

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS**



**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y
COMPUTACIÓN**

**“DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE COLECCIONES
BIOLÓGICAS Y COMPARTICIÓN DE LA INFORMACIÓN PARA MOTORES DE BÚSQUEDA.
CASO DE ESTUDIO MUSEO QCAZ DE LA PUCE.”**

AUTORES:

**PABLO ESTEBAN ALMEIDA ENRÍQUEZ
KARLA VANESSA SORIA CARPIO**

DIRECTOR

MSC. DAMIÁN ANIBAL NICOLALDE RODRÍGUEZ

QUITO, 2016

DEDICATORIA

El presente proyecto de disertación está dedicado para Justo, Josefina y Rodrigo, mis ángeles, porque desde el cielo guían siempre mi camino, por darme fuerza y ser siempre una luz que me impulsa a seguir adelante. Porque siempre serán parte de mi vida.

Vanessa Soria C.

AGRADECIMIENTOS

Cada proceso en la vida está lleno de personas, momentos y acciones que, de una u otra manera son parte esencial de cada etapa.

El agradecimiento va para todas aquellas personas que, con su apoyo, aliento, o solamente con su presencia en mi vida, me han ayudado a culminar este proyecto de disertación. En primer lugar, a Dios, por darme la fuerza de voluntad necesaria y por guiar mi camino.

A mis padres quienes para mí son mi fuerza, un ejemplo de vida, gracias por su apoyo incondicional, por todos los consejos y las enseñanzas, por estar junto a mí en cada etapa y por siempre apoyar y respetar mis decisiones. Gracias a ellos todo esto ha sido posible.

A mis hermanas y hermanos, por ser mis confidentes y siempre alentarme a seguir mis metas, por siempre ser mi soporte, y darme aliento. A Katy y Clarita por cada momento compartido, de esos que llenan el alma. A Kary y José Luis, por siempre estar en mi vida.

A José por compartir este camino junto a mí, por su ayuda en todos estos años, por darme el impulso y apoyo para seguir mis metas, por siempre sacar a flote mi mejor versión.

A los profesores que a lo largo de la carrera han compartido sus conocimientos, sin egoísmo alguno, por siempre estar dispuestos a disipar cualquier duda, por sus consejos y enseñanzas, por su dedicación y compromiso en la formación de nuevos profesionales.

A mi director, por la confianza puesta en nosotros para la realización de esta disertación, por la ayuda incondicional y por el tiempo invertido en el desarrollo del proyecto. A mis correctores, por su apoyo y comentarios, que ayudaron a la presentación de este proyecto.

A mi compañero de tesis Pablo, por compartir este proyecto conmigo, por el buen trabajo y por el compromiso. A mis compañeros, a mis amigos de promoción, por cada momento compartido, por cada sonrisa, por hacer que cada día en la universidad sea una experiencia inolvidable.

Mis más sinceros agradecimientos a todos quienes que han recorrido este camino conmigo.

Vanessa Soria C.

DEDICATORIA

A mi madre y a mi hermana...

Pablo Almeida

AGRADECIMIENTOS

A todos quienes me han apoyado durante mi tiempo de estudio en la universidad, familiares, amigos y docentes.

Gracias.

Pablo Almeida

RESUMEN

El presente trabajo de disertación documenta toda la investigación realizada para el desarrollo del sistema para la administración de colecciones biológicas, el cuál se implementó en el Museo QCAZ de la Facultad de Biología de la PUCE.

Proporciona los datos primordiales para conocer sobre colecciones biológicas, tejidos, extracciones, los procesos y el ambiente donde se iba a aplicar el sistema. Además proporciona una breve descripción de los conceptos técnicos sobre los que se fundamentó la aplicación.

Siguiendo la metodología RUP, se documentó cada etapa para el desarrollo de la aplicación, desde el análisis y levantamiento de requerimientos, siguiendo por el diseño (diagramas), implementación, pruebas y puesta en marcha del sistema.

Indice

DEDICATORIA.....	2
AGRADECIMIENTOS	3
DEDICATORIA.....	4
AGRADECIMIENTOS	5
RESUMEN	6
1. CAPÍTULO 1: FUNDAMENTOS TEÓRICOS	14
1.1. Colecciones biológicas	14
1.1.1. Definición.....	14
1.1.2. Tejidos.....	14
1.1.3. Extracciones	15
1.1.4. PCR.....	15
1.2. Ingeniería de software	16
1.3. Bases de datos.....	18
1.3.1. Introducción a las bases de datos.....	18
1.3.2. Sistemas Gestores de Bases de Datos.....	19
1.3.3. SQLSERVER.....	21
1.3.4. TRANSACT-SQL.....	21
1.4. Lenguajes de programación.....	22
1.4.1. Definición.....	22
1.4.2. C#.....	23
1.4.2.1. Características	23
1.4.3. .NET Framework.....	23
1.5. Sistemas de distribuidos	27
1.5.1. Definición.....	27
1.5.1.1. N Capas.....	28

1.5.2.	XML	29
1.5.3.	HTML.....	29
1.5.4.	Servicios web.....	29
1.5.4.1.	SOAP	31
1.5.5.	ASP.NET	31
1.6.	Metodología de Desarrollo del Proyecto	32
1.6.1.	Método RUP	33
2.	CAPÍTULO 2: ANÁLISIS	37
2.1.	Situación Actual	37
2.2.	Levantamiento de Requerimientos.....	38
2.3.	Sistema Propuesto.....	38
2.3.1.	Casos de uso.....	39
2.3.2.	Diagramas de Secuencia	59
2.3.3.	Diagrama de Actividades	76
2.3.4.	Diagrama de Estereotipos.....	78
3.	CAPÍTULO 3: DISEÑO	82
3.1.	Entorno del Software	82
3.2.	Diagrama de Clases	83
3.3.	Esquema de Datos	83
3.3.1.	Diagrama Conceptual	83
3.3.2.	Diccionario de Datos.....	83
3.4.	Diagrama de Colaboración	84
3.4.1.	F1. Administrar Colección	84
3.4.2.	F3. Administrar Préstamos.....	88
3.4.3.	F4. Administrar Tejido.....	92
3.4.4.	F5. Administrar Extracción	96

3.4.5.	F6. Administrar Amplificación.....	100
3.4.6.	F7. Administrar Cuaderno de Campo.....	104
3.5.	Diseño interno del Software	108
3.6.	Diseño de Interfaces	108
3.7.	Diseño de servicios a utilizarse	112
4.	CAPÍTULO 4: IMPLEMENTACIÓN	113
4.1.	Estándares de Codificación	113
	ProyectoBiologiaQCAZ	113
	QCAZ-Internal.....	115
4.2.	Diagrama de Despliegue.....	116
4.3.	Diagrama de Componentes	117
4.4.	Diagrama de Paquetes	118
4.5.	Pruebas del Sistema.....	119
4.5.1.	Caso de Prueba: F1. Administrar Colección	119
4.5.2.	Caso de Prueba: F3. Administrar Préstamos	120
4.5.3.	Caso de Prueba: F4. Administrar Tejido	120
4.5.4.	Caso de Prueba: F5. Administrar Extracción	121
4.5.5.	Caso de Prueba: F6. Administrar PCR.....	121
4.5.6.	Caso de Prueba: F7. Administrar Cuaderno de Campo	122
4.6.	Instalación y configuración.....	122
	CONCLUSIONES	123
	RECOMENDACIONES	124
	BIBLIOGRAFÍA.....	125

Índice de Ilustraciones

Ilustración 1-1	16
Ilustración 1-2 SGBD	20
Ilustración 1-3 Bases de Datos menos vulnerables.....	21
Ilustración 1-4 .NET Framework.....	24
Ilustración 1-5 Funcionamiento del Entity Framework	25
Ilustración 1-6 ADO.NET.....	26
Ilustración 1-7 N Capas.....	28
Ilustración 1-8 Servicios Web.....	30
Ilustración 1-9 SOAP.....	31
Ilustración 1-10 Fases del método RUP	33
Ilustración 1-11 Roles en RUP	34
Ilustración 1-12 Método SCRUM Características	35
Ilustración 1-13 Iteración.....	36
Ilustración 3-1 Diseño Interno del Software.....	108
Ilustración 3-2 Master Page - Diseño	109
Ilustración 3-3 Pantalla Login - Diseño.....	109
Ilustración 3-4 Pantalla Principal - Diseño	110
Ilustración 3-5 Pantalla Consulta General - Diseño	110
Ilustración 3-6 Pantalla Insertar - Diseño.....	111
Ilustración 3-7 Pantalla Edición - Diseño.....	111

Indice de Tablas

Tabla 2-1 F0. Administrar Sesión - Flujos	41
Tabla 2-2 F1. Administrar Colección - Flujos.....	43
Tabla 2-3 F3. Administrar Préstamos - Flujos	46
Tabla 2-4 F4. Administrar Tejidos - Flujos.....	49
Tabla 2-5 F5. Administrar Extracción - Flujos.....	52
Tabla 2-6 F6. Administrar Amplificación – Flujos.....	55
Tabla 2-7 F7. Administrar Cuaderno de Campo	58
Tabla 3-1 Entorno del Software.....	82

Indice de Diagramas

Diagrama 2-1 General - Casos de Uso	39
Diagrama 2-2 F0. Administrar Sesión - Casos de Uso	40
Diagrama 2-3 F1. Administrar Colección - Casos de Uso	42
Diagrama 2-4 F3. Administrar Préstamos – Casos de Uso	45
Diagrama 2-5 F4. Administrar Tejidos - Casos de Uso.....	48
Diagrama 2-6 F5. Administrar Extracciones - Casos de Uso.....	51
Diagrama 2-7 F6. Administrar PCR.....	54
Diagrama 2-8 F7. Administrar Cuaderno de Campo - Casos de Uso	57
Diagrama 2-9 F1.1. Ingresar Colección Zoología - Secuencia	59
Diagrama 2-10 F1.2. Modificar Colección Zoología - Secuencia	60
Diagrama 2-11 F1.3. Eliminar Colección Zoología - Secuencia.....	61
Diagrama 2-12 F1.4. Consultar Colección Zoología – Secuencia.....	61
Diagrama 2-13 F3.1. Ingresar Préstamo - Secuencia.....	62
Diagrama 2-14 F3.2. Modificar Préstamo - Secuencia	63

Diagrama 2-15 F3.3. Consultar Préstamo - Secuencia	64
Diagrama 2-16 F3.4. Eliminar Préstamo – Secuencia	64
Diagrama 2-17 F4.1. Ingresar Tejido – Secuencia	65
Diagrama 2-18 F4.2. Modificar Tejido - Secuencia.....	66
Diagrama 2-19 F4.3. Eliminar Tejido - Secuencia	67
Diagrama 2-20 F4.4. Consular Tejido - Secuencia	67
Diagrama 2-21 F5.1. Ingresar Extracción - Secuencia	68
Diagrama 2-22 F5.2. Modificar Extracción - Secuencia.....	69
Diagrama 2-23 F5.3. Eliminar Extracción - Secuencia	70
Diagrama 2-24 F5.4. Consultar Extracción - Secuencia	70
Diagrama 2-25 F6.1. Ingresar Amplificación - Secuencia.....	71
Diagrama 2-26 F6.2. Modificar Amplificación - Secuencia	72
Diagrama 2-27 F6.3. Eliminar Amplificación - Secuencia	73
Diagrama 2-28 F6.4. Consultar Amplificación - Secuencia.....	73
Diagrama 2-29 F7.1. Ingresar Cuaderno de Campo - Secuencia.....	74
Diagrama 2-30 F7.2. Modificar Cuaderno de Campo - Secuencia	74
Diagrama 2-31 F7.3. Eliminar Cuaderno de Campo - Secuencia	75
Diagrama 2-32 F7.4. Consultar Cuaderno de Campo - Secuencia.....	75
Diagrama 2-33 Actividades	77
Diagrama 2-34 F1. Administrar Colección - Estereotipos	78
Diagrama 2-35 F3. Administrar Préstamos - Estereotipos.....	79
Diagrama 2-36 F4. Administrar Tejido - Estereotipos	79
Diagrama 2-37 F5. Administrar Extracción - Estereotipos	80
Diagrama 2-38 F6. Administrar Amplificación - Estereotipos.....	80
Diagrama 2-39 F7. Administrar Cuaderno de Campo - Estereotipo.....	81
Diagrama 3-1 F1.1. Ingresar Colección Zoología - Colaboración	84

Diagrama 3-2 F1.2. Modificar Colección Zoología - Colaboración.....	85
Diagrama 3-3 F1.3. Eliminar Colección Zoología - Colaboración	86
Diagrama 3-4 F1.4. Consultar Colección Zoología - Colaboración.....	87
Diagrama 3-6 F3.4. Consultar Préstamo - Colaboración	90
Diagrama 3-8 F4.1. Ingresar Tejido - Colaboración.....	92
Diagrama 3-9 F4.2. Modificar Tejido - Colaboración	93
Diagrama 3-10 F5.1. Ingresar Extracción - Colaboración.....	96
Diagrama 3-11 F5.2. Modificar Extracción - Colaboración	97
Diagrama 3-12 F5.4. Consultar Extracción - Colaboración.....	99
Diagrama 3-13 F6.1. Ingresar Amplificación - Colaboración.....	100
Diagrama 3-14 F6.2. Modificar Amplificación – Colaboración	101
Diagrama 3-15 F6.3. Eliminar Amplificación - Colaboración.....	102
Diagrama 3-16 F7.1. Ingresar Cuaderno de Campo	104
Diagrama 3-17 F7.3. Eliminar Cuaderno de Campo.....	106
Diagrama 3-18 F7.4. Consultar Cuaderno de Campo	107
Diagrama 4-1 Despliegue	116
Diagrama 4-2 Diagrama de Componentes	117
Diagrama 4-3 Diagrama de Paquetes	118

1. CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

1.1. Colecciones biológicas

1.1.1. Definición

Para entender el objetivo y la importancia de las Colecciones Biológicas, podemos partir del hecho que desde el inicio de la humanidad el hombre ha tenido como característica natural el recolectar objetos, ya sea por supervivencia, para estudiar sus componentes, o por el hecho de dejar un registro a futuras generaciones; estas recolecciones, de un grupo de varios objetos de una misma clase es lo que se denomina Colección. (Simmons & Muñoz-Saba, 2005)

En el caso de las Colecciones Biológicas, según datos históricos, en el siglo XVII se inició la recolección y preservación de especímenes animales con fines científicos.

En el artículo *La Importancia Actual de las Colecciones de Tejidos* se menciona que "Una colección biológica es un archivo histórico de utilidad múltiple: es la referencia más directa para la identificación correcta de ejemplares; es el cimiento de la taxonomía y la nomenclatura; es una fuente de información para muchos tipos de proyectos aplicados y es un instrumento para la educación a varios niveles." (García Deras, López de Aquino, Honey Escandón, Cortés Rodríguez, & Hernández Baños, 2001)

Las colecciones biológicas están bajo la responsabilidad de un Museo, el cual debe cumplir con propósitos específicos, que son: Generar, perpetuar, organizar y difundir información. (Simmons & Muñoz-Saba, 2005)

1.1.2. Tejidos

Según M.Abercrombie, C.J.Hickman y M.L. Johnson en su *Diccionario de Biología* un tejido se define como "región formada principalmente por células de la misma clase (que realizan la misma función) asociadas en gran número, unidas por las membranas celulares (vegetales) o por material intercelular (animales); por ejemplo, las células del córtex de las plantas con sus membranas de celulosa; o las numerosas fibras musculares estriadas, unidas por el colágeno, junto con los vasos sanguíneos y linfáticos que las acompañan, los nervios y las células del tejido conjuntivo que en conjunto forman el tejido muscular." (Abercrombie, Hickman, & Johnson, 1970)

El tejido es extraído a partir del espécimen y permite un análisis más profundo de la especie, se puede trabajar con dichos tejidos o mantenerlos guardados para referencias futuras. El tejido también permite realizar el proceso de extracción y posteriormente amplificación que mencionamos a continuación.

1.1.3. Extracciones

La real academia de la lengua española define al proceso de extracción como "obtener uno de los componentes de un cuerpo por la acción de disolventes u otros medios." (RAE, s.f.)

La extracción se define más a profundidad como: "un procedimiento de separación de una sustancia que puede disolverse en dos disolventes no miscibles entre sí, con distinto grado de solubilidad y que están en contacto a través de una interface. La relación de las concentraciones de dicha sustancia en cada uno de los disolventes, a una temperatura determinada, es constante." (LEXICOON, 2016)

Dentro del museo QCAZ la extracción se realiza a partir de un tejido, se pueden utilizar diferentes métodos según los requerimientos y posteriormente se puede almacenar en la ultra congeladora. Es importante mencionar que todos los datos sobre la extracción son debidamente almacenados.

1.1.4. PCR

Polymerase Chain Reaction (PCR) es una técnica de amplificación desarrollada en 1986 por Kary Mullis, "su objetivo es obtener un gran número de copias de un fragmento de AND particular, partiendo de un mínimo; en teoría basta partir de una única copia de ese fragmento original, o molde." (Wikipedia, s.f.)

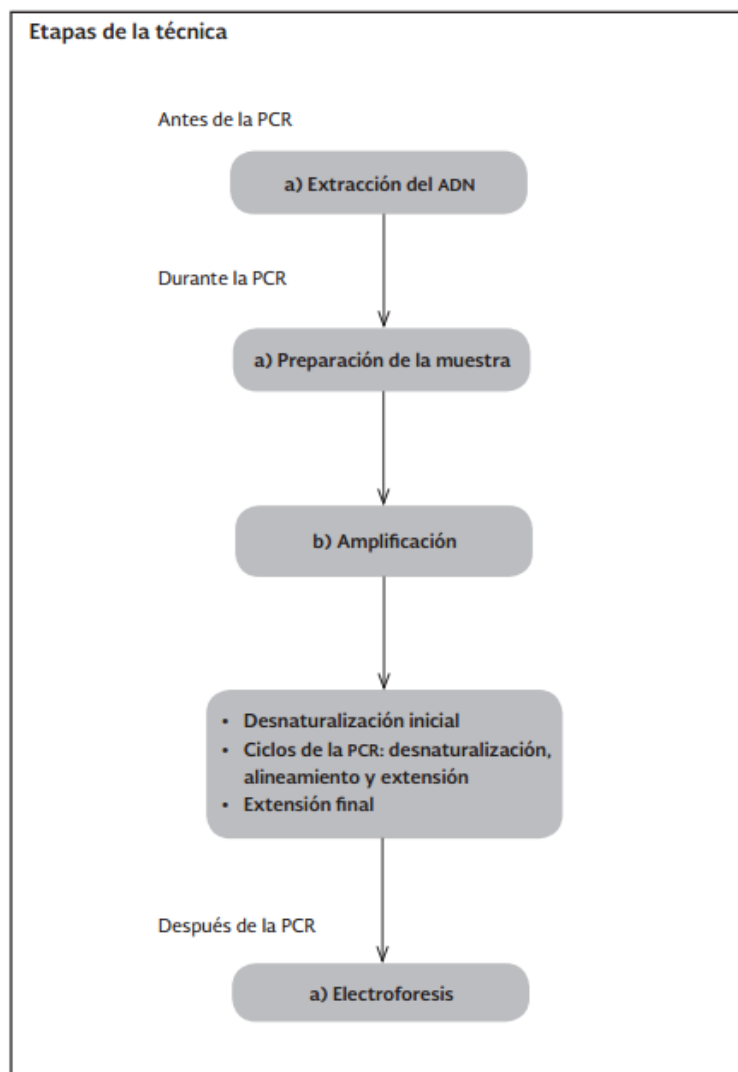
El ciclo para el proceso de PCR consta de 3 pasos, los cuales se realizan una vez que la muestra haya sido preparada (Garrido Ramos, s.f.):

- **Desnaturalización.** Se eleva la temperatura del tubo de reacción hasta 94°C, por un tiempo entre ½ minuto y 2 minutos.

- **Alineamiento.** Se desciende la temperatura hasta que oscile entre 40°C y 60°C. La duración de este paso puede oscilar entre ½ minuto y 2 minutos.
- **Extensión.** Se vuelve a aumentar la temperatura hasta los 72°C y se deja actuar a la Taq polimerasa durante 1 o 2 minutos.

Garrillo menciona que la repetición de este ciclo permite obtener, como resultado de un experimento de amplificación, millones de copias del fragmento de interés.

Ilustración 1-1



Nota: Adaptado de: “PCR: reacción en cadena de la polimerasa”, por Serrano, Alejandra; Flores, Lluvia; Aportela, Jaime; Sierra, Edgar. 2007. <http://www2.inecc.gob.mx/publicaciones/libros/710/pcr.pdf>

1.2. Ingeniería de software

Pontificia Universidad Católica del Ecuador, Autores: Vanessa Soria y Pablo Almeida

Un software consiste en un "producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos." (Pressman, 2010)

Podemos encontrar software en casi todos los aspectos de nuestra vida; se encuentra en el comercio, distribución de materiales, trabajo, estudio. Responder una llamada telefónica en un teléfono inteligente o realizar una compra por internet es un proceso gobernado por software, y de ahí viene su importancia.

El término ingeniería de software fue utilizado por primera vez en el comité de Ciencia de la OTAN celebrada en Garmisch (Alemania), en octubre de 1968, por Fritz Bauer. Los motivos por los cuáles surge la ingeniería de software fueron muy variados y comprendieron desde retrasos en los entregables o baja calidad en el producto hasta una disminución en la productividad de los equipos, era necesario un cambio en el enfoque aplicado.

Rafael Menéndez de la Universidad de Murcia define que "la Ingeniería del software es la rama de la ingeniería que crea y mantiene las aplicaciones de software usando tecnologías y prácticas de las ciencias de la computación, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos." (Menéndez, 2014)

"La ingeniería de software está formada por un proceso, un conjunto de métodos (prácticas) y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad." (Pressman, 2010)

Por lo tanto, la ingeniería de software encierra procesos de modelado, administración de calidad y mecanismos de administración de un proyecto. La comprensión de los requerimientos por parte del usuario es el primer paso. Le siguen diferentes tipos de diseño (arquitectura, interfaz, componentes) antes de entrar al desarrollo en sí. Para asegurar la calidad se utilizan técnicas de revisión y pruebas, en las cuáles es importante la participación del cliente y/o usuario final.

Cada paso debe ser debidamente documentado, no solamente como una referencia, sino también para realizar mejores proyecciones sobre estimaciones en el desarrollo de futuros proyectos.

1.3. Bases de datos

1.3.1. Introducción a las bases de datos

El almacenamiento de datos nace de la demanda, cada vez mayor, al acceso a datos e información, esto se ha visto traducido a gran cantidad de papeles, carpetas y archivadores, que antes del aparecimiento de la informática eran utilizados para el manejo y almacenamiento de datos.

El término fue utilizado en 1963, en un simposio celebrado en California, USA. En un artículo Web, *Damián Pérez* menciona que "Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada." (Pérez Valdés, 2007)

Ya que con el uso de la informática se ha buscado mejorar los procesos y en sí, la vida de las personas, se han desarrollado sistemas gestores de bases de datos, los cuales permiten un mejor manejo de la información, mayor acceso a esta, y por supuesto, mayor seguridad y confiabilidad.

Partiendo del punto de vista de la informática, se puede decir que "la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos." (Pérez Valdés, 2007)

Por otro lado, en el libro *Absolute Beginner's Guide to Databases* se menciona que las mejores definiciones, son las más simple, y basándose en esta primicia define una base de datos como una colección, "una base de datos es una colección de datos de elementos relacionados." (Petersen, 2012)

Una base de datos relacional consta de tres elementos básicos (Petersen, 2012):

- **Tablas (*Entidades*):** Representa los elementos de la información que se desea almacenar. Ej. Tabla personas
- **Columnas (*Campos*):** Representa los atributos de cada tabla. Ej. Nombre, apellido, dirección, etc.
- **Filas (*Registros*):** Representa los datos almacenados.

Además de estos elementos básicos, una base de datos relacional, tal como su nombre lo indica, se basa en relaciones, esto es el vínculo que tienen las tablas entre ellas.

El almacenamiento de información en una base de datos, proporciona mayores ventajas, en términos de competitividad y seguridad, entre ellos (Pérez Valdés, 2007):

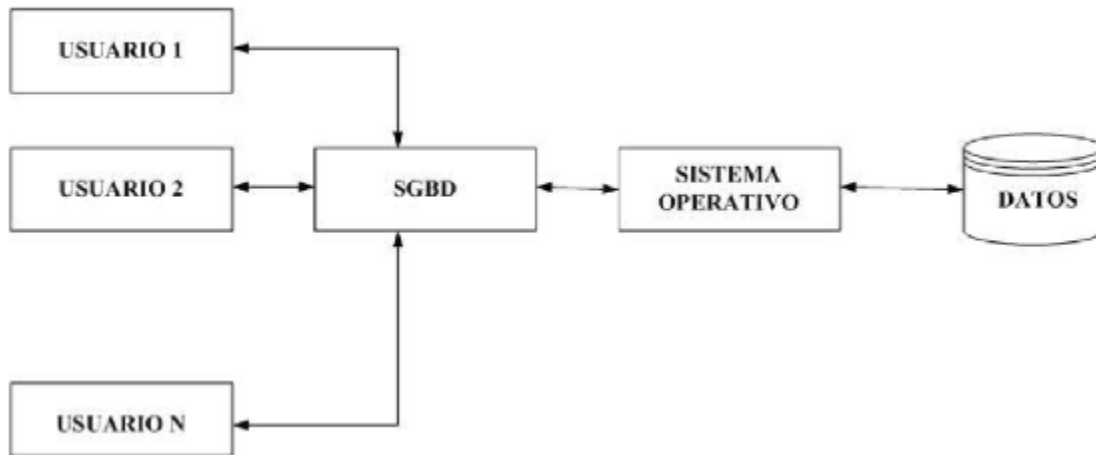
- Control sobre la redundancia de datos
- Consistencia de datos
- Mantenimiento de estándares
- Mejora en la seguridad
- Mejora en la accesibilidad a los datos
- Mejora en la productividad
- Mejora en el mantenimiento
- Mejora en los servicios de copias de seguridad

1.3.2. Sistemas Gestores de Bases de Datos

Con el crecimiento de la informática y de los servicios que esta ofrece, se han desarrollado diversos sistemas que permiten una mejor manipulación de datos, esto es, interfaces más amigables, herramientas para reportes, entre otros.

En el libro *Diseño y programación de bases de datos* se define un SGBD "Un sistema de gestión de bases de datos es un software o conjunto de programas que permite crear y mantener una base de datos.". De forma general un software gestor de base de datos proporciona una interfaz para facilitar la interacción entre el usuario y la base de datos con el objetivo de simplificar los procesos de definir, construir y manipular los datos. (Cobo, s.f.)

Ilustración 1-2 SGBD



Nota: Adaptado de: “Introducción a Las Bases de Datos Relacionales”, por Nevado Cabello María Victoria, Madrid: Visión Libros

Un sistema gestor de base de datos se compone de (Pérez Valdés, 2007):

- Lenguaje de definición de datos (permiten definir estructuras para almacenar los datos)
- Lenguaje de manipulación de datos (permite ejecutar tareas de consulta y modificación de datos)
- Lenguaje de consulta (pueden ser de datos/información, o de consulta de la base)

Las ventajas de un SGBD son básicamente las nombradas durante su descripción junto con las ventajas de una base de datos en sí. Entre las desventajas tenemos:

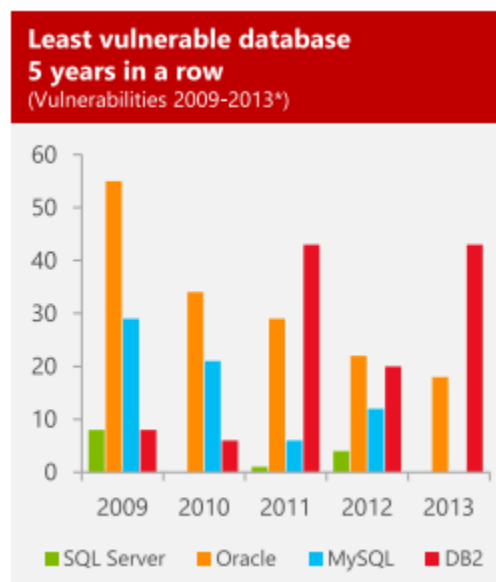
- Complejidad, es preciso comprender su funcionalidad para poder realizar un buen uso.
- Coste del equipamiento adicional.
- Vulnerable a los fallos, por el hecho de que todo esté centralizado en el SGBD.
- Tipos de Campos

1.3.3. SQLSERVER

"Microsoft® SQL Server™ es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos." (Microsoft, s.f.)

Dentro de las características más importantes de sql server se encuentran el soporte de transacciones, creación de store-procedures y modo de trabajo cliente-servidor, además de la seguridad necesaria. (Wikipedia, s.f.)

Ilustración 1-3 Bases de Datos menos vulnerables



Nota: Adaptado de "SQL Server 2014 & the Data Platform", por Microsoft, 2014, Ediciones de SQL Server, <https://www.microsoft.com/es-es/server-cloud/products/sql-server-editions/overview.aspx>

Sql server cuenta además con una interfaz gráfica, llamada Server Management Studio, que facilita el proceso de ingreso, modificación o visualización de datos. El lenguaje que se utiliza en la interfaz es el Transact-SQL.

1.3.4. TRANSACT-SQL

"Transact-SQL es fundamental para trabajar con SQL Server. Todas las aplicaciones que se comunican con SQL Server lo hacen enviando instrucciones Transact-SQL al servidor, independientemente de la interfaz de usuario de la aplicación." (Microsoft, s.f.)

Transact-SQL es simplemente una extensión del lenguaje SQL utilizado por Microsoft para interactuar con la base de datos. Además, permite el manejo de variables y flujos de control.

Se pueden trabajar aplicaciones con interfaces de usuarios o páginas web que indiferentemente enviarán declaraciones en Transact-SQL.

Las funciones integradas que maneja son funciones de conjuntos de filas, funciones de agregado, funciones de categoría y funciones escalares.

Dichas funciones pueden ser deterministas o no deterministas, Microsoft en su página web indica que "Las funciones son deterministas cuando devuelven siempre el mismo resultado cada vez que se llaman con un conjunto específico de valores de entrada. Las funciones son no deterministas cuando es posible que devuelvan distintos resultados cada vez que se llaman con un mismo conjunto específico de valores de entrada." (Microsoft, s.f.)

1.4. Lenguajes de programación.

1.4.1. Definición

La Real Academia de la Lengua Española define lenguaje como "Conjunto de señales que dan a entender algo" (RAE, s.f.). Por otro lado, podemos definir la programación como el proceso de escribir un conjunto de instrucciones para su posterior ejecución por un computador.

A partir de esto, se define un lenguaje de programación como "un sistema de comunicación que posee una determinada estructura, contenido y uso." (Definición, s.f.)

Para entender la evolución de los lenguajes de programación, es necesario centrarnos en la historia de los mismos. Es así como comenzamos por **código máquina**, es comprendido por las máquinas, pero dificultosamente por las personas, esto es, cadenas de 0 y 1. Por esta razón se desarrolló el **lenguaje ensamblador**, utilizado por los primeros programadores, reemplazaba los 0 y 1, por palabras o siglas en inglés. Posteriormente, se desarrollaron diversos lenguajes de programación, los cuales poseen una estructura lo más semejante posible al lenguaje natural, e estos se los denomina, **lenguajes de alto nivel**. (Wikipedia, s.f.)

1.4.2. C#

Según Microsoft "C# es un lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C." (Microsoft, s.f.)

C# aparece en el año 2002 con grandes similitudes con respecto a Java y C++, siendo una de las partes más importantes en el movimiento de .NET que Microsoft proponía. (Bell & Parr, 2010)

C# soporta una de las tendencias más conocidas actualmente en el desarrollo de programas, la programación orientada a objetos.

1.4.2.1. Características

Entre las características más importantes de C# tenemos que (Microsoft, s.f.)

- C# es un lenguaje de programación simple, diseñado para escribir aplicaciones de todo tipo, pero con un énfasis en las empresas.
- C# constituye el siguiente paso de C++, con elementos similares a Java. Por lo tanto, C++ y C# tienen muchas cosas en común
- Dado que es el siguiente paso dentro del lenguaje C, posee considerables mejoras en seguridad de tipos, área de instrucciones, expresiones y operadores.
- C# permite interactuar con .NET framework

1.4.3. .NET Framework

.NET es un framework desarrollado por Microsoft en el 2002, permite la compilación y ejecución de aplicaciones y servicios Web. En la página web de Microsoft podemos encontrar que los objetivos a los que se enfoca .NET son (Microsoft, s.f.):

- Programación orientada a objetos, en la que el código pueda ser almacenado y ejecutado en forma local, web y reporta.
- Minimiza los conflictos en el despliegue y manejo de versiones de un software.
- Su entorno de ejecución elimina problemas de rendimiento.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.

A lo largo de los años se han ido mejorando las versiones del framework, y con esto añadiendo nuevos componentes, en la siguiente imagen se puede apreciar la evolución del framework y a su vez de los componentes que brinda

Ilustración 1-4 .NET Framework



Nota: Adaptado de “The .NET Framework stack”, por Wikipedia, 2007, https://en.wikipedia.org/wiki/.NET_Framework#/media/File:DotNet.svg

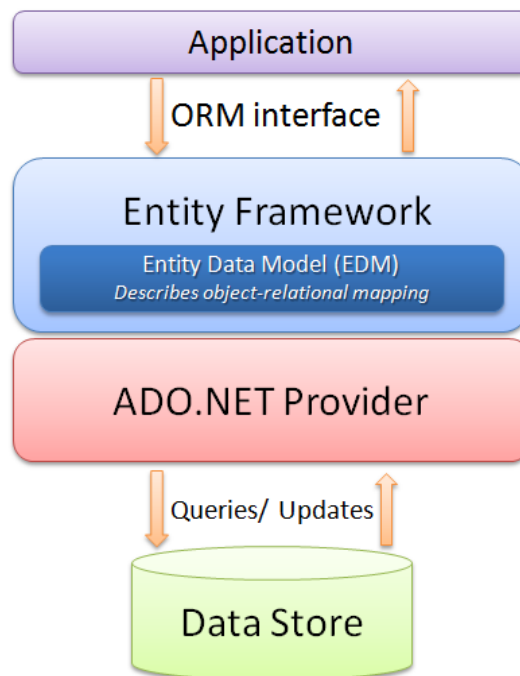
1.4.3.1. Entity Framework

"Entity Framework (EF) es un asignador objeto-relacional que permite a los desarrolladores de .NET trabajar con datos relacionales usando objetos específicos del dominio. Elimina la necesidad de la mayor parte del código de acceso a datos que los desarrolladores suelen tener que escribir." (Microsoft, s.f.)

Básicamente el entity framework es una tecnología desarrollada por Microsoft para el acceso a datos de una aplicación. El EF toma los paradigmas que existían en el acceso a datos y los mejora.

El modelo conceptual es la característica más importante del EF pues permite la transmisión de datos. (Microsoft, s.f.)

Ilustración 1-5 Funcionamiento del Entity Framework



Nota: Adaptado de "Creando la capa de datos: Entity framework vs Linq to SQL", por Victor Fernández Portero, 2011, <http://www.compilando.es/2011/05/12/creando-la-capa-de-datos-entity-framework-vs-linq-to-sql/>

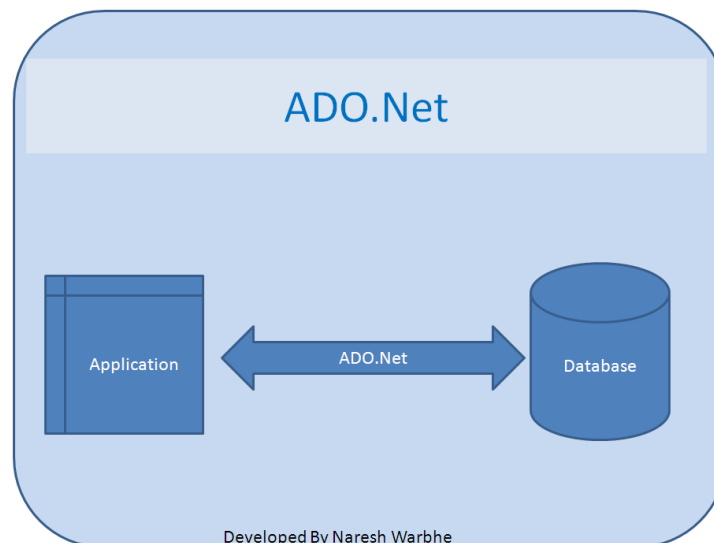
El Entity framework es un paquete que se incluye en la herramienta de desarrollo de software Visual Studio o puede ser instalado por medio de consola.

1.4.3.2. ADO.NET

ADO.NET constituye una parte integral de .NET Framework, ofrece componentes para la creación de aplicaciones de uso compartido de datos. Microsoft menciona que "ADO.NET satisface diversas necesidades de desarrollo, como la creación de clientes de base de datos front-end y objetos empresariales de nivel medio que utilizan aplicaciones, herramientas, lenguajes o exploradores de Internet." (Microsoft, s.f.)

"ADO.NET proporciona acceso a orígenes de datos como SQL Server y XML, así como a orígenes de datos expuestos mediante OLE DB y ODBC." (Microsoft, s.f.)

Ilustración 1-6 ADO.NET



*Nota: Adaptado de “ADO.NET vs Entity Framework”, por Eleazar Caceres, s.f.,
<http://www.ecaceres.com/2013/07/adonet-vs-entity-framework.html>*

La gran ventaja de ADO.NET es que separa la capa de acceso a datos de la manipulación de los mismos.

1.4.3.3. Modelos .edmx

Microsoft define a los modelos emdx como "un archivo XML que define un modelo conceptual, un modelo de almacenamiento y la asignación entre estos modelos. Un archivo .edmx también contiene información que utiliza ADO.NET Entity Data Model Designer (Entity Designer) para presentar un modelo gráficamente." (Microsoft, s.f.)

El modelo es generado a partir de una base de datos y se actualiza únicamente si existen cambios en la base de datos con las cual se comunica el software.

El modelo contiene el motor en tiempo de ejecución (edmx:Runtime), el Contenido de Designer (edmx:Designer), las Propiedades del modelo conceptual y las Propiedades del archivo .edmx. (Microsoft, s.f.)

1.5. Sistemas de distribuidos

1.5.1. Definición

Un sistema distribuido se define a aquel en el que el procesamiento de la información se realiza en varias computadoras. (Sommerville, 2005)

En el libro *Sistemas Distribuidos, Concepto y Diseño*, se dice que "Un sistema distribuido es aquel en que los componentes localizados en computadores, conectados en red, comunican y coordinan sus acciones únicamente mediante el paso de mensajes." (Colouris, Dollimore, & Kindberg, 2001). Como ejemplos de sistemas distribuidos tenemos el internet, intranets, computación móvil, etc.

