

Analista Universitario en Sistemas - 2022

TALLER DE PROGRAMACIÓN I

TRABAJO FINAL
WORDLE

ALUMNO: Ezequiel Spolli

Comisión 3



INDICE

OBJETIVO:	Pág. 3
ELEMENTOS DE C:	Pág. 3
ESTRUCTURAS DE DATOS:	Pág. 4
▪ ARRAYS MULTIDIMENSIONALES O MATRICES.....	Pág. 4
FUNCIONAMIENTO DEL PROGRAMA:	Pág. 5
▪ MAIN.....	Pág. 5
▪ FUNCIONES.....	Pág. 5
EJECUCIÓN:	Pág. 7
MEJORAS A FUTURO:	Pág. 9



OBJETIVO:



El objetivo del ejercicio es programar en el lenguaje de programación C el juego conocido como "WORDLE".

Wordle es un juego donde el objetivo es adivinar una palabra de 5 (cinco) letras, ubicadas en bloques, para lo cual solo se tiene 6(seis) intentos.

Esta misteriosa palabra es conocida como 'WORDLE'.

Cada vez que se hace un intento, es decir se escribe una palabra completa para ver si coincide con la palabra misteriosa, los bloques cambian de color para indicarle al jugador, a modo de pistas, qué tan cerca está de adivinar dicha palabra y ganar.

Si el bloque se torna de color:

- **VERDE**: indica que esa letra forma parte de la palabra y se encuentra en el bloque correcto
- **AMARILLO**: indica que la letra forma parte de la palabra, pero no se encuentra en el bloque correcto
- **GRIS**: la letra no forma parte de la palabra misteriosa.

Al inicio del juego todo se presenta en blanco.

ELEMENTOS DE C:

En este ejercicio se emplean los siguientes elementos de programación C:

- Directivas de preprocesador: Macros
- Estructuras iterativas – Bucles (For, While, Do While)
- Estructuras condicionales (If - else)
- Generación de números aleatorios
- Variables compuestas – Arrays multidimensionales (Matrices)
- Funciones - Recursividad
- Persistencia de datos - Archivos

El programa consiste en dos archivos

- **palabras.txt** - Archivo de palabras de 5 letras
- **wordle.c** – Programa en C

Total de líneas del programa: 365 líneas



ESTRUCTURAS DE DATOS:

ARRAYS MULTIDIMENSIONALES O MATRICES

El programa cuenta de 8 Arrays Multidimensionales o Matrices donde se irá guardando la información de cada una de las partidas.

M [8][13][5]						Matriz por partida		Durante toda el sesión			
						0	1	2	3	4	
0											0 Archivada
1											1 Intento
2											Resultado
3											2 Intento
4											Resultado
5											3 Intento
6											Resultado
7											4 Intento
8											Resultado
9											5 Intento
10											Resultado
11											6 Intento
12											Resultado

A palabra Archivada (Posicion 0)
 I palabra Ingresada (Impares)
 R Resultado (Pares)

Cada una de las 8 matrices tendrá una extensión de M [Partidas][Filas][Columnas]:

- **Partidas** = 8: una por cada partida.
- **Filas** = 13:
 - En la fila 0 se almacena la palabra secreta o "WORDLE"
 - En las filas impares se almacena cada uno de los intentos ingresados (hasta 6 intentos)
 - En las filas pares (a partir del 2) se almacena cada uno de los resultados obtenidos al comparar nuestra palabra ingresada con la palabra secreta o "WORDLE" (hasta 6 resultados).

Este resultado se mostrará de la siguiente manera:

Resultado:	
_	La letra no esta
*	La letra esta en el lugar INCORRECTO
A	La letra "A" esta en el lugar CORRECTO

- **Columnas** = 5: Cantidad de letras de las palabras a adivinar

El programa también cuenta con 2 arrays de extensión [8] (Máxima cantidad de partidas por sesión de juegos) para almacenar información de cada partida:

- **Try_count** [8]: Almacena la cantidad de intentos ejecutados en cada partida.
- **Score** [8]: Se Almacena el puntaje obtenido en cada partida.

