}

**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 | **Nicolas Caviedes, Sergio Plaza, Yohan Cortez, Nestor Barrera** |
| Rut | **21.020.117-3, 18.783.391-4,** |
| Carrera | **Ingeniería Informática** |
| Sede | **Viña del Mar** |

|  |
| --- |
| **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 | **Stockify** |
| Área (s) de desempeño(s) | **Desarrollo de software empresarial y analítica operativa aplicada al sector gastronómico y de entretenimiento nocturno *(p. ej., discotecas, bares y restobares)*.** |
| Competencias | * **Levantamiento y diseño de soluciones orientadas a la captura y consolidación de datos transaccionales (ventas, mermas, movimientos de stock).** * **Modelamiento y gestión de datos: calidad de datos y ETL/ELT desde diversas fuentes (APIs, archivos/planillas).** * **Analítica aplicada: métricas operativas (rotación, merma, quiebres), discrepancias venta esperada vs. real y pronóstico de consumo/reposición.** * **Desarrollo y validación de un MVP con buenas prácticas (versionado, pruebas, documentación) y visualización de resultados (dashboard/alertas).** |

|  |
| --- |
| **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 | **Relevancia del proyecto**  En el sector gastronómico y de entretenimiento nocturno (p. ej., discotecas, bares y restobares) la gestión de inventario suele ser manual y fragmentada, lo que genera quiebres, sobrestock y mermas. Además, la demanda es altamente variable (eventos, temporada, clima), y las decisiones de reposición se toman de forma reactiva. Stockify aborda este problema centralizando datos operacionales y anticipando el consumo para mejorar continuidad de servicio y margen.  **Descripción del proyecto**  Desarrollo de un MVP que:   * Capta y consolida datos operacionales (ventas, movimientos, mermas) desde fuentes existentes (exportaciones de POS/planillas, APIs cuando existan). * Estandariza y valida la calidad de datos; calcula métricas clave (rotación, merma, quiebres, venta esperada vs. real). * Genera pronóstico de consumo y reposición y emite alertas (riesgo de quiebre, reposición sugerida). * Entrega un dashboard operativo para decisiones diarias y revisión por local/sucursal.   **Relación con intereses profesionales**  Alineado con el interés en backend/data y construcción de productos B2B: integra diseño de servicios, analítica aplicada y despliegue de un MVP utilizable por clientes, fortaleciendo experiencia en productos de base de datos y decisiones.  **Factibilidad de desarrollo del Proyecto APT** Abordable en un semestre con dedicación semanal de la asignatura. Requiere un entorno de desarrollo estándar, servicios cloud de nivel gratuito y datasets iniciales (histórico de ventas/consumo; si no existen, se generan datasets sintéticos coherentes). Riesgos: **datos insuficientes** y **validación en terreno**; mitigación: piloto con 1–2 locales, iteraciones cortas y ajuste de modelo. |
| Descripción del Proyecto APT | Se espera **desarrollar un MVP** que **prediga la demanda y optimice la reposición** para **reducir mermas y quiebres** y, con ello, **mejorar la rentabilidad del inventario** en operaciones del **sector nocturno y de entretenimiento** (p. ej., discotecas, bares y restobares). El sistema **captará y consolidará** datos operacionales (ventas, movimientos y mermas) desde **exportaciones** o **APIs** cuando existan, los **depurará y normalizará**, calculará **KPIs clave** (rotación, cobertura, quiebres y diferencia entre venta esperada y real) y mostrará la información en un **dashboard** con **alertas** y **sugerencias de reposición**; el **horizonte de pronóstico** será **ajustable** al **lead time** y al ciclo real de reabastecimiento del local.  La problemática se abordará con una **metodología ágil por sprints**, construyendo incrementos funcionales y validándolos en un **piloto de 1–2 locales**. El equipo se organiza en: **Project Manager** (planificación y coordinación), **Líder/Arquitectura & Datos** (modelo de datos, KPIs, decisiones técnicas) y **dos desarrolladores full-stack** (funcionalidades “de base de datos a pantalla”). Se prioriza partir sin integraciones en vivo (ingesta por exportaciones) y, si faltan históricos, se utilizarán **datasets sintéticos coherentes** para pruebas. Se controlará **calidad de datos ≥ 95%** y **diferencia promedio ≤ 25%** en el piloto como criterios de éxito del semestre. |
