

SCHOOLME

ACADEMICS

Manual Técnico Aplicativo Móvil



**Centro de la industria, la empresa y los servicios
ADSO 2900177**

SCHOOLME

ACADEMICS

Grupo de Desarrollo SchollMe
Jesus Fernando Carvajal
Santiago Chaparro Riaño

jesusanacona017@gmail.com
alfasan1481@gmail.com



Información del Documento

Proyecto: SchoolMeMóvil

Versión: 1.0

Autor: Grupo SchoolMe

Ubicación: Neiva, Huila, Colombia

Fecha: Septiembre 2025

Confidencialidad: Documento interno

Tabla de Contenidos

1. Introducción
 2. Arquitectura del Sistema
 3. Requisitos Técnicos
 4. Instalación y Configuración
 5. Estructura del Proyecto
 6. Gestión de Dependencias
 7. Autenticación y Seguridad
 8. Especificación de API REST
 9. Modelos de Datos
 10. Flujos de Negocio
 11. Pruebas y Validación
 12. Compilación y Despliegue
 13. Resolución de Problemas
 14. Mantenimiento y Monitoreo
 15. Referencias y Contacto
-

1. Introducción

1.1 Propósito del Documento

Este manual técnico proporciona la documentación completa para desarrolladores, administradores de sistemas y personal técnico responsable de la implementación, mantenimiento y soporte de **SchoolMeMóvil**.

1.2 Alcance

SchoolMeMóvil es un aplicativo móvil que digitaliza la agenda escolar, permitiendo a los acudientes consultar la agenda académica de sus hijos y a los usuarios registrados (docentes y administradores) gestionar su información de perfil.

1.3 Audiencia

Este documento está dirigido a:

- Desarrolladores frontend y backend
- Arquitectos de software
- Administradores de sistemas
- Personal de QA y testing
- Equipos de soporte técnico

1.4 Stack Tecnológico

Tecnología

Versión Propósito

React Native

Latest Framework de desarrollo móvil



Tecnología	Versión	Propósito
TypeScript	5.x	Lenguaje de programación tipado
Expo SDK	49+	Plataforma de desarrollo
React Navigation	6.x	Sistema de navegación
Context API	Native	Gestión de estado global
JWT	—	Autenticación y autorización
AsyncStorage	Latest	Almacenamiento local persistente

2. Arquitectura del Sistema

2.1 Arquitectura General

SchoolMeMovil implementa una **arquitectura modular** con los siguientes componentes:

Cliente Móvil (React Native):

- Interfaz de usuario responsive
- Gestión de estado con Context API
- Almacenamiento local con AsyncStorage
- Comunicación HTTP con API REST

Servidor Backend (API REST):

- Autenticación JWT
- Gestión de usuarios y perfiles
- Administración de agendas escolares
- Control de acceso basado en roles

2.2 Patrón Arquitectónico

Presentación (Screens/Modals)

↓

Navegación (React Navigation)

↓

Lógica de Negocio (Context/Store)

↓

Servicios API (api/services)

↓

Backend REST API

2.3 Principios de Diseño

- **Separación de responsabilidades:** Cada módulo cumple una función específica.
- **Reutilización de componentes:** UI components genéricos y configurables.
- **Tipado fuerte:** Uso de TypeScript para evitar errores en tiempo de desarrollo.
- **Estado centralizado:** Context API para datos globales.
- **Modularidad:** Estructura clara y escalable.

3. Requisitos Técnicos

3.1 Requisitos de Desarrollo

Componente	Versión mínima	Propósito
Node.js	18.x+	Runtime de JavaScript



Componente	Versión mínima	Propósito
npm	9.x+	Gestor de paquetes
Git	2.x+	Control de versiones
Java JDK	11+	Compilación Android
Android Studio	Última estable	Emulador Android
Expo CLI	Global	Herramienta de desarrollo Expo

Sistema operativo compatible:

- Windows 10/11 (64-bit)
- macOS 10.15+
- Linux Ubuntu 20.04+

3.2 Requisitos de Hardware

Mínimos:

- CPU: Intel i3
- RAM: 8 GB
- Almacenamiento: 10 GB

Recomendados:

- CPU: Intel i5/i7
- RAM: 16 GB
- SSD con 20 GB disponibles

3.3 Dispositivos Móviles Compatibles

Android:

- Mínimo: Android 6.0 (API 23)
- Recomendado: Android 10+ (API 29+)

iOS (futuro):

