UNIVERSIDAD DE CIENFUEGOS
INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA

# Universidad de Cienfuegos

## “Carlos Rafael Rodríguez”

### Facultad de Ingeniería

## Título:

# Aplicación web de apoyo a la gestión de control interno de la Universidad de Cienfuegos


**Autor:**

Alexis Manuel Hurtado García

**Tutor:**

Dr. Eduardo Concepción Morales


**Cienfuegos, Cuba**

**Curso: 2023**

# DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad Cienfuegos los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ___ días del mes de _____ del año _____.

_____ _____

(Autor) (Autor)

_____ _____

(Tutor) (Cotutor)

# Resumen:

El presente trabajo titulado “Aplicación web de apoyo a la gestión de control interno de la Universidad de Cienfuegos”, tiene como objetivo la realización de una página web que permita gestionar la información de los Activos Fijos Tangibles de las diferentes Áreas de Responsabilidad de la Universidad de Cienfuegos. Esta solución informática viene del problema que presentan los jefes de Áreas y Departamentos al querer realizar el control interno a los Activos Fijos Tangibles, resulta difícil tener que mantener coordinada la información de cada sitio con la del reporte entregado por el software Assets, en el cual se lleva la contabilidad de toda esa información.

Para el desarrollo de este Sistema se emplearon las siguientes herramientas: Visual Studio Code, como entorno de desarrollo, Table Plus y Thunder Client para la revisión del funcionamiento de la base de Datos.

Como lenguajes de programación y frameworks utilizados estuvieron HTML, CSS, Tailwind CSS, JavaScript, TypeScript, React, NextJS, se usó Prisma como ORM (Object Relational Mapping) y como gestor de base de datos PostgreSQL.

Con el desarrollo de este trabajo se estudiaron los elementos que conforman el análisis y diseño del sistema que se propone, donde se utilizó la metodología de desarrollo Proceso Unificado de Desarrollo de Software (AUP) con el Lenguaje Unificado de Modelado (UML).



Palabras Clave:

Área de Responsabilidad, framework, Activos Fijos Tangibles, Assets.



Abstract:

The present work entitled "Web application to support the management of internal control of the University of Cienfuegos", aims to create a web page that allows managing the information on the Tangible Fixed Assets of the different Areas of Responsibility of the University from Cienfuegos. This computer solution comes from the problem that the

heads of Areas and Departments present when wanting to carry out internal control of the Tangible Fixed Assets, it is difficult to have to keep the information of each site coordinated with that of the report delivered by the Assets software, in which accounting of all that information is kept.

For the development of this System, the following tools were used: Visual Studio Code, as a development environment, Table Plus and Thunder Client to review the operation of the database.

The programming languages and frameworks used were HTML, CSS, Tailwind CSS, JavaScript, TypeScript, React, NextJS, Prisma was used as ORM (Object Relational Mapping) and PostgreSQL as a database manager.

With the development of this work, the elements that make up the analysis and design of the proposed system were studied, where the development methodology was used Agile Unified Process (AUP) with the Unified Modeling Language (UML).

Keywords:

Area of Responsibility, framework, Tangible Fixed Assets, Assets.

# Índice

## Introducción

Podemos decir que la gestión eficaz de los bienes y activos fijos tangibles es esencial para cualquier organización que busque maximizar su rentabilidad y lograr sus objetivos estratégicos en el largo plazo[1]. Por esta razón, el control interno se ha convertido en una preocupación cada vez más importante para empresas e instituciones de todo el mundo. Se asume que la situación mundial actual es un reflejo de la importancia del control interno en los centros de trabajo y organizaciones en general. Así como al igual que la pandemia afectó a todo el mundo hace algunos años, la falta de implementación de métodos y estrategias efectivas de control interno en las empresas y organizaciones puede tener graves consecuencias en la calidad y continuidad de sus operaciones.

En un contexto global, se han producido casos significativos de errores y fraudes relacionados con la gestión de bienes y activos fijos tangibles en diferentes tipos de organizaciones, lo que ha llevado a un mayor escrutinio de los controles internos utilizados para prevenirlos y así garantizar la eficiencia y la transparencia en el manejo de los recursos de organizaciones. Las normativas legales y las regulaciones cada vez más rigurosas han aumentado la presión sobre las organizaciones para garantizar que implementen medidas efectivas de control interno, ya que este asegura el correcto funcionamiento y protección de los activos de la organización, lo que se traduce en la optimización de los recursos y en el cumplimiento de las regulaciones y leyes aplicables a las empresas.

Las medidas actuales ayudan a gestionar de forma efectiva la ubicación de estos Activos, en el pasado los principales peligros y riesgos, eran asociados con la naturaleza, catástrofes naturales, ahora primordialmente se imputan a acciones y decisiones humanas.

Con el desarrollo de la informática la contabilidad tuvo grandes transformaciones sobre todo a partir de los años 80 y 90 del pasado siglo, cuando se comenzaron a automatizar los principales libros contables y surgieron los primeros Sistemas Contables Automatizados. El avance de las tecnologías de la información ha permitido la integración de varios sistemas. Ejemplos de estos sistemas son los ERP (Enterprise Resource

Planning)[2] que permiten la integración de ciertas operaciones de una empresa y en el que se armoniza la gestión de información de varios procesos como los de producción, logística, la gestión económica, etc.

El Sistema de Planificación de Recursos denominado ERP surgió de la necesidad de englobar todos los datos referentes a la totalidad de la cadena de producción de las empresas, con el fin de brindar información confiable en tiempo real. Se trata básicamente de un software desarrollado para el manejo eficaz de la información de las empresas, que permite tomar decisiones acertadas en los momentos oportunos, gracias a la veracidad de los datos que se manejan mediante el ERP.

Los ERP o sistema informático de gestión de información para una empresa son muy utilizados a nivel internacional. Este tiene muchos beneficios para la empresa: automatiza su gestión, tiene un mayor control de lo que hace y un ahorro de costes (eficiencia y eficacia). Esto hace a las empresas más competitivas.

A nivel nacional, en Cuba, un país en vías de desarrollo enfrenta mayores desafíos en cuanto a la aplicación de controles internos efectivos. La falta de recursos y la complejidad del entorno operativo pueden dificultar la identificación temprana de problemas relacionados con la gestión financiera y contable de bienes y activos fijos tangibles. La automatización de estos procesos puede ser una solución efectiva para superar estas barreras y mejorar la eficiencia de controles internos.

En el caso específico de la Universidad de Cienfuegos, el control interno de inventarios, bienes y activos fijos tangibles es esencial para mantener la calidad y el orden en los procesos de enseñanza y aprendizaje. Este proceso se complejiza debido a las particularidades variantes de la gestión financiera en una institución académica, estas requieren una atención especial.

Debido al creciente proceso de informatización de la sociedad siempre se debe comprometer la utilización de tecnologías, software y herramientas adecuadas para realizar de formas más rápida cualquier proceso, como un registro oportuno, confiable y eficiente de los bienes y activos fijos tangibles, permitiendo a las facultades mantener un

control constante de los recursos y asegurar que los mismos se encuentren en óptimas condiciones para llevar a cabo las actividades académicas y educativas.

Además, la implementación de un adecuado control interno por parte de la administración promueve la transparencia y la rendición de cuentas dentro de las facultades universitarias. Esto permite a las autoridades universitarias tomar decisiones informadas y tomar medidas correctivas a tiempo, en caso de detectar cualquier anomalía en el manejo de recursos.

El sistema ERP utilizado en la Universidad de Cienfuegos es el ASSETS[3], un Sistema de Gestión Integral estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos. 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.

Dispone, además, de métodos para administración y planificación de inventarios, así como una amplia gama de Análisis y Consultas que le permitirán no sólo conocer exactamente la situación actual, sino proyectar decisiones futuras. La Universidad de Cienfuegos es una institución presupuestada, por lo que debe manejarse adecuadamente el presupuesto para poder cumplir su misión.

A pesar de la utilización del sistema integral Assets existen reportes que no están incorporados en el mismo y por lo tanto se confeccionan de forma manual o son controlados por medio de una hoja de cálculo Excel. Todo esto trae como consecuencia una considerable demora a la hora de entregar la información correspondiente a los especialistas.

Poniendo de ejemplo el proceso de control mensual a los bienes y activos fijos de las Áreas de Responsabilidad; en la actualidad este proceso se realiza personalmente por los encargados de cada Área, no hay manera de generar un informe automático sobre el control llevado mensualmente, así que estos son tomados escritos, documentos y actas, son enviadas para su posterior revisión.

La implementación de este sistema trae ventajas incuestionables como la de la agilización de los trámites, permitirá a los jefes de Áreas tener la información más cercana a ellos, que se puedan ver las fechas de las revisiones, los datos como el estado de los bienes. También es mucho más fácil llenar un formulario y dar en Aceptar que tener que ir a las oficinas a pedir personalmente los documentos para llenarlos o tener que llenar formularios escritos y archivarlos.

Se describe como **situación problémica** la dificultad de los trabajadores y jefes de áreas para gestionar la información sobre los activos fijos tangibles y su ubicación.

A partir de la situación problémica descrita anteriormente podemos asumir como **problema a resolver** ¿Cómo hacer accesible la información de la gestión de Control Interno en la Universidad de Cienfuegos?, este problema se enmarca en el **objeto de estudio:** Sistema de Control Interno.

Su **campo de acción** se enfoca en: El proceso de Control Interno de los Activos Fijos Tangibles.

Para darle solución a todo lo anteriormente dicho se plantea como **objetivo general:** Elaborar una aplicación web que permita a jefes de área y administradores gestionar la información de los activos fijos de las diferentes áreas.

Se plantea como **idea a defender** que, si se implementa una aplicación web para la gestión de los activos fijos se va a favorecer cada área de la Universidad, haciendo accesible esta información.

**Objetivos específicos**:

1. Conocer el proceso de gestión contable y control interno en la Universidad de Cienfuegos.

2. Diseñar un sistema informático para gestionar correctamente la información económica requerida.

3. Implementar el sistema informático propuesto.

4. Determinar la validez del sistema mediante las pruebas de software.

Para alcanzar estos objetivos se precisa la realización de diferentes tareas dentro de la investigación, como son:

- Estudio del proceso de gestión de control interno en la Universidad de Cienfuegos.
- Entrevistas a los trabajadores que intervienen en los procesos de control interno, tanto en economía como en las áreas específicas.
- Estudio de las aplicaciones existentes relacionadas con el problema a resolver.
- Selección de las herramientas, metodologías, lenguajes y tecnologías para la implementación de la aplicación.
- Estudio de la base de datos para la persistencia de la información en el sistema informático.
- Diseño de la interfaz gráfica de la aplicación.
- Validación de la propuesta.
- Documentación de la información generada durante el negocio, diseño e implementación del sistema.

**Posibles resultados en artefactos a entregar:**

**1-** Aplicación Web desarrollada con NextJS, que usa tecnología de React para la visualización de la información.

El **aporte práctico** de esta investigación consiste la facilidad que brindará la puesta en acción de este sistema, permitiendo el acceso a la información a los jefes de Área sin tener que pedirla directamente en economía o esperar su actualización. Este software también agilizará considerablemente el tiempo empleado para el procesamiento de la

información al colocar a disposición del usuario en cada momento las herramientas, y datos necesarios para el desarrollo con éxito del proceso.

El **método científico de investigación** es una forma de estudiar los fenómenos de la naturaleza y la sociedad para descubrir sus relaciones y su esencia, el mismo se puede clasificar en métodos de nivel teórico o empírico, los cuales están relacionados entre sí de forma dialéctica. A continuación, se muestran los métodos utilizados para el desarrollo de esta investigación:

**Métodos teóricos:**

- **Método Analítico-Sintético:** Posibilitó realizar un análisis de las distintas partes que afectan el objeto de estudio y sintetizar los elementos más significativos.
- **Método Histórico-Lógico:** Para la realización de la investigación se hizo necesario estudiar la evolución del problema y la existencia de metodologías, procedimientos y sistemas informáticos similares al que se pretende elaborar, determinando cuales son las tendencias actuales para el desarrollo de sistemas de control interno.
- **Modelación:** La modelación se emplea al determinar, representar y explicar las etapas de los procesos presentes en el negocio, así como los fundamentos en que se sustenta como un reflejo de lo que debe hacerse para la correcta implementación de los mismos en el sistema informático.
- **Inductivo-deductivo:** Se utiliza la inducción para llegar a generalizaciones partiendo del análisis de casos particulares, mientras la deducción expresa el movimiento de lo general a lo particular.

**Métodos empíricos:**

- **Observación:** Se empleó para identificar características en el proceso de control interno, para poder comprender su funcionamiento actual y de esta manera poder realizar una correcta modelación del negocio.
- **Entrevista:** Se realizó para poder obtener información sobre como se realiza el proceso de control interno en las áreas.

Este proceso está estructurado en 3 capítulos:

**Capítulo 1:**

Fundamentos teórico-metodológicos: Se presentan los elementos teóricos referente al proceso de control interno, que sirven de base a la investigación del problema planteado. En este capítulo se analizarán los principales conceptos relacionados con el objeto de estudio, se realiza una descripción de los sistemas existentes de control interno; y una explicación de herramientas y tecnologías potenciales a utilizar en el desarrollo de la propuesta de solución.

**Capítulo 2:**

Descripción de la propuesta de solución: Se describe la propuesta de solución y los artefactos generados por el escenario número 4 en la metodología Agile Unified Process (AUP) en las etapas de análisis, diseño e implementación.

**Capítulo 3:**

En este capítulo se presentan las pruebas funcionales aplicadas al sistema. Esto se realiza utilizando las historias de usuario generadas por la metodología utilizada.

# Capítulo 1: Fundamentación Teórica

## 1.1 Introducción

En este capítulo se brinda una base teórica y conceptual para el desarrollo de la Aplicación web de apoyo a la gestión de control interno de la Universidad de Cienfuegos (UCF). Se presentan un grupo de conceptos asociados con el objeto de estudio, así como también se analizarán los sistemas y métodos ya existentes para el control interno en las distintas entidades. Además, se hace un análisis crítico de estos para argumentar la necesidad de implementar una nueva solución informática con las herramientas, lenguajes y metodología de desarrollo expuestas en este capítulo.

