# Parqueadero

## Enunciado

Se quiere construir una aplicación para administrar un parqueadero. Dicho parqueadero tiene 40 puestos enumerados, en cada puesto se puede parquear un carro. Se conoce la hora actual del parqueadero, la cual corresponde a un valor entre 6:00 y 20:00, dado que el parqueadero está abierto entre 6 de la mañana y cierra a las 8 de la noche. Los carros pueden ingresar únicamente durante las horas de apertura, al alcanzarse la hora de cierre se prohíbe el acceso. El parqueadero tiene una tarifa por hora, la cual se utiliza para estimar el valor que deben cancelar los carros según la cantidad de horas que permanecen dentro del parqueadero. Se pueden modificar la hora actual y la tarifa, al avanzar una hora el reloj o ingresar una nueva tarifa. En todo momento, se conoce la lista de carros que están parqueados en el parqueadero.

De cada puesto se conoce:

- El número del puesto. Este es un identificador único, dos puestos no tienen el mismo número.
- El carro que se encuentra parqueado en el puesto actualmente. Un puesto puede estar vacío.

De cada carro se conoce:

- Placa: Identificador único del carro. No puede haber dos carros con la misma placa.
- La marca del carro.
- El modelo del carro.
- La hora de ingreso.

La aplicación debe permitir:

1. Entrar un carro al parqueadero.
2. Sacar un carro del parqueadero.
3. Informar los ingresos del parqueadero.
4. Consultar la cantidad de puestos disponibles.
5. Avanzar el reloj del parqueadero.
6. Cambiar la tarifa del parqueadero.
7. Ordenar la lista de carros por su marca.
8. Ordenar la lista de carros por su modelo.
9. Ordenar la lista de carros por su hora de ingreso.
10. Buscar un carro por su placa.
11. Buscar un carro por su hora de ingreso.
12. Importar la información del parqueadero de un archivo de texto.
13. Generar un reporte con las ganancias estimadas.
14. Avanzar un día en el parqueadero.

La información del parqueadero debe ser persistente y el proceso debe ser completamente transparente para el usuario. Esto quiere decir que el programa debe dar la posibilidad de guardar la información del mundo en un archivo binario cada vez que el usuario termina la ejecución del mismo y cargar dicho archivo al inicio de la ejecución del programa para reconstruir el estado del mundo. Esta persistencia se debe manejar por medio de serialización/deserialización.

Por otra parte, existe la posibilidad de cargar una lista de carros para ingresarlos, junto con el estado de la tarifa y el valor en caja del parqueadero desde un archivo de texto, el cual debe seguir el formato para poder cargar la información de forma correcta. Al cargarse esta lista, los carros ya parqueados se borran, se reestablece la hora actual a la hora de inicio y con cada 5 carros nuevos ingresados se avanza en una unidad la hora actual.

Adicionalmente se puede generar un reporte, en el cual se calculan las ganancias del parqueadero en caso de que todos los carros actualmente ingresados salieran al momento de generarse el reporte. Este reporte muestra el valor cancelado por cada carro y la suma de estos valores, el formato de este reporte se especifica en la sección “Persistencia” de este documento.

En caso de cualquier error en la ejecución del programa, este debe desplegar un mensaje claro que explique la razón del problema y en algunos casos registrar el problema en un archivo de log de errores llamado reporteExcepciones.txt (en el directorio "data"). Este archivo debe contener el histórico de errores registrados (esto quiere decir que la escritura de un nuevo error debe añadirse al final del archivo existente). El formato de este archivo es especificado en la sección “Persistencia” de este documento. A continuación se listan los errores que se pueden presentar y el tipo de acción que debe realizar el programa para cada uno de ellos:

| Error que se puede presentar | Acción del Programa |
|---|---|
| Al intentar cargar el estado inicial del mundo, hay un error de entrada / salida. | Mensaje al usuario informando el error. Se abre la aplicación sin información y se registra el problema en el log de errores. |
| Al intentar guardar el estado final del mundo, hay un error de entrada / salida. | Mensaje al usuario informando el error, el programa se cierra sin guardar el estado y se registra el problema en el log de errores. |
| Al importar el archivo txt con la lista de carros. | Mensaje al usuario informando el error. No se carga la información y se registra el problema en el log de errores. |
| Al escribir el archivo de reporte. | Mensaje al usuario informando el error. No se genera el reporte y se registra el problema en el log de errores. |
| Al consultar la placa del carro en un puesto que se encuentra vacío. | Mensaje al usuario informando el error únicamente, no se registra en el log de errores. |
| Al ingresar un carro al parqueadero, si el parqueadero se encuentra cerrado o lleno, o si existe un carro con la misma placa. | Mensaje al usuario informando el error. El carro no ingresa al parqueadero y el problema es registrado en el log de errores. |
| Al sacar un carro del parqueadero cuando el parqueadero se encuentra cerrado. | Mensaje al usuario informando el error. El carro no sale del parqueadero y el problema es registrado en el log de errores. |
| Al buscar un carro por una placa pero no existe un carro con la placa especificada. | Mensaje al usuario informando el error. El problema es registrado en el log de errores. |
| En el método buscarPuestoLibre, cuando no hay puestos disponibles. | La excepción se maneja internamente, no se informa al usuario ni se registra el error en el log. |

