Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Estructura de datos P
Vacaciones de junio 2019
Catedrático: Ing. Luis Espino
Auxiliar: Javier Navarro

Coline Airport

Proyecto 2 de laboratorio

Contenido

1 Objetivos

1.1 Objetivo general

Que el estudiante aplique los conocimientos adquiridos sobre estructuras de datos en aplicaciones que permitan manipular la información de forma óptima para el ordenador y permita mostrarla de forma sencilla.

1.2 Objetivos específicos

- Utilizar estructuras de datos como matrices y árboles para almacenar la información en tiempo de ejecución.
- Generar reportes que representen el estado de la memoria de forma gráfica.
- Que el estudiante genere estructuras genéricas y adapte la implementación de dichas estructuras a las necesidades del problema.
- Que el estudiante domine el lenguaje de programación Java.
- Que el estudiante domine el desarrollo para la plataforma Android.
- Que el estudiante genere un nivel de abstracción alto.

2 Descripción

Se requiere de una aplicación móvil para gestionar las rutas de vuelo de una agencia de viajes internacionales. Dicha aplicación deberá de almacenar la información de las rutas de vuelo como tiempo de vuelo y costo del pasaje, además deberá de ser capaz de generar planes de vuelo de forma dinámica, siguiendo patrones benéficos de costo monetario o tiempo.

Existirá una aplicación de escritorio con interfaz gráfica la cual se usará para cargar los archivos de los países, destinos y origen y las reservaciones con la aplicación móvil se realizarán reservaciones y se consultarán las rutas de vuelo.

Un servidor web será el intermediario entre las consultas e ingreso de información que se realizaran desde la aplicación móvil y la carga y consultas de la información desde la aplicación de escritorio.

3 Componentes de la solución

Esta solución estará conformada por tres componentes principales:

- Aplicación de escritorio
- Aplicación móvil
- Web service, que alimente a las anteriores.

3.1 Funcionalidades de la aplicación de escritorio

3.1.1 Cargar nodos destino,

Esta funcionalidad permite recopilar los datos necesarios para realizar la inserción de los nodos del Árbol B (almacenado en el Web Service), que representan los países disponibles en el sistema.

3.1.2 Cargar rutas de vuelo

Esta funcionalidad permite recopilar los datos necesarios para relacionar los nodos destino unos con otros por medio de la creación de un nuevo registro a guardar en una matriz de adyacencia

3.1.3 Modificar ruta

Esta funcionalidad permite acceder a un nodo de la matriz de adyacencia (almacenada en el Web Service) y modificar su valor.

3.1.4 Eliminar ruta

Esta funcionalidad permite acceder a un nodo de la matriz de adyacencia (almacenada en el Web Service) y eliminar el nodo.

3.1.5 Cargar reservaciones tabla hash

Esta funcionalidad permite acceder a un nodo de la matriz de adyacencia (almacenada en el Web Service) y eliminar el nodo.

3.1.6 Ver reporte destinos

Esta funcionalidad permite acceder al árbol B (almacenado en el Web Service) y generar un reporte grafico con Graphviz, renderizando la imagen desde la aplicación.

3.1.7 Ver reporte rutas

Esta funcionalidad permite acceder a un nodo de la matriz de adyacencia (almacenada en el Web Service) y generar un reporte grafico con Graphviz, renderizando la imagen desde la aplicación.

3.2 Android

3.2.1 Ver mapa de destinos

Se mostrará la imagen en la aplicación Android.

3.2.2 Ver reservaciones

Agregar reservaciones selecciona destino y origen, y mostrará las rutas posibles, permitiendo elegir una para reservar.

3.2.3 Eliminar reservaciones

Ver reporte de reservaciones

3.3 Funcionalidades del Web Service JSP

Tanto de la aplicación de escritorio como la móvil se conectarán al web service para consultar y almacenar información, es decir, este contendrá las estructuras necesarias para el funcionamiento de la solución, además será el intermediario.

