



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

Tema: Sistema web para la gestión de Información en el Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales.

Autor: Javier Horsford Rivera

Tutor: Ing. Glennis Tamayo Morales

Co-tutor: Ing. Rafael Mendoza Fernández

DECLARACIÓN DE AUTORÍA

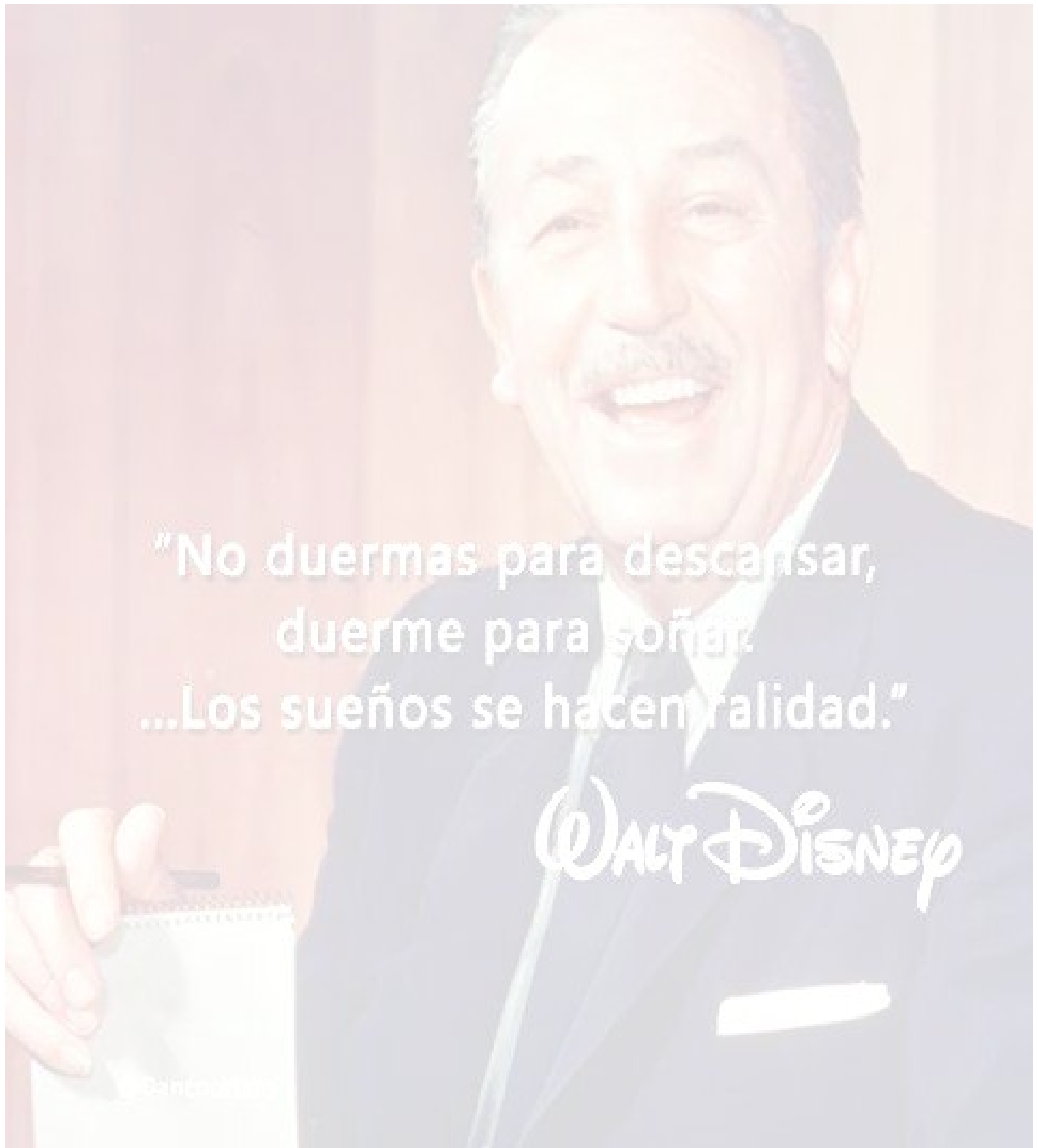
Declaro ser único autor del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo.

Para que así conste firma la presente a los ____ días del mes de noviembre del año 2022.

Firma de Autor
Javier Horsford Rivera

Firma de Tutor
Ing. Glennis Tamayo Morales

Firma de Co-Tutor
Ing. Rafael Mendoza Fernández



A mi madre

Agradecimientos

Durante mi tiempo en la Universidad de las Ciencias Informáticas fueron muchas las personas con las que compartí este camino, algunos se han quedado por mucho tiempo, otros solo llegaron por un rato, pero significaron algo para mí. Realmente el camino no fue fácil, hubo varios tropiezos, pero con esfuerzo, ganas y una madre como la que me tocó todo se puede. A ella le agradezco todo lo que ha hecho por mí y hoy le dedico este día, esta meta cumplida y espero que se sienta orgullosa del hijo que tiene. Agradezco a Rafa que después de tantos años me extendió la mano no solo como tutor, sino como amigo de tantos años, a la profesora Glennis que a pesar de tener muchos tesisistas me hizo un hueco entre su equipo. Le agradezco a Yolanda la paciencia que siempre ha tenido conmigo jaja y yo con ella claro está, por no estar solo en los buenos momentos sino también en los malos y agobiantes. A pesar de no tener hermanos la vida me cruzo en el camino a ti Eduardito, pues empezamos los estudios juntos, siempre uno al lado del otro, y pues este título también es dedicado a ti. No por mencionar de último deja de tener menos significado, pero agradezco a Elio, Cesar, Adrián, Ale, Juni, Alfred, a todos en la Empresa ALME y repito todas esas personas que son mucho y si los menciono a todos aquí, el documento de tesis llega a 100 hojas, Muchas gracias.

Resumen

La gestión de la información de los activos fijos tangibles del Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC) se realiza de forma manual, ello trae consigo que este proceso de gestión sea complejo, debido a la gran cantidad de información sobre esos activos que se maneja en las áreas de la facultad. El presente trabajo de investigación consiste en el desarrollo de un sistema web concebido con tecnologías libres para facilitar dicho proceso. Este sistema web fue guiado por la metodología AUP-UCI en su escenario 2. Como tecnologías del lado del cliente se utilizaron: HTML, JavaScript y CSS, apoyado en el framework de diseño Bootstrap. El lenguaje del lado del servidor empleado fue PHP, a través del framework Laravel. Se utilizó el Visual Studio Code como IDE de desarrollo, Apache como servidor web y MySQL para la gestión de bases de datos. Se implementaron funcionalidades en la gestión de Activos Fijos Tangibles y sus movimientos, Expedientes Técnicos, Locales, Áreas, Usuarios, Permisos, Roles. También se implementaron funcionalidades para exportar archivos y reportes, tales como: mostrar listado de Activos Fijos Tangibles, listado de movimientos realizados de los mismos y listado de Expedientes Técnicos. Fueron realizadas pruebas de software a la propuesta de solución, las cuales ofrecen niveles de validación de las funcionalidades y la satisfacción por parte del cliente.

Palabras claves: activo fijo, gestión, información, sistema web.

ABSTRACT

The management of the information of the tangible assets of the Department of Technology of the Faculty of Sciences and Computational Technologies (FSCT) is carried out manually, which means that this management process is complex, due to the large amount of information on those assets. assets that are handled in the areas of the faculty. This research work consists of the development of a web system designed with free technologies to facilitate this process. This web system was guided by the AUP-UCI methodology in its scenario 2. HTML, JavaScript and CSS were used as client-side technologies, supported by the Bootstrap design framework. The server-side language used was PHP, through the Laravel framework. Visual Studio Code was acquired as development IDE, Apache as web server and MySQL for database management. Functionalities were implemented in the management of Tangible Fixed Assets and their movements, Technical, Local Files, Areas, Users, Permits, Roles. Functionalities were also implemented to export files and reports, such as: showing a list of Tangible Fixed Assets, a list of movements made of them and a list of Technical Files. Software tests were carried out on the solution proposal, which offer validation levels of the functionalities and customer satisfaction.

Keywords: *fixed assets, management, information, web system*

Índice

INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	6
I.1 Fundamentos Teóricos.....	6
Gestión de la información.....	6
Áreas de conocimiento y de proceso de Gestión de la Información.....	7
Áreas del conocimiento.....	7
Procesos de Gestión de Información.....	8
Tecnologías de la Información y Comunicación (TIC).....	8
Activos Fijos Tangibles.....	10
I.2 Análisis de Sistemas Informáticos de Gestión de la Información.....	11
I.2.1 Sistemas de Gestión de la Información de Activos Fijos Tangibles y Útiles a nivel mundial.....	13
Cetaris Fleet.....	13
UpKeep.....	14
AssetTiger.....	15
I.2.2 Sistemas de Gestión de la Información de Activos Fijos Tangibles y Útiles a nivel nacional.....	16
AvilaDocPro.....	16
MEBUS.....	16
ASSETS.....	17
I.2.3 Evaluación de sistemas existentes.....	17
I.3 Tecnologías de desarrollo de software.....	21
I.3.1 Metodologías de desarrollo de software.....	21
I.3.2 Lenguaje de Modelado Unificado (UML v.2.1).....	22

Herramienta de modelado.....	22
Visual Paradigm CE (v.17.0).....	22
I.3.3 Lenguajes y tecnologías de programación.....	23
JavaScript (v. ECMAScript 2021).....	24
PHP (v.8.1.0).....	24
HTML (v.5.0).....	24
CSS (v.3.0).....	24
I.3.4 <i>Frameworks</i> de desarrollo.....	25
Laravel (v.9.2).....	26
Bootstrap (v.5.2.0).....	27
I.3.5 Sistema gestor de base de datos.....	27
MySQL (v.9.1).....	28
I.3.6 Entornos de Desarrollo Integrado.....	28
Visual Studio Code (v.1.70.2).....	28
Conclusiones del capítulo.....	29
CAPÍTULO II: ANÁLISIS Y DISEÑO.....	30
II.1 Mapa Conceptual.....	30
II.2 Requisitos del Sistema.....	31
Requisitos Funcionales.....	31
Requisitos No Funcionales.....	35
Usabilidad.....	35
Requisitos del Software.....	35
Requisitos de Hardware.....	36
Restricciones de Diseño e Implementación.....	36
Interfaz.....	36

II.3 Descripción del Sistema.....	36
Actores del Sistema.....	37
II.4 Arquitectura del Sistema.....	40
II.4.1 Patrones de Diseño.....	40
II.4.1.1 Patrones GRASP.....	40
Experto.....	40
Creador.....	41
Alta cohesión.....	41
Bajo acoplamiento.....	42
Controlador.....	42
II.4.1.2 Patrones GOF.....	42
Singleton.....	43
Decorador.....	43
Observador.....	43
II.4.1.3 Patrones de Arquitectura.....	44
II.4.2 Diagrama de clase del diseño.....	45
II.4.3 Diagrama de Secuencia.....	47
II.4.4 Modelo de Datos.....	47
II.4.5 Diagrama de Despliegue.....	49
Conclusiones del capítulo.....	49
CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN.....	50
III.1 Modelo de Implementación.....	50
Diagrama de Componentes.....	50
III.2 Pruebas de Software.....	51
III.2.1 Pruebas unitarias.....	51

Métodos de pruebas.....	52
Método de Caja Blanca.....	52
Método de Caja Negra.....	52
Técnicas de pruebas.....	52
Camino Básico.....	52
III.2.2 Pruebas de aceptación.....	54
Tipos de pruebas.....	54
Pruebas de Funcionalidad.....	54
Pruebas no Funcionales.....	54
Pruebas de Usabilidad	55
III.3 Casos de pruebas.....	55
Conclusiones del capítulo.....	59
CONCLUSIONES GENERALES.....	60
RECOMENDACIONES.....	61
REFERENCIAS BIBLIOGRÁFICAS.....	62
ANEXOS.....	65

ÍNDICE DE TABLAS

Tabla 1: Comparación de los Sistemas de Gestión de Información.....	20
Tabla 2: Descripción de los actores del sistema.....	37
Tabla 3: Flujos básicos del Caso de Uso "Gestionar ATF".....	39
Tabla 4: Prueba unitaria: Técnica de camino básico.....	54
Tabla 5: Diseño de caso de prueba: RF32 Crear Expediente Técnico.....	56
Tabla 6: Diseño de prueba de validación RF29 Registrar Expediente Técnico.....	57

ÍNDICE DE FIGURAS

Figura 1: Frameworks más populares hasta enero de 2022.....	26
Figura 2: Mapa conceptual.....	31
Figura 3: Diagrama de Caso de uso del sistema.....	38
Figura 4: Caso de uso Gestionar ATF.....	38
Figura 5: Patrón experto.....	40
Figura 6: Patrón Creador.....	41
Figura 7: Patrón Alta cohesión.....	41
Figura 8: Patrón Bajo acoplamiento.....	42
Figura 9: Patrón controlador.....	42
Figura 10: Patrón Singleton.....	43
Figura 11: Patrón Decorador.....	43
Figura 12: Patrón Observador.....	43
Figura 13: Arquitectura Modelo-Vista-Controlador.....	45
Figura 14: Diagrama de Clase de Diseño del CU "Gestionar ATF".....	46
Figura 15: Diagrama de Secuencia del caso de uso Registrar ATF.....	47
Figura 16: Modelo de Datos del Sistema.....	48
Figura 17: Diagrama de Despliegue.....	49
Figura 18: Diagrama de Componentes del CU "Gestionar ATF".....	51
Figura 19: Resultado de las pruebas de aceptación.....	58

INTRODUCCIÓN

En la actualidad, para las empresas es primordial definir cómo hacer la información fructífera en función de su calidad, para que la gestión empresarial se traduzca en resultados convenientes para su desarrollo. Para ello dichas empresas emplean una gama de sistemas y herramientas dirigidas a la administración y creación de conocimiento.

Una de las principales consecuencias de los avances tecnológicos es la cantidad de información de interés generada diariamente, siendo cada vez más importante almacenarla y recuperarla, generalmente esta se encuentra dispersa, poco estructurada e invisible lo que hace muy engorroso el proceso de obtención e imposibilita el rápido acceso a la información.

La gestión de la información agiliza los procesos en los diferentes ambientes laborales utilizando sistemas capaces de reunir, organizar y vincular la información almacenada. Lo fundamental es saber cómo manejar y formalizar los datos que se tienen para proporcionar una óptima toma de decisiones y así agilizar los procesos involucrados. [1]

El Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC), gestiona un gran volumen de información relacionada con el control de los Activos Fijos Tangibles y Útiles, los expedientes técnicos de los ordenadores y el mantenimiento de estos y los movimientos de los medios básicos.

El expediente técnico del ordenador es un documento oficial que registra los datos de cada uno de los componentes y accesorios del ordenador, y el registro histórico de cada una de las veces que ha sido abierto el ordenador y lo realizado en él, ya sea por mantenimiento, por rotura o por modificación de sus componentes. Es controlado por uno de los especialistas del propio departamento de tecnología.

Otro de los procesos que se llevan a cabo en el área es la gestión y control de los activos fijos tangibles y de los útiles. Si bien es cierto que, en la Universidad de Ciencias Informáticas (UCI), en el departamento de economía existe un sistema que registra los movimientos de los activos y donde se encuentran almacenados, para el trabajo diario y la propia dinámica del departamento esto resulta engorroso.

Para realizar un movimiento de un activo debe llenarse un modelo en formato duro ser firmado por varias personas y entregado a economía para que este departamento a su vez realice

los movimientos en el sistema. Sin embargo, a nivel del departamento para consultar los movimientos que se han hecho hasta tanto el departamento de economía no los haga efectivo se deben consultar los modelos que se tienen archivados uno por uno para saber dónde fue colocado el medio.

Entre otras de las tareas que tiene el departamento, se encuentra la de prestar atención y mantenimiento a los ordenadores de las oficinas de los departamentos docentes y no docentes de la facultad. Este proceso inicia cuando el jefe de un departamento o profesor se presenta a las oficinas del Departamento de Tecnología y pone en conocimiento a cualquiera de sus especialistas que tiene su ordenador roto.

El especialista recoge el ordenador (con expediente técnico incluido) y se realiza un movimiento temporal del medio hacia el local departamento de tecnología. Se cita al técnico de la dirección tecnológica y se procede a diagnosticar el problema, de tener solución se resuelve y se entrega el medio a sus responsables.

De todos los procesos anteriormente descritos se han identificado un conjunto de problemas que a continuación se detallan:

- Existencia de expedientes técnicos duplicados.
- Existencia de ordenadores sin expedientes técnicos que se han extraviado.
- Para conocer las prestaciones con la que cuentan el ordenador se hace necesario consultar uno a uno los expedientes técnicos.
- Al ser información que se almacena por largo tiempo, existen expedientes que se han deteriorado.
- El modelo de expediente técnico es obsoleto, carece de elementos que se deben tener en cuenta; como por ejemplo que el responsable del medio autoriza el trabajo que se realizará, quién estuvo presente durante los trabajos.
- No quedan registrados los trabajos que a diario se realizan de mantenimiento y reparación de la tecnología de oficinas, aulas, salones y laboratorios.
- A final de mes se hace realmente imposible tener una estadística de cuanto se hizo en materia de reparaciones o mantenimiento, lo cual no permite tener una noción del trabajo que realizan los especialistas del departamento de tecnología de la facultad.

- No se registran las incidencias de las roturas que pueda tener un ordenador con lo cual no se conoce a ciencia cierta que ordenadores fueron reportados o cuantos fueron diagnosticados de rotos en un periodo de tiempo determinado.

A partir de la situación problemática anteriormente planteada se declara para la presente investigación el siguiente **problema de investigación**: ¿Cómo contribuir a la gestión de la información del Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC)?

Se plantea como **objeto de estudio** de la investigación, los procesos para la gestión de la información de medios básicos en una organización definiendo como **campo de acción** la gestión de información en la Facultad de Ciencias y Tecnologías Computacionales (CITEC).

Por lo que se declara como **objetivo general**: Desarrollar un Sistema web que permita la Gestión de la Información de activos fijos tangibles en el Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC).

Se definieron las siguientes **tareas investigativas**:

1. Caracterización de los diferentes Sistemas web para la Gestión de Información existentes especializados en Activos Fijos Tangibles y Útiles.
2. Caracterización de la metodología, herramientas y tecnologías a utilizar en el desarrollo del Sistema web.
3. Identificación de las funcionalidades del Sistema Web para la Gestión de Información en el Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC).
4. Diseño e implementación de las funcionalidades del Sistema web para la Gestión de la Información a partir de las funcionalidades identificadas.
5. Diseño y aplicación de los casos de pruebas para validar que el Sistema web para la Gestión de Información en el Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC) responde a las necesidades del usuario final.

Los **métodos científicos** utilizados en la investigación fueron los empíricos y teóricos. Dentro de los métodos empíricos se utilizó:

- La **Entrevista**, para identificar las necesidades del Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC) y para obtener las funcionalidades con que contará el sistema informático.
- La **Observación**, que permite constatar a partir de las prácticas cotidianas y la percepción directa en el panorama real el estado actual y funcionamiento del proceso de gestión de información del Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC).

