

**IMPLEMENTACIÓN DE UNA BASE DE DATOS DINÁMICA CON ENFOQUE C.R.U.D PARA
LA GESTIÓN DE INFORMACIÓN ACADÉMICA DE POSGRADOS.**

CASO: MAESTRÍA EN GEOMÁTICA DE LA UNIVERSIDAD DEL CAUCA.



Geógrafa

Luz Alejandra Rojas Calvache

UNIVERSIDAD DEL CAUCA

Facultad de Ingeniería Civil

Maestría en Geomática

Popayán, Cauca.

15 abril 2024

**IMPLEMENTACIÓN DE UNA BASE DE DATOS DINÁMICA CON ENFOQUE C.R.U.D PARA
LA GESTIÓN DE INFORMACIÓN ACADÉMICA DE POSGRADOS.**

CASO: MAESTRÍA EN GEOMÁTICA DE LA UNIVERSIDAD DEL CAUCA.

Luz Alejandra Rojas Calvache

Informe final del Trabajo de Grado

Director:

Ing.- Dr. Nixon Alexander Correa Muñoz

UNIVERSIDAD DEL CAUCA

Facultad de Ingeniería Civil

Maestría en Geomática

Popayán, Cauca.

2024

**IMPLEMENTACIÓN DE UNA BASE DE DATOS DINÁMICA CON ENFOQUE C.R.U.D PARA
LA GESTIÓN DE INFORMACIÓN ACADÉMICA DE POSGRADOS.
CASO: MAESTRÍA EN GEOMÁTICA DE LA UNIVERSIDAD DEL CAUCA.**

Tabla de contenido

1.	Introducción.....	6
1.1.	Descripción Del Problema.....	7
1.2.	Justificación.....	9
1.3.	Objetivos.....	10
1.3.1.	Objetivo General.....	10
1.3.2.	Objetivos específicos.....	10
2.	Estado Del Arte.....	11
2.1.	Bases de datos.....	11
2.2.	Modelo Conceptual.....	13
2.3.	Modelo Entidad-Relación (E-R).....	14
2.4.	Cardinalidad.....	15
2.5.	Modelo Lógico.....	16
2.6.	Modelo Relacional.....	17
2.7.	Clave Primaria Y Clave Foránea.....	17
2.8.	Normalización En Las Bases De Datos.....	18
2.9.	Modelo Interno.....	19
2.10.	Lenguaje SQL.....	19
2.11.	Aplicativo Web.....	21
2.12.	Servidor Web.....	22
2.13.	Protocolo HTTP.....	22
2.14.	Front-End.....	23
2.15.	Back-End.....	23
2.16.	Framework (Marco De Trabajo).....	24
3.	Conceptos Metodológicos De Desarrollo.....	26
3.1.	Patrón De Diseño De Desarrollo Modelo-Vista-Controlador (MVC).....	26

3.2.	SCRUM - Marco De Trabajo Ágil	27
3.3.	Artefactos De Scrum	28
3.4.	Eventos De Scrum	29
3.5.	Herramientas De Desarrollo Software (Stack)	31
4.	Metodología.....	35
4.1.	Caso De Estudio.....	35
4.2.	Diseño Metodológico	37
4.2.1.	Preparación (Backlog Del Producto)	38
	<i>Sprint 1: Estado del arte</i>	39
	<i>Sprint 2: Sesiones de requisitos y análisis del proyecto.</i>	39
4.2.2.	Diseño conceptual.....	41
	<i>Sprint 3: Análisis de requerimientos</i>	41
	<i>Sprint 4: Diseño de las maquetas para la interfaz gráfica</i>	42
4.2.3.	Modelado De Datos	43
	<i>Sprint 5: Modelo Entidad-Relación (Er)</i>	43
	<i>Sprint 6: Diseño De La Estructura De La Base De Datos En PHPmyAdmin</i>	44
	<i>Sprint 7: Construcción De La Base De Datos</i>	44
4.2.4.	Desarrollo Del Aplicativo Web	44
	<i>Sprint 8: Desarrollo Back-end</i>	44
	<i>Sprint 9: Desarrollo Del Front-End</i>	45
	<i>Sprint 10: Control De Calidad</i>	45
4.2.5.	Resultados y evaluación del proyecto.....	46
	<i>Sprint 11 y 12:</i>	46
5.	Resultados y Discusión.....	48
5.1.	Recopilar, revisar, clasificar y sistematizar la información existente del programa de Maestría en Geomática de la Universidad del Cauca.	48
5.1.1.	Sesiones de requisitos y análisis de requerimientos	48
5.1.2.	Diseño de las maquetas para la interfaz gráfica	51
	Login.....	51
	Página Principal.....	52
	Formularios	52

5.2. Diseñar un modelo lógico mediante diagramas entidad/relación, es decir, a través del almacenamiento de datos en tablas (entidades) compuestas por filas (registros) y columnas (atributos)	55
5.3. Adaptar una base de datos en la web con credenciales de acceso para la gestión de la información del programa de Maestría en Geomática.	58
5.4. Validar la base de datos dinámica mediante una prueba piloto con los tomadores de decisión de los programas de posgrado.....	60
6. Conclusiones	64
7. Recomendaciones	66
8. Bibliografía	67

Lista De Tablas

Tabla 1. Dominios de los atributos	15
Tabla 2. Tuplas y atributos (campos) de una tabla.....	17
Tabla 3. Estructura de una C.R.U.D.	21
Tabla 4. Plan general de estudios	36
Tabla 5. Estudiantes de la maestría por cohorte	36
Tabla 6. Modelo de diseño de cada Sprint.....	37
Tabla 7. Modelo de la historia de usuario	40
Tabla 8. Modelo de requerimientos funcionales y no funcionales.....	42
Tabla 9. Resultados encuesta.....	61

Lista de figuras

Figura 1. Etapas en el diseño de una base de datos	13
Figura 2. Conjunto de conceptos del modelo entidad-relación.....	14
Figura 3. Elementos del modelo Entidad-Relación.....	15
Figura 4. Cardinalidad en el modelo entidad-relación.....	16
Figura 5. Ejemplo clave primaria y clave foránea	18
Figura 6. Diagrama de funcionamiento de una petición web	22
Figura 7. Front-end y Back-end: la profundidad del desarrollo web	24
Figura 8. Modelo vista-controlador	27
Figura 9. SCRUM. Marco de trabajo ágil.....	30
Figura 10. Herramientas de desarrollo de software (stack).....	32
Figura 11. Gestión de actividades Scrum	38
Figura 12. Formato aplicación encuesta para evaluadores.	47
Figura 13. Sesiones de requisitos y análisis de requerimientos – Parte 1.	49

Figura 14. Sesiones de requisitos y análisis de requerimientos – Parte 2.	50
Figura 15. Configuración del acceso a la aplicación	51
Figura 16. Recuperar usuario y/o contraseña	52
Figura 17. Vista de la interfaz inicial	52
Figura 18. Modelos de formularios.....	52
Figura 19. Propuesta preliminar del diagrama Entidad/Relación	56
Figura 20. Diagrama Entidad/Relación	57
Figura 21. Estructura base de datos PHPmyAdmin.	58
Figura 22. Página principal del sistema.....	59
Figura 23. Login: Ingreso al aplicativo web.....	60
Figura 24. Formulario final de trabajos de grado.	60

IMPLEMENTACIÓN DE UNA BASE DE DATOS DINÁMICA CON ENFOQUE C.R.U.D PARA LA GESTIÓN DE INFORMACIÓN ACADÉMICA DE POSGRADOS.

CASO: MAESTRÍA EN GEOMÁTICA DE LA UNIVERSIDAD DEL CAUCA.

1. Introducción

El Ministerio de Tecnologías de la Información y las Comunicaciones, en colaboración con empresas innovadoras y diversas universidades, desarrolló una hoja de ruta para capacitar a cien mil colombianos en programación durante el período 2020-2022. Esta iniciativa, enmarcada en el proyecto Jóvenes 4.0, tenía como objetivo generar capacidades y competencias en la resolución de problemas a través de la programación, mejorando así las oportunidades laborales. El plan de capacitación consistió en 800 horas de formación distribuidas en cuatro ciclos, abarcando temas como fundamentos de programación en Python, programación básica en Java, desarrollo de software y desarrollo de aplicaciones web. La formulación de este trabajo se origina en el programa de Maestría en Geomática, específicamente en el área de Fundamentación de Bases de Datos, complementado con los entrenamientos adquiridos durante la misión Mintic 2022, tripulación de la ruta 2.

Este trabajo de maestría utilizó lenguajes de programación, tipos de datos, estructuras secuenciales de asignación, de entrada, de salida, condicionales, iterativas, entre otras. También conceptos de programación orientada a objetos para la relación entre clases de objetos tales como las relaciones de asociación, cardinalidad, agregación, composición, herencia, y polimorfismo. Todo lo anterior para el modelado estructural y codificación del problema de estudio.

El problema de implementar una base de datos con enfoque CRUD exigió un modelamiento para el levantamiento de datos, modelamiento de diagramas de clase, conceptualización del modelo entidad-relación, modelo relacional, y el desarrollo de un SGBD la cual dio la interfaz para la gestión y manipulación para el motor de base de datos. El modelo vista-controlador facilitó el flujo de datos entre los modelos de gestión y la base de datos.

Este informe final del trabajo enfocado en el desarrollo de una aplicación para la gestión de información académica está estructurado en los capítulos: Introducción con el planteamiento del problema, la justificación, y los objetivos que declaran lo que se pretende lograr. El Estado del Arte, con la ontología del campo de estudio de las bases de datos, lenguaje SQL, aplicaciones web y el modelo vista-controlador. Un tercer capítulo presenta los conceptos metodológicos para el desarrollo de la aplicación de gestión de información académica. El capítulo cuatro, describe los métodos implementados para el logro del objetivo general. El capítulo quinto presenta los resultados de la aplicación web, y finalmente, se presentan las conclusiones para declarar el cumplimiento de los objetivos.

La estructura general del informe, el sistema de citación y la presentación de la bibliografía sigue las recomendaciones de la norma APA en su séptima versión.

1.1. Descripción Del Problema

Los modelos de bases de datos tradicionales han sido eficaces en satisfacer diversas necesidades relacionadas con la gestión de datos en aplicaciones convencionales. No obstante, se enfrentan a desafíos y limitaciones significativas al abordar problemas de mayor complejidad y sofisticación. Las nuevas aplicaciones se distinguen por la complejidad en la estructura de datos, la necesidad de almacenar tipos de información multivariable, y la definición de operaciones específicas para cada implementación, entre otras particularidades.

El crecimiento en la complejidad de las bases de datos modernas plantea desafíos en los enfoques para la gestión de datos. En primer lugar, los elementos utilizados deben adaptarse a estructuras de datos complejas, el almacenamiento y recuperación de datos no estructurados, como imágenes y texto, lo cual representa un cambio en la forma en que se manejan los datos, y la implementación de nuevas estrategias y tipos de datos en el sistema de gestión de bases de datos.

Otro desafío, es la necesidad de definir operaciones específicas que no están contempladas en los modelos de bases de datos tradicionales. Estas operaciones

personalizadas son esenciales para satisfacer los requisitos específicos de cada necesidad del usuario, pero los sistemas de bases de datos tradicionales a menudo carecen de flexibilidad para adaptarse a estas necesidades particulares. Por lo tanto, es necesario abordar estos desafíos para permitir la gestión efectiva de datos en aplicaciones más avanzadas, lo que plantea la base para la investigación y desarrollo en el campo de la gestión de bases de datos en entornos contemporáneos.

La gestión académica es un pilar fundamental en el funcionamiento de cualquier institución educativa con enfoque STEM¹, y en este contexto, el Sistema de Matrícula y Control Académico (SIMCA) de la Universidad del Cauca se posiciona como una herramienta esencial para la planeación, registro y control académico en la Institución.

El sistema de información académica ha facilitado las tareas académicas y administrativas para la gestión de la información de los posgrados que ofrece la universidad; sin embargo, esta aplicación presenta algunas limitaciones, entre las que se incluyen la necesidad de un monitoreo constante por parte de los coordinadores de programa, para asegurar el registro apropiado de las actividades académicas de los estudiantes cada semestre. No cuenta con un sistema de cierre automático, lo que requiere solicitudes periódicas a los administradores del sistema en el Centro de Posgrados de la institución.

El sistema controla aspectos como las calificaciones, los créditos, las asignaturas y la información del estudiante, no permite la introducción de datos adicionales, como anteproyectos, trabajos de grado o resoluciones aprobadas por los Consejos de Facultad. Además, la sección de información sobre docentes no permite la consulta de información adicional al nombre del docente asignado a cada asignatura.

En consecuencia, los coordinadores de programa y en este caso, la coordinación del Programa de Maestría en Geomática requiere de una herramienta adicional, que contenga

¹ STEM: Science, technology, engineering, and mathematics.

información estratégica del programa para los actores: estudiantes, y profesores y del plan de estudios: asignaturas, créditos y requisitos de grado. Lo anterior con el fin de automatizar los procesos tradicionalmente manejados en tablas de Excel y agilizar los procesos académicos que beneficien la estandarización y formalización del conocimiento del programa, y enfocarse en tareas de mayor complejidad, como la gestión de la calidad de los estudios a nivel posgradual.

1.2. Justificación

Actualmente, la dependencia de tecnologías de información se ha convertido en un componente esencial para impulsar la eficiencia y la productividad en diversas organizaciones y sectores. La gestión efectiva de bases de datos y el desarrollo de aplicativos webs se han consolidado como herramientas clave para lograr mejoras significativas en la productividad y la toma de decisiones estratégicas.

En particular, la implementación de una base de datos dinámica con un enfoque C.R.U.D (Crear, Leer, Actualizar, y Eliminar) es un enfoque fundamental para optimizar la gestión de información académica en programas de posgrado. La capacidad de almacenar, acceder y actualizar datos de manera eficiente es esencial para administrar con éxito la información académica, que suele ser diversa y cambiante o dinámica.

La combinación de esta base de datos dinámica con aplicativo web proporciona una interfaz accesible para el usuario, lo que facilita la gestión y el acceso a la información en tiempo real, además de que le permite agilizar la gestión de información académica, automatizar tareas repetitivas y mejorar la colaboración entre docentes y estudiantes, mejorando la productividad.

Esta tesis de maestría se fundamenta en la premisa de que la implementación de una base de datos dinámica con un enfoque C.R.U.D, es un elemento crucial en la mejora de la gestión de información académica en programas de posgrado, lo que permitirá optimizar los procesos académicos, formalizar el conocimiento administrativo del programa, y apoyar con

herramientas digitales y automatizadas la toma de decisiones estratégicas en instituciones educativas con oferta posgradual.

1.3. Objetivos

1.3.1. Objetivo General

Implementar una base de datos dinámica con enfoque C.R.U.D para la gestión de información académica de los programas de posgrados. Caso: Maestría en Geomática de la Universidad del Cauca.

1.3.2. Objetivos específicos

Recopilar, revisar, clasificar y sistematizar la información existente del programa de Maestría en Geomática de la Universidad del Cauca.

Diseñar un modelo lógico mediante diagramas entidad/relación, es decir, a través del almacenamiento de datos en tablas (entidades) compuestas por filas (registros) y columnas (atributos).

Adaptar una base de datos en la web con credenciales de acceso para la gestión de la información del programa de Maestría en Geomática.

Validar la base de datos dinámica mediante una prueba piloto con los tomadores de decisión de los programas de posgrado.

2. Estado Del Arte

2.1. Bases de datos

Las bases de datos son sistemas que almacenan y gestionan información interrelacionada, acompañados de programas que permiten su acceso. En esencia, una base de datos organiza la información de manera estructurada, facilitando tanto su almacenamiento como su gestión. Su importancia radica en la capacidad de proporcionar un almacenamiento confiable y accesible, lo que permite un acceso rápido y una gestión efectiva de la información. Este enfoque es fundamental para la toma de decisiones informadas, el análisis de datos, la automatización de procesos y la optimización de la eficiencia en diversas organizaciones y aplicaciones. (Gómez, 2013, p. 5).

