}

**Guía1. Definición Proyecto APT**

**Asignatura Capstone**

1. **PARTE I**

|  |
| --- |
| **1. Antecedentes Personales** |
| A continuación, se presenta una tabla en la que debes completar la información solicitada. |

|  |  |
| --- | --- |
| Nombre estudiante | Ariel Gárate |
| Rut | 16.803.636-1 |
| Carrera | Ingeniería en informática |
| Sede | Padre Alonso de Ovalle |

|  |
| --- |
| **2. Descripción Proyecto APT** |
| En la descripción debes señalar brevemente el nombre de tu proyecto APT y las competencias del perfil de egreso que vas a poner en práctica. Si en tu carrera están definidas las áreas de desempeño, también menciona a qué áreas de desempeño está vinculado el proyecto. |

|  |  |  |
| --- | --- | --- |
| Nombre del proyecto | | *Cybernine (C9).* |
| Área (s) de desempeño(s) | | *Desarrollo e integración de sistemas computacionales para gestión de inventario en microcomercios. (Incluye: API + PWA con escáner, modelamiento y reporte de datos. Seguridad básica y pruebas como criterios transversales.)* |
| Competencias | | *Diseñar y generar soluciones de software innovadoras y de calidad, aplicando el ciclo de vida de éste, según las características del proyecto, las mejores prácticas de la industria y sus estándares de calidad.*  *Diseñar y generar soluciones que permitan resolver los requerimientos de información en el contexto de las organizaciones, considerando bases de datos relacionales y no relacionales.*  *Evaluar y gestionar proyectos en su área de especialización profesional, durante todo el ciclo de vida, de acuerdo a buenas prácticas y utilizando metodologías y herramientas de software, para cumplir con los requerimientos de la organización en contextos tradicionales y ágiles.*  *Diseñar soluciones de software, abarcando todo el ciclo de vida de éste, de acuerdo a estándares, marcos de trabajo y regulatorios, tecnologías y metodologías que promueven la innovación, con foco en la calidad, seguridad y sostenibilidad del proyecto.* |
| **3. Fundamentación Proyecto APT** | | |
| A continuación, se presentan distintos campos que debes completar con la información solicitada. Esta sección busca que describas en detalle tu proyecto y justifiques su relevancia y pertinencia. | | |
| Relevancia del proyecto APT | *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.*  *El acceso móvil es clave para la adopción: la XI Encuesta de Acceso y Uso de Internet 2024 (*[*SUBTEL–Cadem*](https://www.subtel.gob.cl/wp-content/uploads/2025/02/Informe-Final-Subtel-Acceso-y-Uso-Internet-2024.pdf)*) muestra que el teléfono móvil es el principal dispositivo en los hogares (98,9% lo usa para conectarse). En hogares solo de personas mayores, aumenta la proporción de conexión “solo móvil” (37,3%), lo que hace razonable priorizar una PWA “mobile-first” con flujos simples y cámara para escanear. A su vez, aún existen 491 mil hogares desconectados y la brecha etaria es la principal causa, lo que justifica decisiones de UX simple y apoyo guiado.*  *Para el campo de Ingeniería en Informática, la iniciativa es pertinente porque integra levantamiento de requerimientos; modelamiento/gestión de datos; desarrollo e integración de una PWA (móvil/PC) con escaneo por cámara; seguridad básica (autenticación/roles) y aseguramiento de calidad (pruebas funcionales y consistencia de stock). Como aporte de valor, la herramienta agiliza el registro de movimientos, emite alertas de reposición y añade IA ligera que sugiere reposición/rotaciones basadas en los datos del negocio disminuyendo quiebres y mejorando decisiones sin reemplazar el criterio del usuario. En el plano país, el* [*Índice de Transformación Digital 2023*](https://www.ecommerceccs.cl/wp-content/uploads/2023/10/202310_ITD-Nacional_-Presentacion-E-commerce-Summit_2023.pdf) *(CCS) ubica a las empresas en “Intermedio Digital” y advierte que las microempresas retrocedieron a niveles de 2020, por lo que un MVP acotado (inventario, alertas, escáner, Kardex) es una intervención realista y de alto impacto para este segmento.*  *Trabajo futuro: en una etapa posterior, se podrían incorporar funcionalidades para vincular proveedores–clientes y automatizar compras, fuera del alcance del MVP.  PWA: app web instalable que funciona como “casi nativa”. Kardex: historial por producto con saldo acumulado. IA ligera: uso de modelos/servicios existentes para tareas puntuales, sin entrenar modelos propios.* | |