A nivel teórico fueron utilizados los métodos:

- **Analítico–Sintético**, para realizar un estudio bibliográfico de la teoría existente alrededor del objeto de estudio, determinar las características que tendrá la propuesta de solución, y definir las tecnologías y Herramientas más adecuadas para el desarrollo de la propuesta de solución.
- **Histórico–Lógico**, que permite el estudio crítico de soluciones anteriores referidas en la bibliografía científica, para extraer aspectos positivos de ellos, así como el uso de trabajos como punto de referencia para la definición de requisitos preliminares.

El Trabajo de Diploma está estructurado de la siguiente manera: Introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y anexos.

Capítulo I: Fundamentación teórica

En este capítulo se presenta la definición del marco teórico de la investigación. En el mismo se abordan conceptos de sistema informático y procesos. Se exponen características de sistemas informáticos existentes vinculados a la gestión de información. Se aborda el estudio de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo del sistema.

Capítulo II: Análisis y Diseño

En el capítulo se describe la propuesta de solución para el problema de la investigación. Describiendo los principales conceptos del entorno que serán objeto de análisis para la realización de la fase de análisis y diseño del Sistema web para la gestión de información del Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC).

Se hace el levantamiento de requisitos funcionales y no funcionales para darle cumplimiento al objetivo general de la investigación. Además, se muestran los diagramas de modelo con-

ceptual, de caso de uso del sistema de clase de diseño y de despliegue, por medio de los cuales se representa la estructura estática del sistema y los patrones de arquitectura y diseño utilizados.

Capítulo III: Implementación y validación

En el capítulo se aborda las actividades que se llevan a cabo durante la fase de implementación y pruebas. Se analizará el modelo de implementación, así como una descripción de cómo los elementos del modelo de diseño se implementan en términos de componentes. Además, se diseña y aplican las pruebas para comprobar el correcto funcionamiento del sistema.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se estudia el marco teórico de la investigación. Se describen conceptos de los sistemas de web y se hace un estudio de los existentes vinculados a la gestión de información. Se aborda el estudio de la metodología de desarrollo, herramientas, tecnologías y lenguajes a utilizar en el desarrollo del sistema.

I.1 Fundamentos Teóricos

Gestión de la información

La Gestión de la Información (GI) se refiere a un ciclo de actividad organizacional y al desarrollo, simulación o modelado de sistemas de información, aplicables a áreas de gestión en organizaciones para la adquisición de información de una o más fuentes, la custodia y la distribución de esa información a aquellos que la necesitan, y su disposición final a través del archivado o borrado. [2]

Este ciclo de implicación organizacional con la información implica a una variedad de partes interesadas, incluyendo las que son responsables de asegurar la calidad, la accesibilidad y la utilidad de la información adquirida; los responsables de su almacenamiento y eliminación seguros; y los que lo necesitan para tomar decisiones. Los interesados podrían tener derecho a originar, cambiar, distribuir o eliminar información de acuerdo con las políticas de gestión de la información organizativa.

La gestión de la información abarca todos los conceptos genéricos de la gestión, incluyendo la planificación, organización, estructuración, procesamiento, control, evaluación y presentación de informes de actividades de información. [3]

Está estrechamente relacionada con la gestión de datos, sistemas, tecnología, procesos y donde la disponibilidad de información es crítica para el éxito organizacional. Esta visión amplia del ámbito de la gestión de la información contrasta con la visión anterior, más tradicional, de que el ciclo de vida de la gestión de la información es un asunto operacional que requiere procedimientos específicos, capacidades organizativas y normas que tratan con información como un producto o un servicio.

El Cuerpo de Conocimientos de Gestión de la Información (*IMBOK*, por sus siglas en inglés) [4] se hizo disponible en la *World Wide Web* en 2004 y propone demostrar que las gestiones de informaciones necesarias para obtener beneficios reales son complejas y de varias capas.

Áreas de conocimiento y de proceso de Gestión de la Información

El modelo marco que es la base para entender la gestión de información comprende cinco áreas de "conocimiento" y cuatro áreas de "proceso":

Áreas del conocimiento

- Tecnología de la información: El ritmo de cambio de tecnología y la presión para adquirir constantemente los productos tecnológicos más novedosos pueden socavar la estabilidad de la infraestructura que soporta los sistemas y, por lo tanto, optimizar los procesos del negocio y ofrece ventajas. Es necesario gestionar el "lado de la oferta" y reconocer que la tecnología se está convirtiendo cada vez más en una mercancía. [5]
- Sistema de información: Mientras que históricamente los sistemas de información fueron desarrollados internamente, a lo largo de los años ha llegado a ser posible adquirir la mayoría de los sistemas de software que una organización necesita de la industria del paquete de software. Sin embargo, todavía existe la posibilidad de una ventaja competitiva a partir de la implementación de nuevas ideas de sistemas que entreguen a las intenciones estratégicas de las organizaciones. [6]
- Procesos de negocio e información de negocios: Los sistemas de información se aplican a los procesos del negocio para mejorarlos, y aportan datos al negocio que se vuelve útil como información de negocio. La gestión de procesos de negocio sigue siendo considerada como una idea relativamente nueva porque no se adopta universalmente y ha sido difícil en muchos casos; la gestión de la información empresarial es aún más un desafío. [7]
- Beneficio de negocio: Desde el surgimiento y la popularización del *Balanced Scorecard* [8] ha habido un gran interés en la gestión del desempeño empresarial, pero no se ha hecho mucho esfuerzo para relacionar la gestión del desempeño empresarial con los beneficios de la tecnología de la información inversiones y la introducción de nuevos sistemas de información hasta el cambio de milenio.

- Estrategia de negocios: aunque muy lejos de las cotidianas cuestiones de gestión de la información en las organizaciones, la estrategia en la mayoría de las organizaciones simplemente debe ser informada por las oportunidades de tecnología de la información y sistemas de información, ya sea para abordar los pobres rendimientos o para mejorar la diferenciación y la competitividad. [8]

Procesos de Gestión de Información

- Proyectos: La tecnología de la información es sin valor hasta que se diseña en sistemas de información que satisfagan las necesidades del negocio por medio de la buena gestión de proyectos. [9]
- Cambio de negocio: los mejores sistemas de información tienen éxito en la entrega de beneficios a través del logro del cambio dentro de los sistemas de negocio, pero la gente no aprecia el cambio que hace nuevas demandas sobre sus habilidades en las maneras que los nuevos sistemas de información hacen a menudo. Contrariamente a las expectativas comunes, hay algunas pruebas de que el sector público ha logrado un cambio de negocio inducido por la tecnología de la información. [10]
- Operaciones de negocio: Con nuevos sistemas en el lugar, con procesos de negocio e información de negocio mejorada, y con el personal finalmente listo y capaz de trabajar con nuevos procesos, entonces el negocio puede llegar a trabajar, incluso cuando los nuevos sistemas se extienden mucho más allá de los límites de un solo negocio. [10]
- Gestión del desempeño: Las inversiones ya no son solamente sobre los resultados financieros, el éxito financiero debe ser equilibrado con la eficiencia interna, la satisfacción del cliente, y con el aprendizaje y desarrollo organizacional. [8]

Tecnologías de la Información y Comunicación (TIC)

Respecto a la evolución de las Tecnologías de la Información y Comunicación (TIC) en el ámbito tecnológico, es de destacar que los avances producidos en este campo han sido espectaculares y radicales tanto en los soportes físicos, con una mayor velocidad y capacidad de procesamiento y almacenamiento de la información que posibilitan la gestión y digitalización de cualquier tipo de información.[11]

Especial importancia en la evolución de las TIC tiene el desarrollo del software, que ha permitido la aparición de avanzadas herramientas informáticas de gestión con nuevas funcionalidades y aplicaciones empresariales basadas en sistemas web.

Los sistemas web tienen la peculiaridad de estar alojados en un servidor de internet o sobre una intranet (red local/privada), lo cual hace que ellos no dependan de ser instalados sobre una plataforma o sistema operativo en específico. Su aspecto es muy similar a un sitio web que se ve normalmente, pero en realidad los sistemas web van un paso más allá, porque cuentan con funcionalidades que brindan respuesta a casos muy particulares.

Cabe recalcar que estos sistemas son totalmente compatibles en cualquier navegador web (Google Chrome, Firefox, Edge, Safari), este tipo de diferencia se ve reflejada en los costos, en la rapidez de obtención de la información, en la optimización de tareas y por alcanzar una gestión estable.

Los sistemas web tienen varias ventajas sobre los programas de software descargables tradicionales:

- Fáciles de usar: son muy sencillos de utilizar, sólo necesitarás conocimientos básicos de informática para trabajar con ellas. Si sabe escribir un correo electrónico, ya sabe usarlas. Además, en muchos casos podrá personalizarlas a su gusto y adaptarlas a su forma de trabajo.
- Ahorran costes de hardware y software: como se comentó anteriormente, sólo es necesario usar un ordenador con un navegador web y conectarse a Internet o Intranet (red local/privada). Las aplicaciones basadas en web usan menos recursos que los programas instalados. Por otra parte, los sistemas web no requieren canales de distribución como el software tradicional, lo que permite que su precio sea inferior al de los programas instalables.
- Facilitan el trabajo colaborativo y a distancia: pueden ser usados por varios usuarios al mismo tiempo. Al estar toda la información centralizada no tendrá que compartir pantallas o enviar emails con documentos adjuntos. Varios usuarios pueden ver y editar el mismo documento de manera conjunta. Además, son accesibles desde cualquier lugar. Puede trabajar desde un pc, un portátil, un móvil o una tablet, desde la oficina, un parque o un aeropuerto.

- Escalables y de rápida actualización: existe solo una versión del sistema web en el servidor, por lo que no hay que distribuirla entre los demás ordenadores. El proceso de actualización es rápido y limpio. Los sistemas basados en web no requieren que el usuario se preocupe por obtener la última versión ni interfieran en su trabajo diario para descargar, instalar y configurar últimas versiones.
- Provocan menos errores y problemas: son menos propensos a colgarse y crear problemas técnicos debido a conflictos con hardware, con otras aplicaciones existentes, protocolos o con software personal interno. Todos los usuarios utilizan la misma versión de la aplicación web y los posibles fallos pueden ser corregidos tan pronto son descubiertos.
- Los datos son más seguros: no deberá preocuparse de posibles rupturas del disco duro ni de los virus que pueden hacerle perder toda la información. Los proveedores de hosting donde se almacenan las aplicaciones usan granjas de servidores, con altísimas medidas de seguridad, donde guardan los datos de forma redundante y con amplios servicios de backups.

Activos Fijos Tangibles

La Gestión de la Información es utilizada también para llevar el control de los Activos Fijos Tangibles (AFT) que constituyen los bienes materiales que una entidad utiliza de manera continua en sus operaciones, con un carácter permanente, los cuales sufren una depreciación durante su vida útil que afectará al final el nivel de utilidades. [12]

Para la denominación de Activos Fijos Tangibles el criterio más utilizado es el de aquellos objetos materiales con una duración de más de un año. En el transcurso de la utilización de estos medios, estos se desgastan gradualmente, pierden exactitud y potencia y en última instancia se rompe; las roturas también conocidas como averías, crean interrupciones considerables y pueden llegar a detener los procesos, provocando pérdidas económicas.

No obstante, a las diferentes denominaciones expresadas, por lo general los autores coinciden en que los Activos Fijos Tangibles:

- Son medios creados por el trabajo humano.
- Tienen vida limitada.
- Son objetos físicos.
- Transfieren su valor a los productos que crean o a los servicios que prestan.
- Pierden su valor a medida que se desgastan físicamente.

En el área de los laboratorios, aulas, salones y departamento se maneja mucha información relacionada con los medios básicos. Ejemplo, de un Activo Fijo Tangible es importante conocer su no. inventario, no. de serie, su valor, en la medida en que en el departamento tenga una estrategia, identificado un problema para cuando se tome una decisión se piense en soluciones prácticas a partir del conocimiento que se tiene de la gestión de la información y del estado en que se encuentra para realizar algún movimiento de ese medio básico.

Los Útiles están relacionados con los Activos Fijos, son los medios con un periodo de vida útil más corto, generalmente se desgastan con el tiempo por lo que deben reponerse cada cierto periodo. [13] No se les calcula depreciación ya que en su mayoría pasan directamente a la cuenta de gastos.

El Expediente Técnico es un instrumento de trabajo utilizado por el Departamento de Tecnología de la UCI para tener el control de los activos fijos tangibles y útiles, el cual comprende Datos Generales (Área de Responsabilidad, Local, Puesto, Responsable, No. Inventario, No. Serie), Datos Técnicos (Información tipo fabricante, modelo, no. serie, cantidad de los componentes de la torre del ordenador), Datos Periféricos (Información tipo fabricante, modelo, no. serie, cantidad de los componentes de la torre del ordenador) y un Historial de Modificaciones (Donde se registran el historial de los mantenimientos, roturas y modificaciones, modificaciones de por vida).

1.2 Análisis de Sistemas Informáticos de Gestión de la Información

Actualmente en el mundo los procesos económicos y sociales se encuentran totalmente automatizados. De esta manera se obtiene una mayor calidad del proyecto y asociación en el equipo de trabajo, así como una mayor velocidad de desplazamiento de la información. Exis-

ten diversos sistemas web de Gestión de la Información que cumplen funciones específicas en las entidades a las que se asocian. A pesar de que cada entidad funciona de manera particular, el funcionamiento de sus sistemas web de gestión de información sirve de apoyo teórico para el cumplimiento del Objetivo General de la presente investigación. [14]

Entre las clases de sistemas web de Gestión Información se encuentran las siguientes:

- Sistemas de información administrativa (MIS): El *Management Information System* se ocupa de facilitar la información de interés a la gerencia de la compañía, sobre la situación general del negocio.
- Sistemas de procesamiento de transacciones (TPS): El *Transaction Processing System* se ocupa de almacenar y procesar la información referida con las transacciones comerciales y operaciones de la compañía. Estos datos se emplearán a continuación por los sistemas de apoyo a la toma de decisiones.
- Sistemas de soporte de decisiones (DSS): El *Decision Support System* es una herramienta que contribuye a la toma de decisiones. Consiste en la combinación y estudio de datos que aportan información de valor para ayudar a resolver cuestiones concretas.
- Sistemas de apoyo a ejecutivos (EIS): está pensada para extraer información de interés para conseguir las metas estratégicas de la organización.
- Sistemas para la toma de decisiones en grupo (GDSS): el *Group Decision Support System* contribuye a compartir la información entre los componentes del equipo, para que de esta manera tengan la posibilidad de trabajar en equipo y asumir decisiones conjuntas.
- Sistemas expertos de soportes a la toma de decisiones (EDSS): estos sistemas se centran en el conocimiento de áreas concretas y actúan como consultores expertos.
- Sistemas estratégicos: tienen el propósito de lograr ventajas competitivas por medio de la utilización de la tecnología de la información. Buscan alcanzar ventajas que lo competidores no tengan.

- Sistema de Información de Marketing: cuenta con la función de promocionar y vender los productos existentes, además del desarrollo de nuevos artículos para los clientes ya existentes y para los que puedan surgir en el futuro.

Además de estas clases de sistemas de información, también hay otras que recopilan información con el propósito de mejorar las estrategias corporativas, analizar y conocer la contabilidad y finanzas de la empresa, mejorar la relación con los consumidores o la gestión del personal. Cada vez resultan más común la creación de sistemas de información integrales, que reciben el nombre de ERP, que son las siglas de *Enterprise Resource Planning*, que recogen, estudian y transmiten datos de interés referentes a varios campos.

I.2.1 Sistemas de Gestión de la Información de Activos Fijos Tangibles y Útiles a nivel mundial

Entre los más importantes a nivel internacional se encuentran:

Cetaris Fleet

Es un sistema de gestión de activos fijos que se utiliza para optimizar los costes, ahorrar tiempo y mejorar la eficiencia operativa. Las soluciones de Cetaris proporcionan a los equipos de mantenimiento las herramientas para administrar sus activos todos los días: garantía y registro central, mantenimiento preventivo, mano de obra y productividad, además de herramientas de gestión de inventarios. Con un flujo constante de datos, los usuarios de Cetaris pueden medir y optimizar continuamente sus operaciones en tiempo real con informes completos.

Características

- Automatización de inventario: Los recuentos se validan en tiempo real. Si ocurre una variación, se genera una hoja de recuento. Las variaciones se pueden configurar según el porcentaje, la cantidad o el valor de la pieza.
- Inventario completo o recuentos de ciclos: Diseñado para complementar su flujo de trabajo único, Cetaris se puede configurar para completar ciclos o recuentos de inventario completos según las necesidades de su operación.
- Flujos de conteo continuos: Obtiene información clara sobre los recuentos de inventario para optimizar el control de inventario. Alinee a su equipo con un flujo constante de datos de inventario para las personas adecuadas.

- Escanear códigos de barras: Con el escaneo de códigos de barras de piezas, dedica más tiempo al trabajo que importa, como lograr un 100 % de precisión en la sala de piezas. Use la cámara de su dispositivo para escanear partes, seleccione la cantidad y actualice instantáneamente sus conteos.
- Facilidad de visualizar inventarios: Con una línea de visión clara de los conteos, puede identificar y abordar los cambios a medida que ocurren para optimizar el control de inventario.
- Herramientas de datos, análisis e informes: Procesa datos utilizando algoritmos estadísticos avanzados con solo hacer clic en un botón y transforme grandes conjuntos de datos en informes visuales concisos. Convierta datos precisos en decisiones de gestión bien informadas. Reúna los datos para ver información de alto nivel o profundice hasta el último detalle. [15]

UpKeep

El software de gestión de activos fijos de UpKeep es una solución moderna de mantenimiento, se puede acceder fácilmente a UpKeep desde cualquier lugar y en cualquier momento. Crea órdenes de trabajo en el momento, recibe notificaciones cuando las tareas estén actualizadas y recibe alertas directamente desde tu aplicación cuando los activos disminuyan, haciendo que tu negocio se ejecute más eficientemente que nunca. Descarga y registro 100% gratis.

Características

- Tecnología móvil y fácil de usar para ayudar a los técnicos a ser más eficientes: Brinda a cada técnico una forma de acceder a todas las herramientas y conocimientos más rápido. El software altamente adoptado y sin fricciones es muy fácil, conveniente de usar y funciona de la misma manera que lo hacen los técnicos.
- Asegura una documentación consistente con códigos de barras: Crea un código de barras único para cualquier pieza con un solo clic. Los técnicos pueden completar instantáneamente las órdenes de trabajo escaneando los códigos de barras de las piezas.
- Reduce el tiempo de inactividad con cantidades precisas de piezas: Asigna un valor de cantidad mínima a cada parte y recibe una notificación cuando se alcance ese valor o

cuando se esté agotando. Puede reordenar fácilmente las piezas por adelantado para evitar miles de dólares en costos de tiempo de inactividad.

