**UNIVERSIDAD AGRARIA DE LA HABANA**

**“Fructuoso Rodríguez Pérez”**

**Facultad de Ciencias Técnicas**

**Ingeniería informática**

****

**Informe de la Práctica Laboral de Tercer Año**

|  |
| --- |
| **Grupo:301**  **Curso:2024**  **Mayabeque, 2024** |

|  |
| --- |
| **Autores:**  -José Raúl Márquez Estévez  -Luis E. López Nurques |

|  |
| --- |
| **Tutores Entidad:**   * Ing. Idariel Castañeda Castañeda * Ing. Neylis Chávez Millares |

**Sistema Para La Gestión De Información Del Departamento Económico De La Universidad Agraria De La HABANA (SigeWeb2)**

**Resumen**

Este trabajo se centra en la mejora de la generación de reportes financieros y la gestión de bajas técnicas en la Universidad Agraria de La Habana “Fructuoso Rodríguez Pérez”. La necesidad de este trabajo radica en las limitaciones del sistema actual, SigeWeb, que no proporciona resultados precisos ni cumple con los requerimientos actuales. Los objetivos concretos incluyen la implementación de un nuevo sistema basado en SigeWeb, mejorándolo y añadiendo nuevas funcionalidades, como la automatización de procesos de bajas técnicas y la creación de una nueva interfaz de usuario. Los resultados más relevantes incluyen la generación de reportes financieros más precisos y confiables, la reducción de errores y tiempos en los procesos administrativos, y una mayor transparencia y control en la gestión de cuentas y gastos.

**Palabras claves**: generación de reportes, bajas técnicas, automatización, interfaz de usuario, gestión financiera.

**Summary**

This work focuses on improving the generation of financial reports and the management of technical write-offs at the Agrarian University of Havana “Fructuoso Rodríguez Pérez”. The need for this work arises from the limitations of the current system, SigeWeb, which does not provide accurate results nor meets current requirements. The specific objectives include implementing a new system based on SigeWeb, improving it, and adding new functionalities, such as the automation of technical write-off processes and the creation of a new user interface. The most relevant results include the generation of more accurate and reliable financial reports, the reduction of errors and time in administrative processes, and greater transparency and control in the management of accounts and expenses.

**Keywords**: report generation, technical write-offs, automation, user interface, financial management.

Índice

[Introducción 9](#_Toc177439287)

[CAPÍTULO 1: Estado del arte 13](#_Toc177439288)

[1.1. Introducción 13](#_Toc177439289)

[1.2. Flujo Actual de los Procesos 13](#_Toc177439290)

[1.3. Sistemas existentes vinculados al campo de acción 15](#_Toc177439291)

[1.4. Tendencias y Tecnologías Actuales 16](#_Toc177439292)

[1.5. Conclusiones 31](#_Toc177439293)

[CAPÍTULO 2: Negocio y Requerimientos 32](#_Toc177439294)

[2.1. Introducción 32](#_Toc177439295)

[2.2. Actores y Trabajadores del Negocio 32](#_Toc177439296)

[2.3. Diagrama de Casos de Uso del Negocio 34](#_Toc177439297)

[2.4. Diagramas de Actividades de los Casos de Uso 34](#_Toc177439298)

[Explicación: 38](#_Toc177439299)

[Actor 38](#_Toc177439300)

[Evento 38](#_Toc177439301)

[Responsable de Área 38](#_Toc177439302)

[Rellena el formulario de bajas técnicas. 38](#_Toc177439303)

[Responsable de Área 38](#_Toc177439304)

[Envía el formulario de bajas tecinas 38](#_Toc177439305)

[Responsable de Economía 38](#_Toc177439306)

[Procesa la información del formulario 38](#_Toc177439307)

[Personal de Mantenimiento 38](#_Toc177439308)

[Evalúa si el activo es reparable o no 38](#_Toc177439309)

[Personal de Mantenimiento 38](#_Toc177439310)

[Si el activo es reparable se añade el presupuesto de reparación 38](#_Toc177439311)

[Personal de Mantenimiento 38](#_Toc177439312)

[Si el activo no es reparable se da de baja 38](#_Toc177439313)

[Personal de Mantenimiento 38](#_Toc177439314)

[Actualización del Estado del Activo 38](#_Toc177439315)

[2.5. Modelo de Objetos del Negocio 39](#_Toc177439316)

[Explicación 39](#_Toc177439317)

[2.6. Reglas del Negocio a Considerar 41](#_Toc177439318)

[2.7. Conclusiones 42](#_Toc177439319)

[CAPÍTULO 3: Definición de Requisitos del Sistema 43](#_Toc177439320)

[3.1. Introducción 43](#_Toc177439321)

[3.2. Definición de Requisitos Funcionales y No Funcionales 43](#_Toc177439322)

[3.2.1. Listado de Requisitos Funcionales del Sistema a Desarrollar 43](#_Toc177439323)

[3.2.2. Listado de Requisitos No Funcionales del Sistema a Desarrollar 53](#_Toc177439324)

[3.3. Descripción de los Actores del Sistema 55](#_Toc177439325)

[3.4. Diagrama de Casos de Uso del Sistema \*\*\*\*\*\*\*\*\*\*\*\* 56](#_Toc177439326)

[3.5. Descripción de los Casos de Uso del Sistema 57](#_Toc177439327)

[3.6. Conclusiones 78](#_Toc177439328)

[CAPÍTULO 4: Diseño e Implementación del Sistema 80](#_Toc177439329)

[4.1. Introducción 80](#_Toc177439330)

[4.2. Descripción de la Solución Propuesta 80](#_Toc177439331)

[4.3. Diseño del Sistema 81](#_Toc177439332)

[4.4. Modelo de Base de Datos 88](#_Toc177439333)

[4.4.1 Modelo lógico de la Base de Datos 88](#_Toc177439334)

[4.4.2 Modelo físico de la Base de Datos 88](#_Toc177439335)

[4.5. Validación del Sistema 88](#_Toc177439336)

[4.5.1. Pruebas de Caja Negra 88](#_Toc177439337)

[4.5.2. Prueba de caja blanca 95](#_Toc177439338)

[4.6. Conclusiones 112](#_Toc177439339)

[CONCLUSIONES 113](#_Toc177439340)

[RECOMENDACIONES 114](#_Toc177439341)

[BIBLIOGRAFÍA 115](#_Toc177439342)

[Anexos 116](#_Toc177439343)

**Índice de Figuras**

[Ilustración 1 personal de mantenimiento 34](#_Toc177439344)

[Ilustración 2 responsable de Área 34](#_Toc177439345)

[Ilustración 3 generación de reporte 35](#_Toc177439346)

[Ilustración 4 bajas tecnicas 37](#_Toc177439347)

[Ilustración 5 Modelo de negocio: Responsable de mantenimiento 39](#_Toc177439348)

[Ilustración 6 Modelo de negocio: Responsable de Área 39](#_Toc177439349)

[Ilustración 7Diagrama de Casos de Uso 81](#_Toc177439350)

[Ilustración 8 “Generación de Reportes” 82](#_Toc177439351)

[Ilustración 9 “Gestión de Bajas Técnicas” 83](#_Toc177439352)

[Ilustración 10 Diagrama de Clases 84](#_Toc177439353)

[Ilustración 11 Diagrama de Secuencia para “Generación de Reportes” 85](#_Toc177439354)

[Ilustración 12 Diagrama de Secuencia para “Gestión de Bajas Técnicas” 86](#_Toc177439355)

[Ilustración 13 Diagrama de Componentes 87](#_Toc177439356)

[Ilustración 14 Modelo lógico de la Base de Datos 88](#_Toc177439357)

**Índice de Tablas**

[Tabla 1 Actores del negocio 33](#_Toc177439368)

[Tabla 2 Trabajadores del Negocio 34](#_Toc177439369)

[Tabla 3 Explicación Generar Reporte 37](#_Toc177439370)

[Tabla 4 Explicación Bajas Técnicas 39](#_Toc177439371)

[Tabla 5 Requisitos Funcionales del Sistema 44](#_Toc177439372)

[Tabla 6 Requisitos No Funcionales 54](#_Toc177439373)

[Tabla 7 Actores del Sistema 56](#_Toc177439374)

[Tabla 8 Descripción de los Casos de Uso del Sistema: Autenticar Usuario 58](#_Toc177439375)

[Tabla 9 Descripción de los Casos de Uso del Sistema: Generación de Reportes 59](#_Toc177439376)

[Tabla 10 Descripción de los Casos de Uso del Sistema: Gestión de Bajas Técnicas 61](#_Toc177439377)

[Tabla 11 Descripción de los Casos de Uso del Sistema: Administracion de Usuarios 62](#_Toc177439378)

[Tabla 12 Descripción de los Casos de Uso del Sistema: Trazabilidad de Acciones 64](#_Toc177439379)

[Tabla 13Descripción de los Casos de Uso del Sistema: Diseño y Creación de Filtros 65](#_Toc177439380)

[Tabla 14Descripción de los Casos de Uso del Sistema: Uso de Filtros en Reportes 67](#_Toc177439381)

[Tabla 15Descripción de los Casos de Uso del Sistema: Reporte de Activos Fijos por Clasificación 68](#_Toc177439382)

[Tabla 16Descripción de los Casos de Uso del Sistema: Reporte de Útiles y Herramientas 70](#_Toc177439383)

[Tabla 17Descripción de los Casos de Uso del Sistema: Reporte de Productos en Almacen 71](#_Toc177439384)

[Tabla 18Descripción de los Casos de Uso del Sistema: Reporte de Listado de Centro de Costos 73](#_Toc177439385)

[Tabla 19Descripción de los Casos de Uso del Sistema: Reporte de Listado de Activos por Numero de Activos 74](#_Toc177439386)

[Tabla 20Descripción de los Casos de Uso del Sistema: Reporte de Activos Fijos 76](#_Toc177439387)

[Tabla 21Descripción de los Casos de Uso del Sistema: Reporte de Repuestos de Activos Fijos por Clasificación 77](#_Toc177439388)

[Tabla 22 Prueba de Autenticar Usuario 90](#_Toc177439389)

[Tabla 23 Prueba de Generación de Reportes 91](#_Toc177439390)

[Tabla 24 Prueba de Gestión de Bajas Técnicas 92](#_Toc177439391)

[Tabla 25 Prueba de Administración de Usuarios 93](#_Toc177439392)

[Tabla 26 Prueba de Trazabilidad de Acciones 94](#_Toc177439393)

# Introducción

Un sistema de información (SI) es un conjunto interrelacionado de componentes que recopilan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones, la coordinación y el control en una organización. Los SI son esenciales para el funcionamiento moderno de las empresas y las instituciones, ya que permiten automatizar tareas, optimizar procesos, gestionar información de manera eficiente y obtener una visión integral de las operaciones.

Los reportes son una herramienta fundamental dentro de los sistemas de información. Permiten visualizar y analizar la información recopilada, ofreciendo una comprensión clara de la situación actual de la organización. A nivel mundial, los reportes son cruciales para la toma de decisiones estratégicas, la evaluación del desempeño y la gestión de riesgos.

En el contexto cubano, los reportes juegan un papel crucial en la gestión eficiente de los recursos, la optimización de los procesos productivos y el control de las operaciones, especialmente en un entorno donde la información precisa y confiable es fundamental para el desarrollo económico y social del país.

En la Universidad Agraria de La Habana “Fructuoso Rodríguez Pérez”, la gestión de las cuentas y gastos es fundamental para garantizar el adecuado funcionamiento administrativo. Sin embargo, el sistema actual SigeWeb, utilizado para obtener reportes financieros, ha demostrado no proporcionar resultados precisos. Además, no cumple con los requerimientos para la creación de reportes y otras tareas actuales, como el proceso de bajas técnicas de los activos en las diferentes áreas de trabajo, que se efectúa de manera manual. Esto origina tardanzas y errores en el proceso de llenado y obtención de reportes.

Lo anterior expuesto conlleva al siguiente **problema** ¿Cómo mejorar la generación de reportes financieros en la Universidad Agraria de La Habana ante las limitaciones del sistema actual?

Para resolver el problema planteado se traza como **objetivo general**: Implementar un nuevo sistema basado en el SIEWEB, para mejorar la generación de reportes financieros en la Universidad Agraria de la Habana antes las limitaciones del sistema actual.

