

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

| Resumen de avance proyecto APT | Hasta el momento hemos avanzado ágilmente gracias a la metodología Scrum, de nuestros 3 Sprint, estamos en el segundo, el cual abarca las siguientes historias de usuario:   * H1 Registro de usuario * H2 Inicio de Sesión * H3 Recuperación de Contraseña * H7 Gestión de Perfil de Usuario * H8 Procesamiento de Pagos * H5 Confirmación de reservas * H4 Búsqueda de disponibilidad * H9 Notificaciones y Recordatorios * H13 Modificación de Reservas (Admin) * H14 Cancelación de Reservas (Admin)   Nos queda la presentación del segundo Sprint para luego poder empezar con el tercero y terminar con el proyecto de manera exitosa. |
| --- | --- |
| Objetivos | Objetivo general:  Agilizar la gestión de Tinajas Rustic, en otras palabras, mejorar la eficiencia y efectividad de los procesos operativos, administrativos y logísticos de Tinajas Rustic, reduciendo tiempos, optimizando recursos y aumentando la capacidad de respuesta para ofrecer un mejor servicio al cliente y mejorar la rentabilidad del negocio.  Objetivos específicos:  Fidelizar a los clientes, en otras palabras, Lograr incrementar la tasa de retención de clientes en un 15% para el final del año, mediante la introducción de un programa de lealtad y seguimiento postventa. |
| Metodología | Para el proyecto, adoptaremos la metodología ágil Scrum, la cual nos permitirá una gestión eficiente y una colaboración continua para el desarrollo de software. Esta metodología divide el trabajo en iteraciones cortas y llamadas sprints, que nos permitirán adaptarnos a los cambios y entregar incrementos del producto de manera continua. |
| Evidencias de avance | En el informe de avance, presentaremos varias evidencias que documentan el progreso del proyecto, enfocándonos en aspectos claves de la gestión, desarrollo y aseguramiento de la calidad, según los principios de la metodología Scrum. Estas evidencias no solo muestran el trabajo realizado, sino también cómo hemos aplicado correctamente las herramientas y técnicas propias de nuestra disciplina para garantizar la calidad del proyecto. A continuación, detallo las principales evidencias que incluimos y su justificación:   1. Arquitectura del Sistema 2. Gestión de Riesgos 3. Gestión de Recursos 4. Gestión de la Comunicación 5. Cronograma del Proyecto 6. Historias de Usuario 7. Product Backlog and Sprint Backlog 8. Documentación de Requerimientos 9. Gestión de Calidad   Para asegurarnos de la calidad del proyecto, hemos aplicado la metodología Scrum en todas las fases de desarrollo. Esto implica la adecuada organización, ejecución y evaluación de cada sprint, garantizando el cumplimiento de los requisitos y el seguimiento de los principios ágiles por parte del equipo. También, se han empleado plataformas como GitHub para supervisar la versión del código, garantizando un proceso de desarrollo eficaz y colaborativo. Las evaluaciones constantes y las pruebas automáticas garantizan que el producto cumple con los estándares de calidad previstos.  Mediante el siguiente link : [Documentación Proyecto](https://drive.google.com/drive/folders/1BnDPxN3MsOUhR1KCA0afBjiPxZYCGibx?usp=sharing)  confirmamos que el desarrollo del proyecto sigue el plan establecido, aplicando métodos efectivos de ingeniería y gestión de proyectos para garantizar su éxito. |
| **2. Monitoreo del Plan de Trabajo** |