Las bases de datos han evolucionado constantemente para adaptarse a las necesidades tecnológicas y empresariales. Según Marqués (2011), la evolución de las bases de datos se ha desarrollado desde sistemas simples de gestión de archivos en la década de 1960 hasta soluciones altamente sofisticadas en la actualidad. Edgar Frank Codd, publicó en el año 1970 un influyente artículo en el que introdujo el innovador modelo relacional denominado “*A Relational Model of data for Large Shared Data Banks*” (“Un modelo relacional de datos para grandes bancos de datos compartidos”), siendo un avance significativo en la gestión de datos, para Codd (1970), los datos se organizan en relaciones, que agrupan información sobre una misma entidad de manera estructurada (p. 377); pasó una década hasta que se desarrollaran los primeros sistemas de gestión de bases de datos relacionales. Entre estos pioneros se encuentra System R, un proyecto de IBM (International Business Machines) diseñado específicamente para probar la viabilidad y la funcionalidad del modelo relacional y proporcionó una implementación práctica de las estructuras de datos y las operaciones relacionales, marcando así el inicio de una nueva era en la gestión de bases de datos (p. 6).

En la segunda mitad de los años setenta, surgieron sistemas de software más avanzados, conocidos como Sistemas de Gestión de Bases de Datos (DBMS, por sus siglas en inglés), que marcaron un hito significativo en la gestión de datos. Según la definición de Espinosa, (2012), un DBMS se define como *"una colección de datos estructurados y relacionados entre sí"*. Estos sistemas no solo permitieron el almacenamiento de múltiples bases de datos en un disco duro, sino que también introdujeron la capacidad de gestionar los datos de manera más eficiente. Los DBMS se basan en modelos de datos que especifican cómo se organizan y relacionan los datos dentro de una base de datos, lo que proporciona una estructura coherente y facilita la gestión de la información de manera organizada y efectiva. Este avance fue fundamental para la evolución de la gestión de datos (p. 24).

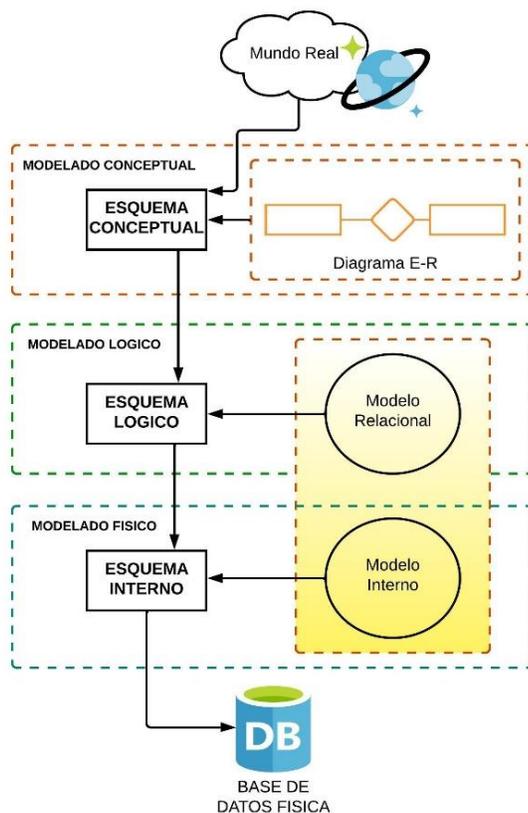
Un modelo de datos se define como:

Un conjunto de conceptos que sirven para describir la estructura de una base de datos, es decir, los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas (lecturas) y actualizaciones de datos. (Marqués, 2011, p. 14).

En este contexto, cada modelo incluye una representación gráfica que ilustra la estructura lógica de la base de datos, incluyendo relaciones y restricciones que determinan cómo se almacenan y acceden a los datos. Por otro lado, según la perspectiva de Sánchez (2009), la clasificación de los modelos de datos abarca diversas etapas que van desde la representación del mundo real hasta la implementación física de la base de datos (p. 17).

En este proceso, el "mundo real" se refiere a la información tal como la percibe el ser humano. Donde, se emplea un *"Esquema conceptual"*, seguido por un *"Esquema lógico"*. Posteriormente, se define el *"Esquema interno"* y, por último, se llega a la *"Base de datos física"* (Sánchez, 2009, p. 17), tal como se puede apreciar en la Figura 1.

Figura 1. Etapas en el diseño de una base de datos



Fuente: propia

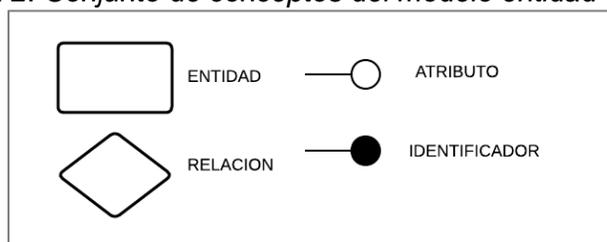
2.2. Modelo Conceptual

Es el proceso de capturar y modelar los conceptos clave que guiarán el diseño de la base de datos en función de los requisitos del usuario. Según Cardona, et al. (2014), este proceso se inicia con la especificación de los requisitos del usuario y tiene como objetivo principal describir la estructura y organización de los datos en la base de datos sin entrar en detalles específicos. En esta etapa, el diseñador utiliza un modelo de datos conceptual para representar de manera abstracta los elementos esenciales que son comprensibles para el usuario. La salida de esta fase es un esquema conceptual, que actúa como una representación de alto nivel de cómo se organizarán y relacionarán los datos en la futura base de datos, proporcionando una base sólida para el desarrollo y diseño subsiguiente. (p.16)

2.3. Modelo Entidad-Relación (E-R)

Presentado por Peter Chen en 1976, permite describir y construir el esquema conceptual de una base de datos y sirve para modelar el almacenamiento de datos. Marqués (2011), considera que este modelo se compone de un “conjunto de conceptos que permiten representar de manera efectiva la realidad, haciendo uso de representaciones gráficas y lingüísticas” (p. 107). Estos conceptos se muestran en la Figura 2.

Figura 2. Conjunto de conceptos del modelo entidad-relación



Fuente: propia

Marqués (2011) explica que los elementos centrales incluyen las "entidades", que son objetos del mundo real con existencia física o abstracta, las "interrelaciones" que capturan las conexiones entre estas entidades, los "dominios" definen el rango de valores que el atributo puede asumir (Tabla 1) y los "atributos" que caracterizan las propiedades de las entidades, tal como se muestra en la Figura 3 (p. 106). Este modelo empleó *diagramas entidad-relación (DER)* para visualizar y comunicar estas relaciones y conceptos de manera clara y efectiva (Cardona et al., 2014), ya que facilita la comprensión y representación de la estructura y organización de los datos en una base de datos, siendo un punto de partida crucial en el diseño de sistemas de gestión de bases de datos (p.116).

Figura 3. Elementos del modelo Entidad-Relación



Fuente: propia

A continuación, en la Tabla 1, se presentan ejemplos que ilustran el dominio del atributo, detallando los distintos valores que este puede adquirir:

Tabla 1. Dominios de los atributos

Atributo	Dominio	Descripción del dominio
Experiencia docente	Int (Integer - Entero)	Sirve para almacenar números enteros, es decir, números sin decimales
Créditos totales	VARCHAR (Variable Character - Character Variable)	Puede almacenar cadenas de hasta 50 caracteres
Fecha inicio estudios	DATE (Fecha)	Podría utilizarse para almacenar la fecha y hora

Fuente: propia

2.4. Cardinalidad

En el modelo entidad-relación, es esencial definir y especificar la cantidad de instancias de una entidad que pueden estar relacionadas con instancias de otra entidad en un sistema de gestión de bases de datos. Este concepto es conocido como "*cardinalidad*" y juega un papel fundamental en la garantía de la integridad de los datos y la correcta implementación de la base de datos. De acuerdo con la definición de Gómez (2013) establece los siguientes tipos de cardinalidad:

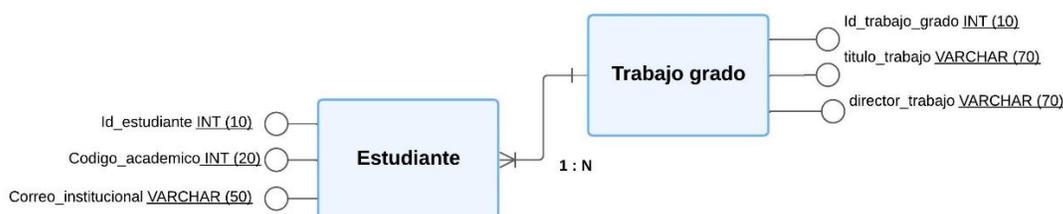
Cardinalidad 1:1 (Una a una): Esto significa que cada ocurrencia de la primera entidad se relaciona con exactamente una y solo una ocurrencia de la segunda entidad, y viceversa. Es una relación de correspondencia uno a uno.

Cardinalidad 1: N (Una a muchas): En este caso, cada ocurrencia de la primera entidad puede relacionarse con más de una ocurrencia de la segunda entidad, pero cada ocurrencia de la segunda entidad se relaciona con no más de una ocurrencia de la primera entidad. Es una relación de uno a muchos.

Cardinalidad N: N (Muchas a muchas): Aquí, cada ocurrencia de la primera entidad puede relacionarse con más de una ocurrencia de la segunda entidad, y viceversa. Se trata de una relación de muchos a muchos. (p.36)

La Figura 4 muestra el modelo entidad-relación utilizando la cardinalidad uno a muchas (1: N). En este contexto, varios estudiantes pueden estar asociados a un trabajo de grado y un trabajo de grado puede ser desarrollado por uno o varios estudiantes.

Figura 4. Cardinalidad en el modelo entidad-relación



Fuente: propia

2.5. Modelo Lógico

Representa una vista estructurada y abstracta de la organización de los datos y sus relaciones dentro de una base de datos. Según la definición de Marqués (2011), esta etapa implica transformar el esquema conceptual, obtenido en la fase previa del diseño, en un esquema lógico que se adapte a las estructuras de datos del modelo de base de datos subyacente al Sistema de Gestión de Bases de Datos (SGBD) que se va a utilizar (p.14).

2.6. Modelo Relacional

Uno de los Modelos Lógicos más ampliamente empleados es el modelo relacional, donde, los datos se organizan y almacenan en tablas separadas que se relacionan o combinan de acuerdo con las necesidades específicas. Estos datos se presentan en un formato tabular de filas y columnas, lo que permite extraer y combinar información de manera flexible (CEPSUNI, 2021, p.14). Es importante destacar que el Modelo Relacional opera siempre sobre relaciones o tablas de datos, no sobre datos individuales contenidos en un archivo (Gómez, 2013, p. 19).

Cada tabla en el modelo relacional está compuesta por filas (registros o tuplas) y columnas (campos o atributos). Las tuplas representan conjuntos de datos, mientras que los atributos corresponden a categorías de información que se aplican a cada tupla, evento o registro (Tabla 2). Los atributos en la base de datos contienen valores que definen el tipo de datos que pueden almacenar, cómo se muestran y las operaciones que se pueden realizar con ellos (Gómez, 2013, p.20).

Tabla 2. Tuplas y atributos (campos) de una tabla

Campos o atributos		
	ESTUDIANTE	TRABAJO DE GRADO
Registros o tuplas	Nombre estudiante	Titulo trabajo
	Identificación estudiante	Resolución aprobada
	Código académico	Fecha de resolución
	Correo Institucional	Director Trabajo de grado

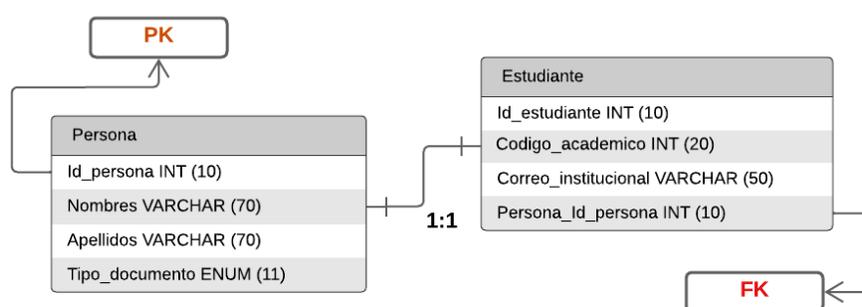
Fuente: propia

2.7. Clave Primaria Y Clave Foránea.

Cuando se habla de claves en una base de datos, se hace referencia la clave principal y clave foránea, la primera conocida también como **Primary Key (PK)**, es aquella que se utiliza

para distinguir de forma única cada instancia de una entidad en particular, en el contexto de una entidad dada, es posible seleccionar más de un atributo o conjunto de atributos como clave principal (CEPSUNI, 2021, p.34) y la clave foránea, conocida como **Foreign Key (FK)**, se refiere “al atributo de una entidad que la relaciona con otra entidad a través de la clave primaria de ésta. La clave foránea es un atributo que es clave primaria en la otra entidad” (CEPSUNI, 2021, p.35). La Figura 5 muestra como el atributo Persona_Id_persona es la clave primaria de la entidad PERSONA, y se ha definido como la clave foránea de la entidad ESTUDIANTE.

Figura 5. Ejemplo clave primaria y clave foránea



Fuente: propia

2.8. Normalización En Las Bases De Datos

Es fundamental comprender la importancia de la normalización de datos en el contexto del modelo relacional. Según Cardona et al. (2014), la normalización tiene como objetivo principal evitar la creación de tablas redundantes e ineficientes, las cuales pueden dar lugar a errores durante la manipulación de la información. En términos generales, “se trata de un proceso reversible que se realiza de manera secuencial y que implica transformar un conjunto de relaciones en otro conjunto que posea características más deseables, aunque sea equivalente en cuanto a la preservación de la información”. En resumen, la normalización representa un conjunto de reglas y pautas que se aplican para estructurar y organizar eficazmente las tablas de una base de datos, minimizando redundancias y garantizando la integridad de los datos (p. 44).

2.9. Modelo Interno

Se encarga de personalizar los datos según las características particulares de un sistema de gestión de bases de datos (SGBD) específico. Siguiendo el diseño lógico de la base de datos y considerando su uso previsto, se crea una configuración física adaptada al entorno de alojamiento. Esto tiene como objetivo habilitar el almacenamiento y el acceso eficiente a los datos, garantizando un rendimiento óptimo. Todo lo anterior mediante un puente esencial, y es el lenguaje SQL (Structured Query Language).

2.10. Lenguaje SQL

Oppel et al., (2010) considera que el lenguaje SQL permite la “*creación y mantenimiento de la base de datos relacional y la gestión de los datos dentro de la base de datos*”, donde, el modelo relacional es quien proporciona las bases teóricas de la base de datos relacional, pero SQL apoya la aplicación física de esa base de datos. En la década de 1970, tras la publicación del artículo de E. F. Codd, IBM (International Business Machines Corporation) inició el desarrollo de un lenguaje y sistema de bases de datos destinados a aplicar dicho modelo. Inicialmente, este lenguaje se conoció como "Structured English Query Language" (SEQUEL), pero se vio obligado a cambiar su nombre debido a problemas de marca registrada. Así nació SQL, el acrónimo de Structured Query Language, este lenguaje emergió como el estándar en el ámbito de las bases de datos relacionales (p.24).

SQL es un lenguaje integral que posibilita la creación y el mantenimiento de objetos en una base de datos, así como la seguridad de estos objetos y la manipulación de la información contenida en los mismos. Para una clasificación más organizada de las instrucciones SQL, estas se dividen comúnmente en función de las tareas que realizan. Siguiendo este enfoque, SQL se divide en tres tipos principales de instrucciones: DDL, DCL, y DML, descritos a continuación.

