

**PROGRAMA DE ESTUDIOS**

# **DESARROLLOS DE SISTEMAS DE INFORMACIÓN**

**DESARROLLO FRONTEND  
DE SISTEMAS DE WEB**

**Tema**

**ANIMACIONES EN JQUERY:  
.ANIMATE()**

## Animaciones en jQuery: .animate()

### 1. ¿Qué es .animate() y cómo funciona?

El método `.animate()` en jQuery permite crear animaciones personalizadas sobre propiedades CSS que sean numéricas. A diferencia de otros métodos como `.fadeIn()` o `.slideUp()`, `.animate()` brinda mayor control sobre los efectos al permitir la modificación progresiva de múltiples propiedades simultáneamente.

#### Sintaxis básica:

```
$(elemento).animate({ propiedadesCSS }, duración, easing, callback);
```

#### Parámetros:

- propiedadesCSS:** Define los estilos a animar (por ejemplo, `width`, `height`, `opacity`, `left`).
- duración (opcional):** Tiempo en milisegundos o valores como "slow" o "fast".
- easing (opcional):** Define el tipo de aceleración ("swing" por defecto o "linear").
- callback (opcional):** Función a ejecutar una vez terminada la animación.

### 2. Diferencia entre .animate() y otros métodos de animación

Método	Descripción
<code>.fadeIn()</code>	Cambia la opacidad del elemento, haciéndolo aparecer progresivamente.
<code>.fadeOut()</code>	Reduce la opacidad hasta hacer desaparecer el elemento.
<code>.slideUp()</code>	Oculta el elemento deslizándolo hacia arriba.
<code>.slideDown()</code>	Muestra el elemento deslizándolo hacia abajo.
<code>.animate()</code>	Permite personalizar animaciones cambiando varias propiedades CSS de forma progresiva.

#### Ejemplo comparativo:

##### Código JavaScript

```
$("#caja").fadeOut(1000); // Se desvanece en 1 segundo  
$("#caja").slideUp(1000); // Se oculta con un deslizamiento
```

```
// Se ensancha y cambia opacidad en 1 segundo
$("#caja").animate({ width: "300px", opacity: 0.5 }, 1000);
```

### 3. Parámetros básicos de .animate()

#### 3.1 Propiedades CSS animables

Algunas de las propiedades que pueden animarse son:

- width, height, opacity
- left, right, top, bottom (posición relativa)
- margin, padding, fontSize, entre otras.

##### Ejemplo:

Código JavaScript

```
$("#caja").animate(
{ width: "300px", height: "150px", opacity: 0.7 }, 1500);
```

#### 3.2 Duración

El tiempo de animación puede definirse en:

- Milisegundos (1000 equivale a 1 segundo).
- Valores predefinidos: "slow", "fast".

##### Ejemplo:

Código JavaScript

```
$("#caja").animate({ width: "250px" }, "slow");
$("#caja").animate({ height: "100px" }, 2000);
```

#### 3.3 Easing: Modificar la velocidad de animación

El parámetro easing define la manera en que la animación progresá.

- "linear": Movimiento uniforme.
- "swing": Movimiento acelerado/desacelerado.

**Ejemplo:**

Código JavaScript

```
$("#caja").animate({ width: "300px" }, 1000, "swing");
$("#caja").animate({ height: "150px" }, 1000, "linear");
```

### 3.4 Callback: Acción al finalizar la animación

Se puede ejecutar una función después de que termine la animación.

**Ejemplo:**

Código JavaScript

```
$("#caja").animate({ width: "300px" }, 1000, function() {
    alert("Animación terminada");
});
```

## 4. Demostración práctica

### 4.1 Implementación de una animación simple con .animate()

**Ejemplo:** Un botón que cambia el tamaño de un div.

html

```
<div id="caja" style="width:100px; height:100px; background:red;">
</div>
<button id="animar">Animar</button>
```

jQuery:

Código JavaScript

```
$("#animar").click(function() {
    $("#caja").animate({ width: "200px", height: "200px", opacity: 0.5
    }, 1000);
});
```

#### 4.2 Creación de una caja que se mueva de izquierda a derecha con cambios en tamaño y opacidad

**Ejemplo:**

html

```
<div id="caja" style="width:100px; height:100px; background:blue;  
position:absolute;"></div>  
<button id="mover">Mover</button>
```

**jQuery:**

Código JavaScript

```
$("#mover").click(function() {  
    $("#caja").animate({ left: "300px", width: "150px", opacity: 0.5  
},  
1500);  
});
```

#### 4.3 Uso de colas de animación y encadenamiento de efectos

jQuery permite encadenar múltiples .animate() para crear secuencias de animaciones.

**Ejemplo:**

Código JavaScript

```
$("#caja").animate({ left: "200px" }, 1000)  
    .animate({ top: "100px" }, 1000)  
    .animate({ width: "250px", height: "150px" }, 1000)  
    .animate({ opacity: 0.5 }, 1000);
```

Esto genera un movimiento en secuencia:

- Mover la caja 200px a la derecha.
- Moverla 100px hacia abajo.
- Aumentar su tamaño.
- Reducir la opacidad.

## 5. Actividad práctica individual

**Objetivo:** Crear un script que utilice .animate() para realizar efectos visuales personalizados.

- Crea una caja que aumente su tamaño y cambie de color al hacer clic en un botón.
- Implementa un div que se desplace a diferentes posiciones con animaciones encadenadas.
- Añade un callback que muestre un mensaje cuando la animación termine.

**Código base:**

```
html


</div>
<button id="iniciarAnimacion">Iniciar Animación</button>

<script>
$(document).ready(function() {
    $("#iniciarAnimacion").click(function() {
        $("#caja").animate({ left: "250px", width: "200px", height: "200px" }, 1000)
            .animate({ top: "150px", backgroundColor: "red" }, 1000)
            .animate({ opacity: 0.5 }, 1000, function() {
                alert("¡Animación completa!");
            });
    });
});
</script>


```

Este ejercicio permitirá a los estudiantes explorar y practicar el uso de .animate() en jQuery, aplicándolo para crear efectos personalizados en una página web.

