| 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 DISEÑO SOFTWARE**

**SDD (Software Design Document)**

| **Nombre del documento** | Documento de Diseño Software |
| --- | --- |
| **Nombre del fichero:** | ISST-SSD-V12.docx |
| **Versión:** | 1 |
| **Fecha de entrega:** | 03/03/2025 |
| **Caso de estudio:** | split.it |
| **Entrega:** | Sprint 1 |

| **Autores/as:** | Pablo Bas Iglesias, Rodrigo de la Nuez Moreno, Santiago Rayán Castro |
| --- | --- |
| **Revisor/a:** | Nicolas García Sobrino, Javier de Ponte Hernando |

**INDICE**

[**1**](#_heading=h.1fob9te) **INTRODUCCIÓN 2**

[**2**](#_heading=h.3znysh7) **ANÁLISIS DE LA ARQUITECTURA 2**

[2.1](#_heading=h.2et92p0) Requisitos y arquitectura 2

[2.2](#_heading=h.tyjcwt) Patrón de arquitectura seleccionado 2

[2.3](#_heading=h.3dy6vkm) Otros criterios 2

[**3**](#_heading=h.1t3h5sf) **DESCRIPCIÓN DE LA ARQUITECTURA 2**

[3.1](#_heading=h.4d34og8) Vista lógica de la arquitectura 2

[3.2](#_heading=h.2s8eyo1) Vista de despliegue de la arquitectura 2

[**4**](#_heading=h.17dp8vu) **MODELO DE DATOS 2**

[**5**](#_heading=h.3rdcrjn) **INTERFAZ DE USUARIO 3**

[5.1](#_heading=h.26in1rg) Estructura y contenido 3

[5.2](#_heading=h.lnxbz9) Flujos de navegación 4

# INTRODUCCIÓN

Este Software Design Document (SDD) tiene como finalidad describir el diseño técnico de Split.it, una aplicación web para la gestión de gastos compartidos. Se detallan la arquitectura del sistema, el modelo de datos y la interfaz de usuario.

La estructura del documento es de la siguiente forma:

**1. INTRODUCCIÓN:**Describe el propósito del documento, contextualiza el proyecto y resume brevemente los temas que se tratarán.

**2. ANÁLISIS DE LA ARQUITECTURA:**Presenta los requisitos clave (funcionales y no funcionales) que influyen en el diseño, y justifica la elección del patrón arquitectónico basado en criterios técnicos y de negocio.

**3. DESCRIPCIÓN DE LA ARQUITECTURA:**Expone la estructura del sistema desde dos perspectivas: la vista lógica, que muestra la organización interna y los componentes, y la vista de despliegue, que ilustra la distribución física en los diferentes nodos.

**4. MODELO DE DATOS:**Define el esquema de la base de datos identificando las principales entidades, sus atributos y relaciones, complementado con un diccionario de datos que especifica tipos y restricciones.

**5. INTERFAZ DE USUARIO:**Detalla el diseño visual e interactivo del sistema, describiendo la estructura de las pantallas y los flujos de navegación para asegurar una experiencia de usuario intuitiva.

# ANÁLISIS DE LA ARQUITECTURA

### 2.1 Requisitos y arquitectura

Los requisitos que tienen un impacto significativo sobre la arquitectura de diseño del producto a desarrollar son los que ya se definieron en el Documento de Visión como requisitos no funcionales:

Este conjunto de requisitos define la base sobre la cual se desarrollará la arquitectura del sistema, asegurando que cumpla con los objetivos de funcionalidad, seguridad y rendimiento.

* 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.

Por otra parte, hemos de recordar las restricciones que ya se definieron en el Documento de Visión:

● Tecnología:

o El sistema se implementará como una aplicación web, sin una versión móvil nativa en esta fase.

o Se utilizarán bases de datos relacionales para almacenar la información de los usuarios y los gastos.

● Alcance del MVP:

o En esta primera versión no se permitirá la integración con plataformas de pago.

o No se incluirán funcionalidades avanzadas como categorización de gastos o reportes detallados.

● Implementación:

o Capa de presentación: HTML y CSS mediante tecnologías de plantillas de servidor (Thymeleaf, por ejemplo).

o 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:

o La funcionalidad de envío de correos depende de la conexión con Gmail.

● Legales:

o 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).