La implementación de una base de datos se logra a través del uso del *Lenguaje de definición de datos (DDL, Data Definition Language)*. Este lenguaje permite crear, modificar o

eliminar objetos en la base de datos, tales como tablas, vistas, esquemas, dominios, disparadores y procedimientos almacenados. En SQL, las palabras clave más frecuentemente asociadas con las instrucciones DDL son CREATE, ALTER y DROP. Por ejemplo, se emplea la instrucción CREATE TABLE para crear una tabla, ALTER TABLE para realizar modificaciones en sus características y DROP TABLE para eliminar la definición de la tabla en la base de datos (Oppel et al., 2010, p.18).

Las instrucciones que regulan el acceso de usuarios o entidades a objetos específicos dentro de una base de datos se gestionan mediante el *Lenguaje de control de datos (DCL, Data Control Language)*. Este lenguaje permite conceder o revocar el acceso utilizando las instrucciones GRANT o REVOKE. No obstante, también faculta la definición del nivel de acceso que cada usuario posee a los objetos de la base de datos. Por ejemplo, es posible especificar cuáles usuarios pueden visualizar un conjunto de datos en particular y cuáles tienen la capacidad de modificar dichos datos (Oppel et al., 2010, p.19).

Ahora bien, el *Lenguaje de manipulación de datos (DML, Data Manipulation Language)* desempeña un papel esencial al permitir la recuperación, adición, modificación o eliminación de datos almacenados en los objetos de una base de datos. Las palabras clave asociadas a las instrucciones DML más utilizadas son SELECT, INSERT, UPDATE y DELETE. Por ejemplo, la instrucción SELECT se emplea para recuperar datos de una tabla, mientras que INSERT se usa para agregar información a una tabla (Oppel et al., 2010, p.19).

Como complemento a lo anterior, Ramírez, D., et al., (2015), señala que el DML es un lenguaje proporcionado por los Sistemas de Gestión de Bases de Datos (SGBD) que permite realizar tareas de consulta y modificación de los datos almacenados en las bases de datos del sistema. Un conjunto fundamental de operaciones de manipulación de datos se conoce como operaciones CRUD, que abarcan la **creación (Create)**, **lectura (Read)**, **actualización (Update)** y **eliminación (Delete)** de datos (p. 2). La Tabla 3 proporciona una representación detallada de estos conceptos.

Tabla 3. Estructura de una C.R.U.D.

SIGLA	CRUD	OPERACIÓN	COMANDO	ESTRUCTURA
C	CREATE	Creación de un nuevo registro en la base de datos	<i>INSERT</i>	INSERT INTO nombreDeTabla (columna1, columna2, columna3...) VALUES (valor1,valor2,valor3,...);
R	READ	Lectura de registros de base de datos	<i>SELECT</i>	SELECT columna1, columna2, ... FROM nombreDeTabla; SELECT*FROM table_name;
U	UPDATE	Actualización de registros de base de datos	<i>UPDATE</i>	UPDATE nombreDeTabla SET columna1=valor1, columna2, ... WHERE condition;
D	DELETE	Eliminación de registros de base de datos	<i>DELETE</i>	DELETE FROM nombreDeTabla WHERE condition;

Fuente: propia

2.11. Aplicativo Web

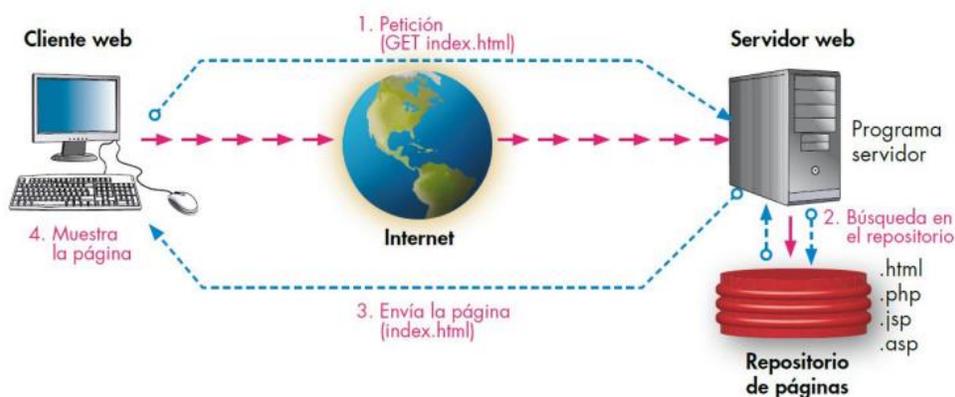
Según Luján, S., (2002) una aplicación web, representa un tipo especial de aplicación cliente-servidor en la que tanto el cliente (por ejemplo, el navegador) como el servidor (el servidor web) y el protocolo de comunicación (HTTP) están estandarizados, lo que significa que el programador de aplicaciones no necesita crearlos desde cero (p.48). Por lo tanto, una aplicación web es una interfaz o conjunto de páginas web que interactúan con el usuario final, permitiendo el acceso a la información solicitada y la manipulación de datos relacionados con el

modelo de negocio. Esto significa que cualquier persona puede interactuar con la aplicación a través de Internet, utilizando simplemente un navegador web (p. 54).

2.12. Servidor Web

De acuerdo con Cases (2014), un servidor web es un programa diseñado para la transferencia de datos de hipertexto, lo que implica la transmisión completa de páginas web con todos sus elementos, como textos, widgets y banners, estos servidores utilizan el protocolo HTTP y están alojados en un ordenador con conexión a Internet. Un servidor web funciona cuando el cliente inicia una solicitud a través de su navegador, esta solicitud se dirige al servidor asociado. A su vez, el servidor responde proporcionando la página o recurso solicitado, utilizando los lenguajes HTML y CSS, estos lenguajes son interpretados por los navegadores web, que utilizan la información recibida para configurar y mostrar de manera adecuada las páginas web. La siguiente figura muestra de manera gráfica el proceso de una solicitud al servidor web y la respectiva respuesta (p. 1).

Figura 6. Diagrama de funcionamiento de una petición web



Fuente: Internet Protocols, 2003

2.13. Protocolo HTTP

Según Primo, (2012) el protocolo HTTP (Hypertext Transfer Protocol) es un protocolo cliente-servidor esencial, facilita las interacciones entre clientes web y servidores HTTP, fue concebido por Tim Berners-Lee para respaldar la distribución global de información a través de

la World Wide Web. HTTP opera mediante operaciones de solicitud/respuesta. Un cliente inicia la conexión con un servidor, transmitiendo una solicitud, y el servidor responde con un mensaje similar, indicando el estado de la operación y su resultado potencial, cada operación se realiza en un objeto o recurso específico, identificado por su URL, que puede ser un documento HTML, archivo multimedia o aplicación CGI (p. 6).

2.14. Front-End

Chapaval (2017) explica que el front-end, en el ámbito del diseño y desarrollo web, comprende todas las tecnologías y elementos que operan en el navegador y permiten la interacción directa con los usuarios. Es, en esencia, la cara visible de una aplicación o sitio web, y se encarga de aspectos cruciales como:

Diseño: Define la apariencia visual, incluyendo el diseño gráfico, la disposición de elementos, la elección de colores y fuentes.

Interfaz: Proporciona botones, menús y controles que permiten a los usuarios interactuar con la aplicación de manera intuitiva.

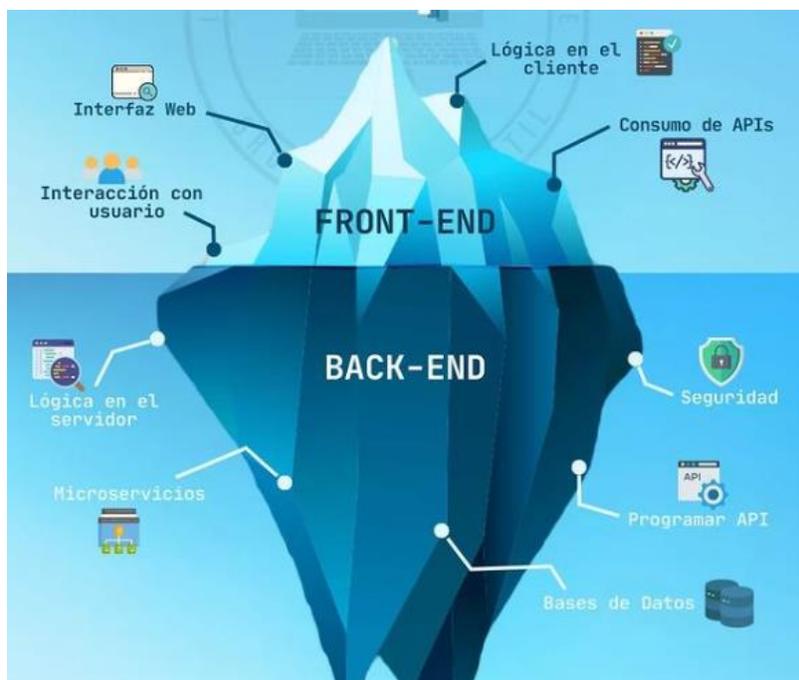
Respuesta a acciones: Responde a las acciones del usuario, como hacer clic en un botón, llenar formularios o navegar por la página, asegurando que todo funcione sin problemas y de manera efectiva (p. 1).

2.15. Back-End

El back-end es la capa lógica y tecnológica que se encarga de manejar y procesar la información que alimenta al front-end, el trabajo en el back-end implica el uso de lenguajes de programación como PHP. El back-end es la parte de una aplicación web o software que opera fuera de la vista del usuario, que trabaja en el servidor y se encarga de procesar datos, manejar la lógica interna del sistema, interactuar con la base de datos y responder a las solicitudes del usuario. Por lo tanto, es una parte invisible pero esencial y fundamental para que una aplicación web funcione y cumpla con las necesidades del usuario (García, 2021, p. 1).

La Figura 7 visualiza la dinámica del desarrollo web, resaltando de manera gráfica y comprensible las diferencias esenciales entre Front-end y Back-end.

Figura 7. Front-end y Back-end: la profundidad del desarrollo web



Tomado de ge_its_fime (2022)

2.16. Framework (Marco De Trabajo)

En el desarrollo de software, Gutiérrez (2017) define un framework como una estructura de soporte que permite organizar y desarrollar proyectos de software. Por lo general, incluye herramientas, bibliotecas y un lenguaje de scripting para facilitar la creación y la integración de componentes en un proyecto. Los frameworks extienden las capacidades de un lenguaje de programación al automatizar patrones comunes de programación para un propósito específico. Esto proporciona una estructura al código, lo que resulta en código más claro y mantenible, al mismo tiempo que simplifica la programación al convertir tareas complejas en instrucciones más sencillas. Los frameworks están típicamente escritos en el mismo lenguaje que se extiende. Los frameworks web, en particular, simplifican el desarrollo de aplicaciones web, como sitios web y aplicaciones intranet. PHP es ampliamente utilizado en este ámbito debido a

su simplicidad y su capacidad para trabajar con una variedad de motores de bases de datos, gestionar sesiones y acceder a archivos en el servidor (p. 1).

3. Conceptos Metodológicos De Desarrollo

Los conceptos metodológicos de desarrollo se refieren a los principios y enfoques que guían el proceso de desarrollo de software, estos permiten planificar, diseñar, implementar y mantener sistemas de software. Algunos de los conceptos metodológicos clave en el desarrollo del aplicativo web se describen a continuación.

3.1. Patrón De Diseño De Desarrollo Modelo-Vista-Controlador (MVC)

Bone (2019), define el patrón MVC como un "concepto y diseño arquitectónico de software utilizado en diversos lenguajes y aplicaciones". Este patrón simplifica el proceso al separar las Vistas, los Modelos y los Controladores, lo que resulta en un entorno más estructurado y organizado. La mayoría de las aplicaciones modernas siguen este patrón de diseño, el cual, fue desarrollado por Trygve Reenskaug para la plataforma SmallTalk (un lenguaje orientado a objetos) en la década de 1970 (p.1).

El patrón de diseño Modelo-Vista-Controlador (MVC) es una guía arquitectónica para organizar y estructurar sistemas de software. Bone (2019) explica que este patrón se divide en tres componentes clave:

Modelo: Representa los datos y la lógica de la aplicación. Es responsable de gestionar cómo se almacenan y manipulan los objetos de datos, a menudo interactuando con una base de datos mediante consultas (como SELECCIONAR, ACTUALIZAR o ELIMINAR).

Vista: se encarga de la representación visual en el patrón MVC, crea la interfaz de usuario que se presenta al usuario final. Aunque la Vista es responsable de la presentación, no contiene la lógica para mostrar datos por sí sola; el Controlador proporciona los datos necesarios para su visualización.

Controlador: es el componente que actúa como el coordinador central del patrón MVC. Su función principal es facilitar la comunicación entre el Modelo y la Vista. Captura la entrada del usuario y la convierte en acciones adecuadas que actualizan la Vista cuando los datos cambian. Es esencialmente el "cerebro" del diseño MVC (p.1).

En la Figura 8, se visualiza de manera clara y detallada el funcionamiento del Patrón de diseño de desarrollo Modelo-Vista-Controlador (MVC).

Figura 8. Modelo vista-controlador



Tomado de EDteam (2020)

3.2. SCRUM - Marco De Trabajo Ágil

La Guía de Scrum descrita por Schwaber, K., et al (2020), establece que Scrum es una metodología que se basa en aprender de la experiencia práctica y la observación directa, lo que se llama "*empirismo*". Utiliza ciclos regulares de trabajo cortos llamados "*Sprints*" para desarrollar el producto paso a paso. Esto ayuda a prever cuánto trabajo se puede realizar (predictibilidad) y a abordar riesgos de manera temprana (control del riesgo). En resumen, Scrum fomenta aprender haciendo, adaptarse según la experiencia y mejorar constantemente para lograr mejores resultados en el desarrollo de proyectos (p. 3).

Dentro de esta metodología, Alzate et al., (2016) destaca los siguientes roles (p.8):

Product Owner: Es el enlace directo entre el equipo y el cliente, representando las necesidades y expectativas del cliente en relación con el proyecto. Su papel es garantizar que el equipo satisfaga las expectativas del cliente y cumpla con sus requisitos clave.

Scrum Master: Dirige las reuniones y proporciona apoyo al equipo en caso de desafíos. Su función principal es eliminar obstáculos que puedan obstaculizar el progreso del equipo y

garantizar que se sigan las prácticas de Scrum. El Scrum Master actúa como un facilitador en lugar de un jefe.

Scrum Team: Son los encargados de desarrollar y cumplir lo que les asigna el Product Owner.

Cliente: Recibe el producto final y puede influir en el proceso, puede influir en el desarrollo proporcionando comentarios y sugerencias que contribuyan al éxito del proyecto.

3.3. Artefactos De Scrum

Schwaber, K., et al (2020), se refiere a los artefactos de Scrum como representaciones tangibles de trabajo o valor, diseñados para brindar transparencia y facilitar la inspección y adaptación. Su función principal es maximizar la claridad de la información esencial, asegurando un entendimiento uniforme entre los miembros del equipo y partes interesadas (p.10). En Scrum los artefactos principales son (Figura 9):

Lista de Producto (Product Backlog):

La Lista de Producto es esencial en el desarrollo del producto, ya que representa un registro organizado de todo lo que potencialmente podría ser necesario, es la única fuente de requisitos para cualquier modificación que se realice en el producto. En este contexto, el dueño de Producto asume la responsabilidad de la Lista de Producto, abarcando aspectos como su contenido, disponibilidad y orden, la Lista de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituirán los cambios a implementar en futuras entregas del producto, los elementos de la Lista de Producto poseen atributos clave como la descripción, la ordenación, la estimación y el valor, proporcionando una guía completa para el desarrollo y la toma de decisiones (Schwaber, K., et al 2020, p.11).

De acuerdo con Schwaber, K., et al (2020), la Lista de Producto no es estática ni completa en su totalidad, ya que, en sus primeras etapas de desarrollo, refleja únicamente los requisitos conocidos y mejor comprendidos en ese momento. Sin embargo, la evolución de la

Lista de Producto es constante, adaptándose al progreso del producto y a los cambios en el entorno de uso (p.11).