3.3.1 Módulo de gestión de rutas

En este módulo ofrecerá las siguientes opciones:

3.3.1.1Cargar nodos destino

Esta opción permite cargar los destinos de viaje desde un archivo CSV externo con la siguiente estructura:

[Código Destino], [Nombre del destino]

Ejemplo:

7720, Guatemala

7721, El Salvador

7722, Honduras

7723, Nicaragua

3.3.1.2Cargar rutas de vuelo

Esta opción permite cargar las rutas de vuelo (en caso de que el destino u origen de la ruta no exista no se debe insertar) desde un archivo CSV externo con la siguiente estructura:

[Código punto 1], [Código punto 2], [Costo del viaje en Q], [Tiempo de vuelo en minutos], [Nombre_piloto]

Nota: El nombre del piloto debe ser encriptado con el cifrado de Verman (Variante ASCII), usando para el alfabeto una codificación ASCII ver: Cifrado_Vernam. Ya que al mostrar la información en las aplicaciones no es necesario

Ejemplo:

7720, 7721, 200, 30, <Harry>

7720, 7722, 215, 45, <Navarro>

7723, 7721, 260, 75, <Nicole>

<>: encriptado.

Considere las rutas como bidireccionales, es decir si hay un viaje de Guatemala (7720) a El Salvador (7721) también hay uno de El Salvador a Guatemala.

3.3.1.3Modificar ruta

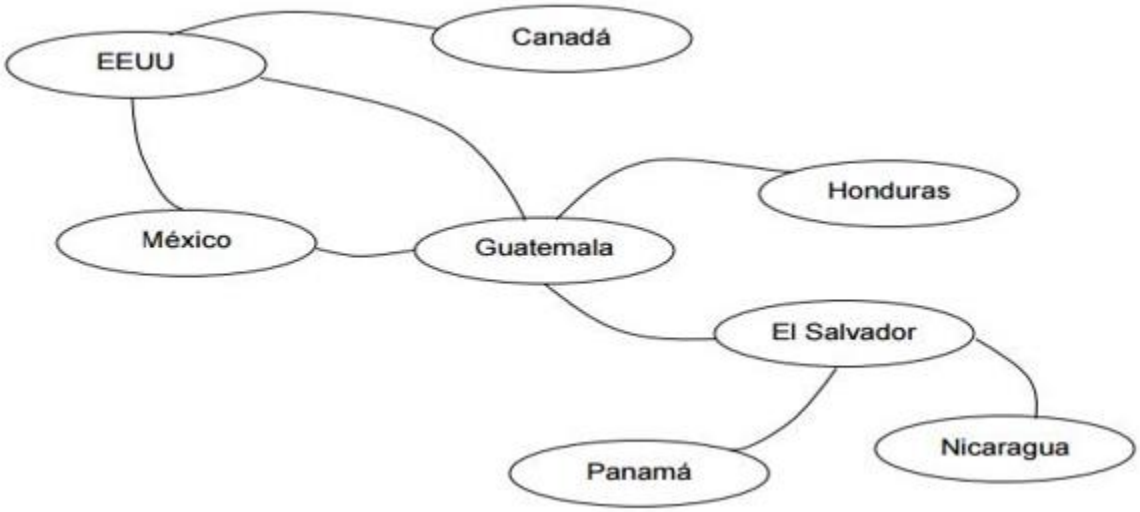
La aplicación deberá desplegar una lista donde se seleccione el nombre del nodo origen y el nombre del nodo destino, si la ruta existe se podrá modificar el tiempo de vuelo y el costo de viaje, si en dado caso no existe la ruta podrá crearse.

3.3.1.4Eliminar ruta

La aplicación deberá desplegar una lista donde se seleccione el origen como el destino de la ruta a eliminar (eliminación en la matriz de adyacencia).

3.3.1.5Generar mapa de destinos

Esta opción generará la gráfica del grafo de destinos y las rutas que estas conectan y la mostrara de forma amigable al usuario.



