

**ItplusNET-crm**

**Sistema de Gestión de Soporte**



Bastian Gutierrez henriquez

Alister Gonzalez

Lhian espinoza  
Duoc uc plaza oeste  
capstone  
[Nombre del Profesor]  
[Fecha de Entrega]

INDICE

[1. Introducción 2](#_Toc181060032)

[2. Objetivos del Proyecto 2](#_Toc181060033)

[2.1Objetivo General 2](#_Toc181060034)

[2.2 Objetivos Específicos 2](#_Toc181060035)

[3. Justificación del Proyecto 2](#_Toc181060036)

[4. Alcance del Proyecto 2](#_Toc181060037)

[5. Arquitectura del Sistema 2](#_Toc181060038)

[5.1 Modelo Arquitectónico 2](#_Toc181060039)

[5.2 Vistas del Modelo 4+1 2](#_Toc181060040)

[6. Tecnologías Utilizadas 2](#_Toc181060041)

[7. Módulos y Funcionalidades 2](#_Toc181060042)

[7.1. Gestión de Tickets 2](#_Toc181060043)

[7.2. Sistema de Roles y Permisos 2](#_Toc181060044)

[7.3. Notificaciones Automáticas 2](#_Toc181060045)

[7.4. Panel de Control (Dashboard) 2](#_Toc181060046)

[7.5. Historial de Interacciones 2](#_Toc181060047)

[7.6. Base de Conocimiento 2](#_Toc181060048)

[7.7. Automatización de Tareas 2](#_Toc181060049)

[7.8. Autenticación y Seguridad 2](#_Toc181060050)

[7.9. Sistema de Asistencia Inteligente 2](#_Toc181060051)

[7.10. Sistema de Reportes 2](#_Toc181060052)

[7.11. Integración de APIs 2](#_Toc181060053)

[8. Integración de Inteligencia Artificial 2](#_Toc181060054)

[8.1 Modelo de Machine Learning para Clasificación de Tickets 2](#_Toc181060055)

[8.2 Asistente Inteligente Basado en ChatGPT 2](#_Toc181060056)

[9. Diseño de la Base de Datos 2](#_Toc181060057)

[10. Análisis de Riesgos y Mitigación 2](#_Toc181060058)

[11. Cronograma del Proyecto 2](#_Toc181060059)

[12. Recursos y Presupuesto 3](#_Toc181060060)

[13. Beneficios y Valor Agregado 3](#_Toc181060061)

[14. Conclusión 3](#_Toc181060062)

# 

# **1. Introducción**

En la era digital actual, las empresas enfrentan desafíos cada vez más complejos en la gestión eficiente de sus operaciones y en la satisfacción de las demandas crecientes de sus clientes. La tecnología avanza rápidamente, y con ella, las expectativas de calidad, rapidez y personalización en los servicios. MENTOR ITPLUS LTDA., una consultora informática ubicada en Santiago, Chile, se dedica desde 2014 a proporcionar soluciones avanzadas en informática y tecnología. Su misión es impulsar la transformación digital de sus clientes mediante servicios de desarrollo de software a medida, consultoría tecnológica y soluciones de infraestructura robustas. Con un equipo de expertos apasionados, combina creatividad y experiencia técnica para ofrecer resultados excepcionales y satisfacer las necesidades más exigentes del mercado.

La falta de un sistema centralizado y eficiente para manejar las solicitudes de soporte ha provocado demoras, pérdida de información y, en última instancia, insatisfacción de los clientes. En un entorno donde la competencia es feroz y la lealtad del cliente es fundamental, es imprescindible que MENTOR ITPLUS LTDA. adopte soluciones tecnológicas que optimicen sus procesos internos y eleven la calidad del servicio ofrecido.

Este proyecto propone el desarrollo de un sistema CRM (Customer Relationship Management) integral y escalable, denominado itplusnet-CRM, que aborda las deficiencias actuales mediante la incorporación de tecnologías modernas como un sistema de asistencia inteligente basado en ChatGPT. Este asistente facilitará la recopilación de información precisa sobre los problemas reportados, mejorando la interacción inicial con los clientes y optimizando el flujo de trabajo del equipo de soporte.

La implementación de itplusnet-CRM representa más que una actualización tecnológica; es una estrategia clave para impulsar el crecimiento sostenible de la empresa, fortalecer las relaciones con los clientes y posicionarse como un referente en soluciones tecnológicas innovadoras. Este documento detalla el desarrollo del proyecto, sus objetivos, alcance y beneficios esperados, proporcionando una guía exhaustiva para su ejecución exitosa.

# **2. Objetivos del Proyecto**

## **2.1Objetivo General**

Desarrollar e implementar un sistema **CRM integral y escalable**, denominado **itplusnet-CRM**, que optimice la gestión del soporte técnico en **itplusnet** mediante la integración de inteligencia artificial y tecnologías avanzadas. Este sistema estará orientado a incrementar en un 30% la eficiencia operativa, fortaleciendo la capacidad de la empresa para fortalecer un soporte mas ágil, preciso y orientado a las necesidades del cliente.

## **2.2 Objetivos Específicos**

1. **Diseñar e implementar un sistema de autenticación** que garantice la protección de la información sensible y un control de acceso efectivo, asegurando que cada usuario tenga asignados roles específicos con permisos correspondientes según sus necesidades.
2. **Desarrollar un dashboard interactivo** que proporcione a gerentes y administradores una visualización en tiempo real de métricas clave, como el número de tickets abiertos, el nivel de prioridad de los tickets y la tasa de resolución de incidencias, facilitando la toma de decisiones informadas basadas en datos de soporte.
3. **Implementar un sistema de asistencia inteligente al cliente**, utilizando inteligencia artificial (ChatGPT) para guiar al usuario en la descripción y el detalle de su problema, asegurando que la información proporcionada sea completa y precisa, con el fin de optimizar el trabajo del técnico y evitar la necesidad de llamadas adicionales.
4. **Integrar y consumir APIs proporcionadas por itplusnet,** permitiendo la sincronización en tiempo real de las operaciones logísticas de la empresa (gestión de sucursales, cajas, usuarios y bodegas) con el CRM, asegurando una frecuencia de actualización constante y una tasa de sincronización del 100% en los datos críticos.
5. **Optimizar la interfaz de usuario**, asegurando que sea responsive y amigable, mejorando la usabilidad y accesibilidad para todos los usuarios del sistema.

