| Descripción: Descripción: https://moodle.lab.dit.upm.es/moodle/pluginfile.php/4552/mod_resource/content/1/ditupm.gif | **Departamento de Ingeniería de Sistemas Telemáticos**  **Ingeniería de Sistemas y Servicios Telemáticos** |
| --- | --- |

**DOCUMENTO DE VISIÓN**

**VD (Vision Document)**

| **Nombre del documento** | Documento de visión |
| --- | --- |
| **Nombre del fichero:** | ISST-VD-V12.docx |
| **Versión:** | 02 |
| **Fecha de entrega:** | 03/03/2025 |
| **Caso de estudio:** | ISST-Grupo12-Split.it |
| **Entrega:** | Sprint 1 |

| **Autores/as:** | Pablo Bas Iglesias y Santiago Rayán Castro |
| --- | --- |
| **Revisor/a:** | Javier de Ponte Hernando |

**INDICE**

[**1**](#_heading=h.gjdgxs) **INTRODUCCIÓN 2**

[1.1](#_heading=h.30j0zll) Propósito del documento 2

[1.2](#_heading=h.3znysh7) Definiciones, acrónimos y abreviaturas 2

[1.3](#_heading=h.2et92p0) Alcance del proyecto 2

[1.4](#_heading=h.3dy6vkm) Materiales de referencia 3

[1.5](#_heading=h.2s8eyo1) Estructura del documento 3

[**2**](#_heading=h.3rdcrjn) **DESCRIPCIÓN GENERAL DEL SISTEMA 3**

[**3**](#_heading=h.lnxbz9) **DESCRIPCIÓN DE CASOS DE USO 4**

[3.1](#_heading=h.1ksv4uv) Acceder al sistema 5

[3.2](#_heading=h.44sinio) Registrar un gasto compartido 5

[3.3](#_heading=h.2jxsxqh) Consultar saldo y deudas del grupo 6

[3.4](#_heading=h.z337ya) Enviar recordatorios de pago 7

[3.5](#_heading=h.3j2qqm3) Administrar grupos 7

[3.6](#_heading=h.4i7ojhp) Enviar sugerencias al equipo desarrollador 8

[**4**](#_heading=h.2xcytpi) **ESPECIFICACIÓN SUPLEMENTARIA 9**

[4.1](#_heading=h.1ci93xb) Reglas de dominio 9

[4.2](#_heading=h.2bn6wsx) Requisitos no funcionales 9

[4.3](#_heading=h.3as4poj) Restricciones 10

# INTRODUCCIÓN

## Propósito del documento

Este documento describe la visión general del sistema Split.it, una plataforma diseñada para la gestión de gastos compartidos entre amigos, familiares o compañeros de piso. Su propósito es proporcionar un marco inicial sobre los intereses, objetivos y requisitos clave del sistema en desarrollo, permitiendo establecer una base sólida para su implementación y futuras mejoras.

## Definiciones, acrónimos y abreviaturas

**Autenticación**: Proceso que verifica la identidad de los usuarios antes de permitirles acceder al sistema.

**Kanban**: Método visual de gestión de tareas basado en tableros que permite optimizar el flujo de trabajo

**MVP (Minimum Viable Product)**: Producto Mínimo Viable con funcionalidades esenciales para la validación del concepto.

**Product Backlog**: Lista priorizada de funcionalidades y tareas pendientes del sistema, administrada por el Product Owner.

**Scrum**: Metodología ágil utilizada en el proyecto para gestionar el desarrollo de software de manera iterativa e incremental.

**Split.it:** Sistema de gestión de gastos compartidos que permite a los usuarios registrar y dividir gastos.

**Sprint**: Ciclo de desarrollo en la metodología Scrum, generalmente con una duración de 1 a 4 semanas, en el que se desarrolla un conjunto de funcionalidades.

**Stakeholders**: Usuarios y entidades interesadas en el desarrollo y funcionamiento del sistema.

## Alcance del proyecto

**Split.it** es una aplicación web diseñada para facilitar la gestión de gastos compartidos en grupos, proporcionando a los usuarios una plataforma intuitiva para organizar pagos y deudas de manera eficiente.

El **MVP** incluirá las siguientes funcionalidades esenciales:

* **Creación de grupos de gastos** donde los usuarios podrán administrar múltiples cuentas compartidas.
* **Registro de gastos** con la opción de dividirlos equitativamente o asignarlos a miembros específicos.
* **Cálculo automático de deudas y pagos pendientes** entre los participantes del grupo.
* **Funciones avanzadas** que incluyen la subida de imágenes de tickets y el envío de recordatorios a los usuarios con deudas pendientes.

