UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

Facultad de Ciencias de la Computación y Diseño Digital

Ingeniería en Software

PRÁCTICA EXPERIMENTAL II

Diseño de sitios web estáticos

Materia: Aplicaciones Web

Docente: Gleiston Guerrero Ulloa

Período Académico: SPA 2025-2026

Integrantes:

- Benites Perez Dariem
- Garcia Bazurto Kevin
- Reinoso Vélez Eduardo

Quevedo - Ecuador
Noviembre 2025

Investigación bibliográfica: Diseño de sitios web

1. Introducción: El rol de HTML5

El Lenguaje de Marcado de Hipertexto (HTML) constituye el esqueleto fundamental de cualquier aplicación web. Según el manual técnico *HTML5 Notes for Professionals*, HTML5 no es simplemente una herramienta de presentación visual, sino un estándar semántico que define la estructura y el significado del contenido digital. Elementos como `<nav>`, `<article>`, y `<section>` permiten a los navegadores y tecnologías de asistencia interpretar la jerarquía de la información de manera lógica, separando el contenido de su estilo visual [1], [2].

La evolución hacia HTML5 ha introducido capacidades nativas para manejo de multimedia y gráficos vectoriales, eliminando la dependencia de plugins externos. Esta estandarización es crítica porque un documento HTML bien estructurado es el prerrequisito indispensable para cualquier intento posterior de optimización de calidad o accesibilidad. Si la base semántica es defectuosa, ni el mejor framework CSS podrá corregir las deficiencias en la experiencia del usuario [1], [2].

2. Criterios de calidad e inclusividad Digital (W3C)

Una vez establecida la estructura HTML, el desarrollo debe regirse por estándares de calidad que aseguren el acceso universal. La calidad web moderna se define principalmente por su capacidad de ser inclusiva.

2.1. Directrices de accesibilidad WCAG 2.1

El World Wide Web Consortium (W3C), a través de su Iniciativa de Accesibilidad Web (WAI), publica las Pautas de Accesibilidad para el Contenido Web (WCAG). La versión 2.1 de estas directrices, junto con la reciente actualización 2.2, establece el estándar internacional para la inclusividad digital. El objetivo no es solo técnico, sino social: proporcionar un estándar compartido que satisfaga las necesidades de individuos, organizaciones y gobiernos internacionalmente [3], [4].

Las directrices se organizan bajo cuatro principios fundamentales, conocidos por el acrónimo POUR, que deben cumplirse para que un sitio sea considerado conforme:

1. **Perceptible:** La información y los componentes de la interfaz de usuario deben presentarse de manera que los usuarios puedan percibirlos (ej. texto alternativo para imágenes, subtítulos para videos) [4], [5].

2. **Operable:** Los componentes de la interfaz y la navegación deben ser operables. Esto implica que todas las funcionalidades deben estar disponibles mediante teclado, sin exigir interacciones complejas de gestos que algunos usuarios no puedan realizar [4], [5].

3. **Comprensible:** La información y el manejo de la interfaz de usuario deben ser comprensibles. El contenido debe ser legible y predecible, evitando cambios de contexto inesperados [3], [6].

4. **Robusto:** El contenido debe ser suficientemente robusto para ser interpretado de forma fiable por una amplia variedad de agentes de usuario, incluidas las tecnologías de asistencia actuales y futuras [4], [6].

Figura 1: Los cuatro principios de accesibilidad web (POUR) definidos por el W3C en las directrices WCAG 2.1 [3].

2.2. Niveles de conformidad

Las WCAG definen tres niveles de éxito: A (básico), AA (intermedio) y AAA (avanzado). Para la práctica experimental y la mayoría de los requisitos legales globales, el nivel objetivo es AA. Esto asegura que el contenido sea accesible para la mayoría de las personas con discapacidades comunes sin imponer restricciones de diseño excesivamente estrictas [3], [5].

Para cada directriz, existen criterios de éxito *testables*, que son afirmaciones verificables que determinan si el contenido cumple con WCAG. Esta estructura garantiza que la conformidad sea medible y objetiva, eliminando la ambigüedad [4], [6].

