

---

# **Especificación de requisitos de software**

**Para**

**Non-Profit Organization**

**Versión 3.0 aprobado**

**Preparado por Juan Silgado, Miguel Villa, Dylan Ecker**

**Universidad Tecnológica de Bolívar**

**23 de abril de 2025**

## Tabla de contenido

<b>1. Introducción.....</b>	<b>1</b>
<b>2. Descripción general .....</b>	<b>2</b>
2.1 Características del producto.....	2
2.2 Clases de usuarios y características .....	2
2.3 Limitaciones de diseño y aplicación.....	2
2.4 Documentación del Usuario .....	3
2.5 Supuestos y dependencias .....	3
<b>3. Características del sistema.....</b>	<b>4</b>
3.1 Características de Sistema .....	4
3.1.1 Descripción y prioridad.....	4
3.1.2 Secuencias de estímulo/respuesta.....	4
3.1.3 Requisitos funcionales .....	4
<b>4. Requisitos de la interfaz.....</b>	<b>6</b>
4.1 Interfaces de Usuario.....	6
4.2 Interfaces de hardware .....	6
4.3 Interfaces de software.....	6
4.4 Interfaces de Comunicación .....	6
4.5 Requerimientos de desempeño .....	7
4.6 Requisitos de seguridad.....	7
4.7 Atributos de calidad del software.....	7
<b>5. Otros Requerimientos .....</b>	<b>8</b>

## Historial de revisiones

Nombre	Fecha	Motivo de los cambios	Versión
Inicio del proyecto	16/03/25	documentación inicial del proyecto	0.1
MVP	16/03/25	documentación inicial del MVP del proyecto	0.2
Actualización 2.0	22/03/25		
Actualización 3.0	23/03/25	Integración completa de la arquitectura, interfaz, MVP y actualización de requerimientos	3.0

## **1. Introducción**

Este documento especifica los requisitos para una plataforma de gestión de datos de una organización sin fines de lucro. Su propósito es integrar y consolidar datos de donantes, voluntarios y programas, optimizando la administración y coordinación de actividades.

La solución busca mejorar la experiencia de voluntarios y donantes mediante un sistema basado en microservicios, que expondrá datos relevantes a través de APIs y almacenará información en un data warehouse en la nube.

## **2. Descripción general**

El producto reemplazará la infraestructura fragmentada existente, consolidando sistemas de recaudación de fondos, gestión de voluntarios y programas en una plataforma unificada. Incluirá una arquitectura de microservicios con APIs centralizadas para facilitar la interoperabilidad entre sistemas.

### **2.1 Características del producto**

El producto cuenta con:

- Microservicios para la gestión de donantes, voluntarios y programas.
- API Gateway para gestionar el acceso a los datos.
- Data warehouse en la nube para consolidación y análisis de datos.
- Sistema de analítica avanzada para optimizar la toma de decisiones.

### **2.2 Clases de usuarios y características**

Estas son las clases de usuarios que se prevén en el sistema:

- Administradores: Acceso completo a la plataforma, gestión de usuarios y datos.
- Coordinadores de voluntarios: Asignación y seguimiento de actividades.
- Donantes: Visualización de historial de donaciones e impacto.

### **2.3 Limitaciones de diseño y aplicación**

- Integración con plataformas de terceros mediante APIs estándar.
- Uso de tecnologías en la nube para almacenamiento y procesamiento de datos.
- Cumplimiento con normativas de privacidad y protección de datos.

### **2.4 Documentación del Usuario**

- Manual de usuario para administradores y coordinadores.
- Guía de integración de APIs para desarrolladores externos.

### **2.5 Supuestos y dependencias**

- Conectividad a internet para el uso de la plataforma.

- Integración con sistemas de terceros a través de APIs.
- Dependencia de infraestructura en la nube para almacenamiento y procesamiento.

### 3. Características del sistema

#### 3.1 Gestión de Donantes y Voluntarios

##### 3.1.1 Descripción y Prioridad

- Este módulo permite la administración integral de la información relacionada con donantes y voluntarios dentro de una plataforma centralizada. Incluye funcionalidades para el **registro, consulta, edición, eliminación lógica y seguimiento histórico** de actividades asociadas. Permite asociar donaciones a campañas específicas, categorizar voluntarios según habilidades o disponibilidad, y mantener contacto a través de notificaciones automáticas. Prioridad: Alta.

##### 3.1.2 Secuencias de Estímulo/Respuesta

1. Un usuario ingresa y consulta información de donantes o voluntarios.
2. El sistema recupera y muestra los datos desde el data warehouse.

