



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 1

**SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS
EXTENSIONISTAS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

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

Autor: Dariel Torres Alvarez

Tutores: MSc. Julio César Espronceda Pérez

Ing. Alvaro Alejandro Acosta Ruiz

La Habana, 2022

“El mayor peligro para la mayoría de nosotros no radica en establecer unos objetivos demasiado altos y fracasar, sino en establecer unos objetivos demasiado bajos, y lograrlos”.

Miguel Ángel

Dedicatoria

Este trabajo de diploma se lo dedico principalmente a mi madre, por estar presente y no dejarme rendir en todos los altos y bajos de la carrera, por la crianza y por luchar cada segundo conmigo, se lo dedico a mi abuelito, que me guía, con sus valores brindados, su crianza de caballero y su consejo más humilde y sencillo "estudia", a mi familia por el apoyo incondicional y a mí, por a pesar de las tormentas, no rendirme, y luchar por cumplir este reto.

Agradecimientos

Agradecido especialmente con mi madre y mi familia, por todos los esfuerzos, durante toda la carrera, agradecido con todas esas personas que confiaron en mí, cuando incluso no me encontraba, agradecido con mis tutores Julio y Álvaro, primero por darme la oportunidad y segundo por la confianza brindada, mis colegas de equipo y de aula por el sentimiento de familiaridad, agradecido con las personas maravillosas que me presentó la vida, en este tiempo de carrera universitaria, de todos algo aprendí, profesores que no se rindieron conmigo, personas especiales como Elianne, los chicos del 89201, donde esta historia empezó, las chicas del 56, mi hermano de armas Yoan, cada compañero de mis aulas y apartamento, a todos, gracias.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Dariel Torres Alvarez
Autor

MSc. Julio César Espronceda Pérez
Tutor

Ing. Alvaro Alejandro Acosta Ruiz
Tutor

La gestión de los proyectos extensionistas en la Universidad de las Ciencias Informáticas (UCI) se lleva a cabo por la Dirección de extensión universitaria, lo que permite mejorar el funcionamiento de la universidad, haciendo énfasis tanto en la comunidad intra y extra universitaria. En la actualidad la gestión de estos proyectos no favorece el control de las actividades, acciones y tareas de la extensión universitaria que se planifican, dado que no existe un aprovechamiento de la infraestructura tecnológica de la universidad. Debido a esto, el flujo de información que se realiza provoca que existan retrasos en la entrega de los documentos y no se conoce en qué paso del proceso se encuentra detenida la información, por lo que se ha visto afectada la toma de decisiones, al no contar con toda la información requerida, para su organización. Como solución al problema de la investigación se desarrolló un sistema que contribuye a la gestión de proyectos extensionistas que se desarrollan en la UCI, desde su proceso de concepción, planificación, ejecución y tratamiento de la información generada haciendo uso de informes estadísticos, reportes y gráficas. Para el desarrollo de la solución informática se utilizaron tecnologías y herramientas de software libre.

Palabras clave: extensión universitaria, proyectos extensionistas, sistema informático.

Introducción	1
1 Marco teórico referencial sobre el sistema para la gestión de los proyectos extensionistas de la Universidad de las Ciencias Informáticas	6
1.1 Fundamentos teóricos sobre la gestión de los proyectos extensionistas	6
1.1.1 Extensión universitaria	6
1.1.2 Gestión de la extensión universitaria	7
1.1.3 Proceso	8
1.1.4 Gestión de procesos	9
1.1.5 Dirección de extensión universitaria	9
1.1.6 Proyectos extensionistas	10
1.1.7 Gestión de proyectos extensionistas	11
1.1.8 Tarea extensionista	12
1.1.9 Acciones extensionistas	13
1.1.10 Actividad extensionista	13
1.1.11 Programa extensionista	14
1.2 Análisis de las soluciones existentes	14
1.2.1 Ámbito internacional	14
1.2.2 Ámbito nacional	16
1.2.3 Comparación entre los sistemas de homólogos	18
1.2.4 Consideraciones del estudio de homólogos	18
1.3 Entorno de desarrollo de la propuesta de solución	18
1.3.1 Metodología	18
1.3.2 Herramientas y tecnologías	19
1.3.3 Framework	22
1.3.4 Sistemas gestores de bases de datos (SGBD)	23
1.3.5 Herramienta para control de versiones	23
1.4 Conclusiones del capítulo	24

2	Características del sistema para la gestión de los proyectos extensionistas de la Universidad de las Ciencias Informáticas	25
2.1	Descripción de la propuesta de solución	25
2.2	Especificación de requisitos del sistema	26
2.2.1	Requisitos funcionales	27
2.2.2	Requisitos no funcionales	27
2.3	Descripción de las historias de usuario	28
2.4	Estilo arquitectónico	31
2.4.1	Patrón arquitectónico	31
2.5	Diagrama de clases de diseño	32
2.6	Patrones de diseño	33
2.7	Modelo de datos	35
2.8	Consideraciones parciales	36
3	Implementación y prueba del sistema para la gestión de los proyectos extensionistas de la Universidad de Ciencias Informáticas	38
3.1	Estándar de codificación	38
3.2	Diagrama de despliegue	42
3.3	Pruebas de software	43
3.3.1	Estrategia de pruebas	43
3.3.2	Pruebas unitarias	44
3.3.3	Pruebas funcionales	47
3.4	Resultados de las pruebas	49
3.4.1	Prueba de rendimiento (Carga y Estrés)	49
3.4.2	Pruebas de seguridad	51
3.4.3	Pruebas aceptación	52
3.5	Resultados de la solución implementada	52
3.6	Validación de la hipótesis	54
3.7	Consideraciones parciales	55
	Conclusiones	56
	Recomendaciones	57
	Acrónimos	58
	Referencias bibliográficas	60
	Apéndices	63

A	Anexos	64
A.1	Modelo de entrevista aplicada a especialistas de Extensión universitaria de la UCI	64
A.2	Guía de observación	64
A.3	Resultado de las entrevistas	65
A.4	Historias de Usuario	66

Índice de figuras

1.1	Procesos de la Extensión Universitaria	8
1.2	Diagrama de flujo	11
1.3	Gestión de Proyectos	13
2.1	Diagrama de caso de uso	26
2.2	Patrón arquitectónico	32
2.3	Diagrama de clases	33
2.4	Modelo de datos	35
3.1	Diagrama de despliegue	42
3.2	Diagrama de flujo	49
3.3	Prueba de rendimiento	50
3.4	Prueba de rendimiento	51
3.5	Acunetix Web Vulnerability Scanner Primera iteración	51
3.6	Acunetix Web Vulnerability Scanner segunda iteración	52

Índice de tablas

1	Operacionalización de la variable independiente (Fuente: Elaboración propia).	3
2	Operacionalización de la variable dependiente (Fuente: Elaboración propia).	4
1.1	Comparación de sistemas homólogos. (Fuente: Elaboración propia)	18
2.1	Descripción de los Requisitos Funcionales. (Fuente: Elaboración propia).	27
2.2	Historia de usuario # 1	28
2.3	Historia de usuario # 2	29
2.4	Historia de usuario # 3	30
3.1	Estándares de codificación a utilizar. (Fuente: Elaboración propia).	39
3.2	Estrategia de pruebas. (Fuente: Elaboración propia).	43
3.3	Cálculo de la complejidad ciclomática del método Crear_Proyectos	44
3.4	Caminos del grafo de flujo (Fuente: Elaboración propia).	45
3.5	Caso de Prueba para el camino básico 1 (Fuente: Elaboración propia).	46
3.6	Caso de prueba “Autenticar usuario” (Fuente: Elaboración Propia).	48
3.7	Variables de caso de prueba “Autenticar usuario”(Fuente: Elaboración Propia).	48
3.8	Cuadro Lógico de Iadov (Fuente: Elaboración propia).	53
A.1	Encuesta.	64
A.2	Guía de observación (Fuente: Elaboración propia).	65
A.3	Historia de usuario # 4	66
A.4	Historia de usuario # 5	66
A.5	Historia de usuario # 6	67
A.6	Historia de usuario # 7	68
A.7	Historia de usuario # 8	69
A.8	Historia de usuario # 9	69
A.9	Historia de usuario # 10	70
A.10	Historia de usuario # 11	70
A.11	Historia de usuario # 12	71
A.12	Historia de usuario # 13	71

A.13 Historia de usuario # 14	72
A.14 Historia de usuario # 15	73
A.15 Historia de usuario # 16	73
A.16 Historia de usuario # 17	74
A.17 Historia de usuario # 18	74
A.18 Historia de usuario # 19	75
A.19 Historia de usuario # 20	75
A.20 Historia de usuario # 21	76

La educación constituye hoy en día un factor esencial para el desarrollo social, cultural y económico ampliamente reconocido y un reto para toda la sociedad. La influencia de la revolución científica, tecnológica y productiva que se lleva a cabo en la actualidad, trae consigo constantes transformaciones en los procesos educativos y por lo tanto en la formación sistemática y continua de los recursos humanos, lo que ratifica a la educación como condición necesaria para el desarrollo y para ello, es necesario considerar también la formación cultural y técnica de la población (González Aportela y Batista Mainegra, 2017).

El Ministerio de Educación Superior (MES) es la institución encargada de dirigir metodológicamente la educación superior en Cuba, con el objetivo de aplicar la política educacional en el nivel de la enseñanza superior (MES, 2022). Dentro del proceso extensionista, dirige la promoción del desarrollo cultural en las Instituciones de Educación Superior (IES) para contribuir a la formación integral de la comunidad universitaria y su entorno, con el propósito de elevar el desarrollo cultural de la comunidad universitaria y la población en general, a través del trabajo sociocultural, estimulando la educación permanente, la difusión de la cultura científica y tecnológica, la artística-literaria, la cultura física y el deporte, la educación ambiental, la cultura política, económica, y otras dimensiones de la cultura.

El Programa Nacional de Extensión Universitaria (PNEU) diseñado por el MES constituye un instrumento eficaz para la gestión del proceso extensionista en las universidades. Este programa está concebido de forma tal, que, a partir de la dinámica y las condiciones del entorno, se pueda buscar un equilibrio entre el proceso extensionista en la educación superior cubana y las necesidades que demanda cada una de las realidades objeto de transformación por las universidades. Por lo tanto, gestionar el proceso de extensión universitaria acorde a las necesidades de los nuevos tiempos precisa una relación de interacción permanente con la realidad mediante un contacto reflexivo. Esto genera nuevas formas de interpretarla y entenderla para intentar transformarla en función de valores democráticos, de justicia y equidad.

Entre las IES se encuentra la UCI, que tiene como misión la formación de profesionales comprometidos con su patria y altamente calificados a través de un modelo de formación caracterizado por el vínculo estudio-trabajo y la realización de tareas investigativas y de carácter extensionistas. Como elemento integrador que facilita el flujo cultural entre la universidad y la sociedad, se cuenta con un proceso transversal a los de formación e investigación que es el extensionista, logrando así preservar y desarrollar la cultura, como necesidad de la sociedad. Entre las salidas del proceso extensionista definidas por el PNEU se puede mencionar: los programas, proyectos, actividades, acciones y tareas extensionistas; estructuras organizativas que se relacionan entre sí y establecen enlaces, para continuar del programa a la tarea. Resaltando el proyecto

extensionista, el cual se ha transformado en un elemento fundamental para la gestión y reforzamiento, para así fomentar las transformaciones que se requieren en este proceso. Debido a que en la actualidad los enfoques son multidisciplinarios, resulta difícil clasificar los proyectos desde un solo punto de vista, porque los mismos tributan a más de una temática o sector. Sin embargo, cuando la clasificación se refiere a concepciones, principios y enfoques, es posible diferenciar uno u otro caso (González Fernández-Larrea y González González, 2004).

Cada sociedad aspira a un determinado modelo de desarrollo deseable como idea a alcanzar. El mismo se materializa en la planificación de políticas que se concretan a través de planes y programas, los cuales se ejecutan mediante proyectos. Todo esto, visto desde la realidad de las IES adscriptas al MES, se concreta en los proyectos extensionistas, que se ejecutan a partir de lo orientado en el PNEU. El proyecto, como unidad de planificación, constituye un plan de acción integrador de carácter prospectivo. Es importante destacar que todo proyecto, por ser un instrumento de transformación, tiene un carácter creador e innovador. Sin embargo, deben ser diseñados con racionalidad para lograr ser eficaces. Los proyectos extensionistas constituyen alternativas para potenciar iniciativas desde la universidad, encaminadas a dinamizar procesos que resuelvan los problemas locales. Una de las premisas que caracteriza un proyecto extensionista es la de generar una dinámica participativa de solución de problemas que propicie un proceso de transformación local basado en la apropiación de valores, conocimientos, métodos y técnicas por parte de los actores (Educación Superior de la República de Cuba, 2022).

De manera general la gestión de los proyectos se realiza de forma empírica por iniciativa de los diferentes grupos presentes en cada una de las instituciones, desde los intereses particulares, necesidades locales o nacionales lo que motiva el desarrollo del proceso de la extensión universitaria. En el caso particular de la UCI la gestión de los proyectos extensionistas presenta un conjunto de limitaciones, dentro de estas se pueden evidenciar las siguientes:

1. No hay garantía de disponibilidad de un espacio para visualizar el impacto de los proyectos, su objetivo, el por qué y posibles contactos, lo que afecta la vinculación de nuevos miembros de la comunidad universitaria a los proyectos de un área en particular, lo que trae consigo la limitación de miembros y gestores.
2. Tras la gran cantidad de información que generan los proyectos, existe poca efectividad en la gestión de dicha información, la cual se realiza de forma semi informatizada. Lo que provoca la poca disponibilidad cuando se requiere.
3. Poco aprovechamiento de la infraestructura tecnológica de la universidad, por lo que existen dificultades para la planificación de los proyectos, el registro y control de los recursos humanos, adquisiciones, presupuesto, así como las actividades y acciones correspondientes, lo que afecta la promoción de las actividades, retrasos en la entrega de información que puede desembocar en que no se apruebe dichas actividades y exista falta de recursos en la concepción del trabajo.

A partir de la situación problemática descrita anteriormente se plantea como **problema de investigación** ¿Cómo contribuir a la gestión de los proyectos extensionistas en la Universidad de las Ciencias Informáticas?

A partir del problema planteado, se define como **objeto de estudio** la gestión de proyectos. Lo anterior planteado permite identificar como **campo de acción** la gestión de los proyectos extensionistas de la Universidad de las Ciencias Informáticas.

Para darle solución al problema anterior, se define como **objetivo general** de la investigación, desarrollar un sistema informático que contribuya a la gestión de los proyectos extensionistas en la Universidad de las Ciencias Informáticas. De donde se derivan los siguientes objetivos específicos:

- Elaborar el marco teórico de la investigación sobre la gestión de la información asociada a la gestión relacionada con los proyectos extensionistas.
- Diagnosticar el estado actual de la gestión de la información asociada a los proyectos extensionistas en la UCI.
- Diseñar el sistema informático para la gestión de información asociada a la gestión de los proyectos extensionistas de la UCI.
- Implementar el sistema informático para la gestión de información asociada a la gestión de los proyectos extensionistas de la UCI.
- Validar el sistema informático para la gestión de información asociada a la gestión de los proyectos extensionistas de la UCI.

Para guiar la investigación se define como **hipótesis**: el desarrollo de un sistema informático contribuye a la gestión de los proyectos extensionistas en la Universidad de Ciencias Informáticas.

- Variable independiente: Sistema informático.
- Variable dependiente: Gestión de proyectos extensionistas.

Las tablas 1 y 2 muestran a la operacionalización de las variables independiente y dependiente respectivamente.

Tabla 1. Operacionalización de la variable independiente (Fuente: Elaboración propia).

Variable independiente	Indicadores	Unidad de medida
Sistema informático	Funcionalidad	Alta: correcto funcionamiento Media: funcionalidades incompletas Baja: errores en la implementación
	Interfaz gráfica	5: Interfaz consecuente y botones coherentes 4: Botones no coherentes 3: No se ajusta a las reglas de oro 2: Errores ortográficos y no se ajusta a las reglas de oro
Continúa en la siguiente página		

Variable independiente	Indicadores	Unidad de medida
	Satisfacción del usuario	6: Contradictoria 5: Clara insatisfacción 4: Más insatisfecho que satisfecho 3: No definida 2: Más satisfecho que insatisfecho 1: Clara satisfacción

Tabla 2. Operacionalización de la variable dependiente (Fuente: Elaboración propia).

Variable dependiente	Dimensión	Indicadores	Unidad de medida
Gestión de proyectos extensionistas en la UCI	Organización	Apoyo a la toma de decisiones	Excelente: Provee de gráficos y reportes Bien: Provee de reportes básicos Mal: No tributa de información para la toma de decisiones
		Solo accede a la información el personal autorizado	Definición de roles.
	Centralización	Información almacenada de manera centralizada	Diseño de datos
	Disponibilidad	Tiempo de consulta de la información en segundos	Bien: Menos de 2 segundos Regular: Hasta 4 segundos Mal: Más de 5 segundos

Para darle solución a los objetivos tratados en la investigación se utilizaron los siguientes **Métodos Científicos**:

- **Métodos teóricos**

- **Histórico-Lógico**: se emplea para estudiar la evolución histórica y tendencias actuales de la dirección de proyectos extensionistas, para la construcción de los referentes teóricos que sustentan la investigación, el estudio de soluciones similares, así como de las metodologías de desarrollo, las tecnologías, herramientas, frameworks (marcos de trabajo), lenguajes de programación y de modelado utilizados.
- **Análisis-Síntesis**: empleado para el análisis, evaluación y selección de las técnicas a emplear en el desarrollo del sistema. Así como para sintetizar la información que se obtuvo mediante la entrevista con el cliente de manera que pudiera ser usada en el desarrollo del mismo, además, en la identificación de los elementos del marco teórico de la investigación.
- **Modelación**: para realizar una representación del proceso estudiado que sirva de guía en el desarrollo del sistema, y mediante este, identificar las características y relaciones fundamentales que den cumplimiento a los requisitos funcionales de la solución propuesta.

- **Métodos empíricos**

- **Entrevista:** empleado en los encuentros con el cliente para obtener la información necesaria que permita determinar las características, cualidades y requisitos con los que debe contar la propuesta de solución.
- **Observación:** se utilizó para identificar buenas prácticas y vulnerabilidades de los sistemas similares para la gestión de procesos de la Dirección de extensión universitaria.

El presente trabajo de diploma está estructurado de la siguiente manera: introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas empleadas durante el desarrollo de la investigación y, por último, se presenta una serie de anexos que complementan la memoria escrita. A continuación, se muestra la descripción de los capítulos.

