	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: René Adrián Dávila Pérez

Asignatura: Programación Orientada a Objetos

Grupo: 01

No. de proyecto: 01

Integrante(s): 322276824
322258516
425037384
320108116
322221415

No. de brigada: 01

Semestre: 2026-1

Fecha de entrega: 26 de septiembre de 2025

Observaciones: _____

CALIFICACIÓN: _____

Índice

1. Introducción	2
2. Marco Teórico	2
3. Desarrollo	4
3.1. Clase Carrito	4
3.2. Clase Proyecto	4
4. Resultados	5
5. Conclusiones	7

1. Introducción

- **Planteamiento del problema:**

El proyecto consistió en el desarrollo de una aplicación en Java que simula el funcionamiento de un carrito de compras. La aplicación debe permitir agregar, eliminar y mostrar artículos, utilizando estructuras de datos dinámicas para la gestión de los elementos. Organizar la solución en dos clases principales: **Proyecto**, que contiene el método **main**, y **Carrito**, encargada de modelar los artículos y proveer los métodos necesarios para la administración de la lista de compras.

- **Motivación:**

Simular un carrito de compras representa un ejemplo claro de cómo abstraer un problema cotidiano al entorno de la programación, fortaleciendo la capacidad de modelar situaciones reales mediante clases y objetos, por lo que implementar este programa significa poner a prueba los conocimientos adquiridos en el curso, en programación orientada a objetos y el manejo de colecciones.

- **Objetivos:**

Reforzar los conceptos de la programación orientada a objetos mediante el desarrollo de una aplicación que simula un carrito de compras. Consolidar la implementación de clases, objetos, atributos, métodos y constructores, así como el uso de estructuras de datos para gestionar dinámicamente la información de los objetos.

2. Marco Teórico

Objetos en Java

La abstracción es un principio fundamental de la programación orientada a objetos que consiste en simplificar la complejidad del mundo real al modelar objetos. De acuerdo con [1] “La esencia de la abstracción en programación es la simplificación. Al ocultar los detalles innecesarios y centrarse en lo esencial.”.

Un objeto es una instancia de una clase. Una clase es la plantilla o plano que define las propiedades (atributos) y los comportamientos (métodos) que tendrán los objetos que se creen a partir de ella. Cada objeto es una entidad única, con su propio estado (los valores de sus atributos) y una identidad que lo distingue de otros objetos. [6]

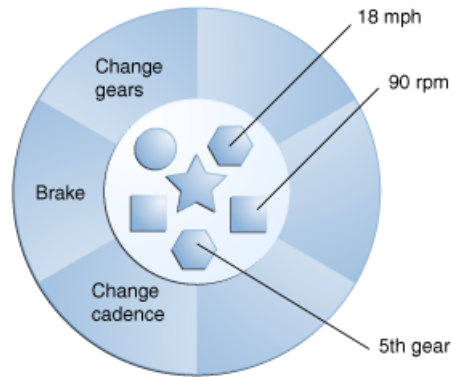


Figura 1: Una bicicleta modelada como un objeto. [6]

Constructores en Java

Un constructor es un método especial cuya finalidad es inicializar los objetos de una clase de una forma específica. Puede sobrecargarse para permitir diferentes formas de inicialización. Además según [4] “De manera predeterminada, el compilador proporciona un constructor predeterminado sin parámetros, en cualquier clase que no incluya un constructor en forma explícita.”, por ello definir constructores explícitos facilita un control más preciso sobre el estado inicial de los objetos.

ArrayList

La clase `ArrayList` es parte del paquete `java.util`. A diferencia de los arreglos tradicionales, que poseen un tamaño fijo, los `ArrayList` pueden crecer y reducirse dinámicamente según las operaciones realizadas. Proveen métodos como `add()` para agregar elementos, `remove()` para eliminarlos y `get()` para acceder a un elemento en una posición específica. Estas características los convierten en una herramienta flexible para gestionar colecciones de datos. [2]

HashMap

La clase `HashMap`, también del paquete `java.util`, permite almacenar pares clave-valor. Un `HashMap` no admite claves duplicadas, aunque sí valores repetidos. Entre sus métodos principales destacan `put()` para insertar un par clave-valor, `get()` para recuperar un valor asociado a una clave y `remove()` para eliminar una entrada. [3]

Modularidad

La modularidad constituye un principio esencial en el diseño de programas, ya que permite dividir un sistema en componentes más pequeños e independientes. En el contexto de la programación orientada a objetos, la modularidad se refleja en la organización del código en clases, lo cual facilita la reutilización, el mantenimiento y la extensión del software. Al estructurar un programa en clases, cada una con

responsabilidades bien definidas, se favorece el desarrollo incremental, es decir, la posibilidad de construir nuevas funcionalidades a partir de elementos ya existentes y probados. Los principales mecanismos que permiten alcanzar esta modularidad son la **composición**, en la que una clase se conforma a partir de objetos de otras, y la **herencia**, que permite crear nuevas clases especializadas a partir de otras sin modificar su implementación original. [5]

3. Desarrollo

La implementación del proyecto se organizó en dos clases principales: **Carrito** y **Proyecto**. La primera se encargó de modelar los artículos y los métodos asociados a la administración de la lista de compras, mientras que la segunda concentró el flujo de ejecución del programa a través del método **main**. De esta manera se mantuvo una estructura modular, clara y coherente con los principios de la programación orientada a objetos.

