

---

# PROGRAMACIÓN AVANZADA

## Práctica de Laboratorio

### Convocatoria Extraordinaria – Junio 2022

---

*Publicado el 07 de junio 2022*

#### **Simulación del funcionamiento de un sistema de peaje de carreteras**

Se pretende simular el comportamiento de **un peaje** que cuenta con diez (10) cabinas de peaje para cobrar a los distintos vehículos que van por una carretera.

El comportamiento del sistema a tener en cuenta es:

- Los vehículos, cada uno de los cuales se representará obligatoriamente como un hilo, llegarán a la zona del peaje (serán creados) con un tiempo aleatorio de entre 0,5 y 3 segundos. Cada vehículo podrá ser de uno de los siguientes tres tipos: Coche, Camión o Ambulancia. Cada vehículo tendrá un identificador único, por ejemplo: “Coche1”, “Coche2”, “Coche3”, “Camión1”, “Camión2”, etc.
- Los vehículos, para entrar a la zona del peaje, esperarán en un único punto de entrada (siguiendo una estrategia FIFO), es decir, irán entrando por orden de llegada.
- Dentro del peaje existen 10 cabinas de peaje, distribuidas de la siguiente manera: 4 cabinas para camiones (2 de pago manual y 2 de pago automático con tarjeta) y 6 cabinas para coches y ambulancias (3 de pago manual y 3 de pago automático con tarjeta).
- Una vez entra un vehículo al peaje, si hay alguna cabina libre, ésta es ocupada por el vehículo y entonces debe esperar a ser atendido por un empleado (en caso de que la cabina sea de pago manual). Si la cabina es de pago automático, podrá comenzar con la operación de pago (que durará entre 4 y 6 segundos) y continuar su camino, abandonando el peaje.
- Hay un total de cinco (2 en las cabinas de los camiones y 3 en las cabinas de los coches) empleados encargados de atender a los distintos vehículos y de cobrarles. Estos operarios serán modelados obligatoriamente como hilos. Cada operario tarda en cobrar a cada vehículo un tiempo aleatorio de entre 6 y 8 segundos.
- Cada 6 vehículos atendidos por un empleado, éste se para a descansar 5 segundos, momento en el que no atiende a ningún vehículo hasta que vuelva al trabajo.
- Una vez que un vehículo ha sido cobrado por un empleado, abandona la cabina de peaje y, por lo tanto, la zona de peaje.
- Se realizará una interfaz gráfica que muestre en todo momento el comportamiento de cada uno de los agentes involucrados. En dicha interfaz debe existir la opción de pausar y reanudar la simulación.
- Se deberá incluir un botón “Ambulancia” que, cuando sea pulsado, permitirá generar un vehículo de tipo Ambulancia (que deberá ser modelado obligatoriamente como un hilo). Al tratarse de un vehículo prioritario, este vehículo se situará en la cabeza de la cola de entrada a la zona de peaje y será el primero en entrar a las cabinas en cuanto haya un hueco. La ambulancia deberá pasar por la zona de cabinas de los coches. Se crearán tantas ambulancias como veces sea pulsado el botón.

El comportamiento de los vehículos se generará aleatoriamente mediante las funciones random de Java, y **todo el comportamiento del sistema se guardará en un log** (un fichero de texto llamado “evolucionPeaje.txt”), **además de mostrarse gráficamente** por pantalla, de forma que sea sencillo analizar lo sucedido. El log guardará los eventos que van teniendo lugar, por ejemplo: “2021-10-14 12:51:20 Coche1 llega a la zona de peaje”, “2021-10-14 12:51:21 Coche2 entra a la cabina de peaje 2”, “2021-10-14 12:51:24 Empleado1 cobra al Camión3”, “2021-10-14 12:52:15 Camión5 abandona la zona de peaje”, etc. En cada línea de dicho log deberá constar la marca de tiempo (incluyendo el segundo determinado en el que tuvo lugar el evento) y el evento en sí.

El sistema creará, como mínimo, 4.000 vehículos: por cada dos coches, se generará un camión. La simulación finaliza una vez se haya terminado de cobrar a todos los vehículos. Esta situación también se grabará en el log.

Para seguir la evolución del sistema, será preciso que se imprima el estado en el log cada vez que ocurra un evento. La interfaz gráfica permitirá visualizar el estado del sistema que contendrá, como mínimo, los datos:

- Qué vehículos están esperando a entrar en la zona de peaje.
- Qué vehículos están ocupando cada una de las cabinas de peaje.
- Situación en la que se encuentra cada empleado en cada momento.

Una posible interfaz del sistema sería la que se puede apreciar en la siguiente imagen:

Peaje

Vehículos esperando entrar al peaje

Coche20, Coche24, Camión13, Coche22, Camión14

Cabina Coches 1 (Manual)	Cabina Coches 2 (Manual)	Cabina Coches 3 (Manual)
Vehículo: Coche14	Vehículo: Coche17	Vehículo: Coche15
Empleado: Empleado1	Empleado: Empleado2	Empleado:

Cabina Coches 4 (Tarjeta)	Cabina Coches 5 (Tarjeta)	Cabina Coches 6 (Tarjeta)
Vehículo: Coche13	Vehículo: Coche12	Vehículo: Coche16

Cabina Camiones 1 (Manual)	Cabina Camiones 2 (Manual)
Vehículo: Camión11	Vehículo:
Empleado: Empleado5	Empleado: Empleado4

Cabina Camiones 3 (Tarjeta)	Cabina Camiones 4 (Tarjeta)
Vehículo: Camión9	Vehículo: Camión10

Parar Reanudar Ambulancia

Figura 1. Ejemplo de posible interfaz.

