***UNIVERSIDAD NACIONAL DE ASUNCIÓN***

***“FACULTAD POLITÉCNICA”***

******

**Primer Examen Final - Grupo 17**

* Abel Moisés Díaz Barrios
* Elías Sebastián Gill Quintana
* Oswald Rodrigo Alvarenga Coronel
* Andrés Moisés Román Medina

**Integrantes:**

**Asignatura:** Lenguaje de Programación 2

**Docente:** Rodrigo Manuel Velázquez Galeano

**Sección:** TQ

**INGENIERÍA EN INFORMÁTICA**

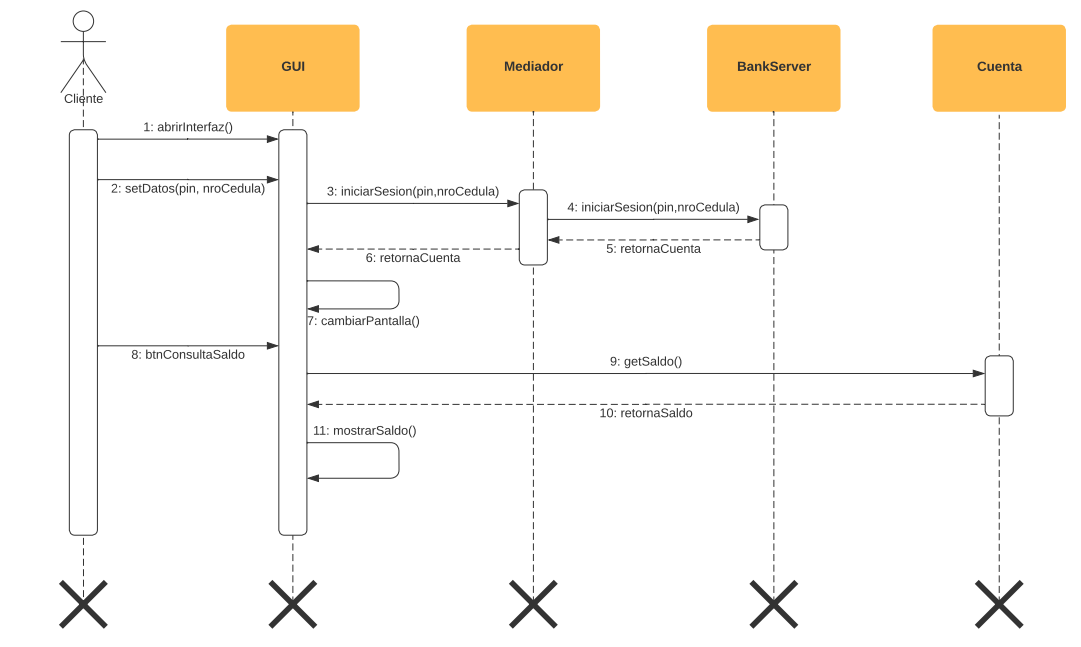
*Tercer Semestre*

*SAN LORENZO – PARAGUAY*

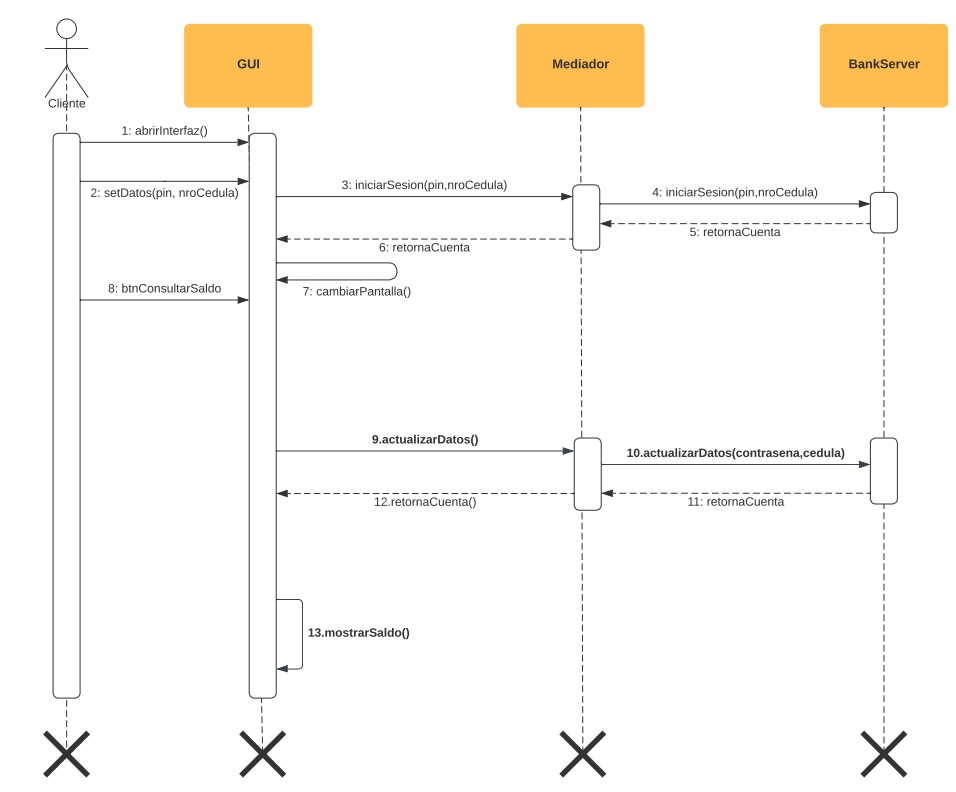
*2022*

**Cambios en los diagramas de consulta de saldo.**

Antes:



Después:



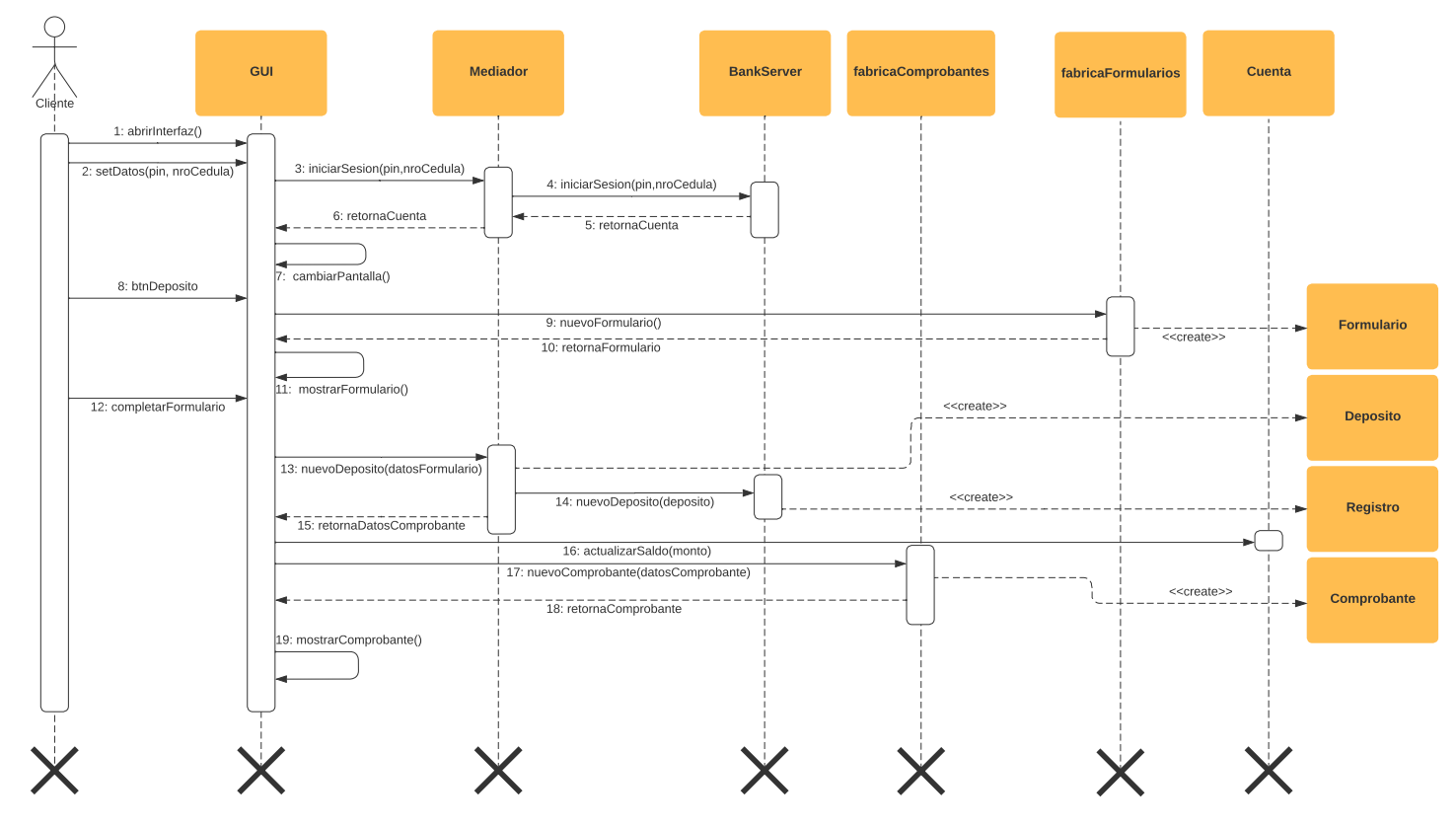
Resumen de cambios:

* El método abrirInterfaz() ahora se llama btnInicioMouseClicked(),
* El botón de consultarSaldo es el botón de refresh que se encuntra en la parte superior de la GUI.
* La clase que se encarga de retornar el saldo es la clase mediador(a través de su variable de clase CuentaActiva), no la clase Cuenta.
* No existe un método de mostrarSaldo() como tal, el metodo del botón se encarga de todo a través de la clase mediador.

