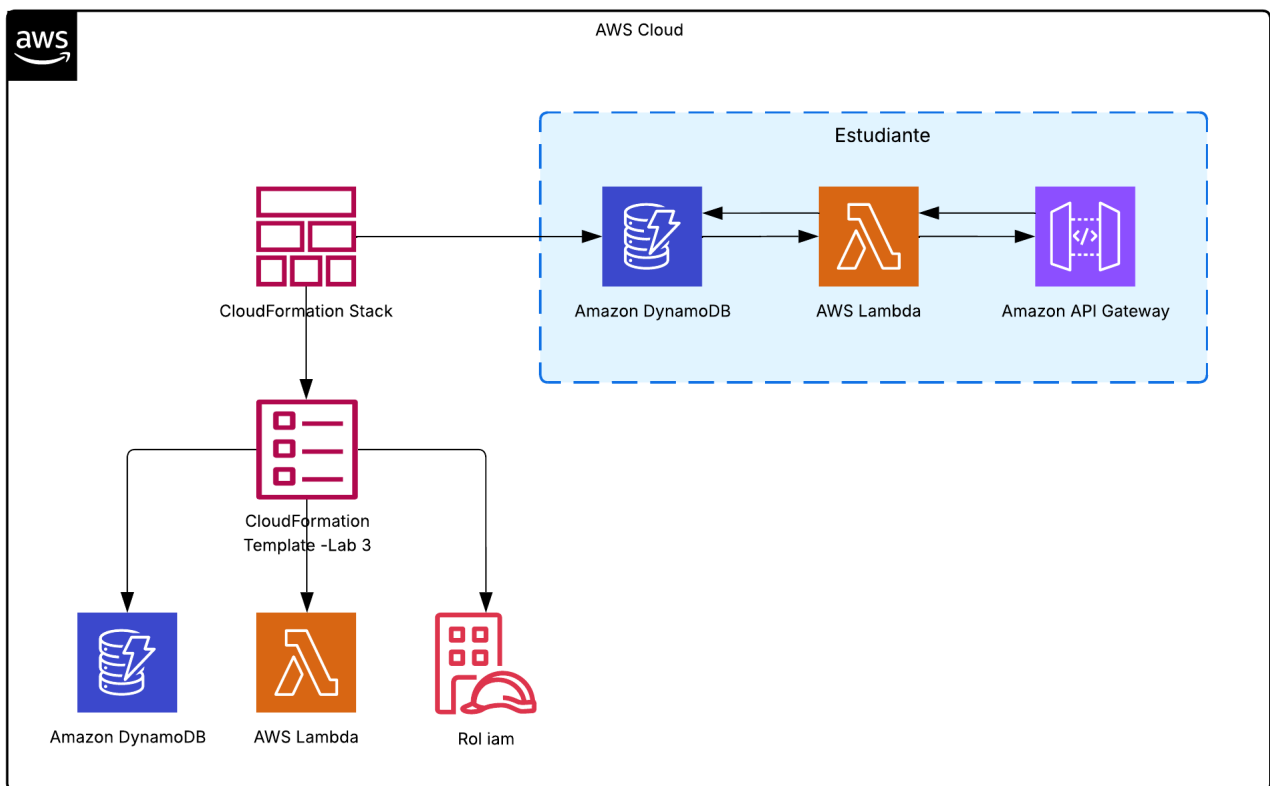


Laboratorio Serverless (DynamoDB - Lambda - API Gateway) AWS

Mi objetivo en este laboratorio fue **aprender a integrar servicios de AWS** como DynamoDB, Lambda y API Gateway para crear una arquitectura *serverless* que simulara una API de usuarios. El proceso incluyó conectar una función Lambda a DynamoDB, exponerla a través de API Gateway y validar su funcionamiento usando los métodos **GET** y **POST** desde el navegador y CloudShell.



Tarea 1: Revisé los recursos de laboratorio

Para empezar, revisé los recursos que ya estaban disponibles. En **DynamoDB**, encontré una tabla llamada **"Usuarios"** con 20 usuarios ya cargados. Usé la opción "Explorar Elementos" para ver los datos y confirmar que la tabla estuviera lista. También verifiqué el rol de **IAM** llamado **"LambdaExecutionRole"**, que era crucial porque le daría a mi función Lambda los permisos necesarios para interactuar con la tabla de DynamoDB, permitiéndole leer y escribir datos. Además, vi que ya existía una función Lambda llamada **"ConfigUsuarios"** que se encargaba de cargar los datos iniciales de la tabla.

Tarea 2: Creé una nueva función Lambda

El siguiente paso fue crear una nueva función en **Lambda**. La nombré, seleccioné **Python** como lenguaje de programación y mantuve la arquitectura predeterminada. En la sección de "Permisos", elegí "Cambiar el rol de ejecución predeterminado" y luego seleccioné la opción "Uso de un rol existente". Aquí, le asigné el rol "**LambdaExecutionRole**" que había revisado antes.

Crear una función [Información](#)

Seleccione una de las siguientes opciones para crear la función.

- ☒ **Crear desde cero**
Empiece con un sencillo ejemplo "Hello World".

- ☐ **Utilizar un proyecto**
Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predeterminados de casos de uso comunes.

Información básica

Nombre de la función

Escriba un nombre para describir el propósito de la función.

Verificar_Usuario

El nombre de la función debe tener entre 1 y 64 caracteres, debe ser exclusivo de la región y no puede incluir espacios. Los caracteres válidos son a-z, A-Z, 0-9, guiones (-) y guiones bajos (_).

Tiempo de ejecución [Información](#)

Elija el idioma para usar para escribir su función. Tenga en cuenta que el editor de código de la consola es compatible con solo Node.js, Python y Ruby.

Python 3.13

Arquitectura [Información](#)

Elija la arquitectura del conjunto de instrucciones que desea para el código de la función.

- ☐ arm64
☒ x86_64

Permisos [Información](#)

De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado.

▼ Cambiar el rol de ejecución predeterminado

Rol de ejecución

Seleccione un rol que defina los permisos de la función. Para crear un rol personalizado, vaya a la [consola de IAM](#).

- ☐ Creación de un nuevo rol con permisos básicos de Lambda
☒ Uso de un rol existente
☐ Creación de un nuevo rol desde la política de AWS templates

Rol existente

Seleccione un rol existente que haya creado para usarlo con esta función de Lambda. El rol debe tener permiso para cargar registros en Amazon CloudWatch Logs.

