

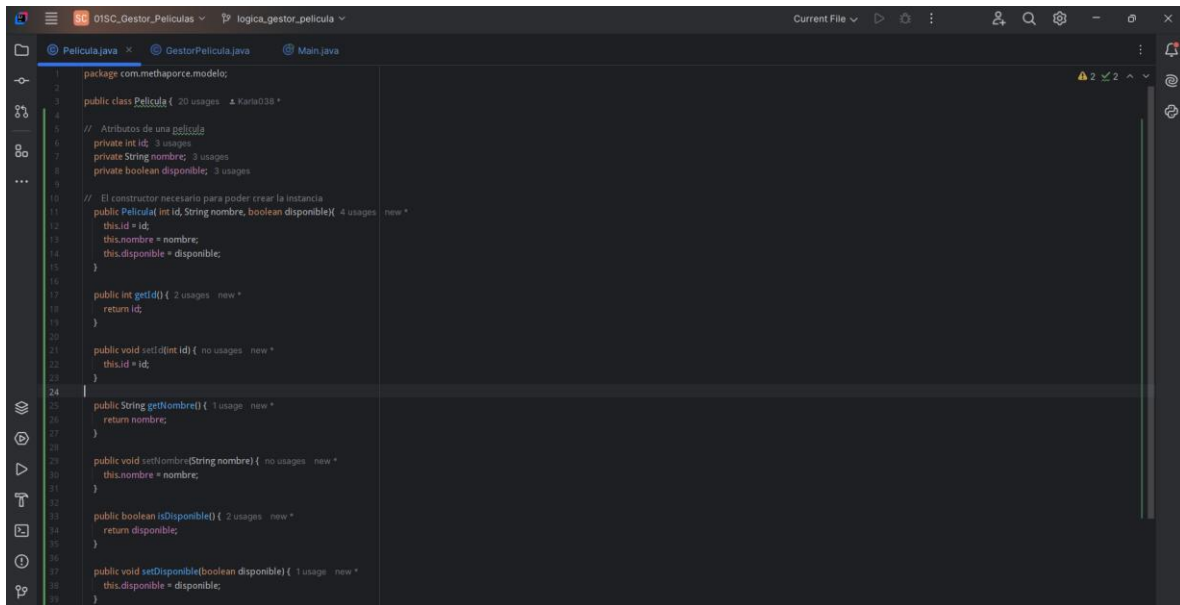
Metaphorce

Java Backend Junior

Karla Moya Díaz

Clase película

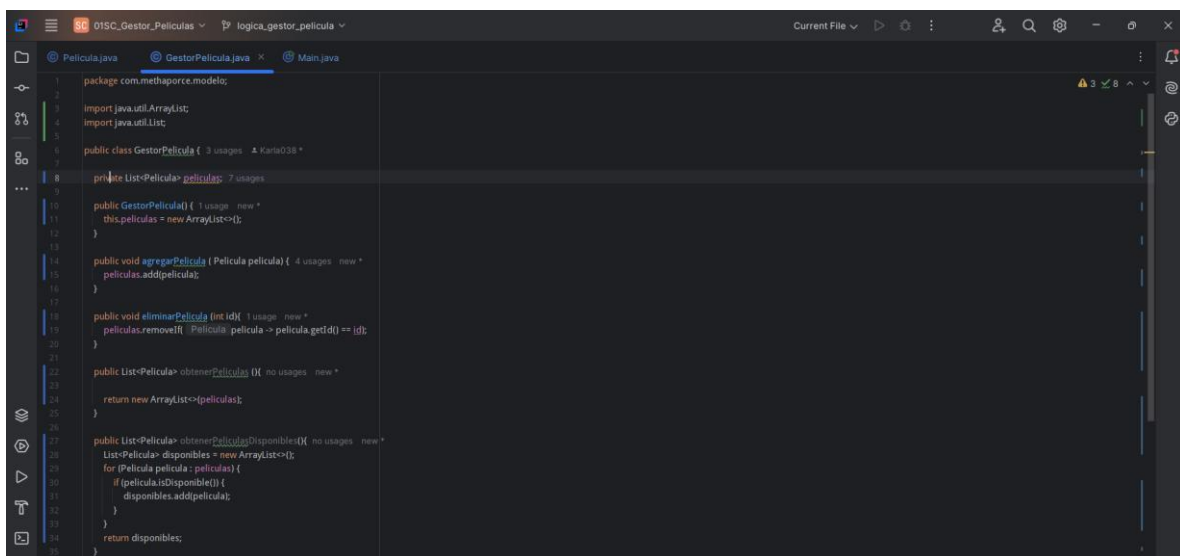
Se definen los atributos del objeto película.



```
1 package com.methaporce.modelo;
2
3 public class Pelicula {
4     // Atributos de una película
5     private int id;
6     private String nombre;
7     private boolean disponible;
8
9     // El constructor necesita para poder crear la instancia
10    public Pelicula(int id, String nombre, boolean disponible) {
11        this.id = id;
12        this.nombre = nombre;
13        this.disponible = disponible;
14    }
15
16    public int getId() {
17        return id;
18    }
19
20    public void setId(int id) {
21        this.id = id;
22    }
23
24    public String getNombre() {
25        return nombre;
26    }
27
28    public void setNombre(String nombre) {
29        this.nombre = nombre;
30    }
31
32    public boolean isDisponible() {
33        return disponible;
34    }
35
36    public void setDisponible(boolean disponible) {
37        this.disponible = disponible;
38    }
39 }
```

