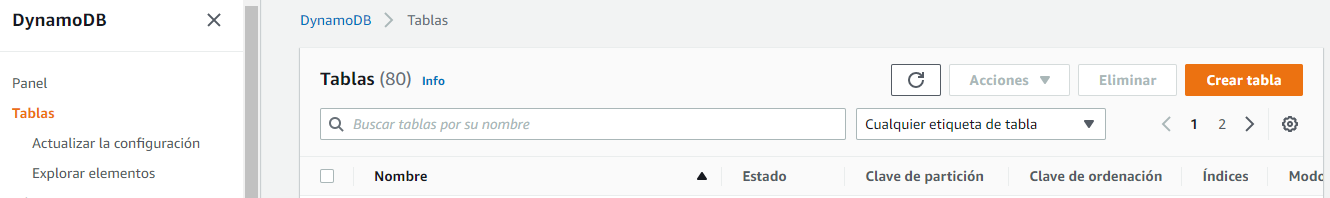
Entregable Prueba Técnica por Jomar Arley Vega Cardona

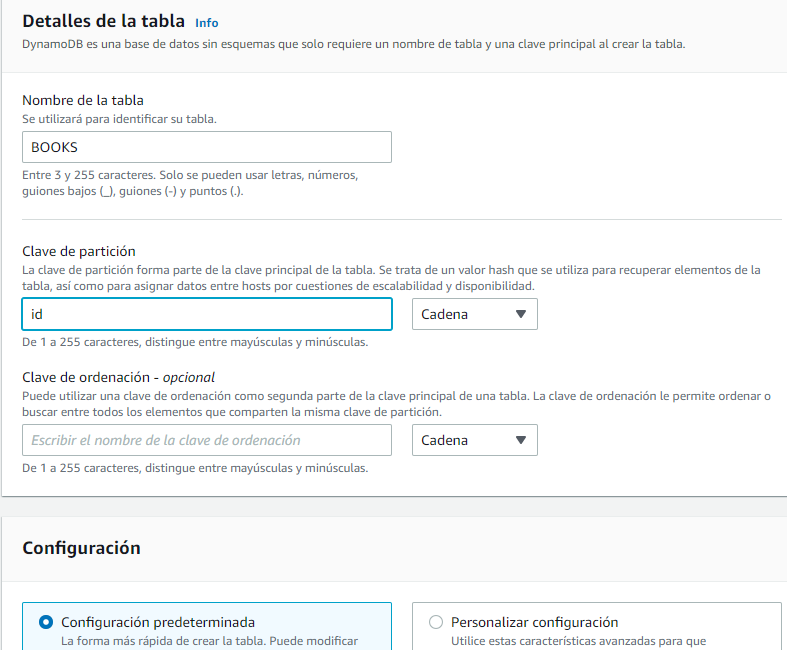
Para esta prueba se trabajo en la cuenta:



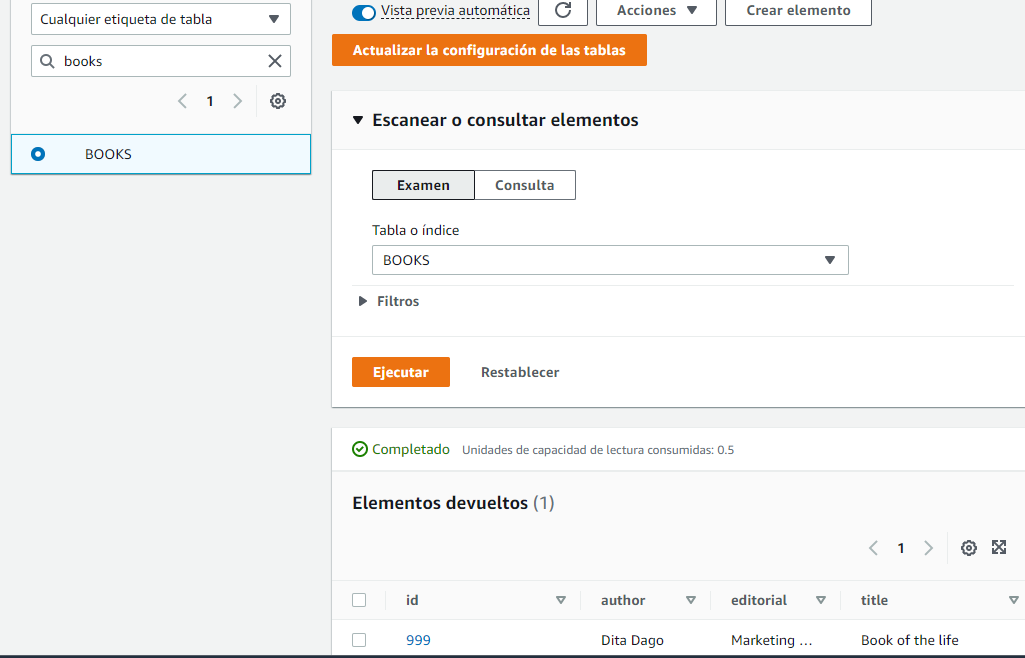
1. Ingresar a DynamoDB en aws y se selecciona la opción de Tablas y se da clic en crear Tabla



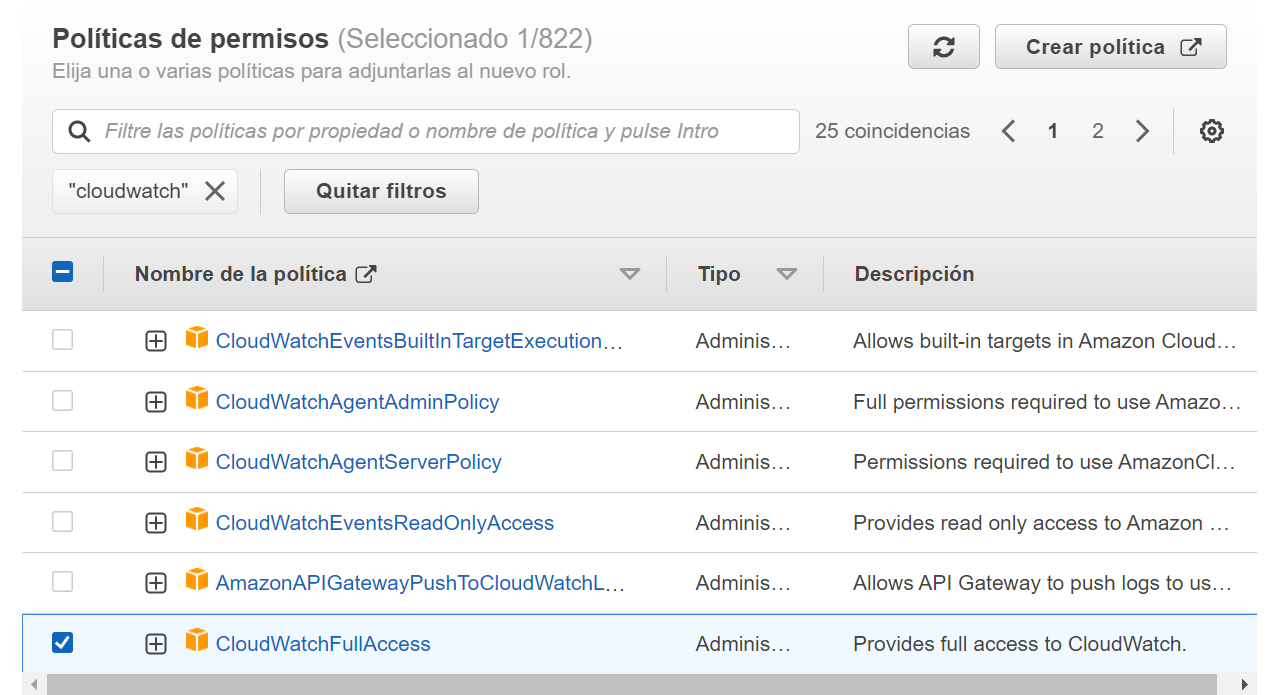
1. Se crea Bd en dynamoDB con el nombre de BOOKS y primary key “id” , predeterminado



