

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

| Resumen de avance proyecto APT | En esta primera iteración del proyecto "Don Terapia", hemos logrado importantes avances en las funcionalidades clave del sistema. Se creó e implementó de forma efectiva nuestra base de datos Sql dentro del sistema con todas las tablas necesarias para el proyecto. Se completaron los módulos de registro y autenticación para los terapeutas y los pacientes , lo que permite que los   2 tipos de usuarios principales accedan de manera segura a la plataforma, además que dentro del desarrollo utilizamos el FrameWork de Django para la creación de un tercer usuario que sirva como admin y manipule nuestros 2 usuarios principales. También se ha trabajado intensamente en la seguridad de los datos, asegurando la implementación de la encriptación y conexiones seguras con la base de datos, lo que garantiza la confidencialidad de la información almacenada, todo esto implementado con la ayuda de Django que nos entrega la característica de tratamientos de usuarios y además la seguridad de estos mediante la encriptación evitando ataques de inyecta de SQL, Sin embargo, nos encontramos con un problema técnico en la implementación de la funcionalidad de creación de actividades personalizadas por parte de los terapeutas (HU-19). Este problema ha retrasado el desarrollo de dicha funcionalidad, pero hemos planificado resolverlo en la siguiente iteración. |
| --- | --- |
| Objetivos | *Opcional en caso de ajuste*  Objetivos alcanzados:   * Sistema de autenticación completo para terapeutas y pacientes. * Implementación de medidas de seguridad mediante encriptación de datos. * Registro de pacientes y almacenamiento de datos demográficos básicos. * Implementación del sistema CRUD (Create, read , update y delete) para la gestión de usuarios, donde los usuarios Admin en Django y Terapeuta pueden crear nuevos pacientes.   Objetivos pendientes:   * Resolver el problema con la creación de actividades personalizadas por parte de los terapeutas (HU-19). * Iniciar el desarrollo de la funcionalidad de análisis de datos mediante IA. |
| Metodología | *Opcional en caso de ajuste*  En este punto del uso de Metodología, no ha sido necesario realizar ajustes significativos en la metodología del proyecto, continuamos utilizando Extreme Programming (XP) como base para el desarrollo ágil del sistema. La metodología XP nos ha permitido trabajar en intervalos cortos, con ciclos de retroalimentación constantes, lo que facilita la adaptación a cualquier inconveniente técnico o cambio en los requisitos. Aunque algunos problemas técnicos han causado retrasos en la implementación de ciertas funcionalidades, la estructura de XP nos ha permitido planificar adecuadamente las iteraciones para corregir estos problemas en la siguiente fase. |
| Evidencias de avance | Evidencias de avance:   * Diagrama de Casos de Uso: Muestra las interacciones principales entre los usuarios (terapeutas y pacientes) y el sistema. Refleja cómo se han organizado las funcionalidades clave hasta el momento, como la autenticación y gestión de pacientes. * Capturas de Pantalla de la Interfaz: Imágenes que evidencian el desarrollo de la interfaz de usuario, incluyendo las pantallas de registro, autenticación, y gestión de pacientes. Estas muestran el avance en la creación de una interfaz intuitiva y funcional. * Script de Base de Datos: Demuestra cómo se ha estructurado la base de datos para almacenar y gestionar la información de usuarios, pacientes, y sesiones, garantizando la integridad de los datos. * Código Fuente: el código que implementa todos los sistemas que hemos realizado hasta el momento, que incluyen autenticación de usuarios, conexión con base de datos y asignación de actividades terapéuticas.   Justificación:  Estas evidencias permiten demostrar el avance en las funcionalidades principales del proyecto, como la autenticación de usuarios, la gestión de pacientes, y la interfaz de usuario. El diagrama y las capturas reflejan el progreso visual y funcional del sistema, mientras que el script de la base de datos y el código fuente muestran cómo se ha implementado la estructura técnica y de seguridad.  Resguardo de la Calidad del Proyecto APT:   * Metodología XP: La aplicación de iteraciones cortas y entregas continuas ha permitido validar y ajustar las funcionalidades del sistema de manera eficiente. * Seguridad de la Información: Se han implementado medidas de seguridad como la autenticación de usuarios y encriptación de datos, garantizando la protección de información sensible.   Todas estas evidencias por motivos de lectura no se encuentran en este archivo , se pueden encontrar en nuestro archivo OneDrive y repositorios de Github |
