Un dibujo animado

El contenido generado por IA puede ser incorrecto.

**UNIVERSIDAD MEXIQUENSE DEL BICENTENARIO**

**UNIDAD DE ESTUDIOS SUPERIORES JIQUIPILCO**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**P R O Y E C T O**

**MESA DE AYUDA PARA LA DIRECCIÓN GENERAL DEL PERSONAL**

**QUE PARA OBTENER EL TITULO DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**PRESENTA**

**MARTINEZ MARTINEZ ANETTE**

**ORDOÑEZ JIMENEZ OSCIEL**

**RICO VALDEZ ELIAS EDUARDO**

**ASESORES DE PROYECTO**

**I.S.C MEDINA MORALES ELIZABETH**

**I.S.C. CHRISTOPHER DELGADILLO RAMIREZ**

**PERIODO**

**01 SEPTIEMBRE 2025 AL 30 ENERO 2026**

**ESTADO DE MÉXICO DIA, MES, 2025**

**Índice**

[Capítulo I. Introducción 4](#_Toc205815173)

[1.1. Planteamiento del problema 5](#_Toc205815174)

[1.2 Justificación del proyecto 6](#_Toc205815175)

[1.3 Objetivo general 7](#_Toc205815176)

[1.4 Objetivos específicos 7](#_Toc205815177)

[1.5 Alcances y limitaciones 8](#_Toc205815178)

[Capítulo II. Marco Teórico 9](#_Toc205815179)

[2.1 Fundamentos de ITIL y gestión de servicios de TI 10](#_Toc205815180)

[2.2 Conceptos clave: Mesa de Ayuda, Incidentes, Problemas, Cambios, Activos. 10](#_Toc205815181)

[2.3 Importancia del licenciamiento de software y gestión de equipos 11](#_Toc205815182)

[Capítulo III. Diagnóstico y análisis de la situación actual 12](#_Toc205815183)

[3.1 Infraestructura de TI actual en la Dirección General de Personal 13](#_Toc205815184)

[3.2 Procesos de soporte antes del proyecto 13](#_Toc205815185)

[3.3 Detección de necesidades críticas 14](#_Toc205815186)

[3.4 Requerimientos funcionales y no funcionales del sistema 14](#_Toc205815187)

[Capítulo IV. Diseño de la Mesa de Ayuda 15](#_Toc205815188)

[4.1 Arquitectura general del sistema 16](#_Toc205815189)

[4.1.1 Tecnología base 16](#_Toc205815190)

[4.1.2 Estructura de la arquitectura 16](#_Toc205815191)

[4.1.3 Comunicación entre capas 17](#_Toc205815192)

[4.1.4 Seguridad de la arquitectura 17](#_Toc205815193)

[4.2 Diagrama de flujo funcional de la Mesa de Ayuda 18](#_Toc205815194)

[4.3 Módulos críticos del sistema 18](#_Toc205815195)

[4.3.1 Gestión de Solicitudes de Servicio 18](#_Toc205815196)

[4.3.2 Gestión de Incidentes 18](#_Toc205815197)

[4.3.3 Gestión de Problemas 18](#_Toc205815198)

[4.3.4 Gestión de Cambios 18](#_Toc205815199)

[4.3.5 Gestión de Activos 18](#_Toc205815200)

[4.3.6 Módulo de Informes y Dashboards 18](#_Toc205815201)

[4.4 Módulos complementarios estratégicos 18](#_Toc205815202)

[4.4.1 Módulo de Resguardos y Cambios de Equipo 18](#_Toc205815203)

[4.4.2 Módulo de Licenciamiento Office 365 y Windows 19](#_Toc205815204)

[4.4.3 Gestión de Correos Electrónicos Institucionales 19](#_Toc205815205)

[4.5 Repositorio de configuración y conocimiento 19](#_Toc205815206)

[4.5.1 Base de Datos de Gestión de Configuración (CMDB) 19](#_Toc205815207)

[4.5.2 Base de Conocimientos 19](#_Toc205815208)

[4.6 Seguridad y control de acceso 19](#_Toc205815209)

[4.7 Mecanismos de autoservicio 19](#_Toc205815210)

[4.8 Plan de escalabilidad y soporte 19](#_Toc205815211)

[Capítulo V. Desarrollo e implementación 20](#_Toc205815212)