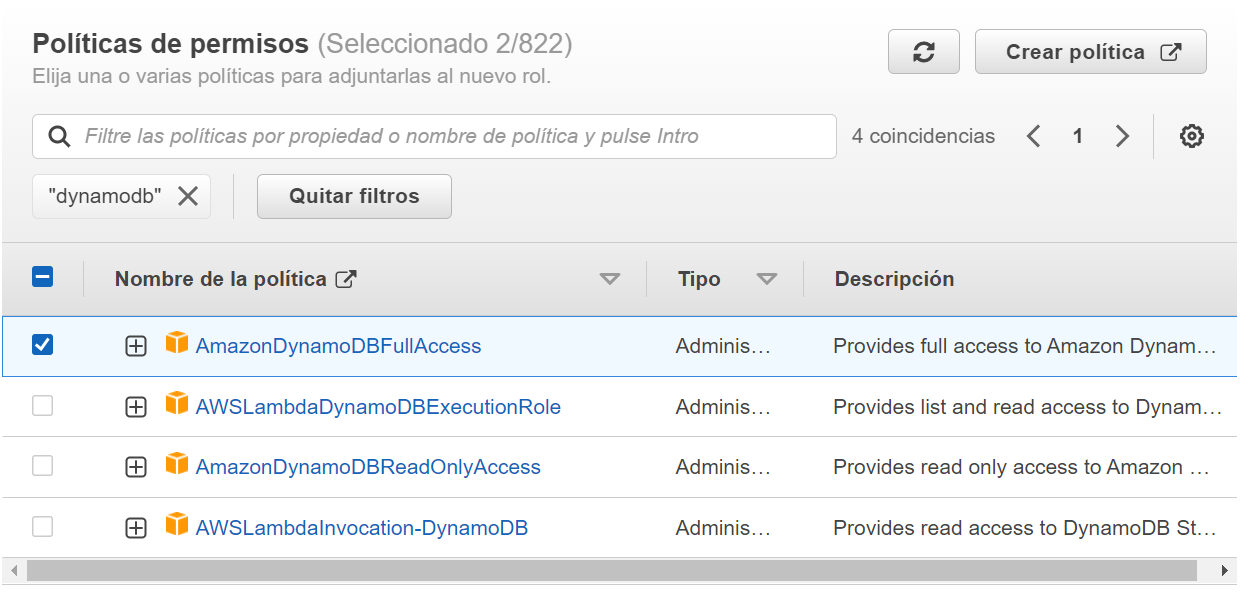
1. Clic en guardar y se puede ver la tabla creada



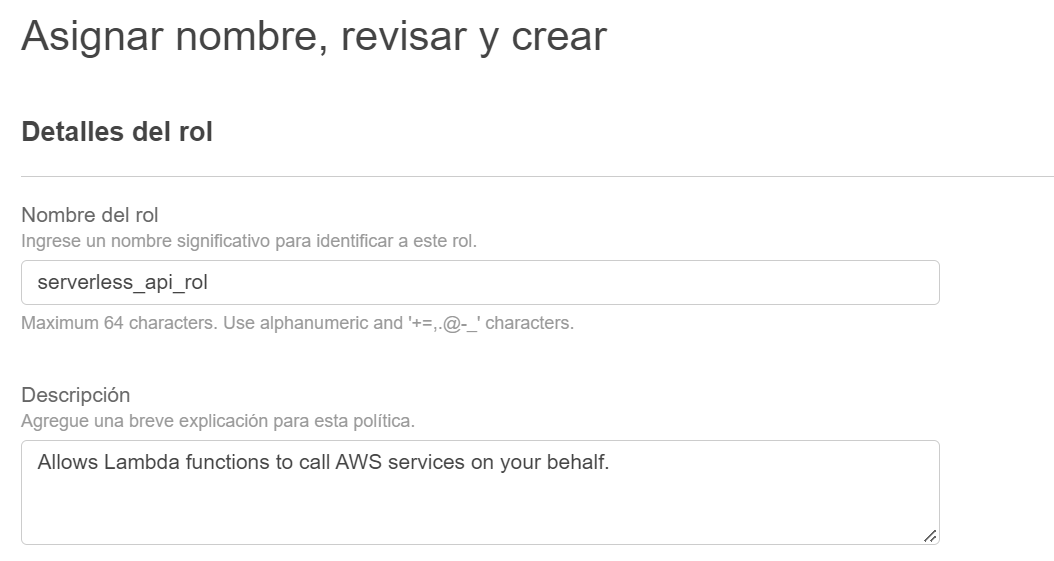
4 ) Para ingresar los datos a esta tabla se rea un Rol en IAM , se seleccionan los permisos de **CloudWatchFullAccess**



Y el permiso de **AmazonDynamoDBFullAccess**

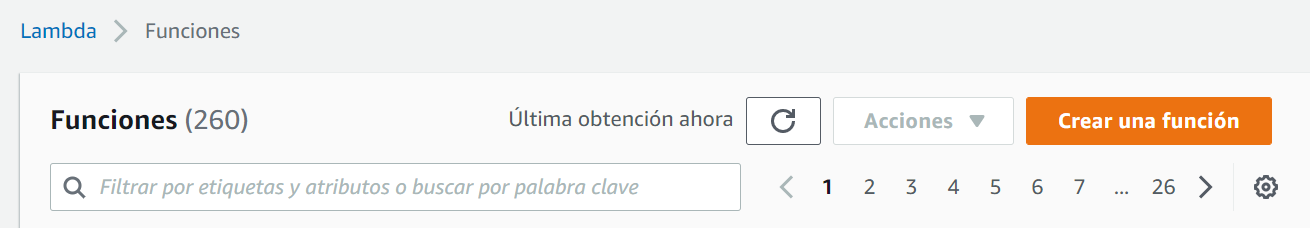


5 ) Se crea un rol con estos dos permisos en este caso el rol se llama **serverless\_api\_rol** y clic en crear Rol

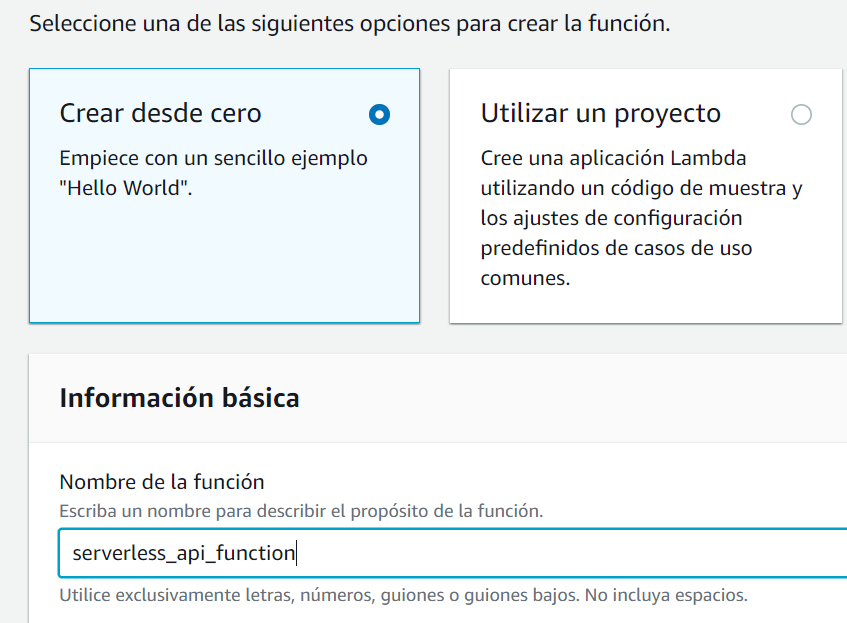




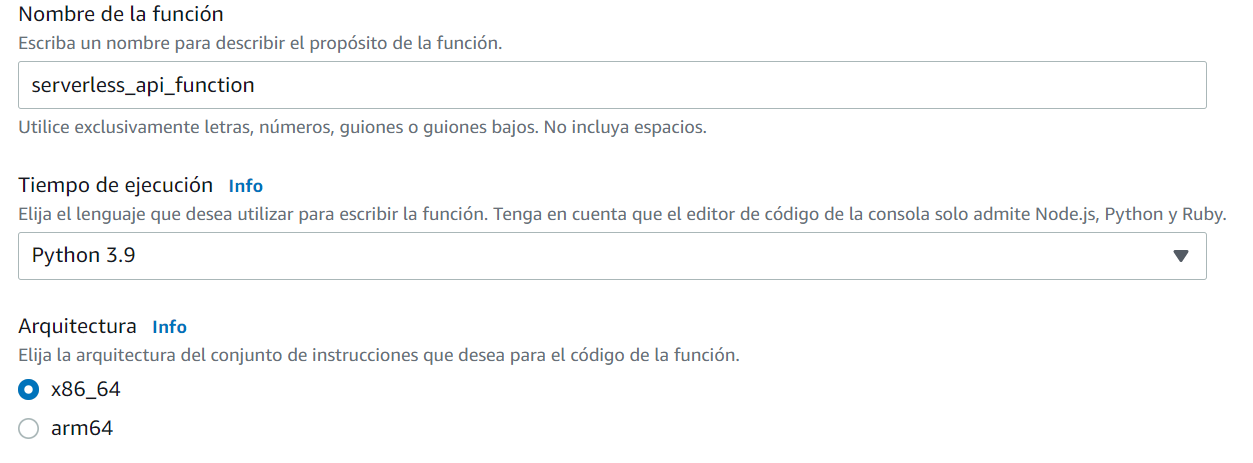
6 ) Creación de Lambda, se ingresa al modulo de lambda de aws y seleccionamos crear funcion



