materia-search">
12
13     <?php $form = ActiveForm::begin([
14         'action' => ['index'],
15         'method' => 'get',
16     ]); ?>
17
18     <?= $form->field($model, 'id') ?>
19
20     <?= $form->field($model, 'materia') ?>
21
22     <?= $form->field($model, 'carrera') ?>
23
24     <?= $form->field($model, 'posicion') ?>
25
26     <?= $form->field($model, 'fecha_creacion') ?>
27
28     <?php // echo $form->field($model, 'admin_id') ?>
29
30     <div class="form-group">
31         <?= Html::submitButton('Search', ['class' => 'btn btn-primary']) ?>
32         <?= Html::resetButton('Reset', ['class' => 'btn btn-default']) ?>
33     </div>
34
35     <?php ActiveForm::end(); ?>
36
37 </div>
```

Figura 4.16 Código de vista search.php.

En la figura 4.16 se muestran los campos utilizados para búsquedas en la forma activa *ActiveForm()*, los cuales se utilizan en la primera parte de la página *Index* al momento de querer realizar una consulta por alguno de ellos, se le agrega un botón de entrada *submit* y uno de reinicio *reset* para borrar búsquedas anteriores.

## MateriaController.php

En el controlador se manejan las funciones con que se accederá a las vistas, ya que una de las particularidades de este Framework, es que utiliza el protocolo MVC (Modelo, Vista, Controlador) y la manera de acceder a las opciones es por medio de cada una de las funciones que se definan para *Index*, *View*, *Create*, *Update*, *Delete* y *Search* y cada una de ellas va precedida de la palabra “*action*”, como se muestra en la figura 4.17.

```
36 public function actionIndex()
37 {
38     $searchModel = new MateriaSearch();
39     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
40
41     return $this->render('index', [
42         'searchModel' => $searchModel,
43         'dataProvider' => $dataProvider,
44     ]);
45 }
46
47 /**
48  * Displays a single Materia model.
49  * @param integer $id
50  * @return mixed
51  * @throws NotFoundHttpException if the model cannot be found
52  */
53 public function actionView($id)
54 {
55     return $this->render('view', [
56         'model' => $this->findModel($id),
57     ]);
58 }
59
60 /**
61  * Creates a new Materia model.
62  * If creation is successful, the browser will be redirected to the 'view' page.
63  * @return mixed
64  */
65 public function actionCreate()
66 {
67     $model = new Materia();
68
69     if ($model->load(Yii::$app->request->post()) && $model->save() {
```

Figura 4.17 Código de vista MateriaController.php.

En la figura 4.17 se puede observar que en la mayoría de las funciones lo que se ejecuta es reenviar el modelo o la búsqueda, sin embargo, en ocasiones es necesario ejecutar código para que los datos se envíen de una manera adecuada o que requieran de algún filtro para que solo se envíen los datos que se vayan a utilizar.

### 4.4.3 Formas Dinámicas

Otra herramienta muy útil fue Formas Dinámicas Wbraganca (Braganca, 2015), donde en base al modelo inicial, se pueden agregar submodelos de forma dinámica, para evitar estar enviando a subpáginas y que se vuelva más tedioso el sistema, en la figura 4.18 se muestra un ejemplo de su utilización.

It is widget to yii2 framework to clone form elements in a nested manner, maintaining accessibility.

Houses	Rooms							
<input type="text" value="House 1"/>	<table border="1"><thead><tr><th>Description</th><th></th></tr></thead><tbody><tr><td><input type="text" value="House 1 - Room 1"/></td><td><input type="button" value="-"/></td></tr><tr><td><input type="text" value="House 1 - Room 2"/></td><td><input type="button" value="-"/></td></tr></tbody></table>	Description		<input type="text" value="House 1 - Room 1"/>	<input type="button" value="-"/>	<input type="text" value="House 1 - Room 2"/>	<input type="button" value="-"/>	<input type="button" value="+"/> <input type="button" value="-"/>
Description								
<input type="text" value="House 1 - Room 1"/>	<input type="button" value="-"/>							
<input type="text" value="House 1 - Room 2"/>	<input type="button" value="-"/>							
<input type="text" value="House 2"/>	<table border="1"><thead><tr><th>Description</th><th></th></tr></thead><tbody><tr><td><input type="text" value="House 2 - Room 1"/></td><td><input type="button" value="-"/></td></tr><tr><td><input type="text" value="House 2 - Room 2"/></td><td><input type="button" value="-"/></td></tr></tbody></table>	Description		<input type="text" value="House 2 - Room 1"/>	<input type="button" value="-"/>	<input type="text" value="House 2 - Room 2"/>	<input type="button" value="-"/>	<input type="button" value="+"/> <input type="button" value="-"/>
Description								
<input type="text" value="House 2 - Room 1"/>	<input type="button" value="-"/>							
<input type="text" value="House 2 - Room 2"/>	<input type="button" value="-"/>							

Figura 4.18 Forma Dinámica Wbraganca.

En la figura 4.18 se puede ver que se pueden agregar modelos de tipo *House* y de cada modelo agregar un submodelo de tipo *Room*.

En el sistema se utiliza para agregar a los ejercicios en tres ocasiones, que son Incisos, Tutoriales y Ayuda PDF.

Incisos. Ya que en cada problema puede haber más de una pregunta, es posible que se pueda dividir en varias interrogantes, el sistema permite que se agreguen cero o hasta 4 incisos, se muestra en la figura 4.19.



**Figura 4.19 Incisos en forma dinámica con Ejercicios.**

Tutoriales. A cada uno de los ejercicios se le podrá agregar links a tutoriales en YouTube previamente revisados por el Maestro, para que sirvan de apoyo en la resolución de los mismos, se pueden agregar como mínimo 1 y máximo 4, se muestra en la figura 4.20.



**Figura 4.20 Tutoriales en forma dinámica con Ejercicios.**

Ayuda PDF. Así como con los Tutoriales, también se podrán agregar documentos que contengan ejemplos escritos, previamente revisados por el Maestro, para que sirvan de apoyo en la resolución de los mismos, se podrán agregar como mínimo 1 y máximo 4, se muestra en la figura 4.21.

The image shows a dynamic form interface. At the top, there is a grey header bar with a close button (X) and the text 'Ayuda PDF'. Below this, there is a white container with a grey header bar containing the text 'PDF' and a green plus button and a red minus button. The main content area of the container has two input fields: one labeled 'Nombre Pdf' and another labeled 'Pdf'.

**Figura 4.21 Ayuda PDF en forma dinámica con Ejercicios.**

En el siguiente código de la forma dinámica Wbagraanca, cambiando los nombres del contenedor, modelo y botones:

```
'widgetBody' => '.container-items'

'insertButton' => '.add-item', // css class

'deleteButton' => '.remove-item', // css class

'model' => $modelsAddress[0],
```

Se pueden utilizar las formas dinámicas que se necesiten, sin que exista conflicto entre ellas.

En una misma pantalla, utilizando formas dinámicas se puede acceder a otros modelos de una forma más eficaz, sin necesidad de que el usuario se llegue a confundir, el código por parte de los creadores se puede ver en el anexo.

---

## CAPÍTULO 5. PRUEBAS Y RESULTADOS

En este capítulo se abordan los puntos siete, ocho y nueve del método utilizado donde se mostraran las diferentes pruebas que se le realizaron a la plataforma, esto para revisar que se cumpla con los requerimientos funcionales, que sea amigable e intuitiva y utilizar el modelo CUEVA para determinar su usabilidad.

