

## TRABAJO FINAL

### Comentarios iniciales

El trabajo debe ser entregado al menos 2 días (hábiles) previos a la mesa examinadora.

El trabajo puede realizarse en equipos de hasta máximo 2(dos) programadores, sin excepción.

Ambos participantes del equipo deben conocer claramente la totalidad del código y su propósito, poder explicarlo y saber como modificar el mismo con fines correctivos o para agregarle funcionalidades.

Cabe mencionar que el trabajo práctico requiere más esfuerzo dependiendo de la cantidad de programadores que trabajen en equipo, será requerido resolver un mayor número de ítems.

Se exige que el código sea acompañado con la documentación solicitada, sólo de esa manera será evaluado el trabajo práctico en su totalidad.

## WORDLE



El **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'.

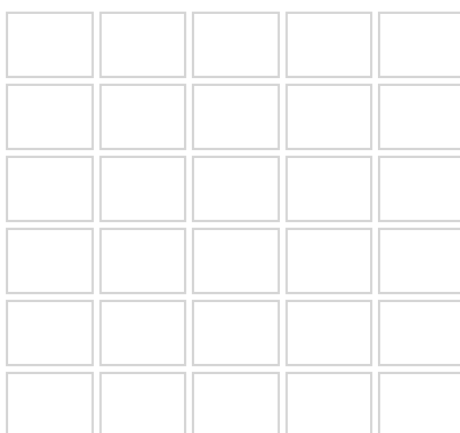
Cada vez que se hace un intento, es decir se escribe una palabra completa para ver si coincide con la palabra misteriosa, los bloques cambian de color para indicarle al jugador, a modo de pistas, qué tan cerca está de adivinar dicha palabra y ganar.

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.

## WORDLE (ES)



Ejemplo de una jugada.

T	I	M	O	N
F	I	J	A	R
L	I	C	E	O
S	I	G	L	O
P	I	D	I	O
D	I	S	C	O

El jugador ingresa **TIMON** → acierta que **I** está en la palabra y en la ubicación correcta, **O** también está, pero en otra posición.

Ingresa **FIJAR** → No es una buena estrategia ignorar que la **O** pertenece a la palabra... pero sigue descartando letras (F, J, A, R)

Ingresa **LICEO** → ahora acierta la posición de la letra **O** y detecta que la letra **C** pertenece a la palabra.

Ingresa **SIGLO**, descubre que la letra **S** pertenece a la palabra.

Ingresa **PIDIO**, descubre que la letra **D** pertenece a la palabra.

Ingresa **DISCO** y *acierta* la palabra.

Como resultado, recibirá una puntuación (la definiremos más adelante) y un *diagrama o mapa* que muestra la “trayectoria o progreso” desde el inicio del juego, hasta la victoria o el fracaso.

En este caso dicho **diagrama o mapa** sería el siguiente:


Pueden repetirse letras dentro de una misma palabra. Por ejemplo, la palabra: **ABEJA**

**Pero para simplificar vamos a evitar estas palabras en este trabajo final, solo consideraremos palabras con todas las letras distintas.**

Cabe mencionar que se ignoran los acentos y diéresis (Á, Ó, Ü), solo se toman en cuenta las letras.

## Puntaje:

- El usuario inicia con 5.000 puntos.
- Si acierta la palabra en el primer intento, gana la jugada con una puntuación de 10.000.
- A medida que realice intentos sin lograr descubrir la palabra, es decir use otra fila, se le descuenta 500.
- Además, en cada intento suman 50 puntos las **nuevas** letras acertadas (en lugar incorrecto) y 100 las **nuevas** letras ubicadas correctamente.
- Finalmente al ganar recibe 2.000 puntos adicionales.  
Si no logra descubrir la palabra la puntuación final es 0.

Puntaje para el caso anterior.

Inicia con **5.000** puntos.

