

SISTEMATIZACIÓN DE LOS PROCESOS DE CREACIÓN DE MÓDULOS DE
FORMACIÓN VIRTUALES MEDIANTE LA HERRAMIENTA SIAF EN LA
FACULTAD DE INGENIERÍAS DE LA UNIVERSIDAD LIBRE SECCIONAL
PEREIRA

Investigador principal

Raúl Alberto Gaviria Valencia

Auxiliar de investigación

Mateo Serna Narváez

UNIVERSIDAD LIBRE, SECCIONAL PEREIRA – SEDE BELMONTE

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

2015

TABLA DE CONTENIDO

1. OBJETIVOS DEL PROYECTO	1
1.1 OBJETIVO GENERAL	1
1.2 OBJETIVOS ESPECÍFICOS	1
2. PLANTEAMIENTO DEL PROBLEMA	2
3. JUSTIFICACIÓN	3
4. MARCOS DE REFERENCIAS	4
4.1 MARCO CONCEPTUAL	4
4.1.1 Sistema de información.	4
4.1.2 Computación en la nube.	4
4.1.3 Patrón de arquitectura de software.	5
4.1.4 Framework.	5
4.2 MARCO TEÓRICO	6
4.2.1 Sistema de información.	6
4.2.2 Computación en la nube.	7
4.2.2.1 Diferencias con el software tradicional.	8
4.2.3 Diseño web adaptable.	9
4.2.4 Patrón de arquitectura de software.	10
4.2.5 Patrón de diseño: Modelo Vista Controlador (MVC)	13
4.2.6 Framework.	14
5. METODOLOGÍA	15
5.1 MODELO EN CASCADA	15
6. DOCUMENTACIÓN DE LOS CASOS DE USO	18
6.1 CREACIÓN DE MÓDULO	18
6.2 LISTAR MÓDULOS	20
6.3 MODIFICAR MÓDULOS	22
6.4 ELIMINAR MÓDULOS	24

6.5	CREACIÓN DE USUARIOS	26
6.6	LISTAR USUARIOS	28
6.7	MOFICIAR USUARIOS	30
6.8	ELIMINAR USUARIOS	32
6.9	CREACIÓN DE EJES TEMÁTICOS	34
6.10	LISTAR EJES TEMATICOS	36
6.11	MOFICIAR EJES TEMÁTICOS	38
6.12	ELIMINAR EJES TEMÁTICOS	40
6.13	CREACIÓN DE TIPOS DE RECURSOS	42
6.14	LISTAR TIPOS DE RECURSOS	44
6.15	MOFICIAR TIPOS DE RECURSOS	46
6.16	ELIMINAR TIPOS DE RECURSOS	48
6.17	CREACIÓN DE TIPOS DE ACTIVIDADES	50
6.18	LISTAR TIPOS DE ACTIVIDADES	52
6.19	MOFICIAR TIPOS DE ACTIVIDADES	54
6.20	ELIMINAR TIPOS DE ACTIVIDAD	56
6.21	CREACIÓN DE RECURSOS	58
6.22	LISTAR RECURSOS	60
6.23	MOFICIAR RECURSOS	62
6.24	ELIMINAR RECURSOS	64
6.25	REVISAR ETAPA	66
6.26	REVISAR MÓDULO	69
6.27	VER TRAZABILIDAD	72
6.28	EXPORTAR TRAZABILIDAD	74
6.29	SECCION – LOGIN	76
6.30	SECCION – LOGOUT	78
7.	DIAGRAMAS UML	80
7.1	DIAGRAMA DE CLASES	80
	Figura 35. Diagrama de clases	80
7.2	DIAGRAMA DE COMPONENTES	81

7.3	DIAGRAMA DE DESPLIEGUE	82
7.4	DIAGRAMA ENTIDAD – RELACIÓN	83
8.	CASOS DE PRUEBA	84
8.1	ALCANCE	84
8.2	ESTRATEGIA	84
8.3	CASOS DE PRUEBA	85
8.3.1	Listar módulos.	85
8.3.2	Crear módulos.	86
8.3.3	Modificar módulos.	87
8.3.4	Eliminar módulos.	89
8.3.5	Revisar módulo.	90
8.3.6	Ver trazabilidad.	93
8.3.7	Log in.	95
8.3.8	Log out.	97
	CONCLUSIONES	98
	RECOMENDACIONES	99
	BIBLIOGRAFIA	100

TABLA DE FIGURAS

FIGURA 1. COMPUTACIÓN EN LA NUBE	7
FIGURA 2. RESPONSIVE WEB DESIGN	9
FIGURA 3. MODELO MVC	13
FIGURA 4. MODELO EN CASCADA	16
FIGURA 5. DIAGRAMA DE SECUENCIA: CREAR MÓDULO	19
FIGURA 6. DIAGRAMA DE SECUENCIA: LISTAR MÓDULOS	21
FIGURA 7. DIAGRAMA DE SECUENCIA: MODIFICAR MÓDULOS	23
FIGURA 8. DIAGRAMA DE SECUENCIA: ELIMINAR MÓDULOS.	25
FIGURA 9. DIAGRAMA DE SECUENCIA: CREAR USUARIOS.	27
FIGURA 10. DIAGRAMA DE SECUENCIA: LISTAR USUARIOS	29
FIGURA 11. DIAGRAMA DE SECUENCIA: MODIFICAR USUARIOS	31
FIGURA 12. DIAGRAMA DE SECUENCIA: ELIMINAR USUARIOS	33
FIGURA 13. DIAGRAMA DE SECUENCIA: CREAR EJE TEMÁTICO	35
FIGURA 14. DIAGRAMA DE SECUENCIA: LISTAR EJES TEMÁTICOS	37
FIGURA 15. DIAGRAMA DE SECUENCIA: MODIFICAR EJES TEMÁTICOS	39
FIGURA 16. DIAGRAMA DE SECUENCIA: ELIMINAR EJE TEMÁTICO	41
FIGURA 17. DIAGRAMA DE SECUENCIA: CREAR TIPO DE RECURSO	43
FIGURA 18. DIAGRAMA DE SECUENCIA: LISTAR TIPO DE RECURSO	45
FIGURA 19. DIAGRAMA DE SECUENCIA: MOD. TIPOS DE RECURSOS	47
FIGURA 20. DIAGRAMA DE SECUENCIA: ELIMINAR TIPOS DE RECURSO	49
FIGURA 21. DIAGRAMA DE SECUENCIA: CREAR TIPO DE ACTIVIDAD	51
FIGURA 22. DIAGRAMA DE SECUENCIA: LISTAR TIPO DE ACTIVIDAD	53
FIGURA 23. DIAGRAMA DE SECUENCIA: MODIFICAR TIPO DE ACTIVIDAD	55
FIGURA 24. DIAGRAMA DE SECUENCIA: ELIMINAR TIPO DE ACTIVIDAD	57
FIGURA 25. DIAGRAMA DE SECUENCIA: CREAR RECURSO	59
FIGURA 26. DIAGRAMA DE SECUENCIA: LISTAR RECURSOS	61

FIGURA 27. DIAGRAMA DE SECUENCIA: MODIFICAR RECURSO	63
FIGURA 28. DIAGRAMA DE SECUENCIA: ELIMINAR RECURSO	65
FIGURA 29. DIAGRAMA DE SECUENCIA: REVISAR ETAPA	68
FIGURA 30. DIAGRAMA DE SECUENCIA: REVISAR MÓDULO	71
FIGURA 31. DIAGRAMA DE SECUENCIA: VER TRAZABILIDAD	73
FIGURA 32. DIAGRAMA DE SECUENCIA: EXPORTAR TRAZABILIDAD	75
FIGURA 33. DIAGRAMA DE SECUENCIA: SECCIÓN - LOGIN	77
FIGURA 34. DIAGRAMA DE SECUENCIA: SECCIÓN - LOGOUT	79
FIGURA 35. DIAGRAMA DE CLASES	80
FIGURA 36. DIAGRAMA DE COMPONENTES	81
FIGURA 37. DIAGRAMA DE DESPLIEGUE	82
FIGURA 38. DIAGRAMA ENTIDAD - RELACIÓN	83

TABLA DE ILUSTRACIONES

TABLA 1. CASO DE PRUEBA: LISTAR MÓDULOS	85
TABLA 2. CASO DE PRUEBA: CREAR MÓDULOS	86
TABLA 3. CASO DE PRUEBA: MODIFICAR MÓDULOS	87
TABLA 4. CASO DE PRUEBA: ELIMINAR MÓDULOS	89
TABLA 5. CASO DE PRUEBA: REVISAR MÓDULO	90
TABLA 6. CASO DE PRUEBA: VER TRAZABILIDAD	93
TABLA 7. CASO DE PRUEBA: LOGIN	95
TABLA 8. CASO DE PRUEBA: LOGOUT	97

1. OBJETIVOS DEL PROYECTO

1.1 OBJETIVO GENERAL

Diseñar un Software web para la sistematización de los procesos de creación de módulos de formación virtuales en la facultad de ingenierías de la Universidad Libre seccional Pereira.

1.2 OBJETIVOS ESPECÍFICOS

- Establecer los requerimientos del sistema de información.
- Diseñar la aplicación basado en patrones arquitectónicos con el cumplimiento de los requisitos establecidos.
- Realizar la codificación y pruebas del sistema de información para la sistematización de los procesos de creación de módulos de formación virtuales en la facultad de ingenierías de la Universidad Libre seccional Pereira.

2. PLANTEAMIENTO DEL PROBLEMA

En la última década la sistematización de la información a través del diseño de softwares especializados se ha convertido en un requerimiento de gestión de la calidad en todas las instituciones. En ese sentido la Universidad Libre de Pereira ha avanzado en procesos como el de gestión de procesos académicos a través de un sistema de información llamado SIUL el cual permite consultar todo lo relacionado con la matrícula académica, notas registradas, calendario académico, planes de estudio, etc.

Sin embargo, en los procesos de creación de los módulos de formación virtuales se han presentado algunas dificultades que obstaculizan la eficiencia y agilidad con la que se puedan llevar a cabo los sistemas de gestión de la información. En primer lugar, la revisión, y posterior aprobación de la información requerida para la apertura de los módulos de formación resulta dispendiosa ya que algunos archivos se presentan en físico y otros de manera digital. Como consecuencia de este sistema de recopilación de información, no es posible establecer datos estadísticos que arrojen información acerca de la trazabilidad de cada módulo.

3. JUSTIFICACIÓN

En la actualidad, con el auge del desarrollo tecnológico y con la incorporación de las tecnologías en los procesos administrativos; nos encontramos con la posibilidad de hacer uso de grandes plataformas que permiten agilizar el ingreso de grandes cantidades de información, así mismo como organizarla de tal manera que se pueda hacer un buen uso de ella.

De acuerdo a lo anterior y teniendo en cuenta que la Universidad Libre presenta falencias con respecto a la administración de módulos de formación virtual, la plataforma de administración de módulos SIAF (Sistema de Información para Asignaturas de Formación) formaría parte de una estrategia de desarrollo profesional de esta universidad cuyo propósito será que los expertos temáticos cuenten con un espacio donde puedan brindar una información concreta y precisa de los módulos de formación. A la vez que sería de gran utilidad al permitir generar datos importantes como la trazabilidad del módulo de formación virtual.

4. MARCOS DE REFERENCIAS

4.1 MARCO CONCEPTUAL

4.1.1 Sistema de información.

Puede definirse como un sistema de información al conjunto de elementos que recolectan, procesan y distribuyen información para posteriormente hacer uso de ella dentro de una compañía u organización.

De esta manera lo que se busca con la presente investigación es mejorar los procesos de creación de módulos de formación virtual dentro de la facultad de ingeniería de la Universidad Libre seccional Pereira, especialmente haciendo uso de la trazabilidad que conlleva la creación de los módulos anteriormente mencionados.

4.1.2 Computación en la nube.

La computación en la nube es un proyecto tecnológico que busca presentar sistemas computacionales e informativos a través de internet y que en la actualidad van en aumento, debido, entre otras cosas a la importancia y el vertiginoso crecimiento de las herramientas que nos brindan acceso a internet (Smartphones, Tablets, etc.). En este orden de ideas, se considera pertinente y necesario el desarrollo de una aplicación web que no solo cumpla con los requerimientos básicos si no que acople en su funcionamiento un sistema accesible y seguro para los usuarios.

Por consiguiente y teniendo en cuenta la accesibilidad como uno de los principales factores dentro de la investigación, es de gran importancia también, que la

aplicación web cuenta con un diseño adaptable. El mismo consiste en que no importa de qué dispositivo ingrese el usuario a la aplicación web el sistema responderá de manera efectiva adaptándose a el tamaño de la pantalla.

4.1.3 Patrón de arquitectura de software.

Se denomina patrón al modelo que sirve de muestra para dar solución a determinado problema utilizando siempre los mismos parámetros ya que estos se ajustan una y otra vez a la situación requerida.

Una de las principales razones para usar un patrón es que este disminuye notablemente el tiempo de trabajo y reduce el margen de error que pueda llegar a presentarse dentro de la aplicación. Existen muchos patrones de diseño, sin embargo se ha optado por utilizar el patrón de arquitectura de software modelo vista controlador (MVC) debido a la buena experiencia que se tiene con la herramienta y que como consecuencia conlleva a ganar mayor agilidad en el proceso de desarrollo de la aplicación web.

4.1.4 Framework.

Puede definirse como un boceto que permite la creación y posterior ejecución de una aplicación y que nos brinda además todas las ventajas que trae consigo la estandarización de los sistemas.

Se considera importante la implementación de esta herramienta debido a que logra acoplar de la mejor manera un patrón de diseño y diferentes componentes que acogen características comunes necesarias para el desarrollo de aplicaciones web:

- Seguridad (autenticación y autorización).
- Enrutamiento.
- Internacionalización y localización.

- Inyección de dependencias.
- Terminal.
- Formularios, entre otros.

