Evaluación del Comportamiento del Software - Plan SQA

Luis Alberto Dueñas Franco, Javier Medrano Hernandez, Lukas Alejando Díaz Cabana, Yani Luna Vigoya, Jhon Freyman Torres Argumedo

Servicio Nacional de Aprendizaje (SENA), Centro para la Industria Petroquímica (CIP)

Análisis y Desarrollo de Software (ADSO), Controlar la Calidad del Servicio de Software

Inst. Mara Sofía Cabrales

04 de Diciembre del 2025

Tabla de Contenido

Propósito del Documento

El propósito de este documento es definir, organizar y comunicar la planeación del diseño, desarrollo, verificación y el aseguramiento de la calidad del sistema Red Social para Aprendices del SENA.

Objetivo General

Describir de manera estructurada todos los elementos fundamentales del sistema con el fin de garantizar su correcta comprensión, implementación, evaluación y mantenimiento, proporcionando una base normativa para la calidad del software a lo largo del proyecto.

Objetivos Específicos

- Establecer los requerimientos funcionales y no funcionales del sistema.
- Definir el alcance y los límites de la solución tecnológica.
- Describir la estructura conceptual y arquitectónica del sistema.
- Determinar los lineamientos de verificación y validación requeridos para asegurar el cumplimiento de los estándares de calidad.
- Proporcionar definiciones, criterios y pautas que permitan una comunicación clara entre los distintos actores del proyecto.

Definiciones y Abreviaturas

RF: Requisito Funcional.

RNF: Requisito No Funcional.

SQA: Software Quality Assurance (Aseguramiento de Calidad de Software).

IA: Inteligencia Artificial.

Chat grupal: Espacio de mensajería entre tres o más usuarios dentro de la plataforma.

Moderador: Usuario con permisos especiales para gestionar contenido inapropiado.

Validación externa: Proceso de verificación del documento del usuario mediante un servicio autorizado por el SENA.

Feed: Sección principal de publicaciones visibles para el usuario.

HTTP: Los comandos estándar del protocolo HTTP utilizados para interactuar con los recursos (GET, POST, PUT, DELETE).

API REST : Es un estilo arquitectónico para el diseño de sistemas de *software* en red, que utiliza el protocolo **HTTP** para la comunicación. Una API que sigue los principios de REST se llama **RESTful**.

SCM (Software Configuration Management): Proceso sistemático para identificar, organizar, versionar, controlar y auditar los elementos de configuración del software durante todo su ciclo de vida.

Elemento de Configuración (CI – Configuration Item): Artefacto del proyecto sujeto a control de versiones, como código fuente, documentación, diagramas, scripts, casos de prueba y archivos de configuración.

Repositorio: Espacio centralizado en GitHub donde se almacenan, versionan y gestionan los elementos de configuración del proyecto.

Control de Versiones: Mecanismo que permite registrar, rastrear y recuperar cambios realizados sobre los elementos de configuración a lo largo del tiempo.

Git: Sistema de control de versiones distribuido utilizado para gestionar el código fuente y los artefactos del proyecto.

GitHub: Plataforma colaborativa utilizada como repositorio remoto, gestor de cambios, control de versiones y soporte del flujo de trabajo del proyecto.

Rama (Branch): Línea independiente de desarrollo dentro del repositorio que permite trabajar cambios sin afectar la versión estable del sistema.

Rama principal (main): Rama que contiene la versión estable y aprobada del sistema, lista para entrega o despliegue.

GitHub Flow: Modelo de trabajo basado en ramas cortas, donde cada cambio se desarrolla en una rama independiente y se integra mediante revisiones antes de ser fusionado a la rama principal.

Pull Request (PR): Solicitud formal para integrar cambios de una rama a otra, sujeta a revisión técnica y validación antes de su aprobación.

Merge: Proceso de integración de los cambios aprobados desde una rama de desarrollo hacia la rama principal.

Commit: Registro de un conjunto de cambios realizados sobre uno o más elementos de configuración, acompañado de un mensaje descriptivo.

Historial de Cambios: Registro cronológico de commits que permite auditar qué se cambió, cuándo y por quién.

Control de Cambios: Procedimiento formal para solicitar, evaluar, aprobar o rechazar modificaciones sobre los elementos de configuración del sistema.

Solicitud de Cambio: Petición documentada que describe una modificación propuesta, su impacto y justificación antes de ser implementada.

Baseline (Línea Base): Versión aprobada y estable de un conjunto de elementos de configuración que sirve como referencia para futuras modificaciones.

Auditoría de Configuración: Actividad de verificación que asegura que los elementos de configuración cumplen con lo definido y que los cambios fueron correctamente autorizados y documentados.

Requerimiento y Análisis del Sistema

Especificación de Requerimientos

Los requisitos funcionales (o RF) que se presentan fueron tomados de nuestro tablero de Scrumban que se encuentra en nuestro repositorio de GitHub Projects (https://github.com/users/Louis-Du/projects/6/views/1) a continuación se presentarán estos requisitos funcionales mediante módulos, resumiéndolos y evitando llenar el documento con los 41 requisitos funcionales. Si se desea ver cada requisito más detalladamente se invita a visitar el enlace anterior.

La siguiente tabla utiliza una estructura donde la primera columna pertenece a los módulos antes mencionados, seguida de otra donde se describe el propósito del módulo, y la última columna se presentan los RF asociados.

