**[MascotaConectada]**

**(DAS) Documento Arquitectura Sistema**

**Versión 1.0**



**Identificación de Documento**

| **Identificación** |  |
| --- | --- |
| **Proyecto** | MascotaConectada |
| **Versión** | 1.0 |

| **Documento mantenido por** | Jaime Canales |
| --- | --- |
| **Fecha de última revisión** |  |
| **Fecha de próxima revisión** |  |

| **Documento aprobado por** | Jaime Canales |
| --- | --- |
| **Fecha de última aprobación** |  |

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 16-09-2025 | 1.0 | Se inicia documentación | Jaime Canales |
| 27-09-2025 | 1.1 | Avance desde punto n°2 | Jaime Canales |
| 29-09-2025 | 1.2 | Avances punto n°3 | Felipe Muñoz |
|  |  |  |  |

**Tabla de Contenidos**

[**1**](#_heading=h.gjdgxs) **Introducción 3**

[1.1](#_heading=h.30j0zll) Contexto del Problema 3

[1.2](#_heading=h.1fob9te) Propósito 3

[1.3](#_heading=h.3znysh7) Ámbito 3

[1.4](#_heading=h.2et92p0) Definiciones, acrónimos y abreviaciones 3

[1.5](#_heading=h.tyjcwt) Referencias 3

[1.6](#_heading=h.3dy6vkm) Resumen ejecutivo 3

[1.7](#_heading=h.1t3h5sf) Representación 4

[**2**](#_heading=h.4d34og8) **Metas y Restricciones de la Arquitectura 5**

[2.1](#_heading=h.2s8eyo1) Metas de la arquitectura 5

[2.2](#_heading=h.17dp8vu) Restricciones de la Arquitectura 5

[2.3](#_heading=h.3rdcrjn) Otros antecedentes y consideraciones 5

[**3**](#_heading=h.26in1rg) **Vista de Escenarios 6**

[3.1](#_heading=h.lnxbz9) Modelo de Casos de Uso 6

[3.2](#_heading=h.35nkun2) Casos de Usos Extendidos 6

[3.3](#_heading=h.1ksv4uv) Especificación de los Escenarios de Calidad Relevantes 7

[**4**](#_heading=h.44sinio) **Vista de Procesos 8**

[**5**](#_heading=h.2jxsxqh) **Vista Lógica 9**

[5.1](#_heading=h.z337ya) Parte Estructural ( Diagrama de Clases y Diagrama Relacional) 9

[*5.1.1*](#_heading=h.3j2qqm3) *Descripción de Clases 9*

[*5.1.2*](#_heading=h.1y810tw) *Descripción de Tablas 10*

[5.2](#_heading=h.4i7ojhp) Parte Dinámica (Diagrama de Secuencias) 11

[**6**](#_heading=h.2xcytpi) **Vista de Desarrollo o Despliegue 12**

[**7**](#_heading=h.1ci93xb) **Vista Fisica 13**

[**8**](#_heading=h.3whwml4) **Decisiones de Diseño y Selección de Alternativas 14**

[**9**](#_heading=h.2bn6wsx) **Análisis de Reutilización 15**

1. **Introducción**
   1. **Contexto del Problema**

En la actualidad, muchas familias cuentan con mascotas, pero carecen de una herramienta tecnológica integrada que permita gestionar de manera sencilla el historial médico, citas veterinarias, recordatorios de vacunas y geolocalización de clínicas. Esta falta de digitalización puede provocar pérdida de información, citas olvidadas y dificultades para encontrar atención oportuna.

* 1. **Propósito**

El propósito de este documento es definir la arquitectura de software del sistema MascotaConectada, detallando sus componentes, relaciones, decisiones técnicas y lineamiento de calidad. Este documento servirá como guía de referencia para el equipo de desarrollo, QA y stakeholders.

* 1. **Ámbito**

MascotaConectada será una plataforma web compuesta por:

* Frontend: Angular 17 + TailwindCSS.
* Backend: Django REST Framework (Python 3.11).
* Base de Datos: MySQL 8.0.
* Despliegue: Servidor en la nube con dominio propio y certificado SSL.

El sistema cubrirá funciones como autenticación de usuarios, registro de mascotas, gestión de historial médico, agenda y recordatorios, y geolocalización de veterinarias cercanas.

