

Especificación de Requisitos de Software (SRS)

Versión: 1.0

Fecha: 12/06/2025

Proyecto: FOURTECH – Sistema de Gestión para Cooperativas de Vivienda

Autores: Grupo FOURTECH (ESI Buceo – UTU)

1. Introducción

1.1 Propósito

El propósito principal de este documento es describir de manera exhaustiva y detallada los requisitos funcionales y no funcionales del sistema de software que será desarrollado por FOURTECH. Este SRS está concebido como un documento de referencia fundamental para todas las partes interesadas en el proyecto, incluyendo a los desarrolladores, diseñadores, docentes, coordinadores de proyecto y el cliente final, que en este caso es la cooperativa de vivienda. Su objetivo es asegurar una comprensión clara y compartida de las capacidades y limitaciones del sistema, lo que es esencial para un desarrollo, prueba y despliegue efectivos.

La inclusión explícita de "docentes" y "coordinadores" como audiencia objetivo subraya la naturaleza dual de este proyecto: no es solamente un esfuerzo de desarrollo de software con fines comerciales, sino también un ejercicio académico integral. Esta particularidad implica que el SRS debe ser no solo técnicamente preciso para los desarrolladores, sino también suficientemente claro y comprensible para la evaluación académica y la supervisión de la gestión del proyecto. Esta necesidad de claridad y exhaustividad se extiende a la trazabilidad de los requisitos, las definiciones precisas y la provisión de texto explicativo más allá de lo que se encontraría en un SRS puramente centrado en la industria. La relevancia del "Documento 'Proyecto de Pasaje de Grado 2025' – UTU" como una referencia clave refuerza esta perspectiva

1.2 Ámbito del Sistema

El sistema FourTech está diseñado para gestionar procesos críticos dentro de las cooperativas de vivienda de ayuda mutua. Su alcance abarca la totalidad del ciclo de interacción con los miembros, desde la postulación inicial hasta la participación continua y las contribuciones financieras. Las funcionalidades incluidas en el sistema son: el registro de usuarios y la gestión de postulaciones; la validación y aprobación de las mismas; la administración de aportes iniciales y pagos mensuales, incluyendo la carga y validación de comprobantes; el seguimiento y la validación de las horas de trabajo de los miembros

(autoconstrucción); la asignación y el mantenimiento del estado de las unidades habitacionales; funcionalidades de backoffice completas para los administradores, que incluyen la gestión de usuarios, la aprobación de pagos y aportes, el monitoreo de horas de trabajo y la asignación de unidades; y la generación de diversos reportes (económicos, de participación y de avance de obra).

Es importante destacar las funcionalidades que están explícitamente excluidas del alcance del sistema para esta fase del proyecto. Estas son: la gestión de obras de construcción física, la elaboración de presupuestos técnicos de construcción, y la implementación de funcionalidades de chat en tiempo real. Esta delimitación clara del alcance representa una decisión estratégica para la gestión del proyecto. Dada la naturaleza académica del proyecto, con un "Presupuesto inicial:80\$ mil dólares" y un tiempo de desarrollo estimado de "unas 120-150 horas", establecer límites estrictos previene la expansión descontrolada del alcance (scope creep). Esto permite que el equipo concentre sus esfuerzos de desarrollo en las funcionalidades centrales que aportan el mayor valor en términos de automatización administrativa, asegurando que el proyecto sea viable dentro de las restricciones de tiempo y recursos disponibles.

1.3 Definiciones, Acrónimos y Abreviaturas

SRS (Software Requirements Specification): Documento que define todos los requisitos funcionales y no funcionales del sistema a desarrollar.

RF (Requisito Funcional): Una función o comportamiento específico que el sistema debe realizar.

RNF (Requisito No Funcional): Una característica de calidad o restricción que el sistema debe satisfacer, como el rendimiento, la seguridad o la usabilidad.

