Sistema de Control de Inventario y Activos Fijos: una entrada al futuro digital para el Colegio San Francisco de Asís

Andrés Felipe Estupiñán Peláez

Asesor

Andrés Felipe Hincapié Saldarriaga

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Ingeniería de sistemas

2024

Dedicatoria

Este documento y proyecto lo dedico a mi familia, a Yolanda Peláez Betancourt, mi madre, por servir de fuente de inspiración y motivación para completar mi titulación profesional, por todo su apoyo y ser mi pilar de fuerza para seguir adelante. Lo dedico a mi abuela Blanca Gilma Betancourt, y mi padre José Felipe Estupiñán, que en paz descansen, quienes me brindaron todo su apoyo en mi proceso de formación desde que era pequeño hasta la adultez; a Dios, quién me ha guiado hasta este momento y me ha brindado sus fuerzas para salir adelante. También, un total agradecimiento a todos los familiares y conocidos quienes me han apoyado con su granito de arena para mi crecimiento personal y profesional.

Este proyecto representa la culminación de mi adquisición de conocimientos necesarios para mi trayectoria profesional en la Universidad Nacional Abierta y a Distancia, así como un nuevo capítulo en mi vida y trayectoria profesional.

Agradecimientos

Deseo brindar un enorme agradecimiento al Colegio San Francisco de Asís, institución en la que estudié desde la primaria hasta el bachiller técnico, por permitirme realizar este proyecto, al docente Juan Carlos Jaramillo por instruirme durante todo el proceso, y educarme en el ámbito de los sistemas de información durante la primaria y grados 10° y 11°; al coordinador académico José Heraldo Criollo, quien fue mi docente de matemáticas durante grados 6° y 7° y me brindó varios consejos al finalizar el bachillerato.

De igual forma, un total agradecimiento a la academia de la Universidad Nacional Abierta y a Distancia por brindarme la oportunidad de formarme como profesional de Ingeniería de Sistemas, así como al tutor Andrés Felipe Hincapié Saldarriaga por su acompañamiento en el desarrollo del proceso del proyecto de grado.

Resumen

Las organizaciones a nivel mundial, sin importar su tamaño y tipo, poseen métodos de control de inventario que permiten realizar un seguimiento a los activos y productos adquiridos por parte de la entidad. En la actualidad, las entidades con procesos de inventario con alta madurez poseen un software de gestión de inventario que permite controlar de manera eficiente todos los procesos relacionados al movimiento de activos fijos, adquisición de productos y el desarrollo de las cotizaciones para adquirir nuevas existencias; todos estos procesos pueden ser objeto de auditoría, por lo que, al llevar un control de manera física o en un documento de Word, está a objeto de error humano. El Colegio San Francisco de Asís, ubicado en la ciudad de Santiago de Cali, Valle del Cauca, presenta necesidades relacionadas a la optimización y sistematización del proceso de gestión de los activos fijos de la institución educativa.

El presente documento detalla el desarrollo de un software que responde a las necesidades específicas de la institución, mediante una plataforma web de alta calidad que es eficaz y eficiente, amigable con el usuario y con un alto nivel de rendimiento y disponibilidad.

***Palabras clave*:** Laravel, Svelte.js, activos fijos, administración, software.

Abstract

Organizations on a worldly scale, no matter their size or type, have methods to control their inventory and track the products and assets they have acquired. Currently, organizations that have highly mature inventory administration processes have an inventory management software that allows them to control efficiently all processes related to fixed assets, product acquisition and estimate developments to acquire new inventory; all these processes are usually subject to auditing so controlling them through physical methods or through a Word document are prone to human error. The San Francisco de Asís School, located in the city of Santiago de Cali, Valle del Cauca, have needs related to optimizing and systematizing the process of managing the fixed assets of the educational institution.

The following document details the development of software program that responds to the specific needs of the organization, through a high-quality web platform that is effective and efficient, user friendly and with a high level of performance and availability.

***Keywords*:** Laravel, Svelte.js, fixed assets, management, software.

Tabla de contenido

Lista de Tablas

Lista de Figuras

Introducción

La gestión correcta del inventario en las organizaciones es de alta importancia, ya que permite llevar un control sobre los activos y productos existentes, de tal manera que las empresas puedan conocer los costos operativos sobre el inventario existente. De acuerdo con Camacho, Ríos, Mojica, Rojas (diciembre, 2020), el inventario de una compañía puede representar hasta el 50% de sus activos, por lo que un mal manejo de este apartado incrementa los costos de manera significativa, dado a que no es planificada la compra o baja de activos fijos en este tipo de entidades, lo cual también puede afectar indirectamente en la calidad de los servicios brindados a los clientes.

Dado a la necesidad de generar control sobre los costos e insumos adquiridos por parte de las organizaciones, las empresas desarrollan sus propios sistemas de inventario de forma interna, y/o adquieren productos de software estándar que permitan solucionar estas problemáticas. En este sentido, existen una gran discrepancia en la calidad del manejo de inventario entre organizaciones de diferentes tamaños y/o carácter. En este caso, se realizó un análisis sobre el caso del Colegio San Francisco de Asís, institución educativa ubicada en la ciudad de Santiago de Cali, Valle del Cauca. Al revisar sus procesos de control de activos fijos, se observó que se tiene un control físico junto a un control llevado en una hoja de cálculo de Microsoft Excel. Si bien esto puede resultar óptimo para empresas pequeñas, la institución educativa cuenta con una trayectoria de más de 60 años al servicio de la comunidad, y cuenta con una edificación bastante amplia, por lo que el control de inventario no es totalmente eficiente.

Se desarrolló un sistema de información eficiente, amigable con el usuario y diseñado a la medida para el Colegio San Francisco de Asís, con el fin de optimizar sus procesos de gestión de inventario, y mejorar el seguimiento a los procesos de movimiento de activos, cotizaciones y facturas. Este sistema de información se genera mediante una plataforma web, lo cual garantiza una alta disponibilidad, usando tecnologías modernas para su desarrollo y subsecuente puesta en marcha en producción, de tal manera que sea configurable por los administradores de sistemas de la institución y el auxiliar de almacén.

Problema de investigación

Según Rodríguez, J. (agosto, 2023), el control de inventarios permite tener un balance en las existencias de un almacén y controlar los productos que tienen mayor demanda al momento de realizar su comercialización. Subsecuentemente, el seguimiento correcto del inventario existente puede reducir los costos a nivel organizacional ya que se controla la cantidad de productos, insumos o bienes que deban ser adquiridos, o los activos a los cuales deba de realizarse mantenimiento, sin errores en la medida de lo posible. Estando en la era de la digitalización, donde la mayoría de los sistemas de información físicos existentes son trasladados a un sistema digital, es de vital importancia disponer de una aplicación que sea fácil de usar, eficiente y amigable con el usuario en un proceso que es tan vital para las organizaciones.

Al realizar una visita al Colegio San Francisco de Asís, ubicado en el barrio de Villacolombia de la ciudad de Santiago de Cali, Valle del Cauca, se observa que tienen procesos de control de inventario y de cotizaciones, donde se documenta la información mediante Microsoft Excel y Microsoft Word. Estas herramientas, si bien son bastante útiles para las organizaciones, pueden convertirse en un problema ya que no son totalmente escalables al momento en que se almacene más información. De acuerdo con Marín, L. (2021), la integridad de los datos puede verse comprometida, al estar sujeta a error humano y, al aumentar la cantidad de datos registrada en una hoja de cálculo, se va reduciendo el rendimiento de forma gradual, de tal manera que usar Excel puede volverse una experiencia tediosa para el usuario.

Formulación de la pregunta

A partir de la problemática encontrada anteriormente, formulamos nuestra pregunta problematizadora, la cual es fundamento para el proyecto desarrollado:

¿Es posible brindar un sistema de información robusto y de alta calidad que permita al Colegio San Francisco de Asís de Cali administrar el inventario y las cotizaciones de productos de manera eficiente y rápida haciendo uso de tecnologías web como Laravel y Svelte.js?

Objetivos

Objetivo General

Desarrollar un sistema de control de activos y gestión de inventarios de alta calidad y eficiencia que permita administrar los bienes del Colegio San Francisco de Asís, así como controlar el movimiento de activos y cotizaciones.

Objetivos Específicos

Observar los procesos actuales de inventario y cotización del Colegio San Francisco de Asís

Realizar la planificación y documentación preliminar del proyecto teniendo en cuenta las necesidades puntuales del colegio.

Desarrollar una aplicación a nivel de backend en Laravel y frontend con Svelte.js que permita administrar toda la sección de activos de la institución educativa.

Generar pruebas automatizadas que cumplan con los casos de uso planteados.

Implementar y probar el aplicativo en las instalaciones del Colegio San Francisco de Asís.

Justificación

De acuerdo con Prieto, E. (2023), la complejidad de un sistema de control de inventarios puede variar entre las diferentes empresas teniendo en cuenta su escala, sin embargo, un elemento en común es que el buen control de entradas y salidas de almacén permite dar un servicio adecuado a los clientes y reduce los costos operativos de las organizaciones. Un sistema maduro tiene control sobre la calidad de los artículos, cantidad de productos en almacén, conocer su ubicación, entre muchas otras propiedades y/o atributos.

De acuerdo a lo anterior, es de vital importancia para las organizaciones madurar sus sistemas de inventario, motivo por el cual, mediante este proyecto de digitalización del control de inventario, se desea desarrollar una plataforma web que reemplace el control realizado en Microsoft Excel y papel que se realiza en el Colegio San Francisco de Asís, con el fin de orientar a la institución a la transformación digital que se vive en el siglo XXI, facilitando el control que se realiza a los activos en circulación, los productos en almacén, y aumentar la velocidad de consulta de la información existente en el momento en que se desee generar reportes y/o realizar movimientos de activos, así como llevar un registro mucho más sencillo de las facturas registradas y las cotizaciones realizadas por parte de la institución, así como evitar pérdidas de información al momento de llevar el control por medio de papel, o perder el control de la secuencia de las cotizaciones y/o placas de activo, garantizando la integridad de la información y su seguridad, al igual que un rendimiento eficiente mediante un desarrollo de un aplicativo eficaz y amigable con el usuario.

El control adecuado es aún más importante, sabiendo que, de acuerdo al Ministerio de Educación Nacional (2020), las instituciones educativas deben de realizar una revisión y actualización del inventario mínimamente cada tres años como recomendación general, por lo que sistematizar este proceso facilita el sondeo de activos en la institución.

Con base a las necesidades particulares del Colegio San Francisco de Asís, nace el proyecto “Sistema de Inventario y Movimiento y Control de Activos Institucional” (SIMCAI), basado en una plataforma web usando los frameworks Laravel y Svelte.js.

Marco de referencia

Marco contextual

Se realizó una visita al Colegio San Francisco de Asís el día 3 de marzo del 2023, con el fin de analizar las condiciones de la gestión de inventario y los procesos existentes en la institución educativa.

Profundizando un poco sobre la institución educativa, el Colegio San Francisco de Asís es un colegio de carácter privado que se encuentra ubicado sobre la Calle 52 #14-40 en el barrio Villacolombia. El colegio se fundó en el año 1960 por el Fray Agustin Acero Pedraza. Hasta el año 2011, el colegio se ubicaba en la parte trasera de la parroquia San Pio X, ubicada en el mismo barrio. El traslado de sede significó un gran cambio para la institución, ya que aumentó la cantidad de espacio disponible con el que el colegio contaba, lo cual contribuyó en un aumento en el volumen de activos y productos de inventario de los cuales había que llevar un control.

El Colegio San Francisco de Asís, de acuerdo a sus procesos, implementa una actualización y revisión del inventario cada año y, dado el tamaño de la planta institucional actual, es una labor de alta complejidad. El colegio cuenta con 3 pisos y un nivel de sotano, en los 3 pisos, se distribuyen más de 30 salones entre los que se incluyen 4 laboratorios y 4 salas de sistemas.

Figura 1.
Vista satelital del Colegio San Francisco de Asís.

Fuente: Google Maps.

El proceso existente relacionado al control de activos, inventariado y cotizaciones se maneja de manera física y es bastante simple, por lo que está sujeto a posibles errores humanos al registrar la información de movimiento de activos, conteo de productos de almacén, pérdida de documentación, difícil acceso a la documentación histórica, factores ambientales y biológicos, entre otros problemas relacionados a la documentación física de este tipo de procedimientos, sumado también a la complejidad relacionada al llevar un control de los productos y activos en la gran cantidad de salones de clase, laboratorios, entre otros. Adicionalmente, llevar un control

físico no contribuye a la reducción del uso del papel, lo que evita la conservación del medio ambiente, que es vital en el siglo XXI.

Marco documental

Teniendo en cuenta lo anterior, se realizó una revisión bibliográfica acerca de proyectos de esta índole que han sido desarrollados anteriormente, con el fin de observar la metodología usada para la creación del software, teniendo en cuenta los conocimientos adquiridos por parte de los autores y limitaciones técnicas y/o detalles presentados en el desarrollo de dichos proyectos. El resultado de la anterior labor de recolección de información es la siguiente:

Se revisaron dos proyectos en el repositorio institucional de la Universidad Nacional Abierta y a Distancia relacionados a la gestión de inventarios, el sistema presentado por Lora, W. Ahumada, D. Muñoz, J, de “Implementación de un sistema de inventario para bodegas a través de un aplicativo móvil nativo en Android”, el cual implementa un sistema general, orientado a empresas con múltiples sedes. Si bien la ejecución de un proyecto a nivel general puede ser óptima en casos de uso generales, se pierde el enfoque en necesidades particulares que tengan las empresas, ya que no todas tienen un procedimiento estándar y pueden diferir entre sí en sus procesos de inventario. La mayor limitación de esta revisión es la plataforma, al encontrarse solamente disponible en Android, se limita el acceso a usuarios de las plataformas iOS y computadores de escritorio en labores de inventario, y no contiene una implementación de un sistema de movimiento de activos, el cual busca implementarse en el colegio. Su mayor fortaleza es la implementación de un sistema de reconocimiento de código de barras, agilizando el sondeo

y toma de información de inventarios. El aplicativo fue validado por la administración del establecimiento comercial “Tienda Rengifo”, la cual realizó una serie de recomendaciones generales al proyecto para futuras mejoras.

Se hizo una revisión adicionalmente al sistema desarrollado por Lotero, J. en su proyecto “Sistema de Inventario Web para el Seguimiento de Recursos Físicos de la Secretaría de Movilidad de Envigado”, el cual utiliza una metodología de desarrollo en la que se elaboran prototipos como casos de prueba, en la que el usuario indica su satisfacción con el proyecto realizado. En este caso, usa una plataforma web, también desarrollada en el lenguaje de programación PHP, sin embargo, no usa ningún framework de backend asociado al lenguaje, lo cual incurre en desarrollar funcionalidades necesarias que ya se encuentran solucionadas al usar frameworks como Symfony, Laravel, CodeIgniter, entre otros. El proyecto propone una solución alternativa interesante a las placas tradicionales con código de barras, y es usar un código QR que contenga la información que describa a los activos existentes, con la posibilidad de imprimirlo directamente desde la aplicación. Adicionalmente, cuenta con un sistema de movimiento de activo y asignación de inventario a los usuarios existentes en el software. El proyecto fue exitoso en su implementación, sin embargo, dada a la naturaleza de la metodología seleccionada, el producto final sigue siendo un prototipo bastante funcional, el cual el autor señala que puede mejorarse aún más realizando mejoras.

Externo a la Universidad Nacional Abierta y a Distancia, se consultó el proyecto desarrollado por Angulo Corzo, D y Nicho Príncipe, N. en “Implementación de un sistema web para la gestión de ventas e inventario de una empresa de calzado”, de la Universidad San Ignacio

de Loyola, el cual fue desarrollado usando el lenguaje de programación PHP usando el framework CodeIgniter y el motor de base de datos MySQL. El proyecto utiliza la metodología RUP para su desarrollo, y se implementó en la empresa Marlene Calzados. Los autores detallan extensivamente el proceso de ventas y de inventario de la empresa, dando un énfasis en la aceleración del proceso de registro de ventas, a comparación de cuanto tomaba anteriormente este proceso en la empresa. El apartado de inventario detalla adicionalmente la interacción con el área de producción y de solicitud de insumos, dada la naturaleza de la organización en que se implementa el software. El resultado de este proyecto fue la reducción de los tiempos en la gestión de los procesos de venta e inventario, validado por medio de encuestas de satisfacción con los usuarios existentes en la organización, y una integración bastante fuerte entre los módulos de venta e inventario, para llevar un control adecuado del stock existente, adicionalmente con un sistema de reportes enfocada en las ventas existentes y en el control del inventario actual.

Marco teórico

A medida que las organizaciones crecen en tamaño, es natural que adopten procesos y/o sistemas que permitan llevar el control de los datos que se generan o que son necesarios para el funcionamiento correcto de la empresa. El control de inventario es un proceso que, como hemos definido anteriormente, no puede ser omitido, ya que es un elemento crucial para gestionar los artículos que compra la organización que permiten su funcionamiento administrativo o que son comercializados a terceros, dado a su importancia, requiere adicionalmente que las entidades brinden la importancia debida, mediante la implementación de procesos de control y/o sistemas de información.

Los avances tecnológicos que se han obtenido desde los años 90, particularmente en la estandarización y adopción de las tecnologías relacionadas al internet permiten que los negocios puedan sistematizar muchos de sus procesos y hacer que tengan una alta disponibilidad no solo localmente, si no a nivel mundial, mediante la World Wide Web.

Dada la alta disponibilidad de los protocolos de internet y la necesidad de las empresas de consultar la información en cualquier momento y cualquier lugar sea garantizada, muchos proyectos de desarrollo de software son realizados en tecnologías web y, en este sentido, el sistema de información que será desarrollado en el transcurso del documento estará orientado a la web, usando el lenguaje de programación PHP y el framework de Laravel, se garantizará un desarrollo ágil y eficiente, con una sintaxis y estructura de código limpia que permita su

mejoramiento fácil y brindar posibilidades de extender el funcionamiento más allá de la entrega inicial del software al colegio.

El lenguaje de programación PHP es un lenguaje de scripting que posee una historia extensa de reescrituras y de reestructuración de código, debido a que el alcance inicial que tenía el proyecto no era necesariamente encaminado a realizar un lenguaje de programación. Inicialmente desarrollado por Rasmus Lerdorf en 1994, surgió como una serie de herramientas personales para controlar la cantidad de visitas a la página web del currículo de Lerdorf. Tuvo una serie de nombres como *PHP Tools, PHP/FI (Forms Interpreter), FI* y, actualmente, *PHP*. Inicialmente, la abreviación de PHP era *Personal Home Page*, dado a que se había envisionado como una serie de herramientas sencillas para páginas web que, de trasfondo, usaba ejecutables de *Common Gateway Interface* (CGI) de C. Actualmente, la abreviación PHP es *PHP Hypertext Protocol*. En 1995, PHP fue liberado como código abierto bajo la licencia de GNU GPL, con el fin de que se pudiesen agregar nuevas funcionalidades al lenguaje de scripting y proveer correcciones. Para el año 1997, se reportó que, aproximadamente, 60.000 dominios web reportaban cabezeras de petición exclusivas de PHP y, para el año 2004, millones de páginas web usaban PHP como lenguaje de programación en backend. Desde entonces, PHP ha madurado en gran medida, aumentando el rendimiento ofrecido y mejorando el lenguaje con el fin de ofrecer funcionalidades similares a las encontradas en otros lenguajes de programación como Java o C#. Su versión actual es la 8.2.

A lo largo de los años, han existido múltiples marcos de trabajo que reducen el código repetitivo para realizar funcionalidades en PHP, frameworks como Symfony, CodeIgniter o

CakePHP se han encargado de ofrecer una mejor experiencia de trabajo con el lenguaje de programación. Entre los marcos de trabajo existentes para este lenguaje de programación se encuentra Laravel. Este proyecto, de código abierto, es el framework más popular de PHP, conocido por ofrecer una sintaxis elegante, ser bastante extensible mediante librerías de Composer y encargarse de múltiples funcionalidades que pueden ser tediosas para los desarrolladores web. Laravel fue creado en 2011 por Taylor Otwell como una alternativa a CodeIgniter, con influencia del framework Ruby On Rails del lenguaje de scripting Ruby. Su popularidad creció de manera exponencial debido a su facilidad de uso y, actualmente, se encuentra en la versión 10.

Las aplicaciones web en la actualidad se benefician de los avances tecnológicos realizados en los estándares de HTML5, JavaScript y CSS, especialmente relacionados a la aparición de nuevos marcos de trabajo en el frontend que abordan la lógica y la estética de las aplicaciones. La aparición de nuevos frameworks como React.js, Angular y Vue.js han permitido que las aplicaciones web sean desarrolladas con la idea de Single Page Applications (aplicaciones de una sola página), con la idea acercar a las páginas web a la fluidez de las aplicaciones de escritorio. Cada marco de trabajo tiene sus diferentes ventajas y métodos para realizar aplicaciones de una sola página, manejar la reactividad de los componentes, entre otros. Svelte.js es un framework de JavaScript creado por Rich Harris de código abierto publicado en 2016, que ganó bastante popularidad en el año 2020, debido a que, a diferencia de otros frameworks de frontend como React.js o Vue.js, actúa más como un compilador que emite un código JavaScript óptimo. Esto se traduce en aplicaciones con un alto nivel de rendimiento, ya

que no se usa un DOM virtual como otros marcos de trabajo, y genera archivos de JS con tamaño reducido.

