}

1. **PARTE I**

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

| **Guía Estudiante - Definición Proyecto APT**  **Fase 1** | http://www.duoc.cl/normasgraficas/normasgraficas/marca-duoc/6logo-fondo-transparente/fondo-transparente.png |
| --- | --- |

| Nombre estudiante | **Keila Nathaly Martinez Acosta** |
| --- | --- |
| Rut | **26.017.234-4** |
| Carrera | **Ingenieria en Informatica** |
| Sede | **Padre Alonso Ovalle** |

| Nombre estudiante | **LuisManuel García de la huerta Paredes** |
| --- | --- |
| Rut | **20.219.930-5** |
| Carrera | **Ingenieria en Informatica** |
| Sede | **Padre Alonso 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 | **Ticket Fast!** |
| --- | --- |
| Área (s) de desempeño(s) | Desarrollo de software, gestión de soporte técnico, administración de sistemas e infraestructura IT |
| Competencias | * Diseñar e implementar soluciones informáticas que integren tecnologías emergentes. * Aplicar herramientas de inteligencia artificial para apoyar la toma de decisiones. * Gestionar incidentes y recursos tecnológicos de forma eficiente. |

| **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 | En el contexto del hotel, la gestión de incidencias técnicas actualmente se realiza de manera informal (llamadas, mensajes, avisos verbales), lo que genera pérdida de trazabilidad, duplicación de problemas y tiempos de respuesta elevados. Un sistema de tickets inteligente permitirá centralizar la gestión, reducir tiempos de atención y mejorar la comunicación entre áreas, aportando mayor eficiencia y calidad en los procesos de soporte técnico. |
| --- | --- |
| Descripción del Proyecto APT | Se desarrollará un sistema de tickets orientado a la gestión de incidencias técnicas dentro del hotel. La plataforma permitirá registrar, asignar y dar seguimiento a cada incidencia, garantizando trazabilidad y comunicación fluida entre usuarios y personal de soporte.  se integrará un módulo de **Inteligencia Artificial** la siguiente funcionalidad:   1. **Priorización y derivación inteligente**: asignar un nivel de criticidad a cada ticket en el dashboard y derivarlo automáticamente al técnico, asegurando que problemas de mayor impacto (ej.: caída de red) sean atendidos antes que otros de menor urgencia (ej.: falla en proyector), la IA detecta el ticket prioritario y lo pondrá de los primeros.   Este enfoque permitirá optimizar los tiempos de respuesta, evitar duplicación de esfuerzos y ofrecer un soporte más eficiente y profesional dentro de la organización. |
| Pertinencia del proyecto con el perfil de egreso | El proyecto demanda competencias propias del perfil de egreso, tales como diseñar soluciones informáticas, implementar sistemas que respondan a necesidades organizacionales, aplicar técnicas de aseguramiento de calidad y utilizar herramientas de inteligencia artificial para la optimización de procesos. De esta forma, se articula la formación académica con una necesidad real del entorno laboral. |
| Relación con los intereses profesionales | El proyecto se encuentra alineado con la experiencia laboral de **Luis** en el área de soporte técnico dentro del hotel, lo que aporta una visión práctica de las incidencias más frecuentes. Asimismo, integra los intereses de **Keila** en la gestión de soluciones tecnológicas, orientando el proyecto hacia la innovación y la optimización de procesos. De esta manera, ambos buscamos fortalecer nuestras habilidades en desarrollo de software, aseguramiento de calidad y aplicación de inteligencia artificial para la automatización de procesos. |
| Factibilidad de desarrollo del Proyecto APT | El sistema se desarrollará con tecnologías open source, utilizando **Framework full-stack:** **Next.js (App Router) + TypeScript**  UI + API en el mismo proyecto (menos piezas, cero CORS, fácil de mantener).  **Base de datos/Auth/Storage:** **Supabase (PostgreSQL + Auth + Storage + RLS)**  DB gestionada, autenticación lista, políticas de seguridad a nivel de fila.  **ORM/Migraciones:** **Drizzle ORM**  Tipado fuerte y migraciones claras (genera SQL para Supabase).  **UI:** **TailwindCSS**  Construyes pantallas rápido y limpias.  **IA (MVP):**   1. **Sugeridor local** (cosine similarity sobre histórico de tickets) — sin depender de APIs externas. 2. (Opcional) integrar **Edge Function** en Supabase o un **endpoint** en Next.js que use un proveedor (cuando el profe lo pida), lo que asegura estabilidad, escalabilidad y facilidad de mantenimiento. El módulo de inteligencia artificial se implementará en **Python (scikit-learn/TensorFlow)**, empleando un dataset simulado de incidencias técnicas que permitirá entrenar un modelo básico de priorización y sugerencia de soluciones.   Consideramos que el proyecto es factible dentro del semestre, ya que las tecnologías seleccionadas son accesibles, cuentan con amplia documentación y facilitan un desarrollo modular con entregables parciales. Los principales riesgos se relacionan con la gestión del tiempo y la coordinación del equipo; sin embargo, estos serán mitigados mediante una planificación semanal estructurada en jira, la distribución clara de tareas entre los integrantes y el uso de herramientas colaborativas de gestión. |

