

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERIA**  
**ESCUELA DE SISTEMAS**  
**ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1**  
**Ing. Luis Espino**  
**Aux. William Perez**

**Fase 3**  
**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

En esta fase del proyecto, Usac Games desea agregar nuevas funcionalidades al juego, se debe agregar una bitácora de transacciones, registro de movimientos por cada partida y hacer uso de la tecnología blockchain para guardar el registro de las transacciones realizadas en la aplicación, relacionadas con la compra de skins.

## 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.

Conocimientos previos

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

- Conocimiento del lenguaje c++ y Python
- Conocimiento de las bibliotecas CROW
- Conocimientos sobre POO
- Conocimientos sobre GUI con Pyside2
- Manejo del editor de texto Vscode

## Requerimientos mínimos para ejecutar en Vscode

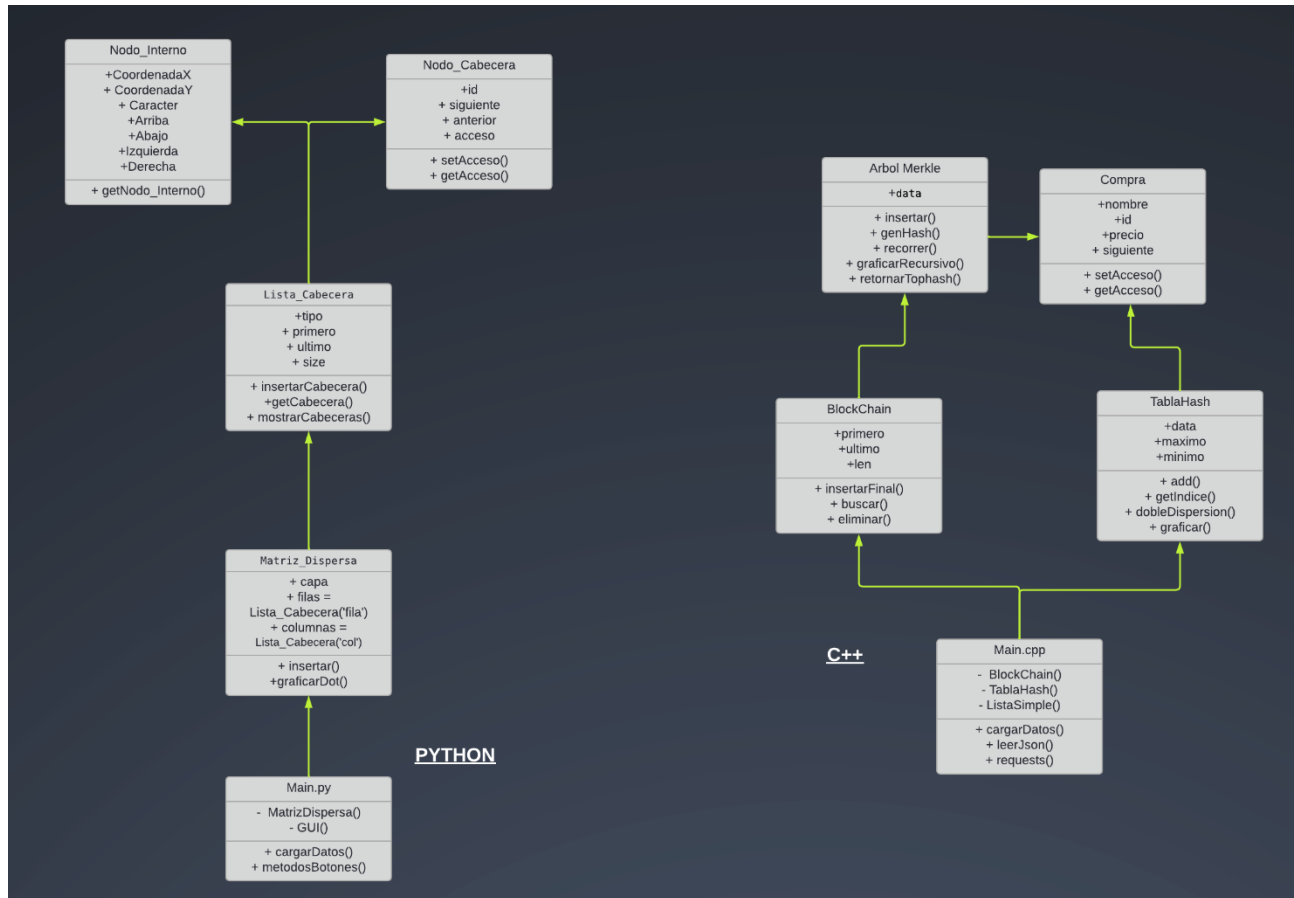
- Compilador de c++ (g++)
- Procesador a 1.6 GHz o superior
- 512 MB de RAM
- 750 MB de espacio disponible en el disco duro.
- Tarjeta de vídeo compatible con DirectX 9
- Linux o windows 7,8,superior

## 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>Thash.cpp</i>	Esta clase se llama desde el archivo main y es la encargada de la tabla hash utilizando métodos de inserción y manejo de colisiones.
Server/src	<i>Merkle.cpp</i>	Esta clase hereda de un archivo header y es la encargada de generar el árbol de merkle y posteriormente comenzar el análisis de la blockchain.
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



## Conclusión

El objetivo de esta fase en el proyecto fue la de aplicar los conocimientos aprendidos en las últimas clases como, teoría sobre Blockchain, Tablas hash y criptografía 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 y a su vez aprender de nuevas tecnologías como Blockchain.

La Blockchain es un excelente caso en donde podemos ver el uso de la criptografía ya que al no haber ninguna empresa centralizada con esta tecnología, se crearon algoritmos de encriptación para poder mantener segura y libre de “Trampas” las transacciones que se generan, la Blockchain utiliza el algoritmo ‘sha256’ el cual nosotros también implementamos en este proyecto.

En general la Blockchain no es más que una lista simple enlazada que en su interior contiene Nodos de tipo transacción, y estos nodos a su vez también tienen información encriptada pero la información sobre los productos comprados son totalmente legibles ya que se debe de mantener la fiabilidad de las compras.