En conjunto, la parte del frontend y backend pueden ser considerarse microservicios que pueden ser ofrecidos por aplicaciones, y se comunican mediante llamados API que son atendidos por el protocolo HTTP, HTTPS o WebSockets en casos particulares. Los desarrolladores de software han encontrado técnicas que permiten distribuir estos microservicios mediante un solo paquete o archivo de configuración, y es allí donde nace el uso de contenedores de aplicación. La tecnología de contenedores es en cierta medida similar a la de una máquina virtual, en la que se distribuye una imagen de una aplicación con su respectiva configuración, lista para ser usada en ambientes de producción y aislada de la máquina física, ya que el código del aplicativo se ejecuta en aislamiento al sistema operativo anfitrión. En este caso, el uso de Docker en este proyecto permitirá otorgar al sistema de información una configuración lista para el despliegue, y un uso sencillo de microservicios, como lo pueden ser bases de datos, servidores web, bases de datos en memoria, entre otras funcionalidades que pueden ser configuradas y distribuidas fácilmente. Docker nace en el año 2013, desarrollado en el lenguaje de programación Go, permite aprovechar las tecnologías de virtualización presentes en los procesadores de AMD e Intel y las interfaces de virtualización en Windows, Linux y macOS. Docker ofrece también una interfaz gráfica mediante el uso de Docker Desktop, en los sistemas operativos Windows y macOS, esta es la forma de instalación principal, en Linux, es posible instalarlo sin la interfaz gráfica. Docker usa las interfaces de virtualización disponibles en cada sistema operativo, pero se beneficia ampliamente de la virtualización ofrecida por Linux, dado a que su implementación usa KVM, un módulo de virtualización que convierte el kernel de Linux en un hipervisor, garantizando un

rendimiento mucho más alto, dada su cercanía al hardware físico, al mismo tiempo que garantiza un alto nivel de seguridad.

Para la realización de pruebas, los desarrolladores cuentan con múltiples herramientas que permiten hacer testing en sus proyectos de software. Para los desarrolladores de Laravel, las pruebas son extremeadamente importantes, y el framework está construido de tal manera que facilite la creación de pruebas automatizadas, esto mediante el conjunto de herramientas de PHPUnit. PHPUnit es un marco de trabajo de pruebas automatizadas que permite a los desarrolladores crear varios tipos de tests, y asertar que los resultados sean los deseados por el programador. Creado en el año 2001 por Sebastian Bergmann, el proyecto continúa su desarrollo y es actualmente uno de los frameworks más populares para la creación de pruebas automatizadas.

Todas las tecnologías mencionadas anteriormente son relevantes en el contexto de desarrollo de software contemporáneo orientado a servicios web, y permiten generar proyectos que son escalables y eficientes. En el desarrollo del aplicativo presentado, se hará uso de las tecnologías presentadas en este marco, así como otras herramientas presentadas en el marco conceptual.

Marco conceptual

De acuerdo con Jasmin Software (n.f), un control adecuado del inventario en las organizaciones permite generar una mayor productividad, brindar un mejor servicio al cliente al tener una cantidad de inventario existente para adquirir o consumir los servicios de la organización, y garantiza una claridad en el stock disponible de las empresas, así como la minimización de los errores operativos por falta de inventario y sobrecostos en la adquisición de inventario adicional, así como garantizar un mayor flujo de caja por controlar de mejor manera el inventario. Sin importar la naturaleza de las empresas, esto es de suma importancia, y en el caso de un colegio, el inventario es destinado en su mayor parte en dar una educación adecuada para los estudiantes.

Según la INESEM Business School (abril, 2023), existen diferentes metodologías de inventario que permiten dar a las organizaciones diferentes formas de llevar un control de sus existencias. En todas ellas, los factores que influyen sobre la gestión que se lleva es relacionadas a características como peso, ubicación, dimensiones y cantidad de existencias. Algunas metodologías de control de inventario son:

- Just In Time (JIT): método de control de existencias orientado a la reducción de tiempos de producción, que se involucra en todo el ciclo de la creación del producto. El mayor beneficio de JIT es que permite calibrar la cantidad exacta de insumos necesarios para producir un artículo, así como el tiempo en que se requiere adquirir dichos bienes. Una desventaja de esta metodología es que se debe tener en cuenta los tiempos de los

proveedores, ya que los proveedores pueden tardar en dar respuesta a las solicitudes de adquisición.

- First-In, First-Out (FIFO): es una metodología usada en inventarios que tienen bastante movimiento de artículos de su almacén o en donde los productos caducan rápidamente, en la que las existencias que entran primero al depósito de inventario deben ser las primeras en enviarse a los consumidores.
- Last-In, First-Out (LIFO): metodología usada para rotaciones de inventario en las que nos productos no tienen fecha de vencimiento o no pierden valor a medida que transcurre el tiempo, contrario a la metodología FIFO.
- Control ABC: también llamado inventario ABC, es un método de clasificación de inventario en los que se etiquetan en 3 categorías que permiten diferenciar los artículos que producen mayores ingresos a las organizaciones. Se categorizan en clases A (que son de alta importancia para la organización y producen mayores ingresos), clase B (de mediana importancia) y clase C (que son de baja importancia y no producen grandes ingresos para la empresa). Este tipo de inventario permite a las organizaciones clasificar sus productos y conocer en mejor medida los productos más usados y/o adquiridos por los clientes.
- Control batch: es una metodología que permite organizar el inventario por su fecha de producción o por número de lote, agrupando dichos productos. Esto es especialmente importante para los productos perecederos.

Las tecnologías actuales permiten dar un seguimiento total a los artículos de inventario existentes, y controlar el flujo de entradas y salidas en forma de historiales. La construcción de

un software dedicado a la gestión de inventarios facilita en mayor medida el seguimiento del inventario en las empresas.

Los sistemas de información desarrollados en la actualidad hacen uso de una multitud de herramientas y paradigmas que apoyan el funcionamiento, mantenimiento o creación de las aplicaciones existentes. Para el desarrollo del sistema de información presentado, se han usado las siguientes herramientas presentadas a continuación:

Un sistema de control de versiones es un tipo de software especializado en el seguimiento de los cambios presentados en una cantidad de archivos delimitada, los cuales son almacenados en repositorios. Posteriormente, estos cambios pueden ser confirmados y compartidos con otros usuarios que hagan uso del software de control de versiones. Los desarrolladores de software hacen uso de estas herramientas con el fin de compartir el código de sus proyectos y controlar las versiones de los cambios en el software desarrollado. Las herramientas más usadas para el control de cambios es Git o Apache Subversion. Git es el más usado de todos, desarrollado inicialmente por Linus Torvalds para mantener su proyecto más complejo, Linux. Se caracteriza por ser un sistema de control de versiones descentralizado, en el que cada usuario tiene su propia copia del repositorio principal. Se lanzó en el año 2004 como un proyecto de código abierto, y es el sistema de control de versiones más dominante en el mundo, inspirando a la creación de servicios como SourceForge, GitHub, BitBucket y GitLab. Para el desarrollo de la plataforma, se usó el servicio de GitHub.

Para las bases de datos, se tiene en cuenta que es un sistema que permite almacenar una gran cantidad de datos de manera estructurada, usando objetos llamados tablas, en los que cada columna tiene asociado un tipo o valor. Las bases de datos más populares hacen uso de SQL (Structured Query Language, lenguaje estructurado de consultas), el cual es un lenguaje estandarizado usado en las bases de datos relacionales creado en los años 1970, de los cuales se han desprendido una multitud de motores de base de datos con sus implementaciones de SQL que extienden las funcionalidades del lenguaje original. Algunos ejemplos de motores populares son MySQL, Oracle, MariaDB, Microsoft SQL Server, entre muchos otros. Para el desarrollo del sistema de información se hará uso del motor de base de datos PostgreSQL, un motor de base de datos popular por ser de código abierto y ser altamente escalable para uso profesional y empresarial, que salió en el año 1996. Inicialmente fue desarrollado por el equipo de la Universidad de California en Berkeley, y surgió como la evolución del proyecto Ingres, un sistema de base de datos relacional de los años 70. Además, el proyecto PostgreSQL buscaba eliminar ciertas limitaciones existentes en los sistemas de base de datos más usados en la década de los 80. La alta escalabilidad y funcionalidades de concurrencia y replicación hacen de PostgreSQL un excelente sistema de base de datos relacional, que es usado en muchas organizaciones a nivel mundial como Microsoft, GitLab, Meta, entre otros.

Adicionalmente, es importante reconocer que el aplicativo hace uso de uno de los marcos de trabajo más populares de PHP, los marcos de trabajo, también llamados frameworks directamente, son un conjunto de herramientas desarrolladas con el fin de simplificar el proceso de desarrollo de software en un lenguaje en específico, siguiendo patrones que permitan ayudar al desarrollador a generar nuevas funcionalidades de manera iterativa y rápida. Al hacer uso de

un framework como Laravel para PHP, se simplifica en gran medida el uso del lenguaje de programación, simplificando el enrutamiento, ciertas medidas de seguridad contra ataques, simplificar la interacción con la base de datos e impulsar un patrón de diseño basado en el Modelo-Vista-Controlador (MVC – Model-View-Controller). Este patrón de diseño se encarga de distribuir la lógica de la aplicación, conteniéndola en diferentes secciones para facilitar el mantenimiento del sistema de información, donde los modelos implementan lógica que interactúa con la base de datos, la vista se encarga de renderizar el contenido para el usuario final, y el controlador se encarga de la lógica general del aplicativo.

En este sentido, el patrón de diseño MVC utiliza de manera extensiva el paradigma de programación orientada a objetos (OOP – Object Oriented Programming). Este paradigma es uno de los más usados en el mundo de desarrollo de software, y consiste en encapsular la lógica, funcionalidad y datos de una estructura en clases que después son transformadas en objetos al momento de ejecutar dicho software. Una de las ventajas principales de este paradigma es la reutilización de código, ya que las clases creadas pueden ser extendidas o heredadas por otras clases, lo que permite que dicha funcionalidad esté disponible para la clase que se está creando. En este caso particular, PHP es un lenguaje de programación multiparadigma que admite programación orientada a objetos y la programación funcional, aunque uno de los paradigmas más usados en PHP es el OOP.

El uso de las herramientas anteriores permite desarrollar un sistema de inventario que es fácil de usar, atractivo de manera visual y de alto rendimiento, que puede ser configurado de manera rápida en caso de que deba de crearse una nueva instancia de la aplicación, o deba de ser

reconfigurada. Debido al framework usado, también es sencillo agregar nuevas funcionalidades a la plataforma existente.

Todo desarrollo de aplicativos debe tener en cuenta la aplicación de una metodología de desarrollo de software, la cual juega un papel crucial en la organización de proyectos y garantizar un producto de calidad al usuario final, esto mediante la aplicación de técnicas, buenas prácticas y herramientas en el desarrollo del proyecto, haciendo uso de una correcta planificación y diseño.

No es posible hablar de la aplicación de metodologías de desarrollo de software sin conocer el ciclo más básico de desarrollo de proyectos de esta índole. Normalmente, un ciclo de desarrollo de software contiene las siguientes etapas:

- Planificación: en esta etapa, se pacta una reunión con el cliente que permita definir los requerimientos del proyecto. Esto se realiza mediante una entrevista que permita conocer las necesidades particulares del cliente. A partir de los requerimientos, se plantea un alcance del proyecto a realizar.
- Análisis: esta fase permite la organización de los requerimientos anteriormente adquiridos, lo cual permite validar nuevamente todos los requisitos del proyecto, y establecer todas las necesidades implícitas del sistema.
- Diseño: la etapa de diseño permite al equipo estudiar todas las opciones que pueden ser implementadas para dar cumplimiento a los requisitos anteriormente pactados. Esto incluye estructuras de base de datos, diseños de interfaces gráficas, frameworks de desarrollo, entre muchos otros.

- Desarrollo o codificación: en esta etapa se hace la construcción del software, de acuerdo a los requerimientos y el diseño propuesto.
- Pruebas: la fase de pruebas consiste en la validación de calidad del programa desarrollado, verificando que haya la cantidad de errores menores posibles antes que sean encontrados por el usuario final. Esta fase también verifica que se cumplan con los requisitos pactados en el proyecto.
- Entrega: esta fase, también conocida como instalación o despliegue, es en la cual se hace la entrega del software al usuario final, colocando el software en funcionamiento para ambientes productivos. Este despliegue varia de acuerdo al tipo de software desarrollado.
- Mantenimiento: esta etapa del software permite hacer las correcciones pertinentes al programa desarrollado. Si se desea, también se pueden agregar funcionalidades nuevas o adaptarlo de acuerdo a nuevas necesidades.

Existen múltiples metodologías que implementan el ciclo de desarrollo de software, de forma más rígida o más flexible, entre los cuales se encuentran:

- Metodologías tradicionales: son metodologías que aplican el ciclo de desarrollo de software de una manera bastante rígida. Algunas de las metodologías más conocidas relacionadas a las tradicionales son:
    - Metodología en cascada: las fases son únicas y dependen entre si, por lo que es un ciclo de desarrollo secuencial.
    - Metodología en espiral: las fases se realizan de manera secuencial, pero al pasar a la fase de mantenimiento nuevamente se pasa por la fase de planificación, por lo que esta metodología es cíclica.

    - Metodología incremental: en esta metodología el software se construye de manera progresiva, por lo que cada funcionalidad se agrega de manera iterativa.

- Metodologías ágiles: las metodologías ágiles son de las más utilizadas en la actualidad, esto debido a la alta flexibilidad y a la minimización de riesgos asociados a la creación de proyectos. Todas las metodologías ágiles comparten los principios del Manifiesto Ágil, propuesto en el año 2001, en el cual se prioriza la cooperación y la participación de los miembros del proyecto, adaptación a los cambios e interacción constante con los clientes. Las diferentes metodologías que implementan el Manifiesto Ágil dividen los requerimientos del proyecto en diferentes tareas que permiten agregar nuevas funcionalidades y correcciones de manera constante, que se aplican en ciclos. Esto se traduce en mayor satisfacción para el cliente y propicia una mejora constante a los proyectos realizados con estos principios, aplicando el ciclo de desarrollo de software tradicional de diferentes formas. Es posible aplicar estos principios sin usar una metodología en particular, pero existen múltiples metodologías que implementan estos valores, como lo son:

    - Metodología SCRUM: es la metodología más conocida relacionada a las metodologías ágiles. El equipo se subdivide en diferentes roles (SCRUM Master – líder del equipo, Product Owner – líder del proyecto, SCRUM Team – equipo de desarrollo) y se realiza una serie de reuniones díarias (daily stand-up, de máximo 15 minutos, en las que se responde: ¿qué se hizo ayer? ¿Qué se hará hoy? ¿Qué problemas presento actualmente?). Debe planificar un ciclo de desarrollo, conocido como sprint, normalmente de 4 semanas, en el que se implementan qué tareas van a ser realizadas por el equipo. Las tareas normalmente son agregadas al

backlog y luego son asignadas a cada sprint. Es recomendable para proyectos en los que se enfatice la cooperación e intervención de todos los miembros del equipo, y en los cuales esta intervención sea crucial para el éxito del proyecto.

- Metodología Kanban: es una metodología basada en encontrar el equilibrio entre el trabajo y la disponibilidad de los miembros del equipo. Esta metodoloigía es mucho más visible, basándose en el uso del tablero Kanban, en el que se muestran las etapas en las que se encuentra el progreso de una tarea en específico (normalmente: pendiente, en progreso y terminado). Dado a que no se involucran o crean roles ni procesos adicionales, esta metodología puede ser mezclada en cierta medida con otras metodologías ágiles, específicamente es vista con SCRUM, por lo que es más un marco de trabajo que una metodología en si.
- Metodología Extreme Programming (XP): es una metodología recomendada únicamente para startups o empresas pequeñas. Su principal objetivo es fortalecer la relación entre empleados y clientes, potenciando el trabajo en equipo y la comunicación, por lo que la calidad del software y su adaptación a los cambios es alta, esto mediante ciclos de desarrollo cortos. Se recomienda para proyectos con alta volatibilidad en los requerimientos y tiempos cortos de desarrollo.
- Rapid Application Development (RAD): podría hablarse de esta metodología como una de las precursoras al Agile. Surgió como respuesta directa a los problemas que posee el método tradicional en cascada. Su enfoque es en el desarrollo y entrega de prototipos de alta calidad y que, en cada entrega, se incorpore una retroalimentación basada en el prototipo anterior. Es recomendable para usar en proyectos de mediana o baja complejidad, en el que se posean una

serie de requerimientos bien definidos. Esta metodología fue la seleccionada para el desarrollo del proyecto SIMCAI.

Marco normativo

En concordancia con el Artículo 65 del Acuerdo 0029 de Diciembre 13 del 2013, acuerdo en el cual se expide el Reglamento Estudiantil de la Universidad Nacional Abierta y a Distancia (UNAD) y se dictan otras disposiciones, el proyecto se acoge como un proyecto aplicado en la línea de ingeniería de sistemas. A partir de lo anterior y teniendo en cuenta el Artículo 66 del mismo Acuerdo, es importante mencionar que el proyecto aquí desarrollado es un desarrollo tecnológico, lo cual implica una transferencia social de conocimiento, saberes que fueron adquiridos durante el pregrado de ingeniería de sistemas.

Teniendo en cuenta lo anterior, en base al Artículo 13 del Acuerdo No. 006 de Mayo 08 del 2014, por el cual se reglamentan en el capítulo 5 (situaciones administrativas), capítulo 6 (situaciones académicas) y el capítulo 8 (opciones de grado) del Acuerdo 029 del 2013, que expidió el Reglamento Estudiantil de la Universidad Nacional Abierta y a Distancia (UNAD) y se dictan otras disposiciones, este documento desarrolla y evidencia las siguientes etapas propias de los proyectos de desarrollo tecnológico, subtipo de proyecto aplicado:

I. Creación de un nuevo producto o proceso.
II. Las pruebas experimentales y ensayos necesarios para su concreción.
III. La elaboración de prototipos previos al inicio de la explotación industrial y comercial.

En base a lo anteriormente expuesto, el documento cumple con los requisitos solicitados para proyectos de desarrollo tecnológico.

Marco empírico

Técnicas de recolección de información

La técnica de recolección de información usada en el desarrollo de este proyecto fue principalmente la entrevista y visitas de campo a la institución. Recordemos, la entrevista, según Kvale (2011), es un método que genera la posibilidad que los sujetos puedan expresar su perspectiva acerca de un evento o suceso usando sus palabras propias, lo cual posibilita a los investigadores inferir e interpretar los hechos desde diferentes perspectivas y no solo su visión del mundo, convirtiéndola en un método de alta importancia para las investigaciones de tipo cualitativo o mixto. Se complementan con la visita de campo o investigación de campo, la cual, según QuestionPro (n.f), permite a los investigadores recolectar datos de forma cualitativa mediante la comprensión e interacción con las personas en su entorno natural, es decir, fuera de un laboratorio o un espacio controlado. La recolección de información al realizar la visita de campo se incluye la observación, el análisis de documentos y de objetos relacionados con el estudio. La entrevista y la visita de campo se aplicó en el desarrollo de este proyecto de la siguiente manera:

En el mes de febrero del 2023, se envió un correo electrónico en el cual se enviaba un saludo y se solicitaba al coordinador del colegio, José Heraldo Criollo, una reunión con el docente encargado del área de sistemas. El coordinador compartió conmigo los datos del docente Juan Carlos Jaramillo, encargado del área de sistemas, con quién luego se pactó una reunión el día 3 de marzo del 2023, con el fin de recolectar la información mediante una entrevista en la

cual se escuchó información acerca de cómo se manejaba el proceso de inventario y cotizaciones actual del colegio. Dicha reunión, realizada en acompañamiento del señor Juan Carlos Aguirre, encargado del almacén, y el rector de la institución educativa, Fray Prospero Arciniegas Zaldua. En dicha reunión, se encontraron los siguientes detalles y/o solicitudes, sin orden particular:

- No se posee un sistema de información de inventario y movimiento de activos
- El control realizado actualmente se lleva en una hoja de cálculo de Excel.
- Los bienes existentes no poseen una hoja de vida sistematizada.
- Se genera un acta de entrega de los activos a los miembros de la institución cuando se les brinda inventario.
- Es importante asociar cada activo a una orden de compra realizada anteriormente.
- En las actas de baja es posible asociar varios activos fijos.
- Cada dada de baja debe de tener asociado un soporte.
- El ingreso a almacén puede realizarse con varios productos al mismo tiempo, y lleva como soporte la factura de la adquisición realizada.
- Se realizan reportes de ingreso y salidas mensuales.
- Las cotizaciones realizadas son de licitación privada, y se basan especialmente en referencias positivas de los proveedores, así como el precio y calidad del producto a adquirir.

Esta información sentó la base para realizar la propuesta del proyecto del Sistema de Información de Movimiento y Control de Activos Institucional, y, a partir de esta información, fue posible formular una serie de requisitos y el generar alcance del proyecto a desarrollar.