### 2.2 Patrón de arquitectura seleccionado

Se opta por una **arquitectura en tres capas** que se estructura de la siguiente manera:

* **Capa de presentación:**
  + Aplicación web desarrollada usando HTML y CSS generada mediante tecnologías de plantillas de servidor, responsable de la interfaz de usuario y la experiencia interactiva.
* **Capa de negocio (API):**
  + Java Persistance API (JPA) con Hibernate sobre base de datos relacional.
* **Capa de persistencia:**
  + Base de datos relacional (SQL) que almacena usuarios, grupos, gastos y participaciones.

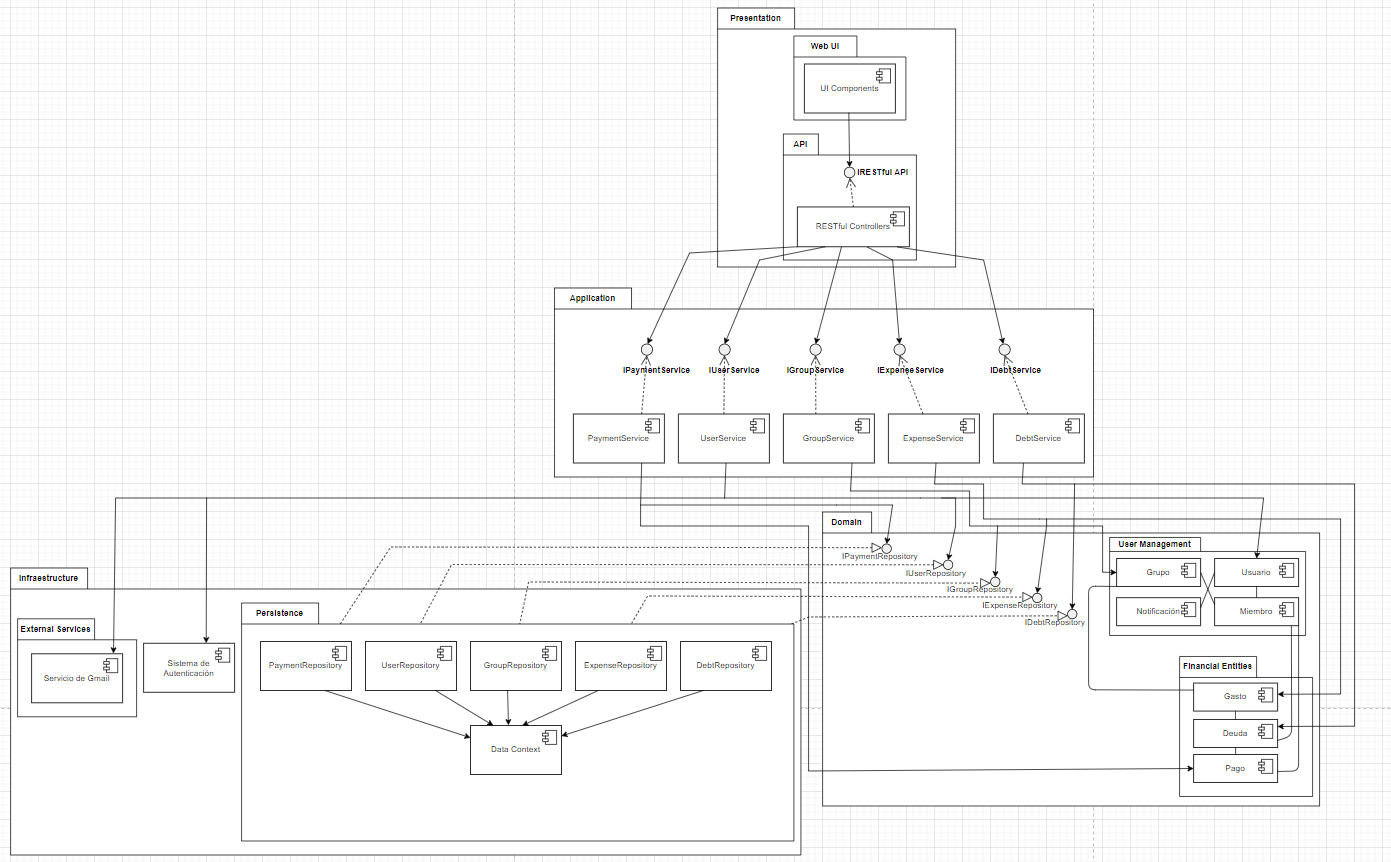
**Justificación:**Esta separación facilita la escalabilidad, el mantenimiento y la posibilidad de desarrollo paralelo entre front-end y back-end, lo que resulta fundamental en un entorno ágil y con entregas iterativas (Scrum).