- Una fuente de verdad para todas las necesidades de inventario: Los conteos de inventario se actualizan automáticamente cuando se agregan a las órdenes de trabajo. El software sabe exactamente dónde y cuántas piezas tiene para evitar pedidos excesivos o costos elevados asociados con el envío de piezas. [16]

AssetTiger

Es una solución gratuita hasta los 250 activos, fácil de usar y totalmente funcional para todas tus necesidades de administración de activos fijos. El programa ofrece todos tus activos al alcance de tu mano en un sitio web elegante, fácil de entender y personalizado. Conecta tus sitios de trabajo en todo el mundo con acceso cuando lo necesites

Características:

- Usuarios ilimitados sin cargo adicional: Cada usuario puede tener su propio nivel único de acceso que va desde visualización limitada hasta administrador total. AssetTiger está diseñado como una solución conveniente para todos en su organización.
- Informes personalizados ilimitados: Puede ver sus activos con los informes predefinidos que incluyen declaraciones sobre sus activos, incorporando factores como su estado, depreciación, mantenimiento, tiempos de salida y más. Alivie su carga de trabajo con una cantidad ilimitada de informes totalmente personalizables para satisfacer sus necesidades. Guarde sus informes y reutilícelos o actualícelos para aprovechar al máximo sus datos.
- Basado en la nube: No tiene que comprar y actualizar continuamente el software físico. Nuestro sistema basado en la nube le permite concentrarse en sus activos en lugar de en su programa de gestión de activos. Estamos en constante evolución para cumplir y superar las expectativas de los clientes, los avances tecnológicos y el mundo que nos rodea. Nuestro sistema lo libera de realizar pruebas o actualizaciones, ahorrándole costosos gastos generales de TI.
- Programación de mantenimiento: Protege sus activos incorporándolos a un programa de mantenimiento programado. AssetTiger brinda la posibilidad de establecer una rutina basada en el horario y las necesidades de los horarios. Supervisa el

mantenimiento de los activos a través de los diversos informes que analizan las etiquetas de activos, las personas y el historial de mantenimiento. [17]

I.2.2 Sistemas de Gestión de la Información de Activos Fijos Tangibles y Útiles a nivel nacional

AvilaDocPro

Es un sistema de gestión de información destinado a la gestión, tramitación y resguardo de información archivística realizado por la Empresa Desarrolladora de Software de la División Ciego de Ávila-Camagüey.

Este sistema facilita la búsqueda o recuperación de forma rápida y sencilla. Incorpora el fichado de la documentación en un expediente como punto de partida, simulando dos flujos de procesos: el flujo de la documentación en una entidad y el de incidencias, e integrándolos sobre la base del uso de normas internacionales que rigen el trabajo con los archivos de información. [18]

MEBUS

Es un sistema automatizado de control de los activos fijos tangibles para el sector de la salud, forma parte del Proyecto SEUS (Sistema Económico para Unidades de Salud) que se está empleando en las entidades del sector de la Salud. La aplicación se programó en Visual Basic 6.0, debido a sus poderosas herramientas en la programación visual, así como en la orientada a objetos.

Características

- Permite obtener una serie de informes significativos para el control de los activos fijos tangibles y el proceso de informaciones esenciales que se brinda a niveles superiores.
- Encierra la información de los activos fijos tangibles.
- Ofrece la desvalorización por cada medio, brinda la posibilidad de revalorizar el medio cuya depreciación haya caducado y, en caso de no revalorizar, admite que el medio no siga depreciándose, informa cuándo la depreciación de cada medio básico este alcanzando su fin y posee una base de datos pasiva con los medios básico que se le haya dado de baja por 5 años. [19]

ASSETS

Es un Sistema de Gestión Integral estándar que permite controlar las operaciones de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos. Fue desarrollado en Visual Basic 6.0 y Microsoft SQL Server 2000, utilizando adicionalmente Crystal Reports 7.0 para la generación de reportes. El Módulo de Inventario permite observar las existencias, disponibilidad y reserva de los diferentes elementos en múltiples almacenes al realizar conteos físicos de los productos y hacer ajustes tanto de existencia, como de precios de costo de los artículos almacenados.

Características

- Como Sistema Integral todos sus módulos trabajan en estrecha relación, generando, automáticamente, al módulo de contabilidad los comprobantes de operaciones por cada una de las transacciones efectuadas, esto permite que se pueda trabajar bajo el principio de contabilidad al día.
- Es flexible, amigable, con ayuda en línea que puede ser instalado en una microcomputadora o sobre varias, funcionando en ambiente multiusuario incluidas estaciones remotas.
- Proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario.
- Facilita el uso de la parametrización para adaptarse a las exigencias de cada entidad en particular, garantizando que sus reportes tengan la forma y el contenido que el usuario les defina.

Este sistema, aunque es eficiente no es cubano, aunque tiene licencia, no es gratuita y se debe pagar regularmente. En un futuro sería muy conveniente crear un sistema integral propio de la Universidad para no tener que depender de entidades externas. [20]

I.2.3 Evaluación de sistemas existentes

Durante la investigación fue encontrada la tesis de investigación “Sistema para la gestión y control de procesos en el departamento de Tecnología de la CITEC” realizada en la Universidad de Ciencias Informáticas (UCI). [21]

Esta tesis propone un sistema web enfocado en la informatización de los procesos en los laboratorios y locales, que trae como beneficio la seguridad y control de los medios, control del registro de entrada y salida del personal en los laboratorios, seguimiento del proceso de entrega de medios en locales y seguimiento a los grupos de guardia en los laboratorios entre otros beneficios.

A pesar de que la solución no fue desarrollada, y solo se diseñó la arquitectura, se tendrá en cuenta en la presente investigación puesto que esta propuesta está dirigida al Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC), teniendo en cuenta esto último se referenciarán fundamentos teóricos, análisis, diseños entre otros como los requisitos no funcionales, ya que ambos trabajos investigativos tienen características similares.

Nombre	Infraestructura	Costo	Cantidad de Activos Fijos	Alojamiento	Documentación	Funcionalidades	Notas
Cetaris Fleet	Desktop	Pago	-	Local/Nube	Si	Gestión de pedidos de trabajo, piezas, ciclo de vida de los recursos. Historial de recursos. Seguimiento de Inventario.	No es Open Source
Upkeep	Web	Versión Gratis y de Pago	-	Local/Nube	Si	Gestión de activos fijos, artículos, inventarios, pedidos de trabajo, mantenimiento de equipos.	La versión gratis es solo online.
Asset tiger	Web	Versión Gratis	250	Nube	Si	Gestión de activos fijos, artículos, inventarios, pedidos de trabajo, mantenimiento de equipos.	Solo maneja 250 activos, para poder

							manejar más hay que pagar.
DOCPR O AVILA	Web	Pago	-	Local	Muy poca	Gestión, tramitación y resguardo de la información	Muy poca información.
Mebus	Desktop	Pago	-	Local	Muy poca	Gestión de información, activos fijos y Valorización de los activos.	Está enfocad o para el servicio de salud del país y hay muy poca información.
Assets	Desktop	Pago	-	Local/Nu be	Poca	Gestión de varios módulos entre ellos inventario y activo fijos.	Sistema con una gran cantida d de módulos, aunque tiene licencia no es gratuito.
Propues	Web	Gratis	-	Local	Si	Gestión y control de	Sistema

<p>ta de Trabajo Investigativo</p>						<p>procesos del Dep. de Tecnología.</p>	<p>que solo llego hasta la fase de Diseño de la Arquitectura</p>
---	--	--	--	--	--	---	--

Tabla 1: Comparación de los Sistemas Web de Gestión de Información.

Después de investigar los Sistemas Web de Gestión de Información anteriores y realizar una comparación de los mismos en la Tabla 1, se decidió no utilizar ninguno de los analizados, debido a que no cumplen con las características básicas que se deben cumplir para poder ser implementado en el Departamento de Tecnología, como, la cantidad de activos fijos manejados, el alojamiento en un servidor local y que sea gratis y *OpenSource*.

Muchos de estos Sistemas Web son integrales y gestionan una variedad de herramientas que para el Departamento no es de interés. Para el desarrollo del sistema web de gestión de información propuesto se utilizarán las siguientes características de los sistemas web anteriormente estudiados:

- Se utilizará una interfaz sencilla como la de AsserTiger, donde se destaca la facilidad para visualizar los activos fijos tangibles, los expedientes técnicos y los movimientos, centralizándolos en una sola vista a cada uno de ellos.
- La facilidad de búsqueda de AvilaDocPro, donde utilizando una vista dinámica del lado del cliente permitirá al usuario ejecutar búsquedas en tiempo real.
- La estrecha relación entre los módulos como Assets, donde esta característica se evidenciará entre los módulos Gestionar Activo Fijo Tangible y Movimientos de Activos Fijos Tangibles.

I.3 Tecnologías de desarrollo de software

Se define como tecnología de desarrollo de software a un conjunto integrado de notaciones, herramientas y métodos, que permiten el desarrollo de un producto software en un contexto organizativo dado. Una tecnología de software es la base que posibilita el desarrollo de sistemas de software capaces de brindar una solución adecuada desde el punto de vista técnico. [22]

A continuación, se analizarán las principales herramientas y tecnologías que se utilizarán en el sistema web para la gestión de información en el departamento de tecnología de la facultad de Ciencias y Tecnologías Computacionales.

I.3.1 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un marco de trabajo eficiente que surgió en la década de los años 70, pues ofrecían una respuesta a los problemas que surgían con los antiguos métodos de desarrollo.

Con la implementación de estas, se ha logrado mejorar de manera significativa las aplicaciones web, debido a que proveen una guía compuesta por etapas y procesos efectivos que permiten obtener mejores resultados y de calidad.

Son muchas las metodologías que existen como RUP, XP, SCRUM, AUP entre otras como OPEN UP. La metodología AUP-UCI es una variante de AUP adaptada a lo que la Universidad UCI ha estado proponiendo como ciclo de vida de los proyectos. La investigación actual se ajusta a del escenario No.2 de esta metodología. Esta Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades.

Se escoge esta metodología ya que:

- Se adapta al ciclo de vida definido para la actividad productiva de la UCI.
- El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.
- Describe de manera simple la forma de desarrollar el sistema.
- Su uso permite una descripción concisa utilizando poca documentación.

- Brinda la posibilidad de utilizar cualquier conjunto de herramientas CASE, apoyándose en un ciclo de vida, para los proyectos marcado por tres fases: inicio, ejecución y cierre.

I.3.2 Lenguaje de Modelado Unificado (UML v.2.1)

El lenguaje de modelado se utiliza en combinación con una metodología de desarrollo de software, proporciona terminologías para permitir el correcto modelado de cada uno de los objetos. En este caso se centra en la representación gráfica de un sistema. Provee características para modelar un sistema de software a nivel arquitectónico. Para la presente investigación se tomó el lenguaje de modelado UML

“El lenguaje unificado de modelado (UML) es un lenguaje estándar de modelado para software, un lenguaje para la visualización, especificación, construcción y documentación de los artefactos de sistemas en los que el software juega un papel importante. Básicamente UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados.” [22]

Herramienta de modelado

Las herramientas CASE (*Computer Aided Software Engineering*, en español Ingeniería de Software Asistida por Computadora.) pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso para realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. [22]

Visual Paradigm CE (v.17.0)

Debido a la experiencia del equipo de desarrollo y a su fácil manejo, la herramienta CASE que se selecciona en el modelado del sistema usando el lenguaje UML anteriormente explicado, es:

Visual Paradigm CE v.17.0 la cual propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas.

Constituye una herramienta disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

I.3.3 Lenguajes y tecnologías de programación

Se determinó desarrollar un sistema web debido a las necesidades del cliente, las cuales requerían versatilidad en el movimiento de datos, y los sistemas web tienen las siguientes ventajas que se acoplan a las características del Departamento de Tecnología:

- La Compatibilidad multiplataforma que comprenden los sistemas web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables.
- Los sistemas web exigen menos requerimientos de memoria, dejando más espacio para correr múltiples aplicaciones al mismo tiempo sin incurrir en frustrantes deterioros en el rendimiento.
- Estas aplicaciones presentan menos Bugs (terminología informática para catalogar errores de programación) y deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno.
- Todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos. Esta es la razón por la cual las aplicaciones basadas en web deberían tener mucho menos bugs que el software de escritorio descargable tradicional.

Combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces. Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más. Su principal objetivo es permitir la construcción de sitios web *responsive* para dispositivos móviles. [23]

JavaScript (v. ECMAScript 2021)

Como lenguaje de programación del lado del cliente es importante JavaScript en la implementación debido a que es un lenguaje sencillo, de programación muy liviana. Utiliza poca memoria, sin embargo, tiene gran variedad de efectos visuales. Tiene fácil manejo de datos. Es ligero de carga y muy fácil de integrar. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. [24]

PHP (v.8.1.0)

El lenguaje PHP v.8.1.0 es de programación de uso general, de código del lado del servidor, originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con una funcionalidad de procesador de PHP que genera la página web resultante. La elección del mismo está determinada por seleccionar a Laravel como *framework* de desarrollo en el servidor, el cual se encuentra escrito en dicho lenguaje. [25]

HTML (v.5.0)

Las siglas de HTML v.5 hacen referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros.

La selección de HTML5 como el lenguaje para realizar el maquetado de la propuesta de solución está sustentada en las novedosas Herramientas que incorpora para el desarrollo web y el hecho de ser este un estándar establecido por la *Wide Web Consortium* (W3C). [26]

CSS (v.3.0)

El lenguaje CSS v.3 es usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Es el más conocido y utilizado para definir las propiedades de formato de los diferentes elementos HTML. [27]

I.3.4 Frameworks de desarrollo

Un *framework* es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software. Sirve para acometer un proyecto en menos tiempo, y en el sector de la programación, con un código más limpio y consistente, de

manera rápida y eficaz. Ofrece una estructura base que los programadores pueden complementar o modificar según sus objetivos.

El uso de *frameworks* permite, principalmente, agilizar procesos de desarrollo porque podemos reutilizar herramientas o módulos: ya tienes el 'esqueleto' sobre el que trabajar. El hecho de escribir código o desarrollar una aplicación más fácilmente, sirve para tener una mejor organización y control de todo el código elaborado, pudiendo usarlo nuevamente en el futuro.

- Se puede reutilizar el código tantas veces como sea necesario. Asimismo, puede ser optimizado, con todas las ventajas que ello conlleva.
- También se puede afrontar tareas propias de programación de forma automatizada, lo que aumentará la velocidad (o la del equipo de trabajo) a la hora de programar.

Reducir tiempos implica una mayor productividad, del mismo modo que reutilizar recursos te lleva a minimizar riesgos. Por ello, usar uno o varios *frameworks* ya sea para la programación *Backend* (es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione) o *Frontend* (es la parte del desarrollo web que se dedica a la parte frontal de un sitio web, en pocas palabras al diseño de un sitio web) supone una gran ayuda para programadores y desarrolladores, ya que facilita sus tareas de forma considerable. [28]

La Figura #1 muestra una comparación de los *framework Backend* más usados hasta enero del 2022:

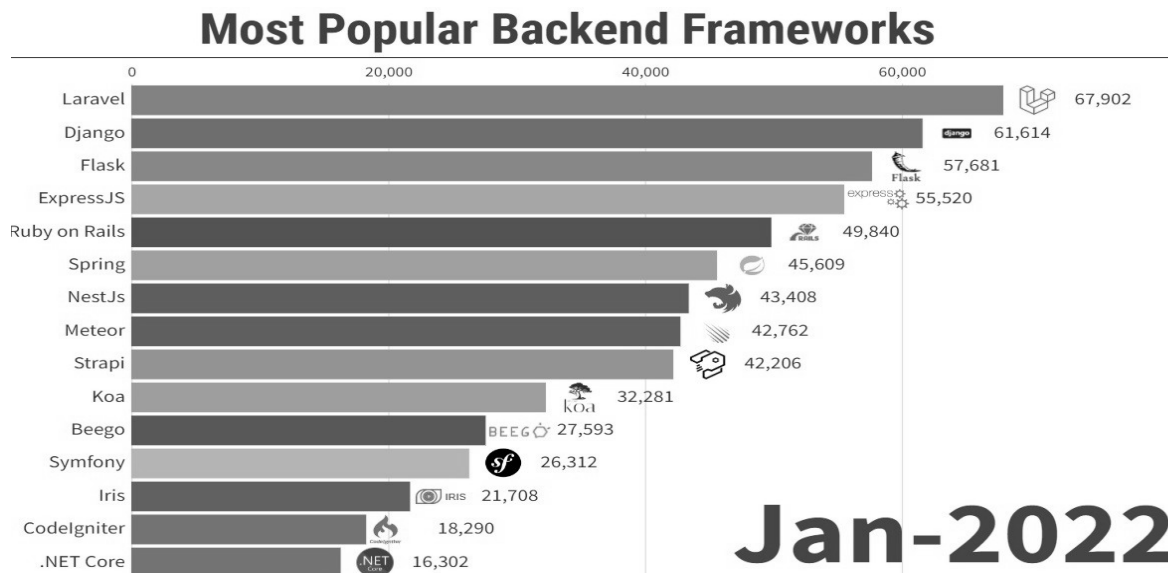


Figura 1: Frameworks más populares hasta enero de 2022.

Laravel (v.9.2)

Para la solución del problema se utilizarán Laravel v.9.2, este *backend framework* permite desarrollar infinidad de aplicaciones web con un alto nivel de personalización y calidad. Es moderno y ofrece multitud de utilidades a los desarrolladores. Por esto es uno de los más utilizados del mundo.

Se puede afirmar que Laravel pone énfasis en la calidad del código. En palabras de sus propios creadores: Laravel ofrece una sintaxis elegante y expresiva, que te permitirá escalar fácilmente todo tipo de desarrollos. Está hecho con arquitectura Modelo-Vista-Controlador y se centra sobre todo en necesidades de desarrollo actuales, como son la gestión de eventos y las autenticaciones de usuarios. Entre las características más importante están:

- Motor de plantillas ligeras e integradas, que permiten crear diseños increíbles utilizando siempre contenido dinámico. Además, proporciona multitud de widgets que incorporan CSS, y JS con estructuras súper robustas. Las plantillas de Laravel están muy bien diseñadas y el resultado es a la vez simple y complejo.
- Compatible con arquitectura MVC: Laravel es un *framework* compatible con patrones de arquitectura MVC. Esto garantiza la separación de la lógica empresarial por un lado y las capas de presentación por el otro. Este sistema es fácilmente escalable y ofrece multitud de funciones ya integradas.
- Una de las características que es más apreciada por los desarrolladores, es la incorporación de una herramienta de línea de comandos (llamada *Artisan*) que ayuda a automatizar las tediosas tareas de programación repetitivas.
- Laravel ofrece sus propias bibliotecas modulares de manera preinstalada en el sistema. Esto no es algo que suela ocurrir en otros *framework* de PHP.
- Está pensado para resolver problemas concretos que son de mucha actualidad. Esto es una gran ventaja. Digamos que se centra en lo importante. [29]

Bootstrap (v.5.2.0)

Bootstrap es una colección gratuita de herramientas utilizadas para crear sitios y aplicaciones web. Contiene plantillas de diseño basadas en HTML y CSS para tipografía, formularios, botones, navegación y otros componentes de la interfaz, así como extensiones de JavaScript

opcionales., creado por Twitter y liberado en 2012. Desde entonces ha ganado muchos adeptos, hasta el nivel de contar con la mayor comunidad de Github del mundo. [30]

Se decidió usar Bootstrap 5 como *framework* de desarrollo, pues su uso permite que un sitio web sea adaptable a la pantalla del dispositivo con el que se accede, ya sea un ordenador, *tablet*, *smartphone*, televisión. Esto significa que tendremos una web sensible o adaptativa que crea un diseño limpio, intuitivo, usable y de poco peso, por lo que la carga de nuestra web será muy rápida.

Además, será muy cómodo, porque muchas de las funcionalidades que se necesitan ya están desarrolladas; y si no, se tiene acceso a una gran cantidad de documentación en varios idiomas y una comunidad que dará respuestas a todas las dudas que se presenten.

