

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA

ESCUELA DE SISTEMAS

ESTRUCTURA DE DATOS

Ing. Luis Espino

Aux. William Perez

FASE 2

USAC GAMES -> Batalla Naval

JOSUE ROLANDO

GRAMAJO ROLDAN

202000895

3021021080101

INDICE

Introducción.....	2
Objetivos.....	2
Conocimientos previos.....	2
Requerimientos.....	2
Descripción de clases.....	5
Diagrama de clases.....	6
Conclusión.....	7

Introducción

La empresa Usac Games desea implementar un videojuego que permitan desarrollar la agilidad mental de los usuarios, por lo cual ha planeado desarrollar una aplicación con el juego Batalla naval y le solicita a usted como estudiante de estructura de datos poder implementar algoritmos, funciones y estructuras que permitan que el juego tenga un rendimiento óptimo y fluido.

Objetivos

Guiar al lector en la manipulación del programa, así mismo dar a conocer la lógica empleada en el desarrollo del programa por si es necesario actualizar, mejorar o dar mantenimiento a la aplicación creada y descrita.

Los conocimientos que deberán tener las personas que manejen el programa son:

- Conocimiento del lenguaje C++ y Python
- Conocimiento de la biblioteca crow
- Conocimientos sobre POO
- Conocimientos sobre GUI con Pyside2
- Manejo del editor de texto Visual Studio Code.

Requerimientos mínimos para ejecutar en VScode

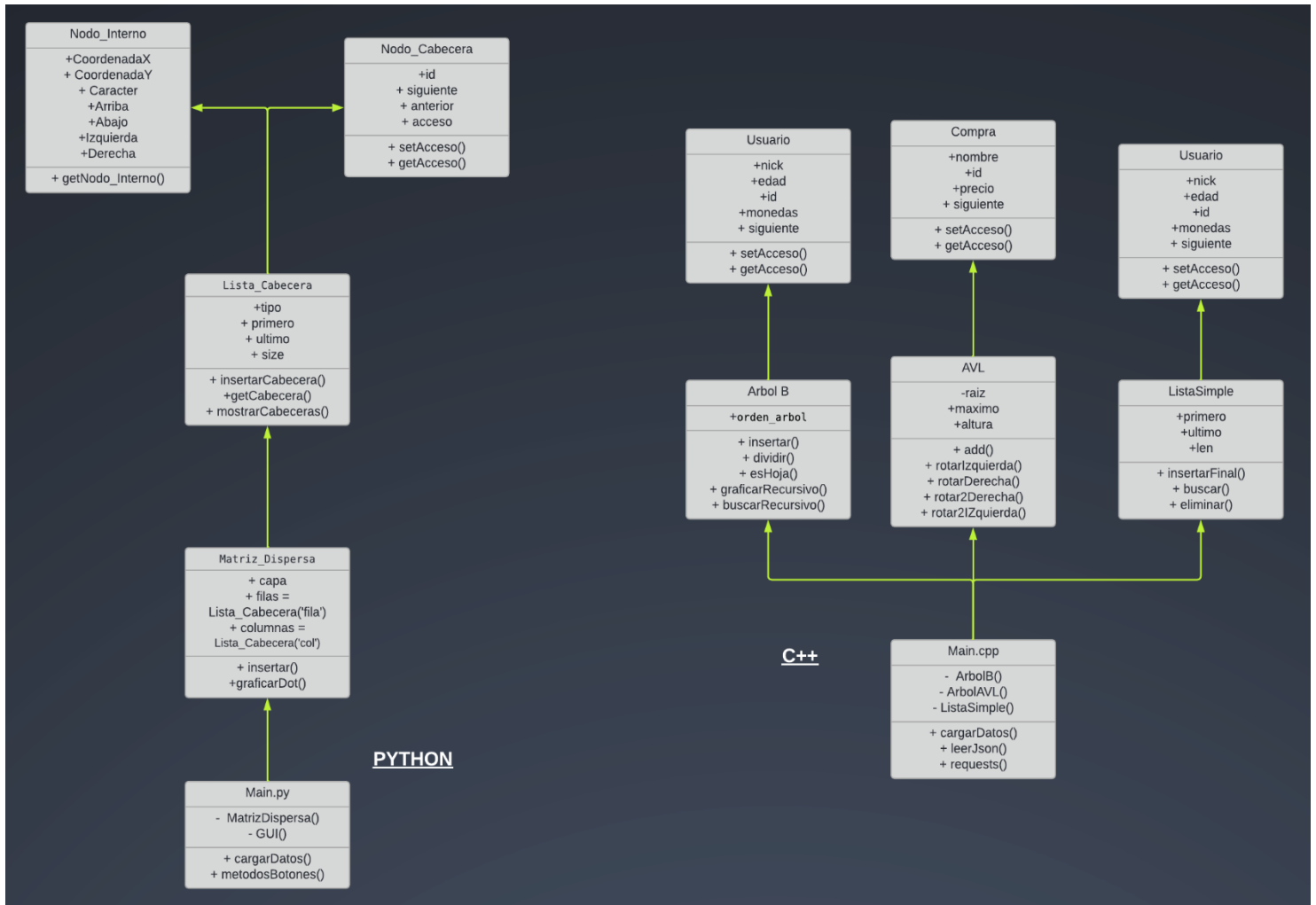
- Procesador a 1.6 GHz o superior
- 512 MB de RAM
- 750 MB de espacio disponible en el disco duro.
- Tarjeta de vídeo compatible con DirectX 9
- Linux, windows 7,8,superior