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 | Desarrollar un sistema de tickets de soporte inteligente para el hotel, que centralice la gestión de incidencias técnicas y utilice IA para sugerir soluciones y priorizar casos. |
| --- | --- |
| Objetivos específicos | Implementar un sistema de tickets de soporte TI que permita registrar, gestionar y dar seguimiento a las incidencias de manera eficiente, mejorando la atención al usuario y optimizando los tiempos de resolución.   1. **Diseñar el modelo de datos relacional** de incidencias, usuarios y roles en **Supabase (PostgreSQL)**, implementando **migraciones controladas con Drizzle ORM** para asegurar consistencia y trazabilidad. 2. **Implementar un sistema de autenticación y autorización** en **Supabase**, aplicando **Row Level Security (RLS)** para garantizar control de acceso según el rol (usuario, técnico, administrador).      1. **Desarrollar el módulo CRUD de tickets** en **Next.js (App Router + TypeScript)** que permita registrar, editar, asignar y cerrar incidencias. 2. **Construir un panel de control (dashboard)** en **Next.js + TailwindCSS** que muestre métricas en tiempo real (tickets abiertos, cerrados, tiempos de resolución). 3. **Integrar notificaciones automáticas** (correo y/o in-app) para eventos clave del ciclo de vida del ticket, implementadas en la misma API de **Next.js**. 4. **Incorporar un módulo de IA** con un sugeridor local basado en **cosine similarity** para recomendar soluciones y un clasificador de prioridad según impacto y urgencia. 5. **Ejecutar pruebas funcionales y de integración** con herramientas como **Postman y Playwright**, asegurando calidad y cumplimiento de requerimientos. 6. **Preparar despliegue en la nube** utilizando **Vercel para frontend/API** y **Supabase Cloud para la base de datos**, acompañado de documentación técnica (ERD, endpoints, manual de usuario). |

