**Documentación RETO TÉCNICO - BACKEND SENIOR**

1. **Mínimo 2 endpoints, GET para recuperar la información y POST para crear un elemento:**
2. Registrar una Persona:

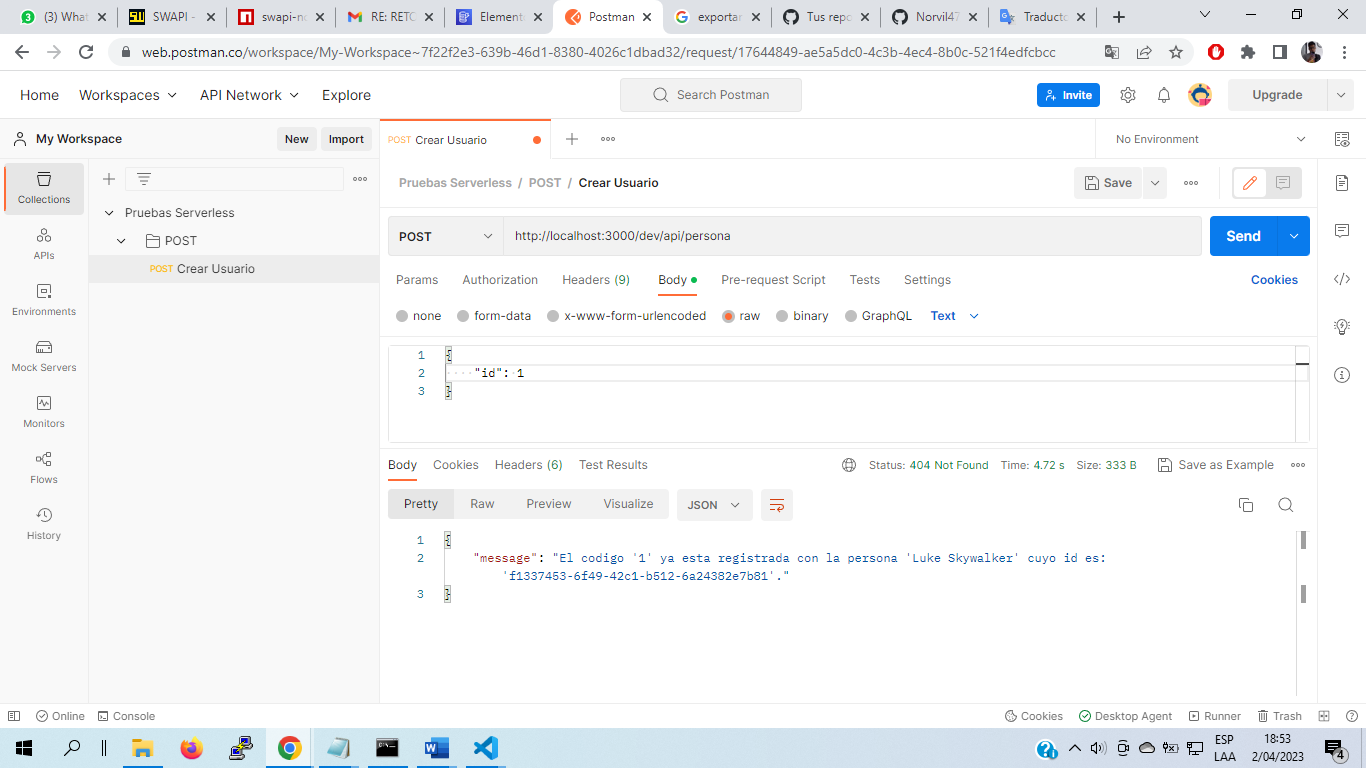
rutaLocal: **http://localhost:3000/dev/api/persona**

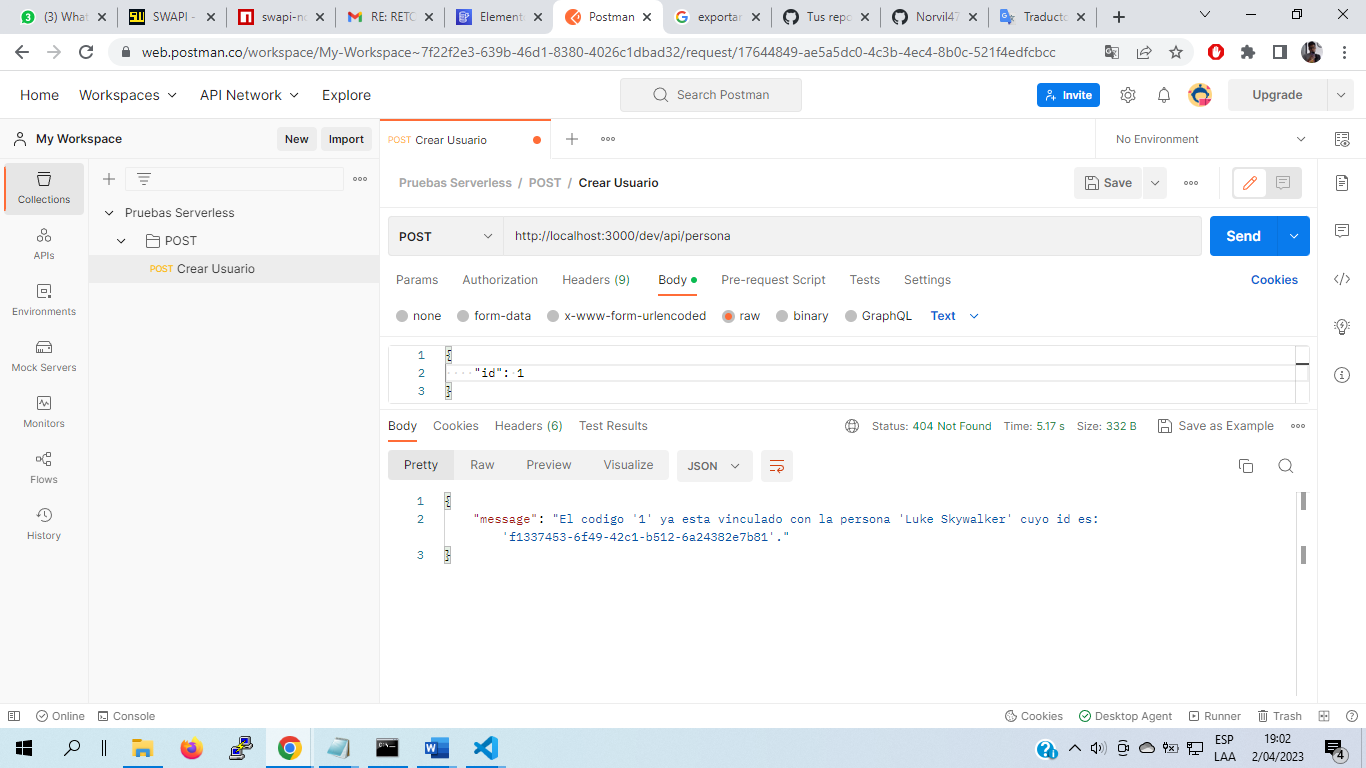
Método: **POST**

Parámetros-Body:

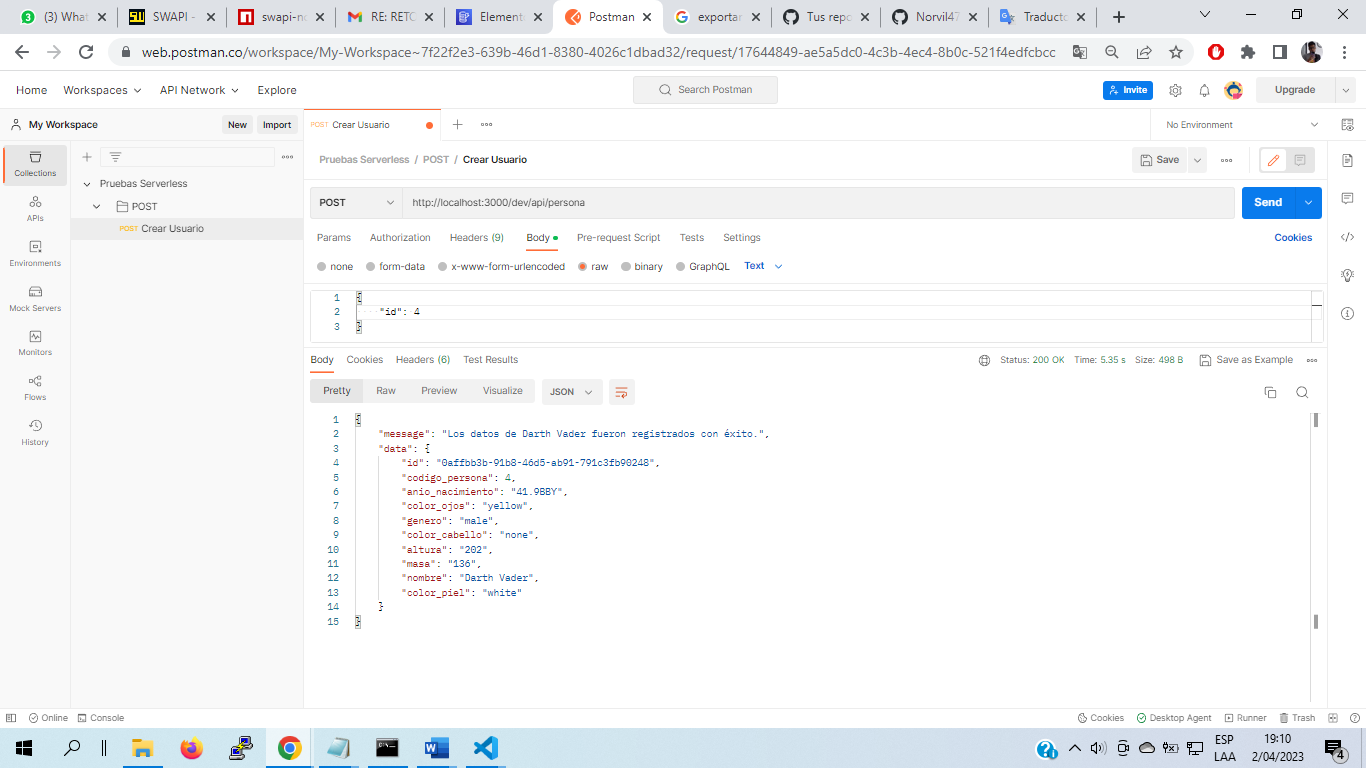
id:int – id del personaje definido en la Api de SW

Proceso:

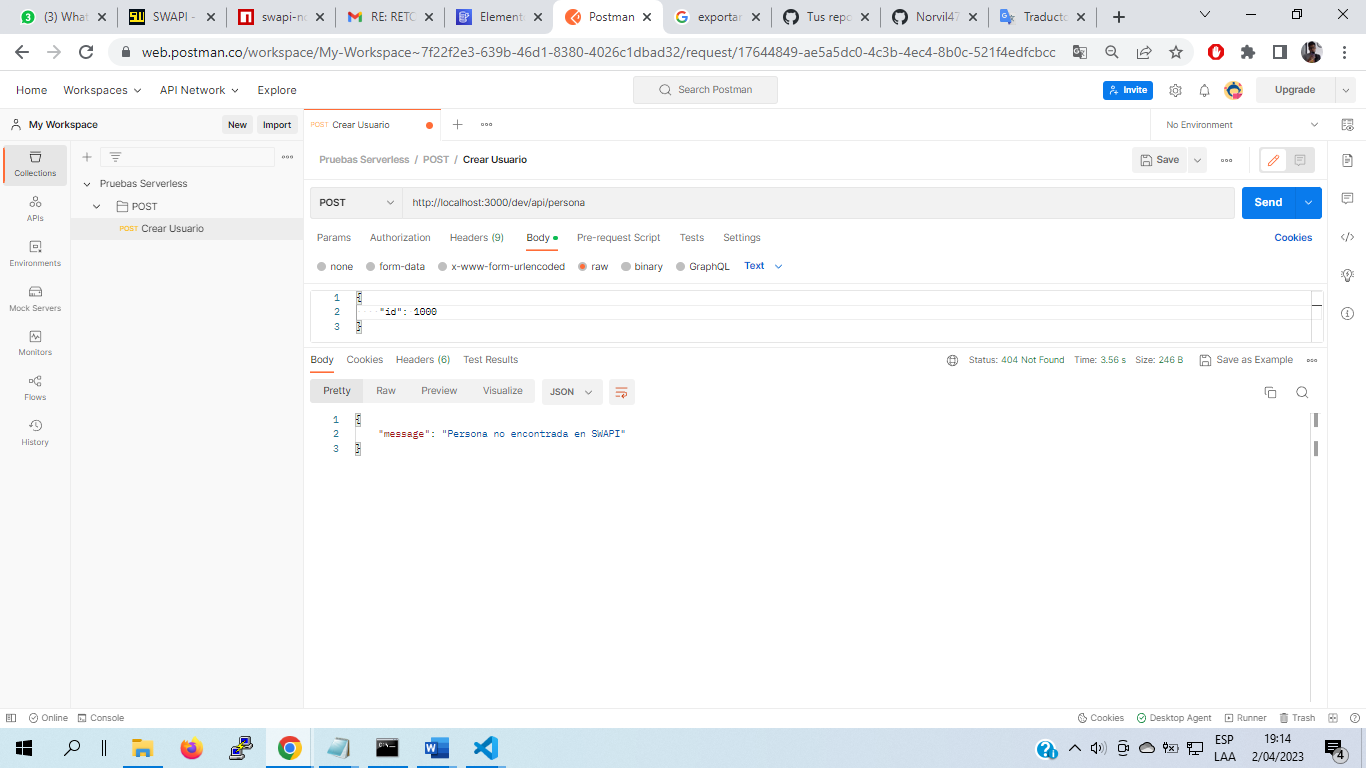
* Mandar por POST un **Json** con el ID indicado
* La API validará si el ID enviado ya se encuentra vinculado a un registro actual; si es así, se retornará un mensaje de alerta.



* Si se envía un ID del SWAPI q no haya sido registrado, entonces se procederá a recolectar la data y almacenarlo en el DB. Se enviará un mensaje de confirmación y la data registrada.



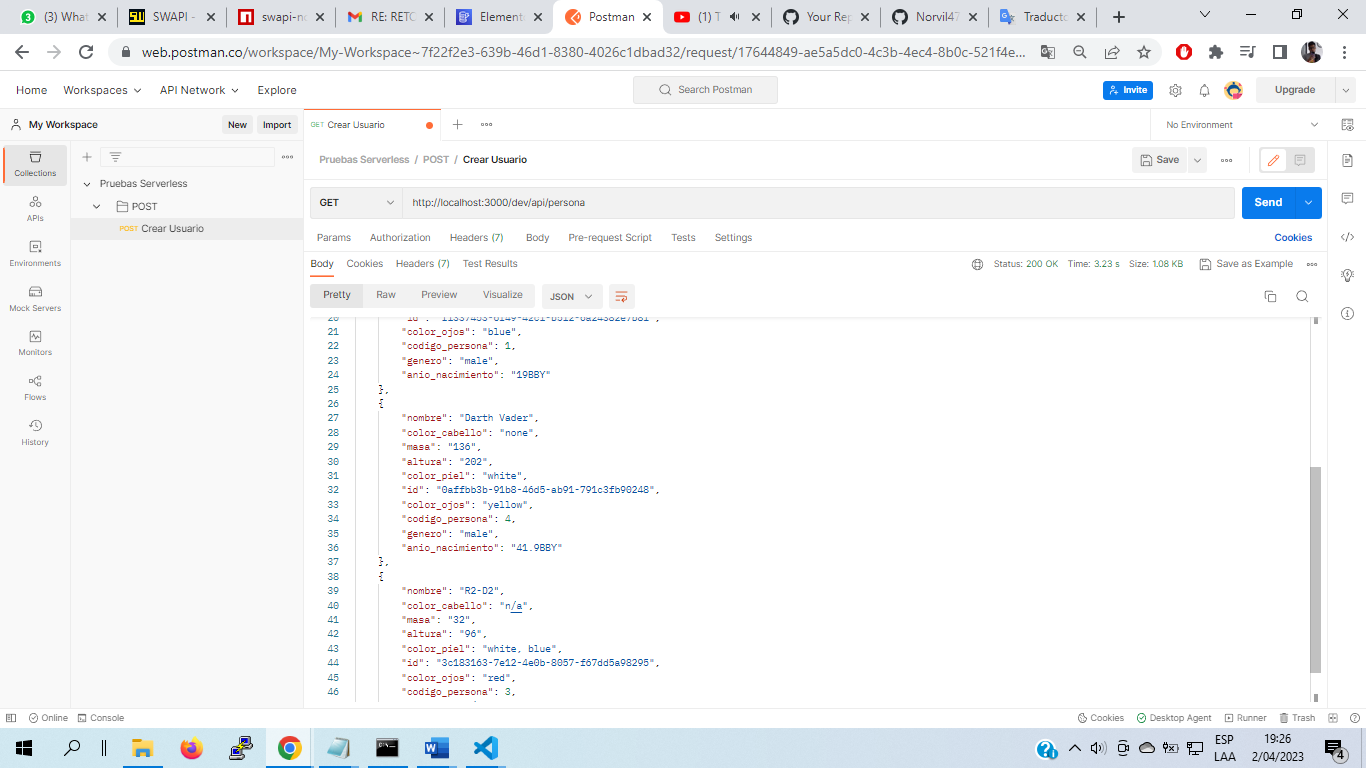
* Cabe resaltar, si se llega a enviar un ID que no se encuentre registrado en la SWAPI, se enviará un mensaje de alerta.



