**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 | *Cuenta Claras* |
| Área (s) de desempeño(s) | *Desarrollo Backend, Desarrollo Frontend, Seguridad de la Información, Bases de Datos y DevOps. Estas áreas se aplicaron para implementar la API y autenticación (JWT/bcrypt/2FA), el modelado relacional y seeds, las interfaces de usuario para creación/edición de cargos, y la orquestación del entorno con docker‑compose para pruebas reproducibles* |
| Competencias | *Diseño y desarrollo web (backend/frontend), modelado de BD, seguridad y control de acceso, integración de pagos, calidad de software y DevOps.* |

|  |  |
| --- | --- |
| **Contenidos del informe final** | |
| 1. Relevancia del proyecto APT | *El proyecto resolvió la gestión digital de cobros, cargos y conciliación para comunidades/condominios, reduciendo trabajo manual y errores contables. Es relevante para desarrollo de software en el ámbito administrativo/proptech en Chile (gestión de comunidades y cumplimiento SII), y afecta directamente a administradores, tesoreros y residentes. Aporta valor al centralizar trazabilidad de pagos, permitir conciliación automática con pasarelas y garantizar separación de datos por comunidad (multi‑tenant), mejorando eficiencia operativa y control financiero.* |
| 2. Objetivos | * *Objetivo general: desarrollar una plataforma web segura y reproducible para administrar comunidades, emitir cargos y conciliar pagos.* * *Objetivos específicos: 1) CRUD de comunidades, unidades y cargos; 2) modelado relacional y seeds para pruebas reproducibles; 3) autenticación segura (JWT, bcrypt) y tenancy + 2FA; 4) diseño e integración con pasarelas (webhooks) y asegurar calidad con lint/tests y docker‑compose.* |
| 3. Metodología | *Se utilizó Scrum (sprints, backlog priorizado, revisiones y retrospectivas). Fases: definición de requisitos → modelado DB y seeds → desarrollo backend (API, auth, tenancy) → frontend (formularios HU) → integración de pagos (diseño y stubs) → QA manual y planificación de automatización/CI. Scrum fue pertinente porque permitió priorizar MVP funcional (CRUD, reproducibilidad) y reagendar tareas dependientes (E2E con pasarelas, 2FA) sin bloquear entregas incrementales.* |
| 4. Desarrollo | * *Etapas realizadas: infraestructura reproducible (docker‑compose + seeds), backend (modelos, endpoints, auth), frontend (crear/editar cargos), documentación de pasarelas, configuración de lint y plan de tests.* * *Facilitadores: trabajo en equipo, seeds y docker para pruebas reproducibles, backlog claro.* * *Dificultades: dependencias externas con pasarelas (sandbox/credenciales), falta inicial de tests automatizados y necesidad de hardening (2FA, observabilidad).* * *Ajustes: uso de stubs/webhooks para pruebas locales; dividir integración de pagos en fases; replanificar 2FA y observabilidad a sprints finales; asignar sprint dedicado a tests críticos y CI.* |
| 5. Evidencias | *Archivos seleccionados para evaluación: [cuentasclaras.sql](vscode-file://vscode-app/c:/Users/Eduardo%20Rom%C3%A1n.DESKTOP-UH9G1BA/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") (esquema + seeds), ccbackend/src/auth/\* (JWT, bcrypt, 2FA), ccbackend/src/middleware/tenancy\* (aislamiento), [nuevo.tsx](vscode-file://vscode-app/c:/Users/Eduardo%20Rom%C3%A1n.DESKTOP-UH9G1BA/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") y editar/[id].tsx (HU), ccbackend/PAYMENT\_GATEWAY\_README.md, .eslintrc\* y [package.json](vscode-file://vscode-app/c:/Users/Eduardo%20Rom%C3%A1n.DESKTOP-UH9G1BA/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") (lint), jest/****tests****(tests) y docker-compose.yml. Estas evidencias permiten reproducir el entorno, verificar seguridad, flujos de negocio y la planificación de integración con pasarelas.* |
| 6. Intereses y proyecciones profesionales | *El proyecto confirmó interés por el desarrollo full‑stack con foco en seguridad y sistemas financieros/multi‑tenant; reforzó valoración de prácticas de calidad (lint/tests) y despliegue reproducible. Tras el APT, se proyecta profundizar en backend seguro, arquitecturas multi‑tenant, CI/CD y roles en fintech/proptech o ingeniería de software con responsabilidad sobre seguridad e integración de pagos.* |