4.2 MARCO TEÓRICO

4.2.1 Sistema de información.

“Un sistema de información se puede definir técnicamente como un conjunto de componentes relacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en una organización.¹” Para poder llevar a cabo estas funciones, los sistemas de información requieren realizar tres actividades principales:

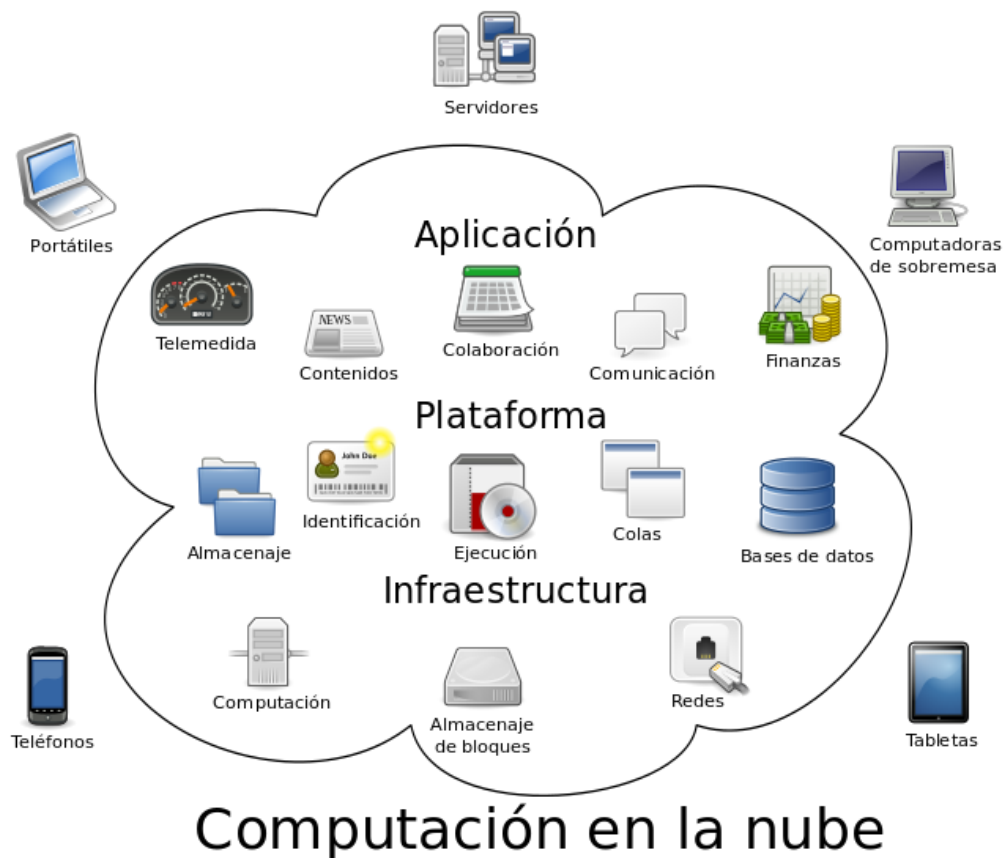
- **Entrada:** Es el primer paso que debe realizar un sistema de información y está relacionado con la toma o ingreso de datos, pueden ser manuales o automáticos. Los manuales son los ingresados directamente por un usuario del sistema mientras que los automáticos provienen normalmente de otros sistemas de información.
- **Procesamiento:** Como su nombre lo indica es la manera como un sistema de información puede procesar los datos ingresados o previamente almacenados y mediante una serie de pasos preestablecidos, convertirlos en información.
- **Salida:** Consiste en la forma como un sistema transfiere la información procesada o almacenada a un medio exterior o usuario final.

¹ Instituto Tecnológico de Sonora. Fuente:
<http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm>

4.2.2 Computación en la nube.

La computación en la nube, comúnmente también conocida como servicios en la nube (proveniente del término en inglés *cloud computing*) es una propuesta tecnológica que pretende ofrecer servicios computacionales o sistemas de información a través de internet. (ver figura 1)

Figura 1. Computación en la nube



Fuente <http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube#mediaviewer/Archivo:Cloud_computing-es.svg>

4.2.2.1 Diferencias con el software tradicional.

Son muchas las diferencias que existen entre el software en la nube y software tradicional, entre las más relevantes se tienen:

Software Tradicional²:

- Coste elevado de la licencia
- Se necesita instalar en el ordenador del cliente
- Costo por Soporte
- Costo por Actualizaciones
- Sin posibilidad de utilizar en otros dispositivos
- Sin copias de seguridad (de otro modo el cliente se encarga)
- Costos de implementación de hardware

Software en la nube³:

- Coste asequible, modelo por suscripción
- No se necesita instalar en el ordenador del cliente
- Sin coste de soporte
- Sin costo por actualizaciones
- Con posibilidad de acceso en cualquier dispositivo conectado a internet
- Con copias de seguridad automatizadas por el proveedor
- Sin costes por implementación de hardware (coste a cargo del proveedor)

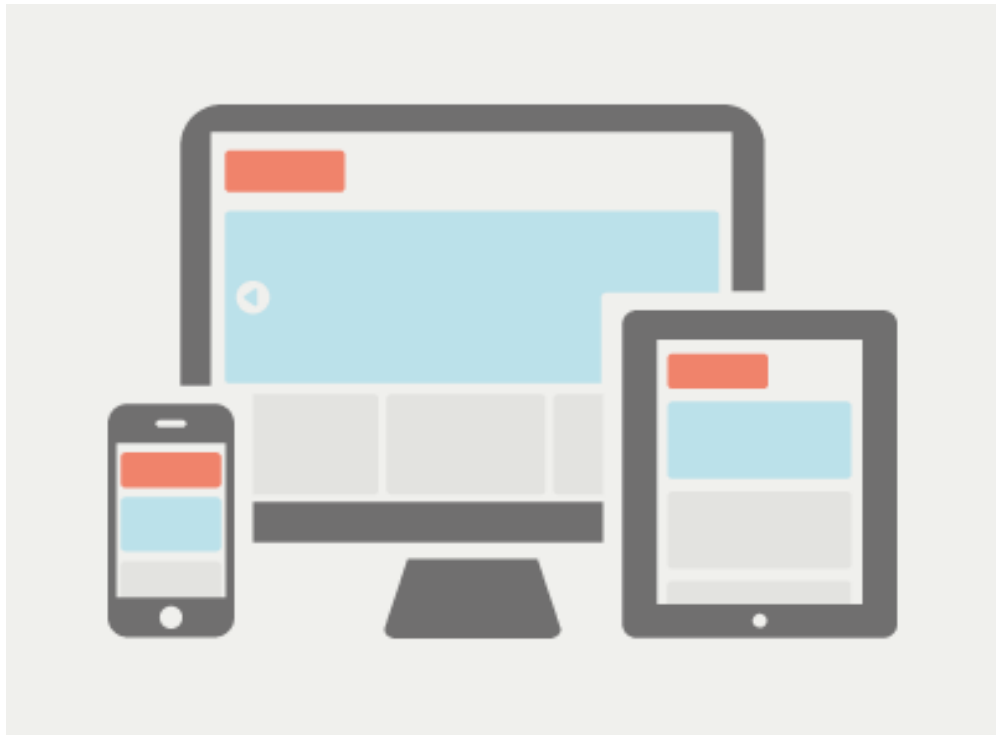
² cisnemedi. Fuente: <<http://cismnemedi.com/que-es-software-en-la-nube/>>

³ cisnemedi. Fuente: <<http://cismnemedi.com/que-es-software-en-la-nube/>>

4.2.3 Diseño web adaptable.

Actualmente, con el rápido avance de la tecnología y su afán por hacer más fácil la vida del hombre, los administradores y dueños de sitios web o sistemas de información en la nube, se ven ante la decisión de tener un sistema completamente accesible desde cualquier dispositivo tecnológico (Computador, Tablet, Smartphone, etc.) y es aquí donde el diseño web adaptable o *responsive web design* cumple un papel fundamental en este proceso, es decir, un sistema de información en la nube con diseño web adaptable significa que éste responderá de manera efectiva ante el tamaño de la pantalla del dispositivo desde donde se esté accediendo (ver figura 2).

Figura 2. Responsive web design



Fuente <<http://www.bossdigitalasia.com/wpcontent/uploads/2014/01/responsive-icon1.png>>

De esta manera el diseño web adaptable presenta muchas ventajas, entre ellas están:

- Adaptación para multi-dispositivos.
- Fácil mantenimiento.
- Mejoramiento de SEO (Search Engine Optimization).
- Incrementar ventas a través de dispositivos móviles.
- Mejorar la experiencia del usuario.

4.2.4 Patrón de arquitectura de software.

Se puede definir un patrón en un ámbito general como un “Conjunto de elementos que forman una unidad diferenciada y que se repiten a lo largo del tiempo, por lo que pueden tomarse como modelo o punto de referencia”⁴. O en otras palabras, como lo define Juan Pavón Mestras (Docente de la facultad de Informática de la Universidad Complutense Madrid – patrones de diseño orientado a objetos, 2004), un patrón es:

- una solución a un problema en un contexto particular.
- recurrente (lo que hace la solución relevante a otras situaciones).
- enseña (permite entender cómo adaptarlo a la variante particular del problema donde se quiere aplicar).
- tiene un nombre para referirse al patrón.

Aproximadamente en el año 1979 un arquitecto llamado Christopher Alexander publicó su libro *the timeless way of building* donde proponía usar patrones desarrollados por él y sus colegas para la construcción de edificaciones de mayor calidad, "Cada patrón describe un problema que ocurre infinidad de veces en nuestro entorno, así como la solución al mismo, de tal modo que podemos utilizar

⁴ The Free Dictionary. Fuente <<http://es.thefreedictionary.com/patr%C3%B3n>>

esta solución un millón de veces más adelante sin tener que volver a pensarla otra vez" (Christopher Alexander).

Años más tarde, Ward Cunningham y Kent Beck, perturbados por el no tan buen nivel de aprendizaje que tenían los nuevos programadores en el área de programación orientada a objetos (POO), buscaban una manera de compartir los conocimientos adquiridos por algunos a los programadores que recién estuviesen entrando en este paradigma de programación. Fue así, como encontraron información sobre Christopher Alexander e investigando acerca de él, se dieron cuenta que a pesar de las distintas áreas en las que se encontraban, existía una gran semejanza; decidieron tomar algunas ideas de las propuestas por Christopher Alexander y lograron desarrollar cinco patrones de interacción hombre-ordenador (HCI).

Sin embargo, fue hasta 1990 aproximadamente cuando el grupo GoF (Gang of Four, por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides) publicaron su libro llamado *Design Patterns: Elements of Reusable Object-Oriented Software*; y fue gracias a esta publicación que el uso de patrones de diseño en el área de la informática tuvo un gran éxito.

Ahora bien, los patrones de diseño están divididos en tres grandes categorías:

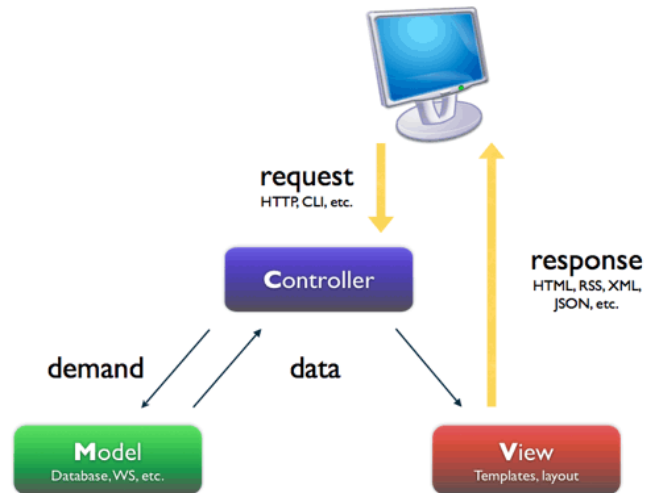
- **Patrones Estructurales:** relacionado con la composición de objetos, algunos son:
 - o Adapter.
 - o Bridge.
 - o Composite.
 - o Decorator.
 - o Facade.
 - o Flyweight.
 - o Proxy.

- **Patrones Creacionales:** Relacionado con el proceso de instanciar objetos, algunos son:
 - Object Pool.
 - Abstract Factory.
 - Builder.
 - Factory Method.
 - Singleton.
 - Model View Controller (MVC).

- **Patrones Comportamentales:** Relacionado con la comunicación de objetos, algunos son:
 - Command.
 - Interpreter.
 - Iterator.
 - Observer.
 - Template Method.
 - Visitor.

4.2.5 Patrón de diseño: Modelo Vista Controlador (MVC)

Figura 3. Modelo MVC



Fuente <http://symfony.com/legacy/doc/jobee/1_4/es/04?orm=Propel>

El Modelo Vista Controlador o MVC es un patrón de diseño de software que propone organizar el código de acuerdo a su naturaleza, es decir, crear tres componentes o tres capas distintas y muy sencillas de entender:

- El Modelo define la lógica de negocio (la base de datos pertenece a esta capa).
- La Vista es con lo que el usuario interactúa (un motor de plantillas es parte de esta capa).
- El Controlador es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la *Vista* para la presentación al cliente.

Aunque en sus inicios, el patrón de diseño MVC fue desarrollado para software de escritorio, con el paso del tiempo se ha ido adaptando al diseño, desarrollo e

implementación de aplicaciones web. Gracias a este hecho, se han desarrollado grandes cantidades de frameworks para la web, donde en la gran mayoría de los casos, un framework difiere de otro en la manera como los componentes y funciones del patrón de diseño MVC se dividen entre cliente y servidor.

4.2.6 Framework.

Un framework (Marco de trabajo), definiéndolo de forma sencilla, es un patrón, esquema o boceto, para el diseño, desarrollo e implementación de una aplicación y que ofrece todas las ventajas que tiene la estandarización, además de las suyas propias:

- Evita al programador tener que preocuparse por como estructurar la aplicación, el framework incorpora su propia estructura, de esta manera, es el programador el que se debe adaptar a ella.
- Desarrollos donde se requiera de múltiples programadores, un framework facilita enormemente la colaboración entre sí, esto gracias a la estandarización que obliga a manejar.
- Normalmente, cada framework tiene una comunidad aportando cosas nuevas cada día, esto facilita el uso de herramientas como librerías que agilizan y potencializan la aplicación.