1. Listar todas las personas registradas.

rutaLocal: **http://localhost:3000/dev/api/persona**

Método: GET



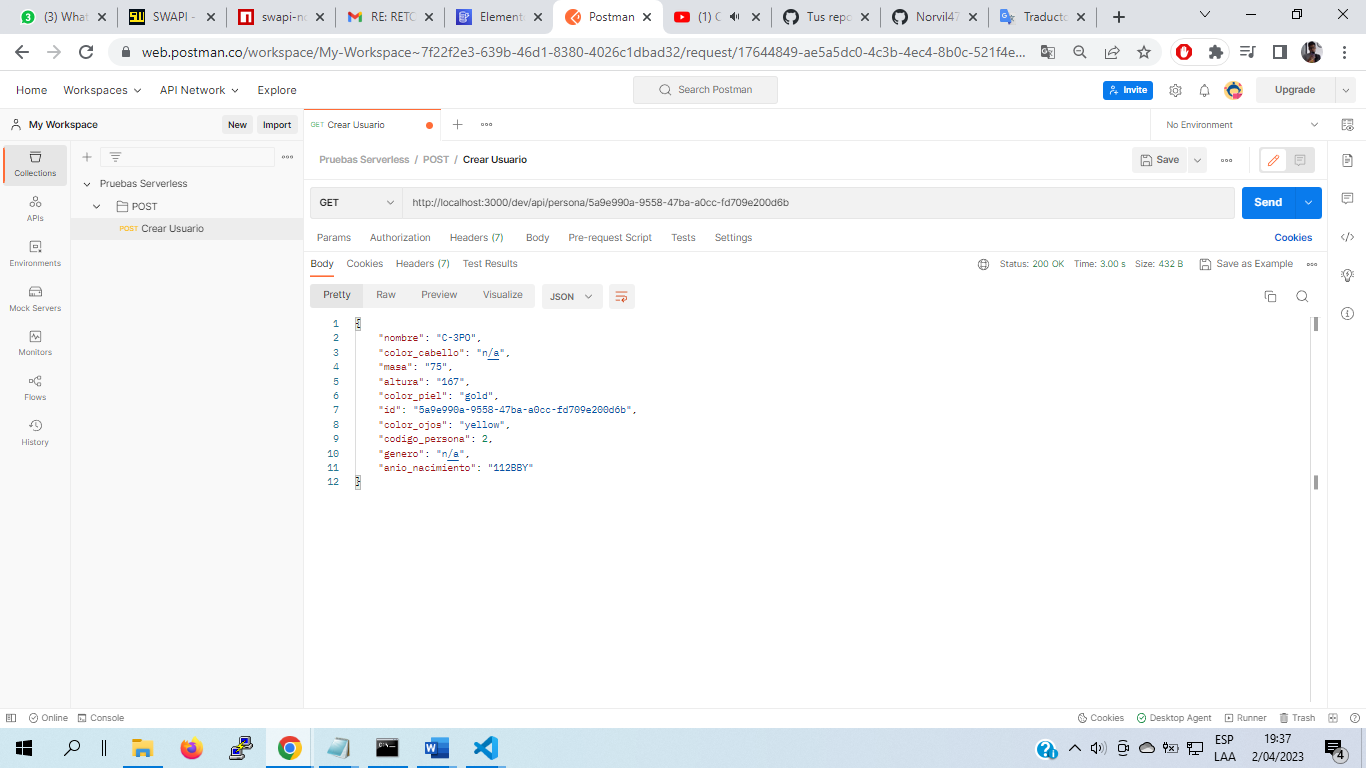
1. Traer datos de una persona según su id único registrado en DB.

rutaLocal: **http://localhost:3000/dev/api/persona/id**

Método: GET

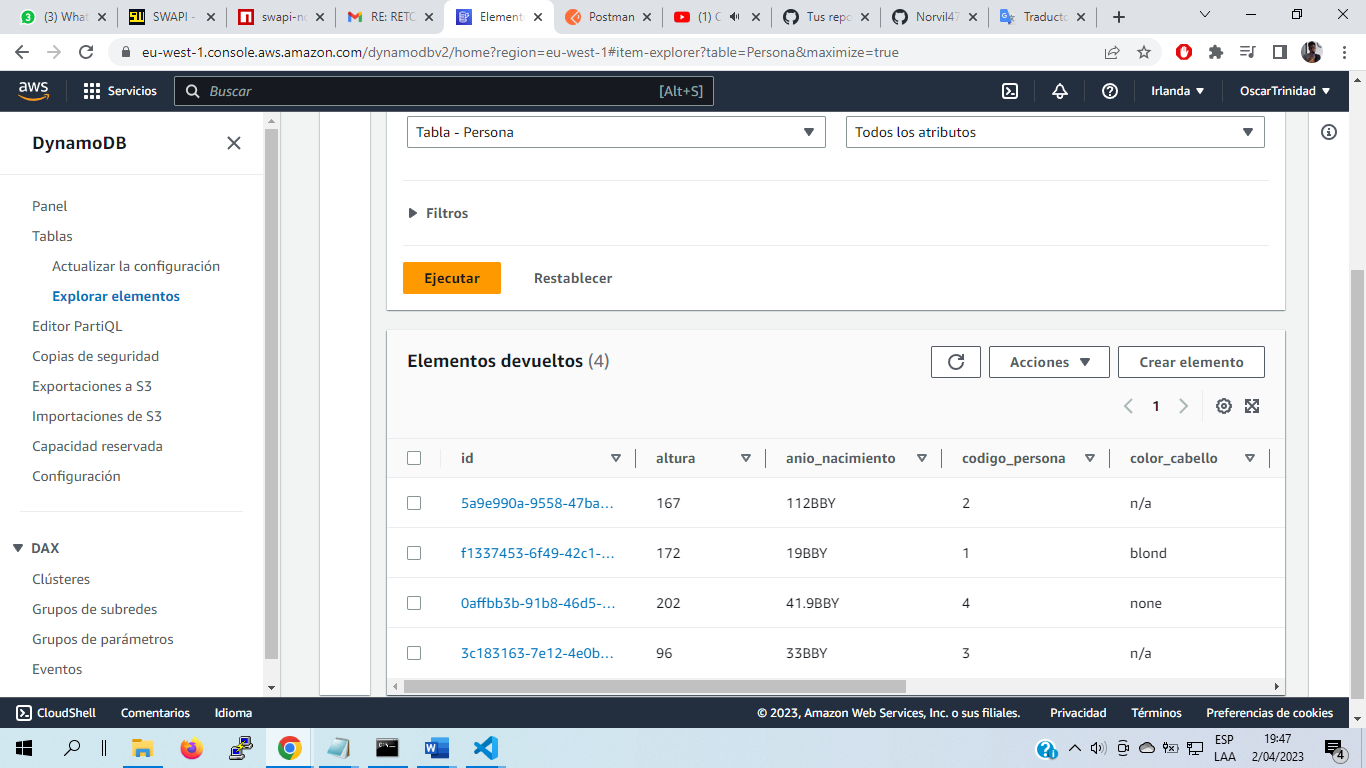
Proceso:

* Mandar el id único generado para el registro de Personas en la ruta por método GET, y la API te mandará la data de la persona con dicho id.



