

Universidad Nacional de Colombia - Sede Medellín

Facultad de Minas
Departamento de Ciencias de la Computación y de la Decisión

**Modelado y Diseño de un Sistema de Gestión de Fondos Compartidos e
individuales mediante la Metodología ICONIX**

Santiago Montoya Orozco
samontoyaor@unal.edu.co

Victoria Adelaida Múnera Leal
vamuneral@unal.edu.co

Juan Jose Monsalve Marin
jjmonsalmem@unal.edu.co

Profesor: ALBEIRO ESPINOSA BEDOYA, Ph.D.
aespinos@unal.edu.co

Medellín
2025

RESUMEN

Este proyecto tiene como objetivo desarrollar una aplicación que permita a los usuarios gestionar sus finanzas personales y fomentar el ahorro individual y colaborativo. La plataforma integrará mecanismos de control para desmotivar retiros impulsivos y promover el cumplimiento de metas financieras.

El sistema contará con dos modalidades de ahorro: personal y grupal. En el ahorro personal, los usuarios podrán establecer metas específicas. En el ahorro colaborativo, los grupos de usuarios definirán objetivos comunes y los retiros van a requerir la aprobación de un porcentaje preestablecido de los integrantes.

La aplicación también incorporará herramientas de visualización de transacciones, estadísticas de ahorro, integración con metas financieras y elementos de gamificación, como insignias y clasificaciones de usuarios.

El desarrollo del sistema se realizará utilizando la metodología ICONIX, asegurando un diseño estructurado con modelos de dominio, diagramas de casos de uso, robustez y secuencia. La implementación técnica se llevará a cabo con Django, permitiendo una arquitectura flexible y escalable.

Este documento se centrará en la documentación del proceso bajo ICONIX, detallando el análisis de requerimientos, diseño del sistema y modelo de datos.

PRESENTACIÓN

En la actualidad, la gestión financiera personal y el ahorro se han convertido en aspectos fundamentales para la estabilidad económica de las personas y grupos. Sin embargo, muchas veces los usuarios enfrentan dificultades para mantener la disciplina en sus ahorros y evitar retiros impulsivos que comprometen sus metas financieras.

Este proyecto propone el desarrollo de una **Aplicación de Gestión Financiera con Ahorros Personales y Colaborativos**, cuyo objetivo es brindar a los usuarios una herramienta que les permita administrar sus finanzas de manera efectiva, promoviendo el ahorro individual y grupal mediante mecanismos de control.

A través de la metodología **ICONIX**, se diseñará la estructura del sistema, asegurando una documentación clara y detallada del proceso de desarrollo. La implementación técnica se realizará con **Django**, garantizando una arquitectura robusta y escalable.

Este documento presenta la planificación, análisis y modelado del sistema, detallando los casos de uso, modelos de dominio y diagramas de diseño que fundamentan la solución propuesta.

Aplicación de Gestión Financiera con Ahorros Personales y Colaborativos

INTRODUCCIÓN

El ahorro es una de las bases fundamentales para una buena gestión financiera, tanto a nivel personal como en proyectos colectivos. Sin embargo, muchas personas encuentran difícil mantener la disciplina en sus ahorros o enfrentan decisiones impulsivas que afectan sus metas a largo plazo. En este contexto, contar con herramientas que faciliten la administración del dinero y fomenten hábitos financieros saludables se vuelve esencial.

Este proyecto propone el desarrollo de una **Aplicación de Gestión Financiera con Ahorros Personales y Colaborativos**, diseñada para ayudar a los usuarios a organizar su dinero de manera efectiva. La plataforma permitirá establecer metas de ahorro individuales, así como formar grupos para ahorrar en conjunto con reglas claras para el uso de los fondos. Además, se incorporarán mecanismos para desmotivar retiros impulsivos y fomentar el cumplimiento de objetivos mediante recordatorios, aprobaciones de terceros y elementos de gamificación.

Para garantizar un desarrollo estructurado y eficiente, se aplicará la metodología **ICONIX**, permitiendo un enfoque detallado en la definición de requerimientos, modelado del sistema y validación de procesos. A nivel técnico, la implementación se realizará con **Django**, aprovechando sus capacidades para la construcción de aplicaciones web escalables y seguras.

Este informe documenta cada paso del proceso, desde la identificación de necesidades hasta el diseño detallado de la aplicación. A través de este trabajo, se busca no solo desarrollar una solución funcional, sino también comprender y aplicar buenas prácticas en la construcción de productos de software, alineadas con metodologías de desarrollo estructurado.

Para el desarrollo de este proyecto nos basamos en un concepto conocido en el área bancaria como **Cuentas de Ahorro Mancomunadas**.

Una cuenta de ahorros mancomunada, también conocida como cuenta conjunta, es un tipo de cuenta bancaria que permite a múltiples titulares administrar fondos en conjunto. Todos los titulares tienen acceso a los recursos, aunque ciertas operaciones pueden requerir la aprobación de todos o de una mayoría, dependiendo de los acuerdos establecidos.

Características principales

- Acceso compartido a los fondos por parte de todos los titulares.
- Posibilidad de establecer restricciones o niveles de aprobación para ciertas transacciones.
- Delegación de un titular como representante del grupo si así se define.

Usos comunes

- Gestión financiera entre parejas, familiares o socios comerciales.
- Facilitación del ahorro y la administración conjunta de dinero.

Consideraciones

- Puede generar conflictos si los titulares tienen hábitos financieros distintos.
- Algunas instituciones pueden retener o deducir fondos para cubrir deudas individuales de un titular.
- En caso de fallecimiento de un titular, el acceso a los fondos puede quedar restringido hasta obtener el consentimiento de los herederos legales.

Objetivo General

Desarrollar una aplicación de gestión financiera que permita a los usuarios administrar su ahorro personal y colaborativo, fomentando hábitos financieros responsables mediante mecanismos de control que desmotivan los retiros impulsivos.

Objetivos Específicos

1. **Implementar un sistema de ahorro personal y colaborativo**, permitiendo a los usuarios definir metas financieras individuales o en grupo.
2. **Establecer mecanismos de control para los retiros de fondos**, incluyendo aprobaciones de terceros y mensajes de confirmación que desincentivan decisiones impulsivas.
3. **Facilitar la gestión de transacciones**, permitiendo el registro detallado de ingresos, gastos y movimientos de ahorro.
4. **Desarrollar una interfaz intuitiva y accesible**, asegurando una experiencia de usuario sencilla y eficiente.
5. **Incorporar elementos de gamificación**, como insignias y clasificaciones, para incentivar el cumplimiento de metas de ahorro.
6. **Aplicar la metodología ICONIX en el diseño del sistema**, documentando cada fase con modelos de dominio, diagramas de casos de uso, robustez y secuencia.
7. **Implementar la solución con Django**, garantizando una arquitectura flexible, segura y escalable.

Descripción del Problema

El manejo del ahorro y la administración financiera conjunta son desafíos comunes para muchas personas. Aunque existen cuentas bancarias mancomunadas, su uso puede ser complicado debido a la falta de flexibilidad en los procesos de autorización y la ausencia de herramientas que fomenten la disciplina financiera.

Actualmente, las personas que desean ahorrar en conjunto, ya sea en familia, entre amigos o con socios comerciales, enfrentan problemas como:

- **Falta de control en los retiros**: No siempre hay mecanismos claros para evitar que un solo miembro acceda a los fondos sin la aprobación del resto.
- **Dificultad para establecer reglas de ahorro personalizadas**: Las cuentas mancomunadas tradicionales no ofrecen configuraciones flexibles que permitan definir restricciones específicas para los retiros.

- **Impulsividad en la gestión del dinero:** Muchas personas retiran fondos sin considerar el impacto en sus metas financieras, afectando su capacidad de ahorro.
- **Falta de motivación:** No hay incentivos que fomenten la constancia en el ahorro ni herramientas que visualicen el progreso de manera clara.

Necesidad de una solución digital

Dado que la tecnología permite automatizar procesos y mejorar la seguridad en la gestión financiera, se hace necesario el desarrollo de una aplicación que facilite el ahorro tanto individual como colaborativo. Esta aplicación debe permitir:

- Crear metas de ahorro con reglas claras para los retiros.
- Establecer mecanismos de aprobación para garantizar el consentimiento de los participantes en una cuenta conjunta.
- Integrar herramientas de gamificación para motivar a los usuarios a cumplir sus objetivos financieros.
- Ofrecer un registro detallado de transacciones para brindar transparencia y control.

PROCESO DE DESARROLLO ICONIX

1. MODELO DE DOMINIO

El modelo de dominio define las entidades principales del sistema y sus relaciones, estableciendo una base conceptual clara para el desarrollo de la aplicación de gestión financiera. A través de este modelo, se representan los elementos clave que intervienen en la administración de cuentas de ahorro, tanto individuales como grupales, así como la gestión de transacciones y notificaciones.

