

Primer Parcial - 28/10/2023

GOTO Game JAM - Sistema de Votación de Jueces

Contexto:

GOTO Game JAM es una competencia anual donde equipos de desarrolladores se reúnen para crear un videojuego en un plazo de 48 horas. Como parte del proceso, un grupo de jueces evalúa y califica los videojuegos en varias categorías. Se te ha contratado para desarrollar el sistema de votación que permita a estos jueces emitir sus calificaciones.

Las entidades ya establecidas son:

- **Judges** con campos:
 - `_id` (identificador único)
 - `name`
- **Games** con campos:
 - `_id` (identificador único)
 - `name`
 - `genre` (nombre del genero del juego)
 - `members` (lista con los nombres de los participantes que hicieron el juego)
 - `edition` (indica el año o edición de la GOTO Game JAM).

Tu tarea es diseñar e implementar el sistema que permita registrar las votaciones de los jueces en las siguientes categorías:

1. Jugabilidad
2. Arte
3. Sonido
4. Afinidad a la temática

Instrucciones:

1. Modelo de Datos:

La entidad de **Votes**. debe tener

- Un identificador único.
- Juez y Juego
- Puntos de cada categoría

2. Acciones que debe permitir el API:

- Un juez debe poder generar una votación a un juego.
- A partir del id de Juez debe poder obtener el nombre de los juegos y los puntos de cada categorías donde ese haya realizado alguna votación.
- A partir del id del juego, se debe poder visualizar los nombres de los jueces y los puntos de cada categoría que puso ese juez.
- A partir de la edición, se debe poder obtener el listado de juegos ordenados de mayor a menor puntaje (el puntaje es calculado a partir de la suma de todos los puntos obtenidos de todas las categorías).
 - Se debe poder filtrar por genero.
- A partir de un id de juego, calcule y devuelva el promedio de puntuaciones de ese juego específico en cada una de las categorías junto con todos los datos del juego.
- CRUD de los juegos.

3. Consideraciones:

- Un juez sólo puede emitir una votación por juego. Si intenta votar nuevamente por el mismo juego, le deberá informar que no se puede realizar dicha acción.
- Las puntuaciones deben estar entre 1 y 10.
- Deberá haber validaciones para asegurarse de que los jueces y juegos existan al registrar una votación.
- Puede modificar cualquier entidad para lograr el objetivo.
- Un juez puede no votar un juego, pero si realiza la votación siempre es para cada una de las categorías.

4. Que se evaluara:

- Correcto uso de la división de responsabilidades (routes, controllers, services, etc).
 - Correcto uso de los filtros de MongoDB.
 - Correcto uso del **API del Driver Nativo** de MongoDB.
 - Correcto uso de las reglas de un **API Rest** (URI, Estados, JSON, Verbos).
 - Que no realicen acciones o tareas que no se piden en el enunciado.
 - Uso correcto de las estrategias **Embed** y **Reference** para realizar las relaciones entre las entidades en una base de datos no relacional.
 - Correcto nombramiento de los endpoint para realizar las tareas solicitadas.
 - Uso correcto del API de **Express JS**.
 - Correcto uso de **ECMAScript** modules.
 - Recuerden que hay cosas que las debe resolver la base de datos y otras que las debe resolver el servicio.
 - Puedes adaptar las entidades a las necesidades de la solución planteada.
-

Concluye el desarrollo siguiendo buenas prácticas y asegurándote de que todas las funcionalidades trabajen de forma correcta y eficiente :Buena suerte!