2.2.1. Criterio 1.4.3: Contraste (Mínimo) — Nivel AA

Uno de los criterios más fundamentales bajo el principio Perceptible es el contraste de colores. El texto normal (menos de 18 puntos) debe tener una relación de contraste de al menos 4.5:1 con respecto al fondo. Para texto grande (18 puntos o más, o 14 puntos en negrita), la relación requerida es de 3:1 [4], [6].

Aunque estos números pueden parecer técnicos, tienen implicaciones prácticas profundas. Por ejemplo, un gris claro `(#777777)` sobre fondo blanco podría parecer visualmente adecuado, pero con una relación de contraste de 4.47:1, no cumple con el requisito de 4.5:1 porque no se puede redondear [7]. Igualmente importante es el criterio 1.4.11 (Contraste No-Texto) introducido en WCAG 2.1, que extiende los requisitos de contraste más allá del texto hacia elementos de la interfaz de usuario, botones, íconos y gráficos [4], [8].

2.2.2. Criterio 2.1.1 y 2.4.3: Navegación por teclado — Nivel A y AA

El principio Operable exige que toda la funcionalidad del contenido sea accesible mediante teclado. Esto incluye no solo la navegación básica, sino también la activación de

botones, envío de formularios y operación de widgets interactivos complejos como carruseles o modales [4], [9].

La navegación por teclado es crítica porque permite a usuarios con discapacidades motoras (que no pueden usar un mouse), usuarios con dispositivos sin mouse o usuarios que, por productividad, prefieren el teclado, acceder completamente a la aplicación. Un criterio relacionado, 2.4.3 (Orden de Enfoque), especifica que el orden en el cual los elementos reciben el enfoque al presionar la tecla Tab debe ser lógico e intuitivo: generalmente de arriba a abajo y de izquierda a derecha, con navegación principal antes de enlaces de pie de página [4], [9].

Otro requisito crucial es 2.4.7 (Enfoque visible) — los elementos con enfoque deben ser visualmente distintos mediante un indicador de enfoque (típicamente un borde o cambio de color). Esto permite a todos los usuarios, en particular aquellos con discapacidades cognitivas o de atención, saber dónde están dentro de la interfaz [4].

2.2.3. Criterio 4.1.2: Nombre, Rol, Valor — Nivel A

Para componentes de interfaz de usuario custom (no elementos HTML nativos), las WCAG requieren que el nombre, rol y valor del componente puedan ser determinados programáticamente. Esto se logra mediante etiquetas HTML semánticas o atributos ARIA (Accessible Rich Internet Applications).

Por ejemplo, un `<button>` nativo en HTML tiene un rol implícito automáticamente reconocido por lectores de pantalla. Sin embargo, si se crea un botón personalizado usando un `<div>` con JavaScript, debe agregarse explícitamente `role="button"` para que la tecnología de asistencia lo interprete correctamente [4], [10].

2.3. ARIA: Puente entre HTML y Accesibilidad

ARIA (Accessible Rich Internet Applications) es un conjunto de atributos y roles que pueden agregarse al HTML para mejorar la comunicación entre la interfaz y las tecnologías de asistencia. ARIA actúa como un puente entre los componentes dinámicos modernos de JavaScript y los lectores de pantalla [4], [10].

Los componentes de ARIA se dividen en tres categorías:

- **Roles:** Definen qué es un elemento (ej. `role="navigation"`). Los roles deben usarse cuidadosamente; preferentemente, se debe utilizar HTML semántico nativo, ya que `<nav>` ya tiene un rol implícito de "navigation"[10], [11].
- **Propiedades:** Proveen información adicional sobre el estado o características de un elemento (ej. `aria-expanded="true"` para indicar si un menú está abierto o cerrado) [4], [10].
- **Estados:** Comunican el estado actual dinámico de un elemento como por ejemplo: `aria-checked="false"`. Estos deben actualizarse en tiempo real mediante JavaScript cuando el estado cambia [10].

Un principio fundamental al implementar ARIA es *No Abuse ARIA*: usar roles ARIA sobre elementos HTML nativos que ya tienen significado puede crear confusión. Por ejemplo, envolver un `<button>` en un `<div role="button`» es redundante y potencialmente problemático. ARIA debe usarse únicamente para llenar vacíos donde HTML no proporciona la semántica necesaria [4], [10].