| Pertinencia del proyecto con el perfil de egreso | **Stockify** es pertinente al perfil de egreso de **Ingeniería en Informática** porque demuestra, en un caso real del sector nocturno y de entretenimiento, las capacidades que el egresado y/o integrantes que debe dominar. La relación es directa en las siguientes dimensiones:   * **Ingeniería de software (requisitos → diseño → construcción → pruebas):** levantamiento de necesidades, diseño de arquitectura modular, desarrollo del **MVP** y validación funcional del flujo de captura–consolidación–reglas de negocio. * **Gestión y gobierno de datos:** modelado lógico, diccionario de datos, reglas de calidad (completitud/consistencia), y un **pipeline ETL/ELT** para integrar ventas, movimientos y mermas desde fuentes heterogéneas. * **Analítica y apoyo a la decisión:** definición de **KPIs operativos** (rotación, merma, quiebre, desvío esperado vs. real) y construcción de un **pronóstico de consumo/reposición** con evaluación de errores (p. ej., MAE/MAPE) para dar trazabilidad a las recomendaciones. * **Gestión de proyectos TI y metodologías ágiles:** planificación iterativa, control de riesgos, seguimiento por hitos y evidencias; integración con la **carta Gantt** del proyecto y entregables incrementales demostrables. * **Calidad, seguridad y ética profesional:** pruebas unitarias/integración, control básico de accesos, bitácora de cambios y resguardo de información sensible del negocio. * **Comunicación técnica y documentación:** manual breve de uso, guía técnica del MVP y presentación ejecutiva de resultados para facilitar adopción por parte de usuarios no técnicos. |
| Relación con los intereses profesionales | Como equipo, nuestro foco es **backend y datos**, **analítica operativa** y **productos B2B para PYMEs**. **Stockify** encaja porque nos exige construir un **MVP** que **capture y consolide** ventas/mermas, calcule **indicadores clave** y genere **pronóstico de consumo/reposición** para el **sector nocturno y de entretenimiento** (discotecas, bares, restobares).  **Aporte a nuestro desarrollo:**   * **Roles complementarios:** backend, modelado/ingesta de datos, analítica, QA/gestión. * **Ingeniería aplicada:** diseño de dominio, pipeline de datos, métricas y alertas, entregas iterativas. * **Validación realista:** piloto acotado y dashboard para decisiones diarias medibles. |
| Factibilidad de desarrollo del Proyecto APT | El alcance está acotado a un MVP de captura–consolidación–análisis con pronóstico básico, entregado en hitos cortos y con herramientas disponibles (gratuitas o académicas).  **(1) Duración del semestre** Plan en fases alineadas al calendario (semanas):   * Descubrimiento y diseño de dominio. * Ingesta y **calidad de datos** (CSV/Excel/APIs cuando existan). * Cálculo de **indicadores clave** y primer **pronóstico de consumo/reposición**. * Dashboard, alertas y validación con un **piloto acotado**.   **(2) Horas asignadas a la asignatura** Las horas de cátedra/laboratorio permiten cubrir los hitos previstos. Se complementa con **bloques adicionales planificados** según carta Gantt (priorizando entregables demostrables).  **(3) Materiales requeridos**   * Entorno de desarrollo (IDE) y **control de versiones (Git)**. * **Base de datos liviana** (p. ej., relacional) para consolidar ventas/mermas/movimientos. * Scripts para **ingesta y validación** (ETL/ELT) desde archivos/exportaciones. * Cálculo de métricas y **modelo de pronóstico clásico** a predicciones. * **Dashboard** web/simple para visualización y alertas. * (Opcional) Cloud en nivel gratuito para despliegue del MVP.   **(4) Factores externos que facilitan**   * **Alcance preciso**: foco en el **sector nocturno y de entretenimiento** (p. ej., discotecas, bares, restobares). * **MVP sin integraciones en vivo** en la primera iteración (ingesta por exportaciones). * **Equipo con roles complementarios** (backend/datos/analítica/QA). * Posibilidad de partir por **top SKUs** y **1–2 locales** para acelerar validación.   **(5) Factores externos que dificultan + mitigación**   * **Datos incompletos/heterogéneos** → reglas de calidad, diccionario de datos y trazabilidad de transformaciones. * **Poca historia para pronóstico** → iniciar con heurísticas y modelos simples; ampliar gradualmente con más semanas de datos. * **Alta variabilidad por eventos** → marcar eventos (promos/fechas especiales) en el dataset para ajustar el modelo. * **Tiempo acotado** → **timeboxing**, priorización de entregables y demos quincenales. * **Acceso a datos reales** → uso de **datasets sintéticos coherentes** y validación mixta (real + simulado) mientras se habilita el piloto. |