* 1. **Definiciones, acrónimos y abreviaciones**

| **ACRONIMO** | **DESCRIPCION** |
| --- | --- |
| *API* | Application Programming Interface. |
| *CRUD* | Create, Read, Update, Delete. |
| *ERS* | Especificación de Requerimientos de Software. |
| *HA* | Alta disponibilidad. |
| *SSL* | Secure Sockets Layer. |

* 1. **Referencias**

A continuación, se listan las referencias a otros documentos :

* [**ERS**](https://docs.google.com/document/d/1r0iGYH4H_DMTBTl58nN-CDSuMs71xV5Y/edit?usp=drive_link&ouid=103522799761880098077&rtpof=true&sd=true)
* [**Caso de Uso**](https://docs.google.com/document/d/1BpeSuUOeQsYUEHw8rIpglMrFctycuTow/edit?usp=drive_link&ouid=103522799761880098077&rtpof=true&sd=true)
* [**Modelo Relacional**](https://drive.google.com/file/d/1Jhh72szSR8KS-n_uaoN0ogh71LPzCTwi/view?usp=drive_link)
* [**Matriz RACI**](https://docs.google.com/spreadsheets/d/1Bo3Wl1PMr5zfLt211xJOU0pMnuJfV0c1dbDiSVOaHHk/edit?usp=drive_link)
  1. **Resumen ejecutivo**

El sistema MascotaConectada se construirá bajo arquitectura cliente-servidor moderna, con separación clara de capas (frontend, backend, base de datos), apoyada en servicios de nube para garantizar escalabilidad y disponibilidad. Se priorizará seguridad, mantenibilidad y experiencia de usuario, considerando restricciones de tiempo, presupuesto y alcance definidos en el proyecto.

* 1. **Representación**

La arquitectura del sistema MascotaConectada está representada siguiendo el enfoque del framework 4+1 y las recomendaciones del proceso unificado. Las vistas incluidas en esta versión del documento son:

* **Vista de Escenarios**: Describe los casos de uso más significativos, presenta los actores principales (usuarios, veterinarios, administrador) y sus casos de uso asociados. Asimismo, define los escenarios de calidad más relevantes (disponibilidad, seguridad, rendimiento)
* **Vista de Procesos**: Describe los procesos de negocio y flujos de ejecución del sistema (por ejemplo, registro de usuario, agendamiento de citas, registro de historial médico), así como las relaciones de comunicación y sincronización entre ellos.
* **Vista Lógica**: Presenta la arquitectura lógica del sistema en distintos niveles de refinamiento, identificando módulos principales (cuentas, mascotas, agenda, historial médico, geolocalización) y sus responsabilidades.
* **Vista de Desarrollo o Despliegue**: Describe los componentes de software construidos y sus dependencias, incluyendo las capas Angular (frontend), Django REST (backend), MySQL (base de datos) y servicios externos (Google Maps API).
* **Vista Física**: Expone las restricciones tecnológicas y de infraestructura, como el despliegue en la nube con dominio propio, el uso de contenedores, la base de datos en MySQL y la necesidad de aplicar normativas de seguridad y protección de datos.

1. **Metas y Restricciones de la Arquitectura**

A continuación, se revisan las metas y restricciones de la arquitectura.

* 1. **Metas de la arquitectura**

La arquitectura del sistema MascotaConectada se diseña considerando los siguientes objetivos principales:

* **Seguridad:** Garantizar la protección de los datos sensibles de usuarios, mascotas e historial médico mediante autenticación segura, cifrado y buenas prácticas de desarrollo.
* **Disponibilidad:** Asegurar un servicio estable con al menos un **99.9% de disponibilidad**, reduciendo al mínimo los tiempos de inactividad.
* **Escalabilidad:** Permitir que la solución pueda crecer horizontal y verticalmente, adaptándose a una mayor cantidad de usuarios y volumen de datos sin afectar el rendimiento.
* **Mantenibilidad:** Implementar una arquitectura modular y desacoplada que facilite la corrección de errores, la incorporación de mejoras y el mantenimiento evolutivo.
* **Usabilidad:** Ofrecer una experiencia de usuario simple, intuitiva y accesible, que permita a cualquier usuario gestionar de manera eficiente la información de sus mascotas.
* **Compatibilidad:** Asegurar interoperabilidad entre frontend Angular, backend Django REST y la base de datos MySQL, junto con servicios externos como APIs de geolocalización.
  1. **Restricciones de la Arquitectura**

