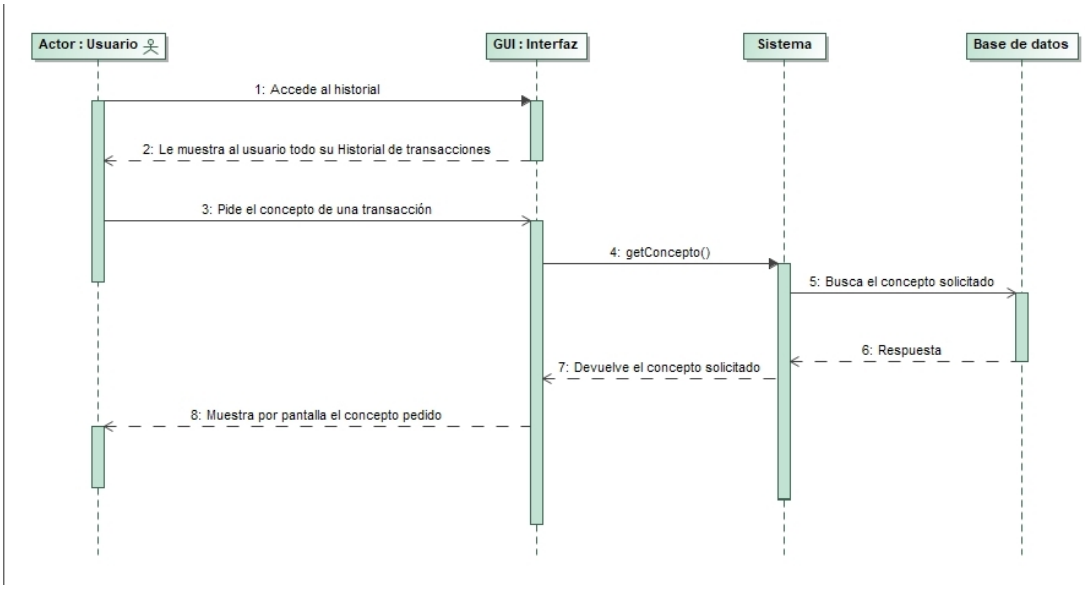
Diagramas de Secuencia - Gualet

Revisión: 1.0

Junio 1, 2021

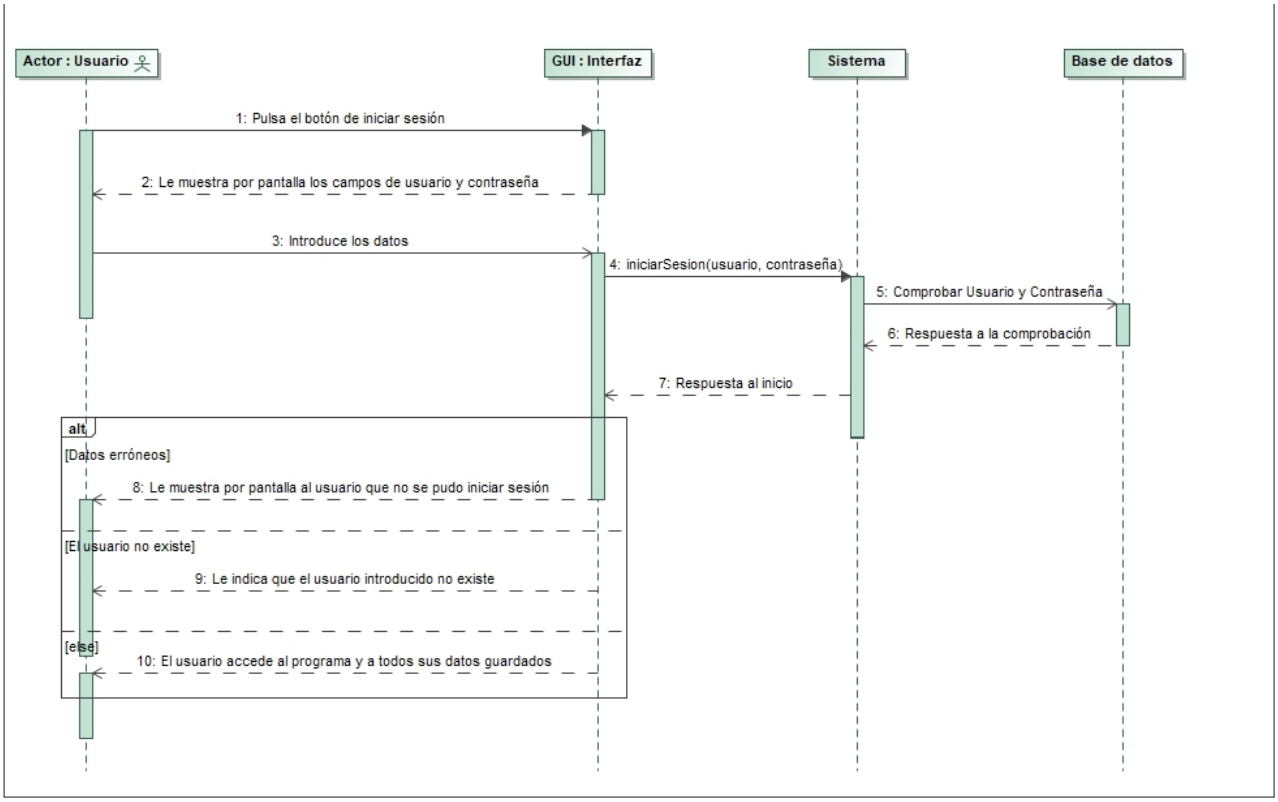
CONSEGUIR EL CONCEPTO



getConcepto() : void

Tenemos el actor Usuario quien es el que comienza toda la historia del método, accediendo al historial y luego a una transacción en específico a la que solicita su concepto, se le manda al Sistema la ejecución del método *getConcepto()*, donde buscará en la Base de Datos y devolverá al Usuario lo pedido.

INICIAR SESIÓN

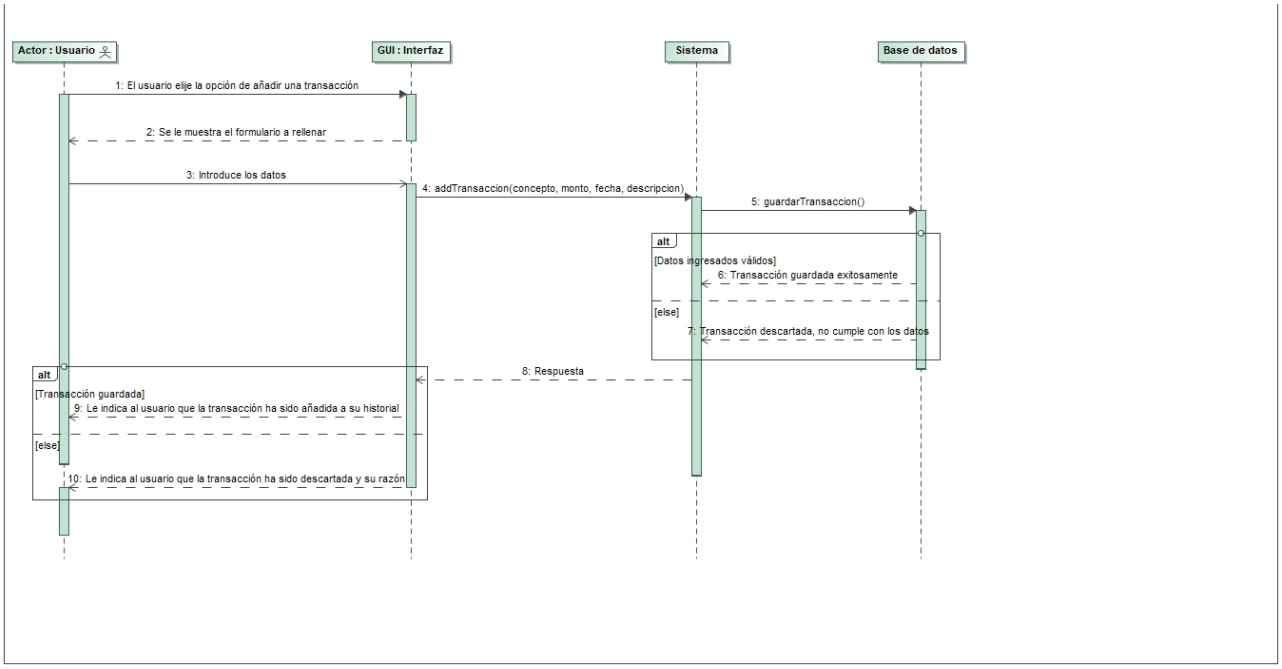


iniciarSesion(nombre : String, contrasena : String) : void

Método indispensable para nuestro sistema de ahorros, necesitarás acceder y guardar de forma fiable todos tus datos bancarios o monetarios.

Para ello el actor Usuario primero deberá pedir iniciar sesión y rellenar los datos necesarios con ayuda de la interfaz gráfica, donde se le mandará la orden iniciarSesion(usuario, contrasena) al Sistema y comprobará en la Base de Datos la validez de lo introducido, existiendo tres posibles salidas o respuestas para el Usuario.