Descripción de las Entidades y Relaciones

Usuario y Administrador

- La entidad **Usuario** representa a cualquier persona que utilice la aplicación para administrar sus ahorros.
- El **Administrador** es un tipo de usuario con permisos adicionales para gestionar grupos de ahorro y cuentas colaborativas.

Grupo de Ahorro y Cuentas

- Un **GrupoAhorro** está compuesto por varios usuarios y administradores que gestionan fondos en conjunto.
- Las **CuentasAhorro** pueden ser personales o grupales (**CuentaGrupal**).
- Un grupo de ahorro puede estar vinculado a una o más cuentas grupales.

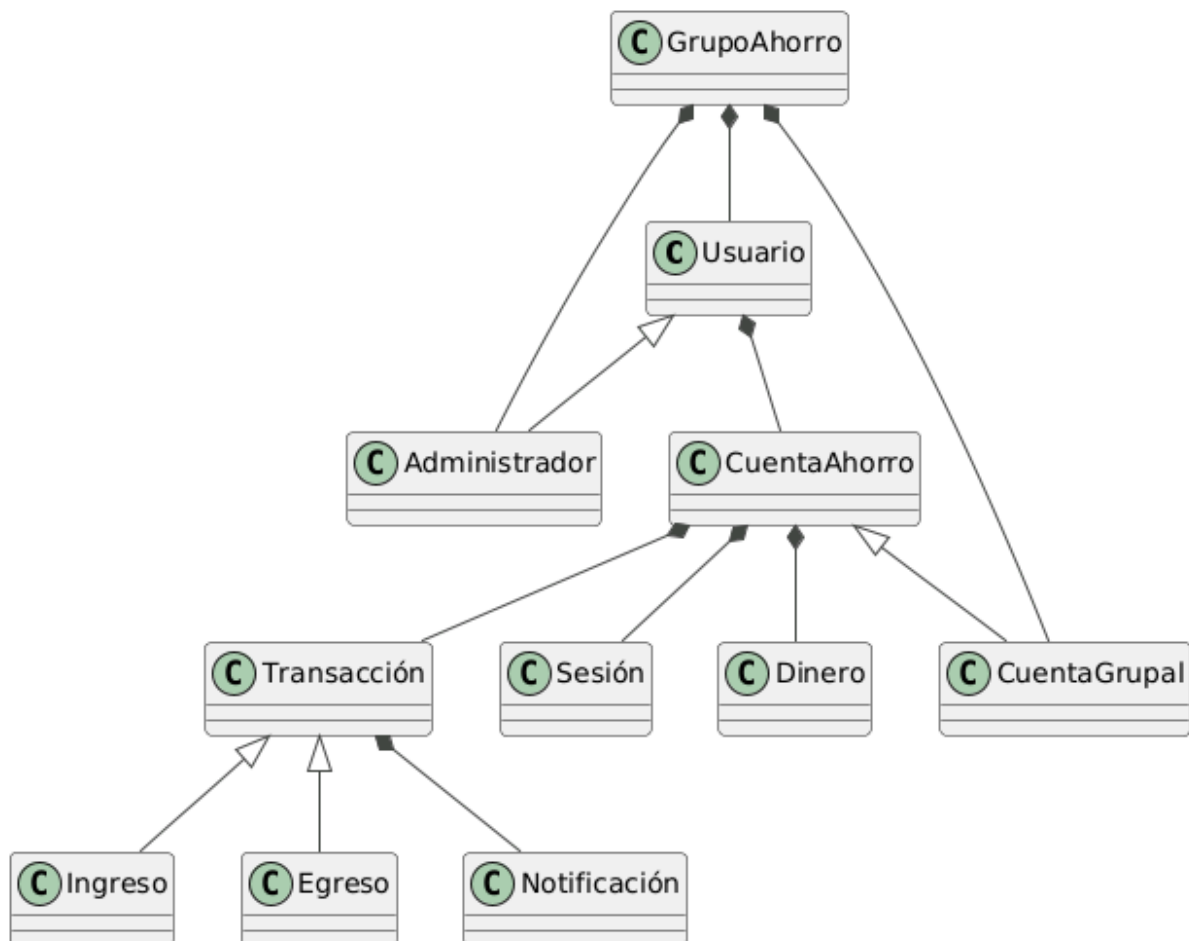
Transacciones y Movimientos Financieros

- Las **transacciones** representan cualquier movimiento de dinero dentro de una cuenta. Pueden ser de tipo **Ingreso** o **Egreso**.
- Cada transacción está asociada a una **CuentaAhorro** y puede generar una **Notificación** para informar al usuario sobre su actividad financiera.

- La entidad **Dinero** representa los fondos almacenados en una cuenta.

Seguridad y Control de Sesiones

- Cada usuario mantiene una **sesión** activa para acceder a la aplicación y administrar sus cuentas.
- La vinculación entre las cuentas y las sesiones permite gestionar el acceso y la seguridad del sistema.



MODELO DE CASOS DE USO

El modelo de casos de uso define las interacciones que los diferentes actores tienen con el sistema, describiendo las funcionalidades principales de la aplicación. Este modelo es clave dentro de la metodología **ICONIX**, ya que permite identificar los procesos esenciales y establecer una estructura clara para el desarrollo del sistema.

Descripción de los Actores

El sistema cuenta con tres actores principales:

ACTOR	DESCRIPCIÓN
Usuario	<p>Es el actor principal del sistema. Puede crear cuentas de ahorro, gestionar sus fondos, realizar depósitos, solicitar retiros y participar en cuentas colaborativas.</p> <p>Representa a cualquier persona que utilice la aplicación para administrar sus ahorros personales o participar en cuentas grupales. Puede gestionar sus cuentas, realizar transacciones y consultar información financiera.</p>
Administrador	<p>Es responsable de la gestión de cuentas grupales, incluyendo la administración de usuarios, permisos y modificaciones en la configuración de la cuenta.</p>
Grupo de Ahorro	<p>Representa un conjunto de usuarios organizados dentro de una cuenta grupal. Puede gestionar aprobaciones de retiros y enviar invitaciones a nuevos miembros.</p>

Casos de Uso Principales

A continuación, se detallan los casos de uso organizados por actor:

Casos de Uso del Usuario

- Registrar usuario.
- Crear cuenta de ahorro.
- Iniciar y cerrar sesión.
- Depositar y retirar dinero.
- Solicitar retiro o devolución de dinero de una cuenta grupal.
- Consultar información de la cuenta y el historial de transacciones.
- Transferir dinero entre cuentas.
- Solicitar ingreso a una cuenta grupal.
- Eliminar cuenta de ahorro.

Casos de Uso del Administrador

- Crear cuenta de ahorro grupal.
- Gestionar recursos dentro de una cuenta grupal.
- Añadir usuarios a una cuenta grupal.
- Dar de baja a usuarios de la cuenta grupal.
- Editar datos de la cuenta grupal.

Casos de Uso del Grupo de Ahorro

- Aprobar retiros de dinero de la cuenta grupal.
- Enviar invitaciones a nuevos usuarios.

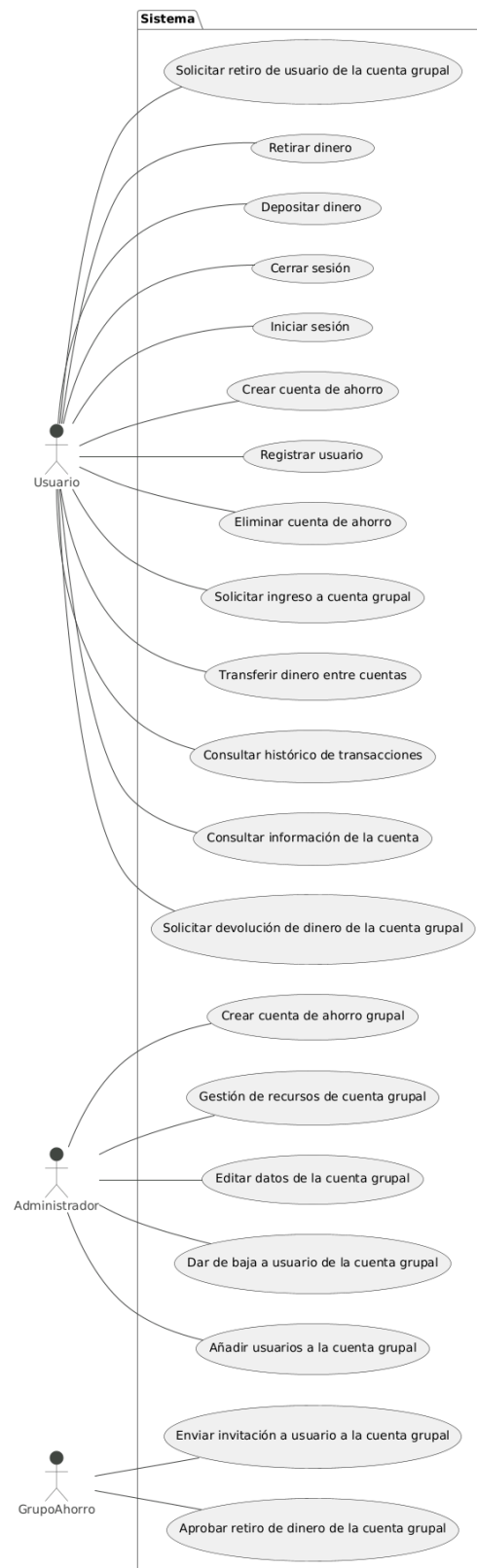


Diagrama de robustez

Los diagramas de robustez son una herramienta fundamental dentro de la metodología **ICONIX**, ya que permiten definir de manera clara la interacción entre los actores del sistema y sus componentes principales. Estos diagramas sirven como un puente entre los casos de uso y el diseño detallado del sistema, asegurando que la funcionalidad descrita en cada caso de uso se traduzca en una estructura lógica y organizada.