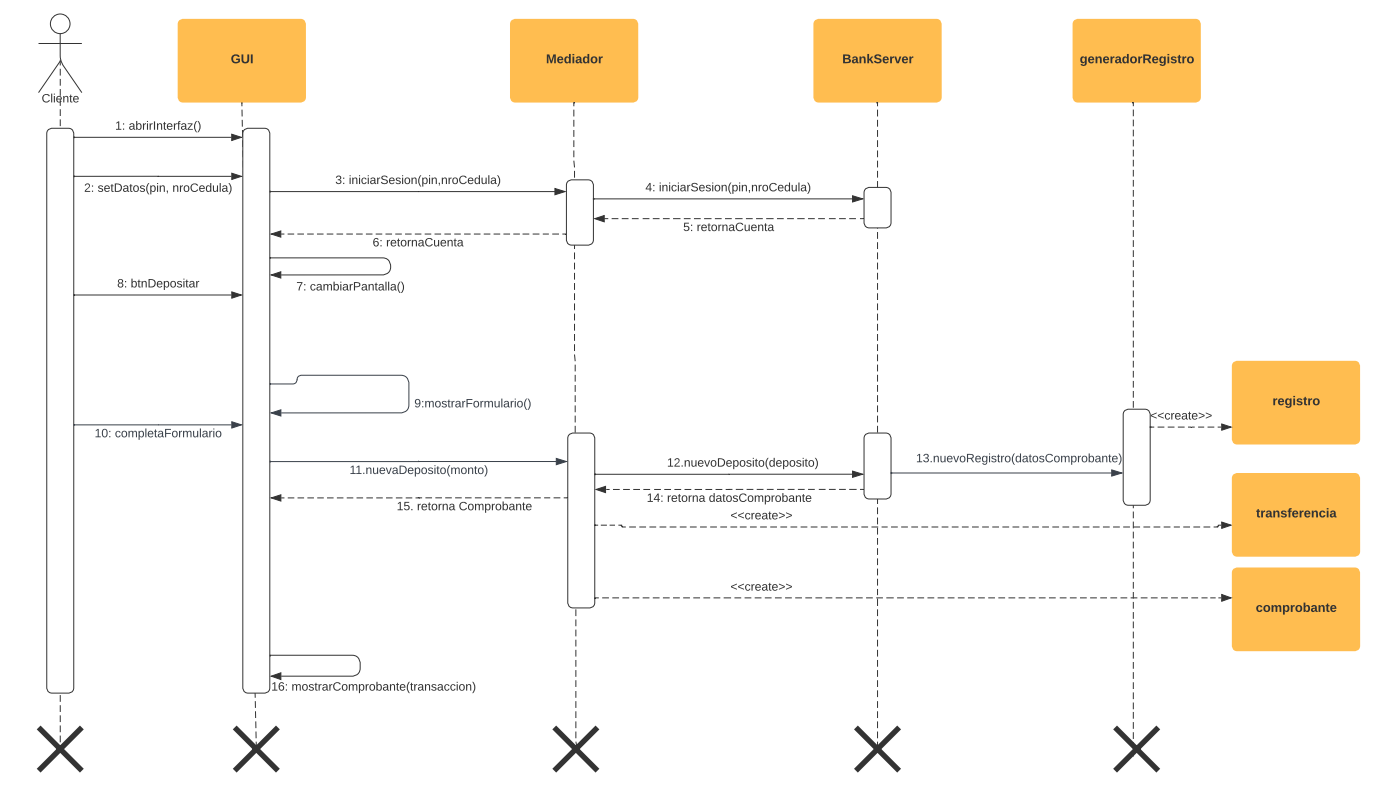


**Cambios en el diagrama de secuencia de deposito**

Antes:



Después:

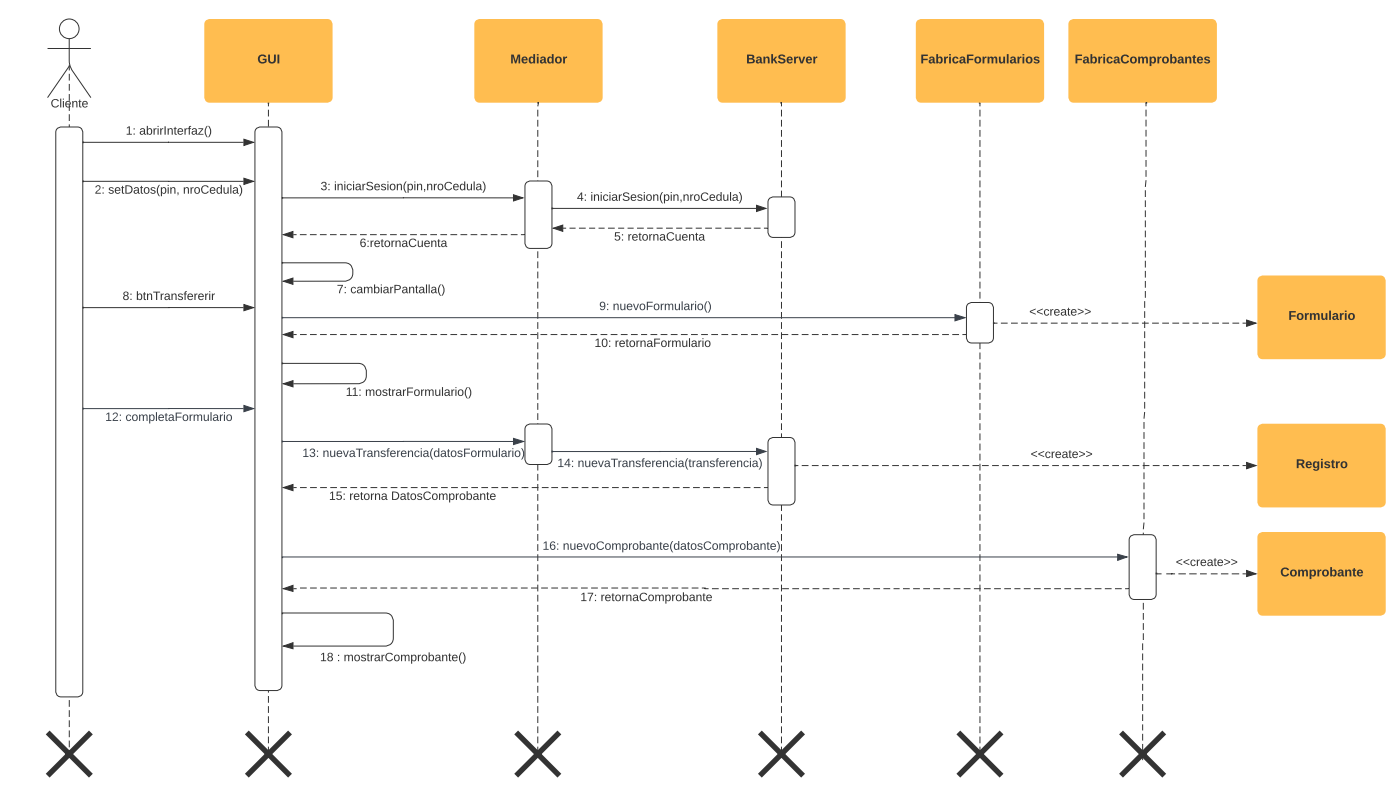


Resumen de cambios:

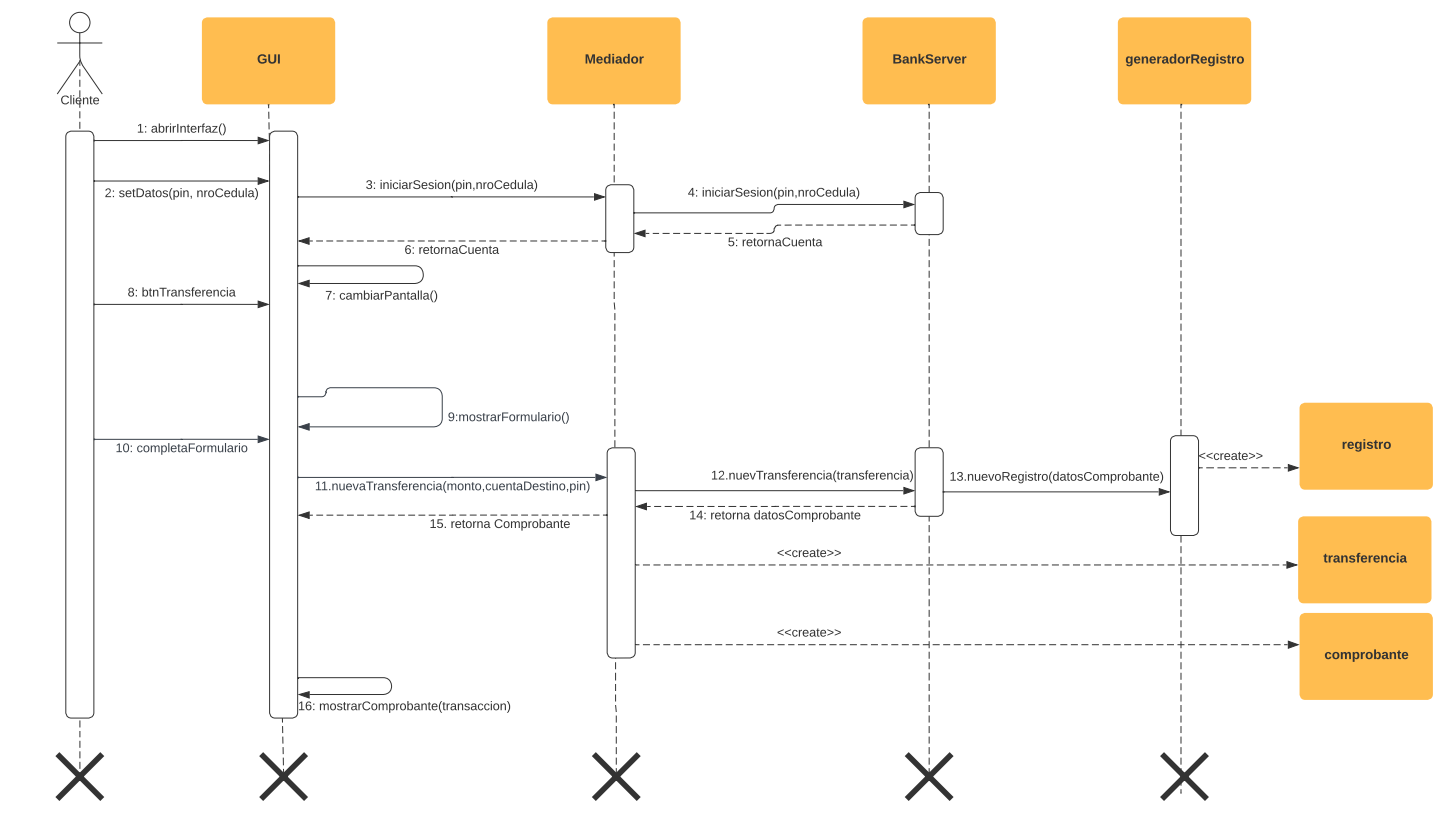
* No utilizamos una clase de “fabricaFormularios” para crear el nuevo formulario, sino que con el evento del botón presionado ya se hace.
* No utilizamos una clase de “fabricaComprobantes” para crear el nuevo formulario, sino que la clase mediador es el que crea y retorna el comprobante.
* El Bank server no retorna un “Registro”, retorna datos del comprobante.
* El Bank server es el que modifica el saldo de la cuenta del usuario, no la clase “Usuario”.
* El que se encarga de crear el registro es la clase generadorRegistro
* El mediador crea el objeto transferencia

**Cambios en el diagrama de transferencia entre cuentas**

Antes:



Después:

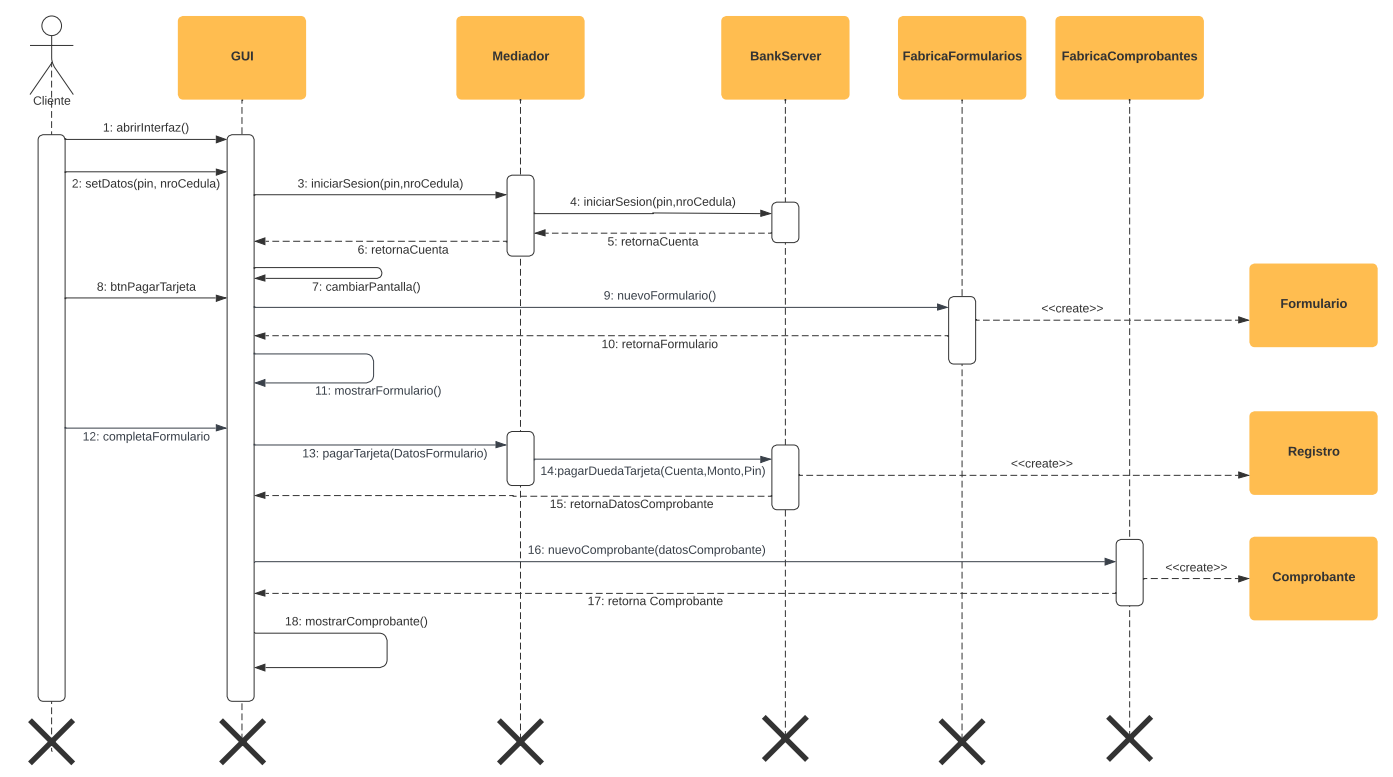


Resumen de cambios:

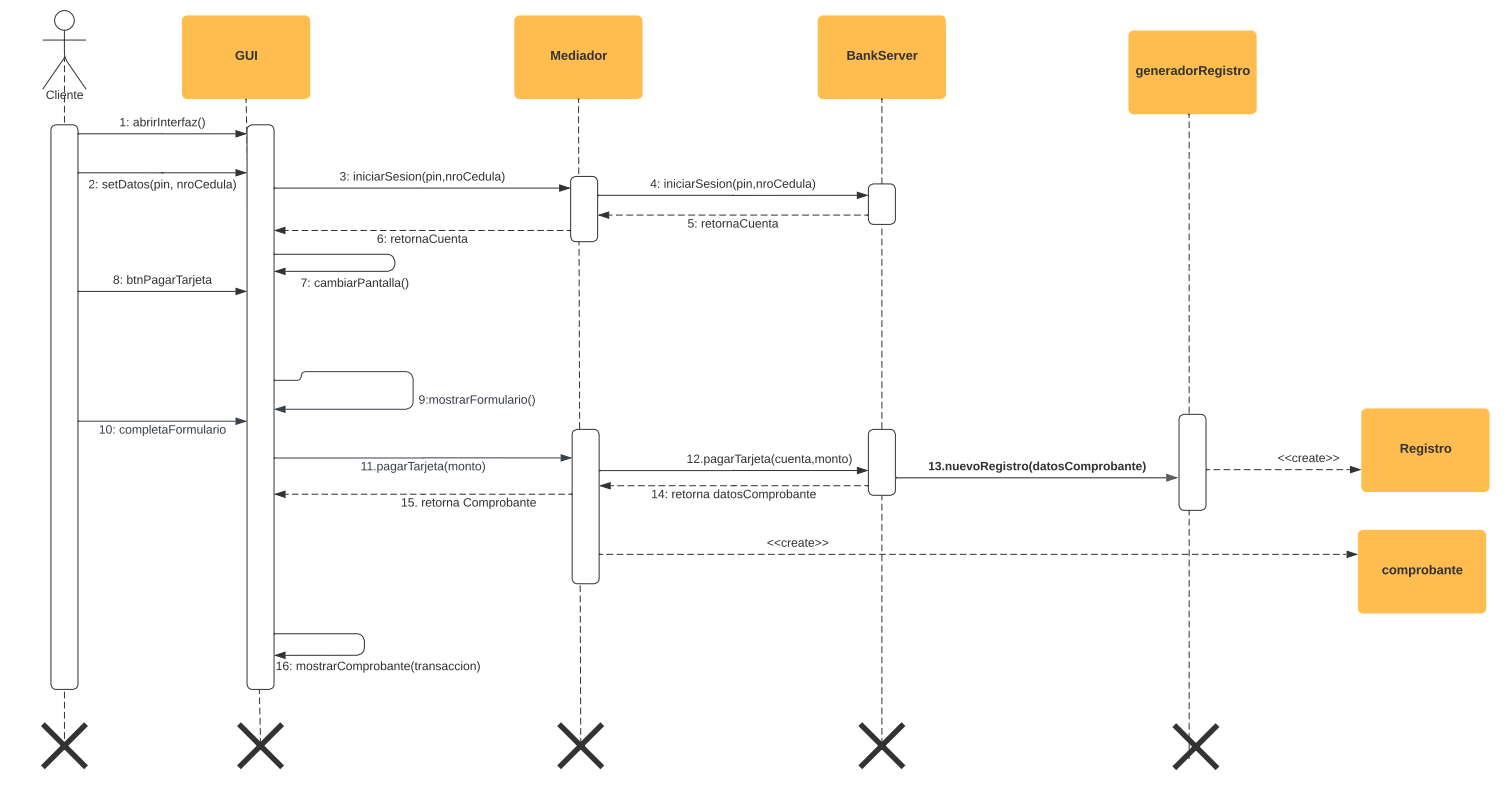
* No utilizamos una clase de “fabricaFormularios” para crear el nuevo formulario, sino que con el evento del botón presionado ya se hace.
* No utilizamos una clase de “fabricaComprobantes” para crear el nuevo formulario, sino que la clase mediador es el que crea y retorna el comprobante.
* El Bank server no retorna un “Registro”, retorna datos del comprobante.
* El que se encarga de crear el registro es la clase “generadorRegistro”.
* El mediador crea el un objeto transferencia que pasara el método “nuevaTransferencia” del bankserver.
* Los métodos de la clase “bankServer” y la clase “mediador” reciben atributos en este nuevo diagrama.

