4. **<u>Requerimientos no funcionales:</u> una lista de los requerimientos que no están relacionados con las funciones de la aplicación web debe realizar, como la seguridad, el rendimiento y la escalabilidad.**

   Seguridad:

   - El sistema de autenticación facial del sistema debe asegurar que solo usuarios autorizados puedan acceder
   - El sistema debe proteger los datos biométricos y datos de usuario mediante un cifrado robusto.
   - La comunicación entre la aplicación y Telegram deben estar protegidas mediante protocolos seguros.
   - Debe cumplir con las normativas legales vigentes en Venezuela respecto a la privacidad.

   Rendimiento:

   - El sistema debe asegurar que el proceso de reconocimiento facial no exceda los 3 segundos desde la captura de la imagen hasta la autenticación.
   - Detectar el 95% de las fallas críticas en un plazo no mayor a 5 minutos de sucedidas.
   - Enviar notificaciones de alerta a telegram en un tiempo que un supere 1 minuto después de la detección de una falla crítica.
   - El sistema debe ser capaz de responder rápidamente a las interacciones del chatbot IA sin demoras perceptibles.

   Escalabilidad:

   - El sistema debe ser capaz de manejar un número creciente de usuarios, alertas y fuentes de datos sin afectar su rendimiento.

   Mantenibilidad:

   - Casos de uso: descripción detallada de cómo los usuarios interactuarán con la aplicación web y cómo se espera que funcione.
   - El sistema debe facilitar la depuración de errores mediante logs detallados, accesibles solo para administradores autorizados.

- El código debe ser modular y seguir principios de diseño que faciliten su mantenimiento y actualización.

Disponibilidad:

- El sistema debe estar disponible el menos el 99% del tiempo mensualmente

Fiabilidad y Resiliencia:

- El sistema debe estar disponible las 24 horas del día, a menos que se planee lo contrario.
- El sistema debe poder recuperarse de fallas menores de manera automática
- Debe tener un mecanismo de respaldo y restauración para proteger la integridad en casos de fallos críticos.

Monitoreo:

- Todas las actividades (Accesos, alertas, interacciones con el chatbot) van a ser registradas con fecha y hora.
- Se debe tener un sistema de monitoreo que evalúe el rendimiento y la disponibilidad de la aplicación.

Usabilidad:

- La interfaz debe ser amigable para todo tipo de usuario
- El chatbot debe adaptar su lenguaje y respuestas al tipo de usuario
- La aplicación debe cumplir con las pautas de accesibilidad WCAG 2.1 para garantizar que sea utilizable por personas con discapacidades.

5. **<u>Riesgos y su mitigación:</u> cualquier restricción que deba tenerse en cuenta durante el desarrollo de la aplicación web, como el presupuesto, el tiempo y los recursos disponibles.**

Para el desarrollo de un proyecto, es fundamental identificar los posibles riesgos que puedan perjudicar el resultado, así como establecer diversas estrategias para poder mitigar y gestionar estos inconvenientes de manera proactiva. A continuación, para el desarrollo e implementación del “Sistema de Alerta de Telegram con Acceso Facial” se han identificado los principales riesgos técnicos, presupuestarios, técnicos, de tiempo y legales junto con su respectiva mitigación.

- **Riesgos Técnicos:**
    - Fallas en el reconocimiento facial: Se refiere a la baja precisión, largos tiempo de espera, errores de autenticación o sesgos en la detección de rostros. Para su mitigación es recomendable el uso de modelos probados y entrenados con bases de datos y la realización de pruebas exhaustivas antes de su despliegue.

    - Interrupciones en la API de Telegram: Esto puede ser causado por la dependencia de un servicio externo para el sistema de notificaciones. Para su mitigación es recomendable implementar reintentos automáticos y mantener medios de notificación de respaldo, por ejemplo, correo electrónico.

    - Pérdida de datos: Se refiere a posibles fallos en el almacenamiento, corrupción de bases de datos o ciberataques. Como medidas de mitigación es recomendable hacer respaldos periódicos, el uso de cifrado robusto y medidas de seguridad avanzada en las bases de datos y servidores.

- **Riesgos de presupuestarios:**
    - Limitaciones en el presupuesto: Se refiere a la capacidad para adquirir equipos, licencia y hardware biométrico. Para su mitigación se propone el uso de herramientas de open source, hardware ya existente y priorizar las funcionalidades esenciales del sistema.

