**Definición Proyecto APT**

**Asignatura Capstone.**

Proyecto APT “UROOM”

Por Nicol Sánchez, Iván Duarte y Benjamín Salas. Duoc UC sede Viña del Mar.

2025.

Abstract

The URoom project focuses on developing a web platform designed for university students who need to rent properties in a secure and reliable way. The system integrates registration and authentication processes for both students and property owners, including validation of student enrollment certificates and property document management. It also provides features such as photo upload and editing, account deletion, a property management panel, advanced search with filters, map visualization, and a real-time chat service using WebSockets. The solution is built on a microservices architecture, emphasizing scalability, data security, and user experience. The project follows agile methodologies to manage the development process, ensuring incremental deliveries and software aligned with the defined requirements.

El proyecto URoom consiste en el desarrollo de una plataforma web orientada a estudiantes universitarios que buscan arrendar propiedades de forma segura y confiable. El sistema integra un proceso de registro y autenticación tanto para estudiantes como para propietarios, incorporando validación de certificados de alumno regular y gestión de documentos de propiedad. Además, ofrece funcionalidades como carga y edición de fotos, eliminación de cuentas, panel de administración de propiedades, búsqueda avanzada con filtros, visualización en mapa y un servicio de chat en tiempo real mediante WebSockets. La solución se construye sobre una arquitectura de microservicios, priorizando la escalabilidad, la seguridad en el manejo de datos y la experiencia del usuario. El proyecto aplica metodologías ágiles para la gestión del desarrollo, asegurando entregas incrementales y un software alineado a los requerimientos planteados.

## Índice

[Introducción 3](#_Toc207929472)

[1. Antecedentes Personales 4](#_Toc207929473)

[2. Descripción Proyecto APT 4](#_Toc207929474)

[3. Fundamentación Proyecto APT 5](#_Toc207929475)

[4. Objetivos 8](#_Toc207929476)

[5. Metodología 8](#_Toc207929477)

[6. Evidencias 10](#_Toc207929478)

[7. Plan de Trabajo 12](#_Toc207929479)

[8. Carta Gantt 14](#_Toc207929480)

[9. Conclusion 15](#_Toc207929481)

[10. Reflection 15](#_Toc207929482)

[Referencias 16](#_Toc207929483)

## Introducción

El acceso a arriendos seguros y confiables se ha convertido en un desafío creciente para los estudiantes universitarios en Chile. Según estadísticas del área de Delitos Económicos de la PDI, durante enero y febrero de 2022 se registraron más de 132 denuncias por estafas vinculadas al arriendo de propiedades, cifra que representa el 88,6 % de todos los casos reportados en 2021, evidenciando un aumento significativo en las fraudes en este sector (PDI Chile, 2022).

Frente a este contexto, surge el proyecto URoom, una plataforma web diseñada para conectar de manera segura a estudiantes con propietarios, garantizando la verificación de documentos, la gestión de inmuebles y la comunicación confiable entre las partes. Este proyecto no solo aborda un problema real del mercado inmobiliario estudiantil, sino que también permite aplicar competencias clave de la Ingeniería en Informática, como desarrollo de software seguro, diseño de bases de datos, gestión ágil de proyectos y pruebas de aseguramiento de calidad.

|  |
| --- |
| 1. Antecedentes Personales |

|  |  |
| --- | --- |
| Nombre estudiante | **Nicol Sánchez - Iván Duarte - Benjamín Salas** |
| Rut | **20723184-3 , 17.812.917-1 , 21247311-1** |
| Carrera | **Ingeniería en Informática** |
| Sede | **Viña del Mar** |

