Imagen que contiene dibujo, reloj, señal

Descripción generada automáticamente

Facultad de Ingeniería

**Carrera Profesional de Ingeniería de Software y Sistemas**

**Curso: Integrador I: Sistemas - Software**

**SISTEMA DE GESTIÓN INVENTARIOS Y VENTAS DE PRODUCTOS DE HIDROCARBUROS PARA LA EMPRESA “LA CASA DEL SERAPHIN GAS”**

Estudiantes:

1. JAHUAR CHIRINOS, Anibal Alejandro
2. ROJAS BARBARAN, Eberth Gianfranco
3. ROMERO RIVAS, Romina Ariadna
4. TASAYCO MAGALLANES, Giordano Martin
5. UIPAN MANTARI, Renato Alonso

**Docente: Dr. Santiago Raul Gonzales Sanchez**

**Lima – Perú**

**2024**

**ÍNDICE**

[CAPÍTULO 1 - ASPECTOS GENERALES 4](#_Toc178436600)

[1.1 Descripción Del Problema 4](#_Toc178436601)

[1.2 Definición De Objetivos 4](#_Toc178436602)

[1.2.1 Objetivo General 4](#_Toc178436603)

[1.2.2 Objetivos Específicos 4](#_Toc178436604)

[1.3 Alcances de la Solución 5](#_Toc178436605)

[1.4 Justificación 5](#_Toc178436606)

[1.5 Estado del Arte 6](#_Toc178436607)

[CAPÍTULO 2 - MARCO TEÓRICO 9](#_Toc178436608)

[2.1 Marco teórico de programación orientada a objetos 9](#_Toc178436609)

[2.2 Marco teórico de las fórmulas y/o el sistema 12](#_Toc178436610)

[CAPÍTULO 3- DESARROLLO DE LA SOLUCIÓN 14](#_Toc178436611)

[3.1. Requerimientos Funcionales Y No Funcionales 14](#_Toc178436612)

[3.2. Diagramas De Procesos 16](#_Toc178436613)

[3.3 Diagrama de clases 21](#_Toc178436614)

[3.4. Diagrama De Entidad-Relación 24](#_Toc178436615)

[3.5 Diseño Lógico Y Físico De La Base De Datos 26](#_Toc178436616)

[3.6 Mockups (Prototipos UX/UI) y Documentación 29](#_Toc178436617)

[3.6.1. Mockups (Prototipos UX/UI) 29](#_Toc178436618)

[3.6.2. Documentación UX/UI 34](#_Toc178436619)

[3.6.3. Identificación De Reportes 37](#_Toc178436620)

[3.7. Diagramas de Arquitectura (estructura MVC, implementación de patrones SOLID e integración de capas DAO) 38](#_Toc178436621)

[3.8 Funcionalidades del código fuente 38](#_Toc178436622)

[CAPITULO 4-RESULTADOS 38](#_Toc178436623)

[CONCLUSIONES 38](#_Toc178436624)

[RECOMENDACIONES 38](#_Toc178436625)

[BIBLIOGRAFÍAS 39](#_Toc178436626)

[ANEXOS 40](#_Toc178436627)

[Lean Canvas 40](#_Toc178436628)

[EL PMBOK (PROJECT MANAGEMENT BODY OF KNOWLEDGE) 44](#_Toc178436629)

[1. INICIO DEL PROYECTO 44](#_Toc178436630)

[2. PLANIFICACIÓN DEL PROYECTO 45](#_Toc178436631)

[3. EJECUCIÓN 47](#_Toc178436632)

[4. MONITOREO Y CONTROL 48](#_Toc178436633)

[5. CIERRE DEL PROYECTO 49](#_Toc178436634)

[PLANIFICACIÓN DEL PROYECTO 51](#_Toc178436635)

[PROJECT CHARTER (ACTA DE CONSTITUCIÓN DEL PROYECTO) 51](#_Toc178436636)

[WORK BREAKDOWN STRUCTURE (WBS – ESTRUCTURA DE DESGLOSE DEL TRABAJO) 52](#_Toc178436637)

[1. Inicio del Proyecto 52](#_Toc178436638)

[2. Planificación del Proyecto 52](#_Toc178436639)

[3. Desarrollo del Sistema 53](#_Toc178436640)

[4. Monitoreo y Control 53](#_Toc178436641)

[5. Cierre del Proyecto 53](#_Toc178436642)

[GANTT CHART 54](#_Toc178436643)

[- Modelo de encuesta **¡Error! Marcador no definido.**](#_Toc178436644)

# CAPÍTULO 1 - ASPECTOS GENERALES

## [1.1](https://docs.google.com/document/d/1-4JYwYGUFEVT4-ObRbx1DWBqkIj23ZF1dhNdgtjFgbE/edit?pli=1#heading=h.30j0zll) Descripción Del Problema

La empresa **La Casa Seraphin Gas S.A.C.** Fue creada el **19 de marzo de 2006**, cuenta con 3 sedes en Lima, la principal está ubicada en **Cal. Alemania nro. 2505 int. 1 urb. La trinidad Lima - Lima – Lima.** Esta es una empresa que se encarga de la venta de diferentes productos de hidrocarburos a los grifos, todo lo que tenga que ver con hidrocarburos. Hasta ahora, la empresa gestiona sus productos con un programa desarrollado por uno de sus empleados; sin embargo, dicho programa está incompleto debido a la falta de tiempo, lo que ha ocasionado problemas en la gestión de producto. Esto, a su vez, ha generado dificultades en la creación de cotizaciones para los productos, provocando retrasos en los procesos de venta y afectando el servicio que se ofrece a los clientes. Estos inconvenientes han derivado en errores y dificultades en la atención al cliente.

Como solución, hemos propuesto el desarrollo de una aplicación integral que incluya diferentes módulos para la administración eficiente de producto y la generación automática de cotizaciones y ventas, todo en un único sistema.

## [1.2](https://docs.google.com/document/d/1-4JYwYGUFEVT4-ObRbx1DWBqkIj23ZF1dhNdgtjFgbE/edit?pli=1#heading=h.1fob9te) Definición De Objetivos

### 1.2.1 Objetivo General

Desarrollar un sistema de gestión de productos de hidrocarburos y servicios para la empresa La Casa Seraphin gas SAC- ruc: 20512972536.

### [1.2.2 Objetivos Específicos](https://docs.google.com/document/d/1-4JYwYGUFEVT4-ObRbx1DWBqkIj23ZF1dhNdgtjFgbE/edit?pli=1#heading=h.2et92p0)

* + Optimizar y mejorar el sistema de ventas de empresa Seraphin.
  + Automatizar la gestión de productos en el sistema de la empresa Seraphin.
  + Desarrollar un módulo que genere las cotizaciones y ventas de la empresa Seraphin.

## 1.3 [Alcances de la Solución](https://docs.google.com/document/d/1-4JYwYGUFEVT4-ObRbx1DWBqkIj23ZF1dhNdgtjFgbE/edit?pli=1#heading=h.tyjcwt)

* Desarrollo de la gestión de inventarios y generación de cotizaciones en la empresa La Casa Seraphin Gas.
* Se investigarán los métodos a la hora de insertar o remover un producto, también un método buscar cuando queremos localizar un producto en específico además de un método que pueda filtrar qué producto fue el más vendido y viceversa. ~~Entre otros~~.
* Se realizará el análisis, diseño y desarrollo del "Sistema de Gestión de Productos de Hidrocarburos y Ventas", que integrará los módulos de productos y ventas en una sola aplicación.
* Se desarrollarán prototipos del sistema que serán probados utilizando datos reales de la empresa La Casa Seraphin Gas para verificar su efectividad.

## 1.4 Justificación

[El presente trabajo de investigación surge de la necesidad de la empresa "La Casa Seraphin Gas" que tiene problemas en su gestión interna, enfrentando problemas significativos en la administración del inventario y en la elaboración de cotizaciones para productos y servicios. Estas deficiencias afectan la eficiencia operativa y la calidad del servicio al cliente, lo que justifica la implementación de una solución integral. El sistema de gestión de productos de hidrocarburos y servicios de la casa de Seraphin Gas. Desde el punto de vista teórico, la complejidad de gestionar manualmente los inventarios y cotizaciones en una empresa que maneja productos de hidrocarburos y servicios diversos sugiere que es necesario un sistema automatizado que integre estas funciones de manera eficiente. El problema radica en la incapacidad de mantener un control preciso del inventario, lo que puede resultar en la falta de productos esenciales o en la adquisición innecesaria de aquellos ya disponibles, impactando negativamente en los costos y tiempos de operación. Y, desde el punto de vista práctico, el desarrollo de un sistema de gestión permitirá la automatización de procesos clave como la entrada y salida de productos, el seguimiento en tiempo real del stock, y la generación de cotizaciones personalizadas. Este sistema no solo reducirá los tiempos de operación, sino que también mejorará la precisión en la gestión del inventario y la automatización de costes.](https://docs.google.com/document/d/1-4JYwYGUFEVT4-ObRbx1DWBqkIj23ZF1dhNdgtjFgbE/edit?pli=1#heading=h.3dy6vkm)

[El sistema propuesto ofrecerá beneficios, tales como:](https://docs.google.com/document/d/1-4JYwYGUFEVT4-ObRbx1DWBqkIj23ZF1dhNdgtjFgbE/edit?pli=1#heading=h.3dy6vkm)

* Reducción de errores en la administración del inventario y cotizaciones.
* Automatización de cotizaciones, adaptándolas a las necesidades específicas de diferentes clientes.
* Desarrollo de una base de datos de la empresa, (gestión de productos e información de clientes) que actualmente no cuenta con una. Esta base de datos se integrará a nuestro sistema, facilitando el acceso y la gestión de la información.

