TEC | Tecnológico de Costa Rica

Investigación de Operaciones

II Semestre de 2020

Proyecto 02: El Problema de la Mochila

Estudiantes:

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

Dynamic Programming

Tabla de Contenidos

1. Descripción del problema

1.1 Descripción general

Este proyecto es una nueva funcionalidad del Proyecto 01 donde la opción del algoritmo del problema de la mochila estará disponible en el menú principal.

1.2 Características del programa

El programa deberá permitir resolver el problema de la mochila en sus versiones de 0/1 Knapsack, Bounded Knapsack y Unbounded Knapsack. Deberá permitir establecer la capacidad de la mochila (de 0 a 20) y la cantidad de objetos (de 1 a 10) y para cada objeto se deberá indicar el costo, el valor y la cantidad disponible por objeto.

La salida gráfica debe ser una tabla utilizada para ilustrar la ejecución del algoritmo. Además, se debe establecer para cada celda un color rojo en caso de que este no sea “seleccionado” y en verde cuando sí lo fue. Se debe presentar la solución final en forma completa.

Además, el programa deberá permitir grabar archivos con los datos ingresados por el usuario para luego poder cargarlos y ejecutarlos sin tener que ingresar estos valores.

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 framwork 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 tipo de mochila, es decir, 0/1, Bounded, Unbounded representados por un entero: 0, 1 y 2 respectivamente.
- El tamaño de la mochila.
- La lista de objetos que el usuario ha indicado con sus respectivos atributos (id, costo, valor, cantidad).

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


```
{
    "kind": 2,
    "knapsackSize": 10,
    "items": [
        {
            "id": 0,
            "cost": "4",
            "value": "11",
            "quantity": null,
            "selected": true
        },
        {
            "id": 1,
            "cost": "3",
            "value": "7",
            "quantity": null,
            "selected": true
        },
        {
            "id": 2,
            "cost": "5",
            "value": "12",
            "quantity": null,
            "selected": false
        }
    ]
}
```


Ilustración 1Formato 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 y mostrar el problema matemático inicial y la solución correspondiente. 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

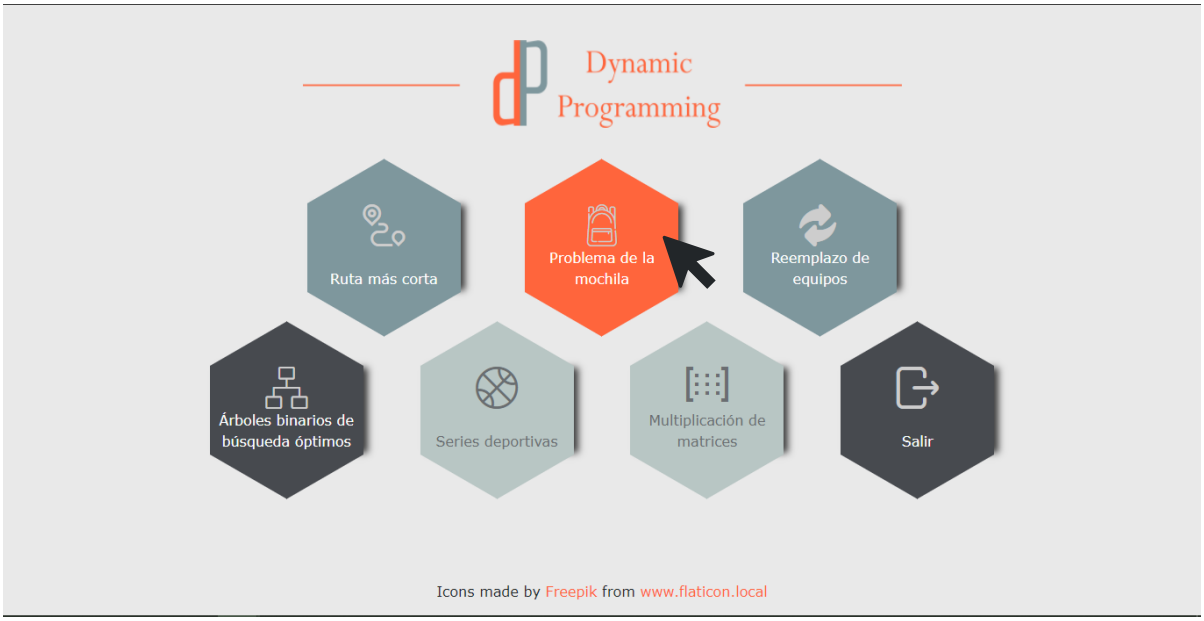
Estás instrucciones de compilación y ejecución son las mismas que las del Proyecto 00. Al ser un proyecto en angular se necesita realizar una serie de pasos para poder compilar y ejecutar el programa. Se debe:

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.

