

**Facultad de Tecnologías Educativas**

**Sistema informático para el desarrollo de habilidades matemáticas esenciales**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autores:**

Rosmery González Suarez

Yordany Ponce Alderete

**Tutores:**

Dr.C. Ailec Granda Dihigo

Ms.C. Reina Victoria Estrada Nelson

Dr.C. Dunia María Colome Cedeño

La Habana, 2024

**DECLARACIÓN DE AUTORÍA**

Los autores del trabajo de diploma con título “**Sistema informático para el desarrollo de habilidades matemáticas esenciales**”, conceden a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declaran como únicos autores de su contenido. Para que así conste firman la presente a los <día> días del mes de <mes> del año <año>.

|  |  |
| --- | --- |
|  |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Autor | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Autor |
|  |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Tutor  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Tutor | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Firma del Tutor |

# 

**DATOS DE CONTACTO**

**Dr.C. Dunia María Colomé Cedeño**

**Profesora Titular, graduada de Ingeniería en Ciencias Informáticas en 2007. Doctora en Ciencias Técnicas, 2013. Máster en Tecnología Educativa por la Universidad de las Islas Baleares, España 2012. Posee la certificación de IREB Foundation Level. Decana, Vicerrectora de Investigación y Posgrado y Secretaria General del Comité del Partido en la Universidad de las Ciencias Informáticas. Tiene 16 años de experiencia en docencia de pregrado y posgrado y en la dirección del trabajo metodológico.**

**Ms.C. Reina Victoria Estrada Nelson**

**Graduada desde 1982 en la Universidad de Granma y MSc en Educación Superior desde 1988 en el Centro de Estudio de perfeccionamiento de la ES en la Universidad de Oriente, con 41 año de experiencia continuada en la Educación Superior. Trabajó en la Universidad de Granma hasta el 2006 año en que pasó a la Universidad de las Ciencias Informáticas. Obtuvo la categoría docente de Profesora Auxiliar en el año 1995. Durante su vida profesional ha sido jefa de distintos niveles metodológicos desde brigada, año, carrera y miembro nacional de la carrera de M.V. Ha cumplido diferentes misiones de trabajo académicas y científicas en varios países, Cuenta con experiencia administrativa desde jefe de dpto, VD, decana y VRF y VRU. Actualmente trabaja en el Centro de Innovación y Calidad de la Educación e imparte docencia en la Facultad de Tecnologías Educativas. Integra tres proyectos de investigación, imparte cursos de posgrado, integró el grupo de trabajo de desarrollo local y formó parte de la conceptualización de La Lisa como municipio inteligente.**

**Ms.C. Ailec Granda Dihigo**

**Ingeniera en Ciencias Informáticas graduada en la Universidad de las Ciencias Informáticas con Titulo de Oro. En el año 2010 se graduó de Master en Tecnología Educativa y en el 2013 de Doctora en Ciencias de la Educación en la Universidad de las Islas Baleares. Es profesora titular de la disciplina Ingeniería y Gestión de Software. Es directora del centro de Innovación y Calidad de la Educación en la UCI y se desempeña como líder de proyectos de investigación.**

**AGRADECIMIENTO**

**DEDICATORIA**

**RESUMEN**

Con frecuencia llegan estudiantes a las aulas de clase con dificultades en el manejo de conceptos y poca comprensión en temas que son básicos para avanzar en contenidos. Un ejemplo de esta dificultad se encuentra en la enseñanza y el aprendizaje de las matemáticas, comenzando por la comprensión y dominio conceptual referido a las cuatro operaciones básicas: adición, sustracción, multiplicación y división. Esta investigación tiene como propósito desarrollar un sistema informático que apoye el proceso de enseñanza y aprendizaje de las operaciones básicas en los niños de educación primaria. Para la realización de este sistema, se empleó como metodología de desarrollo de software XP, PHP como lenguaje de programación, laravel como marco de trabajo y MySQL como gestor de base de datos. Se desarrollaron todas las etapas del ciclo de vida del software culminando con la realización de las pruebas.

**Palabras claves:** sistema informático, matemáticas, operaciones básicas

**ABSTRACT**

*Frequently, students arrive in the classroom with difficulties in handling concepts and little understanding of topics that are basic to advance in content. An example of this difficulty is found in the teaching and learning of mathematics, starting with the comprehension and conceptual mastery of the four basic operations: addition, subtraction, multiplication and division; which are essential to advance in the contents of higher grades. The purpose of this research is to present a proposal for improvement in the levels of comprehension in the process of building basic operations for primary school students. To realize this system, XP was used as a software development methodology, PHP as a programming language, laravel as a framework, and MySQL as a database manager.*

**Keywords***: computer system, mathematics, basic operations*

**ÍNDICE**

[**INTRODUCCIÓN** 1](#_Toc167699239)

[**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA** 5](#_Toc167699240)

[1.1 Introducción 5](#_Toc167699241)

[1.2 Conceptos asociados a la presente investigación 5](#_Toc167699242)

[1.3 Estudio de sistemas informáticos para la enseñanza de la matemática 6](#_Toc167699243)

[1.3.1 Soluciones existentes a nivel internacional 6](#_Toc167699244)

[1.3.2 Soluciones existentes a nivel nacional 7](#_Toc167699245)

[1.4 Metodología de desarrollo de software 8](#_Toc167699246)

[1.4.1 Programación Extrema 11](#_Toc167699247)

[1.5 Herramientas y lenguajes informáticos 13](#_Toc167699248)

[1.6 Conclusiones del capítulo 17](#_Toc167699249)

[**CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA** 18](#_Toc167699250)

[2.1 Introducción 18](#_Toc167699251)

[2.2 Planificación 18](#_Toc167699252)

[2.2.1 Historias de Usuario 18](#_Toc167699253)

[2.2.2 Requisitos no funcionales 24](#_Toc167699254)

[2.2.3 Estimación del esfuerzo 25](#_Toc167699255)

[2.2.4 Plan de Iteraciones 26](#_Toc167699256)

[2.2.5 Plan de Entregas 27](#_Toc167699257)

[2.3 Diseño 27](#_Toc167699258)

[2.3.1 Patrón Arquitectónico 28](#_Toc167699259)

[2.3.2 Patrones de Diseño 29](#_Toc167699260)

[2.3.3 Tarjetas CRC 32](#_Toc167699261)

[2.4 Modelo de datos 34](#_Toc167699262)

[2.5 Conclusiones del capitulo 35](#_Toc167699263)

[**CAPÍTULO 3: Implementación y Pruebas** 36](#_Toc167699264)

[3.1 Introducción 36](#_Toc167699265)

[3.2 Estándares de codificación 36](#_Toc167699266)

[3.3 Desarrollo por iteraciones 37](#_Toc167699267)

[3.3.1 Iteración 1 37](#_Toc167699268)

[3.3.2 Iteración 2 43](#_Toc167699269)

[3.4 Conclusiones del capitulo 50](#_Toc167699270)

[**CONCLUSIONES** 51](#_Toc167699271)

[**Recomendaciones** 52](#_Toc167699272)

[**BIBLIOGRAFÍA** 53](#_Toc167699273)

**ÍNDICE DE TABLAS**

[Tabla 1.1 Análisis de los antecedentes 8](#_Toc167265412)

[Tabla 2.1 HU Autenticar Usuario 19](#_Toc167265413)

[Tabla 2.2 HU Registrar Usuarios 20](#_Toc167265414)

[Tabla 2.3 HU Gestionar Usuarios 20](#_Toc167265415)

[Tabla 2.4 HU Gestionar Tareas 21](#_Toc167265416)

[Tabla 2.5 HU Gestionar Grados 21](#_Toc167265417)

[Tabla 2.6 HU Generar Reportes 22](#_Toc167265418)

[Tabla 2.7 HU Crear Operaciones Matemáticas 22](#_Toc167265419)

[Tabla 2.8 HU Notificar a familiares 23](#_Toc167265420)

[Tabla 2.9 Estimación del esfuerzo 25](#_Toc167265421)

[Tabla 2.10 Plan de Iteraciones 26](#_Toc167265422)

[Tabla 2.11 Plan de Entregas 27](#_Toc167265423)

[Tabla 2.12 Tarjeta CRC User 31](#_Toc167265424)

[Tabla 2.13 Tarjeta CRC Familiar 31](#_Toc167265425)

[Tabla 2.14 Tarjeta CRC Result 32](#_Toc167265426)

[Tabla 2.15 Tarjeta CRC Degree 32](#_Toc167265427)

[Tabla 2.16 Tarjeta CRC Task 33](#_Toc167265428)

[Table 3.1 Tarea de desarrollo Autenticar Usuario 37](#_Toc167265429)

[Table 3.2 Tarea de desarrollo Registrar Usuario 37](#_Toc167265430)

[Table 3.3 Tarea de desarrollo Insertar Usuarios 38](#_Toc167265431)

[Table 3.4 Tarea de desarrollo Eliminar Usuarios 38](#_Toc167265432)

[Table 3.5 Tarea de desarrollo Mostrar Usuarios 38](#_Toc167265433)

[Table 3.6 Tarea de desarrollo Modificar Usuarios 39](#_Toc167265434)

[Table 3.7 Tarea de desarrollo Crear Operaciones Matematicas 39](#_Toc167265435)

[Table 3.8 Tarea de desarrollo Insertar Tareas 41](#_Toc167265436)

[Table 3.9 Tarea de desarrollo Eliminar Tareas 42](#_Toc167265437)

[Table 3.10 Tarea de desarrollo Mostrar Tareas 42](#_Toc167265438)

[Table 3.11 Tarea de desarrollo Modificar Tareas 43](#_Toc167265439)

[Table 3.12 Tarea de desarrollo Insertar Grados 43](#_Toc167265440)

[Table 3.13 Tarea de desarrollo Eliminar Grados 43](#_Toc167265441)

[Table 3.14 Tarea de desarrollo Mostrar Grados 44](#_Toc167265442)

[Table 3.15 Tarea de desarrollo Modificar Grados 44](#_Toc167265443)

[Table 3.16 Tarea de desarrollo Generar Reportes 45](#_Toc167265444)

[Table 3.17 Tarea de desarrollo Notificar a familiares 45](#_Toc167265445)

**ÍNDICE DE FIGURAS**

[*Figura 1: Evidencia del Patrón Experto* 29](#_Toc168685171)

[*Figura 2: Evidencia del Patrón Controlador* 30](#_Toc168685172)

[*Figura 3: Evidencia del Patrón Creador* 30](#_Toc168685173)

[*Figura 4: Evidencia del Patrón Alta Cohesión* 30](#_Toc168685174)

[*Figura 5: Evidencia del Patrón Factory* 31](#_Toc168685175)

[*Figura 6: Modelo de Datos* 35](#_Toc168685176)

[*Figura 7: Diagrama de despliegue* 35](#_Toc168685177)

[*Figura 8: Diagrama de clases* 36](#_Toc168685178)

[*Figura 9: Resultados de las pruebas unitarias Iteración 1* 43](#_Toc168685179)

[*Figura 10: Resultados de las pruebas de aceptación Iteración 1* 44](#_Toc168685180)

[*Figura 11: Resultados de las pruebas unitarias Iteración 2* 49](#_Toc168685181)

[*Figura 12: Resultados de las pruebas de aceptación Iteración 2* 50](#_Toc168685182)

**OPINIÓN DE LOS TUTORES**

# **INTRODUCCIÓN**

La educación es un proceso de socialización, o sea, de preparación del hombre para la vida en sociedad, lo que requiere de aprendizajes muy diversos; de conocimientos, habilidades, normas y valores, que identifican al sujeto como miembro de una comunidad cultural, de un pueblo, de una nación. Es un proceso que se inicia desde el propio nacimiento y continúa a lo largo de toda la vida (Tocora Lozano & García González, 2018).

La educación formal es aquella educación institucionalizada, intencionada y planificada por organizaciones públicas y organismos privados acreditados para ello que, en su conjunto, constituyen el sistema educativo formal del país (CINE, 2011). Esta debe ser un proceso individualizado y basado en la observación de las necesidades y capacidades de cada estudiante. La escuela compone el eje y núcleo central y primordial de la educación; tanto en la comunidad como en la familia, por tanto, su papel como institución está enmarcado en dos direcciones una formar y la otra preparar, para lograr la formación integral del hombre para la sociedad en la cual se va a desenvolver (Tocora Lozano & García González, 2018).

La educación es importante en cualquier etapa de desarrollo social, tiene la función de socializar a las nuevas generaciones; lo cual significa prepararlas para vivir en sociedad, para convivir como ciudadanos responsables, capaces de cumplir funciones cívicas y poseedores de un sistema de valores que les permita actuar en correspondencia con su contexto social (Tocora Lozano & García González, 2018).