## [1.5 Estado del Arte](https://docs.google.com/document/d/1-4JYwYGUFEVT4-ObRbx1DWBqkIj23ZF1dhNdgtjFgbE/edit?pli=1#heading=h.1t3h5sf)

Dentro la revisión de la literatura se han ubicado los siguientes trabajos de investigación:

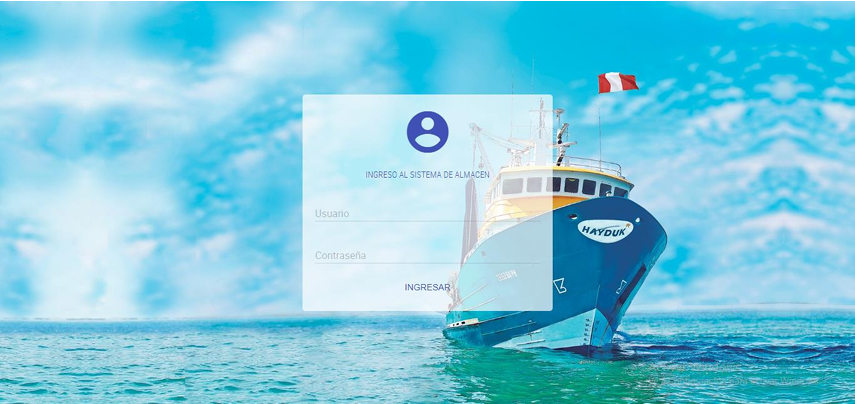
Calderón (2023) realizó la investigación de un *“Si*stema Web para el control del almacén de congelados*”* Chimbote, Perú. para la empresa pesquera Hayduk S.A. En su investigación, se enfoca que se obtuvo la rapidez y eficiencia que beneficiarán al personal del área de almacén para la gestión de sus datos. Este enfoque es esencial para el control diario de la información generada por la empresa. La metodología SCRUM fue utilizada en el proyecto: Es el enfoque más comúnmente empleado para el análisis, implementación y documentación de sistemas orientados a objetos. La empresa DIGITAL SUR S.A.C. Ha manejado todas sus responsabilidades manualmente durante años debido a la falta de actualización tecnológica, lo que ha hecho que el trabajo sea monótono. Por esta razón, se propuso desarrollar un sistema de ventas y almacén como una prioridad clave para mantenerse al día y aumentar sus ventas. Por lo que se concluyó que el 94,12% de los encuestados está a favor de implementar el sistema web, mientras que el 83,01% desaprobó el sistema actual. Además, la elección de la metodología garantizó el desarrollo adecuado de la propuesta para implementar un sistema web que mejore la gestión del almacén de congelados.



***Figura 1.1. Menú inicio de Sistema de Almacén Hayduk S.A***

Fuente: Calderon Lucero, C.E (2023)

Calderon (2023) realizó la investigación *“Un* sistema Web para el control del almacén de congelados*”* Chimbote, Perú. En su estudio, describe cómo la implementación de un sistema web específico permitió optimizar el control de inventarios, mejorando la eficiencia y la gestión de los bienes y activos de la empresa.

****

***Figura 1.2. Menú inicio de Sistema de Almacén Hayduk S.A***

Fuente: Calderon Lucero, C.E (2023)

Yarín Achachagua (2017) llevó a cabo la investigación sobre el "*Diseño e implementación de un sistema de localización y control de inventarios en un almacén de aduanas, utilizando tecnología RFID"*. Lima- Perú. Su estudio se centra en la implementación de un sistema que emplea tecnología de identificación por radiofrecuencia (RFID) para mejorar la gestión de inventarios en un entorno aduanero. La metodología utilizada comprende un enfoque aplicado con un estudio de caso, lo que permitió recopilar datos relevantes sobre el estado actual de los procesos logísticos y el manejo de inventarios dentro de la institución analizada.

La implementación del sistema RFID no solo se basó en la tecnología, sino que también contempló el diseño de procesos más eficientes para la localización y control de los productos almacenados. Los resultados evidenciaron que, tras la aplicación del sistema, se logró una reducción significativa en el tiempo de despacho de contenedores y carga suelta, así como una disminución en los costos operativos. Estos hallazgos indican que la integración de RFID en la gestión de almacenes puede contribuir de manera efectiva a la optimización de procesos logísticos en las aduanas, fortaleciendo el control y trazabilidad de los inventarios en estas instalaciones.

El estudio de Yarín Achachagua demuestra que el diseño e implementación de un sistema basado en tecnología RFID es una herramienta poderosa que mejora la eficiencia operativa y reduce costos en el contexto de la gestión de inventarios en un almacén aduanero, estableciendo un nuevo estándar en la optimización de la logística en este sector.

 ***Figura 1.3. Menú inicio de Sistema RFID.***

Fuente: Yarín Achachagua, Y. H. (2017).

# **CAPÍTULO 2 - MARCO TEÓRICO**

## 2.1 Marco teórico de programación orientada a objetos

**Definición de POO (Programación Orientada a Objetos):** La programación orientada a objetos (POO) es un paradigma que actualmente es el más utilizado en el desarrollo de software. Moreno (2015) menciona:

Con la programación orientada a objetos (Poo) se aumenta el modularidad de los programas y la reutilización de los mismos. además, la Poo se diferencia de la programación clásica porque utiliza técnicas nuevas como el polimorfismo, el encapsulamiento, la herencia, etc. (p.14)

**Origen y evolución de POO:**

Surgen en 1960, y aunque algunos autores diﬁeren en sus orígenes, los conceptos de la POO tienen su inicio en Simula 67, un lenguaje diseñado en el centro de cómputo noruego en Oslo1. Posteriormente, en agosto de 1981, se publica en la revista Byte la descripción del lenguaje de programación Smalltalk2, el cual reﬁnó algunos de los conceptos originados con el lenguaje Simula. Lo anterior dio pie a que, en la década de 1980, los lenguajes de programación Orientados a Objetos (OO) tuvieran un rápido auge y expansión, por lo que la POO se fue convirtiendo en el estilo de programación dominan- te a mediados de los años ochenta del siglo XX, continuando vigente hasta nuestros días. (Ruiz, 2009, p.24)

**Conceptos clave de POO:**

**A. Objeto:**

Es una instancia de un modelo que posee características y métodos específicos. Según García (2010) define que “Un objeto es un concepto, abstracción o cosa con límites bien definidos y con significado dentro del problema” (p.6).

**B. Clase:**

“Una clase describe a un conjunto de objetos que comparten una estructura y un comportamiento común” (García, 2010, p.7).

“Una clase en Programación Orientada a Objetos actúa como una plantilla, modelo, o prototipo, a partir de la cual se obtienen instancias, habitualmente llamadas objetos” (Blasco, 2019, p.89).

**C. Métodos**:

“Como unidad mínima aglutinadora de un conjunto de instrucciones, con entidad propia, y susceptible de ser invocada en bloque, puede recibir datos al ser invocado y/o devolver datos generados durante la ejecución del método” (Blasco, 2019, p100).

**D. Encapsulamiento:**

 “Tiene que ver con reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los módulos o componentes del sistema” (Ruiz, 2009, p.34).

**E. Abstracción:**

En el contexto de programación orientada a objetos, según Moreno (2015) afirma: “Cuando se programa orientado a objetos, lo que se hace es abstraer las características de los objetos que van a formar parte del programa, y crear las clases con sus atributos y sus métodos” (p.101).

**F. Herencia:**

 “La herencia es la base de la reutilización del código. Cuando una clase deriva de una clase padre, esta hereda todos los miembros y métodos de su antecesor” (Moreno, 2015, p.91).

**G. Polimorfismo:**

“El polimorfismo permite crear varias formas del mismo método, de tal manera que un mismo método ofrezca comportamientos diferentes” (Moreno, 2015, p.102).

**H. Método Get:**

En el contexto de POO, es una función que se utiliza para “obtener” un valor de un atributo específico de la clase.

Este método es utilizado comúnmente para recuperar u obtener el valor de un atributo, y al igual que antes, debería haber un método de este tipo por cada atributo que contenga la clase y que se requiera visualizar desde el exterior. (Ruiz, 2009, p.42)

**I. Método Set:**

Es un método que se utiliza para modificar el valor de un atributo privado o protegido de una clase.

Este concepto es relevante para nuestro proyecto, ya que la implementación de métodos Set será crucial para permitir la actualización y gestión adecuada de los atributos privados en las clases que diseñemos, asegurando así la integridad de los datos en nuestro sistema. (Torres Remón, 2018, p. 168)

La programación orientada a objetos se utiliza ampliamente en la industria del software debido a su capacidad para abordar problemas complejos en diversos dominios, como la ingeniería de software, los sistemas empresariales y las aplicaciones móviles. Los lenguajes y herramientas de programación populares como Java, Python, C++ y C# Están diseñados para admitir programación orientada a objetos, lo que facilita su adopción en proyectos grandes.

El marco teórico de POO revisión de literatura, como ¿Qué es un objeto? Clases, objetos e instancias, Principios de la Programación Orientada a Objetos, Características de la POO; Beneficios de Programación Orientada a Objetos

## 2.2 Marco teórico de las fórmulas y/o el sistema

Al desarrollar sistemas de software, el uso de fórmulas y algoritmos matemáticos para resolver problemas específicos e implementar funcionalidades centrales es fundamental. Dependiendo del área de aplicación, las fórmulas matemáticas pueden tener diversa complejidad y cumplir su propósito dentro del sistema. En esta sección, la atención se centra en describir los conceptos teóricos detrás de las fórmulas y sistemas relevantes para el proyecto y la forma en que se integrarán en la solución propuesta.

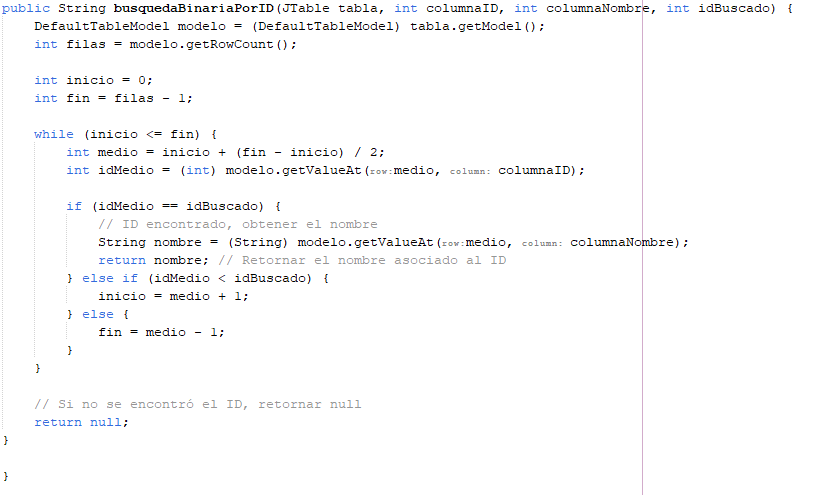
**CONCEPTO DE FÓRMULAS EN PROGRAMACIÓN**.

Una fórmula es una expresión matemática que relaciona varias variables para obtener un resultado específico. En el contexto de la programación, las fórmulas se implementan como expresiones en el código que procesan datos para producir resultados. Las fórmulas pueden variar desde simples operaciones aritméticas hasta complejos algoritmos de cálculo. Las fórmulas son cruciales en aplicaciones que requieren procesamiento numérico, como en sistemas financieros, científicos o de ingeniería. La correcta implementación de las fórmulas asegura la precisión de los cálculos y la validez de los resultados en el sistema.

