**Como usar el Backend**

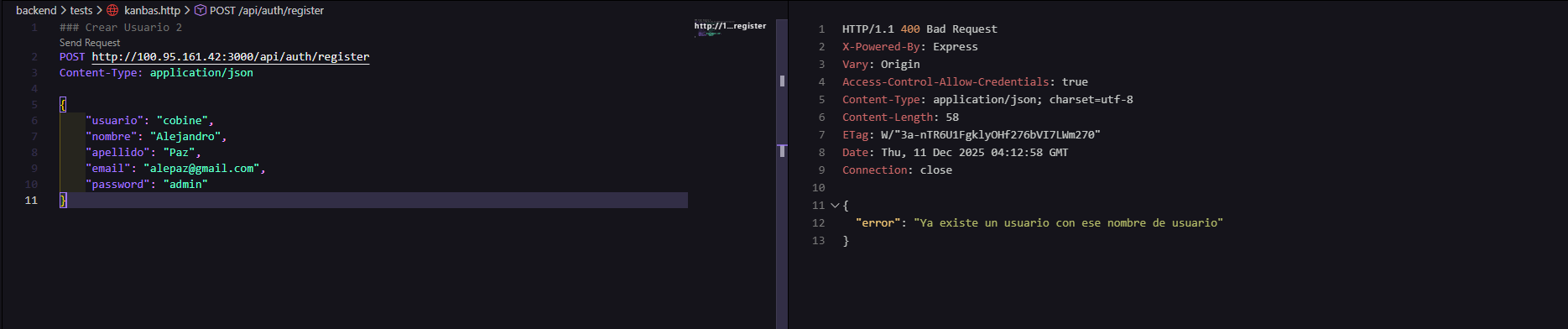
**Registro**



En la siguiente imagen puedes ver lo que necesita recibir el backend para registrar al usuario y lo que devuelve es el JSON que está entre corchetes.



En el caso que alguien quiera usar el mismo usuario le sale el siguiente error.

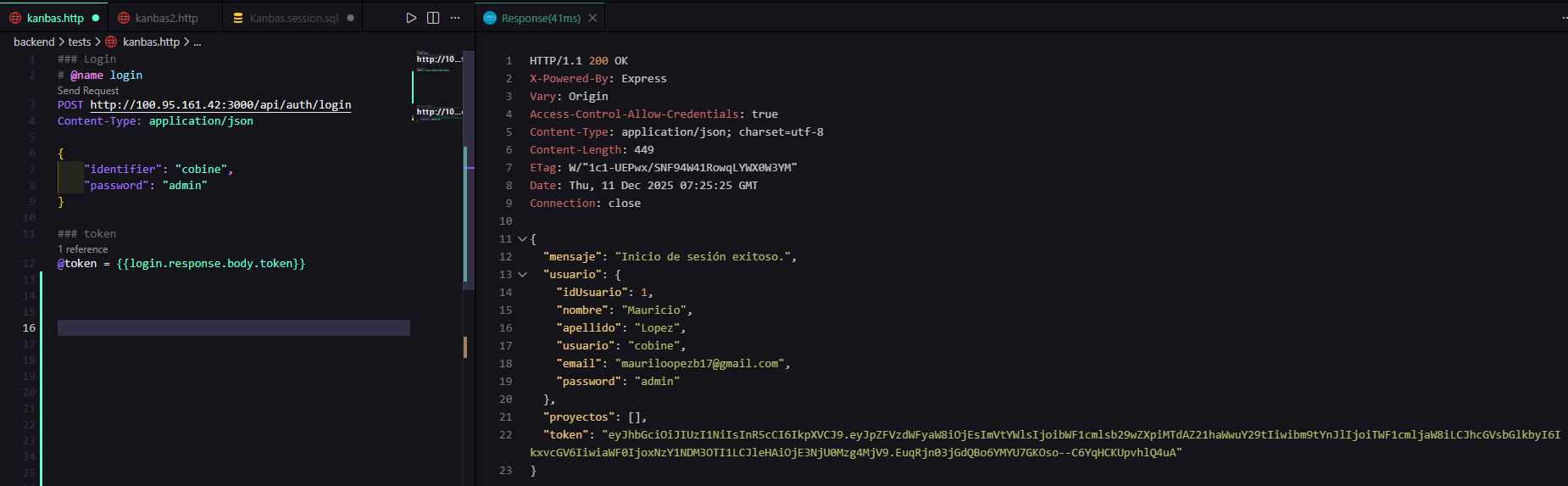


En el caso que alguien quiera usar el mismo email le sale el siguiente error.  


**Login**

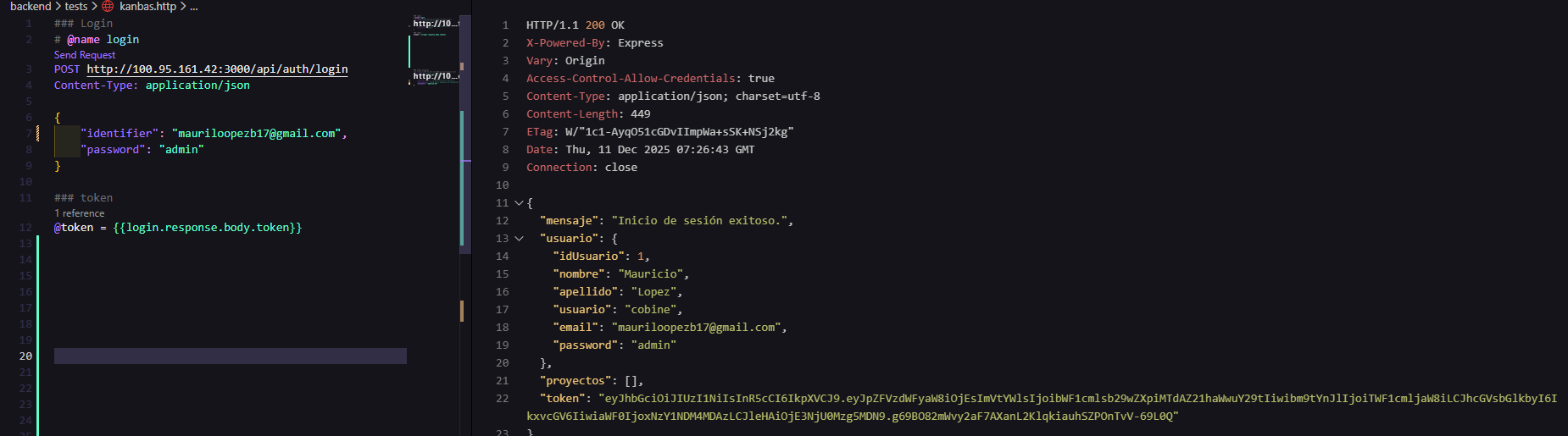


En la siguiente imagen puedes ver lo que necesita el backend para verificar el usuario:



Y esto es lo que devuelve. Ahí se puede ver todos los atributos de usuario, si tuviese proyectos y lo más importante el token JWT para mantener la sesión. Cabe aclarar que las partes de “# @name login” y “@token = {{login.response.body.token}}” no debes colocarlas tú, esa es la forma en que se hacen pruebas. Tu debes agarrar el JSON y ese token guardarlo en un localStorage o sessionStorage.

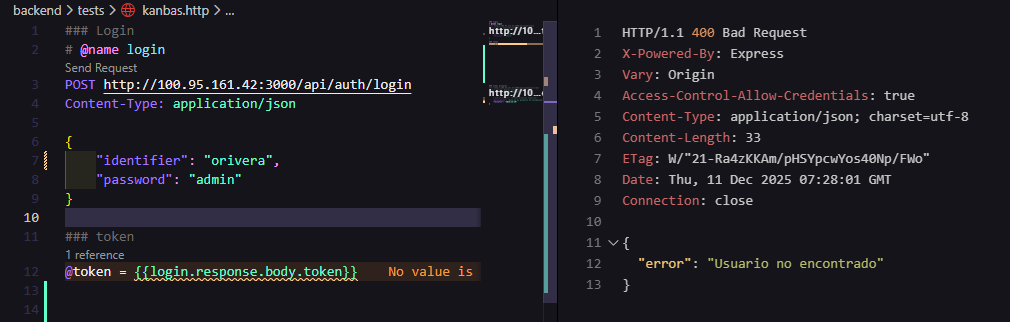
Ahora bien, esto también funciona cuando inicias sesión con usuario (como ya te mostré) y con correo:



Pero qué pasa si pones mal la contraseña? Pues, te sale:



Eso es lo que te devuelve un error de contraseña equivocada y ahora qué pasa si pongo un correo o usuario que no tenga una cuenta registrada:

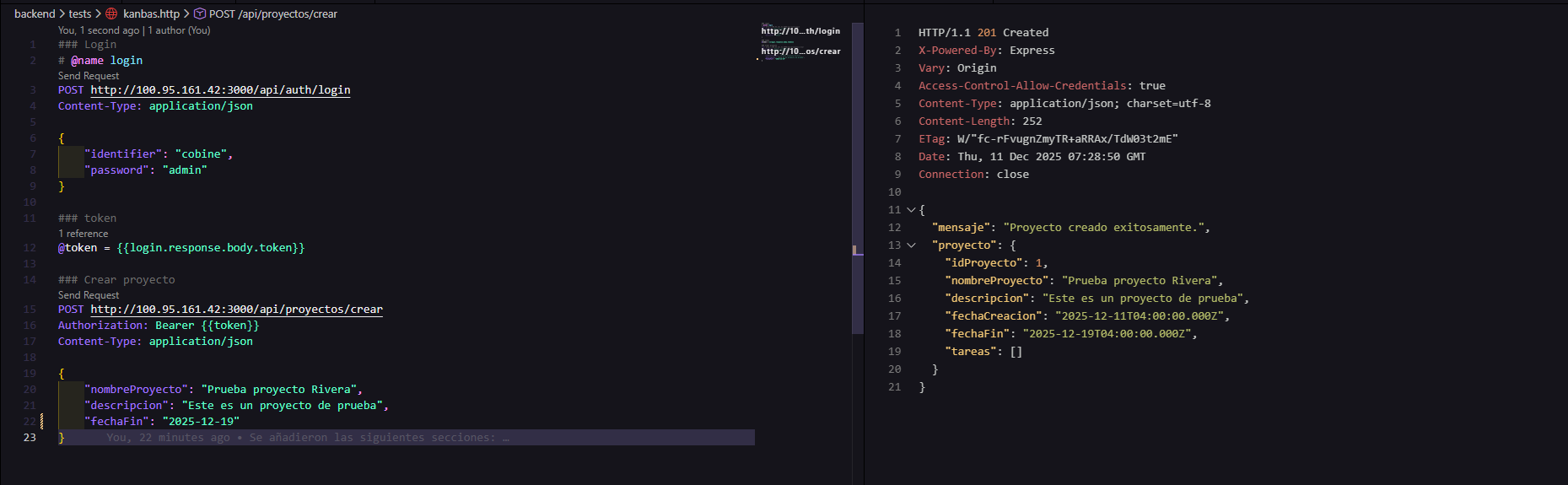


Te sale el error de Usuario no encontrado.