Backoffice: Plataforma de gestión interna del sistema, diseñada principalmente para presidentes y administradores de cooperativas.

Cooperativista: Usuario final del sistema que es miembro aspirante o activo de una cooperativa de vivienda.

Administrador: Usuario con privilegios elevados para supervisar todo el sistema, gestionar las operaciones de la cooperativa y generar reportes globales.

Docker: Plataforma para desarrollar, distribuir y ejecutar aplicaciones en contenedores ligeros y portátiles.

Rocky Linux: Distribución de Linux de nivel empresarial, de código abierto, utilizada como sistema operativo base para los servidores del sistema.

API (Application Programming Interface): Conjunto de funciones y protocolos que permiten la comunicación entre diferentes componentes de software (por ejemplo, frontend y backend).

Landing Page: Página web inicial de presentación pública del sistema, utilizada para la difusión de información y el registro/postulación de nuevos usuarios.

ANV (Agencia Nacional de Vivienda): Organismo gubernamental uruguayo encargado de políticas de vivienda.

INACOOP (Instituto Nacional del Cooperativismo): Instituto nacional uruguayo que promueve y regula el sector cooperativo.

MVOTMA (Ministerio de Vivienda, Ordenamiento Territorial y Medio Ambiente): Ministerio uruguayo con competencias en vivienda y medio ambiente.

MIDES (Ministerio de Desarrollo Social): Ministerio uruguayo encargado de políticas sociales.

BPS (Banco de Previsión Social): Organismo de seguridad social en Uruguay.

DGI (Dirección General Impositiva): Autoridad tributaria en Uruguay.

BSE (Banco de Seguros del Estado): Compañía de seguros estatal en Uruguay.

UTU (Universidad del Trabajo del Uruguay): Institución de educación técnica en Uruguay.

IAT (Instituto de Asistencia Técnica): Entidad que brinda apoyo técnico a cooperativas de vivienda.

UI (Unidades Indexadas): Unidad de cuenta en Uruguay, ajustada por inflación.

SQLi (SQL Injection): Técnica de inyección de código utilizada para atacar aplicaciones basadas en datos.

XSS (Cross-Site Scripting): Tipo de vulnerabilidad de seguridad que se encuentra comúnmente en aplicaciones web.

MVC (Model-View-Controller): Patrón de diseño de software utilizado para desarrollar interfaces de usuario.

1.4 Referencias

Estándar IEEE 830-1998

Ley N° 18.331 de Protección de Datos Personales (Uruguay)

Documento 'Proyecto de Pasaje de Grado 2025' – UTU

1.5 Visión General del Documento

Sección 1: Introducción – Proporciona información fundamental sobre el documento y el proyecto.

Sección 2: Descripción General – Ofrece una visión de alto nivel del producto, su contexto, las características de sus usuarios y las restricciones generales.

Sección 3: Requisitos Específicos – Detalla los requisitos funcionales y no funcionales precisos que el sistema debe cumplir.

2. Descripción General

Esta sección ofrece una visión de alto nivel del sistema FourTech, incluyendo su contexto dentro del proyecto más amplio, sus principales funcionalidades, las características de sus usuarios previstos y cualquier restricción o suposición general que influya en su diseño y desarrollo.

2.1 Perspectiva del Producto

El sistema FourTech se concibe como una aplicación web modular, constituyendo un componente crucial de un proyecto académico interdisciplinario en la UTU. Está diseñado para optimizar los procesos administrativos y participativos de las cooperativas de vivienda de ayuda mutua. El sistema se compone de varios módulos clave: una

Landing Page informativa pública que proporciona información general y un portal para nuevas postulaciones; un **Portal del Cooperativista** seguro y personalizado para que los miembros aceptados gestionen sus contribuciones y sigan su estado; un **Backoffice para Presidentes/Administradores**, que es un panel administrativo para el liderazgo de la cooperativa para supervisar las operaciones, validar datos y generar informes; y una **API RESTful en PHP** que sirve como infraestructura de backend, facilitando la comunicación entre los módulos frontend y la base de datos.

