



Universidad Internacional San Isidro Labrador

Escuela de Ingeniería de Sistemas

Desarrollo y propuesta de implementación del Sistema Integral de Gestión Granitos Naranjo (SIGG) para optimizar los procesos operativos y administrativos, gestionando proyectos, clientes, personal e inventario.

Estudiante:

Alejandro Umaña Pérez

Tutor:

Eric Corella Solís

Licenciatura en Ingeniería de Sistemas

ISL-07 Seminario de Graduación I

Segundo Cuatrimestre 2024

Índice

Introducción	4
Antecedentes.....	5
Antecedentes de la empresa.....	5
Antecedentes del problema.....	6
Descripción del problema	7
Justificación.....	8
Formulación del problema	9
Objetivos.....	10
Objetivo General.....	10
Objetivos Específicos.....	10
Delimitación del proyecto	11
Alcance.....	11
Limitaciones.....	11
Marco teórico	13
Introducción	13
¿Qué es un sistema informático?.....	13
Sistemas de información	13
Sistemas de gestión.....	14
Ciclo de vida de un sistema.....	15
Análisis de requisitos.....	16
Metodologías de desarrollo.....	17
Herramienta de desarrollo (IDE).....	19
Programación orientada a objetos (POO).....	20
¿Qué es una base de datos?	22
Diagramas de flujo	23
Desarrollo del proyecto.....	26
Requisitos del sistema	26
Requerimiento 1.....	26
Requerimiento 2.....	27

Requerimiento 3-1.....	28
Requerimiento 3-2.....	29
Requerimiento 4-1.....	30
Requerimiento 4-2.....	31
Requerimiento 5.....	32
Requerimiento 6.....	33
Requerimiento 7-1.....	34
Requerimiento 7-2.....	35
Requerimiento 8.....	35
Arquitectura del sistema.....	36
Base de datos (SQL Server).....	36
Backend (API´s)	36
Frontend (MVC con ASP.NET)	37
Diagrama de arquitectura	37
Diagramas UML.....	38
Diagramas Casos de uso.....	38
Diagrama de flujo.....	45
Diagrama de base de datos	53
Propuesta implementación	54
Conclusiones	56
Recomendaciones	57
Referencias	59
Anexos	61

Introducción

En el competitivo mercado actual, la capacidad de una empresa para gestionar de manera eficiente sus recursos, proceso y relaciones comerciales es crucial para su éxito y crecimiento. Granitos Naranjo, como empresa líder en el sector de la fabricación e instalación de granitos, reconoce la importancia de adoptar prácticas de gestión modernas y eficaces que le permitan mantener su posición competitiva y seguir satisfaciendo las necesidades de sus clientes de manera óptima.

El desarrollo y propuesta de implementación de un Sistema Integral de Gestión (en adelante SIG) representa un paso estratégico para Granitos Naranjo en su búsqueda por mejorar la eficiencia operativa, la calidad del servicio y la toma de decisiones basada en datos. Este sistema proporcionará una plataforma unificada y centralizada que integrará todos los aspectos clave de la gestión empresarial, desde la planificación y ejecución de proyectos hasta la gestión de recursos humanos, inventarios, relaciones con clientes y proveedores, garantizando una operación fluida y optimizada en toda la organización.

Antecedentes

Antecedentes de la empresa

La empresa, Granitos Naranjo, ha estado operando en el mercado durante los últimos 15 años. Su historia comienza gracias al único proveedor de granito y cuarzo en toda la zona sur del país donde trabajaba Daniel Hidalgo Naranjo. Sin embargo, la gestión inicial era desordenada, lo que llevó a que Daniel decidiera renunciar y empezar su propio negocio en respuesta a la demanda de trabajos que recibía.

Inicialmente, la operación era modesta: Daniel compró una placa de granito y comenzó a trabajar en la cochera de su casa con herramientas básicas como una esmeriladora, algunos bloques de madera en el suelo y una pulidora. Para el transporte, alquilaba un taxi de carga para mover los sobres de granito y cuarzo.

Durante los primeros 11 años, la empresa no tenía un procedimiento para el seguimiento de los proyectos cotizados. Fue solo hace cuatro años en el 2020, cuando se comenzó a implementar un seguimiento semanal de las cotizaciones, contactando a los clientes para dar continuidad a los proyectos.

Con la llegada de Karol Cruz Sánchez una nueva administradora de empresas el 2 de febrero del 2020, Daniel comenzó a profesionalizar la operación. Bajo su gestión, se introdujeron varias mejoras significativas en la estructura y procesos de la empresa:

- Procedimiento para el seguimiento de cotizaciones: Se estableció un proceso semanal para revisar el historial de las cotizaciones y contactar a los clientes, mejorando así la tasa de conversión de cotizaciones a proyectos.
- Uso de Excel para la gestión administrativa: Todos los aspectos administrativos, como inventarios, cuentas por cobrar, cuentas por pagar, comisiones, especificaciones de cada proyecto y gestión de planillas, se comenzaron a gestionar en Excel.
- Mejoras en inventarios: La empresa ahora maneja un inventario en consignación, lo que significa que solo paga por el material que utiliza. Actualmente, el inventario tiene aproximadamente 100 o más placas de granito y cuarzo.

Estas mejoras han transformado significativamente la operación de Granitos Naranjo, permitiendo una mayor eficiencia y capacidad de respuesta a la demanda del mercado. La implementación de procedimientos de gestión han sido clave para este crecimiento, marcando un antes y un después en la historia de la empresa.

Antecedentes del problema

Granitos Naranjo, a pesar de su larga trayectoria y consolidación en el mercado, enfrenta serios desafíos en la gestión y operación de sus procesos debido a la falta de un sistema integral que permita una administración eficiente y coordinada de sus actividades. Desde su fundación hace 15 años, la empresa ha experimentado un crecimiento notable, pero su evolución ha sido limitada por prácticas administrativas desarticuladas y dependientes de herramientas manuales como Excel.

La empresa comenzó como un pequeño negocio familiar, con recursos limitados y un enfoque en la satisfacción del cliente a través de productos de calidad. Los primeros años fueron cruciales para establecer su reputación en el mercado local, pero la falta de una infraestructura administrativa adecuada empezó a mostrar sus efectos negativos a medida que la demanda y las operaciones crecían.

Inicialmente, la empresa operaba de manera rudimentaria, con un equipo básico y sin procesos formales de seguimiento y control. Con el tiempo Granitos Naranjo ha incorporado mejoras significativas como el uso de Excel para su gestión administrativa.

A pesar de estos avances, la falta de un sistema automatizado e integral sigue siendo una barrea importante para su eficiencia operativa. La carencia de un software de gestión impide que Granitos Naranjo optimice sus procesos y se adapte rápidamente a las demandas del mercado, limitando su capacidad de competir eficazmente en un entorno empresarial cada vez más dinámico y exigente.

Descripción del problema

Granitos Naranjo enfrenta múltiples desafíos en la gestión de sus procesos operativos y administrativos debido a la falta de un sistema integral que abarque todas las áreas críticas de la empresa. Actualmente, los procesos de gestión de proyectos, inventario, clientes, proveedores y personal se manejan de manera desarticulada y manual, lo que conlleva a una serie de problemas que afectan la eficiencia y la transparencia de la operación y limitan la toma de decisiones.

Algunos de los problemas que se presentan son los siguientes:

- Falta de integración de datos: la información está dispersa en diferentes sistemas y documentos, lo que dificulta la consolidación de datos y la obtención de una visión completa y actualizada de la operación.
- Ineficiencia operativa: Los procesos manuales consumen una cantidad considerable de tiempo y recursos, reduciendo la productividad.
- Errores Humanos: La manipulación manual de datos y la falta de automatización incrementan la probabilidad de errores, afectando la precisión de la información y las operaciones de la empresa.
- Transparencia limitada: La ausencia de un sistema integral que centralice la información limita la transparencia y dificulta el seguimiento y control de procesos internos.
- Toma de decisiones: La falta de acceso a datos precisos y en tiempo real impide que la dirección tome decisiones informadas y oportunas.

Justificación

El desarrollo y la propuesta de implementación de un Sistema Integral de Gestión (SIG) permitirá a Granitos Naranjo superar estos desafíos al proporcionar una plataforma centralizada y automatizada que optimice los procesos operativos y administrativos. Un SIG bien diseñado y desarrollado mejorara la eficiencia, reducirá errores y costos, aumentara la transparencia y facilitara una toma de decisiones más informada y estratégica.

Serrano (2023) explica que "Un sistema integrado de gestión permite a las organizaciones hacer un seguimiento continuo y en tiempo real de los procesos con acceso inmediato a toda la información de todas las áreas de la empresa, además de mejorar los indicadores de rendimiento."

Un SIG no solo optimiza las operaciones actuales, sino que también prepara a la empresa para enfrentar futuros desafíos con una base sólida de gestión y control, promoviendo un crecimiento sostenible y competitivo en el mercado.

Formulación del problema

¿Cómo influye el desarrollo y propuesta de implementación del Sistema Integral de Gestión Granitos Naranjo (SIGG) en la optimización de los procesos operativos y administrativos?

Objetivos

Objetivo General

- Desarrollar un Sistema Integral de Gestión que optimice los procesos operativos y administrativos de Granitos Naranjo, integrando la gestión de proyectos, inventarios, clientes, proveedores y personal, con el fin de incrementar la eficiencia, promover la transparencia y fortalecer la toma de decisiones en la empresa.

Objetivos Específicos

- Analizar y documentar las necesidades específicas de la empresa para definir los requisitos funcionales que orientarán el desarrollo del SIG.
- Crear un plan detallado que contemple la estructura del sistema, el modelo de datos, la comunicación entre módulos y un diseño de interfaz intuitivo y amigable para el usuario.
- Desarrollar el sistema de acuerdo con las especificaciones técnicas y funcionales, garantizando la calidad y funcionalidad esperadas.
- Elaborar un plan de implementación que contemple pruebas, ajustes y despliegue del sistema en un entorno de producción para asegurar su correcto funcionamiento.

Delimitación del proyecto

Alcance

El proyecto de desarrollo del Sistema Integral de Gestión para Granitos Naranjo (SIGG) se enfocará en la implementación de una solución tecnológica robusta que abarque los aspectos más importantes de la gestión empresarial. A continuación, detallan los principales componentes que formaran parte del alcance del proyecto:

- **Gestión de Proyectos:** El SIGG incluirá un módulo dedicado a la gestión integral de proyectos como la creación, modificación y eliminación, también la creación de cotizaciones y seguimiento de estas.
- **Gestión de inventarios:** Se desarrollará un sistema de gestión de inventarios que proporcionará una visión detallada y actualizada del inventario de granito disponible, incluyendo información sobre tipos de granito, cantidades, ubicaciones en almacén y movimientos de salida.
- **Gestión de clientes y proveedores:** El SIGG contará con un módulo dedicado a la gestión de clientes y proveedores que permitiría mantener un registro completo de la información de contacto.
- **Gestión de personal:** Se desarrollará un sistema de gestión de personal que abarcará aspectos como la administración de datos personales, registros laborales, gestión de planillas para un mayor control.
- **Sistema de Planificación:** El SIGG integrará un sistema de planificación y control de tareas, que permitirá organizar, priorizar y monitorear las tareas de manera eficiente, garantizando el cumplimiento de plazos y una óptima utilización de los recursos disponibles.