Descripción de clases

Paquete	Clase	Descripción
Client/src	<i>Lista_cabecera</i>	Esta clase es la encargada de generar las cabeceras para la matriz dispersa.
Client/src	<i>Nodo_cabecera</i>	Esta clase es un objeto de tipo Nodo Cabecera que contiene apuntadores en X y Y.
Client/src	<i>Nodo_interno</i>	Esta clase es un objeto de tipo Nodo que recibe posición en X,Y, y tiene apuntadores en todas las direcciones.
Client/src	<i>Matriz_dispersa</i>	Esta clase es la que genera la matriz dispersa y consume de todas las anteriores clases, tiene métodos de inserción, búsqueda, eliminación y grafica.
Server/src	<i>AVL.hpp</i>	Esta clase se llama desde el archivo main y es la encargada de generar el árbol AVL junto a sus métodos de inserción, búsqueda, eliminación y grafica.
Server/src	<i>ArbolB.cpp</i>	Esta clase hereda de un archivo header y es la encargada de generar el árbol B junto a sus métodos de inserción, búsqueda, eliminación y grafica.

Server/src	<i>listas.cpp</i>	Esta clase se llama desde el archivo main y le envía parámetros de tipo nodoUsuario para posteriormente manipular con métodos de inserción, búsqueda y eliminación.
Server/src	<i>Main.cpp</i>	Esta clase es la main y es la que consume las estructuras creadas en este mismo paquete, genera carga masiva, graficas y conecta la GUI de Python con C++ usando la biblioteca CROW.

DIAGRAMA DE CLASES



CONCLUSION

El objetivo de esta fase en el proyecto fue la de aplicar los conocimientos aprendidos en las últimas clases como, teoría sobre Árboles y Matriz dispersa y estructuras no líneas en general, el aprender a crear y manejar este tipo de estructuras es de mucha utilidad ya que nos demuestra la manera en la que podemos trabajar de manera dinámica con los datos y así cuidar el uso de memoria al momento de programar un ejecutable.

La matriz dispersa es un excelente caso en donde podemos ver que es mejor el utilizar una estructura no lineal para almacenar datos en memoria a comparación de una lista nativa en lenguajes como JAVA, en donde utilizamos “ `int array [][] = new int [20][20];` ” al momento de utilizar esta estructura estática estaríamos desperdiciando espacio en memoria ya que no sabemos si todos los espacios dentro de ese array se estará utilizando. Este es un caso muy notorio en donde podemos compararlo con una matriz dispersa y las diferencias y mejoras que tiene dicha matriz es superior ya que no necesitamos tener lleno todas las “casillas” de la matriz para poder formarla y no se estaría desperdiciando memoria ya que nunca se guarda en memoria un nodo no existente.