| **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 aplicará un enfoque iterativo dividido en etapas: Requerimientos y Análisis  * **Objetivo principal:** Diseñar y desarrollar un **sistema inteligente** para gestionar incidencias técnicas en el hotel, mejorando la trazabilidad, priorización y tiempos de resolución. * **Alcance funcional:**    + Registro, asignación, seguimiento y cierre de tickets.   + Priorización automática de incidencias.   + Roles: usuario, técnico y administrador.   + Panel de métricas y dashboard de rendimiento. * **Análisis inicial:**    + **Entrevistas y recopilación de incidencias reales del hotel** para obtener datos concretos.   + Identificación de procesos actuales, flujos de soporte y problemas comunes.   + Definición de categorías, urgencias y niveles de impacto.   + Priorización de requerimientos basada en criticidad y valor para el negocio.  Diseño  * **Arquitectura del sistema:**    + **Framework Full-stack:** **Next.js** (App Router + TypeScript).   + **UI + API integrados** en un mismo proyecto → menos piezas, más fácil de mantener.   + **Supabase (PostgreSQL)**: base de datos, autenticación, almacenamiento y seguridad con **RLS**.   + **Drizzle ORM:** migraciones, consultas tipadas y control de versiones de la base de datos.   + **TailwindCSS:** diseño ágil, responsivo y limpio. * **Diagramas:**    + **Casos de uso:** flujos entre usuarios, técnicos y sistema.   + **Modelo entidad-relación (ERD):** perfiles, tickets, comentarios e historial.   + **Arquitectura de componentes:** cliente, API interna, base de datos y módulo IA.  Desarrollo  * **Backend:**    + Next.js API Routes integradas (no backend separado).   + Control de roles, seguridad y autenticación mediante Supabase.   + ORM Drizzle para operaciones CRUD con migraciones controladas. * **Frontend:**    + UI responsive con TailwindCSS.   + Panel de administración, formularios de ticket y dashboard con métricas. * **Base de datos:**    + PostgreSQL gestionado con Supabase.   + Migraciones versionadas y controladas.   + Políticas **RLS** para proteger datos según rol y propietario.   Integración de IA (MVP)   * **Objetivo:** Optimizar la gestión de incidencias con funcionalidades “inteligentes” sin complejidad innecesaria. * **Fases:**    + **Clasificación automática de prioridad**:      - Algoritmo basado en **urgencia + impacto**.   + **Sugerencia de soluciones rápidas**:      - Buscar incidencias similares en el histórico con **cosine similarity**.   + **Dataset inicial**:      - Incidencias reales recolectadas + casos simulados para enriquecer datos. * **Evolución futura (opcional):**    + Integrar **Supabase Edge Functions** para modelos más complejos.   + O un endpoint en Next.js usando **TensorFlow.js** o **scikit-learn**.   Pruebas   * **QA funcional:**    + Verificar que cada módulo cumpla los criterios de aceptación.   + Pruebas unitarias y de integración. * **Pruebas de usabilidad:**    + Validar con personal del hotel.   + Ajustar formularios, flujos y notificaciones según retroalimentación. * **Herramientas:**    + **Postman**: probar API interna.   + **Supabase Studio**: verificación de datos y seguridad.   + **Playwright / Cypress** (opcional): automatizar pruebas de UI.   Entrega Final   * **Despliegue:**    + Contenedores **Docker** para Next.js y Supabase (solo si es requerido).   + Alternativa rápida: **Vercel** para frontend/API y **Supabase Cloud** para BDD. * **Documentación:**    + Arquitectura, diagramas, endpoints, modelo de datos.   + Guía rápida para instalación y despliegue. * **Presentación:**    + Demo en vivo del sistema.   + Dashboard con métricas de tickets abiertos/cerrados.   + Evidencias de pruebas y funcionamiento del modelo IA. |

| **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** | **Documento de requerimientos** | **Contiene casos de uso, roles, categorías y flujos definidos.** | **Asegura claridad en el alcance funcional del sistema.** |
| **avance** | **Prototipo de interfaz (Figma)** | **Bocetos de pantallas principales: login, registro de ticket, dashboard.** | **Permite validar diseño y usabilidad antes de programar.** |
| **avance** | **Modelo de datos en Supabase Migraciones Drizzle** | **Estructura inicial de tablas con relaciones (tickets, usuarios, roles, historial).** | **Evidencia la base técnica del sistema.** |
| **avance** | **Módulo CRUD de tickets (Next.js)** | **Registro, actualización y cierre de incidencias.** | **Marca el inicio del desarrollo funcional del software.** |
| **final** | **Dashboard implementado (Next.js TailwindCSS)** | **Visualización en tiempo real de métricas de tickets.** | **Permite evaluar el cumplimiento del objetivo de monitoreo.** |
| **final** | **Integración IA (cosine similarity)** | **Prototipo funcional que sugiere soluciones y prioriza incidencias.** | **Demuestra la incorporación de inteligencia artificial en el sistema.** |
| **final** | **Pruebas documentadas (unitarias y de integración)** | **Evidencia de validación de API, UI y flujo completo.** | **Garantiza la calidad del sistema desarrollado.** |
| **Final** | **Repositorio en GitHub despliegue (Vercel Supabase)** | **Código fuente versionado y sistema accesible en la nube.** | **Entrega tangible y verificable del producto final.** |
| **Final** | **Manual de usuario y documentación técnica** | **Guía de instalación, uso del sistema, ERD y endpoints.** | **Asegura que el software pueda ser utilizado y mantenido.** |