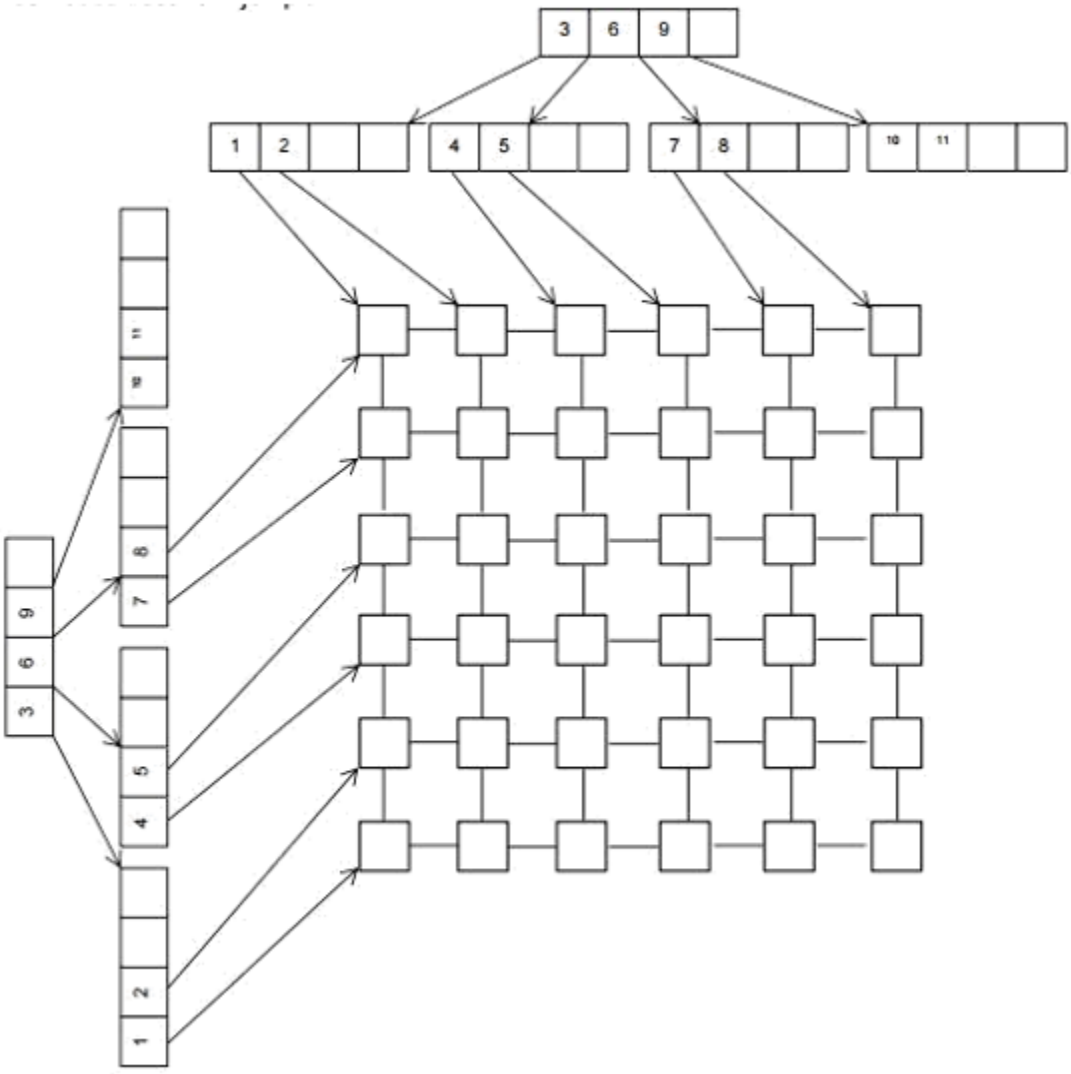
4 Almacenamiento

4.1 Grafo de rutas

Debido a que la aplicación debe brindar un tiempo de respuesta eficiente se requiere que el grafo de destinos y rutas de vuelo no sea almacenado en un grafo propiamente sino en una estructura con acceso más eficiente por lo cual se pretende realizar una matriz de adyacencia que represente al grafo, con indexación por árbol B.