Tipos de fórmulas y su aplicación. Cantidad de stock formula de programación promedio etc., como grafico de barras (esa clase debemos capturas el codigo y lo explicas)

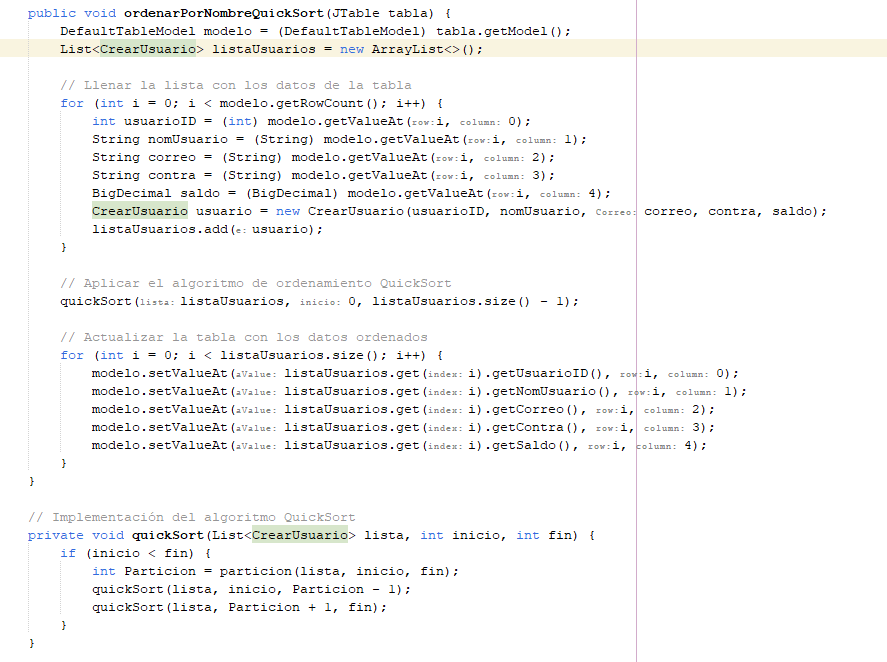
* **Fórmulas aritméticas básicas:** Incluyen operaciones como suma, resta, multiplicación y división. Aparecen en todo tipo de aplicaciones, desde la gestión de inventario hasta la fijación de precios en los sistemas de distribución.
* **Fórmulas algorítmicas:** Se utilizará algoritmos como la búsqueda Binaria y los de ordenamiento esto con el fin de tener control sobre la información de los productos que se estén manejando.

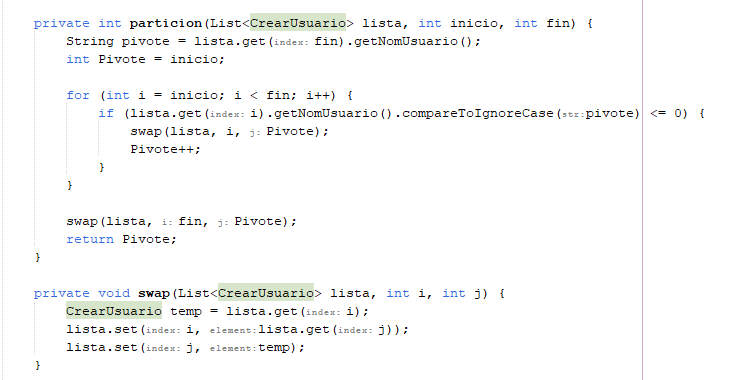
**Búsqueda binaria:** Es el algoritmo mas eficiente al momento de buscar un elemento en una lista ordenada. Su funcionamiento consiste en dividir repetitivamente a la mitad una parte de la lista que puede contener elemento logrando reducir las ubicaciones posibles a solo una. A continuación, se mostrará el código:



Como los datos se mostrarán en tablas los parámetros que le daremos será el modelo de la tabla, la columna donde estará el ID, la columna donde estará el nombre y el ID que se está buscando.

**Quicksort:** Para el algoritmo de ordenamiento tomamos el Quicksort porque actualmente es el más rápido y eficiente para el ordenamiento de los elementos. Su funcionamiento sigue la frase “divide y vencerás” debido a por cada recursión que hace divide la lista en sub-listas haciéndolo más pequeño para poder resolverlas por separado. A continuación, se mostrará el algoritmo a utilizar.





El funcionamiento del algoritmo es el siguiente al método padre se le dará el modelo de la tabla luego de ello llamara a su método Quicksort el cual se encarga del ordenamiento de los datos utilizando los métodos de partición que divide la lista en listas más pequeñas y el swap que hace los intercambios para el ordenamiento de los datos una vez hecho esto se actualiza la tabla con el método padre que es OrdenarporNombreQuickSort.

* **Los gráficos estadísticos:** Herramienta que ayuda en comprender de mejor manera la información siendo mas sencilla y clara de procesar ayudando en identificar mejor que productos son más vendidos a diferencia de otros.

# CAPÍTULO 3- DESARROLLO DE LA SOLUCIÓN

## 3.1. Requerimientos Funcionales Y No Funcionales

**FUNCIONALES:**

Los requerimientos funcionales describen las funciones que el sistema debe realizar. Seguidamente, se detallan los requerimientos funcionales del software.

* **Gestión de inventario:**

El sistema permite gestionar el inventario, incluyendo la actualización y visualización de los productos

* **Emitir de cotizaciones:**

El sistema permite a los usuarios generar cotizaciones para los clientes de manera rápida y eficiente.

* **Generar reportes de cotizaciones:**

El sistema permite la generación de reporte detallado sobre las cotizaciones solicitadas

* **Generar reportes de stock disponible:**

El sistema permite la generación de reporte de stocks disponibles en el inventario en cualquier momento

**NO FUNCIONALES**

Los requerimientos no funcionales describen cómo debe comportarse el sistema, definiendo principalmente la calidad y las restricciones del software. Asi mismo se detallarán los requerimientos no funcionales que garantizan la eficiencia, seguridad, y usabilidad del sistema:

* **Tiempo de Ejecución del Programa:**

El sistema debe completar las operaciones principales en 20 segundos, garantizando una respuesta ágil y eficiente para el usuario final.

* **Usabilidad:**

El programa debe ser intuitivo, ofreciendo una interfaz de usuario clara y fácil de entender, permitiendo a los usuarios, independientemente de su nivel técnico, navegar y utilizar todas las funciones sin dificultad.

* **Tiempo de Alerta de Escasez de Productos:**

El sistema debe generar y mostrar alertas de escasez de productos en un tiempo máximo de 2 segundos desde la detección del umbral crítico, asegurando una respuesta rápida para la gestión del inventario.

* **Compatibilidad con Sistemas Operativos:**

El software debe ser compatible con la mayoría de los sistemas operativos utilizados comúnmente, incluyendo Windows 10 y 11, para maximizar la accesibilidad y versatilidad del sistema.

## 3.2. Diagramas De Procesos

1. **Diagrama De Inventario**

* **Inició:** El proceso de inventariado da inicio con la revisión del stock.
* **Verificación:** Se verifica el stock actual de los productos.
* **Decisión:** Si un producto está por debajo del mínimo se contacta con el proveedor; si no lo está se toma registro.
* **Verificación:** Se consulta al proveedor si es que hay disponibilidad del producto que se necesita reponer.
* **Decisión:** Si hay disponibilidad entonces se crea una orden de compra; si no se contacta con otro proveedor.
* **Espera:** Se esperan los productos.
* **Recepción:** Se reciben los productos
* **Verificación:** Se verifica que los productos que se están entregando sean los correctos y que estén en buen estado.
* **Decisión:** Si todo está conforme se realiza el pago del producto; si no se retorna los productos al proveedor.
* **Derivar:** Una vez hecho el pago se derivan los productos al almacén.
* **Registro:** Se les toma registro a los nuevos productos.
* **Informar:** Se le informa al jefe del almacén de los productos.
* **Guardado:** El jefe de almacén guardará los productos en los estantes correspondientes

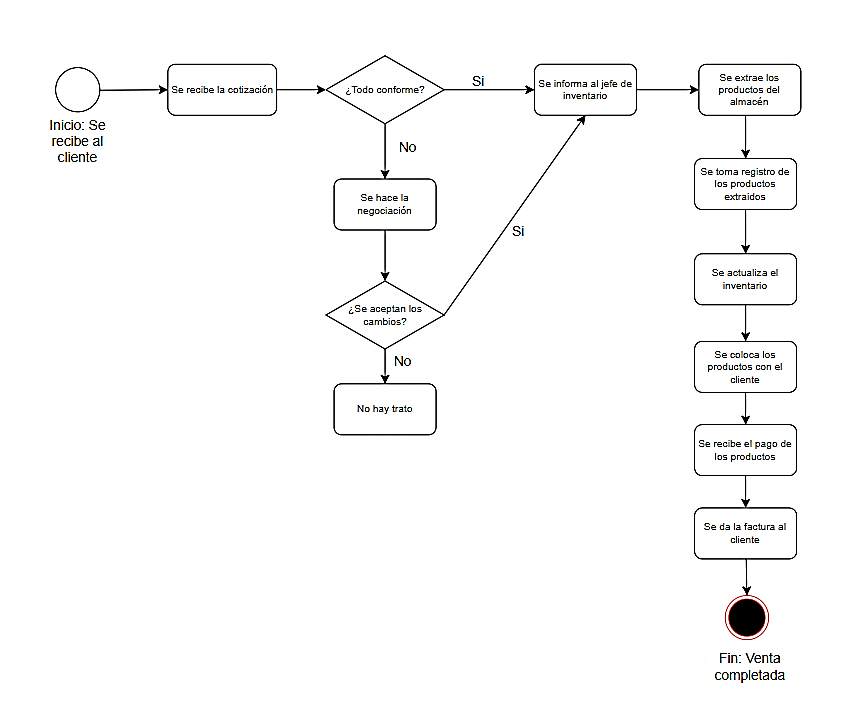
**Fin:** Una vez guardado el jefe de almacén crea un informe con el inventario actualizado.



***Figura 3.1. Sistema de Control de Inventario. Fuente: Elaboración Propia***

**B. Diagrama De Ventas**

* **Inicio:** Se recibe la cotización que fue entregada al cliente.
* **Verificación:** Se pregunta al cliente si está conforme con la cotización
* **Decisión:** Si el cliente está de acuerdo con la cotización entonces se le dará aviso al jefe de almacén; en caso no esté conforme se empezará una nueva negociación.
* **Verificación:** El cliente acepta los nuevos cambios
* **Decisión:** Si lo acepta entonces se procede a dar aviso al jefe de inventario si no lo acepta entonces se cancela el trato.
* **Extracción:** El jefe de almacén extrae los productos del almacén para la venta.
* **Registro:** Se toma registro de los productos que fueron extraídos.
* **Actualización:** Con el registro se actualiza el inventario.
* **Colocación:** Se da los productos al cliente para que los revise.
* **Recepción:** Se recibe el pago de los productos.
* **Entrega:** Entrega de la factura al cliente.
* **Fin:** La venta se ha completado.



***Figura 3.2. Sistema de Control de Ventas. Fuente: Elaboración Propia***

**C. Diagrama de Cotizaciones**

* **Inicio:** Se recibe la solicitud de cotización del cliente.
* **Recolección**: Se recolecta la información del cliente RUC, nombre etc.
* **Identificación:** Se identifica el producto que solicitó el cliente para ver el precio de los mencionados.
* **Verificación:** Se verifica que los productos estén disponibles
* **Decisión:** Si el producto está disponible se procede con la creación de la cotización; si no está disponible entonces se termina el proceso.
* **Generar:** Se genera la cotización para el cliente.
* **Envío:** Una vez generado se le envía la cotización al cliente.
* **Verificación:** El cliente acepta la cotización.
* **Decisión:** Si el cliente acepta la cotización se cierra el trato si no se termina el proceso.
* **Fin:** Trato cerrado.