1. **PARTE II**

|  |
| --- |
| **4. Objetivos** |
| Desarrollar un MVP que prediga la demanda y optimice la reposición, reduciendo mermas y quiebres para disminuir costos y aumentar ingresos por disponibilidad oportuna. |

|  |  |
| --- | --- |
| Objetivo general | Desarrollar un MVP que prediga la demanda y optimice la reposición para reducir mermas y quiebres y mejorar la rentabilidad del inventario en operaciones del sector nocturno y de entretenimiento. |
| Objetivos específicos | **Los siguientes objetivos específicos operacionalizan el objetivo general; traducen la meta en tareas concretas, medibles y verificables durante el semestre, orientando el plan de trabajo y los entregables.**  **- Diseñar el modelo de datos** y **construir el flujo de trabajo** de captura y limpieza, asegurando **≥ 95%** de registros válidos.  **- Calcular y publicar KPIs operativos** (rotación, merma, quiebres, cobertura y diferencia entre venta esperada y real) en un **dashboard** actualizado diariamente.  **- Generar pronósticos de consumo/reposición** para el **top 20 SKUs** con **horizonte referencial de 7 días**, **ajustable** según el **ciclo real de reabastecimiento** y el **lead time del local**, logrando **diferencia promedio ≤ 25%** respecto de las ventas reales en el piloto.  **- Implementar alertas tempranas** de riesgo de quiebre y **sugerencias de reposición** con **≥ 48 horas** de anticipación.  **- Desplegar el MVP** en **1–2 locales piloto**, **recoger feedback** de usuarios y **ejecutar al menos una iteración de mejora**. |

|  |
| --- |
| **5. Metodología** |
| Trabajaremos con **un enfoque ágil e incremental**: iteraciones cortas, entregables demostrables y ajustes sobre evidencia. El objetivo es construir un MVP que capture, depure y consolide datos operacionales para calcular KPIs, generar pronósticos y alertas útiles para la decisión diaria**.** |