| Descripción del Proyecto APT | *Implementar una aplicación web progresiva (PWA), simple y operable en móvil/PC, que permita a microcomercios de barrio registrar y controlar su inventario de forma ágil y confiable, reduciendo quiebres de stock y el uso de planillas/manuales.*  *Cybernine consiste en una aplicación PWA operable en móvil y PC para microcomercios, que incluye los módulos Productos, Movimientos (entradas/salidas con motivo y fecha), Alertas de reposición, Usuarios/Roles (Administrador/Operador) y Reporte por producto. Además, integra escaneo de códigos de barras a través de la cámara del dispositivo y un componente de IA ligera que entrega sugerencias de reposición/rotaciones basadas en los datos del negocio. Se privilegian flujos cortos, interfaz clara y uso desde el teléfono, considerando baja alfabetización digital.*  *El abordaje metodológico considera, en primer lugar, el levantamiento de requisitos y la elaboración de prototipos de pantallas clave con un enfoque centrado en tareas simples. Luego, se definirá el modelo de datos en PostgreSQL, incorporando reglas de consistencia de stock basadas en los movimientos. A continuación, se realizará un desarrollo iterativo (backend Node.js y frontend React PWA) integrando el escaneo por cámara y las validaciones necesarias. En paralelo, se ejecutarán pruebas funcionales y de consistencia (roles, secuencias IN/OUT y verificación de Kardex), dejando registro de evidencias. Finalmente, se efectuará un despliegue accesible y se entregará una guía de instalación/uso, junto con un dataset de ejemplo para la demostración.  Fuera del alcance del MVP (trabajo futuro): vínculo proveedores–clientes y automatización de compras.* | |
| Pertinencia del proyecto con el perfil de egreso | *El proyecto es pertinente al Perfil de Egreso porque aplica en un caso real las competencias seleccionadas. En “Diseñar y generar soluciones de software innovadoras y de calidad”, se materializa un MVP en formato PWA y se recorre el ciclo de vida completo del software: requisitos, diseño, construcción, pruebas y despliegue, con buenas prácticas y evidencias. En “Diseñar y generar soluciones que permitan resolver los requerimientos de información”, se modela e implementa un esquema en PostgreSQL, con reglas de integridad de stock y reporte (historial y saldo). En “Evaluar y gestionar proyectos”, se ejecuta un plan de trabajo con sprints, gestión de riesgos y trazabilidad de evidencias para la toma de decisiones. Finalmente, en “Diseñar soluciones de software… con foco en calidad, seguridad y sostenibilidad”, se implementan autenticación y roles, validaciones y pruebas de consistencia, además de una IA ligera explicable para sugerencias operativas. Si se incluye la competencia de plataformas y dispositivos móviles, queda cubierta por el enfoque PWA mobile-first con escaneo por cámara.*  *Competencias seleccionadas y su aplicación en el proyecto:*  *Construir e implementar modelos de datos: diseño en PostgreSQL y reporte Kardex (historial de entradas/salidas y saldo). Evidencias: DER, diccionario, migraciones y consultas.*  *Desarrollar, integrar e implantar la solución: backend en Node.js, frontend PWA en React, escáner por cámara y componente de IA ligera para sugerencias. Evidencias: endpoints, UI funcional y despliegue.*  *Gestionar proyectos informáticos: plan de trabajo, cronograma, riesgos y trazabilidad de evidencias para la toma de decisiones.*  *Realizar pruebas de validación: casos sobre consistencia de stock (secuencias IN/OUT), roles/seguridad, alertas y Kardex, documentando resultados y mejoras.* | |
| Relación con los intereses profesionales | *Mis intereses profesionales se enfocan en: desarrollo de aplicaciones móviles y trabajo full-stack (frontend + backend), modelamiento y gestión de datos con criterios de calidad/seguridad, UX para baja alfabetización digital (especialmente personas mayores), integración de IA ligera para apoyo a decisiones y gestión de proyectos con evidencias y despliegue.* | |
| Factibilidad de desarrollo del Proyecto APT | *El proyecto es factible porque el alcance está acotado a un MVP implementado con tecnologías estándar (Node.js/React PWA y base de datos relacional) y recursos disponibles (PC/Mac, teléfono con cámara, repositorio y hosting simple). Como fortalezas, existe claridad de alcance, capacidad full-stack y equipamiento propio; como oportunidades, hay paquetes ampliamente utilizados y mantenidos (lector de códigos, IA ligera) y alto uso de acceso móvil que valida el enfoque PWA mobile-first. Entre las debilidades, se reconoce que el proyecto es individual y requiere una curva de aprendizaje en cámara/autenticación; como amenazas, destacan la deriva de alcance y la fricción de despliegue (variables, BD, HTTPS). Para mitigarlas, se incorporan pruebas automatizadas que validan autenticación (401), permisos por rol (403) y continuidad operativa mediante ingreso manual cuando el escáner falla; además, un paquete de puesta en marcha reproducible con README, .env.example, migraciones y seed (dataset de ejemplo) que permite levantar la demo en pocos pasos y con datos consistentes (login/roles, IN/OUT, alertas y Kardex). Se mantiene el alcance congelado (lo extra pasa a trabajo futuro) y se priorizan prototipos tempranos de cámara y autenticación bajo HTTPS, con revisión semanal de evidencias. Con estas medidas, el riesgo se mantiene acotado y el MVP es alcanzable y defendible dentro del semestre.* | |

