

# Universidad Tecnológica de Durango

## Tecnologías de la Información

### Aplicaciones Web Os

#### Actividad

“Actividades y tareas:

API's Base de Datos”

Alumnos de 3 B:

- Castro Iturbide Luis Antonio
- Martinez Monarrez Alondra
- Moran Vázquez Miguel Ángel
- Román Hernández Martha Andrea
- Zaldivar Flores Mauro Omar

Docente:

- Ing. Dagoberto Fiscal Gurrola, M.T.I.

Victoria de Durango a domingo 14 de octubre del 2022

## Contenido

API's Base de Datos .....	3
¿Qué es? .....	3
API's .....	3
API de Base de Datos .....	3
Sintaxis .....	3
Métodos .....	3
Diagrama .....	5
Características .....	5
Ventajas .....	6
Desventajas .....	7
¿Por qué Implementarlo? .....	7
Implementación .....	8
Retroalimentación .....	11

## API's Base de Datos

¿Qué es?

API's

Una API es una aplicación programada para controlar el sistema manejador de la base de datos. Su implementación es conveniente ya que permite el trabajo con múltiples conexiones.

API de Base de Datos

Una API de Base de Datos, también conocida como DB-API (database API), conecta a una aplicación con una base de datos; la API es la ruta de comunicación entre ambos equipos. La API manda llamados HTTP al servidor en donde la base de datos está alojada, después, de acuerdo con la información insertada por el usuario, mostrará la información pedida proveniente de la base de datos, analizarla, y mostrarla al usuario en su aplicación.

Sin embargo, al momento de que se presente un error con SQL, la API no parará de ejecutarse al menos que se use la función `LastError()`.

Esta función no necesita de argumentos, lo único que hace es parar la API hasta donde ocurrió el error.

Sintaxis

**DB : : LastError()**

Métodos

Los métodos que hay en las API's son el CRUD, la conexión a la base de datos, la ejecución del query de la conexión y los datos.

- **SetConnection()**

Crea el enlace con la base de datos que se está usando, cuando se está trabajando con multiples conexiones de bases de datos.

- **Exec()**

Ejecuta un query SQL utilizando la conexión activa actual.

- **Query()**

Ejecuta un query SQL que regresa información.

- **LastId()**

Muestra el valor del ultimo campo autoincrementable que se insertó bajo la conexión actual.

- **LastError()**

Muestra el ultimo mensaje de error bajo la conexión actual.

- **insert()**

Añade registros a la base de datos.

- **Update()**

Actualiza registros en la base de datos.

- **Delete()**

Borra registros de la base de datos.

- **Select()**

Muestra datos de la base de datos.

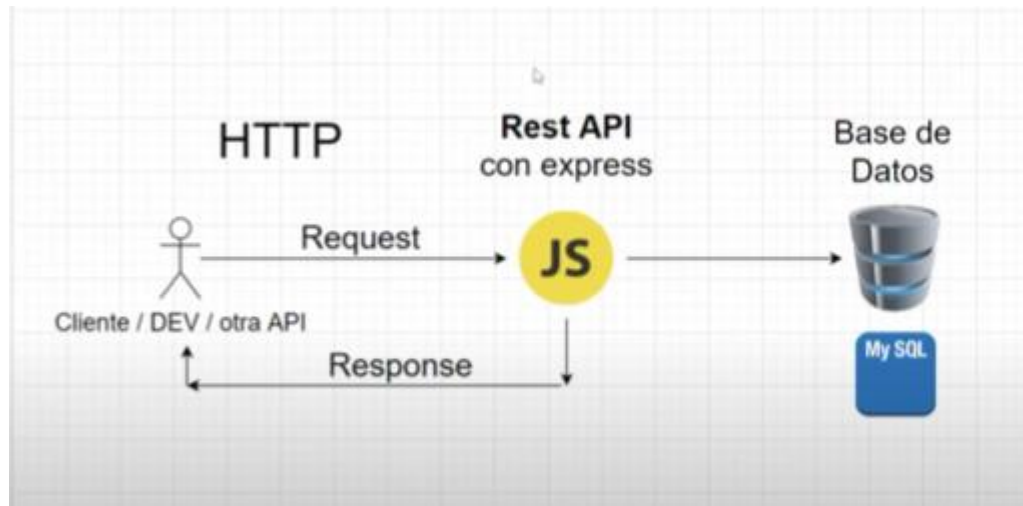
- **PrepareSQL()**

Prepara un query SQL a usar con variables SQL.

- **DBLookup()**

Ejecuta un query SQL pasado un parámetro. Muestra el primer valor de la primera entrada, o nula si no se encuentra nada.

