



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas
Facultad CITEC - Parque Científico Tecnológico 3CE

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Módulo del sistema GAPID para la gestión de proyectos y recursos humanos del PCT de la Habana

Autor(es):

Julio Temistocle Nerey Villanueva
Roddy Javier Garcia Mayea

Tutores:

Dr.C. Arturo Orellana García, P.A.

Consultante:

MSc. Ihoandra Sotolongo Carballo
Ing. Marislay Soria Pérez.

La Habana, 6 de diciembre de 2023

Declaración de autoría

El(Los) autor(es) del trabajo de diploma con título "**Módulo del sistema GAPID para la gestión de proyectos y recursos humanos del PCT de la Habana**" concede(n) al Proyecto PS161LH001-022 Sistema para la administración de programas y proyectos de Ciencia, Técnica e Innovación, asociado al Programa Sectorial de Ciencia, Tecnología e Innovación "PS161LH001, Industria Cubana del Software e Informatización de la Sociedad los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara(n) como único(s) autores de su contenido. Para que así conste firma(n) la presente a los 6 días del mes de diciembre del año 2023.

Julio Temistocle Nerey Villanueva

Firma del Autor

Roddy Javier García Mayea

Firma del Autor

Dr.C. Arturo Orellana García, P.A.

Firma del Tutor

Datos de contacto

Dr.C. Arturo Orellana García: graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2012. Se desempeña como líder del Grupo de Investigación de Minería de procesos y Asesor de Capacitación, Desarrollo e Investigación del Centro de Soluciones de Informática Médica. Ha liderado proyectos I+D+i de desarrollo de componentes de software a partir de minería de procesos para el análisis de procesos de negocio del entorno hospitalario. Investiga la Ingeniería de comportamiento, la medicina de precisión y el procesamiento de imágenes médicas. Tutora varias tesis de grado, maestrías y doctorados enfocados al análisis de procesos de negocio, la informática médica y otras áreas del conocimiento. Obtuvo el grado de Máster en Informática Aplicada en 2015 desarrollando una herramienta informática basada en técnicas de minería de procesos para identificar problemas en la ejecución de procesos de negocio. Doctor en Ciencias Técnicas desde 2016 presentando un modelo computacional para la detección de variabilidad en procesos de negocio del entorno sanitario aplicando minería de procesos. Correo electrónico: aorellana@uci.cu

Resumen

Cuba desarrolla una política basada en la ciencia y luego en la transformación digital, lo que ha provocado que se creen programas para la ejecución de proyectos y en este sentido las universidades, los centros de investigación y las empresas cubanas juegan un papel fundamental. El Parque Científico Tecnológico de La Habana es un ecosistema de innovación para el desarrollo de proyectos y la incubación de nuevas Empresas de Base Tecnológica, en el sector de las TIC. Se concibe además como una organización que combina la investigación científica, la innovación tecnológica, la formación de alto nivel, el desarrollo de productos y servicios de alto valor agregado a partir de la incubación de proyectos. La gestión de los proyectos en el PCT se realiza de forma manual a pesar de ser un ecosistema de innovación, lo que provoca descentralización de los flujos de información y deficiencias en la gestión integral de los proyectos.

El presente trabajo centra su objetivo en desarrollar un módulo en el sistema GAPID que facilite la gestión administrativa de los proyectos incubados en el Parque Científico Tecnológico 3CE. Para ello se asimiló el ambiente de desarrollo del proyecto, guiado por la metodología AUP-UCI en su escenario 4. Adicionalmente, se aplicaron métodos y técnicas para la obtención de los requisitos funcionales y las características deseadas de la propuesta. Como resultado se obtuvo un módulo del sistema GAPID que facilita la gestión de los proyectos y recursos humanos del PCT 3CE con un importante impacto tecnológico para la institución.

Palabras clave:

Ecosistema de innovación, desarrollo de software, gestión de proyectos, recursos humanos.

Abstract

Cuba is developing a science-based policy followed by digital transformation, which has led to the creation of programs for project execution. In this regard, Cuban universities, research centers, and companies play a fundamental role. The Havana Science and Technology Park is an innovation ecosystem for project development and the incubation of new Technology-Based Enterprises in the ICT sector. It is conceived as an organization that combines scientific research, technological innovation, high-level training, and the development of high-value products and services through project incubation. However, project management in the Science and Technology Park is currently done manually, despite being an innovation ecosystem. This leads to decentralization of information flows and deficiencies in comprehensive project management.

The objective of this work is to develop a module in the GAPID system that facilitates the administrative management of the projects incubated in the 3CE Science and Technology Park. The project development environment was assimilated, guided by the AUP-UCI methodology in its scenario 4. Additionally, methods and techniques were applied to obtain the functional requirements and desired characteristics of the proposal. The result was a module of the GAPID system that facilitates the management of projects and human resources in the 3CE Science and Technology Park, with a significant technological impact for the institution.

Keywords:

Innovation ecosystem, software development, project management, human resources.

Índice

Introducción.....	9
CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN.....	15
1.1.1 Conceptos fundamentales.....	15
1.1.2 Sistema de gestión.....	15
1.1.3 Proyectos.....	16
1.1.4 Características de los proyectos.....	18
1.1.5 Tipos de proyectos.....	19
1.1.6 Dirección Integrada de Proyectos - DIP.....	20
1.1.7 Ecosistema de Innovación del PCT 3CE.....	22
1.1.8 Certificación de proyectos.....	25
1.2 Sistemas homólogos.....	27
1.2.1 Project.co.....	27
1.2.2 Akademos.....	27
1.2.3 GESPRO.....	28
1.2.4 Jira Software	28
1.2.5 Análisis comparativos de los sistemas homólogos.....	28
1.2.6 GAPID.....	30
1.3 Ambiente de Desarrollo.....	31
1.3.1 Metodología AUP-UCI, Escenario 4.....	31
1.3.2 Herramienta CASE.....	33
1.3.3 Marco de Trabajo.....	34
1.3.4 Lenguaje de programación.....	34
1.3.5 Entorno de desarrollo integrado.....	36
Conclusiones del capítulo.....	36
CAPÍTULO II: Análisis y diseño del Módulo de Gestión de proyectos.....	38

2.1 Propuesta de Solución.....	38
2.2 Requisitos del Sistema.....	39
2.2.1 Requisitos Funcionales.....	39
2.2.2 Requisitos No Funcionales.....	42
2.2.3 Historias de usuario.....	43
2.3 Fase diseño.....	46
2.3.2 Patrones de diseño.....	47
2.3.3 Modelo de datos.....	52
Conclusiones del Capítulo.....	52
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	54
3.1 Fase de implementación.....	54
3.1.1 Estándar de codificación.....	54
3.1.2 Diagrama de componentes:.....	56
3.2 Estrategia de pruebas.....	58
3.2.1 Pruebas de Software.....	59
3.2.2 Tipos de prueba.....	59
3.2.2.1 Método de caja blanca.....	59
3.2.2.2 Método de caja negra.....	60
3.2.2.3 Prueba de Integración.....	61
3.2.2.4 Prueba de Sistema.....	61
3.2.2.5 Prueba de Aceptación.....	61
3.2.3 Niveles de prueba.....	62
3.2.3.1 Pruebas unitarias.....	62
3.2.3.2 Técnicas automatizadas.....	65
3.2.3.3 Pruebas de sistema.....	67
3.2.3.4 Pruebas de aceptación. Método de caja negra.....	69

Conclusiones del capítulo.....	71
Conclusiones Generales.....	72
Referencias bibliográficas.....	75
Anexos.....	88

Índice de Figuras

Figura 1 :Ejemplo de Modelo Vista Plantilla.....	47
Figura 2 :Fragmento de código del patrón adaptador.....	48
Figura 3 :Fragmento de código del uso del patrón decorador.....	49
Figura 4 :Fragmento de código del patrón iterador.....	49
Figura 5 :Fragmento de código del patrón experto.....	50
Figura 6 :Fragmento de código del patrón Alta cohesión.....	51
Figura 7 :Fragmento de código del patrón Bajo acoplamiento.....	51
Figura 8 :Modelo de datos relacional para el sitio web.....	52
Figura 9 :Ejemplo de código de los estándares de codificación.....	55
Figura 10 : Ejemplo de código de los estándares de codificación.....	56
Figura 11 : Ejemplo de código de los estándares de codificación.....	56
Figura 12 :Diagrama de componentes correspondiente a la HU_1 Crear proyecto.....	57
Figura 13 :Método caja blanca.....	60
Figura 14 :Método caja negra.....	61
Figura 15 :Código del método a realizar la técnica de camino básico.....	63
Figura 16 :Grafo de flujo del método Post.....	63
Figura 17 :Total, de las NC detectadas durante las pruebas de caja negra.....	70
Figura 18 :Clasificación de las NC detectadas corregidas.....	70

Índice de Tablas

Tabla 1 :Comparación de los sistemas homólogos. Fuente: Elaboración propia.....	28
Tabla 2 :Fases de la variación de AUP para la UCI.....	31

Tabla 3 :Roles de los usuarios. Fuente: Elaboración propia.....	38
Tabla 4 :Requisitos Funcionales.....	39
Tabla 5 : Crear proyecto.....	45
Tabla 6 :Desactivar proyecto.....	46
Tabla 7 : Editar Proyecto.....	46
Tabla 8 :Caso de prueba para el camino básico 1. Fuente: Elaboración propia.....	64
Tabla 9 :Caso de prueba para el camino básico 2. Fuente: Elaboración propia.....	65
Tabla 10 :Caso de prueba para el camino básico 3. Fuente: Elaboración propia.....	65
Tabla 11 :Caso de prueba para el camino básico 4. Fuente: Elaboración propia.....	65
Tabla 12 :Resultados de las pruebas de rendimiento y resistencia. Fuente: Elaboración propia.....	68
Tabla 13 :Resultados del método de caja negra mediante la técnica partición equivalente en las pruebas de aceptación. Fuente: Elaboración propia.....	69

Introducción

El continuo desarrollo tecnológico/empresarial ha conducido a que cada empresa busque mejorar sus productos y servicios en cuanto a diferentes indicadores, entre ellos, calidad y tiempo. En este escenario es importante tener en cuenta una serie de acciones y realizar un análisis previo que indique qué, cuándo, dónde, cómo y sobre todo por qué se deben ejecutar las acciones correspondientes; este análisis se corresponde con el inicio de un proyecto.

El *Project Management Institute* (PMI) es una organización estadounidense sin fines de lucro que asocia a profesionales relacionados con la Gestión de Proyectos, es la más grande del mundo en su área. Esta manifiesta que: un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Temporal porque tiene un comienzo y un fin definido; y único porque el producto o servicio es diferente de alguna manera que lo distingue de otros productos o servicios (Salcedo Quevedo, 2020).

Existen varios tipos o clasificaciones de proyectos, dependiendo del área donde se desarrollarán, por ejemplo: proyectos tecnológicos, proyectos informáticos, proyectos de innovación, proyectos de investigación, entre otros. Un proyecto tecnológico es aquel que tiene como propósito generar o adaptar, dominar y utilizar una tecnología nueva en una región, sector productivo o aplicación específica y que permite a quienes lo desarrollen acumular conocimientos y las habilidades requeridas para explicar exitosamente la tecnología y posibilitar su mejora continua (Colciencias, 2023). Los proyectos informáticos son aquellos que se relacionan con la instalación y puesta en marcha de sistemas informáticos con determinados fines. Los proyectos de innovación, son un plan estratégico que supone la creación de nuevas ideas, productos o servicios, que conlleven el desarrollo de un área como la educación, la tecnología, los modelos de negocio, entre otros .

La presente investigación se centra específicamente en proyectos de investigación (I+D+i) los cuales son definidos por (Pérez Porto & Gardey, 2021) como: un procedimiento científico destinado a obtener información y formular hipótesis sobre un determinado fenómeno social o científico. Para obtener los resultados esperados, uno de los aspectos que debe ser

considerado es la calidad en la ejecución y control de la información que se genera y gestiona en un proyecto de este tipo.

Organizaciones, agencias y gobiernos como BID, BM, PNUD, CEPAL, UNESCO; UNICEF, Canadá, Japón, entre otros, aplican programas y metodologías diversas con estos fines (Rodríguez Cardona & Cobas Aranda, 2010). En el contexto actual se hace necesario disponer de herramientas para monitorear y evaluar el avance y el desempeño de los proyectos de investigación, así como sus resultados y el impacto en la sociedad, con el objetivo de asegurar su contribución al desarrollo social y económico del país.

Cuba desarrolla una política basada en la ciencia y luego en la transformación digital, lo que ha provocado que se creen programas para la ejecución de proyectos y en este sentido las universidades cubanas juegan un papel fundamental. Por parte del Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA) se realizan convocatorias a proyectos Nacionales, Sectoriales e Institucionales para favorecer el desarrollo científico del país, así como la creación de ecosistemas de innovación tecnológica en Parques Científicos Tecnológicos. (Gómez Torres et al., 2020)

En el Parque Científico Tecnológico de La Habana (PCT) se incuban proyectos de I+D+i que se diseñan a partir de ideas y necesidades de innovación, que pueden ser presentados por personas jurídicas o naturales, sean nacionales o extranjeras. A partir de una búsqueda en la gaceta oficial de la República de Cuba se identificaron 2 resoluciones sobre el funcionamiento del Parque sobre los proyectos de incubación: (3ce, 2023).

- **Decreto 363/2019**: De los Parques Científicos y Tecnológicos y de las Empresas de Ciencia y Tecnología que funcionan como interface entre las universidades y entidades de ciencia, tecnología e innovación con las entidades productivas y de servicio
- **Resolución 237/2023**: Prestar servicios de asesoramiento jurídico a proyectos de investigación, desarrollo e innovación y empresas de bases tecnológicas incubadas en este centro.

INTRODUCCIÓN

En el Parque Científico Tecnológico (PCT), estos documentos se realizan de forma manual, constituye un proceso engorroso y requiere tiempo y esfuerzo de los jefes de proyectos para generar los documentos. La introducción de cualquier error o incongruencia conlleva a reelaborar toda la documentación, lo que perjudica la entrega en tiempo al Parque Científico Tecnológico de la Habana. Se utilizan documentos Word y Excel, los cuales generan un cúmulo de documentación e información extenso. La problemática anterior incide negativamente sobre los tiempos para certificar los avances de los proyectos de investigación y conlleva a que el trabajo se haga bajo presión, además significa tareas administrativas adicionales para los jefes de proyecto.

Por lo anteriormente descrito se identifica como **problema de investigación**: ¿Cómo facilitar la gestión administrativa de los proyectos incubados en el Parque Científico Tecnológico 3CE? El problema está centrado en el **objeto de estudio**: Proceso de gestión administrativa de proyectos radicados en ecosistemas de innovación tecnológica, enmarcado en el **campo de acción**: Gestión administrativa en proyectos incubados en el Parque Científico Tecnológico 3CE.

Objetivo general: Desarrollar un módulo en el sistema GAPID que facilite la gestión administrativa de los proyectos incubados en el Parque Científico Tecnológico 3CE

Tareas de la Investigación:

1. Elaboración del marco teórico de la investigación relacionado con los ecosistemas de innovación tecnológica y la gestión administrativa en proyectos de investigación y desarrollo.
2. Elaboración de los artefactos ingenieriles relacionados a la gestión administrativa de los proyectos incubados en el Parque Científico Tecnológico 3CE
3. Desarrollo del módulo para la gestión administrativa de los proyectos incubados en el Parque Científico Tecnológico 3CE.
4. Integración del resultado obtenido al sistema GAPID que propicie su funcionamiento con otros módulos del sistema

5. Validación del resultado obtenido a partir de la estrategia de pruebas de software definida para la investigación.

Los **métodos científicos** utilizados para desarrollar la investigación fueron:

Métodos teóricos:

Histórico - lógico: Este método se utilizó para establecer los antecedentes y la evolución de los proyectos de investigación, así como para comprender el estado actual de la gestión administrativa en este contexto. Se realizaron investigaciones sobre proyectos similares, experiencias previas y casos de éxito en la gestión de proyectos en ecosistemas de innovación tecnológica.

Analítico - Sintético: Se aplicó este método para analizar y comparar los sistemas existentes de gestión administrativa en proyectos incubados en parques científico-tecnológicos. Se examinaron los procesos, documentos y flujos de trabajo utilizados en la gestión administrativa actual, identificando fortalezas, debilidades y áreas de mejora.

Dialéctico: Este método se utilizó para comprender los diferentes procesos y elementos que intervienen en el sistema de gestión administrativa de los proyectos incubados. Se analizaron las interacciones y relaciones entre los diferentes actores y componentes del sistema, considerando la posibilidad de cambios y adaptaciones en el tiempo.

Inductivo - Deductivo: A través de este método, se partió del estudio de los procesos de gestión de información de proyectos de investigación en el Parque Científico Tecnológico para identificar la problemática y determinar los elementos clave que darían respuesta a dicha problemática.

Modelación: Se empleó la modelación como método para desarrollar los artefactos ingenieriles necesarios en el proceso de investigación. Esto implicó la creación de modelos gráficos y conceptuales que representaran los diferentes elementos y componentes de la solución propuesta, como el módulo para la gestión administrativa en el sistema GAPID.

Métodos empíricos:

Observación: La observación fue utilizada para adquirir conocimiento directo sobre la gestión administrativa de proyectos de investigación a través de la observación de los procedimientos y técnicas utilizadas en el PCT. Esto permitió al equipo de investigación tener una visión realista de la situación actual y comprender los problemas y desafíos que enfrentan los jefes de proyecto en el manejo de la documentación.

Entrevistas: Las entrevistas fueron una técnica utilizada para obtener información más detallada y específica. El equipo de investigación se acercó a los profesionales del PCT para obtener sus perspectivas y experiencias sobre la gestión documental y los efectos negativos de su deficiente manejo. Mediante las entrevistas, se recopilaron opiniones, ideas y sugerencias para mejorar el proceso de gestión administrativa.

Análisis de antecedentes: El análisis de antecedentes consistió en estudiar las plataformas y sistemas de gestión de información relacionados que ya existen en el ámbito de la gestión de proyectos de investigación. Este análisis permitió al equipo de investigación identificar buenas prácticas, soluciones exitosas y lecciones aprendidas de otros contextos similares, lo que ayudó a informar el diseño de la solución propuesta.

Análisis de documentos: El análisis de documentos implicó consultar y analizar información relevante sobre la gestión documental en proyectos de investigación. El equipo de investigación revisó documentos, normativas y pautas relacionadas con la gestión administrativa, incluyendo cómo se deben registrar y procesar los documentos, y las mejores prácticas en términos de documentación y flujos de trabajo

Resultados Esperados

Al incorporar este módulo al sistema GAPID en el Parque Científico Tecnológico de La Habana, se espera mejorar la gestión administrativa de los proyectos de investigación, optimizar los procesos documentales, agilizar la generación de informes y proporcionar una plataforma más eficiente para los jefes de proyectos. Esto liberaría tiempo y recursos que pueden ser dedicados a la investigación propiamente dicha, mejorando la productividad y el rendimiento general de los proyectos en el PCT.