| Unidades de Competencias | Nombre de Actividades/Tareas | Descripción Actividades/ Tareas | Recursos | Duración de la Actividad | Responsable | Observaciones |
| --- | --- | --- | --- | --- | --- | --- |
| Programación y desarrollo web | Desarrollo del formulario de registro de usuarios | Crear y validar el formulario de registro de usuario en la plataforma web | VS Code, Servidor Web,  Base de datos. | 10 días | Vicente Vallejos, Francisca Sepulveda y  Matias Gatica | Validar que el formulario sea compatible con móviles |
| Gestión de base de datos | Implementación del módulo de confirmación de reservas | Desarrollar la lógica de confirmación de reservas, generando un comprobante para el usuario | VS Code,Servidor Backend, Base de datos. | 12 días | Vicente Vallejos, Francisca Sepulveda y  Matias Gatica | Asegurar de que el comprobante se envíe por correo electrónico |
| Gestión de Pasarelas de pago | Integración de sistema de pagos | Integrar una pasarela de pago para procesar pagos de reservas (transbank) | Pasarela de pago, Base de Datos | 11 días | Vicente Vallejos, Francisca Sepulveda y  Matias Gatica | Validar la correcta implementación de la pasarela de pago y prueba con transacciones |
| Desarrollo de aplicaciones móviles | Creación de interfaz para la gestión de reservas admin | Creación la sección dónde administrador puede ver y modificar reservas desde la app móvil | Ionic, Angular, Base de Datos | 10 días | Vicente Vallejos, Francisca Sepulveda y  Matias Gatica | Validar el diseño responsivo y facilidad de uso en móviles |
| Implementación de notificaciones | Configuración de notificaciones y recordatorios | Programar el sistema de notificaciones que envía recordatorios de reservas a los clientes | Servidor backend, API de email | 9 días | Vicente Vallejos, Francisca Sepulveda y  Matias Gatica | Verificar la entrega oportuna de notificaciones y pruebas en distintos dispositivos |

| **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 el desarrollo de mi plan de trabajo:   * Uso de Firebase: La integración con Firebase ha sido un factor clave que ha facilitado el desarrollo del sistema. Firebase proporciona una solución completa para la autenticación, almacenamiento de datos y notificaciones, lo cual ha simplificado varios aspectos técnicos, evitando la necesidad de implementar estos servicios desde cero. * Uso de Angular e Ionic: El uso de Angular e Ionic ha permitido una buena estructuración del proyecto, sobre todo en el frontend, facilitando la creación de una interfaz responsiva tanto para la versión web como móvil. Esto ha acelerado el desarrollo de las funcionalidades visuales, como el sistema de reservas. * Integración con Transbank: La implementación del procesamiento de pagos con Transbank ha sido bastante exitosa, lo que ha permitido implementar transacciones seguras para los usuarios de manera eficiente, mejorando la confianza en el sistema. * Equipo de Trabajo y Planificación: La planificación del proyecto y la comunicación clara dentro del equipo ha facilitado la ejecución de tareas, ya que los roles están bien definidos y los entregables se han cumplido dentro de los plazos estimados. Esto ha sido un facilitador importante para avanzar según lo planeado.   Factores que han dificultado el desarrollo de mi plan de trabajo:   * Conexión Firebase y MongoDB (dificultad inicial): Uno de los mayores desafíos fue la imposibilidad de conectar Firebase y MongoDB para gestionar datos. Inicialmente se intentó usar ambas tecnologías, pero al no poder resolver la integración, se decidió continuar únicamente con Firebase. Esta decisión simplificó la arquitectura del sistema y permitió concentrar esfuerzos en un solo backend. * Errores en la gestión de formularios en Angular: Al implementar formularios en Angular con ReactiveForms, hubo dificultades relacionadas con validaciones de datos y errores de sincronización. Esto ralentizó el progreso en algunas funcionalidades del frontend, como el registro de usuarios y la gestión del perfil. * Retrasos en el envío de correos de recuperación: Al implementar la funcionalidad de recuperación de contraseñas, se presentaron algunos retrasos en el envío de correos desde Firebase. Este fue un desafío crítico, ya que afectaba directamente la experiencia del usuario. * Problemas con Husky en Windows: Durante el proceso de configuración de Husky en un entorno Windows, surgieron errores relacionados con la falta de bash y comandos obsoletos, lo que dificulta la implementación de hooks para el control de versiones. * Optimización de la búsqueda de disponibilidad: La búsqueda de disponibilidad para las reservas fue otro reto debido a la complejidad de las consultas y la cantidad de datos que deben gestionarse en tiempo real. Esto generó tiempos de respuesta más lentos de lo esperado. |
| --- |

