

31-10-2023

# Base de Datos NoSQL y Api Gateway

Computo en la nube



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



Yadira Estrada Cervantes  
TECNM ITZ

## Índice

Introducción .....	2
Desarrollo .....	2
Crear Api Rest .....	2
Prueba a petición post.....	4
Implementar API .....	7
Prueba a petición get .....	9
Eliminar Api Rest .....	10
Conclusiones .....	11
Referencias .....	11

## Introducción

A continuación, se mostrará el proceso que se siguió para crear un Api Rest de ejemplo y simular una base de datos a utilizando la tecnología Api Gateway de AWS.

## Desarrollo

### Crear Api Rest

Para crear un Api Rest no iremos a la consola de AWS, específicamente en el servicio de Api Gateway en el link de a continuación: [Amazon API Gateway](#) donde daremos clic en

**Comience a utilizar Amazon API Gateway**

Posteriormente, nos iremos a la opción de **API REST** y daremos clic en **crear**

### API REST

Desarrolle una API REST en la que obtenga control total de la solicitud y la respuesta, junto con las capacidades de administración de la API.

Funciona con lo siguiente:  
Lambda, HTTP, servicios de AWS

Seleccionaremos **el api de ejemplo** para practicar. Podemos ver el **contenido** de la api.

Posteriormente daremos clic en **Crear API**.

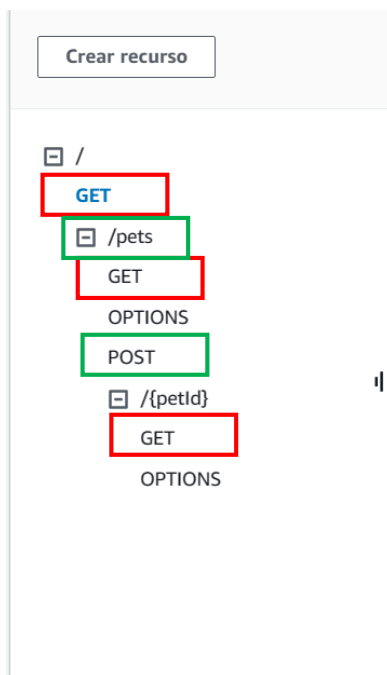
#### Detalles de la API

<input type="radio"/> Nueva API Cree una API de REST nueva.	<input type="radio"/> Clonar API existente Cree una copia de una API en esta cuenta de AWS.
<input type="radio"/> Importar API Importe una API desde una definición de OpenAPI.	<input checked="" type="radio"/> API de ejemplo Obtenga más información sobre API Gateway con un ejemplo de API.

```
1 {
2   "swagger": "2.0",
3   "info": {
4     "description": "Your first API with Amazon API Gateway. This is a sample
API that integrates via HTTP with our demo Pet Store endpoints",
5     "title": "PetStore"
6   },
7   "schemes": [
8     "https"
9   ],
10  "paths": {
11    "/": {
12      "get": {
13        "tags": [
14          "pets"
15        ],
16        "description": "PetStore HTML web page containing API usage informati
on",
```

Podremos ver los diferentes métodos que contiene la aplicación, los cuales son tres **get** (tipo request para obtención de información) y un **post** (tipo de request para enviar información).

Para poder ver la información daremos clic en el método, en este caso el **get que trae toda la información**.



Daremos clic en el **nombre** del método

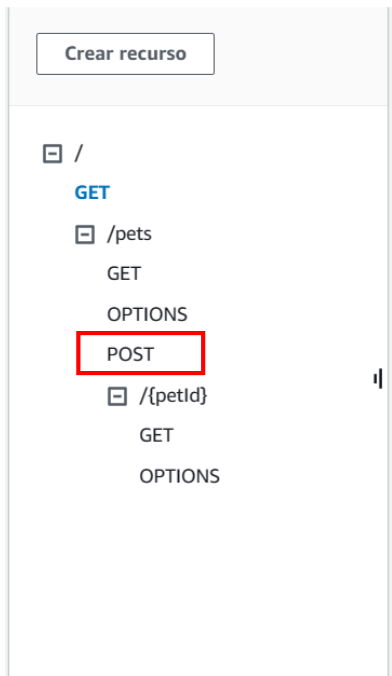
Métodos (1)				Eliminar	Crear método
Tipo de método	Tipo de integración	Autorización	Clave de API		
<input type="radio"/> GET	Simulación	Ninguna	No obligatorio		

Esto nos abrirá la información del método, donde podremos ver el proceso del request y el response.



