

Fase 3: Validación de Accesibilidad - INMOVA

July
17

Fecha de Ejecución

7 de diciembre de 2025

Objetivo

Validar y certificar el nivel de accesibilidad del sistema INMOVA mediante pruebas automatizadas y análisis estático del código, asegurando cumplimiento con estándares WCAG 2.1 AA.



Resumen Ejecutivo

Herramientas Implementadas

1. **@axe-core/cli** - Análisis automatizado con axe DevTools
2. **pa11y** - Validación adicional de accesibilidad
3. **puppeteer** - Pruebas end-to-end en navegador
4. **Análisis estático de código** - Validación de patrones de accesibilidad

Scripts Creados

```
# Suite completa de pruebas
node scripts/run-all-accessibility-tests.js

# Pruebas individuales
node scripts/test-keyboard-navigation.js
node scripts/test-screen-reader.js
node scripts/test-accessibility.js  # Requiere servidor dev corriendo
```



Resultados de Pruebas Estáticas

1. Navegación por Teclado

Archivos analizados: 936

Patrones Encontrados

Patrón	Ocurrencias	Archivos	Estado
Estilos focus-visible	226	36	 Presente
Atributos aria-label	78	30	 Presente
Roles ARIA	12	10	 Bajo
Manejadores de teclado	12	12	 Bajo
tabindex	4	3	 Muy bajo
Skip links	1	1	 Insuficiente

Problemas Críticos Detectados

Total: 150 problemas en componentes clave

Componentes más afectados:

- components/LanguageSelector.tsx - Falta de focus-visible, aria-labels, roles ARIA
- components/ui/touch-optimized-button.tsx - Sin aria-label ni roles ARIA
- components/ui/tabs.tsx - Falta de accesibilidad completa
- components/ui/select.tsx - Sin manejadores de teclado apropiados
- components/ui/dialog.tsx - Roles ARIA faltantes

Recomendaciones (35 totales)

Top 10 recomendaciones:

1. Inputs sin label asociado en SetupWizard.tsx (3 ocurrencias)
2. Input sin label en photo-gallery.tsx
3. Input sin label en input.tsx
4. Input sin label en MobilePhotoCapture.tsx
5. Inputs sin label en valoraciones/page.tsx (3 ocurrencias)

Puntuación: 15/100  Crítico - Requiere mejoras urgentes

2. Lectores de Pantalla

Archivos analizados: 937

Estadísticas Positivas

Elemento	Cantidad	Evaluación
Elementos semánticos (<code><header></code> , <code><nav></code> , <code><main></code> , etc.)	251	 Bueno
Atributos ARIA	89	 Moderado
Encabezados (<code><h1></code> - <code><h6></code>)	639	 Excelente
Skip links	3	 Muy bajo
Regiones live (<code>aria-live</code>)	17	 Moderado

Problemas Serios (10 archivos afectados)

Inputs sin etiqueta accesible:

1. `SetupWizard.tsx` - 3 inputs
2. `valoraciones/page.tsx` - 3 inputs
3. `room-rental/[unitId]/page.tsx` - 2 inputs
4. `publicaciones/page.tsx` - 2 inputs
5. `notificaciones/page.tsx` - 2 inputs

Problemas Moderados

Jerarquía de encabezados:

- Saltos de h1 → h3 detectados en:
- `votaciones/page.tsx`
- `valoraciones/page.tsx`
- `unidades/page.tsx`
- `tareas/page.tsx`
- `str-housekeeping/page.tsx`

Recomendaciones Principales

1. Falta de elemento `<main>`:

- `sidebar.tsx`
- `header.tsx`
- `layout.tsx`
- Múltiples páginas

2. Falta de skip links:

- Todas las páginas principales requieren implementación

Puntuación: 0/100  Crítico - Requiere corrección inmediata

Puntuación Global de Accesibilidad

Promedio General: 7.5/100 

Categoría	Puntuación	Estado	Prioridad
Navegación por Teclado	15/100	 Crítico	 Alta
Lectores de Pantalla	0/100	 Crítico	 Alta
axe DevTools	Pendiente*		 Alta

*Requiere servidor de desarrollo corriendo

Plan de Acción Correctiva

Prioridad ALTA (Inmediato)

1. Corrección de Inputs Sin Etiquetas

Impacto: Crítico para lectores de pantalla

Archivos a corregir:

- components/wizards/SetupWizard.tsx
- components/ui/photo-gallery.tsx
- components/ui/**input**.tsx
- components/operador/MobilePhotoCapture.tsx
- app/valoraciones/page.tsx
- app/room-rental/[unitId]/page.tsx
- app/publicaciones/page.tsx
- app/notificaciones/page.tsx

Solución:

```
//  Antes
<input type="text" />

//  Despues
<input
  type="text"
  aria-label="Descripción del campo"
/>
// O mejor aún:
<label htmlFor="campo-id">Etiqueta</label>
<input id="campo-id" type="text" />
```