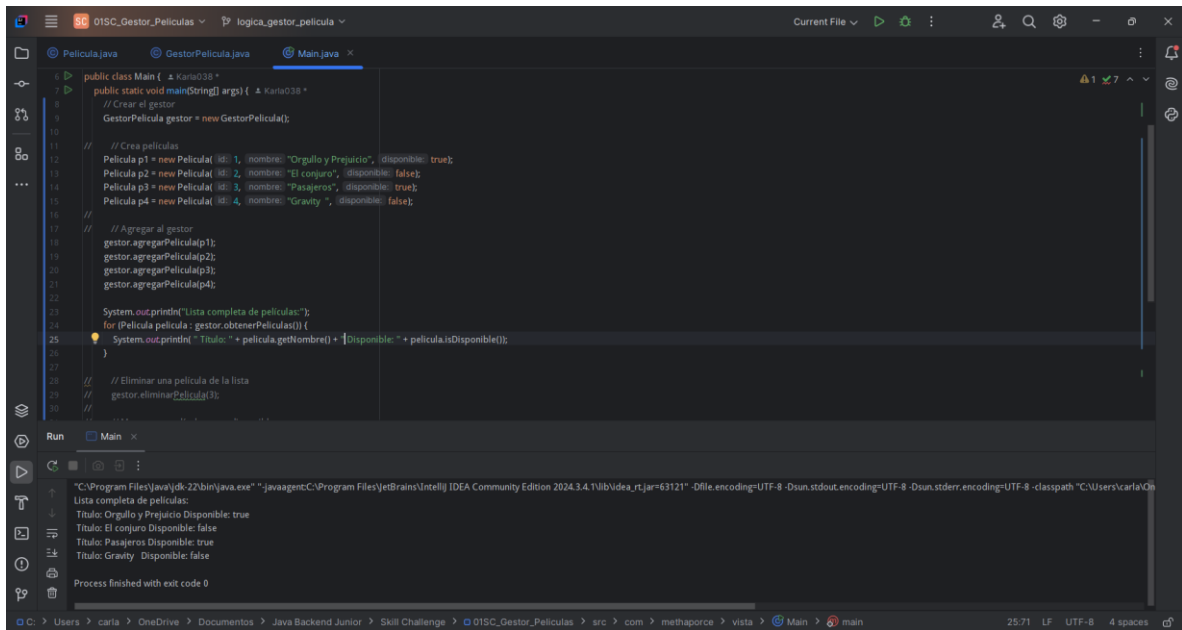
Clase GestorPelicula

Se definen los métodos para realizar las acciones de agregar, eliminar y consultar la disponibilidad.



```
1 package com.methaporce.modelo;
2 import java.util.ArrayList;
3 import java.util.List;
4
5 public class GestorPelicula {
6     private List<Pelicula> peliculas;
7
8     public GestorPelicula() {
9         this.peliculas = new ArrayList<>();
10    }
11
12    public void agregarPelicula(Pelicula pelicula) {
13        peliculas.add(pelicula);
14    }
15
16    public void eliminarPelicula(int id) {
17        peliculas.removeIf(pelicula -> pelicula.getId() == id);
18    }
19
20    public List<Pelicula> obtenerPeliculas() {
21        return new ArrayList<>(peliculas);
22    }
23
24    public List<Pelicula> obtenerPeliculasDisponibles() {
25        List<Pelicula> disponibles = new ArrayList<>();
26        for (Pelicula pelicula : peliculas) {
27            if (pelicula.isDisponible()) {
28                disponibles.add(pelicula);
29            }
30        }
31        return disponibles;
32    }
33 }
```


