

Asunto:

Los **ENDPOINT** de la API en producción en **RENDER** como plataforma alternativa o de respaldo.

Fecha: 22/05/2025

Desarrollador: Luis Mujica

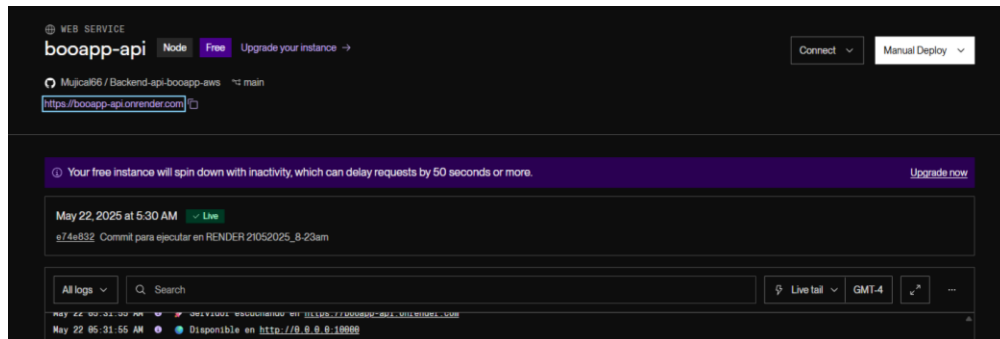
Versión de la API: 1.0

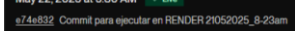
Ambiente: Producción

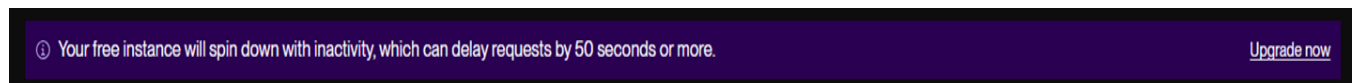
Servicio: RENDER (<https://render.com/>)

Prefijo de los ENDPOINT: <https://booapp-api.onrender.com/v1/backend-api-booapp-aws>

Observación:



La imagen muestra la ejecución de las API (, pero el mensaje mas importante es el siguiente:



Traducción: Su instancia gratuita dejará de funcionar por inactividad, lo que puede retrasar las solicitudes 50 segundos o más.

Este mensaje indica que, al tratarse de una cuenta gratuita, la API entra en estado de inactividad cuando no recibe solicitudes. La primera vez que se realiza una petición, el tiempo de respuesta puede tardar 50 segundos o más.

Por favor, al realizar las pruebas, tengan en cuenta esta información y sean pacientes. Otro aspecto a considerar es la velocidad de conexión a Internet que tengan al momento de ejecutar cualquier prueba con herramientas de TESTING (POSTMAN).

1. Resumen Ejecutivo

Durante el período de ejecución en producción de la **API boo-api**, se realizaron pruebas de estabilidad, rendimiento y seguridad en los diferentes **ENDPOINTS**. Se identificaron áreas de mejora y se implementaron correcciones necesarias para optimizar el servicio.

2. ENDPOINTS Implementados

Lista de los ENDPOINTS que están operando en producción, su funcionalidad y el método HTTP utilizado.

Collection: coldocHistorico https://booapp-api.onrender.com/v1/backend-api-booapp-aws			
ENDPOINT	Método	Descripción	Estado
/coldocHistorico	GET	Obtiene la lista de documento histórico.	Operativo
/coldocHistorico/{id}	GET	Obtiene detalles de un documento histórico.	Operativo
/coldocHistorico	POST	Crea un nuevo documento histórico.	Operativo
/coldocHistorico/{id}	PATCH	Actualiza datos de un documento histórico.	Operativo
/coldocHistorico/{id}	DELETE	Elimina un documento histórico.	Operativo

Collection: colevento https://booapp-api.onrender.com/v1/backend-api-booapp-aws			
ENDPOINT	Método	Descripción	Estado
/colevento	GET	Obtiene la lista del evento paranormal.	Operativo
/colevento/{id}	GET	Obtiene detalles de un evento paranormal.	Operativo
/colevento	POST	Crea un nuevo evento paranormal.	Operativo
/colevento/{id}	PATCH	Actualiza datos de un evento paranormal.	Operativo
/colevento/{id}	DELETE	Elimina un evento paranormal.	Operativo

Collection: colmapa https://booapp-api.onrender.com/v1/backend-api-booapp-aws			
ENDPOINT	Método	Descripción	Estado
/colmapa	GET	Obtiene la lista de todos los mapas (texto).	Operativo
/colmapa/{id}	GET	Obtiene detalles de un mapa.	Operativo
/colmapa	POST	Crea un nuevo mapa.	Operativo
/colmapa/{id}	PATCH	Actualiza datos de un mapa.	Operativo
/colmapa/{id}	DELETE	Elimina un mapa.	Operativo

Collection: colobservacion https://booapp-api.onrender.com/v1/backend-api-booapp-aws			
ENDPOINT	Método	Descripción	Estado
/colobservacion	GET	Obtiene la lista de las observaciones.	Operativo
/colobservacion/{id}	GET	Obtiene detalles de una observación.	Operativo
/colobservacion	POST	Crea una nueva observación.	Operativo
/colobservacion/{id}	PATCH	Actualiza datos de una observación.	Operativo
/colobservacion/{id}	DELETE	Elimina una observación.	Operativo

Collection: colusuario https://booapp-api.onrender.com/v1/backend-api-booapp-aws			
ENDPOINT	Método	Descripción	Estado
/colusuario	GET	Obtiene la lista de los usuarios.	Operativo
/colusuario/{id}	GET	Obtiene detalles de un usuario.	Operativo
/colusuario	POST	Crea un nuevo usuario.	Operativo
/colusuario/{id}	PATCH	Actualiza datos de un usuario.	Operativo
/colusuario/{id}	DELETE	Elimina un usuario.	Operativo

Ejemplo de dirección completa del ENDPOINT:

Método GET: Visualiza toda la información de todos los registros.

