

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

| Resumen de avance proyecto APT | ***Desarrollo de historias de usuario****: Hemos completado las cinco épicas principales, desglosándolas en historias de usuario que cubren las funcionalidades esenciales de la aplicación, como el registro de usuarios, creación de rutinas personalizadas, seguimiento de progreso y visualización de datos de salud. Este trabajo nos ha permitido establecer una base clara para el desarrollo.*  ***Especificación de Requisitos Software (ERS)****: He estado trabajando en la elaboración de este documento, donde definimos el propósito, alcance y objetivos del sistema, así como los actores involucrados. Esta planificación es clave para asegurar que todas las necesidades del cliente se reflejen correctamente en el desarrollo.*  ***Ajustes al proyecto****: Originalmente, se había contemplado la integración con dispositivos wearables, pero decidimos eliminar esta funcionalidad para concentrarnos en las características fundamentales, como el seguimiento de rutinas y el análisis de datos de salud. Este ajuste nos ha permitido priorizar aspectos más importantes para mejorar la experiencia del usuario.*  ***Planificación de sprints****: Como Scrum Master, he coordinado las sesiones de planificación de los sprints y he gestionado el backlog del proyecto. También implementé un burndown chart para monitorear el progreso de nuestro sprint de 20 días, asegurando que el equipo se mantenga alineado con los plazos y objetivos.* |
| --- | --- |
| Objetivos | ***Desarrollar una aplicación integral*** *para usuarios de gimnasio, que facilite la personalización de rutinas de ejercicios y el seguimiento de progreso físico.*  ***Permitir el registro y gestión de usuarios*** *a través de correo electrónico o redes sociales, con opciones para editar el perfil y recuperar contraseñas.*  ***Crear funcionalidades para el seguimiento del progreso*** *de entrenamiento y salud, incluyendo el registro de datos como peso, IMC, y porcentaje de grasa corporal, además de la visualización de gráficos y análisis automáticos.*  ***Proporcionar herramientas de motivación*** *mediante recordatorios de entrenamiento y notificaciones sobre hitos alcanzados en los entrenamientos.*  ***Generar recomendaciones personalizadas basadas en los datos de salud y progreso del usuario, para mejorar la efectividad de sus entrenamientos.***  ***Optimizar la experiencia del usuario al ofrecer sugerencias de ejercicios alternativos y la posibilidad de ajustar sus rutinas de acuerdo con sus objetivos o disponibilidad.***  ***Desarrollar una aplicación integral para usuarios de gimnasio, que facilite la personalización de rutinas de ejercicios y el seguimiento de progreso físico.***  ***Permitir el registro y gestión de usuarios a través de correo electrónico o redes sociales, con opciones para editar el perfil y recuperar contraseñas.***  ***Crear funcionalidades para el seguimiento del progreso de entrenamiento y salud, incluyendo el registro de datos como peso, IMC, y porcentaje de grasa corporal, además de la visualización de gráficos y análisis automáticos.***  ***Proporcionar herramientas de motivación mediante recordatorios de entrenamiento y notificaciones sobre hitos alcanzados en los entrenamientos.***  ***Generar recomendaciones personalizadas basadas en los datos de salud y progreso del usuario, para mejorar la efectividad de sus entrenamientos.***  ***Optimizar la experiencia del usuario al ofrecer sugerencias de ejercicios alternativos y la posibilidad de ajustar sus rutinas de acuerdo con sus objetivos o disponibilidad.*** |
| Metodología | *La metodología del proyecto Hawk Fit se basa en* ***Scrum****, con sprints de 20 días para desarrollar funcionalidades. Utilizamos* ***historias de usuario*** *para organizar el trabajo, y llevamos a cabo* ***reuniones diarias*** *para coordinar el equipo. Un* ***burndown chart*** *nos ayuda a seguir el progreso y ajustar el trabajo según sea necesario. Este enfoque ágil nos permite adaptarnos rápidamente y entregar valor continuo.* |
| Evidencias de avance | *Describe la(s) evidencia(s) que presentarás en este informe de avance y justifica de qué manera esta(s) evidencia(s) permite(n) dar cuenta del desarrollo del proyecto.*  *En caso de ser pertinente explica cómo resguardaste la calidad de tu Proyecto APT a partir de aspectos propios de tu disciplina (correcta aplicación de metodologías, actividades, herramientas, recursos propios, etc.).*  *En este apartado adjuntar la(s) evidencia(s) seleccionada(s) para ser evaluada por el docente.* |
| **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 |
| Gestión de proyectos (Scrum, planificación y control).  Desarrollo de software (diseño, codificación y pruebas).  Análisis de requisitos (documentación y especificación).  Diseño UX/UI (mejora de usabilidad y experiencia).  Análisis de datos (procesamiento y visualización de métricas).  Seguridad informática (protección de datos y autenticación).  Comunicación y trabajo en equipo (coordinación y colaboración). | Recolección de requisitos.  Desarrollo de historias de usuario.  Diseño de la arquitectura del sistema.  Codificación de funcionalidades.  Pruebas de funcionalidad.  Integración y despliegue.  Análisis de datos.  Revisión y ajustes según feedback.  Gestión de sprints y backlog.  Documentación técnica y de usuario. | Humanos:  Desarrolladores (front-end y back-end).  Diseñador UX/UI.  Scrum Master.  Tecnológicos:  Herramientas de desarrollo (IDE, lenguajes de programación).  Software de gestión de proyectos (JIRA, Trello).  Plataforma de control de versiones (Git).  Materiales:  Documentación técnica y de requisitos.  Financieros:  Presupuesto para software y herramientas.  Tiempo:  Planificación de sprints. | Recolección de requisitos: 1 semana  Desarrollo de historias de usuario: 1 semana  Diseño de la arquitectura del sistema: 1 semana  Codificación de funcionalidades: 4 semanas (puede variar según la complejidad)  Pruebas de funcionalidad: 2 semanas  Integración y despliegue: 1 semana  Análisis de datos: 1 semana  Revisión y ajustes según feedback: 1 semana (continuo durante el desarrollo)  Gestión de sprints y backlog: 1 semana (al inicio, luego de manera continua)  Documentación técnica y de usuario: 1 semana (a lo largo del proyecto, pero se puede concentrar al final) | *Recolección de requisitos: Scrum Master (Maximiliano Aedo)*  *Desarrollo de historias de usuario: Scrum Master (Maximiliano Aedo)*  *Diseño de la arquitectura del sistema: Desarrollador back-end (Iván Aguilera)*  *Codificación de funcionalidades:*  *Desarrollador front-end (Jasson Guerra)*  *Desarrollador back-end (Iván Aguilera)*  *Pruebas de funcionalidad: Tester o QA (si se asigna; si no, puede ser responsabilidad de los desarrolladores)*  *Integración y despliegue: Desarrollador back-end (Iván Aguilera)*  *Análisis de datos: Analista de datos (si se asigna; si no, puede ser responsabilidad del Scrum Master)*  *Revisión y ajustes según feedback: Scrum Master (Maximiliano Aedo) y Desarrolladores (Jasson Guerra e Iván Aguilera)*  *Gestión de sprints y backlog: Scrum Master (Maximiliano Aedo)*  *Documentación técnica y de usuario: Scrum Master (Maximiliano Aedo) y Desarrolladores (Jasson Guerra e Iván Aguilera)* | Recolección de requisitos:  Dificultades: Falta de claridad.  Facilitadores: Técnicas de entrevista.  Desarrollo de historias de usuario:  Dificultades: Priorizar historias.  Facilitadores: Colaboración del equipo.  Diseño de arquitectura del sistema:  Dificultades: Complejidad técnica.  Facilitadores: Experiencia previa.  Codificación de funcionalidades:  Dificultades: Comunicación entre desarrolladores.  Facilitadores: Metodologías ágiles.  Pruebas de funcionalidad:  Dificultades: Tiempo limitado.  Facilitadores: Herramientas automatizadas.  Integración y despliegue:  Dificultades: Problemas de compatibilidad.  Facilitadores: Entornos de prueba.  Análisis de datos:  Dificultades: Obtención de datos.  Facilitadores: Herramientas de análisis.  Revisión y ajustes:  Dificultades: Resistencia al cambio.  Facilitadores: Cultura de mejora.  Gestión de sprints:  Dificultades: Falta de claridad en prioridades.  Facilitadores: Herramientas de gestión.  Documentación:  Dificultades: Falta de tiempo.  Facilitadores: Plantillas claras. | Recolección de requisitos: Completado  Desarrollo de historias de usuario: Completado  Diseño de arquitectura del sistema: En curso  Codificación de funcionalidades: En curso  Pruebas de funcionalidad: No iniciado  Integración y despliegue: No iniciado  Análisis de datos: No iniciado  Revisión y ajustes según feedback: En curso  Gestión de sprints y backlog: En curso  Documentación técnica y de usuario: En curso | Ajuste en los requisitos: Se clarificaron algunos requisitos iniciales tras la recolección de información, mejorando la comprensión de las necesidades del usuario.  Reformulación de historias de usuario: Se simplificaron y priorizaron algunas historias de usuario para facilitar su implementación en las primeras etapas del desarrollo.  Modificación en la metodología: Se decidió implementar revisiones semanales en lugar de quincenales para mantener un mejor control del avance y los problemas que puedan surgir.  Ajuste en el cronograma: Se extendió el tiempo estimado para la codificación de funcionalidades debido a la complejidad técnica identificada en las fases iniciales.  Incorporación de feedback temprano: Se estableció un proceso de revisión continua para ajustar las funcionalidades en base a comentarios recibidos, en lugar de esperar al final de cada sprint. |