Población

La implementación del software en el proceso de inventariado tiene un impacto positivo en toda la comunidad educativa, especialmente en el área de inventario y de contabilidad, las cuales son las más beneficiadas con el control de los activos y productos que están en rotación y en el almacén. El inventario es administrado por el señor Juan Carlos Aguirre, la única persona que controla el ingreso y salida de productos del almacén, ubicado en el sótano del colegio. El área de contabilidad es también una sola persona.

Al implementar el software, se benefician los procesos administrativos, ya que se genera una gestión sobre los activos y productos más eficiente, lo cual, a largo plazo, permite controlar el presupuesto relacionado a la adquisición de bienes y materiales, reducir costos, lo cual se traduce en un mayor beneficio que puede ser invertido en la educación impartida a los estudiantes del colegio, siendo la población mayoritaría de la institución educativa.

Metodología investigativa

El desarrollo de este proyecto hace uso de una metodología de investigación cualitativa, en la que se realizó una observación al proceso de inventario del Colegio San Francisco de Asís, así como una entrevista con el técnico encargado del almacén de inventarios y suministros, el docente encargado del área de sistemas de la institución y el rector de la institución. Haciendo uso de esta metodología, se hizo una recopilación de los requerimientos que posee el colegio para el desarrollo del sistema de información, en los que también se realizó una observación a los formatos existentes de control de inventario, y un proceso de identificación a las falencias presentadas a los procedimientos actuales.

Recordemos, una metodología de investigación cualitativa, según Deslauriers, J. (2004), es un tipo de investigación que produce y analiza datos descriptivos acerca de un acontecimiento o suceso. Se caracteriza por tener un interés más profundo sobre los casos de prueba y muestras que son analizadas a mucho más detalle sin necesariamente tener un enfoque totalmente matemático o estadístico. Los datos matemáticos si tienen lugar en estas investigaciones, pero no son el enfoque principal de la indagación realizada. Esta metodología hace uso de las entrevistas y de la observación como su principal método de recolección de datos, y se basa adicionalmente en las experiencias obtenidas por el investigador o, en este caso, los entrevistados, acerca del suceso investigado.

Para el desarrollo de este proyecto aplicado, no se hace uso de recolección de encuestas, dado a que solamente se necesita recolectar la información relacionada a los procesos existentes

del colegio relacionado a las cotizaciones y al sistema de inventario actual del colegio. Es importante resaltar nuevamente que la técnica de recolección de información usada es la entrevista, método tradicional para la adquisición de requerimientos funcionales para el desarrollo de aplicaciones en el mundo de la ingeniería de software.

Metodología de desarrollo

Para el desarrollo de este sistema de información, se ha hecho uso de una metodología de desarrollo ágil orientada a la realización de iteraciones frecuentes, brindando resultados positivos en la construcción del software SIMCAI. La metodología usada es la de Desarrollo Rápido de Aplicaciones (Rapid Application Development) la cual, según Microsoft (n.f), es una metodología de desarrollo ágil propuesta por James Martin en 1991 en la cual el principal objetivo es la creación de aplicaciones mediante iteraciones frecuentes y de aprobación continua de los clientes, usando la retroalimentación de los usuarios finales como la base fundamental para la construcción final de la aplicación. La mayor ventaja de usar esta metodología es que, gracias a la participación frecuente de los clientes y de su aprobación durante la construcción del software, se desarrolla un producto con una alta usabilidad de acuerdo a la retroalimentación de los usuarios finales, y permite dar una entrega rápida gracias a estas iteraciones. El uso de esta metodología directamente influye sobre:

- La reducción del tiempo de desarrollo y la aceleración de la entrega: esta es la ventaja principal, dado a que los clientes reciben un software estable gracias a las pruebas realizadas en diferentes iteraciones y a los comentarios dados a los desarrolladores, que permiten realizar ajustes rápidos.
- Mejora de flexibilidad y adaptabilidad: dada a la entrega constante de resultados, es importante que el equipo desarrolle una aplicación que sea fácil de configurar y que pueda adaptarse a la retroalimentación de los clientes. Esto incurre en que se desarrolle una aplicación flexible y adaptable.

- Mejor gestión de riesgos: dada a la participación activa de los clientes, es posible generar una matriz de riesgos que involucre la retroalimentación de los usuarios finales de la aplicación.
- Menos programación manual y tiempos de prueba más cortos: aunque esto es dependiente de la forma en que esté desarrollada el proyecto, el uso de esta metodología incurre en tiempos de prueba más cortos debido a que ya se han construido prototipos anteriores, sobre los cuales se sigue construyendo el aplicativo.
- Comentarios de los usuarios constantes, relevantes y en tiempo real: esta es la ventaja más importante de la metodología, ya que, con los comentarios de los usuarios finales, es posible realizar ajustes en la aplicación que mejoren el funcionamiento de la aplicación o incurran en facilitar procesos en el sistema de información.

Si bien comparte similitudes con ciertas metodologías ágiles, como por ejemplo SCRUM, RAD se centra en la entrega de prototipos mientras que otras metodologías se centran en la entrega de características en cada sprint del proyecto. En este sentido, comparte más similitudes con metodologías de desarrollo iterativas.

Fases de desarrollo

La metodología RAD cuenta con diferentes fases de desarrollo de los proyectos de software. Dada a la naturaleza iterativa de la metodología, estos pasos pueden repetirse en cada iteración, dependiendo de la retroalimentación de los usuarios finales:

1. Definición de requisitos del proyecto: todos los participantes del proyecto definen, investigan y finalizan el alcance y requisitos del proyecto de software, así como sus objetivos, expectativas y plazos de entrega. Una de las principales ventajas de la metodología es que, aunque los requisitos ya hayan sido pactados, si existe algún cambio relacionado a los requerimientos, pueden ser cambiados fácilmente.
2. Creación de prototipos: el equipo de desarrollo comienza la creación de modelos y prototipos, produciendo un flujo de trabajo que permita la incorporación de cambios de forma rápida. La finalidad de la creación de estos prototipos de forma rápida es presentarlo a las partes interesadas. Las iteraciones iniciales de la creación de estos prototipos son las más cruciales y permitirán definir la facilidad con que los cambios puedan ser incorporados, esto sin sacrificar la calidad del software desarrollado.
3. Creación de pruebas e incorporación de la retroalimentación: el uso de pruebas automatizadas y pruebas de caja negra son esenciales en la construcción de proyectos que usen esta metodología. Dada a la naturaleza de estas pruebas, es fácil encontrar soluciones que permitan incorporarse en futuros prototipos.

Finalización e implementación: esta etapa de finalización consiste en realizar una versión optimizada para uso de producción del prototipo final que sea estable y fácil de mantener.

Aplicación de la metodología de desarrollo

La metodología de Rapid Application Development se aplicó en el proyecto de SIMCAI de la siguiente manera:

- Se realizó una reunión inicial en la que se recolectó la información para generar una serie de requerimientos funcionales y no funcionales suficientes para satisfacer las necesidades particulares del Colegio San Francisco de Asís.
- Se iteró sobre diferentes diseños y mockups que permitieron establecer diferentes prototipos internos. Se evaluaba en cada prototipo que se alineara con los requerimientos inicialmente solicitados y que el software fuese satisfactorio de usar.
    - Adicionalmente, cada prototipo era versionado en GitHub para llevar un control de las versiones iniciales del proyecto.

Figura 2.
Listado de tags de git, en el repositorio de GitHub del proyecto.

Fuente: propia

- Se presentó un prototipo que incorporaba todos los requerimientos inicialmente solicitados, lo cual permitió generar una retroalimentación que subsecuentemente fue aplicada en el desarrollo del siguiente prototipo. El prototipo inicial fue de gran agrado para la institución, por lo que se continuó trabajando sobre este prototipo.
- Se implementaron pruebas automatizadas en el backend para garantizar la calidad del software.

Propósito

El propósito del proyecto es, a partir de la información recolectada, generar un sistema de información que atienda a las necesidades específicas del Colegio San Francisco de Asís. Para ello, se plantean varios mockups, diagramas y posibles casos de uso de acuerdo a los requisitos de la institución educativa.

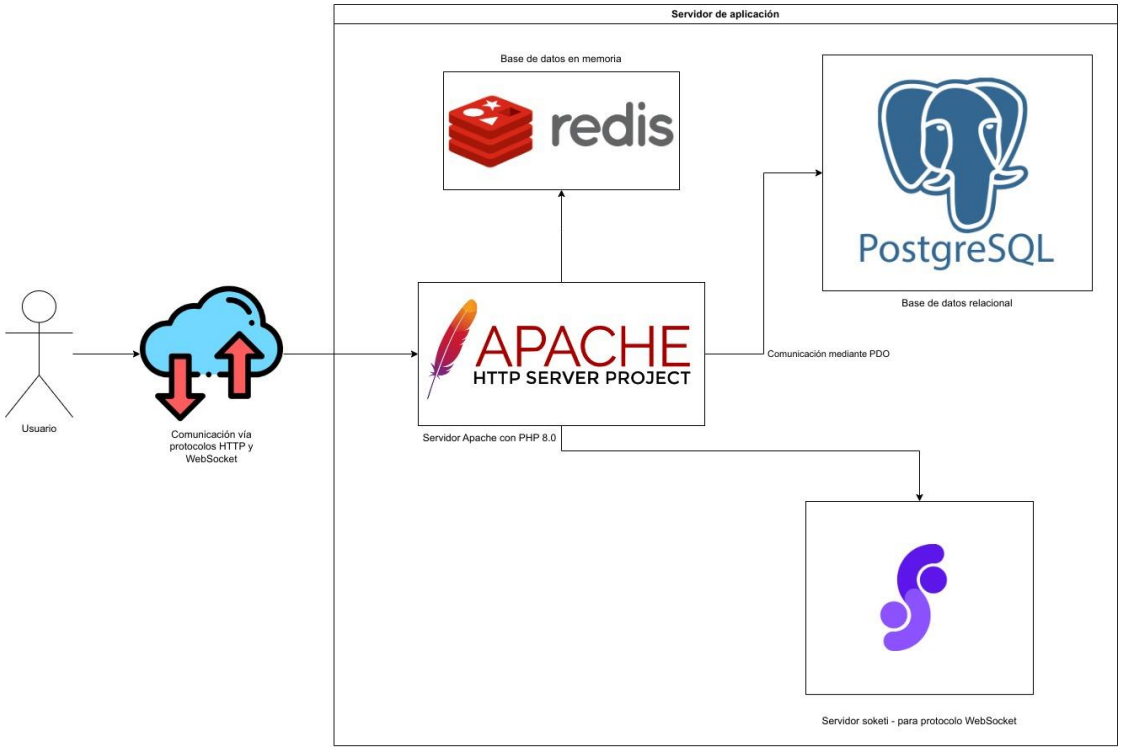
Características de la aplicación

La aplicación fue desarrollada atiende a las necesidades planteadas por el Colegio San Francisco de Asís, adicionando también las siguientes funcionalidades que complementan a los procesos en la aplicación y/o a los miembros de la institución educativa:

- Auditoría a nivel de aplicación de los cambios realizados por los usuarios.
- Sistema de permisos y roles que pueden ser asignados a diferentes usuarios.
- Notificaciones en tiempo real y por correo electrónico de diferentes eventos en el sistema.
- Soporta el inglés y español como idiomas oficiales de aplicación.

Funcionamiento de la aplicación

Figura 3.
Diagrama de arquitectura.

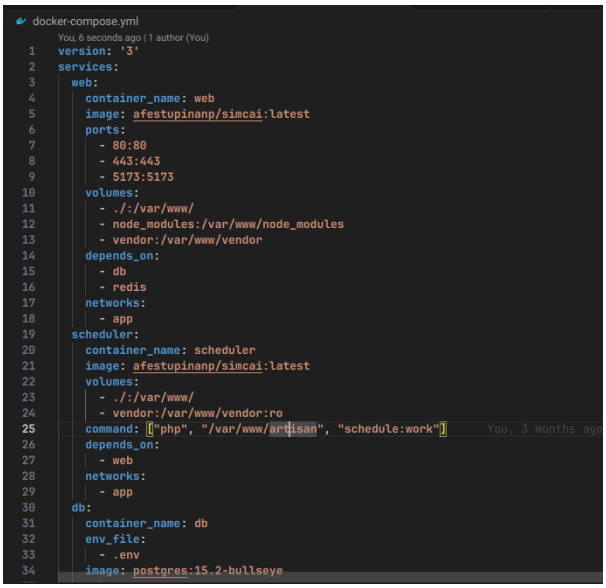


Fuente: propia.

El usuario envía una petición HTTP o se conecta mediante protocolo WebSocket desde su navegador al servidor de aplicación (esto para el sistema de notificaciones en tiempo real), mediante el puerto 80. El servidor Apache, corriendo PHP ejecuta los scripts relacionados a la aplicación de SIMCAI, esto mediante atendiendo solicitudes a ciertos enlaces en la aplicación, redirigiendo estas peticiones a controladores del framework Laravel que, subsecuentemente, ejecutan cierta lógica e interactúan con la base de datos mediante el ORM (Object Relational-Mapping) de Eloquent. El servidor entonces retorna una respuesta en diferentes formatos (ej: HTML, JSON) al navegador que realizó la petición inicial.

En el diagrama de arquitectura ilustrado anteriormente muestra la configuración por defecto de la aplicación usando Docker, donde se instalan estos componentes de forma sencilla, esto mediante el uso de un archivo de configuración para la imagen general usada por la aplicación (Dockerfile), y la configuración a nivel general (docker-compose.yml). Para evitar la reconstrucción frecuente de la imagen principal usada, se genera una vez y se carga en DockerHub, la plataforma de distribución de imágenes oficiales de Docker, y permite su descarga desde cualquier equipo.

Figura 4.
Archivo docker-compose.yml



Fuente: propia.

Paralelamente, se ejecutan tareas programadas usando el scheduler de Laravel y se ejecutan procesos en segundo plano relacionados a funcionalidades pesadas de la aplicación mediante el uso de colas, implementado también por Laravel. Ejecutar estas tareas en segundo plano permite que la aplicación funcione en condiciones óptimas, esto mediante la creación de

procesos que ejecuten dichas tareas, similar a conceptos en otros lenguajes de programación relacionados a los hilos y paralelismo.

Garantizar que se elabore una lógica simple pero eficiente permitirá que el servidor responda a las peticiones de forma rápida, garantizando un tiempo de respuesta adecuado para el usuario final, y mejorando la satisfacción del cliente.

Requerimientos técnicos

Dada a la forma en que está desarrollado el sistema de información, al desplegarse en un contenedor de Docker, se posee todo el software necesario para permitir el funcionamiento correcto del sistema de información. En un caso de uso general, el solo se requiere de la disponibilidad del siguiente software:

- Docker 1.13 o superior
- Docker Compose V1 o superior
- Sistema operativo Windows, macOS o Linux.
    - Dada la virtualización nativa en Linux usando KVM, se recomienda el uso de este sistema operativo mucho más que Windows o macOS, garantizando mejor rendimiento.

A nivel de hardware, se requiere:

- Al menos 600 MB de RAM para un funcionamiento correcto. Se recomienda 1 GB o superior.
- Un procesador con al menos dos núcleos con 2.00 GHz mínimo.

Si se desea usar la aplicación sin usar la virtualización de Docker, a nivel de software, se requiere adicionalmente:

- PHP 8.0 o superior.
    - Extensiones de PHP: zip, xmlreader, pdo, zip, xml, gd, iconv, simplexml, zlib, redis
    - Si se desea usar PostgreSQL, usar también la extensión pdo_pgsql y pgsql.
        - La instalación por defecto usando Docker hace uso de estas librerías.
    - Si se desea usar Oracle, usar también la extensión oci8 y pdo_oci8, así como instalar la librería asociada al Oracle Client. SIMCAI no fue desarrollado con la idea de usar este motor de base de datos.
- Servidor web Apache
    - Extensión de rewrite
    - Si se desea usar certificados SSL, también habilitar la extensión mod_ssl
- Servidor redis para base de datos en memoria.
    - Alternativamente, puede usarse el file cache de proveído por Laravel.
- Instalar las tareas programadas para ejecutar los siguientes comandos:
    - php artisan queue:work: permite correr las tareas en segundo plano a la aplicación principal.
    - php artisan schedule:work: permite correr los comandos programados. Se debe hacer que corra cada minuto.

- Opcionalmente, es posible configurar una conexión a un servidor SFTP para evitar almacenar los archivos guardados en el mismo servidor de la aplicación.

Requerimientos legales

El Sistema de Información de Movimiento y Control de Activos Institucional no tiene requisitos legales a nivel particular por parte del Colegio San Francisco de Asís. Sin embargo, el aplicativo no hace uso de software o dependencias comerciales. Todo el software usado es de código abierto, usando dependencias están licenciadas mediante:

- MIT
- LGPL 2.1
- BSD-3-Clause
- LGPL 3.0
- Apache 2.0

Las anteriores licencias permiten el uso privado y/o comercial de las dependencias usadas en el proyecto a nivel de frontend con NPM y a nivel de backend con Composer/Packagist. No se modifican ninguna de las librerías usadas.

Requerimientos funcionales

Teniendo en cuenta la información obtenida en la entrevista, se formularon los siguientes requerimientos funcionales para responder a la solicitud de la institución educativa.

Tabla 1.
Tabla de requerimientos funcionales de la aplicación.

Código	Descripción
RF-01	El sistema debe permitir a los usuarios administradores asignar y quitar permisos de los usuarios del sistema.
RF-02	El sistema debe permitir a los usuarios administradores asignar y quitar las dependencias a las que pertenecen los usuarios del sistema.
RF-03	El sistema debe permitir a los usuarios administradores listar, registrar y editar los usuarios del sistema.
RF-04	El sistema debe permitir a los usuarios administradores cambiar la contraseña de los usuarios del sistema.
RF-05	El sistema debe permitir a los usuarios administradores ajustar el mensaje del día.
RF-06	El sistema debe permitir a los usuarios visualizar el mensaje del día en la página de inicio.
RF-07	El sistema debe permitir a los usuarios administradores listar y ajustar ciertas configuraciones del sistema desde la aplicación.

RF-08	El sistema debe permitir seleccionar entre el idioma español e inglés.
RF-09	El sistema debe permitir a los usuarios cambiar su contraseña desde la aplicación.
RF-10	El sistema debe permitir a los usuarios actualizar su información personal.
RF-11	El sistema debe permitir a los usuarios seleccionar su foto de perfil desde el aplicativo.
RF-12	El sistema debe permitir al encargado de almacén listar, registrar, editar y eliminar productos del sistema de información
RF-13	El sistema debe permitir al encargado de almacén realizar una carga masiva de productos al sistema de información mediante archivo Excel.
RF-14	El sistema debe permitir al encargado de almacén subir un archivo de factura al registrar el ingreso de un nuevo producto a almacén, o al realizar un cargue masivo.
RF-15	El sistema debe permitir al encargado de almacén listar, registrar y editar marcas de productos en el sistema.
RF-16	El sistema debe permitir al encargado de almacén listar las facturas cargadas al sistema.
RF-17	El sistema debe permitir al encargado de almacén listar, registrar y editar las dependencias de la institución.

RF-18	El sistema debe permitir al encargado de almacén listar, registrar, editar y eliminar los proveedores de productos de la institución.
RF-19	El sistema debe permitir al encargado de almacén listar, registrar y editar activos fijos.
RF-20	El sistema debe permitir al encargado de almacén dar de baja a activos fijos.
RF-21	El sistema debe permitir al encargado de almacén cargar un acta de baja al dar de baja varios activos.
RF-22	El sistema debe permitir al encargado de almacén listar y visualizar las actas de baja.
RF-23	El sistema debe permitir al encargado de almacén realizar movimientos de activos en la aplicación.
RF-24	El sistema debe permitir al encargado de almacén listar, registrar y editar las cotizaciones en la aplicación.
RF-25	El sistema debe permitir al usuario administrador listar, registrar, editar y eliminar cargos del sistema de información.
RF-26	El sistema debe permitir al encargado de almacén listar, registrar, editar y eliminar cotizaciones de productos en el sistema de información.
RF-27	El sistema debe permitir al usuario seleccionar enlaces que puedan agregase en la barra de navegación al submenú de “enlaces rápidos”.

Fuente: propia.

Requerimientos no funcionales

De acuerdo a la entrevista realizada, y a las funcionalidades que se desarrollarán en la aplicación, se formularon los siguientes requerimientos no funcionales.

Tabla 2.
Tabla de requerimientos no funcionales de la aplicación.

Código	Descripción
RNF-01	El sistema debe permitir el cargue de archivos.
RNF-02	El sistema debe permitir enviar notificaciones en tiempo real al realizar ciertas acciones en el sistema.
RNF-03	El sistema debe permitir enviar notificaciones por correo electrónico al realizar ciertas acciones en el sistema.
RNF-04	El sistema debe permitir enviar notificaciones por correo electrónico al realizar ciertas acciones en el sistema.
RNF-05	El sistema debe implementar un método de auditoría a nivel de aplicación para realizar seguimientos de los cambios en el sistema.
RNF-06	El sistema debe enviar una notificación automáticamente al encargado de almacén cuando haya pocos items en almacén al crear un nuevo activo.
RNF-07	El sistema debe enviar una notificación por correo electrónico diaria al encargado de almacén listando los productos con pocos items en almacén.

RNF-08	El sistema debe de enviar automáticamente por correo electrónico un reporte de movimientos del último mes.
RNF-09	El sistema debe de recordar los últimos módulos usados por el usuario, y mostrar accesos rápidos desde la página de inicio.

