



Universidad
Católica del Norte

Taller 0

Programación Avanzada

Elias Manque Olivares

20.475.385-7

18/04/2022

Contenido

Problema

[Estructura de Datos](#)

Pag. 4

[Código](#)

Pag. 5

Inicio

[Menu Admin](#)

Pag. 6

..... **agregarEnemigos()**

Pag. 7

..... agregarHechizos()

Pag. 8

..... estadisticasJugadores()

Pag. 8

[Usuario](#) – usuarioBusqueda()

Pag. 9

..... **usuarioBusqueda()**

[Menu Usuario](#)

Pag. 10

..... **JuegoJCE()**

..... **JuegoJCJ()**

..... **aprenderHechizo()**

..... **busquedaEstadistica()**

..... **hechizoJugador()**

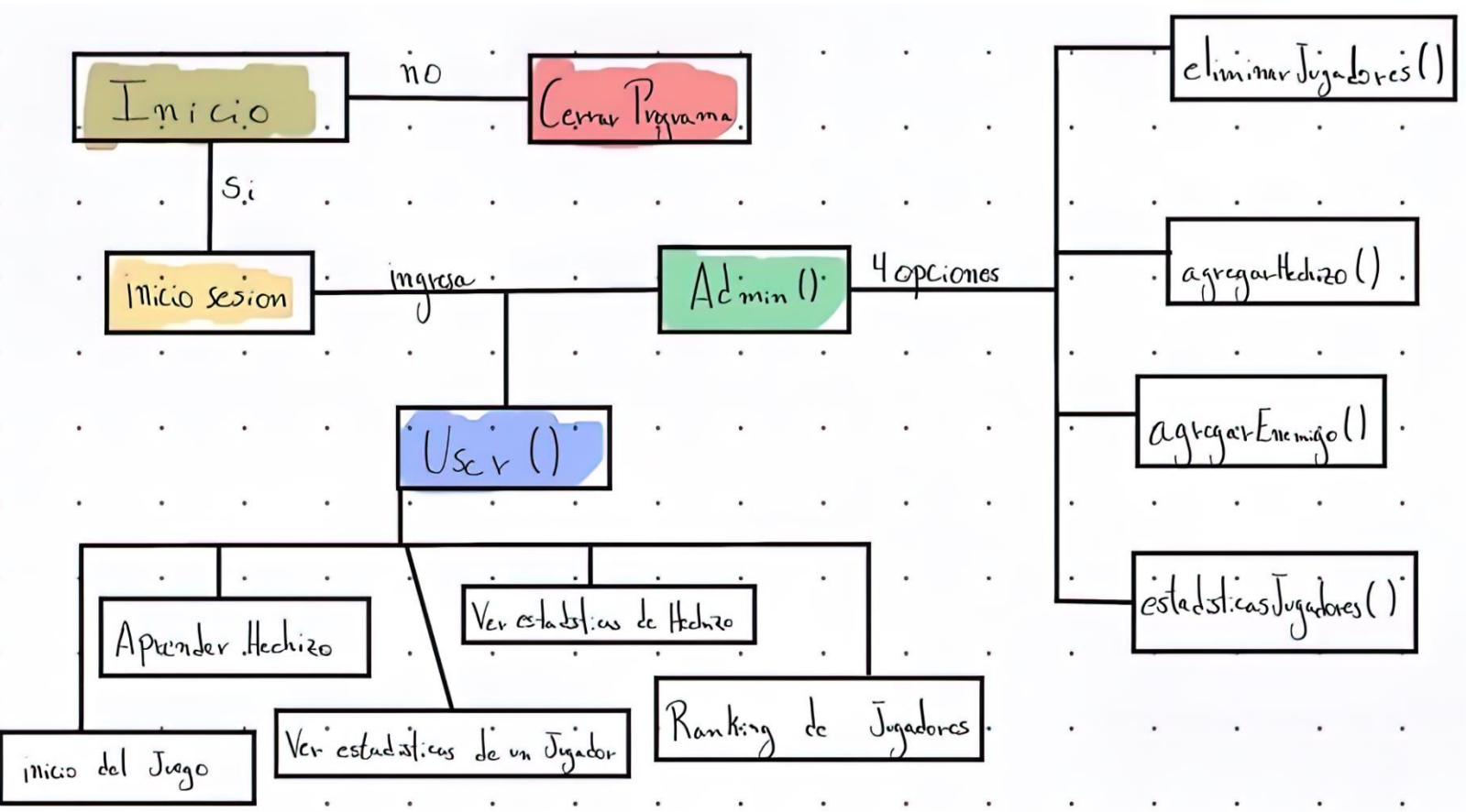
..... **podio()**

Repositorio GitHub



Problema (Magic Art Online)

Estructura de Datos



Enemigos.txt



Hechizos.txt



Jugadores.txt



HechizosJugadores.txt

Código

```
public static void main(String[] args) throws IOException {
    Scanner leer = new Scanner(System.in);
    // Variables a Usar
    String user, password;
    // Inicio de programa
    System.out.println("#####");
    System.out.print("Desea iniciar el programa? SI/NO ==> ");
    String confir = leer.nextLine();
    System.out.println("#####");
    confir = confir.toUpperCase();

    while (!(confir.equals("SI")) && !(confir.equals("NO"))) {
        System.out.println("Ingrese un valor valido");
        System.out.print("Desea iniciar el programa? SI/NO ==> ");
        confir = leer.next();
        System.out.println("");
        System.out.println("#####");
        confir = confir.toUpperCase();
    }
    if (confir.equals("SI")) {
        while (confir.equals("SI")) {
            System.out.print("Ingrese su nombre de Usuario ==> ");
            user = leer.next();
            System.out.print("Ingrese su Contraseña ==> ");
            password = leer.next();
            // Division entre Usuario, Admin y Registro
            // ADMIN
            if (user.equals("Admin") && password.equals("Patata19")) {
                admin();
            }
            // USUARIO
            else {
                usuarioBusqueda(user, password);