Lista de Pendientes del Sprint (Sprint Backlog):

Schwaber, K., et al (2020), considera que la Lista de Pendientes del Sprint constituye un plan detallado que permite comprender los cambios en el progreso durante el Scrum Diario, es dinámica y se ajusta continuamente durante el Sprint a medida que el Equipo de Desarrollo trabaja y adquiere una comprensión más profunda de los requisitos para lograr el Objetivo del Sprint (p.10).

El Equipo de Desarrollo añade nuevo trabajo a la lista según sea necesario y actualiza las estimaciones a medida que avanza, los elementos innecesarios se eliminan, y solo el Equipo de Desarrollo tiene la capacidad de modificar la Lista de Pendientes del Sprint. Esta lista sirve como una representación visual en tiempo real del trabajo planificado y ejecutado, siendo exclusiva del equipo y fundamental para la planificación y ejecución eficiente del desarrollo del producto (Schwaber, K., et al 2020, p.10).

3.4. Eventos De Scrum

En esta metodología ágil se establecen eventos planificados que tienen una duración máxima predeterminada, lo que garantiza un uso eficiente del tiempo, todos los eventos son bloques de tiempo (Sprint). Alzate et al. (2016) explica que una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse, los demás eventos pueden concluir una vez que se logre su objetivo, asegurando que se utilice el tiempo adecuadamente sin desperdiciar recursos (p, 8).

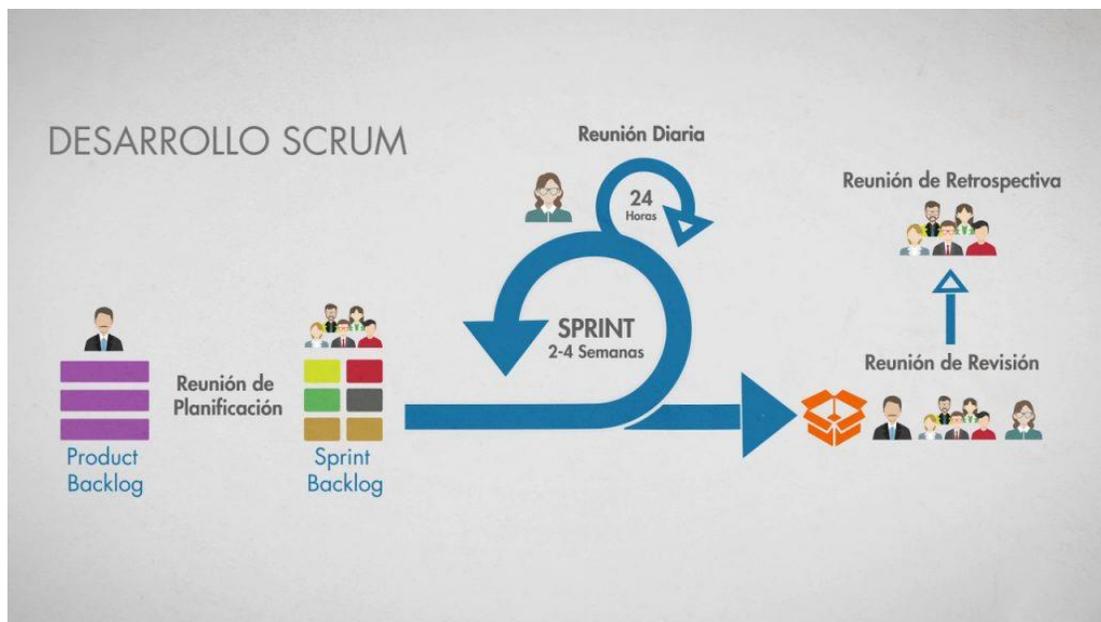
Además del Sprint, que actúa como un marco temporal para los demás eventos, cada uno de estos eventos en Scrum brinda una oportunidad formal para inspeccionar y adaptar aspectos específicos del proyecto. La omisión de cualquiera de estos eventos reduce la visibilidad y representa una oportunidad perdida para realizar evaluaciones y ajustes necesarios (Schwaber, K., et al 2020, p.7). El período de tiempo de un Sprint es de dos

semanas a un mes aproximadamente, cada nuevo Sprint comienza inmediatamente después de que finaliza el Sprint anterior. Los eventos Scrum contienen:

- *Reunión de Planificación del Sprint (Sprint Planning)*
- *Los Scrums Diarios (Daily Scrums)*
- *Revisión del Sprint (Sprint Review)*
- *Retrospectiva del Sprint (Sprint Retrospective)*

La Figura 9 ilustra los artefactos y eventos de Scrum.

Figura 9. SCRUM. Marco de trabajo ágil



Tomado de Metodología Scrum, una herramienta útil para agilizar tus proyectos – Honduras

Digital Challenge

Schwaber, K., et al (2020) define los eventos Scrum de la siguiente manera:

Reunión de Planificación del Sprint (Sprint Planning): “*El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo*”, el equipo Scrum trabaja para decidir qué tareas harán durante el próximo periodo de tiempo (Sprint) (p. 8).

Los Scrums Diarios (Daily Scrums): El Scrum Diario es una breve reunión diaria que dura solo 15 minutos, en la que el equipo de desarrollo se pone al día, coordina sus actividades y planifica lo que hará en las próximas 24 horas. Durante esta reunión, revisan lo que han logrado desde la última vez y discuten lo que esperan lograr antes de la próxima reunión. En resumen, el Scrum Diario es una reunión breve pero esencial que ayuda al equipo de desarrollo a mantenerse en sintonía, resolver problemas y tomar decisiones rápidas, lo que lo convierte en una parte clave del proceso de Scrum (p.9).

Revisión del Sprint (Sprint Review): La Revisión de Sprint es una reunión que ocurre al final de cada Sprint para examinar el trabajo completado y, si es necesario, ajustar la Lista de Producto. Durante esta reunión, se discute sobre qué se logró en el Sprint y cómo mejorar el valor del producto, además, su objetivo principal es obtener retroalimentación sobre el Incremento del producto y fomentar la colaboración entre los participantes (p.10).

Retrospectiva del Sprint (Sprint Retrospective): el equipo Scrum realiza una retrospectiva para reflexionar sobre el Sprint pasado. Se discuten lo que funcionó bien, lo que podría mejorarse y se definen acciones para implementar mejoras en el proceso de desarrollo en el próximo Sprint (p.10).

3.5. Herramientas De Desarrollo Software (Stack)

Para introducir el tema de las herramientas de desarrollo de software, es fundamental comprender el concepto de "stack". Este término en palabras de Chapaval (2017), se refiere a un *“conjunto de tecnologías que abarca todos los aspectos de un sitio web, desde la gestión de la base de datos hasta el procesamiento lógico y la interfaz visual”*, el stack es el núcleo sobre el cual se construye un sitio web. Además, como señala García (2021), el stack engloba todas las *“tecnologías y herramientas que se utilizan para desarrollar una aplicación, es decir, son todas las tecnologías que usa el front-end y el back-end”* (p.1).

En resumen, las herramientas de desarrollo software hacen referencia a diferentes aplicativos, entornos, librerías y utilidades que sirven a los desarrolladores de sistemas

informáticos para la implementación de proyectos software (Chapaval, 2017, p.1), a continuación, se describen un conjunto de herramientas de desarrollo comúnmente utilizadas para diseñar, implementar y mantener un aplicativo web que permita el acceso a información en una base de datos (Figura 10).

Figura 10. Herramientas de desarrollo de software (stack)



Fuente: propia

XAMPP: Valencia (2013) detalla que XAMPP constituye un servidor independiente de plataforma y software libre, el nombre "XAMPP" se origina en un acrónimo que representa los principales componentes del software: "X" para denotar su compatibilidad con diversos sistemas operativos como Windows, Linux y macOS; "A" refiriéndose al servidor web Apache; "M" asociado al sistema de gestión de bases de datos MySQL; "P" indicando el lenguaje de programación PHP; y "P" nuevamente en alusión al lenguaje de programación Perl (p.74).

MySQL: El sitio web oficial de MySQL (<https://www.mysql.com/>) brinda información sobre MySQL - Structured Query Language (Lenguaje de consulta estructurado) que es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que se basa en el lenguaje de consulta estructurado (SQL), es especialmente apreciado en aplicaciones web debido a su facilidad de uso, escalabilidad y alto rendimiento. Además, ofrece un conjunto

completo de controladores de base de datos y herramientas visuales que facilitan la construcción y administración de aplicaciones críticas.

PHP: PHP hace referencia a "*Hypertext Preprocessor*", es el lenguaje de servidor más utilizado en la web, nacido en 1994 pero evolucionando con la velocidad de Internet, PHP es valorado por su potencia y simplicidad, además de ser ampliamente soportado en servidores de hosting, incluso los más asequibles. La facilidad de PHP es que se incorpora directamente en el código HTML permitiendo integrar fácilmente scripts PHP en páginas web y ofrece una amplia gama de funciones para trabajar con diversos recursos, como la manipulación de bases de datos, la gestión de formularios web y la generación dinámica de contenido, lo que facilita la creación de aplicaciones web interactivas y dinámicas (Álvarez et al., 2016, p.4).

PHPmyAdmin: PhpMyAdmin es una herramienta de software gratuita (Aplicativo WEB), escrita en PHP que facilita la administración de MySQL a través de la web, ofrece una amplia gama de funciones, incluyendo la gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios y permisos, todo ello a través de una interfaz de usuario intuitiva. Además, proporciona la flexibilidad de ejecutar directamente declaraciones SQL cuando sea necesario (Sitio oficial phpMyAdmin (<https://www.phpmyadmin.net/>), s.f.).

CSS (*Cascading Style Sheets*): CSS (Cascading Style Sheets), traducido como Hojas de Estilo en Cascada, no se clasifica como un lenguaje de programación ni como un lenguaje de marcado. Más bien, se define como un lenguaje de hojas de estilo que permite aplicar estilos de manera selectiva a elementos dentro de documentos HTML (MDN contributors, 2022, p.1).

Lenguaje HTML: Según Martínez (1995), el **lenguaje HTML** (Lenguaje de Marcado de Hipertexto) es un lenguaje sencillo que posibilita la creación de contenido estructurado y visualmente atractivo. A través del HTML, se pueden agregar enlaces a otros documentos o fuentes relacionadas, e incluir elementos multimedia como imágenes y sonidos. El enfoque principal del HTML radica en definir la estructura lógica del contenido, abarcando aspectos

como títulos, párrafos, listas, citas, entre otros, además de permitir la especificación de estilos y efectos, tales como cursivas, negritas o la inserción de gráficos. La presentación final de este hipertexto se logra a través de un software especializado (p.3).

JavaScript: Álvarez et al. (2016), describen JavaScript como un lenguaje de programación caracterizado por su simplicidad y enfoque en la eficiencia, este lenguaje permite realizar rápidamente una variedad de tareas, a veces con un enfoque ligero. JavaScript posibilita la implementación de efectos especiales en páginas web, facilitando la creación de contenido dinámico y elementos que pueden tener movimiento, cambios de color u otros comportamientos dinámicos. Además, JavaScript permite la ejecución de instrucciones en respuesta a las acciones del usuario, lo que posibilita la creación de páginas interactivas con funcionalidades como calculadoras, agendas o tablas de cálculo (p.5).

Librería Bootstrap: Bootstrap es un framework de desarrollo web de código abierto y gratuito diseñado para simplificar el proceso de creación de sitios web responsivos, ofrece una amplia gama de sintaxis para la construcción de plantillas, permitiendo a los desarrolladores acelerar la creación de interfaces atractivas. Este framework elimina la necesidad de preocuparse por los comandos y funciones básicos al proporcionar scripts basados en HTML para diversas funciones y componentes relacionados con el diseño web. Entre sus características se encuentran elementos esenciales como ventanas modales, menús, cuadros, botones y formularios, abarcando todo lo necesario para la maquetación efectiva de una página web (Deyimar, 2023, p.1).

Librería Font Awesome: El sitio web oficial de Font Awesome (<https://fontawesome.com/>) explica que la librería Font Awesome representa la biblioteca de iconos y conjunto de herramientas en línea más destacada, como la apilar íconos, cambiar el tamaño, estilizar con sombras y ajustar otros atributos visuales, además, es una elección predilecta por millones de diseñadores, desarrolladores y creadores de contenido en la web.

4. Metodología

4.1. Caso De Estudio

La Maestría en Geomática, que forma parte de la oferta académica de la Universidad del Cauca, fue creada mediante el Acuerdo 065 del 9 de diciembre de 2015, expedido por el Consejo Superior de la Universidad del Cauca, con Registro Calificado otorgado por medio de la Resolución N° 5574 del 29 de marzo por el Ministerio de Educación Nacional (MEN) y se encuentra registrada bajo el código SNIES 105455. Hasta la fecha, la maestría ha completado dos promociones: el Cohorte 1 y el Cohorte 2, cuyos programas académicos se iniciaron en los años 2017 y 2019, respectivamente (Calero, C., 2021, p. 11).

Mediante la Resolución No. 011559 del 23 de junio de 2022, el Ministerio de Educación Nacional, concedió la renovación del registro calificado en la modalidad de Investigación, por un período de siete (7) años a la Maestría en Geomática. La denominación del título que otorga este programa es “Magíster en Geomática”, y se ofrece en modalidad presencial, con un total de 48 créditos académicos, enmarcada en la categoría de 'Maestría en Investigación'.

Este programa, adscrito a la Facultad de Ingeniería Civil, consta de cuatro semestres de estudio y tiene como principal objetivo capacitar a profesionales en el diseño y desarrollo de aplicaciones en el ámbito de la ciencia de Sistemas de Información Geográfica (SIG), fotogrametría, bases de datos y otras disciplinas afines. Este enfoque permite a los participantes trabajar eficazmente en equipos multidisciplinarios dedicados a la gestión de información espacial. La Tabla 4 se describe el plan general de estudios.

Tabla 4. Plan general de estudios

ASIGNATURAS	Obligatorio	Electivo	CREDITOS ACADÉMICO	HORAS DE TRABAJO ESTUDIANTE			AREAS O COMPONENTES DE FORMACION DEL				Numero maximo de matriculados
				Horas de trabajo directo	Horas de trabajo independiente	Horas de trabajo TOTAL	Basica	Especifica	Investigación	Complementaria	
SEMESTRE I											
Geomática general	x		4	152	40	192	X			15	
Análisis y razonamiento espacial	x		4	156	40	196	X			15	
Sensores remotos	x		4	164	32	196	X			15	
SEMESTRE II											
Bases de datos espaciales	x		4	138	58	196		X		15	
Estadística espacial	x		4	156	40	196		X		15	
Seminario Investigativo I	x		4	152	44	196			X	15	
SEMESTRE III											
Electiva 1		x	2	68	28	96				X 15	
Electiva 2		x	2	68	28	96				X 15	
Electiva 3		x	2	68	28	96				X 15	
Seminario Investigativo II	x		4	152	44	196			X	15	
SEMESTRE IV											
Electiva 4		x	2	68	28	96				X 15	
Trabajo de grado		x	12	0	576	576			X	15	
Total Número Horas				1342.0	986.0	2328.0					
Total Porcentaje Horas (%)				57.6	42.4	100					
Total Número Créditos del Programa	28	22	48				12	8	20	8	
Total Porcentaje Créditos (%)	58.3	45.8	100				24.5	16.3	40.8	16.3	

Tomado de Programa de Maestría en Geomática modalidad Investigación (Calero, C., 2021, p.24)

A lo largo de las dos cohortes, se ha observado la participación de 10 estudiantes en la primera cohorte y 7 estudiantes en la segunda. Los detalles de la información de los estudiantes de la Maestría en Geomática se encuentran detallados en la Tabla 5.

