

|  |
| --- |

**Especificación de requisitos de software**

**Proyecto: ZONA ASISTIDA PARA ANIMALES EN PROTECCIÓN (ZAAP)**

Revisión

|  |  |  |
| --- | --- | --- |

**Instrucciones para el uso de este formato**

*Este formato es una plantilla tipo para documentos de requisitos del software.*

*Está basado y es conforme con el estándar IEEE Std 830-1998.*

*Las secciones que no se consideren aplicables al sistema descrito podrán de forma justificada indicarse como no aplicables (NA).*

*Notas:*

*Los textos en color azul son indicaciones que deben eliminarse y, en su caso, sustituirse por los contenidos descritos en cada apartado.*

*Los textos entre corchetes del tipo “” permiten la inclusión directa de texto con el color y estilo adecuado a la sección, al pulsar sobre ellos con el puntero del ratón.*

*Los títulos y subtítulos de cada apartado están definidos como estilos de MS Word, de forma que su numeración consecutiva se genera automáticamente según se trate de estilos “Titulo1, Titulo2 y Título 3”.*

*La sangría de los textos dentro de cada apartado se genera automáticamente al pulsar Intro al final de la línea de título. (Estilos Normal indentado 1, Normal indentado 2 y Normal indentado 3).*

*El índice del documento es una tabla de contenido que MS Word actualiza tomando como criterio los títulos del documento.*

*Una vez terminada su redacción debe indicarse a Word que actualice todo su contenido para reflejar el contenido definitivo.*

De la plantilla de formato del documento © & Coloriuris http://www.qualitatis.org

.

**Historial de Revisiones**

| **Fecha** | **Revisión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 07/11/2024 | 1.0 | “Requerimientos de Interfaz” | <Nombre> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Documento validado por las partes en fecha:

| Por el cliente | Por la empresa suministradora |
| --- | --- |
|  |  |
| Fdo. D./ Dña | Fdo. D./Dña |

**Contenido**

[**FICHA DEL DOCUMENTO 3**](#_heading=h.1v1yuxt)

[**CONTENIDO 4**](#_heading=h.30j0zll)

[**1**](#_heading=h.4f1mdlm) **INTRODUCCIÓN 6**

[**1.1**](#_heading=h.3znysh7) **Propósito 6**

[**1.2**](#_heading=h.2et92p0) **Alcance 6**

[**1.3**](#_heading=h.2u6wntf) **Personal involucrado 6**

[**1.4**](#_heading=h.19c6y18) **Definiciones, acrónimos y abreviaturas 6**

[**1.5**](#_heading=h.3tbugp1) **Referencias 6**

[**1.6**](#_heading=h.4d34og8) **Resumen 6**

[**2**](#_heading=h.28h4qwu) **DESCRIPCIÓN GENERAL 7**

[**2.1**](#_heading=h.nmf14n) **Perspectiva del producto 7**

[**2.2**](#_heading=h.37m2jsg) **Funcionalidad del producto 7**

[**2.3**](#_heading=h.1mrcu09) **Características de los usuarios 7**

[**2.4**](#_heading=h.46r0co2) **Restricciones 7**

[**2.5**](#_heading=h.2lwamvv) **Suposiciones y dependencias 7**

[**2.6**](#_heading=h.111kx3o) **Evolución previsible del sistema 7**

[**3**](#_heading=h.3l18frh) **REQUISITOS ESPECÍFICOS 7**

[**3.1**](#_heading=h.2xcytpi) **Requisitos comunes de los interfaces 8**

[3.1.1](#_heading=h.206ipza) Interfaces de usuario 8

[3.1.2](#_heading=h.3whwml4) Interfaces de hardware 8

[3.1.3](#_heading=h.2bn6wsx) Interfaces de software 8

[3.1.4](#_heading=h.qsh70q) Interfaces de comunicación 8

[**3.2**](#_heading=h.4k668n3) **Requisitos funcionales 8**

[3.2.1](#_heading=h.2zbgiuw) Requisito funcional 1 9

[3.2.2](#_heading=h.49x2ik5) Requisito funcional 2 9

[3.2.3](#_heading=h.2p2csry) Requisito funcional 3 9

[3.2.4](#_heading=h.147n2zr) Requisito funcional n 9

[**3.3**](#_heading=h.1egqt2p) **Requisitos no funcionales 9**

[3.3.1](#_heading=h.3ygebqi) Requisitos de rendimiento 9

[3.3.2](#_heading=h.2dlolyb) Seguridad 9

[3.3.3](#_heading=h.sqyw64) Fiabilidad 9

[3.3.4](#_heading=h.3cqmetx) Disponibilidad 9

[3.3.5](#_heading=h.1rvwp1q) Mantenibilidad 10

[3.3.6](#_heading=h.4bvk7pj) Portabilidad 10

[**3.4**](#_heading=h.vx1227) **Otros requisitos 10**

[**4**](#_heading=h.2r0uhxc) **APÉNDICES 10**

1. **Introducción**

*La introducción de la Especificación de requisitos de software (SRS) debe proporcionar una vista general de la SRS. Debe incluir el objetivo, el alcance, las definiciones y acrónimos, las referencias, y la vista general del SRS.*

ZAAP responde a una necesidad social concreta: fortalecer, digitalizar y facilitar la gestión operativa para una fundación que se dedica al rescate y protección de animales en condición de calle en Bogotá. Estas entidades suelen tener recursos limitados, procesos manuales y escasa visibilidad pública.