En los diagramas de robustez se utilizan tres tipos de componentes:

- **Boundary (Interfaz):** Representa las pantallas o formularios con los que interactúan los usuarios.
- **Control:** Define la lógica y validaciones necesarias para procesar la información.
- **Entity (Entidad):** Representa los datos o elementos almacenados en el sistema.

Cada uno de los diagramas a continuación describe la secuencia de acciones y validaciones necesarias para ejecutar correctamente los casos de uso identificados en el sistema. Estos diagramas permiten detectar posibles inconsistencias o mejoras en el diseño antes de pasar a la fase de implementación.

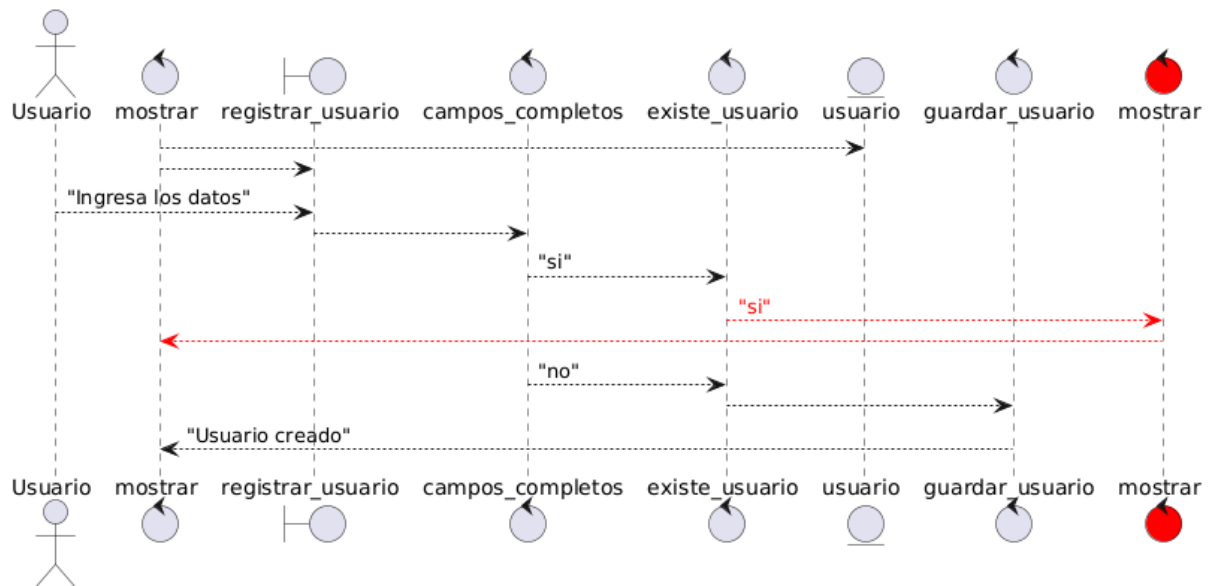
A continuación, se presentan los diagramas de robustez correspondientes a los principales casos de uso del sistema.

1. Registrar usuario

El siguiente diagrama de robustez representa el proceso de **registro de un nuevo usuario** en el sistema. Este proceso garantiza que los datos ingresados sean correctos y que no exista un usuario previamente registrado con la misma información.

Descripción del Flujo:

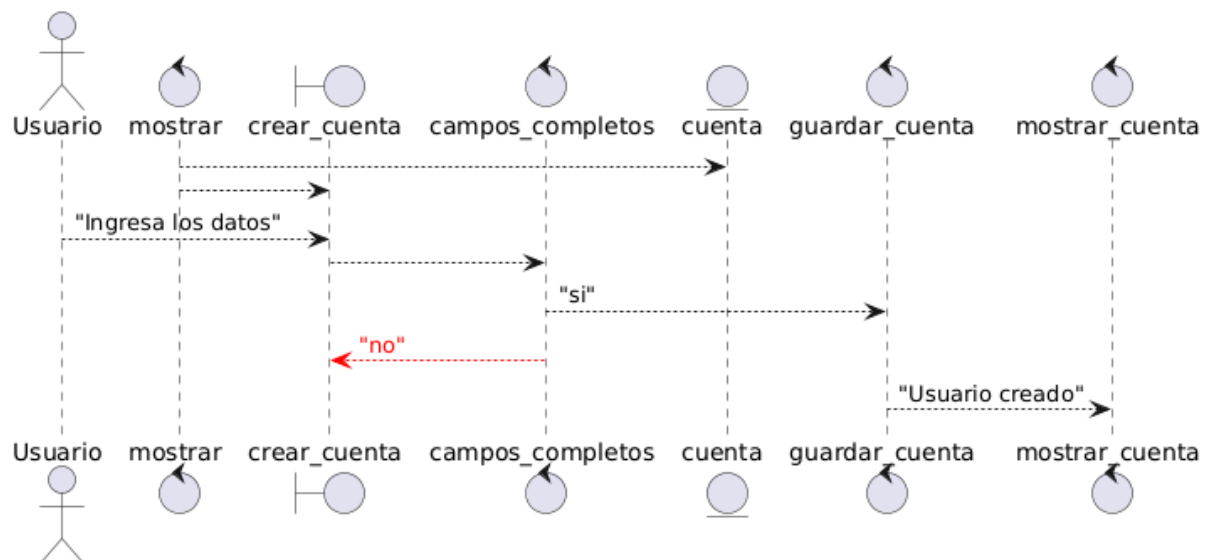
- El usuario ingresa sus datos en el formulario de registro.
- El sistema verifica si los campos están completos.
 - a. Si hay datos faltantes, se devuelve un mensaje de error.
 - b. Si los datos están completos, se continúa con la validación.
- Se revisa si el usuario ya existe en la base de datos.
 - c. Si el usuario ya está registrado, se muestra un mensaje de error.
 - d. Si el usuario no existe, se procede a guardarlo en la base de datos.
- El usuario es creado exitosamente y se muestra un mensaje de confirmación.



2. Crear cuenta de ahorro

El siguiente diagrama de robustez representa el proceso de **creación de una cuenta de ahorro** en el sistema. Este flujo asegura que los datos ingresados sean correctos antes de registrar la cuenta en la base de datos.

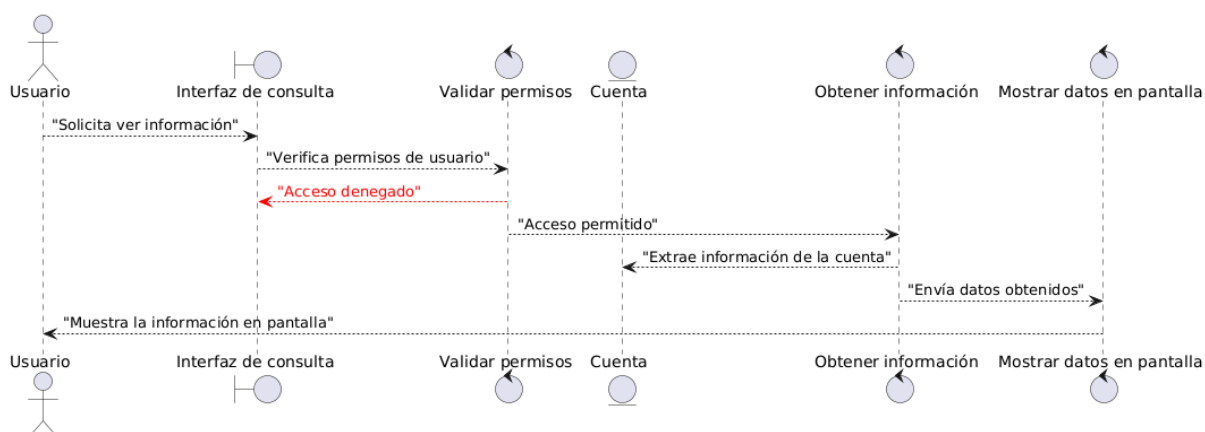
- El usuario accede a la interfaz de creación de cuenta e ingresa los datos requeridos.
- El sistema valida que todos los campos estén completos.
 - a. Si hay datos faltantes, se muestra un mensaje de error y se solicita completar la información.
 - b. Si los datos están correctos, se procede a guardar la cuenta.
- La cuenta es almacenada en la base de datos y se genera un registro asociado al usuario.
- El sistema muestra la información de la nueva cuenta creada, confirmando su registro exitoso.