## Diagrama



## Características

- **Sin estado**

Una de las principales características de una API REST es que su servicio no tiene estado, lo que significa que cada vez que nos referimos a ella, tenemos que recordarle nuestros datos, ya sean nuestras credenciales o cualquier otra información. Lo que, por un lado, podría parecer una desventaja, la tediosa tarea de repetir los datos, en realidad es uno de sus puntos fuertes, pues al no almacenarlos, permite mayor escalabilidad. No serán necesarios servidores tan potentes, capaces de almacenar todos los estados de sus clientes.

- **Soporte para JSON y XML**

Sirve para tanto para los que utilizan JSON como para los que confían en XML. Todo mundo tiene lugar en el mundo de las API's REST.

- **Es más sencillo que SOAP.**

Su aplicación es mucho más sencilla. Ejemplo: Salesforce, dispone de herramientas con ambas arquitecturas, pero permite que REST acceda a servicios potentes, cómodos y más sencillos para interactuar con él.

- **Documentación.**

Permite sincronizar actualizaciones para que se produzca de forma automática al cambiar detalles de la API.

- **Mensajes de error.**

Cuando se produce un error, se incluyen pistas al respecto.

## Ventajas

### 1.- Separación Cliente/Servidor

Al ser sistemas independientes (solo se comunican con un lenguaje de intercambio como JSON) puedes desarrollarlos proyectos autónomos, equipos autónomos. Al cliente le da igual cómo está hecha la API y al servidor le da igual qué vas a hacer con los datos que te han proporcionado.

### 2.- Independencia de tecnologías / lenguajes

Puedes desarrollar en cualquier tipo de tecnología o lenguaje con la que se sienta a gusto o con la que puedas acortar tus tiempos de desarrollo, o encaje con la filosofía o necesidades de tu proyecto.

### 3.- Fiabilidad, escalabilidad, flexibilidad

Al final solo te tienes que preocupar que el nexo cliente/servidor esté correcto. Puedes hacer cambios en tu servidor, lenguajes, bases de datos, etc., y mientras devuelvas los datos que toca todo irá correctamente.

#### 4. Experiencia de usuario

Aunque eso depende más de cómo está hecha la parte del cliente, teóricamente el desarrollo de sitios web basados en un API puede dar mejor desempeño que uno tradicional.

#### 5.- REST requiere menos recursos del servidor.

Esto no es necesariamente cierto aunque en muchos casos sí se pueda deducir. Hay muchas opiniones al rededor de REST. Nosotros basamos esa afirmación en estos motivos:

- No mantener el estado, no requiere memoria, se pueden atender más peticiones.
- No requiere escribir el HTML, por lo tanto tienes menos procesamiento en el servidor.

#### Desventajas

En las API REST no mantenemos estado y eso hace que tengas que montar una infraestructura propia para poder conservar el conjunto de la aplicación. Generalmente mandarás un token que indique quien eres al servidor y qué has realizado ya en tu aplicación.

Requiere más conocimientos, te obliga a salir de la zona de confort a la que quizás estemos. Aparte de conocer tus lenguajes, bases de datos, librerías, necesitarás aprender acerca del protocolo HTTP.

#### ¿Por qué Implementarlo?

Es independiente del lenguaje que se esté usando, mientras se siga teniendo las mismas operaciones y funciones en el API REST, no importa si en un futuro se decide cambiar la tecnología con la que se ha implementado la API REST, puede seguir funcionando sin ningún problema.

Tiene una gran flexibilidad que permite lograr hacer que los desarrolladores puedan ajustarse mejor a las necesidades de cada empresa.

El protocolo que se utiliza en esta API hace que la interfaz de usuario del servidor y el almacenamiento de datos funcionen de manera independiente.

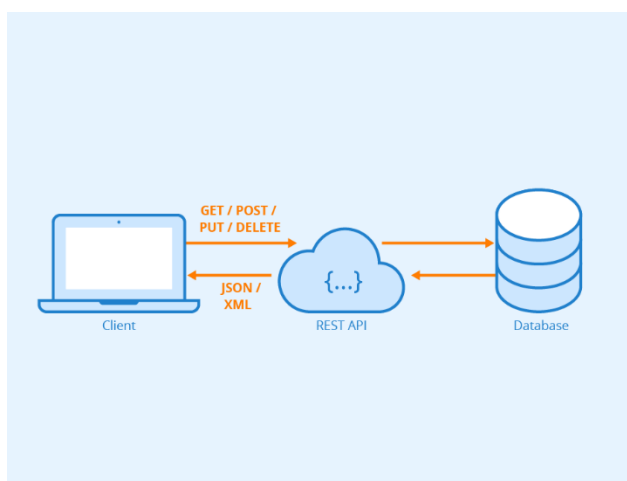
Las principales ventajas de la separación del cliente y el servidor es que mejora la portabilidad de la interfaz a otro tipo de plataformas, siempre y cuando se mantenga la interfaz de la API. Además de que esta separación permite que se puedan desarrollar distintos frontend en un único backend.