Tabla 1 - Requisitos Funcionales por Modulos

Módulo	**Descripción**	**Requisitos Funcionales (RF) Asociados**
Autenticación y Registro	Gestiona los procesos de identificación, creación de cuentas y acceso seguro al sistema por parte de aprendices y egresados.	RF-01 a RF-07
Perfil y Personalización	Permite la administración y actualización de la información personal del usuario, así como	RF-08 a RF-14

Tabla 1 - Requisitos Funcionales por Modulos		
Módulo	**Descripción**	**Requisitos Funcionales (RF) Asociados**
	la personalización de su espacio dentro del sistema.	
Publicaciones	Se encarga de la creación, visualización, edición y filtrado del contenido generado por los usuarios dentro de la red social.	RF-15 a RF-22
Comentarios y Reacciones	Gestiona la interacción secundaria entre usuarios mediante comentarios, respuestas y reacciones sobre las publicaciones.	RF-23 a RF-28
Chats y Mensajería	Provee comunicación privada y grupal entre usuarios, así como el envío de archivos y medios dentro de las conversaciones.	RF-29 a RF-34

Tabla 1 - Requisitos Funcionales por Modulos		
Módulo	**Descripción**	**Requisitos Funcionales (RF) Asociados**
Moderación e Inteligencia Artificial	Encargado del análisis de contenido, detección de material inapropiado y mecanismos de reputación, así como herramientas de reporte y acciones de moderación.	RF-35 a RF-39
Noticias	Permite la visualización y actualización de noticias oficiales del SENA dentro de la plataforma.	RF-40 y RF-41

Para la siguiente tabla se abarca los requisitos no funcionales (RNF) del sistema, donde la primera columna se presenta los RNF según la ISO 25010, seguida de otra donde se desglosa los RNF, y por último una columna donde se encuentra la descripción de cada RNF desglosado.

Tabla 2 - Requisitos no Funcionales		
RNF Según la ISO 25010	**RNF Desglosado**	**Descripción**
Rendimiento	RNF-01. Tiempo de Respuesta	El sistema debe responder a cualquier acción del usuario (cargar publicaciones, abrir un

		perfil, enviar un mensaje) en menos de 2 segundos bajo condiciones normales de carga.
	RNF-02. Carga Máxima	El sistema debe soportar al menos 500 usuarios concurrentes sin degradación perceptible del rendimiento.
	RNF-03. Escalabilidad	La arquitectura debe permitir aumentar la capacidad horizontalmente para soportar tráfico futuro, sin rediseño profundo.
Confiabilidad	RNF-04. Disponibilidad	El sistema debe estar disponible al menos 99% del tiempo mensual, excluyendo mantenimientos programados.
	RNF-05. Tolerancia a Errores	Ante una falla en un módulo, el sistema deberá continuar operando con funcionalidades esenciales intactas.
	RNF-06. Recuperación	Después de una caída inesperada, el sistema debe restaurarse automáticamente en menos de 1 minuto.

Seguridad	RNF-07. Cifrado	Toda comunicación debe viajar cifrada mediante HTTPS/TLS 1.2 o superior
	RNF-08. Protección de Datos	La información sensible del usuario debe almacenarse cumpliendo principios mínimos de privacidad: contraseñas con hashing fuerte (bcrypt/argon2), control de acceso y registros de auditoría.
Usabilidad	RNF-09. Accesibilidad	La interfaz debe cumplir con criterios básicos WCAG 2.1 nivel AA (contrastes, navegación clara, textos descriptivos).
	RNF-10. Intuición de Uso	El usuario debe poder completar tareas básicas (registrarse, publicar, comentar) sin instrucciones adicionales.
	RNF-11. Consistencia Visual	Todos los módulos deben seguir la misma línea gráfica y los mismos patrones de interacción.
Mantenibilidad	RNF-12. Estructura del Código	El repositorio debe mantener

		una arquitectura modular con separación clara entre frontend, backend y servicios.
	RNF-13. Documentación Técnica	Cada módulo debe contar con documentación mínima en el repositorio README.
	RNF-14. Control de Versiones	Todo el código debe gestionarse mediante GitHub, con ramas separadas y revisiones antes de un merge.
Portabilidad	RNF-15. Navegadores Soportados	La aplicación web debe funcionar correctamente en las últimas dos versiones estables de Chrome, Firefox y Edge.
	RNF-16. Adaptabilidad	El diseño debe responder correctamente en pantallas móviles, tabletas y equipos de escritorio.
Compatibilidad	RNF-17. Integración con IA/Moderación	La capa de moderación asistida por IA debe funcionar como servicio desacoplado, de modo que pueda actualizarse o reemplazarse sin afectar el resto del sistema.

Alcance del Sistema

El sistema tiene como propósito ofrecer las funcionalidades esenciales que permitan a los usuarios (aprendices y egresados) interactuar de forma eficiente dentro de la red social. A continuación, se presenta el alcance funcional confirmado para esta versión del proyecto.

1) Gestión de Usuarios
    a) Registro de usuarios nuevos.
    b) Validación de documento mediante servicio externo autorizado.
    c) Autenticación (inicio y cierre de sesión).
    d) Recuperación de contraseña.
    e) Actualización y personalización del perfil del usuario (datos, foto, banner).
2) Gestión de Publicaciones
    a) Creación de publicaciones con texto, imágenes, videos o documentos.
    b) Edición y eliminación de publicaciones propias.
    c) Visualización y filtrado del muro de contenido.
    d) Reacciones a publicaciones.
3) Gestión de Comentarios
    a) Creación, edición y eliminación de comentarios.
    b) Respuestas a comentarios.
    c) Reacciones a comentarios.
4) Sistema de Mensajería
    a) Envío y recepción de mensajes privados.
    b) Envío de imágenes y documentos mediante mensajería.
    c) Creación y administración de chats grupales.
5) Moderación y Seguridad
    a) Reporte de publicaciones y comentarios por parte del usuario.
    b) Detección automatizada de contenido inapropiado mediante técnicas de IA.
    c) Bloqueo de contenido por parte del administrador o moderador.