### 2.3 Otros criterios y restricciones

* **Estándares y herramientas:**
  + Se seguirán convenciones de RESTful API y se utilizarán estándares de codificación y seguridad (por ejemplo, HTTPS, encriptación de contraseñas).
  + El equipo usa Trello para la gestión de tareas, GitHub para el código y Slack para la comunicación interna.
* **Restricciones tecnológicas:**
  + El sistema será una aplicación web (sin versión nativa móvil en el MVP).
  + La elección de tecnologías responde a la experiencia previa del equipo y a la disponibilidad de recursos.

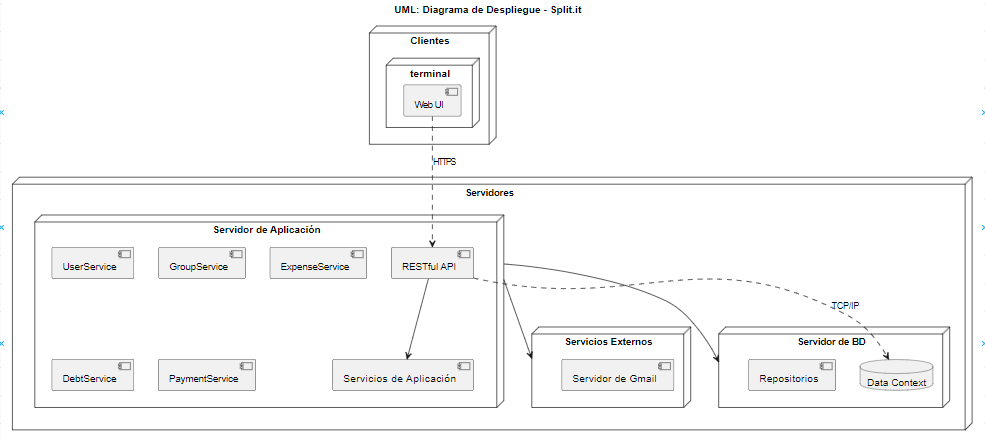
# DESCRIPCIÓN DE LA ARQUITECTURA

## Vista lógica de la arquitectura



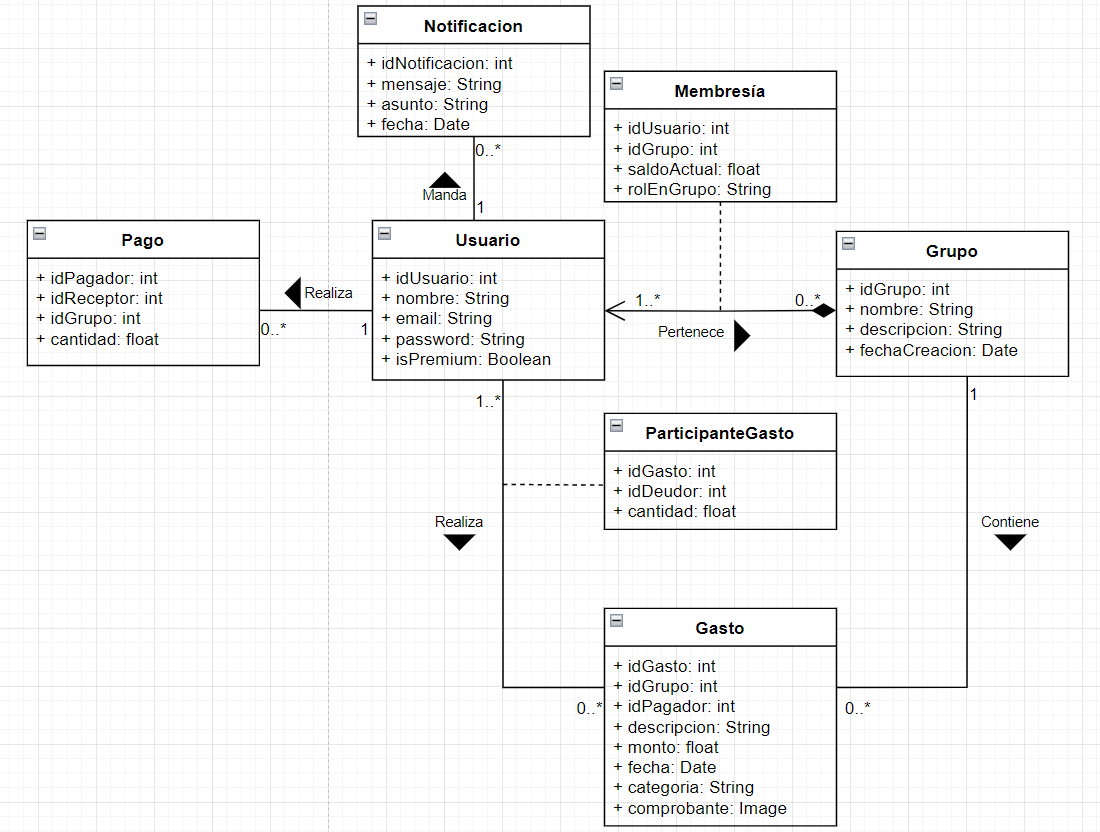
*Diagrama 1: Diagrama de paquete split.it*

