

Tarea Corta 1:

Wordle

Motivación

JavaScript es un lenguaje de programación ideado principalmente para agregar funcionalidades a páginas Web, pero que con el paso de los años y gracias al crecimiento del desarrollo de internet, se ha convertido en la principal opción a la hora de programar aplicaciones web. Con la consecuencia de que el dominio de esta tecnología no solo es deseable, sino imprescindible. Las aplicaciones web se ejecutan en entornos aislados dentro del navegador, constituyéndose en verdaderas “máquinas virtuales”. Dado que una aplicación web puede ser ejecutada por cualquier usuario en cualquier equipo de cómputo, la eficiencia en el código es imperativo.

Es por esta razón que en el contexto de Análisis de Algoritmos tenemos este objetivo para esta primera tarea programada:

- Fortalecer en el estudiante sus competencias en el desarrollo de aplicaciones web, con código eficiente en entornos variables¹.

¹ Aunque la aplicación se desarrolle en una computadora con buenos recursos computacionales, se debe pensar en “cualquier usuario, en cualquier equipo”

Descripción

El objetivo de la actividad es desarrollar una aplicación que imite el funcionamiento de Wordle (<https://wordle.danielfrg.com>), para ello, el estudiante debe integrar las tecnologías web principales (HTML, CSS y JavaScript) de tal manera que la aplicación se ejecute únicamente en el navegador.



El juego consiste en adivinar una palabra de cinco letras en menos de 6 intentos. Cada palabra que el usuario ingrese debe ser una palabra válida y que se encuentre en el diccionario de la aplicación. Luego enviar la palabra, la aplicación le muestra pistas sobre la palabra correcta:

- Se marca en verde las letras correctas y que se encuentran en el lugar correcto.
- Se marca en amarillo las letras correctas pero que no se encuentran en su lugar correcto.
- Se marca en gris las letras que no forman parte de la palabra final.
- Además, se marca en el teclado virtual, las letras ya utilizadas con su color respectivo.

CÓMO JUGAR ×

Adivina la palabra oculta en seis intentos.

Cada intento debe ser una palabra válida de 5 letras.

Después de cada intento el color de las letras cambia para mostrar qué tan cerca estás de acertar la palabra.

Ejemplos

GA T O S

La letra **G** está en la palabra y en la posición correcta.

V O **C** A L

La letra **C** está en la palabra pero en la posición incorrecta.

C A N T **O**

La letra **O** no está en la palabra.

Puede haber letras repetidas. Las pistas son independientes para cada letra.

Si el usuario adivina la palabra correcta, se le muestra un resumen con el resultado:

- Un resumen en forma de emoticones para ser compartido en alguna red social.
- El historial de ejecuciones pasadas.
- El tiempo que hace falta para que se publique la siguiente palabra (en la aplicación original, solo se genera una palabra por día que es la misma para todos los usuarios, en nuestro caso se debe incluir un parámetro para definir este tiempo y un botón con la etiqueta “Nuevo juego”)



Para la interfaz gráfica, puede utilizar como referencia la aplicación original, aunque debe incluir elementos visuales originales.

Criterios de Evaluación

Rubro	Ponderación
Estructural de la aplicación (HTML)	20%
Código javascript funcional (js)	50%
Calidad de los elementos gráficos (css)	30%

Algunas observaciones:

- El código se debe realizar utilizando “JavaScript Vanilla”: sin dependencias o uso de librerías externas.
- Se recomienda usar las propiedades Flexbox de CSS para la ubicación gráfica de los elementos.
- Se le motiva a darle el mayor uso posible a las capacidades de CSS.
- En el directorio de Onedrive se agrega un JSON con una lista de palabras con 5 letras (tomado de [wiktionary](#))
- Se debe incluir alguna animación con CSS en al menos un elemento gráfico que se debe de activar por interacción del usuario.

- La entrega se debe realizar en el TecDigital a más tardar el viernes 17 de octubre a las 10:00 p.m.

Referencias

Repositorio con una gran cantidad de recursos: <https://github.com/snipcart/learn-vanilla-js>

<https://blog.gft.com/es/2019/02/22/vanilla-js-o-la-importancia-de-los-basicos-en-javascript/>