

Memoria Proyecto IS2

GRUPO 2

Ramón Rosa Pérez
Bin Chen
Shuyi Wang

1) Índice

- Introducción
- Contexto
- Planificación
- Modelo entidad-relación
- Diagramas
- Código fuente y explicación de la estructura
- Ejecución y demo
- Extras y pruebas
- Bibliografía

2) Introducción

El proyecto consiste en una aplicación de escritorio de gestión para una empresa de telecomunicaciones. Esta, permite la gestión básica de clientes por los empleados de la empresa, la gestión de empleados por los empleados supervisores (administradores), la gestión de tarifas existentes en la empresa y por último permite la gestión de tarifas contratadas por parte de los clientes.

El proceso consistió, tras tener claros los requisitos de la aplicación, en el diseño del modelo del dominio y la base de datos además de su creación en mysql utilizando phpmyadmin, después se realizó el diseño de aplicación utilizando Modelio y por último se realizó el desarrollo del código utilizando Eclipse como editor y GitHub como sistema de control de versiones.

3) Contexto. Diseño, Implementación y Arquitectura

En la fase de diseño se empezó con el sistema de login y con el sistema de gestión de clientes siguiendo los requisitos especificados en la SRS de IS1. Más adelante se decidió hacer un cambio a los requisitos al tener nuevos integrantes el grupo y se añadieron un módulo de gestión de empleados, un módulo de gestión de tarifas y otro de gestión de contratos. Finalmente se descartó el módulo de creación de gráficas informativas.

En la fase de diseño se optó por el uso de los patrones Modelo Vista Controlador, Singleton y Abstract Factory. El patrón Modelo Vista Controlador nos aporta una mayor escalabilidad y una separación lógica en capas del software. El patrón Singleton decidimos utilizarlo porque nos era de gran utilidad en las clases que debían tener una única instancia como eran el controlador o las factorías. Finalmente, el patrón Abstract Factory ha sido muy útil ya que al tener varias clases para la GUI, varias clases de servicios de aplicación y varias clases de DAO, nos ayuda a instanciarlas sin necesidad de especificar sus clases concretas.

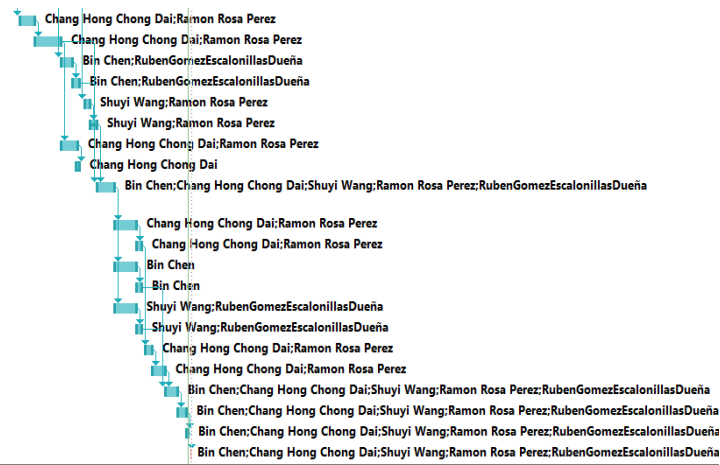
Como lenguaje de programación hemos utilizado Java ya que era conocido por todo el grupo y nos permitía aplicar los patrones de diseño de forma sencilla y finalmente se desarrollaron los módulos de gestión de clientes y gestión empleados. En el punto de código fuente de esta memoria se detalla más acerca de ello.

Por último, hemos utilizado una arquitectura multicapa, que divide la aplicación en la capa de presentación para interactuar con el usuario, la capa de negocio que contiene la lógica de la aplicación y la capa de integración que realiza la comunicación con la base de datos.

4) Planificación

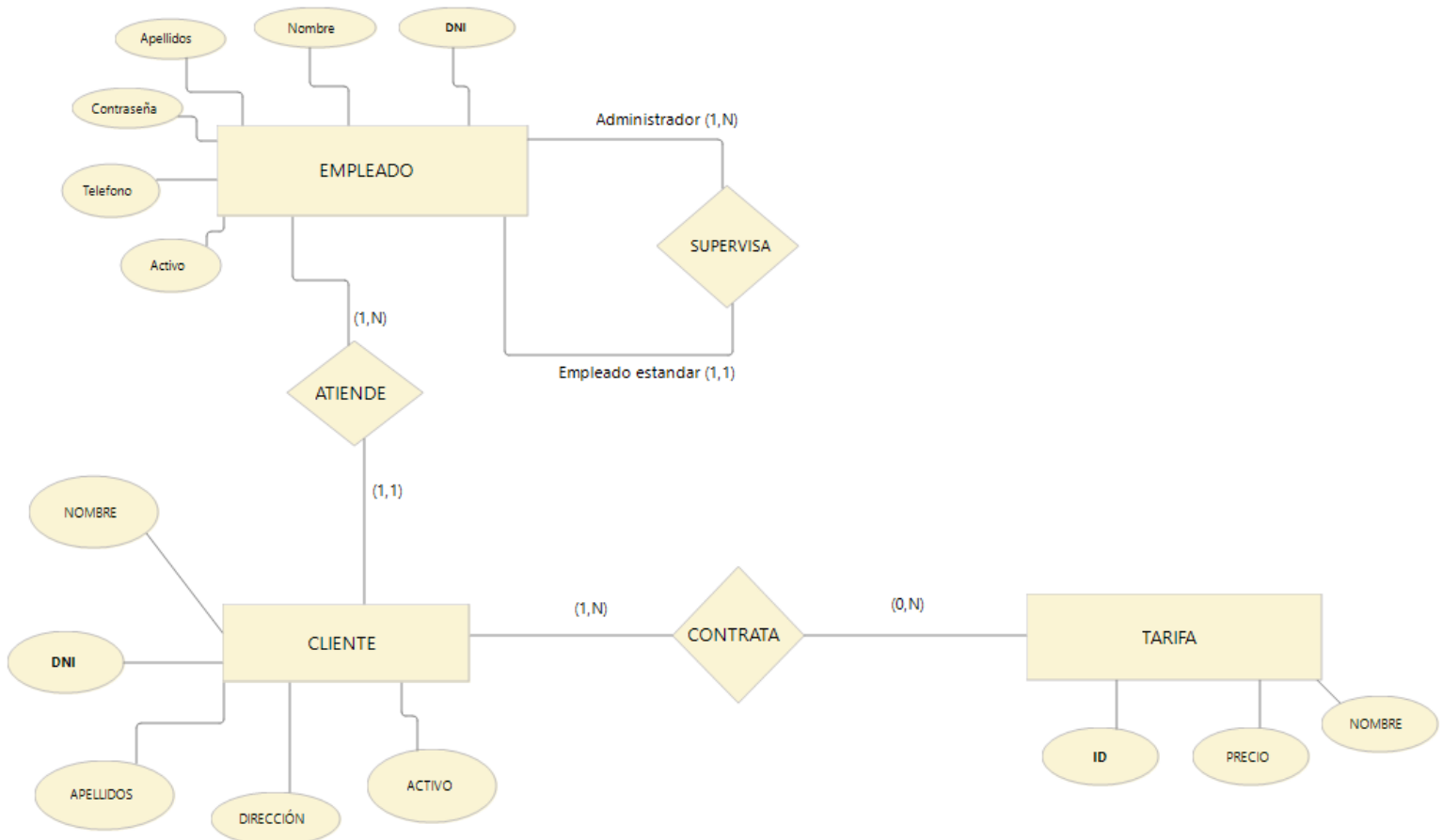
Diagrama de Gantt con la actualización de la planificación teniendo en cuenta fechas reales.

★	SistemaLogueo.Anal	6 días	lun 08/03/21	lun 15/03/21	7	Chang Hong Chong Dai
★	SistemaLogueo.Diseñ	11 días	mar 16/03/21	mar 30/03/21	8	Chang Hong Chong Dai
★	GestiónDatosCliente	4 días	mié 31/03/21	lun 05/04/21	7	Bin Chen;RubenGomez
★	GestiónDatosCliente	4 días	mar 06/04/21	vie 09/04/21	10	Bin Chen;RubenGomez
★	GestiónUsuario.Anal	3 días	lun 12/04/21	jue 15/04/21	7	Shuyi Wang;Ramon R
★	GestiónUsuario.Diseñ	2 días	vie 16/04/21	lun 19/04/21	12	Shuyi Wang;Ramon R
★	Otros.Analisi	7 días	mié 31/03/21	jue 08/04/21	9	Chang Hong Chong Dai
★	Otros.Diseño	2 días	jue 08/04/21	vie 09/04/21	14	Chang Hong Chong Dai
★	EvaluaciónCliente.Ar y Diseño	8 días	mar 20/04/21	jue 29/04/21	13;11;9	Bin Chen; Chang Hong Chong
★	SistemaLogueo.Codi	8 días	vie 30/04/21	mar 11/05/21	16	Chang Hong Chong Dai
★	SistemaLogueo.Prue	3 días	mié 12/05/21	vie 14/05/21	17	Chang Hong Chong Dai
★	GestiónDatosCliente	8 días	vie 30/04/21	mar 11/05/21	16	Bin Chen
★	GestiónDatosCliente	3 días	mié 12/05/21	vie 14/05/21	19	Bin Chen
★	GestiónUsuario.Codi	8 días	vie 30/04/21	mar 11/05/21	16	Shuyi Wang;RubenGomez
★	GestiónUsuarios.Prue	3 días	mié 12/05/21	vie 14/05/21	21	Shuyi Wang;RubenGomez
★	Otros.Codificación	4 días	lun 17/05/21	jue 20/05/21	18	Chang Hong Chong Dai
★	Otros.Prueba	5 días	vie 21/05/21	jue 27/05/21	23	Chang Hong Chong Dai
★	Programa.Emsablaje	5 días	vie 28/05/21	jue 03/06/21	20;22;24	Bin Chen;Chang Hong
★	Programa.Prueba	3 días	vie 04/06/21	mar 08/06/21	25	Bin Chen;Chang Hong
★	Programa.Instalación	1 día	mié 09/06/21	mié 09/06/21	26	Bin Chen;Chang Hong
★	Entrega	1 día	vie 11/06/21	vie 11/06/21	27	Bin Chen;Chang Hong



5) Modelo Entidad – Relación o modelo de Datos

Hemos optado por utilizar el atributo DNI de las entidades cliente y empleado como clave primaria en vez de un identificador autoincremental y por realizar un borrado lógico a través del atributo “Activo”. Además, hemos tenido en cuenta que un cliente es atendido por un empleado que se le asigna al darle de alta para que atienda todas sus consultas en vez de por uno distinto cada vez.

