**UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA**

**FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN**

**LICENCIATURA EN INGENIERIA EN SISTEMAS DE INFORMACIÓN**

**CURSO: 012 - PROGRAMACIÓN I**

Imagen que contiene firmar, alimentos, palo, señal

El contenido generado por IA puede ser incorrecto.

**Manual Técnico**

**NOMBRE DEL ALUMNO: Carne No.:**

**Jefferson Francisco Patzán Pirir 7590-24-888**

**Sistema de Gestión de Tickets**

El proyecto tiene como objetivo ser altamente eficiente en el almacenamiento y manejo de información relacionada con la gestión de tickets, facilitando la atención de solicitudes para la solución de problemas en departamentos específicos de la empresa. En lugar de utilizar archivos de texto, el sistema almacena toda la información en una base de datos robusta, lo que permite una actualización y consulta dinámica y escalable de los registros.

El sistema está diseñado para ofrecer una experiencia intuitiva y fácil de usar, permitiendo que cualquier usuario, independientemente de su nivel técnico, interactúe de manera efectiva con la aplicación. Los tickets se clasifican en categorías específicas y sus características como departamento asignado, prioridad, descripción, código identificador, y estado están claramente definidas y organizadas. Esta estructura ordenada agiliza la entrada de datos, facilitando consultas rápidas y modificaciones precisas, lo que a su vez mejora el control administrativo y operativo.

El alcance del proyecto incluye el acceso completo al programa para usuarios generales, mientras que los administradores, gracias a sus privilegios, podrán gestionar todas las funcionalidades del sistema. Esto abarca la administración de tickets, la creación y modificación de usuarios, el registro de nuevos departamentos, la asignación de roles y permisos y la configuración de las políticas de acceso, asegurando que cada usuario interactúe únicamente con las funciones que le corresponden según su perfil.

**Objetivos Técnicos del Proyecto**

* **Implementación de Algoritmos para el Acceso a la Base de Datos:** Desarrollar métodos eficientes para almacenar, recuperar y actualizar la información de tickets, usuarios, roles, permisos y departamentos en una base de datos relacional. Esto sustituye el antiguo método basado en archivos de texto, ofreciendo mayor seguridad, integridad y velocidad de acceso.
* **Arquitectura Modular:** Utilizar una estructura modular que permita agregar o modificar funcionalidades sin afectar el funcionamiento global del sistema. Esta arquitectura favorece la escalabilidad y el mantenimiento, haciendo posible integrar nuevas características de forma ágil.
* **Gestión Dinámica de Categorías y Atributos:** Permitir que el usuario registre nuevas categorías de tickets y especifique atributos adicionales en la base de datos, adaptándose a las necesidades específicas de cada departamento sin sobrescribir la integridad del sistema.
* **Integración de Control de Acceso y Autenticación:** Implementar un sistema de inicio de sesión que, según el rol asignado, determine los permisos del usuario. Esto permitirá registrar nuevos usuarios, asignar roles y configurar permisos de acceso, garantizando que los datos sean gestionados de forma segura y conforme a las políticas de la empresa.
* **Interfaz Gráfica con JavaFX:** El sistema utiliza Java para la lógica de negocio y JavaFX para la interfaz gráfica, asegurando una experiencia visual interactiva y moderna. Esto permite que la aplicación no solo sea funcional, sino también atractiva y fácil de navegar para el usuario final.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama

El contenido generado por IA puede ser incorrecto.

**A continuación, se detallan las tecnologías utilizadas en el desarrollo del Sistema de Gestión de Tickets:**

1. **Lenguaje de Programación Java:** Se eligió Java debido a su robustez, portabilidad y amplio soporte en entornos empresariales. Su orientación a objetos y rica biblioteca estándar facilitan la construcción de una solución modular y escalable, ideal para la gestión y seguimiento de tickets.
2. **IDE NetBeans:** NetBeans se utiliza como entorno de desarrollo integrado por sus potentes herramientas para la edición, depuración y diseño visual. Su integración con JavaFX agiliza la creación de interfaces gráficas interactivas y modernas, lo que mejora la experiencia del usuario final.
3. **JDK (Java Development Kit):** El JDK es esencial para la compilación, ejecución y depuración de la aplicación. Proporciona todas las herramientas y librerías necesarias, como el compilador javac y el Java Runtime Environment (JRE), asegurando que la aplicación se ejecute de acuerdo con los estándares actuales de desarrollo en Java.
4. **JavaFX:** JavaFX se emplea para desarrollar la interfaz gráfica del sistema. Esta tecnología permite crear aplicaciones visualmente atractivas, con componentes interactivos y diseños responsivos, facilitando la navegación y el manejo de la información por parte del usuario.
5. **Base de Datos PostgreSQL:** Se utiliza PostgreSQL como sistema de administración de bases de datos relacional para almacenar y gestionar de manera segura la información del sistema, incluyendo tickets, usuarios, roles, permisos y departamentos. PostgreSQL destaca por su robustez y escalabilidad, lo que lo convierte en una opción ideal para aplicaciones empresariales que requieran integridad y rendimiento en la gestión de datos