**Cambios en el diagrama de secuencia de pago tarjetas de crédito**

Antes:



Después:

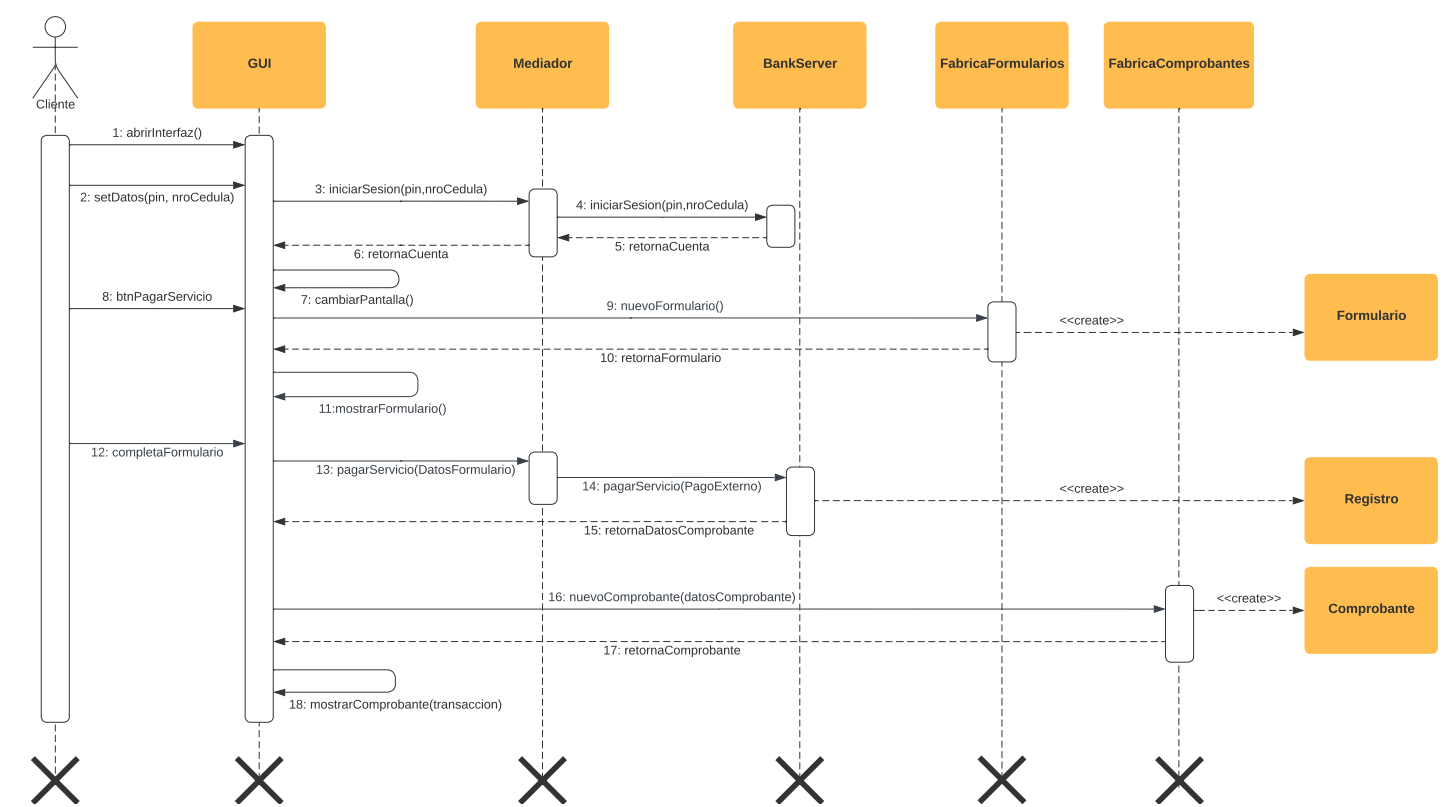


Resumen de cambios:

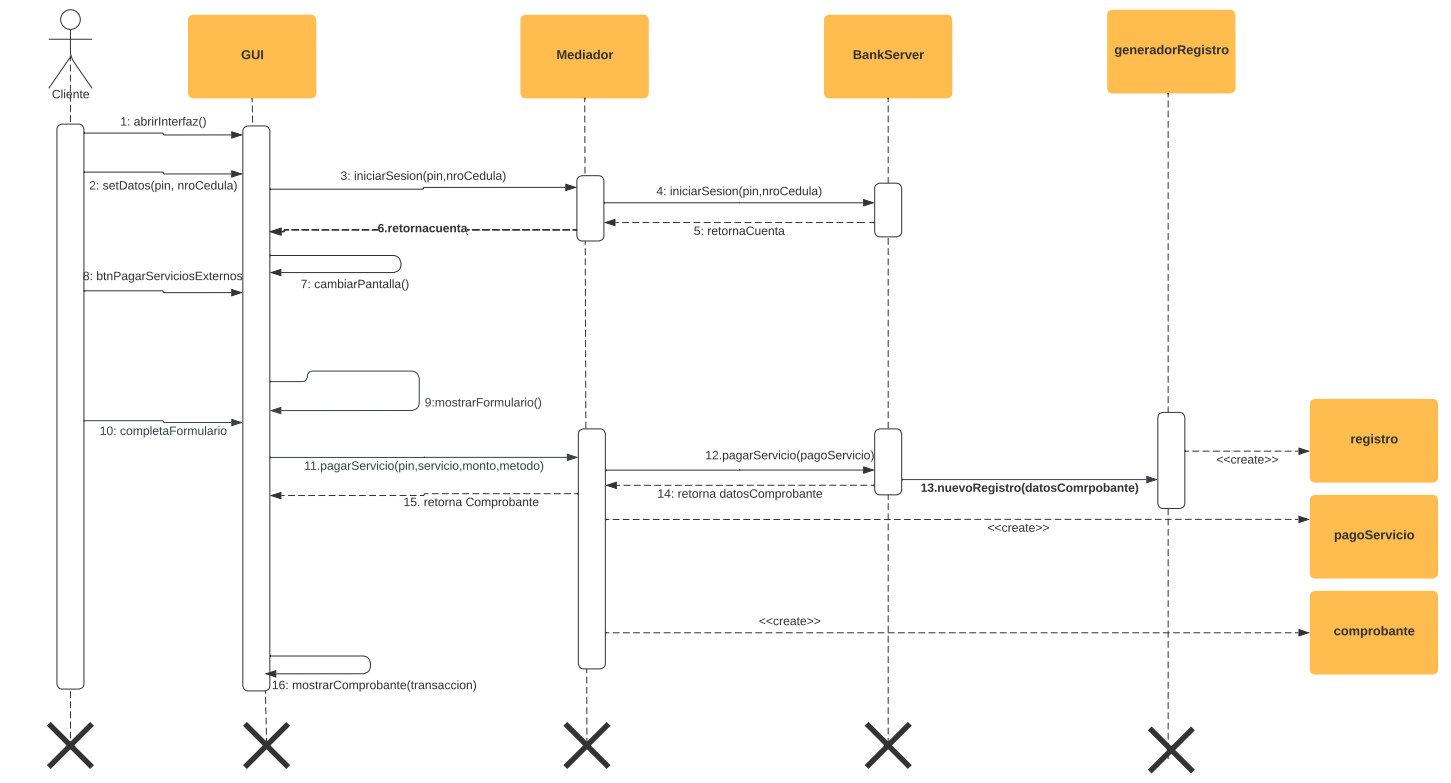
* No utilizamos una clase de “fabricaFormularios” para crear el nuevo formulario, sino que con el evento del botón presionado ya se hace.
* No utilizamos una clase de “fabricaComprobantes” para crear el nuevo formulario, sino que la clase mediador es el que crea y retorna el comprobante.
* El Bank server no retorna un “Registro”, retorna datos del comprobante.
* El que se encarga de crear el registro es la clase “generadorRegistro”.
* Los métodos de la clase “bankServer” y la clase “mediador” reciben atributos en este nuevo diagrama.