[5.1 Metodología de desarrollo utilizada (Scrum, Cascada, etc.) 21](#_Toc205815213)

[5.2 Cronograma de actividades 21](#_Toc205815214)

[5.3 Configuración del entorno de desarrollo y pruebas 21](#_Toc205815215)

[5.4 Implementación de módulos (con evidencias: pantallas, diagramas, código, pruebas unitarias, etc.) 21](#_Toc205815216)

[5.5 Pruebas funcionales y de usuario 21](#_Toc205815217)

[5.6 Integración con infraestructura existente 21](#_Toc205815218)

[Capítulo VI. Resultados y beneficios 22](#_Toc205815219)

[6.1 Comparativo antes/después 23](#_Toc205815220)

[6.2 Tiempo de respuesta y eficiencia del área 23](#_Toc205815221)

[6.3 Control de activos y licenciamiento 23](#_Toc205815222)

[6.4 Mejora en la gestión de cambios y soporte técnico 23](#_Toc205815223)

[6.5 Reducción de errores y documentación de incidencias 23](#_Toc205815224)

[Capítulo VII. Conclusiones y recomendaciones 24](#_Toc205815225)

[7.1 Conclusiones generales del proyecto 25](#_Toc205815226)

[7.2 Recomendaciones para mejoras futuras 25](#_Toc205815227)

[7.3 Sugerencias de escalabilidad 25](#_Toc205815228)

[Capítulo VII. Glosario de términos técnicos 26](#_Toc205815229)

[Capítulo IX. Bibliografías 27](#_Toc205815230)

[Capítulo X. 28](#_Toc205815231)

[Anexos 28](#_Toc205815232)

**Índice de figuras y tablas**

# Capítulo I. Introducción

## Planteamiento del problema

En la actualidad, muchas organizaciones enfrentan dificultades para gestionar de manera eficiente los incidentes, solicitudes y problemas técnicos que surgen en su operación diaria. La falta de un sistema centralizado para registrar, priorizar y dar seguimiento a las solicitudes de soporte provoca retrasos en la atención, pérdida de información, duplicación de esfuerzos y disminución de la satisfacción de los usuarios.

En muchos casos, el soporte técnico se brinda de forma reactiva, sin un historial claro de incidencias ni métricas que permitan identificar áreas de mejora. Esto limita la capacidad de las empresas para tomar decisiones informadas, optimizar recursos y garantizar la continuidad operativa.

La ausencia de una herramienta automatizada para la gestión de tickets y el seguimiento de incidentes también dificulta la comunicación entre usuarios y personal de soporte, lo que genera confusión, tiempos de resolución prolongados y posibles impactos en la productividad.

Además, la falta de procesos estandarizados y de un repositorio histórico de casos impide la generación de conocimiento organizacional que facilite la resolución de problemas recurrentes. Esto deriva en una dependencia excesiva de la experiencia individual del personal y en una menor capacidad de respuesta ante situaciones críticas.

Por lo tanto, es necesario desarrollar una mesa de ayuda que centralice la recepción, registro, clasificación, asignación y seguimiento de solicitudes, permitiendo una atención más ágil, transparente y eficiente, con reportes e indicadores que faciliten la toma de decisiones y mejoren la experiencia de los usuarios.

## 1.2 Justificación del proyecto

La implementación de una mesa de ayuda representa una solución estratégica para optimizar la gestión de soporte técnico dentro de una organización. Un sistema de este tipo no solo mejora la eficiencia en la atención de incidencias y solicitudes, sino que también establece un canal de comunicación claro y documentado entre los usuarios y el personal de soporte, evitando la pérdida de información y agilizando los tiempos de respuesta.

Contar con una herramienta que registre, clasifique y priorice los tickets permite garantizar un seguimiento adecuado de cada caso, lo que incrementa la satisfacción de los usuarios y reduce el impacto de las fallas en las operaciones diarias. Asimismo, la recopilación de datos y generación de reportes facilita la identificación de problemas recurrentes, el análisis de tendencias y la implementación de medidas preventivas.

