Warriors

202010770 – Luis Mariano Moreira García

Resumen

El programa en cuestión es capaz de cargar información que se extrae directamente de un archivo con la extensión mundialmente conocida como XML.
El programa logra funcionar de manera totalmente eficiente ya que funciona a través de listas creadas a partir de objetos y así lograr eliminar espacio en memoria que no se utiliza por el garbage collector.

Por medio de los XML se crean los objetos y así reproducirlos directamente a una lista de pisos para que el usuario en cuestión logre poder interactuar directamente con ellos.

El usuario será capaz de cargar una lista de robots, una matriz de ciudades y esta puede generar a su vez una gráfica.

Palabras clave
XML, matriz de ciudades, objetos, robots, ordenamiento.

Abstract

This program is able to load the information from the file with the XML extension.

The program can work in the most efficient way this is because it works with lists created with python to clean space in the memory with help of the garbage collector.

With the XML the objects are generated and are loaded directly to the floor list to make the interaction between the program and the user.

The user can load a list of robots, a matrix with the cities and this can load the graph.

Keywords
XML, city matrix, objects, robots, sorting.

Introducción

Brindar un programa que ayuda directamente a los usuarios que compren directamente al ejército militar, el programa puede cargar un archivo de tipo XML para que el usuario pueda interactuar de una manera totalmente eficiente con el programa y esto se logra ya que el programa funciona a través de una interfaz gráfica totalmente intuitiva que muestra la información de una manera totalmente organizada.

El usuario será capaz de cargar las listas por medio de un archivo XML, una vez cargadas el usuario puede seleccionar una lista de ciudades que al ser seleccionada devuelve una matriz con la descripción que estaba en el XML.

El programa puede mostrar los patrones de la matriz por medio de graphviz y también describir una serie de movimientos a seguir para poder realizar los cambios que corresponden para una misión.

Desarrollo del tema

Como se ha mencionado anteriormente el programa funciona inicialmente con una lista de ciudades que es doblemente enlazada y esta es así ya que es posible el poder ordenarlas al momento de ingresar un valor, se muestra una breve descripción del algoritmo:

1. Comparar las ciudades existentes.
2. Si la cantidad de ciudades es menor a 1 entonces se insertará por cabeza o por cola.
3. En el caso de que es mayor a uno, primero mira si es mayor que la cabeza, luego mayor que la cola y si ninguna se satisface mira si es mayor que el siguiente, en caso de ser mayor pasará y buscará insertarse en su posición correspondiente.

Debido a que no se utilizaron las matrices y listas integradas directamente desde Python esto complico más el desarrollo del programa ya que se utilizaron varias clases de nodo y lista.

A continuación se enumeran las listas que se utilizaron para poder realizar el programa

a. Lista de ciudades
b. Lista de robots
c. Lista de cabeceras
d. Lista de movimientos

Lista de pisos, muestra de una manera ordenada lo que es la lista de todos los pisos disponibles, tiene una estructura simple ya que los nodos únicamente se conectan entre si por medio de un apuntador siguiente.

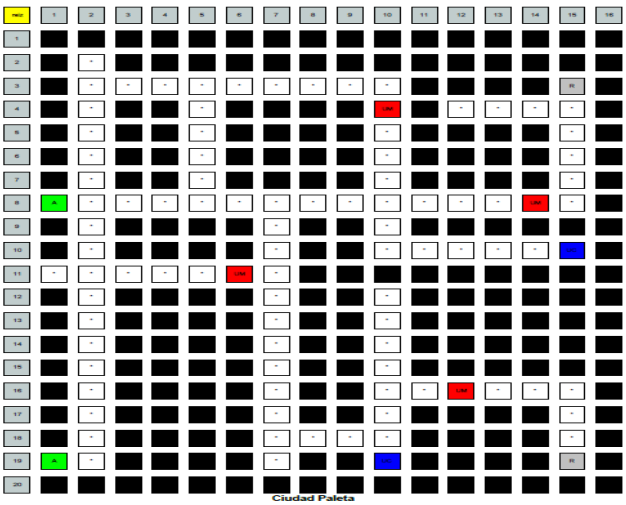


Figura 1. Estructura de la matriz de ciudades

Fuente: elaboración propia, 2022

El algoritmo en cuestión para realizar las misiones es el siguiente (No me dio tiempo de terminar el algoritmo, pero esta es la idea que tenía plateada

