

**PROGRAMA DE ESTUDIOS**

# **DESARROLLOS DE SISTEMAS DE INFORMACIÓN**

**DESARROLLO FRONTEND  
DE SISTEMAS DE WEB**

**Tema**

**MANEJO DE RESPUESTAS JSON  
CON JQUERY**

## Manejo de Respuestas JSON con jQuery

### Objetivo de Aprendizaje

Comprender cómo jQuery permite manejar respuestas en formato JSON a través de peticiones AJAX, aprendiendo a interpretar, manipular y mostrar datos dinámicamente en una página web.

### Parte Conceptual (20 min)

#### 🔍 ¿Qué es JSON?

**JSON (JavaScript Object Notation)** es un formato de texto que representa datos estructurados basados en la sintaxis de objetos de JavaScript.

#### 📌 Características clave:

- Formato de **clave-valor**.
- Compatible con **JavaScript**.
- Utilizado en la mayoría de APIs REST.
- Fácil de transmitir entre cliente y servidor.

```
{  
  "nombre": "Juan",  
  "edad": 25,  
  "habilidades": ["HTML", "CSS", "JavaScript"],  
  "direccion": {  
    "ciudad": "Lima",  
    "pais": "Perú"  
  }  
}
```

¿Por qué usar JSON en lugar de XML?

JSON	XML
Más liviano	Más verboso

---

Más fácil de leer	Más estructurado pero complejo
Compatible con objetos JavaScript	Necesita procesamiento adicional

---

Característica	JSON	XML
Legibilidad	Alta	Media
Tamaño	Más pequeño	Más grande
Estructura	Objetos y arrays	Etiquetas jerárquicas
Procesamiento	Natural con JS (JSON.parse)	Requiere parser XML

---

## Manejo de JSON con jQuery

### Métodos:

- `.getJSON(url, callback)` – método simplificado para recibir respuestas JSON.
- `$.ajax({ url, dataType: "json" })` – configuración más detallada y flexible.

### Conversión:

- `JSON.parse()` – Convierte una cadena JSON a un objeto JS.
- `JSON.stringify()` – Convierte un objeto JS a cadena JSON.

## Parte Demostrativa (20 min)

### 💡 Ejemplo 1: Mostrar JSON en consola y página

```
javascript
CopiarEditar
$.getJSON("https://jsonplaceholder.typicode.com/users", function(data)
{
    console.log(data); // Muestra el JSON completo

    data.forEach(user => {
```

```
$("#usuarios").append(`<li>${user.name} - ${user.email}</li>`);  
});  
});
```

HTML:

```
html  
CopiarEditar  
<ul id="usuarios"></ul>
```

### 🔗 Ejemplo 2: Petición con \$.ajax

```
javascript  
CopiarEditar  
$.ajax({  
    url: "https://jsonplaceholder.typicode.com/users",  
    dataType: "json",  
    success: function(respuesta) {  
        respuesta.forEach(usuario => {  
            $("#contenedor").append(`<div>${usuario.name} -  
${usuario.phone}</div>`);  
        });  
    }  
});
```

## ✍ Parte Práctica (20 min)

### 🛠 Actividad práctica:

Implementa una página que:

- Use .getJSON() o \$.ajax() para obtener datos de una API pública.

- Muestre los datos en una **tabla dinámica**.
- Incorpore un campo de búsqueda para **filtrar por nombre**.

**Sugerencia:**

- API: <https://jsonplaceholder.typicode.com/users>

**Pista:**

javascript

CopiarEditar

```
$("#buscar").on("keyup", function() {  
    const filtro = $(this).val().toLowerCase();  
    $("table tr").filter(function() {  
        $(this).toggle($(this).text().toLowerCase().indexOf(filtro) > -1)  
    });  
});
```

## Cierre de sesión (15 min)

💬 **Reflexión:**

¿Cómo mejora el uso de JSON la comunicación entre cliente y servidor?

Discusión guiada sobre:

- La portabilidad de JSON.
- Su integración nativa con JavaScript.
- La eficiencia en el consumo de APIs REST.

📘 **Tarea para casa:**

Crear una página que cargue dinámicamente datos desde una API y los muestre como tarjetas (div), incluyendo nombre, email y teléfono del usuario.