## Vista de despliegue de la arquitectura

**

*Diagrama 2: Diagrama de despliegue.*

# MODELO DE DATOS



*Diagrama 3: Diagrama de clases.*

*Tabla notificación:*

| ***Atributo*** | ***Descripción*** | ***Tipo de Dato*** | ***PK*** | ***Obligatorio*** | ***Unicidad*** |
| --- | --- | --- | --- | --- | --- |
| *idNotificacion* | *Identificador único del usuario* | *INTEGER* | *S* | *S* | *S* |
| *asunto* | *Asunto del mensaje* | *VARCHAR(255)* | *N* | *S* | *N* |
| *mensaje* | *Contenido del mensaje* | *VARCHAR(255)* | *N* | *S* | *N* |
| *fecha* | *Fecha de la notificación* | *DATE* | *N* | *S* | *N* |

*Tabla usuario:*

| ***Atributo*** | ***Descripción*** | ***Tipo de Dato*** | ***PK*** | ***Obligatorio*** | ***Unicidad*** |
| --- | --- | --- | --- | --- | --- |
| *idUsuario* | *Identificador único del usuario* | *INTEGER* | *S* | *S* | *S* |
| *nombre* | *Nombre del usuario* | *VARCHAR(255)* | *N* | *S* | *N* |
| *email* | *Correo electrónico del usuario* | *VARCHAR(255)* | *N* | *S* | *N* |
| *password* | *Contraseña del perfil del usuario* | *VARCHAR(255)* | *N* | *S* | *N* |
| *isPremium* | *Verificador de usuario premium* | *BOOLEAN* | *N* | *S* | *N* |

*Tabla miembro*

| ***Atributo*** | ***Descripción*** | ***Tipo de Dato*** | ***PK*** | ***Obligatorio*** | ***Unicidad*** |
| --- | --- | --- | --- | --- | --- |
| *idMiembro* | *Identificador único del usuario* | *INTEGER* | *S* | *S* | *S* |
| *saldoActual* | *Saldo actual del miembro del grupo* | *FLOAT* | *N* | *S* | *N* |
| *rolEnGrupo* | *Rol del miembro en el grupo* | *VARCHAR(255)* | *N* | *S* | *N* |
| *idUsuario* | *Referencia al usuario (FK)* | *INTEGER* | *N* | *S* | *N* |
| *idGrupo* | *Referencia al grupo (FK)* | *INTEGER* | *N* | *S* | *N* |

*Tabla gasto:*

| ***Atributo*** | ***Descripción*** | ***Tipo de Dato*** | ***PK*** | ***Obligatorio*** | ***Unicidad*** |
| --- | --- | --- | --- | --- | --- |
| *idGasto* | *Identificador único del gasto* | *INTEGER* | *S* | *S* | *S* |
| *monto* | *Monto total del gasto* | *FLOAT* | *N* | *S* | *N* |
| *fecha* | *Fecha en que se registró el gasto* | *DATE* | *N* | *S* | *N* |
| *descripcion* | *Descripción del gasto* | *VARCHAR(255)* | *N* | *S* | *N* |
| *categoria* | *Categoría del gasto* | *VARCHAR(255)* | *N* | *S* | *N* |
| *comprobante* | *Comprobante del gasto* | *BLOB* | *N* | *N* | *N* |