- **Capítulo 1. Marco teórico referencial sobre el sistema para la gestión de los proyectos extensionistas de la Universidad de las Ciencias Informáticas:** se realiza un estudio del arte sobre los aspectos teóricos que sustentan el desarrollo de la solución propuesta. Se describe el proceso de desarrollo de software, así como las tendencias, técnicas, metodología y tecnologías usadas en la propuesta de solución.
- **Capítulo 2. Características del sistema para la gestión de los proyectos extensionistas de la Universidad de las Ciencias Informáticas:** se identifican y describen los conceptos asociados al dominio del problema y los procesos relacionados con el negocio teniendo en cuenta la metodología seleccionada en el capítulo anterior. Se definen cuáles son los requerimientos funcionales, no funcionales y el modelo de datos.
- **Capítulo 3. Implementación y prueba del sistema para la gestión de los proyectos extensionistas de la Universidad de Ciencias Informáticas:** en este capítulo se define el estándar de codificación que sirve de guía para la implementación de la solución propuesta, así como la estrategia de pruebas a aplicar para lograr un correcto funcionamiento.

Marco teórico referencial sobre el sistema para la gestión de los proyectos extensionistas de la Universidad de las Ciencias Informáticas

En el presente capítulo se realizará un análisis e investigación de los principales términos para una mejor comprensión del trabajo, así como la realización de un estudio sobre los sistemas de gestión de proyectos extensionistas, tanto en el ámbito nacional, como internacional. Se abordará un análisis del estado del arte del tema propuesto. Se incluye además una investigación de la metodología, herramientas y tecnologías a emplear en el desarrollo de la propuesta de solución, lo que facilita el desarrollo de la investigación y permite el cumplimiento del objetivo principal.

1.1. Fundamentos teóricos sobre la gestión de los proyectos extensionistas

Con el propósito de entender el desarrollo de la investigación se hace necesario definir y analizar conceptos de relevancia que brindan soporte teórico-metodológico y conceptual a la misma.

1.1.1. Extensión universitaria

La Extensión universitaria (EU) se puede definir como: el proceso que tiene como objetivo promover la cultura en la comunidad intrauniversitaria y extrauniversitaria, para contribuir a su desarrollo cultural (González Fernández-Larrea y González González, 2004). Así como la promoción del desarrollo cultural en las IES para contribuir a la formación integral de la comunidad universitaria y su entorno, con el propósito de elevar el desarrollo cultural de la comunidad universitaria y la población en general (MES, 2022).

La función de la EU es de forma general aquella que apunta hacia preservar y desarrollar la cultura, como necesidad intrínseca de la sociedad, que parte de un elemento esencial y común en toda la actividad universitaria: la cultura, entendida en su acepción más amplia como todo el sistema de creación del hombre, tanto material como espiritual que coloca a esta institución de educación superior como facilitadora para que la comunidad construya, elabore, reelabore, produzca, cree y se apropie de su propia cultura (González Fernández-Larrea y González González, 2004).

Por otra parte, en la relación sociedad - cultura, además de la preservación y el desarrollo de la cultura, existe la necesidad de la elevación del desarrollo cultural de la población, lo que también forma parte de la misión social de la universidad; pero que no encuentra solución solo en las funciones de docencia e investigación y tiene entonces que, para cumplir dicho encargo, promover la cultura que preserva y desarrolla por medio de la extensión (González Fernández-Larrea y González González, 2004).

La EU como proceso tiene como objetivo promover la cultura en la comunidad intrauniversitaria y extrauniversitaria, para contribuir a su desarrollo cultural (ibíd.).

Este proceso como resultado de las relaciones sociales que se dan entre los sujetos que en él participan está dirigido de un modo sistémico y eficiente, a la promoción de cultura para la comunidad intra y extrauniversitaria como objetivo principal, con vistas a la solución del problema social: necesidad de contribuir al desarrollo cultural de la comunidad, mediante la apropiación de la cultura que ha acumulado la sociedad en su desarrollo; a través de la participación activa de la comunidad universitaria y extrauniversitaria; planificada en el tiempo y observando ciertas estructuras organizativas; con ayuda de ciertos objetos; instrumentando indicadores que permitan medir la calidad y cuyo movimiento está determinado por las relaciones causales entre sus componentes y de ellos con la sociedad que constituyen su esencia (ibíd.).

Por tanto, después de realizar un análisis sobre la EU, se define para la presente investigación el siguiente concepto: “proceso de retroalimentación cultural entre la universidad y la comunidad con el objetivo de fomentar, extender la presencia universitaria en la sociedad y su relación estrecha con el pueblo”.

1.1.2. Gestión de la extensión universitaria

Aportela, 2016 define la gestión de la EU como el sistema de interacciones constantes y sistemáticas de los procesos, entre ellos y el entorno, a partir de una gestión eficaz y eficiente y pendiente de su mejora continua, que permita establecer y cumplir con la política de la organización y lograr sus objetivos y satisfacción de sus grupos de interés teniendo en cuenta sus actores y el entorno.

La interacción en la gestión de los subprocesos de EU con el resto de los procesos sustantivos, los estratégicos y los de apoyo se lleva a cabo en un entorno cambiante, donde los actores deben estar preparados para la dinámica de las condiciones actuales. Por ello, asumir y plantearse el diseño de un sistema de información para la gestión del proceso, no solo contribuirá a su operacionalización a partir del uso de las TIC, sino que lo ubicará en el contexto actual y según su propia dinámica (González Aportela; Batista Mainegra; González Fernández Larrea; Zambrano Loor y Balsinde Herrera, 2022).

En la figura 1.1 se expone cómo se gestiona el proceso de EU definido por (ibíd.). Donde los procesos universitarios se interrelacionan unos con otros, pues los procesos claves, dependen de los procesos estratégicos y de los procesos de apoyo; así como estos procesos se retroalimentan de los procesos clave para perfeccionar su gestión.

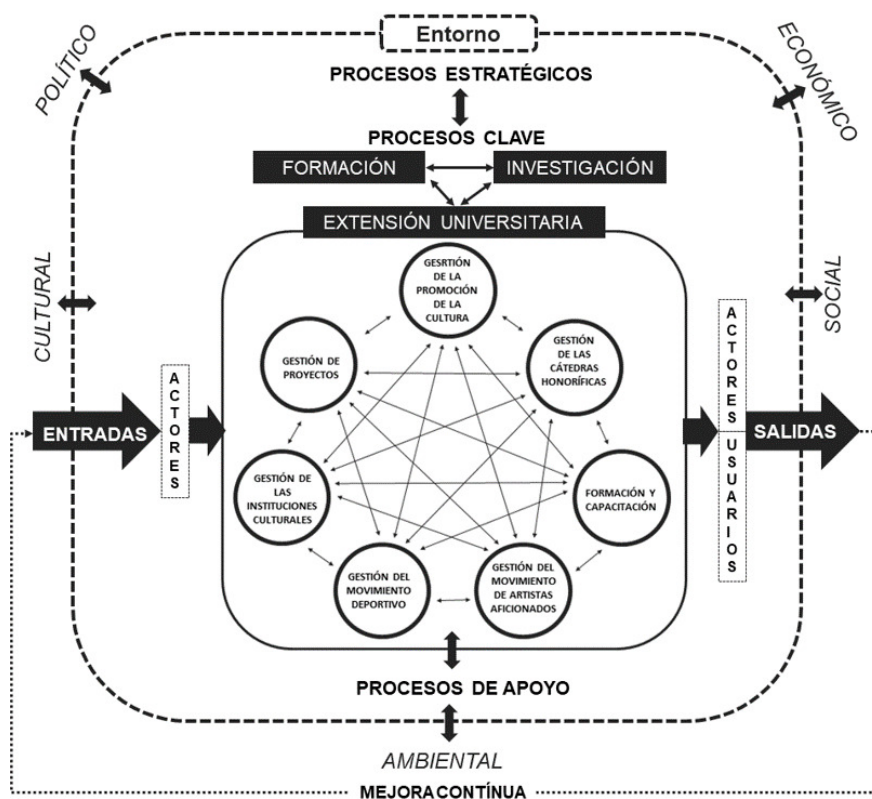


Figura 1.1. Procesos de la Extensión Universitaria (fuente: (González Aportela; Batista Mainegra; González Fernández Larrea; Zambrano Loor y Balsinde Herrera, 2022))

La universidad en su totalidad depende del entorno, de los cambios y perspectivas de las condiciones sociales, políticas, culturales y económicas para llevar a cabo la gestión institucional. Cada proceso se retroalimenta de las salidas de otro, por lo tanto, una salida de uno puede ser la entrada de otro (González Aportela y Batista Mainegra, 2017).

1.1.3. Proceso

Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí que, a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido. Los procesos deben estar correctamente gestionados empleando distintas herramientas de la gestión de procesos (Maldonado, 2018).

Es el conjunto de actividades y tareas, definidas y específicas, que lo particularizan, y que permiten la transformación de las entradas de diferentes insumos en salidas, a partir de una gestión eficiente y eficaz, teniendo en cuenta los recursos humanos, de información, técnicos, financieros y materiales para contribuir al cumplimiento de la misión de la organización en interrelación con la sociedad (Aportela, 2016).

Condiciones de un proceso

- Se pueden describir las entradas y las salidas.
- El proceso cruza uno o varios límites organizativos funcionales.
- Una de las características significativas de los procesos es que son capaces de cruzar verticalmente y horizontalmente la organización.
- Se requiere hablar de metas y fines en vez de acciones y medios. Un proceso responde a la pregunta “qué”, no al “cómo”.
- El proceso tiene que ser fácilmente comprendido por cualquier persona de la organización.
- El nombre asignado a cada proceso debe ser sugerente de los conceptos y actividades incluidos en el mismo.

Subprocesos: Son partes bien definidas en un proceso. Su identificación puede resultar útil para aislar los problemas que pueden presentarse y posibilitar diferentes tratamientos dentro de un mismo proceso.

1.1.4. Gestión de procesos

La gestión de procesos es una disciplina de gestión que propone la optimización de los procesos, de principio a fin, para así generar más valor para el cliente y mejorar el desempeño, orientándolo al logro de sus objetivos estratégicos, es una metodología corporativa y de disciplina de gestión, cuyo objetivo es mejorar el desempeño (eficiencia y eficacia) y la optimización de cada uno de los procesos de negocio de una organización, a través de la gestión de los procesos que se deben diseñar, modelar, organizar, documentar y optimizar de forma continua. Por lo tanto, puede ser descrito como un proceso de optimización de procesos (Maldonado, 2018). Al gestionar procesos, lo que en realidad hacemos es cambiar las unidades de organización a su estructura lógica y natural.

La gestión de procesos es un enfoque metodológico en el uso de técnicas y herramientas que comprende los aspectos relacionados al mejoramiento y al cambio radical de los procesos, su implantación en la organización, la visualización de nuevas estrategias de trabajo, el diseño o rediseño de las actividades y la implementación de un cambio en sus tres dimensiones: la tecnológica, la humana y la organizativa (Silva García, 2016).

Por lo tanto, podemos definir la gestión de procesos enfocado a esta investigación como la aplicación de estrategias, técnicas y herramientas metodológicas y flexibles en un período de tiempo determinado para mejorar el trabajo y darles cumplimiento a los objetivos fundamentales y así optimizar los procesos a los que se enfoque, cada uno con especificaciones y particularidades que los definen.

1.1.5. Dirección de extensión universitaria

La Dirección de extensión universitaria (DEU) es el órgano encargado metodológicamente de coordinar, promover y fomentar el desarrollo de la EU para toda la educación superior (Aliaga; Ruiz y J. C. E. Pérez, 2019). Así como, el órgano rector del proceso integrador y dinamizador del flujo cultural y la comunicación

entre la universidad y la sociedad, así como también entre cada uno de sus miembros (Espronceda Pérez; Romero Ramos y Remedio Frometa, 2021).

La DEU es el área principal dentro de una entidad universitaria encargada de promover, planificar, divulgar, los diferentes procesos de la EU para facilitar el desarrollo cultural entre la sociedad y la comunidad universitaria. En la UCI, esta dirección, dentro de sus líneas de trabajo, cuenta con la creación de proyectos extensionistas con el fin de estrechar cada vez más los lazos de integración de la universidad con la sociedad.

1.1.6. Proyectos extensionistas

El proyecto extensionista es el eje articulador de la gestión de la EU y elemento clave para dinamizarla y promover las transformaciones que se requieren en este proceso. Constituye una unidad más operativa dentro del proceso de planeación del trabajo sociocultural universitario, pues ofrece tratamiento a situaciones y problemas específicos. Es una unidad mínima de asignación de recursos, que a través de un conjunto concreto de actividades, acciones y tareas pretende modificar o transformar una parcela de la realidad sociocultural disminuyendo o eliminando un déficit o solucionando un problema.

Dentro de la UCI los proyectos extensionistas cuentan con ciertas características entre las que se destacan:

- Aportan nuevo conocimiento científico acerca de la realidad.
- Tienen carácter estratégico para la transformación de una problemática o desarrollo científico y económico de la institución.
- Constituyen la base para el desarrollo futuro de nuevos saberes, productos, tecnologías, sistemas y procesos.

Actualmente se reconoce que existen en la UCI, 20 proyectos extensionistas. El estudio del PNEU y los documentos que describen a dichos proyectos permitió comprobar que en la UCI:

- Los proyectos extensionistas son aprobados en el Consejo de dirección del área en que serán implementados y tienen un documento que los describe, pero los criterios para la aprobación no son similares en todas las áreas. Para decidir la aprobación de un proyecto, no en todos los consejos de dirección se exige presentar el desglose de costos y recursos por actividades, acciones y/o tareas.
- No en todos los proyectos extensionistas se registra y controla información sobre sus recursos humanos y adquisiciones, por lo que les resulta difícil planificar eficientemente las actividades, acciones y/o tareas a desarrollar. En aquellos proyectos que sí lo hacen, se realiza con el auxilio de herramientas ofimáticas, que no ofrecen protección a los datos, por lo que pueden ser alterados con facilidad.
- En correspondencia con su duración y disponibilidad de recursos los proyectos extensionistas planifican las actividades, acciones y/o tareas que deben realizar para alcanzar sus objetivos. Cada mes los coordinadores de proyecto deben entregar al Vicedecano de Extensión o la persona designada en el área para atenderlos, la planificación del próximo mes y la relación de recursos que requieren de la Facultad o Universidad, para incluirla en la planificación mensual del área. Las actividades, acciones y/o tareas, no siempre son aprobadas, en ese caso los proyectos deben reajustar su planificación.

- El proceso de gestión de un proyecto extensionista genera volúmenes considerables de información que se gestiona de forma engorrosa pues se encuentra parcialmente informatizada y almacena en múltiples formatos y lugares, por lo que no siempre está disponible cuando se requiere. Entre los formatos que se utilizan para almacenar la información está el papel. El paso del tiempo y la excesiva manipulación han provocado pérdida de información por deterioro de la documentación.
- De las actividades, acciones y/o tareas que ejecutan los proyectos extensionistas para alcanzar sus objetivos se almacenan fotos, archivos de audio y video, papelería utilizada y otras evidencias como constancia de su realización. Actualmente los proyectos no disponen de un espacio donde centralizar estas evidencias, para que estén disponibles cuando se requieran.
- El reflejo en los medios de comunicación de las actividades, acciones y/o tareas realizadas por los proyectos extensionistas, depende de que se informe con tiempo de las mismas y del interés de los medios por cubrir las, por lo que un número considerable de ellas no son divulgadas. En algunos casos los miembros de los proyectos extensionistas asumen el reto de generar artículos sobre sus acciones, pero no siempre cuentan con la calidad ni la estructura requerida para ser publicados.
- Los proyectos extensionistas no disponen actualmente de un espacio donde visualizar su objetivo, justificación, vías de contacto y otras informaciones de interés para sus interesados. Lo anterior no favorece la incorporación de nuevos miembros y provoca que se generen matrices de opinión erróneas sobre algunos proyectos.

La figura 1.2 muestra el flujo del proceso de aprobación de un proyecto extensionista.

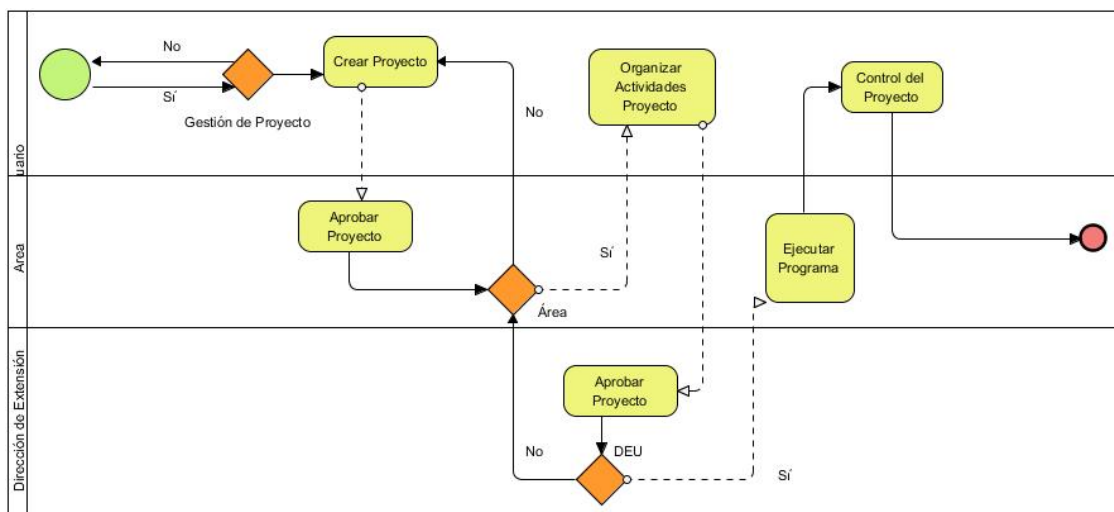


Figura 1.2. Diagrama de flujo. Fuente: (Elaboración propia)

1.1.7. Gestión de proyectos extensionistas

Según el diccionario Ilustrado, 1989, la gestión es: Acción y efecto de administrar, planificación, organización, dirección y control de los recursos de una organización.

Entre los diferentes tipos de gestión se encuentra: la ambiental, la tecnológica, la pública, de proyecto, de conocimiento, de la información, entre otros. La gestión tiene el fin de obtener el máximo beneficio posible; este beneficio puede ser económico o social, dependiendo esto de los fines que persiga la organización.

Según DEU, 2016, un proyecto es el conjunto de acciones relacionadas entre sí, con un fin común y que tienen a su disposición un grupo de recursos materiales y humanos para lograr en un tiempo determinado los objetivos propuestos.

Un proyecto puede ser simplemente un plan o una idea, al menos en la vida cotidiana o en el lenguaje coloquial. Cuando se habla de proyectos en un marco más formal, es habitual que puedan distinguirse diversas etapas en su desarrollo: primero surge una idea que reconoce una oportunidad, luego se diseña el proyecto en sí mismo con la valoración de las estrategias y opciones y finalmente se ejecuta el plan. Tras la concreción del proyecto, llega la hora de evaluar los resultados según el cumplimiento, o no, de los objetivos fijados.

La gestión de proyectos es la disciplina de utilizar principios, procedimientos y políticas establecidos para guiar con éxito un proyecto desde su concepción hasta su finalización. Es un conjunto de metodologías para planificar y dirigir los procesos de un proyecto.

Según Aportela, 2016 la gestión de los proyectos extensionistas desarrolla el trabajo socio comunitario con la comunidad universitaria y la sociedad en interacción dialógica, en función de un problema social, a nivel comunitario, local, regional o nacional, potenciando la formación integral desde la perspectiva de la responsabilidad social universitaria.

