



FACULTAD 2

Sistema de gestión de la información de los ejercicios de culminación de estudios en la Facultad 2

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

Autor(es): Guido Alejandro Morales Bordon
Ricardo Adrian Avila Rodriguez

Tutor: Ing. Antonio Hernández Domínguez

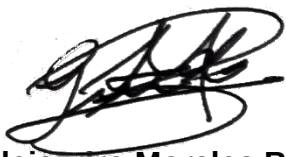
Cotutor: Ing. Thalía Maya Pérez

La Habana, diciembre de 2023

“Año 65 de la Revolución”

DECLARACIÓN DE AUTORÍA

Los autores del trabajo de diploma con título “**Sistema de gestión de la información de los ejercicios de culminación de estudios en la Facultad 2**” conceden a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declaran como únicos autores de su contenido. Para que así conste firman la presente a los 8 días del mes de diciembre del año 2023.



Guido Alejandro Morales Bordon

Firma del Autor



Ricardo Adrian Avi la Rodriguez

Firma del Autor



Firmado digitalmente por
Antonio Hernández Domínguez
Motivo: He revisado este
documento
Ubicación: UCI
Fecha: 2023.12.07 11:44:44
-05'00'

Antonio Hernández Domínguez

Firma del Tutor



Thalía Maya Pérez

Firma del Tutor

DATOS DE CONTACTO

Ing. Antonio Hernández Domínguez

Ingeniero en Ciencias Informáticas 2009 Universidad de las Ciencias Informáticas. Líder y Administrador de la Base de Datos de algunos proyectos informáticos en la UCI durante varios años. Actualmente trabaja como jefe de departamento de informática de la Facultad 2 de la UCI. Imparte docencia de pregrado en las siguientes asignaturas: “Matemática I”, “Matemática II”, “Debate Histórico Contemporáneo” y “Sistemas de Bases de Datos 1”. Ha participado en proyectos informáticos en la República Bolivariana de Venezuela. Jefe de Departamento de Informática Facultad 2. Teléfono +537-8358151.

Correo electrónico: ahdominguez@uci.cu

Ing. Thalía Maya Pérez

Ingeniera en Ciencias Informáticas 2021 Universidad de las Ciencias Informáticas. Especialista en Ciencias Informáticas del Centro de Gestión Documental de algunos proyectos informáticos en la UCI durante dos años. Teléfono +537-8358151.

Correo electrónico: ahdominguez@uci.cu

AGRADECIMIENTOS

Guido Alejandro Morales Bordon

Le agradezco a mis padres, en especial a mi mamá, por apoyarme siempre en cada momento que lo he necesitado. Es una mujer capaz de mover cielo y tierra para conseguir que cada contratiempo y necesidad sea cubierto sin importar las consecuencias. Sí, ya he terminado algo. Agradezco también a mi hermana por nunca faltar y por poner tanto afecto en cada situación en la que me ha ayudado, sin condiciones ni limitaciones. No puede faltar mi pareja en estos agradecimientos, ya que, a pesar de la distancia, nunca ha dejado de brindarme su paciencia y cariño, su apoyo incondicional por encima del tiempo y las tempestades, y por siempre estar presente a pesar de los inmensos problemas vividos y afrontados juntos. Muchas gracias de verdad. Agradezco a todas mis amistades de la universidad, haciendo énfasis en mis compañeros de cuarto. Ellos siempre estuvieron en cada minuto, en los buenos y malos momentos, capaces de acompañarme a pesar de todo, con charlas, consejos y hasta emotivas reuniones, jajaja. Agradezco a mi compañero de tesis por ser confidente y estar, sí o sí, a pesar de las contradicciones de la vida, apoyándome en cada punto de inflexión y a Yordán que ha sido mi cotutor a la distancia que a pesar de su trabajo y responsabilidades siempre ha buscado el tiempo para atenderme con cada duda que le presentaba hasta resolverla juntos. Agradezco a los tantos profesores que me ayudaron a estar donde estoy hoy, en especial a la vicedecana de extensión, cuyas palabras siempre fueron de aliento y defensa en cada locura que yo hacía, como mi segunda mamá. Estoy muy agradecido de todos esos buenos y no tan buenos momentos por los que pasé y me ayudaste. Agradezco a mi cotutora, que, junto a su equipo, ha podido lograr que este documento y exposición fuese todo un éxito con cada corrección y noche de ensayo hasta las tantas horas. Nunca voy a olvidar cómo me regañabas, y menos con la locura del espaciado. Agradezco a todos en general. Realmente nunca olvidaré esta etapa de mi vida ni las personas que dejaron huellas en ella.

AGRADECIMIENTOS

Ricardo Adrian Avila Rodriguez

Gracias a mis padres, Ricardo Adrian Avila Rodriguez y Osiris Aurora Rodriguez Brito, por siempre estar presentes para mí, todo cuanto tengo y soy hoy es por ustedes, a sus consejos y a esa actitud ejemplar que me han mostrado siempre, tanto personal como profesionalmente, a esa perseverancia que ambos me han enseñado a tener ante todo lo que afrontamos en la vida. A mi hermano Adrián por darme ese ejemplo de hermano mayor, esa responsabilidad y disciplina que tanto admiro de tí, a mi hermano Franky por el amor y la dedicación que siempre me has dado. A mi abuela Aurora por ser una de las personas que más tiempo me ha dedicado en la vida sin exigir algo a cambio. A mis primos Alejandro, Daniel, Lizandra y Sorelis a los que quiero como a mis hermanos y que a pesar de la distancia a la que estamos me han respaldado incondicionalmente. A toda mi familia por su preocupación, gracias. A mis amigos que estuvieron presente durante este proceso. A los dos hermanos de vida que me dio la UCI, Guido y Yordán, por sus consejos, sus regaños y su apoyo ayudándome a cumplir mi sueño de ser ingeniero, agradecerles por su tiempo, por todo lo que pasamos juntos, las horas de estudio, de desvelos, de juegos y de fiestas. A los años en el edificio 1 donde pasé la mejor etapa de mi vida universitaria y a todos los que formaron parte de esta. A esa personita especial, mi novia, que no por llevar poco tiempo en mi vida pierde importancia pues ha estado a mi lado en todo este proceso, en las malas noches y en los momentos difíciles que he pasado durante la tesis, recuerda esto es solo el comienzo. A la Universidad de Ciencias Informáticas, esa casa en la que he vivido durante los últimos seis años, y la que me permitió formarme como Ingeniero en Ciencias Informáticas.

A todos ustedes, Gracias...

DEDICATORIA

Guido Alejandro Morales Bordon

Le dedico esta tesis a mi mamá la persona más importante en mi vida la cual me ha dado las fuerzas necesarias para poder continuar con cada una de las dificultades que se me han presentado a lo largo de la carrera y en mi camino de vida, agregar a mi abuela y mi hermana que han formado parte de ese equipo que siempre esta para animarme y ayudarme en cada momento difícil que he tenido que afrontar. No puede faltar en esta dedicatoria a mi pareja que gracias a su esfuerzo he podido caminar con seguridad a pesar de estar vendado por el estrés y la incertidumbre en cada una de las horas antes de la exposición de la tesis.

Ricardo Adrian Avila Rodriguez

A mis padres, por estar en todos los momentos y ser ese apoyo incondicional que necesitamos en la vida. A mis hermanos por el cariño y amor que siempre me han dado, a mi abuela que ha estado en las buenas y las malas y a mi abuelo que, aunque hoy no está entre nosotros siempre estuvo aconsejándome y guiándome para ser mejor cada día.

RESUMEN

La administración del proceso de gestión de los ejercicios de culminación de estudios de la especialidad de Ingeniería en Ciencias Informáticas de la Facultad 2 se ejecuta de manera manual, lo cual conlleva a que sea complejo, dada la considerable cantidad de datos que se generan en este procedimiento y su subsiguiente organización y búsqueda. El presente estudio de investigación se centra en la creación de un sistema web concebido con tecnologías de código abierto para simplificar dicho proceso. Este sistema en línea fue dirigido por la metodología Extreme Programming (XP). En cuanto a las tecnologías del lado del usuario, se emplearon: HTML5, JavaScript, CSS3 y React, respaldadas por el marco de diseño React-Bootstrap. El lenguaje del lado del servidor utilizado fue Python mediante su framework Django, se empleó el Visual Studio Code como entorno de desarrollo, PgAdmin 4 y PostgreSQL para la administración de bases de datos. Se incorporaron funcionalidades en la gestión de tesis, estudiantes, profesores, tribunales, cortes de tesis, entre otros. Asimismo, se introdujeron funcionalidades para cargar documentos e informes en el sistema. Se llevaron a cabo pruebas de software a la propuesta de solución, las cuales proporcionan niveles de confirmación de las funcionalidades y la satisfacción por parte del usuario.

PALABRAS CLAVE: Funcionalidades, pruebas de software, sistema web, ejercicios de culminación de estudios

ABSTRACT

The administration of the management process for the completion exercises of studies in the Computer Science Engineering specialty of Faculty 2 is carried out manually, which makes it complex, given the considerable amount of data that is generated in this process. procedure and its subsequent organization and search. This research study focuses on the creation of a web system designed with open source technologies to simplify this process. This online system was driven by the Extreme Programming (XP) methodology. Regarding user-side technologies, HTML5, JavaScript, CSS3 and React were used, supported by the React-Bootstrap design framework. The server-side language used was Python through its Django framework, Visual Studio Code was used as a development environment, PgAdmin 4 and PostgreSQL for database administration. Functionalities were incorporated in the management of thesis, students, professors, courts, thesis cuts, among others. Likewise, functionalities were introduced to upload documents and reports to the system. Software tests were carried out on the proposed solution, which provide levels of confirmation of the functionalities and user satisfaction.

KEYWORDS: Functionalities, software testing, web system, study completion exercises.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL PROCESO DE GESTIÓN DE LA INFORMACIÓN DE LOS EJERCICIOS DE CULMINACIÓN DE ESTUDIOS EN LA CARRERA DE ICI Y ARSI EN LA FACULTAD 2	6
I.1 Conceptos básicos asociados con el problema planteado.....	6
I.2 Proceso de gestión de tesis en la Facultad 2.....	8
I.3 Soluciones informáticas para el desarrollo del sistema.....	8
I.4 Metodologías, lenguajes, herramientas y tecnologías informáticas para el desarrollo del sistema.....	10
Conclusiones del capítulo.....	16
CAPÍTULO II: DISEÑO DEL SISTEMA DE GESTIÓN DE LOS EJERCICIOS DE CULMINACIÓN DE ESTUDIOS DE LA FACULTAD 2.....	17
II.1 Análisis del proceso de gestión de tesis.	17
II.2 Exploración.....	17
II.3 Historias de usuarios	18
II.4 Planificación	26
II.4.2 Plan de iteraciones	27
II.4.3 Plan de entregas	29
II.5 Diseño	30
II.5.1 Patrón arquitectónico.....	31
II.5.2 Patrón de diseño	32
II.5.3 Tarjetas CRC.....	35
II.5.4 Modelo de datos	39
Conclusiones del capítulo.....	40
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	42
III.1 Implementación	42
III.2 Estándares de codificación.....	42
III.3 Tareas de ingeniería.....	43
III.4 Pruebas de software.....	52
III.4.1 Estrategia de pruebas.....	53

III.4.2 Pruebas Unitarias	53
III.4.3 Pruebas de Aceptación	54
III.4.4 Validación del Sistema	61
Conclusiones del capítulo.....	62
CONCLUSIONES FINALES	63
RECOMENDACIONES.....	64
REFERENCIAS BIBLIOGRÁFICAS.....	65

ÍNDICE DE TABLAS

Tabla 1 HU 1 Autenticar Usuario	19
Tabla 2 HU 2 Gestionar Usuario	20
Tabla 3 HU 3 Gestionar Estudiante	20
Tabla 4 HU 4 Gestionar Profesor.....	21
Tabla 5 HU 5 Gestionar Tribunales	21
Tabla 6 HU 6 Gestionar Tesis.....	22
Tabla 7 HU 7 Gestionar Primer Corte	22
Tabla 8 HU 8 Gestionar Segundo Corte	23
Tabla 9 HU 9 Gestionar Pre-defensa.....	23
Tabla 10 HU 10 Gestionar Defensa.....	24
Tabla 11 HU 11 Buscar Tesis	24
Tabla 12 HU 12 Buscar Usuario	25
Tabla 13 HU 13 Buscar Tribunal.....	25
Tabla 14 HU 14 Agregar Fichero	25
Tabla 15 HU 15 Repositorio de tesis	26
Tabla 16 Estimación de esfuerzos	27
Tabla 17 Plan de iteraciones / cantidad de semanas	29
Tabla 18 Plan de entrega.....	30
Tabla 19 Tarjeta CRC #1 Estudiante	36
Tabla 20 Tarjeta CRC #2 Profesor	36
Tabla 21 Tarjeta CRC #3 Tribunal	37
Tabla 22 Tarjeta CRC #4 Primer Corte.....	37
Tabla 23 Tarjeta CRC #5 Segundo Corte	38
Tabla 24 Tarjeta CRC #6 Pre-defensa.....	38
Tabla 25 Tarjeta CRC #7 Defensa.....	39
Tabla 26 TI 1 Autenticar Usuario	44
Tabla 27 TI 2 Gestionar Usuario	44
Tabla 28 TI 3 Gestionar Estudiante	45

Tabla 29 TI 4 Gestionar Profesor.....	45
Tabla 30 TI 5 Gestionar Tribunales	46
Tabla 31 TI 6 Gestionar Tesis.....	47
Tabla 32 TI 7 Gestionar Primer Corte	47
Tabla 33 TI 8 Gestionar Segundo Corte	48
Tabla 34 TI 9 Gestionar Pre-defensa.....	49
Tabla 35 TI 10 Gestionar Defensa.....	49
Tabla 36 TI 11 Buscar Tesis	50
Tabla 37 TI 12 Buscar Usuario	50
Tabla 38 TI 13 Buscar Tribunal.....	51
Tabla 39 TI 14 Agregar Fichero	51
Tabla 40 TI 15 Repositorio de Tesis	52
Tabla 41 Caso de Prueba de aceptación 1.....	55
Tabla 42 Caso de Prueba de aceptación 2.....	56
Tabla 43 Caso de Prueba de aceptación 3.....	56
Tabla 44 Caso de Prueba de aceptación 4.....	57
Tabla 45 Caso de Prueba de aceptación 5.....	58
Tabla 46 Caso de Prueba de aceptación 6.....	58
Tabla 47 Caso de Prueba de aceptación 7.....	59
Tabla 48 Caso de Prueba de aceptación 8.....	59
Tabla 49 Caso de Prueba de aceptación 9.....	60
Tabla 50 Caso de Prueba de aceptación 10.....	60
Tabla 51 Validación del sistema.	61

ÍNDICE DE FIGURAS

Figura 1 Proceso de gestión de tesis.....	17
Figura 2 Ejemplo de Patrón Experto	34
Figura 3 Ejemplo de Patrón Creador	34
Figura 4 Modelo de datos	40
Figura 5 Estándares de codificación1	43
Figura 6 Estándares de codificación2	43
Figura 7 Estándares de codificación3	43
Figura 8 Pruebas Unitarias	54

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) han sido objeto de estudio y análisis por numerosos investigadores, organizaciones y gobiernos en todo el mundo, debido al gran efecto que tienen en la vida cotidiana. Con un impacto revolucionario en la forma en que se vive y trabaja, han transformado la manera en que las empresas operan, ya sea permitiendo nuevas modalidades de trabajo a distancia, aumentando la eficiencia y productividad, o facilitando la optimización de procesos en diversas áreas. (García, J., 2019)

El desarrollo de las TIC en el ámbito universitario cubano, propicia la existencia leyes y regulaciones que rigen el proceso de elaboración y presentación de los trabajos finales de grado. La Ley No. 20 de Educación Superior establece la obligatoriedad de los trabajos finales de grado y los requisitos para su elaboración y presentación. Por su parte, el Ministerio de Educación Superior emitió la Resolución Ministerial No. 210/2019, que establece los procedimientos y plazos para la elaboración y defensa de los trabajos finales de grado en todas las universidades del país (Pérez, 2020).