##### 3.1.3 Requisitos Funcionales

- REQ-1: El sistema debe permitir registrar nuevos donantes y voluntarios mediante formularios validados en frontend y backend.
- REQ-2: Debe permitir consultar, filtrar, editar y eliminar registros existentes, manteniendo integridad referencial con campañas, eventos o actividades asociadas.
- REQ-4: Los cambios realizados sobre los registros deben quedar trazados en un historial de auditoría (quién, cuándo y qué fue modificado).

## 4. Requisitos de la interfaz

### 4.1 Interfaces de Usuario

- Aplicación web con panel de administración:

Diseñada para ser intuitiva y de fácil navegación, esta interfaz permitirá a los administradores gestionar usuarios, roles, permisos, contenidos y generar reportes personalizados. Se incluirán funcionalidades como filtrado por fechas, exportación de datos en distintos formatos y control de acceso según rol.

- Dashboard interactivo con gráficos y métricas de impacto:

Permitirá visualizar en tiempo real indicadores clave, como número de usuarios activos, cantidad de donaciones recibidas, actividades voluntarias realizadas, y participación en eventos.

### 4.2 Interfaces de hardware

- Servidores en la nube:

Se utilizarán servicios como AWS, Azure o Google Cloud para garantizar escalabilidad automática, balanceo de carga, recuperación ante desastres, copias de seguridad automatizadas, y procesamiento eficiente de grandes volúmenes de datos.

- Dispositivos móviles y computadoras:

El sistema será accesible desde navegadores modernos, con diseño responsivo (mobile-first), garantizando compatibilidad con Android, iOS, Windows y macOS, asegurando una experiencia de usuario óptima en distintos tamaños de pantalla.

### 4.3 Interfaces de software

- Integración con plataformas de pago y CRM:

Se integrará con pasarelas de pago para gestionar donaciones y pagos. También se conectará con CRMs para seguimiento de relaciones con donantes y voluntarios.

- API Gateway:

Servirá como punto de entrada único para acceder a servicios internos relacionados con la gestión de usuarios, actividades, reportes y datos de donantes. Permitirá autenticación, limitación de velocidad, y control de acceso granular a servicios.

*Otros requisitos No funcionales*

#### **4.4 Requerimientos de desempeño**

- Baja latencia en APIs (<200ms):

Las consultas estándar (como obtener usuarios o actividades recientes) deberán responder en menos de 200 milisegundos, gracias a técnicas como caché en memoria (Redis), índices bien optimizados en bases de datos y balanceo de carga.

- Procesamiento masivo de datos (10,000 registros/minuto):

El sistema deberá soportar operaciones de carga, transformación y análisis de grandes volúmenes de datos en tiempo casi real, aprovechando herramientas como ETL automatizados, particionado de tablas y procesamiento paralelo

#### **4.5 Requisitos de seguridad**

- Cifrado de datos:

Toda la información sensible será cifrada en tránsito usando HTTPS con TLS 1.2/1.3 y en reposo mediante técnicas como AES-256. Se incluirán auditorías periódicas de seguridad y cumplimiento de normativas como GDPR o ISO/IEC 27001.

#### **4.6 Atributos de calidad del software**

- Escalabilidad:

Arquitectura basada en microservicios, uso de contenedores (Docker) y orquestadores como Kubernetes para permitir el escalado horizontal y vertical sin afectar la disponibilidad del sistema.

- Alta disponibilidad:

Despliegue en múltiples zonas de disponibilidad (AZ) dentro de la nube, balanceo de carga redundante y sistemas de failover automático aseguran un tiempo de actividad de al menos 99.9%.

- Mantenibilidad:

Código limpio, estructurado y documentado, siguiendo principios SOLID, con pruebas automatizadas y CI/CD para facilitar despliegues y mantenimiento continuo sin afectar a los usuarios finales.



## **5. Otros Requerimientos**

- Compatibilidad con navegadores modernos y dispositivos móviles.
- Cumplimiento con normativas de privacidad como GDPR.

## **6. Características del MVP**

- **Arquitectura basada en microservicios**
  - Para gestionar donantes, voluntarios y programas de forma modular y escalable.
- **API Gateway**
  - Control de acceso centralizado y seguridad en el intercambio de datos.
- **Data Warehouse en la nube**
  - Almacenamiento de datos estructurado y preparado para análisis.
- **Dashboard interactivo**
  - Visualización de métricas y reportes importantes para la toma de decisiones.