

Objetivos de unidad

Este ejercicio busca trabajar los objetivos del **Capítulo 5: Estructuras y algoritmos recursivos** y **Capítulo 6: Bases de datos y distribución básica**

Para la entrega del ejercicio, revise el documento publicado en moodle: Normativas de MiniProyectos.

Enunciado del problema

Nombre del proyecto: Aerolínea EntreNubes

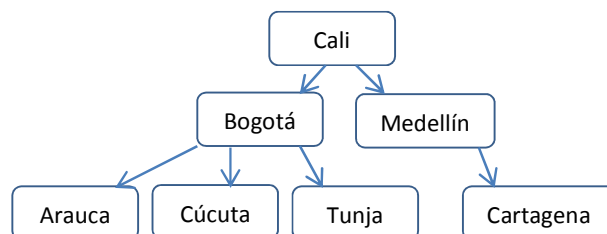
Cansado de las altas tarifas en los vuelos nacionales y el incumplimiento de horarios en las aerolíneas tradicionales, el señor Mediano Gonzalez se motivó a incursionar en el negocio con su Aerolínea *EntreNubes*.

Rodeado de talentosos asesores, todo está siendo preparado para la gran inauguración. Hasta ahora sólo tiene un contratiempo que espera usted le ayude a solucionar, se ha quedado sin los ingenieros que le iban a desarrollar la aplicación de rutas para sus clientes. Un amigo le ha recomendado su trabajo y ahora la tarea del desarrollo de la aplicación es suya.

La aplicación está pensada para ser la guía de viaje de los nuevos clientes de EntreNubes. En el módulo de administración se podrán crear ciudades y conectarlas al árbol de rutas. En el módulo de clientes, se podrá consultar si la Aerolínea ofrece un servicio entre una ciudad y otra, y los datos del viaje.

Detalles para el módulo de administración: EntreNubes empezará su operación únicamente desde Cali, así que la aplicación por ahora siempre considerará a Cali como punto de inicio. Luego, un administrador podrá crear la o las ciudades a las que se llega de forma directa desde Cali, por ejemplo, Bogotá y Medellín. Más tarde, otro administrador puede agregar que desde Bogotá se llega a Cúcuta y así en adelante se podrán ir creando rutas con la única condición de que la ciudad de origen del vuelo ya exista en el sistema.

Árbol de rutas de ejemplo:

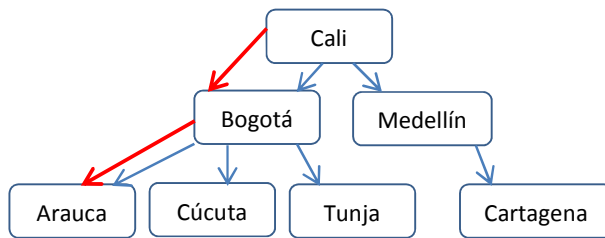


Al momento de agregar la nueva ciudad al árbol de rutas, se debe indicar el nombre del aeropuerto de llegada, la distancia existente entre las dos ciudades (la ciudad origen y destino) y cuánto tiempo de vuelo representa.

Los administradores también tendrán la posibilidad de eliminar ciudades del árbol de rutas, pero el sistema sólo permitirá la eliminación si desde esa ciudad no se llega a ninguna otra. En el árbol de ejemplo, sólo se podría eliminar, Arauca, Cúcuta, Tunja y Cartagena.

Detalles para el módulo de clientes: El cliente selecciona la ciudad origen y la ciudad destino para la cual desea hacer la consulta, el sistema debe revisar su árbol de rutas y determinar si hay una ruta directa entre las ciudades. Como resultado de la operación se espera que en pantalla se muestren las ciudades por las que debe pasar el vuelo, iniciando en ciudad origen y terminando en ciudad destino, adicional con la información de lo que sería el tiempo de vuelo de todo el trayecto (más 1 hora por cada conexión entre ciudades) y la distancia que se recorrería.

Ejemplo 1: Ruta de Cali a Arauca



Ruta: Cali – Bogotá – Arauca

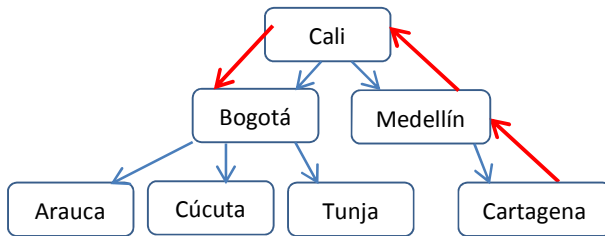
Datos del viaje

Tiempo de vuelo:

Distancia recorrida: ...