# **3. Justificación del Proyecto**

MENTOR ITPLUS LTDA. es una empresa de innovación tecnológica fundada en 2014, dedicada a proporcionar soluciones avanzadas en informática y tecnología. Con una misión centrada en impulsar la transformación digital de sus clientes, la empresa ofrece servicios de desarrollo de software a medida, consultoría tecnológica y soluciones de infraestructura robustas. Su equipo de expertos combina creatividad y experiencia técnica para satisfacer las necesidades más exigentes del mercado, destacándose como un referente en el ámbito tecnológico.

Actualmente, MENTOR ITPLUS LTDA. enfrenta desafíos significativos en la gestión de su soporte técnico, un área clave para garantizar la satisfacción de sus clientes. La falta de un sistema centralizado y eficiente ha generado demoras en la atención, dificultades en el seguimiento de tickets y una asignación ineficiente de recursos. Además, la creciente complejidad de las solicitudes y la necesidad de garantizar la seguridad de la información hacen imprescindible la implementación de una solución avanzada.

**Problemas actuales:**

* **Falta de seguimiento eficiente:** Sin un sistema centralizado, es difícil rastrear el progreso de las solicitudes de soporte, lo que genera demoras y pérdida de información.
* **Ausencia de priorización de tickets:** Todos los tickets son tratados de manera uniforme, sin considerar la urgencia o importancia, lo que resulta en una asignación ineficiente de recursos.
* **Limitaciones en la comunicación con el cliente:** La falta de notificaciones automáticas aumenta la frustración de los clientes y carga de trabajo del equipo.
* **Riesgos de seguridad:** La ausencia de mecanismos robustos de autenticación deja expuesta información sensible a accesos no autorizados.

**Beneficios esperados y métricas asociadas:**

1. **Mejora en la satisfacción del cliente:** A través de encuestas periódicas, se espera incrementar la satisfacción del cliente en un 20% al proporcionar un seguimiento más transparente y eficiente.
2. **Optimización de recursos internos:** Se proyecta reducir los tiempos de atención en un 30% gracias a la priorización y automatización de procesos.
3. **Incremento en la seguridad del sistema:** El número de intentos no autorizados bloqueados se espera aumentar significativamente, reforzando la protección de los datos sensibles.
4. **Facilitación de la toma de decisiones:** Con dashboards interactivos y reportes analíticos, la gerencia podrá identificar tendencias clave, con una mejora estimada del 25% en la precisión de sus estrategias.
5. **Posicionamiento competitivo:** El uso de tecnologías avanzadas, como la inteligencia artificial, permitirá a MENTOR ITPLUS LTDA. diferenciarse y captar nuevos clientes, contribuyendo a un crecimiento proyectado del 15% en su cartera de clientes en el próximo año.

**Preparación para el futuro:**  
La implementación de itplusnet-CRM no solo resuelve problemas actuales, sino que también sienta las bases para futuras expansiones y adaptaciones. Al integrar un sistema escalable y preparado para incorporar nuevas tecnologías, MENTOR ITPLUS LTDA. asegura su capacidad para enfrentar desafíos tecnológicos emergentes y mantenerse como líder en el sector.

# **4. Alcance del Proyecto**

El sistema itplusnet-CRM se centrará en el desarrollo e implementación de funcionalidades clave diseñadas para abordar las necesidades específicas de la empresa y mejorar su gestión de soporte técnico. Este alcance abarca tanto la creación de herramientas avanzadas como la integración de tecnologías de última generación, asegurando una experiencia eficiente y segura para los usuarios. A continuación, se presentan las características principales organizadas en módulos funcionales:

|  |  |
| --- | --- |
| Modulo | Funcionalidad |
| |  | | --- | | Gestión de Tickets |  |  | | --- | |  | | - Creación, seguimiento y cierre de tickets. |
|  | - Registro de una línea de tiempo detallada para cada ticket. |
|  | - Notificaciones automáticas por correo electrónico en eventos clave. |
| Sistema de seguridad | - Autenticación basada en roles (Administrador, Gerente, Cliente, Desarrollador). |
|  | - Implementación de un captcha para prevenir accesos no autorizados. |
|  | - Recuperación de contraseñas y gestión de sesiones seguras. |
| Dashboard y Reportes | - Visualización en tiempo real de métricas clave. |
|  | - Generación de reportes personalizados en formatos como PDF y Excel. |
|  | - Gráficos estadísticos sobre rendimiento y análisis histórico de soporte. |
| Integración con APIs | - Consumo de APIs externas para sincronización de operaciones logísticas (sucursales, cajas, bodegas). |
|  | - Desarrollo de una API propia para facilitar la comunicación entre módulos. |
| Interfaz de Usuario | - Diseño responsive y adaptativo para diferentes dispositivos. |
|  | - Optimización del flujo de navegación y usabilidad. |

El alcance descrito asegura una solución integral que abordará las ineficiencias actuales y establecerá una base tecnológica robusta para futuras expansiones. Cada módulo y funcionalidad ha sido diseñado para garantizar la eficiencia operativa, la seguridad de los datos y la satisfacción del usuario final, con una arquitectura flexible que se adapta a las necesidades cambiantes de la empresa.

# **5. Arquitectura del Sistema**

La arquitectura del sistema describe cómo se organiza y estructura el software para cumplir con los objetivos del proyecto. Se ha diseñado utilizando un modelo que facilita la claridad, el mantenimiento y la escalabilidad del sistema. Esto asegura que cada parte del proyecto funcione de manera eficiente y se adapte a necesidades futuras.

