



Facultad 4

Sistema de Gestión de Información de Doctorandos

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

Autor: Yilber Serrano Figueredo

Tutores: Ing. Luis Manuel Valera Pérez

Dr. C. Yamilis Fernández Pérez

La Habana, 22 de noviembre de 2023

“Año 65 de la Revolución”

**"La única forma de
hacer un gran trabajo,
es amar lo que haces"**

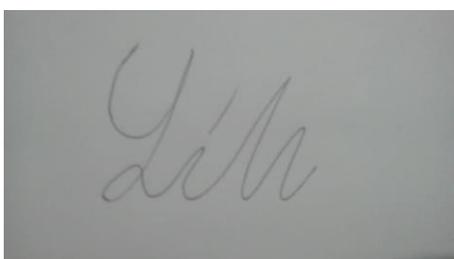
—Steve Jobs



DECLARACIÓN DE AUTORÍA

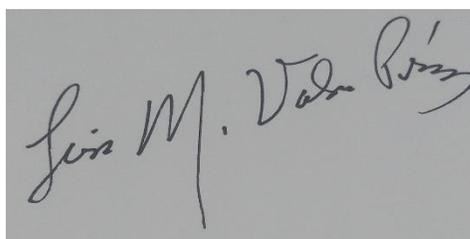
El autor del trabajo de diploma con título “**Sistema de Gestión de Información de Doctorandos**” concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los 22 días del mes de noviembre del año 2023.

Yilber Serrano Figueredo



Firma del Autor

Ing. Luis Manuel Valera Pérez



Firma del Tutor

Dr. C. Yamilis Fernández Pérez



Firma del Tutor

DATOS DE CONTACTO

Dr. C. Yamilis Fernández Pérez, Es graduada de Doctora en Ciencias Técnicas, Máster en Informática Aplicada e Ingeniera en SAD. Labora en la UCI desempeñándose como profesora y jefa del grupo de Grados científicos. Desde el año 1992 trabaja en la Educación Superior como profesora y es fundadora de la UCI, ha desempeñado varias responsabilidades como jefa de departamento de Ingeniería y Gestión de Software, Directora de Formación de Postgrado, jefa de disciplina y de asignatura. Investiga en Toma de Decisiones y Calidad de Software. Correo electrónico: yamilisf@uci.cu.

Ing. Luis Manuel Valera Pérez, Graduado en el curso 2014-2015 como Ingeniero en Ciencias Informáticas. Título obtenido en Cuba. Ha presentado ponencias en eventos nacionales y ha publicado varios artículos en revistas indexadas de bases de datos. Actualmente profesor de Sistemas de Bases de Datos de la Facultad 4. Tiene la categoría docente de Profesor Asistente. Correo electrónico: luismlvalera@uci.cu. Teléfono de contacto: 7 837-2219

DEDICATORIA

A mi mamá que ha sacrificado todo por mí y me ha guiado siempre por el mejor camino, que siempre ha estado ahí cuando la necesito, y a la que amo por encima de todo en este mundo.

A mi hermana, que después de mi mamá es la mujer que más quiero, que me ha dado su amor y su apoyo incondicional.

A mi papá que ha estado ahí siempre para mí, por ayudarme y darme todo su apoyo siempre.

A mis amigos y compañeros Osmany, José Carlos, Jean, Josué, Pepe, Felipe, Elier que han estado durante este viaje a mi lado apoyándonos unos a otro.

A mis tutores Yamilis y Luis Manuel por apoyarme y ayudarme siempre, por guiarme y estar siempre pendiente de mí.

RESUMEN

La presente investigación tiene como objetivo desarrollar un sistema web que mejore el proceso de gestión de la información de los doctorandos en el proceso de formación doctoral en la Universidad de las Ciencias Informáticas. Este proceso es realizado actualmente a través del software de edición de hojas de cálculo Microsoft Excel, el cual presenta varios problemas a la hora de mantener actualizada y segura la información. El estudio y análisis del estado del arte permitió identificar las funcionalidades y tecnologías que facilitaron la comprensión y diseño de la solución propuesta. El desarrollo estuvo guiado por la metodología XP y se seleccionó como principales tecnologías el marco de trabajo SpringBoot, Java como lenguaje de programación, el entorno integrado de desarrollo Netbeans y como gestor de base de datos se utilizó MySQL. Las pruebas aplicadas permitieron validar el cumplimiento de los requisitos planteados por el cliente.

Palabras claves: doctorandos, gestión, información, sistema de gestión

ABSTRACT

The objective of this research is to develop a web system that improves the information management process of doctoral students in the doctoral training process at the University of Computer Sciences. This process is currently carried out through the spreadsheet editing software Microsoft Excel, which presents several problems when it comes to keeping the information updated and secure. The study and analysis of the state of the art made it possible to identify the functionalities and technologies that facilitated the understanding and design of the proposed solution. The development was guided by the XP methodology and the Spring Boot framework, Java as a programming language, the Netbeans integrated development environment, and MySQL was used as the database manager were selected as the main technologies. The tests applied allowed us to validate compliance with the requirements posed by the client.

Keywords: management, doctoral students, management system, information

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO.....	5
1.1 Conceptos asociados al dominio del problema.....	5
1.2 Análisis de sistemas homólogos.....	6
1.3 Herramientas y tecnologías.....	9
1.3.1 Lenguaje del lado del servidor.....	10
1.3.2 Lenguaje del lado del cliente.....	13
1.3.3 Herramienta CASE Visual Paradigm.....	14
1.3.4 Sistema gestor de base de datos MySQL.....	15
1.3.5 Entorno de desarrollo integrado.....	16
1.3.6 Servidor de aplicaciones web Apache.....	17
1.3.7 Marco de trabajo para el desarrollo de la solución informática.....	18
1.4 Metodología de desarrollo de software utilizada.....	18
Conclusiones del capítulo.....	21
CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO.....	22
2.1 Descripción de la propuesta de solución.....	22
2.2 Historias de Usuario.....	22
2.3 Requisitos no funcionales.....	24
2.4 Tarjetas CRC.....	24
2.5 Plan de lanzamiento.....	26
2.6 Patrón de arquitectura.....	27
2.7 Patrones de Diseño.....	28
2.8 Modelo de Datos.....	29
Conclusiones del capítulo.....	30
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	32
3.1 Tareas de Ingeniería.....	32
3.2 Pruebas.....	34
3.2.1 Pruebas Unitarias.....	34
3.2.2 Pruebas de Aceptación.....	37
3.3 Resultado de las pruebas.....	39
Conclusiones del capítulo.....	40
CONCLUSIONES FINALES.....	41
RECOMENDACIONES.....	42
REFERENCIAS BIBLIOGRÁFICAS.....	43
ANEXOS.....	48

ÍNDICE DE TABLAS

Tabla 1: Características de los sistemas homólogos analizados.....	8
Tabla 2: HU 1.....	22
Tabla 3: HU 2.....	23
Tabla 4: HU 3.....	23
Tabla 5: Tarjeta CRC “doctorando”	25
Tabla 6: Tarjeta CRC “doctor”	25
Tabla 7: Tarjeta CRC “tutor”	25
Tabla 8: Tarjeta CRC “publicacion”	26
Tabla 9: Plan de lanzamiento	26
Tabla 10: Tarea de ingeniería 1	32
Tabla 11: Tarea de ingeniería 2	32
Tabla 12: Tarea de ingeniería 3	32
Tabla 13: Tarea de ingeniería 4	33
Tabla 14: Tarea de ingeniería 5	33
Tabla 15: Tarea de ingeniería 6	33
Tabla 16: Tarea de ingeniería 7	34
Tabla 17: Caso de Prueba de Aceptación HU1_P1	37
Tabla 18: Caso de Prueba de Aceptación HU1_P2	38
Tabla 19: Caso de Prueba de Aceptación HU1_P3	38
Tabla 20: Caso de Prueba de Aceptación HU1_P4	39
Tabla 21: HU 4.....	48
Tabla 22: HU 5.....	48
Tabla 23: HU 6.....	49
Tabla 24: HU 7.....	49
Tabla 25: HU 8.....	50
Tabla 26: HU 9.....	50
Tabla 27: HU 10.....	50
Tabla 28: HU 11.....	51
Tabla 29: HU 12.....	51
Tabla 30: HU 13.....	51
Tabla 31: HU 14.....	52
Tabla 32: HU 15.....	52
Tabla 33: HU 16.....	53
Tabla 34: HU 17.....	53
Tabla 35: Tarjeta CRC “graduados”	53
Tabla 36: Tarjeta CRC “programa”	54
Tabla 37: Tarjeta CRC “usuario”	54
Tabla 38: Tarjeta CRC “solicitante”	54
Tabla 39: Tarjeta CRC “areadelconocimiento”	54
Tabla 40: Tarjeta CRC “area”	55
Tabla 41: Tarjeta CRC “centro”	55
Tabla 42: Tarjeta CRC “lineainvest”	55
Tabla 43: Tarjeta CRC “sectorest”.....	56
Tabla 44: Tarjeta CRC “persona”	56
Tabla 45: Tarjeta CRC “indexado”.....	56
Tabla 46: Tarjeta CRC “pais”.....	57
Tabla 47: Tarjeta CRC “seguridad”	57
Tabla 48: Tarjeta CRC “estadísticas”	57
Tabla 49: Tarea de ingeniería 8	57
Tabla 50: Tarea de ingeniería 9	58
Tabla 51: Tarea de ingeniería 10	58

Tabla 52: Tarea de ingeniería 11	58
Tabla 53: Tarea de ingeniería 12	58
Tabla 54: Tarea de ingeniería 13	59
Tabla 55: Tarea de ingeniería 14	59
Tabla 56: Tarea de ingeniería 15	59
Tabla 57: Tarea de ingeniería 16	60
Tabla 58: Tarea de ingeniería 17	60
Tabla 59: Tarea de ingeniería 18	60
Tabla 60: Caso de Prueba de Aceptación HU2_P1	65
Tabla 61: Caso de Prueba de Aceptación HU2_P2	66
Tabla 62: Caso de Prueba de Aceptación HU2_P3	66
Tabla 63: Caso de Prueba de Aceptación HU2_P4	67
Tabla 64: Caso de Prueba de Aceptación HU3_P1	67
Tabla 65: Caso de Prueba de Aceptación HU3_P2	68
Tabla 66: Caso de Prueba de Aceptación HU3_P3	68
Tabla 67: Caso de Prueba de Aceptación HU4_P1	68
Tabla 68: Caso de Prueba de Aceptación HU4_P2	69
Tabla 69: Caso de Prueba de Aceptación HU4_P3	69
Tabla 70: Caso de Prueba de Aceptación HU4_P4	70
Tabla 71: Caso de Prueba de Aceptación HU5_P1	70
Tabla 72: Caso de Prueba de Aceptación HU5_P2	71
Tabla 73: Caso de Prueba de Aceptación HU5_P3	71
Tabla 74: Caso de Prueba de Aceptación HU5_P4	72
Tabla 75: Caso de Prueba de Aceptación HU6_P1	72
Tabla 76: Caso de Prueba de Aceptación HU6_P2	73
Tabla 77: Caso de Prueba de Aceptación HU6_P3	73
Tabla 78: Caso de Prueba de Aceptación HU6_P4	74
Tabla 79: Caso de Prueba de Aceptación HU7_P1	74
Tabla 80: Caso de Prueba de Aceptación HU8_P1	74
Tabla 81: Caso de Prueba de Aceptación HU8_P2	75
Tabla 82: Caso de Prueba de Aceptación HU8_P3	75
Tabla 83: Caso de Prueba de Aceptación HU8_P4	76
Tabla 84: Caso de Prueba de Aceptación HU9_P1	76
Tabla 85: Caso de Prueba de Aceptación HU9_P2	77
Tabla 86: Caso de Prueba de Aceptación HU9_P3	77
Tabla 87: Caso de Prueba de Aceptación HU9_P4	78
Tabla 88: Caso de Prueba de Aceptación HU10_P1	78
Tabla 89: Caso de Prueba de Aceptación HU10_P2	78
Tabla 90: Caso de Prueba de Aceptación HU10_P3	79
Tabla 91: Caso de Prueba de Aceptación HU10_P4	79
Tabla 92: Caso de Prueba de Aceptación HU11_P1	80
Tabla 93: Caso de Prueba de Aceptación HU11_P2	80
Tabla 94: Caso de Prueba de Aceptación HU11_P3	81
Tabla 95: Caso de Prueba de Aceptación HU11_P4	81
Tabla 96: Caso de Prueba de Aceptación HU12_P1	82
Tabla 97: Caso de Prueba de Aceptación HU12_P2	82
Tabla 98: Caso de Prueba de Aceptación HU12_P3	83
Tabla 99: Caso de Prueba de Aceptación HU12_P4	83
Tabla 100: Caso de Prueba de Aceptación HU13_P1	84
Tabla 101: Caso de Prueba de Aceptación HU13_P2	84
Tabla 102: Caso de Prueba de Aceptación HU13_P3	85
Tabla 103: Caso de Prueba de Aceptación HU13_P4	85
Tabla 104: Caso de Prueba de Aceptación HU14_P1	85
Tabla 105: Caso de Prueba de Aceptación HU14_P2	86

Tabla 106: Caso de Prueba de Aceptación HU14_P3	86
Tabla 107: Caso de Prueba de Aceptación HU14_P4	87
Tabla 108: Caso de Prueba de Aceptación HU15_P1	87
Tabla 109: Caso de Prueba de Aceptación HU16_P1	88
Tabla 110: Caso de Prueba de Aceptación HU17_P1	88

ÍNDICE DE FIGURAS

Ilustración 1: Representación de la arquitectura.	28
Ilustración 2: Modelo de datos.....	30
Ilustración 3: Resultados de la prueba unitaria a doctorandoService	37
Ilustración 4: Pruebas realizadas	40
Ilustración 5: Resultados de la prueba unitaria a areaService.	61
Ilustración 6: Resultados de la prueba unitaria a areadeconocimientoService.....	61
Ilustración 7: Resultados de la prueba unitaria a centroService.	61
Ilustración 8: Resultados de la prueba unitaria a dochaslineService.	61
Ilustración 9: Resultados de la prueba unitaria a dcochaspubService.....	62
Ilustración 10: Resultados de la prueba unitaria a doctorService.	62
Ilustración 11: Resultados de la prueba unitaria a graduadoService.	62
Ilustración 12: Resultados de la prueba unitaria a indexadoService.	62
Ilustración 13: Resultados de la prueba unitaria a indhaspubService.	63
Ilustración 14: Resultados de la prueba unitaria a lineainvestService.	63
Ilustración 15: Resultados de la prueba unitaria a paisService.....	63
Ilustración 16: Resultados de la prueba unitaria a personaService.	63
Ilustración 17: Resultados de la prueba unitaria a programaService.....	64
Ilustración 18: Resultados de la prueba unitaria a publicacionService.	64
Ilustración 19: Resultados de la prueba unitaria a sectorestService.	64
Ilustración 20: Resultados de la prueba unitaria a solicitanteService.	64
Ilustración 21: Resultados de la prueba unitaria a tutorService.	65
Ilustración 22: Resultados de la prueba unitaria a usuarioService.	65
Ilustración 23: Resultados de la prueba unitaria a estadisticaService.	65

INTRODUCCIÓN

La universidad cubana es una institución de gran importancia en el país, ya que tiene como misión principal la formación de futuros profesionales altamente calificados en la rama que cursen. A su vez ofrece posibilidades de superación a aquellos profesionales que han obtenido una licenciatura o grado universitario, a través de los programas de posgrado o formación posgraduada.

La educación de posgrado es una de las direcciones principales de trabajo de la educación superior en Cuba, y el nivel más alto del sistema de educación superior, dirigido a promover la educación permanente de los graduados universitarios. En la educación de posgrado concurren uno o más procesos formativos y de desarrollo, no solo de enseñanza aprendizaje, sino también de investigación, innovación, creación artística y otros, articulados armónicamente en una propuesta docente-educativa pertinente a este nivel ¹.

En la actualidad, los estudios de posgrado son importantes para el desarrollo y fortalecimiento de las capacidades en el ámbito de la ciencia, la tecnología y la innovación, y su aplicación en las necesidades de nuestro país. Esto se fundamenta, de un lado, en la evidencia histórica de la centralidad de la educación, la investigación y el aprendizaje colectivo en los procesos de desarrollo; y de otro, en la necesidad de la educación a lo largo de la vida, apoyada en la autogestión del aprendizaje y la socialización en la construcción del conocimiento ².

A tono con esta realidad, la educación de posgrado se constituye en un elemento imprescindible para el desarrollo social y profesional de individuos y de grupos sociales, que necesitan de una preparación continua para su inclusión y permanencia exitosa en la vida laboral. Dicha preparación se favorece en parte por la rápida inserción de las Tecnologías de la Información y la Comunicación (TIC) en los actividades y programas de posgrados en general ³.