|  |
| --- |
| 2. Descripción Proyecto APT |

|  |  |
| --- | --- |
| Nombre del proyecto | URoom |
| Área (s) de desempeño(s) | Gestión de Proyectos Informáticos   * Aplicación de metodologías ágiles (Scrum) en la planificación, ejecución y control del proyecto. * Elaboración de backlog, épicas, historias de usuario, planificación de sprints y gestión de entregables.   Desarrollo de Software y Programación   * Implementación de la plataforma web con tecnologías frontend y backend. * Construcción de APIs para registro de usuarios, carga de documentos y gestión de propiedades. * Uso de buenas prácticas de programación, modularidad y control de versiones.   Bases de Datos e Integración de Sistemas   * Diseño y gestión de la base de datos para usuarios, propiedades y certificados. * Normalización y definición de relaciones para garantizar integridad y eficiencia. * Integración con servicios externos (ej. Google Maps API para geolocalización).   Seguridad Informática   * Implementación de cifrado de contraseñas y manejo de sesiones seguras con JWT. * Gestión de permisos y roles para estudiantes y propietarios. * Almacenamiento seguro de documentos y certificados.   Calidad de Software y Pruebas   * Diseño de casos de prueba unitarios, de integración y end-to-end (E2E). * Validación de la funcionalidad en entorno de staging antes del despliegue. * Uso de herramientas de automatización de pruebas. |
| Competencias | Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria.   * Se aplicarán pruebas unitarias, de integración y end-to-end para validar el correcto funcionamiento de los flujos principales (registro, publicación de propiedades, carga de certificados, chat). * Se utilizarán buenas prácticas de QA y herramientas de testing automatizado para garantizar la calidad del producto.   Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización.   * El proyecto se gestionará mediante metodología ágil (Scrum), planificando sprints, gestionando el backlog y evaluando avances en reuniones de seguimiento. * Se tomarán decisiones sobre prioridades de desarrollo y asignación de recursos, considerando tanto los requerimientos funcionales como los no funcionales.   Construir modelos de datos para soportar los requerimientos de la organización de acuerdo a un diseño definido y escalable en el tiempo.   * Se diseñará una base de datos relacional para manejar usuarios, propiedades, documentos y mensajes, asegurando la integridad de la información. * El modelo de datos será escalable, permitiendo integrar nuevas funcionalidades en el futuro (ej. sistema de pagos o recomendaciones).   Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos.   * Se implementará la plataforma web siguiendo buenas prácticas de arquitectura (separación frontend-backend, APIs RESTful, seguridad con JWT). * Se utilizará control de versiones (GitHub), integración continua (CI/CD) y despliegue en la nube, lo que facilita el mantenimiento y evolución de la solución |