| **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 | Actividades | Descripción Actividades | Recursos | Duración de la actividad | Responsable | Observaciones | Estado de avance | | Ajustes |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Recopilación de requerimientos iniciales | Reunión inicial junto a nuestro cliente para recopilar los requisitos del sistema y generar los documentos necesarios que detallan los requerimientos y funcionalidades del sistema. | Laptop con conexión a internet y cuaderno con lápiz | 1 día | Felipe Vargas, Manuel Cartes | Facilitadores: Buena comunicación con el cliente puede agilizar la recopilación. Dificultades: Posibles malentendidos o cambios en los requisitos a mitad del desarrollo. | Completado | |  |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Creación de Historias de Usuario | Redacción de historias de usuario que describen las funcionalidades del sistema desde la perspectiva de los usuarios (terapeutas y pacientes, | Laptop con conexión a internet | 2 días | Jose Muñoz, Joaquin Brito | Facilitadores: Herramientas colaborativas pueden facilitar la creación en equipo. Dificultades: Malos entendidos en los requisitos puede generar historias ambiguas. | Completado | |  |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Creación del Backlog del Producto | Desarrollo de un backlog priorizado con todas las funcionalidades y mejoras que se deben implementar, según los requerimientos del cliente. | Herramientas de gestión de proyectos (Trello) | 4 días | Felipe Vargas, Jose Muñoz | Facilitadores: Priorizar correctamente las tareas facilita el desarrollo ágil. Dificultades: Repriorización constante debido a cambios de requerimientos del cliente. | Completado | |  |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Cartas CRC Iteración 1 | Diseño de clases utilizando Cartas CRC para definir la arquitectura del sistema en la primera iteración. | Herramientas de diseño de software (Lucidchart) | 5 días | Joaquin Brito, Manuel Cartes | Facilitadores: Claridad en las responsabilidades de las clases mejora el diseño. Dificultades: Falta de consenso en el equipo sobre el diseño puede retrasar la actividad. | Retrasada | |  |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Tareas Técnicas Iteración 1 | Asignación y desarrollo de tareas técnicas necesarias para implementar las funcionalidades de la iteración 1, que se definió dentro del Product Backlog donde se decidió que historias de usuario se trabajarán en esta primera iteración | Entorno de desarrollo (Visual Studio Code), Git , Python , Django, Servicio Cloud de base de datos Sql | 10 días | Felipe Vargas, Joaquin Brito | Facilitadores: Asignación clara de tareas mejora la eficiencia del equipo. Dificultades: Problemas técnicos pueden requerir más tiempo del planeado. | Completado | |  |
| Resolver las vulnerabilidades sistémicas para asegurar que el software construido cumple las normas de seguridad exigidas por la industria. | Pruebas Unitarias Iteración 1 | Desarrollo de pruebas unitarias para validar las funcionalidades desarrolladas en la iteración 1. | Entorno de pruebas | 5 días | Joaquin Brito, Manuel Cartes | Facilitadores: Herramientas automatizadas pueden acelerar las pruebas. Dificultades: Errores inesperados o bugs pueden surgir, necesitando correcciones adicionales. | Retrasada | |  |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Documento de Retroalimentación Iteración 1 | Recopilación de feedback del cliente sobre las funcionalidades entregadas en la iteración 1 y ajustes planificados para la siguiente iteración. | Herramientas de documentación (Word, Google Docs) | 5 días | Felipe Vargas, Jose Muñoz | Facilitadores: Reunión presencial o virtual puede mejorar la claridad del feedback. Dificultades: Feedback ambiguo o contradictorio puede generar confusión en los ajustes. | En curso | |  |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Small Release Iteración 1 | Entrega de una versión funcional del sistema con las funcionalidades completadas en la iteración 1, lista para pruebas del cliente y asi de esta forma testear su funcionamiento. | Herramientas de despliegue (Heroku, Docker) | 3 días | Felipe Vargas, Joaquin Brito | Facilitadores: Versionamiento controlado permite despliegues rápidos y seguros. Dificultades: Problemas de integración pueden retrasar la entrega. | En curso | |  |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Cartas CRC Iteración 2 | Diseño de clases utilizando Cartas CRC para definir la arquitectura del sistema en la segunda iteración, ajustando según el feedback recibido. | Herramientas de diseño de software (Lucidchart) | 5 días | Joaquin Brito, Manuel Cartes | Facilitadores: Claridad en las responsabilidades de las clases mejora el diseño. Dificultades: Falta de consenso en el equipo sobre el diseño puede retrasar la actividad. | No iniciado | |  |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Tareas Técnicas Iteración 2 | Desarrollo de las funcionalidades acordadas para la iteración 2 | Entorno de desarrollo (Visual Studio Code), Git y demas entornos de desarrollo si es necesario. | 4 días | Todo el equipo | Facilitadores: Asignación clara de tareas mejora la eficiencia del equipo. Dificultades: Problemas técnicos pueden requerir más tiempo del planeado. | No iniciado | |  |
