Arellano Granados Angel Mariano

218123444

Computación Tolerante a Fallas

D06 2023B

Generar un programa que sea capaz de restaurar el estado de ejecución.

Application checkpointing

Introducción

Comúnmente durante nuestra trayectoria como estudiantes los programas que creamos suelen perder todos los datos tras cerrar la interfaz o consola, sin embargo, en un ambiente laboral formar perder todos los datos de la ejecución de un programa representa un error diseño muy grave.

Por ello una de las estrategias que primero se nos enseña es a guardar datos en archivos para poder ser recuperados en posteriores usos del programa, los métodos en los que podemos guardar los datos de ejecución son infinitos, estos pueden ser manuales o automáticos.

Hablando de la computación tolerante a fallas los respaldos automáticos de información resultan un método esencial para evitar la perdida de datos por causas exteriores o fallos humanos.

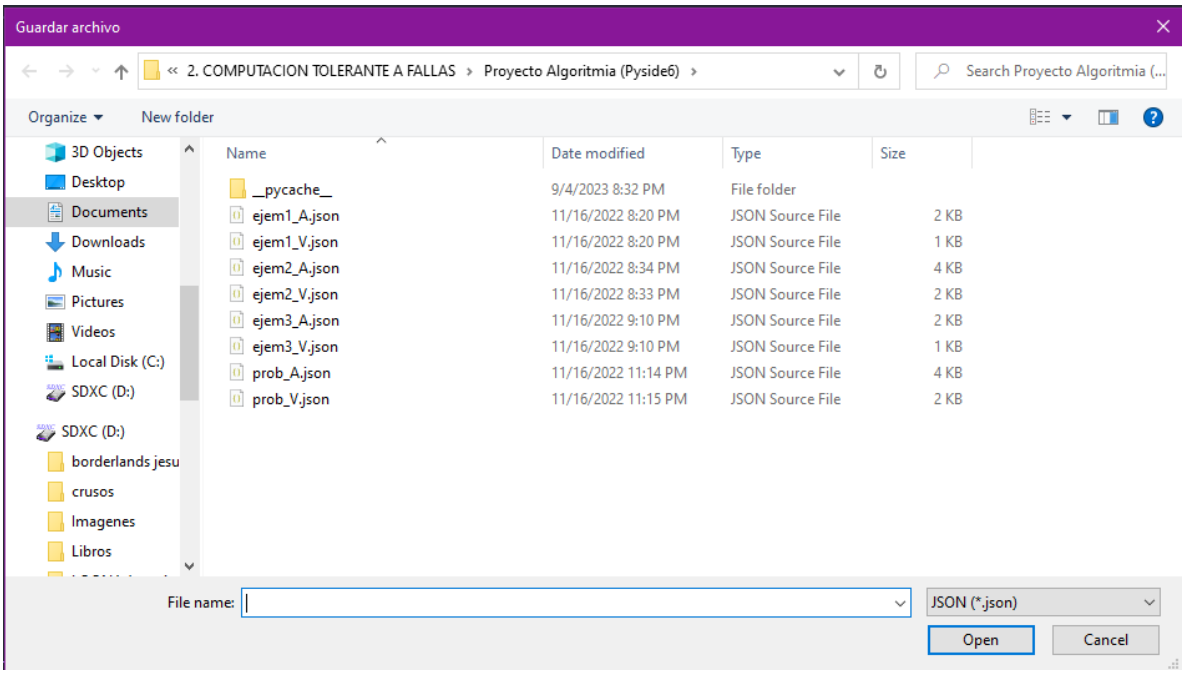
Desarrollo

Para esta actividad usare un proyecto que programa en la materia de Algoritmia, el cual es una interfaz grafica que recibe nodos y se puede crear uniones entre estos nodos con diferentes ponderaciones para crear grafos, la función principal de este programa es encontrar el camino mas corto para recorrer todos los nodos del grafo con el método de Dijkstra.

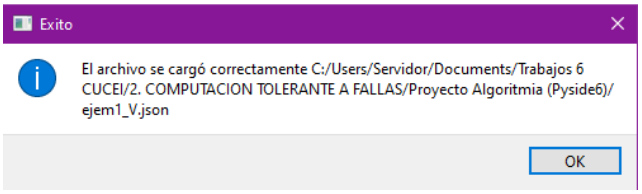
El programa con anterioridad ya era capaz de guardar y recuperar los datos de los diferentes grafos previamente creados por medio de convertir los arreglos en diccionarios y guardando estos diccionarios en archivos JSON.