1. **PARTE II**

|  |
| --- |
| **4. Objetivos** |
| En este apartado debes definir objetivos generales y específicos del Proyecto APT. Es importante aclarar que los objetivos se deben plantear en forma clara, concisa y sin dar mayores explicaciones, es decir, deben entenderse por sí solos. Se sugiere redactarlos utilizando un verbo en infinitivo, pues ello obliga a precisar acciones concretas. |

|  |  |
| --- | --- |
| Objetivo general | *Implementar una aplicación web progresiva (PWA), para el control de inventario en almacenes de barrio con registro por escaneo, alertas de reposición, reportes, autenticación básica y guía de uso con el fin de reducir quiebres de stock, agilizar la operación diaria y apoyar la toma de decisiones en microcomercios de baja adopción digital.* |
| Objetivos específicos | *Modelar e implementar en PostgreSQL el esquema de datos y las migraciones, garantizando la consistencia de stock a partir de los movimientos.*  *Implementar los módulos de Productos y Movimientos IN/OUT con validaciones de negocio y alertas de reposición basadas en stock mínimo.*  *Incorporar el escaneo de códigos de barras vía cámara con flujo mobile-first y fallback manual para asegurar la operación en condiciones reales.*  *Implementar autenticación y control de acceso por roles (Administrador/Operador) para resguardar el uso del sistema.*  *Generar el reporte por producto y rango de fechas, con saldo acumulado y exportación, y documentar la puesta en marcha en un README con dataset de ejemplo.* |

|  |
| --- |
| **5. Metodología** |
| En el siguiente apartado deberás describir la metodología, propia de tu disciplina, que utilizarás para resolver el proyecto APT antes descrito, incluyendo las etapas y métodos de trabajo. |