<https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento>

Métodos GET {id} – PATCH {id} – DELETE {id}

{id} = deben colocar el id del registro al cual se requiere hacer referencia.

"_id": "67f4fc7c9539facb37217adc", este id es un registro de la collection.

<https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento/67f4fc7c9539facb37217adc>

3. Rendimiento y Métricas

Resultados de pruebas de carga, tiempos de respuesta, consumo de recursos y comparación con versiones anteriores.

Prueba:

Se realizaron pruebas de carga con 1 solicitudes simultáneas, obteniendo un tiempo promedio de respuesta de 250 ms para las consultas GET y 400 ms para las operaciones POST. Las mismas métricas pueden ser por el nivel gratuito que estamos utilizando.

4. Errores y Problemas Identificados

Descripción de los errores detectados, impactos y acciones correctivas aplicadas.

Resultados relevantes:

- **Error 500 en /usuario/{id} (PATCH):** Se detectó un problema en la validación de datos.
- **Tiempo de respuesta elevado en /usuario (GET):** Al activar el servidor luego de una inactividad de solicitudes. Luego de su activación luego de los 50 o más segundos que se deben esperar las consultas a la base de datos, reduciendo el tiempo de ejecución.

5. Mejoras Implementadas

Lista de las mejoras realizadas con detalles sobre la implementación:

- Se agregó monitoreo en tiempo real para detectar fallas con mayor rapidez.
- Se optimizó el uso de memoria en la aplicación reduciendo el consumo un 15%.

6. Recomendaciones y Próximos Pasos

Sugerencias para mejoras futuras y próximos cambios previstos en la API:

- Realizar pruebas de carga con volúmenes mayores de datos.
 - Optimizar aún más las consultas en la base de datos para reducir latencia.
 - Implementar autenticación avanzada para mejorar la seguridad.
-

Estructura de los model en códigos de todas las collections de las Base de Datos.

Los siguientes códigos suministrados son la estructura de los modelos que utilizaron para la creación de la estructura de cada collection que tienen las bases de datos.

Cuando realicen las pruebas deben ajustar las variables y los datos según lo que les indican los model suministrados para que no generen ningún tipo de error de variables no encontradas.

// coldochistorico.model.js

// Creación del modelo de Schema de la Collections Documentos Historicos

// Importar mongoose

const mongoose = require("mongoose");

const coldochistoricoSchema = mongoose.Schema(

```
{
  idCodHistorico: {
    type: String,
    required: true,
    trim: true,
    unique: true,
    maxlength: 10,
  },
  titulo: {
    type: String,
    required: true,
    trim: true,
    unique: true,
  },
  descripcion: {
    type: String,
    required: true,
    trim: true,
  },
  fechaDocumento: {
    type: Date,
    default: Date.now,
  },
  urlAlmacenado: {
    type: String,
```

```

    // required: true,
    // trim: true,
  },
  idUsuario: {
    type: String,
    // required: true,
    trim: true,
  },
  idUsuario: {
    type: String,
    required: true,
    trim: true,
  },
},
{
  timestamps: true,
}
);

// Define el Modelo
const colDocHistorico = mongoose.model(
  "colDocHistoricos",
  coldochistoricoSchema
);

// Export el Modelo
module.exports = colDocHistorico;

```

// colevento.model.js

// Creación del modelo de Schema de la Collections coleventoslep / Contenidos

```

// Importar mongoose
// import mongoose from "mongoose";
const mongoose = require("mongoose");

// Definir el Schema
const coleventoSchema = mongoose.Schema(
  {
    idCodEvento: {
      type: String,
      required: true,
      trim: true,
      unique: true,
      maxlength: 10,
    },
    titulo: {
      type: String,
      required: true,

```

```
    trim: true,
    unique: true,
  },
  descripcion: {
    type: String,
    // required: true,
    trim: true,
  },
  ubicacion: {
    type: String,
    // required: true,
    trim: true,
  },
  latitud: {
    type: Number,
    // required: true,
    default: 10.506151128331723,
  },
  longitud: {
    type: Number,
    // required: true,
    default: -66.9130464662426,
  },
  fechaCreacion: {
    type: Date,
    default: Date.now,
  },
  imagen: {
    type: String,
  },
  video: {
    type: String,
  },
  comentario: {
    type: String,
    // required: true,
    trim: true,
  },
  popularidad: {
    type: Number,
    // required: true,
    default: 0,
  },
  estado: {
    type: String,
    // requerid: true,
    trim: true,
  },
  fechaVisual: {
    type: Date,
```

```

    default: Date.now,
  },
  comentarioRev: {
    type: String,
    // required: true,
    trim: true,
  },
  tipoContenido: {
    type: String,
    // required: true,
    trim: true,
  },
  idUsuario: {
    type: String,
    required: true,
    trim: true,
  },
},
{
  timestamps: true,
}
);

// Define el Modelo
const colEventoslep = mongoose.model("colEventoslep", coleventoSchema);

// Export el Modelo
module.exports = colEventoslep;