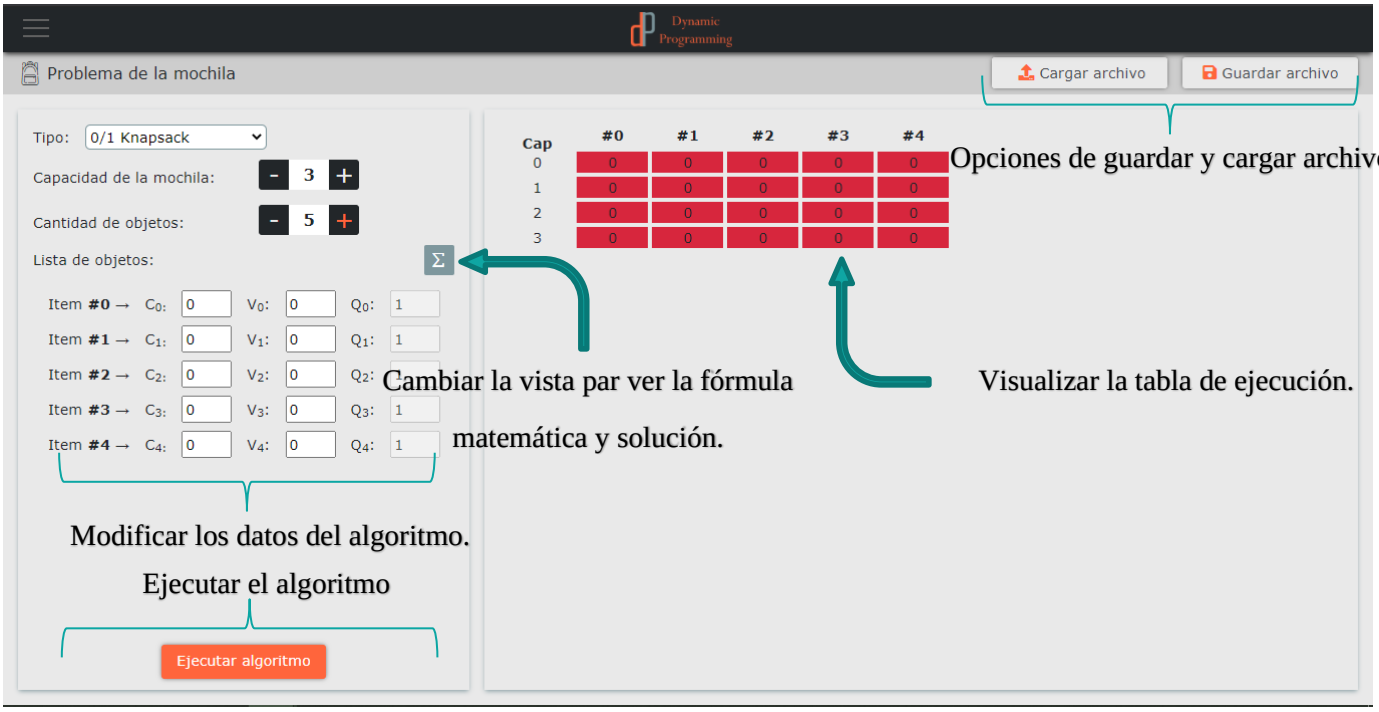
2. Ejecutar el comando *npm install*. Este comando instalará todo lo necesario para correr el proyecto. Como requisito, se debe tener instalado *node.js* (por el comando *npm*) y Angular CLI.
3. Una vez terminado el proceso que ejecuta el comando, se debe ejecutar *ng serve* para poder lanzar el servidor.
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 *Problema de la mochila*.



Se le mostrará una interfaz que le permitirá:



5. Objetivos alcanzados

Se alcanzaron los siguientes objetivos:

- Se logra seleccionar el tipo de mochila de forma correcta.
- Se logra establecer con éxito la capacidad de la mochila y la cantidad de objetos. Además, para cada objeto se crea correctamente los espacios para indicar el costo, valor y cantidad disponible.
- Se logra visualizar la tabla de ejecución correctamente con los colores correspondientes.
- Se logra visualizar la fórmula matemática inicial, es decir, el Z que se debe maximizar y la restricción.
- Se logra visualizar la solución del problema indicando el valor de cada *x*.

6. Objetivos no alcanzados

No se alcanzaron los siguientes objetivos:

- No se logra eliminar el atributo *selected* del archivo de guardado. Aún así no implica ningún mal funcionamiento en el programa.
- No se logra realizar una página web totalmente responsive.

7. Link del repositorio

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

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