En este orden de ideas, es claro que existen múltiples tipos de patrones de arquitectura o diseño de software, donde cada uno plantea su modo de trabajo, pero también pueden existir múltiples frameworks para cada patrón de diseño. Cada uno de estos frameworks además de adaptar el patrón de diseño a usar, también difieren entre sí, dependiendo del tipo de aplicaciones que quieran abarcar, es decir, que lenguaje de programación se implementará, que base de datos o qué tipo de servidor usará la aplicación, que tamaño tendrá, etc.

5. METODOLOGÍA

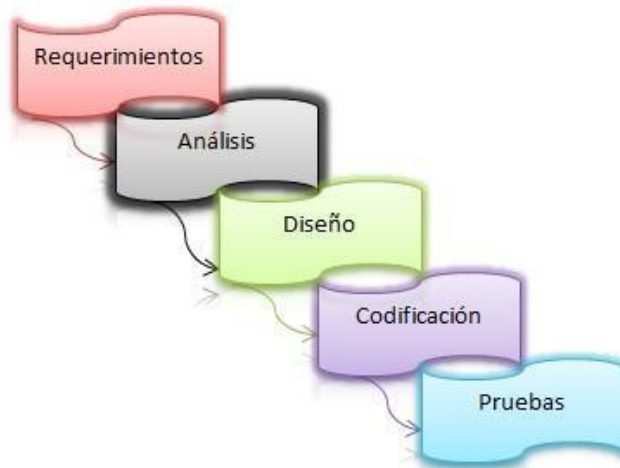
Se hará uso de la ingeniería de software como herramienta y disciplina dentro de las ciencias de la computación que presenta una serie de técnicas, métodos y conocimientos que permiten que se lleve a cabo la labor a través del constante seguimiento y desarrollo de las etapas dentro del proyecto de implementación de software.

El presente proyecto se desarrollará como una investigación aplicada dentro del marco metodológico, en otras palabras el Sistema de Información para Asignaturas de Formación (SIAF), se hará mediante una metodología de desarrollo de software con enfoque de modelo en cascada, que además, se complementará con un modelo iterativo en la etapa de Análisis de requerimientos, el cual consiste en empezar con una sencilla recolección de requerimientos del sistema e iterativamente ir mejorando cada componente del análisis, hasta tenerlo completo.

5.1 MODELO EN CASCADA

El desarrollo o modelo en cascada es un enfoque metodológico de la ingeniería de software que pretende ordenar las etapas del proceso de desarrollo de software y que se caracteriza principalmente porque el inicio de cada etapa está sujeta a la finalización de la etapa anterior. (ver figura 4)

Figura 4. Modelo en cascada



Fuente <<http://lscauml.wordpress.com/2012/01/05/disciplinas/>>

- **Requerimientos:** Se refiere a todos los requisitos que el usuario considera necesarios para el buen funcionamiento del software. Esta etapa se pretende realizar mediante reuniones con los usuarios finales de la aplicación.
- **Análisis:** En esta etapa se parte de toda la información recolectada en la fase de requerimientos con el fin de analizarla y así determinar el alcance de la aplicación. Se pretende implementar ésta etapa, haciendo uso del lenguaje unificado de modelado (UML) y de los tipos de diagramas más relevantes que ofrece esta herramienta.
- **Diseño:** Es la etapa encargada de definir tanto el diseño estético y visual como el diseño estructural de la aplicación. Para esta fase se hará uso de herramientas como el diseño web adaptable, plantillas administrativas, etc.

- **Codificación:** Es la etapa donde se empieza a desarrollar la aplicación mediante su código fuente, y para esto, se pretende usar todas las herramientas descritas en el marco teórico.
- **Prueba:** Una vez codificada la aplicación, la última fase del modelo, consiste en comprobar que todos los componentes de la aplicación funcionen correctamente y que cumplan con todos los requisitos.

6. DOCUMENTACIÓN DE LOS CASOS DE USO

6.1 CREACIÓN DE MÓDULO

Actores: Experto temático (iniciador).

Objetivo: Permitir al experto temático crear un nuevo modulo de formación en el sistema.

Descripción: El experto temático desea crear un nuevo modulo, el sistema deberá permitirselo e informar el estado final de esta operación.

Tipo: esencial y primario.

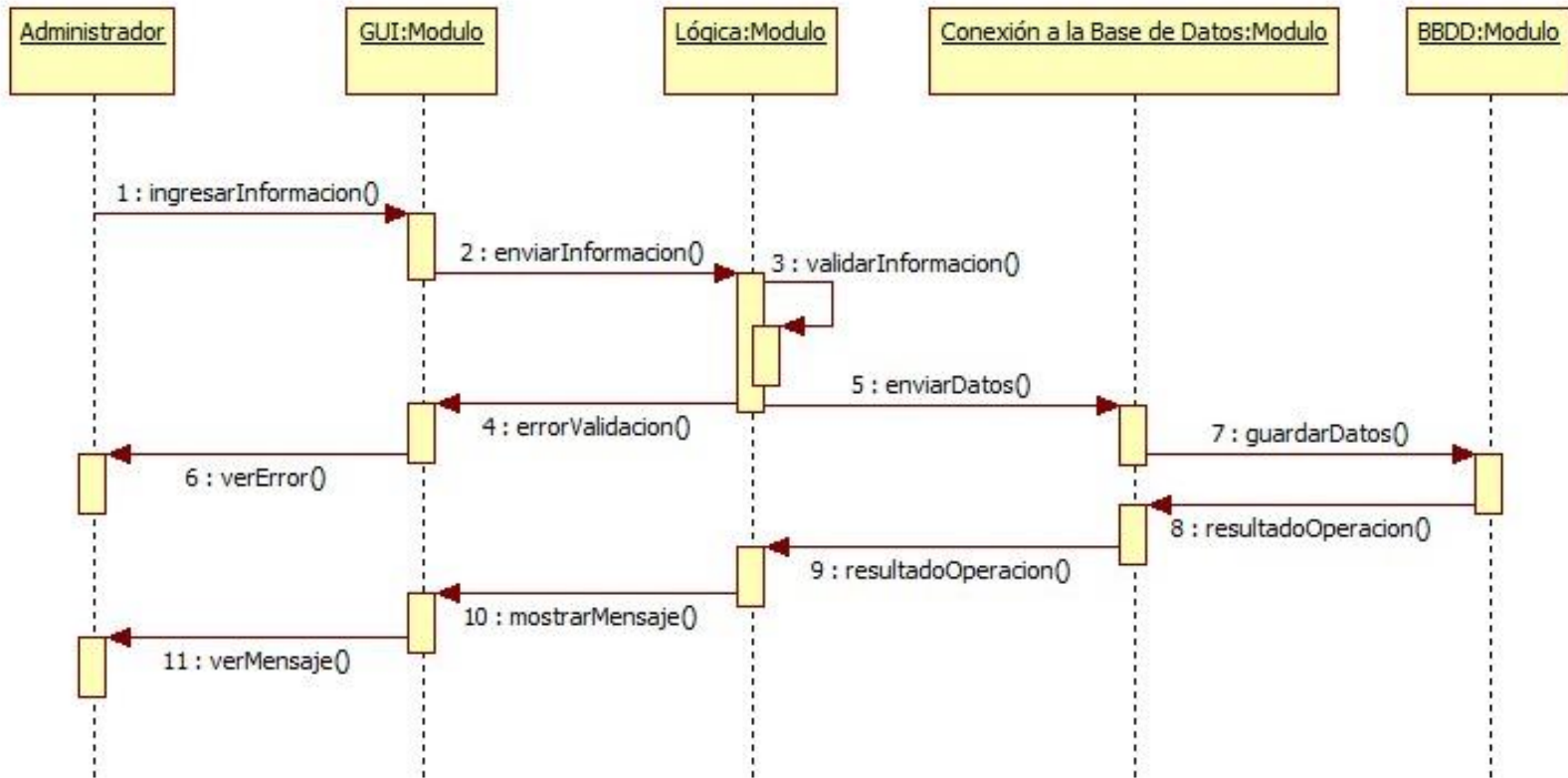
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea crear un nuevo modulo.	3. El sistema despliega el formulario con la información necesaria
4.El usuario llena el formulario	5. El sistema valida la información suministrada en el formulario
	6. El sistema informa el éxito de la operación.

Cursos alternos:

5. La validación de los datos introducidos en el formulario falló. El sistema informa al usuario y vuelve a la línea 3.

Figura 5. Diagrama de secuencia: crear módulo



6.2 LISTAR MÓDULOS

Actores: Experto temático (iniciador).

Objetivo: Realizar un listado detallado de los módulos de formación.

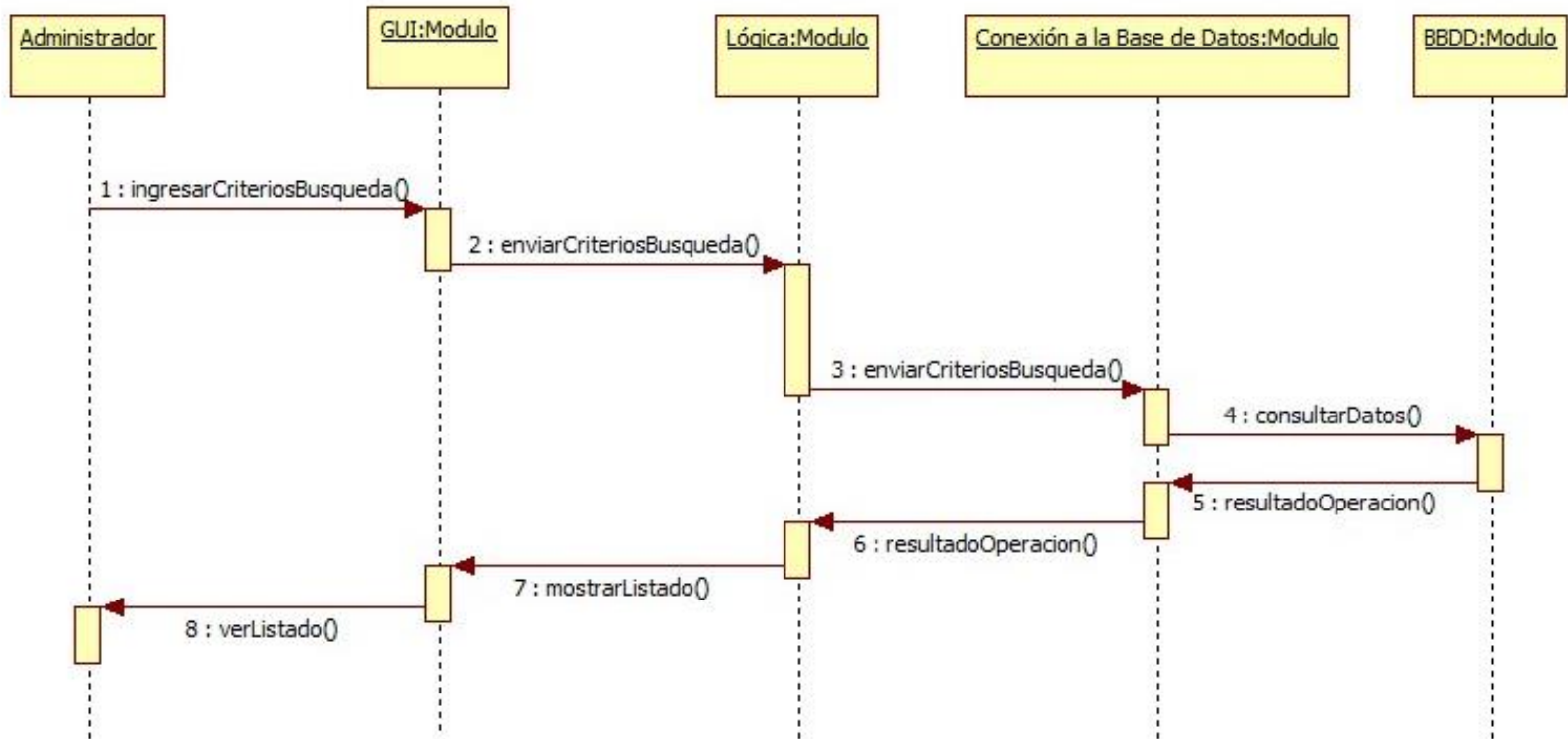
Descripción: el experto temático desea listar los módulos de formación, el sistema responde a esto con el respectivo listado detallado.

Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea realizar un listado de módulos.	3. El sistema le muestra los criterios de búsqueda.
4. El usuario elije.	5. El sistema muestra el listado según la elección del usuario.

Figura 6. Diagrama de secuencia: listar módulos



6.3 MODIFICAR MÓDULOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Realizar una modificación sobre los datos de algún modulo.

Descripción: el experto temático o docente desea modificar los datos de un módulo determinado, el sistema pide que seleccione el modulo, para posteriormente permitirle realizar dicha modificación.

Tipo: esencial y primario.