### 5.1 Usabilidad

La evaluación de usabilidad con el modelo CUEVA se le aplicó a 4 usuarios, 2 sin experiencia, 1 con experiencia y 1 con nivel experto, arrojando los siguientes resultados que se muestran en las tablas 5.1, 5.2, 5.3, 5.4 y figura 5.1.

FACTOR	Abreviatura	Ponderación	Puntos
Comunicación y Comprensibilidad	CC	3	7.2
Facilidad Estetica	CFE	3	7.5
Operatividad	CO	4	10.8
Facilidad de uso	CFU	4	12
Aprendizaje- Factores Humanos	CA	3	7.5
Facilidad de Comprensión	CFC	3	7.5
Entrenamiento	CE	3	6.9
Documentación	CD	3	6.3
	Calificacion		65.7

**Tabla 5.1 Evaluación de usuario sin experiencia 1.**

En la tabla 5.1 se muestra la evaluación del usuario sin experiencia 1, el usuario comentó que la plataforma podría manejar colores más agradables y que los menús podrían tener otro tipo de letra y de tamaño más grande, afectando la calificación de los factores CC y CFE. También comentó que al no haber manejado plataformas similares anteriormente, los ejercicios podrían resolverse en computadora, no solo seleccionar la respuesta de las opciones, afectando el factor CE y CFC.

FACTOR	Abreviatura	Ponderacion	Puntos
Comunicación y Comprensibilidad	CC	4	9.6
Facilidad Estetica	CFE	3	7.5
Operatividad	CO	3	8.1
Facilidad de uso	CFU	4	12
Aprendizaje- Factores Humanos	CA	4	10
Facilidad de Comprensión	CFC	2	5
Entrenamiento	CE	3	6.9
Documentación	CD	3	6.3
Calificacion			65.4

**Tabla 5.2 Evaluación de usuario sin experiencia 2.**

En la tabla 5.2 se muestra la evaluación del usuario sin experiencia 2, el usuario comentó que la plataforma podría mejorar en la selección de ejercicios al momento de realizar una práctica, sin necesidad de seleccionar materia, unidad y tema, aunque se le explicó el motivo por el cual se organiza de esa manera su percepción fue otra, afectando al factor CFE, CO y CFC.

FACTOR	Abreviatura	Ponderación	Puntos
Comunicación y Comprensibilidad	CC	4	9.6
Facilidad Estetica	CFE	3	7.5
Operatividad	CO	3	8.1
Facilidad de uso	CFU	3	9
Aprendizaje- Factores Humanos	CA	2	5
Facilidad de Comprensión	CFC	4	10
Entrenamiento	CE	3	6.9
Documentación	CD	3	6.3
Calificacion			62.4

**Tabla 5.3 Evaluación de usuario con experiencia.**

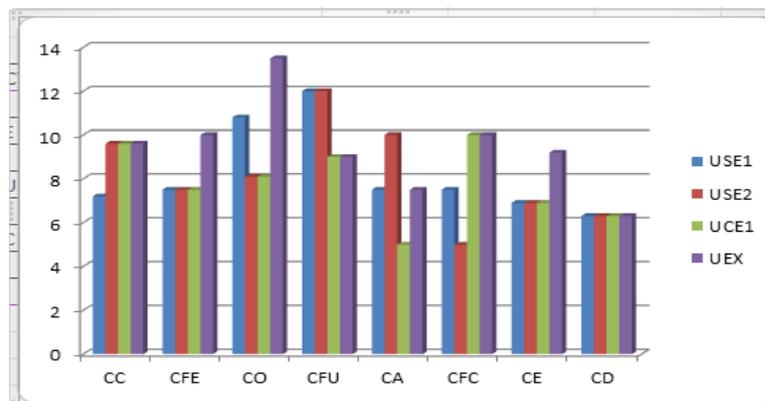
En la tabla 5.3 se muestra la evaluación del usuario con experiencia, el usuario comentó que la plataforma comparada con otras de nivel internacional puede mejorar el menú, los colores y la manera en que está organizado el contenido.

Se le explicó que esta plataforma está orientada al personal del Instituto Tecnológico de Nogales/TecNM, y que su objetivo es la realización de prácticas con ejercicios para mejorar el nivel de aprendizaje, sin embargo, su percepción afectó los factores CFE, CO, CE Y CD.

FACTOR	Abreviatura	Ponderación	Puntos
Comunicación y Comprensibilidad	CC	4	9.6
Facilidad Estetica	CFE	4	10
Operatividad	CO	5	13.5
Facilidad de uso	CFU	3	9
Aprendizaje- Factores Humanos	CA	3	7.5
Facilidad de Comprensión	CFC	4	10
Entrenamiento	CE	4	9.2
Documentación	CD	3	6.3
Calificacion			75.1

**Tabla 5.4 Evaluación de usuario experto.**

En la tabla 5.4 se muestra la evaluación del usuario experto, el usuario entendió el concepto del diseño y desarrollo de la plataforma, mencionando que el proceso de selección de ejercicios y realización de prácticas puede mejorar al momento de desplegar la información de preguntas y respuestas agregando separación de colores y organizar la información para que se muestre menos confusa, así como también agregar un enlace para un manual del usuario electrónico, en base a esto afecto los factores CFU, CA y CD.



**Figura 5.1 Grafica comparativa de los resultados.**

Además, se realizaron pruebas unitarias y de aceptación mediante funciones de *Codeception TDD (Test-Driven Development)* al sitio con operaciones CRUD, no se encontraron problemas al momento de dar de alta materias, unidades, temas y subtemas. No se observaron conflictos de comunicación entre el navegador y la base de datos y no se encontraron problemas de datos perdidos o que no se haya podido realizar el guardado de los mismos, en la figura 5.2 se muestra un ejemplo de una prueba unitaria.

```
1 <?php
2
3 use yii\helpers\Url;
4
5 class LoginCest
6 {
7     public function ensureThatLoginWorks(AcceptanceTester $I)
8     {
9         $I->amOnPage(Url::toRoute('/site/login'));
10        $I->see('Login', 'h1');
11
12        $I->amGoingTo('try to login with correct credentials');
13        $I->fillField('input[name="LoginForm[yorch]"', 'admin');
14        $I->fillField('input[name="LoginForm[KAZAM3107]"', 'admin');
15        $I->click('login-button');
16        $I->wait(2); // wait for button to be clicked
17
18        $I->expectTo('see user info');
19        $I->see('Logout');
20    }
21 }
22
```

Figura 5.2 Prueba unitaria para Login.

Como se muestra en la figura 5.2, utilizando código del *framework Codeception* podemos realizar funciones y ejecutarlas para validar que los datos se estén enviando correctamente, que las páginas se carguen y que se muestren los resultados obtenidos en pantalla, sin necesidad de hacerlo de forma manual con el explorador.

En la función *ensureThatLoginWorks()* para validar el correcto inicio de sesión se observan algunos elementos que se describen a continuación:

- *I->amOnPage()*. Define en qué página se encuentra.
- *I->see()*. Contiene los elementos que están desplegados.

- *l->amGoingTo()*. Se refiere a la página destino.
- *l->fillField()*. Contiene la captura de campos, en este caso sería usuario y contraseña.
- *l->click()*. Menciona el botón que se va a presionar para enviar la información.
- *l->wait()*. El tiempo que se tarda en lo que se presiona el botón.
- *l->expectedTo()*. La información que se espera ver en la página destino.