## **5.1 Modelo Arquitectónico**

El sistema itplusnet-CRM empleará el Modelo Vista-Controlador (MVC), que divide el desarrollo en tres componentes principales:

* **Modelo:** Gestiona los datos, asegurándose de que se almacenen correctamente en la base de datos y se procesen de acuerdo con las reglas del negocio.
* **Vista:** Es la interfaz visible para el usuario, diseñada para ser amigable y fácil de usar, adaptándose a diferentes dispositivos como computadores y teléfonos móviles.
* **Controlador:** Actúa como intermediario entre el Modelo y la Vista, procesando las solicitudes de los usuarios y devolviendo respuestas basadas en los datos.

Esta estructura simplifica el mantenimiento del sistema y asegura que cada módulo cumpla con una función específica.

## **5.2 Vistas del Modelo 4+1**

**1. Vista Lógica**

Para entender mejor cómo funciona el sistema, se utiliza el Modelo 4+1, que organiza la arquitectura en cinco perspectivas:

|  |  |
| --- | --- |
| Descripción | Componentes Clave |
| |  | | --- | |  |  |  | | --- | |  | | - Módulo de Usuarios y Roles |
|  | -Módulo de Autenticación y Seguridad |
|  | -Modulo de Soporte y Gestión de Tickets |
| Estructura modular del sistema  enfocada en la separación de  Responsabilidades. | -Modulo de Inteligencia Artificial |
|  | -Modulo de Reportes y Dashboard |
|  | -Módulo de Integración de APIs |

La Vista Lógica define cómo se estructuran las funciones principales del sistema, asegurando una separación clara entre módulos como la autenticación, la gestión de tickets y la visualización de métricas. Esta separación facilita el mantenimiento y la posibilidad de ampliar el sistema en el futuro sin afectar otras funcionalidades. Además, permite a los desarrolladores identificar rápidamente las áreas que necesitan ajustes o mejoras. El modularidad también garantiza que cada componente pueda operar de manera independiente pero conectado al sistema principal, asegurando una experiencia eficiente para los usuarios.

**2. Vista de Procesos**

|  |  |
| --- | --- |
| Descripción | Aspectos Claves |
| |  | | --- | |  |  |  | | --- | |  | | -Flujo de creación, asignación y resolución de tickets. |
|  | -Comunicación entre el CRM y el sistema de asistencia inteligente. |
| Interacción en tiempo real entre clientes, desarrolladores, gerentes y el sistema de inteligencia. | -Notificación automáticas y actualización en tiempo real de estados. |
|  | Automatización de tareas repetitivas y escalación de tickets. |

La Vista de Procesos explica la interacción dinámica entre los diferentes módulos y usuarios del sistema. Por ejemplo, cuando un cliente genera un ticket, este se procesa automáticamente por el sistema, que lo asigna a un técnico basado en prioridades predefinidas. Esto elimina pasos manuales y minimiza errores humanos. Los procesos automatizados también aseguran que las tareas repetitivas, como el envío de notificaciones o actualizaciones de estado, se realicen en tiempo real. Este enfoque mejora la eficiencia operativa y garantiza que el sistema funcione de manera óptima, incluso bajo alta carga.

**3. Vista de Desarrollo**

|  |  |  |
| --- | --- | --- |
| Tecnologías | Herramientas | Buenas Practicas |
| |  | | --- | |  |  |  | | --- | |  |   Backend: PHP con Laravel | Control de Versiones con Git | Uso de estándares decodificación. |
| Frontend: HTML, CSS, JS, TAILWIND, BOOSTRAP | Entorno de desarrollo con Composer | Pruebas de unidad e integración. |
| MySQL para la base de datos | NPM para la gestión de paquetes | Documentación técnica actualizada |
|  | Automatización de tareas repetitivas y escalación de tickets. |  |

La Vista de Desarrollo se centra en cómo se organiza el código y las herramientas utilizadas en el proyecto. Utilizando Laravel como framework backend, el equipo de desarrollo puede trabajar en un entorno estructurado que promueve la eficiencia y la consistencia en el código. Herramientas como Git aseguran que los cambios se realicen de manera controlada y se puedan revertir si es necesario. Además, la separación entre frontend y backend permite que los desarrolladores trabajen en paralelo, acelerando el tiempo de entrega del sistema. La elección de tecnologías estándar garantiza compatibilidad y soporte a largo plazo.

**4. Vista Física**

|  |  |
| --- | --- |
| Descripcion | Infraestructura Utilizada |
| |  | | --- | |  |  |  | | --- | |  |   Servidor web | Administrado mediante WHM, que facilita la gestión de múltiples cuentas y configuraciones |
| Gestion de aplicaciones y archivos | cPanel se utiliza para manejar archivos, bases de datos y aplicaciones del sistema |
| Base de Datos | MySQL configurado y administrado desde cPanel |
| Seguridad | Integración de certificados SSL para encriptar las comunicaciones y protección mediante firewalls configurados desde WHM. |
| Backup automáticos | Implementación de copias de seguridad periódicas mediante las funciones nativas de WHM y cPanel |

La Vista Física detalla cómo se implementará el sistema itplusnet-CRM en la infraestructura de la empresa. Utilizando WHM y cPanel, la gestión de servidores será centralizada, lo que facilita la configuración y administración de los servicios necesarios. WHM permite gestionar múltiples cuentas y configuraciones del servidor, mientras que cPanel ofrece una interfaz amigable para administrar aplicaciones, bases de datos y archivos. Además, la seguridad se refuerza mediante el uso de certificados SSL, asegurando la encriptación de datos durante la comunicación entre usuarios y el sistema. Las copias de seguridad automáticas y el monitoreo constante reducen riesgos operativos y garantizan la continuidad del servicio, incluso ante posibles fallos técnicos. Esta infraestructura es ideal para manejar un alto volumen de usuarios y asegurar una experiencia confiable.