A mediados del siglo XX surgieron los primeros ordenadores y se inició la digitalización de la información, esto dio paso al nacimiento de las Tecnologías de la Información y la Comunicación (TIC), estas se desarrollan a partir de los avances científicos producidos y las telecomunicaciones y son el conjunto de tecnologías que permiten el acceso, producción, tratamiento y comunicación de información presentada en diferentes códigos (texto, imagen, sonido) (Ortí, 2011).

Las tecnologías informáticas y otros aspectos de la cultura digital han cambiado las formas en las que las personas viven, trabajan, juegan y aprenden, lo que a su vez ha repercutido en cómo se construye y distribuye el conocimiento y el poder en todo el mundo. Un área que ha sido profundamente transformada por las TIC es la educación.

La transformación que ha sufrido las TIC, han logrado convertirse en instrumentos educativos, capaces de mejorar la calidad educativa del estudiante, revolucionando la forma en que se obtiene, se maneja y se interpreta la información (Aguilar, 2012).

Dentro de los roles que asumen cada agente educativo, los estudiantes actuales, utilizan las herramientas tecnológicas para facilitar el aprendizaje; esta evolución surgió desde las primeras concepciones con la calculadora, el televisor, la grabadora, entre otras; sin embargo, el progreso ha sido tal que los recursos tecnológicos se han convertido en recursos educativos, donde la búsqueda por mejorar el aprendizaje trae consigo la tarea de involucrar la tecnología con la educación. Y es con la docencia que se viene completando el proceso de enseñanza-aprendizaje. El uso de las TIC supone romper con los medios tradicionales, pizarras, lapiceros, etc; y dar paso a la función docente, basada en la necesidad de formarse y actualizar sus métodos en función de los requerimientos actuales (Hernandez, 2017).

Con el presente auge de las tecnologías y en el proceso de informatización digital que se encuentra Cuba, es una necesidad que desde todos los ámbitos se analice como estas tecnologías se pueden explotar en el sector educativo.

A partir de un análisis del grupo de proyectos de la ciudad inteligente del municipio de la Lisa para identificar los principales problemas desde el punto de vista de la educación se detectó una deficiencia significativa en el aprendizaje de las operaciones básicas en niños de educación primaria. Con frecuencia, llegan a las aulas de clases estudiantes con dificultades en la comprensión y dominio conceptual de las cuatro operaciones básicas: adición, sustracción, multiplicación y división. Todos los estudiantes no aprenden de la misma manera ni al mismo tiempo y la educación primaria carece de recursos educativos específicamente diseñados para abordar esta diversidad.

A raíz de la situación problemática anteriormente expuesta, se plantea como **problema de la investigación**: ¿Cómo contribuir a la mejora del proceso de enseñanza-aprendizaje de las operaciones básicas en los niños de educación primaria utilizando las tecnologías de la información y la comunicación?

Teniendo como **objeto de estudio** de esta investigación los sistemas informáticos de apoyo al aprendizaje de la matemática.

Para dar solución a la problemática anteriormente descrita se establece como **objetivo general** desarrollar un sistema informático que apoye el proceso de enseñanza-aprendizaje de las operaciones básicas en los niños de educación primaria.

Se define como **campo de acción** los sistemas informáticos de apoyo al aprendizaje de las operaciones básicas de la matemática.

Para dar cumplimiento al objetivo general se derivan los siguientes **objetivos específicos:**

1. Definir el marco teórico conceptual referente al desarrollo de sistemas para el aprendizaje de la matemática
2. Realizar el análisis y el diseño del sistema informático a desarrollar.
3. Implementar el sistema informático.
4. Realizar los diferentes tipos de pruebas de software.

Con el objetivo de dar cumplimiento a lo anteriormente descrito, se elaboran las siguientes **tareas de** **investigación:**

* Definición del marco teórico conceptual referente al desarrollo de sistemas para el aprendizaje de la matemática.
* Análisis de sistemas homólogos para tener más conocimientos en el diseño de sistemas para el aprendizaje de las operaciones básicas de la matemática.
* Selección de las principales herramientas informáticas y metodologías de desarrollo de software para la realización del sistema.
* Implementación del sistema.
* Realización de los principales tipos de prueba de software.

En el desarrollo de la investigación se utilizarán los siguientes **métodos científicos:**

**Métodos teóricos:**

* **Analítico-sintético:** Para examinar una variedad de fuentes bibliográficas y documentos relacionados con el objeto de estudio, identificándose conceptos claves, definiciones relevantes y otros elementos útiles que contribuyan al desarrollo de la investigación.
* **Histórico-lógico:** Para estudiar la existencia de sistemas homólogos empleados por otras organizaciones con el fin de obtener conocimientos para la realización del sistema

**Métodos Empíricos:**

* **Entrevista:** Se aplicó a trabajadores de escuelas primarias para ver las principales necesidades presentes en el proceso de aprendizaje de las operaciones básicas y el grado de informatización con el que cuentan para realizar la gestión de la información.

La presente investigación está estructurada de la siguiente manera: introducción, tres capítulos, conclusiones y recomendaciones. A continuación, una breve descripción de lo que tratará cada capítulo.

**Capítulo 1: “Fundamentación teórica del objeto de estudio”.** En este capítulo se identifican los conceptos claves relacionados con el tema de investigación y se revisan los trabajos más recientes y relevantes en el campo de estudio de los sistemas para la gestión de la información con fines educativos. Se definirá la metodología de desarrollo de software a emplear y las herramientas para el desarrollo del sistema.

**Capítulo 2: “Propuesta de solución”.** En este capítulo es donde comienza la exploración, análisis y diseño del software. Se define la estructura del proyecto, planificación temporal y los requerimientos del sistema.

**Capítulo 3: “Validación de la propuesta de solución”.** En este último capítulo es donde se describen las etapas de implementación y pruebas de software.

# **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

## 1.1 Introducción

En la **actualidad**, las **tecnologías** desempeñan un papel **fundamental** en el ámbito de la **educación**, se ha convertido en una herramienta esencial para el aprendizaje. Plataformas en línea, aplicaciones educativas y recursos digitales permiten a los estudiantes acceder a información de manera rápida y eficiente.

Para comprender lo previamente mencionado, es necesario examinar y analizar los elementos que respaldarán teórica, metodológica y conceptualmente la investigación actual. Los siguientes apartados se centrarán en estos aspectos. Se llevará a cabo un análisis del estado actual de conocimiento, la elección de la metodología de desarrollo de software y las herramientas y lenguajes que se emplearán en el proyecto.

## 1.2 Conceptos asociados a la presente investigación

En el campo de la investigación, es esencial tener un buen dominio de los conceptos básicos. Estos no solo proporcionan una base teórica sólida, sino que también sirven como cimiento para el desarrollo de conocimientos durante la investigación.

**Sistema:** Un sistema es la composición de varios elementos que están conectados entre sí para lograr un objetivo específico. Dichos elementos deben estar ordenados según las tareas que deban cumplir para lograr el propósito planteado (Lifeder, 2022).

Los sistemas tienen entradas, salidas, procesos internos y retroalimentación, lo que les permite funcionar de manera organizada y eficiente.

**Sistema Informático:** Son sistemas que interrelacionan elementos tanto físicos como abstractos, como lo son el hardware (físico, como teclados, pantallas, procesadores, etc.) y el software (abstracto, como sistemas operativos, bases de datos, protocolos de red, etc.) (Lifeder, 2022).

El sistema informático recibe datos de entrada, los procesa utilizando el software y el hardware, y luego genera resultados en forma de datos de salida.

**Matemáticas:** Ciencia deductiva que estudia las propiedades de los entes abstractos, como números, figuras geométricas o símbolos, y sus relaciones. Estudio de la cantidad considerada en abstracto o aplicada(RAE, 2020)**.**

Las matemáticas son un lenguaje universal y una herramienta fundamental para comprender el mundo que nos rodea, resolver problemas y tomar decisiones informadas en diversos contextos.

**Aprendizaje:** Es un proceso mediante el cual adquirimos conocimientos, habilidades, actitudes y comportamientos a través de la experiencia, la práctica y la interacción con el entorno (Kolb, 2023).

El aprendizaje es un aspecto fundamental del desarrollo humano y está presente a lo largo de toda la vida. Permite adaptarse al entorno, resolver desafíos, desarrollar nuevas habilidades y comprender el mundo que nos rodea. Además, el aprendizaje continuo es esencial para el crecimiento personal, profesional e intelectual.

**Operaciones Básicas:** Presentan un procedimiento para resolverse basado en números, para ello es importante que los estudiantes sepan agrupar, desagrupar y escribir cantidades, así como la posición numérica de los números (Cruz, 2016).

Las operaciones básicas constituyen la base de la matemática, son indispensables para adquirir conocimientos superiores.

**Aprendizaje de la matemática:**

Elaprendizaje de la matemática se refiere al proceso mediante el cual los estudiantes adquieren conocimientos, habilidades y comprensión en el campo de la matemática. Esto incluye la comprensión de conceptos matemáticos, la capacidad de resolver problemas, el razonamiento lógico y la aplicación de métodos matemáticos en situaciones del mundo real.

## 1.3 Estudio de sistemas informáticos para la enseñanza de la matemática

 Debido al crecimiento exponencial de Internet y la facilidad para crear sitios web, se ha producido un incremento significativo en su número. Por lo tanto, es crucial llevar a cabo investigaciones sobre sistemas similares en el ámbito educativo, tanto a nivel nacional como internacional.

## 1.3.1 Soluciones existentes a nivel internacional

**Cokitos**

Es un sitio web que surgió en el 2011 con el objetivo de ofrecer juegos educativos en línea para niños. Estos juegos están diseñados para ser divertidos y atractivos, al mismo tiempo que brindan oportunidades de aprendizaje en diversas áreas, como matemáticas, lenguaje, ciencias, geografía y más. Su objetivo es estimular el pensamiento, la resolución de problemas, la memoria y otras habilidades cognitivas. Además de los juegos, también ofrece actividades, ejercicios y recursos educativos complementarios que los padres y educadores pueden utilizar para respaldar el aprendizaje de los niños. Ofrece una experiencia educativa de calidad, presentando contenidos cuidadosamente seleccionados y adaptados a las necesidades y niveles de los niños (*COKITOS | Juegos Educativos Online para Niños y Adultos*).

**IXL**

Es un sitio web creado en 1998 en San Mateo, California, Estados Unidos por Paul Mishkin. Este programa es una experiencia de aprendizaje inmersivo que ofrece contenidos de matemáticas muy completos y adaptados a los programas de estudios de infantil a 6to de primaria (*IXL | Ejercicios de Matemáticas*).

[**Tux of Math Command**](https://tux4kids.alioth.debian.org/tuxmath/download.php)

**Es un juego con el que ejercitar las cuatro operaciones matemáticas básicas, es decir, suma, resta, multiplicación y división. Los ejercicios se precipitan pantalla abajo y el jugador tiene que acabar con ellos antes de que lleguen al suelo poniendo el resultado** (*Jugar Tux Math en línea*)**.**

## 1.3.2 Soluciones existentes a nivel nacional

**Identifica y Cuenta**

“Identifica y cuenta” es un juego didáctico creado en 2021 por la Editorial Pueblo y Educación diseñado para niños de corta edad que les ayudará a identificar los números y a contar. Activará su concentración y preparará su mente para futuros aprendizajes en las matemáticas. Presenta 10 niveles, todos coloridos y variados que están basados en los ejercicios del Cuaderno de Trabajo de Matemática Primer Grado. Contar es algo que a los niños les encanta, pero muchas veces les cuesta memorizar. Por eso, hacerlo mediante este dinámico juego resultará completamente estimulante para ellos. A partir de conjuntos con elementos mixtos, los pequeños deberán comprender primeramente cuántos hay del mismo tipo, y posteriormente reconocer el número asociado a esa cantidad. Mejorará su memoria y desarrollará su pensamiento asociativo ( *Identifica y cuenta*).

Luego de realizar un estudio sobre los sistemas educativos en el mundo y en Cuba, se pudo sacar las siguientes características:

Tabla 1.1 Análisis de los antecedentes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sistemas similares | Autenticación | Nivel de  dificultad | Multiplataforma | Contiene operaciones básicas de la matemática | Retroalimentación | Gratis |
| Cokitos |  | X | X | X | X | X |
| Identifica y cuenta |  |  |  |  | X | X |
| IXL | X | X | X | X | X |  |
| [Tux of Math Command](https://tux4kids.alioth.debian.org/tuxmath/download.php) |  |  | X | X |  |  |

**Según el análisis realizado en el estado del arte**, no se ha encontrado un sistema que aborde completamente los requisitos necesarios para el desarrollo del sistema propuesto en esta investigación, aunque cabe destacar que hay sistemas muy similares, por ejemplo, Cokitos, pero este no tiene la opción de autenticación para una mayor seguridad del sistema y IXL que en su caso no es gratuito. En otras palabras, las soluciones existentes carecen de funcionalidades esenciales para satisfacer plenamente los objetivos de este nuevo sistema. El estudio identifica una brecha en las capacidades actuales y sugiere que aún hay espacio para innovación y mejora en el diseño y desarrollo del sistema en cuestión.

