



## Nivel 8- CupiViajes

### Descripción

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.

### Objetivos

El objetivo de este ejercicio es que el estudiante comprenda y adquiera práctica en:

- El desarrollo de aplicaciones siguiendo un proceso incremental.
- La creación y captura de distintos tipos de excepciones para informar al usuario de manera conveniente cualquier problema detectado.
- La construcción de métodos para hacer persistir la información del modelo del mundo por medio del mecanismo de serialización.
- La construcción de métodos para importar y exportar información desde y hacia archivos secuenciales de texto, utilizando las clases que provee Java para manejo de archivos y manipulación de cadenas de caracteres.

Los siguientes pasos conforman el plan sugerido para desarrollar el ejercicio. La idea es ir desarrollando y probando incrementalmente los métodos de las clases.

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

### Preparación

1. Ejecute del sitio web del curso el archivo demo de la aplicación para conocer el funcionamiento esperado del programa.
2. Descargue del sitio web del curso el esqueleto del ejercicio (del enlace llamado **n8\_cupiViajes\_esqueleto**) del ejercicio del sitio del curso. Descomprima este archivo e importe el proyecto llamado **n8\_cupiViajes** en Eclipse.
3. Lea el enunciado del problema disponible en:
4. **n8\_cupiViajes/docs/specs/Descripcion.pdf**.
5. Estudie el documento de requerimientos funcionales disponible en:
6. **n8\_cupiViajes/docs/specs/RequerimientosFuncionales.pdf**.
7. Estudie todos los modelos disponibles en la carpeta **n8\_cupiViajes/docs/specs/**. Identifique las clases, relaciones entre clases, constantes, atributos y métodos
8. Asegúrese de tener activado el uso de aserciones para la ejecución del programa. Ver el tutorial en: [http://cupi2.uniandes.edu.co/sitio/images/cursosCupi2/apo2/tutoriales/n7\\_assert.pdf](http://cupi2.uniandes.edu.co/sitio/images/cursosCupi2/apo2/tutoriales/n7_assert.pdf)

## Parte1: Ordenamiento

- a. Incluya en su ejercicio la clase **Ordenador** que usted desarrollo durante el nivel 7 del curso. Dado que el ordenador es genérico no tendremos problema en usarlo nuevamente

## Parte2: Creación de nuevas excepciones

- a. Apoyándose en los modelos entregados complete la clase **PersistenciaException**. Tenga en cuenta que esta excepción manejará todos los errores relacionados con la lectura y escritura de archivos.
- b. Apoyándose en los modelos entregados cree la excepción **ClienteTieneReservaException**. Tenga en cuenta que esta excepción manejará le error asociado a crear dos reservas con la misma cédula
- c. En la clase **CupiViajes** complete los **//TODO Parte2.C**
- d. En la clase **InterfazCupiCiajes** complete los **//TODO Parte2.D**
- e. En la clase **CupiViajesTest** complete los **//TODO Parte2.E**

## Parte 3: Implementación de persistencia

- a. En la clase **ManejadorArchivosTexto** complete los **//TODO Parte3.A**
- b. Para este ejercicio serializaremos todo el objeto CupiViajes desde la interfaz. Identifique que clases deben ser serializables y modifique su código para asegurar que se puedan serializar. En la clase **Serializador** complete los **//TODO Parte3.B**
- c. En la clase **Serializador** complete los **//TODO Parte3.C**
- d. En la clase **ManejadorArchivosTexto** complete los **//TODO Parte3.D**

## Parte 4: Herencia y métodos estáticos

- a. En la clase InterfazCupiViajes REVISE los **//TODO Parte4.A**
- b. En la clase InterfazCupiViajes complete los **//TODO Parte4.B**

## Validación

Para comprobar el funcionamiento de su ejercicio usted puede:

1. Ejecutar las pruebas automáticas disponibles en el ejercicio (validación funcional del mundo) que le permitirán verificar la correcta implementación de sus métodos. Para ejecutar las pruebas en el entorno de Eclipse presione click derecho sobre el paquete "uniandes.cupi2.observatorio.test" (o alguna de sus clases internas), seleccione la opción *Run as* y posteriormente la opción *JUnitTest*. En el siguiente vínculo encuentra un video que le explica cómo ejecutar e interpretar las pruebas automáticas:

<https://www.youtube.com/watch?v=h3r7wSFaIOo>

2. Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz.

## Entrega

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

1. Construya el archivo entregable con el ejercicio completo. No olvide revisar que su entrega cumple con lo especificado en las normas del curso referentes a entregas de ejercicios. Consultar:

<http://cupi2.uniandes.edu.co/sitio/index.php/cursos/apo1/normas-administrativas>

2. Entregue el archivo del ejercicio vía SicuaPlus, de acuerdo con las normas, fecha y hora de entrega.

**NOTA.** No olvide:

1. Renombrar el archivo de entrega con su login de estudiante. El nombre del archivo que contiene el ejercicio debe cumplir el siguiente formato:

n<nivel del ejercicio>\_<login estudiante>.zip  
(por ejemplo: n8\_tsuarez.zip)

2. En el siguiente vínculo encuentra un video que le explica cómo indentar el código de su proyecto:

<https://www.youtube.com/watch?v=BH9H0e-Z56E>