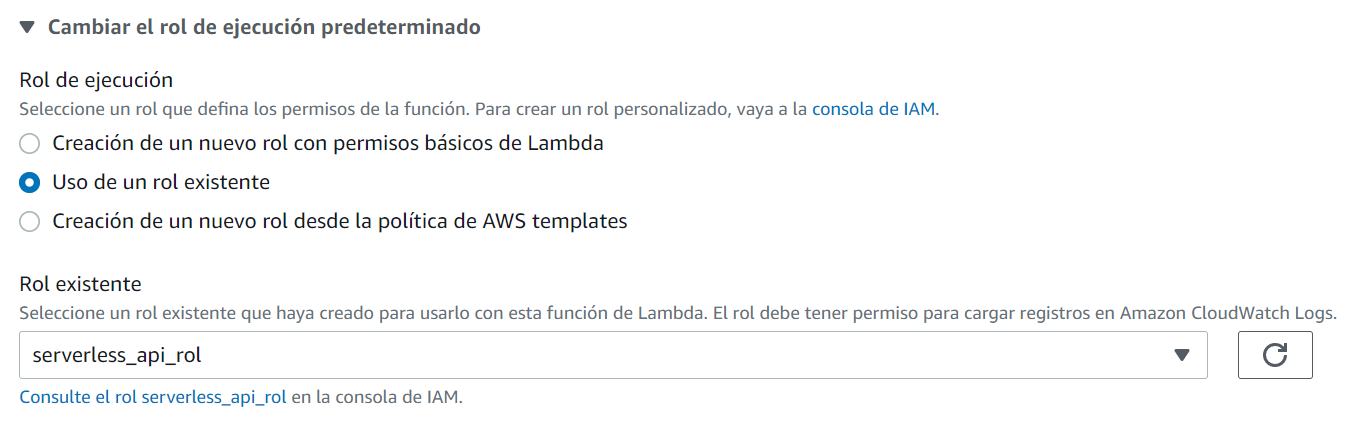
Para este ejemplo la lambda se crea desde Cero y se le da el nombre de **serverless\_api\_function**



Se selecciona el tipo de lenguaje que tendrá la lambda en este caso Python 3.9



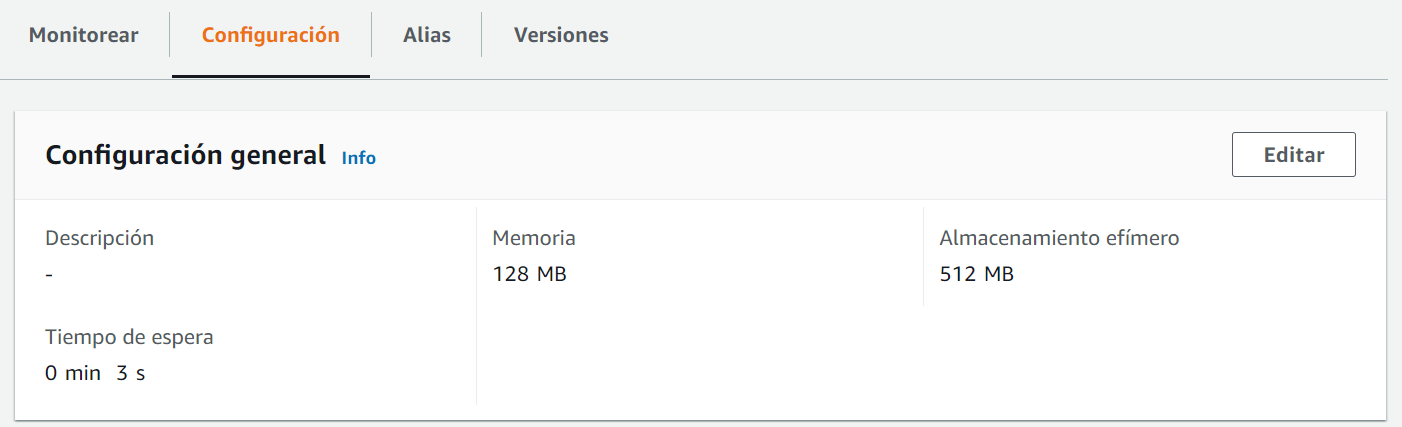
Y seleccionanos uso de un rol existente y seleccionamos el anteriormente creado en AIM **serverless\_api\_rol y le damos crear función**



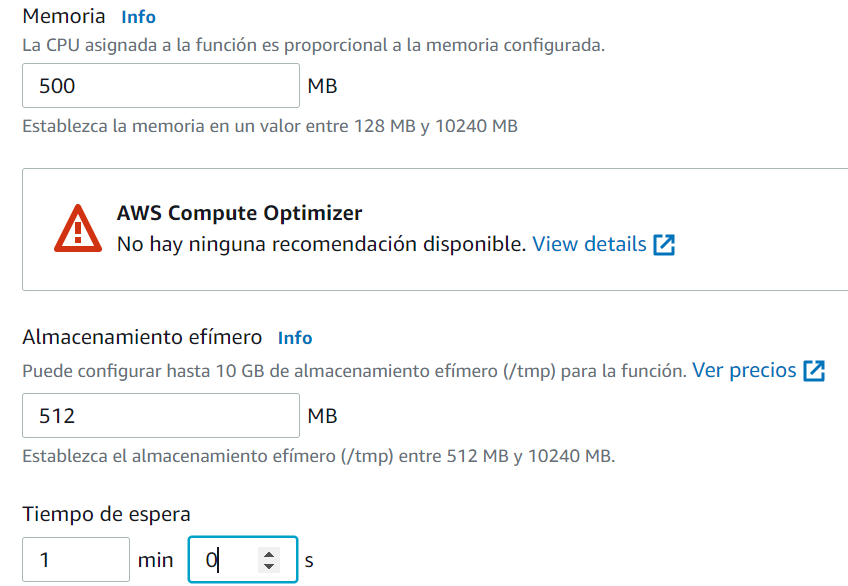
Así aparece cuando se creamos la lambda



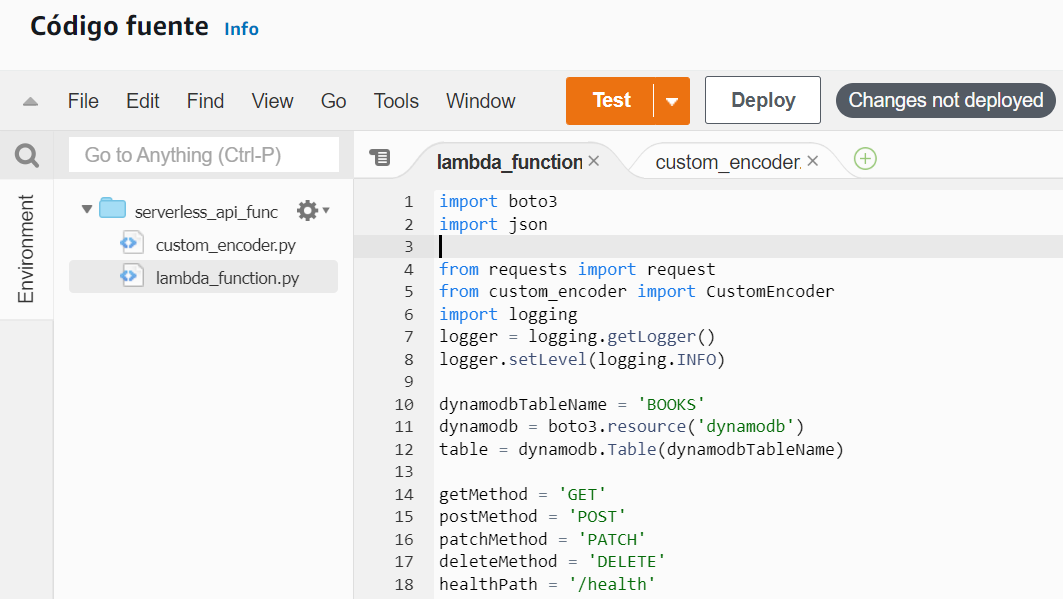
Vamos a configuración y cambiamos los parámetros de memoria asignada a la CPU por 500 MB



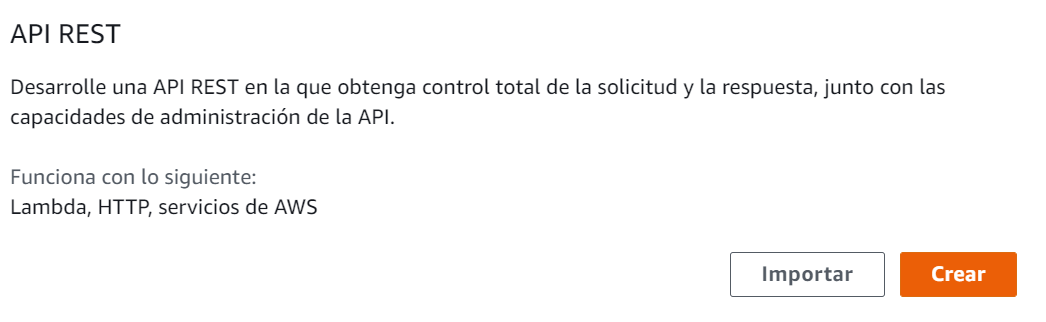
En tiempo de espera colocamos 1 minuto



Se agrega el código fuente a la lambda en el archivo lambda\_function.py y también se crea otro archivo custom\_encoder.py y se da clic en Deplo para guardar los cambios



