Guía de Migración a Mobile-First UI

📱 Fase 3: Mobile-First UI - Implementación Completa

🎯 Objetivo

Transformar INMOVA en una plataforma completamente optimizada para dispositivos móviles, manteniendo la experiencia de escritorio existente.

📦 Componentes Implementados

1. Hooks de Detección de Dispositivos

`useMediaQuery` - Hook genérico para media queries

```
import { useMediaQuery } from '@/lib/hooks/useMediaQuery';

function MyComponent() {
  const isLargeScreen = useMediaQuery('(min-width: 1024px)');

  return (
    <div>
      {isLargeScreen ? 'Desktop View' : 'Mobile View'}
    </div>
  );
}
```

`useIsMobile`, `useIsTablet`, `useIsDesktop`

```
import { useIsMobile, useIsTablet, useIsDesktop } from '@/lib/hooks/useMediaQuery';

function ResponsiveComponent() {
  const isMobile = useIsMobile();     // < 768px
  const isTablet = useIsTablet();     // 769px - 1024px
  const isDesktop = useIsDesktop();   // > 1025px

  if (isMobile) return <MobileView />;
  if (isTablet) return <TabletView />;
  return <DesktopView />;
}
```

`useDeviceType` - Detección simplificada

```
import { useDeviceType } from '@/lib/hooks/useMediaQuery';

function AdaptiveComponent() {
  const deviceType = useDeviceType(); // 'mobile' | 'tablet' | 'desktop'

  return <div>Dispositivo actual: {deviceType}</div>;
}
```

2. Hooks de Gestos

`useSwipe` - Detectar gestos de deslizamiento

```
import { useSwipe } from '@/lib/hooks/useGestures';

function SwipeableCard() {
  const swipeHandlers = useSwipe({
    onSwipeLeft: () => console.log('Deslizado a la izquierda'),
    onSwipeRight: () => console.log('Deslizado a la derecha'),
    threshold: 50, // Mínimo de píxeles para activar
  });

  return (
    <div {...swipeHandlers} className="swipeable">
      Desliza para interactuar
    </div>
  );
}
```

`usePullToRefresh` - Pull-to-refresh nativo

```
import { usePullToRefresh } from '@/lib/hooks/useGestures';

function RefreshableList() {
  const handleRefresh = async () => {
    await fetchData();
  };

  usePullToRefresh({
    onRefresh: handleRefresh,
    threshold: 80,
    enabled: true,
  });

  return <div>Lista de elementos...</div>;
}
```

3. Bottom Navigation

Navegación inferior automática para móviles.

Uso: Se incluye automáticamente en `AuthenticatedLayout`. No requiere configuración adicional.

Características:
- Sólo visible en móviles (< 768px)
- 4 accesos rápidos principales: Inicio, Edificios, Inquilinos, Pagos
- Botón de menú para acceder al sidebar completo
- Indicadores de página activa
- Soporte para badges de notificaciones

4. ResponsiveTable - Tablas adaptativas

Transforma automáticamente tablas en cards en móvil.

```
import { ResponsiveTable } from '@/components/ui/responsive-table';

function BuildingsPage() {
  const buildings = [
    { id: 1, name: 'Edificio A', address: 'Calle Principal 123', units: 10 },
    { id: 2, name: 'Edificio B', address: 'Avenida Central 456', units: 8 },
  ];

  return (
    <ResponsiveTable
      data={buildings}
      columns={[
        {
          key: 'name',
          header: 'Nombre',
          mobileLabel: 'Edificio'
        },
        {
          key: 'address',
          header: 'Dirección',
          hideOnMobile: true, // Ocultar en móvil
        },
        {
          key: 'units',
          header: 'Unidades',
          render: (building) => `${building.units} unidades`,
        },
      ]}
      keyExtractor={(building) => building.id}
      onRowClick={(building) => router.push(`/edificios/${building.id}`)}
      emptyMessage="No hay edificios disponibles"
    />
  );
}
```

Comportamiento:
- **Desktop:** Tabla tradicional con todas las columnas
- **Móvil:** Cards con información vertical