**Diagrama

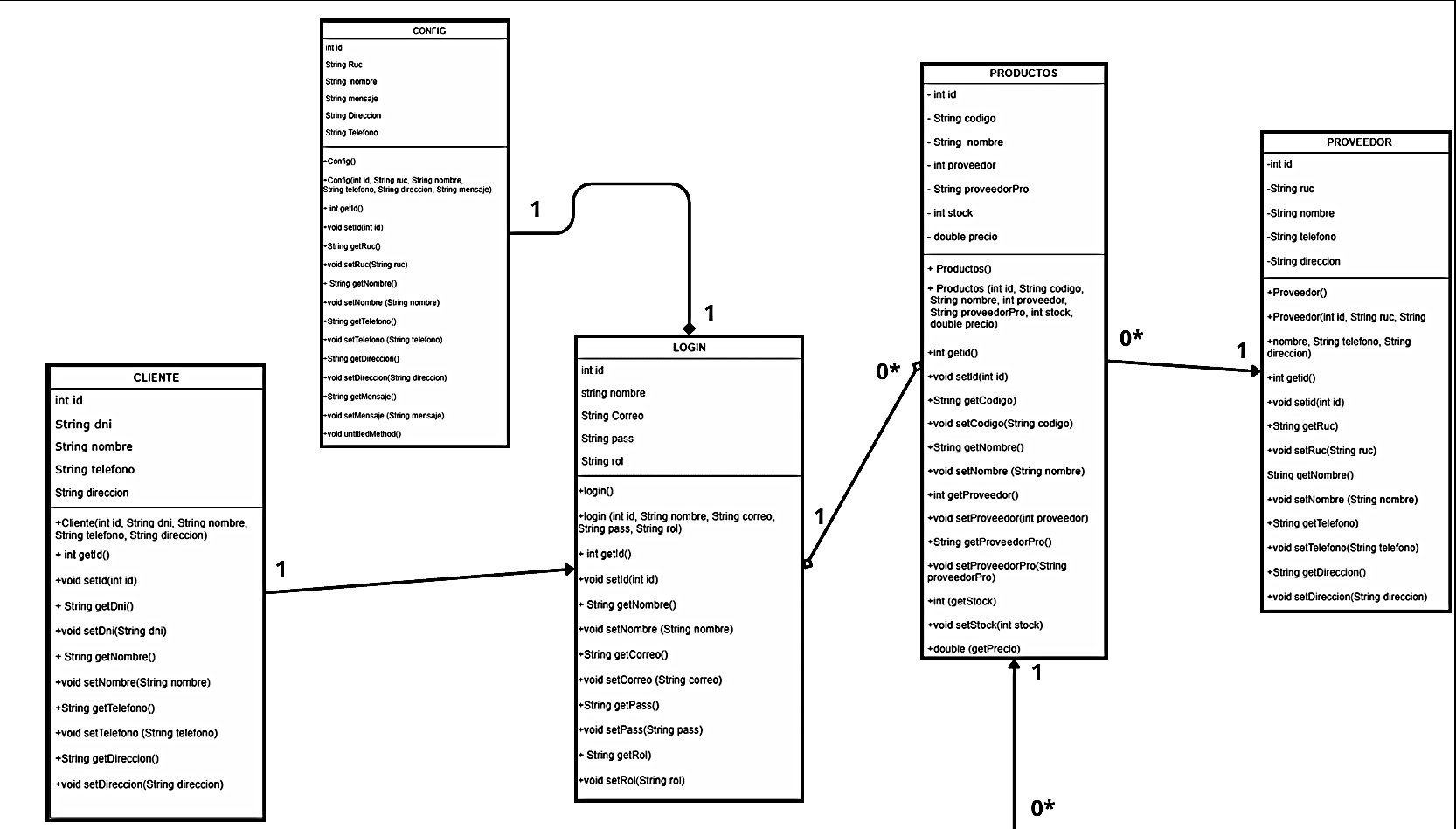
Descripción generada automáticamente**

***Figura 3.3. Sistema de Cotizaciones. Fuente: Elaboración Propia***

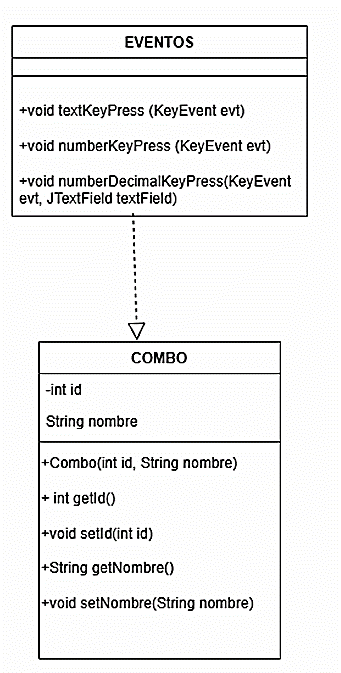
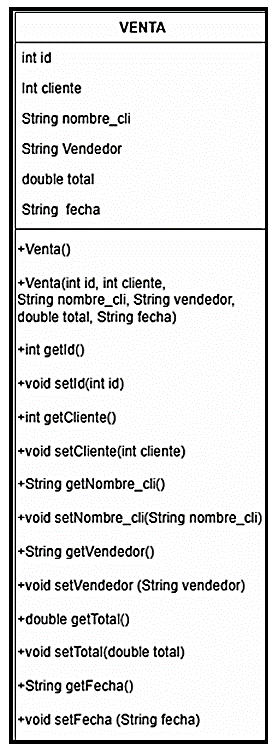
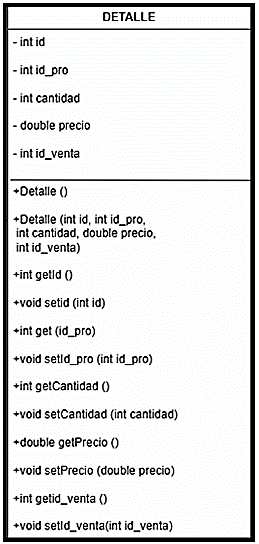
## 

## 3.3 Diagrama de clases

DESCRIB ALO DE FORMA GENERAL



***Figura 3.4. Diagrama de Clases. Fuente: Elaboración Propia***



**Clase CLIENTE:**

La clase CLIENTE representa a los usuarios que interactúan con el sistema, almacenando información relevante como el nombre, dirección, teléfono y correo electrónico. A través de sus métodos, permite crear, actualizar y obtener los datos del cliente, además de verificar y autenticar la información proporcionada. Esta clase se relaciona directamente con la clase VENTA, ya que un cliente puede estar asociado a múltiples ventas, estableciendo una relación de uno a muchos. Además, está relacionada con LOGIN, que maneja la autenticación de los clientes en el sistema.

**Clase LOGIN:**

La clase LOGIN gestiona la autenticación y seguridad de los usuarios del sistema. Contiene los atributos básicos de un login como nombre de usuario, contraseña, y el estado de la sesión. Entre sus métodos principales se encuentran las funciones de inicio de sesión, cierre de sesión, y recuperación de contraseña. Se relaciona directamente con la clase CLIENTE para validar y autenticar a los usuarios registrados, estableciendo una relación uno a uno, y su funcionamiento puede depender de configuraciones adicionales establecidas en la clase CONFIG.

**Clase PRODUCTOS:**

La clase PRODUCTOS almacena la información sobre los productos que están disponibles en el sistema. Esto incluye atributos como nombre, precio, cantidad y proveedor asociado. Sus métodos permiten añadir, actualizar y eliminar productos del inventario, así como obtener detalles de un producto en específico. Esta clase se relaciona con PROVEEDOR, ya que un proveedor suministra varios productos, estableciendo una relación de uno a muchos. También está vinculada con VENTA, donde los productos forman parte de las transacciones.

**Clase PROVEEDOR:**

La clase PROVEEDOR se encarga de almacenar los detalles de los proveedores, como nombre, dirección, teléfono y productos suministrados. Sus métodos permiten agregar, actualizar y eliminar información de los proveedores, además de consultar los productos que ofrecen. La relación con la clase PRODUCTOS es clave, ya que cada proveedor puede suministrar varios productos, estableciendo una relación de uno a muchos.

**Clase EVENTOS:**

La clase EVENTOS gestiona los eventos importantes dentro del sistema, como promociones, ofertas especiales o cambios en el inventario. Sus métodos permiten crear, actualizar y listar eventos, además de notificar a los usuarios cuando ocurre un evento significativo. Aunque no está directamente vinculada con otras clases, puede interactuar con PRODUCTOS para anunciar promociones o con CLIENTE para informar sobre ofertas.

**Clase COMBO:**

La clase COMBO representa una agrupación de productos o servicios ofrecidos como un conjunto a un precio especial o promocional. Sus atributos incluyen el nombre del combo, una descripción y el precio total. A través de sus métodos, permite crear, actualizar y eliminar combos, así como obtener detalles sobre los productos o servicios que lo componen. Aunque no está directamente conectada con otras clases, puede interactuar con PRODUCTOS para asociar productos a un combo y con VENTA para gestionar transacciones que incluyan la compra de un combo.

**Clase VENTA:**

La clase VENTA registra las transacciones de compra que se realizan en el sistema. Sus atributos incluyen detalles del cliente, fecha de la venta, el total de la transacción y el estado del pago. Está fuertemente relacionada con las clases CLIENTE y PRODUCTOS, ya que cada venta involucra a un cliente y uno o más productos. Tambien,está vinculada con DETALLE, que almacena información más específica de cada producto vendido en la transacción

**Clase DETALLE:**

La clase DETALLE almacena la información detallada de cada producto vendido dentro de una transacción de venta. Sus atributos incluyen la cantidad de productos, el precio unitario y los descuentos aplicados. Está relacionada con la clase VENTA, formando una relación de uno a muchos, donde una venta puede tener múltiples detalles asociados, cada uno representando un producto específico.