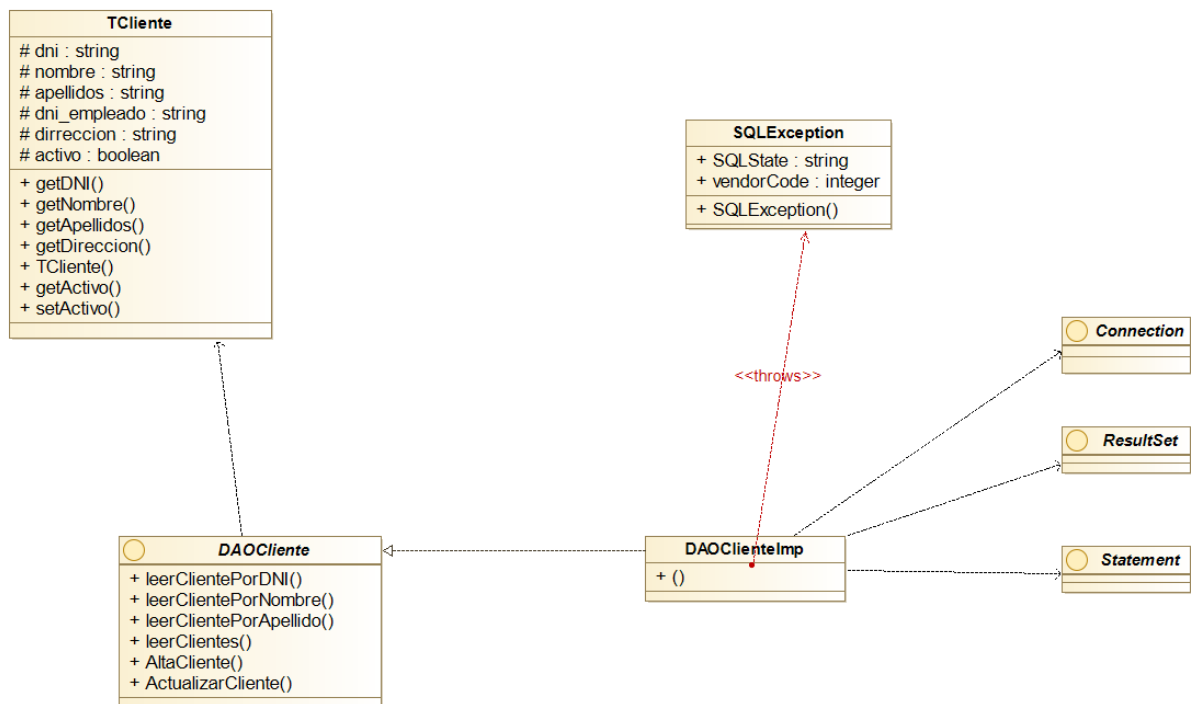


6) Diagramas

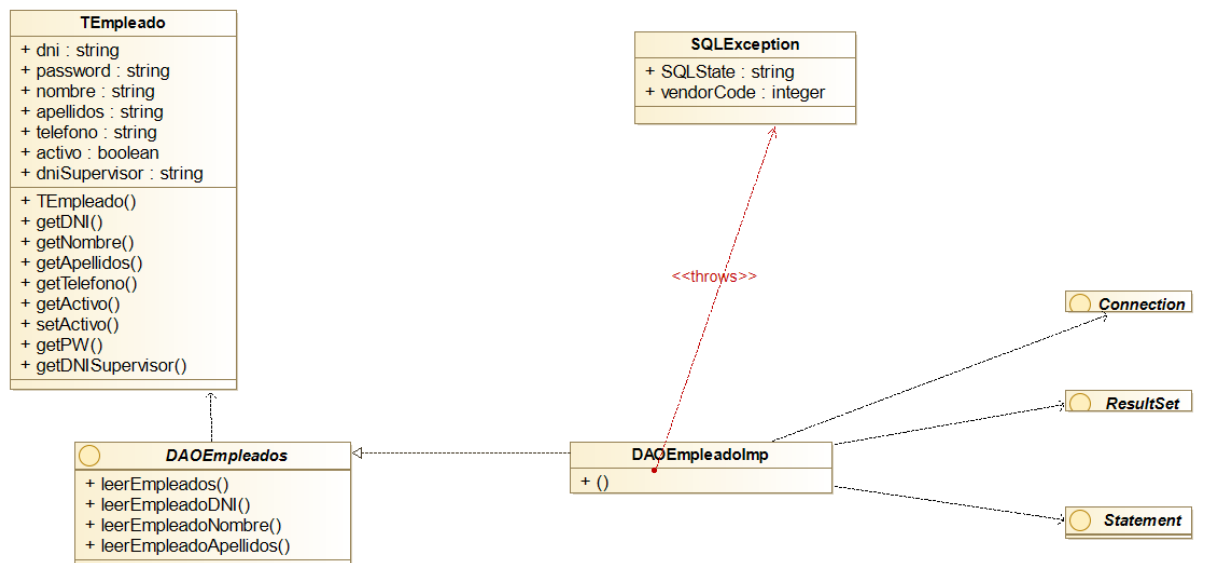
6.1 - De Clase

Integración

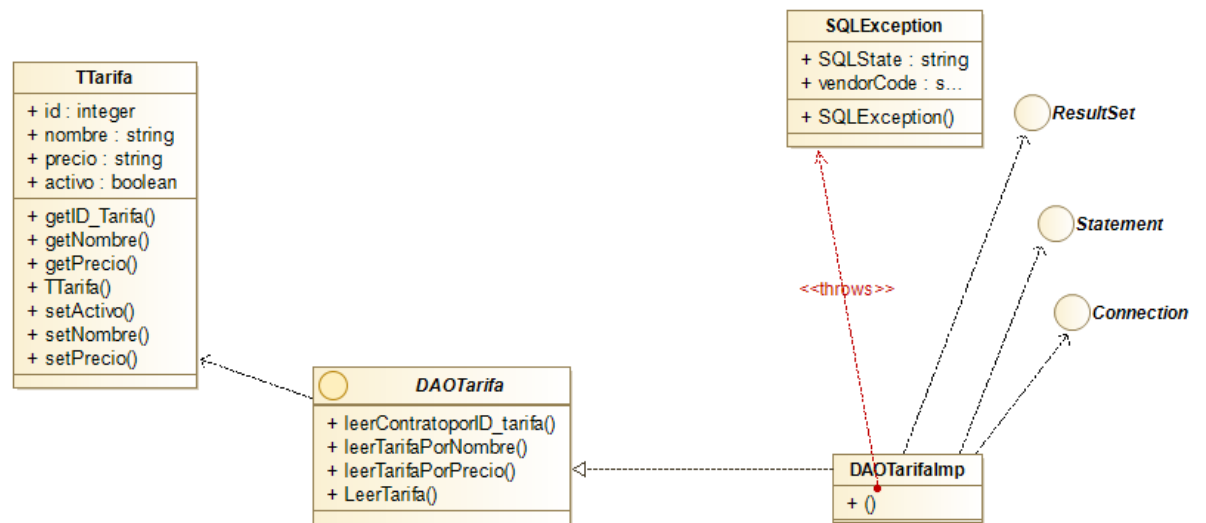
- Cliente



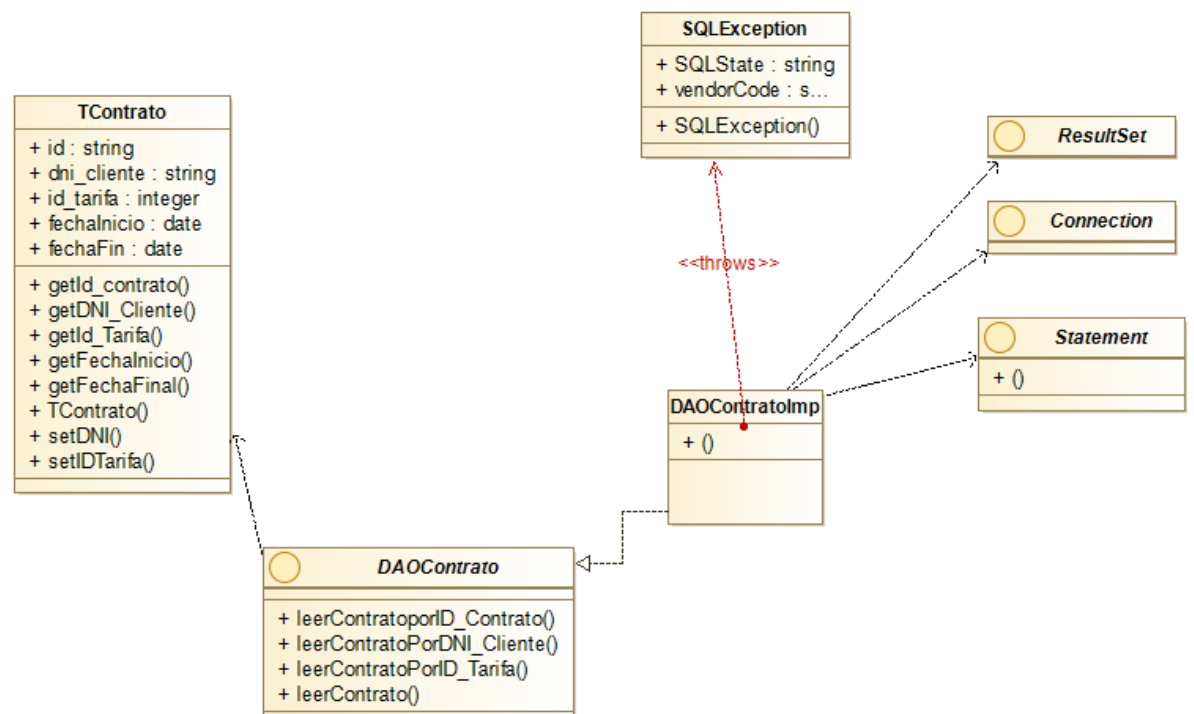
- Empleado



- Tarifa

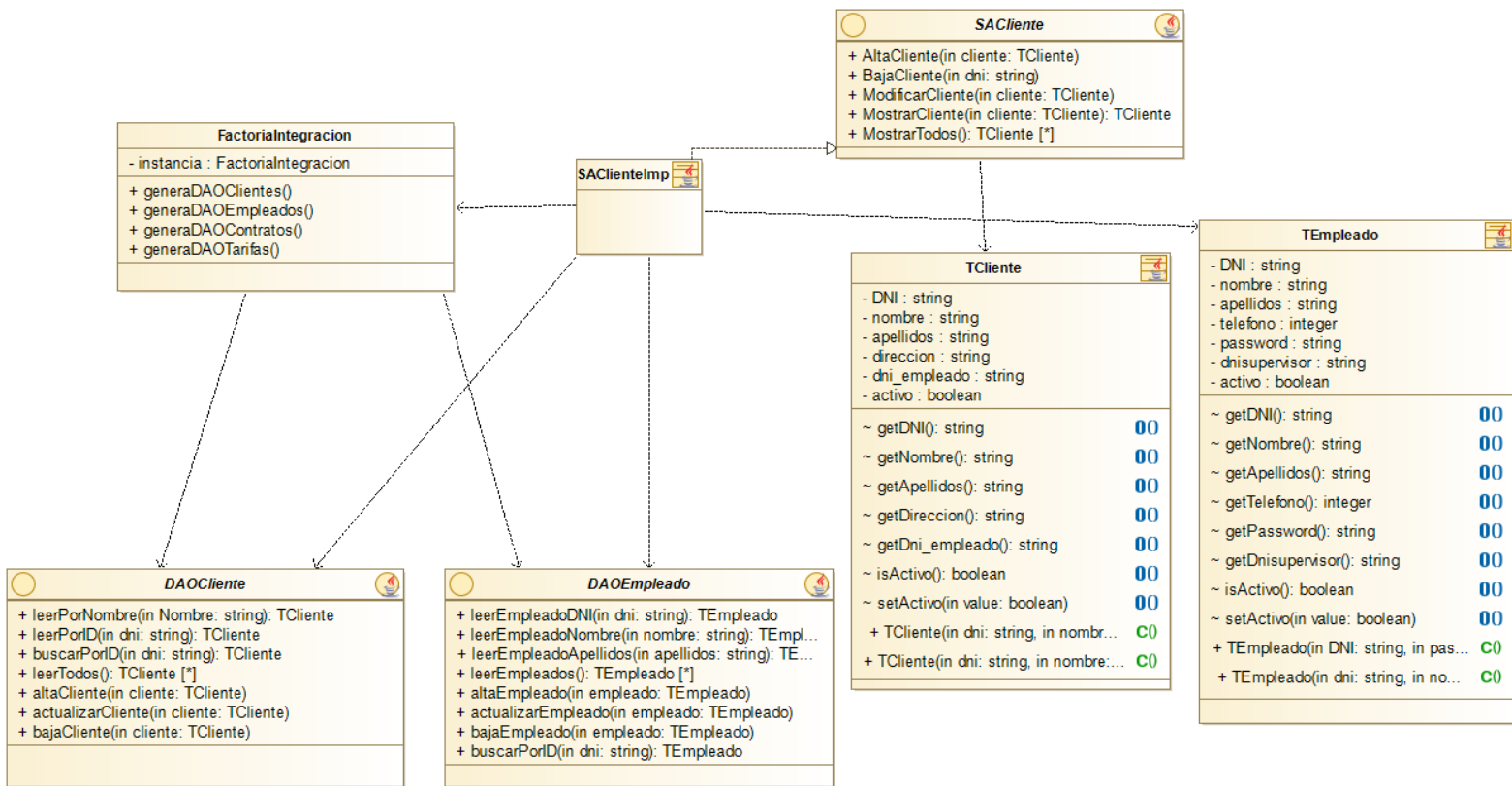


- Contrato

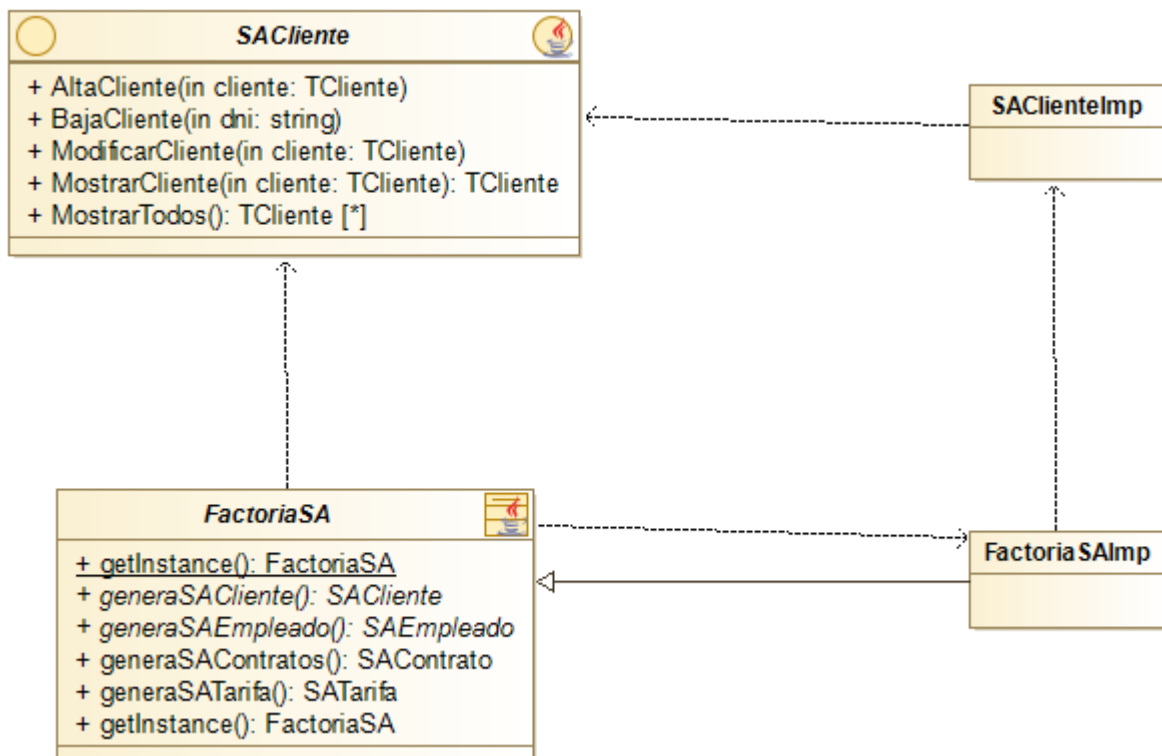


Negocio

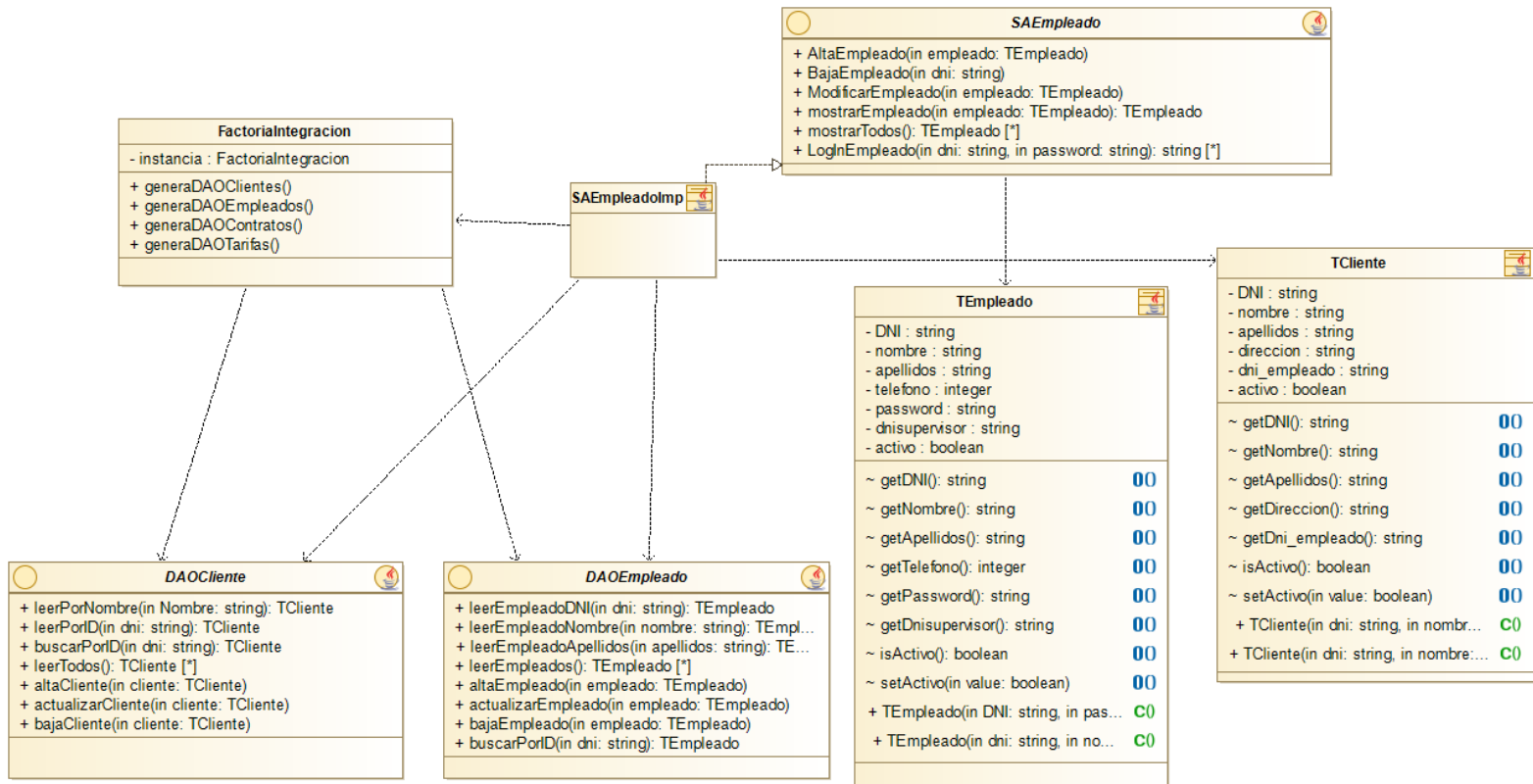
- Cliente SA



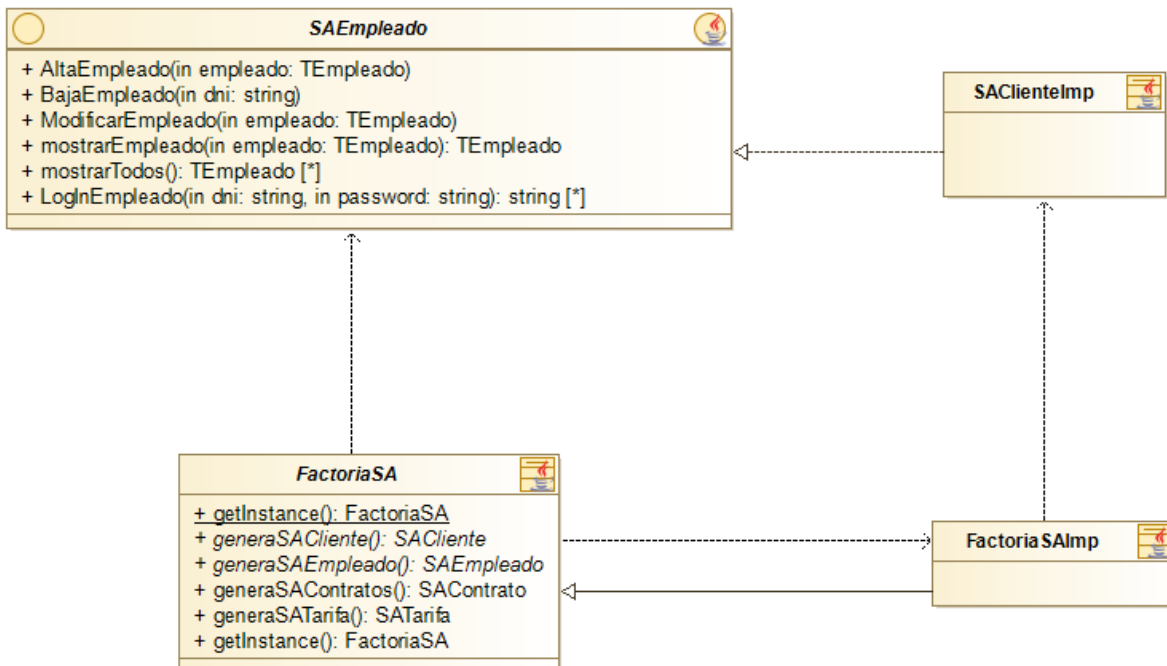
- Cliente Factoría



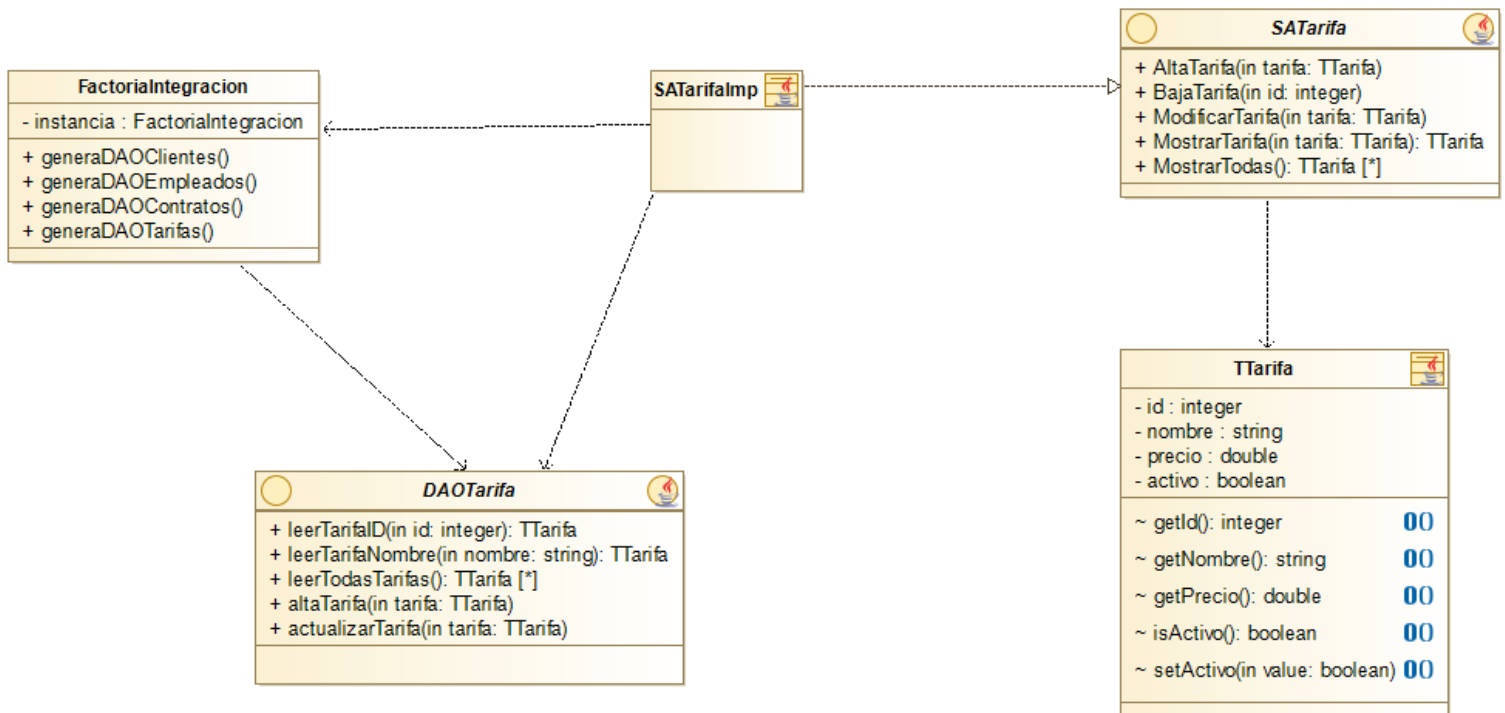
- **Empleado SA**



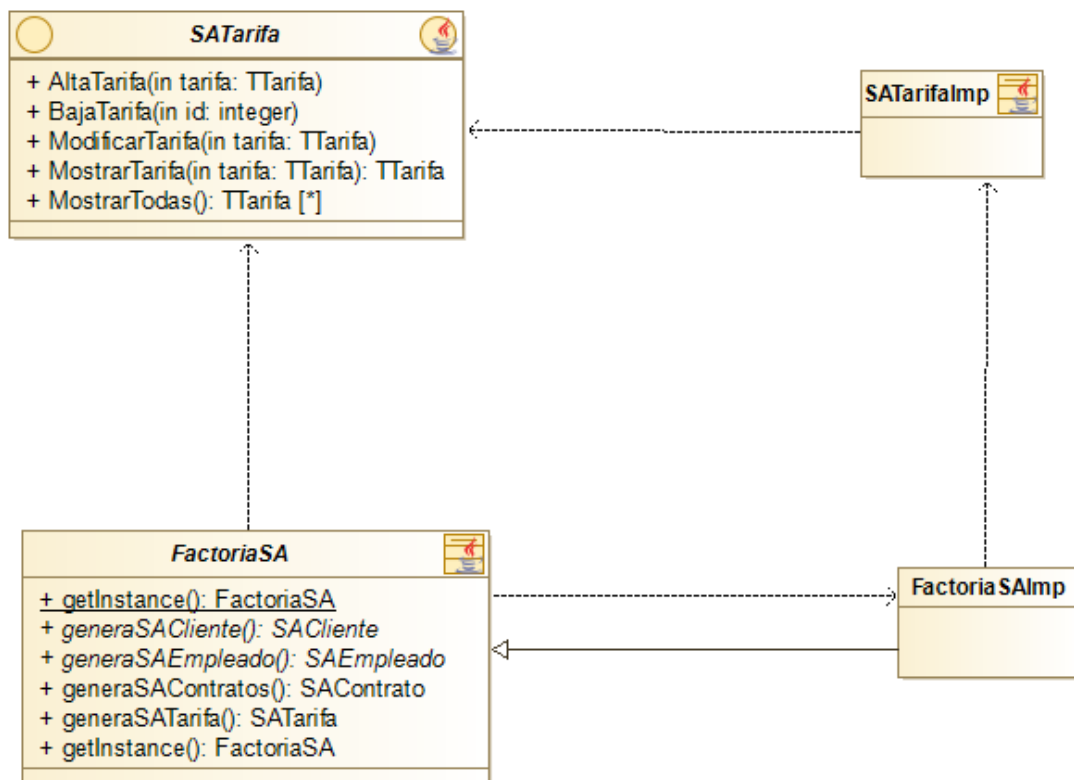
- **Empleado Factoría**



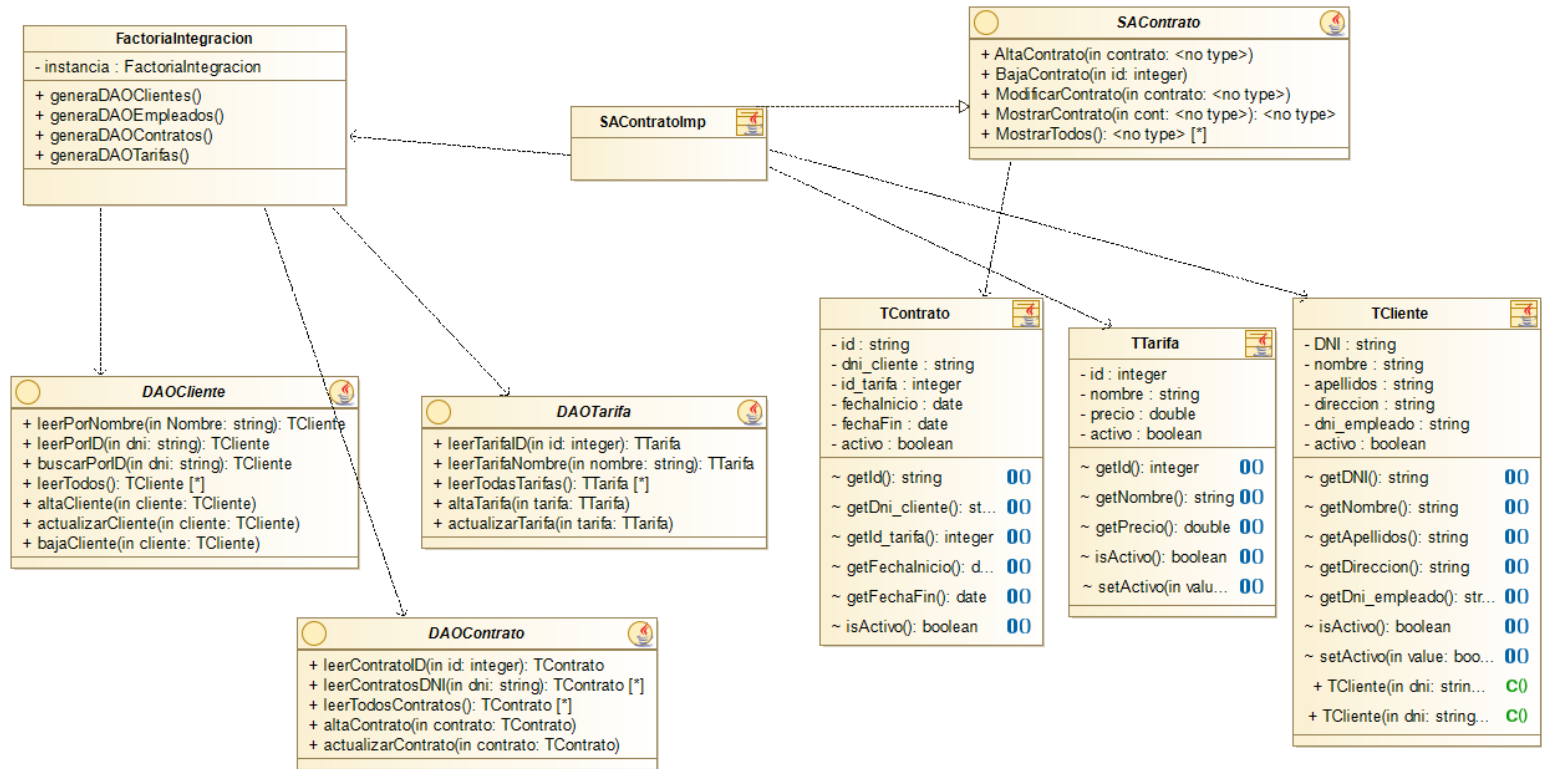
- Tarifa SA



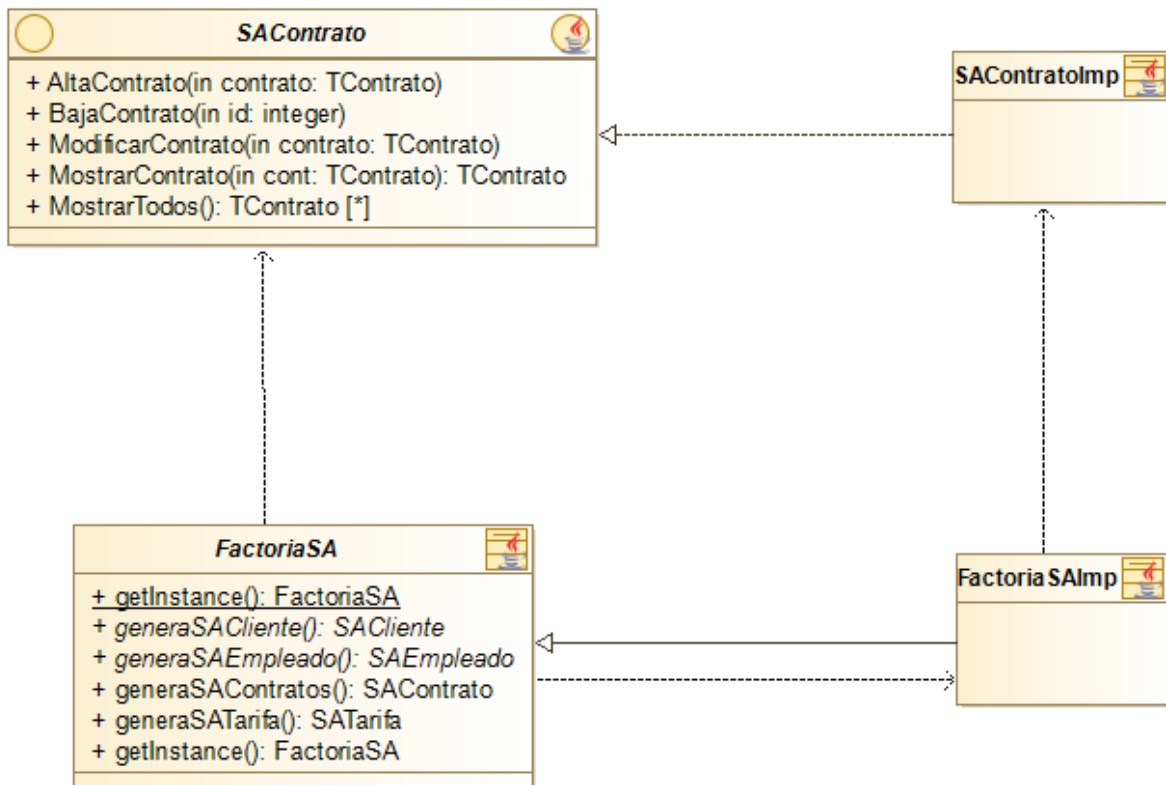
- Tarifa Factoría



- Contrato SA

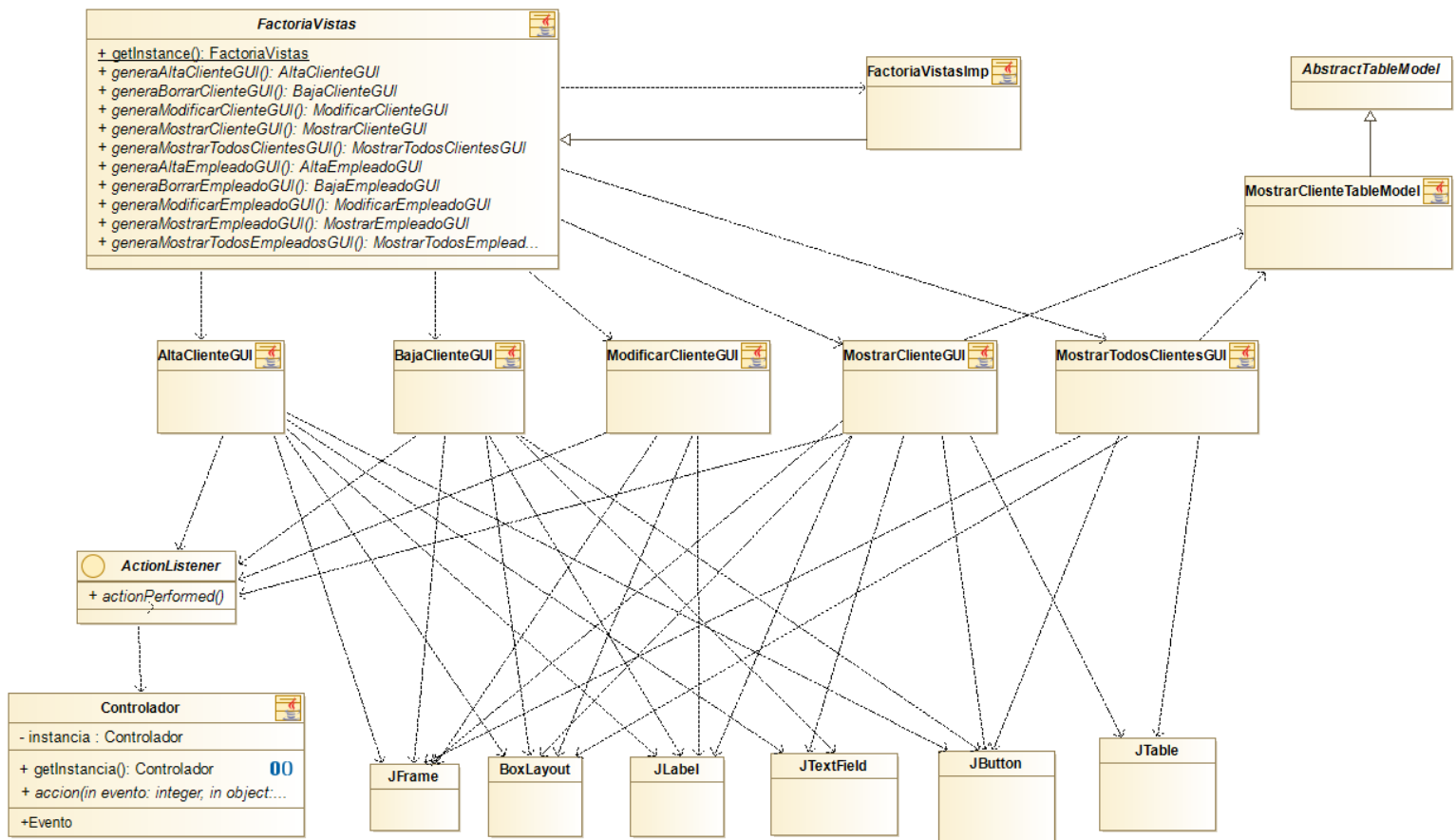


- Contrato Factoría

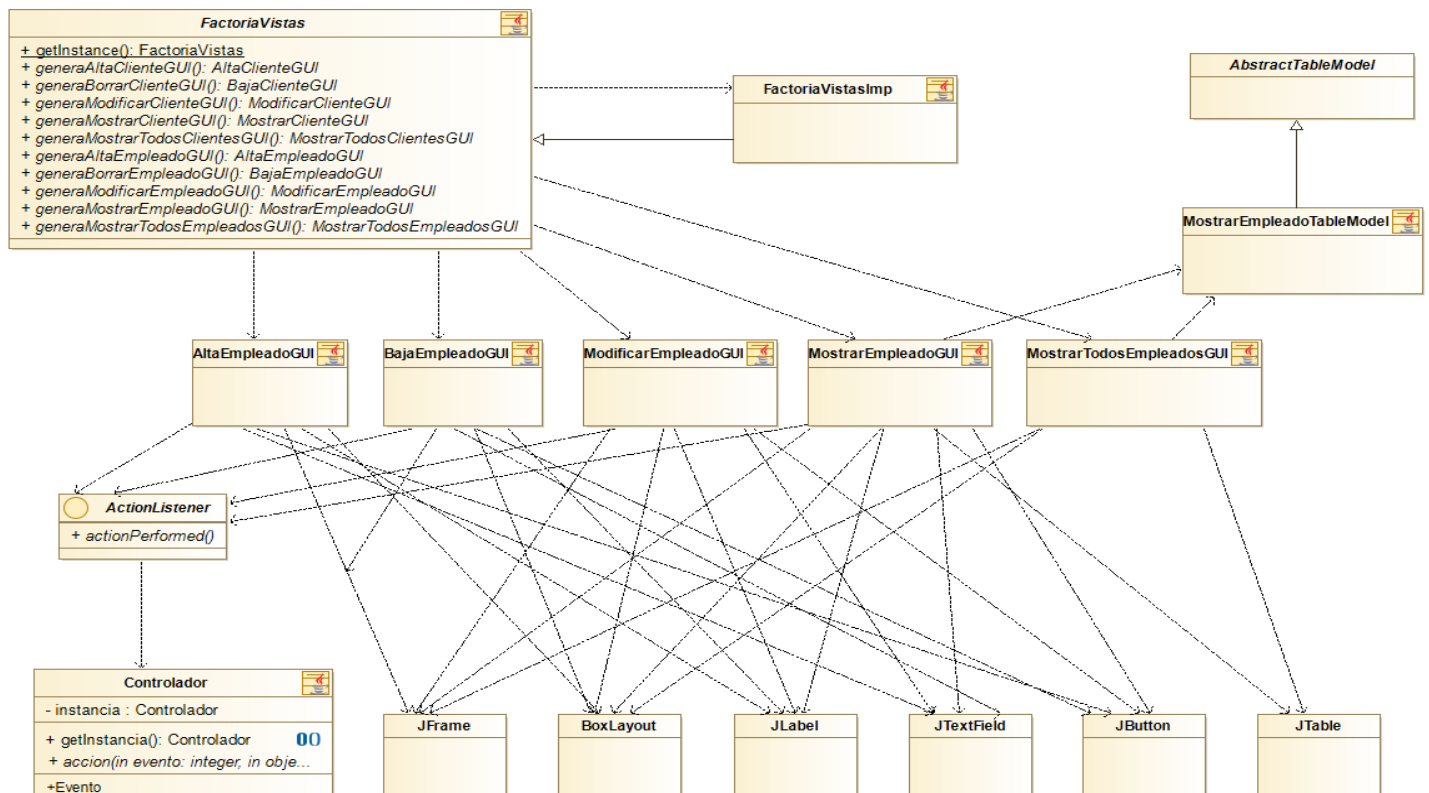


Presentación

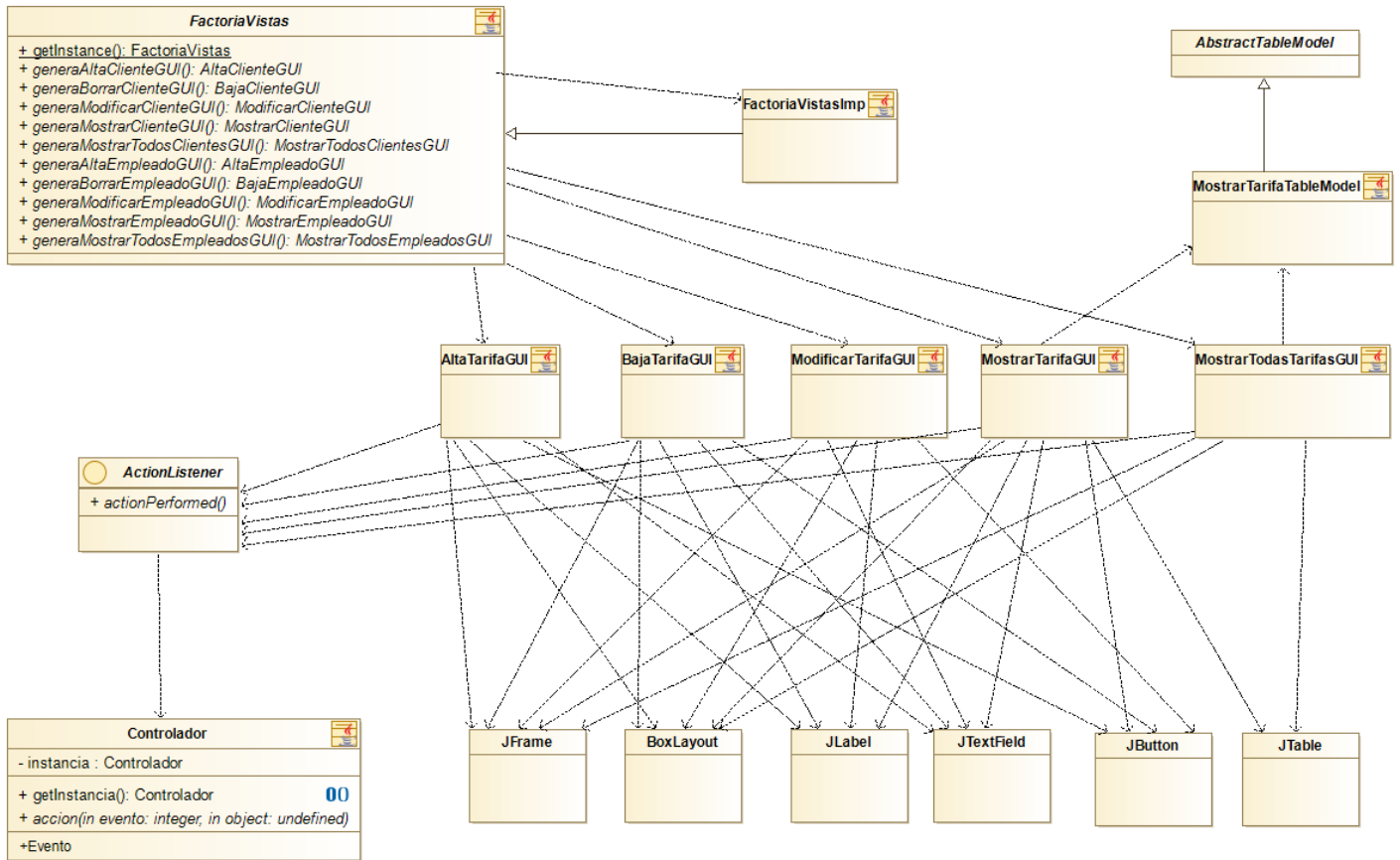