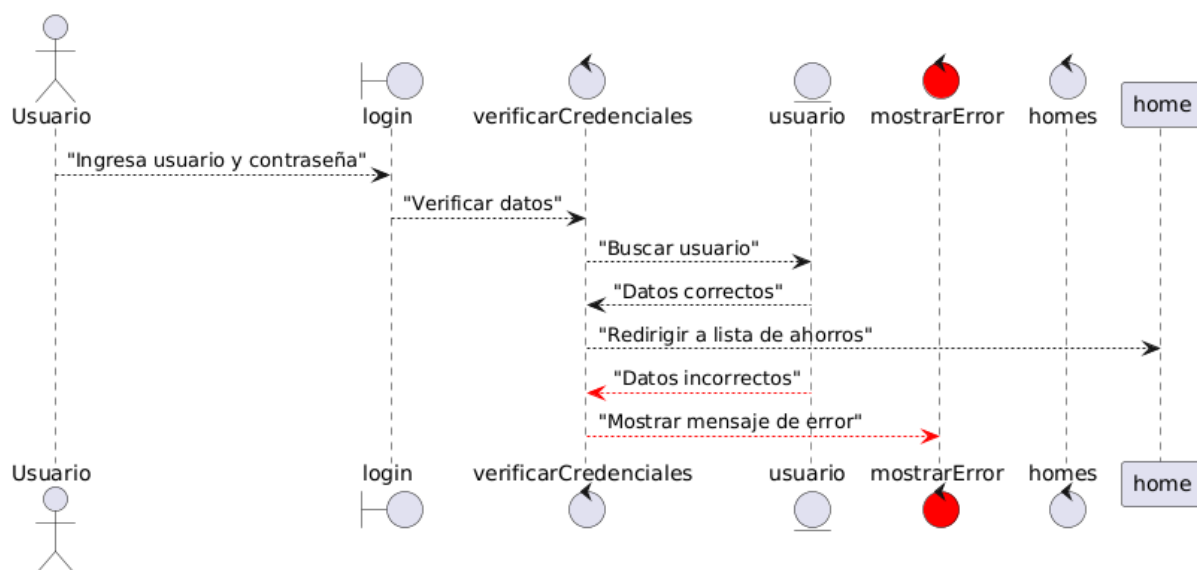
3. Crear cuenta de ahorro grupal

Descripción del Flujo:

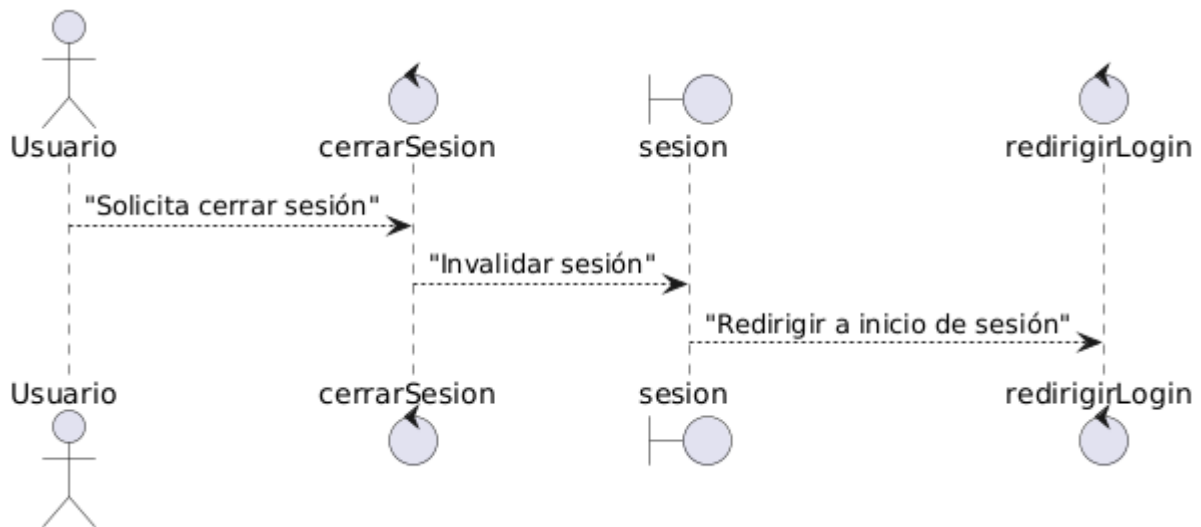
- El administrador accede a la interfaz de la cuenta y selecciona la opción de invitar a un usuario.
- El sistema valida que el usuario sea elegible (ej. que no esté ya en la cuenta, que su correo sea válido, etc.).
 - a. Si el usuario no es válido, se muestra un mensaje de error.
 - b. Si el usuario es válido, se continúa con la invitación.
- El sistema registra al usuario en la cuenta grupal y actualiza la información de la cuenta.
- Se envía una notificación al usuario invitado, informándole sobre su inclusión en la cuenta grupal.



4. Iniciar sesión



5. Cerrar sesión

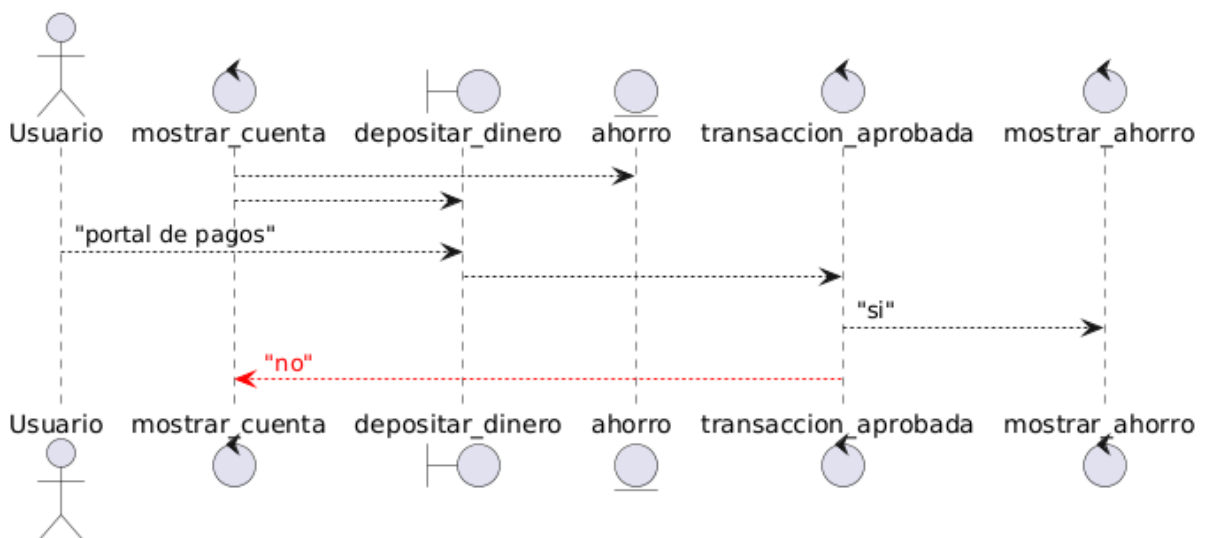


6. Depositar dinero

El siguiente diagrama de robustez representa el proceso de **depósito de dinero en una cuenta de ahorro**. Este flujo garantiza que la transacción se procese correctamente y que el saldo de la cuenta se actualice en función del depósito realizado.

Descripción del Flujo:

- El usuario accede a la interfaz de su cuenta y selecciona la opción de depósito.
- El sistema redirige al usuario al portal de pagos, donde se realiza la transacción.
- Se verifica si la transacción fue aprobada.
 - a. Si la transacción es exitosa, el saldo de la cuenta de ahorro se actualiza y se muestra el nuevo monto.
 - b. Si la transacción es rechazada, el sistema devuelve al usuario a la interfaz de la cuenta sin modificar el saldo y muestra un mensaje de error.