## 3. Fundamentación Proyecto APT

|  |  |
| --- | --- |
| Relevancia del proyecto APT | El proyecto busca solucionar la dificultad y falta de confianza en el proceso de arriendo de viviendas para estudiantes universitarios en Chile. Actualmente, muchos estudiantes enfrentan problemas para encontrar alojamiento seguro, confiable y adaptado a sus necesidades, mientras que los arrendadores también requieren mecanismos para validar la identidad y seriedad de los postulantes. El proceso tradicional suele ser informal, poco transparente y expone a ambas partes a riesgos de fraude, suplantación de identidad y falta de información verificada.  Relevancia para el campo laboral  Este tema es altamente relevante para el campo de la ingeniería en informática y el desarrollo de software, ya que aborda la digitalización de procesos críticos en el sector inmobiliario y educativo. La solución propuesta utiliza tecnologías modernas (Node.js, TypeScript, Prisma, OCR, JWT, microservicios) para automatizar la validación de documentos, mejorar la seguridad y optimizar la experiencia de usuario, aportando valor real a organizaciones que gestionan arriendos y a instituciones educativas que buscan apoyar a sus estudiantes.  **Contexto de la problemática**  La situación se ubica principalmente en Chile, en comunas y regiones con alta concentración de universidades y estudiantes, como Santiago, Valparaíso y Concepción. Estas zonas presentan una alta demanda de arriendos temporales y una oferta variada, pero con poca regulación y escasa verificación de antecedentes. El proyecto impactaría a estudiantes universitarios (grupo etario 18-28 años), arrendadores particulares y empresas inmobiliarias que buscan digitalizar y profesionalizar sus procesos de arriendo.  Impacto y aporte de valor  El proyecto aporta valor al contexto laboral y social al automatizar la validación de identidad y documentos (carnet, certificados universitarios, cuentas de servicios) usando OCR y scraping, reduciendo el riesgo de fraude.  Facilitar la gestión de propiedades para arrendadores, permitiendo registrar y administrar viviendas de forma segura y transparente.  Mejorar la experiencia de los estudiantes, asegurando que solo usuarios verificados puedan acceder a ofertas de arriendo.  Contribuir a la formalización y digitalización del mercado de arriendos estudiantiles, alineándose con las tendencias tecnológicas y necesidades reales del sector. |
| Descripción del Proyecto APT | El proyecto busca generar una conexión entre personas que quieren arrendar su inmueble de forma segura a estudiantes y estudiantes que quieren buscar inmuebles para arrendar de forma segura. Para abordar esto y sobre todo su valor agregado que es la seguridad, se integrará un sistema de verificación tanto para los usuarios arrendadores como a los usuarios estudiantes, ambos verificarán que son reales. |
| Pertinencia del proyecto con el perfil de egreso | El Proyecto APT "URoom" se relaciona de manera integral con el perfil de egreso de Ingeniería en Informática, ya que abarca prácticamente todas las competencias técnicas descritas en el perfil  **1. Levantamiento y análisis de requerimientos:**  Identificación de necesidades específicas del mercado de arriendos estudiantiles  Análisis de requerimientos funcionales (validación de documentos, gestión de usuarios, sistema de propiedades)  Definición de requerimientos no funcionales (seguridad, escalabilidad, disponibilidad)  **2. Desarrollo de sistemas computacionales:**  Implementación de un sistema completo con arquitectura de microservicios  Desarrollo de APIs RESTful usando Node.js y TypeScript  Creación de servicios especializados (autenticación, validación de documentos, gestión de propiedades)  **3. Integración de tecnologías de información:**  Integración de múltiples tecnologías: OCR (Tesseract), bases de datos (MySQL), almacenamiento en la nube (MinIO/S3)  Comunicación entre servicios usando JWT y APIs  Integración con servicios externos (scraping de certificados universitarios)  **4. Desarrollo de soluciones tecnológicas:**  Solución innovadora para un problema real del sector inmobiliario estudiantil  Automatización de procesos manuales mediante tecnología  Implementación de validación automática de documentos usando IA/OCR  **5. Seguridad de sistemas computacionales:**  Implementación de autenticación JWT con claves asimétricas (RS256)  Validación rigurosa de documentos para prevenir fraude  Middleware de autorización por roles (estudiante/arrendador)  Protección de datos sensibles y encriptación de contraseñas  **6. Aseguramiento de la calidad del software:**  Uso de TypeScript para tipado estático y prevención de errores  Validación de datos con Zod schemas  Manejo robusto de errores y logging detallado  Estructura modular y mantenible del código  **7. Gestión de proyectos informáticos:**  Planificación y desarrollo iterativo del sistema  Documentación técnica completa con diagramas de flujo y arquitectura  Gestión de versiones y control de cambios  Implementación de buenas prácticas de desarrollo  **8. Gestión de la información:**  Diseño de base de datos relacional eficiente  Gestión de archivos multimedia (imágenes, PDFs)  Implementación de sistemas de backup y recuperación  Optimización de consultas y rendimiento  Necesidad de las competencias para resolver la problemática  Las competencias seleccionadas son fundamentales para resolver la problemática porque:  La validación automática de documentos requiere conocimientos profundos de integración tecnológica y desarrollo de soluciones innovadoras  La seguridad del sistema es crítica al manejar documentos de identidad y datos personales sensibles  La escalabilidad y mantenibilidad del sistema demanda aplicación de buenas prácticas de ingeniería de software  La gestión eficiente de la información es esencial para manejar grandes volúmenes de usuarios y propiedades  El proyecto no solo aplica las competencias técnicas del perfil de egreso, sino que las integra de manera sinérgica para crear una solución tecnológica real y de impacto social, demostrando la formación integral que promueve Duoc UC. |
| Relación con los intereses profesionales | Dentro de los intereses profesionales gestionar proyectos con metodología ágil, desarrollar el backend como microservicios, integrarlos en el frontend y administrar la base de datos del proyecto. Este desarrollo ayudará bastante para obtener conocimientos necesarios y requeridos en el campo laboral. |
| Factibilidad de desarrollo del Proyecto APT | Las tecnologías que el proyecto necesita para ser desarrollado no tienen problema de implementación alguna o de desarrollo que pueda retrasar o dificultar el desarrollo en el tiempo que se estima el ramo de APT, por lo contrario, gracias al trabajo colaborativo del equipo completo y las soluciones de software que utilizaremos ejemplo *minIO* como almacenamiento o soluciones gratuitas para estudiantes en la nube como *“aws instructure”*, nos facilitan el desarrollo del proyecto. |