Fuente: propia.

Mockups, prototipos iniciales y principios de diseño

Colores

La selección de los colores en la aplicación es de alta importancia, dado a que, según Ackerman, A. (n.f) en artículo para Adobe, los colores poseen un aspecto psicológico que complementan el mensaje que se quiere dar. Una paleta de colores bien seleccionada permite transmitir el mensaje correcto al usuario final.

Bootstrap en gran medida se encarga de la selección de colores, sin embargo, la selección del color principal es donde se debe de tomar una mayor decisión.

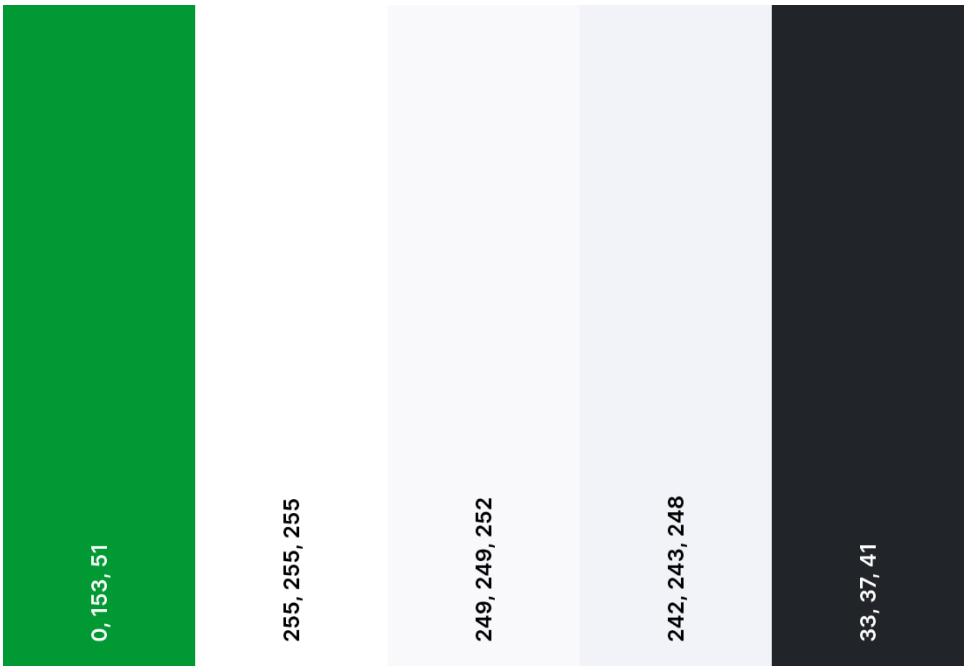
En toda la aplicación, inicialmente se usaba como principal el color verde oscuro, teniendo en cuenta la tonalidad del escudo del Colegio San Francisco de Asís.

Figura 5.
Logotipo del Colegio San Francisco de Asís.

Fuente: Colegio San Francisco de Asís.

El verde es uno de los colores más versátiles de la rueda de colores debido a su multitud de uso en objetos, productos y la misma naturaleza. El tono de verde en este caso es más opaco, lo cual le atribuye un tono más natural que también se suele relacionar, según Francia, G. (febrero, 2021) con la perseverancia, constancia, defensa, estabilidad, resistencia a los cambios, tenacidad, consolidación, posesión y control.

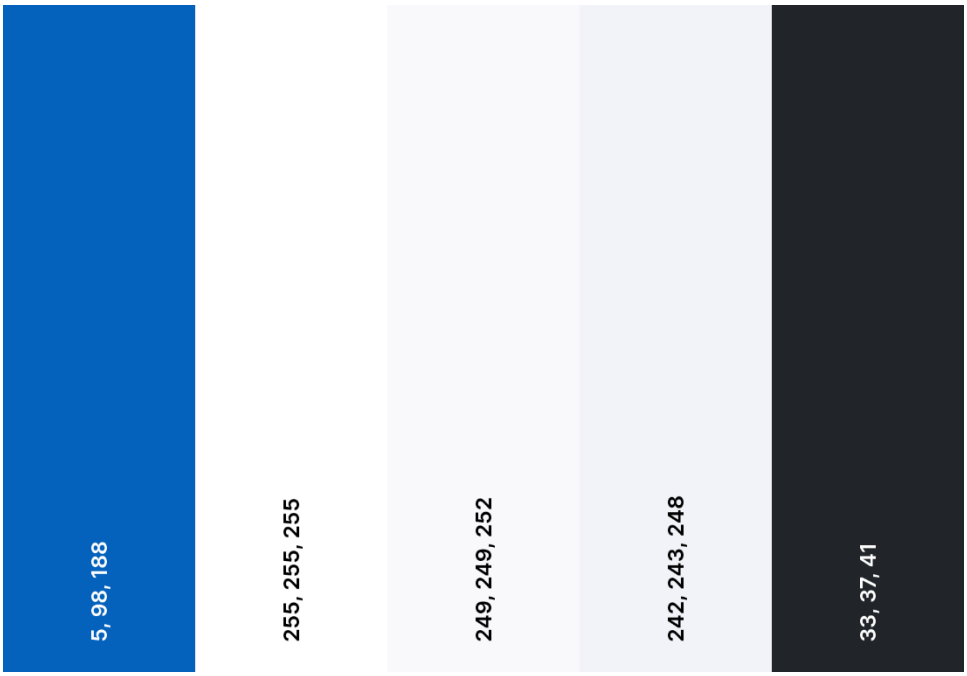
Figura 6.
Paleta de colores básica inicial.



Fuente: propia.

Si bien la idea inicial de tener el color verde podría ser agradable dada su cohesión con la institución educativa, muy prontamente era notorio que quizás, a nivel visual, no generaba mucha satisfacción al usar la aplicación. Dado a lo anterior, se reemplazó por el color azul, que también es bastante usado por el colegio, principalmente en su página institucional.

Figura 7.
Paleta de colores final.



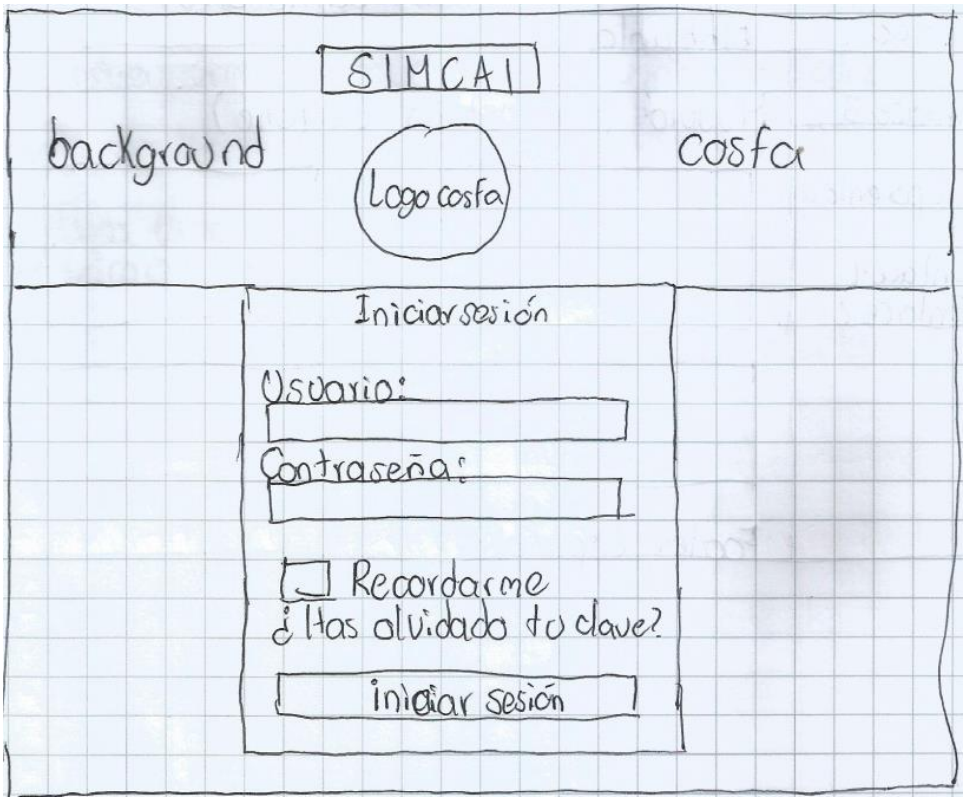
Fuente: propia.

Según Ackerman, el color azul es relajante, calmado y cercano. Es una opción que hace tomar un tono profesional a las aplicaciones desarrolladas. Genera confianza y una buena sensación, dependiendo de la tonalidad de azul. Según Przybyla, D. (n.f), es un color único y versátil que se le asocian términos como confianza, credibilidad, serenidad, inteligencia, confianza, entre otros adjetivos.

Inicio de sesión

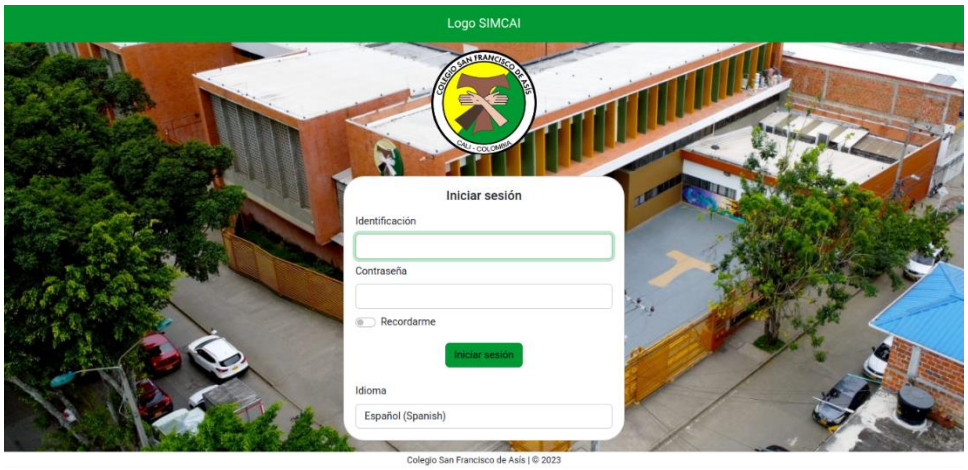
El inicio de sesión pasó por diferentes diseños, incluso desde su concepción como mockup.

Figura 8.
Diseño inicial de la página de inicio de sesión.



Fuente: propia.

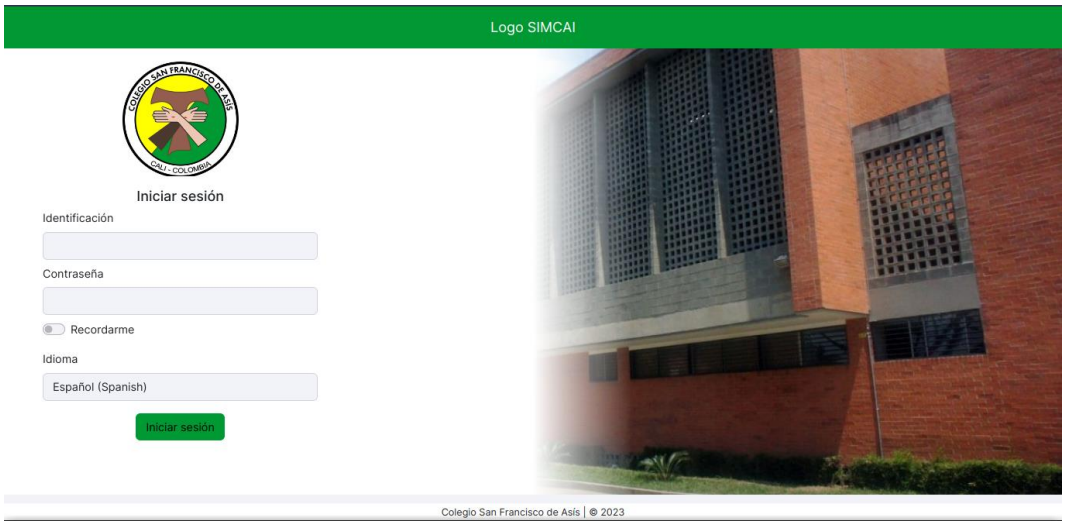
Figura 9.
Primera implementación de la pantalla de inicio de sesión.



Fuente: propia.

Si bien se implementaba de acuerdo a lo deseado en el mockup, no aprovechaba el tamaño de la pantalla, y podría resultar algo raro de usar, dado a que el footer no estaba contenido en el formulario mostrado.

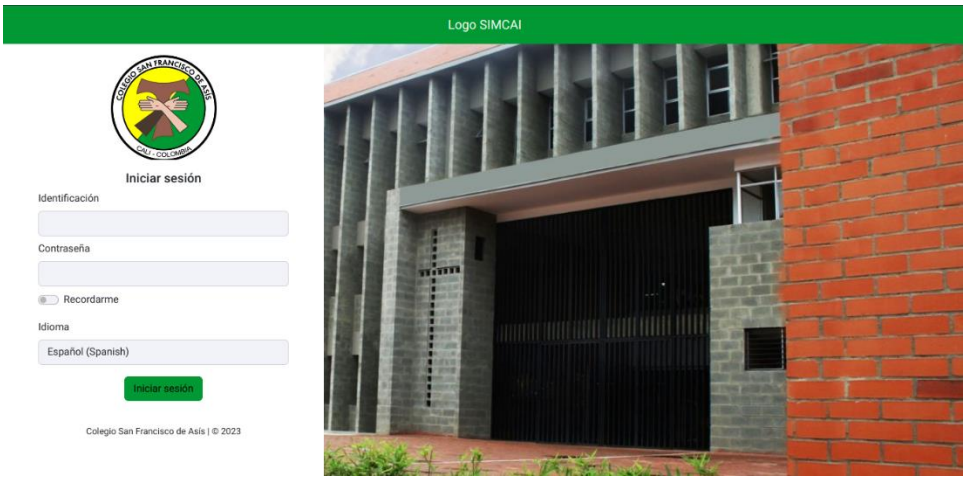
Figura 10.
Segunda versión de la pantallla de inicio de sesión.



Fuente: propia.

La siguiente versión, de julio del 2023, aprovechaba aún más el espacio en pantalla, sin embargo y se agregaba un efecto de difuminado sobre la imagen de fondo. Para este punto, se implementó también una animación al cargar la página, la cual desliza el formulario desde el lado izquierdo de la pantalla. A pesar de lo anterior, el difuminado podía hacer sentir la página como algo sin terminar, por lo cual fue eliminado para la siguiente iteración del inicio de sesión.

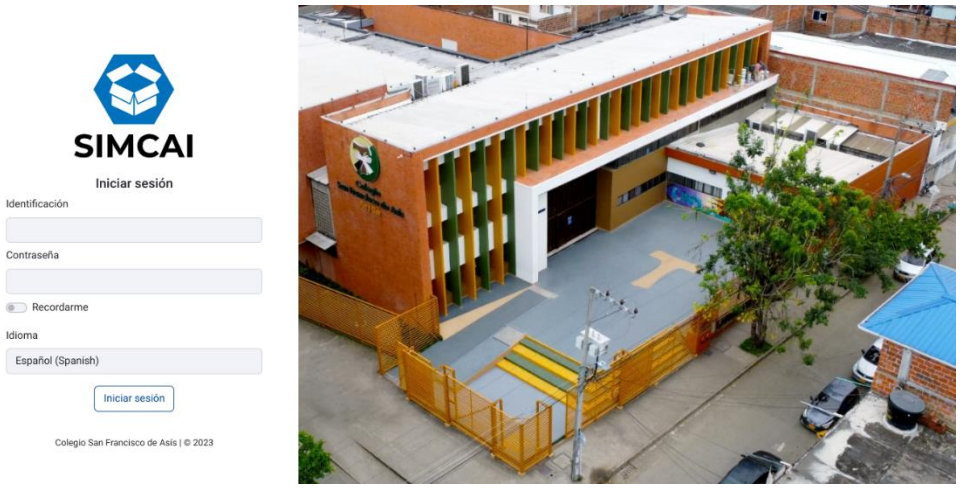
Figura 11.
Tercera iteración de la pantalla de inicio de sesión.



Fuente: propia.

Esta versión, generada en septiembre del 2023, corregía dichos problemas presentados, pero seguía sin aprovechar toda la pantalla, dado a que se tenía la barra de navegación, en la cual, al iniciar sesión, no tenía interacción alguna con el usuario. Al quitar la barra de navegación, e implementar el logo del aplicativo, se obtiene el siguiente resultado:

Figura 12.
Versión final de la página de inicio de sesión, con logotipo de SIMCAI.



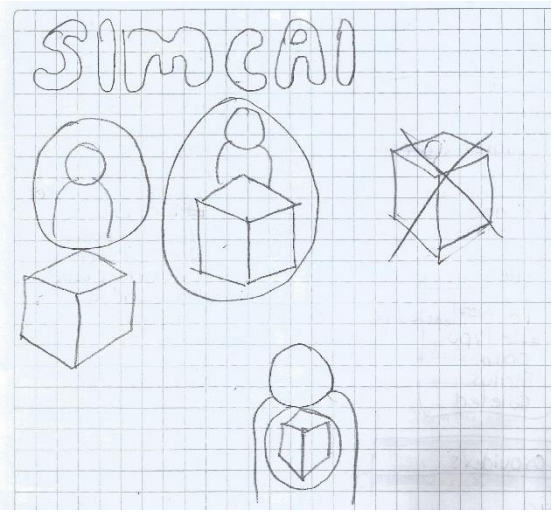
Fuente: propia.

Finalmente, en septiembre del 2023, se obtiene un inicio de sesión bastante limpio y fácil de usar, que invita al usuario a comenzar a usar el aplicativo. Al iniciar sesión, se desliza el formulario nuevamente al lado izquierdo de la pantalla.

Logotipo

Generar un logotipo fue una de las labores más complejas, dado a que, de alguna manera, debería representar el funcionamiento del aplicativo. Los conceptos iniciales presentaban el uso de una caja de inventario con el icono de usuario, pero esto no resultaba atractivo para usar, ni tampoco era único.

Figura 13.
Bosquejos iniciales del logotipo.



Fuente: propia.

Se generó un logotipo usando Adobe Illustrator, usando el formato SVG, el cual puede ser reescalado para ser usado en diferentes medios. El logotipo consiste en un icono de una caja, representando el inventario, y un hexágono, representando solidez y firmeza. Esto, junto al color azul que se desea usar en aplicativo, representa un sistema seguro, eficiente y eficaz, que es sólido en su funcionamiento.

Figura 14.
Logo tipo final de la aplicación.

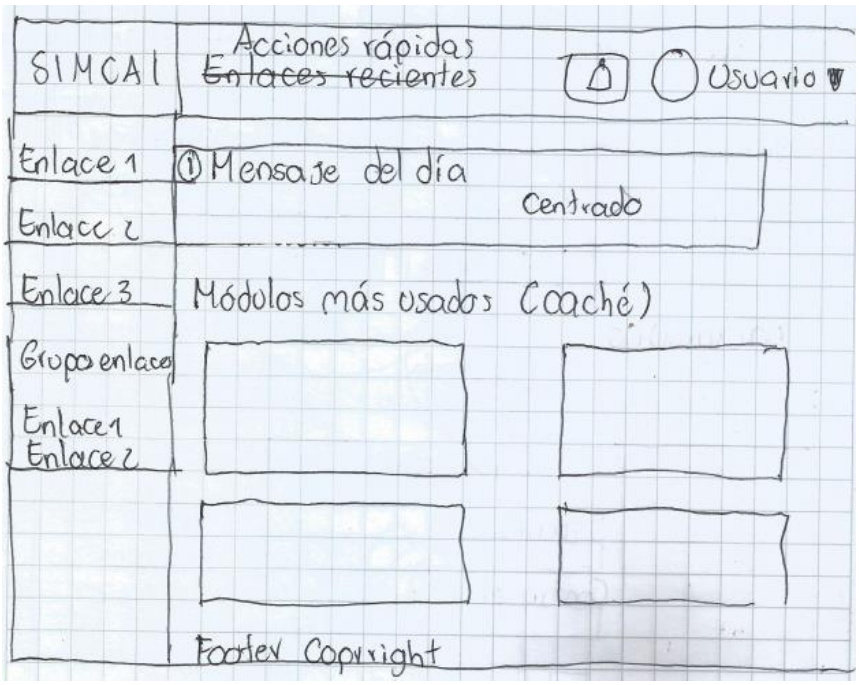
SIMCAI

Fuente: propia.

Plantilla principal

Mediante la plantilla principal se define la forma en que se verá la página de forma general. Esta plantilla es luego reusada en el resto del aplicativo, por lo que es importante generar una página fácil de usar y que sea atractiva para los usuarios.

Figura 15.
Bosquejo inicial de la plantilla del sistema.