- Cliente



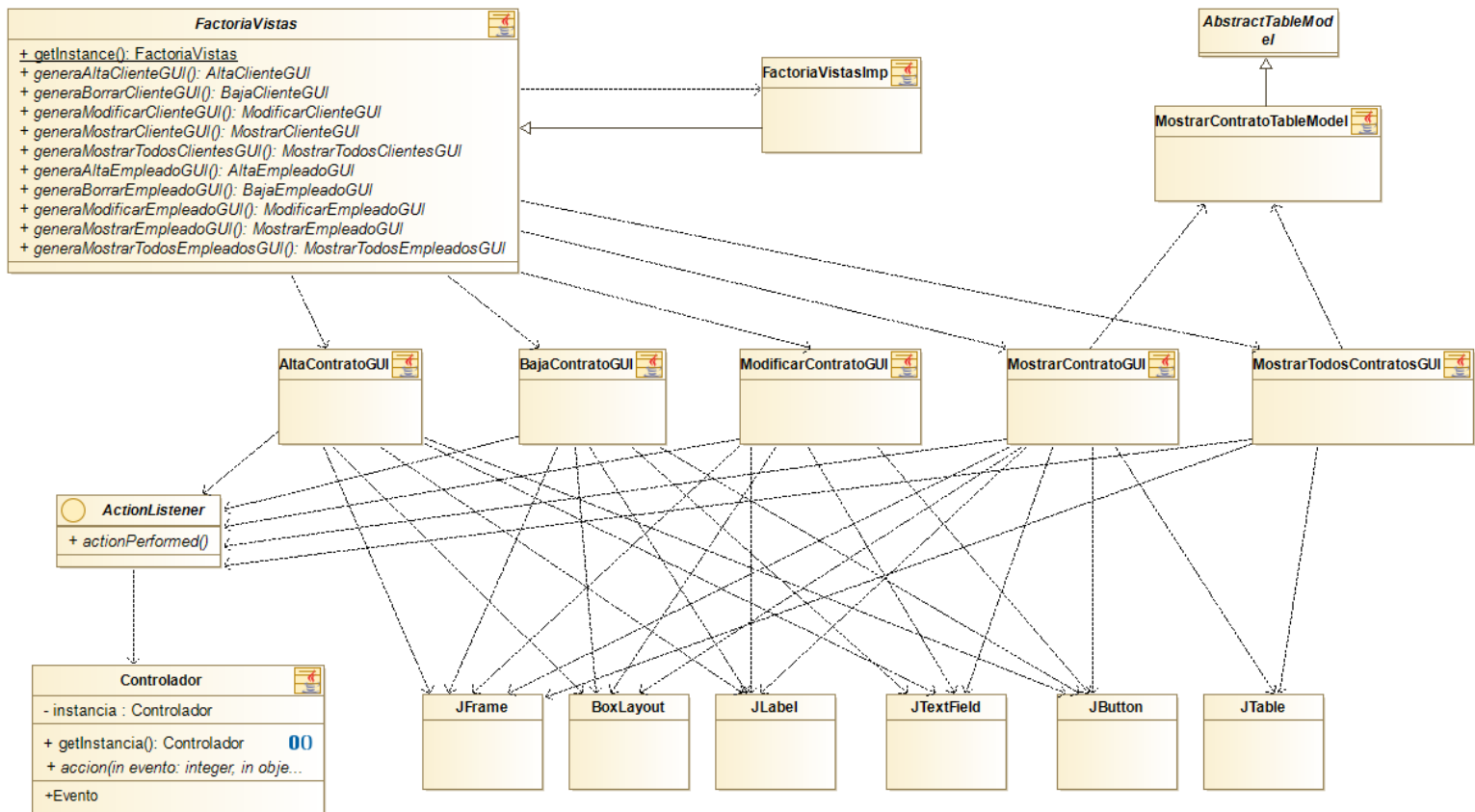
- Empleado



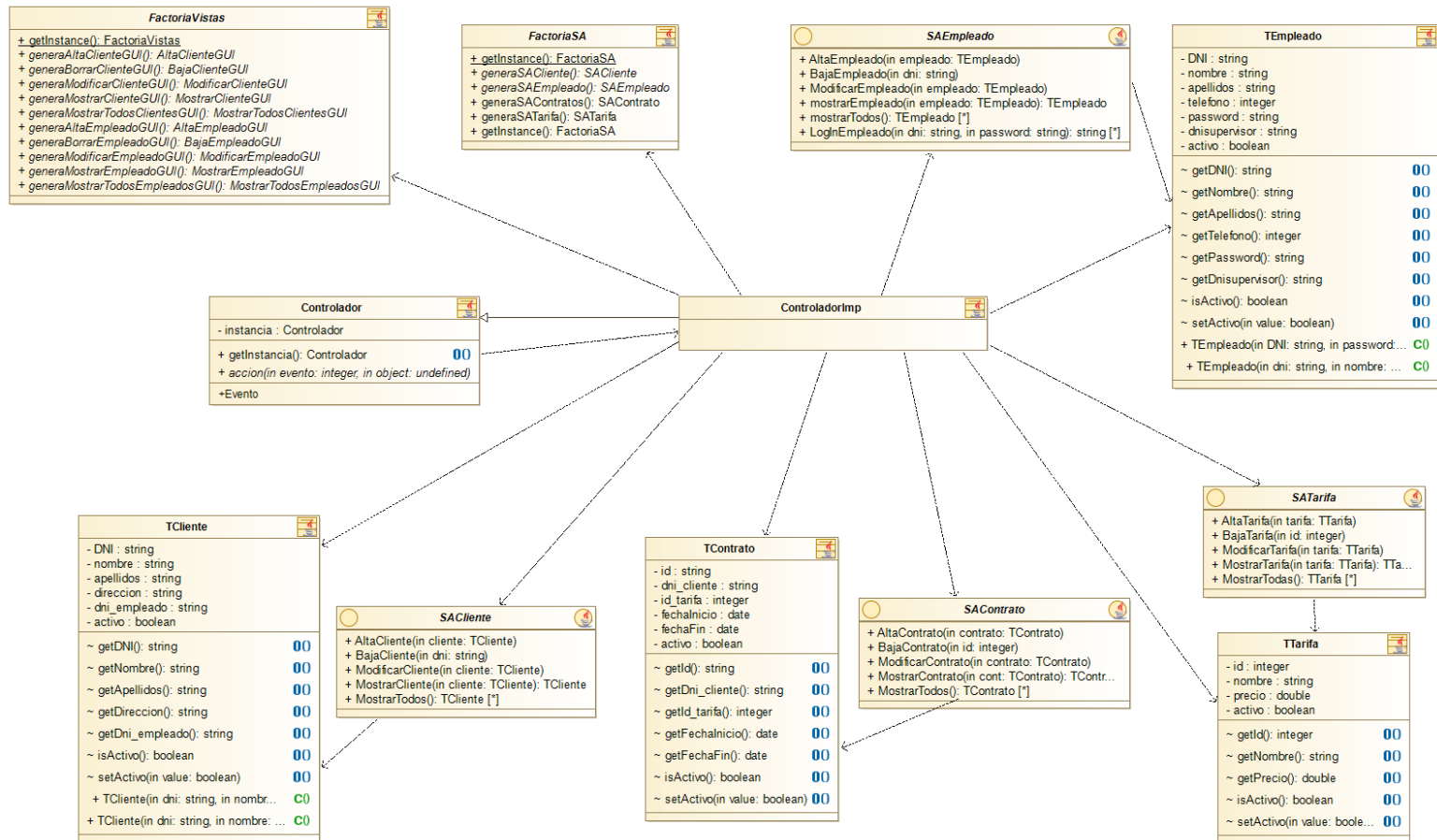
- Tarifa



- Contrato



- Controlador

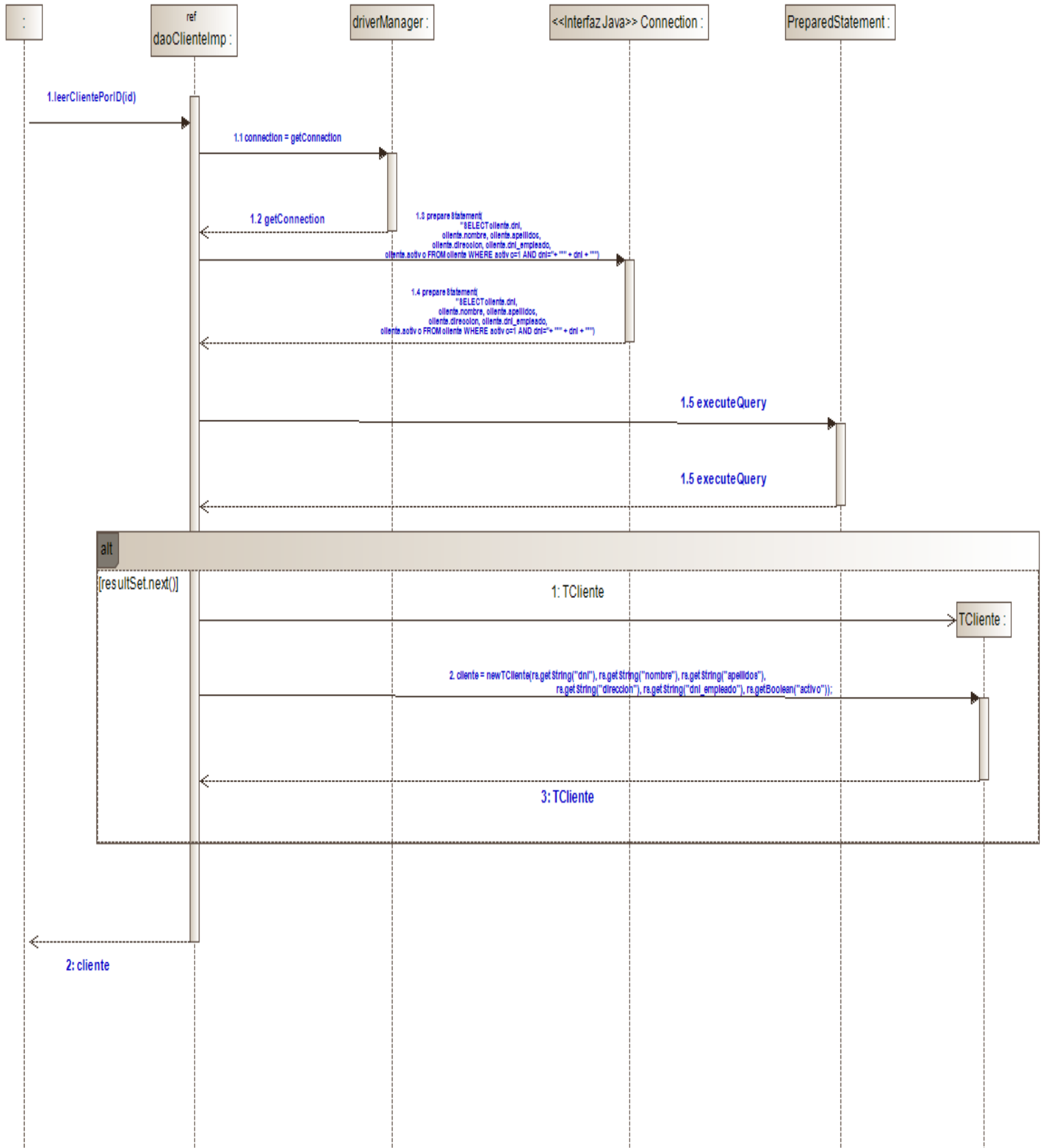


6.2 - De Secuencia

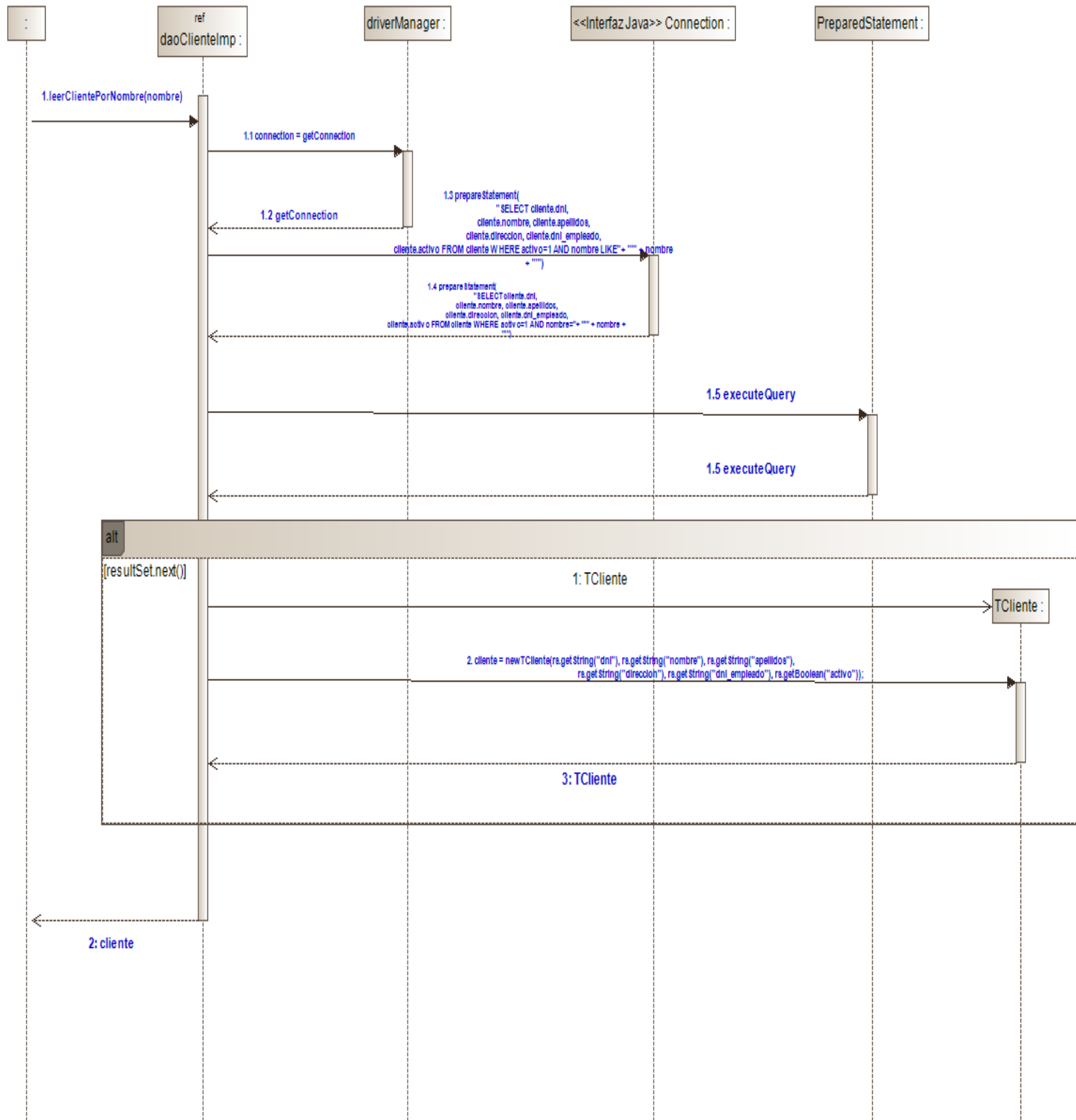
Cliente

Capa de Integración

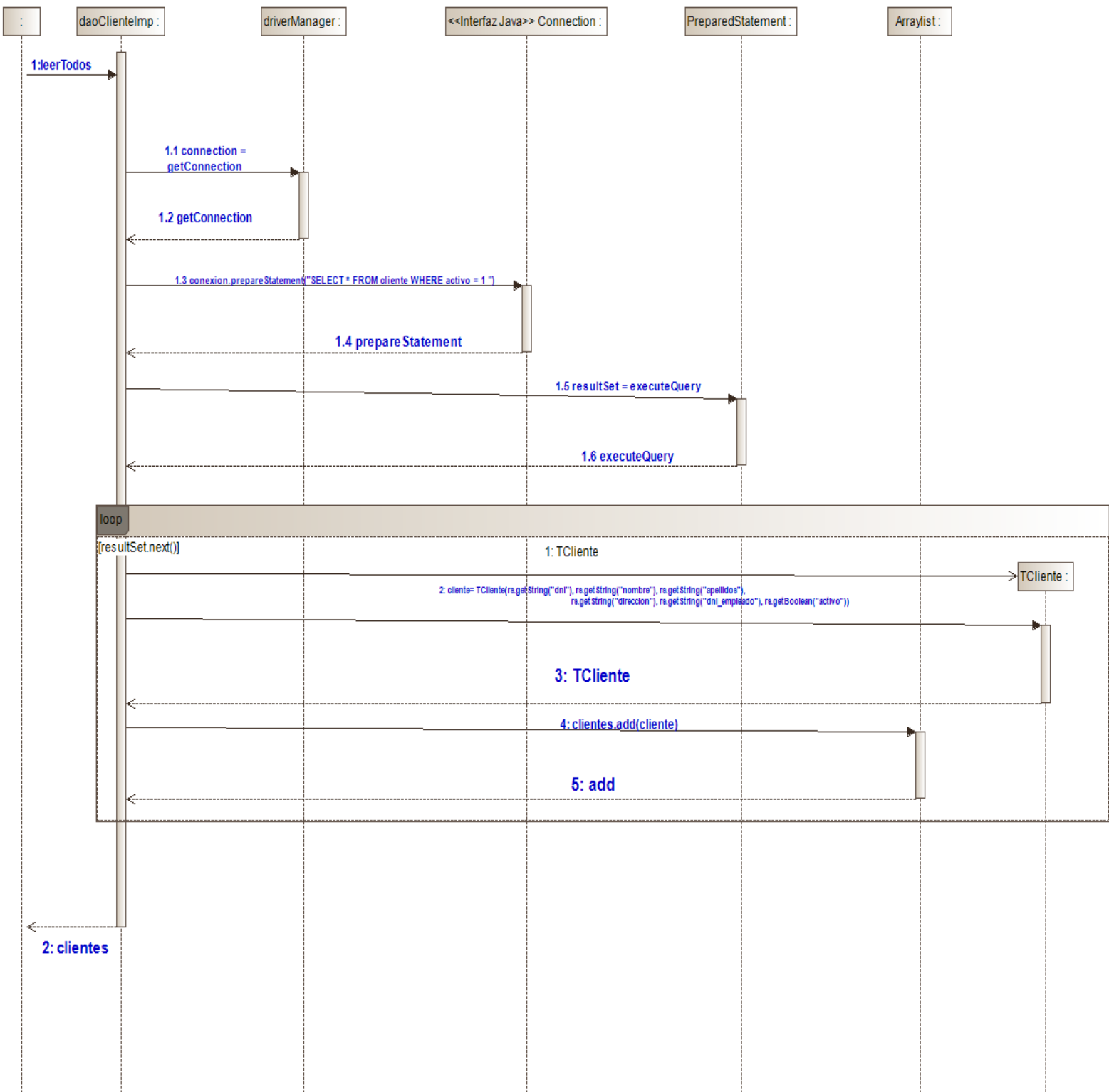
- Leer Cliente por ID



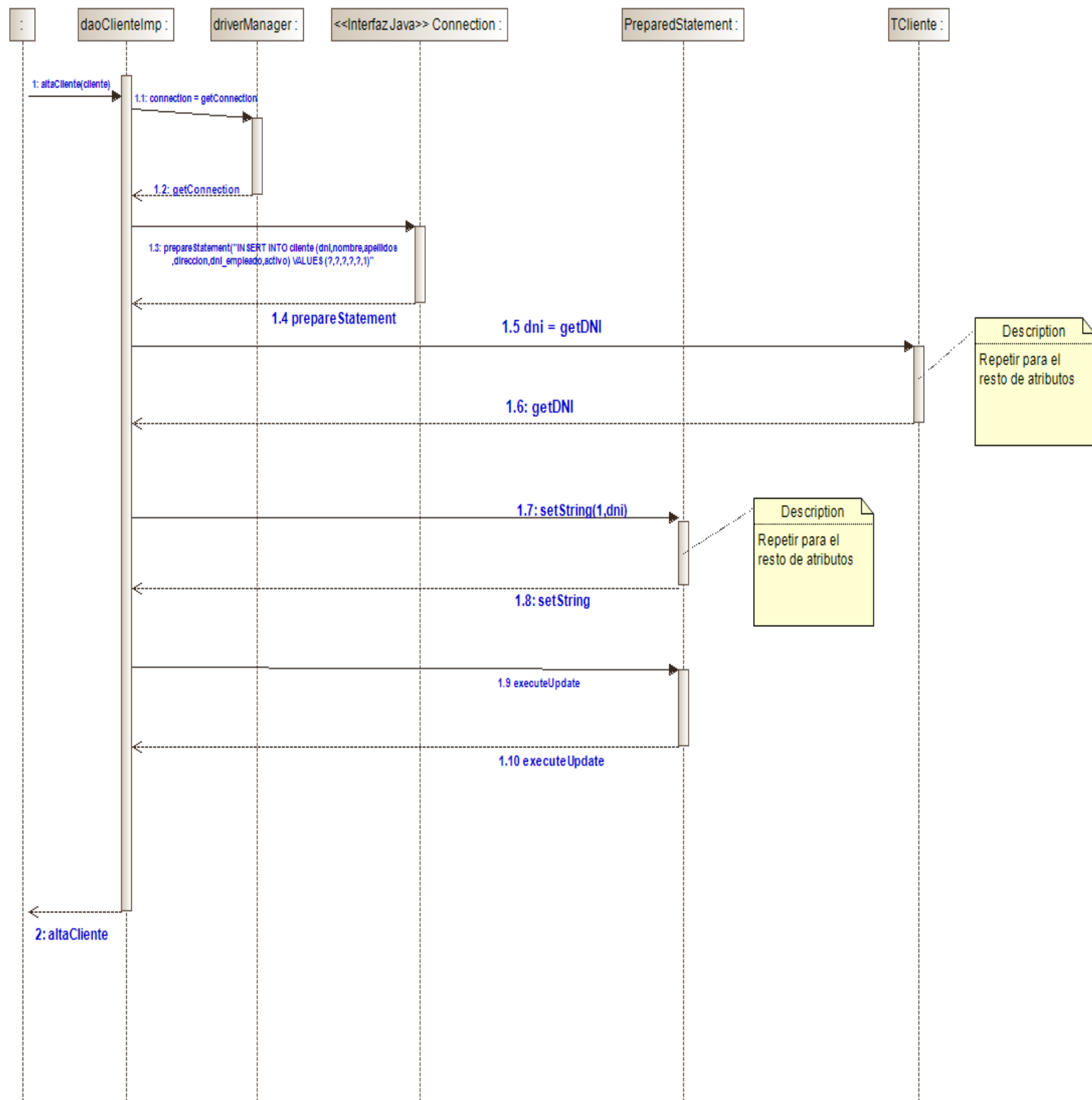
- Leer Cliente por Nombre



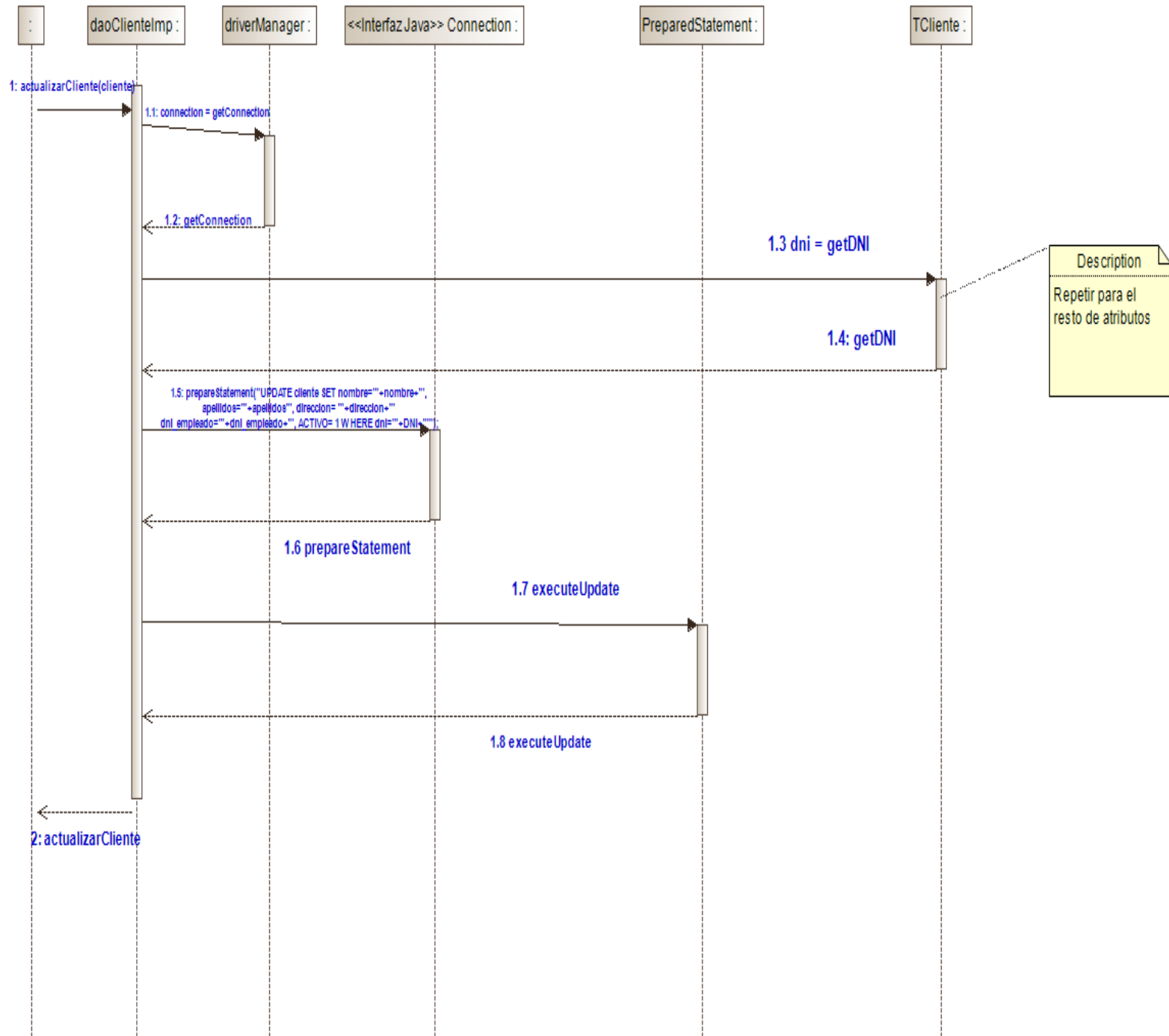
- Leer Todos Cliente



- Alta Cliente

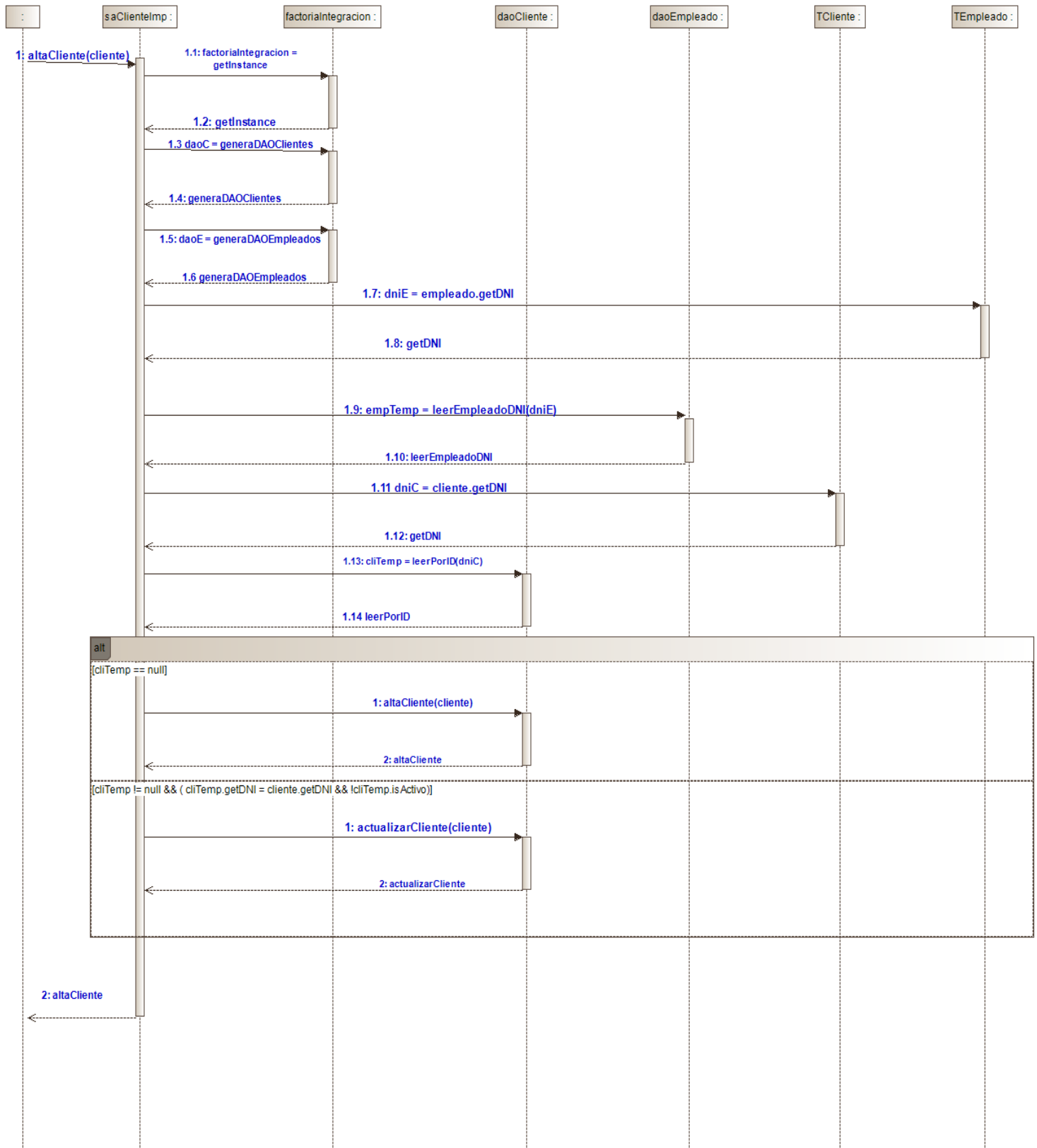


- Actualizar Cliente

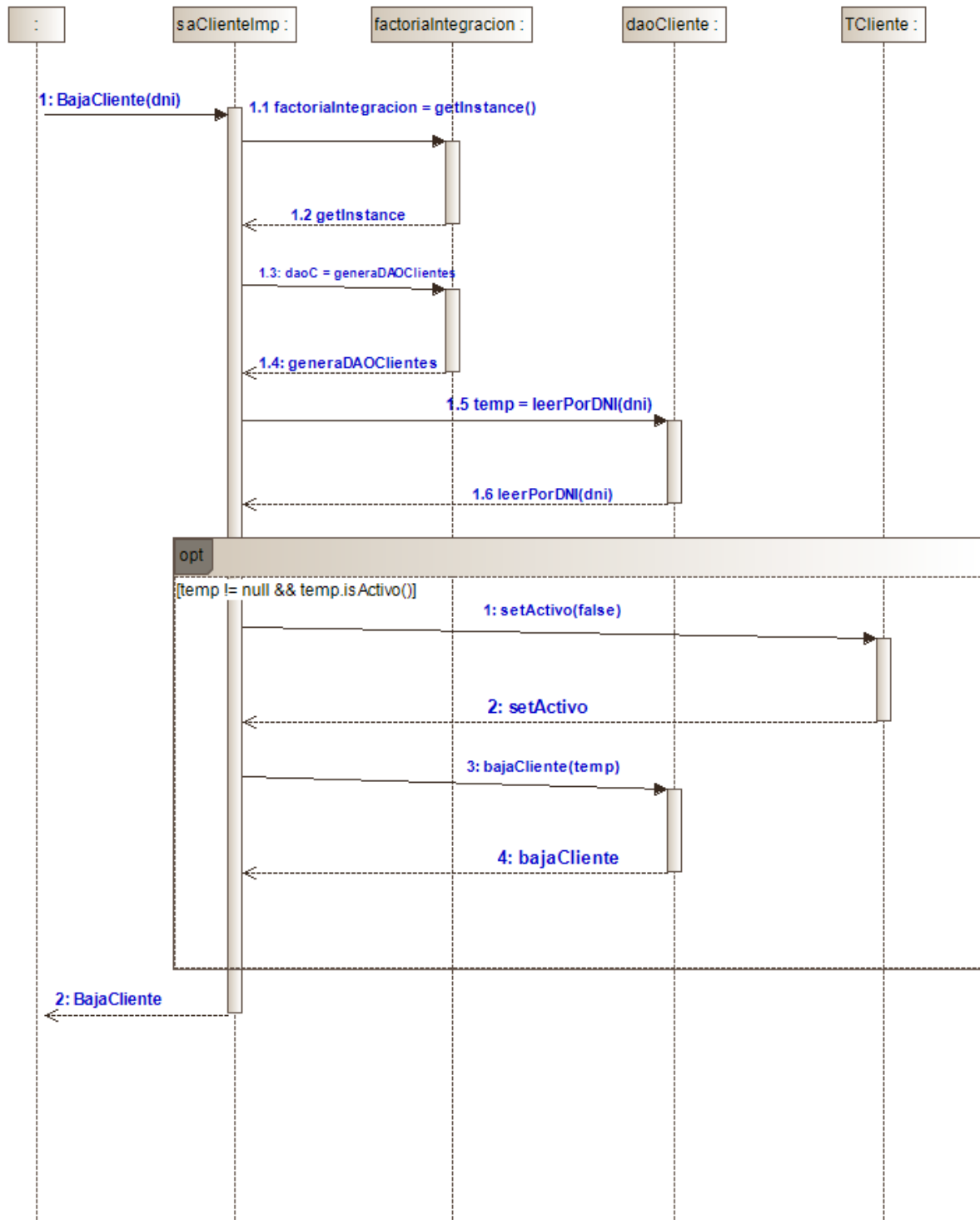


Capa de Negocio

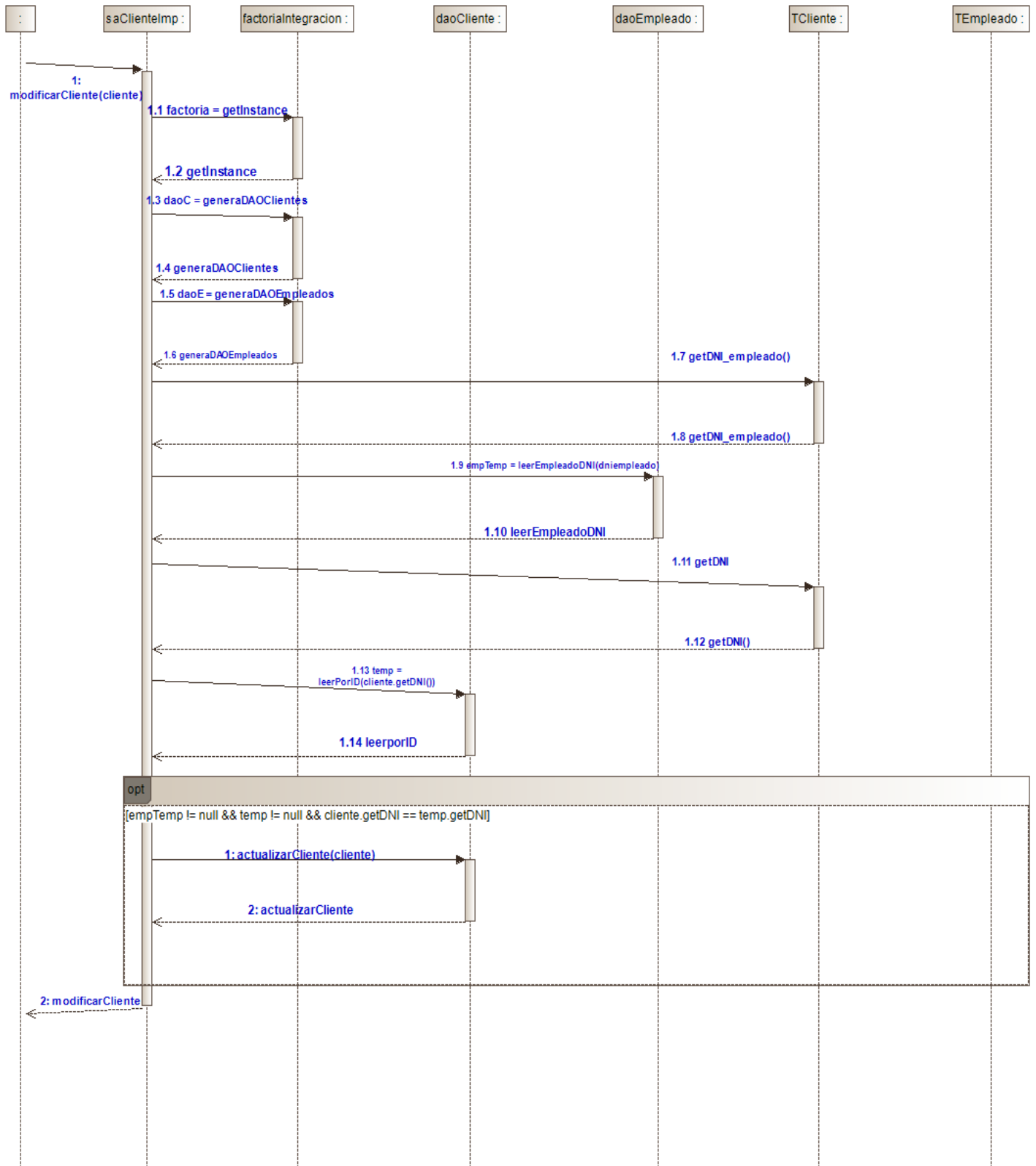
- Alta Cliente SA



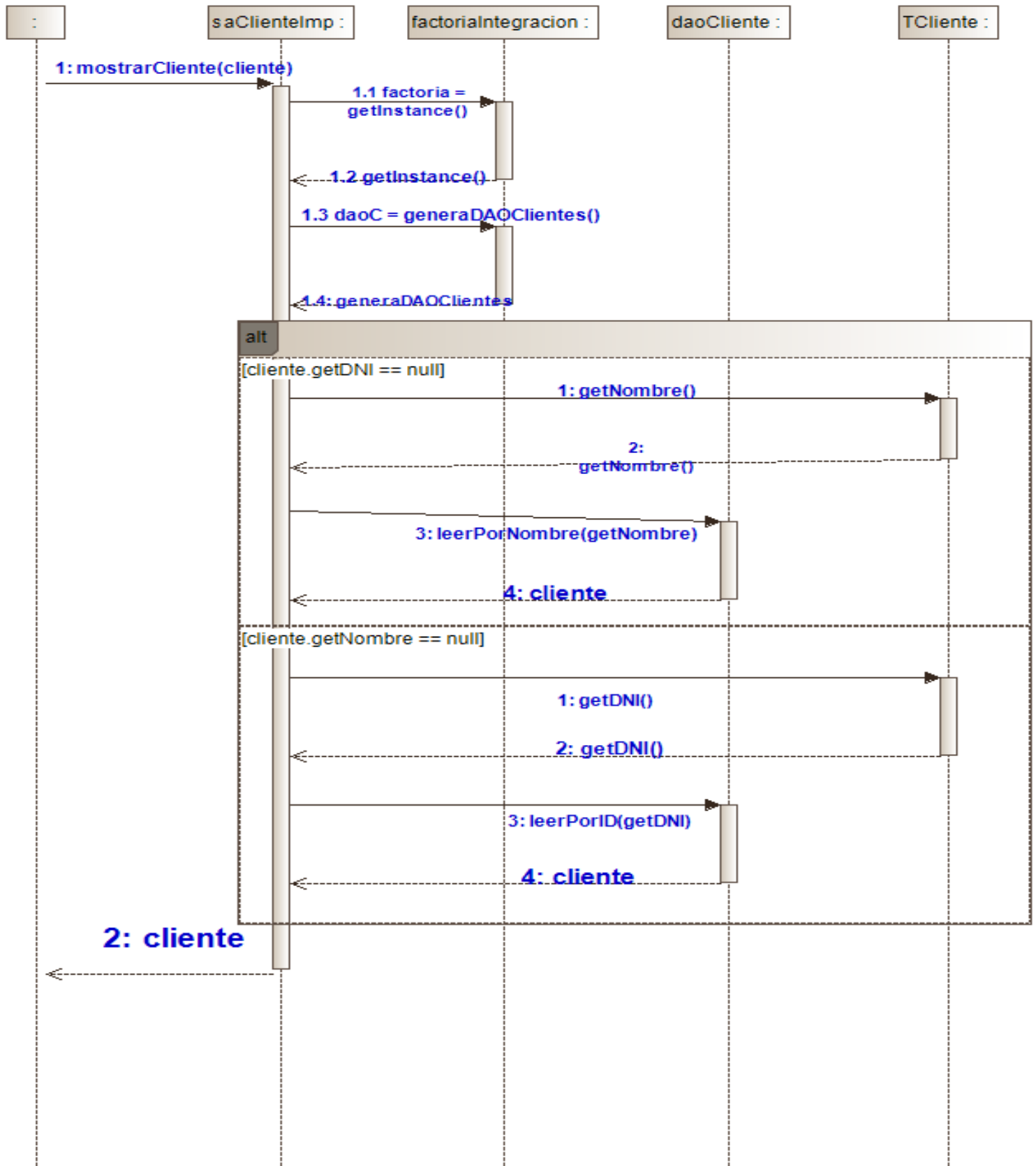
- **Baja Cliente SA**



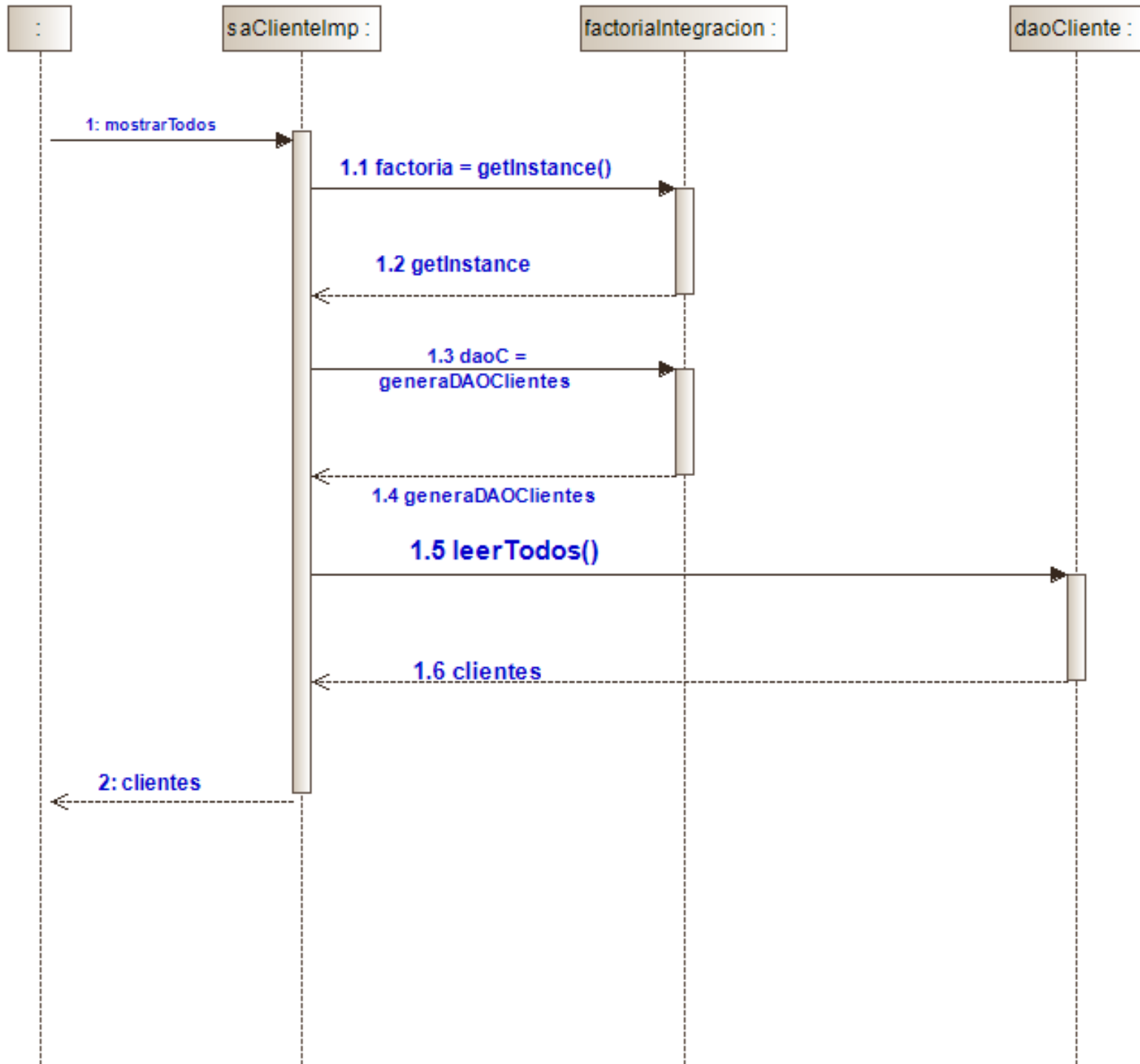
- Modificar Cliente SA



- *Mostrar Cliente SA*

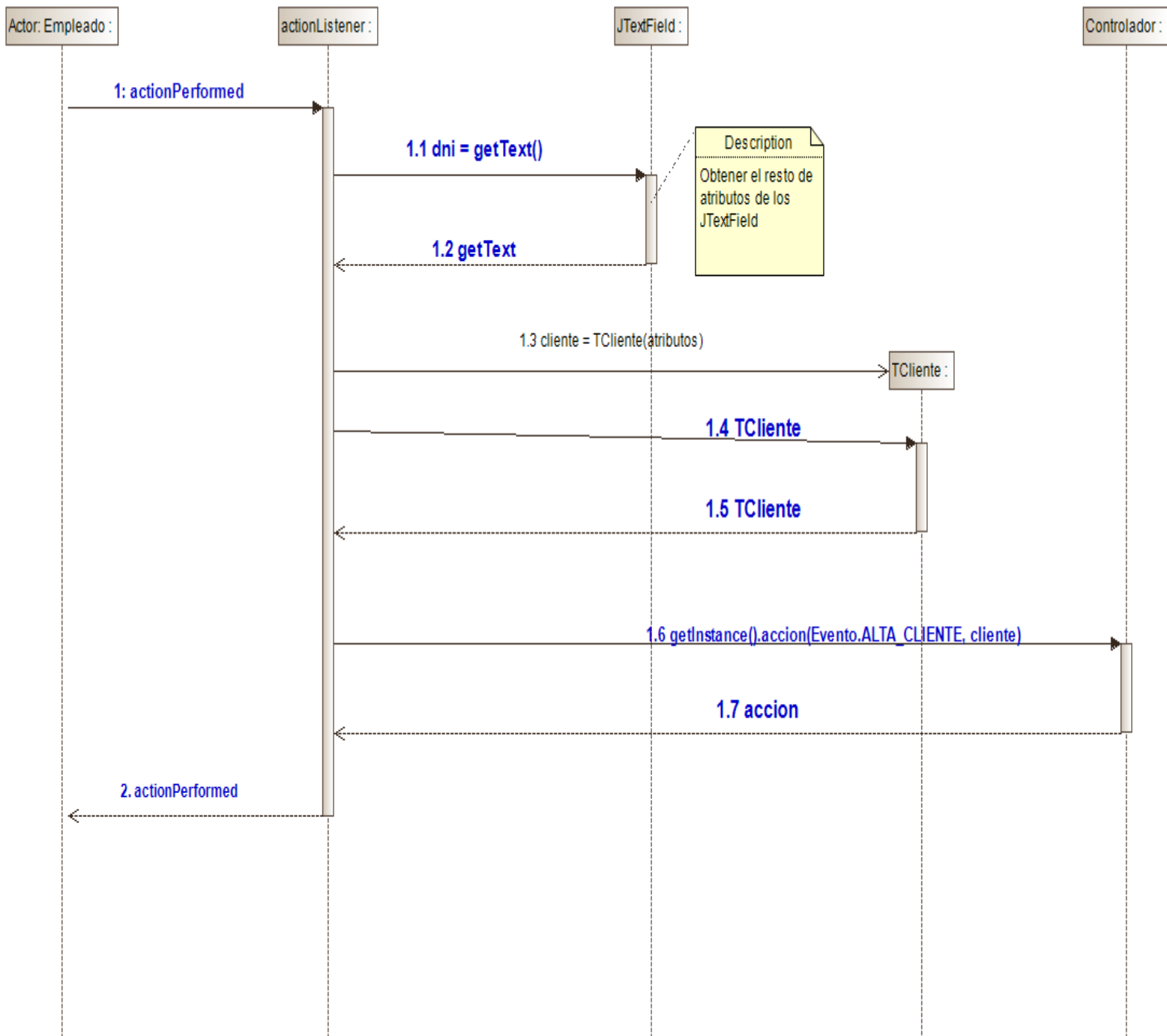