|  |
| --- |
| Descripción de la Metodología |
| *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. En cascada, el feedback llega tarde y los cambios son costosos; en cambio, el ágil permite iterar sobre un MVP acotado, inspeccionar resultados cada semana y adaptar el alcance sin romper el plan general.*  *Metodológicamente, se trabajará con sprints semanales y un backlog priorizado por valor, de modo que cada iteración entregue un incremento verificable (funcionalidad operativa + evidencia de prueba) y permita corregir rumbo a tiempo. Esta forma de trabajo pondera directamente en el cumplimiento del objetivo general —implementar una PWA para control de inventario con escaneo, alertas, informes y autenticación. Porque prioriza primero el núcleo funcional (Productos/Movimientos/Alertas), valida en terreno el flujo mobile-first (con fallback manual) y consolida la calidad mediante pruebas de autenticación (401), permisos (403) y consistencia de stock. Además, la práctica ágil de entregar evidencias incrementales (código, pruebas y demo reproducible) asegura trazabilidad y capacidad de ser defendido con evidencia.*  ***Tareas, Roles y Responsabilidades***   |  |  |  |  |  | | --- | --- | --- | --- | --- | | ***Integrante*** | ***Función / Rol*** | ***Tareas clave*** | ***Responsabilidades*** | ***Entregables asociados*** | | *Ariel Gárate* | *Gestión de proyecto (Scrum individual)* | *Planificar sprints, cronograma y hitos*  *Gestionar riesgos y cambios de alcance*  *Coordinar avances y revisiones con docente* | *Mantener backlog priorizado y alcance congelado (MVP)*  *Monitorear tiempos y cumplimiento de Definition of Done*  *Asegurar trazabilidad de decisiones y evidencias* | *Plan de trabajo + Cronograma/Gantt*  *Registro de riesgos y cambios*  *Bitácora de sprint reviews* | | *Ariel Gárate* | *Análisis de requisitos* | *Levantar Historias de Usuario (HU) y Criterios de Aceptación (CA)*  *Priorizar HU por valor/viabilidad (MVP)* | *Mantener HU/CA actualizados y verificables*  *Asegurar alineación con objetivos del proyecto* | *Backlog HU + CA (MVP)*  *Matriz de priorización (si aplica)* | | *Ariel Gárate* | *Diseño UX/UI (mobile-first)* | *Prototipar pantallas clave para móvil y escritorio*  *Definir flujos cortos y accesibles* | *Garantizar legibilidad, contraste y tamaños táctiles*  *Asegurar ingreso manual como fallback del escáner* | *Prototipos de UI*  *Guía mínima de estilos/Componentes* | | *Ariel Gárate* | *Modelamiento de datos* | *Diseñar DER y diccionario de datos*  *Definir reglas de integridad (stock ≥ 0, orden estable)* | *Evitar inconsistencias (no negativo; reconciliación Kardex)*  *Versionar migraciones y semilla* | *DER + Diccionario*  *Migraciones iniciales (SQL/ORM)* | | *Ariel Gárate* | *Backend / API* | *Implementar endpoints de Productos y Movimientos*  *Aplicar validaciones de negocio y errores controlados* | *Exponer documentación (OpenAPI/README)*  *Respetar transacciones y consistencia de stock* | *API funcional (Productos/Movimientos)*  *Docs de endpoints (OpenAPI/README)* | | *Ariel Gárate* | *Frontend / PWA* | *Construir vistas: Login, Productos, Movimientos, Alertas, Kardex*  *Configurar manifest y service worker (app-shell/cache)* | *Garantizar performance y accesibilidad en móvil*  *Asegurar funcionamiento offline básico* | *PWA instalable (manifest + SW)*  *Vistas operativas y responsive* | | *Ariel Gárate* | *Integración de cámara (escáner)* | *Integrar librería de códigos de barras (ZXing/Quagga)*  *Gestionar permisos y errores de cámara* | *Probar bajo HTTPS en Android/iOS*  *Habilitar fallback manual de código/SKU* | *Flujo de escaneo validado*  *Guía breve de uso/permiso de cámara* | | *Ariel Gárate* | *Seguridad (autenticación y roles)* | *Implementar login y control de acceso (Admin/Operador)*  *Proteger endpoints y sesiones (expiración)* | *Verificar respuestas 401/403*  *Resguardar credenciales/secretos (.env)* | *Autenticación y roles operativos*  *Pruebas 401/403 documentadas* | | *Ariel Gárate* | *Reporte Kardex* | *Construir consulta con saldo acumulado por producto/rango*  *Implementar exportación (CSV/PDF)* | *Asegurar exactitud del saldo y orden estable*  *Validar con datos de semilla y casos límite* | *Vista/endpoint Kardex*  *Archivo exportable (CSV/PDF)* | | *Ariel Gárate* | *Alertas de reposición* | *Implementar lógica de stock mínimo*  *Mostrar listado priorizado de reposición* | *Validar antes/después de movimientos*  *Mantener parámetros configurables* | *Alertas funcionales*  *Evidencias de casos (activación/desactivación)* | | *Ariel Gárate* | *Aseguramiento de calidad* | *Diseñar y ejecutar pruebas funcionales y de consistencia*  *Registrar resultados (capturas/JSON/SQL)* | *Mantener suite de pruebas repetible*  *Corregir defectos y documentar cambios* | *Carpeta /docs/evidencias/*  *Checklist de DoD por HU* | | *Ariel Gárate* | *Despliegue y documentación* | *Preparar README y .env.example*  *Configurar migraciones/seed y demo (URL o guía local)* | *Garantizar reproducibilidad en pocos pasos*  *Verificar HTTPS y credenciales demo* | *Repositorio + README + .env.example*  *Demo y video de 3–5 minutos* | |

