Trabajo de Enfoque

PROGRAMACIÓN

WORDLE

Josefa Macías Manceras

Enlace de Youtube Explicación_Wordle_Java Josefa Macias Manceras:
https://www.youtube.com/watch?v=EeNgcBJLTd8

Código:

```
public class Main {
   public static void main(String[] args) {
      String[] words = WordleFileManager.loadWordsFromFile("words.txt"); //carga las palabras
      if (words.length == 0) { //verifica si hay palabras
         System.out.println("No se han encontrado palabras en 'words.txt'");
         return; //sino hay palabras, termina
      }

      WordleGame game = new WordleGame(words);
      game.start(); //inicia el juego
   }
}

public class WordleFeedback {
   private static final int WORD_LENGTH = 5; //el codigo funciona con palabras de max 5 letras

   public static final String ANSI_RESET = "\u001B[0m";// restablece el color del texto después de
aplicar un color
   public static final String ANSI_GREEN = "\u001B[32m"; // verde correcto con posición correcta
   public static final String ANSI_YELLOW = "\u001B[33m"; // amarillo correcto en posición
incorrecta
   public static final String ANSI_WHITE = "\u001B[37m"; // blanco no está la letra

   private static String applyColor(String letter, String color) {
      return color + letter + ANSI_RESET; //aplica el colora una letra, y restablece el color
   }

   public static String feedBackString(String guess, String secretWord) {
      StringBuilder feedback = new StringBuilder(); //genera una cadena de texto donde cada color
indica su función
      for (int i = 0; i < WORD_LENGTH; i++) { //compara cada letra con la palabra secreta
         char guessChar = guess.charAt(i);
         char secretChar = secretWord.charAt(i); //se obtiene el caracter y se alamacena en las
variables

         if (guessChar == secretChar) {
            feedback.append(applyColor(String.valueOf(guessChar), ANSI_GREEN)); //verde correcta
y posición correcta
         } else if (secretWord.contains(String.valueOf(guessChar))) {
            feedback.append(applyColor(String.valueOf(guessChar), ANSI_YELLOW)); //amarillo
correcta en posición incorrecta
         } else {
            feedback.append(applyColor(String.valueOf(guessChar), ANSI_WHITE)); //blanco no
existe
```

```
            }
        }
        return feedback.toString();
    }
}

import java.io.*; //importa clases para manejar -los archvios
import java.util.*; //clases para manejar las listas

public class WordleFileManager { //cargar las palabras desde words.txt y guardar el historial
    public static String[] loadWordsFromFile(String filePath) {
        List<String> wordsList = new ArrayList<>(); //lista para guardar las palabras
        try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
            String line;
            while ((line = br.readLine()) != null) { //lee linea por línea
                wordsList.add(line.trim().toLowerCase()); //agrega cada palabra en minúscula
            }
        } catch (IOException e) { //captura los errores en la lectura del archivo
            System.out.println("Error al leer el archivo: " + e.getMessage());
        }
        return wordsList.toArray(new String[0]); //convierte la lista en un array y lo devuelve
    }

    public static void saveGameHistory(String filePath, List<String> history) { //guarda el historial de
intentos
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath, true))) {
            for (String attempt : history) {
                writer.write(attempt + "\n"); //escribe cada intento en una nueva línea
            }
            writer.write("----------------------\n"); //agrega un separador entre las partidas
        } catch (IOException e) { //captura errores en la escritura del archivo
            System.out.println("Error al guardar el historial: " + e.getMessage());
        }
    }
}

import java.util.*; //clases para manejar las listas

public class WordleGame { //lógica del juego, intentos del usuario
    private static final int MAX_TRIES = 6; //max intentos
    private static final int WORD_LENGTH = 5; //5letras max

    private String[] wordList;
    private String secretWord; //la palabra secreta
    private int remainingAttempts; //intentos que quedan
    private List<String> triesHistory; //historial de intentos

    public WordleGame(String[] words) { //constructor inicializa la lista de palabras, selecciona y
define el max de intentos
```

```
        this.wordList = words;
        this.secretWord = selectRandomWord(words); //palabra aleatoria
        this.remainingAttempts = MAX_TRIES; //inicializa los intentos restantes
        this.triesHistory = new ArrayList<>(); //crea una lista vacia para el historial de intentos
    }

    private String selectRandomWord(String[] words) { //método para seleccionar palabra aleatoria de la lista
        Random random = new Random();
        return words[random.nextInt(words.length)];
    }

    public void start() { //inicia la partida, guarda el historial
        Scanner scanner = new Scanner(System.in);

        System.out.println("¡Bienvenido a Wordle! Adivina la palabra de 5 letras.");

        while (remainingAttempts > 0) { //bucle mientras quedan intentos
            System.out.println("\nIntentos restantes: " + remainingAttempts);
            showTriesHistory(); //muestra historial de intentos

            String guess = getUserInput(scanner); //solicita una palabra
            triesHistory.add(guess); //guarda la palabra en el historial

            if (guess.equals(secretWord)) { //verifiica si la palabra es la correcta
                System.out.println("¡Has adivinado la palabra!: " + secretWord.toUpperCase());
                break; //finaliza el juego si aciertas
            } else {
                System.out.println(WordleFeedback.feedBackString(guess, secretWord));
                remainingAttempts--; //sino es la palabra reduce el nº de intentos restantes
            }
        }

        if (remainingAttempts == 0) { //si se agotan los intentos muestra la palabra
            System.out.println("Se acabaron los intentos. La palabra era: " + secretWord.toUpperCase());
        }

        WordleFileManager.saveGameHistory("historial.txt", triesHistory); //guarda el historial de la partida
        scanner.close();
    }

    private void showTriesHistory() { //muestra el historial de intentos previos
        if (triesHistory.isEmpty()) {
            System.out.println("No hay intentos previos.");
        } else {
            System.out.println("Historial de intentos:");
            for (String attempt : triesHistory) {
```

```
            System.out.println(attempt);
        }
    }
  }

  private String getUserInput(Scanner scanner) { //método para introducir la palabra y asegurar que sea valida
    String input;
    do {
      System.out.print("Introduce una palabra de 5 letras: ");
      input = scanner.nextLine().trim().toLowerCase(); //convierte la palabra a minus
      if (input.length() != WORD_LENGTH) { //verifica que tenga 5 letras
        System.out.println("Error: La palabra debe tener exactamente 5 letras.");
      }
    } while (input.length() != WORD_LENGTH); //se repite hasta que sean 5 letras
    return input;
  }
}
```