



Ejercicio: N8_cupiViajes

Autor: Equipo Cupi2

Fecha: Febero 2016

Enunciado

Se necesita construir una aplicación para manejar las reservas hechas en una agencia de viajes. La aplicación debe permitir agregar nuevas reservas de viajes, buscar reservas y ordenar las reservas dado un criterio.

Para cada reserva se maneja la siguiente información:

- La cédula del cliente que hace la reserva. Solo puede existir un número de cédula asociado a una reserva
- El nombre del cliente que hace la reserva.
- Hotel dónde se hizo la reserva.
- Fecha de llegada al hotel.
- Cantidad de noches de estadía en el hotel.
- Cantidad de adultos que van a viajar.
- Cantidad de niños que van a viajar.
- Estado de la reserva. Este puede ser Sin pagar, pagado en efectivo o pagado con tarjeta

Costo de la reserva. Este se calcula como el costo por noche en el hotel multiplicado por la cantidad de noches de estadía.

Aerolínea por la que se va a viajar. Actualmente, la agencia de viajes tiene convenio con las siguientes aerolíneas: Avianca, Lan, Viva Colombia, Satena, Jet Blue, Iberia y Air France.

- Del hotel se conoce la siguiente información:
- Nombre del hotel.
- Ciudad en la que se encuentra el hotel.
- Cantidad de estrellas.
- Costo por noche.
- Foto del hotel.

Se requiere que la aplicación permita:

- Agregar una reserva.
- Visualizar una reserva.
- Buscar la reserva de un cliente dado.
- Buscar la primera reserva de un hotel que se encuentra en una ciudad dada.
- Buscar todas las reservas de una aerolínea dada.
- Buscar la reserva con mayor cantidad de personas.
- Buscar la reserva con menor cantidad de personas.
- Ordenar la lista de reservas según el criterio, el orden y el algoritmo indicado.
 - Ordenes: ascendente o descendente
 - Criterios: Costo, cantidad de personas, nombre del cliente
 - Algoritmos: Burbuja, Selección, Inserción, Burbuja bi-direccional, Gnome
- Importar la información inicial de los hoteles.
- Generar el reporte de las reservas hechas.

La información de las reservas debe ser persistente y el proceso debe ser completamente transparente para el usuario. Esto quiere decir que el programa debe ser capaz de guardar la información del mundo en un archivo binario cada vez que el usuario termina la ejecución del mismo y de 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.

Manejo de errores

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.
Al intentar salvar el estado final del mundo, hay un error de entrada / salida.	Mensaje al usuario informando el error. Mensaje al usuario informando el error y preguntando si desea terminar la ejecución sin salvar el estado del mundo.
Al abrir el archivo .csv con los datos de los hoteles que se pueden reservar.	Mensaje al usuario informando el error. Se carga la información de manera parcial de los hoteles hasta el punto que se creó el error.
Al cargar un archivo .csv con un formato incorrecto.	Mensaje al usuario informando el error. Se carga la información de manera parcial de los hoteles hasta el punto que se creó el error.
Al intentar escribir el archivo de reporte.	Mensaje al usuario informando el error. Se mantiene la información ya registrada hasta el punto que se creó el error.
Al crear una reserva nueva, se presenta un error si la reserva tiene una cédula que ya está en el sistema.	Mensaje al usuario informando el error. La reserva no es agregada y el problema es registrado en un log de errores.

Los errores serán manejados con excepciones propias del problema.

Persistencia

1. Archivo para importar datos del sistema.

La información de los hoteles debe ser cargada de un archivo **.csv**. La estructura del archivo es la siguiente:

```
Nombre del hotel; ciudad del hotel; cantidad de estrellas; costo por noche; ruta de la imagen
```

A continuación se muestra un ejemplo:

```
Hotel las Américas;Cartagena;4;150000;./data/imagenes/americas.jpg
Hilton Garden Inn;Orlando;4;260000;./data/imagenes/hilton.jpg
Royal Decameron Barú;Barú;4;878125;./data/imagenes/baru.jpg
```