La educación de posgrado se estructura en superación profesional, formación académica de posgrado y doctorado, de estas se derivan varias formas organizativas que se diferencian por su objetivo y diseño curricular. La primera, la superación profesional propicia cursos, entrenamientos, talleres y diplomados; mientras que la formación se enfoca en maestrías y especialidades, destacando como tercer nivel, formación doctoral y posdoctoral. Todas estas

actividades se pueden realizar en diferentes modalidades como: a tiempo completo, a tiempo parcial o con diferentes niveles de presencialidad, es decir, presencial, semipresencial o a distancia ⁴.

Con la visión de futuro del comandante en jefe Fidel Castro Ruz, surge la Universidad de Ciencias Informáticas (UCI) en el año 2002. Fidel, como estrategia fundacional, recomendó que la universidad fuese concebida como un centro de nuevo tipo, de alcance nacional, atípica y con tareas concretas en el proyecto de informatización de la sociedad cubana, con énfasis en la producción de software. Un centro que formaría a sus educandos desde una destacada actividad política, cultural y deportiva, objetivos que se vienen cumpliendo mediante la interrelación de la formación-investigación-producción-extensión universitaria como un sistema único y coherente ⁵.

Una de las principales líneas de trabajo de la UCI para alcanzar la excelencia académica ha sido el fortalecimiento de su sistema de posgrado, especialmente la formación de doctores en ciencias, con el objetivo de garantizar una sólida actividad científica y una formación doctoral de calidad. Sin embargo, a pesar de estos esfuerzos, aún existen desafíos y problemáticas que limitan el crecimiento y la calidad de la formación doctoral en la UCI ⁶.

Aunque se ha observado un aumento en el número de doctores en los últimos años, aún es insuficiente para fortalecer la actividad científica y la formación doctoral en la universidad. Además, se ha destacado que muchos doctores jóvenes han causado baja de la universidad y ahí una baja proporción de doctores en ciencias en la UCI en comparación con el total del claustro, lo que limita la capacidad de la institución para desarrollar la formación doctoral y la investigación. Además, la mayoría de los doctores son mayores de 50 años, lo que indica que se necesitan más esfuerzos para formar a nuevos doctores y reemplazar a los que se jubilen. De ahí la importancia del proceso de seguimiento y control de la información de la formación doctoral.

Uno de los déficits actuales en este proceso es la gestión de la información de los doctorandos, quienes son los que están realizando estudios avanzados en su campo de interés y trabajando en una investigación original que les permitirá obtener el título de Doctor. La importancia de los doctorandos radica en que son los futuros líderes académicos y profesionales en sus respectivos campos. A través de su investigación, contribuyen al avance del conocimiento y la

innovación en su campo de estudio. Además, a menudo trabajan en estrecha colaboración con sus profesores y otros investigadores, lo que puede llevar a importantes colaboraciones y descubrimientos en el futuro.

El manejo actual de la información respecto a los doctorandos se maneja a través de una hoja de cálculo Excel, que, aunque Excel es una herramienta útil y ampliamente utilizada para la gestión de la información, también tiene algunas desventajas. Una de las principales desventajas es la falta de capacidad para manejar grandes conjuntos de datos y la limitación en la capacidad de análisis de datos complejos. Además, Excel no es una herramienta colaborativa, por lo que no es adecuado para la colaboración en tiempo real entre implicados en el control de la información. También puede haber problemas de seguridad de datos al utilizar Excel, ya que la información puede ser fácilmente perdida o compartida sin autorización debido a la falta de controles de acceso y la dificultad para mantener una copia de seguridad confiable ⁷.

Como consecuencia de lo anteriormente expuesto, surge el siguiente **problema de investigación**: ¿Cómo mejorar el control de la información de los doctorandos en el proceso de formación doctoral?

Como **objeto de estudio** se tiene: El proceso de gestión de la información de los doctorandos en el proceso de formación doctoral.

El **campo de acción** planteado es: Informatización del proceso de gestión de la información de los doctorandos en el proceso de formación doctoral de la UCI.

Para resolver el problema planteado se traza como **objetivo general**: Desarrollar un sistema que mejore el proceso de gestión de la información de los doctorandos en el proceso de formación doctoral de la UCI.

Los **objetivos específicos** de este trabajo son:

1. Desarrollar el marco teórico referencial de la investigación.
2. Definir la arquitectura, herramientas, tecnologías y metodología a emplear para el desarrollo del sistema.
3. Desarrollar el sistema de gestión de la información de doctorandos.
4. Validar las funcionalidades del sistema de gestión de la información de doctorandos.

Como **métodos de investigación científica** se utilizaron:

Métodos teóricos:

- Analítico-sintético: Analizar los documentos y la información referente a la forma de gestión de la información sobre el proceso de doctorados.

Métodos empíricos:

- Entrevista: Se realizan entrevistas a los profesores implicados en el proceso, con el objetivo de precisar el problema a resolver. Se pueden ver las preguntas realizadas en los anexos.

Como **resultado** de la investigación se obtuvo: el sistema para la gestión de la información de los doctorandos en su primera versión.

Estructura capitular:

Capítulo 1: Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio.

En este capítulo se hace un análisis de los principales conceptos relacionados con el objeto de estudio. Además, se realiza un análisis del estado del arte de sistemas de gestión similares, las herramientas, las distintas técnicas, tecnologías y metodologías que serán usadas para el cumplimiento del objetivo general.

Capítulo 2: Diseño de la solución propuesta al problema de investigación. Se especifican las características que tendrá el sistema propuesto, lo que posibilitará un mayor entendimiento de este. Se muestran los artefactos propios de la metodología de desarrollo, que se generan en las distintas fases de la metodología de desarrollo de software seleccionada.

Capítulo 3: Validación de la solución propuesta. En este capítulo se implementan las funcionalidades requeridas por el cliente y se le realizan pruebas de aceptación, con las cuales se verifica su correcto funcionamiento y satisfacción del cliente.

CAPÍTULO I: FUNDAMENTOS Y REFERENTES TEÓRICO-METODOLÓGICOS SOBRE EL OBJETO DE ESTUDIO

En este capítulo se hace un estudio acerca de los sistemas de gestión, abordando definiciones y conceptos importantes utilizados durante la investigación referente al objeto de estudio. Se explicarán también las diferentes tecnologías y herramientas que se seleccionan para desarrollar el sistema.

1.1 Conceptos asociados al dominio del problema

Para obtener mayor conocimiento del dominio del problema, se realizó un estudio donde se analizaron un conjunto de términos como:

Gestión de la información: La gestión de información es el proceso de organizar, almacenar, eliminar y divulgar la información existente en un contexto determinado, garantizando su integridad, confidencialidad y disponibilidad en el momento que sea necesario ⁸.

Sistemas de gestión: Un sistema de gestión es una herramienta que permite controlar, planificar, organizar y automatizar las tareas administrativas de una organización. Un sistema de gestión analiza los rendimientos y los riesgos de una empresa, con el fin de otorgar un ambiente laboral más eficiente y sostenible ⁹.

Sistema de gestión de la información: Un sistema de gestión de la información es un conjunto de herramientas y tecnologías que se utilizan para recopilar, almacenar, procesar, analizar y compartir información de manera efectiva y eficiente en una organización. Este sistema puede incluir componentes como software, hardware, redes, bases de datos, procedimientos, políticas y recursos humanos especializados ¹⁰.

Sistema de gestión académico: Un sistema de gestión académico es un conjunto de herramientas y tecnologías que se utilizan para apoyar la gestión de procesos académicos en instituciones educativas, como universidades, escuelas, colegios, entre otros. El objetivo principal de un sistema de gestión académico es facilitar la planificación, gestión y seguimiento de actividades académicas, administrativas y de investigación en una institución educativa ¹¹.

La educación de posgrado: Es el nivel más alto del sistema de educación del país, garantiza la superación permanente de los graduados universitarios, pueden concurrir uno o más procesos formativos y de desarrollo, como el proceso de enseñanza-aprendizaje, la especialización, investigación, innovación, creación artística y otros, articulados armónicamente en una propuesta educativa pertinente ¹².

Solicitante: Graduado de nivel superior que documente el cumplimiento de los requisitos establecidos para la matrícula en un programa de doctorado ¹².

Doctorando: Solicitante que, por haber cumplido los requisitos establecidos y tener un tema de doctorado aceptado en una de las líneas de investigación del programa de doctorado, haya sido aprobado por la institución autorizada para la formación de doctores con el fin de formalizar su matrícula en este ¹².

Grado científico de Doctor: El grado científico de Doctor en determinada área del conocimiento se otorga a los graduados de nivel universitario que contribuyen significativamente al desarrollo del conocimiento en su tema de investigación y satisfacen a plenitud los requisitos y las evaluaciones correspondientes de sus programas de doctorados ¹³.

Tutor: La persona directamente responsabilizada con la formación científica del doctorando y con el desarrollo de su trabajo de tesis ¹².

1.2 Análisis de sistemas homólogos

En la actualidad se han desarrollado sistemas para la gestión de la información académica y universitaria, los cuales poseen una amplia gama de funcionalidades que facilitan el trabajo. A continuación, se analizan algunos sistemas de gestión, con el objetivo de identificar sistemas capaces de solucionar la problemática en su totalidad o que permita reutilizar algunas funcionalidades para brindar una solución al problema planteado anteriormente.

Ellucian Student Information System (SIS):

SIS es un sistema de gestión de información estudiantil internacional que se utiliza en muchas universidades y colegios de todo el mundo. Su objetivo principal es facilitar la administración y gestión de información estudiantil, incluyendo información académica, asistencia, horarios, calificaciones y registros estudiantiles.

SIS ofrece una amplia variedad de características para ayudar a las instituciones educativas a gestionar el proceso de admisión, la inscripción, la planificación académica, la evaluación y la asistencia estudiantil. La plataforma también ofrece características para la gestión financiera, como la facturación y la gestión de ayudas financieras. También se presenta una amplia variedad de informes y analítica que permiten a los usuarios obtener información valiosa sobre el desempeño de los estudiantes y las operaciones de la institución educativa. Los usuarios pueden crear informes personalizados para monitorear el progreso de los estudiantes, la eficiencia de los procesos y la satisfacción de los estudiantes. SIS ofrece una amplia variedad de funcionalidades para la gestión de estudiantes y la información académica, lo que incluye funciones específicas para doctorados, como la planificación de cursos avanzados, la gestión de tesis y la supervisión académica ¹⁴.

SIGMA SIS:

SIGMA Gestión Universitaria A.I.E. es una agrupación de interés económico integrada por varias universidades españolas y otras instituciones privadas del ámbito de la enseñanza superior.

Esta empresa centra su actividad en el desarrollo de aplicaciones de gestión académica para las Universidades. Entre estas aplicaciones se encuentra el Sistema de Gestión Académica denominado SIGMA SIS, que entre otros dispone de los siguientes módulos:

- Módulo que gestiona los estudios oficiales de Grado, así como los estudios de Máster Oficial y Doctorado.
- Módulo para la gestión de la ordenación docente que permite la definición de oferta académica, planes de docencia, definición de grupos de docencia, asignación de profesorado, entre otros.
- Módulo para la gestión de los estudios propios de la Universidad.

SIGMA Gestión Universitaria A.I.E. ofrece también otras aplicaciones para la gestión de la investigación denominado SIGMA CRIS ¹⁵.

Sistema de Gestión de la Nueva Universidad (SIGENU):

SIGENU surge en el año 2004 como una iniciativa del Ministerio de Educación Superior. Es un sistema desarrollado con el fin de ser una herramienta que permita la gestión de toda la infor-

mación académica vinculada con la Educación Superior en Cuba de manera segura e integrada, adaptándose a las particularidades de cada centro de educación superior. Cuenta con un servidor de aplicaciones que es el encargado de actuar como intermediario entre la base de datos y las aplicaciones clientes, permite la inserción y actualización de toda la información que se registra en el sistema.

La herramienta ofrece una variedad de informes que son de uso común para los distintos niveles jerárquicos presentes en el ámbito de la educación superior. Los profesores tienen la capacidad de planificar asignaturas específicas y el sistema permite la gestión de la información relacionada con los procesos administrativos en las diferentes modalidades. La información de cada uno de los estudiantes registrados en el sistema está disponible en todo momento, siempre y cuando el usuario tenga los permisos necesarios para acceder a ella ¹⁶.

Resultado del análisis de las soluciones:

Durante el estudio de los diversos sistemas encontrados que abordan el registro y control de la información de estudiantes de doctorados, se pudo constatar que todos poseen una variedad de funcionalidades que se ajustan a las área y procesos que desarrollan las instituciones o áreas de trabajo que los albergan, ninguna de ellas se ajusta completamente a las necesidades para la formación doctoral de la UCI. Es por eso, que se determina no utilizar ninguno de ellos como propuesta de solución a la presente investigación, pero si se tomaron algunas ideas que pueden ser de utilidad para el desarrollo del software.

A continuación, una tabla de resumen sobre el análisis de las características de los sistemas homólogos y que se tuvieron en cuenta para la realización de la solución:

Tabla 1: Características de los sistemas homólogos analizados

Sistemas de gestión	Aplicación web o de escritorio	Autenticación de usuario	Pago	Lenguajes de programación	Gestión de estadísticas
SIS	Web	Si	Si	Java	Si
SIGMA SIS	Web	Si	Si	Java	Si
SIGENU	Web	Si	No	PHP	Si

Como conclusión de este estudio se propone el desarrollo de una aplicación web, por solucionar algunos de los problemas mencionados en el proceso actual de gestión de la información de doctorandos y las ventajas que se mencionan a continuación:

- **Accesibilidad:** Son accesibles desde cualquier lugar con conexión a Internet y a través de diferentes dispositivos, como computadoras de escritorio, laptops, tablets y teléfonos móviles.
- **Actualizaciones y mantenimiento simplificados:** Las actualizaciones y el mantenimiento se realizan en el servidor, lo que significa que los usuarios no necesitan instalar actualizaciones manualmente en sus dispositivos. Esto facilita la distribución de nuevas versiones de la aplicación y garantiza que todos los usuarios tengan acceso a la última versión.
- **Colaboración y compartición de datos:** Las aplicaciones web permiten una mayor colaboración entre usuarios, ya que múltiples personas pueden acceder y trabajar en la misma aplicación simultáneamente. Además, compartir datos y colaborar en tiempo real es más fácil en una aplicación web, ya que los datos se almacenan en un servidor centralizado y se pueden acceder desde cualquier dispositivo.
- **Costos de implementación y mantenimiento:** Las aplicaciones web generalmente tienen costos de implementación y mantenimiento más bajos en comparación con las aplicaciones de escritorio. No se requiere la instalación y configuración individual en cada dispositivo de usuario, lo que reduce los costos de implementación. Además, el mantenimiento centralizado en el servidor también puede reducir los costos y la complejidad de soporte técnico.

1.3 Herramientas y tecnologías

En este capítulo, se explorarán las tecnologías y herramientas que se pueden utilizar para el desarrollo de una aplicación web. El objetivo es proporcionar una visión general de las herramientas y tecnologías disponibles para los desarrolladores y evaluar sus ventajas y desventajas.

1.3.1 Lenguaje del lado del servidor

El lenguaje del lado del servidor, también conocido como *backend*, es una forma de programación que se centra en el funcionamiento y procesamiento de los servidores web. Este lenguaje se utiliza para llevar a cabo operaciones y acciones en los servidores, como gestionar bases de datos, procesar formularios, interactuar con otros servicios y generar contenido dinámico para mostrar en las aplicaciones o sitios web ¹⁷.

A continuación, los lenguajes analizados:

PHP:

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en llevar a cabo operaciones y acciones del lado del servidor, pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica ¹⁸.

Algunas ventajas de PHP son ¹⁹:

- Lenguaje totalmente libre y abierto.
- Curva de aprendizaje muy baja.
- Los entornos de desarrollo son de rápida y fácil configuración.
- Fácil despliegue: paquetes totalmente autoinstalables que integran PHP.
- Fácil acceso a bases de datos.
- Comunidad muy grande.

Algunas desventajas de PHP son ¹⁹:

- Al ser de código abierto, todo el mundo puede ver el código fuente. Si hay errores en el código fuente, la gente puede utilizarlo para explorar sus puntos débiles.
- Tiene una gran cantidad de *frameworks* y bibliotecas disponibles, lo que puede ser una ventaja en términos de elección y flexibilidad. Sin embargo, esto también puede llevar a un ecosistema fragmentado, donde diferentes proyectos utilizan diferentes

frameworks y bibliotecas, lo que dificulta la colaboración y la reutilización de código entre proyectos.

- A medida que los proyectos web crecen en tamaño y complejidad, puede tener dificultades para escalar eficientemente. Aunque existen soluciones como la implementación de caché y la optimización del código, no es tan eficiente en términos de rendimiento como algunos otros lenguajes.