## 1.2 Descripción del Dominio del Problema

### 1.2.1 Conceptos asociados

Para lograr una correcta comprensión del problema resulta necesario conocer algunos conceptos relacionados con el objeto de estudio:

**Contabilidad:** Es una técnica o rama de la Contaduría que se encarga de cuantificar, medir y analizar las operaciones, la realidad económica y financiera de las organizaciones o empresas, con el fin de facilitar la dirección y el control; presentando la información, previamente registrada, de manera sistemática para las distintas partes interesadas. Dentro de la contabilidad se registran las transacciones, cambios internos o cualquier otro suceso que afecte económicamente a una entidad.

**Control Interno:** Plan de organización que incluye todos los métodos coordinados y medidas adoptada dentro de una empresa para salvaguardar sus medios y verificar sus datos contables.

Es un proceso integrado a las operaciones con enfoque de mejoramiento continuo, extendido a todas las actividades inherentes de la gestión, efectuado por la dirección y el resto del personal.

**Sistemas de Control Interno:** Son constructos teóricos[4], que se podían elaborar de acuerdo a parámetros clásicos y modernos. Y en ambos puntos podríamos encontramos

con ventajas y desventajas, pero la teoría contable cambia contantemente y con ella la manera de interpretar la realidad que precede la construcción de todo sistema de control. Posibilitan la toma de decisiones de forma rápida midiendo parámetros y analizando la información que contienen, basada en las fórmulas con las que estén hechos.

**Activos Fijos Tangibles:** Representan propiedades físicamente tangibles que han de utilizarse por un periodo largo en las operaciones de la entidad y que normalmente no están destinados a la venta.

**Área Administrativa:** Para el control de activos fijos tangibles el área debe coincidir con el ámbito de responsabilidad o custodia de un jefe, pudiendo este subdividirla y en cada subdivisión designar un subjefe o responsable de la custodia de los bienes ubicados en la misma.

**Centro de Costo:** Son Áreas de Responsabilidad que acumulan costos, generalmente responden a las dependencias de la organización o empresa. Cada Centro de Costo es responsable de los recursos necesarios para realizar sus actividades. La contabilidad se divide por Centros de Costo para determinar la rentabilidad de cada área de producción o servicios.

**Área de Responsabilidad:** Es una sub-área, designada por el jefe de esta, con un subjefe o responsable de los bienes o activos fijos tangibles ubicados en dicha área.

**Local:** Es una sub-área del Área de Responsabilidad, aporte de la tesis. Para poder dividir aún más la ubicación de los Activos Fijos, y así permitir ser más específico a la hora de dar información.

**Costo:** Es el importe de efectivo o equivalentes al precio pagado, o bien un valor razonable de la contraprestación entregada, para adquirir un activo fijo tangible, en el momento de su adquisición o construcción.

**Depreciación:** Es la pérdida o disminución gradual del valor de un activo fijo tangible a lo largo de su vida útil.

**Procedimiento de Control Interno:** Cada responsable de área tiene que firmar un acta de responsabilidad material de los activos fijos tangibles bajo su custodia, los que se relacionan según modelo SC-1-07 “Control de Activos Fijos Tangibles”, basado en el Decreto Ley número 60/11[5].

El área contable debe mantener actualizados los modelos, los responsables de área informaran a contabilidad por escrito de los movimientos efectuados.

Se debe elaborar inmediatamente a su ocurrencia los modelos para las altas o bajas, traslados y ventas.

Los jefes de las áreas de responsabilidad deberán programar chequeos periódicos y sistemáticos durante el año, deberán reportarlo de inmediato a las áreas de contabilidad en caso de detectarse faltantes o sobrantes, como se establece en los procedimientos. Además, aplicarse la responsabilidad material en los casos que corresponda como se encuentra regulado en el Decreto Ley No. 249/07, Consejo de Estado, fecha 23/7/2007[6].

**Responsabilidad Material:** Los jefes de áreas que tengan asignados Activos Fijos Tangibles tienen que tener firmadas Actas de Responsabilidad Material por la custodia de los mismos y en caso de faltantes o pérdidas, aplicársele dicha responsabilidad de acuerdo con lo regulado por el Decreto Ley No. 249/07, Consejo de Estado, fecha 23/7/2007.

Todo trabajador de una entidad que para su tarea necesite de algún bien o recurso del Estado deberá tener firmada Acta de Responsabilidad Material por la custodia de los bienes a él asignado y en caso de faltantes o pérdidas aplicárseles dicha responsabilidad.

**Chequeos periódicos del 10% físico:** Una vez en explotación el Sistema de Control de Activos Fijos Tangibles, así como los movimientos de los mismos y cuadres contables al cierre de cada período, se está en condiciones de asegurar el mantenimiento del mismo,

tanto en el físico como en valores, para lo cual se debe establecer un Cronograma de chequeo mensual del 10% de las existencias de cada Centro de Costo o Área Administrativa, para comprobar su confidencialidad, o su ajuste si fuera necesario, por errores o diferencias en la toma o en su movimiento. Este chequeo deberá responder a un Plan Previo, o a un Cronograma Anual.

**Sistema informático:** es un sistema que permite almacenar y procesar información mediante una serie de parte interrelacionadas, como el hardware, el software y el personal. De hecho, estos son sus tres componentes fundamentales. En otras palabras, se puede decir que los sistemas informáticos son el conjunto de técnicas que permiten guardar y garantizar la seguridad de información mediante sistemas informatizados. La función principal de un sistema informático es el procesamiento de datos almacenados. Gracias a que la utilidad de los sistemas informáticos, son adaptables a casi cualquier sector o actividad económica, pueden ser utilizados casi sin restricciones.

## 1.3 Descripción del objeto de estudio

### 1.3.1 Objetivos estratégicos de la organización.

La Universidad de Cienfuegos tiene sus antecedentes en el Instituto Superior Técnico de Cienfuegos (INSTEC), fundado en el año 1976 como respuesta a las necesidades de formación de profesionales de una región eminentemente industrial y agrícola. La zona industrial, que desde los primeros años del triunfo de la Revolución se construyó en el territorio cienfueguero, generó la necesidad de preparar profesionales en carreras técnicas.

En la Universidad de Cienfuegos se lleva a cabo una gestión por procesos, concebida a partir de una planeación estratégica a corto y mediano plazo en las que se definen los objetivos y medidas para cada proceso. Los procesos se encuentran identificados.

La **misión** de esta institución queda definida como sigue: La Universidad de Cienfuegos, dedicada a la formación integral y continua de profesionales competentes y comprometidos con la Patria Socialista, contribuye mediante el conocimiento y la

innovación a la dinamización del desarrollo socioeconómico del territorio de Cienfuegos y de la sociedad cubana.

Por su parte, la **visión** queda enmarcada en:

Somos una UNIVERSIDAD DE EXCELENCIA que:

- ✓ Promueve una cultura general integral acorde con los valores de la sociedad cubana.
- ✓ Ofrece profesionales líderes comprometidos con la transformación para el desarrollo sostenible de la sociedad cubana.
- ✓ Exhibe una alta visibilidad de sus resultados científicos y de innovación.
- ✓ Impacta en el desarrollo económico y social del territorio y la sociedad, satisfaciendo las necesidades de superación profesional y la implementación de resultados de investigación y la innovación.

### 1.3.2 Flujo actual de los procesos y análisis crítico de ejecución de estos

El ASSETS[3] en la UCF se utiliza en la gestión económica y contable, es empleado por los departamentos de Contabilidad y Finanzas, Recursos Humanos, Logística y Auditoria para poder desempeñar sus tareas diarias de gestión. Permite mantener una Contabilidad fiable y actualizada, garantizando poder realizar los cierres de períodos contables en tiempo y con la calidad requerida.

En la base de datos del ASSETS se almacena la información de todos los trabajadores del centro, desde información personal hasta su categoría, grado científico, puesto de trabajo, salario, etc. En cada departamento que emplea el ASSETS para la gestión de sus tareas diarias se usan diferentes módulos, en dependencia del departamento. El departamento de contabilidad y finanzas hay un personal encargado de llevar las finanzas, otro los útiles y herramientas, en dependencia de los módulos hay distintos trabajadores que se encargan de utilizarlos. Dentro de un mismo departamento hay personal calificado que puede usar de uno a varios módulos.

La persona que se encarga de llevar útiles y herramientas trabaja con el módulo de útiles y herramientas, así como la que tiene que ver con activos fijos trabaja con el módulo correspondiente, en caso de que alguien quiera buscar un activo según su código debe

ir a ver al personal en el sistema que realiza esa función el cual le hace el seguimiento en el sistema y devuelve la información del mismo. Hay personas que llevan inventario en el departamento de contabilidad y finanzas y otras trabajan con ese módulo en logística. Hay otro personal que lleva el combustible que es un activo de los almacenes.[7]

El ASSETS cuenta con un administrador cuyo rol principal es cerciorarse que el sistema esté funcionando sin ningún problema para que los demás usuarios puedan hacer las tareas de su día a día, ya sea tareas de economía, recursos humanos o logística. El administrador debe estar al tanto de que la web esté funcionando, en caso de que a un usuario se le olvidara la contraseña, el administrador tiene que entrar al sistema en modo administrador y restablecerla. Cada vez que el usuario tenía un problema, estos se podían resolver de forma remota, en última instancia debe ir a arreglarlo personalmente.

El ASSETS es un sistema imprescindible para la realización de las tareas de gestión en la universidad, el uso de este programa aumenta en gran manera la productividad de los departamentos que lo emplean a diario, pero el proceso de obtener información almacenada en el sistema se hace engorroso para personal que no tiene acceso a este. Recuperar toda la información en general sobre un activo o trabajador, dado que se encuentra disperso en distintos módulos se hace difícil; si un trabajador quisiera acceder a toda su información, debe ir por cada departamento en correspondencia del módulo del sistema que contenga la información, si quiere información de recursos humanos debe ir a recursos humanos y si quiere ver su nómina debe ir a contabilidad y finanzas en vez de poder conseguir toda la información por una única vía. Otra situación se presenta cuando otro sistema informático necesita emplear información que está almacenada en el ASSETS para llevar a cabo algunos de sus procesos; esta tarea sería imposible de momento porque el ASSETS no provee de un mecanismo de integración para que otras aplicaciones externas puedan hacer uso de la información en él almacenada.

## 1.4 Descripción de los sistemas existentes

Existen varios sistemas informáticos que se utilizan para gestionar el control interno

en las organizaciones, ninguno cumple con los requerimientos que necesitan en este proyecto, pero se pueden usar como alternativas. Los principales ERP[8] utilizados en Cuba son el de la empresa alemana SAP que mantiene el liderazgo a nivel mundial, el Assets ya mencionado y otro basado en software libre, el Odoo antes OpenERP. Una pequeña descripción de estos:

1- **Assets Ultimate:** Es un Sistema de Gestión Integral estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos. 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.[3]
2- **Audita:** Es un software para la administración integral de Auditorías con Orientación a Riesgos. Estandariza el trabajo y mejora la calidad del mismo. Permite la planificación, gestión de los equipos de auditores, la ejecución y el seguimiento de hallazgos y la comunicación con los auditados. Se enfoca en el Control Interno basándose en datos (índices de criticidad, riegos, etc.), permitiendo al auditor planificar lo que será auditado, por quién y en qué momento del ejercicio. Papeles de Trabajo guardados en forma centralizada (una base de datos o una ruta lógica en el disco de un servidor). Su defecto es que su uso es enfocado en Auditorias[9], [10].
3- **SAP:** Es uno de los sistemas de gestión empresarial más utilizados en todo el mundo. Es uno de los sistemas de gestión empresarial más completos en términos de funcionalidades, ofreciendo soluciones para la gestión de finanzas, logística, recursos humanos, ventas, marketing, producción, entre otras. Una de las ventajas de SAP es su flexibilidad y escalabilidad, ya que se adapta a los diferentes tamaños y tipos de organizaciones. Además, SAP es un sistema muy robusto en lo que se refiere al control interno, ya que cuenta con un módulo

específico para auditores que permite realizar una supervisión efectiva de los procesos y transacciones.[11]

4- **Odoo:** Es un software empresarial todo en uno que incluye CRM (Administración de Relaciones con el Cliente), sitio web y comercio electrónico, facturación, contabilidad, fabricación, gestión de almacenes y proyectos, e inventario entre otros. El módulo del servidor está escrito en el lenguaje Python. El cliente se comunica con éste a través de interfaces XML-RPC y JSON. Usa Postgres de base de datos.[12]

Cada uno de estos sistemas tiene sus propias características y detalles, pero en general destacan por ser muy útiles para la gestión empresarial, incluyendo el control interno. Todos ellos ofrecen funcionalidades para la automatización de procesos, generación de informes, análisis de datos, y muchos otros aspectos que permiten mejorar la eficiencia y la eficacia en la gestión y control interno de las organizaciones.

En cuanto a las diferencias entre los sistemas, las principales son las siguientes:

- En cuanto al control interno, todos los sistemas ofrecen soluciones para la supervisión y gestión de procesos y transacciones, pero SAP destaca por tener un módulo específico para auditores. Esto permite tener una supervisión más detallada y precisa de los procesos y transacciones.
- Odoo por su parte es el software más recomendable actualmente, y el más utilizado por Mipymes y pequeñas empresas que quieren gestionar su economía y activos de forma rápida y gratuita.
- Por último, los precios y el modelo de pago de los sistemas pueden variar, por lo que es importante tener en cuenta el presupuesto y las necesidades específicas de la organización antes de elegir uno u otro sistema.

En resumen, todos los sistemas mencionados son muy útiles para la gestión empresarial y el control interno, pero la elección del sistema dependerá de las necesidades específicas de la organización. En este proyecto ninguno puede cumplir con

las necesidades de los clientes, debido a que no permiten gestionar el flujo de los Activos Fijos Tangibles ni mostrar dicha información a los jefes de las Áreas de forma individual.