Curso normal de los eventos

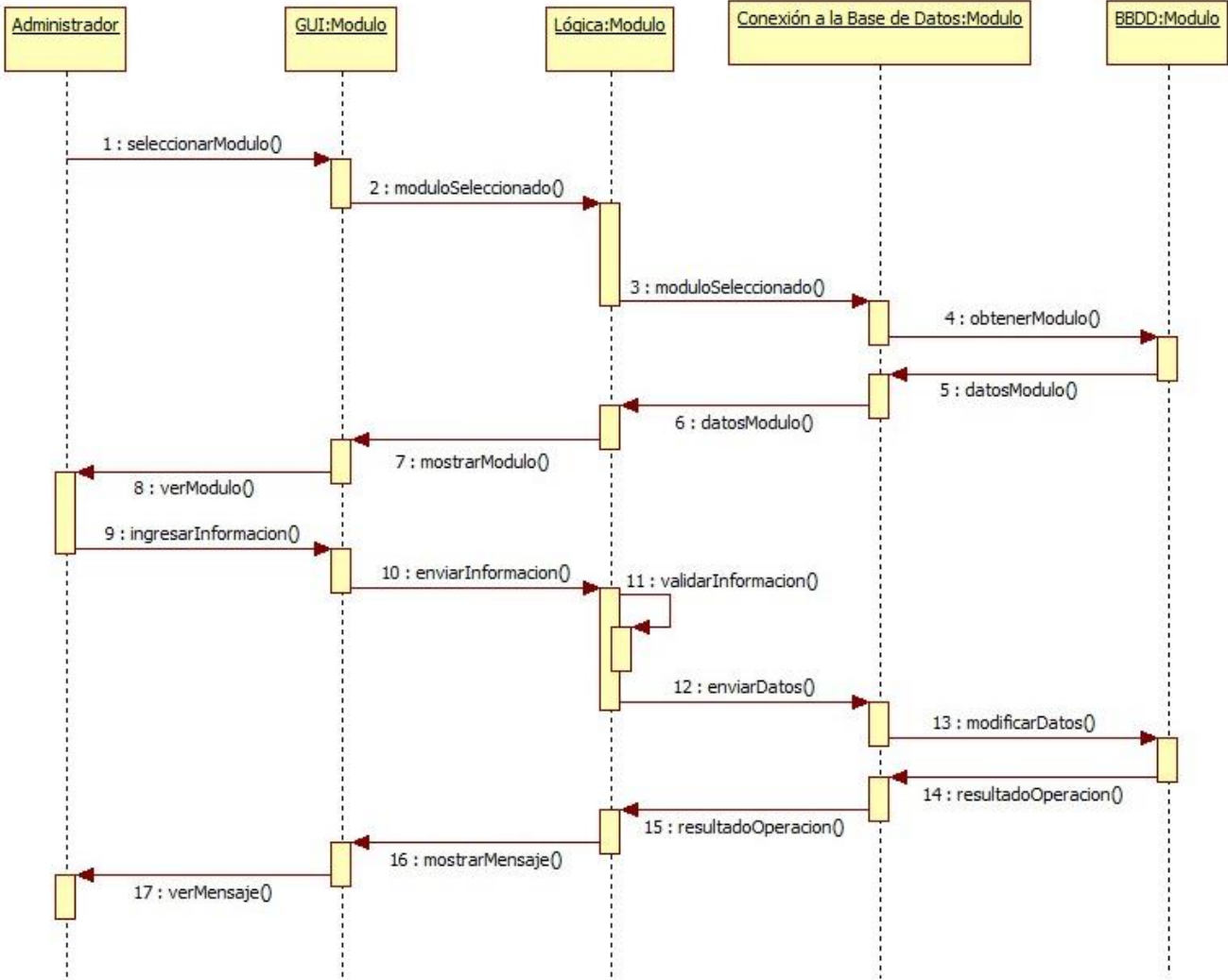
ACTOR	SISTEMA
1. Login (ver sección).	..

2. El usuario desea realizar una modificación sobre un módulo.	3. el sistema pide seleccionar el módulo.
4. El usuario elije.	.. 5. El sistema muestra el respectivo formulario de modificación de datos.
6. El introduce la información en el formulario.	... 7. El sistema valida la información suministrada.
	.. 8. El sistema informa el éxito de la operación.

Cursos alternos:

7. La validación de la información suministrada en el formulario falla. El sistema informa al usuario y vuelve a la línea 5.

Figura 7. Diagrama de secuencia: modificar módulos



6.4 ELIMINAR MÓDULOS

Actores: Experto temático (iniciador).

Objetivo: Eliminar algún módulo de formación.

Descripción: el experto temático desea eliminar algún módulo de formación determinado, el sistema pide que seleccione el modulo, para posteriormente permitirle realizar dicha acción.

Tipo: esencial y primario.

Curso normal de los eventos

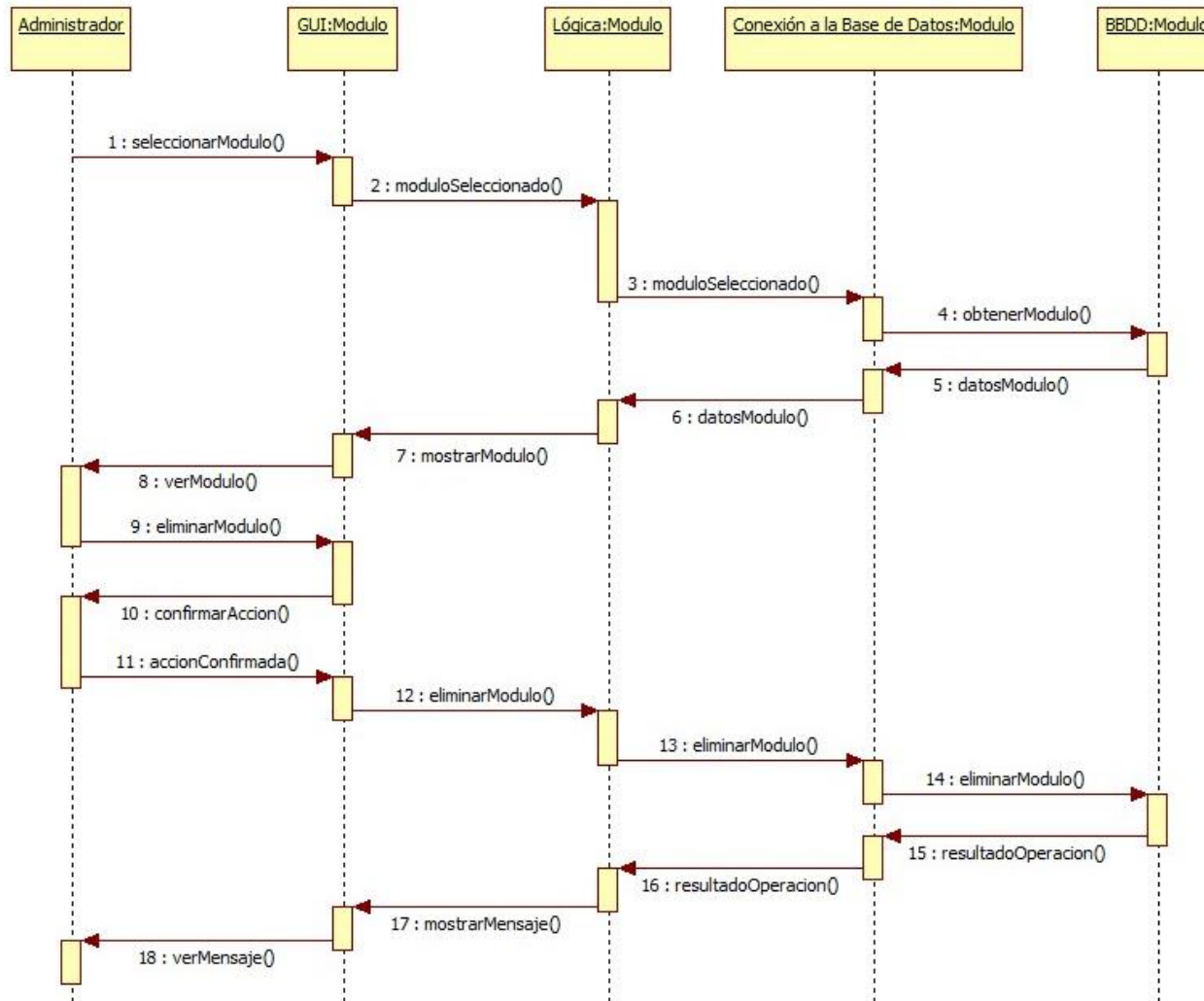
ACTOR	SISTEMA
1. Login (ver sección).	..

2. El usuario desea eliminar un módulo.	3. el sistema pide seleccionar el módulo.
4. El usuario elije.	.. 5. El sistema pide confirmación de la acción a realizar.
6. El usuario confirma.	.. 7. El sistema elimina el modulo del sistema.
	.. 8. El sistema informa el éxito de la operación.

Cursos alternos:

6. El usuario no confirma la acción a realizar. Vuelve a la línea 3.

Figura 8. Diagrama de secuencia: eliminar módulos.



6.5 CREACIÓN DE USUARIOS

Actores: Experto temático (iniciador).

Objetivo: Permitir al experto temático crear un nuevo usuario del sistema.

Descripción: El experto temático desea crear un nuevo usuario, el sistema deberá permitirselo e informar el estado final de esta operación.

Tipo: esencial y primario.

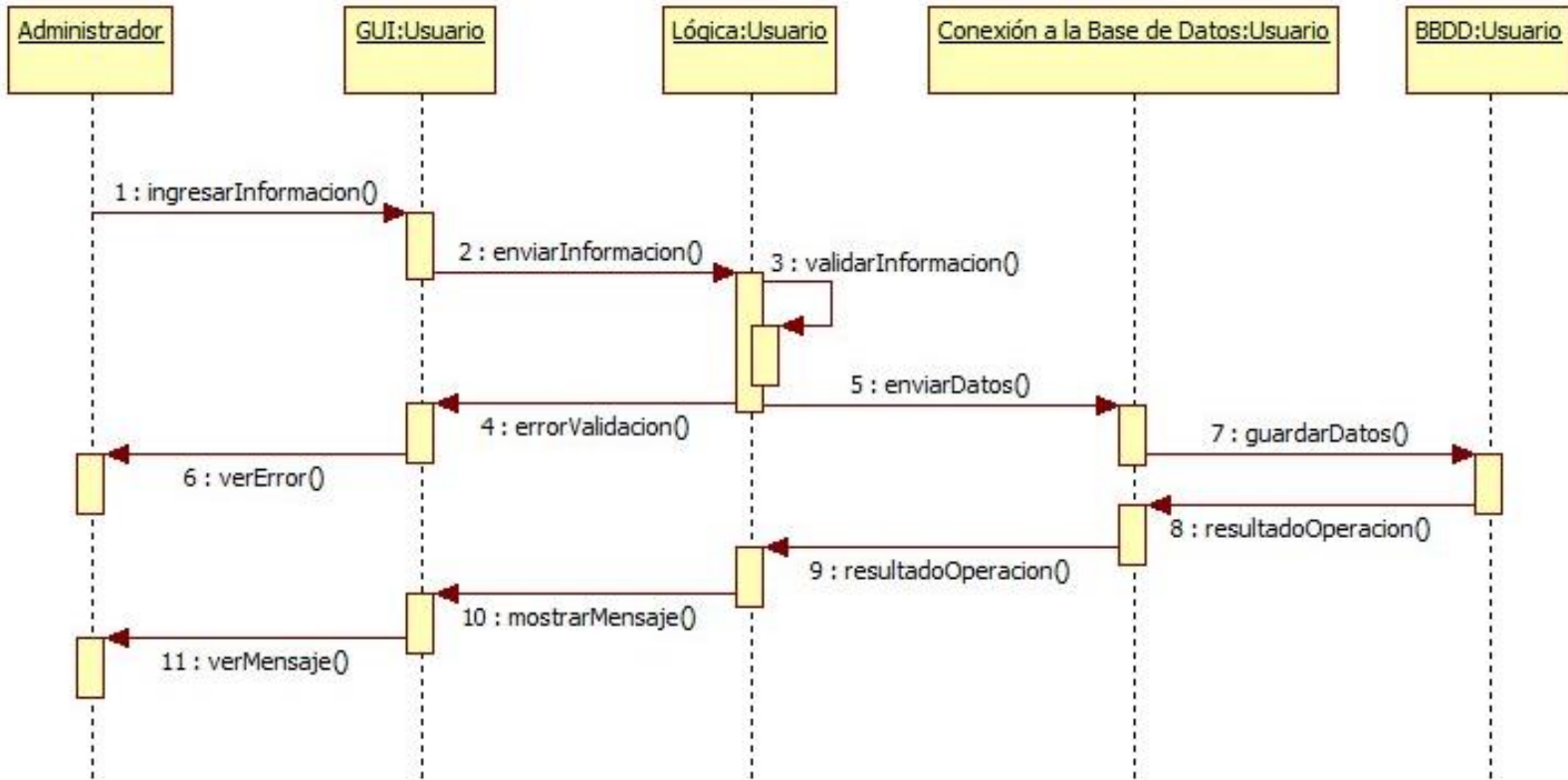
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea crear un nuevo usuario.	3. El sistema despliega el formulario con la información necesaria
4. El usuario llena el formulario.	5. El sistema valida la información suministrada en el formulario
	6. El sistema informa el éxito de la operación.

Cursos alternos:

5. La validación de los datos introducidos en el formulario falló. El sistema informa al usuario y vuelve a la línea 3.

Figura 9. Diagrama de secuencia: crear usuarios.



6.6 LISTAR USUARIOS

Actores: Experto temático (iniciador).

Objetivo: Realizar un listado detallado de los usuarios.