Python:

Python es un lenguaje de programación interpretado y de alto nivel creado por Guido van Rossum en 1991. Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de plataformas ²⁰.

Algunas ventajas de Python son ²¹:

- Los desarrolladores pueden leer y comprender fácilmente los programas de Python debido a su sintaxis básica similar a la del inglés.
- Permite que los desarrolladores sean más productivos, ya que pueden escribir un programa de Python con menos líneas de código en comparación con muchos otros lenguajes.
- La comunidad activa de Python incluye millones de desarrolladores alrededor del mundo que prestan su apoyo. Si se presenta un problema, puede obtener soporte rápido de la comunidad.
- Hay muchos recursos útiles disponibles en Internet si desea aprender Python. Por ejemplo, puede encontrar con facilidad videos, tutoriales, documentación y guías para desarrolladores.

Algunas desventajas de Python son ²²:

- El consumo de memoria, un detalle importante de este lenguaje de programación es que no es el indicado para proyectos que requieren de mucha memoria de procesamiento y espacio.

- Al tratarse de un lenguaje con mucha versatilidad, puede llegar a procesar de forma lenta, puesto que no tiene un uso específico y complicado definido.
- El hecho de que sea un lenguaje bastante versátil, no quiere decir que sea eficiente en todos los campos en los que se puede ejecutar. En el caso del desarrollo de aplicaciones móviles, no es la mejor opción para ello, ya que existen otros lenguajes más especializados y que ofrecen mejores resultados.

Java:

Java es una plataforma informática de lenguaje de programación creada por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, ya que es una plataforma fiable en la que se crean muchos servicios y aplicaciones. Los nuevos e innovadores productos y servicios digitales diseñados para el futuro también siguen basándose en Java ²³.

JAVA es un lenguaje de programación orientado a objetos. Es un lenguaje de programación muy transferible que se usa en plataformas y diferentes tipos de dispositivos, desde teléfonos inteligentes hasta televisores inteligentes. Se usa para crear aplicaciones móviles y web, software empresarial, dispositivos de Internet de las cosas, juegos, aplicaciones distribuidas y basadas en la nube, entre otros tipos ²⁴.

Algunas ventajas de Java son ²⁵:

- Es un lenguaje de programación muy popular y maduro, con una gran comunidad de desarrolladores y una amplia variedad de bibliotecas y herramientas disponibles para trabajar con datos a gran escala.
- Ha sido desarrollado teniendo en cuenta la seguridad, lo que lo hace adecuado para el desarrollo de aplicaciones confiables y seguras. La plataforma de ejecución Java ofrece características de seguridad, como la administración de permisos y configuraciones de *sandbox*, que ayudan a proteger contra amenazas de seguridad.
- Es un lenguaje multiplataforma y portable, lo que permite ejecutar el mismo código en diferentes sistemas operativos y arquitecturas sin necesidad de modificarlo o recompilarlo.

- Está diseñado para seguir el paradigma de programación orientada a objetos, lo que permite organizar el código en clases y objetos, fomentando la reutilización de código y la modularidad.
- Es conocido por su robustez y estabilidad. Su sintaxis estricta, gestión automática de memoria y chequeo de excepciones en tiempo de compilación garantizan una mayor confiabilidad en la ejecución de los programas.

Algunas desventajas de Java son ²⁵:

- Es un lenguaje verboso y complejo, lo que puede dificultar la lectura y la escritura del código, especialmente para principiantes o programadores acostumbrados a lenguajes más simples o expresivos.
- Es un lenguaje orientado a objetos, lo que puede no ser el paradigma más adecuado para trabajar con datos estructurados o no estructurados, que suelen requerir un enfoque más funcional o declarativo.
- Es un lenguaje que consume mucha memoria y recursos, lo que puede afectar al rendimiento y la escalabilidad de las aplicaciones de big data, especialmente si se comparan con lenguajes más ligeros o nativos.

Selección del lenguaje del lado del servidor:

Luego del análisis de los lenguajes anteriores se decide utilizar para el desarrollo del sistema el lenguaje Java en su versión 10.0.1. Por tener una programación orientada a objeto, ser de fácil uso y ofrecer intercambio de información, tiene un modelo de objetos más simple y elimina herramientas de bajo nivel y gestiona datos de fecha y hora de forma bastante natural y fácil de comprender. Además, es uno de los lenguajes más usados en la universidad, del que se posee una amplia documentación y el que domina más el desarrollador.

1.3.2 Lenguaje del lado del cliente

Los lenguajes del lado cliente se usan para su integración en páginas web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado, no compilado. Con la programación del lado del cliente se pueden validar algunos de los datos en la máquina cliente antes de enviarlos al servidor. Esto proporciona a los usuarios informes de error inmediatos, mientras siguen en esa página de formulario y sin necesidad de volver atrás tras recibir un mensaje de error ²⁶.

HTML:

HTML (Lenguaje de Marcas de Hipertexto, del inglés *HyperText Markup Language*) originalmente diseñado como un lenguaje para describir semánticamente documentos científicos, se ha adaptado para describir la estructura básica de páginas web y aplicaciones en línea. HTML utiliza un conjunto de etiquetas que el navegador interpreta y se emplean para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, tablas, vídeos, entre otros ²⁷.

CSS:

CSS (*Cascading Style Sheets* - en español Hojas de Estilo en Cascadas) es usado para darle estilo y diseño a las páginas Web, como cambiar la fuente de letra, color, tamaño y el espaciado de tu contenido; dividir en múltiples columnas, o agregar animaciones y otras propiedades decorativas. Fue creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML bien definidos y con significado completo, también llamados documentos semánticos. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes ²⁸.

JavaScript:

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. JavaScript es uno de los más populares en la actualidad esto se debe por su multiplataforma, ya que se puede utilizar en Windows, Linux o MAC o en el navegador de tu preferencia. Es liviano, imperativo y estructurado, mediante un conjunto de instrucciones indica al computador que tarea debe realizar. Es orientado a objetos y eventos, usa prototipo en vez de clases para el uso de herencia. Es interpretado, no se compila para poder ejecutarse ²⁹.

1.3.3 Herramienta CASE Visual Paradigm

Una herramienta CASE es un programa especializado en el control y desarrollo de aplicaciones informáticas, siguiendo alguna de las metodologías más extendidas como los diagramas de

control de flujo de Yourdon o los diagramas Entidad / relación de P.P. Chen para la normalización de bases de datos. En grandes equipos de desarrollo, el sistema queda centralizado a través de un diccionario de datos o "repositorio" que coordina los desarrollos de todos los participantes ³⁰.

El *Visual Paradigm* es una herramienta CASE que ha sido concebida para soportar el ciclo de vida entero del proceso de desarrollo del software, mediante de representación de todo tipo de diagramas. Este se caracteriza porque dispone en múltiples plataformas como Windows y Linux, usa un lenguaje estándar común para que todo el equipo de desarrollo tenga más facilidad en la comunicación. El diseño está centrado en casos de uso y enfocado al negocio para procrear un software de mayor calidad, también se debe al modelo y código que permanece sincronizado en todo el ciclo de desarrollo, gracias a su generación de código para Java, exportación como HTML y de importación y exportación de ficheros XMI. Su licencia es gratuita y comercial, es fácil de instalar y actualizar, posee varios idiomas y soporta aplicaciones web, tiene compatibilidad entre ediciones, las imágenes, informes y reportes generados, no son de muy buena calidad. Soporta la versión de UML 2.1, este además posee diagramas de Procesos de Negocio – Proceso, diagramas de flujo de datos, Decisión, Actor de negocio, Documento, modelado colaborativo con CVS (Control de versiones y subversión), interoperabilidad con modelos UML2 a través de XMI. Posee las capacidades de ingeniería directa e inversa, de ida y vuelta, en las inversas se encuentra el código de modelo, código a diagrama, Java, C++, Esquemas XML, XML, Net exe/dll, CORBA IDL. Tiene integrado Visio – Dibujo de diagramas UML con plantillas (stencils) de Microsoft Visio. Trae consigo editores de figuras, de detalles de los casos de uso, entorno todo e uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso ³¹.

1.3.4 Sistema gestor de base de datos MySQL

Un sistema gestor de bases de datos (SGBD) es un software constituido por una serie de programas dirigidos a crear, gestionar y administrar la información que se encuentra en la base de datos. Su principal objetivo es servir de interfaz entre los usuarios y las aplicaciones para facilitar la organización de los datos, garantizar su accesibilidad, calidad e integridad, brin-

dando a su vez una manera eficaz de administrar esa información. El lenguaje de manipulación, un lenguaje de definición de datos y un lenguaje de consulta son los tres componentes de un gestor de base de datos; por lo que puede trabajar a diferentes niveles, pero es invisible para el usuario final ³².

El sistema gestor de bases de datos MySQL consiste en un gestor de base de datos relacionales, soporta diferentes tipos de datos y está orientado a objetos. Es de código abierto, es decir, cuentan con una comunidad de desarrolladores que implementan mejoras o resuelve errores de forma altruista. Emplea un lenguaje SQL basado en el estándar ISO/IEC. Por ello, permite importar consultas y código de scripts de otras bases de datos. Cumple con el modelo ACID, es decir, los datos almacenados cuentan con las características de atomicidad, consistencia, aislamiento y durabilidad. Ofrece opciones que otras bases de datos no tienen como la posibilidad de crear tablas heredadas, esquemas o *triggers* ³³.

1.3.5 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente. Aumenta la productividad de los desarrolladores al combinar capacidades como editar, crear, probar y empaquetar software en una aplicación fácil de usar. Así como los escritores utilizan editores de texto y los contables, hojas de cálculo, los desarrolladores de software utilizan IDE para facilitar su trabajo ³⁴.

Visual Studio Code:

El Visual Studio Code es un editor de texto gratuito, de código abierto y multiplataforma lanzado por Microsoft. Se basan en la edición en línea de Visual Studio Online y un editor multiplataforma implementado con Electron de GitHub. Una característica muy útil de resaltado de sintaxis y autocompletado que proporciona autocompletado basado en tipos de variables, definiciones de funciones y módulos de importación. Trae consigo un editor de herramientas de depuración de código incorporado. Todos los paquetes que se encuentren están contruidos con JavaScript, por lo que cualquiera puede escribir fácilmente sus propias extensiones ³⁵.

NetBeans:

NetBeans IDE es una aplicación de código abierto (“open source”) diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de paquetes ³⁶.

1.3.6 Servidor de aplicaciones web Apache

Un servidor web es un software que forma parte del servidor y tiene como misión principal devolver información (páginas) cuando recibe peticiones por parte de los usuarios. Al nivel más básico, cuando un navegador necesita un archivo que está almacenado en un servidor web, el navegador requerirá el archivo al servidor mediante el protocolo HTTP. Cuando la petición alcanza al servidor web correcto (hardware), el servidor HTTP (software) acepta la solicitud, encuentra el documento requerido y lo envía de regreso al navegador, también a través de HTTP ³⁷.

Apache es un servidor web que se encarga de almacenar, procesar y servir las páginas web a los usuarios de las mismas. Creado por el desarrollador de software estadounidense Robert McCool en 1995, es considerado una de las tecnologías que impulsó el crecimiento inicial de Internet. Desde entonces, es ofrecido por la Apache Software Foundation, una organización sin fines de lucro que se encarga de supervisar la comunidad de usuarios que lo desarrollan y mantienen ³⁸.

Entre las principales ventajas del servidor web apache se encuentran las siguientes:

- De código abierto y gratuito, con una gran comunidad de usuarios.
- Parches de seguridad regulares y actualizados con frecuencia.
- Estructura basada en módulos.
- Multiplataforma: está disponible en servidores Windows y Linux.
- Personalización independiente en cada hosting.
- Compatible con los principales CMS y tiendas online y plataformas e-learning.

1.3.7 Marco de trabajo para el desarrollo de la solución informática

Un *framework* es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo de software. Utilizar un *framework* permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código. Un *framework* es por tanto un conjunto de herramientas y módulos que pueden ser reutilizados para varios proyectos ³⁹.

Spring Boot:

El *framework* Spring Boot en cuanto a seguridad tiene activos mecanismos incluidos para proteger tu base de datos, formularios y JavaScript. Se puede utilizar el *framework* para un desarrollo sencillo, hasta uno mucho más complejo, ambos casos funcionaran de manera estable y con rapidez. Su interfaz para acceso a la base de datos y realizar consultas proporciona gran velocidad. Al estar escrito en Java se puede ejecutar en muchas plataformas como Windows, OS X, entre otros, dándole gran libertad al programador al momento de ejecutar las aplicaciones ⁴⁰.

Bootstrap:

El *framework* Bootstrap es una potente herramienta para crear interfaces de usuario pulcras y adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Ofrece un soporte extraordinario con HTML5 y CC3. Así, los usuarios pueden hacer uso de él con una gran flexibilidad, y obteniendo unos resultados excelentes. Es sencillo de manejar gracias a un sistema GRID que permite realizar un diseño haciendo uso de 12 columnas para insertar el contenido, los usuarios pueden crear sitios web *responsive* de una manera mucho más sencilla e intuitiva. Bootstrap permite insertar imágenes *responsive* de una forma muy fácil ⁴¹.

1.4 Metodología de desarrollo de software utilizada

La metodología de desarrollo de software es el conjunto de técnicas y métodos que se utilizan para diseñar una solución de software informático. Trabajar con una metodología es imprescindible por una cuestión de organización. Por otra parte, las metodologías también sirven para controlar el desarrollo del trabajo. Esto sirve para minimizar los márgenes de errores y anticiparse a esa situación. Otra ventaja de utilizar una metodología es que te hace ahorrar tiempo y gestionar mejor los recursos disponibles ⁴².

Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y tradicionales (o robustas) ⁴³.

- **Metodologías de desarrollo de software ágiles:** Se basan en la entrega rápida y frecuente de software funcional, la colaboración entre el equipo de desarrollo y el cliente, y la adaptación constante a los cambios en los requisitos del proyecto.
- **Metodologías de desarrollo de software tradicionales:** Se centran en la planificación y el diseño detallado antes de la implementación, y se basan en la revisión y aprobación de diferentes etapas del proceso de desarrollo antes de avanzar al siguiente paso.

Para la implementación del sistema se decide utilizar una metodología ágil. Las metodologías tradicionales ya no bastan para este cometido, no se adaptan a las nuevas expectativas de los usuarios y a las exigencias del mercado. Las metodologías ágiles permiten centrarse en la correcta obtención del módulo sin importar cuán documentado esté, enfatizando en la rapidez y la calidad con que se desarrolle el mismo.

SCRUM:

Scrum es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990. Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint. El trabajo realizado dentro de un sprint (el número de éstos que requiere cada actividad estructural variará en función de la complejidad y tamaño del producto) se adapta al problema en cuestión y se define —y con frecuencia se modifica— en tiempo real por parte del equipo Scrum ⁴³.

La característica innovadora de Scrum es su fase central, a saber, los ciclos *sprint*. Un sprint de Scrum es una unidad de planeación en la que se valora el trabajo que se va a realizar, se seleccionan las particularidades por desarrollar y se implementa el software. Al final de un sprint, la funcionalidad completa se entrega a los participantes. Las características claves de este proceso son las siguientes ⁴⁴:

- Los sprints tienen longitud fija, por lo general de dos a cuatro semanas.

- El punto de partida para la planeación es la cartera del producto, que es la lista de trabajo por realizar en el proyecto. Durante la fase de valoración del sprint, esto se revisa, y se asignan prioridades y riesgos. El cliente interviene estrechamente en este proceso y al comienzo de cada sprint puede introducir nuevos requerimientos o tareas.
- La fase de selección incluye a todo el equipo del proyecto que trabaja con el cliente, con la finalidad de seleccionar las características y la funcionalidad a desarrollar durante el sprint.
- Una vez acordado, el equipo se organiza para desarrollar el software. Con el objetivo de revisar el progreso y, si es necesario, volver a asignar prioridades al trabajo, se realizan reuniones diarias breves con todos los miembros del equipo. Durante esta etapa, el equipo se aísla del cliente y la organización, y todas las comunicaciones se canalizan a través del llamado “maestro de Scrum”. El papel de este último es proteger al equipo de desarrollo de distracciones externas. La forma en que el trabajo se realiza depende del problema y del equipo. A diferencia de XP, Scrum no hace sugerencias específicas sobre cómo escribir requerimientos, desarrollar la primera prueba, entre otras. Sin embargo, dichas prácticas XP se usan cuando el equipo las considera adecuadas.
- Al final del sprint, el trabajo hecho se revisa y se presenta a los participantes. Luego comienza el siguiente ciclo de sprint.

XP:

La metodología XP es una metodología de desarrollo formulada por Kent Beck, es el más destacado de los procesos ágiles de desarrollo de software. se diferencia de las metodologías tradicionales principalmente porque está diseñada para adaptarse a los procesos de desarrollo en constante cambio. Se aplica de manera dinámica durante todo el ciclo de vida del software. Propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito. El ciclo de vida ideal de XP consiste de cuatro etapas: Planeación, Diseño, Codificación y Pruebas⁴⁵.

