Pràctica Tema 3

Desenvolupament d'una aplicació web Express per al catàleg d'una plataforma de streaming

Seguint amb la temàtica de pràctiques anteriors, desenvoluparem una aplicació web completa per a la gestió i visualització del catàleg de pel·lícules d'una plataforma de streaming, aprofitant gran part de la lògica implementada en la pràctica del tema anterior.

1. Estructura de l'aplicació

Crearem una aplicació anomenada FilmEs_v3, i instal·larem dins els mòduls següents:

- *express*
- *mongoose*
- *nunjucks* com a motor de plantilles
- *method-override* per a simular peticions PUT i DELETE
- *express-session* per a l'autenticació basada en sessions
- *bootstrap* per als estils generals de l'aplicació
- *multer* per a pujar imatges dels cartells de les pel·lícules
- ... qualsevol altre mòdul que vulgues utilitzar

Recorda primer inicialitzar l'arxiu `package.json` amb `npm init`, i després instal·la els mòduls amb `npm install`.

Internament, l'aplicació estarà composta de:

- Arxiu `index.js` en l'arrel del projecte, on anirà el servidor principal.
- Carpeta `models` on emmagatzemarem els esquemes i models de l'aplicació.
- Carpeta `routes` amb els diferents encaminadors que es detallaran més endavant.
- Carpeta `views` amb les vistes que definirem, i que s'explicaran més endavant.
- Carpeta `public` amb un cert contingut estàtic. En concret, hi haurà un arxiu CSS en `public/css/estils.css` amb els estils propis que vulgueu definir, i una carpeta `public/uploads` per a pujar les imatges de les pel·lícules.
- Carpeta `utils` amb alguns fitxers d'utilitat que es comentaran més endavant.

2. Definint esquemes i models

En la carpeta `models` emmagatzemarem la definició d'esquemes i models de la nostra base de dades.

2.1. El catàleg de pel·lícules

Hi haurà un arxiu `pelicula.js` amb la següent informació. S'han de respectar els noms dels camps, que es marquen en negreta, incloent majúscules o minúscules si escau:

Esquema i model *pelicula*

L'esquema de dades de cada pel·lícula contindrà els següents camps:

- El seu **titol** (obligatori, de tipus text, amb 2 caracters com a mínim).
- **Sinopsi** de la pel·lícula (obligatori, de tipus text, amb 10 caracters com a mínim).
- La seua **duracio** en minuts (obligatori, de tipus numèric, amb un valor mínim de 0)
- El seu **genere**, que serà una enumeració que podrà tenir els valors comedia, terror, drama, aventures i altres. Serà també obligatori definir-lo per a cada pel·lícula.
- La **imatge** del cartell de la pel·lícula (text amb la ruta relativa, no obligatòria).
- La seua **valoracio** (valor numèric obligatori, entre 0 i 5, incloent-hi ambdós extrems)
- Les **plataformes** de streaming on s'emet la pel·lícula en questió

Enrecorda't de definir el model associat a l'esquema de pel·lícules per a que les dades es guarden a una col·lecció anomenada peliculas, i exporta el contingut necessari per a la resta d'arxius del projecte.

Esquema de les plataformes de streaming

Per a les diferents plataformes en les quals s'emet una pel·lícula definirem un esquema local al model de pelicula, que contindrà aquests camps:

- El **nom** de la plataforma (obligatori, de 2 caràcters com a mínim). Per exemple, "Netflix", "Prime Video"...
- La **data** fins a la qual es podrà veure el contingut a la plataforma (no obligatori). Si no s'especifica aquest valor, s'entendrà que de moment no hi ha una data específica per a deixar de veure la pel·lícula a la plataforma indicada
- Si s'ha d'abonar alguna **quantitat** adicional en concepte de premium o similar per poder veure la pel·lícula a la plataforma. Per defecte serà que no.

Aquest segon esquema formarà part de l'esquema principal de pel·lícules com a un subdocument, i no tindrà model propi.

2.2. Els directors

A l'arxiu `models/director.js` definirem un esquema per a les dades dels directors de les diferents pel·lícules. Aquest esquema contindrá els següents camps:

- El nom del **director** (obligatori, de 5 caràcters com a mínim)
- L'any de **naixement** del director (no obligatori)

Aquest esquema tindrà un model associat per a guardar les dades dels directors una col·lecció anomenada directors. A més a més, s'associarà al model de pel·lícules mitjançant una relació entre models (afegirem un nou camp a l'esquema de pel·lícules per a indicar el seu director).