- *l->see()*. Los elementos que se esperan ver.

Ejecutando esta función, puede arrojar como resultado que obtuvo algún error en alguno de los elementos mencionados o bien que se logró con éxito el inicio de sesión.

## 5.2 Pruebas del proyecto en relación a los requerimientos funcionales

Un aspecto importante que se debe cubrir al hacer la evaluación de un software es verificar que sus módulos cumplan con lo requerido por el cliente final. Para ello se hace una comparación de los resultados obtenidos contra lo especificado en los requerimientos funcionales descritos en el punto 3.2.

Las tablas 5.5 y 5.6 muestran que todos los requerimientos funcionales cumplieron con el diseño del prototipo final y fueron alcanzados con éxito.

**Tabla 5.5 Cumplimiento de Requerimientos Funcionales (parte 1).**

Ref. #	Requerimiento	Requerimiento fue cumplido	
		SI	NO
REG01	El sistema permite al usuario registrarse en el sitio y crear una cuenta.	X	
LOG01	El sistema permite identificarse en el sitio para poder acceder a las opciones disponibles.	X	
LOG02	El sistema permite al usuario con sesión	X	

	iniciada cerrar la misma.		
<b>ADM01</b>	El sistema permite a los usuarios con privilegios de Administrador dar de alta materias.	X	
<b>ADM02</b>	El sistema permite a los usuarios con privilegios de Administrador registrar unidades en las materias disponibles.	X	
<b>ADM03</b>	El sistema permite a los usuarios con privilegios de Administrador registrar temas en las unidades de las materias disponibles.	X	
<b>ADM04</b>	El sistema permite a los usuarios con privilegios de Administrador agregar ejercicios a los temas disponibles.	X	

Tabla 5.6 Cumplimiento de Requerimientos Funcionales (parte 2).

Ref. #	Requerimiento	Requerimiento fue cumplido	
		SI	NO
<b>ALM01</b>	El sistema permite a los usuarios con privilegios de Alumno registrarse en el sistema.	X	
<b>ALM02</b>	El sistema permite a los usuarios con privilegios de Alumno resolver los ejercicios seleccionados en cada práctica.	X	
<b>ALM03</b>	El sistema permite a los usuarios con privilegios de Alumno ver las calificaciones obtenidas en prácticas realizadas.	X	

### 5.3 Pruebas del sitio en diferentes navegadores.

Se hicieron pruebas del sitio de manera local en los navegadores Microsoft Internet Explorer 11, Microsoft Edge versión 2015, Mozilla Firefox versión 48, Google Chrome en su versión 74 y el funcionamiento fue el mismo, no hubo problemas de visualización de ningún control, imagen o estilo de las páginas.

Todas la operaciones del sitio se apegan a como fueron descritas en los casos de uso de alto nivel y como se planificaron en los diagramas de secuencia.

### 5.4 Caso de Estudio

A continuación se menciona un caso de estudio para el uso de la interface del sistema, iniciando en la figura 5.3 con la selección de registro de nuevo usuario.



Figura 5.3 Captura de pantalla para selección de registro de nuevo usuario.

En el menú inicial se selecciona la opción registro para comenzar con la captura de los datos como se muestra en la figura 5.4.



The screenshot shows a registration form with the following fields and values:

- Correo Electrónico: kazam80@gmail.com
- Contraseña: [Redacted]
- Confirme Contraseña: [Redacted]
- Nombre: Juan
- Apellido Paterno: Pérez
- Apellido Materno: Martínez
- Sexo: M
- Institución: ITN

Figura 5.4 Captura de pantalla de datos de nuevo usuario.

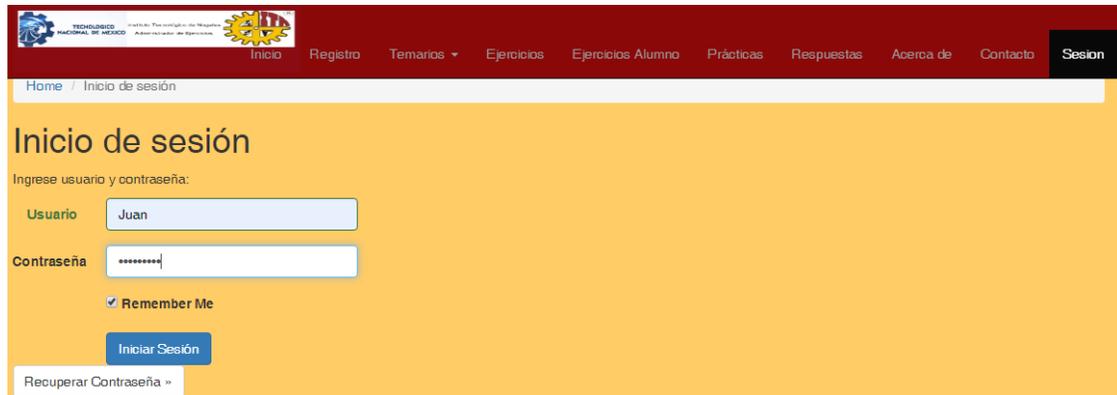
Una vez ingresados los datos se presiona el botón Registrarse para enviar la información a la base de datos y al mismo tiempo se envía un *link* al correo electrónico del usuario para terminar de activar el registro.

Al momento de activar el registro, ya puede iniciar sesión el nuevo usuario seleccionando la opción del menú principal, como se muestra en la figura 5.5.



Figura 5.5 Captura de pantalla de selección de inicio de sesión.

Una vez en la página de inicio de sesión ingresamos usuario y contraseña y presionamos botón Iniciar Sesión, se muestra en la figura 5.6.



The screenshot shows the login interface of the 'Administrador de Ejercicios' system. At the top, there is a navigation bar with links: Inicio, Registro, Temarios, Ejercicios, Ejercicios Alumno, Prácticas, Respuestas, Acerca de, Contacto, and a highlighted 'Sesion' button. Below the navigation bar, the page title is 'Inicio de sesión'. The main content area is yellow and contains the following elements: a prompt 'Ingrese usuario y contraseña:', a 'Usuario' input field containing 'Juan', a 'Contraseña' input field with masked characters, a checked 'Remember Me' checkbox, a blue 'Iniciar Sesión' button, and a link for 'Recuperar Contraseña >'. The system logo is visible in the top left corner.

**Figura 5.6 Captura de pantalla de inicio de sesión.**

Si usuario y contraseña están correctos nos envía a pantalla de inicio y en lugar de estar habilitada la opción de Sesión, ya se puede observar el correo electrónico del usuario, como se muestra en la figura 5.7.



**Figura 5.7 Captura de pantalla de usuario con inicio de sesión.**

Después de registrarse como Administrador (Docente), se comienza dando de alta la materia del temario a dar entrada, como se ve en la figura 5.8 y 5.9.