Los nodos del árbol B almacenaran la información de los nodos destino, mientras los nodos de la matriz de adyacencia (matriz dispersa) almacenaran las rutas que conectan los nodos destino. La información que se guarda en el árbol B, es el código y nombre del país. En la matriz de adyacencia se guardará el costo, el tiempo y el nombre del piloto.

Ejemplo:



4.2 Almacenamiento de reservaciones

Las reservaciones serán almacenadas en una tabla hash con la siguiente información:

- Costo total del viaje
- Tiempo de vuelo
- Número de reservación (generado dinámicamente)
- Nombre del cliente
- Lista enlazada con el plan de vuelo

4.2.1 Función hash

La función hash a utilizar será función por división:

$$H\,(K)\;=\;K\;mod\;T$$

K = llave se tomará en cuenta la suma del ASCII de las tres primeras letras del nombre del cliente y eso será K.

T = Tamaño de la tabla

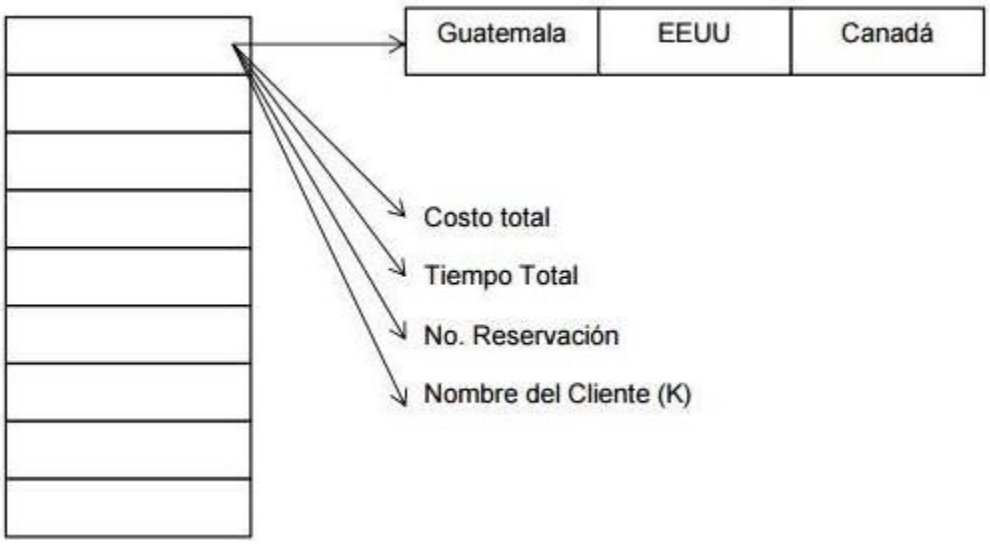
4.2.2 Resolución de colisiones

Será por direccionamiento abierto cuadrático K + (i+1) ^2

i = iteración

El tamaño inicial será 7, en caso de desbordamiento (suponga 50% de la capacidad de la tabla) se aumentará el tamaño al siguiente número primo.

Ejemplo:



4.3 Cargar reservaciones

Cargara las reservaciones desde un archivo CSV

Ejemplo

[Nombre_cliente], [Reservación (id_reservacion)], [Costo], [Tiempo], [Cód. país 1], [Cód. país 2],…, [Cód. país n]

