

Sistema de Control de Acceso Zona Acme - Documentación del Proyecto

1. Introducción

1.1 Propósito

El Sistema de Control de Acceso para el Complejo Empresarial Zona Acme tiene como objetivo principal gestionar de manera eficiente y segura el acceso de trabajadores e invitados al complejo, asegurando la trazabilidad de todos los movimientos y proporcionando herramientas amigables para guardas de seguridad, supervisores y funcionarios de empresas.

1.2 Contexto del Problema

El complejo empresarial requería una solución tecnológica que permitiera:

- Registrar y controlar la entrada y salida de personal
- Mantener un registro detallado y trazable de todas las operaciones
- Generar informes relevantes sobre usuarios y accesos
- Proporcionar una interfaz intuitiva para personal con habilidades informáticas limitadas

2. Descripción Técnica

2.1 Arquitectura del Sistema

- **Patrón Arquitectónico:** Modelo-Vista-Controlador (MVC)
- **Modelo Cliente-Servidor**
- **Lenguaje de Programación:** Java (JDK 17+)

2.2 Tecnologías Utilizadas

- **Interfaz Gráfica:** Swing
- **Base de Datos:** MySQL
- **Conexión a BD:** JDBC
- **Herramientas de Concurrencia:** Hilos para operaciones en tiempo real

3. Funcionalidades Principales

3.1 Gestión de Usuarios

- Roles: Superusuarios, supervisores, guardas y funcionarios
- Control de activación/inactivación de usuarios
- Gestión jerárquica de creación de usuarios

3.2 Registro de Accesos

- Registro de entradas y salidas mediante documento de identidad
- Control de acceso con sistema de anotaciones
- Validación de ingreso para personas registradas

3.3 Control de Vehículos

- Registro de placas de vehículos
- Registro individual de pasajeros

3.4 Manejo de Incidentes

- Registro de anotaciones por supervisores
- Sistema de restricciones de acceso con trazabilidad completa

3.5 Reportes y Trazabilidad

- Reportes de usuarios activos/inactivos
- Informes de accesos en rangos de fechas
- Listados de trabajadores e invitados por empresa

4. Patrones de Diseño Implementados

4.1 Patrones Obligatorios

- **Singleton**: Gestión de conexión a base de datos
- **DAO (Data Access Object)**: Abstracción de acceso a datos

4.2 Patrones Adicionales

- **Observer**: Sincronización en tiempo real
- **Factory**: Creación de usuarios y roles
- **Command**: Manejo de acciones de registro

5. Principios SOLID Aplicados

5.1 Responsabilidad Única

Cada clase tiene una única responsabilidad bien definida, mejorando la modularidad y mantenibilidad del código.

5.2 Principio Abierto/Cerrado

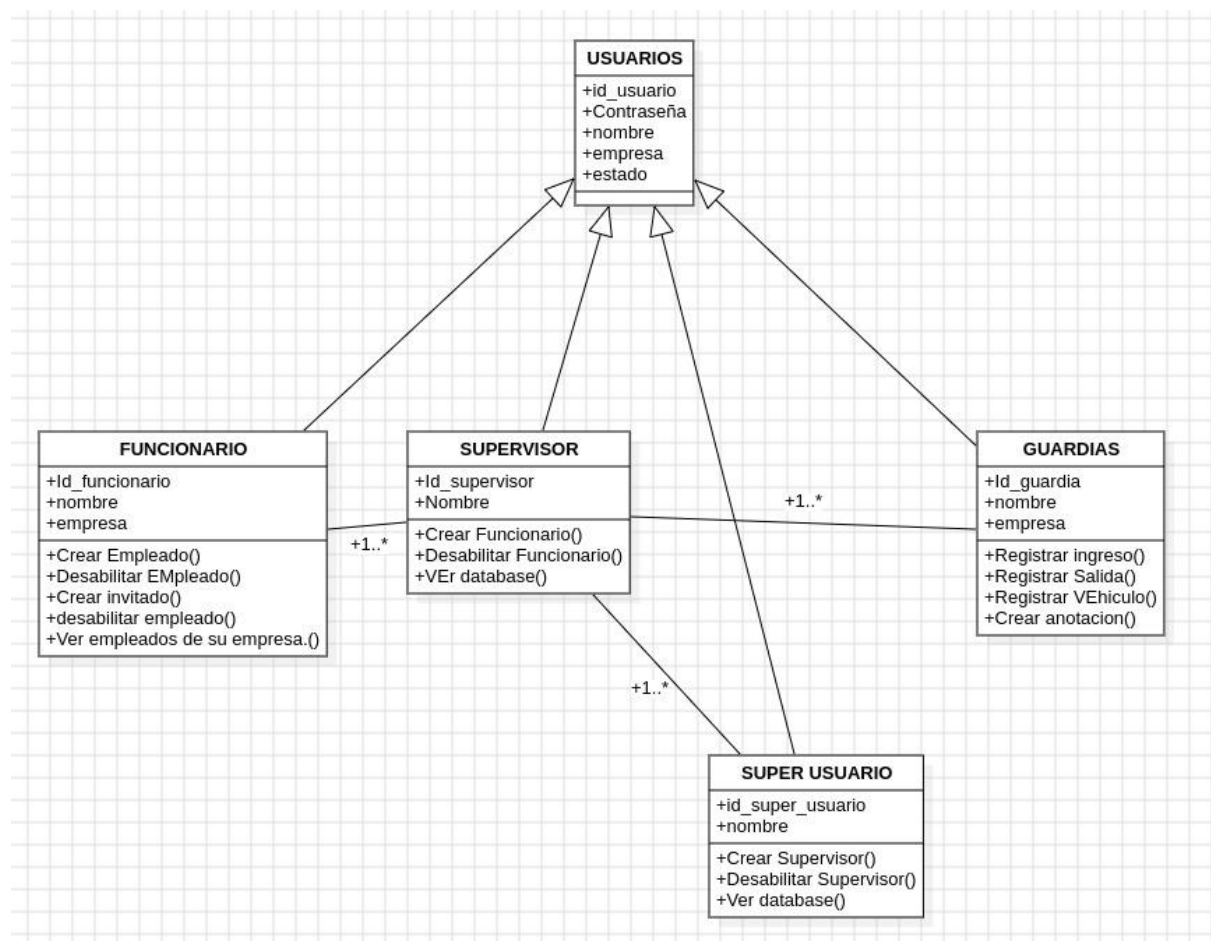
El sistema permite extensiones sin modificar su código fuente principal.