Para darle cumplimiento al objetivo general se identifican los siguientes **objetivos específicos**:

* Establecer los referentes teóricos para el desarrollo del sistema.
* Diseñar un sistema informático basado en versión de SIEWEB.
* Implementar un sistema informático que mejore deficiencias detectadas en la generación de reportes antes las limitaciones del sistema actual.
* Validar el sistema mediante pruebas de aceptación, caja blanca y caja negra.

**Objeto de estudio: Sistema informático de generación de reportes financieros**

**Campo de acción:** Sistema informático para la generación de reportes financieros para la Universidad Agraria de la Habana.

**Actualidad y Necesidad del Trabajo**: La necesidad de este trabajo radica en la urgencia de mejorar la precisión y eficiencia en la gestión de reportes financieros en la Universidad Agraria de La Habana. La implementación de un nuevo sistema que automatice procesos y mejore la interfaz de usuario es crucial para adaptarse a las necesidades actuales del Departamento de Economía, Áreas de Trabajo y de Mantenimiento.

**Aportes Prácticos**

* Se obtendrán resultados financieros más precisos y confiables, reduciendo la posibilidad de errores y discrepancias en la información reportada.
* La automatización de los procesos de bajas técnicas de activos permitirá ahorrar tiempo y reducir los errores asociados a la intervención humana. Esto agilizará el proceso de llenado y obtención de reportes financieros.
* La nueva interfaz de usuario facilitará la interacción con la aplicación, mejorando la experiencia de usuario y reduciendo la curva de aprendizaje para su utilización.
* La implementación de la funcionalidad de guardar traza de acción permitirá tener un registro detallado de las operaciones realizadas en el sistema. Esto facilitará el seguimiento y la auditoría de las actividades financieras, proporcionando una mayor transparencia y control en la gestión de cuentas y gastos.

# CAPÍTULO 1: Estado del arte

## 1.1. Introducción

En este capítulo se abordarán los aspectos fundamentales relacionados con el flujo actual de los procesos en la Universidad Agraria de La Habana “Fructuoso Rodríguez Pérez”, los sistemas existentes vinculados a la problemática en cuestión, y las tendencias y tecnologías actuales que pueden ser aplicadas para mejorar la generación de reportes financieros y el proceso de bajas técnicas de activos.

## 1.2. Flujo Actual de los Procesos

El área de economía de la Universidad Agraria de La Habana requiere la generación de reportes de activos fijos por clasificación, útiles y herramientas. Actualmente, estos reportes se ajustan según la información requerida mediante el uso de filtros en el formulario de requisitos. Los responsables de área tienen la opción de generar diferentes tipos de reportes ajustados a sus áreas de responsabilidad. Además, se generan reportes de productos que muestran la existencia de productos en diferentes tipos de almacenes, también con filtros de formulario.

El proceso de bajas técnicas del sistema actual facilita al responsable de área rellenar un formulario para proporcionar la información necesaria al departamento de economía. Una vez recibida la información, el departamento de economía ejecuta el proceso, proporcionando información sobre el activo y el área donde se encuentra, así como información sobre el personal de mantenimiento responsable del activo. Si el activo es reparable, se proporciona un formulario para añadir el presupuesto de reparación. En caso de que el activo sea irreparable, la información se actualiza y el activo se da de baja. Si el presupuesto de reparación no es aprobado, el activo también puede ser dado de baja por los responsables de economía.

Actualmente, los reportes del sistema incluyen:

* Listado de centro de costos
* Listado de activos por número de activos
* Listado de productos
* Reporte de activos por clasificación
* Reporte de activos fijos
* Repuestos de activos fijos por clasificación
* Reporte de útiles y herramientas

Estos reportes cuentan con sus respectivos filtros y un área de visita con acceso a todos los reportes ajustados para una completa visibilidad.

## 1.3. Sistemas existentes vinculados al campo de acción

La generación de reportes se define como un proceso continuo que reúne, organiza, coordina y controla los datos financieros de una institución, para proporcionar informes precisos y relevantes que respondan adecuadamente a las necesidades de los usuarios.

Existen numerosos sistemas de gestión de información. Un ejemplo es el Sistema para la Gestión de la Información Docente del Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria, que gestiona las demandas de docencia cubiertas y no cubiertas por año, la carga docente de cada profesor, las tutorías, el historial de asignaturas y los proyectos docentes.

En Cuba, también se desarrollan y utilizan estos sistemas. Por ejemplo, el Sistema Informático para la Gestión de la Información de Expedientes Disciplinarios de la Facultad 2 en la Universidad de las Ciencias Informáticas (UCI), que enfatiza la seguridad, eficiencia e integralidad de la información. Otro ejemplo es el Sistema Informático para la Gestión de la Información Profesoral en los Departamentos Docentes de la Universidad de Holguín “Oscar Lucero Moya”, que facilita el registro, organización, actualización y conservación de la información relacionada con los profesores.

En la UNAH se utilizan diversos sistemas de gestión, como el sistema de gestión de plan de trabajo para los profesores, el sistema de gestión de evaluaciones integrales de los estudiantes, el sistema de gestión del currículum de los profesores y el SIGENU, que maneja el proceso docente de los estudiantes. En el Departamento de Informática se emplean estos sistemas, pero surge la necesidad de actualizar el sistema de generación de reportes actual SIEWEB por las deficiencias que presenta.

## 1.4. Tendencias y Tecnologías Actuales

En la actualidad, existen diversas tendencias, técnicas, tecnologías y metodologías que pueden ser aplicadas para mejorar la generación de reportes financieros y la gestión de activos. Es fundamental realizar un estudio crítico y valorativo de estas tendencias para determinar cuáles son las más adecuadas para la solución del problema en cuestión.

Este proyecto se ha desarrollado utilizando un conjunto de tecnologías y herramientas robustas para garantizar la eficiencia, escalabilidad y calidad del sistema de información. A continuación, se presenta una breve descripción de cada una:

**Herramientas de Desarrollo:**

**Apache NetBeans 19**:

**Apache NetBeans 19** es una versión reciente del entorno de desarrollo integrado (IDE) Apache NetBeans, lanzada el 1 de septiembre de 2023. Este IDE es gratuito y de código abierto, y permite desarrollar aplicaciones de escritorio, móviles y web en varios lenguajes, incluyendo Java, HTML5, PHP y C++.[[1](https://netbeans.apache.org/front/main/download/nb19/index.html)][[2](https://docs.oracle.com/cd/E40938_01/doc.74/e40142/gs_nbeans.htm)]

Algunas de las características destacadas de Apache NetBeans 19 incluyen:

* **Compatibilidad con JDK 11+**: Requiere JDK 11 o superior y soporta oficialmente JDK 11, 17 y 20.
* **Edición rápida e inteligente**: Resalta el código fuente sintáctica y semánticamente, y permite refactorizar el código fácilmente.
* **Soporte para el ciclo completo de desarrollo**: Desde la creación del proyecto hasta la depuración, el perfilado y el despliegue.

**TIBCO Jaspersoft Studio 6.20.0**:

Es una herramienta de diseño de informes de escritorio que permite desarrollar visualizaciones de datos y crear informes detallados y altamente formateados. Con un entorno de diseño avanzado, Jaspersoft Studio facilita la creación de informes y visualizaciones de datos precisos y personalizados.

**Obtuve esta información de varias fuentes en la web, incluyendo la documentación de TIBCO y otros sitios de software.**

**Visual Paradigm 6**:

Es una herramienta de diseño y gestión de sistemas de TI que ofrece una amplia gama de funcionalidades para el modelado, análisis y gestión de proyectos. Esta versión incluye herramientas avanzadas para la colaboración en equipo, la gestión de proyectos ágiles y la transformación digital. Entre sus características destacan:

**Modelado y diagramación**: Soporta UML, BPMN, DFD, ERD, SysML y más.

**Gestión de proyectos ágiles**: Herramientas para Scrum, PMBOK, LeSS y Nexus.

**Suite de herramientas DevOps**: Generación y reversión de código, visualización de flujo de pantalla y diseño de prototipos.

**Colaboración en equipo**: Trabajo simultáneo y colaborativo en el mismo proyecto, con almacenamiento seguro en la nube.

**Obtuve esta información de varias fuentes en la web, incluyendo la página oficial de Visual Paradigm.**

**PlantUML**:

Es una herramienta de código abierto que permite a los desarrolladores y diseñadores crear diagramas de manera programática. Utiliza una sintaxis basada en texto para describir los diagramas, que luego se renderizan automáticamente en representaciones visuales.

Con PlantUML, puedes crear una variedad de diagramas UML, como diagramas de secuencia, diagramas de casos de uso, diagramas de clases, diagramas de objetos, diagramas de actividades, diagramas de componentes, diagramas de despliegue, diagramas de estados y diagramas de tiempo. Además, también soporta diagramas no UML, como diagramas de red, diagramas de Gantt, diagramas de mapas mentales y más.[[3](https://github.com/plantuml/plantuml)]

**Lenguajes de Programación y Tecnologías:**

**Java 11**:

**Java 11**, lanzado en septiembre de 2018, es una versión de soporte a largo plazo (LTS) que incluye varias características nuevas y mejoras. Algunas de las características más destacadas son[[4](https://www.baeldung.com/java-11-new-features)][[5](https://www.geeksforgeeks.org/java-11-features-and-comparison/)]:

* **Nuevos métodos de cadena**: Métodos como isBlank(), lines(), strip(), stripLeading(), stripTrailing() y repeat() se han añadido para facilitar la manipulación de cadenas[1](https://www.baeldung.com/java-11-new-features)[2](https://www.geeksforgeeks.org/java-11-features-and-comparison/).
* **Métodos de archivo**: Nuevos métodos estáticos readString() y writeString() en la clase Files para leer y escribir cadenas en archivos de manera más sencilla[1](https://www.baeldung.com/java-11-new-features)[2](https://www.geeksforgeeks.org/java-11-features-and-comparison/).
* **Colección a un array**: La interfaz java.util.Collection ahora contiene un método toArray() que toma un argumento IntFunction, facilitando la creación de un array del tipo correcto a partir de una colección[1](https://www.baeldung.com/java-11-new-features).
* **Método**not**en**Predicate: Se ha añadido un método estático not a la interfaz Predicate[1](https://www.baeldung.com/java-11-new-features).

**JavaScript (JS)**:

Esun lenguaje de programación ligero e interpretado (o compilado justo a tiempo) con funciones de primera clase. Es conocido principalmente como el lenguaje de scripting para páginas web, pero también se utiliza en muchos entornos no basados en navegadores, como Node.js, Apache CouchDB y Adobe Acrobat.

JavaScript permite crear contenido dinámico y actualizaciones en tiempo real en las páginas web, controlar multimedia, animar gráficos 2D/3D y mucho más. Es una tecnología central del desarrollo web junto con HTML y CSS.[[6](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)]

**HTML5**:

Es un lenguaje de marcado utilizado para estructurar y presentar contenido en la web. Es la quinta y última versión importante de HTML, y se ha convertido en un estándar vivo mantenido por el Web Hypertext Application Technology Working Group (WHATWG). HTML5 incluye muchas características nuevas y mejoradas, como[[7](https://en.wikipedia.org/wiki/HTML5)]:

* **Elementos multimedia**: Nuevos elementos como <video>, <audio> y <canvas> para manejar contenido multimedia y gráfico de manera nativa.
* **Elementos semánticos**: Nuevos elementos como <main>, <section>, <article>, <header>, <footer>, <aside>, <nav> y <figure> para enriquecer el contenido semántico de los documentos.
* **Compatibilidad con dispositivos móviles**: Diseñado teniendo en cuenta los dispositivos de baja potencia, lo que lo hace ideal para aplicaciones móviles multiplataforma.
* **APIs y modelos de procesamiento**: Introduce APIs y modelos de procesamiento detallados para fomentar implementaciones más interoperables.

**CSS3**:

Es la última evolución del lenguaje CSS, utilizado para describir la presentación de documentos escritos en HTML o XML. CSS3 introduce muchas características nuevas y mejoras con respecto a versiones anteriores, como[[8](https://maven.apache.org/)]:

* **Esquinas redondeadas**: Permite crear bordes con esquinas redondeadas usando la propiedad border-radius.
* **Sombras**: Añade sombras a elementos y texto con las propiedades box-shadow y text-shadow.
* **Gradientes**: Crea transiciones suaves entre dos o más colores con las propiedades linear-gradient y radial-gradient.
* **Transiciones y animaciones**: Permite animar cambios en las propiedades CSS con las propiedades transition y animation.
* **Nuevos diseños**: Introduce nuevos modelos de diseño como el diseño de múltiples columnas, el diseño flexible (flexbox) y el diseño de cuadrícula (grid).