## 1.4 Metodología de desarrollo de software

Las empresas con el fin de mejorar la calidad del software que producen y reducir los costos asociados con el desarrollo del software han utilizado un grupo de metodologías de desarrollo de software las cuales se pueden definir como un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a la construcción de un software. (Pressman, 2010).

El uso de metodologías de desarrollo de software conlleva importantes beneficios.  Estas metodologías simplifican el proceso de desarrollo, lo que facilita que los equipos comprendan y apliquen las prácticas adecuadas.  Al seguir una metodología, se da prioridad a las necesidades y expectativas del cliente, esto garantiza que el producto final cumpla con sus requerimientos. También proporcionan un marco estructurado para planificar, ejecutar y controlar todo el ciclo de desarrollo, esto incluye desde la concepción hasta la entrega del producto y se logra un producto más sólido y bien diseñado. Esto se traduce en una experiencia de usuario mejorada.

Estas metodologías se han dividido en dos grandes grupos los cuales proponen esquemas de trabajo diferente.

**Metodologías de desarrollo de software tradicionales**:

Las metodologías de desarrollo de software tradicionales se caracterizan por definir total y rígidamente los requisitos al inicio de los proyectos de ingeniería de software. Los ciclos de desarrollo son poco flexibles y no permiten realizar cambios.Dentro de estas metodologías se encuentran: Rational Unified Process (RUP por sus siglas en inglés), Software Capability Maturity Model (SW –CMM por sus siglas en inglés), Microsoft Solution Framework (MSF por sus siglas en inglés) (Universidades, 2020).

**Metodologías de desarrollo de software ágiles**:

En la actualidad, las metodologías ágiles de desarrollo de software son ampliamente utilizadas. Su prioridad más alta es satisfacer al cliente a través de la entrega pronta y continua de software valioso. Son bienvenidos los requerimientos cambiantes, aun en una etapa avanzada del desarrollo. Los clientes y los desarrolladores deben trabajar juntos, a diario y durante todo el proyecto. Dentro de estas metodologías se encuentran: Extreme Programing (XP por sus siglas en inglés), SCRUM y AUP versión UCI (Pressman, 2010).

 En el contexto del proyecto en cuestión, la elección de una metodología ágilapropiada es crucial para el éxito teniendo en cuenta que el equipo de trabajo está compuesto por 2 personas, es un proyecto de poca duración y se necesita comunicación continua con el cliente. Se han evaluado tres opciones ampliamente utilizadas: Extreme Programming (XP), SCRUM y AUP versión UCI. Cada una de estas metodologías tiene características únicas que las hacen adecuadas para diferentes tipos de proyectos.

**SCRUM**

Tamaño del equipo de desarrollo: Puede funcionar bien en equipos de diversos tamaños, pero suele requerir roles específicos como Scrum Master y Product Owner.

Comunicación continua con el cliente: Enfatiza la colaboración con el cliente a través de reuniones regulares, pero no siempre implica la participación diaria del cliente en el desarrollo.

Desarrollo: Apoya el desarrollo iterativo, pero no se centra tanto en prácticas de programación extrema o pruebas unitarias continuas.

Enfoque en pruebas: Si bien promueve las pruebas, no pone tanto énfasis en las pruebas unitarias continuas como XP (Pressman, 2010).

**AUP-UCI**

Tamaño del equipo de desarrollo: Puede adaptarse a equipos de diferentes tamaños.

Comunicación continua con el cliente: No es necesario una comunicación continua con el cliente ya que solo necesita una orientación o información de vez en cuando para saber que se quiere hacer

Desarrollo: Proporciona un marco de desarrollo de software iterativo e incremental.

Enfoque en pruebas: Incluye pruebas internas, pruebas de liberación y pruebas de aceptación.

**Extreme Programming (XP)**

Tamaño del equipo de desarrollo: Se presenta como la opción ideal para equipos pequeños debido a su enfoque en la colaboración estrecha y la toma de decisiones eficiente en equipos de menor tamaño, minimizando la burocracia.

Comunicación continua con el cliente: Fomenta una comunicación continua y directa con el cliente, incluyendo al cliente en actividades diarias de desarrollo

Desarrollo: Aboga por el desarrollo iterativo e incremental, entregando funcionalidades operativas en cortos períodos de tiempo y facilitando la identificación temprana de ajustes o mejoras.

Enfoque en pruebas: Incorpora pruebas unitarias continuas como una práctica central, mejorando así la calidad del código y permitiendo cambios seguros (Pressman, 2010).

Considerando los elementos planteados se puede concluir que **Extreme Programming (XP)** es la opción más idónea para el proyecto en cuestión. Esto se debe a su flexibilidad para adaptarse a equipos reducidos, su enfoque en la comunicación constante con el cliente, la metodología de desarrollo iterativo e incremental, y su énfasis en pruebas unitarias continuas. Estas características se ajustan de manera efectiva a las necesidades y requisitos del proyecto, lo que lo convierte en una elección sólida.

## 1.4.1 Programación Extrema

La metodología de [desarrollo de software](https://softwarepara.es/software-development/) XP (**Extreme Programming**) es una metodología ágil que se enfoca en la calidad del [software](https://softwarepara.es/blank-down-en-el-software-de-limpieza-de-datos/) y en la satisfacción del cliente al brindar un proceso de desarrollo iterativo e incremental.

Entre los valores principales de XP se encuentran: comunicación, simplicidad, retroalimentación constante, valentía y respeto (Pressman, 2010).

Además, XP promueve la colaboración entre los miembros del equipo y el cliente para lograr un resultado final de calidad y cumplir con los requisitos del usuario. También prioriza la entrega constante de software funcional, lo que permite obtener feedback temprano sobre el producto y mejorar los siguientes entregables. Debido a que los proyectos XP son completamente diferentes, es lógico que gestionarlos también sea diferente. Las recetas de Beck son a menudo irónicas, a veces sabias y casi siempre dan en el blanco (Beck & Fowler, 2001).

La metodología XP consta de varias fases que son:

**Planificación:** se establecen los objetivos y requisitos del proyecto, se determina el alcance y se planifican las iteraciones a realizar.

**Diseño:** se diseña la arquitectura del sistema y se definen las especificaciones técnicas para su desarrollo.

**Desarrollo:** se desarrollan iterativamente las funcionalidades del sistema siguiendo las especificaciones técnicas definidas previamente.

**Pruebas:** se realizan pruebas unitarias y de integración para asegurar la calidad del código desarrollado.

Esta metodología cuenta con determinados artefactos para guiar el desarrollo del proyecto.

Estos artefactos son las Historias de Usuario (HU), la estimación de esfuerzos y las tarjetas Clase-Responsabilidad-Colaboradores (CRC).

Las Historias de Usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer. El tratamiento de las historias de usuario es muy dinámico y flexible (Letelier, 2012).

La estimación de esfuerzo es un proceso mediante el cual los desarrolladores calculan el tiempo necesario para llevar a cabo la implementación de cada **historia de usuario**. Esta estimación es crucial durante la fase de planificación del proyecto.

Las tarjetas CRC es un conjunto de tarjetas índice estándar que representan clases. Las tarjetas se dividen en tres secciones. En la parte superior de la tarjeta se escribe el nombre de la clase, en la parte izquierda del cuerpo se enlistan las responsabilidades de la clase y en la derecha, los colaboradores (Pressman, 2010).

La metodología XP presenta varias razones por las cuales es una buena opción en el desarrollo de software.

1. Se enfoca en la comunicación constante entre los miembros del equipo de trabajo y los clientes, a través de reuniones regulares, pruebas y entregas frecuentes, lo que permite una mayor claridad en los objetivos y requisitos del proyecto.
2. Promueve la simplicidad en el proceso de desarrollo, evitando la inclusión de funciones innecesarias y priorizando las tareas más importantes, lo cual se traduce en una mejor eficiencia y adaptabilidad del proyecto.
3. Flexibilidad en cuanto a cambios durante el proceso de desarrollo, lo que permite adaptarse a las necesidades que puedan surgir a lo largo del camino.
4. Se enfoca en la calidad del código, promoviendo prácticas como la programación en parejas y las pruebas automatizadas, lo cual resulta en un software más fiable y fácil de mantener en el tiempo (Pressman, 2010).

## 1.5 Herramientas y lenguajes informáticos

En cualquier proyecto de desarrollo de software seleccionar las herramientas y tecnologías adecuadas es un paso crucial. Estas elecciones moldean la naturaleza de la solución y sientan las bases de su calidad. Luego de un estudio de las tecnologías y en acuerdo con el cliente se seleccionaron las siguientes:

**Microsoft Visual Studio Code**

IDE de Visual Studio es una plataforma de lanzamiento creativa que puede utilizar para editar, depurar y compilar código y, finalmente, publicar una aplicación. Además del editor y depurador estándar que ofrecen la mayoría de IDE, Visual Studio incluye compiladores, herramientas de completado de código, diseñadores gráficos y muchas más funciones para mejorar el proceso de desarrollo de software v.1.82 (*Visual Studio*, 2023).

**Visual Paradigm**

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. Proporciona a los desarrolladores de software la plataforma de desarrollo de vanguardia para construir aplicaciones de calidad más rápidas, mejores y más baratas. Facilita una excelente interoperabilidad con otras herramientas CASE y la mayoría de los IDEs líderes que sobresalen todo su proceso de desarrollo de Model-Code-Deploy en esta solución única v.16.2 (*Visual Paradigm, 2024*).

**PHPMyAdmin**

PHPMyAdmin es una herramienta de software gratuita escrita en PHP que está destinada a manejar la administración de un servidor de base de datos MySQL. Puede utilizar PHPMyAdmin para realizar la mayoría de las tareas administrativas, incluida la creación de una base de datos, ejecutar consultas y agregar cuentas de usuario (*documentación de PHPMyAdmin*, 2021).

**Lenguaje de modelado**

UML (Unified Modeling Language) es un lenguaje estándar para especificar, visualizar, construir y documentar los artefactos de los sistemas de software. En general, los diagramas UML describen los limites, la estructura y el comportamiento del sistema y los objetos que contiene. No es un lenguaje de programación, sino que se pueden utilizar herramientas para generar código en varios idiomas utilizando diagramas UML. Este tiene una relación directa con el análisis y el diseño orientados a objetos (*UML*).

**Lenguajes del lado del servidor**

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Estos son empleados para el diseño de páginas webs dinámicas donde los más utilizados para el desarrollo son: ASP, PHP y PERL.

El lenguaje de Hypertext Pre-processor (Pre-Procesador de Hipertextos) PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto. Esto significa que cualquiera puede hacer cambios en su estructura. En la práctica, esto representa dos cosas importantes: no hay restricciones de uso vinculadas a los derechos. El usuario puede usar PHP para programar en cualquier proyecto y comercializarlo sin problemas. v.8.2 (*PHP*, 2024).

**Lenguajes del lado del cliente.**

Los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre tratamiento.

**HTML**

El lenguaje HyperText Markup Language (Lenguaje de Marcas de Hipertexto) HTML es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc... Es muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida, pueda enfrentarse a la tarea de crear una web. v.5 (*HTML Tutorial*).

**CSS**

Las hojas de estilo en cascada o (Cascading Style Sheets) CSS son un lenguaje formal usado para definir la presentación de un documento estructurado, escrito en HTML o XML. Se le denomina Hojas de Estilos en Cascada porque las características se aplican de arriba a abajo mediante reglas que poseen un esquema prioritario. Esta especificación es un lenguaje de diseño gráfico que se escribe dentro del código HTML del sitio web y, permite crear páginas de una manera más exacta y aplicarles estilos (colores, márgenes, formas, tipos de letras, etc.) por lo que se tiene mayor control de los resultados finales. v.3 (*CSS Tutorial*).

**JAVASCRIPT**

Es un lenguaje de programación que se puede aplicar a un documento HTML y usarse para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, cofundador del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla. Por sí solo es bastante compacto, aunque muy flexible, y los desarrolladores han escrito gran cantidad de herramientas encima del núcleo del lenguaje JavaScript, desbloqueando una gran cantidad de funcionalidad adicional con un mínimo esfuerzo. No requiere compilación, y sirve principalmente para mejorar la gestión de la interfaz cliente/servidor como, por ejemplo, un script JavaScript insertado en un documento HTML permite reconocer y tratar localmente los eventos generados por el usuario (*JavaScript Tutorial*).