AÑADIR TRANSACCIÓN



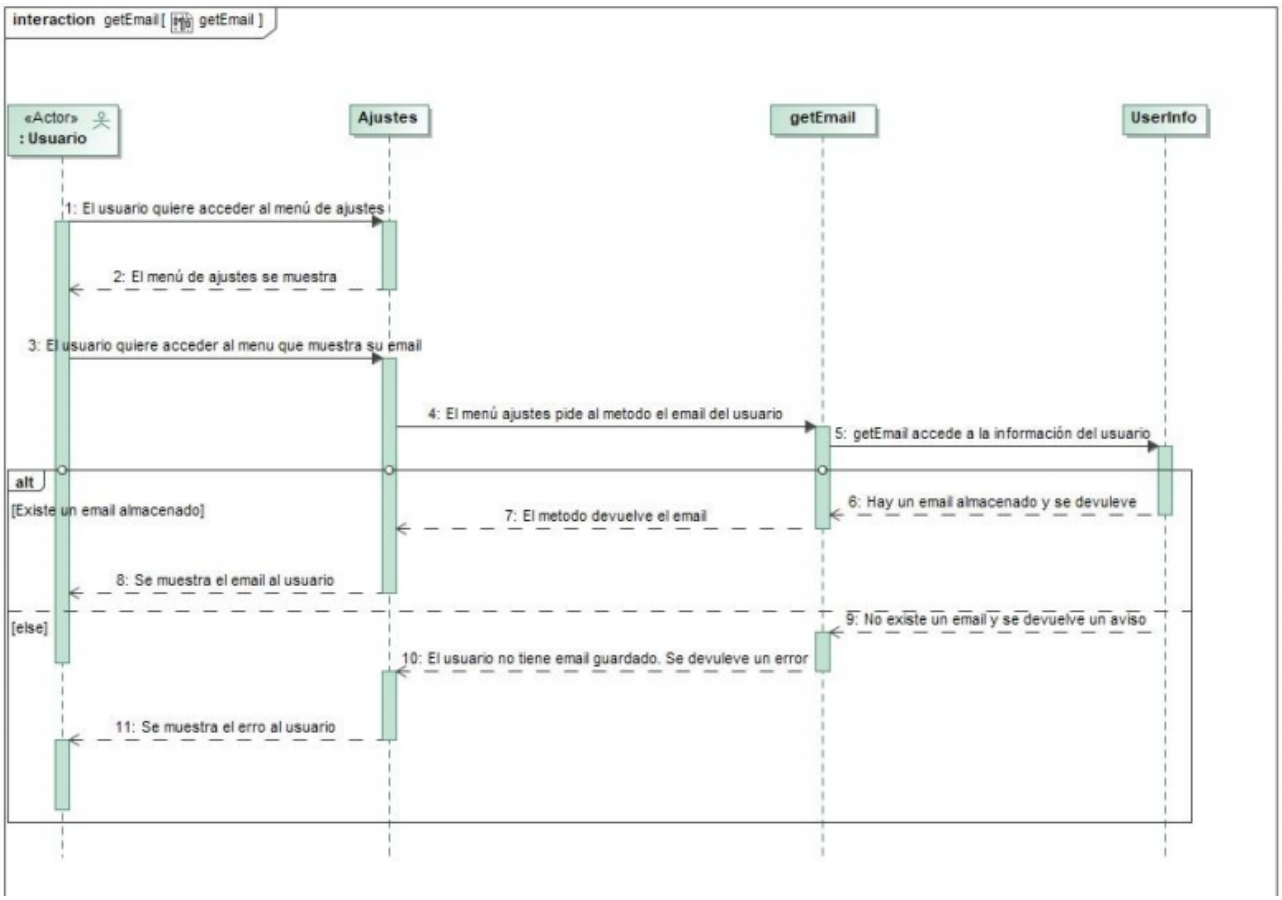
addTransaccion(concepto : String, monto : double, fecha : Date, descripcion : String) : void

Como bien hemos discutido, el programa requiere de conocer todas las transacciones realizadas para dar la mejor ayuda posible. Por ello agregar todas las compras o gastos es fundamental.

El Usuario deberá introducir los datos en el formulario con ayuda de la GUI, lo que le dictará al sistema que active el método addTransaccion(concepto, monto, fecha, descripción), lo que activará a su vez el método guardarTransaccion(transaccion) y procederá a salvar en la Base de Datos la información que insertamos, indicándole al Usuario si se realizo con éxito o no.

Cabe mencionar la posibilidad de otras salidas como que por ejemplo el Usuario no introdujera una descripción, en ese caso se guardaría con éxito la transacción dado que nuestro programa cuenta con dos constructores para la clase Transaccion, uno con descripcion y otro sin ella. Sin embargo los otros datos son indispensables y una vez puestos no se pueden cambiar, solo borrar por completo la transacción (esto para evitar trampas a la hora de mantener un historial veraz).