## 4. Objetivos

|  |  |
| --- | --- |
| Objetivo general | Ofrecer un servicio de arriendo para estudiantes de inmuebles de forma segura tanto para los estudiantes como para los arrendadores, generando seguridad y confianza gracias a nuestros sistemas de seguridad y autenticación, y de esta forma promover una convivencia segura entre ambas partes. |
| Objetivos específicos | Crear un sistema de registros para estudiantes  Crear un sistema de registros para arrendadores  Crear un sistema de verificación para estudiantes y para arrendadores con sus inmuebles  Crear un sistema de autenticación para estudiantes a través de un certificado de alumno regular  Crear un sistema de autenticación de propiedades de arrendadores a través de su información personal y del inmueble  Crear un sistema automatizado que utiliza avanzadas técnicas de scraping de información y reconocimiento de voz con inteligencia artificial.  Crear un sistema de comparación de datos que pueda validar los datos extraídos para generar validaciones de alto nivel para mantener los estándares de seguridad más altos de autenticación de usuarios.  Crear un sistema de administración de inmuebles para los arrendadores  Crear un sistema de valoración de inmuebles para los estudiantes y que así puedan valorar la calidad de una vivienda.  Crear un sistema de chat para conversar con el dueño de un inmueble.  Crear un sistema tipo dashboard administrativo para realizar análisis de datos de la empresa. |

## 5. Metodología

|  |
| --- |
| Descripción de la Metodología |
| Metodología seleccionada: SCRUM  Para el desarrollo del proyecto "URoom API System" se ha seleccionado la metodología ágil SCRUM, debido a que se adapta perfectamente a las características del proyecto y permite un desarrollo iterativo e incremental que facilita la entrega de valor temprano y la adaptación a cambios de requerimientos.  **Justificación de la elección de SCRUM**  **1. Naturaleza del proyecto:**  Sistema complejo con múltiples funcionalidades (autenticación, validación de documentos, gestión de propiedades)  Arquitectura de microservicios que permite desarrollo modular  Requerimientos que pueden evolucionar basados en feedback de usuarios  **2. Beneficios específicos para este proyecto:**  Entregas incrementales: Permite tener funcionalidades básicas operativas rápidamente  Flexibilidad: Facilita la incorporación de mejoras en el OCR y validación de documentos  Transparencia: Seguimiento claro del progreso en cada sprint  Calidad: Revisiones constantes aseguran la robustez del sistema de seguridad  **Estructura del equipo SCRUM**  **Scrum Master/Product Owner: Nicol Sánchez**  Responsabilidades: Facilitar ceremonias SCRUM, eliminar impedimentos, asegurar el cumplimiento de la metodología.  MaximizaR el valor del producto, definir qué se hace y prioriza.  Tareas específicas: Coordinar reuniones diarias, gestionar dependencias entre microservicios, resolver bloqueos técnicos.  **Development Team:**  **Backend Developer Lead: Iván Duarte**  Responsabilidades: Desarrollo de APIs, arquitectura de microservicios, integración de base de datos  Tareas específicas: Implementar servicios de autenticación JWT, desarrollar OCR y validación de documentos, configurar infraestructura de base de datos  **Frontend Developer: Benjamín Salas**  Responsabilidades: Desarrollo de interfaces de usuario, integración con APIs backend  Tareas específicas: Crear formularios de registro, implementar dashboard de arrendadores, desarrollar buscador de propiedades  **Planificación por Sprints**  **Sprint 0 (Setup - 1 semana):**  Configuración del entorno de desarrollo  Setup de repositorios y herramientas  Definición de estándares de codificación  Definición de épicas e historias de usuario  **Sprint 1 (Autenticación básica - 4 semanas):**  Implementar registro y login de estudiantes  Desarrollar sistema JWT  Crear base de datos inicial  Implementar OCR para certificados universitarios  Desarrollar scraping de validación  Integrar validación en el flujo de registro  **Sprint 2 (Gestión de arrendadores - 3 semanas):**  Implementar registro y login de arrendadores  Desarrollar validación de carnet de identidad  Crear perfil básico de arrendador - estudiante  **Sprint 3 (Microservicio de propiedades - 2 semanas):**  Desarrollar API de propiedades  Implementar CRUD de propiedades  Validación de cuentas de servicios para validar inmueble  **Sprint 4 (Chat y sistema de búsqueda- 2 semanas):**  Desarrollar sistema de chat inicializado por estudiantes  Implementar motor de búsqueda  Optimizar consultas de base de datos  **Sprint Planning (inicio de cada sprint):**  Duración: 2-4 horas dependiendo del sprint  Objetivo: Planificar trabajo del sprint, estimar historias de usuario, definir metas.  **Daily Meeting (diario):**  Duración: 15 minutos  Objetivo: Sincronizar progreso, identificar impedimentos, planificar día.  **Sprint Retrospective (final de cada sprint):**  Duración: 1 hora  Objetivo: Identificar mejoras en el proceso, ajustar metodología  Herramientas de gestión  Gestión de proyecto: Jira  Control de versiones: GitHub  Comunicación: Discord para comunicación diaria  Documentación: Confluence para documentación técnica  Testing: Postman para testing de APIs, Jest para testing unitario  **Código desarrollado y revisado (code review)**  Pruebas unitarias implementadas y pasando  Pruebas de integración exitosas  Documentación técnica actualizada  Validación de criterios de aceptación por Product Owner  Deploy exitoso en ambiente de testing  Esta metodología asegura un desarrollo ordenado, con entregas de valor constantes y la flexibilidad necesaria para adaptar el sistema a las necesidades específicas del mercado de arriendos estudiantiles. |