## Prueba a petición post

Iremos al **post** de pets para ingresar un nuevo pet (cabe mencionar que son objetos moqueados, es decir, simulados en memoria)



En la parte inferior iremos a la sección de [pruebas](#)



En el body pondremos un objeto json con la información con el tipo perro y el precio de 249.99

#### Cuerpo de la solicitud

```
1 [{"type": "dog", "price": 249.99}]
```

Podremos ver el estado 200 (exito), con el body que introducimos, el encabezado, y el registro

 **/pets - Resultados de la prueba del método POST**

Solicitud	Latencia	Estado
/pets	32	200

Cuerpo de respuesta

```
{
  "pet": {
    "type": "dog",
    "price": 249.99
  },
  "message": "success"
}
```

Encabezados de respuesta

```
{
  "Access-Control-Allow-Origin": "*",
  "Content-Type": "application/json",
  "X-Amzn-Trace-Id": "Root=1-65411a9d-2b35fedfed3611f8d1270cf6"
}
```

Registro

Execution log for request 594d8b98-e0e0-450a-b8e4-948daaa063ad

Tue Oct 31 15:17:49 UTC 2023 : Starting execution for request: 594d8b98-e0e0-450a-b8e4-948daaa063ad

Tue Oct 31 15:17:49 UTC 2023 : HTTP Method: POST, Resource Path: /pets

Tue Oct 31 15:17:49 UTC 2023 : Method request path: {}

Tue Oct 31 15:17:49 UTC 2023 : Method request query string: {}

Tue Oct 31 15:17:49 UTC 2023 : Method request headers: {}

Tue Oct 31 15:17:49 UTC 2023 : Method request body before transformations: {"type": "dog", "price": 249.99}

Tue Oct 31 15:17:49 UTC 2023 : Endpoint request URI: http://petstore.execute-api.us-west-1.amazonaws.com/petstore/pets

Tue Oct 31 15:17:49 UTC 2023 : Endpoint request headers: {x-amzn-apigateway-api-id=tm9t848mxf, Accept=application/json, User-Agent=AmazonAPIGateway\_tm9t848mxf, X-Amzn-Trace-Id=Root=1-65411a9d-2b35fedfed3611f8d1270cf6, Content-Type=application/json}

Tue Oct 31 15:17:49 UTC 2023 : Endpoint request body after transformations: {"type": "dog", "price": 249.99}

Tue Oct 31 15:17:49 UTC 2023 : Sending request to http://petstore.execute-api.us-west-1.amazonaws.com/petstore/pets

Tue Oct 31 15:17:49 UTC 2023 : Received response. Status: 200, Integration latency: 5 ms

Tue Oct 31 15:17:49 UTC 2023 : Endpoint response headers: {Date=Tue, 31 Oct 2023 15:17:49 GMT, Content-Type=application/json; charset=utf-8, Content-Length=81, Connection=keep-alive, X-Powered-By=Express}

6

```
Tue Oct 31 15:17:49 UTC 2023 : Endpoint response body before transformations: {
  "pet": {
    "type": "dog",
    "price": 249.99
  },
  "message": "success"
}
Tue Oct 31 15:17:49 UTC 2023 : Method response body after transformations: {
  "pet": {
    "type": "dog",
    "price": 249.99
  },
  "message": "success"
}
Tue Oct 31 15:17:49 UTC 2023 : Method response headers: {X-Amzn-Trace-Id=Root=1-65411a9d-2b35fedfed3611f8d1270cf6, Access-Control-Allow-Origin=*, Content-Type=application/json}
Tue Oct 31 15:17:49 UTC 2023 : Successfully completed execution
Tue Oct 31 15:17:49 UTC 2023 : Method completed with status: 200
```

## Implementar API

Iremos a la parte superior y daremos clic en

**Implementar API**

Esto nos abrirá un formulario, donde escogeremos la etapa de **new stage**, además de ponerle **nombre** y una **descripción**, daremos clic en **Implementación** para continuar.

### Deploy API ✕

Elija una etapa donde se implementará su API. Por ejemplo, una versión de prueba de la API se podría implementar en una etapa denominada beta.