El diseño de la arquitectura también debe considerar las siguientes restricciones:

* **Tecnológicas:**
  + Uso obligatorio de **Angular** para el frontend, **Django REST Framework** para el backend y **MySQL** como base de datos.
  + Despliegue en **nube con dominio propio y certificado SSL**.
  + Uso de APIs externas limitadas a servicios de **geolocalización de clínicas veterinarias**.
* **Económicas:**
  + Presupuesto aproximado de **30 millones CLP**, considerando costos de infraestructura, dominio, seguridad y un margen del 15% para rentabilidad.
* **De gestión:**
  + Aplicación de **metodología en cascada**, en la cual cada fase debe completarse antes de iniciar la siguiente.
  + Tiempo total estimado: **76 jornadas laborales** (8 horas por jornada).
* **De seguridad:**
  + Cumplimiento de la **Ley 19.628 (Chile)** sobre protección de la vida privada.
  + Implementación obligatoria de cifrado en tránsito (HTTPS/SSL) y buenas prácticas de almacenamiento de contraseñas.
* **De despliegue:**
  + El sistema debe ser accesible desde navegadores modernos y tener la capacidad de evolucionar a versiones móviles en futuras fases del proyecto.
  1. **Otros antecedentes y consideraciones**

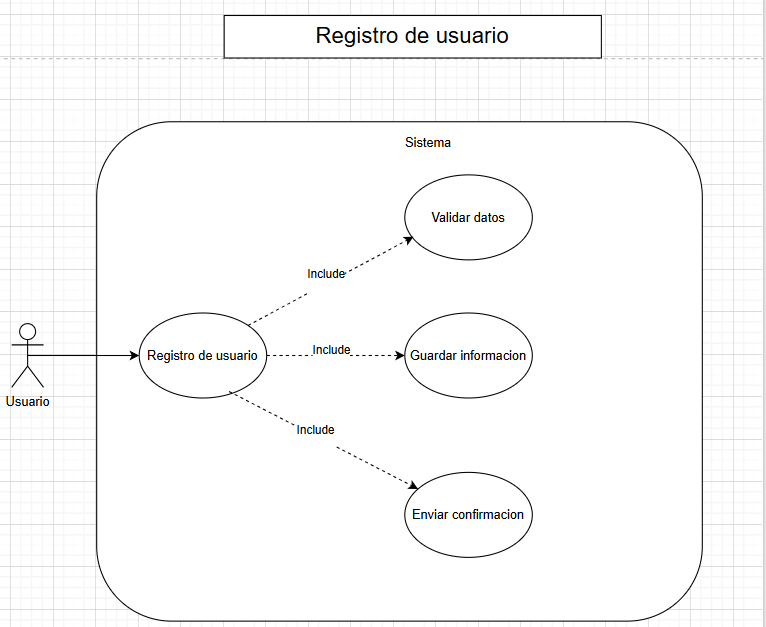
La empresa desarrolladora del proyecto MascotaConectada cuenta con un marco de trabajo tecnológico que considera los siguientes componentes para satisfacer los requerimientos arquitectónicos:

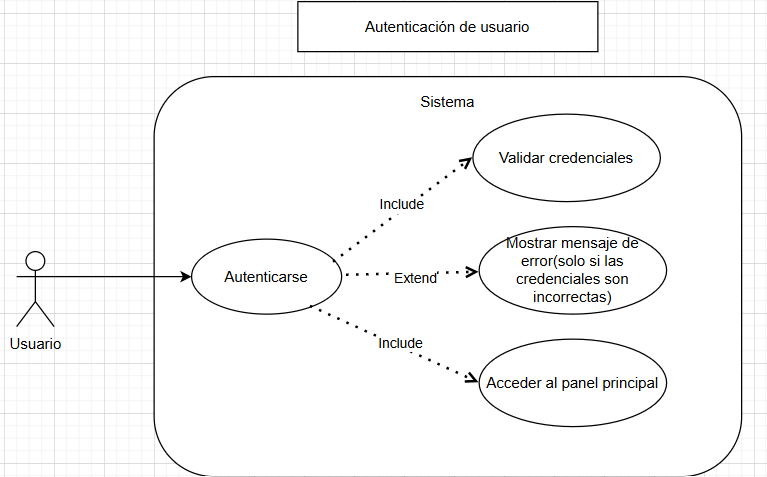
* Framework de Desarrollo Web:  
   Se adopta el uso combinado de Angular (frontend) y Django REST Framework (backend). Esta elección permite la encapsulación y modularización de componentes, lo que facilita la mantenibilidad y escalabilidad del sistema. Angular proporciona una arquitectura basada en componentes reutilizables y fácil integración con servicios, mientras que Django REST asegura un backend robusto, con soporte nativo a buenas prácticas de seguridad y serialización de datos.
* Framework de Seguridad:  
   La arquitectura contempla el uso de django-cors-headers, django-rest-framework auth y buenas prácticas de OWASP para garantizar la meta de seguridad. Esto incluye cifrado de contraseñas con algoritmos de hash seguros (ej. Argon2/Bcrypt), uso de HTTPS con certificado SSL y la posibilidad de integrar autenticación basada en tokens (JWT) para sesiones seguras.
* Framework de Estilos y Experiencia de Usuario:  
   En el frontend, se emplea TailwindCSS como framework de estilos para asegurar un diseño consistente, moderno y responsive, cumpliendo con los criterios de usabilidad definidos en las metas del sistema.

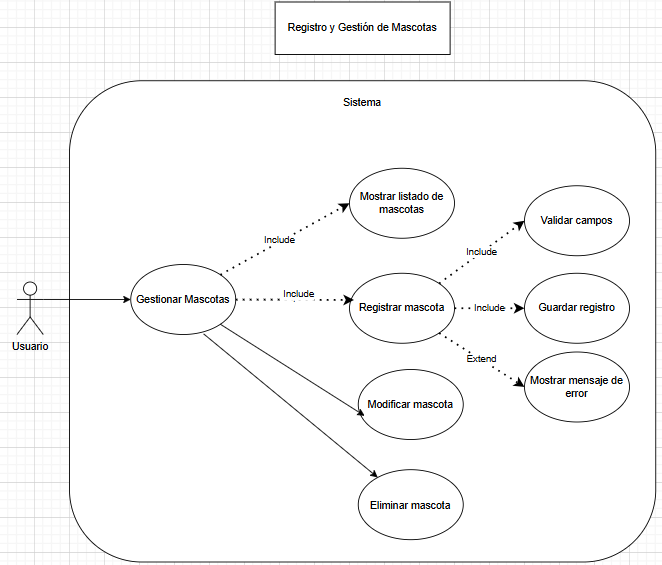
Con esta combinación de frameworks y librerías, se soportan de forma directa las metas de mantenibilidad, performance, seguridad y usabilidad, que son pilares centrales de la arquitectura del sistema.