La Universidad de las Ciencias Informáticas (UCI) es un ejemplo de este avance y se considera una universidad que relaciona el conocimiento universitario con la producción para la sociedad, lo que la convierte en una institución educacional de gran relevancia en el país. En la UCI, los estudiantes realizan un ejercicio de culminación de estudios denominado "Trabajo de Diploma". Es una modalidad de proyecto final que busca integrar los conocimientos y habilidades adquiridos durante la carrera en un proyecto práctico.

El Reglamento de Trabajos de Diploma de la UCI establece los requisitos y procedimientos necesarios para la elaboración y defensa del Trabajo de Diploma, incluyendo la selección del tema, la elaboración de un plan de trabajo, la presentación del trabajo escrito y la defensa ante un tribunal evaluador. Cada tipo de proyecto busca formar a los estudiantes en áreas específicas y fomentar el desarrollo de habilidades y competencias en el campo de la informática y las TIC.

La gestión de la información de los ejercicios de culminación de estudios en la carrera de Ingeniería en Ciencias Informáticas (ICI) y la carrera de Administración en Redes y Seguridad Informática (ARSI) en la Facultad 2, es un proceso complejo que genera grandes volúmenes

de información y que actualmente se realiza de manera manual usando hojas de cálculo Excel. Este proceso involucra a estudiantes, profesores principales de ICI y ARSI, jefes de los Departamentos de Ciberseguridad e Informática, Vicedecana de Formación y Decana de la Facultad 2.

Existe gran cantidad de datos dispersos e ineficiencia en el personal encargado de llevar el control de la información, lo que implica la posible equivocación en la conformación de la misma o su pérdida. Como parte del proceso de informatización que se lleva a cabo en esta institución se pretende, haciendo uso de las TIC: gestionar los distintos tipos de ejercicios de culminación de estudio, la planificación de los tribunales para cada uno de los ejercicios, la asignación a los estudiantes y tribunales los trabajos de diplomas y desarrollar un software de una forma más fácil y eficiente.

Las condiciones que hacen necesarias la implantación de un sistema web para gestionar la información de los ejercicios de culminación de estudios en las carreras de ICI y ARSI en la Facultad 2, conlleva a que se haya identificado como **problema de investigación**: ¿Cómo facilitar la gestión de la información de los ejercicios de culminación de estudios en las carreras de ICI y ARSI de la Facultad 2?

El **objeto de estudio** está orientado al proceso de gestión de la información de los ejercicios de culminación de estudios en las carreras de ICI y ARSI en la Facultad 2 y se define como **objetivo general** desarrollar un sistema web que facilite la gestión de la información de los ejercicios de culminación de estudios en las carreras de ICI y ARSI de la Facultad 2. Derivándose los siguientes **objetivos específicos**:

1. Sistematización los fundamentos y referentes teórico-metodológicos para la gestión de la información de los ejercicios de culminación de estudios.
2. Diagnóstico del estado de transformación digital en el proceso de gestión de la información de los ejercicios de culminación de estudios en ICI y ARSI de la Facultad 2.
3. Desarrollo de la solución informática para facilitar la gestión de la información de los ejercicios de culminación de estudios en ICI y ARSI de la Facultad 2.
4. Validación de la solución informática para la gestión de la información de los ejercicios de culminación de estudios en ICI y ARSI de la Facultad 2.

Se establece como **campo de acción**: Sistemas web para la gestión de información de los ejercicios de culminación de estudios.

Se plantean las siguientes **preguntas científicas** para el mejor desarrollo de la investigación:

1. ¿Cuáles son los principales fundamentos y referentes teóricos-metodológicos para el desarrollo de un sistema informático que facilite la gestión de la información para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2?
2. ¿Cuál es el estado actual del proceso de gestión de la información para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2?
3. ¿Qué componentes tendrá el sistema informático para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2?
4. ¿Qué resultados tendrá la validación del sistema informático desarrollado para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2?

Para cumplir con el objetivo establecido, se plantean las siguientes **tareas científicas específicas**:

1. Sistematización de los principales fundamentos y referentes teóricos-metodológicos para el desarrollo de un sistema informático que facilite la gestión de la información para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2.
2. Diagnóstico del estado actual del proceso de gestión de la información para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2.
3. Desarrollo del sistema informático para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2.
4. Validación de la solución propuesta del sistema informático para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2.

Para el desarrollo de la investigación se utilizaron los siguientes **métodos científicos**:

Métodos teóricos:

- **Analítico-sintético:** Se utiliza para el análisis de la información del negocio de la organización y arribar a las conclusiones de la investigación, así como para precisar las características del trabajo a realizar.
- **Histórico-lógico:** Se utiliza para analizar las tendencias actuales en los sistemas de gestión de los ejercicios de culminación de estudio, de los modelos de desarrollo y las herramientas, lenguajes y técnicas utilizadas durante la investigación.
- **Modelación:** Se emplea para realizar la representación y explicación teórica de la realidad existente y deseada, que tiene como resultado la creación de un modelo con la estructura, relaciones y características de la solución, a través de diagramas.

Métodos empíricos:

- **Observación:** Se utiliza para conocer aspectos esenciales del estado actual de la gestión de la información para los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2.
- **Entrevistas:** Se realizan a los actores del negocio para la obtención de toda la información sobre el desarrollo del proceso de la gestión de la información de los ejercicios de culminación de estudios en ICI y ARSI en la Facultad 2.

El presente documento está estructurado en tres capítulos:

Capítulo I “Fundamentos y referentes teórico-metodológicos sobre el proceso de gestión de información de los ejercicios de culminación de estudios en la carrera de Ingeniería en Ciencias Informáticas y la carrera de Administración en Redes y Seguridad Informática en la Facultad 2”: Se realiza la fundamentación teórica del tema a investigar a partir de un estudio del estado del arte a nivel nacional e internacional teniendo en cuenta las tendencias de la actualidad y todos los conceptos asociados al tema que sirvieron de base para el desarrollo de la propuesta de investigación.

Capítulo II “Diseño de solución al proceso de gestión de información de los ejercicios de culminación de estudios en la carrera de Ingeniería en Ciencias Informáticas y la carrera de Administración en Redes y Seguridad Informática en la Facultad 2”: Se establecen los argumentos a partir de la aplicación del enfoque, métodos y metodologías de Ingeniería de Software

aplicados para cumplir el objetivo, dando así, solución al problema científico planteado y obteniendo un resultado práctico.

Capítulo III “Validación de la Solución Propuesta”: Establece los resultados y argumentos que demuestran factibilidad, fiabilidad e integridad de la propuesta de solución al aplicar técnicas y métodos de verificación y validación de software, técnicas y métodos de validación científica de la transformación o impacto sobre el objeto de estudio.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL PROCESO DE GESTIÓN DE LA INFORMACIÓN DE LOS EJERCICIOS DE CULMINACIÓN DE ESTUDIOS EN LA CARRERA DE ICI Y ARSI EN LA FACULTAD 2

En el presente capítulo se abordaran los fundamentos y referentes teórico-metodológicos sobre el proceso de gestión de información de los ejercicios de culminación de estudios en la carrera de Ingeniería en Ciencias Informáticas y la carrera de Administración en Redes y Seguridad Informática en la Facultad 2”: Se realiza la fundamentación teórica del tema a investigar a partir de un estudio del estado del arte a nivel nacional e internacional teniendo en cuenta las tendencias de la actualidad y todos los conceptos asociados al tema que sirvieron de base para el desarrollo de la propuesta de investigación. Por último, se seleccionan la metodología, el lenguaje de modelado y las herramientas a utilizar, justificando la selección de cada uno de estos sistemas para el desarrollo de la propuesta de solución.

I.1 Conceptos básicos asociados con el problema planteado.

Para la elaboración de este informe se utilizan una serie de conceptos entre los que se definen la clave de esta ponencia.

Gestión de la información:

La gestión de la información es el proceso de planificación, coordinación, control, organización, capacitación, promoción y otros medios para asegurar que la información se adquiriera, se almacene, se recupere, se use, se comparta y se elimine adecuadamente para lograr los objetivos de la organización. Este proceso es fundamental para garantizar la eficiencia y efectividad en el manejo de la información en una organización. La gestión de la información se enfoca en la planificación y control de los recursos de información, incluyendo la tecnología, los datos, los procesos y las personas (CSRC, NIST, 2023).

Sistema Informático:

Un sistema informático es un conjunto de componentes interconectados que trabajan juntos para procesar, almacenar y transmitir información. Los componentes de un sistema informático incluyen hardware, software, datos, personas y procesos. Este conjunto de componentes es fundamental para garantizar la eficiencia y efectividad en el manejo de la información en una organización. El hardware incluye computadoras o cualquier tipo de dispositivo electrónico,

que consisten en procesadores, memoria, sistemas de almacenamiento externo, entre otros. El software incluye al sistema operativo, firmware y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos. Por último, el componente humano incluye al personal técnico que apoya y mantiene el sistema y a los usuarios que lo utilizan (CSRC, NIST, 2023).

Sistema de Gestión de información:

Un sistema de gestión de información (SGI) se refiere a un conjunto de herramientas, técnicas y procesos utilizados para recopilar, almacenar, organizar y analizar datos e información en una organización. El objetivo principal de un SGI es facilitar la toma de decisiones basadas en información precisa y oportuna. Esto se logra a través de la implementación de bases de datos, software de gestión y sistemas de comunicación eficientes. Un ejemplo de SGI es el sistema de gestión de información de una empresa que almacena y organiza datos sobre clientes, proveedores, inventario y ventas (Laudon, K. C., 2016).

Ejercicios de Culminación de Estudios:

Los ejercicios de culminación de estudios son una actividad académica que se realiza al final de un programa de formación, con el objetivo de evaluar los conocimientos y habilidades adquiridos por el estudiante durante su formación. Estos ejercicios pueden tomar diferentes formas, como trabajos de investigación, proyectos, tesis, entre otros, y su realización es un requisito para obtener el título correspondiente. En Cuba, los ejercicios de culminación de estudios son regulados por la Gaceta Oficial de Cuba, que establece los requisitos y procedimientos para su realización. Además, las universidades y centros de formación pueden establecer sus propias normas y procedimientos para la realización de estos ejercicios (Gaceta Oficial de Cuba, 2020).

Tesis:

Es un trabajo de investigación que se realiza al final de un programa de estudios y que tiene como objetivo demostrar la capacidad del estudiante para aplicar los conocimientos adquiridos en la resolución de un problema o la generación de nuevo conocimiento. La tesis es un requisito para obtener el título correspondiente y debe ser defendida ante un tribunal (Universidad de La Habana, 2020).

I.2 Proceso de gestión de tesis en la Facultad 2.

Actualmente, este proceso se lleva a cabo principalmente de manera manual, aunque se utilizan tecnologías para realizar sus tareas a través de diversas plataformas con diferentes formatos y alojamientos de información en ubicaciones diversas, dependiendo del actor que la genera o procesa para la toma de decisiones. En este proceso participan tanto docentes como estudiantes y tutores de las investigaciones, quienes pueden encontrarse en diferentes lugares dentro y fuera del territorio cubano. Se procesan datos numéricos y textuales que experimentan cambios constantes con el tiempo, los cuales deben ser examinados y validados por los distintos actores involucrados en el proceso. Con frecuencia, es necesario analizar los datos contenidos en este proceso para la toma de decisiones y la elaboración de informes precisos sobre el comportamiento de las variables.

I.3 Soluciones informáticas para el desarrollo del sistema.

Es necesario realizar un estudio de los sistemas homólogos existentes a nivel internacional y nacional, para lograr una mejor comprensión de las características del sistema a desarrollar.

Nivel internacional:

Desarrollo e implementación de un sistema para el registro y control de tutorías académicas, basado en una arquitectura de tres capas, para la carrera de Ingeniería en Sistemas de la Universidad Politécnica Salesiana sede Quito:

El proyecto de tesis que se ha desarrollado consiste en una aplicación web que permite el registro y control de las tutorías académicas que brinda la Universidad Politécnica Salesiana sede Quito como parte de la calidad de la educación. Mejorando el proceso de ingreso de la información se puede obtener de una manera ágil indicadores que ayuden a la toma de decisiones y a evitar que muchos estudiantes interrumpan su futuro profesional por falta de un apoyo académico adicional que la Universidad no ha podido brindarle ya que no contaba con la obtención de información de una manera inmediata. El sistema presenta varios reportes de tipo gerenciales y estadísticos que se generan vía online y que contienen la información actualizada y confiable de cómo se está desarrollando este apoyo académico por parte de los docentes y estudiantes (Garinzo, 2015).

Aplicación informática para los procesos de gestión de las tutorías académicas de la carrera de tecnologías de la información de la universidad estatal del sur de Manabí:

La investigación se basa en una aplicación informática para los procesos de tutorías académicas de la carrera de Tecnologías de la Información de la Universidad Estatal del Sur de Manabí. El proyecto es de gran importancia por lo que se debe mantener un estricto control sobre cómo se desarrollan estos procesos para evitar inconvenientes que obstaculicen las tutorías académicas. Su objetivo es desarrollar una aplicación informática donde se refleje todo lo relacionado con la realización de los procesos de tutoría académica. Posibilita obtener, crear, analizar y comunicar información con mayor detalle y más rápido, identificando el problema, desarrollando los objetivos y obteniendo una respectiva justificación del trabajo. (Aranea, 2021).

Nivel nacional:

Akademos:

En el curso 2007 - 2008 se realizó en el proyecto Akademos un trabajo de diploma relacionado con el análisis y diseño de un módulo para gestionar los procesos de tesis que se realizan en la Universidad de Ciencias Informáticas. Ese sistema se hizo lo más genérico posible para que pudiera ser usado por todos los implicados en esos procesos en cada una de las facultades. Por tal razón en el mismo se gestiona un conjunto de información que puede ser usada por cualquier facultad, por tanto, hay muchas funcionalidades que sirven de base para la especificación de requisitos que se necesita en la presente investigación; pero aun así no resuelve el problema planteado, pues sobre esa idea se necesita ser más específico y agregar otras funcionalidades que no están contempladas en dicho diseño (Akademos).