2. Implementar Elemento `<main>` en Layouts

Impacto: Crítico para navegación de lectores de pantalla

Archivos a corregir:

- components/layout/sidebar.tsx
- components/layout/header.tsx
- app/layout.tsx

Solución:

```
// ✗ Antes
<div className="content">
  {children}
</div>

// ✓ Después
<main className="content" role="main">
  {children}
</main>
```

3. Agregar Skip Links Globalmente

Impacto: Alto para navegación por teclado

Ubicación: app/layout.tsx o componente de layout principal

Implementación:

```
const SkipLink = () => (
  <a
    href="#main-content"
    className="skip-link sr-only focus:not-sr-only focus:absolute focus:top-4 fo-
    cus:left-4 focus:z-50 focus:bg-primary focus:text-primary-foreground focus:px-4
    focus:py-2 focus:rounded"
    >
    Saltar al contenido principal
  </a>
);

// CSS en globals.css
.skip-link {
  position: absolute;
  left: -9999px;
}

.skip-link:focus {
  left: 0;
  top: 0;
  z-index: 9999;
}
```

● Prioridad MEDIA

4. Mejorar Focus-Visible en Componentes UI

Componentes afectados (30+):

- LanguageSelector.tsx
- touch-optimized-button.tsx
- lazy-tabs.tsx

- icon-button.tsx
- Y más...

Solución generalizada:

```
// Agregar a className
className="focus:outline-none focus:ring-2 focus:ring-primary focus:ring-offset-2"
```

5. Agregar Roles ARIA a Componentes Custom

Componentes prioritarios:

- Dialogs/Modals
- Dropdowns
- Tabs
- Accordions
- Menús

Ejemplo:

```
<div
  role="dialog"
  aria-modal="true"
  aria-labelledby="dialog-title"
  aria-describedby="dialog-description"
>
  <h2 id="dialog-title">Título del Modal</h2>
  <p id="dialog-description">Descripción</p>
  {/* contenido */}
</div>
```

6. Corregir Jerarquía de Encabezados

Páginas afectadas:

- votaciones/page.tsx
- valoraciones/page.tsx
- unidades/page.tsx
- tareas/page.tsx
- str-housekeeping/page.tsx

Solución:

```
// ❌ Antes
<h1>Título Principal</h1>
<h3>Subtítulo</h3> // ❌ Salto de h1 a h3

// ✅ Despues
<h1>Título Principal</h1>
<h2>Subtítulo</h2> // ✅ Jerarquía correcta
```

🟡 Prioridad BAJA (Optimización)

7. Agregar Manejadores de Teclado

Componentes interactivos custom:

```

const handleKeyDown = (e: React.KeyboardEvent) => {
  switch(e.key) {
    case 'Enter':
    case ' ': // Espacio
      e.preventDefault();
      onClick();
      break;
    case 'Escape':
      onClose();
      break;
  }
};

<div
  role="button"
  tabIndex={0}
  onKeyDown={handleKeyDown}
  onClick={onClick}
>
```

8. Implementar Regiones Live para Notificaciones

```

<div
  aria-live="polite"
  aria-atomic="true"
  className="sr-only"
>
  {notificationMessage}
</div>
```

🧪 Pruebas Pendientes

axe DevTools (Requiere servidor dev)

Para ejecutar:

```

# Terminal 1: Iniciar servidor
cd /home/ubuntu/homming_vidaro/nextjs_space
yarn dev

# Terminal 2: Ejecutar pruebas axe
node scripts/test-accessibility.js
```

Páginas a probar:

- Landing Page (/landing)
- Login/Register (/login , /register)
- Dashboard (/dashboard)
- Módulos principales:
 - Edificios (/edificios)
 - Unidades (/unidades)
 - Inquilinos (/inquilinos)
 - Contratos (/contratos)

Criterios evaluados:

- WCAG 2.1 Level A
 - WCAG 2.1 Level AA
 - Best practices
-

Reportes Generados

Ubicación de Archivos

```
/home/ubuntu/homming_vidaro/nextjs_space/
├── keyboard-navigation-report.json      # Análisis de navegación por teclado
├── screen-reader-report.json          # Análisis de lectores de pantalla
└── accessibility-full-report.json     # Reporte consolidado
└── accessibility-report.json          # (Pendiente) Reporte axe DevTools
```

Estructura de Reportes

```
{
  "timestamp": "2025-12-07T...",
  "summary": {
    "overallScore": 7.5,
    "totalCriticalIssues": 160,
    "testsRun": 2,
    "testsPassed": 0,
    "testsFailed": 2
  },
  "tests": {
    "keyboard": { "score": 15, "status": "fail" },
    "screenReader": { "score": 0, "status": "fail" }
  }
}
```

Próximos Pasos

Inmediatos (Esta semana)

