

Universidad San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y sistemas

Estructuras de Datos

Ingenieros:

- Ing. Luis Espino
- Ing. Edgar Ornelis
- Ing. Álvaro Hernández

Auxiliares:

- Alex Lopez
- Walter Mach
- Wilfred Perez



Proyecto 1 Fase 1

USAC Games, Batalla naval

Índice

Objetivos	3
Objetivo general	3
Objetivos específicos	3
Descripción General del Proyecto	4
Fase 1 (20 pts - C++)	4
Fase 2 (30 pts - C++ y Python)	4
Fase 3 (40 pts - C++ y Python)	4
Consideraciones	4
Fase 1	5
Listado de estructuras a implementar	5
Menú	5
Modalidades de juego	9
Reportes	10
Observaciones	10
Entregables	11
Restricciones	11
Fecha de Entrega	11

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 el 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.

Descripción General del Proyecto

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.

Debido al alcance y complejidad de las funcionalidades del videojuego se ha decidido dividirlo en 3 fases. A continuación se describen de forma general las funcionalidades a cubrir en cada una de las fases.

Fase 1 (20 pts - C++)

- **Listas:** Registro de usuarios
 - **Encriptación:** Aplicar seguridad a la información de los usuarios
- **Lista de listas:** Tienda de Skins del juego (los puntos por partida serán la moneda)
- **Pila:** Retroceder jugadas (Push y pop de movimientos)
- **Cola:** Tutorial del juego (push y pop de información con movimientos)

Fase 2 (30 pts - C++ y Python)

- **Matrices:** Tablero del juego y eventos para realizar movimientos
- **Árbol B:** Indexación de usuarios por orden alfabético
- **Árbol AVL:** Compras del usuario ordenadas por precio.

Fase 3 (40 pts - C++ y Python)

- **Tabla Hash:** Bitácora de eventos del sistema
 - Registro de usuarios, skins, compras de los usuarios y jugadas realizadas
- **Grafos:** Lista adyacente y grafo de movimientos por partida del usuario
- **BlockChain:** Transacciones al realizar compras

Consideraciones

1. Las funcionalidades están descritas de forma general, las mismas pueden sufrir algunas modificaciones en el enunciado final de cada fase.
2. La funcionalidad correcta del videojuego será cubierto por las 3 fases, por lo que de no realizar una de las fases, supondrá un mayor esfuerzo por parte del estudiante para completar la fase siguiente, el desarrollo de las estructuras se realizará en C++ y la funcionalidad visual se realizará en Python.

Fase 1

En esta fase del proyecto Usac Games desea que se puedan implementar las funcionalidades del juego que permitirán gestionar la información de los usuarios y las skins del juego (fondo del tablero, imágenes de los barcos), por lo cual, a continuación se describen las estructuras y el funcionamiento a implementar.

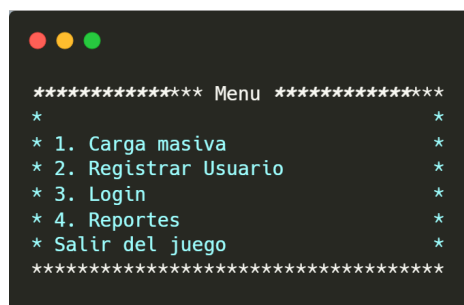
Listado de estructuras a implementar

- **Listas:** Registro de usuarios
 - **Encriptación:** Aplicar seguridad a la información de los usuarios
- **Lista de listas:** Tienda de Skins del juego (los puntos por partida serán la moneda)
- **Pila:** Retroceder jugadas (Push y pop de movimientos)
- **Cola:** Tutorial del juego (push y pop de información con movimientos)

Menú

En esta fase no se implementará interfaz gráfica, por lo cual, para poder utilizar las estructuras se utilizará un menú en consola mismo menú que en la siguiente fase se implementará de forma gráfica, el menú debe contar con al menos las siguientes opciones:

1. Carga masiva de información
 - a. Se podrá cargar un JSON con toda la información inicial de la aplicación, los datos que tendrá dicho archivo son:
 - Usuarios
 - Tutorial
 - Artículos de la tienda
2. Registrar usuario
3. Login: luego de ingresar el usuario tendrá las siguientes opciones
 - a. Editar información
 - b. Eliminar cuenta
 - c. Ver el tutorial
 - d. Ver artículos de la tienda
 - e. realizar movimientos
 - Finalizar partida
 - f. Salir al menú principal
4. Reportes
5. Salir del juego



```
***** Menu *****
*
* 1. Carga masiva      *
* 2. Registrar Usuario *
* 3. Login             *
* 4. Reportes          *
* Salir del juego      *
*****
```