Reducen los costes de mantenimiento e incrementan la agilidad de los procesos de transformación de las empresas.

Mientras que SOAP se centra en el diseño de aplicaciones distribuidas, REST lo hace en la escalabilidad y en el rendimiento a gran escala para sistemas distribuidos hipermedia.

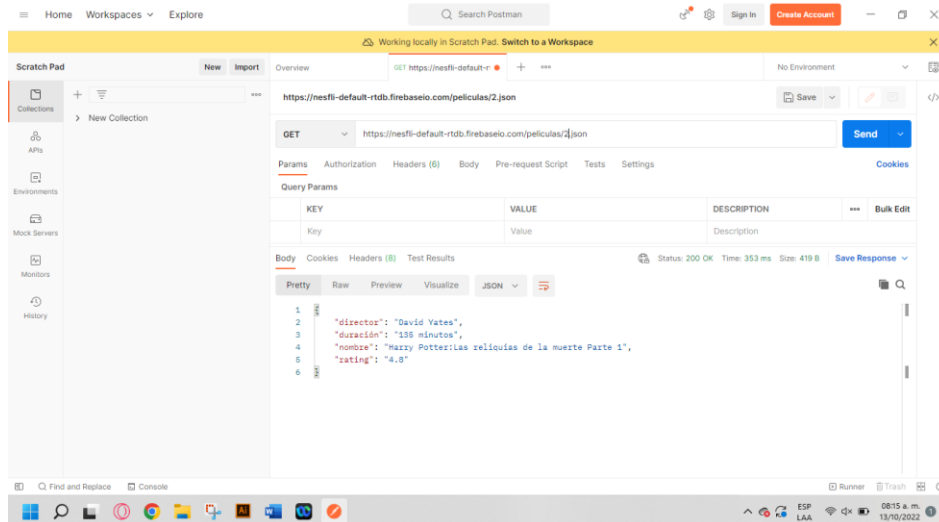
### Implementación

Puedes utilizar la API REST para crear y gestionar los patrones de base de datos, las instancias de base de datos, las imágenes de base de datos, las cargas de trabajo de base de datos, los tipos de patrón y los plugins del sistema.

