

AthletIA - Documentación Técnica

AthletIA

Equipo de Desarrollo de AthletIA

2026-02-05

Tabla de contenidos

1	Introducción	1
2	Arquitectura C4	2
2.1	Nivel 1: Contexto	2
2.2	Nivel 2: Contenedores	3
2.3	Nivel 3: Componentes	4
2.4	Nivel 3: Frontend	5
3	Diagrama de Clases	5
4	Tecnologías por Componente	6
4.1	Backend	6
4.1.1	Módulos Principales	6
4.2	Frontend Web	6
4.2.1	Secciones Principales	6
4.3	Mobile App	7
4.3.1	Pantallas Principales	7
5	Stack Tecnológico General	7
6	Consideraciones de Seguridad	7

1. Introducción

AthletIA es una plataforma integral para gestión de rutinas de ejercicios, entrenamiento y monitoreo de progreso físico. La arquitectura está diseñada para soportar múltiples clientes (Web y Mobile) con un backend robusto.

2. Arquitectura C4

2.1. Nivel 1: Contexto

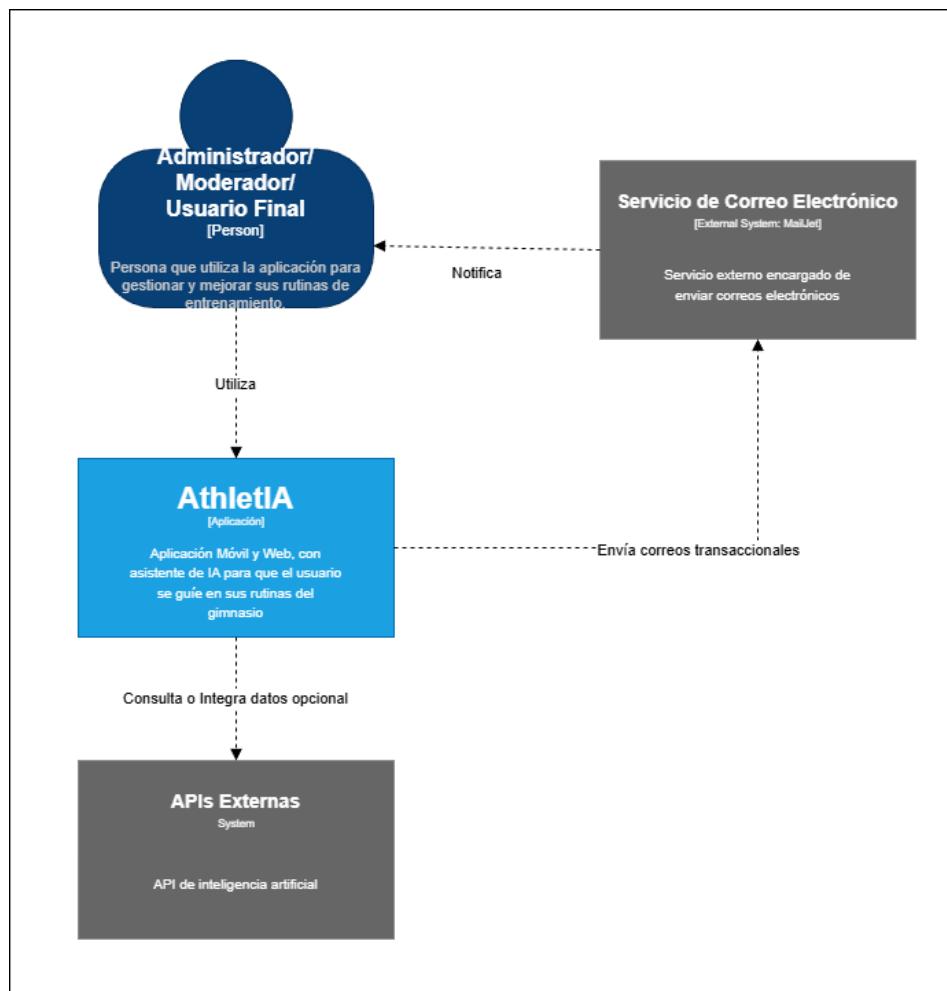


Figura 1: Diagrama C4 - Nivel 1 (Contexto)

El sistema se estructura alrededor del usuario final, proporcionando acceso a través de aplicaciones web y móvil, integradas con servicios de autenticación y notificaciones.