**Cambios en el diagrama de secuencia del caso de uso pago de servicios**

Antes:



Después:

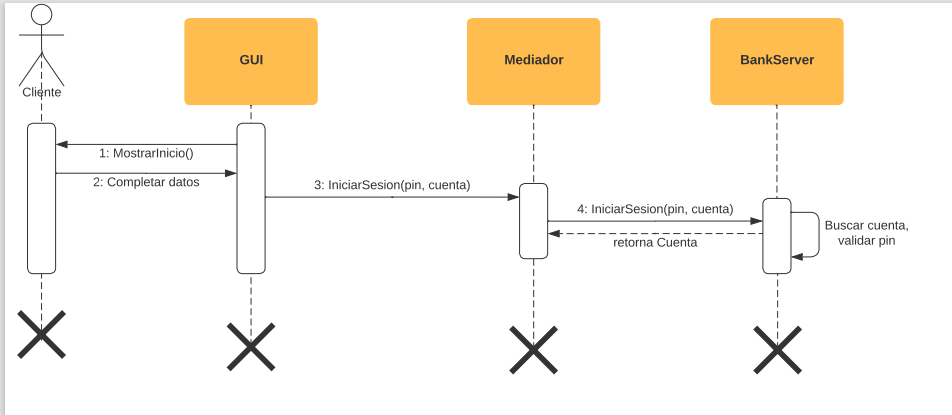


Resumen de cambios:

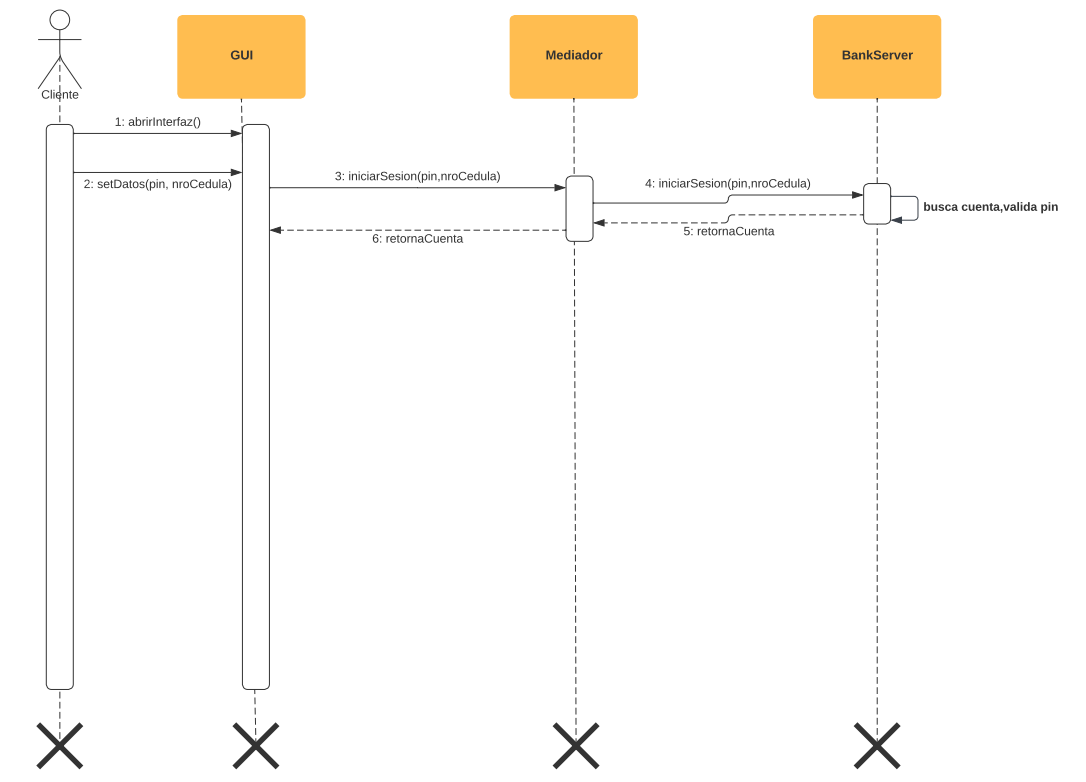
* No utilizamos una clase de “fabricaFormularios” para crear el nuevo formulario, sino que con el evento del botón presionado ya se hace.
* No utilizamos una clase de “fabricaComprobantes” para crear el nuevo formulario, sino que la clase mediador es el que crea y retorna el comprobante.
* El Bank server no retorna un “Registro”, retorna datos del comprobante.
* El que se encarga de crear el registro es la clase “generadorRegistro”.
* Los métodos de la clase “bankServer” y la clase “mediador” reciben atributos en este nuevo diagrama.