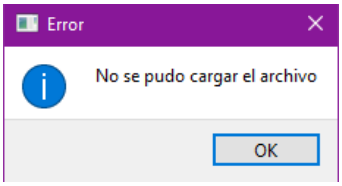
Para esta actividad se agregará la posibilidad de recuperar los nodos justo al inicio de la ejecución, esta será opcional por si el usuario desiara crear un grafo desde cero.

Al iniciar el programa este mostrara un navegador donde podremos ver los grafos que generamos previamente, podemos abrir uno o cancelar para crear uno nuevo.

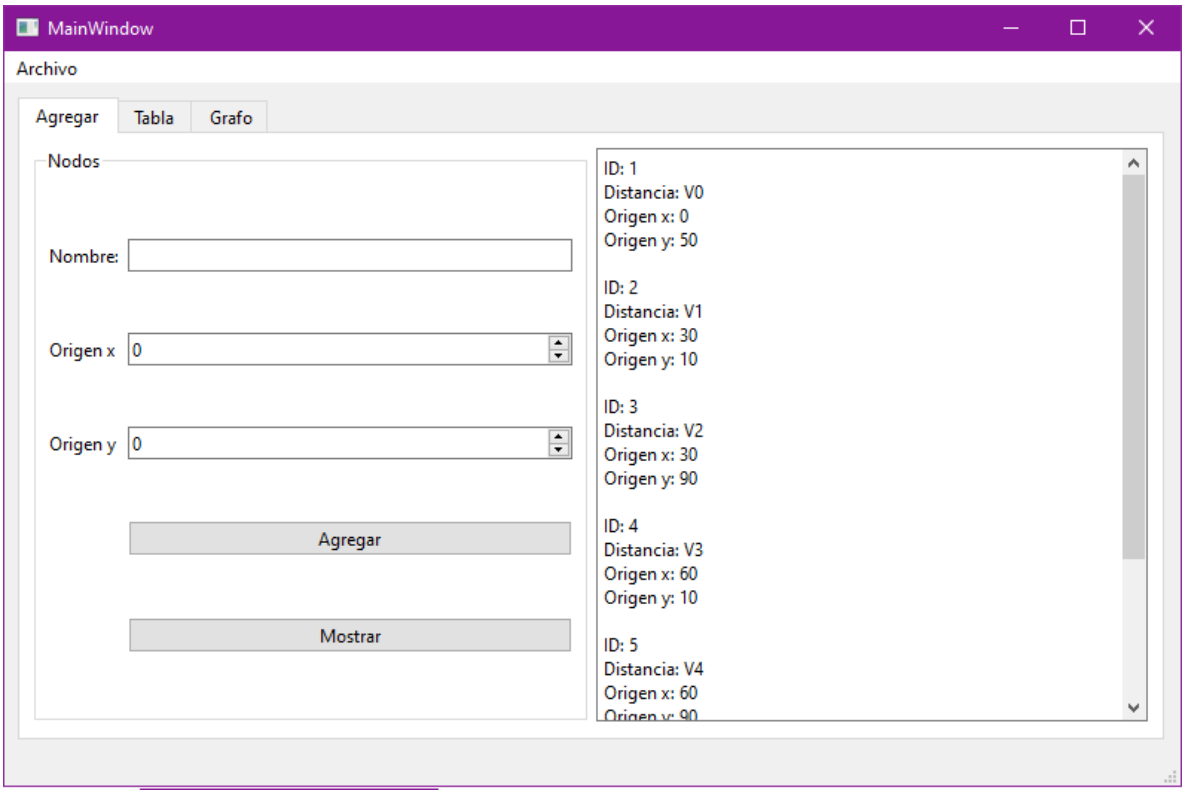


Este mostrara un mensaje de confirmación al cargar los datos o uno de error si no se abre uno.

