Informe de Autoevaluación – Desarrollo Proyecto APT (Semana 9)

|  |  |  |
| --- | --- | --- |
| **Sección:**  001D | | CAPSTONE  PTY4614 |
| **Integrantes :**  2.- Benjamín Sepúlveda | **Presentado a:**  - Juan Pablo Mellado Alarcon  - Jazna Patricia Meza Hidalgo |
| **Institución:**  Duoc UC | **Fecha:**  6 de Octubre 2025 | **Carrera:**  Ingeniería en Informática |

**Indice**

[**Resumen (Abstract) 3**](#_heading=)

[**Desarrollo del Informe Técnico 3**](#_heading=)

[**1. Resumen de avance del Proyecto APT 3**](#_heading=)

[**2. Objetivos específicos y metodología 3**](#_heading=)

[**4. Monitoreo del plan de trabajo 4**](#_heading=)

[**5. Ajustes a partir del monitoreo 4**](#_heading=)

[**Conclusiones individuales 4**](#_heading=)

[**Reflexión individual 4**](#_heading=)

[**Bibliografía 5**](#_heading=)

# Resumen (Abstract)

Este informe presenta los avances alcanzados en el desarrollo del Proyecto APT, enfocado en la creación de una plataforma de monitoreo clínico que permite la continuidad del cuidado, los autoinformes del paciente y la comunicación entre cuidadores, familiares y personal médico. Durante esta etapa, se completó el desarrollo del backend bajo una arquitectura API REST (FastAPI + PostgreSQL), con modelos de datos normalizados, endpoints para la gestión de pacientes y mediciones, y una estructura modular que permite escalar fácilmente. En el frontend, se implementó una base funcional en React + TypeScript, con navegación por roles e integración de servicios para el registro y visualización de datos clínicos. Las evidencias de avance incluyen operaciones CRUD operativas, integración entre backend y frontend, prototipos funcionales, bitácora de commits y documentación técnica. La metodología aplicada sigue un enfoque API-first, con iteraciones cortas, priorización de funcionalidades críticas y aplicación de principios ágiles, asegurando trazabilidad, calidad de datos y resultados medibles.

# Desarrollo del Informe Técnico

## 1. Resumen de avance del Proyecto APT

Backend (API y base de datos): Se diseñó y configuró la API REST por módulos y perfiles de usuario (paciente, cuidador, médico y administrador). Se amplió el esquema de base de datos incorporando tablas y relaciones faltantes como Medición, DetalleMedición, Historiales y Gamificación. También se desarrollaron endpoints CRUD para Paciente y Medición, con validaciones, códigos HTTP adecuados y soporte de transacciones.

Frontend (estado funcional de prueba): Se construyó la base del frontend (React + Vite) con layout unificado, navegación por roles y servicios REST para el consumo de la API. Se integró el flujo de registro de mediciones desde la interfaz del paciente y se simuló el login por perfiles.

Pruebas y validaciones: Se realizaron pruebas de integración entre frontend y backend en el módulo de paciente, ajustando validaciones, manejo de errores y mensajes en la interfaz.

Mi contribución se centró principalmente en la validación de los *schemas* dentro de la API, revisión y depuración del código, ejecución de pruebas y corrección de errores, además del agregado de tablas necesarias para optimizar la estructura del modelo de datos.

## 2. Objetivos específicos y metodología

***Objetivos específicos (versión vigente, sin modificaciones)***

***Continuidad del cuidado.*** *En 6 meses, ≥85% de días con registro por paciente y trazabilidad ≥98%; mejora +5 p.p./mes. Medición semanal, cierre mensual.*

***Respuesta a alertas.*** *En 3 meses, mediana de reconocimiento ≤10 min, P90 ≤30 min y ≥80% de alertas críticas cerradas ≤60 min; mejora +3 p.p./mes. Medición semanal, cierre mensual.*

***Autoinformes (autocuidado).*** *En 6 meses, MAU ≥75%, DAU/MAU ≥40%, ≥3 autoinformes/semana y retención 4 semanas ≥60%; mejora +5 p.p. MAU y +1 informe/mes. Medición semanal, cierre mensual.*

***Comunicación cuidador–familia–equipo.*** *≥90% con cuidador/familiar activo, lectura de notificaciones ≥85% y mediana de respuesta ≤2 h en 4 meses; mejora +4 p.p./mes y −10% en tiempos. Medición semanal, cierre mensual.*

**3. Calidad de datos y analítica**  
*Disponibilidad ≥99.5%, completitud ≥97%, errores ≤1%, backups 100% diarios, y ≥95% de reportes en ≤24 h; mejora +1 p.p./mes hasta meta. Medición diaria, reporte mensual.3. Evidencias del avance*

Las evidencias que acompañan este informe demuestran el desarrollo efectivo y el cumplimiento de los objetivos: Modelo de datos (DDL + ORM), API REST (FastAPI), Servicios de Frontend (TypeScript), Interfaz de usuario (React), Scripts de datos de prueba y Bitácora de commits.

## 4. Monitoreo del plan de trabajo

El seguimiento del plan muestra que las actividades planificadas se ejecutan conforme al cronograma. La mayoría de las tareas aparecen completadas o en curso, y las que aún no inician dependen de las que actualmente están en desarrollo. El proyecto mantiene un ritmo estable gracias a la planificación estructurada, el uso de metodologías ágiles y la gestión de dependencias mediante tableros Kanban.

## 5. Ajustes a partir del monitoreo

Factores que han facilitado el desarrollo: comunicación con ingenieros externos, colaboración efectiva y claridad de objetivos. Factores que han dificultado el desarrollo: limitación de tiempo e integración compleja. Acciones: priorización MVP, planificación semanal, revisión de código y automatización de tareas. No se han realizado ajustes al plan de trabajo; el avance sigue conforme al plan original.

# Conclusiones individuales

La elección del enfoque y la metodología nos dan la certeza del avance del proyecto y del arduo trabajo que conllevará desarrollarlo. Aun con el tiempo cercano al límite, el cumplimiento de los objetivos ha permitido mantener un progreso constante y medible. La integración de herramientas modernas como FastAPI y React ha fortalecido la estructura del sistema, mientras que la colaboración entre los integrantes del equipo ha sido clave para resolver desafíos y mantener la coherencia del desarrollo. En lo personal, este proceso ha representado una experiencia enriquecedora en la aplicación de metodologías ágiles, gestión de versiones y despliegue de servicios, reforzando tanto las competencias técnicas como las de trabajo en equipo.

# Reflexión individual

El desarrollo de este proyecto APT ha sido una gran experiencia, tanto en la aplicación de las habilidades adquiridas durante la carrera como en el crecimiento personal. Aprender nuevos lenguajes y adaptarme a su documentación fue un desafío que fortaleció mi aprendizaje autónomo. Este proceso me permitió afianzar conocimientos en desarrollo de sistemas, organización de tareas y trabajo en equipo. A pesar de las dificultades, el avance logrado refleja compromiso, constancia y motivación por seguir mejorando en futuros proyectos.

# Bibliografía

FastAPI Documentation: https://fastapi.tiangolo.com/  
React Documentation: https://react.dev/  
SQLAlchemy ORM Guide: https://docs.sqlalchemy.org/  
Duoc UC. (2025). Guía de elaboración de informes técnicos APT.