2.2. Nivel 2: Contenedores

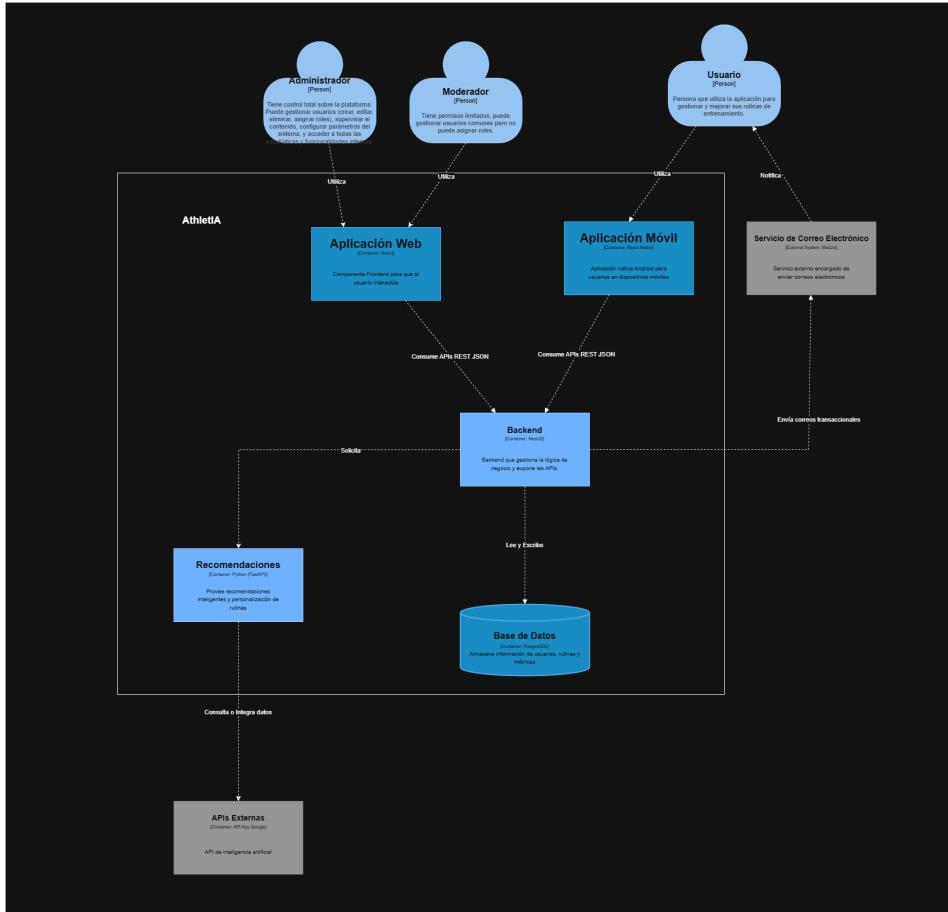


Figura 2: Diagrama C4 - Nivel 2 (Contenedores)

Los contenedores principales son:

- Frontend Web:** Aplicación React + Vite para gestión administrativa
- Mobile App:** Aplicación Expo/React Native para usuarios finales
- Backend API:** NestJS con bases de datos relacionales
- Servicios Externos:** Autenticación Google, notificaciones

