



# Proyecto 1 Abril – Julio 2022

## Wordle para dos

### 1 Introducción

“Wordle” es una versión en línea de “Toros y Vacas” creada por el ingeniero de software galés Josh Wardle en octubre 2021. El juego se volvió viral a finales de diciembre de ese año debido a su naturaleza diaria (Wardle sólo permite jugar una vez por día) y su mecanismo de compartición en las redes (se puede compartir la matriz de toros y vacas sin compartir la respuesta). En pocas semanas de su viralidad, el ingeniero electrónico colombiano Daniel Rodríguez hizo una versión para español, la cual ganó notoriedad al agregar un “modo tildes” en marzo de 2022.

Este proyecto consiste en hacer una versión de Wordle más cercana a sus orígenes: en vez de sólo permitir adivinar una palabra elegida por la computadora, también debe poder adivinar una palabra elegida por el jugador.

#### 1.1 Cómo se juega

Toros y Vacas se juega con dos jugadores: el primero, llamado la “mente maestra” piensa en un código e indica a su contrincante el número de caracteres. El segundo carácter, llamado el “descifrador”, debe adivinar cuál es el código.

Cada ronda comienza con un turno del descifrador, el cual escribe un código. La mente maestra, en su turno, debe indicar cuántos caracteres están en la posición correcta (los toros) y cuántos, a pesar de formar parte del código, están en la posición incorrecta (las vacas).

Si el descifrador adivina el código, gana.

Wordle agrega algunas reglas adicionales:

- Cada código debe ser una palabra válida de 5 letras.
- Se debe indicar cuáles letras son los toros y cuáles son las vacas.
- Si una letra está repetida en el código, y la adivinación del descifrador contiene esa letra repetida pero ambas están en posiciones incorrectas, ambas son marcadas como vacas.
- Si una letra está repetida en la adivinación pero no está repetida en el código, y ninguna de ellas está en la posición correcta, la primera es marcada como vaca y la segunda no se marca. Así, el descifrador puede ver si hay letras repetidas en el código.
- Se asigna un punto por cada adivinación y menos puntos es mejor.
- Luego de 6 adivinaciones, el descifrador pierde.

## 1.2 Juegos de ejemplo

DESCIFRADOR : ESTAR  
MENTEMAESTRA: -VV--

DESCIFRADOR : COSTO  
MENTEMAESTRA: --TTT

DESCIFRADOR : PUDIN  
MENTEMAESTRA: ---V-

DESCIFRADOR : LISTO  
MENTEMAESTRA: TTTTT

¡Ganaste!

Comparte tu resultado:

-VV--

--TTT

---V-

TTTTT

DESCIFRADOR : SUFRE  
MENTEMAESTRA: ---VV

DESCIFRADOR : RETOS  
MENTEMAESTRA: VT---

DESCIFRADOR : MECER  
MENTEMAESTRA: -T--V

DESCIFRADOR : CERDO  
MENTEMAESTRA: -TTT-

DESCIFRADOR : LERDA  
MENTEMAESTRA: -TTT-

DESCIFRADOR : VERDE  
MENTEMAESTRA: -TTT-

La palabra era "PERDI"

## 2 Requerimientos del programa

### 2.1 Modo Mente Maestra

En el modo Mente Maestra, la computadora elige una palabra de la lista de palabras al azar, y presenta un *prompt* al usuario, para indicarle que ya puede introducir su primera adivinación. La lista de palabras se debe cargar desde el archivo proporcionado.

Luego, en cada turno hasta un máximo de 6, el usuario introduce su palabra. La computadora debe validar que la palabra tenga 5 letras y exista en la lista de palabras. Si la palabra no es válida, debe imprimir un error, y el usuario debe introducir una nueva palabra para ese turno. La computadora también debe convertir entre mayúsculas y minúsculas. Si la palabra elegida tiene acentos, los acentos deben ser ignorados. Si tiene eñes, se considera no válida.

La palabra se compara contra la elegida y se imprime el string de calificación:

- T (por "toro") si la letra correspondiente de la adivinación se encuentra en la posición correspondiente en el código
- V (por "vaca") si la letra correspondiente se encuentra en el código en otra posición
- - (guión) si la letra correspondiente no se encuentra en el código

Si la palabra coincide exactamente, la computadora debe indicarle al usuario que ganó y salir. Si el usuario no adivina luego de 6 turnos, se le indica cual era la palabra y se sale.

Para facilidad de lectura y escritura, el string de calificación debe estar alineado verticalmente con el de la adivinación

### 2.2 Modo Descifrador

En el modo descifrador, la computadora elige una palabra al azar y el usuario debe indicar cuáles son los toros y cuáles son las vacas. La computadora debe usar una estrategia de árbol de juego minimax explicada a continuación.

### 2.2.1 Minimizing

A cada palabra se le asigna una calificación dada por las letras que la componen. Esta calificación está inspirada en los puntos de Scrabble:

- 0,1 puntos: A E
- 0,2 puntos: I N O R S
- 0,3 puntos: D L C T U
- 0,5 puntos: B G M P
- 0,8 puntos: F H Q V Y
- 1 punto: J K W X Z

La computadora entonces elige las 10 palabras con la menor puntuación (minimizar) para evaluar todas las respuestas posibles del usuario: 0 toros y 0 vacas, primera letra toro, primera letra vaca, primeras dos letras toro, etc.