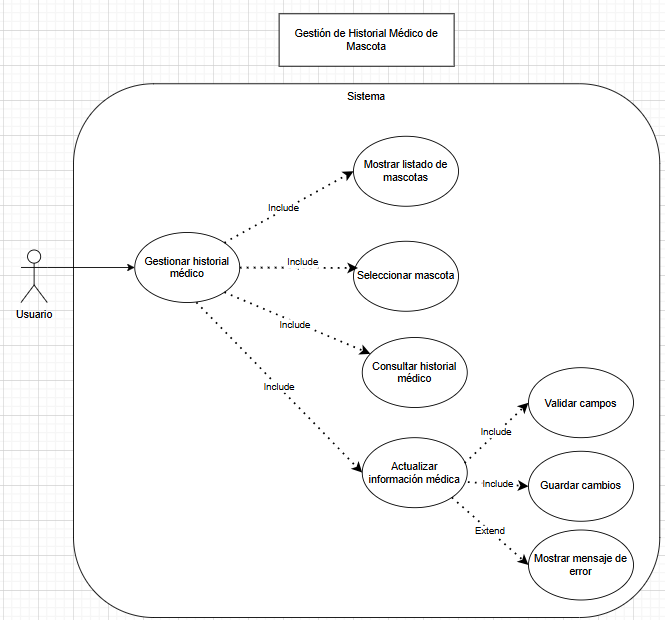
1. **Vista de Escenarios**

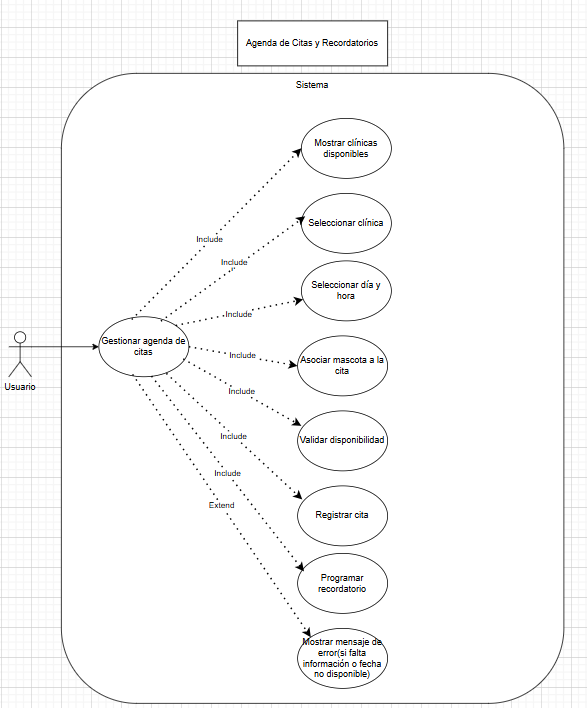
CU001 – Registro de Usuario.



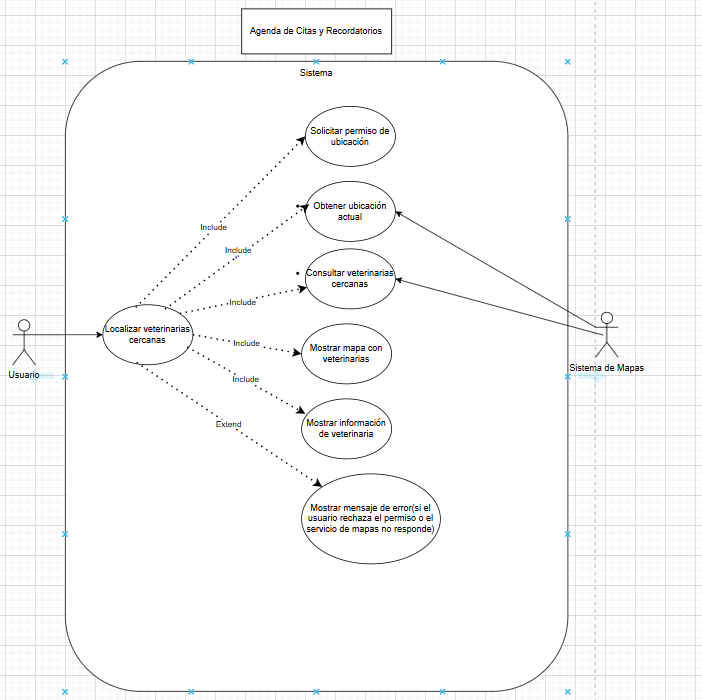
CU 002 – Autenticación de Usuario  


CU 005 – Registro y Gestión de Mascotas  


CU 006 – Gestión de Historial Médico de Mascota  
  


CU 007 – Agenda de Citas y Recordatorios  


CU 008 – Localización de Veterinarias Cercanas



* 1. **Modelo de Casos de Uso**

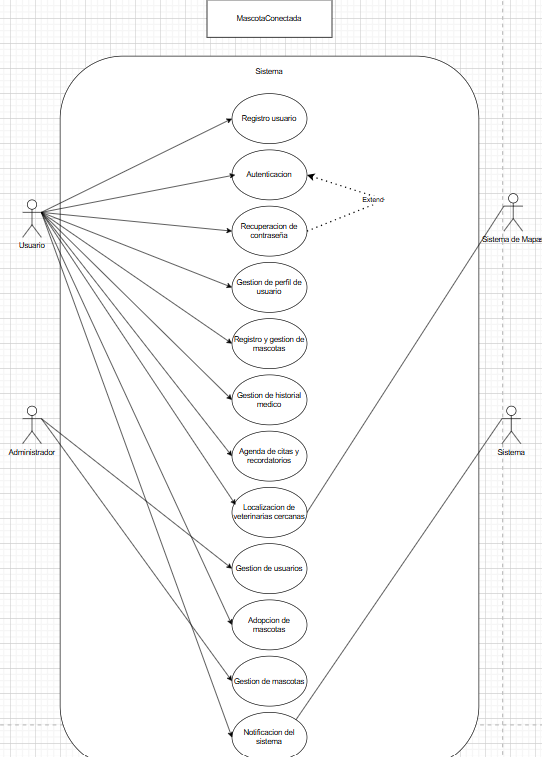
Usuario Dueño de Mascota: Persona que registra y administra la información de sus mascotas.