CONSEGUIR CORREO



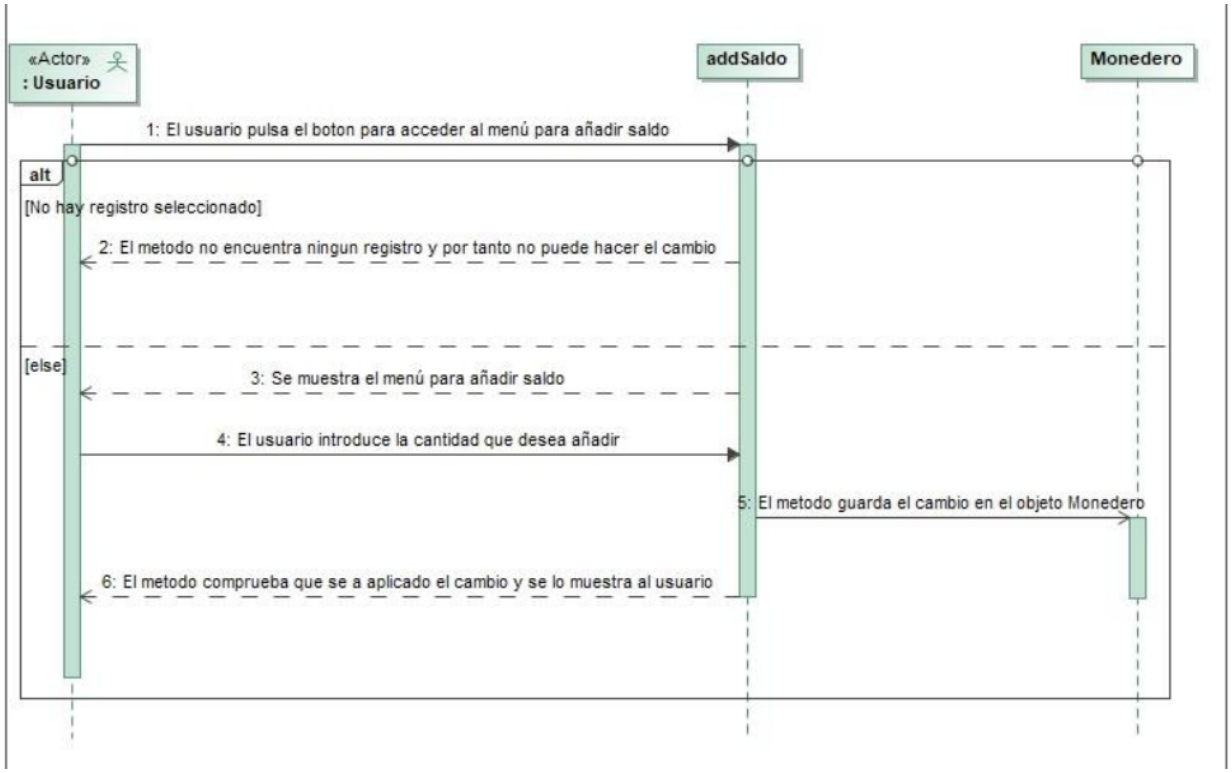
getEmail() : void

Este diagrama muestra una posible interacción entre el usuario y la aplicación para usar el método getEmail.

Para llegar a este método el usuario debe acceder primero al menú de ajuste donde navegará hasta que se muestre la opción para acceder al email con el que se ha iniciado sesión.

Una vez encuentre la opción de mostrar el email y envíe la petición, el menú de ajuste ejecutará el método getEmail que accederá a la información guardada del usuario. En el caso de que exista se devolverá el email hasta que se muestre finalmente al usuario. En caso de que no exista un email almacenado, se le mostrará al usuario un mensaje de que no se ha guardado un email para ese usuario.

AÑADIR SALDO



addSaldo(saldo : double) : void

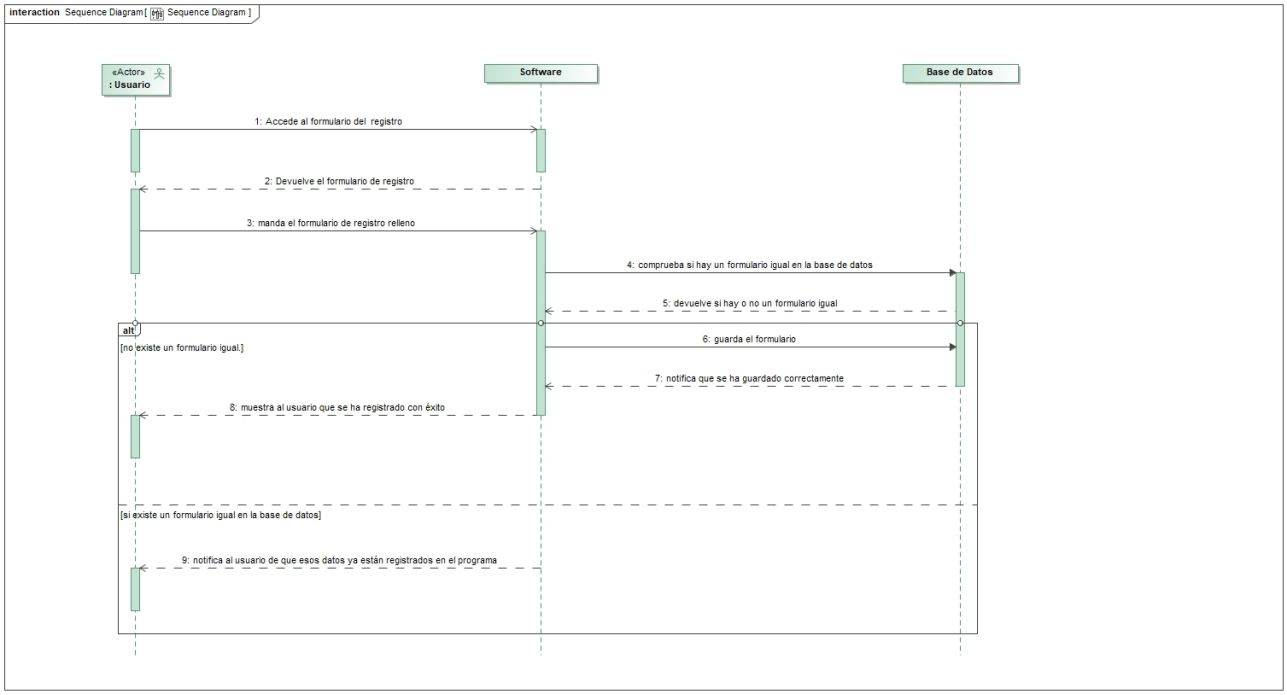
Este diagrama muestra una posible interacción entre addSaldo y el usuario.