## 6. Evidencias

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de evidencia  (avance o final) | Nombre de la evidencia | Descripción | Justificación |
| avance | Registro de daily meetings | Reuniones diarias que permiten actualizar las tareas realizadas el día anterior, las tareas a realizar y si hay dudas | Este registro permite evidenciar las reuniones diarias que realiza el equipo para seguir avanzando durante la creación del software |
| avance | Sprint Planning | Reuniones de 2 a 4 horas al comienzo de cada sprint. | Planificar trabajo del sprint, estimar historias de usuario, definir metas. |
| final | Evidencia de Carta Gantt | Documento de carta gantt con la evidencia de las fases completadas. | Este registro permite tener un control visual claro de las tareas y el tiempo de ejecución de cada etapa |
| avance | Evidencia de flujo Jira | Fotografías del flujo dinámico de Jira. | Con estas evidencias se puede ver o monitorear cómo se ha sido completando el backlog de Jira |
| final | Incremento del Software (versión funcional) | Una versión desplegable y funcional del software que incorpora las características desarrolladas en un sprint o conjunto de sprints. | Demuestra el valor entregado al cliente de forma tangible y permite la retroalimentación temprana sobre el producto. |
| avance | Documento o Herramienta de Product Backlog | Una lista priorizada de todas las funcionalidades, requisitos y mejoras deseadas para el producto. | Sirve como fuente única de verdad para el trabajo a realizar, asegurando que el equipo trabaje en lo más valioso para el negocio. |
| avance | Sprint Backlog (tareas y objetivos del sprint) | Un subconjunto del Product Backlog seleccionado para un sprint, junto con el plan para entregar ese incremento y alcanzar el objetivo del sprint. | Proporciona una visión clara de lo que el equipo se compromete a entregar en un sprint y cómo lo hará. |
| avance | Gráfico de Burn-down del Sprint | Un gráfico que muestra la cantidad de trabajo restante en un sprint a medida que pasa el tiempo. | Ayuda al equipo a visualizar el progreso hacia el objetivo del sprint y a identificar posibles retrasos o problemas. |
| avance | Registro de Impedimentos | Una lista de obstáculos o bloqueos que impiden al equipo avanzar eficientemente. | Facilita la identificación, seguimiento y resolución de problemas que afectan la productividad del equipo. |
| final | Acta o Resumen de Sprint Review | Documentación de la reunión donde se inspecciona el incremento y se adapta el Product Backlog si es necesario. | Asegura la transparencia y la colaboración entre el equipo de desarrollo y los stakeholders. |
| avance | Sprint Retrospective | Documentación de la reunión donde el equipo reflexiona sobre el sprint pasado para identificar mejoras | Fomenta la mejora continua del proceso y la dinámica del equipo. |
| avance | Tablero Kanban Visual | Un tablero que visualiza el flujo de trabajo, mostrando las tareas en diferentes estados (por hacer, en progreso, completado). | Mejora la visibilidad del flujo de trabajo, ayuda a limitar el trabajo en progreso y a identificar cuellos de botella. |