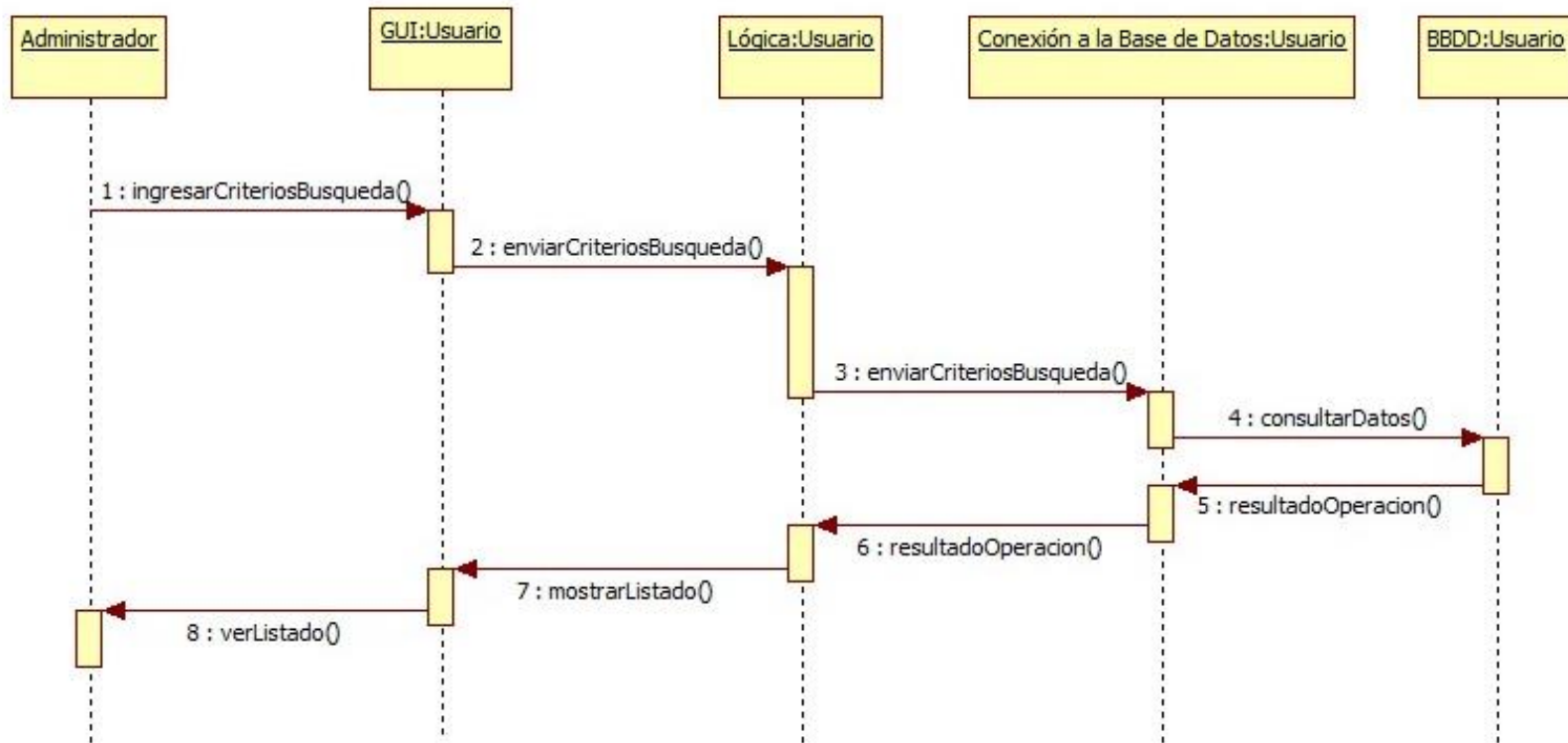
Descripción: El experto temático desea listar los usuarios, el sistema responde a esto con el respectivo listado detallado.

Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar un listado de usuarios.	2. El sistema le muestra los criterios de búsqueda.
3. El usuario elije.	4. El sistema muestra el listado según la elección del usuario.

Figura 10. Diagrama de secuencia: listar usuarios



6.7 MODIFICAR USUARIOS

Actores: Experto temático (iniciador).

Objetivo: Realizar una modificación sobre los datos de algún usuario.

Descripción: el experto temático desea modificar los datos de un usuario determinado, el sistema pide que seleccione el usuario, para posteriormente permitirle realizar dicha modificación.

Tipo: esencial y primario.

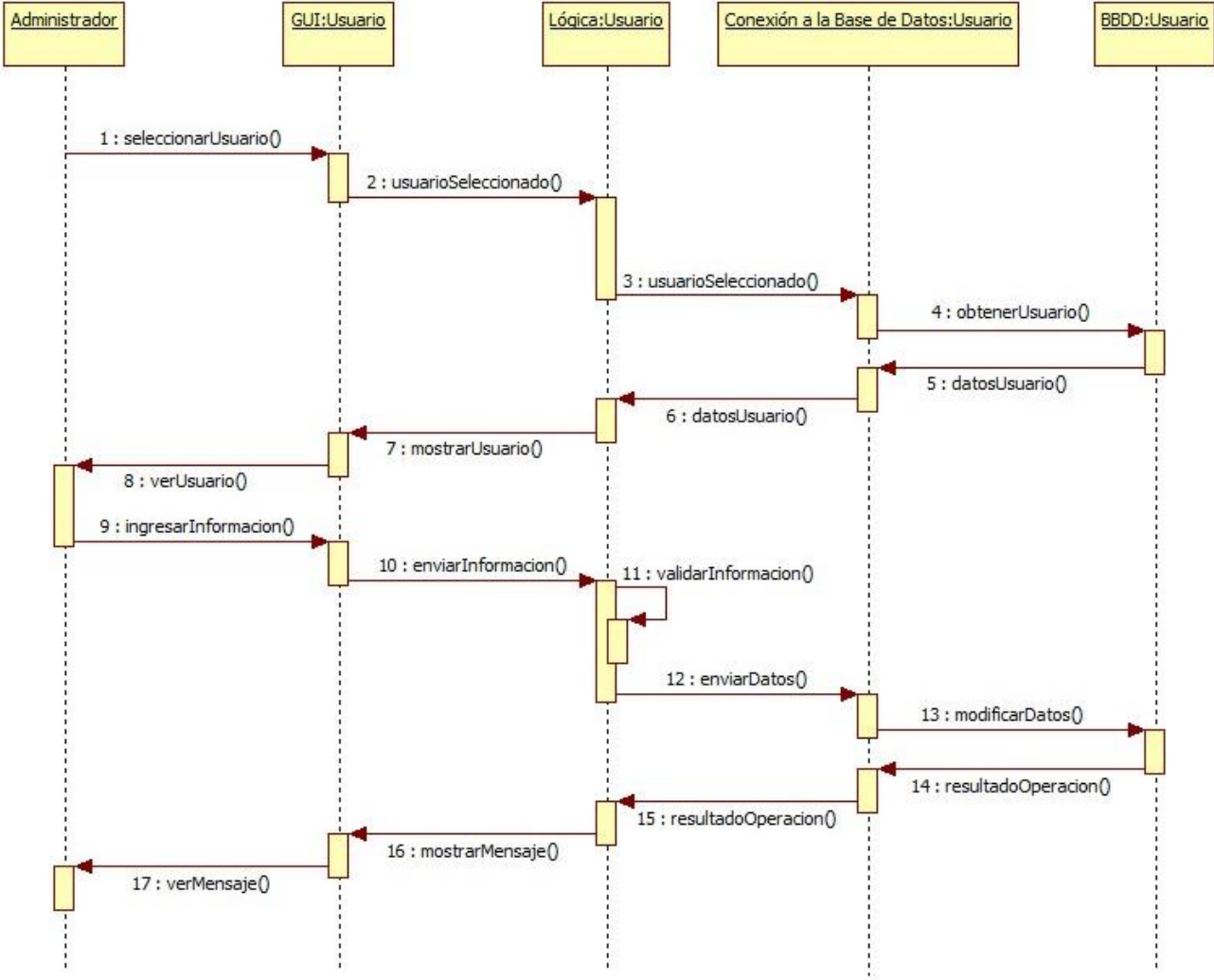
Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar una modificación sobre un usuario.	2. el sistema pide seleccionar el usuario.
3. El usuario elije.	4. El sistema muestra el respectivo formulario de modificación de datos.
5. El introduce la información en el formulario.	6. El sistema valida la información suministrada.
	7. El sistema informa el éxito de la operación.

Cursos alternos:

6. La validación de la información suministrada en el formulario falla. El sistema informa al usuario y vuelve a la línea 4.

Figura 11. Diagrama de secuencia: modificar usuarios



6.8 ELIMINAR USUARIOS

Actores: Experto temático (iniciador).

Objetivo: Eliminar algún usuario del sistema.

Descripción: el experto temático desea eliminar algún usuario del sistema, éste pide que seleccione el usuario, para posteriormente permitirle realizar dicha acción.

Tipo: esencial y primario.

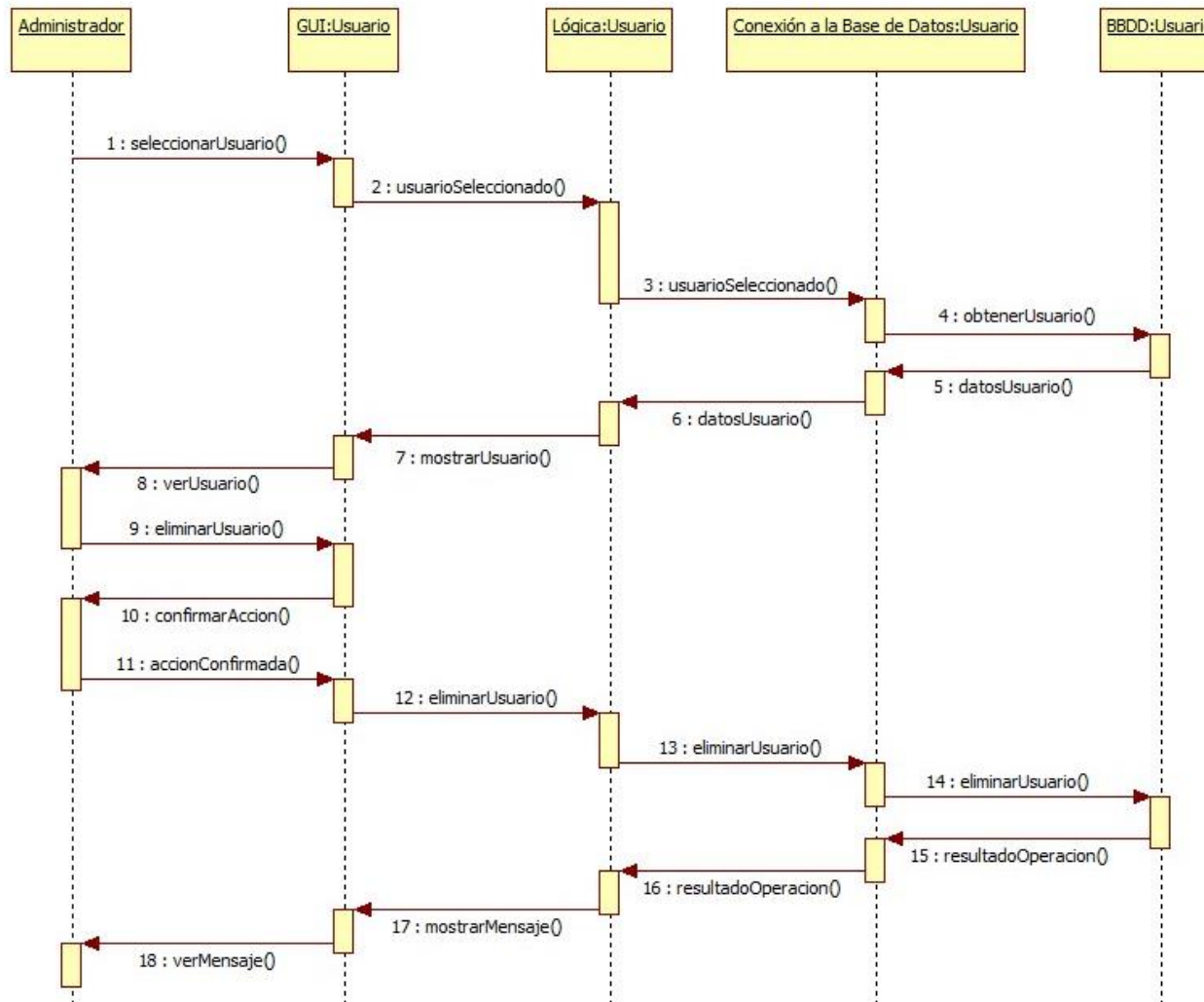
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).
2. El usuario desea eliminar un usuario.	3. el sistema pide seleccionar el usuario.
4. El usuario elije.	.. 5. El sistema pide confirmación de la acción a realizar.
6. El usuario confirma.	.. 7. El sistema elimina el usuario del sistema.
	.. 8. El sistema informa el éxito de la operación.

Cursos alternos:

6. El usuario no confirma la acción a realizar. Vuelve a la línea 3.

Figura 12. Diagrama de secuencia: eliminar usuarios



6.9 CREACIÓN DE EJES TEMÁTICOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Permitir al experto temático y al docente crear un nuevo eje temático perteneciente a un módulo.

Descripción: El experto temático o el docente desean crear un nuevo eje temático, el sistema deberá permitirselo e informar el estado final de esta operación.

Tipo: esencial y primario.

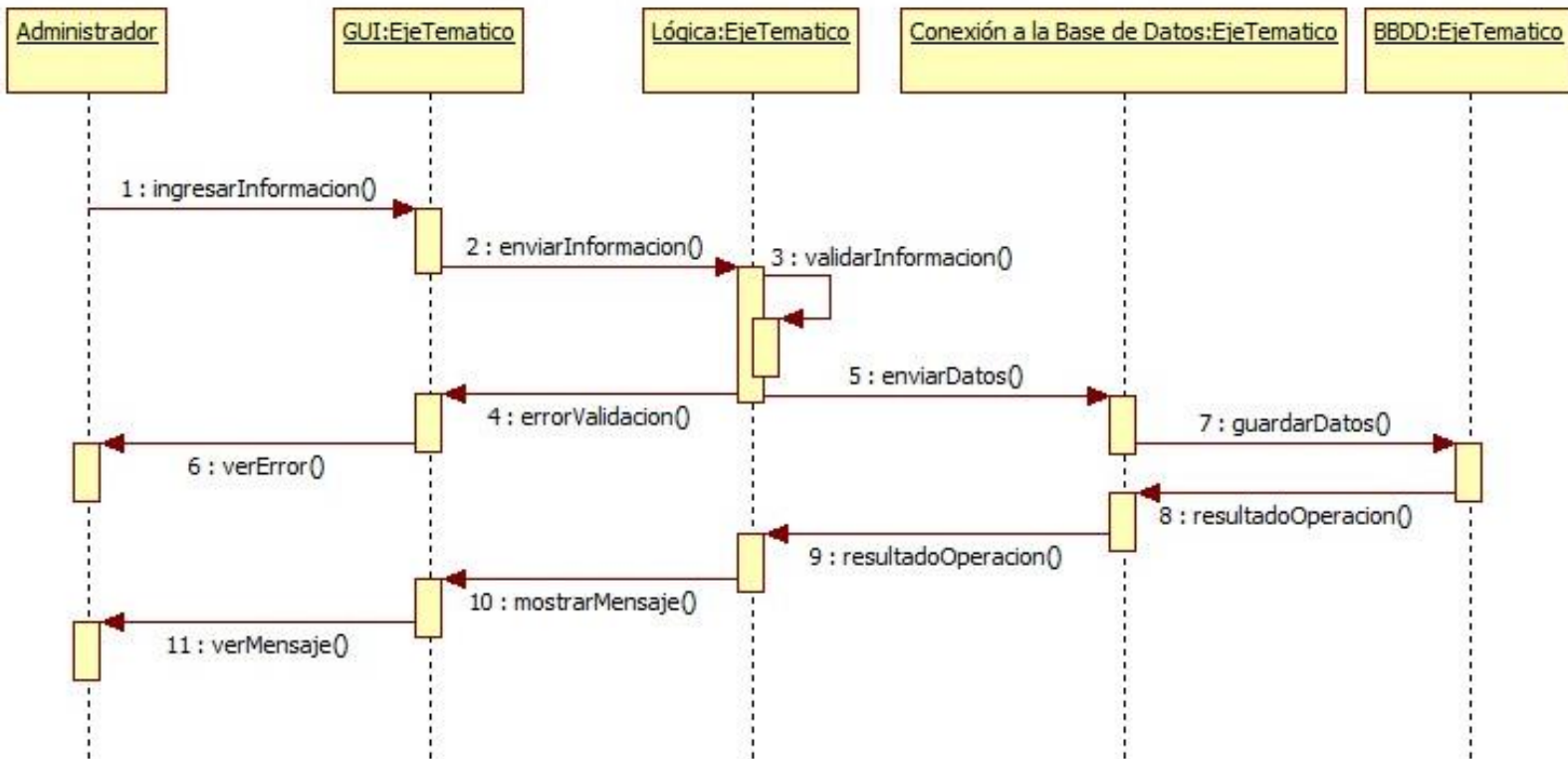
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	..
2. El usuario desea crear un nuevo eje temático.	3. El sistema despliega el formulario con la información necesaria
4. El usuario llena el formulario.	5. El sistema valida la información suministrada en el formulario
	6. El sistema informa el éxito de la operación.