Limitaciones

Presupuesto:

- **Costos de licencias y herramientas:** El uso de tecnologías y herramientas de desarrollo debe estar alineado con el presupuesto asignado, lo que puede implicar la selección de opciones de software y herramientas más económicas o de código abierto en lugar de soluciones comerciales costosas.

Tiempo:

- Cronograma estricto de 4 meses: El proyecto debe completarse dentro de un plazo de 4 meses, con hitos claros y fechas de entrega específicas para cada módulo. Cualquier retraso en una fase del proyecto afectara las siguientes, lo que puede poner en riesgo la finalización del sistema dentro del tiempo previsto.
- Fases de desarrollo y entrega: Se seguirán fases de desarrollo iterativas con entregables funcionales en intervalos cortos (por ejemplo, cada mes). Esto garantizará que cada módulo esté listo y probado antes de pasar al siguiente, pero también significa que el tiempo para cada fase es extremadamente limitado y debe cumplirse estrictamente.
- Pruebas y validaciones: El tiempo dedicado a las pruebas y validaciones debe ser suficiente para garantizar la calidad del sistema, pero ajustado al cronograma general. No se podrá extender el periodo de pruebas indefinidamente.
- Capacitación y documentación: La capacitación del personal y la documentación del sistema deben realizarse dentro del tiempo asignado, lo cual podría limitar la profundidad y extensión de los materiales de capacitación.

Marco teórico

Introducción

El desarrollo de un sistema de gestión integral permite a las organizaciones centralizar y optimizar sus procesos, mejorando la eficiencia y la toma de decisiones. Para comprender plenamente estos beneficios, es esencial definir algunas características y términos clave utilizados en el desarrollo de software.

¿Qué es un sistema informático?

Un sistema informático puede definirse como un conjunto de partes interrelacionadas.

Un sistema informático típico emplea un ordenador que usa dispositivos programables para capturar, almacenar y procesar datos. Dicho ordenador, junto con la persona que lo maneja y los periféricos que lo envuelven, resultan de por sí un ejemplo de un sistema informático. (Raya Cabrera & Raya González, 2015, p. 8).

Sistemas de información

Debido al gran volumen de datos que generan las empresas, es crucial transformar esos datos en información valiosa que pueda apoyar la toma de decisiones y optimizar procesos, como mejorar las operaciones o el servicio al cliente. Para lograr esto, se emplean sistemas de información, los cuales organizan, analizan y presentan los datos de manera que resulten útiles y relevantes para los objetivos empresariales.

Por esto es importante primero tener claro que es un dato y (Cobarsi-Morales, 2011, p.15) afirma: "Los datos son hechos objetivos sobre acontecimientos, que no tienen un significado inherente y que no han sido seleccionados."

También es importante definir que es la información: en términos simples, es el conjunto de datos que adquieren forma y relevancia para quienes la utilizan. Por ejemplo, un número puede representar el teléfono de un cliente, pero como dato, este número es simplemente una secuencia de dígitos sin significado, sin embargo, cuando este se relaciona con la información de contacto del cliente, adquiere relevancia y utilidad, convirtiéndose en información valiosa para la empresa.

La idea de Sistema de Información proviene del concepto más amplio de Sistema, definido como una estructura que gobierna y una actividad que transforma. Un

Sistema de Información en una empresa incluye una estructura de decisión, una actividad operativa basada en Normas de Gestión y un conjunto de datos, procesando solicitudes y generando resultados. (Navarro Huerga & Fernández Otero, 2014)

Un sistema de información en una empresa puede ser de varios tipos por lo tanto se va a definir lo que es un sistema de gestión y cuales son algunos de sus tipos de sistemas que existen.

Sistemas de gestión

Según Evaluando ERP (2021), "Un sistema de gestión es un conjunto de herramientas que ayudan a organizar, planificar y controlar los recursos y procesos de una empresa para alcanzar sus objetivos de manera eficiente."

Debido a esto en las empresas es muy común que usen un sistema de gestión o varios dependiendo del tamaño de la empresa, en el caso de Granitos Naranjo la idea es mezclar en un sistema personalizado con ciertas funciones de los tipos de sistemas de gestión que existen para poder optimizar sus procesos operativos.

Algunos de los tipos de sistema de gestión que existen y que nos interesan para esta investigación son los siguientes:

- **Sistemas de planificación de recursos empresariales de ahora en adelante (ERP) por sus siglas en inglés:** Son sistemas que dan soporte completo e integral la gestión empresarial facilitando el flujo de datos entre procesos.
- **Sistemas de gestión de la relación con el cliente de ahora en adelante (CRM) por sus siglas en inglés:** Son sistemas que apoyan con la gestión comercial y la relación con los clientes normalmente los ERP incluyen en sus sistemas un módulo que puede realizar esta tarea, sin embargo, ciertas empresas desarrollan o contratan software especializado para mejorar la relación con sus clientes.
- **Sistemas de gestión de almacenes de ahora en adelante (SGA):** Son sistemas que se relacionan con las actividades logísticas con la gestión de almacenes o inventarios se igual manera los ERP tienen su modulo para realizar la tarea, pero algunas empresas optan por sistemas más especializados para conseguir mayor productividad.

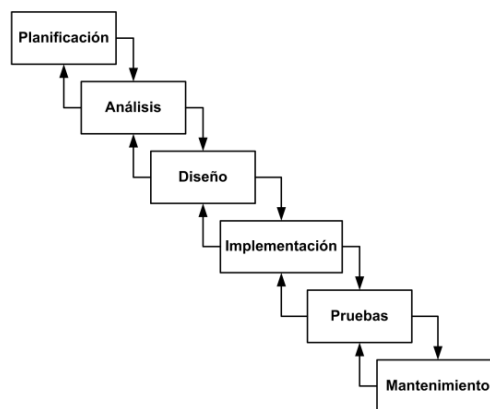
Ciclo de vida de un sistema

En el desarrollo de software, es fundamental seguir una serie de fases o etapas conocidas como el ciclo de vida del sistema. Estas etapas aseguran que el sistema sea escalable, mantenible y evolucione de manera efectiva a lo largo del tiempo. Al seguir un ciclo de vida bien estructurado, se facilita la gestión del proyecto, se optimiza el desarrollo del software y se asegura que el sistema pueda adaptarse a futuras necesidades y cambios.

El más común y utilizado en el mundo es el modelo en cascada que define una serie de pasos con la idea de hacer todo bien desde el principio como se muestra en la Figura 1 las fases de este modelo en cascada.

Figura 1

Ciclo de vida: Modelo en “cascada”



Fuente: Flanagan (p.24)

Este modelo es un poco inflexible por lo cual se ha optado por realizar adaptaciones como es el caso de SCRUM.

Scrum es un marco de trabajo que se ha utilizado desde los años 90 para gestionar el desarrollo de productos complejos. No se trata de un proceso o técnica específica para la creación de productos, sino de un marco en el que se pueden aplicar diversas técnicas y procesos. Este marco permite evaluar la efectividad de las prácticas de gestión y desarrollo de productos, lo que facilita la mejora continua. (Schwaber & Sutherland, 2013).

Análisis de requisitos

El análisis de requisitos es primordial para obtener un buen desarrollo del software debido a que es lo que define las características del sistema en forma y orden.

Por lo tanto, lo primero que se debe hacer es, entender que es son los requisitos, y según lo mencionado por García Notario, D. (2015): “el conjunto de requisitos de un sistema es una explicación en detalle de lo que debe ser implementado en función de las necesidades del cliente, incluyendo la forma de actuación de este.” (p. 42)

Un adecuado levantamiento de requisitos del sistema puede tener un impacto significativo en el éxito del desarrollo de este. Por ello, es crucial entrevistar a todas las personas involucradas, desde el personal operativo hasta los administradores de la empresa o negocio.

Se puede definir 2 tipos de requisitos que siempre deben existen en el desarrollo del software que son los requisitos funcionales y los no funcionales que se les va a llamar “técnicos”.

- **Requisitos funcionales:** Son los requisitos que dictan exactamente como se debe comportar el sistema desde la experiencia y conocimiento de un usuario.
- **Requisitos técnicos:** Son los requisitos que indican las características técnicas de una funcionalidad en concreto como por ejemplo como debe ser el login de un sistema, el usuario no tiene este conocimiento, por lo tanto, es muy normal que un arquitecto de software realice el requerimiento indicando exactamente como debe ser la funcionalidad y la seguridad de la pantalla en mención.

Todo esto se logra gracias a las historias de usuario de ahora en adelante “HU”, pero que es una HU.

Las historias de usuario son descripciones concisas y claras de una funcionalidad o requisito, vistas desde la perspectiva de quienes solicitan esa característica, que generalmente son los clientes. (Canosa Ferreiro, 2024)

Las HU normalmente llevan una estructura muy simple que es la siguiente:

- Yo como = “Rol”

- Quiero = “Objetivo”
- Para = “Motivo”

Para el proyecto se va a seguir la estructura de requerimiento, de la siguiente manera como se muestra en la Tabla 1

Tabla 1

Estructura de requerimiento que se va a trabajar en el proyecto

ID	RF-001
Nombre	Inicio de sesión
Descripción	El sistema debe permitir a los usuarios iniciar sesión ingresando su nombre de usuario y contraseña.
Entradas	Nombre de usuario o correo electrónico Contraseña (enmascarada con asteriscos o puntos)
Proceso	El usuario ingresa sus credenciales. El sistema verifica las credenciales contra la base de datos. Si son correctas, el usuario accede; si no, muestra un error.
Salidas	Acceso al sistema o mensaje de error.
Precondiciones	N/A
Postcondiciones	N/A
Criterios de aceptación	El campo contraseña debe estar enmascarado. El sistema debe mostrar un error en caso de credenciales incorrectas en caso contrario permite acceder al sistema.
Dependencias	Requiere la existencia de una base de datos de usuarios y un servicio de autenticación seguro.

Fuente: (Elaboración propia)

Metodologías de desarrollo

Una metodología de desarrollo según lo mencionado por Maida & Pacienza (2015) “Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información.” (p.13)

Por lo tanto, es como una guía estructurada para poder seguir y tener el control del desarrollo del sistema. Existen las metodologías tradicionales y las ágiles.

- **Metodología tradicional:** Estas se caracterizan por ser regidas en el desarrollo, el propósito es alcanzar un software más eficiente. También se caracteriza por definir y establecer cada uno de los requisitos desde el inicio del proyecto. Normalmente son pocos flexibles y no permiten cambios.
- **Metodologías ágiles:** Estas metodologías son totalmente lo contrario a las tradiciones, estas permiten adaptarse a las condiciones del proyecto, permitiendo cambios sobre el desarrollo del software, consiguiendo flexibilidad inmediata en caso de requerir amoldar el proyecto y su desarrollo a las circunstancias del entorno.

En la Figura 2 se muestra una comparativa entre las metodologías tradicionales y ágiles.

Figura 2

Comparativa entre metodologías tradicionales y metodologías ágiles.