2.3. Els usuaris registrats

A més, hi haurà un arxiu `usuari.js` que definirà un esquema i model per als usuaris registrats en l'aplicació. Els camps a emmagatzemar de cada usuari seran:

- El **login** de l'usuari, de tipus text, amb grandària mínima de 5 caràcters, obligatori i que no admet duplicats(**unique*).
- El **password** de l'usuari, de tipus text, amb grandària mínima de 8 caràcters, que es guardarà encriptat. Pots utilitzar per a això la llibreria bcrypt o bé crypto.js. En aquest últim cas, pots utilitzar un encriptat SHA256.

Per a deixar uns usuaris prèviament emmagatzemats i que l'aplicació no haja d'ocupar-se de registrar usuaris, pots definir un arxiu auxiliar anomenat `generar_usuaris.js`, en la carpeta `utils` de l'aplicació, amb un parell d'usuaris de prova, i executar-lo perquè s'inserisquen. Per exemple, el contingut pot ser semblant a aquest (encara que encriptant els passwords en inserir):

```
const mongoose = require('mongoose');
const Usuari = require(__dirname + '/../models/usuari');

mongoose.connect('mongodb://localhost:27017/FilmEsV3');

Usuari.collection.drop();

let usu1 = new Usuari({
    login: 'may',
    password: '1234'
});
usu1.save();

let usu2 = new Usuari({
    login: 'laura',
    password: '5678'
});
usu2.save();
```

3. Els enrutadors

En la carpeta `routes` definirem els encaminadors associats a cada model.

3.1. La part pública

En l'enrutador `public.js` es definiran aquestes rutes GET:

- `/` (arrel de l'aplicació): renderitzarà la vista `public_index` de la carpeta de vistes, sense paràmetres.
- `/buscar`: buscarà totes les pel·lícules en el títol de les quals aparega el text que se li passa en el cos de la petició, i renderitzarà la vista `public_index` de la carpeta de vistes, passant-li com a paràmetre el conjunt de resultats obtingut, o el missatge "No es van trobar pel·lícules" si no hi ha resultats coincidents.

- `/pelicula/:id`: renderitzarà la vista `public_pelicula` de la carpeta de vistes, passant-li com a paràmetre les dades de la pel·lícula que el seu *id* li arribe com a paràmetre en la ruta. Si no es troba la pel·lícula, es renderitzarà la vista `public_error` amb el missatge "Pel·lícula no trobada".

Si es produeix algun error no contemplat en qualsevol de les rutes (és a dir, si se salta a la clàusula `catch` corresponent), es renderitzarà la vista `public_error`, sense missatge d'error, perquè mostre el missatge per defecte de "Error en l'aplicació".

3.2. L'administració de directors

Per a simplificar la gestió dels directors, pots deixar alguns emmagatzemats prèviament i que l'aplicació no haja d'ocupar-se d'inserir-los. Pots definir un arxiu auxiliar anomenat `generar_directors.js`, en la carpeta `utils` de l'aplicació, amb 4 o 5 directors de prova, i executar-ho perquè s'inserisquen. A l'hora d'inserir una pel·lícula, en el formulari d'inserció, fes que el camp per a assignar un director siga una llista desplegable amb els directors disponibles.

Es proposa com a part opcional crear els formularis d'inserció, esborrat i modificació dels directors.

3.2. L'administració de pel·lícules

En l'encaminador `pelicules.js` es definiran les següents rutes:

- `GET /`: renderitzarà la vista `admin_pelicules` passant-li com a paràmetre el catàleg o llistat de pel·lícules.
- `GET /pelicules/nova`: renderitzarà la vista `admin_pelicules_form`.
- `GET /pelicules/editar/:id`: renderitzarà la vista `admin_pelicules_form` passant-li com a paràmetre la pel·lícula amb el *id* indicat. Si no es troba la pel·lícula, es renderitzarà la vista `admin_error` amb el missatge "Pel·lícula no trobada".
- `POST /pelicules`: recollirà de la petició les dades de la pel·lícula, farà la inserció i, si tot ha sigut correcte, redirigirà a la ruta base d'aquest enrutador.
- `PUT /pelicules/:id`: recollirà de la petició les dades de la pel·lícula que el seu *id* es passa en la URL, i farà la modificació dels seus camps. Si tot ha sigut correcte, redirigirà a la ruta base d'aquest encaminador.
- Com a cas particular d'aquesta ruta, si no es rep el camp amb la imatge de la pel·lícula, s'entendrà que es vol deixar la imatge anterior. La resta de camps de la pel·lícula ha d'enviar-se per a poder-se modificar.
- `DELETE /pelicules/:id`: eliminarà la pel·lícula que el seu *id* es passa en la URL, redirigint si tot és correcte a la ruta base d'aquest encaminador.
- A més, podeu afegir les rutes addicionals que us puguen servir per a la gestió de les pel·lícules, com afegir o llevar plataformes de streaming a una pel·lícula.

