

PTY4614 – Capstone\_Ingeniería en Informática, Sede Padre Alonso de Ovalle

Cybernine\_C9

Definición Proyecto APT – Fase 1

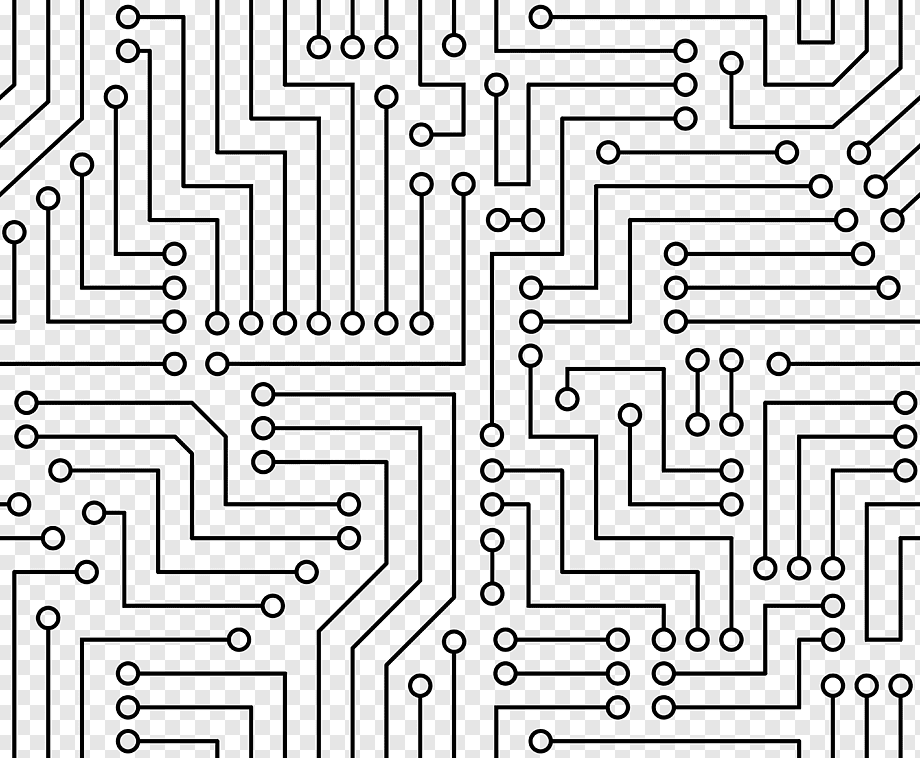
Estudiante: Ariel Gárate — RUT: 16.803.636-1

Sección: 007V — Docente: Cristóbal Beltrán

Correo institucional: ar.garate@duocuc.cl

Santiago, Segundo Semestre 2025

CYBERNINE\_C9



Índice

Contenido

[Índice 2](#_Toc209734825)

[Título del Proyecto 3](#_Toc209734826)

[1. Descripción del proyecto 3](#_Toc209734827)

[2. Relevancia para el contexto laboral/profesional 3](#_Toc209734828)

[3. Objetivos del proyecto 4](#_Toc209734829)

[3.1 Objetivo general 4](#_Toc209734830)

[3.2 Objetivos específicos (SMART) 4](#_Toc209734831)

[4. Alcance (MVP) y requerimientos 4](#_Toc209734832)

[4.1 Historias de usuario (borrador) 4](#_Toc209734833)

[4.2 Requisitos no funcionales 4](#_Toc209734834)

[5. Mapeo a Competencias del Perfil de Egreso 5](#_Toc209734835)

[6. Metodología de trabajo 5](#_Toc209734836)

[7. Plan de trabajo y cronograma (4 meses) 5](#_Toc209734837)

[8. Recursos y factibilidad 5](#_Toc209734838)

[9. Pruebas e indicadores de calidad 6](#_Toc209734839)

[10. Despliegue y guía de instalación 6](#_Toc209734840)

[11. Consideraciones éticas y de seguridad 6](#_Toc209734841)

[12. Abstract (EN) 6](#_Toc209734842)

[13. Conclusions (EN) 6](#_Toc209734843)

[14. Reflection (EN) 7](#_Toc209734844)

[15. Evidencias y anexos 7](#_Toc209734845)

[16. Checklist de entrega (control interno) 7](#_Toc209734846)

Título del Proyecto

**Cibernine – Sistema de Inventario para Almacenes de Barrio (MVP)**

1. Descripción del proyecto

**Problema:** *El proyecto aborda una problemática frecuente en los almacenes de barrio en Chile: la gestión de inventario con planillas/cuadernos, lo que deriva en quiebres de stock, sobreinventario y pérdida de tiempo. En el país operan del orden de 146.000 almacenes, botillerías y negocios de barrio (*[*dato SII citado por Diario Financiero*](https://www.df.cl/noticias/site/docs/20231207/20231207133332/suplemento_20231211.pdf?)*), por lo que la escala del problema es significativa. Además, evidencia reciente señala que 81% de estos locales sigue registrando inventario y ventas con lápiz y papel y solo 19% usa software, reforzando la baja digitalización del canal tradicional.*

**Solución propuesta (MVP):** - PWA responsiva operable desde móvil/PC. - Módulos: Productos, Movimientos (entradas/salidas con motivo), Alertas de reposición, Usuarios/Roles (Admin, Operador), Reporte Kardex. - Captura con cámara para **código de barras** (escaneo web) y carga manual. - “IA ligera”: sugerencias descriptivas sobre reposición y rotación (vía API/servicio externo).