2.3. Nivel 3: Componentes

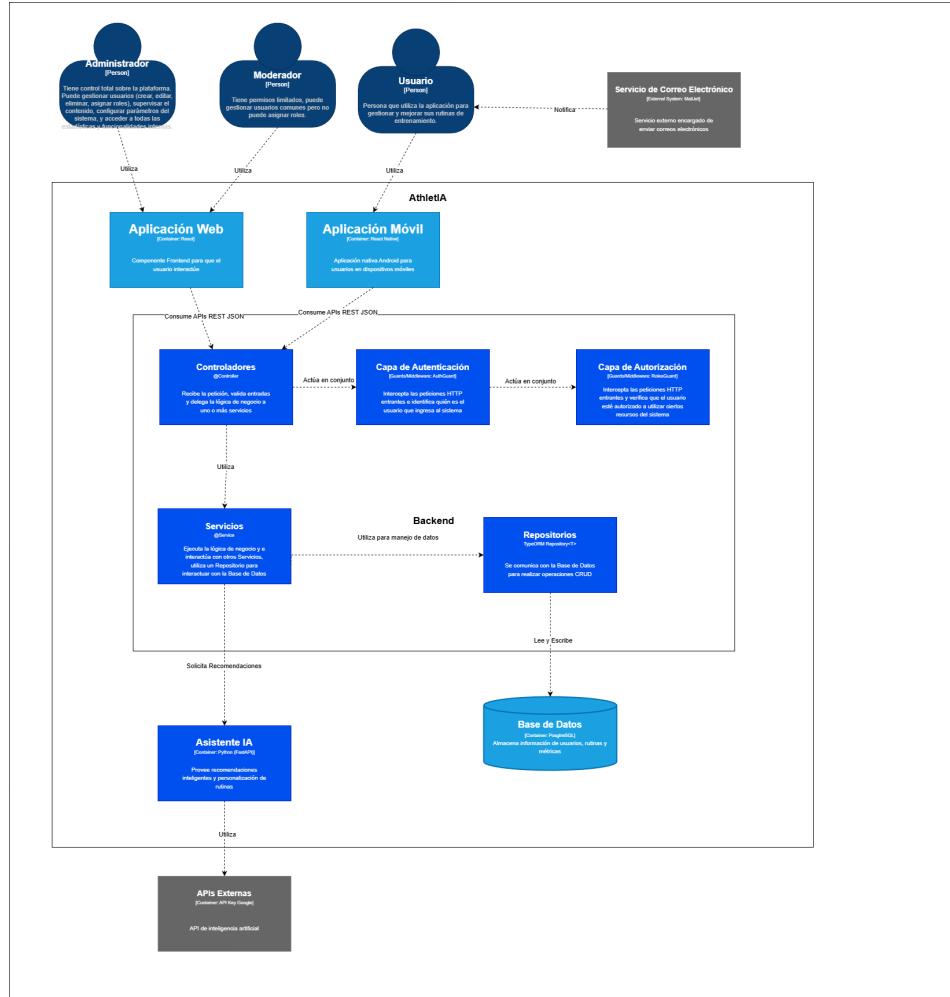


Figura 3: Diagrama C4 - Nivel 3 (Componentes)

El backend se divide en módulos especializados:

- Auth Module:** Autenticación y autorización
- Users Module:** Gestión de perfiles y cuentas
- Workout Module:** Ejercicios, rutinas y splits
- Measurements Module:** Mediciones de progreso
- Notifications Module:** Sistema de notificaciones
- AI Module:** Integración con IA

2.4. Nivel 3: Frontend

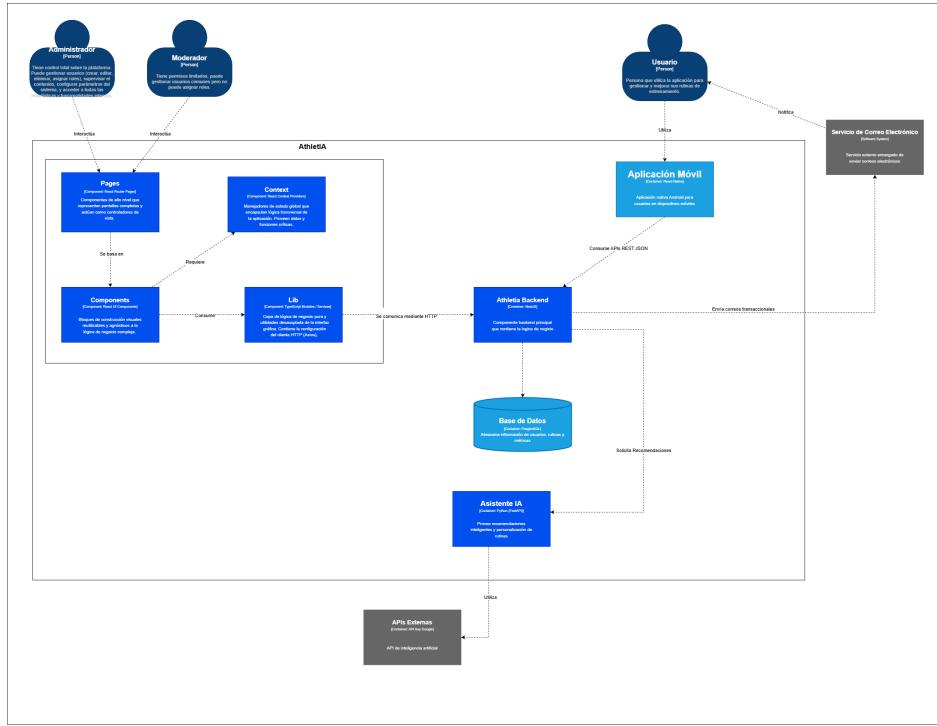


Figura 4: Diagrama C4 - Nivel 3 (Frontend)