- **Mostrar Todos Cliente SA**

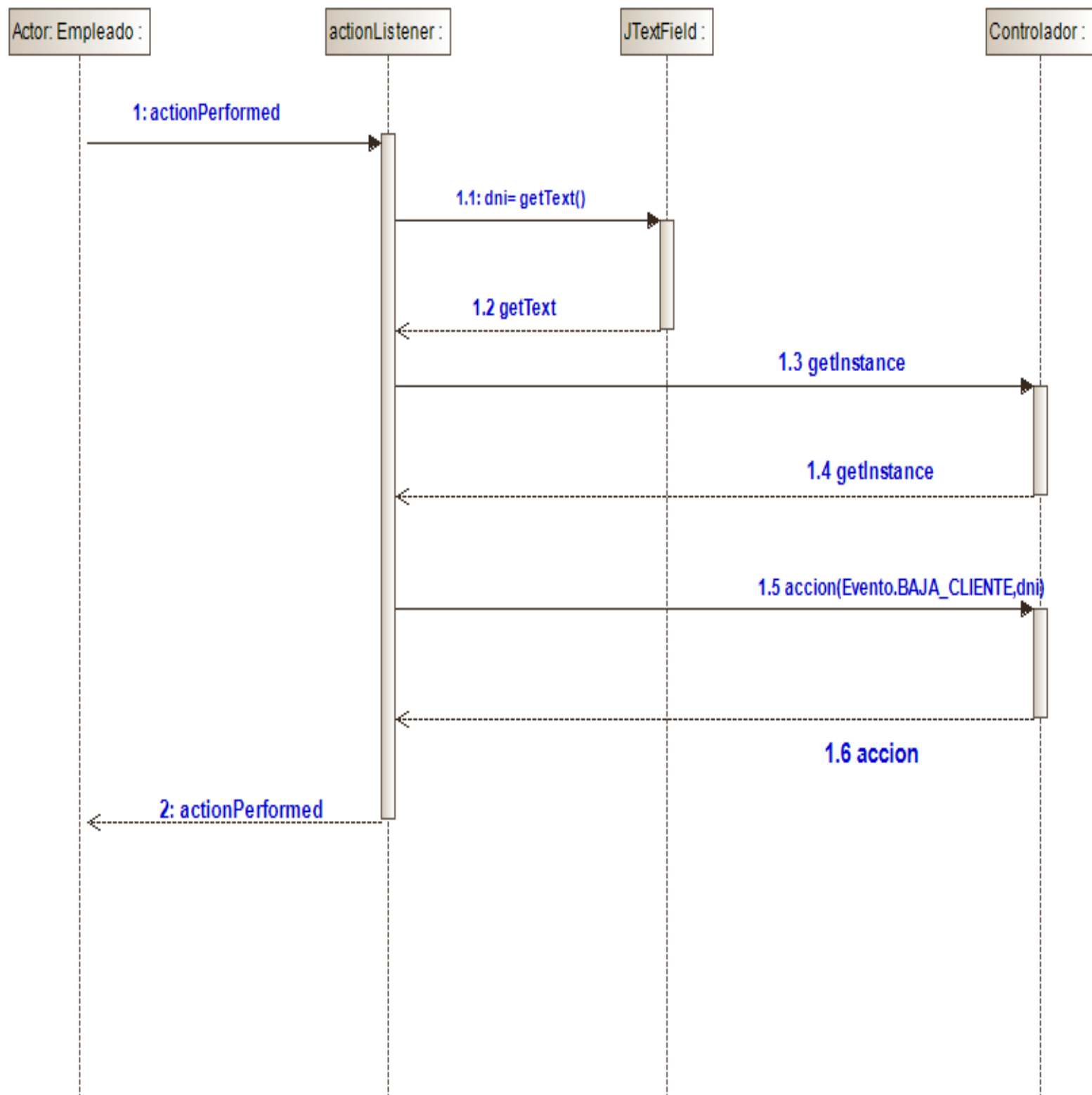


Capa de Presentación

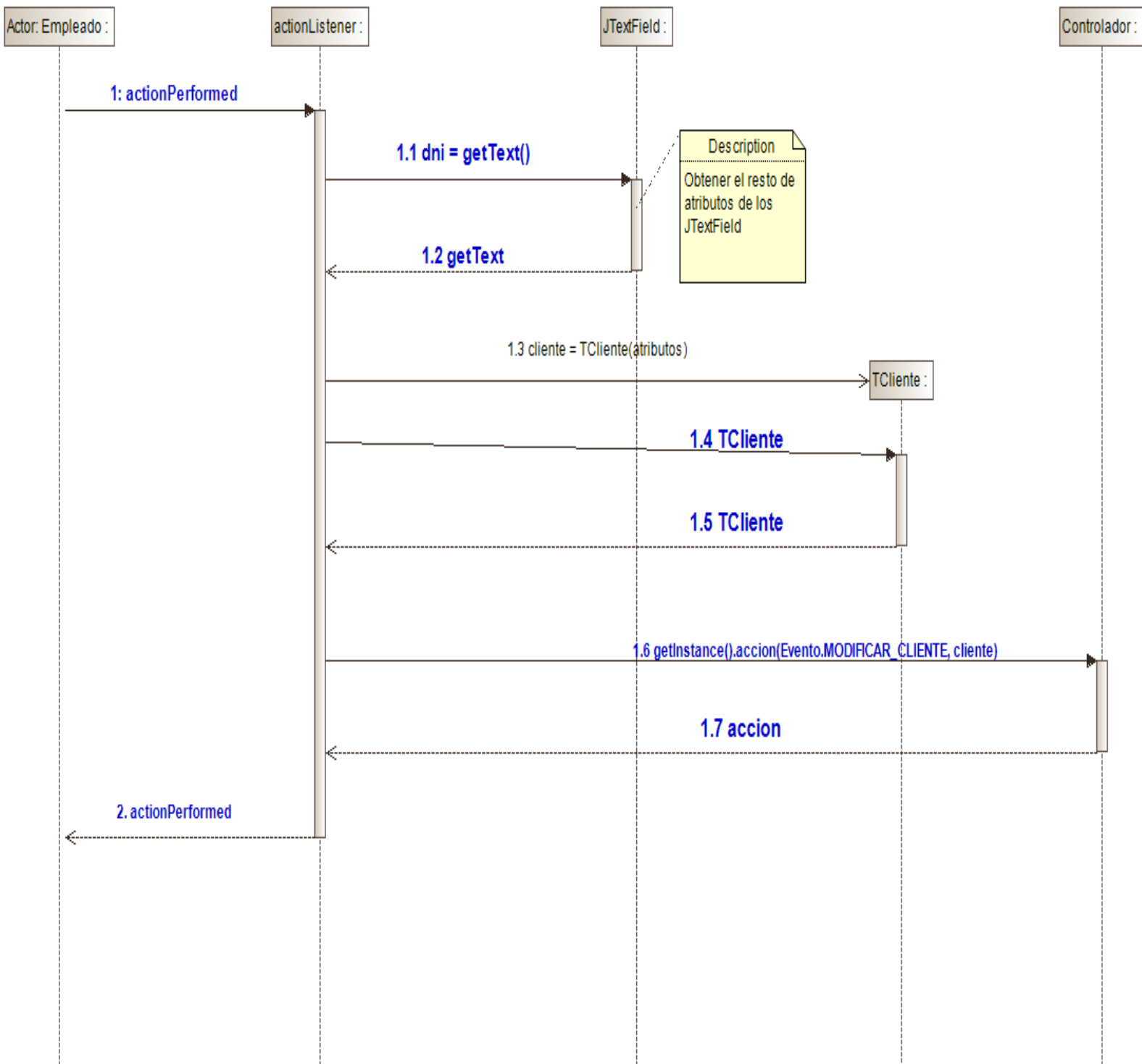
- Alta Cliente Action Listener



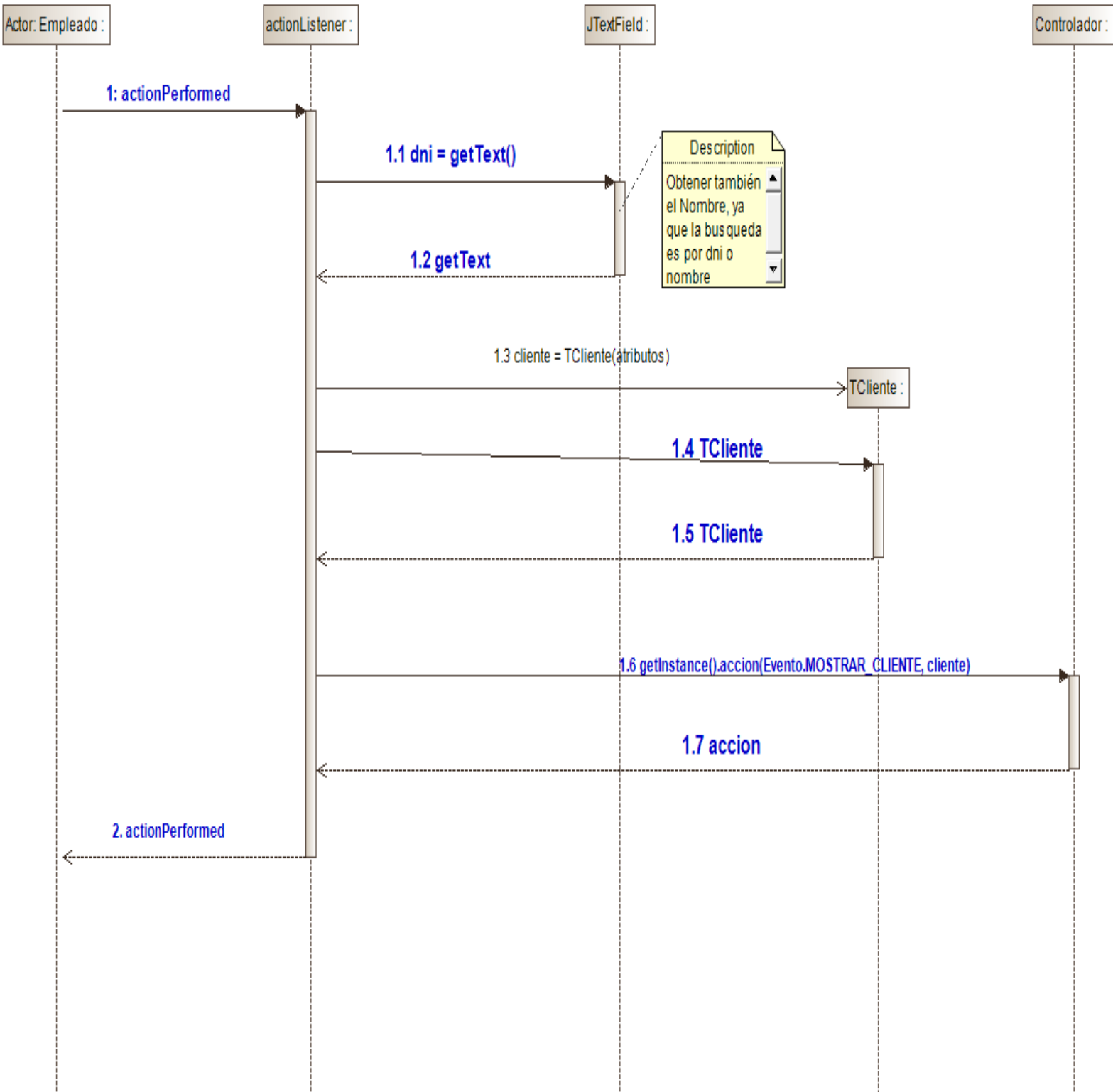
- *Baja Cliente Action Listener*



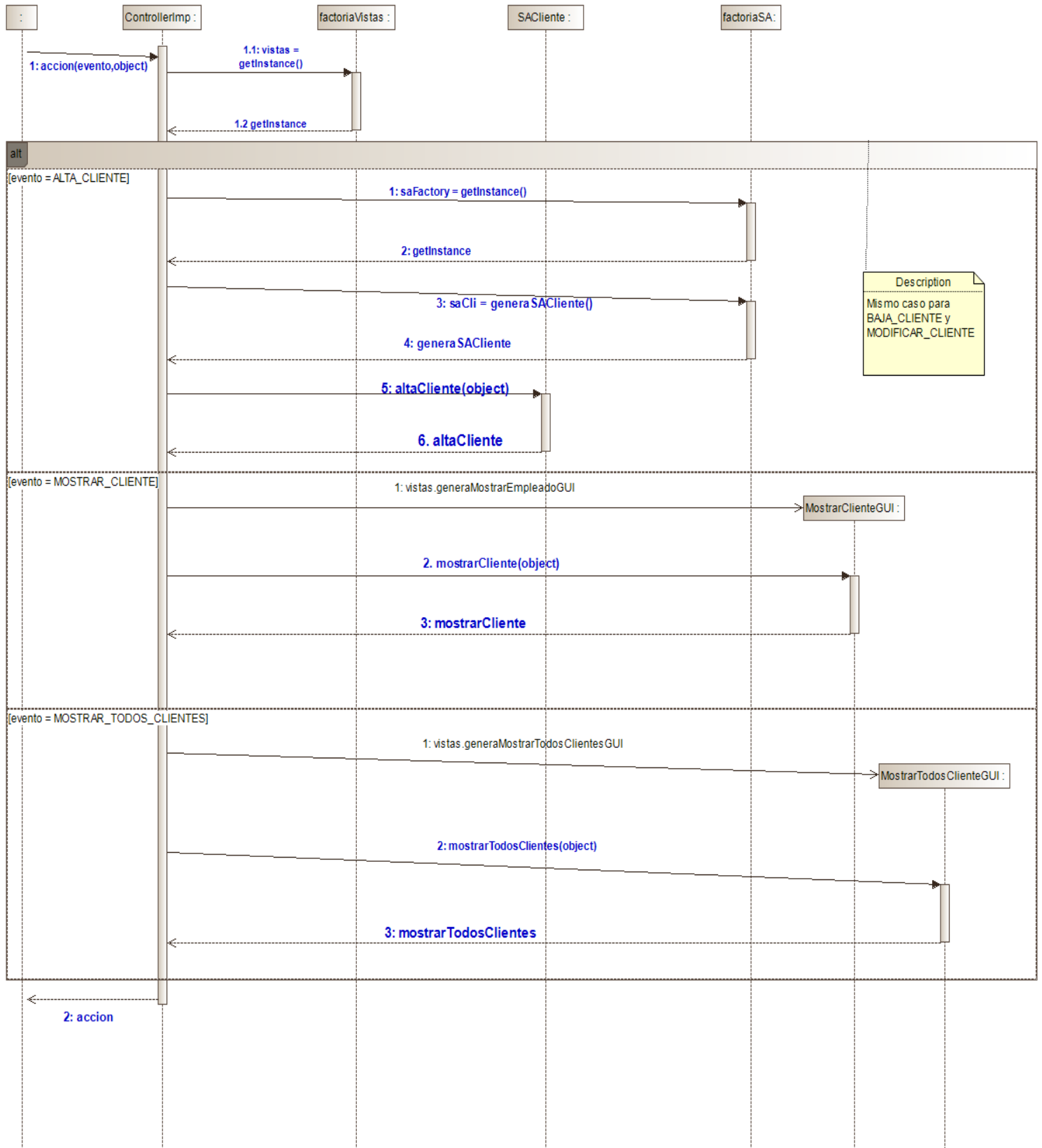
- **Modificar Cliente Action Listener**



- **Mostrar Cliente - Action Listener**



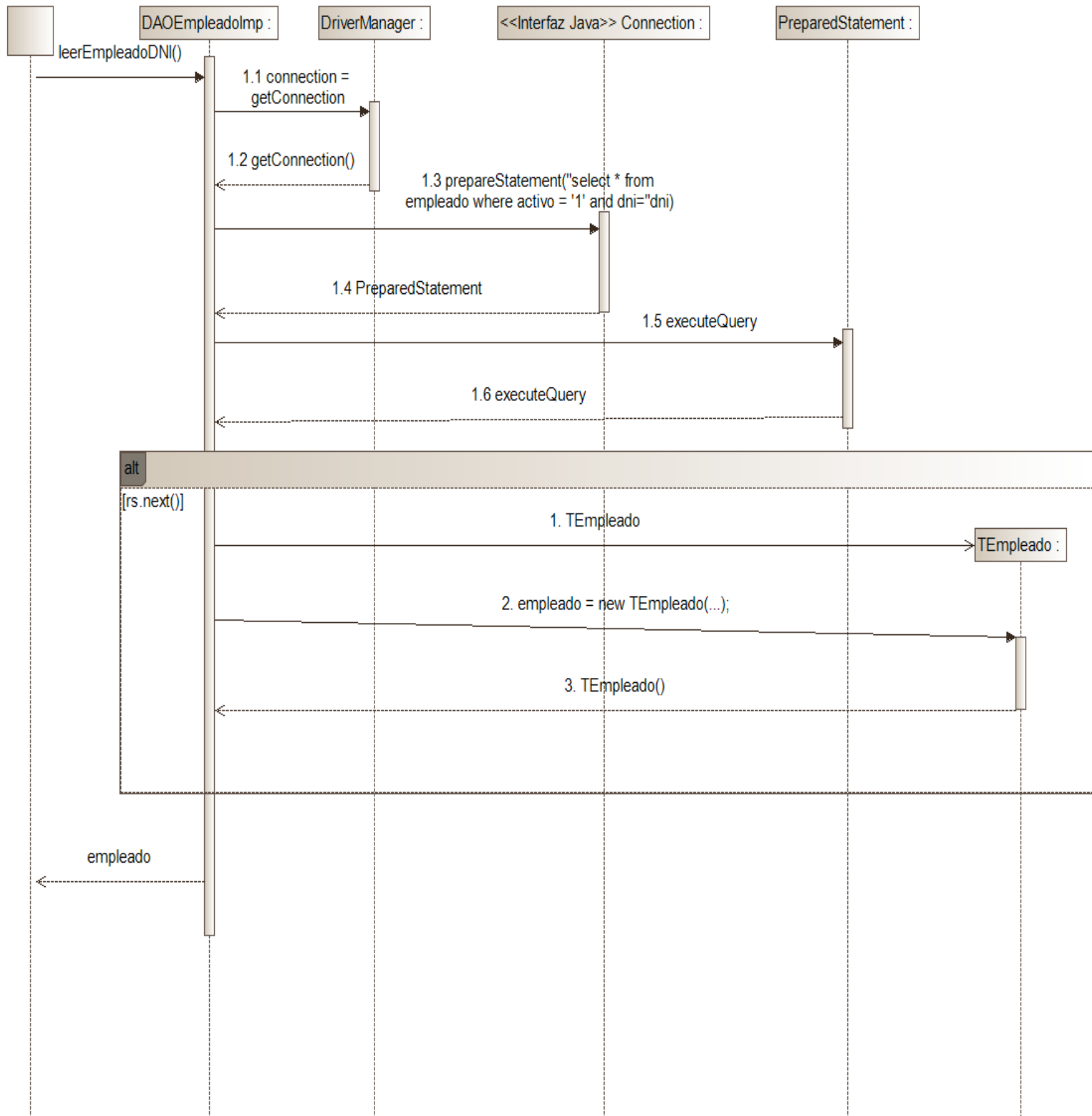
- Controlador Cliente



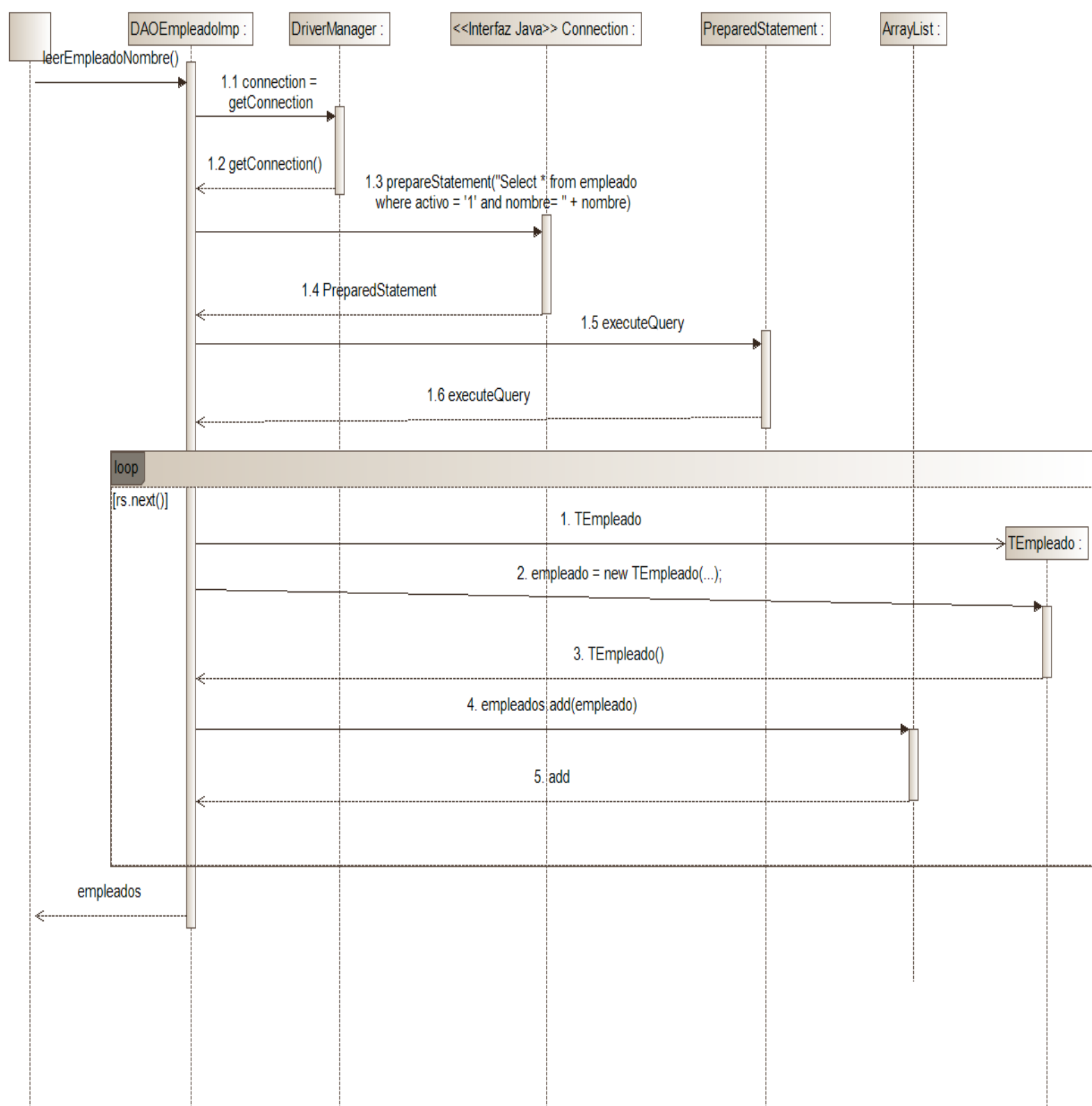
Empleado

Capa de Integración

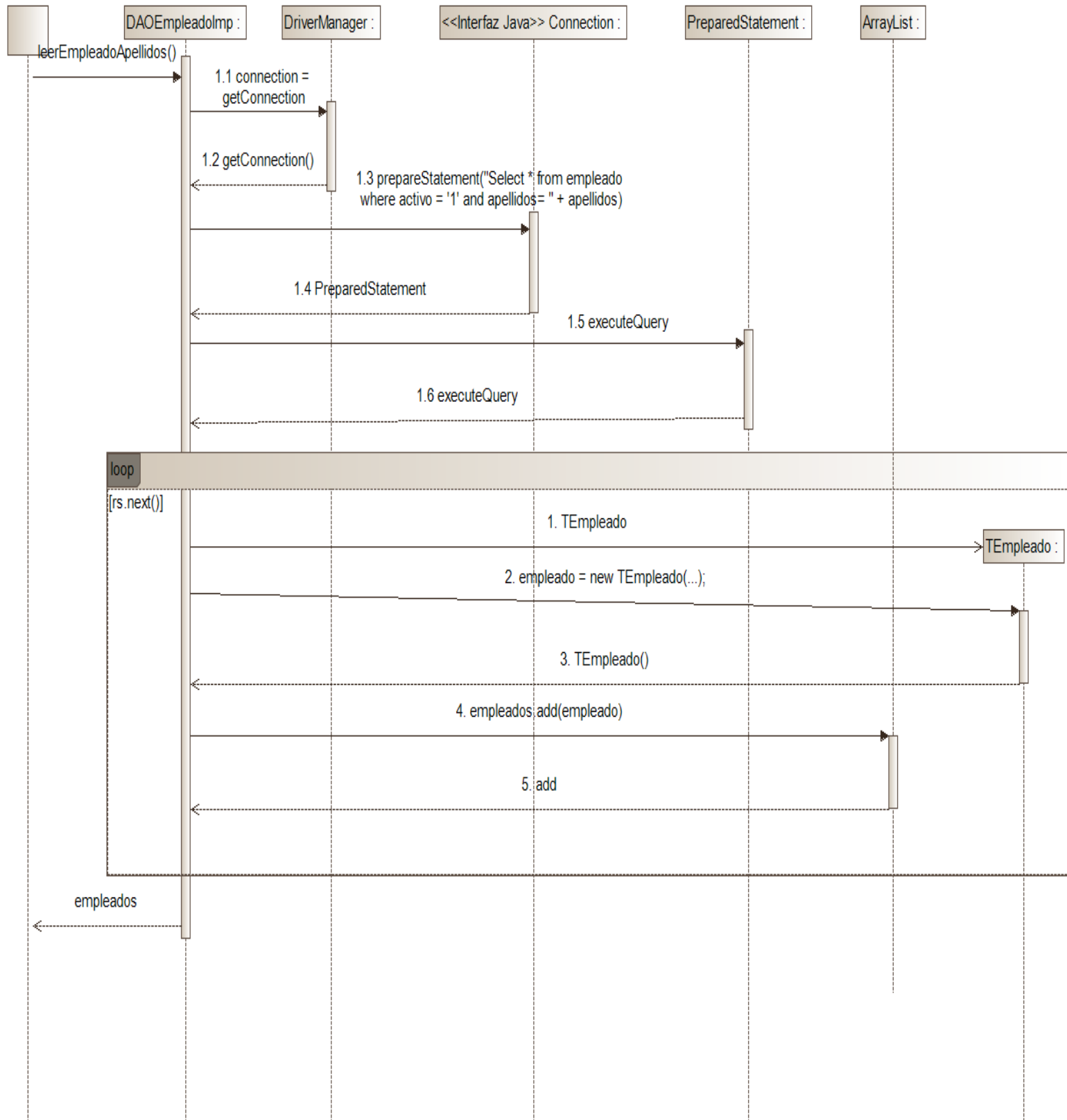
- Leer Empleado por DNI



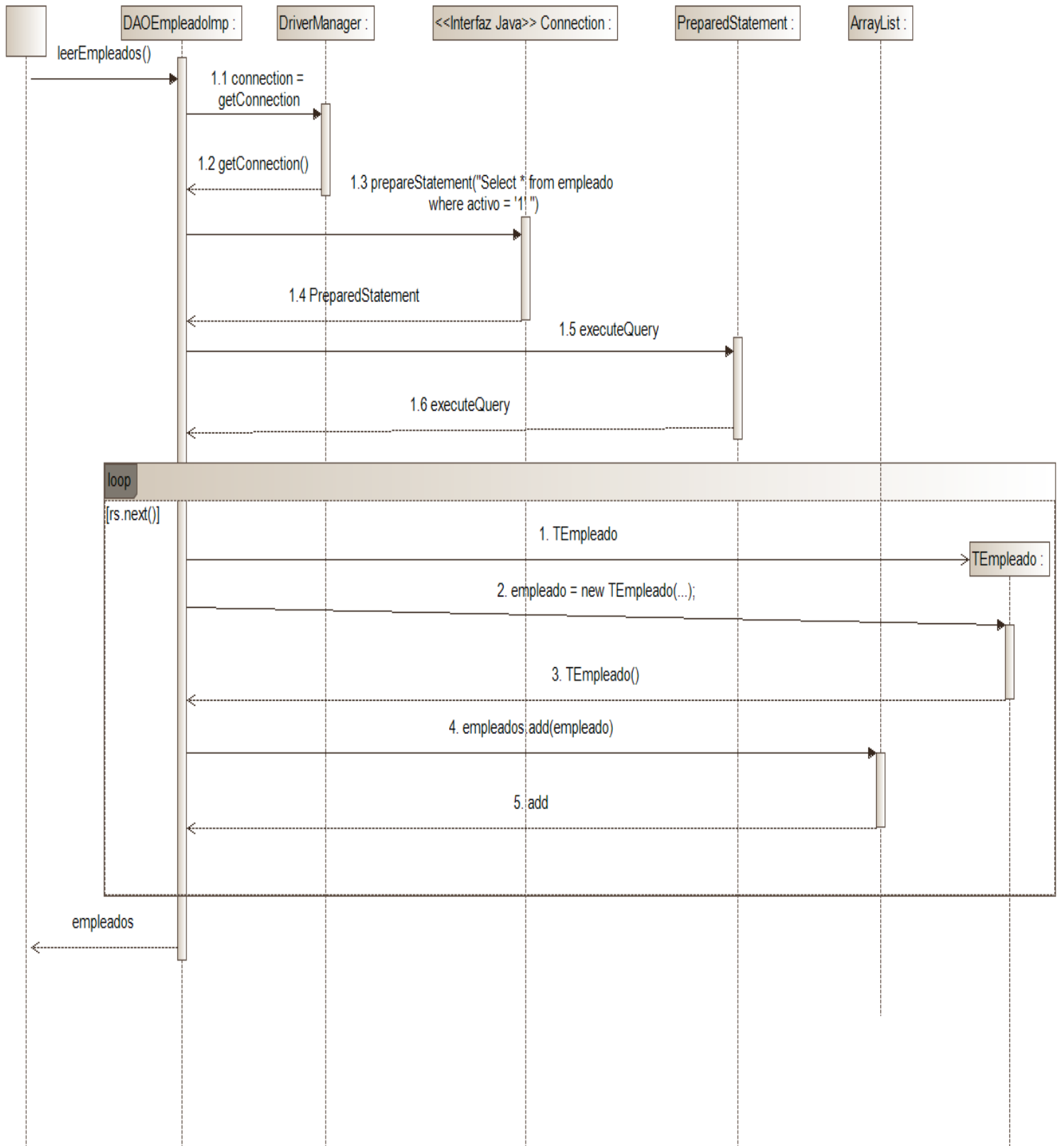
- **Leer Empleado por Nombre**



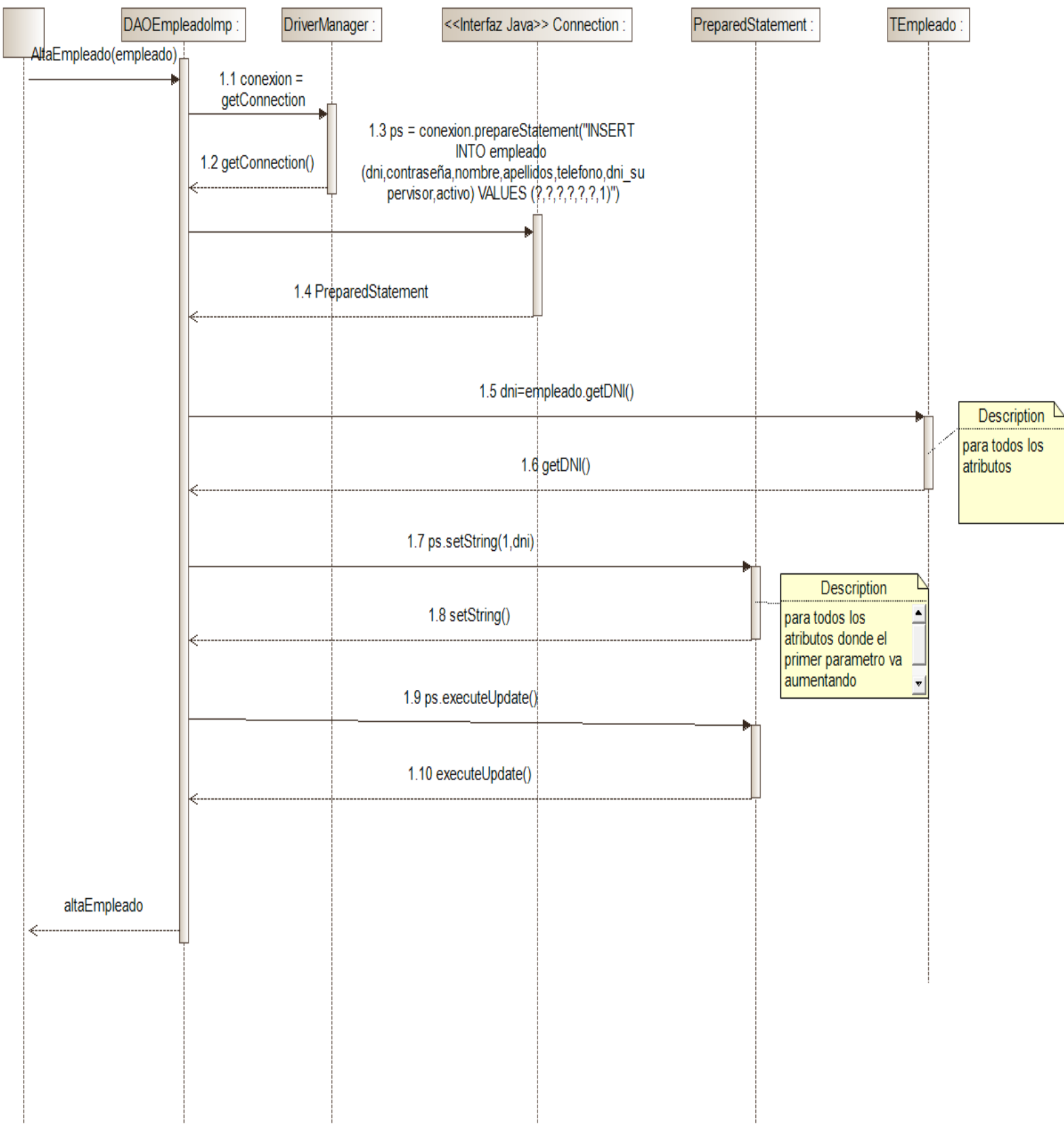
- Leer Empleado por Apellidos



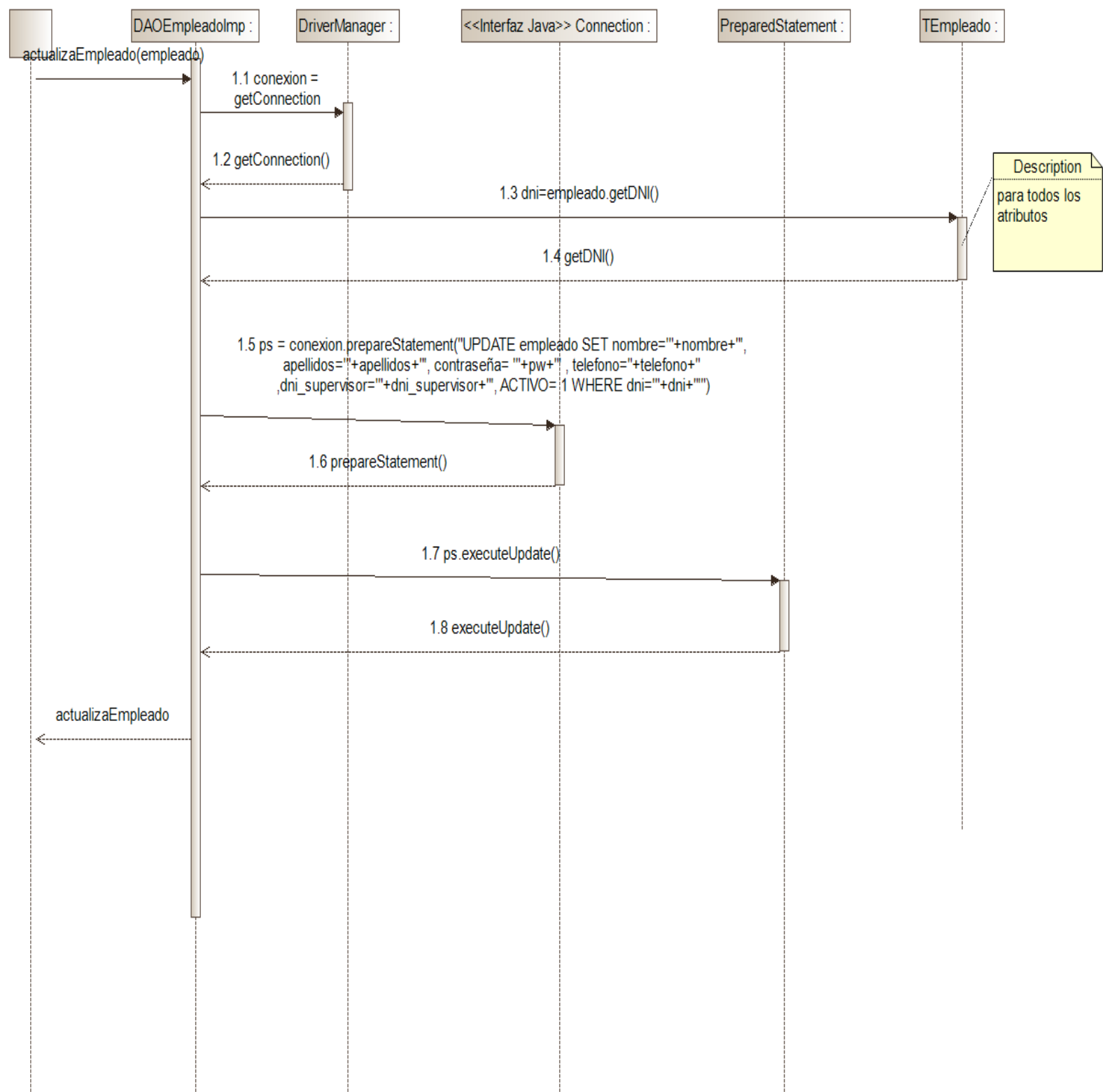
- Leer Todos los Empleados



- Alta Empleado

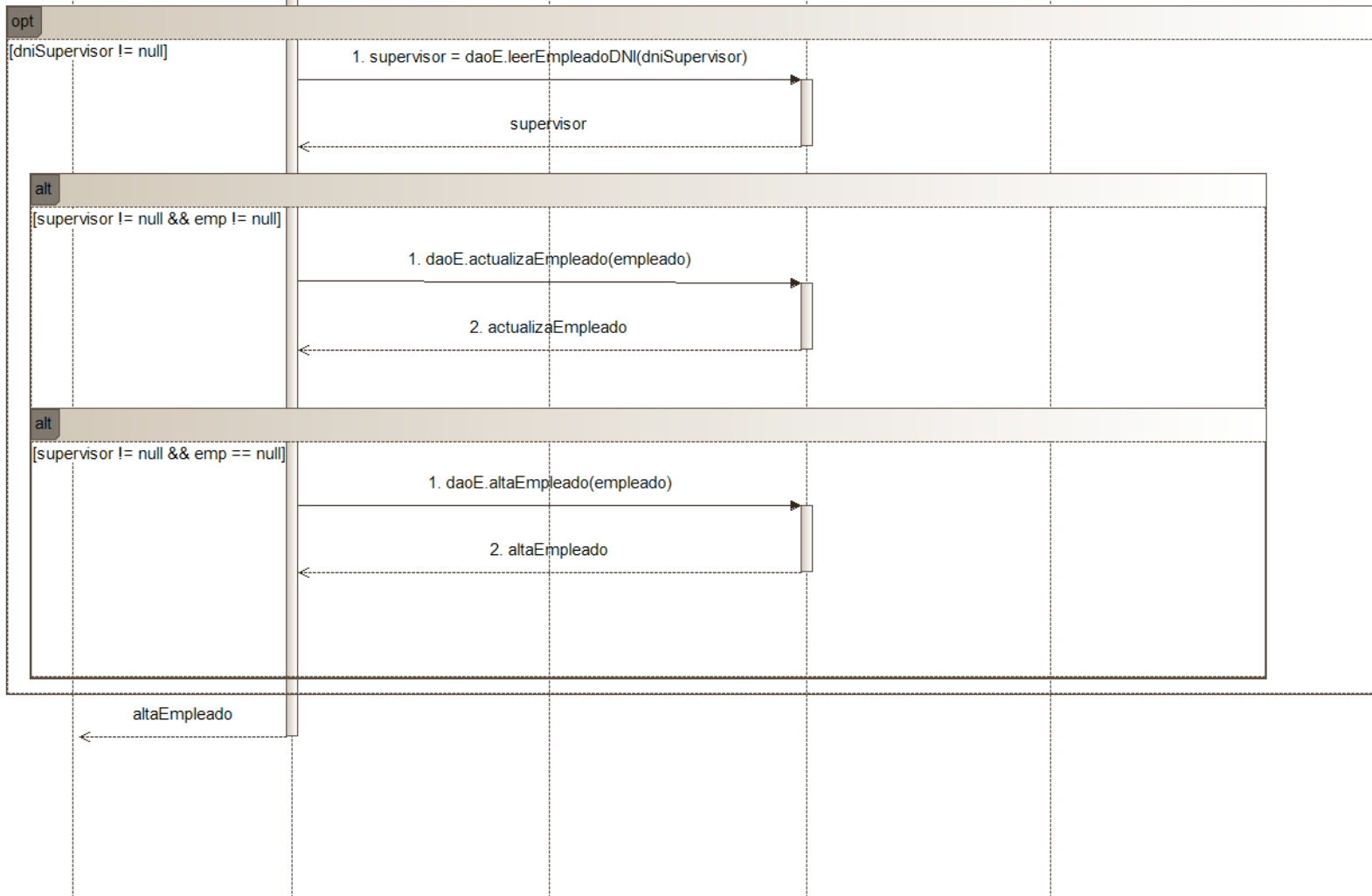
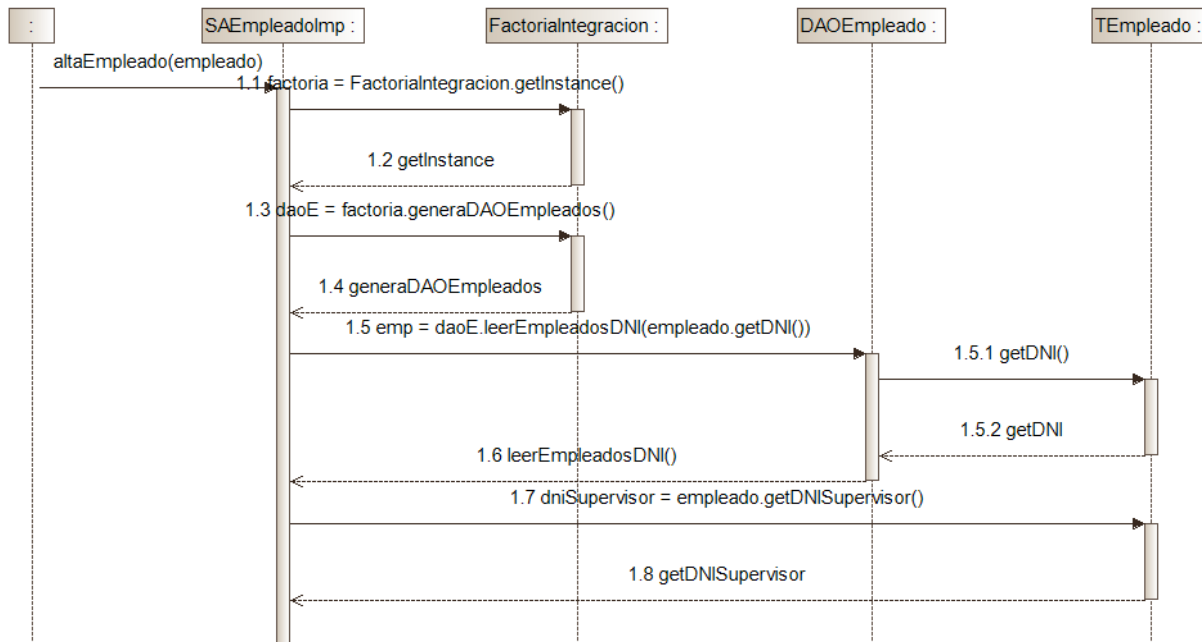


