

PROYECTO: De “Taquilla RockStar” a tu sistema con lista de espera

1. Descripción	2
2. Requisitos mínimos del programa.....	3
3. Informe que debe entregar el alumnado	3
1. Portada	3
2. Descripción del sistema	3
3. Diseño de la solución	4
4. Implementación.....	4
5. Pruebas realizadas	4
6. Uso de IA y otras ayudas	4
7. Conclusiones personales	5
4. Calificación.....	5

1. Descripción

En clase hemos trabajado con un programa base llamado **Taquilla RockStar**, dividido en dos partes:

1. Configuración del sistema

Ventana donde se definen datos del concierto (aforo, precio, nº de fans a simular, etc.).

2. Simulación con hilos

Ventana donde se ve cómo llegan los fans, compran entrada o se quedan fuera según el aforo.

A partir de ese **programa esqueleto**, tu tarea es:

1. Inventar un contexto nuevo (no puede ser un concierto RockStar):

- Parking, cine, piscina, parque de atracciones, gimnasio, museo, escape room, etc.

2. Adaptar el código a tu nuevo sistema:

- Mantener la idea de:
 - Aforo máximo (capacidad).
 - Usuarios/elementos que intentan entrar.
 - “Entradas” o accesos vendidos/permitidos.
 - Algún tipo de recaudación o valor acumulado.

- Renombrar variables, textos e interfaz para que tengan sentido en tu contexto.

3. Mejorar el sistema:

- Añadir al menos **1 mejora funcional** (ejemplos):

- Diferentes tipos de usuario/entrada con precios distintos.
- Mensajes según ocupación (tranquilo / casi lleno / completo).
- Botón de reinicio del sistema.
- Descuentos, bonos, puntos...

4. Implementar una vista de “lista de espera” en otra pestaña/ventana:

- Debe mostrar quién está en la **cola de espera** (los hilos que están esperando turno).
- Esta vista NO estará disponible al principio.
- Cuando termine la simulación se debe:
 - Habilitar y mostrar un botón** llamado “**Lista de espera**”.
 - Al pulsarlo, se abre la pantalla/pestaña con la lista de espera (aunque sea solo para ver cómo fue la cola durante la simulación).

Puedes usar **IA (como ChatGPT u otras herramientas)** para ayudarte:

- Para generar ideas, nombres, mensajes, trozos de código, etc.
- Siempre debes **entender lo que copias** y adaptar el código a tu proyecto.
- Si usas IA, debes indicarlo en el informe (qué has usado y para qué).

2. Requisitos mínimos del programa

El proyecto debe cumplir como mínimo:

- 1. Mantener la estructura lógica de Taquilla RockStar:**
 - a. Aforo máximo.
 - b. Usuarios/hilos que intentan acceder.
 - c. Algun tipo de contador de accesos/entradas.
 - d. Cálculo de recaudación/valor.
- 2. Nuevo contexto inventado:**
 - a. Todo el texto visible, nombres de variables y mensajes deben adaptarse a tu temática.
- 3. Simulación con hilos:**
 - a. Cada hilo representa un usuario/cliente/fan/coche... en tu sistema.
 - b. Debe haber **cola de espera real** (los hilos esperan turno para ser atendidos).
- 4. Vista de lista de espera:**
 - a. En **otra pestaña u otra ventana** (como prefieras).
 - b. Muestra quién está/ha estado en la cola.
 - c. Solo se puede abrir cuando la simulación ha terminado:
 - i. Aparece o se habilita el botón “**Lista de espera**”.
- 5. Al menos una mejora personal** sobre el esqueleto original.

3. Informe que debe entregar el alumnado

1. Portada

- Título del proyecto (ej.: “*Parking CityNight – Control de plazas con lista de espera*”).
- Nombre y apellidos.
- Curso y grupo.
- Fecha.

2. Descripción del sistema

- Explicar brevemente:
 - ¿Qué sistema has creado? (cine, parking, piscina, etc.).

- ¿Qué problema intenta resolver? (control de aforo, entradas, recaudación...).
- Qué partes vienen del esqueleto **Taquilla RockStar** y qué has cambiado.

3. Diseño de la solución

- Describir las **variables más importantes** (aforo, precio, contadores, etc.).
- Explicar brevemente cómo funciona la **simulación con hilos** en tu proyecto.
- Explicar cómo está pensada la **cola de espera**:
 - ¿Qué pasa cuando el usuario llega?
 - ¿Cuándo espera? ¿Cuándo pasa a ser atendido?
- Explicar dónde y cómo se muestra la **vista de “Lista de espera”**:
 - ¿Es otra ventana o una pestaña?
 - ¿Qué se ve en ella?

4. Implementación

- Comentar (en texto, no hace falta código completo):
 - Qué clases has creado o modificado (por ejemplo: Concierto, Cliente, Controller, etc.).
 - 2–3 fragmentos de código que consideres importantes (pegarlos y explicarlos en 2–3 líneas).
 - Ejemplo: cómo gestionas el semáforo o cómo rellenas la lista de espera.

5. Pruebas realizadas

- Explicar al menos **3 casos de prueba**:
 - Ejemplo:
 - Caso 1: aforo pequeño con pocos usuarios.
 - Caso 2: aforo pequeño con muchos usuarios (se llena, hay cola).
 - Caso 3: valores extremos (aforo alto, pocos usuarios...).
- Indicar:
 - Qué esperabas que pasara.
 - Qué ha pasado realmente.
 - Si has tenido errores, cómo los has corregido.

6. Uso de IA y otras ayudas

- Indicar si has usado herramientas de IA (ChatGPT, Copilot, etc.):
 - ¿Para qué las has usado? (ideas, mensajes, revisar código...).
 - ¿Qué cosas has hecho tú solo/a? (adaptar, probar, corregir, etc.).

- Indicar otras fuentes:
 - Páginas web, vídeos, compañeros, etc.

7. Conclusiones personales

- Qué has aprendido:
 - Sobre hilos y concurrencia.
 - Sobre JavaFX / interfaz.
 - Sobre organizar un proyecto.
- Qué mejorarías si tuvieras más tiempo.

4. Calificación

- Informe: **2 p**
- Trabajo diario: **3 p**
- Producto final (programa): **3 p**
- Presentación: **2 p** (28 de noviembre)