---

5. MobileOptimizedForm - Formularios optimizados

Formularios con mejor UX en móvil.

```tsx
import {
  MobileOptimizedForm,
  FormSection,
  FormField
} from '@/components/ui/mobile-optimized-form';
import { Input } from '@/components/ui/input';

function CreateBuildingForm() {
  const [loading, setLoading] = useState(false);

  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    setLoading(true);
    // ... lógica de envío
    setLoading(false);
  };

  return (
    <MobileOptimizedForm
      onSubmit={handleSubmit}
      title="Nuevo Edificio"
      description="Completa la información del edificio"
      submitLabel="Crear Edificio"
      cancelLabel="Cancelar"
      onCancel={() => router.back()}
      loading={loading}
    >
      <FormSection
        title="Información Básica"
        description="Datos principales del edificio"
      >
        <FormField
          label="Nombre"
          required
          hint="Nombre identificativo del edificio"
        >
          <Input name="name" placeholder="Ej: Edificio Central" />
        </FormField>

        <FormField
          label="Dirección"
          required
        >
          <Input name="address" placeholder="Calle, número, ciudad" />
        </FormField>
      </FormSection>

      <FormSection title="Detalles">
        <FormField label="Número de unidades">
          <Input type="number" name="units" placeholder="10" />
        </FormField>
      </FormSection>
    </MobileOptimizedForm>
  );
}
```

Características:

- Botones fijos en la parte inferior en móvil
- Scroll optimizado del contenido
- Secciones colapsables

- Mejor espaciado y tipografía
- Estados de carga integrados

---

6. AuthenticatedLayout - Layout unificado

Layout completo con toda la navegación optimizada.

```tsx
import { AuthenticatedLayout } from '@/components/layout/authenticated-layout';

export default function EdificiosPage() {
  return (
    <AuthenticatedLayout maxWidth="7xl">
      <div className="space-y-6">
        <h1 className="text-3xl font-bold">Mis Edificios</h1>
        {/* ... contenido ... */}
      </div>
    </AuthenticatedLayout>
  );
}
```

Props:

- `maxWidth`: `'full' | '7xl' | '6xl' | '5xl' | '4xl'` - Ancho máximo del contenedor
- `className`: Clases adicionales para el área de scroll
- `containerClassName`: Clases para el contenedor interno

Incluye automáticamente:

- Sidebar (desktop)
- Header (todas las pantallas)
- Bottom Navigation (móvil)
- Espaciado apropiado para cada dispositivo

---

7. PullToRefresh - Actualizar arrastrando

Componente para actualizar contenido con gesto de arrastre.

```tsx
import { PullToRefresh } from '@/components/ui/pull-to-refresh';

function InquilinosPage() {
  const [inquilinos, setInquilinos] = useState([]);

  const handleRefresh = async () => {
    const response = await fetch('/api/inquilinos');
    const data = await response.json();
    setInquilinos(data);
  };

  return (
    <AuthenticatedLayout>
      <PullToRefresh onRefresh={handleRefresh}>
        <div className="space-y-4">
          {inquilinos.map((inquilino) => (
            <InquilinoCard key={inquilino.id} data={inquilino} />
          ))}
        </div>
      </PullToRefresh>
    </AuthenticatedLayout>
  );
}
```

🚀 Migración de Páginas Existentes

Antes (Patrón Antiguo)

```tsx
'use client';

import { Sidebar } from '@/components/layout/sidebar';
import { Header } from '@/components/layout/header';

export default function MiPagina() {
  return (
    <div className="flex h-screen">
      <Sidebar />
      <div className="flex-1 flex flex-col">
        <Header />
        <main className="flex-1 overflow-auto p-6">
          {/* Contenido */}
        </main>
      </div>
    </div>
  );
}
```

Después (Mobile-First)

```
'use client';

import { AuthenticatedLayout } from '@/components/layout/authenticated-layout';

export default function MiPagina() {
  return (
    <AuthenticatedLayout maxWidth="7xl">
      {/* Contenido */}
    </AuthenticatedLayout>
  );
}
```

Beneficios:
- Código más limpio y mantenible
- Bottom Navigation automática en móvil
- Espaciado y padding optimizados
- Soporte para safe areas (notch, etc.)

