

# LENGUAJES DE PROGRAMACIÓN

## JAVA SCRIPT





## Gestión de formularios

La gestión de formularios es una parte fundamental en el desarrollo web, ya que permite la interacción y recopilación de datos por parte de los usuarios. En este contexto, el lenguaje de programación JavaScript juega un papel crucial al proporcionar las herramientas necesarias para manipular y validar los datos ingresados en los formularios.

En esta lección introductoria, exploraremos los conceptos básicos de la gestión de formularios en JavaScript, centrándonos en cómo interactuar con los elementos del formulario, acceder a sus valores y realizar acciones en respuesta a eventos desencadenados por el usuario. A lo largo de esta lección, utilizaremos ejemplos y demostraciones para ilustrar los conceptos discutidos.

# { JavaScript }



## Interacción con los elementos del formulario:

Para empezar, es fundamental entender cómo interactuar con los elementos individuales de un formulario. En JavaScript, podemos acceder a estos elementos utilizando el objeto document y sus métodos para la manipulación del DOM (Document Object Model).

**Por ejemplo**, supongamos que tenemos un formulario con un campo de entrada de texto para el nombre del usuario:

```
<form id="miFormulario">

  <label for="nombre">Nombre:</label>

      <input      type="text"      id="nombre"
name="nombre">

  <button type="submit">Enviar</button>

</form>
```

# LENGUAJES DE PROGRAMACIÓN

## JAVA SCRIPT

Podemos acceder al campo de entrada de texto utilizando el método `getElementById` y manipular su contenido de la siguiente manera:

```
// Accediendo al campo de entrada de texto
const campoNombre =
document.getElementById('nombre');

// Obteniendo el valor actual del campo
const valorNombre = campoNombre.value;

// Modificando el valor del campo
campoNombre.value = 'Nuevo nombre';
```

# {.js}

# JavaScript

# LENGUAJES DE PROGRAMACIÓN

## JAVA SCRIPT

### Validación de formularios:

La validación de formularios es un paso crítico para garantizar que los datos ingresados sean correctos y cumplan con los requisitos establecidos. JavaScript proporciona diversas técnicas para validar los formularios antes de enviarlos al servidor.

Una forma común de validar los formularios es utilizar la propiedad `validity` del objeto del campo de entrada. Por ejemplo, podemos validar si un campo de entrada es obligatorio y ha sido llenado:

```
const campoNombre = document.getElementById('nombre');

if (campoNombre.validity.valueMissing) {
  // El campo está vacío, mostrar mensaje de error
  mostrarError('Por favor, ingresa tu nombre');
} else {
  // El campo no está vacío, realizar acciones
  adicionales
  // ...
}
```

# LENGUAJES DE PROGRAMACIÓN

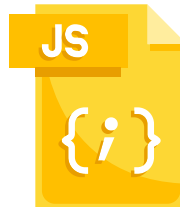
## JAVA SCRIPT



También podemos utilizar expresiones regulares para validar campos de entrada que siguen un patrón específico, como direcciones de correo electrónico o números de teléfono. Por ejemplo, podemos validar un campo de correo electrónico de la siguiente manera:

```
const campoEmail = document.getElementById('email');
const emailRegex = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*
(\\.\\w{2,3})+$/;

if (!emailRegex.test(campoEmail.value)) {
  // El formato del correo electrónico es inválido,
  mostrar mensaje de error
  mostrarError('Por favor, ingresa un correo
  electrónico válido');
} else {
  // El formato del correo electrónico es válido,
  realizar acciones adicionales
  // ...
}
```





## Eventos en formularios:

JavaScript nos permite detectar y responder a eventos desencadenados por los usuarios en formularios. Estos eventos incluyen el envío del formulario, cambios en los campos de entrada, clics en botones, entre otros.

Podemos utilizar el método `addEventListener` para asociar una función de controlador de eventos a un evento específico. Por ejemplo, si queremos realizar una acción cuando se envía el formulario, podemos hacer lo siguiente:

```
const formulario = document.getElementById('miFormulario');

formulario.addEventListener('submit',
function(event) {
    // Evitamos el envío del formulario por defecto
    event.preventDefault();

    // Realizamos acciones adicionales
    // ...
});
```



En esta introducción a la gestión de formularios en JavaScript, hemos explorado cómo interactuar con los elementos del formulario, validar los datos ingresados y responder a eventos desencadenados por el usuario. Estos conceptos son fundamentales para el desarrollo de aplicaciones web interactivas y permiten una experiencia de usuario más robusta.

Es importante recordar que la gestión de formularios es solo el comienzo, y hay muchas técnicas y consideraciones avanzadas que se pueden aplicar en situaciones más complejas. A medida que avancemos en nuestro aprendizaje, exploraremos temas adicionales, como el envío de formularios mediante AJAX, la validación del lado del servidor y la manipulación dinámica de los elementos del formulario.