**No incluido (futuro trabajo):** Incorporación de funcionalidades para vincular proveedores–clientes y automatizar compras, contabilidad.

**Usuarios:** dueño/operador de almacén

**Propuesta de valor:** *Aplicativo simple y operable en móvil/PC que permitirá a microcomercios de barrio registrar y controlar su inventario de forma ágil y confiable, reduciendo quiebres de stock y el uso de planillas/manuales.*

2. Relevancia para el contexto laboral/profesional

*Este proyecto resuelve un problema real de los negocios de barrio: llevar el inventario de forma simple para evitar quiebres y ahorrar tiempo. La solución es una PWA pensada para celular, fácil de usar incluso para personas con baja experiencia digital. En lo profesional, me permite practicar lo que pide el perfil: levantar requisitos, modelar datos, crear API, back-end, front-end, aplicar seguridad y pruebas. Además, desplegar con buena documentación usando un stack común en la industria.*

3. Objetivos del proyecto

3.1 Objetivo general

*Implementar una PWA para el control de inventario en almacenes de barrio con escaneo, alertas, información útil y autenticación para reducir quiebres, agilizar la operación y apoyar decisiones.*

3.2 Objetivos específicos (SMART)

* O1. Diseñar y desplegar una PWA funcional con módulos de Productos, Movimientos, Alertas y Kardex.
* O2. Implementar autenticación y control de acceso básico (Admin/Operador).
* O3. Integrar escaneo de código de barras vía cámara del dispositivo.
* O4. Incorporar un componente de “IA ligera” que genere sugerencias a partir del stock y movimientos.
* O5. Publicar el sistema en un entorno accesible (con guía de despliegue) y documentar pruebas.

4. Alcance (MVP) y requerimientos

4.1 Historias de usuario (borrador)

* **HU1 – Registrar producto:** Como Admin/Operador, quiero crear/editar productos (SKU, nombre, unidad, costo, precio, stock mín.) para mantener el catálogo actualizado. **CA:** validaciones, duplicados, paginado.
* **HU2 – Movimientos:** Como Operador, quiero registrar entradas/salidas con motivo y fecha para actualizar el stock. **CA:** cálculo de stock consistente, bitácora.
* **HU3 – Alertas:** Como Admin, quiero ver productos bajo stock mínimo para priorizar reposición. **CA:** listado filtrable/exportable.
* **HU4 – Código de barras:** Como Operador, quiero escanear códigos con la cámara para acelerar el registro. **CA:** fallback a ingreso manual.
* **HU5 – Kardex:** Como Admin, quiero un reporte por producto y rango de fechas para auditar movimientos. **CA:** totales y estado final.
* **HU6 – Usuarios/Roles:** Como Admin, quiero crear usuarios con rol para controlar permisos. **CA:** autenticación, sesiones/expiración.
* **HU7 – Sugerencias IA:** Como Admin, quiero recibir sugerencias (reposición, rotación) para decidir compras. **CA:** explicación breve y fuente de datos.

4.2 Requisitos no funcionales

* Seguridad básica (hash de contraseñas, validación server-side, manejo de errores).
* Despliegue simple (contenedores o servicio gestionado), disponibilidad demo.
* Documentación técnica y de usuario.

5. Mapeo a Competencias del Perfil de Egreso

* Levantamiento/Análisis de requerimientos → Secciones 4 y 6 (HU, criterios, entrevistas simuladas).
* Desarrollo/Integración de sistemas → API + Front PWA + DB.
* Seguridad de sistemas → Autenticación, roles, validaciones, manejo de errores.
* Aseguramiento de calidad → Casos de prueba, datos semilla, pruebas de integración básicas.
* Gestión de proyectos → Planificación, riesgos, cronograma.
* Gestión de la información → Modelo de datos, Kardex, consultas.
* Innovación/actualización → Escaneo por cámara, IA ligera.

6. Metodología de trabajo

Se adopta un enfoque ágil (Scrum adaptado a proyecto individual) en lugar de un ciclo tradicional en cascada. Esta elección es más adecuada porque el proyecto presenta incertidumbre técnica y de integración (uso de cámara en PWA bajo HTTPS, control de acceso por roles, desempeño en móviles reales), y requiere validaciones tempranas en dispositivo para no descubrir riesgos al final.