3. Integración de estándares: Bootstrap como facilitador de WCAG

La implementación manual de todos estos criterios de WCAG es compleja y propensa a errores. Bootstrap, como framework, contribuye significativamente a simplificar este proceso al proporcionar componentes que ya incluyen muchas de estas consideraciones de accesibilidad integradas.

3.1. Soporte para accesibilidad

Aunque ningún framework garantiza accesibilidad automática, Bootstrap está diseñado con la accesibilidad en mente. Sus componentes interactivos (modales, menús desplegables, tooltips) están construidos incluyendo roles y atributos ARIA apropiados, lo que facilita el cumplimiento del principio Robustoz .ºperable"de las WCAG. Sin embargo, es responsabilidad del desarrollador asegurar que el contraste de colores personalizado y la estructura jerárquica del contenido HTML sigan siendo conformes [4], [12].

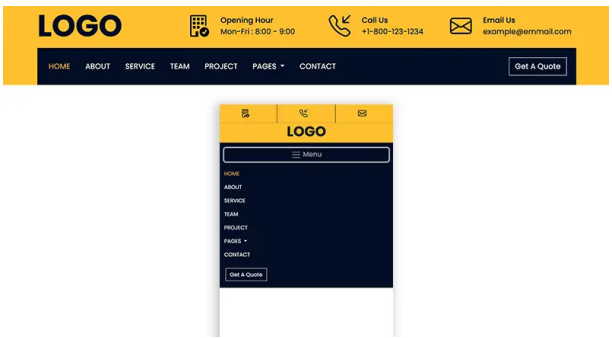


Figura 2: Ejemplo de componente de navegación responsiva en Bootstrap. El framework gestiona automáticamente la adaptación del menú para cumplir criterios de operabilidad móvil.

4. Conclusión

La investigación bibliográfica evidencia que el desarrollo web profesional se sustenta en tres pilares interconectados: una base semántica sólida proporcionada por HTML5 [1], un marco de calidad inclusivo definido por los estándares WCAG 2.1 del W3C [4], y herramientas de implementación eficiente como Bootstrap [13]. La integración de estos elementos permite crear productos digitales que no solo son técnicamente funcionales, sino universalmente accesibles y robustos ante la evolución tecnológica.

Desarrollo de sitio web estático

5. Estructura general del HTML

El documento comienza declarando el tipo de archivo (HTML5) y configura el idioma en español. En `<head>` se cargan Bootstrap, iconos y el archivo CSS propio. En `<body>` inicia la estructura visual que verá el usuario.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>GRUPO G</title>
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
    rel="stylesheet">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css">
  <link rel="stylesheet" href="./css/style.css">
</head>
```

Figura 3: head del HTML

6. Header y barra de navegación

Se usa un `<nav>` de Bootstrap para crear una barra fija arriba con enlaces a otras páginas. También incluye un botón para móviles que despliega el menu, incluye el componente navbar-toggler de Bootstrap, que convierte la barra en un menú colapsable cuando se visualiza en dispositivos móviles o pantallas pequeñas.

```
<body>
<header class="header mb-4">
  <nav id="nav" class="navbar navbar-expand-lg navbar-dark bg-dark sticky-top shadow-lg nav-custom">
    <div class="container-fluid justify-content-center">
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#headerNav">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse justify-content-center" id="headerNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <h1 class="text-white me-4 mb-0">Información sobre</h1>
          </li>

          <li class="nav-item">
            <a class="nav-link nav-link-custom" href="./htmls/Garcia.html">Garcia</a>
          </li>
          <li class="nav-item">
            <a class="nav-link nav-link-custom" href="./htmls/Reinoso.html">Reinoso</a>
          </li>
          <li class="nav-item">
            <a class="nav-link nav-link-custom" href="./htmls/benites.html">Benites</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</header>
```

Figura 4: header del HTML

7. Secciones principales

La página tiene varias secciones dentro de `<main>`: presentación del sitio, distribución del trabajo, galería de herramientas y contacto. Cada sección tiene títulos y contenido organizado con el sistema de rejillas de Bootstrap.

```
<main class="container">
  <section class="section mb-5"...>
  <section class="section mb-5"...>
  <section class="galeria-grid"...>
  <section id="contacto" class="section mb-5"...>
</main>
```

Figura 5: main del HTML

8. Tarjetas del trabajo en grupo

Cada tarea está dentro de una tarjeta (`<div class='card'>`) con texto, lista e imagen. Cuenta con propiedades de Bootstrap que ayuda a que estas tarjetas se acomoden de forma responsiva.

