

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

| Resumen de avance proyecto APT | Se termina el sprint 0 y sprint 1 junto con avanzar con las actividades en el sprint 2 donde por el momento se han realizado los documentos del sprint como product backlog, sprint backlog y burdowchart, además de avanzar en el análisis exploratorio y en el entrenamiento de modelos, aunque se tuvieron que eliminar algunas tareas por falta de tiempo. |
| --- | --- |
| Objetivos | El objetivo principal del proyecto "Optimizador de Cobranza" es desarrollar un modelo predictivo que identifique la acción mínima eficiente de cobranza para maximizar la recuperación de deudas, minimizando los costos operativos y mejorando la relación con los clientes. Se espera que, mediante el uso de técnicas avanzadas de machine learning, como K-Means para segmentación de clientes y LSTM para predicción de comportamientos de pago, las empresas puedan implementar estrategias de cobranza más efectivas y personalizada*s.* |
| Metodología | Para el desarrollo del Proyecto APT "Optimizador de Cobranza," utilizaremos la metodología ágil Scrum, que es ampliamente reconocida por su flexibilidad y capacidad para adaptarse a cambios, lo que resulta ideal para proyectos de desarrollo de software y ciencia de datos.  Etapas de Trabajo:   * Planificación del Proyecto (Sprint 0): La primera etapa se enfocará en la planificación estratégica del proyecto. Durante este sprint inicial, se definirán los objetivos claros del proyecto, las funcionalidades a desarrollar, los entregables esperados, y se realizará una estimación del tiempo necesario para cada tarea. * Desarrollo de funcionalidades principales (Sprint 1 ): En esta etapa, el equipo comenzará a desarrollar las funcionalidades principales del optimizador de cobranza, como la creación de interfaces, algoritmos básicos, y la infraestructura de backend. Se enfocará en la implementación de los aspectos funcionales que soportan el modelo predictivo en fases posteriores. * Desarrollo del modelo (Sprint 2): Esta fase estará enfocada en la creación y entrenamiento del modelo predictivo de cobranza. Se utilizarán técnicas de machine learning para predecir los comportamientos de pago de los clientes, optimizar las acciones de cobranza y mejorar las tasas de recuperación. * Testeo y pruebas de seguridad (Sprint 3 ): En esta etapa, el enfoque estará en la validación del modelo y las pruebas de seguridad del sistema. Se llevará a cabo un análisis exhaustivo para identificar posibles errores en el rendimiento del modelo o en la integridad de los datos, y se implementarán mecanismos para asegurar la fiabilidad y protección de la información sensible. * Marcha Blanca: La marcha blanca consiste en implementar el proyecto en un entorno controlado antes de su lanzamiento definitivo. Durante este período, el modelo y el sistema serán sometidos a pruebas reales con datos del cliente, con el objetivo de identificar y corregir cualquier falla o inconveniente antes de la entrega final. |
| Evidencias de avance | [**https://github.com/Martin3125/alloxentric\_vue**](https://github.com/Martin3125/alloxentric_vue)   1. Repositorio de Código Fuente: El repositorio disponible en [GitHub](https://github.com/Martin3125/alloxentric_vue) contiene el código fuente actualizado del proyecto, que incluye los avances en el desarrollo de la plataforma de optimización de cobranzas. Esta evidencia muestra el trabajo realizado en la implementación de funcionalidades clave como la carga de datos de deudores, la segmentación de clientes, la automatización de acciones de cobranza, y las interfaces de usuario. 2. Estructura del Proyecto: Se destaca la correcta aplicación de metodologías ágiles y prácticas de desarrollo seguro, que incluyen el uso de control de versiones (Git), el seguimiento de avances y pendientes mediante issues, y la aplicación de revisiones de código para garantizar la calidad y estabilidad del proyecto. Además, se ha implementado una arquitectura robusta que permite la escalabilidad del sistema, asegurando que las funciones implementadas se ajusten a los requisitos del proyecto. 3. Automatización de Pruebas y Calidad de Código: Se han configurado pruebas automáticas y validaciones de calidad del código con herramientas de integración continua, lo que permite asegurar que el código cumple con los estándares esperados antes de su implementación definitiva. Esto refuerza la confiabilidad del sistema al detectar posibles errores de manera temprana. 4. Aplicación de Metodologías Propias de la Disciplina: Se han utilizado enfoques disciplinarios como CRISP-DM para la segmentación y análisis de deudores, así como algoritmos de machine learning para la predicción de comportamientos de pago. Además, el uso de herramientas como Vue.js y Node.js refleja la correcta selección de tecnologías modernas para el desarrollo del front-end y back-end del proyecto. |

| **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 Fase 2:**

[Plan de trabajo FASE 2](https://docs.google.com/spreadsheets/d/1ywuqq33kcv98bkbBULpeUhbBemDjk7hgK2lwdjJ5nLw/edit?usp=drive_link)

| **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. |

| Los factores que han dificultado el desarrollo del plan de trabajo han sido los tiempos acotados que se tienen para realizar las actividades donde queda poco margen para corregir o mejorar los modelos durante el periodo establecido.  Otro problema que surgió fue en el equipo debido que durante la gestión del proyecto se tuvo que agregar a un nuevo integrante al equipo de trabajo por obligación donde nosotros como equipo lo aceptamos pero surgió un problema con el integrante donde en un inicio si cumplió con las tareas que se le asignaron, pero después debía cumplir una tarea simple que consiste en la corrección de subir un archivo al procesamiento y que se visualice el nombre, donde no realizó dicha tarea. Se le preguntó vía Whatsapp si estaba haciendo la tarea respondió que sí pero no se evidenciaba en GitHub la realización de la tarea se intentó volver a comunicarme con el integrante pero no respondió ningún mensaje. |
| --- |

| Actividades ajustadas o eliminadas:  El motivo principal por el cual se ajustaron estas tareas fue debido a que los tiempos eran cortos para el desarrollo y tomó más tiempo de lo esperado.  Las actividades que más se han ajustado en base a los tiempos son:   * Entrenamiento del Modelo 1 (ARTEFACTO)(K-means) * Entrenamiento del Modelo 2 (ARTEFACTO)(LSTM) * Validación de los Modelos (ARTEFACTO) * Optimización de los Modelos (ARTEFACTO) * Pruebas Unitarias, integración y aceptación * Documentación del Modelo * Evidencias del Sprint 2 * Revisión y Retroalimentación Sprint 2 * Reunión de retrospectiva   Eliminadas:   * Validación de los Modelos (ARTEFACTO) * Optimización de los Modelos (ARTEFACTO) |
| --- |

| Actividades que no has iniciado o están retrasadas:  En un inicio la tarea del modelo LSTM tenía un retraso debido a que los tiempos eran muy cortos y además que el modelo LSTM estaba mal construido ya que se intentaba pasar una solo variable al modelo y se intenta predecir múltiples variables. Se tuvo que volver a realizar el modelo LSTM pero esta vez se le pasaron múltiples variables para poder hacer la predicción de manera correcta. Aún así se logró terminar a tiempo esta actividad. |
| --- |