## Materiales de referencia

Los siguientes documentos y herramientas han sido utilizados como referencia para la elaboración de este documento:

* Documentación oficial de Scrum.
* Diapositivas disponibles en el Moodle de ISST referidas a los temas 1 (Ingeniería de los productos del software) y 2 (Ingeniería de requisitos).
* Análisis de aplicaciones similares en el mercado para evaluar mejores prácticas en la gestión de gastos compartidos.
* K.S. Rubin. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison Wesley, 2012. Para historias de usuario y metodología Scrum.

## Estructura del documento

Este documento se organiza en las siguientes secciones:

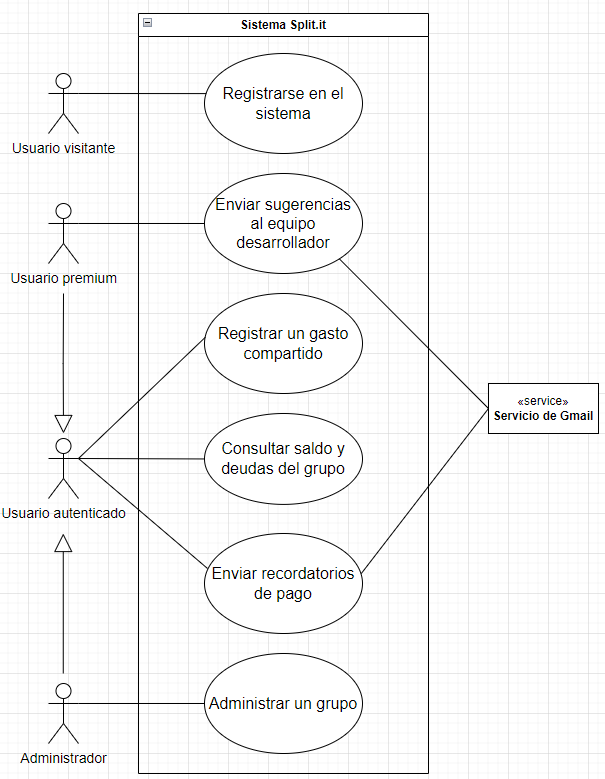
* **Introducción**: Presenta el propósito, alcance y contexto del sistema.
* **Descripción general del sistema**: Explica la funcionalidad principal y los actores involucrados.
* **Casos de uso**: Define los escenarios principales de interacción con el sistema, incluyendo la creación de grupos de gastos, el registro de gastos, el cálculo de saldos y el envío de recordatorios.
* **Especificaciones suplementarias**: Incluye reglas de negocio, requisitos no funcionales y restricciones del sistema.

# descripción general del sistema

**Split.it** es una aplicación web que facilita la gestión de gastos compartidos en grupos, permitiendo a los usuarios registrar pagos, dividirlos entre los participantes y calcular deudas de manera automática. El sistema proporciona un balance actualizado en tiempo real y funcionalidades avanzadas como la subida de tickets y el envío de recordatorios de pago. Los usuarios podrán crear grupos, añadir gastos, consultar saldos y gestionar pagos pendientes, mientras que la plataforma garantizará el correcto almacenamiento y procesamiento de los datos, asegurando una experiencia eficiente y transparente.

# descripción de casos de uso

En este apartado se describen los principales casos de uso del sistema **Split.it**, los cuales representan las interacciones esenciales que los usuarios podrán realizar. Estos casos de uso reflejan las funcionalidades principales del sistema y constituyen la base del **MVP (Minimum Viable Product)**.

****

*Imagen 1: Diagrama de casos de uso de Split.it*

## Registrarse en el sistema

**Descripción**: Permite a un usuario añadir un gasto en un grupo, indicando el monto total, quién lo pagó y cómo se divide entre los participantes.

* **Actores principales**: Usuario sin autenticar.
* **Actores secundarios**: Servicio de autenticación: Auth0.
* **Evento inicial**: Un usuario entra al sistema después de iniciar sesión o crear una cuenta.
* **Flujo principal**:
  1. El usuario crea una cuenta con su nombre y correo electrónico.
  2. El usuario inicia sesión con dicha cuenta.
* **Extensiones**:
  + Si el usuario olvida la contraseña tiene la opción de cambiarla a través del correo electrónico.
* **Casos de error**:
  + El usuario intenta ingresar un correo que no sigue el formato correo@extension.com.
  + El usuario deja alguno de los campos vacíos.
  + El usuario intenta iniciar sesión con una combinación de correo y contraseña que no coincide con la de ninguna cuenta existente.
* **Historias de usuario asociadas:**
  + Yo como usuario no registrado quiero poder crearme una cuenta en el sistema utilizando mi nombre y mi correo electrónico.
  + Yo como usuario registrado en el sistema pero sin autenticar quiero poder iniciar sesión con mi correo y contraseña.
  + Yo como usuario registrado en el sistema pero sin autenticar quiero poder cambiar mi contraseña en caso de que se me olvide a través de mi correo electrónico.
  + Yo como usuario premium quiero acceder a la aplicación libre de anuncios.

