



**UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería en Informática y Sistemas**

**Informe final**

**Proyecto: Sistema de gestión de Minimarket**

**Curso: Programación avanzada**

**Docente: MSc. Ing. Hugo Manuel Barraza Vizcarra**

**Integrantes:**

**Anghelo del Piero Calderon Morales 2025-119049**

**Samuel Alexander Gomez Salcedo 2025-119052**

**Lucía de los Ángeles Angulo Y 2025-119025**

**Tacna – Perú**

**2025**

## ÍNDICE GENERAL

<b>RESUMEN</b>	<b>4</b>
<b>ANTECEDENTES</b>	<b>5</b>
<b>PLANTEAMIENTO DEL PROBLEMA</b>	<b>5</b>
a. Problema	5
b. Justificación	5
c. Alcance	6
<b>OBJETIVOS</b>	<b>6</b>
a. General	6
b. Objetivos específicos	6
<b>MARCO TEÓRICO</b>	<b>7</b>
<b>DESARROLLO DE LA PROPUESTA</b>	<b>7</b>
a. Tecnologías de desarrollo	7
b. Metodología de implementación	7
c. Visión	7
d. Descripción de los interesados y usuarios	7
e. Necesidades de los interesados y usuarios	8
<b>ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE</b>	<b>8</b>
a. Requerimientos funcionales	8
b. Requerimientos funcionales	8
<b>ARQUITECTURA DE SOFTWARE</b>	<b>9</b>
a. Vista de datos	9
b. Módulos	10
<b>CONCLUSIONES Y RECOMENDACIONES</b>	<b>10</b>
<b>BIBLIOGRAFÍA</b>	<b>10</b>
<b>ANEXOS</b>	<b>10</b>

## **RESUMEN**

El presente proyecto consiste en el desarrollo de un Sistema de Inventario para un Minimarket utilizando el lenguaje de programación C++ bajo el paradigma de la Programación Orientada a Objetos (POO). El sistema fue diseñado con el objetivo de gestionar de manera ordenada y eficiente la información relacionada con clientes, productos y ventas, permitiendo automatizar procesos que comúnmente se realizan de forma manual.

El sistema incorpora funcionalidades como el registro de clientes y productos, control de stock, registro de ventas, generación de reportes en archivos de texto, validaciones para evitar errores de ingreso de datos, así como algoritmos básicos de búsqueda y ordenación. Además, se incluye una advertencia de stock crítico que permite alertar al usuario cuando la cantidad de un producto es baja.

Este proyecto no solo busca resolver una problemática real de gestión en pequeños negocios, sino también aplicar y reforzar conceptos fundamentales de la Programación Orientada a Objetos, tales como encapsulamiento, herencia y reutilización de código, cumpliendo con los requisitos académicos establecidos.

## **ANTECEDENTES**

El comercio minorista, en su forma más básica, ha existido durante siglos como un pilar fundamental de la economía. En la era moderna, la gestión eficiente de un minimarket requiere más que solo transacciones manuales, necesita sistemas robustos que manejan inventario, ventas e integrar clientes y empleados. Históricamente, estos procesos se manejan con sistemas rudimentarios. Sin embargo, el avance de la tecnología ha permitido la transición hacia soluciones digitales. El desarrollo de software basado en el paradigma de Programación Orientada a Objetos (POO) en lenguajes como C++ ofrece una estructura modular y escalable ideal para modelar entidades complejas del mundo real, como productos, clientes y transacciones, facilitando el mantenimiento y la expansión futura del sistema de gestión. Este proyecto se enmarca en la necesidad de digitalizar y optimizar las operaciones de pequeños comercios utilizando herramientas de programación avanzadas.

## **PLANTEAMIENTO DEL PROBLEMA**

### **a. Problema**

En las pequeñas empresas minoristas, la administración manual del inventario y el registro de ventas frecuentemente conduce a errores humanos, pérdidas de mercancía por descontrol, lentitud en el servicio al cliente y dificultad para tomar decisiones informadas sobre las compras. La falta de un sistema automatizado impide conocer en tiempo real el estado del stock, gestionar el inventario y la rentabilidad de los productos.

¿Cómo se puede diseñar e implementar un sistema de gestión de inventario y ventas para un minimarket utilizando el paradigma de Programación Orientada a Objetos (POO) en C++, que permita registrar productos, gestionar el inventario, buscar productos por ID e integrar ventas, clientes y empleados para mejorar la eficiencia operativa del negocio?

