Icono

Descripción generada automáticamente

**Ejercicio 1**

**PC 2**

**Dilian Anabel HURTADO PONCE**

Semana: 10

**Integrantes:**

Roberto Agustín Mejía Collazos

17/10/24

**Docente:**

***Caso de Estudio Nº 3: Una Tienda de Libros***

***CarroCompras.java***

Texto

Descripción generada automáticamente

***ItemCompra.java***

Captura de pantalla de computadora

Descripción generada automáticamente

***Libro.java***

**Captura de pantalla de computadora

Descripción generada automáticamente**

***Tienda\_de\_Libros.java***

***Texto

Descripción generada automáticamente***

***El código implementa un sistema simple para una tienda de libros donde los clientes pueden añadir libros a un carrito de compras. Se aplica el patrón Builder para construir de manera flexible un carrito con múltiples ítems (libros y sus cantidades) paso a paso.***

***Clase Libro: Representa los libros disponibles en la tienda, con atributos como el título, el ISBN y el precio. Incluye getters para acceder a estos atributos.***

***Clase ItemCompra: Representa un libro dentro del carrito de compras, junto con la cantidad que el cliente quiere comprar. También incluye getters para acceder al libro y la cantidad.***

***Clase CarroCompras: Es el carrito de compras del cliente. Se utiliza el patrón Builder para crear un carrito de forma flexible, añadiendo libros y cantidades paso a paso, y luego creando el objeto final con el método build().***

***Clase Builder: Es una clase interna que ayuda a construir un objeto CarroCompras. Permite agregar ítems (libros y su cantidad) al carrito y finalmente crear el carrito completo con build().***

***Método calcularTotal(): Este método en CarroCompras recorre todos los ítems en el carrito, multiplicando el precio de cada libro por su cantidad, y sumando los subtotales para devolver el monto total de la compra.***

***Ejemplo de Uso***

***En la clase Main, se crean algunos libros, se añaden al carrito con el Builder, y se calcula el total de la compra.***

***Diagrama UML***

