INFORME TÉCNICO DEL SISTEMA DE CONTROL DE ACCESO A EDIFICIOS

# 1. Datos Generales del Proyecto

- Nombre del Proyecto: Sistema de Control de Acceso a Edificios  
- Autor: Sebastian Garcia Jaramillo – Jose David Polo  
- Asignatura: Herramientas de Programación II  
- Docente: Msc. Alexandra Guerrero Bocanegra  
- Fecha: Mayo - Junio 2025

# 2. Análisis Funcional del Sistema

## 2.1 Objetivo General

Desarrollar una aplicación de escritorio en C# que permita registrar, monitorear y controlar los accesos de personas (empleados y visitantes) a distintas zonas de un edificio, garantizando seguridad mediante alertas y control de roles.

## 2.2 Requisitos Funcionales

1. Login y Autenticación  
 - Acceso al sistema mediante usuario y contraseña.  
 - Control por roles: Administrador, Seguridad, Empleado.  
  
2. Gestión de Empleados  
 - Registrar, consultar, modificar y eliminar empleados.  
 - Asignar rol y zona de acceso.  
  
3. Gestión de Visitantes  
 - Registrar, consultar, modificar y eliminar visitantes.  
 - Registrar propósito de visita y contacto.  
  
4. Registro de Accesos  
 - Control de entrada y salida por fecha y hora.  
 - Control de acceso a zonas por rol.  
  
5. Alertas de Seguridad  
 - Generación automática de alertas en casos de acceso no autorizado o fuera de horario.  
  
6. Reportes  
 - Generación de reportes filtrados por persona, zona, fecha y horario.  
  
7. Monitoreo en Tiempo Real  
 - Visualización de accesos activos en tiempo real (opcional o extendido).

# 3. Justificación Técnica de las Decisiones Tomadas

## 3.1 Lenguaje y Entorno de Desarrollo

C# Windows Forms fue elegido por su integración nativa con Windows, facilidad de uso y soporte de interfaz gráfica rápida, ideal para aplicaciones de escritorio empresariales.

## 3.2 Base de Datos

SQL Server se utilizó por su robustez, facilidad de integración con C# y por contar con herramientas visuales para diseño, gestión y ejecución de scripts.

## 3.3 Arquitectura y Diseño

- Diseño en capas: Separación de interfaz, lógica de negocio y acceso a datos.  
- Patrones de diseño utilizados:  
 - Repository Pattern: Para desacoplar la lógica de acceso a datos.  
 - Singleton: Para mantener una única conexión a la base de datos activa.  
- Principios SOLID aplicados:  
 - Single Responsibility Principle: Cada clase tiene una única responsabilidad (ej. EmpleadoRepository, VisitanteRepository).  
 - Open/Closed Principle: Las clases están abiertas a extensión pero cerradas a modificación (a través de interfaces).

## 3.4 Seguridad

Se definieron roles para restringir accesos a funcionalidades:  
- Administrador: Control total.  
- Seguridad: Solo registro y monitoreo.  
- Empleado: Acceso básico al sistema.

## 3.5 Procedimientos Almacenados

Se implementaron stored procedures en SQL Server para encapsular operaciones críticas:  
- Registro de accesos (SP\_RegistrarAcceso)  
- Generación de alertas (SP\_GenerarAlerta)  
- Reportes (SP\_ReporteAccesosInusuales, SP\_ObtenerHistorialAccesos)  
  
Esto reduce errores y mejora el rendimiento al evitar múltiples consultas desde la aplicación.

# 4. Conclusiones

- El sistema cumple con los requerimientos funcionales definidos.  
- Se implementó una arquitectura escalable y mantenible gracias al uso de patrones de diseño y principios SOLID.  
- Está preparado para ser extendido a futuro con módulos adicionales (cámaras, control biométrico, sensores).  
- La solución es adecuada para edificios corporativos, institucionales o gubernamentales.

# 5. Anexos

- Diagrama Entidad-Relación  
- Capturas de pantalla  
- Código fuente en GitHub  
- Script SQL de base de datos