FUNCIONAMIENTO DEL PROGRAMA:

MAIN

- Define variables y parámetros a utilizar
- Da la bienvenida al juego
- Inicializa variables y parámetros
- Solicita a través de la función **ask_for_games** la cantidad de partidas a jugar.
- A través de la función **get_saved_words** almacena la lista de palabras secretas que se utilizarán a lo largo de la sesión de juegos.
- Muestra el número de partida actual
- A través de la función **play_match** juega la partida hasta que termine (ya sea por adivinar la palabra o por agotar la cantidad de intentos posibles).
- Si luego de finalizar la partida el jugador desea seguir jugando la siguiente partida, el programa vuelve a mostrar el número de la partida actual y repite el ciclo hasta que termine la cantidad de partidas de la sesión de juegos o el jugador quiera salir.
- Una vez terminada la sesión de juegos, se muestra el recorrido por el juego a través de los puntajes obtenidos en cada partida y se muestran estadísticas sobre los mismos. Se utiliza la función **print_results_statistics**.
- Luego, el programa consulta al jugador si desea ver alguna partida en especial y la muestra en pantalla a través de la función **see_match**.
- Por último consulta al jugador si desea iniciar una nueva sesión o dejar de jugar. Se utiliza la función **exit_game**. Si el jugador desea seguir jugando, el juego vuelve a empezar pidiendo nuevamente la cantidad de partidas que desea jugar.

FUNCIONES

- **initialize_array** : Inicializa un array dado en "0"
- **ask_for_games** : Consulta la cantidad de partidas a jugar y chequea que el valor ingresado este entre 1 y el máximo de partidas posible (8).
- **get_saved_words**:
 - Carga un array "**Word_line [MAX_GAMES]**" de extensión "8" (Máximo de partidas) con números aleatorios **no repetidos**. Estos números representan el número de línea donde se encuentran las palabras en el archivo **palabras.txt**. De esta manera, las palabras no se repetirán en una misma sesión de juego.
 - Llama a la función **getWordInLine** y a través de los números almacenados en "**Word_line [MAX_GAMES]**" carga cada una de las matrices en la fila cero con las palabras obtenidas.
- **getWordInLine**: A través de un número, busca en el archivo **palabras.txt** la palabra situada en esa línea, para luego almacenarla en un espacio de memoria dado.



- **play_match:** Básicamente ejecuta la partida.
 - Inicializa parámetros
 - A través de la función **ask_for_try_word** solicita el ingreso de una palabra.
 - A través de la función **compare_try_word** compara la palabra ingresada con la palabra secreta o “WORDLE”, almacena el resultado del intento realizado y va sumando los puntajes correspondientes a nuevas letras en lugares correctos o incorrectos.
 - A través de la función **print_line** muestra el mapa de la jugada realizada, es decir, la palabra ingresada y su resultado al compararla.
 - Muestra la cantidad de intento.
 - Chequea si se adivino la palabra o si se terminó la cantidad de intentos posibles. De no ser así vuelve a pedir una nueva palabra a través de **ask_for_try_word** y vuelve a repetirse el ciclo.
 - Si se adivinó la palabra o si se terminó la cantidad de intentos a través de la función **score_win** asigna los puntajes correspondientes al final del juego.
 - Muestra el número de partida, el puntaje, la cantidad de intentos, la palabra escondida o “WORDLE” y luego, muestra el mapa de la partida a través de la función **print_match**.

- **ask_for_try_word:** Solicita el ingreso de una palabra a comparar y chequea a través de la función **check_word** su correcto ingreso.

- **check word:** Chequea que la palabra dada sea de una longitud igual a 5 letras y que no repita letras.

- **compare_try_word;** Compara la palabra dada con la palabra secreta o “WORDLE” letra a letra.
 - Si la letra de la palabra dada **no está** en la palabra secreta o “wordle” asigna que no está “_” en el resultado del intento.
 - Si la letra de la palabra dada **está** en la palabra secreta o “wordle” entonces:
 - Si **esta en el lugar correcto** asigna que está en el lugar correcto Ej.: “A” en el resultado del intento.
 - Si **es una letra nueva o letra ya ingresada pero en el lugar incorrecto** suma al puntaje el valor correspondiente a una nueva letra en el lugar correcto.
 - Si **esta en el lugar incorrecto** asigna que está en el lugar incorrecto “*”
 - Si **es una letra nueva**, suma al puntaje el valor correspondiente a una nueva letra en el lugar incorrecto.

- **check_win:** Compara letra a letra la palabra ingresada con la palabra secreta y analiza si se adivinó la palabra secreta o “wordle” en el intento.