XP es adecuada para proyectos los cuales tienen requisitos imprecisos y con muchos cambios, donde pueden existir altos riesgos técnicos.

Cuenta con varias características como son:

- Metodología basada en prueba y error para obtener un software que funcione realmente.
- Fundamentada en principios.
- Está orientada hacia quien produce y usa software (el cliente participa muy activamente).
- Reduce el coste del cambio en todas las etapas del ciclo de vida del sistema.
- Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo.
- Cliente bien definido.
- Los requisitos pueden cambiar.
- Grupo pequeño y muy integrado (2-12 personas).
- Equipo con formación elevada y capacidad de aprender.

Selección de la metodología de desarrollo de software

La necesidad de desarrollar el sistema para que esté disponible lo antes posible garantizando su calidad, el número de personas que participan en su desarrollo y la utilización de esta metodología en la construcción de la plataforma a la cual se integrará la solución, llevan a escoger XP como metodología de desarrollo de software.

Conclusiones del capítulo

Durante este capítulo se abordaron los elementos teóricos que sustentan a la propuesta de solución del problema planteado, en este sentido se concluye:

- La definición de conceptos relacionados con el dominio del problema posibilitó una mayor comprensión del caso de estudio.
- El análisis de plataformas homólogas determinó que estas últimas no pueden ser propuestas como solución emergente a la problemática. Sin embargo, sirvieron para detectar características que fueron útiles para el desarrollo de la propuesta de solución.
- A partir de lo requerido por el cliente se estableció como metodología de desarrollo XP; así como lenguajes de programación Java y JavaScript y tecnologías como Spring Boot y MySQL.

CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

En este capítulo se definen los requisitos funcionales y no funcionales con los que deberá cumplir la aplicación web. Igualmente se definen los principales artefactos de ingeniería de software propuestos por la metodología utilizada.

2.1 Descripción de la propuesta de solución

Una vez se acceda a la dirección web de la aplicación se mostrará la ventana de login donde se necesita el usuario y la contraseña correcta para poder acceder al sistema. Una vez autenticado el usuario, se mostrará la ventana principal con las gráficas de estadísticas, y las diferentes opciones para acceder a las diferentes ventanas de gestión donde podrá añadir, actualizar o agregar la información necesaria.

2.2 Historias de Usuario

Se utilizan para especificar los requisitos de las aplicaciones software en las metodologías ágiles (SCRUM, XP, FDD, ASD, AUP, LD, entre otros). Las historias de usuario son tarjetas en donde el interesado describe brevemente (con el fin de que sean dinámicas y flexibles) las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que se pueda implementar en unas tres semanas aproximadamente (para no superar el tamaño de una iteración, que es el tiempo estimado para una entrega de un componente de desarrollo de manera incremental ⁴⁶). A continuación, se describen las HU definidas:

Tabla 2: HU 1

Historia de Usuario	
Número: 1	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Doctorando	
Usuario: Usuario	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1

Descripción: El sistema debe ofrecer la opción de crear un nuevo doctorando, con los campos de nombre, apellidos, carnet de identidad, sexo, centro, plantilla, fecha de defensa, fecha de ingreso, tema de tesis, fecha de especialidad, desarrollo local, programa, facultad, y sectores estratégicos, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar los doctorandos.
Observación: Depende de que haya sector estratégico, área y programas registrados en el sistema.

Tabla 3: HU 2

Historia de Usuario	
Número: 2	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Doctor	
Usuario: Usuario	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear un nuevo doctor, con los campos de nombre, apellidos, carnet de identidad, sexo, centro, plantilla y facultad, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar los doctores.	
Observación: Depende de que haya Área registrada en el sistema.	

Tabla 4: HU 3

Historia de Usuario	
Número: 3	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Tutoría	
Usuario: Usuario	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de asignar un tutor seleccionando un doctor y un doctorando, así como brindar la posibilidad de buscar, quitar y consultar los tutores.	
Observación: Depende de que haya doctorandos y doctores registrados en el sistema.	

El resto de las Historias de Usuarios se encuentra en los [anexos](#).

2.3 Requisitos no funcionales

Los requisitos no funcionales son las restricciones o los requisitos impuestos al sistema. Por lo general, se denotan con adjetivos como seguridad, rendimiento y escalabilidad. Los requisitos no funcionales son importantes porque ayudan a garantizar que el sistema satisfaga las necesidades del usuario ⁴⁷.

A continuación, se muestran los requisitos no funcionales de la propuesta de solución:

Usabilidad:

RNF1. El sistema debe poder ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

Seguridad:

RNF2. El sistema debe garantizar la autenticación como primera acción para el acceso a la aplicación; esta consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica.

Software:

RNF3. El sistema debe tener visibilidad en los principales navegadores web como Chrome v.45, Firefox v.38, Safari v.9, Opera v.30.

2.4 Tarjetas CRC

Las tarjetas CRC son en la práctica pequeñas tarjetas de cartón que se elaboran para ser mostradas al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas, lo que ayuda al equipo durante el diseño e implementación del sistema. Estas constituyen documentación adicional que será adjuntada a las HU ⁴⁸.

Cada tarjeta representa una clase en la programación orientada a objetos. Las tarjetas CRC se componen de los siguientes elementos ⁴⁹:

Clase: es cualquier persona, cosa, evento, concepto, pantalla o reporte.

Responsabilidades: es lo que debe hacer la clase, sus atributos y métodos.

Colaboradores: son el resto de las clases con las que interactúa para llevar a cabo sus responsabilidades.

A continuación, se describen las tarjetas CRC definidas:

Tabla 5: Tarjeta CRC “doctorando”

Tarjeta CRC	
Clase: doctorando	
Responsabilidad:	Colaboración:
-registrar los doctorandos -modificar los doctorandos -eliminar los doctorandos -buscar los doctorandos -mostrar el listado de los doctorandos	-programa -sectorest -area -publicacion -lineainvest -solicitante -persona

Tabla 6: Tarjeta CRC “doctor”

Tarjeta CRC	
Clase: doctor	
Responsabilidad:	Colaboración:
-registrar los doctores -modificar los doctores -eliminar los doctores -buscar los doctores -mostrar el listado de los doctores	-area -persona

Tabla 7: Tarjeta CRC “tutor”

Tarjeta CRC	
Clase: tutor	
Responsabilidad:	Colaboración:
-asignar los tutores -quitar los tutores -buscar los tutores -mostrar el listado de los tutores	-doctor -doctorando

Tabla 8: Tarjeta CRC “publicacion”

Tarjeta CRC	
Clase: publicacion	
Responsabilidad:	Colaboración:
-registrar las publicaciones -modificar las publicaciones -eliminar las publicaciones -buscar las publicaciones -mostrar el listado de las publicaciones	-indexado

El resto de las tarjetas CRC se encuentran en los anexos.

2.5 Plan de lanzamiento

El plan de lanzamiento e iteraciones se define para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha historia⁵⁰. Como resultado de la priorización de historias se llegó a la siguiente planificación:

Tabla 9: Plan de lanzamiento

Lanzamiento	Descripción de la iteración	Orden de las HU a implementar	Duración Total
Iteración 1	En esta iteración se realizarán las HU que tienen prioridad alta.	HU 14, HU 12, HU 11, HU 10, HU 6, HU 1, HU 2, HU 3	8 semanas
Iteración 2	En esta iteración se realizarán las HU que tienen prioridad media.	HU 4, HU 5, HU 7, HU 8, HU 15	6 semanas
Iteración 3	En esta iteración se realizarán las HU que tienen prioridad baja.	HU 9, HU 13, HU 16, HU 17	6 semanas

2.6 Patrón de arquitectura

Un patrón de arquitectura es un conjunto de principios y directrices que ayudan a estructurar y organizar una aplicación o sistema a nivel de arquitectura. Estos patrones proporcionan soluciones probadas y prácticas para desafíos comunes en el diseño de software.

MVC (Modelo-Vista-Controlador) es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización. Esta "separación de preocupaciones" proporciona una mejor división del trabajo y una mejora de mantenimiento ⁵¹.

En Spring Boot, los conceptos de Servicio, Repositorio, Modelo, Controlador y Vista están relacionados con el patrón de arquitectura Modelo-Vista-Controlador (MVC) y se utilizan para organizar y estructurar una aplicación ⁵².

- Servicio: Es una capa de la aplicación que contiene la lógica de negocio y realiza operaciones específicas. Los servicios encapsulan la funcionalidad de la aplicación y se encargan de coordinar las operaciones entre los controladores y los repositorios.
- Repositorio: Es una capa de la aplicación que se encarga de la persistencia y el acceso a los datos. Los repositorios se utilizan para interactuar con la base de datos o cualquier otra fuente de datos, y proporcionan métodos para recuperar, almacenar, actualizar y eliminar objetos. Los repositorios están asociados con las entidades del modelo y suelen utilizar tecnologías de mapeo objeto-relacional (ORM) como Hibernate o JPA (Java Persistence API).
- Modelo: Se refiere a las clases que representan los datos y la lógica de negocio de la aplicación. Las clases de modelo suelen ser entidades persistentes que se almacenan en la base de datos y se utilizan para transferir datos entre diferentes capas de la aplicación. Estas clases pueden contener propiedades, métodos y relaciones con otras entidades.
- Controlador: Es una capa de la aplicación que se encarga de recibir las solicitudes del cliente, procesar la lógica de negocio correspondiente y devolver las respuestas

adecuadas. Los controladores actúan como intermediarios entre las solicitudes del cliente y los servicios o repositorios correspondientes.

- Vista: En Spring Boot representa la interfaz de usuario de la aplicación. En el caso de las aplicaciones web, la vista suele ser una página HTML, una plantilla Thymeleaf, una página JSP o algún otro tipo de representación visual. La vista es responsable de mostrar los datos proporcionados por los controladores y permitir la interacción del usuario.

A continuación, se muestra una ilustración de elaboración propia sobre la representación de la arquitectura en Spring Boot:

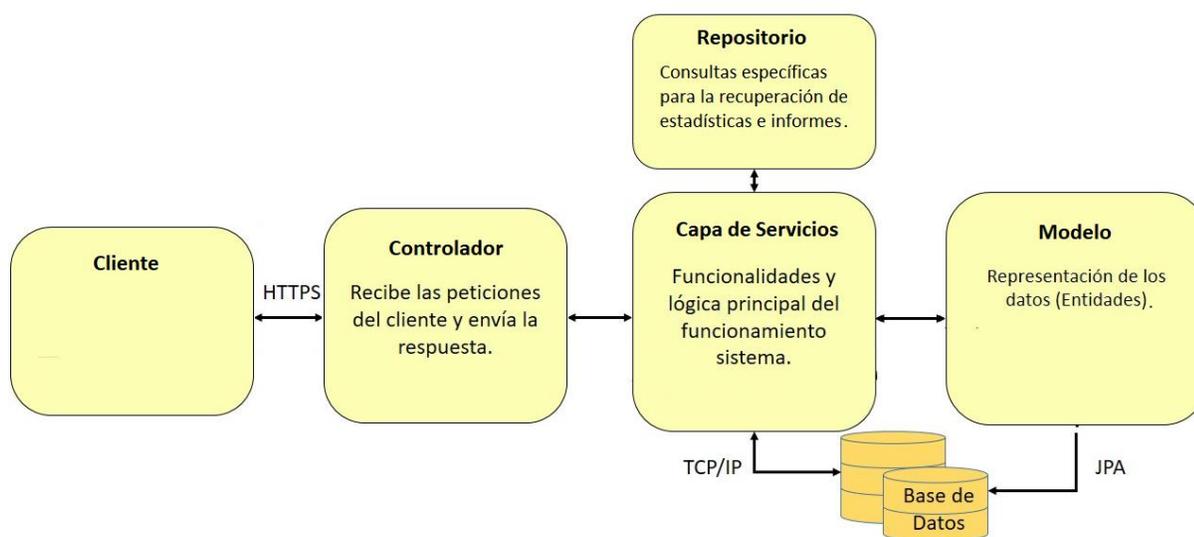


Ilustración 1: Representación de la arquitectura.

2.7 Patrones de Diseño

Si bien la elaboración de un buen diseño del software contribuye directamente a la calidad del producto final, gran parte de esta yace en la utilización adecuada de los patrones de diseño y arquitectónicos existentes en la actualidad.

Los patrones son soluciones simples y elegantes a problemas específicos y comunes en el diseño orientado a objetos. Son soluciones basadas en la experiencia y han demostrado que funcionan ⁵³.

A continuación, se muestran los principales patrones empleados en el diseño e implementación de la propuesta de solución.

Patrones GRASP:

- Alta Cohesión: La información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. Se usó en la clase: *area.java*.
- Bajo Acoplamiento: Tener las clases lo menos ligadas entre sí que se pueda. De tal forma que, en caso de producirse una modificación en alguna. Se usó en la clase: *areadeconocimientoController.java*.
- Experto: Es el principio básico de asignación de responsabilidades. Se indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Se usó en la clase: *doctorandoController.java*.
- Creador: Identificar correctamente el responsable de la instanciación de las diferentes clases, buscando que los objetos sean creados por la clase que dispone de la información que se requiere para crear los mismos. Se usó en la clase: *solicitanteController.java*.
- Controlador: Se utilizan los controladores para servir de intermediarios entre la interfaz y el algoritmo que contiene la lógica del sistema. Se usó en la clase: *tutorController.java*.

Patrones GOF

- Facade: Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema.
- Chain of Responsibility: Permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada.

2.8 Modelo de Datos

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos. Los modelos de bases de datos individuales se diseñan en base a las reglas y los conceptos de cualquier modelo de datos más amplio que los diseñadores adopten. La mayoría de los

modelos de datos se pueden representar por medio de un diagrama de base de datos acompañante ⁵⁴.

A continuación, se muestra una imagen de elaboración propia del modelo de datos del sistema:

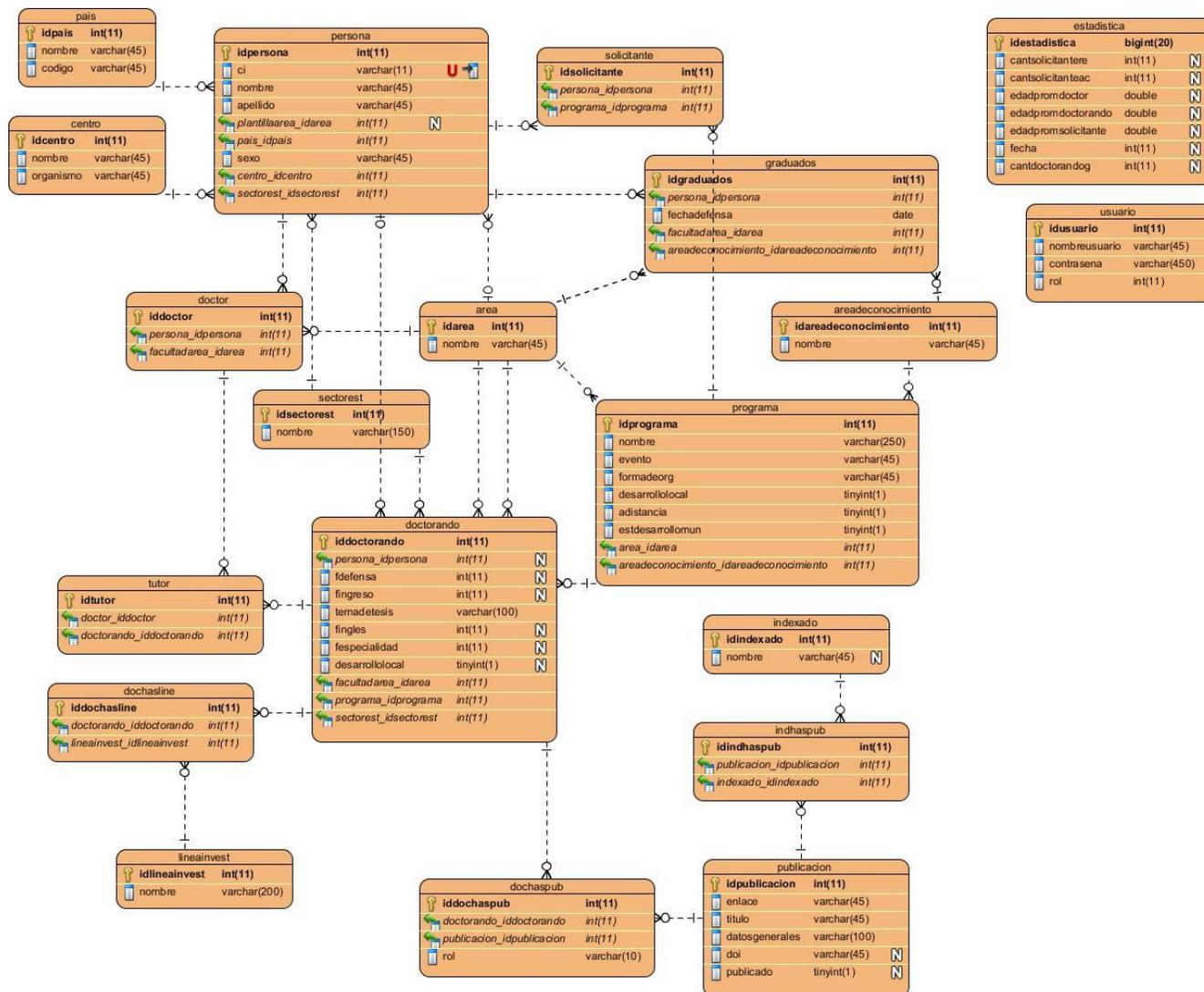


Ilustración 2: Modelo de datos

Conclusiones del capítulo

En este capítulo se muestran los principales artefactos de ingeniería de software propuestos por la metodología utilizada, en este sentido se concluye:

- La definición de las historias de usuarios garantizó que la solución responda a las necesidades del cliente, sirviendo de guía para el desarrollo de las funcionalidades del sistema de gestión de la información.