Metodologías ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos
Poca documentación	Documentación exhaustiva
Muchos ciclos de entrega	Pocos ciclos de entrega

Fuente: Maida & Pacienza (2015)

Herramienta de desarrollo (IDE)

Para llevar a cabo el desarrollo de software de manera efectiva, es esencial utilizar una herramienta adecuada para este propósito, lo que destaca la importancia de comprender qué es un Entorno de Desarrollo Integrado (IDE).

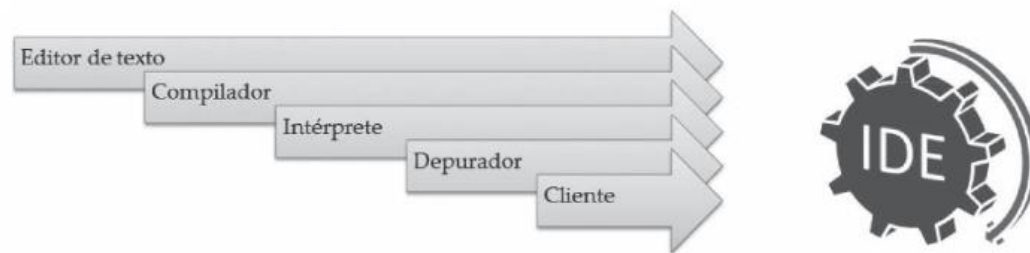
Un entorno de desarrollo integrado o IDE (Integrated Development Environment) es un programa informático que tiene el objetivo de asistir al programador en la tarea de diseñar y codificar un software mediante la inclusión de múltiples herramientas destinadas para dicha tarea. (Casado Iglesias, 2015, p. 46)

De acuerdo con lo anterior se debe buscar un IDE según el lenguaje de programación que se vaya a escoger, debido los IDE están desarrollados para ciertos tipos de lenguajes en específicos.

La mayoría de los IDE están compuestos por unos componentes básicos como se puede observar en la Figura 3

Figura 3

Componentes básicos de un IDE



Fuente: Casado Iglesias (2015)

Para este proyecto se va a utilizar el lenguaje de programación C# el cual es muy flexible y potente a nivel de desempeño, por lo cual se va a utilizar el IDE de Visual Studio Community 2022.

Programación orientada a objetos (POO)

El lenguaje de programación C# es un lenguaje orientado a objetos. Por lo tanto, es fundamental comprender qué es la programación orientada a objetos y en qué consiste.

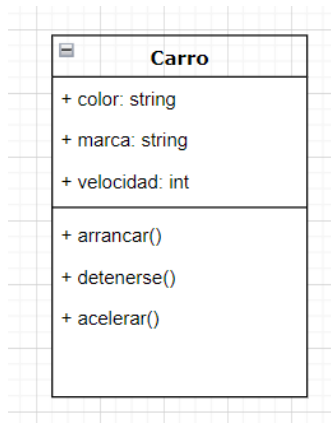
La POO (Object Oriented Programming, OOP, por sus siglas en inglés) es un paradigma en la programación para computadora en la que la forma de programar es de una manera muy parecida a como expresaríamos o describiríamos las cosas o acciones que suceden en el mundo real. La POO busca implementar los algoritmos de una manera diferente a otros paradigmas de programación. Para escribir los algoritmos y programas se deben crear (bajo los conceptos de clases) modificadores de acceso, métodos y variables de clase, utilizando instancias de clase conocidas como objetos con sus propiedades y comportamientos. La POO se basa en objetos o instancias de clase creadas a partir de clases. (Aristizábal Martínez & Quiceno Metaute, 2023, p. 35).

- **Objeto:** Es una instancia de una clase que representa una entidad en el mundo real. Por ejemplo, carro.
Atributo: Es la propiedad o característica de un objeto que define un estado. Por ejemplo, del objeto “carro” los atributos pueden ser color, marca, velocidad entre otros.
- **Método:** Método es la acción o comportamiento que puede realizar dicho objeto, por ejemplo, para el objeto “carro” puede tener los siguientes métodos: arrancar, acelerar, detenerse.
- **Clase:** Una clase especifica los atributos y métodos que los objetos de ese tipo van a tener, por así decirlo una clase es una plantilla del objeto el cual contiene tanto los atributos como los métodos por ejemplo la clase “carro” va a contener los atributos antes mencionados y de igual forma los métodos.

En la figura 4 se muestra el ejemplo de una clase con sus atributos y métodos.

Figura 4

Clase “carro”



Fuente: Elaboración propia

- **Abstracción:** Es el proceso de simplificar la complejidad mediante la reducción de la información y solo exponer los detalles esenciales.
- **Encapsulamiento:** Es el mecanismo que restringe el acceso a hacia los atributos y métodos, permitiendo la modificación solo a través de los métodos públicos definidos. La idea del encapsulamiento es mantener la integridad de los datos.
- **Herencia:** Es la capacidad de que una clase herede sus atributos y métodos de otra clase, esto permite la creación de nuevas clases basadas en clases existentes, facilitando la reutilización y extensión del código. Por ejemplo, una clase llamada “vehículo” puede ser heredada o ser la clase base de “carro” y “moto”.
- **Polimorfismo:** Este concepto es algo complicado de comprender así que se va a explicar de la forma más simple posible. Imagina que tienes un juguete, este juguete tiene un botón en la espalda que hace una acción (método) dependiendo del tipo de juguete que es, ahora imagina que este juego puede hacer acciones diferentes dependiendo del tipo que sea:
 - Un superhéroe, puede levantar la mano con presionar el botón
 - Un dinosaurio, puede abrir la boca con presionar el botón
 - Un robot, puede encender luces en los ojos con presionar el botón

El botón siempre es el mismo, pero dependiendo del tipo que sea, hace cosas diferentes, justamente esto es lo que se le conoce como polimorfismo, donde se tiene un botón (método) que puede hacer cosas diferentes dependiendo de que tipo de objeto sea (superhéroe, dinosaurio, robot) dando así un comportamiento especial.

¿Qué es una base de datos?

Para entender que es una base de datos, hay que recordar cómo se almacenaba la información antes de que estas existieran, y claro cómo te puedes imaginar antes la información se almacenaba en papel en alguna parte de la empresa u organización por lo cual cada vez que se necesitaba realizar un trámite primero tenían que buscar la información lo cual era algo lento y tedioso, si la información o el papel no estaba ya sé porque se perdió o se organizó mal en ese momento, no se encontraba y tenían que volver a generar un documento nuevamente esto era sumamente complicado por lo cual con el aumento de los volúmenes de datos ya no era sostenible mantener la información de esta manera, se necesitaba un lugar centralizado y lo más importante que fuera digital, aquí es donde nacen las bases de datos que vienen a resolver este problema.

Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos se integran con una mínima cantidad de duplicidad. De este modo, la base de datos no pertenece a un solo departamento sino que se comparte por toda la organización. Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción es lo que se denomina metadatos, se almacena en el diccionario de datos o catálogo y es lo que permite que exista independencia de datos lógica-física. (Marqués, 2011, p. 10)

Para gestionar una base de datos, es necesario utilizar un motor de base de datos. En términos simples y para este proyecto, un motor de base de datos es la herramienta que nos permitirá crear, modificar, consultar, eliminar y, en general, administrar nuestras tablas y registros.

En este proyecto, se utilizará una herramienta desarrollada por Microsoft llamada “Microsoft SQL Server” para gestionar la base de datos, la cual estará basada en el lenguaje SQL.

SQL (Structured Query Language) es el lenguaje estándar utilizado en las bases de datos relacionales, el cual permite definir y ejecutar varios tipos de operaciones sobre los datos mediante su enfoque declarativo (Valencia Cabrera, 2021).

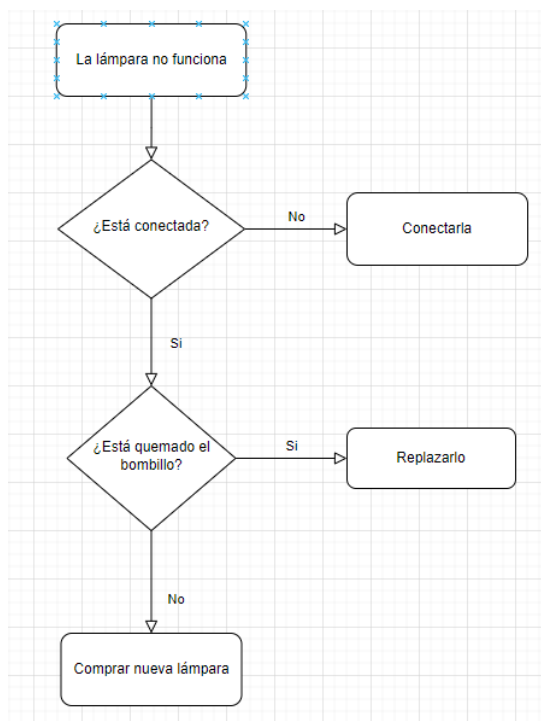
Diagramas de flujo

Un diagrama de flujo es una representación visual que describe la secuencia de pasos en un proceso. Su principal ventaja es que ilustra el orden de las operaciones, los participantes y las responsabilidades asociadas con cada paso, funcionando como una representación simbólica de un procedimiento administrativo. Estos diagramas son útiles para desglosar procesos en cualquier tipo de actividad, ya sea en empresas industriales o de servicios, y en sus respectivas áreas organizativas. Actualmente, son considerados herramientas esenciales en la creación y gestión de métodos o sistemas dentro de las empresas. (Manene, 2011)

En la figura 5 se muestra un diagrama simple sobre el funcionamiento de una lámpara.

Figura 5

Diagrama de flujo sobre el funcionamiento de una lámpara



Fuente: Elaboración propia.

Para este proyecto, se crearán diagramas de flujo para los módulos principales utilizando la herramienta Draw.io. Esta herramienta es eficaz para la visualización y diseño de procesos debido a su interfaz intuitiva y sus amplias funcionalidades.

Las herramientas más conocidas para realizar diagramas de flujo son:

- Draw.io
- Lucidchart
- Microsoft Visio

Estas herramientas facilitan la elaboración de diagramas de flujo detallados y claros, que son fundamentales para documentar y optimizar procesos en cualquier proyecto.

Para recapitular los sistemas informáticos son conjuntos integrados que combinan hardware, software y procedimientos para gestionar y procesar datos de manera eficiente en las organizaciones. Estos sistemas no solo incluyen la tecnología física y digital, sino también los métodos y procesos necesarios para su funcionamiento efectivo. Dentro de estos sistemas, los sistemas de información juegan un papel crucial al reunir tecnología, personas y procesos para recolectar, almacenar y distribuir información valiosa, facilitando así la toma de decisiones y la coordinación de actividades. Los sistemas de gestión, como los ERP y CRM, están diseñados para optimizar la planificación y control de recursos dentro de una organización. El ciclo de vida de un sistema, que abarca desde la planificación y el análisis hasta el diseño, la implementación y el mantenimiento, es esencial para garantizar que el sistema funcione correctamente y cumpla con las expectativas. El análisis de requisitos ayuda a definir claramente lo que necesitan los usuarios, mientras que las metodologías de desarrollo ya sean tradicionales o ágiles, guían el proceso de creación del sistema. Las herramientas de desarrollo, incluidos los Entornos de Desarrollo Integrados (IDE) y los lenguajes de programación, son fundamentales para la construcción y mantenimiento del software. La programación orientada a objetos (POO) ofrece ventajas en términos de modularidad y reutilización del código, facilitando el desarrollo de sistemas complejos. Las bases de datos permiten organizar y gestionar grandes volúmenes de información, siendo cruciales para el funcionamiento eficiente de las aplicaciones. Además, los diagramas de flujo ayudan a representar gráficamente los pasos de un proceso, haciendo más fácil comprender y comunicar procedimientos complejos. En conjunto, estos elementos y conceptos son esenciales para diseñar, desarrollar y gestionar sistemas informáticos y de información de manera efectiva en cualquier entorno organizacional.