El principal motivo para la implementación de sistemas distribuidos es que los recursos son administrados por servidores, y los clientes pueden acceder a ellos.

Colouris establece seis características principales de los sistemas distribuidos

- **Compartición de Recursos:** Los recursos se encuentran en una computadora, y tienen acceso otras computadoras en la red.
- **Apertura:** Un sistema informático es abierto si el sistema puede ser extendido de diversas maneras
- **Concurrencia:** Se da ejecución paralela se da porque: muchos usuarios interactúan simultáneamente con programas de aplicación; o muchos procesos servidores se ejecutan concurrentemente.
- **Escalabilidad:** Capacidad que tiene el sistema para mantener su nivel de complejidad y rendimiento sin importar cuanto crezca el sistema.

- **Tolerancia a Fallos:** El sistema debe de ser capaz de recuperarse de los fallos, y dar continuidad a su servicio.
- **Transparencia:** El sistema es percibido por el cliente, como un todo, es decir no se percibe la separación de los componentes de un sistema distribuido.

1.5.1.1. N Capas

"Aplicaciones de datos de N-Capas son aplicaciones de datos que están separados en múltiples niveles. También se llaman "aplicaciones distribuidas" y "aplicaciones de varios niveles," las aplicaciones de N-Capas separan el procesamiento en niveles que se distribuyen entre el cliente y el servidor. Al desarrollar aplicaciones que accedan a datos, se debe tener una clara separación entre los diferentes niveles que componen la aplicación." (Microsoft, s.f.)

Generalmente existe una capa de presentación que es en dónde el usuario se comunica con la aplicación. Una capa intermedia para realizar toda la lógica del negocio y finalmente una capa de datos, que consiste en el servidor.

Ilustración 1-7 N Capas



Nota: Adaptado de “¿Arquitectura n-Tier o Arquitectura n-Layer?”, por Diego Rojas, 2008, <http://icomparable.blogspot.com/2008/10/arquitectura-n-tier-o-arquitectura-n.html>

1.5.2. XML

XML (Extensible Markup Language) es un lenguaje de marcado que fue desarrollado por el World Wide Web Consortium (W3C) para el almacenamiento de datos. Según la W3C "Extensible Markup Language (XML) es un formato de texto simple, muy flexible derivado de SGML (ISO 8879). Originalmente diseñado para cumplir con los retos de la publicación electrónica a gran escala, XML también está desempeñando un papel cada vez más importante en el intercambio de una amplia variedad de datos en la Web y en otros lugares." (W3C, 2015)

El documento XML está conformado por una cabeza, en la cual se colocan datos sobre las versiones y la declaración del documento, y un cuerpo para colocar la información.

Entre las ventajas más importantes del uso de XML se encuentran que es extensible, es sencillo y se asigna contexto a los datos.

1.5.3. HTML

HTML es un lenguaje de programación para crear instrucciones que serán ejecutadas por un navegador y originar páginas web. (Equipo Vértice, s.f.)

Las siglas significan HyperText Markup Language, "... el hipertexto es aquel texto que pulsamos con el ratón del ordenador y nos conduce a otro texto cuando utilizamos Internet. Pero además de texto, el hipertexto puede estar formado de tablas, imágenes u otros elementos" (Equipo Vértice, s.f.)

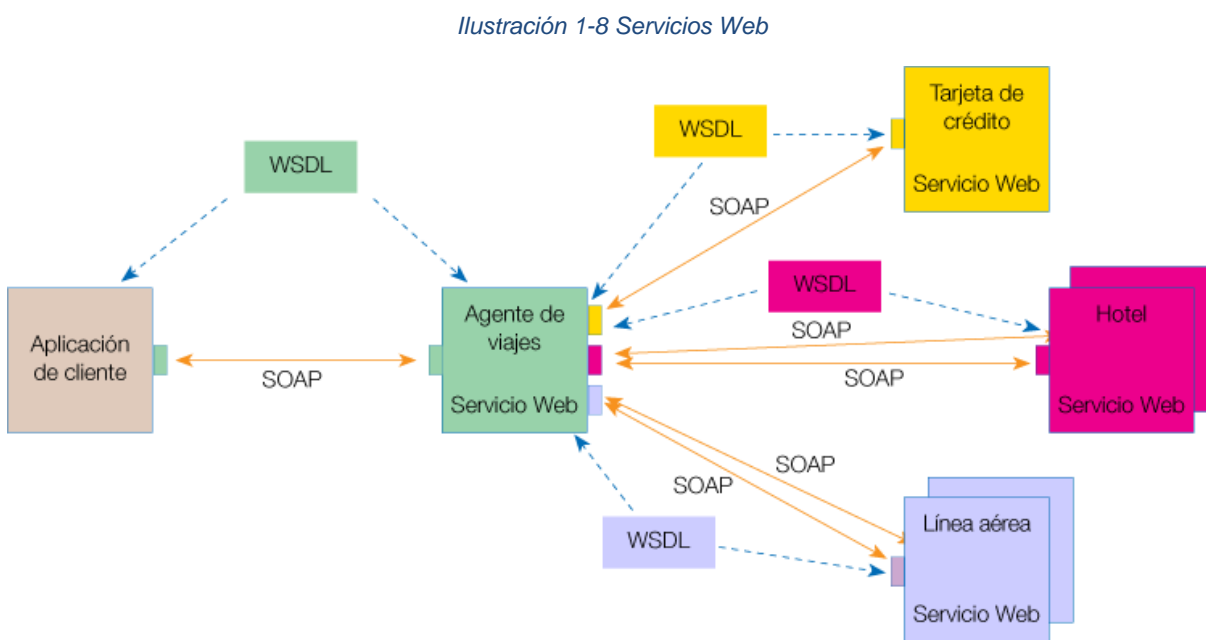
En otras palabras, HTML sirve para dar estructura a un documento web, es decir proporciona las herramientas para dar formato, pero cabe recalcar que no describe la apariencia ni el diseño del documento.

1.5.4. Servicios web

"Existen múltiples definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Una posible

sería hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web." (W3C, s.f.)

Los servicios web permiten comunicación entre diferentes aplicaciones, así como desglosar procesos antes de ser utilizados por el cliente. En la siguiente imagen podemos ver un ejemplo de una aplicación con los servicios web.



*Nota: Adaptado de “Guía Breve de Servicios Web”, por W3C, s.f.,
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>*

En el ejemplo podemos visualizar cómo los servicios web pueden ser consumidos por diferentes agentes para acceder a la información requerida. Si nuevos elementos o páginas desean añadirse será una implementación más sencilla.

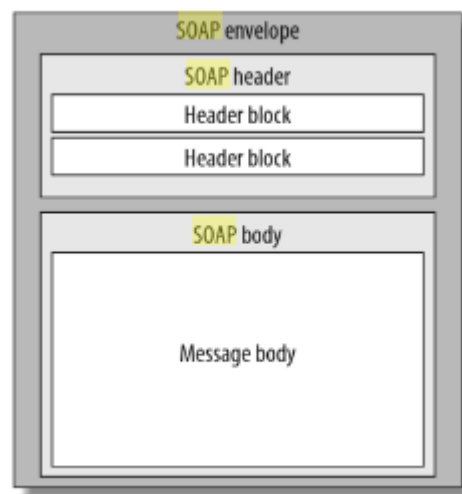
1.5.4.1. SOAP

Simple Object Access Protocol, es un protocolo creado gracias a la cooperación de grandes empresas, como Microsoft, IBM, SAP, entre otras, es una derivación de un protocolo creado por David Winer, 1998. Actualmente SOAP se encuentra bajo auspicio de W3C. (González, 2004)

La W3C menciona que SOAP "Se trata de un protocolo basado en XML, que permite la interacción entre varios dispositivos y que tiene la capacidad de transmitir información compleja. Los datos pueden ser transmitidos a través de HTTP, SMTP, etc." (W3C, s.f.)

Un mensaje SOAP contiene una cabecera (opcional) y un cuerpo (obligatorio). La cabecera contiene la información sobre como el mensaje es procesado. El cuerpo contiene el mensaje que va a ser procesado y entregado. (Snell, Tidwell, & Kulchenko, 2002)

Ilustración 1-9 SOAP



Nota: Adaptado de “Programming Web Services with SOAP”, por Snell, James; Tidwell, Doug; Kulchenko, Pavel; 2002, Sebastopol: O’Reilly & Associates

1.5.5. ASP.NET

Microsoft desarrolla el framework de ASP.NET como una solución para las dificultades que poseía asp (separación de presentación y contenido), se define como "un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales

con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc." (Microsoft, s.f.)

ASP.NET permite el trabajo con controles, posee un compilador, una infraestructura de seguridad, capacidad de depuración y un entorno de diseño. (Microsoft, s.f.)

- El marco se ejecuta en un servidor para una representación dinámica de las páginas.
- El compilador de ASP.NET mejora el rendimiento interno
- El entorno de diseño facilita la creación de páginas web
- Gracias a la infraestructura de seguridad se pueden autenticar usuarios por medio de Windows o por la misma base de datos

1.5.5.1. Web forms

Las páginas de ASP.NET son denominadas Web Forms, formados por etiquetas HTML y etiquetas de controladores que permiten la interacción con el usuario. Los formularios web cuentan con una extensión ASPX. Es muy similar a otras tecnologías de desarrollo web como PHP y JSP.

1.5.5.2. Controles

Microsoft define a los controles de elementos como "Un conjunto integrado de controles concebidos para crear sitios Web que permiten al usuario modificar el contenido, el aspecto y el comportamiento de las páginas Web directamente en un explorador. Los temas de esta sección proporcionan información acerca de los elementos Web, su funcionamiento y su utilización para crear páginas Web de ASP.NET que puede personalizar el usuario." (Microsoft, 2007)

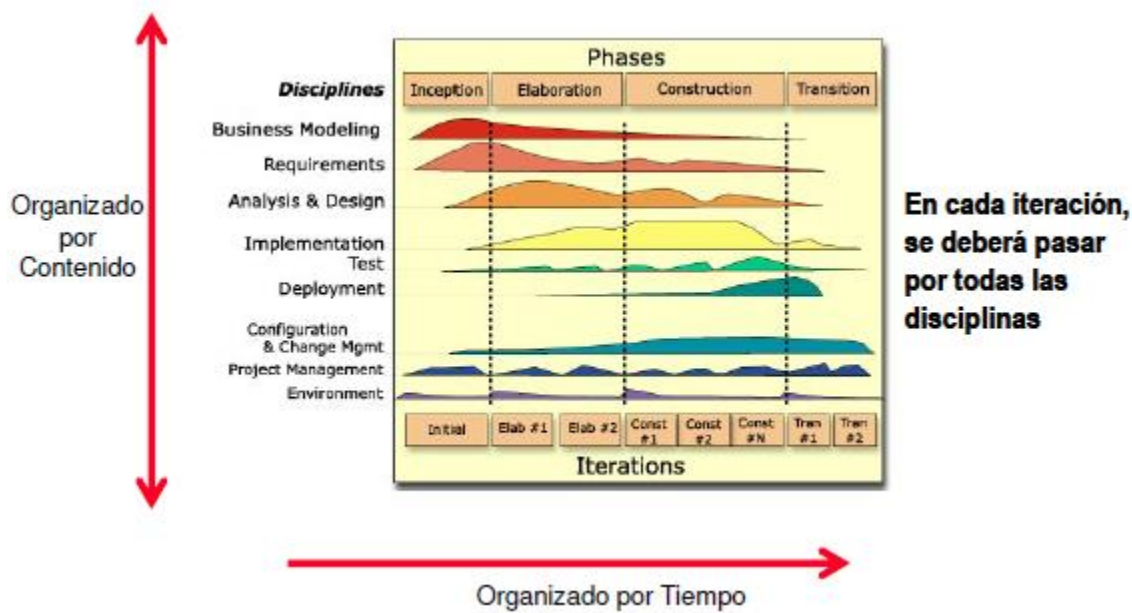
1.6. Metodología de Desarrollo del Proyecto

1.6.1. Método RUP

El Proceso Unificado Rational "es un proceso para el desarrollo de software y junto con UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas" (Hidalgo, s.f.)

Ha sido enriquecido con las experiencias de los métodos de Booch y Rumbaugh (OMT) dentro de la empresa Rational la cual también contribuye en la definición del lenguaje unificado UML. (De la Cruz, 2014)

Ilustración 1-10 Fases del método RUP



Nota: Adaptado de "Método RUP", por De la Cruz, Fabián; 2014, Quito: Clases dictadas PUCE

RUP clasifica las actividades para el desarrollo de software en 9 disciplinas, 5 de ellas se encuentran vinculadas con el producto y tres para el apoyo de las actividades de desarrollo.

RUP Es un proceso que integra (De la Cruz, 2014):

- Ciclos y fases
- Flujos de trabajo

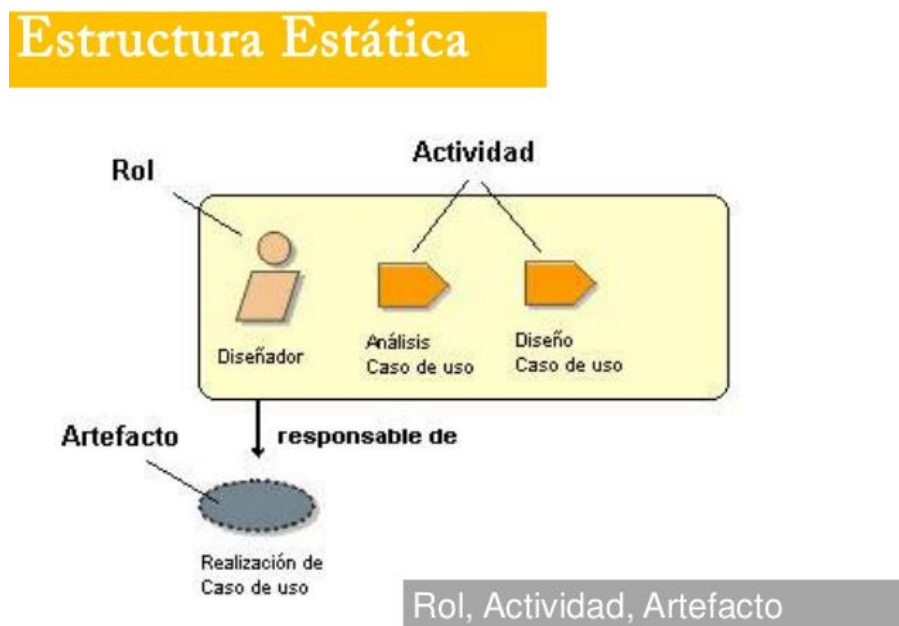
- Manejo de riesgos
- Control de calidad
- Administración de proyecto
- Control de configuración

Es importante mencionar que las estimaciones son la base para una planificación para los proyectos, ya que proporcionan una visión general de los esfuerzos y recursos necesarios para el diseño y desarrollo de un software. (Cardozzo, 2016)

Según Per Kroll y Philippe Kruchten para el éxito de la metodología es necesario asignar roles a cada miembro del equipo. Se menciona que un rol consiste en un papel que se va a desempeñar y que describe qué y cómo se debe trabajar.

Tales roles se ligan a diferentes actividades que se van a desarrollar y generalmente tienen un objetivo claro. (Kroll & Kruchten, 2003)

Ilustración 1-11 Roles en RUP



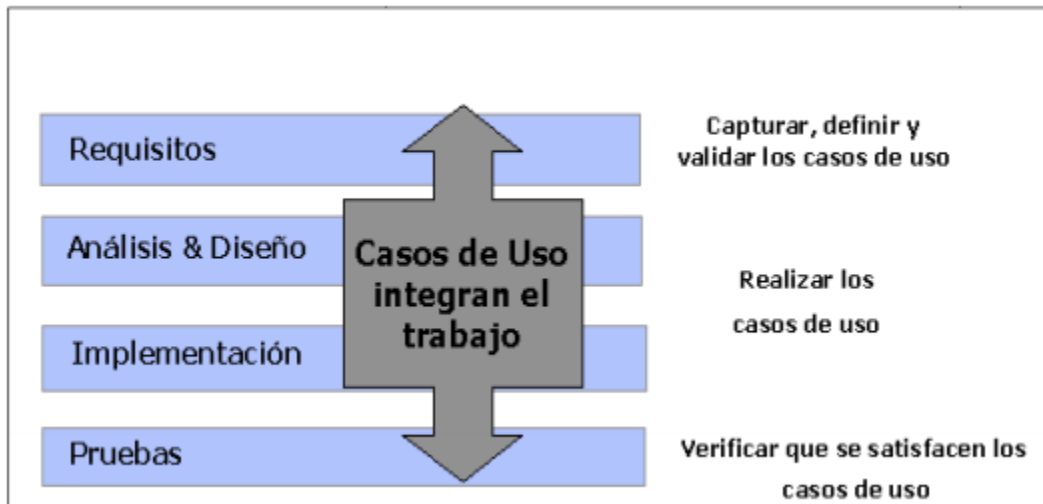
Nota: Adaptado de “The Rational Unified Process Made Easy: A Practitioner’s Guide to the RUP”, por Kroll; Per Kruchten; Philippe; 2003, Boston

Finalmente se menciona un artefacto que captura lo que se ha hecho. En la imagen encontramos al diseñador, responsables de diferentes actividades y el resultado de tales actividades.

Las características de esta metodología son: (Universidad Nacional Autónoma de México, s.f.)

- **Dirigido a casos de uso:** Son una herramienta para especificar los requisitos del sistema, en ellos se basa el diseño, y además, son una guía para el desarrollo y las pruebas.

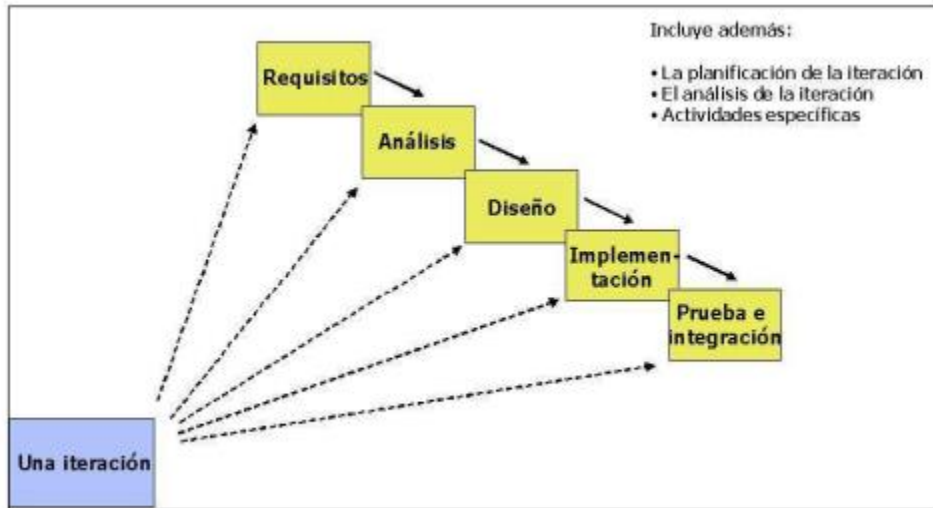
Ilustración 1-12 Método SCRUM Características



Nota: Adaptado de “Proceso Unificado Rational Aplicado”, por UNAM; s.f., México

- **Centrado en la arquitectura:** Establecer la arquitectura en una etapa temprana es esencial para que ante posibles cambios tenga un fuerte impacto.
- **Proceso iterativo e incremental:** El proyecto se divide en partes más pequeñas, lo que permite mayor equilibrio en el desarrollo.

Ilustración 1-13 Iteración



Nota: Adaptado de “Proceso Unificado Rational Aplicado”, por UNAM; s.f., México

2. CAPÍTULO 2: ANÁLISIS

2.1. Situación Actual

El Museo QCAZ de la PUCE ubicado en la Facultad de Ciencias Exactas, Escuela de Biología, maneja los datos de las colecciones biológicas; una colección biológica es el conjunto de datos sobre todas las especies de una rama zoológica, en éste caso la rama de herpetología, que encierra los campos de anfibios (Amphibia) y reptiles (Reptilia).

La colección del Museo QCAZ es de gran relevancia no solo a nivel nacional, sino que, debido a la riqueza en la fauna del Ecuador, los datos tienen importancia internacionalmente. Investigadores de todo el mundo tienen acceso a ésta información y por ello se debe asegurar su integridad.

El Museo QCAZ ha recopilado miles de registros acerca de las especies de anfibios y reptiles del Ecuador, hay que señalar que para recolectar ésta información de las fuentes primarias se han invertido grandes cantidades de recursos humanos, económicos, entre otros. Los datos han sido almacenados en una base de datos que no presenta las recomendaciones técnicas para gestionar la información de forma eficiente, como almacenar la información en un solo lugar, seguridad en la información y ser abierta para futuros cambios. Además, que no se encuentra alojada en un ambiente adecuado, ya que actualmente está configurada en una máquina normal, por tal motivo dicha información está expuesta a pérdidas por daño el equipo, incendios, catástrofes naturales, etc.

Existen problemas actuales que han sido ocasionados por el diseño original de la base, y como hemos mencionado anteriormente la información es de gran importancia no sólo para la Universidad, sino para investigadores de todo el mundo. Los problemas principales están relacionados con el uso de bases de datos ya que sus interfaces permiten manipular la información sin pedir confirmación, por esto algún usuario inexperto fácilmente podría borrar un registro, muchos registros o incluso todos, así como modificar la información. El modelo de la base de datos presenta muchas fallas de concepto, el diseño original crea diferentes bases para guardar la información.

Además, existe heterogeneidad de las bases de datos que maneja el Museo, no hay que olvidar que mucha de la información que está en la base de datos actual, se publica en el sitio web del

Museo. El sitio web del Museo trabaja con otra base de datos, por tal motivo se presentan muchos problemas al momento de migrar la información de un sistema al otro, un problema común es olvidarse de tener que actualizar ambas bases.

2.2. Levantamiento de Requerimientos

La metodología RUP indica que es una necesidad establecer los requerimientos para poder tener así una mejor visión del sistema para su posterior diseño e implementación.

Para el levantamiento de requerimientos se mantuvo reuniones con el personal del Museo QCAZ más vinculado con el Sistema Filemaker, en base a estas reuniones se generaron bitácoras, en las cuales se especifican las funcionalidades deseadas, y problemas que se deberán resolver para el nuevo sistema. **(Ver Anexo 1)**

Para la realización de la documentación de las Especificaciones de Requerimientos de Software (SRS) se tomó como modelo el estándar emitido por la IEE 830 el 22 de octubre de 2008. **(Ver Anexo 2)**

En el SRS se detallan los requerimientos funcionales y no funcionales del sistema, requisitos en del sistema y de los usuarios, así como también las posibles restricciones a tomar en cuenta.

2.3. Sistema Propuesto

Es de gran importancia poseer una base de datos bien organizada y generar facilidades por medio de una buena interfaz en el programa. En general se hace énfasis en el buen diseño de la base de datos y el programa que esté vinculado.

También tenemos en cuenta mantener al sistema con las funcionalidades que permitía anteriormente, y asegurar que sus usuarios no tendrán problemas al realizar actividades comunes.

2.3.1. Casos de uso

2.3.1.1. Caso de uso general

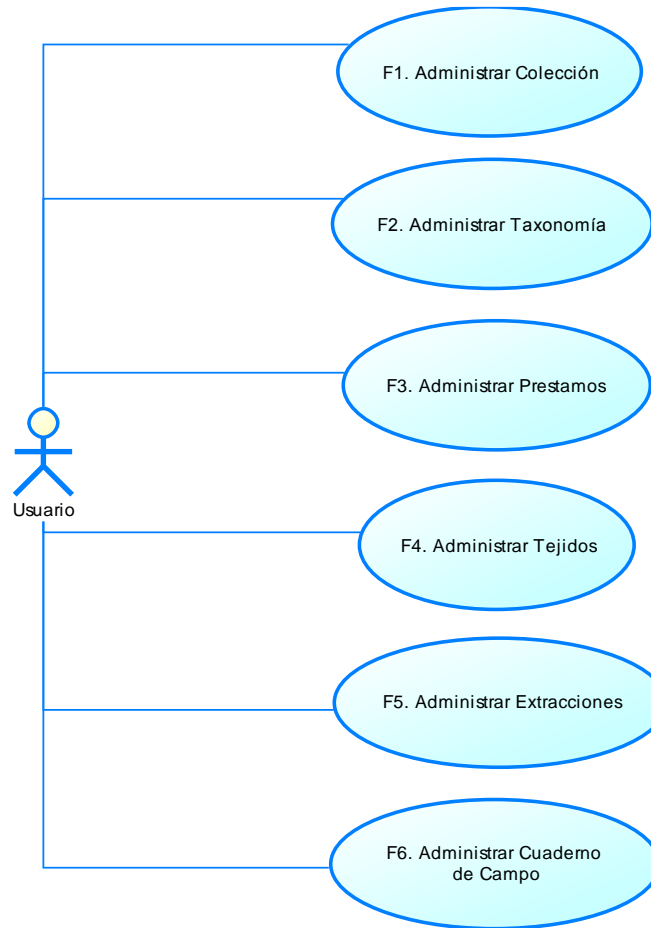


Diagrama 2-1 General - Casos de Uso

2.3.1.2. F0. Administrar sesión

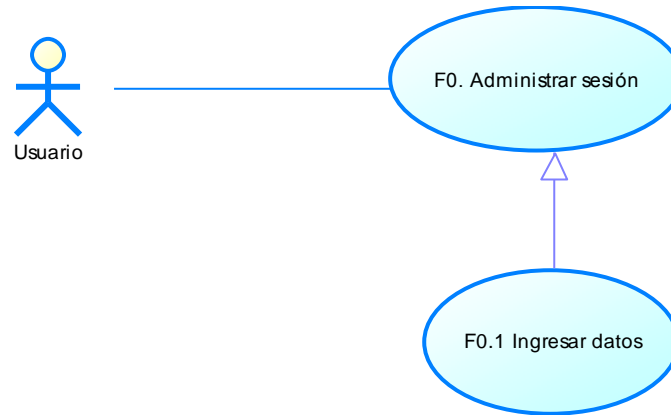


Diagrama 2-2 F0. Administrar Sesión - Casos de Uso

Tabla 2-1 F0. Administrar Sesión - Flujos

F0. Administrar Sesión		
Descripción	Todos los usuarios del sistema harán uso de la Administración de Sesión como requisito para ingresar al sistema	
Flujo Básico	Actor	Sistema
	<i>F0.1. Ingresar Datos</i>	
	• Ingresar al sistema	Muestra una pantalla con los campos necesarios para el ingreso al sistema
	• Ingresar sus datos	
	• Selecciona <i>Inicio de Sesión</i>	Valida los datos y permite el ingreso al sistema
	○ Presiona el botón Insertar	Guarda los Datos Ingresados
Flujos Alternos	F0.1. Datos ingresados erróneos	Envía un mensaje informando que los datos ingresados no son correctos.
Pos condiciones	Todos los Casos de Uso.	

Autores: Pablo Almeida & Vanessa Soria C.

2.3.1.3. F1. Administrar Colección

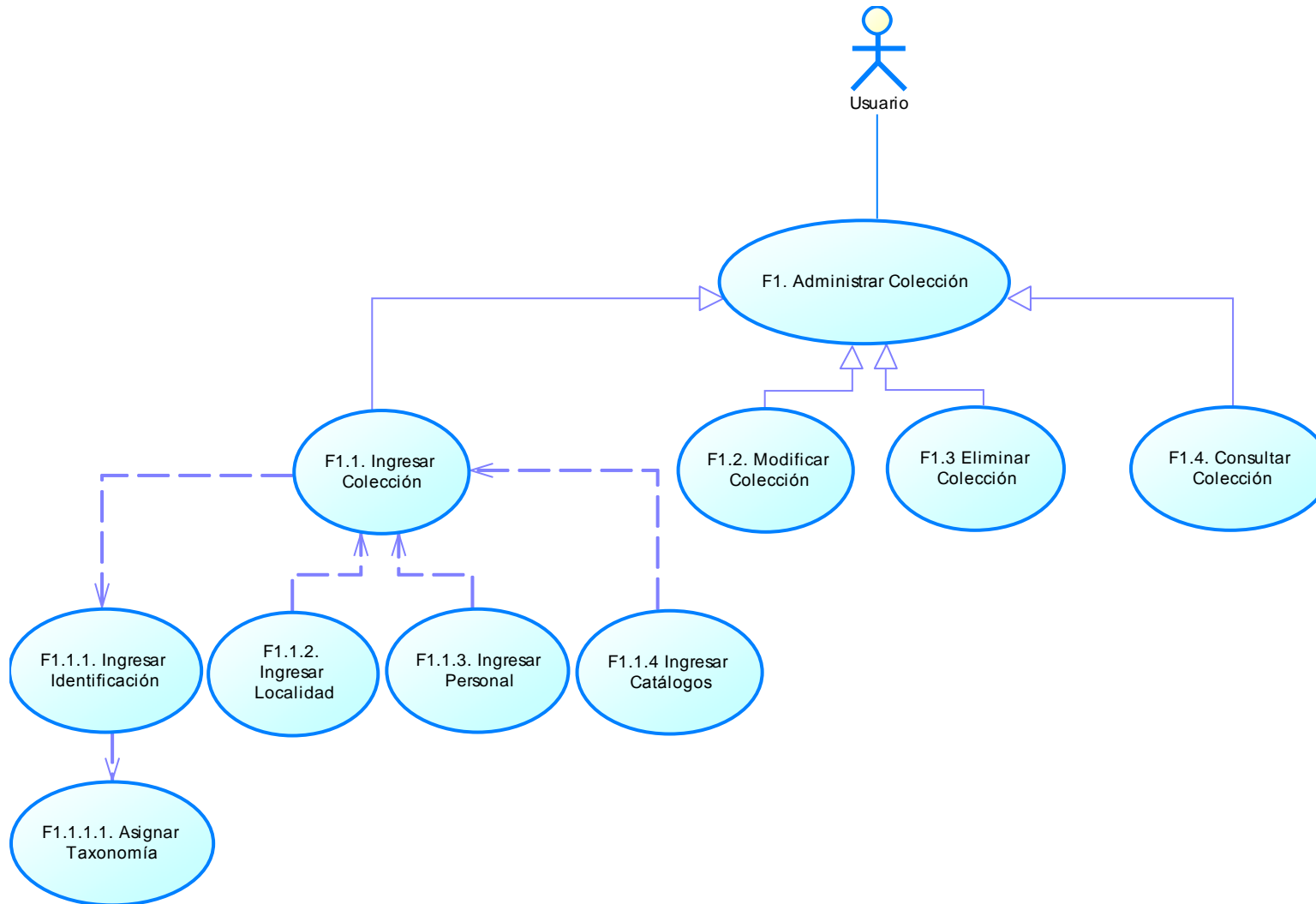


Diagrama 2-3 F1. Administrar Colección - Casos de Uso

Tabla 2-2 F1. Administrar Colección - Flujos

F1. Administrar Colección			
Descripción	Los actores, una vez que ingresen al sistema tendrán acceso a la Administración de Colección y a todas las funciones que esta Administración ofrece.		
Flujo Básico	Actor	Sistema	
	Escoge la opción Administrar Colección Zoología del menú principal.	Muestra en la Pantalla una tabla con la consulta general de las Colecciones registradas	
	F1.1. Ingresar Colección		
	• Selecciona la opción de Ingresar Colección.	Muestra la pantalla con los formularios necesarios para ingresar un nuevo espécimen.	
	• Asigna la taxonomía	Muestra una lista de taxonomías y valida la escogida.	
	• Selecciona la pestaña Colección Zoología y llena el formulario de registro	Muestra el formulario correspondiente al registro y valida los datos.	
	○ Presiona el botón Insertar	Guarda los Datos Ingresados	
	• Selecciona la pestaña Localidad Herpetología y llena el formulario de registro	Muestra el formulario correspondiente a la localidad del Especimen y valida los datos.	
	○ Presiona el botón Insertar	Guarda los Datos Ingresados	
	• Presiona la pestaña Personal y llena los datos	Muestra los formularios para el registro del personal	
	○ Presiona el botón Insertar	Guarda los Datos Ingresados	
	• Presiona la pestaña Catálogos y llena los datos	Muestra los formularios para el registro de los catálogos	
	○ Presiona el botón Insertar	Guarda los Datos Ingresados	
	F1.2. Modificar Colección		
	• Selecciona el espécimen que desea modificar	Muestra la Información del espécimen	
	• Selecciona el botón <i>Editar</i> de la Información que desea modificar e ingresa la nueva información.	Valida los datos	
	• Pulsa el botón <i>Actualizar</i>	Actualiza la información modificada.	
	F1.3. Eliminar Colección		

	<ul style="list-style-type: none"> • Escoge la información del espécimen que se desea eliminar. Pulsa el botón eliminar 	El sistema elimina la información de la base de datos.
	F1.4. Consular Colección	
	<ul style="list-style-type: none"> • Ingresar en el QCAZ de la Colección que desea buscar 	El sistema muestra la información del espécimen
Flujos Alternos	F1.1. No se registran los datos obligatorios.	Envía un mensaje informando que faltan datos por llenar.
	F1.3. No se puede eliminar Colección	Sistema Informa al Usuario
	F1.4. No se encontró el QCAZ buscado	Muestra una tabla vacía
Precondiciones	Administrar Taxonomía.	
Pos condiciones	Todos los Casos de Uso.	

Autores: Pablo Almeida & Vanessa Soria C.

2.3.1.4. F3. Administrar préstamos

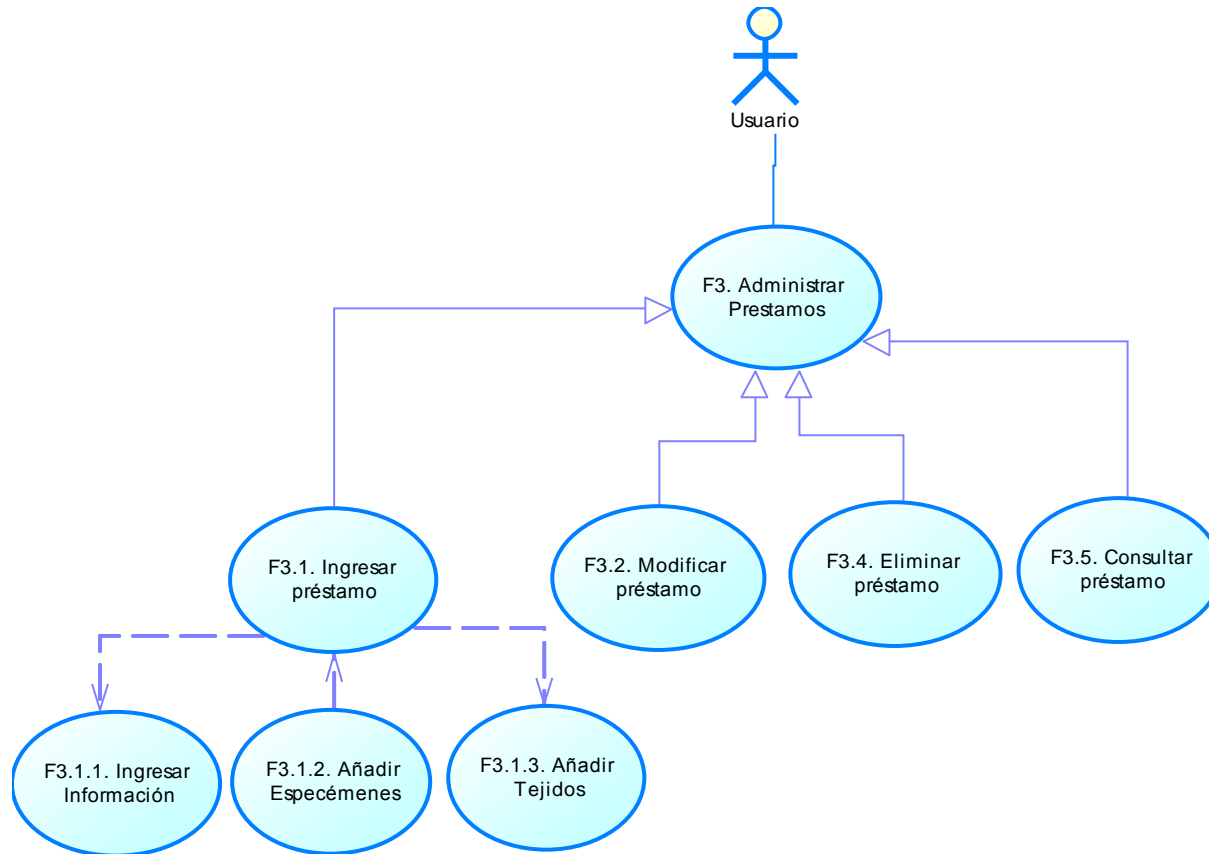


Diagrama 2-4 F3. Administrar Préstamos – Casos de Uso

Tabla 2-3 F3. Administrar Préstamos - Flujos

F3. Administrar Prestamos		
Descripción	Los actores, una vez que ingresen al sistema tendrán acceso a la Administración de Préstamos y a todas las funciones que esta Administración ofrece.	
Flujo Básico	Actor	Sistema
	Escoge la opción Administrar Préstamos del menú principal.	Muestra en la Pantalla una tabla con la consulta general de los Préstamos Registrados
	F3.1. Ingresar Préstamo	
	• Selecciona la opción <i>Ingresar Nuevo</i>	Muestra la pantalla con los formularios necesarios para el registro del préstamo.
	• Ingresa los datos del formulario para el registro del préstamo. (fechas y condiciones)	Valida los datos
	○ Presiona el botón Insertar	Guarda los Datos Ingresados
	• Selecciona la pestaña <i>Préstamo de Tejido</i> y selecciona el tejido.	Muestra el formulario correspondiente al registro de tejidos.
	○ Presiona el botón Insertar	Guarda los Datos Ingresados
	• Selecciona la pestaña <i>Préstamo de Colecciones</i> y selecciona el espécimen.	Muestra el formulario correspondiente al registro de especímenes.
	○ Presiona el botón Insertar	Guarda los Datos Ingresados
	F3.2. Modificar Préstamo	
	• Selecciona el préstamo que desea modificar	Muestra la Información del préstamo
	• Selecciona el botón <i>Editar</i> e ingresa la nueva información.	Valida los datos
	• Pulsa el botón <i>Actualizar</i>	Actualiza la información modificada.
	F3.3. Eliminar Préstamo	
	• Pulsa el botón <i>Eliminar</i> correspondiente al préstamo que desea borrar	El sistema elimina la información de la base de datos.
	F3.4. Consultar Préstamo	

*“Sistema para la administración de colecciones biológicas y compartición de la información
Museo QCAZ”*

	<ul style="list-style-type: none"> • Ingresar la información del préstamo que desea buscar 	El sistema muestra la información del préstamo
Flujos Alternos	F3.1. No se registran los datos obligatorios.	Envía un mensaje informando que faltan datos por llenar.
	F3.4. No se encontró el préstamo buscado	Muestra una tabla vacía
Precondiciones	Administrar Colección Administrar Tejido	

***Autores:** Pablo Almeida & Vanessa Soria C.*

2.3.1.5. F4. Administrar Tejidos

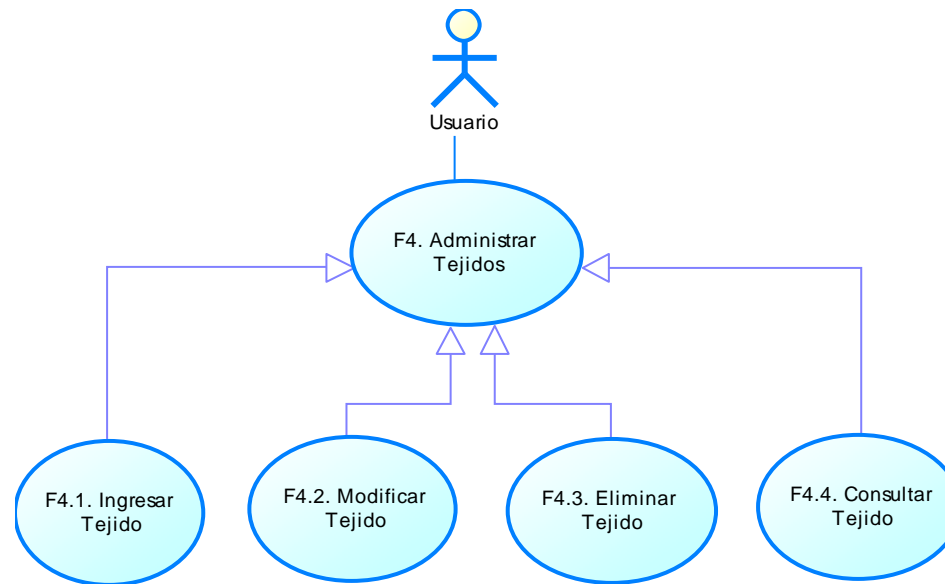


Diagrama 2-5 F4. Administrar Tejidos - Casos de Uso

Tabla 2-4 F4. Administrar Tejidos - Flujos

F4. Administrar Tejidos		
Descripción	Los actores que ingresen al sistema tendrán acceso a la Administración de Tejidos y a todas las funciones que esta Administración ofrece.	
Flujos Básicos	Actor	Sistema
	Escoge la opción Administrar Tejidos del menú principal.	Muestra una pantalla con una tabla de consulta general de los tejidos
	F4.1. Ingresar Tejidos	
	Selecciona la opción de Ingresar Nuevo.	Muestra el formulario para ingresar un nuevo tejido.
	Llena los campos necesarios.	Verifica los datos ingresados.
	Presiona el botón Guardar.	Guarda la información y se muestra el formulario de los catálogos
	Llena la información de los catálogos y presiona el botón guardar	Guarda la información de los catálogos
	F4.2. Modificar Tejido	
	Escoge la información del tejido que desea modificar.	Muestra la información del tejido.
	Ingresar la nueva información en los campos que desea modificar.	Verifica los datos.
	Presiona el botón Guardar.	Actualiza la información modificada.
	F4.3. Eliminar Tejido	
	Escoge el tejido que desea eliminar. Pulsa el botón eliminar.	El sistema elimina la información de la base de datos.