Módulo informático para la gestión de procesos de actas y culminación de estudio para el sector de la educación superior:

La necesidad de realizar este trabajo está dada por la insuficiencia que existe en la gestión de los documentos de la Facultad de Informática y Ciencias Exactas de la Universidad de Ciego de Ávila Máximo Gómez Báez. Su objetivo es brindar una herramienta capaz de dar seguimiento a todo el proceso documental del desarrollo del trabajo de culminación de estudios, desde sus inicios hasta su terminación, además de realizar y llevar el control de todas las actas de dicha facultad. Este sistema está compuesto por los componentes: Trabajo de culminación de estudios y Actas de la facultad. Para la implementación del sistema se utilizó Apache como servidor web, MySQL como gestor de bases de datos. Para la creación de los diagramas, se utilizó la herramienta Visual Paradigm. Como marco de trabajo se utilizó el Framework Yii 2; y como lenguajes de programación PHP y JavaScript. Con el desarrollo y posterior utilización

del módulo informático para la gestión de documentos de la Facultad de Informática y Ciencias Exactas de la Universidad de Ciego de Ávila. Máximo Gómez Báez se facilita tener un mejor control de la documentación de dicha entidad (Victor & Bouza, 2020).

En los sistemas estudiados se encontraron diferentes funcionalidades que pueden ser de utilidad en el desarrollo de un nuevo sistema de acuerdo a las necesidades planteadas en la presente investigación, como son: gestionar tema y gestionar usuarios, sin embargo, no gestionan lo relacionado con los tribunales de tesis, información que es necesaria para el conocimiento de los estudiantes y tutores. Por lo antes mencionado se decide desarrollar un sistema que cumpla con las necesidades del cliente y el dominio del problema.

I.4 Metodologías, lenguajes, herramientas y tecnologías informáticas para el desarrollo del sistema.

Las características del equipo de desarrollo son un factor determinante en la elección de las tecnologías a utilizar. En este caso, se cuenta con un equipo de desarrollo conformado por dos personas con poca experiencia en el trabajo y desarrollo de software, ya que se está en pleno proceso de aprendizaje y desarrollo. Dado que el equipo es pequeño y cuenta con poca experiencia, es fundamental utilizar tecnologías que permitan agilizar el proceso de desarrollo y maximizar la productividad. En este sentido, las metodologías ágiles como XP se adaptan perfectamente a las necesidades. Esta tecnología permitirá aprovechar al máximo las habilidades, conocimientos, y trabajar de manera más eficiente y efectiva. Además, las metodologías ágiles promueven la entrega continua de software funcional, lo cual permitirá obtener retroalimentación temprana del cliente y realizar ajustes rápidos en base a sus necesidades y expectativas. Agregar que es muy probable que surjan cambios y ajustes en los requisitos del proyecto y utilizando este tipo de metodologías se puede responder de manera ágil y eficiente a estos cambios, asegurando que el software entregado cumpla con las expectativas del cliente.

Metodologías de Desarrollo de Software

Metodología XP

La metodología de Programación Extrema (XP) es un enfoque ágil de desarrollo de software que se ha vuelto cada vez más relevante en la industria. La metodología XP se basa en principios como la comunicación constante, la retroalimentación continua y la simplicidad. Esta

metodología se centra en la entrega temprana y frecuente de software funcional, lo que permite una mayor adaptabilidad a medida que los requisitos cambian. Además, XP promueve la colaboración estrecha entre desarrolladores y clientes, lo que facilita la comprensión de las necesidades y expectativas del cliente. La metodología XP es importante debido a su capacidad para mejorar la calidad del software y la satisfacción del cliente. Al enfocarse en la comunicación y la retroalimentación continua, XP ayuda a identificar y corregir errores de manera oportuna, lo que reduce los costos y el tiempo dedicado a la depuración. Además, al priorizar la simplicidad y la entrega frecuente, XP permite una mayor adaptabilidad a los cambios en los requisitos del proyecto. Esto resulta en un software más útil y valioso para el cliente (Beck, K., & Andres, C., 2004).

El equipo, que consta de dos personas, utilizará la metodología XP para trabajar por varias razones. En primer lugar, XP se centra en la comunicación constante y la colaboración estrecha entre los miembros del equipo. Al ser un equipo pequeño, la comunicación directa y efectiva es esencial para asegurar que todos estén alineados en los objetivos del proyecto y para evitar malentendidos. La metodología XP promueve la interacción presencial y la retroalimentación continua, lo que facilita la comunicación fluida y la resolución rápida de problemas. Además, XP se basa en principios de simplicidad y entrega frecuente de software funcional. Esto es especialmente beneficioso para un equipo de dos personas, ya que permite enfocarse en las características esenciales del software y entregar valor de manera temprana y continua. Al evitar la sobrecarga de procesos y documentación excesiva, XP permite una mayor agilidad y adaptabilidad a medida que los requisitos evolucionan, lo que permite maximizar la eficiencia y adaptabilidad en el trabajo conjunto.

Lenguajes de Modelado de Software

Lenguaje de Modelado Unificado (UML)

Mediante el uso de diferentes tipos de diagramas, como casos de uso, clases, secuencia y actividad, el UML facilita la modelación de sistemas, permitiendo comprender y representar los requisitos, la estructura y el comportamiento del sistema de manera clara y concisa (Jeffries, Anderson, Hendrickson, & Roberts, 2001).

Las ventajas del UML incluyen la facilitación de la comunicación y comprensión entre los stakeholders, la creación de una notación visual unificada y comprensible, la posibilidad de documentar conceptos y diseños de manera efectiva, el fomento del diseño estructurado y modular, y la promoción de la reutilización de componentes (Beck, 1999).

Lenguaje de Programación

Lenguaje Marcado de Hipertexto HTML 5

El Lenguaje de Marcado de Hipertexto (HTML5) es una versión actualizada y mejorada de HTML, el lenguaje utilizado para estructurar y presentar contenido en la web. HTML5 combina HTML, CSS (Hojas de estilo en cascada) y JavaScript para ofrecer una experiencia de usuario más rica y dinámica en las páginas web.

HTML5 se caracteriza por ser un lenguaje extensible que permite añadir nuevas características, etiquetas y funcionalidades para el diseño y desarrollo de páginas web. Esto proporciona a los desarrolladores una mayor flexibilidad y potencia a la hora de crear contenido interactivo y multimedia. Una de las ventajas significativas de HTML5 es su capacidad para admitir tanto la sintaxis HTML como XHTML. Esto permite a los creadores de contenido elegir entre un enfoque más práctico y flexible (sintaxis HTML) o una visión más académica y estricta (sintaxis XHTML), dependiendo de sus necesidades y preferencias (Duckett, 2014).

Hojas de estilo en cascada (CSS 3)

La web demanda diseño y funcionalidad, no solo organización estructural o definición de secciones. En este nuevo paradigma, HTML se presenta junto con CSS y JavaScript como un único instrumento integrado. CSS es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, entre otros. El propósito del desarrollo de CSS es separar la estructura y el contenido de la presentación estética en un documento (Alawar & Abu-Naser, 2017).

JavaScript

JavaScript es un lenguaje interpretado usado para múltiples propósitos, pero solo considerado como un complemento hasta ahora. Principalmente es utilizado para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes

de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Marani, 2019).

Lenguajes de Programación (Python)

Python es un lenguaje de programación interpretado de alto nivel, popular y de código abierto. Aunque comparte algunas similitudes con PHP, tiene características únicas y es ampliamente utilizado tanto en el desarrollo web como en otros ámbitos. Python se destaca por su legibilidad y simplicidad, lo que lo convierte en un lenguaje accesible tanto para principiantes como para desarrolladores experimentados.

Es seleccionado Python para el desarrollo de la aplicación web debido a las siguientes características y aplicaciones:

1. Versatilidad: Python es un lenguaje versátil que se utiliza en una amplia gama de aplicaciones, desde desarrollo web y científico hasta automatización de tareas y desarrollo de juegos.
2. Sintaxis clara y legible: Python se caracteriza por su sintaxis clara y legible, lo que facilita la comprensión y el mantenimiento del código. Utiliza una filosofía de diseño que enfatiza la legibilidad y la simplicidad.
3. Amplia disponibilidad de bibliotecas: Python cuenta con una amplia variedad de bibliotecas y módulos disponibles que facilitan el desarrollo de aplicaciones.
4. Comunidad activa: Python tiene una comunidad activa y comprometida de desarrolladores que contribuyen con bibliotecas, tutoriales y recursos. Esto facilita el aprendizaje y el intercambio de conocimientos.
5. Portabilidad: Python es compatible con casi todos los sistemas operativos y plataformas, lo que lo hace altamente portátil y adaptable a diferentes entornos de desarrollo.
6. Enfoque en la legibilidad del código: Python promueve el uso de una sintaxis limpia y estructurada, lo que facilita la lectura y comprensión del código. Esto favorece la colaboración entre desarrolladores y la mantenibilidad del proyecto a largo plazo (Fossati, 2018).

Marcos de Trabajo

Bootstrap

Bootstrap es un framework de diseño de código abierto que se enfoca en la creación de sitios web y aplicaciones móviles responsivas. A diferencia de otros framework, Bootstrap utiliza un enfoque de "diseño basado en componentes", donde se agrupan elementos comunes en componentes reutilizables con estilos predefinidos. Esto facilita la creación de interfaces de usuario coherentes y estandarizadas en todo el sitio o aplicación. Bootstrap también ofrece una amplia gama de componentes y herramientas, como botones, menús desplegables, formularios y cuadros modales, lo que permite una mayor eficiencia en el desarrollo. Bootstrap es uno de los frameworks más populares y ampliamente utilizados en la actualidad (W3Schools, 2021).

Django 4.2

Django es un Framework de desarrollo web de código abierto basado en Python. Su objetivo es facilitar la creación de aplicaciones web robustas y escalables, siguiendo la arquitectura MVC (Modelo-Vista-Controlador). Django incluye características como un ORM (Object-Relational Mapping) para interactuar con bases de datos, un enrutador URL para gestionar las rutas de la aplicación y un sistema de plantillas para generar contenido dinámico. Actualmente, Django se encuentra en su versión 4.2 y es ampliamente utilizado en la comunidad de desarrollo web (Holovaty et al., 2020).

Gestor de Bases de Datos

PostgreSQL:

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto que ofrece una amplia gama de características y funcionalidades avanzadas. Su importancia radica en su capacidad para manejar grandes cantidades de datos de manera eficiente y confiable, lo que lo convierte en una opción preferida para aplicaciones empresariales y de misión crítica. Una de las características destacadas de PostgreSQL es su soporte para múltiples tipos de datos, incluyendo tipos geométricos y de texto completo, lo que lo hace adecuado para aplicaciones que requieren un alto nivel de flexibilidad en el manejo de información. Se puede agregar que PostgreSQL cuenta con una arquitectura robusta que garantiza la integridad y la consistencia de los datos, así como la capacidad de recuperación en caso de fallos. También ofrece un alto nivel de escalabilidad, lo que le permite manejar grandes volúmenes de información y adaptarse a las necesidades crecientes de una organización (PostgreSQL., s.f.).

Herramientas y Tecnologías

Visual Paradigm 16.2

Visual Paradigm es una poderosa herramienta que permite modelar sistemas de una forma eficiente, facilitando el desarrollo de proyectos informáticos. Implementa el estándar UML de manera completa a través de una gran variedad de diagramas como clases, secuencia y estado. Estos diagramas permiten visualizar y comprender claramente todos los requerimientos, relaciones y dinámicas de un sistema. Asimismo, la integración de Visual Paradigm con otras tecnologías como bases de datos o frameworks, agiliza el mapeo de los modelos a código y documentación. Su interfaz intuitiva optimiza la productividad de los equipos de desarrollo al facilitar la creación, navegación y exportación de diagramas de una manera simplificada (OMG, 2019).

PgAdmin 4

PgAdmin 4 es una herramienta de administración gráfica para PostgreSQL. Diseñada para funcionar en diversas plataformas, PgAdmin 4 satisface las necesidades de usuarios con diferentes niveles de experiencia, desde la escritura de consultas SQL simples hasta la administración de bases de datos complejas. Su interfaz gráfica es compatible con todas las características de PostgreSQL, lo que facilita la gestión de la base de datos. Además, la aplicación incluye un editor de sintaxis SQL, un editor de código del lado del servidor, un agente para programación de tareas y soporte para el motor de replicación Slony-I (Niemeyer et al., 2020).

Visual Studio Code 1.83

Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Ofrece funciones avanzadas como autocompletado inteligente, inspección de código, resaltado de errores en tiempo real y soluciones rápidas. Además, cuenta con herramientas de refactorización automática, navegación avanzada y depuración integrada. Visual Studio Code también incluye una terminal incorporada, integración con sistemas de control de versiones (VCS) y herramientas de base de datos. Además, permite el desarrollo remoto con intérpretes remotos y una terminal SSH integrada (Balena et al., 2021).

PyCharm Professional

Es un entorno de desarrollo integrado (IDE) profesional orientado a desarrollar aplicaciones en Python. Está desarrollado por JetBrains S.R.O., formando parte de los IDEs para desarrollo de lenguajes particulares como: IntelliJ IDEA – para Java (PyCharm, s. f.).

Siguiendo la línea y las buenas prácticas de la comunidad se decide usar Pycharm, ya que es un entorno IDE desarrollado específicamente para Python, ofreciendo así muchas ventajas entre ellas: proporciona una finalización del código inteligente, inspecciones del código, indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de código automática y completas funcionalidades de navegación (Funcionalidades - PyCharm, s. f.)

Conclusiones del capítulo

El análisis de los marcos teóricos y metodológicos asociados a la administración de los proyectos finales de investigación facilitó una mejor comprensión sobre el dominio de la indagación. Los enfoques examinados (entorno nacional e internacional) no son completamente factibles, dado que no satisfacen las demandas del cliente. El estudio de las herramientas, lenguajes y tecnologías, tales como Python, Visual Studio Code, PgAdmin, HTML5, CSS, Bootstrap, JavaScript y otros, permitió adentrarse en los conocimientos necesarios para el desarrollo de un sistema informático destinado a la gestión de los proyectos finales de investigación.

CAPÍTULO II: DISEÑO DEL SISTEMA DE GESTIÓN DE LOS EJERCICIOS DE CULMINACIÓN DE ESTUDIOS DE LA FACULTAD 2

Este capítulo estará dedicado a la identificación de las principales funcionalidades de la solución propuesta. Además, en los epígrafes del presente capítulo, se expondrán los artefactos generados por la propuesta dada como metodología de desarrollo de software a emplear, siendo XP la seleccionada, en sus fases de Exploración, Planificación y Diseño.

II.1 Análisis del proceso de gestión de tesis.