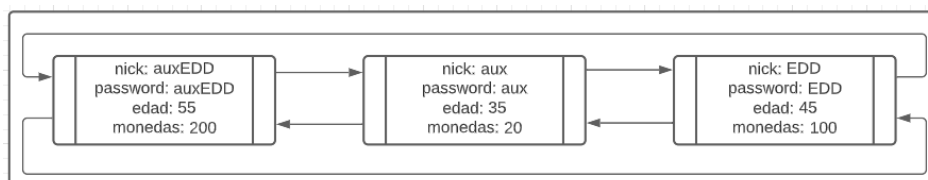
Ejemplo de menú

Carga masiva de información

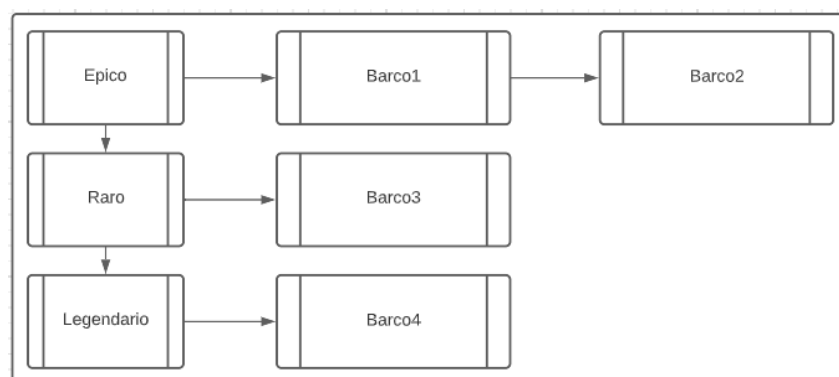
Esta opción permite que se cargue toda la información inicial de la aplicación, dicha información estará contenida en un archivo JSON con la siguiente estructura:

```
{
  "usuarios": [
    { "nick": "", "password": "", "monedas": "", "edad": "" },
    { "nick": "", "password": "", "monedas": "", "edad": "" },
    { "nick": "", "password": "", "monedas": "", "edad": "" }
  ],
  "articulos": [
    { "id": "", "categoria": "", "precio": "", "nombre": "", "src": "" },
    { "id": "", "categoria": "", "precio": "", "nombre": "", "src": "" },
    { "id": "", "categoria": "", "precio": "", "nombre": "", "src": "" }
  ],
  "tutorial": {
    "ancho": "",
    "alto": "",
    "movimientos": [
      { "x": "", "y": "" },
      { "x": "", "y": "" },
      { "x": "", "y": "" }
    ]
  }
}
```

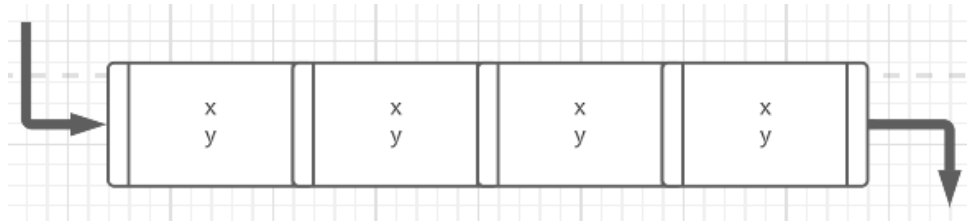
- Usuarios: este bloque del archivo JSON cuenta con un arreglo de n usuarios, los cuales se insertarán en una lista circular doblemente enlazada, tomar en cuenta que la contraseña se debe encriptar con (sha256)



- Artículos: este bloque del archivo JSON cuenta con un arreglo de n artículos relacionados por su categoría, se utilizará una lista de listas para guardar la información.



- tutorial: este bloque del archivo JSON cuenta con la información de las dimensiones del tablero y con una lista de coordenadas de cada uno de los movimientos que se ejecutarán en el tutorial, esta información se guardará en una cola, en donde el primer nodo posee la información de las dimensiones del tablero.



Editar información

En este apartado cada usuario podrá modificar su información personal.

```

Nick: Usuario
Edad: 12
Password: *****

```

Eliminar cuenta

Con esta opción el usuario podrá darle de baja a su cuenta, eliminando toda su información de forma permanente incluyendo todos los artículos adquiridos.

```

Desea eliminar su cuenta permanentemente [y/s] :

```

Ver el tutorial

En esta opción se podrán visualizar ordenadamente cada uno de los pasos que el tutorial contenga haciendo uso de los métodos de una cola.

```

Tutorial
Tablero:
  Ancho: 10
  Alto: 30
Movimientos:
  (1,6) -> (3,9) -> (5,1) -> (5,8) -> (6,3) -> (0,0)

```

Ver artículos de la tienda

En este apartado se podrán visualizar el listado indexado de skins con su respectiva categoría y precio, las cuales se podrán comprar indicando el número del ítem a comprar.

```

Total Tokens 1000

Tienda
Id  Nombre                      Categoria  Precio
1   Barco naval army            Legendaria 600
2   Barco naval rojo            Comun      200
3   Barco naval azul            Comun      200
4   Barco naval guerra          Epico      400
5   Barco naval especial        Epico      400
6   Barco OVNI                  Legendaria 600

Elija opcion a comprar:

```

Realizar movimientos

En este apartado se podrán generar listados de movimientos, primero se ingresarán las coordenadas deseadas, luego se procederá a guardar este listado de movimientos en una pila a la cual se le agrega un nombre y se guardará en su listado de movimientos.

```

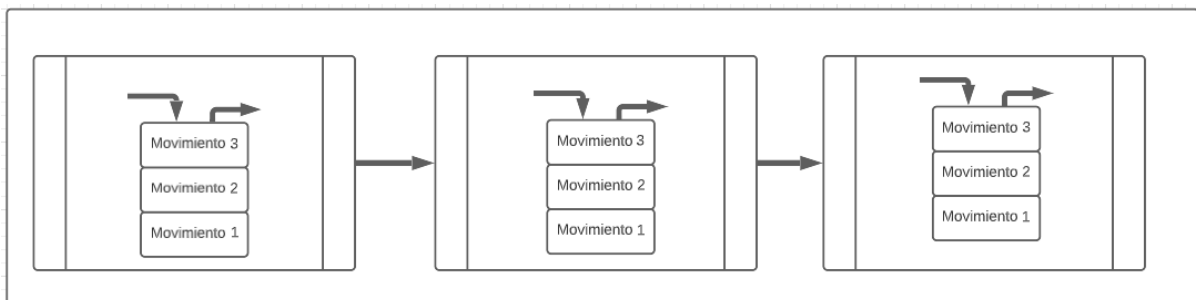
Tokens: +1

Realizar movimientos

1- 5,6
2- 8,3
3- 2,5
4- 6,3
5- 2,7

Nombre para guardar movimientos: Ganador

```



salir al menú principal

Devuelve el menú principal haciendo logout de la cuenta.

Modalidades de juego

1 Jugador

Esta modalidad consiste en hundir los barcos del tablero enemigo (computadora), los barcos serán colocados de forma aleatoria en el tablero.

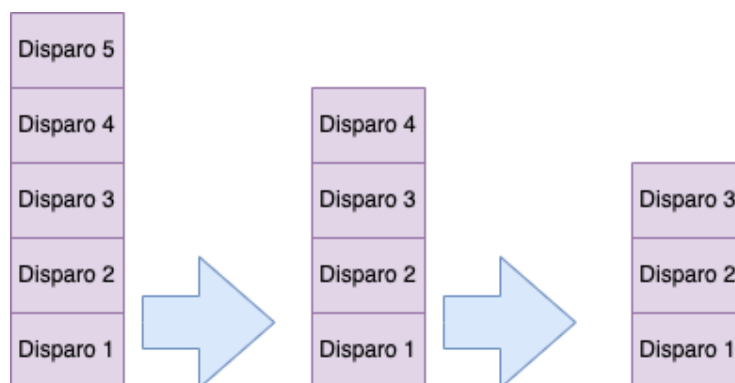
Nota: En esta fase no se implementará la generación del tablero y el posicionamiento de los barcos

Funcionalidades

Retroceso de jugadas

Se deberá contar con un mecanismo que permita almacenar las jugadas que el jugador realiza, para poder contar con la opción de retroceder y restablecer el estado del tablero en cada movimiento ejecutado.

El funcionamiento es el siguiente:



Para esta fase únicamente se debe implementar la estructura indicada que cumpla con el funcionamiento solicitado e implementar los métodos de inserción y eliminación utilizando “jugadas simbólicas”, las jugadas únicamente deberán almacenar la coordenada del disparo.

Reportes

- Visualización gráfica de cada una de las estructuras utilizadas
 - Lista doblemente enlazada circular (Usuarios)
 - Lista de listas (Articulos)
 - Cola de movimientos (Tutorial)
 - Lista de pilas (Listado de jugadas)
- Listado de usuarios ordenados por edad, de forma ascendente o descendente
- Listado de artículos ordenados por precio , de forma ascendente o descendente

Observaciones

- Lenguaje de Programación: C++
- Sistema Operativo: Elección del estudiante.
- El IDE a utilizar queda a discreción del estudiante.
- Librería para graficar las estructuras: Graphviz
- Los archivos de entrada serán documentos en formato JSON (.json)
- El estudiante debe tener un repositorio privado en github con el nombre [EDD_2S]BatallaNaval_#carnet, se crearán 3 carpetas fase1, fase 2 y fase 3.
- Agregar a su tutor como colaborador al repositorio del proyecto.
 - Auxiliar 1: wltomv
 - Auxiliar 2: aexlopez@gmail.com
 - Auxiliar 3: willop
- Se entregará en UEDI un .rar con los entregables solicitados.
- Las copias tendrán nota de 0 puntos y serán reportadas al catedrático y a la escuela de sistemas.

Entregables

- Manual Técnico
- Compilado de C++
- Código fuente en un .rar
- Link a repositorio con el código fuente.

Restricciones

- Las estructuras deben de ser desarrolladas por los estudiantes sin el uso de ninguna librería o estructura predefinida en el lenguaje a utilizar.
- No se permite la modificación de código durante la calificación, únicamente se calificará sobre el ejecutable entregado en UEDI.
- **Última fecha para aceptar invitación de GitHub será el 24 de agosto**

Fecha de Entrega

- 24 de agosto a las 23:59 horas.
- Se aceptarán entregas posteriores con penalización de 8%, 13%, 21% y 34%, por cada día de atraso hasta el domingo 28/agosto.