

Cantidad de integrantes: 1.

Nombre: Genaro Facundo.

Email: [FacundoGenaro@hotmail.com](mailto:FacundoGenaro@hotmail.com)

Ayudante asignado:

---

### **Introducción:**

El problema plantea el armado de un grafo de aeropuertos con sus rutas en base a los datos dados en los archivos CSV provistos por la catedra.

El objetivo, además del armado y la implementación del grafo es lograr tres servicios funcionales, los cuales eran los siguientes:

### **Servicio 1**

Verificar si existe un vuelo directo (es decir, sin escalas) entre un aeropuerto de origen y uno de destino, para una aerolínea particular. De existir, se desea conocer los kilómetros que requiere el viaje y la cantidad de asientos que se encuentran disponibles (es decir, no están reservados).

### **Servicio 2**

Para un par de aeropuertos de origen y destino, obtener todos los vuelos disponibles (directos o con escalas) que se pueden tomar sin utilizar una aerolínea determinada. Para cada vuelo indicar la aerolínea que se puede tomar, el número de escalas a realizar y la cantidad total de kilómetros a recorrer.

### **Servicio 3**

Obtener todos los vuelos directos disponibles desde un país a otro, es decir, donde no se encuentren reservados todos los asientos. Para cada vuelo se deberá indicar los aeropuertos de origen y de destino, las aerolíneas con pasajes disponibles y la distancia en kilómetros.

### **Modelado del problema:**

Al ver los archivos CSV noté que los datos tenían relación, en el archivo Reservas.csv estaban las aerolíneas de origen y destino que tenían asientos reservados, en el archivo Aeropuertos.csv estaban los aeropuertos con sus respectivos países y ciudades, y en el archivo Rutas.csv estaban los viajes que se podían realizar, en dicho archivo se podían apreciar aeropuertos de origen y destino, también decía si el vuelo entre ambos era de cabotaje, la distancia entre dichos aeropuertos y las aerolíneas que hacían el ya mencionado recorrido junto con sus asientos disponibles (sin reservar).

Todos estos datos se relacionaban, lo cual me hizo inclinarme por armar las clases de a partes con los datos que pudiera obtener en cada uno de los archivos y poniéndolos en sus respectivos lugares.

Hecho esto, a la hora de empezar a plantear la solución al armado del grafo, la mejor idea que tuve para resolverlo fue la implementación de un grafo doblemente vinculado ya que los viajes entre distintos aeropuertos deberían poder ser de ida o vuelta sin complicaciones.

Para guardar los datos de "Aeropuertos.csv", decidí implementar una clase Grafo donde los aeropuertos contenidos en dicho archivo serían las aristas que estarían guardadas en una lista de aeropuertos.

Para los datos contenidos en "Rutas.csv" fue un poco mas complicado debido a que los datos debían ir dispersos en varias clases, para hacer esto tuve que recorrer uno por uno los aeropuertos contenidos en la lista de la clase Grafo y ver si eran aeropuertos de origen o de destino para luego guardar todos los datos en una clase Rutas que contiene los datos mencionados anteriormente junto con un mapa de aerolíneas donde se guardan las aerolíneas que realizan el ya mencionado recorrido con sus respectivos asientos disponibles.

Para los datos contenidos en "Reservas.csv" solo tuve que verificar si el origen estaba almacenado en mi lista de aeropuertos para luego recorrer la lista de rutas que dicho aeropuerto poseía hasta encontrar una que tuviera el destino que especificaba el archivo, una vez hecho esto solo tuve que buscar la aerolínea en cuestión que estaba almacenada en un mapa dentro de la ruta y cargarle los asientos reservados que esta poseía.

Para resumir, la mayoría de los datos están almacenados en listas y los aeropuertos están almacenados en un mapa dentro de las rutas.