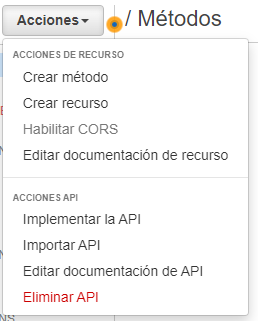
7) creación de api rest, seleccionamos **API GATEWAY** y seleccionamos API REST y crear



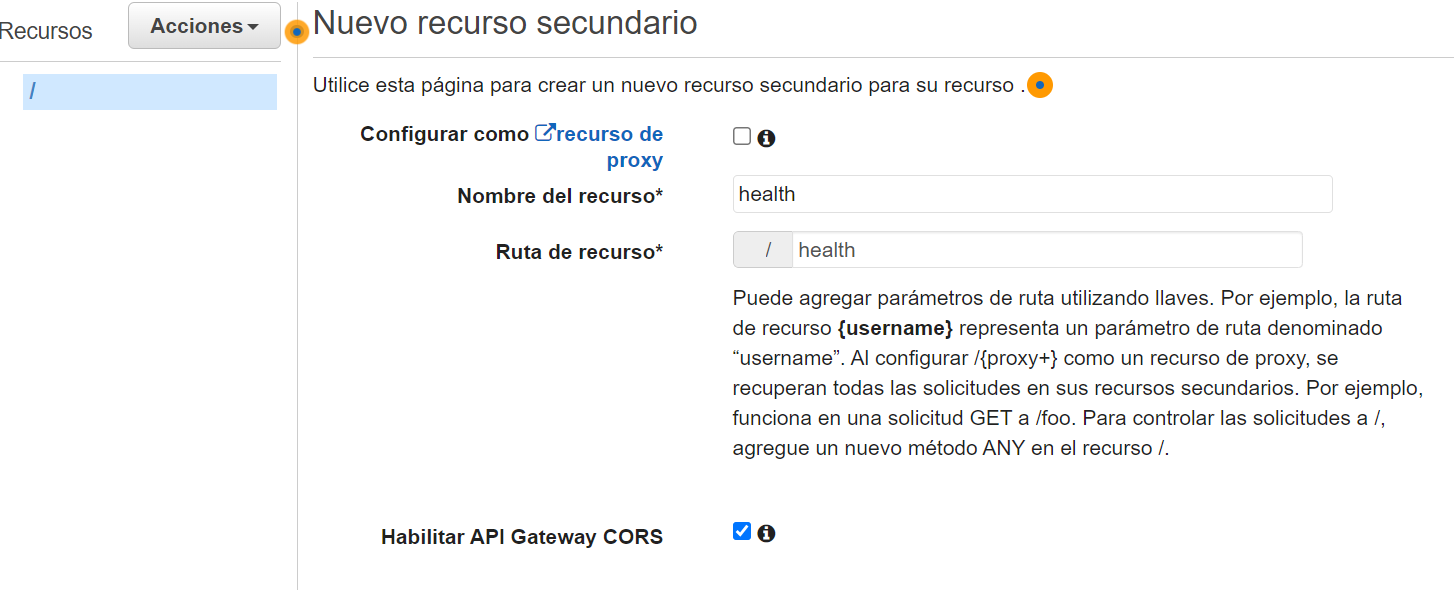
Seleccionamos **Rest** y **Api Nueva** y colocamos el nombre de la api, en esta caso se llama serverless\_api y en punto de enlace Regional y clic en crear API



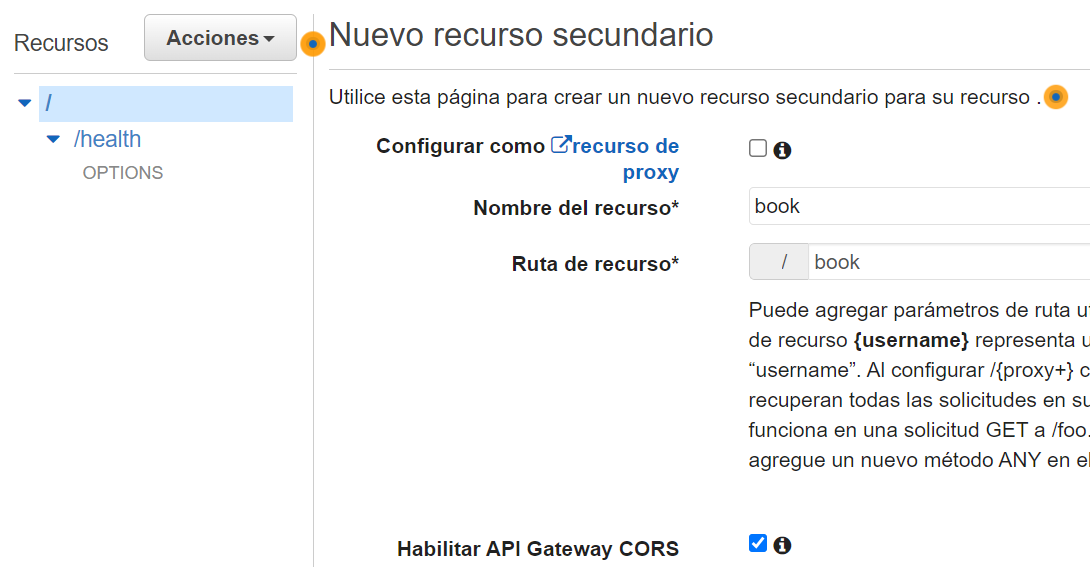
Una vez creada ingresar a la api, nos ubicamos en el / y seleccionamos acciones y escogemos crear recurso



se crea un recurso llamado Health que verificar la puerta de enlace o la API si esta activa **y se marca la opción habilitar API Gateway CORS**



Se agrega un nuevo recurso, se da clic en acciones y se agrega la opción **OPTIONS** con el nombre de recurso **book y se marca la opción habilitar API Gateway CORS, este** Recurso para consultar un solo libro



Se procede a crear todos los recursos para book los cuales son:

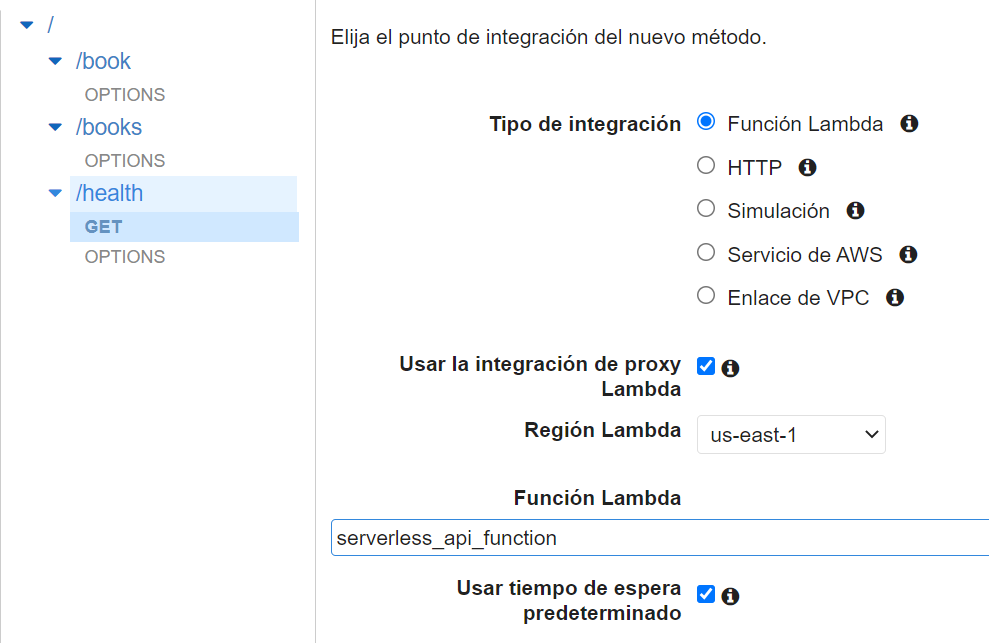
GET para obtener el valor de un libro de la BD de dynamoDB

OPTION para describir las comunicaciones entre los enlaces

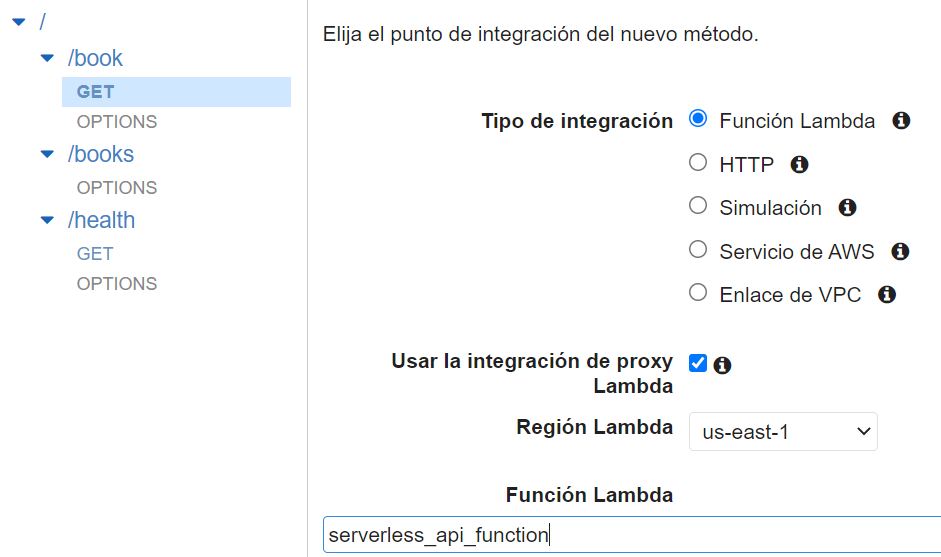
PATCH para actualizar un valor en la BD de DynamoDB