La estrategia de despliegue del sistema implica el uso de contenedores Docker en servidores escolares, lo que indica un enfoque moderno y contenedorizado para la implementación. La elección de un "sistema web modular" y "APIs separadas", junto con el despliegue con Docker, sugiere una filosofía de diseño que prioriza la escalabilidad y la mantenibilidad. Al separar las responsabilidades en módulos distintos y utilizar contenedores, el sistema puede ser desarrollado, probado, desplegado y actualizado con mayor facilidad. Esto apoya directamente los requisitos no funcionales de "Escalabilidad" y "Mantenibilidad", lo que demuestra una arquitectura con visión de futuro para un proyecto que podría evolucionar más allá de su alcance académico inicial.

2.2 Funciones del Producto

A un alto nivel, el sistema FourTech proporciona funcionalidades distintas adaptadas a sus roles de usuario principales:

- **Para Cooperativistas (Usuarios):** El sistema permitirá el registro y envío de postulaciones, la carga de comprobantes de aportes iniciales y pagos mensuales, la consulta del estado de su postulación y su historial de pagos, y el registro de sus horas de trabajo.
- **Para Presidentes (Liderazgo de la Cooperativa):** El sistema les permitirá validar y aprobar o rechazar nuevas postulaciones, aprobar o rechazar comprobantes de pagos y aportes, validar las horas de trabajo de los miembros, y asignar unidades habitacionales.
- **Para Administradores (Nivel de Sistema):** El sistema les facilitará la generación de informes globales (económicos, de participación, etc.) y la supervisión integral de todos los datos y procesos del sistema.

Estas funciones de alto nivel abordan directamente el "Problema identificado", donde "La administración manual provoca errores, demoras y falta de claridad". Las funciones del sistema, como la "validación de postulaciones, pagos y horas" para los presidentes y el "registro, carga de comprobantes" para los usuarios, son respuestas directas a las inefficiencias de los procesos manuales actuales. Esto se confirma en la entrevista con el presidente de "La Fragua", quien menciona el uso de "Planillas Excel" y las dificultades para controlar los pagos. Esta alineación entre el problema y la solución propuesta demuestra una clara adecuación a las necesidades del usuario.

2.3 Características de los Usuarios

Los usuarios esperados tienen conocimientos básicos de informática. Se espera que accedan desde navegadores modernos en PC.

Perfil de Usuario	Nombre	Rol	Conocimiento Tecnológico	Objetivos de Uso del Sistema	Frustraciones
Cooperativista	Sofía Rodríguez	Miembro Aspirante /Activo	Básico (redes sociales, correo electrónico, banca en línea)	Seguir estado de postulación, Subir comprobantes de pago, Consultar historial de pagos, Acceder a información de la cooperativa	Olvidar fechas de pago, No comprender estado de postulación, Dificultades técnicas
Administrador	Marcelo Pereira	Miembro Comisión Administrativa	Básico-intermedio (Excel, correo electrónico, WhatsApp)	Validar solicitudes/documentos, Aprobar pagos, Asignar viviendas, Generar informes, Consultar estado de miembros	Recuperación ineficiente de información, Errores por problemas de validación, Falta de automatización

Objetivos de Uso del Sistema:

- Validar solicitudes y documentos con precisión.
- Aprobar pagos de manera eficiente.
- Asignar unidades habitacionales de forma sistemática.
- Generar diversos informes para la toma de decisiones.
- Consultar el estado de los miembros de manera exhaustiva.

Frustraciones:

- Recuperación ineficiente de información, lo que apunta a la necesidad de potentes funciones de búsqueda y filtrado.
- Errores debido a problemas de validación, lo que requiere mecanismos de validación robustos.
- Falta de automatización en tareas repetitivas, lo que enfatiza la necesidad de automatización de flujos de trabajo.