Por tanto, los proyectos extensionistas se gestionan a partir de ciertas particularidades del proceso de extensión, particularidades brindadas por la metodología que la sustenta como es la promoción cultural. Entre otras particularidades podemos encontrar la flexibilidad, la contextualización y la adaptabilidad. Estos proyectos presentan diferencias debido a que dan respuesta a distintos contextos.

Por lo que podemos definir la gestión de proyectos extensionistas como: Conjunto de procedimientos y estrategias flexibles que definen actividades y acciones desde un inicio, hasta un final, con el objetivo de promover cultura, intra y transdisciplinaria.

1.1.8. Tarea extensionista

Una tarea es una labor o trabajo que realiza alguien. Según González Fernández-Larrea y González González, 2004 las tareas extensionistas representan la célula constitutiva del proceso extensionista y del trabajo sociocultural universitario, por ser la expresión más pequeña del mismo que mantiene sus características y relaciones fundamentales. Por eso la subdivisión de una tarea implicaría el desmembramiento del proceso y la pérdida de sus cualidades. Las tareas constituyen los eslabones primarios que permiten el alcance de los resultados esperados y la solución del problema.

1.1.9. Acciones extensionistas

Definido por (González Fernández-Larrea y González González, 2004) las acciones extensionistas son la forma organizativa del trabajo sociocultural universitario, está integrada por un conjunto de tareas que se identifican fuertemente entre sí por sus características similares y su orientación a propósitos muy específicos. Las acciones pueden existir en casos excepcionales independientes a las actividades y proyectos, pero siempre responden a la política y estrategias del programa. En función de responder a las características del trabajo sociocultural universitario se requiere que su equipo gestor tenga en cuenta las diferentes metodologías y técnicas para su planificación, organización y evaluación.

1.1.10. Actividad extensionista

Una actividad es el conjunto de operaciones o tareas propias de una persona o entidad. Según DEU, 2016 las actividades extensionistas son aquellas que como parte de la operacionalización de la planeación del trabajo sociocultural universitario ofrecen tratamiento a una situación y problema específico. Se identifican como tales el conjunto de acciones y tareas que tienen un carácter similar o están estrechamente relacionadas y que se orientan al cumplimiento de un objetivo específico o a parte del mismo.

Estas actividades poseen algunos procedimientos específicos, pero al mismo tiempo existen pasos comunes para su diseño y ejecución, según Aportela, 2016, es indispensable tomar en consideración las condiciones del entorno, las relaciones con los otros procesos sustantivos, así como con los procesos de apoyo y estratégicos de la universidad. Las actividades definidas para cada proceso se ajustan en las áreas universitarias, teniendo en cuenta el diagnóstico realizado y las características propias de cada unidad organizativa, así mismo se actualizan en función del período de trabajo y las necesidades y demandas del entorno, demostrando el carácter abierto, flexible y contextualizado.

Las actividades pueden formar parte o no de un proyecto, su planificación puede ser independiente de estos, pero siempre estarán en el marco de la política que establece un programa.

ENTRADAS	GESTIÓN DE PROYECTOS	SALIDAS/ RESPONSABLE
<ul style="list-style-type: none"> ▪ Diagnóstico. ▪ Convocatorias. ▪ Objetivos OACE, MES, UH. ▪ Solicitudes y demandas de las organizaciones políticas y de masas, de la comunidad universitaria y extrauniversitaria. 	<ul style="list-style-type: none"> ▪ Organización y planificación del proyecto. ▪ Gestionar el Desarrollo del proyecto. ▪ Gestionar la divulgación del proyecto. ▪ Evaluar y controlar el proyecto. ▪ Evaluación de impacto de los proyectos. ▪ Gestionar la pertinencia y viabilidad de los proyectos. 	<ul style="list-style-type: none"> ▪ Ficha del proyecto. Resp. Directivo de extensión universitaria. ▪ Registro informativo del proyecto. Resp. Jefes de proyecto. ▪ Programación y cronograma de actividades, tareas, acciones del proyecto. Resp. Jefes de proyecto. ▪ Evidencias del proyecto Resp. Jefes de proyecto. ▪ Visitas de control a las áreas. Resp. Directivo de la institución. ▪ Informes parciales y anuales. Resp. Directivo de la institución.

Figura 1.3. Gestión de Proyectos. Fuente: (Aportela, 2016)

1.1.11. Programa extensionista

Un programa extensionista es un proyecto o planificación ordenada de las distintas partes o actividades que componen algo que se va a realizar. Definido por (DEU, 2016) como un conjunto de acciones interrelacionadas y coordinadas, con el fin de alcanzar objetivos determinados dentro de los límites de una política dada, de un presupuesto y períodos de tiempo determinados. Pueden ser más o menos globales de acuerdo con el grado de generalidad que asumen. El programa constituye la forma organizativa más general para la extensión universitaria. Lo que conlleva al PNEU, redactado con rigor científico y lenguaje asequible, visualizar y comprender la extensión como proceso, y las claves de su éxito radican en sus nexos con la política cultural cubana y la posibilidad de materializar mayores niveles de concreción de la estrategia en acciones. El mismo expone los conceptos y formas de gestión y acción vigentes, fruto del análisis y pensamiento colectivo de especialistas y directivos de todas las universidades del país, el cual tiene la intención de convertirse en un instrumento eficaz para la gestión del proceso extensionista en las universidades desde las condiciones y realidades específicas sobre las que actúan.

Según Aportela, 2016 el PNEU está concebido con una máxima flexibilidad en su diseño y aplicación, de forma tal, que, a partir de la dinámica y las condiciones del entorno, se pueda buscar un equilibrio coherente, entre las exigencias que plantea el desarrollo de este proceso en la educación superior cubana y las necesidades que demanda cada una de las realidades objeto de transformación por las universidades. Este tiene como objetivo general desarrollar la extensión universitaria, transformándola, facilitando la promoción y elevación de la cultura general integral de la comunidad universitaria y su entorno social.

A pesar de esto, el programa fue diseñado en el año 2004, y aún no se ha convertido en un instrumento de gestión del proceso de extensión, porque no se ha llegado a alcanzar la adecuación del programa a las peculiaridades de cada universidad en la concepción creativa de la labor extensionista. Lo que lleva a adecuar el PNEU a los tiempos actuales y a las situaciones de cada IES.

1.2. Análisis de las soluciones existentes

En la actualidad, existen diversas soluciones para la gestión de proyectos, con diferentes características, funcionamientos y accesibilidad. A continuación, se describen algunos de estos sistemas de amplio uso a nivel internacional y nacional, así como otras desarrolladas en la UCI .

1.2.1. Ámbito internacional

- **Sistema de gestión patrimonial (SIGPAC):** La sección de patrimonio cultural es una unidad especializada, orientada a gestionar políticas públicas e implementar actividades que promuevan la salvaguarda del patrimonio cultural inmaterial colombiano, apoyando principalmente el fortalecimiento de las manifestaciones y expresiones vivas de las culturas locales. La herramienta permite gestionar los diferentes programas patrimoniales, promocionar y difundir los eventos que se desarrollan en el ámbito de la gestión del patrimonio cultural (Borges Prado y Pozo González, 2019).

Este sistema queda descartado como solución al problema, teniendo en cuenta que solo se enmarca en la gestión patrimonial y no integra todos los procesos que se gestionan en los proyectos extensionistas.

- **Sistema para la prevención de la deserción en la educación superior (SPADIES, por sus siglas en inglés):** El Ministerio de Educación Nacional de Colombia propuso disminuir la deserción estudiantil en la educación superior como parte de la estrategia planteada para aumentar la cobertura, la calidad y la eficiencia educativas. Entre otras acciones, puso en marcha el SPADIES ante la necesidad de contar con una visión sectorial e integrada de la problemática de la deserción, a partir de la cual se pudiera disponer de una conceptualización, una medición y una metodología de seguimiento del fenómeno aplicables a todas las instituciones de educación superior de ese país.

El SPADIES consolida y ordena información que permite hacer seguimiento a las condiciones académicas y socioeconómicas de los estudiantes que han ingresado a la educación superior en el país. De esta manera, permite conocer el estado y evolución de la caracterización y del rendimiento académico de los estudiantes, lo cual es útil para establecer los factores determinantes de la deserción, para estimar el riesgo de deserción de cada estudiante y para diseñar y mejorar las acciones de apoyo a los estudiantes orientados a fomentar su permanencia y graduación (Espronceda Pérez; Romero Ramos y Remedio Frometa, 2021).

Esta propuesta de solución no se ajusta a la presente investigación pues solo se limita al análisis de la permanencia en la educación superior colombiana a partir del seguimiento a la deserción estudiantil, pero sin embargo de este se pudo observar y tomar como referencia el ajuste que presenta el sitio a los requisitos de la web actual con colores asequibles a la vista y con un correcto posicionamiento de su menú y botones.

- **Sistema nacional de información de la educación superior (SNIES):** Es un sistema de información creado para responder a las necesidades de recopilar información de la Educación Superior en Colombia. Este sistema como fuente de información, en relación con las instituciones y programas académicos aprobados por el Ministerio de Educación Nacional, consolida y suministra datos, estadísticas e indicadores. Por la consolidación de información que hace el SNIES de las diferentes entidades del sector, es posible darle a conocer datos de interés a la comunidad en general sobre los avances en la Educación Superior colombiana (ibíd.).

Este sistema no se ajusta a la propuesta de esta presente investigación pues este solo se limita a recopilar y organizar información sobre el sistema de educación superior de Colombia y, además porque para la realización del mismo se utilizaron softwares privativos de pago, pero, sin embargo, de este se pudo observar y estudiar las buenas prácticas de programación empeladas en el mismo.

- **Microsoft project:** Es un software diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar el presupuesto y analizar cargas de trabajo. Es, además, una aplicación de escritorio que dificulta el trabajo fluido en equipo, principalmente si el personal se encuentra disperso físicamente (Silva García, 2016).

Otra desventaja de Microsoft project es su condición de software propietario que implica el pago de las licencias pertenecientes a Microsoft, así como de los cursos de capacitación para el personal que usará la herramienta. Además, no permite medir la productividad de las máquinas ni de las personas y carece de las herramientas básicas para la planeación de la mayoría de proyectos (Silva García, 2016).

1.2.2. Ámbito nacional

- **Sistema de gestión de la Nueva Universidad (SIGENU):** Es un proyecto que se inicia en el Taller Nacional de Informatización de la Gestión Académica organizado por el MES en marzo del 2004 y tiene su origen dadas las urgentes necesidades planteadas por la máxima dirección de este organismo de un sistema de almacenamiento y procesamiento eficiente de la información de la actividad docente universitaria en todo el país.

SIGENU es un sistema informático distribuido del tipo cliente-servidor, construido sobre la base de la utilización de software libre, con el uso de Java como lenguaje de programación. Constituye la principal línea de desarrollo para la informatización de la gestión de la información en la Educación Superior en Cuba. Actualmente cuenta con una base de datos de más de 120 tablas donde se almacenan todos los datos necesarios para el funcionamiento del sistema (Saavedra y Mora, 2018).

- **Sistema informático de gestión universitaria (SIGUA):** Esta aplicación web fue diseñada con el objetivo principal de optimizar el trabajo en los distintos departamentos y secretarías docentes, de igual manera a los estudiantes en el Instituto Superior de Arte. La interfaz web que brinda esta aplicación permite adicionar nuevos datos y consultarlos. La información introducida se almacena en la base de datos modelada para este fin (Espronceda Pérez; Romero Ramos y Remedio Frometa, 2021).

Para la presente investigación esta propuesta de solución no pudo ser utilizada porque solo se encarga de la adición y consulta de los datos de la secretaría docente, pero por otra parte este sirvió de ayuda pues se pudo apreciar las buenas prácticas para el desarrollo de un sistema de gestión en una universidad.

- **Página web extensión universitaria de la Universidad de Holguín (UHo):** Es una página web que facilita la divulgación de actividades, así como su integración, es de acceso rápido y fácil para el público la cual tiene como objetivo mantener informados al estudiantado y al claustro docente. También se muestran los resultados alcanzados hasta la actualidad (Quiñones Laffita; Hernández Torres y Córdón González, 2020).

Este sitio no cumple con las principales funcionalidades de la presente investigación relacionadas con la gestión de los proyectos extensionistas, pues solo se encarga de la divulgación de las actividades para mantener informado al estudiantado de la universidad, pero no cuenta con un módulo para la gestión de las actividades antes mencionadas. De esta propuesta de solución se pudo tomar la correcta utilización de los softwares libres para la realización del sitio.

Universidad de las Ciencias Informáticas

- **Paquete para la gestión de proyectos (GESPRO):** Es una herramienta web para la dirección inte-

grada de proyectos y la ayuda a la toma de decisiones, desarrollada por la UCI . Desde la vista de la arquitectura de procesos, el sistema se rige por el estándar Cuerpo de conocimiento de gestión de proyectos (PMBOK, por sus siglas en inglés) del Instituto de Gestión de Proyectos (PMI, por sus siglas en inglés) y las buenas prácticas del programa de formación del Máster en Gestión de Proyectos de la UCI . Es una herramienta web, desarrollada en 2008 en la UCI . Permite gestionar de una manera simple la documentación, los costos, los riesgos, los recursos y las actividades de un proyecto. La herramienta fue desarrollada tomando como base la Guía de los Fundamentos de la Dirección de Proyectos (PMBOK), del PMI, en su tercera edición (Espronceda Pérez; Romero Ramos y Remedio Frometa, 2021).

Sin embargo la herramienta solo permite el acceso a la información a los miembros que forman parte del proyecto lo cual limita la socialización de la labor realizada desde esa forma organizativa, aborda áreas del conocimiento asociadas a la gestión de proyecto que no son relevantes en la presente investigación como la gestión de costos y facturaciones, la gestión de la logística y contratos con proveedores, así como la gestión de contrataciones e interesados; igualmente la gestión de evidencias asociadas a las actividades no se tiene en cuenta.

- **Sistema de Gestión Universitaria (SGU):** Desarrollado por la Dirección de Informatización perteneciente a la Dirección General de Tecnología de la UCI , gestiona un grupo de información de las áreas docentes, productivas e investigativas de la universidad referente a residencia, cooperación, desarrollo, tecnologías y extensión.

Ambas soluciones sirvieron de gran ayuda para la presente investigación pues ambas utilizan la identidad marcaria propuesta para los centros de desarrollo de la UCI, también utilizada en la propuesta de solución de esta investigación, de estos sistemas se pudo observar la correcta utilización de las herramientas y tecnologías para el desarrollo de los sistemas.

1.2.3. Comparación entre los sistemas de homólogos

Tabla 1.1. Comparación de sistemas homólogos. (Fuente: Elaboración propia)

Indicadores/Sistemas	Ámbito Internacional				Ámbito Nacional				
	SIGPAC	SPADIES	SNIES	Project	SIGENU	SIGUA	UHo	GESPRO	SGU
Interfaz gráfica	SI	NO	SI	SI	SI	SI	NO	SI	SI
Código abierto	NO	NO	SI	NO	SI	SI	SI	SI	SI
Licencia privativa	NO	NO	NO	SI	NO	NO	NO	SI	NO
Permisos de usuarios	SI	SI	SI	NO	SI	NO	NO	NO	NO
Gestión de información	SI	SI	SI	SI	SI	SI	SI	SI	SI

1.2.4. Consideraciones del estudio de homólogos

Como se refleja en el estudio realizado a los sistemas anteriores, de una forma u otra se relacionan con la gestión de proyectos, pero ninguna de las soluciones analizadas responde al problema planteado. Según el análisis realizado ninguno se dedica a la gestión de proyectos extensionistas. Sin embargo, sientan las bases para el desarrollo de una solución propia, ya que constituye una necesidad primordial para la universidad desarrollar un sistema que permita gestionar este proceso. Del estudio realizado se toma en consideración las tecnologías que utilizan y las interfaces, sentando las bases para el desarrollo de la propuesta de solución.

1.3. Entorno de desarrollo de la propuesta de solución

Teniendo en cuenta que el sistema a desarrollar forma parte de un macrosistema para la gestión de información para el proceso de EU, el cliente solicita que se utilice un conjunto de tecnologías específicas y el uso de la metodología de desarrollo que se utiliza en los proyectos de desarrollo de la UCI, siendo esto unos de los requisitos no funcionales de la solución (se detalla en el capítulo 2, epígrafe 2.2.2).

1.3.1. Metodología

La presencia de una metodología en el desarrollo de un proyecto garantiza un producto con más calidad y reduce el tiempo de entrega del producto, ya que al tener una mejor planificación la producción permite cumplir con la fecha establecida.

Se utiliza como metodología de desarrollo de software la modificación realizada por la UCI a la metodología ágil Proceso unificado ágil (AUP, por sus siglas en inglés) y está definida por la universidad como el documento rector de la actividad productiva. La metodología AUP para la UCI está formada por tres fases,

(inicio, ejecución y cierre) para el ciclo de vida de los proyectos de la universidad, las cuales contienen las características de las cuatro fases (inicio, elaboración, construcción y transición) propuestas en AUP (Sánchez, 2015).

Para la presente investigación se utilizará el escenario 4 de la metodología, el cual permite a través de la Historia de usuario (HU), un encapsulamiento de los requisitos del sistema. Este escenario se aplica a proyectos con un negocio bien definido y que no sean muy extensos. Además, el cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

1.3.2. Herramientas y tecnologías

Herramientas para el modelado

- Lenguaje unificado de modelación (UML, por sus siglas en inglés) Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software. Por otra parte, UML es un lenguaje de modelado estandarizado que permite a los desarrolladores especificar, visualizar, construir y documentar artefactos de un sistema de software. Por lo tanto, UML hace que estos artefactos sean escalables, seguros y robustos en ejecución. UML es un aspecto importante involucrado en el desarrollo de software orientado a objetos. Utiliza la notación gráfica para crear modelos visuales de sistemas de software (Barragan, 2016).

Para la investigación se empleará UML en su versión 2.5 pues es el lenguaje propuesto por la metodología de desarrollo a utilizar y permite mejor comprensión de lo que se realiza.

- Ingeniería de software asistida por computadora (CASE, por sus siglas en inglés): Visual Paradigm. Es una herramienta que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE posee una fuerte integración con el gestor de base de datos PostgreSQL, proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (Kefalakis; Soldatos; Anagnostopoulos y Dimitropoulos, 2015).

Entorno de desarrollo integrado (IDE, por sus siglas en inglés)

Microsoft Visual Studio Code: Es un conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento, permitiendo a los desarrolladores crear sitios y aplicaciones web, así como otros servicios web en cualquier entorno que soporte la plataforma.

En palabras más específicas, Visual Studio es un conjunto completo de herramientas de desarrollo para la

generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C y Visual C plus utilizan todos los mismos IDE, que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Así mismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML (Studio, 2019).