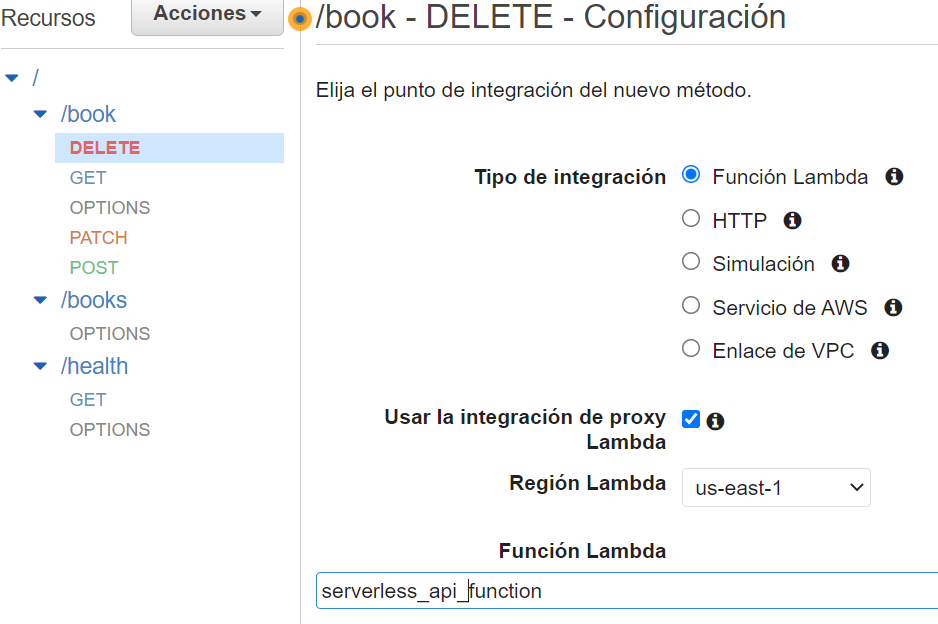
POST para agrega un nuevo libro la BD de dynamoDB

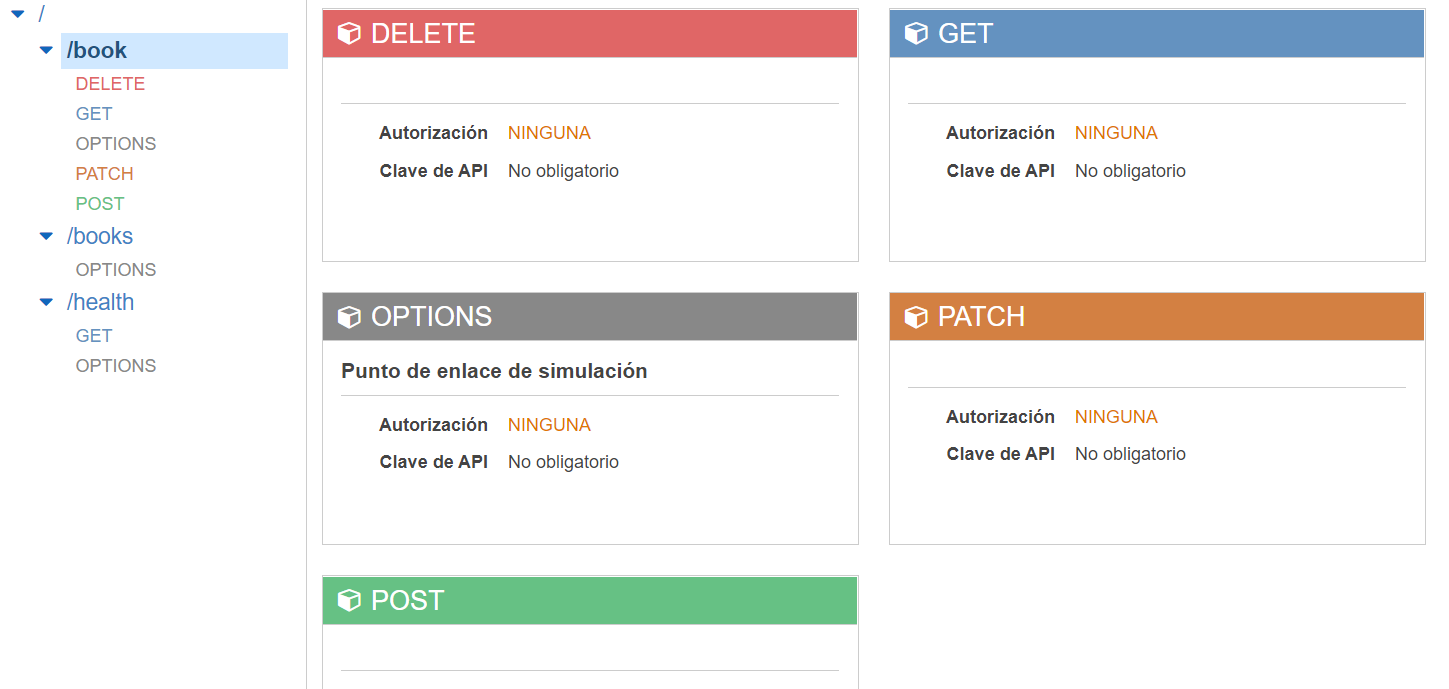
DELETE para eliminar un valor completo de la BD de DynamoDB

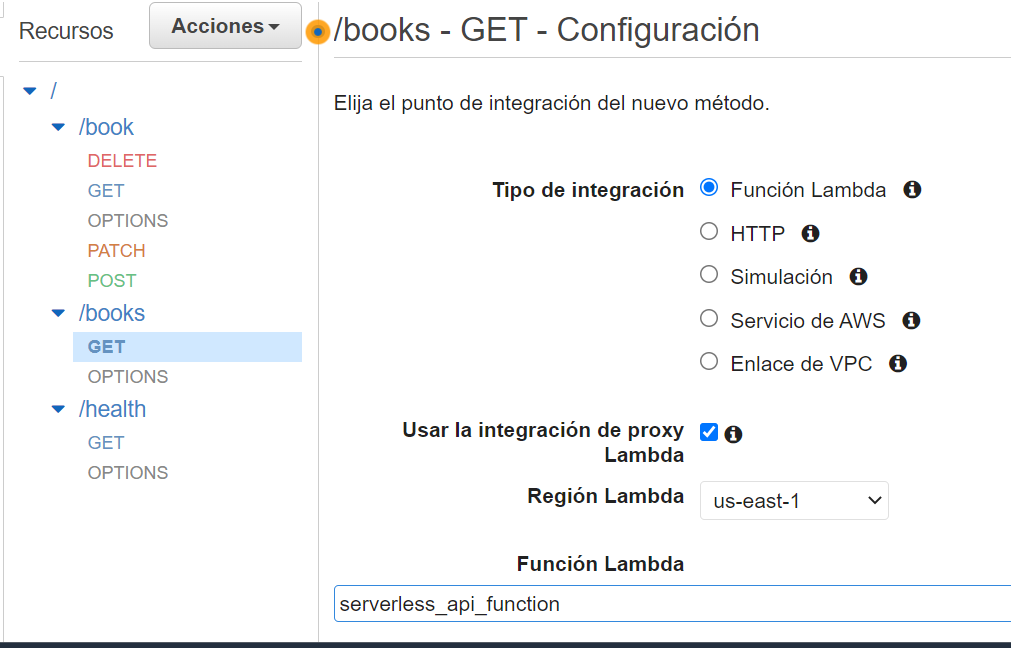


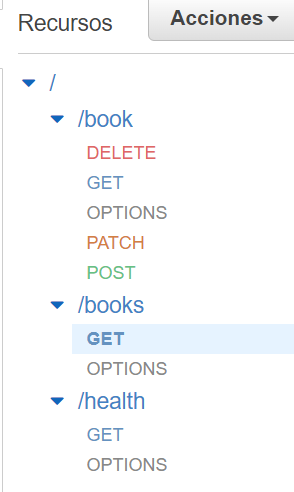
Cada vez que se agrega un método se selecciona que el tipo de integración en Función Lambda , la región y asociado a que función lambda en este caso a la ya creada **serverless\_api\_function**



