Logotipo

Descripción generada automáticamente





Contenido

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

[**2.**](#_heading=h.30j0zll) **Propósito** 3

[**3.**](#_heading=h.1fob9te) **Alcance** 3

[**4.**](#_heading=h.3znysh7) **Equipo de Trabajo-Actores del Desarrollo** 4

[**5.**](#_heading=h.2et92p0) **Recomendaciones de conformidad con esta práctica.** 4

[Para asegurar que la arquitectura del sistema cumpla con los requisitos y estándares establecidos, se recomiendan las siguientes prácticas: 4](#_heading=h.tyjcwt)

[1. Implementar revisiones de arquitectura cada xxxx para asegurar que se ajusten a los cambios de requisitos. 4](#_heading=h.3dy6vkm)

[2. Utilizar patrones de diseño desacoplados como MVC para modularidad y escalabilidad o por capas. 4](#_heading=h.1t3h5sf)

[3. Adoptar metodologías ágiles como Scrum para la entrega continua de mejoras y funcionalidades. 4](#_heading=h.4d34og8)

[**6.**](#_heading=h.2s8eyo1) **Referencias y estándares aplicables a este documento:** 5

[**7.**](#_heading=h.17dp8vu) **Definiciones, acrónimos y abreviaciones.** 5

[**8.**](#_heading=h.3rdcrjn) **DESCRIPCIÓN DE ARQUITECTURA** 6

[8.1 VISTAS: 6](#_heading=h.26in1rg)

[8.2 TIPOS DE VISTAS: 6](#_heading=h.lnxbz9)

[8.3 Framework Conceptual 7](#_heading=h.35nkun2)

[8.3.1 Descripción de la arquitectura en contexto 7](#_heading=h.1ksv4uv)

[8.3.1 Actores Usuarios y sus roles: 8](#_heading=h.44sinio)

[8.3.2 Actividades de arquitectura en el ciclo de vida 9](#_heading=h.2jxsxqh)

[8.3.3 Descripciones prácticas de arquitectura. 9](#_heading=h.z337ya)

[8.4 Documentación de la arquitectura 10](#_heading=h.3j2qqm3)

[8.5 Selección de puntos de vista de la arquitectura 10](#_heading=h.1y810tw)

[8.6 Vistas de la arquitectura 11](#_heading=h.4i7ojhp)

[UML-VISTA DE ESCENARIO 11](#_heading=h.2xcytpi)

[UML-VISTA LÓGICA-DIAGRAMA DE CLASES 12](#_heading=h.1ci93xb)

[UML-VISTA LÓGICA-DIAGRAMA DE BASE DE DATOS 14](#_heading=h.qsh70q)

[UML-VISTA DE PROCESO-DIAGRAMAS DE ACTIVIDAD 16](#_heading=h.1pxezwc)

[UML-VISTA DE DESPLIEGUE 18](#_heading=h.49x2ik5)

[UML-VISTA FÍSICA – DIAGRAMA INFRAESTRUCTURA 19](#_heading=h.2p2csry)

[Usabilidad y Calidad de software- Front-end -Vistas Principales 21](#_heading=h.147n2zr)

[Usabilidad y Calidad de software – Vistas específicas de usuario 22](#_heading=h.3o7alnk)

[8.7 Consistencia en la cantidad de vistas de la arquitectura. 23](#_heading=h.23ckvvd)

[8.7.1 DESCRIPCION DE MODULOS 23](#_heading=h.ihv636)

[8.7.2 DESCRIPCIÓN DE COMPONENTES 23](#_heading=h.32hioqz)

[8.7.3 DESCRIPCIÓN DE CONECTORES 23](#_heading=h.1hmsyys)

[9.1 Acceso a fuentes de información del proyecto 25](#_heading=h.41mghml)

1. **Introducción**

El presente informe tiene como objetivo detallar la arquitectura de un sistema diseñado para automatizar la gestión de adopciones de mascotas a través de una plataforma web. Este sistema busca facilitar el proceso de adopción, conectando a potenciales adoptantes con animales necesitados de un hogar.

La plataforma propuesta se implementará en un entorno accesible y amigable, garantizando que todas las gestiones se realicen de manera eficiente y organizada, cumpliendo con los estándares de usabilidad y accesibilidad. Además, se ofrecerá una interfaz intuitiva que permita a los usuarios navegar fácilmente por las opciones de adopción y realizar consultas sobre el proceso.

1. **Propósito**

Este documento tiene como objetivo proporcionar una guía para el diseño e implementación de la arquitectura del sistema de adopción de mascotas en línea. Se analizan las decisiones técnicas, los componentes clave y los puntos de integración necesarios para desarrollar una solución robusta y escalable que gestione diversas comunidades de adopción de manera unificada.