Estructura	Ventajas	Desventajas
List	$O(1)$ en <code>add()</code> $O(1)$ en <code>get()</code>	$O(n)$ en <code>contains()</code> Ocupa memoria aunque esté vacío. $O(n)$ si debo recorrerlo para ver todos los datos (con un <code>for</code> por ejemplo).
Map	$O(1)$ en <code>put()</code> $O(1)$ en <code>containsKey()</code> $O(1)$ en <code>get()</code>	Debo verificar en todo el <code>keyset</code> si es requerido ( $O(n)$ )

### Implementación de los servicios:

#### Servicio 1:

Para este servicio implementé un método boolean que, dado el caso de retornar verdadero, imprime por pantalla los datos solicitados por la consigna, caso contrario da aviso de que no existen vuelos directos o con esa aerolínea.

La creación de este método se hizo recorriendo la lista de vértices contenida en el grafo y preguntando en cada uno si el nombre del vértice era igual al nombre del aeropuerto de destino ingresado por el usuario, de ser se toman las rutas que están en el aeropuerto previamente elegido y se recorren buscando una donde el valor destino sea igual al ingresado por el usuario, si se encuentra ese valor, verifico si el mapa de aerolíneas del aeropuerto posee la aerolínea especificada por el usuario.

Este método tiene una complejidad **O(n)** debido a que en caso de no encontrar el origen y destino especificados recorrerá las dos listas de principio a fin

#### Ejemplo de uso

```
1. Listar todos los aeropuertos
2. Listar todas las reservas realizadas
3. Servicio 1: Verificar vuelos directos
4. Servicio 2: Obtener todos los vuelos sin especificar aerolinea
5. Servicio 3: Vuelos disponibles de un pais a otro
Ingrese la opcion deseada
3
Ingrese su aeropuerto de origen
Pucon
Ingrese su aeropuerto de destino
Humberto Delgado
Ingrese la aerolinea en la que desea viajar
LATAM
Existe un vuelo directo, la distancia es: 10787.62 y hay 120 asientos disponibles
```

#### Ejemplo de uso sin que haya vuelos directos

```
1. Listar todos los aeropuertos
2. Listar todas las reservas realizadas
3. Servicio 1: Verificar vuelos directos
4. Servicio 2: Obtener todos los vuelos sin especificar aerolinea
5. Servicio 3: Vuelos disponibles de un pais a otro
Ingrese la opcion deseada
3
Ingrese su aeropuerto de origen
John F. Kennedy
Ingrese su aeropuerto de destino
Pucon
Ingrese la aerolinea en la que desea viajar
LATAM
No existe un vuelo directo o la aerolinea no está disponible
```

## Servicio 2:

Para este servicio implementé un método void que dados un aeropuerto de origen y uno de destino realiza una búsqueda en profundidad para encontrar todas las rutas posibles entre dichos aeropuertos. Para lograr esto tuve que crear un mapa de aeropuertos para saber cuáles fueron recorridos e implementar dos métodos extra para hacer una búsqueda recursiva en todo el grafo, para empezar en el método inicial, recorro la lista de vértices hasta encontrar uno cuyo dato origen sea igual al ingresado por el usuario, hecho esto creo una lista de lista de rutas y la cargo recorriendo recursivamente todos los aeropuertos que no estén contenidos en el mapa de aeropuertos previamente creado y si encuentro una ruta cuyo destino sea igual al que estoy buscando la agrego a una lista de rutas y esta lista será agregada a la lista de listas de rutas.

Una vez obtenidos todos los caminos posibles desde el origen al destino, muestro por consola los datos solicitados por la consigna.