Etapa

\*New stage\*

Nombre de etapa

test

*Se creará una nueva etapa con los ajustes predeterminados. Edite la configuración de la etapa en la página **Etapa**.*

Descripción de la implementación

test stage

Cancelar **Implementación**

Nos lanzará un mensaje de que la implementación se encuentra activa.

Se creó correctamente la implementación de PetStore. Esta implementación está activa para test.

Podemos ver los detalles de la etapa, donde abriremos el URL

### Detalles de la etapa Información Editar

Nombre de etapa test	Tasa <span>Información</span> -	ACL web -
Caché de API ⊖ Inactivo	Ráfaga <span>Información</span> -	Certificado de cliente -

URL de invocación  
🔗 <https://tm9t848mxf.execute-api.us-west-1.amazonaws.com/test>

Implementación activa  
r0nnou el October 31, 2023, 09:23 (UTC-06:00)

Nos mostrara un mensaje de bienvenida donde nos informa que el get general (raiz) regresa un objeto moc (el que se está observando).

## Welcome to your Pet Store API

You have successfully deployed your first API. You are seeing this HTML page because the GET method to the root resource of your API returns this content as a Mock integration.

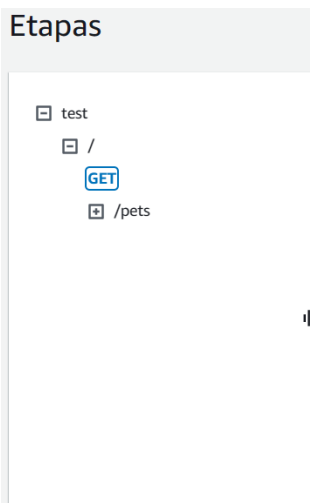
The Pet Store API contains the /pets and /pets/{petId} resources. By making a [GET request](#) to /pets you can retrieve a list of Pets in your API. If you are looking for a specific pet, for example the pet with ID 1, you can make a [GET request](#) to /pets/1.

You can use a REST client such as [Postman](#) to test the POST methods in your API to create a new pet. Use the sample body below to send the POST request:

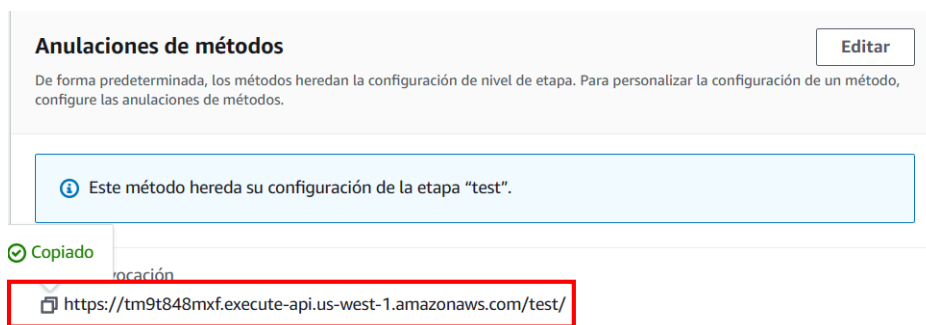
```
{
  "type" : "cat",
  "price" : 123.11
}
```

## Prueba a petición get

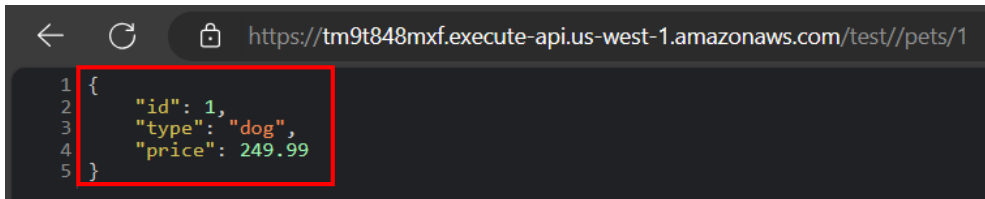
Para la petición get de un objeto en específico iremos a las etapas, donde seleccionaremos el [get](#) en test



Posteriormente, copiaremos el [link](#) que se nos da, añadiendo el id del objeto, quedadndo link- ejemplo/pets/{id}



Al entrar a esa url podremos observar que nos regresa un objeto **json** (el introducido anteriormente) como un objeto mokeado.



```
1 {  
2   "id": 1,  
3   "type": "dog",  
4   "price": 249.99  
5 }
```

