

---

## PROYECTO 2 – Chapín Warriors, S. A.

---

202004745 – Allen Giankarlo Román Vásquez

### Resumen

El proyecto 1 "Chapín Warriors" se realiza para poner en práctica los conocimientos adquiridos en el laboratorio y clase magistral del curso "Introducción a la Programación y Computación 2" de la carrera de Ingeniería en Ciencias y Sistemas de la Universidad San Carlos de Guatemala creando un proyecto en el lenguaje de programación "Python" implementando las funciones básicas del lenguaje de programación, también aplicando los temas y conceptos teóricos aprendidos en el laboratorio, por último creando otro tipo de conocimiento para el estudiante al manejar distintos tipos de archivos, ya sean archivos de entrada para cargar en la memoria, como de salida para creación de reportes del proyecto. El proyecto contempla la implementación de TDA en una aplicación que otorgue una solución a una problemática en la vida real. Para finalizar para este proyecto se aplican los conceptos de: Lista, Matriz Dispersa, Listas enlazadas, Listas doblemente enlazadas, Apuntadores, Nodos, Archivos XML, Graphviz entre otros.

### Palabras clave

Matriz, Lista, Apuntadores, Listas enlazadas, XML.

### Abstract

*The Project 1 "Chapín Warriors" was performed for performance the knowledge acquired in the laboratory class and the masterly class "Introduction to programming and computation 2" of the Science and systems Engineering career from the University "San Carlos de Guatemala" creating a project with programming language "Python" implementing the basic functions of the programming language, also applying the theoretical topics and concepts learned in the laboratory class, and lastly, creating new knowledge for the student handling different types of files like input files to load into memory, as well as output files for creating project reports. The project contemplates the implementation of ADT in an application that provides a solution to a problem in real life. To finish, this project applies the concepts of: List, Array, Linked Lists, Doubly Linked Links, Pointers, Nodes, XML Files, Graphviz, among others.*

### Keywords

*Array, List, Pointers, Linked lists, XML.*

## Introducción

La aplicación “Chapín Warriors” consiste en un programa que cuenta con una interfaz gráfica amigable para el usuario, permitiendo la búsqueda de caminos cortos para llegar a un Recurso o una Unidad Civil que se desea acceder. También permite al usuario cargar archivos de tipo XML para tener un amplio panorama de las ciudades eligiendo entre las ciudades disponibles, robots y unidades disponibles.

## Desarrollo del tema

### app.py

En la clase app.py se define la interfaz gráfica inicial del programa que es generada con tkinter y dos funciones, la primera función es LeerXml esta función despliega un buscador de archivos en el que se va a cargar el archivo de tipo XML, luego de cargar el archivo de tipo XML, se procederá a parsearlo con Element Tree para poder manejarlo con Python, extrayendo la información que contiene, el primer for que revisa si es una ‘listaCiudades’ o ‘robots’, si es ‘listaCiudades’ el siguiente for itera todas las ciudades que contiene esta ‘listaCiudades’, el siguiente for itera cada información que trae el XML si tiene tag ‘nombre’ extrae el nombre, filas, columnas y lo agrega a un nodo, si tiene tag ‘fila’ extrae el número de fila y la cadena que graficará la ciudad, adentro de esta comparación existe un for que itera la cadena para ir agregándola al nodo que a su vez la agrega a la matriz, el último tag de cada ciudad extrae la Unidad Militar con su posición y la agrega al nodo y esta a su vez a la matriz y a la lista de Unidades militares; si es ‘robots’ hace un proceso parecido pero solamente obtiene el tipo, nombre y la capacidad si fuera el caso. Función ElegirCiudad() hace una comparación que revisa si ya se cargo el archivo XML sino tira un error, sino redirige a otra clase que es la de ListaCiudades.py y en la que lleva como parámetro la listarobots.

### ListaCiudades.py

La clase Ciudad es método constructor de la ListaCiudades en el que se encuentra el constructor, agregarMatriz y agregar ListaUnidades que agrega una Unidad Militar.