El sistema busca integrar en un solo entorno las funcionalidades esenciales para cubrir este ecosistema: gestión de animales, programación de voluntariado, alertas de ciudadanos, campañas de adopción y recaudación, trazabilidad del uso de donaciones. Además, debe ser accesible, visualmente atractivo y usable por personas con distintos niveles de formación digital.

* 1. **Propósito**
* *Propósito del documento*
* *Audiencia a la que va dirigido*

Este documento tiene como propósito formalizar los requisitos técnicos y funcionales del sistema ZAAP, estableciendo un entendimiento claro entre todas las partes involucradas: desarrolladores, Alcaldía de Bogotá, la fundación de protección animal, voluntarios, donantes y ciudadanos.

Servirá como base para el diseño, implementación, validación y evolución del sistema. También facilitará la trazabilidad de decisiones y permitirá establecer criterios de aceptación y pruebas.

* 1. **Alcance**
* *Identificación del producto(s) a desarrollar mediante un nombre*
* *Consistencia con definiciones similares de documentos de mayor nivel (ej. Descripción del sistema) que puedan existir*
* ***[Una descripción del entorno afectado; que proyectos se ven afectados o influenciados por esta Especificación de Requerimientos de Software.]***

El producto a desarrollar es una plataforma web denominada **ZAAP (Zona Asistida para Animales en Protección)**. Será un sistema centralizado que funcionará como herramienta gratuita para la fundación de protección animal en Bogotá.

ZAAP permitirá:

* Gestionar animales rescatados.
* Promover adopciones mediante perfiles públicos.
* Programar voluntariado (paseos, baños, interacción).
* Organizar campañas y eventos abiertos.
* Recibir donaciones trazables.
* Crear alertas ciudadanas de animales en la calle.

Se encuentra alineado con políticas distritales de bienestar animal, desarrollo digital y participación ciudadana. Su implementación será coordinada por la Alcaldía como inversión pública.

* 1. **Personal involucrado**

| Nombre | Paula Andrea Forero Borda |
| --- | --- |
| Rol | Analista, diseñador y programador |
| Categoría profesional | TSU-Informática |
| Responsabilidades | Análisis de información, diseño y programación de ZAAP |
| Información de contacto | paufor43@gmail.com |
| Aprobación | Paula Andrea Forero Borda |

* 1. **Definiciones, acrónimos y abreviaturas**

*Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.*

| **Nombre** | **Descripción** |
| --- | --- |
| **ZAAP** | Zona Asistida para Animales en Protección |
| **UX/UI** | Experiencia de usuario / Interfaz de usuario |
| **SRS** | Especificación de Requisitos de Software |
| **RSE** | Responsabilidad Social Empresarial |
| **WCAG** | Web Content Accessibility Guidelines (Pautas de accesibilidad al contenido web) |
| **OSM** | OpenStreetMap (Mapa OpenStreet) |
| **RF** | Requerimiento Funcional |
| **RNF** | Requerimiento No Funcional |

* 1. **Referencias**

| **Título** | **Referencia** |
| --- | --- |
| Standard IEEE 830 - 1998 | IEEE |
|  |  |

*Relación completa de todos los documentos relacionados en la especificación de requisitos de software, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.*

* 1. **Resumen**
* *Descripción del contenido del resto del documento*
* *Explicación de la organización del documento*

Este documento está organizado en las siguientes secciones:

* **Sección 1**: Contexto general (propósito, alcance, usuarios, estructura).
* **Sección 2**: Descripción general del sistema (arquitectura, funcionalidades, usuarios, limitaciones).
* **Sección 3**: Requisitos específicos (funcionales, no funcionales, interfaces).

Se incluye además una clasificación detallada por tipo de usuario, nivel técnico requerido, funciones dentro del sistema y restricciones tecnológicas. Este documento guiará el proceso de desarrollo de una manera estructurada.

1. **Descripción general**

*Se considera en esta parte la descripción de los factores principales que afectan al espacio de la solución. Incluya aquellos ítems como perspectiva del producto, funciones del producto, características de usuario, limitaciones, supuestos y dependencias. No se incluye en esta sección la descripción de los requerimientos.*

ZAAP es una plataforma tecnológica orientada a resolver los principales retos operativos de la fundación que rescata y protege animales. Actualmente, ésta organización trabaja con herramientas manuales, sin digitalización, y con poco alcance comunitario.

Con ZAAP, se busca profesionalizar su operación, abrir canales directos con la ciudadanía, facilitar la colaboración mutua y promover la transparencia mediante herramientas digitales accesibles, intuitivas y sostenibles. ZAAP centralizará todo el ecosistema de atención animal en una solución integral.

* 1. **Perspectiva del producto**

*Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, un diagrama que sitúe el producto dentro del sistema e identifique sus conexiones facilita la comprensión.*

ZAAP es un sistema web autónomo. No depende de sistemas mayores ni forma parte de una solución integrada más amplia. Sin embargo, contempla integraciones externas opcionales, como:

* Pasarelas de pago (Wompi, Stripe, PayU).
* Servicios de mapas (Google Maps, OpenStreetMap).
* Redes sociales (Facebook, Instagram) para difundir campañas.

El sistema será administrado por un equipo técnico designado por la Alcaldía. No será distribuido como software de código abierto.

