

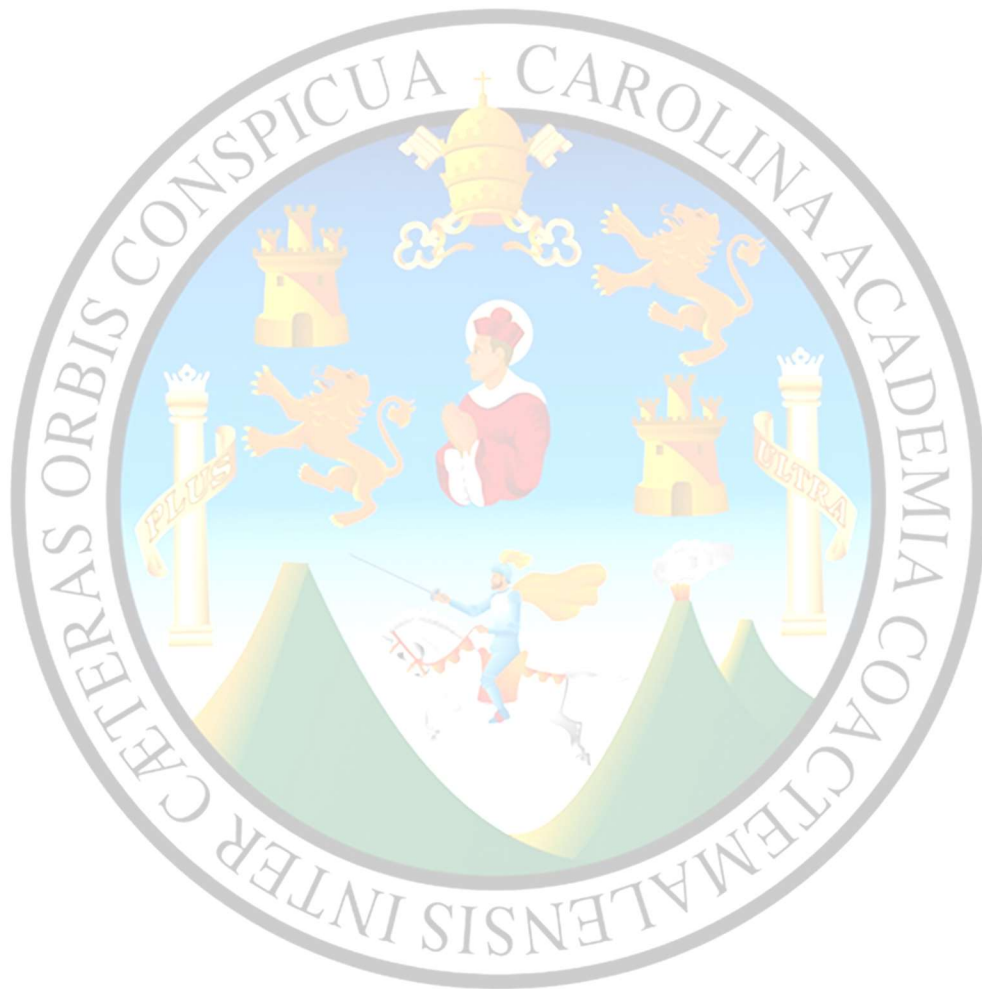
Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Estructuras de datos - Sección A

Catedrático: Ing. Alvaro Obrayan Hernandez Garcia

Tutor académico: Alex René Lopez Rosa



# Manual Técnico Battleship



Indice

Objetivos ..... 3



## Objetivos

### Objetivo general

- Aplicar los conocimientos del curso Estructuras de Datos en el desarrollo de una aplicación que permita manipular la información de forma óptima

### Objetivos específicos

- Demostrar los conocimientos adquiridos sobre estructuras de datos lineales poniéndolos en práctica en el desarrollo del juego batalla naval.
- Utilizar lenguaje C++ para implementar estructuras de datos lineales.
- Utilizar la herramienta graphviz para graficar estructuras de datos lineales.
- Definir e implementar algoritmos de búsqueda, recorrido y eliminación.



## Requisitos mínimos

Para el funcionamiento óptimo del programa se requiere:

- Espacio en disco mínimo de 200 Mb
- Memoria RAM de 2 GB
- Poseer el compilador MinGw instalado en el sistema.
- IDE codeblocks instalado.



## Librerías utilizadas

Para el desarrollo de este programa se utilizaron las siguientes librerías externas:

- JsonCpp
- Sha256

Estas se encuentran en los siguientes repositorios:

<https://github.com/open-source-parsers/jsoncpp>

<https://github.com/gerardet46/GerryVideos/tree/master/Encriptación%20C%2B%2B%20-%20SHA-256>

## Métodos utilizados

Los métodos void no tendrán el tipo de método al lado izquierdo.

Método	Ubicación	Descripción breve
menu();	Main.cpp	Este método muestra el menú de inicio.
carga();	Main.cpp	Este método lee el archivo json y lo carga a las estructuras.
registrar();	Main.cpp	Método que permite la creación de usuarios.
login();	Main.cpp	Método que permite iniciar sesión.
reportes();	Main.cpp	Método que permite acceder al submenú de reportes.
sesion_iniciada(string);	Main.cpp	Método que redirecciona al menú de sesión iniciada
eliminar cuenta(string usuario);	Main.cpp	Método que elimina la cuenta en la cual está iniciada la sesión.
tutorial();	Main.cpp	Muestra un tutorial del juego.
tienda(string usuario);	Main.cpp	Método que muestra la tienda de objetos.
movimientos();	Main.cpp	Método que almacena los movimientos a realizar.
modos_de_juego();	Main.cpp	Método que permite seleccionar el modo de juego.
void insertarCola(int x, int y);	Colah.h	Método que permite insertar las jugadas del tutorial en una cola.
void desplegarCola(int x, int y);	Colah.h	Método que despliega las jugadas en consola.
void dequeue();	Colah.h	Método para hacer dequeue en la cola.
void insertarNodo();	Lista_usuarios.h	Método de inserción de nodo en la lista circular doblemente enlazada
bool buscarNodo(std::string usuario);	Lista_usuarios.h	Método de búsqueda de nodo en la lista circular doblemente enlazada
bool buscarLogin(std::string usuario, std::string contrasena);	Lista_usuarios.h	Método para iniciar sesión
void modificarNodo(std::string usuario);	Lista_usuarios.h	Método para editar información del usuario.



void eliminarNodo(std::string usuario);	Lista_usuarios.h	Método para eliminar una cuenta ya existente.
void desplegarListaPU();	Lista_usuarios.h	Método para desplegar la lista de usuarios.
void insertarmasivo(std::string usuario, std::string contrasena, int monedas, int edad);	Lista_usuarios.h	Método para insertar usuarios por medio de carga masiva
void Ordenarascendente();	Lista_usuarios.h	Método para ordenar los usuarios por edad.
std::string cifrar(std::string contrasena);	Lista_usuarios.h	Método por el cual se encriptan las contraseñas con algoritmo sha256.