Agregar la película



```
1 public class Main {
2     public static void main(String[] args) {
3         // Crear el gestor
4         GestorPelícula gestor = new GestorPelícula();
5
6         // Crea películas
7         Película p1 = new Película(1, "Orgullo y Prejuicio", true);
8         Película p2 = new Película(2, "El conjuero", false);
9         Película p3 = new Película(3, "Pasajeros", true);
10        Película p4 = new Película(4, "Gravity", false);
11
12        // Agregar al gestor
13        gestor.agregarPelícula(p1);
14        gestor.agregarPelícula(p2);
15        gestor.agregarPelícula(p3);
16        gestor.agregarPelícula(p4);
17
18        System.out.println("Lista completa de películas:");
19        for (Película película : gestor.obtenerPelículas()) {
20            System.out.println("Título: " + película.getNombre() + " Disponible: " + película.isDisponible());
21        }
22
23        // Eliminar una película de la lista
24        gestor.eliminarPelícula(3);
25    }
26 }
```

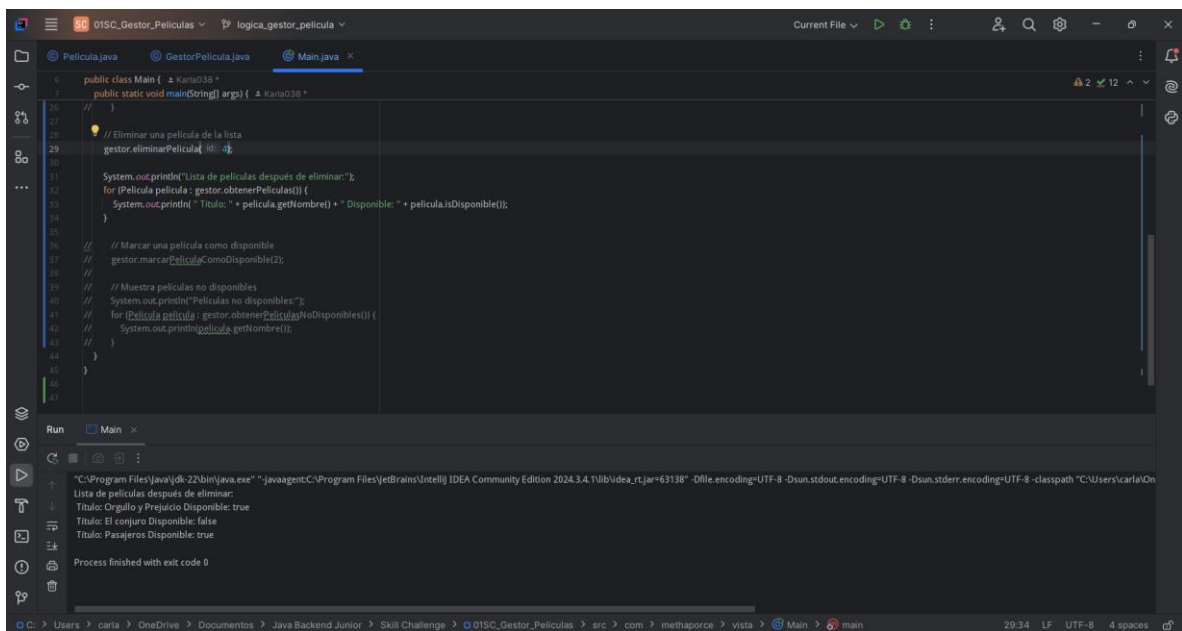
Run Main

Process finished with exit code 0

Output:

```
Lista completa de películas:
Título: Orgullo y Prejuicio Disponible: true
Título: El conjuero Disponible: false
Título: Pasajeros Disponible: true
Título: Gravity Disponible: false
```

