TEC | Tecnológico de Costa Rica

Investigación de Operaciones

II Semestre de 2020

Proyecto 05: Árboles Binarios de Búsqueda

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 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 *Árboles binarios de búsqueda óptimos* estará disponible.

1.2 Características del programa

El programa deberá permitir resolver el problema de encontrar el árbol binario de búsqueda óptimo para un conjunto de llaves, siendo la interfaz gráfica lo más flexible posible permitiéndole al usuario proporcionar los siguientes datos:

- Número de llaves (n) donde se puede suponer que $n \leq 10$
- Para cada llave se debe indicar un texto o código por el cual se harán las búsquedas. El ingreso de estas llaves no necesariamente se hará en orden.
- Para cada llave se debe indicar un peso asociado expresado como un número real (con o sin decimales).

Existe un trabajo opcional sobre manejar una cantidad superior a 10. Además, para ejecutar el algoritmo se deben ordenar lexicográficamente las llaves, básicamente esto quiere decir en orden de diccionario. Los pesos asociados a cada llave se deben convertir en probabilidades, para una llave particular, su probabilidad se conoce al dividir su peso particular entre la suma de los pesos de todas las llaves.

Se debe mostrar las tablas A y R gráficamente, similar a las vistas en clase.

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:

- La lista de llaves con su código, peso y probabilidad. Aunque la probabilidad es irrelevante debido a que igual se calculará al ejecutar el algoritmo. Con la lista se conoce el total de llaves.

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. Para este proyecto la sección izquierda también se utilizará para visualizar las llaves ordenadas.

2.4 Algoritmos usados

El algoritmo utilizado para la realización de este proyecto corresponde al método de Programación Dinámica visto en clase para la búsqueda de árbol binario de búsqueda óptimo.

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.
- **Binary Tree MLM Html**: Para la representación del árbol binario. Este componente fue desarrollado por Avaneesh Kumar y está disponible en CodePen: https://codepen.io/Avaneesh/pen/QWwNrBX

4. Manual de Usuario

4.1 Compilación y ejecución

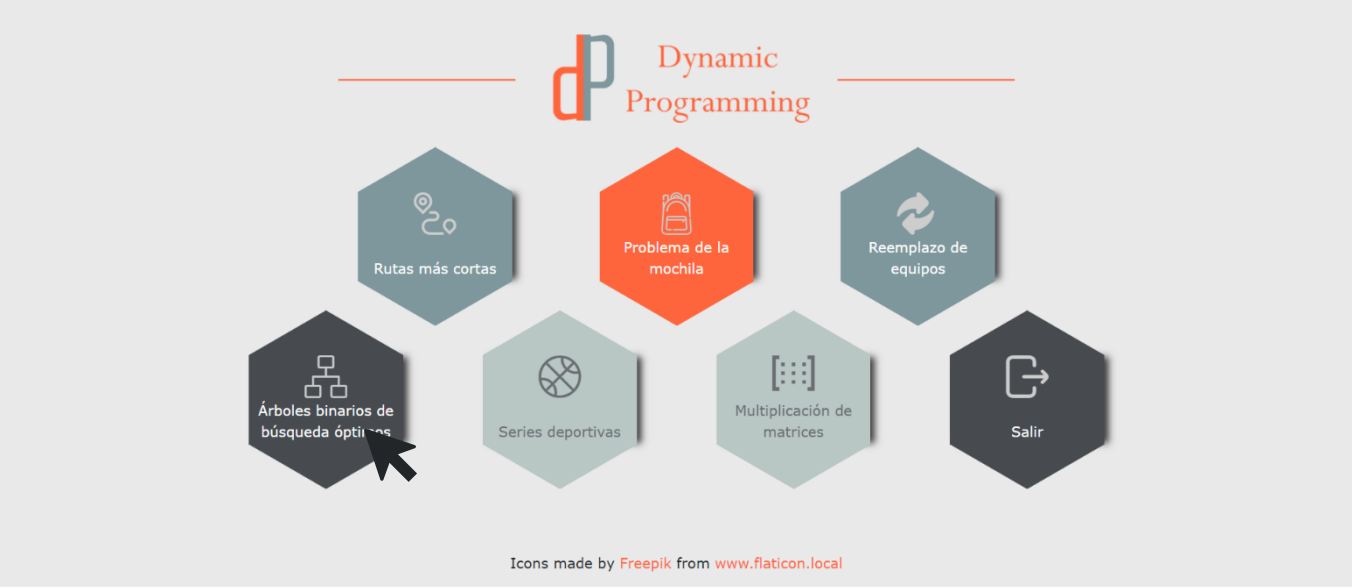
Al ser un proyecto en angular se necesita realizar una serie de pasos para poder compilar y ejecutar el programa. Se debe:

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.