El proceso que se lleva a cabo en la Facultad 2 de la UCI, referente a la evaluación de los ejercicios de culminación de estudios es realizado a partir del tipo de evaluación defensa de la tesis, a continuación, se muestran dichos procesos en la figura:

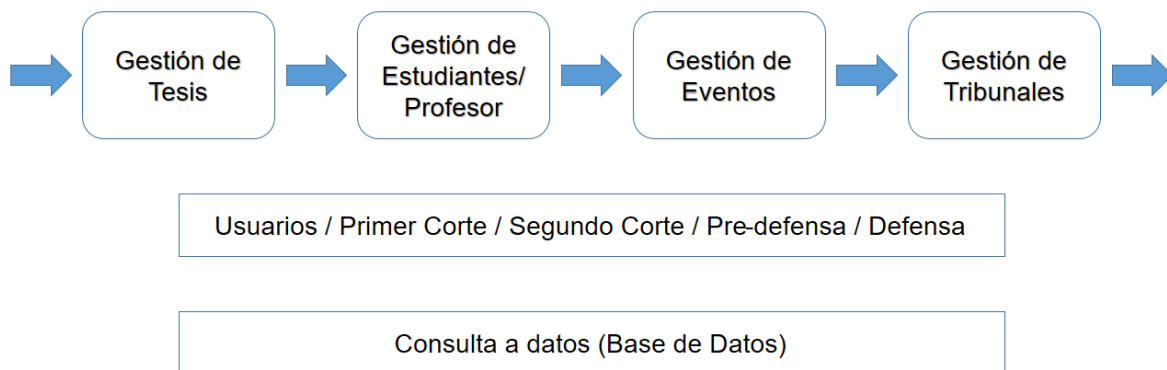


Figura 1 Proceso de gestión de tesis

II.2 Exploración

En la fase exploratoria inicial de un proyecto, el equipo pasa tiempo familiarizándose con el dominio del negocio, los requisitos generales del sistema y las tecnologías que podrían usarse. El objetivo es comprender mejor el problema y probar diferentes enfoques tecnológicos a través de pruebas y prototipos a pequeña escala. Se definen las historias de usuario más importantes, pero no se pretende tener una comprensión clara de todos los requisitos. En cambio, el foco está en identificar los riesgos clave y probar suposiciones a través de un trabajo experimental preliminar (Brehner, 2015).

II.3 Historias de usuarios

Las historias de usuario (HU) son la principal técnica para definir y priorizar los requisitos funcionales del sistema. Se redactan breves tarjetas que describen una funcionalidad que proporciona valor al usuario. Es importante que sean escritas en lenguaje no técnico y que cumplan con las tres C's: concreta, completa y comprobable. Cada historia debe ser lo suficientemente pequeña para ser desarrollada por el equipo en una o dos semanas e ir validando así el producto de forma incremental con el cliente. Las historias se priorizan según su importancia para el negocio y se evalúa el riesgo asociado a su implementación. Todo ello permite planificar las iteraciones de manera flexible para satisfacer primero aquellas funcionalidades más críticas (Schwaber & Sutherland, 2017).

Con el fin de seleccionar que historia de usuario tiene mayor importancia, según las necesidades del cliente y poder realizar una planificación adecuada de la implementación, cada HU es clasificada por el propietario del producto según la prioridad que implique esta para el negocio en:

- **Alta:** Fundamentales en el desarrollo del sistema.
- **Media:** Contienen funcionalidades necesarias, pero no imprescindibles.
- **Baja:** Constituyen funcionalidades que son de ayuda para el control de elementos asociados al equipo de desarrollo, a la estructura, aparte completamente del sistema.

Además, se examina la complejidad y la eventual presencia de fallos durante la ejecución de cada historia de usuario. Por tanto, el conjunto de trabajo evalúa individualmente la caracterización en función del riesgo identificado en su elaboración, como, por ejemplo:

- **Alta:** Cuando se lleva a cabo la implementación de la historia de usuario, se tiene en cuenta la posibilidad de que se produzcan fallos que puedan resultar en la ineficacia del código.
- **Media:** Cuando pueden ocurrir errores durante la implementación de la historia de usuario, lo cual podría generar demoras en la entrega de la versión final.
- **Baja:** Cuando existen casos en los que pueden surgir errores que se manejarán con relativa facilidad sin causar daños significativos en el desarrollo del proyecto.

En relación a la información incluida en la historia de usuario, hay varias plantillas sugeridas, aunque no hay un consenso definitivo al respecto. En este caso, se sugiere utilizar las siguientes plantillas:

- **Número:** Esta sección representa el número, incremental en el tiempo, de la HU que se describe.
- **Nombre de Historia de Usuario:** Reconoce la HU que se describe entre el cliente y los desarrolladores.
- **Usuario:** Rol de usuario que realiza la funcionalidad.
- **Prioridad de negocios:** Se le otorga a la HU una prioridad (Alta, Media, Baja) de acuerdo a las necesidades en cuanto a desarrollo.
- **Riesgo en Desarrollo:** Otorga a la HU una medida de (Alto, Medio, Bajo) la ocurrencia de errores en el proceso de desarrollo.
- **Iteración asignada:** Número de la iteración donde se desea desarrollar la HU.
- **Puntos estimados:** Tiempo estimado en semanas ideales (40 horas) que se demorará el desarrollo de la HU.
- **Descripción:** Descripción breve de la HU.
- **Observaciones:** Señalamiento o advertencia del sistema.

Tabla 1 HU 1 Autenticar Usuario

Número: 1	Nombre: Autenticar Usuario
Prioridad: Alta	Iteración asignada: 1
Riesgo en desarrollo: Medio	Tiempo estimado: 0.2
Usuario: Administrador, Estudiante y Profesor	

Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon
Descripción: En la historia de usuario "Autenticar", el sistema verifica la validez de las credenciales ingresadas y permite el acceso solo si son correctas. Si las credenciales son incorrectas, el sistema muestra un mensaje de error adecuado: "Las credenciales son incorrectas". Una vez autenticado, el usuario tiene acceso a las funcionalidades y características correspondientes a su rol o perfil de usuario.
Observaciones: Ninguna

Tabla 2 HU 2 Gestionar Usuario

Número: 2		Nombre: Gestionar Usuario	
Prioridad: Alta		Iteración asignada: 1	
Riesgo en desarrollo: Medio		Tiempo estimado: 0.5	
Usuario: Administrador			
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon			
Descripción: El sistema permite al administrador gestionar los datos de cada uno de los usuarios, en correspondencia con su nombre de usuario y contraseña.			
Observaciones: Ninguna			

Tabla 3 HU 3 Gestionar Estudiante

Número: 3		Nombre: Gestionar Estudiante	
Prioridad: Alta		Iteración asignada: 1	
Riesgo en desarrollo: Medio		Tiempo estimado: 1	
Usuario: Administrador			

Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon
Descripción: El sistema permite al administrador gestionar los datos de cada uno de los estudiantes, que constan de los siguientes atributos: id, nombre, apellidos, año, nombre_tesis.
Observaciones: Ninguna

Tabla 4 HU 4 Gestionar Profesor

Número: 4	Nombre: Gestionar Profesor
Prioridad: Alta	Iteración asignada: 1
Riesgo en desarrollo: Medio	Tiempo estimado: 1
Usuario: Administrador	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon	
Descripción: El sistema permite al administrador gestionar los datos de cada uno de los profesores, que constan de los siguientes atributos: id, nombre, apellido, rol, categoría, área, sean nuevos o existentes.	
Observaciones: Ninguna	

Tabla 5 HU 5 Gestionar Tribunales

Número: 5	Nombre: Gestionar Tribunales
Prioridad: Alta	Iteración asignada: 2
Riesgo en desarrollo: Medio	Tiempo estimado: 1
Usuario: Administrador	

Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon	
Descripción: El sistema permite al administrador gestionar los datos de cada uno de los Tribunales conformado por cuatro profesores los cuales tienen cargos de presidente, secretario, vocal y oponente, sean nuevos o existentes.	
Observaciones: Ninguna	

Tabla 6 HU 6 Gestionar Tesis

Número: 6	Nombre: Gestionar Tesis
Prioridad: Alta	Iteración asignada: 2
Riesgo en desarrollo: Medio	Tiempo estimado: 1
Usuario: Administrador y Profesores	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon	
Descripción: El sistema permite al administrador y profesores gestionar los datos de cada uno de las Tesis que tiene como atributos: id, nombre, fecha, tutor, resumen y url; sean estas tesis nuevas o existentes.	
Observaciones: Ninguna	

Tabla 7 HU 7 Gestionar Primer Corte

Número: 7	Nombre: Gestionar Primer Corte
Prioridad: Alta	Iteración asignada: 2
Riesgo en desarrollo: Medio	Tiempo estimado: 0.5
Usuario: Administrador y Profesores	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon	
Descripción: El sistema permite al administrador planificar y gestionar los datos del Primer Corte de Tesis. El sistema también permite a los profesores gestionar dichos	

datos. Este corte de tesis cuenta con los siguientes atributos: evaluación y recomendaciones.
Observaciones: Ninguna

Tabla 8 HU 8 Gestionar Segundo Corte

Número: 8	Nombre: Gestionar Segundo Corte
Prioridad: Alta	Iteración asignada: 3
Riesgo en desarrollo: Medio	Tiempo estimado: 0.5
Usuario: Administrador y Profesores	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon	
Descripción: El sistema permite al administrador planificar y gestionar los datos del Segundo Corte de Tesis. El sistema también permite a los profesores gestionar dichos datos. Este corte de tesis cuenta con los siguientes atributos: evaluación y recomendaciones.	
Observaciones: Ninguna	

Tabla 9 HU 9 Gestionar Pre-defensa

Número: 9	Nombre: Gestionar Pre-defensa
Prioridad: Alta	Iteración asignada: 3
Riesgo en desarrollo: Medio	Tiempo estimado: 1
Usuario: Administrador y Profesores	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon	
Descripción: El sistema permite al administrador planificar y gestionar los datos de la Pre-defensa. El sistema también permite a los profesores gestionar dichos datos. Esta Pre-defensa cuenta con los siguientes atributos: nota y recomendaciones.	

Observaciones: Ninguna

Tabla 10 HU 10 Gestionar Defensa

Número: 10	Nombre: Gestionar Defensa
Prioridad: Alta	Iteración asignada: 3
Riesgo en desarrollo: Medio	Tiempo estimado: 1
Usuario: Administrador y Profesores	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon Descripción: El sistema permite al administrador planificar y gestionar los datos de la Defensa de Tesis. El sistema también permite a los profesores gestionar dichos datos. Esta Defensa de tesis cuenta con los siguientes atributos: nota.	
Observaciones: Ninguna	

Tabla 11 HU 11 Buscar Tesis

Número: 11	Nombre: Buscar Tesis
Prioridad: Alta	Iteración asignada: 3
Riesgo en desarrollo: Medio	Tiempo estimado: 0.2
Usuario: Administrador, Profesores y Estudiantes	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon Descripción: El sistema permite buscar una tesis entre las ya existentes en el mismo.	
Observaciones: Ninguna	

Tabla 12 HU 12 Buscar Usuario

Número: 12	Nombre: Buscar Usuario	
Prioridad: Alta		Iteración asignada: 4
Riesgo en desarrollo: Medio		Tiempo estimado: 0.2
Usuario: Administrador y Profesores		
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon		
Descripción: El sistema permite buscar un usuario entre los ya existentes en el mismo.		
Observaciones: Ninguna		

Tabla 13 HU 13 Buscar Tribunal

Número: 13		Nombre: Buscar tribunal	
Prioridad: Alta		Iteración asignada: 4	
Riesgo en desarrollo: Medio		Tiempo estimado: 0.2	
Usuario: Administrador y Profesores			
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon			
Descripción: El sistema permite buscar un tribunal entre los ya existentes en el mismo.			
Observaciones: Ninguna			

Tabla 14 HU 14 Agregar Fichero

Número: 14	Nombre: Agregar fichero

Prioridad: Alta	Iteración asignada: 4
Riesgo en desarrollo: Medio	Tiempo estimado: 0.2
Usuario: Administrador, Profesores y Estudiantes	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon	
Descripción: El sistema permite subir un fichero con la información de la tesis.	
Observaciones: Ninguna	

Tabla 15 HU 15 Repositorio de tesis

Número: 15	Nombre: Repositorio de tesis
Prioridad: Alta	Iteración asignada: 4
Riesgo en desarrollo: Medio	Tiempo estimado: 0.5
Usuario: Administrador, Profesores y Estudiantes	
Programador: Ricardo Adrian Avila Rodriguez y Guido Alejandro Morales Bordon	
Descripción: El sistema permite acceder a un repositorio donde se encuentran los url de las tesis.	
Observaciones: Ninguna	

II.4 Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días (Metodologías ágiles para el desarrollo de software: Xtreme Programming (XP), 2006).

II.4.1 Estimación del esfuerzo

Con el propósito de saber cuánto le tomará al equipo de desarrollo elaborar las historias de usuario en su totalidad y comprender cuántos puntos debe desarrollar, se realiza una estimación de esfuerzos por cada HU. Se establecieron 4 iteraciones.

Tabla 16 Estimación de esfuerzos

Iteración	Historias de usuarios		Puntos estimados(semanas)
1	1	Autenticar Usuario	0.3
	2	Gestionar Usuario	0.5
	3	Gestionar Estudiante	1
	4	Gestionar Profesor	1
2	5	Gestionar Tribunales	1
	6	Gestionar Tesis	1
	7	Gestionar Primer Corte	1
	8	Gestionar Segundo Corte	0.5
3	9	Gestionar Pre-defensa	1
	10	Gestionar Defensa	1
	11	Buscar Tesis	0.5
	12	Buscar Usuario	0.3
4	13	Buscar Tribunal	0.3
	14	Agregar Fichero	0.3
	15	Repositorio de Tesis	0.3
Total			10

II.4.2 Plan de iteraciones

Después de ser identificadas y descritas las características y estimar el esfuerzo dedicado a la realización de cada una de estas, en la metodología elegida (XP) se procede a generar el plan de iteraciones. Este plan muestra cuáles son las características que serán implementadas en

cada iteración para mejorar el rendimiento del equipo de desarrollo. Al finalizar las iteraciones, el sistema estará preparado para pasar a manos de producción.

Después de esto, se procede a planificar la etapa de implementación. En esta etapa, las características se desarrollarán y se aprobarán en un ciclo iterativo. Con el objetivo de fomentar un mayor crecimiento y rendimiento en la organización del equipo de desarrollo, se establece un total de 4 iteraciones.

Iteración 1:

En esta iteración se implementarán las características con un nivel de prioridad Alta, la cual estará vinculada con la fase inicial de implementación del proceso de gestión de ejercicios de culminación de estudios. Además, el cliente tendrá la opción de verificar algunas de las funcionalidades descritas personalmente, a través de la primera versión que le suministrará el equipo de desarrollo. Dentro de estas funcionalidades podrá encontrar: Autenticar Usuario, Gestionar Usuario, Gestionar Estudiante y Gestionar Profesor, descritas en las HU 1, 2, 3, 4.

Iteración 2.

En esta iteración se llevarán a cabo las HU con un nivel de prioridad Alta, la cual estará relacionada con la etapa intermedia del desarrollo del sistema, con funcionalidades importantes y se rectificarán los errores que haya sido encontrados en la anterior iteración. En esta iteración se implementarán las HU 5, 6, 7 y 8 las cuales corresponden a Gestionar Tribunales, gestionar Tesis, Gestionar Primer Corte y Gestionar Segundo Corte.