El documento está estructurado en tres capítulos, siendo estos:

El **Capítulo 1. Fundamentación teórica de la investigación**, se centra en la fundamentación teórica de la investigación. Se analizan las soluciones adoptadas por otros investigadores y profesionales en el campo, identificando las mejores prácticas y limitaciones presentes. Esto establece un marco de referencia sólido para el desarrollo del sistema propuesto. Además, se describen las tecnologías y herramientas de vanguardia utilizadas en el sistema de gestión de información de proyectos de investigación, destacando su elección basada en su capacidad para cumplir con los requisitos y ser compatibles con el entorno del Parque Científico Tecnológico de La Habana. Se realiza un análisis exhaustivo del estado del arte y se establece la base teórica para los siguientes capítulos, que abordarán la metodología y la implementación del sistema.

El **Capítulo 2. Propuesta de solución**, propone una solución para automatizar la gestión de información de proyectos de investigación. Se analizan los procesos de negocio, se determina un Modelo Conceptual y se especifican los requisitos del sistema de manera rigurosa. Esta propuesta sienta las bases para el desarrollo e implementación del sistema en el Parque Científico Tecnológico de La Habana.

Por último, en el **Capítulo 3. Validación de la propuesta de solución**, se respalda y valida la solución propuesta mediante diversos elementos. Se presentan diagramas de componentes que visualizan la estructura y la interacción del sistema. Se aborda el modelo de datos para garantizar la organización y la integridad de la información. Además, se detallan las estrategias y estándares de codificación para asegurar la calidad del código. Se describe la integración del módulo de gestión administrativa y se realizan pruebas y verificaciones que confirman su correcto funcionamiento. Estos elementos validan la efectividad y viabilidad de la solución en el contexto del Parque Científico Tecnológico de La Habana.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

El presente capítulo tiene como objetivo abordar los diferentes elementos que brindan la base teórica y conceptual para el desarrollo de la investigación. Valorar de forma crítica las tendencias y tecnologías actuales, así como los antecedentes asociados al campo de acción. Se realiza un análisis de las características de los diversos sistemas de gestión existentes a nivel nacional e internacional. De este modo, se podrá realizar una correcta interpretación de la situación problemática y del problema a resolver.

1.1.1 Conceptos fundamentales

Para obtener una mejor comprensión del dominio del problema, a continuación, son descritos conceptos asociados al mismo.

1.1.2 Sistema de gestión

Un sistema de gestión es una herramienta que permite controlar, planificar, organizar y automatizar las tareas administrativas de una organización. Un sistema de gestión analiza los rendimientos y los riesgos de una empresa, con el fin de otorgar un ambiente laboral más eficiente y sostenible. Las empresas o PyMEs (pequeñas y medianas empresas) cuentan con actividades que no están automatizadas, que con frecuencia se soportan en sistemas departamentales y casi siempre en hojas Excel desarrolladas individualmente por los usuarios implicados en cada una de las funciones (Significados, 2023).

Un *software* de gestión unifica la operación de todas las áreas del negocio para alinearlas con los objetivos de la empresa. Es importante que cada área tenga claro y definido tanto su objetivo como sus metas. Esto da la garantía de direccionar tu negocio más fácil y eficazmente. Donde un “*Enterprise Resource Planning*” (ERP) es la planificación de recursos empresariales, abarca todos los sistemas y paquetes de *software* que utilizan las organizaciones para gestionar sus actividades empresariales diarias como la contabilidad, las adquisiciones, la gestión de proyectos y la producción. Integra todos los procesos de negocio de una empresa, permitiendo que toda la información esté disponible en cualquier momento

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN de manera centralizada, es decir, con acceso total todo el tiempo. Por su parte, un *software* de gestión normal se basa en múltiples aplicaciones independientes o modulares que duplican los datos o no los centralizan en una única base de datos. Un sistema ERP es un tipo de solución de un sistema de gestión de proyectos. (Evaluando ERP, 2023)

1.1.3 Proyectos

El *Project Management Institute* (PMI) es una organización estadounidense sin fines de lucro que asocia a profesionales relacionados con la Gestión de Proyectos, es la más grande del mundo en su área. Esta manifiesta que: un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Temporal porque tiene un comienzo y un fin definido; y único porque el producto o servicio es diferente de alguna manera que lo distingue de otros productos o servicios[CITATION Sal \l 3082](Salcedo Quevedo, 2020).

En un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos implica que un proyecto tiene un principio y un final definidos. El final se alcanza cuando se logran los objetivos del proyecto, cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto. El resultado del proyecto puede ser tangible o intangible. Aunque puede haber elementos repetitivos en algunos entregables y actividades del proyecto, esta repetición no altera las características fundamentales y únicas del trabajo del proyecto. Los proyectos se llevan a cabo en todos los niveles de una organización y estos pueden involucrar a una única persona o a varias personas, a una única unidad de la organización, o a múltiples unidades de múltiples organizaciones. (PMI, 2013)

Un proyecto es una planificación que consiste en un conjunto de actividades que se encuentran interrelacionadas y coordinadas. La razón de un proyecto es alcanzar objetivos específicos dentro de los límites que imponen un presupuesto, calidades establecidas previamente y un lapso de tiempo previamente definido. (Parodi, 2020)

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

La Norma ISO 21500 define un proyecto como un conjunto único de procesos que consta de actividades coordinadas y controladas, con fechas de inicio y fin, que se llevan a cabo para lograr los objetivos del proyecto. Esta Norma es un conjunto de directrices sobre Dirección y Gestión de Proyectos. Estas especifican la manera en que una organización gestiona sus estándares en relación con sus proyectos, procesos, tiempos de entrega, niveles de servicio, gestión del riesgo. Proporciona una descripción de alto nivel sobre los conceptos y procesos que forman parte de las buenas prácticas en dirección y gestión de proyectos. Su estructura sigue las pautas del *PMBOK(Project Management Body of Knowledge)*, pero permite además que sea aplicada de manera genérica en cualquier tipo de organización, incluyendo organizaciones públicas, privadas o comunitarias, y para cualquier tipo de proyecto, independientemente de su complejidad, tamaño y duración. Además, busca estabilizar y sistematizar tareas y homogeneizar actividades, es decir, que el resultado de un proceso sea el mismo independientemente de la persona que lo lleve a cabo. (García Regüela, 2023).

Su objetivo es ayudar a directores de Proyecto, principiantes o experimentados, a aplicar las mejores prácticas en la gestión de sus proyectos, mejorando los resultados de negocio y concluyendo sus misiones con éxito. La norma europea puede hacerse extensible a multitud de áreas, lo que amplía aún más su utilidad. Está diseñada para permitir su aplicación a: (Pérez, 2014)

- Cualquier tipo de organización: incluidas las organizaciones públicas, privadas o comunitarias.
- Proyectos de diversa índole: independientemente de su complejidad, tamaño y duración.

(ISO Tools Excellence, 2017) comenta los beneficios de la aplicación de norma ISO 21500:

- Fomento de la transferencia de conocimientos entre proyectos y organizaciones.
- Mejora de las condiciones de ejecución de las diferentes etapas de los proyectos.
- Fomentar los procesos de licitación y su eficiencia.
- Promoción del uso de una terminología de gestión de proyectos coherente.
- Aumento de la flexibilidad de los trabajadores de gestión de proyectos.
- Adecuación de la capacidad de los equipos de proyecto para su trabajo en entornos internacionales.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

Referirse además que PMBOK(*Project Management Body of Knowledge*)[CITATION Par00 \l 3082] describe la planificación de actividades, sin embargo, estas para el autor de PMBOK(*Project Management Body of Knowledge*) deben ser un conjunto de actividades interrelacionadas y controladas y para ISO 21500 actividades coordinadas y controladas. La definición más tradicional "*es un esfuerzo planificado, temporal y único, realizado para crear productos o servicios únicos que agreguen valor o provoquen un cambio beneficioso*". (Parodi, 2020)

1.1.4 Características de los proyectos

Independientemente del tipo de proyecto existen características en común para todos ellos, todo proyecto tiene un propósito y para ello se definen distintos objetivos para cumplir las metas. Las metas trazadas y lograr el propósito inicial debe ser en un espacio de tiempo limitado y para ello se debe transitar como mínimo por tres fases necesarias: planificación, ejecución y entrega. Los proyectos se orientan para obtener un resultado e involucran a personas que cumplen distintos roles y responsabilidades. Mientras el proceso está en desarrollo, este es sometido a un seguimiento y control para obtener el resultado deseado porque también se ven afectados por la incertidumbre. Por último, señalar que ningún proyecto se parece a otro, incluso teniendo características similares.

También hay referencias sobre los rasgos más significativos de un proyecto, ejemplo de esto son los propuestos por (Pérez, 2021):

- **Complejidad:** La complejidad se mide tanto en la dificultad de las tareas a realizar como en el nivel de monitorización que estas supongan. También se hace patente en sus riesgos.
- **Marco temporal:** los proyectos deben tener, como mínimo, dos fechas: una de inicio y otra de finalización. Aunque no necesariamente se cumplan tal como se han previsto, sí que deben servir como referencia para la ejecución de tareas, la estimación de costes y las previsiones de resultados. Un proyecto no puede ser indefinido; si eso ocurre, es un proyecto fallido. El marco temporal es otro de los factores que aumentan su complejidad.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

- **Grupal:** al ser varias las tareas, labores o responsabilidades que se generan, los proyectos se ejecutan en grupos de trabajo previamente elegidos. Los miembros de esos grupos no necesariamente deben pertenecer a una misma disciplina o campo; pueden provenir de diversas disciplinas o pertenecer a distintas áreas o departamentos dentro de la propia organización.
- **Coordinación de actividades:** los proyectos están conformados por varias acciones, a veces secuenciales o a veces paralelas. No tienen una sola acción; de ser así, no dejarían de ser tareas aisladas. Además, tampoco son lineales, sino que requieren coordinación, seguimiento y monitoreo de un director de proyecto.
- **Recursos limitados:** como sus tareas se han definido previamente y encajan en un marco temporal específico, los proyectos cuentan con recursos limitados para su ejecución. Esto también determina su complejidad.
- **Resultados concretos:** los proyectos nacen de necesidades concretas y como tal deben servir para sacar a la luz soluciones. Dicho de otra manera, nadie hace un proyecto porque sí; deben apuntar a resultados visibles, materiales o, al menos, que satisfagan la necesidad que dio origen al proceso.

1.1.5 Tipos de proyectos

Existen varios tipos de proyectos como: proyectos de ingeniería, matemáticos y educativos, estos dependen del área o sector donde sus competencias se desarrollarán. Independientemente que la presente investigación se centra específicamente en los proyectos tecnológicos, informáticos, investigación e innovación a continuación, se mencionan distintos tipos tomados de (Pérez, 2021):

- **Proyectos tecnológicos:** son iniciativas que buscan desarrollar soluciones innovadoras y aplicar tecnologías avanzadas para resolver problemas o satisfacer necesidades en diversos campos. Estos proyectos se caracterizan por su enfoque en la creación, implementación y mejora de productos, servicios o sistemas que hacen uso de herramientas tecnológicas. Su objetivo principal es generar valor agregado, impulsar el progreso y fomentar la transformación digital en diferentes sectores, como la industria, la salud, la educación.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

- **Proyectos informáticos:** son aquellos que se enfocan en el desarrollo, implementación y mantenimiento de sistemas de información, software y aplicaciones informáticas. Estos proyectos involucran la planificación, el diseño, la programación y las pruebas de soluciones tecnológicas para abordar necesidades específicas en áreas como la gestión empresarial, la comunicación, la educación, entre otras.
- **Proyectos de investigación:** se centran en la indagación y análisis de áreas o campos específicos, empleando metodologías científicas y dedicando recursos y grupos de trabajo a la generación de conocimiento en un área determinada. Estos proyectos contribuyen al avance del conocimiento en diversas disciplinas y fomentan la publicación y difusión de los resultados obtenidos.
- **Proyecto de innovación:** es un plan estratégico que supone la creación de nuevas ideas, productos o servicios, que conlleven el desarrollo de un área como la educación, la tecnología, los modelos de negocio, la ecología, entre otros. (Significados, 2023)

1.1.6 Dirección Integrada de Proyectos - DIP

Según los procedimientos de *European Cooperation for Space Standardization*:

La Dirección Integrada de Proyectos - DIP - es un proceso dirigido a la ejecución con éxito de un proyecto. La ejecución con éxito se reconoce cuando se cumplen los objetivos del mismo con relación a sus aspectos técnicos, de costo y de programación del plazo. Precisa de la implementación de un proceso de administración mediante la aplicación de métodos y técnicas adecuadas que ayuden a las personas que dirigen los proyectos. La DIP se realiza siguiendo un enfoque estructurado para dirigir el alcance (configuración), la calidad, el plazo, el costo, la organización y la logística del proyecto, en todas las fases de su ciclo de vida y en todos los niveles de su jerarquía (Morales de San Fiel, 2009).

Esta da el marco para la definición y la implementación del proyecto mediante la planificación, la organización, el control (monitoreo) de su ejecución, la evaluación de los resultados y la introducción de acciones de recuperación o de cambios (cuando sea preciso) con el fin de cumplir con los objetivos globales del proyecto. Además, integra las demás

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

funciones principales relacionadas con la ejecución de un proyecto, que pueden ser de ingeniería o de garantía de calidad. Se encarga también de la planificación, la organización y el control de la ejecución de un proyecto, la evaluación de los resultados y la introducción de acciones de recuperación o de cambios cuando sea preciso, con el fin de alcanzar los objetivos propuestos por el mismo (Morales de San Fiel, 2009).

Las características específicas del proyecto y las circunstancias pueden influir sobre las restricciones en las que el equipo de dirección del proyecto necesita concentrarse. Las principales funciones de la dirección en la DIP son: planear, organizar, liderar y controlar (PMI, 2013).

La Guía del PMBOK(*Project Management Body of Knowledge*) (PMI, 2013) describe la naturaleza de los procesos de la dirección de proyectos en términos de la integración entre los procesos, de sus interacciones y de los propósitos a los que responden. Los procesos de la dirección de proyectos se agrupan en cinco categorías conocidas como Grupos de Procesos de la Dirección de Proyectos (o Grupos de Procesos):

1. **Inicio.** Aquellos procesos realizados para definir un nuevo proyecto o nueva fase de un proyecto existente al obtener la autorización para iniciar el proyecto o fase.
2. **Planificación.** Aquellos procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción requerido para alcanzar los objetivos propuestos del proyecto.
3. **Ejecución.** Aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto a fin de satisfacer las especificaciones del mismo.
4. **Monitoreo y Control.** Aquellos procesos requeridos para rastrear, revisar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes.
5. **Cierre.** Aquellos procesos realizados para finalizar todas las actividades a través de todos los Grupos de Procesos, a fin de cerrar formalmente el proyecto o una fase del mismo.
6. Los procesos se superponen e interactúan a través del proyecto o de sus fases. Los procesos se describen en términos de: (Garzón Cabezas & De la Portilla Moncayo, 2011)
 - Entradas (documentos, planes, esquemas, proyectos)

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

- Instrumentos, técnicas y tecnologías (los procesos o mecanismos aplicados a las entradas)
- Salidas (documentos, productos, etc.)

1.1.7 Ecosistema de Innovación del PCT 3CE

El PCT constituyen un mecanismo eficaz para promover la innovación y facilitan el crecimiento y diversificación de nuevas empresas, siendo un instrumento de política de innovación para fomentar el crecimiento y la creación de redes.

Los principales modelos diseñados poseen como elemento común, la proximidad espacial de empresas intensivas en conocimiento, centros de investigación y universidades, lo cual propicia una serie de sinergias y relaciones, que resultan en transferencia de conocimientos y un ambiente favorable para la innovación. El estímulo de esta transferencia, se asegura por medio de una gestión activa del área por parte de profesionales especializados, ofreciendo a centros de investigación y empresas, servicios de alto valor añadido, así como, espacios físicos y servicios básicos. La consolidación de la Universidad de las Ciencias Informáticas (UCI) como un modelo de formación, investigación, producción en el sector de las Tecnologías de la Información y la Comunicación (TIC), propiciaron las condiciones organizativas, de infraestructura y de acceso a capital humano altamente especializado, para la instalación de un PCT en su campus. Como consecuencia de un proceso de análisis, síntesis y experimentación el objetivo del artículo es presentar las bases conceptuales y metodológicas para la creación, organización y gestión del Parque Científico y Tecnológico de La Habana en la UCI, que pueden constituirse en la base de la creación de estructuras similares en el resto del país. Con la aprobación y publicación en la Gaceta Oficial de la República de Cuba, del Decreto Ley 363 del 2019 (Cuba. Consejo de Ministros, 2019), el país contó con un marco legal habilitador para la creación y consolidación de los Parques Científico Tecnológicos (PCT), iniciativas que han proliferado en el mundo, constituyéndose en mecanismos idóneos para acercar los resultados de investigación y desarrollo al sector empresarial, fomentar la transferencia de tecnologías y conocimientos, crear espacios e infraestructura de alta calidad para la prestación de servicios de alto valor añadido y la incubación de nuevas empresas e impulsar el desarrollo local, regional y nacional. Los PCT constituyen un mecanismo eficaz para promover la innovación y facilitan el crecimiento y diversificación de nuevas empresas y agru-

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN paciones industriales en una región o país (Xie, et al., 2018), siendo un instrumento de política de innovadora.

Los esfuerzos por innovar y por desarrollar tecnologías de base científica adoptan muy a menudo la forma de creación y fomento de conglomerados de centros de Investigación y Desarrollo (I+D) y empresas a los que se suele denominar como parques científicos y tecnológicos, o denominaciones similares (Albornoz & Barrere, 2020). Los Parques Científico-Tecnológicos son iniciativas de creación de áreas geográficas delimitadas y destinadas a favorecer el desarrollo y la aplicación de actividades científicas y tecnológicas, con el fin de promover y albergar instituciones de investigación y empresas intensivas en conocimiento, entre las que se estimula y produce la transferencia de tecnologías. Esta transferencia se ejecuta principalmente dentro del área del parque y en su entorno, pero también con instituciones de investigación y empresas localizadas fuera de él. El estímulo para este proceso, ocurre mediante una gestión activa del área por parte de profesionales especializados, ofreciendo a centros de investigación y empresas, servicios de alto valor añadido, así como espacio físico y servicios básicos. Entre los factores claves para el éxito de un Parque Científico y Tecnológico, se considera la propuesta de Cabral-Dahab entre las más abarcadoras, que incorpora, a partir de su priorización, los 3 grupos de actores críticos que intervienen en cada factor (Sanni, et al., 2010):

Determinantes: el staff/nivel de decisión de la administración de las instituciones.

Reactores: aquellos involucrados en la localización, preparación, construcción, administración y expansión del parque.

Ejecutores: aquellos que administran los resultados del parque, que pueden ser la comercialización.

Según el Paradigma de Administración Cabral-Dahab, en un Parque Científico y Tecnológico deberán confluir los factores siguientes (Cabral, 1998):

1. Tener el respaldo de actores económicos poderosos, dinámicos y estables, nacionales y regionales, como un organismo de financiamiento, institución política o universidad local (determinantes).

