Pràctica Tema 2

Desenvolupament d'una API REST per al catàleg d'una plataforma de jocs de taula amb Express i Mongoose

En aquesta pràctica ampliarem el fet en la pràctica del tema anterior, per a construir una API REST més complexa i completa sobre l'aplicació de creació d'una plataforma de jocs de taula, però aquesta vegada utilitzant una base de dades MongoDB i l'API que ofereix la llibreria de Mongoose.

1. Estructura de l'aplicació

Crearem una aplicació anomenada **playREST_V2**, i instal·larem dins els mòduls de *express* i *mongoose*. L'aplicació estarà estructurada en les següents carpetes i arxius (a més de l'arxiu `package.json` que generarem amb `npm init`):

- Arxiu `index.js` en l'arrel del projecte, on anirà el servidor principal.
- Carpeta `models` on emmagatzemarem els esquemes i models de l'aplicació. Concretament tindrà un arxiu: `juegos.js`, que completarem després.
- Carpeta `routes` amb l'encaminador per a la col·lecció de jocs de taula. Contindrà l'arxiu `juegos.js`.

2. Definint esquemes i models

En la carpeta `models` emmagatzemarem la definició d'esquemes i models de la nostra base de dades. En concret serà un, amb els camps que s'indiquen a continuació (s'han de respectar els noms dels camps, que es marquen en negreta, incloent majúscules o minúscules si escau):

Model *juego*

En l'arxiu `models/juego.js` definirem l'esquema per a la col·lecció de jocs de taula. Cada joc tindrà els següents camps:

- El **nombre** del joc (obligatori, de tipus text, amb una longitud mínima de 6 caràcters).
- La **descripcion** el joc (text llarg obligatori, sense grandària mínima).
- La **edad** mínima recomanada en anys (numèric obligatori, major que 0 i menor que 100).
- El nombre de **jugadores** permés (numèric obligatori).
- El **tipo** de joc (tipus enum amb les següents categories: rol, escape, daus, fitxes, cartes i tauler).
- El **precio** del producte (numèric obligatori, major que 0).
- La **imagen** del producte (text amb la ruta relativa, no obligatòria).
- **Ediciones** més modernes o temàtiques, per a jugar partides diferents però sense perdre l'essència del joc original. Per exemple, del Cluedo podem trobar diferents versions com l'edició Juego de Tronos, l'edició Big bang Theory o l'edició Junior. Aquest camp serà un array de subdocumentos de la mena d'esquema que comentarem a continuació.

Per a les edicions, definirem un esquema local a aquest model, que contindrà aquests camps:

- El nom de l' **edicion** del joc. Serà de tipus text obligatori.
- El camp **anyo** de llançament de l'edició. Serà de tipus numèric (major que 2000 i menor que l'any actual).

Recorda exportar el model.

3. Els encaminadors

En la carpeta `routes` definirem l'encaminador del model anterior. En TOTS els serveis, s'ha d'enviar:

- Un codi d'estat apropiat: 200 si tot ha anat bé, 400 si hi ha una fallada en la petició i 500 si hi ha una fallada en el servidor
- Un objecte JSON amb aquests atributs:
  - `ok` : de tipus booleà indicant si la petició s'ha processat satisfactòriament o no
  - `error` : només serà present si `ok` és fals. Contindrà el missatge amb l'error que s'haja produït
  - `resultado` : només serà present si `ok` és vertader. Contindrà el resultat que s'envia com a resposta. Aquest resultat es detalla a continuació per a cada servei.

Encaminador per a *juegos*

Aquest encaminador respondrà a URIs que comencen per `/jocs`. Es demana implementar aquests serveis:

- `GET /juegos` : retornarà com a `resultado` tot el llistat de jocs, amb tots els seus camps, incloent la informació de les edicions. Si no hi haguera jocs, es retornarà un codi 500 amb el missatge `"No se encontraron juegos de mesa"` .

- `GET /juegos/:id` : retornarà com a `resultado` la fitxa per a un joc a partir del seu *aneu*. Si no es troba, es retornarà un error 400 amb el missatge `"Juego no encontrado"` .

- `POST /juegos` : afegirà el joc que es reba en la petició a la col·lecció. En aquesta petició s'enviaran els camps bàsics del joc (és a dir, tot menys el *id* i el array de les edicions). Es retornarà com a `resultado` el joc inserit, o un codi 400 amb el missatge `"Error insertando el juego"` , si hi ha hagut algun error.

- `PUT /juegos/:id` : permetrà modificar les dades bàsiques del joc (tot excepte el *id* i el array d'edicions). De nou, s'enviaran en la petició els camps bàsics del joc, com en POST. Es retornarà com a `resultado` el joc modificat, o un codi 400 amb el missatge `"Error modificando el juego"` .