Desarrollo del proyecto

Para dar inicio al desarrollo del proyecto lo primero que ocupamos conocer son los requerimientos del sistema SIGG, para esto se tuvo reuniones con la usuaria experta para definir detalle a detalle cada requerimiento y funcionalidad del sistema que se requiere. A continuación, se van a detalle los requerimientos que se establecieron durante las reuniones.

Requisitos del sistema

En la tabla 2 se puede observar el requerimiento 1 de inicio de sesión

Tabla 2

Requerimiento 1

ID	RF-001
Nombre	Inicio de sesión
Descripción	El sistema debe permitir a los usuarios iniciar sesión ingresando su correo electrónico y contraseña.
Entradas	Correo electrónico Contraseña (enmascarada con asteriscos)
Proceso	El usuario ingresa sus credenciales. El sistema verifica las credenciales contra la base de datos. Si son correctas, el usuario accede; si no, muestra un error.
Salidas	Acceso al sistema o mensaje de error.
Precondiciones	El usuario debe estar registrado en el sistema.
Postcondiciones	El usuario tiene acceso a las funcionalidades del sistema. El usuario no tiene acceso al sistema si falla las credenciales.
Criterios de aceptación	El campo contraseña debe estar enmascarado. El sistema debe mostrar un error en caso de credenciales incorrectas en caso contrario permite acceder al sistema.
Dependencias	Tabla Usuarios y estar registrado en la base de datos.

En la tabla 3 se puede observar el requerimiento 2 de creación de proyectos

Tabla 3

Requerimiento 2

ID	RF-002
Nombre	Creación de proyectos
Descripción	El sistema debe permitir la creación de nuevos proyectos
Entradas	Nombre del proyecto Descripción Nombre empresa Recursos asignados (Materiales) Precio (En colones y dólares) Tipo cambio (digitado por el usuario) Adelanto o Pago (Con fechas) Abonos (Con fechas) Medio de pago tanto para adelanto o abonos. Monto Original Saldo Estado
Proceso	El usuario ingresa los datos requeridos para crear un nuevo proyecto. El sistema guarda los datos en la base de datos.
Salidas	Confirmación de que el proyecto ha sido creado con éxito.
Precondiciones	El usuario debe llenar los campos requeridos para la creación del proyecto.
Postcondiciones	El proyecto debe estar registrado en el sistema.
Criterios de aceptación	El sistema debe permitir ingresar todos los campos requeridos. La información debe almacenarse correctamente en la base de datos.
Dependencias	Base de datos de proyectos.

En la tabla 4 se puede observar el requerimiento 3-1 de creación de cotización

Tabla 4

Requerimiento 3-1

ID	RF-003-1
Nombre	Creación de Cotización
Descripción	El sistema debe permitir crear una cotización, permitiendo seleccionar materiales, descripciones, cantidad, y mostrando los totales en colones y dólares.
Entradas	Datos del cliente (Nombre, dirección, teléfono) Numero de cotización Fecha Área de sobre o producto Material Descripción Protector contra salpicadura Regrueso Cantidad Tipo de cambio Total colones Total dólares Observaciones
Proceso	El usuario ingreso los datos requeridos de la cotización. El sistema guarda los cambios.
Salidas	Vista previa de la cotización luego de guardar.
Precondiciones	El cliente debe estar guardado previamente en el sistema. El proyecto debe estar guardado previamente en el sistema.
Postcondiciones	El sistema mostrara la cotización del proyecto.
Criterios de aceptación	El sistema debe permitir consultar, crear, modificar o eliminar la cotización. El sistema debe guardar toda la información en la base de datos.
Dependencias	Base de datos de proyectos, clientes, cotizaciones y productos.

En la tabla 5 se puede observar el requerimiento 3-2 del Seguimiento de Cotización

Tabla 5

Requerimiento 3-2

ID	RF-003-2
Nombre	Seguimiento de Cotización
Descripción	El sistema debe permitir el seguimiento de la cotización y actualización del estado de la cotización.
Entradas	Numero de cotización Estado de cotización (Aprobado, Seguimiento, Rechazado). Fecha probable de fabricación Fecha probable de instalación
Proceso	El usuario actualiza el seguimiento de la cotización. Se debe mostrar en pantalla una semana antes de la fecha de fabricación. Para dar seguimiento. El sistema guarda los cambios.
Salidas	Vista de las fechas de fabricación para poder darle seguimiento se debe mostrar una semana antes para el debido seguimiento respectivo.
Precondiciones	La cotización debe estar creada previamente.
Postcondiciones	La cotización se muestra en orden de fecha para su seguimiento.
Criterios de aceptación	El sistema debe permitir registrar el seguimiento de la cotización. Se debe poder actualizar el estado de la cotización. Se debe guardar los cambios en base de datos
Dependencias	Base de datos de cotizaciones. Base de datos de seguimiento.

En la tabla 6 se puede observar el requerimiento 4-1 de Registro de Materiales

Tabla 6

Requerimiento 4-1

ID	RF-004-1
Nombre	Registro de Materiales
Descripción	El sistema debe permitir el registro de los materiales en el inventario.
Entradas	Fecha de ingreso Nombre del material Tipo de granito (Granito, Cuarzo, Mármol, Porcelánico) Atado Boleta Largo Ancho Cantidad de placas Ubicación (Tienda, Taller, Uvita) Precio por metro cuadrado Precio por metro Lineal Proveedor
Proceso	El usuario registra un nuevo material ingresando todos los detalles necesarios. El sistema guarda la información del material en la base de datos.
Salidas	Confirmación del registro del material.
Precondiciones	N/A.
Postcondiciones	El material queda registrado en el inventario.
Criterios de aceptación	El sistema debe permitir registrar el material con todos sus atributos. Los datos deben almacenarse correctamente en la base de datos.
Dependencias	Base de datos de inventario, y proveedores.

En la tabla 7 se puede observar el requerimiento 4-2 de creación de cotización

Tabla 7

Requerimiento 4-2

ID	RF-004-2
Nombre	Control de Salidas del Inventario
Descripción	El sistema debe permitir registrar las salidas de material del inventario.
Entradas	Fecha Material (Seleccionar el material del inventario) Largo Ancho Boleta Atado Cantidad Nombre Proyecto (Seleccionar proyecto) Precios en colones y dólares Proforma Observaciones
Proceso	El usuario registra los movimientos de material. El sistema actualiza las cantidades en el inventario.
Salidas	Visualización en pantalla de movimientos de inventario.
Precondiciones	El material debe estar registrado previamente en el inventario.
Postcondiciones	El inventario queda actualizado con los movimientos recientes.
Criterios de aceptación	El sistema debe permitir registrar salidas de inventario. Los movimientos deben reflejarse inmediatamente en las cantidades del inventario.
Dependencias	Base de datos de inventario.

En la tabla 8 se puede observar el requerimiento 5 de Registro de Clientes

Tabla 8

Requerimiento 5

ID	RF-005
Nombre	Registro de Clientes
Descripción	El sistema debe permitir registrar la información de contacto y proyectos asociados a los clientes.
Entradas	Nombre del cliente. Información de contacto (dirección, teléfono, correo, cedula). Proyectos asociados
Proceso	El usuario ingresa los datos de un cliente nuevo. El sistema almacena la información del cliente en la base de datos.
Salidas	Confirmación del registro del cliente.
Precondiciones	N/A.
Postcondiciones	El cliente queda registrado en el sistema y asociado a sus proyectos.
Criterios de aceptación	El sistema debe permitir ingresar todos los campos necesarios del cliente. Los datos deben almacenarse correctamente en la base de datos.
Dependencias	Base de datos de clientes.

En la tabla 9 se puede observar el requerimiento 6 de Registro de Proveedores

Tabla 9

Requerimiento 6

ID	RF-006
Nombre	Registro de Proveedores
Descripción	El sistema debe permitir registrar a los proveedores, asociándolos con los materiales que proveen.
Entradas	Nombre del proveedor. Información de contacto (teléfono, dirección, correo, cédulas, cuentas) Materiales suministrados Condición de compra (Contado o consignación)
Proceso	El usuario ingresa los datos del proveedor. El sistema almacena los datos y los relaciona con los materiales.
Salidas	Confirmación del registro del proveedor.
Precondiciones	N/A.
Postcondiciones	El proveedor queda registrado en el sistema y relacionado con los materiales que provee.
Criterios de aceptación	El sistema debe permitir ingresar toda la información relevante del proveedor. Los materiales deben asociarse correctamente con el proveedor.
Dependencias	Base de datos de proveedores y materiales.

En la tabla 10 se puede observar el requerimiento 7-1 de Registro de Colaboradores

Tabla 10

Requerimiento 7-1

ID	RF-007-1
Nombre	Registro de Colaboradores
Descripción	El sistema debe permitir registrar a los empleados, incluyendo información personal y laboral.
Entradas	Nombre Información personal (cedulas, dirección, teléfono, cuentas, estado civil) Fecha Ingreso Registro laboral (Fecha, hora de entrada y salida) Licencias Posición Salario * Hora Tipo de contrato (Fijo o Por hora) Pagos (Boleta de pago, Fecha de pago, Horas trabajadas, Monto del pago) Estado (Activo/ Inactivo)
Proceso	El usuario registra los datos de un nuevo empleado y su información laboral. El sistema debe registrar las horas laboradas de cada trabajador por día, El sistema guarda los datos en la base de datos.
Salidas	Confirmación del registro del empleado.
Precondiciones	N/A.
Postcondiciones	El empleado queda registrado en el sistema.
Criterios de aceptación	El sistema debe permitir ingresar toda la información requerida. Los datos deben almacenarse correctamente en la base de datos.
Dependencias	Base de datos de personal.

En la tabla 11 se puede observar el requerimiento 7-2 de Control de garantías sociales

Tabla 11

Requerimiento 7-2

ID	RF-007-2
Nombre	Control de garantías sociales
Descripción	El sistema debe permitir gestionar las garantías sociales del personal, incluyendo vacaciones, aguinaldo.
Entradas	Vacaciones 1 día * Mes Aguinaldo Salario total de los meses trabajados dividido entre 12
Proceso	El sistema debe permitir gestionar y guardar las garantías sociales para cada colaborador.
Salidas	Mensaje de éxito luego de la gestión.
Precondiciones	El colaborador debe estar registrado en el sistema.
Postcondiciones	El registro se modifica en caso de ser necesario.
Criterios de aceptación	El sistema debe permitir la gestión de las garantías sociales crear, modificar, o eliminar. El sistema debe guardar la información en la base de datos.
Dependencias	Base de datos de personal.

