

Universidad del Norte  
Departamento de Ingeniería de Sistemas y Computación  
Estructura de Datos II  
Laboratorio 2  
Prof. Sebastian Racedo

**Fecha de entrega límite: 07 de Abril, 11:59 pm**

**Opción 1**

Se requiere crear un software que permita a un usuario saber las distintas rutas de vuelos a las capitales del mundo (como un reto puede intentar trabajar con todas las capitales, pero también es posible acotar el problema a un continente específico, América debe ser considerado como un solo continente). Para esto se le ha solicitado que desarrolle un sistema que permita dada una distribución de vuelos (al menos lugar de inicio y de destino), y la distancia que hay entre ellos encontrar el camino de menor costo para que el usuario pueda llegar:

- A todos los lugares desde un punto inicial.
- A un destino en específico.
- Mirar los requerimientos mínimos.

**Opción 2**

Se requiere crear un software que permita a un usuario saber las distintas formas con las que dos artistas podrían llegar a colaborar en un track. Para esto se le ha solicitado que desarrolle un sistema que dada una distribución de artistas y los géneros en los que estos artistas han hecho canciones poder:

- Dado un artista inicial, cual es la probabilidad de que pueda colaborar con otro artista.
- Dado un artista inicial, cuáles son los artistas que están en una cercanía N de él.
- Mirar los requerimientos mínimos.

**Opción 3**

Se requiere crear un software que permita a un usuario saber más información sobre un catálogo de películas. Para esto se usara la información proporcionada por IMDB (pueden usar el dataset top 250 películas o la api y hacer un llamado a 250 películas). Se le ha solicitado que desarrolle un sistema que permita, dada una distribución de directores y los géneros donde estos han realizado películas poder:

- Dado un director inicial, cuáles son los géneros que se encuentran a N niveles de él.
- Dado un director Inicial y uno final, que tan probable es que ambos puedan colaborar en una película dado un género específico (debe documentar que métrica uso o desarrollo para poder dar respuesta a este inciso)
- Mirar los requerimientos mínimos.

1. Temática

Programación en Java o Python mediante el uso de estructura de datos dinámica grafos y objetos.

## 2. Requerimientos mínimos

- a. Desarrollar un programa en Java o Python que permita construir manualmente un grafo, que utilice un mapa como imagen de fondo (la imagen únicamente enriquece la interfaz visual del programa, pero no influye en la solución del problema).
- b. A medida que el usuario selecciona los puntos en la pantalla donde desea agregar un edificio/lugar, el programa debe preguntar el nombre de este. (la forma en cómo ingrese los datos puede ser diferente a la aquí propuesta con tal de que siga cumpliendo el requerimiento).
- c. Una vez ingresados todos los puntos de interés, el programa debe permitir ingresar la adyacencia que hay entre cada par de lugares.
- d. El programa debe permitir eliminar un edificio/lugar seleccionado (como se seleccione queda a la imaginación de ustedes). Si se elimina un edificio/lugar, se deben eliminar las aristas (entrada y salida) asociadas a este.
- e. El programa no debe permitir agregar un lugar encima de otro (manejo de colisiones).
- f. Al finalizar de construir el grafo, el programa debe mostrar un botón para ejecutar los recorridos BFS y DFS. Cada recorrido debe recibir como parámetro de entrada inicial.

A tener en cuenta:

Los resultados pueden ser mostrados en un textarea o de cualquier otra forma siempre y cuando sea usando la interfaz gráfica (no terminal).

Bono:

- a. Agregar interactividad al momento de ejecutar los algoritmos.
- b. Investigar y explicar que tan óptima/eficiente es su implementación y como esta se compara a otras implementaciones.

## 3. Requerimientos de código/funcionales

- a. Debe validar todas las entradas de ser necesarias.
- b. Debe cumplir con todos los requerimientos.
- c. El programa no debe cerrarse inesperadamente.

## 4. Reglas:

- a. Lenguaje de programación: Se debe realizar en Java o Python. Para la interfaz gráfica (en Java) puede usar la clase Graphics, Java-Processing, JavaFx, o cualquier otra que Java maneja. (Si usan una librería que no está en el paquete por defecto de Java deben anexar momento de entregar el trabajo final). Si utiliza Python puede usar Dash/Django (web) o alguna librería que le ayude a crear la interfaz gráfica. También puede combinar distintos lenguajes de programación para hacer el desarrollo del lab (esto es porque me he dado cuenta varios han implementado algunos ejercicios en otros lenguajes distintos a los usados en clase).
- b. Archivos de interés: Se debe entregar el diagrama de clases utilizado.
- c. Equipos: Mínimo 3, Máximo 3.

5. Nombre del comprimido a entregar

a. Lab2\_nombre1\_apellido1\_nombre2\_apellido2\_nombre3\_apellido3.ZIP

**Citar cualquier cosa que no sea de su autoría según las reglas IEEE.**

**MAY THE FORCE BE WITH YOU!**