Tabla 5. Estudiantes de la maestría por cohorte

Cohorte	No	Identificación	Nombres completos	Email	Movil	Fecha Inicio Est.
1	1	1.061.701.606	Jorge Armando Alpala Aguilar	jorgealpala@unicauca.edu.co		10/02/2017
	2	1.085.264.525	Yuli Yanira Erazo Montenegro	yulierazo@unicauca.edu.co		10/02/2017
	3	34.327.979	Lady Susana Montenegro Arboleda	lsmontenegro@unicauca.edu.co		10/02/2017
	4	1.061.691.327	Luis Miguel Narvaez Vanegas	lmnarvaez@unicauca.edu.co		10/02/2017
	5	76.309.414	Hernán Nope Rodriguez	hernannope@unicauca.edu.co	3232253939	10/02/2017
	6	12.991.122	Yovany Eduardo Unigarro Santacruz	yunigarro@unicauca.edu.co		10/02/2017
	7	1.085.284.681	Angela Nataly Dorado Calvache	angeladorado@unicauca.edu.co		10/02/2017
	8	1.149.686.102	Claudia Ximena Angola Zapata	claudiangola@unicauca.edu.co	3174649263	10/02/2017
	9	1061.754.391	Jhon Felipe Capote Mosquera	jfelipecapote@gmail.com		10/02/2017
	10	38.361.276	Diana Johanna Rojas Guayabo	dianjarojas@gmail.com	3152055282	10/02/2017
2	1	1.085.264.954	Maria Fernanda Arteaga Rodriguez	mariaarteaga@unicauca.edu.co		8/02/2019
	2	1.075.290.156	Juan Camilo Rojas Lucero	juanrl@unicauca.edu.co		8/02/2019
	3	1.061.689.009	John Fander Higido Castro	jhonhigido@unicauca.edu.co	3104403589	8/02/2019
	4	79.886.737	Juan Pablo Castro Hernandez	juanpcastro@unicauca.edu.co	3103595240	8/02/2019
	5	1.061.769.755	Javier Alejandro Calvo Anacona	javiercalvo@unicauca.edu.co	3104466935	8/02/2019
	6	34.565.569	Clara Stella Jojoa Tobar	clarajojoa@unicauca.edu.co		8/02/2019
	7	1.007.312.059	Luz Alejandra Rojas Calvache	luzr@unicauca.edu.co	3146103190	8/02/2019

Tomado de Programa de Maestría en Geomática modalidad Investigación

4.2. Diseño Metodológico

Este proyecto se enmarca en un trabajo de desarrollo de software basado en la metodología Scrum, donde, el objetivo principal es implementar una base de datos dinámica con enfoque C.R.U.D para la gestión de información académica, la adopción de la metodología Scrum no solo representa un marco de trabajo ágil para la ejecución del proyecto, sino que también ofrece una plataforma versátil que facilita la integración de diversos procesos y técnicas. Scrum, además, fomenta un entorno propicio para mejoras continuas tanto en el producto final como en la dinámica de trabajo del equipo.

Asimismo, la metodología Scrum adopta un enfoque iterativo en el cual cada iteración recibe el nombre de "Sprint": En este contexto, se establecieron doce (12) Sprint distribuidos a lo largo de seis (6) meses, la brevedad temporal es un objetivo clave para los Sprint, buscando la ejecución más eficiente de las tareas. Los Sprint están integralmente ligados a las fases de planificación, ejecución y pruebas; la metodología emplea reuniones con límites de tiempo para planificar, inspeccionar, corregir y adaptar durante cada sprint de trabajo.

La Tabla 6 muestra el modelo empleado para estructurar el diseño de cada sprint, junto con el período asignado para llevar a cabo cada actividad, es importante destacar que este lapso es flexible y puede ajustarse según el ritmo de trabajo. Al concluir cada sprint, se presentan de manera transparente los resultados y avances obtenidos durante su ejecución.

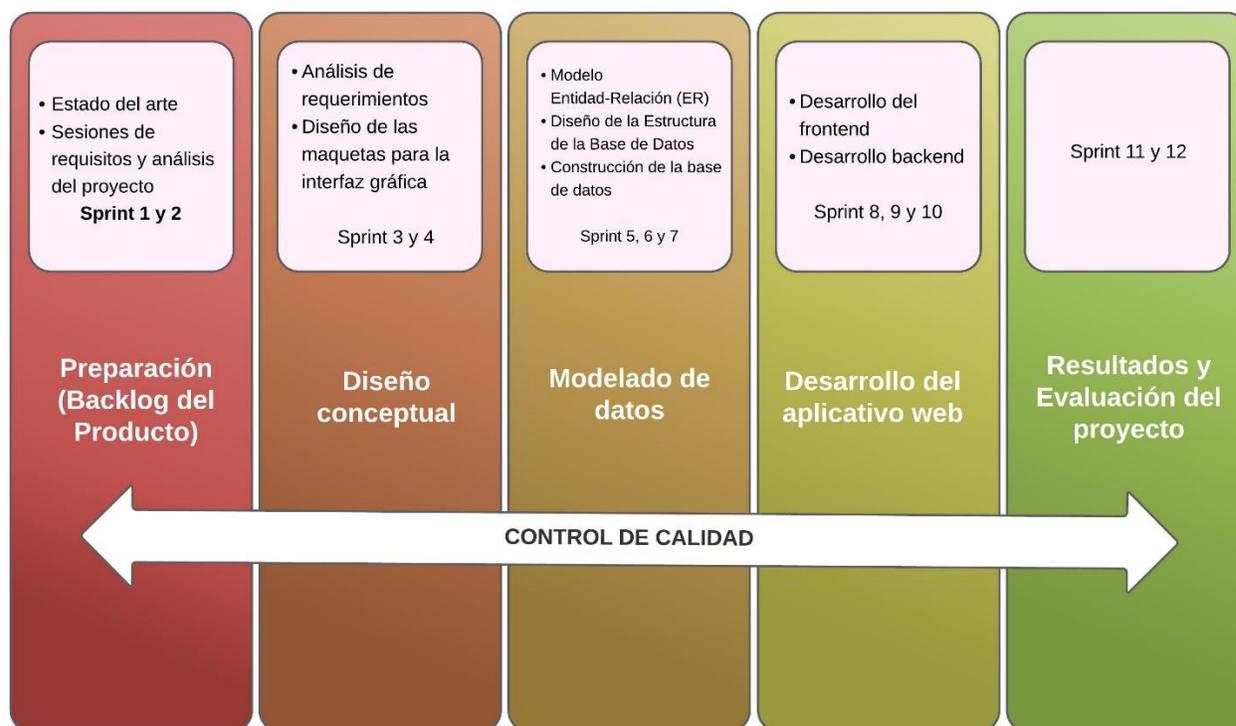
Tabla 6. Modelo de diseño de cada Sprint

SPRINT N°:		
ITEM	ACTIVIDADES	ESTIMACION TIEMPO (HORAS)
1		
2		
3		
TOTAL h/semana		20

Fuente: propia

La Figura 11 presenta un modelo que detalla cada iteración o sprint del proyecto. En cada uno de estos Sprint, como se mencionó anteriormente, se lleva a cabo un proceso de trabajo iterativo con la meta de proporcionar un resultado integral para el producto final, y realizar un control de calidad constante

Figura 11. Gestión de actividades Scrum



Fuente: propia

4.2.1. Preparación (Backlog Del Producto)

Principalmente es fundamental comenzar con una sólida comprensión de los temas relevantes y las necesidades específicas del mismo, los primeros pasos incluyen la identificación de temas clave, la búsqueda de información relevante a través de diversas fuentes y la síntesis de esta información para establecer una base sólida. Además, es crucial llevar a cabo sesiones de requisitos y análisis del proyecto, para garantizar una comprensión integral de las necesidades

y el alcance del proyecto. Este proceso de trabajo metodológico es esencial para establecer una dirección clara y eficiente en la ejecución del proyecto.

Sprint 1: Estado del arte

En el proceso de desarrollo del estado del arte, se prioriza la identificación y comprensión de los temas relevantes, ya que estos servirán como el fundamento esencial para la ejecución del proyecto. Esta fase inicial de la metodología de investigación involucra una serie de tareas críticas que son fundamentales para establecer una base sólida y orientar adecuadamente la investigación. Con este propósito, se llevan a cabo una serie de actividades diseñadas para lograr una comprensión más profunda de los temas a tratar.

Identificación de temas relevantes: esta actividad requiere una comprensión profunda de los objetivos de la investigación y la identificación de los temas que se relacionan con ellos.

Búsqueda bibliográfica: esta actividad implica la exploración de libros, artículos académicos, tesis, informes técnicos y otros recursos relacionados con los temas identificados.

Uso de Bases de Datos Académicas y Motores de Búsqueda: esta actividad es fundamental para acceder a la literatura científica actualizada y pertinente. Herramientas como Google Scholar, Web of Science, entre otros., facilitan la búsqueda y recuperación de publicaciones académicas relevantes.

Síntesis de Información: Una vez que se ha recopilado la bibliografía, es crucial sintetizar la información de manera organizada y coherente. Esto implica la clasificación y organización de los hallazgos clave y la preparación para la redacción del estado del arte.

Sprint 2: Sesiones de requisitos y análisis del proyecto.

A través de entrevistas y reuniones estructuradas con el coordinador de la Maestría en Geomática, se ha logrado obtener una visión detallada de las necesidades y expectativas del proyecto. Adicionalmente, al revisar las hojas de cálculo en Excel utilizadas por el personal de la oficina de coordinación, se ha adquirido una comprensión integral del entorno operativo y de

la gestión de tareas específicas dentro del programa, incluyendo el almacenamiento de información sobre estudiantes y docentes.

Finalmente, tras la recopilación de toda la información obtenida en cada reunión, se procede a redactar las historias de usuario. Estas historias serán fundamentales en el proceso de diseño y desarrollo del proyecto, ya que representarán los elementos clave que guiarán la implementación de las funcionalidades requeridas.

Estas historias de usuario sirven para formalizar la definición del proyecto. Asimismo, se desglosan las tareas que conforman cada historia de usuario, se les asigna un tiempo estimado y se establece su prioridad. Esto da como resultado la formación del Product Backlog, que constituirá la base del proyecto a desarrollar.

En total, 12 historias de usuario, las cuales se definirán utilizando el modelo de la Tabla No. 7.

Tabla 7. Modelo de la historia de usuario

HISTORIA DE USUARIO		
Título	Prioridad :	Estimación: () horas
Objetivo:		
Descripción:	Quien: (rol)	
	Funcionalidad:	
	Tareas:	
Criterio aceptación		

Fuente: propia

Cada campo se define de la siguiente manera (Menzinsky et al., 2022, p.9):

Título: Se trata de un título breve que se utiliza para proporcionar una descripción concisa de la historia de usuario.

Prioridad: Se refiere a la prioridad que se otorga al desarrollo de la historia de usuario en comparación con otras. *Valores: Alta, media y baja.*

Estimación: Se refiere a la asignación de un valor o medida que indica el esfuerzo o el tiempo requerido para completar la tarea o funcionalidad descrita en la historia.

Objetivo: Se refiere a proporcionar una descripción clara y concisa de una funcionalidad o característica que el usuario final del software necesita o desea.

Descripción: Se proporciona una breve explicación de lo que se está intentando lograr o de la funcionalidad que se desea implementar en la historia de usuario.

Quien: Se especifica qué tipo de usuario o rol se beneficiará o estará involucrado en la funcionalidad descrita en la historia.

Funcionalidad: Se detalla la funcionalidad o característica que se desea desarrollar

Tareas: Se refiere a las acciones específicas que deben llevarse a cabo para implementar la funcionalidad descrita.

Criterio de aceptación: Se refiere a las condiciones específicas que deben cumplirse para que la historia de usuario se considere terminada y aceptable.

4.2.2. Diseño conceptual

Sprint 3: Análisis de requerimientos

En esta fase, se creó un diagrama de flujo utilizando la plataforma Lucid chart con el fin de proporcionar al coordinador de la Maestría en Geomática una comprensión detallada del funcionamiento del sistema final. Este diagrama abarca tanto el proceso de acceso al aplicativo web como la gestión de la base de datos interna, lo que facilita la visualización y la comunicación de la funcionalidad y la interacción del sistema de manera clara y efectiva. Este enfoque visual es fundamental para la toma de decisiones informadas y para validar los requisitos del proyecto, garantizando que el sistema satisfaga las expectativas y necesidades del usuario. Además, permite identificar áreas de mejora o refinamiento en el diseño del sistema antes de la implementación, lo que conlleva a un ahorro de tiempo y recursos significativo.

Inicialmente, se organizan los requisitos en categorías distintas, separando los requisitos funcionales (que describen las capacidades del sistema) de los requisitos no funcionales (que especifican atributos de calidad, como rendimiento o seguridad). Esta clasificación permitirá una comprensión más clara de los requisitos y facilitará la gestión a lo

largo del proyecto. Los requerimientos funcionales y no funcionales se definirán utilizando el modelo de la Tabla 8.

Tabla 8. Modelo de requerimientos funcionales y no funcionales

ID. Requisito	
Usuario	
Prioridad	
Nombre del requisito	
Descripción	

Fuente: propia

Cada requisito tiene un identificador único, puede ser un número, una combinación de letras y números, el usuario que corresponde a quien se beneficiará del requisito, la prioridad se refiere a cuán crítico es el requisito para el proyecto, alta, media o baja, el nombre del requisito se refiere a un nombre breve y descriptivo que resume el requisito en pocas palabras y la descripción proporciona detalles más completos y específicos sobre el requisito, debe explicar en profundidad qué se espera lograr con el requisito, cómo se relaciona con otras partes del sistema y cualquier información adicional que sea relevante.

Sprint 4: Diseño de las maquetas para la interfaz gráfica

En el transcurso del proceso de diseño de interfaces de usuario, se emplea de manera esencial la plataforma Moqups para la creación y visualización de las interfaces que darán vida a la aplicación web. Moqups es una herramienta que posibilita la generación de prototipos de alta fidelidad, los cuales capturan con precisión tanto la estructura como la apariencia de las interfaces de usuario. Esto abarca desde la disposición de elementos en pantalla hasta la navegación entre diferentes secciones, incorporando la representación visual de las interacciones previstas para los usuarios.

La versatilidad de Moqups permite la creación de wireframes y maquetas interactivas, aspectos fundamentales que simplifican la comunicación y la comprensión de las ideas de diseño, tanto entre los miembros del equipo de desarrollo como entre los interesados y partes involucradas en el proyecto. El aspecto colaborativo en tiempo real de esta plataforma facilita la revisión constante y la obtención de retroalimentación, lo que resulta en mejoras y refinamientos de las interfaces antes de que se proceda a la fase de implementación.

4.2.3. Modelado De Datos

En el contexto del diseño de la base de datos, el modelado de datos es un componente fundamental de la metodología. Este proceso es esencial, ya que conlleva la representación estructurada y organizada de la información que será empleada en el sistema. A través de la aplicación de técnicas y herramientas específicas, se logra capturar, definir y analizar de manera sistemática los datos que posteriormente serán almacenados y procesados. Este enfoque metodológico proporciona la base para una implementación efectiva y eficiente de la estructura de la base de datos, permitiendo una gestión coherente y optimizada de la información en el contexto del sistema que se está desarrollando (Guerrero, 2014, p. 11).

Sprint 5: Modelo Entidad-Relación (Er)

Primordialmente, se inicia con la identificación de entidades y sus interrelaciones. Cada entidad surge a partir de las especificaciones de requisitos y está dotada de atributos que detallan sus características. Estos atributos se definen de manera precisa, representando detalles específicos que se almacenan y manipulan en el sistema. Además, se establecen elementos fundamentales como claves primarias, foráneas, vistas, índices, entre otros.

Una vez que todos los conceptos han sido identificados y definidos, se procede al diseño del correspondiente diagrama entidad-relación. Este proceso se realiza de manera eficiente a través de la plataforma Lucid Chart, la cual cuenta con símbolos de entidades y relaciones en la barra de herramientas. Facilita la incorporación de campos o atributos, y brinda

la flexibilidad de ajustar el formato, color y tamaño de los elementos para crear una representación visual clara y precisa del modelo de datos.