## 1.5 Tendencias, metodologías y/o tecnologías actuales

El uso de las metodologías de software constituye un elemento clave en el desarrollo de software para guiar, organizar, facilitar, informatizar y agilizar un determinado proyecto. Aporta una garantía de calidad, así como una forma de estimar y gestionar los costos de desarrollo de un proyecto. Al ser un proceso estructurado también organiza la forma en la que el proyecto va a ser realizado, herramientas usadas, como UML[13], [14], y demás, obligando a revisar que los resultados sean los correctos antes de proseguir y marcando metas intermedias para gestionar el avance del proyecto. Así pues, se logra una mayor eficiencia de recursos, es decir, se invierte lo mínimo para obtener lo máximo a cambio. Existen dos tipos principales de metodologías, las ágiles (también llamadas ligeras)[15] y las tradicionales (también conocidas como pesadas). De la correcta selección, el buen uso y dominio que tenga el equipo de desarrollo sobre ellas depende considerablemente la duración del proyecto, la completa utilización de sus ventajas y la calidad con que transcurre el proceso en cuestión hasta llegar al producto final. Para el proceso de desarrollo de software se utilizan las tecnologías mostradas a continuación:

### 1.5.1 Metodología AUP

El Proceso Unificado Ágil (AUP, por sus siglas en inglés Agile Unified Process) es una metodología de desarrollo de software que combina técnicas ágiles como Scrum y XP con los principios de Rational Unified Process (RUP). El AUP se centra en la simplicidad, la usabilidad, la adaptación al cambio y la entrega temprana de software funcional. El objetivo principal del AUP es proporcionar un marco de trabajo adaptable que permita a los equipos de desarrollo entregar software de alta calidad de manera más rápida y eficiente. En lugar de seguir una metodología rígida y estructurada, el AUP utiliza una estrategia iterativa e incremental en la que los equipos trabajan en ciclos cortos y entregan un incremento funcional de software al final de cada iteración. Se estará usando el escenario N4[16], [17].

**Escenario N4 de AUP:** Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que no debe poseer demasiada información. Todas las disciplinas antes definidas (desde Modelado de negocio hasta Pruebas de Aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen Iteraciones y se obtengan resultados incrementales. En una iteración se repite el flujo de trabajo de las disciplinas, Requisitos, Análisis y diseño, Implementación y Pruebas internas. De esta forma se brinda un resultado más completo para un producto final de manera creciente. Para llegar a lograr esto, cada requisito debe tener un completo desarrollo en una única iteración.

**Marco Conceptual:**

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

1- Desarrollo Dirigido por Pruebas (test driven development - TDD en inglés)
2- Modelado ágil
3- Gestión de Cambios ágil
4- Refactorización de Base de Datos para mejorar la productividad.

**Fases:**

Al igual que en RUP, el AUP consta de cuatro fases principales, cada fase tiene objetivos específicos y entregables asociados, y las etapas siguientes solo comienzan después de que se hayan completado los objetivos y entregables de la fase anterior. Estas fases son:

1- **Inicio:** El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.

2- **Elaboración:** El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.

3- **Construcción:** Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.

4- **Transición:** El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Esta secuencia ayuda a garantizar que los requisitos del usuario se entiendan claramente antes de que se inicie el desarrollo y que el software se pruebe exhaustivamente antes de su lanzamiento.

**¿Por qué utilizar AUP?**

En cuanto a por qué utilizar el AUP, una de las principales ventajas es que:

- Permite una mayor flexibilidad y adaptabilidad que las metodologías tradicionales. Debido a su naturaleza iterativa e incremental, el AUP puede ajustarse fácilmente a los cambios en los requisitos del usuario o las condiciones del mercado. Además, como enfatiza la entrega temprana de software funcional, los usuarios pueden comenzar a utilizar el software antes y dar retroalimentación valiosa, lo que ayuda a reducir el riesgo de fracaso del proyecto.

**Disciplinas:**

AUP define 7 disciplinas (4 ingenieriles y 3 de gestión de proyectos), las disciplinas son:

1- **Modelo:** El objetivo de esta disciplina es entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio. Agrupa los flujos de trabajos de Modelado de negocio, Requisitos y Análisis y Diseño.
2- **Implementación:** El objetivo de esta disciplina es transformar su modelo (s) en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.
3- **Prueba:** El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.
4- **Despliegue:** El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo este a disposición de los usuarios finales.
5- **Gestión de configuración:** El objetivo de esta disciplina es la gestión de acceso a herramientas de su proyecto. Esto incluye no sólo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.
6- **Gestión de proyectos:** El objetivo de esta disciplina es dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de riesgos[18], la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc), coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.
7- **Entorno:** El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea el adecuado, la orientación (normas y directrices), y herramientas (hardware, software, etc) estén disponibles para el equipo según sea necesario.

**Roles definidos por AUP:**

1- **Administrador de proyecto:** Maneja a los miembros construye relaciones con los stakeholders, coordina interacciones con los stakeholders, planea, maneja y asigna los recursos.
2- **Ingeniero de procesos:** Desarrolla, adapta y apoya sus materiales del proceso del software.
3- **Desarrollador:** Escribe, testea y construye software.
4- **Administrador de Base de Datos:** Diseña, prueba, desarrolla, y apoya los esquemas de la BD.
5- **Modelador ágil:** Crea y desarrolla modelos, bosquejos o los archivos de la herramienta CASE, de una manera evolutiva y de colaboración.
6- **Administrador de la configuración:** Un encargado de la configuración es responsable de proporcionar la infraestructura total y el ambiente del CM al equipo de desarrollo
7- **Stakeholder:** Es un individuo, grupo u organización que tiene un interés o una participación en un proyecto o negocio. Su adecuada gestión puede contribuir grandemente al resultado final del proyecto.
8- **Administrador de pruebas:** Responsables del éxito de la prueba, incluyendo el planeamiento, la gerencia, y la defensa para la prueba y las actividades de la calidad.
9- **Probador:** Encargado de ejecutar las pruebas.

### 1.5.2 Arquitectura del sistema (Cliente-Servidor)

La arquitectura cliente-servidor es un modelo de diseño para sistemas digitales donde el procesamiento y la responsabilidad están divididos entre dos partes: el cliente y el servidor. El cliente es la interfaz de usuario en la que se realizan las operaciones y solicita información del servidor. El servidor, por otro lado, es el backend encargado de gestionar la lógica de negocio, es responsable de proporcionar los servicios, almacenar datos, realizar las operaciones solicitadas por el cliente y proporcionar los datos a través de la API.

Este modelo ofrece varias ventajas significativas, tales como:

1- Permite una mayor escalabilidad ya que es posible agregar o eliminar clientes sin afectar el procesamiento de datos en el servidor. Además, las actualizaciones pueden ser realizadas en el servidor sin requerir cambios en todos los clientes conectados.
2- La seguridad, ya que el servidor puede controlar y restringir el acceso de los usuarios al sistema, como también puede asegurar la calidad y consistencia de los datos utilizados por el sistema.
3- Una arquitectura cliente-servidor reduce el costo total del sistema a largo plazo. Los clientes se pueden utilizar con hardware menos potente y nivel de complejidad más bajo, lo que significa que son menos costosos de desplegar. El servidor puede mantener un mínimo histórico de cambios/operaciones que ha realizado, facilitando la búsqueda de errores y fallas.

Estas son algunas de las razones principales por las que la arquitectura cliente-servidor es una buena elección para desarrollar sistemas digitales.

### 1.5.3 Patrón de arquitectura Modelo-Vista-Controlador (MVC)

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. Estos componentes son:[19]

1- **Modelo:** representa la lógica de negocio, los datos y las reglas del dominio en tu aplicación.
2- **Vista:** proporciona la interfaz de usuario que permite a los usuarios interactuar con el modelo.
3- **Controlador:** maneja las entradas del usuario y coordina la interacción entre la vista y el modelo.

La idea detrás del patrón MVC es mantener cada uno de estos componentes lo más independiente posible, permitiendo realizar cambios sin afectar al resto de la aplicación. Por ejemplo, podrías cambiar o actualizar la vista (la interfaz de usuario) sin modificar el modelo o el controlador.

Entre las ventajas del patrón de arquitectura MVC, podemos encontrar:

1- **Separación de preocupaciones:** El patrón MVC facilita la separación de preocupaciones y promueve el modularidad dentro de tu aplicación. Esto significa que puedes centrarte en desarrollar cada componente sin tener que preocuparte por cómo afectará esto a los demás.
2- **Reutilización de código:** Al separar los componentes de la aplicación, podrías reutilizarlos en otros proyectos o áreas diferentes dentro de tu proyecto.
3- **Testabilidad:** Es más fácil probar cada componente de forma individual, lo que facilita el mantenimiento y la corrección de errores.
4- **Escalabilidad:** Cuando una aplicación crece, gracias a la estructura que proporciona el patrón MVC será más sencillo añadir nuevas funcionalidades o componentes, ya que no se afectará el resto de la aplicación.

Además, al utilizar el patrón MVC, se aplica una arquitectura probada y utilizada comúnmente en el desarrollo de software. Se tendrá una mayor facilidad para separar preocupaciones, diseñar módulos independientes y realizar pruebas adecuadas que aseguren el correcto funcionamiento de tu software.

### 1.5.4 Entornos de desarrollo y herramientas utilizadas

**Visual Studio Code:** Es un entorno de desarrollo integrado (IDE) de código abierto desarrollado por Microsoft. Es una herramienta muy popular y avanzada para el desarrollo de aplicaciones web y móviles. Visual Studio Code tiene una gran cantidad de extensiones, lo que lo hace altamente personalizable para satisfacer las necesidades de cualquier tipo de proyecto[20].

Algunas de las ventajas de usarlo son:

1- Es un editor de código fuente rápido y altamente personalizable.

2- Permite la integración de diferentes lenguajes de programación, frameworks y herramientas.

3- Cuenta con una amplia biblioteca de extensiones que permite ampliar sus funcionalidades.

4- Dispone de características interesantes como depuración de código

**Table Plus:** Es un cliente de base de datos moderno, fácil de usar y potente, que permite a los usuarios interactuar con sus bases de datos y trabajar adecuadamente.[21]

Algunas de las ventajas de usarlo son:

1- Interfaz de usuario amigable, es fácil de usar y muestra información de base de datos de manera clara y sencilla.

2- Conectividad de base de datos, ya que Table Plus admite conexión con casi cualquier base de datos.

3- La velocidad, ya que es conocido por ser bastante rápido y utiliza un motor de alta calidad y muy optimizado.

4- Puedes realizar múltiples consultas a la vez.

5- Autocompleta código de SQL y consultas.

6- Tiene una serie de herramientas de administración útiles, desde la creación de copias de seguridad hasta la reorganización de tablas.

**Axure RP:** Es una poderosa herramienta de diseño y prototipado que permite a los diseñadores y equipos de desarrollo crear prototipos interactivos y especificaciones de diseño de alta fidelidad. Con Axure RP, los diseñadores pueden colaborar, iterar y validar el diseño de sus productos antes de comenzar la implementación[22].

Ventajas:

1- Creación de prototipos interactivos y fidelidad: Axure RP permite crear prototipos interactivos de alta fidelidad que simulan la experiencia de uso de un producto real. Puedes agregar interacciones, animaciones, transiciones y comportamientos avanzados para mostrar y probar cómo funcionará tu diseño en una etapa temprana.

2- Colaboración y comentarios en tiempo real: Axure RP facilita la colaboración entre diseñadores, desarrolladores y stakeholders. Puedes compartir tus proyectos en línea y permitir que los usuarios proporcionen comentarios directamente en el prototipo. Esto facilita la iteración del diseño y garantiza que se capturen las necesidades y expectativas de todos los involucrados.

3- Documentación y especificaciones de diseño: Axure RP permite generar automáticamente documentación y especificaciones de diseño detalladas. Puedes crear diagramas de flujo, mapas de sitio, anotaciones y otros documentos que ayuden a los desarrolladores a comprender y construir correctamente tu diseño.

**Thunder Client**: Admite colecciones, entornos, colaboración git y almacenamiento local. Es una extensión de Visual Studio liviana y más que sencilla para simular llamadas a las Rest Client API.[23]

Algunas de las ventajas de utilizar Thunder Client son las siguientes:

1- Integración perfecta en Visual Studio Code: Al ser una extensión de Visual Studio Code, Thunder Client se integra perfectamente en el entorno de desarrollo, lo que permite realizar pruebas de API directamente desde el editor de código

2- Interfaz amigable y orientada al desarrollo: Thunder Client ofrece una interfaz de usuario amigable y orientada al desarrollo, con características que facilitan la creación, ejecución y gestión de solicitudes y pruebas de API de manera eficiente.

3- Colecciones de solicitudes: Permite organizar las solicitudes en colecciones, lo que simplifica la gestión y ejecución de conjuntos de pruebas y solicitudes relacionadas.

4- Compartir y exportar: Facilita el intercambio de colecciones y solicitudes a través de la función de exportación, lo que permite compartir configuraciones y pruebas con otros miembros del equipo.

5- Autocompletado y resaltado de sintaxis: Ofrece autocompletado inteligente y resaltado de sintaxis para facilitar la elaboración de solicitudes y pruebas con mayor precisión y eficacia.

**Visual Paradigm:** Es un software de modelado de sistemas que permite a los usuarios crear diferentes tipos de diagramas, como los mencionados anteriormente.[24]

Algunas ventajas de usarlo son:

1- Facilidad de uso: Visual Paradigm tiene una interfaz intuitiva y fácil de usar, lo que hace que sea accesible para usuarios con diferentes niveles de experiencia en modelado.

2- Variedad de diagramas: Visual Paradigm permite crear diferentes tipos de diagramas, lo que hace que sea útil para representar diferentes aspectos de un sistema.

### 1.5.5 Lenguajes de programación utilizados

**HTML 5:** Lenguaje de Marcas de Hipertexto (del inglés ***H****yper****T****ext* ***M****arkup* ***L****anguage*) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Es el lenguaje con el que se diseñan las páginas web. Estas páginas constituyen una forma eficaz de comunicación capaz de llegar a millones de personas. Una página web es un archivo con texto en el que se irán insertando etiquetas HTML, para que ese contenido pueda ser interpretado en el navegador web[25].

Algunas de las ventajas de usarlo son:

1- Permite la inclusión de imágenes, videos y otros elementos multimedia.

2- Es fácil de aprender, ya que utiliza una sintaxis sencilla y clara.