La información presentada es crucial para desarrolladores y stakeholders del proyecto, ya que ofrece una visión clara de la estructura y funcionamiento del sistema, así como de las interrelaciones entre sus distintas funcionalidades y componentes. Además, se establecen fundamentos para futuras expansiones y mejoras, considerando aspectos de disponibilidad, seguridad y usabilidad, con el fin de asegurar una experiencia positiva tanto para los adoptantes como para las organizaciones de rescate.

1. **Alcance**

El sistema gestionará las siguientes áreas:

* A) Adopción de mascotas
* B) Búsqueda de mascotas para adopcion
* C) Creación de nuevos perfiles de mascotas
* D) Gestión de perfiles de las mascotas
* E) Gestión de documentos para la adopción de mascotas

1. **Equipo de Trabajo-Actores del Desarrollo**

El desarrollo del sistema requiere la participación de un equipo multidisciplinario con roles claramente definidos para asegurar que el proyecto se lleve a cabo de manera eficiente. A continuación, se presenta una tabla con los principales roles y sus responsabilidades:

| Rol | Responsabilidades |
| --- | --- |
| Cliente-Stakeholder | Fundación Huella Animal |
| Product Owner | Christian Lazcano |
| Desarrolladores Backend | Isaac Monsalve |
| Desarrolladores Frontend | Tomas Astudillo |
| DBA-Especialista en Base de Datos | Benjamín López |
| Tester/QA | Tomas Astudillo / Isaac Monsalve |
| Analista de Requisitos |  |
| Administrador Plataformas |  |

1. **Recomendaciones de conformidad con esta práctica.**

Para asegurar que la arquitectura del sistema cumpla con los requisitos y estándares establecidos, se recomiendan las siguientes prácticas:

1. Implementar revisiones de arquitectura cada 5 días para asegurar que se ajusten a los cambios de requisitos.
2. Utilizar patrones de diseño desacoplados como MVC para modularidad y escalabilidad o por capas.
3. Adoptar metodologías ágiles como Scrum para la entrega continua de mejoras y funcionalidades.
4. **Referencias y estándares aplicables a este documento:**

El documento sigue los siguientes estándares de calidad para asegurar la robustez y escalabilidad del sistema:

* **IEEE 830-1998 ST**: Estandarización de requisitos de software.
* **ISO 9126-2001**: Métricas de calidad del software.
* **Modelo 4+1 de Kruchten**: Modelo estructural basado en múltiples vistas de escenarios.
* **ATAM (Architecture Tradeoff Analysis Method)**: Metodología para evaluar y documentar decisiones arquitectónicas

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

| Término | Definición |
| --- | --- |
| DAS | Documento de Arquitectura de Software |
| MVC | Modelo-Vista-Controlador, patrón de diseño utilizado para separar la lógica de negocio de la interfaz de usuario |
| UML | Unified Modeling Language, lenguaje gráfico para representar la arquitectura del sistema |
| CRUD | Operaciones básicas de Create, Read, Update, Delete para la gestión de datos. |

1. **DESCRIPCIÓN DE ARQUITECTURA:**La arquitectura del sistema se describe a través de vistas basadas en el Modelo 4+1 de Kruchten, con el objetivo de ofrecer una perspectiva completa y estructurada desde diferentes ángulos. Este enfoque permite analizar el sistema de manera integral, asegurando que todas las dimensiones relevantes sean consideradas en su diseño y funcionamiento.
   1. VISTAS:

Cada vista se representará en un diagrama UML para describir la estructura y comportamiento del sistema.

* 1. TIPOS DE VISTAS:

1. **Vista de Escenarios (Diagrama de Caso de Uso)**: Describe las funcionalidades del sistema y cómo interactúan los diferentes actores con él.
2. **Vista Lógica (Diagrama de Clases)**: Detalla la estructura de datos y las relaciones entre las principales entidades del sistema.
3. **Vista de Procesos (Diagrama de Actividad)**: Muestra el flujo de actividades dentro del sistema y cómo se coordinan los diferentes procesos.
4. **Vista de Despliegue (Diagrama de Componentes)**: Representa la distribución de componentes en la infraestructura física.
5. **Vista Física (Diagrama Topológico)**: Representa la infraestructura de red y la comunicación entre los diferentes servicios del sistema
   1. Framework Conceptual
      1. Descripción de la arquitectura en contexto

