



Guía de Navegación EscalaFin



Resumen

EscalaFin implementa un sistema de navegación dual optimizado para diferentes tipos de dispositivos y experiencias de usuario:

- **Desktop:** Sidebar navegacional sticky persistente
- **Mobile:** Header con menú hamburguesa + Sheet drawer deslizante
- **PWA:** Módulos dinámicos basados en permisos de usuario



Arquitectura de Navegación

Componentes Principales

```
// Estructura general del sistema
<LayoutProvider>
  { /* Visible solo en desktop (≥768px) */ }
  <Sidebar />
  <Header />

  { /* Visible solo en mobile (<768px) */ }
  <HeaderMobile />

  <MainContent>
    {children}
  </MainContent>
</LayoutProvider>
```



Archivos Clave

Componente	Archivo	Responsabilidad
LayoutProvider	/components/layout/layout-provider.tsx	Orchestrador principal de navegación
Sidebar	/components/layout/side-bar.tsx	Navegación desktop sticky
HeaderMobile	/components/layout/header-mobile.tsx	Navegación móvil optimizada
Header	/components/layout/header.tsx	Header desktop minimalista
Sheet	/components/ui/sheet.tsx	Component drawer para mobile

Navegación Desktop (Sidebar)

Características

- 📌 **Sticky positioning** - Siempre visible durante scroll
- 🔄 **Colapsible/expandible** - Usuario puede alternar tamaño
- 📁 **Categorización** - Módulos organizados por función
- 🎯 **Active state** - Indicador visual de página actual
- 👤 **User info** - Avatar y datos del usuario en footer
- ⚡ **Transitions** - Animaciones suaves entre estados

Estados del Sidebar

1. Expandido (Default - w-64)

```
// Ancho completo con iconos + etiquetas
<div className="w-64">
  <Icon />
  <span>Nombre del Módulo</span>
</div>
```

2. Colapsado (w-16)

```
// Solo iconos visibles
<div className="w-16">
  <Icon />
  { /* span oculto */ }
</div>
```

3. Oculto (< 768px)

```
// Completamente oculto en mobile
<div className="hidden md:flex">
  { /* Sidebar content */ }
</div>
```

Estructura de Módulos

```
const navigationItems = [
  {
    category: 'Principal',
    items: [
      {
        title: 'Dashboard',
        icon: LayoutDashboard,
        href: '/admin/dashboard', // Dynamic basado en rol
      }
    ]
  },
  {
    category: 'Gestión',
    items: [
      {
        title: 'Clientes',
        icon: Users,
        href: '/clients',
        moduleKey: 'client_list',
        roles: ['ADMIN', 'ASESOR']
      },
      // ... más módulos
    ]
  }
  // ... más categorías
];
```



Navegación Móvil

Características

- 🍔 **Hamburger menu** - Icono universal de menú
- 📱 **Sheet/Drawer** - Panel deslizante desde la derecha
- 👉 **Touch-optimized** - Botones y áreas de toque ampliadas
- 📋 **Complete navigation** - Acceso a todos los módulos
- 👤 **User profile** - Info del usuario prominente
- ⚡ **Smooth animations** - Transiciones fluidas

Componentes

HeaderMobile

```
<header className="md:hidden sticky top-0 z-50">
  <div className="flex items-center justify-between">
    <Logo />
    <Sheet>
      <SheetTrigger>
        <Menu />
      </SheetTrigger>
      <SheetContent>
        <Navigation />
      </SheetContent>
    </Sheet>
  </div>
</header>
```

Navigation Drawer

```
<SheetContent side="right" className="w-80">
  <UserInfo />
  <NavigationList />
  <ActionButtons />
</SheetContent>
```



Sistema de Permisos y Filtrado

Lógica de Filtrado

```
const getFilteredItems = (items: NavigationItem[]) => {
  return items.filter(item => {
    // 1. Verificar rol del usuario
    if (item.roles && !item.roles.includes(userRole)) {
      return false;
    }

    // 2. Verificar módulo habilitado en PWA
    if (item.moduleKey) {
      return modules.some(module => module.moduleKey === item.moduleKey);
    }

    return true;
  });
};
```

Roles y Permisos

Rol	Acceso	Módulos
ADMIN	 Completo	Todos los módulos + configuración
ASESOR	 Operativo	Gestión, reportes, comunicación
CLIENTE	 Limitado	Dashboard personal, préstamos propios

Verificación PWA

```
-- Verificar módulos habilitados para usuario
SELECT m.*, rp.*
FROM "PWAModule" m
JOIN "ModuleRolePermission" rp ON m.id = rp."moduleId"
WHERE m.status = 'ENABLED'
AND rp.role = 'ADMIN'
AND rp.enabled = true;
```



