

**PROGRAMA DE ESTUDIOS**

# **DESARROLLO DE SISTEMAS DE INFORMACIÓN**

**COMMUNITY MANAGER Y  
MARKETING DIGITAL**

**Tema**

**EVENTOS EN JQUERY: .CLICK(),  
.SUBMIT(), .HOVER()**

## Eventos en jQuery: .click(), .submit(), .hover()

### 1. Introducción a los eventos en jQuery

jQuery es una biblioteca de JavaScript que simplifica la manipulación del **DOM** y la gestión de eventos. Permite escribir código más conciso y compatible con distintos navegadores.

Los **eventos** en jQuery permiten responder a acciones del usuario, como hacer clic en un botón, enviar un formulario o pasar el cursor sobre un elemento.

### 2. Diferencias entre manejar eventos con JavaScript puro y con jQuery

En JavaScript puro, manejar eventos requiere usar `addEventListener()`, mientras que en jQuery se utilizan métodos más simples como `.click()`, `.submit()` y `.hover()`.

👉 Ejemplo con JavaScript puro:

```
document.getElementById("miBoton").addEventListener("click",
function()
    alert("Hiciste clic en el botón");
});
```

👉 Ejemplo con jQuery:

```
$("#miBoton").click(function()
    alert("Hiciste clic en el botón");
});
```

- ✓ jQuery permite escribir menos código y hace que sea más fácil manejar eventos en múltiples elementos sin necesidad de bucles manuales.

### 3. Métodos principales para la gestión de eventos

#### 3.1 .click(): Ejecutar una acción cuando se hace clic en un elemento

El método `.click()` se usa para ejecutar una función cuando el usuario hace clic en un botón, un enlace u otro elemento interactivo.

❖ **Ejemplo:**

```
$("#boton").click(function()
    alert("Botón
});
```

clickeado");

#### 3.2 .submit(): Capturar el envío de un formulario y validararlo

Se usa para interceptar el envío de un formulario y ejecutar acciones, como validar campos antes de enviarlo al servidor.

❖ **Ejemplo:**

```
$("#miFormulario").submit(function(event)
    event.preventDefault(); // Evita el envío automático
    alert("Formulario
});
```

enviado

correctamente.");

#### 3.3 .hover(): Detectar cuando el usuario pasa el cursor sobre un elemento

El método `.hover()` permite ejecutar dos funciones:

- **Cuando el cursor entra** en el elemento.
- **Cuando el cursor sale** del elemento.

❖ Ejemplo:

```
$("#caja").hover(  
    function() { $(this).css("background-color", "yellow"); }, //  
    Cuando entrara  
    function() { $(this).css("background-color", "white"); } // Cuando  
    sale  
)
```

## 4. Demostración práctica

### 4.1 Implementación de un botón que cambia de color al hacer clic

❖ HTML:

```
<button id="boton">Haz clic</button>
```

❖ jQuery:

```
$("#boton").click(function()  
    $(this).css("background-color",  
}); {  
    "red");
```

### 4.2 Creación de un formulario con validación al enviar (.submit())

❖ HTML:

```
<form id="miFormulario">  
    <input type="text" id="nombre" placeholder="Ingresa tu nombre">  
    <button type="submit">Enviar</button>  
</form>
```

```
<p id="mensaje"></p>
```

 **jQuery:**

```
$("#miFormulario").submit(function(event) {
    event.preventDefault();
    let nombre = $("#nombre").val();
    if (nombre.trim() === "") {
        $("#mensaje").text("Por favor, ingresa tu nombre").css("color", "red");
    } else {
        $("#mensaje").text("Hola, " + nombre).css("color", "green");
    }
});
```

### 4.3 Uso de .hover() para cambiar el color de un elemento al pasar el mouse

 **HTML:**

```
<div id="caja" style="width: 100px; height: 100px; background-color: lightgray;"></div>
```

 **jQuery:**

```
$("#caja").hover(
    function() { $(this).css("background-color", "blue"); }, // Cuando entra
    function() { $(this).css("background-color", "lightgray"); } // Cuando sale
);
```

## Conclusión

- jQuery simplifica la gestión de eventos en comparación con JavaScript puro.
- .click(), .submit() y .hover() permiten capturar interacciones del usuario de forma sencilla.
- Se pueden modificar estilos y contenido dinámicamente usando eventos en jQuery.



