Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas
Estructuras de Datos
Catedrático: Ing. Moisés Velásquez
Tutor Académico: José Morejón

Proyecto de laboratorio #2

Guatemala 29 de diciembre de 2018

Objetivos

Generales

- ✓ Que el estudiante aplique los conceptos vistos en clase respecto a estructuras dinámicas de datos.
- ✓ Que el estudiante aprenda y aplique los conceptos referentes a punteros y memoria dinámica en la solución de problemas.
- ✓ Que el estudiante aplique los conceptos y estructuras vistas en la resolución de problemas reales.

Específicos

- ✓ 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 / Kotlin.
- ✓ Que el estudiante domine el desarrollo para la plataforma Android.
- ✓ Que el estudiante genere un nivel de abstracción alto.

Descripción

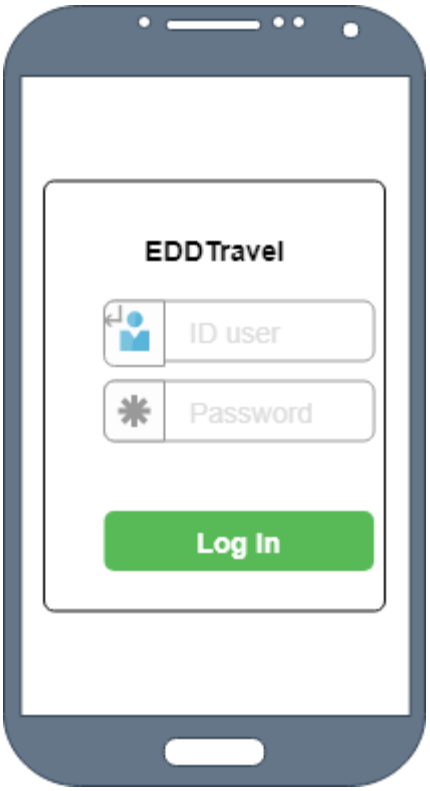
Se requiere de una aplicación móvil para gestionar las rutas de vuelo de una agencia de viajes internacionales. Dicha aplicación deberá de almacenar la información de las rutas de vuelo como tiempo de vuelo y costo del pasaje, además deberá de ser capaz de generar planes de vuelo de forma dinámica, siguiendo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

Funcionalidad

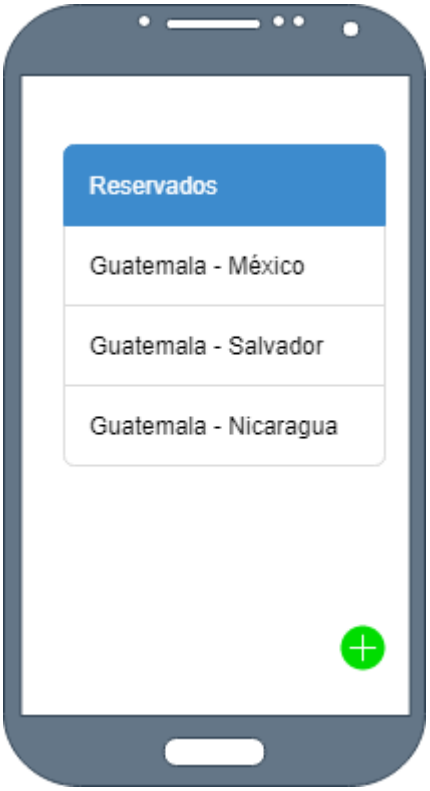
La aplicación deberá contar con los siguientes módulos:

Módulo de Cliente

Inicio de sesión

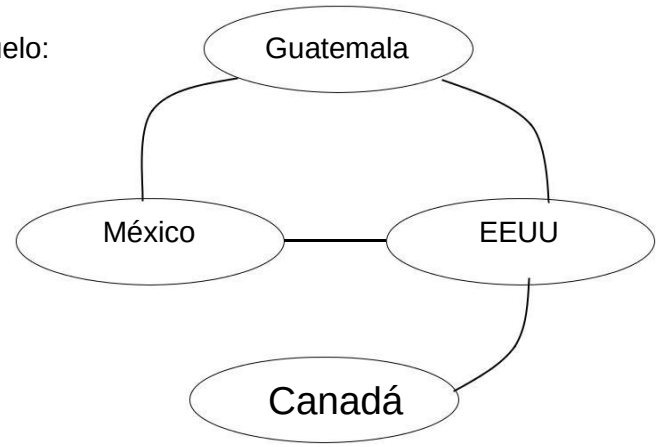


Al iniciar la aplicación debe poder presentar una lista de vuelos reservados o puede tener un botón para reservar vuelos.



Debe de poder ingresar el país origen y destino, la aplicación desplegara 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:

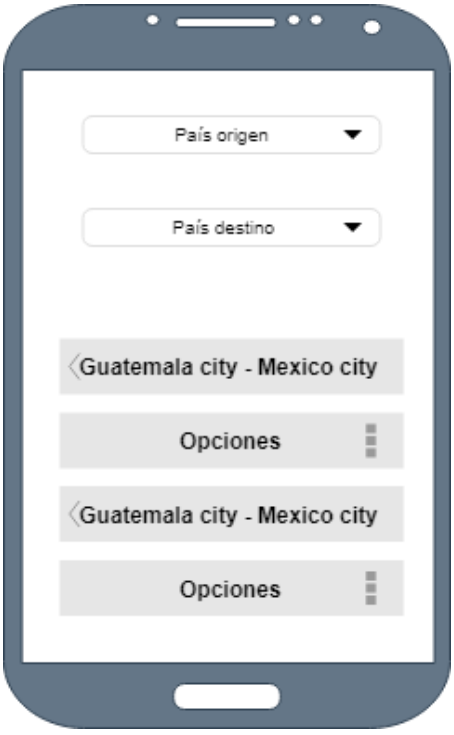


Destino: Canadá
Origen: Guatemala
Planes de vuelo:

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

Nota: debido a que pueden darse bucles al genera rutas considere que un plan de vuelo no puede sobrepasar las 5 rutas.

Para reservar vuelos puede tener orientarse en la siguiente figura.



En la parte de opciones puede desplegar reservar, visualizar la ruta y costo.

Módulo de Administrador (Web)

[user:admin,password:<carné>]

Debe poder crear un dashboard donde el administrador pueda de manera fácil y sencilla elegir la opción que desea trabajar en el sistema.

Módulo de gestión de rutas

En este módulo ofrecerá las siguientes opciones:

Cargar nodos destino

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

[Código], [Nombre del destino]

Ejemplo:

```
7720, Guatemala
7721, El Salvador
7722, Honduras
7723, Nicaragua
```

Debe de tener la opción de un formulario para este proceso de ingreso de nodos destino.

Cargar 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 $], [Tiempo de vuelo en minutos]

Ejemplo:

```
7720, 7721, 200, 30
7720, 7722, 215, 45
7723, 7721, 260, 75
```

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.

Deben crear un formulario de manera alterna para ingresar las rutas por nombres del destino del punto 1 y 2, costo del viaje y tiempo. Recomiendo select option para los nombres destino.

Tabla de rutas de vuelo

En esta tabla se debe poder listar los vuelos, en cada vuelo registrado pueden visualizar el mapa de destinos, editar o eliminar.