**Sistemas Gestor de Base de Datos (SGBD)**

**MySQL**

Es un sistema de administración de bases de datos relacionales de código abierto desarrollado por Oracle. Se considera como la base de datos de código abierto más utilizada en el mundo. Soporta una amplia gama de tipos de datos, lo que permite tener una gran versatilidad en cuanto a las situaciones, industrias o casos de uso donde puede implementarse una base de datos MySQL. Puede emplearse para la industria financiera, al manejar datos con mucha precisión; por otro lado, también puede utilizarse en ámbitos de geolocalización por sus datos de tipo espacial. v.5 (*MySQL Tutorial*).

**Framework**

Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software. Utilizar frameworks puede simplificar (y mucho) una tarea o proceso, de ahí que se trate de una de las herramientas habituales que manejan los Digital Workers, porque les ayuda a ser más ágiles y productivos. Generalmente, los frameworks son usados por programadores porque permiten acelerar el trabajo y favorecer que este sea colaborativo, reducir errores y obtener un resultado de más calidad.

**Laravel**

Laravel es un framework PHP gratis y de código abierto que brinda un conjunto de herramientas y recursos para crear aplicaciones modernas. Posee un ecosistema integral que combina funciones integradas y una variedad de paquetes y extensiones compatibles. Este framework de PHP creció en popularidad rápidamente en los últimos años, y muchos desarrolladores lo adoptaron como su framework de trabajo favorito para lograr un proceso de desarrollo optimizado. Para entender mejor qué es Laravel, citaremos una frase de su página web: “Laravel es un marco de aplicación web con una sintaxis expresiva y elegante. Ya sentamos las bases, liberándote para crear sin preocuparte por las cosas pequeñas”. v.10 (*Laravel*, 2024).

**Características**

* Motor de plantillas ligeras e integradas, que permiten crear diseños increíbles utilizando siempre contenido dinámico. Además, proporciona multitud de widgets que incorporan CSS, y JS con estructuras súper robustas. Las plantillas de Laravel están muy bien diseñadas y el resultado es a la vez simple y complejo.
* Compatible con arquitectura MVC: Laravel es un framework compatible con patrones de arquitectura MVC. Esto garantiza la separación de la lógica empresarial por un lado y las capas de presentación por el otro. Este sistema es fácilmente escalable y ofrece multitud de funciones ya integradas.
* Una de las características que más apreciamos los desarrolladores es la incorporación de una herramienta de línea de comandos (llamada Artisan) que ayuda a automatizar las tediosas tareas de programación repetitivas.
* Laravel ofrece sus propias bibliotecas modulares de manera preinstalada en el sistema. Esto no es algo que suela ocurrir en otros framework de PHP.

Está pensado para resolver problemas concretos que son de mucha actualidad. Esto es una gran ventaja. Digamos que se centra en lo importante.

Se decidió usar el framework Laravel ya que incluye herramientas que facilitan la construcción de aplicaciones web, haciendo de este proceso algo mucho más rápido y dando como resultado un código bien estructurado y fácil de mantener. Hoy en día, existen muchísimos sitios web creados con esta tecnología, incluyendo grandes empresas como Disney, Twitch, The New York Times, entre otros.

## 1.6 Conclusiones del capítulo

En este capítulo, se exploraron los conceptos fundamentales relacionados con el tema, lo que contribuyó a una comprensión más profunda del trabajo actual. Al analizar los sistemas homólogos existentes, se llegó a la conclusión de que ninguno de ellos ofrece una solución adecuada para la presente investigación. Sin embargo, estos sistemas proporcionaron elementos valiosos que serán útiles en el desarrollo del sistema propuesto. La metodología de desarrollo de software elegida fue **Extreme Programming (XP),** que enfatiza las relaciones interpersonales como clave para el éxito en el desarrollo de software. XP promueve el trabajo en equipo, se preocupa por el aprendizaje continuo de los desarrolladores y fomenta un ambiente de trabajo positivo. Además, las herramientas y tecnologías utilizadas mejoraron el conocimiento y la gestión durante la realización de esta investigación.

# **CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA**

## 2.1 Introducción

En este capítulo se describe las funcionalidades principales de la solución que se propone. Se mostrarán los productos obtenidos al aplicar la metodología XP de desarrollo de software en sus etapas de planificación y diseño.

## 2.2 Planificación

Para realizar un buen trabajo es necesario planificarse bien. La dinámica de planificación llevada a cabo al inicio de la iteración, suele ser la siguiente: El usuario presenta la lista de funcionalidades deseadas para el sistema, escrita con formato de Historia de Usuario, en la cual se encuentra definido el comportamiento de la misma con sus respectivos criterios de aceptación.

## 2.2.1 Historias de Usuario

Las historias de usuario es uno de los artefactos principales en las fases de la metodología de desarrollo de software XP. Estas historias que son redactadas por los propios usuarios, representan una estrategia eficiente para gestionar los requisitos de los usuarios. Esta estrategia elimina la necesidad de producir una gran cantidad de documentación formal y reduce el tiempo necesario para su gestión. Además, las historias de usuario ofrecen la flexibilidad necesaria para adaptarse rápidamente a los cambios en los requisitos.

Para identificar cuáles Historias de Usuario (HU) son más críticas para resolver, de acuerdo con las necesidades del cliente y para poder planificar adecuadamente su implementación, el dueño del producto clasifica cada HU según su importancia para el negocio en:

* **Alta:** HU que resultan de vital importancia en el desarrollo del sistema.
* **Media**: HU que son necesarias, pero no absolutamente esenciales.
* **Baja:** HU que no tienen nada que ver con el sistema en desarrollo, pero a su vez constituyen funcionalidades para gestionar elementos relacionados con el equipo de desarrollo.

Las historias de usuario a la hora de ser implementadas pueden ocasionar ciertas dificultades al equipo de desarrollo, por ello según su grado de dificultad se clasifican en:

* **Alto**: Durante la implementación de la HU, se toma en cuenta que pueden surgir errores que podrían resultar en un código inoperante.
* **Medio**: Cuando ocurren errores en el desarrollo de las HU que traen consigo retraso en la entrega del producto.
* **Bajo**: En caso de existir errores el equipo de desarrollo los podrá tratar con relativa facilidad sin que ocasione otros problemas para el desarrollo del proyecto.

Las historias de usurario son tablas las cuales se encuentran conformadas por los siguientes elementos:

* **Número**: Es el número que representa la historia de usuario.
* **Nombre de Historia de Usuario**: Nombre de la HU definida entre los desarrolladores y el cliente.
* **Usuario**: Persona encargada de desarrollar la HU.
* **Prioridad en negocio**: Clasificación de la HU en dependencia de su vitalidad para el negocio. (Alta, Media. Baja)
* **Riesgo en desarrollo**: Clasificación de la ocurrencia de errores en el desarrollo de las HU. (Alto, Medio, Bajo)
* **Puntos Estimados**: Son una forma de calcular la cantidad de esfuerzo necesario para completar una historia de usuario.
* **Iteración asignada**: Número de la iteración donde va a desarrollarse la HU.
* **Programador responsable**: Persona encargada de programar la historia de usuario.
* **Descripción**: Pequeña descripción de la HU.

Tabla 2.1 HU Autenticar Usuario

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: 1 | Nombre: Autenticar usuario |
| Usuario: Todos los usuarios del sistema | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 0.5 | Iteración asignada: 1 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: Cada usuario debe contar con email y password para acceder a la plataforma, además de tener asignado un rol determinado y en función del mismo se le muestran las diferentes opciones a las que tiene permisos de acceder. | |

Tabla 2.2 HU Registrar Usuarios

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: 2 | Nombre: Registrar Usuarios |
| Usuario: Usuarios del sistema | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Medio |
| Puntos estimados: 0.5 | Iteración asignada: 1 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: El sistema debe permitir que los usuarios con rol de estudiantes se registren en la plataforma proporcionando datos básicos como email, password, nombre y grado que cursan. De esta manera los profesores podrán realizar ejercicios para determinados grados y puedan ser identificados los mismos para que los estudiantes con el nivel adecuado puedan acceder a ellos. | |

Tabla 2.3 HU Gestionar Usuarios

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: 3 | Nombre: Gestionar Usuarios |
| Usuario: Administradores del sistema | |
| Prioridad en negocio: Media | Riesgo en desarrollo: medio |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: El sistema debe permitir a los administradores registrar profesores y otros administradores para tener identificado a los usuarios que interactúen en la plataforma, evitando de esta manera que una persona con fines malintencionados se registre con rol de administrador o profesor y pueda tener acceso a todas las funcionalidades que el sistema les provee. | |

Tabla 2.4 HU Gestionar Tareas

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: 4 | Nombre: Gestionar Tareas |
| Usuario: Administradores del sistema | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Medio |
| Puntos estimados: 1 | Iteración asignada: 2 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: El sistema debe permitir a los usuarios con rol de profesor o administrador crear operaciones básicas desde el panel de administración para que posteriormente las mismas les parezcan a los usuarios que tienen el grado suficiente para realizarlas. | |

Tabla 2.5 HU Gestionar Grados

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: 5 | Nombre: Gestionar Grados |
| Usuario: Administradores del sistema | |
| Prioridad en negocio: Media | Riesgo en desarrollo: Medio |
| Puntos estimados: 1 | Iteración asignada: 2 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: El sistema debe permitir a los usuarios con rol de administrador gestionar los grados que estarán disponibles en la plataforma para que los usuarios con rol de estudiante puedan elegir el grado que cursan y los usuarios con rol de profesor asignar las tareas según el grado. | |

Tabla 2.6 HU Generar Reportes

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: 6 | Nombre: Generar Reportes |
| Usuario: Administradores del sistema | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: medio |
| Puntos estimados: 0.5 | Iteración asignada: 2 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: El sistema debe permitir generar reportes para mostrar los resultados de los estudiantes a los profesores, dando facilidad para ver el progreso de los mismos con los diferentes ejercicios. | |

Tabla 2.7 HU Crear Operaciones Matemáticas

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: 7 | Nombre: Crear Operaciones matemáticas |
| Usuario: Administradores del sistema | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: alto |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: El sistema debe permitir crear operaciones matemáticas a los usuarios con rol de estudiante, con el objetivo de que los mismos interactúen con la plataforma y practiquen libremente, eligiendo que tipo de ejercicio realizar y ajustando la dificultad de estos. | |

Tabla 2.8 HU Notificar a familiares

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: 8 | Nombre: Notificar a familiares |
| Usuario: Administradores del sistema | |
| Prioridad en negocio: Baja | Riesgo en desarrollo: bajo |
| Puntos estimados: 0.5 | Iteración asignada: 2 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: El sistema debe permitir a los usuarios con rol de profesor notificar a los familiares del estudiante el progreso de estos en la plataforma, brindando información detallada sobre cantidad de ejercicios resueltos, cuantos acertados, cuantos denegados, entre otros datos. | |

## 2.2.2 Requisitos no funcionales

**RNF 1. Requisitos de Usabilidad**

El sistema debe presentar un acceso fácil e interfaz amigable con informaciones fáciles de comprender, para facilitar el uso del mismo.

**RNF 2. Requisitos de Rendimiento**

El sistema debe ser rápido y eficiente, con tiempos de carga mínimos y respuestas rápidas a las acciones del usuario. Los resultados de las operaciones matemáticas deben realizarse de manera ágil.

**RNF 3. Requisitos de Portabilidad**

El sistema debe ser multiplataforma y permitir que se acceda desde cualquier navegador. La interfaz debe adaptarse a cualquier resolución de pantalla para garantizar una amplia accesibilidad.

**RNF 4. Requisitos de Seguridad**

La autenticación de usuarios debe ser segura y proteger la información confidencial. Validación de credenciales en el lado del cliente y del servidor para evitar posibles ataques al sistema.

## 2.2.3 Estimación del esfuerzo

La estimación del esfuerzo es una actividad que consiste en determinar el tiempo y los recursos necesarios para implementar una historia de usuario. La estimación del esfuerzo se suele expresar en puntos o semanas, dependiendo de la complejidad y el tamaño de la historia de usuario. Esta ayuda a planificar las entregas del software y a medir el progreso del proyecto. Se definieron 8 HU en 2 iteraciones. La determinación del número de iteraciones se basó en dos indicadores fundamentales: la experiencia del equipo de desarrollo y la flexibilidad permitida por la planificación orientada al alcance del sistema (total de puntos/ritmo del proyecto).

Tabla 2.9 Estimación del esfuerzo

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Historias de usuario | | Puntos estimados (semanas) |
| 1 | 1 | Autenticar Usuarios | 0.5 |
| 2 | Registrar Usuarios | 0.5 |
| 3 | Gestionar Usuarios | 1 |
| 4 | Crear Operaciones Matemáticas | 1 |
| 2 | 5 | Gestionar Tareas | 1 |
| 6 | Gestionar Grados | 1 |
| 7 | Generar Reporte | 0.5 |
| 8 | Notificar a familiares | 0.5 |
| **Total** | |  | **6** |

## 2.2.4 Plan de Iteraciones

Una vez que se identifican y describen las Historias de Usuario (HU) y se estima el esfuerzo requerido para su realización, se genera el plan de iteraciones. Este plan especifica las HU que se implementarán en cada ciclo de trabajo, con el fin de mejorar la eficiencia del equipo de desarrollo. Al finalizar la última iteración, el sistema está listo para ser lanzado y puesto en marcha.