StackSet-LabServerlessV2Sept-df-LambdaExecutionRole-neKGO6OexY9a

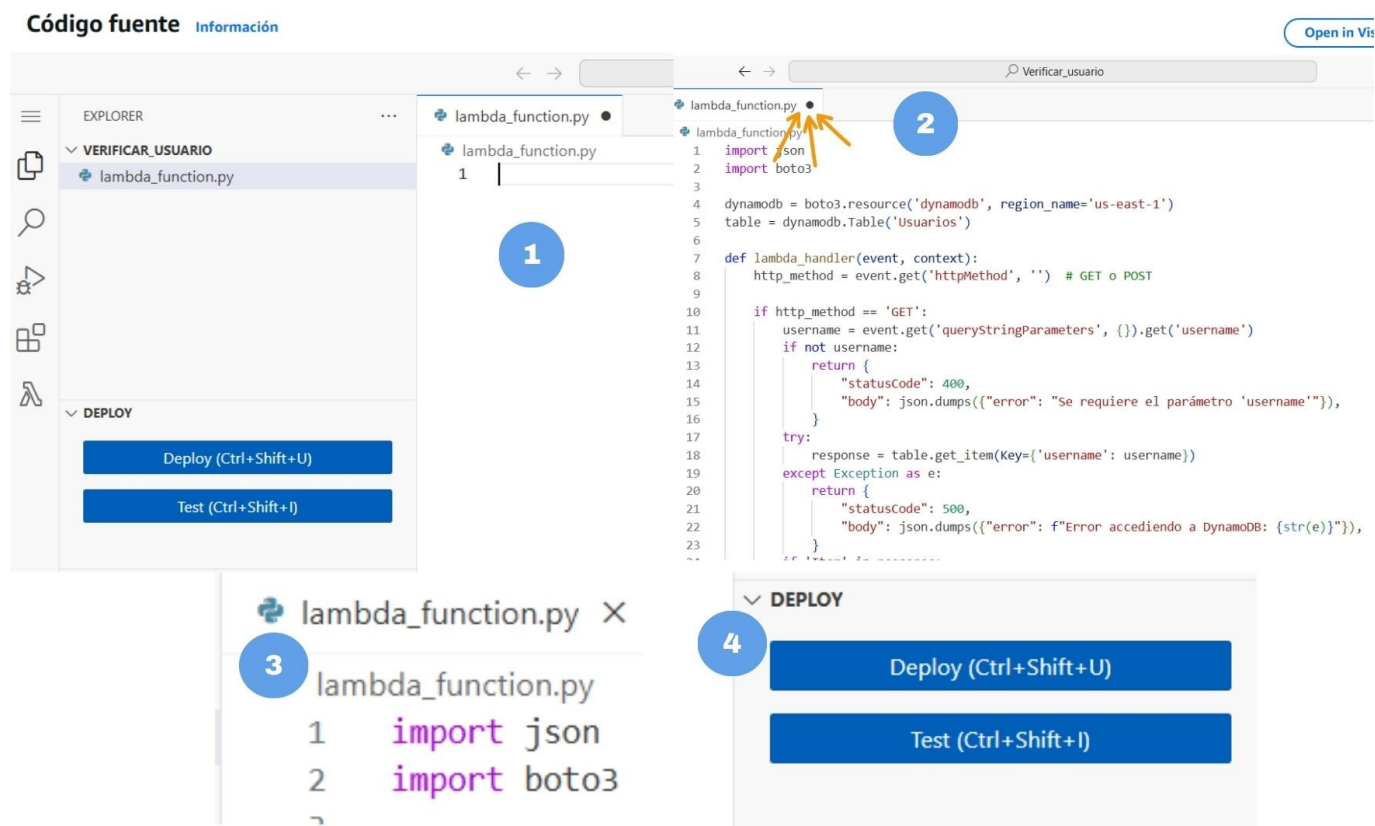
[Consulte el rol StackSet-LabServerlessV2Sept-df-LambdaExecutionRole-neKGO6OexY9a](#) en la consola de IAM.

Tarea 3: Escribí el código de la función Lambda

Una vez que la función Lambda estuvo configurada, me dediqué a escribir el código. El archivo `lambda_function.py` ya tenía un código preconfigurado, así que lo borré por completo. Luego, pegué el **nuevo script** que me permitía manejar tanto las operaciones **POST** como **GET**. Este

código era fundamental, ya que se encargaba de conectar con DynamoDB para insertar y consultar datos.

Me aseguré de que el archivo se guardara correctamente (el círculo junto al nombre del archivo desaparece) y, finalmente, hice clic en **"Deploy"** para aplicar los cambios y actualizar la función.



Tarea 4: Probé la función Lambda

Para validar mi código, utilicé la funcionalidad de prueba de Lambda. Creé un nuevo evento de prueba y lo nombré.

Primero, probé la solicitud **GET**. En el "Evento Json", inserté un JSON que pedía el `username` "Hugo". Hice clic en **"Test"** y el resultado fue `true`, lo que significaba que el usuario "Hugo" sí tenía una contraseña registrada. Para confirmar, cambié el `username` a "Nacho" y volví a probar. Esta vez, el resultado fue `false`, lo que demostraba que mi función estaba funcionando correctamente para las consultas.

Probar evento [información](#)

CloudWatch Logs Live Tail Guardar Probar

Para Invocar la función sin guardar un evento, configure el evento JSON, y luego elija Probar.

Acción de evento de prueba

Crear un nuevo evento

Editar evento guardado

Nombre del evento

test1

Máximo de 25 caracteres compuestos por letras, números, puntos, guiones y guiones bajos.

Configuración de uso compartido de eventos

Privado

Este evento solo está disponible en la consola de Lambda y para el creador del evento. Puede configurar un total de 10. [Más información](#)

Compartible

Este evento está disponible para los usuarios de IAM de la misma cuenta que tienen permisos para obtener acceso y utilizar eventos compartibles. [Más información](#)

Plantilla - Opcional

Hello World

Evento JSON

Formato JSON

```
1 {
2   "httpMethod": "GET",
3   "queryStringParameters": {
4     "username": "Hugo"
5   }
6 }
```

4:26 JSON

Luego, pasé a probar el método **POST**. En el "Evento Json", reemplacé la solicitud por una que incluía el método **POST** y un objeto `body` con los datos de un nuevo usuario. Hice clic en **"Test"** nuevamente. Para verificar si el nuevo usuario se había agregado, regresé a la tabla "Usuarios" en DynamoDB. Antes había 20 elementos, y ahora había **21**, confirmando que el nuevo usuario, "Anto", se había insertado exitosamente.

