



## Ejercicio Nivel 7-CupiViajes

### Descripción global

El Equipo Cupi2 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:

- Conceptos vistos en el curso anterior (Mundo e Interfaz).
- Desarrollará una aplicación siguiendo un proceso incremental.
- Construirá los invariantes de las clases del mundo del ejercicio.
- Utilizará la instrucción *assert* de Java para verificar invariantes.
- Desarrollará pruebas unitarias en Junit para las clases del ejercicio.
- Entenderá y aplicará el concepto de comparación de objetos.
- Entenderá y desarrollará tres algoritmos de ordenamiento (burbuja, inserción y selección).
- Entenderá y desarrollará algoritmos de búsqueda binaria y secuencial sobre una lista ordenada o no ordenada.
- Aprenderá a representar un objeto como un texto con el método `toString()`.
- Utilizará `JList` y `JScrollPane` para presentar listas en la interfaz gráfica.

Los siguientes pasos conforman el plan sugerido para desarrollar el ejercicio. La idea es ir desarrollando y probando incrementalmente los métodos de las clases. **No se preocupe si las clases de la interfaz o de las pruebas (test) tienen errores. Estos desaparecerán cuando termine (correctamente) los cambios en el modelo del mundo.**

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

### Preparación

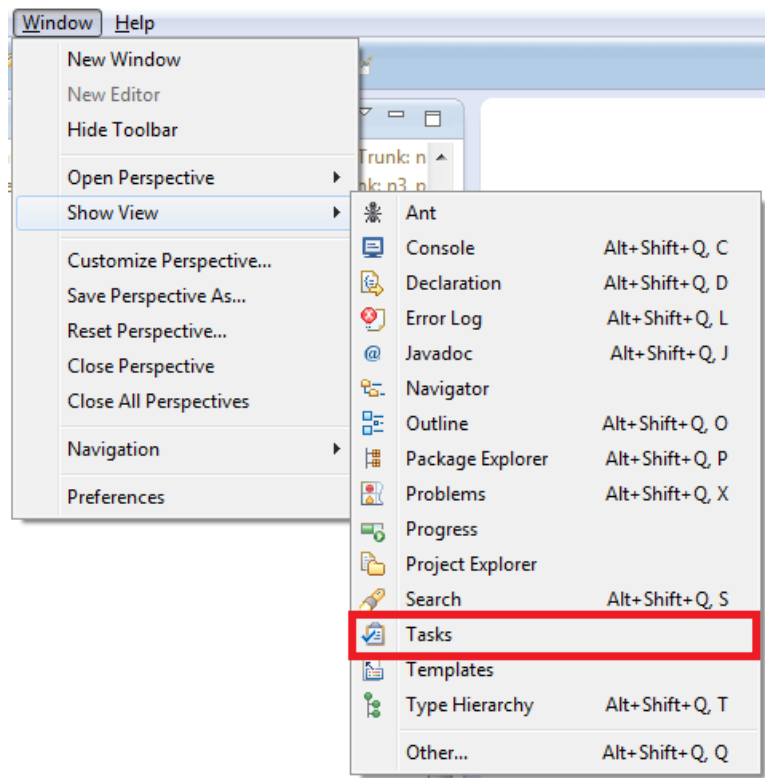
1. Reproduzca el demo de la aplicación en el sitio web del curso para conocer el funcionamiento esperado del programa.
2. Descargue del sitio web del curso el esqueleto del ejercicio (del enlace llamado **n7\_cupiViajes\_esqueleto**) del ejercicio del sitio del curso. Descomprima este archivo e importe el proyecto llamado **n7\_cupiViajes** en Eclipse.
3. Lea el enunciado del problema disponible en:  
**n7\_cupiViajes/docs/specs/Descripcion.pdf**.
4. Estudie el documento de requerimientos funcionales disponible en:  
**n7\_cupiViajes/docs/specs/RequerimientosFuncionales.pdf**.

5. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en: **n7\_cupiViajes/docs/specs/ModeloConceptual.jpg**. Identifique las clases, relaciones entre clases, constantes, atributos y métodos
6. 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)
7. Desde Eclipse revise la documentación de las clases del mundo. Esto le permitirá entender para qué sirve cada método y cada uno de los atributos. También puede generar la documentación del proyecto (archivos .html) ejecutando el programa **doc** que se encuentra en **n7\_cupiViajes/bin/win** (para Windows) y en **n7\_cupiViajes/bin/man** (para Mac).

Dentro del código del esqueleto se encuentran indicados los puntos donde usted debe realizar alguna modificación (añadir atributos, completar métodos, construir nuevos métodos, etc.), por medio de comentarios de la siguiente forma:

**// TODO ParteX PuntoY:** Breve explicación de la modificación que debe realizar.

Donde X hace referencia a una parte de la guía de trabajo, y Y hace referencia a un punto de la parte X. Para visualizar los **TODOs** vaya al menú Window -> Show View -> Tasks como se muestra en la siguiente figura:



En el siguiente vínculo encuentra un video que le explica cómo consultar los TODOs de un ejercicio:

<https://www.youtube.com/watch?v=pV54O42D2ow>

## Parte 1: Pruebas unitarias

1. Complete la clase **CompradorHotelNombre**
  - a. Completar los TODOs que comienzan por: `//TODO Parte1 Punto1.`
2. Complete la clase **CupiViajesTest**
  - a. Completar los TODOs que comienzan por: `//TODO Parte1 Punto2.`

## Parte 2: Comparadores

1. Complete la clase **ComparadorHotelNombre**
  - Completar los TODOs que comienzan por: `//TODO Parte2 Punto1.`
2. Complete la clase **ComparadorReservaNombreCliente**
  - Completar los TODOs que comienzan por: `//TODO Parte1 Punto2.`
3. Complete la clase **ComparadorReservaCantidadPersonas**
  - Completar los TODOs que comienzan por: `//TODO Parte1 Punto3.`
4. Declare la clase **ComparadorReservaCosto** la cual es un comprador de **ReservaViaje** que es capaz de comparar reservas por su costo total.

## Parte 3: Enumeraciones

1. Complete la enumeración **CriterioOrdenReserva**
  - a. Completar los TODOs que comienzan por: `//TODO Parte3 Punto1.`
2. Complete la clase **ReservaViaje**
  - a. Completar los TODOs que comienzan por: `//TODO Parte3 Punto2.`

## Parte 4: Ordenamientos

1. Complete la clase **Ordenador**

- a. Completar los TODOs que comienzan por: `//TODO Parte4 Punto1`
2. Complete la clase **InterfazCupiViajes**
  - a. Completar los TODOs que comienzan por: `//TODO Parte4 Punto2`
3. Complete la clase **ReservaViaje**
  - b. Completar los TODOs que comienzan por: `//TODO Parte4 Punto3`

## Parte 5: Búsquedas

1. Complete la clase **CupiViajes**
  - a. Completar los TODOs que comienzan por: `//TODO Parte5 Punto1.`

## Parte 6: Interfaz Gráfica

1. Complete la clase **PanellistaReservas**
  - a. Completar los TODOs que comienzan por: `//TODO Parte6 Punto1.`
2. Complete la clase **InterfazCupiViajes**
  - a. Completar los TODOs que comienzan por: `//TODO Parte6 Punto2.`

## Parte 7: Verificación de invariantes

1. Complete la clase **Hotel**
  - a. Completar los TODOs que comienzan por: `//TODO Parte7 Punto 1ª`
2. Complete la clase **ReservaViaje**
  - a. Completar los TODOs que comienzan por: `//TODO Parte7 Punto 1a`

## 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 clic derecho sobre el paquete `"uniandes.cupi2.cupiViajes.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: n5\_tsuares.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>