El sistema a desarrollar estará realizado en un metodología ágil para ser más claros esta será enfocada en Scrum, la cual permitirá una entrega de forma incremental de la funcionalidad del proyecto, siendo esta una metodología la cual permite una adaptación a cambios en los requerimientos ya establecidos durante la primera instancia

**El diseño de nuestro sitio de adopción de mascotas se basará en la separación de vistas, que ofrecerán diferentes perspectivas para una experiencia completa y enriquecedora. A continuación, se describen las principales vistas:**

* **Vista de Mascotas:** Representada mediante perfiles detallados de cada animal, que describen sus características, necesidades y la historia detrás de su adopción.
* **Vista de Adopción:** Incluye un formulario sencillo y accesible que guía a los usuarios a través del proceso de adopción, mostrando los requisitos y pasos necesarios para brindar un hogar a una mascota.
* **Vista de Recursos:** Utiliza secciones informativas para ilustrar las interacciones entre los distintos módulos, como consejos sobre cuidados, entrenamiento y recursos para nuevos adoptantes.
* **Vista de Comunidad:** Se implementará a través de un espacio dedicado a historias de adopción y testimonios de adoptantes, reflejando la conexión entre las mascotas y sus nuevas familias.
* **Vista de Contacto:** Utiliza formularios y enlaces directos para facilitar la comunicación entre adoptantes, voluntarios y organizaciones, asegurando que todos tengan acceso a la información que necesitan.

Cada una de estas vistas está interconectada para garantizar la coherencia y la alineación con los requisitos de negocio. No hay ninguna vista separada de una implementación específica, sino que todas contribuyen a una visión holística de la arquitectura del sistema descrita en este documento.

Estas vistas se han creado utilizando el Lenguaje de Modelado Unificado (UML) en su versión 2.0 y han sido desarrolladas empleando herramientas de modelado estándar como **StarUML** y Lucidchart y en front-end **Canva y Trello** para facilitar la creación de diagramas visuales que respalden la documentación arquitectónica.

El estilo arquitectónico se centra en una combinación de patrones de diseño orientados a servicios xxxx (ejemplo: SOA) y arquitectura por capas, en donde cada módulo del sistema se corresponde con un conjunto específico de casos de uso que resuelven los procesos definidos para la operación.

Estos procesos corresponden a:

Ejemplo:

* **Gestión de Adopciones en Línea:** Administración de las solicitudes de adopción a través de un sistema intuitivo que permite a los usuarios completar formularios, subir documentos y recibir actualizaciones sobre el estado de su solicitud de manera fácil y rápida.
* **Gestión de Donaciones en Línea:** Registro y seguimiento de donaciones a través de nuestra plataforma, donde los usuarios pueden contribuir con fondos o recursos.
* **Monitoreo de Actividades del Sitio:** Seguimiento de la interacción de los usuarios en la plataforma, garantizando una experiencia segura y amigable. Incluye herramientas para gestionar accesos y asegurar que la información de los adoptantes y donantes esté protegida.
  + 1. Actores Usuarios y sus roles:

Este documento representa la identificación de Actores/Usuarios Stakeholders y sus roles a partir de la interpretación de los casos de uso del Negocio asociados.

| ACTOR | ROL / CARGO | FUNCION |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

8.3.2 Actividades de arquitectura en el ciclo de vida

Para construir un sitio web adaptable y centrado en la experiencia del usuario, implementamos un **ciclo de vida iterativo e incremental** basado en la metodología **Scrum**. Este enfoque nos permite lanzar mejoras de forma continua, validando funcionalidades importantes con usuarios y colaboradores desde las primeras etapas. Así, el sitio se ajusta rápidamente a las necesidades de los adoptantes y a los cambios en el entorno de adopción.

**Proceso de Desarrollo por Etapas**

Organizamos el desarrollo en **sprints de 2 a 3 semanas**, cada uno dedicado a un módulo específico (como el perfil de mascotas o el proceso de solicitud de adopción). Al concluir cada sprint, evaluamos el progreso, alineamos los resultados con los objetivos de los usuarios y priorizamos nuevas tareas para mejorar continuamente la experiencia en el sitio.

Gracias a este proceso, nuestro sitio web está en constante evolución para facilitar la adopción y conectar a más mascotas con sus futuros hogares.

