

Toros y Vacas

Toros y vacas es un juego tradicional inglés a lápiz y papel para dos jugadores cuyo objetivo es adivinar un número constituido por cuatro dígitos.

En una hoja de papel, un jugador escribe un número de 4 dígitos y lo mantiene en secreto. Las cifras deben ser todas diferentes, no se debe repetir ninguna dentro del mismo número. El otro jugador trata de adivinar el número secreto en varios intentos que son anotados y numerados.

En cada intento anota una cifra de cuatro dígitos completa, ésta es evaluada por el jugador que guarda el número secreto. Si una cifra está presente y se encuentra en el lugar correcto es evaluada como un toro, si una cifra está presente pero se encuentra en un lugar equivocado es evaluada como una vaca. La evaluación se anota al lado del intento y es pública.

Ejemplo:

Número Secreto: 4271			
Intentos	Número	Evaluación	
1	1234	1T2V	(1 toro y 2 vacas, el toro es "2", las vacas son "4" y "1".)
2	5678	1T0V	
3	9012	0T2V	
4	9087	0T1V	
5	
6	
7	
8	4271	4T0V	

Al adivinarse el número termina la partida.

Se necesita crear una RESTful API usando Laravel para ser usada en el Primer Campeonato Mundial de Toros y Vacas. Los requerimientos son los siguientes:

1. Use el prefijo /game/ para todos los endpoints.
2. Use una base de datos SQLite para almacenar los juegos creados y la información relacionada que necesite. No guarde de forma permanente todas las combinaciones recibidas.
3. Especifique en la configuración de la aplicación una variable para especificar el tiempo máximo para solucionar el reto antes de declarar "Game Over".
4. Implemente un endpoint por cada una de las siguientes funcionalidades:
 - a. **Crear un nuevo juego.** Los datos requeridos serán: usuario y edad. Retorna un identificador único para cada juego.
 - b. **Proponer combinación:** Validar los dígitos propuestos y retornar el resultado según las reglas descritas. Usar diferentes códigos HTTP según las posibles respuestas:
 - i. Resultado de la comprobación:
 1. Combinación numérica propuesta.
 2. Cantidad de toros y vacas alcanzados
 3. Número de intentos alcanzados
 4. Tiempo disponible en el juego.

5. Evaluación: "Tiempo en segundos" / 2 + "Cantidad de Intentos"

6. Ranking del juego actual con respecto a otros juegos según su evaluación. Se ubican primero siempre los juegos ganados.

ii. Combinación duplicada: los dígitos ya fueron enviados previamente en el mismo orden.

iii. Game Over: El tiempo máximo del juego fue alcanzado. Retornar la combinación que se trataba de "adivinar".

c) **Eliminar datos del juego:** Se debe especificar el identificador único del juego.

d) **Obtener respuesta previa según el número del intento.** La respuesta sería similar a la que retorna el endpoint: **Proponer combinación.**

5. La aplicación será usada por usuarios anónimos, por lo que se requiere que la información sea protegida de usuarios malintencionados.

6. Documente la API usando el estándar Swagger.

7. Use las mejores prácticas y estándares en la codificación de soluciones con PHP.

8. Sería deseable la inclusión de tests unitarios para las lógicas más críticas, un código documentado y aplicando buenas prácticas de programación y en el uso de Laravel.

9. Publique su solución en <https://about.gitlab.com/> (repositorio privado) e invite a colaborar a todos los contactos del email que incluye este ejercicio.

10. Cree un Readme.md con todos los detalles que considere necesarios.

11. Cree un solo modelo de datos (una sola tabla en la base de datos) para almacenar toda la información relacionada con un juego.

12. Dispone de 40 h laborables para realizar esta solución, aunque puede ser enviada antes si lo desea.