**Guía2. Desarrollo Proyecto APT**

**Asignatura Capstone**

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

|  |  |
| --- | --- |
| Resumen de avance proyecto APT | ***Avances y actividades realizadas***   * ***Back-end (API y BD):***    + *Se diseñó, configuró y probó la* ***API REST*** *por módulos y perfiles de usuario (paciente, cuidador, médico y admin) utilizando un modelo de datos consistente.*   + *Se* ***amplió el esquema de base de datos*** *incorporando tablas y relaciones faltantes (p. ej., medición y detalle de medición, historiales, gamificación, etc.) para cubrir los casos de uso definidos.*   + *Se implementaron* ***endpoints CRUD de Paciente*** *(listar, obtener por RUT, creación, actualización, habilitar/deshabilitar) y* ***endpoints de Medición*** *y* ***MediciónDetalle*** *para registrar y evaluar datos clínicos.* * ***Front-end (estado funcional de prueba):***    + *Se levantó la estructura base del* ***front*** *(React + Vite) con layout unificado, sidebars por rol y navegación.*   + *Se creó un* ***login simulado*** *por roles y se preparó el consumo real de la API en el flujo del* ***paciente****.*   + *Se integró un* ***servicio de pacientes*** *en el front (fetch/REST) y una primera versión de* ***registro de mediciones*** *desde la UI del paciente.* * ***Pruebas y validaciones:***    + *Pruebas de integración entre front y back para el módulo de paciente (lectura e inserción de datos).*   + *Ajustes menores de validación y manejo de errores (mensajes y estados en la UI).*   ***Objetivos específicos cumplidos (parcial o totalmente) y cómo se abordaron***   * ***Continuidad del cuidado:*** *se habilitó el* ***registro clínico trazable*** *mediante entidades de medición y notas, además de vistas enfocadas en paciente/cuidador para facilitar la captura diaria.* * ***Respuesta a alertas:*** *se modelaron campos de* ***tiene\_alerta, severidad y umbrales*** *en medición/detalle, permitiendo calcular y persistir eventos de riesgo para seguimiento.* * ***Autoinformes (autocuidado):*** *el paciente puede* ***registrar mediciones*** *desde la UI y visualizar su progreso; se preparó la estructura para medir frecuencia y retención.* * ***Comunicación con cuidadores/familia:*** *se estructuraron entidades y relaciones (paciente-cuidador) que permiten compartir y consultar información desde distintos perfiles.* * ***Calidad de datos y analítica:*** *se normalizó el* ***modelo de datos****, se agregaron* ***historiales*** *y se definieron* ***endpoints de consulta*** *que permiten generar reportes a futuro.*   ***Ajustes a objetivos / metodología***   * ***Ajuste de objetivos específicos:*** *se precisaron metas en formato KPI (porcentajes, frecuencias y plazos) para facilitar el seguimiento y evaluación. Se mantienen a continuación sin cambios.*   ***Ajuste metodológico:*** *se priorizó un enfoque* ***API-first*** *con iteraciones cortas: primero modelo/BD y endpoints críticos; luego integración gradual en el front con servicios por dominio (paciente/medición) y pruebas de extremo a extremo.* |
| Objetivos | ***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.*  ***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.* |
| Evidencias de avance | *A continuación se detallan las evidencias que adjunto para este informe y la manera en que dan cuenta del desarrollo del proyecto:*   1. ***Modelo de datos (DDL + modelos SQLAlchemy)***    * ***Qué es:*** *script DDL y clases ORM (SQLAlchemy) para entidades clave (Paciente, Medición, MediciónDetalle, Cesfam, Equipo Médico, Cuidador, relaciones y restricciones).*    * ***Por qué demuestra avance:*** *acredita el diseño lógico y físico del dominio, la normalización y la* ***integridad referencial*** *(FK con ondelete, cascadas, índices), habilitando el registro clínico trazable y las mediciones.* 2. ***API REST por dominios (FastAPI) – módulos paciente y medición***    * ***Qué es:*** *routers, esquemas Pydantic (Create/Update/Out), servicios y endpoints CRUD para* ***Paciente****,* ***Medición*** *y* ***MediciónDetalle****; soporte de* ***paginación****,* ***filtros*** *y* ***códigos HTTP*** *apropiados.*    * ***Por qué demuestra avance:*** *evidencia un backend funcional para registrar y consultar datos clínicos, base para continuidad del cuidado, autoinformes y analítica.* 3. ***Servicios de Front-end (TypeScript) para consumo de API***    * ***Qué es:*** *servicios getPacientes, getPacienteByRut, createMedicion y createMedicionWithDetails (manejo de errores y tipado).*    * ***Por qué demuestra avance:*** *valida la* ***integración front–back****, permitiendo flujo real de datos de paciente y registro de mediciones.* 4. ***Interfaz de usuario (React) por perfiles***    * ***Qué es:*** *vistas funcionales de* ***Paciente*** *(Dashboard, Medición, Progreso, Medicación),* ***Cuidador*** *y* ***Administrador*** *con componentes reutilizables (shadcn/ui) y gráficos (Recharts).*    * ***Por qué demuestra avance:*** *muestra la capa de presentación operativa, navegable y conectada a la API (en el caso del paciente), alineada con los objetivos de autocuidado, seguimiento y gestión.* 5. ***Scripts de datos de prueba / inserciones controladas***    * ***Qué es:*** *scripts de inserts y carga de datos con uso de* ***transacciones*** *y estrategias de* ***rollback*** *frente a conflictos (p. ej., claves duplicadas).*    * ***Por qué demuestra avance:*** *permiten validar escenarios end-to-end y comprobar reglas de negocio, umbrales y relaciones sin afectar la integridad.* 6. ***Capturas de funcionamiento y bitácora de commits***    * ***Qué es:*** *capturas de UI (flujos de login simulado, registro de mediciones y consultas) y* ***historial de commits*** *con mensajes descriptivos.*    * ***Por qué demuestra avance:*** *documentan el comportamiento real del sistema y la traza de desarrollo, decisiones y correcciones realizadas.*   ***Todo el material anterior se encuentra versionado en mi repositorio de GitHub, con commits por funcionalidad, mensajes descriptivos y organización por módulos. Las evidencias adjuntas en este informe corresponden a los mismos artefactos registrados en los commits.*** |

