

HelpMe!



ÍNDICE

1.Introducción.....	3
2.Descripción del Proyecto	3
Objetivos Especificos.....	3
Objetivos Generales.....	3
Módulos Docentes Relacionados.....	4
Agradecimientos	4
3. Análisis	4
Diseño.....	4
Definición de Requerimientos.....	5
Pruebas, Depuración y Documentación	8
4.Conclusiones:.....	9
Desafíos durante el desarrollo.....	9
5.Líneas futuras:.....	10
Generales.....	10
6.Bibliografía.	11

1.Introducción

- **Descripción proyecto:**

En el dinámico y exigente entorno empresarial actual, la eficacia en la gestión de soporte técnico se ha convertido en un pilar fundamental para el mantenimiento de operaciones fluidas y eficientes. Este proyecto se centra en el desarrollo de una solución para la gestión de tickets de soporte dentro o fuera de las empresas, una aplicación diseñada para optimizar la comunicación y seguimiento de incidencias técnicas.

2.Descripción del Proyecto

Este sistema de gestión de tickets está diseñado para permitir a los usuarios generar y rastrear tickets de soporte a través de una aplicación intuitiva y fácil de usar. La aplicación no solo facilita la creación de tickets, sino que también asegura que los usuarios se mantengan informados sobre el progreso, mejorando así la transparencia y eficiencia del proceso de soporte técnico.

- **Objetivos Generales**

El objetivo principal del proyecto es desarrollar una solución integral que simplifique y agilice la gestión de tickets de soporte en el entorno empresarial. Se busca mejorar la comunicación entre los usuarios y el equipo de soporte técnico, y proporcionar una plataforma centralizada para el seguimiento eficiente de incidencias.

- **Objetivos Específicos**

- Facilitar la creación y seguimiento de tickets de soporte por parte de los usuarios.
- Asegurar una comunicación fluida y constante entre usuarios y personal de soporte.
- Implementar un sistema de notificaciones para mantener informados a los usuarios sobre el estado de sus tickets.

- Desarrollar una interfaz de usuario intuitiva y accesible para una experiencia de usuario óptima.

- **Módulos Docentes Relacionados**

Este proyecto integra conocimientos y habilidades adquiridos en varios módulos del Ciclo Superior de Desarrollo de Aplicaciones Multiplataforma, incluyendo programación, diseño de interfaces de usuario, gestión de bases de datos y comunicación en red.

- **Agradecimientos**

Quiero dar las gracias a todos los profesores que me acompañaron este año. Su ayuda y consejos han sido clave en mi proyecto. Aprecio mucho todo lo que me enseñaron y el tiempo que dedicaron. ¡Gracias por todo!

3. Análisis

- **Identificación de Necesidades**

He observado que numerosas empresas, especialmente las pequeñas y medianas, no están adecuadamente equipadas para gestionar incidencias de manera eficiente y rápida. Además, tanto empleados como clientes a menudo se encuentran desorientados sobre cómo y con quién gestionar estas incidencias.

Por otro lado, es común que los equipos de IT manejen estas incidencias a través de correo electrónico o llamadas telefónicas. Este método puede complicar el seguimiento efectivo de las incidencias, aumentando el riesgo de que estas se retrasen o incluso se olviden.

- **Definición de Requerimientos**

Requisitos Funcionales

Gestión de Usuarios:

- Creación de Usuarios: El sistema debe permitir al perfil de administrador crear cuentas de usuario. Esto incluye especificar datos como nombre, correo electrónico y rol dentro del sistema.
- Edición de Usuarios: El administrador debe poder editar la información de las cuentas de usuario, incluyendo la capacidad de cambiar roles o actualizar detalles de contacto.
- Eliminación de Usuarios: Debe existir la funcionalidad para que el administrador pueda eliminar cuentas de usuario, garantizando que esta acción cumpla con las normativas de protección de datos pertinentes.

Gestión de Tickets:

- Base de Datos: para almacenar información del usuario que crea el ticket.
- Creación de Tickets: Los usuarios deben poder crear tickets de soporte, especificando detalles como la descripción del problema, categoría o prioridad, y cualquier otra información relevante.
- Priorización de Tickets: Capacidad para priorizar tickets basándose en ciertos criterios como urgencia.
- Generación de alertas: Envío de alertas para mantener informados a los usuarios.
- Edición de Tickets: Los usuarios con los permisos necesarios (como los administradores o personal de soporte) deben poder editar los detalles de los tickets.
- Resolución y Finalización de Tickets: El sistema debe permitir a los usuarios autorizados cambiar el estado de los tickets a "resuelto" o "cerrado", incluyendo la capacidad de añadir comentarios o soluciones implementadas.
- Integración con Otras Herramientas: La aplicación debe ser capaz de integrarse con otras herramientas de la empresa, como sistemas de CRM o bases de datos existentes.