Sprint 6: Diseño De La Estructura De La Base De Datos En PHPmyAdmin

El diseño de la estructura de la base de datos comienza con la instalación de XAMPP, un componente esencial para el desarrollo del proyecto. Una vez instalado, se accede a PHPmyAdmin, donde se lleva a cabo la administración de la base de datos. En esta plataforma, se inicia creando la base de datos al ingresar su nombre, seleccionar un conjunto de caracteres y una colaboración. Posteriormente, se procede a la creación de tablas, definiendo la cantidad de columnas y gestionando los datos en estas tablas. Para cada columna, se especifica el nombre, tipo de dato, y otras propiedades según sea necesario, permitiendo la adición de índices, claves primarias y configuraciones avanzadas. Este proceso integral garantiza una estructura de base de datos sólida y adaptada a los requisitos específicos del proyecto.

Sprint 7: Construcción De La Base De Datos

La construcción de la base de datos implica la definición de relaciones entre tablas, facilitando así la integración y la interpretación de la información. Esto se llevará a cabo mediante consultas SQL basadas en el diseño previamente establecido, las cuales permitirán definir y establecer las restricciones necesarias para mantener la integridad y la coherencia de los datos. Esta fase se enfocará en asegurar que la estructura de la base de datos sea sólida y esté adecuadamente configurada para respaldar el análisis y la interpretación de los resultados de la investigación de manera precisa y confiable.

4.2.4. Desarrollo Del Aplicativo Web

Sprint 8: Desarrollo Back-end

En el desarrollo del back-end utilizando el patrón Modelo-Vista-Controlador (MVC), se inicia con la implementación del modelo, donde se definen las estructuras de datos y la funcionalidad central de la aplicación en PHP. Posteriormente, se configura y utiliza

PHPMYAdmin para gestionar la base de datos, con consultas SQL ejecutadas desde el modelo para interactuar con la base de datos y realizar operaciones de manipulación de datos. La vista se encarga de la presentación de la información al usuario y puede hacer uso de JavaScript para realizar algunas validaciones en el servidor, asegurando la integridad y seguridad de los datos. Finalmente, el controlador actúa como un intermediario entre el modelo y la vista, procesando las solicitudes del usuario, realizando operaciones en el modelo y pasando los datos a la vista para su presentación. Este enfoque modular y organizado del MVC facilita la escalabilidad, mantenibilidad y extensibilidad del back-end de la aplicación.

Sprint 9: Desarrollo Del Front-End

El desarrollo del frontend comienza con la construcción de la estructura básica en HTML, organizando el contenido con elementos semánticos. Luego, se aplica CSS para definir la apariencia visual y asegurar un diseño responsivo que funcione en diferentes dispositivos, utilizando frameworks como Bootstrap para acelerar el proceso. Posteriormente, se implementa funcionalidad interactiva con JavaScript, añadiendo validación de formularios y menús desplegables.

Sprint 10: Control De Calidad

El control de calidad en el back-end se enfoca en asegurar que tanto el código como la funcionalidad del lado del servidor cumplan con los estándares de calidad y los requisitos específicos del proyecto. Esto se logra mediante la realización de pruebas unitarias para cada componente individual y pruebas de integración para verificar la correcta interacción entre estos componentes, así como su integración con otros módulos y servicios externos. Además, se llevan a cabo revisiones de código para identificar posibles problemas y garantizar la coherencia y claridad del código.

En cuanto al control de calidad en el frontend, el enfoque principal recae en garantizar que la interfaz de usuario y la experiencia del usuario satisfagan los requisitos del proyecto,

esto implica realizar pruebas exhaustivas en diversos navegadores y dispositivos para asegurar la adecuada visualización y funcionamiento de la aplicación web en todas las plataformas.

También se realizan pruebas de usabilidad para evaluar la facilidad de uso y la accesibilidad de la interfaz, junto con pruebas de rendimiento para garantizar una respuesta fluida. Además, se llevan a cabo análisis de código y revisiones de diseño para identificar posibles problemas y asegurar la coherencia y eficiencia del código frontend.

4.2.5. Resultados y evaluación del proyecto

Sprint 11 y 12:

Esta fase, se centrará en la entrega del producto final, que en este caso es el aplicativo web diseñado para gestionar la información académica de la Maestría en Geomática de la Universidad del Cauca. El producto final se someterá a un riguroso análisis por parte de un comité académico experto, quienes evaluarán minuciosamente su funcionalidad, usabilidad y adecuación a los requerimientos del programa de maestría, esta revisión por parte del comité garantizará que el aplicativo cumpla con los estándares de calidad académica y contribuya de manera efectiva al proceso de gestión de la información.

Además, se llevará a cabo una encuesta entre los miembros del comité académico para recopilar sus opiniones y sugerencias sobre el aplicativo. Esta retroalimentación directa permitirá identificar áreas de mejora y posibles funcionalidades adicionales que podrían mejorar la experiencia del usuario y la utilidad del aplicativo en el contexto específico de la Maestría en Geomática, además, este proceso de evaluación servirá para prestar atención a la identificación y solución de errores (bugs) en la interfaz de usuario. La siguiente figura, muestra el formato usado para la aplicación de las encuestas.

Figura 12. Formato aplicación encuesta para evaluadores.

ENCUESTA APLICATIVO WEB					
<i>Gracias por participar en esta encuesta. Su opinión es muy importante para evaluar la efectividad del aplicativo web diseñado para el manejo de la información académica del programa de maestría en Geomática. Por favor, indica con una X tu grado de acuerdo con las siguientes afirmaciones utilizando la escala de Likert:</i>					
APLICATIVO WEB	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo, ni desacuerdo	De acuerdo	Totalmente de acuerdo
<i>Es amigable con el usuario</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Las funciones son claras y funcionales.</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Permite realizar las acciones CRUD.</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Permite gestionar información del trabajo de grado.</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Permite la gestión de errores</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Comentarios Adicionales

1. ¿Qué sugerencias tiene usted para mejorar el aplicativo?

Fecha: Día ____ Mes ____ Año _____

Nombre: _____

Afiliación institucional: _____

Dependencia: _____

¡La información que proporcionas en esta encuesta será utilizada exclusivamente para fines académicos y será tratada de manera confidencial!

*Gracias por su participación.
Tu retroalimentación nos ayudará a mejorar nuestro aplicativo.*

Fuente: propia

5. Resultados y Discusión

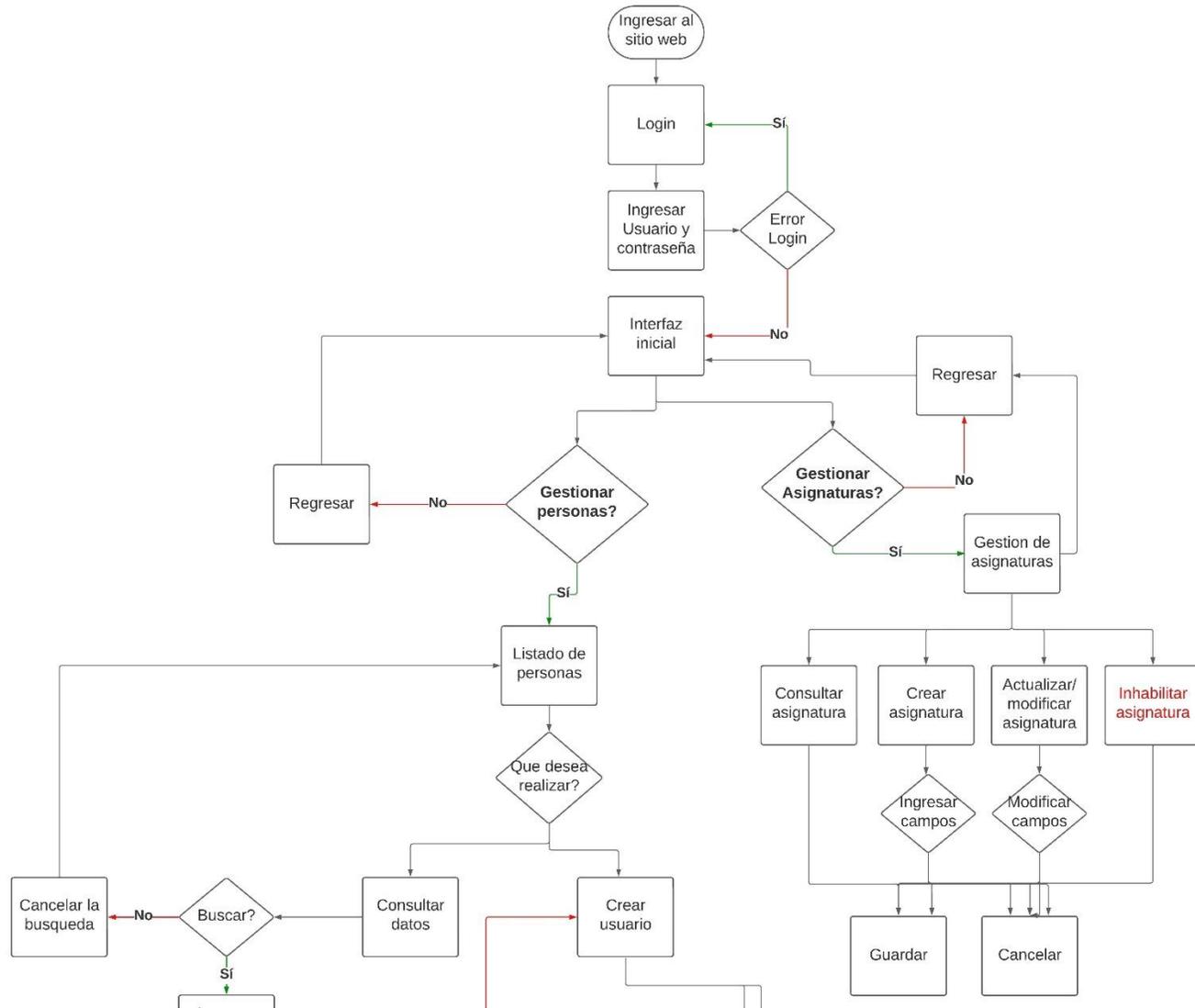
5.1. Recopilar, revisar, clasificar y sistematizar la información existente del programa de Maestría en Geomática de la Universidad del Cauca.

Durante este proceso, se llevó a cabo una búsqueda y evaluación de una amplia variedad de fuentes de información, que abarcó desde documentos oficiales hasta el uso de bases de datos académicas y motores de búsqueda para acceder a literatura relevante y otros recursos pertinentes. Esta recopilación permitió obtener una comprensión completa y detallada de los aspectos fundamentales del proyecto a ejecutar.

5.1.1. *Sesiones de requisitos y análisis de requerimientos*

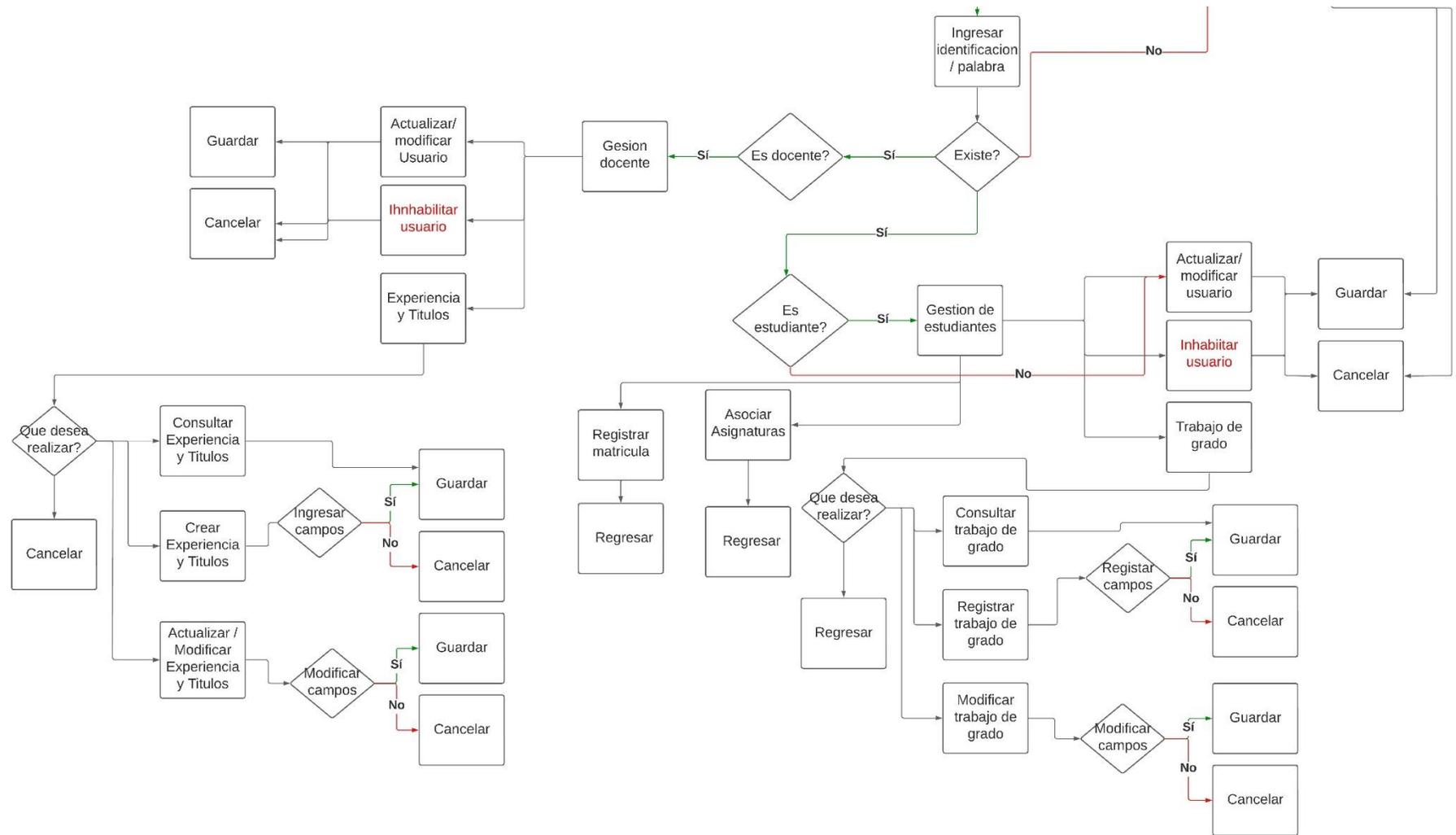
Además de la recopilación de información, se llevaron a cabo sesiones de requisitos y análisis del proyecto con el objetivo de identificar las necesidades y expectativas del programa de Maestría en Geomática. Durante estas sesiones, se realizaron entrevistas y reuniones estructuradas con el coordinador del programa y otros interesados relevantes, este proceso permitió identificar de manera precisa los requisitos funcionales y no funcionales del sistema de gestión de información académica, así como comprender los desafíos y oportunidades que enfrenta el programa. Las siguientes figuras muestran el resultado de las sesiones de requisitos y análisis de requerimientos aprobados por el coordinador.

Figura 13. Sesiones de requisitos y análisis de requerimientos – Parte 1.



Fuente: propia

Figura 14. Sesiones de requisitos y análisis de requerimientos – Parte 2.