|  |
| --- |
| Descripción de la Metodología |
| **Abordaje del problema** Atacaremos la gestión ineficiente de inventario en el **sector nocturno y de entretenimiento** (p. ej., discotecas, bares y restobares) con una solución **data-driven**. El foco es: capturar datos operacionales (ventas, movimientos y mermas), **depurarlos**, convertirlos en **KPIs accionables** y generar **pronósticos y alertas** que anticipen quiebres y sobrestock.  **Metodología seleccionada** Trabajaremos con un **enfoque ágil e incremental por sprints**, entregando incrementos funcionales y ajustando sobre evidencia (build–measure–learn). Las funcionalidades se desarrollan en **vertical slices** (de punta a punta: datos → backend → UI), para reducir dependencias y entregar valor visible en cada iteración.   1. **Descubrimiento y diseño de dominio**: mapeo breve de procesos de inventario, definición de **KPIs** y criterios de éxito; glosario y **modelo conceptua** 2. **Datos – captura y calidad**: diseño del **modelo de datos** y **flujo de trabajo** de ingesta desde exportaciones (CSV/Excel; APIs cuando existan); reglas de **calidad** (completitud, consistencia, duplicados) y **trazabilidad** de transformaciones. Si faltan históricos, se generan **datasets sintéticos** coherentes para pruebas. 3. **Producto – KPIs y visualización**: servicios para consolidar datos y cálculo de indicadores (rotación, merma, quiebres, cobertura, desvío esperado–real); **dashboard operativo** con actualización diaria. 4. **Pronóstico y alertas**: implementación de un **método de pronóstico simple y explicable**; horizonte **referencial de 7 días**, **ajustable** al **ciclo real de reabastecimiento** y **lead time** del local; **alertas tempranas** y **sugerencias de reposición**. 5. **Piloto y mejora**: despliegue controlado en **1–2 locales**, pruebas con usuarios, recolección de feedback y **al menos una iteración de mejora** sobre problemas priorizados; documentación técnica y de usuario.   **Funciones, tareas y responsabilidades (equipo)**   * **Nicolás Caviedes** — Project Manager: planificación por sprints, priorización del backlog, coordinación con stakeholders/piloto, gestión de riesgos, definición de criterios de aceptación y seguimiento de hitos. * **Sergio Plaza** — Líder de Proyecto / Arquitecto & Datos: arquitectura y estándares, modelo de datos y KPIs, lineamientos de calidad, decisiones técnicas, code reviews, soporte a CI/CD y validación de umbrales/alertas. * **Yojan Cortes** — Desarrollador Full-Stack: desarrollo end-to-end por feature (modelo/persistencia, API y UI), ingesta/limpieza básica de datos para su feature, pruebas y documentación breve. * **Néstor Barrera** — Desarrollador Full-Stack: desarrollo end-to-end por feature (persistencia, API y UI), diseño/UX del dashboard, pruebas y documentación breve.   **Dinámica y control (versión clara)**   * **Trabajo por sprints semanales.** Reunión de planificación al inicio; ejecución de tareas; **demo corta** y retro al cierre. * **Backlog visible y priorizado.** Tablero (Jira/Trello/GitHub) con tareas, responsable y **criterios de aceptación** por cada una. * **Paralelo y sin dependencias innecesarias.** Yojan y Néstor toman funcionalidades completas **“de la base de datos a la pantalla”** (BD → API → UI), para mostrar valor real en cada sprint. * **Revisión de código cruzada.** Todo cambio pasa por **Pull Request** y **code review** entre los desarrolladores; Sergio valida decisiones técnicas. * **Definición de “Terminado” (DoD).** Datos validados, lógica implementada, pruebas básicas OK, pantalla funcionando, criterios de aceptación cumplidos y **documentación breve**. * **Control de versiones y CI/CD.** Ramas por funcionalidad (feature/\*), main estable; compilación y pruebas automáticas; despliegue del **MVP** a un ambiente de pruebas. * **Indicadores de avance.** % de tareas completadas por sprint, **calidad de datos ≥ 95%**, tiempos de carga del dashboard, **diferencia promedio ≤ 25%** en pronóstico del piloto. * **Gestión de riesgos.** Matriz simple (riesgo, señal de alerta, plan de mitigación), con foco en: falta de datos, cambios de alcance y tiempos. * **Evidencias.** Actas de sprint, tablero actualizado, scripts de ingesta/limpieza, capturas del dashboard, logs de ejecución y reporte breve de validación. * **Comunicación.** Canales único (WhatsApp/Discord/Correos) y reunión corta semanal con responsables y acuerdos por escrito. |