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

2. Incluir en su gestión una persona activa y con visión (o un grupo de personas), con poder de decisión y con un perfil alto y visible, que es (son) percibida (os) por los actores relevantes de la sociedad como la interfaz entre el mundo académico y la industria, con planes de largo plazo y buena gestión (determinantes).
3. Tener una identidad clara, a menudo expresada simbólicamente, como el nombre elegido por el parque, su logotipo o el discurso de la gestión (determinantes).
4. Estar insertado en una sociedad que permita la protección intelectual de los productos o procesos a través de patentes, de secreto o de cualquier otro medio y tener la capacidad de hacerlo (determinantes).
5. Tener una gestión con experiencia establecida o reconocida en los asuntos financieros, y que ha presentado planes de desarrollo económico a largo plazo
6. Ser capaz de seleccionar o rechazar que empresas entren en el parque. Se espera que el plan de negocios de la empresa debe ser coherente con la identidad del parque (reactores).
7. Tener acceso a la investigación calificada y personal de desarrollo en las áreas de conocimiento en las que el parque tiene su identidad (reactores).
8. Tener la capacidad de proporcionar conocimientos de mercadotecnia y habilidades de gestión a las empresas, especialmente las spin-off, que carecen de este recurso (reactores / ejecutores).
9. Incluir un porcentaje importante de empresas de consultoría, así como empresas de servicios técnicos, incluidos los laboratorios y empresas de control de calidad (ejecutores).
10. Ser capaz de comercializar sus productos y servicios de alto valor agregado (ejecutores).

Los conceptos, tendencias, regularidades y factores claves de éxito analizados de los PCT han permitido construir un modelo de gestión para la operación del PCT de La Habana radicado en la UCI, permitiendo el acercamiento a una cartera de clientes potenciales en el sector estatal y no estatal del país, y con los primeros clientes internacionales. El Parque Científico Tecnológico de La Habana en la UCI, surge como un espacio para la ejecución de proyectos de I+D+i en la rama de las TIC, con capacidades y fortalezas para la conformación

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN en el mediano y largo plazo de un clúster en TI. A partir de la incubación de nuevas entidades, la proporción de beneficios e incentivos, se fortalecerá el tejido empresarial de la industria de aplicaciones y servicios informáticos, que aprovechará la transversalidad del sector de las TIC para generar resultados que impacten en todos los sectores de la economía y la sociedad.

1.1.8 Certificación de proyectos

Para la evaluación de las ideas y/o proyectos, la Sociedad Mercantil se apoya en un Consejo Técnico Asesor (CTA) conformado por expertos. Este Consejo se concibió como un órgano asesor y consultivo, en la toma de decisiones sobre temas estratégicos de carácter técnico, así como para contribuir al óptimo funcionamiento del Parque.

Detallando de forma breve el ciclo de vida de un proyecto, se puede resumir que la idea de investigación debe llegar al Parque en la forma de un Anteproyecto, con un grupo de requisitos mínimos. Este Anteproyecto será evaluado por el CTA, quien propondrá aprobarlo, desecharlo o pedirá al dueño un re análisis con recomendaciones para una nueva evaluación. Al CTA, puede llegar igualmente un proyecto totalmente concebido, si la idea tiene el suficiente grado de madurez y se trabajó con mayor profundidad por el dueño. En este escenario, al producirse la aprobación, se simplifican los pasos posteriores.

- Para un anteproyecto, se deberá presentar al CTA la información siguiente:
- Resumen y principales elementos de la propuesta.
- Objetivo general: definir la acción a ejecutar para dar respuesta a la problemática que da origen a la propuesta.
- Objetivos específicos: relacionar las actividades o tareas que se llevarán a cabo para el cumplimiento del objetivo general en el tiempo requerido. Datos de la (s) persona (s) que presentan la propuesta: especificar si es persona natural o jurídica.
- Participación de los dueños de la propuesta: especificar el % de participación y los aportes de cada uno en recursos humanos, financieros, tecnológicos u otros.
- Alcance de la propuesta: debe redactarse de forma explícita y en correspondencia con los objetivos definidos.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

- Clasificación de la propuesta:
- Investigación básica o aplicada.
- Desarrollo Experimental.
- Innovación.
- Resumen de los resultados esperados y el impacto de la propuesta.
- Cronograma directivo.
- Análisis de Pre-factibilidad.

En caso de aprobación, el CTA propondrá a la Junta Directiva que el Anteproyecto pase a formar parte de la cartera del Parque. El Anteproyecto pasará por un proceso de diseño para convertirse en Proyecto, según los criterios establecidos, que puede ser ejecutado por el dueño de la propuesta o por la Sociedad Mercantil como parte de sus servicios. La información que debe conformar un Proyecto es:

- Análisis de Factibilidad.
- Cronograma detallado.
- Presupuesto (material y financiero).
- Capacidad exportadora.
- Recursos humanos a emplear, roles y competencias.
- Identificación, valoración y tratamiento de riesgos.
- Planes de contingencia.
- Sistema de control, seguimiento y evaluación.
- Luego de diseñado el Proyecto, la Junta Directiva deberá evaluarlo y podrá tener como base los siguientes criterios adicionales:
- Análisis de la calidad de los entregables por etapas.
- Detalle de los resultados esperados y criterios de éxito.
- Evaluación del impacto.

El presupuesto del Proyecto incluirá entre otros conceptos, los valores destinados a la remuneración de los profesores, investigadores, especialistas, estudiantes, directivos y otros trabajadores que en él participen. Posteriormente se efectuará la asignación de recursos

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN humanos, financieros y tecnológicos, gestionados por la Sociedad Mercantil en todos los casos donde se requiera, y el proyecto queda listo para la ejecución. En esa etapa, se gestionarán plazo, costo y calidad, asegurando el cumplimiento de los tiempos pactados, del presupuesto previsto y la obtención del resultado esperado. Los incentivos que proporciona el Parque, juegan un papel decisivo en esta fase.

1.2 Sistemas homólogos

1.2.1 Project.co Es una herramienta completa de gestión de proyectos que ayuda a los equipos a organizar, colaborar y realizar un seguimiento de sus proyectos de manera efectiva, proporcionando una interfaz fácil de usar y diversas características para optimizar la productividad y la colaboración. Ofrece muchas ventajas, es una herramienta sólida de gestión de proyectos, pero siempre hay áreas en las que se puede mejorar. Escuchar los comentarios y sugerencias de los usuarios, ampliar las funcionalidades del plan gratuito, aumentar las integraciones nativas, proporcionar opciones de personalización adicionales y considerar características más avanzadas para proyectos complejos son algunas formas de abordar las posibles desventajas y mejorar la experiencia de los usuarios.

1.2.2 Akademos Por otro lado, es una herramienta diseñada para mejorar los procesos académicos en una institución. Aunque tiene módulos para la planificación curricular, registro de estudiantes, gestión de proyectos y generación de informes, su enfoque principal se encuentra en los procesos académicos y no aborda específicamente la gestión de proyectos y empleados.[CITATION Gru19 \l 8192](Infanta Costa, 2019)

1.2.3 GESPRO Se destaca como una suite de gestión de proyectos basada en la web. Aunque proporciona funcionalidades para la planificación, seguimiento y control de proyectos, su enfoque se centra en la toma de decisiones y el soporte a nivel de proyecto y entidad de ejecución. Sin embargo, no ofrece una gestión integral de empleados ni características específicas de gestión de proyectos como la colaboración en línea, seguimiento de tareas, asignación de recursos, gestión de dependencias, etc.

1.2.4 Jira Software Es una popular herramienta de gestión de proyectos y seguimiento de problemas desarrollada por Atlassian. Facilita la gestión ágil de proyectos, el seguimiento de problemas y la colaboración entre equipos. Proporciona opciones de personalización, informes detallados y una integración sólida con otras herramientas de desarrollo. Posee Curva de aprendizaje pronunciada, interfaz abrumadora, configuración y administración complejas, limitaciones en la asignación de roles y permisos. Destacar que algunas de estas deficiencias pueden abordarse mediante la configuración y personalización de Jira Software, así como mediante el uso de complementos y extensiones disponibles en el Atlassian Marketplace. Además, Atlassian continúa realizando mejoras y actualizaciones en la herramienta para abordar estas deficiencias y mejorar la experiencia del usuario.

1.2.5 Análisis comparativos de los sistemas homólogos

Tabla 1: Comparación de los sistemas homólogos. Fuente: Elaboración propia

	Sistemas			
Características	Project.co	Akadememos	GESPRO	Jira Software
Gestión de proyectos de investigación	Sí	No	Sí	Sí
Gestión de recursos humanos	Sí	Sí	No	No
Gestión de periodos de certificación	Sí	No	No	No
Gestión de la Documentación	Si	Sí	No	No

Curva de aprendizaje pronunciada	No	No	No	Sí
---	----	----	----	----

Todos los sistemas analizados son válidos en el entorno para el que fueron desarrollados, sin embargo, no son útiles para la solución deseada. Aunque todos exportan estadísticas y manejan información de usuarios, la mayoría no tiene administración de información de proyectos ni de empleados, no permiten hacer las certificaciones de los proyectos de investigación según la metodología del PCT, ni generar los documentos asociados a la misma. De ahí que no sea posible utilizarlos, ni personalizar a las características propias del sistema requerido. Dado que ninguno de estos sistemas cumple con todas las características necesarias para la solución deseada, se concluye que es necesario desarrollar un nuevo sistema que incorpore estas funcionalidades faltantes. Al desarrollar un sistema personalizado, se puede tener en cuenta las funcionalidades presentes en los sistemas anteriores y adaptarlas según las especificaciones y necesidades específicas del sistema requerido. como gestión de usuarios, exportar datos a PDF, diseño del sitio, entre otros. Al desarrollar un nuevo sistema que incorpore todas estas características, se puede proporcionar una solución integral y personalizada que satisfaga las necesidades de gestión de proyectos y empleados de manera efectiva.

1.2.6 GAPID

Tiene como objetivo principal desarrollar un sistema informático para la gestión administrativa de Programas y Proyectos de Ciencia, Tecnología e Innovación. Los objetivos específicos incluyen el diseño e implementación de flujos de trabajo para la gestión de Programas, Proyectos y Recursos humanos, así como el desarrollo de módulos para la certificación de actividades/resultados, gestión documental, gestión del conocimiento, planificación de resultados, gestión económica y control.

La propuesta abarca la gestión integral de Programas, Proyectos y Recursos humanos, siguiendo las indicaciones metodológicas y resoluciones del CITMA vigentes. Además, se contempla la certificación de actividades/resultados desde el sistema, la gestión de la documentación relacionada con los programas y proyectos, la gestión del conocimiento de los resultados obtenidos, y la planificación de recursos y actividades para evaluar y controlar la ejecución de los proyectos.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

El proyecto se clasifica como Innovación Tecnológica, ya que busca desarrollar nuevos servicios y procesos. También se enmarca en el Desarrollo Tecnológico, al promover la transferencia de tecnología. Además, se espera generar patentes y ofrecer servicios científico-tecnológicos orientados a la calidad, organización y agilidad en los servicios, así como a la solución de problemas técnicos-económicos.

La implementación de esta solución permitirá aumentar la productividad y eficiencia operativa de los usuarios, así como mejorar la experiencia del cliente al digitalizar el proceso de gestión integral de Programas y Proyectos. Se reducirán los tiempos necesarios para la documentación y gestión de resultados, minimizando errores en el traspaso de información. Asimismo, se facilitará la centralización de los expedientes de proyecto y se logrará un ahorro en gastos de tecnologías de la información y contratación de soluciones complejas. La solución será adaptable a las necesidades de los usuarios, con la posibilidad de ajustar y ampliar servicios según sea necesario, y permitirá la incorporación de nuevas funcionalidades en el futuro.

1.3 Ambiente de Desarrollo

Se exponen todas las metodologías, herramientas, lenguajes y tecnologías que se utilizaron para materializar en un producto de *software* los elementos que se abordan en el presente capítulo de la investigación. Se tienen en cuenta la portabilidad, la facilidad de instalación y la facilidad de cambio. Por otra parte, la asimilación del lenguaje de desarrollo, las librerías y las tecnologías empleadas por el equipo de desarrollo de la plataforma, la estabilidad de los requerimientos, así como el modelo de desarrollo constituyen robustos factores ambientales que inciden de manera directa en el desarrollo de la solución informática. Es un proceso que se suele seguir a la hora de diseñar una solución o un programa específico. Tiene que ver, por tanto, con la comunicación, la manipulación de modelos y el intercambio de información y datos entre las partes involucradas. Una metodología de desarrollo de *software*, es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información, a través de un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software (Villegas La Rosa, 2016).

1.3.1 Metodología AUP-UCI, Escenario 4

El proyecto se ha desarrollado siguiendo las especificaciones de la metodología AUP-UCI en su escenario 4. Esto implica que la investigación llevada a cabo se ha guiado por dicha metodología, la cual permite la integración de los artefactos generados en el expediente del proyecto. Es importante destacar que toda metodología debe adaptarse a las características específicas de cada proyecto, incluyendo el equipo de desarrollo y los recursos disponibles. Por lo tanto, se exige que el proceso sea configurable y se ajuste de acuerdo a las necesidades particulares de la situación.

Tabla 2: Fases de la variación de AUP para la UCI

Fase AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración, Construcción y Transición	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN

	Cierre	En esta fase se analizan tanto los resultados del proyecto como la ejecución y se realizan las actividades formales de cierre del proyecto.
--	--------	---

La metodología de software AUP-UCI a partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos, como son: CUN (Casos de uso del negocio), DPN (Descripción de proceso de negocio) o MC (Modelo conceptual) y existen tres formas de encapsular los requisitos, los cuales son: CUS (Casos de uso del sistema), HU (Historias de usuario), DRP (Descripción de requisitos por proceso), surgen cuatro escenarios para modelar el sistema en los proyectos, los cuales son:

- Escenario No 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- Escenario No 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario No 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario No 4: Proyectos que no modelen negocio, solo pueden modelar el sistema con HU (Sánchez,2015)

Se decide hacer una variación de la metodología AUP-UCI, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Escenario No 4.

Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una Historia de usuarios (HU) no debe poseer demasiada información. Todas las disciplinas antes definidas (desde

Modelado de negocio hasta Pruebas de Aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen Iteraciones y se obtengan resultados incrementales.

1.3.2 Herramienta CASE

Visual Paradigm 17.2 Visual Paradigm es una herramienta CASE multiplataforma que contribuye al desarrollo de sistemas de software fiables, mediante un enfoque orientado a objetos. Soporta el ciclo completo de desarrollo de software y permite su documentación en diferentes formatos, empleando UML como lenguaje de modelado (Visual Paradigm, 2021).

Para la gestión de datos:

DB Browser: es una herramienta de código abierto que proporciona una interfaz gráfica intuitiva para administrar y explorar bases de datos SQLite. Es una aplicación ligera y fácil de usar, diseñada específicamente para trabajar con archivos de bases de datos SQLite. DB Browser permite a los usuarios crear, modificar y eliminar tablas, ejecutar consultas SQL, importar y exportar datos, y visualizar la estructura de la base de datos de manera interactiva. Además, ofrece funcionalidades avanzadas como la capacidad de generar informes y gráficos a partir de los datos almacenados. Con su enfoque amigable para el usuario, DB Browser es una herramienta conveniente y efectiva para la gestión y análisis de bases de datos SQLite, siendo especialmente útil para aquellos que no están familiarizados con la línea de comandos y prefieren una interfaz gráfica.

1.3.3 Marco de Trabajo

Django 4.2: es un marco de trabajo (framework) web de Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Es gratis y de código abierto. Django incluye docenas de extras que puede usar para manejar tareas comunes de desarrollo web. Se encarga de la autenticación de usuarios, la administración de contenido, los mapas del sitio, las fuentes RSS y muchas más tareas, desde el primer momento.

Bootstrap 5.3: es un marco front-end gratuito para un desarrollo web más rápido y fácil. Incluye plantillas de diseño basadas en HTML y CSS para tipografía, formularios, botones, tablas, navegación, modales, carruseles de imágenes y muchos otros, así como complementos de JavaScript opcionales. Bootstrap también le brinda la capacidad de crear fácilmente diseños receptivos[CITATION w3s19 \l 23562](w3schools, 2023).

1.3.4 Lenguaje de programación

Python 3.12: es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. Es multiplataforma, está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que si no utilizamos librerías específicas de cada plataforma nuestro programa podrá correr en todos estos sistemas sin grandes cambios[CITATION ICT22 \l 21514](ictea, 2023).

JavaScript: es un lenguaje de programación interpretado, lo que significa que no necesita ser compilado. Orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Proviene del Java y se utiliza principalmente para la creación de páginas web. El JavaScript es una mezcla entre el Java y el HTML, es un lenguaje que se incorpora dentro de la página web, formando parte del código HTML (MDN web docs, 2023).

HTML 5 Por sus siglas en inglés de (*HyperText Markup Language*) es el lenguaje básico de la web para crear documentos y aplicaciones para que los desarrolladores lo utilicen en cualquier lugar. HTML en su versión 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Es un lenguaje simple que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos (Bos, 2023).

CSS 3: es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN complejas. CSS 3 funciona mediante módulos son sólo categorías en las que se pueden dividir las modificaciones que hacemos al aspecto de nuestro sitio web (Bos, 2023).

SQLite: es una herramienta de gestión de bases de datos relacionales que se destaca por su simplicidad, eficiencia y portabilidad. Se implementa como una biblioteca y no requiere un servidor independiente, lo que facilita su configuración y uso. A pesar de su tamaño reducido, SQLite ofrece funcionalidades esenciales de gestión de bases de datos, como soporte para SQL, creación de tablas, consultas y transacciones. Su rendimiento optimizado y su capacidad para funcionar en diferentes sistemas operativos lo convierten en una opción versátil para una variedad de aplicaciones y dispositivos. En resumen, SQLite es una solución práctica y eficaz para el manejo de bases de datos relacionales en entornos diversos.

1.3.5 Entorno de desarrollo integrado

Visual Studio Code:

Se decidió utilizar Visual Studio Code el cual es un editor de código fuente ligero pero potente que se ejecuta en su escritorio desarrollada por Microsoft y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para Python, JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como .NET y Unity) (Microsoft, [s.f.(a)]). Visual Studio Code combina la simplicidad de un editor de código fuente con potentes herramientas para desarrolladores, como la finalización y depuración de código de IntelliSense. El ciclo de edición-construcción-depuración fluido significa menos tiempo jugando con su entorno y más tiempo ejecutando sus ideas. En esencia, Visual Studio Code presenta un editor de código fuente ultrarrápido, perfecto para el uso diario. Con soporte para cientos de idiomas, VS Code lo ayuda a ser productivo al instante con resaltado de sintaxis, coincidencia de corchetes, sangría automática, selección de cuadros, fragmentos y más. Para la codificación seria, a menudo se beneficiará de herramientas con más comprensión del código que solo bloques de texto. Visual Studio Code incluye soporte integrado para la finalización de código de IntelliSense, comprensión y navegación de código semántico enriquecido, refactorización de código. Y cuando la codificación se vuelve difícil, los difíciles se vuelven depuradores. La depuración

CAPÍTULO I: FUNDAMENTOS TEÓRICOS METODOLÓGICOS DE LA INVESTIGACIÓN
suele ser la característica que más extrañan los desarrolladores en una experiencia de codificación más eficiente, así que lo hicimos realidad. Visual Studio Code incluye un depurador interactivo, por lo que puede recorrer el código fuente, inspeccionar variables, ver pilas de llamadas y ejecutar comandos en la consola. (Microsoft, [s.f.(b)])

Conclusiones del capítulo

- La gestión administrativa de los proyectos de innovación y desarrollo es un problema actual dentro de la ejecución de los proyectos de ciencia en Cuba porque los investigadores y los jefes de proyectos dedican mucho tiempo a procedimientos burocráticos más allá de guiar y dirigir proyectos
- Los sistemas homólogos que se identifican cumplen o solucionan parcialmente la problemática identificada porque algunos se especializan en diferentes temáticas de las que se abordan en la investigación, pero ninguno cumple cabalmente lo necesario
- El ambiente de desarrollo identificado permite una solución de manera ágil y obtener un resultado de manera integrable a la plataforma GAPID

CAPÍTULO II: Análisis y diseño del Módulo de Gestión de proyectos

En el actual capítulo se describe el procedimiento y los elementos que se tuvieron en cuenta para el desarrollo del sitio web. Se presentan los conceptos asociados al modelo conceptual y la representación formal del mismo. Se identifican los requisitos no funcionales y funcionales. Además, se emplea la técnica de descripción de requisitos, acorde con la metodología empleada.

