

Rosa María Rodríguez Lledó

Ilya Slyusarchuk

Introducción

Este proyecto trata de un sistema de gestión de tareas inspirado en Trello.

Recursos y sus campos

Usuario:

- nombre
- email
- contraseña
- id

Tablero:

- nombre
- miembros
- creadoPor
- etiquetas
- id

Columna:

- titulo
- tablero
- id

Tarjeta:

- nombre
- descripción
- fechaVencimiento
- columna
- asignadoA
- etiquetas
- id

Etiqueta:

- nombre
- tablero
- id

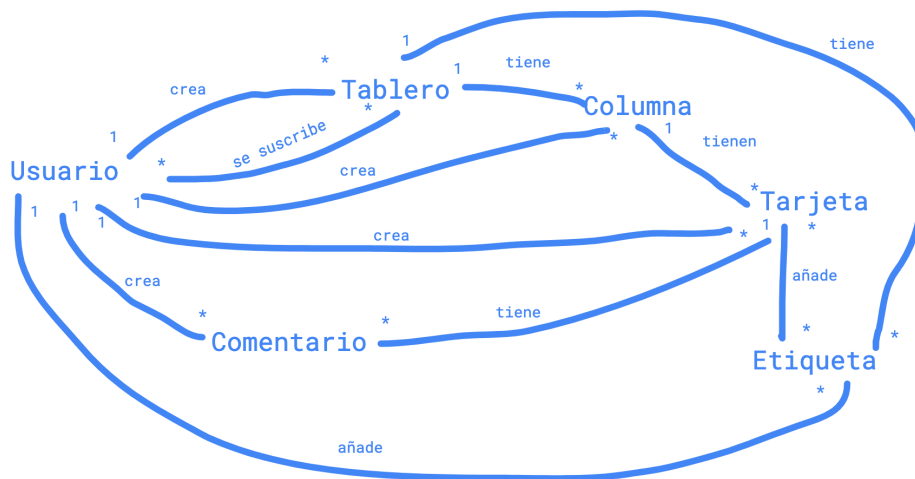
Comentario:

- texto
- escritoPor
- id

Con respecto a la práctica 2, estos son los cambios introducidos en los recursos:

- Se añade un campo id a todos ellos para facilitar el acceso en la base de datos
- Se añade una referencia a tablero desde las etiquetas, puesto que las etiquetas necesariamente deben pertenecer a un tablero.

Diagrama con las relaciones entre los recursos



Listado de casos de uso

- Un usuario no autenticado sólo puede ver los tableros públicos y toda su información, pero no puede crear ni modificar ni suscribirse.
- Un usuario no autenticado puede registrarse e iniciar sesión.
- Un usuario autenticado puede crear un tablero.
- Un usuario autenticado puede crear varias columnas en un mismo tablero.
- Un usuario autenticado puede crear una o varias tarjetas y asignarlas a una columna.
- Un usuario autenticado puede mover una tarjeta de una columna a otra.
- Un usuario autenticado puede añadir etiquetas a la tarjeta.
- Un usuario autenticado puede añadir comentarios a la tarjeta.
- Un usuario autenticado puede cambiar los datos de la tarjeta (título, descripción, fecha límite, participantes...)
- Un usuario autenticado puede crear etiquetas.
- Un usuario autenticado puede eliminar etiquetas, tarjetas, columnas y tableros.
- Un usuario autenticado puede aceptar invitaciones de otros usuarios a ser miembros (suscribirse) a su tablero
- Un usuario autenticado puede suscribirse a un tablero público de otro usuario si éste acepta su invitación.
- Un usuario miembro de un tablero puede modificar el tablero igual que el creador.
- Un usuario autenticado que no sea miembro de un tablero puede ver toda su información, pero no puede crear ni modificar. Puede pedir suscribirse para convertirse en miembro.

Especificación de llamadas al API

- Formato de la petición HTTP
 - La URL
 - El método HTTP (GET, POST, PUT o DELETE)
 - Si el cliente envía datos al servidor cómo lo hace: en parámetros HTTP o en el cuerpo de la petición
 - Cabeceras HTTP que debe enviar el cliente, si las hay
- Formato de la respuesta HTTP
 - Código de estado si éxito
 - Códigos de estado de los errores más habituales
 - Cabeceras HTTP que debe enviar el servidor, si las hay
 - Contenido del cuerpo de la respuesta en caso de éxito
 - Contenido del cuerpo de la respuesta en caso de error, para al menos los 2-3 errores que creéis que pueden ser más habituales
- Petición HTTP: **CREAR TABLERO**