* 1. **Funcionalidad del producto**

*Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle.*

*En ocasiones la información de esta sección puede tomarse de un documento de especificación del sistema de mayor nivel (ej. Requisitos del sistema).*

*Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor pueda entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.*

*[Si usa el modelado de casos de uso, esta sección debe contener la referencia de éste, y una descripción o resumen del modelo o del subconjunto más representativo del mismo.*

*Esto incluye una lista de nombres y breves descripciones de los casos de uso, actores, diagramas aplicables y relaciones.*

*En caso de no existir modelo de caso de uso se deben referenciar todas las descripciones existentes de las funcionalidades, ya sean minutas de reunión, correos electrónicos, etc. Es necesario agregar esas descripciones en esta sección y en el sección Referencias del documento se necesitan mencionar todos los fuentes de los requerimientos.]*

ZAAP incluirá los siguientes módulos funcionales:

* **Gestión de animales:** Registro de animales rescatados con nombre, especie, edad, estado de salud, imágenes y ubicación.
* **Adopciones:** Publicación tipo red social de animales disponibles para adopción.
* **Voluntariado:** Registro de voluntarios, programación de actividades (paseos, baños, juegos) con calendarios y sugerencias según disponibilidad.
* **Donaciones:** Visualización de necesidades específicas de la fundación (comida, medicina, infraestructura), posibilidad de donar por campaña, y seguimiento de uso.
* **Alertas ciudadanas:** Módulo para que cualquier usuario reporte un animal en calle con foto, ubicación y descripción. La fundación podrá verlas y responder.
* **Eventos y campañas:** Panel interactivo para que la fundación organice eventos abiertos al público (jornadas de adopción, colectas, ferias) con geolocalización.
* **Panel administrativo:** Gestión general, validación de usuarios, monitoreo, configuración de parámetros del sistema.
  1. **Características de los usuarios**

| Tipo de usuario | **Administrador** |
| --- | --- |
| Formación | Profesional en sistemas, gestión pública o afines. |
| Habilidades | Supervisión de contenido, validación de usuarios, configuración técnica. |
| Actividades | 1. Validar el registro de la fundación. 2. Moderar publicaciones y campañas. 3. Generar reportes generales. 4. Configurar parámetros del sistema. |

| Tipo de usuario | **Fundación de protección animal** |
| --- | --- |
| Formación | Personal administrativo o técnico con habilidades digitales básicas. |
| Habilidades | Registro de animales, manejo de campañas, gestión de voluntarios. |
| Actividades | 1. Registrar animales y actualizar su historial. 2. Crear campañas y eventos. 3. Asignar tareas a voluntarios. 4. Consultar reportes de donaciones. 5. Responder a alertas ciudadanas. |

| Tipo de usuario | **Voluntarios** |
| --- | --- |
| Formación | Formación general, sin necesidad de conocimientos técnicos. |
| Habilidades | Colaboración, gestión del tiempo, uso básico de plataformas. |
| Actividades | 1. Inscribirse a actividades programadas (paseos, baños, apoyo). 2. Confirmar asistencia y actualizar disponibilidad. 3. Consultar historial de participación. |

| Tipo de usuario | **Donantes** |
| --- | --- |
| Formación | Público general, sin requisitos formativos específicos. |
| Habilidades | Navegación web, conocimiento en el uso de medios de pago. |
| Actividades | 1. Consultar campañas activas. 2. Realizar donaciones. 3. Recibir evidencias del uso de sus aportes. |

| Tipo de usuario | **Ciudadanos** |
| --- | --- |
| Formación | Público general, sin requisitos formativos específicos. |
| Habilidades | Uso básico de apps móviles, fotografía, ubicación GPS. |
| Actividades | 1. Tomar fotos a un animal en la calle. 2. Subir ubicación y breve descripción. 3. Enviar alerta a la fundación. |

*Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.*

* 1. **Restricciones**

*Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.*

* Licencia exclusiva para la fundación autorizada por la Alcaldía.
* Uso de tecnologías modernas: HTML5, CSS3, JavaScript (React), Python, Docker, PostgreSQL.
* Cumplimiento con normas de accesibilidad WCAG 2.1 nivel AA.
* Soporte para acceso vía navegadores actuales.
* Infraestructura basada en servidores Linux con contenedores.

* 1. **Suposiciones y dependencias**

*Descripción de aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo una asunción puede ser que determinado sistema operativo está disponible para el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la SRS debería modificarse*.

**Suposiciones:**

* Los usuarios tienen acceso a conexión estable a internet.
* La fundación cuenta con al menos un equipo con navegador actualizado.
* El equipo técnico oficial estará disponible para mantenimiento.

**Dependencias:**

* APIs de terceros (Stripe, Wompi, Google Maps, OpenStreetMap).
* Plataforma cloud para despliegue inicial.
* Compatibilidad con navegadores que soporten HTML5 y JavaScript moderno.
  1. **Evolución previsible del sistema**

*Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.*

1. **Requisitos específicos**

*Esta es la sección más extensa y más importante del documento.*

*Debe contener una lista detallada y completa de los requisitos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requisitos debe ser el suficiente para que el equipo de desarrollo pueda diseñar un sistema que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen.*

*Los requisitos se dispondrán en forma de listas numeradas para su identificación, seguimiento, trazabilidad y validación (ej. RF 10, RF 10.1, RF 10.2,...).*

*Para cada requisito debe completarse la siguiente tabla:*

| Número de requisito |  | | |
| --- | --- | --- | --- |
| Nombre de requisito |  | | |
| Tipo | ☐ Requisito | ☐ Restricción | |
| Fuente del requisito |  | | |
| Prioridad del requisito | ☐ Alta/Esencial | ☐ Media/Deseado | ☐ Baja/ Opcional |