**CLASE CONFIG.**

La clase CONFIG maneja las configuraciones generales del sistema. Almacena parámetros clave como nombre, valor y descripción de configuraciones, permitiendo ajustar el comportamiento del sistema de forma dinámica sin modificar el código.Asimismo esta clase es para poder, configurar la impresión del progra

## 3.4. Diagrama De Entidad-Relación

DESCRIBALO DE FORMA GENERAL



***Figura 3.5. Modelo de base de datos. Fuente: Elaboración Propia***

## 3.5 Diseño Lógico Y Físico De La Base De Datos

**DISEÑO LÓGICO DE LA BASE DE DATOS:**

**1.Usuario:**

* **id\_usuario** (Primary Key)
* **nombre**
* **correo**
* **pass**
* **rol** (administrador, asistente)

**2.Proveedor:**

* **“Id\_proveedor”** (Primary Key)
* **“ruc”**
* **“nombre”**
* **“telefono”**
* **“ubigeo”** (Foreign Key)

**3.Productos:**

* **“id\_producto”** (Primary Key)
* **“código”**
* **“nombre”**
* **“id\_proveedor”** (Foreign Key)
* **“stock”**
* **“precio”**

**4.Ventas:**

* **“id\_venta”** (Primary Key)
* **“id\_cliente”** (Foreign Key)
* **“id\_vendedor”** (Foreign Key)
* **“total”**
* **“fecha”**

**5.Detalle\_venta:**

* **“id\_detalle”** (Primary Key)
* **“id\_producto”** (Foreign Key)
* **“cantidad”**
* **“precio”**
* **“id\_venta”** (Foreign Key)

**6.Clientes:**

* **“id\_cliente”** (Primary Key)
* **“id\_tipo\_documento”** (Foreign Key)
* **“nombre”**
* **“telefono”**
* **“ubicación** (Foreign Key)

**7.Configuración:**

* **id\_conf** (Primary Key)
* **ruc**
* **nombre**
* **telefono**
* **ubigeo** (Foreign Key)
* **mensaje**

**8.Tipo\_Documento:**

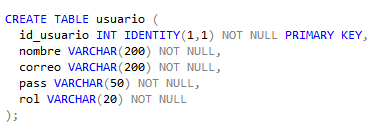
* **id\_tipo\_documento** (Primary Key)
* **descripción**

**9.Ubigeo:**

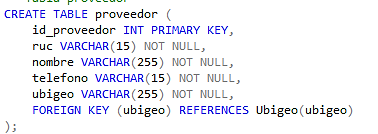
* **id\_ubigeo** (Primary Key)
* **ubigeo1**
* **dpto**
* **prov**
* **distrito**
* **ubigeo2**
* **orden**

**DISEÑO FÍSICO DE LA BASE DE DATOS:**

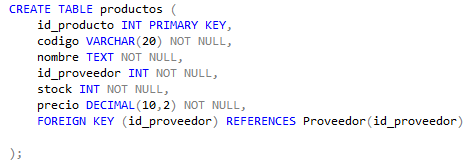
**1. Usuarios:** Función almacena los datos de los usuarios:



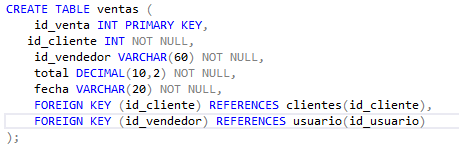
2. **Proveedor:** Función almacena los datos de los proveedores:

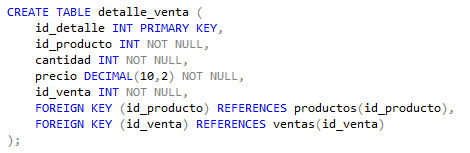


**3.Productos:** Almacena los datos de los productos:

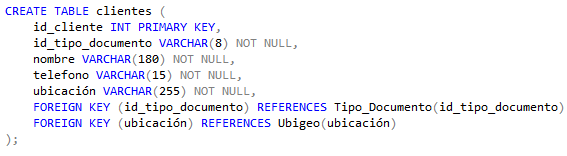


**4.Ventas**: Función información de la venta



**5. Detalle:** Función da el detalle de la venta

**6.Clientes:** Función almacena los datos de los clientes



## 3.6 Mockups (Prototipos UX/UI) y Documentación

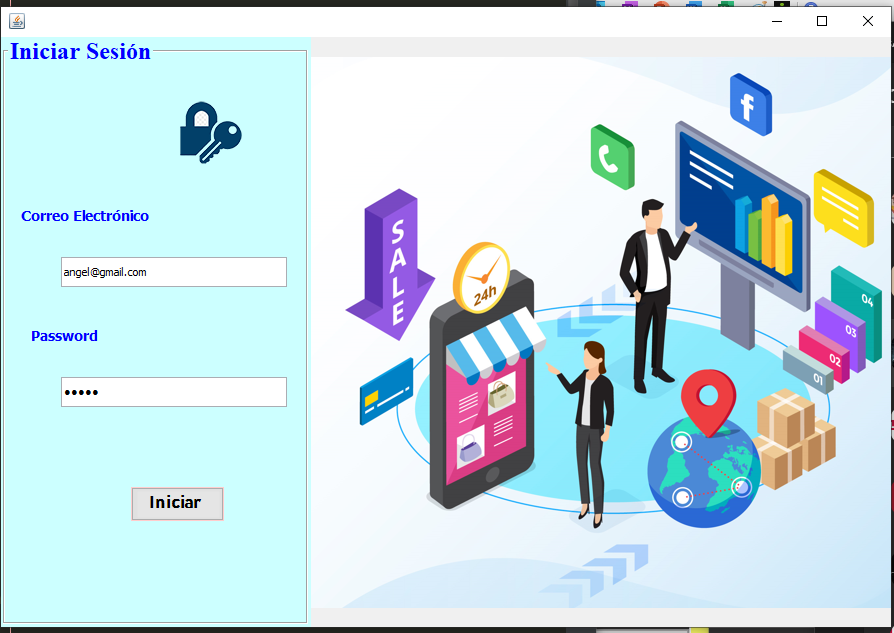
### 3.6.1. Mockups (Prototipos UX/UI)

Son representaciones visuales de las interfaces del diseño del software de “La casa del Seraphin Gas”. En ellos se muestran los elementos gráficos que los usuarios verán e interactuarán, como los botones, menús, cuadros de texto, etc.

**Pantallas Principales:**

1. **Interfaz Usuarios:**

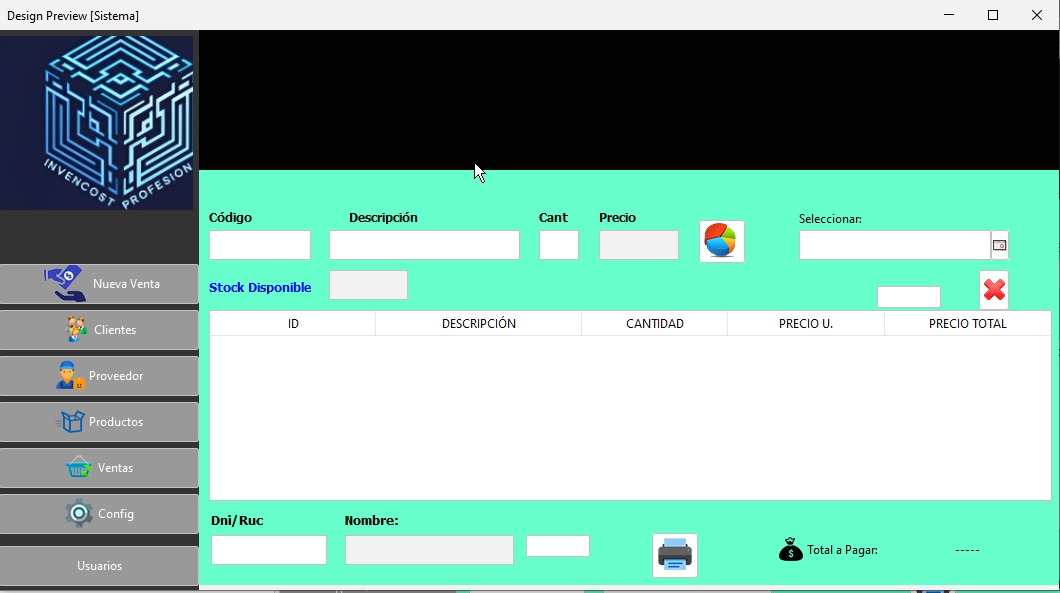
La interfaz de usuarios es en donde podemos iniciar sesión colocando los datos de nuestro usuario y contraseña, ya sea los empleados o el administrador.



***Figura 3.6. Inicio de Sesión. Fuente: Elaboración Propia***

1. **Interfaz Inicio de Nueva venta:**

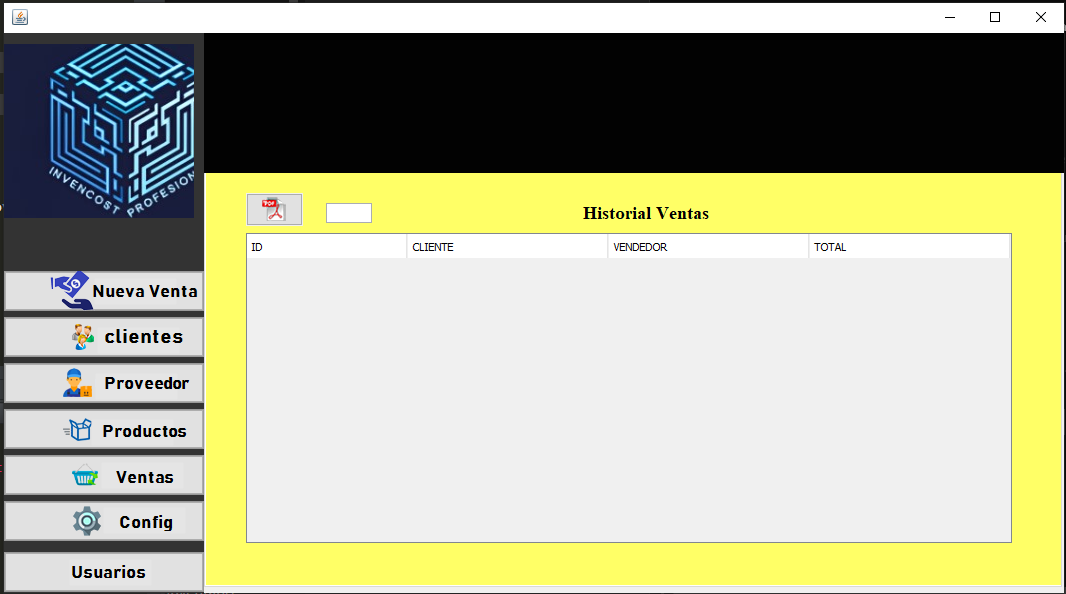
En esta interfaz de gestión de ventas de hidrocarburos, el usuario podrá realizar nuevas ventas de manera eficiente y organizada. El sistema permite ingresar el código del producto, su descripción y la cantidad deseada. Automáticamente, el sistema calculará y mostrará el precio total de la venta. Además, el sistema verifica en tiempo real si el producto está disponible en el inventario, mostrando el estado en una tabla donde también se reflejarán los productos seleccionados para la venta. El sistema también permite ingresar los datos del cliente, como el RUC o DNI y el nombre, facilitando la identificación del comprador. Finalmente, se generará el precio total que debe pagar el cliente, y se genera el reporte de la venta detallada.