Desde una perspectiva organizacional, el desarrollo de una mesa de ayuda contribuye a una gestión más ordenada de los recursos, minimiza costos derivados de ineficiencias y fortalece la calidad del servicio interno o externo que se ofrece. En un entorno donde la rapidez, la trazabilidad y la transparencia son factores clave, contar con este sistema se convierte en una necesidad que impacta directamente en la productividad y la competitividad de la organización.

Adicionalmente, una mesa de ayuda bien diseñada puede convertirse en una herramienta de mejora continua, ya que proporciona datos y métricas que permiten rediseñar procesos, capacitar al personal de manera focalizada y anticipar necesidades futuras. Esto no solo soluciona problemas existentes, sino que también previene la aparición de nuevos incidentes, consolidando un entorno de trabajo más eficiente y confiable.

## 1.3 Objetivo general

Desarrollar un sistema de mesa de ayuda que permita registrar, clasificar, dar seguimiento y resolver incidencias técnicas de usuarios dentro de área de la dirección General de Personal, a través de la comunicación con el personal de soporte técnico.

## 1.4 Objetivos específicos

* Analizar los requerimientos funcionales y no funcionales del sistema para asegurar su alineación con las necesidades de los usuarios.
* Diseñar una interfaz intuitiva y de fácil acceso para la gestión de tickets.
* Implementar un sistema de registro y seguimiento de incidencias que permita priorizar y asignar tareas de manera eficiente.
* Desarrollar módulos para la generación de reportes e indicadores de rendimiento.
* Integrar mecanismos de comunicación entre usuarios y personal de soporte para mejorar la atención y resolución de casos.
* Probar y validar el funcionamiento del sistema, garantizando su estabilidad y seguridad.

## 1.5 Alcances y limitaciones

**Alcances**

* Implementación de un sistema centralizado de registro y seguimiento de tickets.
* Acceso para usuarios y personal de soporte con diferentes niveles de permisos.
* Generación de reportes estadísticos para análisis de desempeño.
* Gestión de prioridades y estados de los tickets.
* Plataforma adaptable a diferentes dispositivos mediante diseño web responsivo.

**Limitaciones**

* El sistema no contempla, en su primera versión, integración con herramientas externas de gestión empresarial.
* La cobertura de soporte se limita a incidentes y solicitudes dentro del alcance definido por la organización.
* No se incluye un módulo de inteligencia artificial para diagnóstico automático en la fase inicial.
* La disponibilidad del sistema dependerá de la infraestructura tecnológica de la organización.

# Capítulo II. Marco Teórico

## 2.1 Fundamentos de ITIL y gestión de servicios de TI

La Mesa de Ayuda es una función clave dentro de la gestión de servicios de TI, diseñada para ser el punto único de contacto (Single Point of Contact - SPOC), entre los usuarios y el área técnica.  
Su función principal es garantizar que cualquier incidente, solicitud o consulta se gestione de manera rápida, trazable y estandarizada.

En este proyecto, la operación de la Mesa de Ayuda se alineará con los principios de ITIL, que incluyen:

* **Enfoque en el valor al cliente:** el soporte se orienta a la continuidad del trabajo de los usuarios, priorizando incidentes según su impacto en el negocio.
* **Optimización de procesos:** reducción de tiempos de resolución mediante procedimientos claros y herramientas automatizadas.
* **Mejora continua (Ciclo PDCA):** revisión constante de indicadores para implementar mejoras.

Además, se tomarán como referencia las siguientes normas ISO:

* **ISO/IEC 20000:** marco de gestión de servicios de TI que certifica la calidad en la prestación de soporte.
* **ISO/IEC 27001:** gestión de la seguridad de la información, especialmente importante en la manipulación de datos personales de empleados.
* **ISO 9001:** gestión de calidad, asegurando que el servicio cumpla con los estándares establecidos.

## 2.2 Conceptos clave: Mesa de Ayuda, Incidentes, Problemas, Cambios, Activos.

* **Mesa de Ayuda:** servicio centralizado para registrar y dar seguimiento a solicitudes e incidentes. Puede operar en niveles (1er nivel: atención inicial; 2do nivel: soporte especializado).
* **Incidentes:** interrupciones no planificadas o degradaciones en un servicio. Ejemplo: falla en la conexión a internet en un área de la oficina.
* **Problemas:** causas raíz de incidentes recurrentes. Ejemplo: la falla de conexión se debe a un switch de red defectuoso.
* **Cambios:** modificaciones controladas en la infraestructura o aplicaciones, evaluadas por un comité para evitar riesgos. Ejemplo: actualización masiva de un software de oficina.
* **Activos:** hardware, software, licencias, periféricos y cualquier recurso de TI que deba registrarse y administrarse.