*y realizar la descripción del requisito*

*La distribución de los párrafos que forman este punto puede diferir del propuesto en esta plantilla, si las características del sistema aconsejan otra distribución para ofrecer mayor claridad en la exposición.*

* 1. **Requisitos comunes de los interfaces**

*Descripción detallada de todas las entradas y salidas del sistema de software.*

* + 1. **Interfaces de usuario**

*Describir los requisitos del interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas del interfaz. Por ejemplo, posiblemente el cliente ha especificado el estilo y los colores del producto. Describa exactamente cómo el producto aparecerá a su usuario previsto.*

### ADMINISTRADOR

**Panel del usuario:**El administrador accede a un panel de control integral que centraliza todas las funcionalidades del sistema. Tiene acceso completo a los módulos disponibles para todos los roles y puede realizar intervenciones sobre cualquier entidad o contenido. Es responsable de la supervisión, auditoría y gestión total del sistema.

**Acciones disponibles:**

* **Gestionar usuarios:** Crear, modificar, consultar, eliminar y/o deshabilitar cuentas de la fundación, ciudadanos, voluntarios y donantes.
* **Gestionar perfil fundación:** Crear, modificar, consultar, eliminar y/o deshabilitar su información organizacional (nombre, logo, contacto, redes, descripción). Consultar solicitudes, aprobar o rechazar registros institucionales
* **Gestionar rescates:** Crear, modificar, consultar, eliminar y/o deshabilitar fichas con datos (nombre, especie, estado, fotos, ubicación, historial médico). Consultar reportes recibidos por los ciudadanos, aceptar, asignar o marcar como resueltos.
* **Gestionar adopciones:** Crear, modificar, consultar, eliminar o deshabilitar publicación de adopción junto a sus datos.
* **Gestionar campañas y eventos:** Crear, modificar, consultar, eliminar y/o deshabilitar campañas según necesidades (comida, medicina, infraestructura). Consultar eventos como jornadas de adopción o actividades comunitarias.
* **Gestionar voluntariado:** Consultar postulaciones; asignar tareas; crear, modificar, eliminar y/o deshabilitar registros de participación.
* **Gestionar donaciones:** Crear, modificar, consultar, eliminar y/o deshabilitar publicaciones, comprobantes (fotos, facturas, entregas), y seguimientos de donaciones.
* **Gestionar reportes estadísticos:** Crear y consultar informes estadísticos de uso, actividad de usuarios, campañas, donaciones y participación por fundación.
* **Gestionar soporte técnico:** Consultar, responder y cerrar solicitudes de ayuda reportadas por la fundación.

### FUNDACIÓN

**Panel del usuario:** La fundación cuenta con un panel que centraliza la gestión operativa de su organización. Tiene acceso a módulos de animales, voluntarios, donaciones, eventos, campañas y alertas ciudadanas.

**Acciones disponibles:**

* **Gestionar perfil fundación:** Modificar, consultar su información organizacional (nombre, logo, contacto, redes, descripción).
* **Gestionar rescates:** Consultar fichas con datos (nombre, especie, estado, fotos, ubicación, historial médico), reportes recibidos por los ciudadanos, aceptar, asignar o marcar como resueltos.
* **Gestionar campañas y eventos:** Crear, modificar, consultar y deshabilitar campañas de donación según necesidades (comida, medicina, infraestructura) y eventos como jornadas de adopción o actividades comunitarias
* **Gestionar adopciones:** Crear, modificar, consultar y deshabilitar publicación de adopción junto a sus datos.
* **Gestionar voluntariado:** Consultar postulaciones, deshabilitar usuarios y asignar tareas.
* **Gestionar donaciones:** Consultar donación junto a los datos, crear reportes con comprobantes de seguimiento (fotos, facturas, entregas).
* **Gestionar reportes estadísticos:** Crear y consultar informes estadísticos de uso, actividad de usuarios, campañas, donaciones y participación por fundación.

### VOLUNTARIOS

**Panel del usuario:**La interfaz del voluntario está centrada en su experiencia participativa: tareas asignadas, eventos, calendario personal, historial y sugerencias.

**Acciones disponibles:**

* **Gestionar usuario:** Crear usuario, consultar y modificar información básica o eliminar/deshabilitar su cuenta en cualquier momento.
* **Gestionar voluntariado:** Consultar tareas disponibles, postularse o cancelar; confirmar asistencia; crear y modificar disponibilidad.
* **Gestionar adopciones:** Consultar publicaciones de adopción junto a sus datos.
* **Gestionar campañas y eventos:** Consultar campañas de donación y eventos como jornadas de adopción o actividades comunitarias.

### DONANTES

**Panel del usuario:**

Enfocado en facilitar la contribución y ofrecer transparencia. El donante puede explorar campañas y recibir seguimiento del uso de sus aportes.

**Acciones disponibles:**

* **Gestionar usuario:** Crear usuario, modificar información básica o eliminar/deshabilitar su cuenta en cualquier momento.
* **Gestionar adopciones:** Consultar publicaciones de adopción junto a sus datos.
* **Gestionar campañas y eventos:** Consultar campañas de donación y eventos como jornadas de adopción o actividades comunitarias.
* **Gestionar donaciones:** Consultar campañas, realizar donaciones; consultar aportes realizados de la fundación beneficiada y estado del seguimiento.

### CIUDADANO

**Panel del usuario:**Diseñado para la participación rápida y directa. El ciudadano puede crear alertas y hacer seguimiento de su estado. Registro opcional.