Rocael, 25, 250, 60, 7720, 7721

Lester, 30, 700, 190, 7720, 7721, 7722, 7723

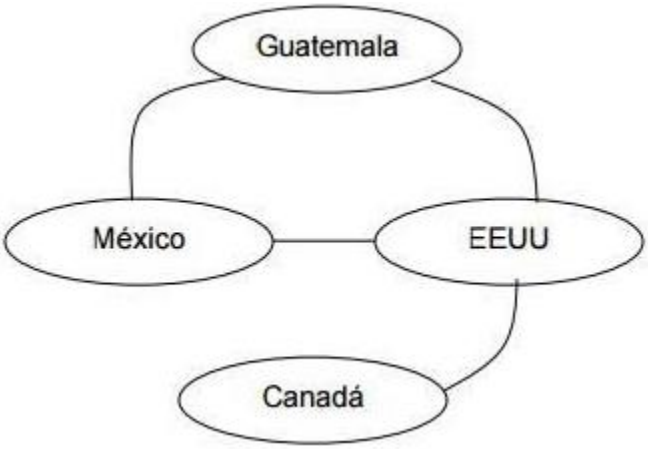
Alejandra, 4, 250, 60, 7720, 7722

4.4 Agregar reservaciones

Este módulo permitirá generar planes de vuelos en base al origen del vuelo y destino, por lo cual se le solicita que la aplicación despliegue un formulario donde se ingresen estas variables.

Una vez insertadas las variables, la aplicación desplegará los posibles planes de vuelo mostrando su costo y tiempo total, seguido de esto el usuario podrá realizar una reservación seleccionando alguno de estos planes.

Ejemplo, teniendo el siguiente grafo de rutas de vuelo:



Destino: Canadá

Origen: Guatemala

Planes de vuelo:

1. Guatemala > México > EEUU > Canadá
2. Guatemala > EEUU > Canadá

El usuario podrá escoger de la lista un plan de vuelo o podrá pedir la ruta de vuelo más barata o la ruta de vuelo con menor tiempo. Los tres casos se tomarán en cuenta en la calificación. Pueden utilizar cualquier algoritmo visto en clase, pero deben defender la razón por la cual la utilizaron y como lo utilizaron.

4.5 Eliminar Reservaciones

Por el nombre de usuario se elegirá una reservación a eliminar.

5 Otros Reportes

El sistema también dará la opción de generar los siguientes reportes:

- Reporte de destinos Genera la gráfica de los encabezados de la matriz de adyacencia (árbol B), debe desplegar tanto el código como el nombre.
- Reporte de Rutas Genera la gráfica de la matriz de adyacencia (tanto la matriz como los árbol B de indexación).
- Reporte de reservaciones Genera la gráfica de la tabla hash (se debe mostrar la posición, los datos de la reserva y la lista que desprende de esta).

6 Entregables

- Código Fuente (app móvil, app desktop y Web Service)
- Ejecutable
- Documentación: Se debe presentar documentación técnica que sea bien realizada. Esta información debe incluir:
    - Diagrama de clases
    - Diagrama de paquetes
    - Diagrama de colaboración
    - Análisis de variables (su uso y su envío)

7 Observaciones

- Lenguaje de programación a utilizar: Java
- Sistema Operativo: Libre
- IDE: Libres.
- Herramientas de reportes: Graphviz
- La gráfica debe mostrarse dentro de la aplicación, no buscarse en carpetas ajenas. La aplicación que no genere gráfica no podrá ser calificada, se debe utilizar Graphviz.
- Todas las estructuras deben de ser realizadas por el estudiante, sin el uso de librerías específicas de ningún IDE o Framework.
- Durante la calificación se harán preguntas para validar que el estudiante realizo el proyecto, de no responder correctamente anulara la nota obtenida en la o las secciones en la que aplique tal concepto.

- Proyectos que den una excepción NULL tendrán una nota de 0 puntos en la sección asociada.
- La aplicación será compilada y ejecutada al momento de la calificación.

8 Fecha y modo de entrega:

- Martes 9 de julio antes de las 8:00 horas vía Classroom, adjuntar entregables en un comprimido con nombre <carnet>.zip
- No se recibirán entregas tarde, sin excepción.
- Si no se entrega en Classroom, no se calificará.
- Copias tendrán nota de 0 puntos de laboratorio y serán reportadas al catedrático y a la Escuela de Sistemas.