| **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. **Colaboración del equipo**: La buena comunicación y el trabajo en equipo han permitido avanzar rápidamente en las actividades.    * **Acción**: Se han realizado reuniones diarias para asegurar que todos estén alineados y resolver dudas al instante. 2. **Uso de herramientas ágiles**: La implementación de metodologías ágiles ha facilitado la organización y gestión del proyecto.    * **Acción**: Se utilizan herramientas como JIRA para el seguimiento del progreso y la gestión del backlog. 3. **Claridad en los objetivos**: Tener una visión clara del proyecto y de los objetivos específicos ha permitido a todos los miembros del equipo trabajar en la misma dirección.    * **Acción**: Se realizaron sesiones de alineación al inicio del proyecto para establecer objetivos comunes.  **Factores que han dificultado el desarrollo:**  1. **Complejidad técnica**: Algunos aspectos de la codificación han resultado más complejos de lo anticipado, lo que ha retrasado el avance.    * **Acción**: Se decidió realizar sesiones de capacitación y revisión técnica entre los desarrolladores para abordar los problemas específicos. 2. **Cambios en los requisitos**: Durante la recolección de requisitos, surgieron cambios que afectaron el diseño inicial.    * **Acción**: Se implementó un proceso de revisión continua para ajustar los requisitos según el feedback recibido, evitando confusiones futuras. 3. **Limitaciones de tiempo**: La presión del tiempo ha dificultado la realización de pruebas exhaustivas.    * **Acción**: Se establecieron hitos intermedios para priorizar las funcionalidades más críticas y asegurar que se realicen pruebas básicas antes del despliegue. 4. **Falta de recursos humanos**: La falta de un tester o QA dedicado ha limitado la capacidad de realizar pruebas de manera efectiva.    * **Acción**: Se considerará la posibilidad de contratar a un tester temporal o redistribuir las responsabilidades dentro del equipo para cubrir esta área. |
| --- |