**Maven**:

Es una herramienta de automatización de compilación y gestión de proyectos desarrollada principalmente para proyectos basados en Java. Utiliza el concepto de un modelo de objeto de proyecto (POM) para gestionar la construcción, la documentación y las dependencias de un proyecto desde una pieza central de información. Algunas de las características clave de Maven incluyen:

* **Gestión de dependencias**: Simplifica el proceso de inclusión de bibliotecas externas en el proyecto y puede descargar automáticamente las bibliotecas necesarias.
* **Estructura de proyecto estandarizada**: Proporciona una estructura de carpetas estándar para el código fuente, el código de prueba y otros recursos.
* **Ciclo de vida de compilación**: Maneja diferentes fases de compilación como validación, compilación, prueba, empaquetado, verificación y despliegue.
* **Plugins**: Soporta diferentes plugins para realizar tareas como compilación, pruebas, empaquetado, despliegue y generación de documentación.
* **Integración con IDEs**: Compatible con varios entornos de desarrollo integrados como Eclipse, IntelliJ IDEA y NetBeans.

**Spring Boot 2.5.6**:

Es una versión del popular framework Spring Boot, lanzada en septiembre de 2021. Esta versión facilita la creación de aplicaciones Java independientes y listas para producción con una configuración mínima. Algunas de las características destacadas incluyen:

* **Compatibilidad con JDK 11+**: Soporta JDK 11 y versiones superiores.
* **Mejoras en la gestión de dependencias**: Simplifica la inclusión de bibliotecas externas en el proyecto[2](https://bing.com/search?q=Spring+Boot+2.5.6+description).
* **Soporte para contenedores servlet**: Facilita la implementación de aplicaciones en contenedores servlet[1](https://docs.spring.io/spring-boot/docs/2.5.6/reference/html/index.html).
* **Herramientas de línea de comandos**: Proporciona herramientas para ejecutar scripts de Spring[3](https://docs.spring.io/spring-boot/docs/2.5.6/reference/html/getting-started.html).

Obtuve esta información de varias fuentes en la web, incluyendo la [documentación oficial de Spring Boot](https://docs.spring.io/spring-boot/docs/2.5.6/reference/html/index.html).

**Apache Tomcat 10.1.20**:

Es una versión del servidor web y contenedor de servlets Apache Tomcat, que proporciona un entorno robusto para desplegar aplicaciones web basadas en Java. Esta versión implementa las especificaciones Servlet 6.0 y Pages 3.1 de Jakarta EE, y ofrece muchas características adicionales que lo hacen útil para desarrollar y desplegar aplicaciones y servicios web.

Algunas de las características destacadas incluyen:

* **Compatibilidad con Java EE a Jakarta EE**: La migración de Java EE a Jakarta EE ha cambiado el paquete principal de javax.\* a jakarta.\*, lo que puede requerir cambios en el código para migrar aplicaciones de versiones anteriores.
* **Ligero y escalable**: Su naturaleza ligera y escalabilidad lo hacen una opción popular entre desarrolladores y organizaciones.
* **Soporte para múltiples especificaciones**: Soporta las especificaciones Java Servlet, JavaServer Pages (JSP) y Java Expression Language (EL).

**Bootstrap (4.6.1):**

Es una versión del popular framework front-end Bootstrap, lanzada el 28 de octubre de 2021. Esta versión incluye varias mejoras y correcciones de errores, así como actualizaciones de accesibilidad. Algunas de las características destacadas incluyen:

* **Reimplementación de funciones de división de Sass**: Se reemplazaron las funciones de división de Sass con multiplicación y una función personalizada divide().
* **Mejoras en la accesibilidad**: Ajustes para mejorar la accesibilidad en varios componentes.
* **Correcciones de errores generales**: Incluye varias correcciones menores para mejorar la estabilidad y el rendimiento.

Obtuve esta información de varias fuentes en la web, incluyendo la [página oficial de Bootstrap](https://getbootstrap.com/docs/4.6/getting-started/introduction/) y el [blog de Bootstrap](https://blog.getbootstrap.com/2021/10/28/bootstrap-4.6.1/).

**Postman (8.0.2):**

Es una versión de la popular plataforma de API que ayuda a los desarrolladores a construir y utilizar APIs de manera más eficiente. Esta versión incluye varias mejoras y características nuevas, como[[9](https://www.postman.com/downloads/)]:

* Interfaz de usuario mejorada: Facilita la navegación y el uso de la plataforma.
* Mejoras en la colaboración: Permite a los equipos trabajar juntos de manera más efectiva en proyectos de API.
* Nuevas herramientas de depuración: Facilita la identificación y resolución de problemas en las APIs.

**Base de Datos y Lenguaje de Consulta:**

**PostgreSQL 13**:

Es una versión avanzada del sistema de gestión de bases de datos relacional de código abierto PostgreSQL, lanzada el 24 de septiembre de 2020. Esta versión incluye mejoras significativas en su sistema de indexación y búsqueda, lo que beneficia a las bases de datos grandes con ahorros de espacio y ganancias de rendimiento. Algunas de las características destacadas incluyen[[10](https://www.postgresql.org/about/news/postgresql-13-released-2077/)]:

* **Desduplicación de entradas de índice B-tree**: Reduce el uso de espacio y mejora el rendimiento general de las consultas.
* **Ordenamiento incremental**: Acelera el ordenamiento de datos en pasos posteriores de una consulta.
* **Vacuuming paralelo**: Mejora el rendimiento del vacuuming de índices.
* **Mejor planificación de consultas**: Utiliza estadísticas extendidas para crear planes de consulta mejorados.

**SQL**:

Es un lenguaje de programación de dominio específico utilizado para gestionar y manipular datos en sistemas de gestión de bases de datos relacionales (RDBMS). Es especialmente útil para manejar datos estructurados, es decir, datos que incorporan relaciones entre entidades y variables[[11](https://en.wikipedia.org/wiki/SQL)].

SQL permite realizar una variedad de operaciones, tales como:

* **Consultas de datos**: Recuperar datos de una base de datos.
* **Manipulación de datos**: Insertar, actualizar y eliminar registros en una base de datos.
* **Definición de datos**: Crear y modificar esquemas de bases de datos.
* **Control de acceso a datos**: Establecer permisos en tablas, procedimientos y vistas.

SQL se ha convertido en un estándar de la American National Standards Institute (ANSI) desde 1986 y de la International Organization for Standardization (ISO) desde 1987.

**Metodología de Desarrollo:**

**RUP (Rational Unified Process)**:

Es un marco de trabajo iterativo para el desarrollo de software, creado por Rational Software Corporation, una división de IBM desde 2003. RUP no es un proceso prescriptivo único, sino un marco adaptable que puede ser personalizado por las organizaciones de desarrollo y los equipos de proyectos de software según sus necesidades[[12](https://en.wikipedia.org/wiki/Rational_Unified_Process)].

Algunas características clave de RUP incluyen:

* **Enfoque iterativo e incremental**: Mejora el conocimiento del problema a través de revisiones consecutivas.
* **Centrado en la arquitectura y dirigido por casos de uso**: Gestiona el riesgo y es flexible al cambio.
* **Fases del ciclo de vida**: Incepción, Elaboración, Construcción, Transición y Producción.

## 1.5. Conclusiones

En este capítulo se ha realizado un análisis crítico del flujo actual de los procesos en la Universidad Agraria de La Habana, identificando las causas de la situación problemática y las posibles mejoras. Además, se ha presentado un estado del arte de las soluciones existentes y un estudio de las tendencias y tecnologías actuales que pueden ser aplicadas para mejorar la generación de reportes financieros y la gestión de activos.

# CAPÍTULO 2: Negocio y Requerimientos

## 2.1. Introducción

En este capítulo se abordarán los aspectos relacionados con la metodología de desarrollo RUP (Rational Unified Process) aplicada al desarrollo del nuevo sistema para la generación de reportes financieros y la gestión de bajas técnicas en la Universidad Agraria de La Habana “Fructuoso Rodríguez Pérez”. Se describirán los actores y trabajadores del negocio, los diagramas de casos de uso y actividades, el modelo de objetos del negocio y las reglas del negocio a considerar.

## 2.2. Actores y Trabajadores del Negocio

En el sistema web desarrollado, se identifican los siguientes actores y trabajadores del negocio:

Tabla 1 Actores del negocio

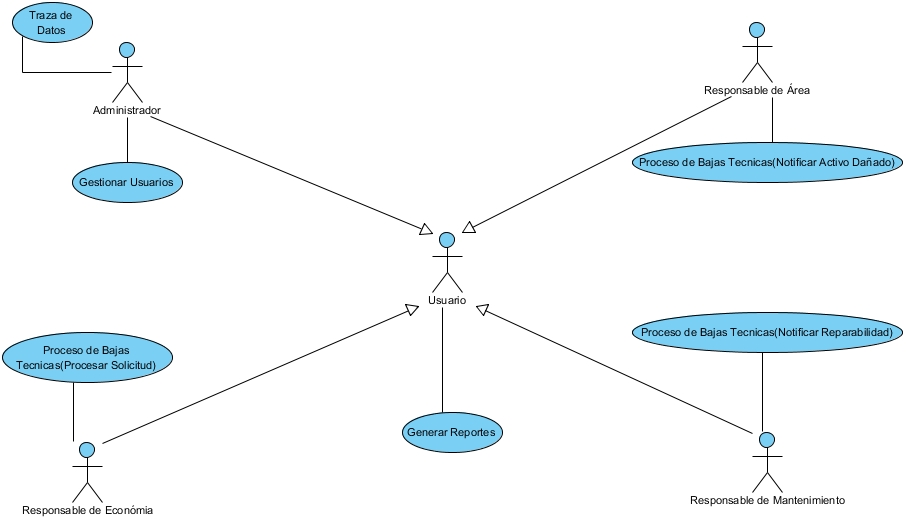
|  |  |
| --- | --- |
| Actores | Descripción |
| **Responsable de Área** | * Encargado de generar reportes específicos para su área de responsabilidad y de rellenar formularios para el proceso de bajas técnicas. |
| **Personal de Mantenimiento** | * Recibe información sobre los activos y, en caso de ser reparables, añade el presupuesto para su reparación. |

Tabla 2 Trabajadores del Negocio

|  |  |
| --- | --- |
| Trabajadores | Descripción |
| Responsable de Economía | * Recibe y procesa la información de los reportes y formularios, y toma decisiones sobre la reparación o baja de activos. |

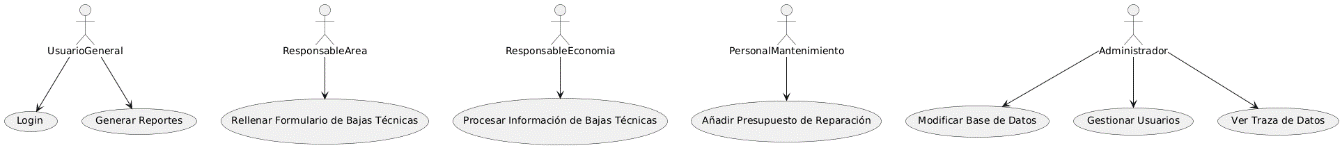
## 2.3. Diagrama de Casos de Uso del Negocio

El diagrama de casos de uso del negocio muestra las interacciones entre los actores y el negocio, destacando las funcionalidades principales como la generación de reportes, la gestión de bajas técnicas y la administración del sistema



## 

## 2.4. Diagramas de Actividades de los Casos de Uso



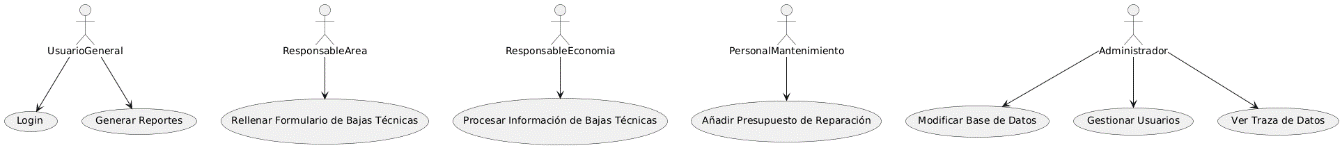
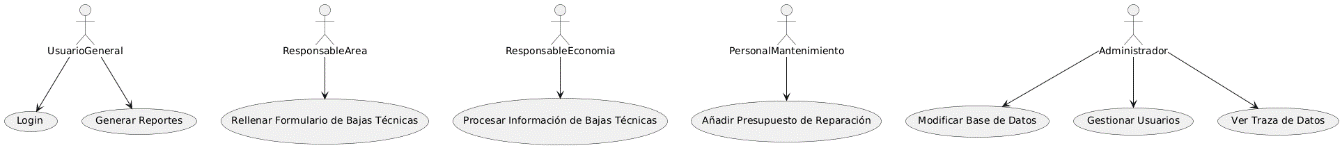


Ilustración 1 personal de mantenimiento

Ilustración 2 responsable de Área

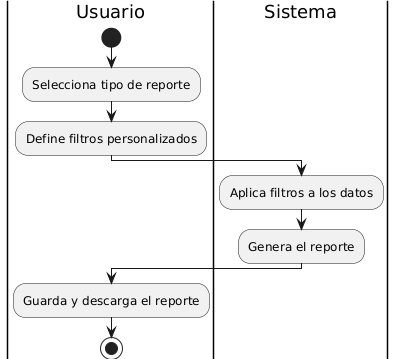
****Los diagramas de actividades detallan los pasos específicos que cada actor sigue para completar los casos de uso identificados. Estos diagramas ayudan a visualizar el flujo de trabajo y a identificar posibles mejoras en los procesos.