6) Gestión de Noticias Institucionales
    a) Visualización de noticias oficiales.
    b) Creación, actualización y eliminación de noticias por parte del administrador.

Aunque el sistema contempla un conjunto amplio de funcionalidades orientadas a la experiencia del usuario, se excluyen las siguientes capacidades por no ser esenciales en esta etapa del proyecto:

1) Procesos académicos (calificaciones, matrículas, horario, certificados).
2) Integración con sistemas educativos externos (Moodle, Teams, Blackboard, etc.).
3) Llamadas de voz o videollamadas.
4) Publicidad comercial, marketplace o anuncios pagados.
5) Analítica avanzada de actividad (tendencias, estadísticas por usuario, minería de datos).
6) Moderación humana externa o servicio de soporte en tiempo real.
7) Funciones administrativas del SENA distintas a noticias y moderación de contenido.
8) Integración con redes sociales externas (Facebook, Instagram, WhatsApp).
9) Almacenamiento o tratamiento de información adicional proveniente del servicio externo de validación, excepto la confirmación de autenticidad del documento.

Conociendo las funciones que incluirá y no el sistema ahora se aclara los límites de nuestro programa:

- Límite con Usuarios
    - El sistema interactúa únicamente con aprendices, egresados y administradores registrados.
    - No se permite acceso a usuarios externos al SENA.
- Límite con Servicio Externo
    - La interacción con sistemas externos se limita exclusivamente a la validación del documento en el proceso de registro.

    - No se establecen otras integraciones ni flujos de información con aplicaciones externas.

- Límite Tecnológico
    - El sistema depende de la infraestructura de hosting y base de datos definida por el equipo de desarrollo.
    - No se gestionan recursos físicos ni redes internas del SENA.
- Límite de Seguridad
    - La moderación automática detecta contenido inapropiado, pero no garantiza la revisión exhaustiva de todo el contenido.
    - El sistema no administrará comportamientos fuera de la plataforma ni interacciones ocurridas fuera del entorno de la aplicación.

Pautas para la Interfaz de Usuario

A continuación se presenta algunas de las pautas que se debe de tener en cuenta al momento de realizar la interfaz tomando en cuenta que se busca establecer un programa accesible, seguro y limpio para todo tipo de usuario:

1. Principios de accesibilidad.

   Como se mencionó con anterioridad, el sistema deberá cumplir con los lineamientos básicos de accesibilidad para garantizar que todos los usuarios puedan interactuar sin impedimentos.

   Esto lo podremos realizar aplicando las siguientes características:

   - Contrastes adecuados entre texto y fondo para asegurar legibilidad en todas las vistas.
   - Tamaños tipográficos mínimos: (14 pt en dispositivos móviles, 16 pt en pantallas grandes.)
   - Indicadores visuales inequívocos para elementos interactivos (botones, menús, enlaces, íconos).

- Navegación comprensible para usuarios con experiencia limitada en aplicaciones digitales.
- Mensajes de error informativos, visibles y específicos.
- Evitar depender exclusivamente del color para transmitir estados o información relevante.
- Compatibilidad con dispositivos móviles, tabletas y equipos de escritorio.

2. Estilo visual.

   El estilo visual se definirá bajo los principios de consistencia, simplicidad y alineación con la identidad del SENA. De acuerdo con el prototipo proporcionado, se establecen las siguientes características:

   - Uso de la paleta institucional del SENA (Verde SENA (#39A900), azul oscuro (#00304D) y tonos grises de apoyo, esto según el manual de identidad corporativa del SENA)
   - Interfaz minimalista orientada al contenido, con jerarquía clara de textos, espacios y secciones.
   - Botones con bordes redondeados y sombreados suaves, manteniendo consistencia entre las vistas del feed, chat, perfil y noticias.
   - Iconografía unificada (Lucide Icons), tal como se evidencia en el prototipo.
   - Estructuras de tarjetas (cards) para publicaciones, noticias y perfiles, preservando la estética definida en el prototipo.
   - Dock lateral claramente visible y persistente.

3. Flujo básico de navegación.

   El sistema definirá un flujo de navegación directo, predecible y fácil de seguir:

   - Pantalla de acceso:
     - Bienvenida
     - Selección de tipo de documento

    - Autenticación

- Ingreso al sistema:
    - Acceso inmediato al feed principal con (Botón para crear publicación (modal o panel rápido), Filtros avanzados, Vista de publicaciones recientes).
- Menú principal (persistente):
    - Inicio (feed)
    - Noticias del SENA
    - Perfil del usuario
    - Chats privados y grupales
    - Configuración
- Navegación estructurada por módulos:
    - Publicaciones: Crear, visualizar, comentar, filtrar.
    - Noticias: Consultar novedades oficiales.
    - Perfil: Visualizar, editar y consultar información de formación.
    - Chat: Interacción directa con aprendices o grupos.
    - Ver perfil de otro usuario: Vista separada con su feed y datos públicos.
- Regla de los tres clics para todas las funciones críticas (crear publicación, enviar comentario, abrir un chat, filtrar contenido) deben ser accesibles en máximo tres acciones.
- Retroalimentación inmediata:
    - Confirmación de acciones.
    - Notificaciones visuales de envío, éxito o error.
    - Actualización instantánea de comentarios, posts y mensajes (simulada en el prototipo).

Actividades dentro del Área de Diseño.

- Revisión de prototipos: revisión de wireframes y maquetas antes de pasar a desarrollo, con checklist de usabilidad y consistencia visual.
- Pruebas de usabilidad: pequeñas pruebas con aprendices para validar que entienden cómo publicar, comentar, chatear, filtrar contenido y ver noticias del SENA.
- Validación de diseño por historia: cada historia de usuario de interfaz no pase a “Listo” sin revisar diseño (alineación, errores de texto, componentes reutilizables).