**Acciones disponibles:**

* **Gestionar rescates:** Crear, modificar, fichas con datos (nombre, especie, estado, fotos, ubicación, historial médico); consultar el estado de sus alertas.
* **Gestionar usuario (opcional):** Crear, modificar, eliminar y/o deshabilitar sus datos.
* **Gestionar adopciones (requiere registro):** Consultar publicaciones de adopción junto a sus datos.
* **Gestionar campañas y eventos (requiere registro):** Consultar campañas de donación y eventos como jornadas de adopción o actividades comunitarias.  
  + 1. **Interfaces de hardware**

*Especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.*

**Dispositivos del usuario final:**

* Computadoras personales (PC / laptop) con sus componentes básicos (torre, teclado mouse).
* Teléfonos móviles inteligentes (Android e iOS), que cuenten con cámara para captura de imágenes de animales y GPS para geolocalización por parte de los ciudadanos o la fundación.

* + 1. **Interfaces de software**

*Indicar si hay que integrar el producto con otros productos de software.*

*Para cada producto de software debe especificarse lo siguiente:*

* *Descripción del producto software utilizado*
* *Propósito del interfaz*
* *Definición del interfaz: contenido y formato*

*Describir los requisitos del interfaces de comunicación si hay comunicaciones con otros sistemas y cuales son las protocolos de comunicación.*

### Sistemas compatibles:

### El sistema será accesible desde navegadores modernos (Chrome, Firefox, Edge, Safari) en dispositivos con sistemas operativos Windows, macOS, Linux, Android o iOS.

### El entorno de servidor estará basado en Linux (Ubuntu Server), por su rendimiento, estabilidad y compatibilidad con servicios en la nube.

### Componentes del sistema:

1. **Base de datos:**
   * Se utilizará **PostgreSQL** como sistema de gestión de base de datos por su robustez, seguridad y compatibilidad con datos geoespaciales (ubicación GPS de alertas ciudadanas).
   * Se podrá gestionar mediante herramientas como **pgAdmin** para facilitar su uso sin conocimientos avanzados.
2. **Lenguajes y frameworks:**
   * **Backend:** Desarrollado en **Python**, utilizando frameworks como **Django**, que ofrece rapidez de desarrollo, seguridad y escalabilidad.
   * **Frontend:** Interfaz desarrollada en **JavaScript**, utilizando framework como **Vue.js** para asegurar una experiencia fluida y moderna al usuario.
3. **Contenedores:**
   * El sistema podrá ser empaquetado en **contenedores Docker** para facilitar su despliegue, escalabilidad, mantenimiento y portabilidad entre servidores.

### Integraciones externas:

1. **Pasarelas de pago:**
   * Integración con **Wompi** y **PayU**, dos de las plataformas más utilizadas en Colombia para recibir donaciones electrónicas con tarjetas débito, crédito y PSE.
2. **Geolocalización y mapas:**
   * Uso de **OpenStreetMap** para georreferenciar ubicaciones de animales reportados y eventos de adopción.
3. **Servicios de correo electrónico (API):**
   * Integración con **SendGrid**, para el envío automático de correos de confirmación, alertas y notificaciones transaccionales del sistema.
4. **Almacenamiento de archivos:**
   * Uso de **Google Cloud Storage** para almacenar de manera segura imágenes, comprobantes de donación y cualquier archivo necesario en la operación del sistema.

* 1. **Requisitos funcionales**

*Definición de acciones fundamentales que debe realizar el software al recibir in*

* + 1. **Requisito funcional 1**

| **Identificación del requerimiento:** | **RF01**. |
| --- | --- |
| **Nombre del requerimiento:** | **Iniciar sesión.** |
| **Características:** | Permite el acceso al sistema mediante credenciales únicas (correo electrónico y contraseña). |
| **Descripción del requerimiento:** | **Administrador**: Podrá ingresar al sistema con credenciales asignadas por el equipo técnico. Desde su panel podrá gestionar usuarios, contenido y configuraciones globales.  **Fundación**: Podrá iniciar sesión con las credenciales generadas al ser validada por el sistema. Desde su panel podrá gestionar animales, campañas, voluntarios y eventos.  **Voluntario**: Podrá acceder con su correo y contraseña registrados. Tendrá acceso a su historial de tareas, calendario de eventos y perfil personal.  **Donante**: Podrá crear una cuenta opcional para acceder a su historial de donaciones, actualizar datos y recibir notificaciones.  **Desarrollador OSS:** No requiere iniciar sesión dentro del sistema, ya que su interacción ocurre a través del repositorio externo (GitHub). |
| **Requerimientos NO funcionales:** | **RNF01:** Interfaz del sistema (pantalla de login clara y accesible). **RNF04:** Fiabilidad (validación segura de credenciales).  **RNF06:** Seguridad a protección de datos.  **RNF09:** Trazabilidad de actividad.  **RNF10:** Seguridad (cifrado de contraseñas con bcrypt, validación con tokens seguros). |

| **Identificación del requerimiento:** | **RF02**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión de usuarios** |
| **Características:** | Permite al administrador crear, modificar, consultar, eliminar y/o deshabilitar la cuenta de la fundación, ciudadanos, voluntarios y donantes. |
| **Descripción del requerimiento:** | 1. **Administrador:** Desde su panel podrá acceder a la gestión total de cuentas en el sistema, incluyendo la activación o desactivación de accesos. 2. **Fundación, Voluntario, Donante, Ciudadano:** Solo podrán gestionar su propia cuenta desde su perfil. |
| **Requerimientos NO funcionales:** | **RNF01:** Interfaz clara para administración de usuarios  **RNF03:** Usabilidad (accesibilidad de los formularios)  **RNF06:** Desempeño (< 2 s por acción)  **RNF10:** Seguridad (roles, validación, tokens) |

