

# **Diseño de Microservicios – Non Profit Organization**

**Miguel Villa, Juan Silgado, Dylan Ecker.**

## **1. Introducción**

Este documento describe el diseño arquitectónico basado en microservicios de nuestro proyecto "Non Profit Organization", una plataforma orientada a la gestión de donantes y voluntarios para organizaciones sin fines de lucro. Cada servicio se desarrolla e implementa de manera desacoplada, con responsabilidad única y comunicación mediante API REST.

## **2. Arquitectura General**

- Lenguaje backend: Python
- Framework: FastAPI
- Base de datos: MongoDB Atlas
- Comunicación: REST (HTTP)
- Almacenamiento por servicio: Cada servicio conecta con la misma base de datos MongoDB pero accede únicamente a su propia colección

## **3. Servicios Implementados**

### **Donor Service**

- Funcionalidad: Gestión CRUD de donantes
- Ruta base: /donors
- Colección MongoDB: donors
- Endpoints:
  - GET /donors
  - POST /donors
  - PUT /donors/{id}
  - DELETE /donors/{id}

### **Volunteer Service**

- Funcionalidad: Gestión CRUD de voluntarios
- Ruta base: /volunteers
- Colección MongoDB: volunteers
- Endpoints:
  - GET /volunteers
  - POST /volunteers
  - PUT /volunteers/{id}
  - DELETE /volunteers/{id}

### **Analytics Service**

- Funcionalidad: Consulta de estadísticas generales sobre donantes y voluntarios
- Ruta base: /analytics
- Colecciones utilizadas: donors, volunteers (sólo lectura)
- Endpoints:
  - GET /analytics/donors/count (Total de donantes)
  - GET /analytics/volunteers/count (Total de voluntarios)
  - GET /analytics/summary (Resumen combinado)

## **4. Base de Datos**

- Motor: MongoDB Atlas (cluster en la nube)
- Conexión: vía URI en archivo connect\_db.py
- Acceso separado por colección:
  - donors\_collection para Donor Service
  - volunteers\_collection para Volunteer Service