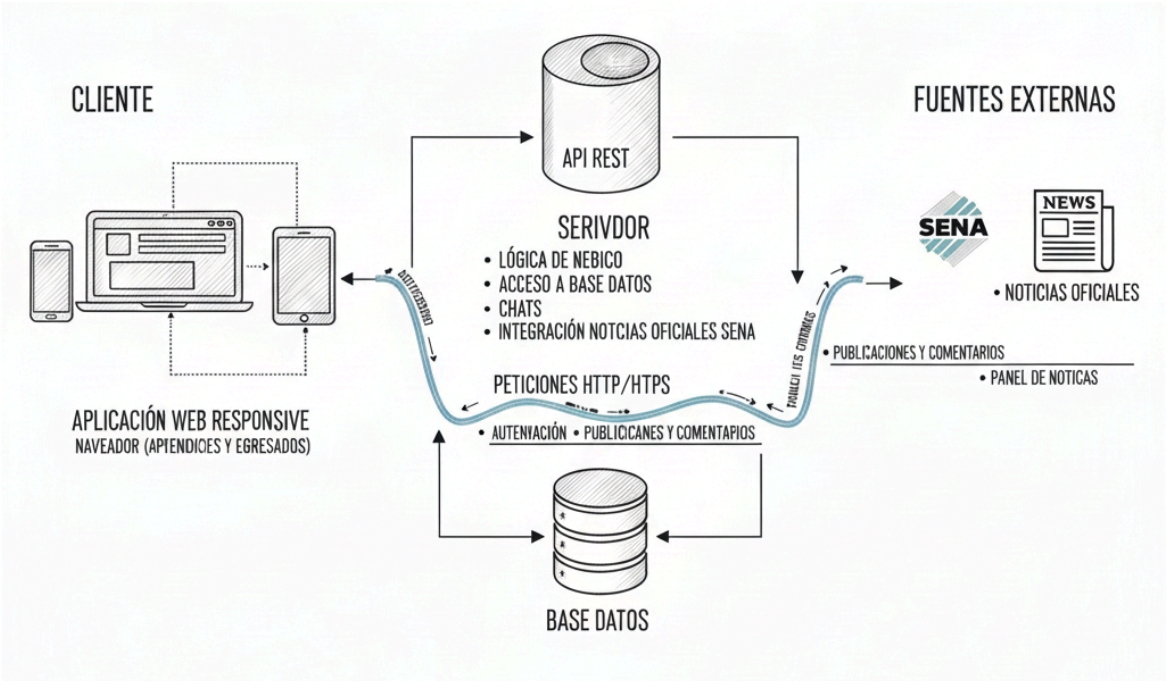
Descripción de la arquitectura

La aplicación se basa en una arquitectura servicio-cliente. El cliente es una aplicación web responsive que se ejecuta en el navegador de los aprendices y egresados, mientras que el servidor expone una API REST que gestiona la lógica de negocio, el acceso a la base de datos y la integración con las fuentes ñoficiales de noticias del SENA. El cliente se comunica con el servidor mediante peticiones HTTP/HTTPS para autenticación, gestión de publicaciones y comentarios, chats, perfiles de usuario y panel de noticias.

Nota sobre el estado actual del sistema

La arquitectura descrita corresponde a la visión final del sistema. En la fase actual del proyecto, se dispone únicamente de un prototipo frontend estático, utilizado para validar diseño, navegación y experiencia de usuario.

La implementación del backend en C#, la API REST, la base de datos y los servicios de autenticación y mensajería se abordarán en fases posteriores del desarrollo.



.

Implementación, Verificación e Implantación

Plan de pruebas unitarias

Objetivo: Definir la estrategia y procedimientos a realizar para validar que cada unidad del código funcione de forma independiente, correcta y consistente para garantizar la calidad del código en el proyecto.

- **Enfoque - pruebas manuales**

  Estás pruebas consisten en que un evaluador ejecuta paso a paso las funcionalidades de la aplicación.

- **Enfoque de pruebas por versión**

  El proyecto se encuentra actualmente en una fase de prototipo funcional frontend, desarrollado en HTML, CSS y JavaScript, por lo que las pruebas unitarias automatizadas no se encuentran implementadas en esta etapa.

  Para la versión objetivo del sistema, se tiene planificado el uso de C# (.NET) como tecnología backend y la herramienta xUnit.net para la ejecución de pruebas unitarias automatizadas sobre la lógica de negocio y los servicios expuestos por la API REST.

En la versión actual del prototipo, la verificación se realiza mediante pruebas manuales, listas de chequeo y validaciones de comportamiento en el navegador.

Las pruebas se realizan en dispositivos móviles (Android y iOS) y dispositivos de escritorio.