Cursos alternos:

5. La validación de los datos introducidos en el formulario falló. El sistema informa al usuario y vuelve a la línea 3.

Figura 13. Diagrama de secuencia: crear eje temático



6.10 LISTAR EJES TEMATICOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Realizar un listado detallado de ejes temáticos de un módulo.

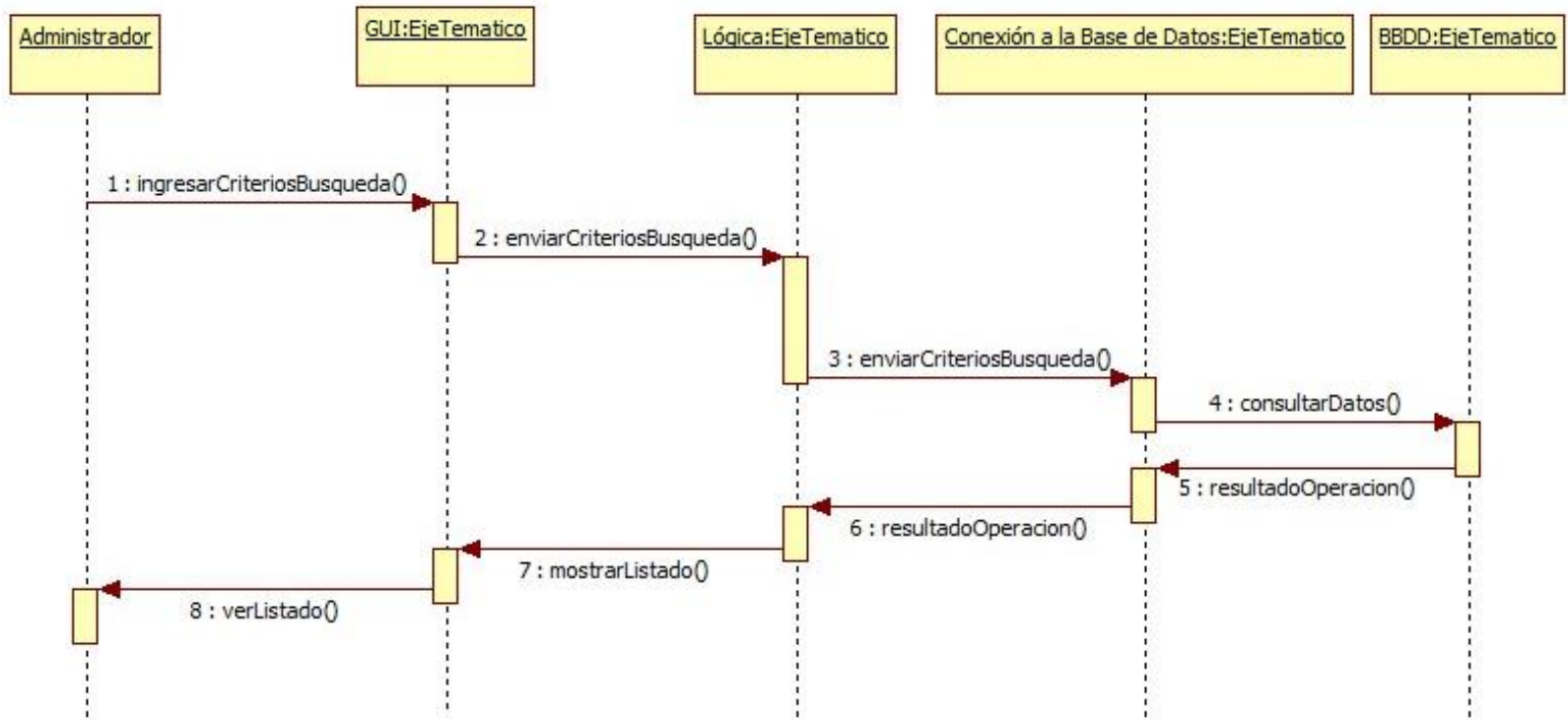
Descripción: El experto temático o docente desea listar los ejes temáticos, el sistema responde a esto con el respectivo listado detallado.

Tipo: esencial y primario.

Curso normal de los eventos

* ACTOR	* SISTEMA
1. El usuario desea realizar un listado de ejes temáticos.	2. El sistema le muestra los criterios de búsqueda, entre ellos un módulo en específico.
3. El usuario elije.	4. El sistema muestra el listado según la elección del usuario.

Figura 14. Diagrama de secuencia: listar ejes temáticos



6.11 MOFICIAR EJES TEMÁTICOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Realizar una modificación sobre algún eje temático.

Descripción: el experto temático o docente desean modificar los ejes temáticos de un módulo determinado, el sistema pide que seleccione el modulo, para posteriormente permitirle realizar dicha modificación.

Tipo: esencial y primario.

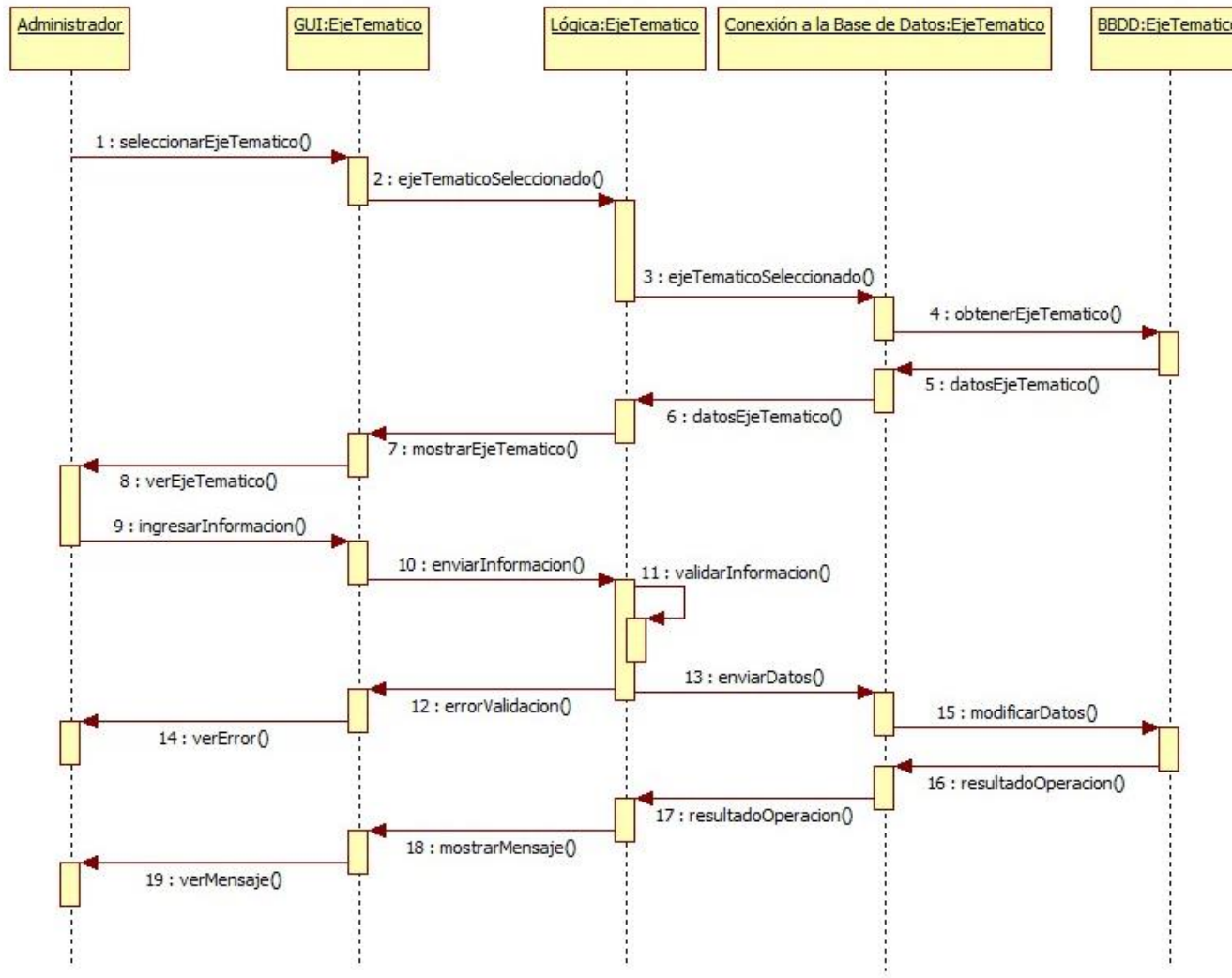
Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar una modificación sobre un eje temático.	2. el sistema pide seleccionar el módulo.
3. El usuario elije.	4. El sistema muestra el respectivo formulario de modificación de datos.
5. El usuario introduce la información en el formulario.	6. El sistema valida la información suministrada.
	7. El sistema informa el éxito de la operación.

Cursos alternos:

6. La validación de la información suministrada en el formulario falla. El sistema informa al usuario y vuelve a la línea 4.

Figura 15. Diagrama de secuencia: modificar ejes temáticos



6.12 ELIMINAR EJES TEMÁTICOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Eliminar algún eje temático del sistema.

Descripción: el experto temático o docente desea eliminar algún eje temático de un módulo determinado, el sistema pide que seleccione el módulo, para posteriormente permitirle realizar dicha acción.

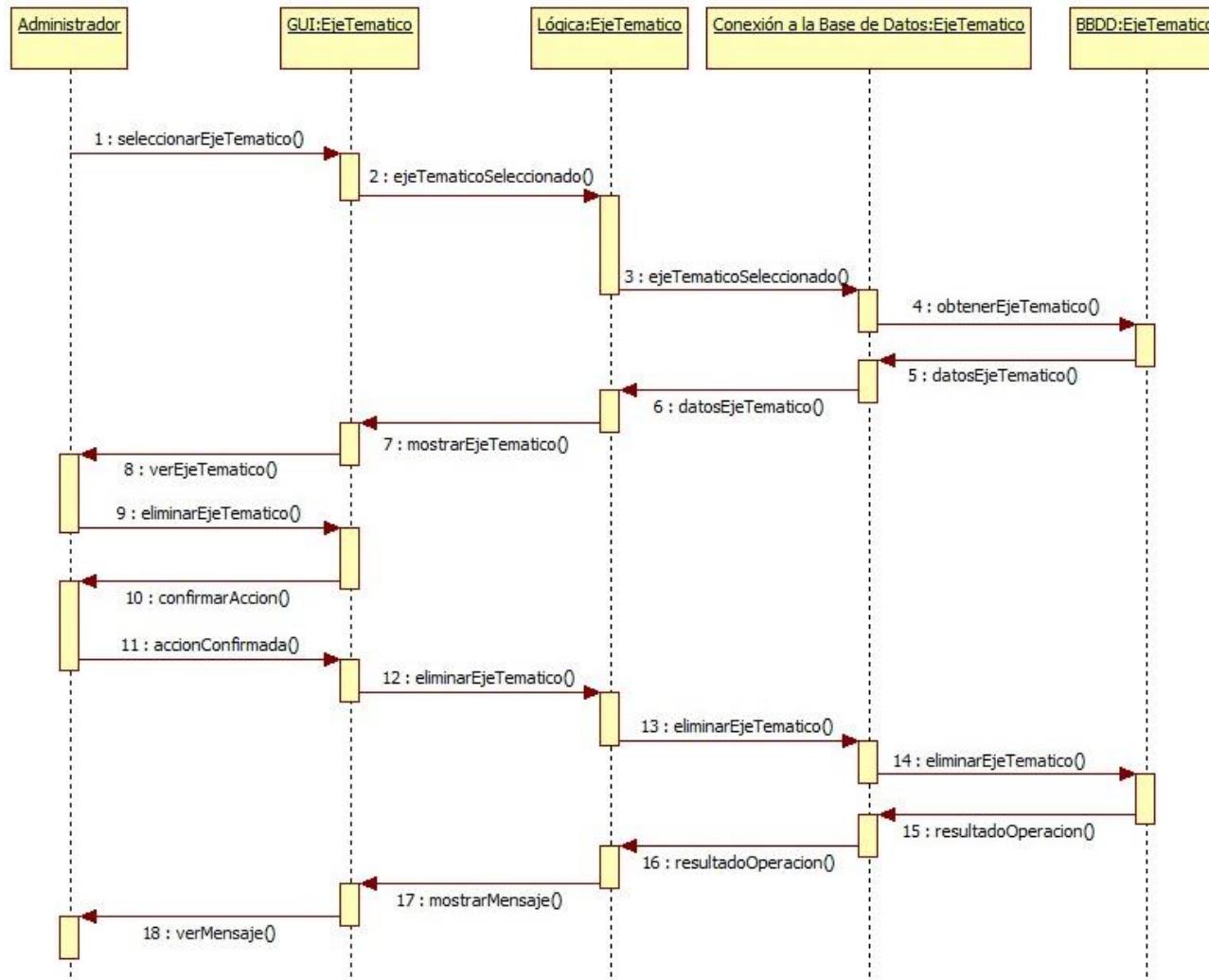
Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea eliminar un eje temático.	3. el sistema pide seleccionar el modulo.
4. El usuario elije.	5. El sistema pide seleccionar un eje temático.
6. El usuario elije.	6. El sistema pide confirmación de la acción a realizar.
7. El usuario confirma.	7. El sistema elimina el eje temático del sistema.
	9. El sistema informa el éxito de la operación.