Modificar ruta

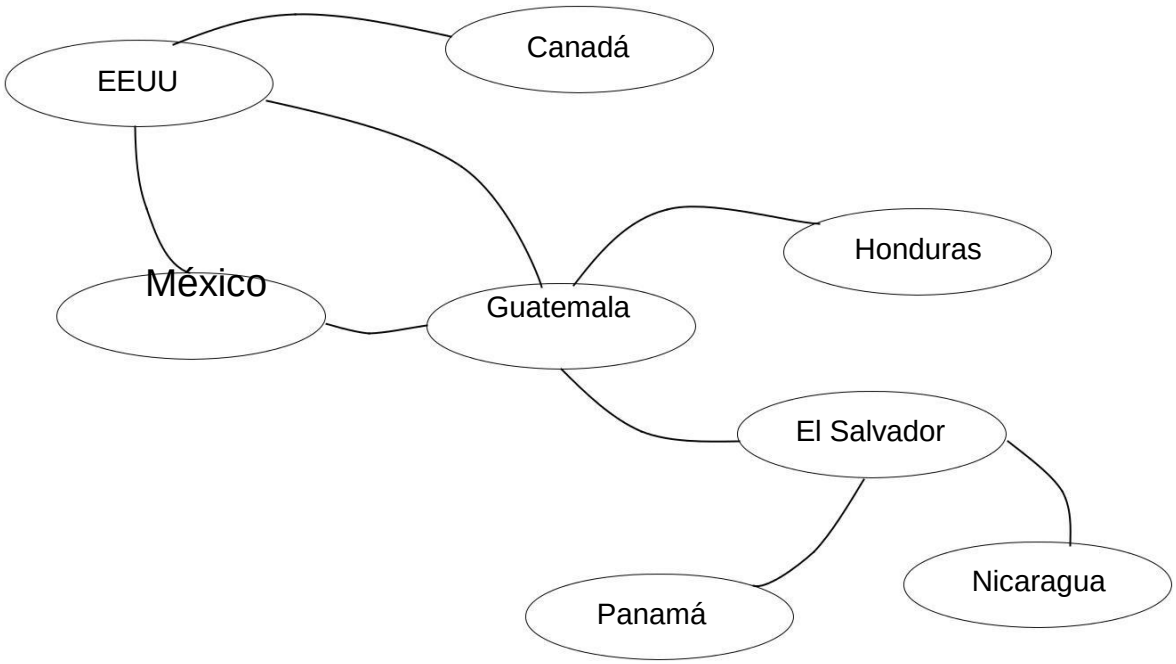
La aplicación deberá desplegar una lista donde se seleccione el nombre del nodo origen y el nombre del nodo destino, si la ruta existe se podrá modificar el tiempo de vuelo y el costo de viaje, si en dado caso no existe la ruta podrá crearse.

Eliminar ruta

La aplicación deberá desplegar una lista donde se seleccione el origen como el destino de la ruta a eliminar.

Generar 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.



Módulo gestión de usuarios

Desarrollar un apartado para la gestión de usuarios, el cual implica poder crear y visualizarlos en una tabla donde puedan editar o eliminar. En cualquier momento poder visualizar la gráfica de la estructura.
Datos para almacenar:
[ID(Alfanumérico),Nombre,password]

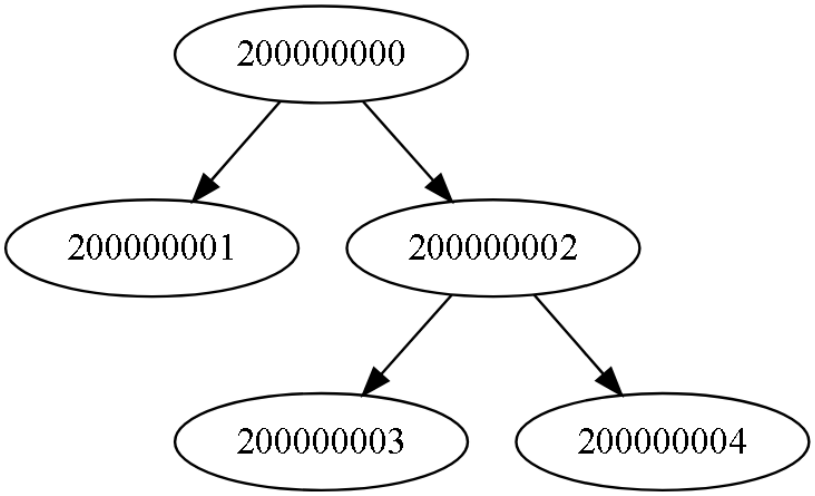
Módulo de gestión de reservas de vuelos

Desarrollar un apartado para la gestión de reservas de vuelos de usuarios, en los cuales puedan cancelar (pero no eliminar de la tabla) y visualizar la gráfica del vuelo mostrando la ruta que reservo. En este apartado no deben poder crear reservas dado que solo usuarios/móvil pueden realizar esta acción.