Casos de pruebas

Tipo de caso de prueba	**Descripción**	**Pasos de prueba**	**Resultados esperados**	**Estado**
Integración	Validar integración con el sistema de verificación de identidad	Registrar usuario con documento real y verificar la respuesta del sistema	El sistema obtiene los datos del aprendiz y los carga en el perfil automáticamente	Pendiente
Funcionalidad	Verificar que el registro por documento funcione correctamente	Ingresar número de documento y enviar formulario de registro	Que el sistema valide el registro y cree la cuenta exitosamente	Pendiente
Funcionalidad	Realizar una publicación en el muro	Iniciar sesión, escribir una publicación y enviar	La publicación aparece visible para otros usuarios en el muro	Pendiente
Rendimiento	Probar carga del muro con múltiples publicaciones	Desplazarse por el muro y cargar más publicaciones	El muro carga el contenido sin retrasos ni errores	Pendiente
Funcionalidad	Comentar una	Escribir un	El comentario se	Pendiente

	publicación	comentario en una publicación existente y enviar	muestra bajo la publicación seleccionada	
Seguridad	Verificar que solo usuarios registrados accedan a un chat	Intentar acceder al chat sin iniciar sesión	El sistema bloquea el acceso y redirige al usuario al inicio de sesión	Pendiente
Chat	Enviar mensajes privados	Abrir chat privado y enviar mensajes	El mensaje aparece en el historial del chat de inmediato	Pendiente
Chat grupal	Verificar envío de mensajes en grupos	Entrar en un grupo y verificar que otros usuarios reciban los mensajes	Los mensajes se muertas para todos los miembros del grupo sin retraso	Pendiente
Usabilidad	Verificar que los enlaces de los paneles de noticias abren correctamente	Hacer click en diferentes noticias del panel	Cada noticia abre su correspondiente enlace correctamente	Pendiente
Usabilidad	Comprobar que la plataforma es responsiva (móvil -> escritorio)	Abrir la aplicación en tamaño móvil y cambiar a tamaño escritorio	Todos los elementos se ajustan correctamente sin desbordes	Pendiente

Plan de validación y verificación

El plan de validación asegura que el software satisface las necesidades reales del usuario y que funcione de acuerdo a lo esperado.

Actividades de verificación	Resultados esperados
Pruebas funcionales del sistema completo	● El sistema cumple los requisitos funcionales. ● Los usuarios pueden registrarse, publicar, comentar y chatear sin problemas. ● La interfaz es clara, rápida y fácil de usar. ● El software está listo para ser desplegado y utilizado.
Pruebas de usabilidad en dispositivos móviles y escritorio	
Pruebas de aceptación por parte de aprendices y egresados	
Pruebas de compatibilidad en navegadores y dispositivos	
Pruebas de rendimiento en el muro, chat y carga general	

El plan de verificación asegura que el software se construya correctamente y que cumpla con los requisitos técnicos definidos.

Actividades de verificación	Resultados esperados
Revisión de requisitos para garantizar consistencia	● El código cumple con los estándares definidos. ● Los módulos funcionan correctamente de manera individual y combinada. ● Se detectan y corrigen errores antes de la entrega del producto al usuario.
Revisión del diseño y arquitectura del sistema	
Revisión de estándares del código	
Pruebas unitarias en funciones, clases y módulos	
Pruebas de integración entre componentes	

Metodología de Desarrollo

El proyecto se desarrollará bajo un enfoque ágil híbrido, combinando prácticas de Scrum y Kanban (Scrumban), con el fin de permitir flexibilidad en la planificación, entrega incremental de funcionalidades y control continuo de la calidad del software.

La metodología adoptada prioriza:

- Entregas incrementales y funcionales.
- Validación temprana de requisitos.
- Retroalimentación constante.
- Aseguramiento de la calidad integrado a cada iteración.

Enfoque Metodológico

Scrumban como marco de trabajo principal para la gestión del flujo de tareas.

- GitHub Projects como tablero visual del backlog.
- Iteraciones (Sprints) con objetivos definidos y entregables verificables.
- GitHub Flow como estrategia de control de versiones.
- SQA continuo, integrado desde el análisis hasta el despliegue.
- Integración del SQA en la Metodología

Aplicación del aseguramiento de la calidad

El aseguramiento de la calidad no se limita a una fase final, sino que se aplica de forma transversal durante todo el ciclo de vida del software:

1. Revisión de requisitos antes de su implementación.
2. Validación de diseño por historia de usuario.
3. Pruebas unitarias y funcionales durante cada sprint.
4. Revisión de código mediante Pull Requests.
5. Control formal de cambios y actualización de la línea base.

Esta integración garantiza trazabilidad, control, consistencia técnica y alineación con los estándares de calidad definidos en el presente Plan SQA.

Gestión del Backlog y Flujo de Trabajo

El proyecto utiliza un tablero **Scrumban** implementado en **GitHub Projects** como herramienta central para la planificación, seguimiento y control del trabajo. El tablero organiza las actividades del proyecto en columnas que representan el estado real de cada elemento dentro del flujo de desarrollo, permitiendo visibilidad, control y trazabilidad continua, algunas columnas se le establece un límite de actividades para evitar posibles cuellos de botellas y garantizar un flujo de trabajo directo. El flujo de trabajo se compone de las siguientes columnas:

- **Backlog**
  Contiene la lista general de ideas, requisitos funcionales (RF) y actividades planeadas para el proyecto.
  Sin límite en específico

- **To Do**
  Actividades priorizadas y listas para iniciar en la siguiente iteración o ciclo de trabajo.
  Límite: 6

- **In Progress**
  Actividades que se encuentran actualmente en desarrollo.
  Límite: 5