Veterinario: Profesional de salud animal que interactúa con el sistema para registrar diagnósticos, tratamientos o controles.

Administrador del Sistema: Usuario con permisos para gestionar la plataforma, usuarios y configuraciones generales.

Agregar el modelo de caso uso general del sistema

**Ilustración 1: Diagrama de Caso Uso General del Sistema**



* 1. **Casos de Usos Extendidos**

[Documento Caso Uso Extendido.docx](https://docs.google.com/document/d/1BpeSuUOeQsYUEHw8rIpglMrFctycuTow/edit?usp=drive_link&ouid=100006434380650039960&rtpof=true&sd=true)

A continuación, se listan los casos de uso relevantes, los cuales pueden ser encontrados con su especificación detallada en el documento “Casos de Uso Extendido”.

| **Código** | **Nombre** | **Actores** | **Prioridad** |
| --- | --- | --- | --- |
| CU-001 | Registro de usuario | Usuario | Alta |
| CU-002 | Autenticacion de usuario | Usuario | Alta |
| CU-003 | Recuperación de contraseña | Usuario | Media |
| CU-004 | Gestion de perfil de usuario | Usuario | Media |
| CU-005 | Registro y gestión de mascotas | Usuario | Alta |
| CU-006 | Gestión de historial médico de mascota | Usuario | Media |
| CU-007 | Agenda de citas y recordatorios | Usuario | Media |
| CU-008 | Localización de veterinarias cercanas | Usuario,Sistema de mapas | Media |
| CU-009 | Gestion de usuarios | Administrador | Baja |
| CU-010 | Módulo de adopción de mascotas | Usuario | Baja |
| CU-011 | Gestion de mascotas | Administrador | Baja |
| CU-012 | Notificaciones del sistema | Usuario,sistema | Media |

* 1. **Especificación de los Escenarios de Calidad Relevantes**

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

| Identificador: EC-01 | | |
| --- | --- | --- |
| Escenario(s): | | Disponibilidad del sistema |
| Atributos de Calidad relevantes: | | Disponibilidad |
| Componentes del Escenario | Estímulos: | Fallo inesperado del servidor principal. |
| Fuente del estimulo | Infraestructura de nube. |
| Ambiente: | Producción, carga normal de usuarios. |
| Artefacto: | Backend (Django REST) y Base de Datos (MySQL). |
| Respuesta: | El sistema redirige automáticamente las peticiones hacia la réplica activa. |
| Medida de Respuesta | Tiempo de recuperación ≤ 1 minuto, SLA ≥ 99.9%. |

| Identificador: EC-02 | | |
| --- | --- | --- |
| Escenario(s): | | Seguridad de la Información |
| Atributos de Calidad relevantes: | | Seguridad |
| Componentes del Escenario | Estímulos: | Un usuario intenta interceptar datos sensibles durante una transacción. |
| Fuente del estimulo | Cliente externo malicioso. |
| Ambiente: | Producción, conexión desde navegador web. |
| Artefacto: | Canal de comunicación entre frontend y backend. |
| Respuesta: | Los datos son transmitidos únicamente a través de HTTPS con cifrado TLS 1.2+. |
| Medida de Respuesta | 100% de las comunicaciones cifradas, credenciales en hash seguro (bcrypt/argon2). |

| Identificador: EC-03 | | |
| --- | --- | --- |
| Escenario(s): | | Rendimiento en Consultas |
| Atributos de Calidad relevantes: | | Rendimiento |
| Componentes del Escenario | Estímulos: | El usuario consulta el historial médico de una mascota con gran cantidad de registros. |
| Fuente del estimulo | Usuario final autenticado. |
| Ambiente: | Producción, carga de hasta 1000 usuarios concurrentes. |
| Artefacto: | Backend (API de historial) y Base de Datos. |
| Respuesta: | El sistema devuelve la información solicitada en tiempo óptimo gracias a índices y cache. |
| Medida de Respuesta | Tiempo de respuesta ≤ 2 segundos en 90% de los casos. |