| **Identificación del requerimiento:** | **RF03**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión de perfil de fundación** |
| **Características:** | Permite crear, modificar, consultar, eliminar y/o deshabilitar información institucional (nombre, logo, contacto, redes, descripción). |
| **Descripción del requerimiento:** | 1. **Administrador:** Podrá crear, consultar y modificar perfiles institucionales, así como aprobar o rechazar nuevas solicitudes de registro eliminándola o deshabilitandola. 2. **Fundación:** Tendrá acceso para modificar o actualizar sus propios datos de perfil institucional. |
| **Requerimientos NO funcionales:** | **RNF01:** Interfaz editable, visualmente intuitiva  **RNF02:** Eficiencia de respuesta  **RNF05:** Compatibilidad en diferentes navegadores y dispositivos |

| **Identificación del requerimiento:** | **RF04**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión de rescates** |
| **Características:** | Permite gestionar fichas de animales y alertas ciudadanas. Se podrán crear, modificar, consultar, eliminar y/o deshabilitar registros. |
| **Descripción del requerimiento:** | 1. **Administrador**: Podrá crear, consultar, modificar, eliminar o deshabilitar todos los reportes y fichas, intervenir o asignar casos a la fundación. 2. **Fundación:** Podrá consultar alertas ciudadanas, asignará internamente y actualizará el estado del animal. 3. **Ciudadano:** Podrá crear alertas con foto, ubicación y descripción. Consultará el estado de la alerta. |
| **Requerimientos NO funcionales:** | **RNF01:** Interfaz con mapas y formularios integrados  **RNF04:** Validación confiable de ubicación e imagen  **RNF05:** Acceso móvil sin pérdida de funcionalidad  **RNF07:** Integración con pasarelas de pago (Wompi, PayU) |

| **Identificación del requerimiento:** | **RF05**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión de adopciones.** |
| **Características:** | Permite publicar animales disponibles para adopción, consultar perfiles y enviar solicitudes. Todos los roles pueden participar. |
| **Descripción del requerimiento:** | 1. **Administrador / Fundación**: Crear, modificar, eliminar y/o deshabilitar información de animales en adopción. 2. **Voluntario / Donante / Ciudadano:** Consultar perfiles de animales en adopción y crear solicitud con datos de contacto y motivación. |
| **Requerimientos NO funcionales:** | **RNF01:** Interfaz visual con fotos y filtros  **RNF02:** Consulta rápida por especie, ubicación, estado  **RNF05:** Compatible con dispositivos móviles |

| **Identificación del requerimiento:** | **RF06**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión de campañas y eventos.** |
| **Características:** | Permite crear, modificar, consultar, eliminar y/o deshabilitar campañas de donación y eventos públicos. |
| **Descripción del requerimiento:** | 1. **Administrador:** Podrá crear, modificar, consultar, eliminar y/o deshabilitar campañas o eventos de cualquier fundación. 2. **Fundación:** Puede crear campañas según necesidades específicas, así como crear eventos. 3. **Voluntario,** Donante, Ciudadano: Podrán consultar campañas activas, eventos cercanos y propuestas de colaboración. |
| **Requerimientos NO funcionales:** | **RNF01:** Visualización clara de fechas, ubicación y estado  **RNF02:** Rendimiento constante bajo múltiples accesos  **RNF03:** Experiencia adaptada para todos los roles  **RNF05:** Accesibilidad desde dispositivos móviles |

| **Identificación del requerimiento:** | **RF07**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión voluntariado** |
| **Características:** | Permite gestionar tareas y participación de voluntarios. La fundación y administrador pueden asignar actividades y dar seguimiento. |
| **Descripción del requerimiento:** | 1. **Administrador:** Podrá supervisar la actividad de los voluntarios y reasignar tareas si es necesario. 2. **Fundación:** Gestionará postulaciones, asignará actividades y evaluará cumplimiento. 3. **Voluntario:** Consultará tareas disponibles, se postulará, confirmará participación y modificará su disponibilidad. |
| **Requerimientos NO funcionales:** | **RNF01:** Calendario interactivo y filtros de búsqueda  **RNF02:** Tiempo de carga optimizado  **RNF03:** Facilidad para seleccionar tareas desde móvil |

| **Identificación del requerimiento:** | **RF08**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión de donaciones** |
| **Características:** | Permite gestionar campañas de donación, registrar aportes y subir evidencia del uso de fondos. Los donantes podrán consultar su historial. |
| **Descripción del requerimiento:** | 1. **Administrador:** Accederá a todas las campañas y podrá auditar evidencias. 2. **Fundación:** Publicará campañas, registrará donaciones recibidas y subirá comprobantes. 3. **Donante:** Consultará campañas activas, donará y podrá ver en qué se usó su aporte |
| **Requerimientos NO funcionales:** | **RNF03:** Confirmaciones en tiempo real  **RNF04:** Validación segura y cifrada  **RNF06:** Seguridad a protección de datos.  **RNF09:** Trazabilidad de actividad.  **RNF07:** Integración con pasarelas de pago (Wompi, PayU) |