Apache JMeter

Es un software de código abierto, una aplicación Java que se utiliza para realizar pruebas de carga de aplicaciones web para conocer la cantidad de usuarios simultáneos que pueden manejar en tiempo real. Esta herramienta prueba el comportamiento funcional y mide el rendimiento. Diseñada en un primer momento para probar aplicaciones web y desde entonces se ha estado expandiendo a otras funciones de prueba.

Acunetix Web Vulnerability Scanner

Es una herramienta de seguridad de aplicaciones Web automatizada. Acunetix WVS es capaz de escanear cualquier sitio web o aplicación web que es accesible a través del protocolo HTTP/HTTPS.

Con escaneo rápido, resultados completos y automatización inteligente, el mismo cierra la brecha entre la seguridad y el desarrollo reduciendo la tensión y creando una cultura de seguridad con el objetivo de empoderar a los equipos de seguridad para reducir el riesgo en todo tipo de aplicaciones web.

Lenguajes de programación

- **Python:** Es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

Según Van Rossum et al., 2007, dentro de las características del lenguaje se encuentran las siguientes:

- Propósito general: Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.
- Multiplataforma: Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- Interpretado: Quiere decir que no se debe compilar el código antes de su ejecución. En realidad, sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.

- **Interactivo:** Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- **Orientado a Objetos:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- **Funciones y librerías:** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en Zip.
- **Sintaxis clara:** Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave `begin` y `end`. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que irá dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.
- **JavaScript:** Es un lenguaje de programación que no forma parte de la plataforma Java; comparte algunos atributos con el lenguaje de programación Java, pero se desarrolló de forma independiente. Es un lenguaje interpretado que permite incluir macros en páginas Web. Proporciona los medios para controlar las ventanas del navegador y el contenido que muestran, permite programar páginas dinámicas simples, evita depender del servidor Web para cálculos sencillos, capturar los eventos generados por el usuario, optimiza los tiempos de carga y el tráfico del servidor y comprueba los datos que el usuario introduce en un formulario antes de enviarlos (J. E. Pérez, 2019).
- **Lenguaje de marcas de hipertexto (HTML, por sus siglas en inglés):** Es el lenguaje con el que se escriben las páginas web. Es un lenguaje que permite escribir texto de forma estructurada y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento. Un documento hipertexto no solo se compone de texto, puede contener imagen, sonido, vídeo, y el resultado puede considerarse como un documento multimedia. Los navegadores se encargan de interpretar el código HTML de los documentos y de mostrar a los usuarios las páginas web resultantes del código interpretado. El HTML 5 desarrolla el lenguaje HTML con mecanismos para hojas de estilo, ejecución de scripts, marcos, objetos incluidos, soporte mejorado para texto de derecha a izquierda y direcciones mezcladas, tablas más ricas y mejoras en formularios, ofreciendo mejoras de accesibilidad para personas con discapacidades (Vaughan-Nichols, 2010).
- **Hoja de estilos en cascada (CSS, por sus siglas en inglés):** Constituyen un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último

caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo `style`. CSS permite el control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente a accesibilidad. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario (Baker, 2014).

Bibliotecas

- **Bootstrap:** Es una colección de herramientas HTML y plantillas CSS, que se utilizan para el desarrollo y diseño de sitios o aplicaciones web. Es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. Trae integrado CSS, JavaScript, sintaxis HTML, diseño en malla y adaptable (Prinz y Shatalin, 2019).

Para el desarrollo del sistema se utiliza en su v5, es una herramienta de código abierto, que permite que el diseño gráfico de la página se ajuste dinámicamente, tomando en cuenta las características del dispositivo usado. Los diseños creados con Bootstrap son simples, limpios e intuitivos, permitiéndole ganar al sistema en agilidad a la hora de cargar y al adaptarse a otros dispositivos. Además, está ampliamente documentada.

- **JQuery:** es una biblioteca multiplataforma que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol Modelo de Objetos del Documento (DOM, por sus siglas en inglés), manejar eventos, desarrollar animaciones y agregar interacciones a páginas web. Es un software libre y de código abierto que ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones JavaScript asíncrono y Lenguaje de Marcado Extensible (AJAX, por sus siglas en inglés) (Denis, 2021).

1.3.3. Framework

- **Django:** Es un framework de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

Django puede ser usado para construir casi cualquier tipo de sitio web desde sistemas manejadores de contenidos y wikis, hasta redes sociales y sitios de noticias. Puede funcionar con cualquier framework en el lado del cliente, y puede devolver contenido en casi cualquier formato. Internamente, mientras ofrece opciones para casi cualquier funcionalidad que desees (distintos motores de base de datos, motores de plantillas, entre otros), también puede ser extendido para usar otros componentes si es

necesario.

Django ayuda a los desarrolladores a evitar varios errores comunes de seguridad al proveer un framework que ha sido diseñado para proteger el sitio web automáticamente. Proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar informaciones de sesión en cookies donde es vulnerable. Permite protección contra algunas vulnerabilidades de forma predeterminada, incluida la inyección SQL, scripts entre sitios, falsificación de solicitudes entre sitios y clickjacking (Vincent, 2021).

1.3.4. SGBD

Los Sistemas gestores de bases de datos (SGBD) permiten a los usuarios poder procesar, describir, administrar y recuperar los datos almacenados en una base de datos. Además, en estos sistemas se proporcionan programas y procedimientos que les brindan la posibilidad a los usuarios de poder realizar sus tareas habituales con los datos, garantizando la seguridad, independencia e integridad de los mismos. A continuación, se describe el gestor de base de datos que se utiliza:

- **PostgreSQL:** Es un SGBD objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones y restricciones; basado en el proyecto POSTGRES, de la Universidad de Berkeley. Soporta los tipos base y otros de tipo como fecha, monetarios, elementos gráficos y cadenas de bits. También permite la creación de tipos propios. Incluye herencia entre tablas, por lo que se incluye entre los gestores de objetos-relacionales. PostgreSQL se desempeña mejor en ambientes con altas cargas de usuario y consultas complejas y donde la integridad de los datos es muy importante. Una vez almacenados los datos, proporciona un sistema de respaldos en línea donde se puede ver una tabla en el estado en el que se encontraba en cierta fecha, proporcionando así un método flexible para la rápida recuperación de datos. Otra ventaja importante de este SGBD es la capacidad de su arquitectura para soportar módulos agregados (Viloria; Acuña; Franco; Hernández-Palma; Fuentes y Rambal, 2019).

1.3.5. Herramienta para control de versiones

- **Git:** Es un sistema de control de versiones distribuido de código abierto y gratuito, diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes, con rapidez y eficiencia. Git es fácil de aprender y tiene una huella pequeña con un rendimiento increíblemente rápido. Supera a las herramientas de SCM como Subversión, CVS, Perforce y ClearCase con características como bifurcaciones locales baratas, áreas de preparación convenientes y múltiples flujos de trabajo (Álvarez Carulla, 2021).

1.4. 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 sistematización de los conceptos fundamentales asociados al dominio de la presente investigación y sus relaciones, proporcionó una mayor comprensión de la propuesta de solución.
- El análisis de los sistemas homólogos existentes, permitió arribar a la conclusión que no se ajustan completamente a las necesidades de la propuesta de solución, pero si aportó una visión más amplia del objeto de estudio de la investigación. Además, posibilitó adquirir y fomentar los conocimientos necesarios para el desarrollo del sistema.
- El estudio de las tecnologías de la informática y las comunicaciones y la observación de las distintas tendencias en el desarrollo de software, permitió identificar al Django como framework de desarrollo, PostgreSQL como sistema gestor de bases de datos, Python como lenguaje de programación del lado del servidor, y como lenguaje del lado del cliente JavaScript mediante la librería JQuery.

Características del sistema para la gestión de los proyectos extensionistas de la Universidad de las Ciencias Informáticas

El presente capítulo aborda los principales aspectos relacionados con las características de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales con los que debe cumplir la solución propuesta, así como estilo arquitectónico y los patrones de diseño para lograr buenas prácticas en el diseño y posterior implementación del sistema. Igualmente se muestran los principales artefactos de ingeniería de software propuestos por la metodología utilizada.

2.1. Descripción de la propuesta de solución

Según lo planteado en la situación problemática de la investigación, la propuesta de solución se define como un sistema informático para la gestión de los proyectos extensionistas en la Universidad de las Ciencias Informáticas.

El sistema permite promover los diferentes proyectos extensionistas, que se encuentran disponibles fuera o dentro de la universidad. Para realizar la gestión de los mismos, el usuario deberá autenticarse, basado en una gestión de usuario, que permite la asignación de permisos a usuarios ya autenticados.

En la sesión de proyectos extensionistas el usuario podrá asignar nombre del proyecto, definir tipo de proyecto, la institución que lo realiza, así como el área, participantes, breve descripción, objetivos generales y específicos, por otra parte, el público al que va dirigido el proyecto y la sistematicidad del mismo. Puede gestionar la información de los participantes, conjuntamente podrá definir el responsable.

Podrá gestionar las actividades correspondientes sobre la temática del proyecto. Se permitirá la gestión de las evidencias pertinentes que se basa en la posibilidad de subir multimedia que valide la realización del proyecto, gestión que se verá limitada porque solo permitirá adicionar y listar estas evidencias dependiendo del rol de los usuarios.

Por otra parte, la gestión de los comentarios los cuales podrán ser introducidos por los usuarios registrados en el sistema, se podrá comentar los proyectos que desee, se realizará una revisión por parte del administrador

o los usuarios con los permisos necesarios para la visualización de los mismos. Una vez revisado y validado el comentario, se publica. El usuario puede gestionar su comentario si lo desea.

También el sistema permite la búsqueda de información mediante un componente de búsqueda. Este componente es de gran importancia debido a sus posibilidades de filtrado y fácil acceso a la información. Permitirá gestionar solicitudes de los usuarios para ser insertados en un proyecto específico. La gestión de la información se realizará a través de una base de datos localizada en el servidor.

Una vez insertado el proyecto, el sistema permite realizar reportes estadísticos en dependencia de los filtrados seleccionados por el usuario, brindando el listado de la información requerida por el mismo. La utilización de reportes permitirá la toma de decisiones en aspectos que el administrador considere necesario. Los reportes se exportarán en formato pdf y podrán ser impresos.

Diagrama de Caso de Uso

A continuación, la figura 2.1 que muestra el caso de uso correspondiente.

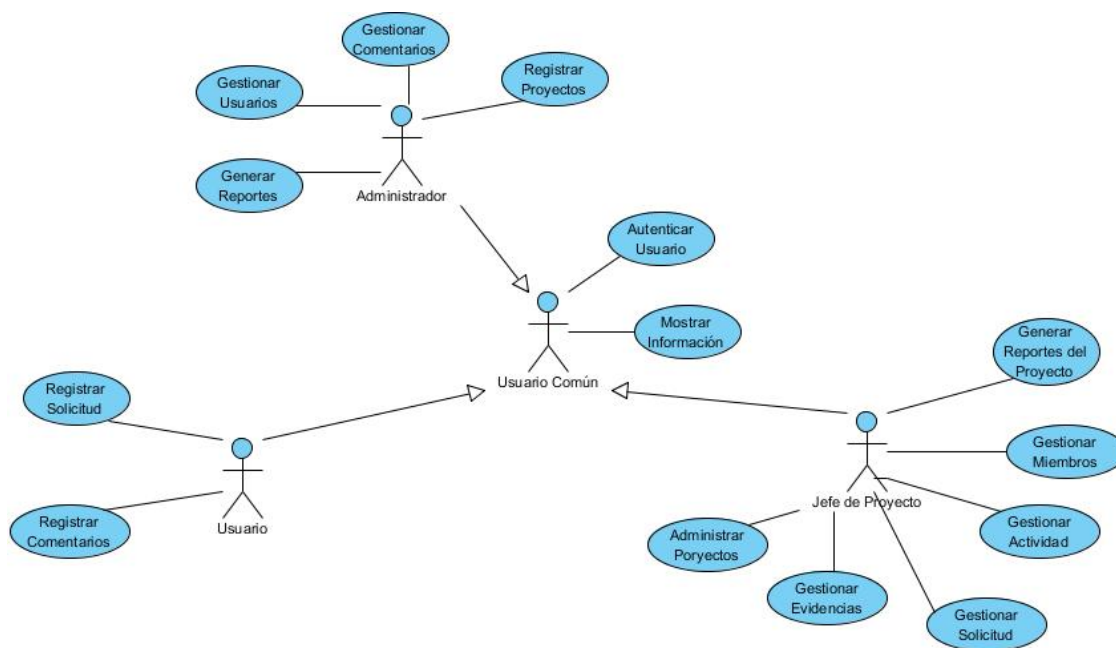


Figura 2.1. Diagrama de caso de uso. Fuente:(Elaboración propia)

2.2. Especificación de requisitos del sistema

Con el objetivo de que el sistema propuesto se desarrolle con las necesidades dictadas por el cliente final a partir del uso de la tormenta de ideas, se obtuvieron los requisitos funcionales y no funcionales.

2.2.1. Requisitos funcionales

Teniendo en cuenta las características de la propuesta en cuanto a prestaciones y operatividad se identifican 19 requisitos funcionales. A continuación, la tabla 2.1 muestra los requisitos del sistema.

Tabla 2.1. Descripción de los Requisitos Funcionales. (Fuente: Elaboración propia).

No.	Nombre	Prioridad	Complejidad
RF1	Autenticar usuario	alta	media
RF2	Solicitar proyecto	alta	alta
RF3	Modificar proyecto	media	media
RF4	Eliminar proyecto	alta	media
RF5	Listar proyectos	media	media
RF6	Crear participante	alta	media
RF7	Editar participante	media	baja
RF8	Eliminar participante	media	baja
RF9	Listar participante	media	baja
RF10	Asignar responsable	alta	media
RF11	Crear solicitud	alta	media
RF12	Modificar solicitud	baja	media
RF13	Eliminar solicitud	alta	media
RF14	Crear comentario	media	media
RF15	Modificar comentario	media	media
RF16	Eliminar comentario	media	media
RF17	Listar comentario	media	media
RF18	Buscar información	media	media
RF19	Generar pdf	media	media

Se determinó la complejidad de los requisitos regidos por lo definido en el expediente 5.0 de la evaluación de requisitos en la UCI.

2.2.2. Requisitos no funcionales

- Usabilidad.
 - RnF1. Tipo de aplicación informática: la herramienta debe ser web, y debe permitir adaptarse a dispositivos móviles.
 - RnF2. El sistema debe poseer una interfaz fácil de utilizar para cualquier tipo de usuario con conocimientos básicos de computación en el manejo de ordenadores, además de permitir realizar la búsqueda y el filtrado de cualquier proyecto extensionista para de esta forma facilitar la usabilidad.
- Confiabilidad.

- RnF3. El sistema debe ser tolerante a fallos, y mostrar solo la información necesaria para orientar al usuario.
- Hardware y software requerido para utilizar la aplicación.
 - RnF4. Sistema operativo: Podrá utilizar cualquier Sistema operativo.
 - RnF5. La propuesta de desarrollo debe ser capaz de ejecutarse en cualquier navegador web.
 - RnF6. La computadora que se utilice el cliente debe contar como mínimo con 2 GB de RAM y la que se utilice para el servidor debe contar como mínimo 500 GB de almacenamiento y como mínimo 4 GB de RAM.
- Eficiencia.
 - RnF7. El sistema debe permitir que los usuarios (100) interactúen con él de manera concurrente.
 - RnF8. El tiempo de demora de una petición al servidor debe ser menor a cinco (5) segundos.
- Seguridad.
 - RnF9. El acceso a la información debe estar restringido por usuario, contraseña y rol.
 - RnF10. El usuario podrá visualizar los proyectos extensionistas, una vez autenticado, tendrá los permisos pertinentes a su rol.
 - RnF11. Cuando un usuario se autentique en el sistema se le brindará la información correspondiente con su rol.
- Restricciones de diseño e implementación.
 - RnF12. El marco de trabajo que se utilizará está basado en el uso de herramientas no privativas como: JavaScript en su versión ECMAScript, Bootstrap en su versión 5, HTML y CSS, del lado del cliente, del lado del servidor Django en su versión 4.0.1 y Python en su versión 3.10.5, Microsoft Visual Studio Code v1.68 como herramienta para la codificación.
 - RnF13. Se utilizará como metodología de software la variación de AUP para la UCI .

2.3. Descripción de las historias de usuario

Para el encapsulamiento de los requisitos del software, tal como lo plantea la metodología AUP para la UCI , en su escenario cuatro, se generaron 19 HU, a continuación, se muestran las correspondientes a los requisitos funcionales “Solicitar proyecto”, “Autenticar usuario” y “Crear comentario” el resto se encuentra en el anexo.

Tabla 2.2. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Solicitar proyecto
Usuario: Usuario	

Continúa en la próxima página

**CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA PARA LA GESTIÓN DE LOS PROYECTOS
EXTENSIONISTAS DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

Tabla 2.2. Continuación de la página anterior

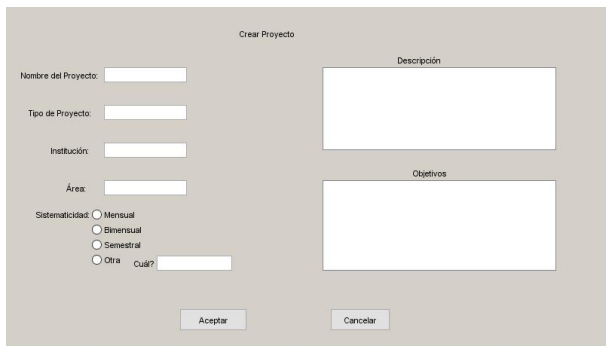
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: No.1
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir solicitar y crear un proyecto, basado en que hay que completar los siguientes campos, nombre del proyecto, área a la que pertenece, fecha de inicial, objetivos, sistematicidad.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz: 	

Tabla 2.3. Historia de usuario # 2

Historia de usuario	
Número: 2	Nombre: Autenticar usuario
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: No.2
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir la autenticación del usuario, completando los siguientes campos, usuario y contraseña. Además, en caso de que algún campo esté incompleto mostrar mensaje de campo vacío y por otra parte en caso de que el usuario o la contraseña estén incorrectos, mostrar mensaje de usuario o contraseña incorrectos	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	

Continúa en la próxima página

Tabla 2.3. Continuación de la página anterior

Interfaz:





Tabla 2.4. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Crear comentario
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 2	Iteración asignada: No.16
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir crear un comentario. Así como eliminarlo, según el permiso o por el propio usuario.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

2.4. Estilo arquitectónico

La arquitectura de software de un programa o de un sistema computacional está definida por la estructura, comprendida por los elementos de software, las propiedades visibles de esos elementos y las relaciones entre ellos. Especifica los roles, estructura, organización, reglas, comportamientos, componentes y capas de acuerdo a las características del sistema y las necesidades del cliente, tales restricciones y condiciones deben ya estar claramente identificadas en la etapa de levantamiento de requisitos (Corredor; Peña; Rodríguez et al., 2016).

Beneficios del uso de Estilos Arquitectónicos:

- Permite seleccionar una solución entendible y probada a ciertos problemas, definiendo los principios organizativos del sistema.
- Al basar la arquitectura en estilos que son conocidos las personas entienden más fácilmente las características importantes de la misma.
- Permiten evaluar arquitecturas alternativas con ventajas y desventajas conocidas ante diferentes conjuntos de requerimientos no funcionales.