**Cambios en el diagrama de secuencia del caso validación del pin de cuenta**

Antes:



Después:

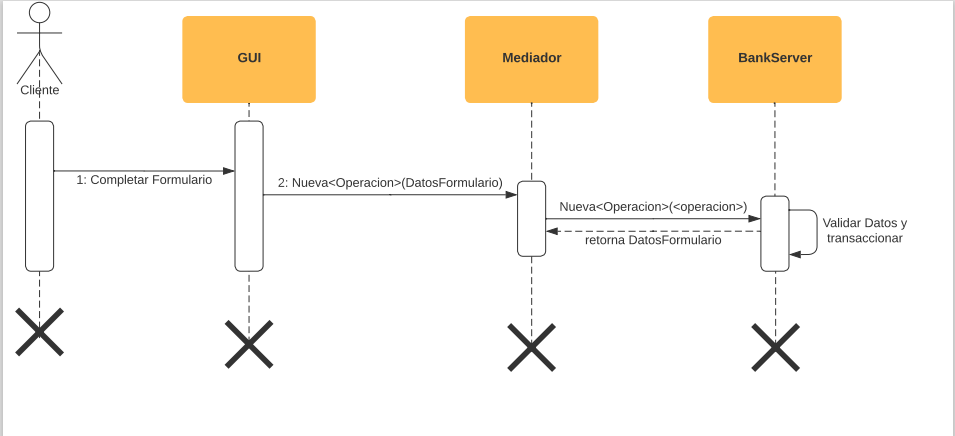


Resumen de cambios:

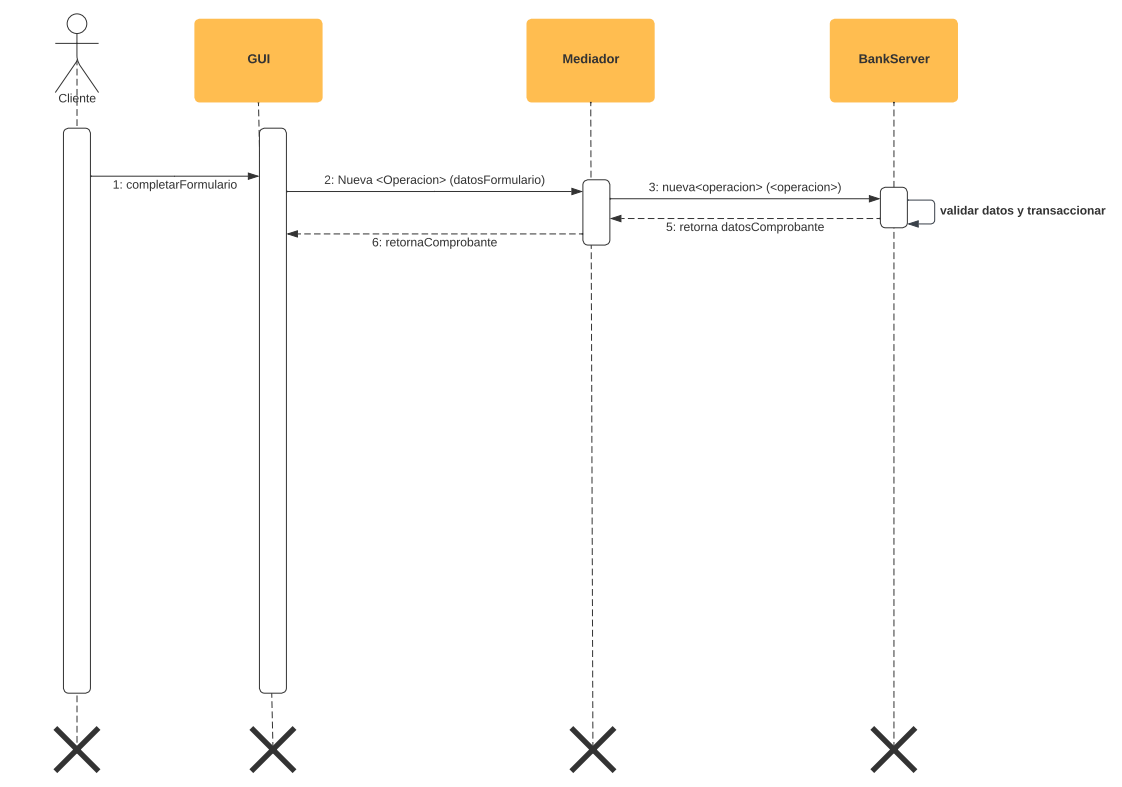
* La clase mediador retorna también la cuenta

**Cambios en el diagrama de secuencia del caso de uso validación del pin de transacción**

Antes:



Después:



Resumen de cambios:

* La clase mediador retorna también la cuenta

**Cambios en el Esquema de Bases de Datos**

**Entidad Cuenta**

En la entidad de la tabla cuenta se han agregado los campos deuda\_Tarjeta, saldo\_Base\_Tarjeta y saldo\_Tarjeta así también se han suplantado los campos saldo\_Disponible y tarjeta.

**Cambios en el Diagrama de Clases**

Las siguientes clases fueron implementadas con sus respectivos métodos

Se implementó la clase generador registro, la clase consultas SQL con los siguientes métodos:

* ObtenerCuentaCedula
* ObtenerCuentaNro
* setSaldo
* PagarTarjeta

ServerServicios (metodos)

* getServicioNombre
* getServicioId
* PagoServicios
* DatosComprobante
* Intercambiar a paneles