- Actualizar Empleado

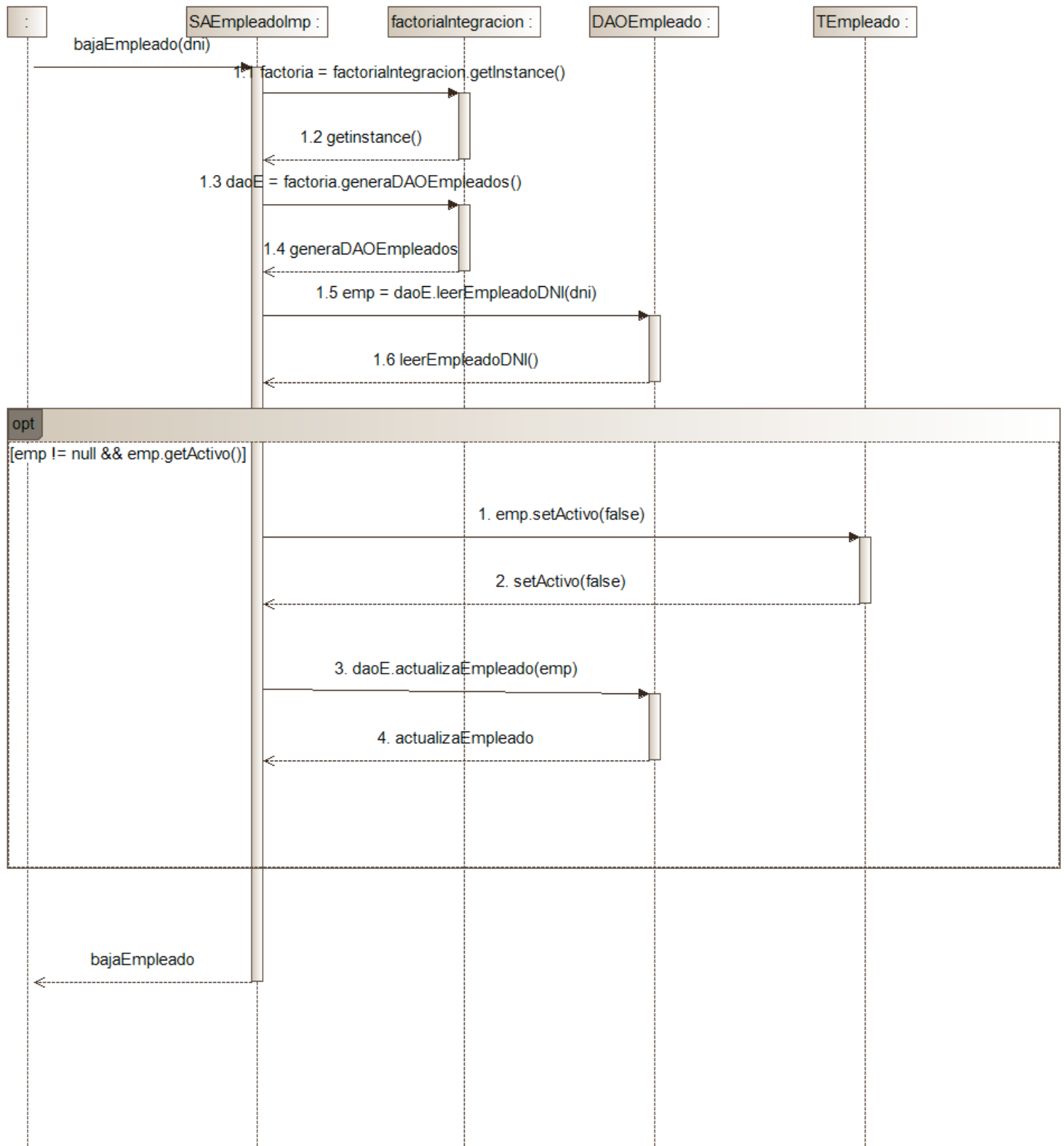


Capa de Negocio

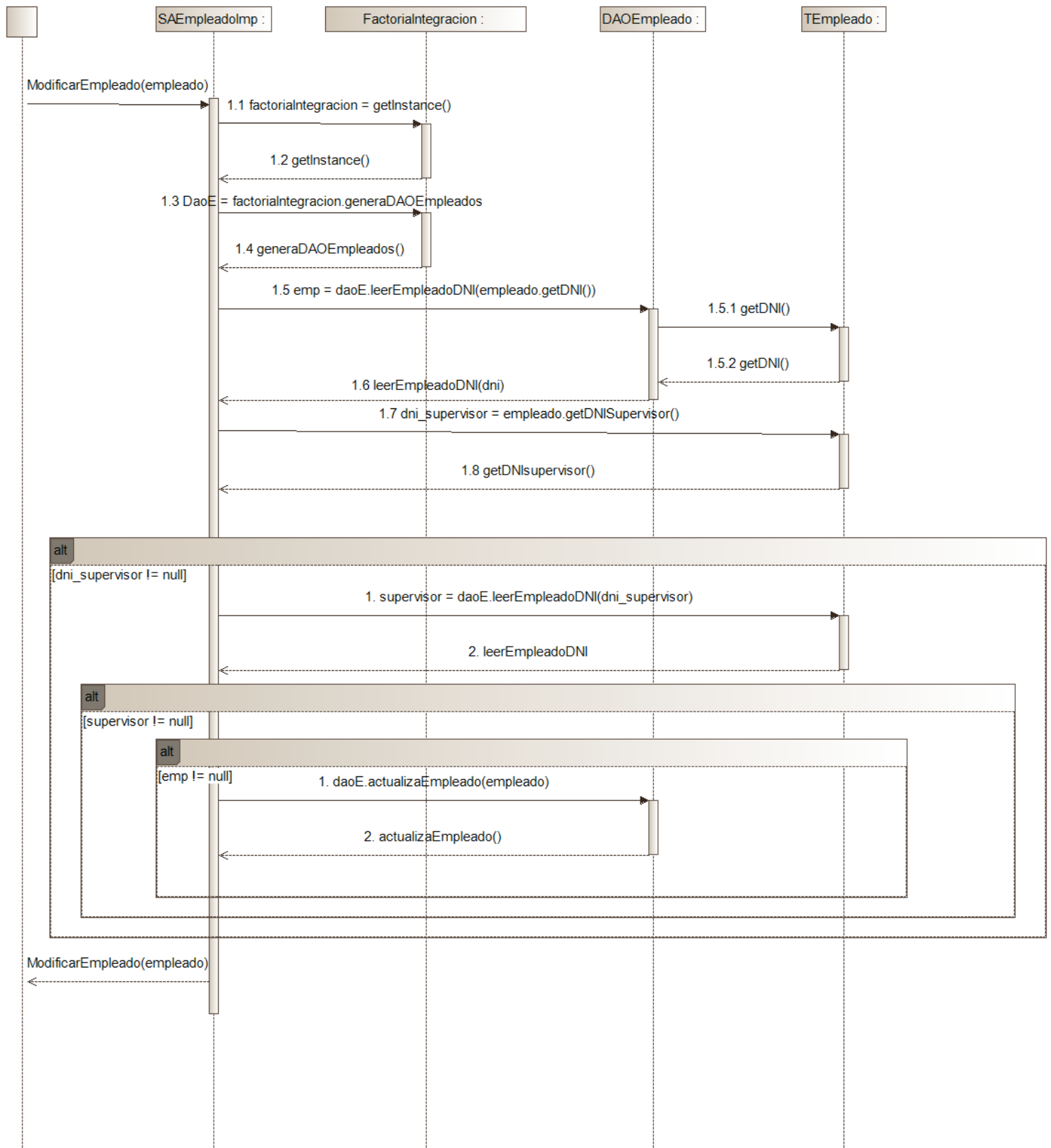
- Alta Empleado SA



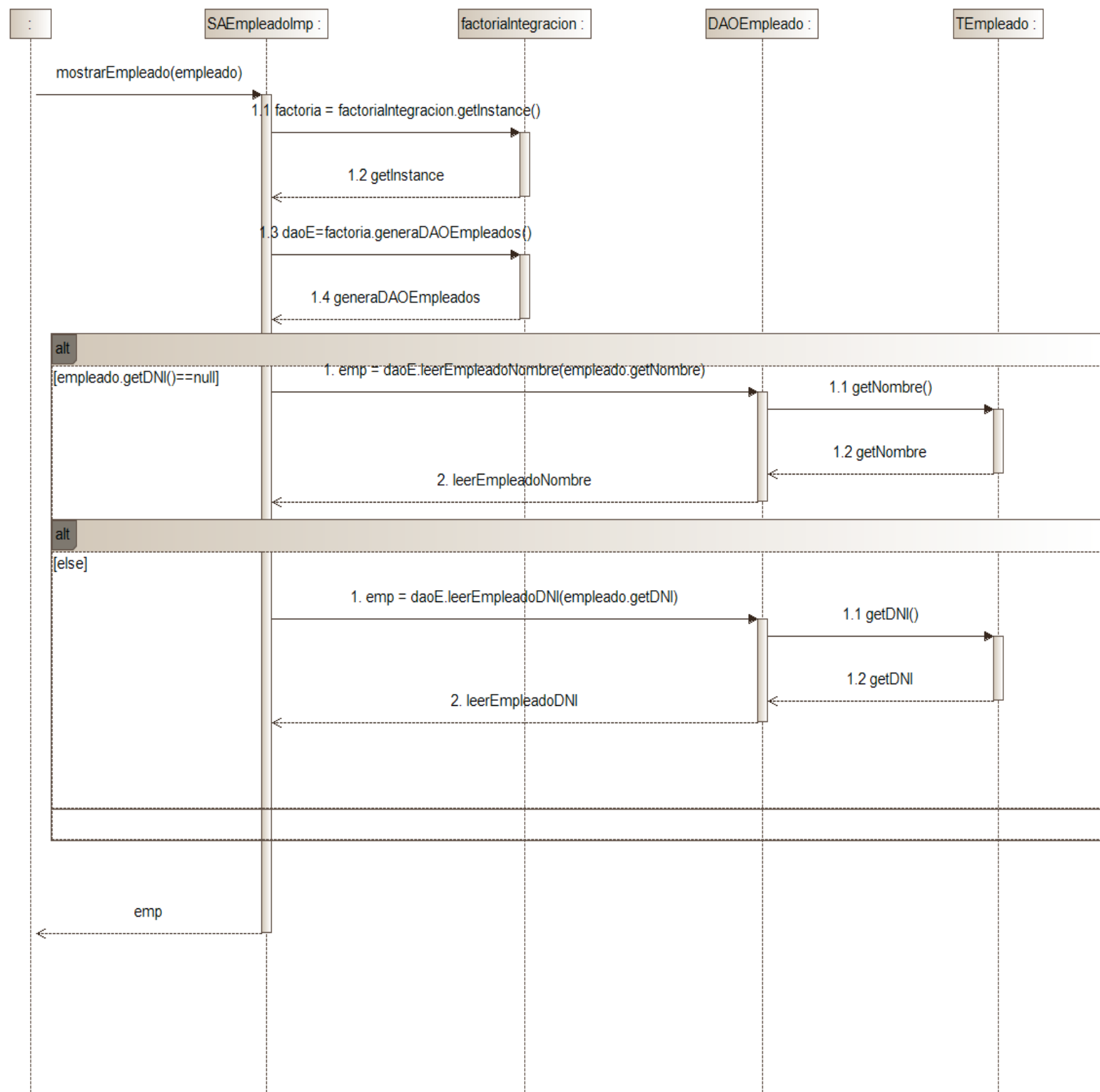
- **Baja Empleado SA**



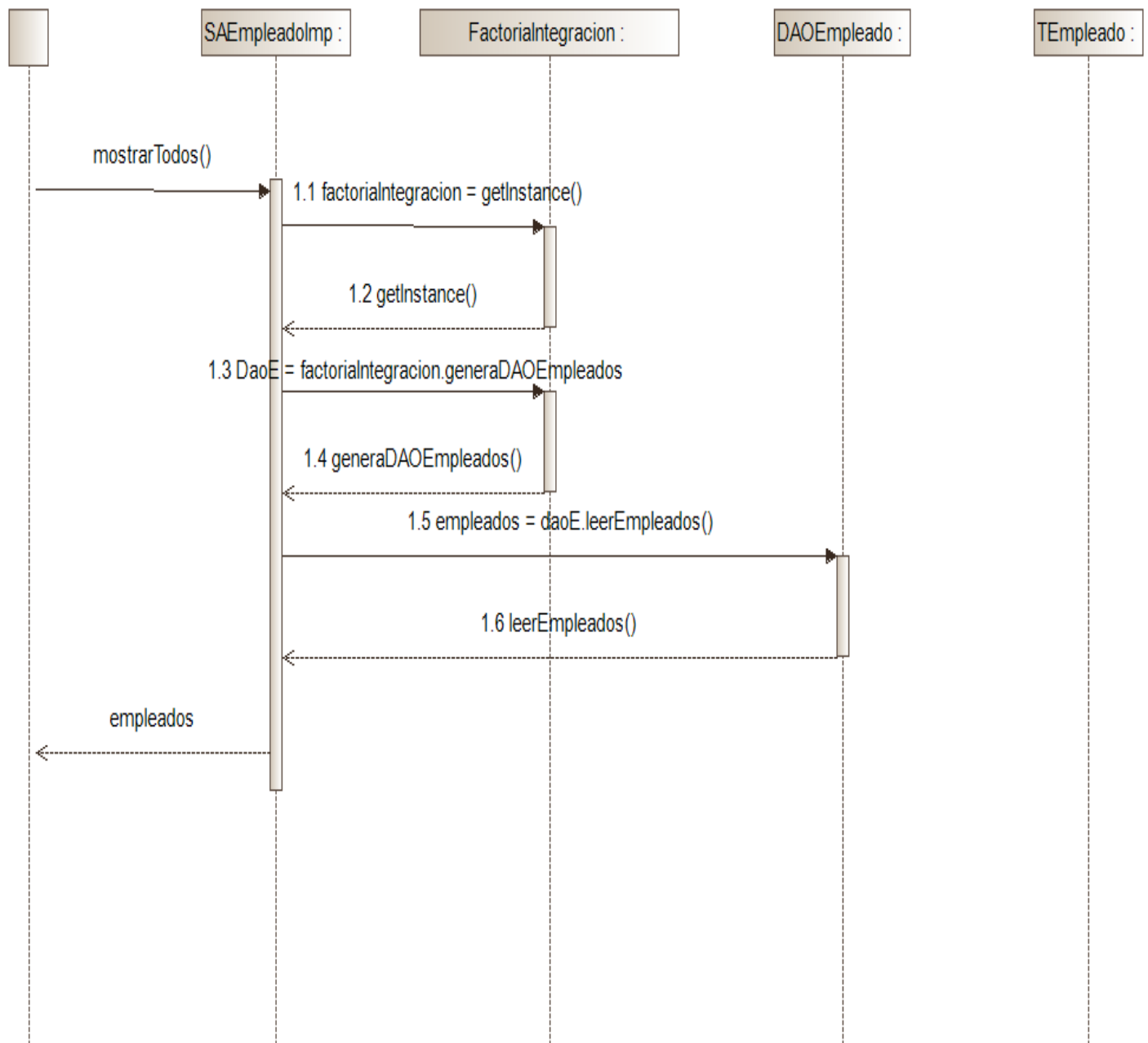
- Modificar Empleado SA



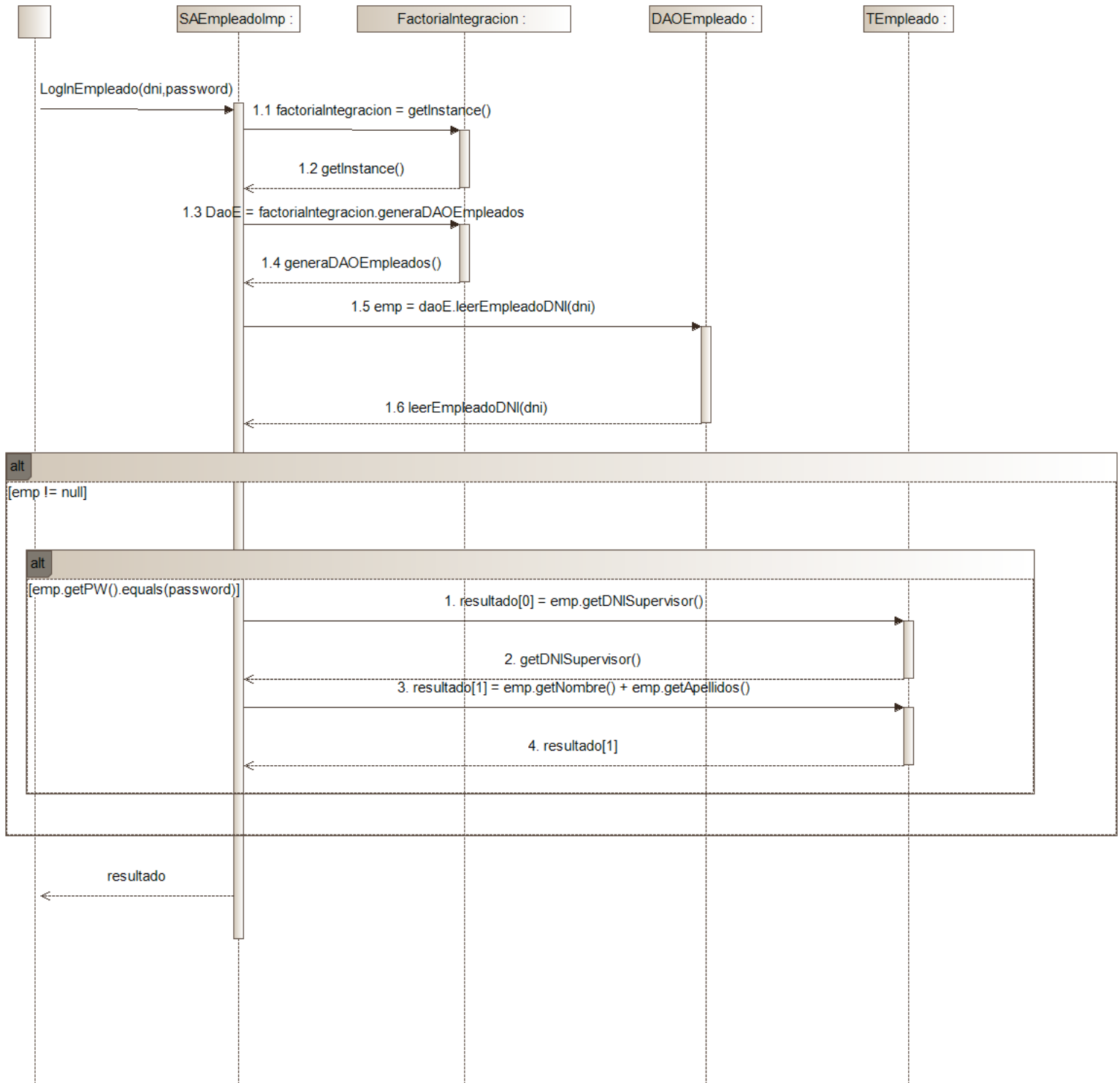
- **Mostrar Empleado SA**



- **Mostrar Todos Empleado SA**

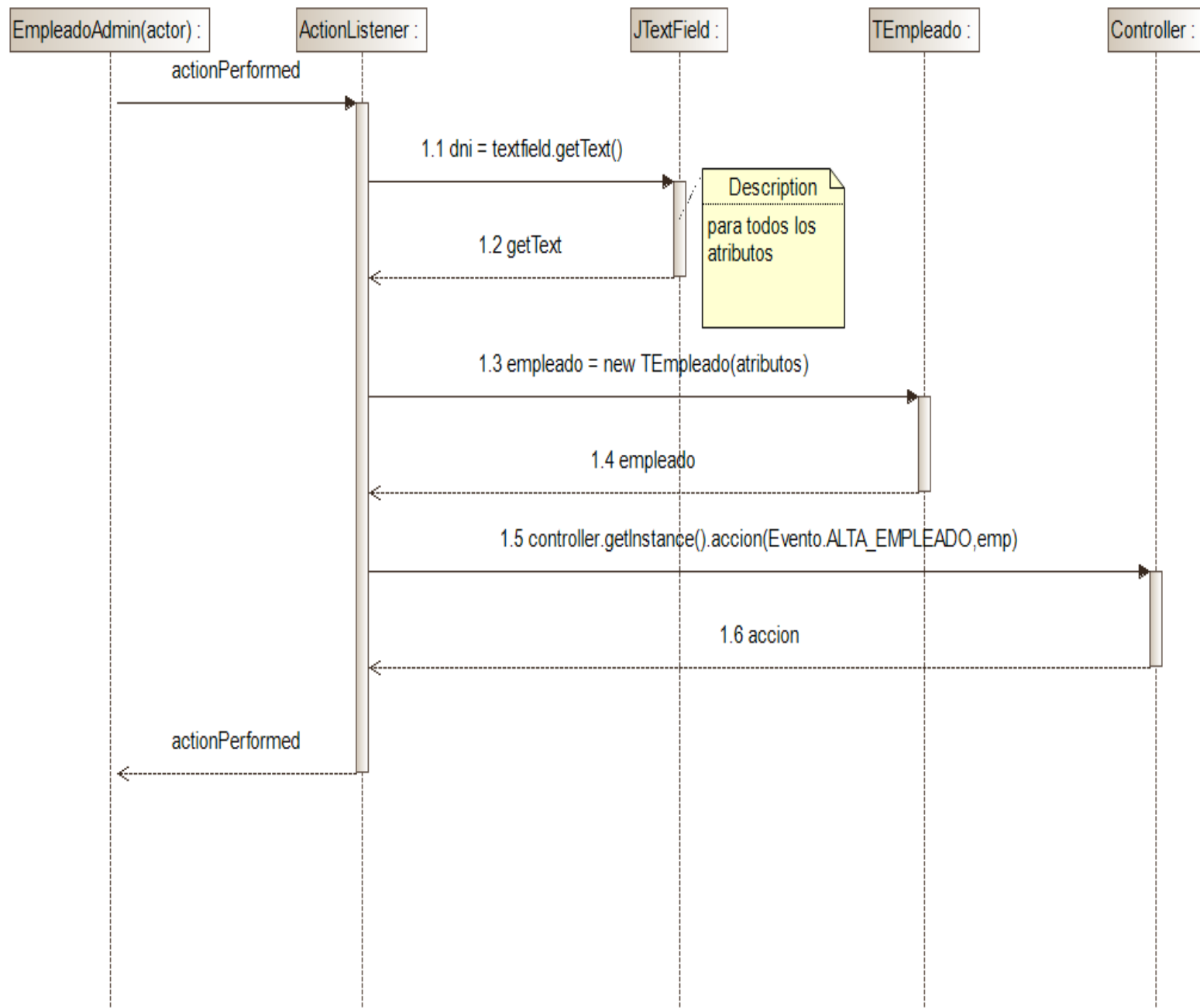


- Login Empleado SA

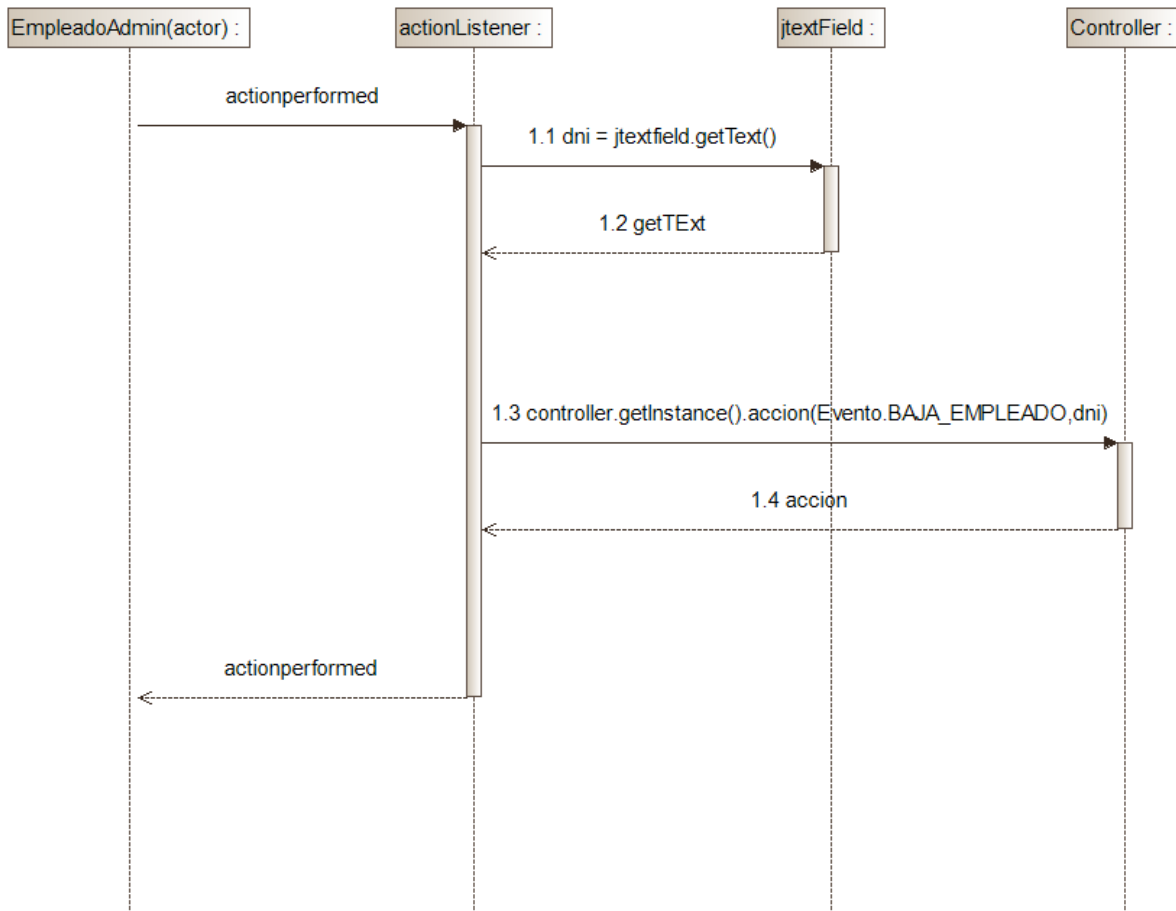


Capa de Presentación

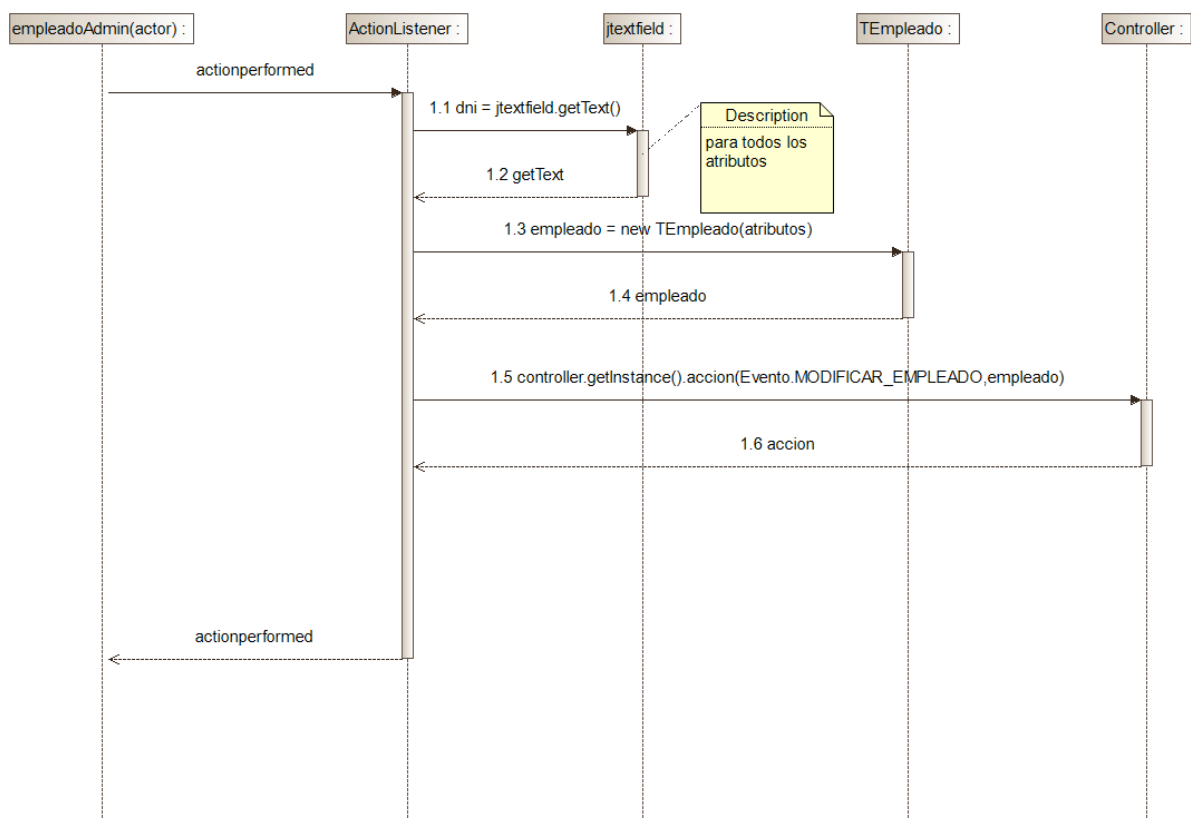
- Alta Empleado Action Listener



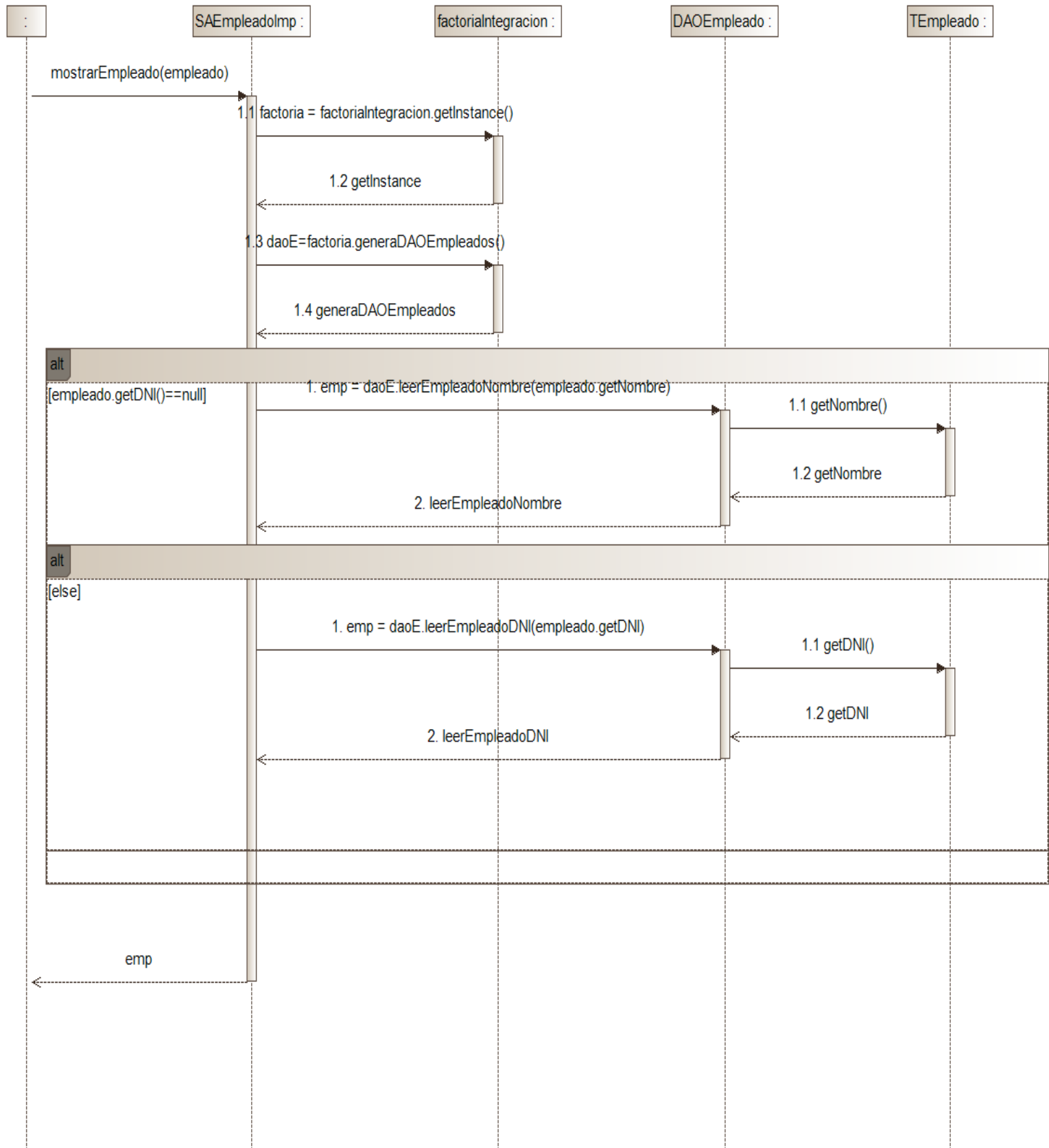
- Baja Empleado Action Listener



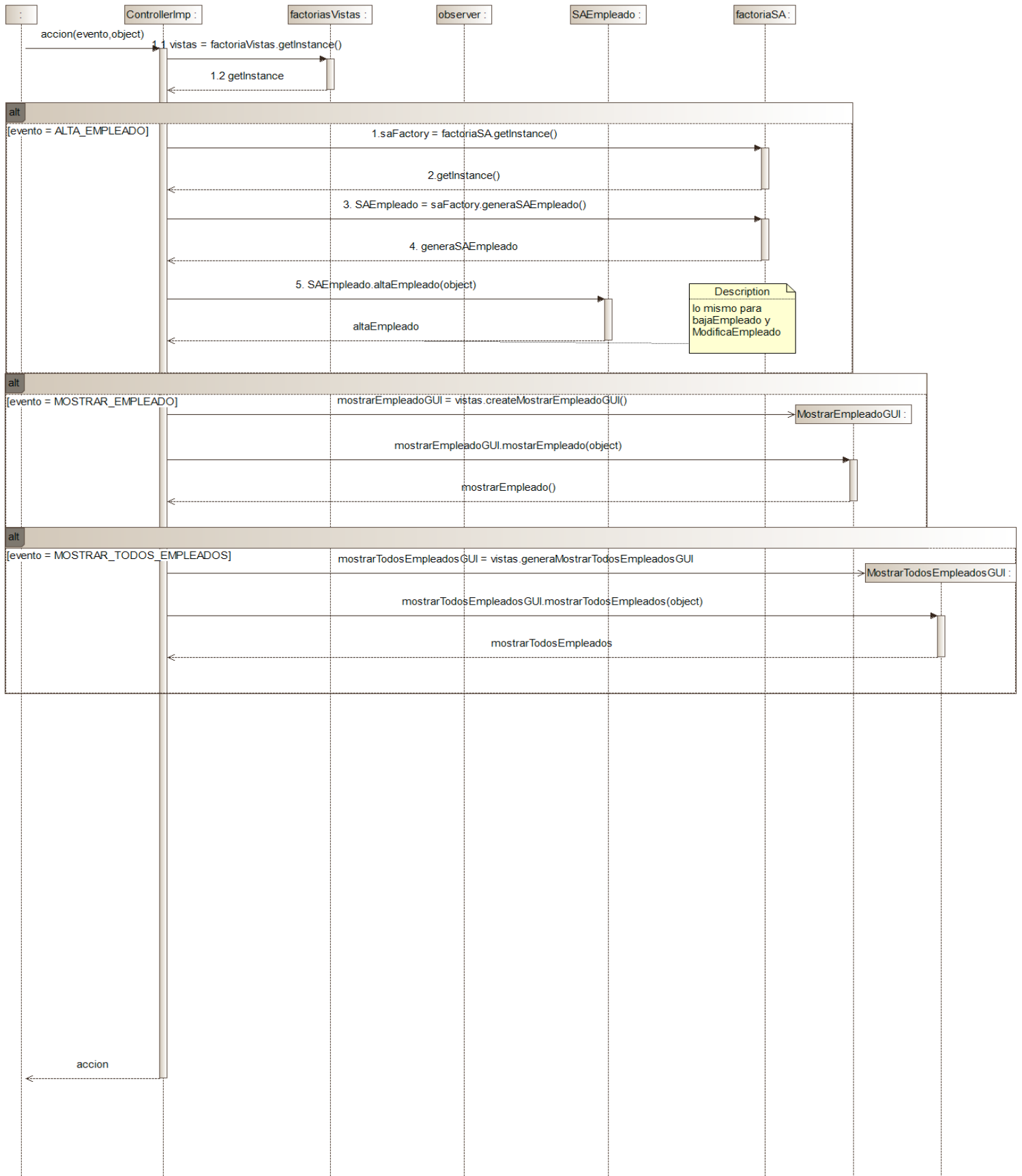
- Modificar Empleado Action Listener



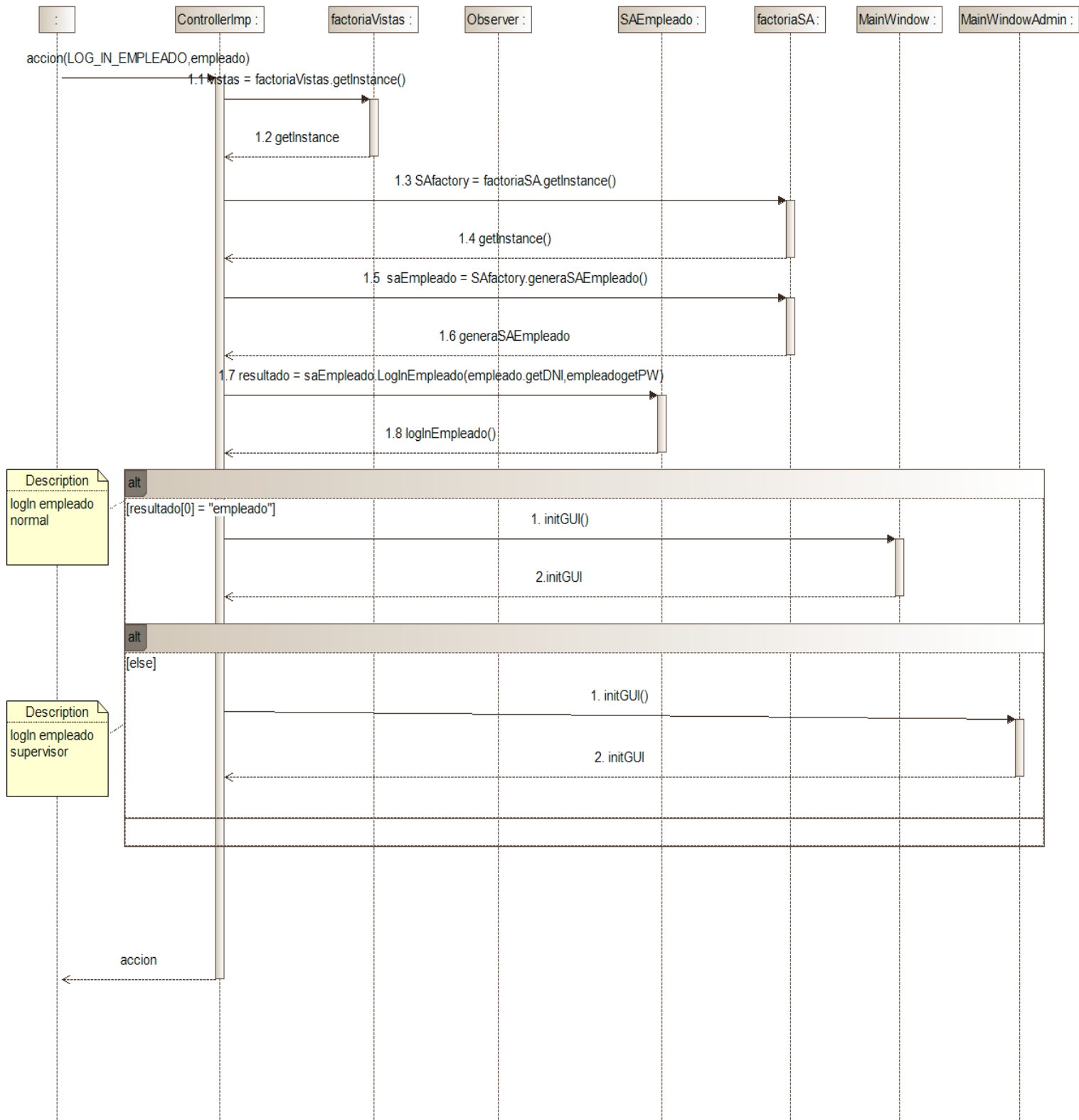
- **Mostrar Empleado Action Listener**



- Controlador Empleado



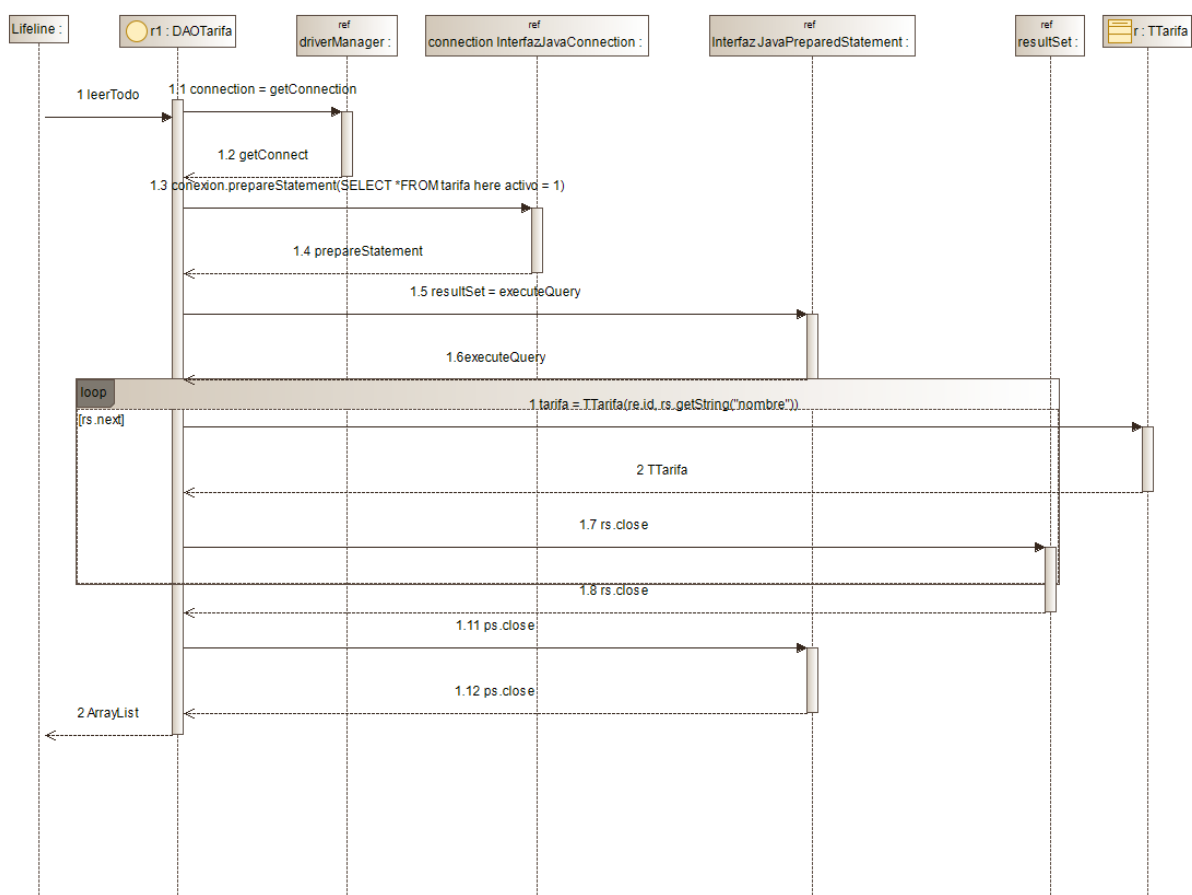
- Controlador Empleado - Login



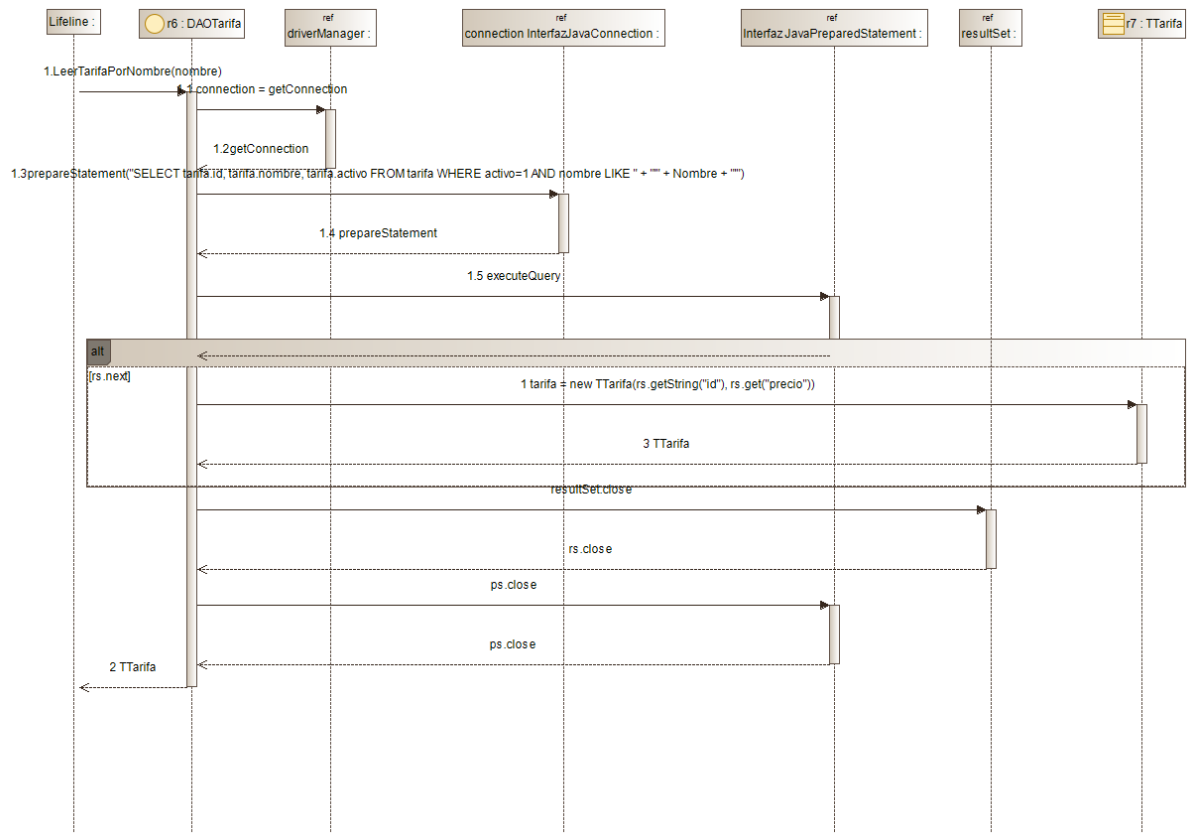
Tarifa

Capa de Integración

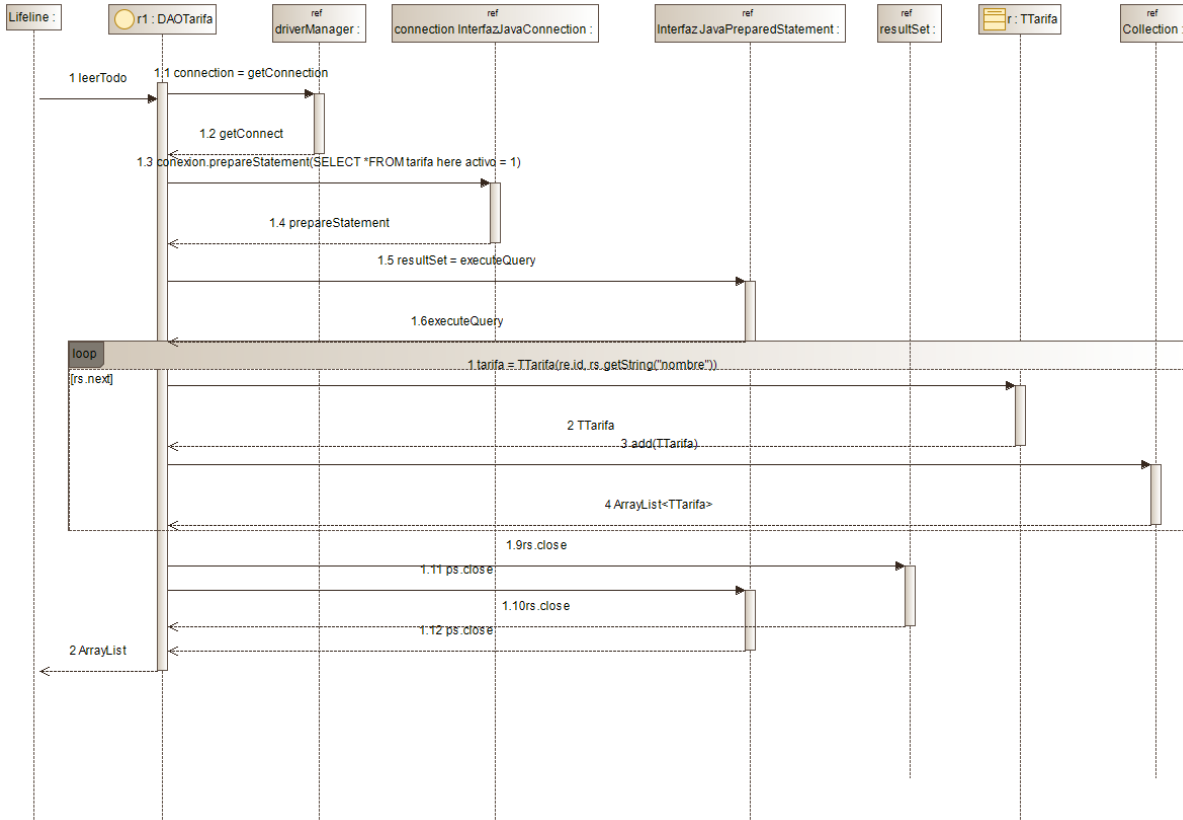
- Leer Tarifa por ID



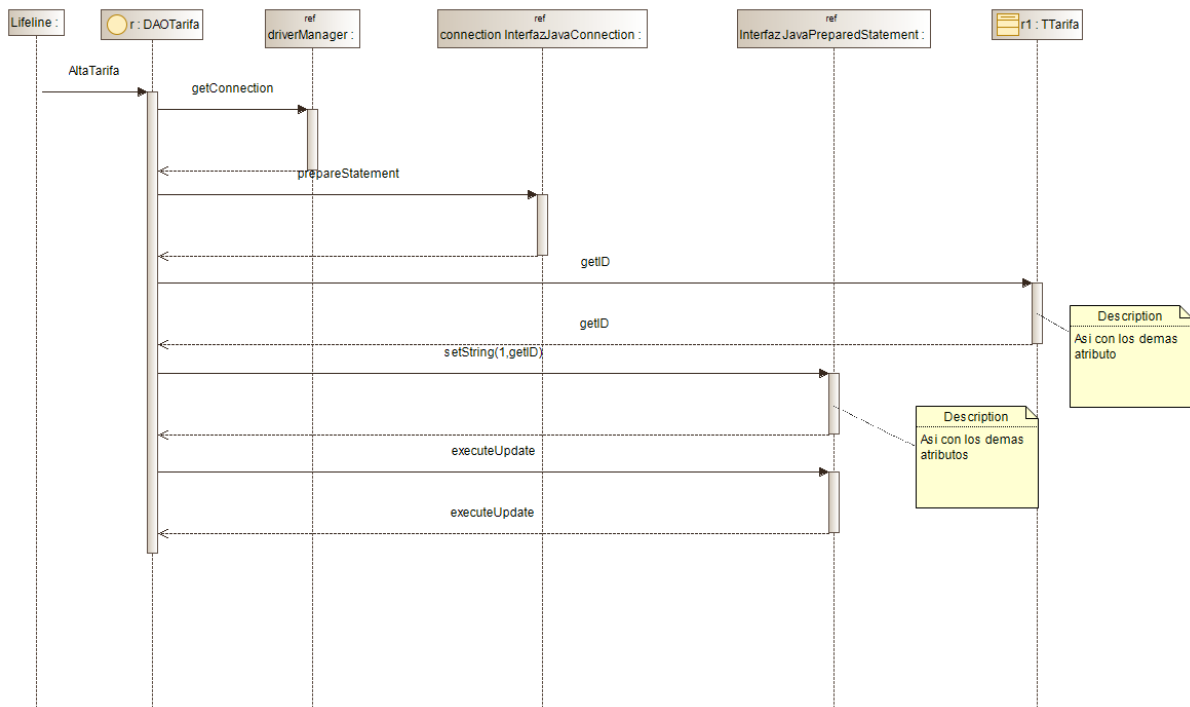
- Leer Tarifa por Nombre



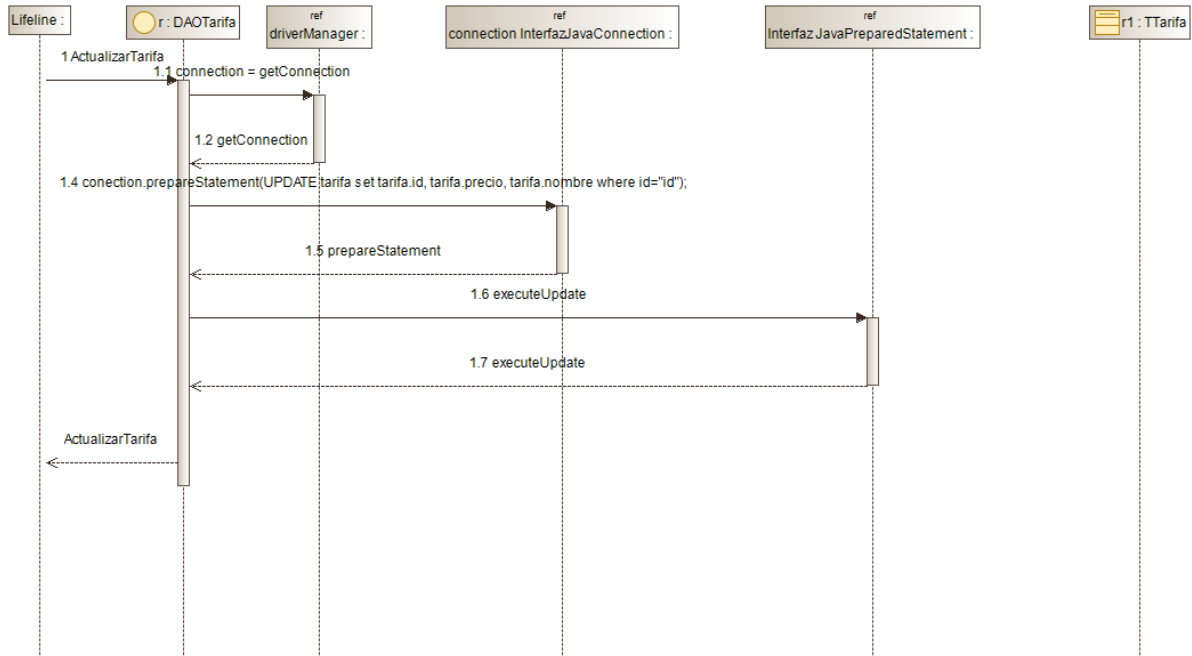
- Leer Todos Tarifa



- Alta Tarifa

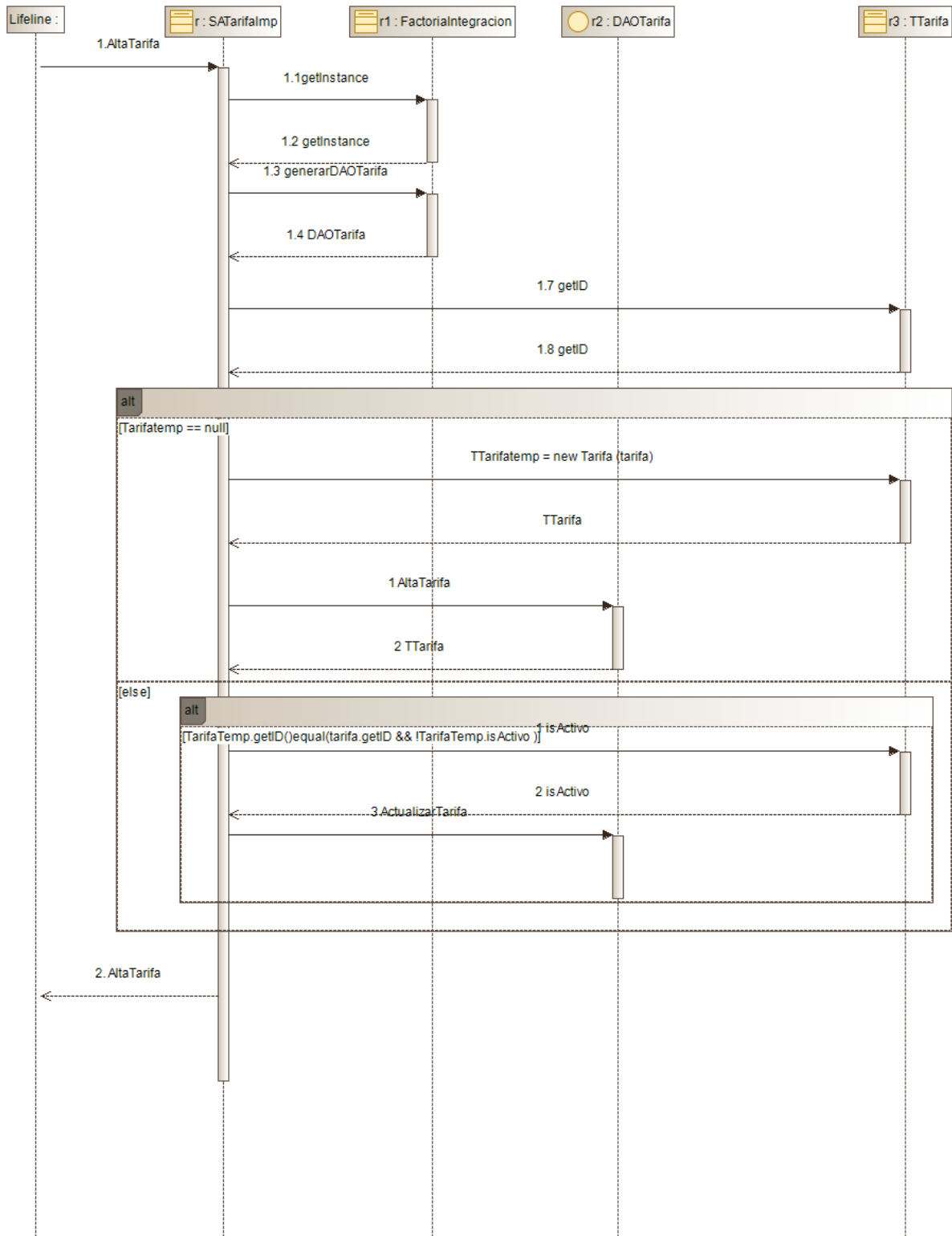


- Actualizar Tarifa

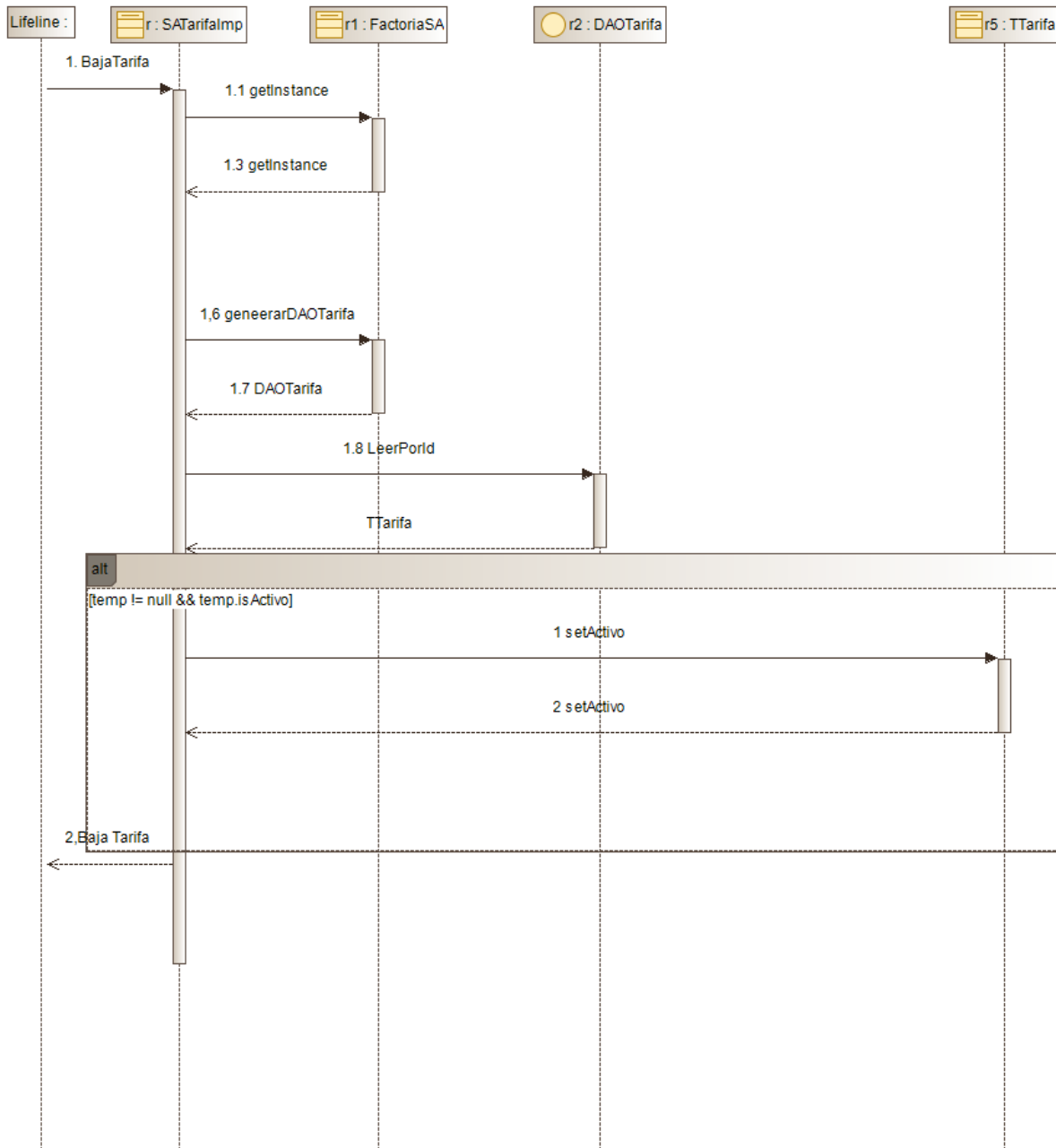


Capa de Negocio

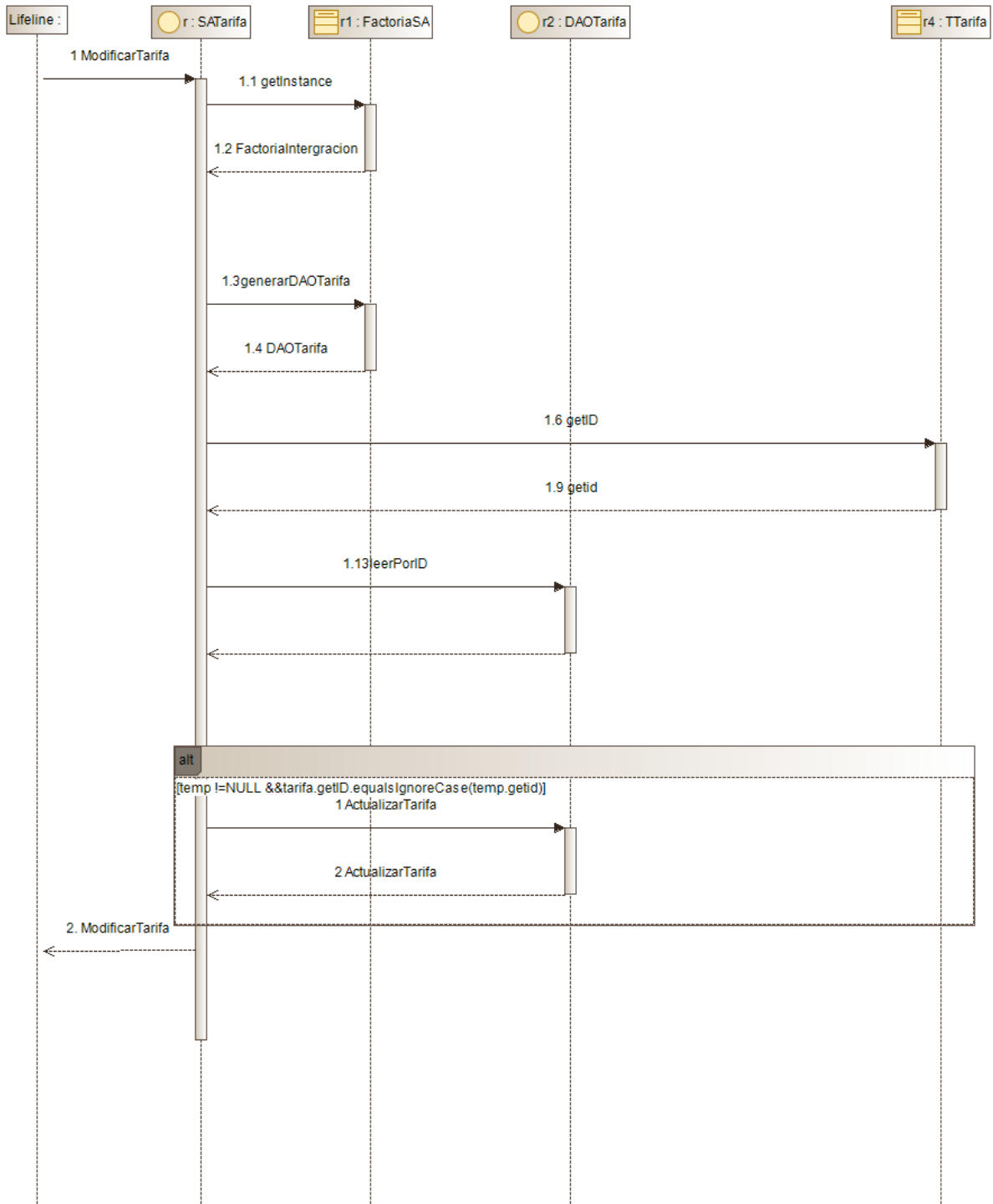
- Alta Tarifa SA



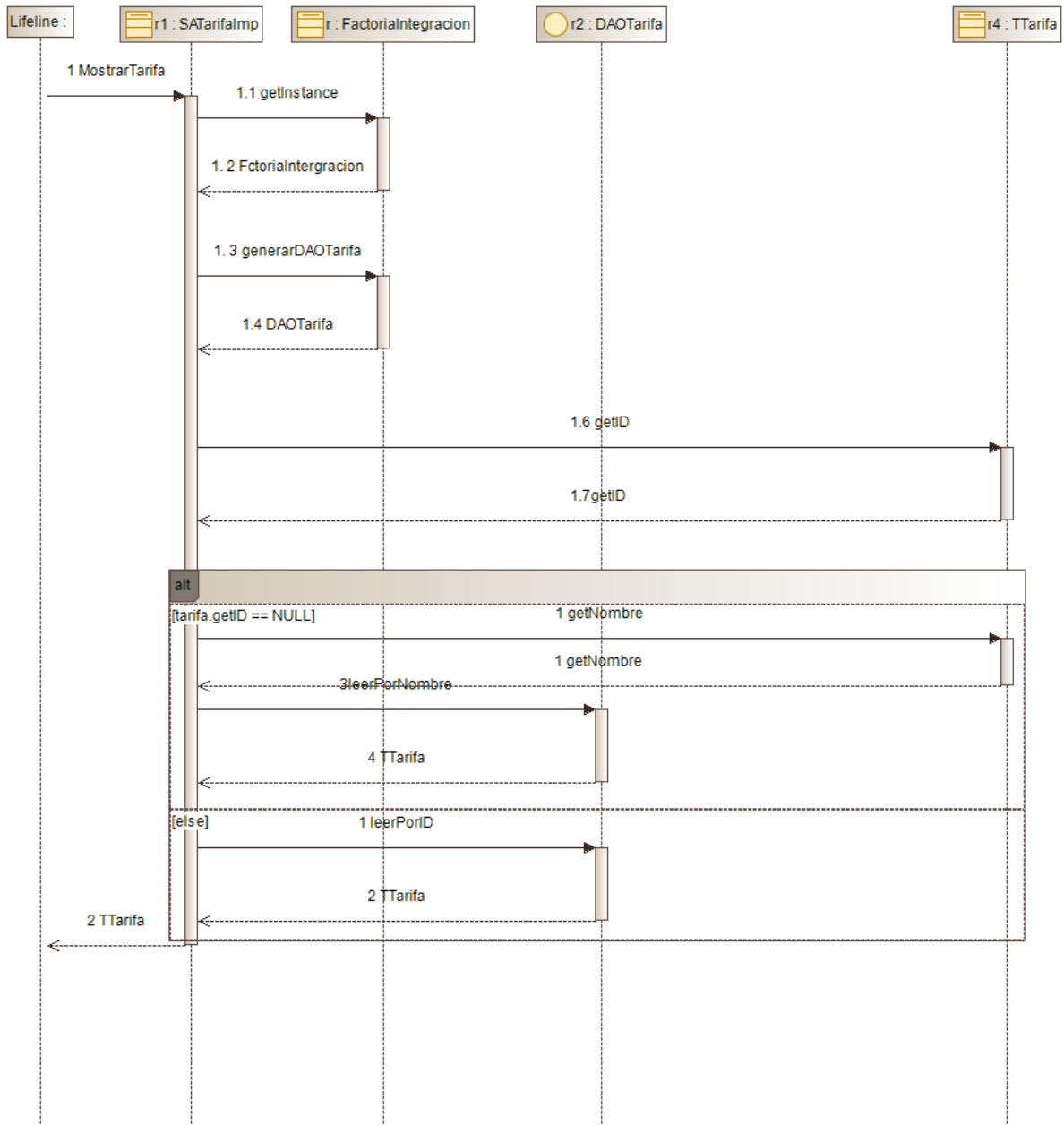
- Baja Tarifa SA



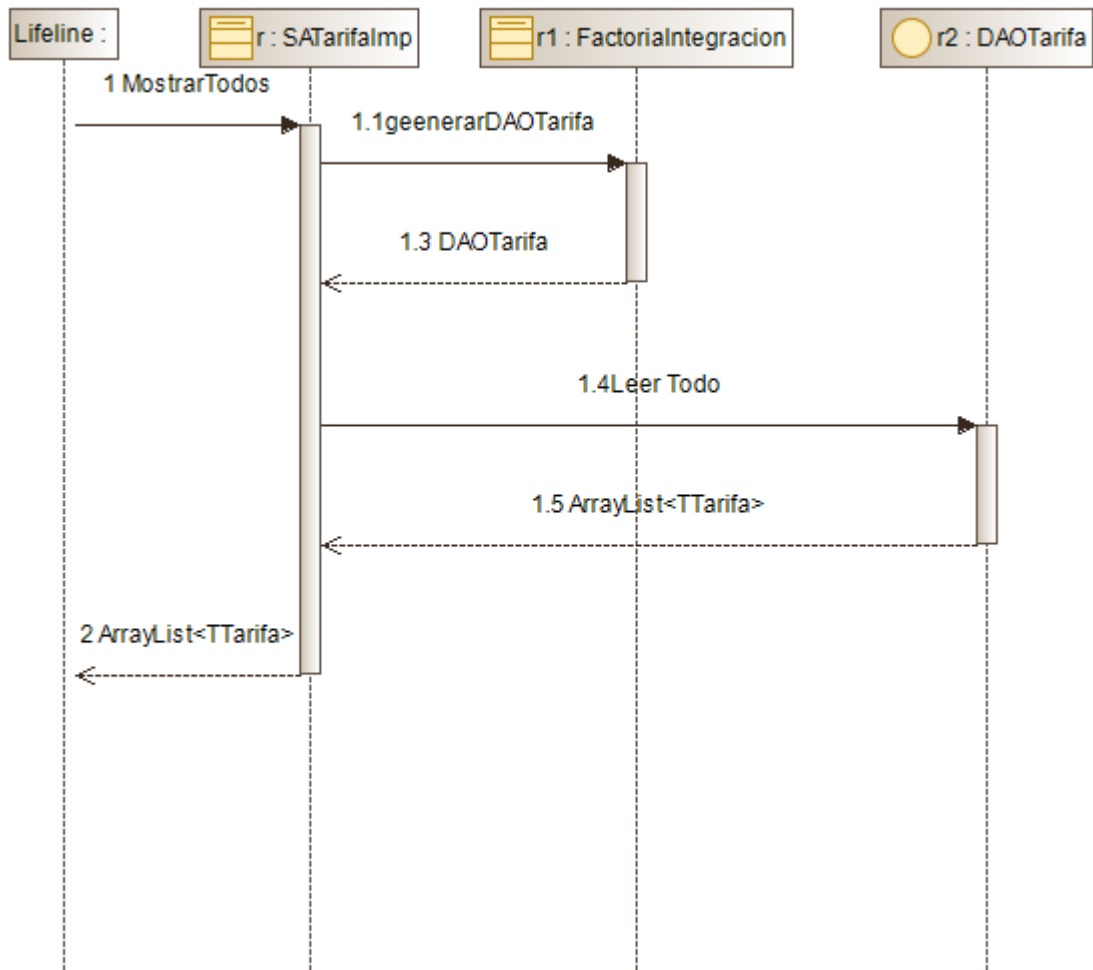