- /tableros
 - Método POST
 - Los datos se envían en el cuerpo de la petición para que no sean visibles desde la URL
 - Content-Type=application/json y un token de autenticación para saber qué usuario está creando el tablero
- Formato de la respuesta HTTP
 - Código de estado si éxito: 201
 - Códigos de estado de los errores más habituales: 401, 400, 500
 - Cabeceras HTTP que debe enviar el servidor, si las hay: Location={url_web}/tableros/{id} si éxito
 - Contenido del cuerpo de la respuesta en caso de éxito: se podría devolver el objetocreado
 - Contenido del cuerpo de la respuesta en caso de error, para al menos los 2-3 erroresque creéis que pueden ser más habituales: 401 {"mensaje": "Credenciales incorrectas"}, 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "Error del servidor"}
- Petición HTTP: **VER TABLERO**
 - /tableros/{id}
 - Método GET
 - Si el cliente envía datos al servidor cómo lo hace: en parámetros HTTP o en elcuerpo de la petición: no es necesario enviar nada
 - No hacen falta cabeceras
- Formato de la respuesta HTTP
 - Código de estado si éxito: 200
 - Códigos de estado de los errores más habituales: 404, 500
 - Cabeceras HTTP que debe enviar el servidor: ninguna
 - Contenido del cuerpo de la respuesta en caso de éxito: {"tablero": {"nombre": "mitablero", "miembros": {...}, "creadoPor": {"nombre": "rosa" ...}}}
 - Contenido del cuerpo de la respuesta en caso de error, para al menos los 2-3 erroresque creéis que pueden ser más habituales: 404 {"mensaje": "Tablero no encontrado"}, 500 {"mensaje": "Error del servidor"}
- Formato de la petición HTTP: **BORRAR TABLERO**
 - La URL /tableros/{id}
 - El método HTTP DELETE
 - Si el cliente envía datos al servidor cómo lo hace: en parámetros HTTP o en elcuerpo de la petición: no es necesario enviar nada
 - Cabeceras HTTP que debe enviar el cliente: cabecera de autenticación
- Formato de la respuesta HTTP
 - Código de estado si éxito: 204
 - Códigos de estado de los errores más habituales: 401, 403, 404, 500
 - Cabeceras HTTP que debe enviar el servidor, si las hay: ninguna
 - Contenido del cuerpo de la respuesta en caso de éxito: se podría devolver el objetoeliminado
 - Contenido del cuerpo de la respuesta en caso de error, para al menos los 2-3 erroresque creéis que pueden ser más habituales: 401 {"mensaje": "Credenciales incorrectas"}, 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "Error del servidor"}
- Formato de la petición **AÑADIR MIEMBROS A UN TABLERO**
 - La URL /tableros/{id}/usuarios
 - El método HTTP POST
 - Si el cliente envía datos al servidor cómo lo hace: en el cuerpo de la petición
 - Cabeceras HTTP que debe enviar el cliente: Content-Type=application/json y untoken de autenticación

- Formato de la respuesta HTTP
 - Código de estado si éxito: 201
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: nada
 - Contenido del cuerpo de la respuesta en caso de éxito: se podría devolver todos los miembros, incluyendo el que sea a crear
 - Contenido del cuerpo de la respuesta en caso de error: 401 {"mensaje": "Credenciales incorrectas"}, 403 {"mensaje": "No tienes permisos"} 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "Error del servidor"}
- Formato de la petición **CREAR COLUMNA EN UN TABLERO**
 - La URL /tableros/{id}/columnas
 - El método HTTP POST
 - Si el cliente envía datos al servidor cómo lo hace: en el cuerpo de la petición
 - Cabeceras HTTP que debe enviar el cliente: Content-Type=application/json y un token de autenticación
- Formato de la respuesta HTTP
 - Código de estado si éxito: 201
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: la url de la columna creada
 - Contenido del cuerpo de la respuesta en caso de éxito: se podrían devolver todas las columnas del tablero
 - Contenido del cuerpo de la respuesta en caso de error: 401 {"mensaje": "Credenciales incorrectas"}, 403 {"mensaje": "No tienes permisos"} 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "Error del servidor"}
- Formato de la petición **CREAR TARJETA**
 - La URL /tableros/{id}/columnas/{id}/tarjetas
 - El método HTTP POST
 - Si el cliente envía datos al servidor cómo lo hace: en el cuerpo de la petición
 - Cabeceras HTTP que debe enviar el cliente: Content-Type=application/json y un token de autenticación
- Formato de la respuesta HTTP
 - Código de estado si éxito: 201
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: nada
 - Contenido del cuerpo de la respuesta en caso de éxito: se podrían devolver todas las tarjetas de esa columna
 - Contenido del cuerpo de la respuesta en caso de error: 401 {"mensaje": "Credenciales incorrectas"}, 403 {"mensaje": "No tienes permisos"} 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "Error del servidor"}
- Formato de la petición **MODIFICAR TARJETA (O MOVER DE COLUMNA)**
 - La URL /tableros/{id}/columnas/{id}/tarjetas/{id}
 - El método HTTP PATCH
 - Si el cliente envía datos al servidor cómo lo hace: en el cuerpo de la petición
 - Cabeceras HTTP que debe enviar el cliente: Content-Type=application/json y un token de autenticación
- Formato de la respuesta HTTP
 - Código de estado si éxito: 204
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: si se mueve de columna se

devuélvela nueva URL de esa tarjeta y si lo que se modifica es algún campo, no se devuelve nada