- **Riegos sobre tiempo:**
    - Retrasos en el cronograma: Refiere a tiempos de desarrollo mayores al estimado por complejidad técnica o cambios de alcance. Para su mitigación se propone el uso de metodologías agiles como Kanban, una planificación realista de sprints y entregas incrementables.
- **Riesgos Legales:**
    - Uso inadecuado de datos: Posibles problemas legales si no se maneja correctamente la información personal. Su mitigación correspondiente es, la implementación del consentimiento informado; uso de políticas clara de privacidad, cifrado de las plantillas faciales y el cumplimiento del normativo nacional.
    - Falta de cumplimiento con leyes de protección de datos: Se refiere a posibles sanciones legales por no adherirse a las leyes vigentes. Para mitigar esto se propone consultar legalmente durante el diseño y revisión periódica del cumplimiento con la legislación venezolana de datos.

6. Casos de uso: descripción detallada de cómo los usuarios interactuarán con la aplicación web y cómo se espera que funcione.

En esta sección se describen los principales escenarios de interacción entre los usuarios y el sistema, con el fin de ilustrar cómo se utilizará la aplicación web.

- **Caso de Uso 1. Registro de Usuario.**
    - **Actor:** Usuario final/ ciudadano
    - **Descripción:** El usuario se registra, configura sus preferencias y si es el caso, registrar si información facial.
    - **Precondiciones:** El usuario accede a la plataforma web desde un equipo con cámara
    - **Flujo principal:**
        - El usuario completa sus datos
        - El sistema solicita captura facial

        - Se almacenan datos biométricos
    - **Postcondición:** El usuario queda registrado y se puede autenticar mediante su rostro.
- **Caso de Uso 2. Detección y notificación automática de fallas.**
    - **Actor Principal:** Sistema automatico
    - **Descripción:** Detecta fallas y enviar las alertas en tiempo real
    - **Precondiciones:** Reglas de monitoreo configuradas y fuentes de datos activas.
    - **Flujo principal:**
        - El sistema detecta una anomalía
        - El sistema genera una alerta
        - Se envía notificaciones vía Telegram a los usuarios responsables.
    - **Postcondición:** Los usuarios son informados inmediatamente sobre la falla detectada.
- **Caso de Uso 3. Autenticación mediante reconocimiento facial.**
    - **Actor Principal:** Usuario, Sistema, Sistema Externo
    - **Descripción:** el acceso al sistema de forma segura mediante un rostro.
    - **Flujo Principal:**
        - El sistema solicita captura facial
        - La cámara captura una imagen del rostro
        - El sistema procesa y compara la imagen con la base de datos de rostros autorizados
        - Se concede o deniega el acceso.
- **Caso de Uso 4. Diagnostico asistido con el ChatBot.**
    - **Actor principal:** Usuario operador, Chatbot
    - **Descripción:** Diagnostico y resolución de las alertas con ayudan el chatbot
    - **Precondiciones:** Hay una alerta activa y hay un usuario en línea
    - **Flujo Principal:**
        - El chatbot da la bienvenida
        - Solicita detalles e información
        - Busca y compara la información con la base de datos.

        - Sugiere diagnóstico y solución con base en la alerta y base de conocimiento
    - **Postcondición:** El usuario obtiene guía para solucionar la falla.
- **Caso de Uso 5. Cierre de añerta tras solución manual**
    - **Actor Principal:** Usuario Operador, Administrador
    - **Descripción:** Marca como resuelta la alerta que ya fue atendida manualmente.
    - **Precondiciones:** El usuario ha ingresado al sistema y cuenta con los permisos necesarios.
    - **Flujo Principal:**
        - El usuario accede a la interfaz de alertas
        - Localiza la alerta activa
        - Marca la alerta como resulta.
    - **Postcondición:** La alerta queda archivada y registrada en el historial del sistema.
- **Caso de Uso 6. Visualización de Reportes e historial de alertas**
    - **Actor principal:** Administrador del sistema
    - **Descripción:** Consulta las estadísticas e informes históricos sobre alertas autenticaciones y uso del chat.
    - **Flujo Principal:**
        - Accede al módulo de reportes
        - Selecciona el rango de fechas o tipo de datos
        - El sistema genera la información solicitada
        - El administrador puede descargar el informe.
    - **Postcondición:** El usuario es capaz de visualizar la información previamente solicitada.