**CSS 3:** Hojas de Estilo en Cascada (del inglés **C***ascading* **S***tyle* **S***heets*) es el lenguaje de estilos utilizado para describir la presentación de documentos HTML. Entonces se

puede decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. Este lenguaje es principalmente utilizado por parte de los navegadores web de internet y por los programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc. La filosofía de CSS se basa en intentar separar lo que es la estructura del documento HTML de su presentación. Por decirlo de alguna manera: la página web sería lo que hay debajo (el contenido) y CSS sería un cristal de color que hace que el contenido se vea de una forma u otra.[26]

Algunas de las ventajas de usarlo son:

1- Proporciona herramientas para crear hojas de estilo, que permiten aplicar estilos a todo el sitio web de manera consistente.

2- Permite la creación de animaciones y efectos visuales avanzados.

**JS o ECMAScript 2021:** (**J**ava**S**cript) Es un lenguaje de programación ligero, interpretado, o compilado *just in time* (no necesita compilador, porque el motor del navegador lo hace instantáneamente) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, es muy flexible y permite crear efectos visuales y animaciones, modificar el contenido de una página sin necesidad de recargar la misma, validar formularios, entre otros.[27]

Algunas de sus características son:

1- Es un lenguaje de programación utilizado para crear interactividad y dinamismo en páginas web.

2- Permite la creación de aplicaciones web complejas.

3- Es compatible con la mayoría de los navegadores y plataformas.

4- Las funciones en JavaScript se pueden asignar a variables, pasar como argumentos a otras funciones o devolver como resultado de una función.

5- Tipado dinámico: JavaScript es un lenguaje de tipado dinámico, lo que significa que no es necesario declarar el tipo de variables antes de usarlas. También permite la conversión automática de tipos en tiempo de ejecución.

6- Asincronismo: JavaScript soporta operaciones asincrónicas, lo cual es muy útil para realizar tareas que pueden tardar en tiempo, como obtener datos de una base de datos o hacer una solicitud HTTP.

### 1.5.6 Otras tecnologías o librerías utilizadas

**TS 5.2:** (**T**ype**S**cript) es un super set de JavaScript, algunos lo consideran lenguaje de programación, pero en realidad es un conjunto de reglas que se le aplican al código de JavaScript para que se escriba de una forma tipada, esto quiere decir que proporciona características adicionales como la validación de tipos y la creación de clases, este código obtenido se transpila a JavaScript al ejecutarse, fue creado por Microsoft. Se utiliza principalmente para la creación de aplicaciones web complejas. Permite hacer el trabajo en equipo mucho más asequible, y ayuda a los desarrolladores a escribir un código más robusto y escalable.[28]

Algunas de sus principales características son:

1- Es un superconjunto de JavaScript que añade características como tipos de datos estáticos opcionales, clases y módulos.

2- Aumenta la capacidad de mantenimiento del código y reduce la posibilidad de errores de programación.

3- Es compatible con JavaScript y se puede utilizar en cualquier entorno donde se pueda usar JavaScript.

**Prisma:** Es un ORM moderno y potente para bases de datos SQL y NoSQL que se integra fácilmente con lenguajes y frameworks populares como TypeScript, Node.js, React y GraphQL. Prisma utiliza un modelo de datos declarativo para definir la estructura

de la base de datos, lo que facilita la creación, modificación y eliminación de tablas y relaciones. Además, Prisma ofrece características avanzadas como consultas anidadas, transacciones y generación automática de esquemas GraphQL. En resumen, Prisma simplifica el acceso y manipulación de datos en aplicaciones modernas y escalables. [29]

Un ORM (Object-Relational Mapping) es una técnica de programación que permite mapear los objetos de una aplicación a las tablas de una base de datos relacional, simplificando así el acceso y manipulación de datos.

La relación entre un ORM y una base de datos es que el ORM actúa como una capa intermedia entre la aplicación y la base de datos, permitiendo a la aplicación trabajar con objetos en lugar de tener que interactuar directamente con la base de datos.

### 1.5.7 Frameworks utilizados

**React 18:** Es un framework de JavaScript utilizado para la creación de interfaces de usuario. Permite la creación de componentes reutilizables para crear interfaces de usuario altamente escalables. React[30] es una forma de mezclar código HTML (o XML) con JavaScript (JS) por eso su extensión es JSX, una mezcla de ambos, también se puede usar con TypeScript, con la extensión TSX. Está hecho para trabajar con aplicaciones de todo tipo de magnitud, desde muy sencillas hasta muy complejas, siempre se recomienda su uso. Es muy eficiente, es predecible, puedes usar muchos patrones, no solo uno, y hacer la construcción del código de maneras diferentes debido a la gran cantidad de librerias. React se utiliza para construir aplicaciones web, incluyendo algunas aplicaciones móviles utilizando React Native.

Algunas de sus ventajas y características son:

1- Es una biblioteca de JavaScript que se utiliza para crear interfaces de usuario (UI) con la que se pueden construir aplicaciones web de forma sencilla.

2- Los componentes de React son reutilizables, lo que significa que puedes crear un componente y utilizarlo en una variedad de lugares en una aplicación.

3- El enfoque declarativo de React hace que la identificación y resolución de errores sea más fácil que con otras metodologías de programación imperativa.

4- La arquitectura de React basada en componentes ayuda a construir aplicaciones escalables y mantenibles.

**Tailwind CSS 3:** Tailwind CSS es un framework de CSS, se usa para crear interfaces de usuario personalizadas de forma rápida y eficiente. Usado para diseñar páginas web front-end y proporciona una serie de clases predefinidas que pueden ser usadas para estilizar los elementos de la página.[31]

Algunas de sus principales ventajas y características son:

1- Diseño predefinido: Tailwind CSS contiene una gran cantidad de clases CSS predefinidas para dar estilo a todos los elementos de la interfaz.

2- Enfoque utilitario: Las clases en Tailwind CSS están destinadas a usarse como utilidades, lo que significa que proporcionan propiedades específicas del diseño, como el tamaño del margen, la tipografía, el color, etc.

3- Personalizable: Aunque ofrece muchas clases predefinidas, Tailwind permite personalizar el archivo de configuración para establecer propiedades personalizadas.

4- Fácil de entender: La nomenclatura utilizada en las clases de Tailwind CSS es intuitiva y fácil de entender, lo que facilita la comprensión para los desarrolladores.

5- Buen rendimiento: Tailwind CSS está orientado a mejorar el rendimiento de las páginas mediante la reducción del tamaño de los archivos CSS.

6- Compatibilidad con otros frameworks: Tailwind CSS puede ser facilmente integrado con otros frameworks como React, Vue.js, Angular, etc.

**NextJS 13.4:** Es un framework de React para el desarrollo de aplicaciones web, tanto estáticas como dinámicas. Es fácil de usar, rápido y ofrece una experiencia de desarrollo moderna para los desarrolladores web. Cuenta con muchísimas ventajas sobre la competencia en cuanto a programación web, algunas las listare a continuación[32].

Algunas de las ventajas de usarlo son:

1- Amplia documentación, ya que NextJS cuenta con una gran ayuda para que los desarrolladores puedan aprender rápidamente y utilizar el framework para desarrollar sus proyectos.

2- Hot realoading, o cambios en caliente, significa que las páginas se refrescan automáticamente a medida que se vayan haciendo cambios en el código.

3- Cuenta con enrutamiento automático, cada página se añade automáticamente a las rutas de navegación de la aplicación lo que facilita este proceso.

4- Está basado en React, lo que significa que pueden usar las características y sintaxis de React, e implementarlo en TypeScript.

5- Aplica mejoras a muchas herramientas y componentes usadas en la programación web, como imágenes, ofuscación de código, y añade carga perezosa (Lazy loading) a la mayoría de componentes de forma automática haciendo la aplicación web mucho más ligera.

### 1.5.8 Sistema Gestor de Base de Datos

Un sistema gestor de base de datos (DBMS, por sus siglas en inglés) es un software encargado de crear, modificar y gestionar bases de datos de manera eficiente y segura. PostgreSQL es uno de los DBMS más populares que existe, debido a lo avanzado y potente que es.

**PostgreSQL 13:** Es un sistema de gestión de bases de datos relacionales de código abierto que se utiliza para almacenar y gestionar grandes cantidades de datos. PostgreSQL es uno de los sistemas más avanzados y potentes disponibles, y es utilizado por empresas y organizaciones de todo el mundo.[33]

Algunas de sus características son:

1- SQL completo: soporta SQL completo, incluyendo subconsultas, transacciones, integridad referencial y vistas.

2- Extensibilidad: es altamente extensible y permite a los desarrolladores crear funciones personalizadas y tipos de datos.

3- Replicación: puede ser utilizado para crear sistemas de alta disponibilidad y de recuperación de desastres,

4- Soporta diferentes plataformas: puede ser usado tanto en Windows y MacOS como en Linux.

5- Soporte para grandes objetos: significa que puede ser utilizado para almacenar y gestionar archivos multimedia, como videos e imágenes.

Algunas de sus ventajas son:

1- Escalabilidad: PostgreSQL es altamente escalable y puede manejar grandes cantidades de datos y usuarios sin comprometer el rendimiento.

2- Fiabilidad: es muy fiable, lo que significa que puede procesar grandes cantidades de datos sin perdidas de información.

3- Seguridad: tiene una amplia variedad de características de seguridad, incluyendo autenticación, autorización y cifrado de datos.

4- Flexibilidad: puede ser utilizado en una gran variedad de entornos y situaciones.

5- Comunidad de usuarios: debido a que cuenta con una gran comunidad, estos trabajas en conjunto para mejorar el software y proporcionar soporte y ayuda a los usuarios.

## 1.6 Conclusiones

- Durante el desarrollo de este capítulo se definieron los principales conceptos asociados al domino del problema para lograr un mayor entendimiento del mismo. Se realizó un análisis crítico del flujo actual de los procesos donde se detectó que existen dificultades para el acceso a la información en el proceso de gestión del Control Interno y Activos Fijos Tangibles.

- El estudio de los sistemas existentes, el funcionamiento y flujo de los procesos actuales permitieron identificar la necesidad de un sistema que permita ese acceso de forma más libre a los administradores y jefes de Áreas.
- Se analizaron las metodologías, tecnologías, lenguajes y herramientas necesarias para el desarrollo del sistema propuesto, concluyendo utilizar AUP como metodología de desarrollo del software, Visual Studio Code como entorno de desarrollo de programación, HTML, CSS, JavaScript y TypeScript como lenguajes de programación, Tailwind CSS, React y NextJs como frameworks y PostgresSQL como sistema gestor de base de datos.

## Capítulo 2: Propuesta de solución

En este capítulo se da una breve descripción del sistema, también se definen los requisitos funcionales y no funcionales, así como los elementos o artefactos creados para el cuarto escenario en la metodología AUP en las etapas de análisis, diseño e implementación.

### 2.1. Concepción General del Sistema

La idea general con la que fue concebido este producto de software fue la de facilitar el acceso a la información de los Activos Fijos. Este está compuesto por dos partes, una API que sirve como backend, y la aplicación web creada con Next que funciona como frontend.

Esta aplicación proporcionará información a los usuarios sobre: Las Áreas de Responsabilidad, Áreas Administrativas, Locales, Activos Fijos Tangibles, Trabajadores, jefes de Áreas Administrativas e Inspecciones.

Permite entre sus funcionalidades, la autenticación, con dos roles, jefe de Área Administrativa y administrador general, cada uno tiene diferentes permisos dentro del sistema, dejando gestionar los activos, áreas, trabajadores y demás elementos de la base de datos.

Utiliza un diseño desacoplado, y la arquitectura está compuesta por 3 capas, modelo, vista y controlador, donde el modelo es el encargado de manejar los datos de la

aplicación y las consultas, el controlador es el que gestiona los procesos y realiza acciones con los datos para permitir que estos lleguen de manera segura a la interfaz, también se encarga de actualizar el estado de la aplicación, las rutas y seguridad, y la vista es la interfaz gráfica que maneja el usuario, además la interfaz de la aplicación es amigable y sencilla, permitiendo un fácil manejo del producto.

La API está creada en Next también, tiene protección de rutas, y sirve para mediante la ORM comunicar la salida y entrada de datos a la base de datos de PostgreSQL.

## 2.2. Requisitos de Software

La etapa análisis de requisitos es la más importante en el desarrollo del proyecto informático porque define lo que será capaz de hacer el sistema. Se deben identificar de manera clara las descripciones de los casos de uso del dominio, así como puntualizar con el usuario el comportamiento y funcionalidad que necesita el programa.

