**Documento de Diseño Arquitectónico (ADD)**

**Proyecto:** Aplicación Móvil "My Vet"

**Versión:** 1.0

**Fecha de Emisión:** 7 de septiembre de 2025

**Autores:** Enzo González, Nicolás Pardo y María Ester Zavala.

**Información del Servicio:**

* **Resumen:** Este documento describe el diseño arquitectónico de alto nivel para la aplicación MyVet. Define la descomposición del software en sus componentes principales, las interfaces entre ellos y las decisiones de diseño clave tomadas para satisfacer los requisitos especificados en el Documento de Requisitos de Software (SRD). El ADD sirve como el plano fundamental para la fase de Diseño Detallado y Producción (DD).
* **Tabla de Contenidos:**
  1. Introducción
  2. Visión General del Sistema
  3. Contexto del Sistema
  4. Diseño del Sistema
  5. Descripción de Componentes
  6. Estimaciones de Viabilidad y Recursos
  7. Matriz de Trazabilidad de Requisitos

Claro, aquí tienes el Documento de Diseño Arquitectónico (ADD) para el proyecto MyVet. Este documento se ha elaborado siguiendo la estructura del estándar ESA PSS-05-0, detallada en la guía PSS-05-04, y se basa en la información de todos los documentos del proyecto proporcionados anteriormente.

**Documento de Diseño Arquitectónico (ADD)**

**Proyecto:** Aplicación Móvil "My Vet" **Versión:** 1.0 **Fecha de Emisión:** 7 de septiembre de 2025 **Autores:** Enzo González, Nicolás Pardo y María Ester Zavala.

**Información del Servicio:**

* **Resumen:** Este documento describe el diseño arquitectónico de alto nivel para la aplicación MyVet. Define la descomposición del software en sus componentes principales, las interfaces entre ellos y las decisiones de diseño clave tomadas para satisfacer los requisitos especificados en el Documento de Requisitos de Software (SRD). El ADD sirve como el plano fundamental para la fase de Diseño Detallado y Producción (DD).
* **Tabla de Contenidos:**
  1. Introducción
  2. Visión General del Sistema
  3. Contexto del Sistema
  4. Diseño del Sistema
  5. Descripción de Componentes
  6. Estimaciones de Viabilidad y Recursos
  7. Matriz de Trazabilidad de Requisitos
* **Hoja de Estado del Documento:**

| Versión | Fecha | Autor | Descripción del Cambio |
| --- | --- | --- | --- |
| 1.0 | 07-09-2025 | Asistente IA | Creación del documento inicial a partir del SRD y las guías PSS-05-04. |

Exportar a Hojas de cálculo

**1. Introducción**

**1.1 Propósito**

El propósito de este ADD es traducir los requisitos del software del proyecto MyVet en una arquitectura coherente y estructurada. Proporciona una descripción de la solución técnica, sirviendo como guía para el equipo de desarrollo durante la implementación y como base para la planificación de las pruebas de integración.

**1.2 Alcance**

Este documento abarca el diseño arquitectónico de la aplicación móvil MyVet para iOS y Android, incluyendo sus componentes de frontend, los servicios de backend y las interfaces con sistemas externos. No se entra en el detalle de la lógica interna de cada módulo, ya que eso corresponde a la fase de Diseño Detallado (DD).

**1.3 Definiciones, Acrónimos y Abreviaciones**

* **ADD:** Architectural Design Document (Documento de Diseño Arquitectónico).
* **API:** Application Programming Interface (Interfaz de Programación de Aplicaciones).
* **Backend:** Componentes del sistema que se ejecutan en el servidor y no son directamente accesibles por el usuario.
* **Frontend:** La aplicación móvil con la que el usuario interactúa directamente.
* **REST:** Representational State Transfer (un estilo de arquitectura para la comunicación entre sistemas).
* **SRD:** Software Requirements Document (Documento de Requisitos de Software).

**1.4 Referencias**

**Documentos Aplicables:**

* ESA PSS-05-04: Guide to the software architectural design phase .
* SRD - MyVet V1.4 .

**Documentos de Referencia:**

* URD - MyVet V1.0.
* PAPS - MyVet V1.0 .
* SCMP/DD - MyVet V1.0 .

**1.5 Vista General**

Este documento comienza con una visión general de la arquitectura. A continuación, define el contexto del sistema y sus interfaces externas. La sección de Diseño del Sistema explica el método de diseño y la descomposición en componentes. Posteriormente, cada componente se describe en detalle. Finalmente, se presentan las estimaciones de recursos y la matriz de trazabilidad que vincula esta arquitectura con los requisitos del SRD.