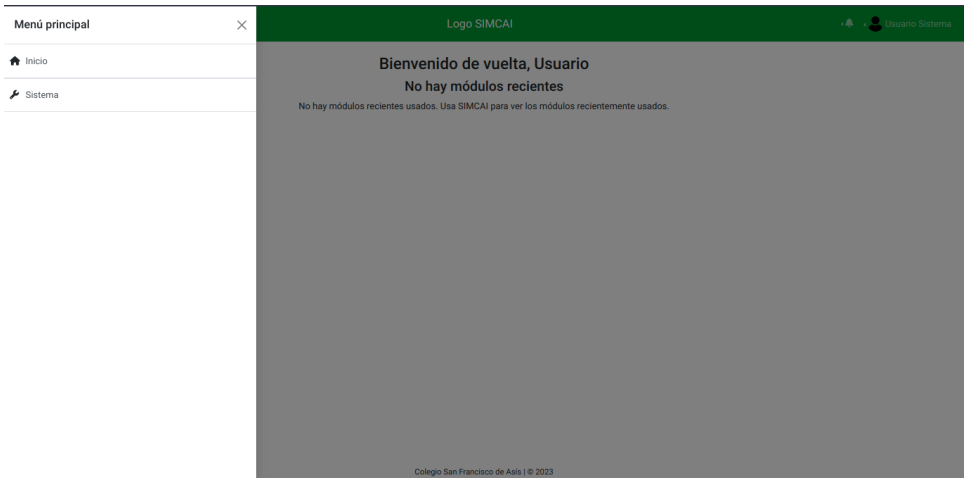


Fuente: propia.

El bosquejo presentado en la anterior figura muestra la página de inicio, en la cual se deseaba añadir una forma de mostrar los módulos del sistema recientemente usados, con el fin de agilizar el uso del sistema de información.

Inicialmente se consideraba el logotipo de la aplicación en el submenú, pero luego se encontró que para mejorar la visibilidad se puede colocar en la mitad de la barra de navegación. El menú se compacta como una barra lateral desplegable, que puede ser invocada en cualquier momento por el usuario, dando más espacio en la pantalla principal para que el usuario interactúe con la aplicación.

Figura 16.
Primera implementación de la plantilla principal realizada en Bootstrap 5.



Fuente: propia.

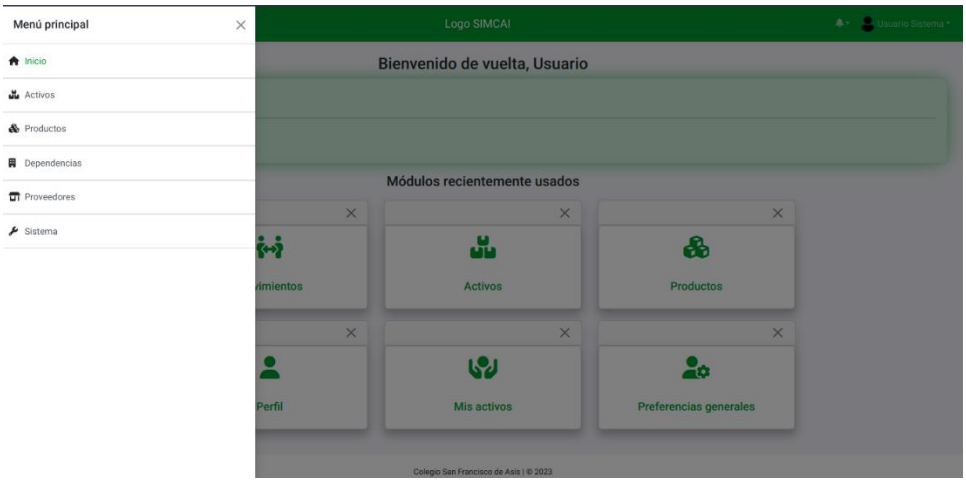
La implementación inicial estaba hecha en el framework Materialize CSS, el cual, debido a ciertas limitaciones que imponía el uso de este marco de trabajo de estilo, se reemplazó por Bootstrap 5. La figura mostrada anteriormente ya hace uso del framework Bootstrap para el estilo visual del aplicativo, e implementa ciertos patrones de diseño del bosquejo inicial.

Figura 17.
Página de inicio de SIMCAI, mostrando los módulos recientes.

Fuente: propia.

En la figura anterior se presenta la implementación de los módulos recientemente usados en el sistema, en un estado mucho más avanzado del original presentado en la Figura 16.

Figura 18.
Plantilla principal con menú lateral desplegado.



Fuente: propia

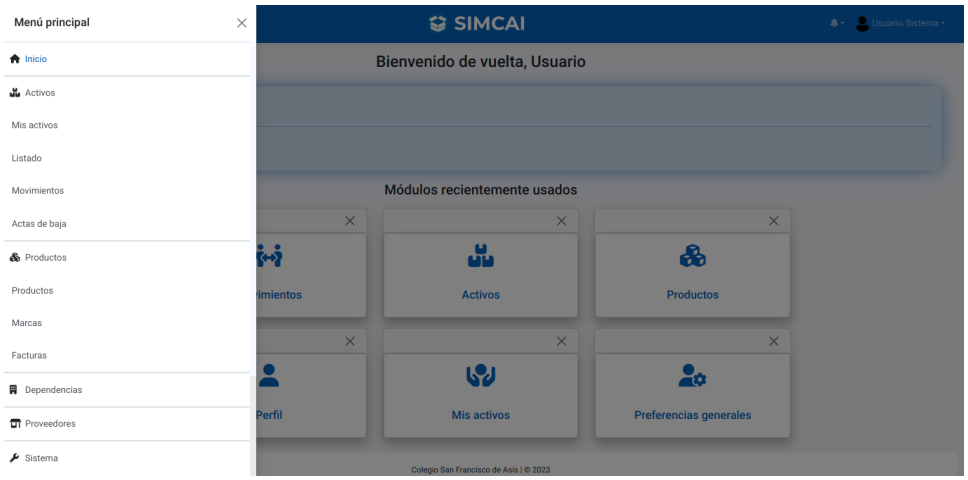
Para este punto, ya se había implementado una gran parte de la funcionalidad requerida por el colegio, sin embargo, el color del aplicativo no era muy convincente cuando solicitaba opiniones de terceros.

Después, se implementó el cambio de color y el logotipo final de la aplicación, dando como resultado una aplicación visualmente atractiva y responsiva.

Figura 19.
Versión final de la página de inicio.

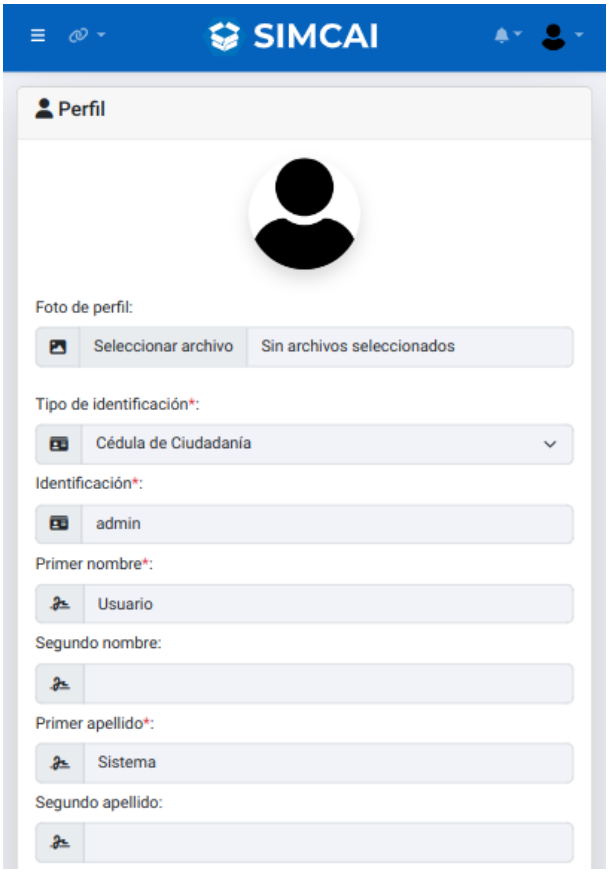
Fuente: propia.

Figura 20.
Versión final de la plantilla principal, con menú lateral desplegado.



Fuente: propia.

Figura 21.
Vista móvil de la página de perfil del usuario



Fuente: propia.

Principios de diseño y otros elementos

La fuente usada en el sistema de información es Roboto. De acuerdo con Bautista, J. (2020), Roboto es una fuente diseñada por Christian Robertson de Google en el año 2011 con el fin de reemplazar la fuente usada por el sistema operativo Android. El principal objetivo de la fuente es lograr una legibilidad alta y ser visualmente atractiva en una amplia gama de diferentes pantallas, desde monitores de computadores hasta celulares y/o relojes inteligentes. Se rediseñó en el año 2014 con el fin de mejorar ciertas críticas que le comparaban con fuentes como Helvética o Myrida. Esta fuente posteriormente fue mucho más adaptada en Android con la llegada de Material Design en Android 5.0 Lollipop, y está liberada bajo la licencia Apache 2.0 en GitHub.

Se seleccionaron diferentes fuentes como Montserrat, Inter o IBM Plex, pero se decidió por usar Roboto dada la familiaridad que tiene en los dispositivos móviles y por su alta legibilidad. A continuación se muestra una comparación de las fuentes que se deseaban usar en el proyecto, de las cuales se seleccionó Roboto, el segundo párrafo:

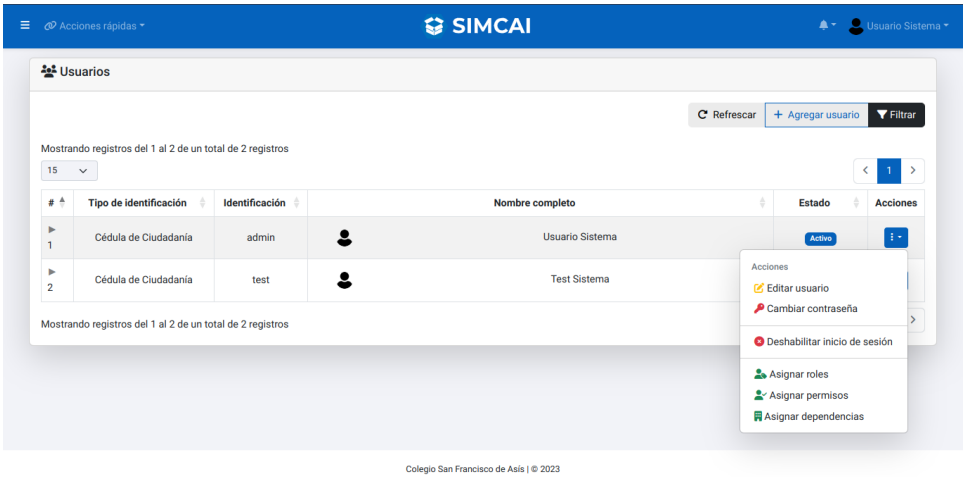
Figura 22.
Comparación de fuentes Montserrat, Roboto, IBM Plex Sans e Inter.

Fuente: propia.

En cuestiones de accesibilidad y/o diseño del aplicativo, se priorizó la siguiente reglamentación:

- Los campos de los formularios y modals deben de mostrar iconos que permitan relacionar fácilmente la información y las entidades registradas con iconografía reconocible.
- Los campos requeridos deben de estar marcados con un asterisco rojo para que el usuario pueda identificarlos mucho más fácil y más rápido.
- Priorizar la facilidad de uso en toda la aplicación.
- Asociar colores a diferentes acciones en el sistema:
    - Amarillo para edición o ajustar.
    - Rojo para acciones peligrosas.
    - Verde para acciones generales
    - Azul para acciones importantes

Figura 23.
Página de gestión de usuarios



Fuente: elaboración propia.

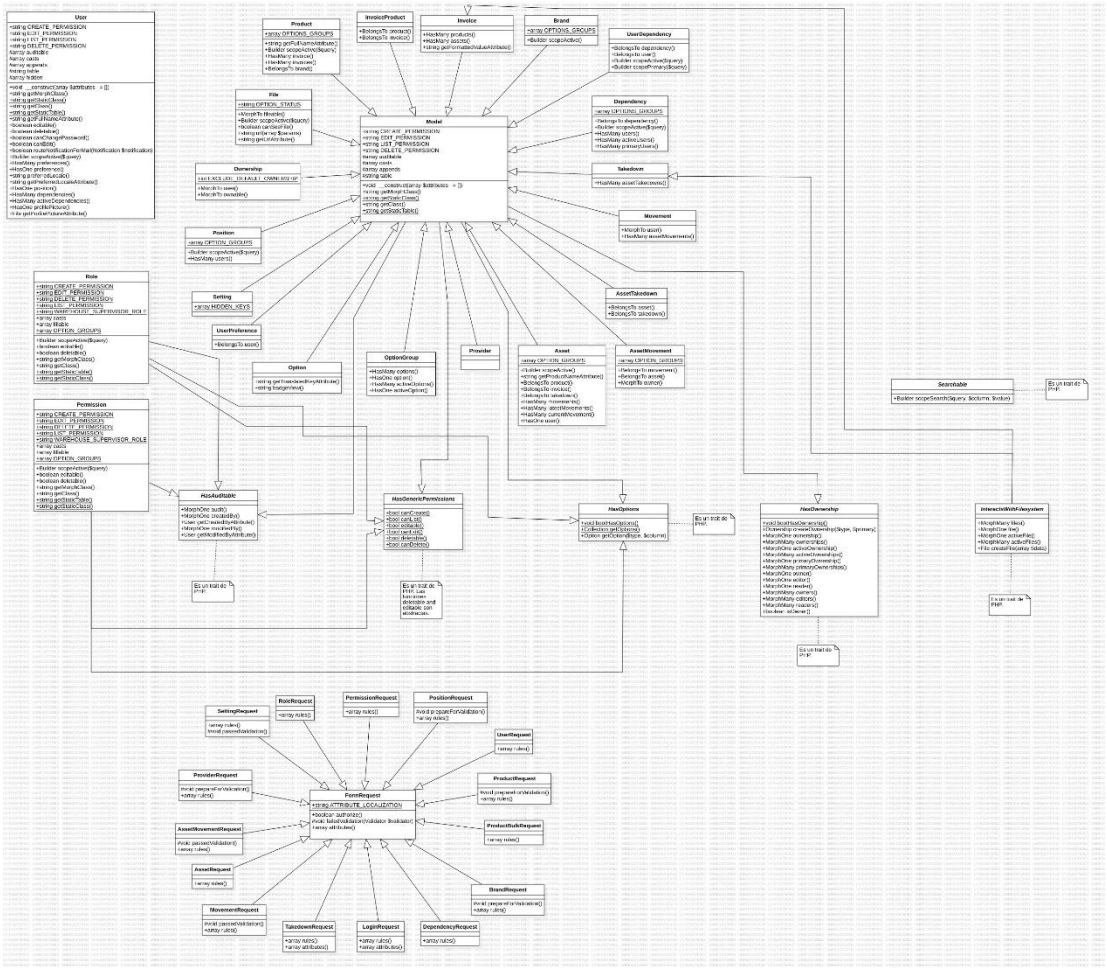
Figura 24.
Página de gestión de activos fijos.

Fuente: propia.

Diagrama de clases

Usando el software de StarUML, se construyó el siguiente diagrama de clases que representa los modelos usados para comunicarse con la base de datos, adicionalmente, se presentan los traits de PHP, usados mediante generalización.

Figura 25.
Diagrama de clases



Fuente: propia.

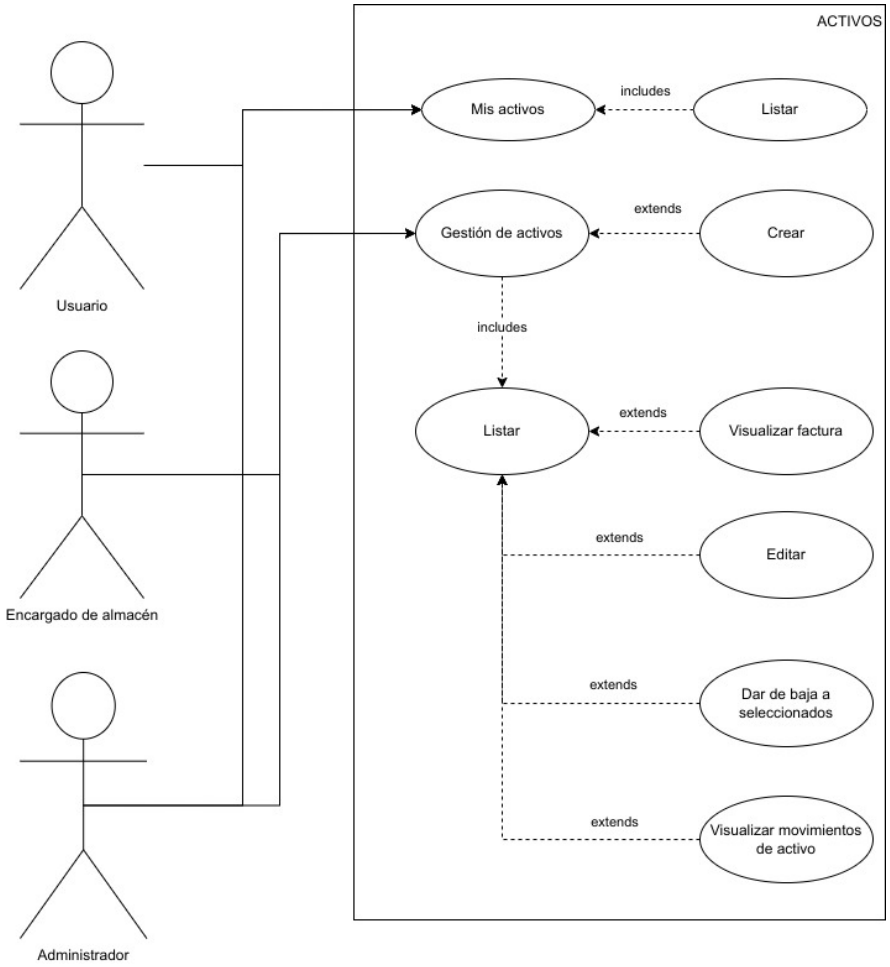
Diagrama de casos de uso

Los siguientes diagramas de casos de uso visualizan el funcionamiento del sistema desde los 3 roles más básicos: usuario general, encargado del almacén y administrador del sistema. Cabe resaltar que el administrador del sistema tiene la posibilidad de gestionar los permisos que tiene cada rol, crear nuevos roles y asignar dichos permisos a otros usuarios, por lo que los diagramas muestran el funcionamiento por defecto del sistema.

Los diagramas fueron realizados con el software de creación de diagramas de diagrams.net, un software de libre de código abierto anteriormente denominado draw.io.

CU-001: Activos

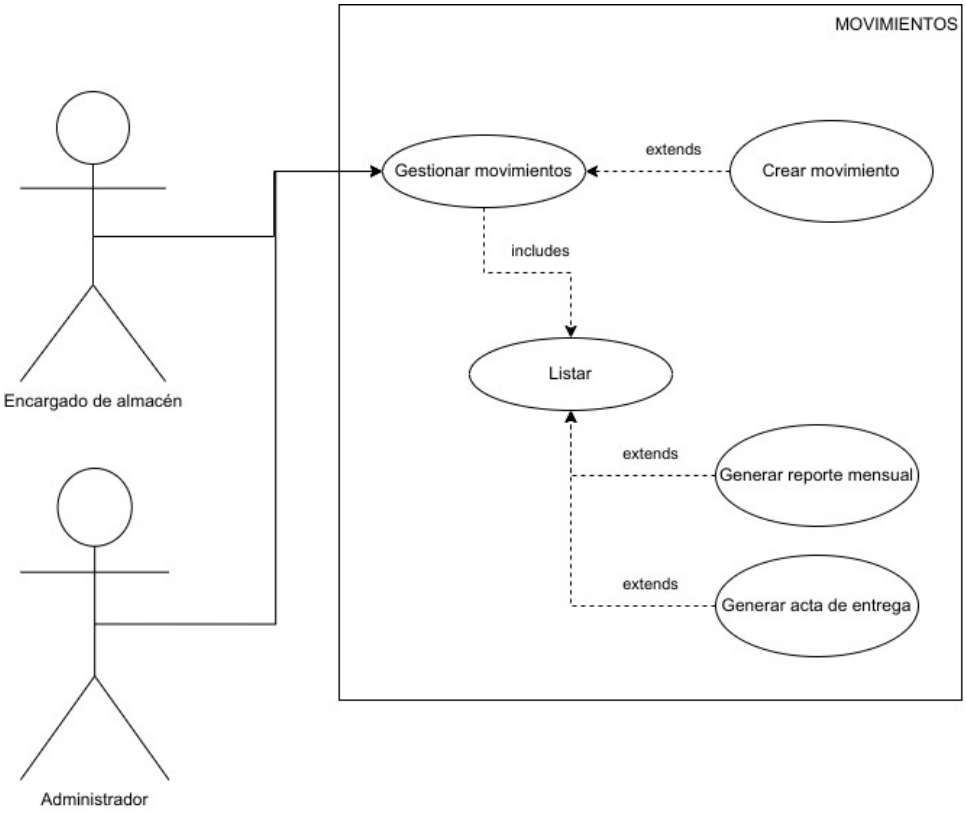
Figura 26.
Caso de uso 1 (CU-001): activos



Fuente: propia.