            }
            System.out.println("#####");
            System.out.print("Desea iniciar nuevamente el programa? SI/NO ");
            confir = leer.next();
            confir = confir.toUpperCase();
        }
    } else {
        System.out.println("");
    }
    leer.close();
}
```

Inicio

Inicio sesion

Admin()

User()

Cerrar Programa.

Menu Admin – admin()

```
// #####//admin//
public static void admin() throws IOException {
    System.out.println("Bienvenido al menu de ADMIN");

    boolean ingresoMenu = true;
    String ingreso;
    String opcion;
    @SuppressWarnings("resource") // Se usa solo para retirar el aviso del Scanner
    Scanner leer = new Scanner(System.in);

    while (ingresoMenu) {
        // 4 opciones disponibles para el admin
        System.out.print("""
            Opciones disponibles:

            A) Eliminar Jugadores
            B) Agregar Enemigos
            C) Agregar Hechizos
            D) Ver las estadísticas de los Jugadores

            Ingrese su opcion ==> """);
        opcion = leer.next();
        opcion = opcion.toUpperCase();
        if (opcion.equals("A")) { // Eliminar Jugadores
            System.out.println("a");
        } else if (opcion.equals("B")) { // Agregar Enemigos OK
            System.out.println(" ");
            agregarEnemigos();
        } else if (opcion.equals("C")) { // Agregar Hechizos OK
            System.out.println(" ");
            agregarHechizos();
        } else if (opcion.equals("D")) { // Ver las estadísticas de los Jugadores OK
            System.out.println(" ");
            System.out.println("Lista de las estadísticas de los jugadores: ");
            estadisticasJugadores();
        }
        System.out.println("Desea salir del menu de admin? SI/NO");
        ingreso = leer.next();
        ingreso = ingreso.toUpperCase();
        if (ingreso.equals("SI")) {
            ingresoMenu = false;
        } else if (ingreso.equals("NO")) {
            ingresoMenu = true;
        } else {
            break;
        }
    }
}
```