Ilustración 3 generación de reporte

**Generar Reporte:**

**Explicación:**

Tabla 3 Explicación Generar Reporte

|  |  |
| --- | --- |
| **Actor** | **Evento** |
| Responsable de Área | El usuario elige el tipo de reporte que desea generar |
| Responsable de Área | El usuario establece los filtros que se aplicaran |
| **----------------------------------------------** | El sistema aplica los filtros definidos por el usuario a los datos |
| **----------------------------------------------** | El sistema genera el reporte con los datos filtrados |
| Responsable de Área | El usuario guarda o descarga el reporte generado |

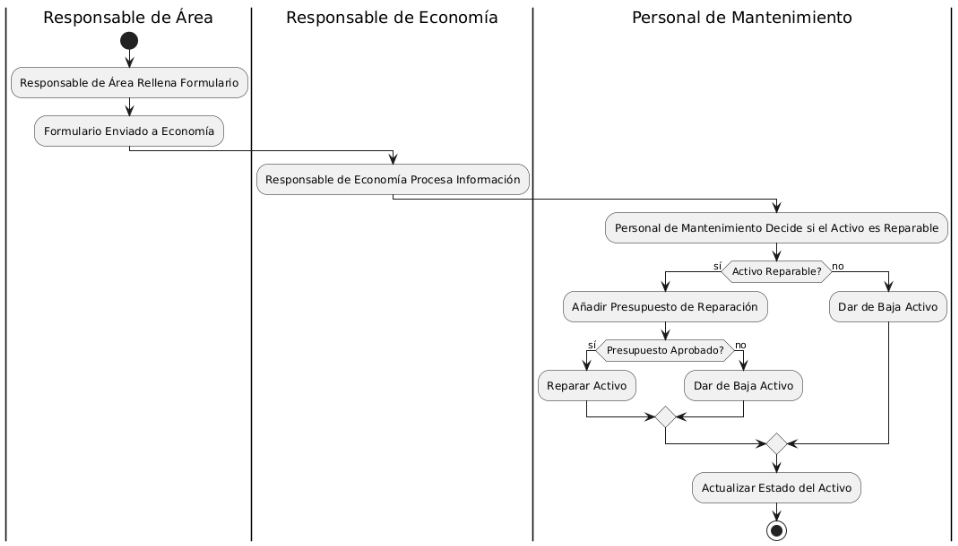
**Bajas Técnicas:**

Ilustración 4 bajas tecnicas

### Explicación:

Tabla 4 Explicación Bajas Técnicas

|  |  |
| --- | --- |
| Actor | Evento |
| Responsable de Área | Rellena el formulario de bajas técnicas. |
| Responsable de Área | Envía el formulario de bajas tecinas |
| Responsable de Economía | Procesa la información del formulario |
| Personal de Mantenimiento | Evalúa si el activo es reparable o no |
| Personal de Mantenimiento | ****Si el activo es reparable s****e añade el presupuesto de reparación |
| Personal de Mantenimiento | Si el activo no es reparable se da de baja |
| Personal de Mantenimiento | Actualización del Estado del Activo |

## 2.5. Modelo de Objetos del Negocio

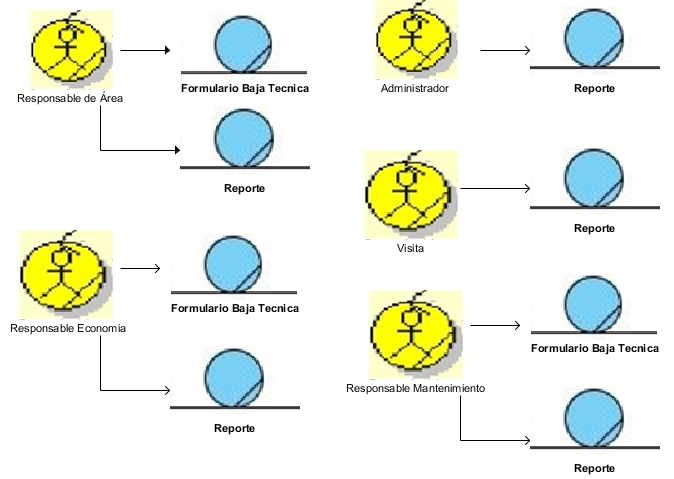
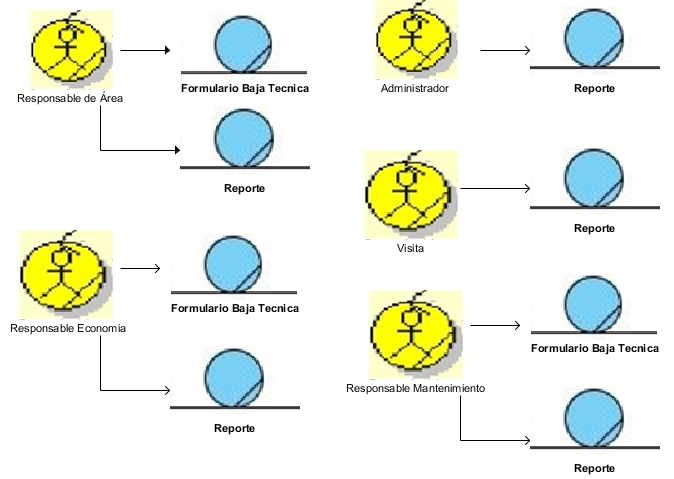
El modelo de objetos del negocio define las entidades utilizadas en el sistema, como los activos, los reportes, los formularios de bajas técnicas, los usuarios y sus roles. Este modelo es fundamental para entender la estructura y las relaciones entre los diferentes componentes del sistema.

Ilustración 5 Modelo de negocio: Responsable de mantenimiento

Ilustración 6 Modelo de negocio: Responsable de Área

### Explicación

1. **Usuario**: Contiene información sobre los usuarios del sistema, como nombre, contraseña, tipo y correo.
2. **Reporte**: Contiene información sobre los reportes generados, como tipo, fecha y filtros aplicados.
3. **Activo**: Contiene información sobre los activos fijos, como código, descripción, valor inicial, tasa de depreciación, estado, fecha de alta, centro de costos, clasificación y número de activo.
4. **Formulario**: Contiene información sobre los formularios de bajas técnicas, como fecha, detalles y estado.
5. **Presupuesto**: Contiene información sobre los presupuestos para la reparación de activos, como monto y estado de aprobación.
6. **Traza**: Contiene información sobre las acciones realizadas por los usuarios, como acción, fecha y usuario que realizó la acción.
7. **Centro de Costos**: Contiene información sobre los centros de costos, como código, descripción y nombre.
8. **Almacén**: Contiene información sobre los almacenes, como nombre, descripción, categoría y subcategoría.
9. **Producto**: Contiene información sobre los productos almacenados, como código, descripción, unidad de medida, precio, existencia actual y valor.
10. **Área**: Contiene información sobre las áreas de trabajo, como descripción y nombre.
11. **Trabajador**: Contiene información sobre los trabajadores, como nombre, unidad organizativa, provincia, municipio, cargo, nivel escolar, profesión, categoría ocupacional, tipo de contrato, categoría docente, criterio docente, sexo, estado (activo o baja) , centro de costos y otros.
12. **Usuario-Almacén**: Representa la relación entre usuarios y almacenes.
13. **Usuario-Área**: Representa la relación entre usuarios y áreas.

## 2.6. Reglas del Negocio a Considerar

Las reglas del negocio a considerar incluyen:

* **Acceso basado en roles**: Los usuarios solo pueden acceder a las funcionalidades y reportes pertinentes a su rol.
* **Validación de datos**: Los formularios deben validar la información ingresada para asegurar su precisión y completitud.
* **Trazabilidad**: Todas las acciones realizadas en el sistema deben ser registradas para facilitar el seguimiento y la auditoría.
* **Automatización de procesos**: Los procesos de bajas técnicas y generación de reportes deben ser automatizados para reducir errores y mejorar la eficiencia.

## 2.7. Conclusiones

En este capítulo se ha descrito la metodología de desarrollo RUP aplicada al nuevo sistema, identificando los actores y trabajadores del negocio, los casos de uso y actividades, el modelo de objetos del negocio y las reglas del negocio a considerar. Estos elementos son fundamentales para el diseño e implementación del sistema, asegurando que cumpla con los requisitos y necesidades de la Universidad Agraria de La Habana.

# CAPÍTULO 3: Definición de Requisitos del Sistema

## 3.1. Introducción

En este capítulo se abordarán los aspectos relacionados con la definición de requisitos funcionales y no funcionales del sistema a desarrollar. Se presentará un listado detallado de los requisitos, una descripción de los actores del sistema, el diagrama de casos de uso y una descripción literal de los casos de uso del sistema.