Almacenamiento

Usuarios

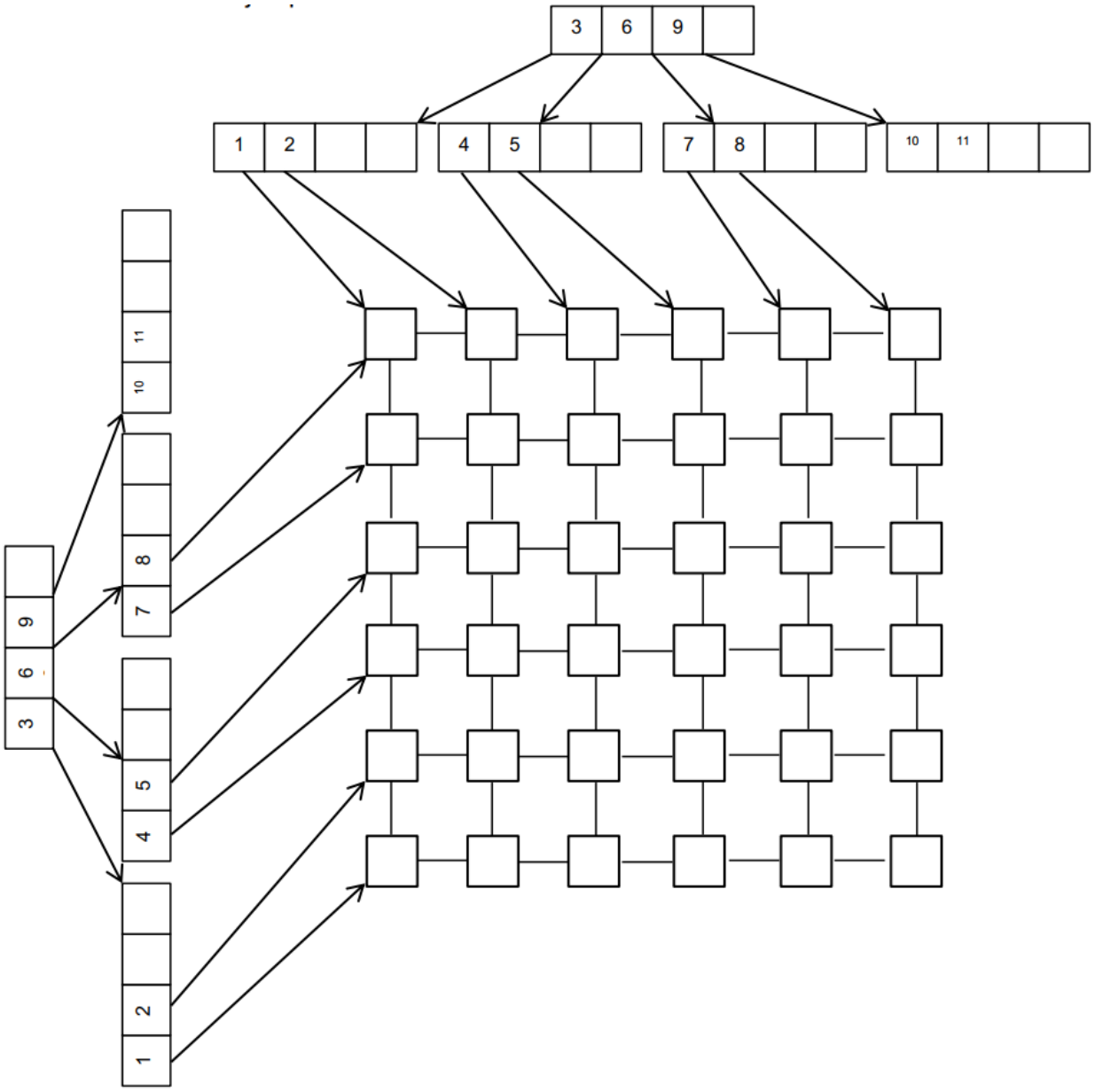
Los que pueden iniciar sesión dentro de la App de android deben de estar en una estructura Árbol AVL (tomar como pivote el ID) y mostrar toda la información del usuario.