2.1 Propuesta de Solución

Todo proceso de negocio es legislativamente dependiente de las reglas de negocio predefinidas para el diseño del proceso. Su ejecución se caracteriza por el empleo de datos manipulados por un conjunto de actividades en las que participan agentes o actores acorde a un flujo de trabajo.

Para el cumplimiento de los objetivos planteados se propone el desarrollo de un sistema para la gestión y administración de la información de los proyectos de investigación desarrollo e innovación del Parque Científico Tecnológico 3CE. El sistema debe contar con 2 módulos, el primero que abarque la gestión de proyectos con miembros asociados, el segundo encargado de la certificación de un período de evaluación y la gestión documental.

El módulo de gestión de proyectos con miembros asociados debe permitir añadir, vincular, editar y quitar a un miembro de un proyecto. Se recogen los datos personales y datos profesionales del miembro para realizar dichas acciones

Tabla 3:Roles de los usuarios. Fuente: Elaboración propia

Nombre	Descripción
Administrador	Encargado de gestionar los usuarios, grupos, permisos y cerrar los proyectos.

Jefe de proyecto	Encargado de gestionar la información de los proyectos y miembros.
Gestor de proyecto	Encargado de certificar, evaluar, subir y descargar documentos asociados a un proyecto

2.2 Requisitos del Sistema

Para desarrollar cualquier producto informático es necesario primeramente una correcta etapa de planificación comenzando con una efectiva recolección de requisitos. Esta es una de las etapas más importantes del proceso, y se puede decir que de esta etapa depende la mayor parte del éxito del proyecto y la satisfacción del cliente. Los requisitos pueden ser clasificados en dos grupos: requisitos funcionales y no funcionales.

2.2.1 Requisitos Funcionales

Los requisitos funcionales definen las acciones que debe realizar el sistema. Son capacidades o condiciones que el sistema debe cumplir, cómo debe comportarse en situaciones específicas. En algunos casos también pueden plantear explícitamente qué no debe hacer el sistema (Pressman, 2010).

Tabla 4: Requisitos Funcionales

	Crear Pro- yecto	Me Permite como adminis- trador crear un nuevo Proyecto	Complejidad Media	Prioridad Alta
RF1	Desactivar proyecto	Permite cerrar el el ciclo de un proyecto	Baja	Media

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

RF2	Extender proyecto	Permite extender la fecha de entrega del proyecto	Media	Media
RF3	Buscar proyecto	Permite buscar un proyecto en la lista de proyectos	Baja	Baja
RF4	Desactivar un proyecto	Permite desactivar un proyecto inhabilitándolo temporalmente	Baja	Alta
RF5	Editar un proyecto	Permite modificar un proyecto registrado en el sistema	Media	Alta
RF6	Listar proyectos	Permite mostrar una lista con todos los proyectos	Baja	baja
RF7	Insertar RRHH	Permite agregar un RH(Recurso Humano)	Baja	Alta
RF8	Editar datos de un RRHH	Permite editar los datos de un RH(Recurso Humano)	Baja	Alta
RF9	Listar RRHH	Permite mostrar	Baja	Baja

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

		una tabla con todos los RH(Recurso Humano)		
RF10	Deshabilitar RRHH	Permite deshabilitar un RH (Recurso Humano)	Baja	Media
RF11	Vincular RRHH a un Proyecto	Permite vincular un RH(Recurso Humano) a un proyecto que este no pertenezca	Media	alta
RF12	Crear Tarifas del proyecto	Permite crear las tarifas de cada proyecto	Media	Alta
RF13	Modificar Tarifas del proyecto	Permite modificar las tarifas de un proyectos	Media	media
RF14	Listar Tarifas del proyecto	Permite mostrar una lista de las tarifas totales del proyecto	Baja	baja

2.2.2 Requisitos No Funcionales

Los requisitos no funcionales son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.

Software

- Para el uso del sistema se necesita un navegador web cualquiera.

Accesibilidad

- Se necesita tener un usuario autorizado en el sistema para acceder a este módulo.

Portabilidad

- Se puede tener acceso al módulo a través del sitio web desde cualquier sistema operativo.

Usabilidad

- Debe tener una interfaz amigable e intuitiva, que pueda ser usada por cualquier usuario sin conocimientos especiales.

Seguridad

- La información que se maneje se mantendrá segura y confidencial, evitando el acceso no autorizado y la divulgación.
- Disponer de un mecanismo de autenticación de usuarios seguro.

Integridad

- Debe tener la capacidad de recuperación ante fallos y de ser posible prevenirlos
- La información debe ser protegida cuidadosamente contra la corrupción de datos.
- Se debe aplicar un mecanismo de copias de seguridad, o de respaldo que puedan restaurar el sistema en caso de fallo crítico y no se pierdan los datos.

2.2.3 Historias de usuario

Las Historias de Usuario (HU) son artefactos generados por metodologías ágiles como XP, que sustituyen a los documentos de especificación funcional, y a los casos de uso. Son utilizadas con el fin de especificar los requisitos del software. Estas son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. A través de un conjunto de tablas se describen brevemente las características deseadas. Contienen la información suficiente para que los desarrolladores puedan producir una estimación razonable del esfuerzo necesario para su implementación (Joskowicz, 2008).

La prioridad en el negocio:

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

- Alta: se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.
- Media: se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
- Baja: se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

El riesgo en su desarrollo:

- Alta: cuando en la implementación de las HU se considera la posible existencia de errores que conlleven a la inoperatividad del código.
- Media: cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.
- Baja: cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU son representadas mediante tablas divididas por las siguientes secciones:

- Número: esta sección representa el número, incremental en el tiempo, de la HU que se describe.
- Nombre de HU: identifica la HU que se describe entre los desarrolladores y el cliente.
- Modificación de HU número: sección que representa si la HU se le realizó alguna modificación con respecto al estado anterior.
- Usuario: los programadores responsables de la HU.
- Iteración asignada: número de la iteración donde va a desarrollarse la HU.
- Prioridad en negocio: se le otorga una prioridad (Alta, Media, Baja) a las HU de acuerdo a la necesidad de desarrollo.
- Riesgo en desarrollo: se le otorga una medida de (Alto, Medio, Bajo), a la ocurrencia de errores en el proceso de desarrollo de la HU.
- Puntos estimados: es el tiempo estimado en semanas que se demorará el desarrollo de la HU.
- Puntos reales: representa el tiempo que se demoró en realidad el desarrollo de la HU.
- Descripción: breve descripción de la HU.
- Observaciones: señalamiento o advertencia del sistema.

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

Las HU también son utilizadas para estimar el tiempo que el equipo de desarrollo tomará para realizar las entregas, deben poder ser programadas en un tiempo entre una y tres semanas. En una entrega se puede desarrollar una o más HU, esto depende solo del tiempo que demore la implementación de cada una de las mismas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia.

Se identificaron y elaboraron 15 historias de usuarios que corresponden a las diferentes funcionalidades solicitadas por el cliente.

Tabla 5: Crear proyecto

Historia de usuario	
Numero: HU 1	Nombre Historia de usuario: Crear Proyecto
Usuario: Roddy Javier García Mayea	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 2
Riesgo en desarrollo: Bajo	Puntos reales: 2
La capacidad de crear un proyecto en el sistema, con todas las tareas y actividades.	
Observaciones:	

Insertar proyecto		
Nombre del proyecto Escriba el nombre completo	Nombre del jefe del proyecto Escriba el nombre completo	Nombre del dueño del proyecto Escriba el nombre completo
Nombre abreviado Escriba las siglas identificativas	Número del contrato Número del contrato	OAEC Órganos y Organismos de la Administración Central del Estado
Resumen del proyecto Breve descripción o resumen del proyecto	Innovación tecnológica Ninguna	Domicilio legal del dueño Domicilio Legal
Código del proyecto Código del proyecto	Desarrollo tecnológico Ninguna	Email del dueño del proyecto Dirección de correo electrónico
Director de la entidad Nombre y apellidos del director de la entidad	Creación científica Ninguna	Código REEUP Código REEUP
Logo del proyecto Examinar... No se ha seleccionado ningún archivo.	Servicios tecnológicos Ninguna	NIT Número de Identificación Tributaria
Fecha de inicio dd/mm/aaaa	Entidad ejecutora principal Entidad ejecutora principal a la que pertenece	Telefax del dueño Número de telefax
Fecha de terminación dd/mm/aaaa	Entidad ejecutora participante Entidad ejecutora participante a la que pertenece	Suplemento Suplemento
Aceptar Cancelar		

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

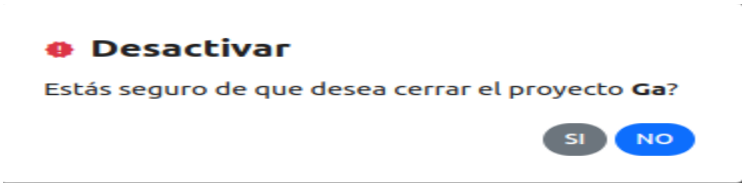
Historia de usuario	
Numero: HU 2	Nombre Historia de usuario: Desactivar Proyecto
Usuario: Roddy Javier García Mayea	Iteración asignada: 3
Prioridad en negocio: Alta	Puntos estimados: 2
Riesgo en desarrollo: Bajo	Puntos reales: 2
Descripción: Quiero tener la capacidad de desactivar un proyecto en el sistema, para indicar que todas las tareas y actividades relacionadas con el proyecto han sido completadas o queda suspendido indefinidamente.	
Observaciones: 	

Tabla 6:Desactivar proyecto

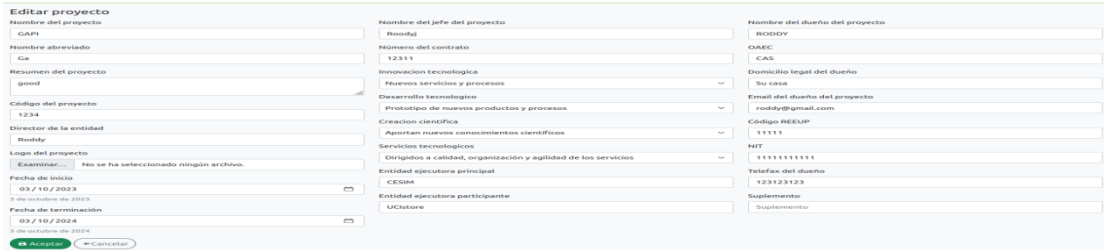
Historia de usuario	
Numero: HU 6	Nombre Historia de usuario: Editar proyecto
Usuario: Julio T. Nerey Villanueva	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 2
Riesgo en desarrollo: Bajo	Puntos reales: 2
Descripción: quiero poder modificar la información de un proyecto registrado en el sistema, para actualizar los detalles del proyecto según sea necesario.	
Observaciones: 	

Tabla 7: Editar Proyecto

2.3 Fase diseño

Para que un *software* sea desarrollado con calidad es necesario establecer desde el inicio una arquitectura que describa sus principios fundamentales, garantizando robustez y escalabilidad. La arquitectura de *software* define la estructura del sistema, constituida por componentes con funciones específicas que interactúan entre sí (Hofmeister et al., 2000).

Para la implementación de la solución será utilizado el Modelo – Vista – Plantilla (MVP). Este patrón tiene tres componentes esenciales:

El **modelo**: representa la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.

La **vista**: la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre los modelos y las plantillas

La **plantilla** es la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.



Figura 1: Ejemplo de Modelo Vista Plantilla Fuente: Elaboración propia

2.3.2 Patrones de diseño

Los patrones de diseño no son códigos, son la forma de documentar las mejores prácticas y lecciones que se han aprendido al momento de resolver un determinado problema complejo, que se encuentra dentro de un dominio de diseño concreto en el desarrollo del software, la arquitectura o el diseño; por lo que esos patrones tienen la capacidad de brindar soluciones reutilizables. Los patrones de diseño pueden ser de dos tipos: GRASP (Patrones Generales de *Software* para Asignación de Responsabilidades) y GoF (Pandilla de Cuatro). A continuación, se presentan los patrones utilizados en el sistema

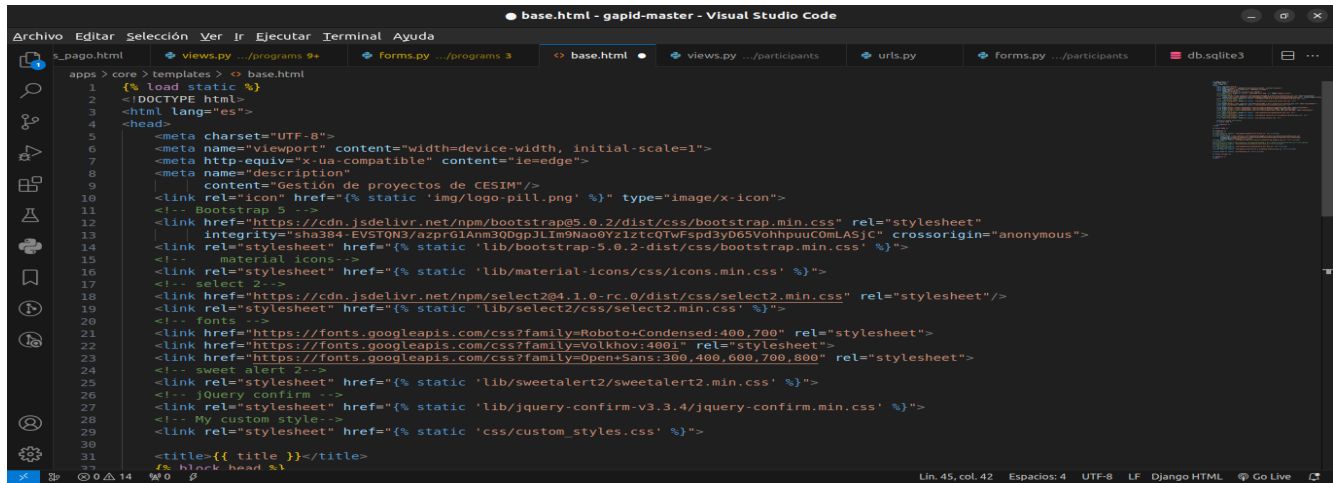
Los patrones de diseño GoF son patrones que describen soluciones simples y elegantes a problemas específicos en el diseño de software. Son un conjunto de 23 patrones de diseño que están divididos en 3 categorías denominadas creacionales, estructurales y de comportamiento. A continuación, se evidencian los patrones de tipo GoF utilizados en el sistema de recomendación

Estructurales: Adaptador

La vista del módulo implementa el uso de las plantillas de *Django*, las cuales contienen variables o expresiones que se reemplazan por valores cuando se evalúa la plantilla, y las etiquetas controlan la lógica de la plantilla. A continuación, se ilustran ejemplos de su

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

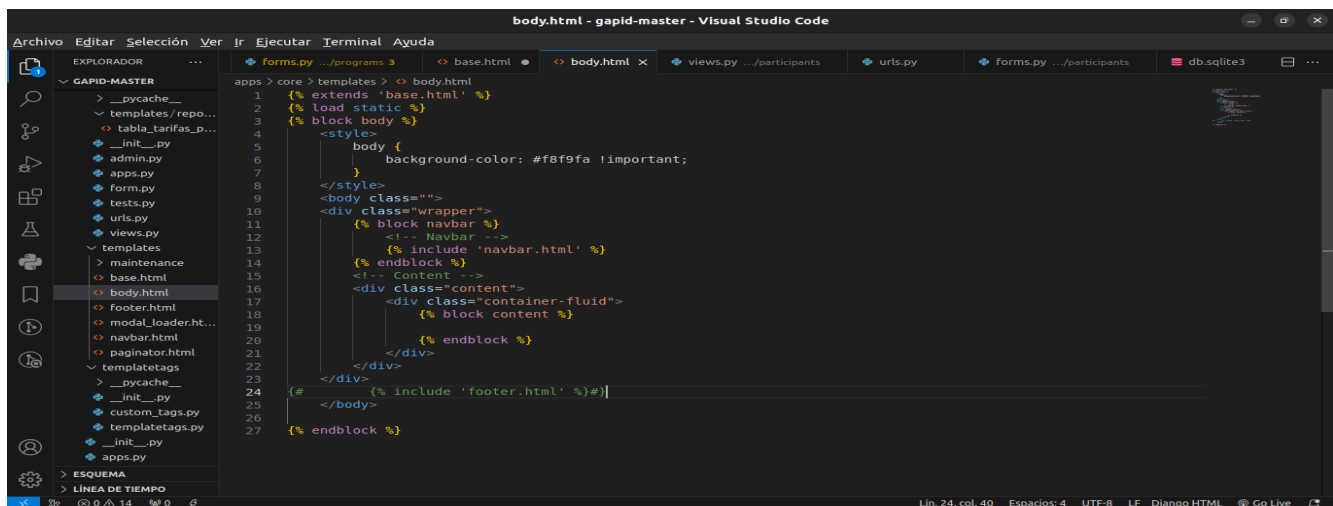
utilización:



```
1 {% load static %}
2 <!DOCTYPE html>
3 <html lang="es">
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <meta http-equiv="x-ua-compatible" content="ie=edge">
8     <meta name="description"
9         content="Gestión de proyectos de CESIM"/>
10    <link rel="icon" href="{% static 'img/logo-pill.png' %}" type="image/x-icon">
11    <!-- Bootstrap 5 -->
12    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
13        integrity="sha384-EVSTQN3/azprlAmm300g3Jl1m9Nao8Vz1t0TlwFspj065V0hpuuComLASjc" crossorigin="anonymous">
14    <link rel="stylesheet" href="{% static 'lib/bootstrap-5.0.2-dist/css/bootstrap.min.css' %}">
15    <!-- material icons -->
16    <link rel="stylesheet" href="{% static 'lib/material-icons/css/icons.min.css' %}">
17    <!-- select 2 -->
18    <link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/css/select2.min.css" rel="stylesheet"/>
19    <link rel="stylesheet" href="{% static 'lib/select2/css/select2.min.css' %}">
20    <!-- fonts -->
21    <link href="https://fonts.googleapis.com/css?family=Roboto:Condensed:400,700" rel="stylesheet">
22    <link href="https://fonts.googleapis.com/css?family=Volkhov:4001" rel="stylesheet">
23    <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700,800" rel="stylesheet">
24    <!-- sweet alert 2 -->
25    <link rel="stylesheet" href="{% static 'lib/sweetalert2/sweetalert2.min.css' %}">
26    <!-- jquery confirm -->
27    <link rel="stylesheet" href="{% static 'lib/jquery-confirm-v3.3.4/jquery-confirm.min.css' %}">
28    <!-- My custom style -->
29    <link rel="stylesheet" href="{% static 'css/custom_styles.css' %}">
30
31    <title>{{ title }}</title>
32    <!-- block head -->
33    {% block head %}
```

Figura 2: Fragmento de código del patrón adaptador Fuente: Elaboración propia.