| Actividades ajustadas  Cambio de Firebase y MongoDB a solo Firebase: Inicialmente, el plan contemplaba utilizar Firebase junto con MongoDB para el manejo de la base de datos. Sin embargo, debido a problemas de conexión y la complejidad de integrar ambas plataformas, se decidió utilizar solo Firebase.  Justificación: La decisión de trabajar solo con Firebase simplificó el manejo de datos, ya que Firebase proporciona todas las herramientas necesarias para la autenticación, almacenamiento y notificaciones. Esto permitió enfocarse en una arquitectura más simple y eficiente, reduciendo la complejidad del desarrollo.  Optimización del sistema de formularios en Angular: El desarrollo de los formularios en Angular con ReactiveForms presentó errores de validación y sincronización de datos, lo que generó la necesidad de ajustar la lógica y las validaciones del frontend.  Justificación: Fue necesario ajustar la lógica de los formularios para garantizar que las validaciones fueran precisas y que los datos ingresados por los usuarios se almacenarán correctamente. Esto se realizó para mejorar la experiencia de usuario y evitar errores en la gestión de usuarios.  Revisión y optimización de la funcionalidad de recuperación de contraseña: Durante el desarrollo, se identificaron retrasos en el envío de correos de recuperación de contraseña, lo cual requería ajustes en la configuración del sistema de notificaciones.  Justificación: Este ajuste era necesario para asegurar que los correos de recuperación llegaran de manera oportuna, mejorando así la usabilidad y la confianza de los usuarios en el sistema. Se trabajó con SendGrid y Twilio para mejorar los tiempos de respuesta.  Reconfiguración de Husky en Windows: La configuración inicial de Husky en el entorno Windows presentó problemas técnicos relacionados con la falta de bash y scripts obsoletos. Se ajustaron los scripts para que funcionaran correctamente en este entorno.  Justificación: Estos ajustes fueron necesarios para implementar correctamente los hooks de control de versiones, permitiendo que el equipo de desarrollo mantuviera buenas prácticas en el uso de Git.  Optimización de la búsqueda de disponibilidad: Se identificaron problemas de rendimiento en la funcionalidad de búsqueda de disponibilidad, lo que llevó a ajustar las consultas a la base de datos para mejorar los tiempos de respuesta.  Justificación: Se ajustó esta funcionalidad para que los usuarios pudieran consultar las fechas disponibles de manera más eficiente y sin largos tiempos de espera. Este ajuste es clave para mantener una experiencia de usuario fluida.  Actividades eliminadas  Eliminación de la implementación de MongoDB: La idea inicial de combinar MongoDB con Firebase fue eliminada del plan de trabajo debido a los problemas de integración y a la redundancia que esto implicaba.  Justificación: Firebase por sí solo ofrece todas las herramientas necesarias para la autenticación, almacenamiento y manejo de datos en tiempo real, por lo que el uso de MongoDB se volvió innecesario. Esto también ayudó a reducir la complejidad y los costos operativos. |
| --- |

| Actividades que no has iniciado o están retrasadas    Hasta ahora, no hay ninguna tarea no iniciada, en fechas que no correspondan, ni retrasada en nuestro proyecto. Gracias a la estrategia de trabajo simultáneo y la coordinación constante entre los miembros del equipo, se ha logrado esta hazaña. El gráfico de Burndown muestra de manera clara el avance continuó, demostrando una disminución constante en las horas de trabajo que faltan y en las tareas que aún deben realizar. Este enfoque ha permitido cumplir con los plazos establecidos y asegurar el avance eficiente de las actividades planificadas, manteniendo el proyecto alineado con los objetivos marcados. |
| --- |