5.3 Inversión de Dependencia

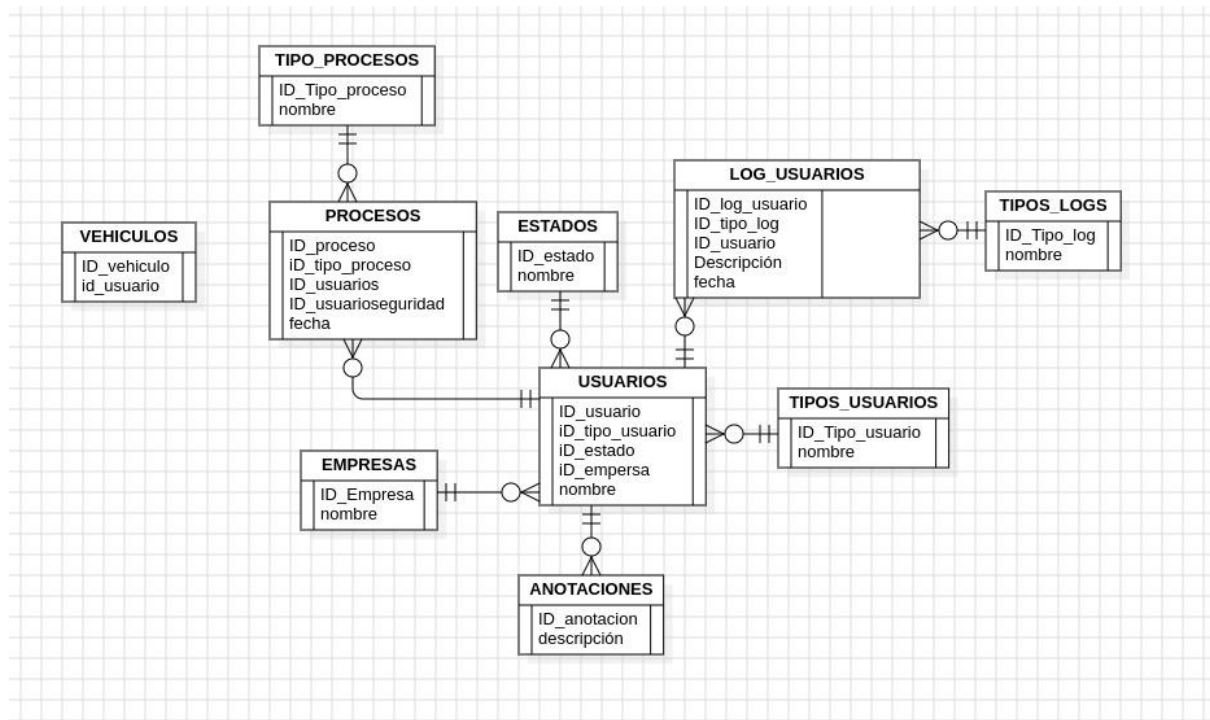
Uso de interfaces para desacoplar la lógica del sistema.