| **Identificación del requerimiento:** | **RF09**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión de reportes estadísticos.** |
| **Características:** | Permite al administrador generar reportes estadísticos del sistema: campañas, actividad de usuarios, aportes, participación. |
| **Descripción del requerimiento:** | 1. **Administrador:** Accede al panel de estadísticas y podrá exportar informes segmentados por fechas, campañas, tipo de usuario o actividad. |
| **Requerimientos NO funcionales:** | **RNF01:** Interfaz de gráficos comprensible **RNF06:** Carga eficiente (< 3 s por reporte) |

| **Identificación del requerimiento:** | **RF10**. |
| --- | --- |
| **Nombre del requerimiento:** | **Gestión de soporte técnico.** |
| **Características:** | Permite registrar y resolver incidencias del sistema. La fundación podrá solicitar ayuda técnica y el administrador gestionará su resolución. |
| **Descripción del requerimiento:** | 1. **Administrador:** Visualizará tickets activos, responderá, marcará como resueltos o archivará. 2. **Fundación:** Podrá enviar solicitudes de soporte y hacer seguimiento al estado del caso. |
| **Requerimientos NO funcionales:** | **RNF01:** Interfaz clara de comunicación  **RNF05:** Compatibilidad móvil y de escritorio |

* 1. **Requisitos no funcionales**

| **Identificación del requerimiento:** | **RNF01.** |
| --- | --- |
| **Nombre del requerimiento:** | **Interfaz del sistema** |
| **Características:** | Diseño visual, responsivo, intuitivo y adaptado a cada tipo de usuario según su rol. |
| **Descripción del requerimiento:** | El sistema debe contar con una interfaz clara, sencilla y fácil de usar. Las pantallas deben estar organizadas según el tipo de usuario (administrador, fundación, voluntario, donante, ciudadano) y ser accesibles desde cualquier dispositivo (móvil, tablet o escritorio). |
| **Prioridad del requerimiento:** | Alta |

| **Identificación del requerimiento:** | **RNF02.** |
| --- | --- |
| **Nombre del requerimiento:** | **Rendimiento del sistema** |
| **Características:** | Velocidad de respuesta en cada módulo. |
| **Descripción del requerimiento:** | Las acciones frecuentes (consultas, registros, edición, navegación) deben ejecutarse en menos de 2 segundos bajo condiciones normales, asegurando fluidez y satisfacción del usuario. |
| **Prioridad del requerimiento:** | Alta |

| **Identificación del requerimiento:** | **RNF03.** |
| --- | --- |
| **Nombre del requerimiento:** | **Usabilidad y experiencia del usuario** |
| **Características:** | Accesibilidad para usuarios sin experiencia técnica. |
| **Descripción del requerimiento:** | El sistema debe estar diseñado para ser utilizado por personas con habilidades básicas en el uso de tecnología, garantizando una experiencia coherente, con instrucciones visibles, flujos simples y formularios claros. |
| **Prioridad del requerimiento:** | Alta |

| **Identificación del requerimiento:** | **RNF04.** |
| --- | --- |
| **Nombre del requerimiento:** | **Fiabilidad y validación** |
| **Características:** | Seguridad en formularios, autenticación y procesos de datos. |
| **Descripción del requerimiento:** | Cada formulario o acción crítica debe estar validada tanto en el cliente (navegador) como en el servidor para evitar errores, mal uso del sistema o vulnerabilidades. |
| **Prioridad del requerimiento:** | Alta |

| **Identificación del requerimiento:** | **RNF05.** |
| --- | --- |
| **Nombre del requerimiento:** | **Accesibilidad y compatibilidad** |
| **Características:** | Inclusión y funcionamiento en múltiples dispositivos y navegadores. |
| **Descripción del requerimiento:** | La plataforma debe cumplir con los criterios de accesibilidad **WCAG 2.1 nivel AA** y ser completamente funcional en los navegadores más comunes y en sistemas operativos móviles y de escritorio. |
| **Prioridad del requerimiento:** | Media |

| **Identificación del requerimiento:** | **RNF06.** |
| --- | --- |
| **Nombre del requerimiento:** | **Seguridad** |
| **Características:** | Protección de datos personales y autenticación segura. |
| **Descripción del requerimiento:** | Las contraseñas deberán cifrarse con **bcrypt**. El sistema debe usar validación con tokens seguros para las sesiones, y aplicar protocolos de seguridad para evitar accesos no autorizados. |
| **Prioridad del requerimiento:** | Alta |

| **Identificación del requerimiento:** | **RNF07.** |
| --- | --- |
| **Nombre del requerimiento:** | **Integración con servicios externos** |
| **Características:** | Conectividad con plataformas externas mediante APIs. |
| **Descripción del requerimiento:** | El sistema debe integrarse correctamente con **Wompi, PayU** (pasarelas de pago), **Google Maps/OpenStreetMap** (mapas), y **SendGrid** (correo electrónico), usando autenticación segura (API keys, tokens, etc.). |
| **Prioridad del requerimiento:** | Alta |

| **Identificación del requerimiento:** | **RNF08.** |
| --- | --- |
| **Nombre del requerimiento:** | **Escalabilidad** |
| **Características:** | Capacidad de crecimiento según demanda. |
| **Descripción del requerimiento:** | El sistema debe estar diseñado para poder escalar horizontalmente en servidores cloud y contenedores Docker, permitiendo más usuarios o carga sin disminuir el rendimiento. |
| **Prioridad del requerimiento:** | Media |