### 2.2.1 Requisitos Funcionales

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones[34].

| Nro. | Nombre | Descripción | Prioridad | Complejidad |
|---|---|---|---|---|
| RF1 | Autenticar Usuario | El usuario introduce sus datos, el sistema comprueba y si son válidos se deja entrar con el rol que corresponda al usuario. | Alta | Media |
| RF2 | Crear Área Administrativa | Los usuarios con rol de administrador pueden crear Áreas Administrativas, estas | Alta | Media |

| Nro. | Nombre | Descripción | Prioridad | Complejidad |
|---|---|---|---|---|
| | | pertenecen a un Centro de Costo | | |
| RF3 | Modificar Área Administrativa | Los usuarios con rol de administrador pueden modificar los datos de las Áreas Administrativas | Media | Baja |
| RF4 | Eliminar Área Administrativa | Los usuarios con rol de administrador pueden eliminar Áreas Administrativas | Media | Baja |
| RF5 | Listar Áreas Administrativas | Los usuarios con rol de administrador pueden ver la información de las Áreas Administrativas | Media | Baja |
| RF6 | Crear Usuario | Un usuario con rol de administrador puede crear nuevos usuarios, definiendo su nombre, correo, Área Administrativa que maneja, contraseña y rol | Alta | Media |
| RF7 | Modificar Usuario | Un usuario administrador puede modificar la información de otros usuarios | Media | Baja |

| Nro. | Nombre | Descripción | Prioridad | Complejidad |
|---|---|---|---|---|
| RF8 | Eliminar Usuario | Un usuario administrador puede eliminar otros usuarios. | Media | Baja |
| RF9 | Listar Usuarios | El sistema permite que los usuarios con rol de administrador vean todos los usuarios existentes en el sistema | Media | Media |
| RF10 | Crear Área de Responsabilidad | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden crear nuevas Áreas de Responsabilidad asociadas a un Área Administrativa y Centro de Costo | Alta | Alta |
| RF11 | Modificar Área de Responsabilidad | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden modificar la información de las Áreas de Responsabilidad | Media | Baja |
| RF12 | Eliminar Área de Responsabilidad | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden eliminar Áreas de Responsabilidad | Media | Baja |

| Nro. | Nombre | Descripción | Prioridad | Complejidad |
|---|---|---|---|---|
| RF13 | Listar Áreas de Responsabilidad de un Área Administrativa | Los usuarios con rol de jefe de Área Administrativa pueden ver la información de sus Áreas de Responsabilidad. | Alta | Media |
| RF14 | Listar todas las Áreas de Responsabilidad | Los usuarios con rol de administrador pueden ver la información de todas las Áreas de Responsabilidad | Alta | Media |
| RF15 | Crear Local | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden crear Locales y asignarlos a Áreas de Responsabilidad | Alta | Media |
| RF16 | Modificar Local | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden modificar la información de los Locales. | Media | Media |
| RF17 | Eliminar Local | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden eliminar Locales. | Media | Baja |
| RF18 | Listar Locales de un Área de Responsabilidad | Los usuarios con rol de jefe de Área Administrativa pueden ver la información de los Locales existentes en las | Media | Baja |

| Nro. | Nombre | Descripción | Prioridad | Complejidad |
|---|---|---|---|---|
| | | Áreas de Responsabilidad que le pertenecen. | | |
| RF19 | Listar todos los Locales | Los usuarios con rol de Administrador pueden ver la información de todos los Locales. | Media | Media |
| RF20 | Crear Activos Fijos | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden crear Activos Fijos y asignarlos a Áreas Administrativas y a Locales. | Alta | Alta |
| RF21 | Modificar Activos Fijos | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden modificar la información de los Activos Fijos. | Media | Baja |
| RF22 | Eliminar Activos Fijos | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden eliminar los Activos Fijos. | Media | Baja |
| RF23 | Listar Activos Fijos de un Área Administrativa | Los usuarios con rol de jefe de Área Administrativa pueden ver la información de los Activos Fijos que le pertenecen. | Media | Media |

| Nro. | Nombre | Descripción | Prioridad | Complejidad |
|---|---|---|---|---|
| RF24 | Listar Activos Fijos totales | Los usuarios con rol de Administrador pueden ver la información de todos los Activos Fijos. | Media | Media |
| RF25 | Crear Trabajador | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden crear Trabajadores y añadirlos a un Área Administrativa y a un Local. | Alta | Media |
| RF26 | Modificar Trabajador | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden modificar la información de los Trabajadores existentes. | Media | Baja |
| RF27 | Eliminar Trabajador | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden eliminar los Trabajadores. | Media | Baja |
| RF28 | Listar Trabajadores de un Área Administrativa | Los usuarios con rol de jefe de Área Administrativa pueden ver la lista de trabajadores que hay en su Área. | Media | Media |

| Nro. | Nombre | Descripción | Prioridad | Complejidad |
|---|---|---|---|---|
| RF29 | Listar todos los Trabajadores | Los usuarios con rol de Administrador pueden ver todos los Trabajadores. | Media | Media |
| RF30 | Crear Inspección | Los usuarios con rol de jefe de Área Administrativa pueden crear Inspecciones, se le harán automáticamente a su Área Administrativa asociada. | Alta | Alta |
| RF31 | Modificar Inspección | Los usuarios con rol de jefe de Área Administrativa o Administrador pueden modificar la información de las inspecciones existentes. | Alta | Alta |
| RF32 | Eliminar Inspecciones | Los usuarios con rol de jefe de Área Administrativa pueden eliminar las Inspecciones. | Media | Media |
| RF33 | Listar Inspecciones | Los usuarios con rol de jefe de Área Administrativa pueden ver la información de las Inspecciones. | Media | Media |
| RF34 | Ver información general | Los usuarios con rol de jefe de Área Administrativa pueden ver un resumen de la información que tiene que | Alta | Alta |

| Nro. | Nombre | Descripción | Prioridad | Complejidad |
|---|---|---|---|---|
| | | ver con su Área Administrativa. | | |
| RF35 | Desconectar del sistema | El usuario puede salir de este sistema cuando termine su trabajo. | Media | Baja |

### 2.2.2 Requisitos No Funcionales

Los requisitos no funcionales son las restricciones o los requisitos impuestos al sistema. Especifican el atributo de calidad del software. Los requisitos no funcionales se ocupan de problemas como la escalabilidad, la mantenibilidad, el rendimiento, la portabilidad, la seguridad, la confiabilidad y muchos más. Los requisitos no funcionales abordan cuestiones vitales de calidad para los sistemas de software[35].

| Nro. | Descripción | Tipo de Requisito |
|---|---|---|
| RNF1 | El sistema presenta una interfaz sencilla, de fácil acceso para los usuarios. El diseño dará la posibilidad de una fácil navegación hacia cualquier lugar del mismo. | Interfaz |
| RNF2 | El subsistema indica directamente los datos que se deben introducir, además de los campos que deben ser completados. | Usabilidad |
| RNF3 | El sistema se puede usar mediante cualquier navegador disponible. | Portabilidad |

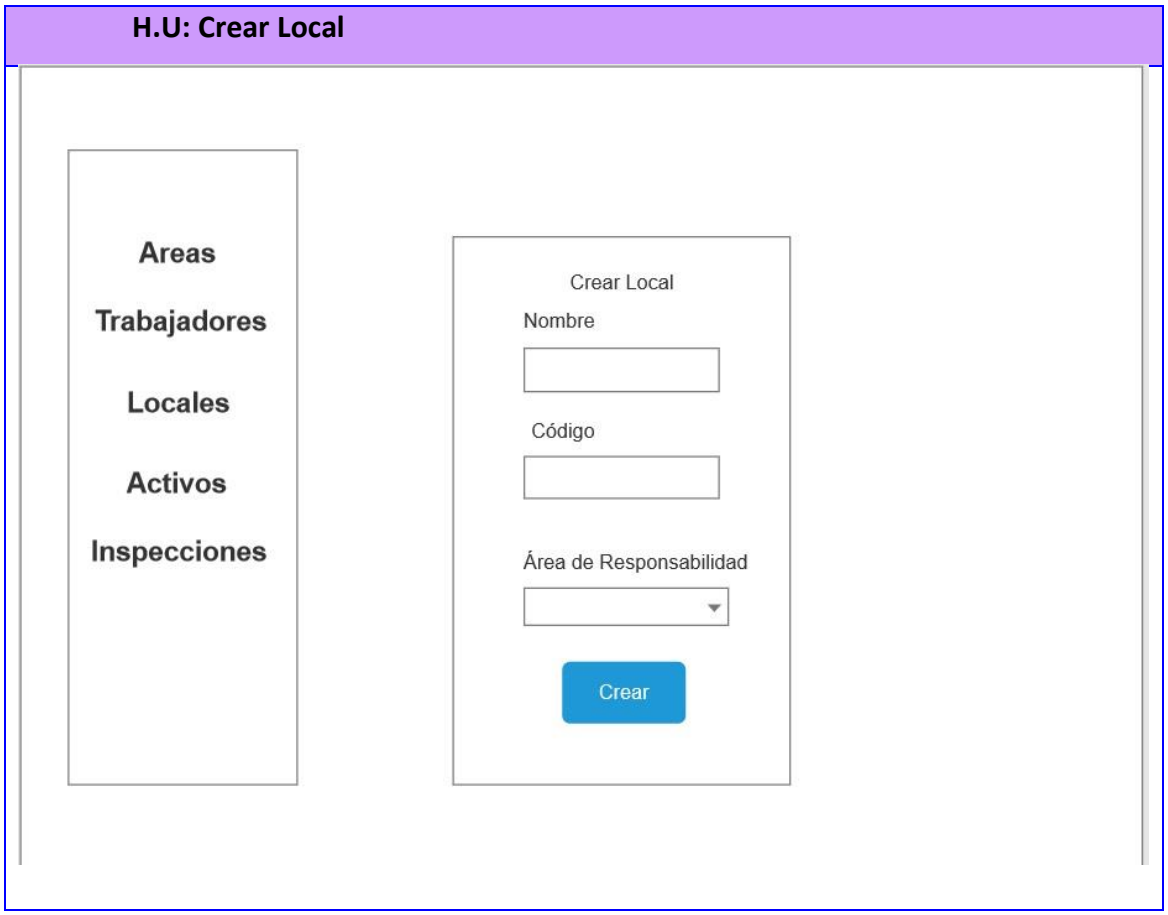
| Nro. | Descripción | Tipo de Requisito |
|---|---|---|
| RNF4 | El sistema procesa la información de la manera más rápida posible. | Rendimiento |
| RNF5 | Para poder desplegar el servidor se necesita tener instalador Postgres o un servidor de este y una versión moderna de NextJs con todos los módulos de NodeJs especificados en package.json. | Software |
| RNF6 | El sistema está protegido por contraseñas, las rutas están protegidas para que no se pueda intentar violentar la autentificación. | Confidencialidad |
| RNF7 | El sistema cumple con los requerimientos legales especificados por los que se rige el proceso de Control Interno en la Universidad de Cienfuegos. | Legales |

### 2.3. Historias de Usuario

Las historias de usuarios son herramientas que se emplean para describir detalladamente los requerimientos del software, y son similares a los casos de uso que se utilizan en el proceso unificado. Estas historias son la base para llevar a cabo las pruebas funcionales.[36]

| **H.U: Crear Local** | |
|---|---|
| **Número: 15** | **Requisito:** Crear Local |
| **Programador:**<br><br>Alexis Manuel Hurtado García | **Iteración Asignada:**<br><br>Primera Iteración |

| H.U: Crear Local | |
|---|---|
| **Prioridad:** Alta | **Tiempo Estimado:** 3 horas |
| **Riesgo en Desarrollo:** Media | **Tiempo Real:** 5 horas |
| **Descripción:** El usuario con rol de administrador o jefe de Área Administrativa introduce los datos del Local a crear, si cumple con la condición entonces se crea. En caso de que no sean válidos, mostrará un mensaje de error.<br><br>**Campos:**<br><br>**Código**: Campo de texto para el código asociado al Local, es único y obligatorio.<br><br>**Nombre:** Campo de texto para introducir el nombre del Local. Es obligatorio.<br><br>**Área de Responsabilidad:** Es un campo de selección para elegir el Área de Responsabilidad a la que pertenece este Local.<br><br>**Botones:**<br><br>**Crear:** Al hacer clic en el botón, la aplicación verificará los datos, y creará el Local. | |
| **Prototipo elemental de interfaz gráfica de usuario:** | |




## 2.4. Descripción de la Arquitectura

El diseño del software se basa en una arquitectura que establece las ideas principales y proporciona una descripción detallada de cómo construir el sistema.

Next.js no impone un patrón de arquitectura específico, sino que proporciona una estructura flexible para que los desarrolladores puedan organizar su código de la manera que mejor se adapte a sus necesidades. Sin embargo, Next.js se integra bien con el patrón de arquitectura Modelo-Vista-Controlador (MVC)[37].

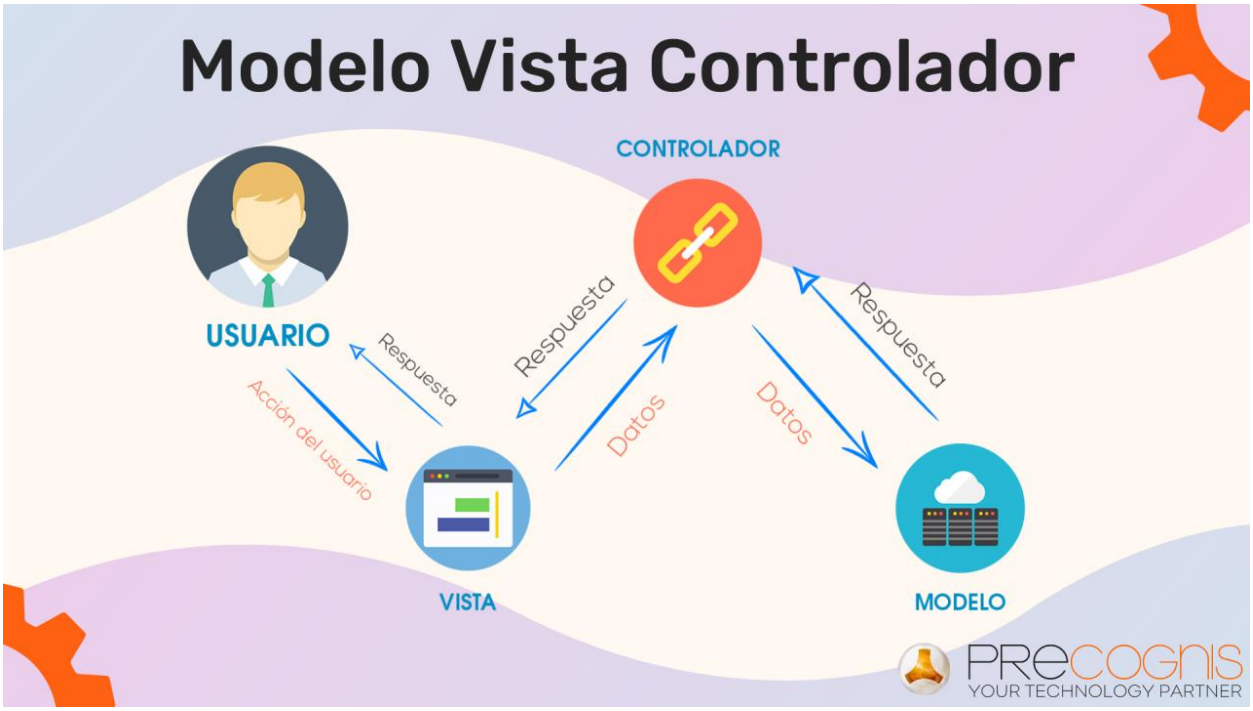


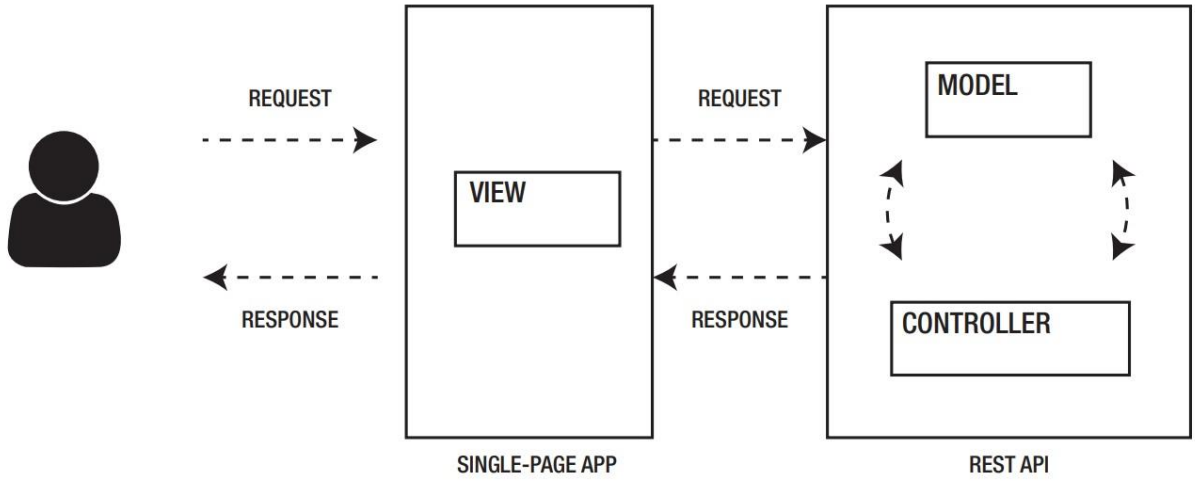

*Imagen 1: Patrón de Modelo - Vista - Controlador*

**Modelo**: Los modelos son generalmente representados por los archivos de datos o servicios que interactúan con la capa de acceso a datos. Son los encargados de obtener información de la base de datos.