Admin() – agregarEnemigos()

```
// //Agregar Enemigos
public static void agregarEnemigos() throws IOException {
    @SuppressWarnings("resource") // Se usa solo para retirar el aviso del Scanner
    Scanner leer = new Scanner(System.in);
    File file = new File(pathname: "Enemigos.txt");
    boolean confirmacion = true;
    String nomEnemigo, hp, ataque, clase, velocidad, opcion;
    while (confirmacion) {
        System.out.print(s: "Ingrese el nombre del enemigo ==> ");
        nomEnemigo = leer.next();

        System.out.print(s: "Ingrese el hp del enemigo ==> ");
        hp = leer.next();

        System.out.print(s: "Ingrese el poder de ataque ==> ");
        ataque = leer.next();

        System.out.print(s: "Ingrese la clase del enemigo del enemigo {S,A,B,C,F} ==> ");
        clase = leer.next();
        clase = clase.toUpperCase();

        System.out.print(s: "Ingrese la velocidad del enemigo ==>");
        velocidad = leer.next();

        BufferedWriter archEnemigo = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(file, append: true)));
        archEnemigo.write(str: "\n");
        archEnemigo.write(nomEnemigo + "," + hp + "," + ataque + "," + clase + "," + velocidad);
        System.out.print(s: "Desea agregar otro enemigo? SI/NO ==> ");
        opcion = leer.next();
        opcion = opcion.toUpperCase();
        if (opcion.equals(anObject: "SI")) {
            confirmacion = true;
            archEnemigo.close();
        } else if (opcion.equals(anObject: "NO")) {
            confirmacion = false;
            archEnemigo.close();
        }
    }
}
```

agregarEnemigos()

Abre el archivo .txt "Enemigos" luego a través de un ciclo while pide por consola al "Admin" que ingrese el nombre y atributos del enemigo

Esta condición solo confirma si repetir o no el ciclo "while"

- Para ingresar a este procedimiento se debe ingresar por consola la letra "B" o b minúscula ya que se agregó la función .toUpperCase() para hacer un pequeño control de "error"

Admin() – agregarHechizos()

```
// #####//SYS.OUT
// // Agregar Hechizos
public static void agregarHechizos() throws IOException {
    @SuppressWarnings("resource") // Se usa solo para retirar el aviso del Scanner
    Scanner leer = new Scanner(System.in);
    File file = new File(pathname: "Hechizos.txt");
    boolean confirmacion = true;
    String hechizo, poderHechizo, opcion;
    while (confirmacion) {
        System.out.print(s: "Ingrese el nombre del hechizo ==>");
        hechizo = leer.next();
        System.out.print(s: "Ingrese el poder del hechizo ==>");
        poderHechizo = leer.next();
        System.out.println("ENTRADA    " + hechizo);
        BufferedWriter archHechizos = new BufferedWriter(new OutputStreamWriter(new FileOutputStream(file, append: true)));
        archHechizos.write(str: "\n");
        archHechizos.write(hechizo + "," + poderHechizo);
        System.out.print(s: "Desea agregar otro hechizo? SI/NO ==> ");
        opcion = leer.next();
        opcion = opcion.toUpperCase();
        if (opcion.equals(anObject: "SI")) {
            confirmacion = true;
            archHechizos.close();
        } else if (opcion.equals(anObject: "NO")) {
            confirmacion = false;
            archHechizos.close();
        }
    }
}
```

Esto escribe en el archivo Hechizos.txt sin errores siempre y cuando existan datos en él, ya que da un salto de línea antes de escribir

Admin() – estadisticasJugadores()

```
// #####//SYS.OUT
// Estadisticas//OK
public static void estadisticasJugadores() throws FileNotFoundException {
    File file = new File(pathname: "Jugadores.txt");
    Scanner arch = new Scanner(file);
    while (arch.hasNext()) {
        String[] partes = arch.next().split(regex: ",");
        String nombreUsuario = partes[0], ptosVida = partes[2], ataque = partes[3], defensa = partes[4],
        velocidad = partes[5], numHechizos = partes[6], exp = partes[7];
        System.out.println("""
#####
Usuario: %s
#####
Puntos de Vida: %s
Ataque: %s
Defensa: %s
Velocidad: %s
Numero de Hechizos: %s
Experiencia: %s""").formatted(nombreUsuario, ptosVida, ataque, defensa, velocidad, numHechizos, exp));
    }
    arch.close();
}
```