8.3.3 Descripciones prácticas de arquitectura.

Durante el proceso de desarrollo, la arquitectura del sistema fue escalada desde un **escenario principal de caso de uso** hasta un diseño modular a través de los siguientes pasos:

1. **Análisis de Requerimientos**: Se analizaron los requisitos iniciales del sistema para identificar los casos de uso principales, tales como "Proceso de adopción", "Planificación de visitas" y "Proceso de donaciones".
2. **Definición de la Vista Lógica**: A partir del escenario principal, se identificaron las entidades y relaciones claves en un diagrama de clases UML, representando el núcleo del sistema.
3. **Modelo de Datos**: Se desarrolló un diagrama de base de datos para definir la estructura de la información, manteniendo la coherencia con la visión lógica.
4. **Vista de Procesos**: Se modelaron los flujos de las actividades y transacciones principales del sistema mediante diagramas de actividad.
5. **Despliegue**: Se definió la infraestructura física y lógica para alojar el sistema en la nube, detallando los servicios necesarios, como balanceadores de carga, servidores de aplicaciones y bases de datos distribuidas.
6. **Definición de Planes de pruebas y técnicas de pruebas testing**.
   1. Documentación de la arquitectura

* **ISO/IEC 25010**: Se utilizó para evaluar la calidad del sistema en términos de funcionalidad, seguridad y eficiencia.
* **IEEE 1471-2000**: Proporcionó la guía para la representación de vistas arquitectónicas y la identificación de stakeholders.
* **Modelo 4+1 de Kruchten**: Estructura la arquitectura en vistas lógicas, de desarrollo, de procesos y física, facilitando la visualización desde múltiples perspectivas.
* **Planes de prueba**: (Declarar los tipos de pruebas a realizar)
  1. Selección de puntos de vista de la arquitectura

| VISTAS | UML | Cantidad |
| --- | --- | --- |
| Escenario Principal | Diagrama de Caso de uso | 1 |
| Vista Lógica | Diagrama de Clases  MER | 1 |
| Vista de Proceso | Diagrama de Actividad:  Proceso adopción.  Agendar visita. | 2 |
| Vista Física | Diagrama Topológico de Servicios | 1 |
| Vista de Despliegue | Diagrama de componentes  Diagrama de Capas (MVC) | 1  1 |

* 1. Vistas de la arquitectura

Ejemplos:

| UML-VISTA DE ESCENARIO |
| --- |
| Diagrama de Caso de uso extendido /Subproyecto [Inserte nombre del Subproyecto asignado] |
| z |

**Página web del refugio**: Permite al Súper Administrador crear, modificar y eliminar mascotas que se encuentren en el refugio.

**Agregar Mascota**: Permite al administrador el incorporar nuevas mascotas al sitio web.

**Banear Usuario**: Esta opción permite al administrador bloquear el acceso a usuarios malintencionados.

**Registro de Usuario**: Esto permite al futuro adoptante crear un usuario el cual le permite entrar al proceso de adopción y visitas.

**Login**: El sistema permite a los usuario (ayudante, adoptante y administrador) iniciar sesión en la página web con su correspondientes beneficios .

**Donaciones**: Los residentes pueden realizar pagos a través del sistema utilizando diversas formas de pago.

**Ver mascota**: El sistema permite a todo usuario sin necesidad de iniciar sesión poder ver un catálogo con los animales del refugio.

**Adoptar**:Para este punto el sistema pedirá estar con sesión iniciada.

| UML-VISTA LÓGICA-DIAGRAMA DE CLASES |
| --- |
| [Diagrama de Clases] |
|  |

| UML-VISTA LÓGICA-DIAGRAMA DE BASE DE DATOS |
| --- |
| [Diagrama de BD - MER] |
|  |

El **Modelo Entidad-Relación (MER)** para el sistema de adopción del sitio web representa las entidades involucradas, sus atributos y las relaciones que existen entre ellas. Este modelo permite visualizar cómo se organiza la información relacionada con el sitio web de adopción para mascotas.