En primer lugar, el usuario debe interactuar con el botón que le sugiere añadir saldo a la cuenta que tiene seleccionada en ese momento. Si no ha seleccionado ningún registro, se le devolverá un mensaje de error pidiéndole que lo haga.

Una vez pulse el botón habiendo seleccionado un registro, se le dirigirá a un menú donde podrá introducir el saldo que desea añadir, una vez confirme la cantidad a añadir, esta será usada por el método addSaldo que lo añadirá a los datos guardados del registro (monedero) y el método addSaldo se encargará de comprobar que esta operación se haya realizado de forma correcta.

Una vez se haya realizado la operación, se devolverá un mensaje con el nuevo saldo disponible en el registro.

REGISTRAR UN USUARIO



registrarUsuario(user : String, contra : String, email : String) : void

En un primer paso, el usuario indicará al software su deseo de registrarse para tener una cuenta en el programa, seleccionando la opción de ‘registrarse’, ya que el usuario podrá cancelar o hacer otra cosa si este lo desea y sin necesidad de esperar a una respuesta, manda el mensaje como asíncrono.

El software recibe el mensaje e instantáneamente manda al usuario el formulario que debe de rellenar para poder registrarse. Siendo una respuesta a la petición del usuario.
El usuario rellenará el formulario y lo enviará al software de nuevo pero relleno. Una vez más como mensaje asíncrono.

El software recibirá el formulario, y con los datos que este tiene enviará un mensaje síncrono a la base de datos para comprobar si esos datos ya existen en esta. El software se quedará esperando a que la base de datos le responda.
La base de datos buscará esos datos y responderá al software o bien diciéndole que ya hay una fila que contiene dichos datos o que no hay.

Si hubiera una fila que contuviera esos datos, eso significaría que ya están registrados los datos en el programa, luego no se podrán registrar de nuevo. Dicho esto el software responderá al usuario notificándole que ya existe tal usuario.

Si no hubiera ninguna fila con dichos datos, eso significaría que no existe el usuario todavía, luego el software mandaría un mensaje a la base de datos para que esta guardara los datos en una nueva fila, creando así el usuario. El software esperaría hasta que la base de datos le confirmara el guardado. La base de datos responderá al software cuando guarde los datos, y el software respondería finalmente al usuario con un mensaje que le notificara que se registró con éxito.

VINCULAR CUENTA BANCARIA

El objetivo del usuario será vincular su cuenta bancaria con el software. El usuario selecciona la opción de vincular cuenta bancaria, y el software le responde mostrándole al usuario el formulario con los datos a rellenar para poder vincular su cuenta bancaria.

Una vez relleno el formulario, el usuario seleccionará en el botón ‘enviar’, enviándole así el formulario relleno al software. El software consultará a la base de datos si dispone de los datos introducidos en el formulario que le dio el usuario. La base de datos buscará entre sus filas y le devolverá al software si se encuentran dichos datos o no. Dependiendo de la respuesta que le dé, el software hará una cosa u otra: Si ya existen en la base de datos, el software notificará al usuario que los datos introducidos ya se están usando mediante un mensaje mostrado en pantalla.

Esto significará que el usuario no podrá vincular dicha cuenta o que ya la ha vinculado con anterioridad, en ambos casos, no se hará nada más. Pero si no existe, el software se comunicará con el banco mandándole un mensaje para consultar si los datos ofrecidos por el cliente son correctos, y el banco responderá al software si lo son o no.

