

Documentación Técnica Detallada - BlindAssist

Índice

1. [Funcionalidades JavaScript](#)
2. [Estilos CSS](#)
3. [Accesibilidad](#)
4. [Estructura del Proyecto](#)

Funcionalidades JavaScript

1. Sistema de Texto a Voz

Variables Globales

```
javascript
let speechSynthesis = window.speechSynthesis;
let isReading = false;
let currentUtterance = null;
let sectionsToRead = ['quienes', 'consiste', 'tecnico', 'contacto'];
let currentSectionIndex = 0;
```

Función getSectionContent

```
javascript
function getSectionContent(sectionId) {
  const section = document.getElementById(sectionId);
  if (!section) return '';

  let content = '';
  const elements = section.querySelectorAll('h2, p:not(.rol):not(.descripcion)');

  elements.forEach(element => {
    content += element.textContent + '. ';
  });
}
```

```
});  
  
    return content.trim();  
}
```

Explicación: Esta función obtiene el contenido de texto de una sección específica, excluyendo ciertos elementos como roles y descripciones. Combina los títulos y párrafos en un solo texto.

Función readNextSection

```
javascript  
function readNextSection() {  
    if (currentSectionIndex >= sectionsToRead.length) {  
        currentSectionIndex = 0;  
        isReading = false;  
        updateReadButton();  
        return;  
    }  
  
    const sectionId = sectionsToRead[currentSectionIndex];  
    const content = getSectionContent(sectionId);  
  
    if (content) {  
        currentUtterance = new SpeechSynthesisUtterance(content);  
        currentUtterance.lang = 'es-ES';  
        currentUtterance.rate = 1;  
  
        currentUtterance.onend = () => {  
            setTimeout(() => {  
                currentSectionIndex++;  
                readNextSection();  
            }, 1000);  
        };  
  
        speechSynthesis.speak(currentUtterance);  
    }  
}
```

Explicación: Esta función maneja la lectura secuencial de las secciones, con pausas entre ellas y reinicio automático al finalizar.

2. Carrusel de Imágenes

Variables y Configuración

```
javascript
let currentSlide = 0;
const slides = document.querySelectorAll('.carrusel-slide');
const totalSlides = slides.length;
const slideWidth = slides[0].offsetWidth;
```

Función showSlide

```
javascript
function showSlide(index) {
  if (index < 0) {
    currentSlide = totalSlides - 1;
  } else if (index >= totalSlides) {
    currentSlide = 0;
  }

  const offset = -currentSlide * slideWidth;
  document.querySelector('.carrusel-container').style.transform = `translateX(${offset}px)`;
}
```

Explicación: Maneja la transición entre slides, incluyendo el loop infinito y las transiciones suaves.

3. Gráficos con Chart.js

Configuración de Gráficos

```
javascript
function initCharts() {
  // Gráfico de línea
  const ctx1 = document.getElementById('grafico1').getContext('2d');
```

```

new Chart(ctx1, {
  type: 'line',
  data: {
    labels: ['2010', '2015', '2020', '2025'],
    datasets: [{
      label: 'Población ciega mundial',
      data: [39, 43, 48, 52],
      borderColor: '#00cc99',
      tension: 0.4
    }]
  },
  options: {
    responsive: true,
    plugins: {
      legend: {
        position: 'top',
      }
    }
  }
});
}

```

Explicación: Configura los gráficos con datos dinámicos y opciones de personalización.

Estilos CSS

1. Variables y Temas

```

css
:root {
  --primary-color: #00cc99;
  --secondary-color: #0e1f1b;
  --text-color: #d2f4e3;
  --background-gradient: linear-gradient(135deg, #003d33, #006655);
  --card-background: rgba(0, 77, 59, 0.2);
  --transition-speed: 0.3s;
}

```

2. Componentes Principales

Header con Patrón SVG

```
css
header {
  background: var(--background-gradient);
  background-image: url("data:image/svg+xml,%3Csvg width='100' height='100' v
}
```

Carrusel con Transiciones

```
css
.galeria-carrusel {
  position: relative;
  overflow: hidden;
  width: 100%;
  height: 400px;
}

.carrusel-container {
  display: flex;
  transition: transform 0.5s ease-in-out;
  height: 100%;
}

.carrusel-slide {
  min-width: 100%;
  height: 100%;
  transition: opacity 0.5s ease-in-out;
}
```

Tarjetas de Integrantes

```
css
.integrante {
  background: var(--card-background);
}
```

```
border-radius: 10px;
padding: 20px;
transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.integrante:hover {
  transform: translateY(-5px);
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.3);
}
```

3. Media Queries y Responsive Design

```
css
@media (max-width: 768px) {
  .equipo-container {
    grid-template-columns: 1fr;
    gap: 2rem;
  }

  .galeria-carrusel {
    height: 300px;
  }

  .accessibility-controls {
    bottom: 1rem;
    right: 1rem;
    top: auto;
  }
}
```

Accesibilidad

1. Características Implementadas

- Skip link para saltar al contenido principal
- Navegación por teclado
- Texto a voz
- Alt text en imágenes

- ARIA labels
- Alto contraste
- Textos descriptivos

2. Atajos de Teclado

- R: Activar/desactivar narrador
- 1-5: Navegar a secciones principales
- Flechas: Navegar el carrusel
- Tab: Navegar entre elementos interactivos

Estructura del Proyecto

```
proyecto/  
├─ index.html  
├─ styles.css  
├─ scripts.js  
├─ Multimedia/  
│   ├── logos/  
│   │   └─ logo.png  
│   ├── integrantes/  
│   │   ├── integrante1.jpg  
│   │   ├── integrante2.jpg  
│   │   └─ integrante3.jpg  
│   ├── galeria/  
│   │   ├── imagen1.jpg  
│   │   ├── imagen2.jpg  
│   │   ├── imagen3.jpg  
│   │   └─ imagen4.jpg  
│   └─ sponsors/  
│       ├── sponsor1.png  
│       ├── sponsor2.png  
│       └─ sponsor3.png  
└─ DOCUMENTACION_TECNICA.md
```

Dependencias Externas

- Font Awesome 6.5.1 (iconos)

- Google Fonts - Poppins (tipografía)
- Chart.js (gráficos)
- SpeechSynthesis API (texto a voz)