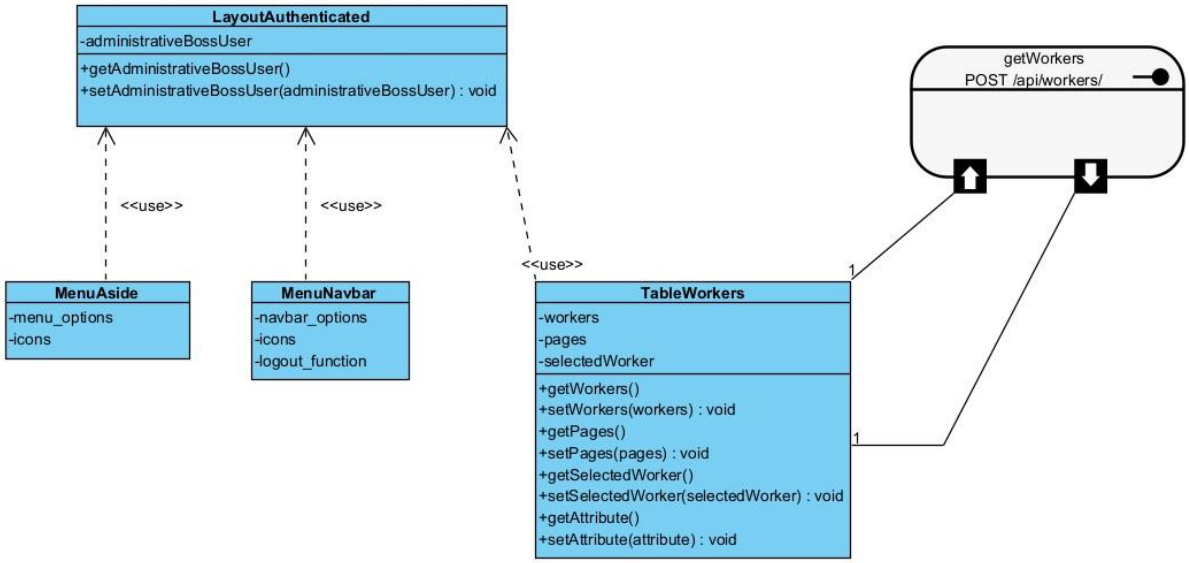
**Vista:** En este caso las vistas son generalmente los componentes de React que representan la interfaz de usuario y manejan la lógica relacionada con la presentación. Estos componentes están ubicados en la carpeta *components* y *app* y son reutilizados en diferentes partes de la aplicación.

**Controlador:** El enrutador de Next.js o la carpeta app/ cumple el rol de controlador, también cabe destacar que los controladores responden a eventos cuando se hace alguna solicitud. Maneja las rutas y las solicitudes del usuario, y se encarga de dirigir las solicitudes a la vista adecuada y mostrar la información de los modelos correspondientes. Hace de intermediario entre la capa Vista y Modelo.

Este proyecto presenta una arquitectura desacoplada, donde tenemos una API REST. Una API REST es una interfaz de comunicación entre sistemas de información que usa el **protocolo de transferencia de hipertexto** (*hypertext transfer protocol* o HTTP, por sus siglas en inglés) para obtener datos o ejecutar operaciones sobre dichos datos en diversos formatos, como pueden ser XML o JSON, se decidió usarla debido a que se usa una arquitectura cliente servidor, y está orientada a recursos usando las operaciones estándar de los verbos HTTP.[38], [39]



## 2.5. Diagramas de Clases




## 2.6. Patrones de Diseño Utilizados

Los patrones de diseño de software son soluciones probadas y comprobadas a problemas comunes en el desarrollo de software. Son enfoques de diseño que ayudan a los desarrolladores a resolver problemas recurrentes de manera eficiente y efectiva. Estos patrones capturan las mejores prácticas y la experiencia acumulada de la industria del desarrollo de software.

Al usar Next como base para el proyecto se usan unos patrones de diseño poco comunes en otros entornos, estos son:

**Render Props:** Este patrón permite pasar una función o componente como prop a otro componente, lo que permite compartir lógica y comportamiento entre componentes. Esto es especialmente útil cuando se desea reutilizar lógica entre varios componentes sin crear una jerarquía compleja de componentes anidados.[40]

**High Order Component (HOC):** Este patrón permite envolver un componente existente en un componente de orden superior para proporcionar funcionalidades adicionales. Los HOCs se utilizan comúnmente para agregar funcionalidad de autenticación, manejo de errores, manejo de permisos, etc. a los componentes de Next.js[41].

**Server Side Rendering (SSR):** permite a Next.js renderizar las páginas en el servidor antes de enviarlas, lo que mejora la velocidad de carga y la experiencia del usuario.[42]

Como base para la arquitectura se usó el patrón **Modelo Vista Controlador**.

También se basó la estructura del proyecto en **Los principios SOLID y Clean Code:**

Estos son dos enfoques importantes en el desarrollo de software que buscan mejorar la calidad, mantenibilidad y escalabilidad del código. El principio SOLID es un acrónimo que representa cinco principios de programación orientada a objetos: Responsabilidad Única (**S**ingle Responsability Principle), Abierto/Cerrado (**O**pen/Close Principle), Sustitución de Liskov (**L**iskov Sustitution Principle), Segregación de Interfaces (**I**nterface Segregation Principle) e Inversión de Dependencia (**D**ependency Inversion Principle). Por otro lado, el principio Clean Code se enfoca en la legibilidad y mantenibilidad del código, promoviendo prácticas como la simplicidad, la claridad, la consistencia y la eliminación de duplicación[43].

Los principios que componen SOLID son:

- Responsabilidad Única (SRP) establece que una clase o módulo debe tener una sola responsabilidad y razón para cambiar. Esto significa que cada componente del sistema debe tener una tarea clara y específica, lo que facilita su mantenimiento y evolución.

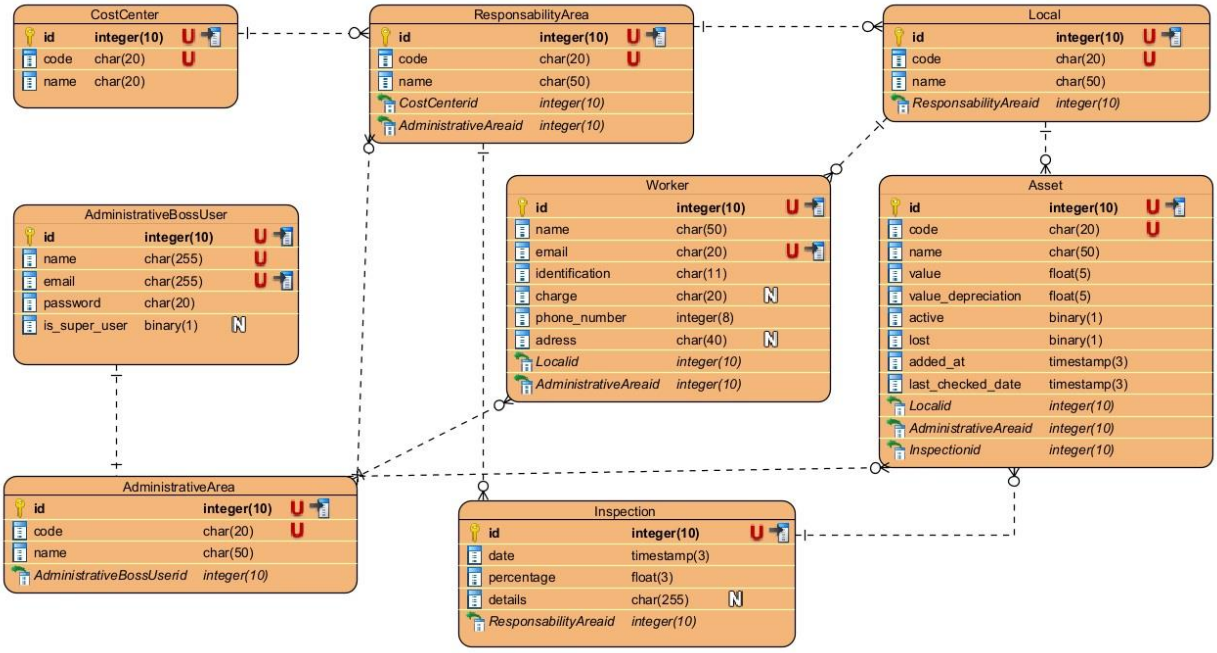
- Abierto/Cerrado (OCP) establece que una clase o módulo debe ser abierto para extensión, pero cerrado para modificación. Esto significa que se deben poder agregar nuevas funcionalidades sin modificar el código existente, lo que evita efectos secundarios no deseados y mejora la escalabilidad del sistema. Este va de la mano con tener una alta cohesión y un bajo acoplamiento.

- Sustitución de Liskov (LSP) establece que una subclase debe ser sustituible por su superclase sin alterar el comportamiento del programa. Esto significa que las clases deben seguir una jerarquía coherente y cumplir con las mismas expectativas de comportamiento.

- Segregación de Interfaces (ISP) establece que una clase no debe depender de interfaces que no utiliza. Esto significa que las interfaces deben ser cohesivas y específicas para cada componente, lo que evita acoplamiento innecesario y mejora la mantenibilidad del sistema.
- Inversión de Dependencia (DIP) establece que los módulos de alto nivel no deben depender de módulos de bajo nivel, sino de abstracciones. Esto significa que se deben utilizar interfaces y abstracciones en lugar de clases concretas, lo que facilita la evolución y el cambio del sistema.

En cuanto al principio Clean Code, promueve prácticas como la simplicidad, la claridad, la consistencia y la eliminación de duplicación. Esto significa que el código debe ser fácil de leer, entender y mantener, evitando complejidad innecesaria y redundancia. Al seguir estas prácticas, se mejora la calidad del código y se reduce el tiempo y costo de mantenimiento del sistema.

## 2.7. Modelos de Base de Datos




## 2.8. Conclusiones

Al llegar a esta sección se ha dado un recorrido por todas las bases y herramientas de la aplicación. Demostrando la calidad con la que se realiza el proceso. Este software garantiza cumplir las necesidades de los clientes, y las características incluidas en el producto, como la generación de PDF, la capacidad de mostrar a cualquier trabajador la información de su Local entre otras hace que sea un poco más profunda la utilidad de este.

## Capítulo 3: Valoración de la viabilidad de la propuesta de solución

En el presente capítulo se realizan la validación de requerimientos, las métricas aplicadas a los requisitos, pruebas de software y pruebas de aceptación.

### 3.1 Métricas aplicadas a los requisitos

Con el objetivo de medir la calidad de la especificación de los requisitos se aplicó una de las métricas Calidad de la especificación (CE). Para obtener cuán entendibles y precisos son los requisitos, primeramente, se calcula el total de requisitos de la especificación como se muestra a continuación:

**Nr**: El total de requisitos de especificación.

**Nf**: Cantidad de requisitos funcionales.

**Nnf**: Cantidad de requisitos no funcionales.

Como resultado de la sustitución de los valores, para el sistema se obtiene:

**Nr** = **Nf** + **Nnf**

**Nr** = 40 + 7

**Nr** = 47

Para determinar, finalmente, la Especificidad de los Requisitos (ER) o ausencia de ambigüedad en los mismos se realiza la siguiente operación:

**ER** = **Nui** / **Nr**

Donde es el número de requisitos para los cuales todos los revisores tuvieron interpretaciones idénticas. Mientras más cerca de 1 esté el valor de ER, menor será la ambigüedad.

Para el caso de los requisitos obtenidos para el portal algunos produjeron contradicción en las interpretaciones. Sustituyendo las variables se obtiene:

**ER** = 41 / 47

**ER** = 0.87

Arrojando un resultado final satisfactorio, indicando que el grado de ambigüedad de los requisitos es bajo (13%) ya que el 87% es entendible. Los requisitos ambiguos fueron modificados y validados para garantizar su correcta comprensión.

### 3.2 Prueba de Caja Negra

Las pruebas de caja negra son las que se llevan a cabo en la interfaz del software. **Se definen como una técnica de análisis de la funcionalidad de un sistema** que no tiene en cuenta la estructura interna del código.[44], [45]

**Caso de Prueba 1:**

Iniciar Sesión: Es una vista que valida que los datos “Correo electrónico” y “Contraseña” sean válidos y no estén vacíos para permitir el acceso al sistema de forma correcta.


Validaciones:

La validación ocurre en 2 casos, el primero si se da “click” a iniciar sesión directamente, y resalta los errores, y el segundo caso si se entra a un campo y se sale sin introducir ningún dato, saliendo el error en ese campo que dejó de agregar.