```

// colmapa.model.js

// Creación del modelo de Schema de la Collections Mapas

```

// import mongoose from "mongoose";
// Importar mongoose
const mongoose = require("mongoose");

const colmapaSchema = mongoose.Schema(
{
  idCodMapa: {
    type: String,
    required: true,
    trim: true,
    unique: true,
    maxlength: 10,
  },
  titulo: {
    type: String,
    required: true,
    trim: true,
  },
}
);

```

```

    unique: true,
  },
  descripcion: {
    type: String,
    required: true,
    trim: true,
  },
  ubicaciones: {
    type: String,
    required: true,
    trim: true,
  },
  idUsuario: {
    type: Object,
    // required: true,
    // trim: true,
  },
  fechaCreacion: {
    type: Date,
    default: Date.now,
  },
  idUsuario: {
    type: String,
    required: true,
    trim: true,
  },
},
{
  timestamps: true,
}
);

```

```

// Define el Modelo
const colmapa = mongoose.model("colmapas", colmapaSchema);

```

```

// Export el Modelo
module.exports = colmapa;

```

// colobservacion.model.js

// Creación del modelo de Schema o estructura de la collection Observaciones del Visitante / Usuario

```

// import mongoose from "mongoose";
// Importar mongoose
const mongoose = require("mongoose");

```

```

const colobservacionSchema = mongoose.Schema(
{
  idCodObser: {
    type: String,

```



```

    required: true,
    trim: true,
    unique: true,
    maxlength: 10,
  },
  idUsuario: {
    type: Object,
    // required: true,
    // trim: true,
  },
  mensaje: {
    type: String,
    required: true,
    trim: true,
    unique: true,
  },
  fechaEnvio: {
    type: Date,
    default: Date.now,
  },
  mRespondido: {
    type: Boolean,
    // required: true,
    // trim: true,
  },
  respuesta: {
    type: String,
    // required: true,
    trim: true,
  },
  subidoPor: {
    type: String,
    // required: true,
    trim: true,
  },
},
{
  timestamps: true,
}
);

// Define el Modelo
const colobservacion = mongoose.model("colObservaciones", colobservacionSchema);

// Export el Modelo
module.exports = colobservacion;

```

```
// colusuario.model.js
```

```
// Creación del modelo de Schema o estructura de la collection Usuarios
```

```
// import mongoose from "mongoose";
```

```
// Importar mongoose
```

```
const mongoose = require("mongoose");
```

```
const colusuarioSchema = mongoose.Schema(
```

```
{
```

```
  IdDni: {
```

```
    type: String,
```

```
    required: true,
```

```
    trim: true,
```

```
    unique: true,
```

```
    maxlength: 10,
```

```
  },
```

```
  nombre: {
```

```
    type: String,
```

```
    required: true,
```

```
    trim: true,
```

```
    unique: true,
```

```
  },
```

```
  email: {
```

```
    type: String,
```

```
    required: true,
```

```
    trim: true,
```

```
    unique: true,
```

```
  },
```

```
  password: {
```

```
    type: String,
```

```
    trim: true,
```

```
    unique: true,
```

```
  },
```

```
  rol: {
```

```
    type: String,
```

```
    required: true,
```

```
    trim: true,
```

```
  },
```

```
  fotoPerfil: {
```

```
    type: String,
```

```
    require: true,
```

```
  },
```

```
  fechaIngreso: {
```

```
    type: Date,
```

```
    default: Date.now,
```

```
  },
```

```
  ultimaActividad: {
```

```
    type: Date,
```

```
    default: Date.now,
```

```
  },
```

```
activo: {
  type: Boolean,
  // required: true,
  // trim: true,
},
},
{
  timestamps: true,
}
);

// Define el Modelo
const colUsuario = mongoose.model("colUsuarios", colusuarioSchema);

// Export el Modelo
module.exports = colUsuario;
```

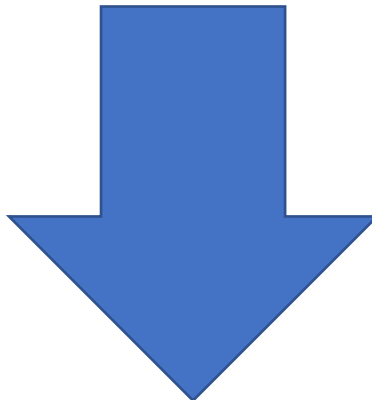
Bien les comparto un resumen de las pantallas de la prueba que realice el día 21-05-2025 no las coloque todas porque las pruebas realizadas fueron a cada ENDPOINT de cada una de las Collections que están en la Base de Datos, ya los encargados de las pruebas pueden realizar sus pruebas para verificar las posibles correcciones y ajustes que deben tener.

Recuerden tomar en consideración las observaciones que les indique al principio al momento de realizar sus pruebas.

Gracias y atento a sus recomendaciones,

Luis Mujica
Desarrollador

PANTALLAS DE LA PRUEBA REALIZADAS EN UN ENTORNO CONTROLADO



Saludos, Team;

ASUNTO: Informe de Primera prueba de uso de la API con los ENDPOINT GET (Consulta General) y POST (Incorporar datos).