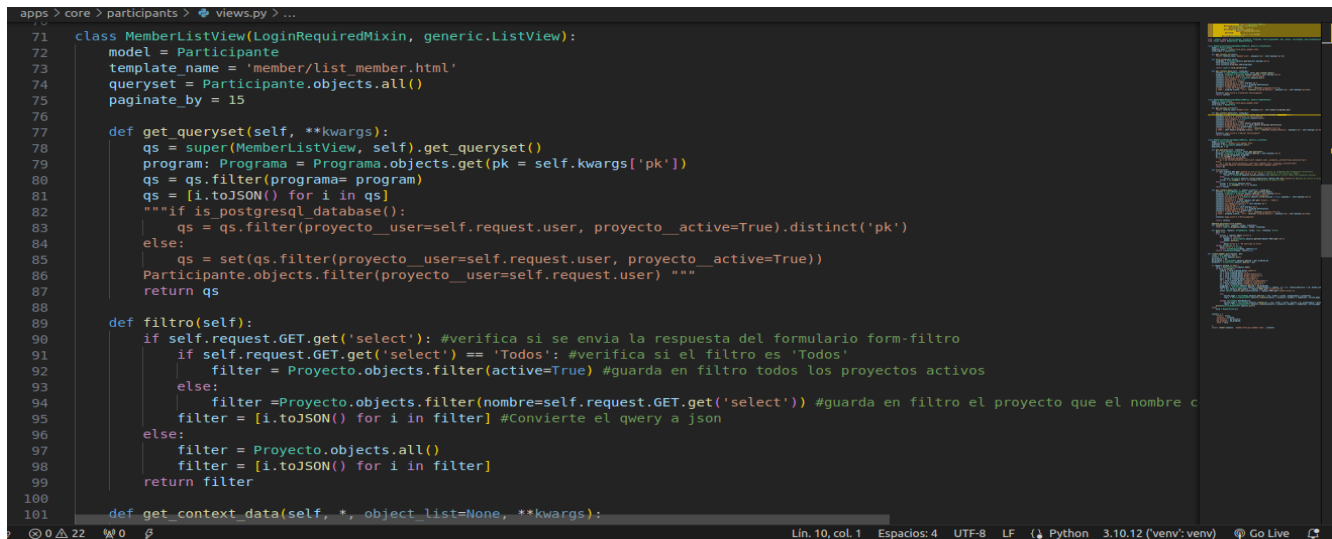
La herencia de plantillas permite crear un “esqueleto” de plantilla base (en el módulo esta plantilla esqueleto es base.html) que contiene todos los elementos comunes del módulo y define bloques que las plantillas descendientes pueden sustituir; tal es el caso de los bloques definidos dentro de las etiquetas { % block ... % } ... { % endblock % }



```
1 {% extends 'base.html' %}
2 {% load static %}
3 {% block body %}
4     <style>
5     <body {
6         background-color: #f8f9fa !important;
7     }
8     </style>
9     <body class="">
10    <div class="wrapper">
11        <!-- block navbar -->
12        <!-- navbar -->
13        <!-- include 'navbar.html' -->
14        <!-- endblock -->
15        <!-- Content -->
16        <div class="content">
17            <div class="container-fluid">
18                <!-- block content -->
19                <!-- endblock -->
20            </div>
21        </div>
22    </div>
23    <!-- include 'footer.html' -->
24    </body>
25    <!-- endblock -->
```

Figura 3: Fragmento de código del uso del patrón decorador. Fuente: Elaboración propia.

Comportamiento: Iterador El patrón iterador, permite el acceso al contenido de una estructura sin exponer su representación interna. Además, diferentes iteradores pueden presentar diferentes tipos de recorrido sobre la estructura. El uso de este patrón se evidencia en la clase *MemberListView*, donde se hace uso de las variables *queryset* y *qs* para acceder a los elementos del objeto Miembro, facilitando las iteraciones para la obtención de cada elemento contenido en los objetos.



```

71 class MemberListView(LoginRequiredMixin, generic.ListView):
72     model = Participante
73     template_name = 'member/list_member.html'
74     queryset = Participante.objects.all()
75     paginate_by = 15
76
77     def get_queryset(self, **kwargs):
78         qs = super(MemberListView, self).get_queryset()
79         program: Programa = Programa.objects.get(pk = self.kwargs['pk'])
80         qs = qs.filter(programa= program)
81         qs = [i.toJSON() for i in qs]
82         """if is_postgresql_database():
83             qs = qs.filter(proyecto__user=self.request.user, proyecto__active=True).distinct('pk')
84         else:
85             qs = set(qs.filter(proyecto__user=self.request.user, proyecto__active=True))
86             Participante.objects.filter(proyecto__user=self.request.user) """
87         return qs
88
89     def filtro(self):
90         if self.request.GET.get('select'): #verifica si se envia la respuesta del formulario form-filtro
91             if self.request.GET.get('select') == 'Todos': #verifica si el filtro es 'Todos'
92                 filter = Proyecto.objects.filter(active=True) #guarda en filtro todos los proyectos activos
93             else:
94                 filter =Proyecto.objects.filter(nombre=self.request.GET.get('select')) #guarda en filtro el proyecto que el nombre c
95                 filter = [i.toJSON() for i in filter] #Convierte el query a json
96             else:
97                 filter = Proyecto.objects.all()
98                 filter = [i.toJSON() for i in filter]
99             return filter
100
101     def get_context_data(self, *, object_list=None, **kwargs):

```

Figura 4: Fragmento de código del patrón iterador. Fuente: Elaboración propia.

Patrones GRASP

En el diseño del sistema de recomendación de servicios se tuvieron en cuenta los patrones GRASP, que describen los principios fundamentales de la asignación de responsabilidades a objetos:

Experto: determina cuál es la clase que debe asumir una responsabilidad a partir de la información que posee cada una. Este patrón se evidencia en la clase Proyecto, ya que contiene las funcionalidades necesarias para acceder a la información de un proyecto.

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

```
86 class Proyecto(models.Model):
87     program = models.ForeignKey(Programa, on_delete=models.SET_NULL, null=True, blank=True)
88     user = models.ForeignKey(User, on_delete=models.SET_NULL, null=True, blank=True)
89     nombre = models.CharField(max_length=200, verbose_name='Nombre del proyecto',
90                             validators=[MinLengthValidator(2, 'El nombre debe tener mínimo 2 caracteres.'),
91                                     validate_upper_nombre])
92     director = models.CharField(max_length=200, verbose_name='Director de la entidad')
93     abreviacion = models.CharField(max_length=100, verbose_name='Nombre abreviado',
94                                 help_text='Debe ser todo letras y números opcionales',
95                                 validators=[
96                                     MinLengthValidator(2, 'El nombre corto debe tener mínimo 2 caracteres.')]
97     logo = models.ImageField(upload_to='fotos/', null=True, blank=True, verbose_name='Logo del proyecto')
98     codigo = models.CharField(max_length=100, verbose_name='Código del proyecto', unique=True,
99                             validators=[MinLengthValidator(4, 'El código debe tener mínimo 4 caracteres.')]
100     tipo = models.CharField(max_length=100, verbose_name='Tipo de proyecto', choices=PROJECT_TYPE, blank=True, null=True)
101     entity_principal = models.CharField(max_length=100, verbose_name='Entidad ejecutora principal',
102                                     validators=[MinLengthValidator(4, 'La Entidad ejecutora principal debe tener mínimo 4 caracteres.')]
103     entity_participant = models.CharField(max_length=100, verbose_name='Entidad ejecutora participante', null=True,
104                                         blank=True,
105                                         validators=[MinLengthValidator(4, 'La Entidad ejecutora participante debe tener mínimo 4 caracteres.')]
106     resumen = models.CharField(max_length=500, verbose_name='Resumen del proyecto')
107     active = models.BooleanField(default=True, verbose_name='Activo')
108     start_date = models.DateField(verbose_name='Fecha de inicio', default=datetime.now)
109     end_date = models.DateField(verbose_name='Fecha de terminación', default=datetime.now)
110
111     class Meta:
112         verbose_name = 'Proyecto'
113         verbose_name_plural = 'Proyectos'
```

Figura 5: Fragmento de código del patrón experto. Fuente: Elaboración propia.

Alta cohesión: se basa en asignar a las clases responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad, evitando así que una clase sea la única responsable de muchas tareas en áreas funcionales muy heterogéneas. Django permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase *RecursosHumanos*, que se encarga de definir atributos comunes entre las entidades Proyecto y Miembro.

```
class RecursosHumanos(models.Model):
    proyecto = models.ForeignKey(Proyecto, on_delete=models.CASCADE)
    miembro = models.ForeignKey(Miembro, on_delete=models.CASCADE)
    cargo = models.CharField(max_length=10, verbose_name='Cargo que ocupa', choices=OCCUPATION, blank=True, null=True)
    institucion = models.CharField(max_length=100, verbose_name='Institución')
    clasificador_entidad = models.CharField(max_length=10, help_text='1 si es Empresa, 0 si es Presupuestada',
                                           verbose_name='Clasificar de entidad', choices=ENTITY_CLASSIFICATION)

    # Salario por Participación en el Proyecto
    porcentaje_participacion = models.DecimalField(verbose_name='Porcentaje de Participación', decimal_places=2,
                                                  validators=[MaxValueValidator(100)], max_digits=10,
                                                  help_text='Debe ser menor o igual que 100')

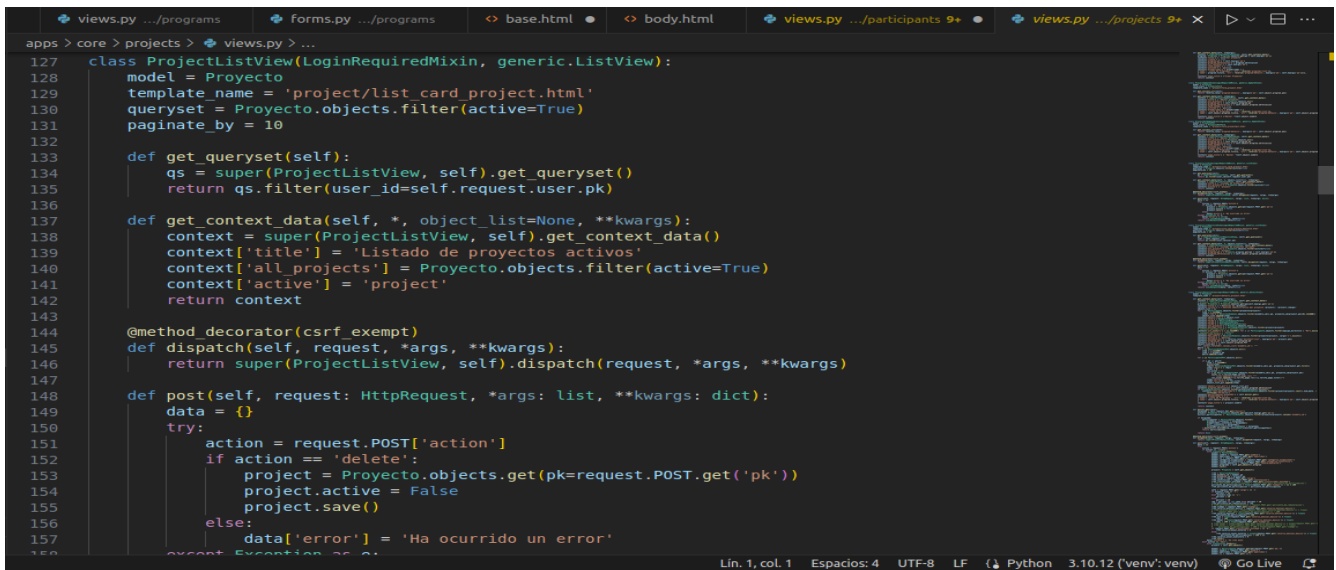
    salario_mensual = models.DecimalField(verbose_name='Salario mensual', decimal_places=2, max_digits=10, null=True,
                                         blank=True) # Salario basico mensual x porcentaje de participacion
    salario_anual_ejecutora = models.DecimalField(verbose_name='Salario Anual Ejecutora Principal', decimal_places=2,
                                                  max_digits=10, null=True, blank=True) # Calcular
    salario_anual_externo = models.DecimalField(verbose_name='Salario anual externo', decimal_places=2, max_digits=10,
                                                null=True, blank=True) # Calcular

    # Remuneración por Participación en el Proyecto
    porcentaje_remuneracion = models.DecimalField(verbose_name='Porcentaje de Remuneración', decimal_places=2,
                                                  max_digits=10, validators=[MaxValueValidator(100)],
                                                  help_text='Debe ser menor o igual que 100')

    mce = models.DecimalField(verbose_name='MCE', decimal_places=2, null=True, blank=True, max_digits=10) # calculados
    tiempo = models.PositiveSmallIntegerField(verbose_name='Tiempo en meses', validators=[MaxValueValidator(11)],
                                              help_text='Debe ser menor o igual que 11')
    anual = models.DecimalField(verbose_name='Anual', decimal_places=2, max_digits=10, null=True,
                               blank=True) # MCE * tiempo
    salario_mensual_basico = models.DecimalField(verbose_name='Salario básico mensual', decimal_places=2, max_digits=10)
```

Figura 6: Fragmento de código del patrón Alta cohesión. Fuente: Elaboración propia.

Bajo acoplamiento: se basa asignar responsabilidades de forma tal que cada clase se comunique con el menor número de clases, minimizando el nivel de dependencia. Esto se evidencia en la clase *ProjectListView* que se encarga de listar los servicios guardados en base de datos y mostrar el contenido de los proyectos sin depender de otra clase.



```

127 class ProjectListView(LoginRequiredMixin, generic.ListView):
128     model = Proyecto
129     template_name = 'project/list_card_project.html'
130     queryset = Proyecto.objects.filter(active=True)
131     paginate_by = 10
132
133     def get_queryset(self):
134         qs = super(ProjectListView, self).get_queryset()
135         return qs.filter(user_id=self.request.user.pk)
136
137     def get_context_data(self, *, object_list=None, **kwargs):
138         context = super(ProjectListView, self).get_context_data()
139         context['title'] = 'Listado de proyectos activos'
140         context['all_projects'] = Proyecto.objects.filter(active=True)
141         context['active'] = 'project'
142         return context
143
144     @method_decorator(csrf_exempt)
145     def dispatch(self, request, *args, **kwargs):
146         return super(ProjectListView, self).dispatch(request, *args, **kwargs)
147
148     def post(self, request: HttpRequest, *args: list, **kwargs: dict):
149         data = {}
150         try:
151             action = request.POST['action']
152             if action == 'delete':
153                 project = Proyecto.objects.get(pk=request.POST.get('pk'))
154                 project.active = False
155                 project.save()
156             else:
157                 data['error'] = 'Ha ocurrido un error'
158         except Exception as e:

```

Figura 7: Fragmento de código del patrón Bajo acoplamiento. Fuente: Elaboración propia.

2.3.3 Modelo de datos Para darle solución a la problemática planteada se obtuvo un modelo relacional de datos, que se desarrolló para el diseño y análisis lógico de los datos. A continuación, se representa en la figura 8 donde se aprecia la relación de datos:

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

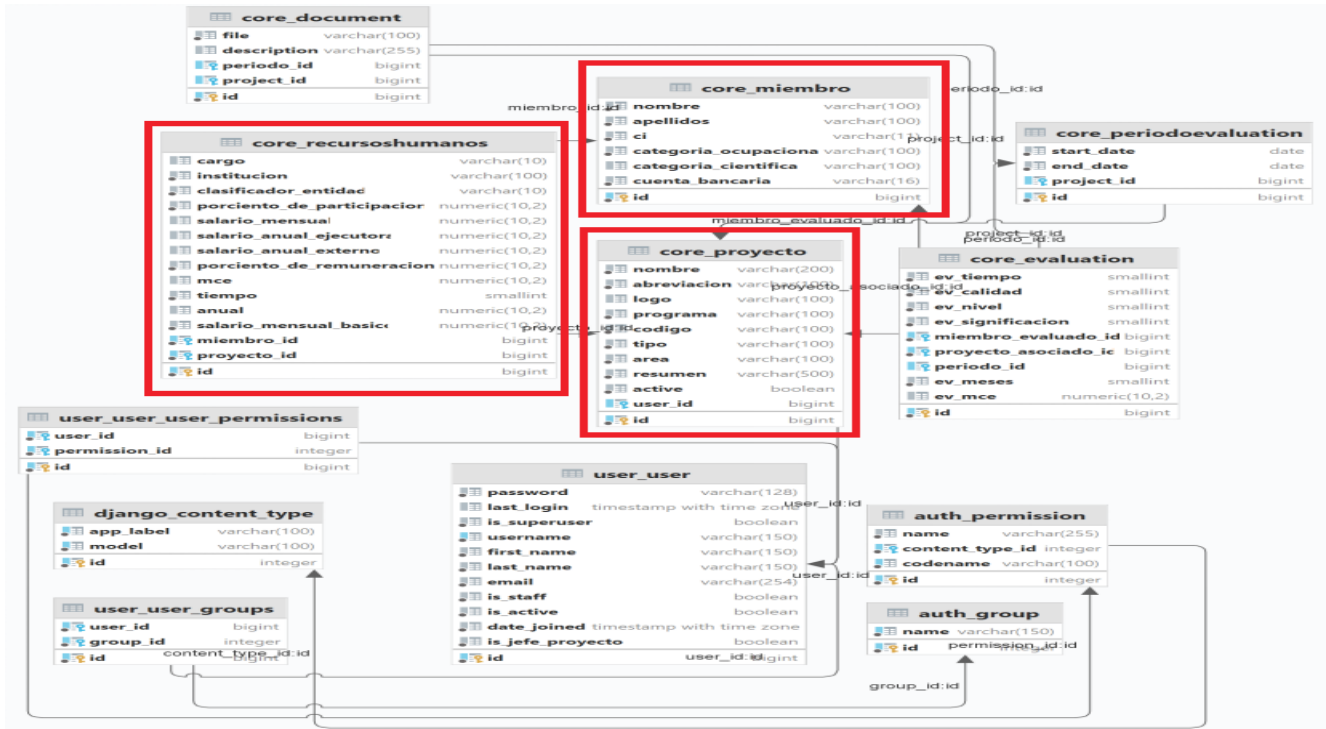


Figura 8: Modelo de datos relacional para el sitio web. Fuente: Elaboración propia.