## 3.2. Definición de Requisitos Funcionales y No Funcionales

### 3.2.1. Listado de Requisitos Funcionales del Sistema a Desarrollar

Tabla 5 Requisitos Funcionales del Sistema

|  |  |  |  |
| --- | --- | --- | --- |
| **CÓDIGO** | **TIPO REQUISITO** | **NOMBRE DEL REQUISITO FUNCIONAL** | **DESCRIPCIÓN DEL REQUISITO FUNCIONAL** |
| RF-1 | Requisito Funcional | RF-1: Autenticar Usuario | El sistema debe permitir a los usuarios autenticarse mediante un formulario de login. |
| RF-2 | Requisito Funcional | RF-2: Generación de Reportes | El sistema debe permitir a los usuarios generar reportes financieros con filtros personalizados. |
| RF-3 | Requisito Funcional | RF-3: Gestión de Bajas Técnicas  3.1: Exportar Informe  3.2: Modificar estado de activo | El sistema debe permitir a los responsables de área rellenar formularios para el proceso de bajas técnicas.  3.1: El sistema debe permitir enviar un informe  3.2: El sistema debe permitir modificar los estados de los activos |
| RF-4 | Requisito Funcional | RF-4: Administración de Usuarios  4.1: Adicionar Usuario  4.2: Eliminar Usuario  4.3: Modificar Usuario | El sistema debe permitir al administrador gestionar los usuarios del sistema.  4.1: El sistema debe permitir al administrador adicionar nuevos usuarios  4.2: El sistema debe permitir al administrador eliminar los usuarios  4.3: El sistema debe permitir al administrador modificar a los usuarios |
| RF-5 | Requisito Funcional | RF-5: Trazabilidad de Acciones | El sistema debe registrar todas las acciones realizadas por los usuarios para facilitar la auditoría. |
| RF-6 | Requisito Funcional | RF-6: Diseño y Creación de Filtros | * El sistema debe permitir el diseño y creación de filtros personalizados utilizando los valores disponibles en la base de datos. |
| RF-7 | Requisito Funcional | RF-7: Uso de Filtros en Reportes | El sistema debe permitir a los usuarios aplicar filtros personalizados a los reportes generados. |
| RF-8 | Requisito Funcional | RF-8: Reporte de Activos Fijos por Clasificación  8.1: Generar reporte  3.2: Guardar reporte  8.3: Imprimir reporte | 8.1: El sistema debe permitir la generación de reportes de activos fijos por clasificación, con filtros personalizados, y la opción de guardar y descargar los reportes.  8.2: El sistema debe permitir guardar los reportes generados  8.3: El sistema debe permitir imprimir los reportes generados |
| RF-9 | Requisito Funcional | RF-9: Reporte de Útiles y Herramientas  9.1: Generar reporte  9.2: Guardar reporte  9.3: Imprimir reporte | 9.1: El sistema debe permitir la generación de reportes de útiles y herramientas, con filtros personalizados, y la opción de guardar y descargar los reportes.  9.2: El sistema debe permitir guardar los reportes generados  9.3: El sistema debe permitir imprimir los reportes generados |
| RF-10 | Requisito Funcional | RF-10: Reporte de Productos en Almacenes  10.1: Generar reporte  10.2: Guardar reporte  10.3: Imprimir reporte | 10.1: El sistema debe permitir la generación de reportes de productos en diferentes tipos de almacenes, con filtros personalizados, y la opción de guardar y descargar los reportes.  10.2: El sistema debe permitir guardar los reportes generados  10.3: El sistema debe permitir imprimir los reportes generados |
| RF-11 | Requisito Funcional | RF-11: Reporte de Listado de Centro de Costos  11.1: Generar reporte  11.2: Guardar reporte  11.3: Imprimir reporte | 11.1: El sistema debe permitir la generación de reportes de listado de centro de costos, con filtros personalizados, y la opción de guardar y descargar los reportes.  11.2: El sistema debe permitir guardar los reportes generados  11.3: El sistema debe permitir imprimir los reportes generados |
| RF-12 | Requisito Funcional | RF-12: Reporte de Listado de Activos por Número de Activos  12.1: Generar reporte  12.2: Guardar reporte  12.3: Imprimir reporte | 12.1: El sistema debe permitir la generación de reportes de listado de activos por número de activos, con filtros personalizados, y la opción de guardar y descargar los reportes.  12.2: El sistema debe permitir guardar los reportes generados  12.3: El sistema debe permitir imprimir los reportes generados |
| RF-13 | Requisito Funcional | RF-13: Reporte de Activos Fijos  13.1: Generar reporte  13.2: Guardar reporte  13.3: Imprimir reporte | 13.1: El sistema debe permitir la generación de reportes de activos fijos, con filtros personalizados, y la opción de guardar y descargar los reportes.  13.2: El sistema debe permitir guardar los reportes generados  13.3: El sistema debe permitir imprimir los reportes generados |
| RF-14 | Requisito Funcional | RF-14: Reporte de Repuestos de Activos Fijos por Clasificación  14.1: Generar reporte  14.2: Guardar reporte  14.3: Imprimir reporte | 14.1: El sistema debe permitir la generación de reportes de repuestos de activos fijos por clasificación, con filtros personalizados, y la opción de guardar y descargar los reportes.  14.2: El sistema debe permitir guardar los reportes generados  14.3: El sistema debe permitir imprimir los reportes generados |

### 3.2.2. Listado de Requisitos No Funcionales del Sistema a Desarrollar

Tabla 6 Requisitos No Funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| **CÓDIGO** | **TIPO REQUISITO** | **CLASIFICACIÓN RQNF** | **DESCRIPCIÓN DEL REQUISITO NF** |
| RNF-1 | Requisito No Funcional | RNF-1: Seguridad | El sistema debe garantizar la seguridad de los datos mediante autenticación y autorización adecuadas. |
| RNF-2 | Requisito No Funcional | RNF-2: Rendimiento | El sistema debe ser capaz de generar reportes en menos de 5 segundos. |
| RNF-3 | Requisito No Funcional | RNF-3: Usabilidad | La interfaz de usuario debe ser intuitiva y fácil de usar para reducir la curva de aprendizaje. |
| RNF-4 | Requisito No Funcional | RNF-3: Mantenibilidad | El sistema debe ser fácil de mantener y actualizar. |
| RNF-5 | Requisito No Funcional | RNF-4: Disponibilidad | El sistema debe estar disponible al menos el 99.9% del tiempo. |

## 3.3. Descripción de los Actores del Sistema

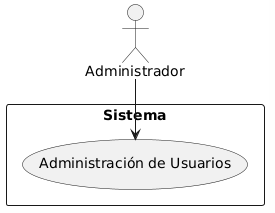
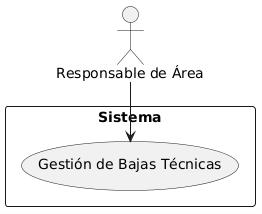
En la tabla se muestra la descripción de los actores del sistema.

Tabla 7 Actores del Sistema

|  |  |
| --- | --- |
| Actores del Sistema | Descripción |
| Usuario | Es el rol encargado de autenticarse en el sistema, dependiendo de su rol accede a las funcionalidades previstas para el mismo. |
| Administrador | Es el rol encargado de gestionar todos los usuarios que estén vinculados al sistema tanto, así como los reportes que se puedan generar. |
| Responsable de Área | Es el rol encargado de la generación de reportes específicos para su área de responsabilidad y de solicitar el proceso de bajas técnicas |
| Responsable de Economía | Es el rol encargado de recibir y procesar la información de los reportes y tomar decisiones sobre la reparación y baja de activos. |
| Responsable de Mantenimiento | Es el rol encargado de recibir información sobre los activos, evaluarlos y analizar el estado de los mismos. |

## 3.4. Diagrama de Casos de Uso del Sistema \*\*\*\*\*\*\*\*\*\*\*\*

En el diagrama de caso de uso del sistema se representan los roles que intervienen en el sistema y cada función que desempeña.

## 3.5. Descripción de los Casos de Uso del Sistema

En la descripción de los casos de uso se explican las distintas operaciones que se efectúan, con el objetivo de que el usuario pueda interactuar fácilmente con el sistema.

Tabla 8 Descripción de los Casos de Uso del Sistema: Autenticar Usuario

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF01 | Autenticar Usuario | | | 29/08/2024 | | Alta |
| **DESCRIPCION RQF:** El sistema debe permitir a los usuarios autenticarse mediante un formulario de login | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Credenciales del usuario (nombre de usuario y contraseña) | | Usuario | Acceso al sistema | | Sistema | Validación de credenciales contra la base de datos de usuarios |
| **PROCESO:** (pasos) | 1. El usuario ingresa sus credenciales en el formulario de login. 2. El sistema valida las credenciales. 3. Si las credenciales son correctas, el usuario accede al sistema. 4. Si las credenciales son incorrectas, se muestra un mensaje de error. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible bloqueo de cuenta tras múltiples intentos fallidos. | | | | | |

Tabla 9 Descripción de los Casos de Uso del Sistema: Generación de Reportes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF02 | Generación de Reportes | | | 29/08/2024 | | Alta |
| **DESCRIPCION RQF:** El sistema debe permitir a los usuarios generar reportes financieros con filtros personalizados. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | Reporte financiero | | Usuario | Los reportes deben cumplir con los estándares financieros establecidos. |
| **PROCESO:** (pasos) | 1. El usuario selecciona los parámetros de filtro. 2. El sistema genera el reporte basado en los filtros seleccionados. 3. El usuario puede visualizar, guardar o descargar el reporte. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible sobrecarga del sistema si se generan múltiples reportes simultáneamente. | | | | | |

Tabla 10 Descripción de los Casos de Uso del Sistema: Gestión de Bajas Técnicas

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF03 | Gestión de Bajas Técnicas | | | 29/08/2024 | | Media |
| **DESCRIPCION RQF:** El sistema debe permitir a los responsables de área rellenar formularios para el proceso de bajas técnicas. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Información de baja técnica | | Responsable de área | Formulario de baja técnica | | Sistema | Las bajas técnicas deben ser aprobadas por un supervisor. |
| **PROCESO:** (pasos) | 1. El responsable de área rellena el formulario de baja técnica. 2. El sistema guarda el formulario y notifica al supervisor. 3. El supervisor revisa y aprueba o rechaza la baja técnica. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible retraso en el proceso si el supervisor no revisa las bajas a tiempo. | | | | | |

Tabla 11 Descripción de los Casos de Uso del Sistema: Administracion de Usuarios

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF04 | Administración de Usuarios | | | 29/08/2024 | | Alta |
| **DESCRIPCION RQF:** El sistema debe permitir al administrador gestionar los usuarios del sistema. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Información del usuario | | Administrador | Actualización de la base de datos de usuarios | | Sistema | Solo los administradores pueden gestionar usuarios. |
| **PROCESO:** (pasos) | 1. El administrador accede a la sección de gestión de usuarios. 2. El administrador añade, edita o elimina usuarios según sea necesario. 3. El sistema actualiza la base de datos de usuarios. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible eliminación accidental de usuarios importantes. | | | | | |

Tabla 12 Descripción de los Casos de Uso del Sistema: Trazabilidad de Acciones

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF05 | Trazabilidad de Acciones | | | 29/08/2024 | | Alta |
| **DESCRIPCION RQF:** El sistema debe registrar todas las acciones realizadas por los usuarios para facilitar la auditoría. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Acciones de los usuarios | | Sistema | Registro de acciones | | Base de datos de auditoría | Todas las acciones deben ser registradas con detalles como fecha, hora y usuario. |
| **PROCESO:** (pasos) | 1. El usuario realiza una acción en el sistema. 2. El sistema registra la acción en la base de datos de auditoría. 3. Los registros pueden ser consultados por los auditores. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible aumento en el tamaño de la base de datos debido a la gran cantidad de registros. | | | | | |

Tabla 13Descripción de los Casos de Uso del Sistema: Diseño y Creación de Filtros

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF06 | Diseño y Creación de Filtros | | | 29/08/2024 | | Media |
| **DESCRIPCION RQF:** El sistema debe permitir el diseño y creación de filtros personalizados para todos los reportes. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | Filtros personalizados | | Sistema | Los filtros deben ser compatibles con los datos disponibles en el sistema. |
| **PROCESO:** (pasos) | 1. El usuario accede a la sección de creación de filtros. 2. El usuario define los parámetros del filtro. 3. El sistema guarda el filtro personalizado. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible complejidad en la gestión de múltiples filtros personalizados. | | | | | |

Tabla 14Descripción de los Casos de Uso del Sistema: Uso de Filtros en Reportes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF07 | Uso de Filtros en Reportes | | | 29/08/2024 | | Media |
| **DESCRIPCION RQF:** El sistema debe permitir a los usuarios aplicar filtros personalizados a los reportes generados. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Filtros personalizados | | Usuario | Reportes filtrados | | Usuario | Los reportes deben reflejar los datos filtrados correctamente. |
| **PROCESO:** (pasos) | 1. El usuario selecciona un filtro personalizado. 2. El sistema aplica el filtro al reporte. 3. El usuario visualiza el reporte filtrado. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible error en la aplicación de filtros si no están bien definidos. | | | | | |

Tabla 15Descripción de los Casos de Uso del Sistema: Reporte de Activos Fijos por Clasificación

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF08 | Reporte de Activos Fijos por Clasificación | | | 29/08/2024 | | Alta |
| **DESCRIPCION RQF:** El sistema debe permitir la generación de reportes de activos fijos por clasificación, con filtros personalizados, y la opción de guardar y descargar los reportes. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | Reporte de activos fijos | | Usuario | Los reportes deben cumplir con los estándares de clasificación de activos. |
| **PROCESO:** (pasos) | 1. El usuario selecciona los parámetros de filtro. 2. El sistema genera el reporte basado en los filtros seleccionados. 3. El usuario puede visualizar, guardar o descargar el reporte. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible sobrecarga del sistema si se generan múltiples reportes simultáneamente. | | | | | |

Tabla 16Descripción de los Casos de Uso del Sistema: Reporte de Útiles y Herramientas

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF09 | Reporte de Útiles y Herramientas | | | 29/08/2024 | | Media |
| **DESCRIPCION RQF:** El sistema debe permitir la generación de reportes de útiles y herramientas, con filtros personalizados, y la opción de guardar y descargar los reportes. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | Reporte de útiles y herramientas | | Usuario | Los reportes deben cumplir con los estándares de clasificación de útiles y herramientas. |
| **PROCESO:** (pasos) | 1. El usuario selecciona los parámetros de filtro. 2. El sistema genera el reporte basado en los filtros seleccionados. 3. El usuario puede visualizar, guardar o descargar el reporte. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible sobrecarga del sistema si se generan múltiples reportes simultáneamente. | | | | | |