FECHA: 21-05-2025

Team les estoy enviando el resultado de las pantallas de la primera prueba de la API con los ENDPOINT mencionados anteriormente a la Collections de Contenido de Eventos paranormales, allí se les muestra las pantallas de la base de datos en MongoDB Atlas y la plataforma de AWS con el Bucket donde se almacenan los videos y la imagen.

Lo importante es el tiempo de respuesta de los envíos o consultas que están bien bajos tanto con el **POST** y **GET**.

Ya pueden iniciar a realizar las pruebas pero recordando las limitaciones que tenemos a nivel de capacidad entre otras.

INICIEN LAS PRUEBAS PERTINENTES PARA VER COMO SE COMPORTA LA API.



Las otras pruebas las hago el viernes, pero ya pueden ir organizando las pruebas finales para la API.

Atentamente,

Luis Mujica.

idCodEvento = 10

titulo = El Silbón

descripcion = El Silbón es una figura de la leyenda latinoamericana, especialmente presente en las llanuras de Venezuela y Colombia, que se describe como un espíritu errante o un alma en pena. Se dice que el Silbón es un hombre alto y delgado, con ropa desgastada, un sombrero grande y un saco lleno de huesos. La leyenda cuenta que el Silbón acecha en la noche y utiliza su silbido para confundir y luego matar a sus víctimas, especialmente borrachos y parranderos. La leyenda del Silbón tiene más de 300 años y se cree que fue inventada por terratenientes y ganaderos para evitar que la gente saliera de noche.

La leyenda del Silbón tiene más de 300 años y sigue viva en la región, donde muchas muertes se atribuyen a él.

El Silbón utiliza un silbido inquietante y confuso que puede sonar cerca o lejos, dependiendo de si está cerca o lejos.

Algunos creen que la leyenda del Silbón fue inventada por terratenientes y ganaderos para evitar que la gente saliera de noche y se dedicara al abigeato.

ubicacion = Se origina en la región de Los Llanos, especialmente en el oriente de Venezuela.

rol = Turista.

latitud = 7.878558

longitud = -67.468125

fechaCreacion = 2025-04-24

imagen =

video =

comentario = Excelente leyenda de los llanos venezolanos.

popularidad = 5

estado = Aprobado

fechaVisual = 2025-05-21

comentarioRev = Ya fue aprobado por el administrador de contenidos.

tipoContenido = Leyenda

idUsuario = COD-10

**Datos Suministrados para la primera prueba sobre
La collections Eventos**

```
idCodEvento = 10

titulo = El Silbón

descripcion = El Silbón es una figura de la leyenda latinoamericana, especialmente presente en las llanuras de Venezuela y Colombia, que se describe como un espíritu errante o un alma en pena. Se dice que el Silbón es un hombre alto y delgado, con ropa desgastada, un sombrero grande y un saco lleno de huesos. La leyenda cuenta que el Silbón acecha en la noche y utiliza su silbido para confundir y luego matar a sus víctimas, especialmente borrachos y parranderos. La leyenda del Silbón tiene más de 300 años y se cree que fue inventada por terratenientes y ganaderos para evitar que la gente saliera de noche. La leyenda del Silbón tiene más de 300 años y sigue viva en la región, donde muchas muertes se atribuyen a él. El Silbón utiliza un silbido inquietante y confuso que puede sonar cerca o lejos, dependiendo de si está cerca o lejos. Algunos creen que la leyenda del Silbón fue inventada por terratenientes y ganaderos para evitar que la gente saliera de noche y se dedicara al abigeato.

ubicacion = Se origina en la región de Los Llanos, especialmente en el oriente de Venezuela.

rol = Turista.

latitud = 7.878558

longitud = -67.468125

fechaCreacion = 2025-04-24

imagen =

video =

comentario = Excelente leyenda de los llanos venezolanos.

popularidad = 5

estado = Aprobado

fechaVisual = 2025-05-21

comentarioRev = Ya fue aprobado por el administrador de contenidos.

tipoContenido = Leyenda

idUsuario = COD-10
```

Ln 16, Col 22 | 1.478 caracteres. | 100% | Windows (CRLF) | UTF-8

SERVIDOR RENDER EN EJECUCIÓN

vercel - Buscar con Google

Mujical66/Backend-api-booapp

booapp-api - Web Service - Re

booapp-api.onrender.com/v1/

Commit para Render 21052025

Precios | Render

Cluster0 Data | Cloud: MongoD

dashboard.render.com/web/srv-d0mr6ah5pdvs739nb3ug/deloys/dep-d0msc095pdvs739o4ong?r=2025-05-21%4012%3A23%3A34~2025-05-21%4012%3A26%3A25

Google Lens

My Workspace

booapp-api

Dashboard

booapp-api

Events

Settings

MONITOR

Logs

Metrics

MANAGE

Environment

Shell

Scaling

Previews

Disks

Jobs

Changelog

Invite a friend

Contact support

Render Status

WEB SERVICE

booapp-api

Node

Free

Upgrade your instance

Connect

Manual Deploy

Mujical66 / Backend-api-booapp-aws

main

https://booapp-api.onrender.com

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

Upgrade now

May 21, 2025 at 8:24 AM

Live

e74e832 Commit para ejecutar en RENDER 21052025_8-23am

All logs

Search

May 21, 8:23 AM - 8:26 AM

GMT-4

May 21 08:24:44 AM

Found 0 vulnerabilities

May 21 08:24:47 AM

--> Uploading build...