5.3 Historias de Usuarios

5.3.1 Gestión de Ticket

|  |  |  |  |
| --- | --- | --- | --- |
| Usuario | Historia de Usuario | Flujo de trabajo Detallado | Criterios de Aceptación |
| |  | | --- | |  |  |  | | --- | |  |   Cliente | Como cliente, quiero acceder al sistema y crear un nuevo ticket a través de un formulario donde pueda seleccionar el tipo de problema, la urgencia, y añadir una descripción detallada.  Una vez que el ticket se genera al completar el formulario, quiero interactuar con una inteligencia artificial basada en ChatGPT, que me ayude a estructurar y detallar mejor el problema para que el desarrollador reciba información clara y precisa. | - El cliente inicia sesión en el sistema.  -crea un nuevo ticket a través de un formulario. - Interactúa con la IA basada en ChatGPT. - La IA guía al cliente para detallar el problema (tipo, urgencia, descripción). - El sistema genera un ticket con los detalles recolectados. - El ticket se asigna automáticamente a un desarrollador o manualmente por el gerente. | - La IA guía al cliente de forma clara y registra toda la información proporcionada. - El ticket incluye los datos capturados por la IA y queda asignado correctamente. |
| Desarrollador | |  | | --- | | Como desarrollador, quiero recibir notificaciones sobre los tickets asignados para saber cuáles son mis tareas pendientes. | |  | |  | |  | |  |  |  | | --- | |  | | - El sistema envía una notificación al desarrollador con los detalles del ticket asignado. - Accede al sistema y revisa el ticket en su lista de asignaciones. | - El desarrollador recibe la notificación en tiempo real. - El sistema registra que el ticket ha sido visualizado por el desarrollador. |

5.3.2 Sistema de roles y Permisos

|  |  |  |  |
| --- | --- | --- | --- |
| Usuario | Historia de Usuario | Flujo de trabajo Detallado | Criterios de Aceptación |
| |  | | --- | |  |  |  | | --- | |  |   Administrador | Como administrador, quiero definir los roles de los usuarios para controlar el acceso a las funcionalidades del sistema. | - El administrador accede al módulo de configuración. - Crea o edita un rol (Cliente, Desarrollador, Gerente, Administrador). - Asigna permisos específicos según las políticas de la empresa. | - Cada usuario tiene asignado un rol con permisos definidos. - El sistema valida el acceso según el rol asignado antes de permitir cualquier acción. |
| Cliente | Como cliente, quiero tener acceso únicamente a los tickets que he creado para proteger la privacidad de mis datos. | - Accede a su cuenta. - El sistema filtra los tickets visibles según el ID del cliente. - Puede ver, comentar y cerrar sus propios tickets. | - El cliente solo puede visualizar sus tickets. - Cualquier intento de acceso a tickets de otros usuarios genera un mensaje de error. |

**5.3.3 Notificaciones Automáticas**

|  |  |  |  |
| --- | --- | --- | --- |
| Usuario | Historia de Usuario | Flujo de trabajo Detallado | Criterios de Aceptación |
| |  | | --- | |  |  |  | | --- | |  |   Cliente | Como cliente, quiero recibir un correo cuando mi ticket cambie de estado para estar al tanto del progreso. | - El cliente crea un ticket. - Cuando el ticket cambia de estado (Ej. En proceso, Resuelto), el sistema genera una notificación automática por correo. - Recibe un resumen del cambio en el ticket. | - Las notificaciones se envían en tiempo real. - Incluyen información clara sobre el estado y cambios recientes en el ticket. |
| Gerente | - Las notificaciones se envían en tiempo real. - Incluyen información clara sobre el estado y cambios recientes en el ticket. | - El sistema monitorea los tiempos de inactividad de los tickets. - Si un ticket supera el tiempo permitido sin avances, se envía una alerta al gerente o al desarrollador asignado. | - Las alertas se generan automáticamente para tickets críticos o no atendidos. - Incluyen detalles claros del ticket y el tiempo desde su creación. |

**5.4.4 Panel de Control(Dashboard)**

|  |  |  |  |
| --- | --- | --- | --- |
| Usuario | Historia de Usuario | Flujo de trabajo Detallado | Criterios de Aceptación |
| |  | | --- | |  |  |  | | --- | |  |   Gerente | Como gerente, quiero ver en tiempo real las métricas clave de los tickets para tomar decisiones informadas sobre la distribución del trabajo. | Como gerente, quiero ver en tiempo real las métricas clave de los tickets para tomar decisiones informadas sobre la distribución del trabajo. | - El panel muestra métricas en tiempo real con gráficos interactivos. - Puede filtrar la información según prioridades o asignaciones. |

**5. Casos de Uso**

* **Descripción**: Escenarios que ilustran la interacción de los usuarios con el sistema.
* **Ejemplos**:
  1. **Creación de Ticket por el Cliente**:
     + El cliente inicia sesión con su cuenta y es validado mediante captcha.
     + Accede al formulario de creación de tickets, selecciona el tipo de problema, urgencia y añade una descripción detallada.
     + El sistema asigna automáticamente el ticket a un desarrollador según prioridad y disponibilidad, o el gerente lo asigna manualmente.
     + El desarrollador recibe una notificación y comienza a trabajar en la resolución.
  2. **Clasificación Automática de Tickets**:
     + Al recibir un nuevo ticket, el modelo de machine learning analiza el contenido.
     + Asigna una prioridad al ticket (alto, medio, bajo) y lo categoriza.
     + El equipo de soporte es notificado según la prioridad asignada.
  3. **Uso del Asistente Inteligente(ChatGPT):**
     + Al crear un ticket, el cliente recibe sugerencias de soluciones basadas en ChatGPT.
     + Si la solución es útil, el cliente puede resolver su problema sin necesidades de asistencia adicional.
     + Sino, procede a enviar el ticket al equipo de soporte.
  4. **Uso del Dashboard por Gerentes**:
     + Un gerente inicia sesión y accede al dashboard interactivo.
     + Visualiza métricas clave, como número de tickets abiertos, tiempos de respuesta y rendimiento del equipo.
     + Genera un reporte mensual en PDF para presentarlo en una reunión.
  5. **Integración con Sistemas Externos**:
     + Un ticket relacionado con problemas de stock es creado.
     + El CRM consulta a través de la API el estado de inventario en tiempo real.
     + La información es proporcionada al cliente o escalada al área correspondiente.

# **6. Tecnologías Utilizadas**

El desarrollo del **itplusnet-CRM** se apoyará en tecnologías modernas y robustas que garantizan un rendimiento óptimo, seguridad y facilidad de mantenimiento.