Existen 4 validaciones:

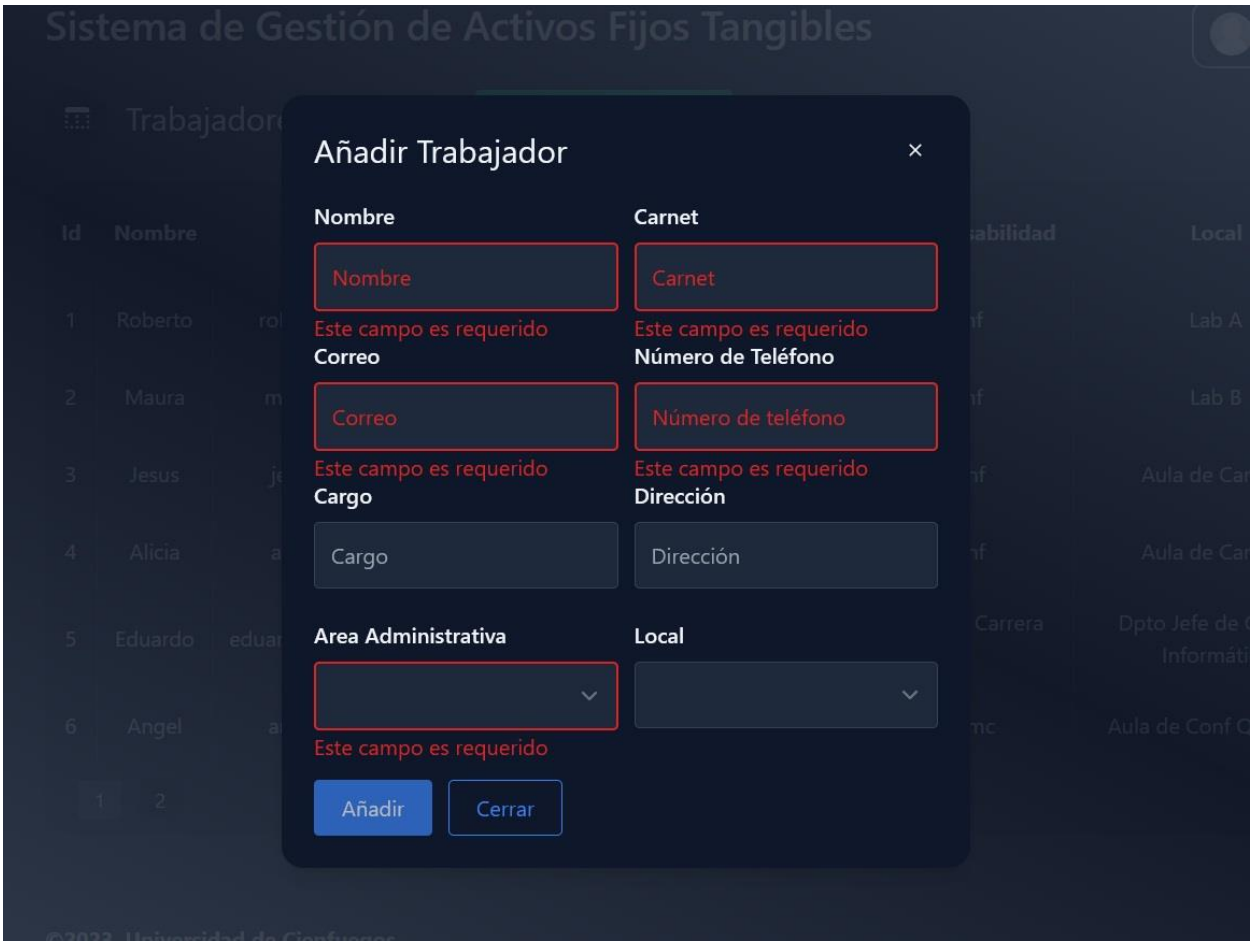
- Si es un correo válido o no.
- Para ambos campos una validación de que no estén vacíos.
- Que la contraseña, así como todas las que hay en el sistema tengan más de 5 caracteres.
- Que el correo y la contraseña existan y estén asociados para permitir la entrada al sistema.

Zona de Errores:

Estará ubicada en el campo de texto donde debe introducir sus datos y debajo de este.

**Caso de Prueba 2:**

Añadir Trabajador: Es una vista que valida que los datos introducidos del trabajador que se desea añadir sean válidos y no estén vacíos para eliminar posibles errores en el sistema.




Validaciones:

La validación ocurre en 2 casos, el primero si se da “click” a Añadir directamente, y resalta los errores, y el segundo caso si se entra a un campo y se sale sin introducir ningún dato, saliendo el error en ese campo que dejó de agregar.

Existen 5 validaciones:

- Todos los campos son requeridos excepto “Dirección” y “Cargo”.
- El carnet debe tener 11 caracteres y el campo está validado para que solo admita números positivos.

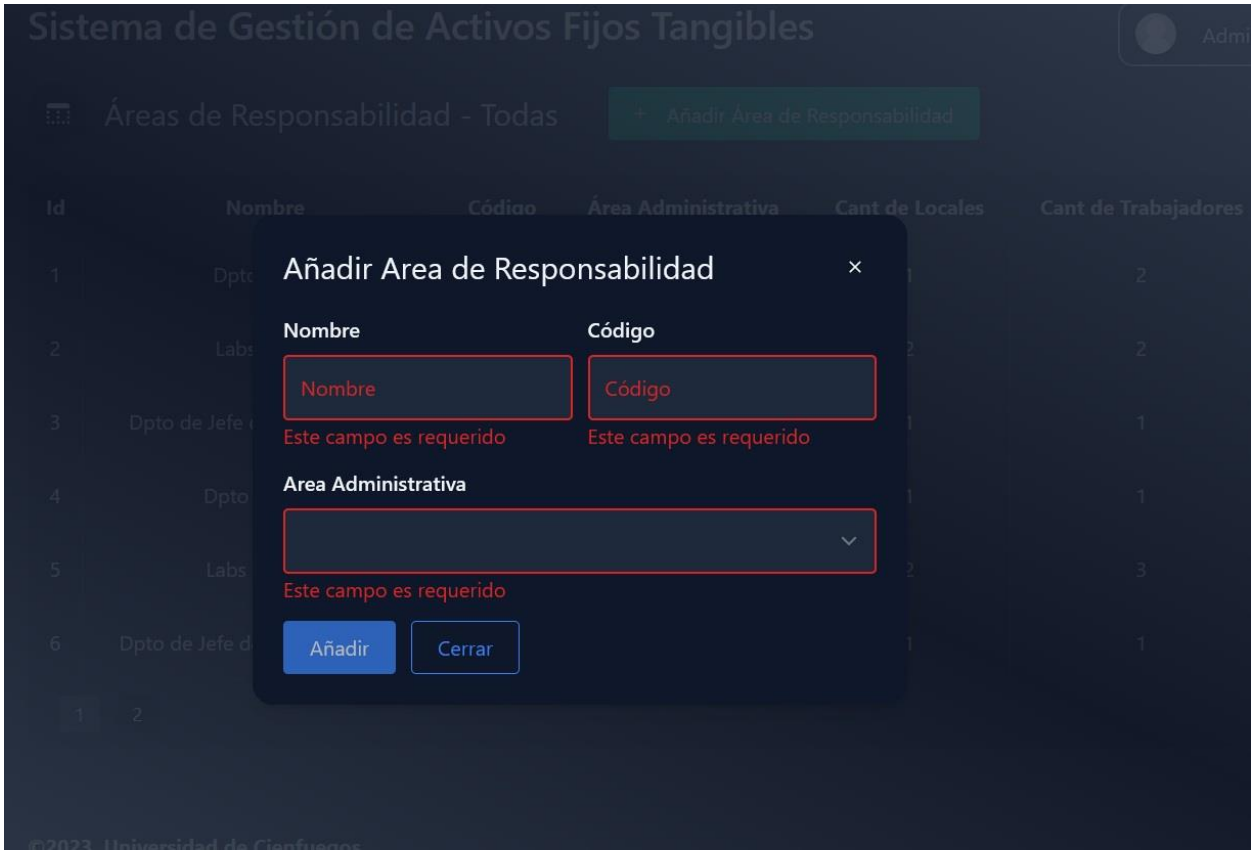
- El campo de correo debe ser un correo válido.
- El campo número de teléfono debe tener 8 caracteres y está validado para que solo admita números positivos.
- El campo Local solo se puede elegir cuando se haya elegido un Área Administrativa en caso de que lo esté creando un usuario con rol de “Administrador”, si es un usuario con rol de “jefe de Área” entonces solo tendrá que elegir el Local.

Zona de Errores:

Estará ubicada en el campo donde debe introducir sus datos y debajo de este.

**Caso de Prueba 3:**

Añadir Área de Responsabilidad: Es una vista que valida que los datos introducidos de la nueva Área de Responsabilidad que se desea añadir sean válidos y no estén vacíos para eliminar posibles errores en el sistema.




Validaciones:

La validación ocurre en 2 casos, el primero si se da “click” a Añadir directamente, y resalta los errores, y el segundo caso si se entra a un campo y se sale sin introducir ningún dato, saliendo el error en ese campo que dejó de agregar.

Existen 5 validaciones:

- Todos los campos son requeridos.
- El campo código está validado para que solo admita números positivos.
- El campo Área Administrativa solo se puede elegir en caso de que lo esté creando un usuario con rol de “Administrador”, si es un usuario con rol de “jefe de Área” entonces el Área se creara automáticamente en el Área Administrativa que este usuario maneje.

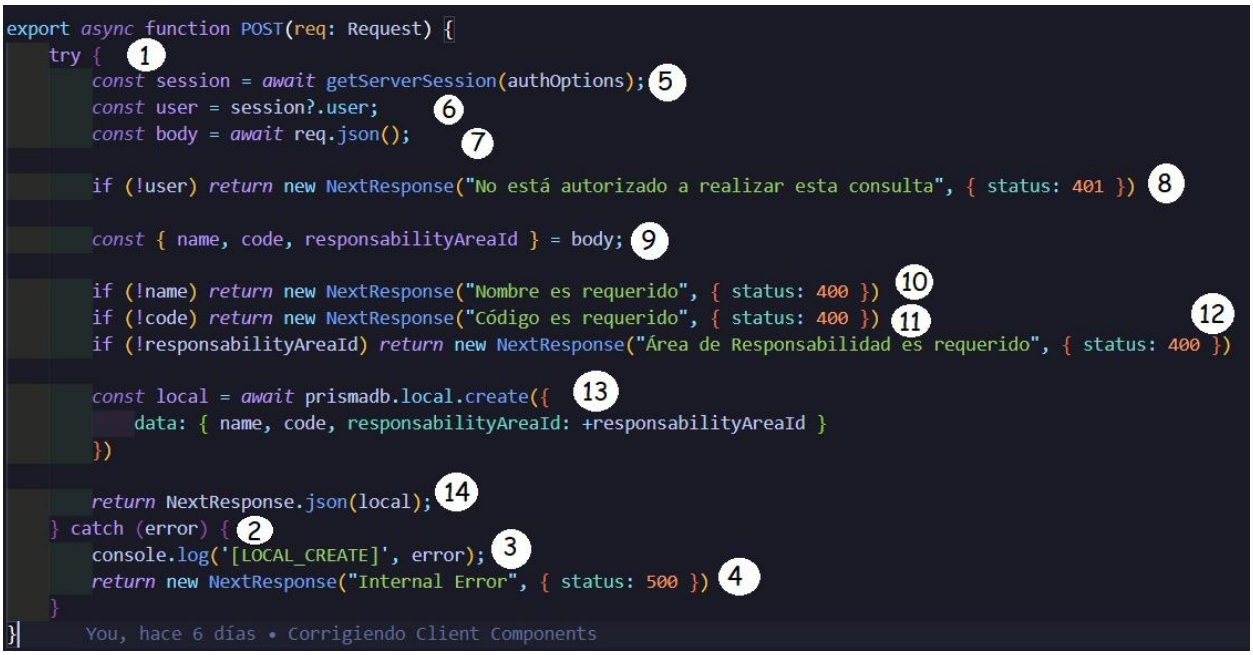
Zona de Errores:

Estará ubicada en el campo donde debe introducir sus datos y debajo de este.

