

Actividad: Paso a Paso Generando Código Javascript con IA

Objetivo

El objetivo de este trabajo práctico es que los alumnos desarrollen un nuevo endpoint en Biblioteca para la administración de usuarios utilizando una herramienta de generación de código guiada por Inteligencia Artificial.

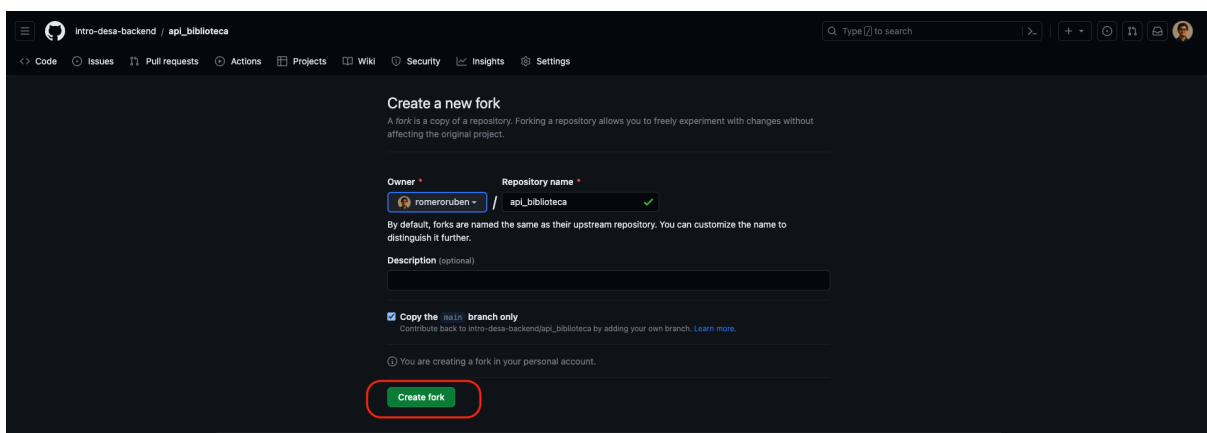
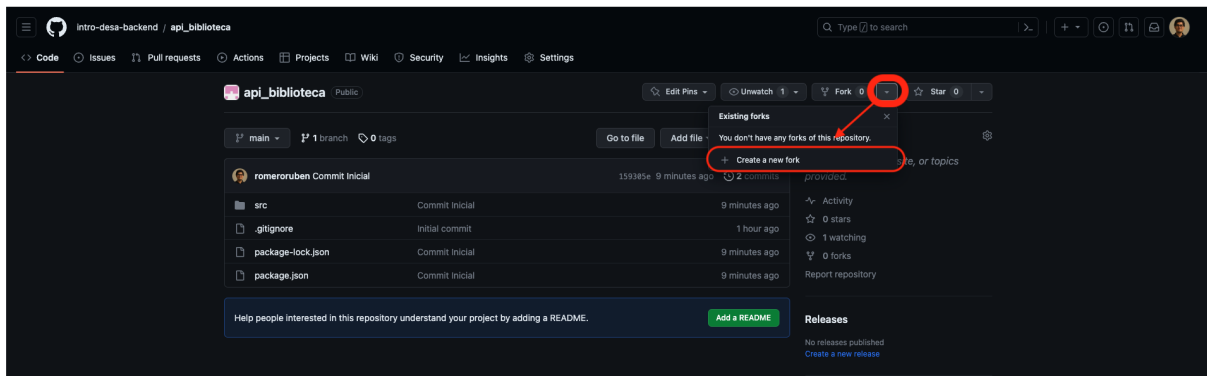
Consigna

1. Hacer un Fork del repositorio base.
2. Implementar las siguientes rutas y funcionalidades, utilizando la herramienta de generación de código **Codeium**:
 - a. GET /usuarios: Devuelve la lista completa de usuarios.
 - b. GET /usuarios/:id: Devuelve los detalles de un usuario específico según su ID.
 - c. POST /usuarios: Crea un nuevo usuario con la información proporcionada.
 - d. PUT /usuarios/:id: Actualiza la información de un usuario específico según su ID.
 - e. DELETE /usuarios/:id: Elimina un usuario específico según su ID.
3. Subir los cambios a GitHub.