| Identificador: EC-04 | | |
| --- | --- | --- |
| Escenario(s): | | Usabilidad de la Plataforma |
| Atributos de Calidad relevantes: | | Usabilidad |
| Componentes del Escenario | Estímulos: | Un usuario accede desde un dispositivo móvil (pantalla pequeña) para agendar una cita.. |
| Fuente del estimulo | Usuario dueño de mascota |
| Ambiente: | Producción, acceso desde navegador móvil. |
| Artefacto: | Interfaz web Angular con diseño responsive (TailwindCSS). |
| Respuesta: | El sistema ajusta la visualización de manera responsiva, manteniendo la funcionalidad completa. |
| Medida de Respuesta | El 95% de las pantallas críticas deben superar pruebas de usabilidad (tiempo de tarea ≤ 3 clics). |

| Identificador: EC-05 | | |
| --- | --- | --- |
| Escenario(s): | | Mantenibilidad del Sistema |
| Atributos de Calidad relevantes: | | Mantenibilidad |
| Componentes del Escenario | Estímulos: | El equipo de desarrollo detecta un bug en el módulo de agenda médica. |
| Fuente del estimulo | Equipo de QA / desarrolladores. |
| Ambiente: | Etapa de desarrollo y pruebas. |
| Artefacto: | Código fuente backend (Django REST) y frontend (Angular). |
| Respuesta: | Se corrige el bug mediante un cambio en el código y se valida con pruebas automatizadas. |
| Medida de Respuesta | El tiempo de corrección no debe superar 1 jornada laboral; cobertura de pruebas ≥ 70%. |

1. **Vista de Procesos**

**Ilustración 2: Diagramas de Actividades**

Carpeta con diagramas de actividades.

[Diagramas actividades](https://drive.google.com/drive/folders/1EKp98ZcOcqnNGxUGwAZ0q7Tw5z3wuebv?usp=sharing)

1. **Vista Lógica**

A continuación, se presenta una vista lógica de la aplicación expresado en tres diagramas, uno de ellos que muestra la parte estructural o estática de la aplicación (clases) y a la base de datos (modelo relacional).otra vista que representa la parte dinámica (secuencias).

* 1. **Parte Estructural ( Diagrama de Clases y Diagrama Relacional)**

**Ilustración 3: Diagrama de Clases**

* + 1. *Descripción de Clases*

| **Código** | **Nombre** | **Descripción** |
| --- | --- | --- |
| CL-001 | Exportar saldos y puntos a vencer |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Ilustración 4: Diagrama de Base Datos (Relacional)**

* + 1. *Descripción de Tablas*

| **Código** | **Nombre** | **Descripción** |
| --- | --- | --- |
| TB-001 | CLIENTE |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. **Parte Dinámica (Diagrama de Secuencias)**

**Ilustración 5: Diagramas de Secuencias**

1. **Vista de Desarrollo o Despliegue**

En esta vista se describen las componente o modulos en las cuales se dividirá o implementará el sistema

**Ilustración 6: Diagrama de componentes**

1. **Vista Fisica**

En esta vista se despliegan los nodos que participan con el sistema.

**Ilustración 7: Diagrama de Despliegue**

1. **Decisiones de Diseño y Selección de Alternativas**

Las principales decisiones arquitectónicas se tomaron en consideración de la restricción Tiempo de Construcción. Dado que el proyecto debe implementarse en un tiempo ajustado y sin holguras, se privilegió la adopción de una arquitectura conocida y que presente un bajo riesgo en su implementación.

Asimismo, la arquitectura se modularizó y cada módulo serán testeables unitariamente de forma de asegurar que cada pieza tenga una baja tasa de fallas.

Un segundo proposito que fue considerado en la seleccion de alternativa son :

1. **Análisis de Reutilización**

* Ejemplo : Se utilizara el codigo del mantenedor de productos en clientes
* Ejemplo: Se utilizaran plantillas gráficas con los mismos colores y logotipos en el mantendor de clientes, productos, ventas.