}

1. **PARTE I**

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

| Nombre estudiante | **BENJAMIN BARRERA / DIEGO INOSTROZA / JUAN SANTANDER** |
| --- | --- |
| Rut | **21004192-3/20819771-1 / 20913200-1** |
| Carrera | **INGENIERÍA EN INFORMÁTICA** |
| Sede | **VESPUCIO** |

| **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 | **Sistema de Ticketera Web** |
| --- | --- |
| Área (s) de desempeño(s) | *Desarrollo de Software, Gestión de Sistemas de Información, Administración de Bases de Datos.* |
| Competencias | *Construir modelos de datos para soportar los requerimientos de la organización de acuerdo a un diseño definido y escalable en el tiempo .*  *Desarrollar una solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos .*  *Gestionar proyectos informáticos, ofreciendo alternativas para la toma de decisiones de acuerdo a los requerimientos de la organización . (Aplicada en la planificación y desarrollo iterativo del proyecto).*  *Realizar pruebas de certificación tanto de los productos como de los procesos utilizando buenas prácticas definidas por la industria . (Aplicada mediante pruebas funcionales continuas durante el desarrollo para asegurar la calidad y el cumplimiento de requisitos).* |

| **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 | * *Este proyecto aborda la necesidad fundamental en muchas organizaciones, especialmente pequeñas y medianas, de gestionar eficientemente las solicitudes de soporte técnico o incidencias (tickets). Actualmente, muchas dependen de métodos poco estructurados como correos electrónicos o planillas, lo que conduce a la pérdida de información, tiempos de respuesta ineficientes, falta de seguimiento y dificultad para medir el desempeño del área de soporte . Este sistema de ticketera web centraliza la recepción, asignación y seguimiento de todas las solicitudes, mejorando la organización y la trazabilidad.* * *Su relevancia en el campo laboral de [Nombre de tu Carrera] es alta, ya que la gestión eficiente de soporte técnico es crucial en cualquier departamento de TI o área de servicio al cliente . La implementación de este sistema permite optimizar flujos de trabajo, reducir tiempos de inactividad por problemas técnicos no resueltos, mejorar la comunicación entre usuarios y técnicos, y generar métricas valiosas para la toma de decisiones (como tiempos de respuesta, resolución, cumplimiento de SLA, etc.) .* * *El contexto de aplicación es amplio, pudiendo ser cualquier organización que necesite gestionar solicitudes internas o externas de soporte. Impacta directamente tanto a los usuarios finales (empleados, clientes) que requieren asistencia, como al equipo de soporte técnico que debe atender dichas solicitudes . El principal aporte de valor es la mejora significativa en la eficiencia operativa del soporte, la transparencia en el seguimiento de casos y la capacidad de análisis del rendimiento del servicio .* |
| --- | --- |
| Descripción del Proyecto APT | *El objetivo principal del proyecto es desarrollar una aplicación web completa para la gestión integral de tickets de soporte. El sistema permite a los usuarios finales registrar incidencias detallando el problema, seleccionando la categoría (ej. Hardware, Software) y la prioridad (Baja, Media, Alta, Crítica), e incluso adjuntar archivos relevantes.*  *Los técnicos (con roles diferenciados Nivel 1 y Nivel 2) pueden visualizar los tickets, asignárselos (o ser asignados automáticamente), actualizar su estado (Abierto, En Proceso, Cerrado), comunicarse con el usuario a través de un sistema de comentarios dentro del ticket, y registrar la solución.*  *Características clave implementadas incluyen:*   * *Autenticación segura de usuarios con contraseñas hasheadas.* * *Roles de usuario (Usuario, Técnico Nivel 1, Técnico Nivel 2) con permisos específicos.* * *Sistema de creación, visualización y gestión de tickets.* * *Asignación automática de tickets a Técnicos Nivel 1 mediante un sistema round-robin.* * *Funcionalidad para que Técnicos Nivel 1 escalen tickets a Nivel 2.* * *Funcionalidad para que Técnicos Nivel 2 reasignen tickets.* * *Sistema de comentarios por ticket con notificaciones.* * *Bloqueo de comentarios en tickets cerrados.* * *Gestión de Categorías con definición de SLAs (tiempos de respuesta/resolución).* * *Cálculo y visualización del vencimiento del SLA por ticket y cálculo de porcentaje de cumplimiento en el dashboard.* * *Gestión de Inventario de activos (equipos) y su asignación a usuarios.* * *Base de Conocimiento (FAQ) gestionable por técnicos y consultable por usuarios, con función de búsqueda.* * *Sistema de Notificaciones internas para usuarios sobre cambios en sus tickets (asignación, cambio de estado, nuevos comentarios).* * *Dashboards con estadísticas clave para usuarios y técnicos (incluyendo gráficos dinámicos).* * *Reportes visuales (gráficos) para técnicos sobre rendimiento y carga de trabajo.* * *Filtros y paginación en todas las listas principales para manejar grandes volúmenes de datos.* * *Interfaz web desarrollada con Flask (Python) y Bootstrap, utilizando una base de datos PostgreSQL.* * *Preparación para despliegue en producción (variables de entorno, servidor WSGI, logging).* |