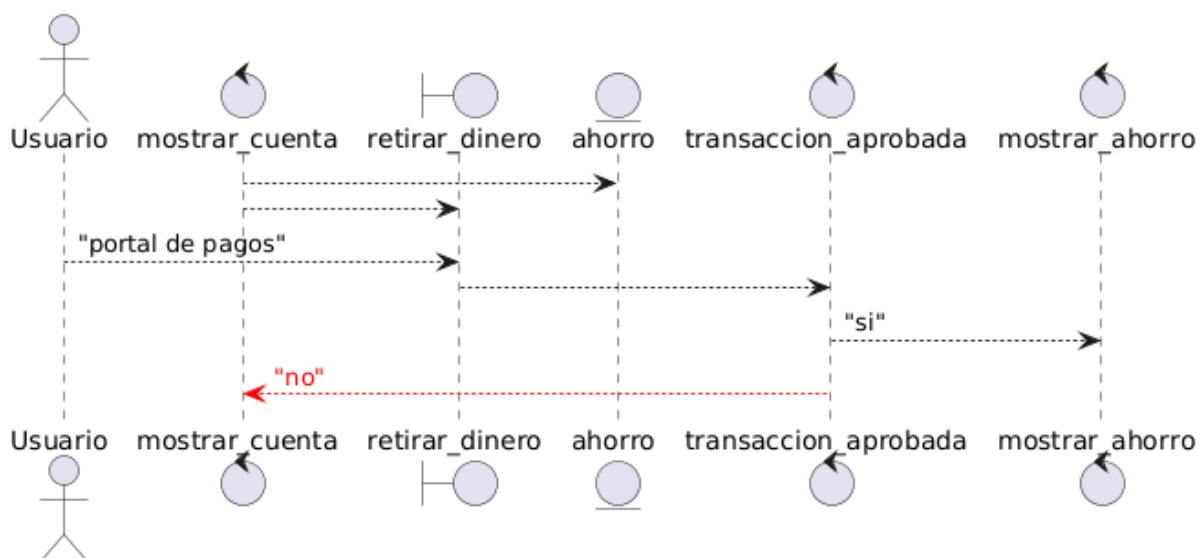


7. Retirar dinero

El siguiente diagrama de robustez representa el proceso de **retiro de dinero desde una cuenta de ahorro**. Este flujo garantiza que las transacciones sean procesadas correctamente y que el saldo de la cuenta se actualice únicamente cuando el retiro es aprobado.

Descripción del Flujo:

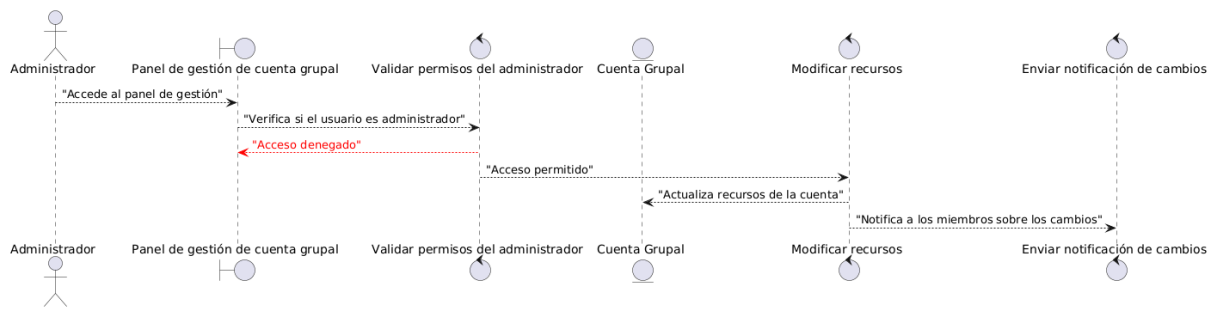
- El usuario accede a la interfaz de su cuenta y selecciona la opción de retiro de dinero.
- El sistema redirige al usuario al portal de pagos, donde se procesa la solicitud de retiro.
- Se verifica si la transacción fue aprobada.
 - a. Si la transacción es exitosa, el saldo de la cuenta de ahorro se actualiza y se muestra el nuevo monto disponible.
 - b. Si la transacción es rechazada, el usuario regresa a la interfaz de la cuenta sin cambios en el saldo, y se muestra un mensaje de error.



8. Gestión de recursos de cuenta grupal

Descripción del Flujo:

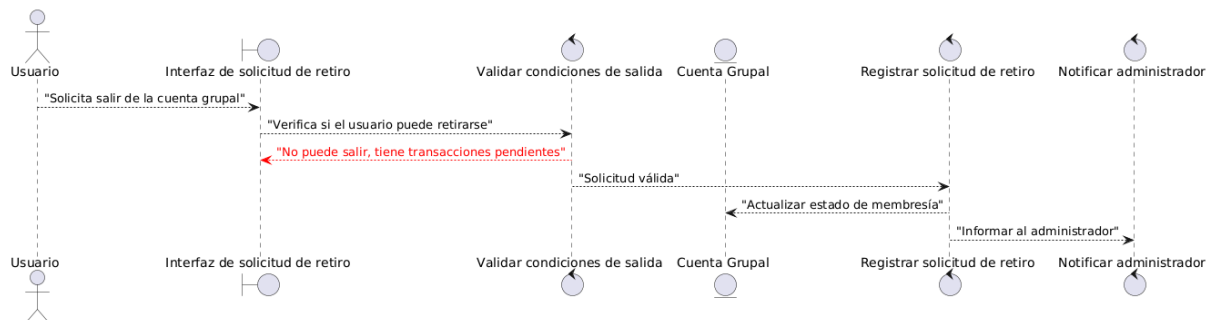
- El administrador accede a la interfaz de gestión de la cuenta grupal.
- El sistema valida si el usuario tiene permisos de administrador.
 - a. Si el usuario no es administrador, se muestra un mensaje de error y se bloquea el acceso.
 - b. Si tiene permisos, puede modificar los recursos de la cuenta.
- El administrador realiza cambios en los recursos de la cuenta, como ajustes en los fondos, reglas de ahorro o configuraciones.
- El sistema actualiza la información en la cuenta grupal.
- Se envía una notificación a los miembros del grupo, informando sobre los cambios realizados.



9. Solicitar retiro de usuario de la cuenta grupal

Descripción del Flujo:

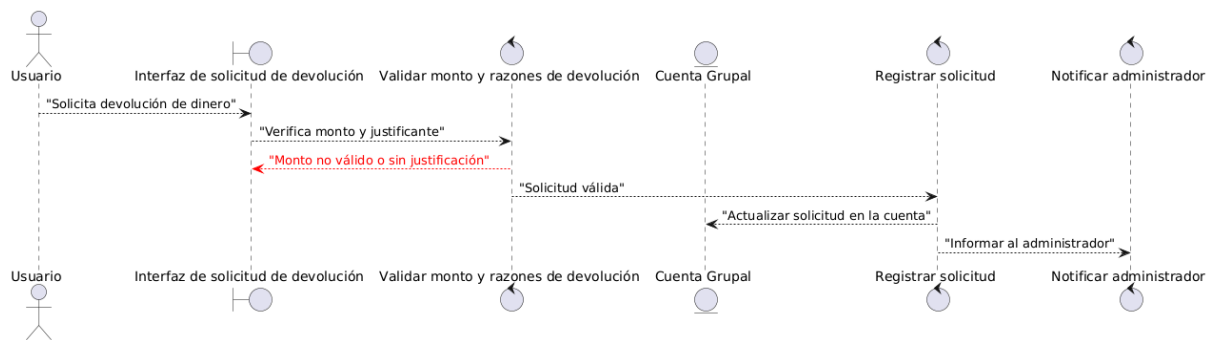
- El usuario accede a la interfaz del sistema y solicita salir de la cuenta grupal.
- El sistema valida si el usuario puede retirarse.
 - a. Si tiene transacciones pendientes o responsabilidades activas dentro del grupo, se muestra un mensaje de error.
 - b. Si cumple con las condiciones, la solicitud es registrada.
- El estado del usuario dentro de la cuenta grupal se actualiza, indicando que ha solicitado su retiro.
- El sistema notifica al administrador de la cuenta grupal, quien podrá aprobar o rechazar la solicitud.