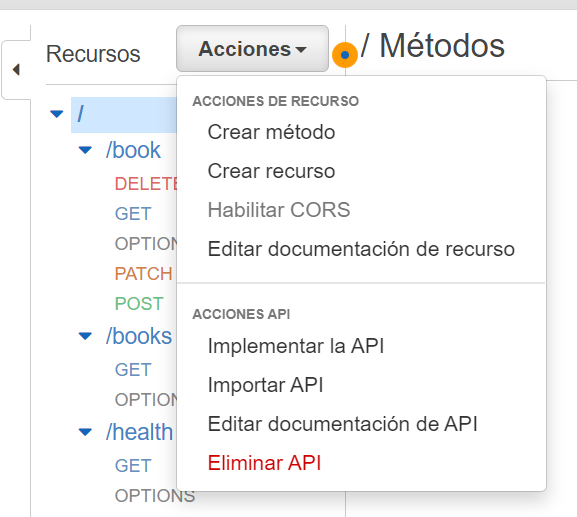




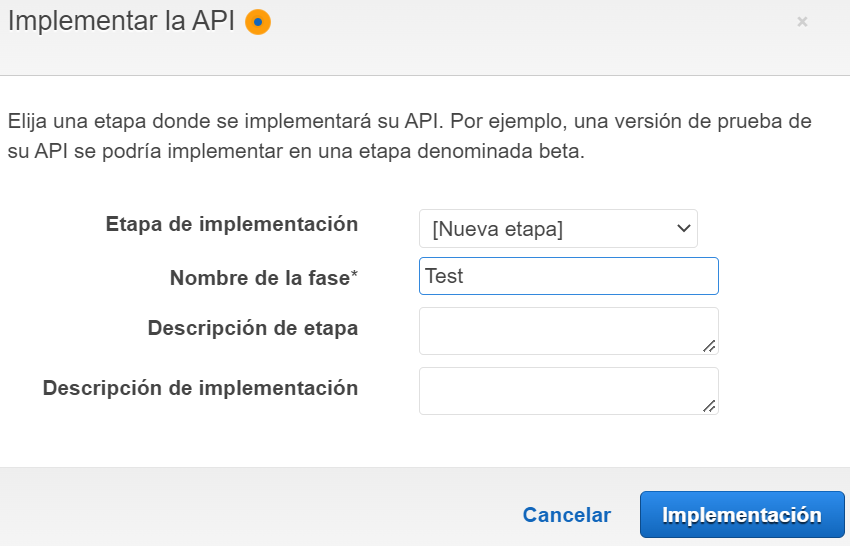


Este serial el resultado final de la creación de los recursos y métodos

Luego seleccionamos Implementar API



Seleccionamos nueva etapa de implementación y el nombre de fase, puede se pruebas o producción y clic en implementacion

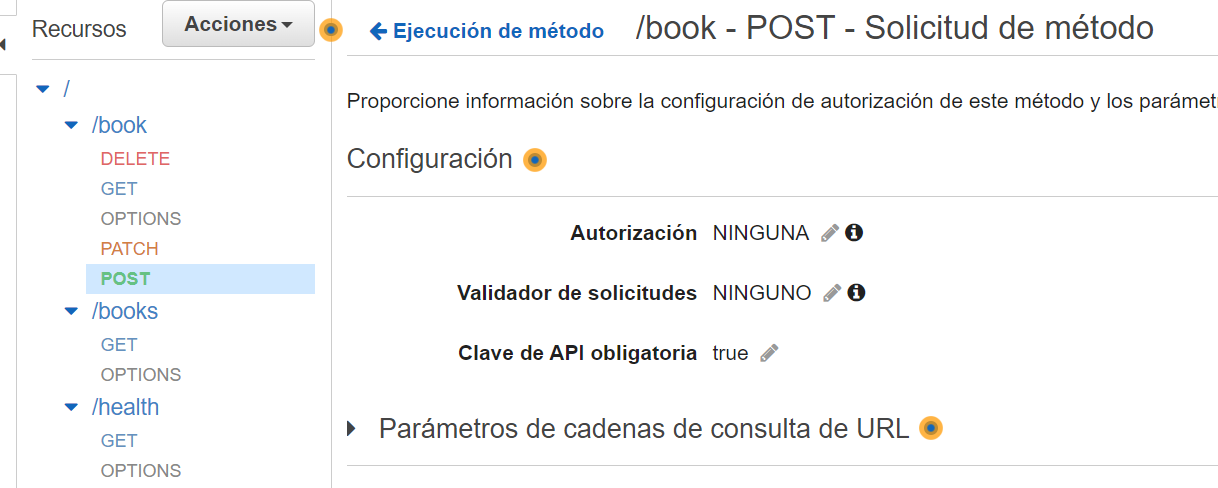


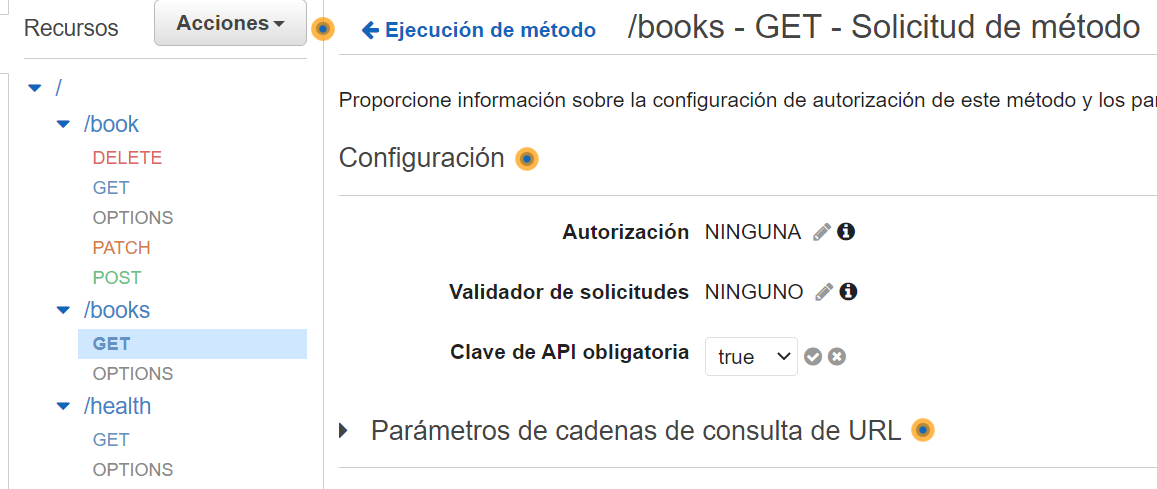
prod

Regresamos a la api a los recursos y seleccionamos uno a uno los metodos y damos clic en solicitud de metodo

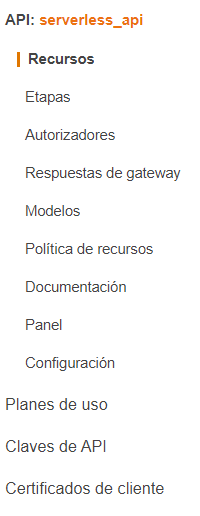


En clave de Api obligatoria damos seleccionamos TRUE y guardamos esto en cada uno de los metodos

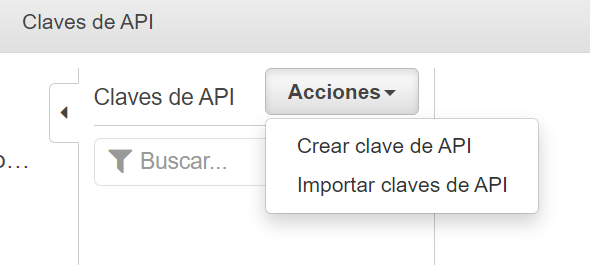




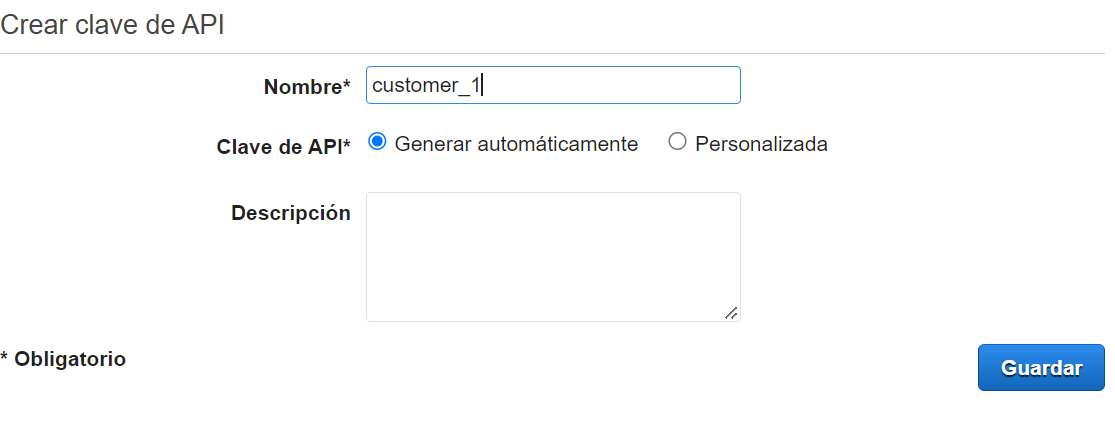
En el menú de la izquierda en api Gateway seleccionamos claves de API