| Pertinencia del proyecto con el perfil de egreso | *Este proyecto se alinea directamente con el perfil de egreso de [Nombre de tu Carrera] al integrar múltiples competencias esenciales .*   * ***Construir modelos de datos :*** *Se diseñó e implementó un modelo relacional en PostgreSQL con tablas para usuarios, tickets, categorías, activos, comentarios, adjuntos, notificaciones y artículos, estableciendo relaciones coherentes y considerando la escalabilidad .* * ***Desarrollar una solución de software :*** *Se construyó una aplicación web completa utilizando el framework Flask (Python) para el backend, implementando la lógica de negocio (roles, permisos, asignación, SLAs, notificaciones, comentarios), la interacción con la base de datos mediante SQLAlchemy (ORM), y la creación de una API RESTful implícita a través de las rutas . Se desarrolló el frontend con HTML, CSS (Bootstrap) y JavaScript para crear interfaces interactivas y responsivas . Se aplicaron técnicas de desarrollo estructurado y modular.* * ***Gestionar proyectos informáticos :*** *El desarrollo siguió un enfoque iterativo, planificando funcionalidades por etapas (conexión BD, login, CRUDs básicos, roles, módulos adicionales, mejoras UX, producción), simulando un ciclo de gestión de proyecto ágil .* * ***Realizar pruebas de certificación :*** *Durante todo el desarrollo, se realizaron pruebas funcionales manuales para validar que cada componente funcionara según lo esperado , asegurando la calidad del producto final.* |
| Relación con los intereses profesionales | *Este proyecto encaja perfectamente con mis intereses en el* ***desarrollo web full-stack*** *y la* ***creación de aplicaciones orientadas a procesos de negocio*** *. Me atrae especialmente la posibilidad de diseñar y construir sistemas que resuelvan problemas reales y mejoren la eficiencia dentro de una organización. El desarrollo de la ticketera me permitió profundizar en:*   * ***Backend con Python y Flask:*** *Fortaleciendo mis habilidades en la creación de APIs web, manejo de sesiones, lógica de negocio compleja y seguridad (hashing, roles).* * ***Frontend con Bootstrap y JavaScript:*** *Mejorando mi capacidad para crear interfaces de usuario limpias, responsivas e interactivas.* * ***Bases de Datos con PostgreSQL y SQLAlchemy:*** *Ganando experiencia en modelado de datos relacional, consultas complejas y el uso de un ORM.* * ***Arquitectura de Software:*** *Comprendiendo cómo integrar diferentes componentes (frontend, backend, base de datos) en una aplicación cohesiva.* * ***Prácticas de Desarrollo Modernas:*** *Utilizando herramientas como Git/GitHub para control de versiones y preparándome para el despliegue en producción. Realizar este proyecto ha sido fundamental para mi desarrollo profesional , ya que me ha dado experiencia práctica en el ciclo completo de vida del software, desde la concepción hasta la preparación para el despliegue, aplicando directamente las competencias de mi perfil de egreso en un producto tangible y relevante para el mercado laboral .* |
| Factibilidad de desarrollo del Proyecto APT | *El proyecto es completamente factible de realizarse en el marco de la asignatura Capstone. Se consideraron los siguientes puntos:*   * ***Tiempo:*** *El desarrollo se planificó y ejecutó de manera incremental a lo largo del semestre, permitiendo completar las funcionalidades principales y las mejoras propuestas dentro del plazo establecido por la asignatura .* * ***Recursos y Materiales:*** *Se utilizaron exclusivamente herramientas de software de código abierto o gratuitas (Python, Flask, PostgreSQL, VS Code, Git) y recursos de aprendizaje en línea (documentación oficial, tutoriales), sin requerir hardware especializado ni licencias costosas .* * ***Factores Externos:*** *El principal facilitador fue la disponibilidad de abundante documentación y comunidades de soporte para las tecnologías elegidas. Las dificultades encontradas (errores de código, problemas de configuración, depuración) fueron superables mediante investigación, pruebas y el enfoque iterativo, permitiendo corregir errores sobre la marcha sin detener el progreso general . La naturaleza modular del desarrollo también ayudó a aislar y resolver problemas específicos.* |