Iteración 3.

En esta iteración se realizarán las HU con un nivel de prioridad Alta y Media, las cuales estarán relacionadas con la etapa intermedia del desarrollo del sistema, con funcionalidades igual de importantes y se rectificarán los errores que haya sido encontrados en la anterior iteración. En esta iteración se implementarán las HU 9, 10, 11 y 12 las cuales corresponden a Gestionar Pre-Defensa, Gestionar Defensa, Buscar Tesis y Buscar Usuario.

Iteración 4.

En esta iteración se ejecutarán las HU con un nivel de prioridad media y baja, completando el proceso de desarrollo del sistema. Se obtendrán las primeras versiones del sistema informático para la Gestión de la Información de los Ejercicios de Culminación de Estudio en la Facultad

2, implementándose las HU Buscar Tribunal, Agregar Fichero y Repositorio de Tesis correspondientes a las HU 13, 14 y 15. Una vez finalizada el cliente podrá realizar una revisión total de las funcionalidades implementadas y a su vez probar el sistema.

Tabla 17 Plan de iteraciones / cantidad de semanas

Iteración	Historias de usuarios		Cantidad de semanas
1	1	Autenticar Usuario	2.8
	2	Gestionar Usuario	
	3	Gestionar Estudiante	
	4	Gestionar Profesor	
2	5	Gestionar Tribunales	3.5
	6	Gestionar Tesis	
	7	Gestionar Primer Corte	
	8	Gestionar Segundo Corte	
3	9	Gestionar Pre-defensa	2.8
	10	Gestionar Defensa	
	11	Buscar Tesis	
	12	Buscar Usuario	
4	13	Buscar Tribunal	0.9
	14	Agregar Fichero	
	15	Repositorio de Tesis	
Total			10

II.4.3 Plan de entregas

Con el fin de mantener al cliente centrado en mejorar el producto, y a su vez hacer que sea el propio cliente quien realice las pruebas pertinentes y para ello se realizan varias entregas pro-

gramadas. A continuación, se presenta una tabla que refleja lo antes mencionado, se especifica la fecha en la cual se realiza el encuentro entre el equipo de desarrollo y el cliente o propietarios del software, para así mostrarles las versiones del producto.

Tabla 18 Plan de entrega

Iteración	Fecha de entrega
Iteración 1	22 de septiembre de 2023
Iteración 2	19 de octubre de 2023
Iteración 3	10 de noviembre de 2023
Iteración 4	21 de noviembre de 2023

II.5 Diseño

El enfoque incremental y evolutivo de XP promueve el diseño y desarrollo de solo aquellas historias de usuario priorizadas por el cliente para cada iteración, evitando intentos iniciales de diseñar el sistema completo. En primer lugar, se reconoce que un diseño total del producto desde el inicio probablemente contenga errores. Además, ante la naturaleza cambiante de los proyectos, realizar un diseño muy amplio en las primeras fases podría resultar en trabajo desperdiciado si luego requiere modificaciones. Del mismo modo que con el diseño, la mejora continua es esencial durante todo el ciclo de vida del proyecto XP, iniciando con un bosquejo inicial que es revisado y optimizado a lo largo del desarrollo. **Invalid source specified.**

Extreme Programming promueve un enfoque de diseño evolutivo mediante iteraciones cortas donde se refactoriza un sistema inicialmente simple con cada incremento. El diseño se centra únicamente en las historias de usuario especificadas para la iteración actual, sin intentar anticipar requerimientos futuros. Esta metodología iterativa logra equilibrar planificación y capacidad de adaptación, permitiendo mejoras continuas al producto a lo largo del proyecto. Un enfoque como este hace que XP sea valorada como una de las metodologías ágiles más eficaces para gestionar la incertidumbre inherente al desarrollo de software. (Highsmith, 2002)

II.5.1 Patrón arquitectónico

Los patrones arquitectónicos establecen la organización fundamental de una aplicación al definir una plantilla básica de construcción. Proporcionan subsistemas predefinidos junto con directrices para su interconexión y disposición, expresando de esta forma una estructura general para el sistema. Esto permite generar componentes y subsistemas de manera guiada. Los patrones arquitectónicos logran una alta abstracción al formalizar lineamientos de uso, lo cual facilita el diseño de soluciones mediante la aplicación de soluciones probadas previamente. (Buschmann et al., 1996)

Para el desarrollo de este proyecto se desea utilizar como patrón arquitectónico el siguiente a explicar:

Modelo Vista Controlador

El patrón de arquitectura MVC (Modelo-Vista-Controlador) divide los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para lograr esto, MVC propone la construcción de tres componentes distintos: el modelo, la vista y el controlador. El modelo se encarga de representar la información, mientras que la vista se ocupa de la interacción con el usuario. El controlador actúa como el intermediario entre el modelo y la vista. Este enfoque de diseño se basa en la reutilización de código y la separación de conceptos, lo cual busca simplificar el desarrollo y el mantenimiento de aplicaciones (Patrones de diseño en la arquitectura de software: MVC, 2018).

Modelo: Es la representación de los datos con los cuales el sistema interactúa, por lo tanto, administra todos los accesos a esa información, tanto las consultas como las actualizaciones, implementando además los permisos de acceso que se hayan descrito en las especificaciones.

Controlador: Responde a eventos e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información.

Vista: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida. Esto implica que el modelo debe ser capaz de proporcionar los datos necesarios para mostrar al usuario de manera clara y comprensible, ya sea a través de elementos visuales, tablas, gráficos u otros medios interactivos. Además, el modelo también debe ser capaz de procesar y manejar las acciones del usuario, como entradas de

datos o selecciones, y aplicar la lógica de negocio correspondiente para generar los resultados esperados. En resumen, el 'modelo' es el núcleo del sistema que se encarga de gestionar la información y la lógica de negocio, y su capacidad para interactuar de manera efectiva con la interfaz de usuario es fundamental para brindar una experiencia de usuario satisfactoria.

Ventajas de usar el MVC

1. Se puede tener diferentes vistas para un mismo modelo.
2. Facilita crear nuevas visitas sin que se modifique el modelo subyacente.
3. Contiene un mecanismo de configuración más tratable a componentes complejos que el basado en eventos (El modelo puede representarse como un estado de interacción con una forma estructurada).
4. La conectividad del Modelo y sus vistas es dinámica, ya que se produce en tiempo de ejecución, no de implementación.**Invalid source specified.**

II.5.2 Patrón de diseño

Un patrón de diseño se define como una regla de tres partes que establece una relación entre un contexto específico, un problema y una solución. En el diseño de software, el contexto proporciona información sobre el entorno en el que se encuentra el problema y cuál sería la solución adecuada en ese entorno. Un conjunto de requisitos, que incluye limitaciones y restricciones, actúa como un sistema de fuerzas que influye en la forma en que se interpreta el problema en este contexto y en cómo se puede aplicar de manera efectiva la solución. Esta perspectiva ayuda a comprender la importancia de considerar el contexto y los requerimientos al aplicar los patrones de diseño en el desarrollo de software (Pressman, 2010).

Los patrones de diseño son soluciones probadas y documentadas para problemas recurrentes en el diseño de software. Proporcionan un enfoque estructurado y reutilizable para abordar desafíos comunes en el desarrollo de sistemas. Los patrones de diseño permiten a los desarrolladores aprovechar las mejores prácticas y la experiencia acumulada en el diseño de software, lo que resulta en sistemas más eficientes y mantenibles. Estos patrones se basan en la idea de capturar soluciones exitosas a problemas comunes y documentarlas en un formato comprensible. Al utilizar patrones de diseño, los desarrolladores pueden evitar repetir errores

y aplicar soluciones probadas en situaciones similares. Esto promueve la reutilización del diseño y acelera el proceso de desarrollo. Además, los patrones de diseño fomentan el modularidad y la flexibilidad en el diseño de software. Al dividir el sistema en componentes interconectados y asignar responsabilidades claras a cada uno, los patrones de diseño facilitan la comprensión y el mantenimiento del código, así como la adaptación a cambios futuros (Freeman y Robson, 2004).

El sistema que se desea desarrollar cuenta con una arquitectura basada en 3 capas, usando a su vez patrones arquitectónicos y de diseño, para ello se desean utilizar los patrones GRASP y GOF.

Patrones GRASP

Los patrones GRASP son patrones generales de software para asignación de responsabilidades. Estos patrones están interrelacionados y se complementan entre sí. El principio del Experto es clave en los patrones GRASP, asignando responsabilidades al objeto con la información necesaria. Además, promueven la alta cohesión entre objetos y consideran el nivel de acoplamiento en conjunto con otros principios. Los patrones GRASP no son soluciones prescriptivas, sino principios flexibles. En resumen, los patrones GRASP ayudan a asignar responsabilidades de manera efectiva, considerando la experticia y promoviendo la cohesión (Larman, 2002).

De los patrones *GRASP* a utilizar se tiene:

Controlador: Es un objeto encargado de recibir o manejar eventos del sistema, y no pertenece a la interfaz de usuario. Su responsabilidad principal es definir el método para la operación del sistema (Sánchez, 2018).

Experto: Este principio establece que la responsabilidad debe asignarse al objeto que posee la información necesaria para cumplir con dicha responsabilidad de manera más experta. Se basa en la idea de asignar la responsabilidad a la clase que tiene el conocimiento necesario (Gonzalez, 2018).

```

app.post("/insertarusuario", async (req, res) => {
  const nombreusuario = req.body.nombreusuario;
  const contrasena = req.body.contrasena;
  let admin = req.body.admin;
  if(admin=="Si"){
    admin=true;
  }else{
    admin=false;
  }
});

```

Figura 2 Ejemplo de Patrón Experto

Creador: El patrón Creador ayuda a determinar quién debe ser responsable de crear nuevos objetos o clases. Este patrón también muestra la relación de visibilidad entre la clase creada y la clase creadora. Al seguir este principio, se facilita el mantenimiento y la reutilización del código (García, 2018).

```

const onSubmit = handleSubmit(async (data) => {
  if (params.tesisId) {
    const formData = {
      nombre: data.nombre,
      fecha: data.fecha,
      resumen: data.resumen,
      cotutor: data.cotutor,
      url: data.url,
      profesor: data.profesor.value,
      tribunal: data.tribunal.value,
      est: data.estudiante.value,
    };
    await updateDocument(params.tesisId, formData);
  } else {
    const formData = {
      nombre: data.nombre,
      fecha: data.fecha,
      resumen: data.resumen,
      cotutor: data.cotutor,
      url: data.url,
      profesor: data.profesor.value,
      tribunal: data.tribunal.value,
      est: data.estudiante.value,
    };
    const res = await addDocument(formData);
    console.log(res);
  }
  navigate("/tesis");
});
console.log(estudianteActual);
console.log(tutorActual);
console.log(tribunalActual);
return (

```

Figura 3 Ejemplo de Patrón Creador

Patrones GoF

Los patrones GoF fueron introducido popularmente a la industria del software con el libro Design Patterns: Elements of reusable object oriented programming escrito por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en 1995. Estos introducen estructuras de clases e interfaces que solucionan un problema general, pensado para que las aplicaciones robustas sean escalables y flexibles, además rápidas de desarrollar y diseñar. **Invalid source specified.**

El empleado en este trabajo de diploma es:

Singleton: está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. No se encarga de la creación de objetos en sí, sino que se enfoca en la restricción en la creación de un objeto.

Decorator: Este patrón es utilizado para agregar responsabilidades a un objeto dinámicamente, lo que puede proporcionar una alternativa flexible a la herencia para extender la funcionalidad.

II.5.3 Tarjetas CRC

La metodología XP usa la técnica de las tarjetas CRC como una extensión informal de UML como representación del sistema.

Una tarjeta CRC representa una clase en la programación orientada a objetos. Son utilizadas por la metodología XP para diseñar el sistema uniendo todas las funcionalidades. Además, ayudan a definir actividades mientras se está diseñando el sistema. **Invalid source specified.**

La elaboración de las tarjetas CRC es una técnica que está orientada a establecer un planteamiento inicial de los objetos que participan en el sistema directamente. El producto generado por el uso de esta técnica, es un conjunto de tarjetas asociadas a las clases identificadas para el sistema, en las que se especifican las responsabilidades y las colaboraciones con otras clases.

Las tarjetas CRC constan de las siguientes partes:

- Nombre de la clase.
- Responsabilidades: son sentencias de alto nivel acerca de las acciones que realiza un objeto como también del conocimiento que mantiene y provee.
- Colaboradores: son aquellas clases con las cuales se van a trabajar.

Crear tarjetas CRC tiene como ventajas:

- Permiten contribuir al diseño de todo el equipo del proyecto.
- Permiten reducir el modo de pensar procedural y apreciar la tecnología de objetos.
- Son simples, fáciles de comprender y flexibles de realizarlas.
- Se visualiza en forma resumida todo el funcionamiento de una clase. **Invalid source specified.**

A continuación, se presentan las tarjetas CRC elaboradas en función del sistema a desarrollar:

Tabla 19 Tarjeta CRC #1 Estudiante

Tarjeta CRC	
Clase: Estudiante	
Responsabilidad	Colaboración
EstudianteList	estudiante_list Estudiante
AddEstudiante	estudiante_add EstudianteForm
EstudianteUpdate	estudiante_update Estudiante
EstudianteDelete	estudiante_delete Estudiante

Tabla 20 Tarjeta CRC #2 Profesor

Tarjeta CRC	
Clase: Profesor	
Responsabilidad	Colaboración
ProfesorList	profesor_list Profesor

ProfesorAdd	profesor_add ProfesorForm
ProfesorUpdate	profesor_update Profesor
ProfesorDelete	profesor_delete Profesor

Tabla 21 Tarjeta CRC #3 Tribunal

Tarjeta CRC	
Clase: Tribunal	
Responsabilidad	Colaboración
TribunalList	tribunal_list Tribunal
TribunalAdd	tribunal_add TribunalForm
TribunalUpdate	tribunal_update Tribunal
TribunalDelete	tribunal_delete Tribunal

Tabla 22 Tarjeta CRC #4 Primer Corte

Tarjeta CRC	
Clase: PrimerCorte	
Responsabilidad	Colaboración
PrimerCorteList	pcorte_list PrimerCorte
PrimerCorteAdd	pcorte_add PrimerCorteForm
PrimerCorteUpdate	pcorte_update

	PrimerCorte
PrimerCorteDelete	pcorte_delete PrimerCorte

Tabla 23 Tarjeta CRC #5 Segundo Corte

Tarjeta CRC	
Clase: SegundoCorte	
Responsabilidad	Colaboración
SegundoCorteList	scorte_list SegundoCorte
SegundoCorteAdd	scorte_add SegundoCorteForm
SegundoCorteUpdate	scorte_update SegundoCorte
SegundoCorteDelete	scorte_delete SegundoCorte

Tabla 24 Tarjeta CRC #6 Pre-defensa

Tarjeta CRC	
Clase: Pre-defensa	
Responsabilidad	Colaboración
PredefensaList	predefensa_list Predefensa
PredefensaAdd	predefensa_add PredefensaForm
PredefensaUpdate	predefensa_update Predefensa
PredefensaDelete	predefensa_delete Predefensa

Tabla 25 Tarjeta CRC #7 Defensa

Tarjeta CRC	
Clase: Defensa	
Responsabilidad	Colaboración
DefensaList	defensa_list Defensa
DefensaAdd	defensa_add DefensaForm
DefensaUpdate	defensa_update Defensa
DefensaDelete	defensa_delete Defensa

II.5.4 Modelo de datos

Luego de ser diseñadas las tarjetas CRC, y definidas las clases entidades, se diseña el modelo de datos que usará el sistema para almacenar la información.