Para el diseño de la propuesta de solución se utilizará el estilo Llamada y Retorno. Este estilo permite que los datos sean pasados como parámetros y el manejador principal proporciona un ciclo de control sobre las subrutinas.

2.4.1. Patrón arquitectónico

Un patrón es una solución de diseño para una serie de problemas, cada patrón está enfocado a resolver un problema determinado, con el objetivo de facilitar el trabajo de un desarrollador, concediendo una solución de un problema que se encuentra de forma habitual en el desarrollo de aplicaciones. Crear un patrón para resolver un problema particular no tiene objetivo general alguno, porque solo sería efectivo para ese problema (Pressman, 2016).

Los patrones arquitectónicos utilizados en la investigación son Modelo-Vista-Controlador el cual es la utilizada por el framework de desarrollo propuesto. Django es un framework Model,Template,View (MTV) es una modificación de Modelo-Vista-Controlador (MVC) debido a que los desarrolladores del framework no tuvieron la intención de seguir algún patrón de desarrollo, sino hacerlo lo más funcional posible (Moreno Montero, 2017).

La analogía con MVC de Django tiene la siguiente forma: .

- El modelo en Django continúa siendo Modelo
- La vista en Django se llama Plantilla
- El controlador en Django se llama Vista

A partir de lo planteado se asumirá el estilo arquitectónico MTV de Django del cual se explica su funcionamiento a continuación:

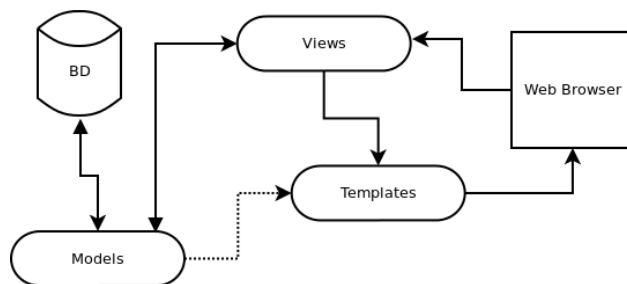


Figura 2.2. Patrón arquitectónico. Fuente:(Elaboración propia)

1. El navegador web envía una solicitud
2. La vista interactúa con el modelo para obtener los datos
3. La vista llama a la plantilla
4. La plantilla muestra la respuesta a la solicitud del navegado

Del funcionamiento del MTV de Django se deriva lo siguiente (Moreno Montero, 2017)

- Modelo: Define los datos almacenados, es representado en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, igualmente poseen métodos. Todo esto permite indicar y controlar el comportamiento de los datos.
- Vista: Su propósito es determinar qué datos serán visualizados, es representado en forma de funciones o también en clases de Python. El mapeo relacional de objetos de Django permite escribir código Python en lugar SQL para hacer las consultas. Aparte, se encarga de tareas como: el envío de correo electrónico, autenticación con servicios externos, así como la validación de datos a través de formularios.
- Plantilla: Recibe los datos que provienen de las vistas y luego organizarlos para la presentación al navegador web. Es básicamente una página HTML con algunas etiquetas extras que son propias de Django que permiten que sea más flexible para los desarrolladores del frontend.

2.5. Diagrama de clases de diseño

Una manera de modelar la vista lógica del sistema son los diagramas de clases de diseño, los cuales permiten, una representación de la estructura estática del sistema, a partir de la definición de los objetos que intervienen en el desarrollo de la propuesta de solución. La 2.3 muestra el diagrama.

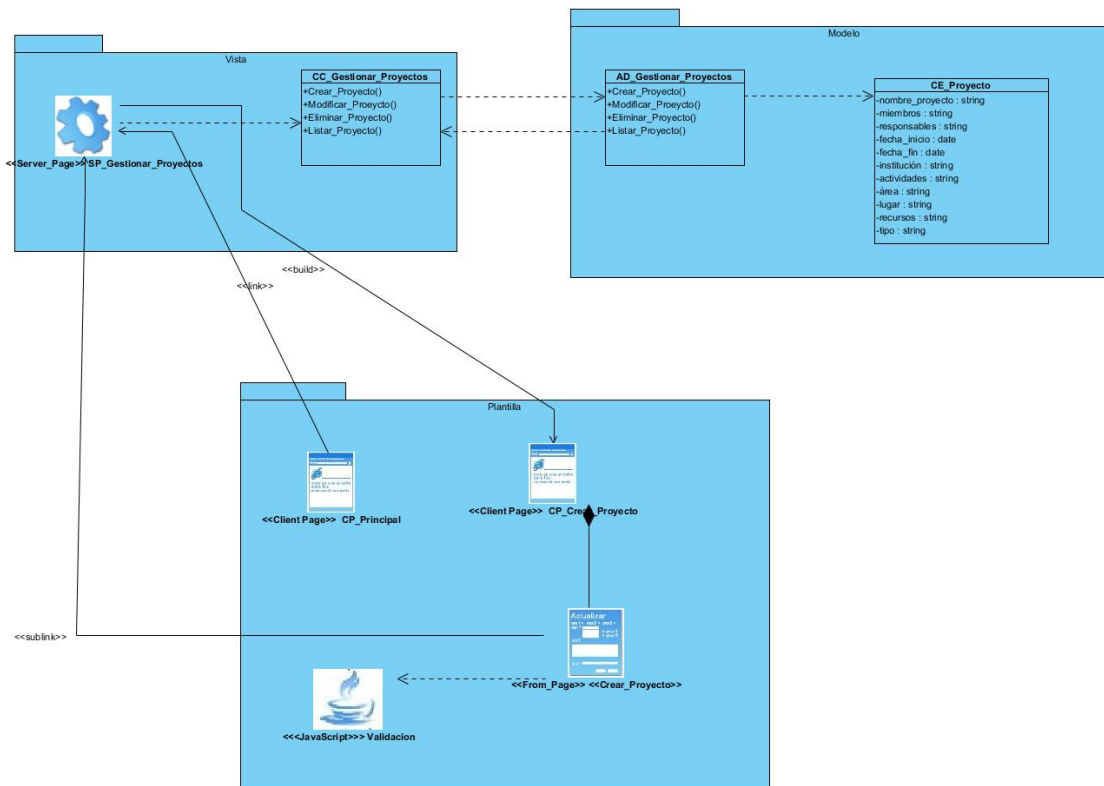


Figura 2.3. Diagrama de clases. Fuente:(Elaboración propia)

Descripción de las clases

- **ClientPage CPPrincipal:** Página principal que se encarga de mostrar los proyectos.
- **ClientPage CPCrearProyecto:** Página que está compuesta por el formulario para crear un nuevo proyecto.
- **FromPage CrearProyecto:** Formulario que contiene la ficha de información para crear un nuevo proyecto.
- **erverPage SPGestionarProyectos:** Gestiona los eventos del usuario y realiza la acción que le corresponde a cada uno de ellos.
- **CEProyecto:** Permite el acceso a la información almacenada de los proyectos.

2.6. Patrones de diseño

Un patrón de diseño describe una estructura que resuelve un problema particular del diseño dentro de un contexto específico. El objetivo de cada patrón de diseño es proporcionar una descripción que permita a un diseñador determinar si el patrón es aplicable al trabajo en cuestión, si puede ser reutilizado y si sirve como guía para desarrollar un patrón distinto en funciones o estructura (Pressman, 2016).

1. **Patrones generales de software de asignación de responsabilidad (GRASP, por sus siglas en inglés)** Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Corrales Sánchez, 2018).

Lo patrones GRASP empleados en la solución son:

- **Experto:** Permite la asignación de responsabilidades tales como la creación o inserción de datos, la eliminación y modificación de los mismos; y la implementación de métodos tales que ayuden al funcionamiento y fácil mantenimiento del sistema.
- **Creador:** Se utiliza a la hora de la inserción o modificación de los datos al añadir los elementos pasados por los formularios correspondientes. Además de permitir la creación de objetos y clases en las vistas y modelos.
- **Bajo acoplamiento:** Este patrón ya viene incluido con Django que permite un bajo acoplamiento entre las piezas, lo que evita las dependencias, por ejemplo, a la hora de realizar cambios en las configuraciones de las URL, en la BD y las plantillas HTML, basta solo con realizarlo una sola vez.
- **Alta Cohesión:** Asigna una responsabilidad de manera que la cohesión permanece alta. Este patrón incrementa la claridad y facilita la comprensión del diseño.
- **Controlador:** Asignar la responsabilidad de administrar un mensaje de evento del sistema a una clase que representa el sistema global, dispositivo, subsistema o representa un escenario de caso de uso en el que tiene lugar el evento del sistema.
Este patrón es empleado en todo el sistema debido a que cada uno de los eventos generados por el usuario es redirigido a una clase o función controladora que realiza las operaciones solicitadas, manteniendo siempre la alta cohesión.

2. **Pandilla de cuatro (GOF, por sus siglas en inglés)**, estos patrones se clasifican según su propósito en creacionales, estructurales y de comportamiento, mientras que respecto a su ámbito se clasifican en clases y objetos. Los empleados en la solución son:

- **Decorator (Decorador):** Permite agregar responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades (Y. Wang; Zhang y F. Wang, 2018). En Python, se puede emplear la función de decorador incorporado. Un decorador de Python es un cambio específico a la sintaxis de este lenguaje, que se utiliza para extender el comportamiento de una clase, método o función sin utilizar la herencia.
- **Observer (Observador):** Este patrón de diseño es uno de los patrones de diseño de software más populares. Ofrece la posibilidad de definir una dependencia uno a uno entre dos o más objetos para transmitir todos los cambios de un objeto concreto de la forma más sencilla y rápida posible. Para conseguirlo, puede registrarse en un objeto (observado) cualquier otro objeto, que funcionará como observador. El primer objeto, también llamado sujeto, informa a los observadores registrados cada vez que es modificado (ibíd.).

- **Flyweight (Peso mosca):** Este método es un patrón de diseño estructural que se enfoca en minimizar el número de objetos que requiere el programa en el tiempo de ejecución. Básicamente, crea un objeto Flyweight que es compartido por múltiples contextos. Se crea de tal manera que no se puede distinguir entre un objeto y un objeto Flyweight. Una característica importante de los objetos de peso mosca es que son inmutables. Esto significa que no se pueden modificar una vez que se han construido (Y. Wang; Zhang y F. Wang, 2018).

2.7. Modelo de datos

El modelado de datos es el proceso de analizar y definir todos los diferentes datos que su empresa recopila y produce, así como las relaciones entre esos bits de datos. El proceso de modelado de sus datos crea una representación visual de datos a medida que se utilizan en su negocio, y el propio proceso es un ejercicio de conocimiento y aclaración de sus requisitos de datos (Sikos y Philp, 2020).

La 2.4 muestra la definición de las principales tablas que componen la base de datos, así como sus relaciones entre ellas, donde se destaca como elemento fundamental la tabla proyectos, en función de la cual gira el resto de los elementos definidos en el modelo de datos.

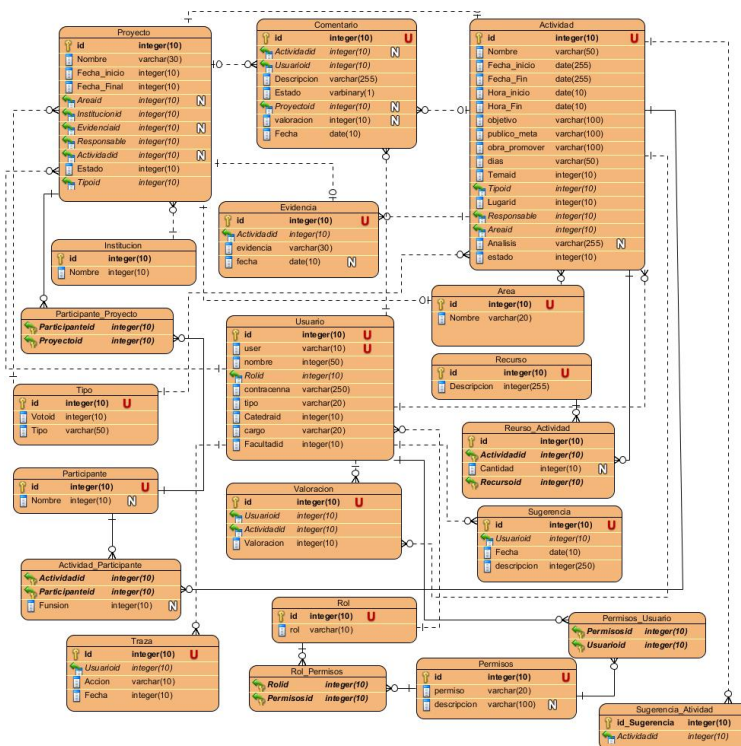


Figura 2.4. Modelo de datos. Fuente:(Elaboración propia)

Descripción de las tablas

1. **Proyectos:** Tabla fundamental del modelo de datos en la que se registran los proyectos con sus respectivas características.
2. **Área:** Tabla nomencladora de las áreas que pueden ser usadas para crear los proyectos.
3. **Tipo:** Tabla nomencladora de los tipos de proyectos que existen.
4. **Actividades:** Tabla nomencladora de las actividades que presenta un proyecto.
5. **Evidencias:** Tabla que recoge todas las evidencias de los proyectos, donde la relación que presenta es que un proyecto puede tener muchas evidencias, pero una evidencia es de un solo proyecto.
6. **Recursos:** Tabla nomencladora de los recursos que existen.
7. **Recursos Actividades:** Tabla que recoge los recursos de cada actividad.
8. **Participantes:** Tabla que recoge todos los participantes de los proyectos extensionistas.
9. **Actividad Participante:** Tabla que relaciona a las actividades con sus participantes.
10. **Participante Proyecto:** Tabla que relaciona a los proyectos con sus participantes, un proyecto puede tener varios participantes, y un participante puede estar en varios proyectos.
11. **Usuario:** Tabla que guarda los usuarios del sistema y permite la asignación de roles, permisos y la autenticación en el sistema.
12. **Rol:** Tabla nomencladora que guardan los roles que existen.
13. **Permisos:** Tabla que recoge los permisos que existen con su descripción correspondiente.
14. **Rol Permisos:** Tabla producto de la relación entre los roles y permisos, en el que cada rol se le corresponde varios permisos y cada permiso le corresponde varios roles.
15. **Permiso Usuario:** Tabla que establece relación entre los permisos con los usuarios directamente en el que a cada usuario le corresponde varios permisos y a cada permiso le corresponde varios usuarios.
16. **Valoración:** Tabla que recoge las valoraciones realizadas por cada proyecto.
17. **Comentarios:** Tabla que almacena los comentarios que se realiza a cada actividad.
18. **Sugerencias:** Tabla que recoge todas las sugerencias de los usuarios para la mejora del proceso.
19. **Sugerencia actividad:** Tabla nomencladora de las sugerencias perteneciente a una actividad.
20. **Institución:** Tabla que recoge las instituciones en las cuales se pueden realizar proyectos.
21. **Traza:** Tabla que almacena las trazas del sistema.

2.8. Consideraciones parciales

Luego de haber realizado el análisis y diseño del sistema propuesto y generar los artefactos que dispone la metodología AUP para la UCI en el escenario 4, se puede concluir lo siguiente:

- La descripción de la propuesta de solución a través del análisis de sus principales características, permitió identificar las funcionalidades del sistema propuesto.
- La especificación de los requisitos del sistema permitió describir cada uno de los requisitos funcionales a través de las historias de usuario, así como identificar los requisitos no funcionales de dicho sistema teniendo en cuenta un conjunto de características generales y restricciones del mismo.

- El establecimiento de la arquitectura del sistema a partir del patrón arquitectónico, facilitó aislarse del problema en que se enmarca la presente investigación, asegurando un entendimiento común entre las partes interesadas.
- La identificación de los patrones de diseño permitió disminuir el impacto de los cambios futuros en el código fuente de la misma.
- Las características abordadas sobre el diseño del sistema a través de la comprensión de cada elemento que lo componen, facilitó el enfoque en cuanto a composición lógica y física de la propuesta de solución.

Implementación y prueba del sistema para la gestión de los proyectos extensionistas de la Universidad de Ciencias Informáticas

En el presente capítulo se lleva a cabo las fases relacionadas con la implementación y las pruebas correspondientes de la propuesta de solución. La implementación ofrece una materialización precisa de los requisitos, mientras que el objetivo de las pruebas es comprobar si el sistema cumple con los requisitos. Dentro de la misma se pueden desarrollar varios tipos de pruebas en función de las metas a alcanzar. Para un mayor conocimiento del código se define un estándar de codificación. Así como se mostrarán las principales interfaces del sistema.

3.1. Estándar de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Se debe implementar un estándar de forma prudente, este debe tender siempre a lo práctico.

Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, es necesario establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego las revisiones del código.

Cada programador tiene su propia forma de escribir los códigos y puede ser completamente diferente a la de otros programadores, pero de la forma que se use depende la facilidad de que otros programadores entiendan el código y se les facilite su reutilización, de ahí se desprende la importancia de los estilos de programación, también conocidos como estándares o convenciones de código los cuales definen un grupo de acuerdos para escribir código fuente en ciertos lenguajes de programación (Testing, 2019).

A continuación, se define el estándar de codificación empleado en el sistema.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA PARA LA GESTIÓN DE LOS PROYECTOS EXTENSIONISTAS DE LA UNIVERSIDAD DE CIENCIAS INFORMÁTICAS

Tabla 3.1. Estándares de codificación a utilizar. (Fuente: Elaboración propia).

