

# Documento Técnico: ColpoTool

---

## 1. Introducción

ColpoTool es una plataforma especializada en el análisis colposcópico, que integra registros clínicos, gestión de imágenes médicas y la detección de lesiones cancerosas y factores de riesgo mediante modelos de IA.

### Objetivos principales del sistema:

- Registrar información médica, imágenes colposcópicas y resultados.
- Generar predicciones clínicas mediante modelos ML y DL.
- Ofrecer un ecosistema que incluya aplicaciones web, móviles y de escritorio.

### Componentes del ecosistema ColpoTool:

- ❖ Aplicación web: interfaz principal de gestión accesible vía navegador.
- ❖ Aplicación de escritorio: ejecutada localmente mediante un launcher.
- ❖ Aplicación móvil: orientada a la captura remota de imágenes.

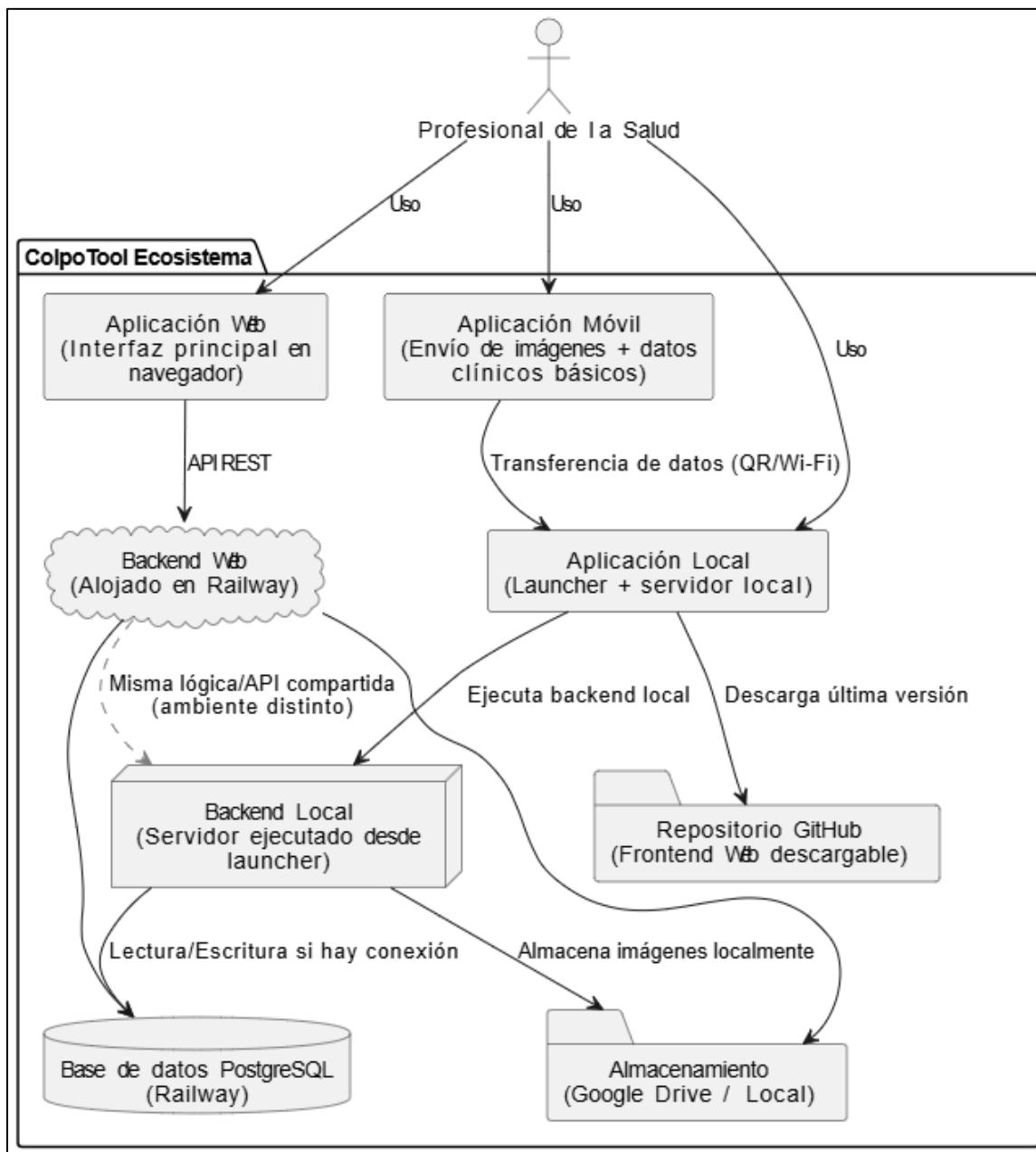
## 2. Arquitectura general del sistema

**Modelo de arquitectura:** Cliente-Servidor + Sincronización Local

### Descripción del flujo de arquitectura:

- La aplicación web accede al backend alojado en la plataforma Railway. Desde allí, se comunica con la base de datos central y el almacenamiento externo para imágenes y documentos.
- La aplicación de escritorio ejecuta un launcher el cual descarga desde GitHub los recursos necesarios para ejecutar el aplicativo .
- La aplicación móvil funciona como cliente auxiliar para captura de imágenes. Transfiere datos a la versión de escritorio mediante código QR o conexión Wi-Fi.
- Ambos backends (web y local) comparten la misma lógica y estructura, pero corren en entornos diferentes.