El modelo de datos permite describir las principales clases que componen la capa modelo del sistema, el tipo de dato de los atributos que contendrá la base de datos y la forma en que se relacionan las entidades. Permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que estos se relacionan.

A continuación, se muestra el diagrama entidad-relación que representa el modelo físico de datos diseñados para el sistema.

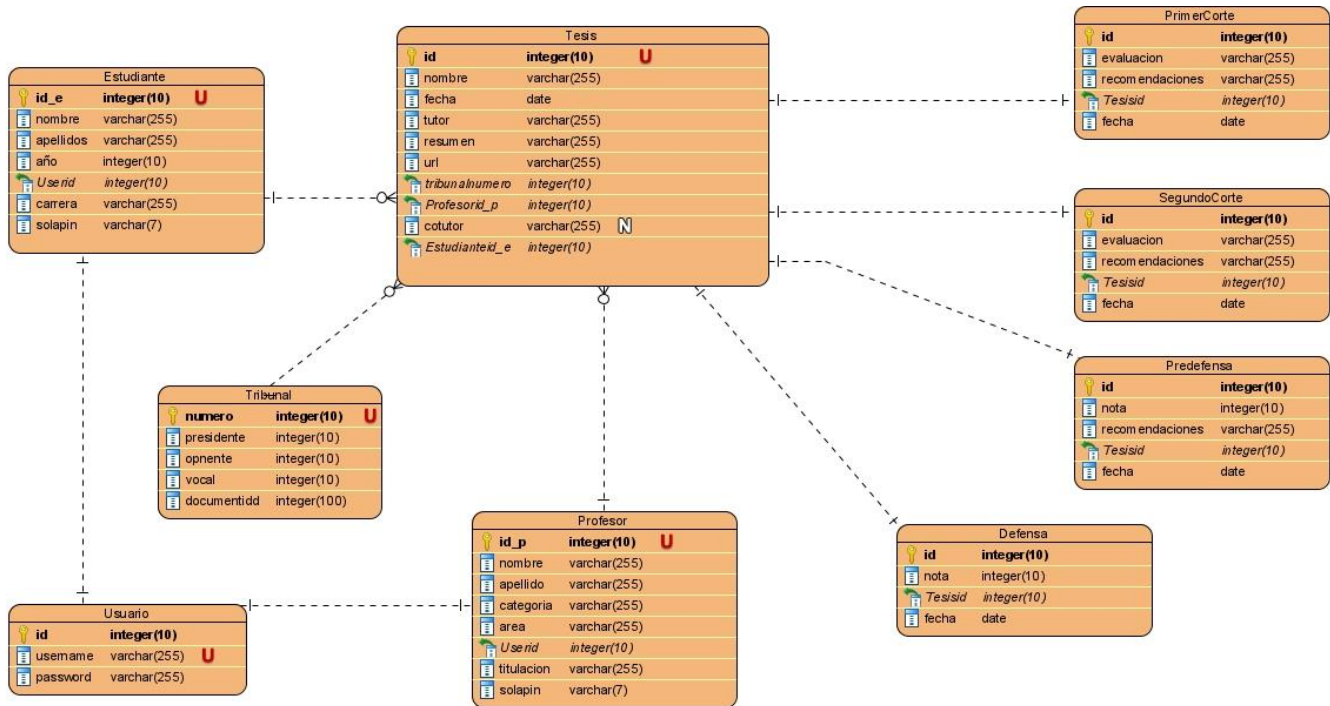


Figura 4 Modelo de datos

Conclusiones del capítulo

En este capítulo se detallaron las etapas que atraviesa la metodología XP.

- Durante la etapa de Exploración, se logró la generación de las Historias de Usuario (HU) de acuerdo con las funcionalidades solicitadas por el cliente, lo cual facilita una mejor comprensión del proceso de desarrollo.
- En la etapa de Planificación, se llevó a cabo la estimación de los esfuerzos y la planificación de las entregas basadas en las iteraciones definidas, lo que permite establecer un plazo para la liberación de las versiones y así el cliente pueda verificar el progreso del proyecto que desea obtener del equipo de desarrollo.
- En la etapa de Diseño, se optó por emplear el patrón arquitectónico MVC, que asegura una solución basada en la reutilización del código y la resolución de problemas similares en el desarrollo de software. Además, se crearon las tarjetas CRC como artefacto principal de esta fase, lo cual permite definir las clases y las relaciones entre ellas a través de sus colaboradores y responsabilidades, logrando un diseño ordenado.

Se puede concluir que la metodología XP es una alternativa eficaz para el desarrollo del proyecto, ya que se enfoca en la satisfacción del cliente, la comunicación constante entre el equipo

de desarrollo y el cliente, y la entrega de software funcional en iteraciones cortas. Las fases de Exploración, Planificación y Diseño son fundamentales para el éxito del proyecto, ya que permiten una comprensión clara de las necesidades del cliente, una planificación adecuada de los esfuerzos y entregas, y un diseño ordenado y reutilizable

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se describen las fases de implementación y validación como últimas a llevar a cabo para darle solución al sistema propuesto en la investigación. Durante la fase de implementación, se establecen las directrices relativas al código y se detallan las tareas de ingeniería que constituyen el artefacto generado por la metodología XP para detallar las Historias de Usuario identificadas en el capítulo anterior. Luego se procede a realizarle la estrategia de pruebas, para verificar si el sistema cumple sus requisitos antes de la entrega final al cliente. Dentro de esta estrategia, la metodología XP define que se deben realizar pruebas unitarias y funcionales.

III.1 Implementación

La colaboración activa del cliente como parte integral del equipo de trabajo resulta fundamental en todas las etapas del enfoque de XP. Durante la fase de codificación e implementación de una HU, la presencia del cliente adquiere incluso mayor relevancia, debido a que son ellos quienes proponen y definen las HU, negocian los plazos para su implementación, brindan detalles precisos sobre su funcionalidad y verifican que la historia implementada cumpla con los requisitos específicos durante las pruebas. El proceso de codificación se rige por estándares establecidos, y los programadores deben seguirlos de manera rigurosa para garantizar que el código del sistema se presente como si hubiera sido desarrollado por una única persona. Esta coherencia en el código facilita su comprensión y mantenimiento.

III.2 Estándares de codificación

La adopción de estándares se considera una práctica altamente beneficiosa en el ámbito del desarrollo de software, no solo en el contexto de la metodología XP, sino también en la aplicación de otros enfoques metodológicos. La implementación de estándares tiene como objetivo principal mejorar la comprensión del código por parte del equipo de desarrollo. Dentro del marco de XP, se fomenta la programación basada en estándares con el propósito de asegurar que el código sea comprensible para todo el equipo y facilite la tarea de realizar modificaciones posteriores. XP hace hincapié en la comunicación entre los programadores a través del código, lo que subraya la necesidad de seguir pautas específicas de programación.

- Las variables, funciones, paquetes y módulos llevan el nombre de 'lower_with_under' (Minúsculas y subrayado), como `model_test`

```
@property
def solapin_Estudiante(self):
    return 'E-' + "{:04}".format(self.solapin)
```

Figura 5 Estándares de codificación1

- Las clases y excepciones llevan el nombre de CapWords (CamelCase: primera letra en mayúscula) como `ModelTest`

```
class Documento(models.Model):
    nombre = models.CharField(max_length=255, verbose_name='Nombre')
    fecha = models.DateField(verbose_name='Fecha')
    resumen = models.TextField(verbose_name='Resumen')
    profesor = models.ForeignKey('Profesor', default=None, on_delete=models.CASCADE, verbose_name='Profesor')
```

Figura 6 Estándares de codificación2

- Al construir listas, diccionarios, tuplas, se deben agregar espacios al final de las comas, punto y coma y dos puntos, y no se deben agregar espacios al principio.

```
presidente = models.CharField(max_length=100, verbose_name='Presidente')
```

Figura 7 Estándares de codificación3

III.3 Tareas de ingeniería

Las tareas de ingeniería (TI) son el artefacto que se encarga en la fase de implementación y pruebas de almacenar toda la información detallada de cada una de las HU. Una HU puede contener más de una TI dependiendo del nivel de complejidad de dicha HU. Cuando el equipo de desarrollo conforma las TI deja bien claro en sus campos a que HU pertenece y los programadores responsables de implementarlas, estableciendo a su vez fecha de inicio y de fin en correspondencia con la información que aporta la HU en cuestión. El tiempo estimado indica el tiempo real que utilizarán los programadores en realizar la funcionalidad, se especifica además el tipo de tarea a realizar (desarrollo, corrección o mejora) y el nombre de la misma,

así como la descripción de la tarea; en este campo como su nombre lo indica se detalla todo lo que se va a realizar y en lo que se van a apoyar para darle solución a la funcionalidad correspondiente.

Tabla 26 TI 1 Autenticar Usuario

Número de la tarea: 1	Nombre de la tarea: Autenticar usuario
Identificador de la HU: 1	Tipo de tarea: Desarrollo
Fecha de inicio: 01/09/2023	Fecha de fin: 04/09/2023
Tiempo estimado: 0.3	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon
Descripción: Todos los usuarios podrán acceder a una vista en donde aparecerá un formulario de dos campos en donde se rellenan con usuario y contraseña respectivamente y dependiendo de cada usuario se le dará o no mas o menos permisos en el sistema	

Tabla 27 TI 2 Gestionar Usuario

Número de la tarea: 2	Nombre de la tarea: Gestionar Usuario
Identificador de la HU: 2	Tipo de tarea: Desarrollo
Fecha de inicio: 05/09/2023	Fecha de fin: 07/09/2023
Tiempo estimado: 0.5	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon
Descripción: El administrador selecciona de la barra de navegación la opción 'Gestionar Usuario', el sistema muestra un menú desplegable en donde se apreciarán 3 opciones las cuales serían 'Gestionar Estudiantes', 'Gestionar Profesores' y 'Gestionar Tribunales'.	

Tabla 28 TI 3 Gestionar Estudiante

Número de la tarea: 3	Nombre de la tarea: Gestionar Estudiante
Identificador de la HU: 3	Tipo de tarea: Desarrollo
Fecha de inicio: 08/09/2023	Fecha de fin: 14/09/2023
Tiempo estimado: 1	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon
Descripción: El administrador selecciona de la barra de navegación la opción ‘Gestionar Usuarios’, luego del menú desplegable selecciona la opción ‘Gestionar Estudiantes’ el sistema muestra una vista en donde aparecerá el botón ‘Añadir Estudiante’ el cual será presionado para poder acceder al formulario donde el administrador podrá llenar los campos con información válida de cada estudiante a insertar. En conjunto al botón , aparecerá una tabla de estudiantes en donde se apreciarán dos botones en cada fila, ‘Eliminar’ que sería para eliminar el estudiante referente a la fila correspondiente y el botón ‘Editar’ el cual mostrará un formulario auto completado con los datos del estudiante y podrán ser modificados sus datos y guardados al presionar el botón ‘Guardar’, si no se desea modificar ni agregar información a la hora de agregar un estudiante se puede pulsar el botón ‘Cancelar’ para abortar cualquier acción.	

Tabla 29 TI 4 Gestionar Profesor

Número de la tarea: 4	Nombre de la tarea: Gestionar Profesor
Identificador de la HU: 4	Tipo de tarea: Desarrollo
Fecha de inicio: 15/09/2023	Fecha de fin: 21/09/2023
Tiempo estimado: 1	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon

Descripción: El administrador selecciona de la barra de navegación la opción 'Gestionar Usuarios', luego del menú desplegable selecciona la opción 'Gestionar Profesores' el sistema muestra una vista en donde aparecerá el botón 'Añadir Profesor' el cual será presionado para poder acceder al formulario donde el administrador podrá llenar los campos con información válida de cada profesor a insertar. En conjunto al botón , aparecerá una tabla de profesores en donde se apreciarán dos botones en cada fila, 'Eliminar' que sería para eliminar el profesor referente a la fila correspondiente y el botón 'Editar' el cual mostrará un formulario auto completado con los datos del profesor y podrán ser modificados sus datos y guardados al presionar el botón 'Guardar', si no se desea modificar ni agregar información a la hora de agregar un profesor se puede pulsar el botón 'Cancelar' para abortar cualquier acción.

Tabla 30 TI 5 Gestionar Tribunales

Número de la tarea: 5		Nombre de la tarea: Gestionar Tribunales	
Identificador de la HU: 5		Tipo de tarea: Desarrollo	
Fecha de inicio: 25/09/2023		Fecha de fin: 29/09/2023	
Tiempo estimado: 1		Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon	
Descripción: El administrador selecciona de la barra de navegación la opción ‘Gestionar Usuarios’, luego del menú desplegable selecciona la opción ‘Gestionar Tribunales’ el sistema muestra una vista en donde aparecerá el botón ‘Añadir Tribunal’ el cual será presionado para poder acceder al formulario donde el administrador podrá llenar los campos con información válida de cada tribunal a insertar. En conjunto al botón , aparecerá una tabla de tribunales en donde se apreciaran dos botones en cada fila, ‘Eliminar’ que sería para eliminar el tribunal referente a la fila correspondiente y el botón ‘Editar’ el cual mostrará un formulario auto completado con los datos del tribunal y podrán ser modificados sus datos y guardados al presionar el botón			

‘Guardar’, si no se desea modificar ni agregar información a la hora de agregar un tribunal se puede pulsar el botón ‘Cancelar’ para abortar cualquier acción.

Tabla 31 TI 6 Gestionar Tesis

Número de la tarea: 6		Nombre de la tarea: Gestionar Tesis	
Identificador de la HU: 6		Tipo de tarea: Desarrollo	
Fecha de inicio: 02/10/2023		Fecha de fin: 06/10/2023	
Tiempo estimado: 1		Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon	
Descripción: El administrador selecciona de la barra de navegación la opción ‘Gestionar Tesis’, el sistema muestra una vista en donde aparecerá el botón ‘Añadir Tesis’ el cual será presionado para poder acceder al formulario donde el administrador podrá llenar los campos con información válida de cada tesis a insertar. En conjunto al botón, aparecerá una tabla de tesis en donde se apreciarán dos botones en cada fila, ‘Eliminar’ que sería para eliminar la tesis referente a la fila correspondiente y el botón ‘Editar’ el cual mostrará un formulario auto completado con los datos de la tesis y podrán ser modificados sus datos y guardados al presionar el botón ‘Guardar’, si no se desea modificar ni agregar información a la hora de agregar una tesis se puede pulsar el botón ‘Cancelar’ para abortar cualquier acción.			