2. Archivo con el reporte del sistema.

La información de los reportes de las reservas hechas se guarda en un archivo de texto, cuyo formato debe ser el siguiente:

```
Reporte reservas hechas
Fecha:<fecha que se hace el reporte>
=====
Cédula del cliente: <cédula del cliente de la reserva 1>
Nombre del cliente: <nombre del cliente de la reserva 1>
Total personas en el viaje: <total personas en la reserva 1>
Noches de estadía: <noches de estadía en el hotel de la reserva 1>
Aerolínea: <aerolínea de la reserva 1>
Nombre del hotel: <nombre del hotel de la reserva 1>
Fecha de llegada: <fecha de llegada al hotel de la reserva 1>
Costo total de la reserva: <costo total de la reserva 1>
=====
Cédula del cliente: <cédula del cliente de la reserva k>
Nombre del cliente: <nombre del cliente de la reserva k>
Total personas en el viaje: <total personas en la reserva k>
Noches de estadía: <noches de estadía en el hotel de la reserva k>
Aerolínea: <aerolínea de la reserva k>
Nombre del hotel: <nombre del hotel de la reserva k>
Fecha de llegada: <fecha de llegada al hotel de la reserva k>
Costo total de la reserva: <costo total de la reserva k>
=====
```

A continuación se muestra un ejemplo del reporte:

```
Reporte reservas hechas
Fecha: Thu Feb 18 05:36:00 COT 2016
=====
Cédula del cliente: 1
Nombre del cliente: Juan
Total personas en el viaje: 4
Noches de estadía: 2
Aerolínea: Avianca
Nombre del hotel: Royal Decameron Barú
Fecha de llegada: Wed Mar 02 05:33:29 COT 2016
Costo total de la reserva: 1756250.0
=====
Cédula del cliente: 2
Nombre del cliente: Ana
Total personas en el viaje: 3
Noches de estadía: 4
Aerolínea: Avianca
Nombre del hotel: Hilton Lima Miraflores
Fecha de llegada: Thu Mar 03 05:34:41 COT 2016
Costo total de la reserva: 1975372.0
=====
```

Interfaz

The screenshot shows the CupiViajes web application interface. At the top, there is a header with a logo featuring a yellow airplane and a smiling sun character, and the text "CUPI VIAJES". Below the header, the interface is divided into several sections:

- Reservas:** A list of reservations with two entries: "12345. Juan (Hotel las Américas)" and "67890. Ana (Royal Decameron Barú)". Below the list are buttons for "Ordenar lista" and "Nueva reserva".
- Información de la reserva:** A form displaying reservation details for the selected reservation (67890).
 - Cédula del cliente: 67890
 - Nombre del cliente: Ana
 - Fecha de llegada: 15 / Jun / 2017
 - Noches de estadía: 7
 - Cantidad adultos: 1
 - Cantidad niños: 0
 - Aerolínea: jetBlue
 - Costo total de la reserva: 6146875.0
 - Estado de la reserva: Sin paqar (with a dropdown arrow and a "Cambiar" button)
 - Hotel reservado:**
 - Ciudad: Barú
 - Hotel: Royal Decameron Barú (with a 4-star rating and a small image of the hotel)
 - Costo por noche: 878125.0
- Opciones:** A row of four buttons: "Buscar reserva", "Generar reporte", "Opción 1", and "Opción 2".

Diálogo para crear una nueva reserva.

The screenshot shows the "Crear nueva reserva" dialog box. It contains the following fields and controls:

- Agregar nueva reserva:** A section header.
- Hotel:** A text input field.
- Buscar:** A button to search for the hotel.
- Cédula del cliente:** A text input field.
- Nombre del cliente:** A text input field.
- Fecha de llegada:** A date picker showing "febrero 5, 2016".
- Noches estadía:** A text input field.
- Adultos:** A dropdown menu with "1" selected.
- Niños:** A dropdown menu with "0" selected.
- Aerolínea:** A dropdown menu with "Avianca" selected.
- Guardar reserva:** A button to save the reservation.

Diálogo para crear un nuevo reporte.