1. Se recorre en búsqueda de la entrada
2. Ver todos los movimientos posibles.
3. Una vez con todos los movimientos posibles se guarda en una lista la posición a mover siempre y cuando sea válida o las posiciones no estén repetidas.
4. Si se satisface esta condición se comprobará se ira pasando en nodo en nodo
5. De no satisfacer las condiciones se retornara a donde tuvo más de un movimiento posible para efectuar un nuevo movimiento (Esto tomando en cuenta a que se verifica que no se repitan las posiciones en un nuevo movimiento para no caer en el mismo error.

El algoritmo para reemplazar los valores de una ciudad que ya estaba:

1. Comprobar si es la pos diferente a 0.
2. Recorrer toda la lista buscando que el nombre de la ciudad sea diferente .
3. En caso de ser iguales se reemplazan los datos mediante setter, en caso de ser distinto a todo lo que este se realiza el ordenamiento (Ver página anterior para el algoritmo).

A pesar de que la lista este completamente enlazada, el programa no hace función o uso a esta opción ya que para este proyecto no fue necesario.

Conclusiones

El buen uso de las clases permite desarrollar programas de una manera más eficiente, ya que por medio de las listas enlazadas es posible desarrollar programas que sean óptimos para el uso cotidiano.

Por medio de las listas uno es capaz de poder realizar los métodos necesarios para llevar a la acción y así poder llevar un flujo más eficiente del programa.

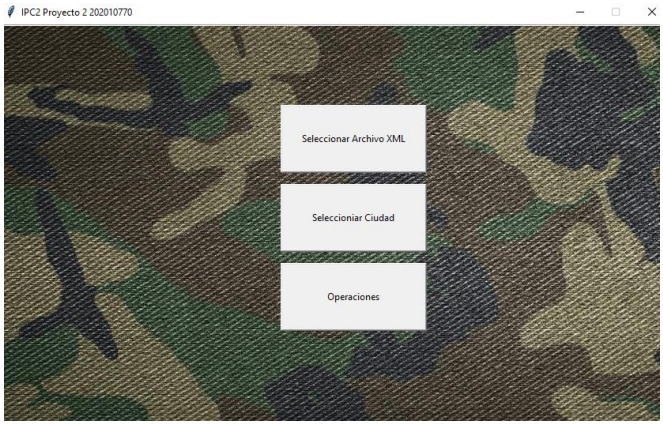
Referencias bibliográficas

1. Calzadilla, J. C. F., Herrera, A. N., & Delfino, E. La enseñanza de los arrays estáticos, dinámicos y listas enlazadas¿ cuál usar? Análisis de códigos The teaching of static arrays, dynamics and linked lists. What to use? Code analysis.
2. Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, *20*(2), 1-13.
3. Fernández, A. (2013). *Python 3 al descubierto*. Alfaomega Grupo Editor.
4. Legarreta, I. V. (1995). *Diseño de un array bidimensional dinámico implementado mediante listas enlazadas y árboles AVL* (Doctoral dissertation, Universidad de Deusto).
5. Van Rossum, G. (2007, June). Python Programming language. In *USENIX annual technical conference* (Vol. 41, No. 1, pp. 1-36).

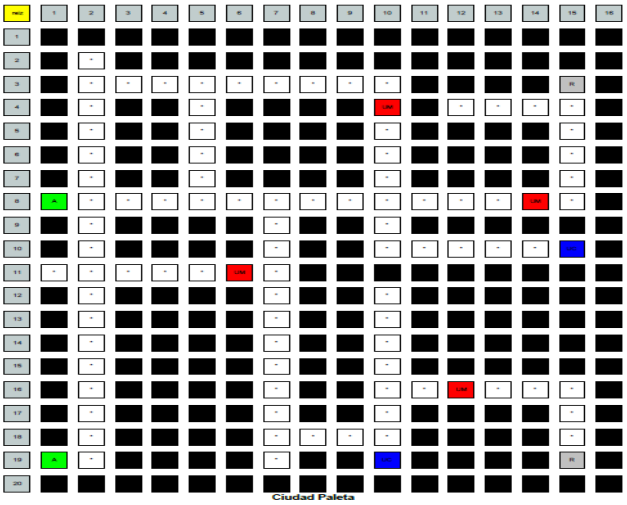
Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2022.