*Tabla grupo:*

| **Atributo** | **Descripción** | **Tipo de Dato** | **PK** | **Obligatorio** | **Unicidad** |
| --- | --- | --- | --- | --- | --- |
| idGrupo | Identificador único del grupo | INTEGER | S | S | S |
| nombre | Nombre del grupo | *VARCHAR(255)* | N | S | N |
| descripcion | Descripción del grupo | *VARCHAR(255)* | N | N | N |
| fechaCreacion | Fecha de creación del grupo | DATE | N | S | N |

*Tabla deuda:*

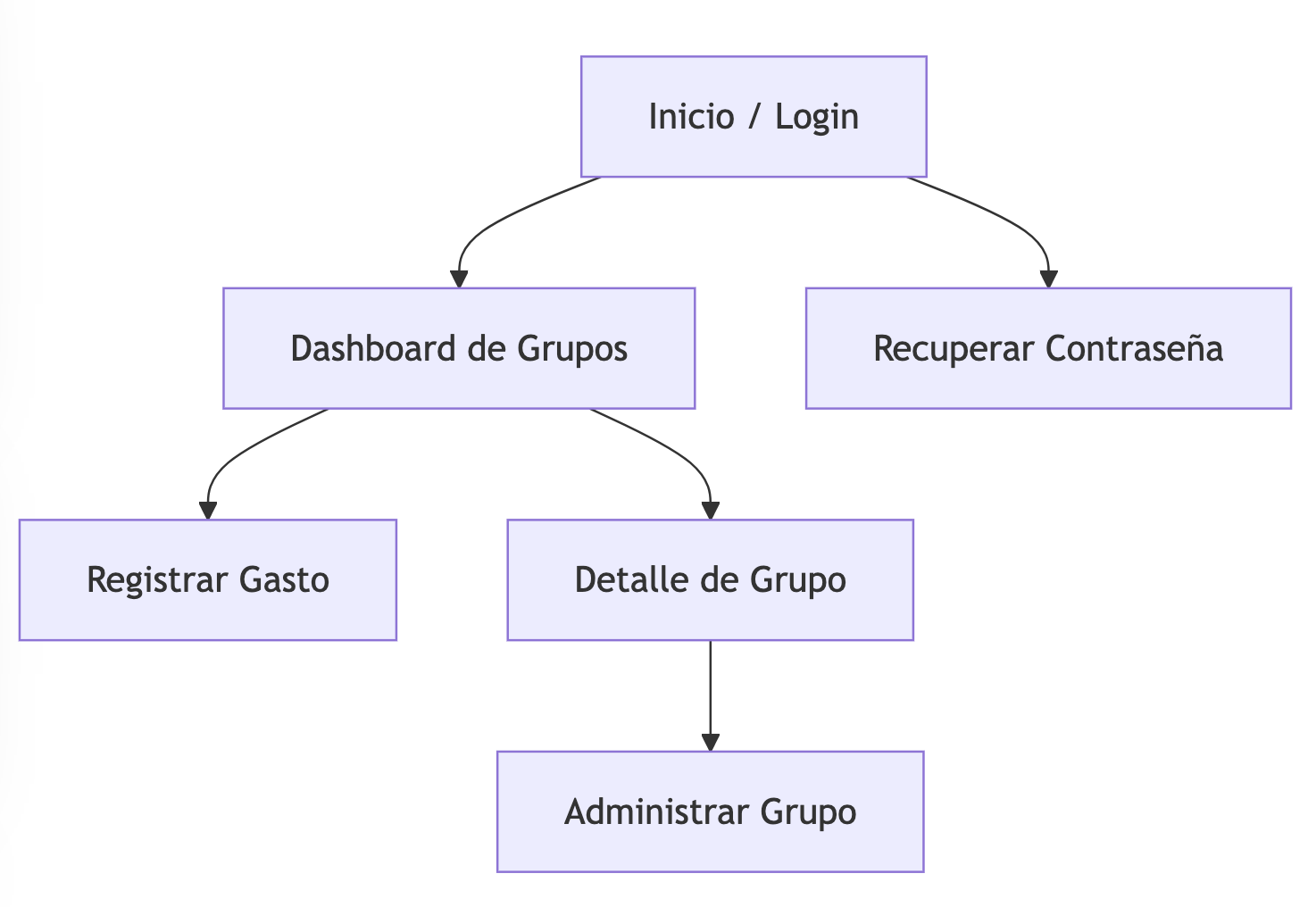
| **Atributo** | **Descripción** | **Tipo de Dato** | **PK** | **Obligatorio** | **Unicidad** |
| --- | --- | --- | --- | --- | --- |
| idDeuda | Identificador único de la deuda | INTEGER | S | S | S |
| monto | Monto de la deuda | *FLOAT* | N | S | N |
| estaSaldada | Indica si la deuda ha sido saldada | *BOOLEAN* | N | S | N |
| fechaCreacion | Fecha de creación de la deuda | DATE | N | S | N |