**Instrucciones de configuración del entorno de trabajo**

1. Requisitos Previos
2. Sistema Operativo: Windows
3. Java Development Kit (JDK): Asegúrate de tener instalado JDK en el computador

**Instalación del JDK**

• Descarga el JDK desde el sitio oficial de Oracle

• Sigue las instrucciones de instalación proporcionadas en el sitio 3. Instalación de un Entorno de Desarrollo Integrado (IDE)

• Recomendado: NetBeans

• NetBeans: 1. Descarga NetBeans en el sitio oficial 2. Sigue las instrucciones de instalación proporcionadas en el sitio

**Proceso de compilación del proyecto Compilación y Ejecución:**

* Compila el proyecto a través del IDE o utilizando el siguiente comando en el terminal (asegúrate de estar en el directorio src): javac “nombreDelArchivo”.java
* Ejecuta el proyecto: java “nombreDelArchivo”

**Sección de Solución de Problemas Comunes**

En esta sección se recopilan algunos de los errores y situaciones más comunes que pueden ocurrir durante la operación del sistema de gestión de tickets, junto con sugerencias para solucionarlos:

**Problemas de Conexión con la Base de Datos**

* **Error:** "Error al conectar a la base de datos"
* **Posibles Causas:**
  + PostgreSQL no se encuentra en ejecución.
  + Configuración incorrecta de credenciales o parámetros de conexión (host, puerto, usuario, contraseña).
  + Problemas en la red o en la configuración del firewall.
* **Solución Recomendada:**
  + Verifica que el servicio de PostgreSQL esté activo y corriendo.
  + Revisa el archivo de configuración del sistema para confirmar que las credenciales y parámetros de conexión sean correctos.
  + Comprueba la conectividad de red y ajusta las configuraciones del firewall si es necesario.

**Errores en la Carga de la Interfaz JavaFX**

* **Error:** "No se pudo cargar el módulo JavaFX" o "Error al inicializar la interfaz gráfica"
* **Posibles Causas:**
  + La ruta del JavaFX SDK no está correctamente configurada.
  + Falta la inclusión de los módulos necesarios en la línea de comandos o en la configuración del IDE.
* **Solución Recomendada:**
  + Asegúrate de haber configurado correctamente el parámetro --module-path para apuntar a la ubicación del JavaFX SDK y de haber añadido los módulos requeridos (por ejemplo, --add-modules=javafx.controls,javafx.fxml).
  + Verifica la documentación del JavaFX SDK y confirma que corresponde a la versión de JDK que utilizas.

**Errores de Autenticación y Permisos**

* **Error:** "Credenciales inválidas" o "Acceso denegado"
* **Posibles Causas:**
  + Datos de autenticación incorrectos o inexistentes en la base de datos.
  + Configuración incorrecta de roles y permisos en el sistema.
* **Solución Recomendada:**
  + Comprueba que el usuario ingrese los datos de inicio de sesión correctos.
  + Revisa la base de datos para confirmar que los registros de usuarios estén correctamente configurados y que los roles asignados cuenten con los permisos adecuados.

**Errores de Validación y Duplicidad de Datos**

* **Error:** "El código ya existe" o "Falta información en los campos requeridos"
* **Posibles Causas:**
  + Intento de agregar un registro con un código duplicado.
  + Omitir campos obligatorios durante el ingreso de información (por ejemplo, al generar un nuevo ticket o usuario).
* **Solución Recomendada:**
  + Asegúrate de que cada registro tenga un identificador único y que no se repitan los códigos.
  + Valida previamente todos los campos de entrada para garantizar que se provea toda la información requerida.

**Excepciones y Errores No Controlados**

* **Error:** Excepciones en tiempo de ejecución (por ejemplo, NullPointerException)
* **Posibles Causas:**
  + Acceso a objetos o variables que no han sido debidamente inicializados.
  + Falta de manejo adecuado de excepciones en ciertas funcionalidades del sistema.
* **Solución Recomendada:**
  + Revisa los logs de la aplicación para identificar el origen exacto del error.
  + Implementa un manejo de excepciones robusto en el código, asegurando que se capturen y gestionen los errores de forma que el usuario reciba mensajes claros y se eviten caídas inesperadas del sistema.