7. Plan de trabajo y cronograma (4 meses)

* **Sem 1–2:** Alcance final, HU/CA, mockups, DER (conceptual/lógico), entorno base.
* **Sem 3–4:** Modelo físico PostgreSQL, API (productos, movimientos), Front (catálogo/movimientos), pruebas básicas.
* **Sem 5–6:** Alertas, roles, Kardex, escáner códigos; IA ligera (endpoint y UI).
* **Sem 7:** Pruebas ampliadas, datos realistas, hardening, guía de despliegue.
* **Sem 8:** Pulido UI/UX, defensa técnica, documentación final.

8. Recursos y factibilidad

* Hardware/Software disponibles.
* Servicios externos (API IA, hosting). Presupuesto estimado.
* Riesgos clave y mitigaciones (alcance, BD, pruebas, despliegue).

9. Pruebas e indicadores de calidad

* **Pruebas funcionales:** HU1–HU7 (happy path y errores).
* **Pruebas de datos:** consistencia de stock tras secuencia IN/OUT.
* **Pruebas de seguridad básicas:** autenticación, roles.
* **Métricas mínimas:** tasa de errores en creación/edición, tiempo de respuesta API en operaciones clave, cobertura de HU con tests.

10. Despliegue y guía de instalación

* Entorno (producción/demo), variables de entorno, pasos de despliegue, cuentas de prueba.
* **Plan B:** ejecución local (script dev), dataset semilla.

11. Consideraciones éticas y de seguridad

* Tratamiento de datos (usuarios, inventario), contraseñas, logs.
* Limitaciones del componente de IA (transparencia y disclaimers).

12. Abstract (EN)

This project presents a Minimum Viable Product (MVP) Progressive Web App (PWA) for small neighborhood stores with low digital adoption. It addresses inefficient inventory control done with spreadsheets or paper logs, which often causes stockouts, overstock, and wasted time. The solution includes Products, Inventory Movements (in/out with reason and date), Replenishment Alerts, Users/Roles, and a Kardex report. It also adds barcode scanning using the device camera and lightweight AI to suggest basic restocking/rotation actions based on the store’s own data.

The method is iterative and incremental, with focus on data modeling (PostgreSQL), development (Node.js API + React PWA), functional and consistency tests (roles, IN/OUT sequences, Kardex), and deployment with an installation guide and a sample dataset. Expected results are fewer stockouts, better decisions, and faster data entry, without requiring high digital skills. As future work, the project may include supplier–client linking and purchase automation. The target deliverable is a functional demo that can be defended with complete evidence and documentation.

13. Conclusions (EN)

The Phase-1 definition shows that the MVP is feasible and fits the graduate profile. By fixing scope to Products, Movements, Alerts, Users/Roles, barcode scanning, and Kardex, the plan stays realistic and measurable. The data model and its rules (no negative stock, stable ordering of movements, Kardex that matches totals) support quality from the start.

Main risks are scope creep, auth/scanner complexity, and deployment setup. The plan reduces them with early prototypes, automated tests, a reproducible setup (.env.example, migrations/seed), and the PWA approach for one codebase across mobile/desktop.

The solution aims to lower stockouts and support decisions for users with low digital literacy, while keeping engineering effort under control. Limits are clear (no multi-tenant, no accounting, no supplier marketplace). Later work can add supplier linking, purchase automation, and a valued Kardex (e.g., weighted average or FIFO) if time allows. Overall, the plan is deliverable within the semester and has a clear path from requirements to a working demo with evidence.

14. Reflection (EN)

This project helped me balance ambition and realism. I chose a PWA instead of native apps to keep one codebase, allow camera access, and make deployment simpler for users who mainly need short and clear flows on their phones.

I learned the importance of data integrity: stock must come from movements, and the Kardex must match totals. These rules guided the schema and the tests. For users with low digital skills, I focused on simple screens, clear labels, and camera-first input.

Adding lightweight AI taught me to keep “intelligence” explainable: suggestions come from existing data and simple logic, not from complex or unclear predictions. On ethics and security, I follow minimal data collection, protected credentials, role-based access, and clear limits of the AI component.

From a management view, weekly sprints, a Definition of Done, and an evidence folder give structure and control. The key lesson is that small, consistent wins (a working report, a passing test, a successful deploy) are better than chasing a big feature that never ships.

15. Evidencias y anexos

* Capturas de pantallas clave, DER, diccionario de datos, casos de prueba, guía de despliegue, cronograma.

16. Checklist de entrega (control interno)

* Objetivos y alcance SMART definidos.
* HU y CA listos.
* DER + diccionario de datos.
* API y Front con módulos del MVP.
* Escaneo de código funcionando + fallback manual.
* IA ligera integrada con ejemplos.
* Pruebas documentadas.
* Despliegue accesible + guía.
* Abstract/Conclusions/Reflection (EN).
* Capturas y evidencias en anexos.