Tipo de estándar	Descripción
Organización del código	<ul style="list-style-type: none">• El código en una página se organizará por bloques• La indentación se realizará solamente con tabulaciones, no debe utilizarse nunca los cuatro (4) espacios
Máxima longitud de las líneas	<ul style="list-style-type: none">• Todas las líneas deben estar limitadas a un máximo de setenta y nueve (79) caracteres• Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas.
Líneas en blanco	<ul style="list-style-type: none">• Separar las funciones de alto nivel y definiciones de clases con dos (2) líneas en blanco• Las definiciones de métodos dentro de una clase deben separarse por una (1) línea en blanco• Se puede utilizar líneas en blanco escasamente para separar secciones lógicas
Importación	<ul style="list-style-type: none">• Las importaciones deben estar en líneas separadas• Siempre deben colocarse al comienzo del archivo• Deben quedar agrupadas de la siguiente forma:<ul style="list-style-type: none">○ Importaciones de la librería estándar○ Importaciones terceras relacionadas○ Importaciones locales de la aplicación / librerías• Cada grupo de importaciones debe estar separado por una línea en blanco
Continúa en la siguiente página	

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA PARA LA GESTIÓN DE LOS PROYECTOS EXTENSIONISTAS DE LA UNIVERSIDAD DE CIENCIAS INFORMÁTICAS

Tipo de estándar	Descripción
Espacios en blanco en expresiones y sentencias	<ul style="list-style-type: none"> ● Evitar utilizar espacios en blanco en las siguientes situaciones: <ul style="list-style-type: none"> ○ Inmediatamente dentro de paréntesis, corchetes y llaves ○ Inmediatamente antes de una coma, un punto y coma o dos puntos ○ Inmediatamente antes del paréntesis que comienza la lista de argumentos en la llamada a una función ○ Inmediatamente antes de un corchete que empieza una indexación ○ Más de un espacio alrededor de un operador de asignación (u otro) para alinearlos con otro ● Deben rodearse con exactamente un espacio los siguientes operadores binarios: <ul style="list-style-type: none"> ○ Asignación (=) ○ Asignación de aumentación (+=, -=, *=, /=) ○ Comparación (==, <, >, >=, <=, =, <>, in, not in, is, is not) ○ Expresiones lógicas (and, or, not) ● Si se utilizan operadores con prioridad diferente se aconseja rodear con espacios a los operadores de menor prioridad ● No utilizar espacios alrededor del igual (=) cuando es utilizado para indicar un argumento de una función o un parámetro con un valor por defecto
Comentarios	<ul style="list-style-type: none"> ● Los comentarios deben ser oraciones completas ● Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula ● Si un comentario es corto el punto final puede omitirse ● Los comentarios de una línea para aclaraciones del código aparecerán seguidos de los caracteres “//” en caso de código JavaScript mientras que en Python por el carácter “#” y deben ubicarse en la misma línea que se desea comentar
Continúa en la siguiente página	

Tipo de estándar	Descripción
Convenciones de nombramiento	<ul style="list-style-type: none"> • Nunca se deben utilizar como simples caracteres para nombres de variables los caracteres ele minúscula “l”, o mayúscula “O”, ele mayúscula “L”, ya que en algunas fuentes son indistinguibles de los números uno (1) y cero (0) • Los módulos deben tener un nombre corto y en minúscula. • Los nombres de clases deben utilizar la convención “CapWords” (palabras que comienzan con mayúsculas) • Los nombres de las funciones deben estar escrito en minúscula separando las palabras con un guion bajo “_” • Las constantes deben quedar escritas con letras mayúsculas separando las palabras por un guion bajo (_)
Codificaciones	<ul style="list-style-type: none"> • Utilizar la codificación UTF-8

A continuación se muestra un ejemplo del código fuente, donde es evidente el uso del estándar de codificación definido en la tabla anterior.

Código fuente 3.1. Uso de la codificación

```

1 def getproyectos(request):
2     proyectos= Proyectos.objects.filter( activo='True', estado='Finalizado')
3
4     return render(request, "Proyectos/listar/proyectos.html", {"lista_objetos":
5         proyectos})
6
7 def proyecto_detalle(request, pk):
8     form_class = Participante
9     template_name = 'Proyectos/formulario/participante_formulario.html'
10    form = form_class(initial={'proyecto': pk,})
11    objeto = Proyectos.objects.all().get(pk=pk)
12    participantes= Proyecto_Usuario.objects.filter(proyecto=pk).values("
13        participante__first_name", "participante__last_name", "participante_id")
14    return render(request, "Proyectos/detalles/proyectos_detalle.html", {"
15        proyecto": objeto , "template_name":form, "participantes": participantes
16        ,})
17
18 def formulario_proyecto (request):
19     form_pro = proyectos.Proyectos.objects.all()
20     return render(request, "Proyectos/formulario_proyecto.html", {"proyecto":
21         form_pro})
22

```

```
18
19 class Formulario_Proyecto(View):
20     form_class = ProyectoForm
21     form_participante= Participante
22     template_name = 'Proyectos/formulario/formulario_p.html'
23     success = 'crear_proyectos'
24
25     def get(self, request):
26         form_participante = self.form_participante()
27         form = self.form_class(initial={'estado': "solicitado", '
28             solicitante_proyecto': request.user.carnet})
29         return render(request, self.template_name, {"form": form, "
30             form_participante": form_participante, "solicitante_proyecto": request.
31             user.carnet})
32
33     def Crear_Proyectos(request):
34         if request.method == "POST":
35             form_nuevo=ProyectoForm(request.POST, request.FILES)
36             if form_nuevo.is_valid():
37                 form_nuevo.save()
38                 return redirect("proyectos")
39             else:
40                 return render(request, "Proyectos/formulario/formulario_p.html" , {"form":
41                     form_nuevo})
```

3.2. Diagrama de despliegue

Para lograr una representación de la distribución física del sistema para la gestión de los proyectos extensionistas para la Dirección de Extensión Universitaria, se muestra en la figura 3.1 el diagrama de despliegue correspondiente.

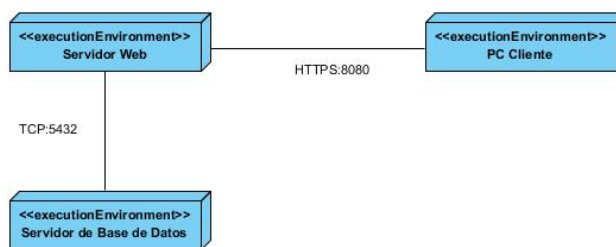


Figura 3.1. Diagrama de despliegue

Este diagrama se considera para lograr un despliegue exitoso de la aplicación y se definen:

- **Dispositivo cliente:** Se refiere a las estaciones de trabajo(PC Cliente) que el usuario utilizará para conectarse, vía HTTPS, con el servidor de aplicaciones utilizando para ello el puerto 8080.

- **Servidor de bases de datos:** Almacena toda la información que brinda el sistema. La información es obtenida o modificada en dependencia del nivel de privilegio del usuario que realiza la petición. La comunicación con el servidor de aplicaciones es a través del protocolo TCP empleando el puerto 5432.
- **Servidor de aplicaciones:** Es el encargado de brindar la interfaz del sistema para que los usuarios puedan hacer uso de este, almacena todo el código fuente del sistema y se comunica por medio del protocolo TCP con el servidor de bases de datos.

3.3. Pruebas de software

El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores. Además, son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2016).

En este epígrafe se muestran los resultados de la estrategia de prueba diseñada para la propuesta de solución (ver Tabla 3.2), en función de garantizar y validar la calidad de este.

3.3.1. Estrategia de pruebas

Tabla 3.2. Estrategia de pruebas. (Fuente: Elaboración propia).

Pruebas	Método	Herramienta	Alcance
Funcional	Caja negra con particiones equivalentes	Selenium	Se probará el funcionamiento del 100 % de los requisitos.
Unitaria	Caja blanca con la técnica del camino básico	Módulo TestCase de Django para la realización de pruebas automatizadas	Se automatizarán pruebas para las unidades de código separadas por módulos.
Rendimiento	Pruebas de carga y estrés	Locust	Se aplicará sobre un entorno de pruebas con prestaciones menores que el de despliegue, debido a falta de recursos. Se probará la aplicación con 100 usuarios concurrentes buscando tiempos de respuesta menores a 5 segundos.
Continúa en la siguiente página			

Pruebas	Método	Herramienta	Alcance
Seguridad		Acunetix web vulnerability scanner 9.5	Se aplicará para detectar vulnerabilidades: <ul style="list-style-type: none"> • Inyección SQL • Programación Cross-Site (o XSS) • Ataques de fuerza bruta a las credenciales • Redirecciones y reenvíos no validados
Aceptación	Pruebas Alpha y beta	Cuestionario	Se le aplicará un cuestionario de preguntas a un grupo de especialistas y trabajadores con años de experiencia en el proceso de promoción de la cultura. El éxito de la prueba se medirá: <ul style="list-style-type: none"> • Exitosa más del 75 % de las preguntas tienen resultados positivos. • Fracaso menos del 75 % de las preguntas tienen resultados positivos.
Ejecutado por:	Dariel Torres Alvarez		

3.3.2. Pruebas unitarias

Método de caja blanca

El método de caja blanca tiene como objetivo principal probar el código. Entre las técnicas existentes de caja blanca se encuentran: prueba del camino básico, prueba de condición, prueba de flujo de datos y prueba de bucles. Para el presente trabajo se utiliza la prueba del camino básico, la misma permite: generar el grafo de flujo, calcular la Complejidad ciclomática, determinar los caminos linealmente independientes y diseñar los casos de prueba para forzar la ejecución de cada camino del conjunto básico.

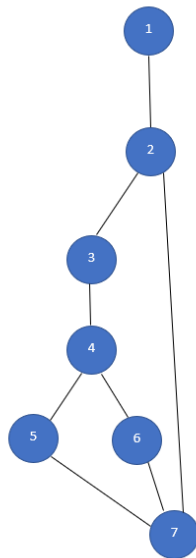
La técnica del camino básico fue aplicada al método Crear_Proyectos, que se encarga de registrar un nuevo proyecto al sistema.

Tabla 3.3. Cálculo de la complejidad ciclomática del método Crear_Proyectos

Método

```
def Crear_Proyectos(request):
    if request.method == "POST":
        form_nuevo=ProyectoForm(request.POST, request.FILES)
        if form_nuevo.is_valid():
            form_nuevo.save()
            return redirect("proyectos")
        else:
            return render(request, "Proyectos/formulario/formulario_p.html" , {"form": form_nuevo})
```

Grafo resultante:



Complejidad Ciclomática:	$V(G) = A - N + 2$	$V(G) = P + 1$
$V(G) = \# \text{ de regiones}$	$V(G) = 8 - 7 + 2$	$V(G) = 2 + 1$
$V(G) = 3$	$V(G) = 3$	$V(G) = 3$

Luego de la determinación de los nodos y flujos de control del código se obtuvo el grafo de flujo y se calculó la complejidad ciclomática del algoritmo.

Como resultado se obtuvo que la complejidad ciclomática es igual a 3, lo que significa que existen tres posibles caminos linealmente independientes y hay que diseñar un mínimo de tres casos de prueba para el algoritmo Crear_Proyectos. La tabla 3.4 muestra los caminos existentes.

Tabla 3.4. Caminos del grafo de flujo (Fuente: Elaboración propia).

No.	Camino
1	1,2,3,4,5,7

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA PARA LA GESTIÓN DE LOS PROYECTOS EXTENSIONISTAS DE LA UNIVERSIDAD DE CIENCIAS INFORMÁTICAS

2	1,2,3,4,6,7
3	1,2,7

Los casos de prueba para las pruebas de caja blanca por la técnica de camino básico se ejecutan por cada camino independiente que se determine en un algoritmo específico. A continuación, se muestra el caso de prueba para el camino básico independiente 1 del algoritmo.

Tabla 3.5. Caso de Prueba para el camino básico 1 (Fuente: Elaboración propia).

Proceso:
Crear Proyecto.
Casos de prueba:
Crear Proyecto. Escenario 1.1.
Camino independiente:
1,2,3,4,5,7
Entradas:
<ul style="list-style-type: none"> • Nombre de proyecto: Programando en Comunidad • Responsable de Proyecto: yami7870 • Contactos: yami7870@gmail.com • Participantes: por definir • Fundamentación: Es necesario para desarrollar habilidades de programación en las comunidades cercanas al centro • Objetivo General: Impartir Conocimientos básicos de programación en la comunidad • Objetivos Específicos: Desarrollar habilidades de programación en comunidad • Area: Facultad 1 • Fecha: 20-10-22 • Tipo: Comunitario • Público Meta: Comunidad extrauniversitaria • Sistematicidad: Mensual • Propuesta para evaluación de Impacto: buena propuesta a presentar el desarrollo de la programación en la comunidad
Resultados esperados:
<ul style="list-style-type: none"> • Crear y solicitar un nuevo Proyecto.
Condiciones de ejecución:

- Usuario autenticado.
- Usuario con rol de administrador.

Con la realización de los casos de prueba diseñados se probó la ejecución de cada sentencia del código al menos una vez, teniendo en cuenta todas las condiciones lógicas en sus variantes verdaderas y falsas. Los resultados del método de caja blanca fueron satisfactorios. La obtención de la complejidad ciclomática de valor tres permitió determinar que el algoritmo es de baja complejidad.

3.3.3. Pruebas funcionales

La prueba funcional se centra en comprobar que los sistemas desarrollados funcionan acorde a las especificaciones funcionales y requisitos del cliente, con el objetivo de validar que las funcionalidades implementadas funcionen correctamente y cumplan con los requisitos definidos con anterioridad. Se realizan para comprobar si el software cumple las funciones esperadas.

Las pruebas funcionales se basan en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. El objetivo principal de las pruebas funcionales es analizar el producto terminado y determinar si hace todo lo que debería hacer y si lo hace correctamente. El método utilizado para llevar a cabo este tipo de prueba es caja negra (Pressman, 2016)

Método de caja negra

Es un método de prueba de software que analiza la funcionalidad de un software o aplicación sin saber mucho sobre la estructura o diseño interno del elemento que se está probando y compara el valor de entrada con el valor de salida (ibíd.). El método de caja negra presenta varias técnicas de prueba como son: partición de equivalencia, análisis de valores límites y grafos de causa-efecto. El enfoque principal de método de Caja Negra es la funcionalidad del sistema en su conjunto. El término “Pruebas de comportamiento” también se utiliza para pruebas de caja negra. El diseño de prueba de comportamiento es ligeramente diferente del diseño de prueba de caja negra porque el uso del conocimiento interno no está estrictamente prohibido.

En la presente investigación se utilizará específicamente dentro del método de caja negra la técnica de partición de equivalencia generando los casos de pruebas de dicha técnica sobre las diferentes interfaces que responden a los requisitos funcionales.

A continuación, la tabla 3.6 muestra el diseño de caso de pruebas del requisito “Autenticar usuario” donde se analizarán las variables y condiciones que puedan determinar la respuesta del sistema.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA PARA LA GESTIÓN DE LOS PROYECTOS EXTENSIONISTAS DE LA UNIVERSIDAD DE CIENCIAS INFORMÁTICAS

Tabla 3.6. Caso de prueba “Autenticar usuario” (Fuente: Elaboración Propia).

Escenario	Descripción	Variables		Respuesta esperada	Respuesta
		Usuario	Contraseña		
EC 1.1. Autenticar usuario correctamente	El usuario debe introducir los datos de los campos mostrados y dar clic a en botón “Autenticar”	V admin	V Admin787	Se accede al sistema, se selecciona el botón “Entrar” se inserta el usuario y contraseña y se muestra un mensaje de confirmación de la autenticación y se accede al sistema	El sistema autentica correctamente y muestra un mensaje de confirmación
EC 1.2. Autenticar usuario incorrectamente		V Admin	I Administrador	Se accede al sistema, se selecciona el botón “Entrar” se inserta el usuario y contraseña y se muestra “Usuario o contraseña incorrecto” no se accede al sistema	El sistema muestra un mensaje con la negación del usuario autenticado y especificando que el usuario o contraseña están incorrecto

No.	Variable	Valor Nulo	Descripción
1	usuario	No	Es un campo que permite al usuario insertar su usuario correctamente. Acepta solo letras y no acepta espacio entre letras
2	Contraseña	No	Es un campo que permite al usuario insertar su contraseña correctamente. Acepta cualquier carácter alfa numérico.

Tabla 3.7. Variables de caso de prueba “Autenticar usuario”(Fuente: Elaboración Propia).

Las pruebas de caja negra se aplicaron con el objetivo de evaluar las interfaces de comunicación con el usuario, las que demostraron coherencia y funcionalidad. La técnica de partición de equivalencia es aplicada para evaluar los diferentes escenarios que pueden tener lugar ante la ejecución de una acción. Como resultado de

la aplicación de estas pruebas se ejecutan las posibles variantes que posee una interfaz de comunicación con el usuario, resolviendo las no conformidades arrojadas y perfeccionando lo obtenido. Este tipo de pruebas se aplica de conjunto con otras, como por ejemplo las pruebas de aceptación.

3.4. Resultados de las pruebas

Durante la realización de las pruebas se detectaron un conjunto de no conformidades relacionadas con errores de validación y funcionalidad. Los resultados se muestran en la figura 3.2, donde se evidencia la cantidad de casos de prueba ejecutados, los casos de prueba con no conformidades y las no conformidades identificadas, las cuales fueron corregidas.

Se realizaron tres iteraciones, durante la primera iteración se analizaron 37 casos de prueba, de los cuales 12 resultaron no conformidades. En la segunda iteración a través de las pruebas de regresión se verificó que las no conformidades anteriores estuviesen solucionadas de estas pruebas se obtuvieron 5 nuevas no conformidades, quedando resuelta en la tercera iteración y cumpliéndose correctamente los requisitos funcionales.

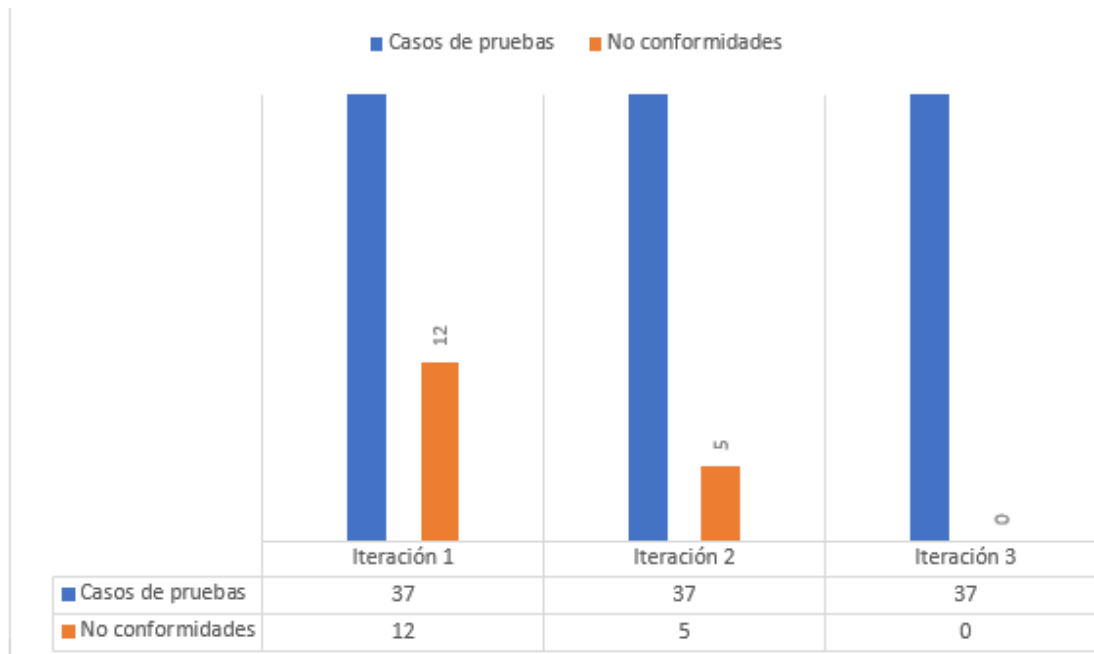


Figura 3.2. Resultado de Pruebas. Fuente:(Elaboración propia)

3.4.1. Prueba de rendimiento (Carga y Estrés)

Las pruebas de rendimiento se realizan para determinar el rendimiento del software correspondiente en tiempo de corrida, dentro del contexto de un sistema integrado de rendimiento de software. Se realizan para medir la respuesta de la aplicación a distintos volúmenes de carga esperados, se enfoca en determinar la

velocidad con la que el sistema bajo pruebas, realiza una tarea en las condiciones particulares del escenario de pruebas (Pressman, 2016).