Responsive Design

Breakpoints Tailwind

```
/* Mobile first approach */
sm: 640px /* Small devices */
md: 768px /* Tablets - PUNTO DE CAMBIO PRINCIPAL */
lg: 1024px /* Laptops */
xl: 1280px /* Desktops */
2xl: 1536px /* Large screens */
```

Estrategia Responsive

```
// Componente que muestra diferente UI según breakpoint
<div className="navigation">
  {/* Desktop: Sidebar siempre visible */}
  <div className="hidden md:block">
    <Sidebar />
  </div>

  {/* Mobile: Header con menú hamburguesa */}
  <div className="md:hidden">
    <HeaderMobile />
  </div>
</div>
```

Estilos y Animaciones

Transiciones CSS

```
/* Sidebar collapse/expand animation */
.sidebar {
  transition: width 300ms cubic-bezier(0.4, 0, 0.2, 1);
}

/* Navigation item hover effects */
.nav-item {
  transition: all 200ms ease-in-out;
}

.nav-item:hover {
  background-color: theme('colors.primary.50');
  transform: translateX(4px);
}

/* Sheet slide animation */
.sheet-content {
  animation: slideIn 250ms ease-out;
}

@keyframes slideIn {
  from {
    transform: translateX(100%);
  }
  to {
    transform: translateX(0);
  }
}
```

Estados Visuales

```
// Active page indicator
<Button
  className={cn(
    'navigation-item',
    isActive && 'bg-primary/10 text-primary border-primary/20'
  )}
>
  <Icon />
  <span>{title}</span>
</Button>
```

Performance Optimizations

Lazy Loading

```
// Cargar módulos de navegación dinámicamente
const NavigationItem = lazy(() => import('./NavigationItem'));

// Suspense boundary
<Suspense fallback={<NavigationSkeleton />}>
  <NavigationItem {...props} />
</Suspense>
```

Memoización

```
// Memorizar items filtrados para evitar re-renders
const filteredItems = useMemo(() => {
  return getFilteredItems(navigationItems);
}, [navigationItems, userRole, modules]);

// Memorizar estado activo
const isActive = useCallback((href: string) => {
  return pathname.startsWith(href);
}, [pathname]);
```

Virtual Scrolling

```
// Para listas muy largas de módulos (futuro)
import { FixedSizeList as List } from 'react-window';

<List
  height={400}
  itemCount={filteredItems.length}
  itemSize={48}
  itemData={filteredItems}
>
  {NavigationItem}
</List>
```

Testing de Navegación

Tests Unitarios

```
// sidebar.test.tsx
describe('Sidebar Navigation', () => {
  test('shows correct modules for ADMIN role', () => {
    render(<Sidebar />, {
      wrapper: createAuthWrapper({ role: 'ADMIN' })
    });

    expect(screen.getByText('Gestionar Usuarios')).toBeInTheDocument();
  });

  test('collapses when toggle button clicked', () => {
    render(<Sidebar />);

    fireEvent.click(screen.getByRole('button', { name: /collapse/i }));
    expect(screen.getByTestId('sidebar')).toHaveClass('w-16');
  });
});
```

Tests de Integración

```
// navigation-flow.test.tsx
describe('Navigation Flow', () => {
  test('desktop to mobile navigation consistency', () => {
    const { rerender } = render(<App />);

    // Desktop navigation
    expect(screen.getByTestId('sidebar')).toBeVisible();
    expect(screen.queryByTestId('mobile-header')).not.toBeVisible();

    // Switch to mobile viewport
    Object.defineProperty(window, 'innerWidth', { value: 600 });
    rerender(<App />);

    expect(screen.queryByTestId('sidebar')).not.toBeVisible();
    expect(screen.getByTestId('mobile-header')).toBeVisible();
  });
});
```


Tests Visuales

```
// visual-regression.test.tsx
describe('Navigation Visual Tests', () => {
  test('sidebar renders correctly', async () => {
    const component = render(<Sidebar />);
    const image = await percySnapshot('Sidebar - Expanded');
    expect(image).toMatchSnapshot();
  });

  test('mobile navigation renders correctly', async () => {
    const component = render(<HeaderMobile />);
    const image = await percySnapshot('Mobile Navigation - Sheet Open');
    expect(image).toMatchSnapshot();
  });
});
```



