



TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLÓGICO DE OAXACA

Carrera:

Ingeniería en Sistemas Computacionales

Materia:

Diseño e Implementación de software con patrones

Documentación

Docente:

Espinoza Pérez Jacob

Equipo:

Ordaz Pacheco Ruudvan

Julia Marleny

Vera Acevedo Héctor Aramís

Grupo:

7SB

Fecha de entrega:

18/03/2025

Documentación del Sistema de Productos

Descripción General

Este sistema implementa un conjunto de clases para la gestión de productos utilizando los patrones de diseño Abstract Factory y Factory Method. El sistema permite manejar diferentes tipos de productos (estándar, con descuento y promocionales) y proporciona una interfaz unificada para su creación y manipulación.

Estructura del Sistema

Diagrama de Clases

El sistema está compuesto por las siguientes clases principales:

- `Producto` (clase abstracta base)
- Clases derivadas de productos: `ProductoEstandar`, `ProductoConDescuento`, `ProductoPromocion`
- Fábricas abstractas: `ProductoAbstractFactory`
- Fábricas concretas: `ProductoEstandarFactory`, `ProductoConDescuentoFactory`, `ProductoPromocionFactory`
- `ProductoFactory`: Simplifica la creación de productos
- `Ctrl_Producto`: Controlador para operaciones de productos

Relaciones entre Clases

- Las clases de productos concretos heredan de la clase abstracta `Producto`
- Las fábricas concretas heredan de `ProductoAbstractFactory`
- El controlador `Ctrl_Producto` utiliza las fábricas y los productos

Clases y Métodos

Producto (Clase Abstracta)

Representa la base para todos los tipos de productos en el sistema.

Atributos

- ``idProducto``: Identificador único del producto
- ``nombre``: Nombre del producto
- ``cantidad``: Cantidad disponible en inventario
- ``precio``: Precio base del producto
- ``descripcion``: Descripción del producto
- ``porcentajeIva``: Porcentaje de IVA aplicable
- ``idCategoria``: Identificador de la categoría a la que pertenece
- ``estado``: Estado del producto (activo/inactivo)

Métodos

- ``calcularPrecioFinal()``: Método abstracto que calcula el precio final del producto
- Getters y setters para todos los atributos

ProductoEstandar

Implementación para productos estándar sin descuentos o promociones.

Métodos

- ``calcularPrecioFinal()``: Calcula el precio con IVA incluido

ProductoConDescuento

Implementación para productos con descuento porcentual.

Atributos Adicionales

- ``porcentajeDescuento``: Porcentaje de descuento aplicable

Métodos

- ``calcularPrecioFinal()``: Calcula el precio con IVA y aplica el descuento
- Getters y setters para ``porcentajeDescuento``

ProductoPromocion

Implementación para productos en promoción especial.

Atributos Adicionales

- ``descripcionPromocion``: Descripción textual de la promoción

Métodos

- ``calcularPrecioFinal()``: Calcula el precio según la promoción
- Getters y setters para ``descripcionPromocion``

ProductoAbstractFactory (Interfaz)

Define la interfaz para las fábricas de productos.

Métodos

- ``crearProducto()``: Crea una instancia básica del producto
- ``crearProductoCompleto()``: Crea una instancia completa con todos los atributos

ProductoEstandarFactory

Fábrica concreta para productos estándar.

Métodos Implementados

- ``crearProducto()``: Crea un nuevo ``ProductoEstandar``
- ``crearProductoCompleto()``: Crea un ``ProductoEstandar`` con todos los atributos

ProductoConDescuentoFactory

Fábrica concreta para productos con descuento.

Atributos

- ``porcentajeDescuento``: Porcentaje de descuento para aplicar

Métodos Implementados

- ``crearProducto()``: Crea un nuevo ``ProductoConDescuento``
- ``crearProductoCompleto()``: Crea un ``ProductoConDescuento`` con todos los atributos

ProductoPromocionFactory

Fábrica concreta para productos en promoción.

Atributos

- ``descripcionPromocion``: Descripción de la promoción

Métodos Implementados

- ``crearProducto()``: Crea un nuevo ``ProductoPromocion``
- ``crearProductoCompleto()``: Crea un ``ProductoPromocion`` con todos los atributos

ProductoFactory

Clase utilitaria para simplificar la creación de productos.

Métodos Estáticos

- ``crearProductoEstandar()``: Crea un producto estándar
- ``crearProductoConDescuento()``: Crea un producto con descuento
- ``crearProductoPromocion()``: Crea un producto en promoción

Ctrl_Producto

Controlador que gestiona las operaciones relacionadas con productos.

Métodos

- ``getFactory()``: Retorna la fábrica apropiada según el tipo de producto
- ``guardar()``: Guarda un producto en la base de datos
- ``crearProducto()``: Crea un nuevo producto según el tipo especificado
- ``existeProducto()``: Verifica si existe un producto con un nombre dado
- ``actualizar()``: Actualiza la información de un producto existente
- ``eliminar()``: Elimina un producto del sistema
- ``actualizarStock()``: Actualiza la cantidad disponible de un producto

Patrones de Diseño Utilizados

Abstract Factory

Implementado por `ProductoAbstractFactory` y sus clases derivadas, permite crear familias de objetos relacionados sin especificar sus clases concretas.

Flujo de Trabajo

Creación de un Producto

1. El cliente solicita la creación de un producto a través del controlador `Ctrl_Producto`
2. El controlador obtiene la fábrica apropiada según el tipo de producto
3. La fábrica crea y devuelve el producto solicitado
4. El controlador guarda el producto en la base de datos si es necesario

Actualización de un Producto

1. El cliente solicita actualizar un producto existente
2. El controlador verifica la existencia del producto
3. Se actualizan los atributos del producto
4. El controlador guarda los cambios en la base de datos