

| **1. Informe final Proyecto APT** |
| --- |
| El objetivo de este informe es que describas los aspectos más relevantes de tu Proyecto APT. Es importante que fundamentes las decisiones que tuviste que tomar a lo largo del proceso.  A continuación, encontrarás distintos campos que deberás completar con la información solicitada, los que dan cuenta del resumen de tu proyecto APT y sus principales resultados. |

| Nombre del proyecto | Sistema Unidad territorial |
| --- | --- |
| Área (s) de desempeño(s) | El proyecto ha permitido abordar diversas áreas de desempeño del Plan de Estudio como:  Desarrollo de Software: Implementación de una solución tecnológica, aplicando lenguajes de programación (Angular, Ionic, TypeScript, Node.js), así como herramientas de desarrollo y bases de datos no relacionales que en nuestro caso es Firebase.  Gestión de Proyectos: Aplicación de la metodología Scrum para la planificación y el control del avance del proyecto, asegurando la entrega oportuna de resultados.  Trabajo en Equipo: Colaboración efectiva entre los miembros del equipo, donde cada uno ha aportado con su experiencia en áreas como front-end, back-end y documentación. |
| Competencias | Desarrollo de Soluciones Tecnológicas: Aplicación de conocimientos en diseño, desarrollo y mantenimiento de aplicaciones móviles y web, utilizando tecnologías modernas como Firebase, Angular, Ionic, Typescript y Node.js.  Gestión de Proyectos: Implementación de la metodología Scrum, gestionando recursos, plazos y priorización de tareas para asegurar la entrega de resultados en cada sprint.  Trabajo en Equipo: Colaboración efectiva con el equipo de desarrollo, asignando roles y responsabilidades, comunicando avances y superando obstáculos en conjunto.  Aplicación de Bases de Datos: Implementación de una base de datos no relacional (Firebase) para manejar la autenticación y almacenamiento de información de los usuarios del sistema.  Solución de Problemas: Identificación y resolución de problemas técnicos y organizativos a lo largo del proyecto, adaptando soluciones y tomando decisiones oportunas. |