I.3.5 Sistema gestor de base de datos

Los Servidores de Bases de Datos, son programas que permiten organizar datos en una o más tablas relacionadas. Son utilizados en todo el mundo en una amplia variedad de aplicaciones. Las bases de datos están situadas en un servidor y se puede acceder a ellas desde terminales o equipos con un programa - llamado cliente- que permita el acceso a la base o bases de datos. Los gestores de base de datos de este tipo permiten que varios usuarios hagan operaciones sobre ella al mismo tiempo. [31]

MySQL (v.9.1)

Para solución del problema y por la gran compatibilidad que tiene con Laravel fue elegido MySQL v.9.1 como gestor de base de datos. El mismo cuenta con más de seis millones de instalaciones, es multihilo, multiusuario, rápido y muy robusto. Es *open source*, publicado bajo la licencia GPL (*GNU Public License*). MySQL es muy utilizado en los sistemas web. Tiene gran popularidad por estar grandemente ligado a PHP. Es capaz de funcionar sobre múltiples plataformas. Las características de MySQL más destacadas son:

- Mediante la verificación basada en el host y tráfico de contraseñas encriptado ofrece un sistema de contraseñas y privilegios seguro, denotando cualidades de seguridad altamente confiables.
- Es capaz de soportar gran cantidad de datos. [32]

I.3.6 Entornos de Desarrollo Integrado

Un ambiente de desarrollo integrado o entorno de desarrollo interactivo, IDE (*Integrated Development Environment*, en español Ambiente de Desarrollo Integrado), es una aplicación informática que proporciona servicios integrales para facilitarle al programador el desarrollo de software. [33]

Visual Studio Code (v.1.70.2)

Fue seleccionado como IDE el Visual Studio Code v.1.70.2 (VS Code, abreviado) por:

- Es un editor de código gratuito y de código abierto de Microsoft que ha ganado una enorme popularidad en los últimos años.
- Soporta la sintaxis PHP por defecto, y puedes descargar más extensiones de PHP del Mercado de Código VS para convertirlo en un editor avanzado de PHP que se adapte a tus necesidades específicas de codificación.
- Puedes personalizar cada aspecto de VS Code, desde los temas a los enlaces de claves hasta las integraciones y la funcionalidad.

Conclusiones del capítulo

En este capítulo se realizó un estudio y análisis de los Sistemas Web de Gestión de Información relacionados con activos fijos tangibles, donde se llegó a la conclusión de no utilizar ninguno de ellos, decidiéndose diseñar e implementar uno más adecuado para el Departamento de Tecnología utilizando algunas características de los sistemas web estudiados. Para ello se establecieron las herramientas y tecnologías que permitirán el diseño e implementación del Sistema web, se define AUP-UCI como metodología de desarrollo de software en su escenario 2, *Visual Paradigm* CE 17.0 como herramienta CASE y UML 2.5 como lenguaje de modelado. El lenguaje de programación a utilizar será PHP 8.1, HTML 5, CSS3 y JavaScript, con *VisualStudio Code* como IDE, como *framework* de desarrollo *Laravel* 9.0 y *Bootstrap* 5.2.0, como sistema gestor de base de datos *MySQL* 9.1.

CAPÍTULO II: ANÁLISIS Y DISEÑO

En este capítulo se describe la propuesta de solución para el problema de la investigación, describiendo los principales conceptos del entorno que serán objeto de análisis para la realización de la fase de análisis y diseño del Sistema para la gestión de información del Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC). Se hace el levantamiento de requisitos funcionales y no funcionales para darle cumplimiento al objetivo general de la investigación. Además, se muestran los diagramas de modelo conceptual, de caso de uso del sistema, de clase de diseño y de despliegue, por medio de los cuales se representa la estructura estática del sistema y los patrones de arquitectura y diseño utilizados.

II.1 Mapa Conceptual

Habiendo seleccionado la metodología de desarrollo de software AUP-UCI en su escenario 2, se hace necesario que se modele el modelo conceptual (MC).

Es una herramienta que posibilita organizar y representar, de manera gráfica y mediante un esquema, el conocimiento. Explica a los creadores los conceptos significativos en un dominio del problema, es el artefacto más importante a crear durante el análisis orientado a objetos.

Una cualidad esencial que debe ofrecer un modelo conceptual es que representa cosas del mundo real, no componentes del software. Es un diagrama estático donde no se define ninguna operación. [34] El modelo conceptual puede mostrar: conceptos, asociaciones entre ellos y atributos de conceptos.

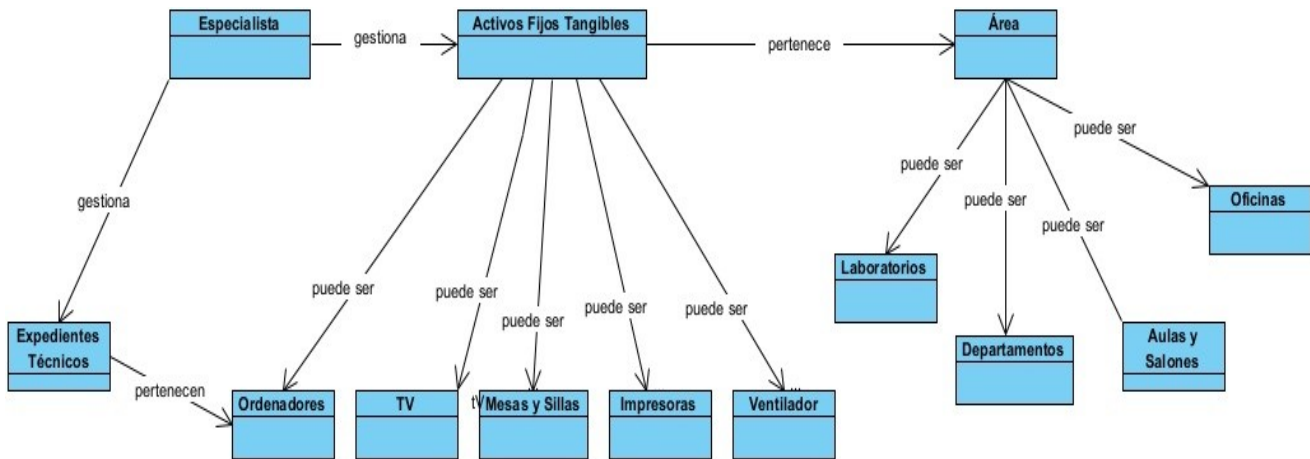


Figura 2: Mapa conceptual.

En la Figura 2 queda plasmado el modelo conceptual que será utilizado para el desarrollo del Sistema. Un especialista es el encargado de gestionar los activos fijos tangibles y los expedientes técnicos. Un activo fijo tangible puede ser un tv, o una mesa, o una silla, o una impresora, o un ventilador, o un ordenador. Los expedientes técnicos gestionados por el especialista pertenecen a los ordenadores. Cada activo fijo tangible pertenece a distintas áreas de la facultad las cuales pudieran ser los laboratorios, o departamentos, o aulas y salones, o las oficinas.

II.2 Requisitos del Sistema

Los requisitos son la base de todos los proyectos que involucren el desarrollo de software. Al acordar los requisitos para el sistema, se proporcionan las bases para la planificación del desarrollo de un proyecto de software. La ingeniería de requerimientos sirve para determinar las necesidades que un cliente tiene en la implementación o desarrollo de software, mediante el proceso de recopilar, analizar y verificar la especificación de los requisitos del software de una manera completa y correcta. [35]

Requisitos Funcionales

Los requisitos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa.

RF-1 Autenticar Usuario

Descripción: El Sistema debe permitir a los usuarios autenticarse y acceder al mismo.

RF-2 Registrar Usuario

Descripción: El Sistema debe permitir al Jefe de Departamento crear nuevos usuarios.

RF-3 Editar Usuario

Descripción: El Sistema debe permitir al Jefe de Departamento modificar los usuarios existentes.

RF-4 Eliminar Usuario

Descripción: El Sistema debe permitir al Jefe de Departamento eliminar usuarios existentes.

RF-5 Listar Usuario

Descripción: El Sistema debe permitir al Jefe de Departamento ver todos los usuarios existentes en una tabla.

RF-6 Registrar Rol

Descripción: El Sistema debe permitir al Jefe de Departamento crear nuevos roles.

RF-7 Editar Rol

Descripción: El Sistema debe permitir al Jefe de Departamento modificar los roles existentes.

RF-8 Eliminar Rol

Descripción: El Sistema debe permitir al Jefe de Departamento eliminar roles existentes.

RF-9 Listar Rol

Descripción: El Sistema debe permitir al Jefe de Departamento ver todos los roles existentes en una tabla.

RF-10 Asignar Rol

Descripción: El Sistema debe permitir al Jefe de Departamento asignar un role a los usuarios existentes.

RF-11 Registrar Área

Descripción: El Sistema debe permitir al Jefe de Departamento crear nuevas Áreas.

RF-12 Editar Área

Descripción: El Sistema debe permitir al Jefe de Departamento modificar las Áreas existentes.

RF-13 Eliminar Área

Descripción: El Sistema debe permitir al Jefe de Departamento eliminar las Áreas existente.

RF-14 Listar Área

Descripción: El Sistema debe permitir al Jefe de Departamento ver todas las Áreas existentes en una tabla.

RF-15 Registrar Local

Descripción: El Sistema debe permitir al Jefe de Departamento crear nuevos Locales.

RF-16 Editar Local

Descripción: El Sistema debe permitir al Jefe de Departamento modificar los Locales existentes.

RF-17 Eliminar Local

Descripción: El Sistema debe permitir al Jefe de Departamento eliminar los Locales existente.

RF-18 Listar Local

Descripción: El Sistema debe permitir al Jefe de Departamento ver todos los Locales existentes en una tabla.

RF-19 Registrar AFT

Descripción: El Sistema debe permitir al Especialista A crear nuevos AFT.

RF-20 Editar AFT

Descripción: El Sistema debe permitir al Especialista A modificar los AFTs existentes.

RF-21 Listar AFTs

Descripción: El Sistema debe permitir al Especialista A ver todos los AFTs existentes en una tabla.

RF-22 Buscar AFTs

Descripción: El Sistema debe permitir al Especialista A buscar un AFT existente en la tabla.

RF-23 Registrar Movimiento de ATF

Descripción: El Sistema debe permitir al Especialista A registrar nuevos Movimientos de ATF.

RF-24 Editar Movimiento de ATF

Descripción: El Sistema debe permitir al Especialista A modificar los Movimientos de ATF existentes.

RF-25 Listar Movimiento de Medio AFT

Descripción: El Sistema debe permitir al Especialista A ver todos los AFT existentes en una tabla.

RF-26 Mostrar los detalles del Movimiento de AFT

Descripción: El Sistema debe permitir al Especialista A ver los detalles de los Movimientos de Medios Básicos existente.

RF-27 Exportar PDF Movimiento de AFT

Descripción: El Sistema debe permitir al Especialista A exportar los detalles de los Movimientos de Medios Básicos existentes en formato PDF.

RF-28 Buscar Movimiento de Medio AFT

Descripción: El Sistema debe permitir al Especialista A buscar un movimiento de algún AFT existente en la tabla.

RF-29 Aprobar Movimiento de AFT

Descripción: El Sistema debe permitir al Jefe de Departamento aprobar los Movimientos de los AFTs existente.

RF-30 Registrar Expediente Técnico

Descripción: El Sistema debe permitir al Especialista B crear nuevos Expedientes Técnicos.

RF-31 Editar Expediente Técnico

Descripción: El Sistema debe permitir al Especialista B modificar los Expedientes Técnicos existentes.

RF-32 Listar Expedientes Técnicos

Descripción: El Sistema debe permitir al Especialista B ver todos los Expedientes Técnicos existentes en una tabla.

RF-33 Buscar Expediente Técnico

Descripción: El Sistema debe permitir al Especialista B buscar un Expedientes Técnico de existente en la tabla.

RF-34 Mostrar los detalles del Expedientes Técnicos

Descripción: El Sistema debe permitir al Especialista B ver los detalles de los Expedientes Técnicos existente.

RF-35 Exportar PDF Expedientes Técnicos

Descripción: El Sistema debe permitir al Especialista B exportar los detalles de los Expedientes Técnicos existentes en formato PDF.

Requisitos No Funcionales

Los requisitos no funcionales manifiestan aquellas cualidades que debe poseer el producto y las características para que sea atractivo, confiable, usable y seguro.

Usabilidad

RNF-1. Finalidad

El objetivo que persigue el sistema es que los especialistas puedan tener una herramienta que les permita gestionar y controlar los procesos que se desarrollan en el departamento.

Requisitos del Software

RNF-2 Servidor de aplicaciones para instalar la aplicación

La máquina donde se instalará la aplicación debe cumplir con los siguientes requisitos de software:

- Usuario con privilegios de administración del SO.
- Tener instalado un servidor web, por ejemplo, Apache.
- Tener instalado Laravel en su versión 9.2 o posterior y Node JS en su versión 16.2 o posterior, para los mantenimientos o actualizaciones futuras que pueda necesitar el sistema.
- Tener instalado PhpMyAdmin en su versión 4.8.3 o posterior.
- Tener instalador MySQL en su versión 5.5 o posterior.

RNF-3 Las PC clientes deben cumplir con los siguientes requisitos de software:

- El cliente debe poseer un navegador web para acceder al sistema desde su estación de trabajo, por ejemplo, Mozilla, Chrome o Edge sin importar la versión.

Requisitos de Hardware

RNF-4 La PC cliente es recomendable que cumpla con los siguientes requisitos mínimos que se especifican a continuación: una Tarjeta de Red, una RAM de 2GB o superior, un procesador a 2GHz o superior.

RNF-5 La PC servidor es recomendable que cumpla con los siguientes requisitos mínimos que se especifican a continuación: una Tarjeta de Red, una RAM de 4GB o superior, un procesador a 2GHz o superior y 40GB de disco duro.

Restricciones de Diseño e Implementación

RNF-6: Para la modelación se utilizará como lenguaje de modelado UML 2.5 y como herramienta el *Visual Paradigm CE for UML 17*.

Interfaz

RNF-7: Las interfaces de usuario deben ser diseñadas de forma tal que brinden al usuario la posibilidad de saber dónde y que acción está realizando, así como mostrar la información de forma clara.

II.3 Descripción del Sistema

La propuesta de solución tiene como propósito mejorar el proceso de gestión de información de los AFT y el control de los expedientes técnicos y movimientos de medios básicos en las diferentes Áreas de la Facultad de Ciencias y Tecnologías Computacionales (CITEC) de la UCI, tareas que son gestionadas por el Departamento de Tecnología, obteniéndose así un mejor control de estos medios básicos. Los roles que tienen actividad dentro del sistema web están en correspondencia con los cargos que se puedan ocupar en el Departamento de Tecnología, estos son: jefe de Departamento y Especialista.

Inicialmente el sistema informático cuenta con un usuario llamado jefe de Departamento en su configuración predeterminada. Este usuario es el encargado de realizar el primer acceso al sistema informático y configurar todo el espacio de trabajo. El jefe de Departamento tiene la responsabilidad de garantizar el acceso al sistema, a través de la gestión de usuarios, roles y permisos. Además, tiene acceso a todas las funcionalidades del sistema sin restricción alguna. El usuario Especialista tiene acceso limitado solo a gestionar AFT, gestionar Expedientes Técnicos y Gestionar el movimiento de un medio básico.

El sistema podrá ser desplegado como un sistema centralizado en lugares donde exista la infraestructura tecnológica necesaria para conectar varias computadoras, ello no quita la posibilidad de usarlo como un sistema distribuido en computadoras que se encuentren aisladas sin conexión a una red.

Actores del Sistema

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor, significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea, un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio; o sea, ser instancias de un mismo actor.

Actores	Descripción
Jefe de Departamento	El Jefe de Departamento tiene la responsabilidad de garantizar el acceso al sistema, a través de la gestión de usuarios, roles y permisos. Además, tiene acceso a todas las funcionalidades del sistema sin restricción alguna.
Especialista A	Tiene acceso limitado solo a gestionar AFT y gestionar el movimiento de un AFT.
Especialista B	Tiene acceso limitado solo a gestionar Expedientes Técnicos.

Tabla 2: Descripción de los actores del sistema.

Diagrama de Caso de Uso del Sistema

Siguiendo lo establecido por la metodología AUP-UCI en su escenario 2, esta plantea que Proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con Casos de Uso del Sistema.

Un diagrama de casos de uso tiene la función de representar de forma gráfica cuales son las funcionalidades de un sistema y las interacciones con los usuarios. Para ello se vale de elementos como el actor, casos de uso, relaciones actor caso de uso y las relaciones entre casos de uso.

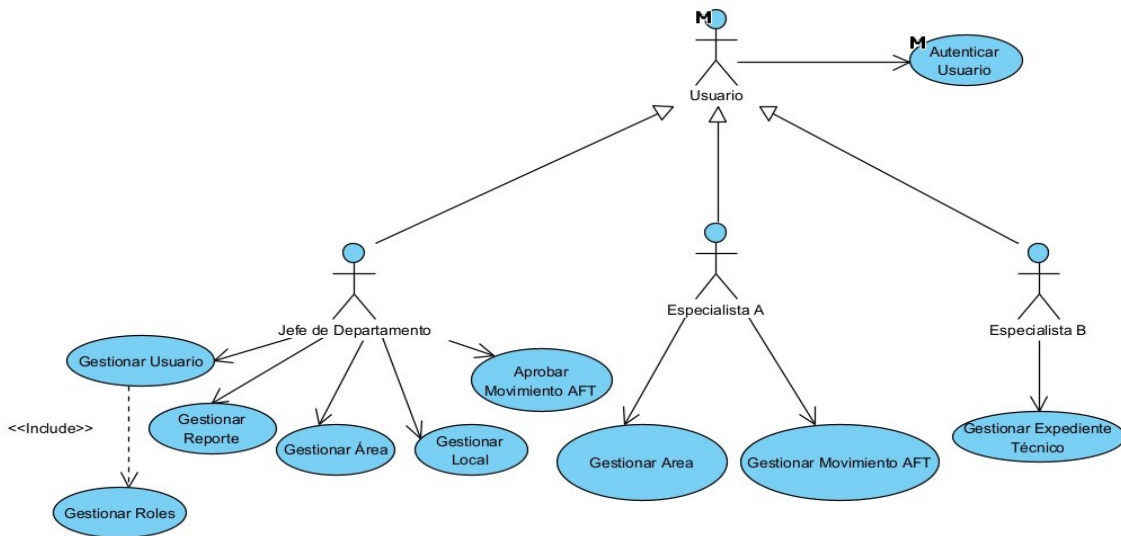


Figura 3: Diagrama de Caso de uso del sistema.

En figura anterior se muestra el diagrama de caso de uso del Sistema, donde se ven reflejado los casos de usos por lo que están compuesto el Sistema y su relación con los actores. Los actores Jefe de Departamento, Especialista A y Especialista B heredan del actor Usuario, el cual tiene el caso de uso autenticarse. El actor jefe de Departamento tiene los casos de usos Gestión de Reportes y Gestión de Usuarios al cual se le incluye el caso de uso Gestión de Roles. El Especialista A tiene el caso de uso Gestionar ATF del cual se extiende el caso de uso Movimiento de ATF. El actor Especialista B tiene el caso de uso Gestionar Expediente Técnico.