En la tabla 12 se puede observar el requerimiento 8 de Registro de Tareas

Tabla 12

Requerimiento 8

ID	RF-008
Nombre	Registro de tareas
Descripción	El sistema debe permitir registrar y organizar las tareas de la semana.
Entradas	Fecha Tarea (Descripción)
Proceso	El usuario registra las tareas. El sistema guarda la información en la base de datos.
Salidas	Lista de tareas organizadas por fecha.

Precondiciones	N/A.
Postcondiciones	Las tareas quedan registradas o actualizadas.
Criterios de aceptación	El sistema debe permitir registrar las tareas. Las tareas deben mostrarse en la semana que corresponda.
Dependencias	Base de datos de tareas.

Arquitectura del sistema

Luego de tener los requerimientos del sistema SIGG ya establecidos y definidos, se requiere diseñar la arquitectura de proyecto que se va a implementar para este sistema.

En este caso, se ha decidido optar por una arquitectura robusta, como la N-Tier, para la distribución física de las capas. Se utilizará un lenguaje de programación y un framework reconocidos a nivel mundial, específicamente C# junto con ASP.NET Core.

Las capas del proyecto se han organizado de la siguiente manera:

Base de datos (SQL Server)

SQL Server se estableció como base de datos para el proyecto SIGG, por su rendimiento robusto, seguridad avanzada y escalabilidad, estas características permiten manejar eficazmente los datos críticos de cada modulo del sistema, garantizando la integridad de los datos como también la optimización de las consultas SQL. También por su integración con ASP.NET que facilita la administración fluida de las operaciones en cada área del sistema.

Backend (API's)

El backend del proyecto SIGG se definió que se va a utilizar ASP.NET Core debido a su capacidad para crear APIs eficientes y escalables, y su integración fluida con SQL Server mediante System.Data.SqlClient para ejecutar procedimientos almacenados, lo que asegura un acceso directo y controlado a la base de datos. La inyección de dependencias permite una gestión modular y flexible de los servicios y conexiones, garantizando una arquitectura limpia y mantenible. Además, la autenticación y autorización se gestionan mediante JWT, proporcionando seguridad robusta para los usuarios. Los controllers delegan la lógica de

negocio a servicios inyectados, manteniendo una separación clara de responsabilidades, mientras que toda la interacción con la base de datos se maneja exclusivamente a través de procedimientos almacenados.

Frontend (MVC con ASP.NET)

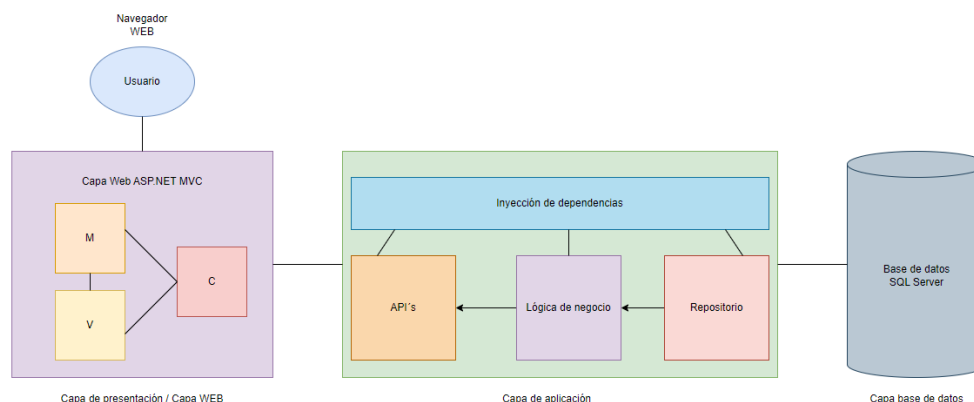
El frontend del proyecto SIGG se definió utilizar el patrón MVC en ASP.NET para estructurar de manera eficiente la presentación de la aplicación. ASP.NET MVC permite gestionar las vistas y la lógica de presentación de forma organizada, mientras que las Views se encargan de renderizar el contenido dinámico en el navegador. Para una interfaz visual atractiva y responsive, se utilizará CSS junto con Bootstrap, y JavaScript/jQuery proporcionará la interactividad necesaria en el cliente. Los controllers gestionan las solicitudes de los usuarios y devuelven los datos a las vistas sin depender de inyección de dependencias. Las views representan las páginas que el usuario ve, mientras que los models transportan los datos entre controladores y vistas. Además, los scripts permiten interacciones dinámicas, como llamadas AJAX hacia la API del backend.

Diagrama de arquitectura

El diagrama de Arquitectura del proyecto es el siguiente como se puede observar en la figura 6.

Figura 6

Arquitectura del proyecto



Fuente: Elaboración propia.

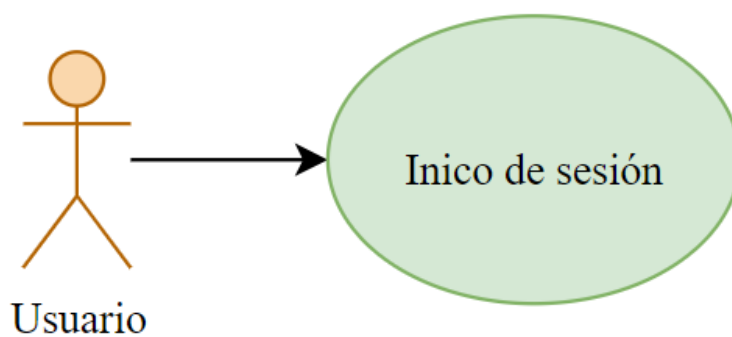
Diagramas UML

Diagramas Casos de uso

En la siguiente figura 7 se puede observar el caso de uso para el inicio de sesión

Figura 7

Caso de uso para el inicio de sesión

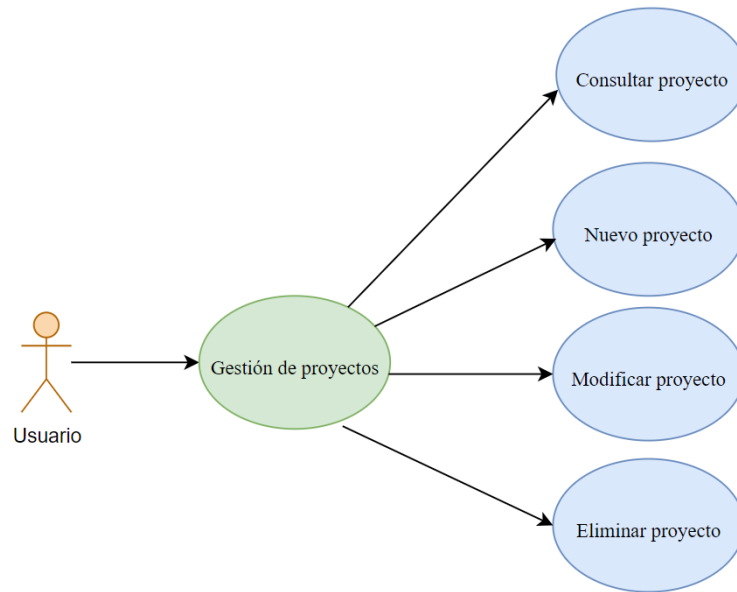


Fuente: Elaboración propia.

En la siguiente figura 8 se puede observar el caso de uso para la gestión de proyectos

Figura 8

Gestión de proyectos

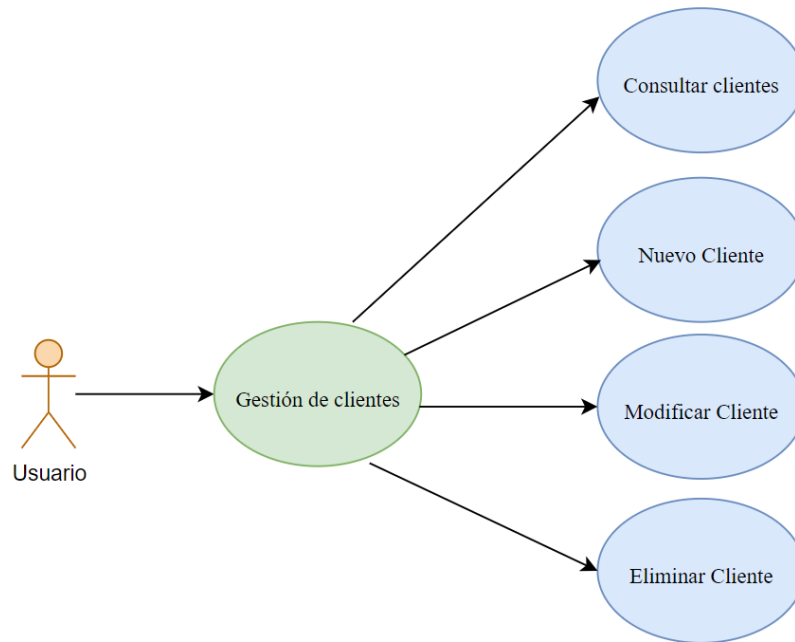


Fuente: Elaboración propia.

En la siguiente figura 9 se puede observar el caso de uso para la gestión de clientes

Figura 9

Gestión de clientes

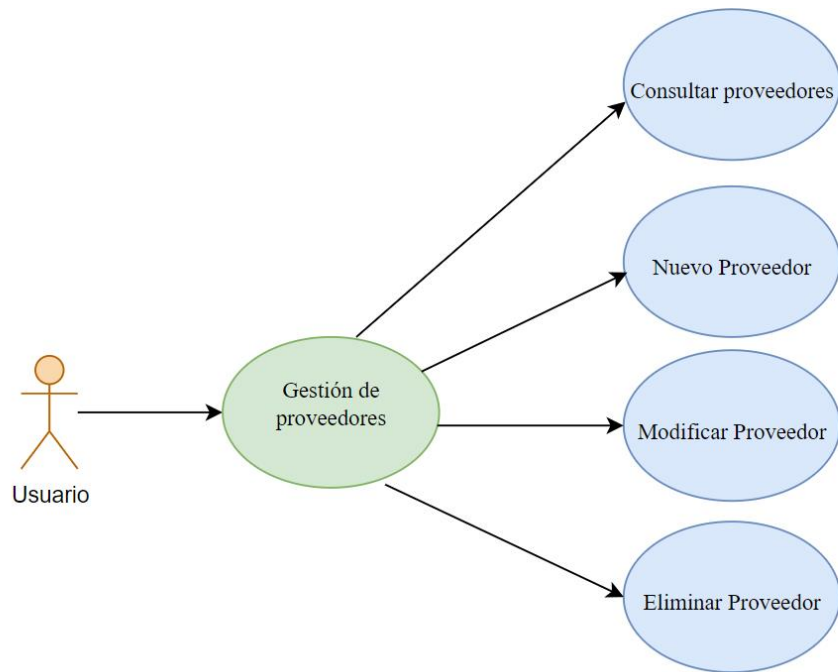


Fuente: Elaboración propia.

En la siguiente figura 10 se puede observar el caso de uso para la gestión de proveedores.