Ejecutando la función: sin errores [registros](#)

▼ Detalles

```
{
  "statusCode": 200,
  "body": "{\"username\": \"Hugo\", \"tiene_password\": true}"
}
```

Resumen

Código SHA-256
HAPq9ReJVEC5gLavtc/gyd5vZtd9eiUGF932t0jBxY=

Tiempo de ejecución
hace 41 segundos

Versión de la función
\$LATEST

ID de solicitud
038195fe-9ef0-4db7-9af6-dd426424b018

Duración
321.30 ms

Duración facturada
773 ms

Recursos configurados
128 MB

Memoria máx. utilizada
88 MB

Duración de inicialización
451.48 ms

Resultado de registro

En el área siguiente se muestran los últimos 4 KB del registro de ejecución. [Haga clic aquí](#) para ver el grupo de registro de CloudWatch correspondiente.

```
START RequestId: 038195fe-9ef0-4db7-9af6-dd426424b018 Version: $LATEST
END RequestId: 038195fe-9ef0-4db7-9af6-dd426424b018
REPORT RequestId: 038195fe-9ef0-4db7-9af6-dd426424b018  Duration: 321.30 ms    Billed Duration: 773 ms Memory Size: 128 MB    Max Memory Used: 88 MB    Init Duration: 451.48 ms
```

Tarea 5: Creé una REST API

Después de asegurarme de que mi función Lambda funcionaba, el siguiente paso fue exponerla a través de una API. Fui a **API Gateway** y creé una nueva **REST API**, dándole un nombre simple y dejando las demás opciones por defecto.

☰ [Gateway de API](#) > [API](#) > [Crear API](#) > Crear API de REST

Detalles de la API



Nueva API

Cree una API de REST nueva.



Clonar API existente

Cree una copia de una API en esta cuenta



Importar API

Importe una API desde una definición de OpenAPI.



API de ejemplo

Obtenga más información sobre API Gateway

Nombre de API

Usuarios_API

Descripción: *opcional*

Tipo de punto de conexión de la API

Las API regionales se implementan en la región de AWS actual. Las API optimizadas para la periferia dirigen las solicitudes al punto de presencia de CloudFront más cercano. Solo

Regional

Tipo de dirección IP [Información](#)

Seleccione el tipo de direcciones IP que pueden invocar el punto de enlace predeterminado de la API.



IPv4

Solo admite tipos de puntos de conexión de API regionales y optimizados para periferia.



Doble pila

Admite todos los tipos de puntos de conexión de la API.

Ya con la API creada, creé un **recurso**. Esto era clave, ya que definía la ruta que API Gateway escucharía para las peticiones. Una vez que el recurso estaba listo, creé los **métodos GET** y **POST** para ese recurso. Fue muy importante que seleccionara la integración con la función **Lambda** que había creado en la Tarea 2 y que activara la opción "**Lambda proxy integration**".

Crear recurso

Detalles del recurso



Recurso de proxy [Información](#)

Los recursos de proxy gestionan las solicitudes a todos los subrecursos. Para crear un recurso de proxy, utilice un parámetro de ruta que termine con un signo más, por ejemplo {proxy+}.

Ruta de recurso

/

Nombre del recurso

usuarios



CORS (uso compartido de recursos entre orígenes) [Información](#)

Cree un método OPTIONS que permita todos los orígenes, todos los métodos y varios encabezados comunes.

[Cancelar](#)

[Crear recurso](#)

Detalles del método

Tipo de método

GET

Tipo de integración

☒ **Función de Lambda**
Integre su API con una función de Lambda.



☐ **HTTP**
Lleve a cabo la integración con un punto de conexión HTTP existente.



☐ **Simulación**
Genere una respuesta basada en las asignaciones y transformaciones de API Gateway.



☐ **Servicio de AWS**
Lleve a cabo la integración con un servicio de AWS.



☐ **Enlace de VPC**
Lleve a cabo la integración con un recurso al que no se pueda acceder a través de la red pública de Internet.



☒ **Integración de proxy de Lambda**
Envíe la solicitud a la función de Lambda como un evento estructurado.

Función de Lambda

Proporcione el nombre de la función de Lambda o un alias. También puede proporcionar un ARN de otra cuenta.

us-east-1

arn:aws:lambda:us-east-1:521557890247:function:Verificar_Usuario

Otorgue permiso a API Gateway para invocar la función de Lambda
Al guardar los cambios, API Gateway actualiza la política basada en recursos de la función de Lambda para permitir que esta API la invoque.

Tiempo de espera de integración [Información](#)

Por defecto, puede introducir un tiempo de espera de integración de 50 a 29 000 milisegundos. Puede usar las Service Quotas para aumentar el tiempo de espera de integración a más de 29 000 ms

29000

Recursos

Crear recurso

/

/usuarios

GET

POST

Finalmente, para que la API estuviera accesible públicamente, hice un **"Deploy API"**. Me pidió nombrar una "etapa" (stage), que es como una versión de la API. Al hacer el *deploy*, API Gateway me generó una **URL de invocación** que usaría para las pruebas.

Deploy API

×

Cree o seleccione una etapa en la que se implementará la API. Puede utilizar el historial de implementaciones para revertir o cambiar la implementación activa de una etapa. [Learn more](#)

Etapas

Nueva etapa

Nombre de etapa

usuariostest

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

Descripción de la implementación

Cancelar

Implementación

Tarea 6: Probé la API en el navegador y CloudShell

Con la URL de invocación en mano, procedí a las pruebas finales.

Para el método **GET**, abrí mi navegador web. Construí una URL con el siguiente formato: `URL_de_invocación/nombre_del_recurso?username=nombre_de_usuario`. Al presionar Enter, el navegador mostró `true` o `false` directamente, confirmando si el usuario tenía una contraseña, tal como lo había hecho en la prueba de Lambda.