1. **Integración con una base de datos (DynamoDB o MySQL)**

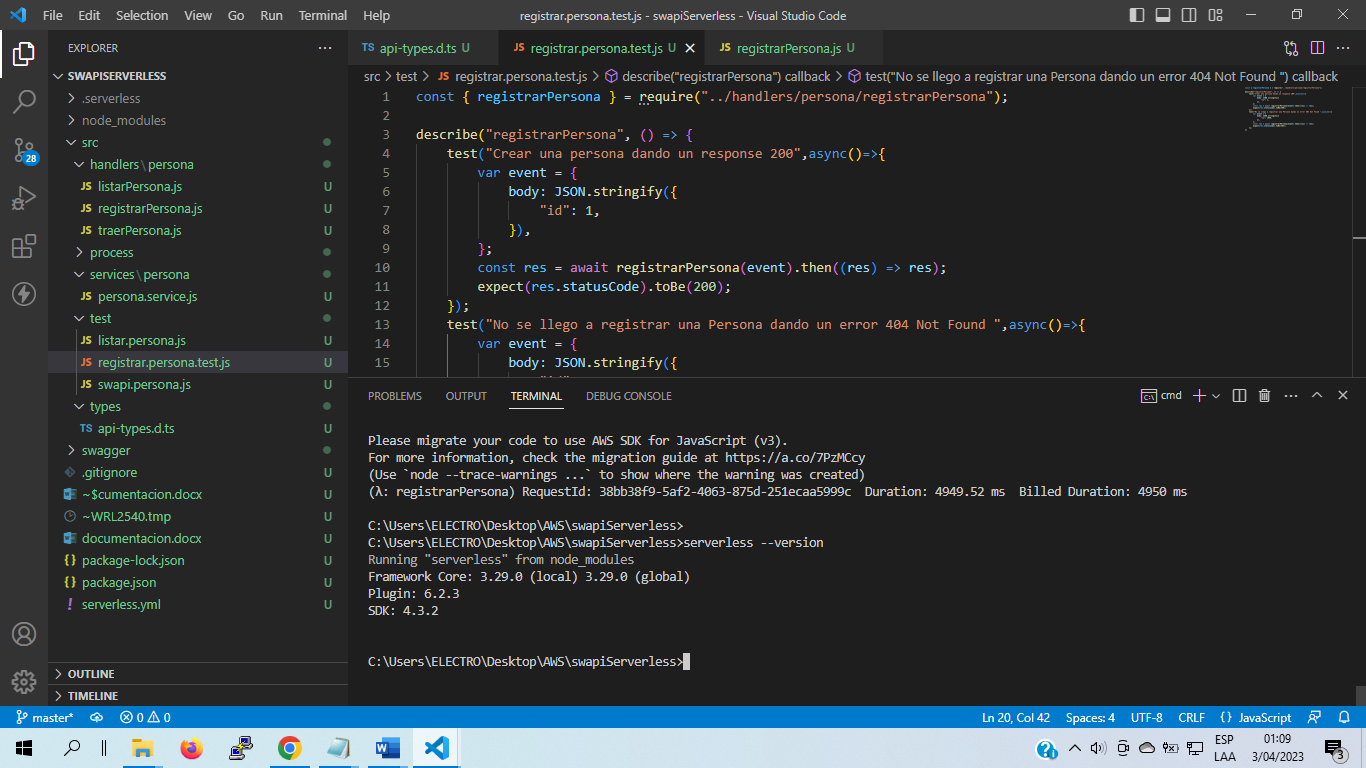
* Integración de una base de datos DynamoDB:

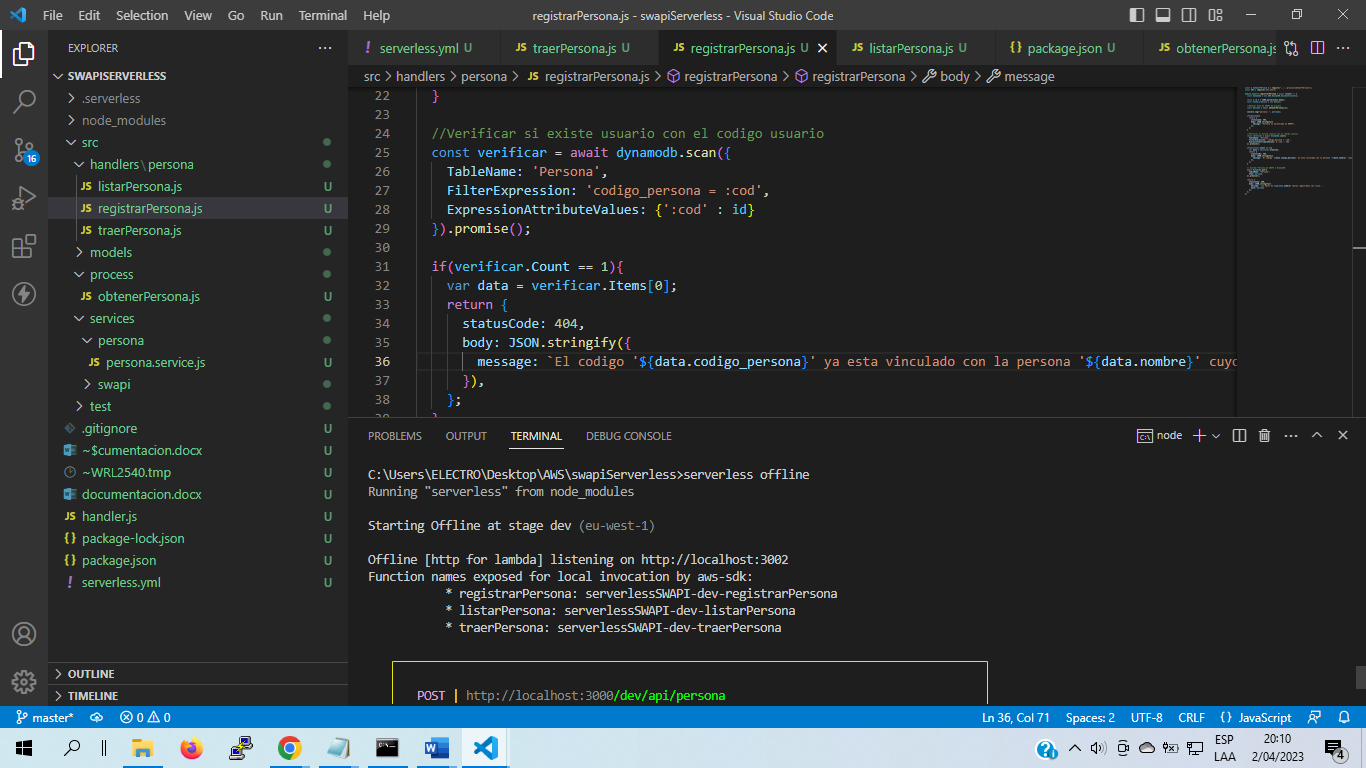


1. **Integración con SWAPI**

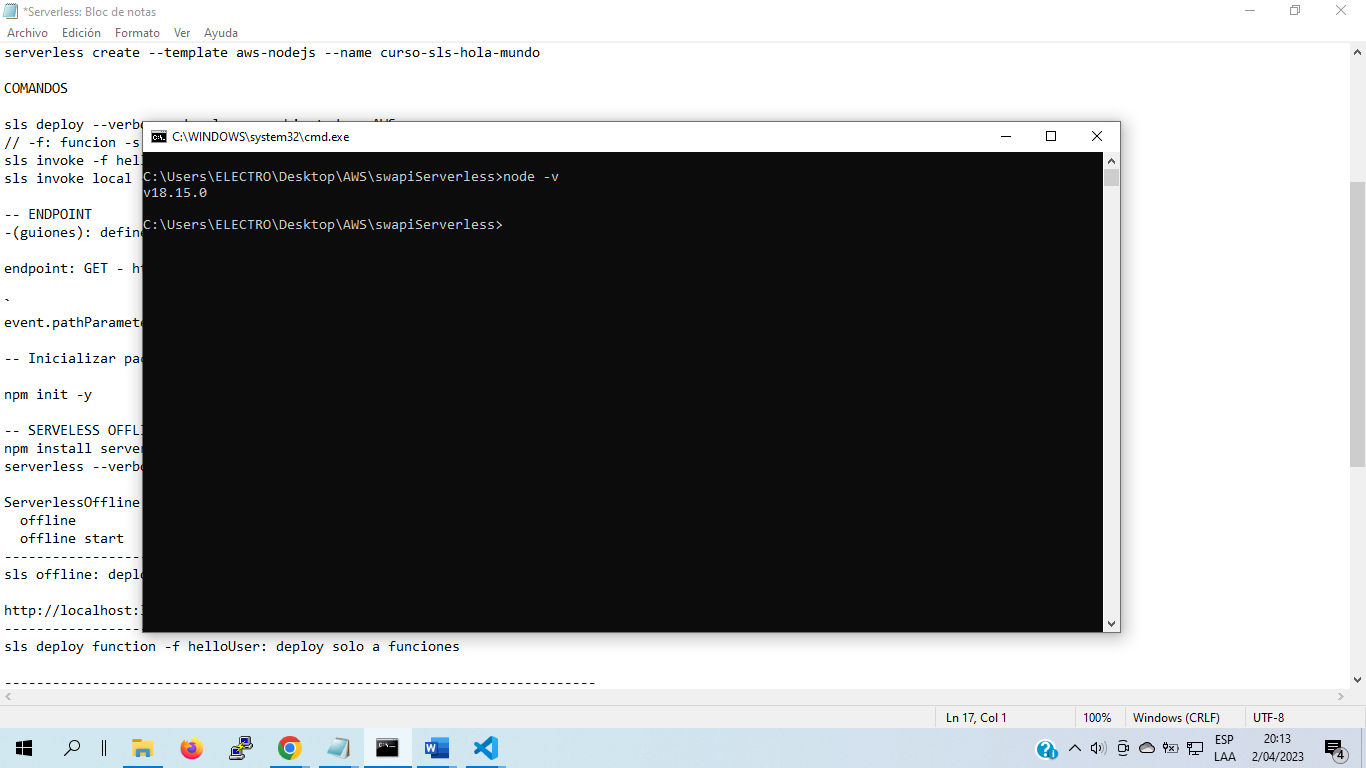
Se instalo la librería auxiliar de SWAPI para JavaScript: **swapi-node.**