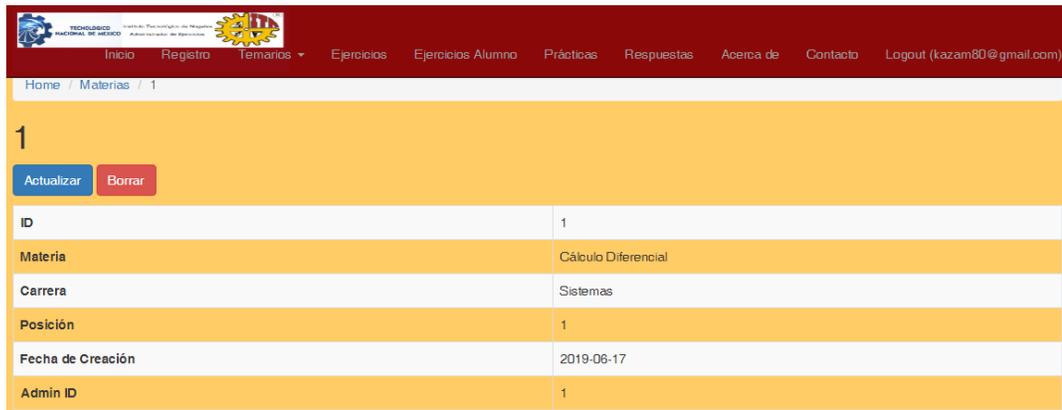


Figura 5.8 Captura de Pantalla de selección de Materias, Unidades, Temas o Subtemas.



Figura 5.9 Captura de pantalla de Alta de Materia.

Una vez ingresado el nombre de la materia, la carrera a la que pertenece, confirmar fecha de creación y administrador, se presiona el botón Guardar y si no existe error en la información de entrada, se guardará el registro en la base de datos y se redirigirá a la pantalla View, como se muestra en la figura 5.10.



The screenshot shows a web application interface with a dark red header containing navigation links: Inicio, Registro, Temarios, Ejercicios, Ejercicios Alumno, Prácticas, Respuestas, Acerca de, Contacto, and Logout (kazam80@gmail.com). Below the header, the breadcrumb trail reads 'Home / Materias / 1'. The main content area features a large number '1' and two buttons: 'Actualizar' (blue) and 'Borrar' (red). Below these is a table with the following data:

ID	1
Materia	Cálculo Diferencial
Carrera	Sistemas
Posición	1
Fecha de Creación	2019-06-17
Admin ID	1

Figura 5.10 Captura de pantalla de View después de alta de Materia.

Una vez dada de alta la materia, lo siguiente es crear las unidades que la integran, por lo que se ingresa a la opción de Temario y se selecciona Unidades, como lo muestra la figura 5.11.



The screenshot shows a web application interface with a dark red header containing navigation links: Inicio, Registro, Temarios, Ejercicios, Ejercicios Alumno, Prácticas, Respuestas, Acerca de, Contacto, and Logout (kazam80@gmail.com). Below the header, the breadcrumb trail reads 'Home / Unidades / Crear Unidad'. The main content area is titled 'Crear Unidad' and contains a form with the following fields:

- Materia:** Cálculo Diferencial (dropdown menu)
- Título:** 1 Números reales (text input)
- Fecha de Creación:** 28/11/2018 (date input)
- Admin ID:** Juan (dropdown menu)

At the bottom of the form is a green 'Guardar' button.

Figura 5.11 Captura de pantalla de Alta de Unidad.

Primero es necesario seleccionar la materia a la que va pertenecer la unidad, después se ingresa el Título de la unidad, se confirma fecha de creación y administrador, se presiona botón Guardar, si no hay error en la información ingresada, se graba en la base de datos y se envía a la pantalla View.

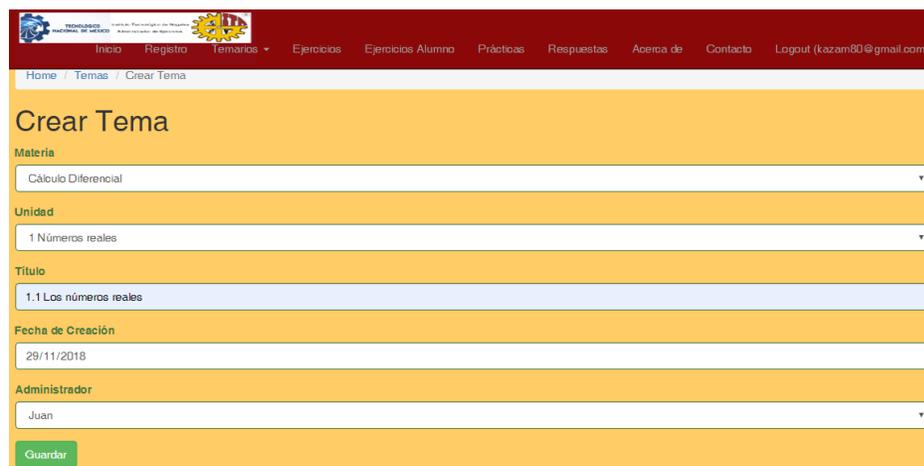
En la pantalla View se puede ver la información que se guardó en la base de datos, como se muestra en la figura 5.12.



ID	1
Título	1 Números reales
Posición	1
Materia	1
Fecha de Creación	2018-11-28
Admin ID	1

**Figura 5.12** Captura de pantalla de View con información de Unidad guardado.

Una vez dada de alta la unidad, lo siguiente es crear los temas que la integran, por lo que se ingresa a la opción de Temario y se selecciona Temas, como lo muestra la figura 5.13.



Crear Tema

Materia:

Unidad:

Título:

Fecha de Creación:

Administrador:

**Figura 5.13** Captura de pantalla de Alta de Tema.

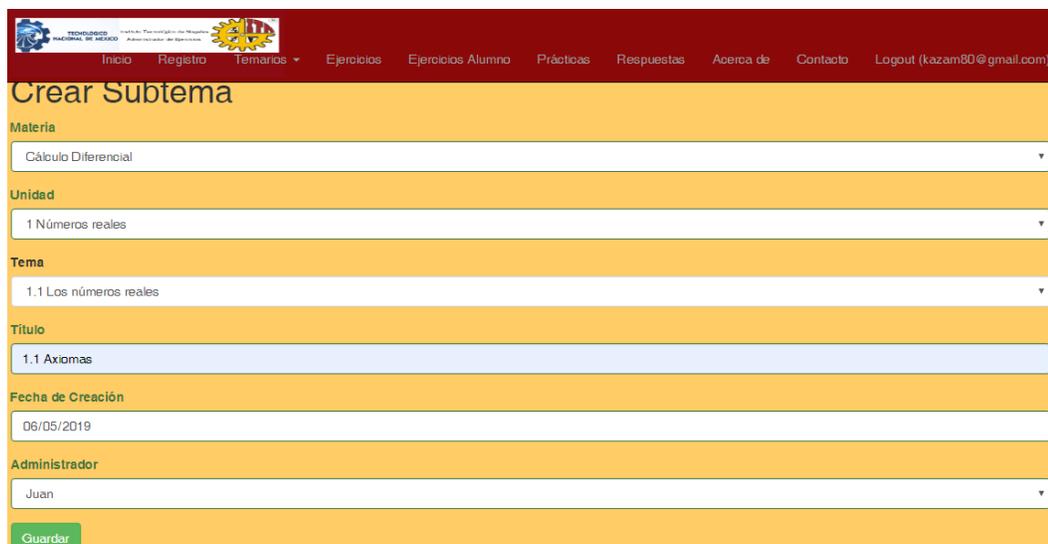
Primero es necesario seleccionar la materia y unidad a la que va pertenecer el tema, después se ingresa el título del tema, se confirma fecha de creación y administrador, se presiona botón Guardar, si no hay error en la información ingresada, se graba en la base de datos y se envía a la pantalla View como se muestra en la figura 5.14.