Para el método **POST**, utilicé **CloudShell**. Abrí la terminal y pegué el comando `curl`. Reemplacé la `InvokeURL` con la mía y modifiqué el JSON para enviar los datos de un nuevo usuario, "Roberto". Después de ejecutar el comando, volví a DynamoDB y confirmé que el nuevo usuario se había agregado a la tabla.

personalizados

Asociaciones de acceso a nombres de dominio

Enlaces de VPC

▼ API: Usuario_API

Recursos

Etapas

Autorizadores

Respuestas de puerta de enlace

Modelos

Política de recursos

Documentación

Panel

Configuración de la API

Planes de uso

Claves de API

Certificados del cliente

Configuración de

usuariotest

Detalles de la etapa

Información

Nombre de etapa

usuariotest

Clúster de caché

Información

Almacenamiento en caché de nivel de método predeterminado

Inactivo

Tasa

Información

10000

Ampliación

Información

5000

URL de invocación

<https://48n2q7a763.execute-api.us-east-1.amazonaws.com/usuariotest>

Implementación activa

ragi7T el September 04, 2025, 20:36 (UTC-06:00)

Registros y seguimiento

Información

Registros de CloudWatch

Inactivo

Seguimiento de X-Ray

Inactivo

Métricas detalladas

Inactivo

CloudShell

us-east-1

```
~ $ curl -X POST https://48n2q7a763.execute-api.us-east-1.amazonaws.com/usuariotest/Usuarios \
> -H "Content-Type: application/json" \
> -d '{"username":"Roberto", "país":"Chile", "color_favorito":"verde", "password":"robert34"}'
{"message": "Usuario agregado/actualizado correctamente"}~ $
```

CloudShell

us-east-1

```
~ $ curl -X POST https://48n2q7a763.execute-api.us-east-1.amazonaws.com/usuariotest/Usuarios \
> -H "Content-Type: application/json" \
> -d '{"username":"Roberto", "país":"Chile", "color_favorito":"verde", "password":"robert34"}'
{"message": "Usuario agregado/actualizado correctamente"}~ $
~ $ curl -X GET \
> "https://48n2q7a763.execute-api.us-east-1.amazonaws.com/usuariotest/Usuarios?username=Roberto"
{"username": "Roberto", "tiene_password": true}~ $
```

CloudShell

us-east-1

```
~ $ curl -X POST https://48n2q7a763.execute-api.us-east-1.amazonaws.com/usuariotest/Usuarios \
> -H "Content-Type: application/json" \
> -d '{"username":"Christhian", "país":"Mexico", "color_favorito":"verde", "password":"Noa123"}'
{"message": "Usuario agregado/actualizado correctamente"}~ $
```


Con esto, quedo funcionando correctamente.

The screenshot shows the AWS DynamoDB console interface. On the left, there's a navigation menu with options like Panel, Tablas, Explorar elementos (selected), Editor PartiQL, Copias de seguridad, Exportaciones a S3, Importaciones de S3, Integraciones, Capacidad reservada, and Configuración. Below this is a section for DAX with options like Clústeres, Grupos de subredes, Grupos de parámetros, and Eventos. The main area displays the 'Tabla: Usuarios: elementos devueltos (1/23)' with a status bar indicating 'Completado' and 'Elementos devueltos: 23'. Below the status bar, a table lists the items. The 'Christhlan' item is selected, highlighted in blue. The table has columns: username (Cadena), color_favorito, país, and password.

	username (Cadena)	color_favorito	país	password
<input type="checkbox"/>	Hugo			hugoPass
<input type="checkbox"/>	Carla			abcdef
<input type="checkbox"/>	Roberto	verde	Chile	robert34
<input type="checkbox"/>	chris			1234
<input type="checkbox"/>	Nacho			
<input type="checkbox"/>	Diego			diegoPass
<input type="checkbox"/>	Luis			
<input type="checkbox"/>	Marta			
<input type="checkbox"/>	Beto			secreto45
<input type="checkbox"/>	Sergio			
<input type="checkbox"/>	Karla			
<input type="checkbox"/>	Gabriela			pass123
<input type="checkbox"/>	Tamara			
<input checked="" type="checkbox"/>	Christhlan	verde	Mexico	Noa123
<input type="checkbox"/>	Elena			789xyz
<input type="checkbox"/>	Rosa			
<input type="checkbox"/>	Irene			zz44qq
<input type="checkbox"/>	...			

Desafío Final: Creé mi propia API de Libros

Como desafío adicional, creé una nueva API para manejar libros. Seguí los mismos pasos del laboratorio:

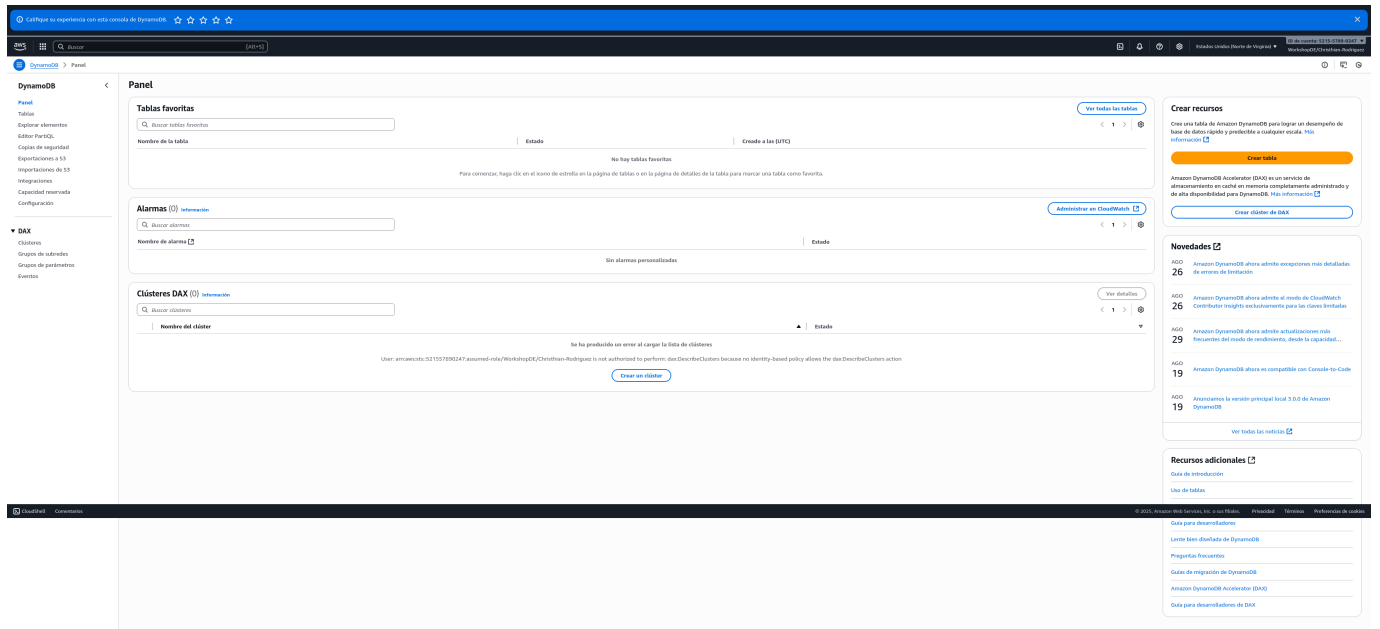
1. Creé una nueva tabla en **DynamoDB** llamada "Libros" con campos como `título` , `autor` y `año` .
2. Desarrollé una nueva función **Lambda** con un código adaptado para esta tabla. La función incluía:
 - Un método **POST** para agregar nuevos libros, retornando un mensaje de éxito como "Libro agregado con éxito".
 - Un método **GET** para consultar un libro específico por su título, que actuaba como mi campo clave.

