

Desarollo de Proyecto ADA

Escrito Por Nexus

Especificación de Requisitos según el estándar de IEEE 830

Versión: 1.1

Fecha:19/06/24

**Índice**

1. **Introducción ----------------------------------------------------------------------------------------------- 1**

- 1.1 Propósito

- 1.2 Alcance

- 1.3 Definiciones, Acrónimos y Abreviaturas

- 1.4 Visión General

2. **Descripción General -------------------------------------------------------------------------------------- 2**

- 2.1 Perspectiva del Producto

- 2.2 Funcionalidades del Producto

- 2.3 Características del Usuario

- 2.4 Restricciones

- 2.5 Suposiciones y Dependencias

3. **Requerimientos Específicos ----------------------------------------------------------------------------- 3**

- 3.1 Requerimientos Funcionales

- 3.2 Requerimientos No Funcionales

4. **Referencias -------------------------------------------------------------------------------------------------- 5**

5. **Enlaces ------------------------------------------------------------------------------------------------------- 6**

**Introducción**

1.1 **Propósito**

El propósito de este documento es definir los requerimientos funcionales y no funcionales para la plataforma Nexa, en conformidad con el estándar IEEE 830.

1.2 **Alcance**

Nexa es una plataforma de comunidades en línea que permite a los usuarios registrarse, crear y gestionar comunidades, publicar contenido, interactuar con publicaciones, enviar mensajes privados y recibir notificaciones.

1.3 **Definiciones, Acrónimos y Abreviaturas**

- Usuario: Individuo registrado en la plataforma.

- Comunidad: Grupo creado por un usuario para compartir intereses comunes.

- Publicación: Contenido creado por un usuario dentro de una comunidad.

- Notificación: Alerta enviada a un usuario sobre eventos relevantes.

1.4 **Visión General**

Este documento detalla los requerimientos necesarios para el desarrollo de Nexa, estructurados en requerimientos funcionales y no funcionales.

1

**Descripción General**

2.1 **Perspectiva del Producto**

Nexa es una aplicación web que proporciona una plataforma para la creación y gestión de comunidades en línea. Permite a los usuarios interactuar mediante publicaciones, comentarios y mensajes privados, y recibir notificaciones sobre actividades relevantes.

2.2 **Funcionalidades del Producto**

- Registro y autenticación de usuarios.

- Gestión de perfil.

- Creación y administración de comunidades.

- Creación y gestión de contenido.

- Interacción con publicaciones.

- Mensajería privada.

- Notificaciones.

- Búsqueda y descubrimiento.

- Moderación y seguridad.

- Configuración y personalización.

2.3 **Características del Usuario**

- Usuarios finales: Individuos que se registran y utilizan la plataforma para interactuar en comunidades.

- Administradores: Usuarios con permisos para gestionar comunidades y moderar contenido.

- Moderadores: Usuarios designados para ayudar en la gestión y moderación de comunidades.

2.4 **Restricciones**

- La plataforma debe cumplir con las regulaciones de privacidad de datos, como el GDPR.

- El desarrollo debe completarse en un marco temporal de 6 meses.

- ASP.NET Core como framework para el desarrollo web

- La aplicación debe ser desarrollada utilizando C# como lenguaje principal para la lógica del servidor.

2.5 **Suposiciones y Dependencias**

- Los usuarios tendrán acceso a dispositivos con navegadores modernos.

- La plataforma estará hospedada en un entorno escalable en la nube.

2

**Requerimientos Especificos**

**3.1 Requerimientos Funcionales**

1. **Registro y Autenticación de Usuarios**

- 1.1 - Se le permitirá al usuario registro con mail y contraseña.

- 1.2 - Se le pedirá confirmación de correo electrónico al usuario.

- 1.3 - El usuario podrá recuperar su contraseña.

- 1.4 - El usuario podrá iniciar sesión con correo electrónico y contraseña.

2. **Gestión de Perfil**

- 2.1 - El usuario podrá actualizar su información personal.

- 2.2 - El usuario podrá cambiar la contraseña.

- 2.3 - El usuario podrá configurar la privacidad de su perfil.

3. **Comunidades**

- 3.1 - El usuario podrá visualizar y visitar comunidades.

- 3.2 - Cada usuario podrá crear comunidades.

- 3.3 - Se le pedirá al usuario agregar información de la comunidad (título, descripción, categoría, foto de comunidad, privada o pública).

- 3.4 - El usuario será el administrador si este ha creado la comunidad.

- 3.5 - El administrador de la comunidad puede elegir a otros usuarios para moderar su propia comunidad.

- 3.6 - Los administradores podrán moderar la comunidad (moderar posts, eliminar usuarios de la comunidad).

- 3.7 - El usuario se podrá unir a comunidades (en caso de ser privada se necesita confirmación de cualquier miembro que ya esté dentro de la comunidad).