* **Framework Backend**: **Laravel** (PHP)
  + Ofrece una estructura sólida basada en MVC.
  + Facilita tareas comunes como enrutamiento, seguridad y gestión de bases de datos.
  + Soporta integración con componentes adicionales y paquetes de terceros.
* **Base de Datos**: **MySQL**
  + Sistema de gestión de bases de datos relacional ampliamente utilizado.
  + Ofrece rendimiento, escalabilidad y seguridad.
  + Compatible con Eloquent ORM de Laravel para manipulación eficiente de datos.
* **Frontend**:
  + **HTML5 y CSS3**: Estructura y estilos básicos de las páginas web.
  + **JavaScript**: Para interacción y dinamismo en el lado del cliente.
  + **Bootstrap**: Framework CSS para diseño responsive y componentes predefinidos.
* **Bibliotecas y Herramientas Adicionales**:
  + **jQuery**: Para simplificar la manipulación del DOM y manejar eventos.
  + **Axios o Fetch API**: Para realizar solicitudes HTTP desde el frontend.
  + **Chart.js o D3.js**: Para generación de gráficos en el dashboard.
  + **ReCaptcha de Google**: Para implementación del captcha en la autenticación.
* **Inteligencia Artificial**:
  + **Python**: Lenguaje de programación para desarrollar modelos de machine learning.
  + **Frameworks de ML**: Scikit-learn para machine learning tradicional y TensorFlow o PyTorch para deep learning.
  + **APIs de OpenAI**: Para integrar ChatGPT en el asistente inteligente.
* **Herramientas de Desarrollo**:
  + **Composer**: Gestión de dependencias de PHP.
  + **NPM**: Gestión de paquetes de JavaScript.
  + **Git**: Control de versiones.
  + **IDE**: Visual Studio Code, PHPStorm u otros según preferencia.
* **Servicios y Plataformas**:
  + **Servicios de Correo Electrónico**: SMTP, SendGrid o Mailgun para notificaciones.
  + **Servicios de Hosting**: AWS, Azure, DigitalOcean u otros proveedores.

# **7. Módulos y Funcionalidades**

## **7.1. Gestión de Tickets**

* **Función:** Permite a los usuarios crear tickets para reportar problemas o solicitar soporte técnico.
* **Subfunciones:**
  + Creación de tickets a través de un formulario intuitivo.
  + Asignación de tickets a desarrolladores de forma automática o manual.
  + Seguimiento del estado del ticket (abierto, en proceso, resuelto, cerrado).
  + Edición de tickets para actualizar información o estado.
  + Cierre de tickets y envío de notificaciones al cliente con la resolución.
* **Flujo de Trabajo:**
  + El cliente crea un nuevo ticket, seleccionando tipo de problema y urgencia.
  + El sistema asigna el ticket a un desarrollador o el gerente lo asigna manualmente.
  + El desarrollador recibe una notificación y trabaja en la resolución, actualizando el estado del ticket.
  + Al resolver el problema, el ticket se cierra y el cliente es notificado.

## **7.2. Sistema de Roles y Permisos**

* **Función:** Define el nivel de acceso y permisos que tiene cada tipo de usuario en el sistema.
* **Roles:**
  + **Clientes:** Pueden crear, ver y comentar sus tickets.
  + **Desarrolladores:** Pueden ver, gestionar y cerrar los tickets asignados.
  + **Gerente:** Puede ver todos los tickets, asignarlos y generar informes.
  + **Administrador:** Acceso completo para gestionar el sistema, incluyendo configuración y creación de usuarios.
* **Flujo de Trabajo:**
  + Un administrador asigna roles y permisos.
  + Al acceder, el sistema verifica el rol del usuario y permite acciones según sus permisos.
  + Cada acción (crear, editar, cerrar tickets) está regulada por los permisos del rol.

## **7.3. Notificaciones Automáticas**

* **Función:** Enviar notificaciones al cliente y al desarrollador cuando el estado del ticket cambia.
* **Subfunciones:**
  + Notificaciones por correo electrónico y dentro del sistema.
  + Personalización de mensajes de notificación.
* **Flujo de Trabajo:**
  + Al crear un ticket, el cliente recibe una confirmación.
  + El desarrollador asignado es notificado de un nuevo ticket.
  + Cambios en el estado del ticket generan notificaciones a las partes involucradas.
  + Si hay inactividad o necesidad de escalación, se envían alertas adicionales.

## **7.4. Panel de Control (Dashboard)**

* **Función:** Visualiza las métricas clave para la gestión del soporte.
* **Subfunciones:**
  + Métricas de tickets abiertos, cerrados y en progreso.
  + Tiempo promedio de respuesta y resolución.
  + Número de tickets por desarrollador y su rendimiento.
  + Comparativa de resolución de tickets urgentes vs. no urgentes.
* **Flujo de Trabajo:**
  + El gerente accede al dashboard y visualiza métricas en tiempo real.
  + Puede profundizar en datos específicos y tomar decisiones informadas.

## **7.5. Historial de Interacciones**

* **Función:** Almacena y muestra todo el historial de interacciones de un cliente con el equipo de desarrollo.
* **Subfunciones:**
  + Visualización de tickets previos.
  + Notas y comentarios agregados por desarrolladores.
  + Adjuntos relevantes para el seguimiento.
* **Flujo de Trabajo:**
  + Clientes y desarrolladores pueden acceder al historial para comprender problemas recurrentes.
  + Facilita la identificación de patrones y soluciones efectivas.

## **7.6. Base de Conocimiento**

* **Función:** Almacena artículos, tutoriales y soluciones comunes que los clientes pueden consultar antes de abrir un ticket.
* **Subfunciones:**
  + Artículos categorizados por tipo de problema.
  + Motor de búsqueda para encontrar soluciones.
  + Sistema de votos para evaluar la utilidad de los artículos.
* **Flujo de Trabajo:**
  + El cliente consulta la base de conocimiento y puede resolver problemas sin asistencia directa.
  + Los desarrolladores actualizan la base con nuevas soluciones.