	F4.4. Consultar Tejido	
	Ingresar la información del tejido que desea buscar	El sistema muestra los datos del tejido.
Flujos Alternos	F4.1. No se registran los datos obligatorios.	Envía un mensaje informando que faltan datos por llenar.
Precondiciones	Administrar Colección	

***Autores:** Pablo Almeida & Vanessa Soria C.*

2.3.1.6. F5. Administrar Extracciones

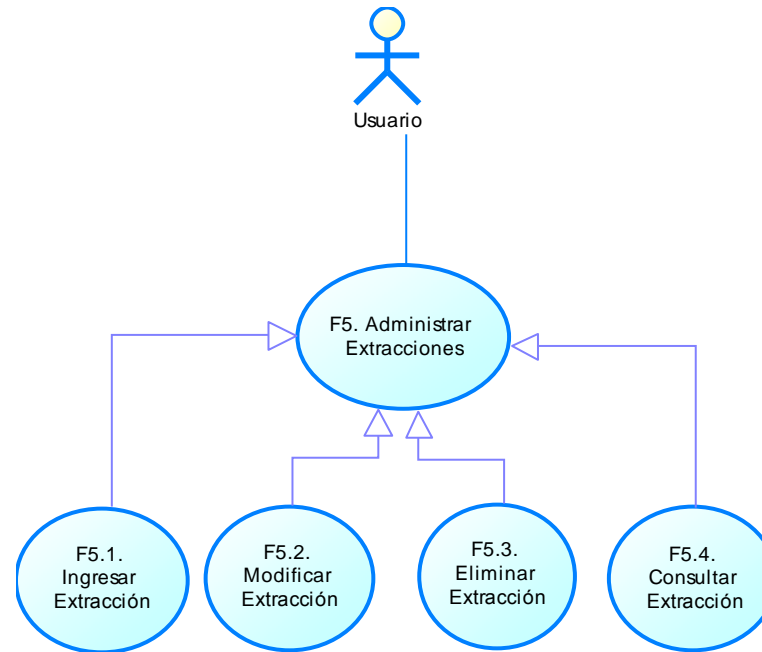


Diagrama 2-6 F5. Administrar Extracciones - Casos de Uso

Tabla 2-5 F5. Administrar Extracción - Flujos

F5. Administrar Extracción		
Descripción	Los actores, una vez que ingresen al sistema, tendrán acceso a la Administración de Extracciones y a todas las funciones que esta Administración ofrece.	
Flujos Básicos	Actor	Sistema
	Escoge la opción Administrar Tejidos del menú principal.	Muestra una pantalla con una tabla de consulta general de los tejidos
	F5.1. Ingresar Extracciones	
	Selecciona la opción de Ingresar Nuevo.	Muestra el formulario para ingresar una nueva extracción.
	Llena los campos necesarios.	Verifica los datos ingresados.
	Presiona el botón Guardar.	Guarda la información y se muestra el formulario de los catálogos
	Llena la información de los catálogos y presiona el botón guardar	Guarda la información de los catálogos
	F5.2. Modificar Extracción	
	Escoge la información de la extracción que desea modificar.	Muestra la información de la extracción.
	Ingresar la nueva información en los campos que desea modificar.	Verifica los datos.
	Presiona el botón Guardar.	Actualiza la información modificada.
	F5.3. Eliminar extracción	
	Escoge la extracción que desea eliminar. Pulsa el botón eliminar.	El sistema elimina la información de la base de datos.
	F5.4. Consultar Extracción	

*“Sistema para la administración de colecciones biológicas y compartición de la información
Museo QCAZ”*

	Ingresar la información de la extracción que desea buscar	El sistema muestra los datos del tejido.
Flujos Alternos	F4.1. No se registran los datos obligatorios.	Envía un mensaje informando que faltan datos por llenar.
Precondiciones	Administrar Colección, Administrar tejido	

***Autores:** Pablo Almeida & Vanessa Soria C*

2.3.1.7. F6. Administrar PCR

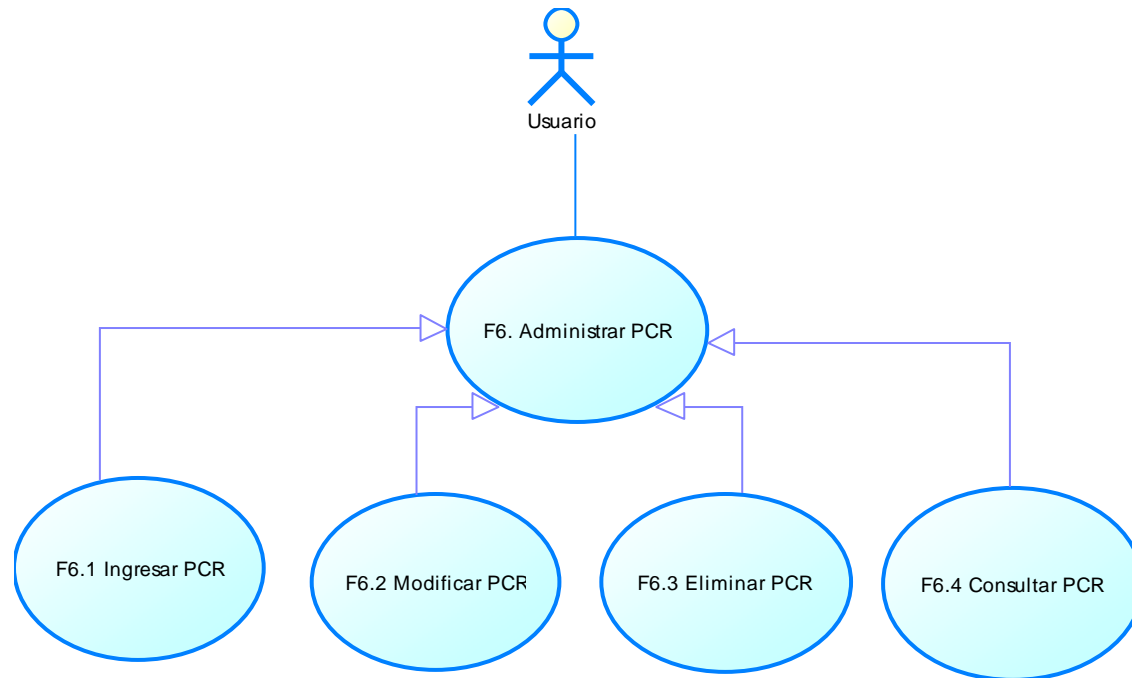


Diagrama 2-7 F6. Administrar PCR

Tabla 2-6 F6. Administrar Amplificación – Flujos

F6. Administrar PCR		
Descripción	Los actores, una vez que ingresen al sistema tendrán acceso a la Administración de Amplificación y a todas las funciones que esta Administración ofrece.	
Flujos Básicos	Actor	Sistema
	Escoge la opción Administrar Amplificaciones del menú principal.	Muestra una pantalla con una tabla de consulta general de las amplificaciones
	F6.1. Ingresar Amplificaciones	
	Selecciona la opción de Ingresar Nuevo.	Muestra el formulario para ingresar una nueva amplificación.
	Llena los campos necesarios.	Verifica los datos ingresados.
	Presiona el botón Guardar.	Guarda la información y se muestra el formulario de los catálogos
	Llena la información de los catálogos y presiona el botón guardar	Guarda la información de los catálogos
	F6.2. Modificar Amplificación	
	Escoge la información de la amplificación que desea modificar.	Muestra la información de la amplificación.
	Ingresa la nueva información en los campos que desea modificar.	Verifica los datos.
	Presiona el botón Guardar.	Actualiza la información modificada.
	F6.3. Eliminar Amplificación	
	Escoge la amplificación que desea eliminar. Pulsa el botón eliminar.	El sistema elimina la información de la base de datos.
	F6.4. Consultar Amplificación	

*“Sistema para la administración de colecciones biológicas y compartición de la información
Museo QCAZ”*

	Ingresar la información de la ampliación que desea buscar	El sistema muestra los datos de la ampliación.
Flujos Alternos	F6.1. No se registran los datos obligatorios.	Envía un mensaje informando que faltan datos por llenar.
Precondiciones	Administrar Colección, Administrar tejido	
Poscondiciones	Todos los Casos de Uso.	

***Autores:** Pablo Almeida & Vanessa Soria C.*

2.3.1.8. F7. Administrar Cuaderno de Campo

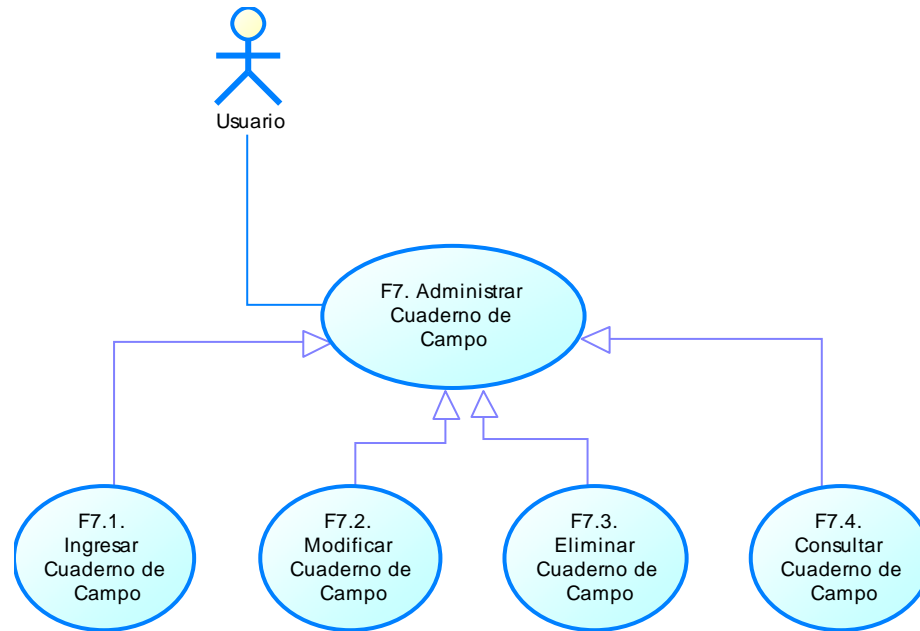


Diagrama 2-8 F7. Administrar Cuaderno de Campo - Casos de Uso

Tabla 2-7 F7. Administrar Cuaderno de Campo

F7. Administrar Cuaderno de Campo		
Descripción	Los actores, una vez que ingresen al sistema tendrán acceso a la Administración de Cuadernos de Campo y a todas las funciones que esta Administración ofrece.	
Flujo Básico	Actor	Sistema
	Escoge la opción Administrar Cuadernos de Campo del menú principal.	Muestra en la Pantalla una tabla con la consulta general de los Cuadernos registrados
	F7.1. Ingresar Cuaderno de Campo	
	• Selecciona la opción <i>Ingresar Nuevo</i>	Muestra la pantalla el formulario para el registro del cuaderno.
	• Ingresa los datos del formulario	Valida los datos
	○ Presiona el botón Insertar	Guarda los Datos Ingresados
	F7.2. Modificar Cuaderno de Campo	
	• Selecciona la opción <i>Editar</i> del cuaderno que desea modificar e ingresa la información	Muestra los campos en modo edición
	• Pulsa el botón <i>Actualizar</i>	Actualiza la información modificada.
	F7.3. Eliminar Cuaderno de Campo	
	• Pulsa el botón <i>Eliminar</i> correspondiente al cuaderno de campo que desea borrar	El sistema elimina la información de la base de datos.
	F7.4. Consultar Cuaderno de Campo	
	• Ingresa la información del cuaderno que desea buscar	El sistema muestra la información del cuaderno
	Flujos Alternos	F7.1. No se registran los datos obligatorios.
	F7.4. No se encontró el cuaderno de campo buscado	Muestra una tabla vacía

Autores: Pablo Almeida & Vanessa Soria C.