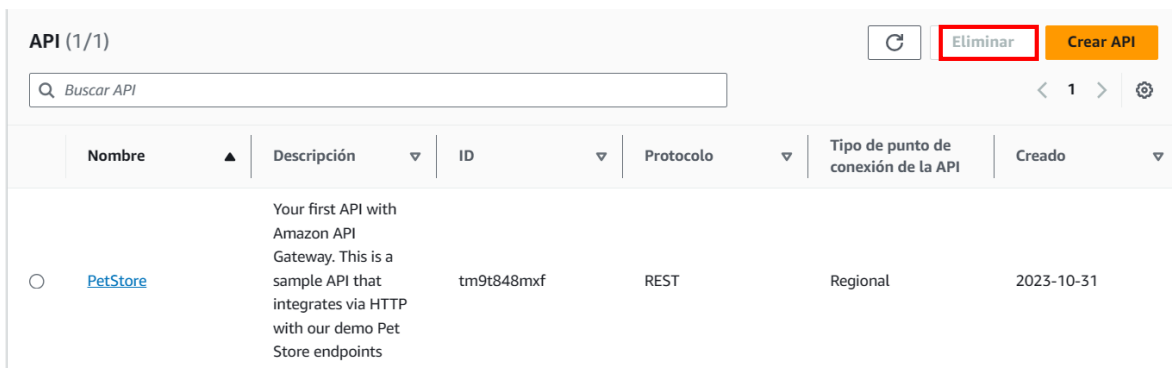
## Eliminar Api Rest

Iremos a la sección de **API**

[Gateway de API](#) > [API](#) > [PetStore \(tm9t848mxf\)](#) > [Etapas](#)

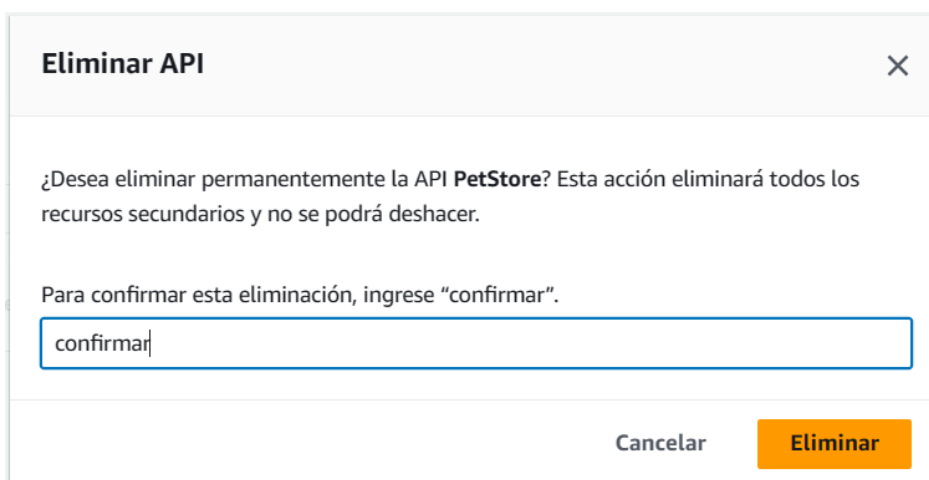
Seleccionamos la **API** que deseamos eliminar y, posteriormente daremos clic en

Eliminar



Nombre	Descripción	ID	Protocolo	Tipo de punto de conexión de la API	Creado
<a href="#">PetStore</a>	Your first API with Amazon API Gateway. This is a sample API that integrates via HTTP with our demo Pet Store endpoints	tm9t848mxf	REST	Regional	2023-10-31

Nos pedirá que escribamos “**confirmar**”, posteriormente daremos clic en “**eliminar**”



**Eliminar API**

¿Desea eliminar permanentemente la API **PetStore**? Esta acción eliminará todos los recursos secundarios y no se podrá deshacer.

Para confirmar esta eliminación, ingrese “confirmar”.

## Conclusiones

Es muy importante saber cómo subir un API REST en la nube debido a que es muy utilizado en la actualidad, pues permite independencia (es decir, que si una parte se cae la otra no sufrirá daños), además de que podemos observar que es sencilla de subir.

## Referencias

(S/f-a). Amazon.com. Recuperado el 31 de octubre de 2023, de <https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-create-api-from-example.html>

(S/f-b). Amazon.com. Recuperado el 31 de octubre de 2023, de <https://aws.amazon.com/es/api-gateway/>