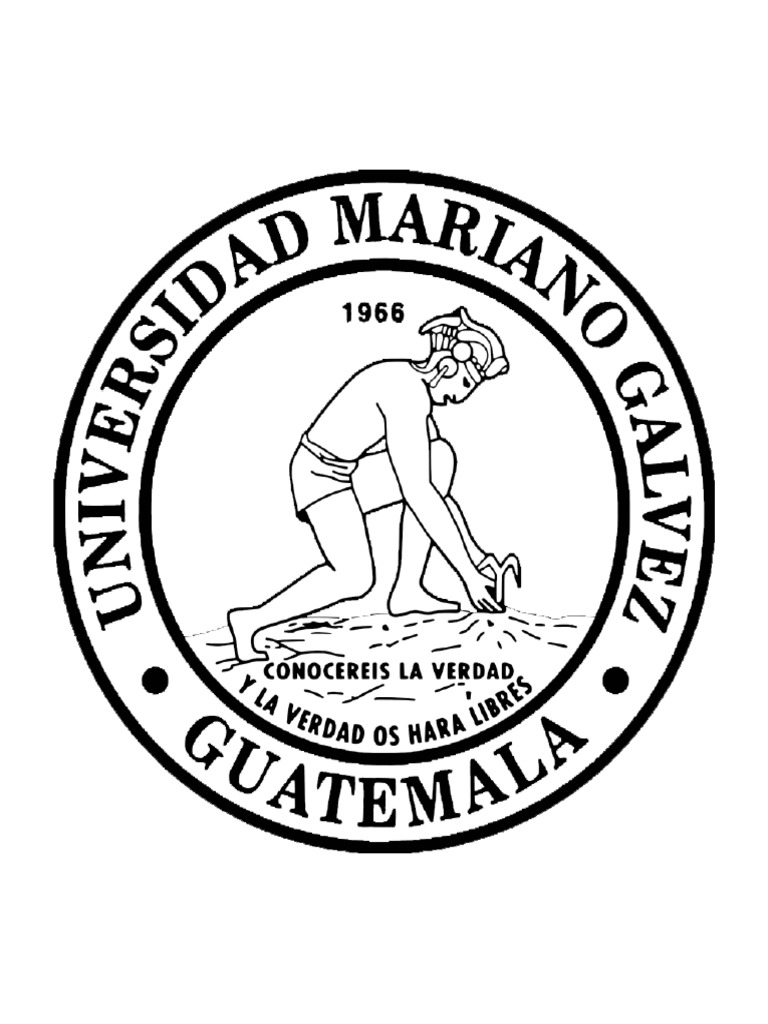
**UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA**

**FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN**



**JULIO ABRAHAN CHET CHAMALE**

**7590-23-5400**

**GUATEMALA, MAYO DEL 2024**

**Manual Técnico - Sistema de Tickets**

**1. Información general del sistema**

Nombre del sistema: Sistema de Tickets

Autor: Julio Abrahan Chet Chamale

Correo: abrahanchet@gmail.com

Este sistema fue desarrollado como proyecto y utiliza el lenguaje Java con JavaFX para la interfaz gráfica y PostgreSQL como sistema de gestión de base de datos, administrado mediante pgAdmin4. El sistema está diseñado bajo el patrón de arquitectura Modelo-Vista-Controlador (MVC) y fue desarrollado en el entorno IntelliJ IDEA.

**2. Requisitos técnicos**

Software necesario:

- Java Development Kit (JDK) 17 o superior

- PostgreSQL 13 o superior

- pgAdmin4

- JavaFX SDK

- IDE recomendado: IntelliJ IDEA

Dependencias utilizadas:

- JDBC PostgreSQL driver

- Librerías JavaFX

**3. Estructura de base de datos**

Nombre de la base de datos: Tickets

Tablas principales:

- configuracion\_sistema

- estado\_ticket

- historial\_configuracion

- niveles\_prioridad

- notas

- tickets

- usuarios

- departamentos

- tecnico\_departamento

- transiciones\_estado

La base de datos incluye relaciones entre las tablas mediante claves foráneas. Algunas restricciones aplicadas incluyen claves únicas y validaciones de integridad referencial.

**4. Arquitectura del sistema**

El sistema está basado en una arquitectura MVC (Modelo-Vista-Controlador). Las clases se organizan en paquetes:

- modelo: contiene las clases que representan las entidades de base de datos

- dao: define las interfaces y sus implementaciones para acceso a datos

- controllers: contiene los controladores JavaFX que manejan la lógica de interfaz

- util: incluye utilidades como la clase Conexion.java para la conexión a la base de datos

- view: contiene los archivos FXML para las vistas

**5. Casos de uso implementados**

- Configurar parámetros del sistema

- Registrar departamentos o áreas de trabajo

- Registrar usuarios iniciales

- Definir estados de los tickets

- Gestionar tickets: crear, asignar, cerrar

**6. Descripción de clases principales**

Las clases principales están organizadas por función:

- Controladores: LoginController, PrincipalController, CrearTicketController, etc.

- DAOs y DAOsImpl: interfaces y clases concretas para operaciones CRUD

- Modelos: clases como Ticket, Usuario, EstadoTicket, Departamento, etc.

- Conexion.java: gestiona la conexión a la base de datos PostgreSQL

**7. Manejo de roles**

El sistema soporta distintos tipos de usuarios definidos en la tabla "usuarios": administrador, técnico y usuario.

- Administrador: acceso completo a todas las funcionalidades

- Técnico: puede gestionar y editar tickets

- Usuario: solo puede crear tickets

La visibilidad de botones y funcionalidades se adapta según el rol que haya iniciado sesión.

**8. Validaciones y restricciones**

- Validación de campos obligatorios en formularios (nombre, correo, contraseña)

- Restricción de acceso según tipo de usuario

- Control de transiciones válidas de estado para los tickets

- Restricciones de integridad entre tablas (técnico\_departamento, transiciones\_estado)

**9. Procedimiento de despliegue**

1. Crear una base de datos en PostgreSQL llamada "Tickets" usando pgAdmin4.

2. Crear las tablas manualmente o usando scripts como los siguientes:

CREATE TABLE departamentos (...);

CREATE TABLE tecnico\_departamento (...);

ALTER TABLE estado\_ticket ADD COLUMN descripcion TEXT;

ALTER TABLE estado\_ticket ADD COLUMN es\_final BOOLEAN DEFAULT FALSE;

CREATE TABLE transiciones\_estado (...);

3. Configurar los parámetros de conexión en la clase Conexion.java.

4. Ejecutar el sistema desde el entorno de desarrollo IntelliJ.

**10. Futuras mejoras**

- Generación de reportes y estadísticas

- Notificaciones por correo

- Historial de cambios en tickets

- Mejora en la seguridad de contraseñas