- Modificar Tarifa SA



- *Mostrar Tarifa SA*

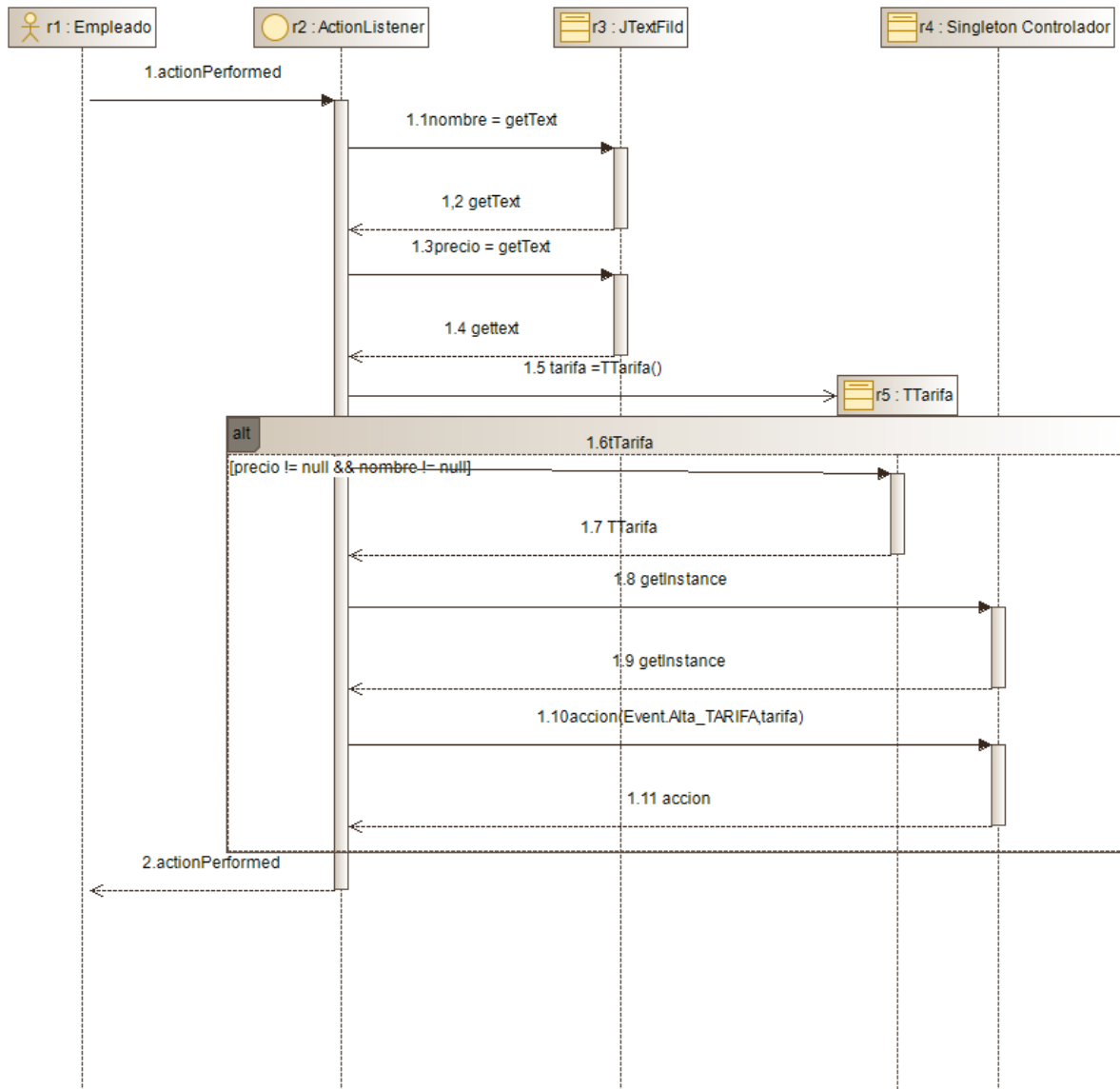


- **Mostrar Todos Tarifa SA**

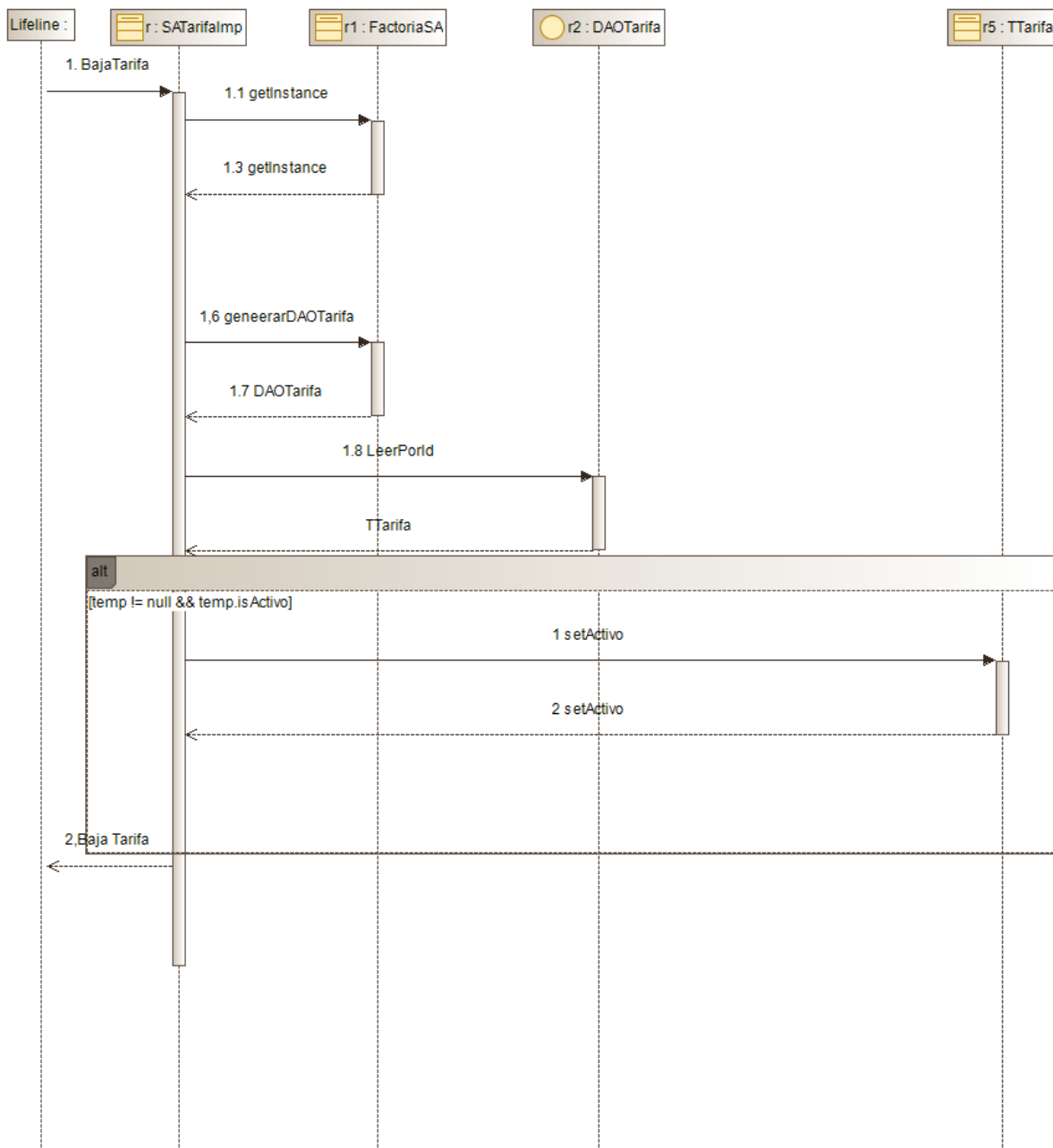


Capa de Presentación

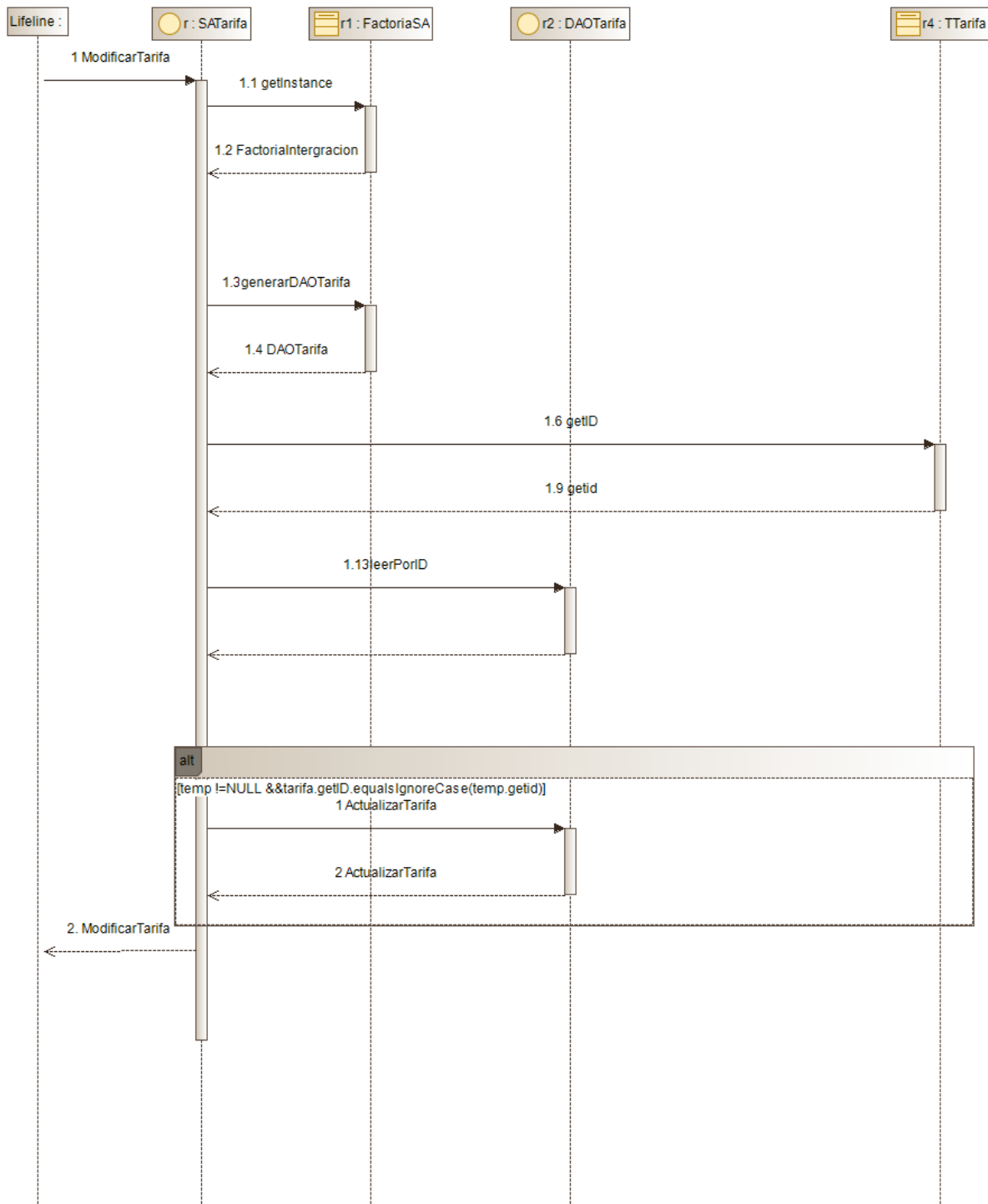
- Alta Tarifa Action Listener



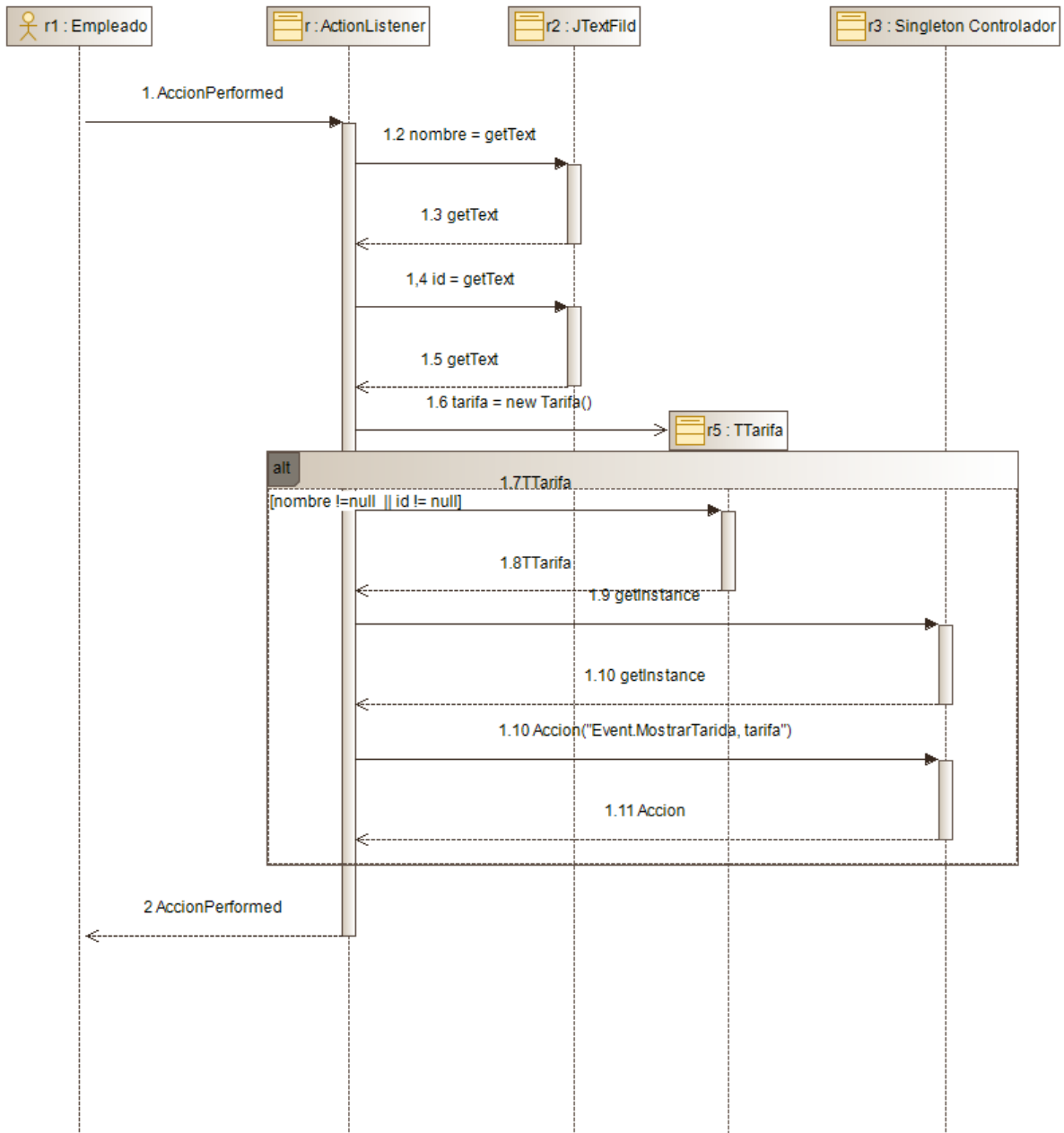
- **Baja Tarifa Action Listener**



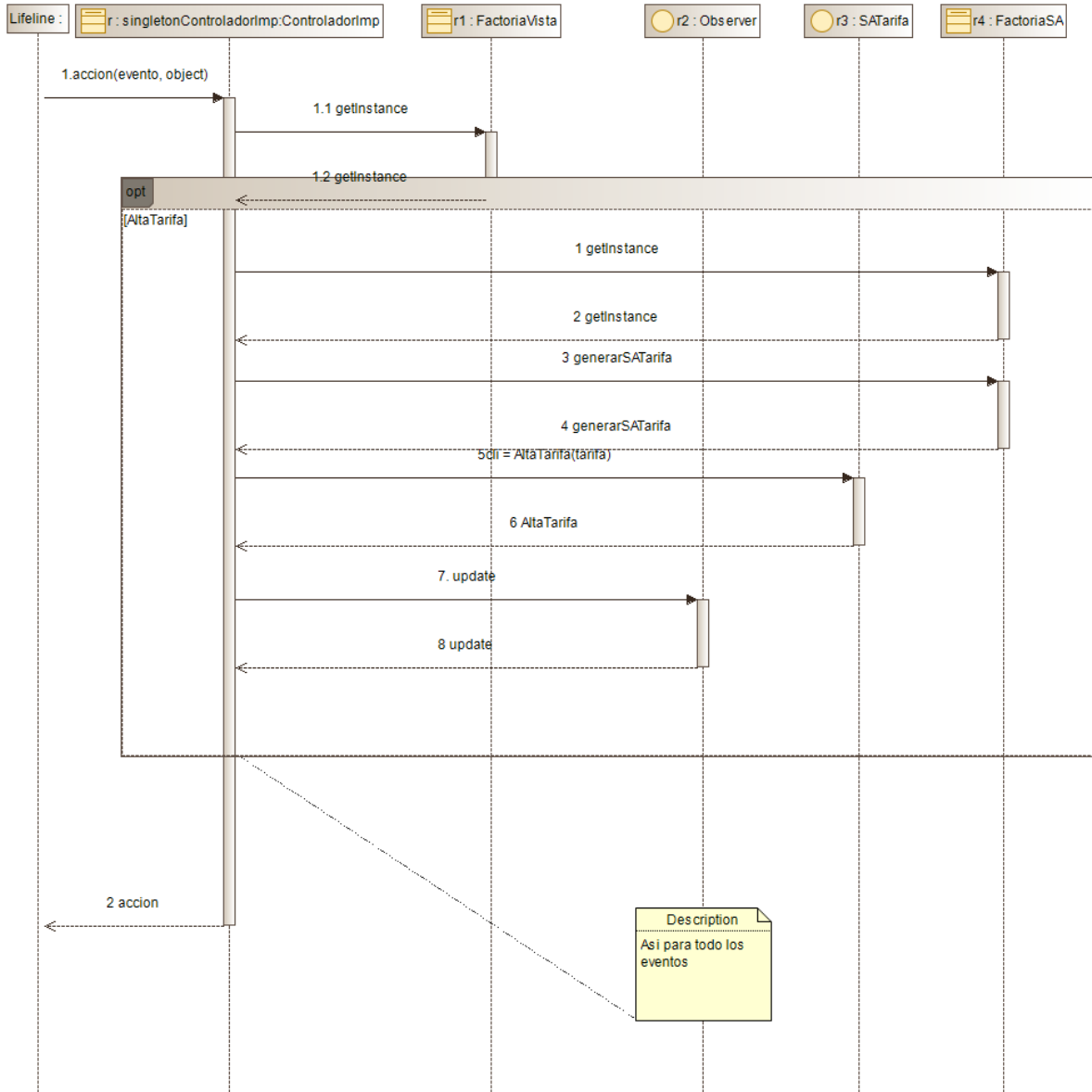
- *Modificar Tarifa Action Listener*



- *Mostrar Tarifa - Action Listener*



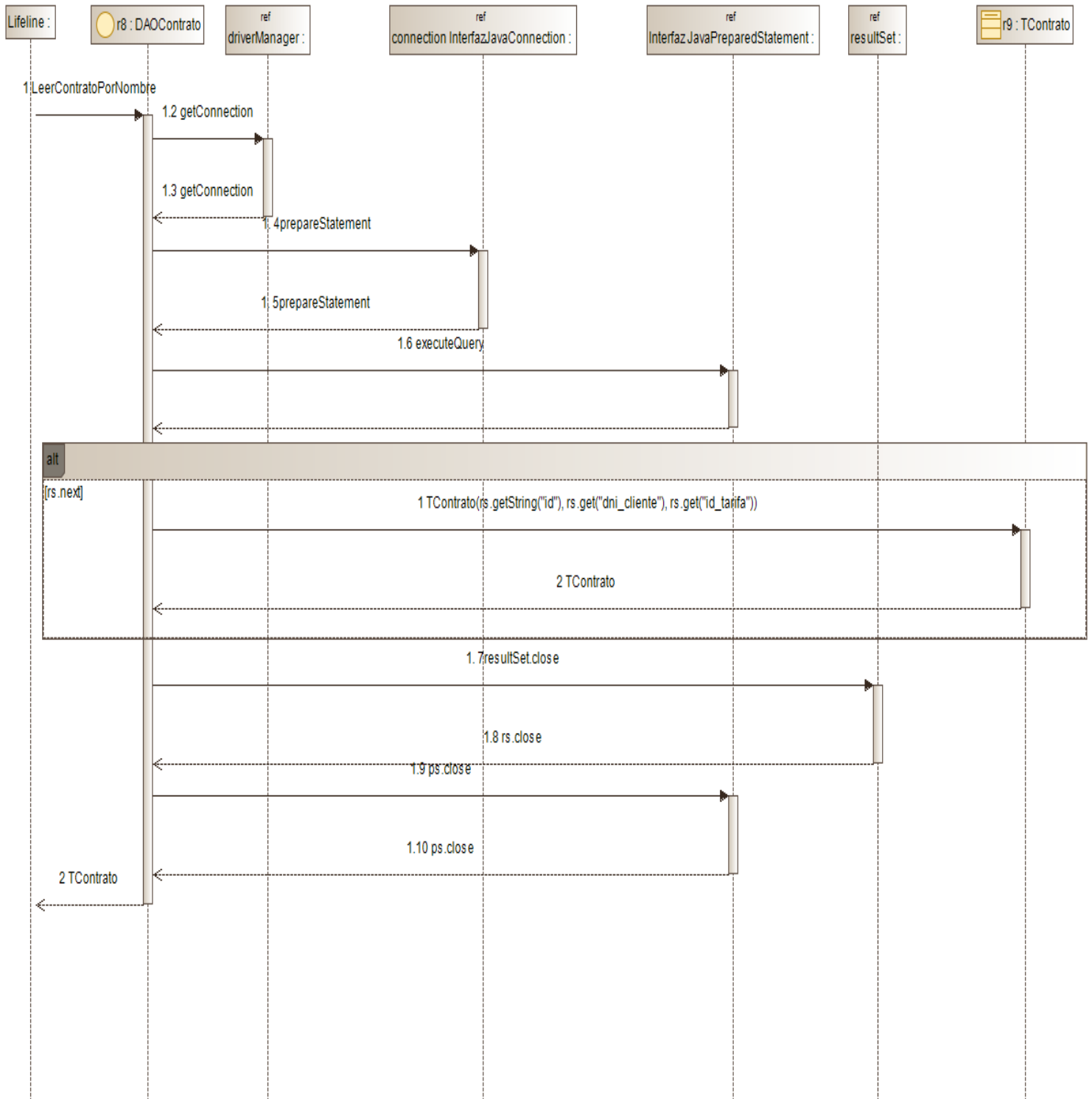
- Controlador Tarifa



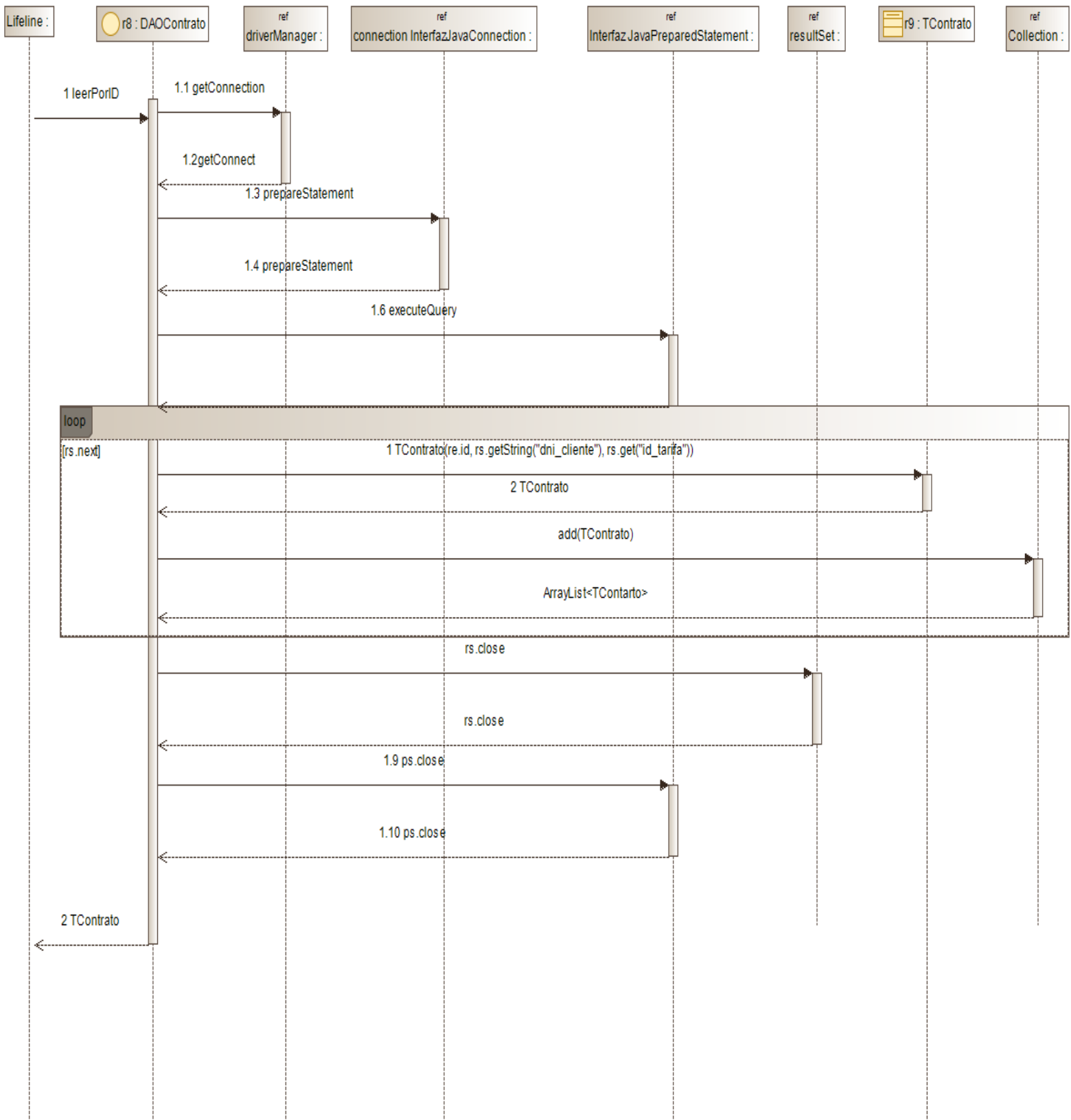
Contrato

Capa de Integración

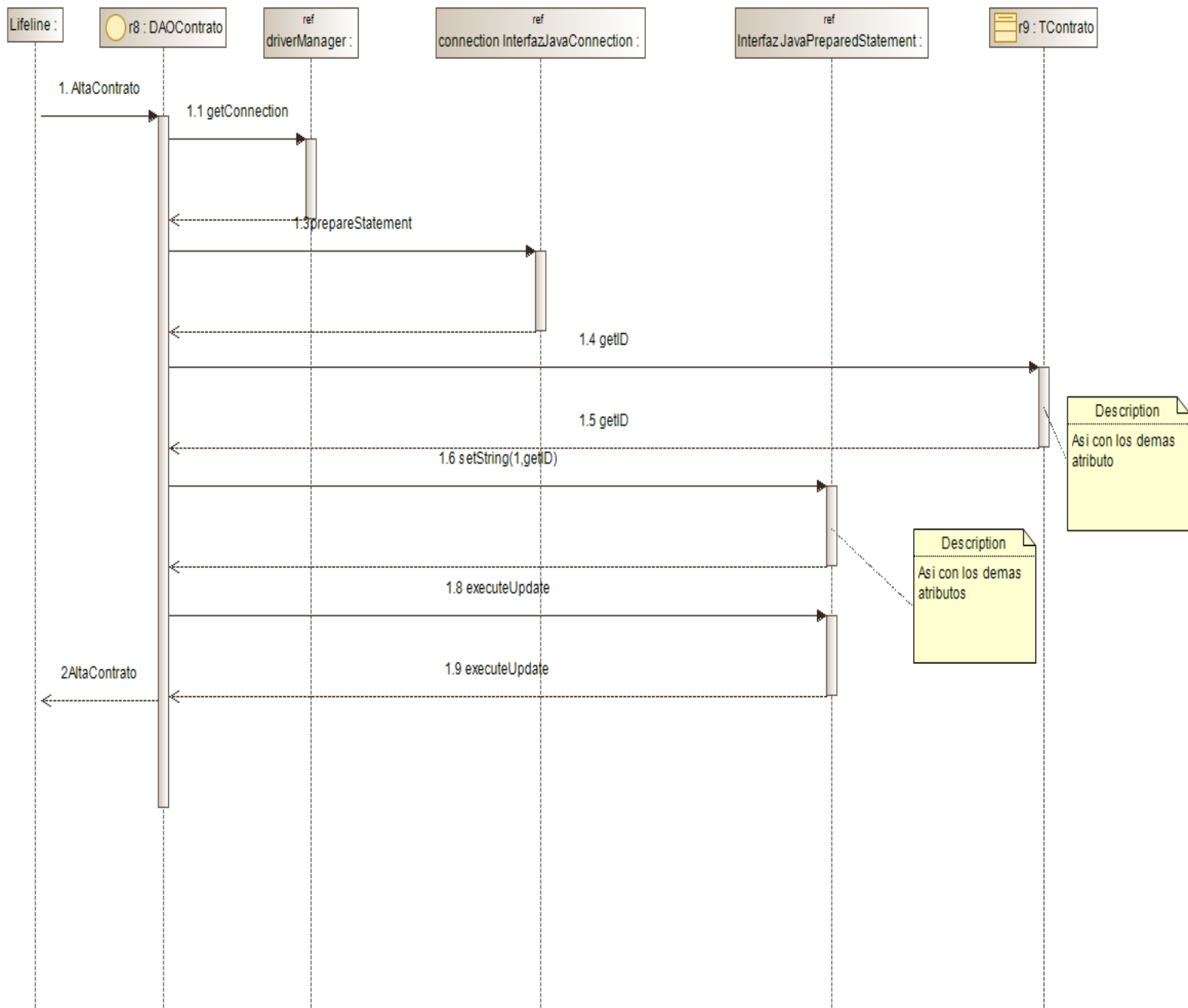
- Leer Contrato por ID



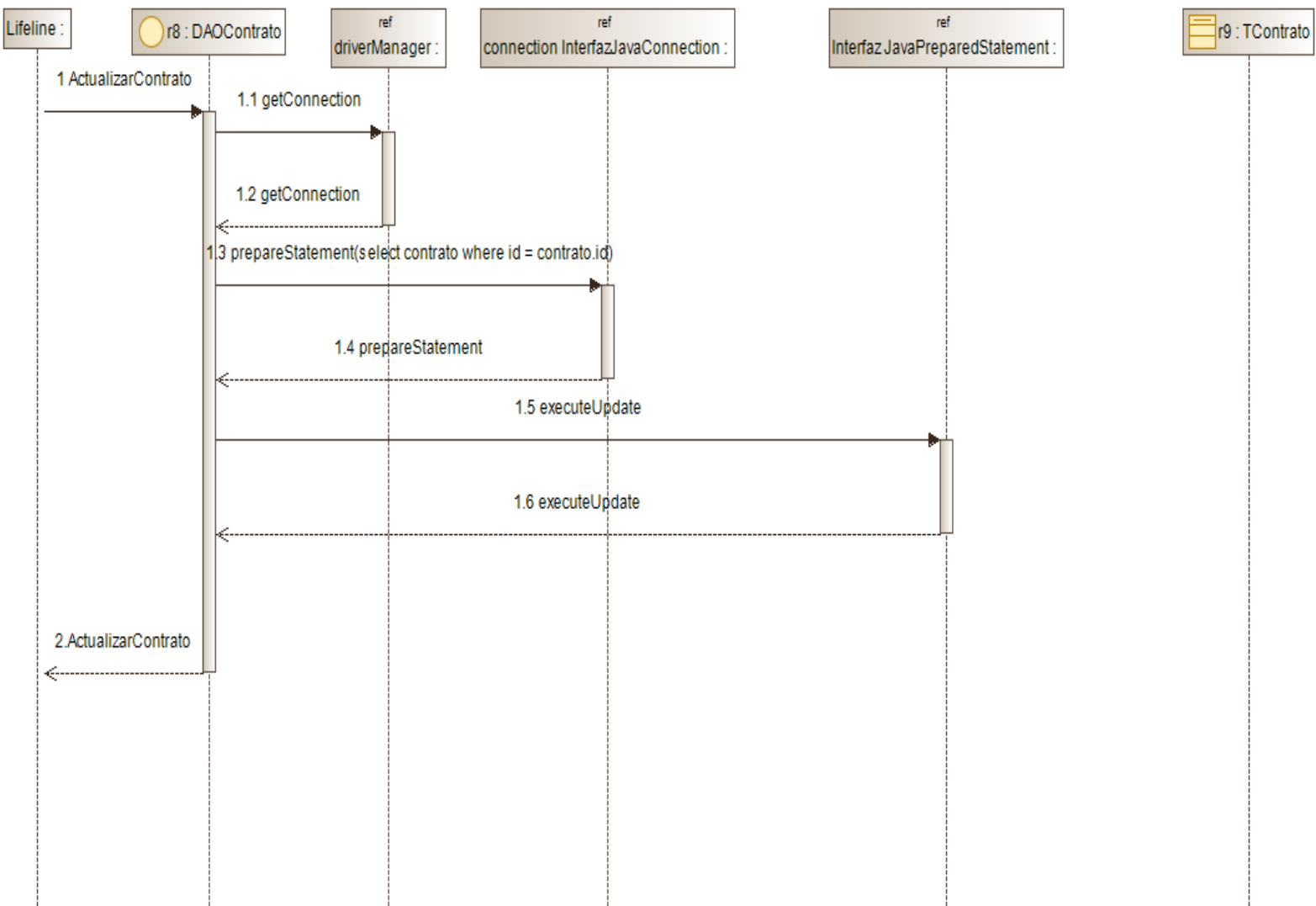
- Leer Todos Contrato



- Alta Contrato

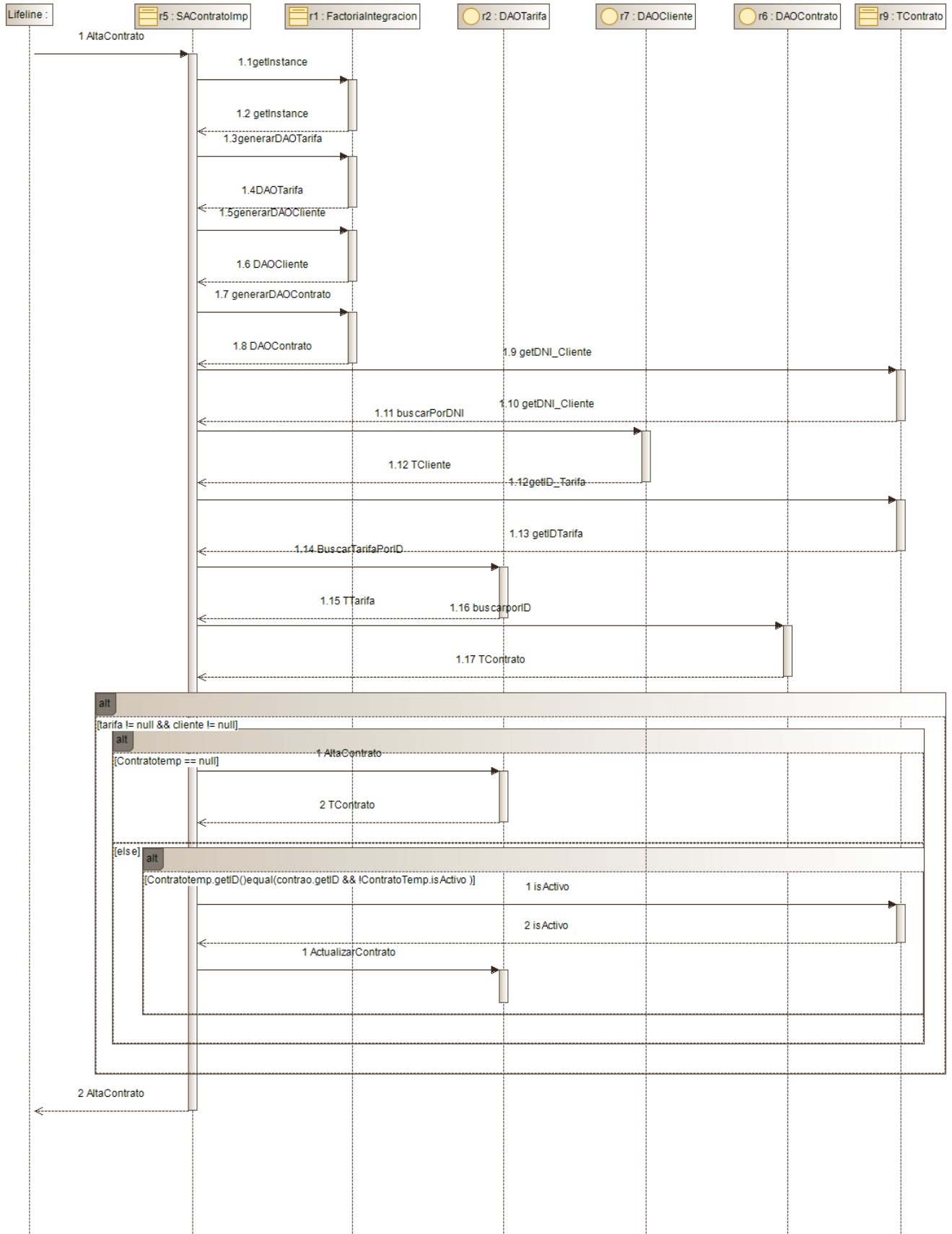


- Actualizar Contrato

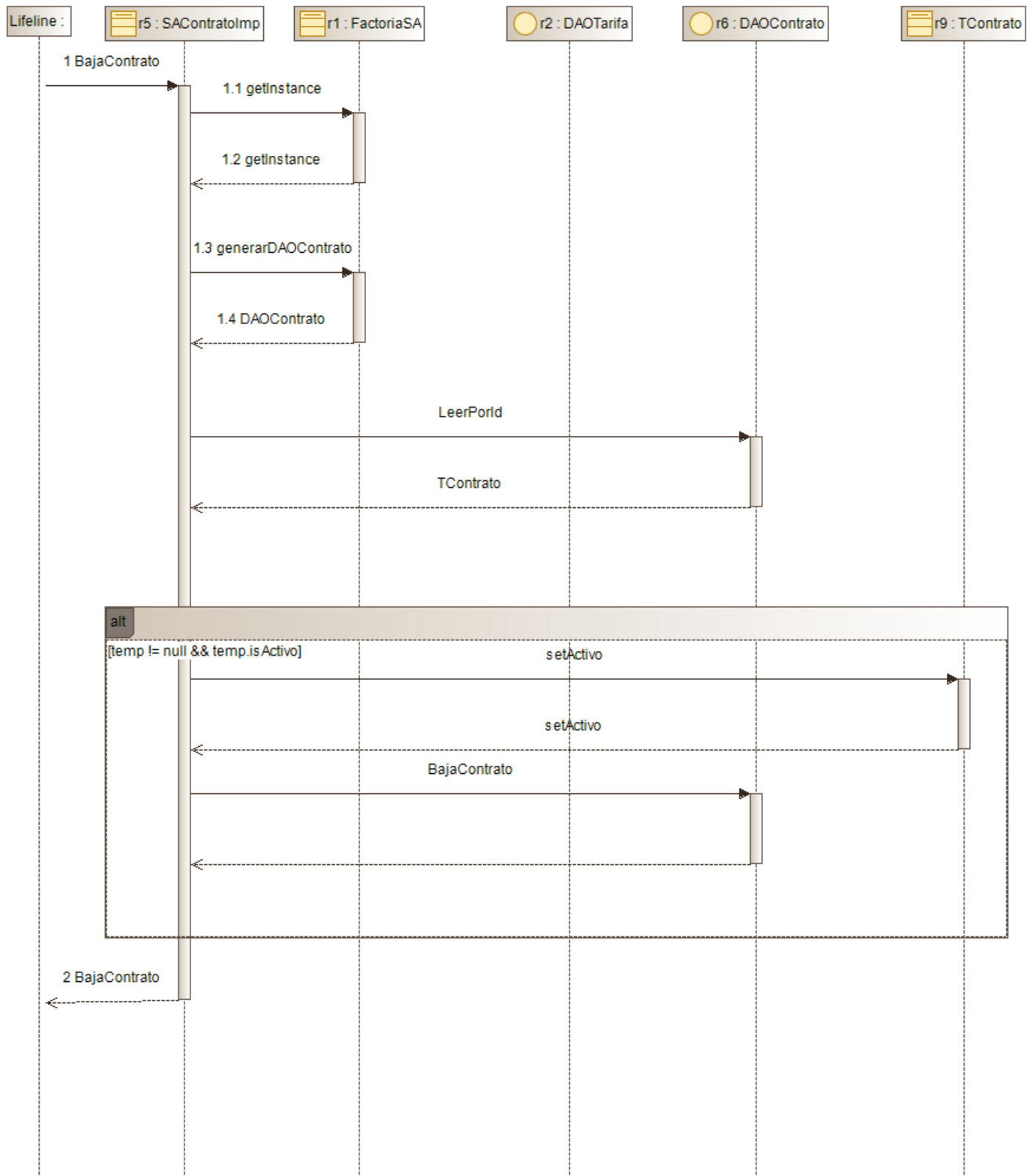


Capa de Negocio

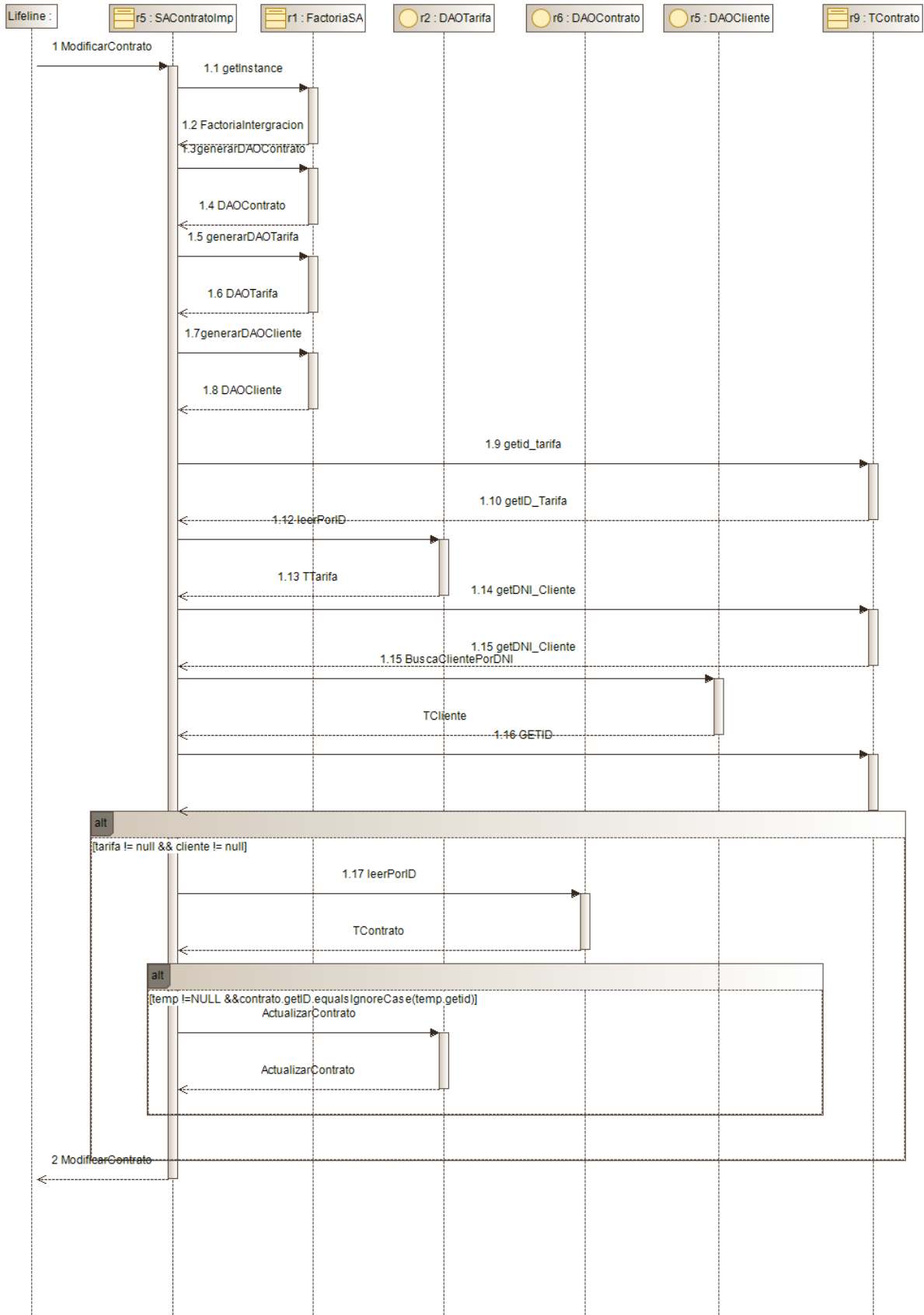
- Alta Contrato SA



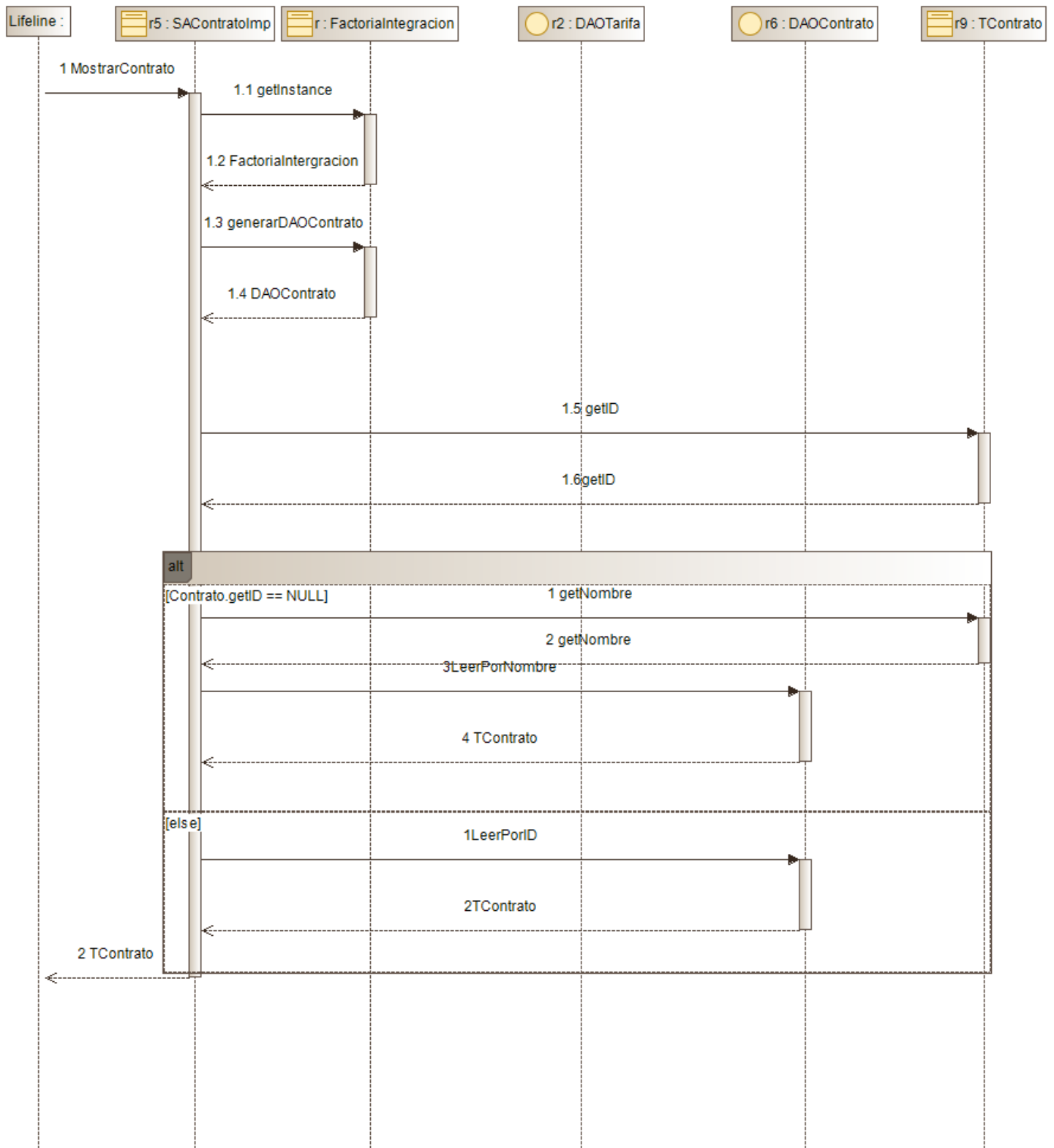
- Baja Contrato SA



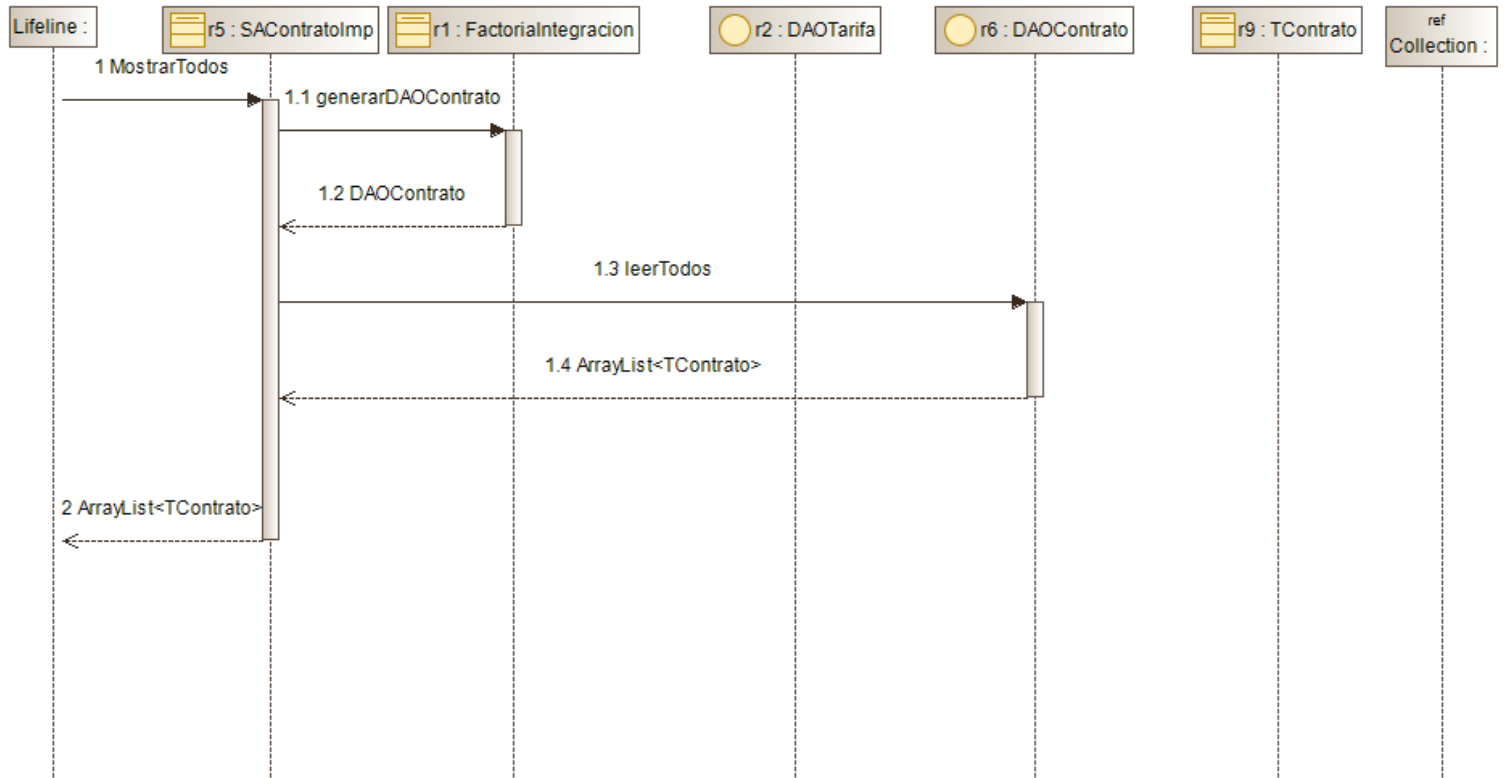
- Modificar Contrato SA



- *Mostrar Contrato SA*

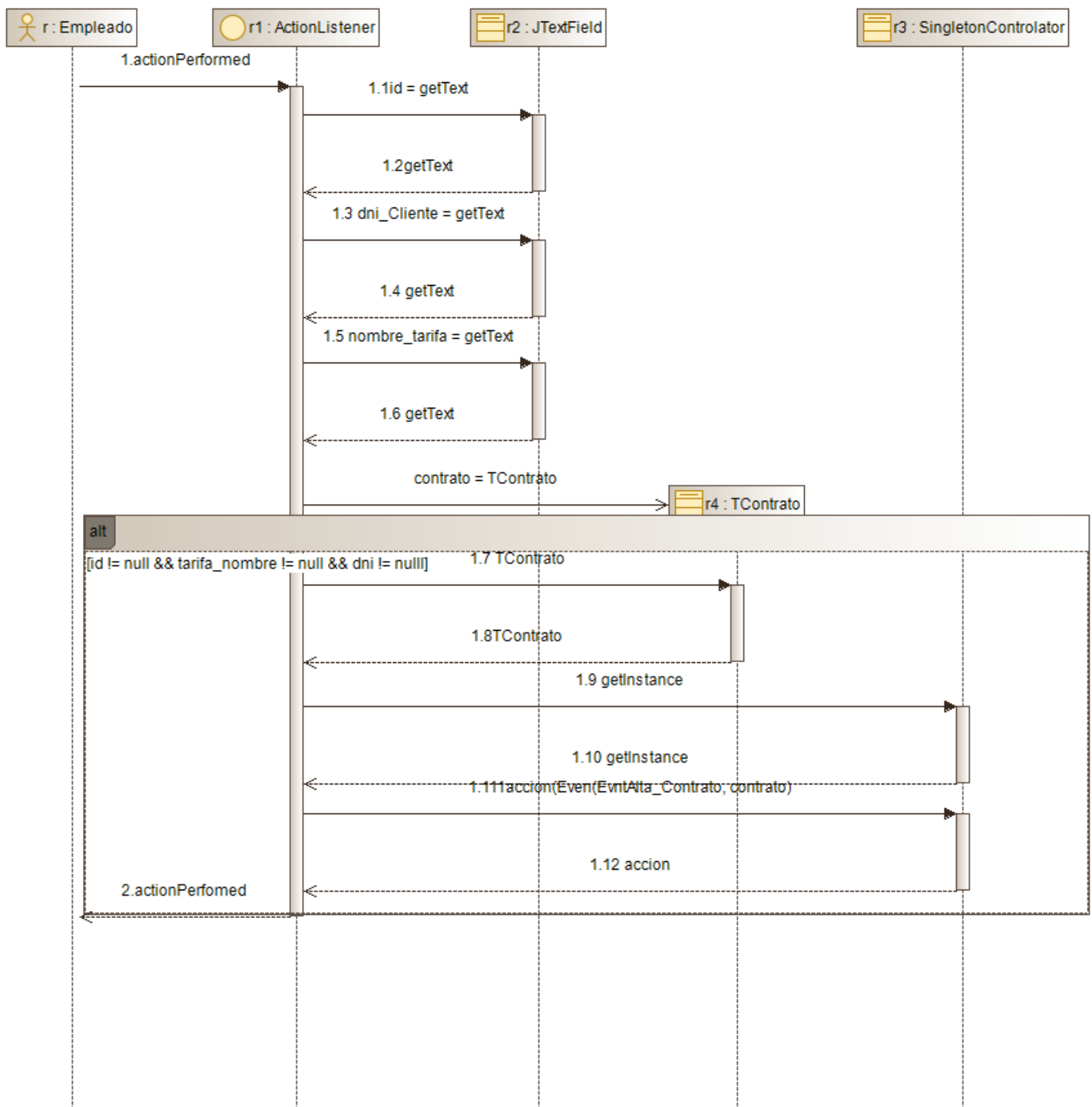


- *Mostrar Todos Contrato SA*

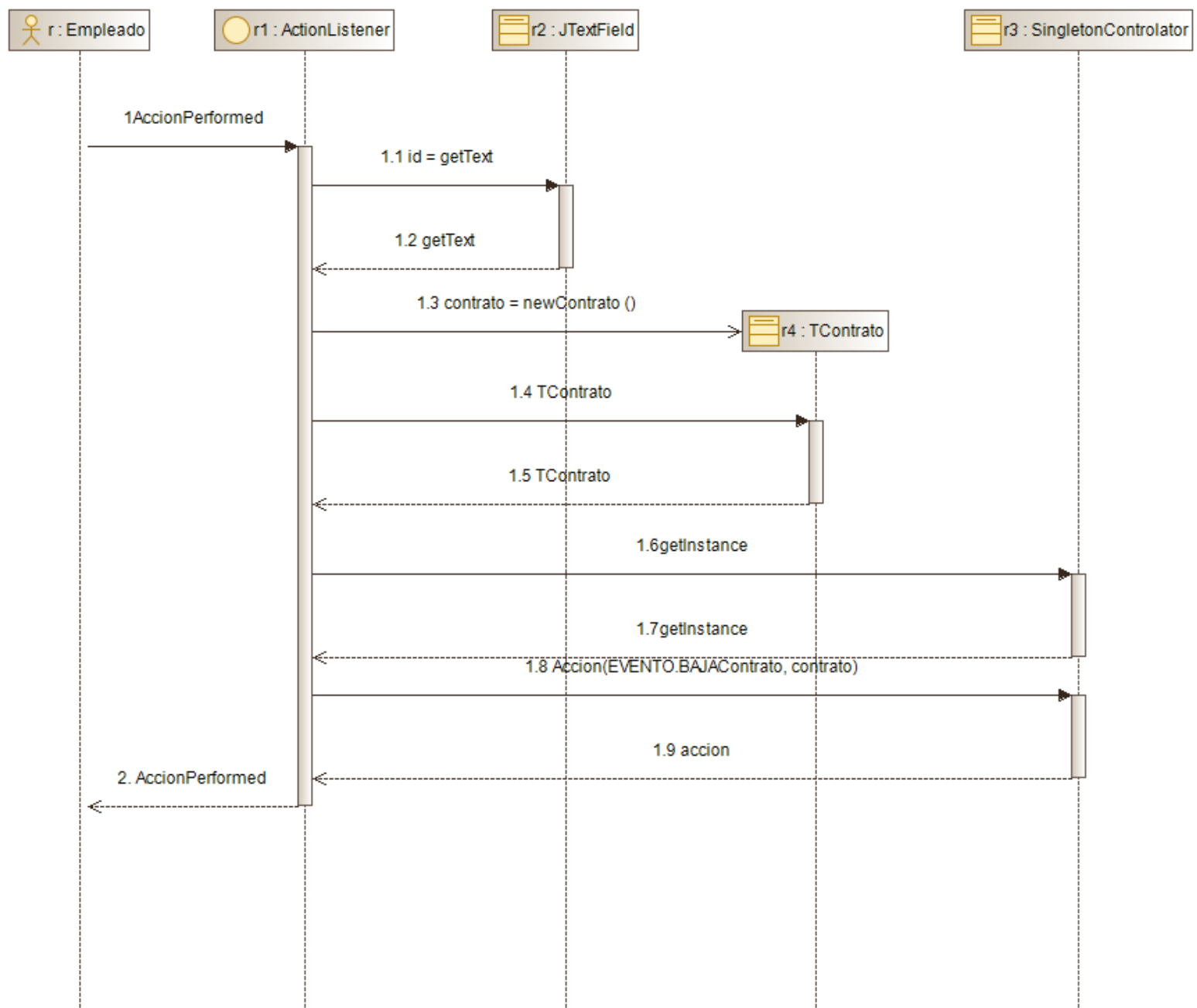


Capa de Presentación

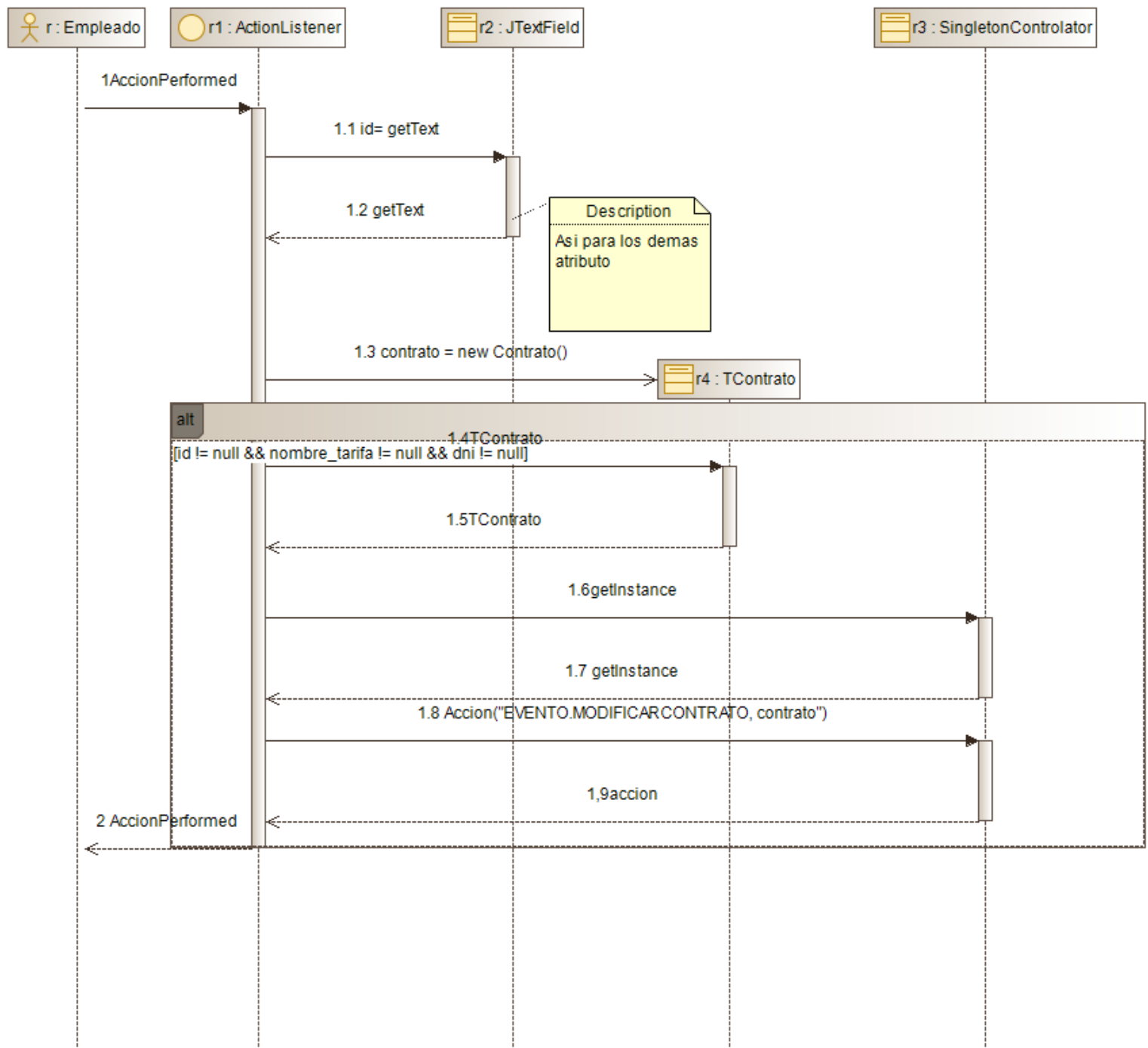
- Alta Contrato Action Listener



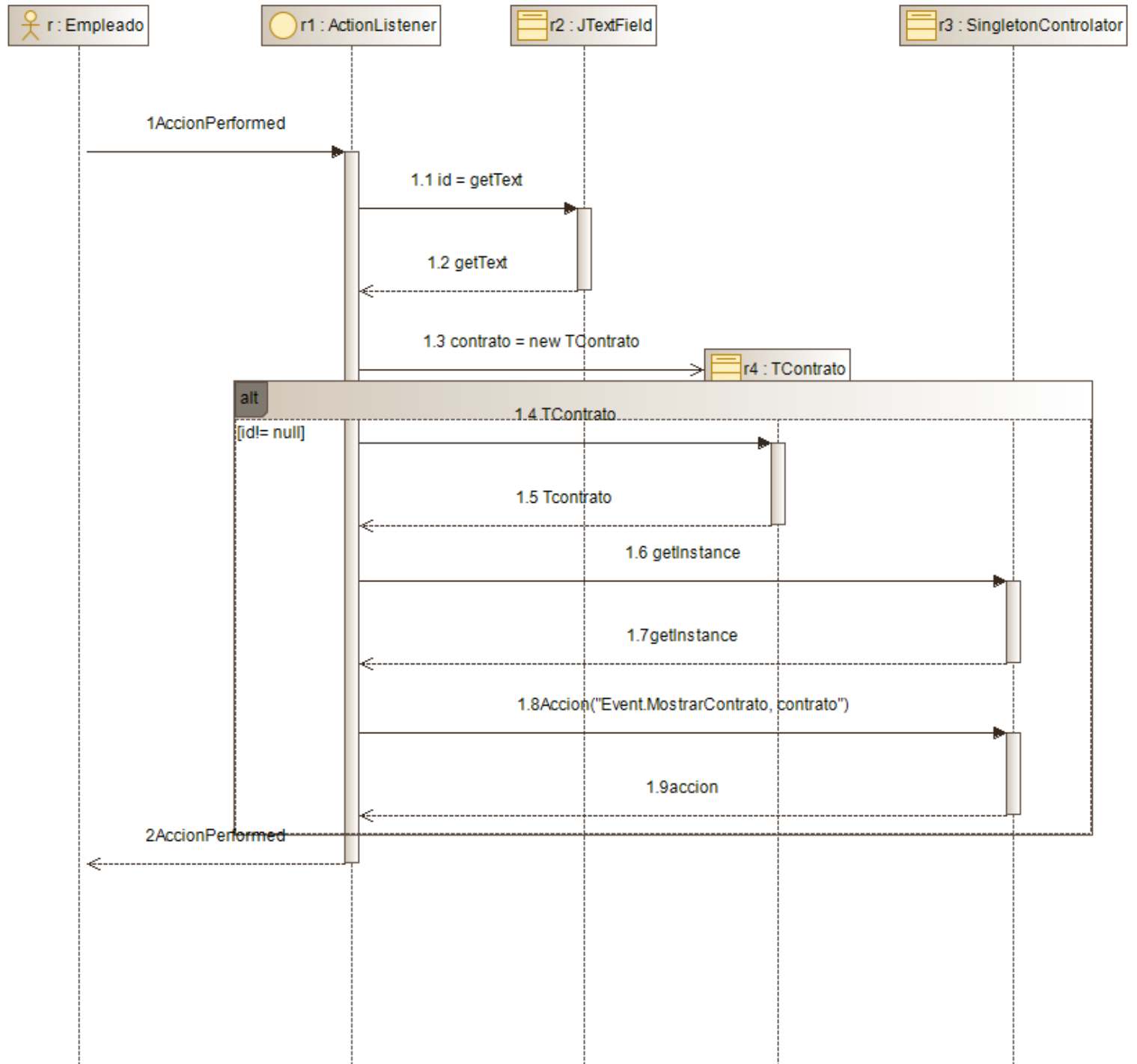
- **Baja Contrato Action Listener**



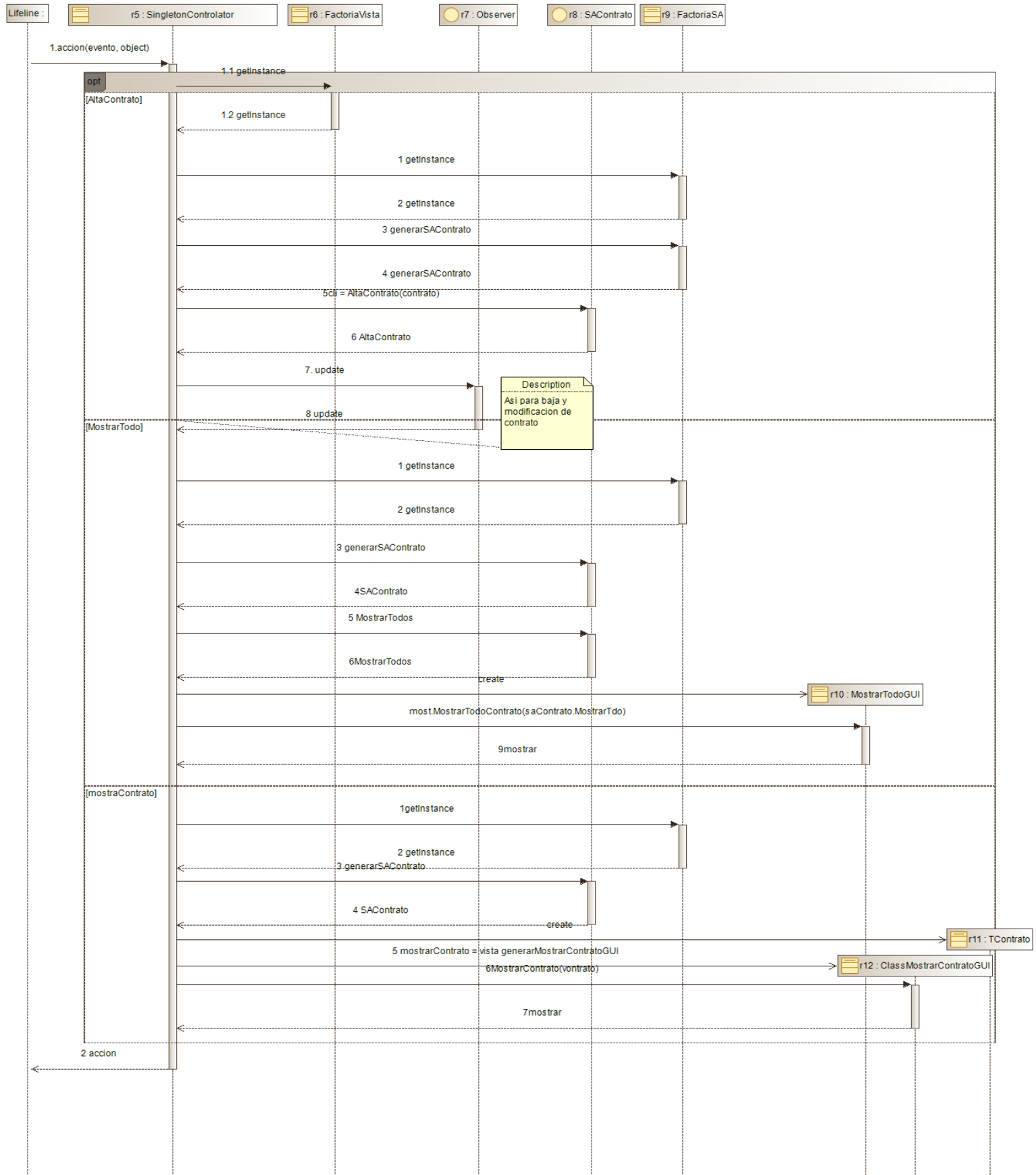
- **Modificar Contrato Action Listener**



- **Mostrar Contrato - Action Listener**