Fuente: propia

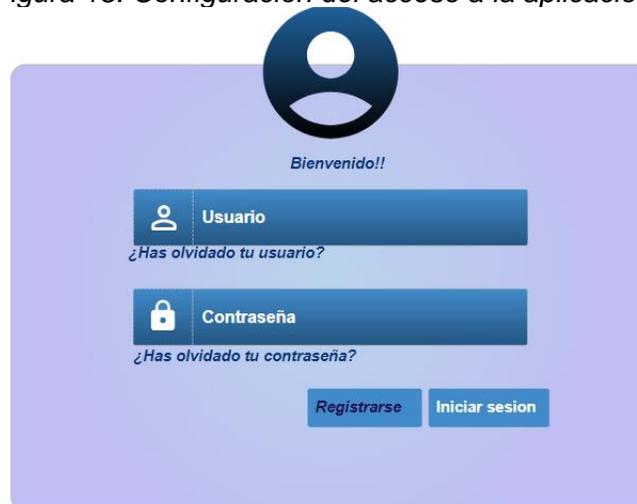
5.1.2. Diseño de las maquetas para la interfaz gráfica

De acuerdo con los requisitos recopilados y analizados, se procedió al diseño de las maquetas para la interfaz gráfica del sistema. Estas maquetas fueron desarrolladas con el objetivo de visualizar de manera clara y concisa las funcionalidades y características del sistema, asegurando su alineación con las necesidades y expectativas identificadas durante el análisis de requisitos. Mediante la representación visual de la interfaz gráfica, se buscó proporcionar una visión tangible sobre cómo se vería y funcionaría el sistema en su versión final, facilitando así la comunicación y la retroalimentación con los interesados del programa de Maestría en Geomática. Dada la variabilidad de páginas dentro de la aplicación, se han especificado los cambios entre ellas para asegurar una distinción clara y eliminar posibles confusiones para el usuario. A continuación, se detallan las características de cada interfaz de usuario.

Login

En la página de “login” (Figura 15) se puede apreciar en el cuerpo los dos campos que se necesitan para empezar a utilizar la aplicación.

Figura 15. Configuración del acceso a la aplicación



Fuente: propia

En caso de olvidar usuario y/o contraseña, se muestra lo siguiente.

Figura 16. Recuperar usuario y/o contraseña

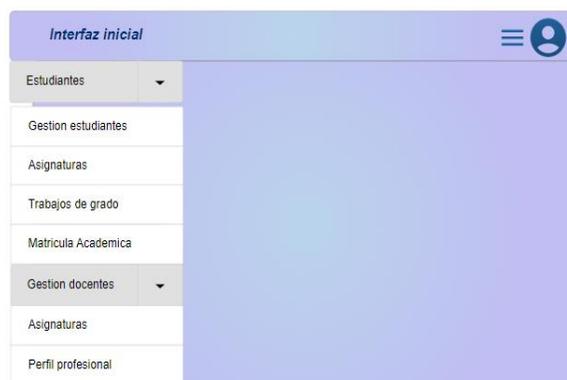


Fuente: propia

Página Principal

La página principal (Figura 17) muestra el menú principal. En la parte del cuerpo se encuentran todos los menús por los cuales se puede navegar. En la parte superior derecha se encuentra el acceso a la información del usuario y el cierre de sesión de la aplicación al salir de la misma.

Figura 17. Vista de la interfaz inicial



Fuente: propia

Formularios

Los formularios tienen en la cabecera el nombre del menú donde se encuentra y el nombre del usuario. En el cuerpo se encuentran los campos a rellenar (Figura 18).

Figura 18. Modelos de formularios

Interfaz editar Trabajos de grado - estudiante

Titulo trabajo

Resolucion aprobacion

Fecha expedicion

Fecha sustentacion

Director trabajo

Interfaz gestion docentes

Estudiantes

Gestion estudiantes

Asignaturas

Seleccionar	Id_docentes	Documento	Nombres	Apellidos	Editar
<input checked="" type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Gestion docentes

Asignaturas

Perfil profesional

Interfaz editar docente

Nombres

Apellidos

Tipo de identificacion

Identificacion

Interfaz asignatura - docente

Estudiantes

Gestion estudiantes

Asignaturas

Seleccionar	Id. asignatura	Nombre asignatura	Creditos	Tipo asignatura	Editar
<input checked="" type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Gestion docentes

Asignaturas

Perfil profesional

Interfaz crear asignatura - docente

Id. asignatura

Nombre asignatura

Creditos

Tipo de asignatura

Interfaz editar asignatura - docente

Id. asignatura

Nombre asignatura

Creditos

Tipo de asignatura

Interfaz hv. docente - docentes

Estudiantes

Gestion estudiantes

Asignaturas

Seleccionar	Exp. docente	Ex. Profesional	CvIac	Correo	Editar
<input checked="" type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Gestion docentes

Asignaturas

Perfil profesional

Interfaz crear H.V. - docente

Exp. docente

Ex. profesional

CvIac

Correo

Fuente: propia

Interfaz crear y editar estudiantes

Estudiantes

Gestion estudiantes

Seleccionar	codigo estudiante	Documento	Nombres	Apellidos	Editar
<input checked="" type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Interfaz crear estudiante

Nombres

Apellidos

Tipo de identificacion

Identificacion

GUARDAR CANCELAR

Interfaz editar estudiante

Nombres

Apellidos

Tipo de identificacion

Identificacion

EDITAR GUARDAR CANCELAR

Interfaz asignatura - estudiantes

Estudiantes

Gestion estudiantes

Seleccionar	Id. asignatura	Nombre asignatura	Creditos	Tipo asignatura	Editar
<input checked="" type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					

Interfaz crear asignatura - estudiante

Id. asignatura

Materia

Creditos

Tipo de asignatura

GUARDAR CANCELAR

Interfaz editar asignatura - estudiante

Id. asignatura

Materia

Creditos

Tipo de asignatura

EDITAR GUARDAR CANCELAR

Interfaz Trabajos de grado - estudiantes

Estudiantes

Seleccionar	Titulo trabajo	Resolucion aprobacion	Fecha expedicion	Fecha sustentacion	Director trabajo	Editar
<input checked="" type="checkbox"/>						
<input type="checkbox"/>						
<input type="checkbox"/>						

Interfaz crear Trabajos de grado - estudiante

Titulo trabajo

Resolucion aprobacion

Fecha expedicion

Fecha sustentacion

Director trabajo

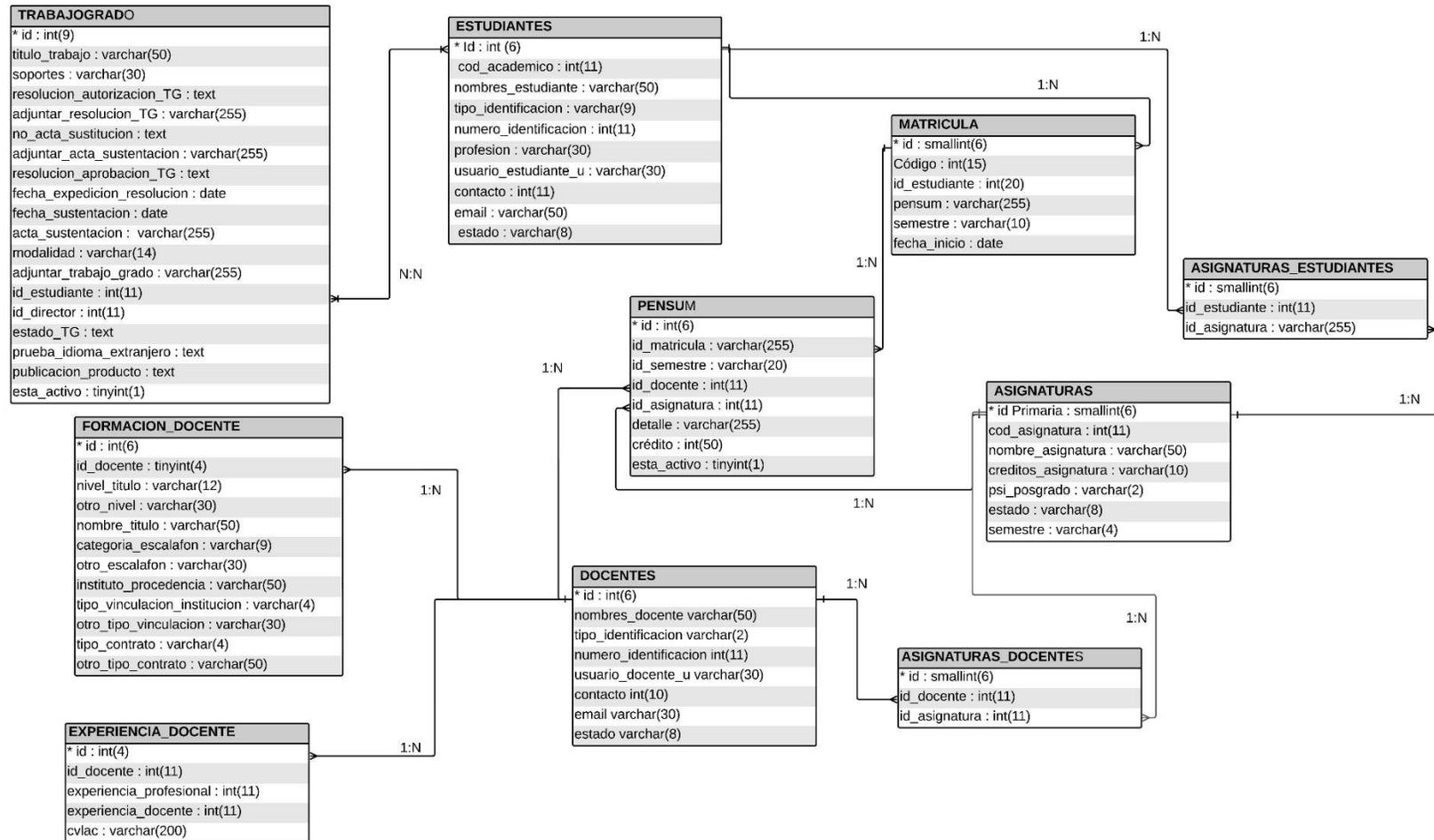
GUARDAR CANCELAR

5.2. Diseñar un modelo lógico mediante diagramas entidad/relación, es decir, a través del almacenamiento de datos en tablas (entidades) compuestas por filas (registros) y columnas (atributos).

Una vez obtenidos los requisitos del sistema y los datos recopilados durante las fases previas del proyecto, se llevó a cabo el diseño de un modelo lógico utilizando la metodología de diagramas Entidad/Relación. Este enfoque permitió representar de manera clara y estructurada las entidades principales del sistema, así como sus atributos y las relaciones entre ellas. Cada entidad fue identificada y definida con sus respectivos atributos, y se establecieron las relaciones necesarias para capturar la interacción entre las diferentes entidades. A continuación, se muestra un bosquejo inicial del diagrama Entidad/Relación, diseñada en Lucid Chart.

Posteriormente, el resultado fue un modelo lógico coherente y completo que sirvió como punto de partida para la implementación del sistema de gestión de información académica, asegurando una estructura de base de datos sólida y adaptada a los requisitos específicos del proyecto. La siguiente figura representa lo mencionado anteriormente.

Figura 20. Diagrama Entidad/Relación



Fuente: propia

5.3. Adaptar una base de datos en la web con credenciales de acceso para la gestión de la información del programa de Maestría en Geomática.

Se configuró la base de datos en PHPmyAdmin. La siguiente figura ofrece una visión detallada de la estructura de la base de datos, incluyendo las tablas diseñadas para integrarla con la aplicación web. Este paso facilita la gestión directa de la información del programa de Maestría en Geomática a través del aplicativo web.

Figura 21. Estructura base de datos PHPmyAdmin.

The screenshot shows the PHPmyAdmin interface for a database named 'sgtg1'. The left sidebar displays a tree view of the database structure, including 'Procedimientos' and 'Tablas'. The main area shows a table with columns: Tabla, Acción, Filas, Tipo, Cotejamiento, Tamaño, and Residuo a depurar. The table lists 20 tables with their respective actions and sizes.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
asignaturas	Examinar Estructura Buscar Insertar Vaciar Eliminar	12	InnoDB	utf8_spanish_ci	16.0 KB	-
asignaturas_docentes	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
asignaturas_estudiantes	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
categorias	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	16.0 KB	-
docentes	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-
estudiantes	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-
experiencia_docente	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-
formacion_docente	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-
matricula	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-
pages	Examinar Estructura Buscar Insertar Vaciar Eliminar	14	InnoDB	utf8mb4_general_ci	32.0 KB	-
pensum	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_spanish_ci	16.0 KB	-
permissions	Examinar Estructura Buscar Insertar Vaciar Eliminar	53	InnoDB	utf8mb4_general_ci	16.0 KB	-
roles	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
semestres	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	16.0 KB	-
tipoidentificacion	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_spanish_ci	16.0 KB	-
trabajogrado	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_spanish_ci	16.0 KB	-

Fuente: propia

El aplicativo web diseñado para la gestión de la información del programa de Maestría en Geomática se basa en la funcionalidad CRUD (Crear, Leer, Actualizar, Eliminar) para interactuar con la base de datos alojada en PHPmyAdmin. A través de esta conexión, los usuarios pueden acceder a una plataforma, la cual, permite realizar diversas acciones, como por ejemplo agregar nuevos registros de estudiantes o profesores (Crear), consultar información existente (Leer), actualizar datos de los trabajos de grado de los estudiantes (Actualizar) y eliminar registros innecesarios (Eliminar). A continuación, las siguientes figuras muestran el ingreso al aplicativo.

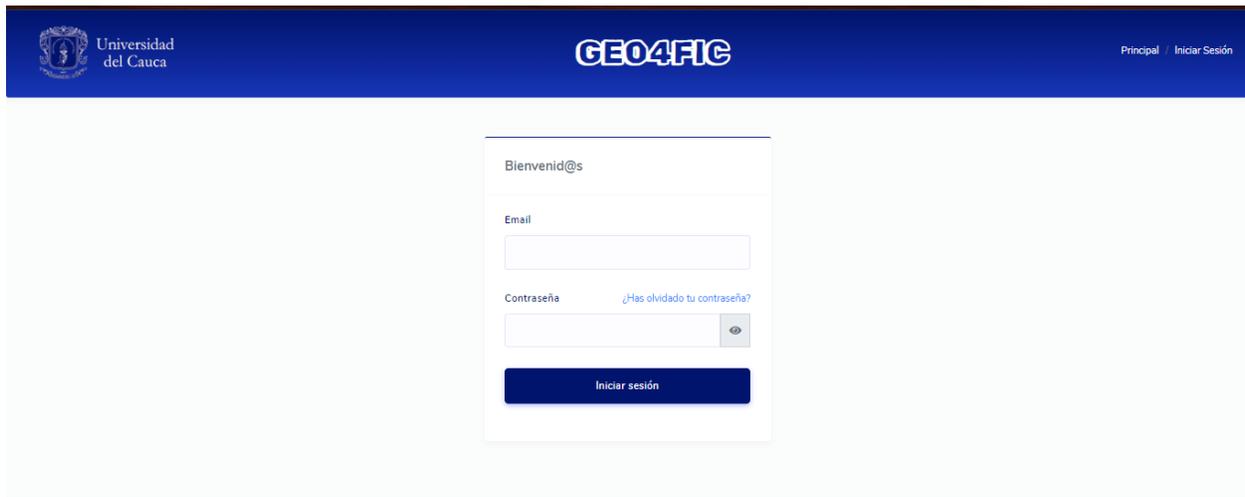
Para facilitar el uso del aplicativo web, se ha diseñado un manual detallado que explica paso a paso el proceso de navegación y las funcionalidades disponibles. Este manual, que se incluye como anexo al final de este documento, proporciona instrucciones claras para gestionar eficientemente la información del programa de Maestría en Geomática.

Figura 22. Página principal del sistema.

