

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

| Resumen de avance proyecto APT | Durante el Sprint 1 (Semanas 1-3), el equipo ha completado exitosamente actividades fundacionales del proyecto OfiSync. Se ha llevado a cabo el alistamiento de los ambientes de desarrollo y pruebas, y la creación de la base de datos centralizada.  En paralelo, se desarrollaron los primeros módulos funcionales clave: el "Módulo de gastos comunes" para el portal web (incluyendo interfaz y backend) y el "Módulo de reservas" para la aplicación móvil. Finalmente, se ejecutó el ciclo de testing inicial para estos módulos.  Estos avances cumplen directamente con los siguientes objetivos específicos:   * "Desarrollar un portal web con vistas diferenciadas..." (Iniciado con el módulo de gastos comunes). * "Desarrollar una aplicación móvil orientada a clientes..." (Iniciado con la funcionalidad de reserva de servicios). * "Asegurar la sincronización en tiempo real... a través de una base de datos centralizada..." (Logrado mediante la configuración de la BDD). * "Estructurar una base de datos robusta...". * "Desarrollar un sistema de reservas eficiente...".   A pesar de las dificultades encontradas, como problemas de compatibilidad que obligaron a un cambio de lenguaje de programación y desafíos en la implementación del testing, el equipo logró adaptarse investigando activamente y aplicando los conocimientos (React, Jest) para cumplir con las actividades definidas en los tiempos establecidos. |
| --- | --- |
| Objetivos | No se han realizado ajustes a los objetivos del proyecto definidos en la Fase 1. |
| Metodología | No se han realizado ajustes a la metodología Scrum. Esta se ha mantenido y ha sido fundamental para gestionar las dificultades técnicas imprevistas (como el cambio de lenguaje de programación) de manera eficiente, permitiendo al equipo adaptarse y cumplir con los plazos del sprint. |
| Evidencias de avance | Las evidencias presentadas para este avance corresponden a los entregables clave del Sprint 1, según lo planificado:   1. Product Backlog 2. Sprint Backlog del Primer Sprint: Documento que detalla las historias de usuario y tareas técnicas seleccionadas del Product Backlog para ser desarrolladas en este primer ciclo. 3. Entregables Funcionales (Software):    * Módulo de gastos comunes completo (web).    * Módulo de reservas completo (móvil). 4. Reporte de Testing del Primer Sprint: Resultados de las pruebas unitarias y manuales realizadas a los módulos desarrollados. 5. Acta de Retrospectiva del Primer Sprint: Documento que resume los aprendizajes, problemas y mejoras detectadas al cierre del Sprint 1.   Justificación y Resguardo de Calidad: Estas evidencias demuestran el avance real del proyecto al mostrar un ciclo Scrum completo. El Sprint Backlog evidencia la planificación detallada. Los módulos funcionales son la prueba tangible de que se ha construido software funcional que cumple con los objetivos específicos y da cuenta del desarrollo de las competencias de "Desarrollar una solución de software" y "Construir programas y rutinas".  Para resguardar la calidad, se aplicó la metodología Scrum y se realizaron actividades de testing (Competencia: "Realizar pruebas de calidad"). Aunque se presentaron dificultades en la implementación de pruebas (Jest), el equipo resguardó la calidad mediante la investigación activa de la documentación y la aplicación de las mejores prácticas identificadas para validar los componentes. El Acta de Retrospectiva demuestra el compromiso con la mejora continua del proceso. |