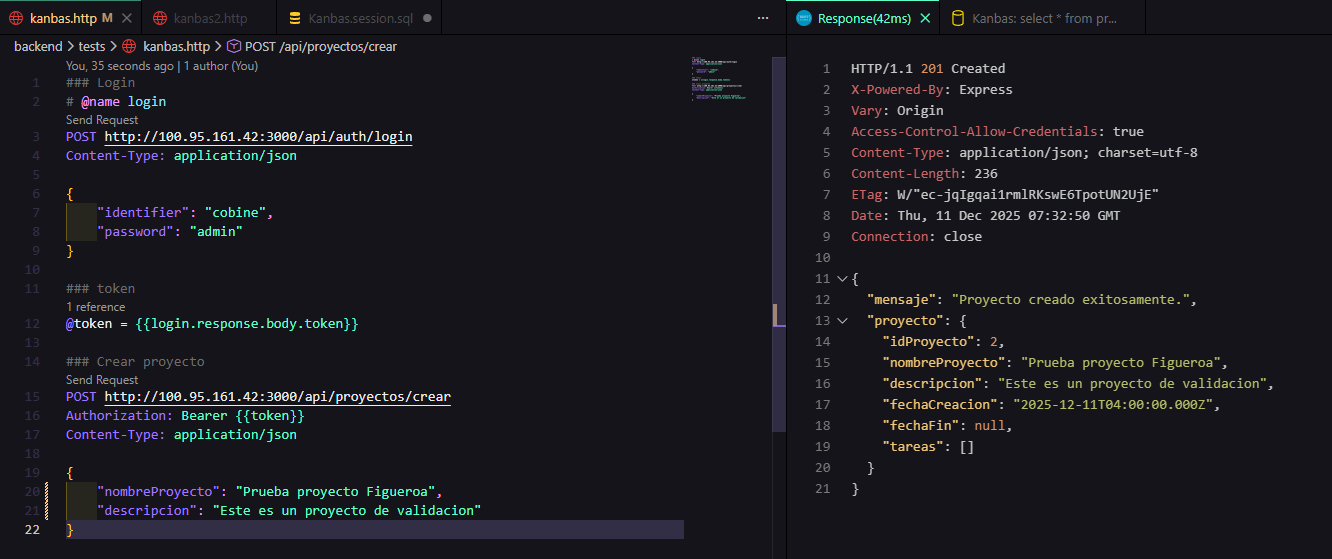
**Para crear proyecto**



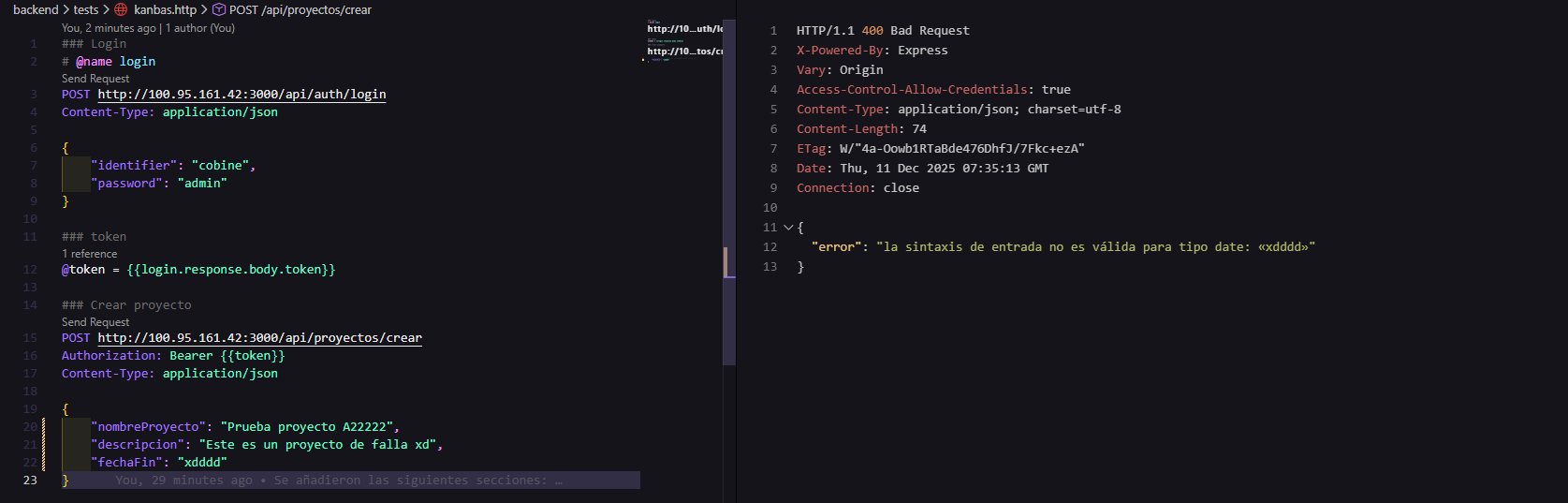
Para crear un proyecto debes:



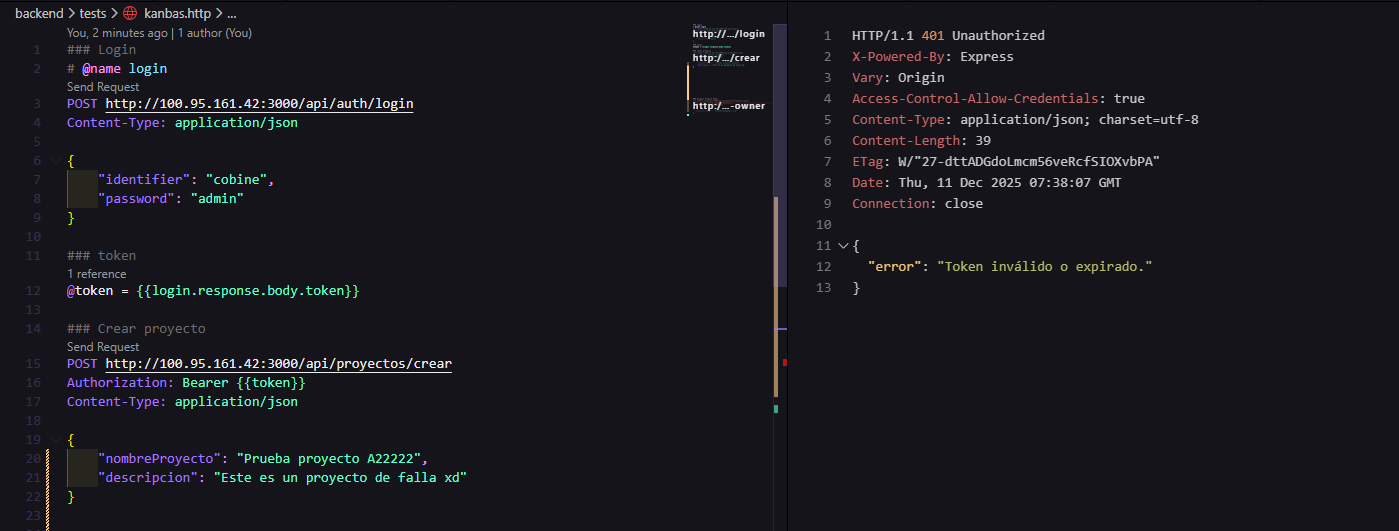
Ahí puedes apreciar los atributos necesarios y lo que te devuelve. Sin embargo, fecha limite es opcional, que pasa si no lo ponemos:

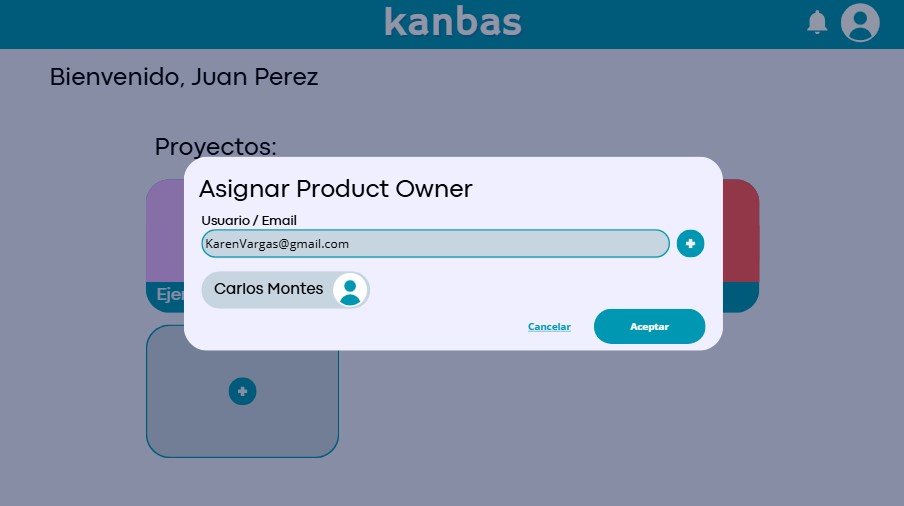


Digamos el hipotético caso que en fecha limite enviemos otra cosa que no sea fecha, esto pasa:

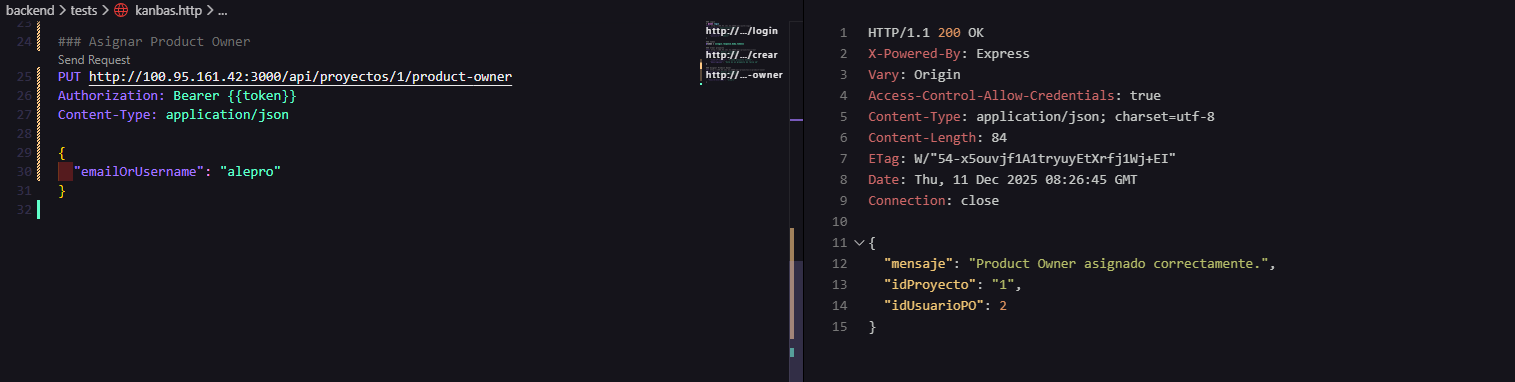


Ahora bien, digamos que se nos cerro la sesión porque expiró el Token, sale esto:

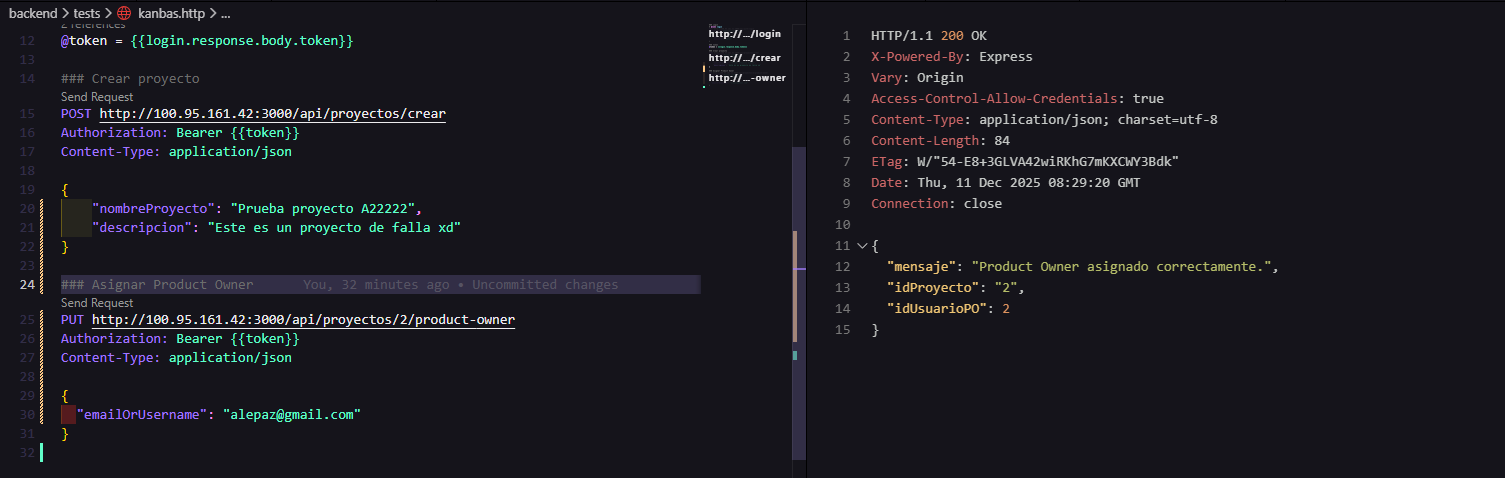




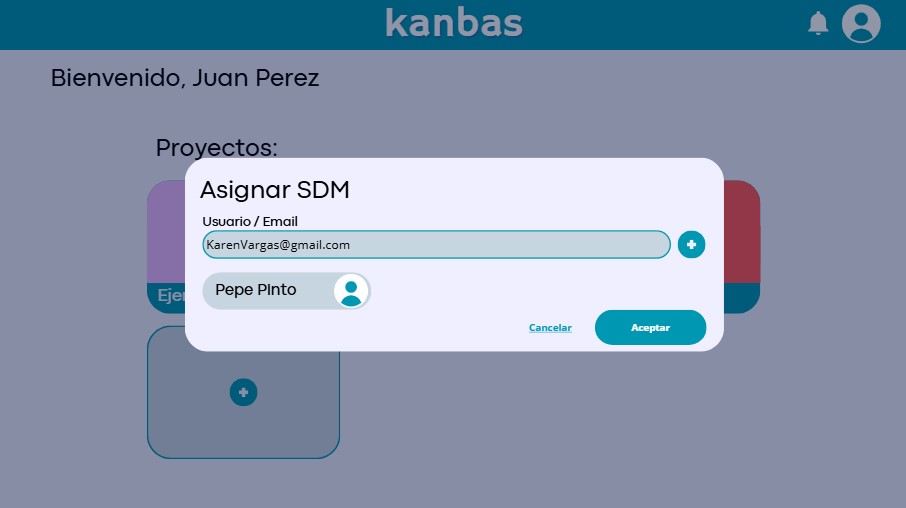
Ahora en la siguiente ventana para asignar product owner se debe hacer lo siguiente:



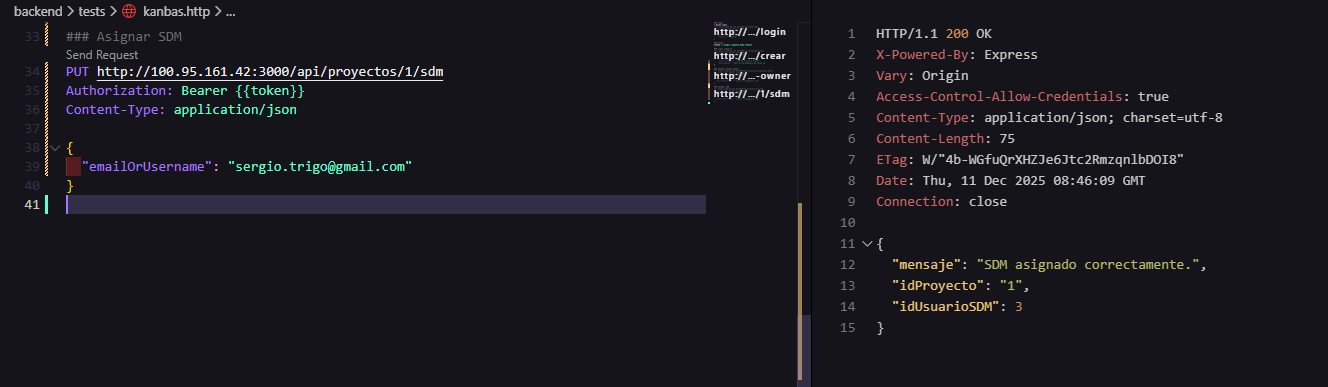
Como puedes ver se puede hacer por usuario, pero tambien con correo:



Funciona, pero aca hay una particulariedad. Al momento de poner la URL “…./api/proyectos/ {idProyecto} /product-owner”. Cuando crees el proyecto, guarda el idProyecto, porque lo vas a necesitar para esto. Lo mismo pasa con el SDM, pero eso lo veras en un momento. Otra cosa, recuerda que solo se puede añadir un PO, entonces una vez selecciones una persona desactiva el boton de +, si envias otra persona al usar un UPDATE a la columna de PO, entonces simplemente se va actualizar y se va sobreescribir al ultimo PO que mandes.



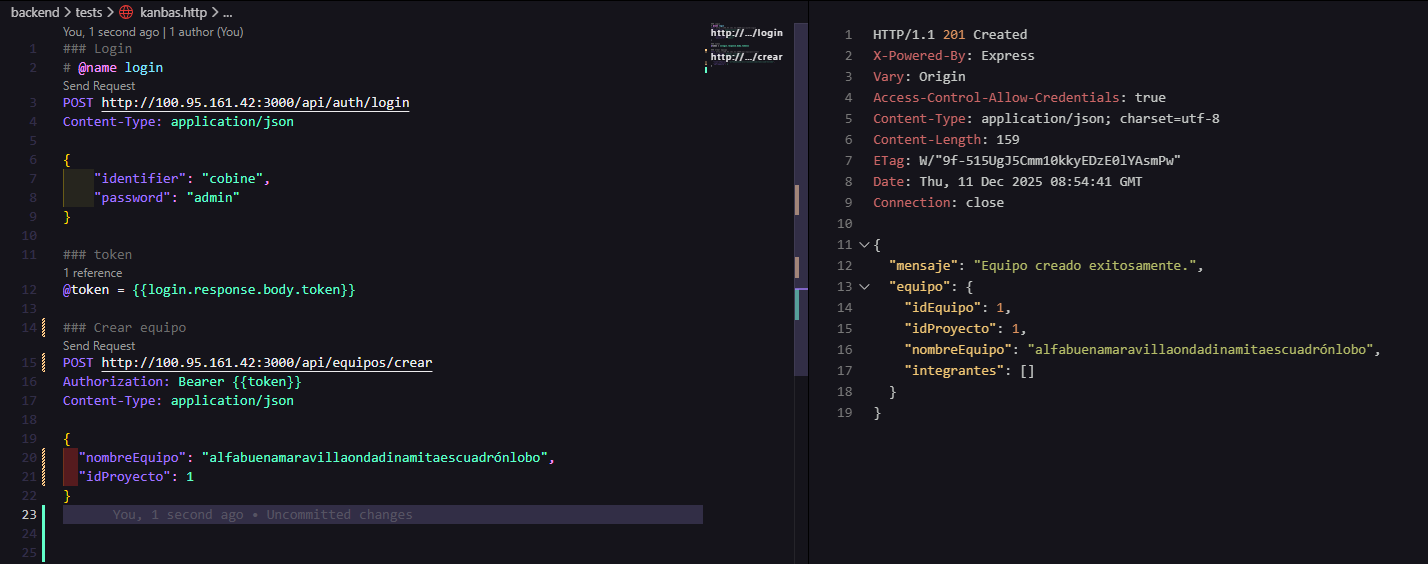
De la misma forma que con el PO, se va asignar el SDM de la siguiente forma:



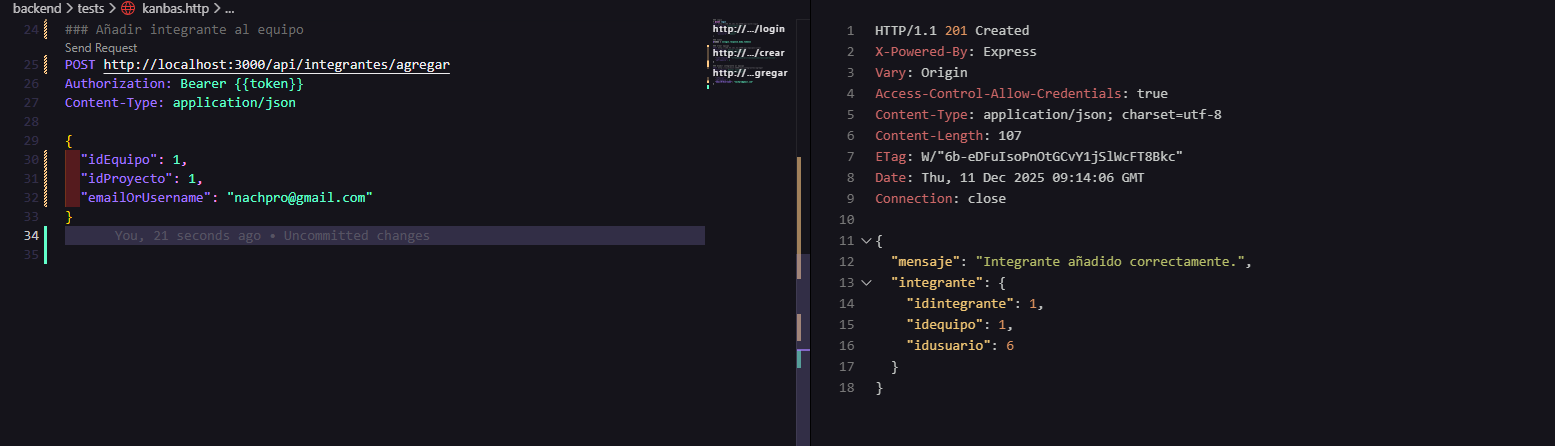
Al igual que con el PO se puede hacer por usuario o por email. Además, no olvides que aca solo podemos hacer una asignacion entonces desactiva el boton de +, porque sino se sobreescribira y pues ahí es ggs.



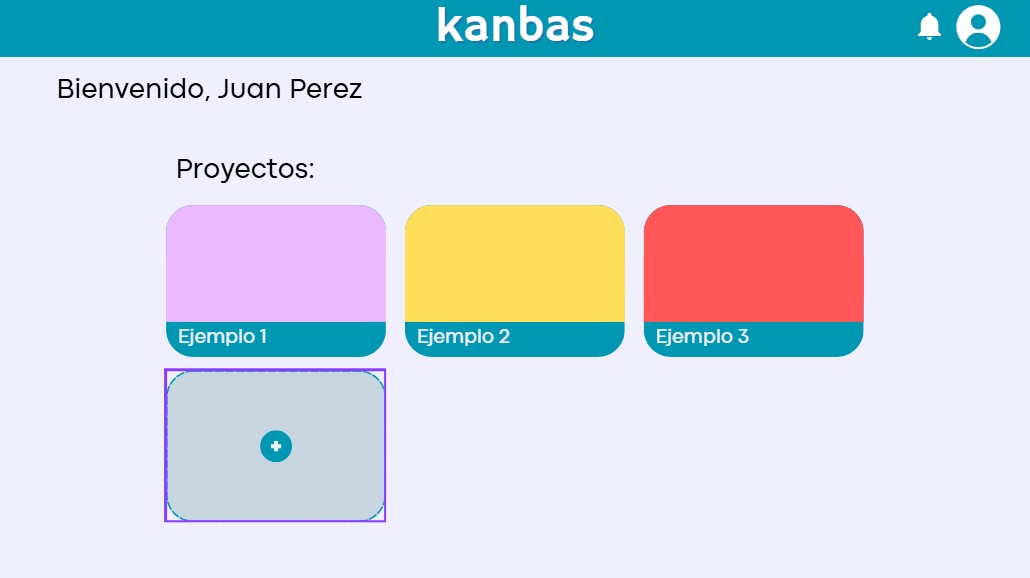
Ahora para crear equipos o agregarlos hacemos lo siguiente:



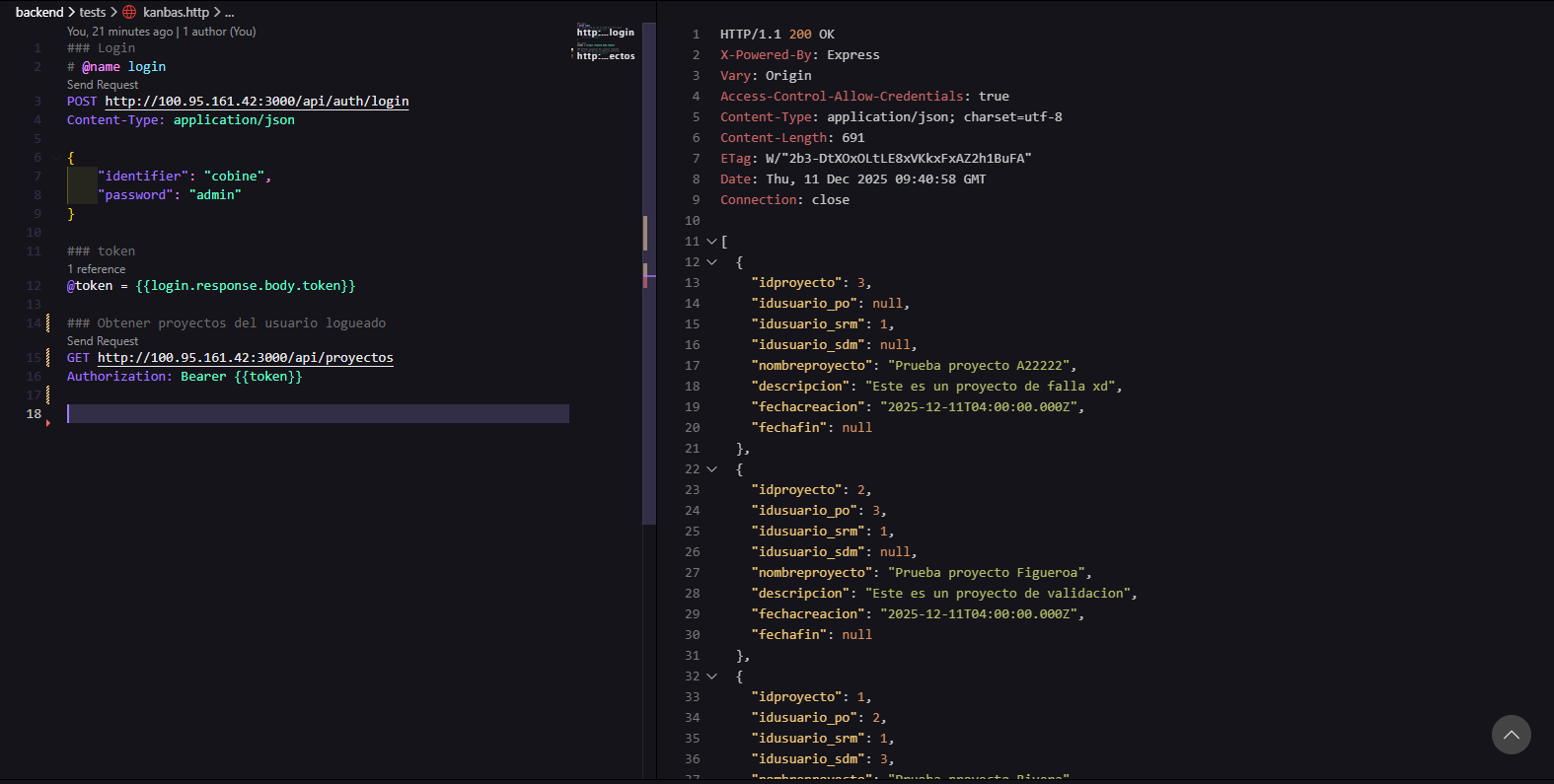
Para esto lo mismo, no olvidar guardar el idProyecto que se necesita ingresar. Ahora añadir integrantes debes mandar lo siguiente:



Aquí es importante que guardes el idEquipo e idProyecto.

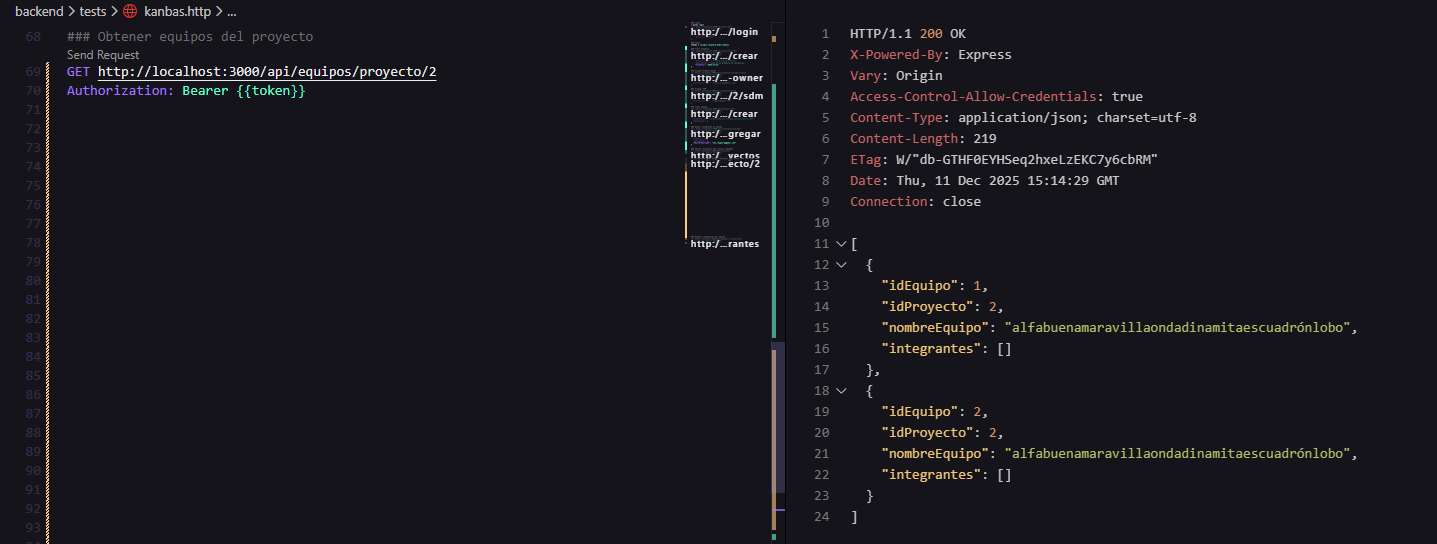


Ahora, si quieres obtener todos los proyectos para armar el tablero de nuevo o algo así, puede hacer lo siguiente:





Para agregar tarea tenemos que hacer varias cositas antes. Antes debemos poder escoger solo los equipos disponibles en el proyecto, así que para eso tenemos lo siguiente:



Cabe aclarar una cosa, al igual que antes, debes poner el idProyecto: [http://localhosto:3000/api/equipos/proyecto/{idProyecto}](http://localhosto:3000/api/equipos/proyecto/%7bidProyecto%7d). Una vez hecho eso, en base al equipo que hayas escogido recién te da las opciones de integrantes que están acá. Por lo que debes ejecutar esto para que salga los integrantes de cada equipo:

Texto

El contenido generado por IA puede ser incorrecto.