En la figura 8 se muestra el modelo relacional donde se evidencia cada entidad usada en el sistema y sus respectivos atributos. Las tablas "core_recursos humanos" y "core_evaluacion" contienen una relación mucho a muchos añadiendo atributos extras entre core_miembro y core_proyecto. La tabla user_user está relacionada con las demás tablas de permisos y grupos generadas por el framework usado, entre ellas manejan toda la gestión de usuarios y permisos, incluyendo la autenticación.

Conclusiones del capítulo

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL MÓDULO DE GESTIÓN DE PROYECTOS

- El análisis y diseño de la solución permitió identificar los elementos importantes y necesarios para poder codificar
- El uso de patrones permitió usar código limpio, fácil de entender y escalable que permite la extensión del trabajo por otros trabajadores
- A partir de la solución se enriqueció el modelo de datos del sistema permitiendo crear y persistir más información alrededor de la gestión de los proyecto

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo muestra el diagrama de componentes, los estándares de codificación usados durante la implementación de la solución y se evalúa el nivel de calidad y fiabilidad de los resultados obtenidos en el desarrollo de la propuesta de solución. Dicha valoración se lleva a cabo a partir del establecimiento de una estrategia de pruebas, aplicando las disciplinas de pruebas internas y de aceptación que define la metodología que guía el desarrollo de la solución propuesta, con el fin de verificar y revelar la calidad del producto antes de su entrega al cliente.

3.1 Fase de implementación

Para implementar la solución planteada en esta fase se incluyen los estándares de codificación y el diagrama de componentes usados para una mayor comprensión del sistema desarrollado.

3.1.1 Estándar de codificación

Los estándares de codificación, son una práctica altamente recomendada para desarrollar *software* de alta calidad, estos estándares establecen criterios únicos que los programadores deben implementar cuando escriben código, para que el código fuente pueda ser entendido por cualquier miembro del equipo de desarrollo, y a su vez permite que el código pueda ser modificado por otro programador evitando que tenga que escribir la totalidad del código, lo que ocasionaría costos extras y mayor tiempo del requerido. A continuación, se presentan algunos estándares de codificación que fueron utilizados durante la implementación del sistema.

Grosor de línea:

Cada línea de código no debe exceder los 80 caracteres en la medida de lo posible (en circunstancias especiales, puede exceder ligeramente los 80, pero la más larga no puede exceder los 120).

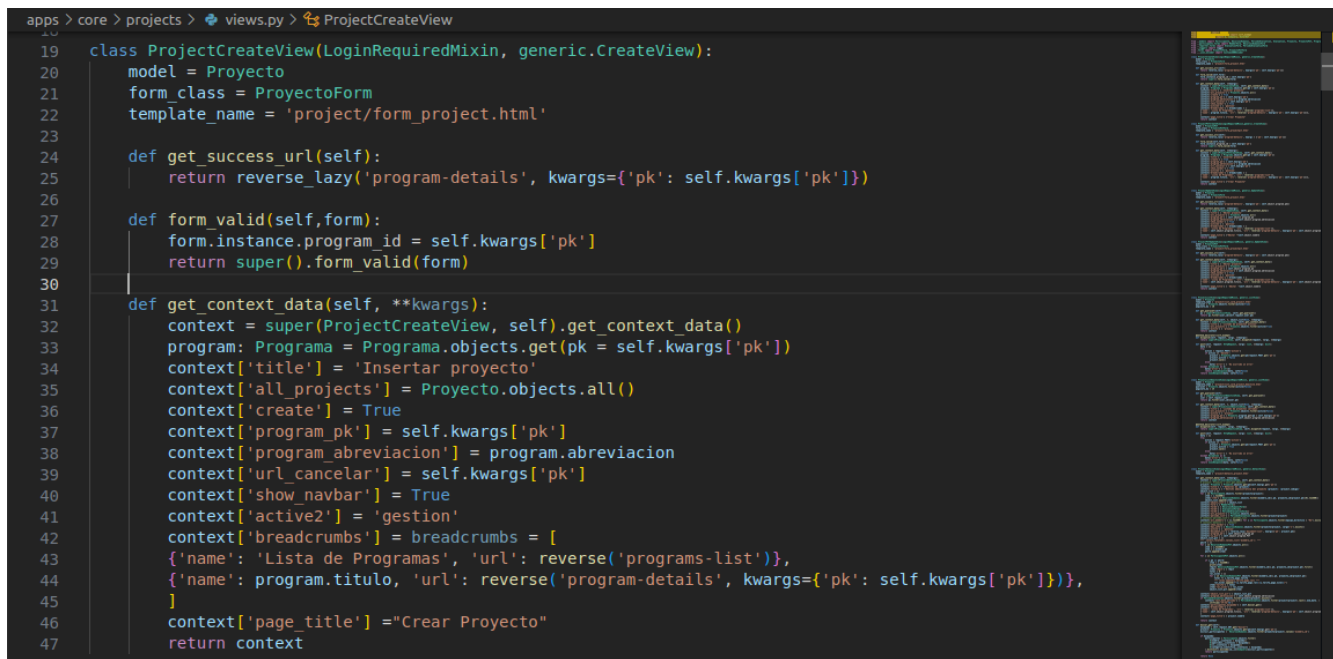
Razones:

- Esto es útil al ver una diferencia de lado a lado.
- Conveniente para ver el código debajo de la consola.
- Demasiado tiempo puede ser un diseño defectuoso.

Líneas en blanco:

- Dos líneas en blanco entre las funciones de nivel de módulo y las definiciones de clase.
- Línea en blanco entre las funciones de los miembros de la clase.

Las líneas en blanco se pueden utilizar en funciones para separar códigos relacionados lógicamente.



```

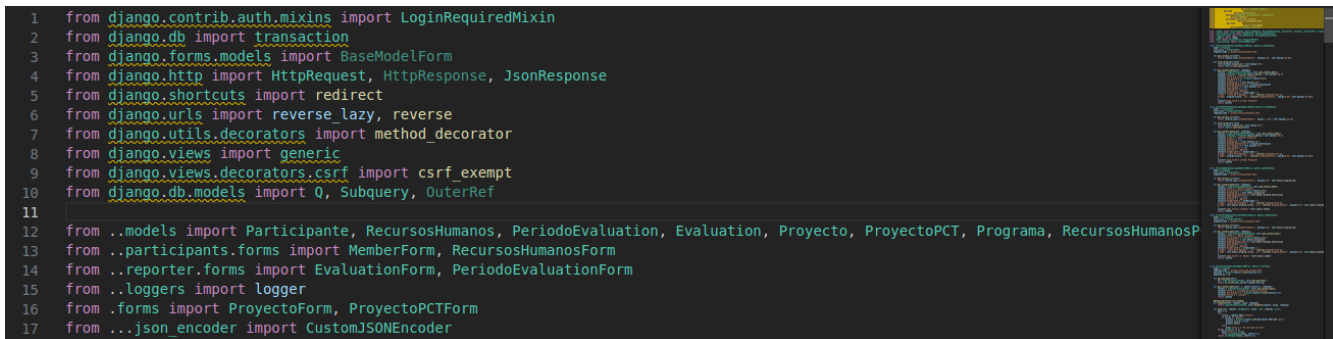
10
11
12
13
14
15
16
17
18
19 class ProjectCreateView(LoginRequiredMixin, generic.CreateView):
20     model = Proyecto
21     form_class = ProyectoForm
22     template_name = 'project/form_project.html'
23
24     def get_success_url(self):
25         return reverse_lazy('program-details', kwargs={'pk': self.kwargs['pk']})
26
27     def form_valid(self, form):
28         form.instance.program_id = self.kwargs['pk']
29         return super().form_valid(form)
30
31     def get_context_data(self, **kwargs):
32         context = super(ProjectCreateView, self).get_context_data()
33         program: Programa = Programa.objects.get(pk = self.kwargs['pk'])
34         context['title'] = 'Insertar proyecto'
35         context['all_projects'] = Proyecto.objects.all()
36         context['create'] = True
37         context['program_pk'] = self.kwargs['pk']
38         context['program_abreviacion'] = program.abreviacion
39         context['url_cancelar'] = self.kwargs['pk']
40         context['show_navbar'] = True
41         context['active2'] = 'gestion'
42         context['breadcrumbs'] = breadcrumbs = [
43             {'name': 'Lista de Programas', 'url': reverse('programs-list')},
44             {'name': program.titulo, 'url': reverse('program-details', kwargs={'pk': self.kwargs['pk']})},
45         ]
46         context['page_title'] = "Crear Proyecto"
47         return context
48

```

Figura 9: Ejemplo de código de los estándares de codificación. Fuente: Elaboración propia.

Declaración de importancia:

- Las declaraciones de importación deben organizarse en orden y deben ser separadas por una línea en blanco entre cada grupo.
- Deben colocarse al principio del archivo, después de la descripción del módulo y la cadena de documentos, y antes de las variables globales.



```

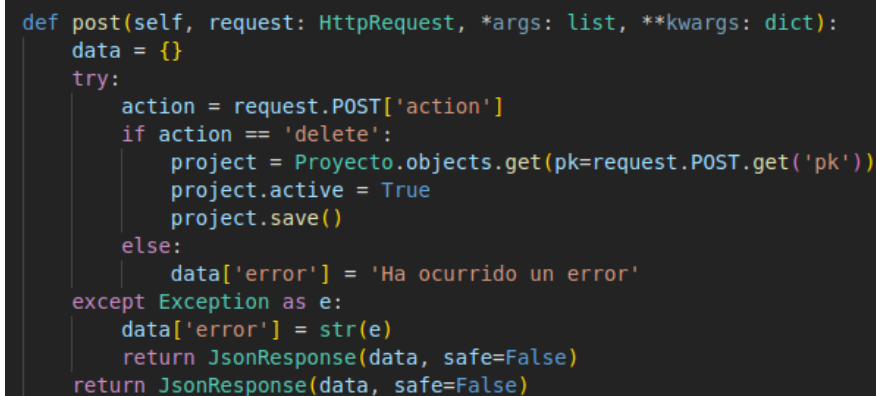
1 from django.contrib.auth.mixins import LoginRequiredMixin
2 from django.db import transaction
3 from django.forms.models import BaseModelForm
4 from django.http import HttpRequest, HttpResponse, JsonResponse
5 from django.shortcuts import redirect
6 from django.urls import reverse_lazy, reverse
7 from django.utils.decorators import method_decorator
8 from django.views import generic
9 from django.views.decorators.csrf import csrf_exempt
10 from django.db.models import Q, Subquery, OuterRef
11
12 from ..models import Participante, RecursosHumanos, PeriodoEvaluation, Evaluation, Proyecto, ProyectoPCT, Programa, RecursosHumanosP
13 from ..participants.forms import MemberForm, RecursosHumanosForm
14 from ..reporter.forms import EvaluationForm, PeriodoEvaluationForm
15 from ..loggers import logger
16 from .forms import ProyectoForm, ProyectoPCTForm
17 from ...json_encoder import CustomJSONEncoder

```

Figura 10: Ejemplo de código de los estándares de codificación. Fuente: Elaboración propia.

Espacio:

- Un espacio a cada lado del operador binario [=, -, +=, ==, >, in, is not, and].
- En la lista de parámetros de la función, debe haber un espacio después.
- No se puede agregar espacios adicionales después del paréntesis izquierdo y antes del paréntesis derecho.
- No puede haber espacios adicionales antes del paréntesis de apertura del objeto de diccionario.
- No se usa espacios adicionales para alinear las declaraciones de asignación.



```

def post(self, request: HttpRequest, *args: list, **kwargs: dict):
    data = {}
    try:
        action = request.POST['action']
        if action == 'delete':
            project = Proyecto.objects.get(pk=request.POST.get('pk'))
            project.active = True
            project.save()
        else:
            data['error'] = 'Ha ocurrido un error'
    except Exception as e:
        data['error'] = str(e)
    return JsonResponse(data, safe=False)

```

Figura 11: Ejemplo de código de los estándares de codificación. Fuente: Elaboración propia

3.1.2 Diagrama de componentes:

En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema Como se describió en el epígrafe 2.1 de la presente investigación, en esta disciplina los autores deciden la elaboración del artefacto que describe la implementación del diseño. En este artefacto se especifican todas las características de implementación de cada tipo de componente de producto. Para un mayor entendimiento a continuación se describe el diagrama de componentes relacionado con la implementación de la HU-1: Crear proyecto

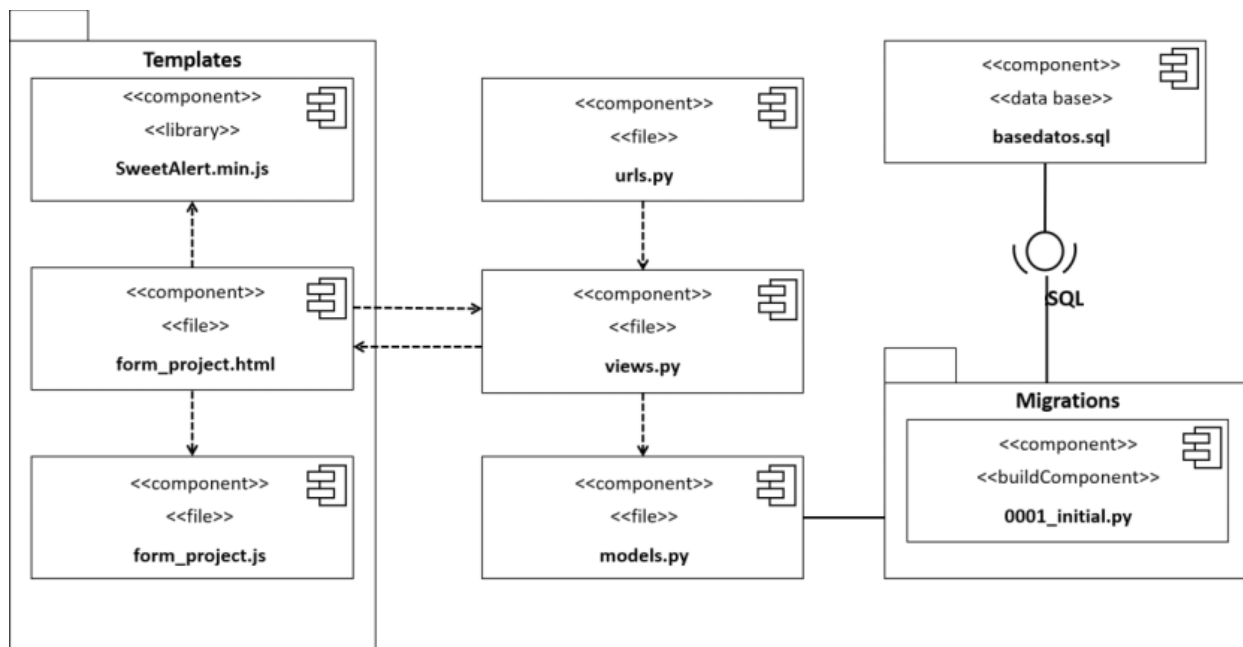


Figura 12: Diagrama de componentes correspondiente a la HU_1 Crear proyecto. Fuente: Elaboración propia.

A continuación, se detalla la descripción de cada componente representado:

1. Template **"form_proyect.html"**: Es una plantilla HTML que se utiliza para representar la interfaz de usuario de un formulario relacionado con crear proyecto. Este componente puede contener elementos interactivos y campos de entrada de información.
2. Archivos **"sweetAlert.min.js"** y **"form_proyect.js"**: Estos archivos contienen código JavaScript que proporciona funcionalidades adicionales a la plantilla "form_proyect.html". "sweetAlert.min.js" es una biblioteca externa utilizada para mostrar alertas o

notificaciones personalizadas, mientras que **"form_proyect.js"** puede contener lógica específica para el comportamiento del formulario.

3. Archivo **"views.py"**: Es un archivo de Python que contiene las funciones y lógica necesarias para procesar las solicitudes relacionadas con la plantilla "form_proyect.html". Este componente define cómo se manejan y procesan los datos enviados desde el formulario.
4. Archivo **"urls.py"**: Es un archivo de configuración que mapea las URL de la aplicación a las funciones correspondientes en "views.py". Este componente define las rutas y las asociaciones entre las URL y las funciones que se encargan de manejar las solicitudes.
5. Archivo **"models.py"**: Es un archivo de Python que define las clases de modelos de datos utilizadas en la aplicación. Estas clases representan la estructura y las relaciones de los datos almacenados en la base de datos.
6. Migrations representan los cambios en el esquema de la base de datos. El archivo **"0001_initial.py"** es una migración inicial que define la estructura inicial de la base de datos y puede contener instrucciones para crear tablas y definir relaciones entre ellas.
7. Base de datos SQL "basedatos.sql": Es una base de datos SQL que almacena los datos relacionados con los proyectos. El archivo "basedatos.sql" puede ser el script de creación y carga inicial de la base de datos.

El diagrama de componentes muestra cómo se relacionan y interactúan los diferentes componentes de una historia de usuario que implica una plantilla HTML, archivos JavaScript, archivos de Python, migraciones de base de datos y una base de datos SQL. Estos componentes trabajan juntos para proporcionar una funcionalidad específica relacionada con la gestión de proyectos.

3.2 Estrategia de pruebas

Una estrategia de prueba del software integra los métodos de diseño de caso de pruebas del software en una serie bien planeada de pasos que desembocará en la eficaz construcción del mismo. La estrategia proporciona un mapa que describe los pasos que se darán como parte de la prueba, indica cuándo se planean, cuándo se dan estos pasos, además de cuanto esfuerzo, tiempo y recursos consumirán. Por tanto, cualquier estrategia de prueba debe

incorporar la planeación de pruebas, el diseño de casos de pruebas, la ejecución de pruebas, la recolección y evaluación de los datos resultantes. (Pressman, 2010) La estrategia de prueba debe indicar los niveles de pruebas (ciclos) que se aplican y la intensidad o profundidad a aplicar para cada nivel de prueba definido.

3.2.1 Pruebas de Software

Las pruebas de software son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información sobre la calidad del producto. Dependiendo del tipo de pruebas, estas actividades pueden ser implementadas en cualquier momento del proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo

3.2.2 Tipos de prueba

Para comprobar el correcto funcionamiento de la solución obtenida se realizan los siguientes tipos de pruebas:

- **Pruebas funcionales:** se basan en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas.
- **Pruebas estructurales:** se basan en medir la totalidad de las pruebas mediante la evaluación de tipo estructura.

En la presente investigación se utilizaron las técnicas de pruebas dinámicas, las cuales permitieron el uso de los métodos de caja negra para las pruebas funcionales y caja blanca para las pruebas estructurales.