- Las tarjetas CRC ayudaron al equipo a definir las interacciones y responsabilidades de las clases durante el diseño e implementación del sistema.

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se analizan los aspectos relacionados con la implementación y las pruebas realizadas a la aplicación.

3.1 Tareas de Ingeniería

En la fase de Planeación se definieron los elementos funcionales necesarios para realizar el desarrollo del sistema, lo que implica que ya está disponible para continuar a la fase de implementación, donde las Tareas de Ingeniería (TI) es el artefacto usado. Estas son el artefacto que se encarga de almacenar los detalles para la realización de cada una de las HU ⁵⁵. A continuación, se describen las TI definidas:

Tabla 10: Tarea de ingeniería 1

Tarea de ingeniería	
Número de tarea:1	Número de Historia de Usuario: n/a
Nombre de la tarea: Diseño de la base de datos	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Diseñar la base de datos con los campos y entidades necesarias.	

Tabla 11: Tarea de ingeniería 2

Tarea de ingeniería	
Número de tarea:2	Número de Historia de Usuario: n/a
Nombre de la tarea: Crear la base de datos	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Crear la base de datos con los campos y entidades necesarias.	

Tabla 12: Tarea de ingeniería 3

Tarea de ingeniería	
Número de tarea:3	Número de Historia de Usuario: 10
Nombre de la tarea: Gestionar Área del Conocimiento	

Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de las áreas de conocimiento.	

Tabla 13: Tarea de ingeniería 4

Tarea de ingeniería	
Número de tarea:4	Número de Historia de Usuario: 11
Nombre de la tarea: Gestionar Área	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de las áreas.	

Tabla 14: Tarea de ingeniería 5

Tarea de ingeniería	
Número de tarea:5	Número de Historia de Usuario: 12
Nombre de la tarea: Gestionar Centro	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de los centros.	

Tabla 15: Tarea de ingeniería 6

Tarea de ingeniería	
Número de tarea:6	Número de Historia de Usuario: 14
Nombre de la tarea: Gestionar Sector Estratégico	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de los sectores estratégicos.	

Tabla 16: Tarea de ingeniería 7

Tarea de ingeniería	
Número de tarea:7	Número de Historia de Usuario: 1
Nombre de la tarea: Gestionar Doctorando	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de los doctorandos.	

El resto de las tareas de ingeniería se encuentran en los [anexos](#).

3.2 Pruebas

La prueba de software es el proceso de evaluar y verificar que un producto o aplicación de software hace lo que se supone que debe hacer. Los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento ⁵⁶. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

3.2.1 Pruebas Unitarias

Las pruebas unitarias son una práctica de desarrollo de software que consiste en probar las unidades más pequeñas de código de un programa de forma aislada y automatizada. Una unidad, en este contexto, puede ser una función, método, clase o incluso un componente completo, dependiendo del nivel de granularidad que se desee abordar ⁵⁷.

El objetivo principal de las pruebas unitarias es verificar que cada unidad de código individual funcione correctamente y cumpla con los requisitos establecidos. Estas pruebas se centran en aislar la unidad bajo prueba de su entorno y de las dependencias externas, utilizando técnicas como el reemplazo de objetos reales por objetos simulados.

Las pruebas unitarias se les realizaron a las clases servicios mediante el uso de los conjuntos de librerías de mockito y junity de java. Se realizaron 20 pruebas unitarias, las cuales resultaron

100%efectivas. A continuación, se muestran parte del código y los resultados de la aplicación de estas pruebas a la clase `doctorandoService.java`.

```
//Se comprueba el método de devolver el listado de todos los doctorandos
@Test
public void testListAll() {
    // Crear una lista de doctorandos de muestra
    List<doctorando> doctorandos = new ArrayList<>();
    doctorandos.add(new doctorando());
    doctorandos.add(new doctorando());
    // Configurar el comportamiento esperado del repositorio mock
    when(repo.findAll()).thenReturn(doctorandos);
    // Llamar al método a probar
    List<doctorando> result = doctorandoService.listAll();
    // Verificar el resultado
    assertEquals(2, result.size());
}

//Se comprueba el método de guardar doctorandos
@Test
public void testSave() {
    // Crear un doctorando de muestra
    doctorando doctorando = new doctorando();
    // Llamar al método a probar
    doctorandoService.save(doctorando);
    // Verificar que el método save() del repositorio se haya llamado una vez con el
doctorando como argumento
    verify(repo, times(1)).save(doctorando);
}

//Se comprueba el método para obtener un doctorando por id
@Test
public void testGet() {
```

```
// Crear un doctorando de muestra con un ID específico
Long id = 1L;
doctorando doctorando = new doctorando();
doctorando.setIddoctorando(id);
// Configurar el comportamiento esperado del repositorio mock
when(repo.findById(id)).thenReturn(Optional.of(doctorando));
// Llamar al método a probar
doctorando result = doctorandoService.get(id);
// Verificar el resultado
assertEquals(id, result.getIddoctorando());
}
//se comprueba el método para eliminar doctorandos
@Test
public void testDelete() {
    // Crear un ID de doctorando de muestra
    Long id = 1L;
    // Llamar al método a probar
    doctorandoService.delete(id);
    // Verificar que el método deleteById() del repositorio se haya llamado una vez con
    el ID del doctorando
    verify(repo, times(1)).deleteById(id);
}
```



Ilustración 3: Resultados de la prueba unitaria a doctorandoService

El resto de las pruebas se muestran en los anexos.

3.2.2 Pruebas de Aceptación

Las pruebas de aceptación o pruebas clientes, se centran en las características y funcionalidades del sistema que son visibles y revisables por parte del cliente. Estas pruebas parten de las historias de usuario y los clientes son los encargados de escribir las pruebas para cada historia de usuario que deba validarse ⁵⁸. A continuación, se describen los casos de pruebas de aceptación definidos:

Tabla 17: Caso de Prueba de Aceptación HU1_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU1_P1	Nombre de Historia de Usuario: Gestionar Doctorando
Nombre: Registrar Doctorando	
Descripción: Prueba la funcionalidad de registrar un doctorando	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de doctorandos.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema registra un nuevo doctorando si los datos fueron correctos, en caso contrario muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 18: Caso de Prueba de Aceptación HU1_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU1_P2	Nombre de Historia de Usuario: Gestionar Doctorando
Nombre: Modificar Doctorando	
Descripción: Prueba la funcionalidad de modificar un doctorando	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de doctorandos.	
Pasos de Ejecución: El usuario busca el doctorando que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica al doctorando si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 19: Caso de Prueba de Aceptación HU1_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU1_P3	Nombre de Historia de Usuario: Gestionar Doctorando
Nombre: Eliminar Doctorando	
Descripción: Prueba la funcionalidad de eliminar un doctorando	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de doctorandos.	
Pasos de Ejecución: El usuario busca el doctorando que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina un doctorando del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 20: Caso de Prueba de Aceptación HU1_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU1_P4	Nombre de Historia de Usuario: Gestionar Doctorando
Nombre: Listar Doctorando	
Descripción: Prueba la funcionalidad de listar los doctorandos.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Doctorandos.	
Resultados esperados: El sistema carga una página con el listado de doctorandos.	
Evaluación de la prueba: Satisfactoria	

El resto de las tablas de las pruebas de aceptación se encuentran en los anexos.

3.3 Resultado de las pruebas

Con el objetivo de verificar que todos los requisitos funcionales fueron desarrollados correctamente se aplicaron las pruebas especificadas por la metodología XP. Se realizaron un total de 56 pruebas de aceptación y 20 pruebas unitarias. A continuación, se muestra una gráfica de elaboración propia con las cantidades de pruebas realizadas por iteración.

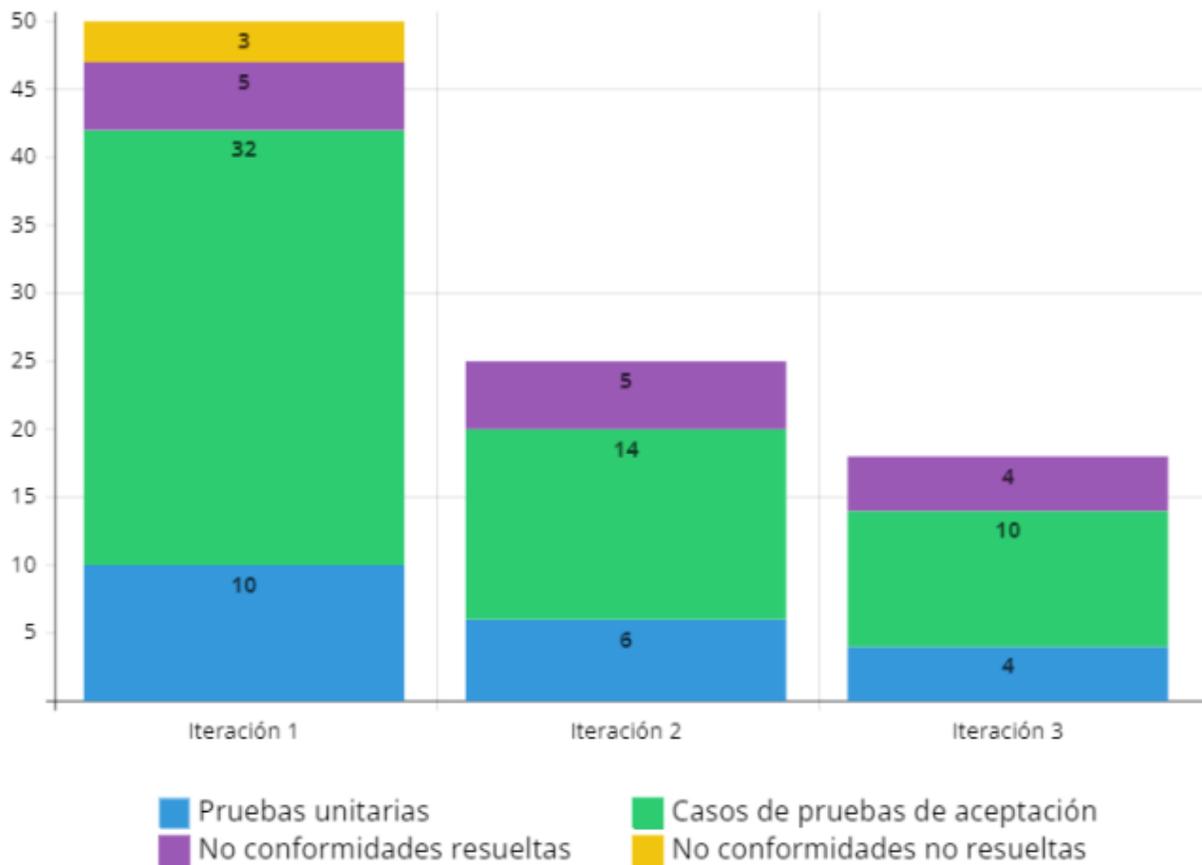


Ilustración 4: Pruebas realizadas

Conclusiones del capítulo

Tras representar las etapas de implementación y pruebas del software en este capítulo, se plantean las siguientes conclusiones:

- Las tareas de ingeniería ayudaron al equipo de proyecto a lograr identificar, controlar, rastrear los requisitos y los cambios en cualquier etapa mientras se desarrollaba el proyecto.
- Las pruebas unitarias al software permitieron comprobar la calidad del sistema propuesto.
- La ejecución de las pruebas de aceptación comprobó el correcto funcionamiento y el cumplimiento de los requisitos definidos.

CONCLUSIONES FINALES

Una vez concluida la investigación se obtuvo como resultado la implementación de la aplicación y todo lo referente al marco teórico de la investigación, lo cual permite arribar a las siguientes conclusiones:

- El análisis de los principales conceptos asociados a la investigación, fundamentó los elementos teóricos que propiciaron un mejor entendimiento de los procesos de gestión de la información de los doctorandos en la UCI.
- Las tecnologías y herramientas empleadas, como lenguajes de programación Java y JavaScript y tecnologías como Spring Boot y MySQL posibilitaron el desarrollo de una aplicación que cumple con los requisitos planteados.
- Se obtuvo un sistema web funcional, que contribuye a mejorar el proceso de gestión de la información de doctorandos en la UCI cumpliendo así con el objetivo general de la investigación.
- La ejecución de pruebas de software definidas permitió la comprobación de la correspondencia de las funcionalidades incluidas en el sistema desarrollado con las necesidades e intereses del cliente.

RECOMENDACIONES

En función del constante proceso de mejora y evolución que es inherente a todo sistema de software se recomienda lo siguiente:

- Incorporar nuevas funcionalidades para el cálculo de estadísticas.
- Mejorar la funcionalidad para la creación de reportes.

REFERENCIAS BIBLIOGRÁFICAS

1. *Postgrado – Universidad de La Habana*. (s. f.). Recuperado 4 de octubre de 2023, de <https://www.uh.cu/estudios/postgrado/>
2. Rodríguez, G. J. B., Suárez, J. L. A., Martínez, E. de la P., Alfonso, A. M. T., & Manzanet, J. E. A. (2020). La educación de posgrado ante el nuevo escenario generado por la COVID-19. *Educación Médica Superior*, 34(4), Article 4. <https://ems.sld.cu/index.php/ems/article/view/2718>
3. Ruiz Ortiz, L., & Pérez Mallea, I. (2021). Acciones para el desarrollo de la educación virtual de posgrado. *UCIENCIA 2021*. <https://repositorio.uci.cu/jspui/handle/123456789/9802>
4. Hernandez Luque, E. (2021). *Estrategia metodologica para la educacion de posgrado desde la gestion del conocimiento* [doctoralThesis, Universidad de las Ciencias Pedagogicas Enrique Jose Varona.]. <https://repositorio.uci.cu/jspui/handle/123456789/10151>
5. *Historia | Universidad de las Ciencias Informáticas*. (s. f.). Recuperado 4 de octubre de 2023, de <https://www.uci.cu/universidad/historia>
6. Benítez Menéndez, O. G., Ruiz Ortiz, L., Arias Verdecia, L. A., & Sánchez Hormigó, Y. (2021). Ecosistema digital educativo integrado al posgrado, para la educación ciencia tecnología y sociedad. *UCIENCIA 2021*. <https://repositorio.uci.cu/jspui/handle/123456789/9815>
7. Padilla, V. A. J. (2021). *LICENCIADO EN INFORMÁTICA ADMINISTRATIVA*.
8. Santiesteban Catallops, L. J. (2019). *Sistema de gestión de información tecnológica para la Facultad 1 de la Universidad de las Ciencias Informáticas* [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 1.]. <https://repositorio.uci.cu/jspui/handle/123456789/10067>
9. *Sistema de gestión ¿Qué es? ¿Cuántos tipos hay? - Evaluando ERP*. (2021, junio 1). <https://www.evaluandoerp.com/software-erp/sistema-de-gestion/>
10. Stair, R. M., & Reynolds, G. W. (2010). *Principles of information systems: A managerial approach* (9th ed). Course Technology Cengage Learning.

