

UNIVERSIDAD SAN CARLOS DE GUATEMALA



MANUAL TECNICO

PROYECTO FASE 1 <NAVAL WARS>

ESTRUCTURAS DE DATOS SECCION A

EDWAR EVERALDO ZACARIAS

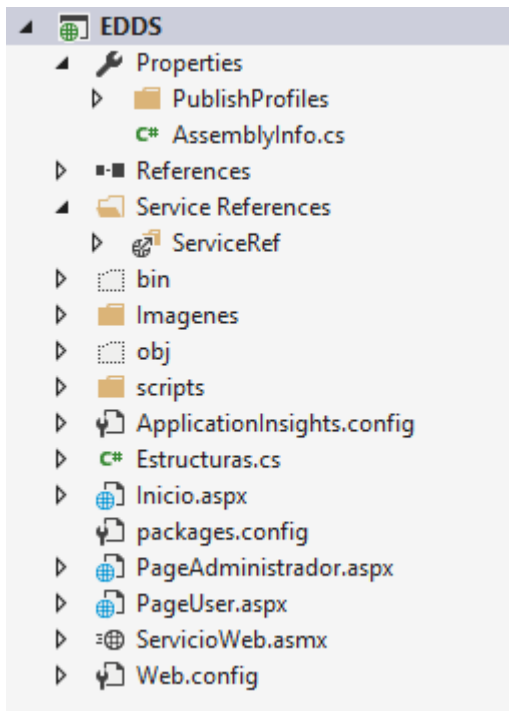
201503986

Proyecto Fase 1 <Naval Wars>

La aplicación titulada Naval Wars está desarrollada sobre el lenguaje C# en el entorno de desarrollo de Visual Estudio 2015, como una aplicación ASP.Net, para proveer de una correcta funcionalidad a la aplicación se realizaron las siguientes inclusiones de librerías:

```
using System.IO;
```

además de la necesidad de utilizar la siguiente estructura interior de archivos necesarios para realizar todo el funcionamiento solicitado:



Las distintas estructuras implementadas están contenidas principalmente en los archivos detallados a continuación:

Árbol Binario de Búsqueda:

Es la estructura utilizada para el almacenamiento y manejo de todos los usuarios registrados en el sistema de la aplicación, conteniendo en detalle los atributos necesarios para su manejo interno, tales como: Nickname, contraseña, estado de conexión, correo, entre otros datos. Con el siguiente código de clase y constructor:

```
public NodoArbol(String Nick, String Pass, String Mail, int Log,
ListaDelArbol lst)
{
    this.Nickname = Nick;
    this.Contrasena = Pass;
    this.Correo = Mail;
    this.Conectado = Log;
    this.Izq = null;
    this.Der = null;
    if (lst == null) { this.Lista = new ListaDelArbol(); }
    else { this.Lista = lst; }

}
public int Perdidas;
public int Ganadas;
public int Destruidas;
public String Nickname;
public String Contrasena;
public String Correo;
public int Conectado;
public NodoArbol Izq;
public NodoArbol Der;
public ListaDelArbol Lista;
}

public NodoArbol Raiz;
public NodoArbol Actual;
public int Contador = 0;
public int Altura = 0;
public int NodosHoja = 0;
public int NodosRama = 0;
public int CantNodos = 0;
public ListaNodoArbol ListaNodos;
```

Lista doblemente enlazada:

Esta estructura se implementó en varios ámbitos del programa, siendo utilizada para almacenar el historial de las partidas jugadas por cada usuario, para contener los nodos que se extraen del árbol original para crear su árbol espejo, para la elaboración de las listas top de las partidas, y finalmente para los encabezados de la matriz ortogonal. Siendo su constructor y clase los siguientes:

```
public class NodoLA
{
    public NodoLA(String Nick, String Pass, String Mail, int Log,
        ListaDelArbol lst, int Gan, int Des)
    {
        this.Nickname = Nick;
        this.Contrasena = Pass;
        this.Correo = Mail;
        this.Conectado = Log;
        this.Ant = null;
        this.Sig = null;
        this.Lista = lst;
        this.Ganadas = Gan;
        this.Destruidas = Des;
    }
    public int Perdidas;
    public int Ganadas;
    public int Destruidas;
    public String Nickname;
    public String Contrasena;
    public String Correo;
    public int Conectado;
    public NodoLA Sig;
    public NodoLA Ant;
    public ListaDelArbol Lista;
}

public NodoLA primero;
public NodoLA ultimo;
```

Matriz Ortogonal

Es la estructura principal para la funcionalidad de la metodología de juego de la aplicación, permitiendo un mecanismo de coordenadas manejado en tres dimensiones a través de una matriz de dos dimensiones y tres niveles adicionales, haciendo un total de cuatro niveles. La clase y constructor de la matriz es de la siguiente manera:

```
public LE EF = new LE();
public LE EC = new LE();
public class Nodom
{
    public Nodom(int fil, int col, int niv, int vis, String jug, String
    uni, int mov, int alc, int dan, int vid)
    {
        this.fil = fil;
        this.col = col;
        this.niv = niv;
        this.vis = vis;
        this.jugador = jug;
        this.unidad = uni;
        this.movimiento = mov;
        this.alcanse = alc;
        this.dano = dan;
        this.vida = vid;
        this.abj = null;
        this.arr = null;
        this.der = null;
        this.izq = null;
        this.sup = null;
        this.inf = null;
    }
    public int fil;
    public int col;
    public int niv;
    public int vis;
    public String jugador;
    public String unidad;
    public int movimiento;
    public int alcanse;
    public int dano;
    public int vida;
    public Nodom der;
    public Nodom izq;
    public Nodom arr;
    public Nodom abj;
    public Nodom sup;
    public Nodom inf;
}
```