## Registrar un gasto compartido

**Descripción**: Permite a un usuario añadir un gasto en un grupo, indicando el monto total, quién lo pagó y cómo se divide entre los participantes.

* **Actores principales**: Usuario registrado.
* **Actores secundarios**: Otros miembros del grupo.
* **Evento inicial**: El usuario selecciona un grupo y elige la opción "Añadir gasto".
* **Flujo principal**:
  1. El usuario ingresa el monto del gasto.
  2. Selecciona quién realizó el pago y los participantes involucrados.
  3. Guarda el gasto en el sistema.
  4. Split.it calcula automáticamente cuánto debe pagar cada usuario.
* **Extensiones**:
  + El usuario puede subir una imagen del ticket/recibo asociado al gasto.
  + Se permite asignar el gasto solo a algunos miembros del grupo, en lugar de dividirlo equitativamente.
* **Casos de error**:
  + El usuario intenta ingresar un gasto sin monto válido.
  + Se intenta agregar un gasto sin seleccionar participantes.
* **Historias de usuario asociadas:**
  + Yo, como usuario autenticado, quiero poder añadir un gasto indicando, el monto del gasto, quién realizó el gasto, qué otros participantes están involucrados en el gasto, en un grupo del que soy participante para actualizar los balances del grupo.
  + Yo, como usuario que ha añadido un gasto al grupo, quiero poder editarlo para añadir o eliminar participantes al gasto.
  + Yo, como usuario autenticado, quiero poder subir una imagen del ticket/recibo asociado al gasto en un grupo del que soy participante.

## Consultar saldo y deudas del grupo

**Descripción**: Permite a los usuarios visualizar un resumen de los gastos y conocer el saldo actualizado del grupo.

* **Actores principales**: Usuario registrado.
* **Actores secundarios**: Otros miembros del grupo.
* **Evento inicial**: El usuario accede al grupo y selecciona la opción "Ver saldo".
* **Flujo principal**:
  1. El sistema muestra el resumen de los gastos del grupo.
  2. Se listan los saldos individuales, indicando quién debe cuánto y a quién.
* **Extensiones**:
  + Posibilidad de filtrar los gastos por fecha o categoría.
* **Casos de error**:
  + No hay gastos registrados en el grupo, por lo que el saldo aparece en cero.
* **Historias de usuario asociadas:**
  + Yo, como usuario autenticado, quiero poder añadir un gasto indicando, el monto del gasto, quién realizó el gasto, qué otros participantes están involucrados en el gasto, en un grupo del que soy participante para actualizar los balances del grupo.
  + Yo, como usuario que ha añadido un gasto al grupo, quiero poder editarlo para añadir o eliminar participantes al gasto.
  + Yo, como usuario registrado, quiero poder subir una imagen del ticket/recibo asociado al gasto en un grupo del que soy participante.

## Enviar recordatorios de pago

**Descripción**: Permite a los usuarios enviar notificaciones a los miembros del grupo con deudas pendientes.

* **Actores principales**: Usuario registrado.
* **Actores secundarios**: Otros miembros del grupo.
* **Evento inicial**: Un usuario con saldo a favor selecciona la opción "Enviar recordatorio".
* **Flujo principal**:
  1. El usuario elige a uno o más miembros con deudas pendientes.
  2. El sistema envía una notificación con el monto a pagar.
* **Extensiones**:
  + Opción para personalizar el mensaje del recordatorio.
* **Casos de error**:
  + No hay deudas pendientes en el grupo, por lo que no se puede enviar el recordatorio.
* **Historias de usuario asociadas:**
  + Yo como usuario registrado quiero poder enviarle un recordatorio predeterminado a través de correo/WhatsApp a los usuarios que me deben dinero de un grupo en el que soy participante y tengo saldo positivo.
  + Yo como usuario registrado quiero poder editar el mensaje del recordatorio a través de correo/WhatsApp a los usuarios que me deben dinero de un grupo en el que soy participante y tengo saldo positivo.