Si es produeix algun error no contemplat en qualsevol de les rutes (és a dir, si se salta a la clàusula `catch` corresponent), es renderitzarà la vista `admin_error`, sense missatge d'error, perquè mostre el missatge per defecte de "Error en l'aplicació".

En el cas dels mètodes POST i PUT, si es puja alguna imatge de la pel·lícula, s'utilitzarà la llibreria `multer` per a copiar-la en la carpeta `public/uploads`, amb el prefix de la data actual.

3.3. L'autenticació d'usuaris

En l'enrutador `auth.js` es definiran les següents rutes:

- `GET /login`: renderitzarà la vista `auth_login`.
- `POST /login`: recollirà de la petició el login i password de l'usuari que intenta accedir i comprovarà si són vàlids. Si ho són, emmagatzemarà en sessió el login de l'usuari, i redirigirà a la ruta base de l'enrutador de pel·lícules. Si falla, renderitzarà el formulari de `auth_login` passant-li el missatge d'error de "Usuari incorrecte", perquè es mostre en aquest formulari.
- `GET /logout`: destruirà la sessió de l'usuari actual i redirigirà a l'arrel de la part pública.

3.4. Definint la zona d'accés restringit

Totes les rutes de `routes/pelicules.js` hauran d'estar protegides. Per a això, s'haurà de crear un arxiu `auth.js` en la carpeta `utils` del projecte. En aquest arxiu es definirà un middleware d'autenticació que verificarà si existeix un usuari emmagatzemat en sessió abans de deixar passar. En cas contrari, redirigirà al formulari de login.

S'haurà d'importar aquest arxiu des d'on corresponga (enrutador `routes/pelicules.js`), i aplicar el middleware en les rutes que es vulguen protegir.

4. Les vistes

En la carpeta de vistes `views` es definiran les següents vistes amb el contingut que es descriu per a cadascuna.

4.1. Part pública

Pel que fa a la part pública, es definiran tres vistes:

- `public_index.njk`: Mostrarà en el seu contingut un formulari de cerca, que s'enviarà per GET a la URL `/buscar`. Si rep un llistat de pel·lícules, els mostrarà sota el formulari, amb un enllaç per a anar a cada pel·lícula.
- `public_pelicula.njk`: Mostrarà la fitxa d'una pel·lícula, indicant tota la seua informació: títol, sinopsi, duració, gènere, imatge, valoració, director i plataformes de streaming en les quals s'emet.
- `public_error.njk`: Mostrarà el missatge d'error que se li indique, o el missatge genèric "Error en l'aplicació" si no rep cap missatge específic.

Aquestes tres vistes heretaran d'una plantilla comuna anomenada `public_base.njk`, que definirà, almenys:

- La capçalera (`head`) amb els estils Bootstrap i un bloc per al títol (`title`) de la pàgina
- El cos (`body`) amb:
- Un menú de navegació amb un enllaç per a anar a l'inici de la web ('/') i un altre per a anar a la zona d'administració (a la ruta arrel d'aquesta zona)
- Un bloc per al contingut de la pàgina

4.2. Part d'administració

Pel que fa a la gestió de pel·lícules (part protegida), es definiran aquestes vistes:

- `admin_pelicules.njk`: mostrarà un llistat amb els títols de les pel·lícules, i enllaces/botons per a editar-les o esborrar-les
- `admin_pelicules_form.njk`: mostrarà un formulari per a editar les dades d'una pel·lícula, bé per a inserir o bé per a editar. Es tindran tots els camps de la pel·lícula.
- Si se li passa com a paràmetre una pel·lícula, s'entendrà que és per a editar-la, i en aqueix cas es deixaran els camps del formulari farcits, i s'afegirà un camp *hidden* per a canviar el mètode del formulari a PUT.
- Es deixa al vostre criteri el determinar com afegir i llevar plataformes de streaming a les pel·lícules. Podeu afegir més vistes addicionals per a això si ho creieu convenient.
- `admin_error.njk`: mostrarà una pàgina d'error amb el missatge que se li indique, o amb el text "Error en l'aplicació" si no rep cap missatge.

