Nom i Cognoms:	**xavier Palma**
URL Repositori GitHub:	**https://github.com/xavi-p-t/FitesUF2-M06-Xavier-Palma.git**

Objectius: - Ampliar una base de codi conegut amb nova funcionalitat
Instruccions: - Es valorarà la presentació i els comentaris al codi
Criteris d'avaluació: - La puntuació està especificada en cada exercici
Entrega: - Repositori GitHub compartir amb l'usuari **jpala4-ieti** o públic

Repositoris de referència:

- https://github.com/jpala4-ieti/DAM-M06-UF02-Fites-Sequelize-Exemple

Instruccions

Fes una còpia del repositori i desenvolupa la funcionalitat que es demana en cada exercici.

Exercici 0 (2 punts)

En aquest exercici cal implementar els models necessaris per suportar un sistema d'usuaris que podran interactuar amb els vídeos a través de comentaris i valoracions (likes/dislikes).

Encara no cal introduir noves dades, només afegir els models i les relacions. El requisits a complir són:

Un usuari tindrà aquesta informació bàsica associada:

- **id**: Identificador únic (clau primària)
- **username**: Nom d'usuari per identificar-se (ha de ser únic)
- **email**: Correu electrònic (únic, per verificacions i recuperacions)
- **password**: Contrasenya (emmagatzemada en text pla per aquest exercici)
- **nom**: Nom real o complet
- **data_registre**: Data en què es va registrar l'usuari
- **idioma**: Idioma preferit per la interfície

La interacció usuari-vídeo tindrà dues formes principals:

1. **Comentaris**: Un usuari pot publicar múltiples comentaris en un mateix vídeo, i a la vegada un vídeo pot rebre comentaris de diferents usuaris. No s'implementaran respostes a comentaris existents.
2. **Valoracions**: Un usuari pot valorar un vídeo amb un "like" o un "dislike", però mai amb ambdós simultàniament. Cada usuari només pot tenir una valoració activa per vídeo. Valors permesos “like” i “dislike”

Exercici 1 (2 punts)

Prepara els fitxers .csv i realitza les modificacions en l'script de càrrega de dades per tal que la base de dades inclogui, a banda de les dades existents:

- 2 nous usuaris (usuari1, usuari2)
- L'usuari 1 haurà realitzat comentaris a 2 videos diferents, i fet un like i un dislike
- L'usuari 2 haurà realitzat 2 comentaris en un mateix video i fet un dislike.

Exercici 2 (2 punts)

Implementar l'endpoint POST `/api/usuaris/` que permeti crear un nou usuari al sistema, amb validació de dades (només username amb 3 o més caràcters) i gestió adequada duplicats.

Exemple entrada

```
{
  "username": "nomUsuari",  "email": "usuari@exemple.com",
  "password": "contrasenya123", "nom": "Nom Complet", "idioma": "ca"
}
```

Exemple sortida en cas d'èxit (Codi 201)

```
{
  "ok": true,
  "missatge": "Usuari creat amb èxit",
  "resultat": {
    "id": 1,
    "username": "nomUsuari",
    "email": "usuari@exemple.com",
    "nom": "Nom Complet",
    "data_registre": "2025-03-13T12:00:00.000Z",
    "idioma": "ca"
  }
}
```

Sortida en cas d'error de validació (Codi 400):

```
{
  "ok": false,
  "codi": "ERROR_VALIDACIO",
  "missatge": "Les dades proporcionades no compleixen els requisits",
  "detalls": [
    {
      "camp": "username",
      "error": "El nom d'usuari ha de tenir com a mínim 3 caràcters"
    }
  ]}
```

Sortida en cas d'usuari existent (Codi 409)

```
{
  "ok": false,
  "codi": "ERROR_DUPLICAT",
  "missatge": "Ja existeix un usuari amb aquest nom d'usuari o email",
  "detalls": [
    {
      "camp": "email",
      "error": "Aquest email ja està registrat"
    }
  ]}
```

Exercici 3 (2 punts)

Crea un nou punt d'accés a la API GET `/api/usuaris/comentaris/{id_usuari}` que retorni en format JSON la llista de comentaris que ha realitzat junt amb informació del video.

Exemple sortida


```
{
  "ok": true,
  "missatge": "Comentaris de l'usuari obtinguts amb èxit",
  "resultat": [
    {
      "id": 1,
      "text": "Gran tutorial, m'ha ajudat molt amb el meu projecte!",
      "data_creacio": "2023-07-15T18:30:45.000Z",
      "video": {
        "id": 3,
        "titol": "Curso GRATIS de JAVASCRIPT desde cero (con ejercicios)",
        "url_video": "https://www.youtube.com/watch?v=e3x1W9r9-rk",
        "youtuber": {
          "nom_canal": "Hola Mundo"
        }
      }
    },
    {
      "id": 5,
      "text": "Podries fer un tutorial sobre TypeScript?",
      "data_creacio": "2023-08-22T10:15:32.000Z",
      "video": {
        "id": 14,
        "titol": "🔥 CURSO de REACT desde CERO - Para PRINCIPIANTES (2023) 🔥",
        "url_video": "https://www.youtube.com/watch?v=6Jfk8VLb720",
        "youtuber": {
          "nom_canal": "SoyDalto"
        }
      }
    }
  ]
}
```

Exercici 4 (2 punts)

Ara cal documentar les dues noves funcionalitats utilitzant Swagger per tal que es pugui entendre i provar fàcilment l'API.

Tasques a realitzar:

1. **Configuració de Swagger**:
    - Modifica la configuració de Swagger perquè estigui disponible a la URL http://localhost:3010/api-documentacio
    - Assegura't d'actualitzar el port de l'aplicació a 3010 al fitxer .env o al fitxer server.js
2. **Documentació dels nous endpoints**:
    - Documenta l'endpoint POST `/api/usuaris/` per a la creació d'usuaris
    - Documenta l'endpoint GET `/api/usuaris/comentaris/{id_usuari}` que rep l'identificador d'usuari a la URL i retorna la llista de comentaris de l'usuari amb informació del vídeo corresponent
3. **Per cada endpoint, especifica**:
    - Els paràmetres d'entrada (de ruta o cos, segons correspongui)
    - Format de la resposta amb exemples
    - Possibles codis d'estat i els seus significats