



Tarea #1

Erin Camacho González

Instituto Tecnológico de Costa Rica

Lenguajes de Programación

Prof. Allan Rodríguez Dávila

Semestre II

24/08/2025

Descripción del problema

Se hizo una aplicación web para que dos personas jueguen a adivinar números.

La idea es simple: el sistema elige un número secreto entre 1 y 100 y los jugadores adivinan por turnos. El juego tiene 6 rondas en total (3 por cada uno), siempre alternando quién adivina. En cada intento el sistema dice si el número secreto es más grande o pequeño que el que se escribió.

Al final, el ganador es quien necesitó menos intentos para acertar en sus rondas. Si empatan en intentos, gana quien tardó menos tiempo en total. Después de terminar, se muestra un resumen con intentos y tiempo por jugador y una tabla con los datos de cada ronda.

Las partidas que ya terminaron quedan guardadas en un archivo JSON y se pueden ver en la pantalla de historial.

La aplicación está separada en dos partes:

- Frontend (la interfaz): pantallas de Menú, Juego e Historial.
- Backend (la lógica): crea partidas, maneja los intentos, calcula los resultados y guarda el historial en un archivo JSON.

Análisis de resultados (checklist):

Objetivos alcanzados

En el inicio, el usuario escribe los nombres de los dos jugadores y puede empezar una partida nueva. El sistema elige al azar quién será el Jugador 1 y, a partir de ahí, quedan registrados ambos nombres para toda la partida. La conexión entre frontend y backend se hace por medio de rutas /api; el frontend llama a estas rutas y recibe las respuestas con la información actualizada del juego.

Durante el juego, todo lo importante se actualiza desde el backend: el turno de quién adivina, el número de la ronda, la cantidad de intentos que lleva cada jugador y el mensaje con la pista después de cada intento (si el número es mayor o menor, o si se acertó). Cada ronda guarda su tiempo (inicio y fin) para que, además de los intentos, también se pueda medir cuánto tardó cada jugador. Cuando alguien acierta, la ronda se cierra, se pasa a la siguiente y se genera un nuevo número secreto. Así se completa el ciclo de las 6 rondas.

Al terminar la partida, el backend calcula el ganador con las reglas definidas (menos intentos; si hay empate, menos tiempo). En la pantalla final se muestra un resumen ordenado: dos tarjetas con los totales de cada jugador (intentos y tiempo) y una tabla con las 6 rondas, indicando quién adivinó, cuántos intentos usó y cuánto tardó cada una. Esta información es fácil de leer y permite ver de un vistazo cómo se desarrolló el juego.

El historial funciona con un archivo data.json donde se guardan solo las partidas terminadas. De esta forma, la lista del historial es limpia y muestra partidas completas, con los nombres de los jugadores, el ganador y los totales de intentos y tiempo que quedaron registrados al cierre. Cada partida guardada tiene un ID numérico incremental (1, 2, 3, ...), lo que facilita identificar y ordenar las partidas. La pantalla del historial mantiene el mismo estilo visual que el resto de la aplicación, con una tabla simple y clara, y un botón para volver al menú.

En cuanto a la robustez básica, el backend valida que el número ingresado esté dentro del rango (1–100) y que sea realmente el turno del jugador que intenta. Si hay errores (por ejemplo, un turno equivocado o un número fuera del rango), se devuelve un mensaje entendible y no se rompe la partida. La separación entre interfaz y lógica se mantiene: la interfaz muestra y envía datos, y el backend decide lo que debe pasar en el juego. Esto ayuda a que el proyecto se entienda mejor y sea más fácil de mantener.

El juego funciona bien y cumple lo pedido: jugar por turnos, dar pistas, llevar el conteo, decidir un ganador y mostrar un resumen. Además, el historial permite revisar partidas anteriores que ya terminaron.

Enlace de GitHub:

https://github.com/Erincam09/Tarea01_AplicacionWeb.git