**Iteración 1**

En esta iteración se implementarán las HU de nivel de prioridad alta las cuales son de vital importancia en el funcionamiento del negocio. Se obtendrá una primera versión del producto donde el cliente podrá probar las funcionalidades.

**Iteración 2**

En esta iteración se completarán las Historias de Usuario (HU) pendientes que son de alta prioridad para el negocio y se corregirán los errores o desacuerdos de los usuarios con los componentes implementados en la iteración anterior. Se obtendrá la versión 1.0 del sistema informático para el desarrollo de habilidades matemáticas esenciales. Los usuarios podrán ejecutar todas las funcionalidades del sistema.

Tabla 2.10 Plan de Iteraciones

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Historias de usuario | | Duración (semanas) |
| 1 | 1 | Autenticar usuarios | 3 |
| 2 | Registrar Usuarios |
| 3 | Gestionar Usuarios |
| 4 | Crear operaciones matemáticas |
| 2 | 5 | Gestionar Tareas | 3 |
| 6 | Gestionar Grados |
| 7 | Generar Reporte |
| 8 | Notificar a familiares |
| **Total** | |  | **6** |

## 2.2.5 Plan de Entregas

Con el objetivo de hacer partícipe al cliente dentro del proceso de desarrollo, se realiza un plan de entregas con las fechas de entrega de cada iteración para que este realice las pruebas al software entregado. La siguiente tabla muestra el número de iteración con su correspondiente fecha de entrega.

Tabla 2.11 Plan de Entregas

|  |  |  |
| --- | --- | --- |
| **Iteración** | **Duración** | **Fecha de entrega** |
| Iteración 1 | 3 semanas | 22-5-2024 |
| Iteración 2 | 3 semanas | 12-6-2024 |

## 2.3 Diseño

En la metodología XP (Programación Extrema), el diseño es una fase vital en la que se definen normas y patrones para la programación. En esta fase, se pretende crear un código simple, que solo incluya el flujo necesario para que funcione la historia del usuario.

## 2.3.1 Patrón Arquitectónico

Un patrón arquitectónico, también conocido como arquetipo es la conceptualización de una solución genérica y reutilizable, aplicable a un problema de diseño de software en un contexto determinado, satisfaciendo las necesidades del negocio. Los patrones arquitectónicos son la representación de las buenas prácticas y estructuras de diseño probadas, de modo que puedan reutilizarse. El objetivo es reutilizar las experiencias y conocimientos arquitectónicos que han dado buenos resultados en el pasado (Castro Ramírez, 2023).

Para este proyecto se seleccionó el patrón Modelo-Vista-Controlador

**Modelo-Vista-Controlador**

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. A partir del uso de frameworks basados en el patrón MVC se puede lograr una mejor organización del trabajo y mayor especialización de los desarrolladores (González & Romero, 2012).

**Modelo:** Es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas y notificar a las Vistas cuando cambia el Modelo.

**Vista:** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.

**Controlador:** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

## 2.3.2 Patrones de Diseño

Un patrón de diseño describe una estructura de diseño que resuelve un problema particular del diseño dentro de un contexto específico y entre “fuerzas” que afectan la manera en la que se aplica y en la que se utiliza dicho patrón. El objetivo de cada patrón de diseño es proporcionar una descripción que permita a un diseñador determinar si el patrón es aplicable al trabajo en cuestión, si puede volverse a usar (con lo que se ahorra tiempo de diseño) y si sirve como guía para desarrollar un patrón distinto en funciones o estructura (Pressman, 2010).

**Patrones GRASP**

Los patrones GRASP constituyen un apoyo en la enseñanza que nos ayuda a entender el diseño de objetos esencial. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades (*Larman y Applying*, 2004). A continuación, se presentan los patrones GRASP empleados en el diseño del sistema:

**Patrón Experto:**

Se usa más que cualquier otro al asignar responsabilidades. Consiste en la asignación de una responsabilidad a la clase que cuenta con la información necesaria para llevarla a cabo.

Se ve reflejado en la clase User.

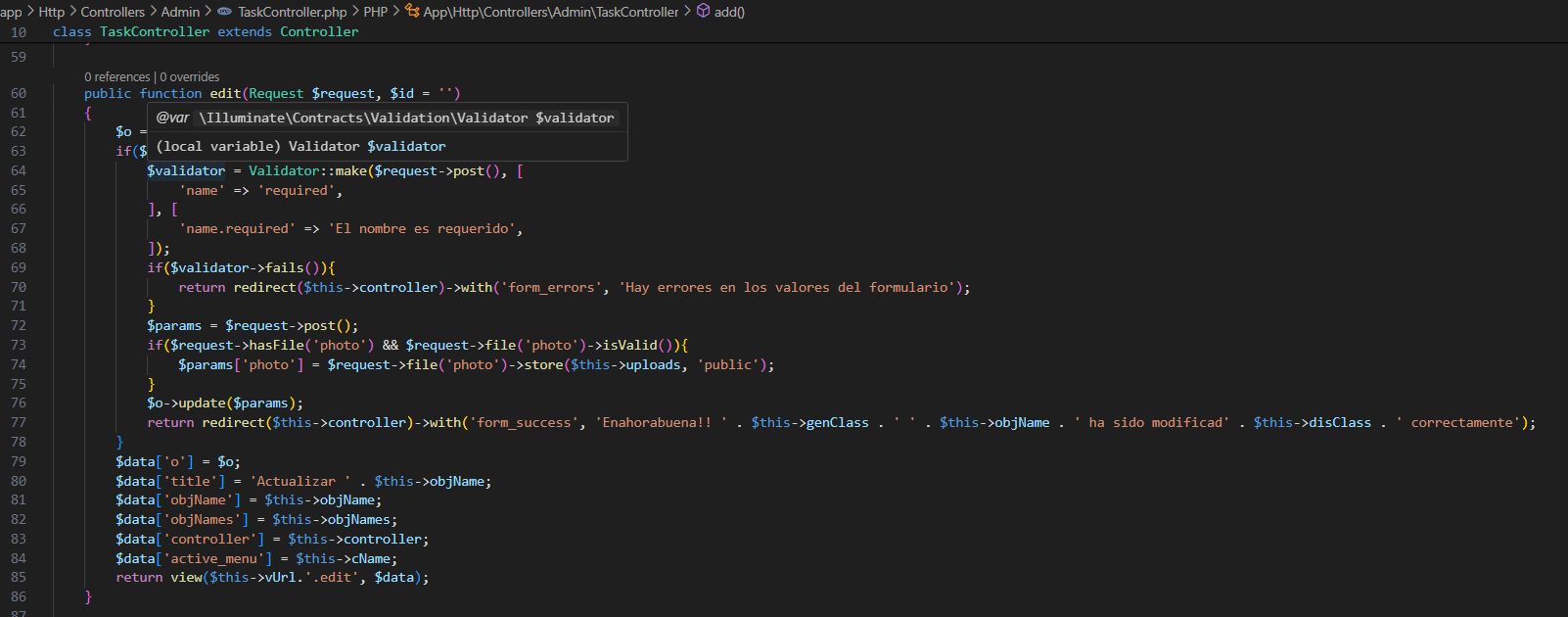
*Figura 1: Evidencia del Patrón Experto*



**Patrón Controlador:**

Este patrón va a actuarcomo un puente entre una interfaz específica y el algoritmo que la ejecuta. Va a permitir recoger la información del usuario y distribuirla a las diferentes clases de acuerdo con el método invocado. Se ve reflejado en la clase Task.

*Figura 2: Evidencia del Patrón Controlador*



**Patrón Creador:**

Este patrón se encarga de la asignación de la responsabilidad de crear o instanciar nuevos objetos o clases a la clase más adecuada. Se ve reflejado en la clase Task.

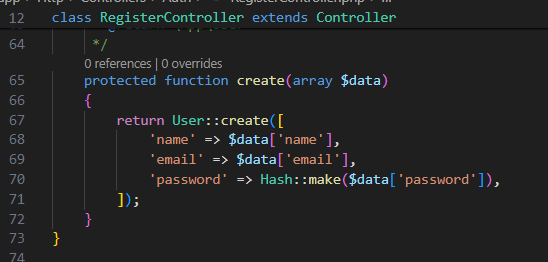
*Figura 3: Evidencia del Patrón Creador*

****

**Patrón Alta Cohesión:**

Cada elemento del diseño debe realizar una labor única dentro del sistema, lo cual expresa que la información que almacena una clase, debe ser coherente y estar lo más relacionada con ella posible. Este patrón es el encargado de asignar responsabilidades, de manera que la información que se almacena en una clase, sea la necesaria y esté bien delimitada.

*Figura 4: Evidencia del Patrón Alta Cohesión*



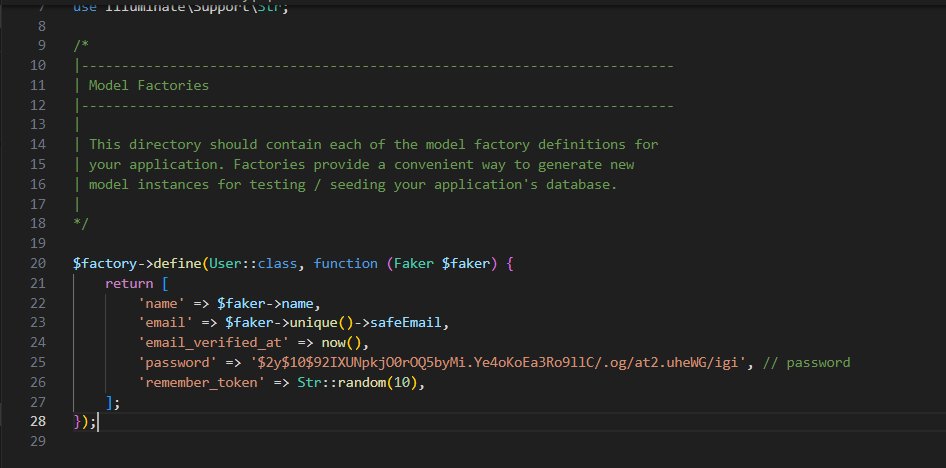
**Patrones GOF**

Los patrones GOF son alternativas de solución a problemas conocidos, pero son mucho más específicas las situaciones en las que se aplican. Los cuales se clasifican en tres grupos fundamentales: los patrones Creacionales que es donde se inicializan y configuran los objetos, los patrones Estructurales donde se separa la interfaz de la implementación y los patrones de Comportamiento los cuales describen el comportamiento entre las clases y los objetos (*Larman y Applying* , 2004).

**Patrón Factory:**

Este patrón ayuda a crear instancias de otras clases, con la finalidad de ocultar la complejidad que se requiere para crearlas. Es una instancia Factory definida a nivel global de aplicación.

*Figura 5: Evidencia del Patrón Factory*



## 2.3.3 Tarjetas CRC

Un modelo CRC en realidad es un conjunto de tarjetas índice estándar que representan clases. Las tarjetas se dividen en tres secciones. En la parte superior de la tarjeta se escribe el nombre de la clase, la parte izquierda del cuerpo se enlistan las responsabilidades de la clase y en la derecha, los colaboradores (Pressman, 2010).