May 21 08:24:52 AM

--> Uploaded in 4.2s. Compression took 1.4s

May 21 08:24:52 AM

--> Build successful

May 21 08:24:54 AM

--> Deploying...

May 21 08:25:11 AM

--> Running 'npm start'

May 21 08:25:12 AM

May 21 08:25:12 AM

> backend-api-booapp-aws@1.0.0 start

May 21 08:25:12 AM

> node ./src/index.js

May 21 08:25:12 AM

May 21 08:25:19 AM

MongoDB Atlas conectado correctamente

May 21 08:25:19 AM

Conexión a MongoDB establecida

May 21 08:25:19 AM

Servidor escuchando en https://booapp-api.onrender.com

May 21 08:25:19 AM

Disponible en http://0.0.0.0:10000

May 21 08:25:19 AM

5/21/2025, 12:25:19 PM

May 21 08:25:25 AM

--> Your service is live

Need better ways to work with logs? Try the Render CLI or set up a log stream integration

Noten el tiempo de respuesta de la consulta a la Collection de Usuarios: 31,70 s

HomeWorkspacesExplore

Search Postman

Sign InCreate Account

You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

HistoryNewImport

Today

Yesterday

May 19

May 1

GET https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento

GET https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento

GET https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento

GET https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento

GET https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento

GET https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento

GET https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/api/colusuario/67f509d9d9f

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/api/colusuario

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/v1/backend-api-booapp-aw

GET http://localhost:5000/v1/backend-api-booapp-aw

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

GET https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento

Send

ParamsAuthorizationHeaders(8)BodyPre-request ScriptTestsSettings

Query Params

Key	Value	Bulk Edit
Key	Value	

BodyCookiesHeaders(15)Test Results(1/2)

PrettyRawPreviewVisualizeJSON

```
1  {
2    "success": true,
3    "message": "Successfully retrieved all events",
4    "data": {
5      "colEventosleps": [
6        {
7          "_id": "67f4fc7c9539facb37217adc",
8          "titulo": "Atardecer en la playa",
9          "descripcion": "Una vista espectacular del sol ocultándose en el horizonte.",
10         "ubicacion": "Playa El Agua, Nueva Esparta, Venezuela",
11         "latitud": 10.9975,
12         "longitud": -63.8292,
13         "comentario": "¡Increíble paisaje! Definitivamente vale la pena verlo.",
14         "popularidad": 4.8,
15         "estado": "Activo",
16         "fechaVisual": "2025-04-07T18:30:00.000Z",
17         "comentarioRev": "La vista es impresionante, pero puede haber mucha gente al atardecer.",
18         "tipoContenido": "Imagen",
19         "fechaCreacion": "2025-04-08T10:37:48.874Z",
20         "__v": 0
21       },
22     ]
23   }
24 }
```

Status: 200 OKTime: 893 msSize: 1.18 KBSave Response

Noten el tiempo de respuesta de la consulta a la Collection de Eventos en 893 ms.

INICIO DE PRUEBA PARA INCORPORAR DATOS EN LA COLLECTION DE EVENTOS PARANORMALES

vercel - Buscar con Google

Mujical66/Backend-api-booapp

booapp-api - Web Service - R

booapp-api.onrender.com/v1/

Commit para Render 21052025

Precios | Render

Cluster0 Data | Cloud: MongoD

cloud.mongodb.com/v2/67978ae684e2051b06879b6a#/metrics/replicaSet/67978db0ca8fc7a44c34ebd/explorer/booappDB/coleventosleps/find

Atlas

Luis's Org - 2...

Access Manager

Billing

All Clusters

Get Help

Luis

BooApp

Data Services

Charts

Overview

DATABASE

Clusters

SERVICES

Atlas Search

Stream Processing

Triggers

Migration

Data Federation

SECURITY

Quickstart

Backup

Database Access

Network Access

Advanced

Goto

LUIS'S ORG - 2024-12-17 > BOOAPP > DATABASES

ClusterO

VERSION 8.0.9

REGION AWS Sao Paulo (sa-east-1)

Overview

Real Time

Metrics

Collections

Atlas Search

Query Insights

Performance Advisor

Online Archive

Cmd Line Tools

Infrastructure As Code

DATABASES: 1

COLLECTIONS: 5

+ Create Database

Search Namespaces

booappDB

ooldochistoricos

coleventosleps

oolmapas

oolobservaciones

oolusuarios

booappDB.coleventosleps

STORAGE SIZE: 86KB

LOGICAL DATA SIZE: 1.98KB

TOTAL DOCUMENTS: 3

INDEXES TOTAL SIZE: 72KB

Find

Indexes

Schema Anti-Patterns

Aggregation

Search Indexes

Generate queries from natural language in Compass

INSERT DOCUMENT

Filter

Type a query: { field: 'value' }

Reset

Apply

Options

QUERY RESULTS: 1-3 OF 3

_id: ObjectId('67f4fc7c9539facb37217adc')

titulo: "Atardecer en la playa"

descripcion: "Una vista espectacular del sol ocultándose en el horizonte."

ubicacion: "Playa El Agua, Nueva Esparta, Venezuela"

latitud: 10.9975

longitud: -63.8292

comentario: "¡Increíble paisaje! Definitivamente vale la pena verlo."

popularidad: 4.8

estado: "Activo"

fechaVisual: 2025-04-08T18:30:00.000+00:00

comentarioRev: "La vista es impresionante, pero puede haber mucha gente al atardecer."

tipoContenido: "Imagen"

fechaCreacion: 2025-04-08T10:37:48.874+00:00

__v: 0

_id: ObjectId('67f543a045ed8f469b750a93')

titulo: "Titulo Prueba 2 - Contenido / Atardecer en la playa."

descripcion: "Titulo Prueba 2 - Contenido / Una vista espectacular del sol ocultándose..."