2.3.2. Diagramas de Secuencia

2.3.2.1. F1. Administrar Colección

F1.1. Ingresar Colección Zoología

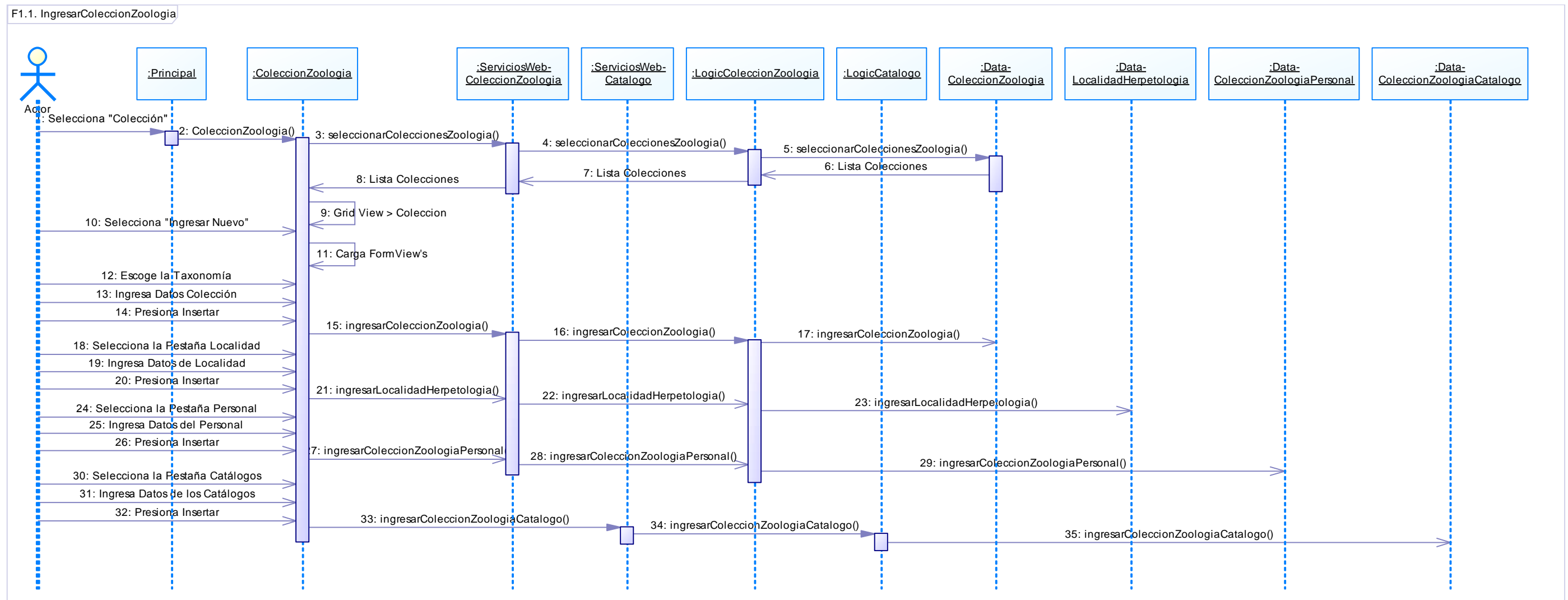


Diagrama 2-9 F1.1. Ingresar Colección Zoología - Secuencia

F1.2. Modificar Colección Zoológica

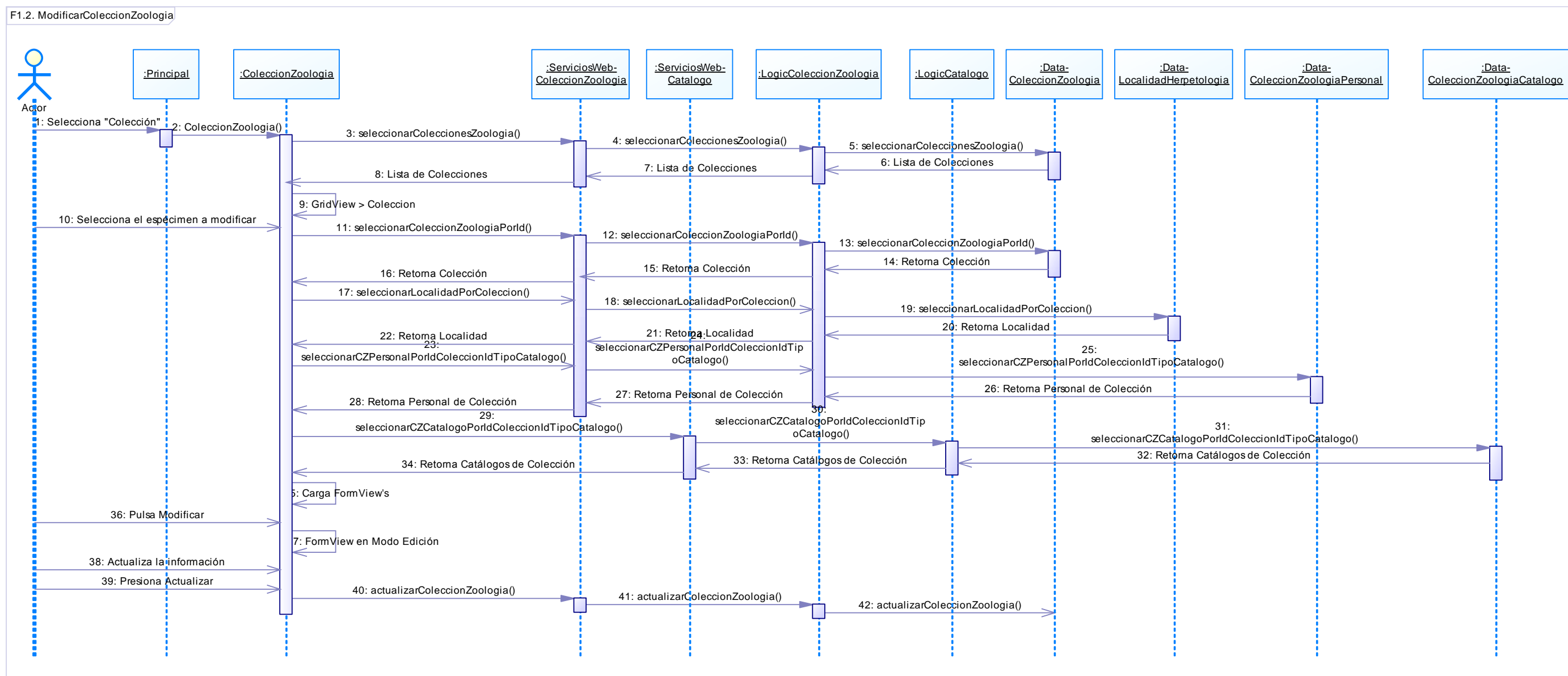


Diagrama 2-10 F1.2. Modificar Colección Zoológica - Secuencia

F1.3. Eliminar Colección Zoológica

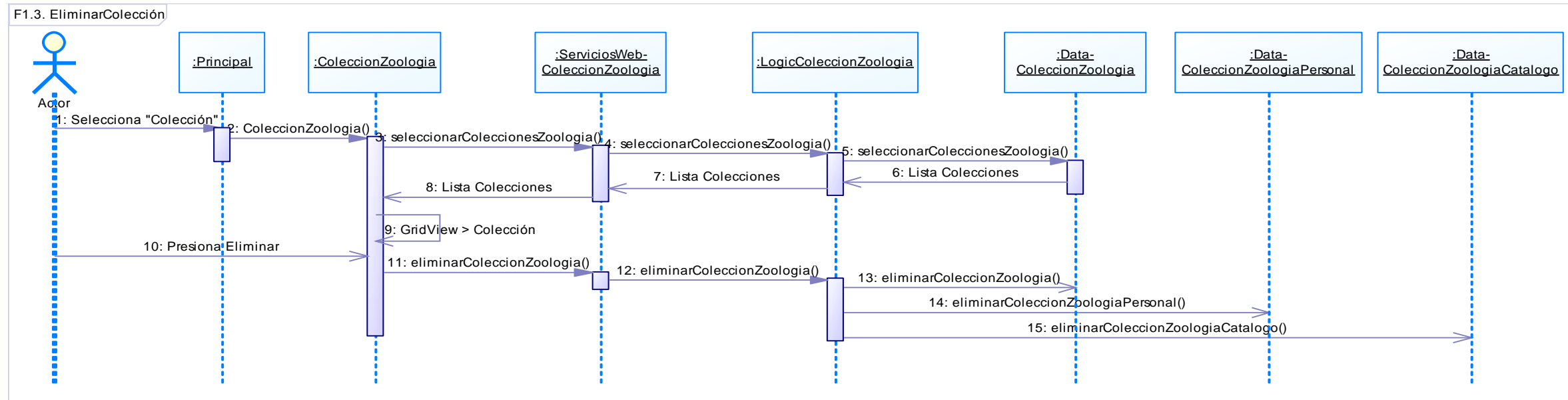


Diagrama 2-11 F1.3. Eliminar Colección Zoológica - Secuencia

F1.4. Consultar Colección Zoológica

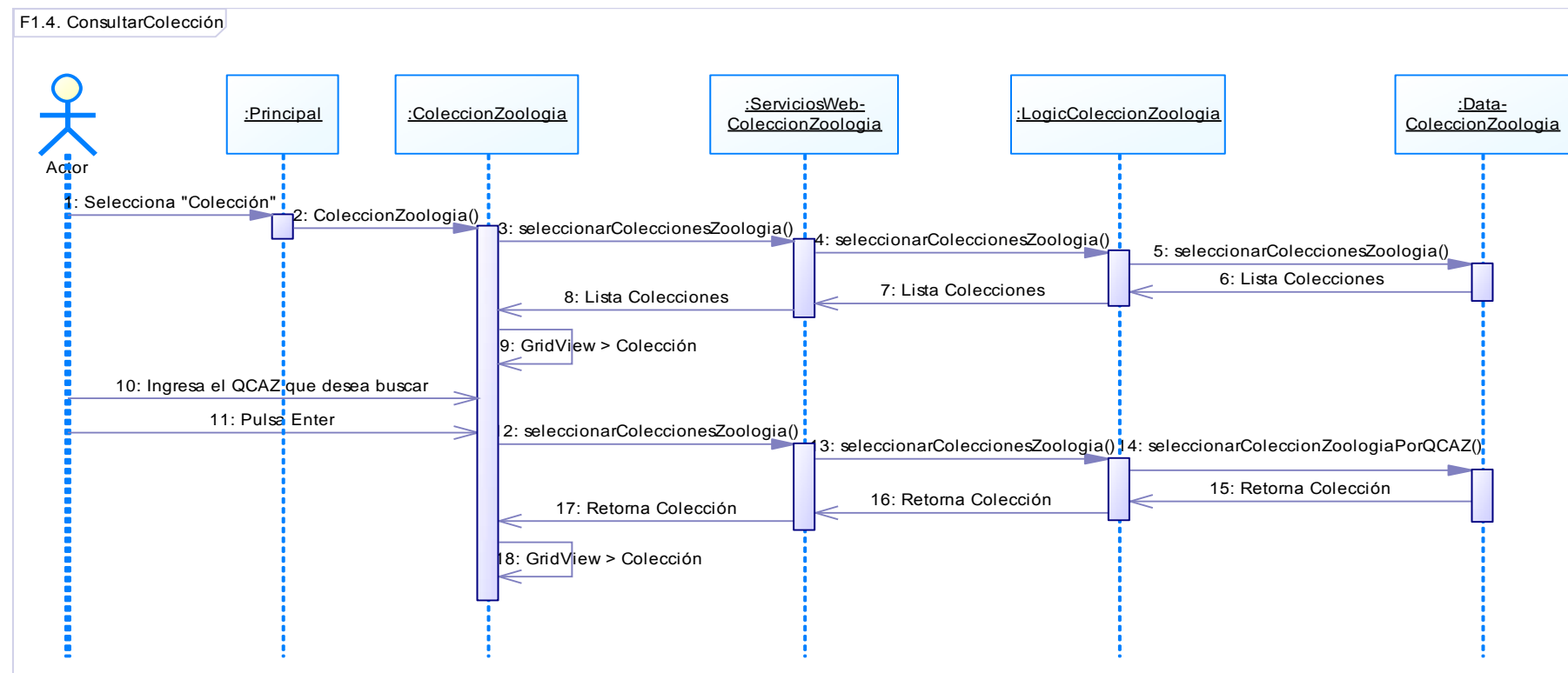


Diagrama 2-12 F1.4. Consultar Colección Zoológica – Secuencia

2.3.2.2. F3. Administrar Préstamos

F3.1. Ingresar Préstamo

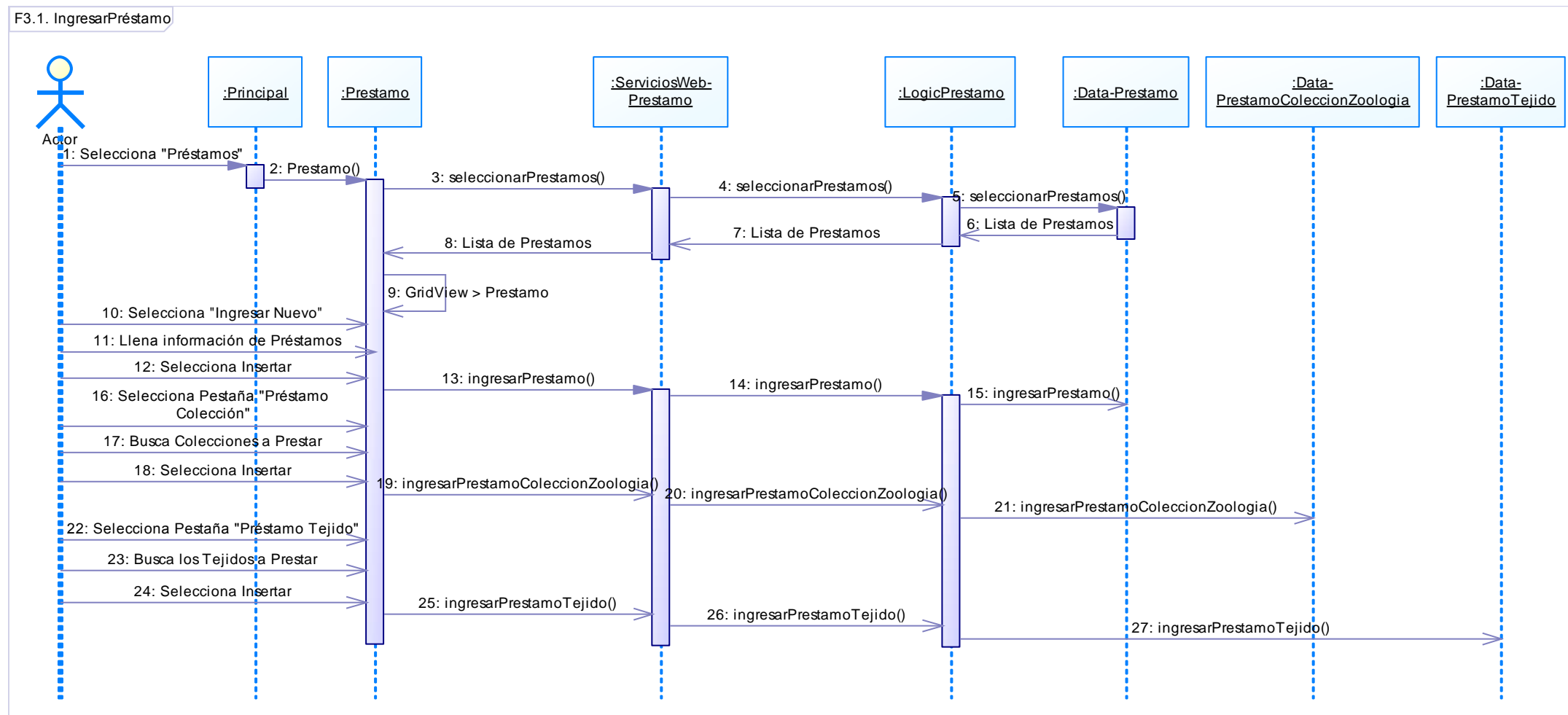


Diagrama 2-13 F3.1. Ingresar Préstamo - Secuencia

F3.2. Modificar Préstamo

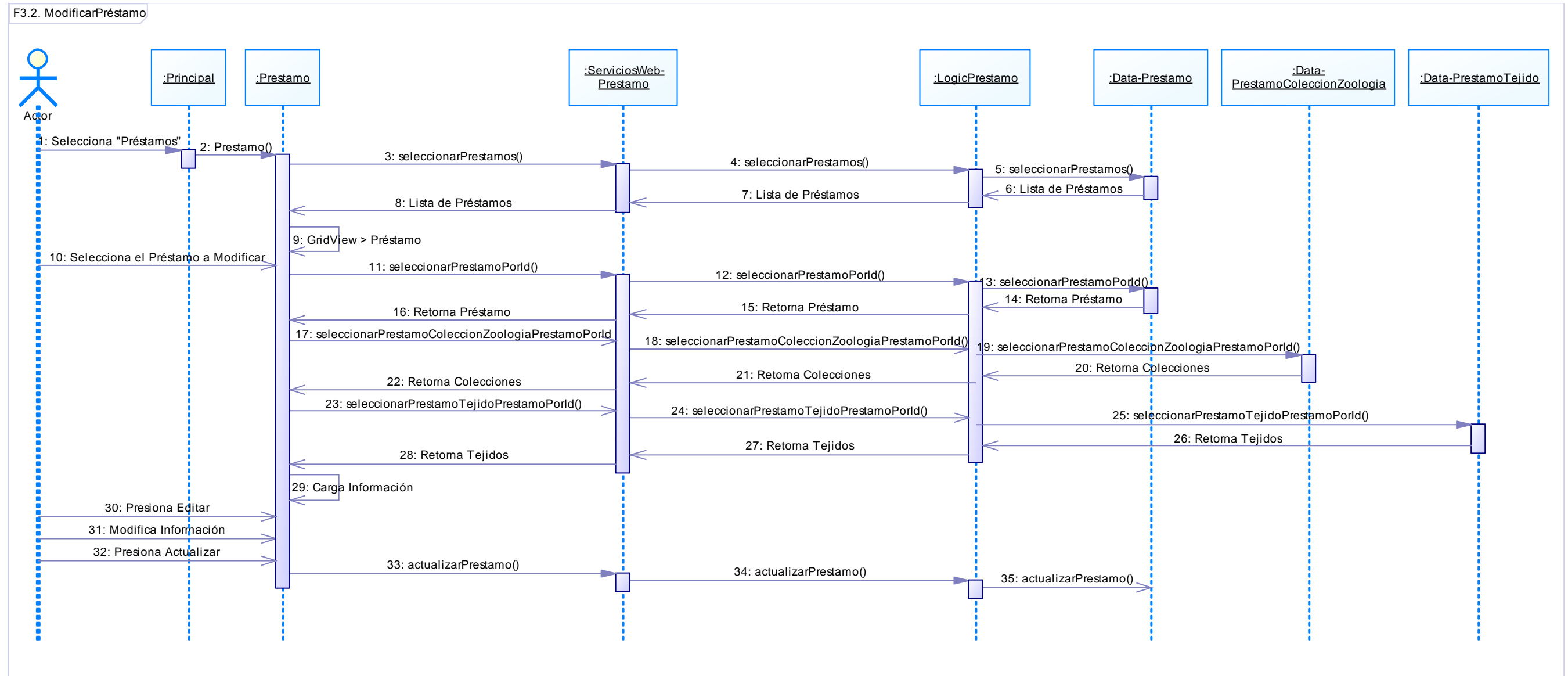


Diagrama 2-14 F3.2. Modificar Préstamo - Secuencia

F3.3. Consultar Préstamo

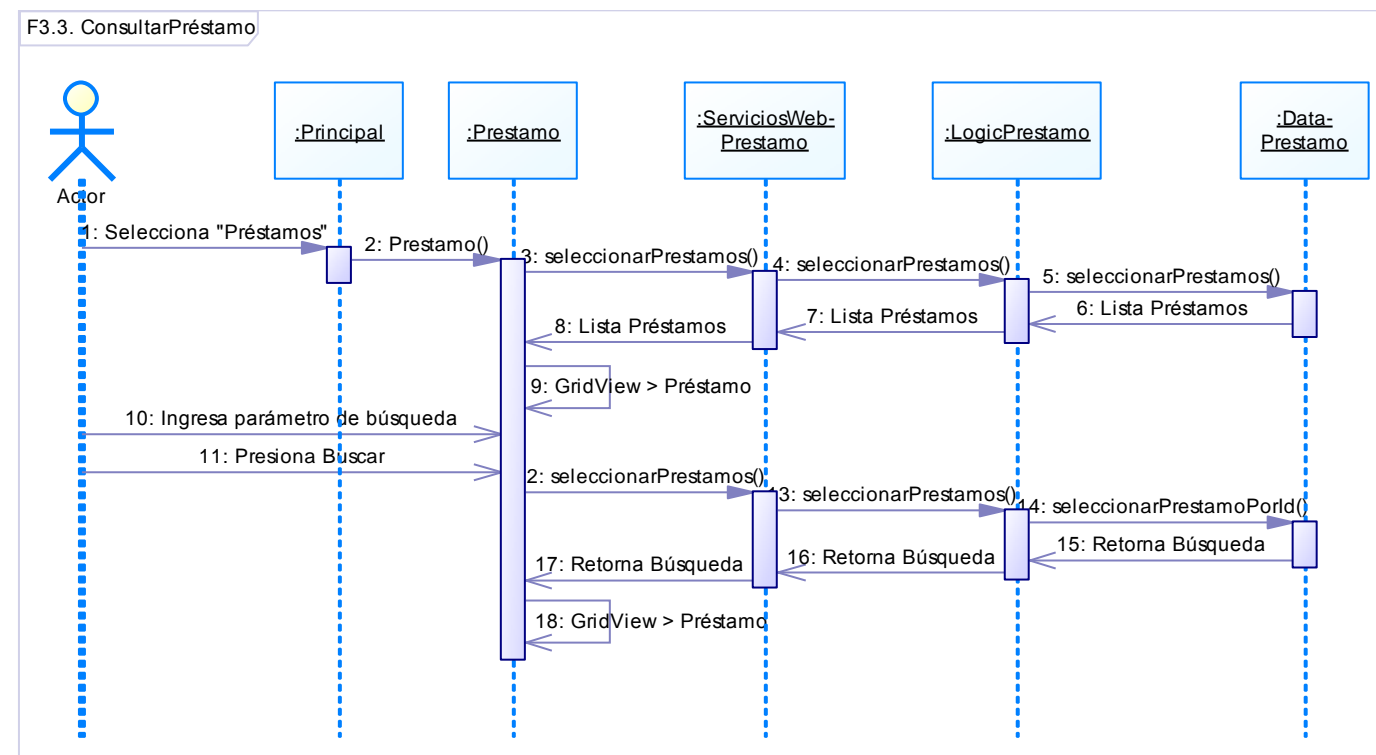


Diagrama 2-15 F3.3. Consultar Préstamo - Secuencia

F3.4. Eliminar Préstamo

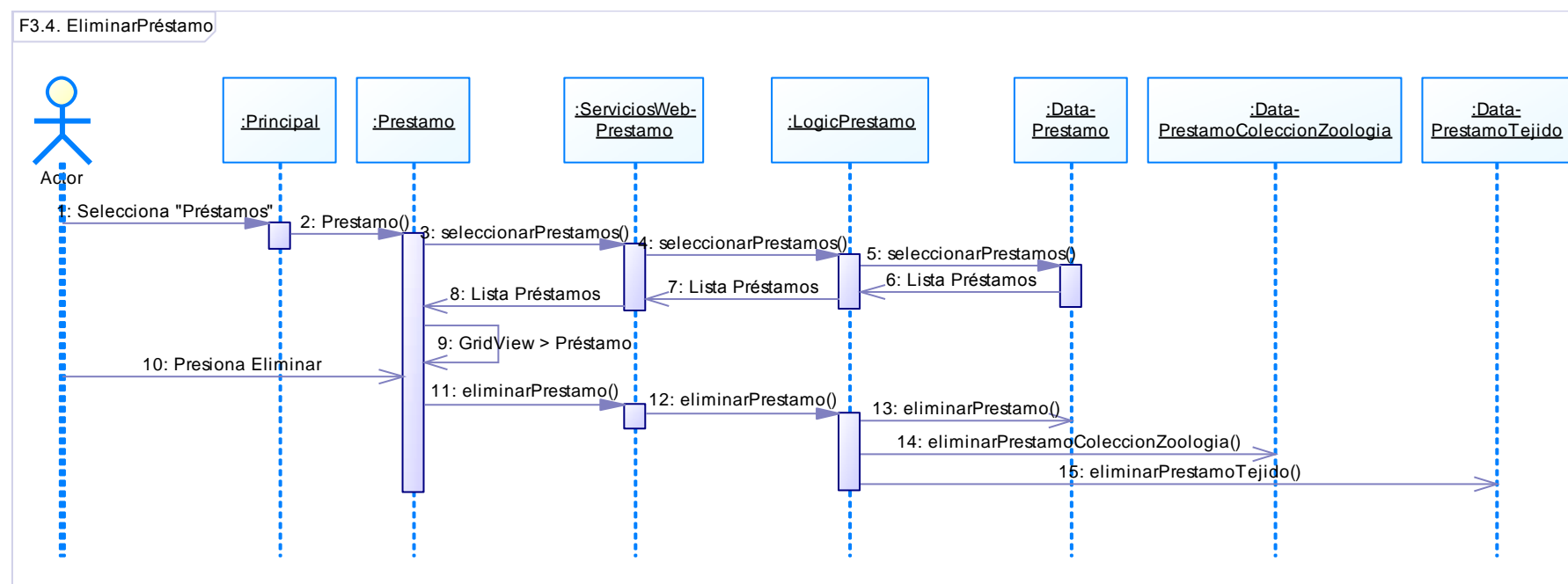


Diagrama 2-16 F3.4. Eliminar Préstamo – Secuencia

2.3.2.3. F4. Administrar Tejido

F4.1. Ingresar Tejido

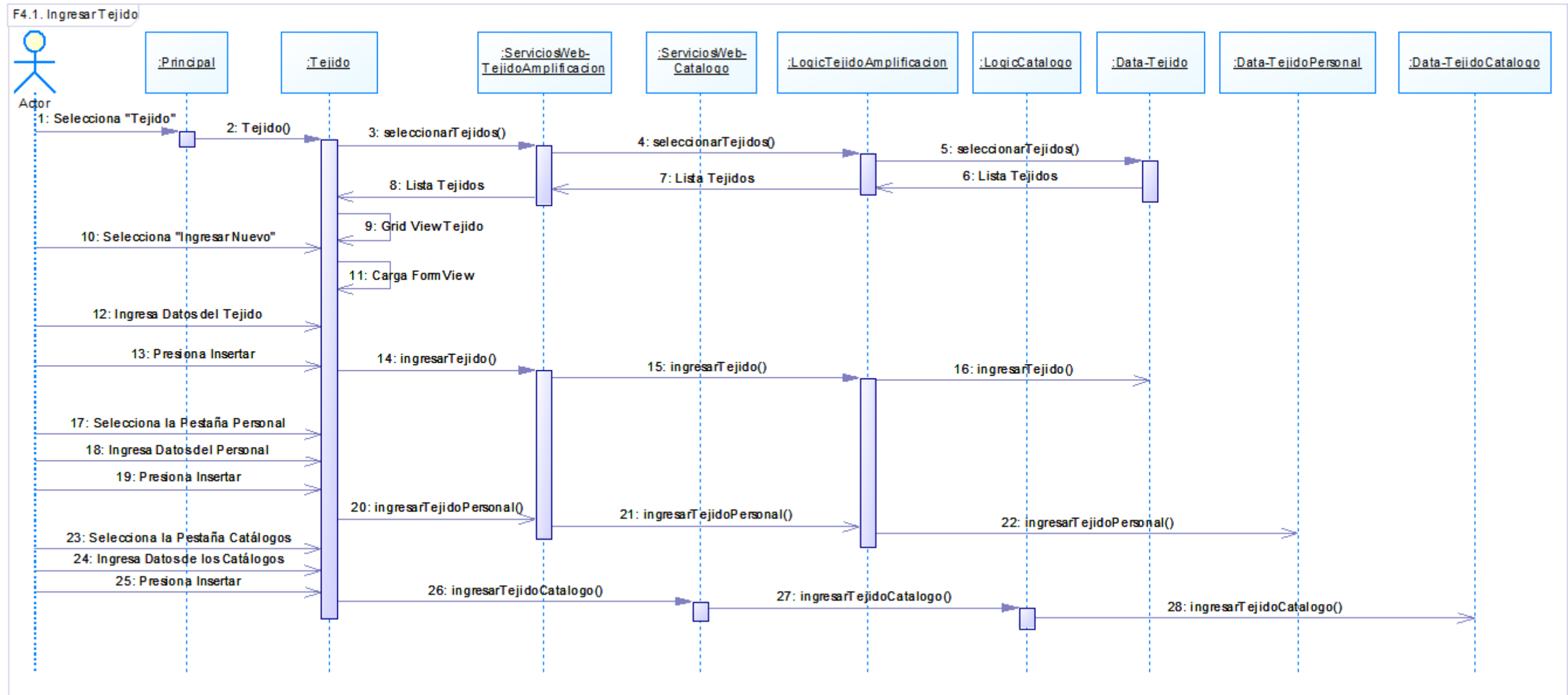


Diagrama 2-17 F4.1. Ingresar Tejido – Secuencia

F4.2. Modificar Tejido

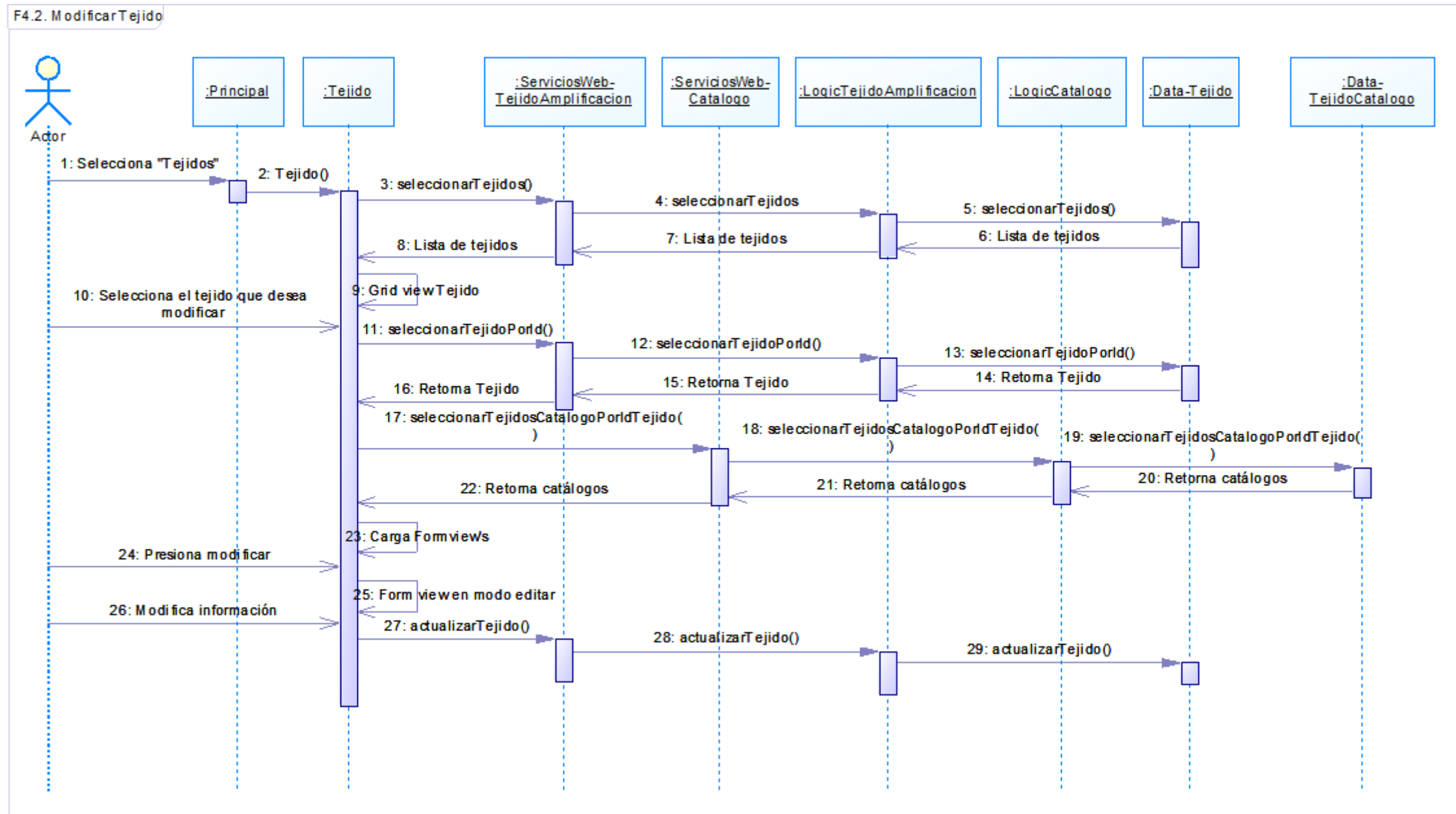


Diagrama 2-18 F4.2. Modificar Tejido - Secuencia

F4.3. Eliminar Tejido

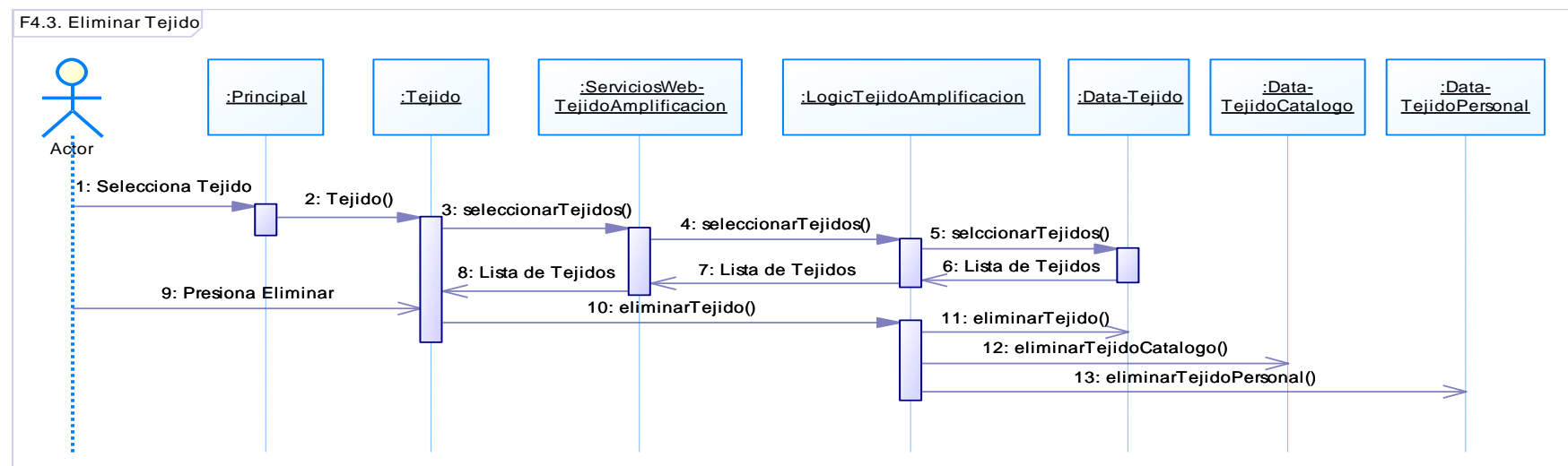


Diagrama 2-19 F4.3. Eliminar Tejido - Secuencia

F4.4. Consultar Tejido

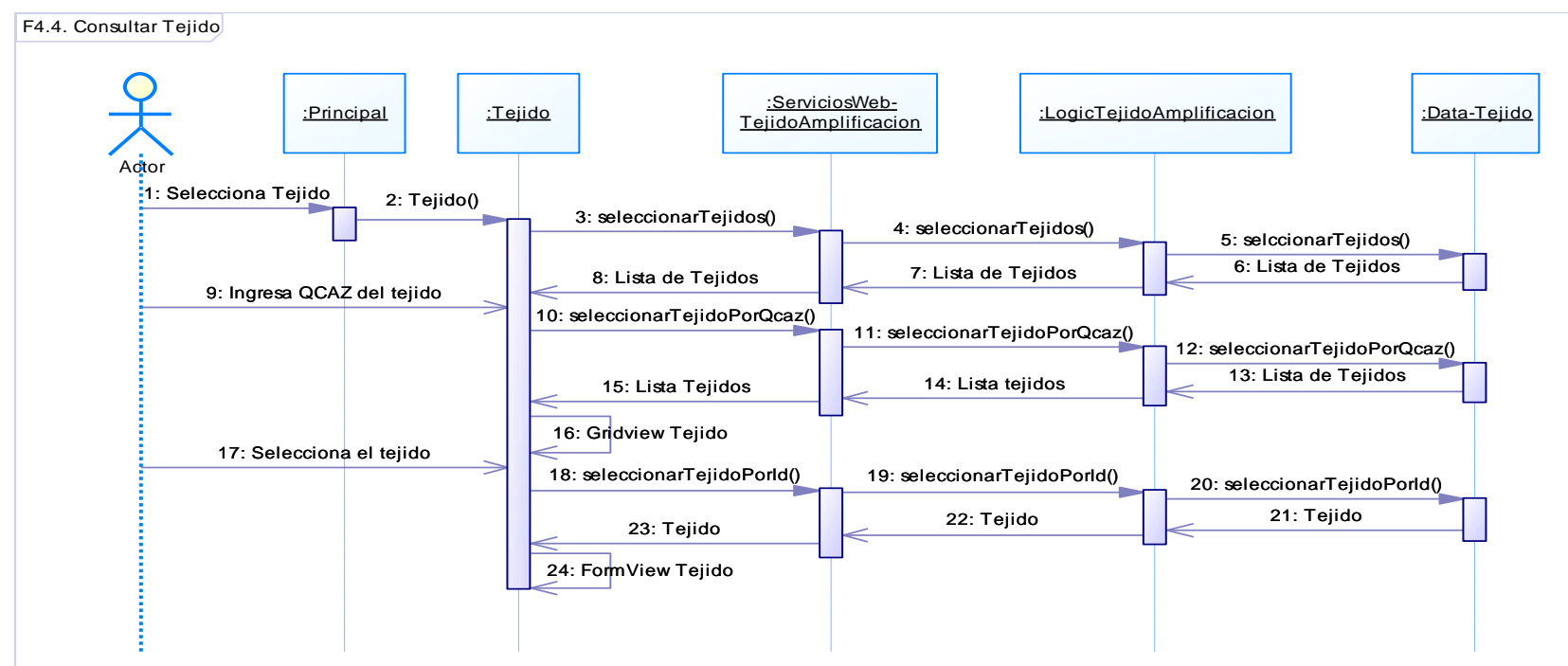


Diagrama 2-20 F4.4. Consultar Tejido - Secuencia

2.3.2.4. F5. Administrar Extracción

F5.1. Ingresar Extracción

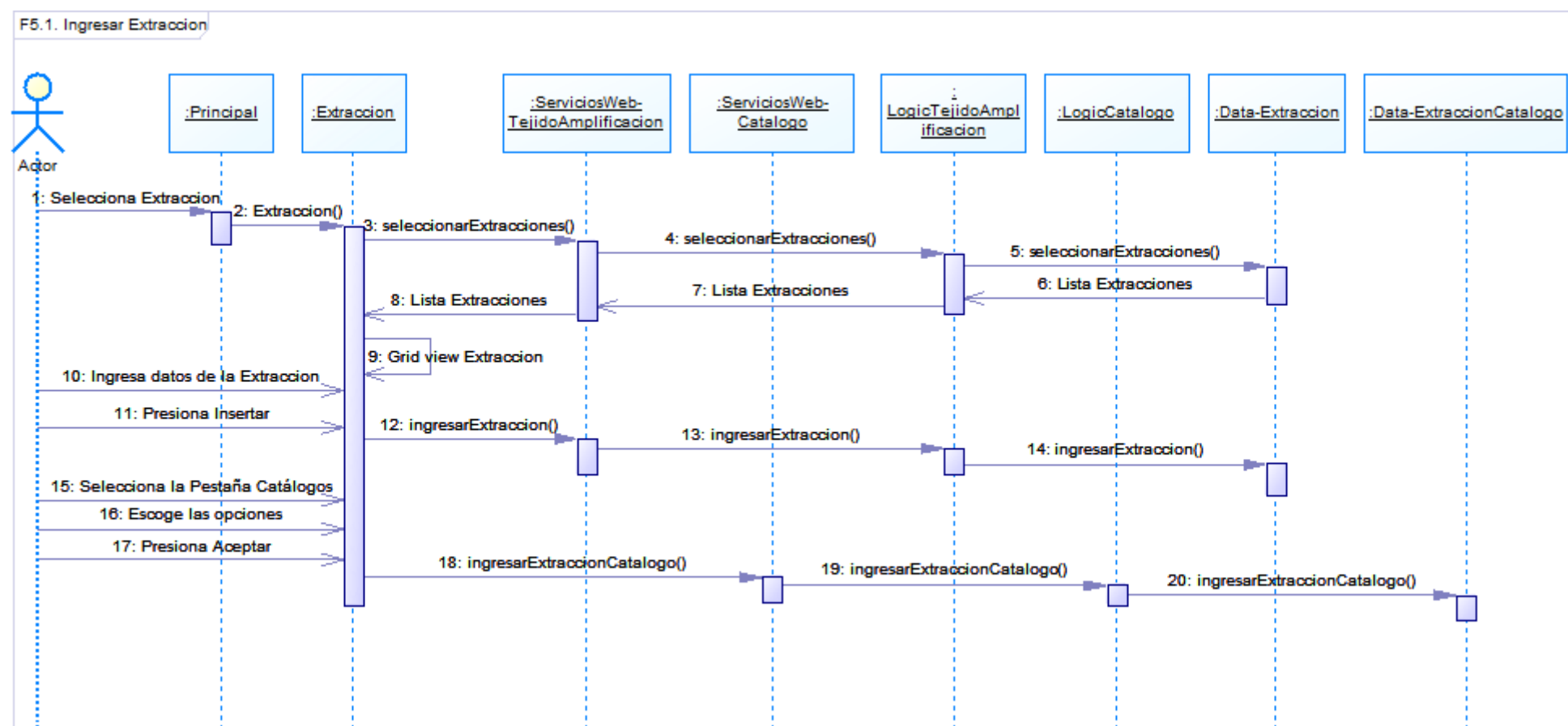


Diagrama 2-21 F5.1. Ingresar Extracción - Secuencia

F5.2. Modificar Extracción

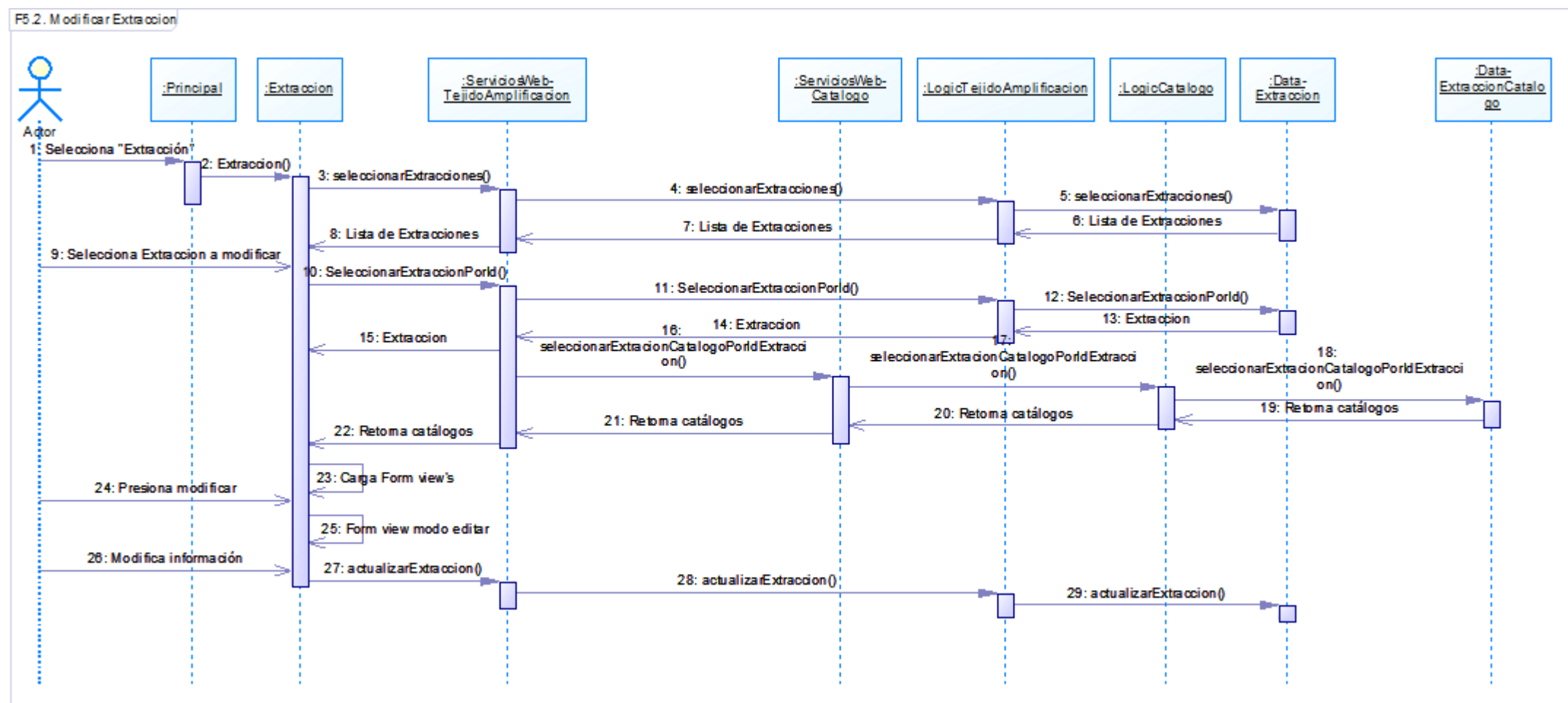


Diagrama 2-22 F5.2. Modificar Extracción - Secuencia

F5.3. Eliminar Extracción

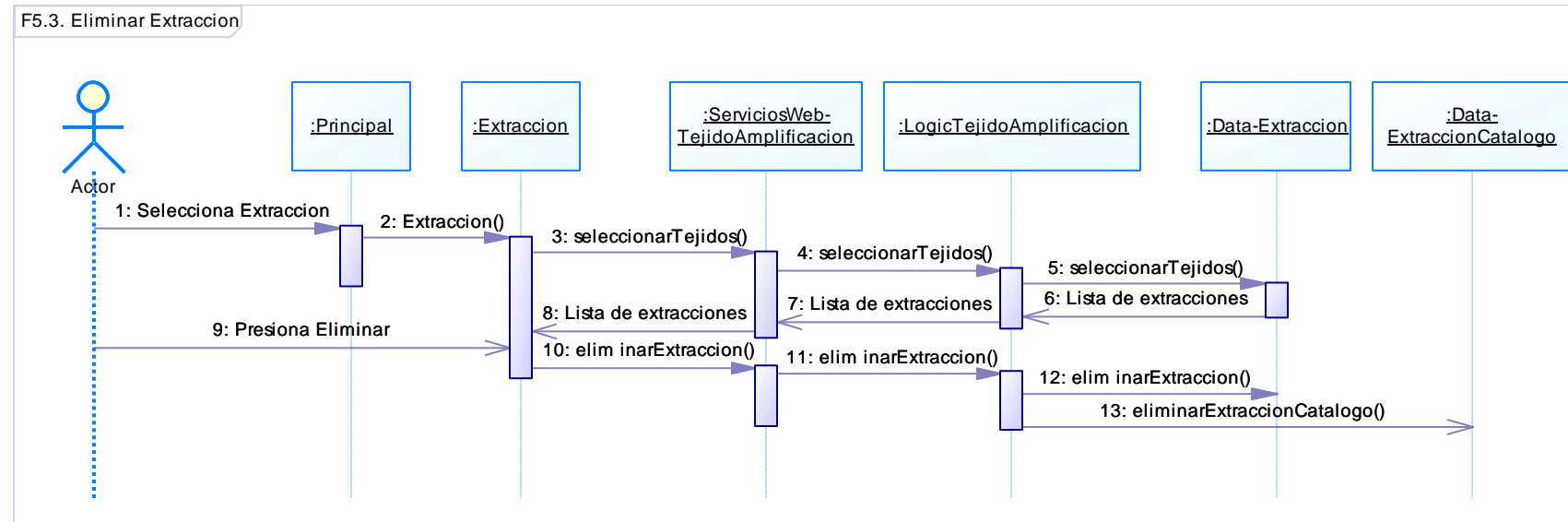


Diagrama 2-23 F5.3. Eliminar Extracción - Secuencia

F5.4. Consultar Extracción

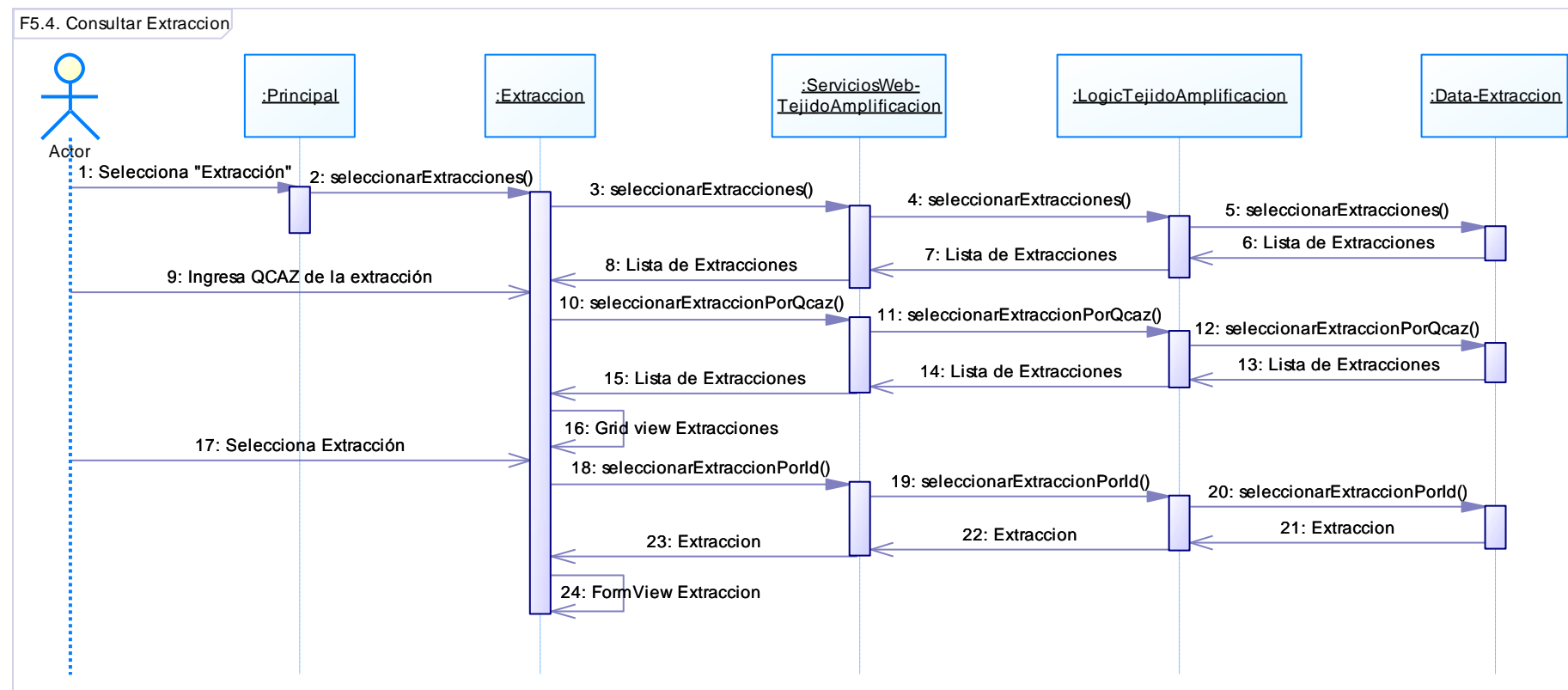


Diagrama 2-24 F5.4. Consultar Extracción - Secuencia

2.3.2.5. F6. Administrar Amplificación

F6.1. Ingresar Amplificación

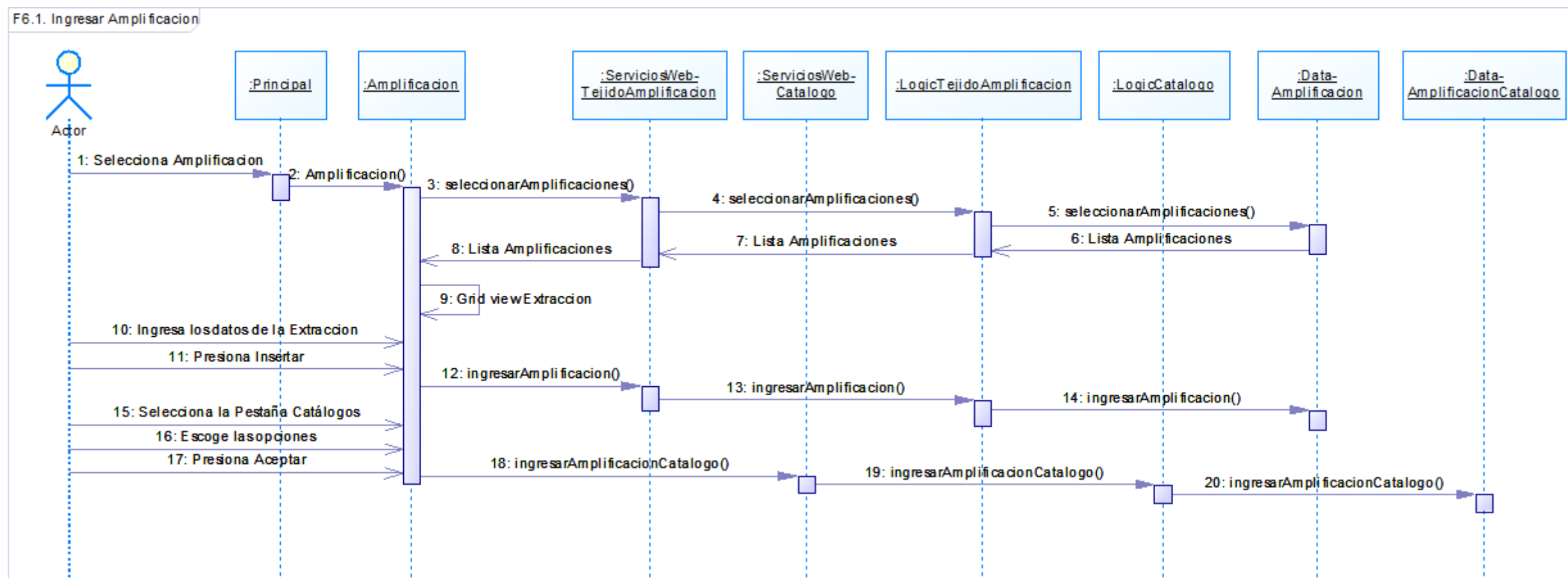


Diagrama 2-25 F6.1. Ingresar Amplificación - Secuencia

F6.2. Modificar Amplificación

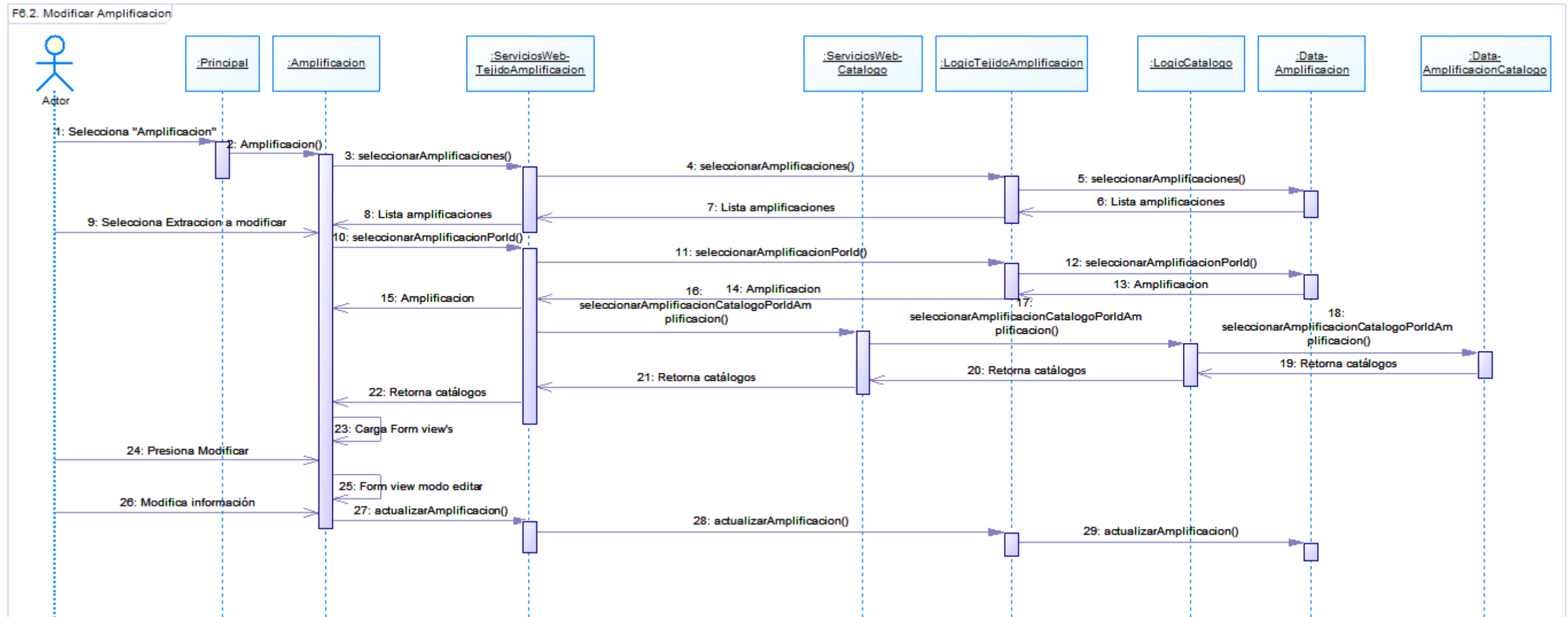


Diagrama 2-26 F6.2. Modificar Amplificación - Secuencia

F6.3. Eliminar Amplificación

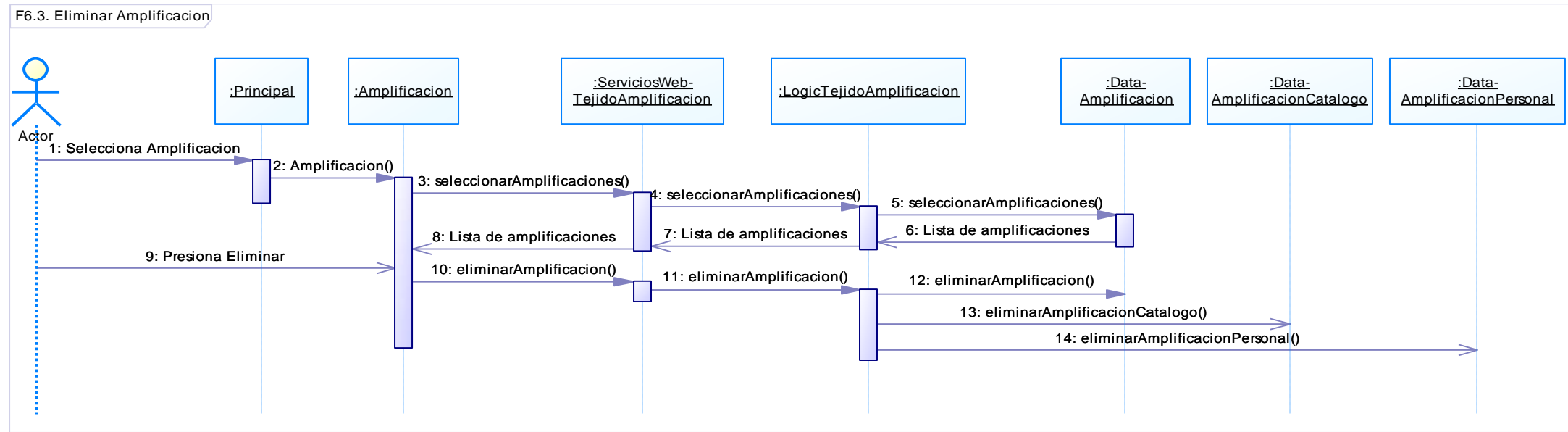


Diagrama 2-27 F6.3. Eliminar Amplificación - Secuencia

F6.4. Consultar Amplificación

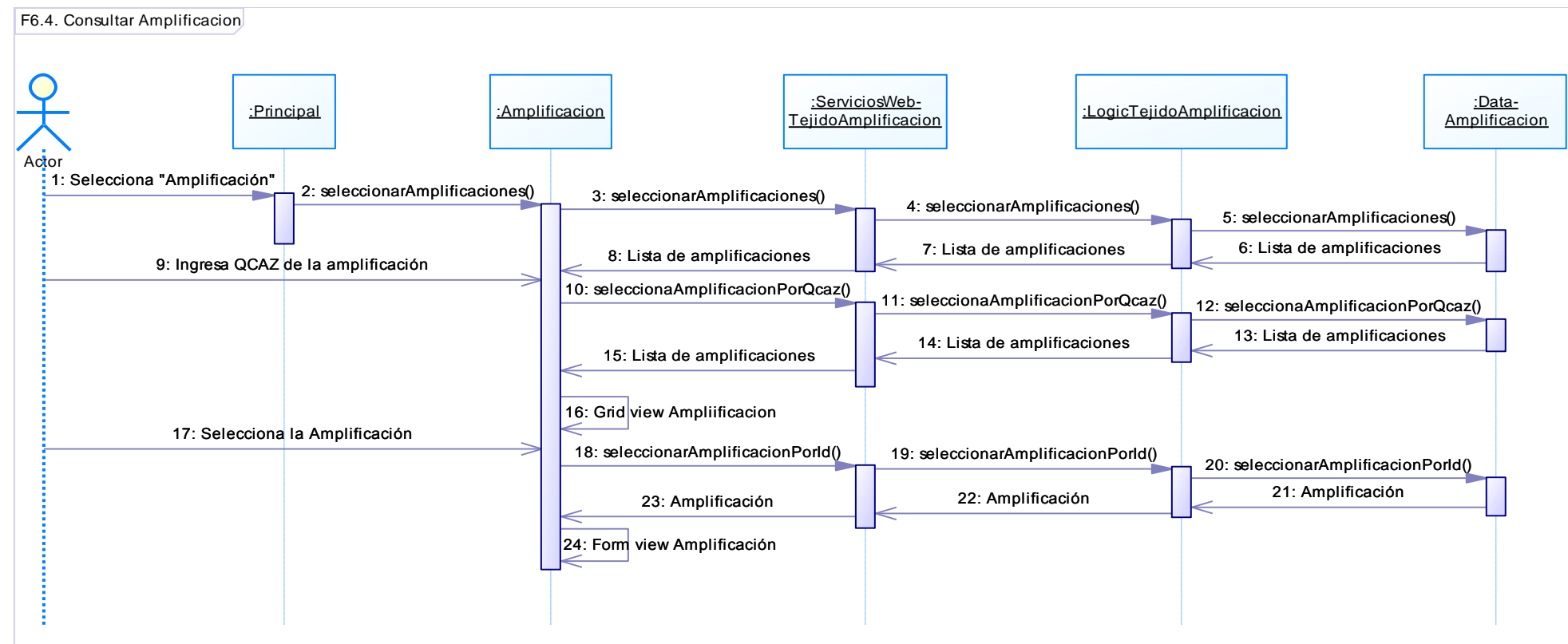