3. Configuré una nueva **REST API** en API Gateway con los métodos **GET** y **POST** apuntando a mi nueva función Lambda.

4. Finalmente, probé la API en el navegador y en CloudShell para asegurarme de que todo funcionaba como esperaba.

¡Con esto, di por completado el laboratorio y el desafío! 💪

Creacion de la tabla DynamoDB



Configuraciones de la tabla

Crear tabla

Detalles de la tabla [Información](#)

DynamoDB es una base de datos sin esquemas que solo requiere un nombre de tabla y una clave principal al crear la tabla.

Nombre de la tabla

Se utilizará para identificar su tabla.

Libros

Entre 3 y 255 caracteres. Solo se pueden usar letras, números, guiones bajos (_) y puntos (.).

Clave de partición

La clave de partición forma parte de la clave principal de la tabla. Se trata de un valor hash que se utiliza para recuperar elementos de la tabla, así como para asignar datos entre hosts por cuestiones de escalabilidad y disponibilidad.

Id

Cadena

De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.

Clave de ordenación - *opcional*

Puede utilizar una clave de ordenación como segunda parte de la clave principal de una tabla. La clave de ordenación le permite ordenar o buscar entre todos los elementos que comparten la misma clave de partición.

Índices secundarios

Cadena

De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.

Configuración de la tabla

☒ Configuración predeterminada

La forma más rápida de crear su tabla. Puede modificar la mayor parte de la configuración después de crear la tabla. Para modificar esta configuración ahora, elija "Personalizar la configuración".

☐ Personalizar configuración

Utilice estas características avanzadas para que DynamoDB funcione mejor de acuerdo a sus necesidades.

Con los recursos credos, podemos seguir con la configuracion completa.

Tablas (3) Información

<input type="checkbox"/>	Nombre	Estado	Clave de partición	Clave de ordenación	Índices
<input type="checkbox"/>	Libros	✓ Activo	id (S)		índices secundarios (S)
<input type="checkbox"/>	ResourcesTable	✓ Activo	userId (S)		resourceId (S)
<input type="checkbox"/>	Usuarios	✓ Activo	username (S)		-

Creamos la funcion lambda y la condiguramos segun nuestros objetivos.

Crear una función Información

Seleccione una de las siguientes opciones para crear la función.

☒ Crear desde cero

Emplee con un sencillo ejemplo "Hello World".

☐ Utilizar un proyecto

Cree una aplicación Lambda utilizando un código de muestra y los ajustes de configuración predefinidos de casos de uso comunes.

☐ Imagen del contenedor

Seleccione una imagen de contenedor para implementar para la función.

Información básica

Nombre de la función

Escriba un nombre para describir el propósito de la función.

El nombre de la función debe tener entre 1 y 64 caracteres, debe ser exclusivo de la región y no puede incluir espacios. Los caracteres válidos son a-z, A-Z, 0-9, guiones (-) y guiones bajos (_).

Tiempo de ejecución Información

Elija el idioma para usar para escribir su función. Tenga en cuenta que el editor de código de la consola es compatible con solo Node.js, Python y Ruby.

Arquitectura Información

Elija la arquitectura del conjunto de instrucciones que desea para el código de la función.

☐ arm64☒ x86_64

Permisos Información

De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado más adelante al agregar los disparadores.

▼ Cambiar el rol de ejecución predeterminado

Rol de ejecución

Seleccione un rol que defina los permisos de la función. Para crear un rol personalizado, vaya a la [consola de IAM](#).

☐ Creación de un nuevo rol con permisos básicos de Lambda☒ Uso de un rol existente☐ Creación de un nuevo rol desde la política de AWS templates

Rol existente

Seleccione un rol existente que haya creado para usarlo con esta función de Lambda. El rol debe tener permiso para cargar registros en Amazon CloudWatch Logs.

[Consulte el rol StackSet-LabServerlessV2Sept-df-LambdaExecutionRole-neKGO6OexY9a](#) en la consola de IAM.

► Configuraciones adicionales

Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function deployment.

[Cancelar](#)[Crear una función](#)

Agregamos el script necesario para que funcione según nuestra necesidad.

Se ha actualizado correctamente la función Verificar_Libros.

Verificar_Libros

Limitación

Copiar ARN

Acciones

▼ Información general de la función

Información

Diagrama

Plantilla



+ Agregar desencadenador

+ Agregar destino

Descripción

Última modificación

hace 56 segundos

ARN de la función

arn:aws:lambda:us-east-1:521557890247:function:Verificar_Libros

Código

Probar

Monitorear

Configuración

Alias

Versiones

Código fuente

Información

Open in Visual Studio Code

Cargar desde

```
lambda_function.py X
lambda_function.py
1 import json
2 import boto3
3 import uuid
4
5 dynamodb = boto3.resource('dynamodb')
6 table = dynamodb.Table('Libros')
7
8 def lambda_handler(event, context):
9     method = event.get('httpMethod', '')
10
11     if method == 'POST':
12         body = json.loads(event['body'])
13         libro_id = str(uuid.uuid4())
14         item = {
15             'id': libro_id,
16             'titulo': body['titulo'],
17             'autor': body['autor'],
18             'anio': body['anio']
19         }
20         table.put_item(Item=item)
21         return f
```

Creamos el metodo para realizar las consultas.

Detalles del método

Tipo de método

GET

Tipo de integración

☒ Función de Lambda

Integre su API con una función de Lambda.



☐ HTTP

Lleve a cabo la integración con un punto de conexión HTTP existente.



☐ Simulación

Genere una respuesta basada en las asignaciones y transformaciones de API Gateway.



☐ Servicio de AWS

Lleve a cabo la integración con un servicio de AWS.



☐ Enlace de VPC

Lleve a cabo la integración con un recurso al que no se pueda acceder a través de la red pública de Internet.



☒ Integración de proxy de Lambda

Envíe la solicitud a la función de Lambda como un evento estructurado.