|  |
| --- |
| **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-1) | Observaciones | Estado de avance | Ajustes |
| *Gestión de proyectos y comunicación con stakeholders* | *Reunión de arranque* | *Plataforma de videoconferencia, presentaciones, actas* | *7* | *Erwin Nuñez, Benjamin Sepúlveda* | *Conflictos de prioridades* | *Completada* |  |
| *Documentación técnica y gestión del conocimiento* | *Creación de documentos iniciales* | *Suite ofimática, repositorio documental* | *7* | *Erwin Nuñez, Benjamin Sepúlveda* | *Información incompleta* | *Completada* |  |
| *Gestión ágil (Scrum)* | *Definición de roles y responsabilidades* | *Recursos estándar de desarrollo* | *7* | *Benjamin Sepúlveda* | *Resistencia al cambio* | *Completada* |  |
| *Administración de configuración de ambientes* | *Configuración del entorno de trabajo* | *Repositorio Git, CI/CD, entornos de desarrollo* | *7* | *Erwin Nuñez* | *Errores de versión* | *Completada* |  |
| *Gestión de proyectos y comunicación con stakeholders* | *Reunión con stakeholders* | *Plataforma de videoconferencia, presentaciones, actas* | *7* | *Erwin Nuñez, Benjamin Sepúlveda* | *Falta de alineación* | *Completada* |  |
| *Levantamiento y análisis de requerimientos* | *Identificación de requerimientos funcionales* | *Encuestas/entrevistas, plantillas de requisitos, tablero* | *7* | *Benjamin Sepúlveda* | *Requerimientos ambiguos* | *Completada* |  |
| *Levantamiento y análisis de requerimientos* | *Identificación de requerimientos no funcionales* | *Encuestas/entrevistas, plantillas de requisitos, tablero* | *7* | *Benjamin Sepúlveda* | *Falta de claridad* | *Completada* |  |
| *Gestión ágil (Scrum)* | *Priorización del backlog inicial* | *Recursos estándar de desarrollo* | *7* | *Erwin Nuñez* | *Retrasos en entregas* | *Completada* |  |
| *Diseño arquitectónico de soluciones* | *Diseño de la arquitectura* | *Herramientas de modelado (Draw.io, ERD), repositorio* | *18* | *Erwin Nuñez* | *Complejidad técnica* | *Completada* |  |
| *Modelamiento y gestión de datos* | *Diseño del modelo de datos* | *Herramientas de modelado (Draw.io, ERD), repositorio* | *18* | *Erwin Nuñez* | *Datos inconsistentes* | *Completada* |  |
| *Diseño de experiencia de usuario (UX/UI)* | *Prototipado de interfaz de usuario* | *Figma/Adobe XD, guía de estilos* | *18* | *Benjamin Sepúlveda* | *Retroalimentación insuficiente* | *Completada* |  |
| *Competencia general de ingeniería en informática* | *Validación del diseño con el equipo* | *Recursos estándar de desarrollo* | *18* | *Benjamin Sepúlveda* | *Actualización tecnológica* | *Completada* |  |
| *Desarrollo e integración de software* | *Implementación del registro de parámetros clínicos* | *IDE, frameworks web, base de datos, API* | *17* | *Erwin Nuñez, Benjamin Sepúlveda* | *Bugs frecuentes* | *En curso* |  |
| *Desarrollo e integración de software* | *Implementación de gestión de usuarios (pacientes, cuidadores, médicos)* | *IDE, frameworks web, base de datos, API* | *17* | *Erwin Nuñez, Benjamin Sepúlveda* | *Integración difícil* | *En curso* |  |
| *Desarrollo e integración de software* | *Integración básica de base de datos* | *IDE, frameworks web, base de datos, API* | *17* | *Erwin Nuñez, Benjamin Sepúlveda* | *Dependencias externas* | *En curso* |  |
| *Aseguramiento de la calidad de software (QA)* | *Pruebas iniciales de los módulos desarrollados* | *Frameworks de testing, datos de prueba, tablero QA* | *17* | *Erwin Nuñez, Benjamin Sepúlveda* | *Pruebas incompletas* | *No iniciado* |  |
| *Desarrollo e integración de software* | *Desarrollo del sistema de alertas automáticas* | *IDE, frameworks web, base de datos, API* | *17* | *Erwin Nuñez, Benjamin Sepúlveda* | *Cambios frecuentes* | *No iniciado* |  |
| *Desarrollo e integración de software* | *Desarrollo del dashboard médico interactivo* | *IDE, frameworks web, base de datos, API* | *17* | *Erwin Nuñez, Benjamin Sepúlveda* | *Conflictos de código* | *No iniciado* |  |
| *Aseguramiento de la calidad de software (QA)* | *Pruebas unitarias y de integración* | *Frameworks de testing, datos de prueba, tablero QA* | *17* | *Erwin Nuñez, Benjamin Sepúlveda* | *Escenarios no contemplados* | *No iniciado* |  |
| *Gestión de proyectos y comunicación con stakeholders* | *Feedback del equipo médico* | *Plataforma de videoconferencia, presentaciones, actas* | *17* | *Erwin Nuñez, Benjamin Sepúlveda* | *Falta de comunicación* | *No iniciado* |  |
| *Desarrollo e integración de software* | *Implementación de notificaciones a familiares/cuidadores* | *IDE, frameworks web, base de datos, API* | *18* | *Erwin Nuñez, Benjamin Sepúlveda* | *Falta de documentación* | *No iniciado* |  |
| *Desarrollo e integración de software* | *Desarrollo de reportes personalizados* | *IDE, frameworks web, base de datos, API* | *18* | *Erwin Nuñez, Benjamin Sepúlveda* | *Requerimientos cambiantes* | *No iniciado* |  |
| *Competencia general de ingeniería en informática* | *Validación con pacientes y cuidadores* | *Recursos estándar de desarrollo* | *18* | *Erwin Nuñez, Benjamin Sepúlveda* | *Conocimientos desactualizados* | *No iniciado* |  |
| *Competencia general de ingeniería en informática* | *Ajustes según retroalimentación* | *Recursos estándar de desarrollo* | *18* | *Erwin Nuñez, Benjamin Sepúlveda* | *Falta de experiencia práctica* | *No iniciado* |  |
| *Aseguramiento de la calidad de software (QA)* | *Pruebas finales del sistema* | *Frameworks de testing, datos de prueba, tablero QA* | *1* | *Erwin Nuñez, Benjamin Sepúlveda* | *Cobertura limitada* | *No iniciado* |  |
| *Gestión de proyectos y comunicación con stakeholders* | *Capacitación a usuarios* | *Plataforma de videoconferencia, presentaciones, actas* | *1* | *Erwin Nuñez, Benjamin Sepúlveda* | *Stakeholders no disponibles* | *No iniciado* |  |
| *Documentación técnica y gestión del conocimiento* | *Documentación final* | *Suite ofimática, repositorio documental* | *1* | *Erwin Nuñez, Benjamin Sepúlveda* | *Duplicidad de información* | *No iniciado* |  |
| *Gestión de proyectos y mejora continua* | *Entrega del proyecto y retrospectiva* | *Presentación, repositorio, checklist de entrega* | *1* | *Erwin Nuñez, Benjamin Sepúlveda* | *Resistencia a mejoras* | *No iniciado* |  |