**Figura 1. Diagrama de arquitectura del Sistema**



#### Tecnologías utilizadas:

Componente	Tecnología
Frontend Web	React + Tabler UI
Backend	Django REST Framework + JWT
Base de Datos	PostgreSQL (Railway)
Aplicación Móvil	React Native CLI
Aplicación Escritorio	Python (Launcher + ejecución local)
Almacenamiento externo	Google Drive / Local

### 3. Modelo de base de datos

La base de datos de ColpoTool fue diseñada bajo un enfoque orientado a garantizar la integridad de los datos. El sistema emplea PostgreSQL como motor principal, desplegado en la plataforma Railway para asegurar alta disponibilidad y conexión constante desde los distintos componentes del ecosistema.

#### Conexión principal:

- Plataforma: **Railway.app**
- Motor: **PostgreSQL 14+**
- Esquema: **colpotool\_db** (estructurado para múltiples tablas clínicas, IA y administrativas)
- Acceso: restringido mediante credenciales privadas y roles definidos

**Figura 2. Diagrama del modelo relacional de la base de datos**



Diagrama disponible en: <https://dbdiagram.io/d/685eee6df413ba35083f4eda>

Tablas principales del sistema:

<b>Tabla</b>	<b>Propósito</b>
<code>users</code>	Credenciales, roles y metadatos de acceso
<code>roles</code>	Niveles de acceso del sistema
<code>patients</code>	Datos demográficos del paciente
<code>clinical_info</code>	Historial clínico detallado
<code>exams</code>	Exámenes colposcópicos por paciente
<code>images</code>	Imágenes asociadas a exámenes
<code>image_versions</code>	Versiones derivadas
<code>lesion_detections</code>	Detección de lesiones IA
<code>swede_scores</code>	Evaluación clínica Swede
<code>colpo_findings</code>	Hallazgos físicos colposcópicos
<code>cancer_stages</code>	Predicción de etapa de cáncer
<code>risk_predictions</code>	Estimación del riesgo clínico
<code>models</code>	Catálogo de modelos IA
<code>model_metrics</code>	Métricas de modelos IA
<code>feature_importance</code>	Importancia de variables
<code>coco_annotations</code>	Anotaciones en formato COCO

Las demás tablas (prefijo auth\_, django\_, etc.) son propias del sistema Django y permiten gestionar permisos, sesiones y migraciones sin intervención directa del usuario final.

## 4. Flujo de datos y funcionalidad

1. **Registro de paciente** → se ingresan los datos personales y antecedentes clínicos en el sistema (**aplicación web o escritorio**).
2. **Registro de examen colposcópico** → se asocia un nuevo examen al paciente, incluyendo detalles clínicos estructurados.
3. **Carga de imágenes** → se capturan y cargan imágenes colposcópicas desde la aplicación móvil o escritorio.
4. **Evaluación clínica e IA** → las imágenes son analizadas mediante hallazgos clínicos estructurados (**formulario Swede, findings**) y por modelos de inteligencia artificial (**segmentación, clasificación, predicción de riesgo**).
5. **Almacenamiento y consulta** → los resultados se almacenan en la base de datos PostgreSQL (**Railway**), junto con las imágenes procesadas, y están disponibles desde la ficha médica del paciente.
6. Sincronización local → la aplicación de escritorio puede operar offline y sincronizar datos con la base remota cuando se restablece la conexión.
7. Conexión con la aplicación móvil → se realiza mediante escaneo de código QR desde la app móvil hacia la app escritorio, habilitando la transferencia directa de imágenes y datos.

## 5. Funcionalidades

- ❖ Gestión completa de pacientes: registro, visualización, y acceso a ficha médica estructurada.
- ❖ Registro de exámenes con soporte multimedia: asociación de imágenes, hallazgos y evaluaciones clínicas.
- ❖ Evaluación clínica estructurada: formularios personalizados como Swede Score y hallazgos colposcópicos.
- ❖ Visualización de resultados IA: acceso a segmentación de lesiones, clasificación y predicción de riesgo.
- ❖ Módulo de modelos IA: permite cargar modelos, visualizar métricas (IoU, mAP, Swede, etc.) y evaluar su rendimiento.
- ❖ Seguridad y control de acceso: roles diferenciados de administrador y usuario para asegurar integridad y trazabilidad.

## 6. Accesos y credenciales

Para fines de prueba, se establece la siguiente configuración de acceso:

### Railway DB:

Host: metro.proxy.rlwy.net
Usuario: postgres
Contraseña: YDJHNXFqkTzcWdKsLNpDZrOSwDkYugLU
Base de datos: railway
Puerto: 39965

### Usuarios del sistema:

admin / admin123
user / user123