10. Solicitar devolución de dinero de la cuenta grupal

Descripción del Flujo:

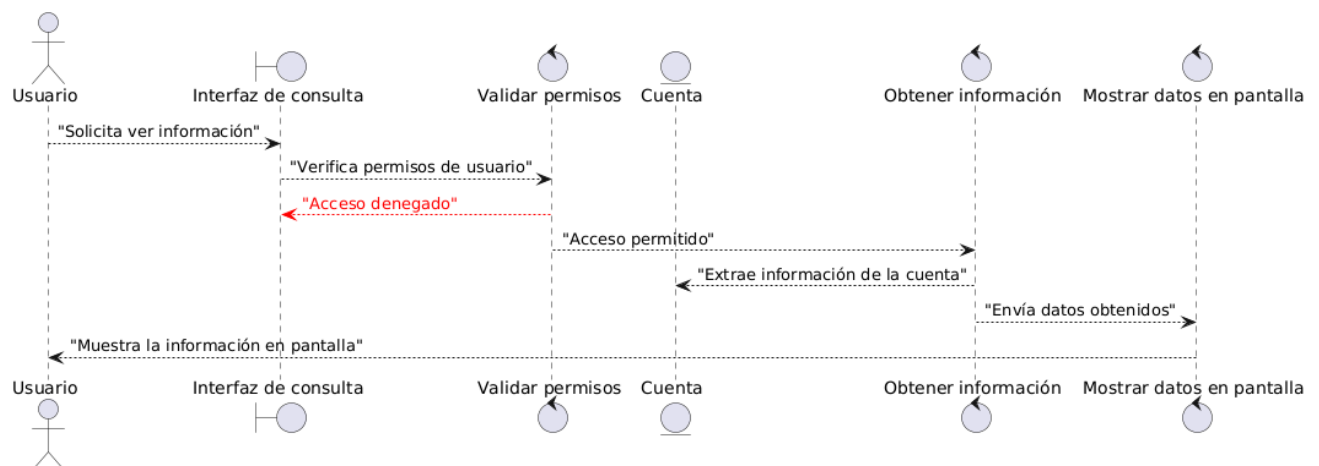
- El usuario accede a la interfaz del sistema y solicita la devolución del dinero, indicando el monto y la razón.
- El sistema valida si la cantidad solicitada es correcta y si hay una justificación válida.
 - a. Si el monto no es válido o la solicitud no tiene justificación, se muestra un mensaje de error.
 - b. Si se cumplen los criterios, la solicitud es registrada.
- La solicitud de devolución queda registrada en la cuenta grupal, pendiente de aprobación.
- El administrador de la cuenta grupal es notificado para revisar la solicitud y proceder con la devolución si corresponde.



11. Consultar información de la cuenta

Explicación del flujo:

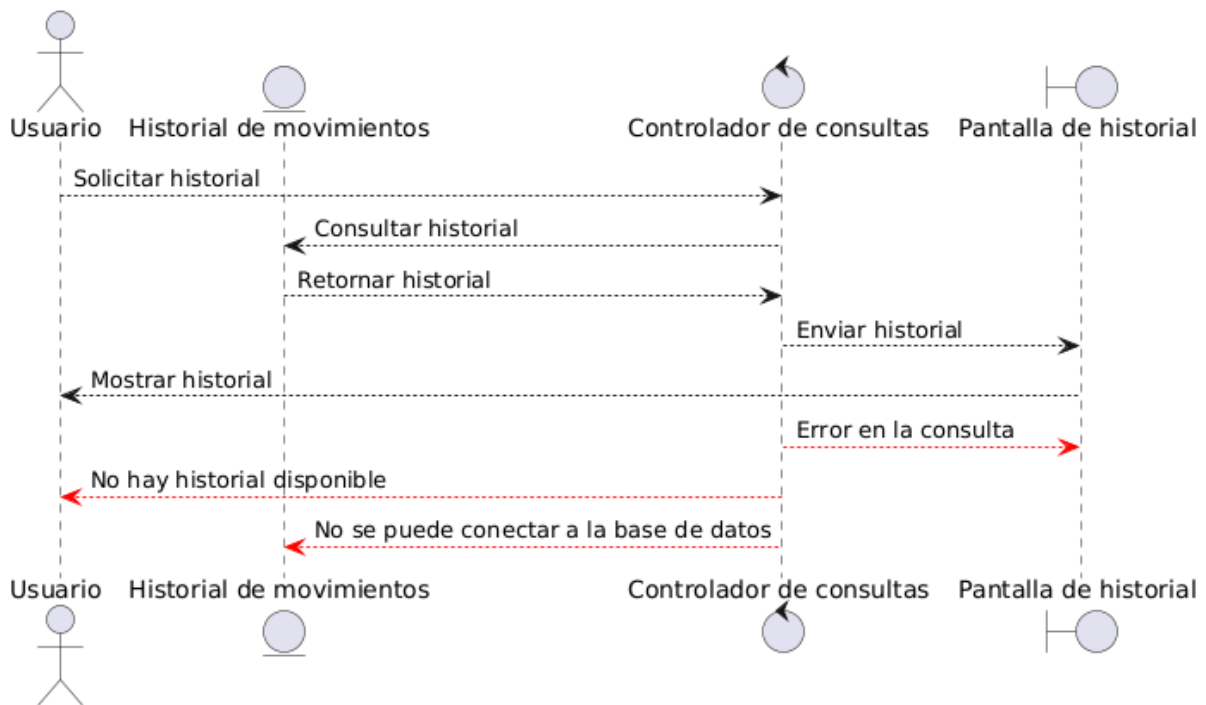
- El usuario solicita ver la información de la cuenta desde la interfaz.
- El sistema valida si el usuario tiene permisos para acceder a la cuenta.
 - a. Si no tiene acceso, se muestra un mensaje de error.
 - b. Si tiene acceso, se procede a recuperar los datos.
- El sistema obtiene la información de la cuenta y la envía a la interfaz.
- Los datos se presentan al usuario en la pantalla.



12. Consultar histórico de transacciones

Descripción del Flujo:

- El usuario accede a la interfaz del sistema y solicita consultar el historial de transacciones.
- El sistema valida que el usuario tenga permisos para acceder a la información.
 - a. Si no tiene acceso, se muestra un mensaje de error.
 - b. Si tiene permisos, se continúa con la consulta.
- El sistema recupera el historial de transacciones desde la base de datos.
- Los datos obtenidos se presentan al usuario en la pantalla, permitiéndole visualizar la información de sus movimientos financieros.

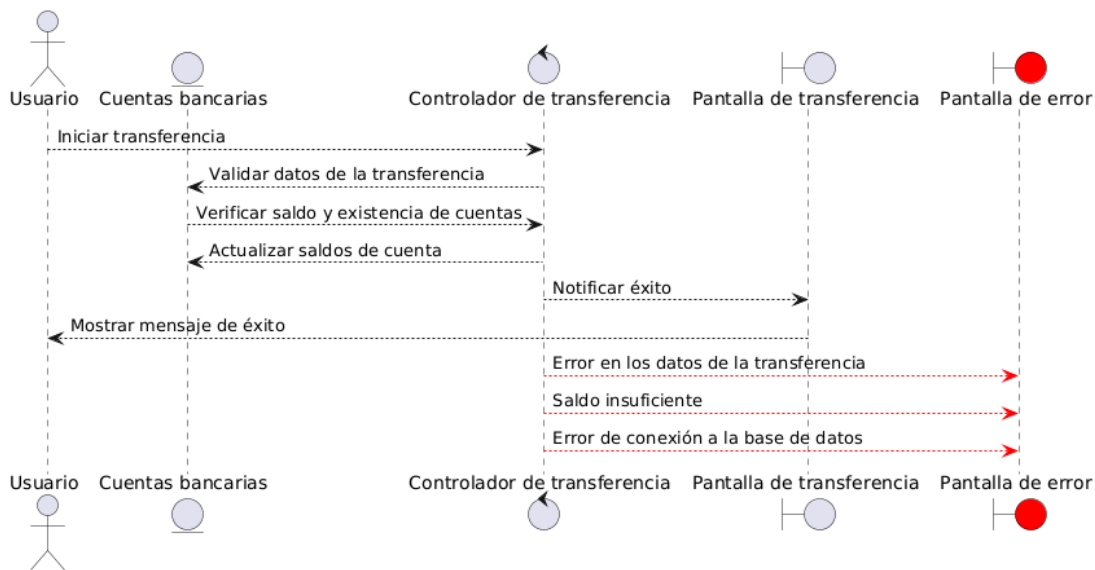


13. Transferir dinero entre cuentas

El siguiente diagrama de robustez representa el proceso de **transferencia de dinero entre cuentas de ahorro**. Este flujo garantiza que las transacciones sean procesadas correctamente y que los saldos de ambas cuentas se actualicen solo si la transferencia es aprobada.

Descripción del Flujo:

- El usuario accede a la interfaz del sistema y selecciona la opción de transferencia de dinero.
- El sistema solicita los datos de la transferencia, verificando que los campos estén completos.
- Se redirige al usuario al portal de pagos, donde se inicia la transacción.
- El sistema valida la disponibilidad de fondos en la cuenta de origen.
 - a. Si los fondos son insuficientes, se muestra un mensaje de error y la operación no se realiza.
 - b. Si hay saldo suficiente, la transacción se procesa.
- Si la transacción es aprobada, el monto se descuenta de la cuenta de origen y se acredita en la cuenta de destino.
- El sistema actualiza el saldo de ambas cuentas y muestra una confirmación de la transferencia.