2.4 Restricciones

Limitaciones de Alcance: Como se definió en la Sección 1.2, el sistema excluye explícitamente la gestión de obras de construcción física, los presupuestos técnicos de construcción y las funcionalidades de chat en tiempo real.

Pila Tecnológica: Backend en PHP 8, base de datos en MySQL 8, cumplimiento de la Ley 18.331, versionado con Git y GitHub, y soporte mínimo de 3 servidores.

Entorno de Despliegue: El sistema debe ser implementable con contenedores Docker sobre Linux Rocky 8

Contexto del Proyecto Académico: El proyecto opera dentro de las limitaciones de un cronograma académico y la disponibilidad de recursos.

2.5 Suposiciones y Dependencias

Suposiciones:

Alfabetización Digital del Usuario: Se asume que los miembros de la cooperativa y los administradores poseen al menos un nivel básico de alfabetización digital, suficiente para interactuar con un sistema basado en la web.

Conectividad a Internet: Se asume que los usuarios (tanto cooperativistas como administradores) tienen acceso a internet confiable para utilizar el sistema basado en la web.

Cumplimiento de la Cooperativa: Se asume que la cooperativa misma cumple con los marcos legales y regulatorios que rigen las cooperativas de vivienda de ayuda mutua en Uruguay (por ejemplo, regulaciones de ANV, INACOOP, MVOTMA, Ley 18.407). El sistema apoyará, pero no impondrá, todos los aspectos de estas regulaciones.

Disponibilidad de Datos: Se asume que los datos necesarios para la inicialización del sistema (por ejemplo, listas de miembros existentes, detalles de unidades habitacionales) se proporcionarán en un formato estructurado.

Aceptación de Procesos Digitales: Se asume que los miembros y el liderazgo de la cooperativa están dispuestos a hacer la transición de procesos manuales (por ejemplo, "Planillas Excel") a procesos digitales para las tareas administrativas.

Dependencias:

- **Infraestructura:** El sistema depende de la disponibilidad y estabilidad de la infraestructura de servidores de la institución educativa para el despliegue de Docker.
- **Servicios Externos (Futuro):** La futura integración con servicios de correo electrónico para notificaciones depende de la disponibilidad y configuración de dichos servicios.
- **Actualizaciones Regulatorias:** Las características de cumplimiento del sistema dependen de actualizaciones oportunas con respecto a los cambios en las leyes cooperativas y de protección de datos de Uruguay.

2.6 Requisitos Futuros

Expansión del Servicio de Notificaciones: Ya está planificada la integración con un sistema integral de notificación por correo electrónico como una interfaz futura. Esto podría extenderse a notificaciones por SMS, como lo desea el presidente de la cooperativa.

3. Requisitos Específicos

Esta sección detalla los requisitos funcionales y no funcionales específicos del sistema FourTech.

3.1 Interfaces Externas

3.1.1 Interfaz Web del Usuario (Frontend):

- Será accesible a través de navegadores web estándar en dispositivos móviles y de escritorio.
- Utilizará HTML, CSS y JavaScript para la interacción del usuario y la presentación de contenido dinámico.
- Estará diseñado para ser responsive, asegurando una visualización óptima en diversas pantallas.

3.1.2 Interfaz de Administración (Backoffice):

- Será una plataforma web exclusiva para los presidentes y administradores de la cooperativa.

- Requerirá autenticación de inicio de sesión segura para acceder a las funcionalidades administrativas.
- Proporcionará herramientas para gestionar postulaciones, pagos, horas de trabajo y asignaciones de unidades habitacionales.

3.1.3 APIs REST:

- Facilitarán la comunicación entre las interfaces frontend (de usuario y de administración) y el sistema backend.
- Serán desarrolladas en PHP 8, proporcionando una forma estructurada para el intercambio de datos.
- Incluirán puntos finales específicos para autenticación, gestión de usuarios, manejo de comprobantes (pagos, aportes), generación de informes y otras operaciones del sistema.