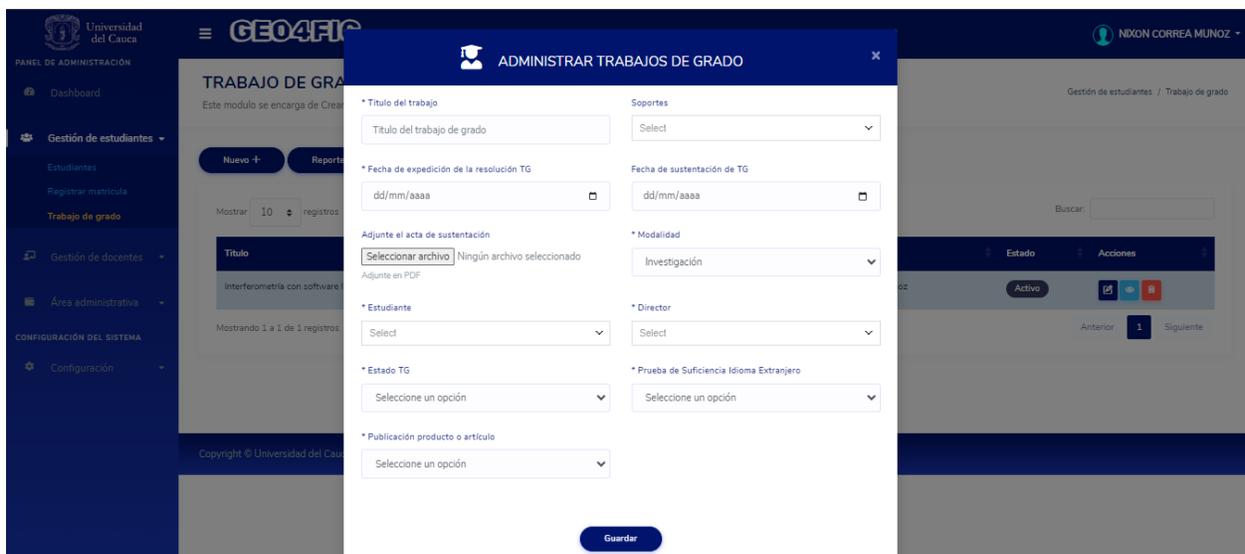


Fuente: propia

Figura 23. Login: Ingreso al aplicativo web.



Fuente: propia
Figura 24. Formulario final de trabajos de grado.



Fuente: propia

5.4. Validar la base de datos dinámica mediante una prueba piloto con los tomadores de decisión de los programas de posgrado.

Esta validación se llevó a cabo mediante encuestas diseñadas para evaluar la funcionalidad, usabilidad y efectividad de la base de datos. Se diseñaron y distribuyeron 4 encuestas a los tomadores de decisión de los programas de posgrado, para evaluar la

efectividad del aplicativo web diseñado para el manejo de la información académica, dichas encuestas son basadas en la escala de Likert. La encuesta incluía cinco afirmaciones sobre la usabilidad y funcionalidad del aplicativo, así como un espacio para comentarios adicionales y sugerencias.

Se analizaron las respuestas cuantitativas utilizando la escala de Likert, con opciones que iban desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo". A continuación, en la siguiente tabla se presentan los resultados obtenidos:

Tabla 9. Resultados encuesta.

CRITERIO	RESULTADOS
Amigable con el usuario:	<input type="checkbox"/> Totalmente en desacuerdo: 0% <input type="checkbox"/> En desacuerdo: 0% <input type="checkbox"/> Ni de acuerdo, ni desacuerdo: 0% <input type="checkbox"/> De acuerdo: 50% <input type="checkbox"/> Totalmente de acuerdo: 50%
Claridad y funcionalidad de las funciones:	<input type="checkbox"/> Totalmente en desacuerdo: 0% <input type="checkbox"/> En desacuerdo: 0% <input type="checkbox"/> Ni de acuerdo, ni desacuerdo: 0% <input type="checkbox"/> De acuerdo: 50% <input type="checkbox"/> Totalmente de acuerdo: 50%
Acciones CRUD (Crear, Leer, Actualizar, Eliminar):	<input type="checkbox"/> Totalmente en desacuerdo: 0% <input type="checkbox"/> En desacuerdo: 0% <input type="checkbox"/> Ni de acuerdo, ni desacuerdo: 0% <input type="checkbox"/> De acuerdo: 50% <input type="checkbox"/> Totalmente de acuerdo: 50%
Gestión de información del trabajo de grado:	<input type="checkbox"/> Totalmente en desacuerdo: 0% <input type="checkbox"/> En desacuerdo: 0% <input type="checkbox"/> Ni de acuerdo, ni desacuerdo: 0%

	<input type="checkbox"/> De acuerdo: 50% <input type="checkbox"/> Totalmente de acuerdo: 50%
Gestión de errores:	<input type="checkbox"/> Totalmente en desacuerdo: 0% <input type="checkbox"/> En desacuerdo: 0% <input type="checkbox"/> Ni de acuerdo, ni desacuerdo: 25% <input type="checkbox"/> De acuerdo: 50% <input type="checkbox"/> Totalmente de acuerdo: 25%

Fuente: propia

En los comentarios adicionales, los tomadores de decisión proporcionaron sugerencias específicas para mejorar el aplicativo. Los temas comunes incluyen:

- Apoyo en el proceso de acreditación y renovación de registro calificado.
- Adicionar recibos de pago, notas, resoluciones, etc.
- Involucrar el seguimiento de la propuesta en los Seminarios de Investigación.
- Involucrar los porcentajes correspondientes a las notas emitidas por los docentes (involucrar estadísticas).
- Gestión de encuestas por asignatura
- Gestión de calidad para la autoevaluación
- Guardar notas por el docente con firma y generales
- Diferenciar por modalidad estudiante regular y plan coterminal
- Certificado de votación

Los resultados de la encuesta indican que el aplicativo web es, en general, bien recibido por los tomadores de decisiones. La mayoría de los participantes estuvieron de acuerdo o totalmente de acuerdo con las afirmaciones sobre la usabilidad y funcionalidad del sistema. Esto sugiere que el diseño del aplicativo cumple con los requisitos y expectativas previamente definidas.

- **Amigable con el usuario:** La respuesta fue uniforme, con el 50% de los participantes "De acuerdo" y el 50% "Totalmente de acuerdo". Esto indica que el diseño del aplicativo es intuitivo y fácil de usar para los tomadores de decisión, lo que es esencial para la aceptación y uso continuo del sistema.
- **Usabilidad y Funcionalidad:** La unanimidad en las respuestas "De acuerdo" y "Totalmente de acuerdo" respecto a la amigabilidad con el usuario y la claridad y funcionalidad de las funciones indica un diseño centrado en el usuario.
- **Acciones CRUD:** La capacidad del sistema para realizar acciones CRUD fue bien valorada, lo que es crucial para la gestión dinámica de datos.
- **Gestión de Información del Trabajo de Grado:** La mayoría estuvo de acuerdo en que el sistema permite gestionar esta información adecuadamente.
- **Gestión de Errores:** Las respuestas sugieren que esta es un área donde se puede mejorar, con un cuarto de los encuestados indicando una postura neutral.

La prueba piloto a través de la encuesta demostró que la base de datos dinámica y el aplicativo web son efectivos para el manejo de la información académica. Los resultados positivos y las sugerencias de mejora proporcionadas por los tomadores de decisión validan la funcionalidad del sistema y ofrecen una guía para futuras mejoras.

6. Conclusiones

Como resultado de las etapas trabajadas para el diseño del aplicativo web, se pueden obtener las siguientes conclusiones:

- La recopilación, revisión, clasificación y sistematización de la información del programa de Maestría en Geomática de la Universidad del Cauca permitió una base para la construcción del aplicativo web. Se logró una comprensión integral de los requerimientos y expectativas del coordinador del programa, asegurando que el sistema se adapte a sus necesidades específicas.
- A través del diseño del modelo lógico utilizando diagramas entidad/relación, se estructuró una base de datos eficiente y bien organizada. Este modelo permitió un almacenamiento de datos coherente y accesible, facilitando la gestión de la información académica del programa de maestría.
- El desarrollo back-end, implementado principalmente con PHP, SQL y JavaScript, garantizó la funcionalidad y seguridad del sistema. Se logró una integración efectiva con la base de datos, permitiendo operaciones CRUD (Crear, Leer, Actualizar, Eliminar) y una gestión eficiente de la información.
- La prueba piloto con los tomadores de decisión mostró resultados positivos, destacando la amigabilidad del usuario y la funcionalidad clara del sistema. Las sugerencias recogidas proporcionaron una valiosa retroalimentación para futuras mejoras. La mayoría de los usuarios estuvo de acuerdo o totalmente de acuerdo en que el sistema permite realizar las acciones necesarias y gestionar la información académica de manera efectiva.
- Las encuestas revelaron áreas potenciales de mejora, como la inclusión de funcionalidades adicionales para apoyar procesos de acreditación, seguimiento de propuestas en seminarios de investigación, y gestión de encuestas y calidad; temas que en un principio no se tuvieron en cuenta, sin embargo, estas sugerencias indican la

necesidad de una evolución continua del aplicativo para adaptarse a las cambiantes necesidades académicas y administrativas.

- Finalmente, se concluye que el proyecto demostró ser viable y de gran impacto para la gestión académica del programa de maestría en Geomática. La base de datos dinámica y el aplicativo web desarrollados permiten una gestión eficiente de la información, mejorando los procesos administrativos y académicos.

7. Recomendaciones

Basado en los resultados obtenidos de la prueba piloto mediante encuestas dirigidas a los tomadores de decisiones de los programas, se derivan varias recomendaciones para mejorar la funcionalidad y la usabilidad del aplicativo web diseñado para la gestión de la información académica:

- Apoyo en procesos de acreditación y renovación de registro calificado, dado que los encuestados expresaron interés en esta área, se recomienda incorporar características que faciliten el seguimiento y la recopilación de información relevante para estos procesos administrativos.
- La inclusión de características como la capacidad para adjuntar recibos de pago, notas, resoluciones, entre otros documentos, se considera valiosa para complementar la utilidad del aplicativo.
- Implementar funcionalidades que permitan el registro detallado de notas por parte de los docentes, incluyendo firmas y datos generales del estudiante, podría mejorar la transparencia y la precisión de la información académica.
- Considerando las necesidades específicas de diferentes modalidades académicas, como regulares y plan coterminales, se sugiere desarrollar características que se adapten a cada perfil, proporcionando una experiencia personalizada.

Estas recomendaciones están destinadas a mejorar la funcionalidad y la usabilidad del aplicativo web, basadas en las sugerencias proporcionadas por los tomadores de decisiones durante la prueba piloto, su implementación podría contribuir significativamente a la eficiencia y efectividad del sistema.

8. Bibliografía

Álvarez, M., et al., (2001). Manual de JavaScript. Disponible en <https://blogsaverroes.juntadeandalucia.es/plataformaeiv/files/2016/09/manual-javascript.pdf>.

Álvarez, R., et al., (2016). Manual de PHP. Desarrolloweb.com. Disponible en <https://es.scribd.com/document/646369546/Manual-de-Php>.

Apache HTTP Server Project. (s.f.). Acerca de Apache.

https://httpd.apache.org/ABOUT_APACHE.html.

Alzate et al., (2016). Análisis y diseño de una aplicación web para gestionar eficiente y eficazmente la información en los procesos de calidad de servicio para la empresa SICTE. Universidad Tecnológica de Pereira.

<https://repositorio.utp.edu.co/server/api/core/bitstreams/533709e0-1434-46a4-900c-b5082e1aed6e/content>

Bone, K., (2019)., MVC Design Pattern Broken Down. Medium

https://medium.com/@k_bone/mvc-design-pattern-broken-down-d07df59bbd43

Bootstrap. (s.f.). Visión general. <https://getbootstrap.com/docs/5.3/about/overview/>

<https://fontawesome.com/docs>.

Calero, C., (2021). *Condiciones de Calidad -Programa de Maestría en Geomática modalidad Investigación*. Universidad del Cauca.

Cardona, H., et al., (2014) *Diseño e Implementación de Bases de Datos desde una Perspectiva Práctica*. 1a ed. - Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn).

[https://mountainscholar.org/bitstream/handle/20.500.11785/586/BookId-516-](https://mountainscholar.org/bitstream/handle/20.500.11785/586/BookId-516-DisenoImplement.pdf?sequence=1&isAllowed=y)

[DisenoImplement.pdf?sequence=1&isAllowed=y](https://mountainscholar.org/bitstream/handle/20.500.11785/586/BookId-516-DisenoImplement.pdf?sequence=1&isAllowed=y).

Cases, E., (2014). *Apache HTTP Server: ¿Qué es, ¿cómo funciona y para qué sirve?* Agencia Marketing Digital en Barcelona. <https://www.ibxagency.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.

Chapaval, N. (2017). *¿Qu as Frontend Y Back-end? Plats Blog*. <https://platzi.com/blog/que-es-frontend-y-back-end/>.

CEPSUNI (2021). *Diseño de base de datos relacionales*. Universidad Nacional de Ingeniera. Lima – Perú. <https://www.ceps.uni.edu.pe/wp-content/uploads/2021/01/AVANCE-SEPARATA-DESNO-DE-BASE-DE-DATOS-2021.pdf>.

Codd, E., (1970)., *A relational model of data for large shared data banks*. *Comunicaciones de la ACM, Volumen 13, Número 6, Junio de 1970, págs. 377-387*.
<https://doi.org/10.1145/362384.362685>

Deyimar A. (2023), *¿Qué es Bootstrap? – Una guía para principiantes*. Hostinger Tutoriales.
<https://www.hostinger.co/tutoriales/que-es-bootstrap>.

Espinosa, E., (2012). *IMPLEMENTACIÓN DE UN SISTEMA SERVICE DESK BASADO EN ITIL*. UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.
<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/2749/Tesis.pdf?sequence=1&isAllowed=y>.

Garcia, M., (2021). *FULLSTACK REST API: ANGULAR FRONTEND + JAVA (SPRING) BACK-END*. <https://gonzalogarciamr.com/2021/04/28/fullstack-rest-angular-spring/>

Gómez, C., (2013). *Bases de Datos*. Universidad Autónoma Metropolitana, Unidad Cuajimalpa.
[https://www.cua.uam.mx/pdfs/conoce/libroselec/Notas del curso Bases de Datos.pdf](https://www.cua.uam.mx/pdfs/conoce/libroselec/Notas%20del%20curso%20Bases%20de%20Datos.pdf).

Guerrero, M., (2014). UF1471: *Base de datos relacionales y modelado de datos*. España.
https://www.editorialelearning.com/catalogo/media/iverve/uploadpdf/1526023279_UF1471_demo.pdf.

Gutiérrez, J., (2017). *¿Qué es un framework web?*
http://www.lsi.us.es/~javier/investigacion_ficheros/Framework.pdf.

Luján, S., (2002). Programación de aplicaciones web: historia, principios básicos, y clientes web. San Vicente - Alicante. https://rua.ua.es/dspace/bitstream/10045/16995/1/sergio_lujan-programacion_de_aplicaciones_web.pdf.

Marqués, M., (2011). *Bases de datos*. Primera edición, 2011 ISBN: 978-84-693-0146-3 <https://bdigital.uvhm.edu.mx/wp-content/uploads/2020/05/Bases-de-Datos.pdf>.

Martínez, A., (1995). *MANUAL PRÁCTICO DE HTML*. Universidad Politécnica de Madrid, España. <http://bioinf.ibun.unal.edu.co/servicios/electiva/manhtml/HTML.pdf>.

MDN contributors, 2022., CSS. : <https://developer.mozilla.org/es/docs/Learn/CSS>.

MySQL. (s.f.). Documentación oficial. <https://dev.mysql.com/doc/>.

Oppel, A., et al., (2010). *Fundamentos de SQL*. México. https://pedrobeltrancaessa-biblioteca.weebly.com/uploads/1/2/4/0/12405072/fundamentos_de_sql_3edi_oppel.pdf.

Primo, A., (2012). *Protocolo HTTP*. Teoría de SRI. https://alvaroprimguijarro.files.wordpress.com/2012/01/ud04_http_alvaroprimguijarro.pdf.

PhpMyAdmin. (s.f.). <https://www.phpmyadmin.net/>

Ramírez, D., et al., (2015). Generador automático de CRUD para la LPS Aplicativos SIG. Universidad de las Ciencias Informáticas. La Habana. https://repositorio.uci.cu/bitstream/123456789/7443/1/TD_08281_15.pdf

Sánchez, J., (2009). *Apuntes completos Sistemas Gestores de Bases de Datos*. https://www.academia.edu/44415776/Sistemas_gestores_de_Bases_de_Datos

Schwaber, K., et al (2020). La Guía de Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>.

Valencia, E., 2013. Sistema Académico Web SAITICC.: <https://core.ac.uk/download/pdf/200329057.pdf>