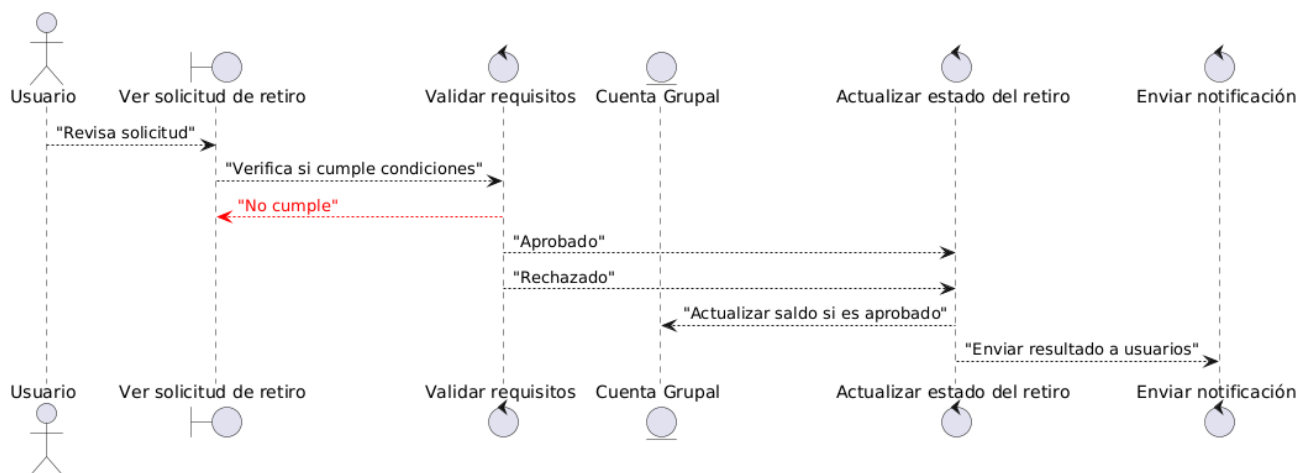


14. Aprobar retiro de dinero de la cuenta grupal

El siguiente diagrama de robustez representa el proceso de **aprobación de un retiro en una cuenta grupal**, asegurando que solo se procesan solicitudes que cumplan con los requisitos establecidos en la cuenta.

Descripción del Flujo:

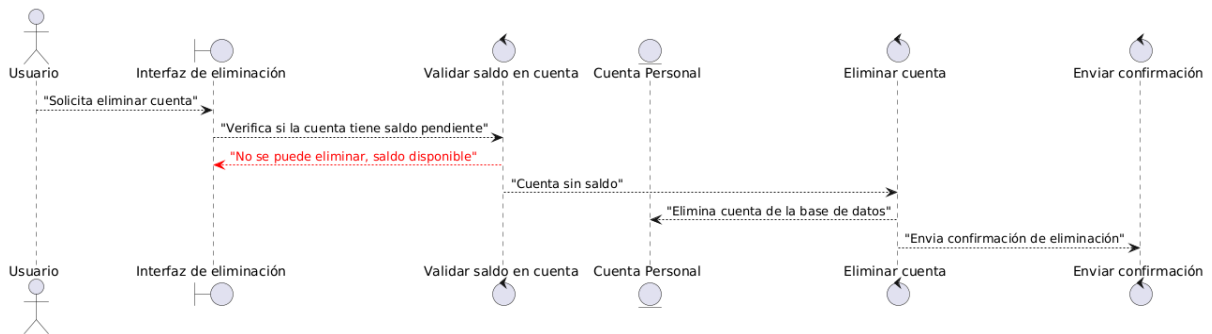
- El usuario accede a la interfaz del sistema y revisa una solicitud de retiro en la cuenta grupal.
- El sistema valida que la solicitud cumpla con los requisitos establecidos, como la aprobación de un porcentaje mínimo de miembros del grupo.
 - a. Si la solicitud no cumple con los requisitos, se muestra un mensaje de error y el proceso finaliza.
 - b. Si la solicitud es válida, se procede a actualizar su estado.
- Si el retiro es aprobado, el saldo de la cuenta grupal se actualiza.
- El sistema envía una notificación a los usuarios informando el resultado de la solicitud (aprobada o rechazada).



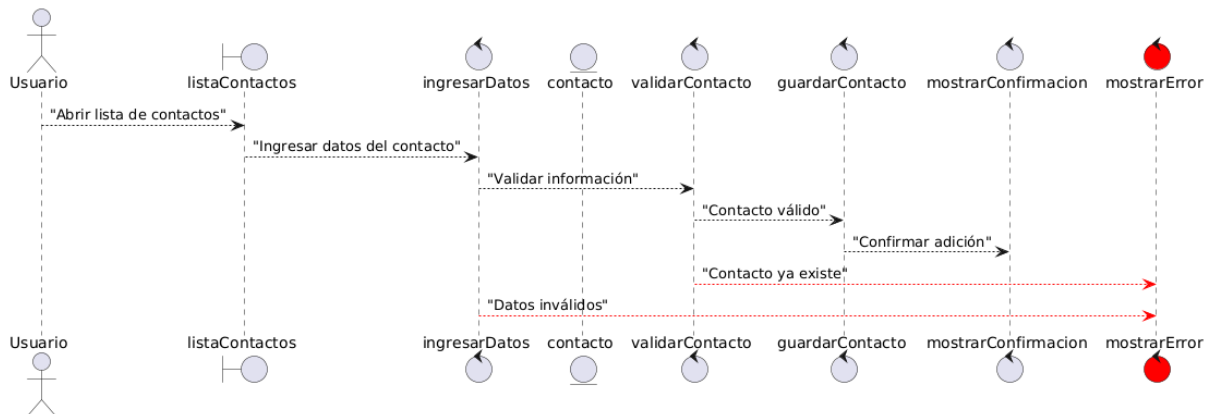
15. Eliminar cuenta de ahorro

Explicación del flujo:

- El usuario solicita eliminar su cuenta personal desde la interfaz.
- El sistema valida que la cuenta no tenga saldo pendiente.
 - a. Si hay saldo, se muestra un mensaje de error.
 - b. Si la cuenta está en cero, se procede a la eliminación.
- El sistema elimina la cuenta personal de la base de datos.
- Se envía una notificación al usuario confirmando la eliminación.



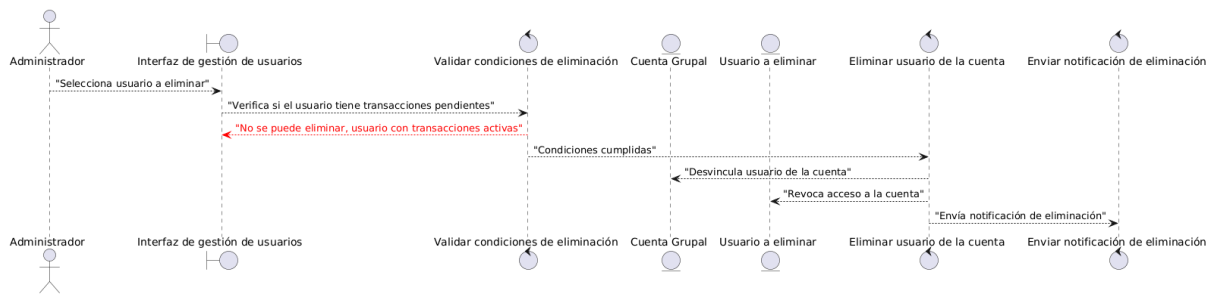
16. Añadir usuarios a la cuenta grupal



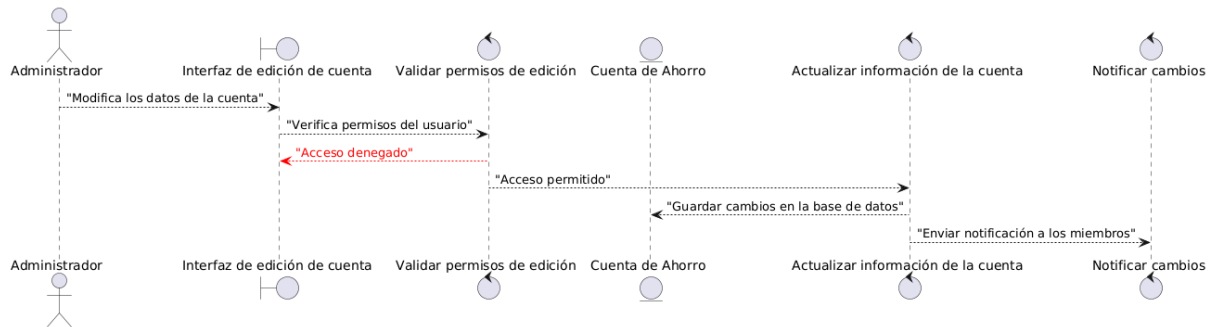
17. Dar de baja a usuario de la cuenta grupal

Descripción del Flujo:

- El usuario envía una solicitud para retirarse de la cuenta grupal.
- El sistema verifica si el usuario cumple con las condiciones (ej. no tener transacciones pendientes).
 - a. Si no cumple, se devuelve un mensaje de error.
 - b. Si cumple, se registra la solicitud.
- Se actualiza el estado de la cuenta grupal, reflejando la salida del usuario.
- Se notifica al administrador para su revisión y posible aprobación.