ID	1
Título	1.1 Los números reales
Posición	1
Unidad	1
Fecha de Creación	2018-11-29
Administrador	1

**Figura 5.14** Captura de pantalla de View con información de Tema guardado.

Una vez dado de alta el tema, lo siguiente es crear los subtemas que lo integran, por lo que se ingresa a la opción de Temario y se selecciona Subtemas, como lo muestra la figura 5.15.



**Crear Subtema**

Materia:

Unidad:

Tema:

Título:

Fecha de Creación:

Administrador:

**Figura 5.15** Captura de pantalla de Alta de Subtema.

Primero es necesario seleccionar la materia, unidad y tema al que va pertenecer el subtema, después se ingresa el Título del subtema, se confirma fecha de creación y administrador, se presiona botón Guardar, si no hay error en la información ingresada, se graba en la base de datos y se envía a la pantalla View como se muestra en la figura 5.16.



ID	2
Título	1.1 Axiomas
Tema	3
Fecha de Creación	2019-05-06
Administrador	1

**Figura 5.16 Captura de pantalla de View con información de Subtema guardado.**

Este proceso se repite para todo el temario como lo son Materias, Unidades, Temas y Subtemas que se desee ingresar. Cada uno de ellos cuenta con su respectivo CRUD, donde se puede realizar cualquier consulta, modificación e incluso baja de algún elemento del Temario.

Después de tener listo nuestro Temario, lo siguiente es ingresar la información que será necesaria para el alta de los ejercicios que se realizan y que forman parte de las prácticas de los alumnos.

Primero, es necesario ingresar la respuesta o respuestas que se enlazarán con los ejercicios o incisos, por lo que del menú principal, se selecciona Respuestas, como lo muestra la figura 5.17.



**Figura 5.17 Captura de pantalla de entrada de Respuesta.**

En esta pantalla es necesario formular una clave de la respuesta a ingresar para fácil identificación al momento de enlazarlo con el ejercicio o inciso correspondiente, donde las primeras dos letras pertenecen a la materia, el siguiente es un número que pertenece a la unidad, después el número de tema y otro de subtema y al final, el número del ejercicio que se ingresará.

Esto se realiza ya que al momento de ingresar un nuevo ejercicio o inciso, la selección de la respuesta es por medio de un *dropdownlist* y así es más fácil su localización.

Después de formular la clave, se procede a ingresar una descripción de la respuesta, por lo que puede ser un texto, número, fórmula, entre otros. Continuando después con la selección de la imagen, que se relaciona con la descripción, donde esta puede encontrarse en formato .pdf, .img, .png, .jpg con la finalidad que el docente pueda agregar algún archivo de internet o bien, alguna foto de su misma clase desarrollada en pizarrón, esto se muestra la figura 5.18.

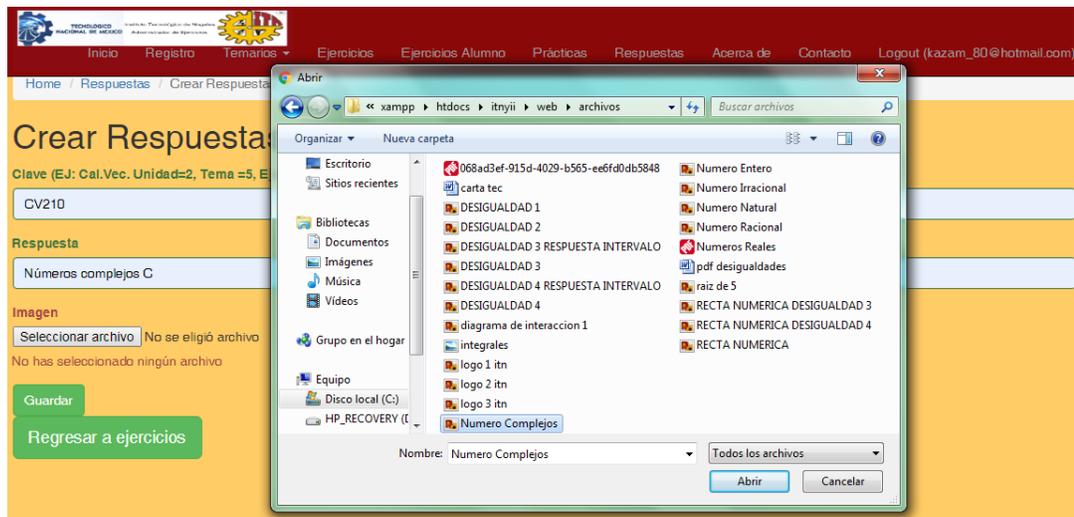


Figura 5.18 Captura de pantalla de selección de imagen de respuesta.

Una vez ingresada la información, se presiona el botón Guardar, si no se encuentran errores, la información se guarda en la base de datos y se redirige a la pantalla View como se muestra en la figura 5.19.



Figura 5.19 Captura de pantalla de View con información de respuesta guardada.

Una vez ingresadas las respuestas, lo siguiente es dar de alta el o los ejercicios, así como los incisos correspondientes, por lo que se ingresa desde el menú principal en la opción Ejercicios, como lo muestra la figura 5.20.



The screenshot shows a web interface for creating exercises. At the top, there is a navigation bar with the following items: Inicio, Registro, Temarios (with a dropdown arrow), Ejercicios, Ejercicios Alumno, Prácticas, Respuestas, Acerca de, Contacto, and Logout (kazam\_80@hotmail.com). Below the navigation bar is a yellow header with the text 'Crear Ejercicios'. The main form area is also yellow and contains several dropdown menus: 'Materia' (Cálculo Diferencial), 'Unidad' (1 Números reales), 'Tema' (1.1 Los números reales), 'Subtema' (1.1 Números reales), and 'Respuesta' (QD111). At the bottom of the form is a green button labeled 'Crear nueva respuesta'.

Figura 5.20 Captura de pantalla de Alta de Ejercicio (sección 1).

Para el ingreso de un ejercicio, son algunas secciones, por lo que se inicia seleccionando la materia, unidad, tema y subtema, todos estos, desde los *dropdownlist* correspondientes, así como la clave de la respuesta previamente dada de alta, en caso de no haberlo hecho, se presiona el botón Crear nueva respuesta, y se redirige a la alta correspondiente.

Una vez seleccionada esta información, se avanza hacia abajo en la pantalla para continuar con la captura de la siguiente sección, que es ingresar los incisos necesarios en caso de que el ejercicio cuente con más de un problema a resolver y varias respuestas a formular, esto se muestra en la figura 5.21.

TECNOLOGÍAS  
NACIONAL DE MÉXICO

Inicio Registro Temarios Ejercicios Ejercicios Alumno Prácticas Respuestas Acerca de Contacto Logout (kazam\_80@hotmail.com)

Incisos

Incisos + Agregar

Inciso

Descripción del Inciso No. de Respuesta

Calcula Intervalo en la Gráfica CD111

Inciso

Descripción del Inciso No. de Respuesta

Representa la solución en intervalo CD111

**Figura 5.21** Captura de pantalla de ingreso de incisos a un ejercicio.

En esta sección se agrega la descripción de cada inciso y se selecciona del *dropdownlist* la respuesta correspondiente. El sistema puede agregar hasta 4 incisos como máximo o bien ninguno como mínimo.

La siguiente sección es agregar los enlaces de apoyo para la resolución de los ejercicios, como lo son los tutoriales en video y los archivos PDF, esto se muestra en la figura 5.22.