1. **Uso de Serverless Framework**

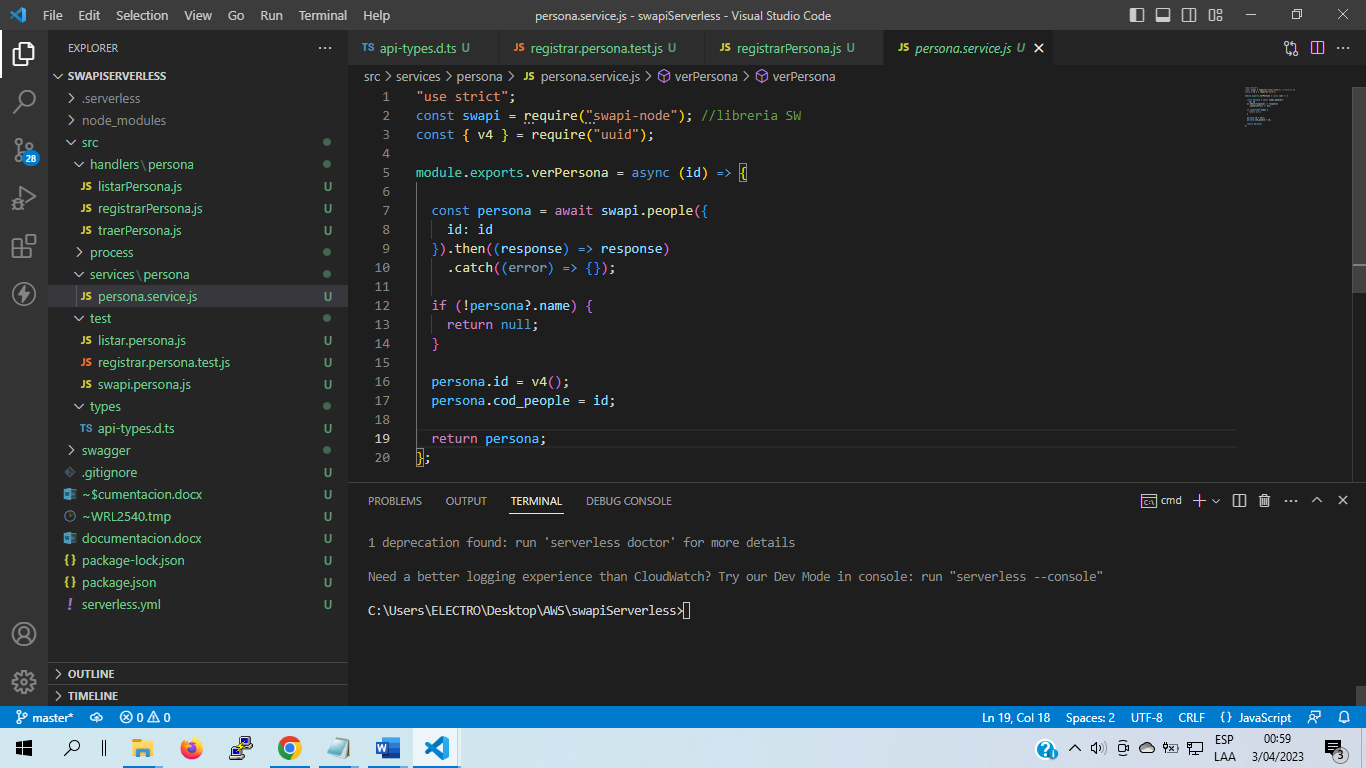




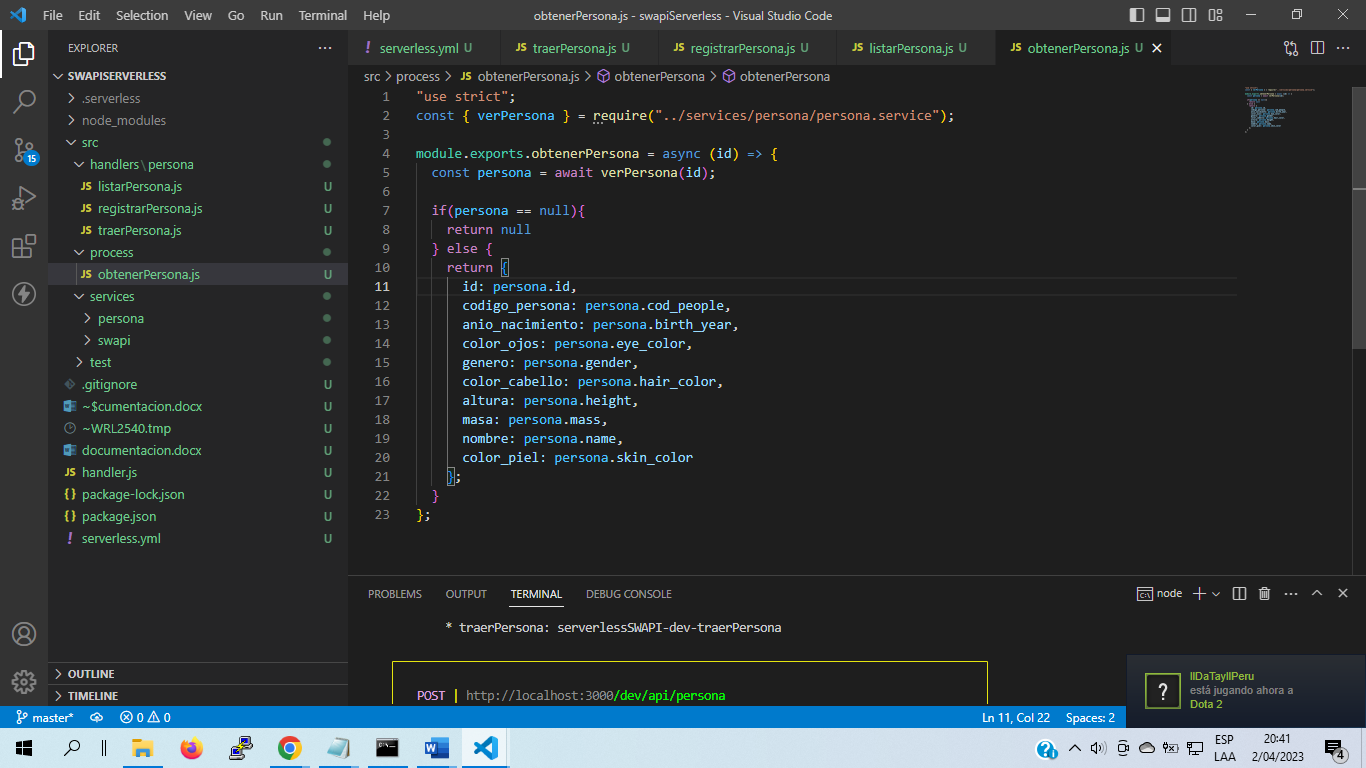
1. **Uso de Node.js**



1. **Respeto de las buenas prácticas de desarrollo**



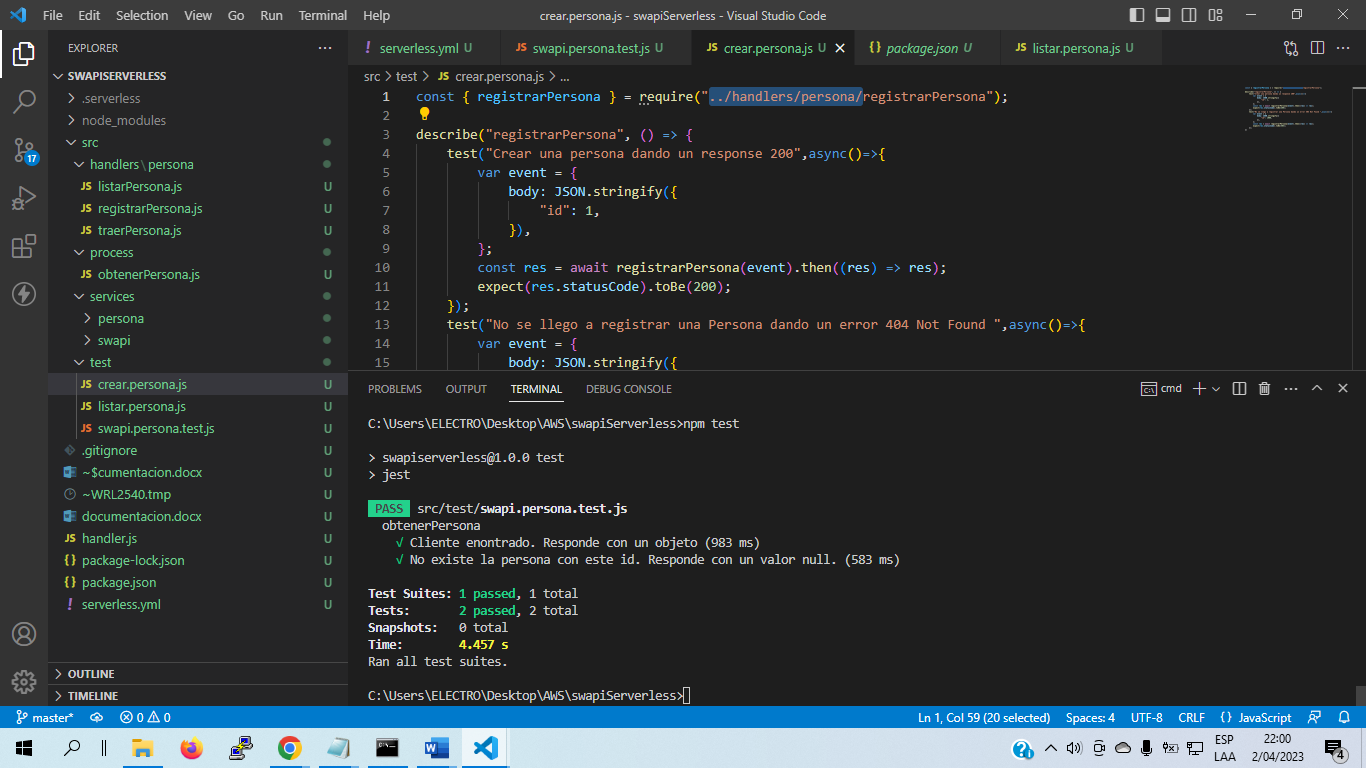
1. **Traducción de campos de inglés a español**



