



FACULTAD DE INFORMÁTICA
UNIVERSIDAD NACIONAL DEL COMAHUE



TRABAJO PRÁCTICO FINAL **PROGRAMACIÓN CONCURRENTE**

Aeropuerto “Viaje Bonito”



ALUMNO:

- Kevin Manuel Quintero Martinez - FAI 2771

INTRODUCCIÓN:

El presente informe describe el desarrollo de una simulación concurrente del funcionamiento del aeropuerto "VIAJE BONITO", desde la llegada de los pasajeros hasta su embarque en el avión. El objetivo principal del trabajo es modelar el comportamiento de los distintos hilos que intervienen en el aeropuerto, asegurando una correcta coordinación y sincronización entre ellos. Todo se modeló en base a los requerimientos planteados en el trabajo práctico final obligatorio.

A continuación, se detallarán los métodos utilizados para la sincronización de los hilos, la coordinación entre los distintos componentes y las decisiones de diseño adoptadas para asegurar un comportamiento correcto y lo más realista posible.

DECISIONES DE DISEÑO

Hall

Para hacer la simulación lo más realista posible, se implementan colas de espera en el hall para cada uno de los puestos de atención del aeropuerto. Si un pasajero llega a un puesto y este está lleno, deberá esperar en la cola del hall perteneciente al puesto por el cual está esperando a que se libere un espacio, permitiéndole ser el primero en ingresar en cuanto haya disponibilidad.

Aeropuerto

Si el aeropuerto está cerrado y hay pasajeros dentro, estos permanecen, no son obligados a salir.

Tren

Solo inicia el recorrido si la capacidad está llena.

Se crean conjuntos de espera por cada terminal del aeropuerto, donde los pasajeros esperan a ser notificados cuando se llega a la terminal correspondiente para descender.

HILOS UTILIZADOS

Para modelar la concurrencia en el aeropuerto "VIAJE BONITO", se implementaron distintos hilos que representan a las entidades principales del sistema.

Pasajero

Cada pasajero es representado por un hilo independiente. Al ingresar al aeropuerto, el pasajero sigue un flujo que incluye la consulta en el puesto de informes, el check-in en la aerolínea correspondiente, el traslado a la terminal asignada y la espera hasta el embarque. Durante su tiempo en la terminal, puede optar por visitar el free-shop si el tiempo y la disponibilidad de la tienda se lo permite.

Tren

El tren es un hilo que simula el transporte de pasajeros entre las terminales del aeropuerto. Se mueve a lo largo de una vía y realiza paradas en las terminales para permitir el descenso de pasajeros. Al completar su recorrido, regresa vacío al punto de inicio y espera a que se llene su capacidad para iniciar un nuevo trayecto.

PuestoAtencion

Cada aerolínea cuenta con un hilo que representa su puesto de atención. Este hilo gestiona la atención de los pasajeros que llegan para realizar el check-in. Dado que los puestos tienen una capacidad máxima, el hilo se encarga de atender a los pasajeros en orden de llegada y permitir el ingreso de nuevos pasajeros conforme se liberen los espacios.

Guardia

El guardia es un hilo que regula el ingreso de los pasajeros a los puestos de atención. Su función es controlar el acceso al área de check-in, asegurando que solo ingresen pasajeros cuando haya lugar disponible en el puesto correspondiente.

Cada puesto de atención tiene su respectivo guardia.

Vuelo

Cada vuelo es representado por un hilo que administra el proceso de embarque. Este hilo se encarga de anunciar el inicio del embarque y permitir que los pasajeros esperando al vuelo aborden en el momento correspondiente.

Reloj

El reloj es un hilo que mantiene la sincronización del sistema. Su función es avanzar el tiempo de la simulación y coordinar los eventos que dependen del horario, como la apertura y cierre del aeropuerto y los tiempos de embarque para los respectivos vuelos.

MECANISMOS DE SINCRONIZACIÓN UTILIZADOS

Semáforos

Semáforos Binarios

- Se utilizan en el **puesto de informes**, donde un mutex garantiza la atención exclusiva de un pasajero a la vez, asegurando la correcta asignación de un puesto de atención.
- En los **puestos de atención y tiendas de las terminales**, se emplean mutex para gestionar secciones críticas, evitando conflictos de concurrencia.
- También se implementa un semáforo para **notificar al guardia** cuando se libera un espacio en la cola del puesto de atención, permitiendo el ingreso de un nuevo pasajero en caso de que haya alguno esperando en el hall.