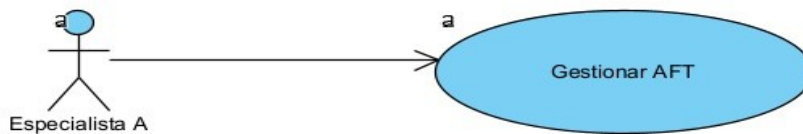


Figura 4: Caso de uso Gestionar ATF.

Para la modelación de este caso de uso se utilizó **el patrón de caso de uso CRUD**, este patrón se utiliza en los casos donde se requiere realizar altas, bajas, cambios y consultas a alguna entidad del Sistema en este caso la de ATF. En la tabla a continuación se describen cada uno de ellos.

Flujo	Descripción
Ver información de AFT	El caso de uso comienza cuando uno de los actores decide ver la información de algún AFT registrado, en la vista el Sistema recupera todos los AFT registrados y mediante la opción Buscar el actor puede encontrar el AFT a consultar. En la misma fila donde aparece el usuario el actor encontrará la tarea ver en forma de botón y accederá a la información pudiéndola consultar. El caso de uso termina.
Registrar AFT	El caso de uso comienza cuando uno de los actores decide registrar un nuevo AFT, en la vista del sistema se muestran varias tareas, entre las cuales aparece el botón llamado Nuevo donde el actor después de seleccionarlo deberá ingresar la información requerida. El Sistema comprueba que los datos son válidos y registra el nuevo AFT. El caso de uso termina. Este caso de uso queda representado en la figura 5, en un diagrama de secuencia.
Modificar AFT	El caso de uso comienza cuando uno de los actores decide modificar la información de algún AFT registrado, en la vista el Sistema recupera todos los AFT registrados y mediante la opción Buscar el actor puede encontrar el AFT a modificar. En la misma fila donde aparece el usuario el actor encontrará la tarea modificar en forma de botón y accederá a la información pudiéndola modificar. El sistema valida la información y guarda los cambios realizados. El caso de uso termina.

Tabla 3: Flujos básicos del Caso de Uso "Gestionar AFT".

II.4 Arquitectura del Sistema

En este epígrafe se expondrán los patrones de diseño y arquitectura, diagrama de clase del diseño y de despliegue, y el modelo de dato utilizados en el Sistema.

II.4.1 Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características, una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores, otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. [36]

Para el desarrollo de la propuesta de solución que se describe en la investigación solo fueron usados los implementados por el *framework*, por tal motivo serán los descritos a continuación.

II.4.1.1 Patrones GRASP

Son una colección de principios de diseño orientados a objetos que guían la asignación de responsabilidades sobre los objetos. GRASP es un acrónimo que significa *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades) El nombre se eligió para indicar la importancia de captar (*grasping*) estos principios, si se quiere diseñar eficazmente el software orientado a objetos. [37]

Experto

Este patrón resuelve el problema de “asignar una responsabilidad al experto en información -la clase que tiene la información necesaria para realizar la responsabilidad”.

Es utilizado en la capa de abstracción del modelo de datos. Con el uso del ORM Eloquent, Laravel genera automáticamente las clases que representan las entidades del modelo de datos. Asociado a cada una de estas clases son generadas un conjunto de funcionalidades que las relacionan de forma directa con la entidad que representan.

```
a make:model --all Generate a migration, seeder, factory, policy, and resou...  
a make:model -a Generate a migration, seeder, factory, policy, and resou...
```

Figura 5: Patrón experto.

Creador

Este patrón resuelve el problema de asignar responsabilidades relacionadas con la creación de objetos.

Es utilizado en los controladores, en ellos se encuentran las acciones definidas para el sistema. En la implementación de las acciones se crean instancias de las clases del modelo y de los formularios que representan a estas clases. En Laravel en la clase Controller.php se definen y ejecutan las acciones. En la implementación de este sistema al utilizar el componente *Livewire*, el mismo hace la función de Controller.php.

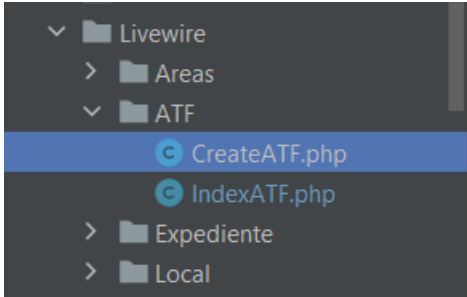


Figura 6: Patrón Creador.

Alta cohesión

Este patrón resuelve el problema de “asignar una responsabilidad de manera que la cohesión permanezca alta”. La cohesión es una medida de la fuerza con que se relacionan y del grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas, y que no hace una gran cantidad de trabajo, tiene alta cohesión.

La alta cohesión se evidencia en los controladores que poseen un conjunto de funcionalidades, existiendo estrecha relación entre algunas. Ejemplo de ello lo constituyen las acciones *create* y *update*, que al crear o actualizar un objeto, realizan las validaciones mediante la acción *validate*.

**Bajo
mien-**
Este
suelve
ma de
una
bilidad

```
public function saveEdit()
{
    $this->validate();
    $mov = Movimiento::find($this->selected_id);
    $area = Area::find($this->area_destino);

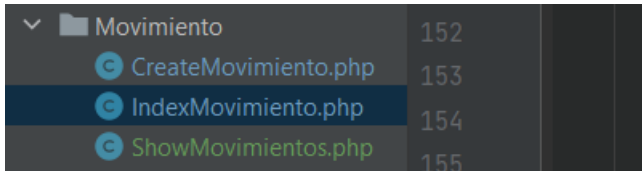
    $mov->update([
        'area_destino' => $area->name,
        'area_id' => $this->area_destino,
        'type_movimiento_id' => $this->type_movimientos,
```

acopla-
to
patrón re-
el proble-
“asignar
responsa-
de manera
acopla-

Figura 7: Patrón Alta cohesión.

miento permanezca bajo. Un elemento con bajo (o débil) acoplamiento no depende de demasiados elementos.

En la solución las clases `Create<nombre>.php`, `Edit<nombre>.php`, `Show<nombre>.php` heredan de `Index<nombre>.php`, que es una clase estable en cuanto a su implementación por lo que se obtiene un grado bajo de acoplamiento entre las clases.



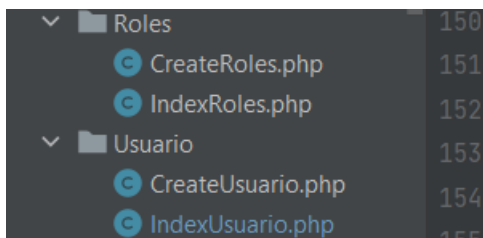
▼	▼	Movimiento	152
		○ CreateMovimiento.php	153
		○ IndexMovimiento.php	154
		○ ShowMovimientos.php	155

redan de `Index<nombre>.php`, que es una clase estable en cuanto a su implementación por lo que se obtiene un grado bajo de acoplamiento entre las clases.

Figura 8: Patrón Bajo acoplamiento.

Controlador

Este patrón resuelve el problema de asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase. Un controlador sirve como intermediario entre una interfaz y la acción que se desee ejecutar.



▼	▼	Roles	150
		○ CreateRoles.php	151
		○ IndexRoles.php	152
▼	▼	Usuario	153
		○ CreateUsuario.php	154
		○ IndexUsuario.php	155

En Laravel el patrón Controlador es utilizado en las clases de la capa del controlador del patrón MVC, como lo son los componentes *Livewire*: `IndexUsuario.php` e `IndexExpediente.php`.

Figura 9: Patrón controlador.

II.4.1.2 Patrones GOF

Los patrones *Gang of Four* por sus siglas en inglés se utilizan en situaciones frecuentes. Debido a que se basan en la experiencia acumulada al resolver problemas reiterativos. Ayudan a construir software basado en la reutilización, a construir clases reutilizables. [38]

Singleton

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=sgi
DB_USERNAME=root
DB_PASSWORD=
```

En ocasiones es necesario mantener una visibilidad global a una única instancia de una clase de tal forma que sea accedida en cualquier parte de la aplicación, este patrón garantiza el comportamiento descrito anteriormente. La conexión a la base de datos (BD) es un ejemplo de la aplicación de este patrón.

Figura 10: Patrón Singleton.

Decorador

Es un patrón de tipo estructura, ya que permite que clases y objetos sean utilizados para componer estructuras de mayor tamaño. Resuelve el problema de añadir responsabilidades adicionales a un objeto dinámicamente.

En Laravel este patrón es utilizado en la capa Vista del patrón arquitectónico MVC. En el componente se utiliza el patrón Decorador para la vista, el *layout* decora el contenido de la plantilla *Blade*.

```
return view( view: 'livewire.areas.index-area', compact( var_name: 'infos'))
->layout('components.layout', ['title' => 'Lista de Áreas', 'header' => 'Lista de Áreas']);
```

Figura 11: Patrón Decorador.

Observador

Un efecto muy frecuente en aquellos sistemas que se fragmentan en un conjunto de clases que cooperan, es la necesidad de mantener la consistencia entre los distintos objetos interrelacionados. Para no recurrir a soluciones fuertemente acopladas (que reducen la posibilidad de reutilización), este patrón define una dependencia “uno a muchos” entre objetos, para que, cuando uno de ellos cambie su estado, los que dependan de él sean avisados y puedan actualizarse convenientemente.

II.4.1.3 Patrones de Arquitectura

Involucrados en una arquitectura se encuentran los estilos arquitectónicos. Los diferentes es-

