



JS



PROFE VIDERMID

# Programación en Java Script

## (Parte 9)

Facilitador: Ing. Esp. Vidermid Sánchez



@vidermid



+584147106623



@ingenieriadigitalsc



+584147464801



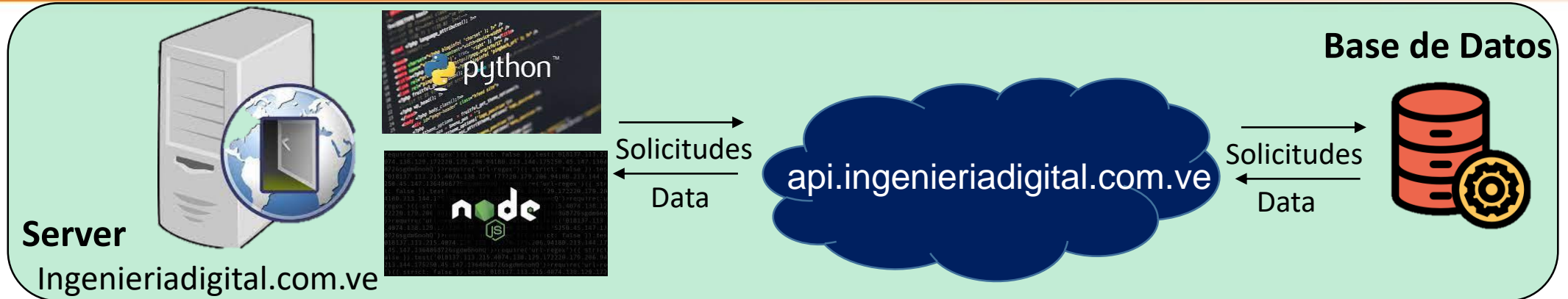
# JS

# Aplicación Web Full Stack



PROFE VIDERMID

Back - End



Script del lado del servidor  
**Server-Side Scripting (SSR)**

Peticiones  
Recarga  
Página



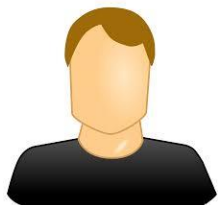
JSON / XML



GET / POST /  
PUT / DELETE

API  
REST

Front - End



Usuario



Las API (Interfaz de Programación de Aplicaciones), son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. En otras palabras, las API le permiten interactuar con una computadora o un sistema para obtener datos o ejecutar una función, de manera que el sistema comprenda la solicitud y la cumpla.

**Ejemplo:** En la Formula 1

El diseño de una API de la FIA podría requerir que el usuario escribiera un GP en específico y que el administrador de la F1 diera una respuesta en dos partes: la primera sería la fecha y lugar o pista y la segunda, las estadísticas y podios de carreras ya terminadas en otras temporadas. <http://ergast.com/api/f1/constructors/mclaren/circuits/monza/drivers>

```
myParser({
  "MRData": {
    "xmlns": "http://ergast.com/mrd/1.5",
    "series": "f1",
    "url": "http://ergast.com/api/f1/drivers.json",
    "limit": "30",
    "offset": "0",
    "total": "857",
    "DriverTable": {
      "Drivers": [
        {
          "driverId": "abate",
          "url": "http://en.wikipedia.org/wiki/Carlo_Mario_Abate",
          "givenName": "Carlo",
          "familyName": "Abate",
          "dateOfBirth": "1932-07-10",
          "nationality": "Italian"
        }
      ]
    }
  }
})
```

#### Query Details

Series	CircuitId	ConstructorId	Page	Results
f1	monza	mclaren	1 of 2	39

#### Driver Table

Driver Name	Permanent Number	Nationality	DOB	Information
Fernando Alonso	14	Spanish	1981-07-29	<a href="#">Biography</a>
Michael Andretti		American	1962-10-05	<a href="#">Biography</a>
Gerhard Berger		Austrian	1959-08-27	<a href="#">Biography</a>
Mark Blundell		British	1966-04-08	<a href="#">Biography</a>
Jo Bonnier		Swedish	1930-01-31	<a href="#">Biography</a>
Martin Brundle		British	1959-06-01	<a href="#">Biography</a>
Jenson Button	22	British	1980-01-19	<a href="#">Biography</a>
David Coulthard		British	1971-03-27	<a href="#">Biography</a>



REST (Representational State Transfer- Transferencia de Estado Representacional), REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON.

En la actualidad no existe proyecto o aplicación que no disponga de una API REST para la creación de servicios profesionales a partir de ese software. Twitter, YouTube, los sistemas de identificación con Facebook... hay cientos de empresas que generan negocio gracias a REST y las APIs REST. Sin ellas, todo el crecimiento en horizontal sería prácticamente imposible. Esto es así porque REST es el estándar más lógico, eficiente y habitual en la creación de APIs para servicios de Internet.

Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).

Los objetos en REST siempre se manipulan a partir de la URI (Identificador Uniforme de Recursos). Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI nos facilita acceder a la información mediante una dirección URL para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.



# Consumo de API - REST

En internet existen API-REST publicas y privadas, además con acceso libre o mediante autenticación (api-key), en *github* existe un gran repositorio que contiene infinidad de API-REST con su respectiva documentación que ayuda a su consumo correcto.

<https://github.com/public-apis/public-apis>

Ejemplo de api: <https://dog.ceo/api/breeds/image/random>

```
→ ↻ 🔒 dog.ceo/api/breeds/image/random

// 20230617214552
// https://dog.ceo/api/breeds/image/random

{
  "message": "https://images.dog.ceo/breeds/hound-walker/n02089867_2365.jpg",
  "status": "success"
}
```





# Consumo de la API-REST

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="shortcut icon" href="#" type="image/x-icon">
  <title>Perros desde una API</title>
</head>
<body>
  <h1>Perros desde una API</h1>
  <center><img src="" alt="Un Perrito" height="300" width="300"></center>
  <script src="api.js"></script>
</body>
</html>
```

HTML



# Consumo de la API-REST

```
console.log('Estoy en el JS de API');  
let rutaApi='https://dog.ceo/api/breeds/image/random';  
  
fetch(rutaApi)  
.then(respu => respu.json())  
.then(datos =>{  
  console.log(datos.message);  
  const matacho= document.querySelector('img');  
  matacho.src=datos.message;  
})  
.catch(error =>{console.error('Atencion: '+error)})
```

JS