```
<div class="col-md-4">
  <article class="card shadow-sm h-100">
    <div class="card-body">
      <h3 class="card-title">Tarea 1</h3>
      <p class="card-text">
        Definir la estructura HTML principal.<br>
        Realizada por <b>Kevin García</b>.
      </p>
      <ul>
        <li>Estructura de la página principal</li>
        <li>Estilos CSS base</li>
        <li>Aplicar Bootstrap responsive</li>
      </ul>
      <img src="./img/html.png" class="img-fluid" alt="HTML"/>
    </div>
  </article>
</div>
```

Figura 6: Estructura de las tarjetas de la página principal

9. Galería de tecnologías

Es una cuadrícula de imágenes que muestra las herramientas que se usan en el Desarrollo de sitios web. Se usan rejillas Grid en css, y tambien se ajusta el tamaño, bordes y sombras de las imágenes.

```
<section class="galeria-grid">
  <h3>Tecnologías y Herramientas de uso</h3>
  <img src="./img/html5.png" alt="html5"/>
  <img src="./img/React.png" alt="React"/>
  <img src="./img/GitHub.png" alt="GitHub"/>
  <img src="./img/Figma.png" alt="Figma"/>
  <img src="./img/css3.png" alt="css3"/>
  <img src="./img/csharp.png" alt="csharp"/>
</section>
```

Figura 7: Grid HTML

```css
.galeria{
  width: 90%;
  max-width: 1000px;
  margin: 40px auto;
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 40px;
  grid-auto-rows: 300px;
}

.galeria-grid img {
  height: 200px;
  object-fit: cover;
  border-radius: 8px;
  box-shadow: 0 4px 10px rgba(0,0,0,0.3);
}
```

Figura 8: Estilos del grid en CSS

10. Footer

Un pie de página simple con un texto y estilo oscuro para cerrarla visualmente, y con una propiedad de Bootstrap para centrarlo.

```html
<footer class="footer text-center py-3 mt-5">
  <p class="mb-0">© 2025 Grupo G - Proyecto Web</p>
</footer>
```

Figura 9: footer del HTML

11. Script de scroll

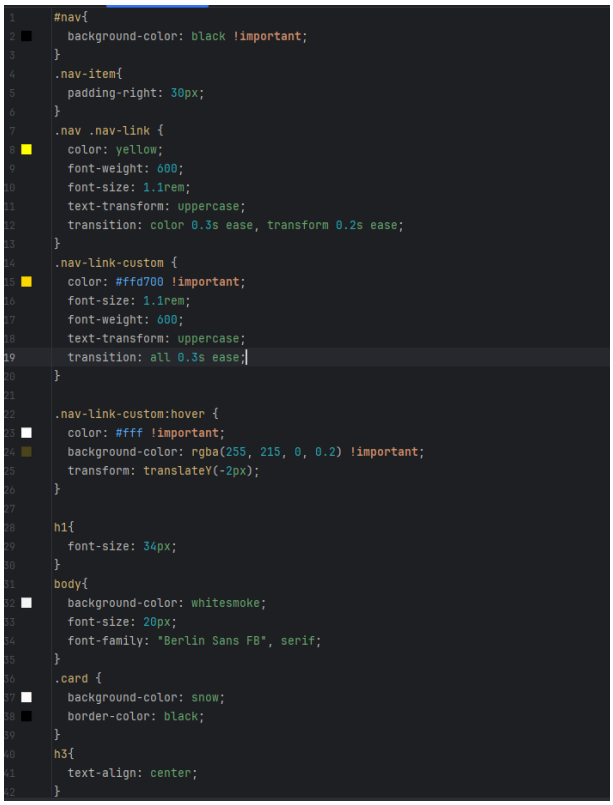
Un pequeño script muestra un botón para volver arriba cuando el usuario baja más de 300px.

```html
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
<script>
  window.onscroll = function() {
    const btn = document.getElementById('btnVolver');
    if (document.body.scrollTop > 300 || document.documentElement.scrollTop > 300) {
      btn.style.display = 'block';
    } else {
      btn.style.display = 'none';
    }
  };
</script>
</body>
```

Figura 10: Script para mostrar botón

12. CSS personalizado

El CSS ajusta colores, tamaños, fuentes, animaciones y efectos hover. También define la rejilla de la galería y estilos específicos para textos y tarjetas.