TECNOLOGÍAS  
NACIONAL DE MÉXICO

Inicio Registro Temarios Ejercicios Ejercicios Alumno Prácticas Respuestas Acerca de Contacto Logout (kazam\_80@hotmail.com)

Tutoriales

Tutorial + -

Tutorial Link Tut

Números Reales https://www.youtube.com/watch?v=eAgQLrZDdk

Ayuda PDF

PDF + -

Nombre del Archivo Enlace Pdf

Números Reales C:\xampp\htdocs\itny\weblarchivos\numeros\_reales.pdf

**Figura 5.22** Captura de pantalla de selección de enlaces de apoyo.

Para agregar *links* a video tutoriales, o archivos PDF, se agrega una descripción y la ruta donde se encuentra ubicado, el sistema permite agregar como máximo 4 y como mínimo 1.

Ya para finalizar, la última sección solicita ingresar el nivel de dificultad del ejercicio, el enunciado, la imagen y confirmar fecha de creación y administrador, como se muestra en la figura 5.23.

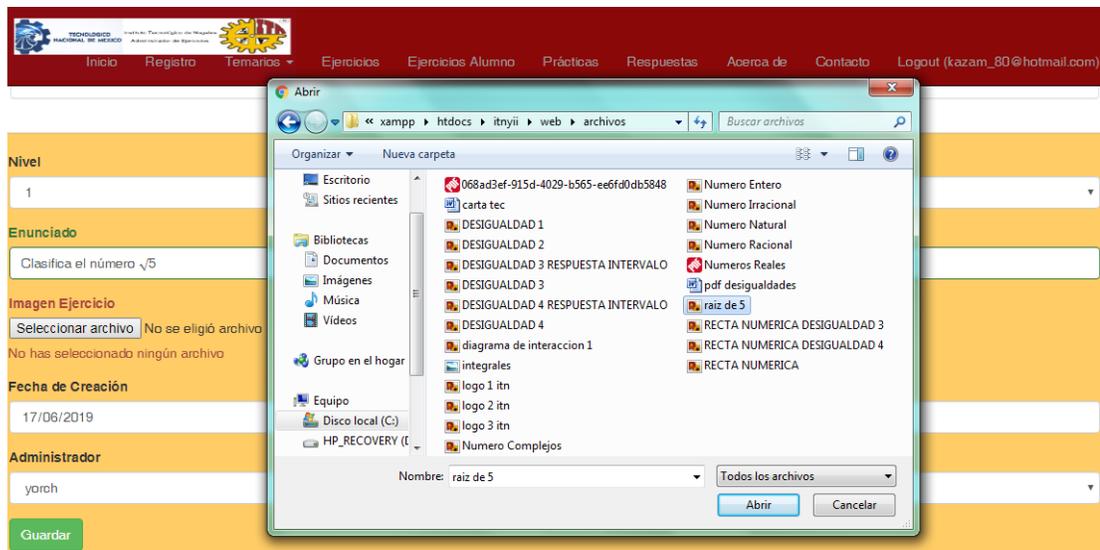


Figura 5.23 Captura de pantalla de sección final del ingreso del Ejercicio.

Una vez ingresada la información de todas las secciones del ejercicio, se presiona el botón guardar, si no existe error en la entrada de datos, se guarda el registro en la base de datos y se redirige a la pantalla View, como se muestra en la figura 5.24.



23	
Actualizar	Borrar
ID	23
Subtema	1
Respuesta	1
Tutorial (Agregar enlace con URL completo)	
Pdf	
Nivel	1
Enunciado	Clasifica el número raíz de 5
Imagen	archivos/raiz de 5.png
Tipo Respuesta	1
Fecha de Creación	2019-05-06
Administrador	1

Figura 5.24 Captura de pantalla de View con información del ejercicio guardado.

Después de agregar los ejercicios de cada subtema, ya es posible realizar una práctica, por lo que desde el menú principal se selecciona Ejercicios Alumno, como se muestra en la figura 5.25.



Figura 5.25 Captura de pantalla de menú de Ejercicios Alumno.

En esta pantalla se selecciona la materia, unidad, tema y subtema de los ejercicios que se desea realizar una práctica, una vez seleccionados se presiona botón Ver Ejercicios para que muestre los resultados obtenidos, como se muestra en la figura 5.26.

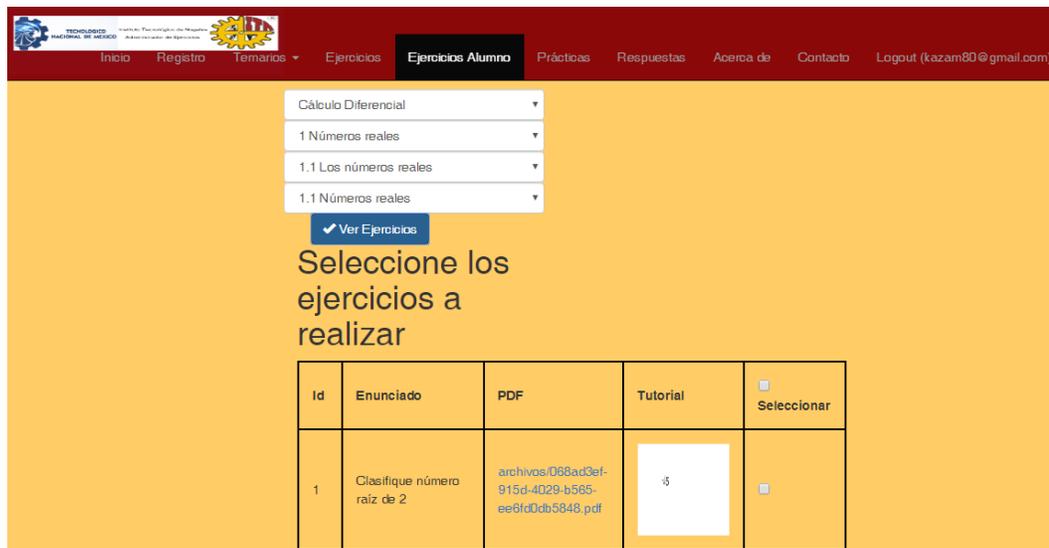


Figura 5.26 Captura de pantalla de selección de ejercicios para una práctica.

Una vez seleccionados los ejercicios haciendo *click* en los *checkbox*, presionar el botón Enviar Ejercicios como se muestra en la figura 5.27.

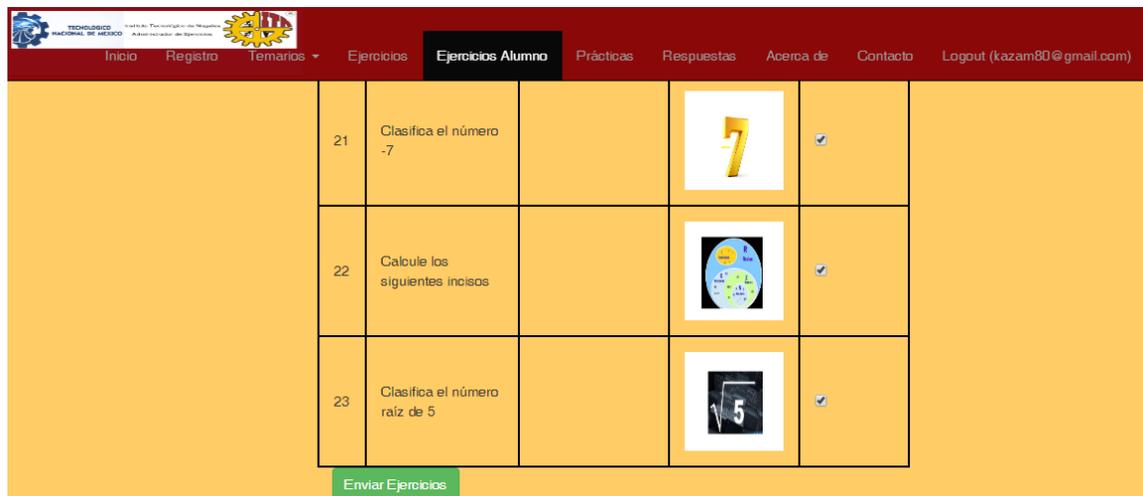


Figura 5.27 Captura de pantalla de ejercicios seleccionados.

