



Documentación proyecto 01

Investigación de operaciones gr 60

**Daniel Villatoro Cantarero(2018155978)
Daniel Calderón Díaz(2018229200)**

Profesor: Jean Carlos Miranda Fajardo

2 Semestre 2021

Manual de Usuario	3
Instrucciones de compilación	3
Instrucciones de uso	3
Descripción del problema	4
Librerías usadas	5
Análisis de resultados	5

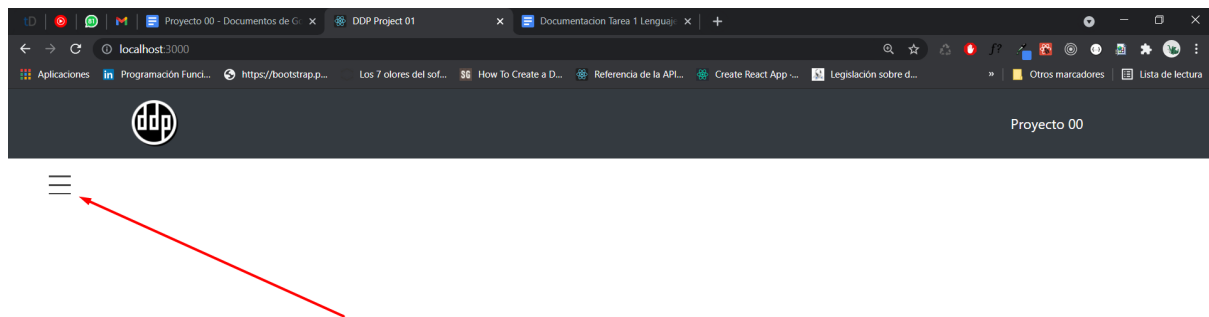
Manual de Usuario

Instrucciones de compilación

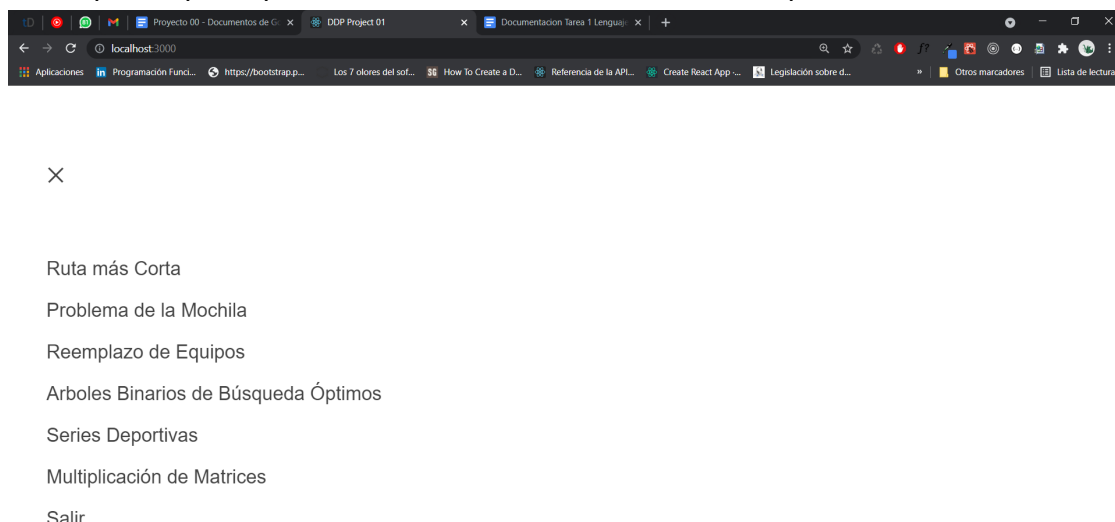
1. Debemos tener primeramente instalado en nuestra máquina Node JS
2. Debemos abrir una terminal en el directorio donde se encuentra el proyecto.
3. Debemos ejecutar el siguiente comando de compilación:
 - `npm start`
4. Posteriormente en el navegador predeterminado se nos mostrará toda la información del sistema web desarrollado en React.

Instrucciones de uso

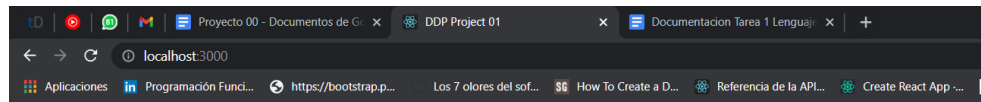
Antes de ver este apartado es necesario primeramente haber visto las instrucciones de compilación. Cuando ejecutamos `npm start` en el directorio del proyecto nos mostrara esta página:



En la parte superior a la izquierda podemos observar el icono que hace mención al menú, por lo que al presionarlo se nos mostrará diferentes opciones como:



Debemos seleccionar la opción de Ruta más corta:



Ruta más Corta

Problema de la Mochila

Reemplazo de Equipos

Arboles Binarios de Búsqueda Óptimos

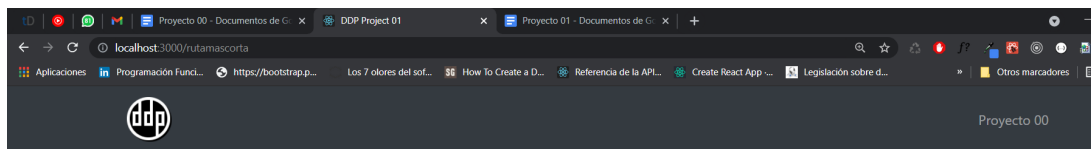
Series Deportivas

Multiplicación de Matrices

Salir

El algoritmo de la ruta más corta consiste, en una modalidad de problemas de redes, en la cual se debe determinar el plan de rutas que genere la trayectoria con la mínima distancia total, que una un nodo fuente con un nodo destino, sin importar el número de nodos que existan entre estos.

