

Investigación de Operaciones

II Semestre de 2020

Proyecto 01: Rutas Óptimas (Algoritmo de Floyd)

Estudiantes:

- Michelle Alvarado Zúñiga – 2017163102
- Luis Molina Juárez - 2015047814

 **Dynamic
Programming**

Tabla de Contenidos

1.	Descripción del problema.....	3
1.1	Descripción general	3
1.2	Características del programa.....	3
2.	Diseño del programa	4
2.1	Diseño del archivo de carga.....	4
2.2	Diseño de interfaz gráfica	5
3.	Componentes externos	5
4.	Manual de Usuario	5
4.1	Compilación y ejecución	5
4.2	Uso.....	6
5.	Objetivos alcanzados	7
6.	Objetivos no alcanzados	7
7.	Link del repositorio	7

1. Descripción del problema

1.1 Descripción general

Este proyecto es una nueva funcionalidad del Proyecto 01 donde la opción del algoritmo de las rutas más cortas, o Floyd, estará disponible en el menú principal. Este programa podrá resolver el algoritmo de Floyd para un grafo con ponderaciones en los arcos.

1.2 Características del programa

El programa deberá permitir que se indique la cantidad de nodos del grafo (de 1 a 10) y las distancias directas entre ellos. Se debe manejar un mecanismo en que las distancias sean infinitas. Se requiere que *infinito* sea el valor por defecto de todas las entradas de la tabla exceptuando la diagonal, la cual es 0. Cada nodo tendrá un nombre editable, por defecto se identificarán como A, B, C, etc.

Esencialmente, el usuario deberá ingresar los datos de la tabla D (0) y luego solicitará el cálculo de las demás tablas. Se requiere que en cada iteración se muestre la tabla P de rutas. Luego de mostrar la última tabla, se debe mostrar un diálogo que permita consultar la ruta óptima entre cualquier par de nodos.

2. Diseño del programa

2.1 Diseño del archivo de carga

El equipo de trabajo decide que usará archivos *JSON* para guardar y cargar los datos del usuario. La razón es que *typescript*, el lenguaje detrás del framework de Angular, es muy flexible con este tipo de archivos y tiene funciones muy útiles para leer y crear en este formato. Más específicamente se decide que en el archivo se guardará:

- La lista de nodos con su respectivo nombre.
- La tabla D (0). Importante recalcar que cada celda de la tabla D (0) se debe indicar con el formato `{“value”: [value]}`, donde `[value]` es el valor de la celda.

A continuación, se muestra un ejemplo de la estructura del archivo:

```
{
  "nodes": [
    {
      "name": "A"
    },
    {
      "name": "B"
    },
    {
      "name": "C"
    }
  ],
  "table": [
    [{"value": null}, {"value": null}, {"value": null}],
    [{"value": null}, {"value": null}, {"value": null}],
    [{"value": null}, {"value": null}, {"value": null}]
  ]
}
```

Ilustración 1 Formato del archivo JSON

2.2 Diseño de interfaz gráfica

Para la interfaz gráfica que se mostrará para la ejecución del algoritmo se decide dividir la pantalla en dos secciones, la sección izquierda permitirá configurar los parámetros de este, es decir, la cantidad de nodos y la posibilidad de cambiar su nombre. La sección derecha permitirá visualizar la tabla donde se visualiza la ejecución del algoritmo. Todas las decisiones de diseño del Proyecto 00 se toman en cuenta en este programa.

3. Componentes externos

Para la creación de algunos componentes del programa se han usado componentes externos a nuestro desarrollo. Dentro de estos están incluidos:

- **plus minus number picker:** este componente se tomó como base y fue modificado para seleccionar la capacidad de la mochila y la cantidad de objetos. Creado por Toby, un usuario de CodePen que puso a disponibilidad el componente en: <https://codepen.io/tobyj/pen/vGbpPj>
- **Bootstrap:** está librería se incluyó por su clase de estilo *modal*, principalmente para mostrar un mensaje de confirmación para cuando el usuario quiera navegar a otro algoritmo o al menú principal, en caso de que.
- **Flaticon:** se usaron algunos íconos proporcionados por Flaticon, los créditos se encuentran en la página principal del programa.

4. Manual de Usuario

4.1 Compilación y ejecución

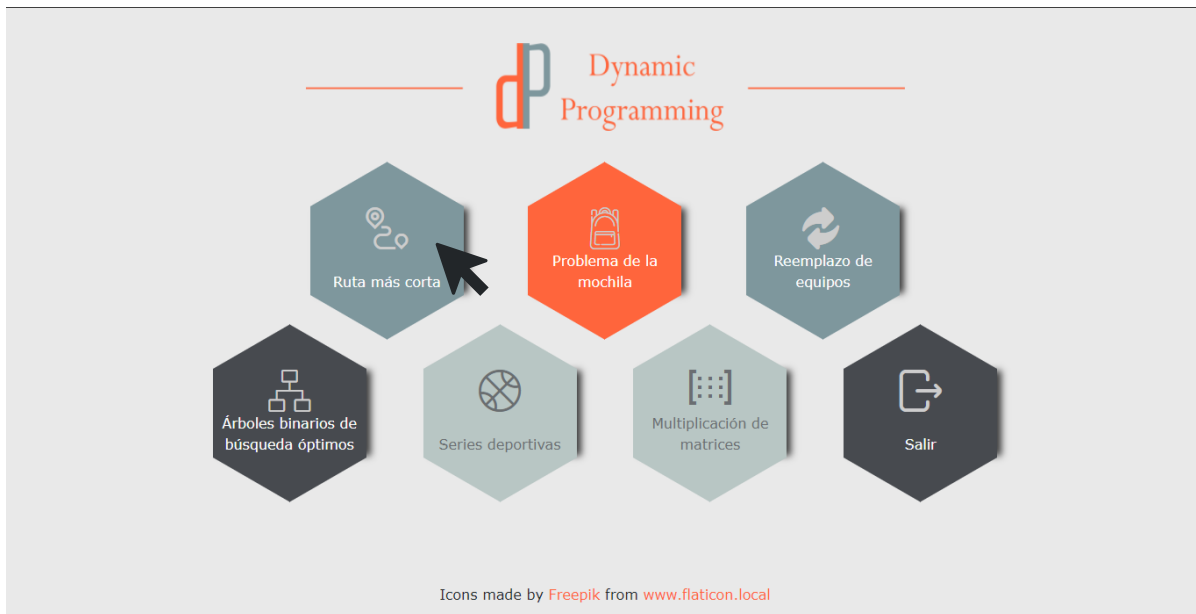
Estas instrucciones de compilación y ejecución son las mismas que las del Proyecto 00. Al ser un proyecto en angular se necesita realizar una serie de pasos para poder compilar y ejecutar el programa. Se debe:

1. Abrir la carpeta que contiene el proyecto de Angular, en este caso se llama IC6400-PRY y abrir un comando de línea de Windows en este directorio.

2. Ejecutar el comando `npm install`. Este comando instalará todo lo necesario para correr el proyecto. Como requisito, se debe tener instalado `node.js` (por el comando `npm`) y Angular CLI.
3. Una vez terminado el proceso que ejecuta el comando, se debe ejecutar `ng serve` para poder lanzar el servidor.
4. Se debe acceder desde un navegador a la dirección: `localhost:4200`.

4.2 Uso

El uso de este programa es muy sencillo, se le mostrará el menú principal y será posible seleccionar cualquiera de las opciones. En este caso seleccionará *Rutas más cortas*.



5. Objetivos alcanzados

Se alcanzaron los siguientes objetivos:

- Se logra seleccionar la cantidad de nodos y modificar su nombre.
- Se logra modificar el valor de una arista entre dos nodos.
- Se logra la ejecución del algoritmo y mostrar las diferentes tablas $D(k)$ y $P(k)$.

6. Objetivos no alcanzados

No se alcanzaron los siguientes objetivos:

- El usuario debe borrar el infinito para agregar el valor de una arista en la tabla inicial, de lo contrario podría ocurrir un comportamiento inesperado. No se pudo validar este caso.
- No se pudo mostrar los cambios entre las diferentes tablas $D(k)$, es decir, colorear los cambios.
- Posibles objetivos no alcanzados faltantes.

7. Link del repositorio

El proyecto se almacenó en un repositorio de GitHub, disponible en el siguiente enlace:

<https://github.com/LuisMJ1197/IC6400-ProyectoSemestral>