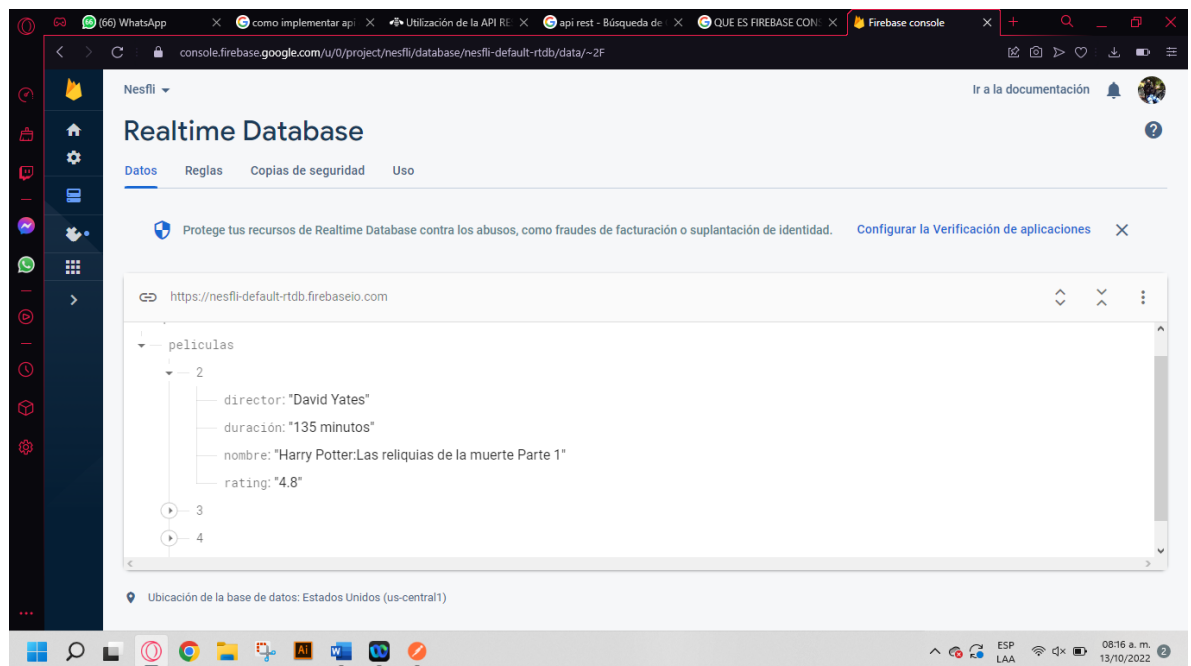




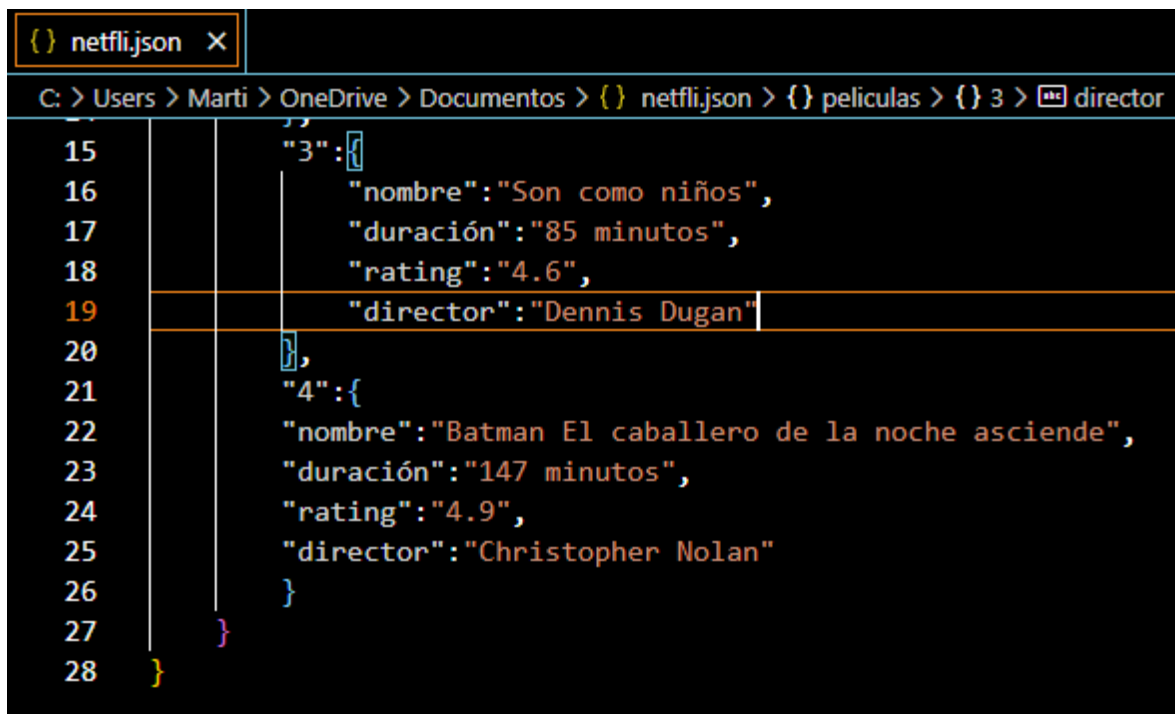
Se utilizaron los siguientes SW:



Postman es una plataforma de API para que los desarrolladores diseñen, construyan, prueben e iteren sus API.



Firebase se trata de una plataforma móvil creada por Google, cuya principal función es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida.



```
15  "3":{  
16    "nombre":"Son como niños",  
17    "duración":"85 minutos",  
18    "rating":"4.6",  
19    "director":"Dennis Dugan"  
20  },  
21  "4":{  
22    "nombre":"Batman El caballero de la noche asciende",  
23    "duración":"147 minutos",  
24    "rating":"4.9",  
25    "director":"Christopher Nolan"  
26  }  
27 }  
28 }
```

JSON, que significa JavaScript Object Notation, es una formatación usada para estructurar datos en forma de texto y transmitirlos de un sistema a otro, como en aplicaciones cliente-servidor o en aplicaciones móviles.

## Retroalimentación

Ha sido de gran importancia para nosotros el haber hecho esta investigación, ya que es de gran relevancia el que nosotros conozcamos los demás tipos de API que existen para base de datos, dichas APIs son de gran ayuda a nuestra integradora ya que se basa en lo que ejecuta la API REST, ya que esta puede consultar, insertar, modificar y eliminar registros de una base de datos, básicamente es muy parecido a lo que hace nuestra integradora y los buscadores que conocemos como lo es Google.

Las API's de base de datos permiten el procesamiento de datos, ya que todo se crea con la satisfacción de usuarios, por lo que al implementar la API REST este logra hacer el intercambio de datos satisfactorio hacia alguna base de datos a la cual esté conectada. Todo esto permita realizar un CRUD de la forma mas cómoda y rápida posible, ya que esta API no almacena la información en la nube sino en la base de datos.