Tabla 2.12 Tarjeta CRC User

|  |  |
| --- | --- |
| **Tarjeta CRC** | |
| **Clase:** User | |
| **Responsabilidad** | **Colaboración** |
| Almacenar información correspondiente a los usuarios de la plataforma | * Familiar * Result * Degree |

Tabla 2.13 Tarjeta CRC Familiar

|  |  |
| --- | --- |
| **Tarjeta CRC** | |
| **Clase:** Familiar | |
| **Responsabilidad** | **Colaboración** |
| Almacenar información correspondiente a los familiares de los usuarios | * User |

Tabla 2.14 Tarjeta CRC Result

|  |  |
| --- | --- |
| **Tarjeta CRC** | |
| **Clase:** Result | |
| **Responsabilidad** | **Colaboración** |
| Almacenar información correspondiente a los resultados de los ejercicios de los alumnos | * User * Task |

Tabla 2.15 Tarjeta CRC Degree

|  |  |
| --- | --- |
| **Tarjeta CRC** | |
| **Clase:** Degree | |
| **Responsabilidad** | **Colaboración** |
| Almacenar información correspondiente a los grados. Encargada de almacenar los datos de los grados que posteriormente son asignados a los usuarios y así poder ver que tarea es más adecuada en dependencia de dicho grado | * Task |

Tabla 2.16 Tarjeta CRC Task

|  |  |
| --- | --- |
| **Tarjeta CRC** | |
| **Clase:** Task | |
| **Responsabilidad** | **Colaboración** |
| Almacenar información correspondiente a las tareas. Se relaciona con grados por que una vez elegido el grado para el cual la tarea es asignada, dicha tarea se les muestra a los usuarios que estén cursando ese grado | * Degree |

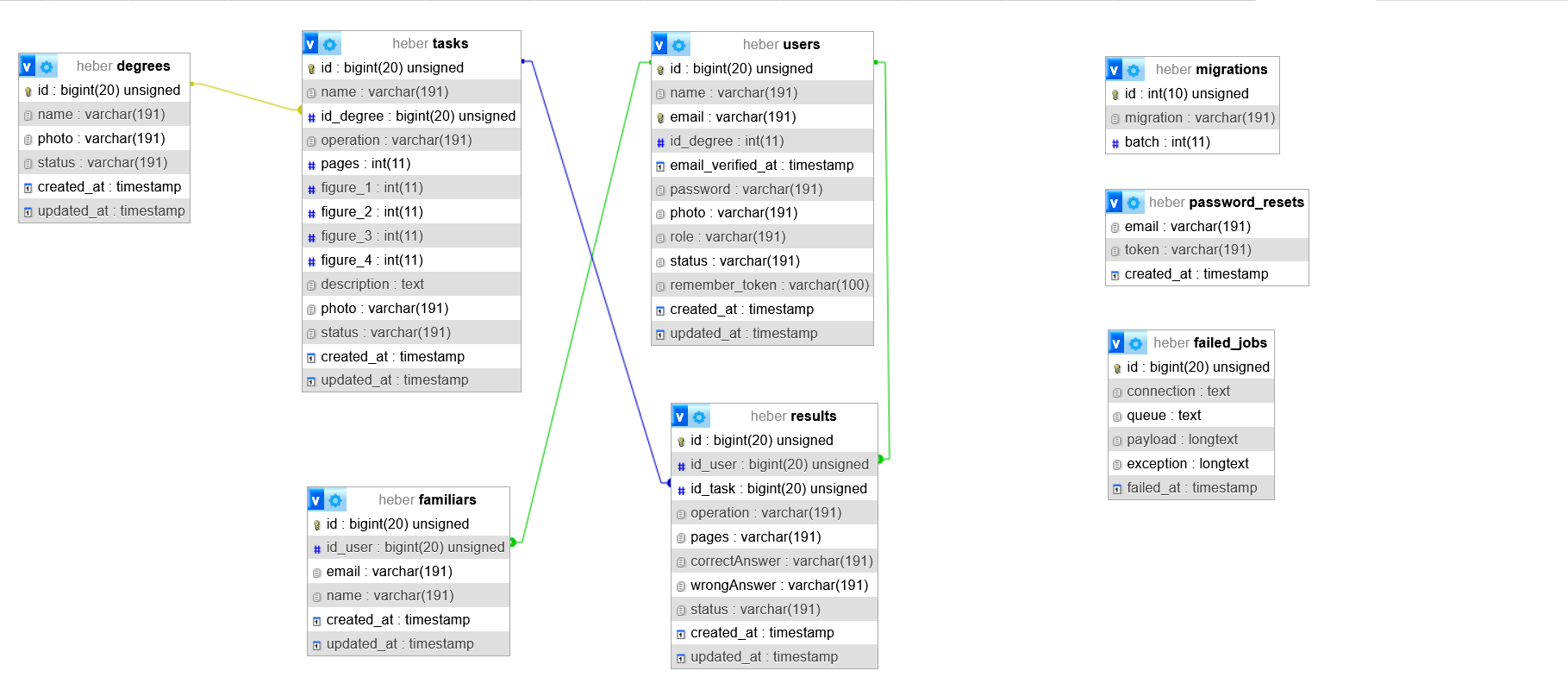
## 2.4 Modelo de datos

Después de la elaboración de las tarjetas CRC y la definición de las clases de entidades, se avanza hacia la construcción del modelo físico de la base de datos que el sistema usará para almacenar la información.

Este modelo de datos tiene la capacidad de ilustrar las clases principales que constituyen la capa del modelo del sistema, el tipo de dato de los atributos que la base de datos albergará y cómo se interconectan las entidades. Asimismo, permite visualizar los elementos de la realidad que son pertinentes para un problema determinado y cómo estos elementos se interrelacionan entre sí.

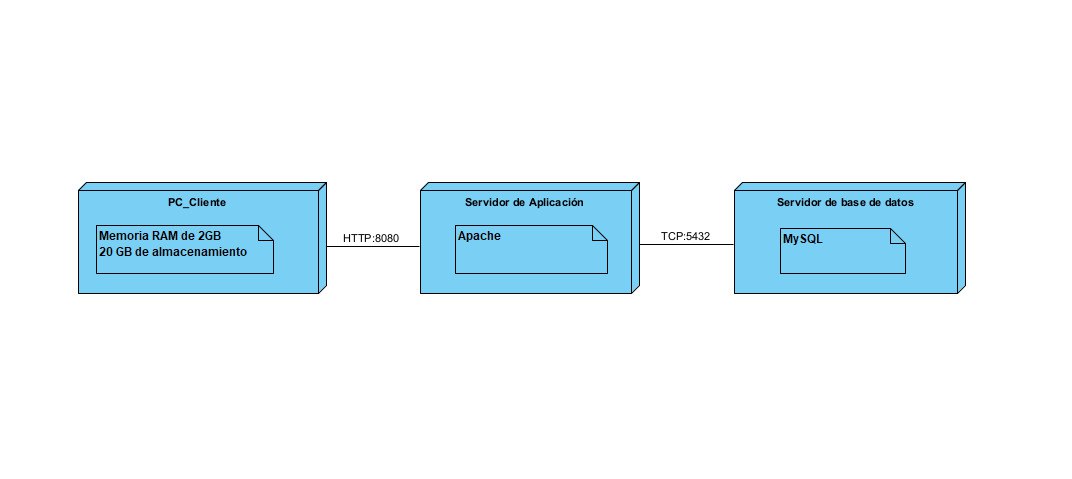
Seguidamente, se muestra el diagrama de entidad-relación que simboliza el modelo físico de datos diseñado para el sistema.

*Figura 6: Modelo de Datos*



## 2.5. Diagrama de despliegue

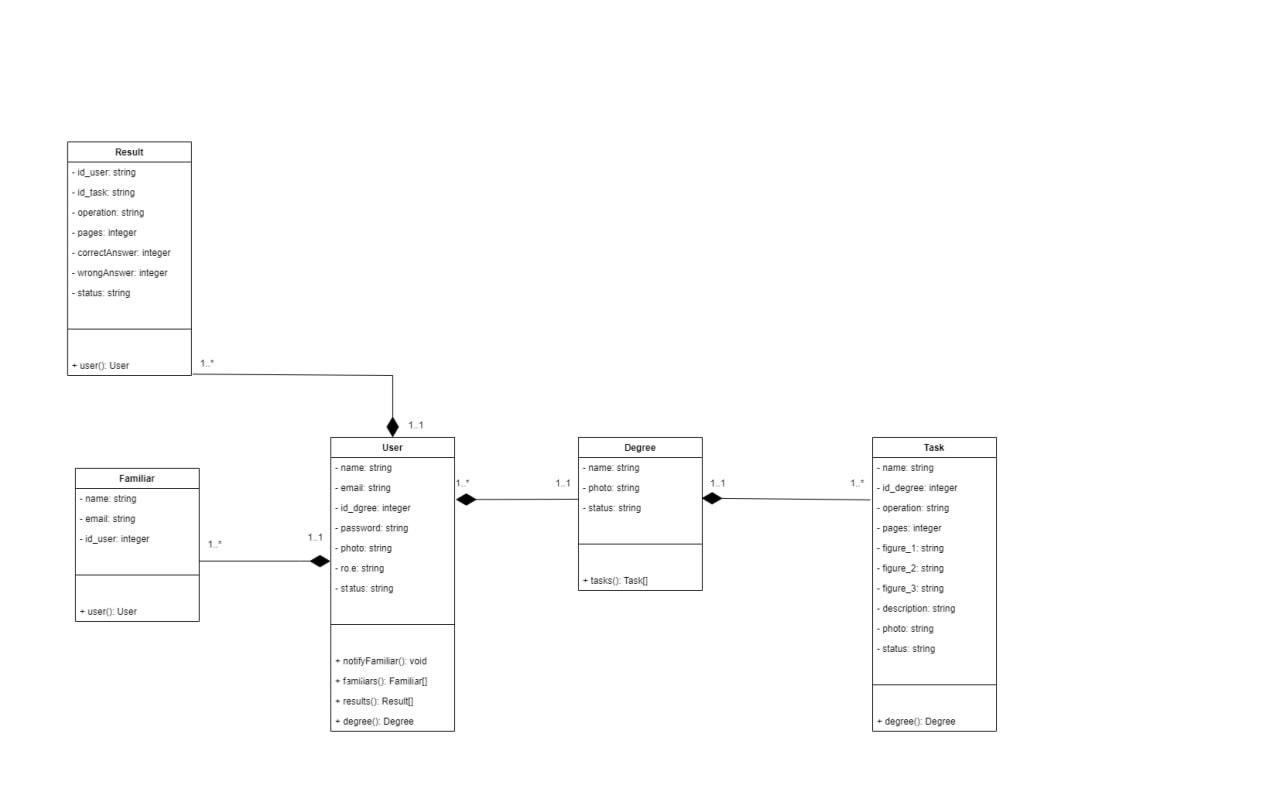
Para visualizar la infraestructura física en la que se ejecuta el sistema se realiza el diagrama de despliegue.

*Figura 7: Diagrama de despliegue*

## 2.6. Diagrama de clases

Para mostrar las clases del sistema, sus atributos, métodos y las relaciones entre ellos se realizan el diagrama de clases.

*Figura 8: Diagrama de clases*



## 2.7. Conclusiones del capitulo

Durante la etapa de planificación, se definieron las Historias de Usuario (HU) que caracterizan al sistema. Posteriormente, se realizó una estimación del esfuerzo necesario para implementar dichas HU. Además, se desarrolló un plan de iteraciones que establece las HU a implementar en cada iteración y su prioridad. Se acordó con los propietarios del software un cronograma de entregas que incluye la liberación de versiones del sistema después de cada iteración y las fechas de entrega. Finalmente, en la etapa de diseño, se establecieron los elementos que conforman la estructura del sistema. Se definió el patrón arquitectónico a emplear y los patrones de diseño los cuales van a permitir un mejor manejo de habilidades al equipo de desarrollo. Se utilizaron las tarjetas CRC para definir las clases, establecer las relaciones entre ellas a través de sus colaboradores y asignar sus responsabilidades. Esto contribuye a reducir el acoplamiento y promover la reutilización en la herramienta.

# **CAPÍTULO 3: Implementación y Pruebas**

## 3.1 Introducción

Tras la finalización de la fase de diseño, donde se establecen los fundamentos estructurales del sistema, se inicia la etapa de implementación y pruebas. En esta etapa, se efectúa una partición de las Historias de Usuario en Tareas de Implementación (TI) con el propósito de simplificar su comprensión y ejecución. A continuación, se procede a codificar las pruebas unitarias, implementar las mencionadas Tareas de Implementación y ejecutar las pruebas de aceptación. La fase de implementación y pruebas se caracteriza por ser el periodo en el cual se obtienen los logros más concretos y relevantes, concluyendo con un sistema íntegro y validado por el cliente.

El capítulo en cuestión aborda una serie de temas, incluyendo los lineamientos que supervisan el código producido, el proceso de iteraciones llevado a cabo para poner en práctica las Historias de Usuario (HU) y las pruebas que se han aplicado, así como los resultados obtenidos en el proceso.

## 3.2 Estándares de codificación

Los estándares de codificación son reglas, técnicas y directrices para crear un código más limpio, legible y eficiente, con un mínimo de fallos y errores. Además, ofrecen a los desarrolladores una forma uniforme de construir código altamente funcional. El cumplimiento de estos estándares reduce los problemas de seguridad y rendimiento que podrían derivarse de unas malas prácticas de codificación**,** ayuda a garantizar la calidad del código, facilitando su lectura, análisis y trabajo, el código también resulta más fácil de mantener y ampliar, incluso para los nuevos desarrolladores**,** tambiénreducen la complejidad del código y aportan soluciones de diseño más elegantes (Vergara Carmona, 2023).