- `POST /juegos/ediciones/:idJuego` : permetrà modificar el *array d'edicions del joc amb el *id* indicat, afegint la nova versió del joc que s'envie en la petició. S'enviaran en la petició els camps amb el nom de l'edició i l'any de llançament, i es retornarà com a resultat el joc sencer modificat, o un codi 400 i el missatge `"Error modificando las ediciones del juego"` , en cas d'error.

- `DELETE /juegos/:id`: permitirá eliminar un juego, junto con sus ediciones, a partir de su *id*. Como `resultado` se devolverá el juego eliminado con todos sus campos, o un error 400 y el mensaje `"Error eliminando el juego"`.

- `DELETE /juegos/ediciones/:idJuego/:idEdicion`: permetrà eliminar l'edició amb el *id* de l'edició indicada, per al joc amb el *id* de joc indicat. Retornarà com a `resultat` el joc en el seu estat actual, o un error 400 i el missatge `"Error eliminado la edición del juego"`.

4. El servidor principal

El servidor principal deurà:

- Carregar les llibreries necessàries (almenys, *express* i *mongoose*)
- Carregar els encaminadors de juegos
- Connectar a una base de dades "juegos" de MongoDB
- Inicialitzar Express, i aplicar el middleware de *express.json* per a processar dades en format JSON, i associar els encaminadors respectivament a les rutes de `/juegos`.
- Posar en marxa el servidor Express pel port 8080.

5. Proves amb Postman o Thunder Client

Es demana, a més, que elaboreu una col·lecció de proves Postman o THunder Client anomenada **playREST_V2**, per a provar cadascun dels serveis indicats anteriorment. Exporta-la a un arxiu amb el mateix nom.

6. Lliurament i qualificació

Haureu d'entregar un arxiu ZIP o similar, amb el vostre nom i el prefix "PracT2". Per exemple, si us dieu José Pérez, l'arxiu de lliurament haurà de ser `PracT2_Jose_Perez.zip`. Dins, haurà de contindre:

- El projecte **playREST_V2** de Node, sense que continga la carpeta `node_modules`.
- L'arxiu de Postman o Thunder Client exportat, amb les proves de la col·lecció.

6.1. Qualificació de la pràctica

Els criteris per a qualificar aquesta pràctica són els següents:

- Estructura correcta del projecte, amb les carpetes i noms d'arxius indicats en l'enunciat, arxiu `package.json` correctament definit amb les dependències incorporades: **0,75 puntos**
- Model de dades: **2,5 puntos**:
    - Esquema i model bàsic per als jocs (sense les edicions): **1,5 punto**
    - Esquema local per a les edicions: **1 puntos**
- Encaminadors: **3,5 puntos**, repartits així:
    - Serveis implementats: **0,5 puntos** cada servei (7 en total)

- Funcionalitat de l'arxiu principal `index.js`: **1 puntos**
- Col·lecció Postman, amb els serveis correctament afegits per a provar-se: **1,75 puntos** (*0,25 punts* per a cada petició)
- Claredat i neteja del codi, i ús d'un comentari inicial en cada fitxer font explicant què fa: **0,5 puntos**.

Penalitzacions a tindre en compte

- Si algun servei no rep o retorna els atributs amb el nom indicat, o no respon a l'URI indicada, es qualificarà amb **0 puntos**, independentment del bé o malament que estiga el seu codi.

> **Ejemplo**: si en el servei `POST /juegos` rep un camp `nombreJuego` en lloc de l'original `nombre` que figura en l'esquema, el servei es qualificarà amb un 0.

> **Ejemplo 2**: si en el servei `GET /juegos` enviem en l'objecte de resposta un atribut `result`, en lugar de uno `resultado`, el servei es qualificarà amb un 0.

> **Ejemplo 3**: si en el servei d'esborrat (DELETE) es respon a l'URI `/juegos/borrar/id` en lloc de `/juegos/id`, el servei es qualificarà amb un 0.

- La no eliminació de la carpeta `node_modules` en l'arxiu ZIP de lliurament es penalitzarà amb 1,5 punts menys de nota global de la pràctica.

- Si se segueix una estructura de projecte, codi i/o noms d'arxiu diferent a la proposta, i no es justifica degudament, o aquesta justificació no és satisfactòria, es penalitzarà la qualificació global de la pràctica amb fins al 50% de la nota d'aquesta.

- En l'apartat de qualificació dels **enrutadores**, s'hauran d'obtindre **almenys 1,75 punts** del total de 3,5 per a considerar aprovada la pràctica (a més d'arribar a la nota mínima exigible).