El sistema se redirige a la pantalla de resolución de los ejercicios, donde se muestra la descripción del enunciado, la imagen, los video tutoriales y documentos PDF de apoyo, cuatro respuestas aleatorias con su respectiva imagen.

En caso de existir incisos correspondientes, de igual manera los despliega con sus respectivas cuatro respuestas aleatorias, mencionando además, que el orden de los ejercicios aparecen en forma aleatoria a la selección de la pantalla anterior, esto se muestra en la figura 5.28.

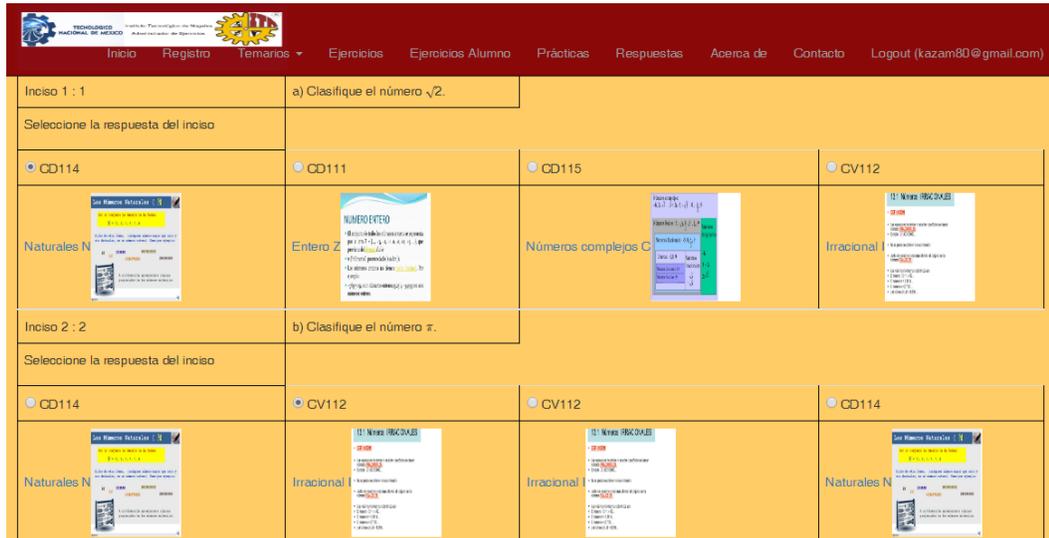


Figura 5.28 Pantalla de resolución de ejercicios.

Una vez seleccionada la respuesta de cada ejercicio e inciso en caso de existir, se presiona botón Enviar Práctica, como se muestra en figura 5.29.



Figura 5.29 Captura de pantalla de selección de respuestas.

El sistema redirige a la pantalla donde muestra los resultados de la resolución de ejercicios e incisos, tanto correctos como incorrectos, calculando la calificación en base al total de ejercicios correctos (o ejercicios que se compongan de varios incisos) desplegando el número de ejercicios, la calificación y el usuario que la realizó, esto se muestra en la figura 5.30.



Id	Enunciado	Incisos	Respondió
23	Clasifica el número raíz de 5		1 1 Correcto
22	Calcule los siguientes incisos		
1	a) Clasifique el número $\sqrt{2}$ .	2 2 Correcto	0.5 1 2 1
2	b) Clasifique el número $\pi$ .	2 2 Correcto	1 2 2 1
21	Clasifica el número -7		3 3 Correcto

Figura 5.30 Captura de pantalla de Resultados de Ejercicios.

Para guardar la práctica, se le agrega un nombre y se presiona botón Guardar Práctica, si es un nombre válido, el sistema almacenará el registro en la base de datos. Para consultar alguna práctica, desde el menú principal se selecciona la opción Prácticas, esto se muestra en la figura 5.31.



Seleccione Práctica Juan

Práctica4

Ver Ejercicios

Figura 5.31 Captura de pantalla de consulta de prácticas.

Una vez seleccionada la práctica, se presiona el botón Ver Ejercicios y el sistema redirige a ver los resultados de la práctica, esto se observa en la figura 5.32.

Resultados de la práctica

Nombre de la práctica: Práctica4  
Calificación: 100

Id	Enunciado	Clave Respuesta	Clave Respondida	Id Inc	Inciso	Inciso Respuesta	Inciso Respondido
23	Clasifica el número raíz de 5	CD111	CD111				
21	Clasifica el número -7	CD113	CD113				
22	Calcule los siguientes incisos			1	a) Clasifique el número $\sqrt{2}$ .	CV112	CV112
22	Calcule los siguientes incisos			2	b) Clasifique el número	CV112	CV112

**Figura 5.32 Captura de pantalla de resultados de una Práctica.**

Con esto concluye el proceso del caso de estudio que comenzó con la alta de un nuevo administrador (Docente), el paso por la alta de la materia, unidad, tema y subtema para después ingresar los ejercicios relacionados y concluir con la realización de una práctica y su consulta.

---

## CAPÍTULO 6. CONCLUSIONES Y COMENTARIOS FINALES

Este capítulo aborda el punto diez y final del método, donde en base a las pruebas realizadas podemos emitir resultados y conclusiones.

### 6.1 Análisis de resultados

Al llevar a cabo pruebas con la plataforma de manera local y en la red “Depi”, se observó que con los resultados obtenidos con el modelo CUEVA la usabilidad percibida para los usuarios con o sin experiencia es muy similar, mostrando que la plataforma se desarrolló de manera intuitiva.

Con las pruebas realizadas mediante el *framework Codeception* se observó que la interfaz es estable, las páginas se cargan con los elementos e información correctos y no se encontró pérdida de datos.

Con el cumplimiento de los requerimientos funcionales, se muestra que el planteamiento del proyecto se hizo de manera correcta, así como el diseño y finalización.

El caso de estudio que se realizó, donde el docente dio de alta contenido para agregar ejercicios al temario, logrando que el alumno realice una práctica y pueda consultar su calificación en base a los aciertos obtenidos, se observó que en el proceso no se encontró ningún problema y la información se almacenó de manera correcta.

Se logró que el sitio web sea de fácil uso, amigable, estable y puede ser considerado en un futuro su implementación como ayuda en la gestión de plan de estudios de los cursos de acuerdo al programa del TecNM.

## 6.2 Conclusiones

Con la revisión de las pruebas realizadas, se concluye que la plataforma es estable y sólida, así como tener una percepción de usabilidad correcta y un grado de intuición notable para el usuario, no encontrando errores al momento de ingresar y procesar información, ni tampoco al guardar y consultar información de la base de datos.

