

Prueba Técnica- Desarrollo con NodeJs y React

Objetivo:

Desarrollar una API que permita explorar el universo Disney, facilitando conocer y modificar los personajes, y entender las películas en las que estos participaron. Además, deberá exponer la información de manera que cualquier frontend pueda consumirla.

Herramientas a Utilizar:

- NodeJs
- ExpressJs
- Base de datos MYSQL
- React con @mui React Text Field component - Material UI
- Patrón REST para rutas
- Librería Sequelize

Instrucciones de Entrega:

Publicar el código en GitHub y compartir con el usuario johanCarvajalRPA. Se evaluarán los commits y el manejo de ramas.

Realice una segregación en tareas de los requerimientos y proponga una estimación para las mismas. Utilice el README de GitHub para este propósito.

Requerimientos Técnicos:

A continuación, se presenta una serie de requerimientos que sirven como orientación. **No es necesario implementar todos**, seleccione los que considere pertinentes para la entrevista, teniendo en cuenta que se revisará la organización y estructura del código.

Modelado de Base de Datos:

Personaje: Debe incluir:

- Imagen
- Nombre
- Edad
- Peso
- Historia
- Películas o series asociadas

Película o Serie: Debe incluir:

- Imagen
- Título

- Fecha de creación
- Calificación (del 1 al 5)
- Personajes asociados

Género: Debe incluir:

- Nombre
- Imagen
- Películas o series asociadas

Autenticación de Usuarios: Para hacer peticiones a los siguientes endpoints, el usuario necesita un token que obtendrá al autenticarse.

- Endpoints:
 - /auth/login
 - /auth/register

Listado de Personajes: Debe mostrar:

- Imagen
- Nombre
- Endpoint: /characters

Operaciones CRUD de Personajes: Incluir creación, edición (ambas con todos los atributos del modelo) y eliminación soft usando paranoid de Sequelize.

Detalle de Personaje: Mostrar todos los atributos y las películas o series relacionadas.

Búsqueda de Personajes: Permitir buscar por nombre y filtrar por edad, peso o películas/series.

Ejemplos:

- GET /characters?name=nombre
- GET /characters?age=edad
- GET /characters?movies=idMovie

Listado de Películas: Mostrar imagen, título y fecha de creación.

- **Endpoint:** GET /movies
- **Detalle de Película/Serie con Personajes:** Devolver todos los campos de la película o serie y los personajes asociados.

Operaciones CRUD de Películas/Series: Creación, edición y eliminación soft.

Búsqueda de Películas/Series: Permitir buscar por título, filtrar por género y ordenar por fecha de creación (ascendente o descendente). Ejemplos:

- GET /movies?name=nombre
- GET /movies?genre=idGenero
- GET /movies?order=ASC | DESC