}

1. **PARTE I**

| **1. Antecedentes Personales** |
| --- |
| A continuación, se presenta una tabla en la que debes completar la información solicitada. |

| Nombre estudiante | **Felipe Vargas, Manuel Cartes, Jose Muñoz, Joaquin Brito** |
| --- | --- |
| Rut | **21.235.170-9, 20.431.906-5, 20.221.810-5, 20.246.380-0** |
| Carrera | **Ing. En informática** |
| Sede | **Antonio Varas** |

| **2. Descripción Proyecto APT** |
| --- |
| En la descripción debes señalar brevemente el nombre de tu proyecto APT y las competencias del perfil de egreso que vas a poner en práctica. Si en tu carrera están definidas las áreas de desempeño, también menciona a qué áreas de desempeño está vinculado el proyecto. |

| Nombre del proyecto | **“Don Terapia”** |
| --- | --- |
| Área (s) de desempeño(s) | En este proyecto se abordarán estas áreas de desempeño:  -Desarrollo de Software  -Gestión de Bases de Datos  -Gestión de Proyectos  -Análisis de Datos  -Machine Learning  -Seguridad Informática |
| Competencias | En este proyecto se abordarán estas competencias del perfil de egreso:  -Ofrecer propuestas de solución informática analizando de forma integral los procesos de acuerdo a los requerimientos de la organización.  -Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos.  -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.  -Resolver las vulnerabilidades sistémicas para asegurar que el software construido cumple las normas de seguridad exigidas por la industria.  -Desarrollar la transformación de grandes volúmenes de datos para la obtención de información y conocimiento de la organización a fin de apoyar la toma de decisiones y la mejora de los procesos de negocio, de acuerdo a las necesidades de la organización. |

| **3. Fundamentación Proyecto APT** |
| --- |
| A continuación, se presentan distintos campos que debes completar con la información solicitada. Esta sección busca que describas en detalle tu proyecto y justifiques su relevancia y pertinencia. |