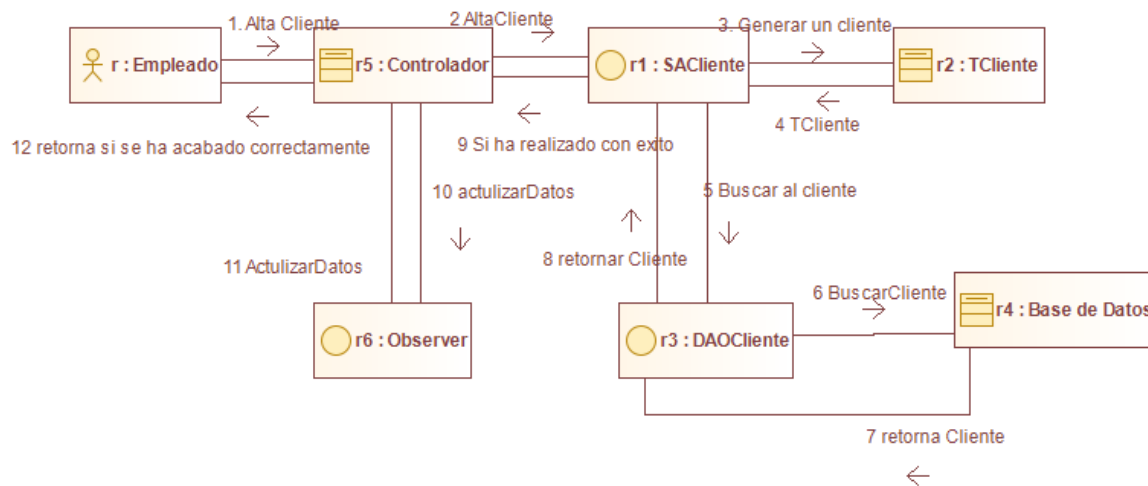


- Controlador Contrato

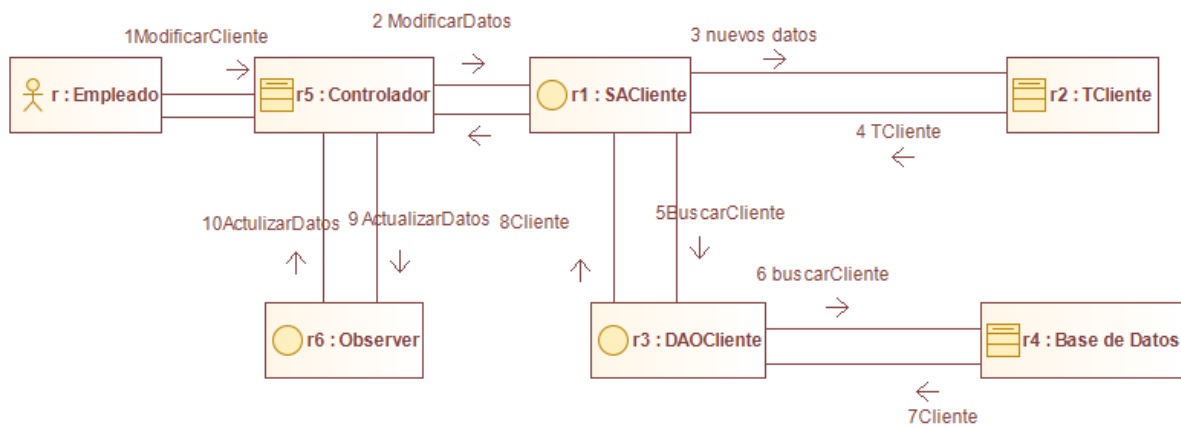


6.3 - 5 diagramas más significativos de Colaboración/Comunicación

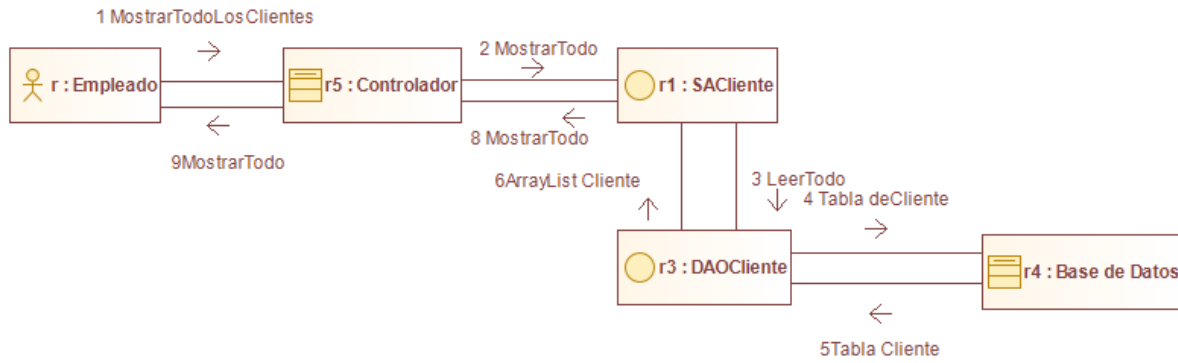
- Alta Cliente



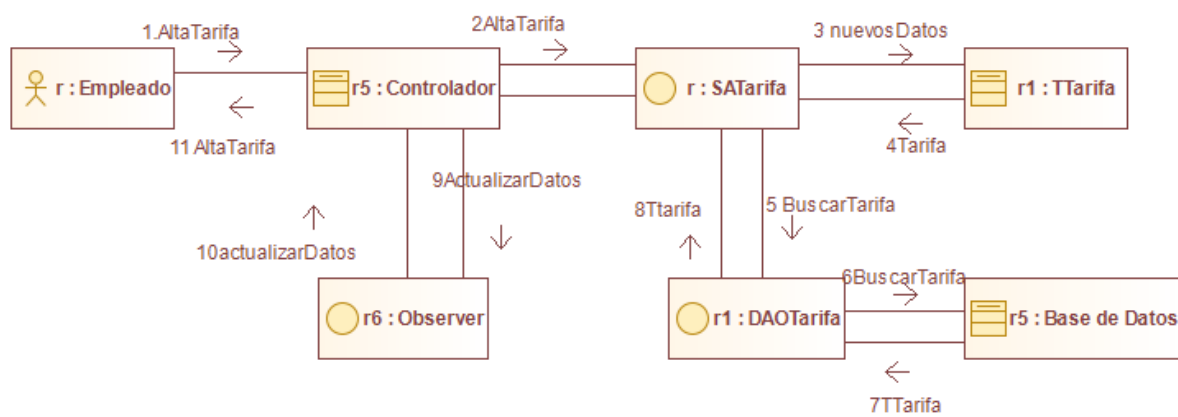
- Modificar Cliente



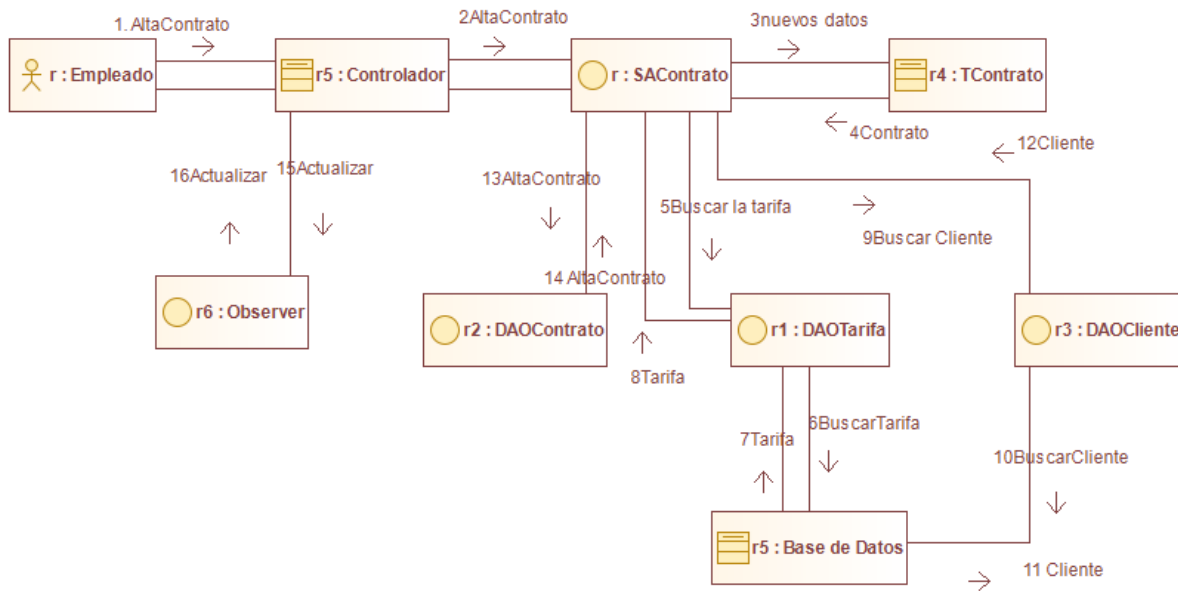
- **Mostrar Todo**



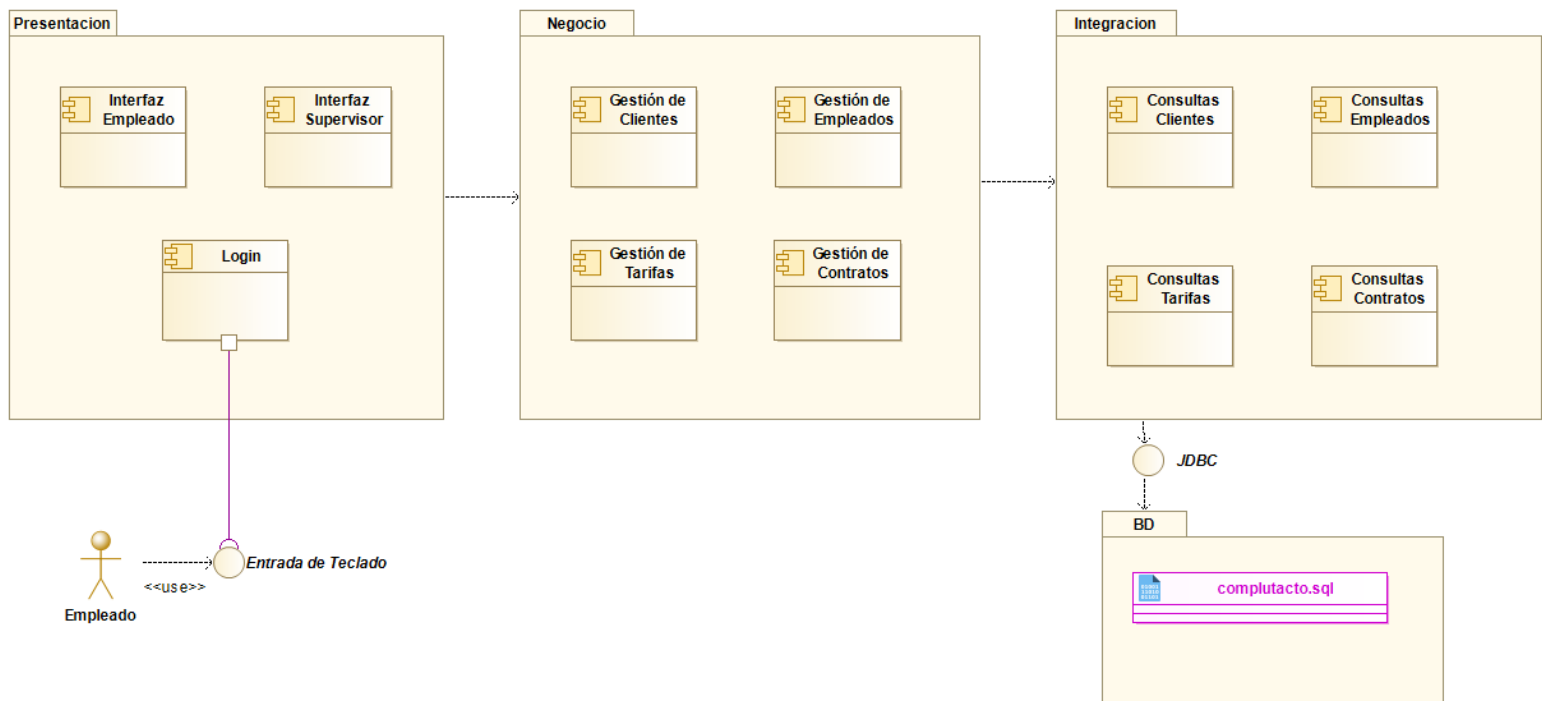
- **Alta Tarifa**



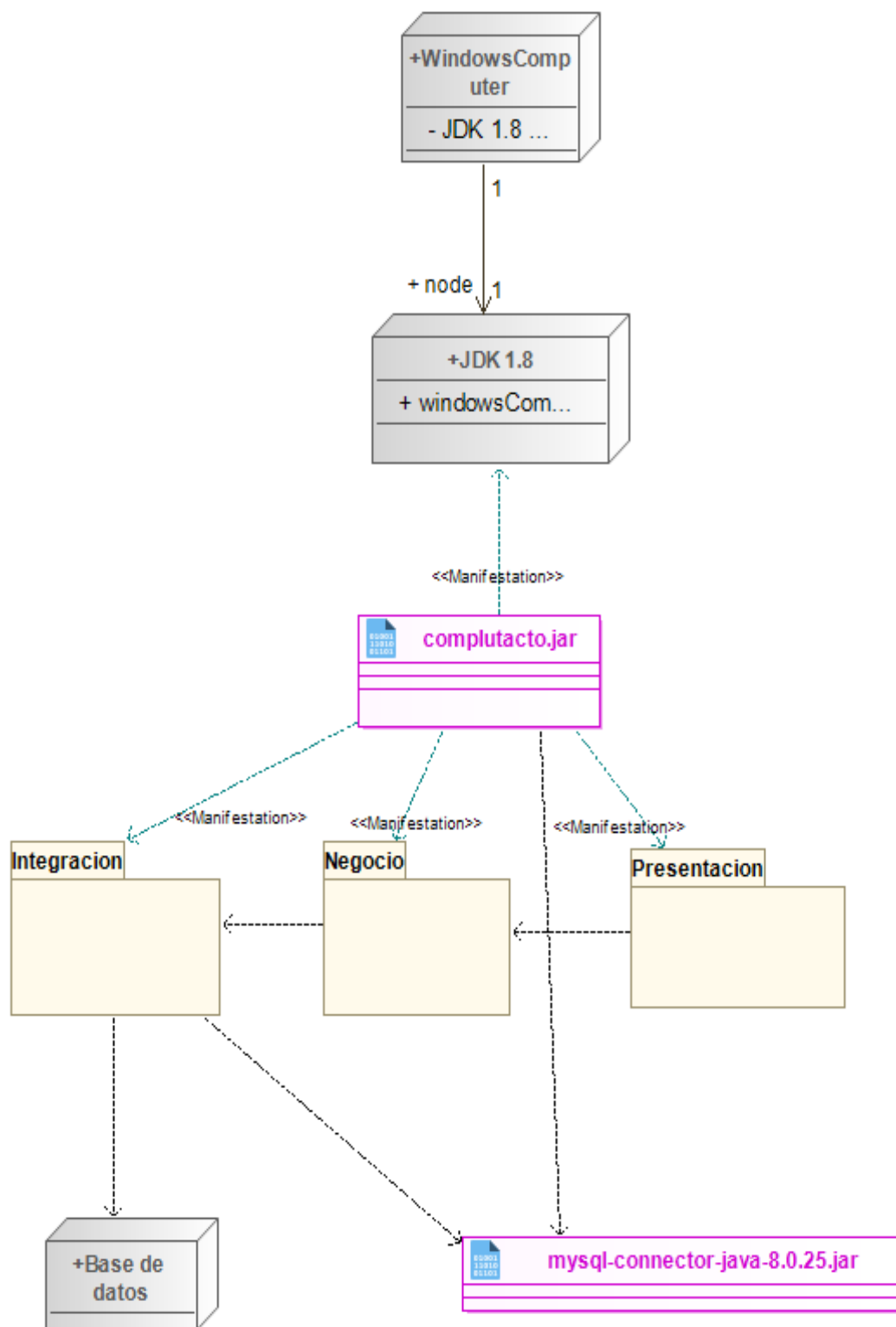
- Alta Contrato



6.4 - De Componentes



6.5 - De Despliegue



7) Código fuente y explicación de la estructura del mismo

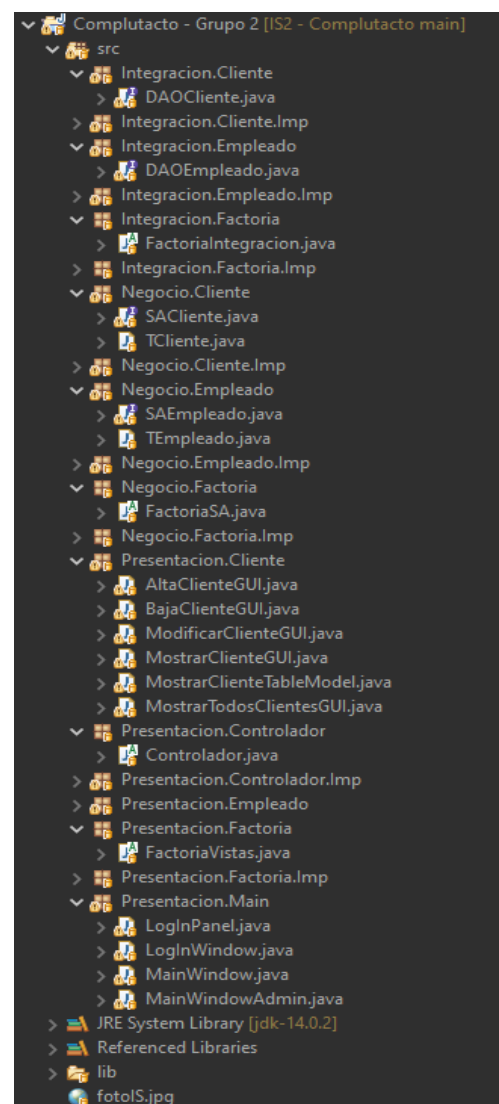
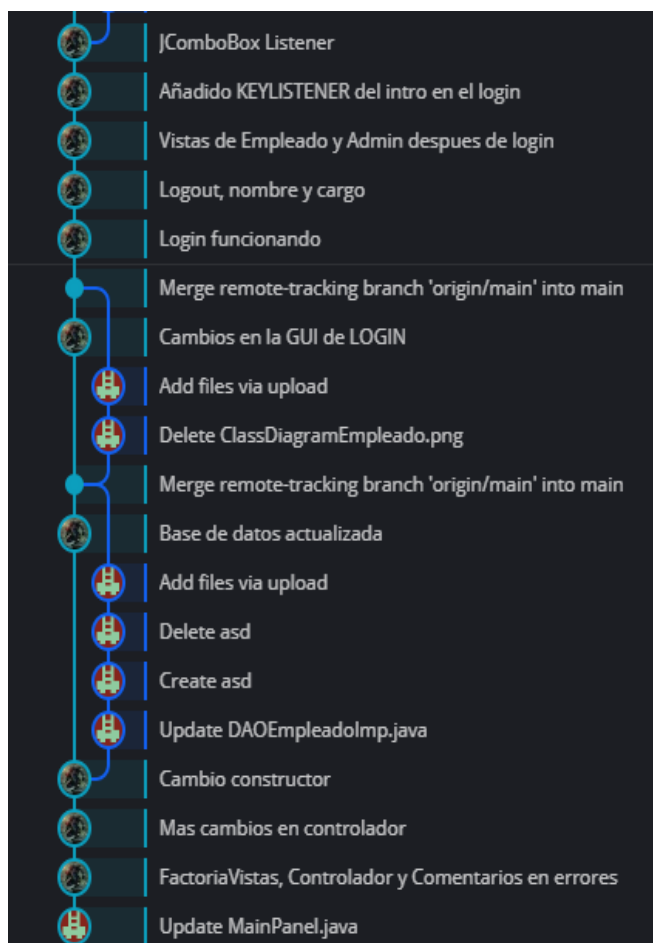
El proyecto está compuesto por cuatro módulos: Módulo de Gestión de Clientes, Módulo de Gestión de Empleados, Módulo de Gestión de Tarifas y Módulo de Gestión de Contratos.

Al tener que desarrollar el 50% del código hemos optado por desarrollar el módulo de gestión de clientes y el módulo de gestión de empleados, que además incluye la posibilidad de hacer login y cambiar de usuario, ya que nos han resultado los más óptimos para su desarrollo y no requerían invertir mucho tiempo en buscar cómo generar gráficas 2D en java que hubiesen sido necesarias para el módulo de contratos o para el de tarifas.

En cuanto a la estructura, hemos dividido el código en 3 paquetes que representan las 3 capas de nuestra arquitectura: Presentación, Negocio e Integración. Dentro de ellos existe un paquete para cada módulo (Cliente o Empleado) y dentro se desarrolla el código de esa parte separando mediante una interfaz las funciones de su implementación.