| Resolver las vulnerabilidades sistémicas para asegurar que el software construido cumple las normas de seguridad exigidas por la industria. | Pruebas Unitarias Iteración 2 | Desarrollo de pruebas unitarias para validar las funcionalidades desarrolladas en la iteración 2 | Entorno de pruebas | 6 días | Joaquin Brito, Manuel Cartes | Facilitadores: Herramientas automatizadas pueden acelerar las pruebas. Dificultades: Errores inesperados o bugs pueden surgir, necesitando correcciones adicionales. | No iniciado | |  |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Documento de Retroalimentación Iteración 2 | Recopilación de feedback del cliente sobre las funcionalidades entregadas en la iteración 2 y ajustes planificados para la siguiente iteración | Herramientas de documentación (Word, Google Docs) | 3 días | Felipe Vargas, Jose Muñoz | Facilitadores: La familiaridad con el cliente puede facilitar el recibir feedback por parte de el cliente  Dificultades: Feedback ambiguo o contradictorio puede generar confusión en los ajustes. | No iniciado | |  |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Small Release Iteración 2 | Entrega de una nueva versión del sistema con las funcionalidades de la iteración 2 completadas y listas para pruebas del cliente. | Herramientas de despliegue | 3 días | Felipe Vargas, Jose Muñoz | Facilitadores: Versionamiento controlado permite despliegues rápidos y seguros. Dificultades: Problemas de integración pueden retrasar la entrega. | No iniciado | |  |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Cartas CRC Iteración 3 | Diseño de clases utilizando Cartas CRC para definir la arquitectura del sistema en la tercera iteración, basándose en los resultados de las iteraciones anteriores. | Herramientas de diseño de software (Lucidchart). | 4 días | Joaquin Brito, Manuel Cartes | Facilitadores: Claridad en las responsabilidades de las clases mejora el diseño. Dificultades: Falta de consenso en el equipo sobre el diseño puede retrasar la actividad. | No iniciado | |  |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Tareas Técnicas Iteración 3 | Desarrollo de nuevas funcionalidades para la iteración 3. | Entorno de desarrollo (Visual Studio Code), Git y demas entornos de desarrollo si es necesario. | 2 días | Todo el equipo | Facilitadores: Asignación clara de tareas mejora la eficiencia del equipo. Dificultades: Problemas técnicos pueden requerir más tiempo del planeado. | No iniciado | |  |
| Resolver las vulnerabilidades sistémicas para asegurar que el software construido cumple las normas de seguridad exigidas por la industria. | Pruebas Unitarias Iteración 3 | Pruebas de integración de las nuevas funcionalidades y ajustes necesarios para mantener la seguridad del sistema. | Entorno de pruebas. | 2 días | Joaquin Brito, Manuel Cartes | Facilitadores: Herramientas automatizadas pueden acelerar las pruebas. Dificultades: Errores inesperados o bugs pueden surgir, necesitando correcciones adicionales. | No iniciado | |  |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Documento de Retroalimentación Iteración 3 | Recopilación de feedback del cliente sobre las funcionalidades entregadas en la iteración 3 y ajustes planificados para la iteración final. | Herramientas de documentación (Word, Google Docs). | 2 días | Felipe Vargas, Jose Muñoz | Facilitadores: La familiaridad con el cliente puede facilitar el recibir feedback por parte de el cliente  Dificultades: Feedback ambiguo o contradictorio puede generar confusión en los ajustes. | No iniciado | |  |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Small Release Iteración 3 | Entrega de la versión del sistema con todas las funcionalidades de la iteración 3 listas para ser probadas por el cliente. | Herramientas de despliegue. | 1 día | Felipe Vargas, Joaquin Brito | Facilitadores: Versionamiento controlado permite despliegues rápidos y seguros. Dificultades: Problemas de integración pueden retrasar la entrega. | No iniciado | |  |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Cartas CRC Iteración 4 | Diseño de clases utilizando Cartas CRC para la última iteración, considerando todas las funcionalidades del sistema. | Herramientas de diseño de software (Lucidchart). | 2 días | Joaquin Brito, Manuel Cartes | Facilitadores: Claridad en las responsabilidades de las clases mejora el diseño. Dificultades: Falta de consenso en el equipo sobre el diseño puede retrasar la actividad. | No iniciado | |  |
| Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación. | Tareas Técnicas Iteración 4 | Implementación final de funcionalidades críticas y ajustes finales de la plataforma. | Entorno de desarrollo (Visual Studio Code), Git y demás entornos de desarrollo si es necesario. | 2 días | Todo el equipo | Facilitadores: Asignación clara de tareas mejora la eficiencia del equipo. Dificultades: Problemas técnicos pueden requerir más tiempo del planeado. | No iniciado | |  |