Figura 10

Gestión de proveedores

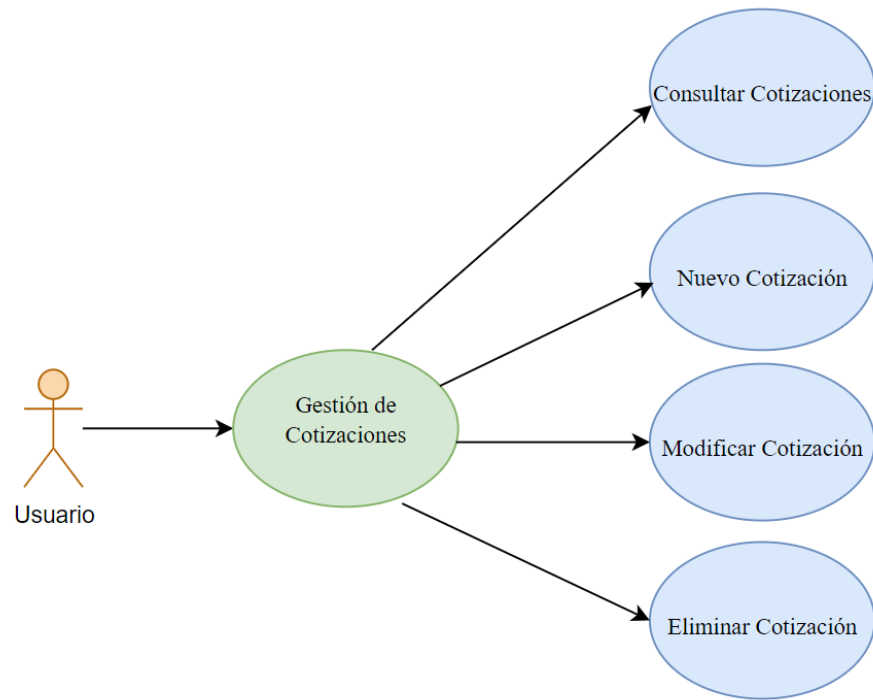


Fuente: Elaboración propia.

En la siguiente figura 11 se puede observar el caso de uso para el inicio de sesión

Figura 11

Gestión de cotizaciones

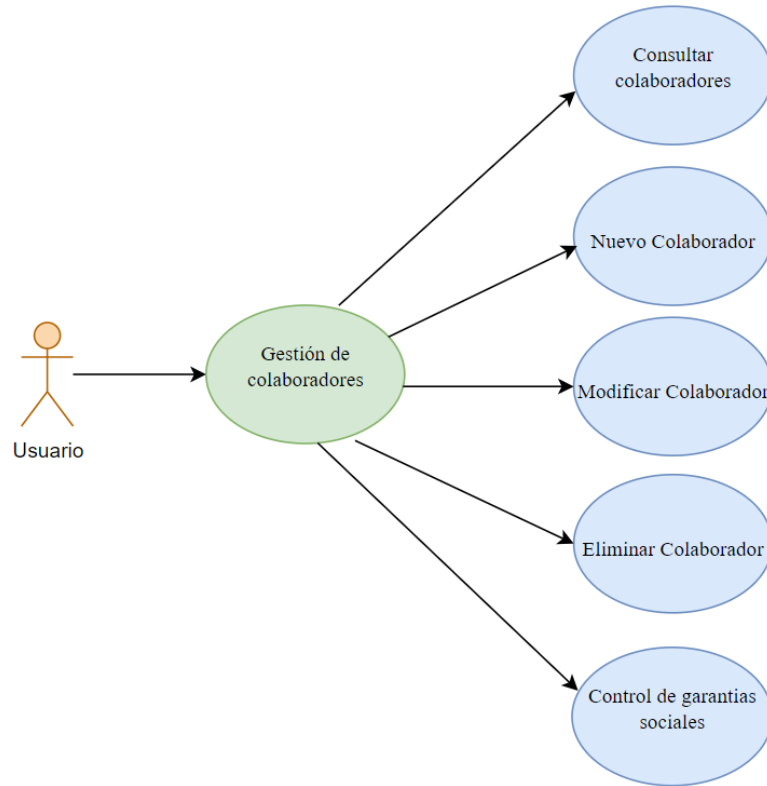


Fuente: Elaboración propia.

En la siguiente figura 12 se puede observar el caso de uso para la gestión de colaboradores.

Figura 12

Gestión de colaboradores

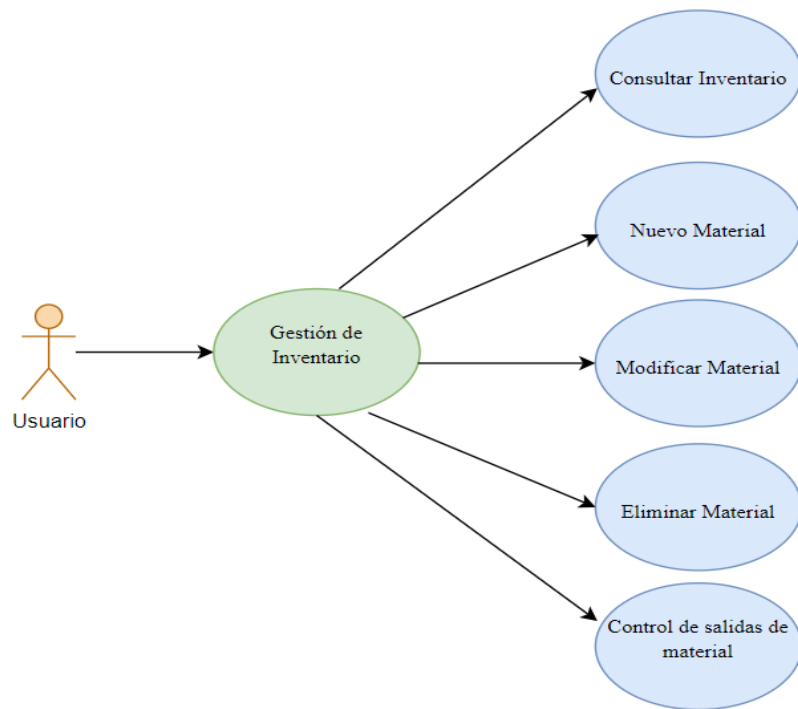


Fuente: Elaboración propia.

En la siguiente figura 13 se puede observar el caso de uso para la gestión de inventario.

Figura 13

Gestión de inventario

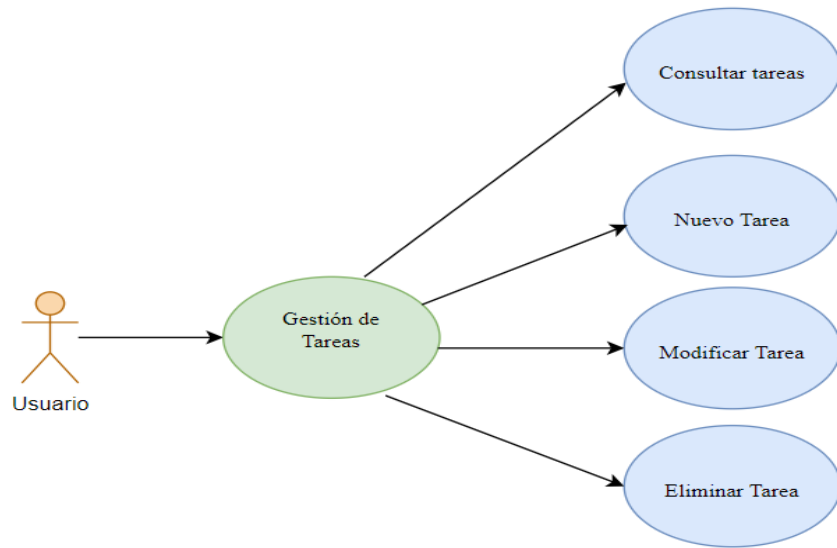


Fuente: Elaboración propia.

En la siguiente figura 14 se puede observar el caso de uso para la gestión de tareas.

Figura 14

Gestión de tareas



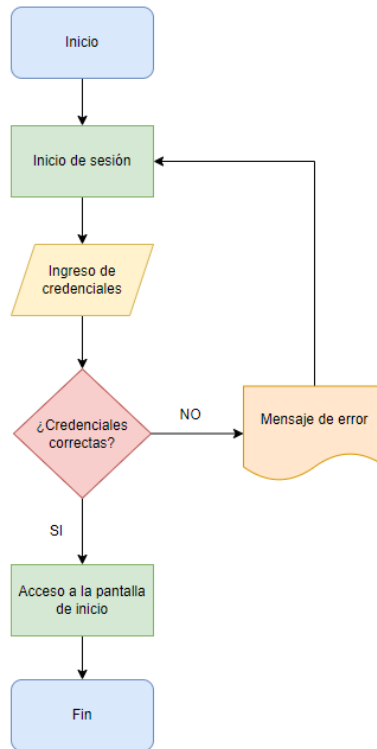
Fuente: Elaboración propia.

Diagrama de flujo

En la siguiente figura 15 se puede observar el diagrama de flujo para el inicio de sesión

Figura 15

Inicio de sesión

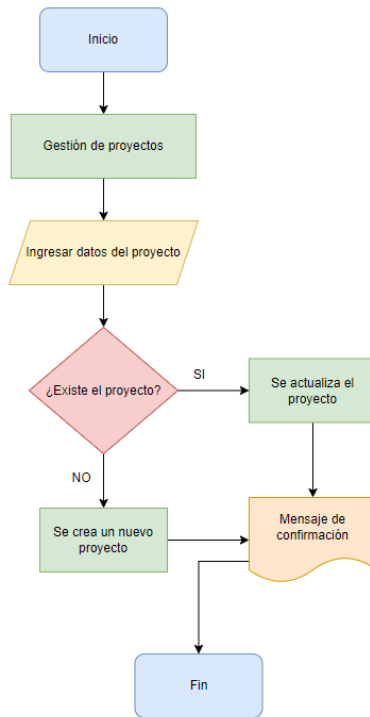


Fuente: Elaboración propia.

En la siguiente figura 16 se puede observar el diagrama de flujo para la gestión de proyectos.

Figura 16

Gestión de proyectos

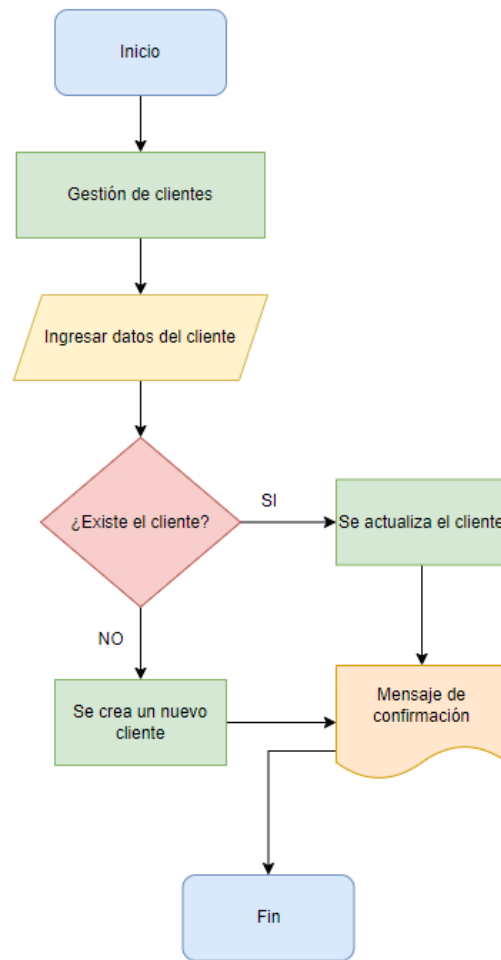


Fuente: Elaboración propia.