18. Editar datos de la cuenta ahorros



19. Enviar invitación a usuario a la cuenta de ahorros grupal

Descripción del Flujo:

- El administrador accede a la interfaz de la cuenta grupal y selecciona la opción de invitar a un usuario.
- El sistema valida que el usuario sea elegible (ej. que no esté ya en la cuenta, que su correo sea válido, etc.).
 - a. Si el usuario no es válido, se muestra un mensaje de error.
 - b. Si el usuario es válido, se continúa con la invitación.
- El sistema registra la invitación y la asocia a la cuenta grupal.
- Se envía una notificación al usuario invitado, informando sobre la invitación para unirse a la cuenta grupal.

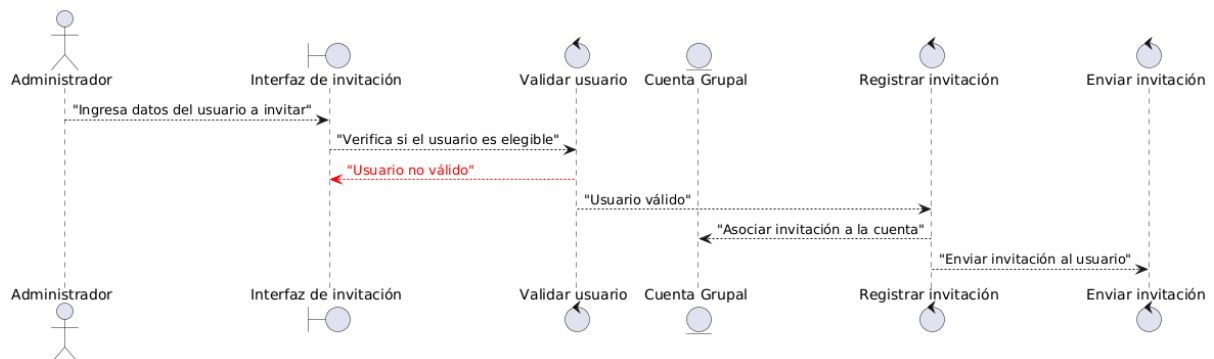


Diagrama de Secuencia

El diagrama de secuencia representa la interacción entre los diferentes actores del sistema y los componentes principales durante la ejecución de los casos de uso más relevantes. Este diagrama permite visualizar el flujo de información y las llamadas entre los distintos módulos, asegurando una comprensión clara del comportamiento del sistema en tiempo de ejecución.

Descripción del Flujo General

1. Registro e Inicio de Sesión

- El usuario ingresa sus datos de registro.
- El sistema verifica si el usuario ya existe en la base de datos.
- Si no está registrado, se crea una nueva cuenta y se envía una confirmación.
- Para iniciar sesión, el usuario ingresa sus credenciales y el sistema las valida antes de otorgar acceso.

2. Creación de Cuenta de Ahorro

- El usuario ingresa los datos de la cuenta que desea crear.
- El sistema valida la información y la disponibilidad.
- Si los datos son correctos, la cuenta se registra en la base de datos y se envía una notificación al usuario confirmando su creación.

3. Gestión de Cuentas Grupales

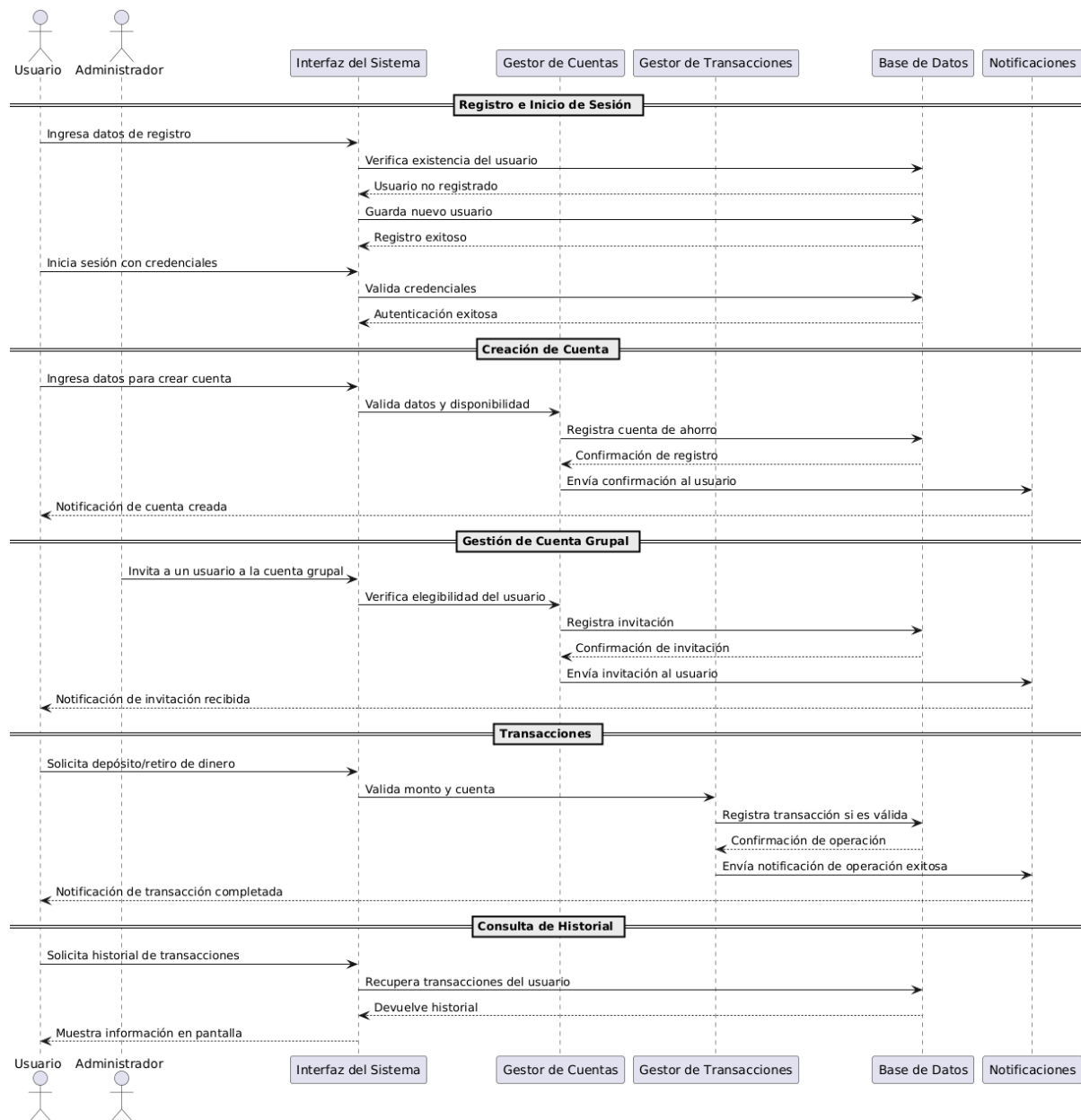
- Un administrador invita a un usuario a participar en una cuenta de ahorro grupal.
- El sistema verifica si el usuario es elegible y registra la invitación en la base de datos.
- Se envía una notificación al usuario para informarle sobre la invitación.

4. Transacciones (Depósitos y Retiros)

- El usuario solicita realizar una transacción (depósito o retiro).
- El sistema valida que el monto sea correcto y que la cuenta tenga saldo suficiente (en caso de retiros).
- Si la transacción es válida, se registra en la base de datos y se actualiza el saldo de la cuenta.
- Finalmente, el usuario recibe una notificación de la operación completada.

5. Consulta del Historial de Transacciones

- El usuario solicita ver el historial de transacciones de su cuenta.
- El sistema consulta la base de datos y recupera la información correspondiente.
- Los datos son enviados a la interfaz del usuario, donde se presentan de manera clara y organizada.



A lo largo del desarrollo del proyecto, se siguió una metodología estructurada basada en ICONIX, lo que permitió definir de manera clara y precisa los distintos elementos del sistema.

El proceso inició con la identificación de los **casos de uso**, estableciendo las funcionalidades principales del sistema y las interacciones de los actores con la aplicación. Posteriormente, los **diagramas de robustez** nos ayudaron a modelar la relación entre la interfaz, los controles y las entidades del sistema, asegurando un flujo lógico y validado en cada funcionalidad.

Con base en estos modelos previos, se elaboraron los **diagramas de secuencia**, los cuales permitieron visualizar la interacción dinámica entre los componentes, asegurando una correcta comunicación entre los módulos del sistema. Finalmente, toda esta información se consolidó en el **diagrama de clases**, el cual representa la estructura definitiva del sistema y sus relaciones.

El **diagrama de clases final** es el resultado de un análisis detallado de los requerimientos del sistema, reflejando de manera estructurada las entidades principales, sus atributos, métodos y asociaciones. Se incluyeron elementos clave como:

