

Investigación de Operaciones

II Semestre de 2020

Proyecto 03: Reemplazo de Equipos

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	3
2.1	Decisiones de diseño de la estructura del código	3
2.2	Diseño del archivo de carga.....	4
2.3	Diseño de interfaz gráfica	4
3.	Componentes externos	5
4.	Manual de Usuario	5
4.1	Compilación y ejecución	5
4.2	Uso.....	5
5.	Objetivos alcanzados	6
6.	Objetivos no alcanzados	6
7.	Link del repositorio	6

1. Descripción del problema

1.1 Descripción general

Este proyecto es una aplicación de interfaz gráfica creada en Angular6+ utilizando HTML5 Y CSS3, el cual corresponde a una nueva funcionalidad del Proyecto 00 donde la opción del algoritmo Problema de Reemplazo de Equipos estará disponible.

1.2 Características del programa

El programa deberá permitir resolver el problema de reemplazo de equipos, siendo la interfaz gráfica lo más flexible posible permitiéndole al usuario proporcionar los siguientes datos:

- El costo inicial del equipo.
- El plazo del proyecto (entre 1 y 30).
- La vida útil del equipo (entre 1 y 10).
- Para cada unidad de tiempo de uso del equipo el precio de reventa y el costo de mantenimiento.
- En caso de que el usuario lo desee podrá indicar la ganancia recibida por usar el equipo en cada unidad de tiempo de su vida útil.

2. Diseño del programa

2.1 Decisiones de diseño de la estructura del código

Por requisito ya establecido, la ejecución de un algoritmo se debe hacer por medio de un componente de angular. Sin embargo, para separar el control de la interfaz gráfica y el algoritmo correspondiente como tal, se decide que el componente de angular será el encargado de controlar la interfaz gráfica y todos los eventos que esta genere y, además, usará los servicios de una clase (específica por algoritmo) que le dará los resultados correspondientes del algoritmo en ejecución. Esto con el fin de separar responsabilidades y no tener código en el componente de Angular que maneje la interfaz y ejecute la lógica del algoritmo.

2.2 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:

- El plazo del proyecto.
- La vida útil del equipo.
- El costo inicial.
- Un valor booleano que indica si está activada la opción de indicar la ganancia de uso por cada unidad de tiempo de vida útil del equipo.
- Una lista de objetos por cada unidad de tiempo de vida útil del equipo que indican el año, el precio de reventa, el precio de mantenimiento y la ganancia recibida por el uso del equipo.

```
{
  "term": 5,
  "lifeSpan": 3,
  "initialCost": 500,
  "isGainActivated": false,
  "lifeSpans": [
    {
      "year": 1,
      "resale": 400,
      "maintenance": 30,
      "gain": 0
    },
    {
      "year": 2,
      "resale": 300,
      "maintenance": 40,
      "gain": 0
    },
    {
      "year": 3,
      "resale": 250,
      "maintenance": 60,
      "gain": 0
    }
  ]
}
```

2.3 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. 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**: esta 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

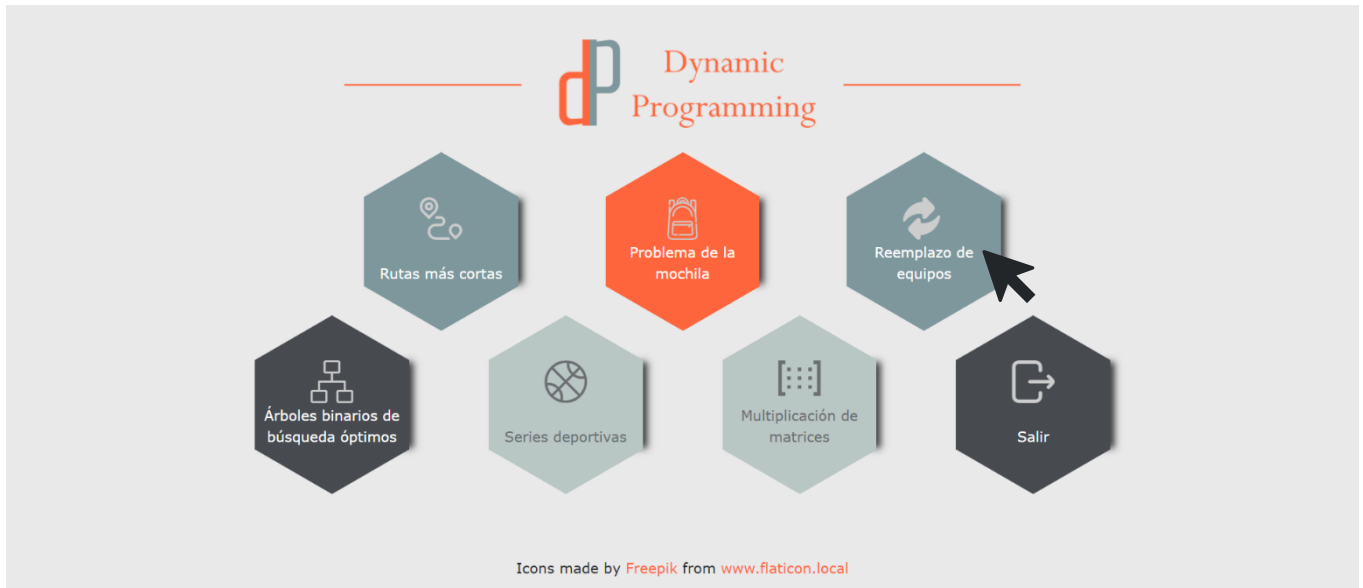
4.1 Compilación y ejecución

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 *Reemplazo de equipos*.



Se le mostrará una interfaz que le permitirá:

Reemplazo de Equipos

Cargar archivo

Guardar archivo

Costo inicial:

Plazo del proyecto:

- 5 +

Vida útil del equipo:

- 3 +

Tabla de configuración:

☐ Añadir ganancias

Año	Reventa	Mantenimiento	Ganancia
1	400	30	0
2	300	40	0
3	250	60	0

Ejecutar algoritmo

Tabla de análisis:

t	G(t)	Próximo
0	640	1,3
1	510	2,4
2	380	5
3	260	4
4	130	5
5	0	

Planes:

Plan #1

0 → 1 → 2 → 5

Plan #2

0 → 1 → 4 → 5

Plan #3

0 → 3 → 4 → 5

Opciones de guardar y cargar archivos

Visualizar los planes óptimos

Visualizar la tabla de solución.

5. Objetivos alcanzados

Se alcanzaron los siguientes objetivos:

- Se cargan y guardan archivos.
- Se logra establecer con éxito el costo inicial del equipo, además de la vida útil del equipo y el plazo del proyecto.
- Para cada unidad de tiempo de vida útil del equipo se logra indicar el precio de reventa y el costo de mantenimiento.
- Se logra permitir que el usuario indique si desea agregar la ganancia recibida por usar el equipo en cada unidad de tiempo de su vida útil.
- Se logra visualizar la tabla de análisis y los planes óptimos.

6. Objetivos no alcanzados

No se alcanzaron los siguientes objetivos:

- No se logró desarrollar el trabajo extra opcional 1.

7. Link del repositorio

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

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