

# Resumen del Estado del Proyecto - Holocron SWU Card Scanner

---

## ✓ Configuraciones Completadas

---

### Dependencias Instaladas




- ✓ React Native 0.80.0
- ✓ React Navigation (native + native-stack)
- ✓ React Native Vision Camera 4.7.0
- ✓ AsyncStorage para almacenamiento local
- ✓ Axios para peticiones HTTP
- ✓ React Native Vector Icons
- ✓ React Native Image Picker
- ✓ React Native FS
- ✓ TypeScript con tipos configurados

### Estructura del Proyecto




```
src/
├── components/
│   ├── Button.tsx          ✓ Componente Button reutilizable
│   └── index.ts            ✓ Exports centralizados
├── screens/
│   ├── HomeScreen.tsx      ✓ Pantalla principal
│   ├── ScanScreen.tsx      ✓ Pantalla de escaneo
│   ├── CollectionScreen.tsx ✓ Pantalla de colección
│   └── CardDetailScreen.tsx ✓ Pantalla de detalle
├── navigation/
│   └── AppNavigator.tsx    ✓ Navegación configurada
├── services/
│   ├── api.ts              ✓ Cliente HTTP
│   ├── storage.ts          ✓ Almacenamiento local
│   └── camera.ts           ✓ Servicios de cámara
├── types/
│   └── index.ts            ✓ Tipos TypeScript
├── utils/
│   └── index.ts            ✓ Utilidades
└── assets/                 📁 Carpeta para recursos
```

# Configuraciones de Plataforma







## Android

-  Permisos de cámara en AndroidManifest.xml
-  React Native Vector Icons configurado en build.gradle
-  Permisos de almacenamiento configurados





## iOS

-  Permisos de cámara en Info.plist
-  Descripciones de uso en español
-  Configuración de micrófono para cámara

## Herramientas de Desarrollo

-  TypeScript configurado con paths aliases
-  Metro Bundler con aliases configurados
-  ESLint y Prettier
-  Scripts npm personalizados
-  Tareas de VS Code para Metro y Android
-  Variables de entorno preparadas





## Documentación

-  [README.md](#) traducido al español
-  CONFIGURACION\_ENTORNO.md con guía detallada
-  [copilot-instructions.md](#) para desarrollo
-  Este archivo de resumen

## Pendientes por Resolver

---

### Entorno de Desarrollo

-  Android SDK no encontrado en ubicación estándar
-  ANDROID\_HOME no configurado permanentemente
-  ADB no disponible en PATH
-  Java/JDK posiblemente no configurado

### Dependencias Opcionales

-  TensorFlow Lite (conflictos de dependencias detectados)

- ⚠ Integración con API real pendiente
- ⚠ Configuración de backend

## Testing y CI/CD

- ⚠ Tests unitarios no implementados
- ⚠ Tests de integración pendientes
- ⚠ Pipeline de CI/CD no configurado

## Próximos Pasos Recomendados

---

### Inmediatos (Para continuar desarrollo)

#### 1. Configurar Android SDK

```
# Verificar ubicación del SDK en Android Studio
# Configurar ANDROID_HOME permanentemente
# Agregar platform-tools al PATH
```

#### 2. Probar compilación

```
npm run android
# o
npm run ios # en macOS
```

#### 3. Verificar todas las funcionalidades

```
npm run doctor
```

## Desarrollo de Funcionalidades

#### 1. Implementar escáner real

- Integrar Vision Camera
- Procesamiento de imágenes
- Integración con ML (TensorFlow Lite)

#### 2. Conectar con Backend

- Configurar URLs de API
- Implementar autenticación
- Manejar estados de loading/error

### 3. Mejorar UI/UX

- Agregar más componentes
- Implementar tema/diseño
- Agregar animaciones

## Testing y Calidad

### 1. Implementar Tests

```
npm test
```

### 2. Configurar Linting estricto

```
npm run lint
```



## Estado de Metro Bundler

---

- ☒ Metro se ejecuta correctamente en puerto 8081
- ☒ Sin errores de compilación TypeScript
- ☒ Aliases de importación funcionando
- ☒ Hot reload habilitado



## Comandos Útiles Disponibles

---

#### *# Desarrollo*

<code>npm start</code>	<i># Iniciar Metro Bundler</i>
<code>npm run android</code>	<i># Ejecutar en Android</i>
<code>npm run ios</code>	<i># Ejecutar en iOS</i>

#### *# Mantenimiento*

<code>npm run clean</code>	<i># Limpiar React Native</i>
<code>npm run clean-android</code>	<i># Limpiar build Android</i>
<code>npm run clean-install</code>	<i># Reinstalar node_modules</i>
<code>npm run reset-cache</code>	<i># Reset caché Metro</i>

#### *# Debug y Logging*

<code>npm run log-android</code>	<i># Logs Android</i>
<code>npm run log-ios</code>	<i># Logs iOS</i>
<code>npm run doctor</code>	<i># Verificar entorno</i>
<code>npm run info</code>	<i># Info del entorno</i>

#### *# Testing y calidad*

npm `test`  
npm run lint

# *Ejecutar tests*  
# *Linting*



## Logros Destacados

---

1. Proyecto React Native funcional con estructura modular
2. TypeScript configurado con types y aliases
3. Navegación implementada entre pantallas
4. Servicios base creados para API, storage y cámara
5. Documentación completa en español
6. Entorno de desarrollo preparado con tareas VS Code
7. Metro Bundler ejecutándose sin errores



## Avisos Importantes

---

- El proyecto está listo para desarrollo pero necesita configuración final de Android SDK
- No ejecutar en dispositivos reales hasta resolver configuración de entorno
- El escáner de cámara está implementado como mockup, necesita integración real
- TensorFlow Lite debe agregarse después de resolver conflicts de dependencias

Estado General: ● LISTO PARA DESARROLLO

Última actualización: 24 de junio de 2025