



**Universidad**  
**Zaragoza**

## Practica 3

Pablo Villa  
Alvaro Perez

# Índice

Portada -----1

Índice -----2

Ejercicio 1 -----3

hemos definido varias variables estáticas al principio del programa:

- **Num\_Atracciones:** El número de atracciones en el parque.
- **Capacidad:** La capacidad máxima de cada atracción.
- **Vendedor:** Un semáforo que controla el acceso al vendedor de boletos. Solo un pasajero puede interactuar con el vendedor a la vez.
- **Operador:** Un array de semáforos, uno para cada atracción. Controla el acceso a los operadores de las atracciones.
- **Atracciones:** Otro array de semáforos, uno por atracción, que controla el acceso a las atracciones.
- **Salir:** Un semáforo que controla la salida del parque. Solo permite que un pasajero salga del parque a la vez.

En el método **main**, se inicializan los semáforos para los operadores y las atracciones, mediante un **for** en el cual recorreremos el número de atracciones que hay, de esta forma se irán accediendo a todas las atracciones y a estas asignándoles un operador.

Luego, se crean cinco hilos de pasajeros (**Pasajeros**) y se inicializan.

La clase **Pasajeros** es una clase anidada que representa a un pasajero. Cada pasajero tiene un nombre y sigue una serie de pasos:

- El pasajero llega al parque y genera un tiempo de espera aleatorio.
- El pasajero se mantiene en un bucle mientras decide si quiere abandonar el parque.
- Dentro del bucle, el pasajero intenta adquirir el semáforo **Vendedor** para tomar una decisión.
- El pasajero selecciona aleatoriamente una atracción.
- El pasajero libera el semáforo **Vendedor** y adquiere el semáforo **Operador** de la atracción elegida.

- El pasajero intenta adquirir un espacio disponible en la atracción .
- El pasajero viaja en la atracción, simulando el tiempo de viaje con **Thread.sleep**.
- Después del viaje, el pasajero libera el semáforo **Operador** y el espacio en la atracción.
- El pasajero decide si quiere abandonar el parque o quedarse para otra vuelta o montar en otra atracción, basado en una elección aleatoria.
- Si el pasajero decide abandonar el parque, adquiere el semáforo **Salir** para imprimir un mensaje sobre su salida y luego lo libera.

El programa utiliza los semáforos para gestionar el acceso al vendedor de boletos, a los operadores y a las atracciones. Esto asegura que solo un pasajero pueda interactuar con el vendedor a la vez y que los pasajeros puedan subir a las atracciones dentro de los límites de capacidad.

El programa utiliza la número aleatorios para determinar los tiempos de espera y la elección de la tracción, de esta forma no se sabe de antemano que atracción elegirá ni cuanto dura un viaje.

Y en cuanto a errores el único que tenemos actualmente es que genera una atracción 0, y esto se debe a que inicializamos la i de el for a 0, pero es que si la inicializamos a 1 a la hora de ejecutar el programa se produce un desbordamiento en el array y sale un error.

Previamente durante la realización del programa hemos encontrado problemas principalmente con los Semáforos a la hora de no saber exactamente cuando adquirirlos y cuando liberarlos, pero finalmente creemos que de esta forma esta bien ejecutado.