A continuación, se muestran pautas y buenas prácticas para escribir código en Laravel

Laravel sigue los estándares PSR-1 y PSR-4. Y además tiene algunas recomendaciones propias. Lo que en algunos entornos llaman el ‘*Laravel «flavor» of PSR-2*‘. Estos son los estándares extraídos de su guía de colaboración:

* La declaración del namespace debe estar en la misma línea que **<?** php
* La llave de apertura de las clases deben ir en la misma línea que el nombre de la clase.
* Las funciones y estructuras de control deben seguir el estilo de llaves Allman. El estilo Allman define que la llave de apertura de las estructuras de control debe ir en la línea siguiente. La llave de cierre debe estar al mismo nivel que la de apertura. Y el cuerpo de la estructura debe estar indentado.
* Usa siempre el tag de apertura largo de PHP <?php. No uses el tag corto <?.
* No dejes espacios antes del tag de apertura.
* Declara las propiedades de las clases antes de los métodos
* Declara los métodos en este orden: public, protected y private.

## 3.3 Desarrollo por iteraciones

A continuación, se ejecutan las iteraciones planificadas. En el transcurso de estas iteraciones, se procede a la ejecución de las historias de usuario que se han seleccionado para cada una de ellas. Al comienzo de cada iteración, se lleva a cabo una revisión exhaustiva del plan de iteraciones, y se realiza cualquier ajuste que pueda ser requerido. Dentro de este plan, se desglosan las Historias de Usuario en Tareas de Implementación (TI) o tareas de desarrollo. Estas tareas están diseñadas para ser utilizadas por los programadores y, por lo tanto, pueden redactarse en un lenguaje técnico que no necesariamente debe ser comprensible para el cliente.

Antes de avanzar con la implementación de las tareas, se procede a diseñar pruebas unitarias específicas para cada una de ellas. Estas pruebas tienen como finalidad evaluar la funcionalidad de manera exhaustiva y detectar cualquier fallo en el proceso. Una vez que se completa una iteración, se procede a entregar una versión del producto al propietario del software, seguida de la ejecución de pruebas de aceptación con el objetivo de verificar la satisfacción del cliente con el sistema.

Siguiendo el plan previamente establecido, se han llevado a cabo dos iteraciones de desarrollo en el sistema, y al finalizar este proceso, se ha obtenido un producto completamente preparado para su implementación. A continuación, se detallan las particularidades de cada una de estas iteraciones.

## 3.3.1 Iteración 1

En esta iteración se abordaron las HU de nivel de prioridad alta las cuales son de vital importancia en el funcionamiento del negocio. Será la base de la estructura del sistema y luego de concluidas estas HU, el cliente podrá probar las funcionalidades implementadas y brindar su criterio. Se implementaron las HU: Autenticar usuario, Registrar usuario, Gestionar usuarios y Crear operaciones matemáticas. Para su implementación las mismas fueron divididas en Tareas de Ingeniería (TI).

**HU1: Autenticar Usuario**

Table 3.1 Tarea de desarrollo Autenticar Usuario

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 1 | Número de Historia de usuario: 1 |
| Nombre de la tarea: Autenticar usuario | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Fecha de inicio: 21/4/2024 | Fecha de fin: 24/4/2024 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: Para llevar a cabo la funcionalidad de autenticar usuarios se hizo uso del paquete laravel/ui el cual provee las funcionalidades y rutas pertinentes para hacer el proceso más ágil. Para darle estilos a las vistas se publicaron desde la carpeta vendor las que vienen por defecto con este paquete. | |

**HU2: Registrar Usuario**

Table 3.2 Tarea de desarrollo Registrar Usuario

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 2 | Número de Historia de usuario: 2 |
| Nombre de la tarea: Registrar Usuario | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Fecha de inicio: 26/4/2024 | Fecha de fin: 29/4/2024 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: Para llevar a cabo la funcionalidad de registrar usuarios se hizo uso del paquete laravel/ui el cual provee las funcionalidades y rutas pertinentes para hacer el proceso más ágil. Para darle estilos a las vistas se publicaron desde la carpeta vendor las que vienen por defecto con este paquete. | |

**HU3: Gestionar Usuarios**

Table 3.3 Tarea de desarrollo Insertar Usuarios

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 3 | Número de Historia de usuario: 3 |
| Nombre de la tarea: Insertar Usuarios | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 1/5/2024 | Fecha de fin: 3/5/2024 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: Para llevar a cabo la funcionalidad de crear usuarios fueron creados los ficheros UserController, User como nombre del modelo y dentro de las vistas de users la vista créate.blade.php. | |

Table 3.4 Tarea de desarrollo Eliminar Usuarios

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 4 | Número de Historia de usuario: 3 |
| Nombre de la tarea: Eliminar Usuarios | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 5/5/2024 | Fecha de fin: 7/5/2024 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: Para llevar a cabo la funcionalidad de eliminar usuarios fueron creados los ficheros UserController, User como nombre del modelo. | |

Table 3.5 Tarea de desarrollo Mostrar Usuarios

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 5 | Número de Historia de usuario: 3 |
| Nombre de la tarea: Mostrar Usuarios | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 9/5/2024 | Fecha de fin: 11/5/2024 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: Para llevar a cabo la funcionalidad de mostrar usuarios fueron creados los ficheros UserController, User como nombre del modelo y dentro de las vistas de users la vista index.blade.php. | |

Table 3.6 Tarea de desarrollo Modificar Usuarios

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 6 | Número de Historia de usuario: 3 |
| Nombre de la tarea: Modificar Usuarios | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 12/5/2024 | Fecha de fin: 14/5/2024 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: Para llevar a cabo la funcionalidad de modificar usuarios fueron creados los ficheros UserController, User como nombre del modelo y dentro de las vistas de users la vista edit.blade.php. | |

**HU7: Crear Operaciones Matemáticas**

Table 3.7 Tarea de desarrollo Crear Operaciones Matematicas

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 7 | Número de Historia de usuario: 7 |
| Nombre de la tarea: Crear Operaciones Matemáticas | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha de inicio: 15/5/2024 | Fecha de fin: 20/5/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: Para llevar a cabo la funcionalidad de crear operaciones matemáticas se hizo uso de la librería jquery la cual provee muchas funcionalidades para darle interactividad al sitio. | |

**Pruebas de software**

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento ( Turrado, 2020).

El lenguaje de codificación de un programa de software puede probarse en busca de errores mediante un proceso de caja negra o de caja blanca. Las [pruebas de caja](https://www.diariodeqa.com/post/tecnicas-de-prueba-de-caja-negra) negra se centran en evaluar la funcionalidad de un sistema sin conocer su estructura interna. Es decir, el probador no tiene acceso al código fuente ni a la lógica interna del software. En lugar de eso, se centra en las entradas y salidas del sistema y cómo se comporta ante diferentes condiciones. Por otro lado, las [pruebas de caja blanca](https://www.diariodeqa.com/post/pruebas-de-caja-blanca) se centran en evaluar la estructura interna del software. Los probadores tienen acceso al código fuente y a la lógica interna, lo que les permite diseñar pruebas específicas para cubrir diferentes rutas de ejecución y segmentos de código. Como prueba de caja negra se utilizaron las pruebas de aceptacion y como pruebas de caja blanca las pruebas unitarias utilizando como herramienta phpunit.

**Pruebas Unitarias**

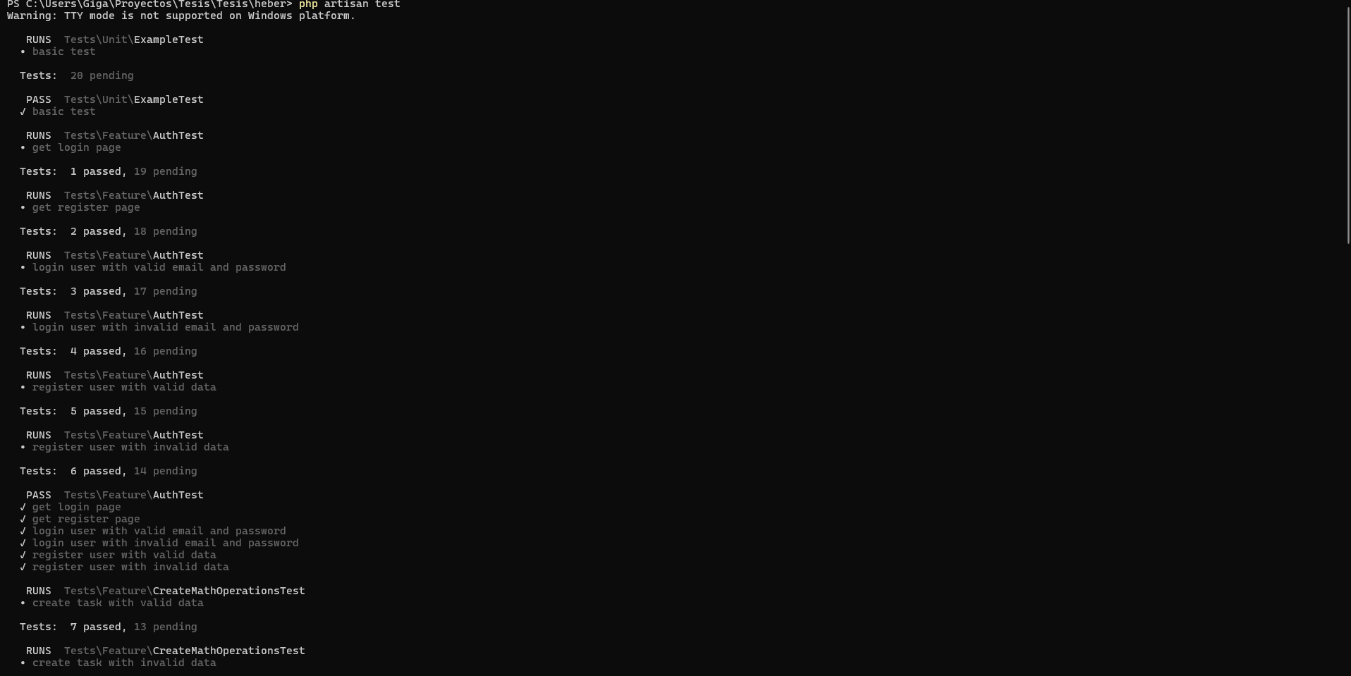
Las pruebas unitarias son una práctica popular en el campo de las pruebas de software que ayuda a los desarrolladores a encontrar errores en el código para poder solucionarlos pronto y ofrecer el mejor producto a sus usuarios finales.

Estos tests se redactan en el mismo lenguaje de programación que el código que están validando y pueden incluir la creación de bases de datos auxiliares para su ejecución. Un test unitario se establece para cada funcionalidad implementada, y se considera que una funcionalidad está completa cuando ha pasado exitosamente su correspondiente test unitario.

En la fase inicial del proyecto de desarrollo, se llevaron a cabo un total de 6 pruebas unitarias, en las cuales los componentes demostraron su éxito al superarlas de manera satisfactoria. Esto permitió avanzar hacia la producción de la primera versión del sistema, marcando así el cierre exitoso de la primera iteración.

En la figura 3.1 se muestran los resultados de la ejecución de los test unitarios de la primera iteración

*Figura 9: Resultados de las pruebas unitarias Iteración 1*



**Pruebas de aceptación**

Las pruebas de aceptación se realizan para establecer el grado de confianza en un sistema.

Estas evaluaciones determinan la calidad del software en relación a las expectativas del cliente, además de identificar posibles discrepancias. Durante esta iteración, se llevaron a cabo cuatro pruebas de aceptación, una por cada Historia de Usuario (HU), con la participación activa del cliente.

*Figura 10: Resultados de las pruebas de aceptación Iteración 1*

Durante la fase de pruebas de aceptación, se verificó que las cuatro Historias de Usuario implementadas satisfacen los requisitos de aceptación establecidos por el cliente. Durante este proceso, el cliente solicitó una modificación en la Historia de Usuario denominada "Crear Operaciones Matemáticas", que implicaba la opción de que el propio estudiante pueda crear sus operaciones matemáticas.

## 3.3.2 Iteración 2

En esta iteración se completaron las Historias de Usuario pendientes que son: Gestionar tareas, Gestionar grados, Generar reportes y Notificar a familiares. Para ello se definieron las siguientes tareas de investigación.