La clase ListaCiudades que tiene los apuntadores, la función de insertar que obtienen un nodo del tipo Ciudad y lo agrega a la lista, función elegir ciudad que recibe el parámetro de listarobots de app.py la cual tiene un recorrido para desplegar un combobox con la lista de ciudades, dos botones para elegir si es misión de extracción de recursos o misión de rescate. Ya sea el de misión o extracción de recursos, hace validaciones para comprobar que se pueda llevar a cabo la misión esperada, si se puede despliega una nueva interfaz gráfica en la que se puede seleccionar el objetivo, punto de entrada y el robot a utilizar. Y la misión completada, toma los atributos seleccionados en los combobox y realiza las acciones pertinentes en una nueva función que se llama misión Completada que genera el reporte con los datos guardados y con la salida esperada.

### ListaUnidades.py

Es la lista doble que almacena las Unidades Militares, tiene dos clases, la clase UnidadMilitar que es la clase constructora en la que se almacenan atributos de fila, columna y combate de cada Unidad Militar.

Y la clase ListaUnidades que cuenta con los apuntadores, y cuenta con su función insertar que recibe los atributos para crear un nodo Unidad Militar y agregarlo a la Lista.

### ListaRobots.py

Es la lista doble que almacena los Robots, tiene dos clases, la clase Robots que es la clase constructora en la que se almacenan atributos de nombre, tipo y capacidad.

Y la clase ListaRobots que cuenta con los apuntadores, cuenta con función recorrerRescate en la que se obtienen los nombres de los Robots de tipo

Chapin Rescue, la función recorrerRecursos que se obtienen los nombres de los Robots de tipo Chapin Fighter, función insertar que recibe los atributos que son nombre, tipo y capacidad, genera un nodo de tipo Robots, en ella también cuenta con un contador para ver la cantidad de robots tipo ChapinRescue o ChapinFighter.

### **MatrizCiudades.py**

Cuenta con dos clases la clase Nodo\_Celda que es el método constructor cuenta con los atributos, x, y, tipo, arriba, abajo, izquierda, derecha. Y los getters and setters para los atributos arriba, abajo, izquierda, derecha.

La clase MatrizCiudades cuenta con los atributos filas y columnas que es una Lista\_Cabecera, la función insertar recibe la posición en x, posición en y y tipo genera un Nodo\_Celda y la inserta a la Matriz Dispersa. La función ubicarRecursos recorre toda la matriz para ver la posición (x,y) de un recurso y lo registra. La función ubicarUnidadCivil recorre de nuevo toda la matriz para ver la posición (x,y) y de una unidad civil la registra. La función ubicarEntrada hace lo mismo pero con las entradas. La función recorridoPorFila recibe una fila y la recorre por columna. La función recorridoPorColumna recibe una columna y la recorre por fila. La función UbicarCoordenad, recibe, columna, fila y tipo, la ubica y la reescribe. La función graficarNeato recibe los parámetros de la MatrizDispersa y la grafica con ayuda de graphviz y genera una imagen png y un txt con el código.

### **Clase ListaCabecera.py**

Cuenta con dos clases, la clase Nodo\_Cabecera que es el nodo\_Cabecera constructor, get y set Acceso.

La clase Lista\_Cabecera es la lista\_Cabecera en la que se insertan las filas y columnas de MatrizCiudades, cuenta con insertar\_nodoCabecera que inserta un nodo de tipo Cabecera, mostrarCabeceras que recorre la Lista\_Cabecera y get\_Cabecera que devuelve un nodo cabecera.

### **Referencias bibliográficas**

- A. Donwney, (2002). *Aprenda a pensar como un Programador con Python*. Green Tea Press.
- C. Severance, (2009). *Python para todos. Explorando la información con Python 3*. Green Tea Press
- J. Bullinaria, (2019). *Data Structures and Algorithms*. Green T. School of Computer Science, University of Birmingham, UK.
- M. Acera, (2011). *XML Edición 2012*. Anaya Multimedia
- V. Das, (2006) *Principles of Data Structures using C and C++*. New Age International (P) Ltd., Publishers.