En la siguiente figura 17 se puede observar el diagrama de flujo para la gestión de clientes

Figura 17

Gestión de clientes

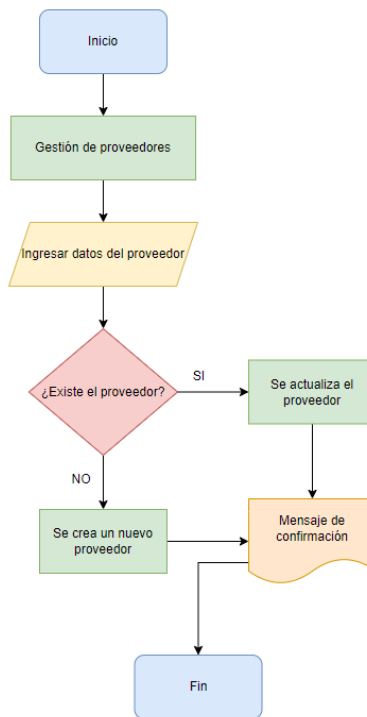


Fuente: Elaboración propia.

En la siguiente figura 18 se puede observar el diagrama de flujo para la gestión de proveedores.

Figura 18

Gestión de proveedores

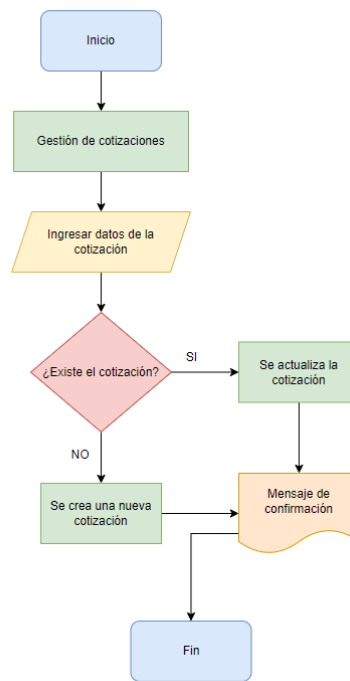


Fuente: Elaboración propia.

En la siguiente figura 19 se puede observar el diagrama de flujo para la gestión de cotizaciones.

Figura 19

Gestión de cotizaciones

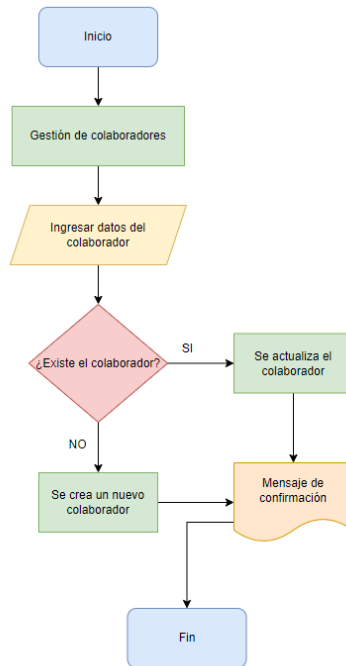


Fuente: Elaboración propia.

En la siguiente figura 20 se puede observar el diagrama de flujo para la gestión de colaboradores

Figura 20

Gestión de colaboradores

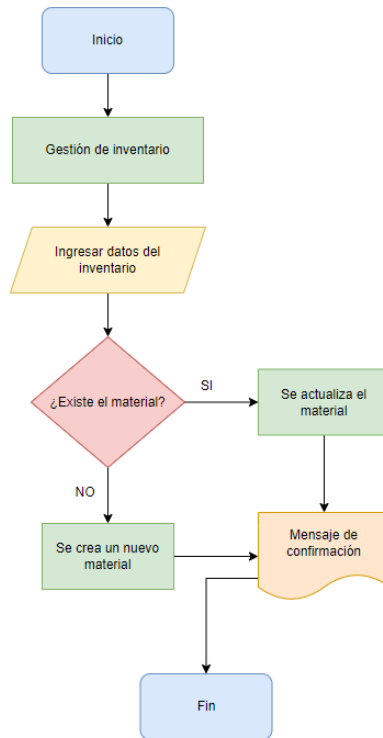


Fuente: Elaboración propia.

En la siguiente figura 21 se puede observar el diagrama de flujo para la gestión de inventario

Figura 21

Gestión de inventario

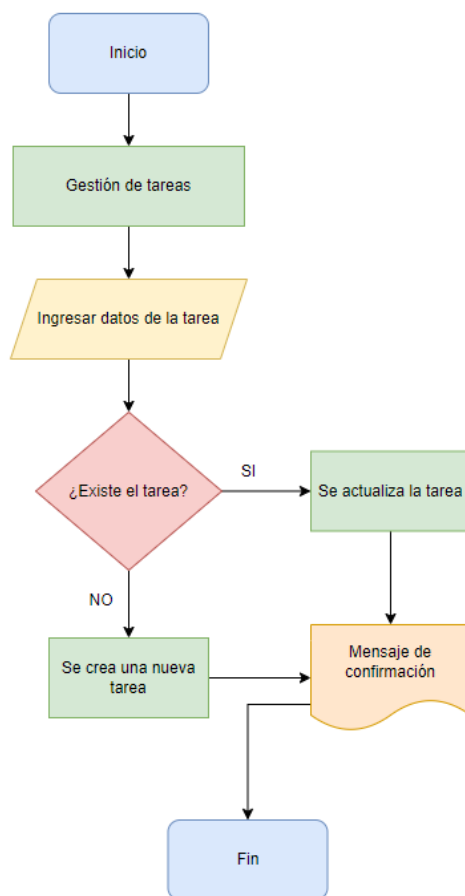


Fuente: Elaboración propia.

En la siguiente figura 22 se puede observar el diagrama de flujo para la gestión de tareas

Figura 22

Gestión de tareas



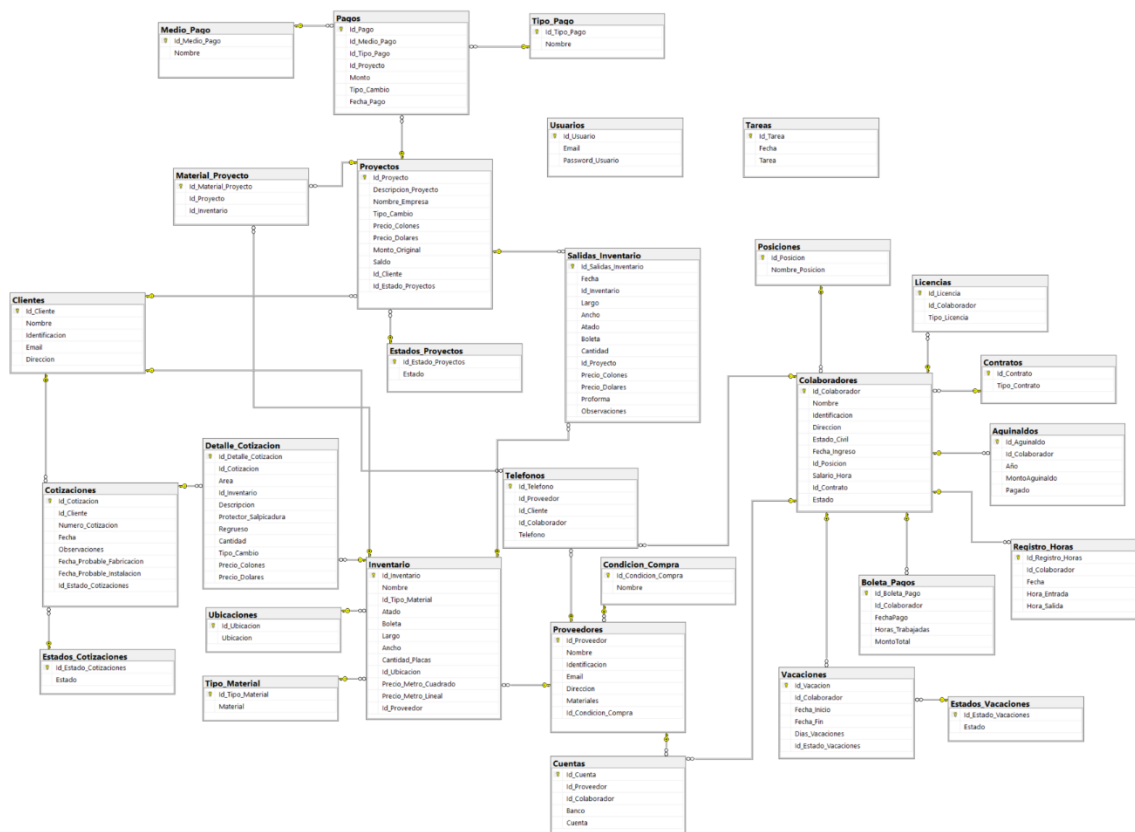
Fuente: Elaboración propia.

Diagrama de base de datos

En la siguiente figura 23 se puede observar el diagrama de base de datos

Figura 23

Diagrama de Base de Datos



Fuente: Elaboración propia.

Propuesta implementación

La propuesta de implementación del SIGG en un entorno de producción considera la arquitectura tecnológica utilizada durante su desarrollo, que incluye SQL Server para la gestión de datos, un backend desarrollado en .NET 8 con APIs, y un frontend basado en MVC, igualmente implementado en .NET 8.

Ahora bien, luego de las consideraciones que se tomaron para una propuesta de implementación se deben seguir varios pasos a tomar en cuenta para una implementación exitosa.

1. **Preparación del entorno de producción:** se debe asegurar un entorno que cuente con una infraestructura adecuada de acuerdo con el diseño de arquitectura presentado en este documento. Eso quiere decir que se debe contar con servidores que soporten .NET 8 tanto para el backend como el frontend del sistema y un servidor para base de

datos con SQL Server. Con este primer paso se puede asegurar un ambiente optimo para una implementación en producción.

2. **Despliegue del Backend en entorno productivo:** El backend desarrollado en .NET 8 debe ser desplegado en un servidor que soporte aplicaciones ASP.NET. Se recomienda utilizar un servidor web como IIS (Internet Information Services) que es lo tradicional, configurado para gestionar las conexiones entre las APIs y la base de datos. Es crucial configurar el acceso a las APIs mediante protocolos de seguridad como HTTPS.
3. **Despliegue y configuración del Frontend:** El frontend, desarrollado en MVC sobre .NET 8, se integrará de manera fluida con las APIs del backend. Su despliegue se puede realizar en el mismo servidor que el backend o en servidores separados, dependiendo de la infraestructura disponible.
4. **Configuración de la base de datos en SQL Server:** La base de datos SQL Server debe configurarse en el entorno de producción con las mismas estructuras que en el entorno de desarrollo, asegurando que las tablas y procedimientos almacenados sean replicados correctamente. Se deben activar los mecanismos de seguridad de SQL Server para proteger los datos, como la encriptación de datos sensibles y la configuración de roles de acceso según cada perfil administrativo del sistema.
5. **Pruebas y validación:** Antes de poner el SIGG en marcha, es necesario realizar pruebas de aceptación en el entorno de producción para garantizar que todo funcione como se espera. Estas pruebas deben abarcar la funcionalidad de las APIs, la correcta interacción del frontend con el backend, la integridad de los datos en SQL Server. También es esencial verificar que las medidas de seguridad implementadas, tanto en el acceso a la base de datos como en la comunicación entre las capas del sistema, estén funcionando correctamente.
6. **Capacitación y soporte técnico:** Una vez desplegado, se debe ofrecer capacitación al personal encargado de utilizar el SIGG, de manera que comprendan las funciones clave del sistema y puedan aprovechar su potencial. Además, debe establecerse un plan de soporte técnico para solucionar problemas que puedan surgir en el uso diario del sistema, así como garantizar la actualización y el mantenimiento del sistema para adaptarse a futuras necesidades.

