

Programación Avanzada



Taller N°3

Asignatura: Programación Avanzada

Profesor: Tomás Reimann (tomas.reimann@ce.ucn.cl)

Ayudantes: Rodrigo Aguilera, Cristian Jiménez, Diego Cortés

Integrantes:

- Vicente Espinoza Lazo, 21.551.776-4, vicente.espinoza02@alumnos.ucn.cl
- Benjamín Nuñez Cepeda, 21.228.231-6, benjamin.nunez05@alumnos.ucn.cl

Facultad de Ingeniería y Ciencias Geológicas – Universidad Católica del Norte

1. Introducción

El presente proyecto corresponde al desarrollo de un sistema informático para la cadena de minimarkets “*Distribución Segura*”, cuyo objetivo es digitalizar los procesos de gestión de empleados, productos y ventas en su primera sucursal ubicada en Antofagasta.

Actualmente estas tareas se realizan de forma manual, lo que ocasiona errores, pérdida de información y dificultad en el control de stock. El sistema propuesto busca automatizar estas operaciones utilizando un enfoque de **Programación Orientada a Objetos (POO)**, incorporando herencia, clases abstractas e interfaces para modelar adecuadamente los distintos tipos de empleados y productos.

El diseño obtenido permite construir una arquitectura flexible, extensible y de fácil mantenimiento.

Roles del sistema

- **Administrador:** posee acceso completo al sistema. Puede registrar nuevos empleados, consultar información general y visualizar todas las ventas.
- **Cajero:** realiza ventas aplicando un descuento aleatorio por sesión (sobre Alimentos o Bebidas) y recibe un incentivo del 5 % del total vendido.

- **Reponedor:** gestiona el stock, repone productos según su categoría asignada y registra la cantidad total repuesta durante su jornada.

Archivos manejados

El sistema inicializa su información a partir de dos archivos de texto:

- **empleados.txt:** contiene el tipo de empleado, RUT, contraseña, nombre, remuneración y campos adicionales específicos para cada rol.
- **productos.txt:** contiene el tipo de producto, código, nombre, precio, marca, cantidad, proveedor y atributos adicionales según el subtipo (bebidas o alimentos).

2. Diagrama de Clases

El sistema se organizó en tres paquetes principales: **Dominio**, **Servicio** y **Vista**, siguiendo una arquitectura por capas que facilita la separación de responsabilidades y mejora la mantenibilidad.

2.1 Paquete Dominio

A) Empleados

Se define la clase abstracta **Empleado**, que incluye los atributos comunes:

- RUT
- Contraseña
- Nombre
- Tipo de empleado
- Remuneración

Subclases:

- **Administrador:**
 - superAdministrador: boolean
- **Cajero:**
 - numeroCaja
 - totalDineroRecaudado

- ventasRealizadas[]
 - cantidadVentasRealizadas
 - comisionAcumulada
 - Métodos:
 - registrarVenta()
 - verVentasRealizadas()
 - calcularComision()
 - **Reponedor:**
 - zona
 - tipoProductoAsignado
 - totalRepuesto
-

B) Productos

La clase base **Producto** define los atributos:

- código
- nombre
- precio
- marca
- cantidad
- proveedor

A partir de ella se extiende a dos jerarquías principales:

Bebidas (clase abstracta)

- volumenLitros
- cantidadSellos
- tipoEnvase

Subtipos concretos:

- **BebidaGaseosa** (retornable, nivelAzucar, versión)
- **BebidaAlcoholica** (gradosAlcohol, paísOrigen, controlEdad)
- **BebidaCaliente** (tipoBebida, temperaturaServicio, tamañoEnvase)

Alimentos (clase abstracta)

- fechaVencimiento
- peso
- origen

Subclases:

- **AlimentoPerecedero**
- **AlimentoNoPerecedero**
- **AlimentoCongelado**

C) Venta

La clase **Venta** representa una transacción realizada por un cajero.

Contiene:

- RUT del empleado
- RUT del cliente
- Fecha
- Arreglo de productos
- Cantidades
- Montos por producto
- Monto total

Y los métodos principales:

- agregarProducto()

- `calcularTotal()`
-

D) ContenedorMinimarket

Almacena todas las estructuras centrales del sistema:

- empleados
- productos
- ventas

Actúa como repositorio base de la información cargada desde los archivos.

2.2 Paquete Servicio

Define la interfaz **Sistema**, que contiene las operaciones del sistema:

- Manejo de sesiones
- Carga de archivos

- Registro de empleados
- Validación de RUT y número de caja
- Registro de ventas
- Listados de empleados, productos y ventas
- Aplicación de descuentos por sesión

La clase **SistemaMinimarket** implementa esta interfaz y mantiene:

- Una instancia de ContenedorMinimarket
- tipoDescuentoSesion
- porcentajeDescuento

2.3 Paquete Vista

Contiene la clase **Vista**, encargada de la interacción con el usuario mediante un menú que llama a los métodos del sistema.

3. Horas dedicadas

- Vicente Espinoza Lazo: 5 horas dedicadas (Diagrama de clases, código, contratos).
- Benjamin Nuñez Cepeda: 4 horas dedicadas (Diagrama de clases, Informe, código, contratos).

4. Conclusiones

La estructura diseñada permite desarrollar un sistema modular y escalable, organizado en paquetes que separan claramente el dominio, la lógica de negocio y la vista.

La utilización de POO facilita el modelado de los diferentes tipos de empleados y productos, permitiendo una asignación clara de responsabilidades mediante herencia y clases abstractas.

La lectura y carga de datos desde archivos externos, junto con la implementación del contenedor principal, deja preparado el sistema para futuras extensiones como cálculo de comisiones, reposición dinámica de productos y control exhaustivo de inventario.

El diseño obtenido constituye una base sólida para la implementación de la parte funcional del proyecto en Java.