## **7.7. Automatización de Tareas**

* **Función:** Automatizar tareas repetitivas dentro del sistema.
* **Subfunciones:**
  + Asignación automática de tickets basada en criterios predefinidos.
  + Respuestas automáticas para tickets de baja prioridad.
  + Escalación automática de tickets si no han sido atendidos en un tiempo determinado.
* **Flujo de Trabajo:**
  + El sistema automatiza procesos, mejorando la eficiencia y reduciendo tiempos de respuesta.

## **7.8. Autenticación y Seguridad**

* **Registro y Autenticación de Usuarios:**
  + Proceso de inicio de sesión con validación de credenciales y captcha.
  + Recuperación y restablecimiento de contraseñas.
* **Control de Acceso Basado en Roles (RBAC):**
  + Roles y permisos definidos para cada usuario.
  + Acceso restringido a funcionalidades según el rol.
* **Medidas de Seguridad Adicionales:**
  + Encriptación de contraseñas y datos sensibles.
  + Protección contra inyección SQL y validación de entradas.

## **7.9. Sistema de Asistencia Inteligente**

* **Asistente Inteligente al Cliente:**
  + Al crear un ticket, el cliente recibe sugerencias de soluciones basadas en *ChatGPT*.
  + Permite resolver problemas sin intervención humana.
* **Integración con *ChatGPT*:**
  + Uso de modelos avanzados de lenguaje natural.
  + Personalización para el contexto de **Itplusnet**.
* **Retroalimentación del Cliente:**
  + Opción para evaluar la utilidad de las soluciones proporcionadas.
  + Datos recolectados para mejorar el asistente.

## **7.10. Sistema de Reportes**

* **Dashboard Interactivo:**
  + Visualización de métricas clave y gráficos.
* **Generación de Reportes Personalizados:**
  + Filtrado y exportación de datos.
  + Programación de reportes automáticos.
* **Análisis y Tendencias:**
  + Identificación de patrones y planificación de recursos.

## **7.11. Integración de APIs**

* **Consumo de APIs de Itplusnet:**
  + Integración con sistemas de gestión de sucursales, cajas y bodegas.
* **Desarrollo de API Propia:**
  + Comunicación entre el CRM y el asistente inteligente.
* **Seguridad en la Integración:**
  + Autenticación y encriptación en comunicaciones API.

# **8. Integración de Inteligencia Artificial**

## **8.1 Modelo de Machine Learning para Clasificación de Tickets**

* **Objetivo**:
  + Automatizar la asignación de prioridades a los tickets de soporte.
  + Mejorar los tiempos de respuesta al enfocar recursos en tickets de alta prioridad.
* **Desarrollo del Modelo**:
  + **Recolección de Datos**: Historial de tickets anteriores, incluyendo descripciones y categorías.
  + **Preprocesamiento**: Limpieza de datos, tokenización, eliminación de ruido y palabras vacías.
  + **Entrenamiento**: Uso de algoritmos como Naive Bayes, SVM o Random Forest para clasificación de texto.
  + **Validación y Evaluación**: Medición de precisión, recall y F1-score para asegurar un desempeño adecuado.
  + **Actualización Continua**: Retrain del modelo con nuevos datos para mejorar su precisión con el tiempo.
* **Integración con el CRM**:
  + Al recibir un nuevo ticket, el sistema envía la descripción al modelo de ML.
  + El modelo devuelve la prioridad asignada, que se registra en el ticket.
  + El equipo de soporte es notificado según la prioridad.

## **8.2 Asistente Inteligente Basado en ChatGPT**

* **Objetivo**:
  + Proporcionar soluciones inmediatas y relevantes a los clientes al momento de crear un ticket.
  + Reducir la carga del equipo de soporte al resolver consultas comunes automáticamente.
* **Implementación**:
  + **Integración con API de OpenAI**: Configuración de las credenciales y parámetros necesarios.
  + **Contextualización**: Proporcionar al modelo información específica de **itplusnet**, como productos, servicios y terminología.
  + **Gestión de Conversaciones**: Manejo de tokens y seguimiento de contexto para interacciones coherentes.
  + **Filtros y Moderación**: Implementación de mecanismos para evitar respuestas inapropiadas o incorrectas.
* **Experiencia del Usuario**:
  + Al ingresar la descripción del problema, el asistente inteligente analiza y proporciona posibles soluciones.
  + Interfaz amigable que permite al cliente aceptar la solución o solicitar asistencia adicional.
  + Opción para calificar la utilidad de la respuesta y proporcionar retroalimentación.
* **Mejora Continua**:
  + Recolección de datos sobre interacciones para ajustar y mejorar el desempeño del asistente.
  + Actualizaciones periódicas del contexto y base de conocimientos según cambios en los servicios o productos de **itplusnet**.

# **9. Diseño de la Base de Datos**

El diseño de la base de datos es fundamental para garantizar la eficiencia, integridad y escalabilidad del sistema. Se utilizará un esquema relacional en **MySQL**, aprovechando las capacidades de Eloquent ORM de Laravel.