La integración de estos conceptos en la Mesa de Ayuda permite establecer una **trazabilidad completa**: desde la primera notificación del usuario, pasando por el análisis técnico, hasta la resolución y documentación final.

## 2.3 Importancia del licenciamiento de software y gestión de equipos

Un inventario de software bien gestionado:

* Evita sanciones legales y auditorías negativas por uso de software no autorizado.
* Garantiza el acceso a actualizaciones y soporte oficial.
* Reduce vulnerabilidades de seguridad al evitar aplicaciones obsoletas.

La gestión de equipos en el hardware implica:

* **Registro detallado:** marca, modelo, fecha de adquisición, ubicación y usuario asignado.
* **Mantenimiento preventivo:** limpieza física, revisión de componentes, actualizaciones de firmware.
* **Mantenimiento correctivo:** reparación o sustitución de piezas defectuosas.
* **Baja de activos:** eliminación segura de datos y disposición adecuada del

# Capítulo III. Diagnóstico y análisis de la situación actual

## 3.1 Infraestructura de TI actual en la Dirección General de Personal

Actualmente se dispone de:

* **Equipos de usuario final:** estaciones de trabajo, laptops y periféricos distribuidos entre el personal administrativo y de recursos humanos.
* **Servidores internos:** gestionan correo institucional, nómina, base de datos de personal y almacenamiento compartido.
* **Red de comunicaciones:** cableado estructurado y puntos de acceso Wi-Fi, con routers y switches gestionables.
* **Sistemas y aplicaciones críticas:** sistema de nómina, intranet corporativa, plataforma de comunicación interna y gestor documental.

Limitaciones detectadas:

* Ausencia de un sistema especializado para la gestión centralizada de tickets y solicitudes.
* Inventario de activos incompleto y sin actualización en tiempo real.
* Falta de reportes estadísticos que permitan medir la eficiencia del soporte.

## 3.2 Procesos de soporte antes del proyecto

El soporte previo al proyecto presenta las siguientes características:

* **Canales dispersos:** usuarios reportan problemas por teléfono, correo o incluso en persona.
* **Sin priorización:** problemas críticos y solicitudes simples reciben el mismo nivel de atención.
* **Ausencia de métricas:** no se miden tiempos de respuesta ni se analiza el desempeño del área.

Esto genera baja trazabilidad, tiempos de resolución prolongados y repetición de fallas sin solución definitiva.

## 3.3 Detección de necesidades críticas

Las necesidades detectadas incluyen:

* Un **portal web de autoservicio** donde los usuarios puedan registrar solicitudes y consultar su estado.
* **Clasificación automática** de incidentes para priorizar los más urgentes.
* **Integración con inventario de activos**, para que cada ticket esté vinculado al equipo afectado.
* **Sistema de notificaciones** que informe al usuario sobre cambios en el estado de su solicitud.
* **Paneles de control** para la gerencia, con métricas como:
  + Tiempo promedio de resolución (MTTR).
  + Porcentaje de incidentes resueltos en primer contacto (FCR).
  + Distribución de incidentes por área.

## 3.4 Requerimientos funcionales y no funcionales del sistema

**Funcionales:**

* Interfaz intuitiva y fácil de usar.
* Base de datos robusta y escalable
* Seguridad básica: encriptación de contraseñas y control de acceso.
* Portabilidad: debe correr en cada sistema operativo.
* Documentación del código y del sistema.

**No funcionales:**

* Disponibilidad para garantizar servicio continuo.
* Cifrado de datos en tránsito y en reposo.
* Acceso seguro mediante autenticación y permisos por rol.
* Compatibilidad multiplataforma (web, móvil).
* Capacidad para integrarse con herramientas de monitoreo y gestión de red.

# Capítulo IV. Diseño de la Mesa de Ayuda

## 4.1 Arquitectura general del sistema

La solución diseñada para la Mesa de Ayuda de la Dirección General de Personal, coordinada a través de la Subdirección de Desarrollo Tecnológico, se fundamenta en las arquitecturas MVC y SRUM escalables que permiten ofrecer una plataforma eficiente, segura y fácil de mantener, además son capaces de adaptarse a cambios. El sistema está compuesto por tres capas principales, presentación (frontend), lógica de negocio (backend) y persistencia de datos (base de datos).