Por último, cabe destacar el uso de GitHub como sistema de control de versiones, en el que cada integrante del grupo ha ido aportando a la rama principal cada funcionalidad que desarrollaba.

<https://github.com/Ramxnchv/IS2---Complutacto>



8) Ejecución y demo

Para realizar la demo de la aplicación realizaremos los siguientes pasos:

- Levantar el servicio de MySQL en su puerto por defecto (3306). La forma más sencilla de realizar este paso es a través de un gestor que incluya todo como xampp.
- Importar la base de datos 'complutacto.sql' incluida en el directorio raíz del fichero zip de la entrega en nuestro gestor MySQL
- Ejecutar mediante doble click el ejecutable .jar incluido en el zip de la entrega.
- Iniciar sesión en la aplicación con los siguientes datos de empleado administrador (incluidos en fichero .sql importado) o crear unos propios en la tabla empleado de la base de datos:
 - **Usuario: 51139430B**
 - **Contraseña: 1234**
- Por último, elegir en los desplegados el caso de uso que se desea probar y hacer click en Aceptar.

Si deseamos probar también un usuario empleado normal debemos iniciar sesión con estos datos:

- **Usuario: 12345678A**
- **Contraseña: binchen**

9) Extras (Pruebas – JUnit – JMeter– Sonar, etc.)

No se han realizado.

10) Bibliografía y referencias

- API de JAVA
- Transparencias del campus virtual de IS2
- Transparencias de Java Swing de TP2