

Diseño Estructural - Servicios de Non Profit Organization

Integrantes: Dylan Ecker, Miguel Villa, Juan Silgado.

Fecha: 5 de mayo de 2025

Índice

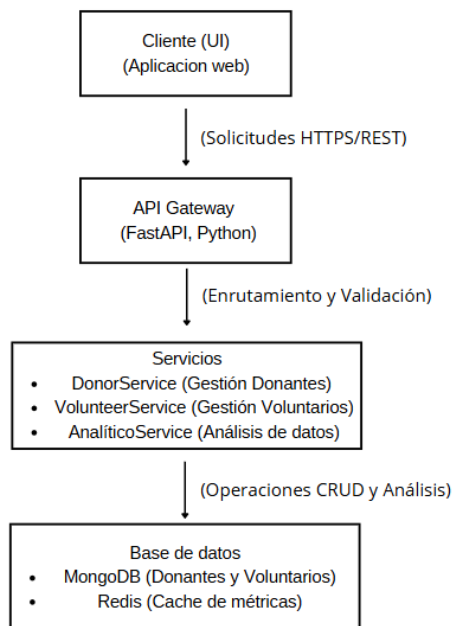
1. Introducción
2. Diagrama por capas
3. Diagrama de clases
4. Descripción de los microservicios
 - 4.1. Servicio de Donantes (DonorService)
 - 4.2. Servicio de Voluntarios (VolunteerService)
 - 4.3. Servicio Analítico (AnalíticoService)
5. Patrones de Diseño Implementados
6. Flujos de Comunicación
7. Ventajas de la Arquitectura

1. Introducción

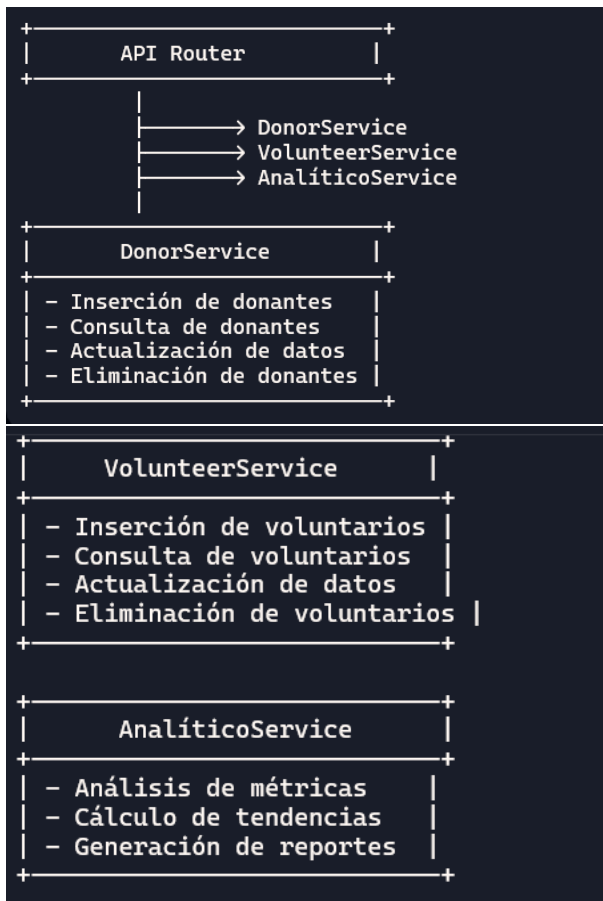
Este documento describe la arquitectura de los servicios para la gestión de donantes y voluntarios en el sistema **Non Profit Organization**, basado en FastAPI y MongoDB. La estructura sigue principios de diseño modular y desacoplamiento, permitiendo escalabilidad y fácil mantenimiento.

Los servicios gestionan datos de usuarios, operaciones CRUD y análisis de métricas para obtener tendencias y reportes.

2. Diagrama por Capas



3. Diagrama de Clases



4. Descripción de los Microservicios

4.1 Servicio de Donantes (DonorService)

- **Función principal:** Gestiona operaciones sobre donantes.
- **Componentes:**
 - DonorsCollection: Base de datos para almacenamiento.
 - DonorSerializer: Convierte BSON a JSON para respuesta.
- **Operaciones clave:**
- **POST** /donors → Inserción de donante.
- **GET** /donors → Obtención de todos los donantes.
- **PUT** /donors/{donor_id} → Actualización de datos de donante.
- **DELETE** /donors/{donor_id} → Eliminación de donante.

4.2 Servicio de Voluntarios (VolunteerService)

- **Función principal:** Gestiona operaciones sobre voluntarios.
- **Componentes:**
 - VolunteersCollection: Base de datos para almacenamiento.
 - VolunteerSerializer: Convierte BSON a JSON para respuesta.
- **Operaciones clave:**
- **POST** /volunteers → Inserción de voluntario.
- **GET** /volunteers → Obtención de todos los voluntarios.
- **PUT** /volunteers/{volunteer_id} → Actualización de datos de voluntario.
- **DELETE** /volunteers/{volunteer_id} → Eliminación de voluntario.

4.3 Servicio Analítico (AnalíticoService)

- **Función principal:** Realiza análisis de datos sobre donantes y voluntarios.
- **Componentes:**
 - DonorsCollection, VolunteersCollection: Fuentes de datos.
- **Operaciones clave:**
- **GET** /analytics/donors/trends → Devuelve tendencias de donación.
- **GET** /analytics/volunteers/trends → Analiza patrones de voluntariado.
- **GET** /analytics/reports → Genera reportes detallados.

5. Patrones de Diseño Implementados

- **Facade Pattern:** API centralizada para gestionar servicios.
- **Singleton Pattern:** Servicio de autenticación único (si se integra).
- **Adapter Pattern:** Serialización de datos BSON a JSON para respuestas.
- **Repository Pattern:** Abstracción de operaciones CRUD sobre MongoDB.

6. Flujos de Comunicación

1. **El cliente** envía solicitudes a los endpoints de los servicios.
2. **El API Gateway** enruta la solicitud al servicio correspondiente.
3. **Los servicios** procesan la solicitud mediante operaciones CRUD y análisis.

4. **MongoDB** almacena los datos y responde a consultas.
5. **Redis** optimiza el almacenamiento temporal de métricas.
6. **AnalíticoService** realiza análisis avanzados sobre los datos y genera reportes.

7. Ventajas de la Arquitectura

- **Escalabilidad:** Los servicios operan de manera independiente.
- **Mantenibilidad:** Código modular permite mejoras sin impacto en otros módulos.
- **Eficiencia:** Serialización optimizada para respuestas rápidas.
- **Consistencia:** Cada servicio mantiene la integridad de su información.