| **Contenidos del informe final** | |
| --- | --- |
| 1. Relevancia del proyecto APT | El Proyecto Sistema de unidad territorial buscó solucionar la gestión ineficiente de las juntas de vecinos mediante una plataforma digital que facilita la organización, comunicación y manejo de documentos. Este problema es relevante para el campo de la ingeniería de software, ya que responde a la creciente necesidad de digitalizar procesos administrativos y comunitarios en el ámbito local, particularmente en Chile, donde la descentralización y la participación ciudadana requieren herramientas más accesibles y modernas.  El proyecto se enfoca en la comunidad vecinal de las comunas de Santiago, donde las juntas de vecinos juegan un rol crucial en la gestión local y el contacto con los ciudadanos. Los usuarios principales afectados por esta situación son los directivos de juntas de vecinos y los ciudadanos que participan en actividades comunitarias, solicitan servicios o certificaciones.  El aporte del Proyecto radica en la automatización de tareas, como la inscripción de vecinos, la obtención del certificado de residencia, la reserva de espacios comunes y la postulación a proyectos vecinales, lo que genera una mejora en la eficiencia, transparencia y comunicación entre las juntas y la comunidad. Además, su implementación (real o simulada) optimiza la gestión administrativa, disminuyendo errores y tiempo de espera en los procedimientos tradicionales. |
| 2. Objetivos | Objetivo General: Desarrollar una plataforma digital que optimice la gestión y comunicación de las juntas de vecinos, facilitando la organización de actividades, manejo de documentos y acceso a servicios comunitarios para mejorar la eficiencia administrativa y la participación ciudadana.  Objetivos Específicos:  1. Crear un sistema de inscripción y gestión de perfiles de vecinos, permitiendo la fácil administración de usuarios.  2. Implementar un módulo para la solicitud de certificados y servicios comunitarios de manera digital.  3. Desarrollar una funcionalidad para la reserva y administración de espacios comunes en la junta de vecinos.  4. Crear un sistema de postulación y seguimiento de proyectos vecinales para mejorar la transparencia y participación.  5. Desplegar una interfaz intuitiva y accesible tanto para los administradores de la junta como para los vecinos. |
| 3. Metodología | Para el desarrollo del Proyecto, se utilizó la metodología ágil Scrum, que se caracteriza por su enfoque iterativo e incremental. Esta metodología permitió una gestión efectiva del equipo, facilitando la adaptación a cambios y el enfoque en la entrega de valor constante durante cada sprint.  Planificación del Proyecto:  Se definieron los objetivos generales y específicos, las tareas técnicas y los entregables mediante herramientas como Trello para la asignación de tareas.  Se identificaron los principales riesgos y requerimientos tanto funcionales como no funcionales.  El proyecto se dividió en 6 Sprint de duración fija, cada uno con objetivos específicos como la gestión de usuarios, la reserva de espacios comunes, y la postulación a proyectos.  En cada sprint se realizaron reuniones diarias para el seguimiento del progreso y la identificación de obstáculos.  Al final de cada sprint, se realizó una revisión del avance, ajustando los siguientes pasos según la retroalimentación recibida del equipo y las pruebas de usabilidad.  Se implementaron pruebas funcionales y de integración para asegurar el correcto funcionamiento de cada módulo desarrollado.  La plataforma se desplegó en entornos controlados para pruebas finales antes de su implementación en producción.  Scrum fue elegido por su flexibilidad y capacidad para adaptarse a los cambios y desafíos que surgieron a lo largo del proyecto. Dado que el desarrollo de un sistema como este implica la iteración constante sobre funcionalidades críticas, la metodología ágil facilitó el ajuste de objetivos y la entrega incremental de valor. Además, permitió una colaboración más cercana con los stakeholders para asegurar que el sistema cumpliera con sus expectativas y requerimientos específicos. |
| 4. Desarrollo | El Proyecto se desarrolló en las siguientes etapas clave:  Definición de los objetivos generales y específicos.  Establecimiento del plan de trabajo y asignación de roles.  Identificación de riesgos, requerimientos funcionales y no funcionales.  Sprint 1: Administración de cuentas de usuario, espacios públicos y actividades.  Sprint 2: Gestión de solicitudes de registro, actividades y postulaciones a proyectos.  Sprint 3: Administración de roles, gestión de cambios de contraseña, postulación a actividades y petición de espacios públicos.  Se realizaron pruebas funcionales y de integración en cada sprint para verificar que las funcionalidades implementadas cumplieran con los requerimientos.  Al final de cada sprint, se realizaron revisiones para ajustar las tareas y metas, adaptando el desarrollo a los cambios o imprevistos detectados. Implementación en el entorno de pruebas para evaluación y despliegue final.  Trabajo en equipo: La comunicación fluida y colaboración entre los miembros del equipo facilitó la identificación y solución rápida de problemas.  Herramientas como Trello y el marco de trabajo Scrum permitieron una mejor gestión del tiempo y las tareas.  Las reuniones diarias y la retroalimentación continua permitieron realizar ajustes tempranos y evitar retrasos.  Problemas técnicos: En algunas fases del desarrollo, la integración de módulos presentó dificultades que obligaron a dedicar más tiempo a la depuración.  Desajustes en el alcance del proyecto: Algunos requerimientos se redefinieron, lo que requirió ajustes en el cronograma de actividades.  Limitaciones de recursos: Hubo momentos en los que la disponibilidad de recursos técnicos, como el entorno de pruebas, fue limitada, lo que generó demoras.  Ajustes Realizados:  Ajustes en los plazos: Ante las dificultades técnicas y ajustes de requerimientos, fue necesario reprogramar algunos sprints para dar más tiempo al desarrollo y pruebas.  Priorización de tareas: Algunas funcionalidades menos críticas se relegaron a fases posteriores para priorizar aquellas de mayor impacto.  Estos ajustes permitieron cumplir con los objetivos establecidos y mantener el ritmo de trabajo, garantizando el éxito en el desarrollo. |
| 5. Evidencias | Diagrama de casos de uso: Muestra las funcionalidades del sistema y cómo interactúan los usuarios.  Capturas del sitio web y app: Visualizan el desarrollo del sistema web y móvil.  Actas de reuniones y sprint reviews: Reflejan el progreso y los ajustes realizados.  Base de datos en Firebase: Evidencia de la estructura y gestión de datos.  Trello: Detalles del seguimiento de tareas y estados de avance en cada sprint. |
| 6. Intereses y proyecciones profesionales | Pruebas de Funcionalidad (Informe de Testing):  Descripción: Informe que detalla las pruebas realizadas a las distintas funcionalidades del sistema, incluyendo resultados, errores encontrados y soluciones aplicadas.  Relevancia: Muestra cómo se aseguró la calidad del producto final, garantizando que las funcionalidades operen según lo esperado.  Repositorio en GitHub:  Descripción: El repositorio contiene todo el código fuente del proyecto y evidencia de los comentarios realizados por cada miembro del equipo.  Relevancia: Permite verificar el aporte individual de cada integrante, así como la evolución del proyecto desde su inicio hasta su implementación final.  Justificación Estas evidencias permiten visualizar y comprender el desarrollo del proyecto desde sus fases iniciales de planificación hasta la implementación final. Además, muestran la aplicación correcta de metodologías y herramientas técnicas propias de la disciplina, asegurando la calidad y funcionalidad del sistema. |