Lo mismo que antes, para poder ejecutarlo necesitas poner el idEquipo correspondiente de la siguiente forma [http://localhost:3000/api/equipos/{idEquipo}/integrantes](http://localhost:3000/api/equipos/%7bidEquipo%7d/integrantes) y listo, con eso tienes las lista de todos los integrantes de los equipos. Finalmente, para crear la tarea debes hacer:

Texto

El contenido generado por IA puede ser incorrecto.

Para esto es importante que hayas guardado los ids del proyecto y de cada integrante (no el idUsuario, sino el idIntegrante) para poner en sus respectivos atributos.



Pasando a editar tarea, tenemos que primero obtener la tarea seleccionada de la siguiente forma:

Texto

El contenido generado por IA puede ser incorrecto.

Cabe aclarar que al igual que antes la URL es [http://localhost:3000/api/tareas/{idTarea}/{idProecto}](http://localhost:3000/api/tareas/%7bidTarea%7d/%7bidProecto%7d). Luego vamos a reutilizar lo de Equipo e Integrantes de arriba, en caso que quieran editarlo, no?

Texto

El contenido generado por IA puede ser incorrecto.

Texto

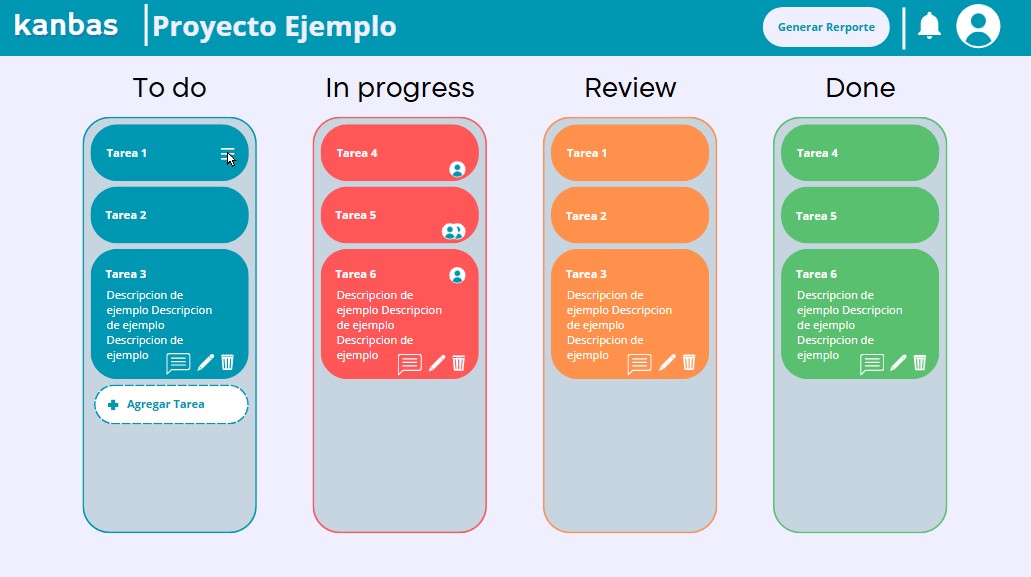
El contenido generado por IA puede ser incorrecto.

Finalmente, para poder editar la tarea tenemos que hacer lo siguiente:

Texto

El contenido generado por IA puede ser incorrecto.

Cabe aclarar que la url es [http://localhost:3000/api/tareas/editar/{idTarea}/{idProyecto}](http://localhost:3000/api/tareas/editar/%7bidTarea%7d/%7bidProyecto%7d). Entonces ten en cuenta que al seleccionar la tarea debes guardar los ids de tarea e idProyecto. Luego lo de Equipo e integrante funciona de la misma forma que en crear.

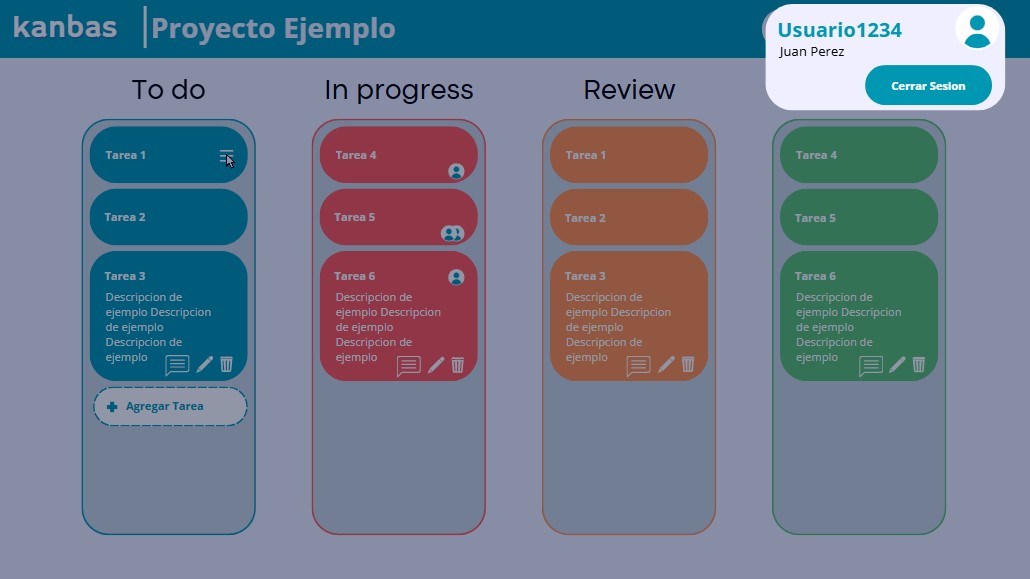


Para el tema del dashboard y obtener todas las tareas tenemos lo siguiente:

Texto

El contenido generado por IA puede ser incorrecto.

Si te das cuenta tenemos la URL: [http://localhost:3000/api/tareas/proyecto/{idProyecto}](http://localhost:3000/api/tareas/proyecto/%7bidProyecto%7d) de la misma forma que hicimos antes, necesitamos el idProyecto. Ahora respecto a la salida, fíjate que hay un atributo “todo” y de ahí empieza todo, lo que te devuelve son los 4 estados y dentro de cada estado una lista de tareas según su estado, para facilitarte las cosas, además están los usuarios asignados por si los necesitaras.



Para obtener el perfil del usuario simplemente haz esto:

Texto

El contenido generado por IA puede ser incorrecto.



Para el nuevo mensaje tenemos las siguientes alternativas, la primera es que comencemos consultando todos los usuarios a los cuales podemos mandar mensajes:

Texto

El contenido generado por IA puede ser incorrecto.

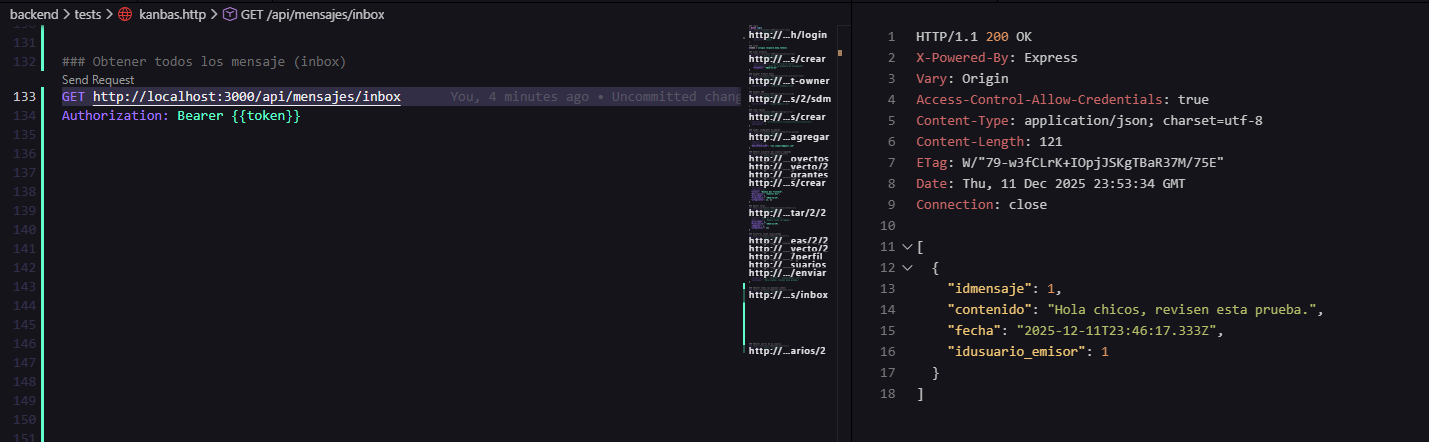
Y luego al momento de crear el mensaje y enviarlo tengamos la posibilidad de poner el id. La otra opción es que en vez del id, pongamos directamente su user o correo.

Texto

El contenido generado por IA puede ser incorrecto.

