



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

# Laboratorio de Computación Salas A y B

*Profesor(a):* M.C. RENE ADRIAN DAVILA PEREZ

*Asignatura:* PROGRAMACIÓN ORIENTADA A OBJETOS

*Grupo:* 01

*No de Proyecto(s):* 01

*Integrante(s):* 322118311

322094028

322092842

322078673

322067738

*No. de lista o  
brigada:* 03

*Semestre:* 2026-1

*Fecha de entrega:* 26/09/2025

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Marco Teórico</b>	<b>2</b>
2.1. Clases y objetos: . . . . .	2
2.2. Constructores: . . . . .	2
2.3. Estructuras de Datos: . . . . .	3
2.3.1. ArrayList . . . . .	3
<b>3. Desarrollo</b>	<b>4</b>
3.1. Carrito.java . . . . .	4
3.2. Archivo Proyecto.java . . . . .	4
3.3. Metodo main: . . . . .	4
<b>4. Resultados</b>	<b>4</b>
<b>5. Conclusiones</b>	<b>5</b>
<b>6. Bibliografia</b>	<b>6</b>

# 1. Introducción

- **Planteamiento del problema.** Para este proyecto se creara un programa en java el cual simulara un punto de venta, el programa nos debera pemitir agregar, mostrar y eliminar objetos, los cuales estaran almacenados en una estructura de datos.
- **Motivación.** Implementar este proyecto nos permitira reforzar los conceptos sobre programacion orientada a objetos vistos a lo largo del parcial (atributos, constructores, estructuras de datos y la comunicación entre clases mediante objetos.) todo esto en una aplicacion util de la vida real.
- **Objetivos.** Los objetivos seran implementar dos clases (Carrito y Proyecto) en los cuales definiremos y manipulares objetos contenidos en una estructura de datos, los cuales tendran los atributos, articulo y precio, el programa nos permitira agregar, eliminar y mostrar los objetos.

## 2. Marco Teórico

### 2.1. Clases y objetos:

Una clase es una "plantilla.<sup>a</sup> partir de la cual se crean objetos, definiendo sus atributos y métodos mientras que un objeto es una instancia de una clase que contiene estado (atributos) y comportamiento (métodos), y se utiliza para modelar elementos del mundo real.

```
//Ejemplo de clase  
class Persona { String nombre; }
```

```
Persona p = new Persona();  
p.nombre = "Alejandra";
```

[1]

### 2.2. Constructores:

Un constructor es un bloque especial de código que se ejecuta automáticamente al crear un objeto, permitiendo inicializar sus atributos. Su nombre coincide con el de la clase y no tiene tipo de retorno.

```
//Ejemplo de constructor  
class Libro {  
    String titulo;  
    Libro(String t){ titulo = t; }  
}  
Libro l = new Libro("1984");
```

[2]

## 2.3. Estructuras de Datos:

Las estructuras de datos son tipos abstractos de datos que nos permiten almacenar y organizar información de manera eficiente en este caso Java ofrece estructuras lineales como arreglos y listas. [3]

### 2.3.1. ArrayList

Un ArrayList es una lista dinámica que puede crecer o reducir su tamaño, permitiendo almacenar elementos en orden y acceder a ellos mediante índices. Incluye métodos para agregar, eliminar, modificar y recorrer elementos.

```
//Ejemplo de uso de ArrayList  
import java.util.ArrayList;  
ArrayList<String> lista = new ArrayList<>();  
lista.add("Manzana");
```

[4]

## 3. Desarrollo

### 3.1. Carrito.java

**Clase Carrito :** Dentro de esta clase definiremos el objeto Carrito, el cual tendra dos atributos, articulo el cual sera de tipo string y su precio el cual sera de tipo double, ademas de un constructor por defecto que permite crear objetos sin inicializar valores, y un constructor con parámetros que facilitara asignar directamente el artículo y el precio al momento de instanciar la clase.