Tabla 17Descripción de los Casos de Uso del Sistema: Reporte de Productos en Almacen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF10 | Reporte de Productos en Almacenes | | | 29/08/2024 | | Media |
| **DESCRIPCION RQF:** El sistema debe permitir la generación de reportes de productos en diferentes tipos de almacenes, con filtros personalizados, y la opción de guardar y descargar los reportes. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | Reporte de productos en almacenes | | Usuario | Los reportes deben cumplir con los estándares de clasificación de productos en almacenes. |
| **PROCESO:** (pasos) | 1. El usuario selecciona los parámetros de filtro. 2. El sistema genera el reporte basado en los filtros seleccionados. 3. El usuario puede visualizar, guardar o descargar el reporte. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible sobrecarga del sistema si se generan múltiples reportes simultáneamente. | | | | | |

Tabla 18Descripción de los Casos de Uso del Sistema: Reporte de Listado de Centro de Costos

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | | **FECHA** | | **PRIORIDAD** |
| RF11 | Reporte de Listado de Centro de Costos | | | | 29/08/2024 | | Media |
| **DESCRIPCION RQF:** El sistema debe permitir la generación de reportes de listado de centro de costos, con filtros personalizados, y la opción de guardar y descargar los reportes. | | | | | | | |
| **ENTRADA** | | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | | Reporte de listado de centro de costos | | Usuario | Los reportes deben cumplir con los estándares de clasificación de centros de costos. |
| **PROCESO:** (pasos) | 1. El usuario selecciona los parámetros de filtro. 2. El sistema genera el reporte basado en los filtros seleccionados. 3. El usuario puede visualizar, guardar o descargar el reporte. | | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible sobrecarga del sistema si se generan múltiples reportes | | | | | | |

Tabla 19Descripción de los Casos de Uso del Sistema: Reporte de Listado de Activos por Numero de Activos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF12 | Reporte de Listado de Activos por Número de Activos | | | 29/08/2024 | | Media |
| **DESCRIPCION RQF:** El sistema debe permitir la generación de reportes de listado de activos por número de activos, con filtros personalizados, y la opción de guardar y descargar los reportes. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | Reporte de listado de activos | | Usuario | Los reportes deben cumplir con los estándares de clasificación de activos. |
| **PROCESO:** (pasos) | 1. El usuario selecciona los parámetros de filtro. 2. El sistema genera el reporte basado en los filtros seleccionados. 3. El usuario puede visualizar, guardar o descargar el reporte. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible sobrecarga del sistema si se generan múltiples reportes simultáneamente. | | | | | |

Tabla 20Descripción de los Casos de Uso del Sistema: Reporte de Activos Fijos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF13 | Reporte de Activos Fijos | | | 29/08/2024 | | Alta |
| **DESCRIPCION RQF:** El sistema debe permitir la generación de reportes de activos fijos, con filtros personalizados, y la opción de guardar y descargar los reportes. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | Reporte de activos fijos | | Usuario | Los reportes deben cumplir con los estándares de clasificación de activos fijos. |
| **PROCESO:** (pasos) | 1. El usuario selecciona los parámetros de filtro. 2. El sistema genera el reporte basado en los filtros seleccionados. 3. El usuario puede visualizar, guardar o descargar el reporte. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible sobrecarga del sistema si se generan múltiples reportes simultáneamente. | | | | | |

Tabla 21Descripción de los Casos de Uso del Sistema: Reporte de Repuestos de Activos Fijos por Clasificación

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CODIGO** | **NOMBRE RQF** | | | **FECHA** | | **PRIORIDAD** |
| RF14 | Reporte de Repuestos de Activos Fijos por Clasificación | | | 29/08/2024 | | Media |
| **DESCRIPCION RQF:** El sistema debe permitir la generación de reportes de repuestos de activos fijos por clasificación, con filtros personalizados, y la opción de guardar y descargar los reportes. | | | | | | |
| **ENTRADA** | | **FUENTE** | **SALIDA** | | **DESTINO** | **REGLAS DEL NEGOCIO** |
| Parámetros de filtro | | Usuario | Reporte de repuestos de activos fijos | | Usuario | Los reportes deben cumplir con los estándares de clasificación de repuestos de activos fijos |
| **PROCESO:** (pasos) | 1. El usuario selecciona los parámetros de filtro. 2. El sistema genera el reporte basado en los filtros seleccionados. 3. El usuario puede visualizar, guardar o descargar el reporte. | | | | | |
| **EFECTO COLATERAL** (Adverso) | Posible sobrecarga del sistema si se generan múltiples reportes simultáneamente. | | | | | |

## 3.6. Conclusiones

En este capítulo se ha definido de manera detallada los requisitos funcionales y no funcionales del sistema a desarrollar, incluyendo la generación y uso de filtros para todos los reportes mencionados, así como la funcionalidad de guardar y descargar los reportes. Se han identificado y descrito los actores del sistema, así como los casos de uso que representan las principales funcionalidades del sistema. Además, se ha proporcionado una descripción literal de los casos de uso, incluyendo los requisitos funcionales y no funcionales asociados.

Las conclusiones parciales de este capítulo son las siguientes:

**Identificación Clara de Requisitos**: Se ha logrado identificar y clasificar de manera clara los requisitos funcionales y no funcionales del sistema, lo cual es fundamental para el diseño e implementación del mismo.

* **Descripción Detallada de Actores y Casos de Uso**: La descripción detallada de los actores y casos de uso permite entender las interacciones y responsabilidades de cada usuario dentro del sistema.
* **Prototipo de Interfaz de Usuario**: La inclusión de prototipos de interfaz de usuario en la descripción de los casos de uso facilita la visualización de cómo los usuarios interactuarán con el sistema.
* **Requisitos No Funcionales**: La identificación de requisitos no funcionales asegura que el sistema no solo cumpla con las funcionalidades esperadas, sino que también sea seguro, eficiente, usable, mantenible y disponible.
* **Filtros Personalizados y Descarga de Reportes**: La inclusión de filtros personalizados y la opción de guardar y descargar los reportes mejora la precisión y relevancia de la información generada, adaptándose a las necesidades específicas de los usuarios.

# CAPÍTULO 4: Diseño e Implementación del Sistema

## 4.1. Introducción

En este capítulo se abordarán los aspectos relacionados con la solución propuesta para mejorar la generación de reportes financieros y la gestión de bajas técnicas en la Universidad Agraria de La Habana “Fructuoso Rodríguez Pérez”. Se describirá la solución propuesta, el diseño del sistema, el modelo de base de datos y la validación del sistema mediante pruebas de caja negra y blanca.

## 4.2. Descripción de la Solución Propuesta

La solución propuesta consiste en el desarrollo de un nuevo sistema basado en el SigeWeb, mejorándolo y añadiendo nuevas funcionalidades. El sistema permitirá la generación de reportes financieros con filtros personalizados, la gestión automatizada de bajas técnicas de activos, y la administración de usuarios y trazabilidad de acciones. Además, se implementará una nueva interfaz de usuario para mejorar la experiencia y reducir la curva de aprendizaje.

## 4.3. Diseño del Sistema

El diseño del sistema se basa en la metodología RUP y se representa mediante diagramas UML. A continuación, se presentan los diagramas de casos de uso, actividades, clases, secuencia y componentes.

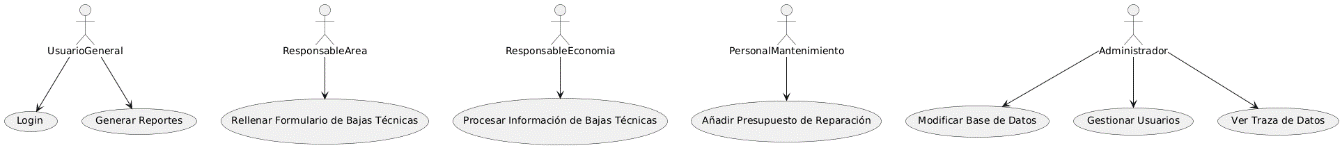
**Diagrama de Casos de Uso**

Ilustración 7Diagrama de Casos de Uso

**“Generación de Reportes”**

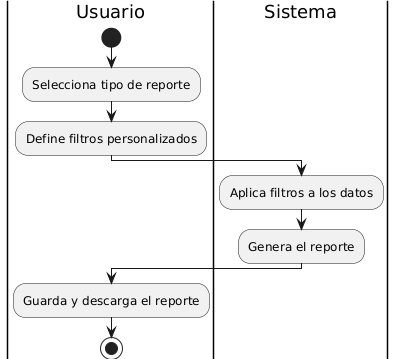


Ilustración 8 “Generación de Reportes”

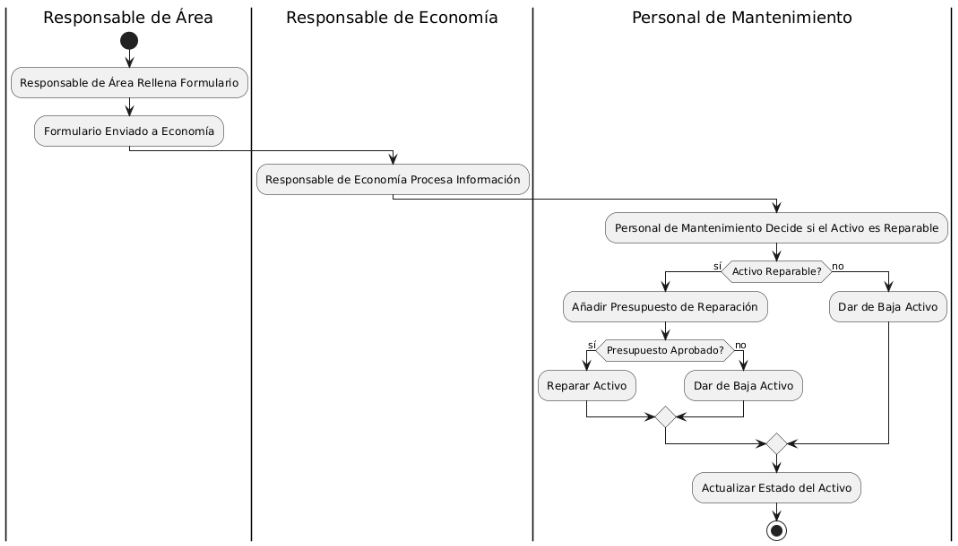
** “Gestión de Bajas Técnicas”**

Ilustración 9 “Gestión de Bajas Técnicas”

**Diagrama de Clases**

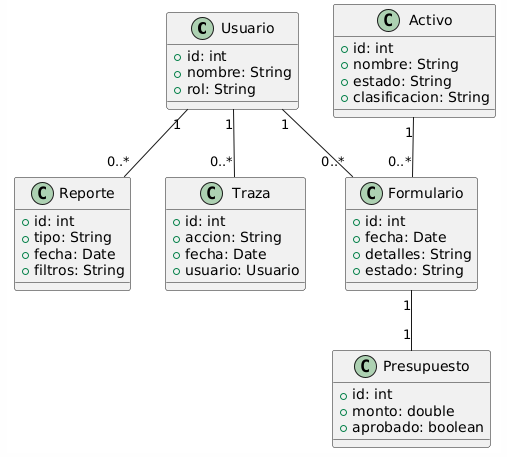


Ilustración 10 Diagrama de Clases

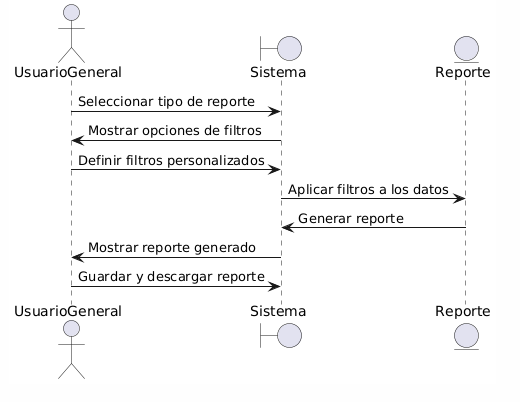
**Diagrama de Secuencia para “Generación de Reportes”**

Ilustración 11 Diagrama de Secuencia para “Generación de Reportes”

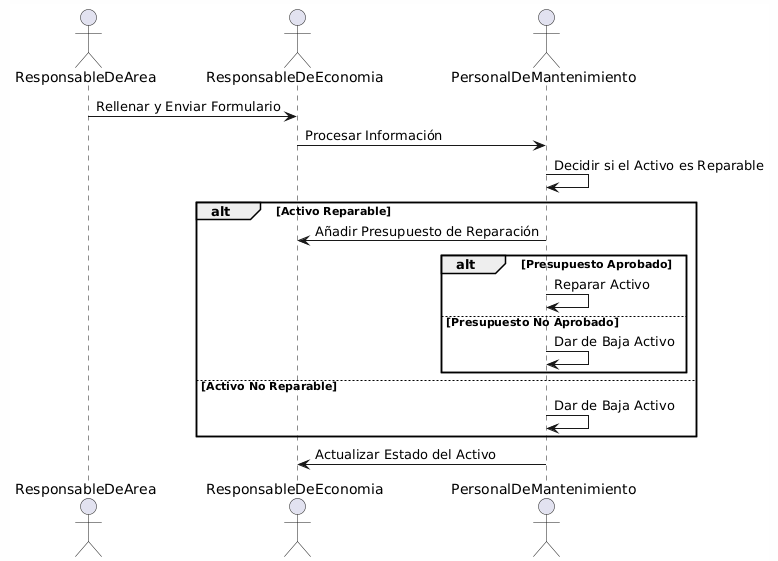
**Diagrama de Secuencia para “Gestión de Bajas Técnicas”**

