

# **Proyecto 2ª Evaluación.**



**Por: Deyvid Krasimirov Uzunov**

## Descripción del proyecto.

Para la elaboración del ahorcado he necesitado el uso de 7 clases.

He intentado poner todo donde debería corresponder y ordenarlo de la forma más eficiente.

En la clase **FICHERO** tengo los métodos:

Inicio();

Le pone nombre al fichero.

leeFichero();

En el leo todo lo que hay en el fichero y lo vas añadiendo al Array de listaPalabras.

escriboFichero();

Se encarga al acabar el juego de escribir las palabras nuevas en el fichero.

Ambos capturan los errores que se puedan dar y se encargan de que el fichero se cierre después de acabar.

En la clase **JUGADOR** tengo los métodos:

private String nombre; Es el nombre del jugador.

private int fallos; Numero de fallos del jugador.

private boolean descalificado; Booleano para saber si esta descalificado.

private String secretaGuiones; La palabra secreta pasada a guiones para separar los resultados de cada jugador.

private int aciertos; Variable para guardar el numero de aciertos que a tenido cada jugador.

Constructor de jugador, donde pido nombre y el número de fallos que siempre se pondrá 0.

Y otro vacío para poder crear sin dar parámetros.

toString(); sobrescrito.

Sobrescribo el toString para mostrar los jugadores y sus aciertos.

cuentaFallos();

Suma los fallos que va cometiendo cada jugador.

Comprueba si el jugador tiene 5 fallos y lo descalifica

verificarFinDeJuego();

Comprueba si la palabra del jugador ya es la que hay que adivinar.

Devuelve true si lo es y false si no.

juega();

Es el menú dentro de la ronda

Se muestra el jugador que está jugando y sus fallos.

Se muestra la palabra a adivinar, y se le pide letra llamando a comprueba letra,

Después se vuelve a mostrar la palabra guiones y si tiene acertadas su posición.

Comprueba si el juego a acabado.

palabraSecretaGuiones();

Transforma la palabra secreta en una palabra con guiones de la misma longitud y la devuelve.

Inicio();

Pone los jugadores a 0 y les asigna la palabra secreta.

muestraPalabraGiones();

Muestra la palabra secreta en guiones con espacio para que se pueda apreciar cuantos caracteres tiene.

compruebaLetra();

Pide letra, si la letra está en la palabra, comprueba en que posición esta y la escribe en palabra guiones de ese jugador que esté jugando.

En las posiciones que no esté escribe el guion de la palabra con guiones.

Si la letra no está en la palabra llama al método cuentaFallos() que se encarga de sumarle un fallo al jugador.

Si la letra que introduce es incorrecta te la pide de nuevo.

Tiene todos los Getters y Setters.

En la clase **JUGADORES** tengo los métodos:

```
private ArrayList<Jugador> listaJugadores = new <Jugador> ArrayList();
```

El Array que se encarga de guardar los jugadores.

nuevoJugador();

Método que pide un nuevo jugador y lo añade.

muestraPuntuación();

Muestra el menú de jugadores con sus nombres y los aciertos que han tenido.

Si no hay jugadores no se puede haber jugado así que no se muestra el menú.

Inicio();

Para todos los jugadores del Array, los inicia a 0 y asigna palabrasecreta.

obtenerNumeroJugadores();

Devuelve el número de jugadores.

obtenerJugador();

Obtiene el jugador de la posición que le indiques.

En la clase **MyError** tengo los métodos:

Tengo un extends de Exception y un constructor que se encarga de pasarle un mensaje a su superclase.

En la clase **JUGADORES** tengo los métodos:

static ArrayList<String> listaPalabras; Array en el que se guardan las palabras al cargarlas desde el fichero.

static String palabra; Es la palabra secreta.

Palabras();

Lee del fichero todas las palabras.

nuevaPalabra();

Añade una palabra al array, pidiendo la palabra y dando como letras validas el abecedario.

guardoPalabras();

meto en el fichero el array.

palabraSecreta();

Coge una palabra de listaPalabras y la elige como la palabraSecreta que tienen que adivinar.

inicioTemporal();

Inicio de palabras que se añaden al array.

Tengo los Getters y los Setters.

En la clase **PANTALLA** tengo los métodos:

Solo he añadido el método pideLetra(); y menú();

Muestra el menú y pide un entero para elegir la opción deseada.

Se encarga de pedir una letra y solo una de las letras que le e puesto como letrasValidas.

En la clase **JUEGO** tengo los métodos:

Private static Jugadores jugadores.

Constructor de juego donde le paso un jugador, el lo guarda en la variable común para utilizarla.

compruebaJugadorYPalabra();

Comprueba si tenemos jugadores y devuelve true o false.

jugarRonda();

Inicializamos los jugadores, comprueba si hay jugadores y palabras si no las hay nos dice que no hay jugadores o palabras.

Por el contrario si tiene, empieza el juego.

menuJugador();

El juego a comenzado, este es el método donde discurre el juego.

Mientras no haya ganador, o no hayan fallado todos, Se juega la ronda y hasta que no haya ganador, no acaba.

También comprueba si hay descalificados, si los hay el jugador que lo esté no juega.

En Proyecto2ªEv, la clase que contiene el main();

Nada más empezar, al llamar al constructor, cargo el fichero en el ArrayList llamando al constructor de palabras.

Muestro el menú guardando la opción que eligen en (op).

Y mientras la (op) este entre 1 y 4 le doy diferentes funciones.

1- Añade la palabra nueva.

2- Añade jugador nuevo.

3- Juega una ronda.

4- Muestra las puntuaciones.

Después de que cada una de ellas finalice muestro otra vez el menú.

Cuando introducen un número que no esté entre 1 y 4 se sale del juego, pero antes guarda las nuevas palabras introducidas en el fichero y lo cierra.

## Problemas e inconvenientes.

Muchas veces me quedaba atascado en algún fallo que no sabía cómo resolver y e tenido que recurrir a algún profesor, pero con la explicación del porque entendía y podía pensar en cómo solucionario.

Creo que he tenido mucha ayuda de los profesores y eso en el examen no ocurrirá asique espero estar preparado.

## Conclusiones.

El proyecto ha servido mucho para repasar todo lo visto este trimestre y creo que ha ayudado en parte para la preparación el examen.

Creo que con este proyecto he aprendido cosas en las que no estaba tan bien preparado y espero que en el examen me vaya todo igual de bien.

He dedicado dos horas cada día, mas las 3 horas de programación en clase.

Solía realizar algún método en casa y al quedarme atascado en un fallo tener que recurrir a la profesora para resolver la duda y así poder continuar.

Podía haberme esforzado mas y así será en los siguientes.