ANEXOS

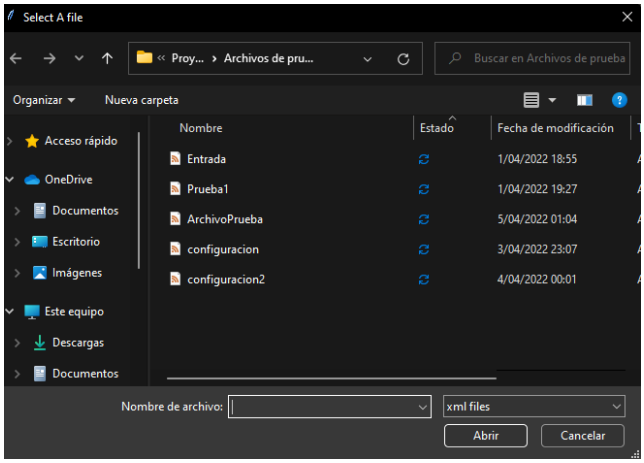
INTERFAZ:



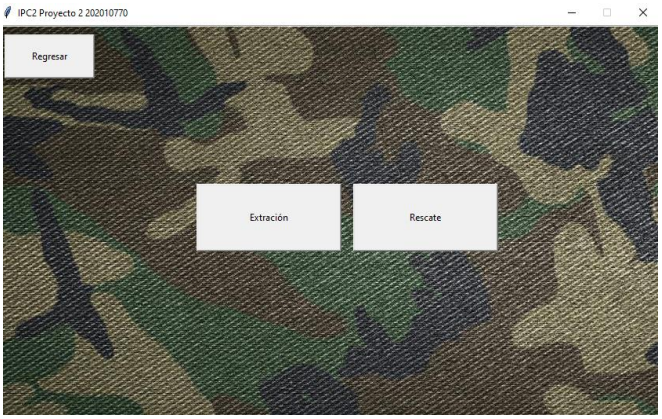
Grafica de ciudades



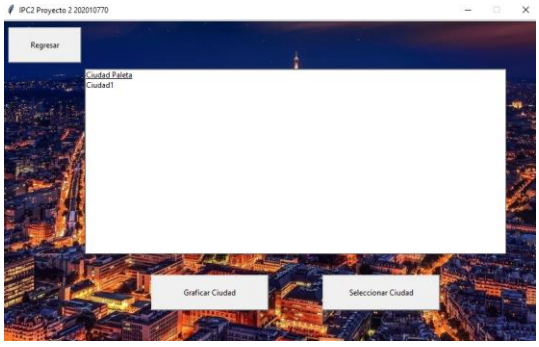
Seleccionar una ciudad XML



Seleccionar una mission



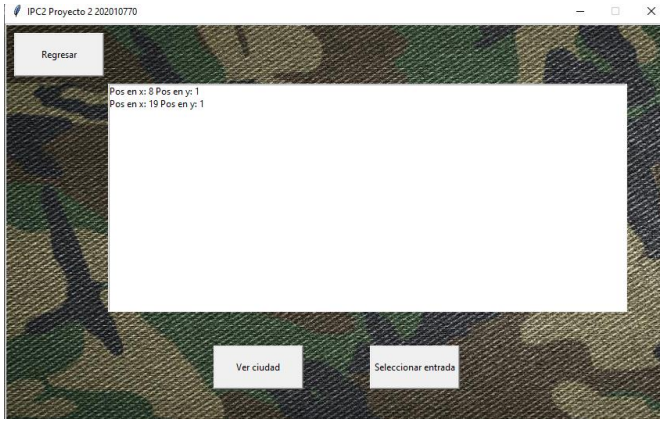
Ver ciudades



Seleccionar robot

Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2022.

Seleccionar cuando hay más de una entrada



Seleccionar cuando hay más de una salida (recurso, unidad civil)