

Requisitos No Funcionales

Estos requisitos no funcionales son esenciales para garantizar que el Sistema de Alerta de Telegram con Acceso por Reconocimiento Facial cumpla con las expectativas de los usuarios y los estándares de calidad necesarios para su operación efectiva y segura.

1. Rendimiento

1.1 Tiempo de Respuesta: La aplicación debe procesar el reconocimiento facial y enviar alertas a Telegram en un tiempo máximo de 2 segundos desde la detección del rostro.

1.2 Escalabilidad: La aplicación debe ser capaz de manejar al menos 1000 usuarios concurrentes sin degradar el rendimiento.

2. Seguridad

2.1 Autenticación y Autorización: La aplicación debe implementar un sistema de autenticación basado en reconocimiento facial, asegurando que solo los usuarios autorizados puedan acceder.

2.2 Protección de Datos: Todos los datos biométricos y de usuario deben ser almacenados de forma encriptada utilizando estándares de encriptación robustos (por ejemplo, AES-256).

2.3 Seguridad en la Comunicación: Las comunicaciones entre la aplicación y Telegram, así como las interacciones con la base de datos, deben estar protegidas mediante HTTPS y protocolos seguros.

3. Usabilidad

3.1 Interfaz de Usuario: La interfaz debe ser intuitiva y fácil de usar, permitiendo a los usuarios interactuar con la aplicación sin necesidad de capacitación extensa.

3.2 Accesibilidad: La aplicación debe cumplir con las pautas de accesibilidad WCAG 2.1 para garantizar que sea utilizable por personas con discapacidades.

4. Mantenibilidad

4.1 Documentación: La aplicación debe contar con documentación técnica completa que incluya guías de instalación, configuración y mantenimiento.

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

5. Disponibilidad

5.1 Tiempo de Actividad: La aplicación debe garantizar un tiempo de actividad del 99.9%, lo que significa que no debe estar inactiva más de 43.2 minutos al mes.

5.2 Recuperación ante Desastres: Debe existir un plan de recuperación ante desastres que permita restaurar el servicio en un tiempo máximo de 1 hora tras una falla crítica.

6. Compatibilidad

6.1 Dispositivos: La aplicación debe ser compatible con dispositivos móviles (iOS y Android) y navegadores web modernos (Chrome, Firefox, Safari).

6.2 Integración con Telegram: La aplicación debe integrarse sin problemas con la API de Telegram, asegurando que las alertas se envíen de manera efectiva y eficiente.

7. Portabilidad

7.1 Plataforma: La aplicación debe ser capaz de ejecutarse en diferentes sistemas operativos (Windows, Linux, macOS) con el mismo nivel de funcionalidad.

8. Monitoreo y Auditoría

8.1 Registro de Actividades: La aplicación debe registrar todas las actividades relevantes, incluidas las alertas enviadas y los intentos de acceso, para su posterior auditoría.

8.2 Monitoreo de Rendimiento: Debe implementarse un sistema de monitoreo que permita evaluar el rendimiento y la disponibilidad de la aplicación en tiempo real.

Estructura apropiada para el Documento de Requisitos de la aplicación

Para definir una estructura apropiada para el Documento de Requisitos de una aplicación, es importante incluir secciones que aborden tanto los requisitos funcionales como los no funcionales, así como otros aspectos relevantes del proyecto# Estructura del Documento de Requisitos

Esta estructura proporciona un marco claro y organizado para documentar los requisitos de la aplicación, asegurando que todos los aspectos relevantes sean considerados y comunicados efectivamente.