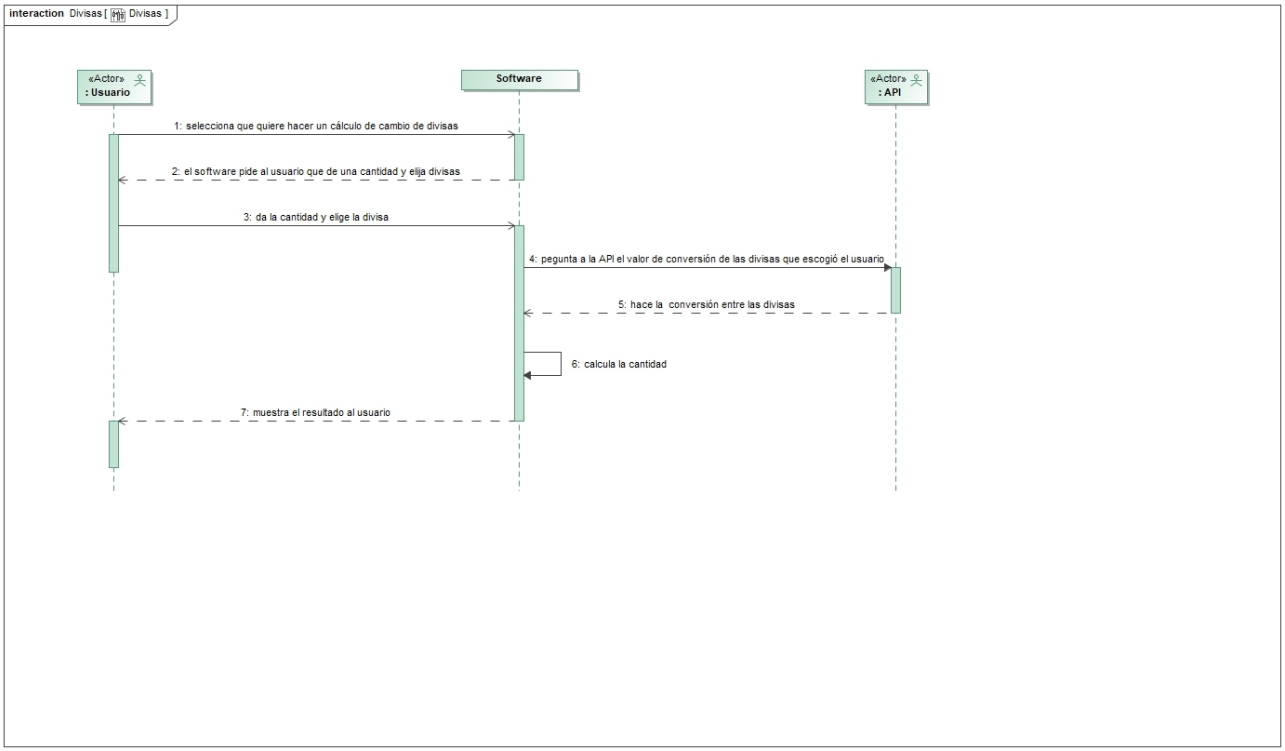
Si los datos son incorrectos, el software comunicará al usuario que los datos que ha otorgado no son correctos, y este deberá repetir el proceso si desea o no.

Si los datos son correctos, el usuario volverá a mandar un mensaje al banco para conseguir el permiso y vinculación de la cuenta bancaria dada por el usuario en el formulario, a lo que el banco responderá al software con la vinculación aceptada o denegada.

Si es denegada, el software informará al usuario de que ha ocurrido un problema con la vinculación de su cuenta bancaria, y debe revisarlo con su banco.

Si es aceptada, el software informará al usuario de que la vinculación se ha realizado con éxito. Ambos se mostrarán al usuario con un mensaje por pantalla.

