# 📖 AlfaGestion – Backend

AlfaGestion es una plataforma de gestión de servicio social estudiantil, desarrollada en Django + Django REST Framework. Permite administrar estudiantes, docentes, instituciones, actividades y validaciones de servicio social.

## 🚀 Instalación y Ejecución

* Clonar el repositorio:  
    
   git clone https://github.com/tu-org/alfagestion.git  
   cd alfagestion
* Crear entorno virtual e instalar dependencias:  
    
   python -m venv venv  
   source venv/bin/activate # En Linux / Mac  
   venv\Scripts\activate # En Windows  
   pip install -r requirements.txt
* Aplicar migraciones:  
    
   python manage.py migrate
* Crear superusuario:  
    
   python manage.py createsuperuser
* Ejecutar servidor:  
    
   python manage.py runserver  
    
   El proyecto estará disponible en: http://127.0.0.1:8000/

## 🗂️ Estructura de Apps

🔹 accounts: Maneja el registro, login y logout de usuarios. Soporta roles: Estudiante y Docente.

🔹 institutions: Administra las instituciones educativas y sus encargados.

🔹 students: Permite a los estudiantes subir actividades, consultar validaciones y gestionar su portafolio.

🔹 teachers: Permite a los docentes consultar estudiantes, actividades y emitir validaciones.

(También incluye apps auxiliares como evidence y reports que soportan la gestión de actividades y validaciones).

## 📡 APIs REST

### Accounts (Autenticación)

* POST /accounts/register/ → Registrar usuario
* POST /accounts/login/student/ → Login estudiantes
* POST /accounts/login/teacher/ → Login docentes
* GET /accounts/logout/ → Logout

### Institutions

* GET /institutions/<id>/ → Detalles institución
* GET /institutions/encargado/<id>/ → Detalles encargado
* POST /institutions/create/ → Crear institución
* POST /institutions/encargado/create/ → Crear encargado

### Students

* GET /students/actividades/ → Listar actividades del estudiante autenticado
* POST /students/actividades/ → Subir nueva actividad
* GET /students/validaciones/ → Consultar estados de validación
* GET /students/actividades/<pk>/ → Detalles de actividad
* DELETE /students/actividades/<pk>/ → Eliminar actividad

### Teachers

* GET /teachers/grados/ → Listar grados únicos
* GET /teachers/grupos/ → Listar grupos únicos
* GET /teachers/estudiantes/?grado=10&grupo=A → Listar estudiantes por grado/grupo
* GET /teachers/actividades/?estudiante\_id=1 → Listar actividades de un estudiante
* POST /teachers/validaciones/<estudiante\_id>/create/ → Crear validación
* PUT/PATCH /teachers/validaciones/<id>/update/ → Actualizar validación

## ✅ Roles y Flujos Principales

### Estudiante

* Se registra e inicia sesión
* Carga actividades de servicio social
* Consulta validaciones emitidas por docentes

### Docente

* Accede al panel docente
* Filtra estudiantes por grado y grupo
* Revisa actividades y genera validaciones

## 📌 Tecnologías Usadas

* Python 3.x
* Django 4.x
* Django REST Framework
* SQLite / PostgreSQL