| **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 Proyecto APT** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable[[1]](#footnote-0) | Observaciones | Estado de avance | Ajustes |
| Comunicación y coordinación de equipo. | Reuniones diarias | Reuniones de 15 min cada día para sincronizar avances, bloqueos y próximos pasos. | Videollamadas, chat, tablero Scrum. | 15 a 20 minutos por día/Todo el proyecto | Equipo completo | Clave para mantener alineación y comunicación. | En curso | Ninguno |
| Administrar la configuración de ambientes, servicios de aplicaciones y bases de datos. | Configuración VPS y ambientes | Configuración de servidor AWS, instalación de dependencias y setup de entornos de desarrollo/pruebas. | AWS, Docker, Git. | 2 días | Equipo completo | Actividad inicial obligatoria para todo el desarrollo. | Completado | Ninguno |
| Construir modelos de datos escalables / Programar consultas o rutinas para manipular información en la base de datos. | Configuración base de datos | Diseño y creación de la base centralizada en PostgreSQL. | PostgreSQL, pgAdmin. | 3 días | Equipo completo | Requiere definir esquema y relaciones iniciales. | Completado | Ninguno |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Interfaz módulo gastos comunes (web) | Pantallas para visualización y edición de gastos comunes en Django frontend. | React, Vite | 1 semana | John Herrera | Se conecta a API de gastos comunes. | Completado | Se cambió la tecnología de "Django frontend" a "React, Vite" por problemas de compatibilidad. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Construir programas y rutinas de variada complejidad con buenas prácticas de codificación. | Backend cálculo de gastos comunes | Desarrollo de lógica y endpoints para el cálculo de gastos comunes. | Node, JavaScript Express, PostgreSQL. | 2 semanas | Erick San Martín y John Herrera | Integración directa con frontend y app. | Completado | Se cambió la tecnología a Node. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Módulo de reservas (app móvil) | Desarrollo de pantallas y lógica de reservas en Ionic. | React Native, Expo, Node, JavaScript Express, PostgreSQL. | 2 semanas | Alexander Pulgar | Requiere endpoints backend de reservas. | Completado | Se cambió la tecnología de "Ionic" a "React Native, Expo" por problemas de compatibilidad. |
| Realizar pruebas de calidad tanto de los productos como de los procesos. | Testing inicial | Pruebas de los módulos de gastos comunes y reservas. | Jest, pruebas manuales. | Media semana | Equipo completo | Validación del primer incremento del producto. | Completado | Requirió más tiempo del estimado debido a dificultades en la implementación de Jest, necesitando investigación adicional. |
| Realizar pruebas de calidad tanto de los productos como de los procesos. | Interfaz inventario de insumos (web) | Desarrollo de formulario web para CRUD de insumos. | React, Vite | 1 semana | John Herrera | Validaciones de stock. | No iniciado | Ninguno |
| Construir modelos de datos escalables / Programar consultas o rutinas para manipular información en la base de datos. | Backend inventario de insumos | API CRUD para insumos con persistencia en PostgreSQL. | Node, JavaScript Express, PostgreSQL. | 2 semanas | Erick San Martín y John Herrera | Se conecta con interfaz web. | No iniciado | Ninguno |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Módulo carga de comprobantes (app móvil) | Función en la app para subir comprobantes (PDF/imagen). | React Native, Expo, Node, JavaScript Express, PostgreSQL. | 1 semana | Alexander Pulgar | Validar formato de archivo antes de subir | No iniciado | Ninguno |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Módulo consulta de gastos comunes (app móvil) | Pantalla en app para visualizar gastos comunes a pagar. | React Native, Expo, Node, JavaScript Express, PostgreSQL. | 1 semana | Alexander Pulgar | Requiere API estable de gastos comunes. | No iniciado | Ninguno |
| Realizar pruebas de calidad tanto de los productos como de los procesos. | Testing intermedio | Pruebas de inventario, comprobantes y gastos comunes en móvil. | Jest, pruebas manuales. | Media semana | Equipo completo | Validaciones cruzadas de módulos. | No iniciado | Ninguno |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Interfaz consulta de reservas (web) | Pantallas para que conserje/administrador vea reservas. | React, Vite | Media semana | John Herrera | Se conecta al backend de reservas. | No iniciado | Ninguno |
| Construir programas y rutinas de variada complejidad / Implementar soluciones sistémicas integrales. | Backend consulta de reservas | Endpoints para devolver reservas al frontend web. | Node, JavaScript Express, PostgreSQL. | 1 semana | Erick San Martín y John Herrera | Conexión con módulo de reservas móvil. | No iniciado | Ninguno |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Interfaz bitácora (web) | Pantalla para notas, control de acceso, registro de bitácora. | React, Vite | Media semana | John Herrera | Sin observaciones. | No iniciado | Ninguno |
| Construir programas y rutinas de variada complejidad / Programar consultas o rutinas para manipular información en la base de datos. | Backend bitácora | API CRUD para almacenar registros de bitácora. | Node, JavaScript Express, PostgreSQL. | 1 semana | Erick San Martín y John Herrera | Debe garantizar persistencia y consulta rápida. | No iniciado | Ninguno |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Vista login y registro (web) | Desarrollo de pantallas de login y registro de usuarios. | React, Vite | Media semana | John Herrera | Requiere integración con backend. | No iniciado | Ninguno |
| Construir programas y rutinas de variada complejidad / Resolver vulnerabilidades sistémicas y cumplir normas de seguridad. | Backend login y registro | API de autenticación y roles de usuario. | Node, JavaScript Express, PostgreSQL. | Media semana | Erick San Martín | Incluye validación y cifrado de credenciales. | No iniciado | Ninguno |
| Implementar soluciones sistémicas integrales / Resolver vulnerabilidades sistémicas y cumplir normas de seguridad. | Login app móvil | Implementación de login en la app móvil (Ionic). | React Native, Expo, Node, JavaScript Express, PostgreSQL. | Media semana | Alexander Pulgar | Sin observaciones | No iniciado | Ninguno |
| Construir el modelo arquitectónico de la solución sistémica / Implementar soluciones sistémicas integrales. | Integración de módulos | Conexión de todos los módulos web, backend y app móvil. | Git, APIs REST, Postman | 2 semanas | Equipo completo | Puede requerir ajustes de endpoints y UI. | No iniciado | Ninguno |
| Realizar pruebas de calidad tanto de los productos como de los procesos / Implementar soluciones sistémicas integrales. | Testing final y cierre | Pruebas de regresión, funcionales y de usabilidad para el sistema completo. | Jest, pruebas manuales. | 1 semana | Equipo completo | Validación final previa a puesta en marcha. | No iniciado | Ninguno |

