MEMORIA EXPLICATIVA AHORCADO

IGNACIO MARÍA GALLEGO TORRES

Ingeniería Informática + ADE

**SUBPROGRAMAS y UTILIDAD**

**PROCEDURE** OrdenarJugadores(VAR list: tJugadores);

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de jugadores ordena estos alfabéticamente según su nombre de usuario

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** ListadoJugadores(list: tJugadores);

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de jugadores muestra el nombre del usuario, su nombre real y su puntuación

---------------------------------------------------------------------------------------------------------------------------------

**FUNCTION** ExisteJugador (player: string; list: tJugadores):boolean;

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de jugadores y un nombre de usuario, devuelve TRUE si pertenece a la lista, FALSE en caso contrario

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** DarAltaJugador(player: string; VAR list: tJugadores);

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de jugadores y un nombre de usuario, añade el jugador a la lista si la lista de jugadores no está llena.

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** DarBajaJugador(player: string; VAR list: tJugadores);

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de jugadores y un nombre de usuario, elimina al jugador no sin antes comprobar si este pertenece o no

---------------------------------------------------------------------------------------------------------------------------------

**FUNCTION** ExistePalabra(wordle: tPalabra; list:tPalabras):boolean;

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de palabras y una palabra, devuelve TRUE en caso de que la palabra pertenezca y FALSE en caso contrario

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** AltaPalabra(wordle: tPalabra; VAR list: tPalabras);

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de palabras y una palabra, añade la palabra a la lista, comprobando que la palabra no exista y que tenga 10 letras

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** BajaPalabra(wordle: tPalabra; VAR list: tPalabras);

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de palabras y una palabra, elimina dicha palabra de la lista en el caso de que dicha palabra pertenezca a esta

------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** MostrarListaPalabras(listWords: tPalabras);

------------------------------------------------------------------------------------------------------------------------------

Dada una lista de palabras, las muestra por pantalla.

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** ModificarPalabra(wordle: tPalabra; VAR list: tPalabras);

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de palabras y una palabra, primeramente comprueba si la palabra existe, y en caso afirmativo pregunta la nueva palabra, comprobando si esta nueva palabra no pertenece y si tiene 10 letras

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** HacerBackup(VAR filePlayers:tArchivoJugadores; listPlayers: tJugadores; VAR fileGames: tArchivoPartidas; listGames: tPartidas; VAR fileWords: text; listWords: tPalabras );

---------------------------------------------------------------------------------------------------------------------------------

Dados archivos de jugadores, palabras y partidas y listados de jugadores, palabras y partidas, copia los datos de los listados en sus respectivos archivos

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** RestaurarBackup(VAR filePlayers:tArchivoJugadores; VAR listPlayers: tJugadores; VAR fileGames: tArchivoPartidas; VAR listGames: tPartidas; VAR fileWords: text; VAR listWords: tPalabras);

---------------------------------------------------------------------------------------------------------------------------------

Dados archivos de jugadores, palabras y partidas y listados de jugadores, palabras y partidas, reescribe los datos de dichos archivos en sus correspondientes listados

---------------------------------------------------------------------------------------------------------------------------------

**FUNCTION** PerteneceLetra(letter:char; wordle:tPalabra):boolean;

---------------------------------------------------------------------------------------------------------------------------------

Dada una palabra y una letra, comprueba si dicha letra pertenece (TRUE) o no (FALSE) a dicha palabra

---------------------------------------------------------------------------------------------------------------------------------

**FUNCTION** Puntuacion(failsNum:integer):integer;

---------------------------------------------------------------------------------------------------------------------------------

Dado un número de fallos, devuelve los puntos correspondientes a este número

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** Jugar(listWord: tPalabras; VAR player: tJugador; VAR listGame: tPartidas);

---------------------------------------------------------------------------------------------------------------------------------

Dado una lista de palabras, de partidas y un jugador, este generara una palabra aleatoria de dicha lista de palabras y el jugador tendrá un máximo de 13 intentos o 3 fallos para adivinar dicha palabra.

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** VerRanking(list:tJugadores);

---------------------------------------------------------------------------------------------------------------------------------

Dada una lista de jugadores, ordena esta de mayor a menor según la puntuación de estos y posteriormente los muestra por pantalla; su puesto, su usuario y puntuación

---------------------------------------------------------------------------------------------------------------------------------

**FUNCTION** HaJugado(user:string; listGames: tPartidas):boolean;

---------------------------------------------------------------------------------------------------------------------------------

Dado una lista de partidas y un nombre de usuario, devuelve TRUE si dicho usuario a jugado, FALSE en caso contrario

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** VerDetallePartidas(user:string; listGames: tPartidas);

---------------------------------------------------------------------------------------------------------------------------------

Dado una lista de partidas y un nombre de usuario, muestra sus partidas jugadas (en caso de que haya jugado), dando la información acerca del progreso del jugador en cada uno de los intentos

---------------------------------------------------------------------------------------------------------------------------------

**PROCEDURE** DarBajaPartidas(user:string; VAR listGames: tPartidas);

------------------------------------------------------------------------------------------------------------------------------

Dado un nombre de usuario y una lista de partidas, elimina todas las partidas de dicho usuario.

---------------------------------------------------------------------------------------------------------------------------------