|  |
| --- |
| **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**  **Facilitadores.** El principal facilitador ha sido la **comunicación con ingenieros externos al proyecto**. Esta interacción nos entregó ideas frescas y validaciones tempranas de diseño, permitiéndonos **anticipar riesgos técnicos** y corregir supuestos antes de implementarlos. Además, el trabajo colaborativo y la retroalimentación externa aceleraron la toma de decisiones.  **Dificultades.** La mayor dificultad ha sido la **limitación de tiempo efectivo de desarrollo del equipo**, lo que tensionó los plazos de integración y pruebas.  **Acciones realizadas y próximas para mitigarlo.**   * **Priorización estricta** (enfoque MVP) y recorte de “nice to have” para proteger el camino crítico. * **Planificación en sprints cortos** con metas semanales medibles y tableros Kanban (límites WIP) para mejorar foco y flujo. * **Bloques de tiempo en calendario** y sincronizaciones breves (stand-ups) para alinear tareas y destrabar dependencias. * **Paralelización y pairing** en piezas críticas; revisión de código puntual con pares externos. * **Automatización y plantillas** (scripts de datos de prueba, generación de endpoints/servicios repetitivos) para reducir tiempo operativo.   Estas medidas ya están en curso y continuarán reforzándose para asegurar ritmo sostenible sin comprometer calidad. |