- **Carga:** Determina y valida la respuesta de la aplicación cuando es sometida a una carga de usuarios y/o transacciones que se esperan en el ambiente de producción. Estas pruebas consisten en la simulación de una carga de trabajo similar y superior a la que tendrá cuando el sitio se encuentre en funcionamiento, con el objetivo de detectar si el software instalado cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware (servidor y el equipamiento computacional de redes y enlace que lo conecta a Internet) es capaz de soportar la cantidad de visitas esperadas (ibíd.).
- **Estrés:** Encuentra el volumen de datos o de tiempo en que la aplicación comienza a fallar o es incapaz de responder a las peticiones. Son pruebas de carga o rendimiento, pero superando los límites esperados en el ambiente de producción y/o determinados en las pruebas. Las pruebas de estrés evalúan la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas (ibíd.).

Para realizar las pruebas de rendimiento se utiliza el software Apache Jmeter v5.5. Para esto, se definen las propiedades de las PC utilizadas tanto la cliente como la utilizada como servidor.

Hardware de prueba (PC cliente):

- Tipo de procesador: Intel(R) Dual Core(R) Celeron(R) n3060 CPU @2.00GHz 2.00GHz.
- RAM: 4 GB DDR3.
- Tipo de Red: Ethernet 10/100Mbps.

Hardware de prueba (PC servidor):

- Tipo de procesador: Intel(R) Core(TM) i36006U CPU @2.00GHz 2.00GHz.
- RAM: 8 GB DDR3.
- Tipo de Red: Ethernet 10/100Mbps.

Software instalado en ambas PC:

- Tipo de servidor web: Apache.
- Plataforma: SO Windows (PC servidor) y SO Windows (PC cliente).
- Servidor de BD: PostgreSQL

Para las pruebas de carga en el sistema se realizó la prueba con 100 usuarios concurrentes. Como se muestra en la figura 3.3 se obtuvo un tiempo promedio de respuesta del servidor de 3.1 por segundo cumpliendo satisfactoriamente dicho requisito no funcional.

Etiqueta ^	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
proyectos	200	2967	601	7770	1336.72	62.50%	3.1/sec	20.93	0.16	6875.8
Total	200	2967	601	7770	1336.72	62.50%	3.1/sec	20.93	0.16	6875.8

Figura 3.3. Prueba de rendimiento

Para la prueba de estrés como se muestra en la figura 3.4 se obtuvo para 200 usuarios concurrentes un tiempo de respuesta promedio mayor que 5 segundos superando el tiempo establecido.

Etiqueta ^	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
proyectos	200	2734	852	7210	1099.69	78.00%	19.0/sec	95.53	0.58	5151.8
Total	200	2734	852	7210	1099.69	78.00%	19.0/sec	95.53	0.58	5151.8

Figura 3.4. Prueba de rendimiento

3.4.2. Pruebas de seguridad

Garantiza que el software y las aplicaciones web estén libres de lagunas, vulnerabilidades, amenazas, riesgos que puedan causar una gran pérdida a la empresa / organización, y verifique si sus datos y recursos están protegidos de posibles intrusos. Las pruebas de seguridad se tratan de encontrar lagunas o ataques imprevistos al sistema que podrían resultar en la pérdida de datos. El propósito principal de las pruebas de seguridad es detectar vulnerabilidades y luego repararlas. Ayuda a impulsar el sistema actual y asegurarse de que el sistema pueda funcionar durante un tiempo prolongado. Para notar lagunas que provocarán la pérdida de información vital (Pressman, 2016).

La prueba realizada utilizando la herramienta Acunetix Web Vulnerability Scanner v9.5, mostró que en la primera iteración realizada se detectaron 62 no conformidades, de las cuales 55 alertas fueron informativas, 4 alertas de riesgo medio y 3 alertas de riesgo bajo.

La 3.5 muestra los resultados lanzados en la primera iteración

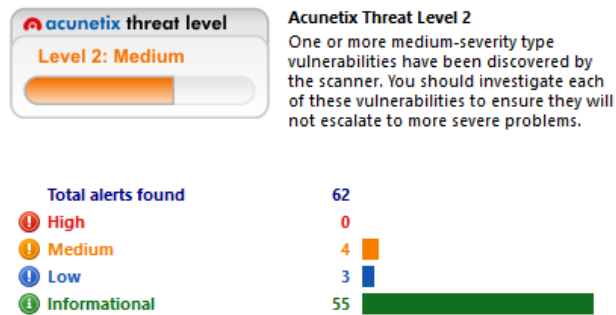


Figura 3.5. Acunetix Web Vulnerability Scanner Primera iteración

En la segunda iteración no se encontraron vulnerabilidades, debido a esto, los problemas que pudieran poner en riesgo la integridad y seguridad del sistema detectados en la primera iteración fueron resueltos. Esto se puede evidenciar en la 3.6

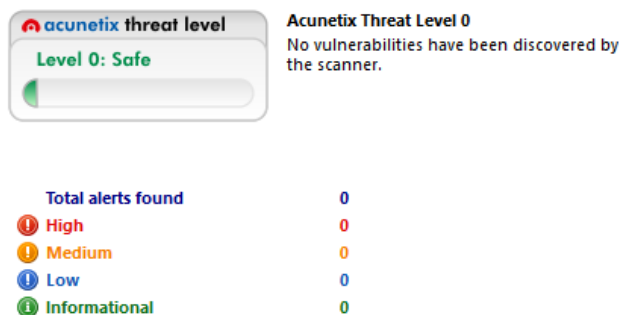


Figura 3.6. Acunetix Web Vulnerability Scanner segunda iteración

3.4.3. Pruebas aceptación

Para la revisión final de las especificaciones del diseño y de la implementación se realizaron las pruebas de aceptación con el objetivo de determinar diferentes tipos de errores y resolverlos (Pressman, 2016), para ello se emplean las pruebas Alfa y Beta.

Pruebas Alfa: Se desarrollan en conjunto, el desarrollador y los usuarios finales. Con el objetivo de registrar los errores y problemas del uso del software.

Pruebas Beta: Se realiza en el sitio final donde se va a desplegar el software. A diferencia de la prueba alfa, la prueba beta es una aplicación del software en su ambiente final. El objetivo es registrar todos los problemas que se encuentran durante la prueba.

Para la aplicación de las pruebas de aceptación se seleccionó un grupo de profesores y especialistas que han impartido durante 6 años diversos cursos en EU de la UCI. A continuación, se relacionan los involucrados en las pruebas:

- MSc. Julio César Espronceda Pérez (Director de Extensión Universitaria)
- Ing. Alvaro Alejandro Acosta Ruiz (Especialista B en Ciencias Informáticas)
- Lic. Jesús Ramón González Crespo (Especialista Superior)
- Lic. Madelaine Bárbara Labadié Martínez (Especialista General)
- MSc. Serguey González Garay (Vicedecano de Extensión)
- Lic. Berta Zelaida López Pérez (Especialista Superior)
- Lic. Yanaida Delgado González (Especialista Superior)
- MSc. Mirta Beltrandez Sardiñas (Profesor)
- Ing. Elianne Vizcay Álvarez (Especialista B en Ciencias Informáticas)
- Ing. Jessie de la Caridad García Rojas (Especialista B en Ciencias Informáticas)

3.5. Resultados de la solución implementada

Con el objetivo de evaluar el sistema implementado se utiliza la técnica de Iadov, esta técnica evalúa el nivel de satisfacción del usuario, permitiendo conocer si la solución propuesta cumple con las expectativas

esperadas. Esta técnica constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas (preguntas 1, 2 y 3) que se intercalan dentro de un cuestionario A.1. Estas tres preguntas se relacionan a través de lo que se denomina el “Cuadro Lógico de Iadov”, el cual se muestra a continuación en la tabla 3.8.

Tabla 3.8. Cuadro Lógico de Iadov (Fuente: Elaboración propia).

	1- ¿Se pueden gestionar correctamente los proyectos extensionistas sin la documentación requerida?								
	NO			NO SÉ			SÍ		
3- ¿Satisface sus necesidades a la hora de gestionar los proyectos extensionistas?	2- ¿Utilizaría usted la aplicación propuesta para gestionar los proyectos extensionistas?								
	SÍ	NO SÉ	NO	SÍ	NO SÉ	NO	SÍ	NO SÉ	NO
Me satisface mucho	1	2	6	2	3	4	3	5	6
No me satisface tanto	5	3	2	3	2	5	2	3	6
Me da lo mismo.	3	6	5	3	6	3	3	4	5
Me disgusta más de lo que me satisface.	5	3	6	3	5	4	3	4	6
No me satisface nada.	6	6	6	6	4	4	6	4	5
No sé qué decir.	3	3	5	3	3	3	6	3	6

El número resultante de la interrelación de las tres preguntas indica la posición de cada sujeto en la escala de satisfacción.

Escala de satisfacción:

- Clara satisfacción (1).
- Más satisfecho que insatisfecho (2).
- No definida (3).
- Más insatisfecho que satisfecho (4).
- Clara insatisfacción (5).
- Contradictoria (6).

Para medir el grado de satisfacción se tomó una muestra de 10 personas, todas ellas con más de 5 años de experiencias en el trabajo con proyectos extensionistas. De ellos el 10 % son Doctores en Ciencias, el 30 % son Máster es Ciencia, el 50 % son Ingenieros y el resto son Licenciados.

La técnica de Iadov permite conocer el Índice satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en una escala numérica que oscila entre +1 y -1 de la siguiente forma:

Índice	Escala
+1	Máximo de satisfacción.
0.5	Más satisfecho que insatisfecho.
0	No definido y contradictorio.
-0.5	Más insatisfecho que satisfecho.
-1	Máxima insatisfacción.

La satisfacción grupal se calcula utilizando la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

Donde:

- A, B, C, D, E representan el número de sujetos con índice individual 1, 2, 3 o 6, 4, 5 respectivamente.
- N representa el número total de sujetos del grupo.

Esto permite reconocer las categorías grupales:

- Insatisfacción: desde (-1) hasta (-0,5)
- Contradictorio: desde (-0,49) hasta (+0,49)
- Satisfacción: desde (+0,5) hasta (1).

$$ISG = \frac{8(+1) + 1(+0,5) + 1(0) + 0(-0,5) + 0(-1)}{10} = 0.85$$

Luego de haber aplicado la técnica, calculando el ISG se obtiene como resultado 0.85 lo que significa una clara satisfacción con el uso del sistema de gestión para los proyectos extensionistas.

3.6. Validación de la hipótesis

Partiendo de la operacionalización de las variables precisadas en la introducción de la investigación (variables dependientes e independientes) se concreta que en la variable independiente, de acuerdo a sus dimensiones y las pruebas asociadas, en cuanto a usabilidad, luego de aplicadas las pruebas de alfa y beta y su respuesta satisfactoria, se evidencia dicha usabilidad del sistema y en la dimensión de funcionalidad, luego de analizadas, mediante los métodos de caja negra y caja blanca, según sus resultados satisfactorios, se evidencia la funcionalidad del sistema y a partir de la aplicación de la técnica de Iadov que validó la satisfacción de los usuarios, se demuestra la validación de la variable dependiente y que el desarrollo del sistema propuesto sí contribuye al proceso de gestión de los proyectos extensionistas en la Universidad de las Ciencias Informáticas.

3.7. Consideraciones parciales

En este capítulo se abordaron los elementos del sistema para la gestión de los proyectos extensionistas, así como las pruebas realizadas al mismo y los resultados obtenidos; lo que permite concluir:

- La implementación de la propuesta de solución facilitó obtener una aplicación funcional capaz de contribuir con la gestión de los proyectos extensionistas
- La validación de la propuesta de solución, mediante una estrategia de pruebas de software facilitó corroborar la calidad de la misma, estableciendo así el cumplimiento de los requisitos definidos por el cliente.
- La aplicación de la técnica de Iadov arrojó como resultado un alto índice de satisfacción de la solución propuesta por parte de un grupo de profesores y especialistas de la EU en la UCI, lo cual se traduce en una clara satisfacción con el uso del sistema informático propuesto.

Con el desarrollo de la presente investigación se obtuvo un sistema informático para la gestión de los proyectos extensionistas en la Universidad de las Ciencias Informáticas, por lo que se concluye que:

- El estudio asociado al proceso de gestión de proyectos extensionistas, permitió definir que la propuesta de solución deberá ser de tipo web.
- El análisis de los sistemas homólogos permitió identificar las principales funcionalidades que debe tener el sistema a desarrollar.
- Las herramientas y tecnologías a utilizar asegura el ambiente de desarrollo necesario para la implementación de la propuesta de solución, destacando como lenguajes de programación HTML5, CSS3, Python, Javascript y como framework de desarrollo Django.
- La descripción de la propuesta de solución mediante la investigación y análisis de sus particularidades fundamentales, permitió identificar el carácter funcional del sistema propuesto, acotado por las necesidades del cliente para colaborar a la gestión los proyectos extensionistas de la Universidad de las Ciencias Informáticas.
- La validación de la propuesta de solución, mediante una estrategia de pruebas de software facilitó corroborar la calidad de la misma, estableciendo así el cumplimiento de los requisitos definidos por el cliente.

Recomendaciones

Para dar continuidad a la presente investigación se recomienda:

- Llevar a cabo la integración con otros subprocesos de la Extensión universitaria para lograr así un mayor alcance de infomatizar dicho proceso.

- AJAX** JavaScript asíncrono y Lenguaje de Marcado Extensible. 21
- AUP** Proceso unificado ágil. 18, 28, 36
- CASE** Ingeniería de software asistida por computadora. 18
- CSS** Hoja de estilos en cascada. 20, 21
- DEU** Dirección de extensión universitaria. 9
- DOM** Modelo de Objetos del Documento. 21
- EU** Extensión universitaria. 6, 7, 9, 10, 17, 50, 54
- GESPRO** Paquete para la gestión de proyectos. 16, 17, 59
- GOF** Pandilla de cuatro. 33
- GRASP** Patrones generales de software de asignación de responsabilidad. 33
- HTML** Lenguaje de marcas de hipertexto. 20, 21, 32, 33
- HU** Historia de usuario. 18, 28
- IDE** Entorno de desarrollo integrado. 18, 19
- IES** Instituciones de Educación Superior. 1, 2, 6, 14
- ISG** Índice satisfacción grupal. 53
- MES** Ministerio de Educación Superior. 1, 2, 15
- MTV** Model,Template,View. 31
- MVC** Modelo-Vista-Controlador. 31
- PMBOK** Cuerpo de conocimiento de gestión de proyectos. 16
- PMI** Instituto de Gestión de Proyectos. 16
- PNEU** Programa Nacional de Extensión Universitaria. 1, 2, 10, 13, 14
- SGBD** Sistemas gestores de bases de datos. 22
- SGBD** Sistemas gestores de bases de datos. 22
- SGU** Sistema de Gestión Universitaria. 16, 17, 59
- SIGENU** Sistema de gestión de la Nueva Universidad. 15, 17, 59

SIGPAC Sistema de gestión patrimonial. 14, 17, 59

SIGUA Sistema informático de gestión universitaria. 15, 17, 59

SNIES Sistema nacional de información de la educación superior. 15, 17, 59

SPADIES Sistema para la prevención de la deserción en la educación superior. 14, 17, 59

UCI Universidad de las Ciencias Informáticas. 1–4, 10, 14, 16–18, 27, 28, 36, 50, 54, 58

UHo Universidad de Holguín. 16, 17, 59

UML Lenguaje unificado de modelación. 18

Referencias bibliográficas

- ALIAGA, Adis Mirtha Canet; RUIZ, Álvaro Alejandro Acosta y PÉREZ, Julio César Espronceda, 2019. Sistema para la gestión de activos multimedia para la Dirección de Extensión Universitaria. *Serie Científica de la Universidad de las Ciencias Informáticas*. Vol. 12, n.º 12, págs. 1-10 (vid. pág. 9).
- ÁLVAREZ CARULLA, Albert, 2021. Introducción al control de versiones (Git) (vid. pág. 23).
- APORTELA, Odette González, 2016. *Sistema de gestión de la calidad del proceso de extensión universitaria en la Universidad de La Habana*. Editorial Universitaria (vid. págs. 7, 8, 12-14).
- BAKER, Stewart C, 2014. Making it work for everyone: HTML5 and CSS level 3 for responsive, accessible design on your library's web site. *Journal of Library & Information Services in Distance Learning*. Vol. 8, n.º 3-4, págs. 118-136 (vid. pág. 22).
- BARRAGAN, Luis Alejandro Arias, 2016. Lenguaje de modelamiento unificado (UML) para modelamiento de embotelladora. *Scientia et technica*. Vol. 21, n.º 1, págs. 38-42 (vid. pág. 19).
- BORGES PRADO, Cesar y POZO GONZÁLEZ, Francis, 2019. *Sistema para la gestión de Proyectos Extensionistas*. B.S. thesis. Universidad de las Ciencias Informáticas. Facultad 3 (vid. pág. 14).
- CORRALES SÁNCHEZ, Naylin, 2018. *Sistema para recomendar patrones de diseño de recursos educativos basado en el tratamiento de agrupaciones y la calidad percibida por los usuarios*. B.S. thesis. Universidad de las Ciencias Informáticas. Facultad 4 (vid. pág. 34).
- CORREDOR, Alejandro Paolo Daza; PEÑA, John Freddy Parra; RODRIGUEZ, Lilia Marcela Espinosa et al., 2016. Metodología de representación de software orientada al desarrollo ágil de aplicaciones: Un enfoque arquitectural. *Redes de Ingeniería*. Vol. 7, n.º 1, págs. 104-111 (vid. pág. 31).
- DENIS, Kuivalainen, 2021. *Web Application Development* (vid. pág. 22).
- DEU, 2016. Manual de procedimiento para la construcción de proyectos socioculturales en la Universidad de las Ciencias Informáticas (vid. págs. 12-14).
- EDUCACIÓN SUPERIOR DE LA REPÚBLICA DE CUBA, Ministerio de, 2022. Url: <https://www.mes.gob.cu> (vid. pág. 2).
- ESPRONCEDA PÉREZ, Julio César; ROMERO RAMOS, David Heberto y REMEDIO FROMETA, Suleika, 2021. Sistema de Gestión para los Proyectos Socioculturales de la Universidad de Ciencias Informáticas. *Universidad de las Ciencias Informáticas* (vid. págs. 10, 15-17).