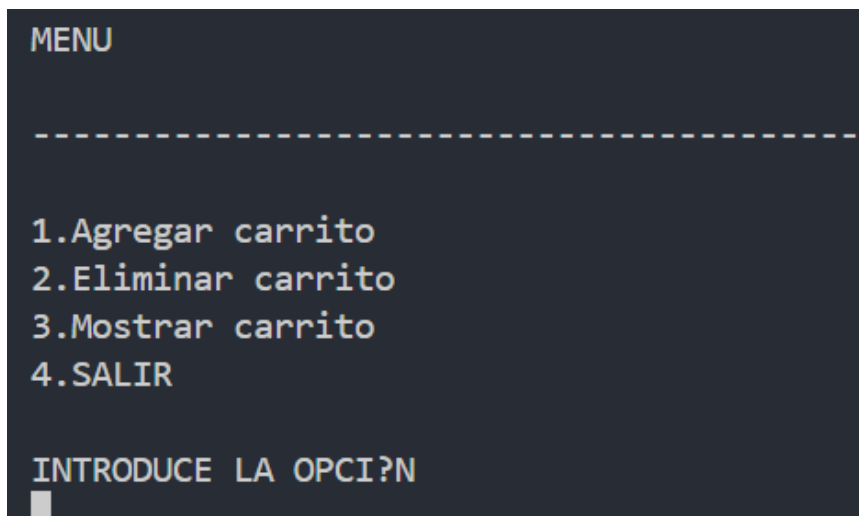
### 3.2. Archivo Proyecto.java

**Clase Proyecto :** La clase proyecto contendra el metodo main.

### 3.3. Metodo main:

Primero declaramos un `ArrayList<Carrito>` que funcionara como una lista dinámica donde se almacenaran los objetos Carrito, posteriormente mediante la clase Scanner, el programa recibira la opción del usuario y mediante la implementación de un `switch case` cuando el usuario seleccione la opción 1, se solicitara el nombre y precio de un artículo para crear un nuevo objeto Carrito y añadirlo al ArrayList, con la opción 2, se le pedira al usuario un número de artículo para eliminarlo de la lista solo si existe, la opción 3 mostrara todos los artículos y sus precios almacenados; en caso de que no haya elementos, se le indicara al usuario de que la lista está vacía. Finalmente, la opción 4 sera para cerrar el programa.

## 4. Resultados



```
MENU
-----

1.Agregar carrito
2.Eliminar carrito
3.Mostrar carrito
4.SALIR

INTRODUCE LA OPCI?N
```

Figura 1: Lista de opciones del menú

```
INTRODUCE LA OPCION
1
ingresa el nombre del articulo
gansito
ingresa el precio del articulo
15
```

Figura 2: Implementación de la función agregar

```
INTRODUCE LA OPCION
3
lista de articulos:

articulo 1: gansito precio: 15.0
```

Figura 3: Opción Mostrar carrito

```
INTRODUCE LA OPCION
2
Ingresa el articulo que quieres eliminar
1
Se ha eliminado el articulo numero: 1
```

Figura 4: Opción Eliminar carrito

```
INTRODUCE LA OPCION
3
lista vacia :(
```

Figura 5: Resultado de mostrar carrito despues de eliminar el unico articulo agregado

## 5. Conclusiones

Logramos darle solucion al problema al implementar una clase Carrito y una clase principal que interactúan mediante objetos, incorporando atributos, constructores y estructuras de datos que permitieron agregar, eliminar e imprimir artículos, lo cual nos confirma la importancia de los conceptos teoricos vistos en clase para la solucion de problemas reales.

Fue una aplicacion similar a la que se realizo con la practica 4, sin embargo, el proceso con el cual se desarrollo este proyecto no fue tan complejo, solo se necesito de dos clases:

- Proyecto
- Carrito

En donde la clase Proyecto se podia agregar o eliminar articulos, o cuando no habia ningun articulo mandaba el mensaje de lista vacia y la clase carrito donde se creaba el objeto, posteriormente hicimos que ambos objetos interactuen entre si. Cabe mencionar que en este caso el codigo podria haberse realizado con un arraylist o con un hashmap, en nuestro caso fue un arraylist, para facilitar el proceso de elaboracion del codigo.

## 6. Bibliografia

[1] Oracle, “Lesson: Classes and Objects,” The Java™ Tutorials, Oracle. [En línea]. Disponible: <https://docs.oracle.com/javase/tutorial/java/javaOO/index.html> . [Accedido: 25-septiembre-2025].

[2] Oracle, “Providing Constructors for Your Classes,” The Java™ Tutorials, Oracle. [En línea]. Disponible: <https://docs.oracle.com/javase/tutorial/java/javaOO/constructors.html> . [Accedido: 26-septiembre-2025].

[3] W3Schools, “Data Structures in Java,” W3Schools. [En línea]. Disponible: [https://www.w3schools.com/java/java\\_data\\_structures.asp](https://www.w3schools.com/java/java_data_structures.asp). [Accedido: 26-septiembre-2025].

[4] Oracle, “ArrayList (Java Platform SE 8),” Java Platform SE 8 Documentation, Oracle. [En línea]. Disponible: <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html> . [Accedido: 26-septiembre-2025].