***Figura 3.7. Interfaz de Inicio de Nueva Ventana. Fuente: Elaboración Propia***

1. **Interfaz Ventas:**

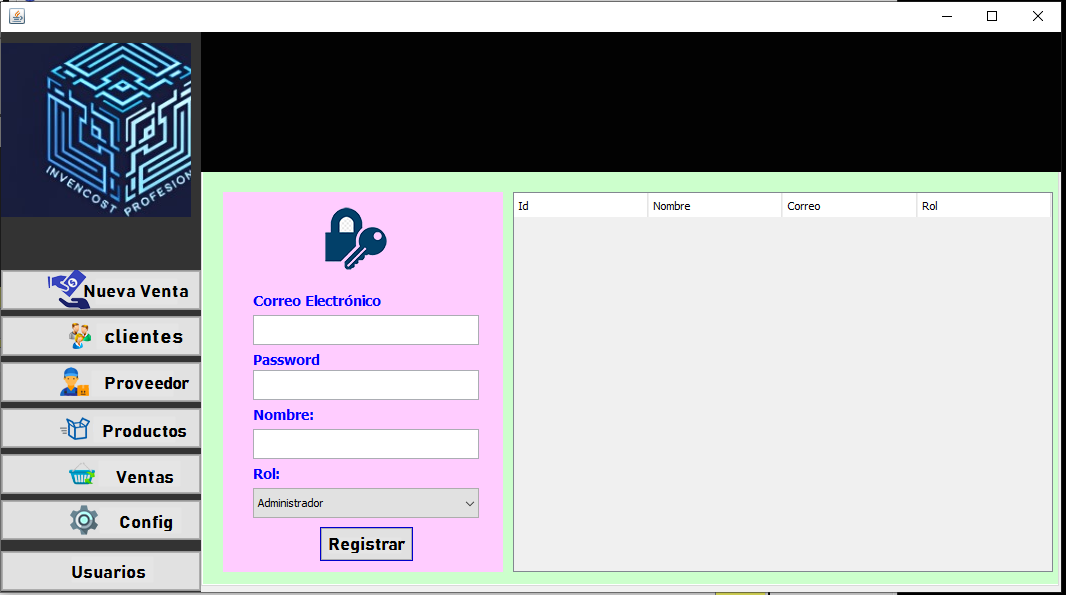
Esta interfaz muestra el historial completo de ventas, incluyendo la cantidad de transacciones realizadas. El sistema permite visualizar información detallada de cada venta, como los productos vendidos, el cliente, vendedor, y el total de pago por el producto. Además, el historial de ventas puede ser impreso fácilmente, generando un reporte detallado que incluye todas las transacciones realizadas, permitiendo un seguimiento preciso de las ventas y de quiénes las han efectuado.



***Figura 3.8. Interfaz de Ventas. Fuente: Elaboración Propia***

1. **Interfaz Usuarios:**

En la interfaz de usuario, se presenta una tabla de registro accesible únicamente para el administrador. Esta tabla permite al administrador gestionar los roles de los usuarios, asignando permisos específicos según las necesidades de la organización. A través de esta interfaz, el administrador puede crear, modificar o eliminar roles, garantizando que cada usuario tenga acceso únicamente a las funciones que le correspondan, mejorando la seguridad y control del sistema.



***Figura 3.9. Interfaz de Usuarios. Fuente: Elaboración Propia***

1. **Interfaz de Configuración**

La interfaz de configuración de la empresa está diseñada para permitir a los usuarios actualizar información esencial como RUC, nombre, dirección y teléfono de la empresa.



***Figura 3.10. Interfaz de Configuración Fuente: Elaboración Propia***

### 3.6.2. Documentación UX/UI

En la documentación se describe de manera detallada los aspectos sobre el diseño y experiencia del sistema

1. **Diseño UX**

Se enfoca como el sistema que satisface las necesidades de los usuarios de la empresa.

Contexto: El sistema de gestión de productos de hidrocarburos de **La Casa Seraphin Gas S.A.C.** se enfoca en resolver las necesidades de los empleados en la gestión de inventarios, ventas y cotizaciones. Actualmente, estos procesos se realizan de manera manual debido a que no cuenta con una base de datos para el almacenamiento de la información, lo que dificulta gestionar el control de clientes, ventas e inventarios. Esto obliga al personal a generar reportes de estos manualmente, lo que genera ineficiencias. El nuevo sistema busca mejorar el flujo de trabajo diario mediante una interfaz intuitiva y funcional que simplifica las tareas. La experiencia del usuario estará orientada a facilitar la gestión de productos, permitiendo agregar, editar y eliminar productos de forma rápida y precisa. Asimismo, la generación de cotizaciones será automatizada, lo que reducirá los tiempos de respuesta a los clientes y mejorará la exactitud de los precios y la información del producto. Al integrar un acceso rápido y directo a las funciones clave, el sistema minimizará errores operativos y aumentará la satisfacción tanto de los empleados como de los clientes. Por lo tanto, el diseño UX se busca garantizar que las tareas de los usuarios sean simples, eficientes y sin fricciones, mejorando así el rendimiento general de la empresa. El sistema de “la casa Seraphin Gas”.

1. **Diseño UI**

El diseño UI del sistema de gestión de productos de hidrocarburos de La Casa Seraphin Gas S.A.C. está orientado a crear una interfaz visualmente clara, intuitiva y fácil de usar. La prioridad es que los empleados puedan interactuar con el sistema de forma rápida y sin complicaciones, lo que mejora la eficiencia en la gestión diaria. El diseño incluirá menús organizados para acceder a funciones clave como la gestión de inventarios, la creación de cotizaciones y el control de ventas. Se utilizarán colores que diferencien claramente las distintas acciones (como agregar, editar o eliminar productos), y los botones y formularios estarán dispuestos de manera lógica para guiar al usuario en cada paso.

**1. Paleta de Colores**

* **Descripción**: Se usarán colores que transmitan confianza profesionalismo y estéticos para la empresa.

**Paleta De Colores**:

* **Gris claro (RGB (239, 239, 239))**: Se utiliza en el fondo de las tablas y algunos campos de texto donde se muestran los detalles de los productos (ID, descripción, cantidad, etc.).
* **Negro (RGB (2, 2, 2))**: Es el color del texto principal y del fondo, como las etiquetas de los campos "Código", "Descripción", "Precio", "Stock Disponible.
* **Verde menta (RGB (102, 252, 203))**: Este es el color predominante de fondo de la interfaz de nueva Venta, ocupando gran parte del área de trabajo donde se ingresan y visualizan los datos.
* **Verde azulado (RGB (84, 141, 143))**: Se utiliza en algunos íconos y detalles menores para crear contraste y resaltar ciertos elementos, como los botones de la barra lateral.
* **Menta pálida (RGB (173, 223, 207))**: Este tono se aplica en ciertas áreas de relleno en la interfaz, como los fondos de algunos campos, para dar una apariencia más suave y diferenciada.
* **Azul gris oscuro (RGB (49, 60, 77))**: Este color se usa en el panel lateral izquierdo, donde se encuentran los íconos y botones de navegación (como "Nueva Venta", "Clientes", "Proveedor", etc.), creando contraste con el resto de la interfaz.

**2. Tipografía**

* **Títulos:** Tipografía: Roboto Bold  
   Tamaño: 24px  
   Propósito: Se utilizó para los títulos de las secciones y encabezados, brindando un sentido de modernidad y profesionalismo.
* **Texto Principal:** Tipografía: Arial Regular  
   Tamaño: 16px  
   Propósito: Esta tipografía se eligió por su claridad y familiaridad. Es ideal para el texto principal ya que facilita la lectura, haciendo que la información sea accesible y fácil de entender.

**3. Iconografía y Botones**

**Diseño de Botones:**  
 Los botones serán diseñados con un enfoque en la simplicidad y la claridad. Tienen bordes redondeados y un fondo que cambia de color al pasar el mouse (hover), lo que indica que son interactivos. La elección de colores es consistente con la paleta de la interfaz, lo que refuerza la identidad visual de la aplicación.

### 3.6.3. Identificación De Reportes

La identificación de reportes tiene como objetivo principal detectar los problemas específicos de la empresa y proponer soluciones a través de reportes clave. Estos reportes son herramientas que ayudarán a resolver las dificultades identificadas.

**Problema de la empresa:** Seraphin Gas enfrentaba dificultades para llevar un control eficiente del inventario de materiales y equipos, lo que comprometía la gestión de suministros. Además, la generación de cotizaciones rápidas y precisas era un desafío, afectando la capacidad de respuesta a las solicitudes comerciales.

**Solución propuesta:**

* **Reporte de inventario operativo:** Se implementará un informe diario que mostrará el estado del inventario, destacando los productos críticos con bajo stock. Esto permitirá a la gerencia tomar decisiones rápidas sobre reabastecimiento y prevenir desabastecimientos.
* **Reporte de cotización operativo:** Se desarrollará un sistema que generará cotizaciones automáticas basadas en los costos actuales de productos y servicios, ajustando los precios en tiempo real según las fluctuaciones del mercado.

**Gráfico Sugerido:**

Gráficos de barras y circulares que muestran cómo los datos de inventario y cotización se recopilan en tiempo real y se procesan para generar los reportes operativos, ilustrando el flujo de información y su impacto en la toma de decisiones diaria.

**MAPEO DE REPORTES CLAVE PARA EL PROYECTO**

El mapeo de reportes organiza y categoriza los reportes que ya se han identificado, pero relacionándolos con objetivos de la empresa y su impacto en la toma de decisiones.

**Objetivos:**

1. Mantener un control detallado y en tiempo real del inventario.
2. Mejorar la eficiencia en la generación de cotizaciones para reducir los tiempos de respuesta a los clientes.
3. Predecir necesidades de inventario basadas en el consumo histórico y ajustar precios de forma competitiva.

**Mapeo de reportes:**

* **Reporte de inventario operativo:** Un listado diario de productos con bajo stock, lo que ayudará a prevenir desabastecimientos y optimizar el reabastecimiento.
* **Reporte de cotización operativo:** Cotizaciones automáticas generadas en función de los precios actuales del mercado, con capacidad para ajustarse dinámicamente ante fluctuaciones en los costos.

**Gráfico Sugerido:**

Gráficos de barras y circulares que representan cómo los reportes de inventario y cotización se integran en las decisiones inmediatas, mensuales y estratégicas, facilitando una toma de decisiones informada y precisa.

## 3.7. Diagramas de Arquitectura (estructura MVC, implementación de patrones SOLID e integración de capas DAO)

En la arquitectura Modelo-Vista-Controlador (MVC), cada componente tiene diferentes responsabilidades, para organizar y estructurar la arquitectura del sistema. En esta interfaz de gestión de ventas de hidrocarburos, este sistema ha sido organizado siguiendo la estructura del Modelo-Vista-Controlador:

Modelo: El modelo contiene la lógica del sistema, para poder generar los reportes.