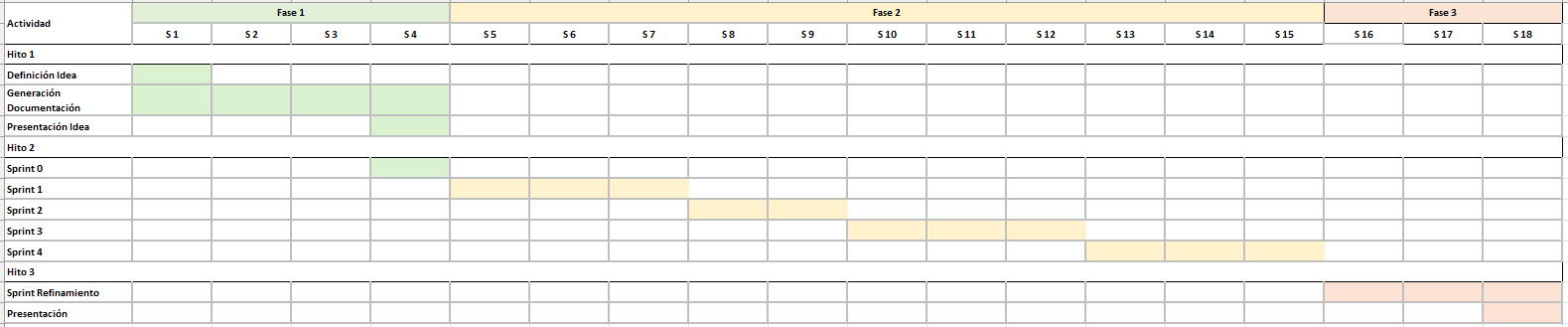
|  |
| --- |
| **6. Evidencias** |
| A continuación, describe qué evidencias serán evaluadas en el informe de avance y en el informe final de tu proyecto APT. Estas evidencias deben ser acordadas con tu docente. Se entenderá por evidencia los productos que se desarrollen durante el proyecto y cuyo propósito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

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de evidencia**  **(avance o final)** | **Nombre de la evidencia** | **Descripción** | **Justificación** |
| **Avance** | **Prototipo navegable (mockups + flujo)** | Diseño de pantallas en Figma con flujo básico de inventario, reportes y predicción. | Permite validar el alcance, la usabilidad y los criterios de aceptación antes de implementar. |
| **Avance** | **Arquitectura & Modelo de Datos** | Diagrama de componentes de infraestructura (OCI + GCP) y modelo lógico de datos (ventas, movimientos, mermas). | Define la base técnica para un desarrollo estructurado y mantenible. |
| **Avance** | **Flujo de trabajo de datos + Reporte de calidad** | Proceso de ingesta desde CSV/Excel con reporte de validación (≥95% registros válidos). | Garantiza que los datos sean confiables antes de generar KPIs o predicciones. |
| **Final** | **Dashboard con KPIs y reportes** | Interfaz con métricas clave (rotación, merma, quiebres, cobertura y diferencia esperada-real). | Soporta la toma de decisiones diaria y la trazabilidad del inventario. |
| **Final** | **Informe de pronóstico y recomendaciones** | Documento con método de predicción, resultados del piloto y sugerencias de reposición por SKU. | Demuestra el valor práctico del proyecto (menos quiebres/sobrestock, guía de acción para negocios). |
| **Final** | **Documentación y manual de usuario** | Guía técnica de instalación y manual breve para uso del MVP. | Facilita adopción del sistema por usuarios no técnicos y asegura transferencia del conocimiento. |