Con estos pasos se puede asegurar que el SIGG este correctamente implementado en un entorno o ambiente de producción, ofreciendo una solución escalable para la gestión integral de la empresa.

Conclusiones

Para cerrar el proyecto, es fundamental analizar cómo se alcanzaron los objetivos específicos planteados desde el inicio. En primer lugar, se identificaron de manera precisa los requisitos funcionales del sistema SIGG, lo que permitió establecer una base sólida para su desarrollo. Este análisis detallado consideró tanto las necesidades operativas de los usuarios como las funciones clave que debía cumplir el sistema, asegurando que el SIGG respondiera a los objetivos de gestión y control establecidos.

Posteriormente, se diseñó la arquitectura del sistema, la base de datos, la comunicación entre los componentes y la interfaz de usuario de manera detallada y estructurada. Este proceso permitió desarrollar una infraestructura técnica robusta, que optimiza el flujo de información y mejora la experiencia del usuario. La elección de herramientas y tecnologías se basó en criterios de escalabilidad, rendimiento y usabilidad, permitiendo una comunicación fluida entre los distintos módulos del sistema.

El objetivo de desarrollar el SIGG en base a los requisitos y diseños establecidos, se cumplió mediante un enfoque iterativo de desarrollo, que permitió la implementación y prueba de cada módulo de manera gradual. El uso de buenas prácticas de desarrollo, junto con pruebas continuas, garantizó la calidad y estabilidad del sistema.

Finalmente, se propuso una estrategia de implementación del SIGG en un entorno de producción. Esta propuesta incluyó recomendaciones para la integración progresiva del sistema en los procesos de la empresa, tomando en cuenta aspectos como la capacitación del personal, la migración de datos y la gestión del cambio.

Como resultado, los objetivos específicos del proyecto se han logrado plenamente, lo que ha permitido desarrollar un sistema robusto y funcional, listo para su implementación en un entorno productivo. Este sistema ha sido diseñado no solo para mejorar la eficiencia operativa, sino también para apoyar a la empresa en la toma de decisiones estratégicas mediante la optimización de sus procesos y el manejo eficiente de la información.

Recomendaciones

Con base en el proyecto realizado, desde la redacción de los requerimientos hasta el desarrollo del sistema, es posible identificar una serie de recomendaciones que podrían ser de gran valor para el futuro. Estas recomendaciones buscan asegurar que tanto la empresa como el sistema logren un crecimiento sostenible a lo largo del tiempo.

Para empezar, sería ideal contar con un módulo administrador en el sistema que gestione el cambio de contraseñas, roles del sistema, y algunos aspectos configurativos, como el tipo de cambio del día. Además, este módulo podría facilitar la adición o eliminación de datos en algunos mantenimientos que no requieren su propio módulo, como el tipo de contratación o la condición de compra de los materiales, entre otros temas configurativos.

Luego de esto, se podría optar por una integración con soluciones móviles. Desarrollar una aplicación que permita el acceso al sistema desde cualquier lugar sería beneficioso tanto para los colaboradores como para los administradores del negocio. Esto sería especialmente relevante para la gestión de inventarios, proyectos y cotizaciones en tiempo real, permitiendo un manejo más ágil y eficiente mientras los colaboradores estén en el campo o fuera de la oficina.

En el futuro, se puede considerar la adición de un módulo de gestión financiera dentro del SIGG. Este incluiría el control detallado de presupuestos, cuentas por pagar y por cobrar, análisis de rentabilidad de proyectos y generación de informes financieros automáticos, brindando una visión financiera integral que facilitaría la toma de decisiones estratégicas.

Una integración con sistemas de terceros también sería clave. Dado que la facturación electrónica es un requisito en el país, si el sistema no cuenta con un módulo específico para este proceso, se recomienda integrar un sistema de facturación electrónica ya existente, asegurando el cumplimiento normativo y facilitando las operaciones.

Finalmente, a medida que la empresa crezca, se recomienda evaluar la escalabilidad y adaptación a la nube. Migrar el sistema a una plataforma basada en la nube, como Azure, Google Cloud o AWS, permitiría una mayor escalabilidad, acceso remoto más seguro, respaldo de datos automatizado y una gestión de infraestructura simplificada. Estas

plataformas ofrecen un soporte sólido para .NET 8, lo que facilitaría la implementación del sistema en producción y garantizaría su sostenibilidad en el largo plazo.

Para asegurar el éxito del sistema en el futuro, es esencial mantener una cultura de actualización continua y soporte técnico. A medida que el SIGG se implemente y use a mayor escala, es fundamental realizar revisiones periódicas, optimizar el rendimiento y seguir actualizando la tecnología utilizada. Esto no solo garantizará que el sistema esté alineado con las necesidades de la empresa, sino que también permitirá que evolucione de manera fluida junto con el negocio.

Referencias

- Aristizábal Martínez, D. A., & Quiceno Metaute, S. M. (2023). Lógica de programación básica orientada a objetos con ejercicios resueltos (1ª ed.). Instituto Tecnológico Metropolitano. <https://elibro.net/es/lc/uisil/titulos/253798>
- Canosa Ferreiro, A. J. (2024). SCRUM: Teoría e implementación práctica (1ª ed.). RA-MA Editorial. <https://elibro.net/es/lc/uisil/titulos/267660>
- Casado Iglesias, C. (2015). Entornos de desarrollo (1ª ed.). RA-MA Editorial. <https://elibro.net/es/lc/uisil/titulos/62495>
- Cobarsi-Morales, J. (2011). Sistemas de información en la empresa (1ª ed.). Editorial UOC. <https://elibro.net/es/lc/uisil/titulos/33493>
- Evalando ERP. (2021, 1 de junio). ¿Qué es un sistema de gestión? En Evalando ERP. Recuperado de <https://www.evalandoerp.com/sistema-de-gestion/sistema-erp/que-es-un-sistema-de-gestion/>
- Flanagan. (2005-2006). Ciclo de vida. Recuperado de <https://flanagan.ugr.es/docencia/2005-2006/2/apuntes/ciclovida.pdf>
- García Notario, D. (2015). Análisis de requisitos en el desarrollo del software. Universidad Carlos III de Madrid. <https://e-archivo.uc3m.es/rest/api/core/bitstreams/a66b0a2d-fa7c-483f-ac5e-1476ff2da8eb/content>
- Manene, L. M. (2011). Los diagramas de flujo: Su definición, objetivo, ventajas, elaboración, fases, reglas y ejemplos de aplicaciones. https://d1wqtxts1xzle7.cloudfront.net/60656037/Los_diagramas20190920-8696-u4r0qz-libre.pdf?1568999126=&response-content-disposition=inline%3B+filename%3DEl_28_julio_2011_en_Estructura_Organizat.pdf&Expires=1722821872&Signature=BWZ0-dmyJKkDOUCds9pqTMvqJQE7sWEPqrvQJ4uwPhReMBSwHt2U-LgMpQ3-cPHWzV97LEiGM4MqyAQDevcARi4gUvN5o1vt5wGrJ3g7Vs~PVYhIVCg7Yzo6O6VUSqFB5m06nUrg054tzeFL5wa3OocsGkUb3Vu4Xu4lgBKEl1hkPwbUnnR-2WQc1APH6TMKw81JyRuDO~k9TOMvsJsdTKoGZTEZQ8hS-

6ZlBIf6SfTAu5nwc-8esz4cvwh6OaeFTPJWkeD2zZniMQqO9J14guImXDVyexKIwtyoBfaTgpWgKCA DjD4DKySxTbVmfJkevyOn2jU2xXUrYDGJgbTJEw_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

- Marqués, M. (2011). Bases de datos. Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions. <https://bdigital.uvhm.edu.mx/wp-content/uploads/2020/05/Bases-de-Datos.pdf>
- Maida, E. G., & Pacienza, J. (2015). Metodologías de desarrollo de software [Tesis de Licenciatura, Universidad Católica Argentina, Facultad de Química e Ingeniería “Fray Rogelio Bacon”]. Biblioteca Digital de la Universidad Católica Argentina. <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Navarro Huerga, M., & Fernández Otero, M. (2014). Sistemas de información en la Empresa (1ª ed.). Editorial Universidad de Alcalá. <https://elibro.net/es/lc/uisil/titulos/42929>
- Raya Cabrera, J. L., & Raya González, L. (2015). Sistemas informáticos (1ª ed.). RA-MA Editorial. <https://elibro.net/es/lc/uisil/titulos/62481>
- Schwaber, K., & Sutherland, J. (2013, julio). Guía de Scrum: La guía definitiva para Scrum: Las reglas del juego. Scrum Guides. <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Serrano, C. (2023, abril 24). Beneficios de un sistema integrado de gestión. Nalanda Global. <https://www.nalandaglobal.com/blog/beneficios-de-un-sistema-integrado-de-gestion/>
- Valencia Cabrera, L. (2021). Introducción a SQL. Research Group on Natural Computing, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Sevilla. <https://www.cs.us.es/cursos/bd-2021/temas/BD-Tema-5.pdf>

Anexos



Granitos Naranjo

Granito - Mármol - Cuarzo - Onix
2771-4489 / 85762852
Los detalles hacen la diferencia...

San Isidro del General, San José Costa Rica

4 de abril del 2025.

Ing. Ruddy Rodríguez Acuña
Director de la Escuela de Ingeniería en Sistemas
Universidad Internacional San Isidro Labrador

Estimado Señor

Yo Daniel Hidalgo Naranjo en calidad de dueño de la empresa Granitos Naranjo, hago constar por este medio que el estudiante Alejandro Umaña Pérez cedula 11577 0634 de la carrera de Licenciatura de Ingeniería en Sistemas, cumplió satisfactoriamente la creación, propuesta de implementación de un sistema para el área administrativa, donde el control de clientes y proyectos de la empresa se verán beneficiados ; ya que el mismo mejora la manipulación de la información haciendo que el proceso de las actividades de la empresa sea más ágil, cumpliendo con ello el principal requisito de graduación.

Atentamente

Daniel Hidalgo Naranjo
Ceo



Tribunal Examinador

Proyecto Programado grado de Licenciatura en Ingeniería de Sistemas,
presentado en abril del 2025, en la Universidad Internacional San Isidro Labrador
ante el siguiente tribunal examinador.

Ing. Ruddy Rodríguez Acuña
Director de La Escuela de Ingeniería en Sistemas

Ing. Eric Corella Solís
Profesor Tutor

Ing. Michael Corrales Oviedo
Profesor Lector

Ing. Alejandro Umaña Pérez