Troubleshooting Common Issues

Sidebar no aparece

```
// Debug checklist
const debugSidebar = () => {
  // 1. Verificar breakpoint
  console.log('Window width:', window.innerWidth); // Should be ≥ 768px

  // 2. Verificar sesión
  console.log('User session:', session); // Should exist

  // 3. Verificar módulos
  console.log('Available modules:', modules); // Should not be empty

  // 4. Verificar CSS
  const sidebar = document.querySelector('[data-testid="sidebar"]');
  console.log('Sidebar styles:', window.getComputedStyle(sidebar));
};
```

Módulos no se filtran correctamente

```
-- Verificar en base de datos
-- 1. Módulo existe y está habilitado
SELECT * FROM "PWAModule" WHERE "moduleKey" = 'client_list';

-- 2. Permisos por rol configurados
SELECT * FROM "ModuleRolePermission"
WHERE "moduleId" IN (
  SELECT id FROM "PWAModule" WHERE "moduleKey" = 'client_list'
) AND role = 'ADMIN';

-- 3. Usuario tiene el rol correcto
SELECT role FROM "User" WHERE email = 'admin@escalafin.com';
```

Performance lenta en navegación

```
// Profiling de performance
import { Profiler } from 'react';

<Profiler id="Navigation" onRender={({id, phase, actualDuration}) => {
  console.log('Navigation render:', { id, phase, actualDuration });
}}>
  <NavigationComponent />
</Profiler>
```



Migración de Navegación Legacy

Pasos para migrar navegación existente

1. **Backup** de componentes actuales
2. **Instalar** nuevos componentes de layout
3. **Configurar** LayoutProvider en layout raíz
4. **Migrar** rutas y enlaces existentes
5. **Testing** completo en todos los breakpoints
6. **Deploy** gradual con feature flags

Script de migración

```
#!/bin/bash
# migrate-navigation.sh

echo "🔄 Iniciando migración de navegación..."

# Backup de componentes existentes
mkdir -p backup/components/navigation
cp -r components/navigation/* backup/components/navigation/

# Instalar nuevos componentes
echo "📦 Instalando nuevos componentes..."
# (Los nuevos componentes ya están instalados)

# Actualizar layout raíz
echo "🔧 Actualizando layout..."
# (Ya actualizado en app/layout.tsx)

# Ejecutar tests
echo "🧪 Ejecutando tests de navegación..."
yarn test --testPathPattern=navigation

echo "✅ Migración completada!"
```

Métricas y Analytics

KPIs de Navegación

```
// Métricas a monitorear
const navigationMetrics = {
  // Performance
  sidebarRenderTime: '< 100ms',
  mobileMenuOpenTime: '< 50ms',
  moduleFilterTime: '< 200ms',

  // Usage
  sidebarCollapseRate: '15%', // Usuarios que colapsan sidebar
  mobileMenuUsage: '85%',    // Usuarios que usan menú móvil
  moduleClickthrough: '92%', // Navegación exitosa a módulos

  // Errors
  navigationErrors: '< 0.1%', // Errores de navegación
  moduleLoadErrors: '< 0.5%', // Errores de carga de módulos
};
```

Tracking de Eventos

```
// analytics.ts
export const trackNavigation = (event: string, properties: object) => {
  // Google Analytics 4
  gtag('event', event, {
    event_category: 'navigation',
    ...properties
  });

  // Custom analytics
  analytics.track(event, {
    category: 'navigation',
    timestamp: new Date().toISOString(),
    ...properties
  });
};

// Uso en componentes
const handleSidebarToggle = () => {
  setCollapsed(!collapsed);
  trackNavigation('sidebar_toggle', {
    action: collapsed ? 'expand' : 'collapse',
    userRole: session?.user?.role
  });
};
```

Futuras Mejoras

Roadmap de Navegación

v2.2.0

- 🔍 **Búsqueda** en sidebar con filtrado inteligente
- 📌 **Bookmarks** módulos favoritos del usuario

- 🎨 **Temas personalizables** colores de navegación
- ⌨️ **Keyboard shortcuts** navegación por teclado

v2.3.0

- 🤖 **AI-powered** sugerencias de navegación
- 📊 **Usage analytics** dashboard interno
- 🎯 **Context-aware** navigation basada en flujo de trabajo
- 📱 **Gestures support** swipe navigation en mobile

v3.0.0

- 🌐 **Multi-tenant** navegación personalizada por organización
- 🔧 **Custom modules** capacidad de agregar módulos externos
- 🎨 **Visual builder** editor drag-and-drop para navegación
- 📈 **A/B testing** framework para optimizar UX

Documentación actualizada: Septiembre 2025
EscalaFin v2.1.0 - Navigation System