## 7. Plan de Trabajo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Plan de Trabajo Proyecto APT | | | | | | |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable[[1]](#footnote-1) | Observaciones |
| Gestión de proyectos. | Planificar tareas | Planificar tareas de desarrollo del proyecto, asignar responsables, definir Sprints, historias de usuario, épicas y requerimientos | Jira  Scrum Master | 27/08 – 03/09 | Nicol Sánchez | Estimar la carga de trabajo realista (story points) |
| Desarrollo de software, pruebas de certificación. | Registro y acceso de estudiante | Implementar UI + API de registro/login con validaciones y scrapper de certificados. | Frontend (React), Backend (Node.js, Express), BD, Scrapper Python. | 4/09 – 12/09 | Iván Duarte  Benjamín Salas | Requiere pruebas de seguridad en formularios. |
| Desarrollo de software | Registro y acceso de propietario | Crear interfaz y endpoints de registro/login para arrendadores con validación de identidad básica. | Backend + Frontend. | 30/09 – 06/10 | Iván Duarte  Benjamín Salas | Similar al de estudiante, pero con atributos distintos. |
| Modelado de datos, desarrollo seguro. | Subir certificado de alumno regular | Subida del documento PDF y vinculación al estudiante. Validación de formato. | Microservicio cert\_storage\_service, MinIO | 22/09 – 29/09 | Iván Duarte | Depende del scrapper. |
| pruebas de seguridad. | Seguridad de contraseñas y sesiones | Hash de contraseñas (bcrypt), JWT para sesiones, expiración | Node.js, librerías de seguridad. | 3/9 – 9/10 | Iván Duarte  Benjamín Salas | Crítico para cumplir buenas prácticas de industria. |
| Desarrollo de software | Panel propietario: CRUD propiedades | Crear, eliminar, editar, actualizar y consultar propiedades. | React, API Node.js, BD. | 10/10 – 16/10 | Iván Duarte  Benjamín Salas | Integrar validación de fotos/documentos. |
| Modelado de datos, desarrollo de software. | Almacenamiento seguro de certificado | Guardar certificado en MinIO con URLs y permisos por usuario. | MinIO, cert\_storage\_service | 14/09 – 17/10 | Iván Duarte | Debe cumplir con normas de seguridad. |
| desarrollo de software. | Subida / edición / eliminación de foto de usuarios | Gestión de fotos de perfil en MinIO. | MinIO, Backend Node.js. | 20/10 -21/10 | Iván Duarte  Benjamín Salas | Validación de formatos (JPG/PNG). |
| Modelado de datos, desarrollo de software. | Documentos de propiedad (boletas, comprobantes) | Adjuntar documentos asociados a propiedades en MinIO. | MinIO, Backend. | 17/10 -24/10 | Iván Duarte | Deben relacionarse con cada propiedad registrada. |
| Modelado de datos, desarrollo de software | Eliminación de cuenta | Endpoint para eliminar usuario y datos asociados | Backend Node.js, MinIO. | 27/10 -29/10 | Iván Duarte | Asegurar borrado lógico/físico según políticas. |
| pruebas de certificación, desarrollo de software. | Búsqueda y filtros de propiedades | Implementar buscador con filtros por precio, comuna, disponibilidad. | Backend (consultas SQL optimizadas), Frontend React. | 03/11 – 07/11 | Nicol Sánchez | Optimizar rendimiento con índices en BD. |
| Desarrollo de software, Modelado de datos. | Visualización en mapa (estudiante) | Integración con Google Maps / Leaflet para mostrar propiedades en mapa. | API Maps, Frontend React. | 10/11 – 14/11 | Nicol Sánchez | Depende del CRUD de propiedades. |
| Desarrollo de software, pruebas de certificación | Chat estudiante propietario (websocket) | Microservicio de mensajería en tiempo real con WebSocket. | Backend Node.js, WebSockets, BD mensajes. | 17/11 - 28/11 | Nicol Sánchez | Requiere autenticación JWT para los mensajes. |