|  |
| --- |
| **Actividades ajustadas o eliminadas**  No se han **eliminado ni ajustado** actividades del plan de trabajo. El proyecto avanza conforme a la planificación original (WBS/hitos) y al cronograma previsto.  **Justificación (por qué no hubo ajustes):**   * **Alcance acotado a un MVP** con criterios de salida claros y definición de “listo” (DoD), evitando desbordes. * **Estimaciones realistas y buffers** en tareas críticas, lo que absorbió variaciones menores sin tocar la ruta crítica. * **Validación temprana y continua** con actores clave e ingenieros externos, permitiendo correcciones de diseño antes de ejecutar. * **Reutilización de componentes** (UI, plantillas de API, scripts de datos de prueba) que redujo incertidumbre y tiempos. * **Gestión de riesgos activa**: riesgos identificados y mitigados sin impacto en alcance, costos ni plazos. * **Gobierno de cambios**: se evaluaron solicitudes; ninguna ameritó cambio formal en el plan.   Seguiremos monitoreando indicadores y riesgos; si emergen desviaciones, activaremos las contingencias definidas sin comprometer los objetivos específicos. |

|  |
| --- |
| **Actividades no iniciadas o retrasadas**  A la fecha **no hay actividades retrasadas** respecto del plan. Las tareas **aún no iniciadas** son **dependencias naturales** de los paquetes de trabajo actualmente en ejecución (p. ej., integración final y pruebas end-to-end que requieren que la API y el front queden cerrados en sus contratos).  **Estrategia para evitar desvíos y asegurar el avance:**   * **Gestión de dependencias y ruta crítica:** tablero Kanban con enlaces de dependencias y límites de WIP para evitar cuellos de botella. * **Desacople temprano:** uso de *mocks/stubs* y contratos API versionados para avanzar en paralelo (front/back) sin bloquearse. * **Integración continua:** pipelines CI/CD con pruebas unitarias y de contrato para detectar quiebres de inmediato. * **Criterios de aceptación claros (DoD):** evitar “trabajo casi listo” y garantizar que cada entrega sea integrable. * **Sincronización corta y frecuente:** *stand-ups* diarios y *demos* semanales para visibilidad y resolución temprana de impedimentos. * **Gestión de riesgos activa:** revisión semanal de riesgos y activación de planes de contingencia si surge algún bloqueo.   Con estas prácticas, las tareas dependientes podrán iniciarse apenas se complete su prerrequisito, manteniendo el cronograma del Proyecto APT. |

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-1)