

Tp final LAB2

Alumnos: Brocardo Sofia, Cunsolo Mateo, Oliva Giuliana.

Tiene que tener:

- Archivos
- Recursión
- Listas
- Filas -----> incorporar en el juego
- Arboles
- Estructuras compuestas

```
typedef struct Jugador
```

```
{  
  
    char alias[50]; //irrepetibles  
  
    char password[50];  
  
    int partidasJugadas;  
  
    int partidasGanadas;  
  
    int partidasPerdidas;  
  
} Jugador;
```

```
typedef struct Partida
```

```
{  
  
    char jugador[50]; //para buscar datos del jugador  
  
    int activo;  
  
    int puntaje;  
  
} Partida;
```

```
typedef struct NodoPuntaje
```

```
{
```

```
    Partida puntaje;

    int idPartida;

    struct * NodoPuntaje siguiente;
} NodoPuntaje;
```

```
-----

typedef struct nodoArbol
{
    Jugador player;

    NodoPuntaje* partidas Jugadas;

    nodoArbol *izq;

    nodoArbol *der;
};
```

```
-----

typedef struct
{
    char name[50];

    char password[50];
}Usuario;
```

```
-----

typedef struct
{
    registro_partida dato;

    struct nodoTOP* siguiente;
}nodoTOP;
```

```
-----

typedef struct
{
```

```
char jugador[50]; //para buscar datos del jugador

int activo;

int puntaje;

int idPartida;

}registro_partida;
```

```
-----

typedef struct top10

{

    Partida puntaje[10];

    int validos;

}
```

Árbol de jugadores (Árbol de lista simple)

- Ordenado por string de nombre de la estructura Jugador, para buscar más rápido a un jugador
- El árbol contiene una lista simple de puntajes en cada nodo y un jugador
- El árbol se guarda en un archivo que se sobrescribe

Puntaje del jugador

- Se agregan nodos al final de la lista
- Se puede mostrar:
 - En orden del puntaje más viejo al más nuevo
 - En orden del puntaje más nuevo al más viejo de manera recursiva

TOP 10 puntajes

- Es un arreglo de 10 puntajes que se guarda en un archivo top10puntajes, siempre se sobrescribe
- Si validos es menor a 10 el puntaje ingresa siempre en orden
- Si validos es 10 se compara el puntaje a ingresar con el del índice 9 (el puntaje más bajo) y si es menor se busca el hueco y se reemplaza, descartando al último jugador

- Cada vez que un jugador ingresa al top el arreglo se guarda en un archivo que contiene el historial de top 10, este puede ser accedido desde el modo admin

Modo admin

- Se puede:
 - Borrar a un jugador, en caso de borrarlo se lo guarda en un archivo de JugadoresBorrados
 - Reincorporar a un Jugador borrado
 - Mostrar de TOP10

Modo Jugador

- Se puede
 - Ingresar con alias y contraseña o crear un jugador
 - Jugar una nueva partida
 - Ver top10
 - Ver analíticas de jugador
 - Ver historial de partidas del jugador

Librerías

- Jugador
- Árbol
- Puntaje
- Lista
- Juego

Juego

const NIVELES 5

const MAX ?? //máxima cantidad de muros que puede tener

typedef struct Coordenada

```
{
    int ejeX;//coordenadas de cada muro o inicio/fin
    int ejeY;
}
```

```
typedef struct Mapa
```

```
{
    int tiempo; //tiempo de cada nivel
    Coordenada muro [MAX]; //contiene las coordenadas de cada muro
    int validosMuro;
    Coordenada inicio;
    Coordenada fin;
    char cancion[30];
    int colorMapa;
}
```

```
typedef struct NodoNivel
```

```
{
    Mapa nivel;
    struct * NodoNivel siguiente;
}NodoNivel;
```

```
typedef struct FilaNivel
```

```
{
    struct * NodoNivel primero;
    struct * NodoNivel ultimo;
}FilaNivel;
```

las coordenadas se cargan primero en un arreglo de coordenadas para dibujar en el mapa

cuando carga el nivel a jugar las coordenadas de cargan en una matriz de 6pixels mas de alto para verificar mas rapido la posicion que se quiere mover, esta tiene 0 (no hay muro), y 1(hay muro) en sus respectivas posiciones