# Persistencia

1. **Archivo para importar datos del sistema.**

La información de los carros puede ser cargada de un archivo txt. La estructura del archivo es la siguiente:

```
tarifa=<tarifa>
valorEnCaja=<valor en la caja>
cantidadCarros=<n>
marca carro 1;modelo carro 1;placa carro 1
marca carro 2;modelo carro 2;placa carro 2
…
marca carro n;modelo carro n;placa carro n
```

A continuación se muestra un ejemplo:

```
tarifa=2000
valorEnCaja=18000
cantidadCarros=5
BMW;Serie 1;AMD384
Chevrolet;Aveo;TRS778
Fiat;Punto;CMT115
Ford;Ka;WEM378
Honda;Civic;SDL792
```

Para la lectura de este archivo se debe manipular la hora de la siguiente forma: la hora actual del parqueadero se reestablece a la hora de inicio al cargar la lista, y con cada 5 carros ingresados la hora actual aumenta una unidad, para que al cargar una extensa lista los carros tengan horas de ingreso distintas.

2. **Archivo con el reporte del sistema.**

La información de las ganancias estimadas del parqueadero se guarda en un archivo de texto, cuyo formato debe ser el siguiente:

```
Reporte estimación ganancias actuales
Fecha: <Fecha y hora al momento de generar el reporte>

----------Carros en el Parqueadero----------
<Marca 1> <Modelo 1> - <Placa 1>: <Valor a cancelar 1>
<Marca 2> <Modelo 2> - <Placa 2>: <Valor a cancelar 2>
…
<Marca n> <Modelo n> - <Placa n>: <Valor a cancelar n>
---------TOTAL: <Suma de los valores a cancelar>----------
```

A continuación se muestra un ejemplo del reporte:

```
Reporte estimación ganancias actuales
Fecha: Mon Jan 29 17:00:12 COT 2018

----------Carros en el Parqueadero----------
Volvo V90 - WMS394: $4800.0
Tesla Model S - SAG476: $4800.0
Honda Accord - DMW965: $2400.0
----------TOTAL: $ 12000.0----------
```

### 3. Archivo de log de errores

Cada vez que se genera una excepción por problemas de persistencia o por errores en formatos de archivo, el programa debe registrar el problema en un archivo de log de errores (de tipo texto). El formato de este archivo es el siguiente:

```
--//ERROR//-- @ <Fecha en la que se generó>: <Causa de la excepción>.
```

Cuando se intenta ingresar un carro con una placa ya existente o se busca un carro por una placa no existente, el formato se extiende de la siguiente forma:

```
--//ERROR//-- @ <Fecha en la que se generó>: <Causa de la excepción>; Placa: <placa especificada>.
```

Cuando se intenta ingresar o sacar un carro pero el parqueadero ya se encuentra cerrado, o se intenta ingresar un carro al parqueadero lleno, el formato se extiende de la siguiente forma:

```
--//ERROR//-- @ <Fecha en la que se generó>: <Causa de la excepción>; No pudo ser <acción> el carro con placa <placa del carro>.
```

A continuación se muestra cómo sería un archivo de log de errores generado por la aplicación:

```
--//ERROR//-- @ Mon Jan 29 17:01:08 COT 2018: Ya se encuentra un carro en el parqueadero con la placa especificada.; Placa: WMS394.
--//ERROR//-- @ Mon Jan 29 17:02:11 COT 2018: Parqueadero Lleno; No pudo ser ingresado el carro con placa AMS642.
--//ERROR//-- @ Mon Jan 29 17:08:03 COT 2018: Error en el formato del archivo.
```

# Interfaz


Parqueadero
PARQUEADERO
Parqueadero
5
6
13
15
22
27
29
33
39
40
Carros en el parqueadero
Audi TTS - ROV662 10:00
Aston Martin Vanquish - MSD27
Fiat 500 - QMW394 10:00
Hyundai I40 - YMS329 11:00
Volvo V90 - WMS394 11:00
Volkswagen Beetle - SFM473 11
Toyota Prius - SNM286 11:00
Renault Twingo - AGD767 12:00
Peugeot 208 - JAJ846 12:00
Nissan PULSAR - MSJ238 12:00
Mercedes Clase A - SAH823 13:
Kia Picanto - OQW346 13:00
Hyundai Veloster - QWU723 13:
Honda Jazz - AGD673 13:00
Fiat Bravo - JZV784 14:00
Citroen C4 - LEJ836 14:00
Hora actual:
14:00
Avanzar
Siguiente Día
Tarifa:
$1800
Cambiar
Ingresar Carro
Sacar Carro
Información
Valor en Caja: $ 124500.0
Puestos Vacios: 10
Ordenamientos
Ordenar por marca
Ordenar los carros
Opciones
Cargar
Generar Reporte
Opción 1
Opción 2
Búsquedas
Buscar por placa
Buscar un carro