3.1.4 Base de Datos:

- Se utilizará MySQL 8 como sistema de gestión de bases de datos relacionales.
- El esquema de la base de datos seguirá una estructura normalizada (específicamente la 3ra. Forma Normal) para garantizar la integridad y eficiencia de los datos.

3.1.5 Servicio de Notificaciones (Futuro):

- Se planifica una futura integración con un sistema de correo electrónico para notificaciones automatizadas.
- Existe el potencial de expansión a notificaciones por SMS, según el deseo del presidente de la cooperativa.

3.2 Funciones

- RF-01: El usuario podrá registrarse mediante un formulario web en la Landing Page.
- RF-02: El usuario podrá postularse para ingresar a una cooperativa.
- RF-03: El presidente podrá revisar las postulaciones de su cooperativa.
- RF-04: El presidente podrá aceptar o rechazar nuevas postulaciones.
- RF-05: El usuario podrá cargar el comprobante de aporte inicial.
- RF-06: El presidente podrá validar el comprobante de aporte inicial.
- RF-07: Una vez validado el aporte inicial, el sistema permitirá asignar una unidad habitacional al usuario.
- RF-08: El usuario podrá registrar sus horas de trabajo semanales.

- RF-09: El usuario podrá justificar las horas no cumplidas y, en caso necesario, registrar un pago compensatorio.
- RF-10: El sistema calculará automáticamente si el usuario cumple con las 21 horas semanales requeridas.
- RF-11: El usuario podrá cargar comprobantes de pagos mensuales.
- RF-12: El usuario podrá consultar su estado de pagos y horas (al día o en atraso).
- RF-13: El sistema notificará al usuario su estado (aceptado, pendiente, rechazado, al día o en atraso).
- RF-14: El backoffice permitirá al presidente gestionar las viviendas, las postulaciones, los comprobantes y los registros de horas.
- RF-15: El administrador global podrá emitir reportes de pagos, horas de trabajo y estado general de la cooperativa.

3.3 Requisitos de Rendimiento

- RR-01:Tiempo de Respuesta: Las interacciones del usuario (por ejemplo, envíos de formularios, cargas de página) deberían completarse idealmente en 3 segundos.
- RR-02: Debe soportar al menos 100 usuarios simultáneos en la red escolar.(resultado aproximado)
- RR-03:Capacidad inicial para 500 usuarios cooperativistas con sus registros asociados(resultado aproximado).

3.4 Restricciones de Diseño

- RD-01:Arquitectura MVC obligatoria para todas las aplicaciones.
- RD-02:Backend implementado exclusivamente en PHP 8.
- RD-03:Base de datos MySQL 8.
- RD-04:Versionado del código con Git en repositorios de GitHub.
- RD-05:Implementación con contenedores Docker sobre Linux Rocky 8.

3.5 Atributos del Sistema

- AS-01:Disponibilidad: 99% durante horario escolar.
- AS-02:Seguridad: Control de acceso con autenticación, protección contra SQLi y XSS.
- AS-03:Escalabilidad: Diseño modular con APIs separadas.
- AS-04:Usabilidad: Interfaz amigable para usuarios con poca experiencia digital.
- AS-05:Mantenibilidad: Código estructurado y documentado.

3.6 Otros Requisitos

- OR-01:El sistema debe tener interfaz en español.
- OR-02:Manuales de usuario diferenciados por perfil (cooperativista, presidente, administrador).
- OR-03:Documentación técnica de instalación y despliegue con Docker.
- OR-04:- La interfaz del sistema debe estar disponible en español como inglés.

4. Apéndices

- Diagramas de casos de uso y arquitectura del sistema.
- Ejemplos de formularios de registro, carga de comprobantes y reportes.
- Esquema inicial de base de datos (estructura en SQL).
- Referencias legales y técnicas: IEEE 830, Ley 18.331, requerimientos UTU.