Tabla 32 TI 7 Gestionar Primer Corte

Número de la tarea: 7	Nombre de la tarea: Gestionar Primer Corte		
Identificador de la HU: 7		Tipo de tarea: Desarrollo	
Fecha de inicio: 09/10/2023		Fecha de fin: 13/10/2023	

Tiempo estimado: 1	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon
Descripción: El administrador selecciona de la barra de navegación la opción 'Documento', el sistema muestra una tarjeta con la información de la tesis a la que hace referencia como usuario y un menú desplegable más abajo en donde tendrá cuatro opciones y la primera será 'Primer Corte del Documento', dicha opción brindará al administrador la posibilidad de presionar el botón 'Agregar Primer Corte' y automáticamente aparecerá un formulario para rellenar los campos relacionados al primer corte. Conjunto a la información neceará aparecerán dos botones en donde se podrá accionar 'Guardar', guardando así la información relacionada y 'Cancelar' que regresará el formulario sin haberse ejecutado ningún cambio en la aplicación.	

Tabla 33 TI 8 Gestionar Segundo Corte

Número de la tarea: 8	Nombre de la tarea: Gestionar Segundo Corte
Identificador de la HU: 8	Tipo de tarea: Desarrollo
Fecha de inicio: 16/10/2023	Fecha de fin: 18/10/2023
Tiempo estimado: 0.5	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon
Descripción: El administrador selecciona de la barra de navegación la opción 'Documento', el sistema muestra una tarjeta con la información de la tesis a la que hace referencia como usuario y un menú desplegable más abajo en donde tendrá cuatro opciones y la primera será 'Segundo Corte del Documento', dicha opción brindará al administrador la posibilidad de presionar el botón 'Agregar Segundo Corte' y automáticamente aparecerá un formulario para rellenar los campos relacionados al primer corte. Conjunto a la información neceará aparecerán dos botones en donde se podrá accionar 'Guardar', guardando así la información relacionada y 'Cancelar' que regresará el formulario sin haberse ejecutado ningún cambio en la aplicación.	

Tabla 34 TI 9 Gestionar Pre-defensa

Número de la tarea: 9	Nombre de la tarea: Gestionar Pre-defensa	
Identificador de la HU: 9		Tipo de tarea: Desarrollo
Fecha de inicio: 20/10/2023		Fecha de fin: 26/10/2023
Tiempo estimado: 1	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon	
Descripción: El administrador selecciona de la barra de navegación la opción ‘Documento’, el sistema muestra una tarjeta con la información de la tesis a la que hace referencia como usuario y un menú desplegable más abajo en donde tendrá cuatro opciones y la primera será ‘Pre-defensa del Documento’, dicha opción brindará al administrador la posibilidad de presionar el botón ‘Agregar Pre-defensa y automáticamente aparecerá un formulario para rellenar los campos relacionados al primer corte. Conjunto a la información necearía aparecerán dos botones en donde se podrá accionar ‘Guardar’, guardando así la información relacionada y ‘Cancelar’ que regresará el formulario sin haberse ejecutado ningún cambio en la aplicación.		

Tabla 35 TI 10 Gestionar Defensa

Número de la tarea: 10	Nombre de la tarea: Gestionar Defensa	
Identificador de la HU: 10		Tipo de tarea: Desarrollo
Fecha de inicio: 27/10/2023		Fecha de fin: 02/11/2023
Tiempo estimado: 1	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon	
Descripción: El administrador selecciona de la barra de navegación la opción ‘Documento’, el sistema muestra una tarjeta con la información de la tesis a la que hace referencia como usuario y un menú desplegable más abajo en donde tendrá cuatro		

opciones y la primera será 'Defensa del Documento', dicha opción brindará al administrador la posibilidad de presionar el botón 'Agregar Defensa' y automáticamente aparecerá un formulario para rellenar los campos relacionados al primer corte. Conjunto a la información necearía aparecerán dos botones en donde se podrá accionar 'Guardar', guardando así la información relacionada y 'Cancelar' que regresará el formulario sin haberse ejecutado ningún cambio en la aplicación.

Tabla 36 TI 11 Buscar Tesis

Número de la tarea: 11	Nombre de la tarea: Buscar tesis
Identificador de la HU: 11	Tipo de tarea: Desarrollo
Fecha de inicio: 03/11/2023	Fecha de fin: 07/11/2023
Tiempo estimado: 0.5	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon
Descripción: En la esquina superior derecha de la barra de navegación aparece un sistema de búsqueda que consta de un campo de entrada donde se puede introducir datos en específico y validos de la búsqueda en cuestión y un botón que al pulsarlo devolverá todas las coincidencias entre los datos introducidos y la información existente en la base de datos	

Tabla 37 TI 12 Buscar Usuario

Número de la tarea: 12	Nombre de la tarea: Buscar usuario
Identificador de la HU: 12	Tipo de tarea: Desarrollo
Fecha de inicio: 08/11/2023	Fecha de fin: 10/11/2023
Tiempo estimado: 0.3	Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon

Descripción: En la esquina superior derecha de la barra de navegación aparece un sistema de búsqueda que consta de un campo de entrada donde se puede introducir datos en específico y validos de la búsqueda en cuestión y un botón que al pulsarlo devolverá todas las coincidencias entre los datos introducidos y la información existente en la base de datos

Tabla 38 TI 13 Buscar Tribunal

Número de la tarea: 13	Nombre de la tarea: Buscar tribunal
Identificador de la HU: 12	Tipo de tarea: Desarrollo
Fecha de inicio: 13/11/2023	Fecha de fin: 14/11/2023
Tiempo estimado: 0.3	Programador: Ricardo A. Avila Rodríguez y Guido A. Morales Bordon
Descripción: En la esquina superior derecha de la barra de navegación aparece un sistema de búsqueda que consta de un campo de entrada donde se puede introducir datos en específico y validos de la búsqueda en cuestión y un botón que al pulsarlo devolverá todas las coincidencias entre los datos introducidos y la información existente en la base de datos	

Tabla 39 TI 14 Agregar Fichero

Número de la tarea: 14	Nombre de la tarea: Agregar fichero
Identificador de la HU: 14	Tipo de tarea: Desarrollo
Fecha de inicio: 15/11/2023	Fecha de fin: 16/11/2023
Tiempo estimado: 0.3	Programador: Ricardo A. Avila Rodríguez y Guido A. Morales Bordon

Descripción: En cada uno de los cortes y la pre-defensa los usuarios tendrán la accesibilidad a un botón llamado adjuntar fichero, donde podrá subir y descargar el fichero correspondiente a cada uno de los cortes y pre-defensa de la tesis en cuestión.

Tabla 40 TI 15 Repositorio de Tesis

Número de la tarea: 15		Nombre de la tarea: Repositorio de tesis	
Identificador de la HU: 15		Tipo de tarea: Desarrollo	
Fecha de inicio: 17/11/2023		Fecha de fin: 20/11/2023	
Tiempo estimado: 0.3		Programador: Ricardo A. Avila Rodriguez y Guido A. Morales Bordon	
Descripción: En la barra de navegación el usuario podrá pulsar el botón ‘Documentos’ en donde automáticamente accederá a una tabla donde podrá observar cada uno de los documentos de la base de datos, además tendrá la posibilidad que al pulsar una fila accederá a la información específica de ese documento.			

III.4 Pruebas de software

La metodología XP se caracteriza por su enfoque en la realización de pruebas a lo largo de todo el proyecto, con el objetivo de garantizar la ejecución exitosa de lo planificado al inicio de cada iteración. Este proceso implica una estrecha colaboración entre el equipo de desarrollo y el cliente, quienes aportan sus conocimientos y perspectivas. En XP, las pruebas del sistema se dividen en dos grupos principales: las pruebas unitarias, responsabilidad de los programadores, que se enfocan en verificar el funcionamiento del código, y las pruebas funcionales, diseñadas por el cliente final, que evalúan si se logró la funcionalidad requerida al finalizar cada iteración. Estas pruebas funcionales representan una validación crucial para asegurar el cumplimiento de los objetivos establecidos.

“La prueba es el proceso de ejecución de un programa con la intención de encontrar errores”.
(Myers, 2004)

Según Cem Kaner, profesor de ingeniería de software define las pruebas de software como “la investigación empírica y técnica realizada para facilitar a los interesados información sobre la calidad del producto o servicios bajo pruebas”.

Según Edsger W. Dijkstra, científico de la computación define las pruebas de software como “una manera muy eficaz de mostrar la presencia de errores, pero son totalmente inadecuadas para mostrar su ausencia”.

III.4.1 Estrategia de pruebas

En la elaboración de la estrategia de pruebas, se consideró la propuesta de la metodología XP de llevar a cabo un amplio conjunto de pruebas con el fin de identificar la mayor cantidad de problemas en las funcionalidades del sistema. Dicha estrategia comprende la ejecución de pruebas unitarias y pruebas de aceptación. Las pruebas unitarias se enfocan en verificar el comportamiento adecuado de las unidades individuales de código, mientras que las pruebas de aceptación están diseñadas para validar si el sistema cumple con los requisitos establecidos por los usuarios finales. La inclusión de ambas categorías de pruebas garantiza una cobertura integral y exhaustiva del software desarrollado, permitiendo detectar y corregir posibles fallos en las funcionalidades implementadas.

III.4.2 Pruebas Unitarias

Las pruebas unitarias representan una etapa fundamental e imprescindible dentro de la metodología XP. Estas pruebas son aplicadas y ejecutadas de manera constante en respuesta a cada modificación realizada en el sistema. Son diseñadas por los programadores con el propósito de verificar la correcta funcionalidad de todo el código desarrollado, con miras a su posterior publicación. Es importante destacar que tanto el sistema como el conjunto de pruebas deben ser almacenados conjuntamente con el código, permitiendo que otros desarrolladores puedan utilizarlos en caso de requerir correcciones, modificaciones o recodificaciones en partes del mismo.

Las pruebas unitarias fueron realizadas al concluir la implementación de cada una de las funcionalidades de las HU. Las mismas se aplicaron en dos iteraciones, mediante el uso del módulo ‘pytest’ que ofrece el lenguaje Python, reduciendo así el tiempo y el esfuerzo en esta fase.

Luego de realizada la primera iteración se encontró con el siguiente resultado:

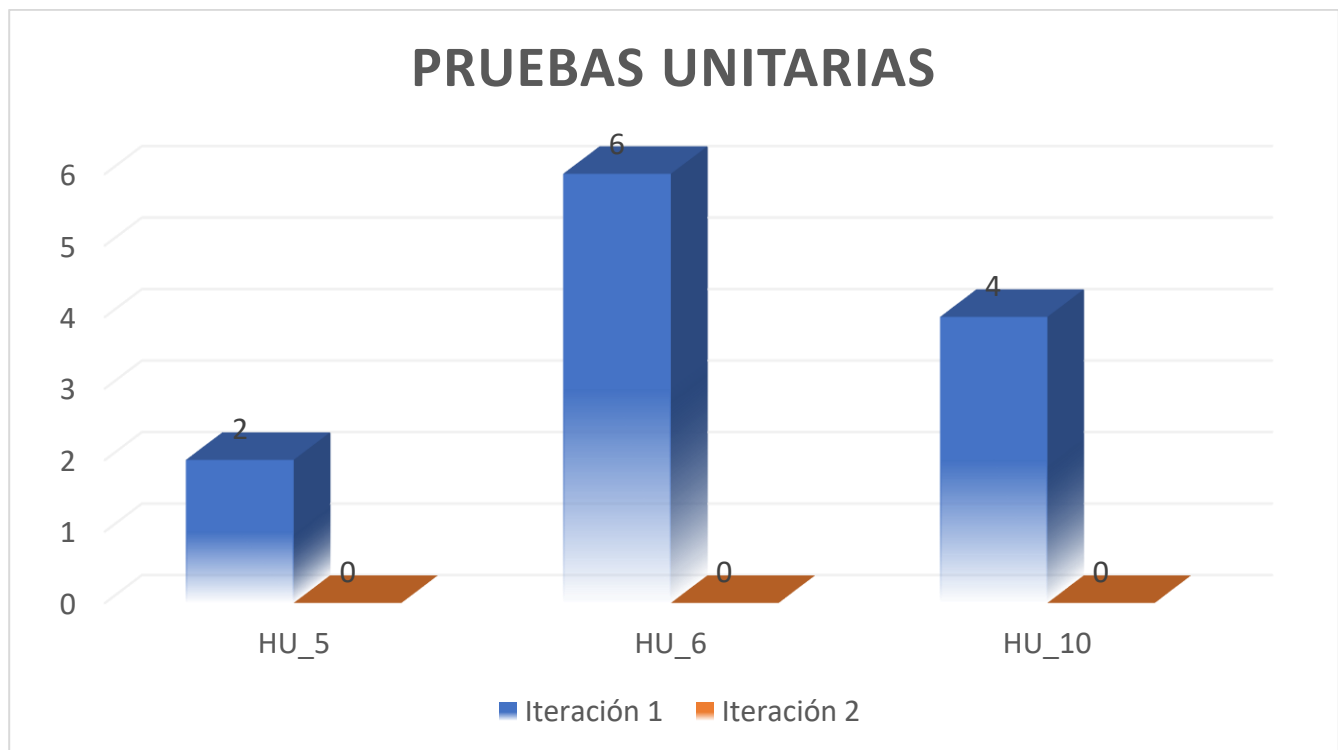


Figura 8 Pruebas Unitarias

III.4.3 Pruebas de Aceptación

Las pruebas de aceptación desempeñan un papel crucial en la metodología XP, siendo creadas en base a las HU en cada ciclo de iteración del desarrollo. Es responsabilidad del cliente especificar uno o varios escenarios que permitan comprobar la correcta implementación de una HU. Estas pruebas de aceptación se consideran como "pruebas de caja negra" ("Black box system tests"), donde el cliente asume la responsabilidad de verificar la precisión de los resultados obtenidos. En caso de que se produzcan fallas en varias pruebas, el cliente debe indicar el orden de prioridad para su resolución. Es importante destacar que una HU no puede considerarse completa hasta que supere exitosamente todas las pruebas de aceptación. Dado que la responsabilidad es compartida por todo el equipo, es recomendable que los resultados de las pruebas de aceptación se publiquen, de manera que todos los miembros estén informados al respecto.

Para llevar a cabo las pruebas de aceptación de usuarios, se utiliza una herramienta manual basada en casos de prueba diseñados por los programadores, con el objetivo de guiar a los clientes en la realización de pruebas efectivas que validen la funcionalidad de manera adecuada. Es importante destacar que una HU puede contar con tantas pruebas de aceptación como sea necesario para garantizar su correcto funcionamiento. El objetivo específico de estas pruebas es asegurar que los requisitos han sido cumplidos y que el sistema es aceptable en su conjunto.