CU-002: Movimientos

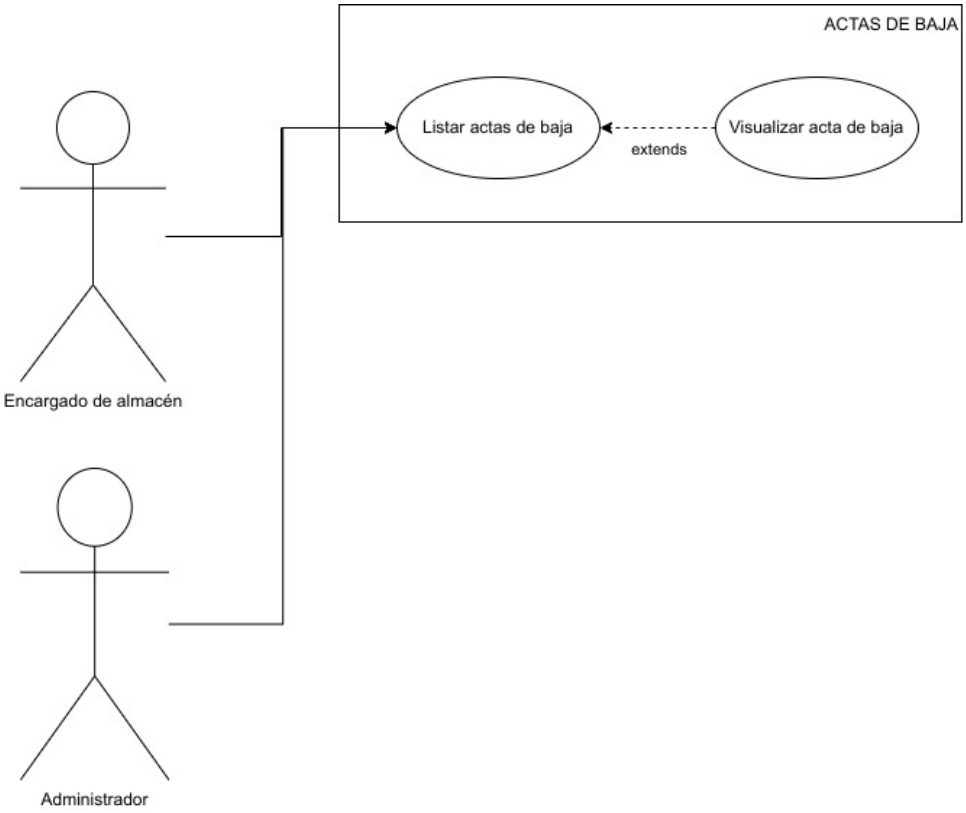
Figura 27.
Caso de uso 2 (CU-002): movimientos.



Fuente: propia.

CU-003: Actas de baja

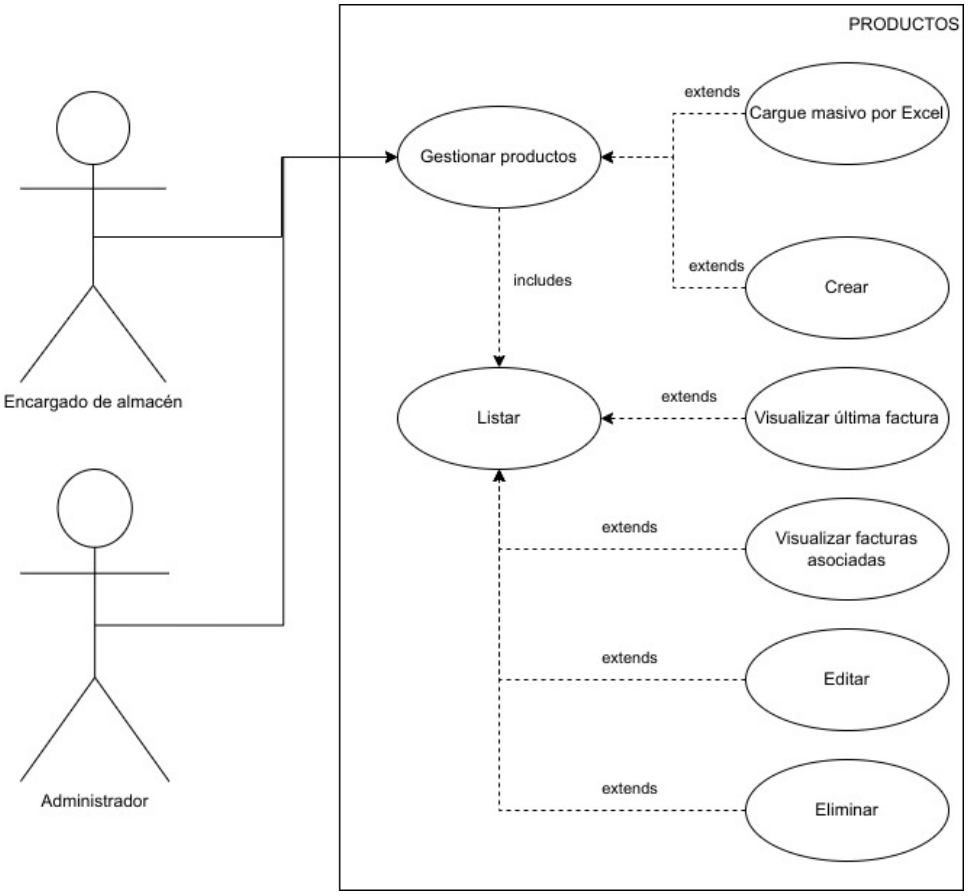
Figura 28.
Caso de uso 3 (CU-003): actas de baja.



Fuente: propia.

CU-004: Productos

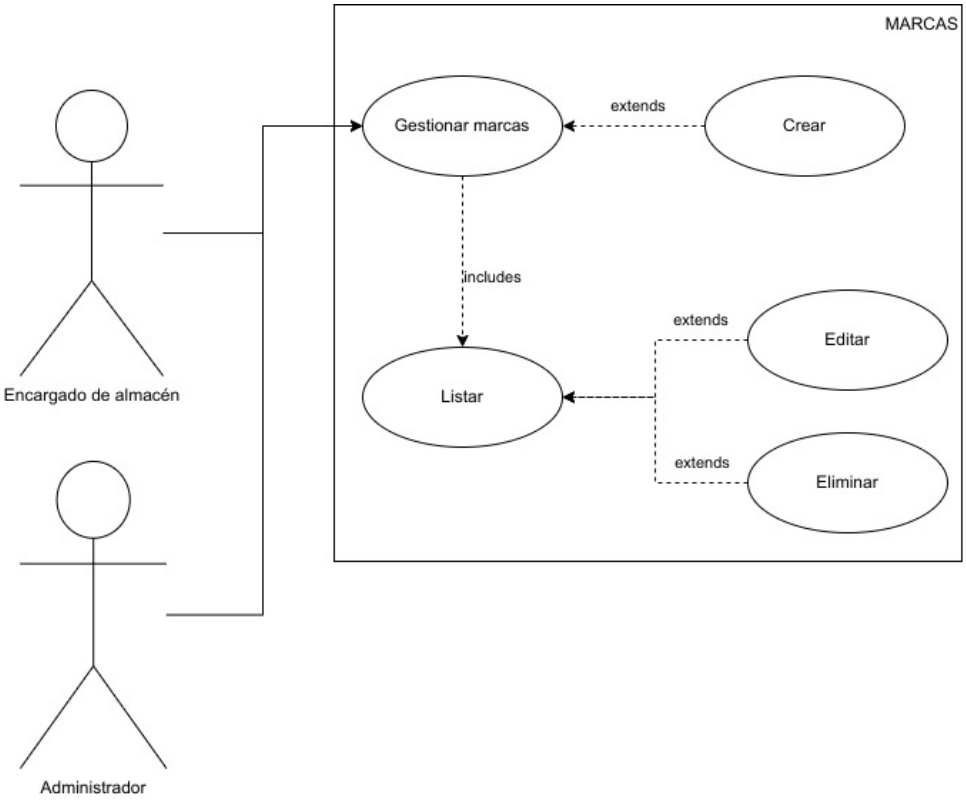
Figura 29.
Caso de uso 4 (CU-004): productos



Fuente: propia.

CU-005: Marcas

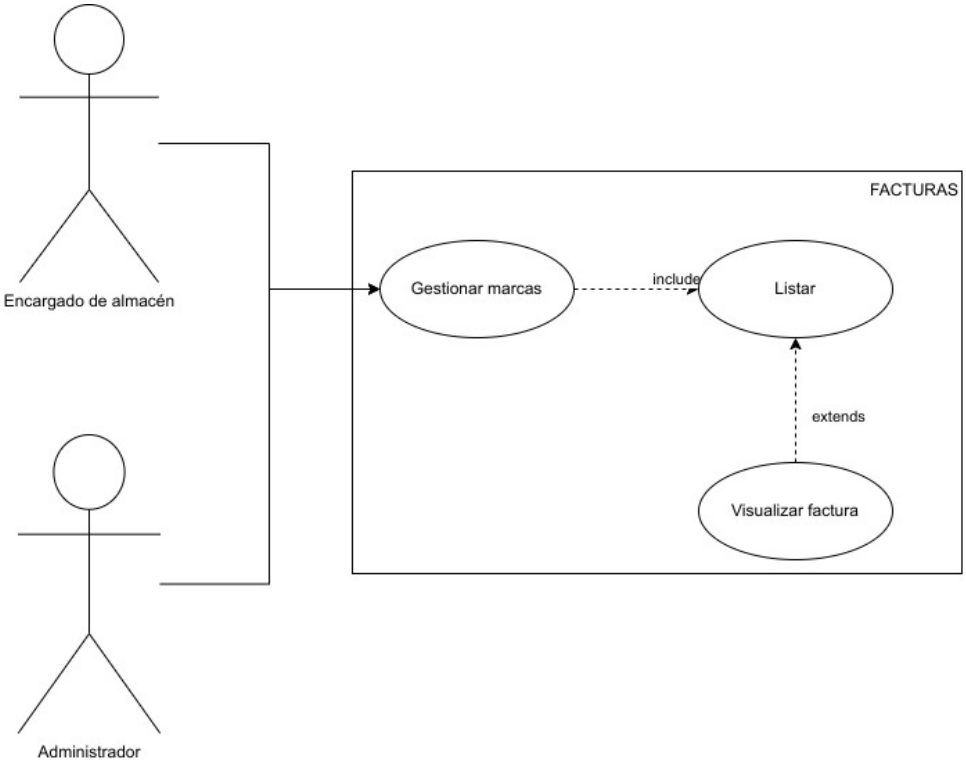
Figura 30.
Caso de uso 5 (CU-005): marcas.



Fuente: propia.

CU-006: Facturas

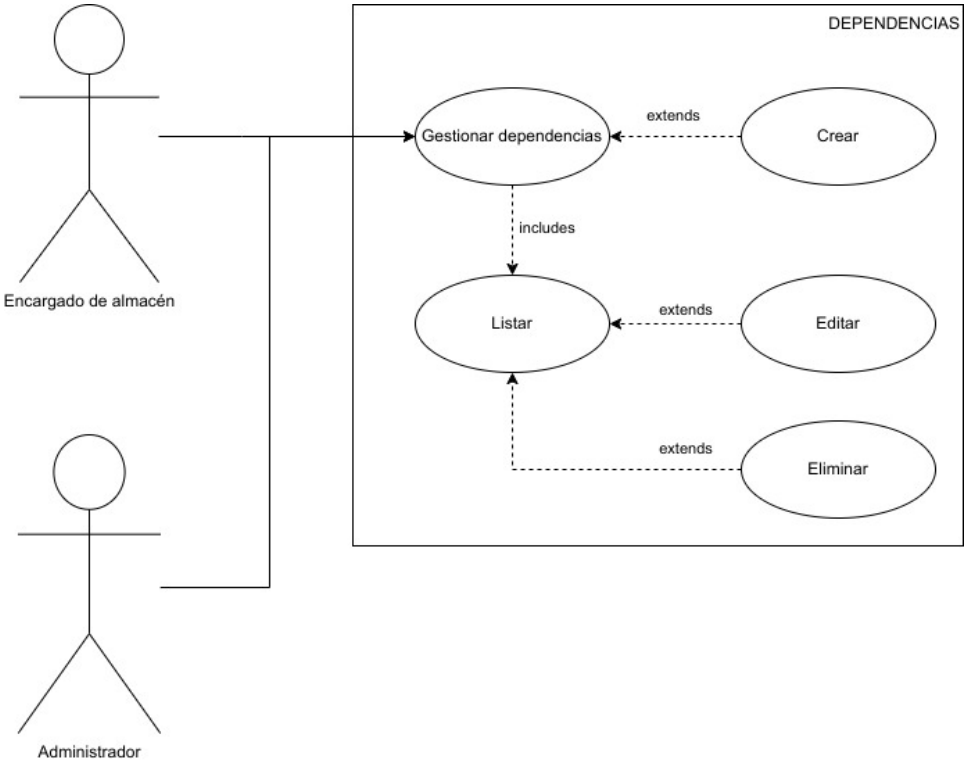
Figura 31.
Caso de uso 6 (CU-006): facturas.



Fuente: propia.

CU-007: Dependencias

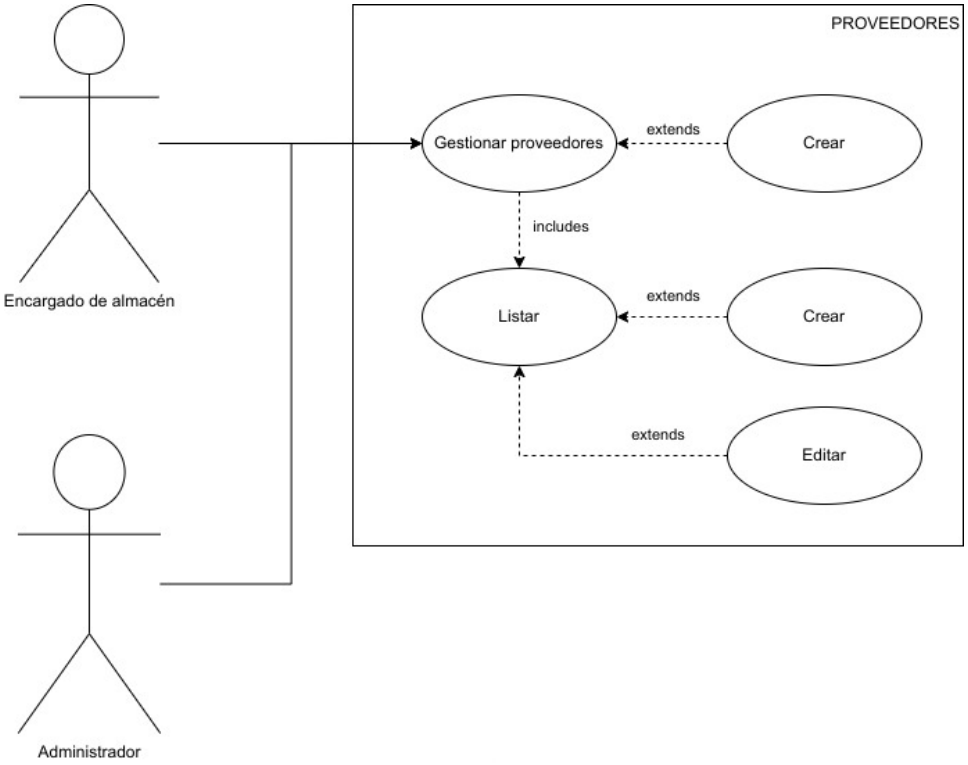
Figura 32.
Caso de uso 7 (CU-007): dependencias.



Fuente: propia.

CU-008: Proveedores

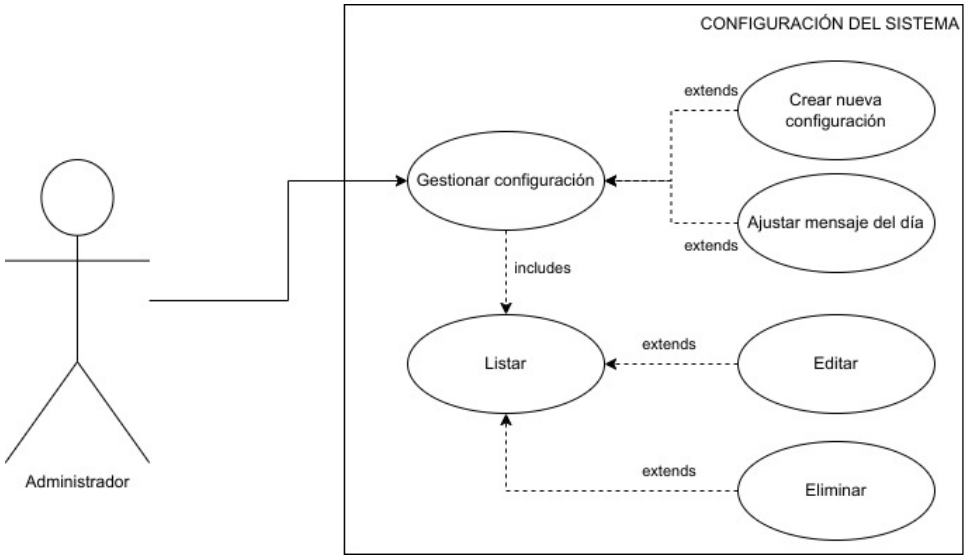
Figura 33.
Caso de uso 8 (CU-008): proveedores.



Fuente: propia.

CU-009: Configuración del sistema

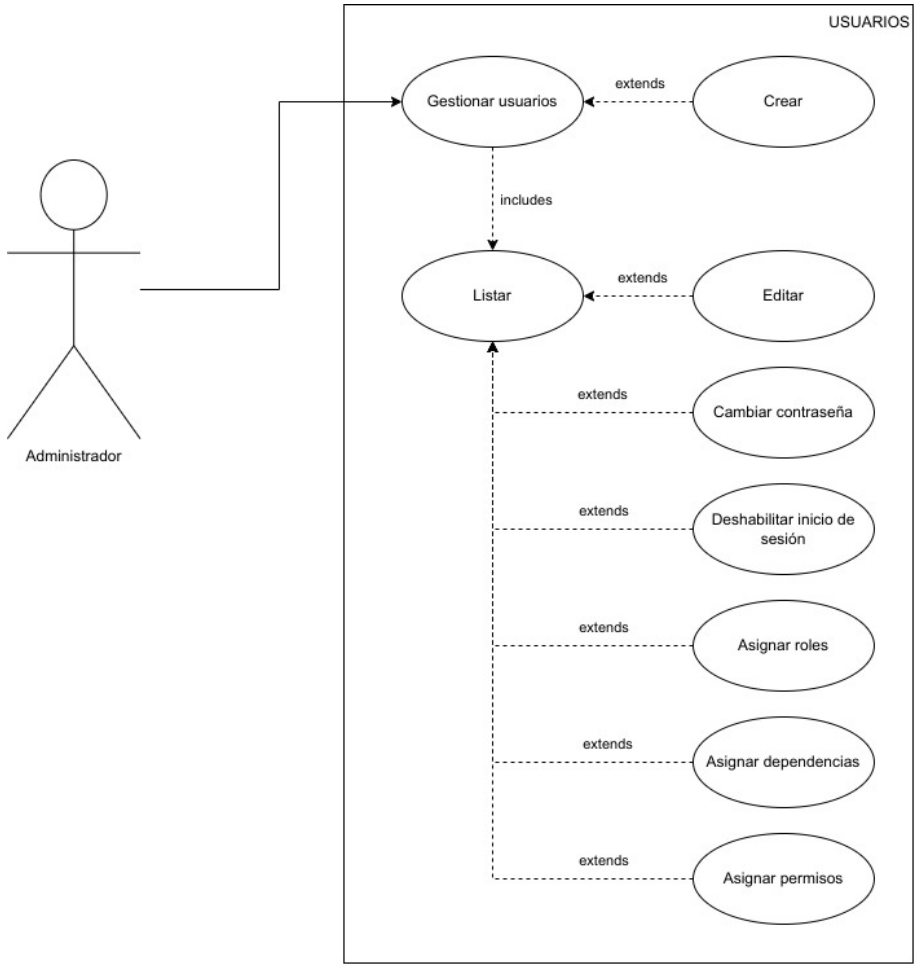
Figura 34.
Caso de uso 9 (CU-009): configuración del sistema.



Fuente: propia.

CU-010: Usuarios

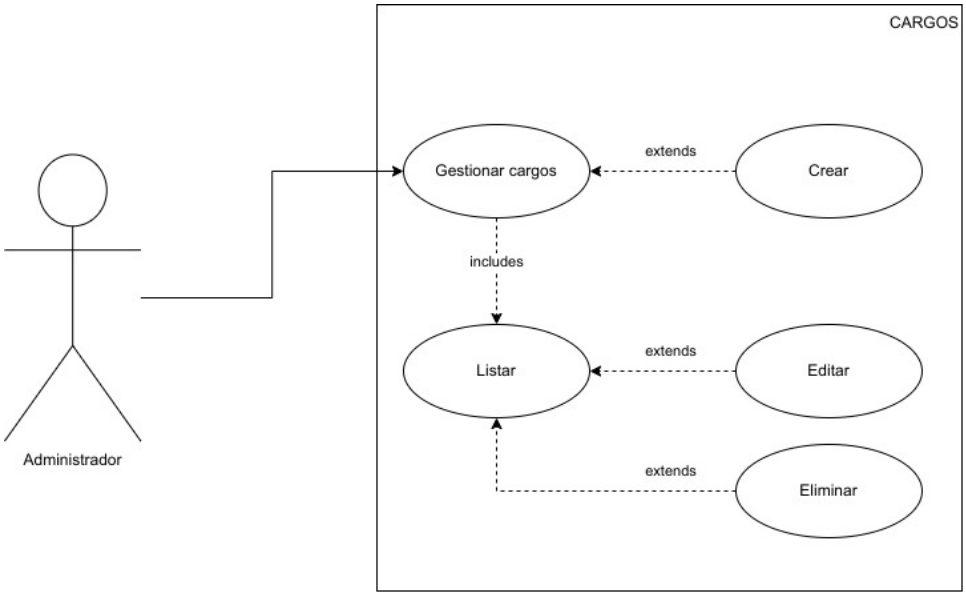
Figura 35.
Caso de uso 10 (CU-010): usuarios.