Eliminar películas



```
1 public class Main {
2     public static void main(String[] args) {
3         // Eliminar una película de la lista
4         gestor.eliminarPelícula(4);
5
6        System.out.println("Lista de películas después de eliminar:");
7        for (Película película : gestor.obtenerPelículas()) {
8            System.out.println("Título: " + película.getNombre() + " Disponible: " + película.isDisponible());
9        }
10
11        // Marcar una película como disponible
12        gestor.marcarPelículaComoDisponible(2);
13
14        // Muestra películas no disponibles
15        System.out.println("Películas no disponibles:");
16        for (Película película : gestor.obtenerPelículasNoDisponibles()) {
17            System.out.println(película.getNombre());
18        }
19    }
20 }
```

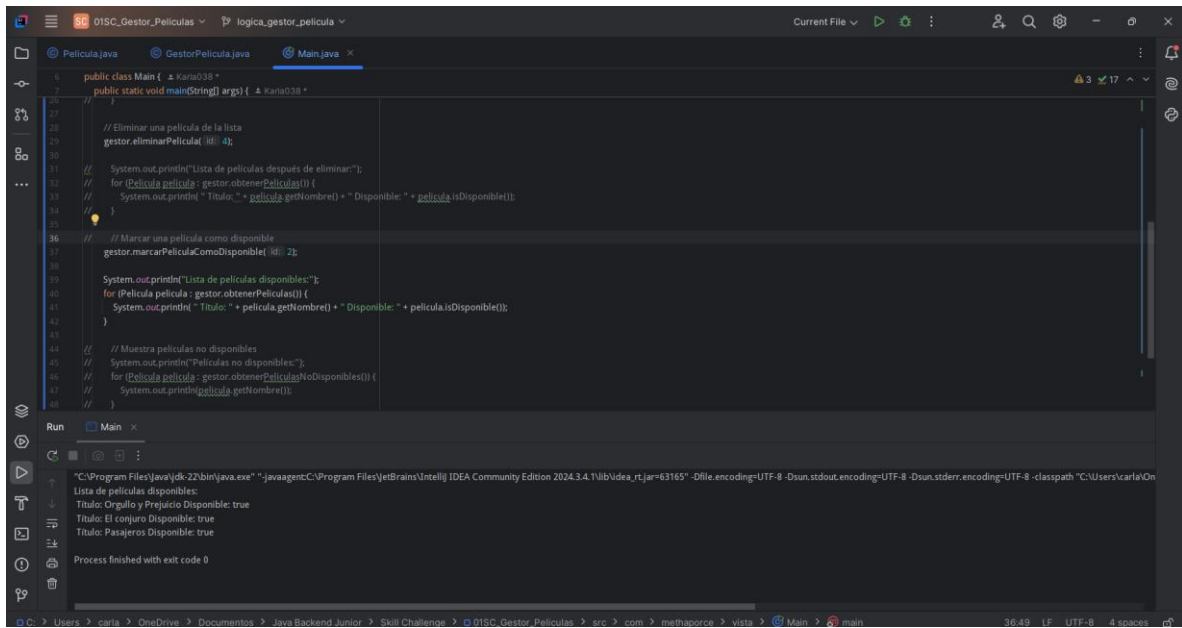
Run Main

Process finished with exit code 0

Output:

```
Lista de películas después de eliminar:
Título: Orgullo y Prejuicio Disponible: true
Título: El conjuero Disponible: false
Título: Pasajeros Disponible: true
```