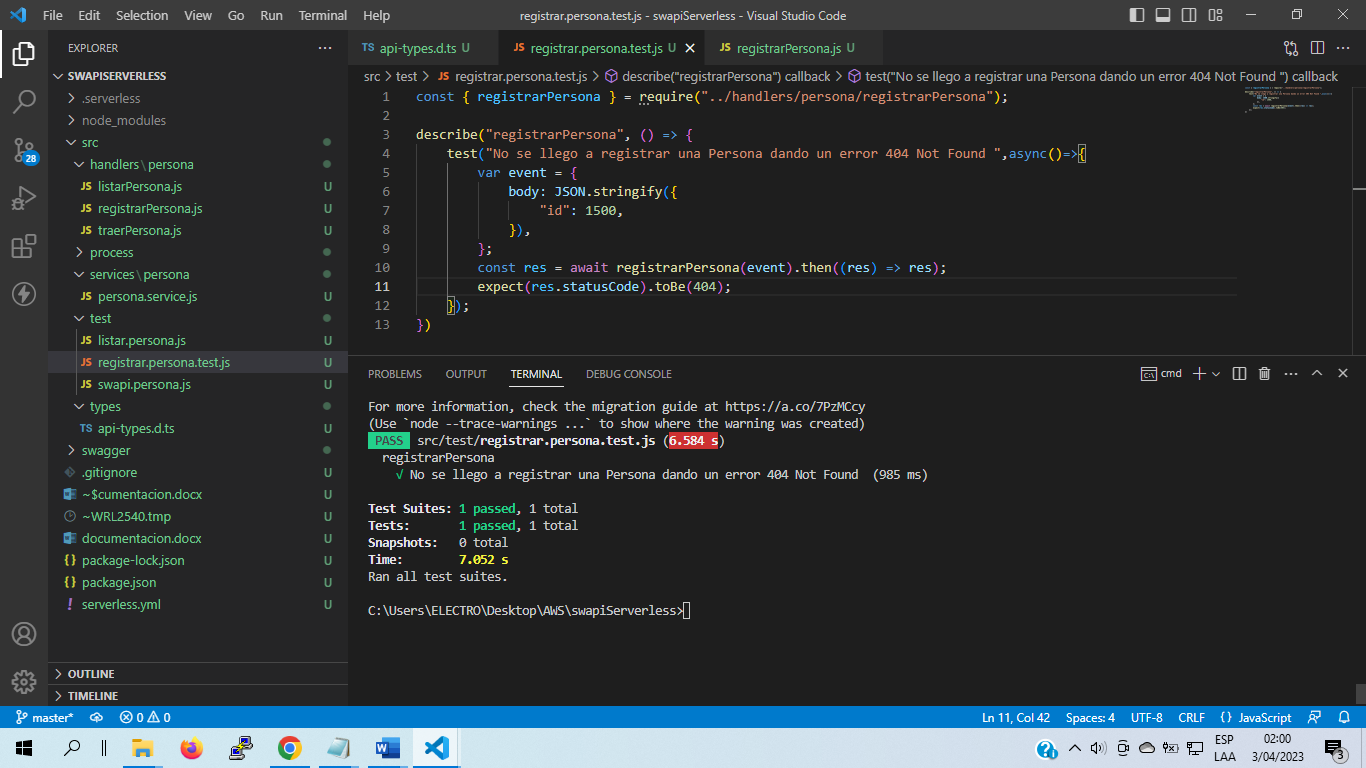
**PRUEBAS UNITARIAS**

Las pruebas unitarias fueron empleadas instalando la librería Jest

**Archivo:** swapi.persona.test.js

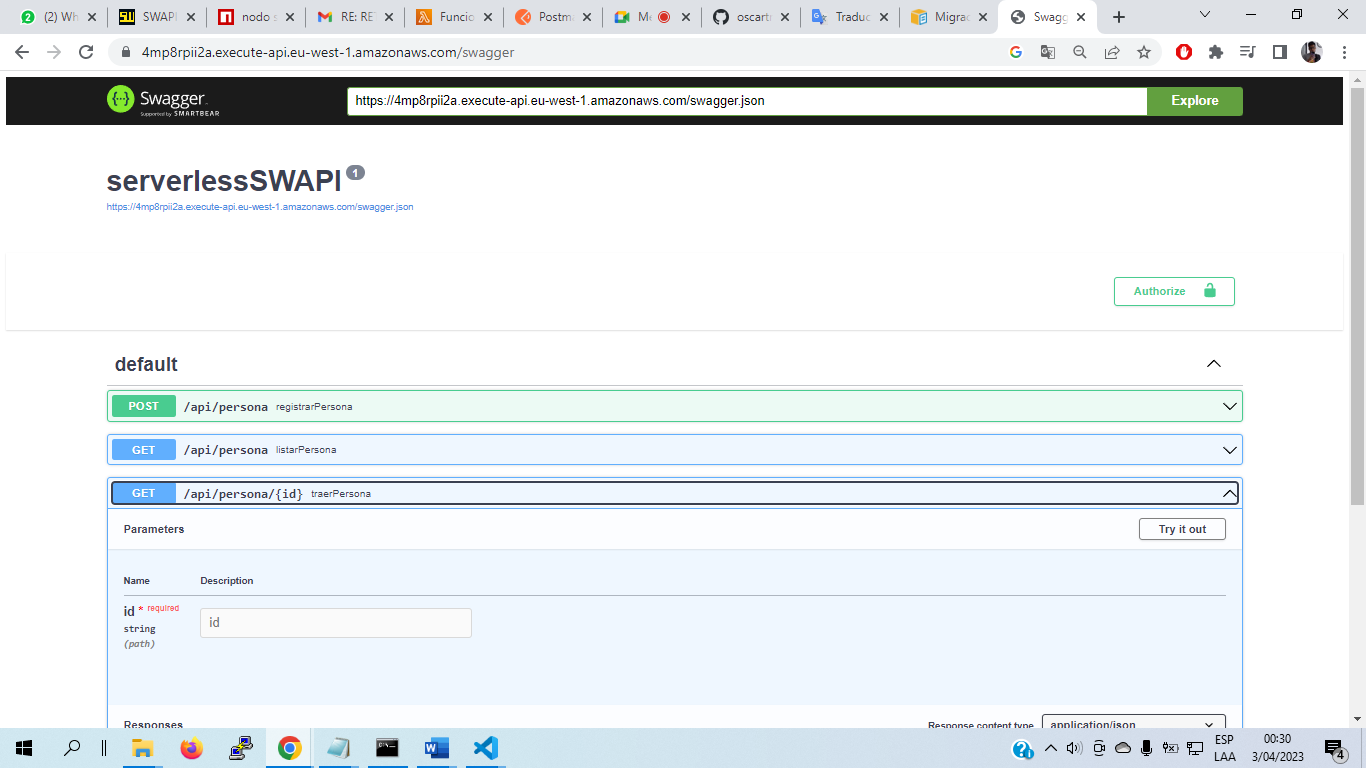
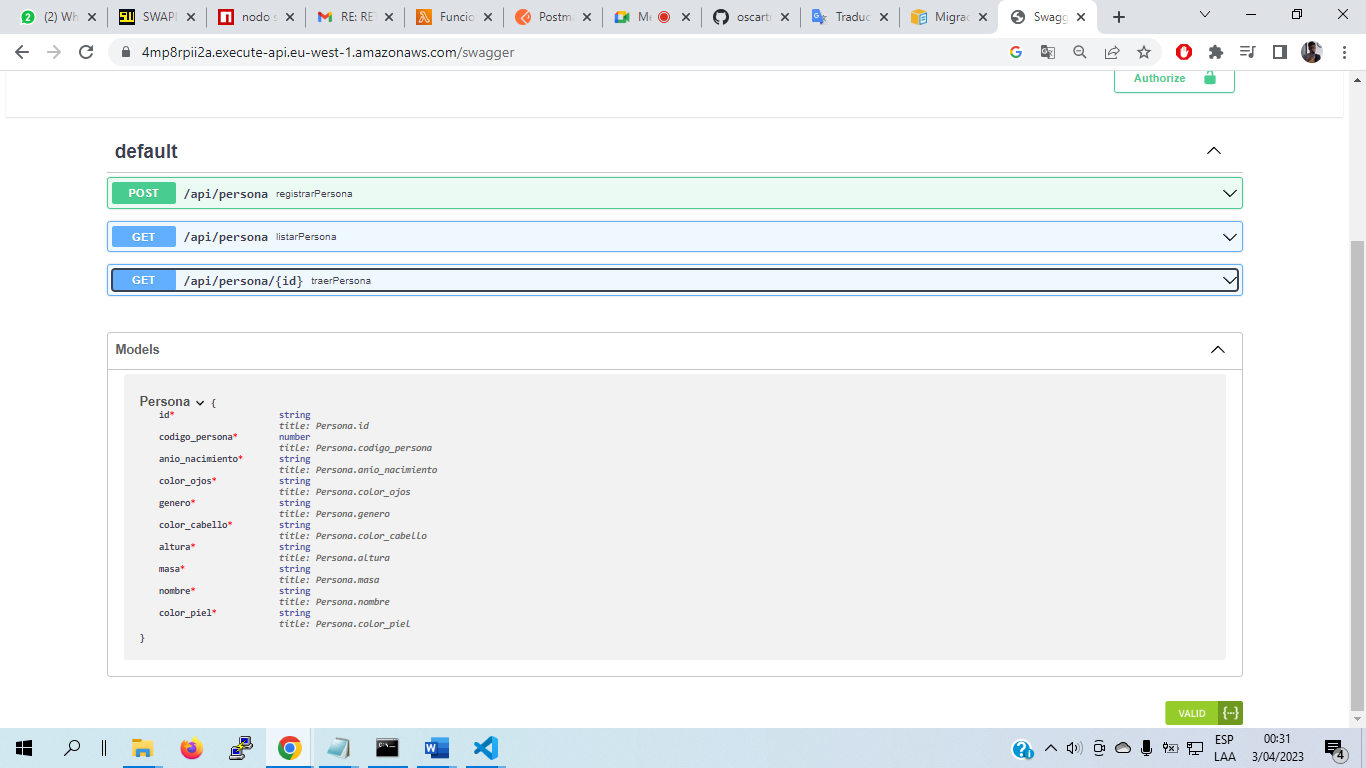


