

# UNIVERSIDAD DEL ISTMO



## Programación Web II

Ingeniería en Computación

---

**Utilizar AJAX y JSON para crear una aplicación  
de varias páginas**

**Profesor: *Omar Nieva García***

---

Joan Pablo Alvarado Garfias

Grado: Sexto Semestre

Parcial: Primer parcial

# Archivos JSON y peticiones HTTP con AJAX

## Objetivos

- Utilizar AJAX y JSON para crear una aplicación de varias páginas

---

## Estructura del Proyecto

```
json_2
├── index.html
├── producto.html
├── css/ (carpeta de hojas de estilo)
├── js/ (carpeta de códigos javascript)
├── storage/ (carpeta con el archivo json de los productos)
└── img/ (carpeta de imágenes)
```

## Tecnologías Utilizadas

Tecnología	Descripción
HTML5	Estructura de la página web.
CSS3	Estilo de la página web.
JavaScript	Manipulación del DOM y AJAX.
JSON	Formato para intercambio de datos.

## Configuración del Proyecto

Para este proyecto se necesita un servidor local debido al manejo de peticiones con AJAX. Por lo que se recomienda utilizar algún software como Xampp, Wampp, extensión de VSC (live server), entre otras.

## Maquetado de la página de inicio

Lo primero que hice fue maquetar la página de inicio con html y de esta forma ir dandome una idea de cómo puedo estructurar los módulos necesarios para el funcionamiento.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/...>
  <link rel="stylesheet" href="css/menu.css">
  <link rel="stylesheet" href="css/products.css">
```

```

    <link rel="stylesheet" href="css/footer.css">
    <title>Comercio IT</title>
</head>
<body>
    <nav>
        <h1 class="title">Comercio<span class="title-bold">IT</span></h1>
        <ul class="options">
            <li><i class="fa-solid fa-lock"></i></li>
            <li><a href="#">INGRESAR</a></li>
            <li> | </li>
            <li><a href="#">REGISTRARME</a></li>
            <li> | </li>
            <li><a href="#">CONTACTO</a></li>
        </ul>
    </nav>
    <main>
        <section class="products">
        </section>
    </main>
    <footer>
        <p class="creditos">Desarrollado por Joan Garfias</p>
    </footer>
    <script src="js/leerProductos.js"></script>
    <script src="js/obtenerProductos.js"></script>
    <script src="js/crearProductos.js"></script>
</body>
</html>

```

## Creación del Archivo JSON

El archivo datos.json simula la base de datos con los productos.

```

{
  "datos": [
    {
      "id": 1,
      "titulo": "iPhone X",
      "descripcion": "Último modelo de gama alta, de la prestigiosa marca de Apple. Cu",
      "imagen": "P001.jpg",
      "precio": 10000.00,
      "stock": 120
    },
    {
      "id": 2,
      "titulo": "iPad",
      "descripcion": "El iPad PRO es ideal para llevar archivos y documentos a cualquier"
    }
  ]
}

