**Guía3. Informe final Proyecto APT**

**Asignatura Capstone**

|  |
| --- |
| **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 | ClaroVTR – Gestión y Auditoría de Visitas Técnicas |
| Área (s) de desempeño(s) | * Desarrollo de Software (backend y frontend). * Bases de Datos y APIs REST. * Aseguramiento de Calidad (QA). * Gestión de Proyectos Ágiles (Scrum). |
| Competencias | * Levantamiento y análisis de requerimientos (criterios de aceptación). * Modelado de datos y diseño de endpoints en Django REST Framework. * Desarrollo web con autenticación segura (JWT en cookies + CSRF). * Diseño y ejecución de pruebas funcionales e integración (QA). * Planificación y gestión por sprints con Jira |

|  |  |
| --- | --- |
| **Contenidos del informe final** | |
| 1. Relevancia del proyecto APT | El proyecto busca mejorar la gestión de visitas técnicas a domicilio (internet/TV) y la auditoría asociada, problema común en empresas de telecomunicaciones. En nuestro caso, se sitúa en Santiago de Chile y se enfoca en automatizar tareas claves: carga de direcciones, asignación a técnicos, cambios de estado del cliente y registro de auditorías. |
| 2. Objetivos | **Objetivo general**  Diseñar e implementar un MVP web que permita gestionar y auditar visitas técnicas, con autenticación segura, carga masiva de direcciones, autoasignación controlada, cambios de estado y registro de auditorías.  **Objetivos específicos**   * Implementar autenticación con JWT en cookies (HttpOnly) y protección CSRF. * Modelar las entidades clave (Usuarios, Asignaciones, Reagendamientos, Auditorías) e indexar para buen rendimiento. * Habilitar carga masiva de direcciones desde CSV/XLSX con validaciones y upsert. * Desarrollar autoasignación de visitas por técnico con control de concurrencia. * Gestionar estados del cliente (autoriza, sin moradores, rechazo, contingencia, masivo, reagendo). * Registrar auditorías y trazabilidad (historial de acciones y reagendamientos). |
| 3. Metodología | **Metodología utilizada**  Utilizamos un enfoque ágil (Scrum) con sprints cortos. Fechas planificadas: Sprint 1 (02–15 Sep), Sprint 2 (16–29 Sep), Sprint 3 (30 Sep–13 Oct) y Sprint 4 (14–31 Oct). Trabajamos con backlog en Jira, dailys breves y retrospectivas para replanificar.  **Pertinencia de la metodología**  Scrum fue pertinente porque hubo cambios del cliente durante el proceso. El marco nos permitió priorizar el MVP, hacer ajustes controlados (re-estimación y mover alcance) y mantener visibilidad del avance y bloqueos. |
| 4. Desarrollo | **Etapas/actividades realizadas**   * Alcance y diseño inicial: consolidación de roles (técnico/administrador) y uso de RUT como identificador. * Configuración del backend (Django/DRF) y base de datos (Supabase). * Modelado y endpoints: usuarios, asignaciones, reagendamientos y auditorías. * Seguridad: JWT en cookies HttpOnly, protección CSRF, permisos por rol. * Carga masiva CSV/XLSX con validaciones, normalización y upsert por id\_vivienda. |
| 5. Evidencias | **Se consideran como evidencias del desarrollo y resultados:**   * Capturas de Swagger/ReDoc mostrando endpoints principales (auth, asignaciones, auditorías). * Archivo CSV/XLSX de ejemplo y respuesta de validación. * Video corto mostrando login, tablero técnico y autoasignación. * Capturas del tablero de Jira con historias completadas. * Diagrama de datos y/o script SQL de referencia utilizado. |
| 6. Intereses y proyecciones profesionales | Este proyecto me ayudó a confirmar mi interés por dos áreas: la gestión de producto (PO) y la calidad de software (QA). Me gustó priorizar el MVP y ordenar el backlog, y también diseñar pruebas para flujos críticos del sistema.  **Proyecciones laborales**   * Seguir aprendiendo automatización de pruebas (Selenium/Playwright) y pruebas de API (Postman/Newman). * Profundizar en métricas de calidad (defectos por sprint, tiempo de ciclo) y buenas prácticas de control de cambios. * En el corto plazo, postular a QA Automation Jr. o a un rol PO/Analista funcional Jr., según disponibilidad. |