

# BACKEND 2

Bootcamp Presencial



# Full Stack Developer



**GeeksHubs**  
academy\_

## Descripción

En esta prueba técnica se espera que muestres todos los conocimientos de los que dispones para hacer frente al desarrollo de un backend. Para ello y puesto que el fin de esta prueba es determinar los conocimientos que posees sobre JavaScript, Express, Node y Mongo, deberás desarrollar una red social en la que los usuarios puedan registrarse, acceder a la red e interactuar.

Como punto importante se valorará la posibilidad de seguir y dejar de seguir a otros usuarios, así como dejar comentarios en su perfil o en sus posts.

## Requisitos funcionales:

Desarrollarás una red social en la que los usuarios se registrarán, logearán y publicarán.

Para ello se necesitarán como mínimo las colecciones:

- users
- posts (publicaciones)

Se creará y presentará un diagrama explicando las relaciones entre las entidades.

## Entregable esperado:

Una vez analizada las necesidades de la aplicación, se espera que puedas desarrollar una API REST para nutrir las vistas de frontend correspondientes:

## Endpoints:

### Autenticación

Método	URI	Acción	Extra
POST	/api/auth/register	Registro de usuarios	
POST	/api/auth/login	Login de usuarios	

## Usuarios

Método	URI	Acción	Extra
GET	/api/users	Ver todos los usuarios (super_admin)	
GET	/api/users/profile	Ver perfil de usuario	
PUT	/api/users/profile	Modificar datos del perfil (al menos un campo)	
GET	/api/users?email=ejemplo@ejemplo.com	Filtrar usuario por email (super_admin)	SI
DELETE	/api/users/{id}	Eliminar usuario (super_admin)	SI
PUT	/api/users/{id}/role	Cambio de role (super_admin)	SI

## Citas

Método	URI	Acción	Extra
POST	/api/posts	Crear post	
DELETE	/api/posts/{id}	Eliminar post por id	
PUT	/api/posts	Actualizar Post por id	
GET	/api/posts/own	Recuperar mis propios posts	
GET	/api/posts	Recuperar todos los posts	
GET	/api/posts/{id}	Recuperar post por id	
GET	/api/users/posts/{user-id}	Recuperar posts de un usuario	

## Likes

Método	URI	Acción	Extra
PUT	/api/posts/like/{id}	Dar y quitar like	

## Otros aspectos requeridos:

- Seeders para las diferentes colecciones.
- Middlewares para comprobar la autoría del post a la hora de editar/eliminar el mismo.
- Deploy a producción del código de backend

## Super Extra Points

- El usuario puede seguir a otros usuarios (Following) y por lo tanto el usuario puede tener seguidores (Followers)

Método	URI	Acción	Extra
PUT	<a href="#">/api/users/follow/{user-id}</a>	follow y unfollow user	SI

- El usuario recuperará en su timeline solo los posts de aquellos usuarios a los que siga.

Método	URI	Acción	Extra
GET	<a href="#">/api/posts/timeline</a>	Recuperar posts de seguidores	SI

## Tecnologías

Para el desarrollo de la API utilizaremos node + Express + mongo



## Requisitos indispensables para la corrección

Será requisito indispensable que los repositorios dispongan de un README completo. La buena explicación del mismo con ilustraciones y animaciones (GIF, svg, etc.) se tendrá en cuenta.

El uso de git, será requisito indispensable la existencia de ramas y un número coherente de commits. (Mínimo dos ramas: develop y master)

Será requisito indispensable que el código a corregir este en la rama MASTER / MAIN

Para la entrega del proyecto se deberá entregar el link del repositorio donde está alojado el código en classroom.

### **Dudas**

Para cualquier duda puedes dirigirte al Slack de Geekshubs. En el grupo formativo al que perteneces.