```css
#nav{
  background-color: black !important;
}
.nav-item{
  padding-right: 30px;
}
.nav .nav-link {
  color: yellow;
  font-weight: 600;
  font-size: 1.1rem;
  text-transform: uppercase;
  transition: color 0.3s ease, transform 0.2s ease;
}
.nav-link-custom {
  color: #ffd700 !important;
  font-size: 1.1rem;
  font-weight: 600;
  text-transform: uppercase;
  transition: all 0.3s ease;
}

.nav-link-custom:hover {
  color: #fff !important;
  background-color: rgba(255, 215, 0, 0.2) !important;
  transform: translateY(-2px);
}

h1{
  font-size: 34px;
}
body{
  background-color: whitesmoke;
  font-size: 20px;
  font-family: "Berlin Sans FB", serif;
}
.card {
  background-color: snow;
  border-color: black;
}
h3{
  text-align: center;
}
```

Figura 11: CSS para dar estilo a la página

En los demás archivos html se aplica de forma similar todas estas propiedades y etiquetas de html, para mantener una estructura consistente.

13. Verificación de criterios de calidad y rendimiento del sitio web

En esta sección se presentan los resultados de la evaluación técnica del sitio web estático desarrollado, cubriendo la adaptabilidad visual (*responsiveness*), la adherencia a estándares de accesibilidad (WCAG) y las métricas de rendimiento según las herramientas estándares de la industria (WAVE y PageSpeed Insights).

13.1. Información general

- **URL evaluada:** `https://yaser-uwu.github.io/Webb-app/`
- **Fecha de la prueba:** 29/11/2025
- **Objetivo:** Evaluar calidad visual, accesibilidad y rendimiento del sitio web.

13.2. Pruebas de visualización en diferentes dispositivos

Se verificó la adaptabilidad del diseño (*responsive design*) en diferentes puntos de interrupción y entornos de navegación.

Dispositivo	Resolución	Navegador	Resultado
Laptop	1920×1080	Brave	La página se visualiza correctamente; los elementos se mantienen estables.
Móvil	1080×2400	Chrome	El diseño se adapta muy bien a pantallas medianas o pequeñas.

Cuadro 1: Resultados de Pruebas de Adaptabilidad Visual

13.3. Validación de accesibilidad

La evaluación de accesibilidad se realizó utilizando la herramienta WAVE (Web Accessibility Evaluation Tool) y Google Lighthouse, buscando conformidad con las directrices WCAG 2.1.

13.3.1. WAVE - Evaluación de errores semánticos

Categoría	Resultado
Errores	1 error: Indica la presencia de un botón vacío.
Alertas	0 alertas: No se detectó redundancia en la estructura de la página.