```
$table->foreign( columns: 'user_id')->references( columns: 'id')->on( table: 'users' )
->onUpdate( action: 'cascade');
```

Figura 12: Patrón Observador.

tilos tienen sus fortalezas y debilidades, y ciertos estilos hacen que sea más fácil o más difícil trabajar con diferentes obstáculos.

“Un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software.” [39]

Existen numerosos estilos arquitectónicos entre los que se encuentra el Estilo de llamada y retorno y dentro de él, el patrón de arquitectura Modelo-Vista-Controlador (MVC). Este patrón de arquitectura de software separa la lógica de control, la interfaz de usuario y los datos de una aplicación en tres componentes distintos. Se utiliza este patrón puesto que se tiene una

capa de acceso a datos donde van a estar todas las clases que de una forma u otra piden información a una Base de Datos.

El paquete Modelo contiene todas las clases que tienen el código relacionado con el acceso a datos, para que este sea lo más genérico posible y se pueda reutilizar en otras situaciones y proyectos. Se incluirá consultas a las bases de datos y validaciones de entrada de datos. Se evidencia dentro del modelo 9 clases donde una de las más importante es la clase principal, la cual contiene varios métodos que envían a ejecutar consultas SQL al gestor obteniendo y devolviendo los resultados de las mismas para luego ser procesados. [40]

El paquete Vista contiene todas las clases que poseen el código representando la parte que será visualizada en pantalla por el usuario. Se evidencia dentro de este paquete 11 clases donde la principal es editor- grafico, la misma es una interfaz visual que se le muestra al usuario para que el mismo interactúe con la aplicación. [40]

El paquete Controlador contiene las clases que ejecutan la lógica de la aplicación, realizan llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. Responden a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Se evidencia en este paquete la clase principal-interface donde esta clase obtiene los datos introducidos por el usuario en una de las vistas de la aplicación, los procesa pidiendo la información requerida a una de las clases del modelo para luego devolver y mostrar estos resultados al usuario. [40]

Con el uso de este patrón se persigue mejorar la reusabilidad y que las modificaciones en las vistas impacten en menor medida en la lógica de negocio o de datos. En la siguiente figura se ejemplifica de manera resumida el funcionamiento de la arquitectura MVC:

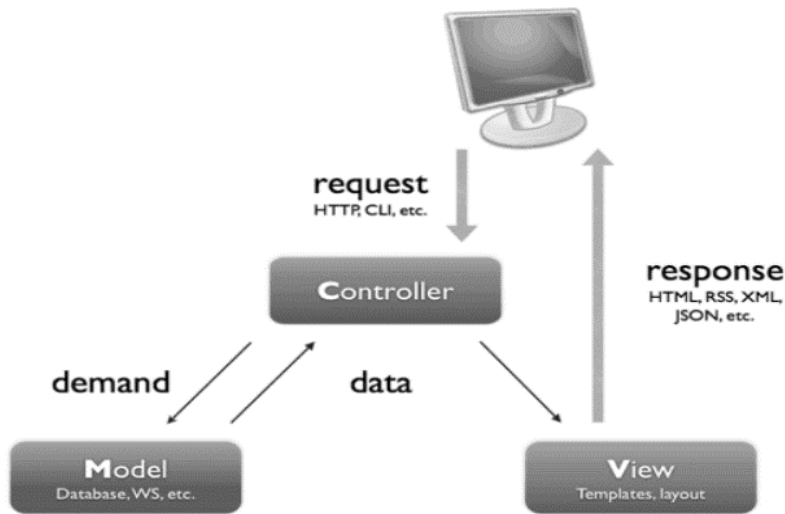


Figura 13: Arquitectura Modelo-Vista-Controlador.

II.4.2 Diagrama de clase del diseño

Los diagramas de clases del diseño son un tipo de diagrama estático que describe gráficamente la estructura de una aplicación, así como la especificación de sus clases. Permite en términos visuales la descripción del modelo, así como los atributos y comportamientos que poseen los objetos que intervienen en el diseño [41].

A continuación, en la siguiente figura se muestra una representación del diagrama de clases del diseño del caso de uso Gestionar AFT:

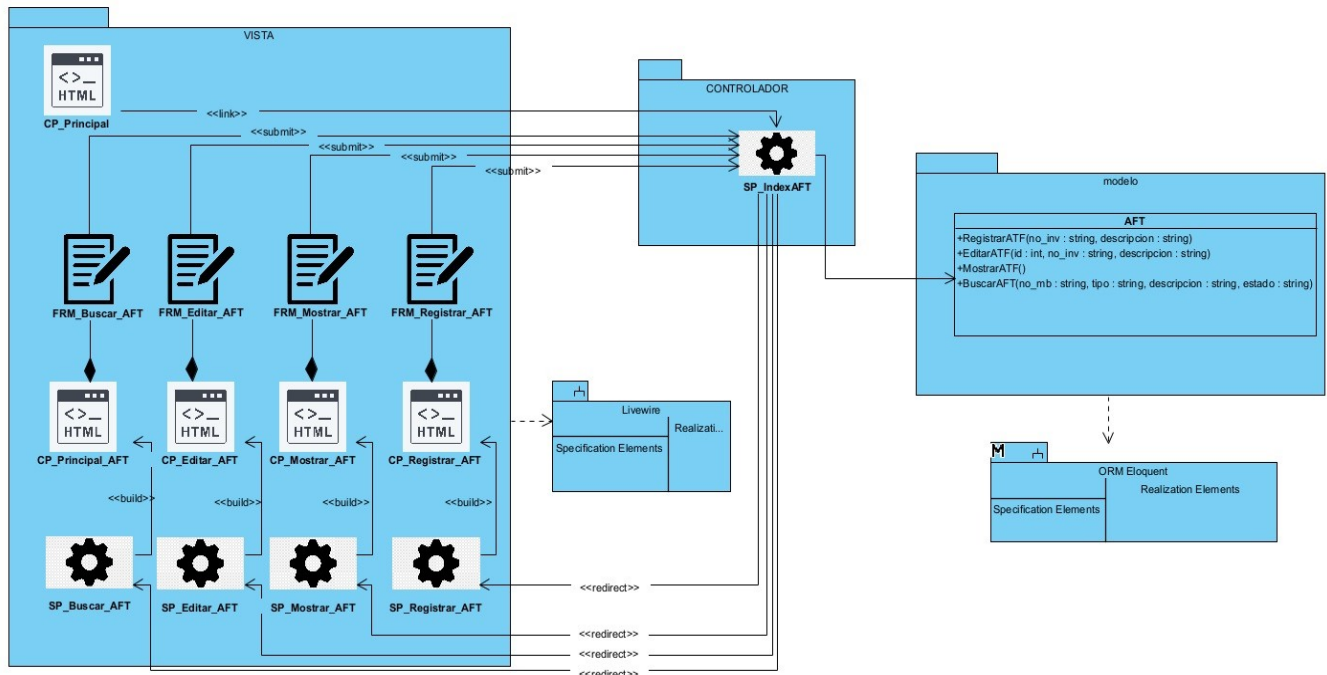


Figura 14: Diagrama de Clase de Diseño del CU "Gestionar AFT".

En la figura anterior queda representado el Diagrama de Clase de Diseño del CU "Gestionar AFT", donde siguiendo el patrón arquitectónico MVC utilizado en la solución planteada, se separa en 3 componentes la lógica de control, el interfaz de usuario y los datos del Sistema. En el componente Vista se representa la Client-Page Principal la cual es la primera interfaz que verá el usuario luego de loguearse en el Sistema y mediante un link esta lleva al usuario hacia la Server-Page Index_AFT la cual se encuentra en el componente Controlador y es la encargada de redireccionar hacia las otras Server-Page según la acción del usuario. Estas Server-Page son Registrar_AFT, Mostrar_AFT y Editar_AFT estas se encargan mediante un build de mostrar las Client-Page Registrar_AFT, Mostrar_AFT y Editar_AFT en formato HTML y están incluidos sus correspondientes formularios, que mediante un submit serán enviados los datos hacia el componente Controlador y este hacia el Componente Modelo, en este se encuentra el Modelo AFT, donde se encuentra los métodos y variables del CU, y donde mediante el ORM *Eloquent* los datos son guardados en la tabla de Activos Fijos.

II.4.3 Diagrama de Secuencia

Los diagramas de secuencia son una solución de modelado dinámico popular en UML porque se centran específicamente en líneas de vida o en los procesos y objetos que coexisten simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine.

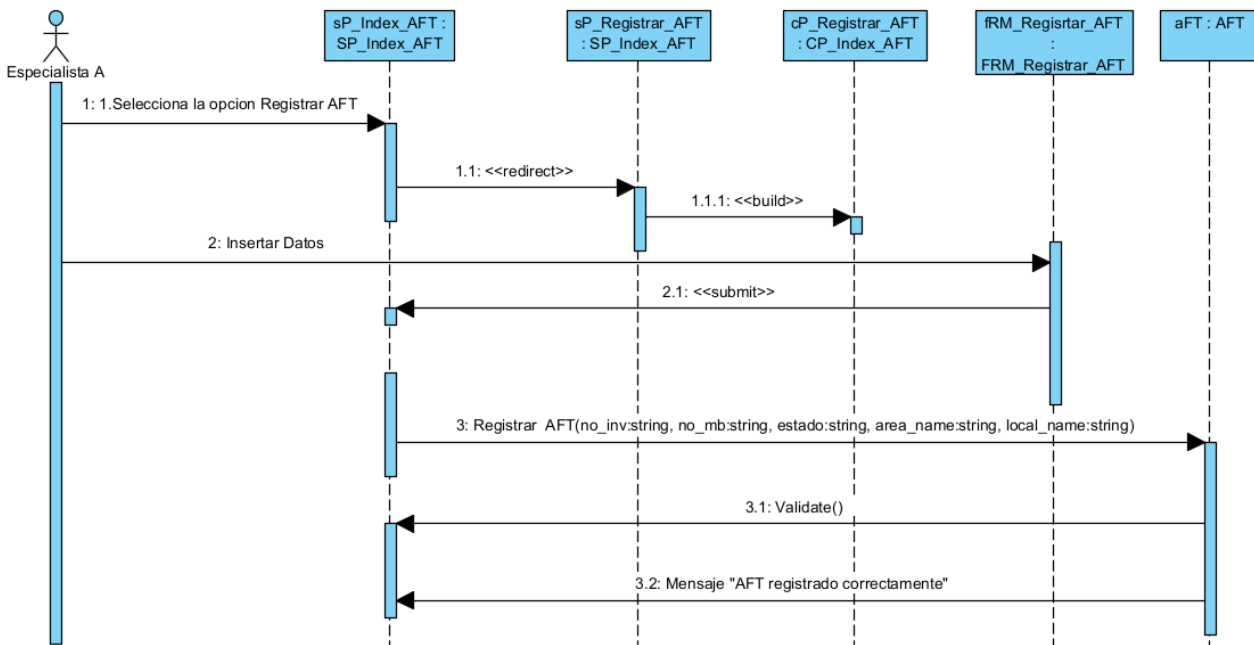


Figura 15: Diagrama de Secuencia del caso de uso Registrar AFT.

La Figura anterior muestra la secuencia de acciones que debe realizar el Especialista A para poder registrar un AFT y los componentes que intervienen.

II.4.4 Modelo de Datos

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades. [42]

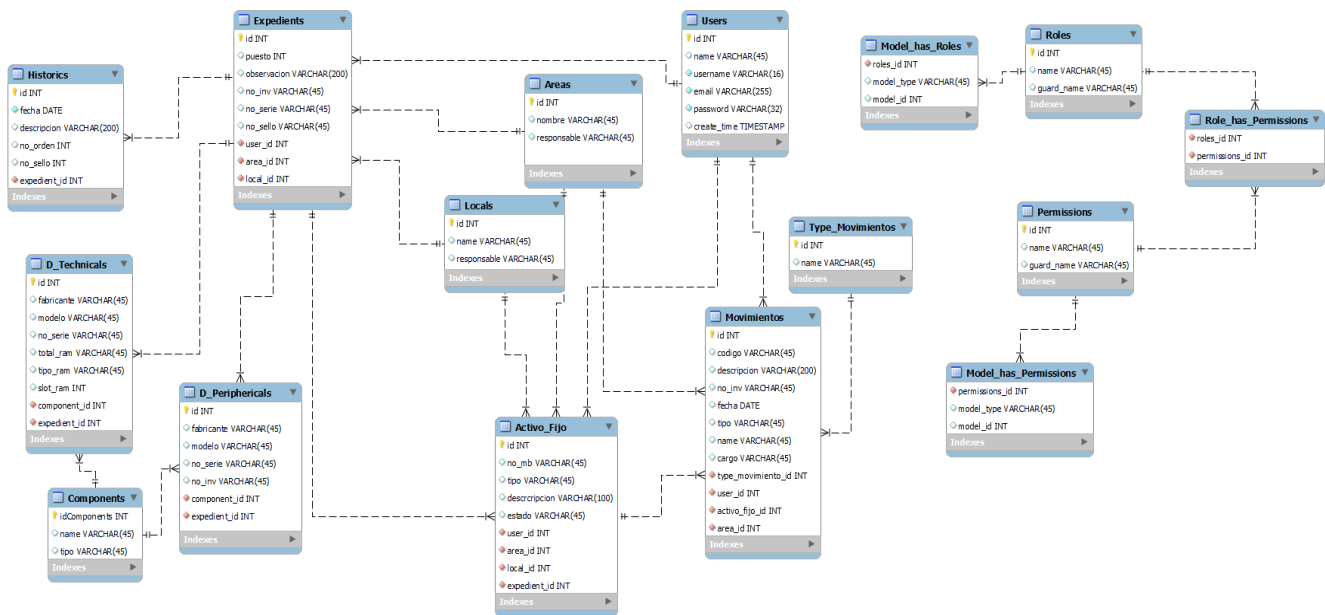


Figura 16: Modelo de Datos del Sistema.

En la figura anterior muestra el modelo de datos del Sistema. La tabla principal es Users, ya que está pensado para un área de la UCI no es conveniente utilizar el LDAP, la misma cuenta con los atributos name, username, email, password. Esta tabla tiene una relación de mucho a mucho polimórfica con la clase roles, teniendo como tabla pivote models_has_role, y la tabla Role tiene una relación de mucho a mucho polimórfica con Permission teniendo como tabla pivote models_has_permission. Estas tablas pertenecen al paquete Laravel Permission. La tabla Users tiene una relación de uno a muchos con la tabla Expedients, esta es la tabla principal del caso de uso Gestionar Expediente y a ella se relacionan varias tablas como Areas, Locales, Historics, D_Peripherals, D_Technicals (estas dos últimas tienen una relación inversa de uno a muchos con la tabla Components, esta guarda el nombre y el tipo de los componentes de las tablas relacionadas con ella) de las cuales todas tienen una relación de uno a muchos. Las otras tablas relacionadas con Users son Activo_Fijo y Movimientos. La tabla Activo_Fijos tiene una relación inversa de uno a muchos con Expedients y una relación de uno a muchos con Areas y Movimientos. La tabla Movimientos tiene una relación inversa de uno a muchos con la tabla Type_Movimiento.

II.4.5 Diagrama de Despliegue

El diagrama de despliegue describe la distribución física del sistema. Se utiliza como entrada principal en las actividades de diseño e implementación, debido a que la distribución del sistema tiene una influencia principal en su diseño. [43]

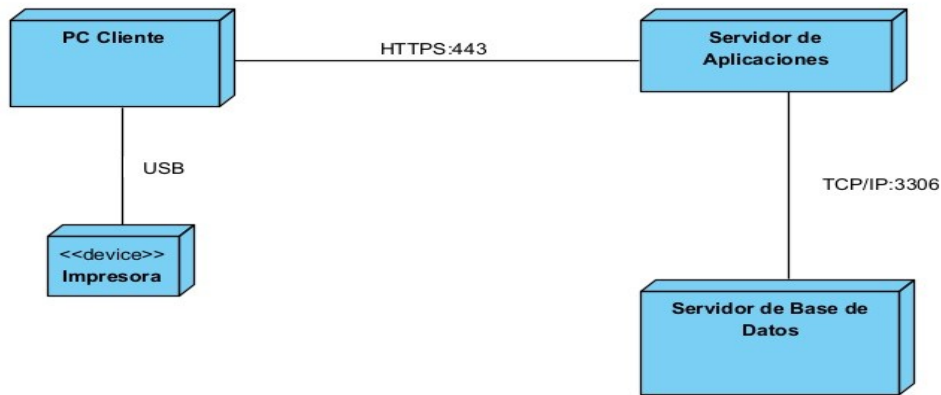


Figura 17: Diagrama de Despliegue.

El diagrama de despliegue anterior muestra la descripción física del Sistema, compuesto por una PC cliente la cual tiene una conexión con una impresora mediante el puerto USB, con el servidor de Aplicaciones mediante el protocolo HTTP (puerto 443) y este a su vez con el servidor de Base de Datos mediante el protocolo TCP/IP (puerto 3306).

Conclusiones del capítulo

En el presente capítulo que finaliza se abordaron los aspectos básicos relacionados con el análisis y diseño. Se definieron 35 requisitos funcionales en 7 casos de usos del sistema. Se definió el modelo conceptual, los diagramas de casos de uso del sistema, de clase de diseño, de entidad-relación, de despliegue. Se expusieron los actores del sistema. Se definieron los patrones de caso de uso, de diseño y los arquitectónicos utilizados en el sistema. Lo antes mencionado sentó las bases para la siguiente etapa de construcción del sistema de gestión de información en el Departamento de Tecnología.

CAPÍTULO III: IMPLEMENTACIÓN Y VALIDACIÓN

Como resultado de las fases anteriores se obtendrá un producto funcional para entregar al cliente, el cual debe garantizar los requisitos definidos con la calidad requerida. En el capítulo se aborda sobre las acciones que se llevan a cabo durante la fase de implementación y pruebas. Se analiza el modelo de implementación y se diseña y proponen las pruebas para comprobar el correcto funcionamiento del sistema web.

III.1 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. [44]

Diagrama de Componentes

Un diagrama de componentes muestra cómo un software es dividido en componentes, los artefactos por los que está compuesto como los archivos de código fuente, las librerías o las tablas de una base de datos y representa las dependencias entre estos componentes. Uno de los usos principales es que puede servir para ver como los componentes pueden compararse entre sistemas o bien entre diferentes partes de un sistema. [45]

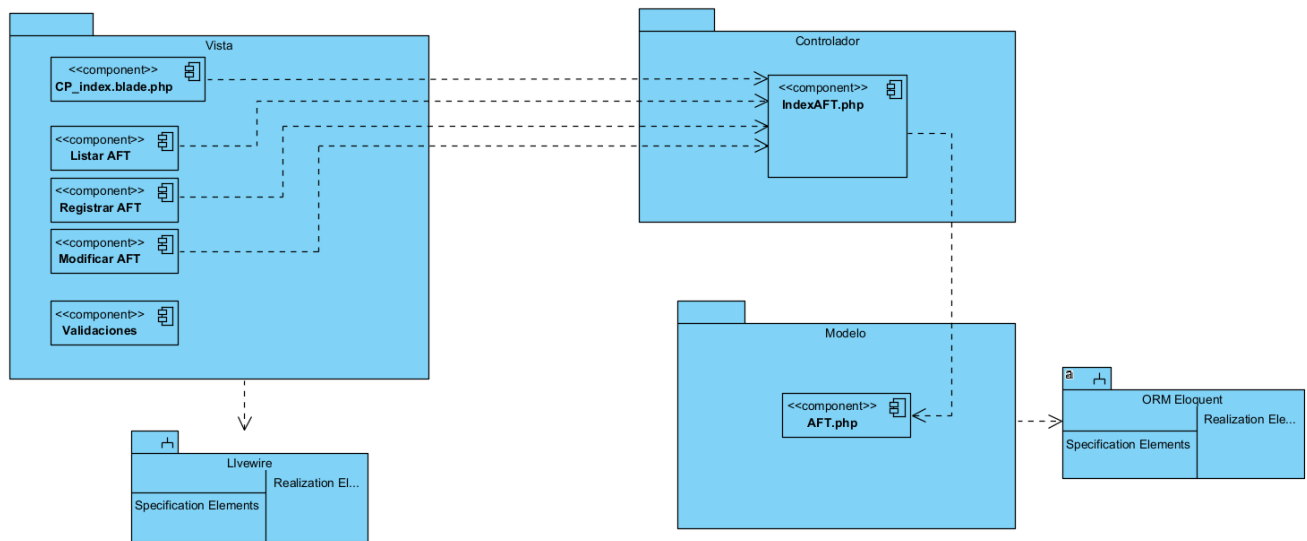


Figura 18: Diagrama de Componentes del CU “Gestionar AFT”.

III.2 Pruebas de Software

Las pruebas de software son un conjunto de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa; probando el comportamiento del mismo. Estas se ejecutan primeramente sobre un grupo de componentes y luego sobre el sistema como un todo, observando el comportamiento del software en un conjunto finito de casos de pruebas que como describe Somerville; son especificaciones de las entradas para la prueba y la salida esperada del sistema, más una afirmación de lo que se está probando. [46]

III.2.1 Pruebas unitarias

Corresponden a la validación de una pieza, componente, módulo o subprograma específico de un sistema para comprobar que cada una de esas partes funciona correctamente por separado.

Métodos de pruebas

Método de Caja Blanca

Se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. [47] Para la realización de este método se optó por la utilización de la técnica de Camino Básico.

Método de Caja Negra

Conocidas como *black box testing*, pueden definirse como una técnica donde se busca la verificación de las funcionalidades del software o aplicación analizada, sin tomar como referente la estructura del código interno, las rutas de tipo internas ni la información referente a la implementación.

Las pruebas de caja negra se caracterizan por su utilidad para la detección de errores de interfaz, de rendimiento, de inicio y terminación, así como funciones que no se ejecutan de manera correcta o que, directamente, no se ejecutan en el sistema. [47]

Técnicas de pruebas

Camino Básico

Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico.

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos inde-

pendientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática.

Los pasos que se siguen para aplicar esta técnica son:

- A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
- Se calcula la complejidad ciclomática del grafo.
- Se determina un conjunto básico de caminos independientes.
- Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Partición de Equivalencia: Esta técnica también se conoce como Partición de clases de equivalencia (ECP). En esta técnica, los valores de entrada al sistema o la aplicación se dividen en diferentes clases o grupos según su similitud en el resultado.

Por lo tanto, en lugar de usar todos y cada uno de los valores de entrada, ahora podemos usar cualquier valor del grupo/clase para probar el resultado. De esta manera, podemos mantener la cobertura de la prueba mientras podemos reducir una gran cantidad de re trabajos y, lo más importante, el tiempo invertido.

La siguiente tabla muestra el resultado de la prueba unitaria al fragmento del código de la clase RegistrarAFT.php, utilizando el método de Caja Blanca y la técnica Camino Básico, arrojando un error de validación en el campo area_id en el primer caso de uso de prueba y arrojando una evaluación satisfactoria en el segundo caso de uso de prueba:

Prueba unitaria	
Realizador: Javier Horsford Rivera	
<pre>public function save() { \$this->validate(); \$activo = ActivoFijo::create(['no_mb' => \$this->no_mb, 'tipo' => \$this->tipo, 'descripcion' => \$this->descripcion, 'estado' => \$this->estado, 'area_id' => \$this->area_name, 'local_id' => \$this->local_name,]); \$this->emit('save'); \$this->clearInput(); \$this->emit('alert', 'El Activo Fijo se creó correctamente'); }</pre>	<p>Representación del grafo de flujo</p> <pre>graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 2 --> 4((4)) 4 --> 5((5))</pre>

Complejidad ciclomática $V(G) = A - N + 2 = 5 - 5 + 2 = 2$	Caminos independientes 1,2,4,5 1,2,3,1,2,4,5
Caso de prueba 1 para el camino básico	
Descripción: No se acepta crear un AFT.	
Condición de ejecución: La acción tiene que ser cancelar.	
Datos de entrada	tipo, descripción, estado