- **score_win:** Asigna puntajes correspondientes al final del juego, ya sea por adivinar la palabra o por agotar la cantidad de intentos



- **print_results_statistics:** Es llamada al final de la sesión de juego (ya sea por haber jugado la cantidad de partidas ingresadas o por salir antes) y muestra los resultados obtenidos en cada partida jugada y llama a la función **eststatistics** para calcular y mostrar estadísticas de la sesión de juegos.
- **statistics:** Calcula y muestra el promedio de los puntajes donde se obtuvieron victorias, y el puntaje más alto y más bajo y en que partida fueron obtenidos.
- **see_match:** Consulta si se quiere visualizar el mapa de alguna partida en especial. Y a través de las funciones **print_line** y **print_match** la imprime en la pantalla.
- **print_line:** Muestra en pantalla el contenido de una fila dada de una matriz específica
- **print_match:** Muestra en pantalla el contenido de una matriz específica, por lo que En este caso, muestra el mapa de una partida.
- **exit_game:** Consulta si el jugador desea seguir jugando y chequea si el valor ingresado es correcto.

EJECUCIÓN:

```

Bienvenide a WORDLE!!!
Por favor, ingresa el número de partidas que deseas jugar (Máximo 8 partidas)
> 3

Por favor, ingresa las palabras en LETRAS MAYÚSCULAS
Partida nro 1 de 3:

Por favor, ingresa una palabra de 5 letras (NO repetir letras)
> LETRA

L E T R A
L E T R A

Intento: 1

Partida nro: 1 de 3 - Palabra secreta: L E T R A

L E T R A

L E T R A
L E T R A

Puntaje: 10000
Intentos: 1

Deseas seguir jugando?
1.SI
2.NO

```

- El programa le da la bienvenida al jugador, le solicita el número de partidas que desea jugar en este caso **3 partidas** y luego le pide el ingreso de la primera palabra a probar. En este caso el jugador acierta en el primer intento, llevándose el puntaje mayor de 10000 puntos.


```

Partida nro 2 de 3:

Por favor, ingresa una palabra de 5 letras (NO repetir letras)
> COPAS

C O P A S
_ * _ _ *

Intento: 1

Por favor, ingresa una palabra de 5 letras (NO repetir letras)
> SUELO

S U E L O
S U E L O

Intento: 2

Partida nro: 2 de 3 - Palabra secreta: S U E L O

S U E L O

C O P A S
_ * _ _ *
S U E L O
S U E L O

Puntaje: 7100
Intentos: 2

Deseas seguir jugando?
    1.SI
    2.NO

> 1

Partida nro 3 de 3:

Por favor, ingresa una palabra de 5 letras (NO repetir letras)
> AUDIO

```

- El jugador continua con la próxima partida y esta vez acierta la palabra en el segundo intento con un puntaje de 7100 puntos.

```

A U D I O
* _ _ _ *
C O B R A
C O _ _ *
C O P A S
C O P A S

Puntaje: 6750
Intentos: 3

Tu recorrido por el juego fue:

Partida nro 1 - Puntaje: 10000 - Palabra secreta: L E T R A
Partida nro 2 - Puntaje: 7100 - Palabra secreta: S U E L O
Partida nro 3 - Puntaje: 6750 - Palabra secreta: C O P A S

El puntaje más alto fue: 10000 puntos en:
Partida nro: 1

El puntaje más bajo fue: 6750 puntos en:
Partida nro: 3

El promedio de todas tus victorias fue: 7950

Deseas ver alguna partida?
    1.SI
    2.NO

> █

```


- En la tercera partida, el jugador descubre la palabra secreta o “wordle” en el tercer intento y como es la última partida de la cantidad ingresada, la sesión de juego termina dando los resultados y palabras secretas de cada partida jugada. A continuación el juego también muestra estadísticas sobre las partidas en las que se obtuvo el puntaje más alto y más bajo, y el promedio de los puntajes donde obtuvo una victoria.

```

El puntaje más bajo fue: 6750 puntos en:
Partida nro: 3

El promedio de todas tus victorias fue: 7950

Deseas ver alguna partida?
    1.SI
    2.NO

> 1

Que numero de partida deseas ver?

> 2

Partida nro 2 de 3 - Palabra secreta: S U E L O

S U E L O
C O P A S
 *   *
S U E L O
S U E L O

Puntaje: 7100
Intentos: 2

Deseas ver alguna partida?
    1.SI
    2.NO

> █

```

- Por último, el jugador desea ver el mapa de la partida jugada numero 2. Luego de mostrar la partida el programa consulta si quiere ver alguna otra partida.

MEJORAS A FUTURO:

- Analizar que las palabras en el archivo **palabras.txt** usadas como palabras secretas o “wordle” sean de correcta longitud y que no presenten repetición entre sus letras.
- Que el programa no discrimine entre letras mayúsculas y minúsculas o entre letras con y sin tildes al ingresar la palabra a comparar.
- Dar la opción al jugador de elegir la longitud de la palabra secreta a adivinar o “wordle”.
- Incluir el factor tiempo de resolución y reflejarlo en el puntaje obtenido.

