

| **1. Resumen avance Proyecto APT** |
| --- |
| A continuación, encontrarás distintos campos que deberás completar con la información solicitada. |

| Resumen de avance proyecto APT | Durante el primer sprint del Proyecto APT se avanzó hasta el requerimiento 11, logrando implementar y validar las funcionalidades esenciales para la gestión de usuarios (Dueño) y mascotas:  **Funcionalidades desarrolladas:**   1. Registro, autenticación y recuperación de contraseña de usuarios. 2. Creación, edición y visualización del perfil de usuario. 3. Registro, edición y generación de perfiles de mascotas. 4. Asociación de fotografías a perfiles de mascotas. 5. Listado de mascotas por usuario y visualización de sus datos completos.   **Objetivos alcanzados:**   * Se consolidó la base del sistema, integrando **frontend y backend** mediante React y Supabase. * Las imágenes de mascotas se suben correctamente al **bucket pets** organizado por user\_id. * La interfaz permite **editar perfiles y actualizar información de forma inmediata**, incluyendo fotos, asegurando una experiencia de usuario fluida.   **Ajustes realizados:**   * Se priorizó que la edición de imágenes se refleje automáticamente en la base de datos y en la interfaz. * Se decidió organizar las imágenes de mascotas por user\_id dentro del bucket de Supabase para una gestión más ordenada. |
| --- | --- |
| Objetivos | **Objetivo General**  Desarrollar una aplicación web que permita a los usuarios gestionar y monitorear el cumplimiento de las obligaciones asociadas a la tenencia responsable de sus mascotas en un período de 3 meses y 1 semana.  **Objetivos Específicos**   1. Lograr que al menos el 50% de los usuarios registren la información personal, clínica y de hábitos de sus mascotas en los primeros 6 meses de uso de la plataforma. 2. Conseguir que el 80% de los usuarios activos programe y cumpla rutinas con alertas inteligentes (alimentación, paseos, vacunas, medicación) en un periodo de 6 meses, desde la implementación. 3. Asegurar que el 70% de los usuarios utilicen el repositorio digital para subir y consultar documentos clínicos y legales en los primeros 6 meses de uso, desde la implementación. 4. Garantizar que el 75% de los usuarios consulte al menos un reporte mensual sobre indicadores de bienestar (peso, actividad, gastos, rutinas), con generación automática de resúmenes cada 30 días. 5. Lograr que al menos el 60% de los usuarios utilice la función de comparativas de gastos durante los primeros 6 meses. 6. Entregar al menos 3 tipos de informes anuales mediante inteligencia de negocios, dirigidos a actores externos (municipios, ONGs, aseguradoras o comercios). |
| Metodología | Se mantiene la metodología híbrida Agile-Waterfall. |
| Evidencias de avance | El desarrollo del proyecto para el primer sprint llegaría hasta el **requerimiento 11 en donde los requerimiento del proyecto en total son 46** , se presentan las siguientes evidencias:   1. **Planilla de Requerimientos**     * Contiene la lista completa de los requerimientos funcionales definidos donde se realizaron para esta fase 11 requerimientos para este avance .    * Permite verificar que se identificaron y priorizaron correctamente las funcionalidades clave del sistema, como registro de usuario, gestión de perfiles de mascotas y asociación de fotografías.    * Justificación: Esta evidencia demuestra que el proyecto se está desarrollando conforme a las necesidades del usuario y que los objetivos específicos del primer sprint fueron cumplidos. 2. **Planilla de Especificación de Requerimientos**     * Documenta en detalle cada requerimiento: descripción, prioridad, criterios de aceptación y responsables.    * Justificación: Permite evidenciar la correcta planificación y seguimiento de los requerimientos, asegurando que el desarrollo del software cumple con las expectativas y estándares de calidad definidos. |

| **2. Monitoreo del Plan de Trabajo** |
| --- |
| Examina cuidadosamente tu plan de trabajo, enfocándote especialmente en la columna de estado de avance y ajustes. |

| Plan de Trabajo | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Competencia o unidades de competencias | Actividades | Recursos | Duración de la actividad | Responsable[[1]](#footnote-0) | Observaciones | Estado de avance | Ajustes |
| Levantamiento de requerimientos | Análisis de la problemática y actores | Reuniones, docs, PC con internet | 1 semana | Equipo completo | Dificultad en priorización inicial | Completado | — |
| Diseño de sistemas | Modelado BD y diagramas | Draw.io/Figma, computadora | 1 semana | Alessandra | Campos faltantes o sobrantes | Completado | Ajustes tras feedback |
| Desarrollo de software | Autenticación y perfiles | VS Code, Supabase | 2 semanas | Equipo completo | Solucionado 401/403 y permisos de schema | En curso | — |
| Desarrollo de software | CRUD de mascotas + Storage | Entorno React+Supabase | 1–2 semanas | Equipo completo | Pasar por alto validar algún campo. | Completado | Cambio a soft delete para trazabilidad |

| **3. Ajustes a partir del monitoreo** |
| --- |
| Profundiza en las observaciones de tu plan de trabajo. Analiza las actividades planificadas y señala qué aspectos facilitaron u obstaculizaron la ejecución del plan. Plantea cómo abordaste y/o abordarás los obstáculos. Por último, señala los ajustes que realizaste al plan de trabajo a partir de este análisis. |