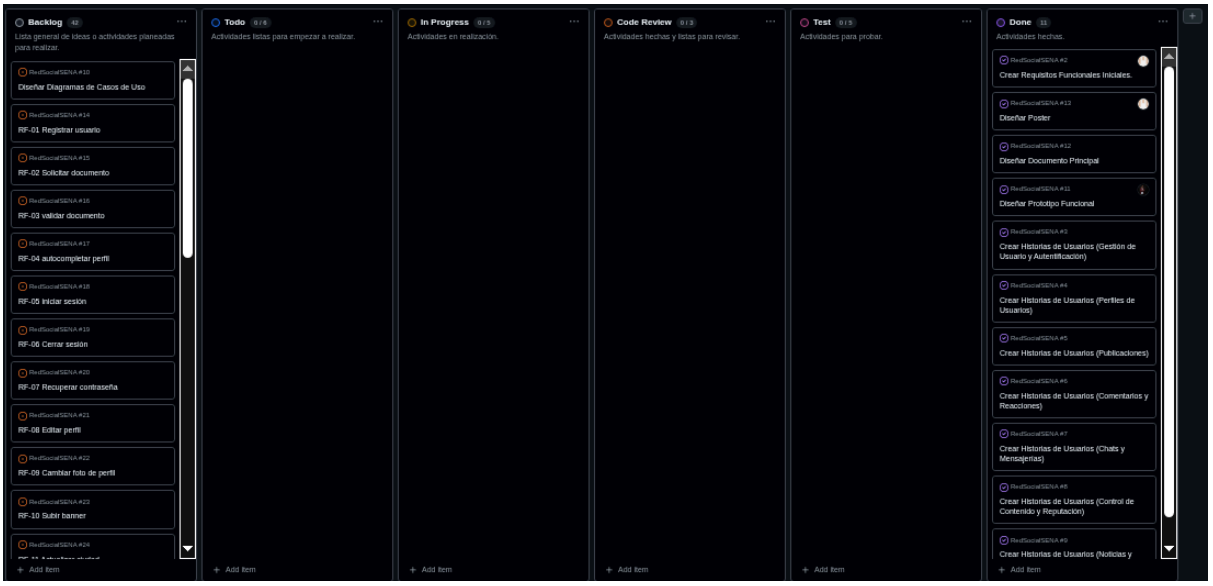
- **Code Review**
  Actividades finalizadas a nivel de implementación y pendientes de revisión técnica mediante Pull Requests.
  Límite: 2

- **Test**
  Actividades listas para pruebas funcionales, de usabilidad o verificación de requisitos.
  Límite: 5

- **Done**
  Actividades completadas, validadas y aceptadas conforme a los criterios definidos.
  Sin límite en específico

Tablero Scrumban

A continuación se presenta el tablero de scrumban realizado evidenciando la estructura antes mencionada:


Backlog 42
Lista general de ideas o actividades planeadas para realizar.
RedSocialSENA #19
Diseñar Diagramas de Casos de Uso
RedSocialSENA #14
RF-01 Registrar usuario
RedSocialSENA #15
RF-02 Solicitar documento
RedSocialSENA #16
RF-03 validar documento
RedSocialSENA #17
RF-04 autocompletar perfil
RedSocialSENA #18
RF-05 iniciar sesión
RedSocialSENA #19
RF-06 Cerrar sesión
RedSocialSENA #20
RF-07 Recuperar contraseña
RedSocialSENA #21
RF-08 Editar perfil
RedSocialSENA #22
RF-09 Cambiar foto de perfil
RedSocialSENA #23
RF-10 Subir banner
RedSocialSENA #24
Add item
Todo 0 / 6
Actividades listas para empezar a realizar.
Add item
In Progress 0 / 5
Actividades en realización.
Add item
Code Review 0 / 3
Actividades hechas y listas para revisar.
Add item
Test 0 / 5
Actividades para probar.
Add item
Done 11
Actividades hechas.
RedSocialSENA #2
Crear Requisitos Funcionales Iniciales.
RedSocialSENA #13
Diseñar Poster
RedSocialSENA #12
Diseñar Documento Principal
RedSocialSENA #11
Diseñar Prototipo Funcional
RedSocialSENA #3
Crear Historias de Usuarios (Gestión de Usuario y Autentificación)
RedSocialSENA #4
Crear Historias de Usuarios (Perfiles de Usuarios)
RedSocialSENA #5
Crear Historias de Usuarios (Publicaciones)
RedSocialSENA #6
Crear Historias de Usuarios (Comentarios y Reacciones)
RedSocialSENA #7
Crear Historias de Usuarios (Chats y Mensajerías)
RedSocialSENA #8
Crear Historias de Usuarios (Control de Contenido y Reputación)
RedSocialSENA #9
Crear Historias de Usuarios (Noticias y
Add item

GESTIÓN DEL PROYECTO

Plan del proyecto

Entregables:

- Documentación de requerimientos del sistema.
- Lista de chequeos.
- Tablero Scrumban, historias de usuarios
- Prototipo del sistema.
- Matriz de trazabilidad y casos de pruebas

Roles de equipo

RESPONSABLE	ROL
Luis Dueñas	Desarrollador - Analista - Tester
Jhon Torres	Desarrollador - Analista - Tester
Yani Luna	Desarrollador - Analista - Tester
Javier Medrano	Desarrollador - Analista - Tester
Lukas Diaz	Desarrollador - Analista - Tester

Cronograma del proyecto

Sprint	Duración	Objetivos / Entrega funcional	Funcionalidades incluidas
Sprint 1	Semanas 1 - 4	Producto mínimo viable funcional de la red social	Registro y login básico, creación de perfil inicial, publicaciones de texto, visualización del feed, navegación básica responsive.
Sprint 2	Semanas 5 - 8	Red social funcional con verificación de usuarios	Verificación por número de documento, autocompletado de perfil, mejoras en autenticación y seguridad, validaciones de formularios

Sprint 3	Semanas 9 - 12	Interacción avanzada entre usuarios.	comentarios en publicaciones, sistema de votaciones, optimización de feed, mejoras UI/UX, control básico de abuso.
Sprint 4	Semanas 13 - 17	Comunicación privada funcional.	Chat privado en tiempo real, lista de contactos, indicadores en línea, notificaciones básicas.
Sprint 5	Semanas 18 - 21	Plataforma informativa y personalizable.	Panel de noticias oficiales del SENA, personalización avanzada del perfil, mejoras del diseño y moderación.
Sprint 6	Semanas 22 - 25	Sistema estable listo para entrega.	Integración completa, corrección de defectos, optimización de rendimiento, pruebas completas, documentación final.