El modelo maneja las indentidades como el cliente, que se encarga de actualizar información de cliente las subrayadas son las indentidades debido a que estas hacen operaciones como agregar, modificar y eliminar

* Cliente.java: Modelo que se encarga de representar la información del cliente.
* Combo.java: Modelo que representa el nombre del producto con su ID.
* Config.java: Modelo que almacena la configuración de una entidad.
* Detalle.java: Modelo que representa los detalles de un producto de una venta.
* Productos.java: Modelo que representa un producto en el sistema.
* Proveedor.java: Modelo que representa a un proveedor del sistema.
* Venta.java: Modelo que almacena la transacción de venta del sistema.
* login.java: Modelo que representan los datos de autenticación de un usuario del sistema.
* Conexión.java: Establece una conexión con la base de datos. (controlador)

Vista: Es la parte visual de la aplicación (front-end). Estos elementos pueden permitir al usuario interactuar con la aplicación. Que resultado voy a ver del modelo

* Login.java: Vista que contiene la interfaz de inicio de sesión.
* Sistema.java: Vista principal del sistema.

Controlador: Este actúa como intermediario entre el modelo y la vista, recibe órdenes del usuario y solicita los datos del modelo. Las funciones de modificar son intermediario con la base de datos

* ClienteDao.java: Gestiona las operaciones de acceso con el modelo “Cliente”.
* Eventos.java: Maneja eventos para controlar la entrada de datos en los campos de texto.
* LoginDAO.java: Gestiona las operaciones de acceso y la manipulación de datos de usuarios.
* ProductosDao.java: Implementa el acceso de datos de la base de datos.
* Gráfico.java: Genera un gráfico que representa los totales de ventas por fecha determinada. (vista)
* ProveedorDao.java: Maneja operaciones de acceso con el modelo “Productos”.
* VentaDao.java: Gestiona operaciones relacionadas con el modelo “Ventas”:

Avanzar el DAO, TDD Y las consideraciones de seguridad se puede hacer gráficamente como si fuera un diagrama de clase.

## 3.8 Funcionalidades del código fuente

# CAPITULO 4-RESULTADOS

## CONCLUSIONES

## RECOMENDACIONES

# BIBLIOGRAFÍAS

Calderon Lucero, C.E. (2023). *Implementación de un sistema web para el control del almacén de congelados de la empresa pesquera . - Hayduk* [Tesis de licenciatura, Universidad Católica Los Ángeles de Chimbote]. Repositorio Institucional de la Universidad Católica Los Ángeles de Chimbote. Recuperado de

<https://repositorio.uladech.edu.pe/handle/20.500.13032/34519>

Yarín Achachagua, Y. H. (2017). *Diseño e implementación de un sistema de localización y control de inventarios en un almacén de aduanas, utilizando tecnología RFID* [Tesis de maestría, Universidad Nacional Mayor de San Marcos]. Repositorio de la universidad.

[~~tesis9.pdf (unmsm.edu.pe)~~](https://eventosindustrial.unmsm.edu.pe/upg_old/archivos/TESIS2018/MAESTRIA/tesis9.pdf)

Fernandez, Y. B. (2020). *Introducción a Programación Orientada a Objetos*.

*Moreno Pérez, J. C. (2015). Programación orientada a objetos: ( ed.). Madrid, RA-MA Editorial.*

Ruiz Rodríguez, R. (2009). Fundamentos de la programación orientada a objetos: una aplicación a las estructuras de datos en Java: ( ed.). Miami, Argentina: El Cid Editor.

# ANEXOS

## Lean Canvas

**DESCRIPCIÓN DEL PROYECTO:**

Nuestro proyecto consiste en desarrollar un sistema de gestión y manejo de productos además de la creación y automatización de cotizaciones y ventas para la empresa Seraphin.

**OBJETIVOS DEL PROYECTO:**

* Facilitar la gestión de inventario de los productos de la empresa SERAPHIN.
* Automatizar las cotizaciones de los productos y servicios que ofrece la empresa SERAPHIN.

**COMPONENTES DEL LEAN CANVAN:**

1. SEGMENTOS CLIENTES:

* Pequeñas empresas
* Importadores y exportadores con almacén
* Minoristas

Early adopters: Empresas que se quieren organizar: Dirigido específicamente a pequeñas empresas como “La Casa de Seraphin Gas” que buscan mejorar su gestión de inventario y las cotizaciones de sus productos y servicios.

1. PROBLEMA

* Descoordinación en el manejo de inventario de productos en la empresa Seraphin.
* Retrasos en la cotización de servicios y productos en la empresa Seraphin.

1. PROPUESTA DE VALOR ÚNICA

Gestión de inventarios y cotizaciones automatizadas en una plataforma centralizada para un control eficiente de tus operaciones.

1. SOLUCIÓN

* El software llevará registro de los productos que entran y salen en tiempo real todo ese movimiento que quedará registrado en una base de datos.
* El software permitirá generar cotizaciones de manera automática ya que estará conectada con el inventario de la empresa o con los empleados disponibles para un servicio.

1. CANALES

Los canales que utilizaremos para la venta de nuestro software será por página web, Marketing gratuito (TikTok, Instagram) y otras pequeñas empresas.

1. FLUJO DE INGRESOS

El flujo de ingresos que tendremos para nuestro proyecto será a través de las ventas de licencias, dando pruebas gratuitas por tiempos limitados, comisiones por integración de proveedores.

1. ESTRUCTURA DE COSTES

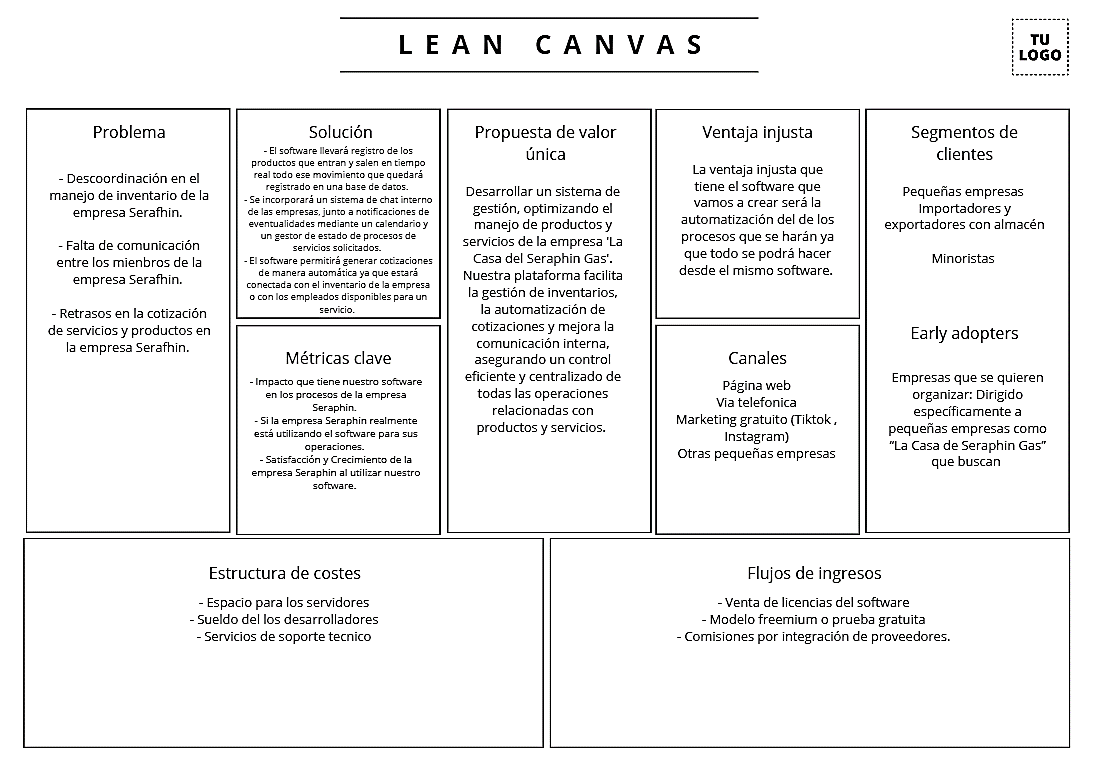
La estructura de costes que tenemos para la creación de nuestro software sería el espacio para los servidores, los sueldos de nuestros desarrolladores y servicios de soporte técnico.

1. MÉTRICA CLAVES

* Impacto que tiene nuestro software en los procesos de la empresa Seraphin mediante reportes de las empresas.
* Si la empresa Seraphin realmente está utilizando el software para sus operaciones. Mediante indicadores de rendimiento como la optimización de procesos, reducción de tiempos de operación y el incremento en la productividad de la empresa.
* Satisfacción y Crecimiento de la empresa Seraphin al utilizar nuestro software mediante encuestas de satisfacción.

1. VENTAJA INJUSTA

La ventaja que tiene el software que vamos a crear será la automatización del de los procesos que se harán ya que todo se podrá hacer desde el mismo software.



## EL PMBOK (PROJECT MANAGEMENT BODY OF KNOWLEDGE)

### 1. INICIO DEL PROYECTO

#### Desarrollar el Acta de Constitución del Proyecto:

* Propósito del Proyecto: Justificar la necesidad del sistema, como la necesidad de mejorar la eficiencia operativa y el control de inventarios.
* Objetivo General:
* Desarrollar un sistema de gestión de productos de hidrocarburos y servicios para la empresa La Casa Seraphin Gas S.A.C.
* Objetivos Específicos:
  + - Mejorar la gestión del inventario de la empresa Seraphin.
    - Mejorar la elaboración de las cotizaciones de la empresa Seraphin
* Beneficios Esperados:
  + Reducción de costos operativos.
  + Mejora en la satisfacción del cliente.
  + Cumplimiento de las normativas de seguridad y calidad.
* Patrocinador del Proyecto: Identificar a la persona o entidad que autoriza y apoya el proyecto, como un ejecutivo senior de la empresa.

#### Identificar a los Interesados:

* Interesados Claves:
  + Interno:
    - Personal de ventas y atención al cliente.
    - Personal de logística y almacén.
    - Equipo de IT y soporte técnico.
  + Externo:
    - Proveedores de hidrocarburos.
    - Clientes.
    - ~~Autoridades regulatorias.~~
* Análisis de Interesados:
  + Expectativas y Necesidades: Comprender qué espera cada grupo del sistema.
  + Impacto e Influencia: Evaluar cómo cada interesado puede afectar o ser afectado por el proyecto.

### 2. PLANIFICACIÓN DEL PROYECTO

#### Planificar el Alcance:

* Declaración del Alcance:
  + Alcance del Proyecto: Definir las funcionalidades del sistema, como la gestión de inventarios, ventas, compras, y reportes.
  + Entregables Principales:
    - Módulo de gestión de inventarios.
    - Módulo de facturación.
    - Módulo de gestión de pedidos y distribución.
    - Reportes de desempeño y auditoría.
* Requisitos del Sistema:
  + Requisitos Funcionales: ¿Qué debe hacer el sistema?

El sistema debe tener la capacidad de gestionar el inventario y generar cotizaciones de los productos o servicios.

* + Requisitos No Funcionales: ¿Cómo debe comportarse el sistema? (rendimiento, seguridad, usabilidad).

El sistema debe ser ligero para las computadoras de oficina.