Diagrama 2-28 F6.4. Consultar Amplificación - Secuencia

2.3.2.6. F7. Administrar Cuaderno de Campo

F7.1. Ingresar Cuaderno de Campo

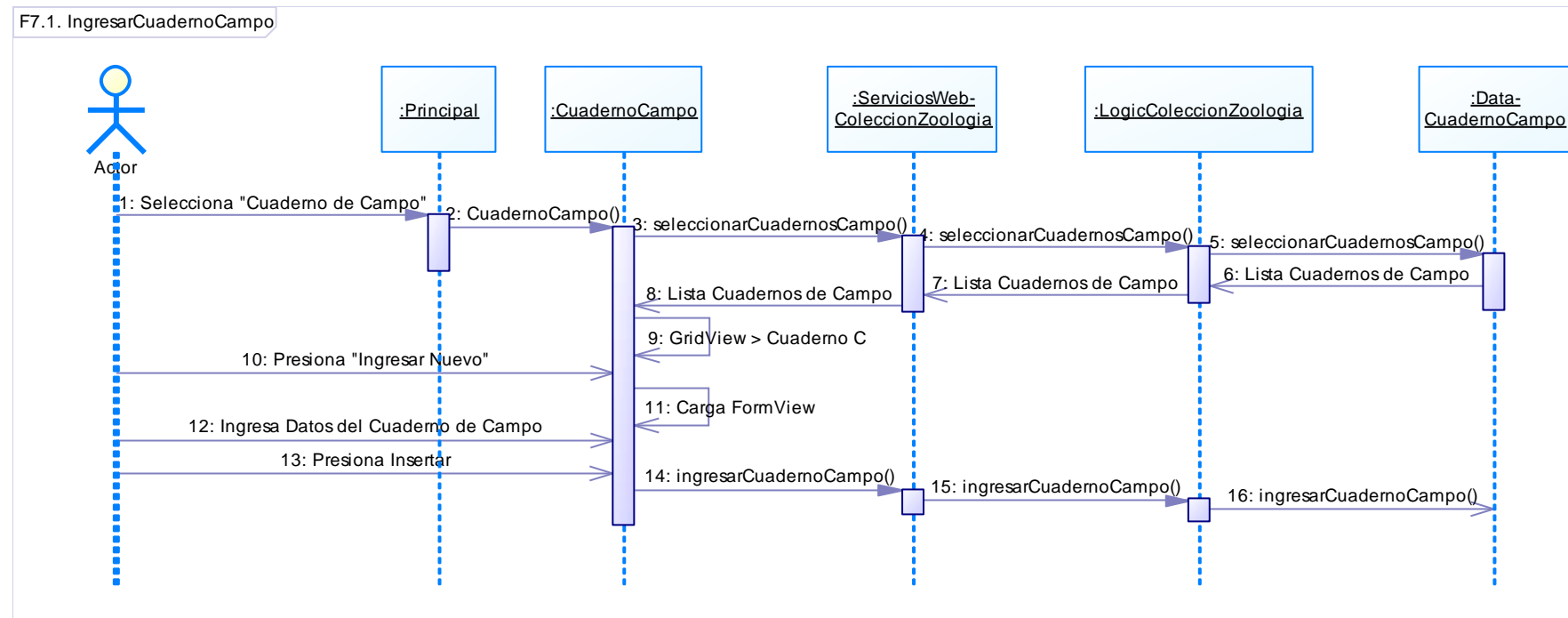


Diagrama 2-29 F7.1. Ingresar Cuaderno de Campo - Secuencia

F7.2. Modificar Cuaderno de Campo

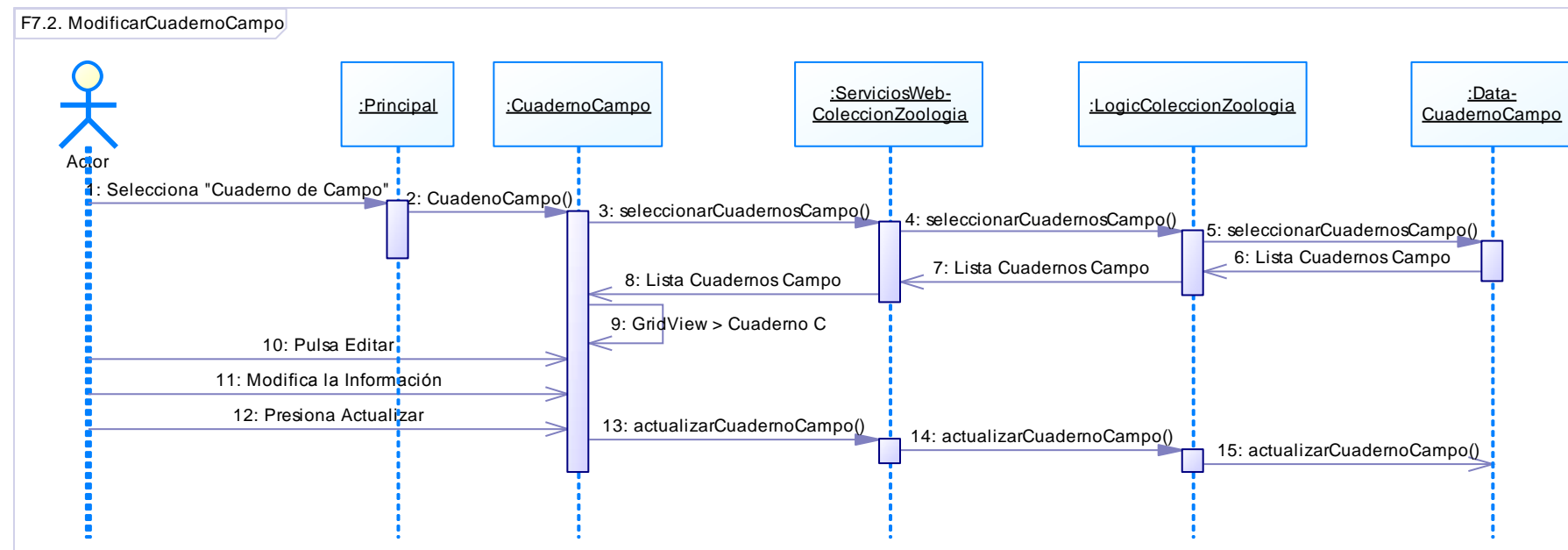


Diagrama 2-30 F7.2. Modificar Cuaderno de Campo - Secuencia

F7.3. Eliminar Cuaderno de Campo

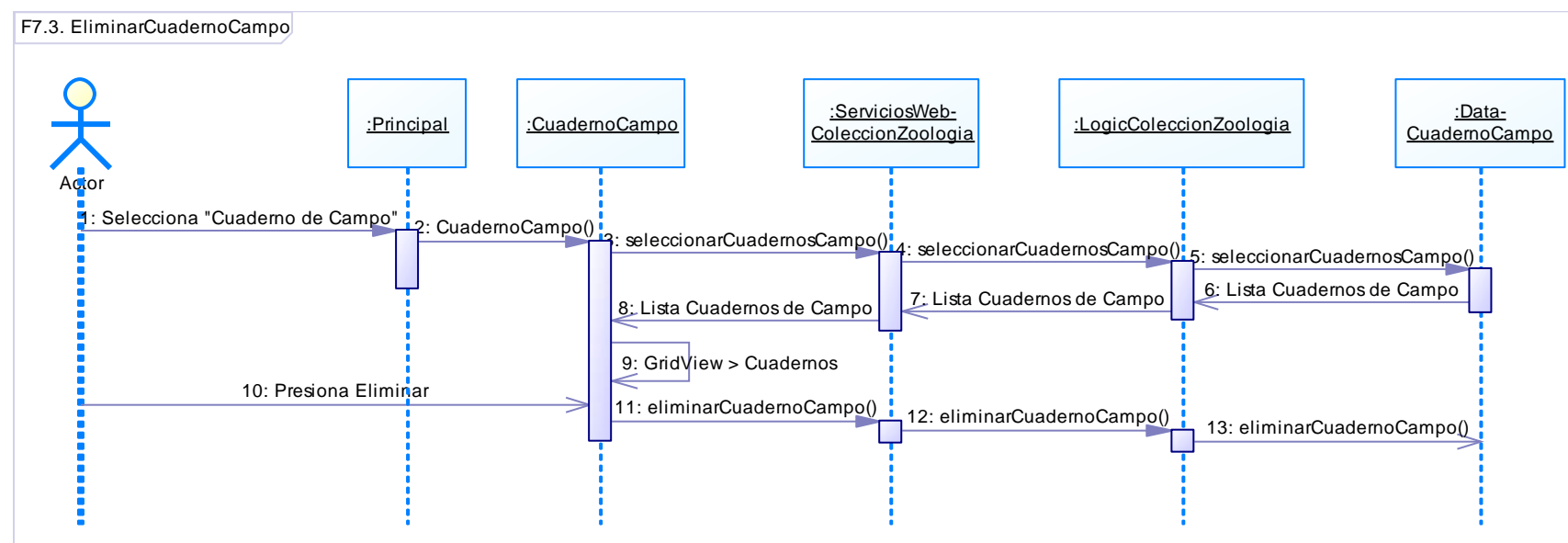


Diagrama 2-31 F7.3. Eliminar Cuaderno de Campo - Secuencia

F7.4. Consultar Cuaderno de Campo

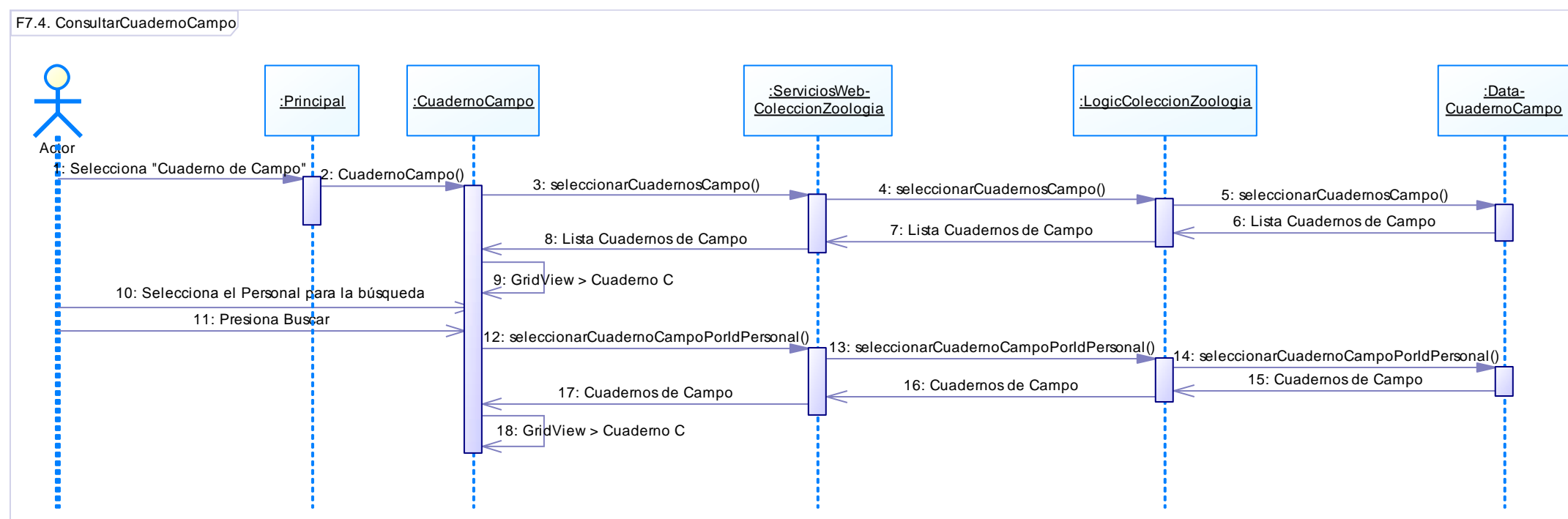


Diagrama 2-32 F7.4. Consultar Cuaderno de Campo - Secuencia

2.3.3. Diagrama de Actividades

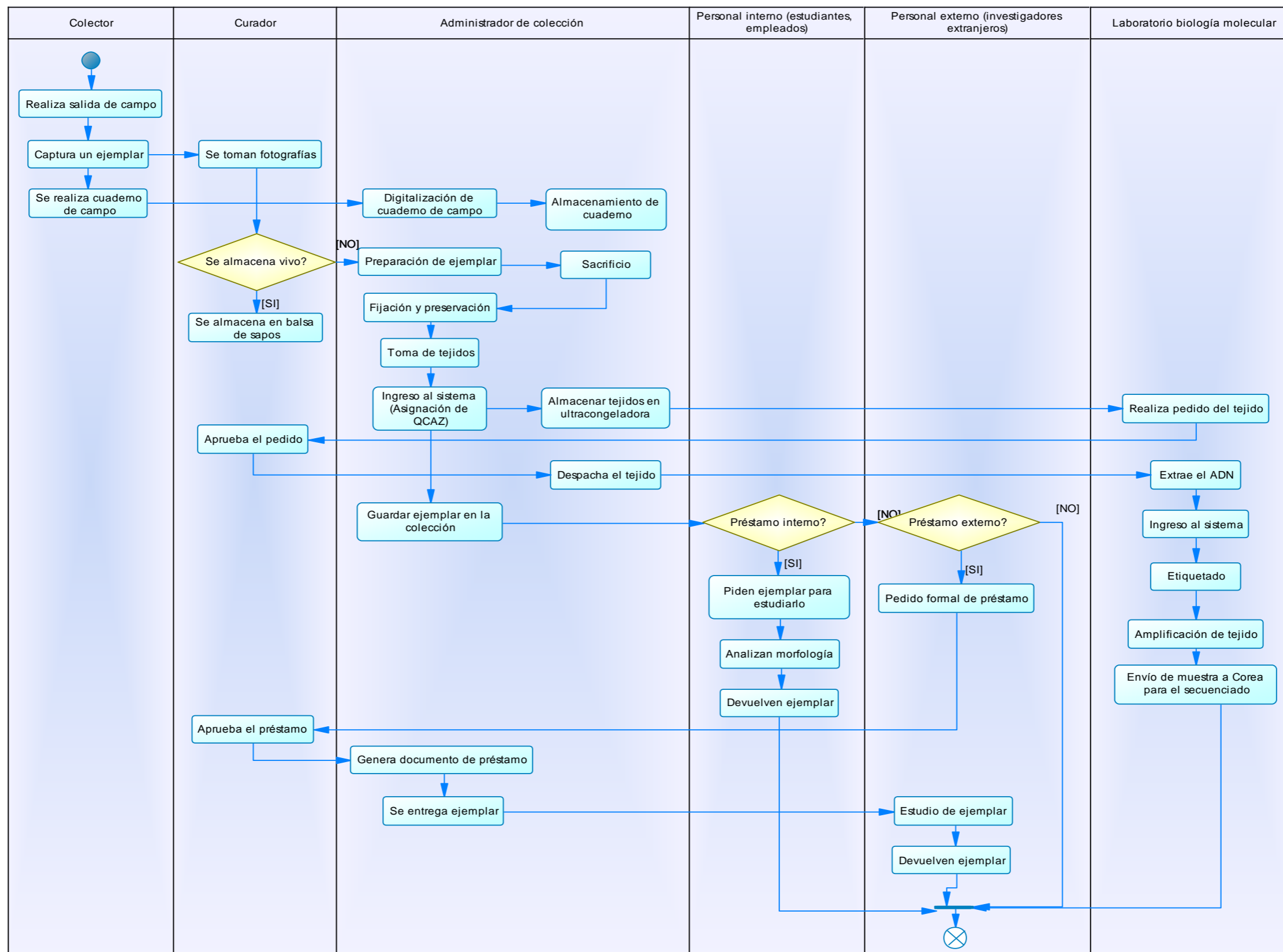


Diagrama 2-33 Actividades

2.3.4. Diagrama de Estereotipos

2.3.4.1. F1. Administrar Colección

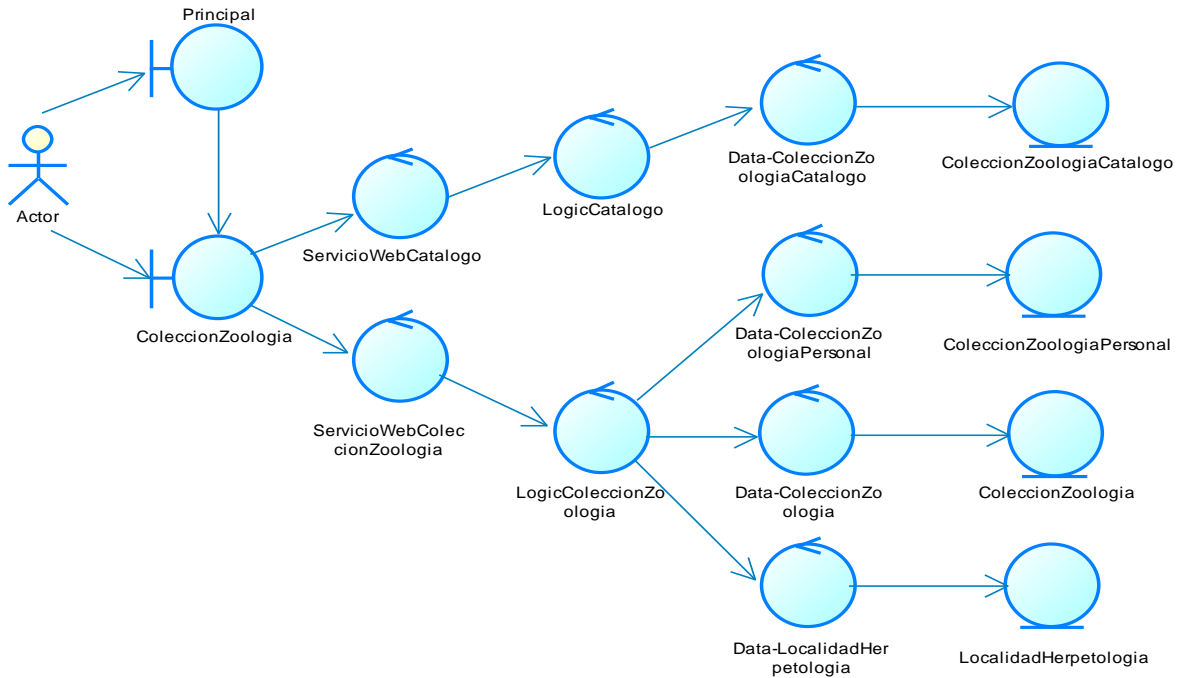


Diagrama 2-34 F1. Administrar Colección - Estereotipos

2.3.4.2. F3. Administrar Préstamos

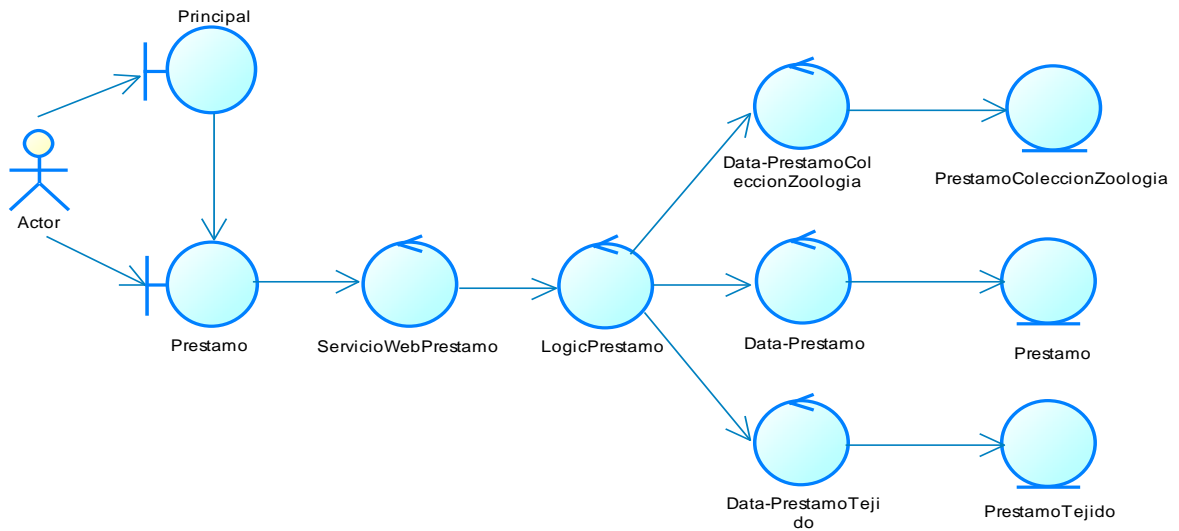


Diagrama 2-35 F3. Administrar Préstamos - Estereotipos

2.3.4.3. F4. Administrar Tejido

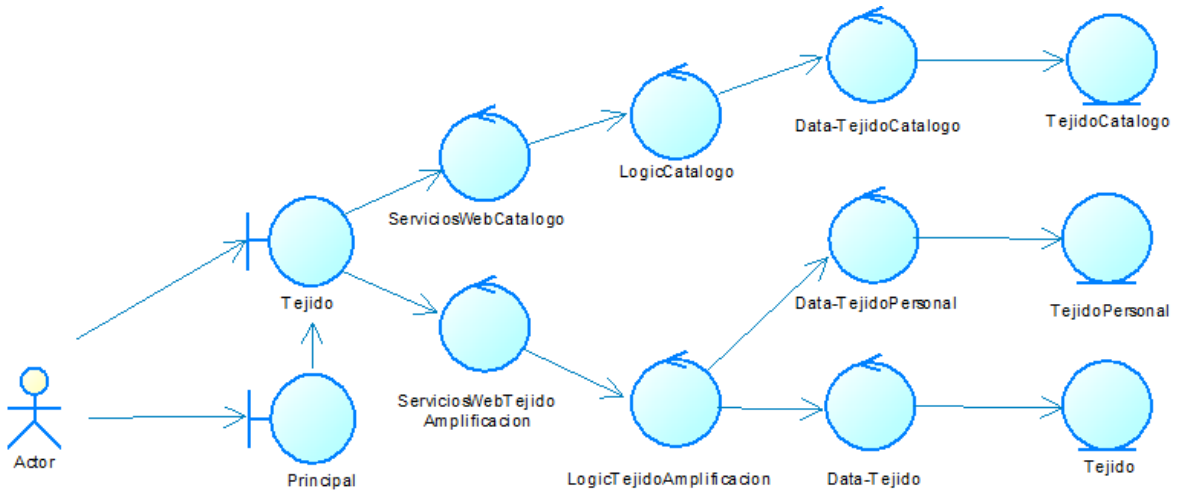


Diagrama 2-36 F4. Administrar Tejido - Estereotipos

2.3.4.4. F5. Administrar Extracción

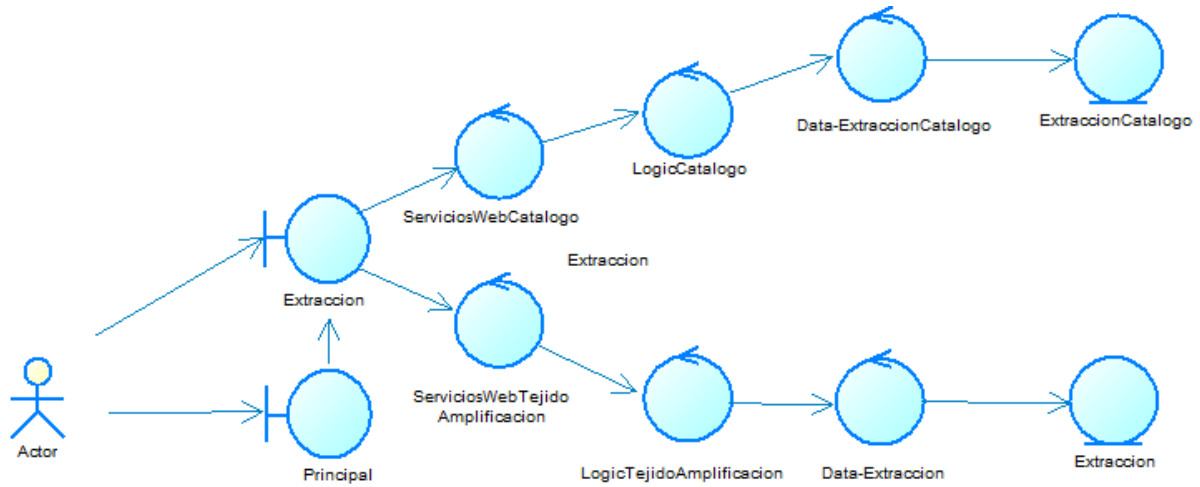


Diagrama 2-37 F5. Administrar Extracción - Estereotipos

2.3.4.5. F6. Administrar Amplificación

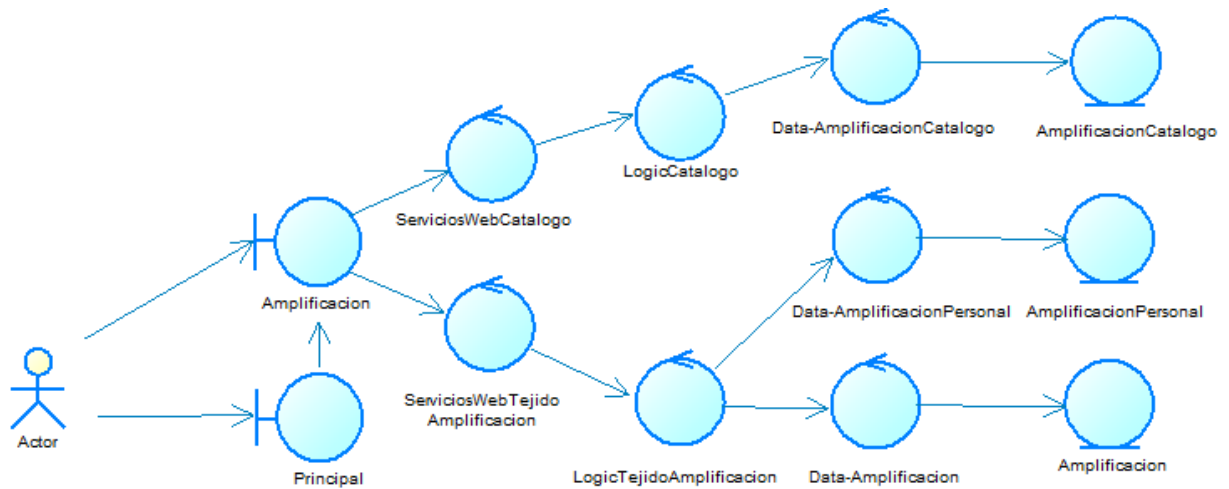


Diagrama 2-38 F6. Administrar Amplificación - Estereotipos

2.3.4.6. F7. Administrar Cuaderno de Campo

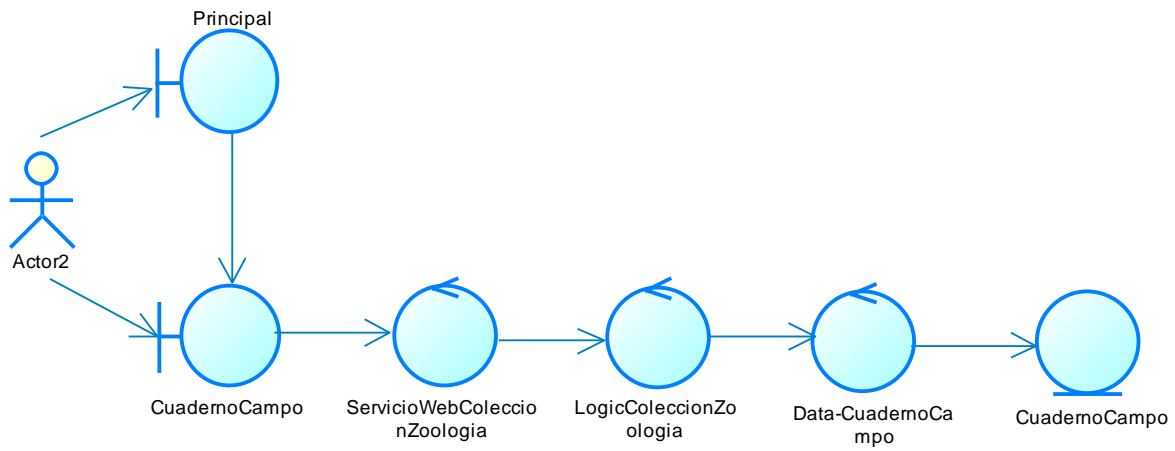


Diagrama 2-39 F7. Administrar Cuaderno de Campo - Estereotipo

3. CAPÍTULO 3: DISEÑO

3.1. Entorno del Software

Tabla 3-1 Entorno del Software

C#	<ul style="list-style-type: none">• Lenguaje de programación orientado a objetos
Servicios SOAP	<ul style="list-style-type: none">• Basado en XML
ASP. NET	<ul style="list-style-type: none">• Framework para desarrollo de aplicaciones Web
SQL Server Express	<ul style="list-style-type: none">• Motor de Bases de Datos
IIS	<ul style="list-style-type: none">• Servidor de Aplicaciones
Visual Studio	<ul style="list-style-type: none">• IDE de Programación
SQL Server Managment Studio	<ul style="list-style-type: none">• SGBD para SQL Server

Autores: Pablo Almeida & Vanessa Soria C.

3.2. Diagrama de Clases

(Ver Anexo 3)

3.3. Esquema de Datos

3.3.1. Diagrama Conceptual

(Ver Anexo 4)

3.3.2. Diccionario de Datos

Contiene las características correspondientes a los datos de la base, concretamente, una descripción de la estructura de la base, las tablas que la componen, y a su vez, los tipos de datos que componen dichas tablas. *(Ver Anexo 5)*