- Contenido del cuerpo de la respuesta en caso de éxito: la tarjeta modificada
 - Contenido del cuerpo de la respuesta en caso de error: 401 {"mensaje": "Credenciales incorrectas"}, 403 {"mensaje": "No tienes permisos"} 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "Error del servidor"}
- Formato de la petición **BORRAR TARJETA**
 - La URL /tableros/{id}/columnas/{id}/tarjetas/{id}
 - El método HTTP DELETE
 - Si el cliente envía datos al servidor cómo lo hace: no hace falta enviar nada
 - Cabeceras HTTP que debe enviar el cliente: un token de autenticación
 - Formato de la respuesta HTTP
 - Código de estado si éxito: 204
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: ninguna
 - Contenido del cuerpo de la respuesta en caso de éxito: se podría borrar el objeto eliminado
 - Contenido del cuerpo de la respuesta en caso de error: 401 {"mensaje": "Credenciales incorrectas"}, 403 {"mensaje": "No tienes permisos"} 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "Error del servidor"}
 - Formato de la petición **ESCRIBIR COMENTARIO**
 - La URL /tableros/{id}/columnas/{id}/tarjetas/{id}/comentarios
 - El método HTTP POST
 - Si el cliente envía datos al servidor cómo lo hace: en el cuerpo de la petición
 - Cabeceras HTTP que debe enviar el cliente: Content-Type=json y un token de autenticación
 - Formato de la respuesta HTTP
 - Código de estado si éxito: 201
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: una cabecera Location con la url del recurso creado
 - Contenido del cuerpo de la respuesta en caso de éxito: devolver el recurso creado
 - Contenido del cuerpo de la respuesta en caso de error: 401 {"mensaje": "Credenciales incorrectas"}, 403 {"mensaje": "No tienes permisos"} 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "Error del servidor"}
 - Formato de la petición **CREAR ETIQUETA**
 - La URL /tableros/{id}/etiquetas
 - El método HTTP POST
 - Si el cliente envía datos al servidor cómo lo hace: en el cuerpo de la petición
 - Cabeceras HTTP que debe enviar el cliente: Content-Type=json y un token de autenticación
 - Formato de la respuesta HTTP
 - Código de estado si éxito: 201
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: una cabecera Location con la url del recurso creado
 - Contenido del cuerpo de la respuesta en caso de éxito: devolver el recurso creado
 - Contenido del cuerpo de la respuesta en caso de error: 401 {"mensaje": "Credenciales incorrectas"}, 403 {"mensaje": "No tienes permisos"} 400

