



TECNOLOGO ANALISIS Y DESARROLLO DESOFTWARE

Ficha 2758368



GA7-220501096-AA1-EV02

APRENDIZ ANDREA VILLARREAL MEGLAN

SERVICIO NACIONAL DE APRENDIZAJE
Regional Quindio

Introducción

En este ejercicio, se desarrollará una aplicación web sencilla para una empresa de agroindustria que incluye un formulario de inscripción. La aplicación permite a los usuarios navegar entre diferentes secciones como “Inicio”, “Quienes Somos”, “Catálogo de Productos”, “Pague Aquí”, “Mi Factura” e “Inscripciones”. En la sección de inscripciones, los usuarios pueden registrar sus datos personales, los cuales se mostrarán en una lista de personas inscritas.

Objetivo

El objetivo de este ejercicio es aprender a:

1. **Crear una estructura HTML** para una aplicación web con múltiples secciones.
2. **Utilizar JavaScript** para manejar la navegación entre secciones y la gestión de formularios.
3. **Aplicar estilos CSS** para mejorar la apariencia visual de la aplicación.
4. **Entender las convenciones de nomenclatura** en JavaScript, como PascalCase para clases y camelCase para variables.
5. **Implementar funciones JavaScript** para manipular el DOM, obtener valores de formularios y actualizar el contenido de la página dinámicamente.

Este ejercicio proporciona una base sólida para desarrollar aplicaciones web interactivas y gestionar datos de usuarios de manera eficiente. Si tienes alguna pregunta adicional o necesitas más detalles, ¡no dudes en preguntar!

Código JavaScript

<script>

```
function showPage(pageId) {  
    const pages = document.querySelectorAll('.page');  
    pages.forEach(page => {  
        page.classList.remove('active');  
        page.style.opacity = 0;  
    });  
    const activePage = document.getElementById(pageId);  
    activePage.classList.add('active');  
    setTimeout(() => {  
        activePage.style.opacity = 1;  
    }, 50);  
}  
  
function registrar() {  
    const usuario = document.getElementById('usuario').value;  
    const contrasena = document.getElementById('contrasena').value;  
    const nombre = document.getElementById('nombre').value;  
    const apellido = document.getElementById('apellido').value;  
    const cedula = document.getElementById('cedula').value;  
    const fechaNacimiento = document.getElementById('fechaNacimiento').value;  
  
    const listaInscritos = document.getElementById('listaInscritos');  
    const nuevoInscrito = document.createElement('li');  
    nuevoInscrito.textContent = `${nombre} ${apellido} - ${cedula} - ${fechaNacimiento}`;  
    listaInscritos.appendChild(nuevoInscrito);  
}
```

</script>

Descripción de Nombres de Clases y Variables

1. Clases:

- Las clases en JavaScript se nombran utilizando la convención de **PascalCase**. Esto significa que cada palabra en el nombre de la clase comienza con una letra mayúscula.
- Ejemplo: `class Persona { ... }`

2. Variables:

- Las variables en JavaScript se nombran utilizando la convención de **camelCase**. Esto significa que la primera palabra comienza con una letra minúscula y cada palabra subsiguiente comienza con una letra mayúscula.
- Ejemplo: `let nombreCompleto = 'Juan Pérez';`
- Las variables de instancia dentro de una clase también siguen la convención de camelCase.
- Ejemplo: `this.nombre, this.edad`

Explicación Detallada del Código JavaScript

1. Función showPage:

- `function showPage(pageId) { ... }`
 - Esta es una función declarada utilizando la palabra clave `function`. Las funciones en JavaScript se utilizan para agrupar código que realiza una tarea específica.
- Dentro de la función, utilizamos `document.querySelectorAll('.page')` para seleccionar todos los elementos con la clase `page`.
- `pages.forEach(page => { ... });`
 - Utilizamos un bucle `forEach` para iterar sobre cada elemento de la lista `pages` y remover la clase `active` y establecer la opacidad a 0.
- `const activePage = document.getElementById(pageId);`
 - Seleccionamos la página activa utilizando el `pageId` proporcionado.
- `activePage.classList.add('active');`
 - Añadimos la clase `active` a la página seleccionada.
- `setTimeout(() => { activePage.style.opacity = 1; }, 50);`
 - Utilizamos `setTimeout` para establecer la opacidad de la página activa a 1 después de 50 milisegundos.

2. Función registrar:

- `function registrar() { ... }`
 - Esta función maneja el registro de un nuevo usuario.
- Dentro de la función, obtenemos los valores de los campos del formulario utilizando `document.getElementById`.
- `const listaInscritos = document.getElementById('listaInscritos');`
 - Seleccionamos el elemento `ul` donde se mostrarán los inscritos.
- `const nuevoInscrito = document.createElement('li');`
 - Creamos un nuevo elemento de lista (`li`) que se añadirá a la lista de inscritos.
- `nuevoInscrito.textContent = {nombre}{apellido} - {cedula} - {fechaNacimiento};`

- Asignamos el contenido de texto al nuevo elemento de lista utilizando una plantilla de cadena (template string) para incluir los valores de nombre, apellido, cedula y fechaNacimiento.
- `listaInscritos.appendChild(nuevoInscrito);`
 - Añadimos el nuevo elemento de lista (li) a la lista de inscritos (ul).

Conclusión

Este ejercicio ha demostrado cómo combinar HTML, CSS y JavaScript para crear una aplicación web interactiva y funcional. A través de la implementación de un formulario de inscripción y la navegación entre diferentes secciones de la página, hemos aprendido a:

1. **Estructurar una página web** con múltiples secciones y formularios.
2. **Utilizar JavaScript** para manipular el DOM, gestionar eventos y actualizar dinámicamente el contenido de la página.
3. **Aplicar estilos CSS** para mejorar la apariencia y la usabilidad de la aplicación.
4. **Seguir las convenciones de nomenclatura** en JavaScript, asegurando un código limpio y mantenible.