Función de Lambda

Proporcione el nombre de la función de Lambda o un alias. También puede proporcionar un ARN de otra cuenta.

us-east-1

arn:aws:lambda:us-east-1:521557890247:function:Verificar_Libros

☒ Otorgue permiso a API Gateway para invocar la función de Lambda

Al guardar los cambios, API Gateway actualiza la política basada en recursos de la función de Lambda para permitir que esta API la invoque.

Tiempo de espera de integración

Por defecto, puede introducir un tiempo de espera de integración de 50 a 29 000 milisegundos. Puede usar las Service Quotas para aumentar el tiempo de espera de integración a más de 29 000 ms

29000

Con esto queda terminado nuestro metodo para las consultas.

Recursos

Crear recurso

/

/Libros

GET

POST

Detalles del recurso

Ruta
/Libros

ID de recurso
7stqgy

Acciones de API

Implementar API

Eliminar

Actualizar documentación

Habilitar CORS

Métodos (2)

Eliminar

Crear método

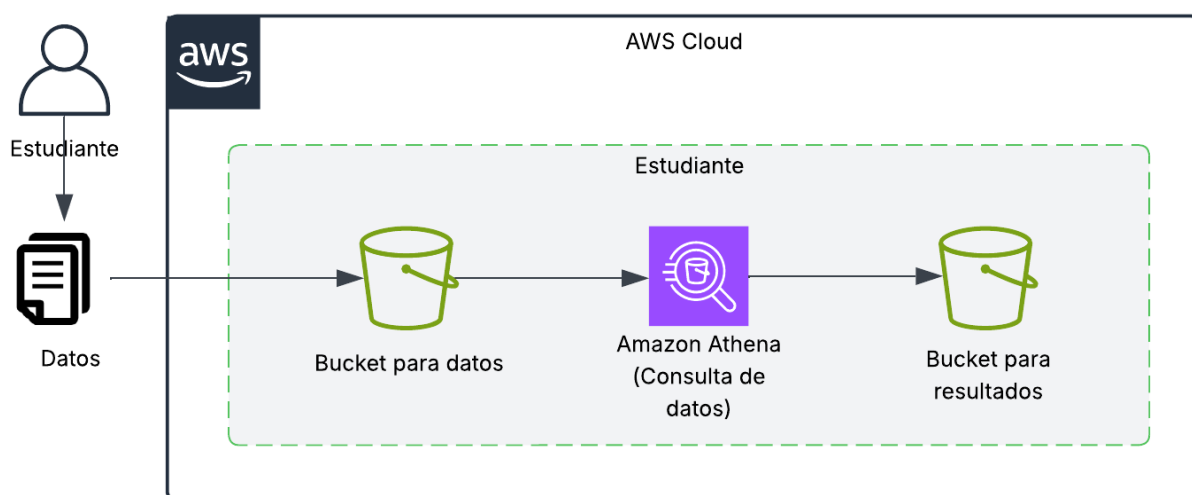
	Tipo de método	Tipo de integración	Autorización	Clave de API
<input type="radio"/>	GET	Lambda	Ninguna	No obligatorio
<input type="radio"/>	POST	Lambda	Ninguna	No obligatorio

No les voy a mentir, entre tarde, y no alcance a completar el resto, me cerraron el laboratorio y mientras avanzaba con el resto, dejo de cargar y ahí quedo...

Laboratorio Athena Quest

Resumen del Laboratorio

En este laboratorio, mi objetivo principal fue aprender a utilizar **Amazon S3** y **Amazon Athena** para el análisis de datos. La idea era crear un flujo completo: desde subir un archivo de datos a S3 hasta realizar consultas SQL para explorarlo. Entendí que S3 es un servicio de almacenamiento de objetos, mientras que Athena me permite consultar esos datos directamente desde S3 sin la necesidad de servidores. Un detalle clave que aprendí fue que Athena siempre guarda los resultados de sus consultas en un bucket de S3, lo que hacía necesario crear dos buckets diferentes para este laboratorio.



Tarea 1 y 2: Descargué y Organicé mis Datos

Primero, descargué el archivo `books.csv` que contenía información sobre libros. Después, me dirigí a **Amazon S3** para crear un bucket. Le di un nombre único y mantuve el resto de la configuración por defecto, asegurándome de que el acceso público estuviera bloqueado por seguridad. Luego, abrí el bucket que acababa de crear y subí el archivo `books.csv` que había descargado. Esto sería la fuente de mis datos.

Crear bucket [Información](#)

Los buckets son contenedores de datos almacenados en S3.

Configuración general

Región de AWS

EE.UU. Este (Norte de Virginia) us-east-1

Tipo de bucket [Información](#)

☒ Uso general

Recomendado para la mayoría de los casos de uso y patrones de acceso. Los buckets de uso general son del tipo de bucket de S3 original. Permiten una combinación de clases de almacenamiento que almacenan objetos de forma redundante en múltiples zonas de disponibilidad.

☐ Directorio

Recomendado para casos de uso de baja latencia. Estos buckets utilizan únicamente la clase de almacenamiento S3 Express One Zone, que proporciona un procesamiento más rápido de los datos dentro de una única zona de disponibilidad.

Nombre del bucket [Información](#)

DatosDelLibroLab4

Los nombres de los buckets deben tener entre 3 y 63 caracteres y ser únicos dentro del espacio de nombres global. Los nombres de los buckets también deben empezar y terminar con una letra o un número. Los caracteres válidos son a-z, 0-9, puntos (.) y guiones (-). [Más información](#)

Copiar la configuración del bucket existente: *opcional*

Solo se copia la configuración del bucket en los siguientes ajustes.

[Elegir el bucket](#)

Formato: s3://bucket/prefijo

Se ha realizado la carga correctamente

Para obtener más información, consulte la tabla Archivos y carpetas.

Cargar: estado

Cerrar

Después de salir de esta página, la siguiente información ya no estará disponible.

Resumen

Destino

s3://datosdelibroLab4

Realizado correctamente

1 archivo, 2.5 KB (100.00%)

Con errores

0 archivos, 0 B (0%)

Archivos y carpetas

Configuración

Archivos y carpetas (1 total, 2.5 KB)

Buscar por nombre

< 1 >

Nombre	Carpeta	Tipo	Tamaño	Estado	Error
libros_laboratorio.csv	-	text/csv	2.5 KB	Realizado correctamente	-

Tarea 3 y 4: Preparé el Entorno de Athena

Como Athena necesita un lugar para guardar los resultados de sus consultas, volví a **S3** y creé un **segundo bucket**. Este lo designé exclusivamente para los resultados de Athena.