Ilustración 12 Diagrama de Secuencia para “Gestión de Bajas Técnicas”

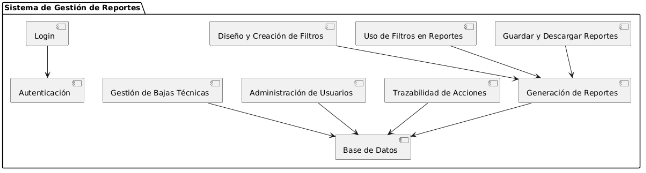
**Diagrama de Componente**

Ilustración 13 Diagrama de Componentes

## 4.4. Modelo de Base de Datos

### 4.4.1 Modelo lógico de la Base de Datos

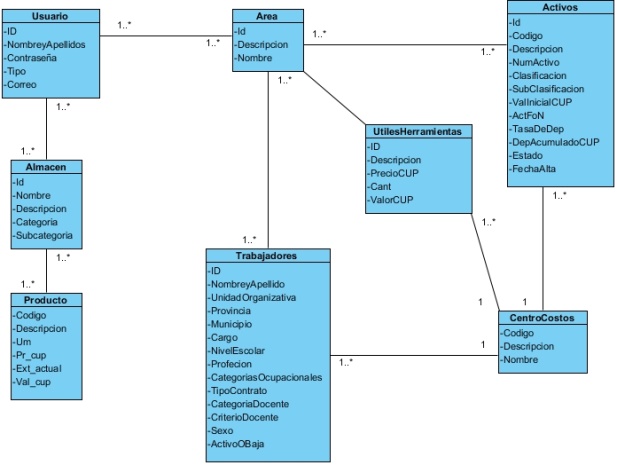
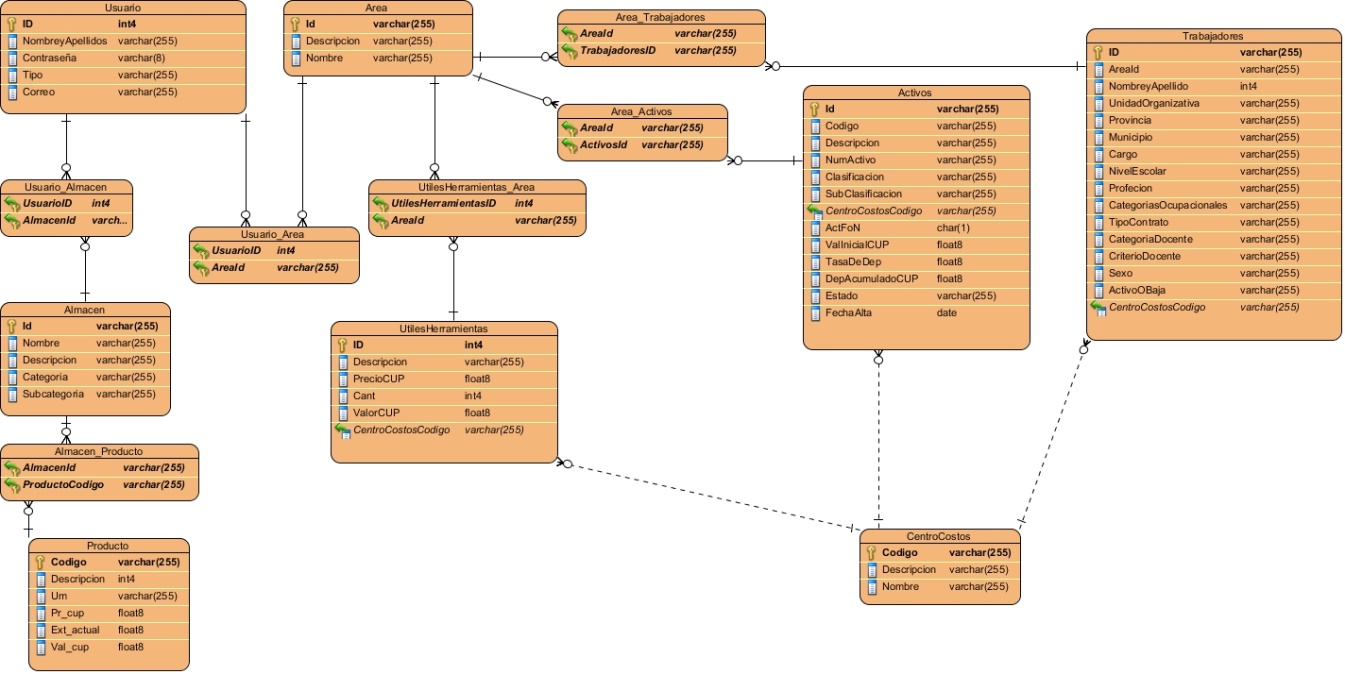


Ilustración 14 Modelo lógico de la Base de Datos

### 

### 4.4.2 Modelo físico de la Base de Datos



## 4.5. Validación del Sistema

### 4.5.1. Pruebas de Caja Negra

La prueba de caja negra consiste en probar un sistema informático sin tener conocimiento previo de su funcionamiento interno.

#### 4.5.1.1. Prueba de Autenticar Usuario

Tabla 22 Prueba de Autenticar Usuario

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del requisito** | **Descripción general** | **Escenarios de pruebas** | **Flujo del escenario** |
| RF-01: Autenticar Usuario | Verificar que los usuarios pueden autenticarse correctamente. | EP 1. Login exitoso | - El Usuario ingresa credenciales correctas y es redirigido a su página principal. |
| EP 2. Login fallido por credenciales incorrectas. | - EL Usuario ingresa credenciales incorrectas y recibe un mensaje de error. |
| 3. Login fallido por usuario inexistente | - EL usuario ingresa un nombre de usuario inexistente y recibe un mensaje de error. |

#### 4.5.1.2 Prueba de Generación de Reportes

Tabla 23 Prueba de Generación de Reportes

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del requisito** | **Descripción general** | **Escenarios de pruebas** | **Flujo del escenario** |
| RF-02: Generación de Reportes | Verificar que los usuarios pueden generar reportes con filtros personalizados. | EP 1. Generación de reporte con filtros correctos. | - Usuario selecciona tipo de reporte, define filtros y genera el reporte. |
| EP 2. Generación de reporte sin filtros. | - Usuario selecciona tipo de reporte sin definir filtros y genera el reporte. |
| 3. Generación de reporte con filtros incorrectos | - . Usuario selecciona tipo de reporte, define filtros incorrectos y genera el reporte. |

#### 4.5.1.3 Prueba de Gestión de Bajas Técnicas

Tabla 24 Prueba de Gestión de Bajas Técnicas

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del requisito** | **Descripción general** | **Escenarios de pruebas** | **Flujo del escenario** |
| RF-3: Gestión de Bajas Técnicas | Verificar que los responsables de área pueden rellenar formularios de bajas técnicas. | EP 1. Rellenar formulario correctamente. | - Responsable de área rellena formulario con datos correctos y lo envía. |
| EP 2. Rellenar formulario con datos incompletos. | - Responsable de área rellena formulario con datos incompletos y recibe un mensaje de error. |
| 3. Rellenar formulario con datos incorrectos | - . Responsable de área rellena formulario con datos incorrectos y recibe un mensaje de error. |

#### 4.5.1.4 Prueba de Administración de Usuarios

Tabla 25 Prueba de Administración de Usuarios

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del requisito** | **Descripción general** | **Escenarios de pruebas** | **Flujo del escenario** |
| RF-4: Administración de Usuarios | Verificar que el administrador puede gestionar los usuarios del sistema. | EP 1. Crear usuario. | - Administrador ingresa datos del nuevo usuario y lo crea. |
| EP 2. Modificar usuario. | - Administrador modifica datos de un usuario existente. |
| 3. Eliminar usuario | - Administrador elimina un usuario existente. |

#### 4.5.1.5 Prueba de Trazabilidad de Acciones

Tabla 26 Prueba de Trazabilidad de Acciones

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del requisito** | **Descripción general** | **Escenarios de pruebas** | **Flujo del escenario** |
| RF-5: Trazabilidad de Acciones | Verificar que el sistema registra todas las acciones realizadas por los usuarios. | EP 1. Registro de acción exitosa. | - Usuario realiza una acción y el sistema la registra correctamente. |
| EP 2. Registro de acción fallida | - Usuario realiza una acción y el sistema no la registra debido a un error. |

**Número de Pruebas Realizadas**: Se realizaron un total de 20 pruebas de caja negra para evaluar la funcionalidad del sistema.

**Errores Encontrados**:

1. Login de Usuario: 3 errores

- Error: Fallo en la autenticación de usuarios con credenciales correctas.

- Solución Implementada: Corrección en la validación de credenciales.

- Estado: Implementado.

2. Generación de Reportes: 5 errores

- Error: Reportes generados con datos incorrectos debido a filtros mal aplicados.

- Solución Implementada: Mejora en la lógica de aplicación de filtros.

- Estado: Implementado.

3. Gestión de Bajas Técnicas: 4 errores

- Error: Formularios de bajas técnicas no se enviaban correctamente.

- Solución Implementada: Corrección en el envío y procesamiento de formularios.

- Estado: En proceso.

4. Administración de Usuarios: 3 errores

- Error: Problemas al añadir y eliminar usuarios.

- Solución Implementada: Ajustes en la gestión de la base de datos de usuarios.

- Estado: En proceso.

5. Trazabilidad de Acciones: 2 errores

- Error: Acciones de usuarios no se registraban correctamente.

- Solución Implementada: Mejora en el registro de acciones.

- Estado: En proceso.

6. Compatibilidad con Nuevas Tecnologías: 3 errores

- Error: Incompatibilidad con dispositivos móviles y nuevas integraciones.

- Solución en Proceso: Adaptación del sistema para compatibilidad móvil y soporte de integraciones modernas.

- Estado: En proceso.

### 4.5.2. Prueba de caja blanca

La prueba de caja blanca es una técnica de prueba de software que consiste en probar la estructura interna y el diseño de un sistema informático.

La prueba del camino básico es una técnica de prueba de caja blanca basada en la estructura de control de un programa o módulo. Dado que esta prueba se basa en la estructura de control del programa, requiere un conocimiento completo de la estructura del programa. Para diseñar casos de prueba utilizando esta técnica, se siguen cuatro pasos:

1. Dibuja un grafo asociado al flujo.
2. Calcular la complejidad ciclomática del grafo.
3. Determinar un conjunto básico de caminos independientes.
4. Preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

**Grafo de flujo:** Está formado por 3 componentes fundamentales que ayudan a su elaboración, comprensión y brinda información para confirmar que el trabajo se está haciendo adecuadamente.

* Nodos (Nodos predicados): Está caracterizado porque dos o más aristas emergen de él.
* Aristas: Las flechas del grafo.
* Regiones: Áreas delimitadas por las aristas y nodos (el área fuera del grafo también se cuenta como una región).

**Complejidad ciclomática:** se dice que la complejidad ciclomática V (G) es una medida de la complejidad lógica de un programa. Se puede calcular utilizando tres fórmulas diferentes:

1. **V(G) = R**
2. **V(G) = A-N+2**
3. **V(G) = P+1**

Conociendo que:

G: Grafo de flujo (grafo)

R: El número de regiones contribuye a estimar el valor de la complejidad ciclomática.

A: Número de aristas.

V (G): Complejidad ciclomática.

N: Número de nodos del grafo.

P: Número de nodos predicados incluidos en el grafo.

**Camino Independiente:** Es cualquier camino del programa que introduce por lo menos un nuevo conjunto de sentencias de procesamiento o una nueva condición. El camino independiente se debe mover por lo menos por una arista que no haya sido recorrida anteriormente.