Cursos alternos: 6. El usuario no confirma la acción a realizar. Vuelve a la línea 5.

Figura 16. Diagrama de secuencia: eliminar eje temático



6.13 CREACIÓN DE TIPOS DE RECURSOS

Actores: Experto temático (iniciador).

Objetivo: Permitir al experto temático crear un nuevo tipo de recurso.

Descripción: El experto temático desea crear un nuevo tipo de recurso, el sistema deberá permitirle e informar el estado final de esta operación.

Tipo: esencial y primario.

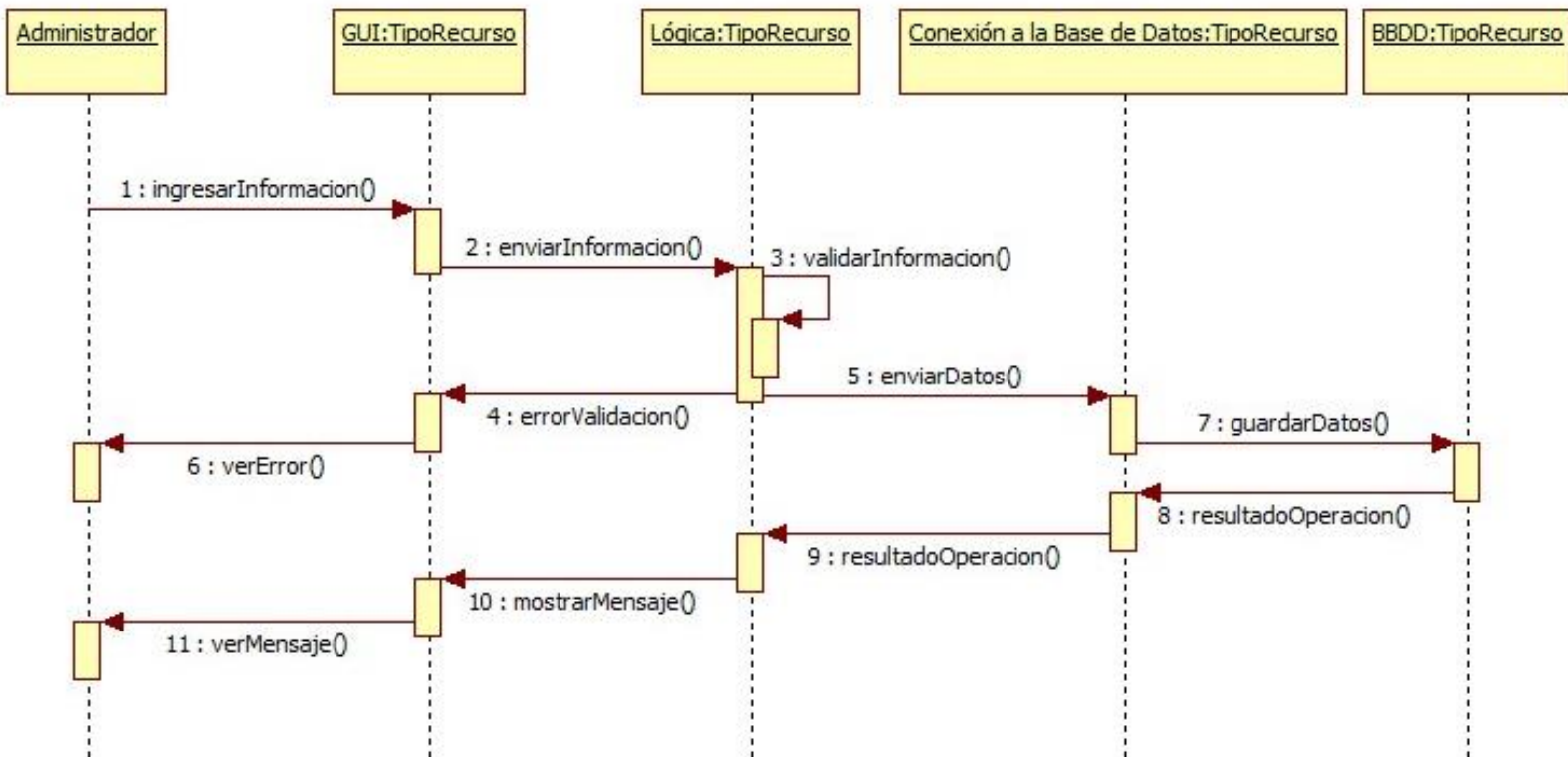
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	..
2. El usuario desea crear un nuevo tipo de recurso.	3. El sistema despliega el formulario con la información necesaria
4. El usuario llena el formulario.	.. 5. El sistema valida la información suministrada en el formulario
	.. 6. El sistema informa el éxito de la operación.

Cursos alternos:

5. La validación de los datos introducidos en el formulario falló. El sistema informa al usuario y vuelve a la línea 3.

Figura 17. Diagrama de secuencia: crear tipo de recurso



6.14 LISTAR TIPOS DE RECURSOS

Actores: Experto temático (iniciador).

Objetivo: Realizar un listado detallado de los tipos de recursos.

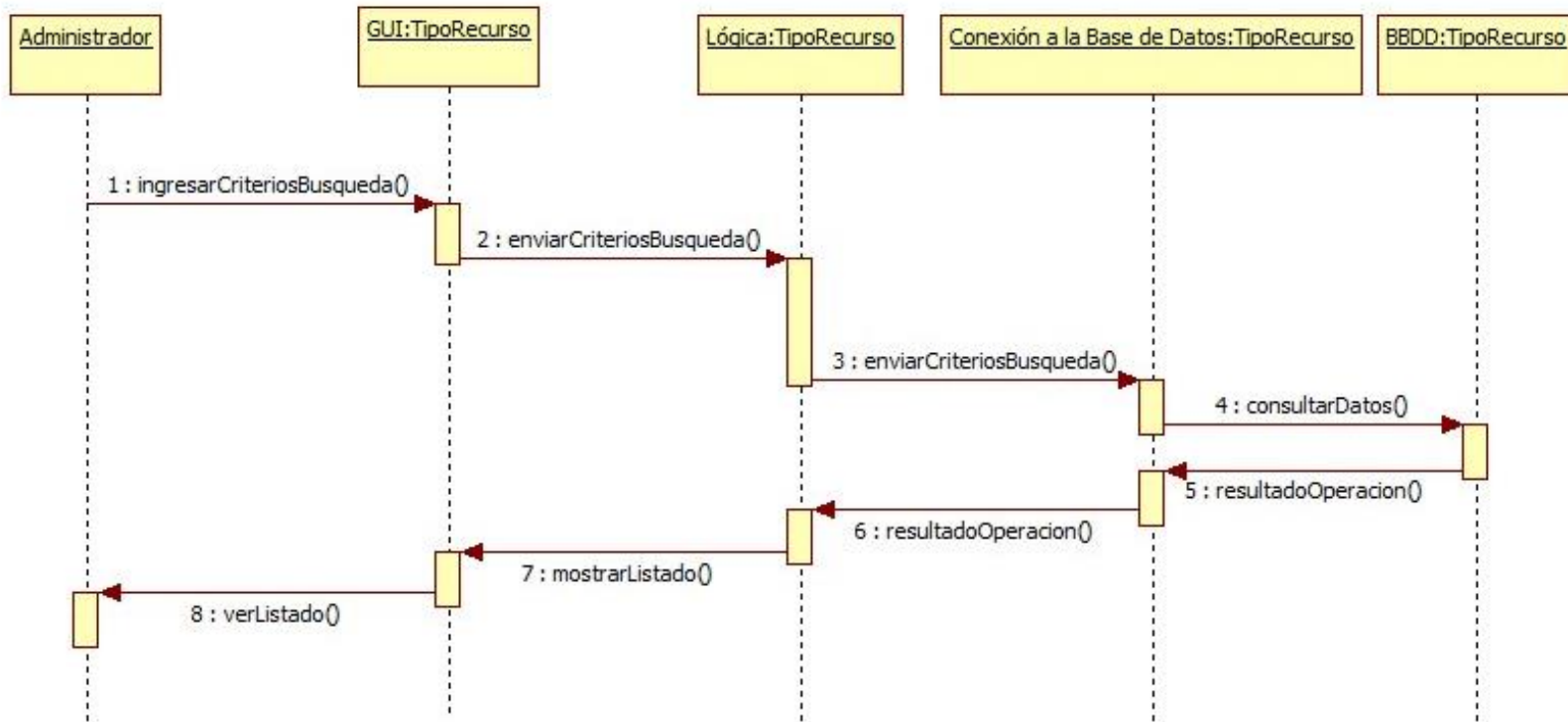
Descripción: El experto temático desea listar los tipos de recursos, el sistema responde a esto con el respectivo listado detallado.

Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar un listado de tipos de recursos.	2. El sistema le muestra los criterios de búsqueda.
3. El usuario elije.	4. El sistema muestra el listado según la elección del usuario.

Figura 18. Diagrama de secuencia: listar tipo de recurso



6.15 MOFICIAR TIPOS DE RECURSOS

Actores: Experto temático (iniciador).

Objetivo: Realizar una modificación sobre algún tipo de recurso.

Descripción: el experto temático desea modificar los tipos de recursos, el sistema pide que seleccione el tipo de recurso, para posteriormente permitirle realizar dicha modificación.

Tipo: esencial y primario.

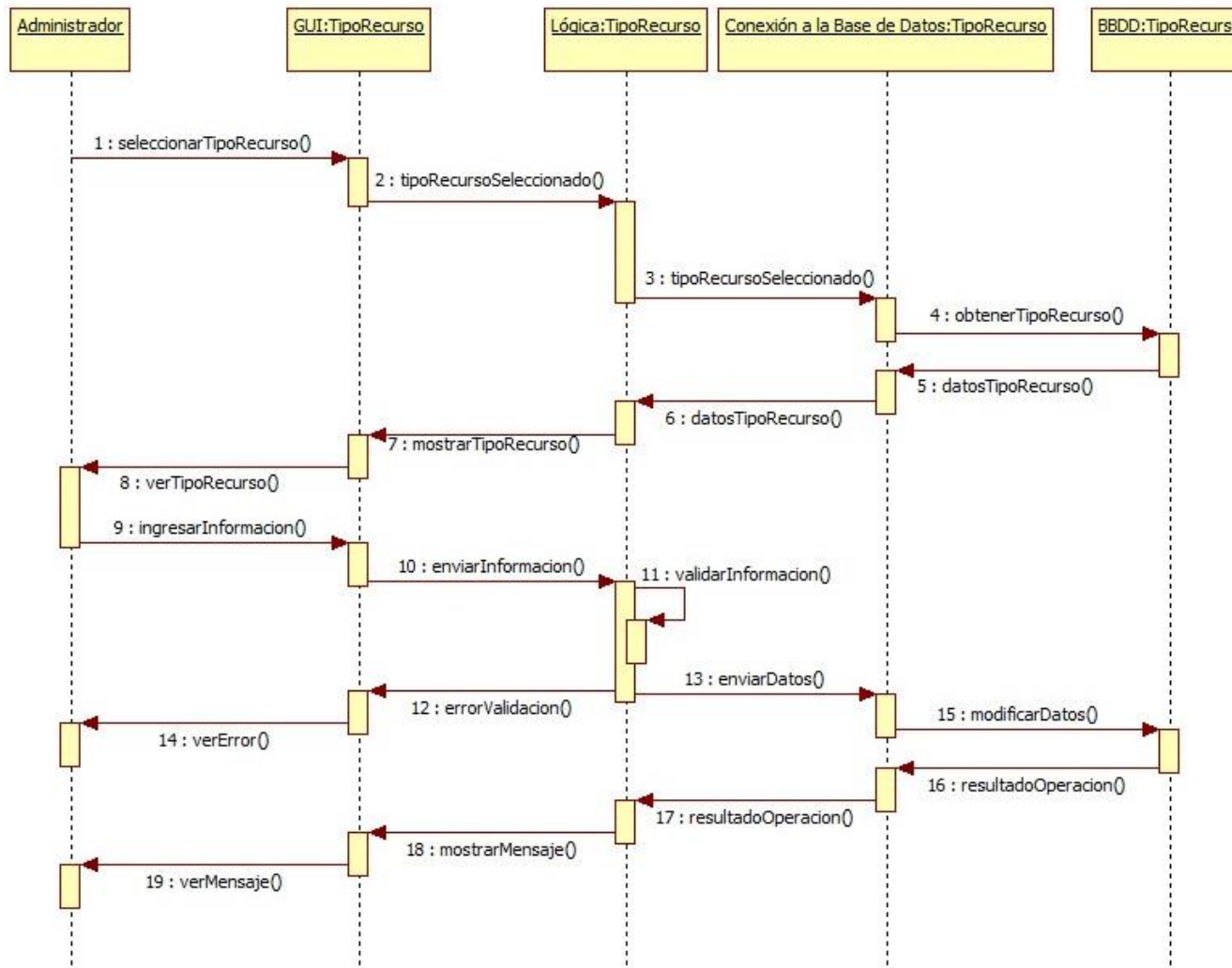
Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar una modificación sobre un tipo de recurso.	2. El sistema muestra el respectivo formulario de modificación de datos.
3. El usuario introduce la información en el formulario.	4. El sistema valida la información suministrada.
	6. El sistema informa el éxito de la operación.