### 4.1.1 Tecnología base

La arquitectura propuesta se compone de los siguientes elementos tecnológicos:

|  |  |  |
| --- | --- | --- |
| Capa | Tecnología | Descripción |
| Frontend | **IntelliJ IDEA** | Interfaz web moderna, rápida y responsiva con renderizado híbrido (SSR/CSR). |
| Backend | **Spring Boot (Java)** | API REST robusta que expone los servicios del sistema y contiene la lógica de negocio. |
| Base de datos  Control de versiones | **Oracle o PostgreSQL**  **Git + GitHub** | Sistema de gestión de base de datos relacional de código abierto, confiable y seguro.  Git es la herramienta para realizar el seguimiento de los cambios, mientras que GitHub es el lugar para almacenar los proyectos y trabajar en equipo |

### 4.1.2 Estructura de la arquitectura

La arquitectura general del sistema se puede representar mediante las siguientes capas:

**a) Capa de Presentación (Next.js)**

* Se encarga de la interacción directa con los usuarios finales y técnicos.
* Desarrollada con Next.js, permite:
  + Renderizado del lado del servidor (SSR) para mayor velocidad y SEO.
  + Uso de componentes React reutilizables.
  + Gestión de sesiones y roles mediante JWT o tokens de autenticación.
* Módulos disponibles en esta capa:
  + Registro de tickets
  + Seguimiento de solicitudes
  + Dashboard de indicadores
  + Panel de técnico y administrador

**b) Capa de Negocio (Spring Boot)**

* Implementa todos los procesos lógicos del sistema.
* Expone servicios web mediante API RESTful.
* Gestiona:
  + Procesamiento de solicitudes, incidentes, problemas, cambios y activos.
  + Validaciones de reglas de negocio.
  + Seguridad (roles, permisos, autenticación).
  + Integración con servicios externos si se requiere (correo electrónico, Active Directory, etc.).
* Incluye controladores (Controllers), servicios (Services) y repositorios (Repositories) como parte de la arquitectura en capas.

**c) Capa de Persistencia (PostgreSQL)**

* Define y gestiona el almacenamiento de todos los datos del sistema.
* Principales entidades del modelo:
  + Usuarios
  + Tickets
  + Equipos y resguardos
  + Licencias (Office 365 y Windows)
  + Correos institucionales
  + Catálogos y configuración
  + Bitácoras y logs de auditoría
* PostgreSQL fue seleccionado por su confiabilidad, soporte de tipos complejos, extensibilidad, y capacidad para integrarse con herramientas de monitoreo y backup.

### 4.1.3 Comunicación entre capas

* La comunicación entre el **frontend (Next.js)** y el **backend (Spring Boot)** se realiza mediante llamadas **HTTP (API REST)** utilizando **JSON** como formato de intercambio de datos.
* La seguridad de las transacciones está asegurada mediante:
  + CORS configurado en Spring Boot.
  + Autenticación basada en JWT y políticas de autorización por rol.
  + Conexión a la base de datos mediante JDBC y control de acceso por usuario y rol.

### 4.1.4 Seguridad de la arquitectura

* Autenticación y autorización: gestionadas con Spring Security y tokens JWT.
* Protección contra CSRF y ataques de inyección SQL.
* Validaciones tanto en frontend como en backend.
* Registro de actividades y cambios críticos en una bitácora.
* Configuración de roles: usuario final, técnico, administrador.

## 4.2 Diagrama de flujo funcional de la Mesa de Ayuda

* Presentación gráfica del flujo de atención (basado en el diagrama proporcionado).
* Flujo desde el usuario final hasta la resolución: entrada, asignación, seguimiento, cierre.
* Roles involucrados: usuario, técnico, supervisor, administrador.

## 4.3 Módulos críticos del sistema

### 4.3.1 Gestión de Solicitudes de Servicio

* Registro, seguimiento y cierre de solicitudes estándar (no incidentes).
* Ejemplos: instalación de software, asignación de recursos, creación de cuentas.

