

ISFT N°151 – Tecnicatura Superior en Análisis de Sistemas

Materia: Prácticas Profesionalizantes III

Autores: Valentina Tiberio y Martina Lapenta

Docente: Matías Gastón Santiago

Año: 2025



RescatePET - Guía para ejecutar el proyecto

1. Requisitos previos

Antes de ejecutar el proyecto, asegurate de tener instaladas las siguientes herramientas:

- **Python 3.11+**
 - **Flask** (framework principal del backend)
 - **SQLite** (base de datos por defecto)
 - **Visual Studio Code** o cualquier editor de texto
 - **Git** y cuenta en **GitHub** (para control de versiones)
-

2. Instalación

Clonar repositorio desde GitHub:

```
git clone https://github.com/ValentinaTiberio/isft151-pp3.git
```

Entrar al directorio del proyecto:

```
cd isft151-pp3/fp
```

Instalar las dependencias necesarias:

```
pip install flask flask_sqlalchemy flask_cors jwt
```

Crear la base de datos:

Ejecutar el script SQL ubicado en

```
/doc/rescatepet_db.sql
```

Este archivo genera las tablas principales: usuarios, animales, adopciones y hogares de tránsito.

3. Cómo ejecutar el proyecto

-Desde la carpeta /fp, iniciar el servidor con:

```
python -m backend.main
```

El backend levanta desde fp/backend/main.py

-El sistema se ejecutará en:

`http://127.0.0.1:5000`

El frontend se carga automáticamente al acceder a esa dirección desde el navegador.

4. Endpoints principales

Endpoint	Método	Descripción
/api/users/register	POST	Registrar nuevo usuario
/api/users/login	POST	Iniciar sesión y obtener token JWT
/api/animals/	GET / POST	Listar o registrar animales
/api/adopciones/	GET / POST	Listar o crear solicitudes de adopción
/api/adopciones/usuario/<id>	GET	Listar solicitudes de adopción por usuario
/api/adopciones/<id>	PUT	Actualizar estado de solicitud (Aprobada/Rechazada)
/api/transitos/	GET / POST	Listar o asignar hogares de tránsito
/api/transitos/usuario/<id>	GET	Listar solicitudes de tránsito por usuario
/api/transitos/<id>	PUT	Actualizar estado de solicitud de tránsito
/api/reportes/	GET	Consultar estadísticas de rescates y adopciones

5. Estructura del proyecto

```
fp/
├─ backend/
│  │├─ app/
│  │ │├─ models/
│  │ │├─ routes/
│  │ │├─ services/
├─ instance/
│  └─ uploads/
│  └─ main.py

    doc/
    │├─ api_doc.pdf
    │├─ rescatepet_db.sql
    └─

        frontend/
        │├─ components/
        │├─ admin_reportes.html
        │├─ admin_solicitudes.html
        │├─ admin_transitos.html
        │├─ animales.html
    └─ └─ index.html
        │├─ inicio.html
        │├─ registro.html
        │└─ solicitudes.html

        release/
    └─ srs/
    └─ media/
```

6. Notas finales

- El sistema permite registrar usuarios, animales, adopciones y hogares de tránsito.
 - Los datos se almacenan en una base de datos SQLite.
 - Las imágenes se guardan en el servidor backend dentro de la carpeta /uploads.
 - El token JWT es requerido para acceder a ciertas rutas protegidas.
 - El frontend está desarrollado con HTML, CSS y JavaScript, con un diseño simple y adaptable.
 - El módulo de reportes muestra los totales de animales registrados, adopciones y tránsitos aprobados.
 - Se agregó panel de administración, donde se pueden aprobar o rechazar solicitudes de adopción y tránsito.
 - Los usuarios pueden ver el estado actualizado de sus solicitudes desde “Mis Solicitudes”.
 - El frontend incluye mensajes dinámicos y actualizaciones automáticas de la galería de animales.
-

7. Conclusión

El sistema **RescatePET** promueve la adopción responsable y la colaboración entre usuarios, ofreciendo una gestión completa de animales, adopciones y hogares de tránsito. Su arquitectura modular (Flask + SQLite + frontend HTML/JS) facilita el mantenimiento y la ampliación de funcionalidades futuras, como notificaciones por correo y paneles analíticos más avanzados.