- GONZÁLEZ APORTELA, Odette y BATISTA MAINEGRA, Amado, 2017. Gestión de la calidad del proceso extensionista en la Universidad de La Habana. *Revista cubana de educación superior*. Vol. 36, n.º 1, págs. 94-108 (vid. págs. 1, 8).
- GONZÁLEZ APORTELA, Odette; BATISTA MAINEGRA, Amado; GONZÁLEZ FERNÁNDEZ LARREA, Mercedes; ZAMBRANO LOOR, Tania Miladi y BALSINDE HERRERA, Jose de la Caridad, 2022. Retos en la Gestión del proceso extensionista cubano: Sistema de Información Gerencial (vid. págs. 7, 8).
- GONZÁLEZ FERNÁNDEZ-LARREA, M y GONZÁLEZ GONZÁLEZ, M, 2004. Programa Nacional de Extensión Universitaria. *La Habana: Ministerio de Educación Superior de Cuba* (vid. págs. 2, 6, 7, 12, 13).
- ILUSTRADO, Pequeño Larousse, 1989. Diccionario. *México. DF: Edición*. Vol. 19 (vid. pág. 11).
- KEFALAKIS, Nikos; SOLDATOS, John; ANAGNOSTOPOULOS, Achilleas y DIMITROPOULOS, Panagiotis, 2015. A visual paradigm for IoT solutions development. En: *A visual paradigm for IoT solutions development. Interoperability and Open-Source Solutions for the Internet of Things*. Springer, págs. 26-45 (vid. pág. 19).
- MALDONADO, José, 2018. Gestión de procesos. *Santiago-Chile* (vid. págs. 8, 9).
- MES, 2022. Ministerio de Educación Superior de la República de Cuba. <https://www.mes.gob.cu/extension-universitaria> (vid. págs. 1, 6).
- MORENO MONTERO, Mario Alejandro, 2017. *Aplicación web para la generación de solicitudes académicas de la Pontificia Universidad Católica del Ecuador Sede Ambato*. B.S. thesis. Pontificia Universidad Católica del Ecuador (vid. págs. 31, 32).
- PÉREZ, Javier Eguíluz, 2019. *introduccion a JavaScript* (vid. pág. 21).
- PRESSMAN, Roger S, 2016. *Software engineering: a practitioner's approach*. Palgrave macmillan (vid. págs. 31, 33, 43, 47, 50-52).
- PRINZ, Andreas y SHATALIN, Alexander, 2019. How to Bootstrap a Language Workbench. En: *How to Bootstrap a Language Workbench. MODELSWARD*, págs. 345-352 (vid. pág. 22).
- QUIÑONES LAFFITA, Arney; HERNÁNDEZ TORRES, Iosvani y CORDÓN GONZÁLEZ, Francisco, 2020. La comunicación como elemento clave en la gestión de la informatización del proceso extensión universitaria. *Revista de Desarrollo Sustentable, Negocios, Emprendimiento y Educación*. N.º enero (vid. pág. 16).
- SAAVEDRA, Rubisnel Almeida y MORA, Gisber Miguel Góngora, 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). *Redel. Revista granmense de Desarrollo Local*. Vol. 2, n.º 4, págs. 130-142 (vid. pág. 16).

- SÁNCHEZ, Tamara Rodríguez, 2015. Metodología de desarrollo para la Actividad productiva de la UCI. *La Habana* (vid. pág. 19).
- SIKOS, Leslie F y PHILP, Dean, 2020. Provenance-aware knowledge representation: A survey of data models and contextualized knowledge graphs. *Data Science and Engineering*. Vol. 5, n.º 3, págs. 293-316 (vid. pág. 35).
- SILVA GARCÍA, Liván, 2016. *Sistema para la gestión de los procesos de la Dirección de Proyectos Extensionistas en la UCI*. B.S. thesis. Universidad de las Ciencias Informáticas. Facultad 4 (vid. págs. 9, 15, 16).
- STUDIO, Visual, 2019. Visual Studio, 2018. URL: <https://visualstudio.microsoft.com> (vid. pág. 20).
- TESTING, PARASOFT Automated Software, 2019. How To Select and Implement the Right Secure Coding Standard. *Technical Whitepaper* (vid. pág. 38).
- VAN ROSSUM, Guido et al., 2007. Python Programming language. En: *Python Programming language. USENIX annual technical conference*. Vol. 41, págs. 1-36. N.º 1 (vid. pág. 20).
- VAUGHAN-NICHOLS, Steven J, 2010. Will HTML 5 restandardize the web? *Computer*. Vol. 43, n.º 4, págs. 13-15 (vid. pág. 21).
- VILORIA, Amelec; ACUÑA, Genesis Camargo; FRANCO, Daniel Jesús Alcázar; HERNÁNDEZ-PALMA, Hugo; FUENTES, Jorge Pacheco y RAMBAL, Etelberto Pallares, 2019. Integration of data mining techniques to PostgreSQL database manager system. *Procedia Computer Science*. Vol. 155, págs. 575-580 (vid. pág. 23).
- VINCENT, William S, 2021. *Django for Beginners: Build websites with Python and Django*. WelcomeToCode (vid. pág. 23).
- WANG, Yuting; ZHANG, Cheng y WANG, Futian, 2018. What Do We Know about the Tools of Detecting Design Patterns? En: *What Do We Know about the Tools of Detecting Design Patterns? 2018 IEEE International Conference on Progress in Informatics and Computing (PIC)*, págs. 379-387 (vid. págs. 34, 35).

Apéndices

A.1. Modelo de entrevista aplicada a especialistas de Extensión universitaria de la UCI

Tabla A.1. Encuesta.

<p>Estimado(a):</p> <p>Lea con detenimiento cada una de las preguntas antes de responder. Te agradecemos tu participación y franqueza al decirnos honestamente lo que piensa sobre el uso de los videojuegos para la rehabilitación cognitiva específicamente las funciones ejecutivas.</p>
<p>1- ¿Considera usted qué se deba llevar a cabo un proceso de desarrollo de la gestión de los proyectos extensionistas?</p> <p>_ No _ No sé _ Sí</p>
<p>2- ¿Si usted fuera a realizar un proyecto extensionista utilizaría el sistema informático propuesto?</p> <p>_ No _ No sé _ Sí</p>
<p>3- ¿Satisfacen sus expectativas, desde la posición de especialista de la extensión universitaria la aplicación propuesta?</p> <p>_ Me satisface mucho.</p> <p>_ No me satisface tanto.</p> <p>_ Me da lo mismo.</p> <p>_ Me disgusta más de lo que me satisface.</p> <p>_ No me satisface nada.</p> <p>_ No sé qué decir.</p>
<p>4- ¿Se pueden gestionar correctamente los proyectos extensionistas sin la documentación requerida?</p>
<p>5- ¿Utilizaría usted la aplicación propuesta para gestionar los proyectos extensionistas? Argumente.</p>

A.2. Guía de observación

Tabla A.2. Guía de observación (Fuente: Elaboración propia).

No.	Indicadores	D	R	B	MB	E
1	¿Son herramientas web?	4	2	3	6	7
2	¿Gestionan proyectos?	9	1	5	4	8
3	¿Interfaz gráfica?	1	2	9	5	4
4	¿Busca información?	4	2	1	7	9
5	¿Crean proyectos?	9	1	5	4	8
6	¿Presentan permisos de usuario?	4	1	2	5	8
7	¿Presentan licencias privativas?	5	2	3	8	4

- D: Deficiente
- R: Regular
- B: Bien
- MB: Muy bien
- E: Excelente

Las soluciones existentes evaluadas fueron las siguientes, las cuales se enumeraron para poder hacer el estudio:

- SIGPAC (1)
- SPADIES (2)
- SNIES (3)
- Microsoft project (4)
- SIGENU (5)
- SIGUA (6)
- UHo (7)
- GESPRO (8)
- SGU (9)

A.3. Resultado de las entrevistas

En las entrevistas realizadas se muestran los siguientes resultados de forma cuantificada:

- ¿Considera usted qué se deba llevar a cabo un proceso de desarrollo de la gestión de los proyectos extensionistas?
En esta pregunta las respuestas de las 10 personas a las cuales se le realizó la entrevista fue “Si”.
- ¿Si usted fuera a realizar un proyecto extensionista utilizaría el sistema informático propuesto?
En esta pregunta la respuesta fue 1 No sé y 9 Si

- ¿Satisfacen sus expectativas, desde la posición de especialista de la extensión universitaria la aplicación propuesta?

En esta pregunta las respuestas de las 10 personas a las cuales se le realizó la entrevista fue “Si”.

A.4. Historias de Usuario

Tabla A.3. Historia de usuario # 4

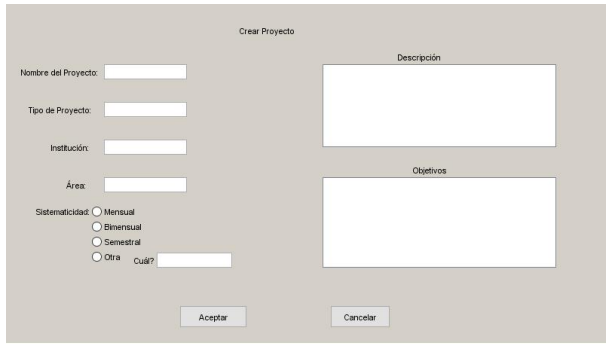
Historia de usuario	
Número: 4	Nombre: Solicitar proyecto
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: No.1
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir crear un proyecto, basado en que hay que completar los siguientes campos, nombre del proyecto, institución a la que pertenece, lugar fecha de inicial, fecha final. Además, se debe mostrar un mensaje de confirmación una creado el proyecto	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.4. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Listar proyecto
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: No.3
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir listar los proyectos, basado en el tipo de proyecto y el estado en que se encuentra.	

Continúa en la próxima página

Tabla A.4. Continuación de la página anterior


Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.5. Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Modificar proyecto
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: No.4
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir modificar un proyecto, basado en que hay que completar los siguientes campos, con la información nueva, nombre del proyecto, institución a la que pertenece, lugar fecha de inicial, fecha final. Además, se debe mostrar un mensaje de confirmación de proyecto modificado	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	

Continúa en la próxima página

Tabla A.5. Continuación de la página anterior

Interfaz:



The screenshot shows a form titled "Modificar Proyecto" with the following fields and buttons:

- Nombre del Proyecto:
- Institución:
- Lugar:
- Fecha Inicial:
- Fecha Final:
- Buttons: Aceptar, Cancelar

Tabla A.6. Historia de usuario # 7

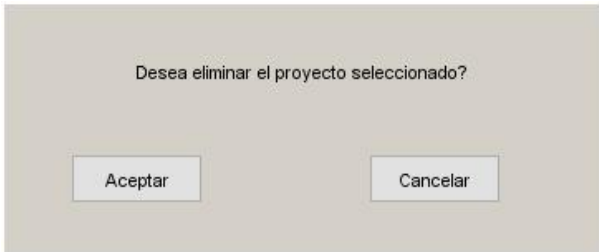
Historia de usuario	
Número: 7	Nombre: Eliminar proyecto
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: No.5
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir eliminar los proyectos seleccionados.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
 <p>The screenshot shows a confirmation dialog box with the text "Desea eliminar el proyecto seleccionado?" and two buttons: "Aceptar" and "Cancelar".</p>	

Tabla A.7. Historia de usuario # 8


Historia de usuario	
Número: 8	Nombre: Crear participante
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: No.6
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir crear o adicionar un participante, completando el campo, nombre de usuario.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.8. Historia de usuario # 9

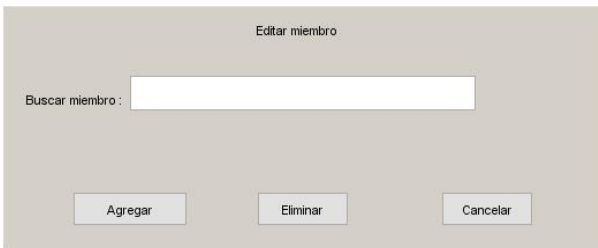
Historia de usuario	
Número: 9	Nombre: Editar participante
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: No.7
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir modificar los participantes correspondientes a un proyecto.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.9. Historia de usuario # 10

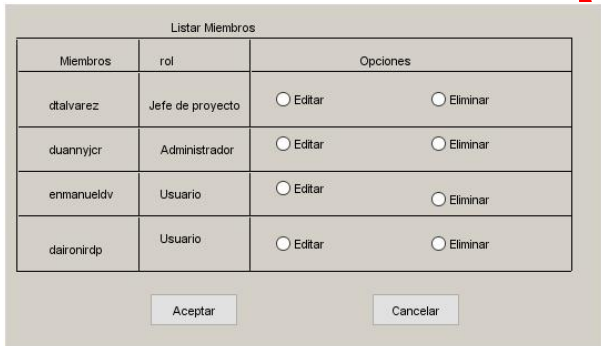
Historia de usuario	
Número: 10	Nombre: Listar participante
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: No.8
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir listar los participantes, basado en el rol correspondiente.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.10. Historia de usuario # 11

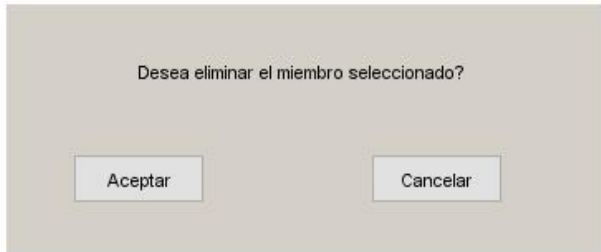
Historia de usuario	
Número: 11	Nombre: Eliminar participante
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: No.9
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir eliminar los participantes de un proyecto seleccionado.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.11. Historia de usuario # 12

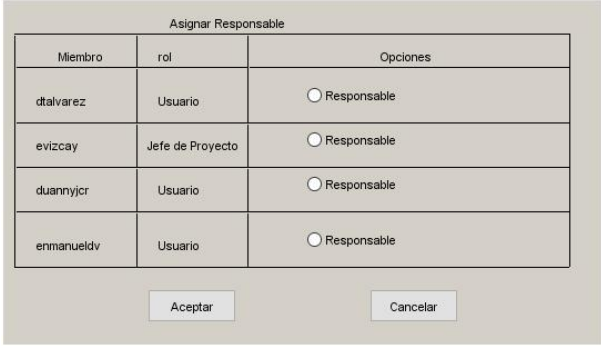
Historia de usuario	
Número: 12	Nombre: Asignar responsable
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: No.10
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir Asignar a uno o más responsables para el cumplimiento de las actividades de un proyecto seleccionado.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.12. Historia de usuario # 13

Historia de usuario	
Número: 13	Nombre: Crear solicitud
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: No.11
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir crear una solicitud, completando los siguientes campos. Además, se debe mostrar un mensaje de que su solicitud está en estado de observación	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	

Continúa en la próxima página

Tabla A.12. Continuación de la página anterior

Interfaz:




Tabla A.13. Historia de usuario # 14


Historia de usuario	
Número: 14	Nombre: Modificar solicitud
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: No.12
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir modificar la solicitud correspondiente para entrar a un proyecto por parte del usuario.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.14. Historia de usuario # 15

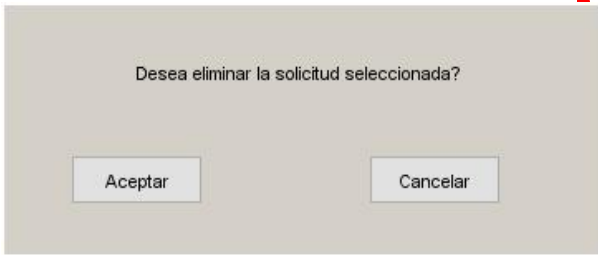
Historia de usuario	
Número: 15	Nombre: Eliminar solicitud
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: No.13
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir eliminar la solicitud por parte del usuario o en caso de ser rechazada.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.15. Historia de usuario # 16

Historia de usuario	
Número: 16	Nombre: Crear comentario
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 2	Iteración asignada: No.14
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir crear un comentario. Además, se debe mostrar un mensaje de que su comentario está en estado de observación, luego según cumpla con las políticas de aceptación pasa al estado correspondiente.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	

Continúa en la próxima página

Tabla A.15. Continuación de la página anterior

Interfaz:




Tabla A.16. Historia de usuario # 17

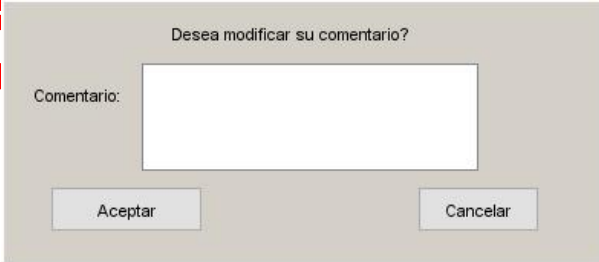
Historia de usuario	
Número: 17	Nombre: Modificar comentario
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: No.15
Programador responsable: Daríel Torres Álvarez	
Descripción: El sistema debe permitir modificar el comentario realizado por el usuario correspondiente.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.17. Historia de usuario # 18

Historia de usuario	
Número: 18	Nombre: Eliminar comentario
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta

Continúa en la próxima página

Tabla A.17. Continuación de la página anterior

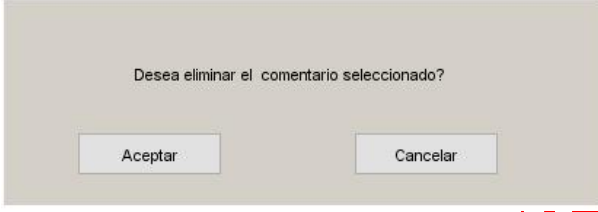
Puntos estimados: 1	Iteración asignada: No.16
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir eliminar comentarios del usuario correspondiente.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.18. Historia de usuario # 19

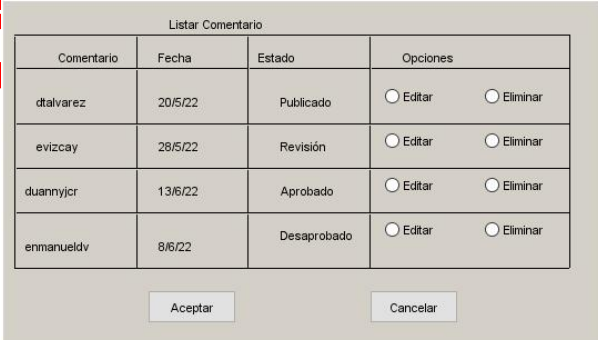
Historia de usuario	
Número: 19	Nombre: Listar comentario
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: baja
Puntos estimados: 1	Iteración asignada: No.17
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir listar los comentarios, por fecha de publicación y si es aprobado o no.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.19. Historia de usuario # 20

Historia de usuario	
Número: 20	Nombre: Buscar información

Continúa en la próxima página

Tabla A.19. Continuación de la página anterior


Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: No.18
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir la búsqueda de información requerida por el usuario según su rol. Además, se debe mostrar la información buscada y en caso de no tener permiso para acceder a esa información notificarle al usuario.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
	

Tabla A.20. Historia de usuario # 21

Historia de usuario	
Número: 21	Nombre: Generar pdf
Usuario: Usuario	
Prioridad en negocio: Prioridad	Riesgo en desarrollo: media
Puntos estimados: 2	Iteración asignada: No.19
Programador responsable: Dariel Torres Alvarez	
Descripción: El sistema debe permitir generar la información requerida de un proyecto, según el rol del usuario solicitante. En caso de no tener permiso para generar esa información notificarle al usuario.	
Observaciones: El usuario autenticado debe tener los permisos correspondientes	
Interfaz:	