3.4. Diagrama de Colaboración

3.4.1. F1. Administrar Colección

F1.1. Ingresar Colección Zoología

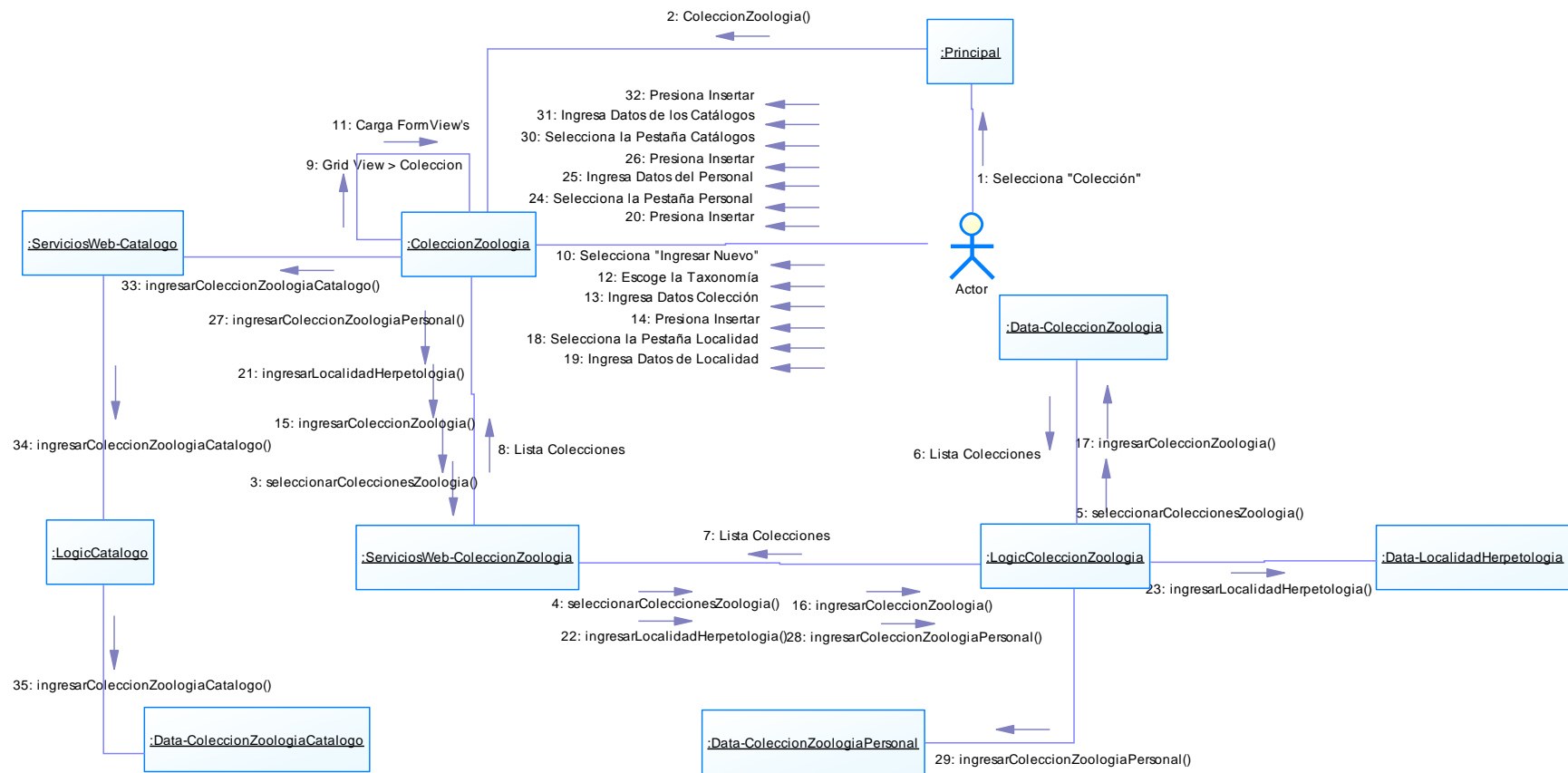


Diagrama 3-1 F1.1. Ingresar Colección Zoología - Colaboración

F1.2. Modificar Colección Zoología

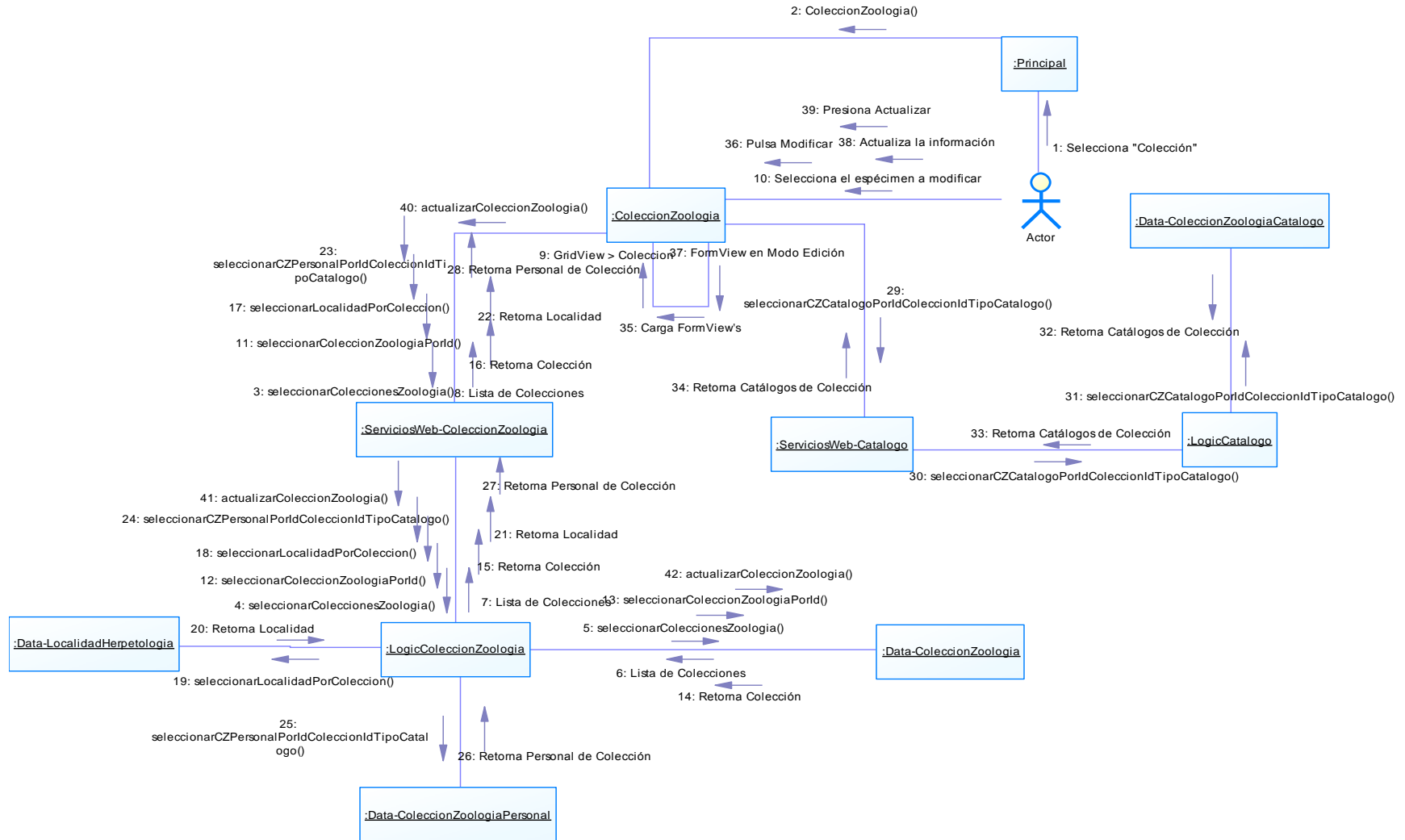


Diagrama 3-2 F1.2. Modificar Colección Zoología - Colaboración

F1.3. Eliminar Colección Zoológica

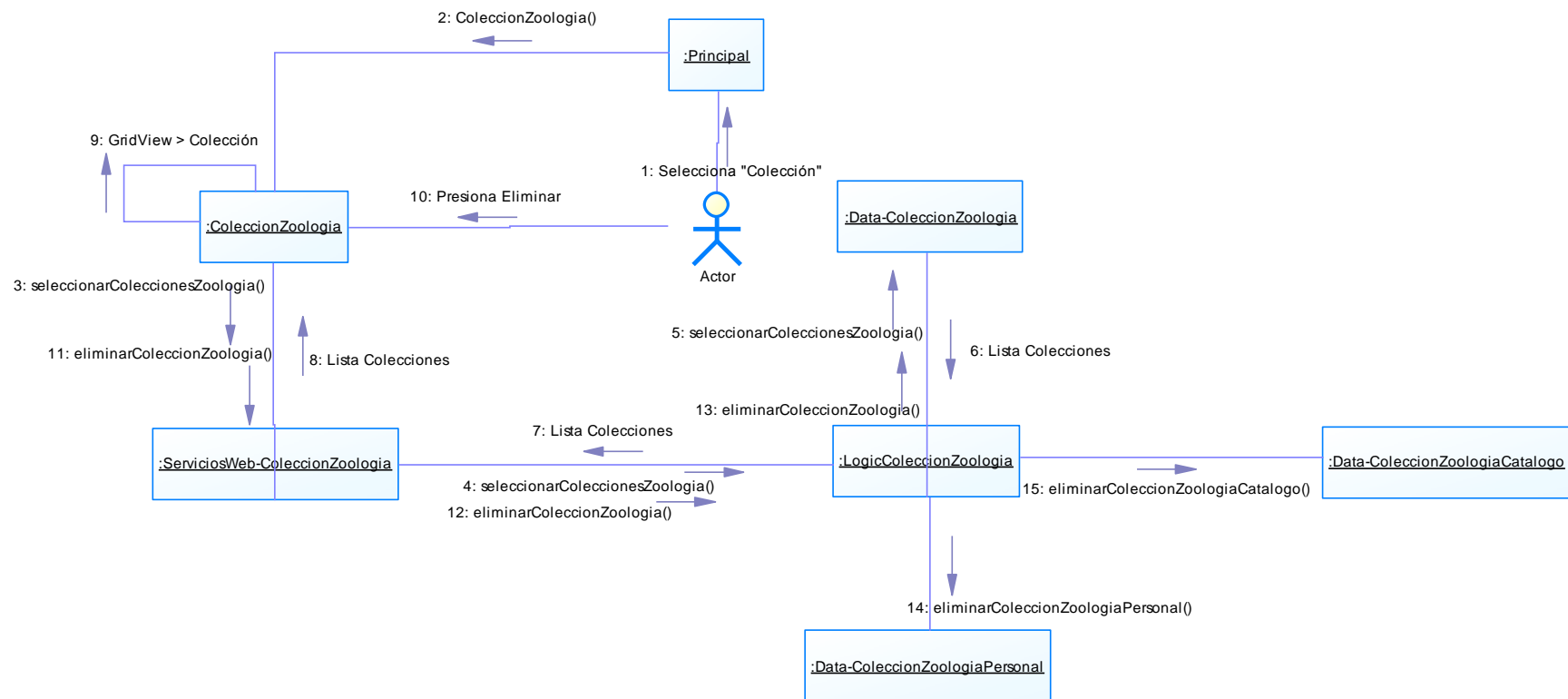


Diagrama 3-3 F1.3. Eliminar Colección Zoológica - Colaboración

F1.4. Consultar Colección Zoológica

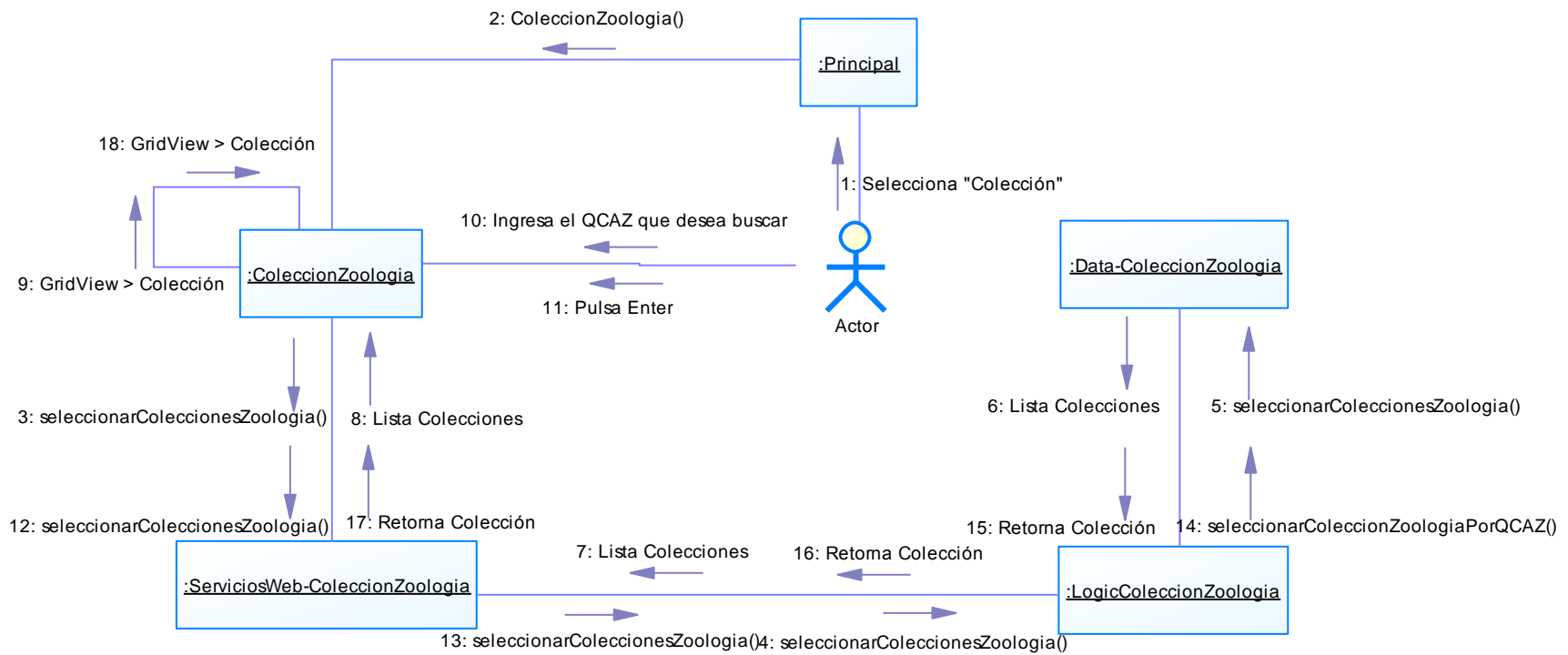


Diagrama 3-4 F1.4. Consultar Colección Zoológica - Colaboración

3.4.2. F3. Administrar Préstamos

F3.1. Ingresar Préstamo

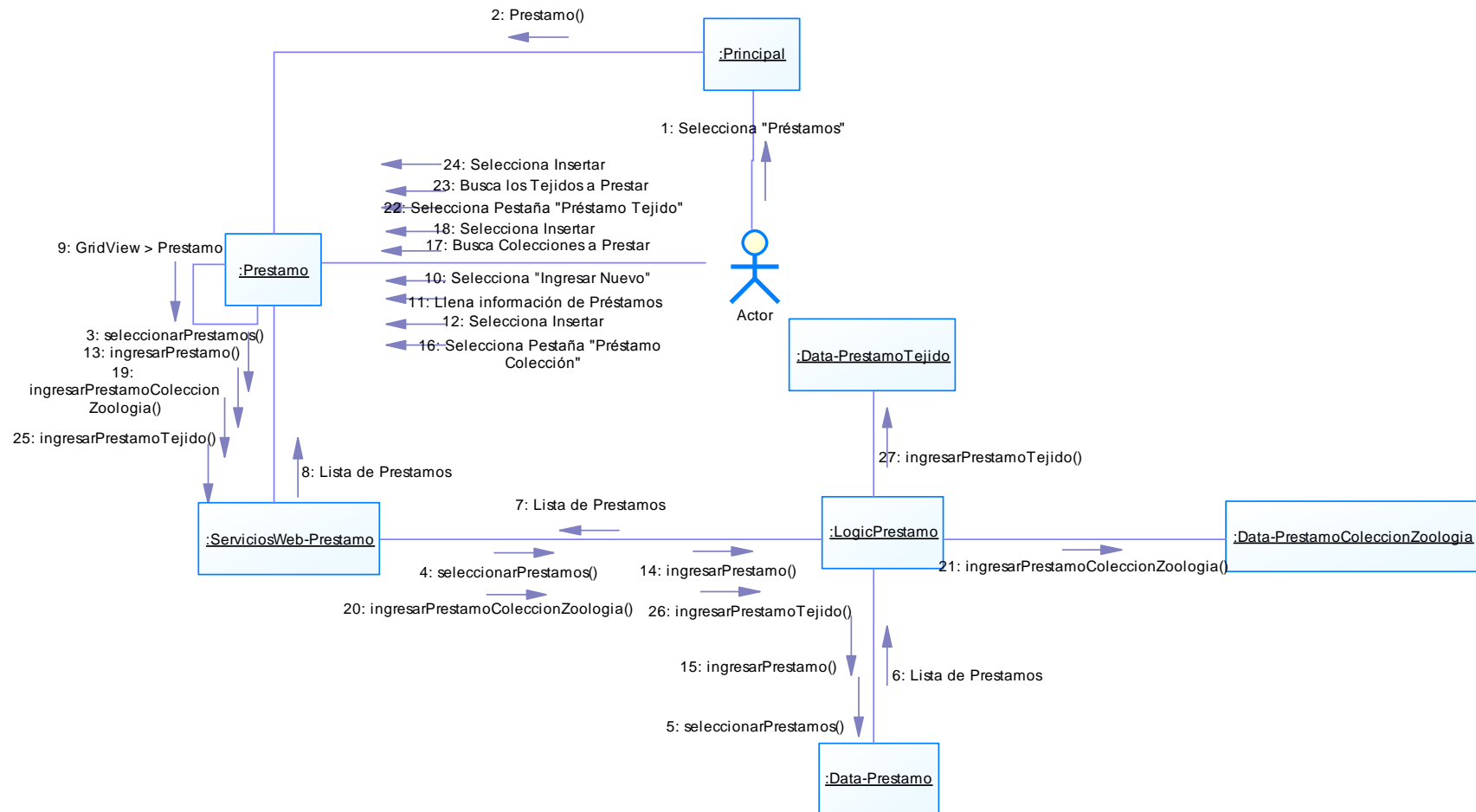


Diagrama 3 5 F3.1. Ingresar Préstamo - Colaboración

F3.3. Modificar Préstamo

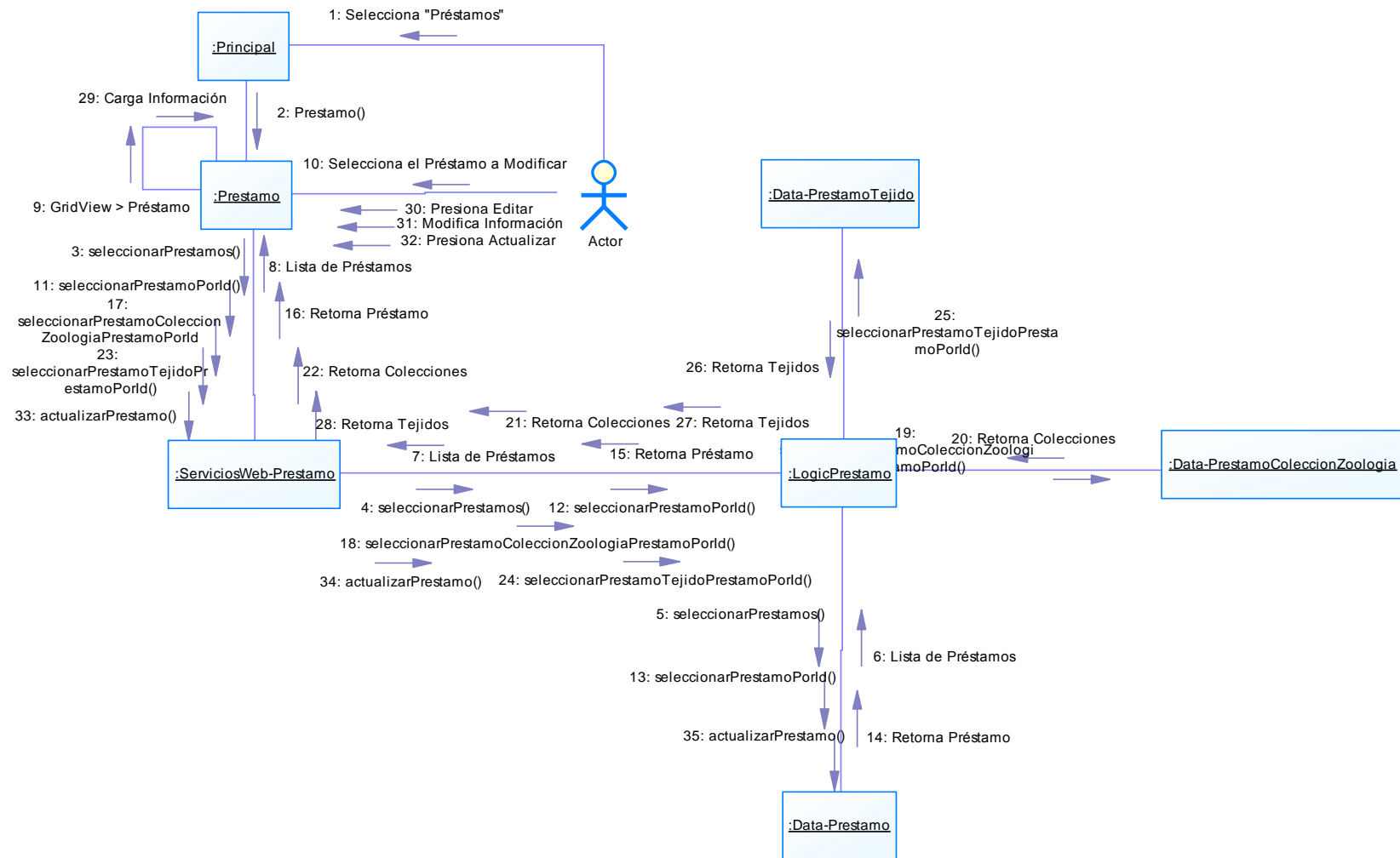


Diagrama 3 7 F3.3. Modificar Préstamo - Colaboración

F3.4. Consultar Préstamo

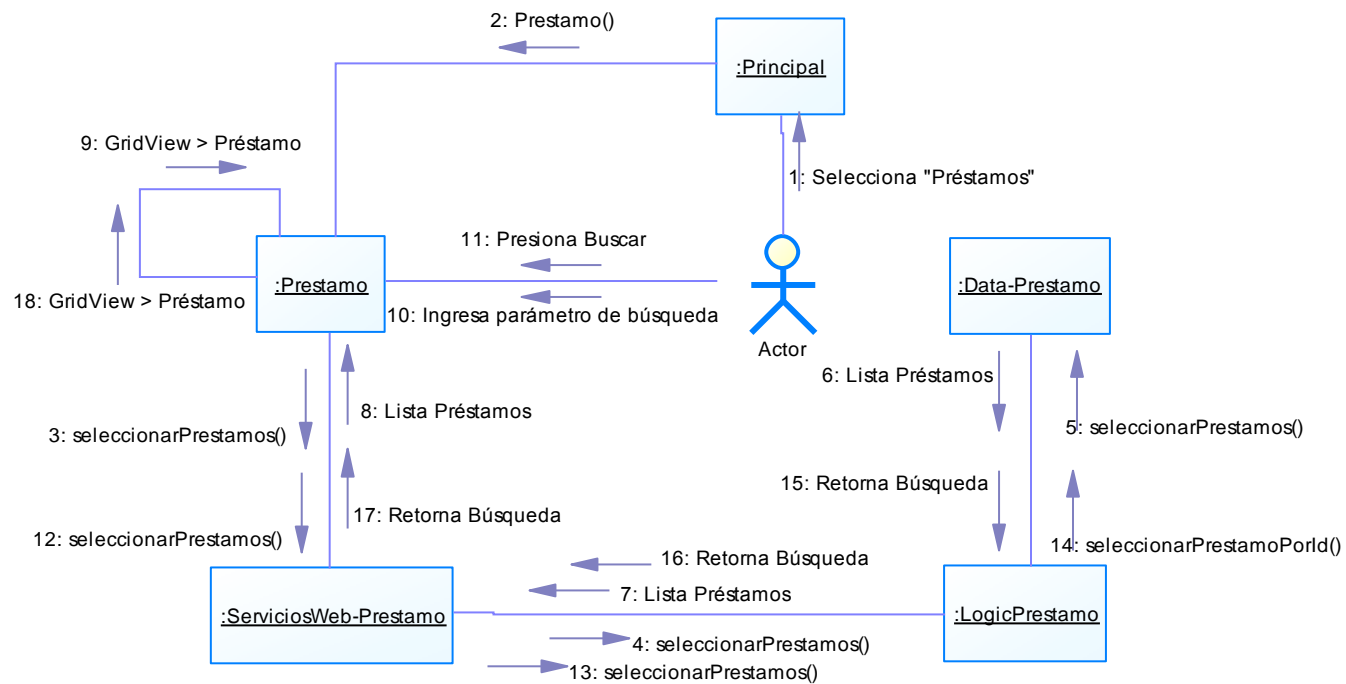


Diagrama 3-5 F3.4. Consultar Préstamo - Colaboración

F3.5. Eliminar Préstamo

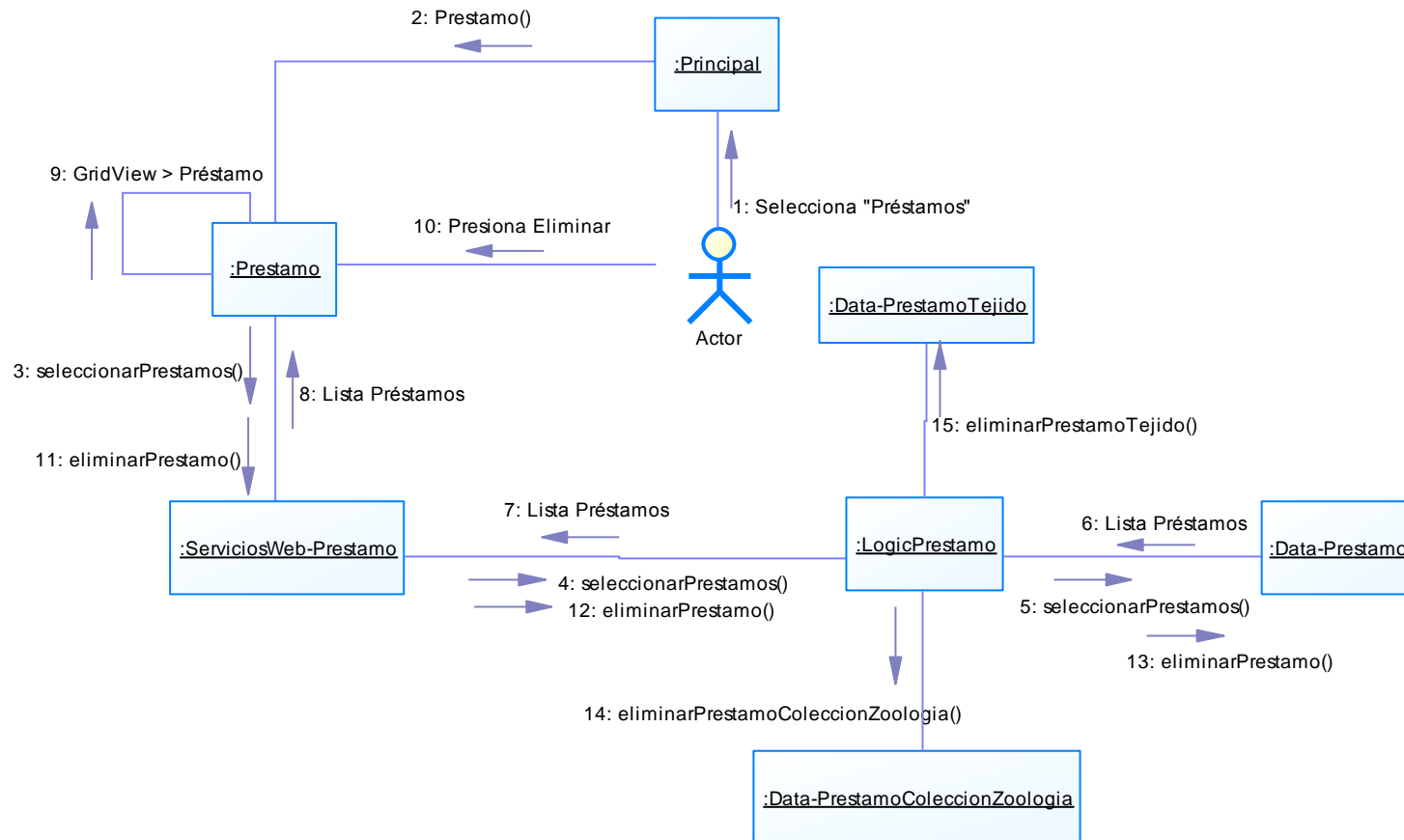


Diagrama 3 9 F3.5. Eliminar Préstamo - Colaboración

3.4.3. F4. Administrar Tejido

F4.1. Ingresar Tejido

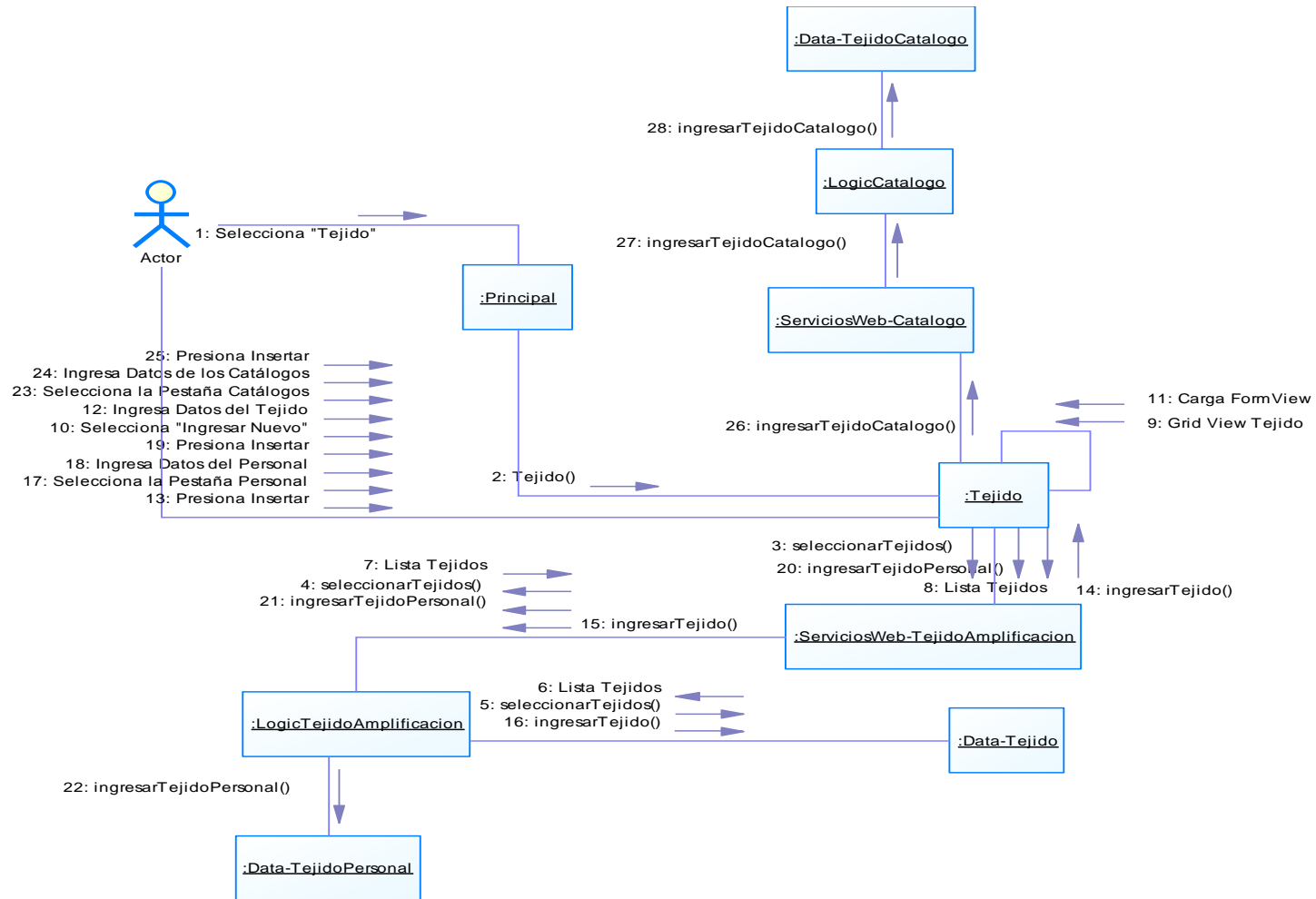


Diagrama 3-6 F4.1. Ingresar Tejido - Colaboración

F4.2. Modificar Tejido

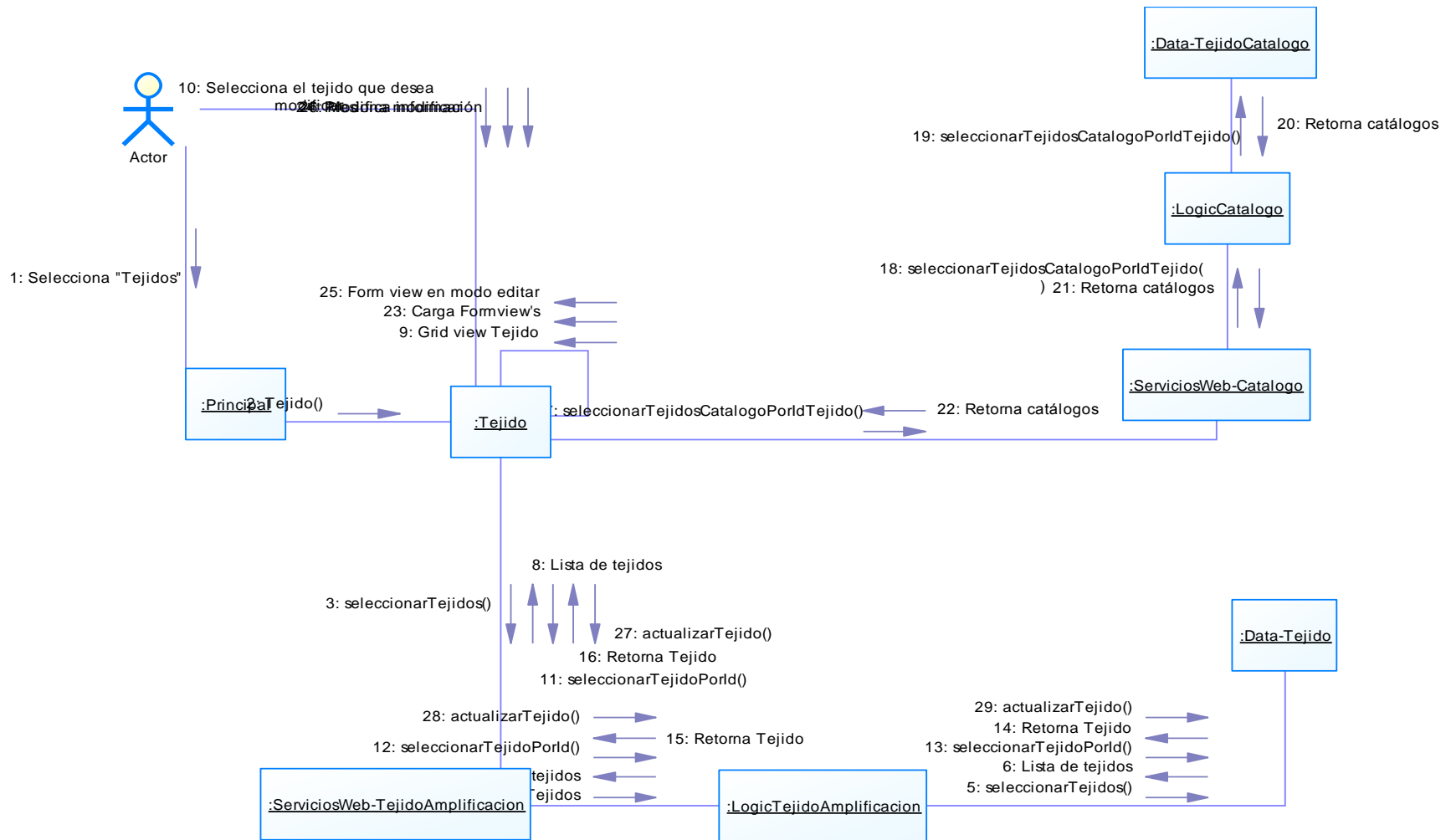


Diagrama 3-7 F4.2. Modificar Tejido - Colaboración

F4.3. Eliminar Tejido

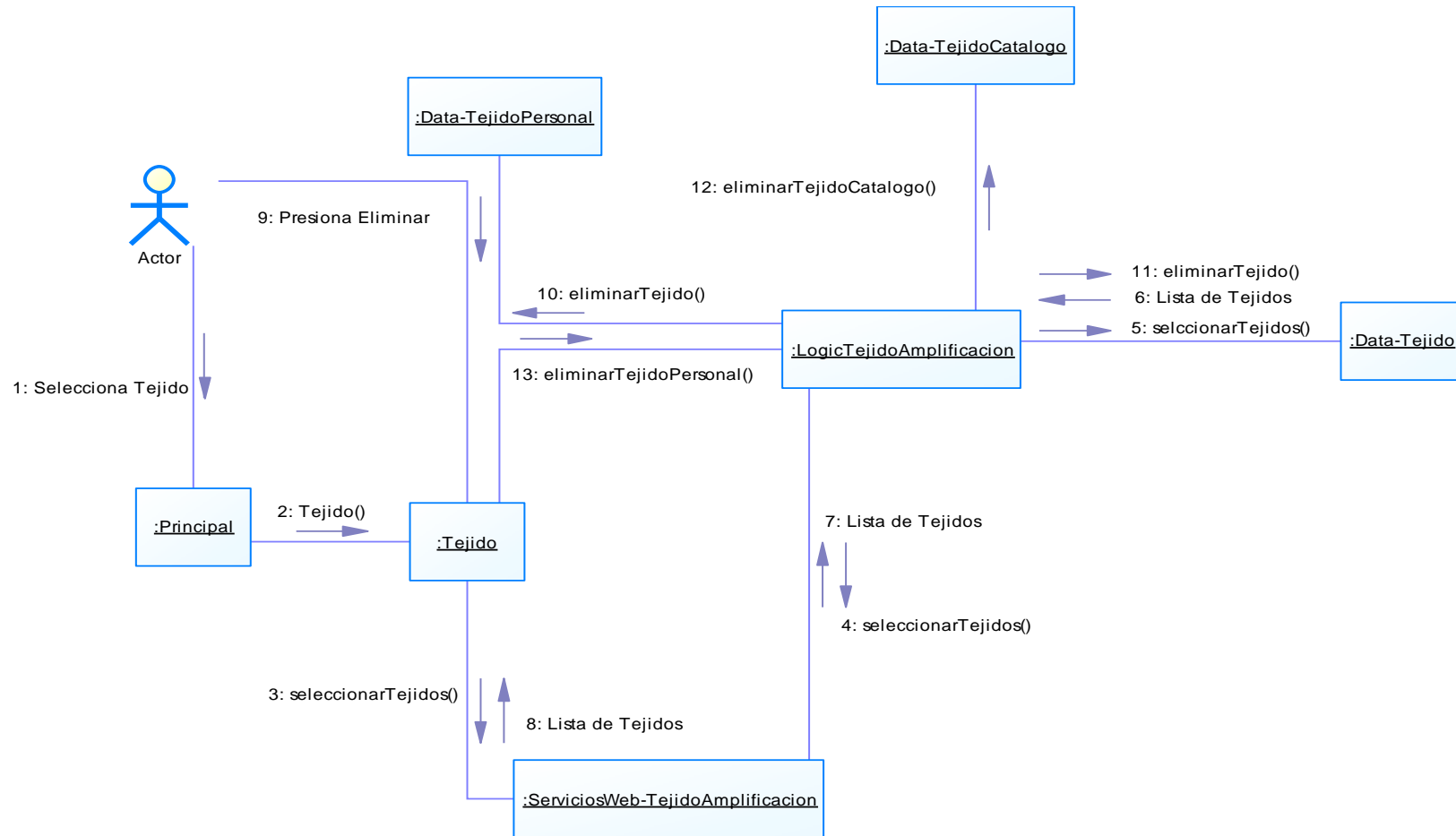


Diagrama 3 13 F4.3. Eliminar Tejido - Colaboración

F4.4. Consultar Tejido

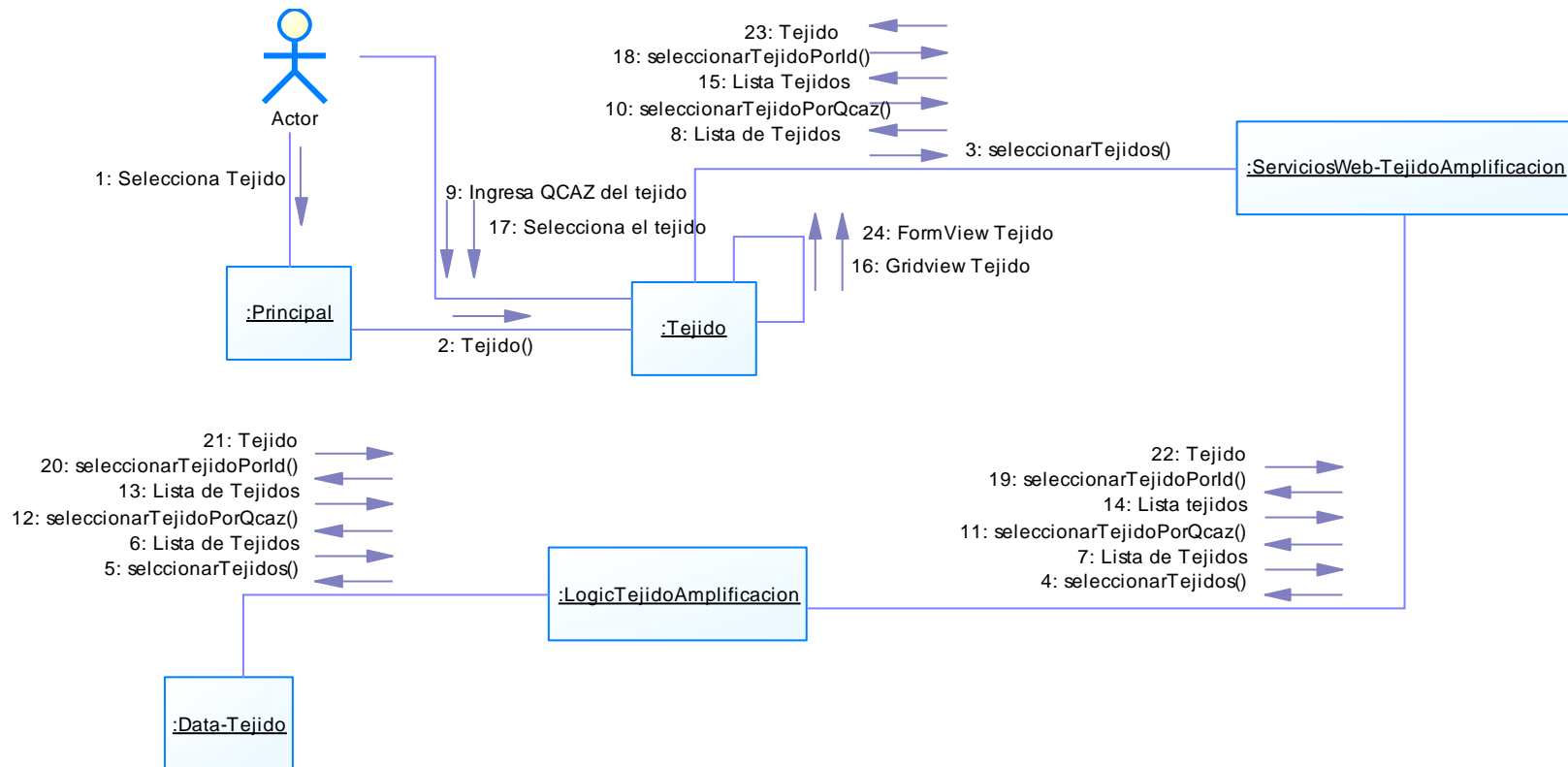


Diagrama 3 14 F4.4. Consultar Tejido - Colaboración

3.4.4. F5. Administrar Extracción

F5.1. Ingresar Extracción

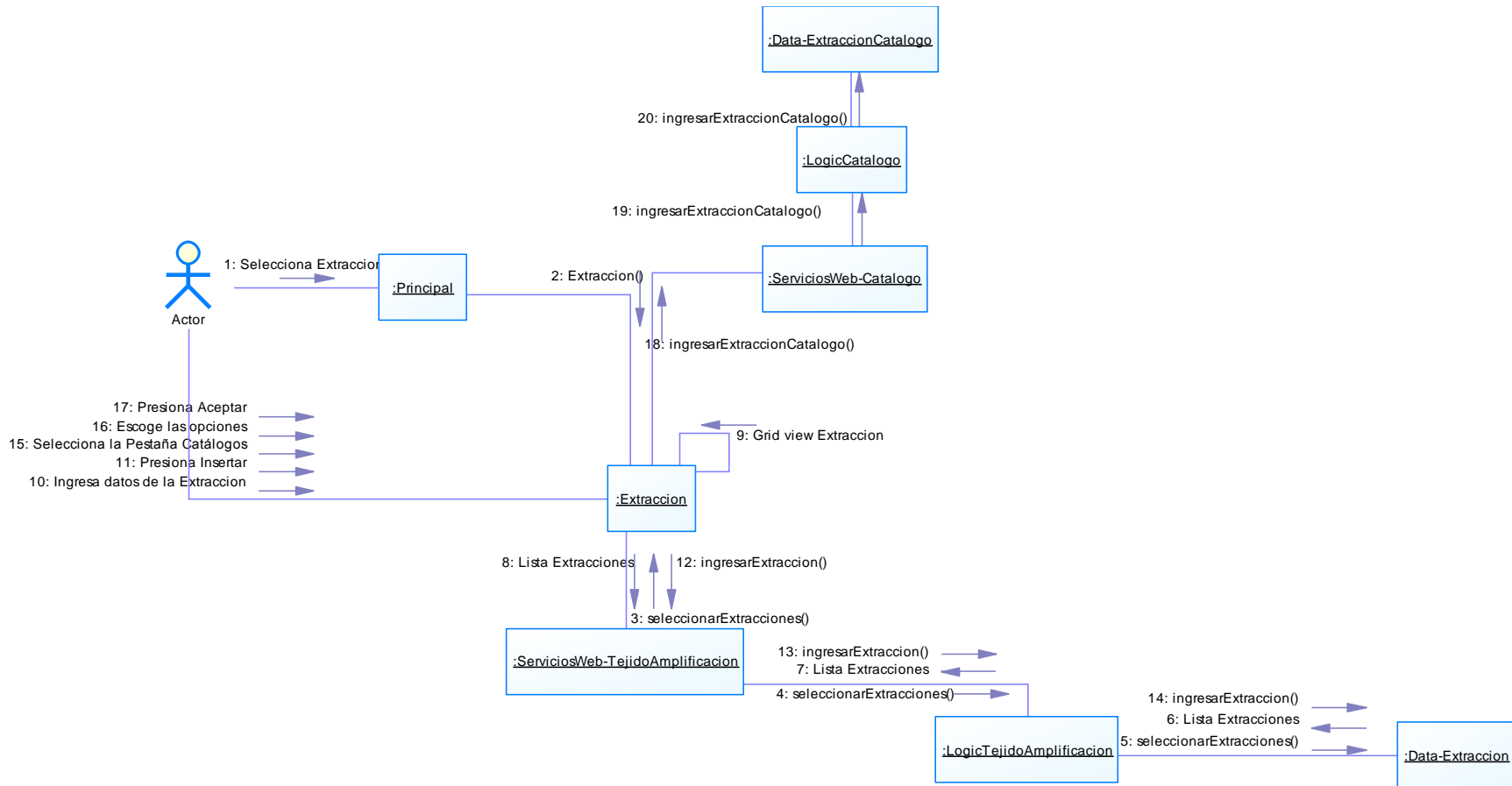


Diagrama 3-8 F5.1. Ingresar Extracción - Colaboración

F5.2. Modificar Extracción

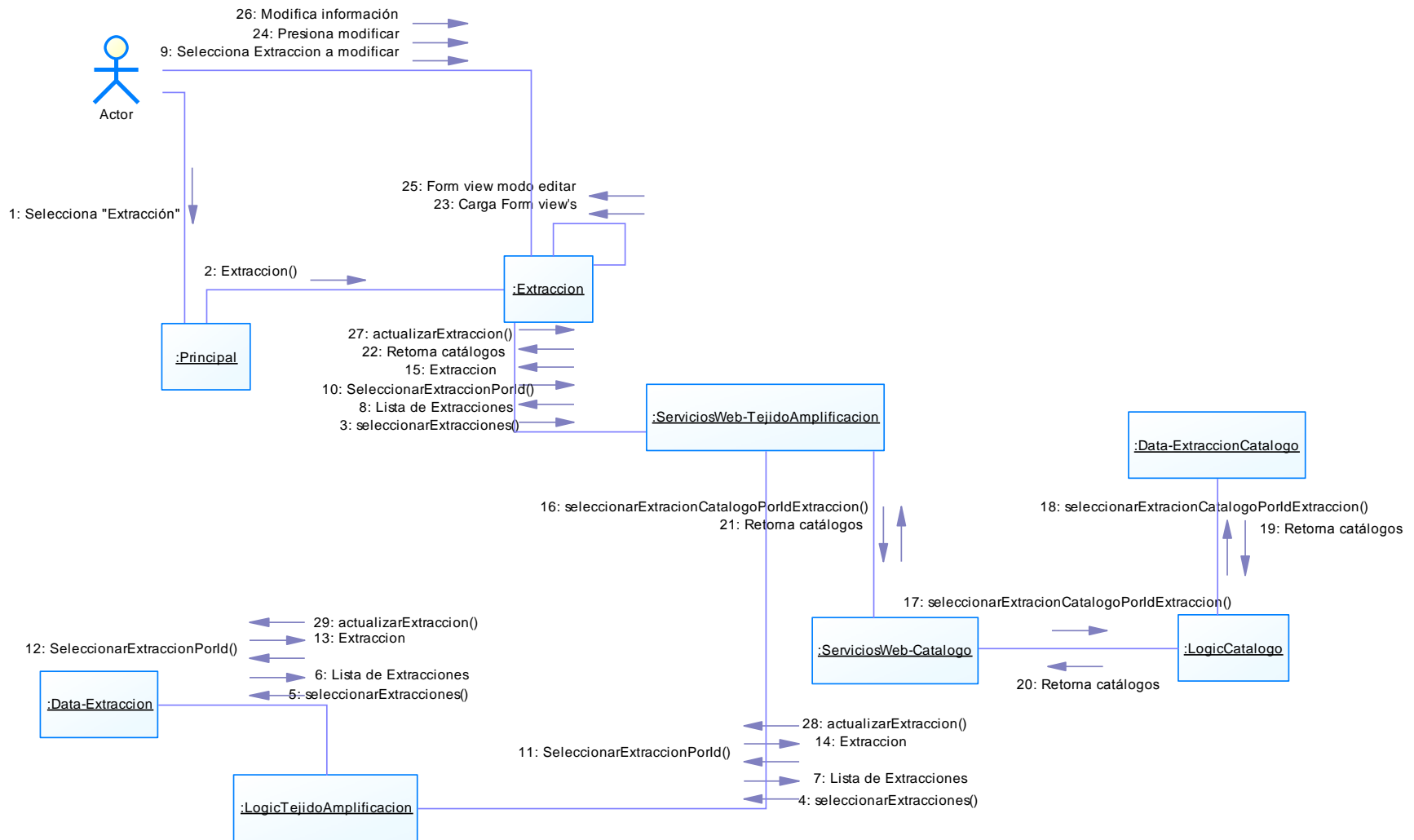


Diagrama 3-9 F5.2. Modificar Extracción - Colaboración

F5.3. Eliminar Extracción

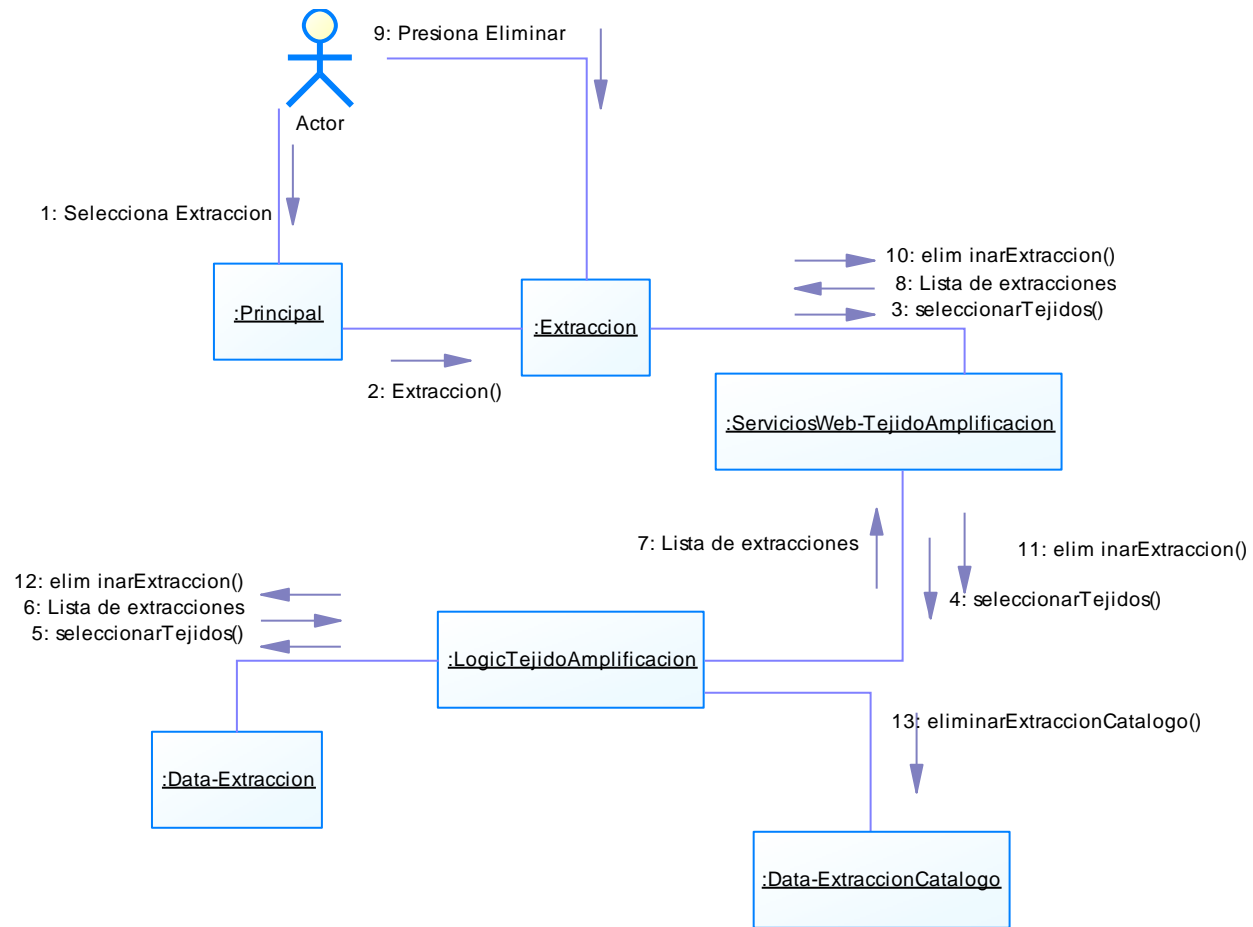


Diagrama 3 17 F5.3. Eliminar Extracción - Colaboración

F5.4. Consultar Extracción

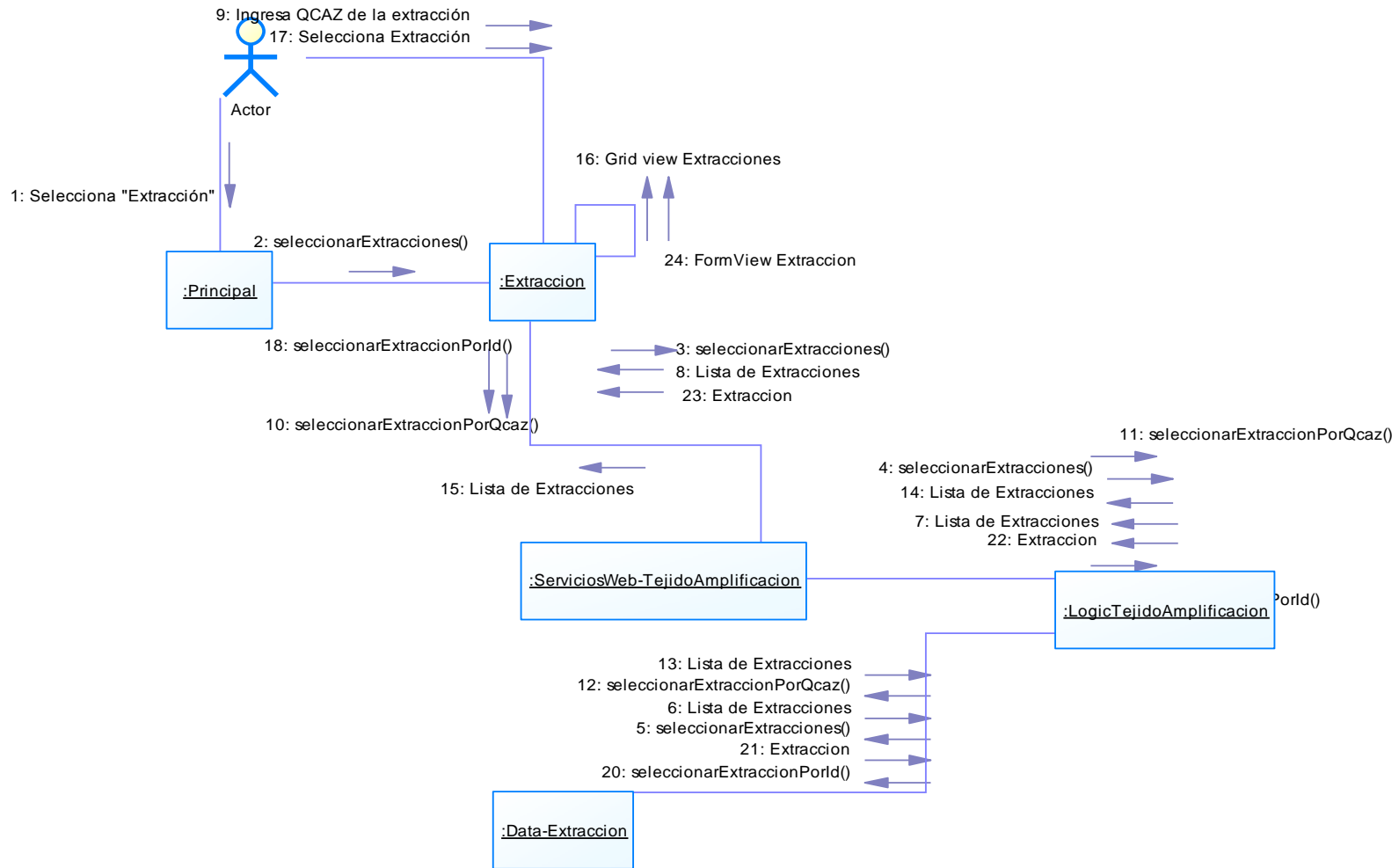


Diagrama 3-10 F5.4. Consultar Extracción - Colaboración

3.4.5. F6. Administrar Amplificación

F6.1. Ingresar Amplificación

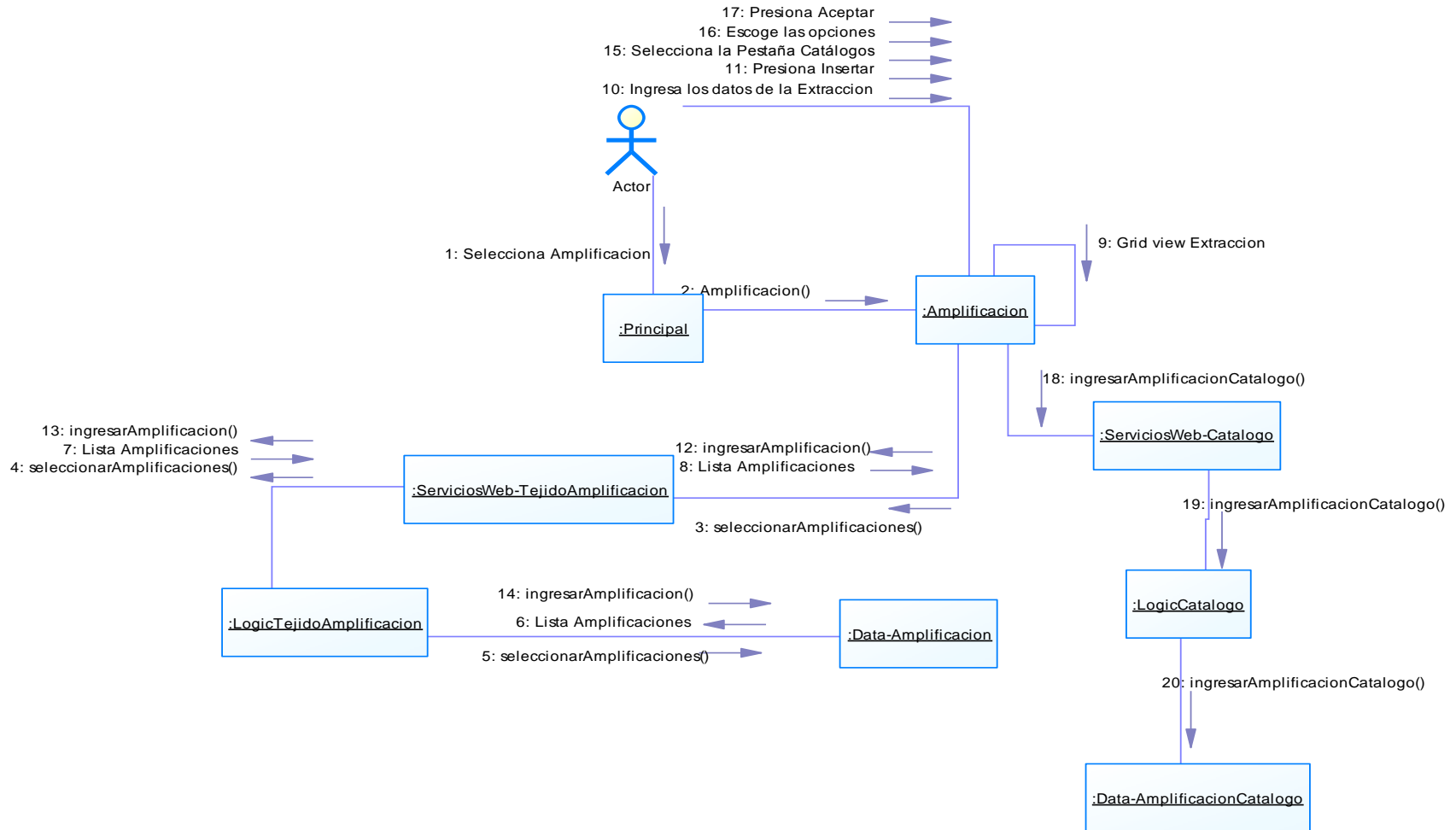


Diagrama 3-11 F6.1. Ingresar Amplificación - Colaboración

F6.2. Modificar Amplificación

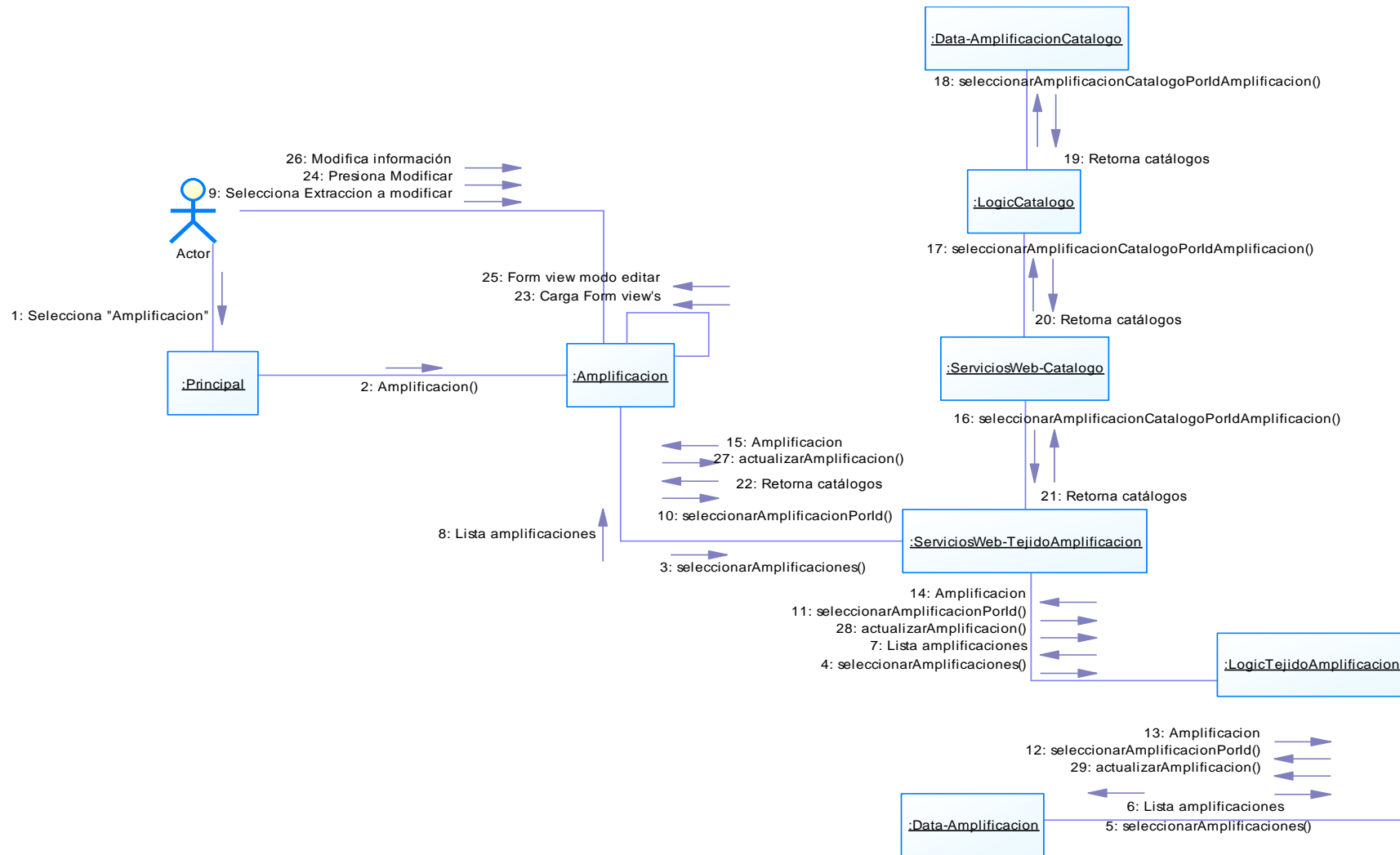


Diagrama 3-12 F6.2. Modificar Amplificación – Colaboración

F6.3. Eliminar Amplificación

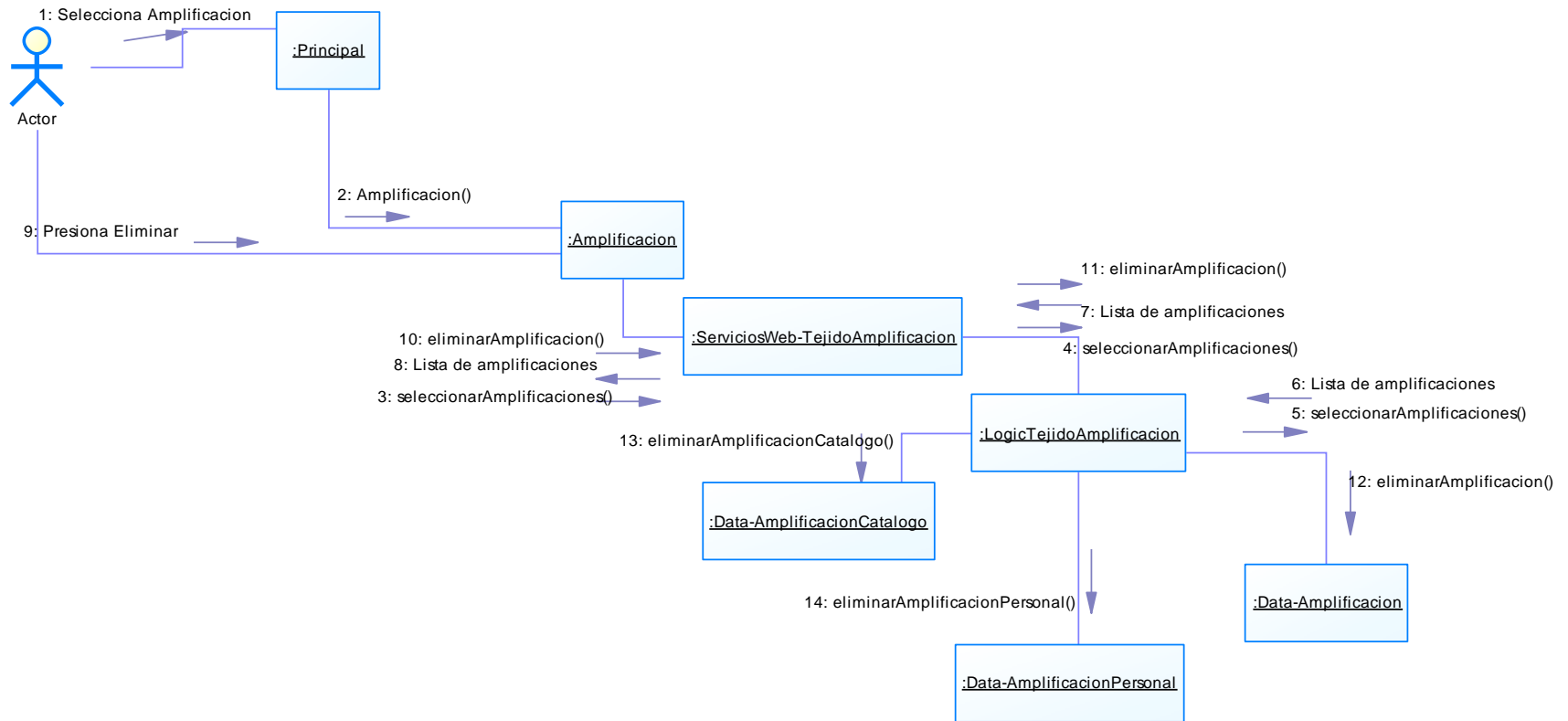


Diagrama 3-13 F6.3. Eliminar Amplificación - Colaboración

F6.4. Consultar Amplificación

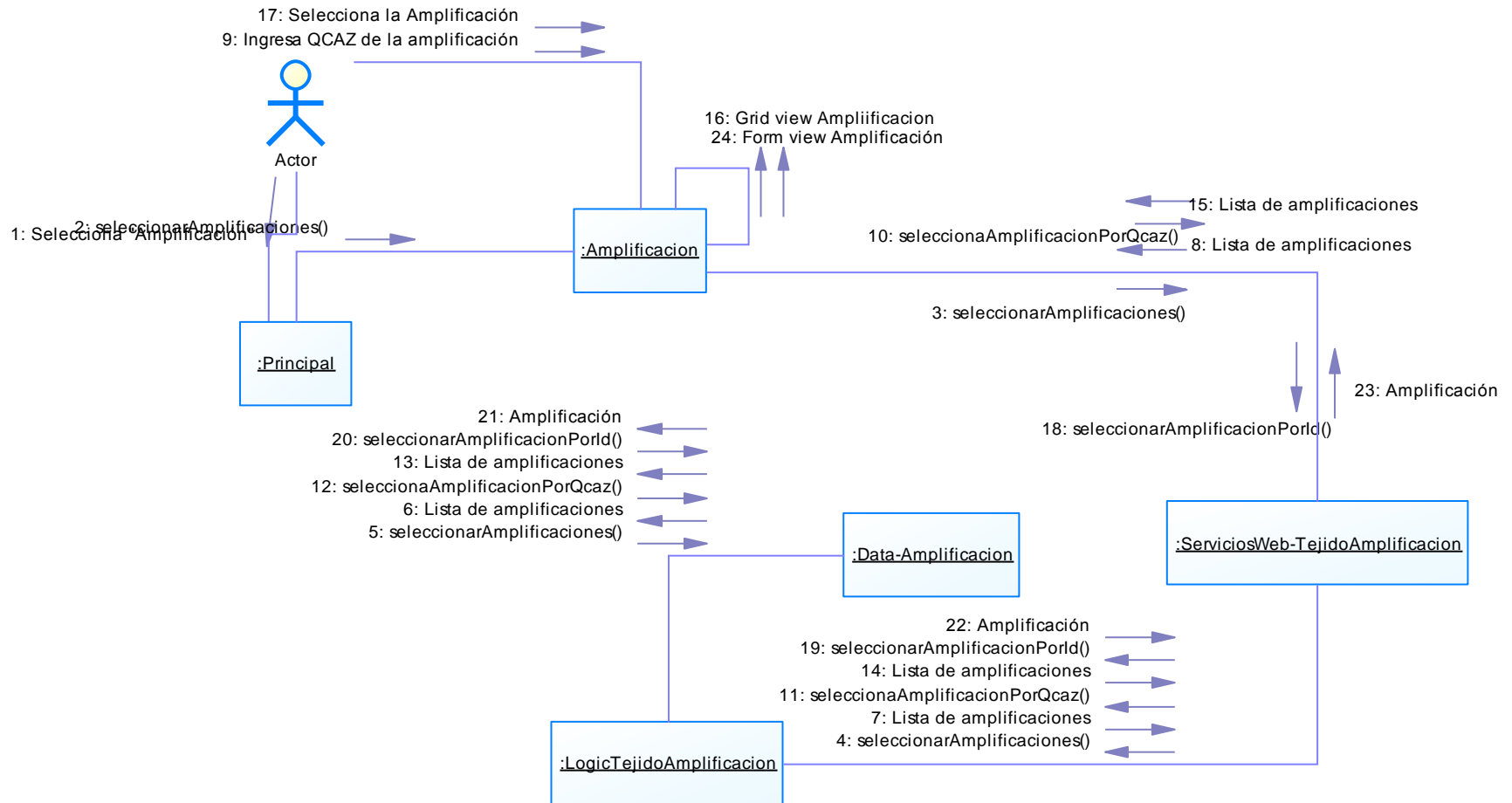


Diagrama 3 22 F6.4. Consultar Amplificación - Colaboración

3.4.6. F7. Administrar Cuaderno de Campo

F7.1. Ingresar Cuaderno de Campo

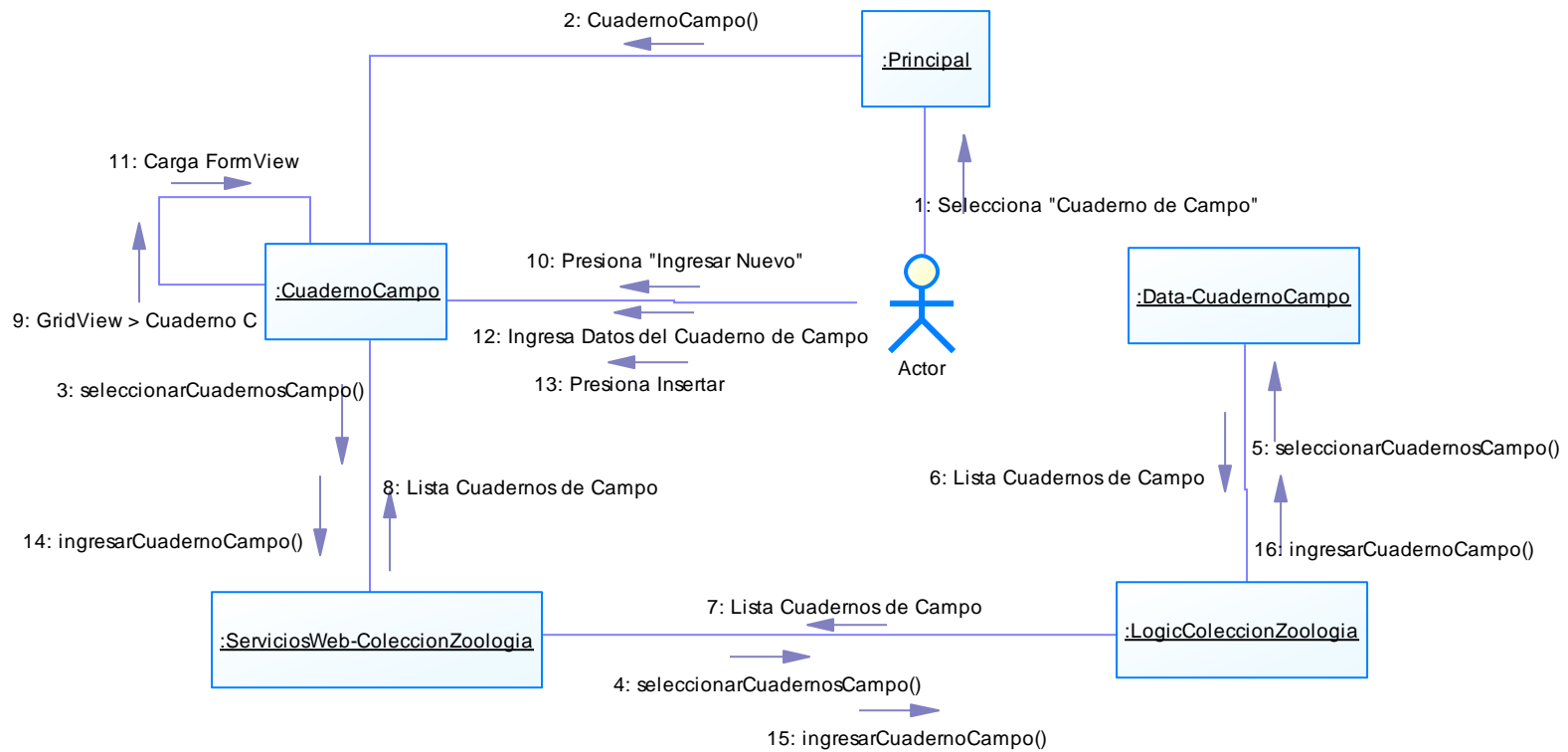


Diagrama 3-14 F7.1. Ingresar Cuaderno de Campo

F7.2. Modificar Cuaderno de Campo

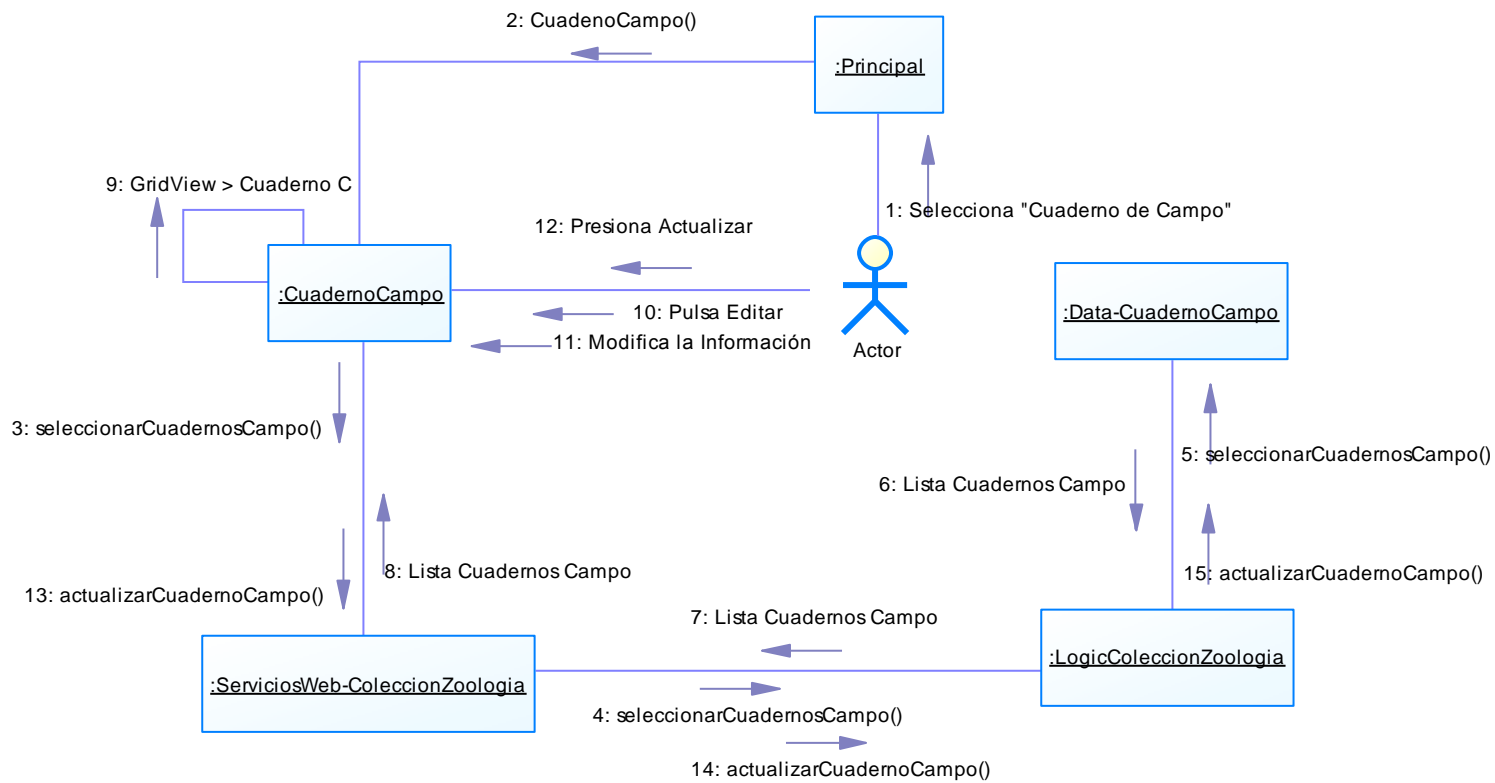


Diagrama 3 24 F7.2. Modificar Cuaderno de Campo

F7.3. Eliminar Cuaderno de Campo

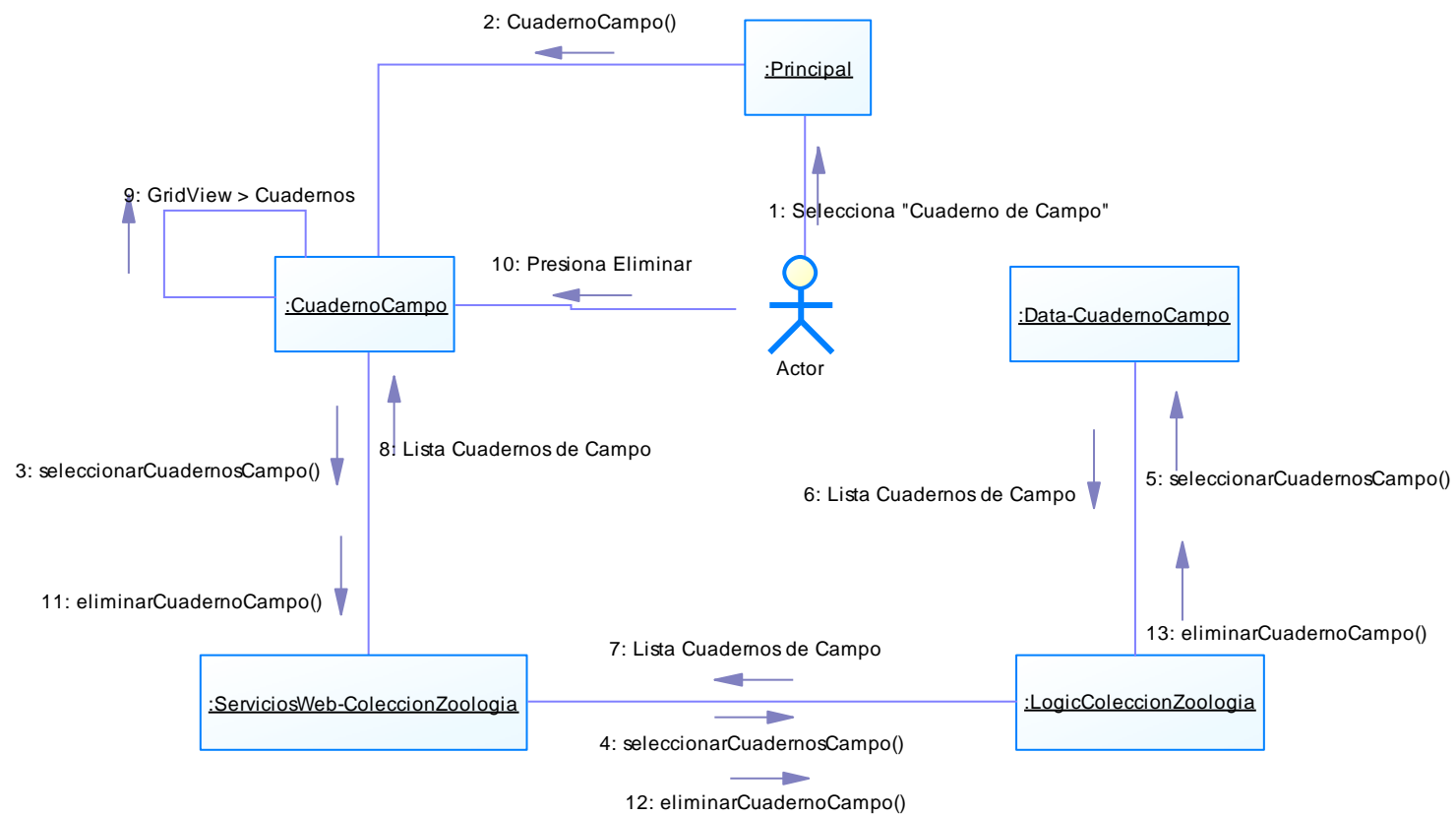


Diagrama 3-15 F7.3. Eliminar Cuaderno de Campo

F7.4. Consultar Cuaderno de Campo

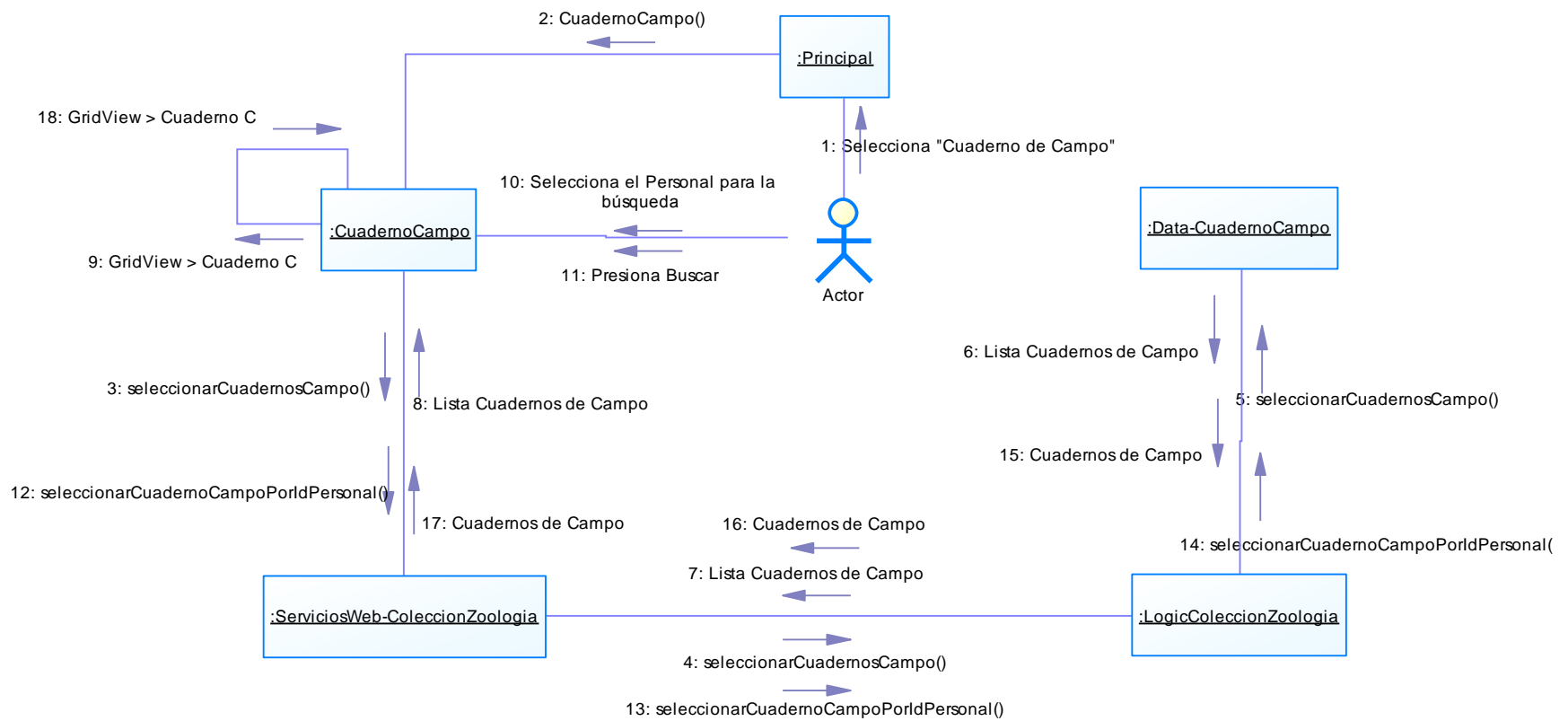
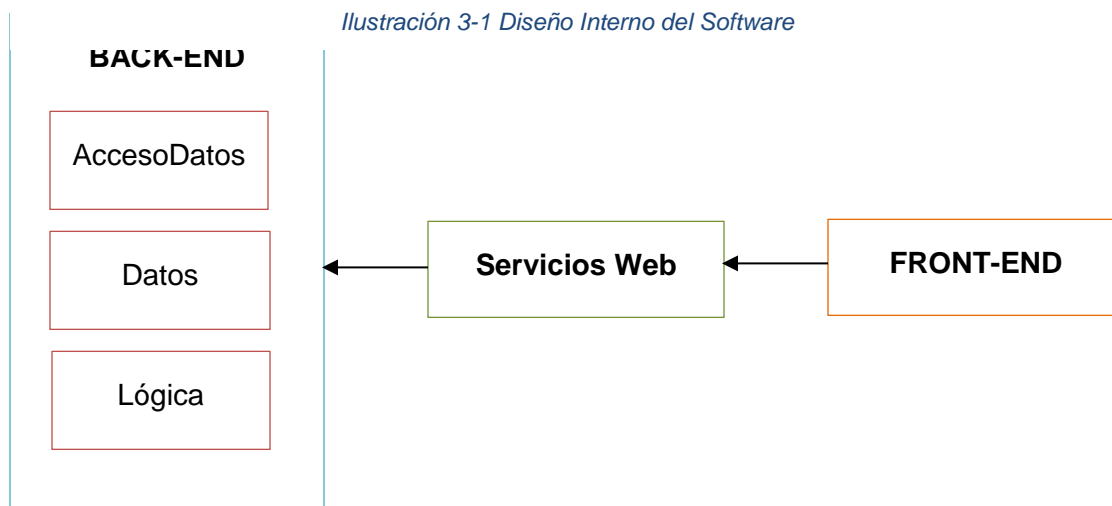


Diagrama 3-16 F7.4. Consultar Cuaderno de Campo

3.5. Diseño interno del Software

La implementación del sistema se realizó en base a una arquitectura de 3 capas y publicación de servicios web. La primera capa “Acceso a datos” es la encargada de conectarse a la base de datos y realizar las modificaciones que sean requeridas. La capa de “Datos” contiene cada clase diseñada en la base de datos y métodos para ingresar, consultar, modificar o eliminar información. En la capa “Lógica” agrupamos las clases para una mejor organización de las mismas, se distinguen seis grupos principales: los catálogos, colección zoología, personal, préstamo, taxa y tejido-amplificación. Esta capa posee los métodos de cada clase que ahora son agrupados asociados a cada grupo principal; por ejemplo, la clase Lógica tejido-amplificación encierra los métodos de tejido, extracción, amplificación, ultra congeladora y cuaderno laboratorio; la decisión sobre los grupos es basada en su relación en los procesos y en la base de datos. Finalmente, los servicios son publicados para ser usados en el diseño de la página web.



3.6. Diseño de Interfaces

Para el diseño de interfaces se utilizó un programa (JustinMind), el cual facilitó el diseño de las interfaces, ya que cuenta con herramientas propicias para dicho trabajo.

“Sistema para la administración de colecciones biológicas y compartición de la información Museo QCAZ”

El software se incorporará en con el sistema interno del museo QCAZ, el diseño se realizó siguiendo los estándares ya proporcionados, tomando como base la MasterPage:

Ilustración 3-2 Master Page - Diseño



Autores: Pablo Almeida & Vanessa Soria C.

La pantalla del Login y la pantalla principal ya serán las del sistema ya existente:

Ilustración 3-3 Pantalla Login - Diseño

El formulario de inicio de sesión tiene un título 'Iniciar sesión' en un encabezado azul. Incluye dos campos de entrada de texto para el nombre de usuario y la contraseña. Debajo de estos campos hay un checkbox con la etiqueta 'Recordármelo la próxima vez.' y un botón de 'Inicio de sesión' en la parte inferior derecha.

Autores: Pablo Almeida & Vanessa Soria C.

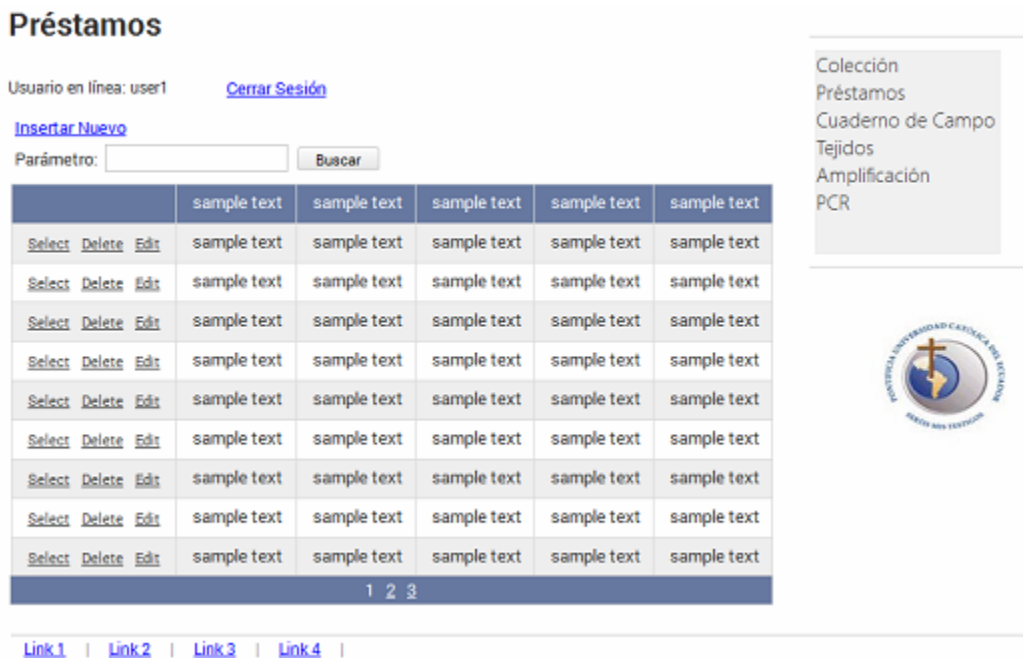
Ilustración 3-4 Pantalla Principal - Diseño



Autores: Pablo Almeida & Vanessa Soria C.

La pantalla de Consulta general será la pantalla principal para cada módulo, la cual, siguiendo el diseño de la MasterPage, el área de contenido tendrá la tabla de visualización de información:

Ilustración 3-5 Pantalla Consulta General - Diseño



Autores: Pablo Almeida & Vanessa Soria C.

La pantalla insertar mostrará formularios necesarios para permitir el ingreso de información.

Ilustración 3-6 Pantalla Insertar - Diseño

Autores: Pablo Almeida & Vanessa Soria C.

Para la edición, si la información es extensa, se mostrará una pantalla como la de ingreso, por el contrario, si los campos son pocos, se realizará la edición en la tabla de consulta:

Ilustración 3-7 Pantalla Edición - Diseño

Autores: Pablo Almeida & Vanessa Soria C

3.7. Diseño de servicios a utilizarse

Los servicios se dividen para mejorar la organización del código, los que utilizamos son:

- *ServiciosWeb-Catalogo*
- *ServiciosWeb-ColeccionZoologia*
- *ServiciosWeb-Personal*
- *ServiciosWeb-Prestamo*
- *ServiciosWeb-Taxa*
- *ServiciosWeb-TejidoAmplificacion*

Cada servicio cuenta con métodos generales para:

- Seleccionar una lista de todos los elementos ingresados
- Seleccionar elementos por ID
- Eliminar elementos por ID
- Ingresar nuevos elementos
- Actualizar elementos

Existen además métodos específicos para cada servicio, según las necesidades del proyecto.

4. CAPÍTULO 4: IMPLEMENTACIÓN

4.1. Estándares de Codificación

Existirán dos soluciones que encierran al proyecto. En uno se agrupan las clases de acceso a datos, datos, lógica y los servicios. La segunda solución está relacionada con la página web.

La primera solución será llamada "ProyectoBiologiaQCAZ", la página web tiene el nombre de "QCAZ-Internal"

ProyectoBiologiaQCAZ

Proyectos

Cada proyecto tendrá la primera letra mayúscula y si posee otra palabra en su nombre se colocará sin espacios con otra letra mayúscula, se omiten tildes y caracteres especiales para facilitar su uso. Ejemplo: "AccesoDatos", "Datos", "ServiciosSOAP", en el caso final usamos las siglas para evitar nombres demasiado largos.

Carpetas

Si es necesario crear una carpeta dentro de un proyecto respeta el formato de los proyectos, sin usar caracteres especiales, con la primera letra de cada palabra en mayúscula y sin espacios. Ejemplo: "ClasesParciales", "FakeTest"

Clases

Cada proyecto posee clases únicas que deben ser fácilmente reconocidas. Por lo tanto, una clase debe poseer un identificador antes del nombre de la misma. Los identificadores se muestran a continuación:

Proyecto	Prefijo
AccesoDatos	DataAccess_
Datos	Data_
Logica	Logic
ServiciosSOAP	ServiciosSOAP_