Nro. de registro en la Collection son 3.

INICIO DE PRUEBA PARA INCORPORAR DATOS EN LA COLLECTION DE EVENTOS PARANORMALES

us-east-2.console.aws.amazon.com/s3/buckets/s3-uploadimages-booapp-bucket?region=us-east-2&tab=objects&bucketType=general

Buscar

S3

Lambda

Amazon S3

Buckets

s3-uploadimages-booapp-bucket

Amazon S3

Buckets de uso general

Buckets de directorio

Buckets de tablas

Concesiones de acceso

Puntos de acceso

Puntos de acceso del objeto Lambda

Puntos de acceso de varias regiones

Operaciones por lotes

Analizador de acceso de IAM para S3

Configuración de bloqueo de acceso público correspondiente a esta cuenta

Storage Lens

Paneles

Grupos de Storage Lens

Configuración de AWS Organizations

Características destacadas 11

AWS Marketplace para S3

s3-uploadimages-booapp-bucket

Información

Objetos

Metadatos

Propiedades

Permisos

Métricas

Administración

Puntos de acceso

Objetos (1)

Copiar URI de S3

Copiar URL

Descargar

Abrir

Eliminar

Acciones

Crear carpeta

Cargar

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Buscar objetos por prefijo

<input type="checkbox"/>	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	imgEvento_e1fd0996f6774471e207af5a8ee67e479a451e430d98a4e589452503491d63e7	-	26 Apr 2025 9:01:13 AM -04	568.2 KB	Estándar

Solo existe una sola imagen de prueba

CloudShell

Comentarios

© 2025, Amazon Web Services, Inc. o sus filiales. [Privacidad](#) [Términos](#) [Preferencias de cookies](#)

INICIO DE PRUEBA PARA INCORPORAR DATOS EN LA COLLECTION DE EVENTOS PARANORMALES

The screenshot displays the Postman API client interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'Explore' options. A search bar labeled 'Search Postman' is also present. Below the navigation bar, a message states: 'You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.'

The main workspace is divided into three sections: 'History', 'Request Editor', and 'Response Viewer'. The 'History' section on the left shows a list of recent requests, including several GET requests to 'https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento' and some to 'http://localhost:5000/v1/backend-api-booapp-aws/colevento'.

The 'Request Editor' section is the central focus. It shows a POST request to 'https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento'. The request body is set to 'form-data'. The form fields are as follows:

Key	Value
ubicacion	Se origina en la region de Los Llanos, especialmente en el oriente de venezuela.
rol	Turista.
latitud	7.878558
longitud	-67.468125
fechaCreacion	2025-04-24
imagen	elSilbon.jpg
video	El silbon.mp4
comentario	Excelente leyenda de los llanos venezolanos.
popularidad	5
estado	Aprobado
fechaVisual	2025-05-21
comentarioRev	Ya fue aprobado por el administrador de contenidos.
tipoContenido	Leyenda

Below the form fields, the 'Body' tab is selected, showing the JSON representation of the request body. The JSON is as follows:

```
{  "tipoContenido": "Imagen",  "fechaCreacion": "2025-04-08T10:37:48.874Z",  "__v": 0,  "_id": "67f543a045ed8f469b750a93",  "titulo": "Título Prueba 2 - Contenido / Atardecer en la playa.",  "descripcion": "Título Prueba 2 - Contenido / Una vista espectacular del sol ocultándose en el horizonte."}
```

The 'Response Viewer' section at the bottom right shows the response status: 'Status: 200 OK', 'Time: 893 ms', and 'Size: 1.18 KB'. The response body is also visible, showing the same JSON structure as the request body.

Annotations on the image include a red box around the POST method and URL in the request editor, with an arrow pointing to the text 'ENDPOINT para ingresar datos de Evento.'. Another red box highlights the response status and size information. A large text overlay in the center of the form fields reads 'Área de Insertar los datos para su envío'.

INICIO DE PRUEBA PARA INCORPORAR DATOS EN LA COLLECTION DE EVENTOS PARANORMALES

The screenshot shows the Postman interface with a POST request configured to `https://booapp-api.onrender.com/v1/backend-api-booapp-aws/colevento`. The request body is in JSON format, containing event data. The response status is 200 OK, and the response body shows the event data successfully stored in the database.

Endpoint for entering event data.

