

PROGRAMA DE ESTUDIOS

DESARROLLOS DE SISTEMAS DE INFORMACIÓN

**DESARROLLO FRONTEND
DE SISTEMAS DE WEB**

Tema

**CREACIÓN DE INTERFACES
RESPONSIVE CON CSS Y
JQUERY**

Creación de Interfaces Responsive con CSS y jQuery

Objetivo:

Dominar el uso avanzado de eventos en JavaScript, como la **delegación**, la **propagación**, la **prevención de comportamiento por defecto** y su uso eficiente para mejorar el rendimiento de Aplicar técnicas de diseño responsive usando **CSS y jQuery**, adaptando interfaces web a distintos tamaños de pantalla para mejorar la experiencia del usuario en **dispositivos móviles y de escritorio**.

PARTE CONCEPTUAL (20 minutos)

Pregunta detonadora:

¿Por qué es importante que una página web sea responsive?

Ideas clave:

- Cada vez más usuarios acceden desde móviles o tablets.
- Un sitio no adaptado genera mala experiencia y abandono.
- Mejora la accesibilidad y el SEO.

Conceptos esenciales

Diseño adaptable (Responsive Design)

La estructura del sitio se **ajusta automáticamente** al tamaño de pantalla (flexibilidad, no versiones separadas).

Herramientas CSS:

- **Media queries** para aplicar estilos según el ancho de pantalla.
- **Flexbox y Grid** para layouts fluidos.

- **Unidades relativas:** %, em, rem, vh, vw.

⌚ Media Query básica:

css

```
@media (max-width: 768px) {
```

```
    .menu {  
        display: block;  
    }  
}
```

⌚ **Viewport:**

html

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

⌚ ¿Dónde entra jQuery?

Con jQuery se pueden:

- Detectar cambios de tamaño:

javascript

```
$(window).resize(function() {  
    console.log("Pantalla redimensionada");  
});
```

- Modificar clases o estilos para adaptar componentes.

- Mostrar u ocultar elementos como menús móviles.



PARTE DEMOSTRATIVA (20 minutos)



Ejemplo: Estructura responsive básica

HTML:

```
<header class="menu">  
  <nav>  
    <ul id="menu-desktop">  
      <li>Inicio</li>  
      <li>Servicios</li>  
      <li>Contacto</li>  
    </ul>  
    <div id="menu-mobile">☰</div>  
  </nav>  
</header>
```

CSS:

```
#menu-desktop {  
  display: flex;  
}  
#menu-mobile {  
  display: none;  
}  
  
@media (max-width: 768px) {  
  #menu-desktop {  
    display: none;  
  }
```

```
#menu-mobile {  
    display: block;  
    cursor: pointer;  
}  
}
```

jQuery:

javascript

```
$(document).ready(function() {  
    $("#menu-mobile").click(function() {  
        $("#menu-desktop").slideToggle();  
    });  
});
```

ACTIVIDAD PRÁCTICA INDIVIDUAL (20 minutos)

Enunciado:

Diseñar una página responsive que cumpla con:

- Un **contenedor flexible** que reorganice el contenido en columnas o filas según el ancho.
- Un **menú responsive**: horizontal en desktop, hamburguesa en móvil.
- Ajuste de **tamaños de fuente o imágenes** con unidades relativas o jQuery.

Pistas para la implementación:

Contenedor con Flexbox:

css

```
.flex-container {  
    display: flex;  
    gap: 20px;  
}  
  
@media (max-width: 600px) {  
    .flex-container {  
        flex-direction: column;  
    }  
}
```

Ajuste dinámico con jQuery:

javascript

```
$(window).resize(function() {  
    if ($(window).width() < 500) {  
        $("h1").css("font-size", "1.2rem");  
    } else {  
        $("h1").css("font-size", "2rem");  
    }  
});
```



CIERRE DE SESIÓN (15 minutos)

💬 Reflexión:

¿Qué diferencias existen entre CSS puro y jQuery para la adaptación de interfaces?

CSS	jQuery
Declarativo	Imperativo
Alto rendimiento	Flexibilidad dinámica

Ideal para estructura

Útil para efectos y control en tiempo real

 **Revisión:**

- ¿El sitio se adapta correctamente en diferentes tamaños?
- ¿El menú cambia su comportamiento de forma fluida?
- ¿Hay interacción dinámica usando jQuery?

 **Tarea:**

Crear una **galería de imágenes responsive** y un **menú dinámico** usando CSS para diseño y jQuery para control interactivo:

- Imágenes que se ajusten al ancho del contenedor.
- Menú colapsable en pantallas pequeñas.
- Opcional: efectos visuales con `.fadeToggle()` o `.slideToggle()`.