CALCULAR UNA DIVISA

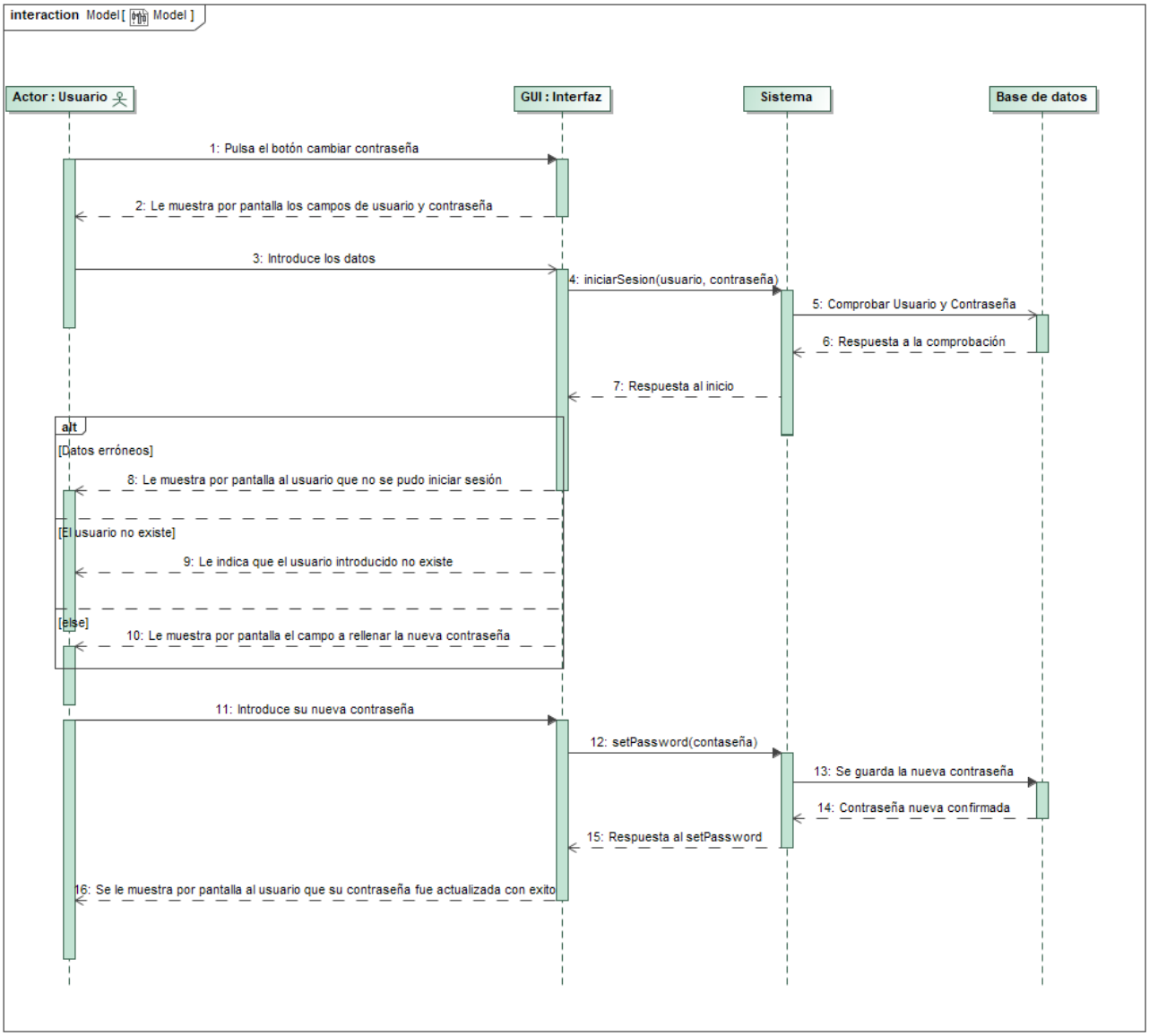


En este tercer y último diagrama nos encontramos con dos actores (el usuario y la API) y un componente (el software). El objetivo del usuario será saber cuánto vale ‘x’ cantidad de ‘y’ divisa en otra divisa ‘z’.

El usuario seleccionará la opción ‘Cambio de divisas’ , mandándole así un mensaje al software que este responderá mostrándole al usuario un pequeño formulario donde pueda introducir la cantidad que desea pasar a otra divisa y seleccionar la divisa de inicio y la final. Una vez introducido los datos el usuario seleccionará el botón de ‘calcular’.

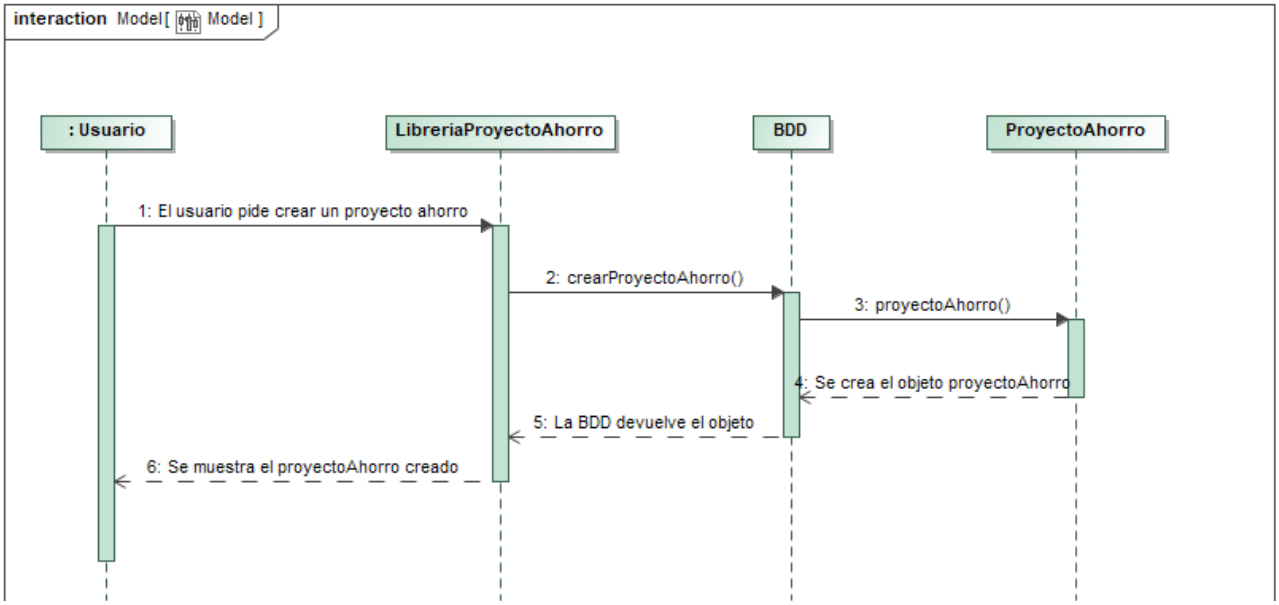
Con estos datos el software calculará la cantidad introducida por el usuario con la divisa final que introdujo y guardará el resultado durante un periodo de tiempo. El software entonces mostrará al usuario el resultado de la operación.

