Boxful App - Prueba Técnica

Aplicación móvil desarrollada con React Native y Expo, siguiendo el diseño proporcionado en Figma. Este proyecto demuestra la implementación de navegación, animaciones fluidas y una experiencia de usuario pulida.

Tecnologías Utilizadas

Tecnología	Versión	Descripción
React Native	0.81.5	Framework base para desarrollo móvil
Expo	54.0.31	Plataforma de desarrollo y build
Expo Router	6.0.21	Navegación basada en archivos
React Native Reanimated	4.1.1	Animaciones de alto rendimiento
NativeWind	4.2.1	Tailwind CSS para React Native
Zod	4.3.5	Validación de esquemas TypeScript-first
TypeScript	5.9.2	Tipado estático

Funcionalidades Implementadas

1. Splash Screen

- Configurado mediante `expo-splash-screen`
- Transición suave al cargar la aplicación
- Soporte para modo claro y oscuro

2. Onboarding

- Carrusel horizontal con 3 slides
- Animación de flotación continua en las imágenes
- Indicadores de paginación animados
- Botón "Iniciar" con aparición animada en el último slide
- Animación de salida al navegar al login (fade + translate)
- Funcionalidad para mostrar el onboarding una sola vez (Desactivada para la demo)

3. Login

- Validación de formulario con Zod v4
- Errores específicos por campo
- Estado de carga (loading) en el botón
- Animaciones de entrada escalonadas (staggered animations)
- Soporte para modo oscuro/claro

4. Tab Navigation

- 5 tabs: Delivery, Historial, Analíticas, Facturación, Cuenta
- Iconos personalizados con `iconsax-react-nativejs`
- Animaciones de transición entre tabs (slide horizontal)
- Feedback háptico en iOS al cambiar de tab

5. Animaciones (React Native Reanimated)

- **Onboarding**: Flotación continua, transiciones de slide, fade de salida
- **Login**: Entrada escalonada de elementos (logo, header, formulario, botones)
- **Tabs**: Animación slide horizontal entre pantallas
- **AlertModal**: Entrada con scale/translate, salida con secuencia de animaciones
- **Imágenes**: Optimizadas con `expo-image` (cache, priority, transitions)

6. Componentes UI Reutilizables

- `Button` : Variantes (primary/secondary), estados (loading/disabled), iconos
- `Input` : Labels, errores, toggle de contraseña
- `Text` : Variantes de color, tamaños, pesos tipográficos
- `Header` : Navegación con notificaciones y selector de país
- `AlertModal` : Alertas tipo SweetAlert con múltiples tipos (success, error, warning, info)
- `AnimatedTabScreen` : Wrapper para animaciones entre tabs

Estructura del Proyecto

```
boxful-app/
├── app/                          # Rutas (file-based routing)
│   ├── (public)/                 # Rutas públicas (sin autenticación)
│   │   ├── _layout.tsx
│   │   ├── login.tsx
│   │   └── onboarding.tsx
│   ├── (tabs)/                   # Tab navigation
│   │   ├── _layout.tsx
│   │   ├── delivery.tsx
│   │   ├── history.tsx
│   │   ├── analitics.tsx
│   │   ├── facturacion.tsx
│   │   └── cuenta.tsx
│   ├── _layout.tsx               # Layout raíz
│   └── index.tsx                 # Punto de entrada
├── assets/
│   ├── icons/                    # Iconos SVG
│   └── images/                   # Imágenes y assets
│       └── onboarding/           # Assets del onboarding
├── components/
│   ├── Home/                     # Componentes específicos del home
│   │   ├── Actions.tsx
│   │   ├── ActionsHome.tsx
│   │   └── SendList.tsx
│   ├── ui/                       # Componentes UI reutilizables
│   │   ├── AnimatedTabScreen.tsx
│   │   ├── Button.tsx
│   │   ├── Header.tsx
│   │   ├── Input.tsx
│   │   └── Text.tsx
│   ├── AlertModal.tsx
│   ├── Haptics.tsx
│   └── index.tsx                 # Barrel exports
```

```
├── constants/
│   └── theme.ts                # Colores y tema
├── hooks/
│   ├── use-color-scheme.ts     # Hook para detectar tema
│   └── use-theme-color.ts      # Hook para colores del tema
└── utils/
    └── fonts.tsx               # Configuración de fuentes
```

Instalación y Ejecución

Prerrequisitos

- Node.js 18+
- npm o yarn
- Expo CLI (`npm install -g expo-cli`)
- Para Android: Android Studio con emulador configurado
- Para iOS: Xcode (solo en macOS)

Pasos de Instalación

1. **Clonar el repositorio**

   ```
   git clone https://github.com/dcodesv/boxful-app.git
   cd boxful-app
   ```

2. **Instalar dependencias**

   ```
   npm install
   ```

3. **Iniciar el servidor de desarrollo**

   ```
   npm start
   # o
   npx expo start
   ```

4. **Ejecutar en dispositivo/emulador**

   - Presionar `a` para Android
   - Presionar `i` para iOS (requiere macOS)
   - Presionar `w` para Web
   - Escanear el QR con Expo Go (dispositivo físico)

Scripts Disponibles

Comando	Descripción
`npm start`	Inicia el servidor de desarrollo
`npm run android`	Ejecuta en Android
`npm run ios`	Ejecuta en iOS

`npm run web`	Ejecuta en navegador
`npm run lint`	Ejecuta el linter

Características Adicionales

- **Tema oscuro/claro**: Soporte completo con detección automática del sistema
- **Optimización de imágenes**: Cache en memoria y disco con `expo-image`
- **Barrel exports**: Importaciones simplificadas desde `@/components`
- **Feedback háptico**: Vibración sutil al interactuar en iOS
- **Validación de formularios**: Con errores específicos por campo
- **TypeScript estricto**: Tipado completo en toda la aplicación

Notas de Desarrollo

- Las credenciales de login son de prueba (cualquier email válido + contraseña de 6+ caracteres)
- Los datos de envíos son mock data para demostración
- El proyecto está preparado para integrar Zustand como estado global
- Las pantallas de Analíticas y Facturación muestran placeholder "En construcción"

Autor

Diego Villalobos

Ingeniero de Software & Freelancer

PORTFOLIO DIEGOCODESV.NETLIFY.APP GITHUB DCODESV

Ver Repositorio en GitHub

Desarrollado con React Native & Expo