**2. Visión General del Sistema**

La arquitectura de MyVet se concibe como un

**sistema cliente-servidor de tres capas**, diseñado para ser modular, escalable y mantenible.

1. **Capa de Presentación (Cliente):** Una aplicación móvil multiplataforma (frontend) que proporciona la interfaz de usuario para los clientes y los profesionales.
2. **Capa de Lógica de Negocio (Servidor):** Un conjunto de servicios de backend responsables de gestionar los datos, la autenticación, la lógica de la aplicación y la comunicación con el motor de IA.
3. **Capa de Datos:** Una base de datos NoSQL en la nube para persistir toda la información de la aplicación.

Este enfoque permite separar las responsabilidades, facilitando el desarrollo paralelo y la futura evolución de cada capa de forma independiente.

**3. Contexto del Sistema**

El sistema MyVet operará interactuando con los usuarios a través de sus dispositivos móviles y con varios servicios externos en la nube.

* **Usuarios (Cliente y Profesional):** Interactúan con el sistema a través de la aplicación móvil MyVet.
* **Servicios de Autenticación (Firebase/AWS):** Gestionan el registro e inicio de sesión de los usuarios, proporcionando un método seguro y estándar.
* **Base de Datos (MongoDB Atlas):** Almacena todos los datos persistentes de la aplicación, como perfiles de usuario, datos de mascotas, citas e historiales clínicos.
* **Motor de IA (TensorFlow Lite o similar):** Un modelo de Machine Learning, posiblemente ejecutado en el dispositivo o en la nube, que procesa los datos de síntomas para generar diagnósticos preliminares.

**4. Diseño del Sistema**

**4.1 Método de Diseño**

El diseño se basa en una combinación de

**Arquitectura por Capas** y un enfoque **Orientado a Servicios** para el backend. Se ha adoptado un método de diseño

**top-down**, comenzando por la identificación de los componentes principales del sistema y descomponiéndolos progresivamente en subcomponentes más específicos y manejables. El diseño busca maximizar la

**cohesión** dentro de cada componente y minimizar el **acoplamiento** entre ellos para mejorar la mantenibilidad y la modularidad.

**4.2 Descripción de la Descomposición**

El sistema MyVet se descompone jerárquicamente en los siguientes componentes principales:

1. **C1 - Aplicación Móvil (Frontend)**
   * C1.1 - Módulo de Autenticación de Usuario
   * C1.2 - Módulo de Interfaz Cliente
   * C1.3 - Módulo de Interfaz Profesional
2. **C2 - Servicios de Backend (API)**
   * C2.1 - Servicio de Gestión de Usuarios
   * C2.2 - Servicio de Gestión de Mascotas e Historiales
   * C2.3 - Servicio de Gestión de Citas
   * C2.4 - Servicio de Integración con IA
3. **C3 - Base de Datos**
4. **C4 - Motor de IA**

**5. Descripción de Componentes**

**5.1 C1 - Aplicación Móvil (Frontend)**

* **5.1.1 Tipo:** Aplicación ejecutable para iOS y Android, desarrollada con un framework multiplataforma (Flutter o React Native).
* **5.1.2 Propósito:** Proporcionar la interfaz gráfica para que los usuarios interactúen con las funcionalidades del sistema. Traza a los requisitos RF-01, RF-02, RF-03.
* **5.1.3 Función:** Renderizar vistas, capturar la entrada del usuario, comunicarse con el backend a través de una API REST y gestionar el estado local de la aplicación.
* **5.1.4 Subordinados:** C1.1, C1.2, C1.3.
* **5.1.5 Dependencias:** Depende de los Servicios de Backend (C2) para toda la lógica de negocio y persistencia de datos.
* **5.1.6 Interfaces:** Se comunica con C2 a través de una API REST sobre HTTPS.
* **5.1.7 Recursos:** Requiere un dispositivo móvil con sistema operativo iOS o Android y conexión a internet.
* **5.1.8 Referencias:** SRD , URD.
* **5.1.9 Procesamiento:** Gestiona la navegación entre pantallas, valida formularios y maneja las respuestas de la API.
* **5.1.10 Datos:** Maneja datos transitorios de la interfaz de usuario.

