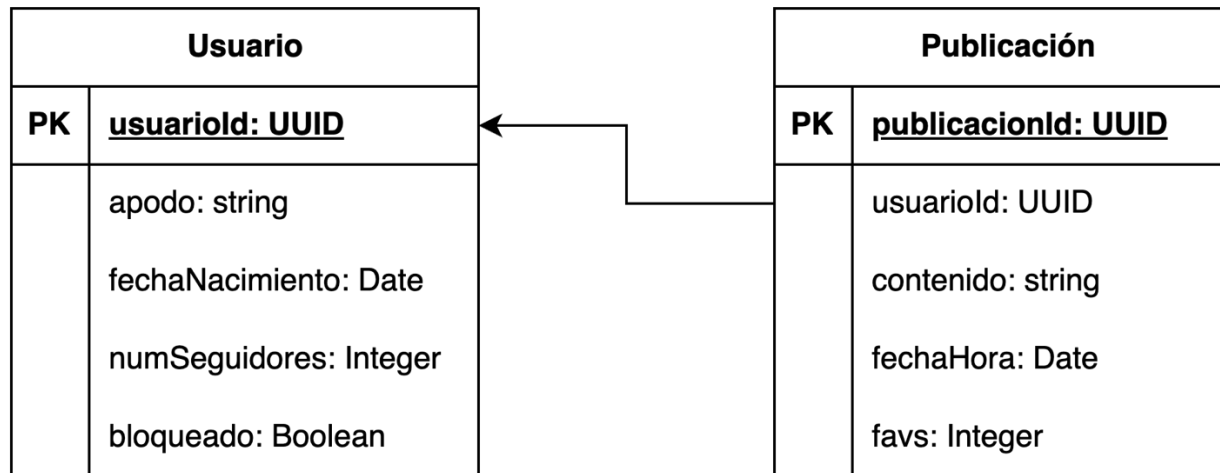


API REST red social

Implementar un servidor API REST para operar sobre una base de datos que tiene el siguiente modelo de datos:



A continuación, se especifican los controladores de la API:

UsuarioController		
Endpoint	Request	Response
GET /usuarios	-	200 OK [{ "usuarioId": <UUID>,

		<pre> “apodo”: <STRING>, “fechaNacimiento”: <DATE>, “numSeguidores”: <INTEGER>, “bloqueado”: <BOOLEAN> }] </pre>
GET /usuarios/{usuarioId}	-	200 OK <pre> { “usuarioId”: <UUID>, “apodo”: <STRING>, “fechaNacimiento”: <DATE>, “numSeguidores”: <INTEGER>, “bloqueado”: <BOOLEAN> } </pre>
		404 NOT FOUND Usuario con ID <usuarioId> no encontrado
POST /usuarios	<pre> { “apodo”: <STRING>, “fechaNacimiento”: <DATE>, } </pre>	201 CREATED <pre> { “usuarioId”: <UUID>, “apodo”: <STRING>, “fechaNacimiento”: <DATE>, “numSeguidores”: 0 } </pre>

		<pre> "bloqueado": false } </pre>
PUT /usuarios/{id}	<pre> { "apodo": <STRING>, "fechaNacimiento": <DATE>, "numSeguidores": <INTEGER>, "bloqueado": <BOOLEAN> } </pre>	200 OK <pre> { "usuarioId": <UUID>, "apodo": <STRING>, "fechaNacimiento": <DATE>, "numSeguidores": <INTEGER>, "bloqueado": <BOOLEAN> } </pre>
DELETE /usuarios/{usuarioId}	-	204 NO CONTENT -
		404 NOT FOUND Usuario con ID <usuarioId> no encontrado

PublicacionController		
Endpoint	Request	Response
GET /usuarios/{usuarioId} /publicaciones	-	200 OK <pre> [{ "publicacionId": <UUID>, "usuarioId": <UUID>, </pre>

		<pre> “contenido”: <STRING>, “fechaHora”: <DATE>, “favs”: <INTEGER> },]</pre>
GET /usuarios/{usuarioId} /publicaciones /{publicacionId}	-	200 OK <pre> [{ “publicacionId”: <UUID>, “usuarioId”: <UUID>, “contenido”: <STRING>, “fechaHora”: <DATE>, “favs”: <INTEGER> },]</pre>
		404 NOT FOUND Publicación con ID <publicacionId> no encontrada
POST /usuarios/{usuarioId} /publicaciones	<pre> { “contenido”: <STRING> }</pre>	201 CREATED <pre> { “publicacionId”: <UUID>, “usuarioId”: <UUID>, “contenido”: <STRING>, “fechaHora”: <FECHA ACTUAL>, “favs”: 0 }</pre>

PATCH /usuarios/{id} /publicaciones /{publicacionId}	<pre>{ "contenido": <STRING> }</pre>	200 OK <pre>{ "publicacionId": <UUID>, "usuarioId": <UUID>, "contenido": <STRING>, "fechaHora": <FECHA ACTUAL>, "fav": 0 }</pre>
		404 NOT FOUND Publicación con ID <publicacionId> no encontrada
DELETE /usuarios/{usuarioId} /publicaciones /{publicacionId}	-	204 NO CONTENT -
		404 NOT FOUND Publicación con ID <publicacionId> no encontrada

Los errores de cualquier tipo (formato inválido, recursos no encontrados, errores inesperados, etc.) deben manejarse con controller advices para evitar que el servidor se rompa.

Insertar en la "BD" 2 usuarios y 2 publicaciones para cada usuario. Crear una colección en Postman para probar el funcionamiento de todos los endpoints.