Tipo de datos esperado	Error en la validación de la entrada de datos.
Evaluación del caso de Prueba: Satisfactoria.	
Caso de prueba 2 para el camino básico	
Descripción: Se registra correctamente un AFT.	
Condición de ejecución: La acción tiene que ser crear.	
Datos de entrada	
Tipo de datos esperado	Tiene que mostrarse el mensaje: "El Activo Fijo ha sido creado satisfactoriamente".
Evaluación del caso de Prueba: Satisfactoria.	

Tabla 4: Prueba unitaria: Técnica de camino básico.

III.2.2 Pruebas de aceptación

Según, el International Software Testing Qualification Board (ISTQB, por sus siglas en inglés) define la "Aceptación" como: Pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema. [46]

Tipos de pruebas

Pruebas de Funcionalidad

Las pruebas funcionales son pruebas diseñadas tomando como referencia las especificaciones funcionales de un componente o sistema (lo que se va a testear, el software o una parte de él). Se realizan para comprobar si el software cumple las funciones esperadas y se enfocan sobre un conjunto de pruebas que intentan descubrir errores en el mismo. [47]

Pruebas no Funcionales

Para la realización de las pruebas no funcionales se optó por aplicar la siguiente prueba que a continuación se describe:

Pruebas de Usabilidad

Se basan principalmente en determinar cuán bien el usuario podrá usar y entender la aplicación además de identificar las áreas de diseño que hacen al sistema de difícil uso para el usuario. [47]

III.3 Casos de pruebas

En ingeniería del software, un caso de prueba (en inglés, *test case*) es un conjunto de condiciones o variables bajo las cuales se determinará si una aplicación, un sistema de software o una característica o comportamiento de estos resulta o no aceptable. Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio.

Lo que caracteriza un escrito formal de caso de prueba es que hay una entrada conocida y una salida esperada, los cuales son formulados antes de que se ejecute la prueba. La entrada conocida debe probar una precondición y la salida esperada debe probar una postcondición. Los casos de prueba escritos, incluyen una descripción de la funcionalidad que se probará, la cual es tomada ya sea de los requisitos o de los casos de uso, y la preparación requerida para asegurarse de que la prueba pueda ser dirigida.

Se presenta el caso de prueba del requisito funcional Registrar Expediente Técnico utilizando del método de Caja Negra la técnica Partición de Equivalencia, donde se elige uno de los posibles escenarios que se pueden presentar durante la ejecución del requisito, detectando un error en el escenario EP2.2 del campo No. Inventario mostrado en la Tabla 5:

Nombre del requisito	Descripción general	Escenario de pruebas	Flujo del escenario
RF29 Registrar Expediente Técnico.	Permite crear un nuevo Expediente Técnico.	EP1: Adicionar Expediente Técnico de forma correcta.	<p>1- Se muestra una interfaz Lista de Expediente Técnico con el listado de los Expediente Técnico existentes.</p> <p>2- Se selecciona la opción Nuevo.</p> <p>3- Se introducen los siguientes datos: Área de Responsabilidad, Local, Puesto, Responsable, No. Inventario, No. Serie, No. Sello, los datos de los Componentes y las Observa.</p> <p>4- Se selecciona la opción Guardar.</p> <p>5- El sistema muestra un mensaje de notificación: "El Expediente Técnico ha sido creado satisfactoriamente".</p>
		EP1.2: Adicionar Expediente Técnico de forma incorrecta.	<p>1- Se muestra una interfaz Lista de Expediente Técnico con el listado de los dominios existentes.</p> <p>2- Se selecciona la opción Nuevo.</p> <p>3- Se introducen datos, dejando campos obligatorios vacíos.</p> <p>4- Se señalan los campos y se muestra el mensaje: "Este campo es requerido".</p>
		EP1.3: Cancelar Adicionar dominio.	<p>1- Se muestra una interfaz Lista de Expediente Técnico con el listado de los dominios existentes.</p> <p>2- Se selecciona la opción Nuevo.</p> <p>3- Se dejan en blanco los campos a introducir o se llenan los siguientes datos: Área de Responsabilidad, Local, Puesto, Responsable, No. Inventario, No. Serie, No. Sello, los datos de los Componentes y las Observa.</p> <p>4- Se selecciona la opción Cancelar.</p> <p>5- Se anulan las acciones y se cierra la interfaz, regresando a la interfaz Lista de Expediente Técnico.</p>

Tabla 5: Diseño de caso de prueba: RF32 Crear Expediente Técnico.

Nombre del requisito	Escenario	Descripción	Juego de datos	Resultados obtenidos	Observaciones
Registrar Expediente Técnico	EP2.1	Se adiciona un Expediente Técnico para mostrar si los campos No. Inventario, No. Serie, No. Sello, Área, Local y Observaciones aceptan campos vacíos.	Área: "vacío" Local: "vacío" No. Inventario: "vacío" No. Serie: "vacío" No. Sello: "vacío" Observaciones: "vacío"	El sistema muestra el mensaje: "El campo es obligatorio"	
	EP2.2	Se adiciona un Expediente Técnico para mostrar si los campos No. Inventario, No. Serie y No. Sello aceptan campos con valores ya existentes.	No. Inventario: "vacío" No. Serie: "vacío" No. Sello: "vacío"	El sistema muestra el mensaje: "El campo es obligatorio"	El error fue corregido satisfactoriamente. Para adicionar un dominio el campo de Nombre del Expediente Técnico no debe estar vacío.
	EP2.3	Se adiciona un Expediente Técnico con todos los campos existentes.	Todos los campos del formulario	El sistema muestra un error: "El valor del campo slot es integer"	El error fue corregido satisfactoriamente añadiendo la validación correspondiente.

Tabla 6: Diseño de prueba de validación RF29 Registrar Expediente Técnico.

Resultados de las pruebas de aceptación

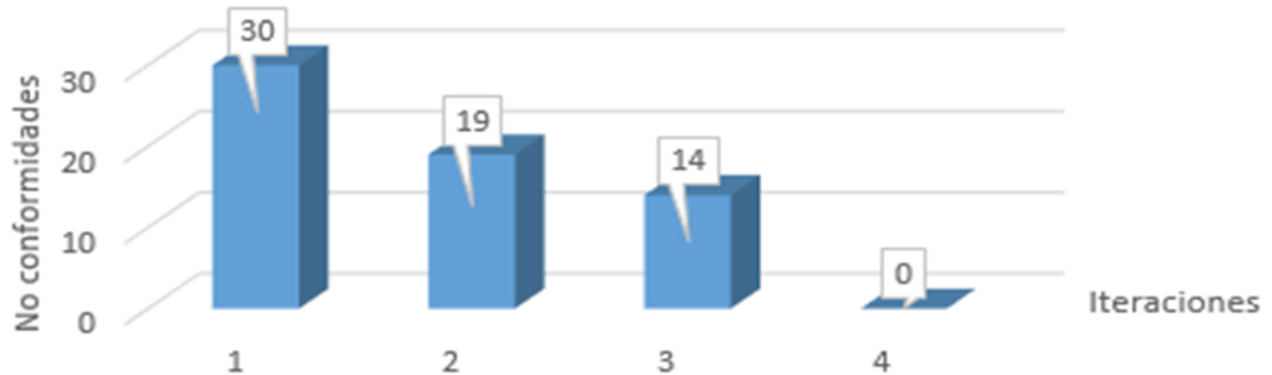


Figura 19: Resultado de las pruebas de aceptación.

Se realizaron cuatro iteraciones de pruebas al módulo de gestión del servidor de correo electrónico, haciendo uso de los Diseños de casos de prueba elaborados, con lo que se obtuvieron los resultados siguientes:

- En la primera iteración se detectaron 30 no conformidades, principalmente en campos de los formularios que aceptan entradas incorrectas o valores vacíos y algunos funcionales, por ejemplo, el botón Editar ATF no funcionan correctamente, la funcionalidad Nuevo Registro Histórico no listaba los registros históricos del expediente existente.
- Para una segunda iteración se resolvieron 17 no conformidades y se detectaron 6, 2 de aplicación, 1 de diseño y 3 de validación.
- Para una tercera iteración se resolvieron 5 no conformidades de validación.
- La cuarta iteración arrojó resultados satisfactorios, ya que se resolvieron las no conformidades restantes y no se encontraron nuevas.

Conclusiones del capítulo

En el desarrollo del presente capítulo se realizó el modelo de implementación del sistema con el propósito de mostrar los componentes del sistema y sus relaciones, a través del diagrama de componentes. Se realizaron las pruebas al sistema para evaluar su calidad, el caso de uso de prueba funcional utilizando el método Caja Negra arrojó un error el cual fue corregido, la prueba de aceptación en la primera iteración lanzó 30 no conformidades, la segunda 17 no conformidades, la tercera 6 no conformidades y la cuarta resultó satisfactoria, en la prueba caja blanca arrojó un error en el primer caso de prueba y el segundo caso de prueba resultó satisfactorio. Luego de realizar estas pruebas se obtuvo un producto libre de errores y listo para su ejecución. Se comprobó que el Sistema web para la gestión de información en el Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC) satisface las necesidades.

CONCLUSIONES GENERALES

- ✓ Se realizó un estudio y análisis de los Sistemas web de Gestión de Información relacionados con activos fijos tangibles, que permitió llegar a la conclusión de que era necesario diseñar e implementar un sistema que se ajustara a las necesidades del Departamento de Tecnología utilizando algunas características de los sistemas web estudiados.
- ✓ El análisis y obtención de los 35 requisitos funcionales y 7 no funcionales del Sistema web para la gestión de información en el Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC), permitieron el correcto diseño de las clases mediante la utilización de los patrones de diseño.
- ✓ La implementación de las funcionalidades del sistema web permitió obtener un sistema web que responde a las necesidades del cliente definidas durante la etapa de análisis.
- ✓ Las pruebas diseñadas y propuestas para ser aplicadas al sistema web garantizan el correcto funcionamiento del mismo.
- ✓ El desarrollo del Sistema web para la gestión de información en el Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales (CITEC) posibilitó un mayor control de los procesos y que los mismos puedan ser realizados de forma óptima y en el tiempo estimado.

RECOMENDACIONES

- ✓ Implementar un módulo para la gestión de útiles.
- ✓ Implementar un módulo para la gestión de los mantenimientos de los ordenadores en las diferentes áreas de la Facultad.

REFERENCIAS BIBLIOGRÁFICAS

1. Ledo, M. J. V., & Pérez, A. B. A. (2012). Gestión de la información y el conocimiento. Revista Cubana de educación médica superior, 26(3), 474-484. [en línea]. [visitado: 10 de junio 2022], Disponible en: <https://www.medigraphic.com/pdfs/educacion/cem-2012/cem123m.pdf>
2. Gestión de la Información versus Gestión del Conocimiento. Fernández Marcial, Viviana. Revista Investigación Bibliotecológica. Vol.20 No 41. México Jul/Dic 2006. [en línea]. [visitado: 10 de junio 2022], Disponible en: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0187-358X2006000200003
3. Bytheway, A. (2015). «Investing in Information: The Information Management Body of Knowledge» (en inglés). Geneva: Springer. p. 31.
4. Nurzman, Fahrul. Seminar Nasional" Data & Information Management, and Data Management Framework from Data Management Body of Knowledge (DMBOK)". 2021. [visitado: 10 de noviembre 2022] Disponible en: <http://repository.upi-yai.ac.id/4426/>
5. Carr, N. (2003). «Wringing Real Value from IT». HBR OnPoint. p. 3–10. [en línea]. [visitado: 10 de noviembre 2022] Disponible en: http://www.larryblakeley.com/Articles/monthly_articles/i_t_make_difference.pdf
6. Andreu, Rafael; Ricart, Joan; Valor, Josep. Estrategia y sistema de información. En Estrategia y sistema de información. 1991. p. 187-187.
7. Kaplan, Robert S.; Kaplan, Robert S.; P., Robert S. |Norton Kaplan (David; Norton, David P.; Norton, David P. (1996). The Balanced Scorecard: Translating Strategy into Action. Harvard Business Press. ISBN 9780875846514. [en línea]. visitado: [10 de noviembre 2022] Disponible en: https://books.google.com.ar/books/about/The_Balanced_Scorecard.html?id=mRHC5KHxczEC&redir_esc=y
8. Bytheway, A. (2015). «Investing in Information: The Information Management Body of Knowledge». Geneva: Springer. p. 31 visitado: [10 de noviembre 2022] Disponible en: <https://www.imbok.info/>
9. Scholl, Hans J. (2005). «E-Government-Induced Business Process Change (BPC): An Empirical Study of Current Practices». International Journal of Electronic Government Research (IJEGR) (en inglés) 1: 27-49. [en línea]. visitado: [10 de noviembre 2022] Disponible en: <https://web.archive.org/web/20180519121532/https://ideas.repec.org/a/igg/jegr00/v1y2005i2p27-49.html>
10. Saeed, K.A., Malhotra, M.K. & Grover, V. (2005). Decision Sciences, ed. «Examining the Impact of Interorganizational Systems on Process Efficiency and Sourcing Leverage in Buyer–Supplier Dyads.». pp. 36, pp.365-396. [en línea]. visitado: [10 de noviembre 2022] Disponible en: <https://web.archive.org/web/20180520053742/https://pdfs.semanticscholar.org/c6eb/ab359171c2fed6063ff7d0880308909dbe1e.pdf>
11. Ramírez, Jorge. Evolución de las Tecnologías de la Información. 2003. [en línea]. visitado: [10 de noviembre 2022] Disponible en: <http://eprints.rclis.org/4657/>
12. López Clemente, Hurtado Beltrán y Jiménez González: "Propuesta de acciones para solucionar las deficiencias en cuanto al control de los activos fijos tangibles en Cuba" en Observatorio de la Economía Latinoamericana, No. 169, 2012. [en línea]. [visitado: 5 de noviembre 2022], Disponible en: <https://www.eumed.net/cursecon/ecolat/cu/2012/cbg.pdf>

13. Remedios Arenciba, Gilberto. Sistema de gestión de útiles y herramientas de la Facultad 2. 2019. Tesis de Licenciatura. Universidad de las Ciencias Informáticas. Facultad 2. [en línea]. [visitado: 20 junio 2022]. Disponible en: <https://repositorio.uci.cu/handle/123456789/10261>
14. Albuja-Cortés, Emilio; Medina-Chicaiza, Patricio; Herrera-Herrera, José. Procedimiento para gestión de la información en el área de servicios municipales mediante la aplicación de Tecnologías de Información y Comunicación. Polo del Conocimiento, 2018, vol. 3, no 1 Mon, p. 144-163.