#### 4.5.2.1: Inicio de Sesión

**Código:**

// Nodo A: Inicio del programa

<%@ page import="java.sql. \*" %>

<%@ page import="jakarta.servlet.http.HttpSession" %>

<%

// Nodo B: Obtener parámetros

String usuario = request.getParameter("usuario");

String password = request.getParameter("password");

// Nodo C: Inicializar variables

Connection con = null;

PreparedStatement ps = null;

ResultSet rs = null;

boolean loginExitoso = false;

int userId = -1;

String tipo = "";

try {

// Nodo D: Cargar driver y establecer conexión

Class.forName("org.postgresql.Driver");

con = DriverManager.getConnection("jdbc:postgresql://localhost:5432/SigeWeb2", "postgres", "\*\*\*\*\*\*\*");

// Nodo E: Preparar y ejecutar consulta

String query = "SELECT id, tipo FROM usuario WHERE nombreyapellidos = ? AND contraseña = ?";

ps = con.prepareStatement(query);

ps.setString(1, usuario);

ps.setString(2, password);

rs = ps.executeQuery();

// Nodo F: Verificar resultado

if (rs.next()) {

loginExitoso = true;

userId = rs.getInt("id");

tipo = rs.getString("tipo");

// Nodo G: Crear sesión y asignar atributos

HttpSession httpSession = request.getSession();

httpSession.setAttribute("logeado", "1");

httpSession.setAttribute("userId", userId);

httpSession.setAttribute("tipo", tipo);

}

} catch (Exception e) {

// Nodo H: Manejo de excepciones

e.printStackTrace();

} finally {

// Nodo I: Cerrar recursos

if (rs != null) try { rs.close(); } catch (SQLException e) { e.printStackTrace(); }

if (ps != null) try { ps.close(); } catch (SQLException e) { e.printStackTrace(); }

if (con != null) try { con.close(); } catch (SQLException e) { e.printStackTrace(); }

}

// Nodo J: Verificar login exitoso

if (loginExitoso) {

%>

<script>

// Nodo K: Mostrar mensaje de éxito y redirigir a index.jsp

alert("Login exitoso. ¡Bienvenido, <%= usuario %>!");

window.location.href = "index.jsp";

</script>

<%

} else {

%>

<script>

// Nodo L: Mostrar mensaje de error y redirigir a login.html

alert("Usuario o contraseña incorrectos.");

window.location.href = "login.html";

</script>

<%

}

%>

**GRAFO ASOCIADO:**

Descripción de los Nodos:

* **A**: Inicio del programa.
* **B**: Obtener parámetros usuario y password.
* **C**: Inicializar variables (con, ps, rs, loginExitoso, userId, tipo).
* **D**: Cargar driver y establecer conexión.
* **E**: Preparar y ejecutar consulta.
* **F**: Verificar resultado.
* **G**: Crear sesión y asignar atributos.
* **H**: Manejo de excepciones.
* **I**: Cerrar recursos.
* **J**: Verificar login exitoso.
* **K**: Mostrar mensaje de éxito y redirigir a index.jsp.
* **L**: Mostrar mensaje de error y redirigir a login.html.

**COMPLEJIDAD CICLOMÁTICA:**

Número de Nodos (N): 12  
Número de Aristas (A): 14

Aplicando la fórmula V(G) = A - N + 2:

V(G) = 14 - 12 + 2 = 4V(G)=14−12+2=4

Por lo tanto, el índice de complejidad ciclomática del grafo es V(G) = 4. Esto indica que hay 4 caminos lineales independientes a través del código.

**CAMINOS INDEPENDIENTES:**

1. **Inicio de sesión exitoso**:
   * Caminos: A -> B -> C -> D -> E -> F -> G -> J -> K
2. **Inicio de sesión fallido**:
   * Caminos: A -> B -> C -> D -> E -> F -> J -> L
3. **Excepción en la conexión**:
   * Caminos: A -> B -> C -> D -> H -> I -> J -> L
4. **Excepción en la consulta**:
   * Caminos: A -> B -> C -> D -> E -> H -> I -> J -> L

**CASOS DE PRUEBA:**

1. **Camino 1**:
   * Credenciales correctas.
   * Usuario y contraseña válidos.
2. **Camino 2**:
   * Credenciales incorrectas.
   * Usuario o contraseña inválidos.
3. **Camino 3**:
   * Error en la conexión a la base de datos.
   * Problema con el driver o la URL de conexión.
4. **Camino 4**:
   * Error en la ejecución de la consulta.
   * Problema con la consulta SQL o los parámetros.

#### 4.5.2.2. GENERAR REPORTE DE ACTIVOS FIJOS POR CLASIFICACIÓN

**Código:**

// Nodo A: Inicio de la función

function generarReporteActivosFijosPorClasificacion() {

// Nodo B: Obtener ID del usuario

const idUsuario = document.getElementById('userId').value; // Obtén el ID del usuario desde la sesión

// Nodo C: Obtener clasificación

const clasificacion = document.getElementById('listaClasificaciones');

const valorClasificacion = encodeURIComponent(clasificacion.value);

// Nodo D: Obtener subclasificación

const subclasificacion = document.getElementById('listaSubClasificaciones');

const valorSubCla = encodeURIComponent(subclasificacion.value);

// Nodo E: Construir URL

const url = `http://localhost:8211/GenerarReportedeactivosfijosClasif?clasificacion=${valorClasificacion}&subclasificacion=${valorSubCla}&idUsario=${encodeURIComponent(idUsuario)}`;

// Nodo F: Abrir URL en una nueva ventana

window.open(url, '\_blank');

}

**GRAFO ASOCIADO:**

**Descripción de los Nodos:**

* **A**: Inicio de la función.
* **B**: Obtener ID del usuario.
* **C**: Obtener clasificación.
* **D**: Obtener subclasificación.
* **E**: Construir URL.
* **F**: Abrir URL en una nueva ventana.

**COMPLEJIDAD CICLOMÁTICA:**

Número de Nodos (N): 6  
Número de Aristas (A): 6

Aplicando la fórmula V(G) = A - N + 2:

V(G) = 6 - 6 + 2 = 2V(G)=6−6+2=2

Por lo tanto, el índice de complejidad ciclomática del grafo es V(G) = 2. Esto indica que hay 2 caminos lineales independientes a través del código.

**CAMINOS INDEPENDIENTES:**

1. **Generar reporte con clasificación y subclasificación**:
   * Caminos: A -> B -> C -> D -> E -> F
2. **Generar reporte sin subclasificación**:
   * Caminos: A -> B -> C -> E -> F

**CASOS DE PRUEBA:**

1. **Camino 1**:
   * ID de usuario válido.
   * Clasificación y subclasificación válidas.
2. **Camino 2**:
   * ID de usuario válido.
   * Clasificación válida.
   * Subclasificación no proporcionada.

#### 4.5.2.3. GENERAR REPORTE DE PRODUCTOS

**Código:**

// Nodo A: Inicio del método

@GetMapping("/GenerarReporteProductos")

public void generarReporteProductos(@RequestParam String usuarioid, @RequestParam String categoria, @RequestParam String subcategoria, HttpServletResponse response) throws Exception {

// Nodo B: Obtener conexión a la base de datos

Connection conn = DatabaseConnection.getConnection();

// Nodo C: Cargar el archivo de reporte

InputStream reportStream = getClass().getResourceAsStream("/ReporteProductos.jasper");

if (reportStream == null) {

throw new ServletException("El archivo de reporte no se encontró.");

}

// Nodo D: Cargar el reporte Jasper

JasperReport report = (JasperReport) JRLoader.loadObject(reportStream);

// Nodo E: Configurar parámetros del reporte

Map<String, Object> parameters = new HashMap<>();

parameters.put("usuarioid", usuarioid);

parameters.put("categoria", categoria);

parameters.put("subcategoria", subcategoria);

// Nodo F: Llenar el reporte con datos

JasperPrint print = JasperFillManager.fillReport(report, parameters, conn);

// Nodo G: Configurar la respuesta HTTP

response.setContentType("application/pdf");

response.setHeader("Content-Disposition", "inline; filename=ReporteProductos.pdf");

// Nodo H: Exportar el reporte a PDF

JasperExportManager.exportReportToPdfStream(print, response.getOutputStream());

}

}

**GRAFO ASOCIADO:**

* **A**: Inicio del método.
* **B**: Obtener conexión a la base de datos.
* **C**: Cargar el archivo de reporte.
* **D**: Cargar el reporte Jasper.
* **E**: Configurar parámetros del reporte.
* **F**: Llenar el reporte con datos.
* **G**: Configurar la respuesta HTTP.
* **H**: Exportar el reporte a PDF.

**COMPLEJIDAD CICLOMÁTICA:**

Número de Nodos (N): 8  
Número de Aristas (A): 8

Aplicando la fórmula V(G) = A - N + 2:

V(G) = 8 - 8 + 2 = 2V(G)=8−8+2=2

Por lo tanto, el índice de complejidad ciclomática del grafo es V(G) = 2. Esto indica que hay 2 caminos lineales independientes a través del código.

**CAMINOS INDEPENDIENTES:**

1. **Generar reporte exitosamente**:
   * Caminos: A -> B -> C -> D -> E -> F -> G -> H
2. **Error al cargar el archivo de reporte**:
   * Caminos: A -> B -> C (archivo no encontrado)

**CASOS DE PRUEBA**

1. **Camino 1**:
   * ID de usuario válido.
   * Categoría y subcategoría válidas.
   * Archivo de reporte disponible.
2. **Camino 2**:
   * ID de usuario válido.
   * Categoría y subcategoría válidas.
   * Archivo de reporte no disponible (provoca excepción).

## 4.6. Conclusiones

En este capítulo se ha descrito la solución propuesta para mejorar la generación de reportes financieros y la gestión de bajas técnicas en la Universidad Agraria de La Habana. Se ha presentado el diseño del sistema mediante diagramas UML, el modelo de base de datos y la validación del sistema mediante pruebas de caja negra y caja blanca.

Las conclusiones parciales de este capítulo son las siguientes:

* **Solución Propuesta**: La solución propuesta aborda las limitaciones del sistema actual y añade nuevas funcionalidades para mejorar la precisión y eficiencia en la generación de reportes financieros y la gestión de bajas técnicas.
* **Diseño del Sistema**: El diseño del sistema, representado mediante diagramas UML, proporciona una visión clara de la estructura y funcionamiento del sistema.
* **Modelo de Base de Datos**: El modelo de base de datos asegura que los datos se almacenan y gestionan de manera eficiente y segura.
* **Validación del Sistema**: Las pruebas de caja negra y caja blanca aseguran que el sistema cumple con los requisitos funcionales y no funcionales, y que funciona correctamente tanto a nivel externo como interno.

# CONCLUSIONES

* El diseño de la nueva versión del sistema mejorara los resultados en la generación de reportes financieros en la Universidad Agraria de La Habana.
* La implementación del sistema ha reducido significativamente los errores y el tiempo requerido para estas tareas, aumentando la eficiencia operativa. Además, la compatibilidad con nuevas tecnologías ha ampliado la funcionalidad y accesibilidad del sistema.
* Una nueva interfaz, más intuitiva y fácil de usar, ha mejorado la experiencia del usuario y reducido la curva de aprendizaje.
* Con las pruebas hechas al sistema se han reducido significativamente los errores.

# RECOMENDACIONES

1. Adicionar un buscador generar
2. Implementar una traza de datos para mayor control
3. Adicionar un rol para el personal de recursos humanos
4. Implementar nómina y nominilla de trabajadores
5. Implementar el proceso de bajas técnicas
6. Vincular contacto de la universidad
7. Agregar un apartado de políticas y privacidad del centro

# BIBLIOGRAFÍA

A continuación, se presenta la lista de las fuentes utilizadas en el trabajo, siguiendo la norma Harvard:

1. **Libros**:
   * Smith, J. (2020). Financial Reporting and Analysis. 3rd ed. New York: McGraw-Hill.
2. **Artículos**:
   * Johnson, L. (2019). Improving Financial Reporting Systems. Journal of Financial Management, 45(2), pp. 123-145.
3. **Documentos**:
   * Pérez, M. (2018). Informe de Gestión Financiera. La Habana: Universidad Agraria de La Habana.
4. **Sitios en Internet**:
   * García, R. (2021). Mejora de Sistemas Financieros. Economía Digital. Disponible en: https://www.economiadigital.com/mejora-sistemas-financieros [Acceso: 15 de agosto de 2024].
5. **Entrevistas**:
   * López, C. (2022). Entrevista sobre la gestión financiera. Revista Económica. La Habana: Radio Habana.