El bucket "resultadodelcvlab4" se creó correctamente

Para cargar archivos y carpetas, o para configurar ajustes adicionales del bucket, elija [Ver detalles](#).

Buckets de uso general

Todas las regiones de AWS

Buckets de directorio

Buckets de uso general (2)

Información

Los buckets son contenedores de datos almacenados en S3.

Buscar buckets por nombre

Nombre

▲

Región de AWS

▼

Fecha de creación

▼

[datosdelibrolab4](#)

EE.UU. Este (Norte de Virginia) us-east-1

5 Sep 2025 7:26:26 PM CST

[resultadodelcvlab4](#)

EE.UU. Este (Norte de Virginia) us-east-1

5 Sep 2025 7:29:00 PM CST

Copiar ARN

Vaciar

Eliminar

Crear bucket

▶

Instantánea de la cuenta

Información

Actualizado a diario

Storage Lens ofrece visibilidad sobre el uso del almacenamiento y las tendencias de actividad.

▶

Resumen de acceso externo: *nuevo*

Información

Actualizado a diario

Los resultados de acceso externo le ayudan a identificar los permisos de los buckets que permiten el acceso público o desde otras cuentas de AWS.

Administrar la configuración

Ver los resultados

Administrar la configuración

Administrar la configuración

Ver los resultados

Administrar la configuración

Ubicación y codificación de los resultados de la consulta

Location of query result - optional

Enter a 256-character path where the query result will be saved as an object

View

Review S3

Expected bucket owner - optional

Specify the AWS account ID that you intend to use for the owner of your query result output location bucket

Enter AWS account ID

☐ Assign bucket owner full control over query results

Enabling this option grants the owner of the S3 query result bucket full control over the query results. This means that if your query result bucket is owned by another account, you grant full control over your query results to the other account.

Choose S3 data set

13 buckets

Bucket (1/2)

Filter bucket

Name

Creation date

☐ dataset1@us-east-1

2023-09-05T19:26:24-00:00-06:00

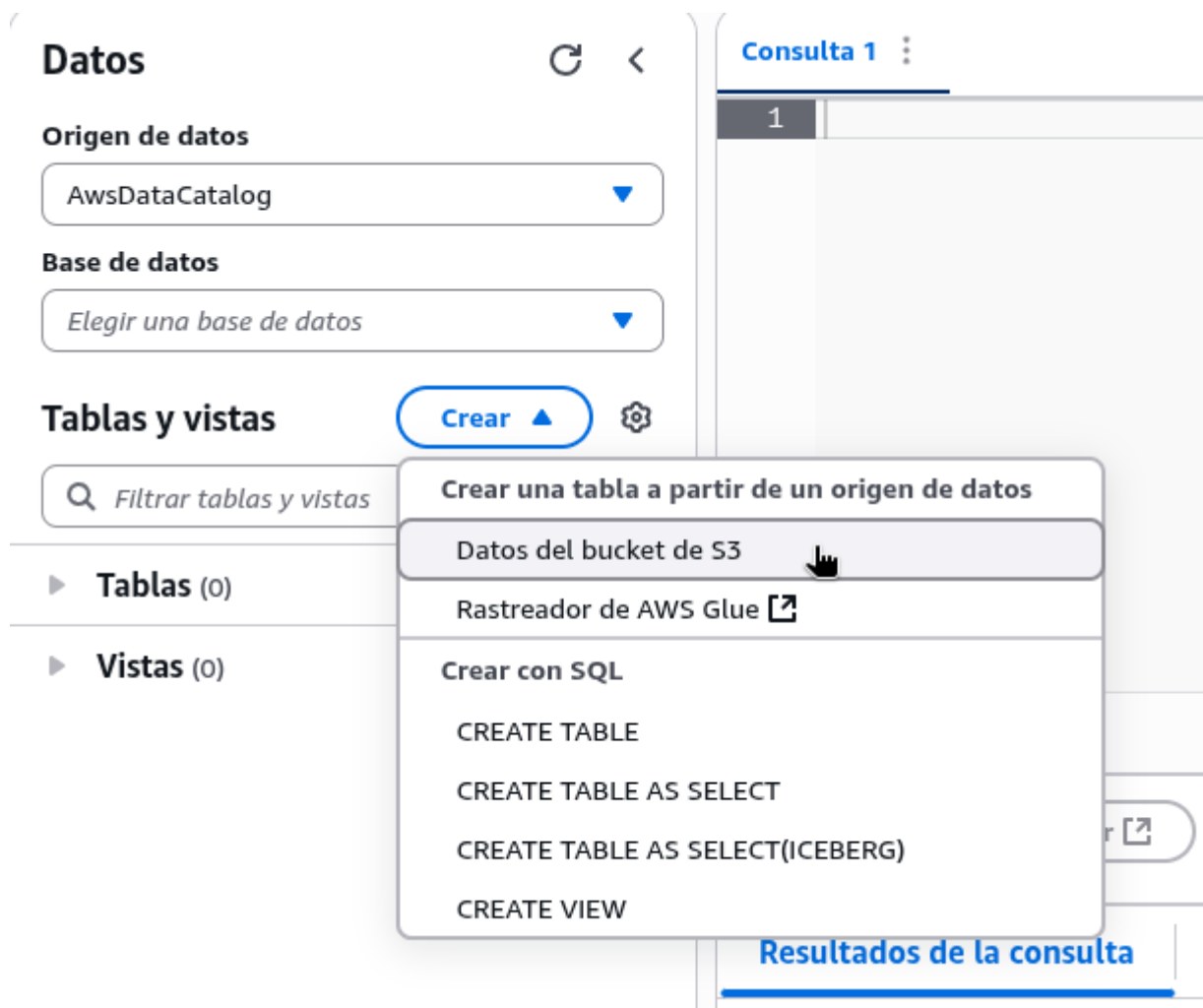
☒ resultbucket@us-east-1

2023-09-05T19:29:00-00:00-06:00

Cancel

Choose

Administrar la configuracion



Conjunto de datos [Información](#)

Ubicación del conjunto de datos de entrada

Choose S3 data set

S3 buckets

Bucket (1/2)

	Name
<input checked="" type="radio"/>	datosdelibrolab4
<input type="radio"/>	resultadodelcvlab4

Formato de datos [Información](#)

Tipo de tabla

Formato de archivo

Biblioteca SerDe

Propiedades de SerDe - optional

Nombre	Valor
<input type="text" value="field.delim"/>	<input type="text" value=","/>

Luego, seleccioné el bucket donde había subido el archivo `books.csv`. En las configuraciones, cambié el formato del archivo a **CSV**. El siguiente paso fue muy importante: en los detalles de la tabla, agregué las columnas manualmente, definiendo sus nombres y tipos de datos: `autor string`, `libro string`, `year int`.