15. Cetaris, Inventory Management App. [en línea]. [visitado: 21 junio 2022], Disponible en: <https://cetaris.com/inventory-count-app>
16. UpKeep, Management Software. [en línea]. [visitado: 21 junio 2022], Disponible en: <https://www.upkeep.com/product/inventory-management>
17. Assetiger, Asset Management tool. [en línea]. [visitado: 21 junio 2022], Disponible en: <https://www.myassettag.com/assettiger/>
18. Empresa Desarrolladora de Software (DESOFT). 2010. Manual de usuarios del software AvilaDoc. División Ciego de Ávila - Camagüey: DESOFT, 2010.
19. Díaz, Pedro M. Puig; Caballero, Yanisleidi González. Estado de las herramientas para la auditoría y control de los Activos Fijos Tangibles. Serie Científica de la Universidad de las Ciencias Informáticas, 2019, vol. 12, no 4, p. 43-50.
20. Sáudo, Armando Lauchy, et al. Contribución a la gestión económico-financiera del Ministerio de Educación Superior. Anales de la Academia de Ciencias de Cuba, 2019, vol. 9, no 3, p. 675.
21. Diaz Hernandez, Kimady. Sistema para la gestión y control de los procesos en el departamento de Tecnología CITEC. Trabajo de Diploma. Universidad de Ciencias Informáticas (UCI). 2021.
22. Pressman RS, Ingeniería de software: Un enfoque práctico, 5.a ed. 2003
23. Sierra González, Julio César. Métodos de Evaluación de Usabilidad para Sistemas de Información Web: una revisión. Departamento de Ingeniería de Sistemas e Industrial, 2019.
24. LUNA, Fernando. JavaScript-Aprende a programar en el lenguaje de la web. RedUsers, 2019.
25. PHP, A popular general-purpose scripting language. [en línea]. [visitado: 23 de junio 2022], Disponible en: <https://www.php.net/manual/es/index.php>
26. HTML, Lenguaje HTML. [en línea]. [visitado: 23 de junio 2022], Disponible en: <https://lenguajehtml.com/html/>
27. CSS, Lenguaje CSS. [en línea]. [visitado 23 de junio 2022], Disponible en: <https://lenguajecss.com/>
28. Codina, Lluís. Revisiones bibliográficas sistematizadas: procedimientos generales y Framework para ciencias humanas y sociales. 2018. en línea]. [visitado: 23 de junio 2022], Disponible en: <https://repositori.upf.edu/handle/10230/34497>
29. Laravel, The PHP framework for Web Artisans. [en línea]. [visitado: 23 de junio 2022], Disponible en: <https://laravel.com/>
30. Bootstrap, Build fast responsive sites witch Bootstrap. [en línea]. [visitado: 23 de junio 2022], Disponible en: <https://getbootstrap.com/docs/5.2/getting-started/introduction/>
31. Jiménez, José Luis Ávila. UF2215-Herramientas de los sistemas gestores de bases de datos. Pasarelas y medios de conexión. Editorial Elearning, SL, 2018
32. Santillán, Luis Alberto Casillas; Ginestá, Marc Gibert; Mora, Óscar Pérez. Bases de datos en MySQL. Universidad Obrera de Catalunya, 2014.

33. Fezari, Mohamed; Al Dahoud, Ali. Integrated development environment “IDE” for Arduino. WSN applications, 2018, p. 1-12.
34. Pino, María Teresa Pérez, “Experiencia en la utilización de mapas conceptuales interactivos en la asignatura Introducción a las Ciencias Informáticas”. Revista Tecnología Educativa, 2020, vol. 5, no 2.
35. Toro, Amador Durán; Jiménez, Beatriz Bernárdez. Metodología para la elicitación de requisitos de sistemas software. Informe Técnico LSI-2000-10. Facultad de Informática y Estadística Universidad de Sevilla, 2000.
36. ADMIN. Desarrollador Senior: Patrón Simple Factory en PHP5. Desarrollador Senior [en línea]. [visitado 21 de agosto 2022]. Disponible en: <http://desarrolladorsenior.blogspot.com/2009/09/patron-simple-factory-en-php5.html>
37. Gamma, E., Helm, R. y Johnson, R. y Vlissides, J. 2009. Patrones de diseño. Patrones de diseño. vico, 2009. [en línea], [visitado: 22 de agosto de 2022.], Disponible en: <http://www.vico.org/pages/PatronsDisseny.html>
38. Guerrero, Carlos A.; Suárez, Johanna M.; Gutiérrez, Luz E. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. Información tecnológica, 2013, vol. 24, no 3, p. 103-114. [en línea]. [visitado: 23 de junio 2022], Disponible en: <http://www.javier8a.com/itc/bd1/gof.pdf>
39. Booch, Grady, Rumbaugh, James, Jacobson y Ivar. 2009. El lenguaje unificado de modelado. Manual de 2009.
40. Otálora Hernández, Cindy Viviana. Patrón arquitectónico MVC (Modelo Vista Controlador). 31614-Nuevas Tecnologías de Desarrollo, 2018.
41. Pressman, Roger S. y Maxim, Bruce R. 2015. Software engineering: a practitioner's approach, Eighth Edition. New York: McGraw-Hill Education, 2015. ISBN 978-0-07-802212-8.
42. Cama, Zara Yujra. Desarrollo de una Herramienta CASE para el Diseño de Diagramas Entidad-Relación Extendido y su mapeo al Modelo Relacional Orientado a Estudiantes en el contexto. INF-FCPN-PGI Revista PGI, 2021, p. 210-213. [en línea], [visitado: 22 de agosto de 2022.], Disponible en: https://ojs.umsa.bo/ojs/index.php/inf_fcpn_pgi/article/view/87
43. Higuera Villamil, Daniel Ricardo; Ramos Ebratt, Sebastián Daniel. Diseño de software para el módulo de cliente usuario web de la plataforma de aprendizaje de inglés para la Universidad Santo Tomas contemplando tres habilidades de aprendizaje. 2020.
44. Alzate-Ibañez, Angélica María; Ramírez Río, John Fredy; Bedoya Montoya, Laura María. Modelo para la implementación de un sistema integrado de gestión de calidad y ambiental en una empresa siderúrgica. Ciencias administrativas, 2019, no 13, p. 3-13.
45. Ermel, Guilherme, et al. Composición del diagrama de componentes de apoyo UML. En Actas del XIV Simposio Brasileño de Sistemas de Información. SBC, 2018. p. 441-448.
46. Sánchez Peño, José Manuel. Pruebas de software. fundamentos y técnicas. 2015. [en línea]. [visitado: 23 de octubre 2022], Disponible en: <https://www.redalyc.org/pdf/922/92210205.pdf>
47. Cardona, Juan P.; Leal, John J.; Ustariz, José E. Modelado matemático de caja blanca y negra en educación en ingeniería. Formación universitaria, 2020, vol. 13, no 6, p. 105-118. [en línea]. [visitado: 10 de junio 2022], Disponible en: https://www.scielo.cl/scielo.php?pid=S0718-50062020000600105&script=sci_arttext

ANEXOS

Entrevista semiestructurada

Con fines del Trabajo de Diploma, Sistema Web para la gestión de Información en el Departamento de Tecnología de la Facultad de Ciencias y Tecnologías Computacionales, ubicada en la Universidad de Ciencias Informáticas, requerimos realizar una investigación sobre el funcionamiento y análisis de la gestión de información implementada en el Departamento.

La información brindada en esta entrevista es de carácter académico y confidencial, solo será utilizada para cumplir los propósitos de la investigación. Agradecemos su colaboración.

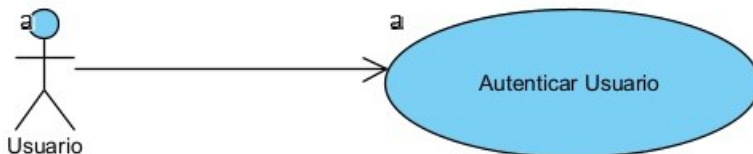
Persona entrevistada: Ing. Glennis Tamayo Morales

Cargo: Jefe de Departamento

Banco de Preguntas

1. ¿Qué procesos se realizan en el Departamento?
Gestión de Activos Fijos Tangibles y Útiles, Movimiento de medios básicos, Gestión de los Expedientes Técnicos de los ordenadores, Mantenimiento de los ordenadores, el servicio de los laboratorios, aulas y salones de conferencia.
2. ¿Qué es un modelo de Movimiento de Medio Básico y un Expediente Técnico?
Ambos son documentos oficiales el primero registra los movimientos de los medios básicos en las diferentes partes de la Universidad y el otro registra los datos de los ordenadores y un historial de modificaciones realizadas en él.
3. ¿Existe dificultad para realizar estos procesos?
Realmente se hace complicado en reiteradas ocasiones, ya que se dificulta encontrar los modelos de movimientos porque se encuentran archivados en copia dura en archivos donde hay muchos de ellos y hay que buscar uno por uno hasta encontrar el modelo. En el caso de los Expedientes Técnicos muchos de ellos se encuentran deteriorados y se dificulta la lectura o consulta de ellos.
4. ¿Se utiliza algún Sistema para la Gestión de Información en el Departamento?
Para realizar los movimientos tenemos que reportarlos con Economía, existe un sistema ahí que lleva la información de todos estos, pero demoran en actualizarlo y pues para el trabajo diario resulta engorroso, ya que como te explique anteriormente es muy pesado revisar modelo por modelo, muchas veces toma mucho tiempo encontrarlos. En el caso de los Útiles sucede igual solo que estos se llevan mediante una tabla de Excel.
5. ¿Qué importancia le concede a la Gestión de Información?
Es importante para los tiempos en que vivimos, pues ayuda a manejar la información, al archivado y recuperación de la misma de una manera más sencilla.

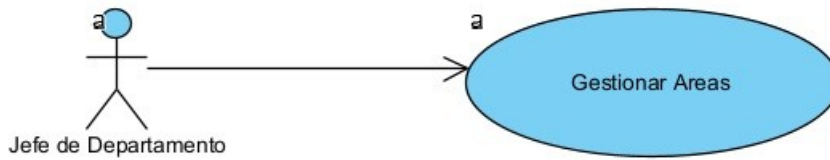
Anexo 1 - Entrevista realizada al Jefe de Departamento de Tecnología



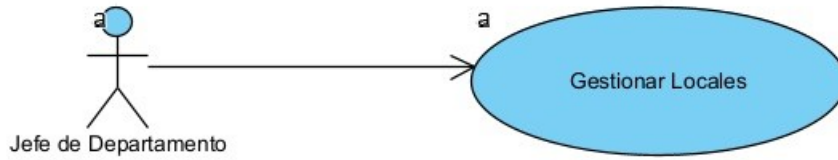
Anexo 2 - Diagrama de Caso de Uso "Autenticar Usuario"



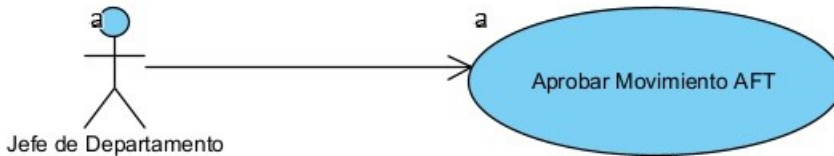
Anexo 3 - Diagrama de Caso de Uso "Gestionar Usuario"



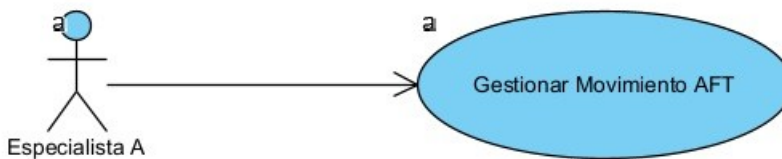
Anexo 4 - Diagrama de Caso de Uso "Gestionar Áreas"



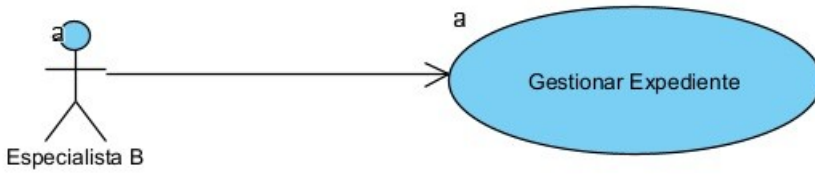
Anexo 5 - Diagrama de Caso de Uso "Gestionar Locales"



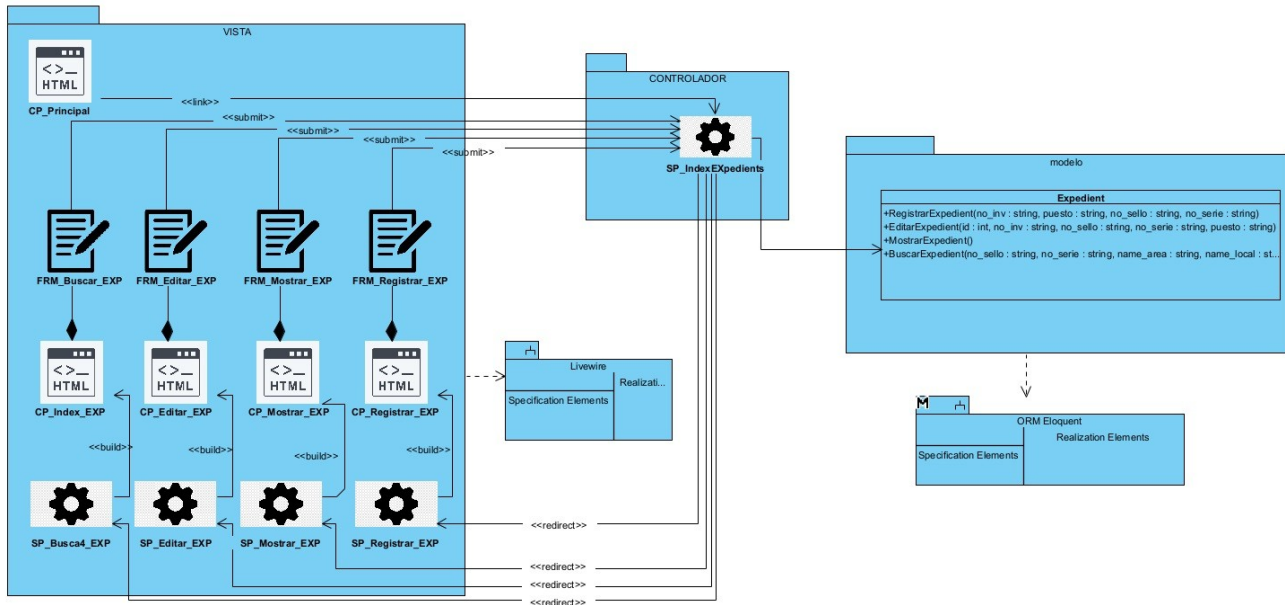
Anexo 6 - Diagrama de Caso de Uso "Aprobar Movimiento AFT"



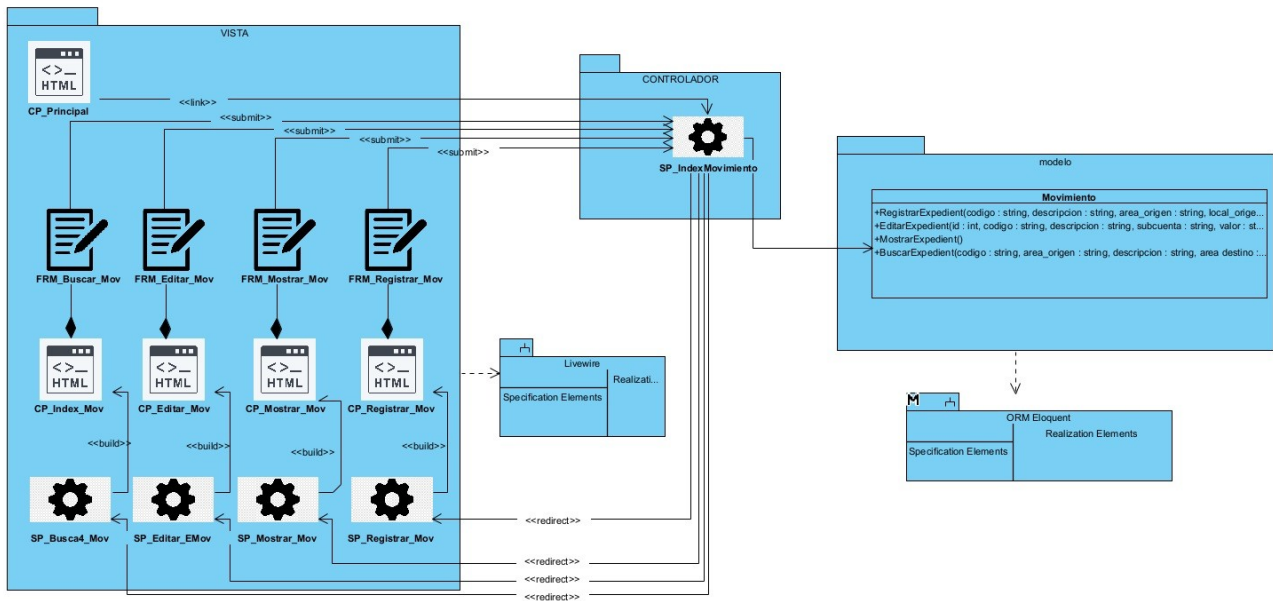
Anexo 7 - Diagrama de Caso de Uso "Gestionar Movimiento de AFT"



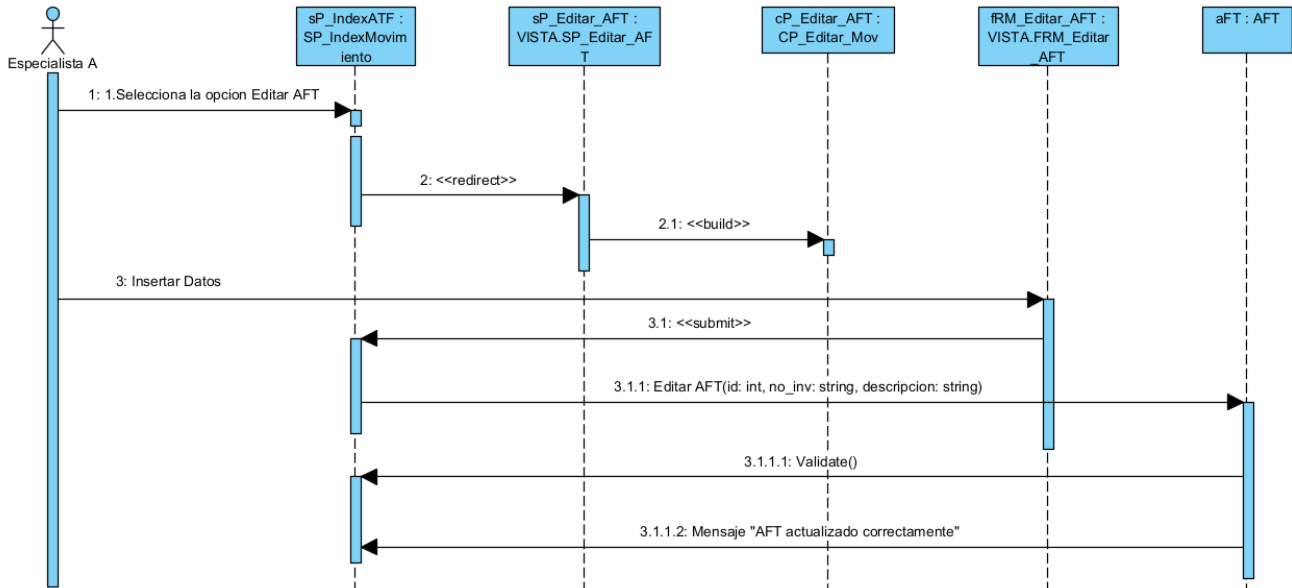
Anexo 8 - Diagrama de Caso de Uso "Gestionar Expediente"



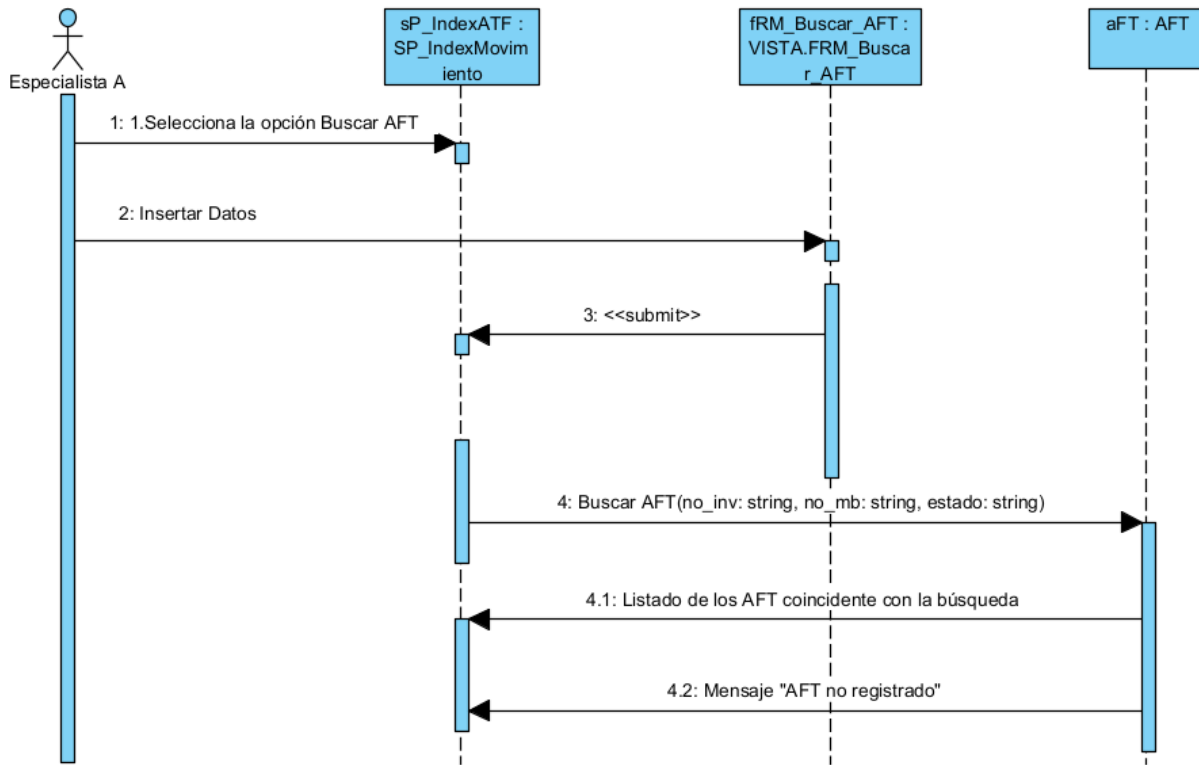
Anexo 9 - Diagrama de Clase del Diseño CU "Gestionar Expediente"



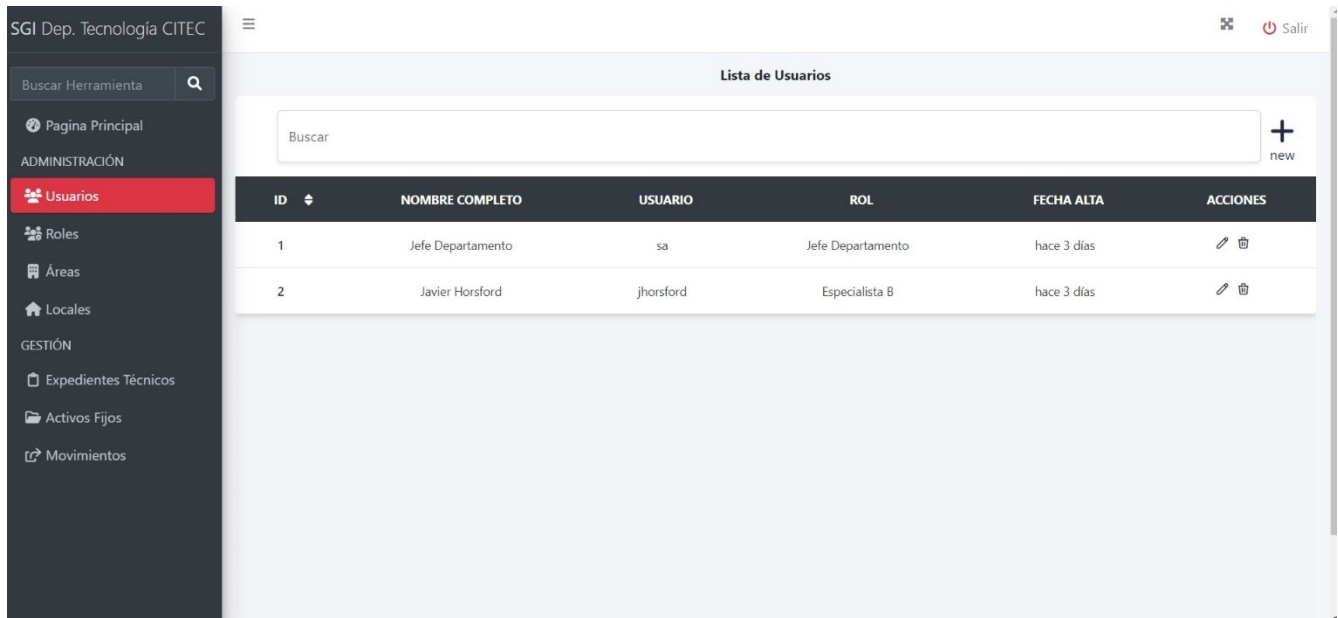
Anexo 10 - Diagrama de Clase del Diseño CU "Gestionar Movimientos"



Anexo 11 - Diagrama de Secuencia CU "Editar AFT"



Anexo 12 - Diagrama de Secuencia CU "Buscar AFT"



Anexo 13 - Interfaz Gestionar Usuarios

The screenshot shows the 'Lista de Roles' (Roles List) interface. On the left is a dark sidebar with the logo 'SGI Dep. Tecnología CITEC' and a search bar 'Buscar Herramienta'. Below the search bar are menu items: 'Pagina Principal', 'ADMINISTRACIÓN', 'Usuarios', 'Roles' (highlighted in red), 'Áreas', 'Locales', 'GESTIÓN', 'Expedientes Técnicos', 'Activos Fijos', and 'Movimientos'. The main content area has a title 'Lista de Roles', a search bar 'Buscar', and a '+ new' button. Below is a table with three columns: 'ID', 'NOMBRE', and 'ACCIONES'. The table contains three rows of role data.

ID	NOMBRE	ACCIONES
1	Jefe Departamento	
2	Especialista A	
3	Especialista B	

Anexo 14 - Interfaz Gestionar Roles

The screenshot shows the 'Lista de Áreas' (Areas List) interface. The sidebar is identical to the previous screenshot, but 'Áreas' is highlighted in red. The main content area has a title 'Lista de Áreas', a search bar 'Buscar', and a '+ new' button. Below is a table with four columns: 'ID', 'NOMBRE DE ÁREA', 'RESPONSABLE', and 'ACCIONES'. The table contains two rows of area data.

ID	NOMBRE DE ÁREA	RESPONSABLE	ACCIONES
1	Departamento 1	Responsable 1	
2	Departamento 2	Responsable 2	

Anexo 15 - Interfaz Gestionar Áreas

SGI Dep. Tecnología CITEC

Buscar Herramienta

Página Principal

ADMINISTRACIÓN

- Usuarios
- Roles
- Áreas
- Locales**

GESTIÓN

- Expedientes Técnicos
- Activos Fijos
- Movimientos

Lista de Locales

Buscar

+ new

ID	NOMBRE DE LOCAL	RESPONSABLE	ACCIONES
1	Local 1	Responsable 1	

Anexo 16 - Interfaz Gestionar Locales

SGI Dep. Tecnología CITEC

Buscar Herramienta

Página Principal

ADMINISTRACIÓN

- Usuarios
- Roles
- Áreas
- Locales
- Expedientes Técnicos**

GESTIÓN

- Activos Fijos
- Movimientos

Lista de Expedientes

Buscar

+ new

ID	LOCAL	RESPONSABLE	NÚMERO INVENTARIO	NÚMERO SELLO	HISTORIAL	ACCIONES
1	Local 1	Responsable 1	21312312	123213		
2	Local 1	Responsable 1	55965	69595		

Anexo 17- Interfaz Gestionar Expedientes

SGI Dep. Tecnología CITEC

Buscar Herramienta

Pagina Principal

ADMINISTRACIÓN

- Usuarios
- Roles
- Áreas
- Locales

GESTIÓN

- Expedientes Técnicos
- Activos Fijos**
- Movimientos

Lista de Activos Fijos

Buscar

ID	NÚMERO MB	TIPO	DESCRIPCION	ÁREA	LOCAL	ESTADO	ACCIONES
1	342424234	PC	fsd fsdfsdf sdf	Departamento 2	Local 1	activo	
3	123	Mesa	123	Departamento 2	Local 1	roto	

Anexo 18 - Interfaz Gestionar AFT

SGI Dep. Tecnología CITEC

Buscar Herramienta

Pagina Principal

ADMINISTRACIÓN

- Usuarios
- Roles
- Áreas
- Locales

GESTIÓN

- Expedientes Técnicos
- Activos Fijos
- Movimientos**

Lista de Movimientos

Buscar

ID	CÓDIGO	ÁREA ORIGEN	ÁREA DESTINO	DESCRIPCION	APROBADO	ACCIONES
1	435345	Departamento 1	Departamento 2	fsd fsdfsdf sdf	x	
2	4545	Departamento 1	Departamento 2	123	✓	

Anexo 19 - Interfaz Gestionar Movimientos AFT