

Prueba 2: Git y GitHub - Desarrollo de Página Básica con HTML, CSS y JS

Instrucciones Generales:

1. Esta prueba es **individual** y debes completarla en **1 hora y 20 minutos**.
2. Vas a desarrollar una página web básica utilizando **HTML**, **CSS**, y **JavaScript**. Debes crear la estructura y personalizar algunos elementos con tus propios colores y textos.
3. Al finalizar, debes subir el proyecto a **GitHub** en la rama `main` y luego crear la rama `gh-pages` para visualizar el sitio en **GitHub Pages**.
4. Sigue los pasos detallados a continuación y realiza una investigación breve sobre cada comando nuevo que utilices en **Git** o **GitHub** para comprender mejor el proceso.

Instrucciones Paso a Paso

1. Crear el proyecto y la estructura de carpetas en tu computadora

- Crea una nueva carpeta en tu computadora con el nombre **pagina-basica-html-css**.
- Dentro de esta carpeta, organiza tus archivos de la siguiente manera:

```
pagina-basica-html-css/  
├─ index.html  
├─ css/  
│   └─ styles.css  
├─ js/  
│   └─ script.js
```

2. Crear el archivo HTML básico

- En el archivo `index.html`, crea una estructura básica HTML y completa los espacios con tus propios textos:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Título de tu página</title>
  <link rel="stylesheet" href="css/styles.css">
  <script src="js/script.js" defer></script>
</head>
<body>
  <header>
    <h1>Título Principal</h1> <!-- Reemplaza con un título relevante -->
  </header>
  <main>
    <section>
      <p>Párrafo introductorio sobre el contenido de la página.</p> <!-- Escribe un breve párrafo -->
      <button onclick="mostrarMensaje()">Haz clic aquí</button> <!-- Este botón activará el mensaje en JS -->
    </section>
  </main>
  <footer>
    <p>&copy; 2024 - Tu Nombre</p>
  </footer>
</body>
</html>
```

3. Crear el archivo CSS

- En el archivo `css/styles.css`, agrega tus propios colores y estilos para personalizar la página:

```
body {  
  font-family: Arial, sans-serif;  
  margin: 0;  
  padding: 0;  
  /* Cambia el color de fondo a uno de tu preferencia */  
  background-color: #f0f0f0;  
}  
  
header {  
  /* Agrega un color de fondo y color de texto de tu elección */  
  background-color: #333;  
  color: white;  
  text-align: center;  
  padding: 1em 0;  
}  
  
main {  
  padding: 20px;  
  text-align: center;  
}  
  
footer {  
  background-color: #333;  
  color: white;  
  text-align: center;  
  padding: 1em 0;  
  position: fixed;  
  width: 100%;  
  bottom: 0;  
}
```

- **Nota:** Personaliza el color de fondo y de texto del `header` y del `footer` según tus preferencias.

4. Crear el archivo JavaScript

- En el archivo `js/script.js`, agrega una función simple que muestre un mensaje cuando el usuario haga clic en el botón:

```
function mostrarMensaje() {  
    alert("Escribe aquí el mensaje que deseas mostrar");  
}
```

- **Nota:** Elige el mensaje que quieres mostrar en la alerta y pruébalo haciendo clic en el botón en la página.

5. Inicializar un repositorio local y hacer el primer commit

- Abre una terminal en la carpeta del proyecto y realiza lo siguiente:

```
git init  
git add .  
git commit -m "Primera versión de la página básica con est  
ructura de carpetas y personalización"
```

- **Nota:** Investiga brevemente qué hace cada uno de estos comandos (`git init`, `git add`, y `git commit`) para comprender mejor el flujo de trabajo en Git.

6. Crear un repositorio en GitHub y subir tu proyecto a la rama `main`

- Ve a **GitHub** y crea un nuevo repositorio con el mismo nombre, **pagina-basica-html-css**.
- Vuelve a la terminal y conecta tu repositorio local con el repositorio remoto en GitHub:

```
git remote add origin <https://github.com/tu-usuario/pagina-basica-html-css.git>
```

```
git branch -M main
git push -u origin main
```

- **Nota:** Investiga qué hace cada uno de estos comandos para entender cómo conectas tu repositorio local con GitHub.

7. Crear una nueva rama `gh-pages` para publicar en GitHub Pages

- En la terminal, crea y cambia a la rama `gh-pages` :

```
git checkout -b gh-pages
```

- Sube esta rama a GitHub:

```
git push -u origin gh-pages
```

- **Nota:** Investiga qué hace la rama `gh-pages` y cómo GitHub Pages utiliza esta rama para visualizar tu sitio en línea.

8. Configurar GitHub Pages para ver tu página en línea

- Entra a la configuración de tu repositorio en **GitHub**.
- En la pestaña **Pages**, selecciona la rama `gh-pages` como la fuente de GitHub Pages.
- Una vez configurado, GitHub te proporcionará un enlace donde podrás ver la página publicada.
- Espera unos minutos para que la página se publique y verifica que el enlace funcione.

9. Finalización

- Asegúrate de que la página esté publicada correctamente y accesible en GitHub Pages.
- Envía el enlace del repositorio de GitHub **y** el enlace a la página publicada en **GitHub Pages** a tu profesor al correo

Criterios de Evaluación:

1. **Cumplimiento de los requisitos:** La página debe contener los elementos HTML, CSS y JavaScript indicados, junto con los colores y textos personalizados.
2. **Uso correcto de Git y GitHub:** Se evaluará que hayas hecho el commit y el push a las ramas `main` y `gh-pages` correctamente.
3. **Publicación en GitHub Pages:** La página debe estar visible públicamente en GitHub Pages.
4. **Calidad del código:** Se valorará que el código esté bien organizado, con estructura de carpetas adecuada y estilos personalizados.
5. **Prueba Sumativa:** Además, debes realizar la prueba **Sumativa 2** de **Quinttus** y entregarla junto con esta actividad.

Esta prueba te permitirá demostrar tu capacidad para trabajar con **Git**, **GitHub**, y crear una página web básica utilizando **HTML**, **CSS**, y **JavaScript**, además de completar la evaluación adicional desde Quinttus. ¡Buena suerte!