**Archivo**: registrar.persona.test.js

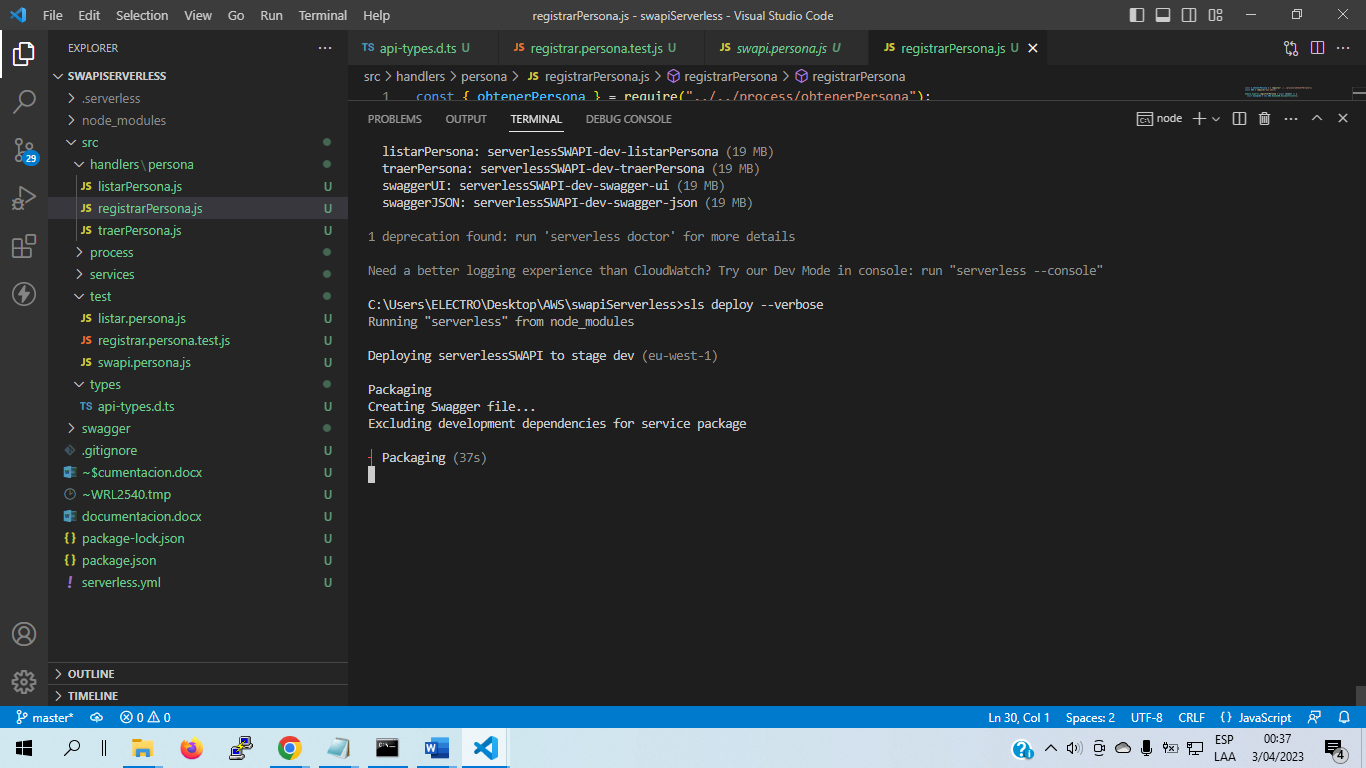


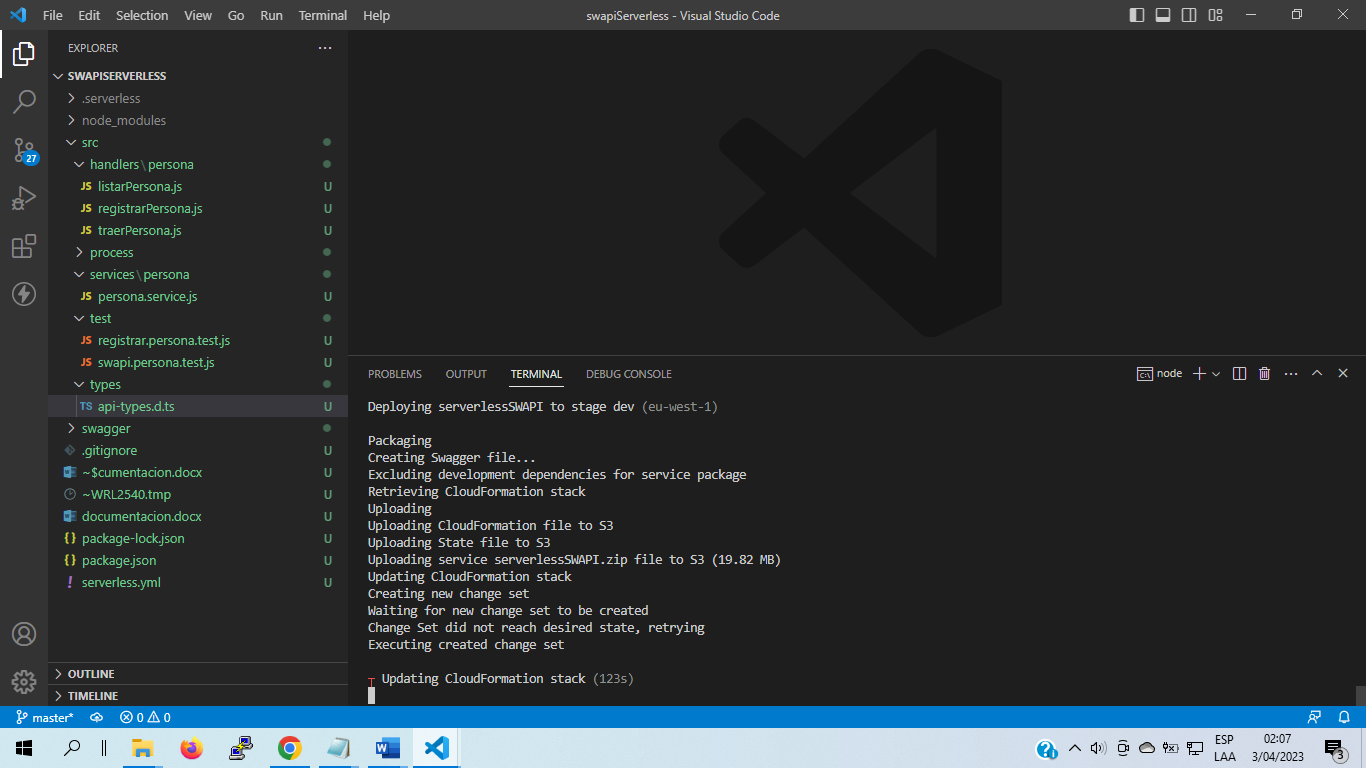
**SWAGGER**

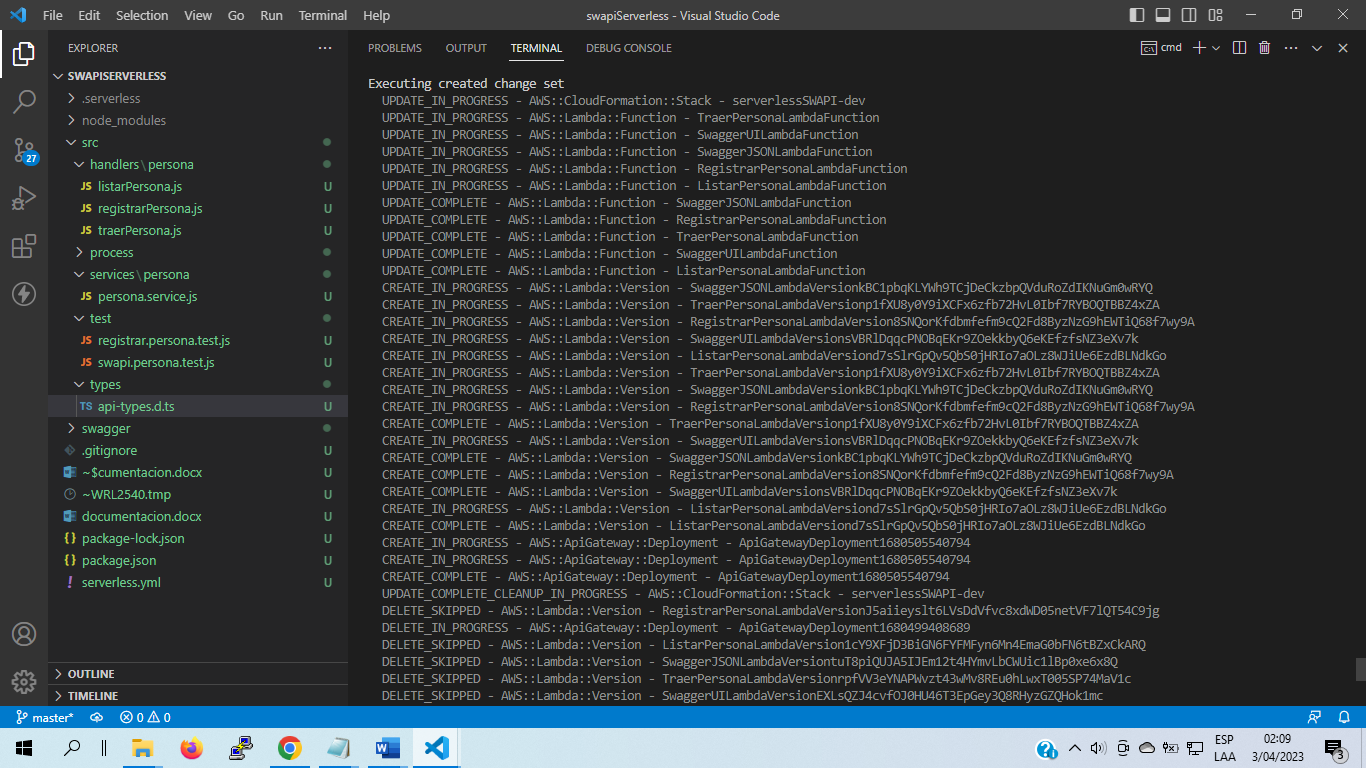