- **Diseño:**

Para el diseño tanto en la aplicación de escritorio como en la app móvil decidí usar binding y el patrón MVVM (Model-View-ViewModel) ya que nos permite facilitar la sincronización de datos entre la interfaz de usuario y el modelo de datos, lo que es crucial para la escalabilidad y mantenimiento de nuestras aplicaciones.

Aplicación de Escritorio (Cliente de Soporte)

- **Data Binding en WPF:**

Vinculación de Datos de la UI a los ViewModels: Cada elemento de la interfaz de usuario, como listas, formularios y botones, está vinculado a propiedades en los respectivos ViewModels. Esto asegura que los cambios en la interfaz de usuario se reflejan automáticamente en el modelo de datos y viceversa.

Notificaciones de Cambios: Implementación del interfaz INotifyPropertyChanged en los ViewModels para actualizar la UI cuando cambian los datos del modelo.

- **Validación y Gestión de Errores:**

Uso del binding para validar la entrada de datos en formularios y mostrar mensajes de error o validación directamente en la UI.

Aplicación Móvil (Seguimiento de Tickets)

- **Binding en la Interfaz de Usuario:**

Implementación similar de binding en la aplicación complementaria para asegurar una experiencia de usuario coherente.

Vinculación de los datos de tickets (como estado, comentarios, etc.) a los elementos de la UI para una visualización y actualización en tiempo real.

- **Implementación para la Aplicación de Tickets de Soporte:**
- Implementación de Interfaces de Usuario en WPF:
 - Se desarrollan las interfaces de usuario según el diseño, utilizando XAML para la definición visual y la disposición de los elementos de la interfaz.
 - Se presta atención especial a la usabilidad y accesibilidad para garantizar una experiencia de usuario óptima.
- Creación de Clases ViewModel en C#:
 - Se implementan los ViewModels para cada vista, manejando la lógica de presentación y el procesamiento de eventos de la UI.
 - Estos ViewModels actúan como un intermediario entre la UI y el modelo de datos, facilitando la separación de preocupaciones.
- Modelo de Datos y Acceso a Datos con Dapper:
 - Se definen clases de modelo para representar entidades clave como tickets y usuarios.
 - Se utiliza Dapper para realizar operaciones de base de datos, optimizando el rendimiento y facilitando la gestión de datos.
- Integración de ViewModel y Modelos de Datos:
 - Se implementa el binding entre la UI y los modelos de datos a través de los ViewModels, utilizando INotifyPropertyChanged para reflejar los cambios en la UI de manera reactiva.
- **UI y Binding para Creación y Seguimiento de Tickets(App):**
 - Desarrollo de la interfaz de usuario enfocada en la funcionalidad de creación y seguimiento de tickets.
 - Implementación eficaz del binding para actualizar dinámicamente la UI en respuesta a los cambios en el estado de los tickets.

- **Comunicación con el Backend a través de Sockets:**

- Desarrollo de la lógica para establecer y manejar la comunicación con el backend mediante sockets, garantizando una interacción fluida y en tiempo real.

- **Servidor de Sockets en C#:**

- Creación de un servidor robusto y eficiente que gestione las solicitudes entrantes de las aplicaciones cliente.
- Implementación de la lógica para procesar estas solicitudes y realizar operaciones correspondientes en la base de datos.
- Uso de Dapper para interactuar con la base de datos, implementando consultas y procedimientos almacenados para manejar las operaciones de datos de manera eficiente.

- **Seguridad y Manejo de Excepciones**

- Robusto Manejo de Excepciones y Errores: Se implementan estrategias para manejar excepciones y errores tanto en la aplicación como en el backend, asegurando la estabilidad y la confiabilidad del sistema.
- Seguridad en la Comunicación de Datos: Se garantiza la seguridad en la transferencia de datos, utilizando encriptación y protocolos seguros para proteger la información sensible.

- **Pruebas, Depuración y Documentación**

- Pruebas Unitarias y de Integración: Se llevan a cabo pruebas exhaustivas para validar la funcionalidad y la integración de los distintos componentes del sistema.
- Documentación del Código Fuente: Se documenta el código fuente de manera exhaustiva, proporcionando comentarios que faciliten la comprensión y el mantenimiento futuro del sistema.