3.2.2.1 Método de caja blanca

Técnica de camino básico El método de caja blanca se centra en los detalles procedimentales del software, por lo que su diseño está ligado al código fuente (Sánchez Peño, 2015). Garantiza que el ingeniero del software pueda obtener casos de pruebas que se ejerciten por lo menos una vez todos los caminos independientes de cada método y se

ejecuten todos los bucles en sus límites operacionales. Permite que disminuya el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

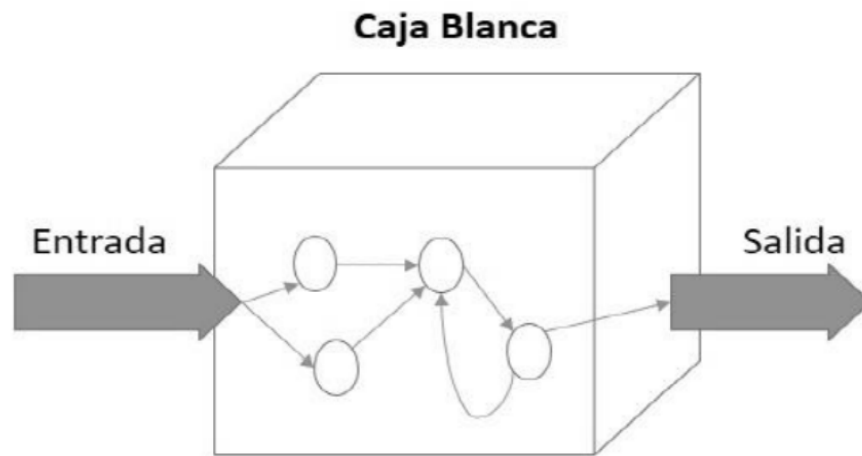


Figura 13: Método caja blanca. Fuente: (Sánchez Peño, 2015).

La técnica empleada en este tipo de prueba fue el camino básico, el cual consiste en diseñar un caso de prueba por cada camino independiente del programa. Con esta técnica se garantiza que se prueben todos los caminos de ejecución del programa, al menos una vez (Sánchez Peño, 2015).

Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que:

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.

Revisen estructuras de datos internas para garantizar su validez.

3.2.2.2 Método de caja negra

El método de caja negra examina el programa para que cuente con todas las funcionalidades analizando los resultados que devuelve y probando todas las entradas en sus valores válidos e

inválidos. Con este método se intenta encontrar los errores: de inicialización y terminación, de interfaz y en las estructuras (Sánchez Peño, 2015).

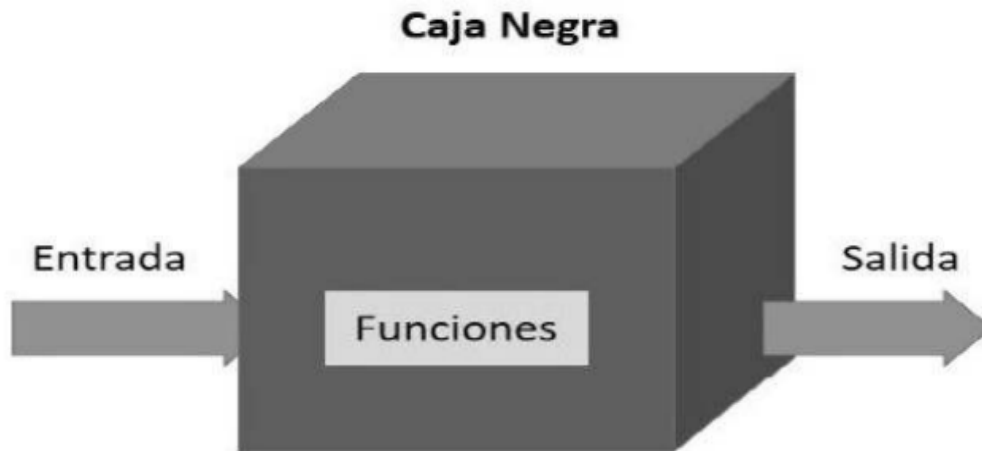


Figura 14: Método caja negra. Fuente: (Sánchez Peño, 2015).

Existen varias técnicas que se pueden aplicar al método de caja negra, pero en la presente investigación se empleará partición equivalente, el cual consiste en que los valores de entrada del sistema se dividen en grupos que vayan a tener un comportamiento similar para ser procesados de la misma forma.

3.2.2.3 Prueba de Integración

Las pruebas de integración verifican la interacción y compatibilidad entre los componentes del software. Se realizan para detectar errores que puedan surgir al combinar las partes individuales del sistema. Estas pruebas ayudan a garantizar un funcionamiento correcto y sin problemas del software en su conjunto (Pressman, 2010).

3.2.2.4 Prueba de Sistema

Las pruebas de sistema evalúan el sistema completo para garantizar su funcionamiento correcto y cumplimiento de requisitos. Se realizan después de las pruebas de integración y antes de las pruebas de aceptación. Estas pruebas ayudan a identificar posibles fallas, asegurando la calidad y confiabilidad del sistema (Sommerville, 2011).

3.2.2.5 Prueba de Aceptación

Las pruebas de aceptación se realizan para confirmar que el sistema cumple con los requisitos y expectativas del cliente. Son llevadas a cabo en un entorno similar al de producción y son ejecutadas por los usuarios finales o representantes del cliente. Estas pruebas aseguran que el sistema esté listo para su implementación y uso (Sommerville, 2011).

3.2.3 Niveles de prueba

Los niveles de prueba con un conjunto de pruebas que se le aplican al software en diferentes etapas del proceso de desarrollo y que son agrupadas en niveles, en donde el probador debe asegurarse de que los resultados cumplen con la verificación y validación del software. Los niveles de pruebas utilizados en la propuesta de solución, se describen a continuación:

3.2.3.1 Pruebas unitarias. Método de caja blanca

Técnica de camino básico:

Las pruebas unitarias consisten en la verificación del correcto funcionamiento de una unidad de código. Suelen ser realizadas por los desarrolladores, ya que es importante conocer el código fuente del programa y mejoran la calidad del software evitando errores de programación

En este tipo de prueba se aplicó el método de caja blanca al método post () de la clase ProjectListTest () utilizando la técnica de camino básico. A continuación, se muestran los métodos anteriormente mencionados:

```

apps > core > projects > views.py > ...
147
148 def post(self, request: HttpRequest, *args: list, **kwargs: dict):
149     1 data = {}
150     2 try:
151         3 action = request.POST['action']
152         4 if action == 'delete':
153             5 project = Proyecto.objects.get(pk=request.POST.get('pk'))
154             project.active = False
155             project.save()
156         else:
157             6 data['error'] = 'Ha ocurrido un error'
158     except Exception as e:
159         7 data['error'] = str(e)
160     return JsonResponse(data, safe=False)
161     8 return JsonResponse(data, safe=False)
162

```

Figura 15: Código del método a realizar la técnica de camino básico. Fuente: Elaboración propia.

Para obtener la cantidad de casos de prueba a partir de esta técnica se debe construir el grafo correspondiente al código de la función de la siguiente manera:

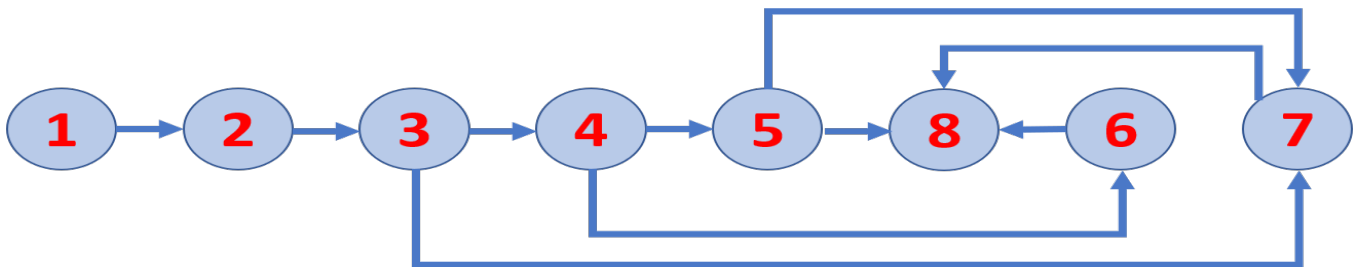


Figura 16: Grafo de flujo del método Post. Fuente: Elaboración propia.

A partir del grafo resultante se calcula la complejidad ciclomática $V(G)$, la cual se puede calcular de tres formas:

Cálculo de la Complejidad Ciclomática:

1. $V(G) = A - N + 2$, siendo A la cantidad de aristas o arcos del grafo y N la cantidad de nodos del grafo.
2. $V(G) = P + 1$, siendo P los nodos predicados, es decir los que tienen más de una arista de salida.
3. $V(G) = R$, siendo R el número de regiones cerradas del grafo.

Al realizar el cálculo por cualquiera de las fórmulas anteriores se obtienen los siguientes resultados:

$$1. V(G) = A - N + 2$$

$$V(G) = 10 - 8 + 2$$

$$V(G) = 4$$

$$2. V(G) = P + 1$$

$$V(G) = 3 + 1$$

$$V(G) = 4$$

$$3. V(G) = R$$

$$V(G) = 4$$

Al aplicar el cálculo se obtiene como solución $V(G) = 4$, definiendo como posibles caminos básicos:

Camino básico 1: 1, 2, 3, 4, 5, 8

Camino básico 2: 1, 2, 3, 4, 6, 8

Camino básico 3: 1, 2, 3, 7, 8

Camino básico 4: 1, 2, 3, 4, 5, 7, 8

Luego de tener elaborados el Grafo de flujo y los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos, cada camino básico es un caso de prueba a realizar. En este caso se obtuvieron cuatro caminos básicos, que dan lugar a la confección de igual número de casos de pruebas, para aplicar las pruebas a este método. A continuación, se muestra el caso de prueba para cada uno de los caminos:

Tabla 8: Caso de prueba para el camino básico 1. Fuente: Elaboración propia

Descripción	Verificar que exista el proyecto buscado
Condición de ejecución	Que exista una petición por el método POST con la variable <i>action</i> y que sea igual <i>delete</i> .
Entrada	Variable <i>action</i> y variable <i>pk</i>
Resultado	Envía una respuesta sin contenido en formato JSON

Tabla 9: Caso de prueba para el camino básico 2. Fuente: Elaboración propia

Descripción	Verificar que exista el proyecto buscado
Condición de ejecución	Que exista una petición por el método POST con la variable <i>action</i> y que sea igual a <i>delete</i> .
Entrada	Variable <i>action</i> y variable <i>pk</i>
Resultado	Envía una respuesta sin contenido en formato JSON

Tabla 10: Caso de prueba para el camino básico 3. Fuente: Elaboración propia

Descripción	Verificar que exista el proyecto buscado
Condición de ejecución	Que exista una petición por el método POST con la variable <i>action</i> y que sea distinto de <i>delete</i> .
Entrada	Variable <i>action</i> y variable <i>pk</i>
Resultado	Envía una respuesta en formato JSON con los errores encontrados durante la iteración.

Tabla 11: Caso de prueba para el camino básico 4. Fuente: Elaboración propia

Descripción	Verificar que exista el proyecto buscado
Condición de ejecución	Que exista una petición por el método POST con la variable <i>action</i> y que sea igual a <i>delete</i> .
Entrada	Variable <i>action</i> y variable <i>pk</i>
Resultado	Envía una respuesta en formato JSON con los errores encontrados durante la iteración.

3.2.3.2 Técnicas automatizadas:

A medida que crecen los sitios web se vuelven más difíciles de probar a mano, no sólo hay más para probar, sino que, además, a medida que las interacciones entre los componentes se vuelven más complejas, un pequeño cambio en un área puede suponer muchas pruebas adicionales para verificar su impacto en otras áreas. Una forma de mitigar estos problemas es escribir pruebas automatizadas, que pueden ser ejecutados de manera fácil y fiable cada vez que hagas un cambio[CITATION MDN221 \l 21514].

Django proporciona un marco de prueba con una pequeña jerarquía de clases que se basan en la librería *unittest* estándar Python. A pesar del nombre, este marco de prueba es adecuado tanto para pruebas unitarias como de integración. El marco de Django agrega métodos y herramientas API para ayudar a probar el comportamiento web y específico de Django. Estos le permiten simular solicitudes, insertar datos de prueba e inspeccionar la salida de su aplicación[CITATION MDN221 \l 21514]

Se realizaron un total de tres pruebas Integración:

Creación de 10 proyectos:

```
$ python -m unittest
.....
-----
---
Ran 10 tests in 5.237s

OK
```

Figura 17:Prueba de Creación de 10 proyectos

Creación Tabla tarifas:

```

$ python -m unittest
.
-----
---
Ran 1 test in 0.123s

OK

```

Figura 18: Creación de Tabla tarifas

Creación de 5 miembros(RRHH):

```

$ python -m unittest
.....
-----
---
Ran 5 tests in 0.678s

OK

```

Figura 19: Creación de miembros

La figura 17, 18 y 19 muestra que las 3 pruebas automatizadas fueron aceptadas y sus respectivos tiempos de ejecución.

3.2.3.3 Pruebas de sistema. Método de caja negra

Las pruebas de sistema están constituidas por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas[CITATION Som07 \l 3082]. Entre las pruebas de sistema que se ejecutan están las pruebas de rendimiento y resistencia.

Pruebas de rendimiento: están diseñadas para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado. Lo que posibilita determinar cuán rápida es la respuesta del sistema ante un conjunto de peticiones concurrentes[CITATION Tol14 \l 3082].

Pruebas de resistencia: están diseñadas para enfrentar a los programas con situaciones anormales. La prueba de resistencia ejecuta un sistema de forma que demande recursos en cantidad o volúmenes anormales[CITATION Tol14 \l 3082].

Resultados de las pruebas de rendimiento y resistencia:

Una prueba de rendimiento y resistencia se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. El sistema se prueba con un número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la prueba. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Para llevar a cabo las pruebas de rendimiento y resistencia se utilizó la herramienta JMeter 2.12. La prueba consistió en realizar a una funcionalidad tres pruebas de 50, 100 y 200 hilos, los cuales simulan 50, 100 y 200 creación de proyectos respectivamente. Se simuló la creación aleatoria y a partir de ahí, se recolectaron los datos necesarios para su interpretación.

Para un mejor entendimiento de los datos que se verán a continuación, se explica cada parámetro que compone la tabla.

#Muestras: cantidad de hilos utilizados para la URL.

Media: tiempo promedio en milisegundos para un conjunto de resultados.

Min: tiempo mínimo que demora un hilo en acceder a una página.

Max: tiempo máximo que demora un hilo en acceder a una página.

Pet/seg: hace referencia al número de peticiones que el servidor puede procesar en un segundo.

Kb/seg: rendimiento medido en Kilobytes por segundo.

A continuación, se muestran en la Tabla 12 los resultados obtenidos con la herramienta:

Tabla 12: Resultados de las pruebas de rendimiento y resistencia. Fuente: Elaboración propia.

Aplicado a:	Cantidad de hilos	Tiempo de ejecución (ms)			Rendimiento		
		Min.	Max.	Media	% Error	Pet/seg	Kb/seg
Crear Proyecto	50	232	1683	963	0	18.6	133.2
	100	471	1724	1116	0	25.8	185.2
	200	624	1832	1358	0	36.2	231.3

3.2.3.4 Pruebas de aceptación. Método de caja negra

Técnica de partición equivalente:

Las pruebas de aceptación se caracterizan por la participación activa del usuario, que debe ejecutar los casos de prueba ayudados por miembros del equipo de pruebas. Están enfocadas a verificar que los requisitos desarrollados cumplan con las expectativas del usuario y deben realizarse antes de la salida del producto.

Se le realizó a la presente propuesta de solución el método de caja negra mediante la técnica de partición equivalente. Se tomó como base todos los requisitos funcionales

Tabla 13: Resultados del método de caja negra mediante la técnica partición equivalente en las pruebas de aceptación. Fuente: Elaboración propia.

No. Iteración	NC	Funcionalidad	Interfaz	Correspondencia	Resueltas
1	11	3	6	2	11
2	4	1	2	1	4
3	0	0	0	0	0
Total:	15	4	8	3	15

Las causas de las no conformidades fueron:

- Errores de interfaz: paneles que no cumplían con las pautas de diseño establecidas.
- Errores de correspondencia: las descripciones de varios escenarios no están en correspondencia con su nombre.

- Errores de funcionalidad: algunos formularios no enviaban los datos correctamente y algunos botones no funcionaban.

Para la evaluación a través de las técnicas de caja negra, utilizando como herramienta los DCP, se efectuaron un total de tres iteraciones para poder alcanzar resultados satisfactorios, atendiendo al correcto comportamiento de la solución ante diferentes situaciones. En la siguiente figura se realiza un resumen con el número total de No conformidades (NC) por iteración.

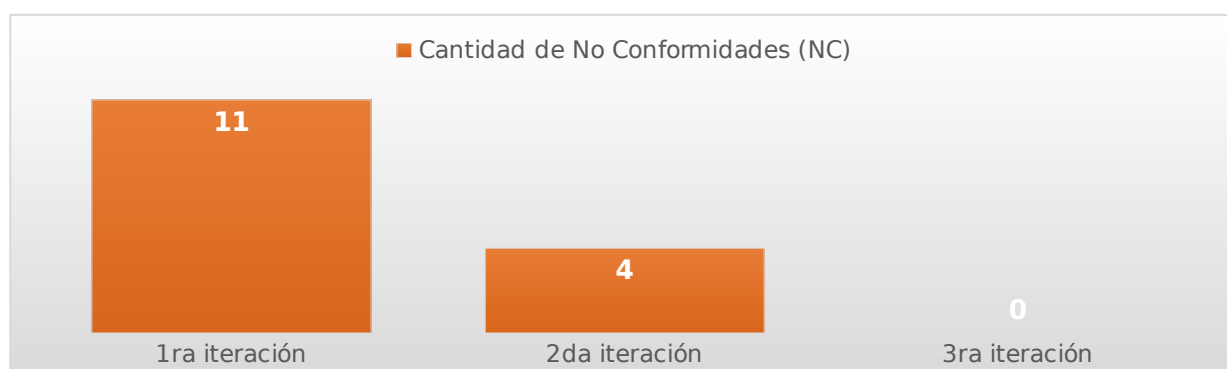


Figura 20: Total, de las NC detectadas durante las pruebas de caja negra. Fuente: Elaboración propia.

Estas NC se clasificaron en: Funcionalidad, Interfaz y Correspondencia. A continuación, se muestra un resumen de las NC por tipo de clasificación en cada iteración de prueba:

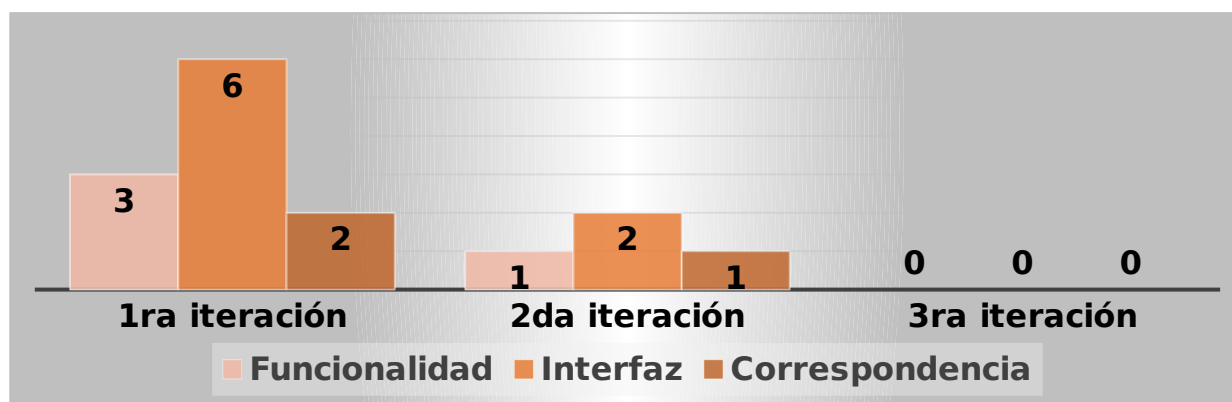


Figura 21: Clasificación de las NC detectadas corregidas. Fuente: Elaboración propia

Conclusiones del capítulo

- Los estándares de codificación permitieron un código limpio, y el framework Django provee medidas de seguridad
- El Software funcionalmente es correcto mediante las pruebas implementadas, caja negra, caja blanca, prueba de integración.