|  |
| --- |
| **6. Evidencias** |
| A continuación, describe qué evidencias serán evaluadas en el informe de avance y en el informe final de tu proyecto APT. Estas evidencias deben ser acordadas con tu docente. Se entenderá por evidencia los productos que se desarrollen durante el proyecto y cuyo propósito sea visibilizar o documentar cómo se ha implementado el trabajo. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de evidencia**  **(avance o final)** | **Nombre de la evidencia** | **Descripción** | **Justificación** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **7. Plan de Trabajo** |
| En la siguiente tabla define la planificación de tu Proyecto APT de acuerdo a lo requerido. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Plan de Trabajo Proyecto APT** | | | | | | |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable | Observaciones |
| Gestión de proyectos informáticos | Plan de trabajo, cronograma y riesgos | Definir hitos, entregables, dependencias y mitigaciones. | Plantilla plan, calendario, rúbrica curso | 6 h (Sem 1) | Ariel Gárate | Ajustar con feedback docente; congelar alcance MVP. |
| Levantamiento / análisis de req. | Backlog HU + Criterios de Aceptación | Redactar HU de Productos, Movimientos, Alertas, Roles, Kardex, Escáner, IA ligera. | Editor/Docs, pauta HU/CA | 8 h (Sem 1) | Ariel Gárate | Máx. 7 HU para no sobredimensionar. |
| Gestión de la información | DER + Diccionario de datos | Diseñar entidades, claves y reglas (stock no negativo; orden por occurred\_at, id). | dbdiagram/draw.io | 8 h (Sem 1–2) | Ariel Gárate | Revisar con casos límite (OUT mayor a saldo). |
| Desarrollo / integración de sistemas | Entorno, migraciones y seed | Preparar proyecto, migraciones y datos de ejemplo (activar alertas). | Node.js, PostgreSQL, Prisma/Knex | 6 h (Sem 2) | Ariel Gárate | Dataset con 2–3 productos y movimientos. |
| Desarrollo / integración de sistemas | Módulo Productos (API + UI) | CRUD con validaciones (SKU único, min\_stock ≥ 0). | Express/React | 8 h (Sem 2) | Ariel Gárate | Considerar búsqueda/paginación si hay tiempo. |
| Desarrollo / integración de sistemas | Módulo Movimientos IN/OUT | Endpoints y UI para entradas/salidas con motivo/fecha; regla de saldo. | Express/React | 10 h (Sem 2–3) | Ariel Gárate | Rechazar OUT que deja saldo < 0. |
| Integración de TI | Escaneo de código por cámara | Integrar getUserMedia + librería (ZXing/Quagga) con fallback manual. | React PWA + lib barcode | 6 h (Sem 3) | Ariel Gárate | Probar en Android/iOS; HTTPS requerido. |
| Gestión de la información | Alertas de reposición | Lista dinámica de productos con saldo < min\_stock. | SQL/ORM + UI | 4 h (Sem 3) | Ariel Gárate | Validar antes/después de movimientos. |
| Seguridad de sistemas | Autenticación y roles | Login, JWT/sesión, permisos Admin/Operador en API y UI. | Auth lib, middleware | 8 h (Sem 3–4) | Ariel Gárate | Tests 401/403 y expiración de sesión. |
| Gestión de la información | Reporte Kardex + exportación | Consulta con saldo acumulado por producto y rango; export CSV. | SQL (ventanas), UI | 8 h (Sem 4) | Ariel Gárate | Orden por occurred\_at, id para evitar empates. |
| Aseguramiento de la calidad | Pruebas funcionales y consistencia + evidencias | Pruebas HU (roles, IN/OUT, alertas, escáner, Kardex) y carpeta /docs/evidencias/. | Jest/Playwright, SQL | 10 h (Sem 4) | Ariel Gárate | Guardar capturas, JSON y consultas verificadoras. |
