



# EVENTOS EN JAVASCRIPT

MATEO ALTAMIRANO

PROGRAMACION INTEGRATIVA DE COMPONENTES WEB

NRC: 16496

# INTRODUCCION

- ▶ Aplicativo web que maneja eventos de teclado, mouse, formulario y DOM.
- ▶ Uso de HTML, CSS y JavaScript para la implementación
- ▶ Interactividad y dinámica en la página web.

# Estructura del Proyecto

- ▶ HTML: Estructura y contenido de la página.
- ▶ CSS: Estilos y diseño visual.
- ▶ JavaScript: Manejo de eventos y lógica de la aplicación.

# EXPLICACION

```
const mouseButton = document.getElementById("mouseButton");
const mouseOutput = document.getElementById("mouseOutput");

mouseButton.addEventListener("click", () => {
  mouseOutput.textContent = "Hiciste clic en el Mouse ";
});

const keyboardInput = document.getElementById("keyboardInput");
const keyboardOutput = document.getElementById("keyboardOutput");

keyboardInput.addEventListener("keyup", (event) => {
  keyboardOutput.textContent = `Tecla presionada : ${event.key}`;
});

const sampleForm = document.getElementById("sampleForm");
const formOutput = document.getElementById("formOutput");

sampleForm.addEventListener("submit", (event) => {
  event.preventDefault();
  const name = document.getElementById("nameInput").value;
  formOutput.textContent = `Formulario subido por el nombre de : ${name}`;
});

const domButton = document.getElementById("domButton");
const domContainer = document.getElementById("domContainer");

domButton.addEventListener("click", () => {
  const newElement = document.createElement("p");
  newElement.textContent = "Nuevo elemento al DOM";
  domContainer.appendChild(newElement);
});
```

- ▶ Mouse Event: Muestra un mensaje al hacer clic en el botón.
- ▶ Keyboard Event: Muestra la tecla presionada en el campo de texto.
- ▶ Form Event: Muestra el valor del formulario al enviarlo.
- ▶ DOM Event: Añade un nuevo elemento al DOM al hacer clic en el botón.

# CONCLUSIONES

- ▶ Los eventos permiten crear aplicaciones web interactivas y dinámicas.
- ▶ Mejoran la experiencia del usuario al proporcionar respuestas inmediatas a sus acciones.
- ▶ JavaScript ofrece una amplia variedad de eventos: mouse, teclado, formularios, eventos del DOM, entre otros.
- ▶ La comprensión de los eventos permite ampliar y personalizar las aplicaciones según las necesidades específicas del proyecto.