

Instituto Tecnológico de Costa Rica

Escuela de ingeniería en computación

IC4700– Bases de Datos I

Profesor: Allan Rodríguez Dávila

Tarea I

Estudiante: Diego Araya Ureña

Grupo 60

I Semestre

2025

Documentación Externa

Descripción del problema

Para la tarea I, el objetivo es realizar una aplicación web que combine el frontend con el backend para implementar el juego de el Ahorcado. Esta es una adaptación del juego para dos jugadores que les permite competir entre sí, adivinando letras entre rondas alternadas, donde el objetivo es adivinar la palabra primero que el otro jugador, en la menor cantidad de intentos posibles. La aplicación se encarga de gestionar la lógica del juego, a su vez de registrar y mostrar un historial de partidas con su información respectiva.

Requerimientos Funcionales

La aplicación debe cumplir con los siguientes requerimientos:

a) Jugar Partida de Ahorcado

1. Iniciar Partida:

- El sistema debe permitir iniciar partidas, solicitando los nombres de los dos jugadores.
- Cada partida constará de 2 rondas de juego.

2. Inicio de Partida:

- El sistema debe elegir aleatoriamente quién será el Jugador 1 y quién el Jugador 2.
- El Jugador 1 siempre inicia la partida.

3. Jugar Rondas:

- El sistema debe permitir jugar en turnos alternos. Cada jugador juega su turno y luego le toca al otro.
- A cada jugador se le asigna una palabra aleatoria para adivinar. Las palabras deben tener un largo similar y ser seleccionadas de una "base de datos" en formato JSON.

4. Interfaz de Usuario:

- La aplicación debe contar con una interfaz amigable, ya sea en modo texto o gráfico, que permita a los jugadores interactuar con el sistema de manera intuitiva.

5. Finalización del Juego:

- El sistema debe detectar cuándo el juego ha terminado y mostrar el resultado:
 - Ganador: Indicar el nombre del jugador que ganó.
 - Empate: En caso de que no haya un ganador claro.
- Factores de Desempate:
 - El primer factor de desempate es la cantidad de rondas completadas con éxito.
 - El segundo factor es la sumatoria del tiempo efectivo que cada jugador invirtió en la partida.
- Al finalizar, el sistema debe mostrar:
 - La cantidad de rondas completadas.
 - El tiempo invertido por cada jugador.

b) Historial de Resultados

- El sistema debe mostrar en una página web todos los resultados de las partidas jugadas.
- La información mostrada debe incluir:
 - Nombres de los jugadores.
 - Resultado de la partida (ganador o empate).
 - Ganador (en caso de que haya uno).

Descripción de la solución

La aplicación está dividida en dos componentes principales: Backend y Frontend.

Backend

- Tecnologías: Node.js, Express.js.
- Funcionalidades:
 - Gestiona la lógica del juego (inicio de partidas, asignación de palabras, turnos, finalización del juego).
 - Almacena los datos de las partidas en un archivo JSON (games.json).
 - Proporciona endpoints para:
 - Iniciar partidas (/api/start-game).
 - Realizar adivinanzas (/api/make-guess).
 - Obtener el historial de partidas (/api/get-games).

Frontend

- Tecnologías: React.js, HTML, CSS.
- Funcionalidades:
 - Interfaz gráfica para que los usuarios ingresen sus nombres y jueguen.
 - Muestra el estado actual del juego (palabra oculta, letras adivinadas, intentos restantes).
 - Muestra el resultado final (ganador, tiempo invertido, rondas completadas).
 - Proporciona una página para ver el historial de partidas.

Base de Datos de Palabras

- Se utiliza un archivo JSON (words.json) que contiene una lista de palabras relacionadas con la programación, con un largo similar (entre 5 y 8 letras). Ejemplos de palabras incluyen: "codigo", "clase", "objeto", "metodo", "funcion".

Análisis de Resultados

A continuación, se presenta un análisis de los resultados obtenidos en el desarrollo de la aplicación, dividido en objetivos alcanzados y objetivos no alcanzados. Este análisis servirá como una revisión crítica del proyecto y permitirá identificar áreas de mejora para futuras iteraciones.

Objetivos Alcanzados

1. Ingreso de Jugadores:

- ☒ El sistema permite ingresar los nombres de dos jugadores diferentes.
- ☒ Los nombres de los jugadores se utilizan para identificar a cada participante durante la partida.

2. Inicio de Partida:

- ☒ El sistema inicia una partida correctamente, asignando aleatoriamente quién será el Jugador 1 y quién el Jugador 2.
- ☒ Se asigna una palabra aleatoria a cada jugador, seleccionada de una base de datos en formato JSON.

3. Juego por Turnos:

- ☒ Los jugadores alternan turnos para adivinar letras de su palabra asignada.
- ☒ El sistema registra el tiempo efectivo de cada jugador, sumando el tiempo invertido en cada turno.
- ☒ Se muestra la cantidad de intentos restantes para cada jugador.

4. Letras Probadas:

- ☒ El sistema tacha o marca las letras que ya han sido probadas por cada jugador, evitando que se repitan.

5. Detección del Ganador:

- ☒ El sistema detecta automáticamente al ganador cuando un jugador adivina su palabra antes que el otro.
- ☒ Si un jugador agota sus intentos, el otro jugador es declarado ganador automáticamente.

6. Historial de Partidas:

- ☒ El sistema permite acceder a un historial de partidas donde se muestran los resultados de todas las partidas jugadas.
- ☒ En el historial, se muestran detalles como los nombres de los jugadores, el resultado (ganador) y el tiempo invertido.

Objetivos No Alcanzados

1. Sistema de Desempate:

- ☒ No se implementó un sistema de desempate basado en la cantidad de rondas completadas con éxito y el tiempo efectivo.
- ☒ El juego no maneja empates, ya que siempre declara un ganador cuando uno de los jugadores adivina su palabra o agota sus intentos.

2. Empates:

- ☒ No se logró implementar la funcionalidad de empates. Actualmente, el juego siempre declara un ganador, incluso si ambos jugadores no logran adivinar sus palabras.
- ☒ Falta la lógica para comparar el progreso de ambos jugadores y determinar un empate en caso de que ambos pierdan.