Cada una de estas posibilidades se revisa si es válida (es decir, si existen palabras que puedan causar esta respuesta). A cada posibilidad se le asigna 1 punto, y se restan 0,2 puntos por cada toro y 0,1 puntos por cada vaca (naturalmente, la posibilidad de que sea la respuesta correcta valdrá 0 puntos). Se eligen las 10 posibilidades con la mayor puntuación (maximizar) para evaluar la siguiente palabra a adivinar. Note que pueden haber menos de 10 posibilidades válidas (si todas las palabras válidas resultan en la misma respuesta, por ejemplo --TTT).

Se repiten ambos pasos una vez más, creando un árbol 10-ario de profundidad 4. A cada nodo, se le suman los puntos de sus nodos hijo. El nodo de primer nivel con la puntuación menor será la palabra que adivinará la computadora. Obsérvese que si hay un camino que lleve a una sola palabra posible, este será el preferido. Obsérvese también que la computadora sólo analiza los peores casos para cada una de sus adivinaciones.

### 2.2.2 Respuesta del usuario

Una vez se elige y presenta la adivinación al usuario, el usuario introduce su string de calificación. El string de calificación debe ser validado (no puede tener caracteres que no sean T, V ni guión). Esto se repite hasta que el usuario indique el string "TTTTT", en cuyo momento el programa debe imprimir un mensaje de despedida y salir. Si luego de 6 rondas, el usuario no le ha indicado a la computadora que adivinó la palabra, el programa debe indicarle al usuario que ganó y salir.

Si en cualquier momento no quedan palabras válidas posibles, la computadora debe imprimir "tramposo" y salir.

## 3 Requerimientos de la entrega

El programa debe poderse compilar con `hmake` y llamarse desde la línea de comandos como:

```
$ wordle <modo>
```

Donde `<modo>` es, o bien "mentemaestra", o bien "descifrador".

Todos sus códigos deben estar debidamente documentados en estilo Haddock.

## 4 Requerimientos del Informe

Debe entregar un informe que contenga

- Introducción, resumiendo y dando la estructura del informe
- Decisiones de diseño, incluyendo en cuántos módulos decidieron dividir el código y cuáles características de Haskell decidieron explotar y porqué
- Dificultades en la implementación, incluyendo aspectos inesperadamente rápidos o lentos (tanto desde el punto de vista de programación como desde el punto de vista de ejecución)
- Cualquier optimización que hayan decidido incorporarle al programa
- Conclusiones y recomendaciones para quienes pudieran realizar este proyecto en el futuro

## 5 Evaluación

El proyecto tiene una ponderación de 30 puntos. Se asignarán

- 15 puntos por código
  - 5 puntos por las características generales del programa
    - 1 punto por su forma de leer el archivo
    - 1 punto por su programa principal
    - 1 punto por su forma de controlar el número de rondas
    - 1 puntos por sus estructuras de datos
    - 1 punto por la documentación de sus estructuras de datos
  - 5 puntos por los módulos relacionados con el modo Mente Maestra
    - 1 punto por convertir correctamente entre mayúsculas y minúsculas
    - 1 punto por su forma de buscar en la lista de palabras
    - 1 punto por su forma de construir el string de calificación
    - 1 punto por su forma de detectar si el usuario adivinó
    - 1 punto por la documentación de estas funciones
  - 5 puntos por los módulos relacionados con el modo Descifrador
    - 1 punto por su forma de leer el string de calificación
    - 1 punto por su forma de elegir las 10 palabras a evaluar en los nivel impares del árbol minimax
    - 1 punto por su forma de validar las posibilidades de respuesta del usuario para los niveles pares del árbol minimax
    - 1 punto la forma de totalizar el árbol minimax y escger la adivinación
    - 1 punto por la documentación de estas funciones
- 10 puntos por ejecución
  - 1 punto por la legibilidad del programa
  - 4 puntos por el modo Mente Maestra
    - 1 punto por indicar los toros y vacas consistentemente
    - 1 punto por indicar correctamente cuando una palabra no es válida sin que el usuario pierda el turno
    - 1 punto por indicar correctamente cuando el usuario ha ganado
    - 1 punto por indicar correctamente cuando el usuario ha perdido

- 5 puntos por el modo Descifrador
  - 1 punto por detectar correctamente strings de calificación inválidos
  - 1 punto por qué tan adecuadas son las adivinaciones de la computadora
  - 1 punto por detectar correctamente que ganó
  - 1 punto por detectar correctamente que perdió
  - 1 punto por detectar correctamente al “tramposo”
- 5 puntos por su informe
  - 0,5 puntos por su introducción
  - 2 puntos por sus decisiones de diseño (y optimizaciones decididas en la etapa de diseño)
  - 1,5 puntos por sus dificultades de implementación (y optimizaciones motivadas por las mismas)
  - 1 por sus conclusiones y recomendaciones

Se evaluará cada optimización agregada para un punto adicional, siempre que sea adecuada al problema y esté bien explicada en el informe.

No es necesario imprimir la matriz de toros y vacas para compartir en las redes sociales.