*Tabla pago:*

| **Atributo** | **Descripción** | **Tipo de Dato** | **PK** | **Obligatorio** | **Unicidad** |
| --- | --- | --- | --- | --- | --- |
| idPago | Identificador único del pago | INTEGER | S | S | S |
| monto | Monto del pago | *FLOAT* | N | S | N |
| metodoPago | Método de pago utilizado | *VARCHAR(255)* | N | S | N |
| fechaPago | Fecha de creación del pago | DATE | N | S | N |

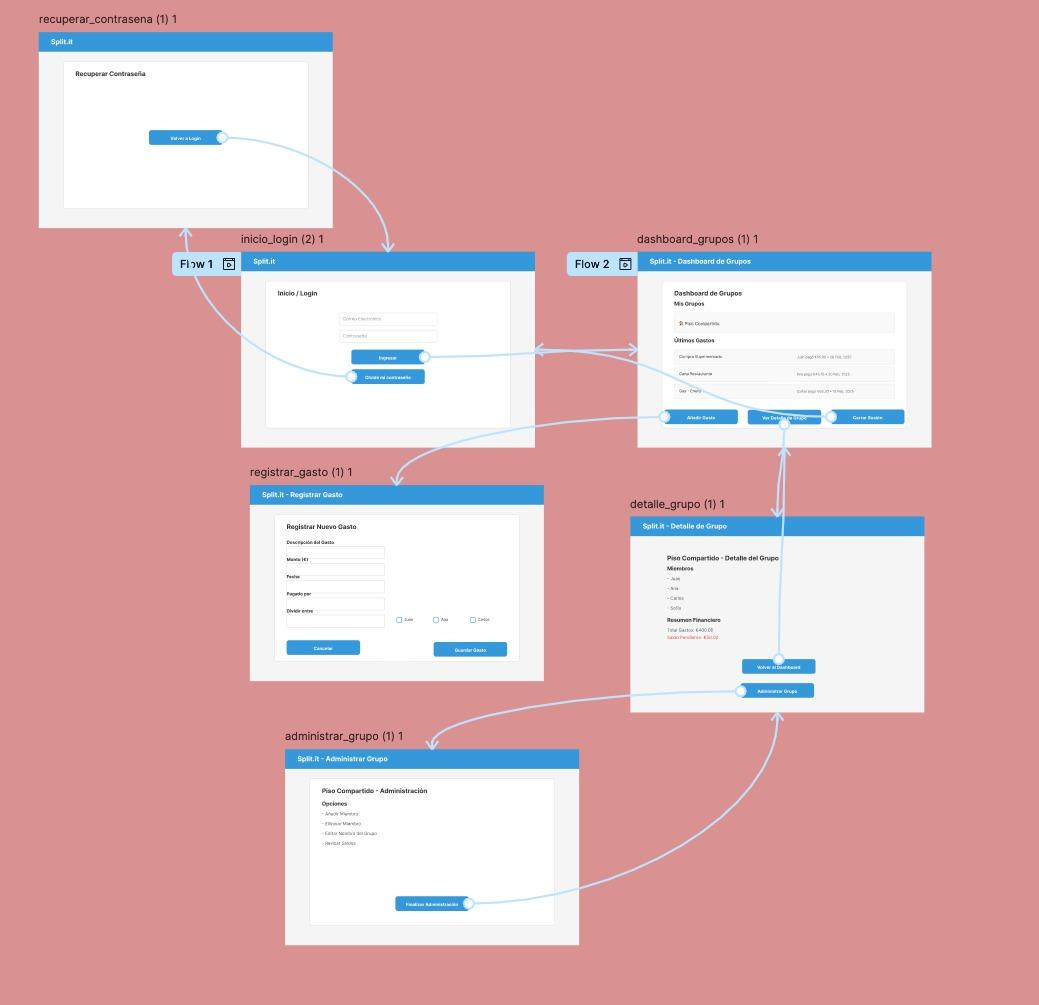
# interfaz de usuario

## Estructura y contenido

**

*Diagrama 4: Diagrama de estructura.*

## Flujos de navegación

**

*Diagrama 5: Diagrama de flujos.*