## Administrar un grupo

**Descripción**: Permite al usuario que ha creado un grupo gestionar sus participantes, gastos y permisos.

* **Actores principales**: Administrador.
* **Actores secundarios**: Otros miembros del grupo.
* **Evento inicial**: Un usuario selecciona la opción de “crear grupo”.
* **Flujo principal**:
  1. El usuario crea un grupo, convirtiéndose en administrador de este.
  2. El usuario, ahora administrador del grupo, añade a los participantes que desee al grupo.
  3. El administrador añade o elimina gastos al grupo.
  4. El administrador cierra el grupo.
* **Extensiones**:
  + El usuario, ahora administrador del grupo, puede editar los permisos del grupo.
* **Casos de error**:
  + El usuario intenta crear un grupo con un nombre vacío.
* **Historias de usuario asociadas:**
  + Yo como usuario quiero poder crear un grupo convirtiéndome en el administrador de este para posteriormente invitar a otros participantes.
  + Yo como administrador del grupo quiero poder añadir o eliminar participantes de un grupo existente.
  + Yo, como administrador quiero poder cerrar un grupo.
  + Yo, como administrador del grupo quiero poder controlar los permisos del grupo.
  + Yo, como administrador del grupo quiero poder añadir, editar y eliminar gastos de otros usuarios.

## Enviar sugerencias al equipo desarrollador

**Descripción**: Permite a los usuarios premium hacer peticiones de funcionalidades deseadas directamente al equipo desarrollador.

* **Actores principales**: Usuario premium.
* **Actores secundarios**: Servicio de Gmail.
* **Evento inicial**: Un usuario selecciona la opción de “Enviar sugerencias”.
* **Flujo principal**:
  1. El usuario premium, accede a la sección de enviar sugerencias.
  2. El usuario rellena los campos (título y mensaje) con la petición a realizar.
  3. Una vez rellenados los campos, el usuario le da a la opción de enviar.
  4. Se envía un correo electrónico a través de Gmail al equipo desarrollador.
* **Casos de error**:
  + El usuario intenta mandar una petición sin rellenar alguno de los campos.
* **Historias de usuario asociadas:**
  + Yo como usuario premium, quiero poder enviar sugerencias directamente al equipo desarrollador para poder mejorar la aplicación por la que estoy pagando. como usuario premium

# especificación suplementaria

## Reglas de dominio

Las siguientes reglas definen el comportamiento y las restricciones del sistema:

* **Políticas de la organización:**
  + Un usuario solo podrá registrar gastos en los grupos en los que esté registrado.
  + Los gastos solo pueden ser editados o eliminados por el usuario que los creó.
  + Cada gasto debe tener un importe mayor a 0 y debe estar asociado a al menos un miembro del grupo.
  + Los saldos de los miembros del grupo se actualizan automáticamente cada vez que se registra o edita un gasto.
  + Un usuario podrá enviar recordatorios de pago únicamente a los miembros del grupo que tengan una deuda pendiente.
* **Marco regulatorio:**
  + Al tratar con datos personales de los usuarios tendremos que seguir la normativa establecida por la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales y normativa en desarrollo. Dicha ley es la adaptación del RGPD al contexto español.

## Requisitos no funcionales

Los requisitos no funcionales garantizan aspectos como la seguridad, el rendimiento y la usabilidad del sistema:

* **Interfaz:**
  + El sistema tendrá que interactuar con los servicios de Gmail para poder mandar recordatorios de pago por correo electrónico.
* **Seguridad**:
  + Se implementarán controles de acceso y autenticación mediante JWT.
  + La comunicación entre el cliente y el servidor debe estar cifrada mediante HTTPS.
  + Solo los miembros de un grupo podrán acceder a sus gastos y balances.
* **Usabilidad**:
  + La aplicación será compatible con navegadores web Firefox y Chrome.
* **Mantenibilidad:**
  + La aplicación utiliza una arquitectura basada en REST API del lado del servidor.
* **Rendimiento**:
  + La aplicación se mantendrá estable y receptiva, con el 95% de las solicitudes de usuario para operaciones típicas completadas en un máximo de 2 segundos cuando sea accedida por hasta 50 usuarios concurrentes.

## Restricciones

Las siguientes restricciones definen los límites y consideraciones técnicas del desarrollo del sistema:

* **Tecnología**:
  + El sistema se implementará como una aplicación web, sin una versión móvil nativa en esta fase.
  + Se utilizarán bases de datos relacionales para almacenar la información de los usuarios y los gastos.
* **Alcance del MVP**:
  + En esta primera versión no se permitirá la integración con plataformas de pago.
  + No se incluirán funcionalidades avanzadas como categorización de gastos o reportes detallados.
* **Implementación:**
  + Capa de presentación: HTML y CSS mediante tecnologías de plantillas de servidor (Thymeleaf, por ejemplo).
  + Lógica de aplicación: Java con Spring Boot, implementando una REST API que se comunica con la base de datos a través de repositorios utilizando JPA integrado con Hibernate.
* **Interfaz:**
  + La funcionalidad de envío de correos depende de la conexión con Gmail.
* **Legales:**
  + Se llevará un registro de los datos recabados y sus finalidades (incluido en el repositorio del código), según lo especificado en la normativa mencionada en las reglas de dominio (Ley Orgánica 3/2018).