Tabla 41 Caso de Prueba de aceptación 1

Código: HU3_P1	Historia de Usuario: 3
Nombre: Insertar Estudiante	
Descripción: Prueba para la funcionalidad de insertar un estudiante en el sistema.	
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como administrador.	
Pasos de ejecución: ➤ El usuario selecciona de la barra de navegación la opción '<i>Gestionar Usuarios</i>' ➤ El usuario selecciona del submenú la opción '<i>Gestionar Estudiantes</i>' ➤ El usuario presiona el botón '<i>Añadir Estudiante</i>' ➤ El usuario llena todos los campos requeridos con información válida. ➤ El usuario presiona el botón '<i>Guardar</i>'	
Resultados esperados: Se insertará un estudiante en la base de datos y se mostrará un mensaje que indica que se ha insertado satisfactoriamente.	
Evaluación de la prueba: Satisfactoria	

Tabla 42 Caso de Prueba de aceptación 2

Código: HU3_P2	Historia de Usuario: 3
Nombre: Modificar Estudiante	
Descripción: Prueba para la funcionalidad de modificar un estudiante en el sistema.	
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como administrador. ➤ El estudiante debe existir en la base de datos.	
Pasos de ejecución: ➤ El usuario selecciona de la barra de navegación la opción '<i>Gestionar Usuarios</i>' ➤ El usuario selecciona del submenú la opción '<i>Gestionar Estudiantes</i>' ➤ El usuario presiona el botón '<i>Editar</i>' ➤ El usuario llena todos los campos requeridos con información válida. ➤ El usuario presiona el botón '<i>Guardar</i>'	
Resultados esperados: Se modificará el estudiante en la base de datos y se mostrará un mensaje que indica que se ha modificado satisfactoriamente.	
Evaluación de la prueba: Satisfactoria	

Tabla 43 Caso de Prueba de aceptación 3

Código: HU4_P1	Historia de Usuario: 4
Nombre: Insertar Profesor	
Descripción: Prueba para la funcionalidad de insertar un profesor en el sistema.	
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como administrador.	
Pasos de ejecución:	

➤ El usuario selecciona de la barra de navegación la opción '<i>Gestionar Usuarios</i>' ➤ El usuario selecciona del submenú la opción '<i>Gestionar Profesores</i>' ➤ El usuario presiona el botón '<i>Añadir Profesor</i>' ➤ El usuario llena todos los campos requeridos con información válida. ➤ El usuario presiona el botón '<i>Guardar</i>'
Resultados esperados: Se insertará el profesor en la base de datos y se mostrará un mensaje que indica que se ha insertado satisfactoriamente.
Evaluación de la prueba: Satisfactoria

Tabla 44 Caso de Prueba de aceptación 4

Código: HU4_P2	Historia de Usuario: 4
Nombre: Modificar Profesor	
Descripción: Prueba para la funcionalidad de modificar un profesor en el sistema.	
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como administrador. ➤ El profesor debe existir en la base de datos.	
Pasos de ejecución: ➤ El usuario selecciona de la barra de navegación la opción '<i>Gestionar Usuarios</i>' ➤ El usuario selecciona del submenú la opción '<i>Gestionar Estudiantes</i>' ➤ El usuario presiona el botón '<i>Editar</i>' ➤ El usuario llena todos los campos requeridos con información válida. ➤ El usuario presiona el botón '<i>Guardar</i>'	

Resultados esperados: Se modificará el profesor en la base de datos y se mostrará un mensaje que indica que se ha modificado satisfactoriamente.
Evaluación de la prueba: Satisfactoria

Tabla 45 Caso de Prueba de aceptación 5

Código: HU6_P1	Historia de Usuario: 6
Nombre: Insertar Tesis	
Descripción: Prueba para la funcionalidad de insertar una tesis en el sistema.	
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como administrador.	
Pasos de ejecución: ➤ El usuario selecciona de la barra de navegación la opción '<i>Gestionar Tesis</i>' ➤ El usuario presiona el botón '<i>Añadir Tesis</i>' ➤ El usuario llena todos los campos requeridos con información válida. ➤ El usuario presiona el botón '<i>Guardar</i>'	
Resultados esperados: Se insertará la tesis en la base de datos y se mostrará un mensaje que indica que se ha insertado satisfactoriamente.	
Evaluación de la prueba: Satisfactoria	

Tabla 46 Caso de Prueba de aceptación 6

Código: HU6_P2	Historia de Usuario: 6
Nombre: Modificar Tesis	
Descripción: Prueba para la funcionalidad de modificar una tesis en el sistema.	
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como administrador. ➤ La tesis debe existir en la base de datos.	

Pasos de ejecución: ➤ El usuario selecciona de la barra de navegación la opción '<i>Gestionar Tesis</i>' ➤ El usuario presiona el botón '<i>Editar</i>' ➤ El usuario llena todos los campos requeridos con información válida. ➤ El usuario presiona el botón '<i>Guardar</i>'
Resultados esperados: Se modificará la información referente a la tesis en la base de datos y se mostrará un mensaje que indica que se ha modificado satisfactoriamente.
Evaluación de la prueba: Satisfactoria

Tabla 47 Caso de Prueba de aceptación 7

Código: HU7_P1	Historia de Usuario: 7
Nombre: Insertar Primer Corte	
Descripción: Prueba para la funcionalidad de insertar un primer corte de una tesis en el sistema.	
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como profesor administrador.	
Pasos de ejecución:	
Resultados esperados: Se insertará un primer corte relacionado a una tesis en la base de datos y se mostrará un mensaje que indica que se ha insertado satisfactoriamente.	
Evaluación de la prueba: Satisfactoria	

Tabla 48 Caso de Prueba de aceptación 8

Código: HU7_P2	Historia de Usuario: 7

Nombre: Modificar Primer Corte
Descripción: Prueba para la funcionalidad de modificar un primer corte de una tesis en el sistema.
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como profesor o administrador. ➤ El primer corte de tesis debe existir en la base de datos.
Pasos de ejecución:
Resultados esperados: Se modificará el primer corte relacionado a una tesis en la base de datos y se mostrará un mensaje que indica que se ha modificado satisfactoriamente.
Evaluación de la prueba: Satisfactoria

Tabla 49 Caso de Prueba de aceptación 9

Código: HU9_P1	Historia de Usuario: 9
Nombre: Insertar Pre-defensa	
Descripción: Prueba para la funcionalidad de insertar una pre-defensa de una tesis en el sistema.	
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como profesor administrador.	
Pasos de ejecución:	
Resultados esperados: Se insertará una pre-defensa relacionada a una tesis en la base de datos y se mostrará un mensaje que indica que se ha insertado satisfactoriamente.	
Evaluación de la prueba: Satisfactoria	

Tabla 50 Caso de Prueba de aceptación 10

Código: HU9_P2	Historia de Usuario: 9

Nombre: Modificar Pre-defensa
Descripción: Prueba para la funcionalidad de modificar una pre-defensa de una tesis en el sistema.
Condiciones de ejecución: ➤ El usuario debe estar previamente registrado como profesor administrador. ➤ La pre-defensa debe existir en la base de datos.
Pasos de ejecución:
Resultados esperados: Se modificará una pre-defensa relacionada a una tesis en la base de datos y se mostrará un mensaje que indica que se ha modificado satisfactoriamente.
Evaluación de la prueba: Satisfactoria

III.4.4 Validación del Sistema

Las deficiencias mencionadas en la introducción de este estudio tienen un impacto directo en la gestión de la información de los ejercicios de culminación de estudios en la Facultad 2. En este sentido, se presenta la solución informática propuesta como una forma de abordar y resolver estos problemas:

Tabla 51 Validación del sistema.

Indicadores	Antes	Después
Disponibilidad	No existía	24h
Integridad	La información no era exacta y tenía probabilidades altas de alteración o pérdida.	La información es válida y segura.

Fuente: elaboración propia.

Tras llevar a cabo la evaluación de estos indicadores, se evidencia la efectividad y capacidad de la solución informática propuesta para abordar y resolver la problemática descrita en la introducción de la presente investigación.

Conclusiones del capítulo

El capítulo III se centró en la validación de la solución propuesta, a través de la implementación y las pruebas. Durante la implementación, se establecieron directrices para el código y se crearon tareas de ingeniería para cada Historia de Usuario. La colaboración del cliente fue fundamental en este proceso. Se adoptaron estándares de codificación para mejorar la comprensión y el mantenimiento del código. Las pruebas unitarias y las pruebas de aceptación se llevaron a cabo para verificar el funcionamiento correcto del sistema. Las pruebas unitarias demostraron que el código funcionaba adecuadamente. Las pruebas de aceptación, basadas en casos de prueba diseñados por los programadores, validaron la implementación exitosa de las Historias de Usuario. La solución informática propuesta cumple con todos los requisitos para mejorar la disponibilidad y la integridad de la información de los ejercicios de culminación de estudios en la Facultad 2.

CONCLUSIONES FINALES

Luego de exponer el problema que se desea resolver, los diferentes conceptos asociados a la investigación, los detalles de los elementos que conforman la propuesta de solución, implementación y pruebas correspondientes, se puede expresar que los objetivos han sido cumplidos con el mayor rigor.

1. Se logró establecer los fundamentos y referentes teórico-metodológicos para el desarrollo de un sistema informático para la gestión de la información de los ejercicios de culminación de estudios en la Facultad 2.
2. Se diagnosticó el estado actual del proceso de la gestión de la información de los ejercicios de culminación de estudios en la Facultad 2 en cuanto a la disponibilidad e integridad de los datos.
3. La selección de XP como metodología de desarrollo de software permitió guiar y documentar el ciclo de vida, permitiendo que se cumpliera con los estándares y estilos definidos por el cliente.
4. La utilización de las tecnologías y herramientas definidas permitió que la solución informática desarrollada gestione el proceso de gestión de la información de los ejercicios de culminación de estudios en la Facultad 2 contribuyendo a la disponibilidad e integridad de los datos.
5. Las pruebas realizadas permitieron verificar que el sistema para la gestión de la información de los ejercicios de culminación de estudios en la Facultad 2 posee un correcto funcionamiento.

RECOMENDACIONES

Como resultado del desarrollo de la aplicación aquí propuesta y la necesidad de una primera versión capaz de resolver el problema inicial, surgen elementos que pudieran formar parte de este, dándole mayores prestaciones y facilidades de explotación. A continuación, se enumeran algunos de los más importantes:

1. Añadir a los usuarios la funcionalidad de contar con chats privados directamente con el tutor, cotutor y los profesores asignados a su tribunal, permitiendo así una mejor comunicación ante cambios inesperados.
2. Dar mantenimiento sistemático al sistema permitiendo añadir nuevas mejoras en un futuro.
3. Ampliar más allá de la facultad, permitiendo sus servicios a todas las facultades de la comunidad universitaria.

REFERENCIAS BIBLIOGRÁFICAS

- Ahmad, T. (2019). *Model-based testing using UML activity diagrams: A systematic mapping study*. Computer Science Review, 33.
- Alawar, M., & Abu-Naser, S. (2017). *CSS-Tutor: An intelligent tutoring system for CSS and HTML*. Retrieved from dstore.alazhar.edu.ps
- Aranea, L. E. (2021). *APLICACIÓN INFORMÁTICA PARA LOS PROCESOS DE GESTIÓN DE LAS TUTORÍAS ACADÉMICAS DE LA CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN DE LA UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ*. Retrieved from repositorio.unesum.edu.ec
- Arias, M. Á. (2017). *Aprende Programacion Web con PHP y MySQL: 2ª Edicion*. IT campus Academy.
- Asencio, P. L. (2016). *Desarrollo e implementación de un sistema web para generar valor en una pyme aplicando una metodología ágil*. Manufibras Perez SRL [online] [visitado 2021-11-10]. Retrieved from cybertesis.unmsm.edu.pe/handle/cybertesis/4668/Castillo_ap.pdf
- Ayo, F. P. (2018). *SISTEMA DE LEVANTAMIENTO DE ACTIVOS FIJOS EN ANDROID PARA LA EMPRESA CAYMAN SYSTEMS*. Quito: Bachelor's Thesis.
- Canals, C. S. (2015). *Programa informático para el seguimiento y control de camiones frigoríficos*. Universitat Politècnica de Catalunya.
- Castillo, A. A. (2017). *Curso de Programación Web: JavaScript, Ajax y jQuery*. Campus Academy IT.
- Dueñas, Y. M. (2021). *Sistema informático para el control docente educativo en la Universidad de Ciego de ávila, Cuba*.
- Fossati, M. (2018). *Introducción a PHP y HTML*. Retrieved from books.google.com.cu
- Garinzo, D. M. (2015). *Desarrollo e implementación de un sistema para el registro y control de tutorías académicas, basado en una arquitectura de tres capas, para la carrera de Ingeniería en Sistemas de la Universidad Politécnica Salesiana Sede Quito*. Retrieved from dspace.ups.edu.ec
- Garzón, C. Á. (2019). *Modelo Vista Controlador*. Ingeniería de Software.
- Lazo, A. T., & Botero, J. G. (2016). *Especificación de requisitos de software: una mirada desde la revisión teórica de antecedentes*. Entre Ciencia e Ingeniería, 10.

López, D. V. (2016). *Aplicación para gestión de procesos de desarrollo de software basados en la metodología ágil XP Extreme Programming para Software de la Sierra SA*. Retrieved from repositorio.pucesa.edu.ec

López, E. G. (2013). *BPMN: Estándar para modelar procesos de negocio*. INNOTECH Gestión.

Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Retrieved from repositorio.uca.edu.ar

Marani, F. (2019). *Adding CSS/JavaScript to Our Django Project*. Practical Django 2 y Channels 2. Apress, Berkeley, CA.

Márquez, Y. M. (2018). *Manual DOcente Metodológico para el Vicedecanato de Formación*.

Myers. (2004). *The Art of software testing*.

Paradigm, V. (2015). *Sitio Web Oficial Visual Paradigm*.

Peña, A. D. (2016). *Arquitectura de software para el sistema de visualización médica Vismedic*. Revista Cubana de Informática Médica.

Pressman, R. S. (2011). *Ingeniería del software . Un Enfoque*.

Sánchez, J. M. (2018). *Informática básica Concepto de informática, sistema informático, el computador como herramienta fundamental de la informática. Conceptos básicos y terminología de un computador, el computador y la educación, la informática y la educación*. Retrieved from repositorio.une.edu.pe

Sandó, P. V. (2021). *Diversificación de las formas de culminación de estudios universitarios de pregrado en países de Iberoamérica*. Transformación.

Stauffer, M. (2019). *Laravel: Up & running: A framework for building modern php apps*. O'Reilly Media.

Tuquerres, J. E., & Pullupaxi Cando, D. M. (2020). *Sistema informático de gestión de información para los procesos de producción e inventarios en la florícola ROSADEX CIA. LTDA*. Ecuador: Latacunga: Universidad Técnica de Cotopaxi.

Venkat, N. (2017). *RDBMS PostgreSQL-Create database using PgAdmin*. Retrieved from doercol.org

Victor, Y. H., & Bouza, R. I. (2020). *MÓDULO INFORMÁTICO PARA LA GESTIÓN DE PROCESOS DE ACTAS Y CULMINACIÓN DE ESTUDIO PARA EL SECTOR DE LA EDUCACIÓN SUPERIOR/COMPUTER MODULE FOR THE PROCESS MANAGEMENT OF*

MINUTES AND CULMINATION OF STUDY FOR THE SECTOR OF HIGHER EDUCATION.
Universidad&Ciencia.