**Tablas Principales**

* **Usuarios**:
  + Campos: id, nombre, apellido, email, contraseña (encriptada), rol\_id, estado, fecha\_creación, fecha\_actualización.
  + Relaciones: Uno a Muchos con **Roles**, Uno a Muchos con **Tickets**.
* **Roles**:
  + Campos: id, nombre (Administrador, Gerente, Cliente, Desarrollador), descripción.
  + Relaciones: Uno a Muchos con **Usuarios**.
* **Tickets**:
  + Campos: id, usuario\_id, título, descripción, estado\_id, prioridad\_id, dificultad\_id, fecha\_creación, fecha\_actualización.
  + Relaciones: Muchos a Uno con **Usuarios**, Muchos a Uno con **Estados**, Muchos a Uno con **Prioridades**, Muchos a Uno con **Dificultades**, Uno a Muchos con **Comentarios**, Uno a Muchos con **Historiales**.
* **Estados**:
  + Campos: id, nombre (Abierto, Pendiente, En Desarrollo, Cerrado).
  + Relaciones: Uno a Muchos con **Tickets**.
* **Prioridades**:
  + Campos: id, nivel (Alto, Medio, Bajo), descripción.
  + Relaciones: Uno a Muchos con **Tickets**.
* **Dificultades**:
  + Campos: id, nivel (Alta, Media, Baja), descripción.
  + Relaciones: Uno a Muchos con **Tickets**.
* **Comentarios**:
  + Campos: id, ticket\_id, usuario\_id, comentario, fecha\_creación.
  + Relaciones: Muchos a Uno con **Tickets**, Muchos a Uno con **Usuarios**.
* **Historiales**:
  + Campos: id, ticket\_id, acción, usuario\_id, fecha, detalles.
  + Relaciones: Muchos a Uno con **Tickets**, Muchos a Uno con **Usuarios**.
* **Empresas**:
  + Campos: id, nombre, rut, razón\_social, dirección, fecha\_creación.
  + Relaciones: Uno a Muchos con **Usuarios** (clientes asociados a empresas).
* **Sucursales**, **Cajas**, **Bodegas**:
  + Campos específicos según necesidades.
  + Relaciones definidas para reflejar la estructura logística de **itplusnet**.

**Consideraciones de Diseño**

* **Integridad Referencial**:
  + Uso de claves foráneas para mantener relaciones consistentes entre tablas.
  + Definición de restricciones y reglas de cascada para actualizaciones y eliminaciones.
* **Índices y Rendimiento**:
  + Creación de índices en campos frecuentemente consultados, como email, estado, prioridad.
  + Optimización de consultas mediante análisis de planes de ejecución.
* **Seguridad de Datos**:
  + Encriptación de contraseñas y datos sensibles.
  + Control de acceso a nivel de base de datos si es necesario.

# **10. Análisis de Riesgos y Mitigación**

Identificar posibles riesgos es esencial para garantizar el éxito del proyecto. A continuación, se presentan los principales riesgos y sus estrategias de mitigación:

**1. Retrasos en la Integración de APIs Externas**

* **Riesgo**: Dependencia de terceros puede causar demoras en la disponibilidad o cambios inesperados en las APIs.
* **Mitigación**:
  + Planificación anticipada y comunicación constante con los proveedores de APIs.
  + Implementación de capas de abstracción para facilitar adaptaciones a cambios.
  + Preparación de datos mock para pruebas en caso de indisponibilidad temporal.

**2. Fallos en el Modelo de Inteligencia Artificial**

* **Riesgo**: El modelo de IA puede no clasificar correctamente los tickets o proporcionar soluciones inadecuadas.
* **Mitigación**:
  + Realizar pruebas exhaustivas con datos de entrenamiento y validación.
  + Ajustar y retrain del modelo periódicamente con nuevos datos.
  + Establecer un mecanismo de supervisión humana para revisar y corregir errores.

**3. Vulnerabilidades de Seguridad**

* **Riesgo**: Posibles ataques cibernéticos que comprometan la integridad y confidencialidad de los datos.
* **Mitigación**:
  + Implementación de prácticas de desarrollo seguro (OWASP Top Ten).
  + Auditorías de seguridad y pruebas de penetración periódicas.
  + Actualizaciones regulares de software y parches de seguridad.

**4. Resistencia al Cambio por Parte del Personal**

* **Riesgo**: Los usuarios pueden mostrar reticencia a adoptar el nuevo sistema.
* **Mitigación**:
  + Realizar sesiones de capacitación y entrenamiento.
  + Involucrar al personal en etapas tempranas del proyecto para obtener retroalimentación.
  + Destacar los beneficios y mejoras que el sistema aporta a sus labores diarias.

**5. Sobrecarga del Servidor**

* **Riesgo**: Incremento inesperado en el uso del sistema puede afectar el rendimiento.
* **Mitigación**:
  + Monitoreo constante del rendimiento y uso de recursos.
  + Implementación de escalabilidad vertical u horizontal según necesidad.
  + Optimización del código y consultas a la base de datos.

**6. Dependencia de Servicios de Terceros**

* **Riesgo**: Fallos o cambios en servicios como APIs de IA o servicios de correo electrónico.
* **Mitigación**:
  + Tener alternativas o proveedores secundarios.
  + Diseñar el sistema para manejar fallos de servicios externos sin afectar la funcionalidad principal.
  + Contratos de nivel de servicio (SLA) con proveedores críticos.

**7. Problemas de Compatibilidad**

* **Riesgo**: Incompatibilidad entre tecnologías o versiones de software.
* **Mitigación**:
  + Definir y documentar las versiones de software y dependencias desde el inicio.
  + Realizar pruebas de integración continuas.
  + Mantener un entorno de desarrollo y producción alineados.

# **11. Cronograma del Proyecto**

El proyecto se desarrollará en un período de **5 meses** (20 semanas), dividido en tres fases principales. A continuación, se presenta un cronograma detallado:

**Fase 1: Planificación y Análisis (Semanas 1-4)**

* **Semana 1**:
  + Reuniones iniciales con stakeholders.
  + Definición de requisitos funcionales y no funcionales.
  + Análisis de viabilidad y riesgos.
* **Semana 2**:
  + Diseño de la arquitectura del sistema.
  + Definición de tecnologías y herramientas.
  + Planificación del proyecto y asignación de tareas.
* **Semana 3-4**:
  + Diseño de la base de datos.
  + Creación de prototipos de interfaces.
  + Definición de casos de uso y diagramas UML.

**Fase 2: Desarrollo e Implementación (Semanas 5-15)**

* **Semana 5-7**:
  + Desarrollo del módulo de autenticación y seguridad.
  + Implementación del sistema de roles y permisos.
  + Integración del captcha en el proceso de inicio de sesión.
* **Semana 8-10**:
  + Desarrollo del módulo de gestión de tickets.
  + Implementación de la línea de tiempo de soporte.
  + Integración con APIs de **itplusnet** para gestión logística.
* **Semana 11-12**:
  + Desarrollo del modelo de machine learning para clasificación de tickets.
  + Entrenamiento y validación del modelo.
  + Integración del modelo con el CRM.
* **Semana 13-14**:
  + Implementación del asistente inteligente basado en ChatGPT.
  + Desarrollo de la API propia para comunicación con el asistente.
  + Pruebas y ajustes en el flujo de asistencia al cliente.
* **Semana 15**:
  + Desarrollo del dashboard interactivo y sistema de reportes.
  + Integración de gráficos y métricas clave.
  + Optimización de la interfaz de usuario.

