

Práctica I – Listas enlazadas (C++) – Lenguajes de programación

Contexto

Wordle es un juego de palabras donde el usuario busca adivinar una palabra de n letras en 6 intentos. En cada uno de estos intentos, el juego le provee pistas informativas al jugador indicando cuáles letras de la palabra del intento efectivamente corresponden a las letras de la palabra a adivinar, teniendo en cuenta su posición. Cuando el jugador evalúa un intento, recibe de vuelta su intento evaluado con un color para cada una de las letras del intento:

- Sin color para las letras que no están en la palabra a adivinar.
- Color amarillo para las letras que aparecen en la palabra a adivinar, pero no está en la posición adecuada.
- Color verde para las letras que aparecen en la palabra a adivinar y están en la posición adecuada.

Ejemplo: simulando un intento para adivinar la palabra “HOLA”.

- Intento 1: “**COSA**”
- Evaluación 1. “**COSA**”
- Intento 2: “**LORA**”
- Evaluación 2: “**LORA**”
- Intento 3: “**HOLA**”
- Evaluación 3: “**HOLA**”

Si el jugador consigue adivinar la palabra en a lo sumo 6 intentos, ganará el juego.

Este juego es propiedad del *New York Times*. Puede probar algunos intentos para que entienda a profundidad el juego: <https://www.nytimes.com/games/wordle/index.html>

Enunciado de la práctica

Recrear el juego *Wordle* usando listas enlazadas y C++.

Flujo de trabajo:

1. El programa le dará la bienvenida al usuario y le dará la opción de iniciar el juego.
2. El programa principal cargará una palabra de manera aleatoria de un banco de palabras en mayúsculas que usted mismo creará y le informará al usuario cuántas letras conforman la palabra a adivinar.
3. Esta palabra debe ser almacenada en una lista enlazada donde cada nodo contenga el valor de cada una de sus letras, LLP para el resto de la explicación.
4. El usuario ingresará palabras como intentos hasta adivinar la palabra o terminar sus 6 intentos sin éxito.
5. La palabra ingresada por el usuario deberá ser almacenada en una lista enlazada donde cada nodo contenga una letra, LLI para el resto de la explicación.
6. Los nodos de LLI deben tener un atributo adicional llamado “color” para almacenar el color con el que se deberá imprimir al hacer el proceso de evaluación, antes de mostrársela al usuario.

7. La lista LLI se compara contra la lista LLP nodo a nodo y se actualiza el atributo “color” de cada nodo a medida que se atraviesa LLI con base en las reglas del juego: sin color para los nodos que estén en LLI, pero no en LLP, color amarillo para los nodos que estén en LLI y LLP, pero no en la misma posición y color verde para los nodos que estén en LLI y LLP en la misma posición.
8. Una vez termine la evaluación y la actualización de LLI, el programa le mostrará por consola los valores de cada uno de los nodos de LLI con el color asignado según la evaluación y le solicitará la palabra para su nuevo intento hasta terminar el juego.
9. Si el usuario ingresa una palabra que contenga más o menos letras que la palabra a adivinar, el programa le deberá informar que es un intento inválido y no lo evaluará ni contará como un intento de los 6 permitidos. El programa le solicitará de nuevo una palabra para el intento actual y seguirá su ejecución normal hasta el fin del juego.
10. Si el usuario acierta la palabra a adivinar, se terminará el programa informándole al usuario que ganó el juego.
11. Si el usuario no acierta la palabra a adivinar en los 6 intentos, en el último intento actualizará todos los nodos con color rojo y lo mostrará por consola, acompañado de un mensaje informándole al usuario que perdió el juego.
12. Tanto LLP como LLI deben ser gestionadas usando punteros.
13. Puede crear tantas clases como crea necesarias para resolver el problema. Cada una de estas debe ser desarrollada en un archivo *header* y cargada al programa principal.

Entrega y sustentación

- **Fecha de entrega y sustentación:** septiembre 26 de 2022 6:00 a.m.
- Esta práctica se puede desarrollar de manera individual o en parejas.
- La entrega se hará vía *interactiva* con un enlace a un repositorio de *github* con la solución propuesta.
- La sustentación se hará, en el caso de que se hagan en parejas, sobre uno de los integrantes del grupo de manera aleatoria.