{“mensaje”:
“Faltan campos necesarios”}, 500 {“mensaje”:“Error del servidor”}

- Petición HTTP: **VER COLUMNA**
 - /tableros/:idtablero/columnas/:idcolumna
 - Método GET
 - Si el cliente envía datos al servidor cómo lo hace: en parámetros HTTP o en el cuerpo de la petición: no es necesario enviar nada
 - No hacen falta cabeceras
- Formato de la respuesta HTTP
 - Código de estado si éxito: 200
 - Códigos de estado de los errores más habituales: 400, 404, 500
 - Cabeceras HTTP que debe enviar el servidor: ninguna
 - Contenido del cuerpo de la respuesta en caso de éxito: {“id”:“1”, “titulo”:“Backlog”}
 - Contenido del cuerpo de la respuesta en caso de error, para al menos los 2-3 errores que creéis que pueden ser más habituales: 404 {“mensaje”: “La columna no existe”}, 500 {“mensaje”:“Error del servidor”}, 400 {“mensaje”: “El id no es un número”}
- Petición HTTP: **VER TARJETA**
 - /tableros/:idtablero/columnas/:idcolumna/tarjetas/:idtarjeta
 - Método GET
 - Si el cliente envía datos al servidor cómo lo hace: en parámetros HTTP o en el cuerpo de la petición: no es necesario enviar nada
 - No hacen falta cabeceras
- Formato de la respuesta HTTP
 - Código de estado si éxito: 200
 - Códigos de estado de los errores más habituales: 400, 404, 500
 - Cabeceras HTTP que debe enviar el servidor: ninguna
 - Contenido del cuerpo de la respuesta en caso de éxito: {“id”:“1”, “titulo”:“Backlog”}
 - Contenido del cuerpo de la respuesta en caso de error, para al menos los 2-3 errores que creéis que pueden ser más habituales: 404 {“mensaje”: “La tarjeta no existe”}, 500 {“mensaje”:“Error del servidor”}, 400 {“mensaje”: “El id no es un número”}
- Petición HTTP: **VER TABLEROS**
 - /tableros
 - Método GET
 - Si el cliente envía datos al servidor cómo lo hace: en parámetros HTTP o en el cuerpo de la petición: no es necesario enviar nada
 - No hacen falta cabeceras
- Formato de la respuesta HTTP
 - Código de estado si éxito: 200
 - Códigos de estado de los errores más habituales: 400, 404, 500
 - Cabeceras HTTP que debe enviar el servidor: ninguna
 - Contenido del cuerpo de la respuesta en caso de éxito: {“id”:“1”, “nombre”:“MADS”, “user_id”:1}, {“id”:2, “nombre”:“TAES”, “user_id”=1}
 - Contenido del cuerpo de la respuesta en caso de error: 401 {“mensaje”: “Credenciales incorrectas”}, 403 {“mensaje”:“No tienes permisos”} 400 {“mensaje”: “Faltan campos necesarios”}, 500 {“mensaje”:“Error del servidor”}
 -
- Petición HTTP: **AÑADIR IMAGEN AL TABLERO**
 - /tableros/:id/background
 - Método POST
 - Si el cliente envía datos al servidor cómo lo hace: en el cuerpo de la petición

- Cabeceras HTTP que debe enviar el cliente: Content-Type=json y un token de autenticación
- Formato de la respuesta HTTP
 - Código de estado si éxito: 200
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: una cabecera Location con la url del recurso creado
 - Contenido del cuerpo de la respuesta en caso de éxito: los campos del archivo creado y un mensaje de éxito
 - Contenido del cuerpo de la respuesta en caso de error, para al menos los 2-3 errores que creéis que pueden ser más habituales: 404 {"mensaje": "El ítem no existe"}, 500 {"mensaje": "Error del servidor"}
 -
- Formato de la petición **MODIFICAR TABLERO**
 - La URL /tableros/:idtablero/columnas/:idcolumna/tarjetas/:idtarjeta
 - El método HTTP PUT
 - En el cuerpo de la petición se envían los campos "nombre" y "user_id".
 - Cabeceras HTTP que debe enviar el cliente: Content-Type=application/json y un token de autenticación
- Formato de la respuesta HTTP
 - Código de estado si éxito: 200
 - Códigos de estado de los errores más habituales: 401, 403, 400, 500
 - Cabeceras HTTP que debe enviar el servidor: si se modifica devuelve la URL de ese tablero
 - Contenido del cuerpo de la respuesta en caso de éxito: el tablero modificado
 - Contenido del cuerpo de la respuesta en caso de error: 401 {"mensaje": "Credenciales incorrectas"}, 403 {"mensaje": "No tienes permisos"}, 400 {"mensaje": "Faltan campos necesarios"}, 500 {"mensaje": "No se ha podido actualizar"}

Llamadas al API implementadas en esta entrega

Petición get de un recurso sabiendo su id:

- /tableros/:id (VER TABLERO)
- /tableros/:idtablero/columnas/:idcolumna (VER COLUMNA)
- /tableros/:idtablero/columnas/:idcolumna/tarjetas/:idtarjeta (VER TARJETA)

Petición get de una colección:

- /tableros (VER TABLEROS)

Petición delete:

- /tableros/:id (BORRAR TABLERO)

Petición post:

- /tableros (CREAR TABLERO)
- /tableros/:id/background (AÑADIR IMAGEN AL TABLERO)

Petición patch:

- /tableros/:idtablero/columnas/:idcolumna/tarjetas/:idtarjeta (MOVER DE COLUMNA)

Petición put:

- /tableros/:idtablero (MODIFICAR TABLERO)

Requisitos adicionales

La persistencia de datos se obtiene mediante una base de datos sqlite3. En la carpeta /db proporcionada se incluyen las migraciones y las semillas necesarias. Para ejecutarlos:

```
npm run migrate
```

```
npm run seed
```

Además, se pueden añadir archivos binarios en el método post /tableros/:id/background para añadir una imagen de fondo a un tablero, que se guarda en la carpeta uploads.

También hemos creado unos tests: Ejercicio_2.postman_collection.json

Por último, hemos implementado el paginado en la colección /tableros.