|  |
| --- |
| **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[[1]](#footnote-1) | Observaciones |
| *Gestión de proyectos / Ingeniería de software* | *Planificación Sprint 0 y backlog* | *Definir alcance, criterios de aceptación, priorización y riesgos iniciales.* | *Jira/Trello, Instagantt, Google Drive* | *1 semana (S1)* | *Nicolas* | |  | | --- | |  |   *Base para evitar desvíos de alcance* |
| *Arquitectura / Modelamiento de datos* | *Modelo de datos y diccionario* | *Diseñar entidades (ventas, movimientos, mermas) y reglas de integridad; glosario.* | *Draw.io, DB designer, Docs* | *1–2 semanas (S1–S2)* | *Sergio* | *Se ajusta según fuentes reales/sintéticas.* |
| *UX/UI* | *Mockups y flujo de pantallas* | *Maquetas de inventario, movimientos, dashboard y navegación.* | *Figma/Excalidraw* | *1 semana (S1)* | *Nestor* | *Validación temprana con el equipo.* |
| *DevOps / Buenas prácticas* | *Repositorios y CI/CD* | *Creación de repos, ramas, PR/reviews y pipeline de build/pruebas.* | *GitHub/GitLab,* | *1 semana (S1–S2)* | *Yojan* | |  | | --- | |  |  |  | | --- | | *Asegura integraciones estables.* | |
| *Full-Stack (backend+frontend)* | *CRUD de Productos* | |  | | --- | |  |  |  | | --- | | *Persistencia, API y vista de gestión de productos.* | | *IDE, Java (o stack definido), DB* | *2 semanas (S3–S4)* | *Yojan* | *Endpoints versionados y pruebas básicas.* |
| *Full-Stack (backend+frontend)* | *Movimientos y Mermas* | *Persistencia, API y vista de movimientos/mermas con validaciones.* | *IDE, DB, Postman/Insomnia* | *2 semanas (S3–S4)* | *Nestor* | *Reglas de negocio revisadas por Sergio.* |
| *Calidad de datos / Gobierno* | *Flujo de captura y limpieza* | *Ingesta desde CSV/Excel/APIs; reglas de completitud/consistencia;* ***reporte de calidad ≥95%****.* | *Python/ETL ligero, Sheets/Excel/Spring Batch* | *2 semanas (S3–S5)* | ***Sergio*** *(apoyo Yojan)* | *Sin datos íntegros no avanzan KPIs/pronóstico.* |
| *Analítica operativa / Visualización* | *Dashboard inicial + KPIs* | *Cálculo y visualización de rotación, cobertura, quiebres y desvío esperado–real.* | *DB, consultas, Frontend* | *2 semanas (S5–S6)* | *Nestor* | *Actualización diaria; métricas acordadas.* |
| *Validación / Gestión de interesados* | *Piloto y feedback* | *Despliegue controlado, sesión de prueba, levantamiento de hallazgos y mejoras.* | *Ambiente de pruebas, encuestas* | *1–2 semanas (S10–S11)* | ***Nicolás*** *(coord.), equipo* | *Ejecutar al menos* ***1 iteración*** *de mejora.* |
| *QA / Documentación* | *Pruebas y correcciones* | |  | | --- | |  |  |  | | --- | | *Pruebas funcionales y de integración; estabilización técnica.* | | *Plan de pruebas, Logs* | *1 semana (S11)* | ***Yojan*** *y* ***Néstor*** | |  | | --- | |  |  |  | | --- | | *Cerrar issues críticos*  *la demo final.* | |
| *Documentación / Comunicación* | *Manual de usuario, guía técnica y presentación* | *Manual breve, guía técnica de instalación/uso y PPT final.* | *Docs, PPT* | *1 semana (S12)* | |  | | --- | |  |  |  | | --- | | ***Néstor*** *(manual),*  ***Sergio*** *(guía),*  ***Nicolás*** *(PPT)* | | *Evidencias finales para evaluación.* |

|  |
| --- |
| **8. Carta Gantt** |
| La carta Gantt estructura el proyecto en tres fases: **Fase 1** con definición de idea, arquitectura/modelo de datos, mockups y setup de repositorios y CI/CD; **Fase 2** con desarrollo incremental por sprints (BD conectada, backend CRUD de productos y frontend login+listado; luego dashboard con reportes y alertas; después procesamiento batch a cloud storage y módulo de predicción; y finalmente estabilización); y **Fase 3** con documentación, presentación final y cierre. Cada sprint termina con demo y evidencias en el repositorio, con responsables asignados (PM, Líder/Arquitectura & Datos y dos desarrolladores full-stack) y dependencias claras para asegurar avance medible. |



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)