| Resolver las vulnerabilidades sistémicas para asegurar que el software construido cumple las normas de seguridad exigidas por la industria. | Pruebas Unitarias Iteración 4 | Pruebas de aceptación finales para garantizar la estabilidad, funcionalidad y seguridad del sistema antes de la entrega final al cliente | Entorno de pruebas. | 2 días | Joaquin Brito, Manuel Cartes | Facilitadores: Herramientas automatizadas pueden acelerar las pruebas. Dificultades: Errores inesperados o bugs pueden surgir, necesitando correcciones adicionales. | No iniciado | |  |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Documento de Retroalimentación Iteración 4 | Recopilación de feedback final del cliente sobre las funcionalidades entregadas en la iteración 4 y preparación de ajustes menores para la entrega definitiva. | Herramientas de documentación (Word, Google Docs). | 2 días | Felipe Vargas, Jose Muñoz | Facilitadores: La familiaridad con el cliente puede facilitar el recibir feedback por parte de el cliente  Dificultades: Feedback ambiguo o contradictorio puede generar confusión en los ajustes. | No iniciado | |  |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Small Release Iteración 4 | Entrega final del sistema al cliente con todas las funcionalidades completadas, documentadas y listas para ser usadas en un entorno real. | Herramientas de despliegue. | 1 día | Felipe Vargas, Joaquin Brito | Facilitadores: Versionamiento controlado permite despliegues rápidos y seguros. Dificultades: Problemas de integración pueden retrasar la entrega. | No iniciado | |  |
| Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización. | Preparación del Informe Final de Proyecto | Documento final que detalla todo el proceso del proyecto, incluyendo objetivos alcanzados, problemas enfrentados, soluciones implementadas y resultados obtenidos. | Herramientas de documentación. | 2 días | Todo el equipo | Facilitadores: Acceso a documentación completa y plantillas  agiliza la redacción. Dificultades: La falta de conocimientos del proyecto completo puede retrasar la entrega. | No iniciado | |  |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Presentación del Proyecto Final | Entrega y presentación final del proyecto al cliente, demostrando todas las funcionalidades y explicando el proceso de desarrollo, resultados y posibles mejoras futuras. | Herramientas de presentación (PowerPoint). | 2 días | Todo el equipo | Facilitadores: Dominio del software y ambiente de demostración controlado aseguran una presentación fluida. Dificultades: Equilibrar tecnicidad y claridad para el cliente puede ser desafiante. | 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. |

| En el desarrollo de mi Proyecto APT, varios factores han facilitado y dificultado el proceso.  Entre los factores que han facilitado el trabajo, destacó la buena comunicación dentro del equipo y la colaboración constante, lo que ha permitido una rápida resolución de problemas y un ambiente de trabajo positivo. Además, contar con recursos adecuados y un cronograma claro ha sido fundamental para mantenernos en el camino correcto.  Por otro lado, he enfrentado algunas dificultades, como la falta de puntualidad en las reuniones y algunos retrasos en la entrega de tareas. Para abordar estos problemas, planeó establecer horarios más estrictos y recordar al equipo la importancia de cumplir con los plazos. También promoverá un diálogo abierto para que todos puedan expresar sus preocupaciones y sugerencias.  En resumen, aunque hemos encontrado obstáculos, estamos trabajando activamente para superarlos y asegurar el éxito del proyecto. |
| --- |

| Actividades ajustadas o eliminadas:  Durante el desarrollo de mi plan de trabajo, realicé algunos ajustes a las actividades iniciales. Ajusté la duración de ciertas tareas para permitir un enfoque más profundo en las áreas que requerían más atención, lo que ha mejorado nuestra eficiencia. Además, eliminé algunas actividades que resultaron ser menos relevantes para los objetivos del proyecto, lo que nos permitió optimizar el tiempo y recursos disponibles. Estos cambios fueron necesarios para adaptarnos a las circunstancias del equipo y garantizar que todos estuviéramos alineados con las prioridades del proyecto. A pesar de estos ajustes, la estructura general del plan se mantuvo sólida gracias a la buena comunicación y el compromiso del equipo, lo que nos permitió avanzar de manera efectiva hacia nuestras metas. |
| --- |

| Actividades que no has iniciado o están retrasadas:  Actualmente, hay algunas actividades de nuestro plan que no hemos podido iniciar o que están retrasadas. Los motivos principales incluyen la falta de recursos necesarios y algunas dificultades en la coordinación del equipo, lo que ha afectado nuestra capacidad para cumplir con los plazos establecidos. Para avanzar en estas actividades y minimizar el impacto en el proyecto APT, estamos implementando varias estrategias. Primero, estamos priorizando la comunicación entre todos los miembros del equipo para asegurarnos de que estemos alineados y para resolver cualquier inconveniente que pueda surgir. Además, planeamos reorganizar el cronograma, estableciendo nuevas fechas límite realistas y asignando tareas de manera más eficiente. Al aplicar estas estrategias, confiamos en que podremos retomar el ritmo del proyecto y cumplir con nuestros objetivos a tiempo. |
| --- |