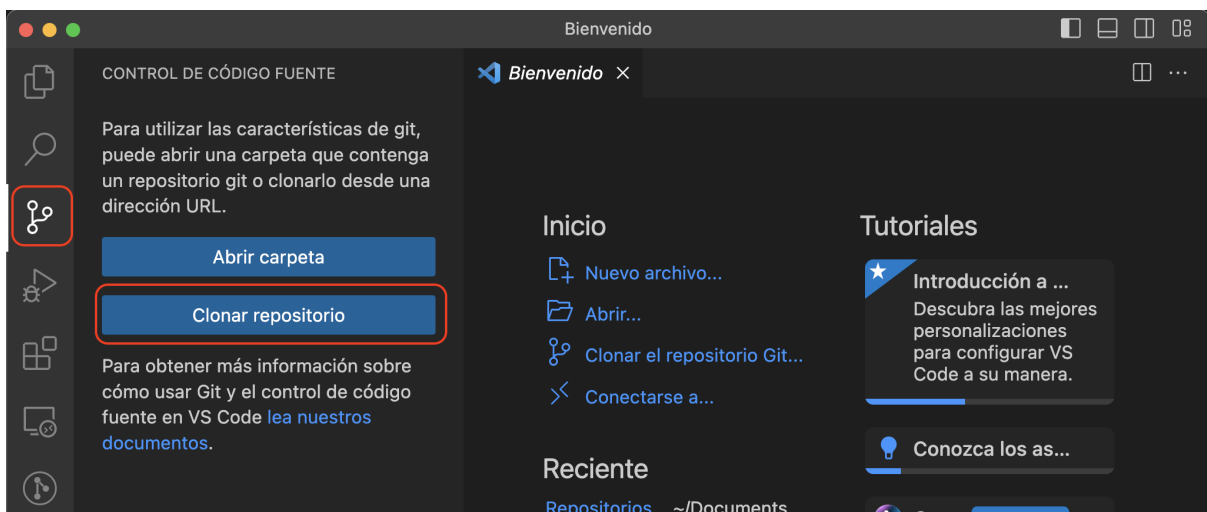
Desarrollo Paso a Paso

1. Primero vamos a ingresar al siguiente link que contiene el repositorio api_biblioteca de GitHub:

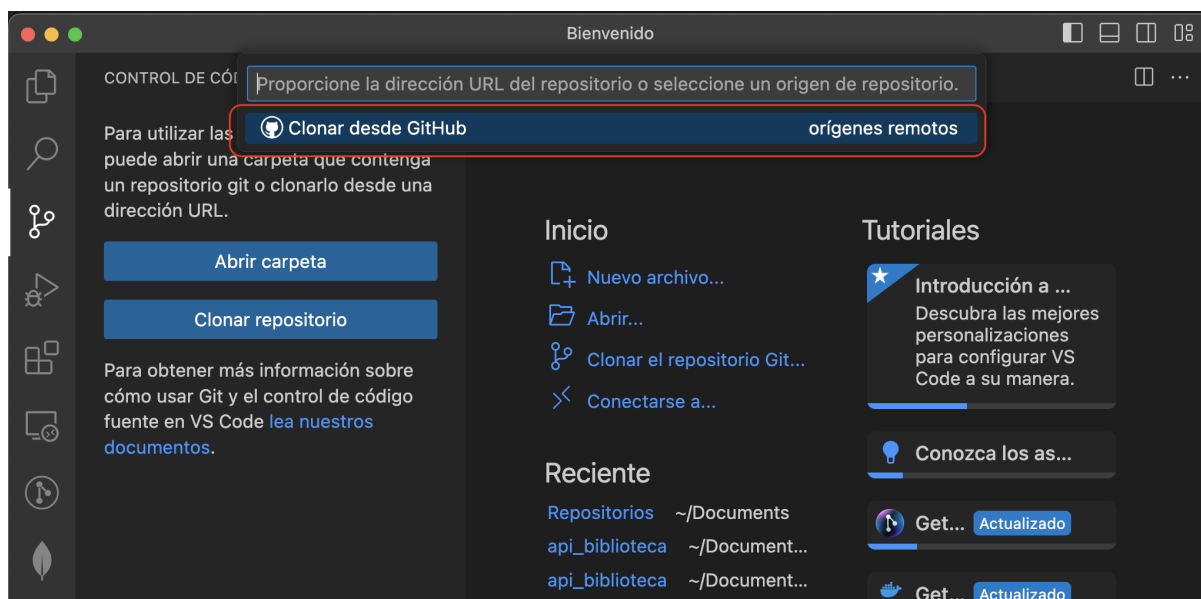
https://github.com/intro-des-a-backend/api_biblioteca
2. Vamos a crear un FORK de este repositorio para nuestra cuenta de GitHub. FORK en GitHub es una funcionalidad que permite a los usuarios hacer una copia de un repositorio ajeno en su cuenta de GitHub.