Gestión de riesgos

Riesgos Técnicos

- Fallos en verificación
- Problemas con chat en tiempo real
- Incompatibilidades en dispositivos

Riesgos de planificación

- Retraso en chat o verificación
- subestimación del tiempo de pruebas

Riesgos de calidad

- Baja accesibilidad en dispositivos móviles
- Baja usabilidad
- Vulnerabilidad en el manejo de contenido
- Bajo nivel de pruebas automatizadas
- Incompatibilidad entre navegadores

Plan de iteración

Actividades por iteración

1. Refinamiento del Backlog
2. Planificación del Sprint
3. Diseño rápido
4. Desarrollo frontend y backend
5. Pruebas unitarias del desarrollador
6. Pruebas QA
7. Integración y entrega del incremento
8. Revisión del Sprint
9. Retrospectiva

Revisión de Avances

- Reuniones diarias
- Revisión del Sprint
- Auditorías de calidad
- Seguimiento en GitHub

Plan para la Gestión de la Configuración y Control de Cambios

Elementos de Configuración (CI)

Los siguientes elementos se considerarán como elementos de configuración del proyecto:

Tipo de CI	Descripción	Ubicación en Repositorio
Código fuente	Archivos HTML, CSS y JavaScript correspondientes al prototipo funcional de la aplicación web	/prototipo/
Documentación	Plan SQA, informes de evaluación, listas de chequeo, matrices y documentos de soporte	/documents/
Gestión de Proyecto	Historias de usuario (RF), issues, tablero Scrumban y planificación	GitHub Issues + Projects
Configuración del Repo	Archivos .gitignore, configuración de GitHub Actions, reglas del repositorio	.github/, raíz
Pruebas	Casos de prueba, listas de chequeo de requisitos no funcionales	/documents/
Despliegue	Configuración y versión publicada en GitHub Pages	GitHub Pages (gh-pages)
Templates	Plantillas de Issues y Pull Requests	.github/ISSUE_TEMPLATE/, .github/PULL_REQUEST_TEMPLATE.md

Herramientas de Gestión de la Configuración

La gestión de la configuración se realizará mediante las siguientes herramientas:

- Sistema de control de versiones: Git
- Plataforma de repositorio: GitHub
- Gestión de tareas y seguimiento: GitHub Projects (Scrumban)

- Despliegue del prototipo: GitHub Pages

Enlaces relevantes:

- Repositorio: https://github.com/Louis-Du/RedSocialSENA
- GitHub Pages (prototipo): https://louis-du.github.io/RedSocialSENA/
- Tablero Scrumban: https://github.com/users/Louis-Du/projects/6
- Issues del proyecto: https://github.com/Louis-Du/RedSocialSENA/issues

Estrategia de Control de Versiones

Para mantener el orden y la trazabilidad, se establece la siguiente convención de GitHub Flow que utilizará nuestro repositorio como estrategia para la gestión de ramas, estas serán estructuradas de la siguiente manera:

Tipo de rama	Rama	Descripción
Rama Principal	main	Contiene la versión estable del proyecto aprobada para evaluación y despliegue del prototipo.
Ramas de desarrollo de funcionalidades	feature/RF-XX-descripcion	Se utilizan para implementar nuevas funcionalidades asociadas a requisitos funcionales.
Ramas de Soporte	test	Rama para pruebas experimentales
Correción de errores	bugfix/issue-XX-descripcion	Destinadas a la corrección de defectos identificados durante pruebas o revisiones.
Documentos	docs/descripcion	Para actualizaciones o correcciones de documentación.

Convención de Versionamiento para commits

Todos los cambios deben registrarse siguiendo la convención Conventional Commits, con el objetivo de mantener claridad, trazabilidad y consistencia:

Formato:
tipo(alcance): descripción breve
[cuerpo opcional con detalles]
[referencia a issue: Closes #número]

Tipo de commits		
Tipo	Descripción	Ejemplo
feat	Nueva funcionalidad	feat(publicaciones): agregar formulario de creación de posts
fix	Corrección de error	fix(login): corregir validación de credenciales
docs	Cambios en documentación	docs(sqa): actualizar plan de calidad con nuevos criterios
style	Cambios de formato (CSS, espacios, etc.)	style(css): mejorar diseño responsive del muro principal
refactor	Refactorización sin cambiar funcionalidad	refactor(alcance): descripción breve
test	Agregar o modificar pruebas	test(mensajeria): agregar pruebas unitarias para chat
chore	Tareas de mantenimiento	chore(deps): actualizar dependencias de desarrollo

Control de Cambios

Todo cambio en el proyecto debe seguir el siguiente flujo controlado:

1. **Identificación del cambio**
    - Requisito funcional (RF) o issue existente
    - Solicitud de mejora o corrección
2. **Registro y planificación**
    - Creación o actualización de un issue en GitHub
    - Asignación de responsable
    - Etiquetado y ubicación en el tablero Scrumban
3. **Creación de rama**
    - Basada en main
    - Nombrada según la convención establecida
4. **Implementación**
    - Desarrollo del cambio
    - Commits frecuentes y descriptivos
5. **Pull Request**
    - Envío de la rama a GitHub
    - Asociación con el issue correspondiente
    - Uso del template de Pull Request
6. **Revisión y validación**
    - Revisión por el equipo
    - Verificación de criterios de aceptación
    - Pruebas funcionales básicas
7. **Merge y cierre**
    - Integración a main

- Cierre automático del issue
- Eliminación de la rama

8. **Despliegue**
    - Actualización del prototipo en GitHub Pages
    - Validación post-despliegue
    - Actualización de la línea base del proyecto

Criterios de Aceptación para Merge

Un Pull Request puede ser mergeado a main solo si cumple:

1. Todos los criterios de aceptación del issue están cumplidos
2. El código está libre de errores sintácticos
3. No introduce regresiones en funcionalidades existentes
4. La documentación está actualizada (si aplica)
5. Al menos 1 aprobación de revisión de código
6. El issue vinculado está correctamente referenciado

Trazabilidad de Cambios

La trazabilidad del proyecto se garantiza mediante la relación directa entre:

No se permite ningún cambio que no esté debidamente registrado y trazado dentro del repositorio del proyecto.