🎨 Clases CSS Utility Nuevas

Touch Targets

```
<!-- Asegurar áreas táctiles mínimas (44x44px) -->
<button className="touch-target">Botón accesible</button>
```

Feedback Táctil

```
<!-- Feedback visual al tocar -->
<div className="touch-feedback">
  <Card>...</Card>
</div>
```

Visibilidad Condicional

```
<!-- Solo visible en móvil -->
<div className="mobile-only">
  Contenido móvil
</div>

<!-- Solo visible en desktop -->
<div className="desktop-only">
  Contenido desktop
</div>
```

Safe Areas

```
<!-- Respetar áreas seguras (notch, etc.) -->
<div className="safe-bottom safe-top">
  Contenido con padding seguro
</div>
```

Gestos

```
<!-- Habilitar gestos -->
<div className="gesture-enabled swipeable">
  Desliza para interactuar
</div>
```

Grid Responsive

```
<!-- Grid que se adapta automáticamente -->
<div className="mobile-grid">
  <Card>Item 1</Card>
  <Card>Item 2</Card>
  <Card>Item 3</Card>
</div>
```

✅ Checklist de Migración

Para cada página:

- [ ] Reemplazar layout manual con `AuthenticatedLayout`
- [ ] Convertir tablas grandes usando `ResponsiveTable`
- [ ] Migrar formularios a `MobileOptimizedForm`
- [ ] Añadir `PullToRefresh` en listas y feeds
- [ ] Probar en dispositivos móviles reales
- [ ] Verificar touch targets (mínimo 44x44px)
- [ ] Validar contraste y legibilidad
- [ ] Comprobar safe areas en dispositivos con notch

🔧 Mejores Prácticas

1. Diseñar Mobile-First

```
// ✅ Bueno: Empezar con móvil y expandir
const buttonSize = isMobile ? 'default' : 'lg';

// ❌ Evitar: Desktop-first
const buttonSize = isDesktop ? 'lg' : 'default';
```

2. Touch Targets Adecuados

```
// ✅ Bueno: Áreas táctiles grandes
<Button className="h-12 px-6">Acción</Button>

// ❌ Evitar: Botones muy pequeños
<Button className="h-6 px-2 text-xs">Acción</Button>
```

3. Evitar Hover en Móvil

```
// ✅ Bueno: Usar estados activos
className="active:bg-accent"

// ❌ Evitar: Hover en touch devices
className="hover:bg-accent"
```

4. Optimizar Animaciones

```
// ✅ Bueno: Animaciones simples en móvil
const animationClass = isMobile
  ? 'transition-opacity duration-200'
  : 'transition-all duration-300';
```

5. Prevenir Zoom No Deseado

```
<!-- Los inputs con font-size >= 16px previenen zoom en iOS -->
<Input className="text-base" />
```

📊 Mejoras de Performance

Optimizaciones Automáticas:

1. **Animaciones reducidas** en móvil (200ms vs 300ms)
2. **Blur y shadows simplificados** para mejor rendimiento
3. **Scroll nativo optimizado** con `-webkit-overflow-scrolling: touch`
4. **Prevención de zoom** en inputs (font-size: 16px mínimo)
5. **Overscroll behavior** contenido para mejor UX

📝 Notas Importantes

Breakpoints:

- **Mobile:** < 768px
- **Tablet:** 769px - 1024px
- **Desktop:** > 1025px

Safe Areas:

Los componentes respetan automáticamente las safe areas de dispositivos modernos (iPhone con notch, etc.)

Testing:

Prueba siempre en:
- Chrome DevTools (modo responsive)
- Dispositivos reales (iOS y Android)
- Diferentes tamaños de pantalla

🚀 Próximos Pasos

1. Migrar páginas principales: `/edificios`, `/inquilinos`, `/pagos`, `/contratos`
2. Añadir gestos avanzados (pinch-to-zoom, long-press)
3. Implementar vibración háptica en acciones críticas
4. Crear variantes de componentes específicas para tablet
5. Optimizar imágenes y assets para móvil

👥 Soporte

Para dudas o problemas con la implementación mobile-first:
- Revisar esta guía
- Consultar ejemplos en componentes existentes
- Contactar al equipo de desarrollo

Fecha de implementación: Diciembre 2024
Versión: 1.0.0
Estado: ✅ Fase 3 Completa