**Docente: Cristian cuadrado**

**Estudiante: jonathan David ornett castro**

**Universidad del Sinú**

**Actividad de Sistema de Gestión de Envíos**

**2024**

**Cartagena de indias**

**# Manual de Usuario - Sistema de Gestión de Envíos**

**## Introducción**

Este sistema está diseñado para facilitar la gestión de envíos en una empresa de paquetería. Permite calcular el costo de envío de diferentes tipos de paquetes, como paquetes estándar, documentos urgentes y artículos frágiles, en función de sus características y reglas específicas.

**## Tipos de Envíos**

**### Paquete Estándar**

Un paquete estándar tiene un costo base de $5.0, más un cargo adicional por peso ($0.5 por kg) y un cargo por volumen (0.01 por unidad de volumen en cm³).

**### Documento Urgente**

Los documentos urgentes tienen un costo base de $10.0, más un cargo adicional por peso ($1.0 por kg).

**### Artículo Frágil**

Los artículos frágiles tienen un costo base de $8.0, más un cargo adicional por peso ($0.8 por kg), un cargo por volumen (0.02 por unidad de volumen en cm³) y un cargo adicional por fragilidad de $2.5.

**## Cálculo del Costo de Envío**

Para calcular el costo de envío de un paquete, simplemente cree una instancia de la clase correspondiente al tipo de envío y llame al método `calcularCosto () `. Por ejemplo:

Envio paquete = new Paquete Estándar (2.5, 30, 20, 10, "Calle Principal 123");

double costoPaquete = paquete. calcularCosto ();

**Manual de Requerimientos Funcionales y No Funcionales:**

**# Manual de Requerimientos**

**## Requerimientos Funcionales**

- El sistema debe permitir calcular el costo de envío de paquetes estándar, documentos urgentes y artículos frágiles.

- El cálculo del costo de envío debe considerar el peso, las dimensiones y la fragilidad del paquete.

- El sistema debe tener una estructura modular y escalable para facilitar la incorporación de nuevos tipos de envíos en el futuro.

**## Requerimientos No Funcionales**

- El sistema debe ser implementado en Java.

- El código debe seguir los principios de la programación orientada a objetos.

- El sistema debe ser eficiente en el cálculo del costo de envío.

- El código debe estar bien documentado y seguir las mejores prácticas de programación.

**# Manual Técnico del Software**

**## Estructura del Código**

**### Clase Abstracta `Envio`**

Esta clase abstracta define los atributos comunes a todos los tipos de envíos, como el peso, las dimensiones y la dirección de destino. También declara el método abstracto `calcularCosto()`, que debe ser implementado por las clases derivadas.

**### Clases Concretas**

- `PaqueteEstandar`: Extiende de `Envio` e implementa el cálculo de costo para paquetes estándar.

- `DocumentoUrgente`: Extiende de `Envio` e implementa el cálculo de costo para documentos urgentes.

- `ArticuloFragil`: Extiende de `Envio` e implementa el cálculo de costo para artículos frágiles.

Cada clase concreta tiene sus propias tarifas base, costos por peso y cargos adicionales, según las reglas específicas para cada tipo de envío.

**## Flujo de Ejecución**

1. Se crea una instancia de la clase correspondiente al tipo de envío (por ejemplo, `PaqueteEstandar`).

2. Se llama al método `calcularCosto()` de la instancia creada.

3. Dentro del método `calcularCosto()`, se realizan los cálculos necesarios utilizando los atributos del objeto y las reglas específicas para ese tipo de envío.

4. Se devuelve el costo calculado.

**## Extensibilidad**

Si se necesita agregar un nuevo tipo de envío, simplemente se crea una nueva clase que extienda de `Envio` e implemente su propio método `calcularCosto () ` según las reglas correspondientes.

**## Consideraciones**

- El código sigue los principios de la programación orientada a objetos, como el encapsulamiento, la herencia y el polimorfismo.

- Se utilizan constantes para las tarifas y cargos, lo que facilita su modificación en caso de ser necesario.

- El código está documentado y sigue las convenciones de nomenclatura de Java.