Si no hay una ruta directa, el sistema entonces deberá calcular el camino de la ciudad origen a Cali y de allí a la ciudad destino y mostrar al cliente toda la información del trayecto.

Ejemplo 2: Ruta Cartagena - Bogotá



Ruta: Cartagena – Medellín - Cali – Bogotá

Datos del viaje

Tiempo de vuelo:

Distancia recorrida: ...

RESTRICCIONES DEL PROBLEMA

1. Esta unidad evalúa el uso de estructuras recursivas específicamente Árboles. Para este ejercicio usted debe implementar esta estructura.
2. Esta unidad introduce elementos gráficos que deben ser incluidos dentro de su solución.
3. La interfaz gráfica debe permitir ver el árbol gráficamente.

Publicación: 25 de Marzo de 2014

Entrega 1: 35% - 1 de Abril 2014

Usted debe entregar los siguientes ítems:

- **Planning - Planeación del proyecto:**
 1. Cronograma de trabajo y los formatos PSP –Opcional-
Significa la asignación de responsabilidades específicas a cada uno de los integrantes del grupo, detalle de las tareas a realizar y fechas de revisión entre el grupo de estudiantes. Además de los formatos donde se registre su avance real en el desarrollo del proyecto.
 2. Documento de requerimientos funcionales y no funcionales con formato Cupi2.
- **Design - Diseño:**
 1. Diagrama de clases del mundo y de la interfaz.
- **Code - Código (Implementación):**
 1. Implementación de la interfaz gráfica.

Rubrica del ejercicio (entrega 1)

Criterio	Puntos
Planning - Planeación del proyecto	Subtotal 40
(Opc)Cronograma de trabajo y Formatos PSP	20
5 puntos por cada formato.	
Documento de requerimientos funcionales y no funcionales con formato Cupi2.	20
Design – Diseño	Subtotal 55
Estándar de la documentación	5
Compleitud de los requerimientos	10
Diagrama de clases del mundo del problema	25
Diagrama de clases de la interfaz	15
Code - Código (Implementación)	Subtotal 25
Implementación de la interfaz gráfica	25
TOTAL	120

Entrega 2: 65% - 12 de Abril 2014

Usted debe entregar: los mismos ítems de la entrega 1 (si les ha realizado modificaciones o no) **más** los siguientes:

- **Code - Código (Implementación):**
 1. Paquetes Java con la implementación de la solución (incluye el paquete test asociado a las pruebas).
- **Design – Diseño:**
 1. Diagrama de clases del mundo, de la interfaz y de las pruebas.
- **Test – Pruebas:**
 1. Diseño de los casos de prueba formato Cupi2 (escenarios y casos de prueba en formato Cupi2)

Rubrica del ejercicio (entrega 2)

Recuerde que en esta segunda entrega debe enviar su proyecto completo, con todas las partes detalladas a continuación. Si usted no modificó ninguna de las partes enviadas en la primera entrega, de igual modo debe enviarlas con esta segunda parte y serán calificadas nuevamente aunque esta nueva ponderación solamente será tomada en cuenta en la nota de la segunda entrega. La nota de la Primera Entrega no cambia. Por tanto, si usted actualizó alguna de estas partes y desea que le sean tomadas en cuenta estas mejoras, no olvide incluir los documentos actualizados (no los anteriores).

Criterio	Puntos
Planning - Planeación del proyecto:	Subtotal 10
(Opc) Cronograma de trabajo y Formatos PSP	5
Documento de requerimientos funcionales y no funcionales con formato Cupi2.	5
Design – Diseño	Subtotal 21
Estándar de la documentación	2
Compleitud de los requerimientos	2
Diagrama de clases del mundo del problema	6
Diagrama de clases de la interfaz	3
Casos de uso y Diagrama de clases de las pruebas	8
Code - Código (Implementación)	Subtotal 59
Cumplimiento de los estándares de programación	4
Cumplimiento de la documentación	4
Cumplimiento de las restricciones del enunciado en cuanto a los temas de la unidad	10
Funcionalidad en general de la aplicación	25
Funcionalidad en general de las pruebas	16
Test – Pruebas:	Subtotal 15
Estándar de la documentación	5
Calidad de los escenarios y casos de prueba	10
TOTAL	105