| Relevancia del proyecto APT | El proyecto "Don Terapia" nace en Chile, más específicamente en la comuna de Providencia de nuestra mano junto a la ayuda de nuestro cliente “Fabian Opazo” y está diseñado para abordar la falta de herramientas sofisticadas y especializadas en la psicoterapia que permitan a los terapeutas gestionar de manera integral sus sesiones, actividades, y la relación con sus pacientes. Actualmente, los psicoterapeutas incluyendo a nuestro cliente enfrentan desafíos al intentar organizar y monitorear el progreso terapéutico de sus pacientes y esto a logrado que los clientes potenciales que desean participar de la psicoterapia estén reacios debido a que no existe forma de medir la efectividad de manera sencilla los resultados de la terapia, por lo cual para mejorar esto se requiere de un software robusto que facilite y optimice el proceso terapéutico, lo cual afectará en un inicio a psicoterapeuta y, pacientes de terapia además de potencialmente a instituciones de salud mental.  Este proyecto tiene mucha relevancia para el campo laboral debido a que aplica una integración de tecnología en el sector de la salud, además de la implementación de herramientas de inteligencia artificial. |
| --- | --- |
| Descripción del Proyecto APT | El proyecto APT tiene como objetivo desarrollar un sistema informático que optimice las sesiones terapéuticas, permitiendo a terapeutas y pacientes medir de manera efectiva el progreso de las sesiones, destacando hitos e información relevante. Para lograr esto, se creará una plataforma web centrada en dos tipos de usuarios principales: Terapeuta y Paciente, donde  terapeuta podrá: -Gestionar pacientes, creará y evaluará notas terapéuticas, y asignará actividades, además existira una ia en el sistema que servirá para analizar tanto las notas subidas por el terapeuta como las actividades realizadas por el paciente, generando una evaluación automática que será evaluada por el terapeuta logrando aceptar o corregir la evaluación generada por ia según los objetivos específicos de cada paciente, asegurando que el progreso esté alineado con el tratamiento.  Paciente podrá: -subir actividades que el terapeuta les asignó para realizar en diferentes fuentes , como pueden ser por ejemplo, textos o fotos, además el paciente podrá revisar cuando el lo desee el progreso de las sesiones terapéuticas y de sus actividades para ver la efectividad de las mismas , además de ver los hitos y logros que a conseguido en cada una de estas. |
| Pertinencia del proyecto con el perfil de egreso | El proyecto APT se alinea con el perfil de egreso de mi carrera, ya que requiere competencias clave como la creación de soluciones informáticas, desarrollo de software, y manejo de datos, aplicadas al contexto terapéutico. La plataforma que desarrollaremos optimiza el proceso terapéutico al permitir una gestión y evaluación integral de pacientes y actividades, apoyada por inteligencia artificial para evaluar el progreso de cada paciente. Esto exige tanto un análisis integral como la implementación de software seguro y eficaz, acorde tanto a los requerimientos de la industria como los requerimientos resultantes de la primera reunión con nuestro cliente, asegurando la efectividad en las sesiones terapéuticas. Todas estas competencias dadas por nuestra carrera son necesarias para que nos permitan tanto la creación de la solución detrás del proyecto como para ejecutar esta solución y crear el sistema que cubra todos los requerimientos. |
| Relación con los intereses profesionales | Felipe Vargas: El Proyecto APT se relaciona con mis intereses profesionales debido a que dentro de mis intereses profesionales se encuentran la gestión de proyectos tecnológicos, análisis de datos y soluciones innovadoras en salud. También me permite aplicar mis fortalezas en gestión y análisis, mientras desarrollo habilidades en programación que es una de mis debilidades, además de seguridad informática e implementación de IA. Este proyecto contribuirá a mi desarrollo profesional al integrar y expandir mis conocimientos en estas áreas clave, especialmente en un contexto desafiante y significativo como el de la salud, además de la posibilidad de aprender mis habilidades obtenidas dentro de mi especialidad que es el área de ciencia de datos.  Manuel Cartes: Don Terapia es un proyecto que está directamente relacionado con mis intereses profesionales, ya que consta de varias aristas de la informática que me intrigan y me generan satisfacción. Una de ellas es el liderazgo de proyectos informáticos junto con la gestión de proyectos y riesgos. Existe también el modelamiento e implementación de bases de datos que es algo que me gustaría mejorar para futuras oportunidades. Siento además que el proyecto me ayudará a mejorar algunos aspectos en los cuales necesito más experiencia, como la programación de algoritmos complejos y la implementación de machine learning en un sistema real y funcional. Por último, el proyecto tiene un potencial real de ser implementado en el área de la salud mental y poder generar un cambio positivo para la mejor organización y utilización de software sofisticado para la psicoterapia, algo que me motiva bastante al tener la oportunidad de ofrecer soluciones tangibles que mejoren la calidad de vida de las personas.  Juan Nuñez: El proyecto de Don Terapia se alinea perfectamente con mis intereses profesionales en arquitectura de datos y gestión de bases de datos. Abarca áreas clave como el diseño y la optimización de infraestructuras de datos, y la administración eficiente de bases de datos, aspectos que deseo perfeccionar para futuras oportunidades. Además, el proyecto me permitirá aplicar tecnologías como machine learning en sistemas reales, con un impacto potencial en sectores como la salud mental. Esto me motiva, ya que puedo ofrecer soluciones tangibles que mejoren la calidad de vida de las personas.  Joaquin Brito: El Proyecto APT está directamente relacionado con mis intereses profesionales en ciencia de datos y machine learning. Me brinda la oportunidad de aplicar mis habilidades en análisis de datos y gestión de proyectos tecnológicos, mientras desarrollo competencias en programación y seguridad informática. Además, el proyecto me permitirá profundizar en la implementación de IA, integrando y expandiendo mis conocimientos en estas áreas clave. Trabajar en un contexto tan significativo como el de la salud me motiva a seguir desarrollándome en soluciones innovadoras que aprovechen al máximo el potencial del machine learning. |
| Factibilidad de desarrollo del Proyecto APT | Nuestro Proyecto APT es posible dentro del semestre de alrededor de 16 semanas gracias a varios factores clave, como que nosotros asignamos alrededor de 3 horas diarias asignadas al proyecto, lo que proporciona tiempo suficiente para completar las tareas programadas, aunque el tiempo real puede variar debido a otras responsabilidades. Utilizaremos herramientas y recursos gratuitos como GitHub para el versionamiento, Python para el backend, y la API de GPT, con un costo estimado de hasta 20 dólares para pruebas iniciales. Además, emplearemos plataformas gratuitas para el desarrollo web y la base de datos.  Entre los factores que facilitan el desarrollo están la disponibilidad de estas herramientas y el tiempo asignado. Sin embargo, también enfrentamos posibles desafíos como problemas técnicos con la integración de la IA y limitaciones de tiempo. Para abordar estos retos, planificamos pruebas y ajustes continuos, priorizando las tareas críticas para asegurar que el proyecto avance según lo previsto todo esto posible por nuestra metodología ágil utilizada que es programación extrema. |