| Gestión de proyectos / despliegue | README + .env.example + demo | Guía instalación/uso, variables, scripts (migrar/seed/run) y URL o guía local. | GitHub, Docs | 6 h (Sem 4) | Ariel Gárate | Demo reproducible en <10 min; credenciales de prueba. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Función / Unidad | Actividad / Tarea | Descripción | Recursos | Duración | Responsable | Observaciones |
| Gestión de proyectos informáticos | Plan de trabajo, cronograma y riesgos | Definir hitos, entregables, dependencias y mitigaciones. | Plantilla plan, calendario, rúbrica curso | 6 h (Sem 1) | Ariel Gárate | Ajustar con feedback docente; congelar alcance MVP. |
| Levantamiento / análisis de req. | Backlog HU + Criterios de Aceptación | Redactar HU de Productos, Movimientos, Alertas, Roles, Kardex, Escáner, IA ligera. | Editor/Docs, pauta HU/CA | 8 h (Sem 1) | Ariel Gárate | Máx. 7 HU para no sobredimensionar. |
| Gestión de la información | DER + Diccionario de datos | Diseñar entidades, claves y reglas (stock no negativo; orden por occurred\_at, id). | dbdiagram/draw.io | 8 h (Sem 1–2) | Ariel Gárate | Revisar con casos límite (OUT mayor a saldo). |
| Desarrollo / integración de sistemas | Entorno, migraciones y seed | Preparar proyecto, migraciones y datos de ejemplo (activar alertas). | Node.js, PostgreSQL, Prisma/Knex | 6 h (Sem 2) | Ariel Gárate | Dataset con 2–3 productos y movimientos. |
| Desarrollo / integración de sistemas | Módulo Productos (API + UI) | CRUD con validaciones (SKU único, min\_stock ≥ 0). | Express/React | 8 h (Sem 2) | Ariel Gárate | Considerar búsqueda/paginación si hay tiempo. |
| Desarrollo / integración de sistemas | Módulo Movimientos IN/OUT | Endpoints y UI para entradas/salidas con motivo/fecha; regla de saldo. | Express/React | 10 h (Sem 2–3) | Ariel Gárate | Rechazar OUT que deja saldo < 0. |
| Integración de TI | Escaneo de código por cámara | Integrar getUserMedia + librería (ZXing/Quagga) con fallback manual. | React PWA + lib barcode | 6 h (Sem 3) | Ariel Gárate | Probar en Android/iOS; HTTPS requerido. |
| Gestión de la información | Alertas de reposición | Lista dinámica de productos con saldo < min\_stock. | SQL/ORM + UI | 4 h (Sem 3) | Ariel Gárate | Validar antes/después de movimientos. |
| Seguridad de sistemas | Autenticación y roles | Login, JWT/sesión, permisos Admin/Operador en API y UI. | Auth lib, middleware | 8 h (Sem 3–4) | Ariel Gárate | Tests 401/403 y expiración de sesión. |
| Gestión de la información | Reporte Kardex + exportación | Consulta con saldo acumulado por producto y rango; export CSV. | SQL (ventanas), UI | 8 h (Sem 4) | Ariel Gárate | Orden por occurred\_at, id para evitar empates. |
| Aseguramiento de la calidad | Pruebas funcionales y consistencia + evidencias | Pruebas HU (roles, IN/OUT, alertas, escáner, Kardex) y carpeta /docs/evidencias/. | Jest/Playwright, SQL | 10 h (Sem 4) | Ariel Gárate | Guardar capturas, JSON y consultas verificadoras. |
| Gestión de proyectos / despliegue | README + .env.example + demo | Guía instalación/uso, variables, scripts (migrar/seed/run) y URL o guía local. | GitHub, Docs | 6 h (Sem 4) | Ariel Gárate | Demo reproducible en < 10 min; credenciales de prueba. |