| **Identificación del requerimiento:** | **RNF09.** |
| --- | --- |
| **Nombre del requerimiento:** | **Trazabilidad** |
| **Características:** | Registro de acciones realizadas por los usuarios. |
| **Descripción del requerimiento:** | El sistema debe almacenar automáticamente logs de actividad relevantes (quién hizo qué, cuándo y dónde), con posibilidad de exportarlos por parte del administrador. |
| **Prioridad del requerimiento:** | alta |

| **Identificación del requerimiento:** | **RNF10.** |
| --- | --- |
| **Nombre del requerimiento:** | **Escalabilidad** |
| **Características:** | Acceso continuo al sistema con tolerancia a fallos. |
| **Descripción del requerimiento:** | El sistema debe estar disponible al menos el 99.5% del tiempo al mes. Además, debe contar con respaldo de base de datos automático (mínimo diario) y mecanismos de recuperación ante fallos. |
| **Prioridad del requerimiento:** | Alta |

* + 1. **Requisitos de rendimiento**

*Especificación de los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.*

*Todos estos requisitos deben ser mensurables. Por ejemplo, indicando “el 95% de las transacciones deben realizarse en menos de 1 segundo”, en lugar de “los operadores no deben esperar a que se complete la transacción”.*

El sistema debe responder a cualquier acción del usuario (consulta, edición, carga de datos) en un tiempo no mayor a **2 segundos** bajo condiciones normales de uso.

Debe soportar **al menos 500 usuarios concurrentes** sin pérdida de rendimiento perceptible.

Las búsquedas de animales, campañas, eventos y usuarios deben realizarse en menos de **1 segundo** con filtros activos.

Las cargas de imágenes o archivos (donaciones, reportes, evidencias) no deben exceder los **5 segundos** por operación.

El sistema debe mantener un tiempo de respuesta estable al ser accedido desde redes móviles (4G/5G) y conexiones residenciales estándar (>5 Mbps).

* + 1. **Seguridad**

*Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requisitos pueden especificar:*

* *Empleo de técnicas criptográficas.*
* *Registro de ficheros con “logs” de actividad.*
* *Asignación de determinadas funcionalidades a determinados módulos.*
* *Restricciones de comunicación entre determinados módulos.*
* *Comprobaciones de integridad de información crítica.*

El acceso al sistema estará restringido por autenticación mediante **correo electrónico y contraseña cifrada con bcrypt**.

Las sesiones activas se mantendrán mediante **tokens seguros (JWT)** con expiración programada.

Se utilizarán conexiones **HTTPS** con certificados válidos para proteger toda la información en tránsito.

Se aplicará validación en todos los formularios, tanto del lado cliente como del servidor, para prevenir inyecciones, ataques CSRF, y entradas maliciosas.

El sistema limitará el número de intentos de acceso fallido por usuario y por IP.

Las acciones críticas del sistema (modificación de datos, aprobaciones, donaciones) serán registradas y auditadas por el administrador.

* + 1. **Fiabilidad**

*Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permisibles.*

El sistema debe garantizar **alta estabilidad** durante el uso prolongado.

Las operaciones esenciales como registro de animales, donaciones, adopciones y alertas ciudadanas deben funcionar con **tasa de error inferior al 1%**.

Se deben implementar **pruebas automatizadas** para cada módulo crítico y validación manual antes de cada despliegue.

El sistema debe conservar la **integridad de los datos** ante errores de usuario o interrupciones inesperadas.

* + 1. **Disponibilidad**

*Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad.*

La plataforma debe estar disponible **al menos el 99.5% del tiempo mensual**.

Se deben establecer mecanismos automáticos de **respaldo diario** de la base de datos y archivos adjuntos.

En caso de mantenimiento planificado, se notificará con mínimo 24 horas de anticipación.

En caso de falla crítica, se activará un **plan de recuperación automática**, con restablecimiento máximo en 2 horas.

La arquitectura del sistema debe prever el despliegue en **infraestructura cloud escalable** (Docker, balanceadores de carga, réplicas de base de datos si aplica).

* + 1. **Mantenibilidad**

*Identificación del tipo de mantenimiento necesario del sistema.*

*Especificación de quien debe realizar las tareas de mantenimiento, por ejemplo usuarios, o un desarrollador.*

*Especificación de cuando debe realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanales*

El sistema estará construido en una arquitectura modular que permita mantener cada componente (módulo de usuarios, animales, donaciones, etc.) de forma independiente.

Se documentará el código fuente, endpoints, estructuras de base de datos y flujos funcionales para facilitar futuras mejoras.

Se habilitará un entorno de pruebas (staging) para validar nuevas funcionalidades antes de publicarlas en producción.

Los cambios al sistema deben poder desplegarse sin detener el funcionamiento general, usando prácticas como **CI/CD y contenedores (Docker)**.

Se designará un equipo de mantenimiento responsable de aplicar correcciones y mejoras periódicas.

* + 1. **Portabilidad**

*Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas y entornos. Pueden incluirse:*

* *Porcentaje de componentes dependientes del servidor.*
* *Porcentaje de código dependiente del servidor.*
* *Uso de un determinado lenguaje por su portabilidad.*
* *Uso de un determinado compilador o plataforma de desarrollo.*
* *Uso de un determinado sistema operativo.*

* 1. **Otros requisitos**

*Cualquier otro requisito que no encaje en ninguna de las secciones anteriores.*

*Por ejemplo:*

*Requisitos culturales y políticos*

*Requisitos Legales*

1. **Apéndices**

*Pueden contener todo tipo de información relevante para la SRS pero que, propiamente, no forme parte de la SRS.*