Línea Base del Proyecto

La línea base del proyecto está conformada por:

- Requisitos funcionales y no funcionales aprobados

- Prototipo funcional desplegado en GitHub Pages
- Documentación SQA validada
- Versiones estables del proyecto almacenadas en la rama main

Cualquier modificación a la línea base deberá seguir el proceso formal de control de cambios definido en este plan, garantizando trazabilidad, revisión y validación previa.

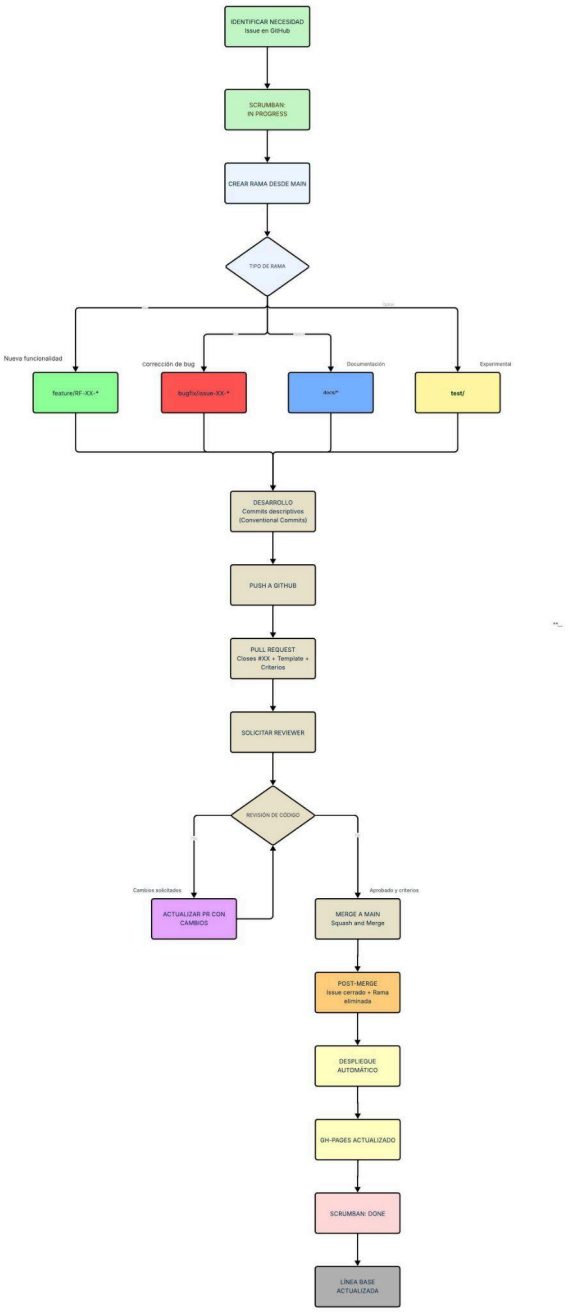
A continuación se presenta una evidencia de como se encuentra estructurada la línea base del repositorio, para esto se clonó el repositorio localmente y se usó el comando tree:

```
luisalbertoduenas@fedora:~/RedSocialSena$ tree
.
├── documents
│   ├── academico
│   │   ├── PosterTecnova2025.pdf
│   │   └── Proyecto Red Social para Aprendices del SENA (justificación).pdf
│   ├── pruebas
│   │   └── README.md
│   ├── requisitos
│   │   └── historias_usuario_sena_mejorada.docx
│   └── sqa
│       ├── Evaluación del comportamiento... - Listas de chequeo correccion.pdf
│       ├── Informe sobre la lista de chequeo y evaluación.pdf
│       ├── Plan de SQA.pdf
│       └── README.md
├── prototipo
│   ├── assets
│   │   ├── css
│   │   │   └── styles.css
│   │   ├── firma-digital-sena.png
│   │   ├── js
│   │   │   └── lucide.min.js
│   │   ├── logo-sena-blanco.png
│   │   ├── logo-sena-verde.png
│   │   └── noticias
```

```
│   │       ├── noticia1.png
│   │       ├── noticia1.webp
│   │       ├── noticia2.png
│   │       └── noticia2.webp
│   ├── index.html
│   ├── README.md
│   └── scripts
│       └── convert_images.py
└── README.md
```

Flujo principal

A continuación se presenta un diagrama que representa el flujo principal que utilizará el sistema:



Referencias

Fing, Universidad de la República. (2009). SQA PLA Grupo 5 v1.1. https://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/dvd02/experiencia2009/material/grupo5/sqa/inicial/iter1/SQAPLAG5v1.1.pdf

Louis-Du. (s.f.). Project board: Red Social SENA [Repositorio y tablero de proyecto]. GitHub. https://github.com/users/Louis-Du/projects/6/views/1

Louis-Du. (s.f.). RedSocialSENA [Repositorio]. GitHub. https://github.com/Louis-Du/RedSocialSENA

Servicio Nacional de Aprendizaje – SENA. (2024). Manual de identidad corporativa 2024.