#### Desarrollar el Cronograma:

* Definir Actividades: Crear una lista detallada de todas las tareas necesarias, como la instalación del sistema, pruebas, y capacitación.
* Secuenciar Actividades: Organizar las tareas en el orden en que deben completarse.
* Estimar Duración: Evaluar cuánto tiempo tomará completar cada tarea y crear un cronograma detallado.
* Herramientas: Utilizar herramientas de gestión de proyectos como Microsoft Project o herramientas en línea como Asana o Trello.

#### Planificar los Recursos:

* Recursos Humanos:
  + Roles y Responsabilidades: Definir quién hará qué, como el líder del proyecto, analistas, desarrolladores, y personal de soporte.
  + Equipo de Proyecto: Asegurarse de contar con personal adecuado para cada fase del proyecto.
* Recursos Materiales:
  + Hardware y Software: Servidores, estaciones de trabajo, licencias de software.
  + Infraestructura: Espacio para el servidor y sistemas de respaldo.

#### Planificar el Presupuesto:

* Estimación de Costos:
  + Costos Directos: Costos del software, hardware, y salarios.
  + Costos Indirectos: Costos de capacitación, consultoría externa, y otros gastos asociados.
* Presupuesto Detallado: Crear un presupuesto detallado con categorías específicas y asignar fondos.

#### Planificar la Gestión de Riesgos:

* Identificación de Riesgos:
  + Posibles riesgos: Problemas técnicos, resistencia al cambio, retrasos en el desarrollo.
* Análisis de Riesgos:
  + Probabilidad e Impacto: Evaluar la probabilidad de cada riesgo y su impacto potencial.
* Estrategias de Respuesta:
  + Mitigación: Planes para reducir la probabilidad o impacto de los riesgos.
  + Contingencia: Planes alternativos si los riesgos se materializan.

#### Planificar la Gestión de la Calidad:

* Criterios de Calidad:
  + Normas y Estándares: Definir los estándares que el sistema debe cumplir, como normas de calidad de software y regulaciones del sector.
* Control de Calidad:
  + Métodos de Evaluación: Pruebas de funcionalidad, revisiones de código, y validaciones.

### 3. EJECUCIÓN

* Realización del proyecto

Llevar a cabo las tareas descritas en el plan, asegurando que se cumplan los requisitos y entregables del proyecto.

* Gestión de la ejecución del proyecto

Asegurar que el trabajo esté alineado respecto a las necesidades de la empresa y que esté de acuerdo con el plan del proyecto.

* Gestión del Equipo:
* Dirigir y Motivar al Equipo: Coordinar y liderar al equipo para que cumpla con sus responsabilidades de manera eficiente.
* Gestión de Recursos Humanos: Asegurar que los roles y responsabilidades estén claros, y brindar el soporte necesario al equipo.
* Gestión de calidad
* Control de calidad: Realizar revisiones de manera continua para asegurar que cumplen con los estándares de calidad establecidos.
* Implementación de pruebas: Implementar las pruebas debidas para garantizar que el proyecto final sea el adecuado.
* Gestión de las comunicaciones
* Distribución de la información**:** Mantener a todos los interesados informados sobre el progreso del proyecto, utilizando canales de comunicación efectivos.
* Feedback: Recoger y analizar las retroalimentaciones de los interesados para implementar las mejoras debidas.
* Gestión de adquisiciones
* Relación con los proveedores: Mantener una comunicación con los proveedores para poder resolver problemas o inquietudes, asegurando así que todos estemos alineados.
* Supervisión y cumplimiento de contratos: Asegurarse de que los proveedores y contratistas cumplan con los términos y condiciones establecidos en los contratos a la hora de usar el sistema.

### 4. MONITOREO Y CONTROL

* Desempeño del proyecto
* Análisis de variaciones: Evaluar las diferencias entre los resultados planificados y los reales, identificando las causas y estableciendo las mejoras debidas.
* Revisiones: Comparar el progreso actual del proyecto con el cronograma, presupuesto y alcance definidos, identificando los posibles atrasos.
* Gestión de cambios
* Impacto de cambios: Analizar cautelosamente los cambios hechos y si dan un gran impacto en el cronograma, calidad, entre otros y ajustar los planes si es necesario.
* Verificación de alcance
* Confirmación de cierre del proyecto: Validar formalmente que no hay tareas pendientes relacionadas con la planificación y que se han cumplido todos los objetivos definidos al comienzo del proyecto.
* Aprobaciones finales: Obtener las aprobaciones finales de los interesados que confirmen que el proyecto ha sido aceptado y cumple con los estándares.
* Gestión de tiempos y costos
* Informe de estado: Preparar informes sobre el estado del cronograma y los costos, mostrando a todos los interesados sobre el avance del proyecto.
* Monitoreo de tiempos y presupuestos: Controlar que el proyecto se mantenga dentro de los plazos y costos establecidos, realizando ajustes si se detectan desviaciones.
* Control de calidad
* Revisiones finales de calidad: Asegurar que el proyecto final cumpla con los estándares establecidos y si es necesario realizar las últimas mejoras.
* Pruebas: Realizar pruebas del proyecto finalizado para poder validar que el proyecto es el adecuado para los interesados.

### 5. CIERRE DEL PROYECTO

* Finalización del Proyecto:
* Entrega del software: Realizar la entrega del sistema de gestión de productos de hidrocarburos y servicios a los usuarios finales. Asegurarse de que todas las funcionalidades estén en óptimas condiciones y cumplan con los requisitos definidos.
* Capacitación final: Asegurar que los usuarios y el personal de soporte tengan completamente la capacitación en el uso y mantenimiento del sistema.
* Documentación del proyecto: Compilar y entregar toda la documentación del proyecto, incluidos manuales de usuario, documentación técnica y reportes finales.
* Evaluación del Proyecto:
* Revisión Post-Implementación: Evaluar el desempeño del proyecto en relación con los objetivos iniciales, identificando aciertos y áreas de mejora. Esto incluye la revisión del cumplimiento de los cronogramas, presupuestos y calidad.
* Encuestas de satisfacción: Recoger retroalimentación del cliente para evaluar su nivel de satisfacción con el proyecto y el sistema implementado.
* Cierre Administrativo:
* Informe final: Preparar un informe final del proyecto que resuma todo el trabajo realizado y los resultados obtenidos.
* Cierre formal: Obtener la aprobación formal del cliente y otros interesados clave para declarar oficialmente el proyecto cerrado.

## Anexo: PLANIFICACIÓN DEL PROYECTO

### PROJECT CHARTER (ACTA DE CONSTITUCIÓN DEL PROYECTO)

* Nombre del Proyecto: Sistema De Gestión De Productos De Hidrocarburos Y Servicios De La Empresa La Casa Seraphin Gas
* Fecha de Inicio: 12 de agosto de 2024
* Fecha de Finalización Estimada: 10 de diciembre de 2024
* Propósito del Proyecto:

Desarrollar un sistema que optimice la administración del inventario y automatice la creación de cotizaciones, mejorando así la eficiencia operativa de La Casa Seraphin Gas.

* Objetivos del Proyecto:
* Facilitar la gestión de inventario de los productos de la empresa SERAPHIN.
* Automatizar las cotizaciones de los productos y servicios que ofrece la empresa SERAPHIN.
* Alcance del Proyecto:

El proyecto incluye el desarrollo de un módulo de gestión de inventario, facilitando la entrada, salida y seguimiento de productos en tiempo real, con generación de reportes de stock y alertas de inventario bajo. También se implementará un sistema de automatización en la creación de cotizaciones, generando cotizaciones automáticas basadas en la información del inventario y permitiendo su personalización para diferentes tipos de clientes e integrando con bases de datos existentes.

* Equipo del Proyecto:

Project Manager: ROJAS BARBARAN, Eberth Gianfranco

Desarrollador Líder: JAHUAR CHIRINOS, Anibal Alejandro

Diseñador UX/UI: ROMERO RIVAS, Romina Ariadna

Arquitecto de software: UIPAN MANTARI, Renato Alonso

Administrador de Base de Datos: TASAYCO MAGALLANES, Giordano Martin

* Partes Interesadas:
* La casa de gas Seraphin
* Los empleados de la casa seraphin
* Los clientes potenciales
* Departamento de ventas y logística de la empresa

### Anexo: WORK BREAKDOWN STRUCTURE (WBS – ESTRUCTURA DE DESGLOSE DEL TRABAJO)

### 1. Inicio del Proyecto

* **1.1 Definición del Proyecto**
  + 1.1.1 Objetivos del Proyecto
  + 1.1.2 Alcance del Proyecto
* **1.2 Identificación de Stakeholders**
  + 1.2.1 Identificación de Interesados
  + 1.2.2 Estrategia de Comunicación
* **1.3 Aprobación del Proyecto**
  + 1.3.1 Estimación de Costos
  + 1.3.2 Aprobación del Presupuesto

### 2. Planificación del Proyecto

* **2.1 Análisis de Requerimientos**
  + 2.1.1 Recolección de Requerimientos
  + 2.1.2 Análisis de Viabilidad
* **2.2 Diseño del Sistema**
  + 2.2.1 Arquitectura del Sistema
  + 2.2.2 Diseño de Base de Datos
  + 2.2.3 Diseño de Interfaz de Usuario (UI)
* **2.3 Planificación del Desarrollo**
  + 2.3.1 Cronograma de Desarrollo
  + 2.3.2 Asignación de Recursos

### 3. Desarrollo del Sistema

* **3.1 Desarrollo del Backend**
  + 3.1.1 Implementación de la Lógica de Negocio
  + 3.1.2 Conexión con la Base de Datos
  + 3.1.3 Desarrollo de Servicios de API
* **3.2 Desarrollo del Frontend**
  + 3.2.1 Desarrollo de la Interfaz Gráfica de Usuario (GUI)
  + 3.2.2 Integración con el Backend
* **3.3 Pruebas del Sistema**
  + 3.3.1 Pruebas Unitarias
  + 3.3.2 Pruebas de Integración
  + 3.3.3 Pruebas de Usuario

### 4. Monitoreo y Control

* **4.1 Monitoreo de Progreso**
  + 4.1.1 Seguimiento de Tareas
  + 4.1.2 Reportes de Avance
* **4.2 Gestión de Cambios**
  + 4.2.1 Registro de Cambios
  + 4.2.2 Aprobación e implementación de cambios
* **4.3 Monitoreo de Calidad**
  + 4.3.1 Evaluación de Cumplimiento de Requisitos
  + 4.3.2 Pruebas de Calidad Finales

### 5. Cierre del Proyecto

* **5.1 Implementación y Despliegue**
  + 5.1.1 Configuración en Entorno de Producción
  + 5.1.2 Migración de Datos
* **5.2 Capacitación y Transferencia**
  + 5.2.1 Capacitación del Personal
  + 5.2.2 Transferencia de Conocimientos
* **5.3 Cierre Administrativo**
  + 5.3.1 Documentación Final
  + 5.3.2 Informe de Cierre y Lecciones Aprendidas

### Anexo: GANTT CHART