### **b. Justificación**

La implementación de un sistema desarrollado bajo los principios de POO para la gestión de un minimarket se justifica por dos motivos generales. Inicialmente, desde una perspectiva técnica, el uso de C++ y POO permite modelar las relaciones complejas entre los datos (productos, precios, proveedores) y las operaciones (venta, compra) de manera intuitiva y

robusta, utilizando conceptos como encapsulamiento, herencia y polimorfismo. En segundo lugar, el sistema proporcionará reportes claros y accesibles que facilitarán la toma de decisiones gerenciales, contribuyendo directamente a la rentabilidad y sostenibilidad del negocio.

### **c. Alcance**

El alcance de este proyecto se limita al desarrollo de una aplicación de escritorio en C++ que implementa las funcionalidades esenciales para la operación diaria de un minimarket. Esto incluye:

- Módulo para integrar ventas, clientes y producto.
- Módulo de gestión de productos (registro de productos, inventario, ...).
- Módulo de registro de ventas (cálculo de totales, agregar productos, ...).

## **OBJETIVOS**

### **a. General**

Diseñar, desarrollar e implementar un sistema informático funcional para la administración de inventario y ventas de un minimarket, aplicando los principios de la Programación Orientada a Objetos (POO) en el lenguaje C++.

### **b. Objetivos específicos**

- Diseñar una arquitectura basada en clases aplicando encapsulamiento y herencia.
- Implementar algoritmos de búsqueda para localizar productos y clientes.
- Aplicar un algoritmo de ordenación para organizar los productos por precio.
- Incorporar validaciones que eviten errores de ingreso de datos.
- Utilizar archivos de texto para generar reportes de ventas.

## MARCO TEÓRICO

La Programación Orientada a Objetos es un paradigma de programación que organiza el software en clases y objetos, facilitando la reutilización del código, la modularidad y el mantenimiento de los sistemas. En este proyecto se utiliza POO para modelar entidades reales como clientes, productos y ventas. El encapsulamiento consiste en restringir el acceso directo a los atributos de una clase, permitiendo su manipulación únicamente a través de métodos públicos. Esto protege la integridad de los datos y evita modificaciones indebidas. La herencia permite que una clase hija reutilice los atributos y métodos de una clase base. En el sistema desarrollado, la clase Cliente hereda de la clase Persona, reutilizando información común. Se utiliza la búsqueda lineal como algoritmo para localizar elementos dentro de los arreglos, permitiendo encontrar clientes o productos a partir de su identificador. El algoritmo de ordenación burbuja se emplea para ordenar los productos según su precio, facilitando la visualización ordenada de la información. Los archivos de texto se utilizan para almacenar reportes de ventas, permitiendo la persistencia de información y la revisión posterior de las operaciones realizadas.

## DESARROLLO DE LA PROPUESTA

### a. Tecnologías de desarrollo

- Lenguaje: C++
- Compilador: Embarcadero DEV C++
- Paradigma: Programación Orientada a Objetos

### b. Metodología de implementación

Se utilizó una metodología incremental, desarrollando primero las clases base y luego integrando funcionalidades como ventas, validaciones, búsqueda y ordenación.

### c. Visión

El sistema busca ser una solución simple, funcional y extensible para la gestión básica de un minimarket.

### d. Descripción de los interesados y usuarios

- Dueño del minimarket
- Empleado encargado del registro de ventas

- e. Necesidades de los interesados y usuarios
  - Control del stock
  - Registro confiable de ventas
  - Información clara y ordenada