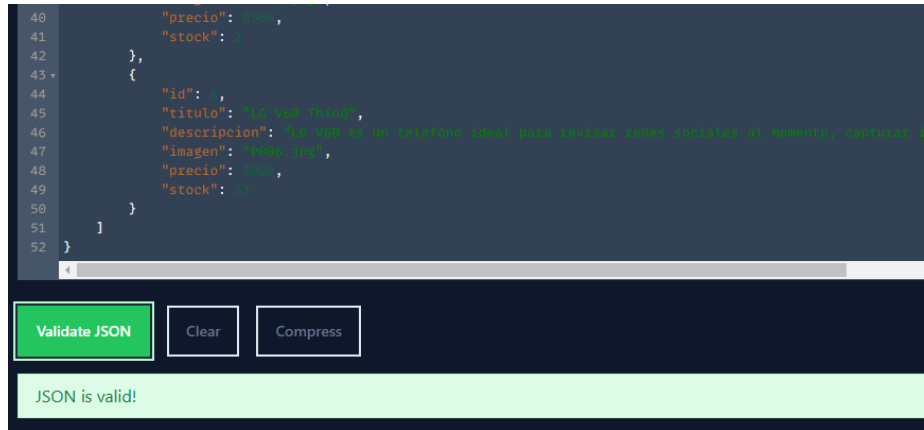
```

```

        "imagen": "P002.jpg",
        "precio": 7000.00,
        "stock": 10
    },
    {
        "id": 3,
        "titulo": "Nexus xD101",
        "descripcion": "Celular Nexus de amplia pantalla y con sistema operativo Android",
        "imagen": "P003.jpg",
        "precio": 5030.00,
        "stock": 51
    },
    {
        "id": 4,
        "titulo": "Samsung Galaxy",
        "descripcion": "El Samsung Galaxy es un móvil con la tecnología 5G de última generación",
        "imagen": "P004.jpg",
        "precio": 9000.00,
        "stock": 45
    },
    {
        "id": 5,
        "titulo": "Pixel 4",
        "descripcion": "El Pixel 4 es el móvil de Google con una de las mejores cámaras",
        "imagen": "P005.jpg",
        "precio": 8500.00,
        "stock": 2
    },
    {
        "id": 6,
        "titulo": "LG V60 ThinQ",
        "descripcion": "LG V60 es un teléfono ideal para revisar redes sociales al momento",
        "imagen": "P006.jpg",
        "precio": 3000.00,
        "stock": 33
    }
]
}

```

## Validar sintaxis del archivo JSON



> **Nota:** La sintaxis fue validada con jsonlint

## Obtener los datos del json desde JavaScript con AJAX

Aquí se muestra cómo hacer una petición AJAX para obtener los datos del JSON y mostrarlos en la página.

```
const CREAR_PRODUCTOS = 998;
const CARGAR_PRODUCTO = 999;

var productos;
var request = new XMLHttpRequest();
let peticion;

request.open('GET', '../storage/datos.json', true);

request.onreadystatechange = function() {
  if (request.readyState === 4) {
    if (request.status == 200) {
      productos = JSON.parse(request.responseText);
      switch(peticion){
        case CREAR_PRODUCTOS:
          crearProductos();
          break;
        case CARGAR_PRODUCTO:
          buscarPorId();
          break;
      }
    } else {
      console.log("Ha ocurrido un error al enviar la peticion");
      estado = false;
    }
  }
}
```

```

    }
  }
}

```

## Mostrar los productos recolectados

Una vez que tenemos los datos de los productos tenemos que mostrar la información, el archivo *crearProductos.js* contiene la función *crearProductos()* y *crearProducto()* las cuales funcionan para la página *index.html* . Sin embargo, la función *crearProductoIndividual()* es una función que se requiere en el *producto.html*.

```

const catalogo = document.querySelector(".products");
function crearProductos(){
  productos.datos.forEach(prod => {
    crearProducto(prod);
  });
}

function crearProducto(p){
  const img = document.createElement("img");
  const titulo = document.createElement("h2");
  const descripcion = document.createElement("p");
  const precio = document.createElement("p");

  img.src = `img/${p.imagen}`; //Con esto podemos mostrar la imagen
  titulo.innerHTML = `

```

**Nota:** La función *crearProductoIndividual* es muy parecida a *crearProducto*, la diferencia es que *crearProductoIndividual* no crea solamente las etiquetas para el html, sino que crea divisiones para el html y agrupa algunas etiquetas nuevas que no tiene *index.html*. Por ejemplo: el botón de compra y el stock.

## Resultados Esperados

- Cuando se haga clic en el nombre de un producto desde *index.html*, debe cargar los datos del producto en una página nueva con la estructura *producto.html#ID*.
- Ejemplo de salida esperada:



Figure 1: image

•

## Conclusión

En esta práctica aprendí: - Crear un archivo JSON con su correcta estructura. - Hacer peticiones AJAX con XMLHttpRequest. - Conocer los estados y funciones necesarias para las peticiones AJAX con XMLHttpRequest. - Cargar dinámicamente datos desde un archivo JSON. - Crear etiquetas nuevas y estilarlas desde JavaScript.

## Referencias

- Trabajando con JSON - mdn web docs
- Document.createElement() - mdn web docs
- XMLHttpRequest - W3 Schools

## Repositorio

Se puede acceder al repositorio para analizar el código más a detalle en la siguiente liga. Repositorio de GitHub