1. **PARTE II**

| **4. Objetivos** |
| --- |
| En este apartado debes definir objetivos generales y específicos del Proyecto APT. Es importante aclarar que los objetivos se deben plantear en forma clara, concisa y sin dar mayores explicaciones, es decir, deben entenderse por sí solos. Se sugiere redactarlos utilizando un verbo en infinitivo, pues ello obliga a precisar acciones concretas. |

| Objetivo general | Desarrollar una plataforma web para optimizar las sesiones terapéuticas, permitiendo a terapeutas gestionar, evaluar y medir el progreso de sus pacientes de manera efectiva mediante el uso de inteligencia artificial y permitirles a los pacientes entender la efectividad de la terapia mediante hitos y logros destacados por la inteligencia artificial . |
| --- | --- |
| Objetivos específicos | Alguno de los objetivos específicos que se rescatan del objetivo general serían:  -Diseñar e implementar la arquitectura del sistema para la plataforma web, incluyendo la gestión de usuarios, base de datos , notas terapéuticas y actividades.  -Integración de la API de GPT para proporcionar análisis automatizado de las notas y actividades, y generar evaluaciones precisas que podrán ser aceptadas y guardadas en la base de datos para futuros análisis de progreso.  -Desarrollar la interfaz de usuario para facilitar la interacción entre terapeutas y pacientes, permitiendo la carga y revisión de actividades y evaluaciones además de visualización de datos de una forma sencilla para los pacientes.  -Asegurar la protección de datos sensibles mediante la implementación de medidas de seguridad adecuadas.  -Realizar pruebas continuas del sistema para identificar y corregir errores en conjunto con nuestro cliente y algunos pacientes seleccionados, asegurando su funcionalidad y rendimiento óptimos. |

| **5. Metodología** |
| --- |
| En el siguiente apartado deberás describir la metodología, propia de tu disciplina, que utilizarás para resolver el proyecto APT antes descrito, incluyendo las etapas y métodos de trabajo. |

| Descripción de la Metodología |
| --- |
| Para abordar el desarrollo del Proyecto APT, utilizaremos la metodología de Extreme Programming o XP, la cual es ideal para proyectos de pocos integrantes como el nuestro y que requieren adaptabilidad y rapidez en la implementación de cambios. Esta metodología se caracteriza por fomentar la colaboración estrecha con el cliente que en nuestro caso es extremadamente necesaria,además de la entrega frecuente de versiones funcionales del software.  Para este proyecto donde estamos trabajando utilizando la metodología de Extreme Programming es necesario definir no solo las actividades necesarias para el desarrollo del proyecto sino que también las necesarias para la implementación de esta metodología.  1- User Stories (Historias de Usuario)   * Aquí definimos las funcionalidades que debe tener la plataforma según las necesidades del cliente, que incluye tareas como: Reunir requisitos del cliente (Fabian Opazo) y escribir historias de usuario que reflejen los requerimientos tomados en las reuniones con el cliente. (Responsable: Manuel Cartes, Jose Muñoz)   2- Release Planning (Planificación de Lanzamiento)   * Aquí planificamos cómo y cuándo se entregarán las funcionalidades de la plataforma que se definieron anteriormente, que incluye tareas como: Estimar el esfuerzo y tiempo de desarrollo , otra funcionalidad es Priorizar historias de usuario. (Responsable: Felipe Vargas,Jose Muñoz)   3- Spike (Investigación y Pruebas)   * Aquí realizamos pruebas exploratorias para resolver dudas técnicas antes de implementar características complejas, que incluyen en esta fase tareas como: Realizar pruebas de viabilidad técnica y documentar los resultados obtenidos. (Responsable: Felipe Vargas, Joaquin Brito)   4- Iteration (Iteración)   * Aquí desarrollamos el software en ciclos cortos y repetitivos, cada uno resultando en una versión funcional del sistema tomando en cuenta las fases anteriores, y esta fase incluye tareas como: Programar las funcionalidades que creamos en la fase anterior , Desarrollar la interfaz de usuario y implementar APIS en caso de ser requeridas. (Responsable: Felipe Vargas y Manuel Cartes, Jose Muñoz, Joaquin Brito)   5- Acceptance (Aceptación)   * Aquí validamos el software creado en la fase anterior con el cliente para asegurarnos de que cumple con sus expectativas además de que efectivamente cumpla con las necesidades que tenga el cliente, y en esta fase incluye tareas como: Realizar pruebas de aceptación con el cliente en una reunion y Implementar correcciones según el feedback recibido en esa reunión. (Responsable: Manuel Cartes, Jose Muñoz, Joaquin Brito, Felipe Vargas)   6- Small Releases (Pequeños Lanzamientos)   * Aquí le entregamos al cliente una versión del software que se aprobaron con el cliente en la fase anterior destinado para su implementación con algunos clientes para medir su efectividad y recibir retroalimentación, y en esta fase incluye tareas como: Preparar lanzamientos funcionales para producción usando el feedback anterior , además de documentar y capacitar al cliente en el uso de la plataforma. (Responsable: Felipe Vargas y Manuel Cartes) |