Marcar película como disponible



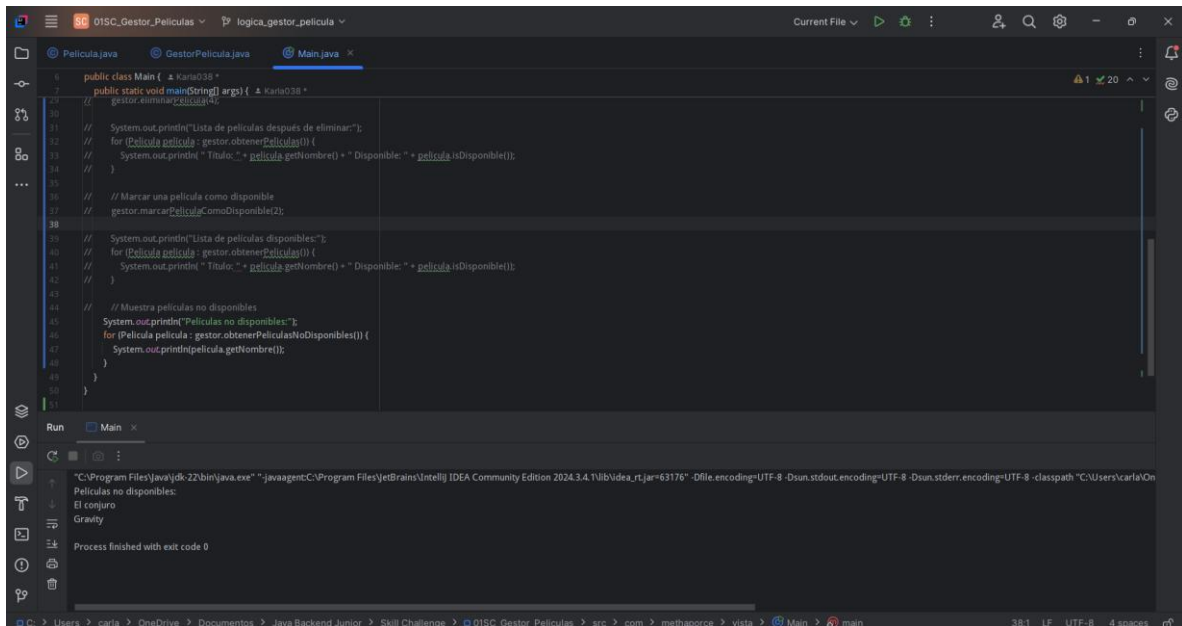
The screenshot shows the IntelliJ IDEA interface with a Java project named "OTSC_Gestor_Peliculas". The "Main.java" file is open, displaying the following code:

```
1 public class Main {  
2     public static void main(String[] args) {  
3         // Eliminar una película de la lista  
4         gestor.eliminarPelicula(4);  
5  
6         // Muestra películas no disponibles  
7         System.out.println("Lista de películas después de eliminar:");  
8         for (Pelicula pelicula : gestor.obtenerPeliculas()) {  
9             System.out.println("Titulo: " + pelicula.getNombre() + " Disponible: " + pelicula.isDisponible());  
10        }  
11  
12        // Marcar una película como disponible  
13        gestor.marcarPeliculaComoDisponible(2);  
14  
15        // Muestra películas no disponibles  
16        System.out.println("Lista de películas disponibles:");  
17        for (Pelicula pelicula : gestor.obtenerPeliculas()) {  
18            System.out.println("Titulo: " + pelicula.getNombre() + " Disponible: " + pelicula.isDisponible());  
19        }  
20  
21        // Muestra películas no disponibles  
22        System.out.println("Películas no disponibles:");  
23        for (Pelicula pelicula : gestor.obtenerPeliculasNoDisponibles()) {  
24            System.out.println(pelicula.getNombre());  
25        }  
26    }  
27 }
```

The Run window shows the output of the program:

```
Process finished with exit code 0  
Lista de películas disponibles:  
Titulo: Orgullo y Prejuicio Disponible: true  
Titulo: El conjuero Disponible: true  
Titulo: Panajeros Disponible: true
```

Películas no disponibles



The screenshot shows the IntelliJ IDEA interface with the same Java project. The "Main.java" file is open, displaying the following code:

```
1 public class Main {  
2     public static void main(String[] args) {  
3         // Eliminar una película de la lista  
4         gestor.eliminarPelicula(4);  
5  
6         // Marcar una película como disponible  
7         gestor.marcarPeliculaComoDisponible(2);  
8  
9         // Muestra películas no disponibles  
10        System.out.println("Lista de películas disponibles:");  
11        for (Pelicula pelicula : gestor.obtenerPeliculas()) {  
12            System.out.println("Titulo: " + pelicula.getNombre() + " Disponible: " + pelicula.isDisponible());  
13        }  
14  
15        // Muestra películas no disponibles  
16        System.out.println("Películas no disponibles:");  
17        for (Pelicula pelicula : gestor.obtenerPeliculasNoDisponibles()) {  
18            System.out.println(pelicula.getNombre());  
19        }  
20    }  
21 }
```

The Run window shows the output of the program:

```
Process finished with exit code 0  
Películas no disponibles:  
El conjuero  
Gravity
```

Enlace de git

https://github.com/Karla038/Skill_Challenge/tree/main/01SC_Gestor_Peliculas