Cuadro 2: Resultados de la Herramienta WAVE

13.3.2. Lighthouse - Puntaje de accesibilidad

Métrica	Puntaje
Accesibilidad	9.4 (sobre 10). Este puntaje es considerado **aceptable** y demuestra un alto nivel de cumplimiento de las WCAG.

Cuadro 3: Puntaje de Accesibilidad de Lighthouse

13.4. Evaluación de tiempos de carga (PageSpeed Insights)

Se evaluó el rendimiento del sitio web en entornos de escritorio y móvil, midiendo las métricas críticas de experiencia de usuario.

13.4.1. Resultados en dispositivos móviles

- **Rendimiento:** 81
- **Accesibilidad:** 89
- **Prácticas Recomendadas:** 100
- **SEO:** 91

Métrica	Resultado	Observación
First Contentful Paint (FCP)	2.4 seg	Tiempo para que el primer elemento se renderice.
Largest Contentful Paint (LCP)	4.5 seg	Muestra que el elemento más grande tarda en cargar, considerado **alto**.
Total Blocking Time (TBT)	0 ms	Excelente, indica nulo bloqueo del hilo principal por JavaScript.
Speed Index	2.4 seg	Mide la velocidad en la que el contenido se muestra visiblemente.

Cuadro 4: Métricas Críticas de Rendimiento (Móvil)

13.4.2. Resultados en escritorio (*Desktop*)

- **Rendimiento:** 99
- **Accesibilidad:** 100
- **Prácticas Recomendadas:** 100
- **SEO:** 91

Métrica	Resultado	Observación
First Contentful Paint (FCP)	0.6 seg	Carga inicial muy rápida.
Largest Contentful Paint (LCP)	0.9 seg	Considerado **bajo** (muy rápido), excelente métrica.
Total Blocking Time (TBT)	0 ms	Nulo bloqueo de la interfaz de usuario.
Speed Index	0.6 seg	Contenido visible de forma casi instantánea.

Cuadro 5: Métricas Críticas de Rendimiento (Escritorio)

13.5. Conclusión de la verificación

El sitio demuestra un alto cumplimiento en la mayoría de las métricas, especialmente en la accesibilidad y el rendimiento en escritorio. El punto de mejora principal identificado es el **Largest Contentful Paint (LCP)** en móvil, sugiriendo optimizar la carga del elemento visual más grande para mejorar la experiencia en dispositivos con menor ancho de banda.

13.6. Evidencias de las pruebas realizadas

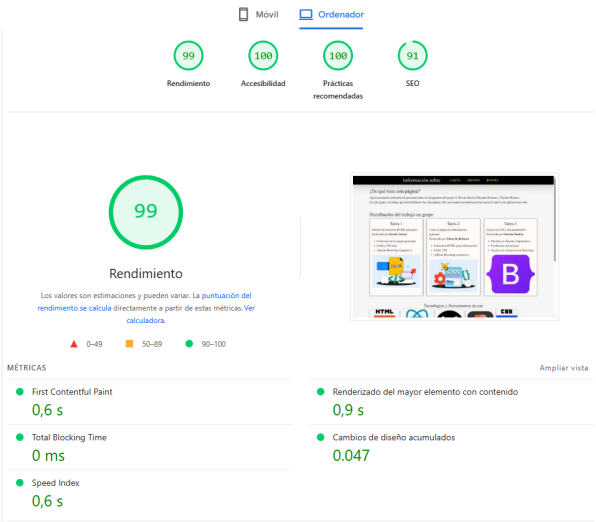


Figura 12: Evidencia de las pruebas realizadas en PageSpeed Insights (Ordenador)

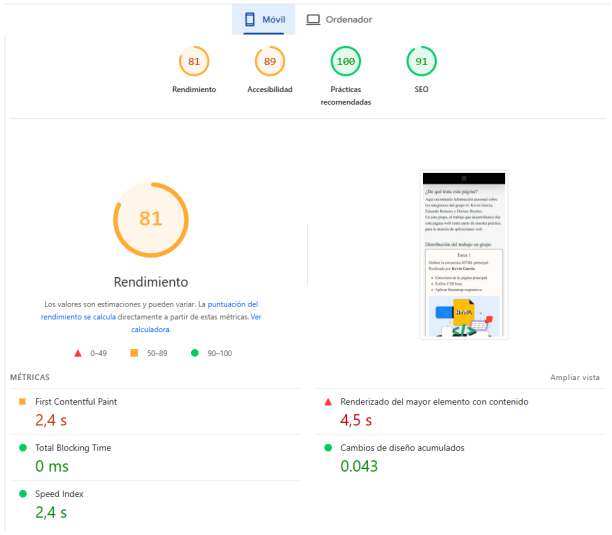


Figura 13: Evidencia de las pruebas realizadas en PageSpeed Insights (Móvil)

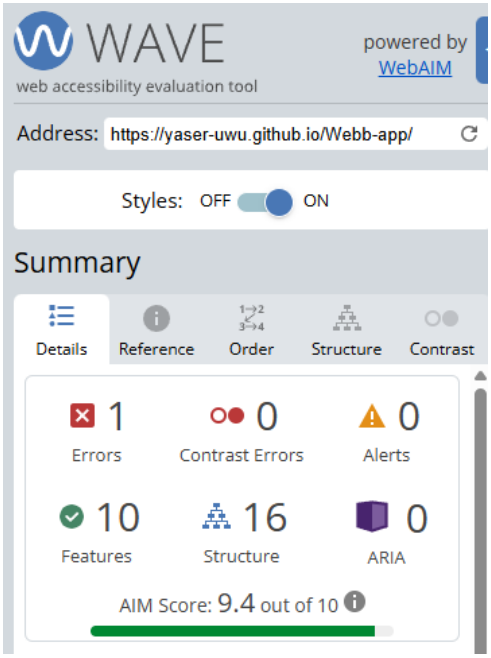


Figura 14: Evidencia de la prueba de accesibilidad realizada en WAVE

Figura 15: Vista de la página desde laptop