Con el desarrollo del prototipo de este trabajo de tesis se confirma que la hipótesis planteada es verdadera y es posible implementar un ambiente virtual para la administración y evaluación de ejercicios de matemáticas, específicamente, Calculo Diferencial, utilizando ambientes de programación Web y almacenar sus resultados en una base de datos para su administración y control.

Este sitio puede ser utilizado por el TecNM como la herramienta de apoyo principal para el mejoramiento del aprendizaje de matemáticas, ya que consolida la manera de estudiar del hoy en día, como lo son ejercicios de internet y tutoriales, así como la guía del temario oficial.

Inicialmente el enfoque fue con la materia de Cálculo Diferencial, pero el software es factible de utilizarse en otras asignaturas, siempre y cuando los ejercicios se adapten al diseño creado.

## 6.3 Trabajos futuros y mejoras

Teniendo como base esta tesis y prototipo se sugieren los siguientes temas.

1. Implementación del Laboratorio Virtual para dispositivos móviles.
2. Materializar su implementación trabajando de manera conjunta con los maestros para la realización de ejercicios y tutoriales.

3. Seguimiento en la actualización de ejercicios por parte de los docentes.
4. Combinar ejercicios de varios subtemas para una misma práctica.
5. Agregar la opción de crear cursos y tareas con el propósito de que cuenten como puntos para los alumnos en su asignatura.

En versiones posteriores se propone agregar la opción de realizar exámenes, inscribirse a cursos de algún docente en particular para poder asignar tareas y llevar un seguimiento de calificaciones por alumno de sus diferentes prácticas, con la finalidad de que el docente y el alumno tengan más interacción a través de la plataforma.

---

## ANEXO.- Código de formas dinámicas Wbraganca.

```
<div class="panel panel-default">
  <div class="panel-heading"><h4><i class="glyphicon glyphicon-
envelope"></i> Addresses</h4></div>
  <div class="panel-body">
    <?php DynamicFormWidget::begin([
      'widgetContainer' => 'dynamicform_wrapper', // required: only
alphanumeric characters plus "_" [A-Za-z0-9_]
      'widgetBody' => '.container-items', // required: css class
selector
      'widgetItem' => '.item', // required: css class
      'limit' => 4, // the maximum times, an element can be cloned
(default 999)
      'min' => 1, // 0 or 1 (default 1)
      'insertButton' => '.add-item', // css class
      'deleteButton' => '.remove-item', // css class
      'model' => $modelsAddress[0],
      'formId' => 'dynamic-form',
      'formFields' => [
        'full_name',
        'address_line1',
        'address_line2',
        'city',
        'state',
        'postal_code',
      ],
    ]); ?>
    <div class="container-items"><!-- widgetContainer -->
    <?php foreach ($modelsAddress as $i => $modelAddress): ?>
      <div class="item panel panel-default"><!-- widgetBody -->
        <div class="panel-heading">
          <h3 class="panel-title pull-left">Address</h3>
          <div class="pull-right">
            <button type="button" class="add-item btn btn-success
btn-xs"><i class="glyphicon glyphicon-plus"></i></button>
            <button type="button" class="remove-item btn btn-
danger btn-xs"><i class="glyphicon glyphicon-minus"></i></button>
          </div><div class="clearfix"></div>
        </div><div class="panel-body">
```

---

## REFERENCIAS

Alarcón-Aldana, A. C. (2014, 02 12). *https://scielo.conicyt.cl*. Retrieved 07 31, 2019, from [https://scielo.conicyt.cl/scielo.php?script=sci\\_arttext&pid=S0718-07642014000300016](https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642014000300016)

Alvarez, M. A. (1999, 12 01). *desarrolloweb.com*. Retrieved 06 07, 2019, from <https://desarrolloweb.com/articulos/que-es-mvc.html>

Braganca, W. (2015, 05 01). *github.com/wbraganca*. Retrieved 03 01, 2019, from <https://github.com/wbraganca/yii2-dynamicform>

Davidson, L. (2008). *Pro Sql Server*. Apress: Apress, Kline, Kevin; Klein, Scott; Windisch, Kurt.

Dougiamas, M. (2002, feb 08). *Moodle*. Retrieved feb 10, 2018, from <https://moodle.org>

José A. Ruipérez-Valiente, P. J.-M. (2015). ALAS-KA: A learning analytics extension for better understanding the learning process in the Khan Academy platform. *Science Direct*, 139-148.

Khan, S. (2006, feb 08). *Khan Academy*. Retrieved feb 10, 2018, from <https://www.khanacademy.org/>

Khilkov, S. (2001, 01 05). *https://www.yiiframework.com/doc/guide/2.0/es/test-overview*. Retrieved 05 03, 2019, from <https://www.yiiframework.com/doc/guide/2.0/es/test-overview>

Larman, C. (2004). *UML y patrones: Introducción al análisis y diseño orientado a objetos*. Pearson.

---

McGraw-Hill Education. (2013, FEB 08). *ALEKS.COM*. Retrieved FEB 09, 2018, from [www.aleks.com](http://www.aleks.com)

Pearson. (2018, abr 23). *www.pearsonmylabandmastering.com*. Retrieved abr 23, 2018

PHP GROUP. (2001, 01 01). *php.net*. Retrieved 06 07, 2019, from <https://www.php.net/copyright.php>

Princely Ifinedo, J. P. (2017, Octubre 01). *Science Direct*. Retrieved Febrero 14, 2018, from Business undergraduates' perceived use outcomes of Moodle in a blended learning environment: [www.sciencedirect.com/science/article/pii/S0736585317304860](http://www.sciencedirect.com/science/article/pii/S0736585317304860)

Puncey, I., & York, R. (2011). *Beginning CSS Cascading Style Sheets for web design*. Indianapolis, Indiana: Wiley Publishing, Inc.

Qin, Z., Xing, J., & Zheng, X. (2008). *Software Architecture*. New York: Springer.

Scotty D. Craig, X. H. (2013, Junio 10). *The impact of a technology-based mathematics after-school program using ALEKS on student's knowledge and behaviors*. Retrieved Febrero 14, 2018, from [www.sciencedirect.com:www.sciencedirect.com/science/article/pii/S0360131513001619](http://www.sciencedirect.com:www.sciencedirect.com/science/article/pii/S0360131513001619)

Sharma, N., Perniu, L., Chong, R., Iker, A., Nandan, C., Mitea, A.-C., y otros. (2010). *Database Fundamentals Ideal for application developers and administrators*. Markham, Canada: IBM Corporation.

Skinner, J. (2008, 01 18). *sublimetext.com*. Retrieved 06 07, 2019, from <https://www.sublimetext.com/>

---

UNESCO. (2005). Las tecnologías de la información y la comunicación en la enseñanza. *Cómo crear nuevos entornos de aprendizaje abierto por medio de las TIC*. Recuperado el 2 de Enero de 2014, de <http://unesdoc.unesco.org/images/0013/001390/139028s.pdf>

Vega, J. A. (2015). *DESARROLLO DE UN SITIO WEB PARA EL MANEJO Y CONTROL DE LOS CURSOS DE CÁLCULO DIFERENCIAL DEL TECNOLÓGICO UTILIZANDO ASP.NET, SQL SERVER y AZURE*. NOGALES SONORA.

Weinschenk, S. (1997). *GUI design essentials for Windows 95, Windows 3.1 and World Wide Web*. John Wiley & Sons, Inc., ; Jamar, Pamela; Yeo, Sarah.