### 4.3.2 Gestión de Incidentes

* Registro de fallas técnicas, errores de software, bloqueos de usuario, etc.
* Priorización y asignación automática.
* Enlace con la base de conocimientos.

### 4.3.3 Gestión de Problemas

* Identificación de causas raíz a partir de incidentes recurrentes.
* Documentación y propuesta de soluciones permanentes.

### 4.3.4 Gestión de Cambios

* Flujo de aprobación: solicitud, análisis, comité de cambios, ejecución.
* Gestión de riesgos y respaldo antes de aplicar cambios críticos.

### 4.3.5 Gestión de Activos

* Inventario automatizado de hardware y software.
* Relación con usuarios, resguardos y licenciamiento.
* Control de alta, baja y modificación de activos.

### 4.3.6 Módulo de Informes y Dashboards

* Indicadores clave: tickets abiertos/cerrados, tiempos de atención, SLA.
* Reportes personalizados para áreas directivas y operativas.
* Integración de dashboards interactivos y widgets.

## 4.4 Módulos complementarios estratégicos

### 4.4.1 Módulo de Resguardos y Cambios de Equipo

* Asignación de equipos a usuarios mediante formatos digitales.
* Control de cambios de equipo, préstamo temporal y devoluciones.
* Relación con inventario y sistema de activos.

### 4.4.2 Módulo de Licenciamiento Office 365 y Windows

* Gestión del licenciamiento institucional (por usuario, dispositivo o volumen).
* Seguimiento de fechas de activación, renovación y disponibilidad.
* Relación con activos registrados.

### 4.4.3 Gestión de Correos Electrónicos Institucionales

* Alta, baja y modificaciones de cuentas institucionales.
* Reasignación por cambios de unidad o titular.
* Plantillas de solicitud y flujos de autorización.

## 4.5 Repositorio de configuración y conocimiento

### 4.5.1 Base de Datos de Gestión de Configuración (CMDB)

* Registro de componentes de TI y sus relaciones.
* Vinculación con incidencias, cambios y activos.

### 4.5.2 Base de Conocimientos

* Soluciones a problemas frecuentes.
* Manuales técnicos y administrativos.
* Disponible para técnicos y usuarios finales (autoservicio).

## 4.6 Seguridad y control de acceso

* Definición de roles y permisos: técnicos, administradores, supervisores, usuarios.
* Registro de auditoría y bitácora de cambios.
* Control de sesiones y autenticación (LDAP, Active Directory o institucional).

## 4.7 Mecanismos de autoservicio

* Portal del usuario final:
  + Registro de tickets.
  + Consulta de estado.
  + Acceso a la base de conocimientos.
* Interfaz amigable y accesible.

## 4.8 Plan de escalabilidad y soporte

* Posibilidades de crecimiento: integración con nuevas áreas, servicios o sedes.
* Mantenimiento preventivo y correctivo del sistema.
* Proyección para automatización de respuestas mediante inteligencia artificial o chatbots.

# Capítulo V. Desarrollo e implementación

## 5.1 Metodología de desarrollo utilizada (Scrum, Cascada, etc.)

## 5.2 Cronograma de actividades

## 5.3 Configuración del entorno de desarrollo y pruebas

## 5.4 Implementación de módulos (con evidencias: pantallas, diagramas, código, pruebas unitarias, etc.)

## 5.5 Pruebas funcionales y de usuario

## 5.6 Integración con infraestructura existente

# Capítulo VI. Resultados y beneficios

## 6.1 Comparativo antes/después

## 6.2 Tiempo de respuesta y eficiencia del área

## 6.3 Control de activos y licenciamiento

## 6.4 Mejora en la gestión de cambios y soporte técnico

## 6.5 Reducción de errores y documentación de incidencias

# Capítulo VII. Conclusiones y recomendaciones

## 7.1 Conclusiones generales del proyecto

## 7.2 Recomendaciones para mejoras futuras

## 7.3 Sugerencias de escalabilidad

# Capítulo VII. Glosario de términos técnicos

# Capítulo IX. Bibliografías

# Capítulo X. Anexos

* Diagrama de flujo detallado
* Evidencias fotográficas
* Manual de usuario
* Manual técnico
* Bitácora de actividades
* Cronograma de avance
* Formatos de tickets, licencias y resguardos