ESTABLECER CONTRASEÑA



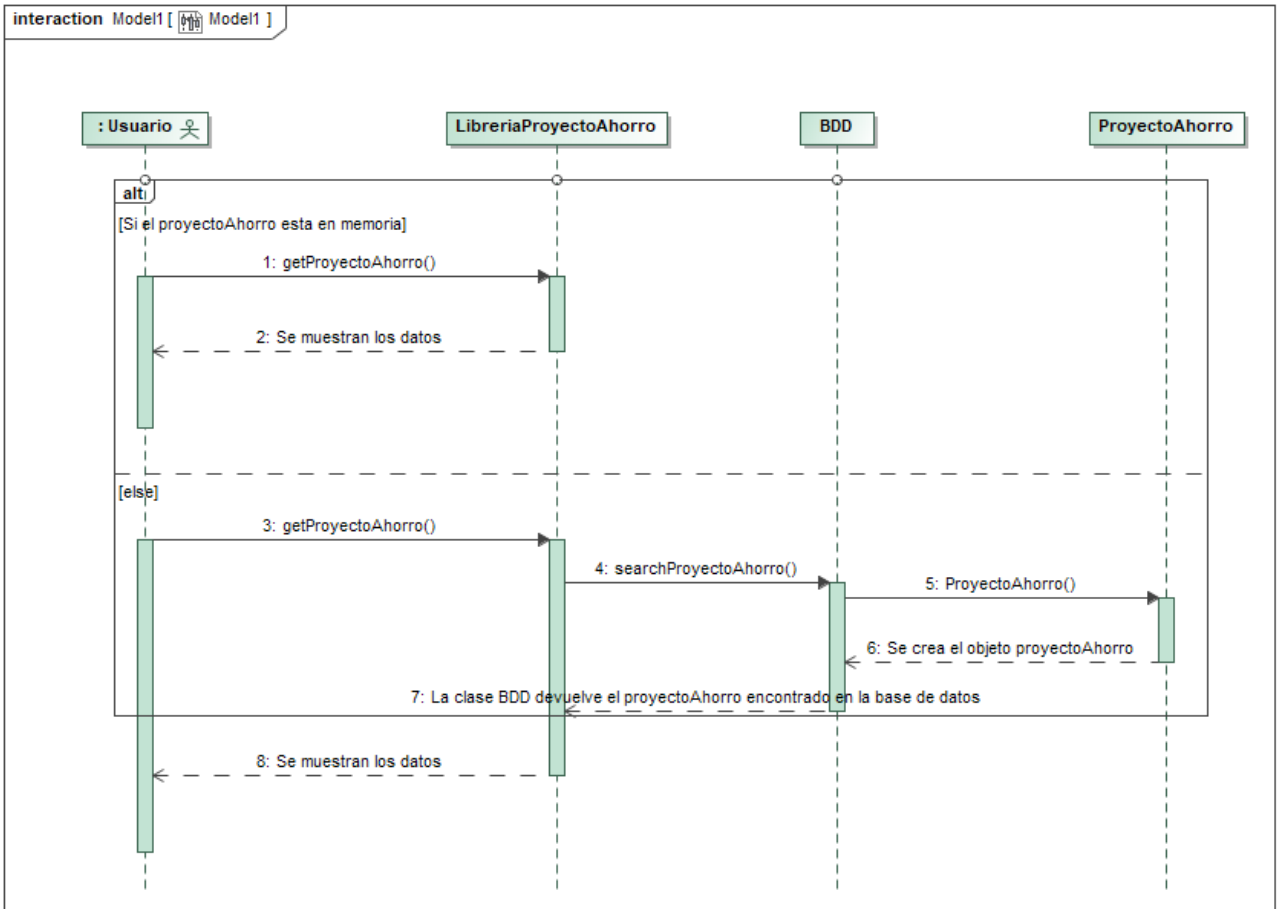
setPassword(contra : String) : void

CREAR UN PROYECTO AHORRO



crearProyectoAhorro(nombre : String, objetivo : double) : void

CONSEGUIR UN PROYECTO AHORRO



getProyectoAhorro(nombre : String) : void