3.1. Clase Carrito

La clase **Carrito** se creó para poder modelar los artículos que irán en el carrito así como los métodos necesarios para la administración del carrito. Cuenta con dos atributos: **articulo** de tipo **String** y **precio** de tipo **Integer**. Contiene un constructor que ayuda a definir los atributos del objeto, para mostrar el contenido de un producto, la clase dispone del método **imprimir()**, que devuelve en formato de cadena los valores de sus atributos. Asimismo, contiene tres métodos **static** de acceso público que facilitan la gestión del carrito:

- **agregar()**: recibe un objeto de tipo **Carrito** y una lista de artículos, añadiendo el producto al **ArrayList** correspondiente.
- **mostrar()**: recorre la lista de artículos e imprime en consola cada producto utilizando el método **imprimir()**.
- **quitar()**: recibe un nombre y una lista de artículos, busca un objeto cuyo atributo **articulo** coincida con el nombre y lo elimina si existe, notificando al usuario si se encontró o no el producto.

3.2. Clase Proyecto

La clase **Proyecto** concentra la lógica principal del programa en su método **main** en el que se instancian cinco objetos de tipo **Carrito** usados para mostrarse como opciones al cliente. La gestión de los datos se realizó con un **ArrayList** denominado **tuCarrito**, usado para almacenar los productos seleccionados, y un **HashMap** llamado **productos**, en el cual se asociaron identificadores numéricos con los objetos creados anteriormente.

El programa funciona mediante un ciclo **while** que mantiene activo un menú de

opciones hasta que el usuario decida salir. Dentro del menú, un **switch** gestiona las siguientes acciones:

1. **Agregar artículo:** se muestran los productos disponibles y, tras la elección del usuario, se invoca el método `agregar()` para añadir el objeto al carrito.
2. **Quitar artículo:** se solicita el nombre del producto a eliminar y se ejecuta el método `quitar()` para retirarlo de la lista en caso de que se encuentre.
3. **Mostrar carrito:** se imprime el contenido actual del carrito mediante el método `mostrar()`.
4. **Salir:** se finaliza el programa y se imprime un mensaje avisando de su finalización.

4. Resultados

```
Carrito de compras

Elige una opción:
1. Agregar artículo
2. Quitar artículo
3. Mostrar carrito
4. Salir
1
1. Artículo: Pan Precio: 10
2. Artículo: Jabón Precio: 20
3. Artículo: Calcetines Precio: 30
4. Artículo: Garrafón Precio: 40
5. Artículo: Libro Precio: 50
Qué producto deseas agregar?

5
```

Figura 2: Adición de un artículo al carrito.

```
Carrito de compras

Elige una opción:
1. Agregar artículo
2. Quitar artículo
3. Mostrar carrito
4. Salir
3
Tu carrito contiene:

Artículo: Libro Precio: 50
Artículo: Jabón Precio: 20
```

Figura 3: Listado de artículos actuales en el carrito.

```
Carrito de compras

Elige una opción:
1. Agregar artículo
2. Quitar artículo
3. Mostrar carrito
4. Salir
2
Qué producto quieres eliminar?

Jabón
Se eliminó el producto: Jabón

Carrito de compras

Elige una opción:
1. Agregar artículo
2. Quitar artículo
3. Mostrar carrito
4. Salir
3
Tu carrito contiene:

Artículo: Libro Precio: 50
```

Figura 4: Eliminación de un artículo y visualización del carrito tras la eliminación.

```
Carrito de compras

Elige una opción:
1. Agregar artículo
2. Quitar artículo
3. Mostrar carrito
4. Salir
4
Terminando el programa...
```

Figura 5: Finalización del programa mediante la opción de salida del menú.

5. Conclusiones

El desarrollo de este proyecto permitió consolidar los conocimientos adquiridos en sesiones teóricas mediante la creación de clases, atributos, métodos y constructores. La interacción entre las clases **Proyecto** y **Carrito** evidenció la importancia de la modularidad y la reutilización de código en la construcción de aplicaciones más organizadas y escalables.

El uso de estructuras de datos dinámicas como **ArrayList** y **HashMap** resultó fundamental para la gestión de los artículos, demostrando cómo estas colecciones permiten almacenar, recuperar y modificar información de manera eficiente.

La implementación de un menú interactivo con operaciones de agregar, mostrar y eliminar artículos hizo más sencilla la interacción con el usuario, cumpliendo los objetivos planteados y reforzando la capacidad de abstraer procesos reales al ámbito de la programación.

Referencias

- [1] *¿Qué es la abstracción en programación?* Dic. de 2024. URL: <https://keepcoding.io/blog/que-es-la-abstraccion-en-programacion/>.
- [2] *Class ArrayList<E>*. Sep. de 2025. URL: <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>.
- [3] *Class HashMap<K, V>*. Sep. de 2025. URL: <https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html>.
- [4] Paul J. Deitel y Harvey M. Deitel. *Cómo programar en Java*. Pearson, 2008.
- [5] Roberto Rodríguez Echeverría, Encarna Sosa Sánchez y Álvaro Prieto Ramos. *Programación Orientada a Objetos*. Creative Commons, 2012. ISBN: 84-609-0003-7.
- [6] *What Is an Object?* Sep. de 2025. URL: <https://docs.oracle.com/javase/tutorial/java/concepts/object.html>.