

## PRUEBAS - POSTMAN

### 1. Registro de Usuario:

- Método: POST
- URL: <http://localhost:3000/api/auth/register>

```
{  
  "username": "testuser",  
  "email": "testuser@example.com",  
  "password": "password123"  
}
```

### 2. Inicio de Sesión:

- Método: POST
- URL: <http://localhost:3000/api/auth/login>
- Body (raw JSON):

```
{  
  "email": "testuser@example.com",  
  "password": "password123"  
}
```

- Guarda el token JWT que recibes en la respuesta

### 3. Obtener Perfil del Usuario:

- Método: GET
- URL: <http://localhost:3000/api/auth/me>
- Headers:
  - Key: Authorization
  - Value: <tu\_token\_jwt>

### 4. Crear una Publicación:

- Método: POST

- URL: <http://localhost:3000/api/posts>
- Headers: Authorization: <tu\_token\_jwt>

```
{
  "content": "Esta es mi primera publicación",
  "privacy": "public"
}
```

#### 5. Obtener Publicaciones:

- Método: GET
- URL: <http://localhost:3000/api/posts>
- Headers: Authorization: <tu\_token\_jwt>

#### 6. Actualizar Privacidad de una Publicación:

- Método: PATCH
- URL: [http://localhost:3000/api/posts/<id\\_de\\_la\\_publicacion>/privacy](http://localhost:3000/api/posts/<id_de_la_publicacion>/privacy)
- Headers: Authorization: <tu\_token\_jwt>

```
{
  "privacy": "friends"
}
```

#### 7. Compartir una Publicación:

- Método: POST
- URL: [http://localhost:3000/api/posts/<id\\_de\\_la\\_publicacion>/share](http://localhost:3000/api/posts/<id_de_la_publicacion>/share)
- Headers: Authorization: <tu\_token\_jwt>

#### 8. Obtener News Feed:

- Método: GET
- URL: <http://localhost:3000/api/posts/news-feed>
- Headers: Authorization: <tu\_token\_jwt>

#### 9. Crear un Comentario:

- Método: POST
- URL: <http://localhost:3000/api/comments>
- Headers: Authorization: <tu\_token\_jwt>

```
{
  "postId": "<id_de_la_publicacion>",
  "content": "Este es un comentario de prueba"
}
```

#### 10. Obtener Comentarios de una Publicación:

- Método: GET
- URL: [http://localhost:3000/api/comments/<id\\_de\\_la\\_publicacion>](http://localhost:3000/api/comments/<id_de_la_publicacion>)
- Headers: Authorization: <tu\_token\_jwt>

#### 11. Dar Like a una Publicación:

- Método: POST
- URL: [http://localhost:3000/api/likes/post/<id\\_de\\_la\\_publicacion>](http://localhost:3000/api/likes/post/<id_de_la_publicacion>)
- Headers: Authorization: <tu\_token\_jwt>

#### 12. Dar Like a un Comentario:

- Método: POST
- URL: [http://localhost:3000/api/likes/comment/<id\\_del\\_comentario>](http://localhost:3000/api/likes/comment/<id_del_comentario>)
- Headers: Authorization: <tu\_token\_jwt>

#### 13. Enviar Solicitud de Amistad:

- Método: POST
- URL: <http://localhost:3000/api/friends/request>
- Headers: Authorization: <tu\_token\_jwt>

```
{
  "receiverId": "<id_del_usuario_receptor>"
}
```

14. Obtener Solicitudes de Amistad Pendientes:

- Método: GET
- URL: <http://localhost:3000/api/friends/requests>
- Headers: Authorization: <tu\_token\_jwt>

15. Responder a una Solicitud de Amistad:

- Método: POST
- URL: <http://localhost:3000/api/friends/respond>
- Headers: Authorization: <tu\_token\_jwt>

```
{  
  "requestId": "<id_de_la_solicitud_de_amistad>",  
  "status": "accepted" //o "rejected"  
}
```

16. Eliminar una publicación:

- Método: DELETE
- URL: [http://localhost:3000/api/posts/<id\\_de\\_la\\_publicacion>](http://localhost:3000/api/posts/<id_de_la_publicacion>)
- Headers: Authorization: <tu\_token\_jwt>

17. Eliminar un comentario:

- Método: DELETE
- URL: [http://localhost:3000/api/comments/<id\\_del\\_comentario>](http://localhost:3000/api/comments/<id_del_comentario>)
- Headers: Authorization: <tu\_token\_jwt>

18. Eliminar un like de una publicación:

- Método: DELETE
- URL: [http://localhost:3000/api/likes/post/<id\\_de\\_la\\_publicacion>](http://localhost:3000/api/likes/post/<id_de_la_publicacion>)
- Headers: Authorization: <tu\_token\_jwt>

19. Eliminar un like de un comentario:

- Método: DELETE
- URL: [http://localhost:3000/api/likes/comment/<id\\_del\\_comentario>](http://localhost:3000/api/likes/comment/<id_del_comentario>)
- Headers: Authorization: <tu\_token\_jwt>

20. Eliminar un amigo:

- Método: DELETE
- URL: [http://localhost:3000/api/friends/<id\\_del\\_amigo>](http://localhost:3000/api/friends/<id_del_amigo>)
- Headers: Authorization: <tu\_token\_jwt>

21. Eliminar cuenta:

- Método: DELETE
- URL: <http://localhost:3000/api/users/account>
- Headers: Authorization: <tu\_token\_jwt>

22. Editar una publicación:

- Método: PUT
- URL: [http://localhost:3000/api/posts/<id\\_de\\_la\\_publicacion>](http://localhost:3000/api/posts/<id_de_la_publicacion>)
- Headers:
  - Authorization: <tu\_token\_jwt>

```
{  
  "content": "Este es el contenido actualizado de la publicación"  
}
```

23. Editar un comentario:

- Método: PUT
- URL: [http://localhost:3000/api/comments/<id\\_del\\_comentario>](http://localhost:3000/api/comments/<id_del_comentario>)
- Headers:
  - Authorization: <tu\_token\_jwt>

```
{  
  "content": "Este es el contenido actualizado del comentario"  
}
```

24. Buscar usuarios por nombre:

- Método: GET
- URL: [http://localhost:3000/api/users/search?username=<nombre\\_a\\_buscar>](http://localhost:3000/api/users/search?username=<nombre_a_buscar>)
- Headers: Authorization: <tu\_token\_jwt>

1. Asegúrate de reemplazar <tu\_token\_jwt>, <id\_de\_la\_publicacion>, <id\_del\_comentario>, <id\_del\_usuario\_receptor>, e <id\_de\_la\_solicitud\_de\_amistad>, etc con los valores reales que obtengas durante las pruebas.

2. Para probar las funcionalidades de amigos, necesitarás crear al menos dos usuarios diferentes.
3. Guarda los IDs de las publicaciones, comentarios y solicitudes de amistad que crees, ya que los necesitarás para las pruebas subsiguientes.
4. Si obtienes errores, verifica que estés usando el token JWT correcto y que no haya expirado.
5. Para las rutas que requieren autenticación, asegúrate siempre de incluir el header de Authorization con el token JWT.