

PROYECTO FINAL - SISTEMA DE ALERTAS URGENTES

Nombre del Estudiante:

Litzi Yasmira Pineda Cermeño

Carne:

0905-24-10033

Curso:

Programación 1

Nombre del Proyecto:

Sistema de Alertas Urgentes



1. Introducción

El presente proyecto consiste en el análisis y registro de alertas urgentes, utilizando una simulación de inteligencia artificial (IA). Fue desarrollado en el lenguaje C#, con el uso de Windows Forms y la librería MaterialSkin para mejorar la interfaz gráfica. Además, el sistema incorpora funcionalidades como el envío de correos electrónicos automáticos y el registro de alertas en una base de datos SQL Server.

2. Justificación

En una era donde las alertas ciudadanas son fundamentales para la seguridad pública, contar con un sistema que pueda analizar, registrar y notificar alertas importantes es crucial. Este proyecto busca simular el comportamiento de un sistema real que, mediante el uso de IA, identifique alertas relevantes, las comunique de forma oportuna, y almacene su historial para consultas futuras. Asimismo, se buscó implementar una interfaz moderna y amigable para mejorar la experiencia del usuario.

3. Objetivos

- **Objetivo General:** Desarrollar una aplicación de escritorio que analice alertas escritas por el usuario, utilice una simulación de inteligencia artificial para clasificarlas, envíe correos si son relevantes y las almacene en una base de datos.
- **Objetivos Específicos**

- Implementar una interfaz de usuario moderna utilizando MaterialSkin.
- Simular el análisis de alertas mediante una clase que represente el uso de IA.
- Automatizar el envío de correos electrónicos en casos de alertas urgentes.
- Registrar y recuperar información desde una base de datos SQL Server.
- Mostrar las 5 alertas más importantes registradas.

4. Desarrollo Técnico

Lenguaje y Entorno

- Lenguaje: C#
- Entorno: Visual Studio
- Interfaz: Windows Forms + MaterialSkin
- Base de datos: SQL Server
- Librerías adicionales: MaterialSkin.dll, System.Net.Mail, System.Data.SqlClient

Componentes Principales del Sistema

Formulario Principal (Form1.cs)

Contiene todos los elementos visuales: cajas de texto, botones, etiquetas, y la lógica de eventos.

- **GroqClient.cs** Simula una clase de IA que recibe el texto ingresado y genera una clasificación de la alerta.
- **EmailHelper.cs** Clase encargada del envío de correos electrónicos cuando se detectan alertas graves.
- **DatabaseHelper.cs** Clase que maneja la conexión a SQL Server, inserta nuevos registros de alertas y obtiene el Top 5.

Interfaz de Usuario

La interfaz utiliza Material Design, que proporciona una apariencia moderna. Incluye:

- Caja de texto para ingresar la alerta.
- Botón "Enviar Alerta" que procesa el análisis.
- Cuadro de resultados.
- Top 5 de alertas importantes.
- Íconos e imágenes decorativas para mejor presentación.
- Flujo de Ejecución
- El usuario escribe una alerta.
- Al hacer clic en "Enviar Alerta":
- Se valida que el texto no esté vacío.
- Se analiza con GroqClient.
- Se muestra el resultado.
- Si es grave, se envía un correo automáticamente.
- Se guarda en la base de datos.
- Se actualiza el listado de las 5 alertas más importantes.

5. Resultados Obtenidos

El sistema permite analizar eficazmente una alerta escrita. Por ejemplo:

- Alerta 1: "Han secuestrado a un niño en zona 1" → Clasificada como grave → se envía correo.
- Alerta 2: "Lluvias fuertes en zona 12" → Clasificada como moderada → se guarda.
- Alerta 3: "Se cayó una rama en la calle" → Clasificada como leve → no se envía correo.
- Además, se registran correctamente los datos en la base de datos, y se visualiza el Top 5 de alertas con mayor prioridad.

6. Conclusiones

- El uso de C# y Windows Forms permite construir aplicaciones robustas y funcionales con conexión a bases de datos.
- MaterialSkin mejora significativamente la experiencia del usuario con un diseño moderno.
- Simular IA dentro de una aplicación es una buena forma de preparar una arquitectura futura que pueda conectarse a verdaderas APIs de lenguaje natural.
- Automatizar el envío de correos es útil para sistemas de emergencia o monitoreo.

- El trabajo modular (con clases separadas) hace que el sistema sea fácil de mantener y expandir.

7. Recomendaciones

- Para versiones futuras, se recomienda integrar una API real de IA, como OpenAI o Azure Cognitive Services.
- Agregar filtros de búsqueda y fechas en el historial de alertas para mejorar la consulta.
- Implementar control de usuarios con roles (administrador, operador, etc.).
- Agregar confirmaciones antes de enviar correos.
- Optimizar el análisis con aprendizaje automático real para identificar patrones de alertas.

8. Capturas de Pantalla

- Pantalla principal con MaterialSkin.
- Envío de alerta.
- Análisis y resultado mostrado.
- Envío de correo confirmado.
- Consulta del Top 5 desde la base de datos.