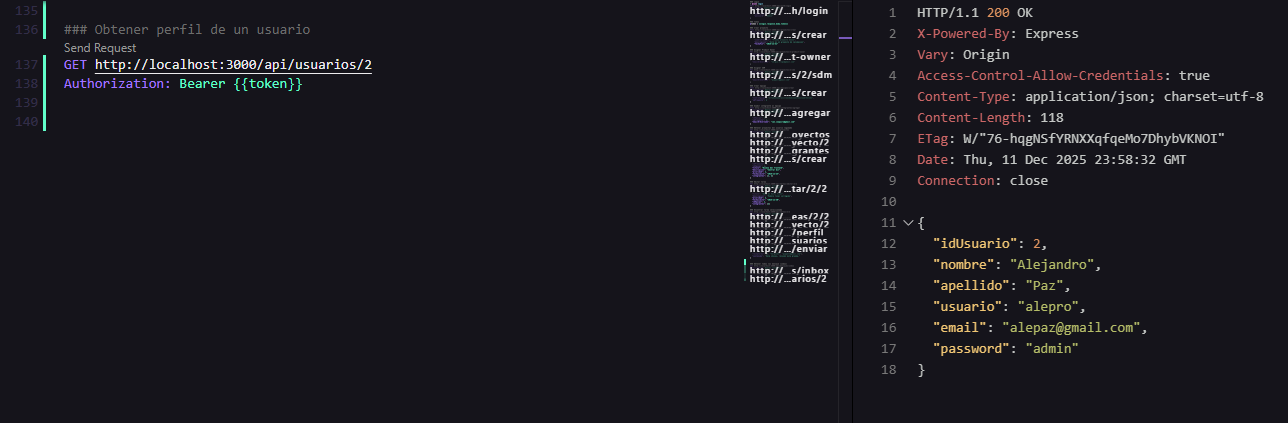
No importa cual reciba, igual se guarda, el por los ids es una opción para que aparezcan las opciones al momento de querer escoger a los receptores.



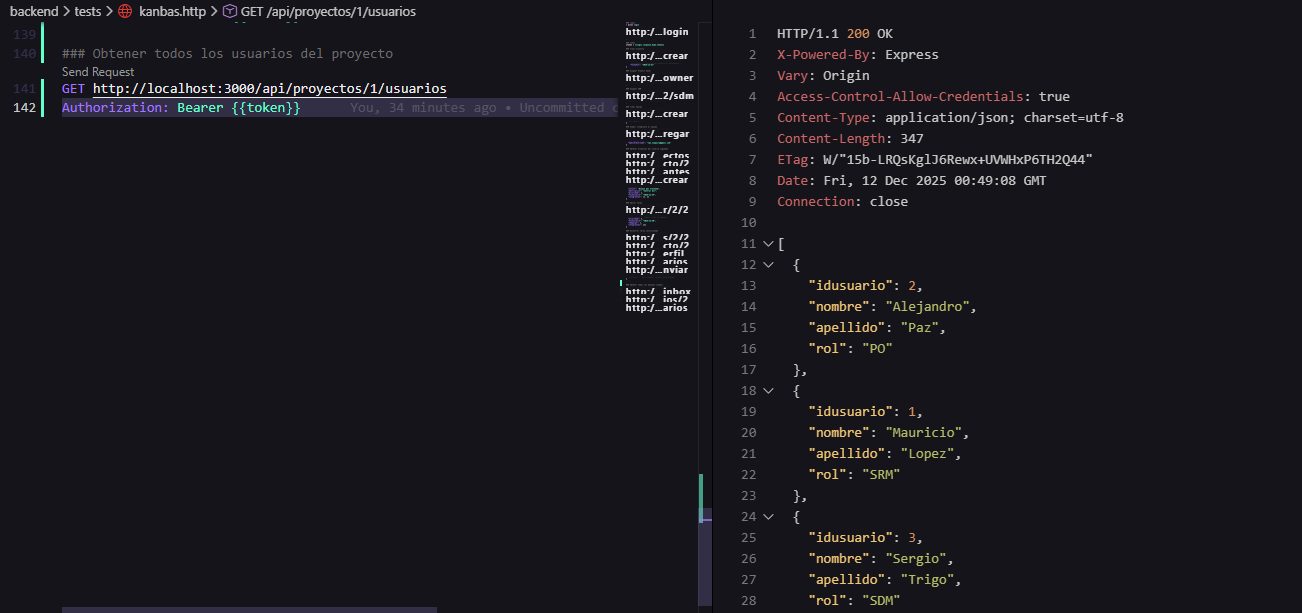


Con este GET obtenemos todos los mensajes via inbox que tiene nuestro usuario.

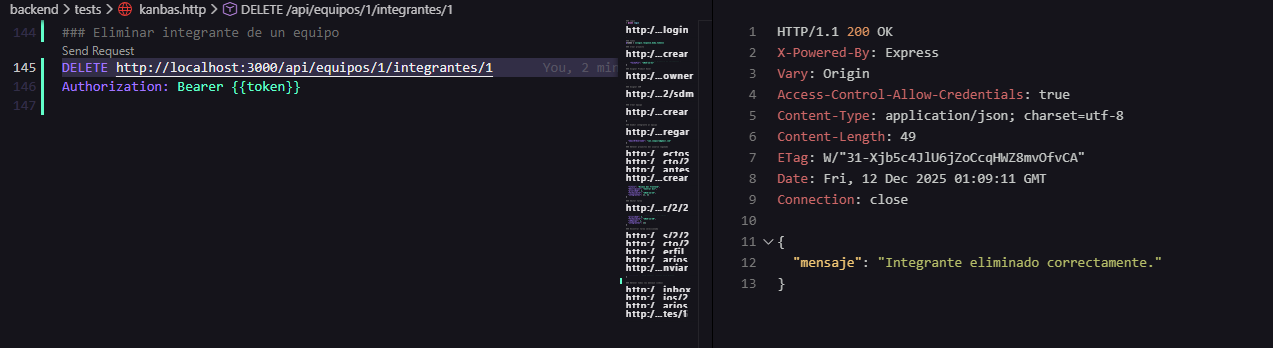
**Funcionalidades extra pedido ale xd**



Aquí tienes lo de obtener un usuario según su perfil, la url es [http://localhost:3000/api/usuarios/{idUsuario}](http://localhost:3000/api/usuarios/%7bidUsuario%7d)

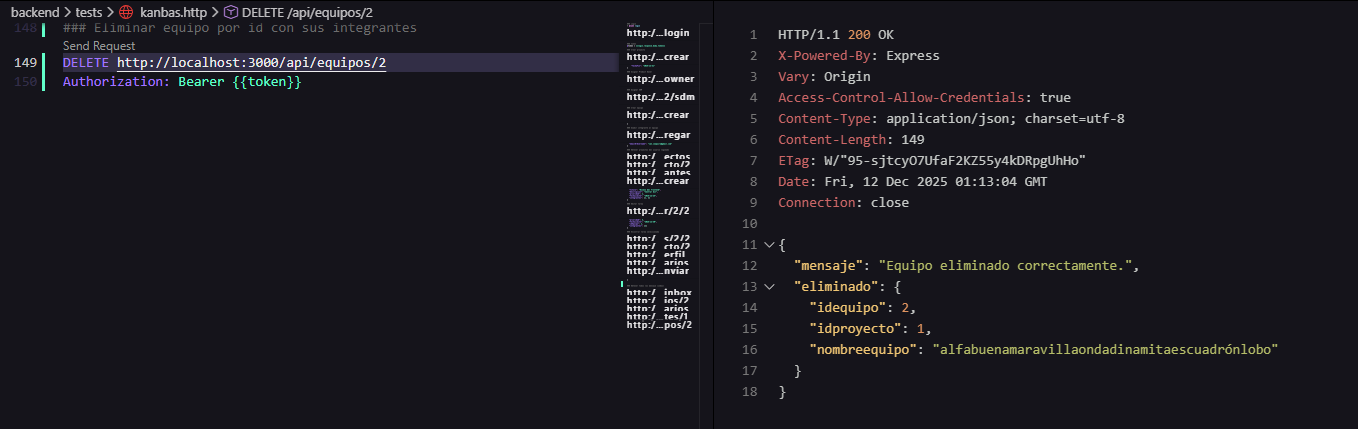


Aca tienes lo de obtener todos los usuarios que están en un proyecto. La url es: [http://localhost:3000/api/proyectos/{idProyecto}/usuarios](http://localhost:3000/api/proyectos/%7bidProyecto%7d/usuarios)



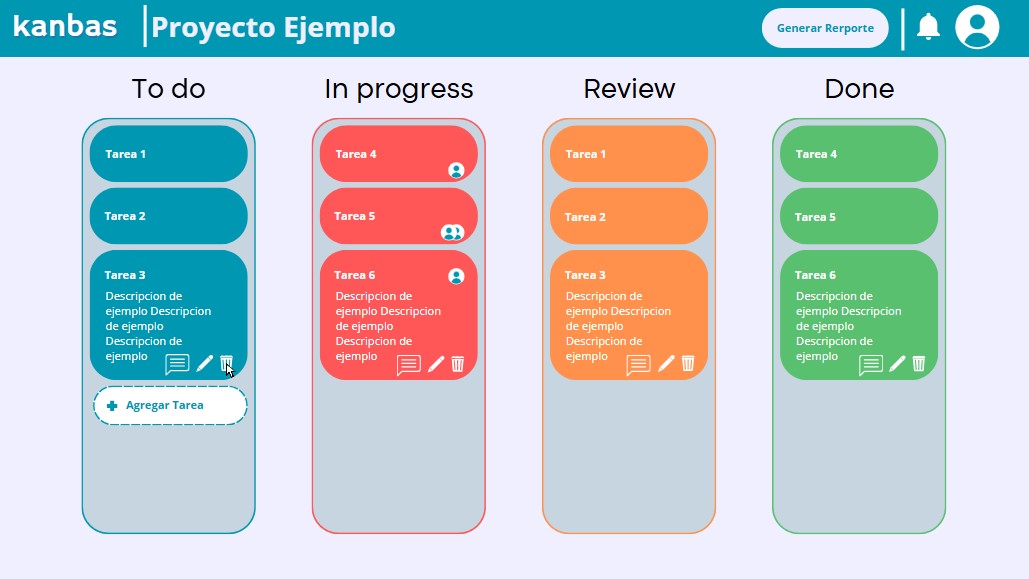
Aca tienes lo de eliminar integrantes de un equipo la url es:

[http://localhost:3000/api/equipos/{idEquipo}/integrantes/{idIntegrante}](http://localhost:3000/api/equipos/%7bidEquipo%7d/integrantes/%7bidIntegrante%7d)



Aca tienes lo de equipo eliminado, la url es: [http://localhost:3000/api/equipos/{idEquipo}](http://localhost:3000/api/equipos/%7bidEquipo%7d)

**Continuando con los mockups y lo que necesita**

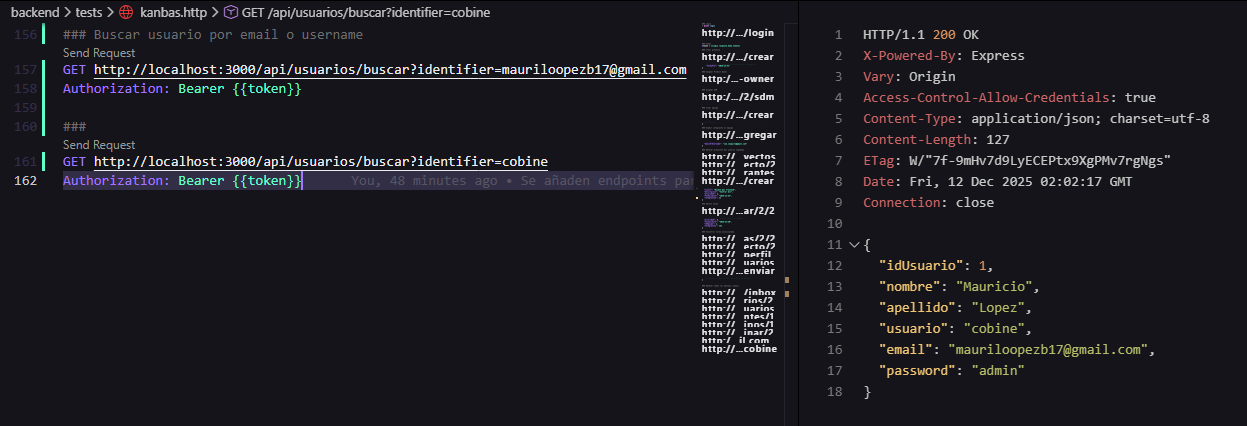


Para eliminar una tarea se puede hacer:

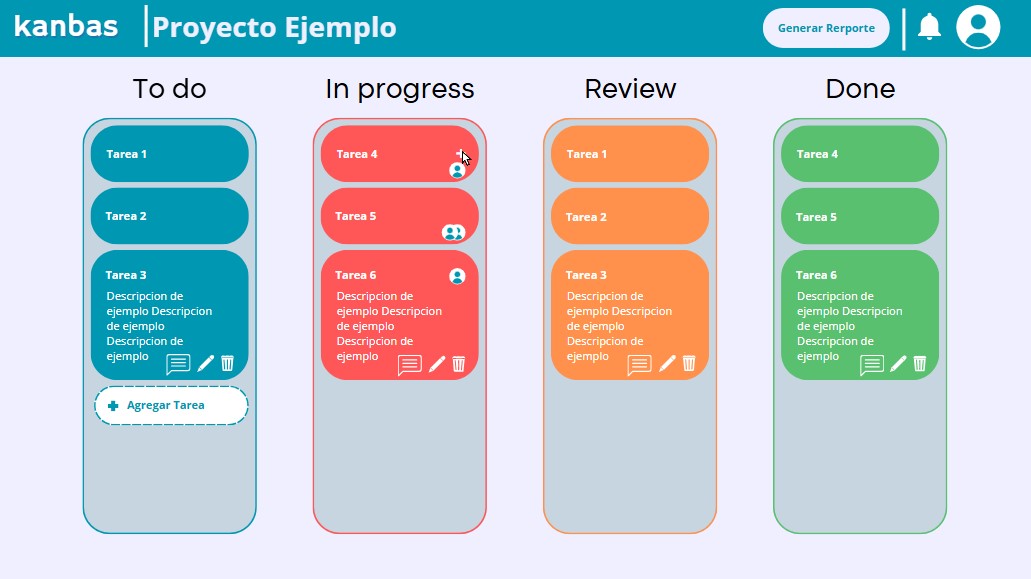


La URL es: [http://localhost:3000/api/tareas/eliminar/{idTarea}](http://localhost:3000/api/tareas/eliminar/%7bidTarea%7d)

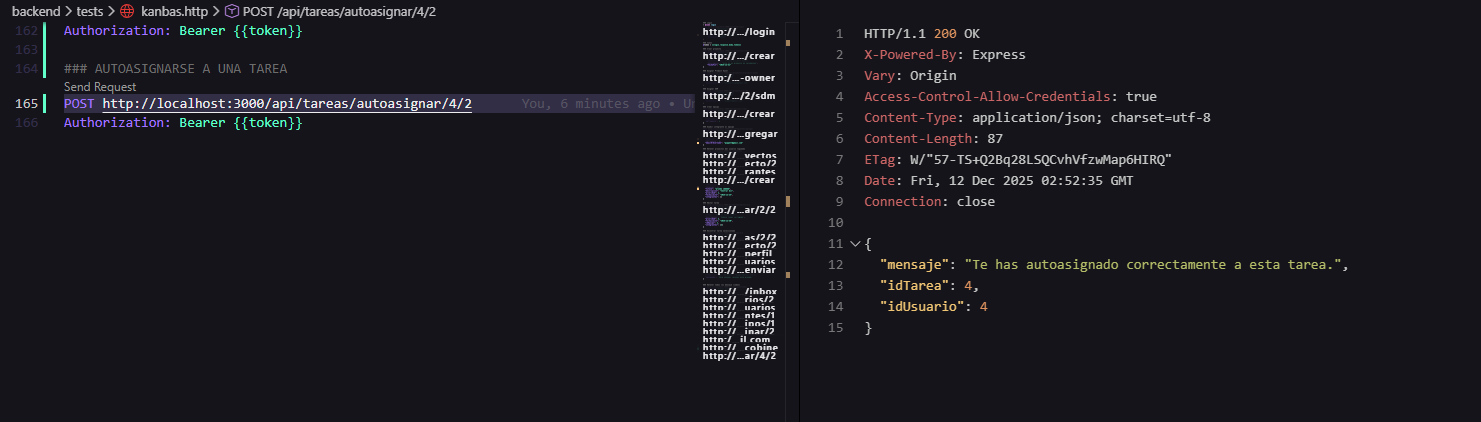
**Funcionalidad extra ale**

****

Url: [http://localhost:3000/api/usuarios/buscar?identifier={identificador](http://localhost:3000/api/usuarios/buscar?identifier=%7bidentificador) email o username}

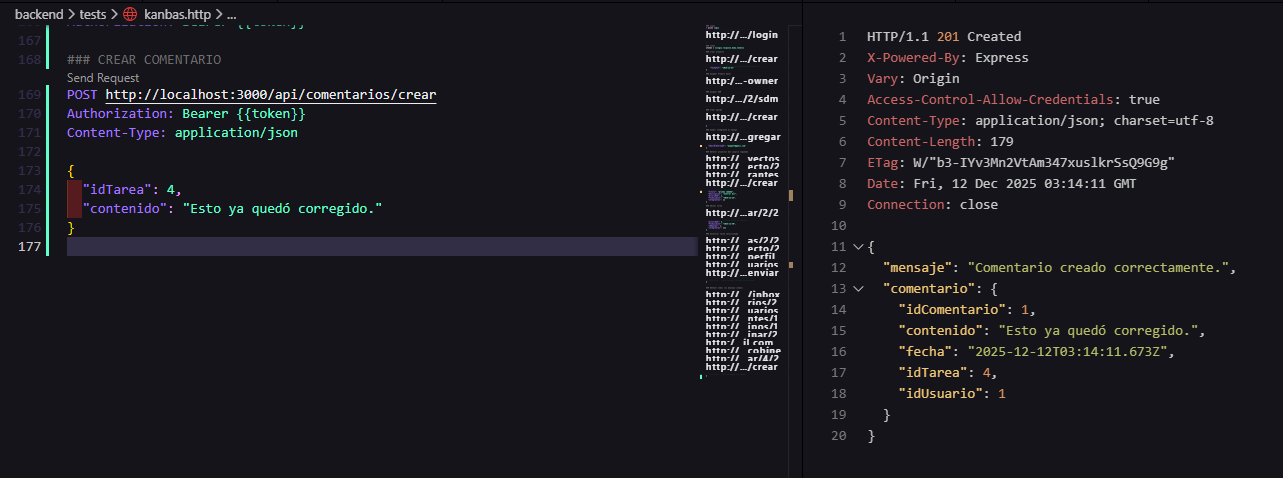


Para autoasignarte tareas puedes hacer:



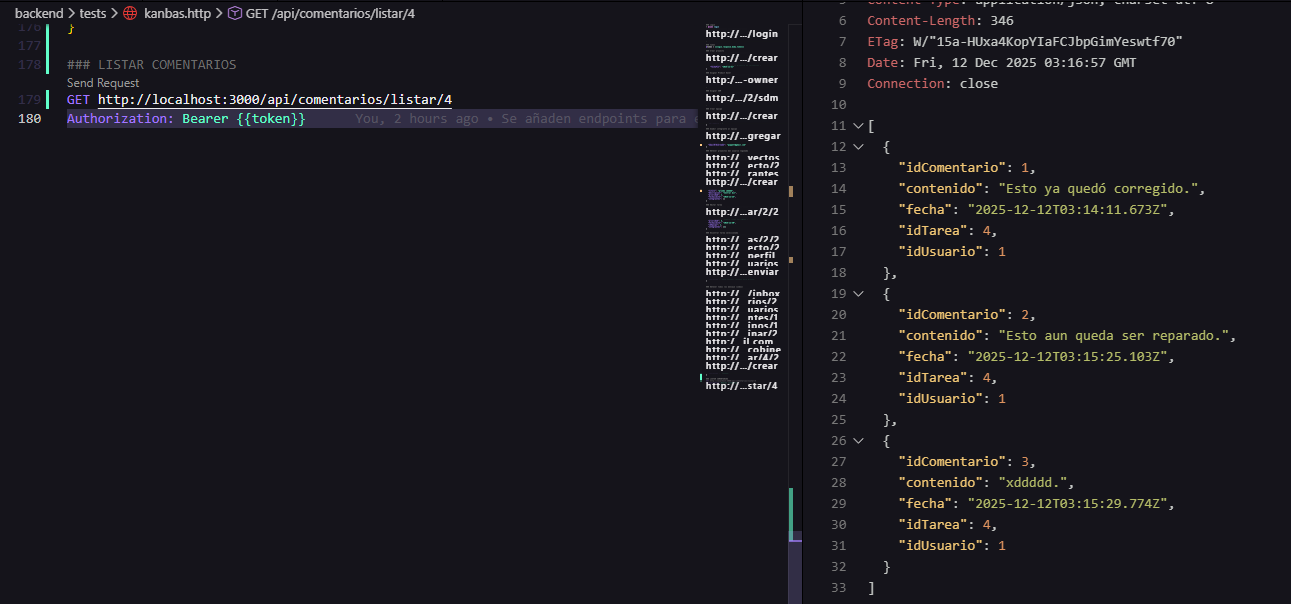
La URL: [http://localhost:3000/api/tareas/autoasignar/{idTarea}/{idProyecto}](http://localhost:3000/api/tareas/autoasignar/%7bidTarea%7d/%7bidProyecto%7d)