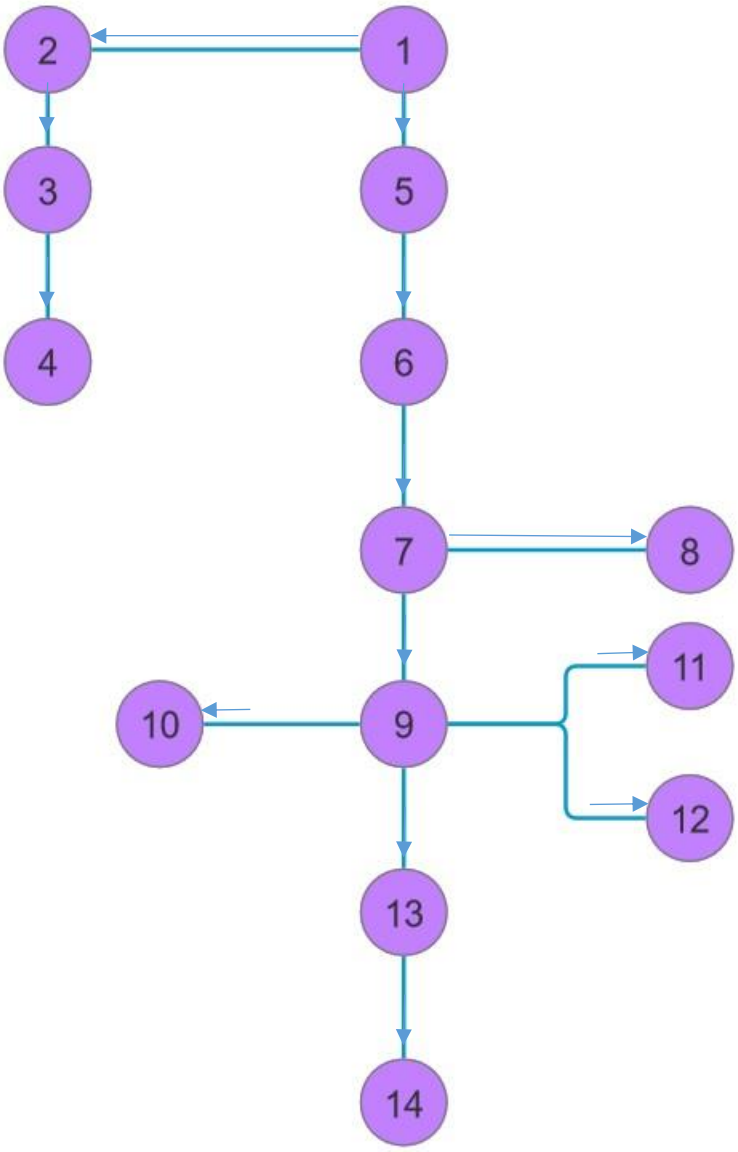
### 3.3 Prueba de Caja Blanca

Las pruebas de caja blanca del software se basan en el examen cercano de los detalles de procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones o bucles.[46], [47]

**Caso de Prueba 1:** Crear Local



Grafo de Caja Blanca 1: Crear Local



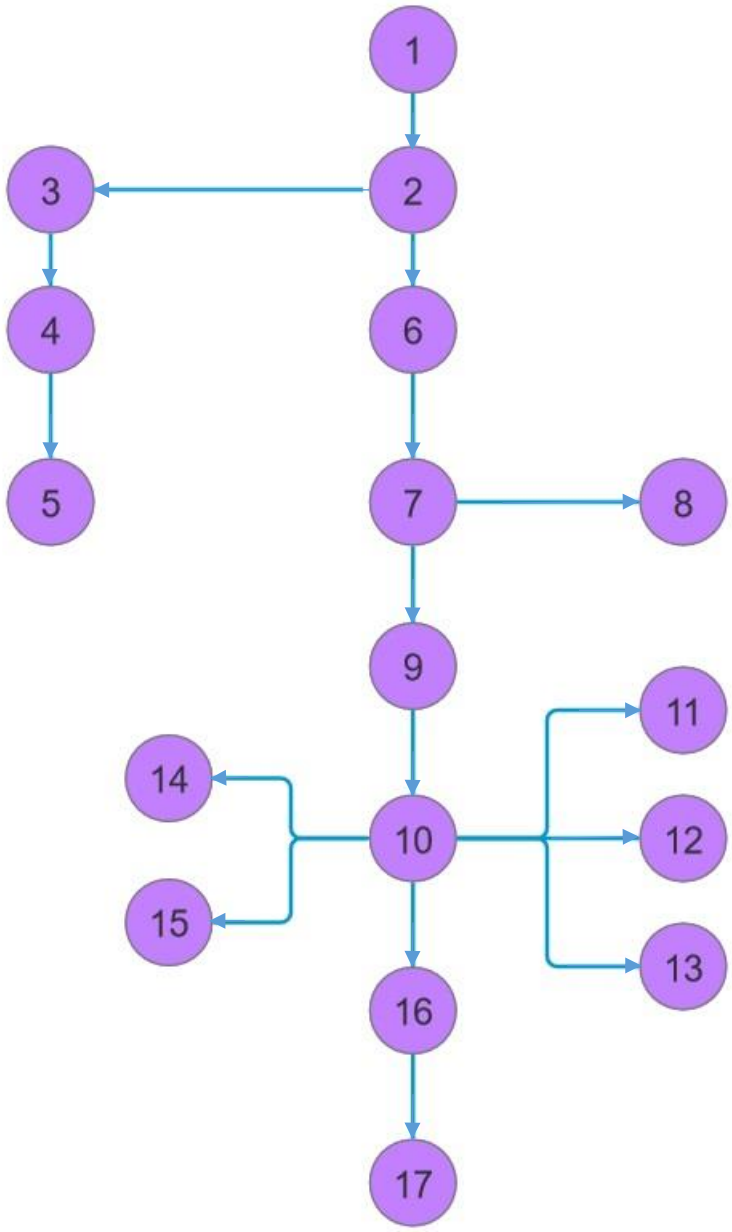

Luego se determina la complejidad ciclomática V (G) del grafo resultante, la cual es un indicador del número de caminos independientes que existen en un grafo, es decir, es cualquier camino dentro del código que introduce por lo menos un nuevo conjunto de sentencias de proceso o una nueva condición. La complejidad ciclomática puede ser calculada de la siguiente forma: V (G) = c + 1, siendo c el número de nodos de condición Realizando los cálculos correspondientes se obtiene el siguiente resultado:

V(G)= c + 1

V(G)= 4

**Caso de Prueba 2:** Modificar Trabajador


Grafo de Caja Blanca 2: Modificar Trabajador




Realizando los cálculos correspondientes se obtiene el siguiente resultado:

V(G)= c + 1

V(G)= 4

### 3.4 Pruebas de Aceptación

Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado.

Prueba de Aceptación: Historia de Usuario “Iniciar Sesión”

| | Correo | Contraseña | Aceptación |
|---|---|---|---|
| 1 | admin@ucf.edu.cu (V) | Admin987 (V) | V |
| 2 | adm@ucf.edu.cu (F) | adm (F) | F |
| 3 | robert@ucf.edu.cu (V) | Roberto123 (V) | V |
| 4 | roberto@ (F) | rob (F) | F |

### 3.5 Conclusiones

Al llegar a esta sección se pudieron ver algunas de las pruebas funcionales para verificar el correcto funcionamiento del sistema implementado.

Las pruebas de caja blanca fueron centradas en lógica interna y condiciones del código, mientras que las de caja negra se encargaron de validar el comportamiento externo del sistema.

Se comprobó que el sistema rechaza entradas inválidas de manera consistente y aceptaron entradas válidas, lo que asegura la integridad de los datos del sistema.

## Conclusiones Generales

Como resultado de la presente investigación se puede concluir que con la finalización del sistema desarrollado se puede satisfacer las necesidades de los jefes de Áreas, así poder brindar la información requerida.

Por lo tanto, se puede afirmar lo siguiente:

✓ Se describió el dominio del problema que es necesario conocer para entender el negocio del cliente, además se describió el flujo actual de los procesos.

✓ Se diseñó un sistema informático que responde a las necesidades planteadas soportando el flujo actual de los procesos.

✓ Se implementó un sistema informático que responde a las necesidades planteadas soportando el flujo actual de los procesos.

✓ Se validó el sistema realizando pruebas de caja blanca y caja negra demostrando la optimización lograda en cuanto a los errores.

**Recomendaciones**

Para dar continuidad a esta investigación se recomienda:

- Implementar una funcionalidad que permita generar un PDF con información de las diferentes áreas, así como informes de datos.
- Investigar sobre la existencia de alguna API que permita usar los datos directamente del software ASSETS, o implementarla.
- Actualizar a una versión superior de NextJS para utilizar ServerActions y mejorar el funcionamiento de la pagina y la experiencia de usuario.

## Referencias


[1] L. V. de la Cruz y F. M. Delgado, «Evolución del control interno hacia una gestión integrada al control de gestión», *Estud. Gest. Rev. Int. Adm.*, n.º 10, Art. n.º 10, jul. 2021, doi: 10.32719/25506641.2021.10.10.

[2] «What is ERP? The Essential Guide | SAP». Accedido: 6 de diciembre de 2023. [En línea]. Disponible en: https://www.sap.com/products/erp/what-is-erp.html

[3] «Assets - EcuRed». Accedido: 11 de junio de 2023. [En línea]. Disponible en: https://www.ecured.cu/Assets

[4] *Libro Control Interno*.

[5] «Gaceta Oficial de la República no. 13 2011 Resolución No. 60/11».

[6] «Decreto Ley No. 249/07 Responsabilidad Material».

[7] «Manual de Procedimiento para el Control Interno de Activos Fijos Tangibles».

[8] «Enterprise Resource Planning (ERP): Meaning, Components, and Examples», Investopedia. Accedido: 12 de mayo de 2023. [En línea]. Disponible en: https://www.investopedia.com/terms/e/erp.asp

[9] «Audita». Accedido: 10 de mayo de 2023. [En línea]. Disponible en: https://audita.cimex.com.cu/

[10] T. R. Ramirez y B. H. González, «Sistema para la Informatización del proceso de Auditoría y Control de los Activos Fijos Tangibles en la Facultad de Ciencias y Tecnologías Computacionales», vol. 9, n.º 11, 2016.

[11] «La empresa inteligente», SAP. Accedido: 14 de septiembre de 2023. [En línea]. Disponible en: https://www.sap.com/latinamerica/intelligent-enterprise.html

[12] «Open Source ERP and CRM | Odoo», Odoo S.A. Accedido: 29 de octubre de 2023. [En línea]. Disponible en: https://www.odoo.com/es_ES

[13] X. F. Grau y M. I. S. Segura, «Desarrollo Orientado a Objetos con UML».

[14] «Qué es el lenguaje unificado de modelado (UML)», Lucidchart. Accedido: 12 de octubre de 2023. [En línea]. Disponible en: https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml

[15] «Metodologías Agiles».

[16] A. Gandarillas, «AUP», Metodología. Accedido: 10 de abril de 2023. [En línea]. Disponible en: https://metodologia.es/aup/

[17] «Metodología Aup».

[18] U. de Holguín y O. L. Moya, «Tesis presentada en opción al Título de Licenciado en Contabilidad y Finanzas.».

[19] Y. D. González y Y. F. Romero, «Patrón Modelo-Vista-Controlador.», *Telemática*, vol. 11, n.º 1, Art. n.º 1, jun. 2012.

[20] «Documentation for Visual Studio Code». Accedido: 19 de julio de 2023. [En línea]. Disponible en: https://code.visualstudio.com/docs

[21] «Overview - TablePlus Documentation». Accedido: 19 de julio de 2023. [En línea]. Disponible en: https://docs.tableplus.com/

[22] «Axure RP - UX Prototypes, Specifications, and Diagrams in One Tool». Accedido: 19 de abril de 2023. [En línea]. Disponible en: https://www.axure.com/

[23] «Thunder Client - Rest API Client Extension for VS Code». Accedido: 12 de agosto de 2023. [En línea]. Disponible en: https://www.thunderclient.com/

[24] «Ideal Modeling & Diagramming Tool for Agile Team Collaboration». Accedido: 12 de junio de 2023. [En línea]. Disponible en: https://www.visual-paradigm.com/

[25] «Estructurando la web con HTML - Aprende desarrollo web | MDN». Accedido: 18 de mayo de 2023. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Learn/HTML

[26] «CSS - Aprende desarrollo web | MDN». Accedido: 12 de abril de 2023. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Learn/CSS

[27] «JavaScript - Aprende desarrollo web | MDN». Accedido: 12 de julio de 2023. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Learn/JavaScript

[28] «TypeScript: The starting point for learning TypeScript». Accedido: 19 de octubre de 2023. [En línea]. Disponible en: https://www.typescriptlang.org/docs/

[29] «Prisma Documentation | Concepts, Guides, and Reference», Prisma. Accedido: 9 de septiembre de 2023. [En línea]. Disponible en: https://www.prisma.io/docs

[30] «Inicio rápido – React». Accedido: 23 de junio de 2023. [En línea]. Disponible en: https://es.react.dev/learn

[31] «Installation - Tailwind CSS». Accedido: 10 de julio de 2023. [En línea]. Disponible en: https://tailwindcss.com/docs/installation

[32] «Docs | Next.js». Accedido: 19 de mayo de 2023. [En línea]. Disponible en: https://nextjs.org/docs

[33] «PostgreSQL: Documentation». Accedido: 19 de septiembre de 2023. [En línea]. Disponible en: https://www.postgresql.org/docs/

[34] «Requerimientos funcionales: Ejemplos - La Oficina de Proyectos de Informática». Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html

[35] «Qué son los Requisitos No Funcionales: Ejemplos, Definición, Guía Completa - Visure Solutions». Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://visuresolutions.com/es/blog/non-functional-requirements/

[36] «Historias de Usuario, Escritura, Definición, Contexto y Ejemplos», SCRUM MÉXICO. Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://scrum.mx/informate/historias-de-usuario

[37] «MVC (Modelo-Vista-Controlador): ¿qué es y para qué sirve?» Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve

[38] M. Coppola, «Qué es una API REST, para qué sirve y ejemplos». Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://blog.hubspot.es/website/que-es-api-rest

[39] V. Gagliardi, *Decoupled Django: Understand and Build Decoupled Django Architectures for JavaScript Front-ends*. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-7144-5.

[40] «Render Props – React». Accedido: 9 de septiembre de 2023. [En línea]. Disponible en: https://es.legacy.reactjs.org/docs/render-props.html

[41] «Higher-Order Components – React». Accedido: 19 de octubre de 2023. [En línea]. Disponible en: https://legacy.reactjs.org/docs/higher-order-components.html

[42] «Server Side Rendering I - Conceptos», Lemoncode formacion. Accedido: 9 de octubre de 2023. [En línea]. Disponible en: https://lemoncode.net/lemoncode-blog/2018/5/13/server-side-rendering-i-conceptos

[43] «Clean code y solid : arquitectura mantenible | by Javier Aparisi Valdés | Medium». Accedido: 19 de abril de 2023. [En línea]. Disponible en: https://japarisivaldes.medium.com/clean-code-solid-y-test-arquitectura-mantenible-462f5efd5b23

[44] K. Team, «¿Qué son las pruebas de caja negra? | KeepCoding Bootcamps». Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://keepcoding.io/blog/que-son-las-pruebas-de-caja-negra/

[45] «Pruebas de caja negra: proceso, herramientas, lista de comprobación y mucho más». Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://www.zaptest.com/es/pruebas-de-caja-negra-que-son-tipos-procesos-enfoques-herramientas-y-mucho-mas

[46] C. Singureanu, «Pruebas de caja blanca: tipos, proceso, herramientas y mucho más.», https://www.zaptest.com/es. Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://www.zaptest.com/es/pruebas-de-caja-blanca-que-es-como-funciona-retos-metricas-herramientas-y-mas

[47] K. Team, «¿Qué son las pruebas de caja blanca?» Accedido: 11 de diciembre de 2023. [En línea]. Disponible en: https://keepcoding.io/blog/que-son-pruebas-de-caja-blanca/