| Actividades ajustadas o eliminadas: ***Actividades ajustadas:***  1. ***Revisión de historias de usuario****:*    * ***Ajuste****: Se simplificaron y priorizaron las historias de usuario para enfocarse en las funcionalidades esenciales en las primeras fases del desarrollo.*    * ***Justificación****: Este ajuste se realizó para facilitar la implementación inicial y asegurar que el equipo pudiera entregar valor rápidamente al usuario, permitiendo iteraciones basadas en feedback temprano.* 2. ***Ajuste en el cronograma de desarrollo****:*    * ***Ajuste****: Se extendió el tiempo asignado a la codificación de funcionalidades debido a la complejidad técnica identificada.*    * ***Justificación****: Esto se hizo para evitar comprometer la calidad del código y permitir al equipo abordar los desafíos técnicos de manera más efectiva.* 3. ***Incorporación de feedback continuo****:*    * ***Ajuste****: Se implementó un proceso de revisión continua en lugar de esperar al final de cada sprint.*    * ***Justificación****: Esto permite realizar ajustes más rápidos y precisos en las funcionalidades, mejorando la adaptación a las necesidades cambiantes del usuario.*  ***Actividades eliminadas:***  1. ***Sesiones de revisión quincenales****:*    * ***Eliminada****: Se optó por no realizar sesiones de revisión quincenales, considerando que podrían interrumpir el flujo de trabajo.*    * ***Justificación****: En su lugar, se implementaron reuniones diarias que han resultado más efectivas para mantener la comunicación y abordar problemas de manera oportuna.*  ***Justificación de la estabilidad del plan de trabajo:*** *Si bien se han realizado ajustes, el plan de trabajo general no ha requerido cambios drásticos. Esto se debe a los siguientes* ***facilitadores****:*   * ***Colaboración efectiva****: La buena comunicación y el trabajo en equipo han permitido a los integrantes adaptarse a los cambios sin perder el enfoque en los objetivos.* * ***Uso de metodologías ágiles****: La flexibilidad inherente a las metodologías ágiles ha facilitado la adaptación a las necesidades del proyecto sin afectar significativamente el cronograma general.* * ***Clara definición de objetivos****: Tener objetivos bien definidos ha mantenido al equipo enfocado y alineado, minimizando la necesidad de cambios mayores en el plan de trabajo.* |
| --- |

| Actividades que no has iniciado o están retrasadas: ***Actividades no iniciadas o retrasadas:***  1. ***Pruebas de funcionalidad****:*    * ***Estado****: No iniciada*    * ***Motivos****: Retraso en la codificación de funcionalidades.*    * ***Estrategias****:*      + *Priorizar la finalización de la codificación y realizar pruebas unitarias en paralelo.* 2. ***Integración y despliegue****:*    * ***Estado****: No iniciada*    * ***Motivos****: Funcionalidades incompletas.*    * ***Estrategias****:*      + *Planificar la integración y establecer fechas límite para la finalización de la codificación.* 3. ***Análisis de datos****:*    * ***Estado****: No iniciada*    * ***Motivos****: Falta de datos debido a funcionalidades incompletas.*    * ***Estrategias****:*      + *Crear un conjunto de datos de prueba y colaborar con el equipo de desarrollo para identificar funcionalidades que puedan completarse rápidamente.*  ***Estrategias generales:***  * *Reevaluar el cronograma y ajustar según sea necesario.* * *Mantener reuniones de seguimiento para monitorear el progreso.* * *Redistribuir tareas entre el equipo para asegurar atención en actividades críticas.* |
| --- |

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)