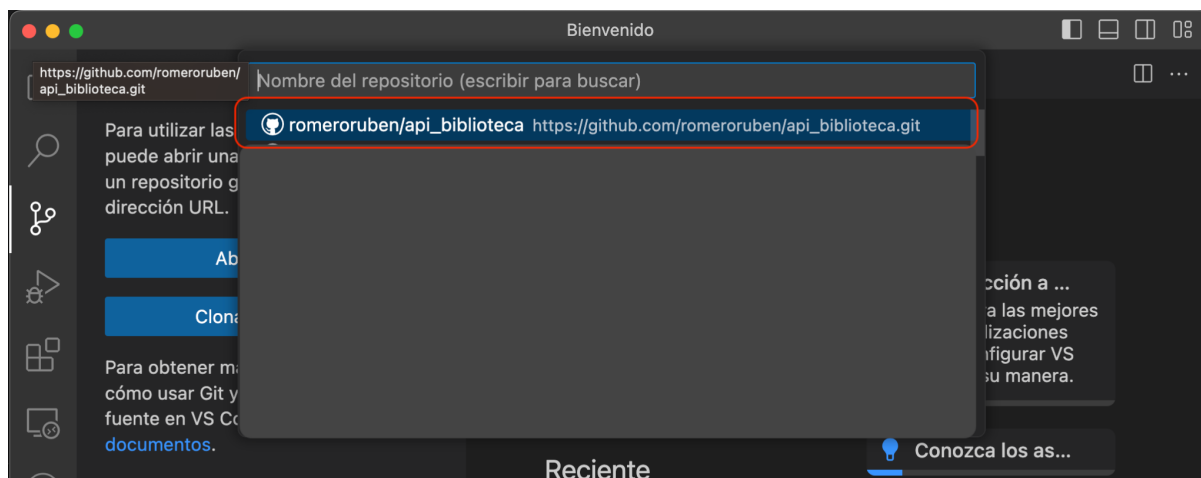
- Una vez que hicimos el FORK ya podemos clonar el repositorio localmente. Iniciamos la aplicación VSCode.
- Vamos a la opción de “Control de código fuente”, y seleccionamos la opción “Clonar repositorio” y seguimos los pasos:



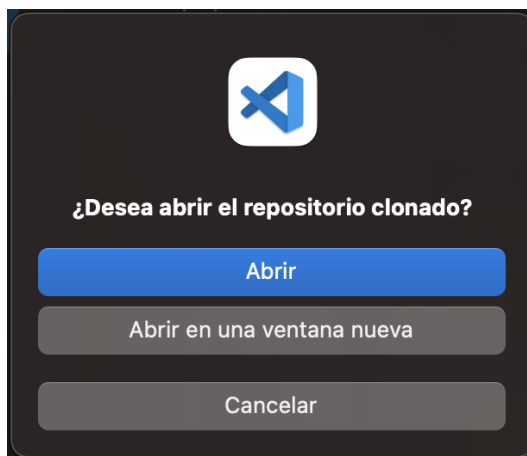
5. Hacemos click en “Clonar desde GitHub”



6. Seleccionamos el repositorio api_biblioteca que esta en nuestra cuenta:



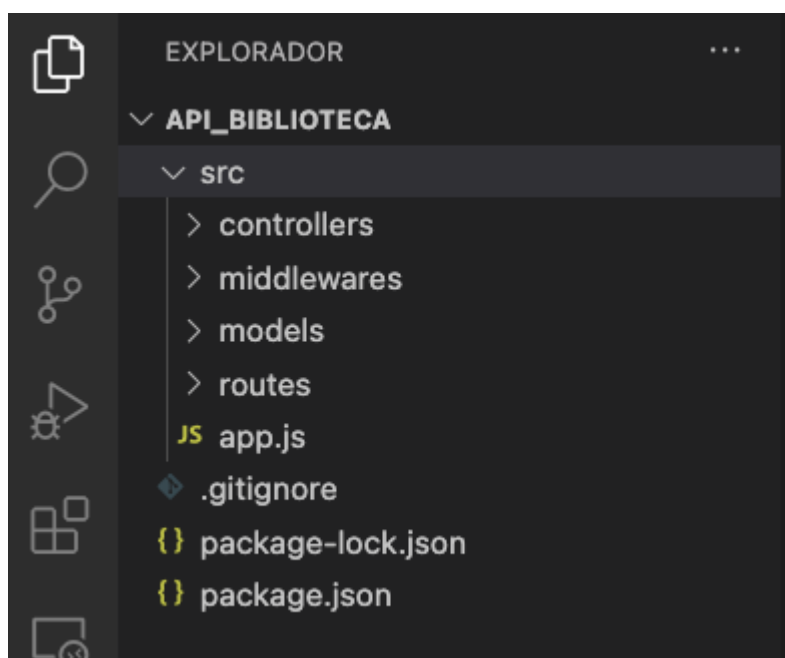
7. En el siguiente paso nos muestra el mensaje “Desea abrir el repositorio clonado?”, y hacemos click en **Abrir**