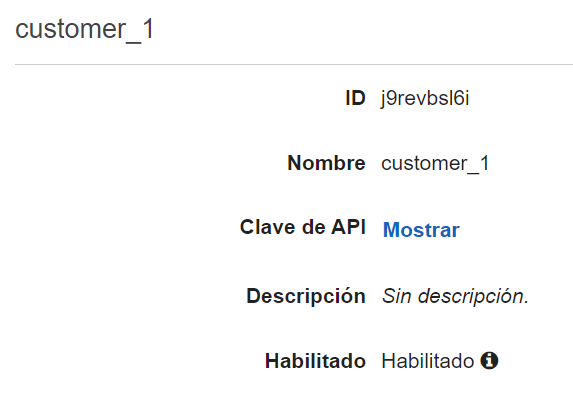
Damos clic en acciones y creamos una nueva clave de API



Le damos un nombre a la clave que vamos a manejar y que se genere automáticamente y clic en guardar



Así se ve la llave ya creada y en la clave de nos muestra la clave



Seleccionamos plan de uso



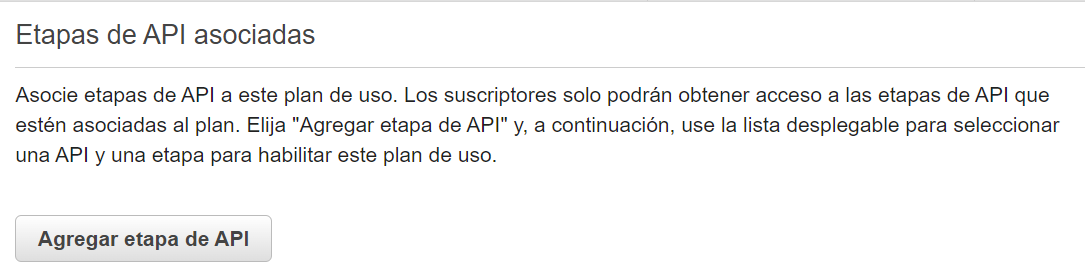
Le damos clic en crear y selecciamos un nombre para el plan de uso



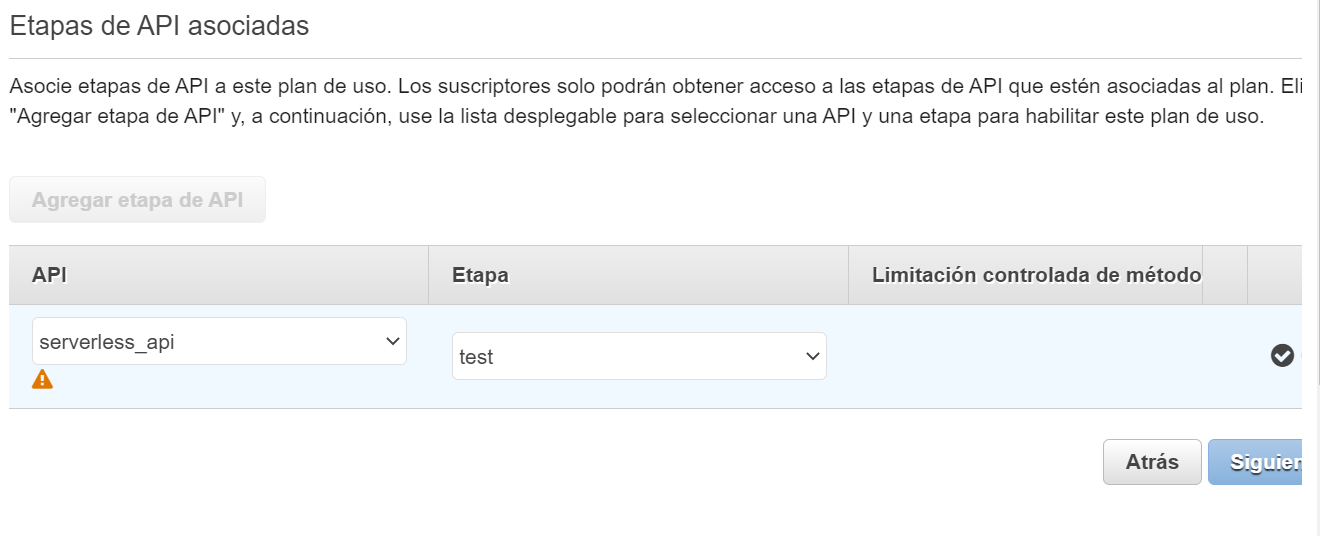
Indicamos que velocidad va a tener y la ráfaga también la cuota y damos clic en siguiente



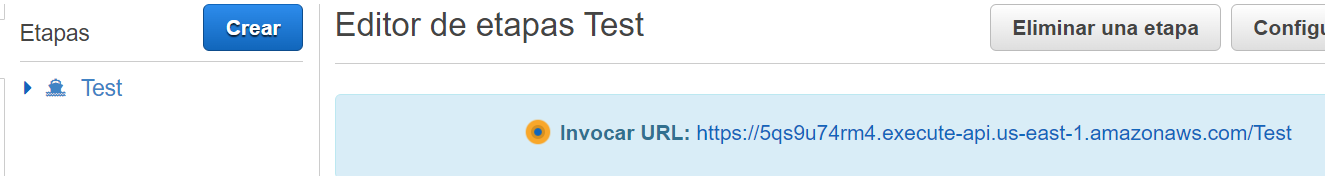
Agregamos etapa de la API



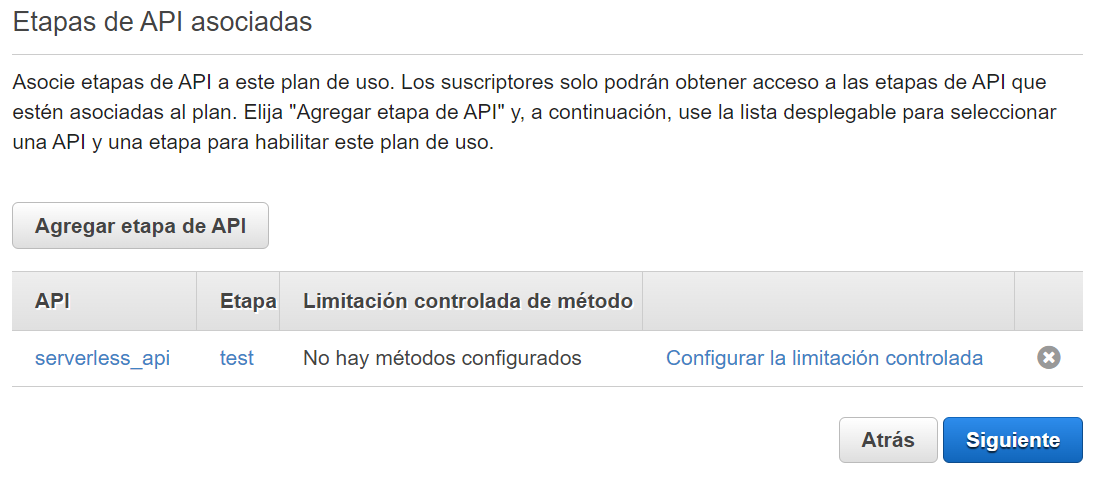
Buscamos el nombre de nuestra api ya creada y la etapa que previamente habíamos ingresado



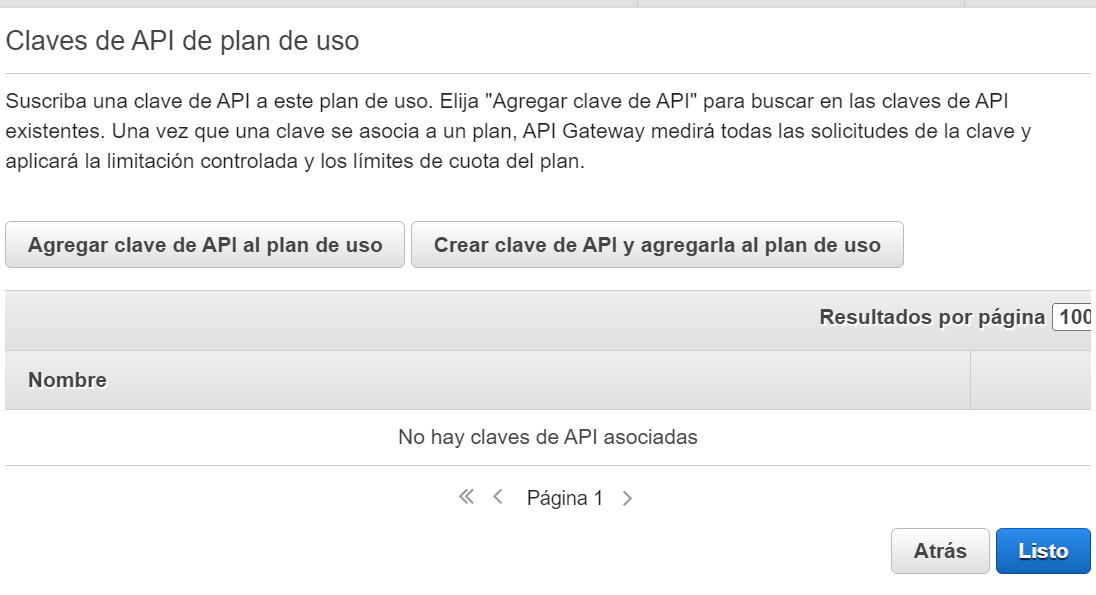
Ahora nos muestra cual es nuestra url para invocar el servicio