Fuente: propia.

CU-011: Cargos

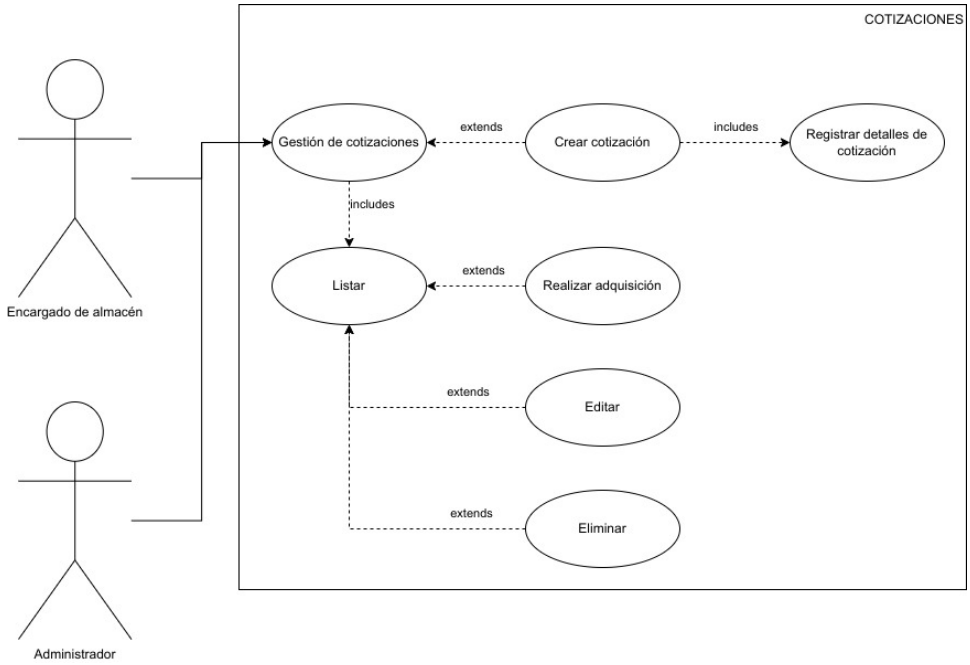
Figura 36
Caso de uso 11 (CU-011): cargos.



Fuente: propia.

CU-012: Cotizaciones

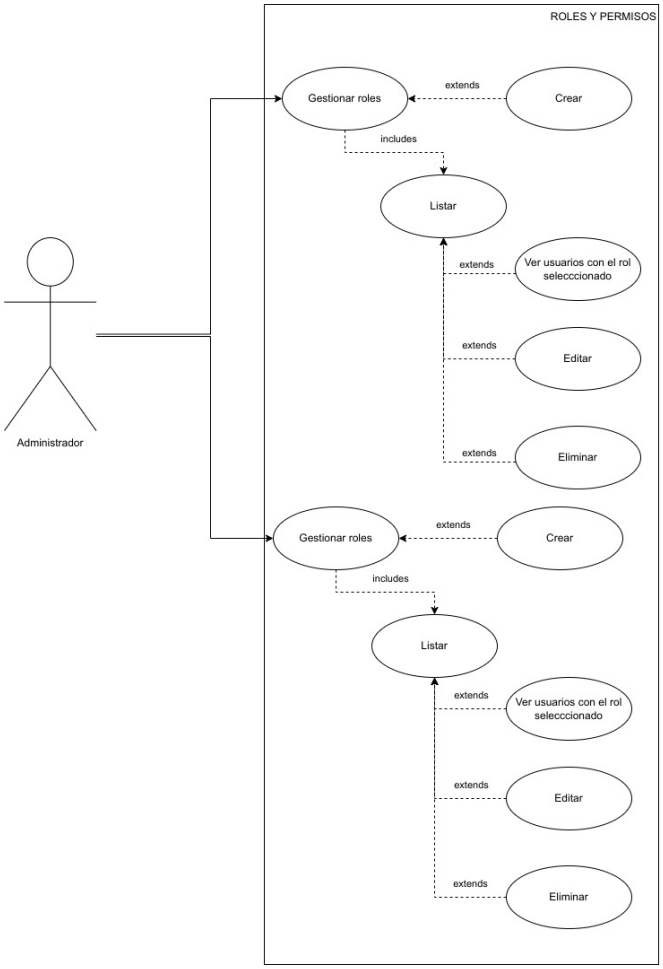
Figura 37.
Caso de uso (CU-012): cotizaciones



Fuente: propia.

CU-013: Roles y permisos

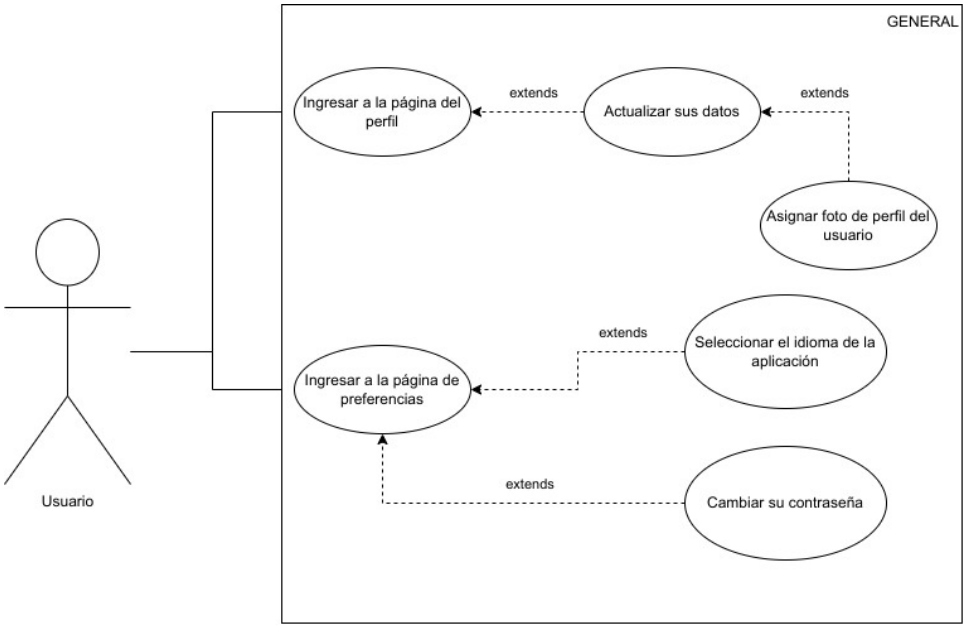
Figura 38.
Caso de uso 13 (CU-013): roles y permisos.



Fuente: propia.

CU-014: Acciones generales

Figura 39.
Caso de uso 14 (CU-014): acciones generales.



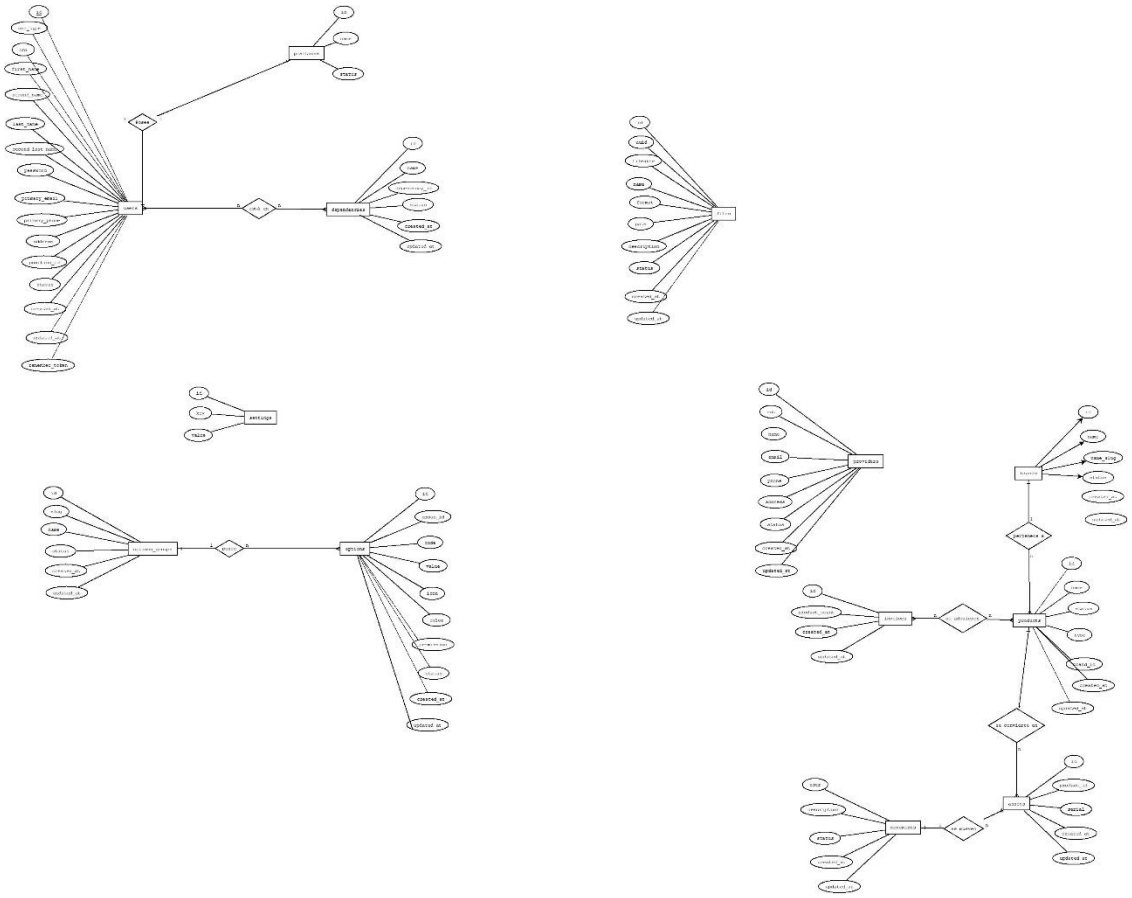
Fuente: propia.

Diagrama de base de datos

Modelo entidad relación

A partir de la información recolectada, se crea un modelo entidad relación, que servirá de fuente para generar el modelo relacional físico de base de datos. Se construye usando el software Dia, un software libre del proyecto GNOME.

Figura 40.
Modelo Entidad Relación.

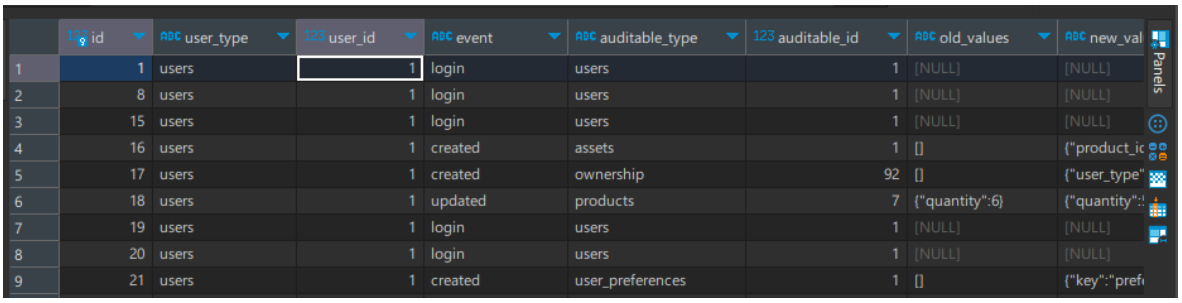


Fuente: elaboración propia.

Cabe destacar que el modelo hace uso de dos atributos específicos en algunas de las entidades obtenidas, específicamente user en la entidad movements, y fileable en la tabla files. Estos son atributos que son base para relaciones polimórficas. Las relaciones polimórficas son un tipo de relación especial usado en ciertos ORM (Object-Relational-Mapping) que permiten asociar dos campos en una tabla, donde uno indica la tabla foránea y el otro campo indica la llave del registro de la tabla foránea. Permite la reutilización de una misma tabla para diferentes implementaciones.

Un claro ejemplo de esto es el uso de la dependencia de auditorías a nivel de aplicación (owenit/laravel-auditing), y la librería de permisos y roles (spatie/laravel-permission). Ambos hacen uso extensivo de este tipo de relaciones para permitir que los registros de otras tablas sean referenciados fácilmente, sin tener que implementar diferentes llaves foráneas.

Figura 41.
Captura de tabla de audits en Dbeaver.



	id	user_type	user_id	event	auditable_type	auditable_id	old_values	new_val
1	1	users	1	login	users	1	[NULL]	[NULL]
2	8	users	1	login	users	1	[NULL]	[NULL]
3	15	users	1	login	users	1	[NULL]	[NULL]
4	16	users	1	created	assets	1	[]	{"product_ic
5	17	users	1	created	ownership	92	[]	{"user_type"
6	18	users	1	updated	products	7	{"quantity":6}	{"quantity":
7	19	users	1	login	users	1	[NULL]	[NULL]
8	20	users	1	login	users	1	[NULL]	[NULL]
9	21	users	1	created	user_preferences	1	[]	{"key":"pref

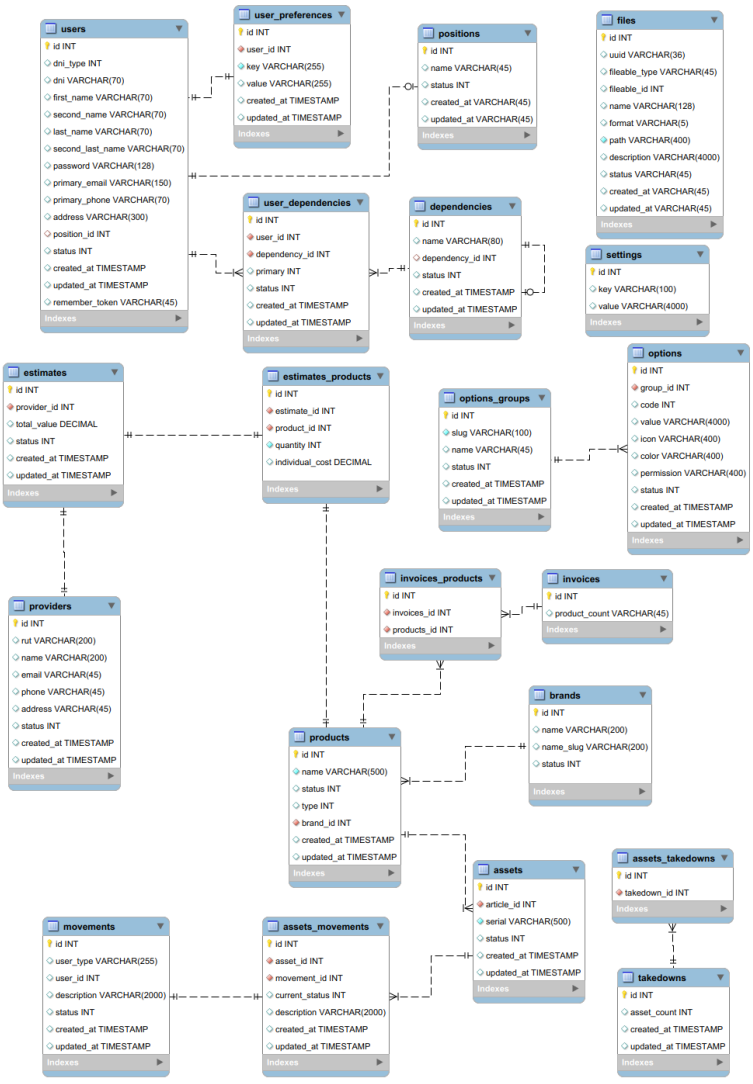
Fuente: propia.

En la figura anterior, la tabla de auditoria muestra los cambios realizados por el user_type users, con el user_id 1, que modifica registros de la tabla users, assets o user_preferences.

Modelo relacional

A partir del modelo entidad relación generado, se construye con la herramienta de modelado de base de datos que posee MySQL Workbench el diagrama relacional del sistema de información.

Figura 42.
Modelo relacional de base de datos.



Fuente: propia

Haciendo uso de Laravel, es posible recrear este modelo relacional mediante las migraciones de Eloquent. Esto permite que la creación de las tablas, relaciones, claves primarias y otros elementos de la base de datos sean agnósticas a la implementación de la base de datos. Si se hace uso de Docker, por defecto se configurará que se use PostgresSQL.

Diccionario de datos

Una vez implementado el modelo relacional, es posible generar un diccionario de datos que describa los campos y sus limitaciones. Algunas de las entidades mostradas en este diccionario de datos hacen parte de las dependencias usadas por el proyecto, como lo son:

- Audits
- Model_has_permissions
- Model_has_roles
- Migrations
- Role_has_permissions
- Notifications

A partir de la realización de la migración de Laravel, se obtiene lo siguiente:

Tabla 3.
Diccionario de datos.

Campo	Tipo de dato	Tamaño/precisión	Puede ser nulo	Relacionado a
		assets		
id	int8	64	No	
product_id	int4	32	No	products.id
serial	varchar	500	No	
user_type	varchar	255	Si	
user_id	int8	64	Si	
description	varchar	2000	Si	
plate	varchar	255	No	
invoice_id	int4	32	No	invoices.id
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		assets_movements		
id	int8	64	No	

movement_id	int4	32	No	movements.id
asset_id	int4	32	No	assets.id
description	varchar	2000	Si	
current_status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		assets_takedowns		
id	int8	64	No	
takedown_id	int4	32	No	takedowns.id
asset_id	int4	32	No	assets.id
created_at	timestamp		Si	
updated_at	timestamp		Si	
		audits		
id	int8	64	No	
user_type	varchar	255	Si	
user_id	int8	64	Si	
event	varchar	255	No	
auditable_type	varchar	255	No	
auditable_id	int8	64	No	
old_values	text		Si	
new_values	text		Si	
url	text		Si	
ip_address	inet		Si	
user_agent	varchar	1023	Si	
tags	varchar	255	Si	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		brands		
id	int8	64	No	
name	varchar	200	No	
name_slug	varchar	200	No	
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		dependencies		
id	int8	64	No	
name	varchar	80	No	
dependency_id	int4	32	Si	dependencies.id
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		estimates		
id	int8	64	No	
total_value	float8	53	No	
status	int4	32	No	

created_at	timestamp		Si	
updated_at	timestamp		Si	
		estimates_details		
id	int8	64	No	
estimate_id	int4	32	No	estimates.id
provider_id	int4	32	No	providers.id
total_value	float8	53	No	
chosen	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		estimates_products		
id	int8	64	No	
estimate_id	int8	64	No	estimates.id
product_id	int8	64	No	products.id
quantity	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		failed_jobs		
id	int8	64	No	
uuid	varchar	255	No	
connection	text		No	
queue	text		No	
payload	text		No	
exception	text		No	
failed_at	timestamp		No	
		files		
id	int8	64	No	
uuid	uuid		No	
fileable_type	varchar	255	No	
fileable_id	int8	64	No	
name	varchar	128	No	
format	varchar	5	No	
path	varchar	400	No	
description	varchar	4000	Si	
requires_login	int2	16	No	
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		invoices		
id	int8	64	No	
product_count	int4	32	No	
value	float8	53	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		invoices_products		

id	int8	64	No	
invoice_id	int4	32	No	invoices.id
product_id	int4	32	No	products.id
created_at	timestamp		Si	
updated_at	timestamp		Si	
		migrations		
id	int4	32	No	
migration	varchar	255	No	
batch	int4	32	No	
		model_has_permissions		
permission_id	int8	64	No	
model_type	varchar	255	No	
model_id	int8	64	No	
role_id	int8	64	No	
model_type	varchar	255	No	
model_id	int8	64	No	
		movements		
id	int8	64	No	
user_type	varchar	255	No	
user_id	int8	64	No	
description	varchar	2000	Si	
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		notifications		
id	uuid		No	
type	varchar	255	No	
notifiable_type	varchar	255	No	
notifiable_id	int8	64	No	
data	text		No	
read_at	timestamp		Si	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		options		
id	int8	64	No	
group_id	int4	32	No	options_groups.id
code	int4	32	No	
name	varchar	4000	No	
translate_key	varchar	200	No	
value	varchar	4000	Si	
icon	varchar	400	Si	
color	varchar	400	Si	
permission	varchar	400	Si	
status	int4	32	No	
created_at	timestamp		Si	

updated_at	timestamp		Si	
		options_groups		
id	int8	64	No	
slug	varchar	255	No	
name	varchar	255	No	
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		ownership		
id	int8	64	No	
ownable_type	varchar	255	No	
ownable_id	int8	64	No	
user_type	varchar	255	No	
user_id	int8	64	No	
primary	int4	32	No	
type	int4	32	No	
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		permissions		
id	int8	64	No	
name	varchar	255	No	
legible_name	varchar	255	No	
status	int4	32	No	
guard_name	varchar	255	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		positions		
id	int8	64	No	
name	varchar	45	No	
name_slug	varchar	45	No	
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		products		
id	int8	64	No	
name	varchar	500	No	
name_slug	varchar	500	No	
type	int4	32	No	
brand_id	int4	32	No	brands.id
quantity	int4	32	No	
low_quantity_notify	int4	32	No	
last_notification	timestamp		Si	
status	int4	32	No	
created_at	timestamp		Si	

updated_at	timestamp		Si	
		providers		
id	int8	64	No	
rut	varchar	70	No	
name	varchar	200	No	
name_slug	varchar	200	No	
email	varchar	70	Si	
phone	varchar	45	Si	
address	varchar	70	Si	
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		role_has_permissions		
permission_id	int8	64	No	
role_id	int8	64	No	
		roles		
id	int8	64	No	
name	varchar	255	No	
legible_name	varchar	255	No	
status	int4	32	No	
guard_name	varchar	255	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		settings		
id	int8	64	No	
key	varchar	100	No	
value	varchar	4000	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		takedowns		
id	int8	64	No	
asset_count	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		user_dependencies		
id	int8	64	No	
user_id	int4	32	No	users.id
dependency_id	int4	32	No	dependencies.id
primary	int4	32	No	
status	int4	32	No	
created_at	timestamp		Si	
updated_at	timestamp		Si	
		user_preferences		
id	int8	64	No	
user_id	int4	32	No	users.id

