



**Universidad Tecnológica de Querétaro**

Ingeniería en Desarrollo y Gestión de Software

# **Implementación de Landing Page**

## **con CI/CD de GitHub**

*Proyecto: MusicVibe - Plataforma Musical*

**Estudiante:** Alfonso Esaú Leyva Pérez  
**Matrícula:** 2022371140  
**Materia:** Desarrollo Web Integral  
**Profesor:** M.C.C.Emmanuel Martínez Hernández  
**Fecha:** 21 de julio de 2025

**Documento elaborado para fines académicos**

# Índice

<b>1</b>	<b>Introducción</b>	<b>2</b>
1.1	Objetivos . . . . .	2
1.2	Alcance . . . . .	2
<b>2</b>	<b>Análisis y Planificación</b>	<b>2</b>
2.1	Análisis de Requisitos . . . . .	2
2.2	Arquitectura del Proyecto . . . . .	3
<b>3</b>	<b>Implementación Técnica</b>	<b>3</b>
3.1	Estructura HTML . . . . .	3
3.2	Estilos CSS . . . . .	4
3.3	Funcionalidad JavaScript . . . . .	4
<b>4</b>	<b>Configuración CI/CD</b>	<b>5</b>
4.1	GitHub Actions Workflow . . . . .	5
4.2	Optimización y Performance . . . . .	5
<b>5</b>	<b>Pruebas y Validación</b>	<b>6</b>
5.1	Pruebas de Funcionalidad . . . . .	6
5.2	Validación de Código . . . . .	6
<b>6</b>	<b>Despliegue y Monitoreo</b>	<b>6</b>
6.1	GitHub Pages . . . . .	6
6.2	Monitoreo . . . . .	6
<b>7</b>	<b>Conclusiones</b>	<b>7</b>
7.1	Logros Alcanzados . . . . .	7
7.2	Lecciones Aprendidas . . . . .	7
7.3	Mejoras Futuras . . . . .	7
<b>8</b>	<b>Referencias</b>	<b>7</b>
8.1	Enlaces del Proyecto . . . . .	7
8.2	Tecnologías y Herramientas . . . . .	8
8.3	Recursos de Aprendizaje . . . . .	8

# 1. Introducción

Este documento presenta la implementación completa de una landing page musical llamada "MusicVibe" utilizando tecnologías web modernas y un sistema de integración y despliegue continuo (CI/CD) mediante GitHub Actions y GitHub Pages.

El proyecto tiene como objetivo demostrar las mejores prácticas en desarrollo web frontend, incluyendo diseño responsivo, optimización de rendimiento, y automatización del proceso de despliegue.

## 1.1. Objetivos

- Crear una landing page atractiva y funcional sobre música
- Implementar un diseño responsivo que funcione en todos los dispositivos
- Configurar un pipeline de CI/CD automatizado
- Desplegar la aplicación en GitHub Pages
- Documentar todo el proceso de desarrollo e implementación

## 1.2. Alcance

El proyecto abarca desde la conceptualización y diseño de la interfaz hasta la implementación técnica y el despliegue automatizado, incluyendo:

- Desarrollo frontend con HTML5, CSS3 y JavaScript
- Implementación de animaciones y efectos visuales
- Configuración de GitHub Actions para CI/CD
- Optimización para SEO y rendimiento
- Documentación técnica completa

# 2. Análisis y Planificación

## 2.1. Análisis de Requisitos

Para el desarrollo de MusicVibe, se identificaron los siguientes requisitos:

### **Requisitos Funcionales:**

- Navegación intuitiva entre secciones
- Formulario de contacto funcional
- Reproductor musical interactivo (simulado)
- Diseño responsivo para móviles y desktop
- Animaciones suaves y atractivas

### Requisitos No Funcionales:

- Tiempo de carga menor a 3 segundos
- Compatibilidad con navegadores modernos
- Accesibilidad web (WCAG 2.1)
- SEO optimizado
- Código mantenible y escalable

## 2.2. Arquitectura del Proyecto

La arquitectura del proyecto sigue un patrón de aplicación web estática:

- **Frontend:** HTML5, CSS3, JavaScript vanilla
- **Hosting:** GitHub Pages
- **CI/CD:** GitHub Actions
- **Control de versiones:** Git/GitHub

## 3. Implementación Técnica

### 3.1. Estructura HTML

La estructura HTML utiliza elementos semánticos para mejorar la accesibilidad y SEO:

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>MusicVibe - Tu Mundo Musical</title>
7   <!-- Meta tags para SEO -->
8   <meta name="description" content="Plataforma musical para descubrir, crear y compartir música">
9   <meta name="keywords" content="música, streaming, producción musical">
10 </head>
11 <body>
12   <header class="header">
13     <!-- Navegación -->
14   </header>
15
16   <main>
17     <section id="home" class="hero">
18       <!-- Sección principal -->
19     </section>
20
21     <section id="about" class="about">
22       <!-- Acerca de nosotros -->
23     </section>
```