3. Diagrama de Clases

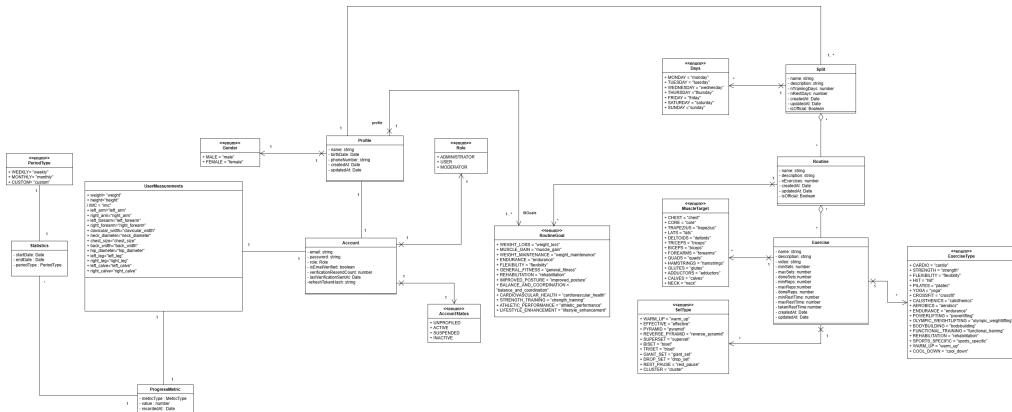


Figura 5: Diagrama de Clases

4. Tecnologías por Componente

4.1. Backend

- **Framework:** NestJS (Node.js)
- **Base de Datos:** PostgreSQL
- **Autenticación:** JWT + Google OAuth
- **Validación:** Class Validator
- **API Documentation:** Swagger/OpenAPI
- **Rate Limiting:** Sistema integrado
- **Correo:** Nodemailer

4.1.1. Módulos Principales

Módulo	Responsabilidad
Auth	Autenticación, autorización, roles
Users	Perfiles, cuentas, datos de usuario
Workout	Ejercicios, rutinas, splits
Measurements	Seguimiento de progreso
Notifications	Sistema de notificaciones
AI	Integración con servicios IA

4.2. Frontend Web

- **Framework:** React 18 + TypeScript
- **Build Tool:** Vite
- **Estilos:** Tailwind CSS
- **UI Components:** Componentes personalizados
- **Internacionalización:** i18n
- **Gestión de Estado:** Context API
- **HTTP Client:** Axios

4.2.1. Secciones Principales

- Dashboard
- Gestión de Usuarios
- Gestión de Ejercicios
- Rutinas y Splits
- Mediciones

- Perfil de Usuario

4.3. Mobile App

- **Framework:** Expo + React Native
- **Lenguaje:** TypeScript
- **Navegación:** Expo Router
- **Estilos:** NativeWind (Tailwind para React Native)
- **Almacenamiento:** AsyncStorage
- **Internacionalización:** i18n
- **Mapeo de Pantallas:** Routing File-based

4.3.1. Pantallas Principales

- Login / Signup
- Dashboard (Inicio)
- Explorar Ejercicios
- Mis Rutinas
- Splits/Entrenamientos
- Mediciones
- Notificaciones
- Perfil

5. Stack Tecnológico General

Capa	Tecnología
Backend	NestJS, TypeScript, PostgreSQL
Frontend Web	React, TypeScript, Vite, Tailwind
Mobile	Expo, React Native, TypeScript
Autenticación	JWT, OAuth2 (Google)
Comunicación	REST API
Deployment	Node.js (Backend), Vite (Frontend)

6. Consideraciones de Seguridad

- Autenticación basada en JWT
- Validación de pertenencia de datos (Ownership Guard)

- Control de acceso basado en roles
 - Rate limiting en endpoints
 - Encriptación de contraseñas
 - CORS configurado
-