|  |
| --- |
| **8. Carta Gantt** |
| Busca un formato de Carta Gantt que te acomode y organiza en este las actividades planificadas en el punto anterior considerando el periodo asignado para el desarrollo de tu Proyecto APT. Debes mantener la temporalidad del periodo académico en el desarrollo de las tres fases que contempla la Asignatura de Portafolio de Título. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividad** | **S01** | **S02** | **S03** | **S04** | **S05** | **S06** | **S07** | **S08** | **S09** | **S10** | **S11** | **S12** | **S13** | **S14** | **S15** | **S16** | **S17** | **S18** |
| Plan de trabajo y riesgos | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| HU + Criterios de Aceptación (MVP) | ■ | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| DER + Diccionario de datos (PostgreSQL) |  | ■ | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prototipos UI (PWA) |  |  | ■ | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Migraciones y dataset semilla |  |  |  |  | ■ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Módulo Productos (API + UI) |  |  |  |  | ■ | ■ |  |  |  |  |  |  |  |  |  |  |  |  |
| Movimientos IN/OUT (reglas de stock) |  |  |  |  |  | ■ | ■ | ■ |  |  |  |  |  |  |  |  |  |  |
| Alertas de reposición |  |  |  |  |  |  | ■ | ■ |  |  |  |  |  |  |  |  |  |  |
| Autenticación y roles (Admin/Operador) |  |  |  |  |  |  |  | ■ | ■ | ■ |  |  |  |  |  |  |  |  |
| Escáner por cámara (barcode + fallback) |  |  |  |  |  |  |  |  | ■ | ■ |  |  |  |  |  |  |  |  |
| Reporte Kardex + exportación |  |  |  |  |  |  |  |  |  | ■ | ■ |  |  |  |  |  |  |  |
| IA ligera (sugerencias) |  |  |  |  |  |  |  |  |  |  | ■ | ■ |  |  |  |  |  |  |
| Pruebas funcionales/consistencia – Ciclo 1 |  |  |  |  |  |  |  |  |  |  |  | ■ | ■ |  |  |  |  |  |
| Correcciones y hardening |  |  |  |  |  |  |  |  |  |  |  |  | ■ | ■ |  |  |  |  |
| Pruebas E2E + cierre de evidencias |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ | ■ |  |  |  |
| README + .env.example + Deploy + Dataset |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ |  |  |
| Video demo + ensayo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ |  |
| Entrega final |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ■ |

**Resumen (ES)**

Este proyecto propone un MVP de aplicación web progresiva (PWA) para almacenes de barrio con baja adopción digital. Aborda la gestión ineficiente del inventario realizada en planillas o registros manuales, que genera quiebres de stock, sobreinventario y uso poco eficiente del tiempo. La solución incluye módulos de Productos, Movimientos (entradas/salidas con motivo y fecha), Alertas de reposición, Usuarios/Roles y Reporte Kardex, además de escaneo de códigos de barras mediante la cámara del dispositivo y un componente de IA ligera que sugiere reposiciones/rotaciones a partir de datos del propio negocio.

La metodología es iterativa–incremental, con énfasis en modelamiento de datos (PostgreSQL), desarrollo (API en Node.js + PWA en React), pruebas funcionales y de consistencia (roles, secuencias IN/OUT, Kardex) y despliegue con guía de instalación y dataset de ejemplo. Se espera reducir quiebres, mejorar decisiones y acelerar registros sin exigir alta alfabetización tecnológica. Como trabajo futuro se considera la vinculación proveedores–clientes y la automatización de compras. El resultado buscado es una demo funcional defendible académicamente, con evidencias y documentación completas.

**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.

**Individual 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.

**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.