Area of Result of the information send

Data uploaded to Database and to AWS Bucket both the video and the Image

```
{
  "data": {
    "Eventos": {
      "idCodEvento": "10",
      "titulo": "El Silbón",
      "descripcion": "El Silbón es una figura de la leyenda latinoamericana, especialmente presente en las llanuras de Venezuela y Colombia, que se describe como un espíritu errante o un alma en pena. Se dice que el Silbón es un hombre alto y delgado, con ropa desgastada, un sombrero grande y un saco lleno de huesos. La leyenda cuenta que el Silbón acecha en la noche y utiliza su silbido para confundir y luego matar a sus víctimas, especialmente borrachos y parranderos. La leyenda del Silbón tiene más de 300 años y se cree que fue inventada por terratenientes y ganaderos para evitar que la gente saliera de noche. La leyenda del Silbón tiene más de 300 años y sigue viva en la región, donde muchas muertes se atribuyen a él. El Silbón utiliza un silbido inquietante y confuso que puede sonar cerca o lejos, dependiendo de si está cerca o lejos. Algunos creen que la leyenda del Silbón fue inventada por terratenientes y ganaderos para evitar que la gente saliera de noche y se dedicara al abigeato.",
      "ubicacion": "Se origina en la región de Los Llanos, especialmente en el oriente de Venezuela.",
      "latitud": 7.878558,
      "longitud": -67.468125,
      "fechaCreacion": "2025-04-24T00:00:00.000Z",
      "imagen": "imgEvento_8c6eb0bf9da52997b4922005ef55c29f9ee019a3fdf6b0fa14b3bdd7f49ec025",
      "video": "vidEvento_03e873ea4984ab805fb8145d7ee899aedc1889bab52b8e59593f73bccf67b1cd",
      "comentario": "Excelente leyenda de los llanos venezolanos.",
      "popularidad": 5,
      "estado": "Aprobado",
      "fechaVisual": "2025-05-21T00:00:00.000Z",
      "comentarioRev": "Ya fue aprobado por el administrador de contenidos.",
      "tipoContenido": "Leyenda",
      "idUsuario": "COD-10",
      "_id": "682e1f4a8a8f128843e61adb",
      "createdAt": "2025-05-21T18:45:30.761Z",
      "updatedAt": "2025-05-21T18:45:30.761Z"
    }
  }
}
```

Pantalla de Resultados en la Base de Datos Collections Eventos y Bucket de AWS

vercel - Buscar con Google x Mujical66/Backend-api-booapp x booapp-api - Web Service - Re x booapp-api.onrender.com/v1/l x Commit para Render 21052025 x Precios | Render x Cluster0 Data | Cloud: Mongo x +

cloud.mongodb.com/v2/67978ae684e2051b06879b6a#/metrics/replicaSet/67978db0ca8cf7a44c34ebd/explorer/booappDB/coleventosleps/find

Atlas Luis's Org - 2... Access Manager Billing All Clusters Get Help Luis

BooApp Data Services Charts

Overview DATABASE Clusters SERVICES Atlas Search Stream Processing Triggers Migration Data Federation SECURITY Quickstart Backup Database Access Network Access Advanced Goto

LUIS'S ORG - 2024-12-17 > BOOAPP > DATABASES

Cluster0 Nombre del Cluster de almacenamiento. VERSION 8.0.9 REGION AWS Sao Paulo (sa-east-1)

Overview Real Time Metrics Collections Atlas Search Query Insights Performance Advisor Online Archive Cmd Line Tools Infrastructure As Code

DATABASES: 1 COLLECTIONS: 5

+ Create Database

Search Namespaces

booappDB

coldochistoricos

coleventosleps

colmapas

colobservaciones

colusuarios

booappDB.coleventosleps

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 3.63KB TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 72KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

INSERT DOCUMENT

Filter Type a query: { field: 'value' }

Reset Apply Options

QUERY RESULTS: 1-4 OF 4

```
{
  "_id": ObjectId('67f4fc7c9539facb37217adc'),
  "titulo": "Atardecer en la playa",
  "descripcion": "Una vista espectacular del sol ocultándose en el horizonte.",
  "ubicacion": "Playa El Agua, Nueva Esparta, Venezuela",
  "latitud": 10.9975,
  "longitud": -63.8292,
  "comentario": "¡Increíble paisaje! Definitivamente vale la pena verlo.",
  "popularidad": 4.8,
  "estado": "Activo",
  "fechaVisual": 2025-04-07T18:30:00.000+00:00,
  "comentarioRev": "La vista es impresionante, pero puede haber mucha gente al atardecer.",
  "tipoContenido": "Imagen",
  "fechaCreacion": 2025-04-08T10:37:48.874+00:00,
  "__v": 0
}
```

```
{
  "_id": ObjectId('67f543a045ed8f469b750a93'),
  "titulo": "Titulo Prueba 2 - Contenido / Atardecer en la playa.",
  "descripcion": "Titulo Prueba 2 - Contenido / Una vista espectacular del sol ocultándose..."
}
```

Área de Datos Subidos a Base de Datos

Pantalla de Resultados en la Base de Datos Collections Eventos y Bucket de AWS

Amazon S3

Buckets de uso general

Buckets de directorio

Buckets de tablas

Concesiones de acceso

Puntos de acceso

Puntos de acceso del objeto Lambda

Puntos de acceso de varias regiones

Operaciones por lotes

Analizador de acceso de IAM para S3

Configuración de bloqueo de acceso público correspondiente a esta cuenta

Storage Lens

Paneles

Grupos de Storage Lens

Configuración de AWS Organizations

Características destacadas

11

AWS Marketplace para S3

s3-uploadimages-booapp-bucket

Información

Objetos

Metadatos

Propiedades

Permisos

Métricas

Administración




Puntos de acceso

Nro. De Objetos Almacenados

Objetos (3)

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Buscar objetos por prefijo

	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	 imgEvento_8c6eb0bf9da52997b4922005ef55c29f9ee019a3fdf6b0fa14b3bdd7f49ec025	-	21 May 2025 2:45:30 PM -04	128.7 KB	Estándar
<input type="checkbox"/>	 imgEvento_e1fd0996f6774471e207af5a8ee67e479a451e430d98a4e589452503491d63e7	-	26 Apr 2025 9:01:13 AM -04	568.2 KB	Estándar
<input type="checkbox"/>	 vidEvento_03e873ea4984ab805fb8145d7ee899aedic1889bab52b8e59593f73bccf67b1cd	-	21 May 2025 2:45:30 PM -04	36.1 MB	Estándar

Video e Imagen Almacenados en el Bucket de AWS noten la fecha de almacenamiento (21-Mayo-2025 hora) son: **imgEvento_** para representar la imagen y **vidEvento_** para representar el video.