Detalles de la columna

Nombre de columna debe tener entre 1-128 caracteres. Los caracteres válidos son a-z, A-Z, 0-9, _(guión bajo). Ciertos tipos de columna avanzados (en concreto, las estructuras) no están expuestos en esta interfaz.

Nombre de columna	Tipo de columna	Descripción - <i>optional</i>	
<input type="text" value="autor"/>	<input type="text" value="string"/>	<input type="text" value="Introduzca una descripción"/>	<button>Eliminar</button>
<input type="text" value="libro"/>	<input type="text" value="string"/>	<input type="text" value="Introduzca una descripción"/>	<button>Eliminar</button>
<input type="text" value="year"/>	<input type="text" value="string"/>	<input type="text" value="Introduzca una descripción"/>	<button>Eliminar</button>

Agregar una columnaAgregar columnas en bloque

► **Propiedades de la tabla - *optional*** [Información](#)

► **Detalles de la partición - *optional*** [Información](#)

Las particiones son una forma de agrupar información específica. Nombre de columna debe tener entre 1-128 caracteres. Los caracteres válidos son a-z, A-Z, 0-9, _(guión bajo). Ciertos tipos de columna avanzados (en concreto, las estructuras) no están expuestos en esta interfaz.

► **Agrupación en buckets - *optional***

Vista previa de la consulta de la tabla

La vista previa de la consulta de la tabla se completará en el editor de consultas. La creación de tablas le permite estar preparado para realizar consultas en tiempo real en el editor de consultas.

```
1 CREATE EXTERNAL TABLE IF NOT EXISTS `LibrosDB`.`TablaLibros` (  
2   `autor` string,  
3   `libro` string,  
4   `year` string  
5 )  
6 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'  
7 WITH SERDEPROPERTIES ('field.delim' = ',')  
8 STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  
9 LOCATION 's3://datosdellibrolab4/'  
10 TBLPROPERTIES ('classification' = 'csv');
```

CancelarCrear tabla

Finalmente, al hacer clic en **"Crear tabla"**, Athena procesó mi solicitud y la tabla quedó lista para ser consultada.

Detalles de la columna

Nombre de columna debe tener entre 1-128 caracteres. Los caracteres válidos son a-z, A-Z, 0-9, _(guión bajo). Ciertos tipos de columna avanzados (en concreto, las estructuras) no están expuestos en esta interfaz.

Nombre de columna <input type="text" value="autor"/>	Tipo de columna <input type="text" value="string"/>	Descripción - <i>optional</i> <input type="text" value="Introduzca una descripción"/>	<button>Eliminar</button>
Nombre de columna <input type="text" value="libro"/>	Tipo de columna <input type="text" value="string"/>	Descripción - <i>optional</i> <input type="text" value="Introduzca una descripción"/>	<button>Eliminar</button>
Nombre de columna <input type="text" value="year"/>	Tipo de columna <input type="text" value="string"/>	Descripción - <i>optional</i> <input type="text" value="Introduzca una descripción"/>	<button>Eliminar</button>

Agregar una columna Agregar columnas en bloque

► Propiedades de la tabla - *optional* [Información](#)

► Detalles de la partición - *optional* [Información](#)

Las particiones son una forma de agrupar información específica. Nombre de columna debe tener entre 1-128 caracteres. Los caracteres válidos son a-z, A-Z, 0-9, _(guión bajo). Ciertos tipos de columna avanzados (en concreto, las estructuras) no están expuestos en esta interfaz.

► Agrupación en buckets - *optional*

Vista previa de la consulta de la tabla

La vista previa de la consulta de la tabla se completará en el editor de consultas. La creación de tablas le permite estar preparado para realizar consultas en tiempo real en el editor de consultas.

```
1 CREATE EXTERNAL TABLE IF NOT EXISTS 'LibrosDB'. 'TablaLibros' (  
2   'autor' string,  
3   'libro' string,  
4   'year' string  
5 )  
6 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'  
7 WITH SERDEPROPERTIES ('field.delim' = ',')  
8 STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'  
9 LOCATION 's3://datosdellibrolab4/'  
10 TBLPROPERTIES ('classification' = 'csv');
```

[Cancelar](#)

[Crear tabla](#)

Tarea 6: Realicé Consultas SQL 🔍

Con la tabla ya creada y seleccionada, pude empezar a hacer consultas SQL para explorar los datos.

1. Para ver **todos los libros**, usé `SELECT * FROM tablalibros;` .
2. Para encontrar los libros escritos por **"Dean Koontz"**, escribí `SELECT * FROM tablalibros WHERE autor = 'Dean Koontz';` .
3. Para ver los libros publicados **después del año 2000**, la consulta fue `SELECT * FROM tablalibros WHERE year > 2000;` .
4. Para obtener una lista de los **autores únicos**, utilicé `SELECT DISTINCT autor FROM tablalibros;` .
5. Para ordenar los libros **del más antiguo al más reciente**, usé `SELECT * FROM tablalibros ORDER BY year ASC;` .
6. Y para encontrar los libros publicados **en 1999**, la consulta fue `SELECT libro, year FROM tablalibros WHERE year = 1999;` .

Consulta 1 ✕

Consulta 2 ✕

1

SELECT DISTINCT autor FROM tablalibros;

SQL Ln 1, Col 40

Ejecutar de nuevo

Explicar ↗

Cancelar

Borrar

Crear ▼

Resultados de la consulta

Estado de la consulta

✓ Completado

Resultados (50)

🔍 Filas de búsqueda

#	▼	autor
1		Autor
2		Richard Bruce Wright
3		Jack Canfield
4		Robert Hendrickson
5		Julia Oliver
6		Toni Morrison
7		Stephan Jaramillo
8		LAURA HILLENBRAND
9		Jane Austen
10		Anne Rivers Siddons

Adicionalmente, aprendí que para eliminar una tabla se usa `DROP TABLE nombre_tabla` y para una base de datos, `DROP DATABASE nombre_base_de_datos`. Es importante recordar que hay que borrar las tablas antes de borrar la base de datos.

Con esto, he logrado completar el laboratorio con éxito. ✓