Semáforos Genéricos

- Controlan el **ingreso a las tiendas** de cada terminal. Cada tienda tiene una capacidad máxima, y su semáforo correspondiente gestiona la entrada de pasajeros según la disponibilidad.
- Administran las **cajas de pago** en las tiendas. La cantidad de permisos del semáforo indica el número de cajas disponibles; si están ocupadas, los pasajeros deben esperar a que se libere una.

Monitor

- Se emplea en la **entrada al aeropuerto**: los pasajeros que intentan ingresar cuando el aeropuerto está cerrado quedan bloqueados hasta que se les notifique la apertura.
- Se implementa en los **pasajeros**, permitiendo que realicen un **wait** en diferentes momentos:
 - Durante la **espera en la cola del puesto de atención**, donde se despiertan mediante el método **atenderPasajero** ejecutado por el hilo del puesto de atención una vez que han llegado a la primera posición de la cola.
 - También durante la **espera en la cola del hall del puesto de atención**, donde se despiertan mediante el método **permitirIngresoDesdeHall** ejecutado por el hilo del guardia luego de ser agregado a la cola del puesto.
- También es utilizado en el **reloj**, donde los vuelos consultan la hora del aeropuerto para determinar si es el momento de iniciar el embarque. Si aún no ha llegado el horario establecido, los vuelos permanecen bloqueados a la espera de una notificación del reloj, que les indicará cuándo pueden proceder con el embarque.

Locks y Conditions

- Se utilizan en el **tren** para gestionar la espera de los pasajeros en las terminales y evitar que ingresen si el tren está lleno o si ya inició su recorrido. Estos esperan en la condición **“esperandoNuevoRecorrido”**. Son notificados una vez que el tren llega al inicio del recorrido.
- Se crean **condiciones de espera** para cada terminal, donde los pasajeros esperan a ser notificados para descender al llegar a la terminal correspondiente.
- También se utiliza una condición **“avanzar”** donde el tren espera a ser notificado por el último pasajero que baja en una determinada terminal para que pueda seguir con el recorrido.

BlockingQueue

- Se implementa en los **puestos de atención de cada aerolínea** y en el **hall**:
 - Los pasajeros verifican si hay espacio en la cola del puesto de atención, si hay, se agregan, de lo contrario, esperan en el hall. En el hall cada puesto de atención tiene su respectiva cola de espera.
 - Los hilos de los puestos de atención ejecutan **atenderPasajero**, atendiendo al primer pasajero en la cola o bloqueándose si no hay ninguno disponible.
 - Cuando se libera un espacio, se notifica al **guardia**, quien, si hay pasajeros esperando en el hall, traslada al primero a la cola del puesto de atención.

CyclicBarrier

- Se usa para **sincronizar el inicio del recorrido del tren**. Se establece una barrera con un valor de **capacidad + 1**, dónde **capacidad** representa el número máximo de pasajeros y **+1** incluye al propio tren. El recorrido comienza una vez que todos los permisos han sido tomados.

CountDownLatch

- Se implementa en los **vuelos** con un valor inicial de 1, permitiendo sincronizar el **embarque de los pasajeros**.
- Los pasajeros realizan un **await** en el Latch al llegar al puesto de embarque si su vuelo aún no ha despegado.
- El **hilo del vuelo**, notificado por el reloj del aeropuerto, decrementa el Latch a 0 para iniciar el embarque, permitiendo que todos los pasajeros en espera suban al avión.

ARCHIVO LOG

El programa cuenta con un archivo “**log.txt**”, que es un registro detallado de todas las acciones realizadas por los pasajeros, el tren, el guardia, los puestos de atención, los vuelos, etc. dentro del sistema de simulación del aeropuerto. Este archivo permite rastrear cada evento en el orden en que ocurre con una marca de tiempo, facilitando el análisis del comportamiento del sistema.

En el archivo Log.java, se encuentra la ruta del registro “log.txt”.