4.Conclusiones:

Cuando se empezó el proyecto los objetivos que tenía era conseguir que tanto la aplicación de móvil como la aplicación de escritorio pudieran complementarse para dar una buena experiencia al usuario. Por supuesto, conseguir cumplir una necesidad como es la gestión de tickets que es tan importante dentro de una empresa.

Finalizado el plazo puedo decir que tanto la aplicación de escritorio como la aplicación móvil puede realizar sus funciones principales. Pero aun así queda tiempo de desarrollo para conseguir un resultado más profesional.

Me hubiera gustado que la aplicación móvil permitirá al usuario realizar más acciones como por ejemplo borrar tickets, modificar los datos del usuario, uso de biometría para iniciar sesión, notificaciones, etc.

- **Desafíos durante el desarrollo:**

- **WPF:**

Durante el desarrollo del proyecto me encontré con varios desafíos que vale la pena mencionar. Por una parte cuando empecé a desarrollar la aplicación de escritorio decidí meterme en WPF para el desarrollo de la interfaz gráfica. Mis conocimientos en WPF eran básicos y supuso un reto. Esto también me llevo a cometer errores los cuales dejare reflejados en las mejoras futuras.

- **MVVM**

Desde hace unos años trabajo usando la arquitectura MVVM en aplicaciones móviles y decidí implementarlo en mi aplicación de escritorio. Esto me llevó un tiempo aprender cómo implementarlo en mi proyecto.

- **Servidor Socket**

Cuando empecé a montar el servidor mediante sockets me encontré con un problema y era que nunca había enviado grandes cantidades de datos a través de sockets. Después de una búsqueda por documentación online decidí usar JSON para el envío de la información ya que es el método más usado para realizar estos tipos de conexiones. Aparte la facilidad para convertir los envíos JSON en objetos.

5. Líneas futuras:

- **Generales:**

Me gustaría mejorar el envío de tickets dando la posibilidad de añadir imágenes ya que facilitan su revisión. Tanto el envío como luego que se muestre en la aplicación de escritorio.

- **Aplicación de escritorio:**

Para un futuro me gustaría cambiar el comportamiento de las vistas en WPF ya que están pensadas como ventanas y existe la posibilidad de generar páginas para ir abriendo las ventanas dentro de una principal haciendo que se vea más profesional.

El campo email¹ fue añadido en los usuarios, pero no se permite insertar cuando se crea el usuario.

- **Aplicación de móvil:**

- Me gustaría añadir una barra de navegación para añadir opciones como cerrar sesión, ayuda, políticas etc.
- Permitir filtrar en el listado de tickets ya sea por estado, fechas, tipo de incidencia.
- Aplicación para IOS.
- Añadir la opción de enviar imágenes.

- **Servicio Windows:**

Este servicio lo pensé con un servidor de alertas. Podría ser muy interesante añadir avisos diarios de informes (Tickets diario, tiempos de resolución, número de urgencias) para enviar a nuestros clientes.

6. Bibliografía.

Tutoriales y Guías Online sobre WPF y ViewModel

- Dot Net Perls. Tutorial sobre WPF y XAML. Recuperado de [Dot Net Perls](#)
AngelSix. WPF UI Programming Tutorial.
 - Enlace: [AngelSix en YouTube](#)
 - Descripción: Una serie de tutoriales que abarca la creación de interfaces de usuario en WPF, con ejemplos prácticos y explicaciones claras.
- Recursos y Artículos sobre Dapper
 - C# Corner. Artículos sobre el uso de Dapper en .NET. Recuperado de [C# Corner](#)
- Recursos sobre Desarrollo de Aplicaciones Android
 - Codecademy. Curso de Desarrollo de Aplicaciones Android. Recuperado de [Codecademy](#)
 - Raywenderlich. Android & Kotlin Tutorials. Recuperado de [Raywenderlich](#)
 - Videos y Tutoriales sobre MVVM y Binding en Android
 - Stevdza-San. MVVM en Android. Recuperado de [YouTube](#)
 - Coding in Flow. MVVM Tutorial for Android. Recuperado de [YouTube](#)

https://www.youtube.com/watch?v=a9vpykOf7ck&ab_channel=Programaci%C3%B3nAndroidbyAristiDevs

Recursos sobre Sockets y Comunicación en C#

- GeeksforGeeks. Socket Programming in C#. Recuperado de GeeksforGeeks [GeeksforGeeks](#)
- The Code Project. C# Socket Programming. Recuperado de The Code Project [The Code Project](#)

Videos Educativos sobre JSON y Comunicación de Datos

- Traversy Media. JSON Crash Course. Recuperado de YouTube [YouTube](#)
- Academind. Understanding JSON - JSON Tutorial for Beginners. Recuperado de [YouTube](#)