- iOS 13.0 o superior

4. Instalación y Configuración

4.1 Configuración del Entorno

Paso 1: Instalar Node.js y npm

node --version

npm --version

Paso 2: Instalar Expo CLI

npm install -g expo-cli

expo --version

Paso 3: Configurar Android Studio

- Instalar Android SDK (API 33 recomendado)
- Configurar variables de entorno
- Crear emulador Android o conectar dispositivo físico

4.2 Clonar e Instalar el Proyecto

git clone https://github.com/JesusCarvajal017/SchoolMeMovil.git

cd SchoolMeMovil

npm install

4.3 Configurar Variables de Entorno

Editar src/api/constant/Enviroment.ts:

const uri = "IP Local del portátil donde esté la API";

export const environment = {



```
urlApi: `http://${uri}:5052/api`  
};
```

5. Estructura del Proyecto

Árbol principal:

SchoolMeMovil/

```
  └── .expo/  
  └── android/  
  └── assets/  
  └── src/  
    └── api/  
    └── components/  
    └── context/  
    └── navigation/  
    └── screens/  
    └── modals/  
    └── util/  
    └── App.tsx  
    └── Dockerfile  
    └── package.json  
    └── README.md
```

Módulos principales:

- api/: Comunicación con el backend
 - components/: Componentes reutilizables
 - context/: Manejo de sesión y estado global
 - navigation/: Configuración de rutas
 - screens/: Pantallas principales de la app
-

6. Gestión de Dependencias

6.1 Dependencias Principales

npm install react-native react-native-safe-area-context react-native-reanimated

npm install @react-navigation/native @react-navigation/native-stack

npm install expo expo-status-bar expo-linear-gradient

npm install jwt-decode @react-native-async-storage/async-storage

npm install yup react-hook-form

6.2 Dependencias de Desarrollo

npm install -D typescript jest @testing-library/react-native



7. Autenticación y Seguridad

SchoolMeMóvil utiliza **JWT (JSON Web Tokens)** para autenticación y control de acceso basado en roles.

Flujo de autenticación:

1. Usuario ingresa credenciales.
2. Backend valida y responde con token.
3. Token se guarda en AsyncStorage.
4. Context API mantiene sesión activa.

Roles del sistema:

Rol	Permisos principales
-----	----------------------

acudiente	Consultar agenda, editar perfil
-----------	---------------------------------

docente	Editar perfil, consultar agenda
---------	---------------------------------

admin	Gestión de usuarios y permisos
-------	--------------------------------

8. Especificación de API REST

URL base:

<http://localhost:5052/swagger/index.html>

Ejemplo de respuesta exitosa:

```
{  
  "status": 200,  
  "data": { ... }  
}
```

Ejemplo de error:

```
{  
  "status": 400,  
  "error": "Mensaje de error"  
}
```

Incluye endpoints de:

- Autenticación
 - Usuarios
 - Personas
 - Agenda
 - Roles
-

9. Modelos de Datos

Ejemplo: User

```
export interface User {  
  id: number;  
  email: string;  
  personId?: number;  
  roles: string[];  
  status: 'active' | 'inactive';  
}
```



10. Flujos de Negocio

Flujo de Login:

1. Ingreso de credenciales
2. Validación en backend
3. Almacenamiento de token
4. Redirección a HomeScreen

Flujo de Consulta de Agenda (Acudiente):

- GET /api/AgendaDayStudent?studentId={id}
- Muestra tareas por fecha

Flujo de Edición de Perfil:

- PATCH /api/User
 - Actualiza información en tiempo real
-

11. Pruebas y Validación

- **Unitarias:** con Jest
- **Integración:** flujos de login, agenda, edición de perfil
- **E2E:** simulación completa con Expo o Detox

Comando principal:

npm test

Cobertura esperada: 80 %

12. Compilación y Despliegue

12.1 Entornos

Entorno Propósito API URL

Desarrollo Local <http://localhost:5052/api>

Staging Preproducción http://IP_LOCAL_API:5052/api

12.2 Ejecutar en Modo Desarrollo

expo start

npm run android

12.3 Compilar APK

eas build --platform android --profile preview

Producción (Play Store):

eas build --platform android --profile production

13. Resolución de Problemas

- Error 401 → Token expirado.
 - Error 500 → Verificar configuración del backend.
 - App no inicia → Revisar dependencias o caché (npm start -c).
-



14. Mantenimiento y Monitoreo

- Actualizar dependencias periódicamente (npm update).
 - Revisar logs con expo logs.
 - Monitorear rendimiento con Expo Go.
-

15. Referencias y Contacto

Repositorio: <https://github.com/JesusCarvajal017/SchoolMeMovil>

Autor: Grupo SchoolMe

Correo: soporte@schoolme.co

Ciudad: Neiva – Huila, Colombia

Versión: 1.0 – Septiembre 2025