**5.2 C2 - Servicios de Backend (API)**

* **5.2.1 Tipo:** Conjunto de servicios sin estado (stateless) ejecutados en una plataforma en la nube (Firebase/AWS).
* **5.2.2 Propósito:** Centralizar la lógica de negocio de MyVet, asegurando la integridad y seguridad de los datos. Traza a todos los requisitos funcionales.
* **5.2.3 Función:** Exponer endpoints de API para el frontend, procesar solicitudes, interactuar con la base de datos y coordinar con otros servicios (Autenticación, IA).
* **5.2.4 Subordinados:** C2.1, C2.2, C2.3, C2.4.
* **5.2.5 Dependencias:** Depende de C3 - Base de Datos para la persistencia, de los servicios de autenticación externos y de C4 - Motor de IA.
* **5.2.6 Interfaces:** Expone una API REST para ser consumida por C1. Se comunica con C3 mediante el driver de la base de datos y con C4 a través de una interfaz de servicio específica.
* **5.2.7 Recursos:** Requiere un entorno de servidor en la nube (CPU, RAM, almacenamiento).
* **5.2.8 Referencias:** SRD .
* **5.2.9 Procesamiento:** Cada servicio subordinado se especializa en un dominio:
  + **C2.1 - Gestión de Usuarios:** Maneja la creación y actualización de perfiles de usuario.
  + **C2.2 - Gestión de Mascotas e Historiales:** Lógica para CRUD de mascotas e historiales clínicos.
  + **C2.3 - Gestión de Citas:** Lógica para agendamiento, cancelación y recordatorios.
  + **C2.4 - Integración con IA:** Recibe los datos de síntomas, los envía a C4 y devuelve el resultado.
* **5.2.10 Datos:** Define los modelos de datos de la aplicación (Usuario, Mascota, Cita, Historial).

**6. Estimaciones de Viabilidad y Recursos**

* **Viabilidad Técnica:** La arquitectura propuesta se basa en tecnologías maduras y ampliamente utilizadas en la industria (frameworks móviles multiplataforma, servicios en la nube, bases de datos NoSQL) . Se considera técnicamente viable para el equipo de desarrollo actual. El principal riesgo técnico reside en la obtención de datos y el entrenamiento de un modelo de IA con la precisión requerida.
* **Recursos Requeridos (AD15):**
  + **Personal:** 3 personas (Directora de Proyecto, Desarrollador Front-End, Desarrollador Back-End) durante 6 meses.
  + **Hardware:** El desarrollo se realizará en equipos personales. Para la operación se requerirán servicios en la nube (hosting, base de datos).
  + **Software:** Se utilizará Android Studio, Xcode, Git, Figma, y un lenguaje de programación compatible con el framework móvil elegido (Dart para Flutter, JavaScript/TypeScript para React Native).
  + **Presupuesto Estimado:** $345.000 CLP para costos iniciales de dominio, hosting y publicidad.

**7. Matriz de Trazabilidad de Requisitos (SRD a ADD)**

Esta tabla asegura que todos los requisitos del software están cubiertos por la arquitectura diseñada (AD21).

|  |  |  |
| --- | --- | --- |
| ID Requisito (SRD) | Descripción Breve | Componente(s) ADD que lo implementan |
| RF-01.01 | Registro e inicio de sesión para dos tipos de usuario | C1.1, C2.1, Servicio de Autenticación Externo |
| RF-01.02 | Interfaz de usuario diferente por tipo de cuenta | C1.2, C1.3 |
| RF-02.01 | Cliente registra mascotas | C1.2, C2.2, C3 |
| RF-02.02 | Cliente agenda, cancela y reprograma citas | C1.2, C2.3, C3 |
| RF-02.03 | Envío de recordatorios automáticos a clientes | C2.3 |
| RF-02.04 | Cliente llena formulario de síntomas | C1.2, C2.4, C4 |
| RF-03.01 | Profesional accede a agenda de citas | C1.3, C2.3, C3 |
| RF-03.02 | Profesional visualiza historial clínico | C1.3, C2.2, C3 |
| RF-03.03 | Profesional accede a diagnóstico preliminar de IA | C1.3, C2.4, C4 |
| RF-03.04 | Profesional registra nuevos animales e historiales | C1.3, C2.2, C3 |
| RF-04.01 | Modelo de IA analiza síntomas | C4 |
| RNF-01.03 | Cumplimiento Ley 19.628 | C2 (en toda su lógica), C3 (en su configuración) |
| RNF-03.01 | Tiempo de respuesta de IA < 10 segundos | C2.4, C4 |
| RNF-04.01 | Precisión de IA > 70% | C4 |
| RNF-05.01 | Compatibilidad con iOS y Android | C1 |