| Factores que han facilitado y/o dificultado el desarrollo de mi plan de trabajo:  **Factores que han facilitado el desarrollo:**   1. Claridad en los requerimientos iniciales: La definición precisa de los primeros 11 requerimientos permitió organizar el primer sprint con objetivos claros, evitando retrabajos innecesarios. 2. Trabajo colaborativo eficiente: La coordinación del equipo y la asignación de roles específicos agilizaron el desarrollo de funcionalidades clave, como el registro de usuario y la gestión de perfiles de mascotas. 3. Uso de tecnologías con amplia documentación: Frameworks y herramientas como React, Supabase y Vercel poseen una amplia y accesible biblioteca de documentación, ya sea en formato blog o videotutorial, permitiendo una adaptabilidad rápida a los componentes y servicios backend, reduciendo la curva de aprendizaje. 4. Retroalimentación constante: Las reuniones periódicas y la revisión de avances permitieron identificar errores temprano y ajustar detalles de manera oportuna.   **Factores que han dificultado el desarrollo:**   1. Integración de servicios de almacenamiento: Inicialmente hubo complicaciones al asociar imágenes de mascotas al bucket de Supabase y vincularlas correctamente con el user\_id.     * Acción tomada: Se investigó y aplicó la estructura de carpetas por user\_id en el bucket, logrando que la imagen se suba correctamente y se visualice al guardar cambios. 2. Gestión de estados y formularios dinámicos: Manejar la edición de perfiles de usuario y mascotas con datos dependientes de catálogos generó complejidad en el manejo de state en React.     * Acción tomada: Se implementó un formData centralizado y funciones de carga y actualización modularizadas, simplificando el control de cambios. 3. Coordinación de tiempos: La planificación de tareas entre integrantes con diferentes niveles de experiencia generó leves retrasos en la sincronización de avances.     * Acción futura: Establecer puntos de control más frecuentes y checklists diarios para asegurar que todos los integrantes avancen de manera alineada. |
| --- |

| Actividades ajustadas o eliminadas:  **“Eliminar mascota” Soft delete (archivar con deleted\_at)**  Ajustada para evitar pérdida definitiva de datos, facilitar restauración y trazabilidad de historial.  De esta forma se da la posibilidad de consultas más seguras, posibilidad de desarchivar, mejor UX (se añadió confirmación “escribe el nombre”).  **Qué se eliminó:** la acción de *hard delete* en la UI general; queda reservada solo para mantenimiento/administración si se requiere.  **Secuencia de logout Por qué se ajustó:** prevenir “volver atrás” a rutas privadas tras cerrar sesión. De tal forma que la navegación es más segura (primero navigate("/"), luego signOut()). **Herramienta de build y configuración de estilos → Vite + Tailwind (ajuste de instalación)** Se ajustó porque al inicio se intentó instalar Tailwind con los comandos npm install -D tailwindcss postcss autoprefixer y npx tailwindcss init -p, pero se estaba forzando una **versión anterior** que provocó incompatibilidades en la configuración inicial. Se utilizó la documentación oficial **“Tailwind con Vite”** para un proyecto React en Vite, asegurando versiones compatibles (Tailwind, PostCSS y Autoprefixer), configurando tailwind.config.js con las rutas de contenido y declarando @tailwind base; @tailwind components; @tailwind utilities; en el CSS principal. De esta forma, el entorno de desarrollo es más rápido y estable, menor fricción en estilos y builds, y alineamiento con documentación vigente. No cambió el alcance funcional ni los entregables. |
| --- |

| Actividades que no has iniciado o están retrasadas:  **Perfiles adicionales (cuidador, veterinario, etc.) — No iniciado** Se priorizó dejar operativo el flujo base del **dueño** (signup, perfil, mascotas) y resolver bloqueadores técnicos (RLS, PII, validaciones). La definición fina de campos por rol quedó para esta semana.  **Estrategia para avanzar sin afectar el APT:**   * **Reutilización de la base ya creada**: aprovechar app\_user/user\_pii y el flujo de autenticación existente; solo agregar **atributos por rol** (p. ej., credenciales/registro para veterinario, disponibilidad para cuidador) y un **selector de rol** (enum o catálogo de roles). * **Implementación incremental**: partimos con un **catálogo de roles** y UI derivada del dueño (misma página, secciones visibles según rol). * **Seguridad**: extender RLS con condiciones por rol (mismas políticas, agregando checks simples por rol). * **Pruebas rápidas**: crear cuentas de prueba por rol y validar accesos cruzados (negativos).  **Criterios de aceptación:** selección de rol en signup/edición, persistencia de campos específicos por rol, vistas/acciones condicionadas en la UI y pruebas RLS aprobadas con usuarios de cada rol.  **Plazo estimado (corto):** **2–4 días**, dado que la base y los formularios ya existen. |
| --- |

1. En caso de que el Proyecto APT sea grupal, en esta columna deben indicar el nombre de los responsables de cada tarea o actividad. Esto posteriormente permitirá diferenciar la evaluación por cada integrante. [↑](#footnote-ref-0)