1. Introducción

1.1 Propósito: Describir el objetivo del documento y su importancia.

1.2 Alcance: Definir el alcance de la aplicación y las limitaciones.

1.3 Definiciones, Acrónimos y Abreviaturas: Explicar términos técnicos y acrónimos utilizados en el documento.

1.4 Referencias: Listar documentos y fuentes consultadas.

2. Descripción General

2.1 Perspectiva del Producto: Describir cómo se integra la aplicación en un sistema mayor.

2.2 Funciones del Producto: Resumen de las funciones principales que ofrecerá la aplicación

2.3 Usuarios Previstas: Identificar los diferentes tipos de usuarios que interactuarán con la aplicación

2.4 Restricciones: Detallar limitaciones técnicas y operativas.

3. Requisitos Funcionales

3.1 Requisitos de Usuario: Listar los requisitos desde la perspectiva del usuario.

3.2 Requisitos del Sistema: Detallar los requisitos técnicos y operativos que debe cumplir el sistema.

3.3 Casos de Uso: Incluir diagramas y descripciones de casos de uso relevantes.

4. Requisitos No Funcionales

4.1 Rendimiento: Especificar requisitos de velocidad, capacidad y eficiencia.

4.2 Seguridad: Detallar requisitos de seguridad y protección de datos.

4.3 Usabilidad: Describir los requisitos relacionados con la experiencia del usuario.

4.4 Mantenibilidad: Especificar requisitos para la facilidad de mantenimiento y actualización.

5. Requisitos de Interfaz

5.1 Interfaz de Usuario: Describir cómo será la interacción del usuario con la aplicación.

5.2 Interfaz del Sistema: Detallar la comunicación entre la aplicación y otros sistemas.

6. Requisitos de Implementación

6.1 Entorno de Desarrollo: Especificar el entorno y las herramientas que se utilizarán.

6.2 Lenguajes de Programación: Listar los lenguajes y tecnologías a utilizar.

7. Anexos

7.1 Diagramas: Incluir diagramas relevantes (UML, flujos de trabajo, etc.).

7.2 Documentación Adicional: Listar cualquier documento adicional que complemente el documento de requisitos.

8. Aprobaciones

8.1 Firmas: Espacio para las firmas de aprobación de las partes interesadas.

Estrategias y técnicas que el grupo o estudiante deberá emplear para:

- Validar los requisitos.
- Hacerle seguimiento a los requisitos.
- Controlar los cambios.

La implementación de estas estrategias y técnicas permitirá un manejo efectivo de los requisitos en el desarrollo del sistema de alerta de Telegram con reconocimiento facial, asegurando que se cumplan las expectativas de los usuarios y se minimicen los riesgos asociados con cambios no controlados.

	1. Estrategias	2. Técnicas
1. Validación de Requisitos	- Reuniones de Requisitos: Organizar sesiones con los interesados (stakeholders) para discutir y aclarar los requisitos del sistema.	- Revisión de Requisitos: Realizar revisiones formales de los documentos de requisitos con el equipo y los interesados.
	- Prototipos: Crear prototipos del sistema que permitan a los usuarios visualizar y validar los requisitos antes de la implementación.	- Técnicas de Modelado: Usar diagramas de casos de uso o diagramas de flujo para representar visualmente los requisitos y facilitar su validación.
	- Historias de Usuario: Utilizar historias de usuario para capturar requisitos desde la perspectiva del usuario final, facilitando la comprensión y validación.	- Pruebas de Aceptación: Definir criterios de aceptación claros que se utilizarán para validar que los requisitos se han cumplido en el producto final.

2. Seguimiento de Requisitos	Gestión de Requisitos: Implementar herramientas de gestión de requisitos para rastrear el estado de cada requisito a lo largo del ciclo de vida del desarrollo.	Matriz de Trazabilidad: Crear una matriz de trazabilidad que vincule los requisitos a sus respectivas pruebas, asegurando que todos los requisitos sean cubiertos y verificados.
	Documentación Continua: Mantener una documentación actualizada que refleje el estado de los requisitos y cualquier cambio realizado.	Revisiones Periódicas: Establecer reuniones periódicas para revisar el progreso de los requisitos y ajustar el enfoque según sea necesario.
3. Control de Cambios	Proceso de Control de Cambios: Establecer un proceso formal para solicitar, evaluar y aprobar cambios en los requisitos.	Registro de Cambios: Mantener un registro detallado de todos los cambios realizados en los requisitos, incluyendo la razón del cambio y su impacto en el proyecto.
	Comunicación Clara: Asegurar que todos los interesados estén informados sobre los cambios en los requisitos y sus implicaciones	Evaluación de Impacto: Realizar análisis de impacto para cada cambio propuesto, evaluando cómo afectará el alcance, el tiempo y el costo del proyecto.