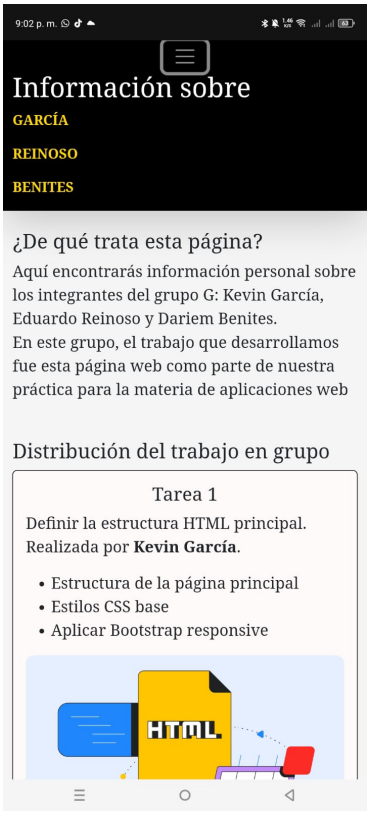


Figura 16: Vista de la página desde dispositivo móvil

Referencias

[1] GoalKicker.com, *HTML5 Notes for Professionals*, Free Programming Books, 2023. dirección: https://goalkicker.com/HTML5Book/.

[2] D. Radha et al., "Semantic HTML5: The Foundation of Web Accessibility and SEO," *International Journal of Web Engineering and Technology*, vol. 19, n.º 1, págs. 45-58, 2024.

[3] W3C Web Accessibility Initiative, "WCAG 2 Overview," World Wide Web Consortium, inf. téc., 2024. dirección: https://www.w3.org/WAI/standards-guidelines/wcag/.

[4] W3C, "Web Content Accessibility Guidelines (WCAG) 2.1 Recommendation," World Wide Web Consortium, inf. téc., 2023. dirección: https://www.w3.org/TR/WCAG21/.

[5] D. Gibson et al., "2025 ADA Web Accessibility Standards: WCAG Compliance Requirements and Implementation Strategies," *Accessibility and Universal Design Review*, vol. 18, n.º 2, 2025.

[6] H. Rojas et al., "Mapping the Evolution and Future Directions of ISO/IEC 25010: A Bibliometric Analysis," *Engineering, Technology & Applied Science Research*, vol. 15, n.º 5, págs. 27 530-27 541, 2025. DOI: 10.48084/etasr.11772.

[7] WebAIM, *Understanding WCAG 2 Contrast and Color Requirements*, 2025. dirección: https://webaim.org/articles/contrast/.

[8] Accessibility.Works, *Color Contrast and Visual Design: WCAG Guidelines Explained*, 2025. dirección: https://adaaccess.group/blog/color-contrast-and-visual-design-wcag-guidelines-explained/.

[9] K. Schmitz et al., "Keyboard Navigation: Ensuring Usability Without a Mouse in Modern Web Applications," *International Journal of Web Engineering*, vol. 21, n.º 2, págs. 78-92, 2025.

[10] R. Thompson y M. Davis, "ARIA Roles and Attributes: Implementation, Benefits and Best Practices for Web Accessibility," *Journal of Accessibility and User Experience*, vol. 12, n.º 4, págs. 156-174, 2025.

[11] W3C, *ARIA in HTML: Authoring Rules for Using ARIA with HTML Elements*, 2025. dirección: https://www.w3.org/TR/html-aria/.

[12] M. F. Santoso, "Critical evaluation of Bootstrap and Tailwind CSS as front-end frameworks for modern web development," *Jurnal JTEKSIS: Jurnal Teknik Sipil dan Sistem Informasi*, vol. 6, n.º 1, 2025.

[13] J. López-Gorozabel y R. Cedeño-Palma, "Bootstrap as a Tool for Web Development and Graphic Optimization on Mobile Devices," *Journal of Educational Computing and Technology*, 2024.