Configuración:

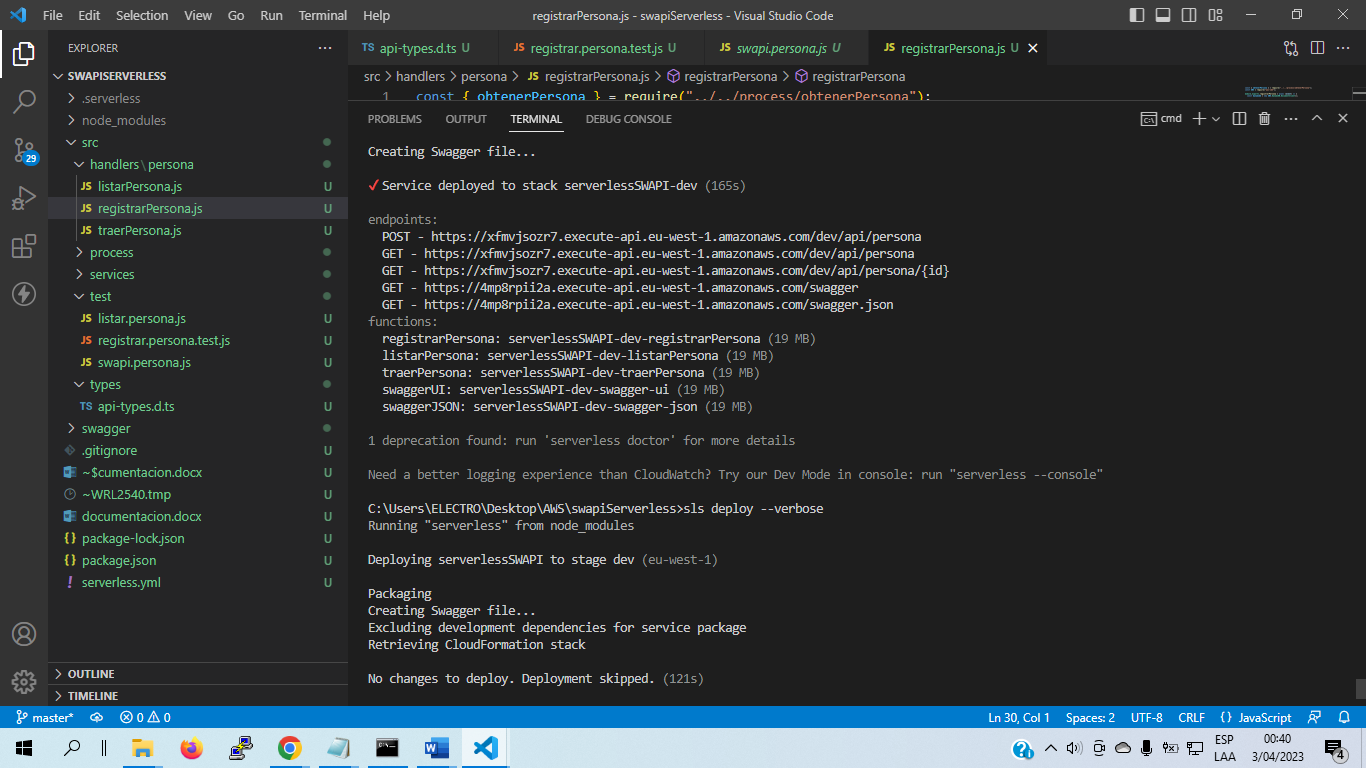


**Desplegar sin errores en AWS con el comando deploy del framework serverless**









**AWS Lambda**