| **6. Evidencias** |
| --- |
| A continuación, describe qué evidencias serán evaluadas en el informe de avance y en el informe final de tu proyecto APT. Estas evidencias deben ser acordadas con tu docente. Se entenderá por evidencia los productos que se desarrollen durante el proyecto y cuyo propósito sea visibilizar o documentar cómo se ha implementado el trabajo. |

| Tipo de evidencia  (avance o final) | Nombre de la evidencia | Descripción | Justificación |
| --- | --- | --- | --- |
| Avance | Documentos de  Requerimientos Iniciales | Documento que detalla las necesidades y expectativas iniciales del cliente (Fabian Opazo) para la plataforma de terapia. | Proporciona una base clara para el desarrollo y permite evaluar si el proyecto cumple con los objetivos iniciales. |
| Avance | Historias de  Usuario | Conjunto de descripciones cortas de funcionalidades desde la perspectiva del usuario final (terapeuta y paciente). | Ayuda a mantener el enfoque en las necesidades del usuario y facilita la planificación de iteraciones. |
| Avance | Backlog del  Producto | Lista priorizada de todas las funcionalidades deseadas para el producto(Extraídas de los requerimientos). | Permite visualizar el alcance total del proyecto y facilita la planificación de releases. |
| Avance | Release Plan  (Planificación de  Lanzamiento en  cada iteración) | Planificación detallada de qué funcionalidades se implementarán en cada iteración, incluyendo las fechas de cada iteración. | Proporciona una hoja de ruta clara para el desarrollo y ayuda a medir el progreso. |
| Avance | Cartas CRC  Iteración X | Tarjetas que describen las Clases, Responsabilidades y Colaboradores para el diseño orientado a objetos de la iteración actual. | Facilita el diseño del sistema y la comunicación entre el equipo sobre la estructura del software. |
| Avance | Tareas Técnicas  Iteración X | Lista de tareas específicas de programación para implementar las historias de usuario de la iteración actual. | Permite un seguimiento detallado del progreso técnico y facilita la distribución del trabajo. |
| Avance | Pruebas Unitarias  Iteración X | Conjunto de pruebas automatizadas para verificar el funcionamiento correcto de componentes individuales del sistema. | Asegura la calidad del código y facilita la refactorización continua. |
| Avance | Documento de  Retroalimentación Iteración X | Resumen de la retroalimentación del cliente después de cada demostración de la iteración. | Permite ajustar el desarrollo basado en las necesidades cambiantes del cliente y mejora la satisfacción del usuario final. |
| Avance | Small Release  Iteración X | Versión funcional del software que implementará el cliente para entregarnos su FeedBack. | Permite la validación temprana con usuarios reales. |
| Final | Manual de  Usuario | Documento que explica cómo usar la plataforma, tanto para terapeutas como para pacientes. | Facilita la adopción del sistema por parte de los usuarios finales y reduce la necesidad de soporte. |
| Final | Informe Final de Proyecto | Documento que resume todo el proceso de desarrollo, incluyendo desafíos, soluciones implementadas y resultados obtenidos. | Proporciona una visión completa del proyecto, útil para futuras referencias y para demostrar el valor entregado al cliente. |

| 7. Plan de Trabajo |
| --- |
| En la siguiente tabla define la planificación de tu Proyecto APT de acuerdo a lo requerido. |

| Plan de Trabajo Proyecto APT | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable | Observaciones |
| 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. |
| 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. |
| 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. |
| Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos. | Planificación de Release (Release Plan) | Creación de un plan detallado de lanzamientos iterativos, definiendo qué funcionalidades estarán disponibles en cada iteración y sus fechas de entrega. | Herramientas de gestión de proyectos (Jira) | 3 días | Felipe Vargas, Jose Muñoz | Facilitadores: Herramientas de planificación visual permiten ajustes rápidos. Dificultades: Dependencias entre tareas pueden afectar el cronograma. |
| 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. |
| 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, seleccionadas anteriormente en el Release Plan. | Entorno de desarrollo (Visual Studio Code), Git y demás entornos de desarrollo si es necesario. | 5 días | Todo el equipo | Facilitadores: Asignación clara de tareas mejora la eficiencia del equipo. Dificultades: Problemas técnicos pueden requerir más tiempo del planeado. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |

| **8. Carta Gantt** |
| --- |
| **Busca un formato de Carta Gantt que te acomode y organiza en este las actividades planificadas en el punto anterior considerando el periodo asignado para el desarrollo de tu Proyecto APT. Debes mantener la temporalidad del periodo académico en el desarrollo de las tres fases que contempla la Asignatura de Portafolio de Título.** |