## 8. Carta Gantt

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **Fase 1** | | | | | | **Fase 2** | | | | | | | | | | | | | | | **Fase 3** | | | | |
| **S 1** | | **S 2** | | **S 3** | **S 4** | **S 5** | **S 6** | **S 7** | **S 8** | **S 9** | **S 10** | | | **S 11** | **S 12** | **S 13** | **S 14** | | **S 15** | **S 16** | | | **S 17** | **S 18** |
| **Desarrollo de idea del proyecto.** |  |  | |  | |  |  |  |  |  |  |  | | |  |  |  |  | |  |  | | |  |  |
| **Definición de tecnologías a utilizar.** |  |  | |  | |  |  |  |  |  |  |  | | |  |  |  |  | |  |  | | |  |  |
| **Sprint 0**  **(Formalización del proyecto – configuración de entorno de desarrollo.** |  |  | |  | |  |  |  |  |  |  |  | | |  |  |  |  | |  |  | | |  |  |
| **Hito 1 (Definición del proyecto APT).** |  |  | |  | |  |  |  |  |  |  |  | | |  |  |  |  | |  |  | | |  |  |
| **Sprint 1 Sistema de certificación.** |  | |  | |  |  |  | | | |  |  | | |  |  |  |  | |  |  | | |  |  |
| **Sprint 2 (Gestión propia de usuarios y creación de propiedades).** |  | |  | |  |  |  |  |  |  |  | | | | |  |  |  | |  |  | | |  |  |
| **Hito 2**  **Entrega avance de proyecto.** |  | |  | |  |  |  |  |  |  |  | |  |  | |  |  |  | |  |  | | |  |  |
| **Sprint 3 (Sistema de geolocalización de propiedades).** |  | |  | |  |  |  |  |  |  |  |  | | |  |  | |  | |  |  | | |  |  |
| **Sprint 4 (Chat de comunicación Estudiante y Propietario).** |  | |  | |  |  |  |  |  |  |  |  | | |  |  |  |  | | |  | | |  |  |
| **Hito 3**  **Entrega Final del proyecto.** |  | |  | |  |  |  |  |  |  |  |  | | |  |  |  |  |  | |  | | |  |  |
| **Hito 4 Presentación proyecto APT.** |  | |  | |  |  |  |  |  |  |  |  | | |  |  |  |  |  | |  | | |  |  |
| **Finalización del proyecto.** |  | |  | |  |  |  |  |  |  |  |  | | |  |  |  |  | |  |  | |  | |  |

## 9. Conclusion

**Iván Duarte Herrera**

The *URoom* project demonstrates the practical application of the knowledge and skills acquired throughout the Computer Engineering program. By addressing a real and relevant social problem, the lack of secure and reliable housing options for university students in Chile, the project validates the importance of combining technical expertise with secure user-centered solutions. The implementation of a microservices architecture, secure authentication systems, OCR-based document validation, and agile methodologies has proven that it is possible to deliver a scalable, maintainable, and high-quality software product within the constraints of academic and professional practice.

**Nicol Sánchez**

The URoom project is fully feasible to implement, as its technical and functional design meets the established requirements and relies on well-defined methodologies. It allows me to directly apply the competencies I have developed during my Computer Engineering studies. Furthermore, this project is closely aligned with my professional interests.

**Benjamín Salas**

I am excited to be able to participate in the creation of URoom and to put to the test my knowledge and skills that I have learned throughout my time at Duoc, and I hope to continue learning in the future thanks to more projects like this.

## 10. Reflection

This project challenges us to put into practice a wide range of skills related to the development of computer solutions that we acquired throughout our career in Computer Engineering.

Thanks to this experience, we now have a clearer perspective on the expectations and demands of the professional world. Beyond the technical aspects, it also allowed us to strengthen our teamwork, communication, and project management abilities by applying agile methodologies in a real context. Additionally, working on topics such as security, scalability, and system integration gave us the opportunity to face real-world challenges and prepare ourselves for future roles in the software development industry.

## Referencias

PDI Chile. (2022, marzo 23). *Verano 2022: suben estafas en arriendo de casas*. Policía de Investigaciones de Chile. [https://www.pdichile.cl/centro-de-prensa/detalle-prensa/2022/03/23/verano-2022-suben-estafas-en-arriendo-de-casas](https://www.pdichile.cl/centro-de-prensa/detalle-prensa/2022/03/23/verano-2022-suben-estafas-en-arriendo-de-casas?utm_source=chatgpt.com)

1. En caso de que el Proyecto APT sea grupal, en esta columna deben indicar el nombre de los responsables de cada tarea o actividad. Esto posteriormente permitirá diferenciar la evaluación por cada integrante. [↑](#footnote-ref-1)