```
24      <!-- M s secciones -->
25    </main>
26
27    <footer class="footer">
28      <!-- Pie de p gina -->
29    </footer>
30  </body>
31 </html>
```

Listing 1: Estructura básica HTML

### 3.2. Estilos CSS

Se implementó un sistema de diseño moderno utilizando:

- **CSS Grid y Flexbox:** Para layouts responsivos
- **Custom Properties:** Variables CSS para consistencia
- **Media Queries:** Adaptabilidad a diferentes pantallas
- **Animaciones CSS:** Transiciones suaves

```
1 :root {
2   --primary-color: #667eea;
3   --secondary-color: #764ba2;
4   --accent-color: #ffd700;
5   --text-color: #333;
6   --bg-color: #f8f9fa;
7 }
8
9 .hero {
10   background: linear-gradient(135deg, var(--primary-color) 0%, var(--
11     secondary-color) 100%);
12   min-height: 100vh;
13   display: flex;
14   align-items: center;
15 }
16 @media (max-width: 768px) {
17   .hero-content {
18     grid-template-columns: 1fr;
19     text-align: center;
20   }
21 }
```

Listing 2: Variables CSS y diseño responsivo

### 3.3. Funcionalidad JavaScript

El JavaScript implementa:

- Navegación móvil responsiva
- Scroll suave entre secciones

- Validación de formularios
- Animaciones de scroll
- Reproductor musical interactivo

## 4. Configuración CI/CD

### 4.1. GitHub Actions Workflow

Se configuró un pipeline automatizado que:

1. Detecta cambios en la rama principal
2. Ejecuta pruebas de calidad de código
3. Minifica archivos CSS y JavaScript
4. Despliega automáticamente en GitHub Pages

```
1 name: Deploy to GitHub Pages
2
3 on:
4   push:
5     branches: [ main ]
6
7 jobs:
8   build-and-deploy:
9     runs-on: ubuntu-latest
10
11     steps:
12     - name: Checkout
13       uses: actions/checkout@v3
14
15     - name: Build
16       run: |
17         mkdir -p dist
18         npx html-minifier --collapse-whitespace index.html -o dist/index.html
19         npx cleancss -o dist/styles.css styles.css
20
21     - name: Deploy
22       uses: peaceiris/actions-gh-pages@v3
23       with:
24         github_token: ${ secrets.GITHUB_TOKEN }
25         publish_dir: ./dist
```

Listing 3: Configuración GitHub Actions

### 4.2. Optimización y Performance

Se implementaron las siguientes optimizaciones:

- **Minificación:** Reducción del tamaño de archivos

- **Compresión:** Optimización de imágenes
- **Lazy Loading:** Carga diferida de elementos
- **Critical CSS:** CSS crítico inline

## 5. Pruebas y Validación

### 5.1. Pruebas de Funcionalidad

Se realizaron pruebas exhaustivas:

- **Navegación:** Verificación de todos los enlaces
- **Formularios:** Validación de campos y envío
- **Responsividad:** Pruebas en diferentes dispositivos
- **Performance:** Análisis con Lighthouse

### 5.2. Validación de Código

- **HTML:** Validación W3C
- **CSS:** Validación CSS3
- **JavaScript:** ESLint para calidad de código
- **Accesibilidad:** Pruebas con herramientas WAVE

## 6. Despliegue y Monitoreo

### 6.1. GitHub Pages

La aplicación se despliega automáticamente en GitHub Pages:

- **URL de producción:** <https://tu-usuario.github.io/musicvibe-landing>
- **SSL/TLS:** Certificado automático
- **CDN:** Distribución global de contenido

### 6.2. Monitoreo

Se configuró monitoreo básico:

- **Uptime:** Verificación de disponibilidad
- **Performance:** Métricas de velocidad
- **Analytics:** Seguimiento de usuarios (opcional)

## 7. Conclusiones

### 7.1. Logros Alcanzados

- Implementación exitosa de una landing page moderna y atractiva
- Configuración completa de pipeline CI/CD automatizado
- Despliegue en producción con GitHub Pages
- Documentación técnica completa
- Código optimizado y mantenible

### 7.2. Lecciones Aprendidas

- Importancia de la planificación en proyectos web
- Beneficios de la automatización en el despliegue
- Valor de las pruebas continuas
- Relevancia del diseño responsivo

### 7.3. Mejoras Futuras

- Implementación de Progressive Web App (PWA)
- Integración con APIs de música reales
- Sistema de autenticación de usuarios
- Dashboard de administración
- Optimización avanzada de performance

## 8. Referencias

### 8.1. Enlaces del Proyecto

- Repositorio GitHub: <https://github.com/tu-usuario/musicvibe-landing>
- Demo en vivo: <https://tu-usuario.github.io/musicvibe-landing>
- Documentación técnica: Disponible en el repositorio



## 8.2. Tecnologías y Herramientas

- HTML5: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- CSS3: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- GitHub Actions: <https://docs.github.com/en/actions>
- GitHub Pages: <https://pages.github.com/>

## 8.3. Recursos de Aprendizaje

- MDN Web Docs: <https://developer.mozilla.org/>
- W3C Standards: <https://www.w3.org/>
- GitHub Documentation: <https://docs.github.com/>
- Web.dev: <https://web.dev/>