11. Ortigón Cortázar, G. (2015). Optimización de sistemas de gestión académica. Una propuesta de gestión, medición y procesamiento de datos en un entorno virtual de aprendizaje para la toma de decisiones en instituciones educativas. *Revista EAN*, 79, 80-97.
12. *Gaceta Oficial No.65 Ordinaria de 2019*. (2019). <https://www.gacetaoficial.gob.cu/es/gaceta-oficial-no-65-ordinaria-de-2019>
13. *DECRETO-LEY No. 372 | Juriscuba*. (s. f.). Recuperado 3 de abril de 2023, de <http://juriscuba.com/decreto-ley-no-372/>
14. *Student Information System (SIS) Solutions for Higher Ed. | Ellucian*. (s. f.). Recuperado 24 de junio de 2023, de <https://www.ellucian.com/solutions/student-information-systems>
15. *Universidad de Córdoba*. (s. f.). Recuperado 3 de octubre de 2023, de <http://www.uco.es/>
16. Lic. Rubisnel Almeida Saavedra & M. Sc. Gisber Miguel Góngora Mora. (2018). *Sistema de gestión de la Nueva Universidad (SIGENU v4.0.0) como herramienta para la automatización de los procesos docentes y estadísticos de la Universidad de Granma(Revisión)*. https://node2.123dok.com/dt02pdf/123dok_es/001/828/1828612.pdf.pdf?X-Amz-Content-Sha256=UNSIGNED-PAYLOAD&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=aa5vJ7sqx6H8Hq4u%2F20230625%2F%2Fs3%2Faws4_request&X-Amz-Date=20230625T182034Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Signature=69c4ac9ceb37ed79112f1f3da374ddc2c8968a845298292cdee10ceb2114e7fc
17. *Introducción al lado servidor—Aprende desarrollo web | MDN*. (2023, agosto 2). https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Introduction
18. *PHP: ¿Qué es PHP? - Manual*. (s. f.). Recuperado 5 de octubre de 2023, de <https://www.php.net/manual/es/intro-what-is.php>
19. *Historia de PHP sus ventajas, desventajas y para que se usa actualmente*. – *Academia de Joystick*. (2021, junio 1). <https://www.joystick.com.mx/historia-de-php-y-sus-ventajas-y-desventajas-y-para-que-se-usa-actualmente/>
20. *El tutorial de Python*. (s. f.). Python documentation. Recuperado 5 de octubre de 2023, de <https://docs.python.org/3/tutorial/index.html>
21. *¿Qué es Python? - Explicación del lenguaje Python - AWS*. (s. f.). Amazon Web Services, Inc. Recuperado 5 de octubre de 2023, de <https://aws.amazon.com/es/what-is/python/>

22. *Universidades de Chile*. (s. f.). Universidades de Chile. Recuperado 5 de octubre de 2023, de <http://universidadesonline.cl/>
23. *¿Qué es Java y por qué lo necesito?* (s. f.). Recuperado 5 de octubre de 2023, de https://www.java.com/es/download/help/whatis_java.html
24. *Java Documentation*. (s. f.). Oracle Help Center. Recuperado 3 de abril de 2023, de <https://docs.oracle.com/en/java/>
25. *Java—Ventajas y Características | Apuntes de Programación Java | Docsity*. (s. f.). Recuperado 6 de octubre de 2023, de <https://www.docsity.com/es/java-ventajas-y-caracteristicas/5795074/>
26. *Lenguajes de Cliente-Servidor | PDF | Xml | Lenguaje de programación*. (s. f.). Recuperado 7 de octubre de 2023, de <https://es.scribd.com/document/461464171/LENGUAJES-DE-CLIENTE-SERVIDOR>
27. *HTML: Lenguaje de etiquetas de hipertexto | MDN*. (2023, julio 24). <https://developer.mozilla.org/es/docs/Web/HTML>
28. *CSS | MDN*. (2023, marzo 13). <https://developer.mozilla.org/es/docs/Web/CSS>
29. *JavaScript | MDN*. (2023, febrero 20). <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
30. Buñay, R. I. H. (2018). *ANÁLISIS DE LA IMPORTANCIA DE LAS HERRAMIENTAS CASE EN EL DESARROLLO DE SOFTWARE*.
31. *Visual Paradigm User's Guides*. (s. f.). Recuperado 3 de abril de 2023, de <https://www.visual-paradigm.com/support/documents/>
32. *¿Para qué sirve un sistema gestor de base de datos? | Blog UE*. (2021, diciembre 27). Universidad Europea. <https://universidadeuropea.com/blog/para-que-sirve-gestor-base-datos/>
33. *MySQL :: MySQL Documentation*. (s. f.). Recuperado 3 de abril de 2023, de <https://dev.mysql.com/doc/>
34. *¿Qué es un IDE? - Explicación de los entornos de desarrollo integrado - AWS*. (s. f.). Amazon Web Services, Inc. Recuperado 7 de octubre de 2023, de <https://aws.amazon.com/es/what-is/ide/>
35. *Documentation for Visual Studio Code*. (s. f.). Recuperado 3 de abril de 2023, de <https://code.visualstudio.com/docs>
36. Böck, H. (2009). *The Definitive Guide to NetBeans™ Platform*. Apress. <https://doi.org/10.1007/978-1-4302-2418-1>
37. *Que es un servidor WEB? - Aprende desarrollo web | MDN*. (2023, agosto 2). https://developer.mozilla.org/es/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server
38. School, E. B. (s. f.). *¿Qué es Apache? | Euroinnova*. Euroinnova Business School. Recuperado 8 de octubre de 2023, de <https://www.euroinnova.edu.es/blog/que-es-apache>

39. Benito, M. (2019, mayo 14). *¿Qué es un Framework en Programación?* FP Online. <https://fp.uoc.fje.edu/blog/que-es-un-framework-en-programacion/>
40. *Spring Boot Reference Documentation*. (s. f.). Recuperado 3 de abril de 2023, de <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
41. contributors, M. O., Jacob Thornton, and Bootstrap. (s. f.). *Get started with Bootstrap*. Recuperado 3 de abril de 2023, de <https://getbootstrap.com/docs/5.2/getting-started/introduction/>
42. *Metodologías de desarrollo de software | Universitat Carlemany*. (2022, septiembre 12). UCMA. <https://www.universitatcarlemany.com/actualidad/blog/metodologias-de-desarrollo-de-software/>
43. Pressman, R. S. (2010). *Ingeniería del Software. Un Enfoque Practico*.
44. Sommerville, I. (2011). *Software engineering* (9th ed). Pearson.
45. Monagas Torrecilla, A. D. (2022). *Sistema para la gestión de la prenomina del Departamento de Tecnología de la Facultad 4* [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 4.]. <https://repositorio.uci.cu/jspui/handle/123456789/10601>
46. Castillo Chávez, A. (2016). *Módulo para la gestión del plan anual de eventos y del catálogo de revistas científicas* [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 1]. <https://repositorio.uci.cu/jspui/handle/123456789/7581>
47. *Requerimientos Funcionales y No Funcionales | PDF | Usabilidad | Ingeniería de software*. (s. f.). Recuperado 12 de octubre de 2023, de <https://es.scribd.com/document/412330851/Requerimientos-funcionales-y-no-funcionales>
48. Fong Rios, V. A., González Rodríguez, R. A., & Díaz Barrios, L. F. (2014). *Sistema de gestión de información para los resultados de las pruebas de eficiencia física en la Universidad de las Ciencias Informáticas*. <https://repositorio.uci.cu/jspui/handle/ident/8122>
49. Armas Pérez, A. (2020). *Herramienta de configuración para análisis y entrenamiento de grandes volúmenes de datos digitales* [bachelorThesis, Universidad de las Ciencias Informáticas. FACULTAD 1]. <https://repositorio.uci.cu/jspui/handle/123456789/10343>
50. Salabarría Cárdenas, F. L. (2022). *Repositorio para la gestión de recursos educativos* [bachelorThesis, Universidad de las Ciencias Informáticas. FACULTAD 6 . Facultad de Ciencias y Tecnologías Computacionales (CITEC)]. <https://repositorio.uci.cu/jspui/handle/123456789/10502>

51. *MVC - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web* / MDN. (2022, diciembre 5). <https://developer.mozilla.org/es/docs/Glossary/MVC>
52. *17. Web MVC framework*. (s. f.). Recuperado 3 de noviembre de 2023, de <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>
53. Ortiz Jova, S. D. (2022). *Sistema de Simulación de Penales de Futbol* [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 4]. <https://repositorio.uci.cu/jspui/handle/123456789/10604>
54. *¿Qué es el modelado de datos?* / IBM. (s. f.). Recuperado 24 de octubre de 2023, de <https://www.ibm.com/es-es/topics/data-modeling>
55. Massengo Nekaka, P. (2020). *Sistema de Gestión de Estudiantes para el Sector de Apoyo Estudiantil de Angola en Cuba* [bachelorThesis, Universidad de las Ciencias Informáticas. Facultad 1]. <https://repositorio.uci.cu/jspui/handle/123456789/10357>
56. *¿Qué es la prueba de software y cómo funciona?* / IBM. (s. f.). Recuperado 24 de octubre de 2023, de <https://www.ibm.com/es-es/topics/software-testing>
57. *Datos fundamentales de las pruebas unitarias—Visual Studio (Windows)*. (2023, mayo 8). <https://learn.microsoft.com/es-es/visualstudio/test/unit-test-basics?view=vs-2022>
58. Ríos, Jimmy Rolando Molina & María de las Nieves Pedreira Souto. (2019). *“SWIRL”, metodología para el diseño y desarrollo de aplicaciones web*.

ANEXOS

Entrevista:

Preguntas realizadas a la Dr. C. Yamilis Fernández Pérez:

- ¿Cuál es el objetivo principal del programa de formación doctoral?
- ¿Cuáles son los tipos de datos e información que se manejan actualmente para los doctorandos?
- ¿Cómo se almacenan y organizan actualmente los datos de los doctorandos?
- ¿Cuáles son los desafíos o dificultades que enfrenta actualmente al gestionar la información de los doctorandos?
- ¿Qué funcionalidades o características específicas le gustaría tener en el sistema de gestión de información de doctorandos?

Historias de Usuarios:

Tabla 21: HU 4

Historia de Usuario	
Número: 4	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Publicación	
Usuario: Usuario	Iteración Asignada: 2
Prioridad en negocio: Media	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear una nueva publicación, con los campos de enlace, título datos generales, doi y si fue publicado, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar las publicaciones.	
Observación:	

Tabla 22: HU 5

Historia de Usuario	
Número: 5	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Graduados	
Usuario: Usuario	Iteración Asignada: 2
Prioridad en negocio: Media	Puntos Estimados: 1

Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear un nuevo graduado, con los campos de nombre, apellidos, carnet de identidad, sexo, centro, plantilla, fecha de defensa, facultad y área de conocimiento, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar los graduados.	
Observación: Depende de que haya área del conocimiento y área registrados en el sistema.	

Tabla 23: HU 6

Historia de Usuario	
Número: 6	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Programa	
Usuario: Usuario	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear un nuevo programa, con los campos nombre, evento, forma de organización, desarrollo local, a distancia, desarrollo municipal, área y área de conocimiento, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar los programas.	
Observación: Depende de que haya área del conocimiento y área registrados en el sistema.	

Tabla 24: HU 7

Historia de Usuario	
Número: 7	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Autenticar Usuario	
Usuario: Usuario	Iteración Asignada: 2
Prioridad en negocio: Media	Puntos Estimados: 2
Riesgo en desarrollo: N/A	Puntos Reales: 2
Descripción: El sistema debe permitir al usuario acceder al sistema al introducir su usuario y contraseña.	
Observación: Depende de que haya usuarios registrados en el sistema.	

Tabla 25: HU 8

Historia de Usuario	
Número: 8	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Usuario	
Usuario: Usuario	Iteración Asignada: 2
Prioridad en negocio: Media	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear un nuevo usuario, con los campos nombre y contraseña, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar los usuarios.	
Observación:	

Tabla 26: HU 9

Historia de Usuario	
Número: 9	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Solicitante	
Usuario: Usuario	Iteración Asignada: 3
Prioridad en negocio: Baja	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear un nuevo solicitante, con los campos de nombre, apellidos, carnet de identidad, sexo, centro, plantilla y programa, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar los solicitantes.	
Observación: Depende de que haya programas registrados en el sistema.	

Tabla 27: HU 10

Historia de Usuario	
Número: 10	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Área del Conocimiento	
Usuario: Usuario	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1

Descripción: El sistema debe ofrecer la opción de crear una nueva área del conocimiento, con el campo nombre, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar las áreas del conocimiento.
Observación:

Tabla 28: HU 11

Historia de Usuario	
Número: 11	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Área	
Usuario: Usuario	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear una nueva área, con el campo nombre, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar las áreas.	
Observación:	

Tabla 29: HU 12

Historia de Usuario	
Número: 12	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Centro	
Usuario: Usuario	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear un nuevo centro, con los campos de nombre y organismo, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar los centros.	
Observación:	

Tabla 30: HU 13

Historia de Usuario	
Número: 13	Modificación de HU Número: Ninguna

Nombre de Historia de Usuario: Gestionar Línea de Investigación	
Usuario: Usuario	Iteración Asignada: 3
Prioridad en negocio: Baja	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear una nueva línea de investigación, con el campo nombre, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar las líneas de investigación.	
Observación:	

Tabla 31: HU 14

Historia de Usuario	
Número: 14	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Gestionar Sector Estratégico	
Usuario: Usuario	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe ofrecer la opción de crear un nuevo sector estratégico, con el campo nombre, así como brindar la posibilidad de poder modificar, buscar, eliminar y consultar los sectores estratégico.	
Observación:	

Tabla 32: HU 15

Historia de Usuario	
Número: 15	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Cerrar Sesión	
Usuario: Usuario	Iteración Asignada: 2
Prioridad en negocio: Media	Puntos Estimados: 1
Riesgo en desarrollo: N/A	Puntos Reales: 1
Descripción: El sistema debe permitir al usuario cerrar la sesión y salir del sistema al seleccionar la opción de cerrar sesión.	
Observación: Depende de que haya usuarios autenticado en el sistema.	

Tabla 33: HU 16

Historia de Usuario	
Número: 16	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Mostrar Estadísticas	
Usuario: Usuario	Iteración Asignada: 3
Prioridad en negocio: Baja	Puntos Estimados: 2
Riesgo en desarrollo: N/A	Puntos Reales: 2
Descripción: El sistema debe permitir al usuario visualizar las estadísticas obtenidas de los datos en el sistema.	
Observación:	

Tabla 34: HU 17

Historia de Usuario	
Número: 17	Modificación de HU Número: Ninguna
Nombre de Historia de Usuario: Crear reportes	
Usuario: Usuario	Iteración Asignada: 3
Prioridad en negocio: Baja	Puntos Estimados: 2
Riesgo en desarrollo: N/A	Puntos Reales: 2
Descripción: El sistema debe permitir al usuario crear un reporte.	
Observación:	

Tarjetas CRC:

Tabla 35: Tarjeta CRC "graduados"

Tarjeta CRC	
Clase: graduados	
Responsabilidad:	Colaboración:
-registrar los graduados	-persona
-modificar los graduados	-doctorando
-eliminar los graduados	-area
-buscar los graduados	-areadelconocimiento
-mostrar el listado de los graduados	

Tabla 36: Tarjeta CRC “programa”

Tarjeta CRC	
Clase: programa	
Responsabilidad:	Colaboración:
-registrar los programas -modificar los programas -eliminar los programas -buscar los programas -mostrar el listado de los programas	-area -areadelconocimiento

Tabla 37: Tarjeta CRC “usuario”

Tarjeta CRC	
Clase: usuario	
Responsabilidad:	Colaboración:
-registrar los usuarios -modificar los usuarios -eliminar los usuarios -buscar los usuarios -mostrar el listado de los usuarios	-persona

Tabla 38: Tarjeta CRC “solicitante”

Tarjeta CRC	
Clase: solicitante	
Responsabilidad:	Colaboración:
-registrar los solicitantes -modificar los solicitantes -eliminar los solicitantes -buscar los solicitantes -mostrar el listado de los solicitantes	-persona -programa

Tabla 39: Tarjeta CRC “areadelconocimiento”

Tarjeta CRC

Clase: areadelconocimiento	
Responsabilidad:	Colaboración:
<ul style="list-style-type: none"> -registrar las áreas del conocimiento -modificar las áreas del conocimiento -eliminar las áreas del conocimiento -buscar las áreas del conocimiento -mostrar el listado de las áreas del conocimiento 	

Tabla 40: Tarjeta CRC "area"

Tarjeta CRC	
Clase: area	
Responsabilidad:	Colaboración:
<ul style="list-style-type: none"> -registrar las áreas -modificar las áreas -eliminar las áreas -buscar las áreas -mostrar el listado de las áreas 	

Tabla 41: Tarjeta CRC "centro"

Tarjeta CRC	
Clase: centro	
Responsabilidad:	Colaboración:
<ul style="list-style-type: none"> -registrar los centros -modificar los centros -eliminar los centros -buscar los centros -mostrar el listado de los centros 	

Tabla 42: Tarjeta CRC "lineainvest"

Tarjeta CRC	
Clase: lineainvest	

Responsabilidad:	Colaboración:
<ul style="list-style-type: none"> -registrar las líneas de investigación -modificar las líneas de investigación -eliminar las líneas de investigación -buscar las líneas de investigación -mostrar el listado de las líneas de investigación 	

Tabla 43: Tarjeta CRC “sectorest”

Tarjeta CRC	
Clase: sectorest	
Responsabilidad:	Colaboración:
<ul style="list-style-type: none"> -registrar los sectores estratégicos -modificar los sectores estratégicos -eliminar los sectores estratégicos -buscar los sectores estratégicos -mostrar el listado de los sectores estratégicos 	

Tabla 44: Tarjeta CRC “persona”

Tarjeta CRC	
Clase: persona	
Responsabilidad:	Colaboración:
<ul style="list-style-type: none"> -registrar las personas -modificar las personas -eliminar las personas -buscar las personas 	

Tabla 45: Tarjeta CRC “indexado”

Tarjeta CRC	
Clase: indexado	
Responsabilidad:	Colaboración:

-buscar los indexados	
-----------------------	--

Tabla 46: Tarjeta CRC "pais"

Tarjeta CRC	
Clase: país	
Responsabilidad:	Colaboración:
-buscar los países	

Tabla 47: Tarjeta CRC "seguridad"

Tarjeta CRC	
Clase: seguridad	
Responsabilidad:	Colaboración:
-autenticar los usuarios en el sistema -cerrar la sesión de los usuarios	-usuarios

Tabla 48: Tarjeta CRC "estadísticas"

Tarjeta CRC	
Clase: estadísticas	
Responsabilidad:	Colaboración:
-Calcular las estadísticas -Mostrar las estadísticas -Crear reporte	-doctorando -doctor -solicitante -graduados -tutor

Tareas de Ingeniería:

Tabla 49: Tarea de ingeniería 8

Tarea de ingeniería	
Número de tarea:8	Número de Historia de Usuario: 2
Nombre de la tarea: Gestionar Doctor	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	

Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de los doctores.