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. Ejemplo:



Ejemplo de la estructura descrita anteriormente.

Almacenamiento de reservaciones

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

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

La función hash a utilizar será **función por división**

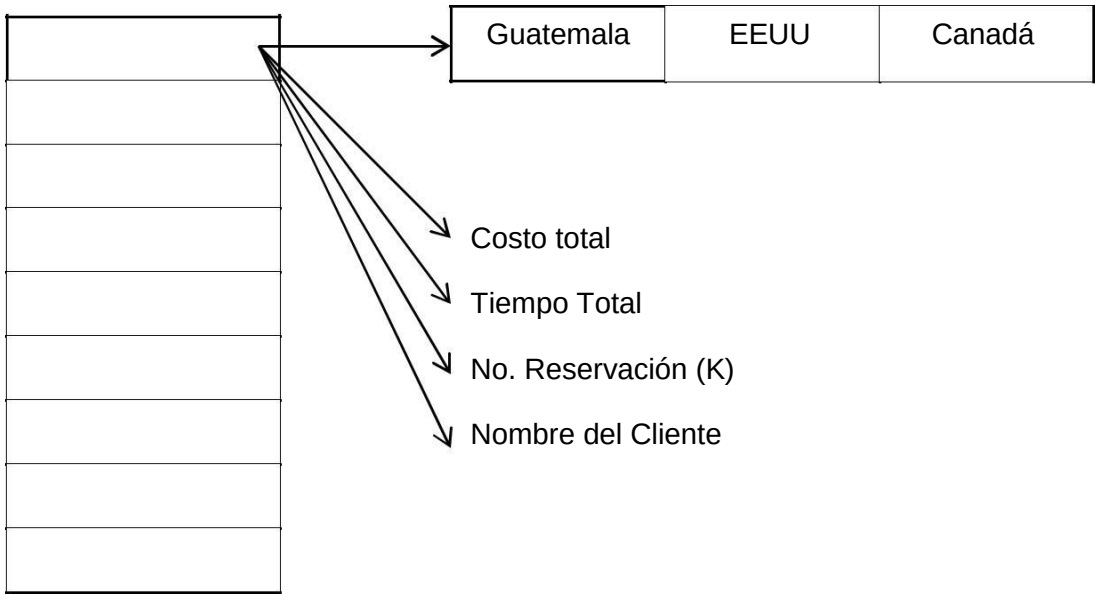
K = llave

T = Tamaño de la tabla

Y resolución de colisiones por **direccionamiento abierto** lineal por 2

i = iteración

El tamaño inicial será 43, en caso de desbordamiento (suponga 50% de la capacidad de la tabla) se aumentara el tamaño al siguiente número primo. Ejemplo:



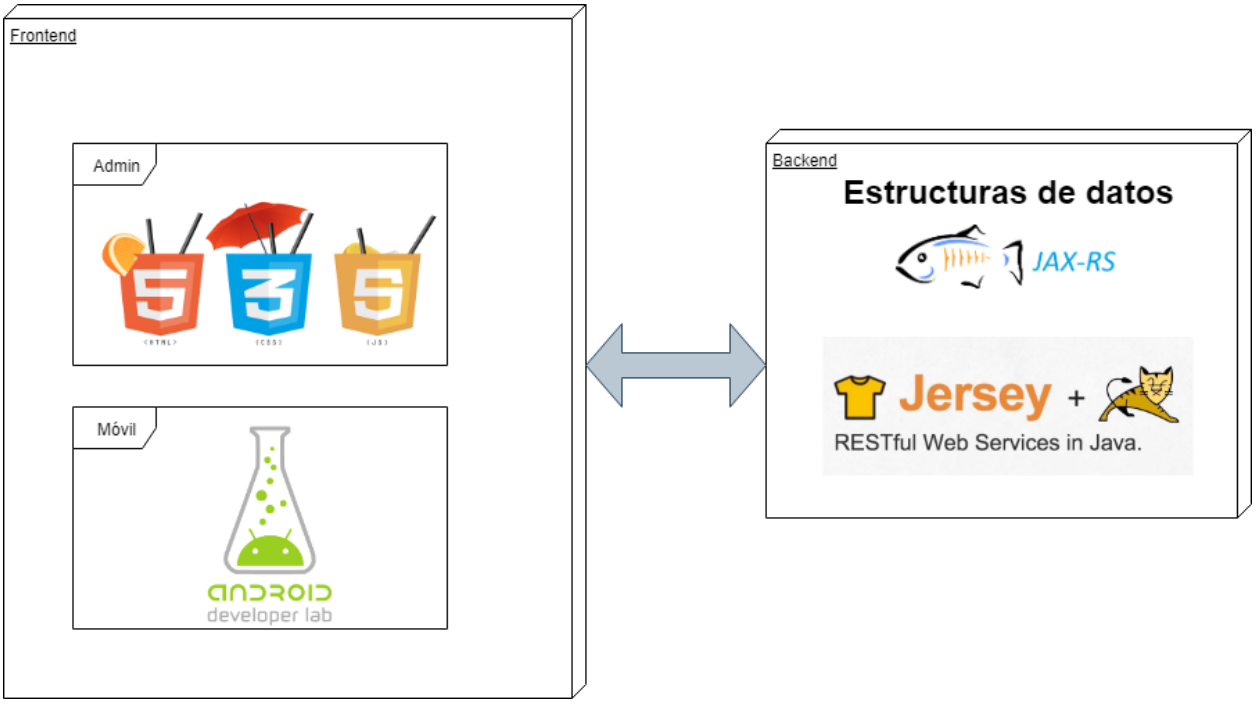
Reportes

En el módulo de administrador 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)

Estos deben mostrarse en un collage donde pueda visualizar todas las imágenes generadas.

Vista de herramientas y tecnologías



Consideraciones

- La App será realizada para la plataforma Android (nativo), se prohíbe el uso de algún framework o librería para crear aplicación en android como Xamarin, ionic, react native, etc.
- Utilizar el template Matrix para la parte del frontend para el módulo del administrador.
- Los reportes y graficas serán realizadas con Graphviz desde un servidor utilizando javax rs / jersey.
- Lenguaje: Java, kotlin.
- IDE: Eclipse, Android Studio
- No se pueden usar librerías o interfaces para las estructuras.
- Se prohíbe el uso de vectores o matrices en lugar de estructuras (exceptuando tabla hash).
- Sólo se puede calificar desde el frontend para el módulo administrador y la reservación desde la aplicación de android.
- **Proyectos que no muestren todas las gráficas de las estructuras en la aplicación del frontend no serán calificados**.
- Toda comunicación entre los clientes y el backend deben de utilizar javax rs / jersey opcionalmente con los servidores de aplicaciones apache tomcat, glassfhis, JBoss, WildFly, Payara, etc.
- Pueden trabajar todo, tanto backend y frontend en su sistema operativo físico, para el caso de no contar con un dispositivo con android pueden emular.
- Copias tendrán automáticamente nota de 0.
- El proyecto debe ser enviado de la siguiente forma:
  - Se entrega una carpeta comprimida en formato .zip de nombre [EDD]Proyecto2<carnet> con lo siguiente:
    - Código fuente del proyecto frontend y backend (Requisito para tener derecho a calificación)
    - Código fuente de App
- Fecha de entrega: 11 de enero del 2019 a las 7:00 am.
- Calificación: 11 de enero del 2019, pendiente la hora de inicio.

Referencias:
https://jersey.github.io/
https://tomcat.apache.org/download-80.cgi
https://www.mkyong.com/maven/how-to-create-a-java-project-with-maven/
http://www.vogella.com/tutorials/REST/article.html
https://www.youtube.com/watch?v=fa1AO1cp9NM