Aquestes vistes heretaran d'una plantilla comuna anomenada `admin_base.njk`, que definirà, almenys:

- La capçalera (`head`) amb els estils Bootstrap i un bloc per al títol (`title`) de la pàgina
- El cos (`body`) amb:
- Un menú de navegació comuna definit en la vista `admin_menu.njk`, amb enllaços per a anar a l'arrel de la zona d'administració i per a afegir una nova pel·lícula.
- Un bloc per al contingut de la pàgina

4.3. Part d'autenticació

Pel que fa a l'apartat d'autenticació d'usuaris, només serà necessari definir la vista `auth_login.njk`, amb un formulari de login que s'enviarà per POST a la corresponent ruta de l'enrutador `routes/auth.js`.

5. El servidor principal

El servidor principal deurà:

- Carregar (`require`) les llibreries necessàries
- Carregar (`require`) els enrutadors
- Connectar a una base de dades "FilmEsV3" de MongoDB
- Inicialitzar Express
- Configurar l'autenticació basada en sessions
- Configurar el motor de plantilles Nunjucks, apuntant a la carpeta `views`
- Aplicar el middleware següent (almenys):
- Per a processar dades en format *urlencoded*
- *method-override* per a preprocesar les peticions adequadament i poder utilitzar PUT i DELETE quan corresponga.
- *multer* perquè es pugen les imatges a la carpeta `public/uploads`, anteposant-los com a prefix la data actual.
- Associar l'enrutador `routes/public.js` amb el prefix `/` (ruta arrel)
- Associar l'enrutador `routes/pelicules.js` amb el prefix `/admin`

- Associar l'enrutador `routes/auth.js` amb el prefix `/auth`
- Posar en marxa el servidor Express pel port que vulgueu (per exemple, el port 8080).

6. Lliurament i qualificació

Haureu d'entregar un arxiu ZIP o similar, amb el vostre nom i el prefix "PracT3". Per exemple, si us dieu Laura Pérez, l'arxiu de lliurament haurà de ser `PracT3_Laura_Perez.zip`. Dins, haurà de contindre el projecte **FilmEs_V3** de Node, sense que continga la carpeta `node_modules`.

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: **0,25 puntos**
- Model de dades: **0,5 puntos**, repartits així:
    - Esquema i model per als **usuaris**: **0,1 puntos**
    - Esquemes i model per a les **pelicules**: **0,4 punts**
- Enrutadors: **5 punts**, repartits així:
- Part pública (`routes/public.js`): **1 punt** (0,5 punts la pàgina d'inici i cercador, i 0,5 punts la fitxa de la pel·lícula)
- Part d'administració de pel·lícules (`routes/pelicules.js`): **3 punts** (0,5 punts per cadascuna de les 6 rutes exigides, independentment que després pugueu afegir més rutes si us són necessàries).
- Part d'autenticació d'usuaris: **1 punt** (0,5 punts per al POST de login, i 0,25 punts cadascuna de les altres dues rutes)
- Vistes: **3 punts**, repartits així:
- Vistes de la part pública: **1,25 punts** (0,25 per vista: inici, fitxa de la pel·lícula, pàgina d'error, vista base i menú de navegació)
- Vistes de la part d'administració: **1,5 punts** (0,25 punts per vista: llistat de pel·lícules, error, vista base, menú de navegació, i després 0,5 punts el formulari conjunt d'inserció i edició, independentment d'altres vistes que pugueu necessitar afegir)
- Vistes d'autenticació (formulari de login): **0,25 punts**
- Arxiu principal `index.js` : **0,5 punts**
- Protecció adequada de les rutes d'administració des de l'arxiu `utils/auth.js` : **0,5 punts**
- Claredat i neteja del codi, i ús d'un comentari inicial en cada fitxer font explicant què fa: **0,25 punts**.

Penalitzacions a tindre en compte

- La no eliminació de la carpeta `node_modules` en l'arxiu ZIP de lliurament es penalitzarà amb 2 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 **enrutadors**, s'hauran d'obtindre **almenys 2,5 punts** del total de 5 per a considerar aprovada la pràctica (a més d'arribar a la nota mínima exigible).
- En l'apartat de qualificació de les **vistes**, s'hauran d'obtindre almenys **1,5 punts**, del total de 3, per a considerar aprovada la pràctica (a més d'arribar a la nota mínima exigible).