Tabla 50: Tarea de ingeniería 9

Tarea de ingeniería	
Número de tarea:9	Número de Historia de Usuario: 3
Nombre de la tarea: Gestión de Tutoría	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la asignación de tutores a los doctorandos.	

Tabla 51: Tarea de ingeniería 10

Tarea de ingeniería	
Número de tarea:10	Número de Historia de Usuario: 6
Nombre de la tarea: Gestionar Programa	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de los graduados.	

Tabla 52: Tarea de ingeniería 11

Tarea de ingeniería	
Número de tarea:11	Número de Historia de Usuario: 4
Nombre de la tarea: Gestionar Publicación	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de las publicaciones.	

Tabla 53: Tarea de ingeniería 12

Tarea de ingeniería	
Número de tarea:12	Número de Historia de Usuario: 5

Nombre de la tarea: Gestionar Graduados	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de los graduados.	

Tabla 54: Tarea de ingeniería 13

Tarea de ingeniería	
Número de tarea:13	Número de Historia de Usuario: 8
Nombre de la tarea: Gestionar Usuario	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de los usuarios.	

Tabla 55: Tarea de ingeniería 14

Tarea de ingeniería	
Número de tarea:14	Número de Historia de Usuario: 7,15
Nombre de la tarea: Seguridad del Sistema	
Tipo de tarea: Desarrollo	Puntos Estimados: 3
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la autenticación de los usuarios en el sistema y poder cerrar sesión.	

Tabla 56: Tarea de ingeniería 15

Tarea de ingeniería	
Número de tarea:15	Número de Historia de Usuario: 9
Nombre de la tarea: Gestionar Solicitante	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de los solicitantes.	

Tabla 57: Tarea de ingeniería 16

Tarea de ingeniería	
Número de tarea:16	Número de Historia de Usuario: 13
Nombre de la tarea: Gestionar Línea de Investigación	
Tipo de tarea: Desarrollo	Puntos Estimados: 1
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para la gestión de la información de las líneas de investigación.	

Tabla 58: Tarea de ingeniería 17

Tarea de ingeniería	
Número de tarea:17	Número de Historia de Usuario: 16
Nombre de la tarea: Mostrar Estadísticas	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces necesarias para mostrar las estadísticas.	

Tabla 59: Tarea de ingeniería 18

Tarea de ingeniería	
Número de tarea:18	Número de Historia de Usuario: 17
Nombre de la tarea: Crear reporte	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Programador responsable: Yilber Serrano	
Descripción: Desarrollar los métodos y las interfaces para crear reporte.	

Pruebas de unitarias:

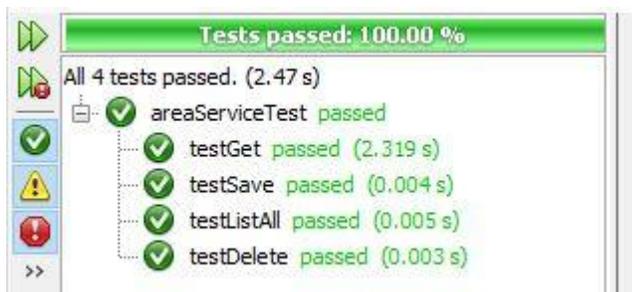


Ilustración 5: Resultados de la prueba unitaria a areaService.



Ilustración 6: Resultados de la prueba unitaria a areadeconocimientoService.

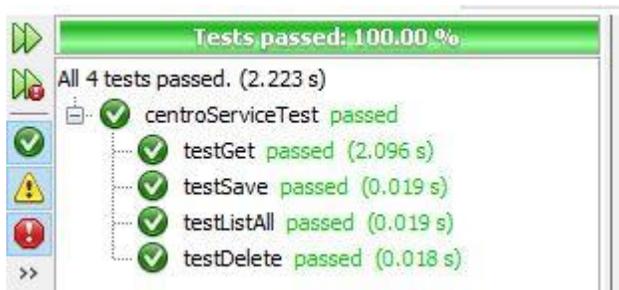


Ilustración 7: Resultados de la prueba unitaria a centroService.

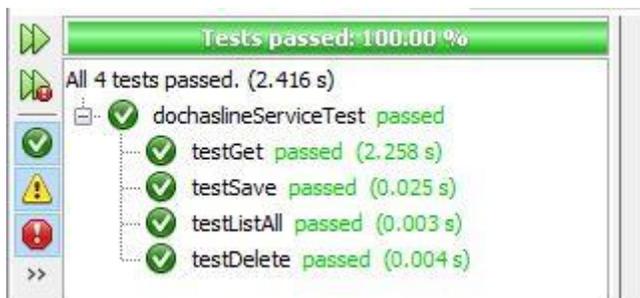


Ilustración 8: Resultados de la prueba unitaria a dochaslineService.

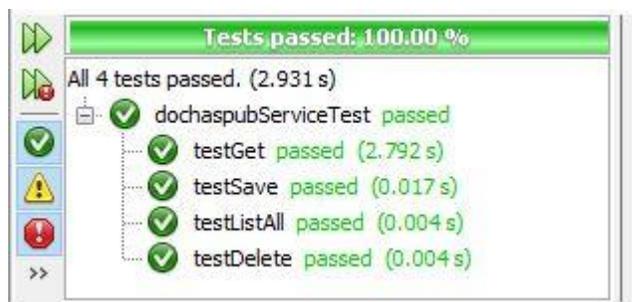


Ilustración 9: Resultados de la prueba unitaria a dochaspubService.

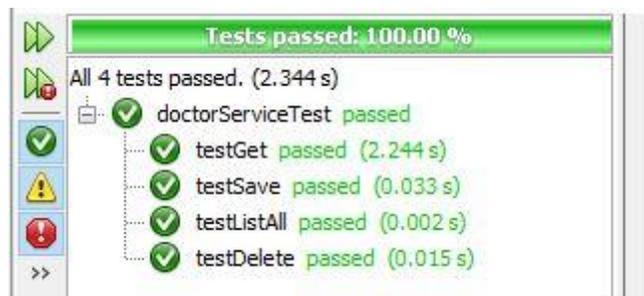


Ilustración 10: Resultados de la prueba unitaria a doctorService.

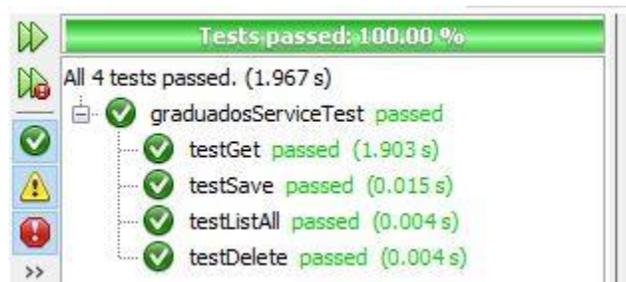


Ilustración 11: Resultados de la prueba unitaria a graduadoService.



Ilustración 12: Resultados de la prueba unitaria a indexadoService.



Ilustración 13: Resultados de la prueba unitaria a indhaspubService.

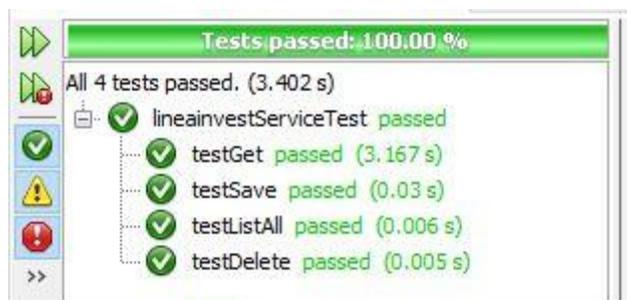


Ilustración 14: Resultados de la prueba unitaria a lineainvestService.

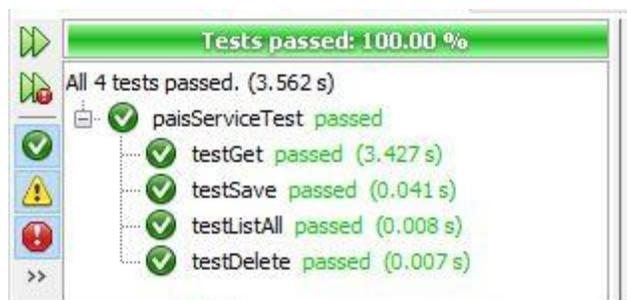


Ilustración 15: Resultados de la prueba unitaria a paisService.



Ilustración 16: Resultados de la prueba unitaria a personaService.

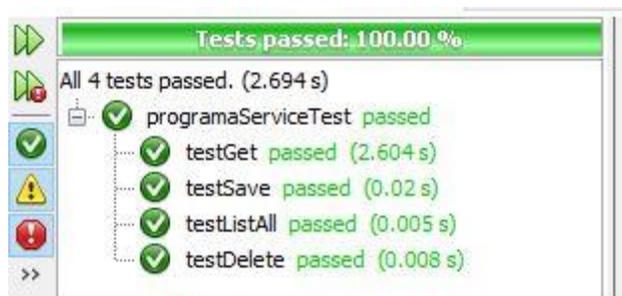


Ilustración 17: Resultados de la prueba unitaria a programaService.

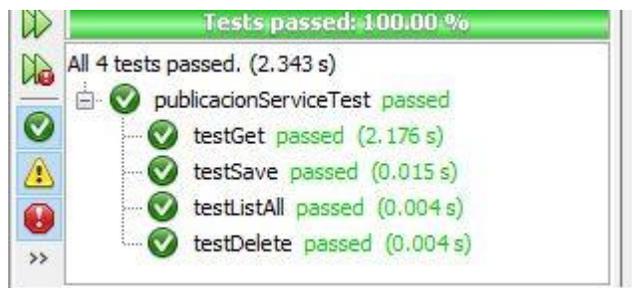


Ilustración 18: Resultados de la prueba unitaria a publicacionService.



Ilustración 19: Resultados de la prueba unitaria a sectorestService.

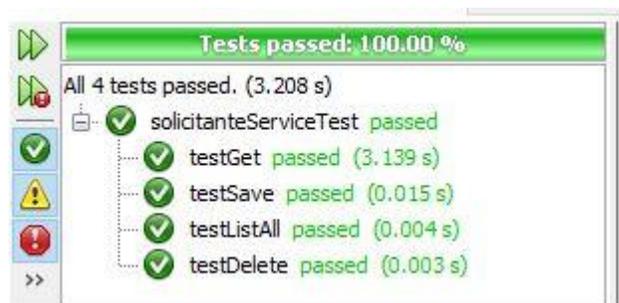


Ilustración 20: Resultados de la prueba unitaria a solicitanteService.



Ilustración 21: Resultados de la prueba unitaria a tutorService.



Ilustración 22: Resultados de la prueba unitaria a usuarioService.

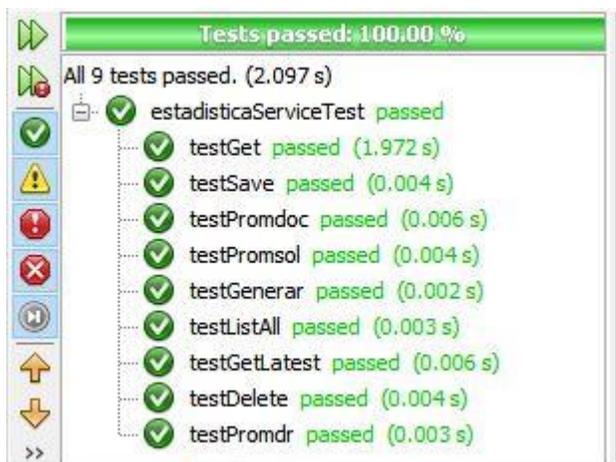


Ilustración 23: Resultados de la prueba unitaria a estadisticaService.

Casos de pruebas de aceptación:

Tabla 60: Caso de Prueba de Aceptación HU2_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU2_P1	Nombre de Historia de Usuario: Gestionar Doctor
Nombre: Registrar Doctor	
Descripción: Prueba la funcionalidad de registrar un doctor.	

Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de doctores.
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar
Resultados esperados: El sistema registre un nuevo doctor si los datos fueron correctos, en caso contrario muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 61: Caso de Prueba de Aceptación HU2_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU2_P2	Nombre de Historia de Usuario: Gestionar Doctor
Nombre: Modificar Doctor	
Descripción: Prueba la funcionalidad de modificar un doctor.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de doctores.	
Pasos de Ejecución: El usuario busca el doctor que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica al doctor si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 62: Caso de Prueba de Aceptación HU2_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU2_P3	Nombre de Historia de Usuario: Gestionar Doctor
Nombre: Eliminar Doctor	
Descripción: Prueba la funcionalidad de eliminar un doctor.	

Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de doctores.
Pasos de Ejecución: El usuario busca el doctor que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.
Resultados esperados: El sistema elimina un doctor del sistema.
Evaluación de la prueba: Satisfactoria

Tabla 63: Caso de Prueba de Aceptación HU2_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU2_P4	Nombre de Historia de Usuario: Gestionar Doctor
Nombre: Listar Doctor	
Descripción: Prueba la funcionalidad de listar los doctores.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Doctores.	
Resultados esperados: El sistema carga una página con el listado de doctores.	
Evaluación de la prueba: Satisfactoria	

Tabla 64: Caso de Prueba de Aceptación HU3_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3_P1	Nombre de Historia de Usuario: Gestión de Tutoría
Nombre: Asignar Tutor	
Descripción: Prueba la funcionalidad de asignar un tutor.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de tutores.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario selecciona un doctorando y el doctor que será su tutor, el usuario selecciona el botón Guardar.	

Resultados esperados: El sistema asigna un tutor al doctorando si los datos fueron correctos, en caso contrario muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 65: Caso de Prueba de Aceptación HU3_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3_P2	Nombre de Historia de Usuario: Gestión de Tutoría
Nombre: Quitar Tutor	
Descripción: Prueba la funcionalidad de quitar un tutor	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de doctorandos.	
Pasos de Ejecución: El usuario busca el doctorando que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina un doctorando del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 66: Caso de Prueba de Aceptación HU3_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3_P3	Nombre de Historia de Usuario: Gestión de Tutoría
Nombre: Listar Tutores	
Descripción: Prueba la funcionalidad de listar los tutores.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Tutores.	
Resultados esperados: El sistema carga una página con el listado de tutores.	
Evaluación de la prueba: Satisfactoria	

Tabla 67: Caso de Prueba de Aceptación HU4_P1

Caso de Prueba de Aceptación

Código Caso de Prueba: HU4_P1	Nombre de Historia de Usuario: Gestionar Publicación
Nombre: Registrar Publicación	
Descripción: Prueba la funcionalidad de registrar una publicación.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de publicaciones.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema registre una nueva publicación si los datos fueron correctos, en caso contrario muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 68: Caso de Prueba de Aceptación HU4_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU4_P2	Nombre de Historia de Usuario: Gestionar Publicación
Nombre: Modificar Publicación	
Descripción: Prueba la funcionalidad de modificar una publicación.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de publicaciones.	
Pasos de Ejecución: El usuario busca la publicación que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica la publicación si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 69: Caso de Prueba de Aceptación HU4_P3

Caso de Prueba de Aceptación

Código Caso de Prueba: HU4_P3	Nombre de Historia de Usuario: Gestionar Publicación
Nombre: Eliminar Publicación	
Descripción: Prueba la funcionalidad de eliminar una publicación.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de publicaciones.	
Pasos de Ejecución: El usuario busca la publicación que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina una publicación del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 70: Caso de Prueba de Aceptación HU4_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU4_P4	Nombre de Historia de Usuario: Gestionar Publicación
Nombre: Listar Publicación	
Descripción: Prueba la funcionalidad de listar las publicaciones.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Publicaciones.	
Resultados esperados: El sistema carga una página con el listado de publicaciones.	
Evaluación de la prueba: Satisfactoria	

Tabla 71: Caso de Prueba de Aceptación HU5_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU5_P1	Nombre de Historia de Usuario: Gestionar Graduados
Nombre: Registrar Graduado	
Descripción: Prueba la funcionalidad de registrar un graduado.	

Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de Graduados.
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.
Resultados esperados: El sistema registre un nuevo graduado si los datos fueron correctos, en caso contrario muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 72: Caso de Prueba de Aceptación HU5_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU5_P2	Nombre de Historia de Usuario: Gestionar Graduados
Nombre: Modificar Graduado	
Descripción: Prueba la funcionalidad de modificar un graduado.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de graduados.	
Pasos de Ejecución: El usuario busca el graduado que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica al graduado si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 73: Caso de Prueba de Aceptación HU5_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU5_P3	Nombre de Historia de Usuario: Gestionar Graduados
Nombre: Eliminar Graduado	
Descripción: Prueba la funcionalidad de eliminar un graduado.	

Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de graduados.
Pasos de Ejecución: El usuario busca el graduado que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.
Resultados esperados: El sistema elimina un graduado del sistema.
Evaluación de la prueba: Satisfactoria

Tabla 74: Caso de Prueba de Aceptación HU5_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU5_P4	Nombre de Historia de Usuario: Gestionar Graduados
Nombre: Listar Graduados	
Descripción: Prueba la funcionalidad de listar los graduados.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Graduados.	
Resultados esperados: El sistema carga una página con el listado de graduados.	
Evaluación de la prueba: Satisfactoria	

Tabla 75: Caso de Prueba de Aceptación HU6_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU6_P1	Nombre de Historia de Usuario: Gestionar Programa
Nombre: Registrar Programa	
Descripción: Prueba la funcionalidad de registrar un programa.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de programas.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	

Resultados esperados: El sistema registre un nuevo programa si los datos fueron correctos, en caso contrario muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 76: Caso de Prueba de Aceptación HU6_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU6_P2	Nombre de Historia de Usuario: Gestionar Programa
Nombre: Modificar Programa	
Descripción: Prueba la funcionalidad de modificar un programa.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de programas.	
Pasos de Ejecución: El usuario busca el programa que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica al programa si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 77: Caso de Prueba de Aceptación HU6_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU6_P3	Nombre de Historia de Usuario: Gestionar Programa
Nombre: Eliminar Programa	
Descripción: Prueba la funcionalidad de eliminar un programa.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de programas.	
Pasos de Ejecución: El usuario busca el programa que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina un programa del sistema.	

Evaluación de la prueba: Satisfactoria

Tabla 78: Caso de Prueba de Aceptación HU6_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU6_P4	Nombre de Historia de Usuario: Gestionar Programa
Nombre: Listar Programa	
Descripción: Prueba la funcionalidad de listar los programas.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Programas.	
Resultados esperados: El sistema carga una página con el listado de programas.	
Evaluación de la prueba: Satisfactoria	

Tabla 79: Caso de Prueba de Aceptación HU7_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU7_P1	Nombre de Historia de Usuario: Autenticar Usuario
Nombre: Autenticar Usuario	
Descripción: Prueba la funcionalidad de entrar al sistema mediante el usuario.	
Condiciones de Ejecución: El usuario debe estar registrado en el sistema.	
Pasos de Ejecución: El usuario accede a la URL del sistema a través de un navegador web, el sistema muestra el formulario de inicio de sesión, el usuario introduce el nombre y contraseña y selecciona la opción Entrar.	
Resultados esperados: El sistema muestra la página principal en caso de que estén los datos correctos sino muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 80: Caso de Prueba de Aceptación HU8_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU8_P1	Nombre de Historia de Usuario: Gestionar Usuario

Nombre: Registrar Usuario
Descripción: Prueba la funcionalidad de registrar un usuario.
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuario.
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.
Resultados esperados: El sistema registre un nuevo usuario si los datos fueron correctos, en caso contrario muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 81: Caso de Prueba de Aceptación HU8_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU8_P2	Nombre de Historia de Usuario: Gestionar Usuario
Nombre: Modificar Usuario	
Descripción: Prueba la funcionalidad de modificar un usuario.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuarios.	
Pasos de Ejecución: El usuario busca el usuario que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica al usuario si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 82: Caso de Prueba de Aceptación HU8_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU8_P3	Nombre de Historia de Usuario: Gestionar Usuario
Nombre: Eliminar Usuario	

Descripción: Prueba la funcionalidad de eliminar un usuario.
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de usuarios.
Pasos de Ejecución: El usuario busca el usuario que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.
Resultados esperados: El sistema elimina un usuario del sistema.
Evaluación de la prueba: Satisfactoria

Tabla 83: Caso de Prueba de Aceptación HU8_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU8_P4	Nombre de Historia de Usuario: Gestionar Usuarios
Nombre: Listar Usuario	
Descripción: Prueba la funcionalidad de listar los usuarios.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Usuario.	
Resultados esperados: El sistema carga una página con el listado de usuarios.	
Evaluación de la prueba: Satisfactoria	

Tabla 84: Caso de Prueba de Aceptación HU9_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU9_P1	Nombre de Historia de Usuario: Gestionar Solicitante
Nombre: Registrar Solicitante	
Descripción: Prueba la funcionalidad de registrar un solicitante.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de solicitantes.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	

Resultados esperados: El sistema registre un solicitante si los datos fueron correctos, en caso contrario muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 85: Caso de Prueba de Aceptación HU9_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU9_P2	Nombre de Historia de Usuario: Gestionar Solicitante
Nombre: Modificar Solicitante	
Descripción: Prueba la funcionalidad de modificar un solicitante.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de solicitantes.	
Pasos de Ejecución: El usuario busca el solicitante que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica al solicitante si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 86: Caso de Prueba de Aceptación HU9_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU9_P3	Nombre de Historia de Usuario: Gestionar Solicitante
Nombre: Eliminar Solicitante	
Descripción: Prueba la funcionalidad de eliminar un solicitante.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de solicitantes.	
Pasos de Ejecución: El usuario busca el solicitante que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina un solicitante del sistema.	

Evaluación de la prueba: Satisfactoria

Tabla 87: Caso de Prueba de Aceptación HU9_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU9_P4	Nombre de Historia de Usuario: Gestionar Solicitante
Nombre: Listar Solicitante	
Descripción: Prueba la funcionalidad de listar los solicitantes.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Solicitantes.	
Resultados esperados: El sistema carga una página con el listado de solicitantes.	
Evaluación de la prueba: Satisfactoria	

Tabla 88: Caso de Prueba de Aceptación HU10_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU10_P1	Nombre de Historia de Usuario: Gestionar Área de Conocimiento
Nombre: Registrar Área de Conocimiento	
Descripción: Prueba la funcionalidad de registrar un área del conocimiento.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de áreas del conocimiento.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema registre una nueva área del conocimiento si los datos fueron correctos, en caso contrario muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 89: Caso de Prueba de Aceptación HU10_P2

Caso de Prueba de Aceptación

Código Caso de Prueba: HU10_P2	Nombre de Historia de Usuario: Gestionar Área de Conocimiento
Nombre: Modificar Área de Conocimiento	
Descripción: Prueba la funcionalidad de modificar un área del conocimiento.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de áreas del conocimiento.	
Pasos de Ejecución: El usuario busca el área del conocimiento que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica al área del conocimiento si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 90: Caso de Prueba de Aceptación HU10_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU10_P3	Nombre de Historia de Usuario: Gestionar Área de Conocimiento
Nombre: Eliminar Área de Conocimiento	
Descripción: Prueba la funcionalidad de eliminar un área del conocimiento.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de áreas del conocimiento.	
Pasos de Ejecución: El usuario busca el área del conocimiento que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina un área del conocimiento del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 91: Caso de Prueba de Aceptación HU10_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU10_P4	Nombre de Historia de Usuario: Gestionar Área de Conocimiento

Nombre: Listar Área de Conocimiento
Descripción: Prueba la funcionalidad de listar las áreas del conocimiento.
Condiciones de Ejecución: El usuario debe estar autenticado.
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Área de Conocimiento.
Resultados esperados: El sistema carga una página con el listado de áreas del conocimiento.
Evaluación de la prueba: Satisfactoria

Tabla 92: Caso de Prueba de Aceptación HU11_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU11_P1	Nombre de Historia de Usuario: Gestionar Área
Nombre: Registrar Área	
Descripción: Prueba la funcionalidad de registrar un área.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de áreas.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema registre una nueva área si los datos fueron correctos, en caso contrario muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 93: Caso de Prueba de Aceptación HU11_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU11_P2	Nombre de Historia de Usuario: Gestionar Área
Nombre: Modificar Área	
Descripción: Prueba la funcionalidad de modificar un área.	

Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de áreas.
Pasos de Ejecución: El usuario busca el área que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.
Resultados esperados: El sistema modifica al área si los datos fueron correctos, en otro caso muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 94: Caso de Prueba de Aceptación HU11_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU11_P3	Nombre de Historia de Usuario: Gestionar Área
Nombre: Eliminar Área	
Descripción: Prueba la funcionalidad de eliminar un área.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de áreas.	
Pasos de Ejecución: El usuario busca el área que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina un área del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 95: Caso de Prueba de Aceptación HU11_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU11_P4	Nombre de Historia de Usuario: Gestionar Área de Conocimiento
Nombre: Listar Área de Conocimiento	
Descripción: Prueba la funcionalidad de listar las áreas del conocimiento.	
Condiciones de Ejecución: El usuario debe estar autenticado.	

Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Área de Conocimiento.
Resultados esperados: El sistema carga una página con el listado de áreas del conocimiento.
Evaluación de la prueba: Satisfactoria

Tabla 96: Caso de Prueba de Aceptación HU12_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU12_P1	Nombre de Historia de Usuario: Gestionar Centro
Nombre: Registrar Centro	
Descripción: Prueba la funcionalidad de registrar un centro.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de centros.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema registre un nuevo centro si los datos fueron correctos, en caso contrario muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 97: Caso de Prueba de Aceptación HU12_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU12_P2	Nombre de Historia de Usuario: Gestionar Centro
Nombre: Modificar Centro	
Descripción: Prueba la funcionalidad de modificar un centro.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de centros.	

Pasos de Ejecución: El usuario busca el centro que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.
Resultados esperados: El sistema modifica el centro si los datos fueron correctos, en otro caso muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 98: Caso de Prueba de Aceptación HU12_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU12_P3	Nombre de Historia de Usuario: Gestionar Centro
Nombre: Eliminar Centro	
Descripción: Prueba la funcionalidad de eliminar un centro.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de centros.	
Pasos de Ejecución: El usuario busca el centro que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina un centro del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 99: Caso de Prueba de Aceptación HU12_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU12_P4	Nombre de Historia de Usuario: Gestionar Centro
Nombre: Listar Centro	
Descripción: Prueba la funcionalidad de listar los centros.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Centros.	
Resultados esperados: El sistema carga una página con el listado de los centros.	
Evaluación de la prueba: Satisfactoria	

Tabla 100: Caso de Prueba de Aceptación HU13_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU13_P1	Nombre de Historia de Usuario: Gestionar Líneas de Investigación
Nombre: Registrar Línea de Investigación	
Descripción: Prueba la funcionalidad de registrar un doctor.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de líneas de investigación.	
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema registre una nueva línea de investigación si los datos fueron correctos, en caso contrario muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 101: Caso de Prueba de Aceptación HU13_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU13_P2	Nombre de Historia de Usuario: Gestionar Líneas de Investigación
Nombre: Modificar Líneas de Investigación	
Descripción: Prueba la funcionalidad de modificar una línea de investigación.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de líneas de investigación.	
Pasos de Ejecución: El usuario busca la línea de investigación que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica la línea de investigación si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 102: Caso de Prueba de Aceptación HU13_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU13_P3	Nombre de Historia de Usuario: Gestionar Líneas de Investigación
Nombre: Eliminar Líneas de Investigación	
Descripción: Prueba la funcionalidad de eliminar una línea de investigación.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de líneas de investigación.	
Pasos de Ejecución: El usuario busca la línea de investigación que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.	
Resultados esperados: El sistema elimina una línea de investigación del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 103: Caso de Prueba de Aceptación HU13_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU13_P4	Nombre de Historia de Usuario: Gestionar Líneas de Investigación
Nombre: Listar Líneas de Investigación	
Descripción: Prueba la funcionalidad de listar las líneas de investigación.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Líneas de Investigación.	
Resultados esperados: El sistema carga una página con el listado de las líneas de investigación.	
Evaluación de la prueba: Satisfactoria	

Tabla 104: Caso de Prueba de Aceptación HU14_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU14_P1	Nombre de Historia de Usuario: Gestionar Sector Estratégico

Nombre: Registrar Sector Estratégico
Descripción: Prueba la funcionalidad de registrar un sector estratégico.
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de sectores estratégico.
Pasos de Ejecución: El usuario selecciona el botón de Agregar, el sistema muestra un formulario, el usuario introduce los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.
Resultados esperados: El sistema registre un nuevo sector estratégico si los datos fueron correctos, en caso contrario muestra un mensaje de error.
Evaluación de la prueba: Satisfactoria

Tabla 105: Caso de Prueba de Aceptación HU14_P2

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU14_P2	Nombre de Historia de Usuario: Gestionar Sector Estratégico
Nombre: Modificar Sector Estratégico	
Descripción: Prueba la funcionalidad de modificar un sector estratégico.	
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de sectores estratégico.	
Pasos de Ejecución: El usuario busca el sector estratégico que desea modificar, selecciona la opción de Modificar, el sistema muestra un formulario, el usuario modifica los datos necesarios y verifica que sean correctos, el usuario selecciona el botón Guardar.	
Resultados esperados: El sistema modifica al sector estratégico si los datos fueron correctos, en otro caso muestra un mensaje de error.	
Evaluación de la prueba: Satisfactoria	

Tabla 106: Caso de Prueba de Aceptación HU14_P3

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU14_P3	Nombre de Historia de Usuario: Gestionar Sector Estratégico
Nombre: Eliminar Sector Estratégico	

Descripción: Prueba la funcionalidad de eliminar un sector estratégico.
Condiciones de Ejecución: El usuario debe estar autenticado, debe tener cargado el listado de sectores estratégico.
Pasos de Ejecución: El usuario busca el sector estratégico que desea eliminar, selecciona la opción de eliminar, el sistema muestra un mensaje para verificar la acción, el usuario selecciona el botón aceptar.
Resultados esperados: El sistema elimina un sector estratégico del sistema.
Evaluación de la prueba: Satisfactoria

Tabla 107: Caso de Prueba de Aceptación HU14_P4

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU14_P4	Nombre de Historia de Usuario: Gestionar Sector Estratégico
Nombre: Listar Sector Estratégico	
Descripción: Prueba la funcionalidad de listar los sectores estratégicos.	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Pasos de Ejecución: El usuario selecciona la opción Gestión, selecciona la opción Sector Estratégico.	
Resultados esperados: El sistema carga una página con el listado de sectores estratégico.	
Evaluación de la prueba: Satisfactoria	

Tabla 108: Caso de Prueba de Aceptación HU15_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU15_P1	Nombre de Historia de Usuario: Cerrar Sesión
Nombre: Cerrar Sesión	
Descripción: Prueba la funcionalidad de cerrar sesión.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de Ejecución: El usuario selecciona la opción Usuario luego cerrar sesión.	
Resultados esperados: El sistema muestra la página de login y cierra la sesión.	
Evaluación de la prueba: Satisfactoria	

Tabla 109: Caso de Prueba de Aceptación HU16_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU15_P1	Nombre de Historia de Usuario: Mostrar estadísticas
Nombre: Mostrar estadísticas	
Descripción: Prueba la funcionalidad de mostrar estadísticas.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de Ejecución: El usuario accede a la pantalla principal.	
Resultados esperados: El sistema muestra las gráficas de estadísticas.	
Evaluación de la prueba: Satisfactoria	

Tabla 110: Caso de Prueba de Aceptación HU17_P1

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU15_P1	Nombre de Historia de Usuario: Crear reporte
Nombre: Crear reporte	
Descripción: Prueba la funcionalidad de crear reporte.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de Ejecución: El usuario accede a la pantalla principal, selecciona la opción de generar reporte.	
Resultados esperados: El sistema descarga un reporte en formato pdf.	
Evaluación de la prueba: Satisfactoria	