Esto muestra una lista de los atributos o estadísticas de los jugadores del archivo "Jugadores.txt"

User()

```
public static void usuarioBusqueda(String usuario, String pass) throws IOException {
    // Si no encontramos al usuario debemos llevarlo a registrarlo sino ya podremos
    // ingresar al menu
    File file = new File(pathname: "Jugadores.txt");
    Scanner arch = new Scanner(file);
    // Este while sera de busqueda
    Boolean encontrado = false;
    while (arch.hasNext()) {
        String[] partes = arch.next().split(regex: ",");
        String user = partes[0], contraseña = partes[1];
        // 4 posibles causas
        // Usuario encontrado y Contraseña coinciden
        if (user.equals(usuario)) {
            System.out.println(x: "");
            System.out.println(x: "Usuario encontrado");
            System.out.println(x: "");
            encontrado = true;
            if (contraseña.equals(pass)) {
                System.out.println(x: "Acceso correcto");
                System.out.println(x: "");
                menuUsuario(user);
                break;
            } else {
                System.out.println(x: "[Contraseña Incorrecta!!]");
                System.out.println(x: "");
                System.out.println(x: "#####");
                break;
            }
        }
        // Usuario encontrado y Contraseña no coincide
    }
    arch.close();
    if (!encontrado) {
        System.out.println(x: "Usuario no encontrado");
        Registro();
    }
}
```

Busca de forma lineal a través del archivo **sin** cargar los datos a una lista para luego buscar en la misma

Una vez encontrado el usuario se revisa si su contraseña coincide con la que esta guardada en la lista

Si coincide la contraseña se nos lleva al menuUsuario() (nos llevamos la variable "user" para usarla más adelante)

Menu Usuario – menuUsuario()

```
// #####//menuUsuario//
public static void menuUsuario(String user) throws FileNotFoundException {
    @SuppressWarnings("resource") // Se usa solo para retirar el aviso del Scanner
    Scanner leer = new Scanner(System.in);
    String opcion;

    System.out.println("Bienvenido " + "[" + user + "]");
    System.out.println(x: "");
    System.out.println(x: "Opciones del menu de usuario: ");
    System.out.print("""
        A)Inicio del juego
        B)Aprender hechizo
        C)Ver estadisticas de un Jugador
        D)Ver estadisticas de Hechizo
        E)Ver ranking de jugadores con mas Exp
        F)Salir

        Ingrese su opcion ==> """);
    opcion = leer.next();
    opcion = opcion.toUpperCase();
    if (opcion.equals(anObject: "A")) {
        System.out.print("""
            Ingrese modo de juego

            1 ==> JcE
            2 ==> JcJ

            Ingrese su opcion ==>
            """);
        int modoJuego = leer.nextInt();
        if(modoSJuego == 1){
            JuegoJCE();
        }else if (modoJuego == 2){
            JuegoJcJ();
        }
    } else if (opcion.equals(anObject: "B")) {
        System.out.println(x: "Aprender hechizo");
        aprenderHechizo(user);
    } else if (opcion.equals(anObject: "C")) {
        System.out.println(x: "Busqueda de estadisticas: ");
        System.out.print(s: "Ingrese el nombre del usuario que busca ==> ");
        String busqUser = leer.next();
        busqeEstadisticas(busqUser);
    } else if (opcion.equals(anObject: "D")) {
        System.out.println(x: "Ingrese el nombre de ususario a buscar ==> ");
        String userHechizo = leer.next();
        hechizoJugador(userHechizo);
    } else if (opcion.equals(anObject: "E")) {
        System.out.println(x: "Ranking de los jugadores con mas exp: ");
        podio(user);
    } else if (opcion.equals(anObject: "F")) {
        System.out.println(x: "Saliendo del menu jugador ... ");
    }
}
```