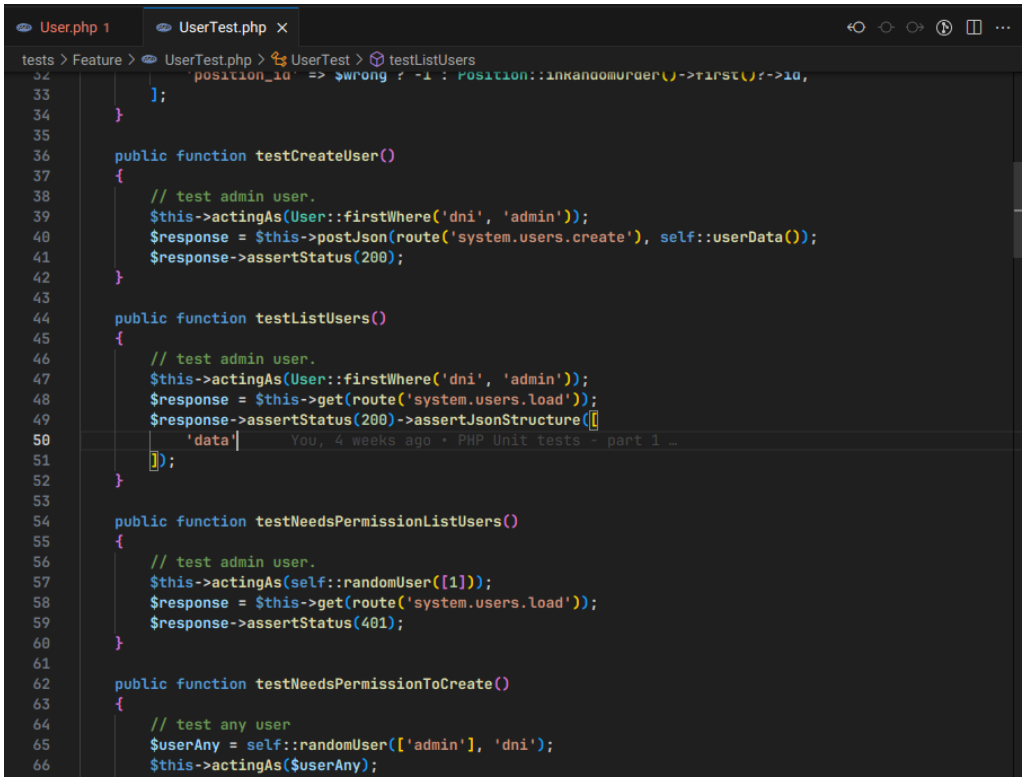
key	varchar	255	No
value	varchar	255	No
created_at	timestamp		Si
updated_at	timestamp		Si
	users		
id	int8	64	No
dni_type	int4	32	No
dni	varchar	70	No
first_name	varchar	70	No
second_name	varchar	70	Si
last_name	varchar	70	No
last_second_name	varchar	70	Si
address	varchar	300	Si
password	varchar	128	No
primary_email	varchar	150	No
primary_phone	varchar	70	No
position_id	int4	32	Si
status	int4	32	No
can_login	int4	32	No
remember_token	varchar	100	Si
created_at	timestamp		Si
updated_at	timestamp		Si

Fuente: propia.

Pruebas automatizadas

La implementación de pruebas automatizadas usando PHPUnit es uno de los elementos que más ha brindado resultados positivos en la creación del proyecto, ya que permitió encontrar diferentes errores en la aplicación que fueron subsecuentemente corregidos. La siguiente figura hace uso de PHPUnit en el módulo de usuarios para automatizar ciertas acciones a nivel del backend, con el fin de encontrar bugs antes que los usuarios finales y realizar las revisiones respectivas.

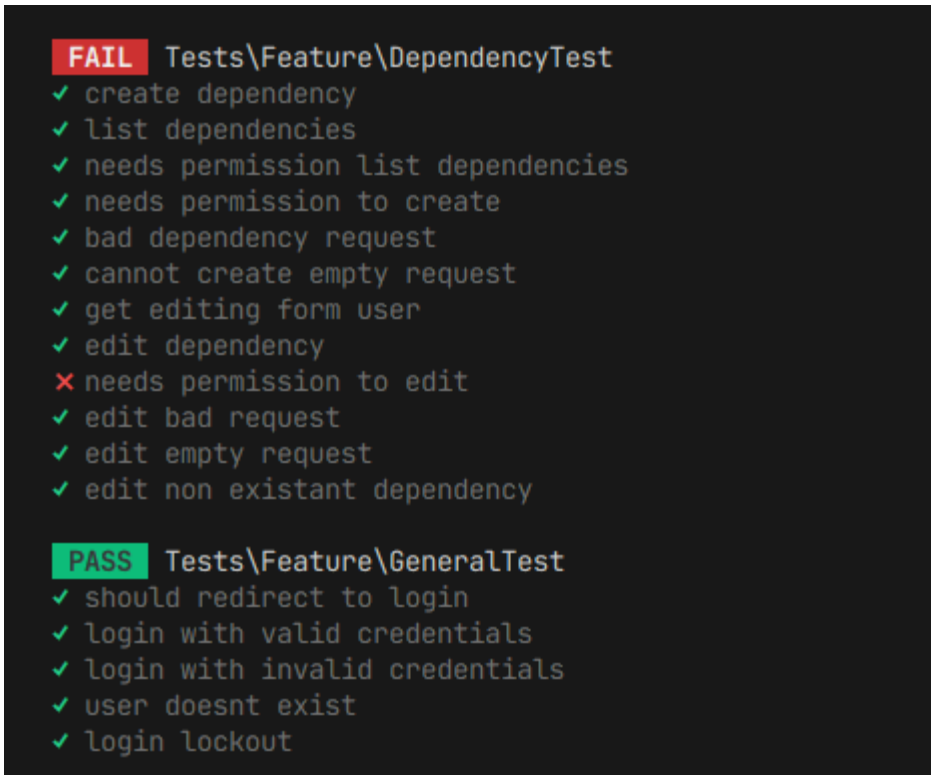
Figura 43.
Automatización de pruebas del módulo de usuarios.



Fuente: propia.

Al correr el comando “php artisan test”, se corren todas las pruebas automatizadas, en la que se observan diferentes resultados, mostrando, por ejemplo, resultados individuales del módulo de dependencias y de pruebas generales:

Figura 44.
Puebas automatizadas.

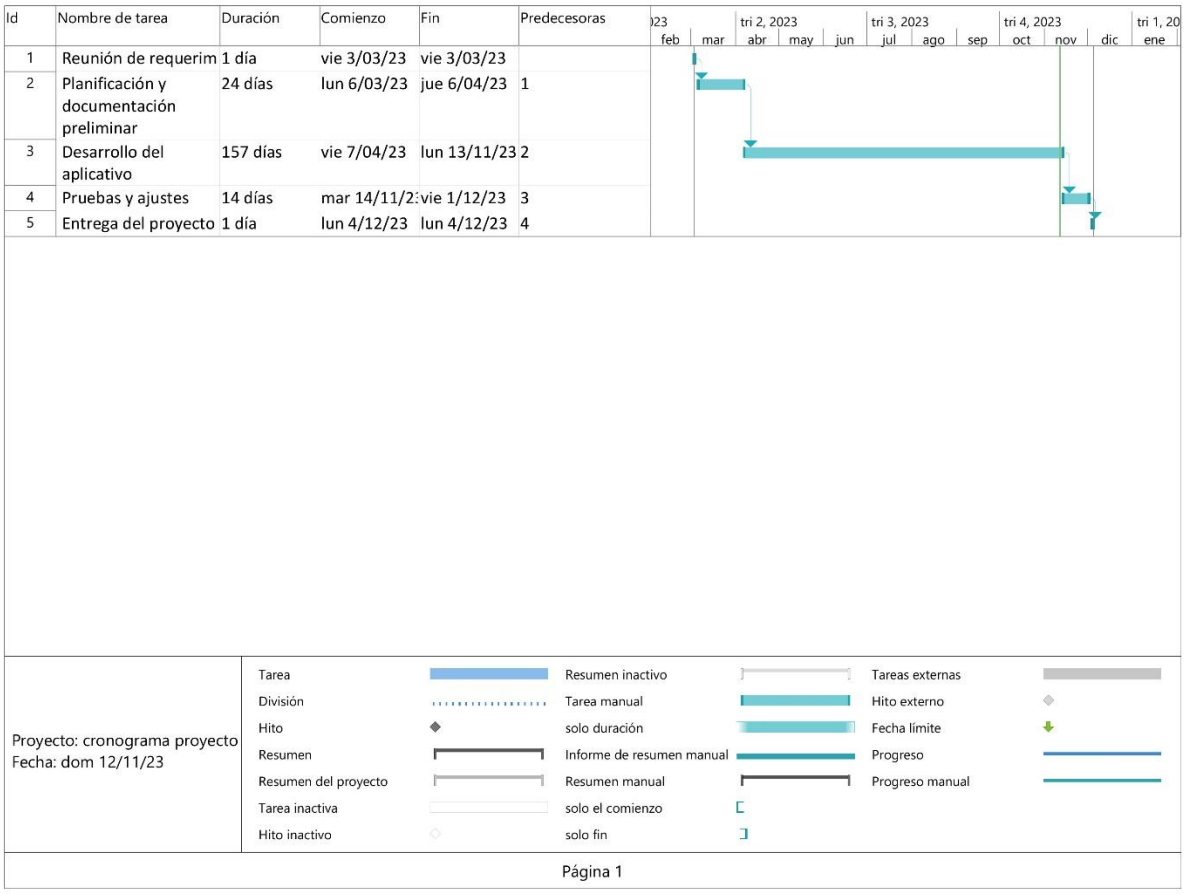


Fuente: propia.

Cronograma de actividades

Usando el software de Microsoft Project, se generó un cronograma de los tiempos usados para el proyecto realizado.

Figura 45.
Cronograma de actividades.



Fuente: propia.

Recursos requeridos

Los siguientes recursos fueron de insumo para la elaboración del software SIMCAI:

Tabla 4.
Recursos usados para la elaboración del software.

Recurso	Descripción	Presupuesto
Equipo humano		
Equipos y software	Portátil Lenovo IdeaPad S145 14API 8 GB RAM, Ryzen 5 3500u, 256 GB NVME M.2 SSD	Propio
Viajes y salidas de campo	Computador de mesa Intel i3-4170 3.7 GHz, 12 GB RAM, 500 GB SATA SSD Visitas al Colegio San Francisco de Asís	Propio

Fuente: propia.

Resultados

El resultado del proyecto realizado es un software web que mejora en gran medida el seguimiento y control del inventario en el Colegio San Francisco de Asís. La experiencia ha tenido un impacto altamente positivo, ya que sistematiza en mayor medida todo lo relacionado a los procesos de inventario, control de activos y cotizaciones.

El 17 de octubre del 2023 se hizo una presentación del proyecto al docente encargado de la parte de sistemas, y al encargado del almacén. El proyecto fue recibido de manera muy positiva, realizando únicamente sugerencias y observaciones en el proceso realizado, de las cuales se obtuvo:

- Las placas de inventario deben comenzar con el prefijo “cosfa”.
- Las placas de inventario deberían numerarse automáticamente, con la posibilidad de modificarlo si es el caso.
- Los salones deben de aparecer como opciones al realizar movimientos de activos.

Proyecciones

Se espera que con el software generado a partir de la realización de este proyecto, se mejoren radicalmente los procesos de inventario, movimiento de activos y cotizaciones, simplificando muchas de las tareas y labores existentes en la institución educativa.

Adicionalmente, el uso continuo del aplicativo permitirá identificar nuevas oportunidades de mejora que permitirán encaminar a la transformación digital, no solamente del Colegio San Francisco de Asís, sino también de otras instituciones que deseen sistematizar sus procedimientos relacionados a la gestión de inventario.

Conclusiones

En virtud de lo anteriormente expuesto en este documento, es posible decir que:

- Se observaron los procesos de inventario y cotización de la institución educativa, y se implementó un software que facilita y mejora estos procesos. El software implementado permite llevar un control eficiente de la información registrada.
- Se realizó una planificación y documentación que, al aplicar una metodología de desarrollo como RAD, permitió dar como resultado un programa de alta calidad que daba respuesta a las necesidades particulares del Colegio San Francisco de Asís.
- Ante lo anterior, se desarrolló una aplicación que incorporaba los frameworks de Laravel y Svelte.js que incorporaba los requerimientos solicitados por el área de inventario de la institución educativa.
- Las pruebas automatizadas permitieron atrapar errores y regresiones causadas por nuevas funcionalidades que se agregaban poco a poco en el aplicativo, lo cual garantizó un programa de alta calidad al usuario final.
- La implementación y prueba del software en la institución fue exitosa y fue de alto agrado, dado que el programa cumplía con las necesidades y expectativas solicitadas inicialmente.

Para finalizar, es importante reconocer que un software robusto y de alta calidad es alcanzado mediante la correcta planificación e investigación de las herramientas que se consideren deben de usarse. En el caso del desarrollo del aplicativo SIMCAI, los frameworks y metodologías usadas permitieron generar un aplicativo que fue de alto agrado para la institución

educativa y responden directamente a las necesidades del colegio, mediante la implementación de un proceso a nivel de software sencillo para el registro de cotizaciones, seguimiento de activos fijos y gestión de inventario, lo que permite aplicar los conocimientos de ingeniería de software adquiridos durante el transcurso de la carrera y continúa encaminando a la institución a un futuro digital seguro, eficiente y con altos estándares de calidad.

Recomendaciones técnicas

- Se recomienda configurar un servidor SFTP para separar los archivos usados en la aplicación de los que son cargados por los usuarios.
- Se recomienda configurar certificados SSL para la aplicación, o configurar esto a nivel de dominio.

Referencias bibliográficas

Ackerman, A. (s.f.). *Una guía sobre el simbolismo cromático*. Adobe: https://www.adobe.com/co/creativecloud/design/discover/color-meaning.html

Altube, R. (marzo de 2021). *Que es Laravel: características y ventajas*. OpenWebinars: https://openwebinars.net/blog/que-es-laravel-caracteristicas-y-ventajas/

Amazon Web Services. (s.f.). *¿Qué es Docker?* Amazon: https://aws.amazon.com/es/docker/

Angulo, D. N. (2021). *Implementación de un sistema web para la gestión de ventas e inventario de una empresa de calzado.* Universidad San Ignacio de Loyola. : https://hdl.handle.net/20.500.14005/11984

Atlassian. (s.f.). *¿Qué es Git?* Atlassian: https://www.atlassian.com/es/git/tutorials/what-is-git

Bautista, J. (5 de mayo de 2019). *UI Case Study: Roboto*. Medium: https://medium.com/@jaimie.bautista/ty-roboto-9244ce6343a5

Bergmann, S. (22 de febrero de 2021). *PHPUnit: Past, Present, Future*. The PHP Consulting Company: https://thephp.cc/presentations/phpunit-past-present-future

BindERP. (s.f.). *10 razones por las que hay que darle importancia al control de inventario*. BindERP: https://www.bind.com.mx/blog/control-de-inventarios/10-razones-por-las-que-hay-que-darle-importancia-al-control-de-inventario

Camacho, A. R. (2021). Importancia de la gestión de inventario en empresa de manufactura. *Boletín de Innovación, Logística y Operaciones, 2*(2), 37-42. https://doi.org/10.17981/bilo.02.02.2020.05

Cruz, J. (s.f.). *¿Qué es Svelte.js?* EDTeam: https://ed.team/blog/que-es-sveltejs-1

Deslauriers, J. (2004). Investigación cualitativa: guía práctica. Papiro. https://hdl.handle.net/11059/3365

Drumond, C. (s.f.). *Atlassian*. Qué es scrum y cómo empezar: https://www.atlassian.com/es/agile/scrum

Francia, G. (3 de febrero de 2021). *Significado del color verde en psicología*. Psicología-Online: https://www.psicologia-online.com/significado-del-color-verde-en-psicologia-5468.html

Fresneda, J. (17 de abril de 2023). *Pasos de gestión de inventarios y beneficios para la empresa*. INESEM Business School: https://www.inesem.es/revistadigital/gestion-empresarial/el-proceso-de-gestion-de-inventarios/

Intelequia. (28 de noviembre de 2020). *Ciclo de vida del software: todo lo que necesitas saber*. https://intelequia.com/es/blog/post/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber

Lora, W., Ahumada, D., & Muñoz, J. (18 de julio de 2019). *Implementación de un Sistema de Inventario para Bodegas a través de un Aplicativo Móvil Nativo en Android.* Universidad Nacional Abierta y a Distancia: https://repository.unad.edu.co/handle/10596/39168

Lotero, J. (15 de diciembre de 2022). *Sistema de Inventario Web para el Seguimiento de Recursos Físicos de la Secretaría de Movilidad de Envigado.* Universidad Nacional Abierta y a Distancia: https://repository.unad.edu.co/handle/10596/54354

Marin, L. (14 de mayo de 2021). *Por qué no debes utilizar Excel como Base de Datos*. Biuwer: https://biuwer.com/es/blog/por-que-no-debes-utilizar-excel-como-base-de-datos/

Martins, J. (19 de enero de 2024). *Asana*. ¿Qué es la metodología Kanban y cómo funciona?: https://asana.com/es/resources/what-is-kanban

Microsoft. (26 de marzo de 2023). *¿Qué es el desarrollo rápido de aplicaciones o RAD?* Power Apps: https://powerapps.microsoft.com/es-es/rapid-application-development-rad/

Ministerio de Educación. (29 de septiembre de 2020). *Guía No. 20 Organización y administración de bienes* . Ministerio de Educación Nacional de Colombia: https://www.mineducacion.gov.co/1780/articles-106040_archivo_pdf.pdf

Prieto, E. (31 de agosto de 2023). *¿Por qué es importante el control de inventarios?* Southern New Hampshire University: https://es.snhu.edu/noticias/por-que-es-importante-el-control-de-inventarios

Przybyla, D. (23 de julio de 2019). *La Psicología y el Significado del Color Azul*. ColorPsychology.org: https://www.colorpsychology.org/es/azul/

QuestionPro. (s.f.). *Investigación de campo*. https://www.questionpro.com/es/investigacion-de-campo.html

Ractem Racking System. (25 de febrero de 2019). *¿Qué es el método FIFO y LIFO en un almacén?* Ractem Racking System: https://www.ractem.es/blog/metodo-fifo-lifo-almacen#:~:text=FIFO%3A%20Primero%20En%20Entrar%2C%20Primero,que%20tien en%20fecha%20de%20caducidad

Rodríguez, J. (18 de julio de 2023). *Control de inventarios: definición, importancia y sistemas*. HubSpot: https://blog.hubspot.es/sales/que-es-control-de-inventarios

StarAgile. (03 de octubre de 2023). *Agile vs RAD*. https://staragile.com/blog/agile-vs-rad

The PHP Group. (s.f.). *History of PHP*. PHP: https://www.php.net/manual/en/history.php.php

Universidad Nacional Abierta y a Distancia. (28 de mayo de 2014). *Acuerdo No. 006 de mayo 28 de 2014.*

https://sgeneral.unad.edu.co/images/documentos/consejoAcademico/acuerdos/2014/COAC_ACUE_20140528_006.pdf

Universidad Nacional Abierta y a Distancia UNAD. (13 de diciembre de 2013). *Acuerdo 0029 del 13 de diciembre del 2013.* https://sgeneral.unad.edu.co/images/documentos/consejoSuperior/acuerdos/2013/COSU_ACUE_029_20131229.pdf

Villareal-Puga, J. C. (2022). La aplicación de entrevistas semiestructuradas en distintas modalidades durante el contexto de la pandemia. *Hallazgos21*, 54.

W3Schools. (s.f.). *Laravel - History*. W3Schools: https://www.w3schools.in/laravel/history#:~:text=Laravel%20%2D%20History&text=Laravel%20was%20developed%20and%20created,authentication%20and%20proper%20user%20authorization