

Informe TPO Aeropuerto “Viaje bonito”

Metodología de trabajo:

Cuando un pasajero entra al aeropuerto, se dirige a un puesto de informes y desde allí a un puesto de atención según su aerolínea (hay tantos puestos de atención como aerolíneas. Es decir que si hay 'n' aerolíneas, hay 'n' puestos de atención).

Cada puesto de atención tiene una capacidad máxima y se atiende por orden de llegada.

Los que llegan después de que se llenó el puesto de atención, esperan en un hall y un guardia les va avisando cuando se desocupa un lugar.

Cuando un pasajero hace el check-in, se le informa su terminal y puesto de embarque. A continuación se lo traslada a la terminal y puesto de embarque que le corresponde mediante un tren interno.

En el aeropuerto hay un tren interno que traslada los pasajeros a la terminal que le corresponde a cada uno. El tren se va vaciando a medida que para en las terminales y va dejando pasajeros, cuando para en la última terminal (la C) vuelve al inicio del recorrido. El tren sale de nuevo una vez que se llena por completo nuevamente.

En el aeropuerto hay 3 terminales (A, B y C). Cada una tiene puestos de embarque y una sala de embarques compartida.

A: Puestos de embarque 1 a 7.

B: Puestos de embarque 8 a 15.

C: Puestos de embarque 16 a 20.

(Esto es tal cual dice el enunciado)

Cada pasajero puede ir al free-shop de su terminal (si le sobra bastante tiempo) mientras espera su vuelo o simplemente esperar sentado.

En cada terminal hay un free-shop. Cada uno tiene una capacidad máxima de pasajeros. Hay 2 cajas para procesar los pagos.

Las aerolíneas se generan de manera aleatoria (como dice el enunciado) de entre todas las aerolíneas que hay en el aeropuerto.

n° aerolínea = n° puesto de atención (Es decir que si por ejemplo un pasajero tiene su vuelo en la aerolínea número 2, entonces el puesto de atención que le corresponde ser atendido es el puesto de atención número 2).

Mecanismos de sincronización utilizados:

Aeropuerto: ReentrantLock (Como solo hay un puesto de atención, se utiliza este lock para sincronización.)

TrenInterno: ReentrantLock (Se utiliza uno por terminal (A, B y C) y su función es la de mantener los pasajeros en espera mientras viajan con el tren interno, luego se van liberando a medida que el tren para en las estaciones a dejar los pasajeros).

Semaphore (Uno para saber cuando el tren está lleno y debe partir y el otro para saber cuando el tren está en la estación esperando pasajeros.)

Freeshop: ReentrantLock (Se usa un lock para mutex de una variable).

Semaphore (Un semáforo utilizado para simular las cajas del freeshop.)

HallCentral: Semaphore[]. (Los pasajeros esperan en el hall central porque su puesto de atención está lleno. Cuando se libera un lugar en

x puesto de atención, un guardia avisa a los pasajeros de ese puesto de atención x, que hay un lugar libre.)

PuestoDeAtencion: ArrayBlockingQueue. (Los pasajeros se van colocando en esta cola para ir siendo atendidos en orden de llegada.)

ReentrantLock. (Utilizado para mutex a la hora de cambiar variables compartidas.)

Vuelos: Monitores. (Los pasajeros esperan en el monitor de su vuelo a que sea la hora de partir.)

Nota:

Se recomienda ejecutar test.java en VS Code. Dejar correr durante un minuto o así y luego abrir el Auxiliares/log.txt para seguir el algoritmo que se ejecutó.

Utilizar la extensión 'ANSI Colors' de 'Iliia Pozdnyakov' para abrir el log.txt y poder ver los colores de cada print. Para ello, abrir log.txt en una ventana de VS Code, presionar 'F1' y buscar la opción 'ANSI Text: Open Preview', en esa preview se deberían ver los colores de cada print. (se puede resaltar/seleccionar el nombre de un pasajero particular para que sea "highlighted" y seguir su progreso a lo largo de la ejecución del programa)