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 e implementar una aplicación web funcional para la gestión centralizada de tickets de soporte técnico, que permita a usuarios registrar incidencias y a técnicos administrarlas eficientemente, mejorando los tiempos de respuesta y la trazabilidad de las soluciones* |
| --- | --- |
| Objetivos específicos | *Diseñar y construir una base de datos relacional (PostgreSQL) capaz de almacenar de forma segura y estructurada la información de usuarios, tickets, categorías, activos, comentarios y notificaciones.*  *Implementar un sistema de autenticación seguro basado en roles (Usuario, Técnico Nivel 1, Técnico Nivel 2) con control de acceso diferenciado a las funcionalidades de la aplicación.*  *Desarrollar las funcionalidades CRUD (Crear, Leer, Actualizar, Borrar) para la gestión de Tickets, Categorías, Inventario (Activos) y Artículos de la Base de Conocimiento (FAQ).*  *Implementar un sistema de asignación automática de tickets a Técnicos Nivel 1 mediante un algoritmo round-robin.*  *Desarrollar funcionalidades para la escalada y reasignación de tickets entre técnicos.*  *Crear un sistema de comentarios internos por ticket para facilitar la comunicación entre usuarios y técnicos.*  *Implementar un sistema de notificaciones para informar a los usuarios sobre cambios relevantes en sus tickets.*  *Desarrollar dashboards visuales con estadísticas clave y reportes gráficos para usuarios y técnicos.*  *Implementar filtros funcionales y paginación en las vistas de listas para mejorar la usabilidad.*  *Asegurar la aplicación mediante hashing de contraseñas y preparación para un entorno de producción (variables de entorno, servidor WSGI, logging).* |

| **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 |
| --- |
| ***Descripción de la Metodología:*** *El proyecto se desarrolló siguiendo una* ***metodología ágil e iterativa****, enfocada en la entrega incremental de funcionalidades. Se adoptó un enfoque similar a Scrum, dividiendo el trabajo en fases con objetivos claros, aunque sin ceremonias formales debido a la naturaleza individual del desarrollo.*   * ***Fase 1: Planificación y Diseño Inicial (Semanas 1-2):*** *Se definieron los requisitos iniciales, se diseñó el modelo de datos relacional (entidad-relación), se seleccionaron las tecnologías (Python, Flask, PostgreSQL, SQLAlchemy, Bootstrap) y se configuró el entorno de desarrollo y el repositorio Git. Se crearon las maquetas iniciales (frontend estático).* * ***Fase 2: Desarrollo del Núcleo (Semanas 3-8):*** *Se implementó la conexión a la base de datos, el sistema de autenticación (login/logout) con hashing de contraseñas, la gestión básica de sesiones y los modelos de datos iniciales (Usuario, Ticket). Se desarrolló el CRUD básico para tickets (creación, visualización) y se implementaron los roles de usuario iniciales.* * ***Fase 3: Desarrollo de Módulos (Semanas 9-13):*** *Se añadieron los módulos CRUD para Categorías, Inventario (Activos) y Base de Conocimiento (FAQ). Se implementó la lógica de asignación automática (round-robin), escalada y reasignación de tickets. Se desarrolló el sistema de comentarios y notificaciones.* * ***Fase 4: Integración y Mejoras (Semanas 14-16):*** *Se implementaron los dashboards con gráficos dinámicos (Chart.js), los filtros funcionales y la paginación en las vistas de listas. Se integró el cálculo y visualización de SLAs y la funcionalidad de adjuntar archivos. Se realizaron pruebas funcionales exhaustivas y se refinó la interfaz de usuario.* * ***Fase 5: Preparación para Producción y Documentación (Semanas 17-18):*** *Se configuró la aplicación para usar variables de entorno (.env),* |

| **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** |
| --- | --- | --- | --- |
| **Tipo de evidencia (avance o final)** | **Nombre de la evidencia** | ***Descripción*** | **Justificación** |
| **Avance (Ej: Semana 8)** | **Repositorio Git (Hito 1)** | **Versión del código fuente con la base de datos conectada, login funcional y CRUD básico de tickets implementado.** | **Demuestra la correcta configuración del backend, la base de datos y las funcionalidades esenciales del sistema.** |
| **Avance (Ej: Semana 13)** | **Prototipo Funcional (Hito 2)** | **Aplicación web con los módulos CRUD completos (Categorías, Inventario, FAQ), sistema de roles, asignación y comentarios funcionando.** | **Muestra el progreso significativo en la implementación de la mayoría de las funcionalidades requeridas.** |
| **Final** | **Código Fuente Completo** | **Repositorio Git finalizado en GitHub ([Enlace a tu repositorio]) con todo el código fuente (Python, HTML, CSS, JS), archivos de configuración (.env, .gitignore, requirements.txt) y script de datos de prueba (seed.py).** | **Es el producto principal y tangible del proyecto, que permite evaluar la calidad del código, la arquitectura y la implementación de todas las características.** |
| **Final** | **Aplicación Funcional** | **Demostración en vivo de la aplicación ejecutándose localmente (o desplegada, si aplica).** | **Permite verificar el funcionamiento real de todas las características, la interfaz de usuario y la interacción entre los diferentes roles.** |
| **Final** | **Informe Final APT** | **Este documento completo, siguiendo el formato y requisitos de la asignatura.** | **Cumple con la entrega formal requerida, detallando la definición, planificación, desarrollo y resultados del proyecto.** |
| **Final** | **Presentación Final** | **Diapositivas y/o demostración en vivo resumiendo el proyecto, los desafíos, las soluciones y los aprendizajes.** | **Permite comunicar de forma efectiva el alcance y los logros del proyecto.** |