## Apéndice A

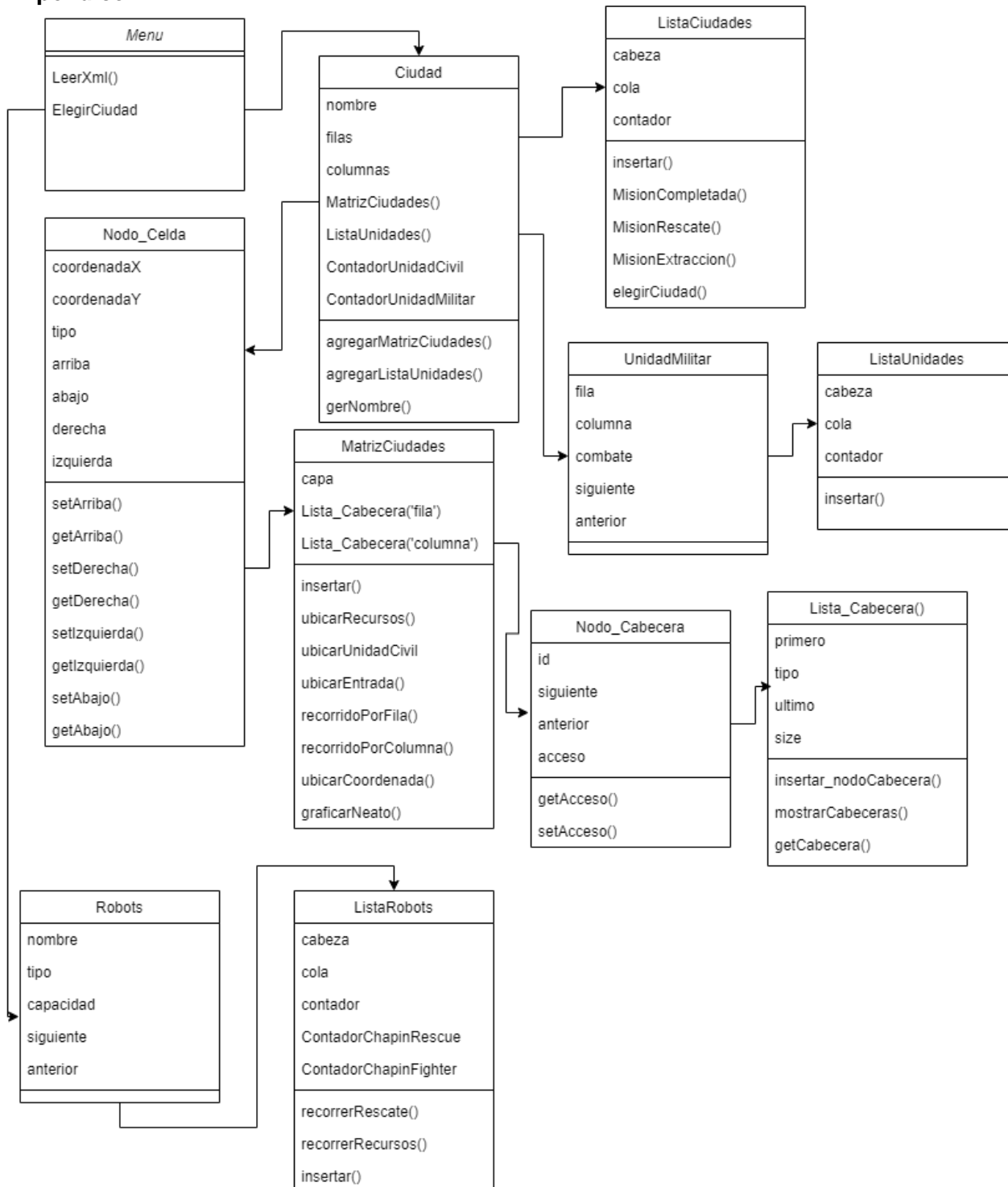


Figura VIII. Diagrama de clases proyecto 2

Fuente: Elaboración propia