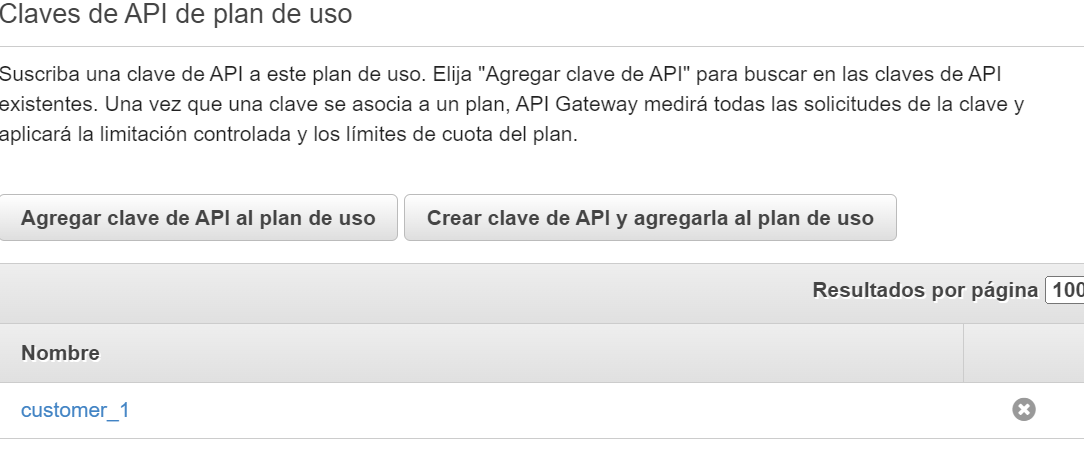
Damos clic en siguiente



Y clic en listo



Queda la clave de api creada

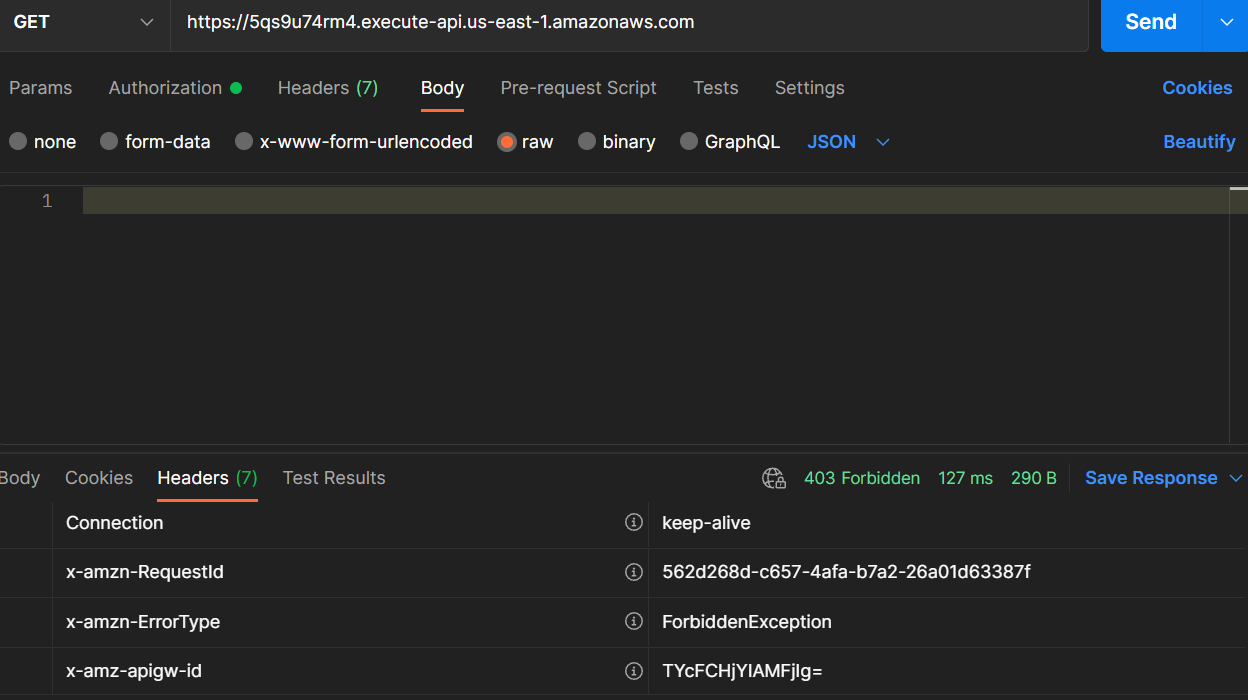


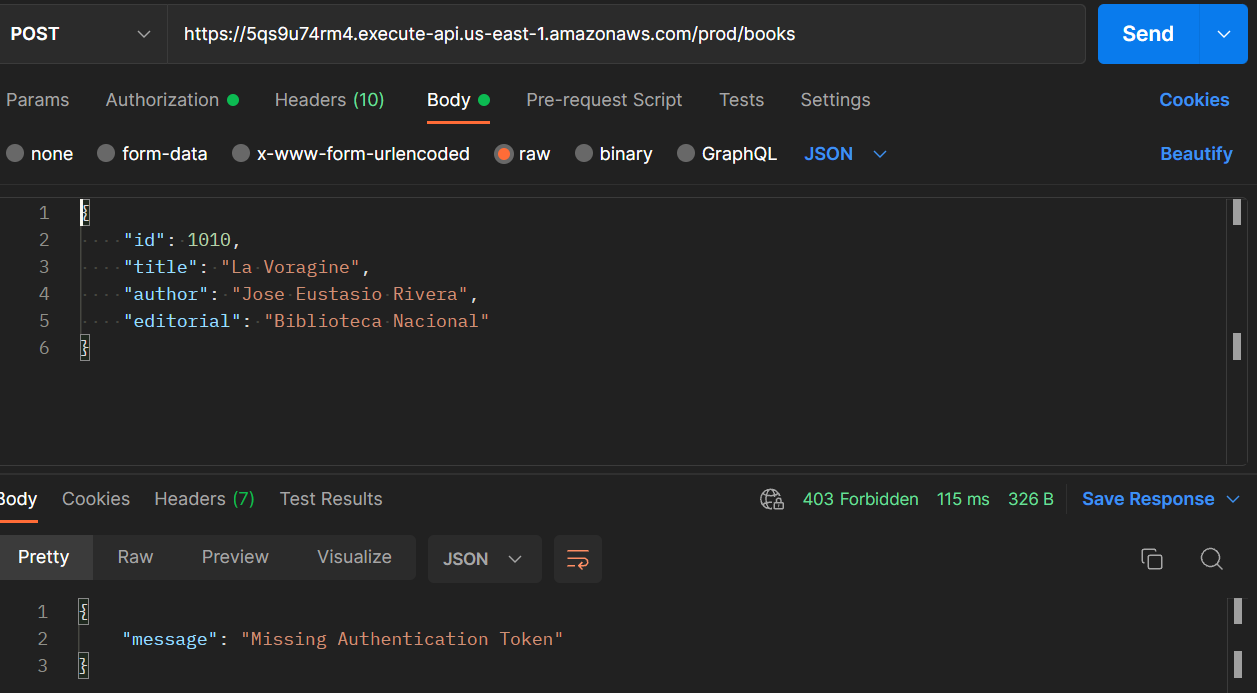
Ahora con la url y la llave ya creada vamos a postman para probar

URL de invocación

<https://5qs9u74rm4.execute-api.us-east-1.amazonaws.com/prod>

<https://5qs9u74rm4.execute-api.us-east-1.amazonaws.com/test>





**Código Python de function lambda en repositorio github**

<https://github.com/dexteram/crud_dynamo_api_python.git>

**eventos para probar en la lambda serverless\_api\_function**

VALIDOS

{

"HttpMethod": "DELETE",

"id": "101"

}

{

"HttpMethod": "GET",

"id": "102",

"title": "",

"author": "",

"editorial": ""

}

{

"HttpMethod": "GET\_ALL",

"id": "",

"title": "",

"author": "",

"editorial": ""

}

{

"HttpMethod": "POST",

"id": "102",

"title": "Book of the life",

"author": "Dita Dago",

"editorial": "Marketing Blocks"

}

{

"HttpMethod": "PUT",

"id": "888",

"title": "Viaje al centro de la tierra",

"author": "Julio Verne",

"editorial": "Basr-slow-PR"

}

EVENTO INVALIDO

{

"id": 1010,

"title": "La Voragine",

"author": "Jose Eustasio Rivera",

"editorial": "Biblioteca Nacional"

}