**Añadir comentario**

****

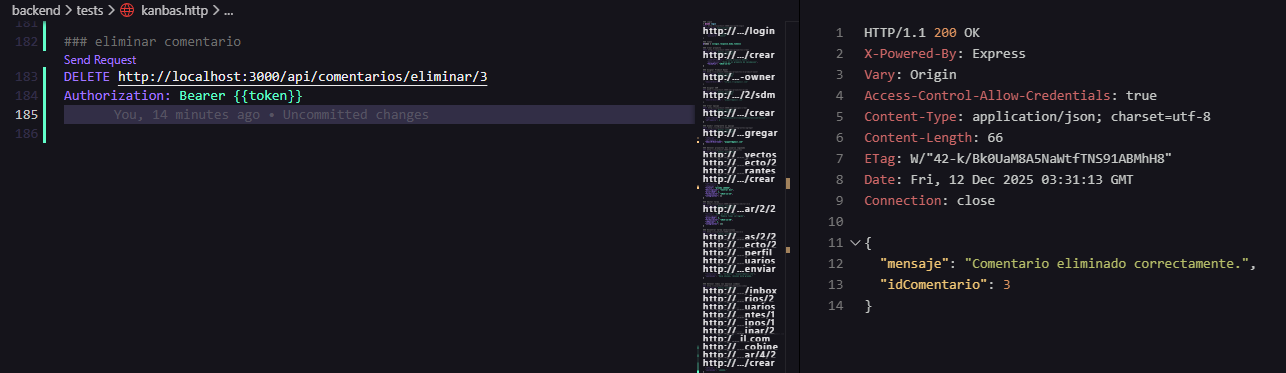
Para añadir el comentario usamos este endpoint. La URL es: <http://localhost:3000/api/comentarios/crear>

Para listar todas las tareas de un proyecto usamos:

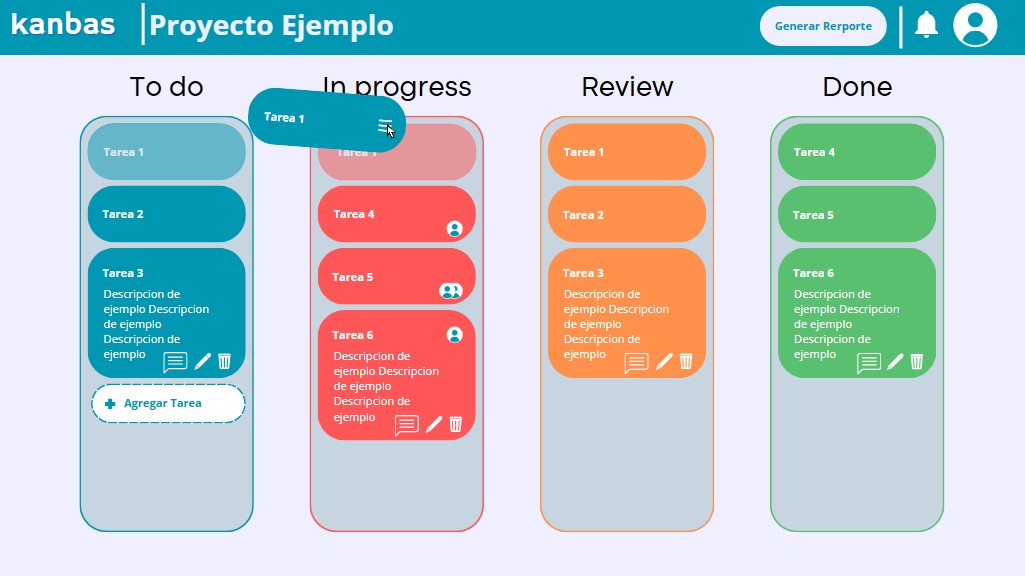


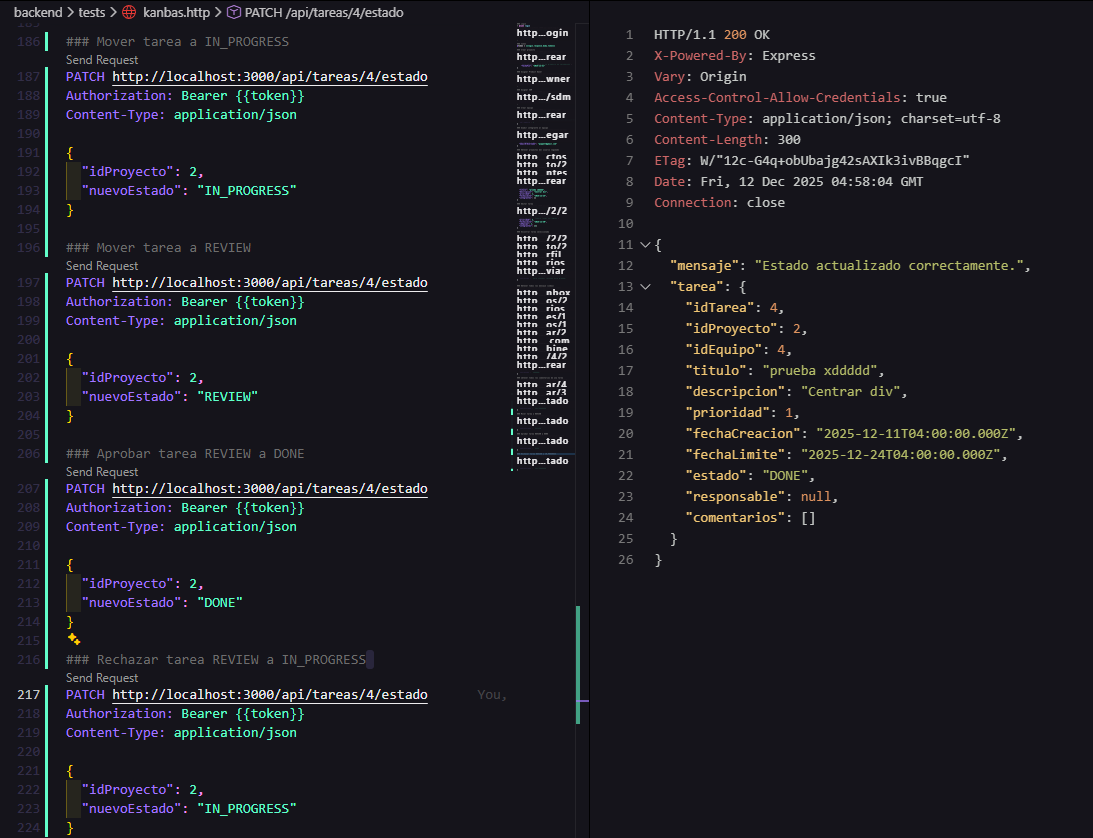
La URL es: [http://localhost:3000/api/comentarios/listar/{idTarea}](http://localhost:3000/api/comentarios/listar/%7bidTarea%7d)

Para eliminar la tarea tenemos:



La URL es: [http://localhost:3000/api/comentario/eliminar/{idComentario}](http://localhost:3000/api/comentario/eliminar/%7bidComentario%7d)

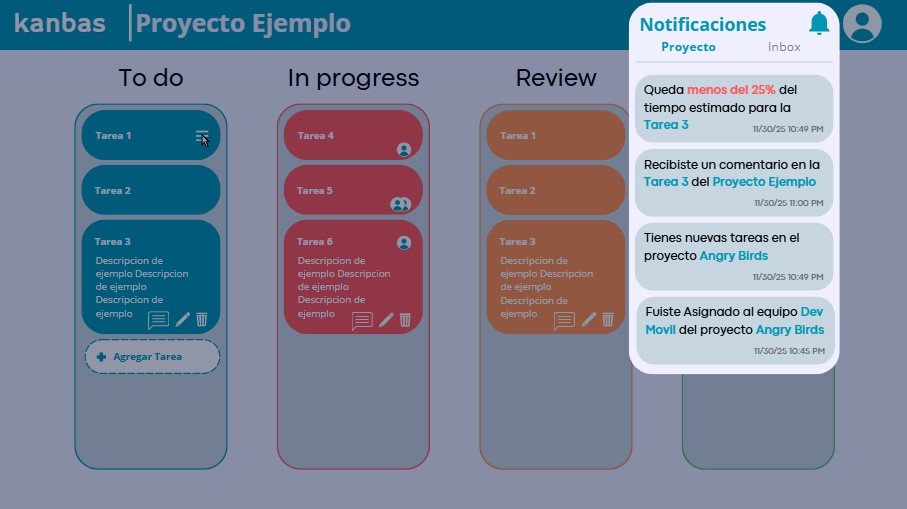




Para actualizar el estado de una tarea usamos estos endpoints, que prácticamente son el mismo, pero enviamos un diferente nuevoEstado. La URL seria: [http://localhost:3000/api/tareas/{idTarea}/estado](http://localhost:3000/api/tareas/%7bidTarea%7d/estado)

De forma interna se hacen las validaciones de:

* De To do –> In Progress (solo integrante)
* De In Progress –> Review (solo integrante)
* De Review –> Done (Solo por SRM o PO)
* De Review –> In Progress (Solo por SRM o PO)





Con este endpoint obtenemos la lista de todas las notificaciones de un usuario. La URL: <http://localhost:3000/api/notificaciones>