**2. Elementos Principales**

* **Entidades**:
  + **Usuario**:
    - **Atributos**: id\_user(PK), Nombre, Apellido, rut, Email, password, fono.
  + **Donaciones**:
    - **Atributos**: id\_donacion(PK), tipo\_donacion, fecha\_donacion, message, id\_user.
  + **DetallesDonacion**:
    - **Atributos**: id\_detalleDonacion(PK), monto, Detalle, tipoTarjeta, id\_donacion.
  + **Fecha\_visitas**:
    - **Atributos**: id\_visita(PK), fecha\_visita, hora\_visita, id\_user, id\_pets.
  + **Documentos**:
    - **Atributos**: id\_documento(PK), documento, id\_user, id\_pets, estado.
  + **Adoptante**:
    - **Atributos**: id\_adoptante(PK), fecha\_adopcion, id\_user, id\_pets, id\_documento.
  + **Pets**:
    - **Atributos**: id\_pets(PK), nombre, id\_especie, id\_raza, edad, sexo, descripcion, fecha\_llegada, disponible, id\_refugio, img, comentario.
  + **Raza:**
    - **Atributos:** id\_raza(PK), raza.
  + **Reuniones:**
    - **Atributos:** id\_reunion(PK), fecha\_reunion, message, link\_reunion, id\_adoptante
  + **Comuna:**
    - **Atributos:** id\_comuna(PK), comuna
  + **Refugio:**
    - **Atributos:** id\_refugio(PK), nombre, dirección, teléfono, id\_ comuna.
  + **Especie**:
    - **Atributos**: id\_especie(PK), especie.
* **Relaciones**:
  + **Usuario** puede estar asociado a **múltiples mascotas**.
  + **Donaciones** tiene **múltiples donaciones y múltiples donantes**
  + **Detalle de donaciones** tiene solo un donante y una sola donación
  + **Fecha visitas** está asociado a una **mascota** y es administrado por un **administrador**.
  + **Documento** se asocia a un **solo usuario y mascota**.
  + **Adoptante** se asocia con múltiples mascotas.
  + **Pets** está asociada a múltiples adoptantes y solo se maneja por un administrador
  + **Raza** estas se asocia a una sola mascota
  + **Reuniones** estas se asocian a un solo adoptante, un solo administrador y una sola mascota
  + **Comuna** esta se asocia una por refugio
  + **Refugio** esta se asocia una por administrador, una por mascota y una por reunión(visita)
  + **Especie** esta se relaciona a múltiples mascotas

| UML-VISTA DE PROCESO-DIAGRAMAS DE ACTIVIDAD |
| --- |
| [Diagramas de Actividad] |
|  |

El **diagrama de actividad** representa el flujo de actividades que se llevan a cabo en un proceso específico, en este caso, el proceso de adopción de una mascota. Este diagrama es útil para visualizar los pasos secuenciales y los procesos requeridos para poder agendar una visita con la mascota

**Componentes**:

1. **Inicio**: El usuario ingresa al sitio web para adoptar una mascota.
2. **Adopcion de mascota**: El sistema le pregunta si tiene cuenta registrada en el sitio
3. **Inicio de sesión**: Se inicia sesión en el sitio con usuario previamente creado.
4. **Se selecciona una mascota de interés**:
   * Si la mascota está disponible para adopción:
     + El sistema mostrará datos de interés de la mascota y se pedirán documentos requeridos.
   * Si la mascota no está disponible para adopción:
     + El sistema dará un aviso que la mascota no está disponible para adopción.
5. **Carga de documentación** :
   * El sistema te mencionara que debes cumplir con cierta documentación a tu disposición cual deberá ser cargada al sitio web.
6. **Verificación de documentación**: Una vez la documentación solicitada sea revisada y aceptada, se le avisará al futuro adoptante que fue seleccionado entre un grupo de personas como el nuevo propietario de la mascota .
7. **Fin**: El proceso web finalizará al momento de que reciba la notificación antes mencionada(La firma de la adopción se realizará de forma presencial )

| UML-VISTA DE DESPLIEGUE |
| --- |
| Sistema de Adopciones |
|  |

**Servidor de Aplicaciones**:

* **Sitio web adopción de mascotas**: Proporciona una interfaz amigable para todos los actores del sistema (Adoptante, administradores y mascotas). Permite la adopción web, las donaciones y la reversa de visitas para conocer las mascotas

**Servidor de Base de Datos**:

* **Base de Datos del sitio web**: Almacena toda la información relevante, incluyendo detalles de los adoptantes, donadores, datos de las mascotas y genera historiales de futuras modificaciones

**Sistema de registro**:

* **Sistema de registro**: Se encarga de enviar correos electrónicos automáticos a los nuevos usuarios que sean registrados y general un canal/puente de comunicaciones entre el refugio y el futuro adoptante

| Usabilidad y Calidad de software- Front-end -Vistas Principales |
| --- |
| Vistas principales del sistema – Vista Home Principal de entrada |
|  |

La **Vista Front-End** de nuestro sitio de adopción de mascotas ha sido diseñada para ofrecer una experiencia de usuario intuitiva, moderna y fácil de navegar. La interfaz se organiza en secciones claras y con diferentes opciones como visita para conocer las mascotas y donaciones al refugio

Como presentación del sitio web buscamos marcar la alegría de lo que sería tener una mascota junto a cada familia, siendo esto enfocado en la utilización de colores alegres y claros con un “Mapeo de diferente mascotas”

Nuestras vistas estarán disponibles en diferentes tipos de dispositivos móviles los cuales tengan un navegador web y una conexión a internet.