| **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 |
| *Nombra las competencias o unidades de competencias que se relacionan con las diferentes actividades requeridas para el desarrollo de la actividad.* | *Señale el nombre de la tarea o actividad.* | *Describe la tarea o actividad.* | *Nombra los recursos necesarios para llevar a cabo las actividades definidas.* | *Escribe la duración de actividades o tarea.* | *Escribe el nombre del integrante del equipo responsable de la actividad y tareas asociadas.* | *Escribe las dificultades o facilitadores que se podrían presentar durante la ejecución de cada una de las actividades propuestas para llevar a cabo el plan de trabajo.* |
| ***Competencia(s) Relacionada(s)*** | ***Nombre de Actividades/Tareas*** | ***Descripción Actividades/Tareas*** | ***Recursos*** | ***Duración Estimada*** | ***Responsable*** | ***Observaciones*** |
| *Construir modelos de datos, Desarrollar solución de software* | ***Fase 1: Configuración y Base*** | *Instalación, diseño BD, estructura Flask, conexión BD, login (hash), plantillas base, decoradores.* | *PC, Python 3.11+, Flask, SQLAlchemy, PostgreSQL, pgAdmin, Git, VS Code* | *3 Semanas* | *[Tu Nombre / Equipo]* | *Asegurar compatibilidad de versiones. Definir modelo de datos escalable.* |
| *Desarrollar solución de software, Construir modelos de datos* | ***Fase 2: CRUDs y Roles*** | *CRUD Tickets (con adjuntos y SLA), CRUD Categorías, Inventario, FAQ. Implementar roles T1/T2 y permisos.* | *Mismos recursos* | *4 Semanas* | *[Tu Nombre / Equipo]* | *Validar formularios. Manejar correctamente la subida y almacenamiento de archivos.* |
| *Desarrollar solución de software* | ***Fase 3: Lógica Avanzada*** | *Implementar asignación automática (round-robin), escalada/reasignación, comentarios, notificaciones.* | *Mismos recursos* | *4 Semanas* | *[Tu Nombre / Equipo]* | *La lógica round-robin requiere persistencia del último índice. Las notificaciones deben generarse en los puntos correctos del flujo.* |
| *Desarrollar solución de software, Realizar pruebas* | ***Fase 4: UX y Reportes*** | *Desarrollar dashboards (contadores, gráficos Chart.js), implementar filtros funcionales y paginación, autoformato RUT, pruebas integrales.* | *Mismos recursos + Chart.js* | *4 Semanas* | *[Tu Nombre / Equipo]* | *Optimizar consultas para filtros complejos. Asegurar que la paginación funcione correctamente con los filtros aplicados.* |
| *Gestionar proyectos informáticos* | ***Fase 5: Producción y Cierre*** | *Configurar variables de entorno (.env), servidor WSGI (run.py con Waitress), logging, crear script seed.py, redactar documentación, preparar presentación, subir a GitHub.* | *Mismos recursos + python-dotenv, waitress, Faker, Word, PowerPoint, GitHub* | *3 Semanas* | *[Tu Nombre / Equipo]* | *Asegurar que .gitignore esté correcto. Verificar que requirements.txt esté completo. Cumplir con formatos de entrega.* |

| **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** |
| **Fase 1: Config & Base** | **■** | **■** | **■** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Fase 2: CRUDs y Roles** |  |  |  | **■** | **■** | **■** | **■** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Fase 3: Lógica Avanzada** |  |  |  |  |  |  |  | **■** | **■** | **■** | **■** |  |  |  |  |  |  |  |  |
| **Fase 4: UX y Reportes** |  |  |  |  |  |  |  |  |  |  |  | **■** | **■** | **■** | **■** |  |  |  |  |
| **Fase 5: Producción y Cierre** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **■** | **■** | **■** |  |  |
| ***Entrega Final*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **■** |  |

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)