

**1 Introducción: descripción general del proyecto y su propósito, justificación y alcance.**

**2 Objetivos: los objetivos del proyecto y lo que se espera lograr con la aplicación web.**

**3 Requisitos funcionales: una lista de las funciones que la aplicación web debe realizar, como la autenticación de usuarios, la gestión de contenido y la generación de informes.**

## **1. Introducción**

En el contexto actual de alta conectividad, garantizar comunicaciones seguras y rápidas se ha convertido en un factor clave. Este proyecto plantea la creación de un Sistema de Alertas vía Telegram con Autenticación mediante Reconocimiento Facial, una solución que integra la inmediatez de la mensajería de Telegram con la seguridad que ofrece la biometría facial.

El objetivo principal es ofrecer una herramienta confiable para el envío de alertas relevantes únicamente a usuarios previamente validados. La propuesta surge como respuesta a las limitaciones de los sistemas de notificación convencionales, que suelen carecer de mecanismos sólidos de autenticación y pueden ser susceptibles a accesos no autorizados. La incorporación del reconocimiento facial refuerza la seguridad al garantizar que solo personas autorizadas accedan a información sensible o ejecuten acciones críticas.

El desarrollo contempla una plataforma web que permitirá gestionar usuarios, almacenar imágenes faciales, procesar la autenticación y coordinar el envío de alertas a través de la API de Telegram. Asimismo, se incluirá una interfaz administrativa para la configuración del sistema. El diseño se enfocará en asegurar la escalabilidad, permitiendo adaptar la solución a diferentes contextos, desde el ámbito doméstico hasta aplicaciones corporativas.

## **2. Objetivos del Proyecto**

**Los principales objetivos de esta iniciativa son:**

Diseñar un sistema de autenticación confiable: Implementar reconocimiento facial con alta precisión y baja latencia para validar el acceso de los usuarios de forma segura.

Automatizar el envío de alertas: Desarrollar un mecanismo eficiente para emitir notificaciones a usuarios o grupos de Telegram, ya sea de forma manual o en respuesta a eventos configurados.

Proteger la información sensible: Aplicar estándares avanzados de seguridad en el manejo de datos e imágenes, incluyendo cifrado y control estricto de accesos.

Ofrecer una interfaz de administración clara e intuitiva: Permitir una experiencia de uso sencilla para la gestión de usuarios, alertas y parámetros del sistema.

Integrar eficazmente con Telegram: Asegurar una comunicación fluida mediante el uso de la API de Telegram para el envío de mensajes y administración del bot.

Preparar una arquitectura escalable: Diseñar el sistema para soportar futuros crecimientos en usuarios, reglas de alerta y funcionalidades adicionales.

### **3. Requisitos Funcionales**

El sistema deberá contar con las siguientes capacidades:

#### **Gestión de Usuarios**

Alta de usuarios: Registrar nuevos perfiles con datos personales e imágenes faciales para su posterior identificación.

Edición de perfiles: Permitir la actualización de información o el reemplazo de imágenes faciales.

Eliminación de usuarios: Posibilidad de eliminar registros completos del sistema.

Control de roles y permisos: Definir distintos niveles de acceso (administrador, operador, usuario) según funciones específicas.

#### **Autenticación y Control de Acceso**

Reconocimiento facial: Validar la identidad mediante comparación de imagen facial con la base de datos.

Acceso alternativo por credenciales: Activar ingreso por usuario y contraseña para administradores como mecanismo complementario.

Recuperación de acceso: Proveer una opción para restablecer contraseñas cuando aplique.

#### **Manejo de Alertas**

Creación de alertas: Configurar tipos de alerta, prioridad, contenido y destinatarios.

Envío manual: Emitir notificaciones en tiempo real a contactos específicos.

Automatización de alertas: Definir eventos disparadores para envío automático (como detección de movimiento o lectura de sensores).

Historial de alertas: Registrar cada alerta enviada, incluyendo fecha, remitente y estado.

### **Integración con Telegram**

Conexión segura con API: Establecer comunicación estable y protegida con la plataforma de Telegram.

Envío de contenidos: Permitir el envío de mensajes de texto, multimedia y documentos.

Recepción de comandos: Procesar respuestas de los usuarios al bot (por ejemplo, confirmaciones o solicitudes).

### **Gestión de Contenido**

Plantillas de mensaje: Crear estructuras predefinidas para agilizar el proceso de notificación.

Personalización dinámica: Insertar datos variables en los mensajes, como nombres o detalles específicos del evento.

### **Reportes y Auditoría**

Reportes de actividad: Generar informes de uso del sistema y comportamiento del usuario.

Bitácora de acciones: Registrar todas las operaciones realizadas para auditoría y seguimiento.

Estadísticas: Mostrar métricas generales como número de alertas enviadas, usuarios activos, entre otros.

### **Configuración General**

Parámetros del sistema: Ajustar opciones generales como umbrales de reconocimiento, horarios y zonas.

Gestión de notificaciones internas: Determinar qué avisos administrativos deben generarse ante cambios o incidencias del sistema.