1		100		50		No ganó aún. Se descuenta <b>500</b> por pasar de fila y se le suma <b>100</b> por la letra acertada en posición correcta y <b>50</b> por la letra acertada en posición incorrecta. Puntaje: <b>5000</b> - <b>500</b> + 100 + 50 = 4650
2		0				No ganó aún. Se descuentan 500 por pasar de fila. No acertó letras. Puntaje: 4650 - <b>500</b> = 4150
3		0	50		100	No ganó aún. Se descuentan <b>500</b> . Se le suma <b>50</b> por la <i>nueva</i> letra acertada en posición incorrecta y <b>100</b> por <i>otra</i> correcta. Puntaje: 4150 - <b>500</b> + 150 = 3500
4	50	0			0	No ganó aún. Se descuentan <b>500</b> . Se le suma <b>50</b> por la <i>nueva</i> letra acertada en posición correcta. Puntaje: 3500 - <b>500</b> + 50 = 3050
5		0	50		0	No ganó aún. Se descuentan <b>500</b> . Se le suma <b>50</b> por la <i>nueva</i> letra acertada en posición incorrecta. Puntaje: 3050 - <b>500</b> + 50 = 2600
6						<b>Ganó! Adivinó la palabra misteriosa.</b> Se suman <b>2000</b> puntos a lo acumulado. Puntaje final: 2600 + 2000 = 4600.

*Nota:*

*Tener particular cuidado con lo de letras acertadas en posición incorrecta, que no sume siempre una letra mal ubicada más de una vez.*

## Palabras:

Las posibles palabras misteriosas para las diferentes partidas o jugadas deben:

- pertenecer al idioma español
- contar con 5(cinco) letras
- no tener letras repetidas (para simplificar)
- puede tratarse de un verbo conjugado (ejemplo: *comia* - ignoramos el acento)
- ser tomadas de un archivo a modo de base de datos de palabras
- no repetirse en una misma *sesión de juego*

## Sesión de Juego:

El jugador elegirá participar de una sesión de juego, donde se le solicitará indicar cuantas partidas o jugadas quieren realizarse, con un máximo de 8(ocho).

Deberá ofrecerse también la posibilidad de finalizar el juego luego de que cada partida finalice.

Durante el juego, y antes de iniciar una partida debe indicarse el número de partida y cuantas se jugarán en total. Por ejemplo indicar - Partida Nro 2 de 6.

Al finalizar la sesión de juego se deberá:

(En el caso de que sea un solo programador)

1. Indicar las palabras empleadas en cada partida con los puntajes obtenidos
2. Señalar en cuál o cuáles partidas se obtuvo el puntaje más alto y el más bajo
3. Señalar el promedio de los puntajes en que logró una victoria

(En el caso de que sean dos programadores)

1. Realizar los tres ítems anteriores.
2. Permitir que el jugador visualice el mapa o diagrama de la jugada que desee.
3. Analizar y mostrar de qué manera incluiría el factor tiempo de resolución, mientras más segundos lleve dar con la palabra misteriosa, más puntos serán descontados.

## Recomendaciones finales

- Analizar el enunciado atentamente y prestar atención a los detalles, requerimientos y valores
- Definir el conjunto de variables y funciones necesarias
- Diseñar el algoritmo antes de programar, verificando que los requerimientos puedan satisfacerse
- Comenzar con la documentación en etapas tempranas, no dejarlo para último momento, recordar que es un requerimiento la entrega de la misma.
- Escribir código claro y comentar pertinentemente
- Considerar el uso de macros y funciones, discernir cuándo es conveniente cada herramienta.
- Realizar pruebas parciales del programa para depurarlo
- Exponer a diferentes pruebas al programa final y revisar que no se reciban errores ni warnings durante la compilación
- Asegurarse que al momento de enviar el programa, se trate de su versión final
- Al trabajar en equipo, comunicarse activamente y no desentenderse de los avances del resto de los integrantes del equipo