Al seleccionar dicha opción nos aparecerá los siguiente:



Rutas más cortas

Ingrese las dimensiones:

X

Generar

Guardar

Seleccionar archivo

Ningún archi... seleccionado

Matriz D(0):

Generar Resultados

Matriz Mínima:

Matriz P:

En ese mismo apartado podremos generar la matriz 0 en donde tendremos que llenar los valores de cada nodo para posteriormente mostrar la matriz mínima y matriz p.



Rutas más cortas

Ingrese las dimensiones:

X

Generar

Guardar

Seleccionar archivo

Ningún archi... seleccionado

Matriz D(0):

D(0)	A	B	C	D
A	<input type="text" value="0"/>	<input type="text" value="∞"/>	<input type="text" value="∞"/>	<input type="text" value="∞"/>
B	<input type="text" value="8"/>	<input type="text" value="0"/>	<input type="text" value="8"/>	<input type="text" value="9"/>
C	<input type="text" value="5"/>	<input type="text" value="4"/>	<input type="text" value="0"/>	<input type="text" value="3"/>
D	<input type="text" value="11"/>	<input type="text" value="∞"/>	<input type="text" value="∞"/>	<input type="text" value="0"/>

Generar Resultados

Matriz Mínima:

Matriz mínima:

Generar Resultados

Matriz Mínima:

D(0)	A	B	C	D
A	0	∞	∞	∞
B	8	0	8	9
C	5	4	0	3
D	11	∞	∞	0

Matriz P:

Matriz P:

D(0)	A	B	C	D
A	0	0	0	0
B	0	0	0	0
C	0	0	0	0
D	0	0	0	0

Descripción del problema

Descripción general:

Se debe programar el Algoritmo de Floyd para obtener las rutas óptimas entre cualquier par de nodos en un grafo ponderado con distancias. La interfaz debe ser gráfica. Toda la programación debe realizarse en Angular 6+. No se pueden cambiar las especificaciones de este documento.

Entrada y salida:

La interacción con el usuario se hará por medio de interfaces gráficas que deben ser desarrolladas con HTML5 y CSS3. Debe haber un estilo uniforme en el look & feel de todas las interfaces. El programa tendrá la posibilidad de grabar archivos con los datos particulares de cada problema ingresado por el usuario, para que puedan ser cargados de nuevo y editados si el usuario así lo desea. El formato de este archivo queda a discreción de cada grupo de trabajo. Se espera que, el día de la revisión, cada grupo cuente con bastantes archivos de pruebas para mostrar las capacidades de su proyecto.

Menú principal:

En el Proyecto 0 de este curso se desarrolló un menú principal que podía “lanzar” diversos algoritmos. El algoritmo solicitado en este proyecto será implementado como un componente Angular que será añadido al Proyecto 0 que será ejecutado desde el menú mencionado. Se debe activar una “burbujita” con una descripción general de este algoritmo cuando el cursor se pose sobre la opción respectiva en el menú principal (tooltip).

Problema de las rutas más cortas:

Usando el algoritmo de Floyd estudiado en clases, este programa resolverá el problema de encontrar las rutas más cortas entre cualquier par de nodos de un grafo con ponderaciones en los arcos.

Se espera que la interfaz gráfica sea lo más flexible posible. El usuario debe proporcionar la cantidad de nodos del grafo y las distancias directas entre ellos. Debe haber un mecanismo para indicar el caso en que las distancias son infinitas (i.e., no ruta directa) entre 2 nodos. Se sugiere que ese sea el valor por defecto para todas las entradas de la tabla al inicio (excepto la diagonal de la tabla que tendrá siempre ceros). Para este proyecto, el usuario podrá indicar entre 1 y 10 nodos. Cada nodo tendrá un nombre editable pero por defecto se identificarán como A, B, C, etc.

Esencialmente, el usuario ingresará los datos de $D(0)$. A continuación, usando algún mecanismo apropiado (e.g., un botón) el usuario solicitará el cálculo de las siguientes tablas intermedias ($D(1)$, $D(2)$, $D(3)$, etc.) hasta llegar a la tabla final. Cada tabla mostrará de una manera apropiada los cambios con respecto a la tabla anterior. También en cada iteración se mostrará el estado actual de la tabla $P(k)$ de rutas.

Después de la última tabla, se activará un diálogo que permite solicitar la ruta óptima entre cualquier par de nodos. Esta información está codificada en la tabla P

Librerías usadas

Librería	Uso
alertifyjs	Para alertar de posibles errores.

Análisis de resultados

Se implementó cada uno de los criterios necesarios para el desarrollo completo de este proyecto, excepto la visualización de las matrices intermedias y las rutas.