| **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-0) | Observaciones |
| **Diseñar e implementar soluciones informáticas** | **Levantamiento de requerimientos** | **Identificación de roles, casos de uso, categorías de tickets y flujos actuales.** | **Laptop, Google Docs, entrevistas, Draw.io** | **2 semanas** | **Luis** | **Puede retrasarse si no se obtiene toda la información del hotel.** |
| **Diseñar interfaces de usuario** | **Prototipo en Figma** | **Creación de mockups de login, registro de tickets y dashboard.** | **Laptop, Figma** | **1 semana** | **Keyla** | **Validar con docente a iniciar desarrollo.**   |  | | --- |  |  | | --- | |
| **Implementar bases de datos relacionales** | **Definición de modelo de datos + migraciones** | **Creación de tablas en Supabase y migraciones con Drizzle ORM.** | **Supabase, Drizzle ORM, laptop** | **1 semana** | **Keyla** | **Revisar integridad referencial.** |
| **Programar sistemas de información** | **Desarrollo de módulo CRUD de tickets** | **Implementación de registro, edición, asignación y cierre de incidencias.** | **Next.js, Supabase, VS Code** | **2 semanas** | **Luis** | **Requiere validación de reglas de negocio.** |
| **Desarrollar software con enfoque modular** | **Implementación de autenticación y roles** | **Configuración de Supabase Auth y políticas RLS.** | **Supabase, Next.js, laptop** | **1 semana** | **Luis / Keyla** | **Validar seguridad con pruebas de acceso.** |
| **Analizar datos y aplicar IA** | **Integración módulo IA (cosine similarity)** | **Desarrollo de recomendador de soluciones y clasificador de prioridad.** | **Python, Next.js API, dataset simulado** | **2 semanas** | **Luis / Keyla** | **Se parte con MVP, puede escalar a TensorFlow** |
| **Construir interfaces responsivas** | **Desarrollo del dashboard en TailwindCSS** | **Programación de métricas y visualización en tiempo real.** | **Next.js, TailwindCSS** | **2 semanas** | **Luis / Keyla** | **Ajustar diseño según retroalimentación,** |
| **Asegurar la calidad de software** | **Pruebas unitarias y de integración** | **Validación de API, UI y flujos con Postman y Playwright.** | **Postman, Playwright, Supabase Studio** | **2 semanas** | **final** | **Pruebas de usabilidad.** |
| **Documentar proyectos de software** | **Elaboración de documentación técnica y manual de usuario** | **Guía de instalación, endpoints, ERD, casos de uso y tutorial de uso.** | **Google Docs, Draw.io, GitHub** | **1 SEMANA** | **final** | **Entrega junto con despliegue final.** |
| **Implementar y desplegar software** | **Deploy en Vercel + Supabase Cloud** | **Publicación de frontend/API en Vercel y BDD en Supabase.** | **Vercel, Supabase, GitHub** | **1 semana** | **final** | **Verificar conexión y credenciales.** |

| **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** | **Fase 1** | | | | **Fase 2** | | | | | | | | | | | | **Fase 3** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S 1** | **S 2** | **S 3** | **S 4** | **S 5** | **S 6** | **S 7** | **S 8** | **S 9** | **S 10** | **S 11** | **S 12** | **S 13** | **S 14** | **S 15** | **S 16** | | **S 17** | **S 18** |
| **Levantamiento de requerimientos** | **x** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Prototipo en Figma** |  | **x** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Modelo de datos + Migraciones (Supabase + Drizzle)** |  |  |  |  | **x** | **x** |  |  |  |  |  |  |  |  |  |  | |  |  |
| **Módulo CRUD de tickets (Next.js)** |  |  |  |  |  |  | **x** | **x** |  |  |  |  |  |  |  |  | |  |  |
| **Autenticación y roles (Supabase Auth + RLS)** |  |  |  |  |  |  |  | **x** | **x** |  |  |  |  |  |  |  | |  |  |
| **Dashboard (Next.js + TailwindCSS)** |  |  |  |  |  |  |  |  |  | **x** | **x** |  |  |  |  |  | |  |  |
| **Integración IA (cosine similarity)** |  |  |  |  |  |  |  |  |  |  |  | **x** | **x** |  |  |  | |  |  |
| **Pruebas (unitarias, integración, usabilidad)** |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** | **x** |  | |  |  |
| **Documentación técnica + Manual de usuario** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** | | **x** |  |
| **Deploy en Vercel + Supabase Cloud** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **x** | **x** |

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-0)