**Fase 3: Pruebas, Despliegue y Capacitación (Semanas 16-20)**

* **Semana 16-17**:
  + Pruebas unitarias y de integración.
  + Corrección de errores y optimización de rendimiento.
  + Pruebas de seguridad y auditorías.
* **Semana 18**:
  + Implementación de feedback de usuarios piloto.
  + Ajustes finales en funcionalidades y diseño.
* **Semana 19**:
  + Preparación del entorno de producción.
  + Despliegue del sistema en servidores finales.
  + Configuración de dominios y certificados SSL.
* **Semana 20**:
  + Sesiones de capacitación para el personal de **itplusnet**.
  + Entrega de documentación y manuales de usuario.
  + Lanzamiento oficial del sistema y seguimiento inicial.

# **12. Recursos y Presupuesto**

**Recursos Humanos**

* **Equipo de Desarrollo**:
  + **1 Líder de Proyecto**: Coordinación general y gestión del proyecto.
  + **2 Desarrolladores Backend**: Especializados en PHP y Laravel.
  + **1 Desarrollador Frontend**: Experto en HTML, CSS, JavaScript y Bootstrap.
  + **1 Especialista en Inteligencia Artificial**: Desarrollo e integración de modelos de machine learning y ChatGPT.
  + **1 Diseñador UI/UX**: Diseño de interfaces y experiencia de usuario.
  + **1 Ingeniero de DevOps**: Configuración de servidores, despliegue y escalabilidad.
  + **1 Analista de QA**: Pruebas de calidad y seguridad.
* **Recursos Adicionales**:
  + **Consultor de Seguridad**: Auditorías y recomendaciones de seguridad.
  + **Capacitador**: Formación al personal de **itplusnet**.

**Herramientas y Software**

* **Licencias y Servicios**:
  + **Servicios de Hosting**: Costos asociados a servidores y almacenamiento.
  + **APIs de Inteligencia Artificial**: Suscripciones a servicios como OpenAI.
  + **Servicios de Correo Electrónico**: SendGrid, Mailgun u otros.
  + **Herramientas de Monitoreo**: New Relic, Sentry, etc.
* **Equipamiento**:
  + Computadoras y dispositivos para desarrollo y pruebas.
  + Software de diseño y prototipado (Adobe XD, Figma).

**Presupuesto Estimado**

* **Recursos Humanos**: [Monto estimado según tarifas y duración del proyecto].
* **Herramientas y Licencias**: [Detalle de costos mensuales y totales].
* **Infraestructura**: [Costos de servidores, servicios en la nube, seguridad].
* **Contingencias**: Reserva del 10-15% del presupuesto para imprevistos.
* **Total Estimado**: [Suma de todos los costos].

*Nota*: Los montos específicos deben ser calculados en base a tarifas locales y acuerdos con proveedores.

# **13. Beneficios y Valor Agregado**

* **Mejora en la Eficiencia Operativa**:
  + Automatización de procesos que reducen el tiempo y esfuerzo manual.
  + Priorización inteligente de tickets que optimiza la asignación de recursos.
* **Incremento en la Satisfacción del Cliente**:
  + Respuestas más rápidas y soluciones inmediatas gracias al asistente inteligente.
  + Transparencia y seguimiento detallado de sus solicitudes.
* **Fortalecimiento de la Seguridad**:
  + Protección de datos sensibles mediante autenticación robusta y medidas de seguridad avanzadas.
  + Confianza de los clientes en la integridad y confidencialidad de sus datos.
* **Decisiones Informadas**:
  + Acceso a datos y métricas en tiempo real que facilitan la toma de decisiones estratégicas.
  + Identificación de tendencias y áreas de mejora.
* **Innovación y Competitividad**:
  + Uso de inteligencia artificial que posiciona a **itplusnet** como una empresa vanguardista.
  + Diferenciación en el mercado al ofrecer servicios avanzados y personalizados.
* **Escalabilidad y Preparación para el Futuro**:
  + Arquitectura diseñada para adaptarse a nuevas necesidades y crecimiento.
  + Base sólida para integrar futuras tecnologías y expandir funcionalidades.

# **14. Conclusión**

El desarrollo e implementación del **itplusnet-CRM** representa un paso significativo para **itplusnet** en su camino hacia la excelencia operativa y la innovación tecnológica. Al abordar las deficiencias actuales en la gestión de soporte y al integrar soluciones avanzadas como inteligencia artificial y aprendizaje automático, la empresa no solo mejora su eficiencia interna, sino que también eleva la experiencia de sus clientes a un nuevo nivel.

La incorporación del asistente inteligente basado en *ChatGPT* permite ofrecer soluciones inmediatas y personalizadas a los clientes, reduciendo tiempos de espera y mejorando la satisfacción. Además, la automatización de tareas y la priorización inteligente de tickets optimizan la asignación de recursos y agilizan los procesos internos.

Este proyecto ofrece una solución integral que impactará positivamente en múltiples aspectos del negocio: desde la satisfacción del cliente hasta la toma de decisiones estratégicas basadas en datos. La inversión en recursos humanos, tecnológicos y financieros está justificada por los beneficios tangibles y el valor agregado que el sistema aportará a corto y largo plazo.

En un entorno empresarial cada vez más competitivo y dinámico, **Itplusnet** se posicionará como un referente en la implementación de tecnologías de vanguardia, reforzando su compromiso con la calidad y la innovación. El **ITPLUSNET-CRM** no es solo una herramienta tecnológica, sino una estrategia integral que impulsa el crecimiento sostenible y la capacidad de adaptación de la empresa frente a los desafíos futuros.

**Anexos**: Se podrán incluir diagramas UML, esquemas de base de datos, mockups de interfaces y otros documentos relevantes para complementar la información presentada en este documento.