4. **Creación y Gestión de Contenido**

- 4.1 - El usuario podrá crear publicaciones (texto, archivos y links).

- 4.2 - El usuario puede administrar sus publicaciones (eliminar y editar publicaciones propias únicamente).

- 4.3 - El usuario podrá interactuar con publicaciones (dar likes y comentarios).

5. **Mensajería y Notificaciones**

- 5.1 - Los usuarios se podrán mandar mensajes privados (únicamente de un usuario a un usuario).

- 5.2 - Los usuarios recibirán notificaciones (ya sea de comunidades o privadas).

6. **Búsqueda y Descubrimiento**

- 6.1 - Los usuarios podrán buscar tanto posts como usuarios y comunidades.

7. **Moderación y Seguridad**

- 7.1 - Los administradores podrán moderar contenido dentro de su propia comunidad (permitir qué tipo de contenido se enviará o no estará permitido).

**8. Eventos**

- 8.1 - Los usuarios podrán crear eventos dentro de las comunidades.

- 8.2 - Los eventos deben incluir información como título, descripción, fecha, hora y lugar.

- 8.3 - Los eventos pueden ser visibles para todos los miembros de la comunidad.

- 8.4 - Los usuarios pueden unirse a eventos y recibir notificaciones relacionadas.

3

**3.2 Requerimientos No Funcionales**

1. **Rendimiento y Escalabilidad**

- 1.1 - El tiempo de respuesta de la plataforma debe ser inferior a 2 segundos en el 95% de las interacciones.

- 1.2 - La plataforma debe ser capaz de soportar al menos 10,000 usuarios concurrentes sin degradación significativa del rendimiento.

- 1.3 - La infraestructura debe ser escalable para manejar un crecimiento del 200% en la carga de usuarios.

2. **Disponibilidad y Fiabilidad**

- 2.1 - La plataforma debe tener una disponibilidad del 99.9%, lo que equivale a un tiempo de inactividad máximo de 8.76 horas al año.

- 2.2 - En caso de fallos, el sistema debe recuperar todas las funcionalidades en menos de 1 hora.

3. **Seguridad**

- 3.1 - Los datos de los usuarios deben estar encriptados tanto en tránsito como en reposo utilizando estándares de cifrado fuertes.

- 3.2 - Implementar autenticación en dos pasos para el acceso de usuarios.

- 3.3 - Monitorear continuamente y alertar sobre actividades sospechosas en las cuentas de usuario.

**4. Restricciones de Diseño**

- 4.1 - La interfaz de usuario debe ser desarrollada siguiendo el principio de diseño responsivo, asegurando que la aplicación sea usable en computadoras de escritorio.

- 4.2 - El diseño debe seguir las demostraciones dadas en los bocetos para mantener la consistencia y usabilidad.

5. **Compatibilidad**

- 5.1 - La plataforma debe ser compatible con las versiones actuales y dos versiones anteriores de los navegadores web más utilizados (Chrome, Firefox, Safari, Edge).

- 5.2 - Debe funcionar sin problemas en dispositivos móviles, tabletas y computadoras de escritorio, garantizando una experiencia uniforme.

6. **Mantenibilidad**

- 6.1 - El código fuente debe ser modular y bien documentado, siguiendo estándares de codificación para facilitar el mantenimiento y la actualización.

- 6.2 - Debe ser posible implementar actualizaciones y parches sin tiempos de inactividad significativos.

7. **Monitoreo y Logística**

- 7.1 - Implementar herramientas de monitoreo para el rendimiento del sistema y el comportamiento de los usuarios.

- 7.2 - Configurar alertas automáticas para detectar y notificar problemas de rendimiento y seguridad en tiempo real.

Directorio con Diagrama Caso de Uso - <https://github.com/ErTite/Trabajos-ADA.git>

4

**Referencias**

-IEEE 830: Estándar para la especificación de requerimientos de software.

-Scrum Guide: Guía oficial de la metodología Scrum.

-WCAG 2.1: Pautas de Accesibilidad para el Contenido.

-Web.ISO/IEC 25010:2011: Estándar de Calidad de Producto de Software.

-NIST SP 800-53: Controles de Seguridad y Privacidad para Sistemas de Información Federales.

-Software Requirements Specification (SRS) Examples.

-GDPR Overview.

5

**Enlaces**

**Fuente de requerimientos:**

<https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>

**Documentación Scrum:**

<https://www.scrum.org/resources/what-is-scrum> - <https://www.scrumguides.org/scrum-guide.html>

**Modelo IEEE830:**

<https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf> -

**GDRP Overview:**

<https://gdpr.eu/what-is-gdpr/>

**WCAG 2.1:**

<https://www.w3.org/TR/WCAG21/>

**Software Requirements Specification (SRS) Examples:**

<https://www.geeksforgeeks.org/software-requirements-specification-srs/>

6