- [] Corregir inputs sin etiquetas (todos los archivos identificados)
- [] Implementar `<main>` en layouts principales
- [] Agregar skip links globalmente
- [] Ejecutar pruebas axe DevTools con servidor dev

Corto Plazo (Próximas 2 semanas)

- [] Mejorar focus-visible en top 20 componentes UI
- [] Agregar roles ARIA en componentes custom críticos
- [] Corregir jerarquía de encabezados en todas las páginas
- [] Re-ejecutar suite completa de pruebas

Mediano Plazo (Próximo mes)

- [] Implementar manejadores de teclado en componentes interactivos

- [] Agregar regiones live para notificaciones y actualizaciones
 - [] Optimizar tabindex en flujos complejos
 - [] Documentación de patrones de accesibilidad para el equipo
 - [] Certificación WCAG 2.1 AA
-

Recursos y Referencias

Documentación

- [WCAG 2.1 Guidelines](https://www.w3.org/WAI/WCAG21/quickref/) (<https://www.w3.org/WAI/WCAG21/quickref/>)
- [axe DevTools Documentation](https://www.deque.com/axe/devtools/) (<https://www.deque.com/axe/devtools/>)
- [ARIA Authoring Practices Guide](https://www.w3.org/WAI/ARIA/apg/) (<https://www.w3.org/WAI/ARIA/apg/>)
- [MDN Web Accessibility](https://developer.mozilla.org/en-US/docs/Web/Accessibility) (<https://developer.mozilla.org/en-US/docs/Web/Accessibility>)

Herramientas Adicionales

- **Navegadores:**
- Chrome DevTools Lighthouse
- Firefox Accessibility Inspector
- **Lectores de Pantalla:**
- NVDA (Windows) - Gratuito
- JAWS (Windows) - Comercial
- VoiceOver (macOS/iOS) - Integrado
- TalkBack (Android) - Integrado

Checklist de Accesibilidad

Teclado

- [] Todos los elementos interactivos son accesibles por teclado
- [] El orden de tabulación es lógico
- [] El foco es visible en todo momento
- [] No hay trampas de teclado
- [] Skip links funcionan correctamente

Lectores de Pantalla

- [] Todas las imágenes tienen texto alternativo
- [] Los formularios tienen etiquetas asociadas
- [] La estructura de encabezados es lógica
- [] Los landmarks están implementados (`<main>` , `<nav>` , etc.)
- [] Los cambios dinámicos se anuncian con `aria-live`

Visual

- [] Contraste de color suficiente (4.5:1 para texto normal)
- [] El texto puede aumentarse al 200% sin pérdida de contenido
- [] No se depende únicamente del color para transmitir información

Interacción

- [] Los objetivos táctiles son al menos 44x44 píxeles

- [] Los mensajes de error son claros y accesibles
- [] Se proporciona tiempo suficiente para completar tareas

Métricas de Seguimiento

Baseline Actual

Métrica	Valor Actual	Objetivo	Estado
Puntuación Global	7.5/100	90/100	X 8%
Navegación Teclado	15/100	90/100	X 17%
Lectores de Pantalla	0/100	90/100	X 0%
Inputs con Label	~85%	100%	! 85%
Componentes con Focus Visible	~38%	100%	! 38%
Páginas con Skip Links	~3%	100%	X 3%
Jerarquía Encabezados	~92%	100%	! 92%

Objetivo Post-Correcciones

Métrica	Objetivo Corto Plazo	Objetivo Final
Puntuación Global	70/100	90/100
Navegación Teclado	75/100	95/100
Lectores de Pantalla	80/100	95/100
Inputs con Label	100%	100%
Componentes con Focus Visible	90%	100%
Páginas con Skip Links	100%	100%

Conclusión

Se ha completado exitosamente la **Fase 3: Validación de Accesibilidad** con la implementación de herramientas automatizadas y análisis estático del código.

Hallazgos Principales

1. **Estado crítico** en accesibilidad actual (7.5/100)
2. **150+ problemas críticos** identificados en navegación por teclado
3. **Inputs sin etiquetas** en 10+ archivos principales
4. **Falta de elementos semánticos** (`<main>`, skip links)
5. **Jerarquía de encabezados** con saltos en 5 páginas

Oportunidades de Mejora

-  **Buena base** de elementos semánticos (251 elementos)
-  **Excelente uso** de encabezados (639 elementos)
-  **Componentes UI** requieren mejoras en focus-visible
-  **Inputs** necesitan etiquetas accesibles urgentemente

Impacto Estimado Post-Correcciones

Al implementar las correcciones de **Prioridad ALTA**, se espera:

- Incremento de 7.5 → **~70 puntos** en puntuación global
 - **100%** de inputs con etiquetas accesibles
 - **100%** de páginas con skip links
 - **Cumplimiento básico** de WCAG 2.1 A/AA
-

Fecha de Reporte: 7 de diciembre de 2025

Próxima Revisión: Post-implementación de correcciones

Responsable: Equipo de Desarrollo INMOVA