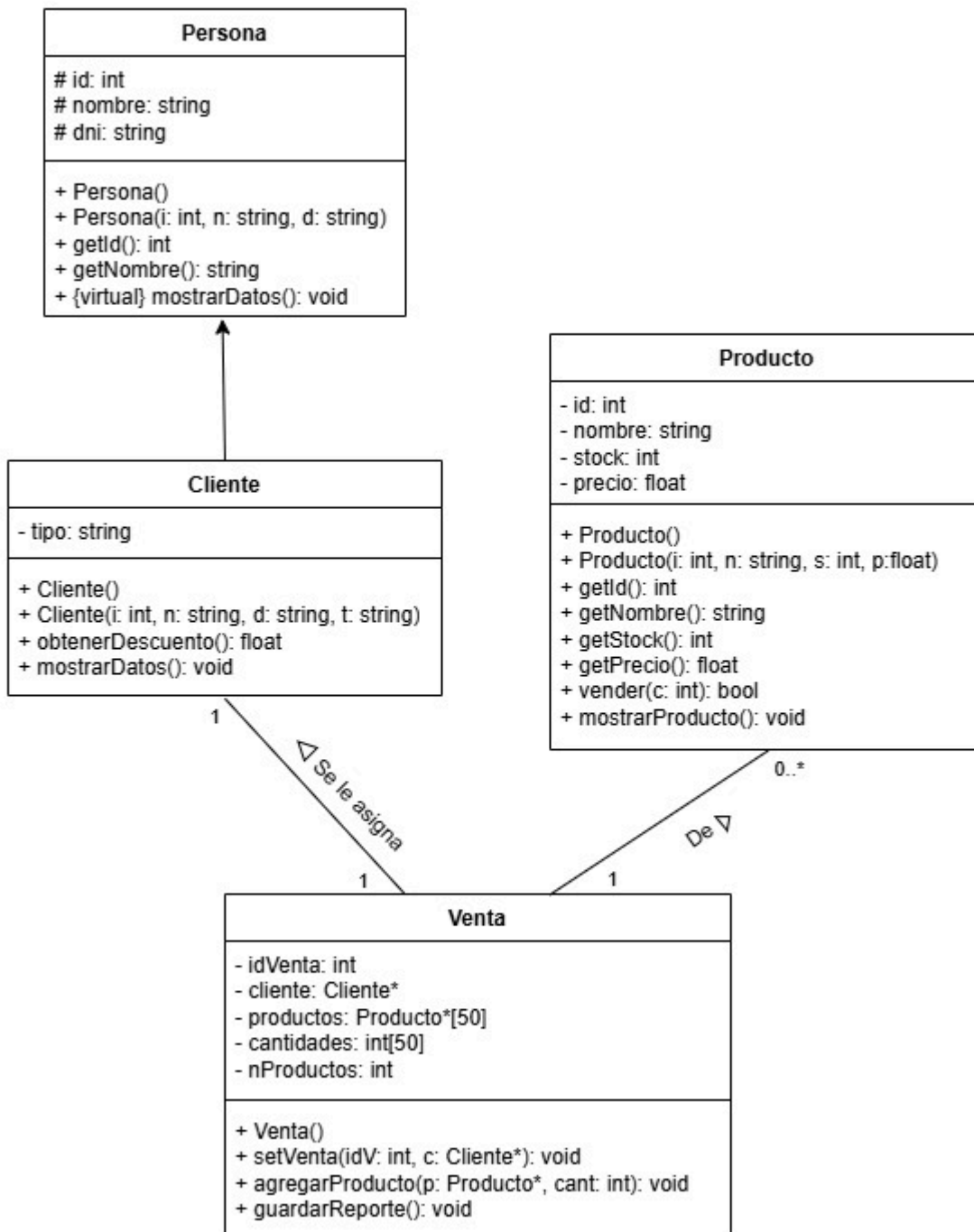
## **ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE**

- a. Requerimientos funcionales
  - Registrar clientes
  - Registrar productos
  - Buscar productos por ID
  - Ordenar productos por precio
  - Registrar ventas
  - Generar reporte de ventas
  - Mostrar advertencia de stock crítico
- b. Requerimientos funcionales
  - El sistema debe ejecutarse sin errores
  - Interfaz por consola clara
  - Uso de arreglos estáticos

## ARQUITECTURA DE SOFTWARE

### a. Vista de datos

El sistema está compuesto por las clases Persona, Cliente, Producto y Venta, relacionadas mediante herencia y asociaciones.





#### b. Módulos

- Persona.h / Persona.cpp
- Cliente.h / Cliente.cpp
- Producto.h / Producto.cpp
- Venta.h / Venta.cpp
- main.cpp

### CONCLUSIONES Y RECOMENDACIONES

El proyecto permitió aplicar correctamente los conceptos de Programación Orientada a Objetos, así como algoritmos básicos de búsqueda y ordenación. Se recomienda como mejora futura implementar almacenamiento completo de datos en archivos y una interfaz gráfica, al igual que mejorar en validaciones en algunas partes del código y prevenir errores dentro del programa hecho.

### BIBLIOGRAFÍA

Deitel, P., & Deitel, H. (2016). *C++ cómo programar* (9.<sup>a</sup> ed.). Pearson Educación.

[https://elhacker.info/manuales/Lenguajes%20de%20Programacion/C++/C%C3%B3mo%20programar%20C++,%209na%20Edici%C3%B3n%20\(2014\).pdf](https://elhacker.info/manuales/Lenguajes%20de%20Programacion/C++/C%C3%B3mo%20programar%20C++,%209na%20Edici%C3%B3n%20(2014).pdf)

### ANEXOS

Link	al	repositorio:
<a href="https://github.com/Anghelo21/Proyecto_Sistema_De_Gestion_Minimarket">https://github.com/Anghelo21/Proyecto_Sistema_De_Gestion_Minimarket</a>		