



Conceptos y Paradigmas de Lenguajes de Programación TRABAJO PRÁCTICO

Integrantes:

- Nahuel Flores –
- Patricio Arango –
- Valentin-
- Nehuen Rosa–
- Erika Valdez–
- Susana Varela-

Desarrollamos un sistema para la venta de productos, específicamente bebidas con y sin alcohol. La empresa posee una lista de clientes con información sobre sus compras, incluyendo los productos adquiridos y las cantidades correspondientes.

Los objetivos a alcanzar son:

- 1- Obtener un listado de los artículos vendidos, junto con las cantidades.
- 2- Identificar el artículo más vendido.
- 3- Identificar el artículo menos vendido.
- 4- Obtener las ventas realizadas antes de una fecha determinada.
- 5- Listar las compras de un cliente específico, junto con las fechas correspondientes.

En cuanto a la implementación, se utilizará el *paradigma orientado a objetos* y el lenguaje de programación Java. Como entorno de desarrollo integrado (IDE), se emplea Eclipse IDE for Java Developers. Para el lenguaje del *paradigma lógico* utilizamos el lenguaje Prolog.

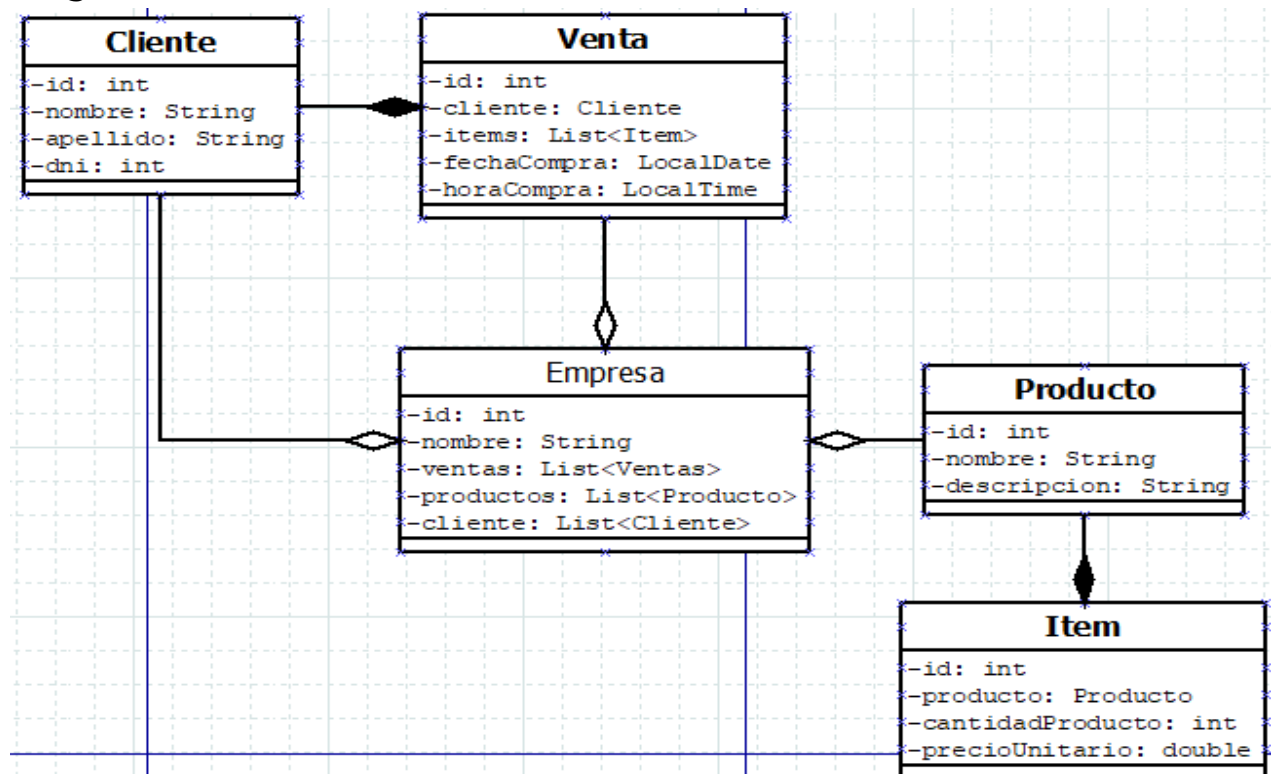
El proyecto JAVA se organiza en dos paquetes:

[foto de clases JAVA]

Paquete Clases: Este paquete contiene las siguientes clases: Empresa, Producto, Cliente, Ítem y Venta. Cada una de estas clases representan entidades relevantes para el sistema, como la empresa en sí, los productos que se venden, los clientes que realizan compras, los ítems de venta y las ventas en sí.

Paquete Test: En este paquete se incluye la inserción de datos y las pruebas de los casos de uso implementados.

Diagrama UML



Java es un lenguaje de programación con un enfoque orientado a objetos, lo que lo convierte en una opción sólida para modelar los productos, ventas, clientes y su gestión en el sistema.

Con él se pueden crear objetos de productos que encapsulan sus características y comportamientos. Cada producto puede ser representado como una clase en Java, con atributos como nombre, descripción, precio, fecha de creación, entre otros. La orientación a objetos de Java permite definir métodos dentro de la clase Producto para realizar operaciones específicas, como calcular el total de ventas, obtener la fecha de creación o aplicar filtros personalizados. Además, proporciona estructuras de datos dinámicas, como las listas, que son ideales para almacenar y administrar una colección de productos. Puede utilizarse una lista en Java, como ArrayList, para agregar y gestionar eficientemente los productos en el sistema de venta. Con las listas, es posible agregar nuevos productos, eliminarlos, modificar sus atributos o realizar búsquedas y filtrados en base a diferentes criterios, como fechas o cualquier otro atributo definido en las clases.

La capacidad de filtrar por fechas u otras opciones es una funcionalidad fundamental en un sistema de venta de productos. Java ofrece herramientas que facilitan la implementación de dichos filtros. Se pueden utilizar estructuras de control y algoritmos de búsqueda y filtrado para seleccionar los productos según los criterios especificados, como fechas de creación, precios, categorías, etc.

Conclusión:

Utilizar Java en un proyecto de venta de productos permite crear objetos de productos, agregarlos a una lista y facilitar la implementación de filtros por fechas u otras opciones. La orientación a objetos de Java, su capacidad para trabajar con estructuras de datos dinámicas y su rica biblioteca de herramientas hacen de este lenguaje una elección sólida para desarrollar un sistema de venta de productos flexible, escalable y fácil de mantener.

Proyecto en Prolog:

Primero establecemos una serie de hechos para definir cuáles son las bebidas. Después, determinamos que hay dos tipos: gaseosas o alcohólicas. En el siguiente paso definimos de qué tipo es cada bebida.

Dado estos predicados, sabemos que podemos consultar a Prolog sobre qué tipo es cada bebida ingresando el predicado "bebida_es_de_tipo" seguido de una bebida y un tipo y se evaluará true o false según el caso.

El predicado "mismo_tipo_bebida" recibe dos bebidas como parámetro y evalúa si ambas "bebida_es_de_tipo" coinciden. En ese caso devolverá true. También hacemos una revisión extra mediante la función dif para determinar que ambas bebidas ingresadas son distintas.

Conclusión:

Para el armado de listas y visualización de información en general, resultó mucho más sencillo el paradigma OOO, sin embargo, para las comparaciones y operaciones lógicas, Prolog se mostró mucho más confiable. Su cercanía con el lenguaje natural hace mucho más entendible el código, siendo más sencillo para su uso.

Es fácil para escribir "reglas" y luego inferir de manera lógica.

Si bien no se disponen de datos, a simple vista el Prolog "se siente" mucho más ágil, rápido y liviano.