6. Diagramas

6.1 Diagrama de Clases

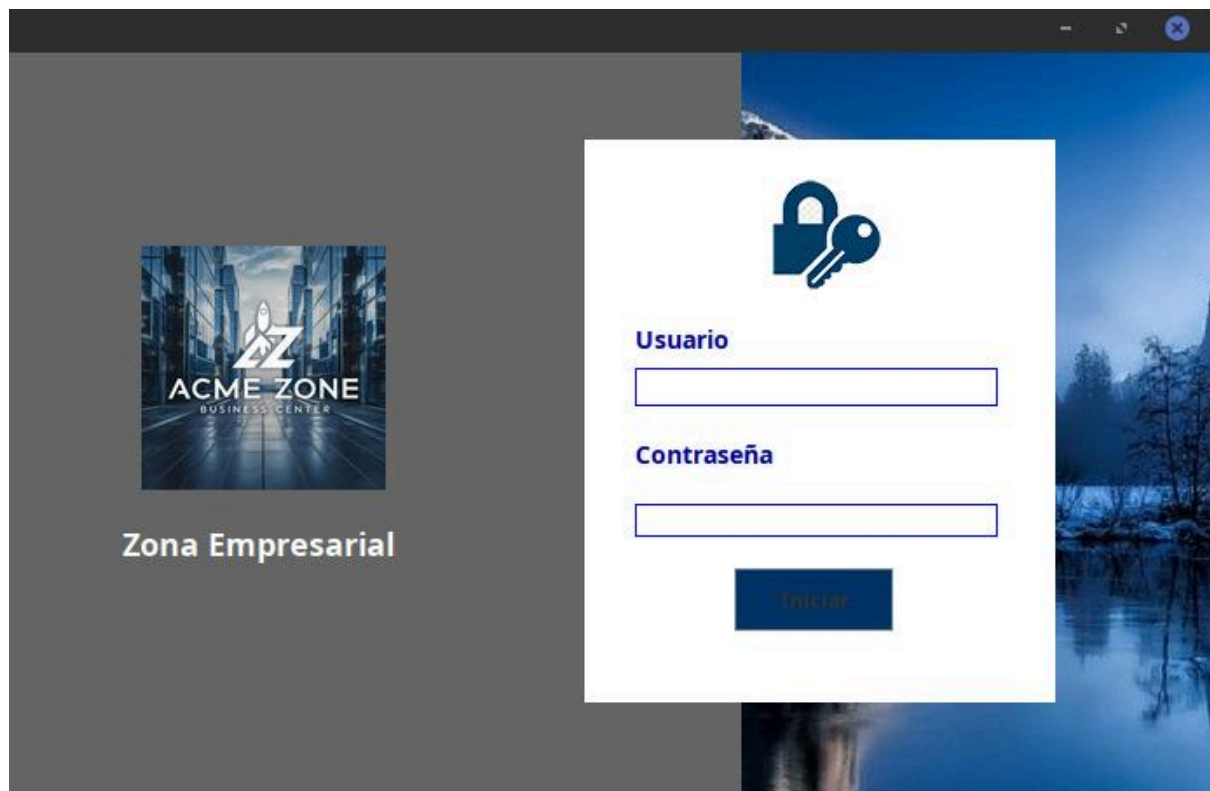


6.2 Diagrama Entidad-Relación (ER)




7. Capturas de Pantalla

7.1 Pantalla de Login



7.2 Pantalla Principal del Guarda



Zone Guardias

Registrar Entrada

Registrar Salida

Registro Vehiculo

Anotaciones

Cerrar Sesion


Documento

Registrar

Cancelar

Nombre	Documento	Tipo de proceso
--------	-----------	-----------------

7.3 Pantalla de Superusuario



Zone Super Users

Registrar Supervi...

Desabilitar

Cerrar Sesion

Nombre completo


Documento

Registrar

Cancelar

Nombre	Documento	Tipo de usuario
--------	-----------	-----------------

7.4 Pantalla de Supervisor



Zone Supervisor

Registrar Funcionario

Desabilitar Funcionario

Registrar Guardia

Desabilitar Guardia

Cerrar Sesión


Nombre completoDocumentoEmpresa

Registrar

Cancelar

Nombre	Documento	Empresa
--------	-----------	---------

7.5 Pantalla de Funcionario



Zone Funcionario

Registrar Empleados

Desabilitar Empleados

Registrar Invitados

Desabilitar Invitados

Nombre compl...DocumentoEmpresa

Registrar

Cancelar

Nombre	Documento	Empresa
--------	-----------	---------

8. Requisitos del Sistema

8.1 Software Necesario

- Java JDK 17+
- MySQL Server
- IDE: IntelliJ IDEA, Eclipse o NetBeans

8.2 Instalación

Pasos de Configuración:

1. Clonar repositorio
2. Configurar base de datos
3. Establecer conexión en archivo de configuración
4. Compilar y ejecutar desde IDE

9. Creadores del Proyecto

9.1 Equipo de Desarrollo

- **Arley Mantilla:**Base de datos
- **Nicolas Ruiz:**Interfaces gráficas
- **Carlos Cotes:**Java

10. Tareas Futuras

- Integración con dispositivos biométricos
- Exportación de reportes a PDF/Excel
- Migración a JavaFX

11. Contribución

Proceso de Contribución

1. Realizar fork del repositorio
2. Crear rama de características
3. Realizar cambios
4. Enviar Pull Request