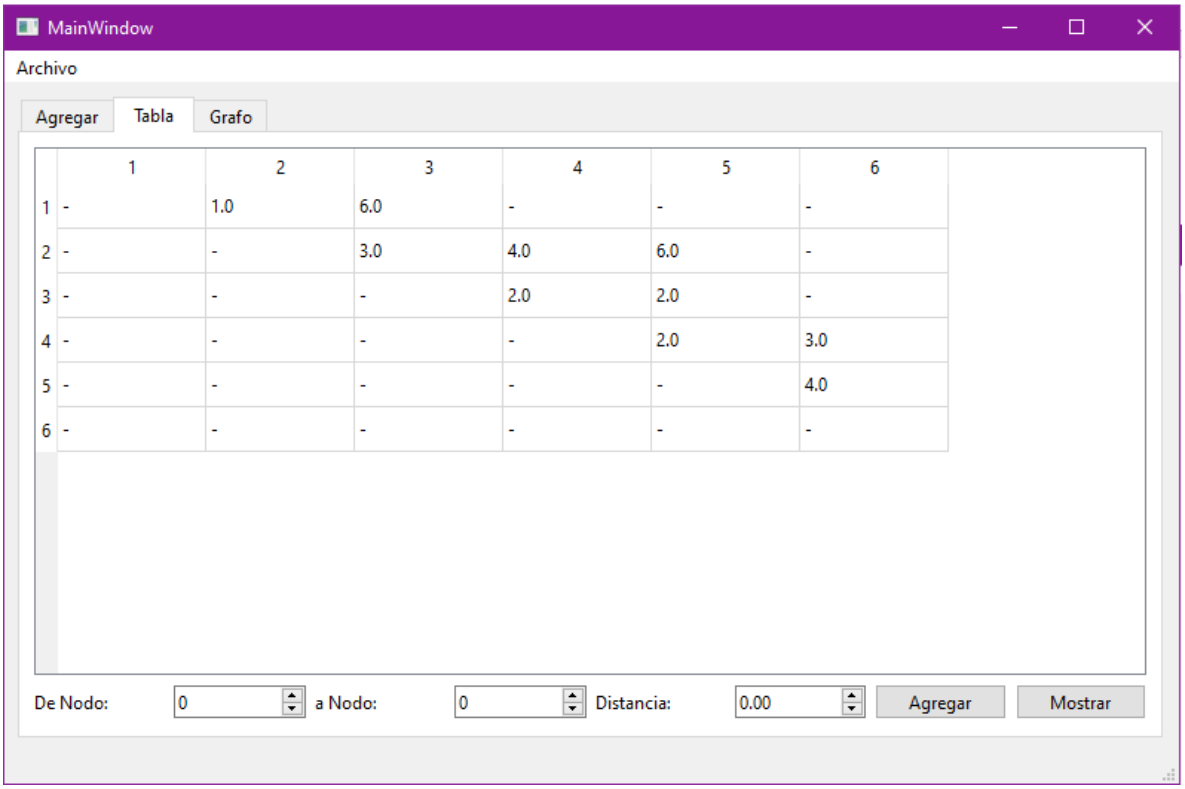




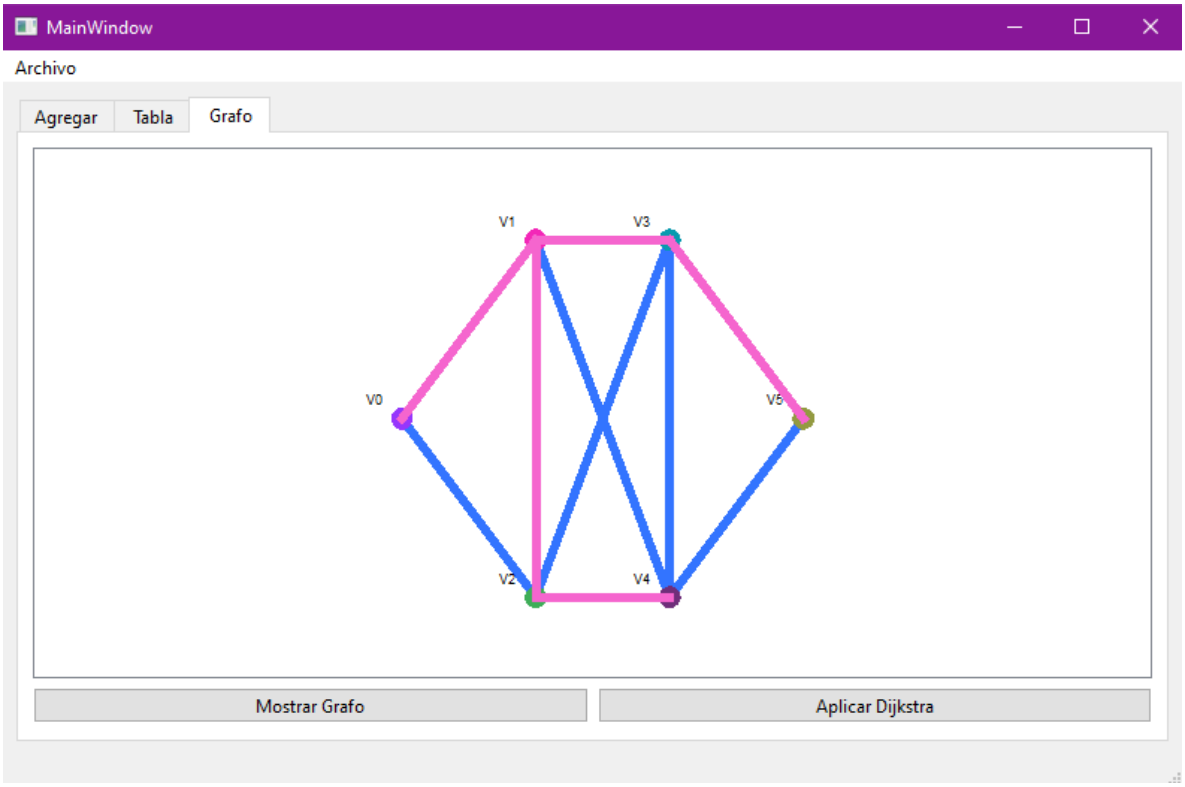
Ya en el programa este consiste de 3 pestañas, la primera sirve para agregar y visualizar nodos al grafo.