**HU4: Gestionar tareas**

Table 3.8 Tarea de desarrollo Insertar Tareas

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 8 | Número de Historia de usuario: 4 |
| Nombre de la tarea: Insertar tareas | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 24/5/2024 | Fecha de fin: 25/5/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: Para llevar a cabo la funcionalidad de insertar tareas fueron creados los ficheros TaskController, Task como nombre del modelo y dentro de las vistas de tasks la vista add.blade.php. | |

Table 3.9 Tarea de desarrollo Eliminar Tareas

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 9 | Número de Historia de usuario: 4 |
| Nombre de la tarea: Eliminar tareas | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 26/5/2024 | Fecha de fin: 27/5/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: Para llevar a cabo la funcionalidad de eliminar tareas fueron creados los ficheros TaskController y Task como nombre del modelo. | |

Table 3.10 Tarea de desarrollo Mostrar Tareas

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 10 | Número de Historia de usuario: 4 |
| Nombre de la tarea: Mostrar tareas | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 29/5/2024 | Fecha de fin: 30/5/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: Para llevar a cabo la funcionalidad de mostrar tareas fueron creados los ficheros TaskController, Task como nombre del modelo y dentro de las vistas de tasks la vista index.blade.php. | |

Table 3.11 Tarea de desarrollo Modificar Tareas

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 11 | Número de Historia de usuario: 4 |
| Nombre de la tarea: Modificar tareas | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 1/6/2024 | Fecha de fin: 2/6/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: Para llevar a cabo la funcionalidad de insertar tareas fueron creados los ficheros TaskController, Task como nombre del modelo y dentro de las vistas de tasks la vista edit.blade.php. | |

**HU5: Gestionar grados**

Table 3.12 Tarea de desarrollo Insertar Grados

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 12 | Número de Historia de usuario: 5 |
| Nombre de la tarea: Insertar grados | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 3/6/2024 | Fecha de fin: 4/6/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: Para llevar a cabo la funcionalidad de insertar grados fueron creados los ficheros DegreeController, Degree como nombre del modelo y dentro de las vistas de degree la vista add.blade.php. | |

Table 3.13 Tarea de desarrollo Eliminar Grados

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 13 | Número de Historia de usuario: 5 |
| Nombre de la tarea: Eliminar grados | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 5/6/2024 | Fecha de fin: 6/6/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: Para llevar a cabo la funcionalidad de insertar grados fueron creados los ficheros DegreeController y Task como nombre del modelo. | |

Table 3.14 Tarea de desarrollo Mostrar Grados

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 14 | Número de Historia de usuario: 5 |
| Nombre de la tarea: Mostrar grados | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 7/6/2024 | Fecha de fin: 8/6/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: : Para llevar a cabo la funcionalidad de insertar grados fueron creados los ficheros DegreeController, Degree como nombre del modelo y dentro de las vistas de degree la vista edit.blade.php. | |

Table 3.15 Tarea de desarrollo Modificar Grados

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 15 | Número de Historia de usuario: 5 |
| Nombre de la tarea: Modificar grados | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Fecha de inicio: 9/6/2024 | Fecha de fin: 9/6/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: : Para llevar a cabo la funcionalidad de insertar grados fueron creados los ficheros DegreeController, Degree como nombre del modelo y dentro de las vistas de degree la vista edit.blade.php. | |

**HU6: Generar reportes**

Table 3.16 Tarea de desarrollo Generar Reportes

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 16 | Número de Historia de usuario: 6 |
| Nombre de la tarea: Generar reportes | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Fecha de inicio: 9/6/2024 | Fecha de fin: 9/6/2024 |
| Programador responsable: Rosmery González Suarez | |
| Descripción: Para llevar a cabo la funcionalidad generar reporte fue creado el fichero ReportController, y mediante una consulta retorna los resultados a la vista index.blade.php de la carpeta report. | |

**HU8: Notificar a familiares**

Table 3.17 Tarea de desarrollo Notificar a familiares

|  |  |
| --- | --- |
| Tarea | |
| Número de tarea: 17 | Número de Historia de usuario: 8 |
| Nombre de la tarea: Notificar a familiares | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Fecha de inicio: 9/6/2024 | Fecha de fin: 11/6/2024 |
| Programador responsable: Yordanis Ponce Alderete | |
| Descripción: Para llevar a cabo la funcionalidad de notificar a familiares fue creada una función en el fichero ReportController, y una notificación en la carpeta Notifications del proyecto, esta función mediante consultas a la tabla genera el reporte pertinente y envía el resultado vía email. | |

**Pruebas Unitarias**

En la figura 3.3 se muestran los resultados de la ejecución de los test unitarios de la segunda iteración

*Figura 11: Resultados de las pruebas unitarias Iteración 2*



**Pruebas de aceptación**

*Figura 12: Resultados de las pruebas de aceptación Iteración 2*

En el transcurso de esta iteración, se llevaron a cabo un total de cuatro pruebas de aceptación, las cuales fueron realizadas por el cliente. Durante la ejecución de las pruebas de aceptación, se identificó una discrepancia con respecto a la Historia de Usuario (HU) denominada "Notificar familiares". El cliente expresó la necesidad de que el sistema envié los reportes al email de los familiares en formato pdf.

## 3.4 Conclusiones del capítulo

En este capítulo, se observa la realización de las etapas de implementación y pruebas. Se llevaron a cabo las dos iteraciones previstas, implementando un total de 8 Historias de Usuario (HU) a través de 17 Tareas de Implementación (TI). El uso de los estándares de codificación definidos permitió a los desarrolladores leer, entender y mantener el código con facilidad.

Para verificar el producto, se llevaron a cabo pruebas unitarias y de aceptación. Las pruebas unitarias se utilizaron para verificar internamente las funciones del código, lo que facilitó la identificación de errores que podrían llevar a futuras no conformidades. Las pruebas de aceptación aseguraron que las funciones son operativas a través de la interfaz del software, que la entrada se acepta correctamente y se produce un resultado adecuado. Se codificaron un total de 19 pruebas unitarias y se realizaron pruebas de aceptación basadas en un total de 8.

# **CONCLUSIONES**

Con la realización del presente Trabajo de Diploma se arribaron a las siguientes conclusiones:

1. La creación del marco teórico de la investigación proporcionó los fundamentos conceptuales del sistema informático a desarrollar y definió la naturaleza del comportamiento de sus funciones.
2. El análisis del proceso de aprendizaje de las operaciones básicas de la matemática en la educación primaria ayudó a esclarecer el flujo lógico de la información, a reconocer los roles que interactúan con el sistema y aportó en la construcción de una propuesta de solución para el sistema informático a desarrollar.
3. Se desarrolló un sistema informático para el desarrollo de habilidades matemáticas esenciales.
4. El uso de pruebas unitarias y de aceptación para validar el sistema resultó en una solución sólida y precisa, que asegura resistencia y adaptabilidad a futuros cambios.

# **Recomendaciones**

Una vez concluida la investigación y el desarrollo de la Propuesta de solución, los autores del presente trabajo recomiendan:

* Añadir la funcionalidad de chat en tiempo real donde los estudiantes puedan comunicarse entre ellos y hacer comunidad, con el objetivo de aclarar dudas en determinados ejercicios.
* Implementar la funcionalidad de multi idioma con el objetivo de abrir la plataforma a nivel internacional.

# **BIBLIOGRAFÍA**

Aguilar, M. (2012). *Aprendizaje y Tecnologías de Información y Comunicación: Hacia nuevos escenarios educativos*.

Beck, K., & Fowler, M. (2001). *Planning Extreme Programming*. Addison-Wesley Professional.

Castro Ramírez, E. R. (2023). *Patrones en la Arquitectura de Software*. https://es.linkedin.com/pulse/patrones-en-la-arquitectura-de-software-elmo-renato-castro-ramirez

CINE. (s. f.). *Glosario de Conceptos*. Recuperado 31 de marzo de 2024, de https://www.ine.es/DEFIne/es/concepto.htm?c=5325&tf=&op=30459

*COKITOS | Juegos Educativos Online para Niños y Adultos*. (s. f.). Recuperado 25 de mayo de 2024, de https://www.cokitos.com/

Cruz, D. M. H. (2016). *LAS OPERACIONES BÁSICAS COMO PARTE FUNDAMENTAL EN LA SOLUCIÓN DE PROBLEMAS MATEMÁTICOS*.

*CSS Tutorial*. (s. f.). Recuperado 2 de abril de 2024, de https://www.w3schools.com/css/default.asp

González, Y. D., & Romero, Y. F. (2012). Patrón Modelo-Vista-Controlador. *Telemática*, *11*(1), Article 1.

*Grasp—Diseño de software con responsabilidades—GRASP Diseño de objetos con Patrones—Studocu*. (s. f.). Recuperado 17 de abril de 2024, de https://www.studocu.com/latam/document/universidad-ort-uruguay/diseno-de-aplicaciones-1/grasp-diseno-de-software-con-responsabilidades/8717772

Hernandez, R. M. (2017). Impacto de las TIC en la educación: Retos y Perspectivas. *Propósitos y Representaciones*, *5*(1), 325. https://doi.org/10.20511/pyr2017.v5n1.149

*HTML Tutorial*. (s. f.). Recuperado 2 de abril de 2024, de https://www.w3schools.com/html/default.asp

*Introducción—Documentación de phpMyAdmin—6.0.0-dev*. (s. f.). Recuperado 1 de abril de 2024, de https://docs.phpmyadmin.net/es/latest/intro.html

*IXL | Ejercicios de Matemáticas*. (s. f.). Recuperado 25 de mayo de 2024, de https://la.ixl.com/

*JavaScript Tutorial*. (s. f.). Recuperado 2 de abril de 2024, de https://www.w3schools.com/js/default.asp

*Jugar Tux Math en línea*. (s. f.). Recuperado 25 de mayo de 2024, de https://tuxmath.org/index-es.html?opt\_lang=es

*Laravel—The PHP Framework For Web Artisans*. (s. f.). Recuperado 2 de abril de 2024, de https://laravel.com/

*Librería Virtual Cuba | Comprar online | Identifica y cuenta*. (s. f.). Recuperado 25 de mayo de 2024, de https://www.libreriavirtual.cu/identifica-y-cuenta

Lifeder. (2022). *Sistemas: Qué es, significado, tipos y ejemplos*. https://www.lifeder.com/sistema/

*MySQL Tutorial*. (s. f.). Recuperado 2 de abril de 2024, de https://www.w3schools.com/mysql/default.asp

Ortí, C. B. (2011). *LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN (T.I.C.)*.

*PHP: Documentation*. (2024). https://www.php.net/docs.php

Pressman, R. S. (2010). *Ingenieria del Software. Un Enfoque Practico*.

RAE. (2020, junio 25). *Matemática | Diccionario de la lengua española (2001)*. «Diccionario esencial de la lengua española». https://www.rae.es/drae2001/matemática

Tocora Lozano, S. P., & García González, I. (2018). La importancia de la escuela, el profesor y el trabajo educativo en la atención a la deserción escolar. *Varona. Revista Científico Metodológica*, *66*. http://scielo.sld.cu/scielo.php?script=sci\_abstract&pid=S1992-82382018000300024&lng=es&nrm=iso&tlng=en

*UML - Overview*. (s. f.). Recuperado 1 de abril de 2024, de https://www.tutorialspoint.com/uml/uml\_overview.htm

Universidades, S. (2020). *Metodologías de desarrollo de software: ¿qué son?* https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html

Vergara Carmona, M. (2023). Estándares de codificación. *Manuel Vergara Carmona*. https://vergaracarmona.es/estandares-de-codificacion/

*Visual Paradigm—UML, Agile, PMBOK, TOGAF, BPMN and More!* (s. f.). Recuperado 31 de marzo de 2024, de https://www.visual-paradigm.com/features/

*Visual Studio: IDE y Editor de código para desarrolladores de software y Teams*. (s. f.). Visual Studio. Recuperado 31 de marzo de 2024, de https://visualstudio.microsoft.com/es/

# **Anexos**

Entrevista a un profesor de educación primaria en el municipio de La Lisa:

Entrevistador: Hola, gracias por participar en esta entrevista. Me gustaría saber cuáles desafíos has enfrentado al enseñar matemáticas a niños de edad primaria.

Cliente: Mantener el interés y la motivación de los niños es uno de los desafíos más comunes. Algunos encuentran las matemáticas aburridas o difíciles de entender, lo que afecta su participación y rendimiento. También tienen dificultades para comprender conceptos abstractos y aplicarlos en situaciones prácticas.

Entrevistador: ¿Qué tipo de actividades o enfoques has encontrado más efectivos para enseñar matemáticas a estos niños?

Cliente: Las actividades prácticas y concretas son muy efectivas. Los juegos interactivos, manipulaciones físicas y ejemplos del mundo real hacen las matemáticas más accesibles y significativas. Además, los enfoques visuales y el uso de recursos visuales, como gráficos y diagramas, ayudan a facilitar la comprensión.

Entrevistador: ¿Qué características te gustaría ver en una aplicación de desarrollo de habilidades matemáticas para niños de edad primaria?

Cliente: Una aplicación interactiva y visualmente atractiva sería ideal. Debería tener actividades y juegos que sean divertidos pero educativos. Adaptarse al nivel de cada niño y ofrecer un seguimiento personalizado sería genial. También serían útiles explicaciones claras y ejemplos prácticos para facilitar la comprensión.

Entrevistador: Gracias por compartir tus ideas, son realmente valiosas. Si necesitamos más información, definitivamente te contactaremos. ¡Gracias nuevamente y que tengas un buen día!