8. Esta es la estructura de archivos que deberían ver en su VSCode:

Nota: Tener en cuenta los siguientes puntos:

1. Se agregó una carpeta **src** donde colocamos el código que antes teníamos en la raíz del repositorio. Esto es para poder separar la carpeta del código fuente de los tests que vamos a hacer.
2. Agregamos una carpeta **controllers** donde tenemos el **libroController.js**, para poder separar la lógica de los routes y poder probar las funciones de manera aislada.



9. Ahora vamos a instalar los paquetes de npm localmente con el comando desde un terminal:

```
npm install
```

10. Debemos crear el archivo **.env** (de variables de entorno) con el siguiente contenido:

```
OAUTH_AUDIENCE=http://localhost:3000/api/biblioteca  
OAUTH_URL=https://dev-utn-frc-iaew.auth0.com/
```

11. En el archivo package.json se definieron los scripts **start** (para iniciar la API) y **test** (para ejecutar los tests):

```
"scripts": {  
  "start": "node src/app.js",  
  "test": "jest"  
},
```

12. Vamos a iniciar la API para probar que funcione todo bien:

```
npm start
```

13. Para probar la API necesitan crear un TOKEN, con el siguiente CURL, lo pueden importar desde Postman y crear un access_token:

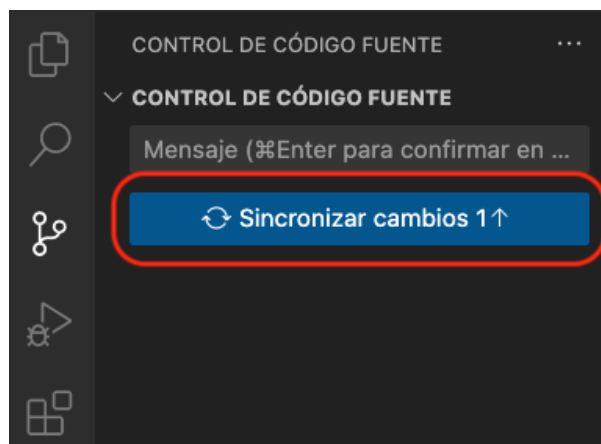
```
curl --location 'https://dev-utn-frc-iaew.auth0.com/oauth/token' \  
--header 'content-type: application/json' \  
--header 'Cookie:  
did=s%3Av0%3A76204e80-0ef1-11ee-8e12-b99ba3014b47.fKR174NkC7L0s9Z8riCysG  
37mMeRVakoS%2BTtye2lC3g;  
did_compat=s%3Av0%3A76204e80-0ef1-11ee-8e12-b99ba3014b47.fKR174NkC7L0s9Z  
8riCysG37mMeRVakoS%2BTtye2lC3g' \  
--data '{  
  "client_id": "Qiw8AlH9oykBg7ofBrHs6ToYvrdmhOeE",  
  "client_secret":  
"7kZPQqNnhRAuCXipSVSdHUsV9MzgfUzBB3AYbfemGPqxtpxI6j1GNxDBfyBSUume",  
  "audience": "http://localhost:3000/api/biblioteca",  
  "grant_type": "client_credentials"  
}'
```

14. Importando este CURL en Postman pueden probar que la API funcione ok, reemplazando el TOKEN.

```
curl --location 'http://localhost:3000/api/libros' \
--header 'Authorization: Bearer TOKEN'
```

Desarrollar con Codeium

15. Desarrollar el nuevo endpoint de la consigna utilizando la herramienta Codeium con las recomendaciones que nos va indicando hasta lograr la funcionalidad definida en la consigna.
16. Ahora vamos a subir los cambios a GitHub.
- Primero vamos a crear el Commit, en VSCode buscamos la opción de “Control de Código Fuente”, ingresamos un comentario “Nuevo endpoint Usuarios” y luego hacemos click en Confirmar, como se ve a continuación:
 - Luego hacemos el PUSH de los archivos a GitHub



17. **IMPORTANTE!** Para dar por finalizada la actividad compartir la url de GitHub del repositorio en su cuenta de GitHub en la actividad del aula virtual en <https://uve.frc.utn.edu.ar/>.