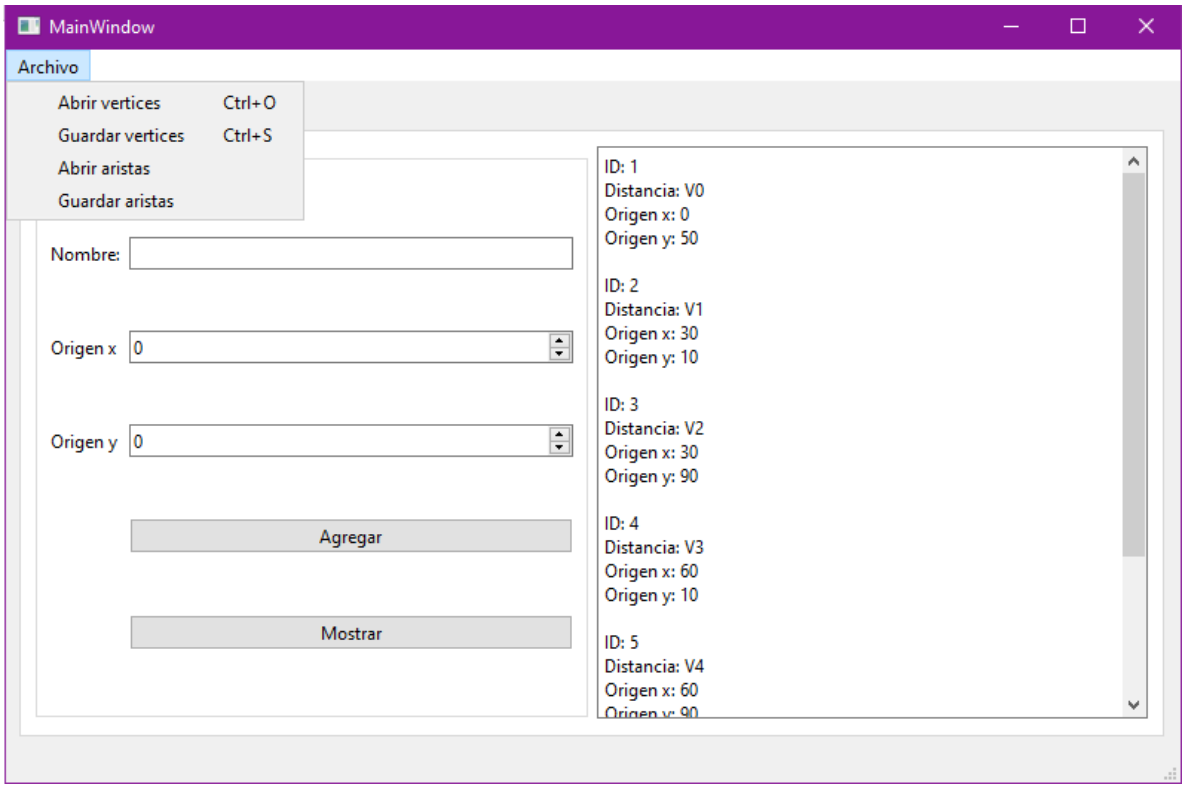
La segunda sirve para agregar aristas entre los diferentes nodos y para ver el grafo en notación de una matriz.



Y la tercera y ultima pestaña sirve para visualizar el grafo y aplicar el método de Dijkstra.



El programa puede guardar y abrir archivos JSON dentro de un menú desplegable.



Implementar un autoguardado no era viable en este proyecto pues al no poder eliminar nodos equivocarse en un dato podría causar que el grafo entero se arruinara o quedara obsoleto para el método de Dijkstra, por ello solo se guardan los datos si el usuario lo desea.

La estructura el los archivos JSON es la siguiente:

Conclusión

La recuperación de datos se ha convertido en un estándar de la industria, pues a nadie nos gustaría que por un apagón o por desconectar el ordenador por accidente se perdiera todos los datos de los programas en los que trabajamos por bastante tiempo, por ello los programadores emplean diferentes estrategias para incluir un auto guardado que no afecte al rendimiento de sistema.