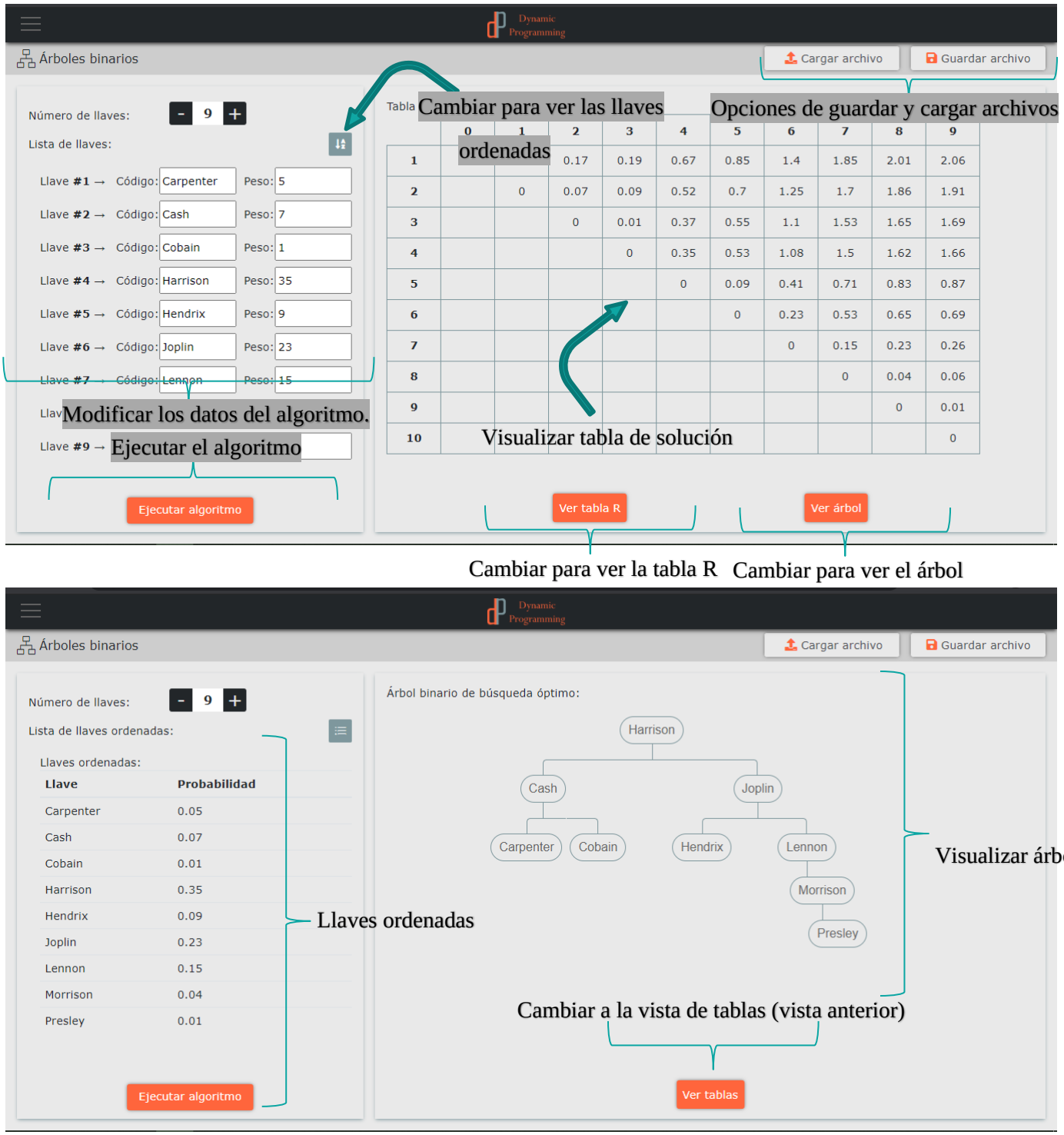
2. Ejecutar el comando *npm install*. Este comando instalará todo lo necesario para correr el proyecto. Como requisito, se debe tener instalado *node.js* (por el comando *npm*) y Angular CLI.
3. Una vez terminado el proceso que ejecuta el comando, se debe ejecutar *ng serve* para poder lanzar el servidor.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



5. Objetivos alcanzados y no alcanzados

Se alcanzaron los siguientes objetivos:

- Se cargan y guardan archivos.
- Se logra establecer el número de llaves, y por cada llave se puede indicar el código y peso.
- Se muestran las llaves ordenadas.
- Se calculan las tablas A y R y se muestran gráficamente.
- Se logra desplegar el árbol de una forma gráfica y agradable.

No se alcanzo realizar el punto de extra.

6. Link del repositorio

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

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