Ejemplo de uso (Adjunto una captura de una parte de la respuesta de la consola debido a que ocuparía mucho espacio en el documento)

```
Ingrese su aeropuerto de origen
John F. Kennedy
Ingrese su aeropuerto de destino
Ministro Pistarini
Camino nº 1 para ir desde John F. Kennedy hasta Ministro Pistarini
.....-
Escala nº0 en John F. Kennedy que parte hacia Ministro Pistarini
Las aerolineas disponibles son las siguientes
[United Airlines 20, Avianca 0, Aerolineas 0, Delta 152, LATAM 0]

La distancia para esta ruta fue: 8535.74 Kilometros y hubo 0 escalas
.....-
Camino nº 2 para ir desde John F. Kennedy hasta Ministro Pistarini
.....-
Escala nº0 en John F. Kennedy que parte hacia Jorge Newbery
Las aerolineas disponibles son las siguientes
[United Airlines 0, Aerolineas 0, Delta 0, LATAM 0]

Escala nº1 en Jorge Newbery que parte hacia Francisco Gabrielli
Las aerolineas disponibles son las siguientes
[Aerolineas 0]

Escala nº2 en Francisco Gabrielli que parte hacia Comandante Espora
Las aerolineas disponibles son las siguientes
[Aerolineas 0]

Escala nº3 en Comandante Espora que parte hacia Comodoro Benitez
Las aerolineas disponibles son las siguientes
[Aerolineas 0]

Escala nº4 en Comodoro Benitez que parte hacia Ministro Pistarini
Las aerolineas disponibles son las siguientes
[United Airlines 0, Delta 0]

La distancia para esta ruta fue: 12120.171 Kilometros y hubo 4 escalas
```

### Servicio 3:

Para este servicio implementé un método void que tras recibir un país de origen y uno de destino, recorre los vértices hasta encontrar un aeropuerto cuyo país sea igual al de origen, entonces busca en las rutas de dicho aeropuerto uno donde el aeropuerto destino sea del país que estamos buscando, terminado esto, el método busca en el mapa de aerolíneas de la ruta seleccionada por aerolíneas que tengan asientos disponibles hacia ese destino y muestra los datos solicitados por consola.

Este método tiene una complejidad **O(n)** debido a que en caso de no encontrar los datos solicitados recorrerá las listas y el keyset del mapa de aerolíneas de principio a fin

### Ejemplo de uso

```
1. Listar todos los aeropuertos
2. Listar todas las reservas realizadas
3. Servicio 1: Verificar vuelos directos
4. Servicio 2: Obtener todos los vuelos sin especificar aerolínea
5. Servicio 3: Vuelos disponibles de un país a otro
Ingrese la opción deseada
5
Ingrese su país de origen
ARG
Ingrese su país de destino
USA
Vuelo disponible en la aerolínea United Airlines donde el origen es: Jorge Newbery-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea Aerolíneas donde el origen es: Jorge Newbery-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea Delta donde el origen es: Jorge Newbery-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea LATAM donde el origen es: Jorge Newbery-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea United Airlines donde el origen es: Ministro Pistarini-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea Avianca donde el origen es: Ministro Pistarini-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea Aerolíneas donde el origen es: Ministro Pistarini-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea Delta donde el origen es: Ministro Pistarini-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea LATAM donde el origen es: Ministro Pistarini-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea United Airlines donde el origen es: Armando Tola-ARG y el destino es: John F. Kennedy-USA
Vuelo disponible en la aerolínea United Airlines donde el origen es: Armando Tola-ARG y el destino es: Logan-USA
```

Cabe aclarar que para el correcto funcionamiento de la aplicación se deberán ingresar los nombres de países, aeropuertos, aerolíneas y cualquier otro dato solicitado por consola tal cual esta escrito en los archivos CSV provistos por la catedra.

### Conclusiones:

El grafo doblemente vinculado fue difícil de armar debido a como estaban distribuidos los datos en los archivos csv, debo decir que me perdí más de una vez intentando hacerle seguimiento a medida que lo creaba, pero luego resultó ser bastante útil tener todos los datos distribuidos de la manera en que lo hice.

Por otro lado, creo que las complejidades temporales de toda la implementación en general se podrían mejorar, pero me faltaron oportunidades para hacer consultas por no poder ir a cursar.

La implementación de los servicios 1 y 3 fue simple, no tengo comentarios al respecto, pero el servicio 2 me pareció muy difícil de resolver, creo que se podría resolver de una manera mas optima, pero luego de pensarlo durante un largo rato no se me ocurrió otra manera que utilizar ese recorrido recursivo con listas de listas de rutas