Copiar URI de S3

Copiar URL

Descargar

Abrir

Eliminar

Acciones

Crear carpeta

Cargar

CloudShell

Comentarios

© 2025, Amazon Web Services, Inc. o sus filiales.

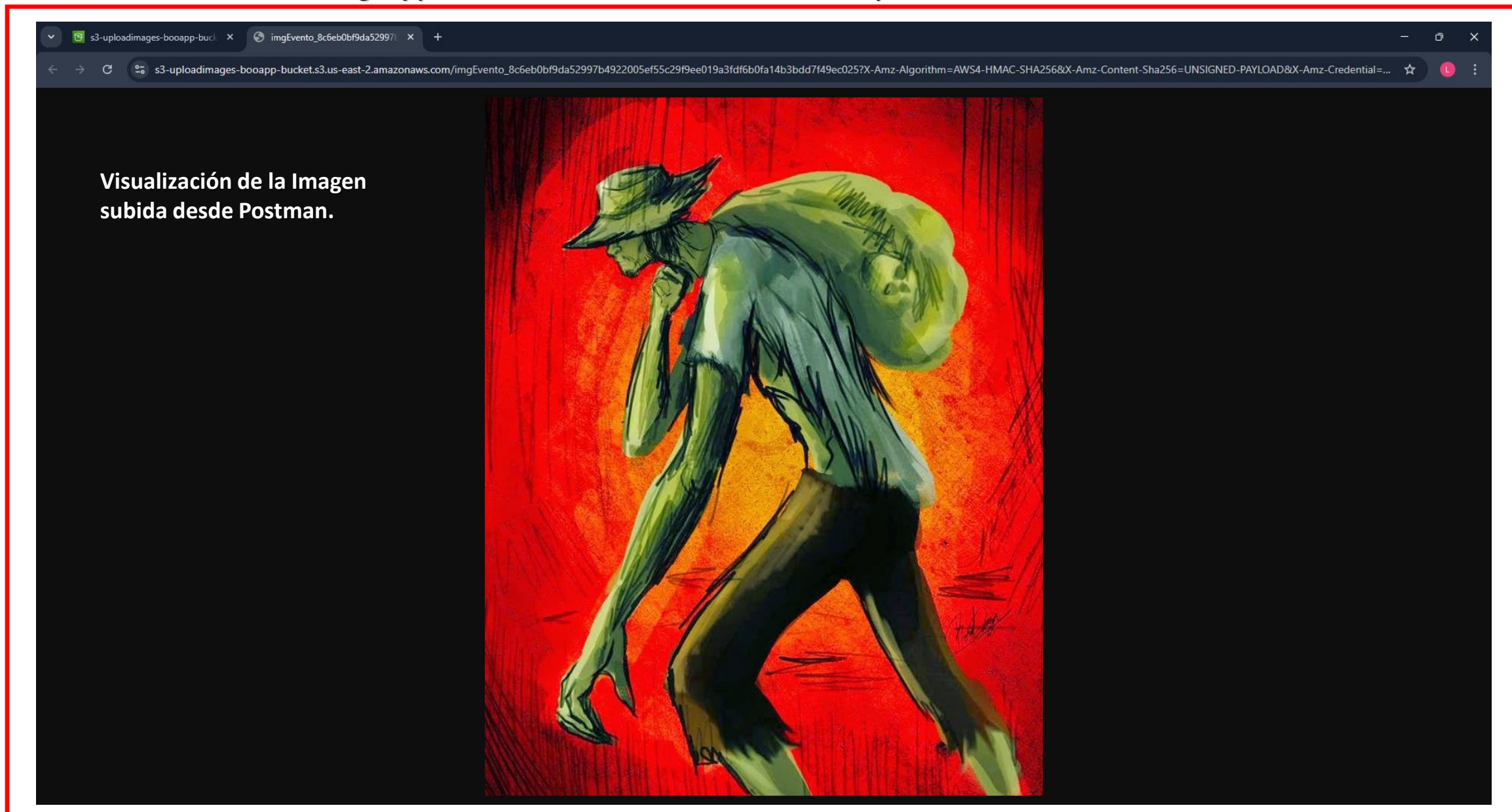
Privacidad

Términos

Preferencias de cookies

Pantalla de Resultados en la Base de Datos Collections Eventos y Bucket de AWS

Al seleccionar la imagen y pulsar el botón de abrir la muestra en la otra pestaña.



Pantalla de Resultados en la Base de Datos Collections Eventos y Bucket de AWS

Al seleccionar el video y pulsar el botón de abrir la muestra en la otra pestaña.

**Visualización del Video subido
desde Postman.**



Pantalla de Resultados en la Base de Datos Collections Eventos y Bucket de AWS

Al seleccionar el video y pulsar el botón de abrir la muestra en la otra pestaña.