Las tres primeras capas poseen el prefijo en inglés para diferenciarlos de los servicios, mientras que la capa lógica no poseerá un guion bajo dado que sirve para agrupar las clases anteriores. Si es necesario utilizar clases parciales se respeta el nombre original de la clase.

Métodos

Los métodos asociados a cada clase son programados en la capa de "Datos" haciendo referencia a la capa de "AccesoDatos", para este estándar tenemos en cuenta que el nombre del método debe dejar claro cuál es su función y que argumentos necesita.

La primera letra de cada método es minúscula, no habrá espacios y si se coloca otra palabra tendrá la primera letra mayúscula. En general cada método comienza con un verbo en infinitivo de la acción que se va a realizar: seleccionar, eliminar, actualizar, ingresar; seguidos de la clase a la que afectan: "eliminarCuadernoCampo" o "actualizarTejido".

Es importante recalcar que el nombre del método será en singular o plural dependiendo si devuelve una Lista o un solo elemento en particular, ejemplo: "seleccionarTejidos" si devuelve una lista o "seleccionaTejido" si devuelve un sólo elemento. Actualizar, eliminar e ingresar generalmente actúan sobre un único elemento.

Si se necesita un argumento se colocará el sufijo "Por" seguido del argumento necesario, por ejemplo: "eliminarCuadernoPorId", "seleccionarTejidosPorIdColeccionZoologia"; así es fácil reconocer los parámetros necesarios.

Si son necesarios varios argumentos sólo se colocarán seguidos uno tras otro, ejemplo: "seleccionarCatalogoPorIdTejidoIdTipoCatalogo".

El nombre de un argumento puede estar relacionado con la acción que realice: "nuevaColeccionZoologia" o "actualizaTejido" pero no es un punto necesario si el nombre se vuelve muy largo.

QCAZ-Internal

En la página web se crearán carpetas para agrupar páginas que trabajan sobre clases relacionadas directamente, realizan una función similar a la capa lógica, permitir mayor organización.

Carpetas

Las carpetas tienen la primera letra mayúscula, no habrá espacios si existen más palabras y cada palabra extra tendrá la primera letra mayúscula también. Ejemplo: "TejidoAmplificación".

Web Form

Las páginas asociadas a la administración de cada clase poseen el mismo nombre de la clase, por ejemplo "ColeccionZoologia". Sin embargo, para los elementos que utilizamos dentro de cada página tendremos prefijos previamente definidos como se muestra en la tabla:

Elemento	Prefijo
ObjectDataSource	ods
HiddenField	hdf
TextBox	txt
Button	btn
DropDownList	dpd
Label	lbl
LinkButton	lbt
Accordion	acordion
FormView	fvw
GridView	gvw

Después del prefijo se coloca el elemento sobre el que trabajan, ejemplo: "gwwTejido", "odsCuadernoCampo" en el caso de un botón se coloca la acción que vaya a realizar: "btnIngresar".

Si el formato del código es autogenerado (en el caso de una formview o gridview, por ejemplo) se puede respetar dicho formato.

Funciones

El nombre de las funciones nuevas tiene por defecto un sufijo sobre su acción, por ejemplo: "odsCatalogo_Deleting" o "odsAmplificacion_Inserting" tal formato será respetado.

4.2. Diagrama de Despliegue

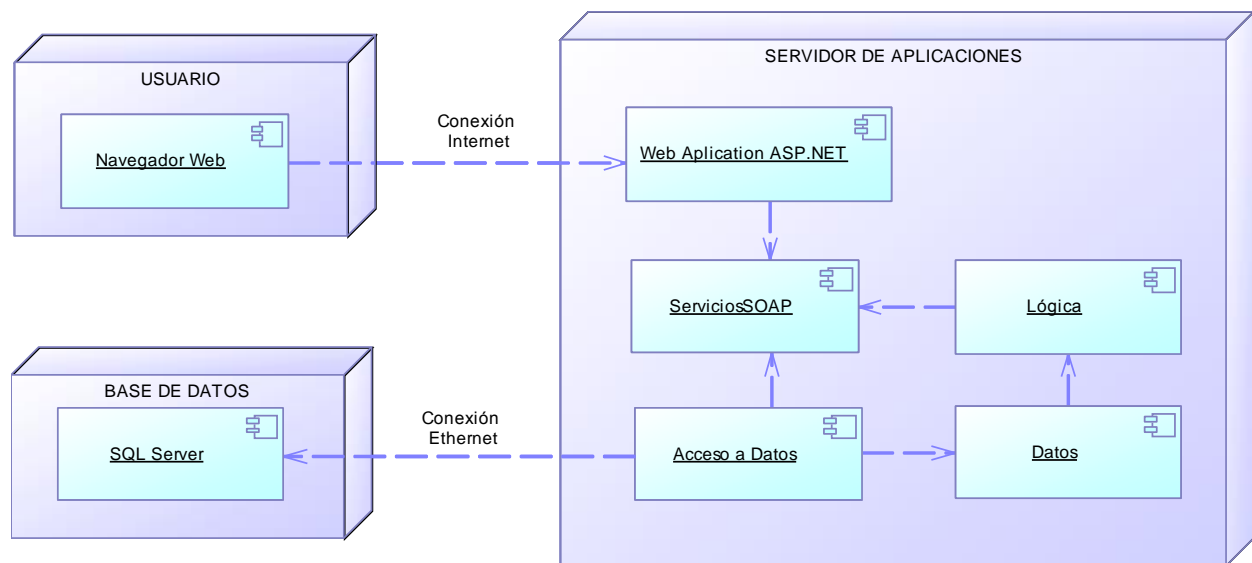


Diagrama 4-1 Despliegue

Existen tres bloques principales, el usuario, el servidor de aplicaciones y la base de datos. El usuario accede a través de la web a la aplicación y por medio de la misma a la base de datos.

El servidor de aplicaciones asp.net engloba a los servicios web, y las diferentes capas del sistema.

4.3. Diagrama de Componentes

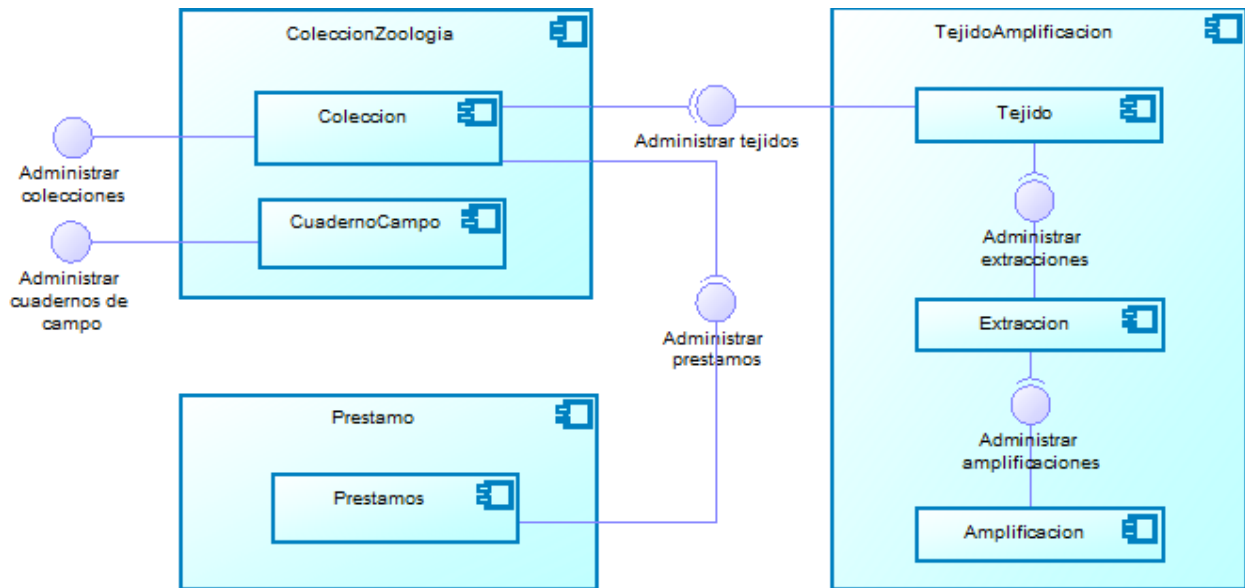


Diagrama 4-2 Diagrama de Componentes

En el diagrama se muestran las interfaces y componentes necesarios para permitir el flujo del sistema. Los componentes se engloban para facilitar la organización.

Diferentes componentes dependen de la creación previa de ciertos elementos para poder existir. Por ejemplo, para realizar un préstamo es necesario haber ingresado una colección.

4.4. Diagrama de Paquetes

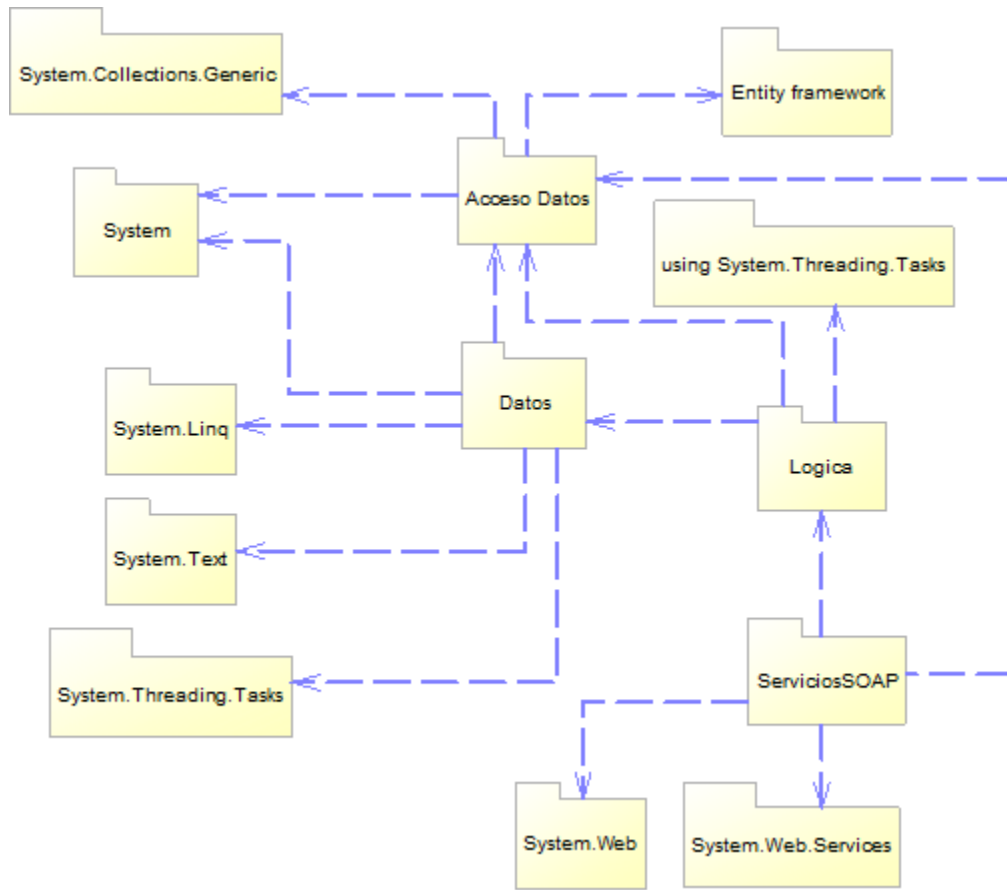


Diagrama 4-3 Diagrama de Paquetes

Los paquetes más importantes son los de las clases creadas más el entity framework, linq, web services y system.

En el diagrama se pueden observar las relaciones entre los paquetes, al igual que las dependencias entre ellos.

4.5. Pruebas del Sistema

4.5.1. Caso de Prueba: F1. Administrar Colección

Precondición: Administrar Cuaderno de Campo

Entradas	Resultado Esperado	Caso de Uso	Estado
Asignar Taxonomía - Selecciona - No Selecciona	- Desbloquea formularios - Formularios bloqueados	F1.1.1.1	Correcto
Ingresar Identificación - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F1.1.1	Correcto
Ingresar Localidad - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F1.1.2	Correcto
Ingresar Personal - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F1.1.3	Correcto
Ingresar Catálogos - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F1.1.4	Correcto
Modificar Colección - Datos Correctos - Datos Incorrectos	- Actualiza Información - Mensaje de Error	F1.2	Correcto
Eliminar Colección - Acepta - Cancela	- Actualiza estado a borrado - Se cancela la acción	F1.3	Correcto
Consulta Colección - QCAZ Correcto - QCAZ Incorrecto	- Se muestran datos - Mensaje de Error	F1.4	Correcto

Pos condiciones: Ejecutar 20 veces F1.1. para probar los demás módulos

4.5.2. Caso de Prueba: F3. Administrar Préstamos

Precondición: F1. Administrar Colección Zoología y F3. Administrar Tejidos

Entradas	Resultado Esperado	Caso de Uso	Estado
Ingresar Información - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F3.1.1	Correcto
Añadir Especímenes - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F3.1.2	Correcto
Añadir Tejidos - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F3.1.3	Correcto
Generar Documento	Muestra documento	F3.2	Correcto
Modificar Préstamo - Datos Correctos - Datos Incorrectos	- Actualiza Información - Mensaje de Error	F3.3	Correcto
Eliminar Préstamo - Acepta - Cancela	- Actualiza estado a borrado - Se cancela la acción	F3.4	Correcto
Consultar Préstamo - Existen datos - No existen datos	- Se muestran datos - Mensaje de Error	F3.5	Correcto
Realizar devolución - Acepta - Cancela	- Actualiza Información - Mensaje de Error	F3.6	Correcto

4.5.3. Caso de Prueba: F4. Administrar Tejido

Entradas	Resultado Esperado	Caso de Uso	Estado
Ingresar Tejido - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F4.1	Correcto
Modificar Tejido - Datos Correctos	- Actualiza Información - Mensaje de Error	F4.2	Correcto

- Datos Incorrectos			
Eliminar Tejido - Acepta - Cancela	- Actualiza estado a borrado - Se cancela la acción	F4.3	Correcto
Consultar Tejido - Existen datos - No existen datos	- Se muestran datos - Mensaje de Error	F4.4	Correcto

4.5.4. Caso de Prueba: F5. Administrar Extracción

Entradas	Resultado Esperado	Caso de Uso	Estado
Ingresar Extracción - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F5.1	Correcto
Modificar Extracción - Datos Correctos - Datos Incorrectos	- Actualiza Información - Mensaje de Error	F5.2	Correcto
Eliminar Extracción - Acepta - Cancela	- Actualiza estado a borrado - Se cancela la acción	F5.3	Correcto
Consultar Extracción - Existen datos - No existen datos	- Se muestran datos - Mensaje de Error	F5.4	Correcto

4.5.5. Caso de Prueba: F6. Administrar PCR

Entradas	Resultado Esperado	Caso de Uso	Estado
Ingresar PCR - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F6.1	Correcto
Modificar PCR - Datos Correctos - Datos Incorrectos	- Actualiza Información - Mensaje de Error	F6.2	Correcto
Eliminar PCR - Acepta	- Actualiza estado a borrado - Se cancela la acción	F6.3	Correcto

- Cancela			
Consultar PCR	- Se muestran datos - Mensaje de Error	F6.4	Correcto
- Existen datos - No existen datos			

4.5.6. Caso de Prueba: F7. Administrar Cuaderno de Campo

Entradas	Resultado Esperado	Caso de Uso	Estado
Ingresar Cuaderno de Campo - Datos Correctos - Datos Incorrectos	- Guarda Información - Mensaje de Error	F7.1	Correcto
Modificar Cuaderno de Campo - Datos Correctos - Datos Incorrectos	- Actualiza Información - Mensaje de Error	F7.2	Correcto
Eliminar Cuaderno de Campo - Acepta - Cancela	- Actualiza estado a borrado - Se cancela la acción	F7.3	Correcto
