



La presente evaluación es de carácter **individual**, evite cualquier tipo de plagio o será penalizado. Evite compartir su código en chats grupales ya que podría verse afectado en su evaluación al identificar código copiado y modificado.

El objetivo de la evaluación es desarrollar el Backend del sitio web que toma como referencia la aplicación **Microsoft Teams**, (Si..., otra vez).

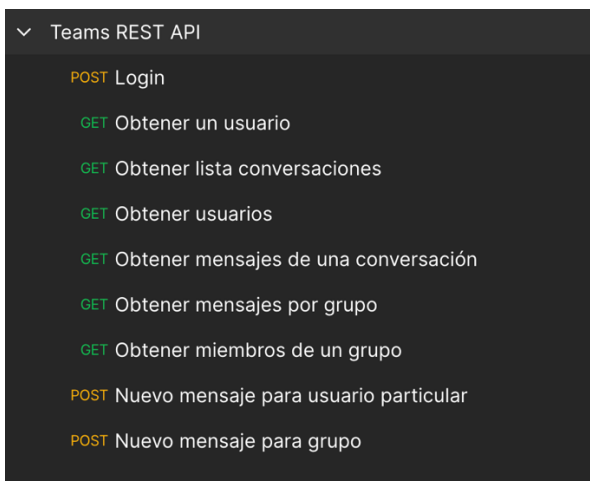
En esta evaluación solo debe trabajar en el backend.

#### Importante:

- Puede utilizar sus apuntes, repositorios, Google (incluso chatgpt) únicamente como guía. Evite copiar y pegar código que no entiende o que no ha escrito por su cuenta.
- En caso de que su solución sea correcta pero su código sea complejo o haya implementado conceptos, técnicas o similares no vistos en clase, se contactará al estudiante para solicitar la explicación de la solución (evite copiar de chatgpt, tutoriales o similares)
- Al subir su solución, eliminar la carpeta node\_modules y dist de su backend.
- Puede utilizar Typescript o NestJS como opcional en el backend.

#### Se incluye:

- Backend en nodeJS (recuerde primero **npm install** para instalar las dependencias)
- Este backend es el mismo que se incluyó en la unidad anterior, únicamente úselo como guía, tendrá que desarrollar uno nuevo.
- La información base se encuentra en los archivos **usuarios.json**, **conversaciones.json** y **grupos.json**, esta deberá de modificarla para cargarla a MongoDB, utilice ObjectId en vez de los ids incrementales.
- Colección de postman con los siguientes servicios web (**Teams REST API.postman\_collection.json**, importar desde postman):



*\*Recuerde que para las peticiones POST se deben enviar parámetros en el cuerpo de la petición, para ver los parámetros necesario desde postman, selecciones Body->Raw->JSON, además puede dar click en beautify para formatear el JSON.*



## Funcionalidades

- Crear una base de datos llamada **teams**, luego crear colecciones en mongoDB que se adapten a la información de los archivos JSON proveídos, utilice ObjectId en vez de los ids numéricos que poseen los archivos. Para ello se recomienda primero modificar el archivo de usuarios y cambiar el atributo id por `_id = ObjectId` y utilizar un `insertMany`, luego en base a los ids generados modificar las demás referencias de los otros JSONs de los otros archivos. Adjuntar exports de las colecciones de su base de datos en su solución.
- Crear un backend con nodejs, typescript (opcional), express, cors y demás módulos que considere necesarios para crear todos los endpoints utilizados en la unidad anterior, pero utilizando la información de MongoDB. Para ello crear la estructura de controladores, routers, modelos y esquemas correspondientes.
- Agregar un endpoint para agregar un miembro a un grupo.
- Agregar un endpoint para obtener la lista de llamadas, crear una colección básica en mongoDB con la información necesaria.
- Agregar un endpoint para eliminar una conversación.
- Agregar un endpoint para eliminar un mensaje.

## Subir:

- Backend sin la carpeta `node_modules`
- Exports de las colecciones creadas (aunque haya utilizado Mongo Atlas).
- Export de una colección de postman con los endpoints creados. Asegúrese de guardar cada endpoint antes de hacer el export.

<<A veces solo hay que decir "Diooooo mio, como e posiblee eteee sucesoooo" y seguir adelante>>, Anónimo.

*¡Éxito!*