| **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:   * Trabajo en equipo y comunicación: La capacidad del grupo para entendernos y trabajar de manera armoniosa ha sido fundamental. Todos los miembros hemos trabajado a la par, lo que nos permitió enfrentar los desafíos de manera coordinada. * Metodología Scrum: La aplicación de la metodología ágil nos ha proporcionado la flexibilidad necesaria para adaptarnos a los cambios y problemas imprevistos sin perder el ritmo de trabajo. * Proactividad en la investigación: El equipo ha demostrado una gran capacidad para la autoformación, lo que nos ha permitido resolver dudas y problemas técnicos de forma autónoma.   Factores que han dificultado el desarrollo:   * Problemas de compatibilidad tecnológica: La mayor dificultad fue encontrar problemas de compatibilidad con las tecnologías que se habían planificado originalmente (Django y Ionic), lo que nos obligó a realizar un cambio significativo en las herramientas de desarrollo. * Implementación del Testing: El proceso de configuración y ejecución de pruebas (específicamente con React y Jest) resultó más complejo de lo anticipado, requiriendo un tiempo de aprendizaje y depuración mayor al estimado.   Acciones para solucionar las dificultades: Para superar estos obstáculos, investigamos activamente, revisando la documentación oficial de las nuevas tecnologías (React, React Native, Jest), viendo tutoriales en video y aplicando de manera práctica lo que íbamos entendiendo. Esta estrategia de prueba y error nos permitió validar las soluciones y avanzar con el desarrollo. |
| --- |

| **Actividades ajustadas o eliminadas:**  Se han realizado los siguientes ajustes técnicos en las actividades, pero no se han eliminado tareas del plan:   * Ajuste en "Interfaz módulo gastos comunes (web)": Se cambió la tecnología de desarrollo de "Django frontend" a "React, Vite". * Ajuste en "Módulo de reservas (app móvil)": Se cambió la tecnología de desarrollo de "Ionic" a "React Native, Expo". * Ajuste en "Testing inicial": Esta actividad requirió más tiempo del planificado debido a la curva de aprendizaje de Jest.   Justificación de los ajustes: Los cambios de tecnología fueron necesarios e inevitables debido a los problemas de compatibilidad que impedían el correcto desarrollo de la aplicación. Se optó por un stack tecnológico basado en JavaScript (React/React Native) para asegurar la coherencia y viabilidad del proyecto.  El plan de trabajo en cuanto a la distribución de actividades y funcionalidades a desarrollar no ha requerido ajustes mayores, ya que el alcance y los objetivos del proyecto se mantienen. Los facilitadores, como la buena coordinación del equipo y la metodología ágil, nos han permitido absorber estos cambios técnicos sin necesidad de reestructurar el cronograma general. |
| --- |

| **Actividades que no has iniciado o están retrasadas:**  Actividades no iniciadas: Todas las actividades correspondientes a los Sprints 2 y 3 se encuentran en estado "No iniciado", lo cual es normal y está alineado con la planificación del proyecto. Estas actividades se abordarán en las próximas semanas según el cronograma establecido.  Actividades retrasadas: Actualmente, no hay actividades retrasadas que hayan incumplido los plazos de entrega del Sprint 1. Aunque la tarea de "Testing inicial" presentó dificultades, el equipo logró completarla dentro del sprint gracias a un esfuerzo adicional en investigación y desarrollo.  Estrategias para el futuro: Para evitar retrasos en las futuras fases de testing, utilizaremos la experiencia y el conocimiento adquirido durante el Sprint 1. Se asignará tiempo para una planificación más detallada de las pruebas y se continuará con la estrategia de investigación proactiva para resolver cualquier nuevo desafío técnico que pueda surgir. |
| --- |

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)