Consultar Cuaderno de Campo - Existen datos - No existen datos	- Se muestran datos - Mensaje de Error	F7.4	Correcto

4.6. Instalación y configuración

(Ver manual de uso)

CONCLUSIONES

- En la actualidad el uso de Web Services, más que una buena práctica, es parte fundamental para el desarrollo de aplicaciones, ya que proporciona mayor seguridad y una mejor estructura en los proyectos.
- ASP.NET es un framework para el desarrollo de aplicaciones web, el cual ofrece una gran cantidad de herramientas que facilitan la programación y el diseño de diferentes partes del sistema.
- Es necesario seguir estándares de programación de esta manera la mantenibilidad de los sistemas se hace viable, de igual manera se debe de dejar abierto para futuras modificaciones según las necesidades del cliente.
- La implementación de una arquitectura en capas, es primordial para que el sistema sea seguro, ya que la capa de la base de datos se encuentra separada de la aplicación en sí.

RECOMENDACIONES

- Utilizar un repositorio para el versionamiento de la aplicación es importante para poder tener un control sobre los cambios que se realizan en el proyecto, de esta manera el proyecto no se ve comprometido.
- El diseño es la parte esencial para el desarrollo de un proyecto de Software, en este se debe proyectar el alcance del proyecto y también dejar una ventana sobre las posibles necesidades futuras del cliente, para evitar tener que realizar mayores cambios en el futuro.
- El desarrollador debe considerar que un usuario final no va a utilizar el sistema como se espera y debe estar preparado para evitar fallas.
- Es necesario tener en cuenta posibles errores en el ingreso de la información, una buena forma de mitigar errores de información, es realizar las verificaciones en el front end de la aplicación, antes de realizar el ingreso a la base.
- Cuando se maneja una base de datos con una gran cantidad de información se debe tener en cuenta la velocidad de respuesta para la obtención de información.
- El desarrollo de un sistema debe progresar con reuniones frecuentes con el usuario final, para asegurar el cumplimiento de los requerimientos.

BIBLIOGRAFÍA

- Abercrombie, M., Hickman, C., & Johnson, M. (1970). *Diccionario de biología*. Barcelona: Labor.
- Bell, D., & Parr, M. (2010). *C# para estudiantes* (Primera edición ed.). México: PEARSON EDUCACIÓN.
- Cardozzo, D. (2016). *Desarrollo de Software: Requisitos, Estimaciones y Análisis*. IT Campus Academy.
- Cobo, Á. (s.f.). *Diseño y programación de bases de datos*. Madrid: Vision Libros.
- Colouris, G., Dollimore, J., & Kindberg, T. (2001). *Sistemas Distribuidos Conceptos y Diseño*. Madrid: Pearson Educación.
- De la Cruz, F. (2014). *Método RUP*. Quito: Clases dictadas PUCE.
- Definición. (s.f.). *Lenguaje de Programación*. Obtenido de Definición.DE:
<http://definicion.de/lenguaje-de-programacion/>
- Equipo Vértice. (s.f.). *Diseño básico de páginas web en HTML*. Málaga: Vértice.
- García Deras, G., López de Aquino, M., Honey Escandón, N., Cortés Rodríguez, B., & Hernández Baños. (2001). *La importancia actual de las colecciones de tejidos*. Mexico: CONABIO - Biodiversitas.
- Garrido Ramos, M. (s.f.). *Reacción en Cadena de la Polimerasa (PCR)*. Obtenido de <http://www.ugr.es/~mgarrido/PCR.htm>
- González, B. (07 de Julio de 2004). *SOAP (Simple Object Access Protocol)*. Obtenido de desarrolloweb.com: <http://www.desarrolloweb.com/articulos/1557.php>
- Hidalgo, M. (s.f.). *Informe de Ingeniería - Diseño un Sistema de Información para el control del Patrimonio Predial*. Cacatachi.

Kroll, P., & Kruchten, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Boston: Pearson Educación.

LEXICOON. (Febrero de 2016). *Extracción*, Edición 3.8. Obtenido de <http://lexicoon.org/es/extraccion>

Menéndez, R. (05 de Abril de 2014). *Capítulo 1. Ingeniería del software*. Obtenido de Universidad de Murcia: <http://www.um.es/docencia/barzana/IAGP/IAGP2-Ingenieria-software-introduccion.html>

Microsoft. (2007). *Utilizar controles de ASP.NET*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/bb386451\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/bb386451(v=vs.100).aspx)

Microsoft. (s.f.). *ADO.NET*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/e80y5yhx\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/e80y5yhx(v=vs.110).aspx)

Microsoft. (s.f.). *Arquitectura de ADO.NET*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/27y4ybxw\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/27y4ybxw(v=vs.110).aspx)

Microsoft. (s.f.). *C#*. Obtenido de Microsoft Developer Network: <https://msdn.microsoft.com/es-es/library/kx37x362.aspx>

Microsoft. (s.f.). *Entity Framework*. Obtenido de Microsoft: <https://msdn.microsoft.com/es-es/data/ef.aspx>

Microsoft. (s.f.). *Funciones integradas (Transact-SQL)*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/ms174318\(v=sql.110\).aspx](https://msdn.microsoft.com/es-es/library/ms174318(v=sql.110).aspx)

Microsoft. (s.f.). *Información general acerca de .NET Framework*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)

Microsoft. (s.f.). *Información general sobre ASP.NET*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/4w3ex9c2\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx)

- Microsoft. (s.f.). *Información general sobre el archivo .edmx (Entity Framework)*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/cc982042\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/cc982042(v=vs.100).aspx)
- Microsoft. (s.f.). *Microsoft SQL Server*. Obtenido de Microsoft: <https://msdn.microsoft.com/es-es/library/bb545450.aspx>
- Microsoft. (s.f.). *Modelo Conceptual*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/bb738642\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/bb738642(v=vs.110).aspx)
- Microsoft. (s.f.). *N-Tier Data Applications Overview*. Obtenido de Microsoft Developer Network: <https://msdn.microsoft.com/en-us/library/bb384398.aspx>
- Microsoft. (s.f.). *Referencia de Transact-SQL (Transact-SQL)*. Obtenido de Microsoft TechNet: [https://technet.microsoft.com/es-es/library/ms189826\(v=sql.90\).aspx](https://technet.microsoft.com/es-es/library/ms189826(v=sql.90).aspx)
- Microsoft. (s.f.). *Resumen de las características de C#*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/aa287483\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa287483(v=vs.71).aspx)
- Nevado Cabello, M. V. (s.f.). *Introducción a Las Bases de Datos Relacionales*. Madrid: Vision Libros.
- Pérez Valdés, D. (06 de Octubre de 2007). *¿Qué son las bases de datos?* Obtenido de <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- Petersen, J. V. (2012). *Absolute Beginner's Guide to Databases*. Indianapolis: Que Publishing.
- Pressman, R. S. (2010). *Ingeniería de Software: Un enfoque práctico*. México: McGrawHill.
- RAE. (s.f.). *extracción*. Obtenido de REAL ACADEMIA DE LA LENGUA ESPAÑOLA: <http://dle.rae.es/?id=HOFmHUL>
- RAE, R. A. (s.f.). *Lenguaje*. Obtenido de Diccionario de la Lengua Española: <http://dle.rae.es/?id=N7BnIFO>

Simmons, J., & Muñoz-Saba, Y. (2005). *Cuidado, manejo y conservación de las colecciones biológicas*. Bogotá: Universidad Nacional de Colombia.

Snell, J., Tidwell, D., & Kulchenko, P. (2002). *Programming Web Services with SOAP*. Sebastopol: O'Reilly & Associates.

Sommerville, I. (2005). *Ingeniería del Software*. Madrid: Pearson Educación.

Universidad Nacional Autónoma de México. (s.f.). *Repositorio digital de la Facultad de Ingeniería*.
Obtenido de UNAM:
<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/175/A8%20Cap%C3%ADtulo%205.pdf?sequence=8>

W3C. (19 de Mayo de 2015). *Extensible Markup Language (XML)*. Obtenido de W3C:
<https://www.w3.org/XML/>

W3C. (s.f.). *Guía Breve de Servicios Web*. Obtenido de W3C:
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

Wikipedia. (s.f.). *Lenguaje de programación*. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n#Historia

Wikipedia. (s.f.). *Microsoft SQL Server*. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Microsoft_SQL_Server#Caracter.C3.ADsticas

Wikipedia. (s.f.). *Reacción en cadena de la polimerasa*. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Reacci%C3%B3n_en_cadena_de_la_polimerasa