Conclusiones Generales

Al concluir la investigación para el desarrollo del sistema de gestión para la administración de información de proyectos de investigación, desarrollo e innovación para el Parque Científico Tecnológico, se pudo arribar a las siguientes conclusiones:

- El análisis de los sistemas homólogos y del objeto de estudio permitió constatar las novedades de la investigación al no identificarse soluciones que propicien la gestión administrativa específica de los proyectos del PCT.
- A partir del análisis realizado, las entrevistas y aplicando la metodología AUP-UCI escenario 4 fue posible elaborar los artefactos ingenieriles que permitieron la comprensión general de la propuesta de solución.
- Teniendo en cuenta los artefactos generados, el ambiente de desarrollo y el uso de la metodología se obtuvo un módulo que permite la gestión administrativa de los proyectos y los recursos humanos incubados en el PCT.
- La propuesta de solución fue integrada al sistema GAPID y permite que se ejecuten los flujos de sistema; complementando y siendo más robusto la solución.
- Las pruebas de software definidas propiciaron asegurar que el sistema cuenta con la calidad requerida y funciona correctamente.

En resumen, al finalizar la investigación, se logró desarrollar un sistema de gestión que satisface las necesidades específicas del Parque Científico Tecnológico. El marco teórico, las herramientas y metodologías utilizadas, así como las pruebas realizadas, desempeñaron un papel crucial en el éxito de la implementación del sistema.

Recomendaciones

Una vez concluida la presente investigación se recomienda lo siguiente:

- Implementar componentes de nomencladores para facilitar información de selección dentro del módulo.
- Implementar un componente de certificaciones que permita generar información necesaria para certificar un proyecto.

Referencias bibliográficas

- Python Software Foundation . (2021). *Python*. Obtenido de <https://www.python.org/>
- Aachen University. (2021). Recuperado el 25 de junio de 2021, de <https://www.rwth-aachen.de/>
- Akadememos, G. d. (23 de mayo de 2019). *Serie Científica*. Obtenido de Serie Científica: <https://publicaciones.uci.cu/index.php/serie/article/view/253>
- Altamira. (2021). *Altamira Employees*. Obtenido de Altamira Employees: <https://www.altamirahrm.com/es/software/software-rrhh>
- Amazon Web Services. (2021). Recuperado el 25 de junio de 2021, de <https://aws.amazon.com/es/>
- Angular. (20 de 09 de 2021). Obtenido de <https://angular.io/>
- Bos, B. (2019). *w3schools*. Obtenido de W3C: <https://www.w3.org/Style/CSS/Overview.en.html>
- Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., & Yergeau, F. (2008). *Extensible Markup Language (xml)*. Obtenido de <http://www.w3.org/TR/2008/REC-xml-20081126/>
- Canarias, G. d. (2018). *INCENTIVOS FISCALES A LA I+D+i*. Obtenido de Gobierno de Canarias: <https://www3.gobiernodecanarias.org/aciisi/obidic/incentivos-fiscales/que-se-considera-idi/#:~:text=I%2BD%2Bi%20son%20las,2014%20del%20Impuesto%20sobre%20Sociedades.>
- Cohn, M. (2006). *Agile Estimating and Plannning*. Pearson Education.
- COLCIENCIAS. (13 de 05 de 2022). *Departamento administrativo de ciencia, tecnología e innovación*. . Obtenido de <https://legadoweb.minciencias.gov.co/faq/qu-es-un-proyecto-de-innovaci-n-tecnol-gica>
- Crispin, L. (2002). *Testing Extreme Programming*. Addison Wesley Professional.

- DE, D. (2022). *PROYECTO DE INVESTIGACIÓN*. Obtenido de DEFINICIÓN DE: <https://definicion.de/proyecto-de-investigacion/#:~:text=Un%20proyecto%20de%20investigaci%C3%B3n%20es,del%20fen%C3%B3meno%20que%20se%20investigar%C3%A1.>
- Díaz, A. M. (2018). <http://scielo.sld.cu>. Obtenido de <http://scielo.sld.cu>: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992018000200006&lng=es&nrm=iso
- Django Software Foundation . (2021). *Django*. Obtenido de <https://www.djangoproject.com/>
- EAE Business School. (29 de 09 de 2020). *Tipos de proyectos. Entiende qué tipologías de proyectos existen, sus características y sus fases*. Obtenido de <https://recursos-project-management.obsbusiness.school/descargate-nuestra-guia-gratuita-sobre-tipos-proyectos>
- EcuRed*. (s.f.). Obtenido de https://www.ecured.cu/Versat_Sarasola
- Ecured. (2022). *Pruebas de software*. Obtenido de Ecured: https://www.ecured.cu/Pruebas_de_software
- Erick, G. G. (2020). Obtenido de <https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/6783/Gonzales%20Gonzales%20Christian%20Erick.pdf?sequence=1&isAllowed=y>
- Evaluando, G. (2022). *Evaluando ERP*. Obtenido de Evaluando ERP: <https://www.evaluandoerp.com/software-erp/sistema-de-gestion/>
- Evaluando, G. (2022). *Evaluando ERP*. Obtenido de Evaluando ERP: <https://www.evaluandoerp.com/software-erp/sistema-de-gestion/>
- García Regüela, A. (27 de 06 de 2017). *La norma ISO 21500 sobre gestión de proyectos*. Obtenido de International Dynamic Advisors: <http://www.intedya.com/internacional/1493/noticia-la-norma-iso-21500-sobre-gestion-de-proyectos.html>

- Garzón, R., & De la Portilla, O. (2011). *METODOLOGÍA PARA EVALUAR LA CALIDAD DE PLANEACIÓN DE PROYECTOS INFORMÁTICOS*. Santiago de Cali: UNIVERSIDAD ICESI. Facultad de Ingeniería.
- Gómez Torres, J., Cruz Díaz, R. O., Páez Moro, M., & González Rodríguez, Y. (2020). *Indicaciones metodológicas para la actividad de programas y proyectos de CTI*. La Habana: CITMA.
- Hunt, J. (s.f.). *Scala Design Patterns: Patterns For Practical Reuse And Design*. Obtenido de VDOC.PUB: <https://vdoc.pub/documents/scala-design-patterns-patterns-for-practical-reuse-and-design-78703apnuci0>
- ICTEA. (2022). Obtenido de ICTEA: <https://www.ictea.com/cs/index.php?rp=/knowledgebase/8649/iQue-es-el-lenguaje-de-programacion-PYTHON.html#:~:text=Python%20es%20un%20lenguaje%20de,en%20menor%20medida%2C%20programaci%C3%B3n%20funcional>.
- Labrada Guió, L. M. (2009). *Procedimiento para evaluar proyectos informáticos y establecer un orden de prioridades*. La Habana: Universidad de Ciencias Informáticas.
- Lam Díaz, R. M. (2005). *Metodología para la confección de un proyecto de investigación*. La Habana: Instituto de Hematología e Inmunología.
- Larman, C. (2000). *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México: Prentice Hall.
- Letelier, P., & M.C, P. (2003). *Métodologías Ágiles en el Desarrollo de Software: Extreme Programming(XP)*. Universidad Politécnica de Valencia.
- Maastricht University. (2021). Recuperado el 25 de junio de 2021, de <https://www.maastrichtuniversity.nl/>
- McGill University. (2021). Recuperado el 25 de junio de 2021, de <https://www.mcgill.ca/>
- MDN. (2022). *Introducción a Django*. Obtenido de Resources for Developers: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>

- MDN. (2022). *MDN Web docs*. Obtenido de Probando una aplicación web Django: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Testing#:~:text=Para%20escribir%20una%20prueba%2C%20se%20deriva%20de%20cualquiera,m%20C3%A9todos%20de%20prueba%20elegidos%20en%20sus%20clases%20derivadas>.
- MDN web docs. (27 de 4 de 2020). *developer.mozilla.org*. Obtenido de developer.mozilla.org: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- <http://uniciencista.gfrodriguez.online/2018/02/herramientas-case-principales-usos.html>
- Morles, V. (2011). Guía para la elaboración y evaluación de proyectos de investigación. *Revista de pedagogía*, 131-146.
- OBS Business School. (29 de 09 de 2020a). *Tipos de proyectos y sus principales características*. Obtenido de Universitat de Barcelona: <https://obsbusiness.school/es/blog-project-management/administracion-de-proyectos/tipos-de-proyectos-y-sus-principales-caracteristicas>
- OBS Business School. (2020b). *Qué hay que saber de la ISO 21500?* Obtenido de Universitat de Barcelona: <https://obsbusiness.school/es/blog-project-management/temas-actuales-de-project-management/que-hay-que-saber-de-la-iso-21500>
- Page, D., Saito, H., & Yeatman, M. (28 de Noviembre de 2018). *PgAdmin*. Recuperado el Marzo de 2019, de PgAdmin: <http://www.pgadmin.org/>
- Parodi, C. (2000). *El lenguaje de los proyectos*. Gerencia social. Diseño, monitoreo y evaluación de proyectos sociales. Lima-Perú: Universidad del Pacífico. ISBN 9972-603-32-6.
- Pérez Porto, J., & Gardey, A. (2013). *Definición De*. Obtenido de <https://definicion.de/proyecto-de-investigacion/>
- PMI. (2013). *Guía de los Fundamentos de la Dirección de Proyectos. (PMBOK 5ª. Ed.)*. PMI.
- Postgresql*. (18 de 09 de 2021). Obtenido de <https://www.postgresql.org/>
- Pressman, R. (2007). *Ingeniería de Software. Un enfoque práctico* (ed.: Mc Graw Hill ed.).

- Python Colombia. (9 de agosto de 2011). Obtenido de <https://sites.google.com/site/pythoncolombia/articulos/django>
- Rodríguez Cardona, M., & Cobas Aranda, D. (11 de 09 de 2010). *Metodología de evaluación de impactos de proyectos de investigación*. Obtenido de International Nuclear Information System (INIS): https://inis.iaea.org/collection/NCLCollectionStore/_Public/45/078/45078486.pdf.
- Rosa, A. V. (15 de septiembre de 2016). *repositorio.uci.cu*. Recuperado el 26 de 10 de 2018, de [repositorio.uci.cu: https://repositorio.uci.cu/jspui/bitstream/123456789/7860/1/TD_08708_16.pdf](https://repositorio.uci.cu/jspui/bitstream/123456789/7860/1/TD_08708_16.pdf)
- Salcedo Quevedo, R. (05 de 09 de 2020). *Web oficial de Universidad de San Martín de Porres*. Obtenido de <https://www.usmp.edu.pe/publicaciones/boletin/fia/info46/sistemas/articulo3.htm>
- Significados. (01 de 11 de 2020). *Significados.com*. Obtenido de <https://www.significados.com/proyecto-innovador/>
- Sommerville, I. (2007). *Software Engineering*. Edition ed.: Pearson Education.
- Sriramakrishnan, P., Kalaiselvi, T., Padmapriya, S., Shanthi, N., Ramkumar, S., & Kalaichelvi, N. (2019). An Medical Image File Formats and Digital Image Conversion. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9, 74-78. doi:10.35940/ijeat.A1093.1291S419
- Taylor, J. R., Williams, N., Cusack, R., Auer, T., Shafto, M. A., & Cam-CAN. (2017). The Cambridge Centre for Ageing and Neuroscience (Cam-CAN) data repository: Structural and functional MRI, MEG, and cognitive data from a cross-sectional adult lifespan sample. *NeuroImage*, 144, 262-269. doi:http://dx.doi.org/10.1016/j.neuroimage.2015.09.018
- The HDF Group. (2021). *The HDF Group*. Obtenido de THE HDF5 LIBRARY & FILE FORMAT: <https://www.hdfgroup.org/solutions/hdf5/>

The National Institute of Mental Health. (s.f.). (National Institute of Mental Health) Recuperado el 5 de Mayo de 2021, de The National Institute of Mental Health: <https://www.nimh.nih.gov/>

The PostgreSQL Global Development Group. (2019). *PostgreSQL*. Recuperado el Marzo de 2019, de PostgreSQL: <https://www.postgresql.org/docs/10/release-10-1.html>

Toledo. (2014).

Ubunlog. (2020). *ubunlog.com*. Obtenido de ubunlog.com: <https://ubunlog.com/pencil-modelos-prototipos-ubuntu/>

uci. (s.f.). *uci.cu*. Obtenido de <https://www.uci.cu/investigacion-y-desarrollo/productos/xauce/xauce-akademos-10>

Universidad de Ciencia y Tecnología Electrónica de China. (2021). Recuperado el 25 de junio de 2021, de <https://www.uestc.edu.cn/>

Visual Paradigm. (20 de 09 de 2021). Obtenido de <https://www.visual-paradigm.com/>

W3C. (2019). *we.org*. Obtenido de w3.org: <https://www.w3.org/Style/CSS/Overview.en.html>

w3schools.com. (2019). *www.w3schools.com*. Obtenido de [www.w3schools.com: https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp](https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp)

Anexos

Tabla 7 HU_3 Extender proyecto

Historia de usuario	
Numero: HU 3	Nombre Historia de usuario: Extender proyecto
Usuario: Julio T. Nerey Villanueva	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción: Quiero poder extender la fecha de entrega de un proyecto en el sistema, para tener flexibilidad en caso de que sea necesario más tiempo para completar las tareas y alcanzar los objetivos del proyecto.	

Tabla 8 HU 4 Buscar proyecto

Historia de usuario	
Numero: HU 4	Nombre Historia de usuario: Buscar proyecto
Usuario: Julio T. Nerey Villanueva	Iteración asignada: 1
Prioridad en negocio: Media	Puntos estimados: 1
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción: Quiero tener la capacidad de buscar un proyecto en la lista de proyectos del sistema, para poder encontrar rápidamente el proyecto que necesito revisar o trabajar.	
Observaciones: --	

Tabla 9 HU_5 Desactivar un proyecto

Historia de usuario	
Numero: HU 5	Nombre Historia de usuario: Desactivar un proyecto

Usuario: Julio T. Nerey Villanueva	Iteración asignada: 2
Prioridad en negocio: Media	Puntos estimados: 1
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción: Quiero poder desactivar un proyecto en el sistema, para inhabilitarlo temporalmente y indicar que no se está trabajando activamente en él.	
Observaciones: --	

Tabla 11 HU_7 Listar proyectos

Historia de usuario	
Numero: HU 7	Nombre Historia de usuario: Listar proyectos
Usuario: Julio T. Nerey Villanueva	Iteración asignada: 1
Prioridad en negocio: Baja	Puntos estimados: 1
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción: quiero poder ver una lista con todos los proyectos registrados en el sistema, para tener una visión general de los proyectos existentes.	
Observaciones: --	

Tabla 12 HU_8 Insertar RH

Historia de usuario

Numero: HU 8	Nombre Historia de usuario: Insertar RH
Usuario: Roddy Javier Garcia Mayea	Iteración asignada: 3
Prioridad en negocio: Media	Puntos estimados: 2
Riesgo en desarrollo: Medio	Puntos reales: 2
Descripción: quiero poder insertar un recurso humano en un proyecto	
Observaciones: --	

Tabla 13 HU_9 Editar datos de un RH

Historia de usuario	
Numero: HU 9	Nombre Historia de usuario: Editar datos de un RH proyecto
Usuario: Roddy Javier Garcia Mayea	Iteración asignada: 4
Prioridad en negocio: Alta	Puntos estimados: 2
Riesgo en desarrollo: Bajo	Puntos reales: 2
Descripción: quiero poder editar los datos de un recurso humano registrado en el sistema, para mantener la información actualizada y precisa.	
Observaciones: --	

Tabla 14 HU_10 Listar RH

Historia de usuario	
Numero: HU 10	Nombre Historia de usuario: Listar RH

Usuario: Roddy Javier Garcia **Iteración asignada:** 1
Mayea

Prioridad en negocio: Baja

Puntos estimados: 1

Riesgo en desarrollo: Bajo

Puntos reales: 1

Descripción: quiero poder ver una tabla con todos los recursos humanos disponibles, para facilitar la asignación de recursos a los proyectos.

Observaciones: --

Tabla 15 HU_11 Deshabilitar RH

Historia de usuario

Numero: HU 11 **Nombre Historia de usuario:** Deshabilitar RH

Usuario: Julio T. Nerey Villanueva

Iteración asignada: 2

Prioridad en negocio: Alto

Puntos estimados: 1

Riesgo en desarrollo: Bajo

Puntos reales: 1

Descripción: quiero poder deshabilitar un recurso humano en el sistema, para indicar que el recurso ya no está disponible para trabajar en proyectos.

Observaciones: --

Tabla 16 HU_12 Vincular RH a un Proyecto

Historia de usuario	
Numero: HU 12	Nombre Historia de usuario: Vincular RH a un Proyecto
Usuario: Roddy Javier Garcia Mayea	Iteración asignada: 3
Prioridad en negocio: Media	Puntos estimados: 2
Riesgo en desarrollo: Medio	Puntos reales: 2
Descripción: quiero poder asignar un recurso humano a un proyecto específico en el sistema, para establecer una relación entre el recurso y el proyecto y facilitar la asignación de tareas y seguimiento.	
Observaciones: --	

Tabla 17 HU_13 Crear Tarifas del proyecto

Historia de usuario	
Numero: HU 13	Nombre Historia de usuario: Crear Tarifas del proyecto
Usuario: Julio T. Nerey Villanueva	Iteración asignada: 4
Prioridad en negocio: Media	Puntos estimados: 2
Riesgo en desarrollo: Bajo	Puntos reales: 2
Descripción: quiero poder editar los datos relacionados con las tarifas de un recurso humano en un proyecto, para establecer las tarifas de facturación o remuneración asociadas al recurso.	
Observaciones: --	

Tabla 18 HU_14 Modificar Tarifas del proyecto

Historia de usuario	
Numero: HU 14	Nombre Historia de usuario: Modificar Tarifas del proyecto
Usuario: Julio T. Nerey Villanueva	Iteración asignada: 5
Prioridad en negocio: Alta	Puntos estimados: 3
Riesgo en desarrollo: Medio	Puntos reales: 3
Descripción: quiero poder listar la relación de los recursos humanos asociados a un proyecto y ver las tarifas establecidas para cada uno, para tener una visión general de la relación entre los recursos y las tarifas en el proyecto.	
Observaciones: --	

Tabla 19 HU_15 Listar Tarifas del proyecto

Historia de usuario	
Numero: HU 15	Nombre Historia de usuario: Listar Tarifas del proyecto
Usuario: Roddy Javier Garcia Mayea	Iteración asignada: 2
Prioridad en negocio: Baja	Puntos estimados: 1
Riesgo en desarrollo: Bajo	Puntos reales: 1
Descripción: Quiero poder ver una lista de las tarifas totales del proyecto, para tener una comprensión clara de las tarifas asociadas a los recursos humanos involucrados en el proyecto.	

Observaciones: --