Además, será necesario incluir un nuevo módulo de consulta (utilizando programación concurrente distribuida).

Este nuevo módulo permitirá consultar de forma remota el estado del peaje. La situación de las cabinas de peaje, de los vehículos y de los empleados se actualizará automáticamente, es decir, sin intervención del usuario, actualizando la interfaz con una periodicidad de 1 segundo. Este módulo, además de la consulta, permitirá cerrar y abrir cada una de las cabinas de peaje de pago con tarjeta (tanto las de coches como las de camiones).

La información a mostrar será la misma que la de la interfaz principal de la práctica, es decir, se podrá utilizar la misma disposición de los elementos gráficos para la visualización en remoto. Una posible interfaz sería la descrita en la Figura 2.

Se deben desarrollar, en total, dos programas:

- Un servidor, cuyo código base será el programa desarrollado en la primera mitad de la práctica, ampliado con la funcionalidad correspondiente para dar soporte al módulo de consulta de programación distribuida.

- Un programa cliente (módulo de consulta) que realice la consulta y permita mostrar por pantalla (mediante jFrames de Java) el estado de la zona del peaje con los datos recibidos, así como el cierre y apertura de las cabinas de pago con tarjeta.

Peaje - Módulo Consulta

Vehículos esperando entrar al peaje

Coche20, Coche24, Camión13, Coche22, Camión14

Cabina Coches 1 (Manual)

Vehículo: Coche14

Empleado: Empleado1

Cabina Coches 2 (Manual)

Vehículo: Coche17

Empleado: Empleado2

Cabina Coches 3 (Manual)

Vehículo: Coche15

Empleado:

Cabina Coches 4 (Tarjeta)

Vehículo: Coche13

Cerrar

Cabina Coches 5 (Tarjeta)

Vehículo: Coche12

Cerrar

Cabina Coches 6 (Tarjeta)

Vehículo: Coche16

Cerrar

Cabina Camiones 1 (Manual)

Vehículo: Camión11

Empleado: Empleado5

Cabina Camiones 2 (Manual)

Vehículo:

Empleado: Empleado4

Cabina Camiones 3 (Tarjeta)

Vehículo: Camión9

Cerrar

Cabina Camiones 4 (Tarjeta)

Vehículo: Camión10

Cerrar

Figura 2. Ejemplo de posible interfaz para el módulo de consulta.

Se podrán utilizar todos los mecanismos vistos en clase para resolver todos los problemas de comunicación y sincronización que se plantean en este enunciado. No obstante, se deben utilizar los mecanismos de sincronización y comunicación que resuelvan el problema **de la forma más eficiente y óptima** posible. La parte de programación distribuida podrá ser resuelta utilizando Sockets o RMI.

## Condiciones de entrega

1. La práctica se realizará (*opcionalmente*) **por parejas** y deberá ser entregada **antes** de la fecha indicada en el Aula Virtual, a través de la tarea correspondiente, mediante la subida de dos

archivos: la memoria de la práctica en formato DOC y el proyecto Netbeans completo, comprimido como **ZIP** (no utilizar extensión .rar). No se aceptarán trabajos enviados pasada la fecha límite de entrega.

2. La entrega fuera del plazo indicado en el Aula Virtual supondrá una reducción en la calificación final, siendo del 25% si se entrega el día siguiente a la fecha límite, o del 50% si se entrega dentro de los dos días siguientes. La entrega más allá de esos dos días no será admitida bajo ninguna circunstancia.
3. El proyecto entregado deberá ser un **proyecto de NetBeans**. No se admitirán proyectos realizados con otros entornos de desarrollo.
4. Para aprobar, es condición necesaria que todos los programas funcionen correctamente y de acuerdo a las especificaciones indicadas en los enunciados.
5. Se debe desarrollar la solución haciendo uso de buenas prácticas de programación. Por ejemplo, es necesario que todos los nombres de las clases comiencen por una letra mayúscula y todos los nombres de atributos y métodos comiencen por una letra minúscula; los atributos deberán ser privados, y sólo se podrá acceder a ellos mediante métodos getter y setter.
6. Si la práctica es realizada por una pareja, **sólo uno de los integrantes deberá subirla** al aula virtual, indicando el nombre de ambos alumnos.
7. En la portada de la memoria deberán figurar los datos siguientes:
  - a. **Grado en Ingeniería en Ingeniería Informática /Computadores**
  - b. **Curso 2021/2022 – Convocatoria Extraordinaria**
  - c. **DNI – Apellidos, Nombre**
8. **La memoria deberá incluir, como anexo, el código fuente del programa. Si esto no fuera así, la práctica no podrá ser aprobada.**
9. La memoria explicativa de la práctica realizada deberá incluir, en el orden siguiente: 1) un análisis de alto nivel; 2) diseño general del sistema y de las herramientas de sincronización utilizados; 3) las clases principales que intervienen con su descripción (atributos y métodos); 4) un diagrama de clases que muestren cómo están relacionadas; y 5) el código fuente, como anexo.
10. Dicha documentación, exceptuando el código, no deberá extenderse más de 20 páginas. La calidad de la documentación – presentación, estructura, contenido, redacción – será un elemento básico en la evaluación de la práctica.
11. Para la defensa de la práctica, si el profesor de laboratorio así lo estimara necesario, deberá presentarse una copia en papel de la memoria, impresa por las dos caras y grapada. Este documento podrá ser utilizado por el estudiante como base para responder a las cuestiones que se le planteen en el ejercicio escrito sobre la realización de la aplicación.
12. Para mostrar el funcionamiento de los programas, es conveniente que cada estudiante utilice su propio ordenador portátil, en previsión de posibles problemas al instalarlos en alguno de los ordenadores del laboratorio.