Cursos alternos

6. La validación de la información suministrada en el formulario falla. El sistema informa al usuario y vuelve a la línea 2.

Figura 19. Diagrama de secuencia: modificar tipos de recursos



6.16 ELIMINAR TIPOS DE RECURSOS

Actores: Experto temático (iniciador).

Objetivo: Eliminar algún tipo de recurso del sistema.

Descripción: el experto temático desea eliminar algún tipo de recurso, el sistema pide que seleccione el tipo de recurso, para posteriormente permitirle realizar dicha acción.

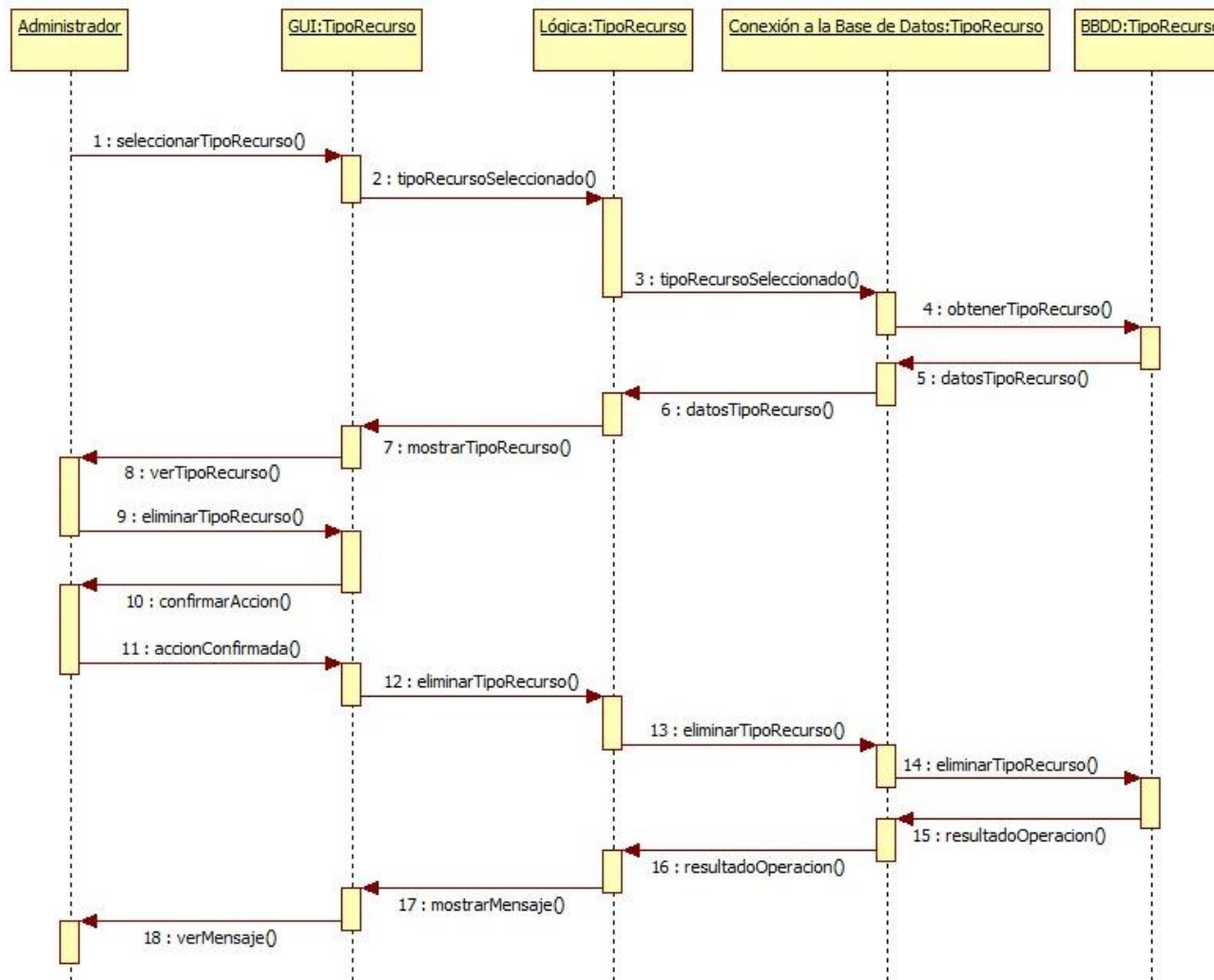
Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea eliminar un tipo de recurso.	3. el sistema pide seleccionar el tipo de recurso.
4. El usuario elije.	5. El sistema pide confirmación de la acción a realizar.
6. El usuario confirma.	7. El sistema elimina el tipo de recurso del sistema.
	8. El sistema informa el éxito de la operación.

Cursos alternos: 6. El usuario no confirma la acción a realizar. Vuelve a la línea 3

Figura 20. Diagrama de secuencia: eliminar tipos de recurso



6.17 CREACIÓN DE TIPOS DE ACTIVIDADES

Actores: Experto temático (iniciador).

Objetivo: Permitir al experto temático crear un nuevo tipo de actividad.

Descripción: El experto temático desea crear un nuevo tipo de actividad, el sistema deberá permitirselo e informar el estado final de esta operación.

Tipo: esencial y primario.

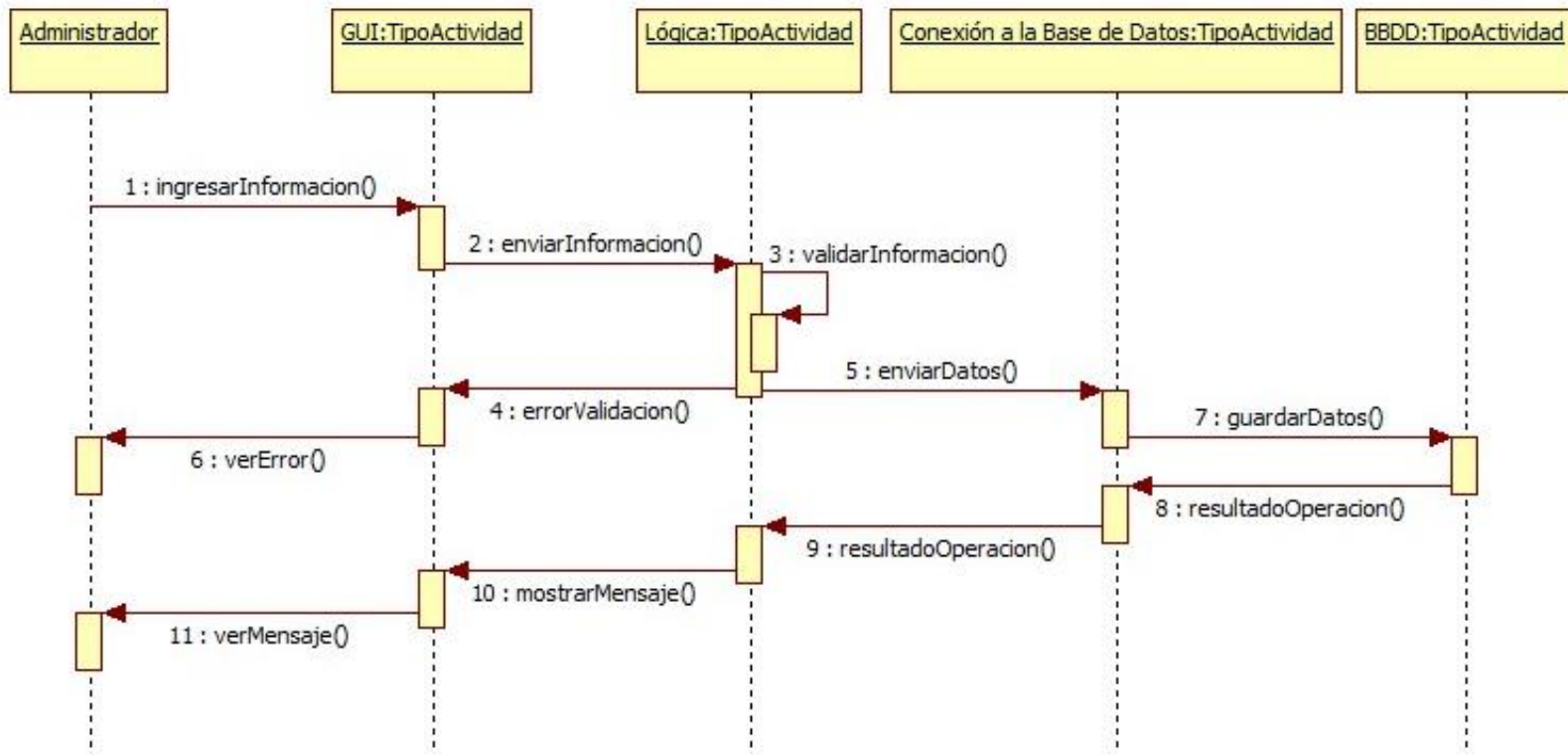
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	..
2. El usuario desea crear un nuevo tipo de actividad.	3. El sistema despliega el formulario con la información necesaria
4. El usuario llena el formulario.	.. 5. El sistema valida la información suministrada en el formulario
	.. 6. El sistema informa el éxito de la operación.

Cursos alternos

5. La validación de los datos introducidos en el formulario falló. El sistema informa al usuario y vuelve a la línea 3.

Figura 21. Diagrama de secuencia: crear tipo de actividad



6.18 LISTAR TIPOS DE ACTIVIDADES

Actores: Experto temático (iniciador).

Objetivo: Realizar un listado detallado de los tipos de actividades.

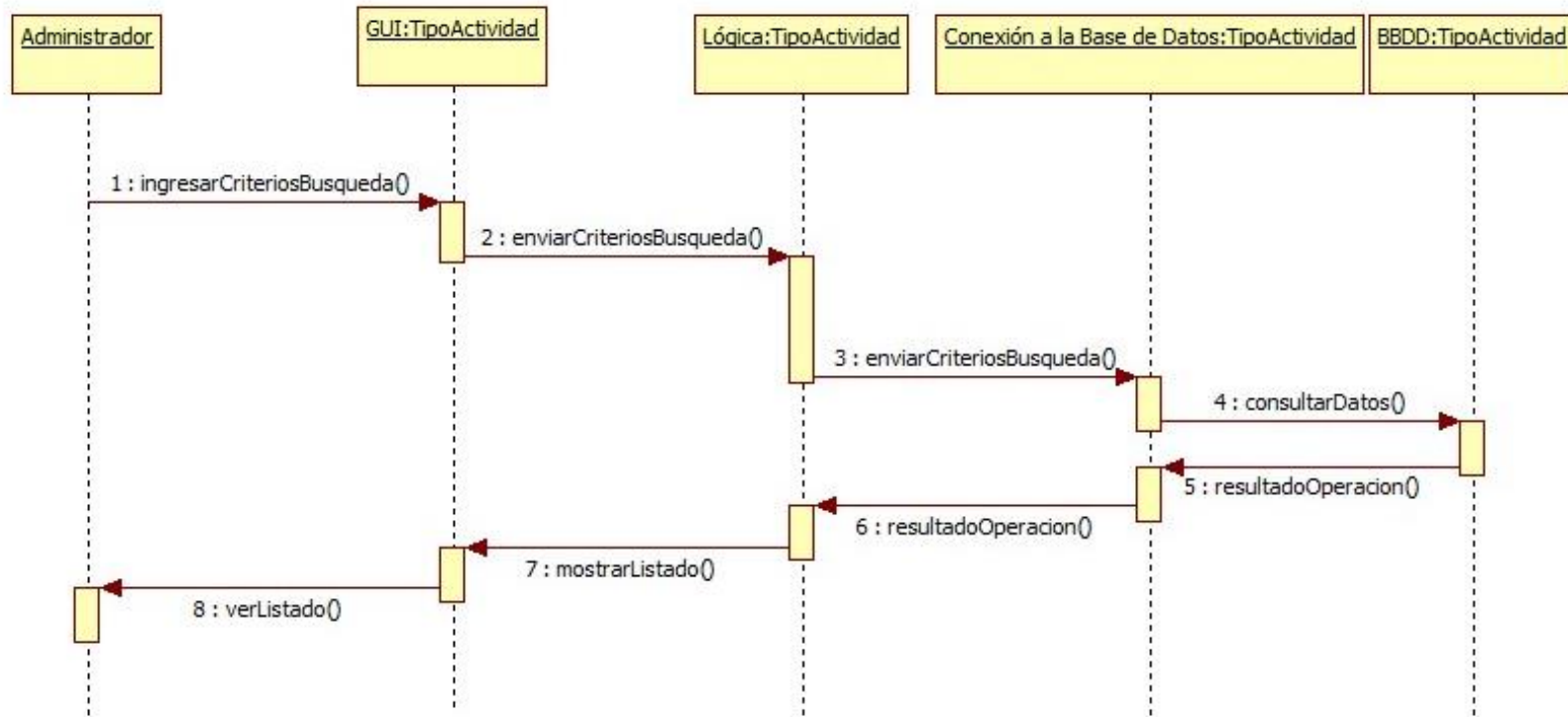
Descripción: El experto temático desea listar los tipos de actividades, el sistema responde a esto con el respectivo listado detallado.

Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar un listado de tipos de actividades.	2. El sistema le muestra los criterios de búsqueda.
3. El usuario elije.	4. El sistema muestra el listado según la elección del usuario.

Figura 22. Diagrama de secuencia: listar tipo de actividad



6.19 MOFICIAR TIPOS DE ACTIVIDADES

Actores: Experto temático (iniciador).

Objetivo: Realizar una modificación sobre algún tipo de actividad.

Descripción: el experto temático desea modificar los tipos de actividades, el sistema pide que seleccione el tipo de actividad, para posteriormente permitirle realizar dicha modificación.

Tipo: esencial y primario.

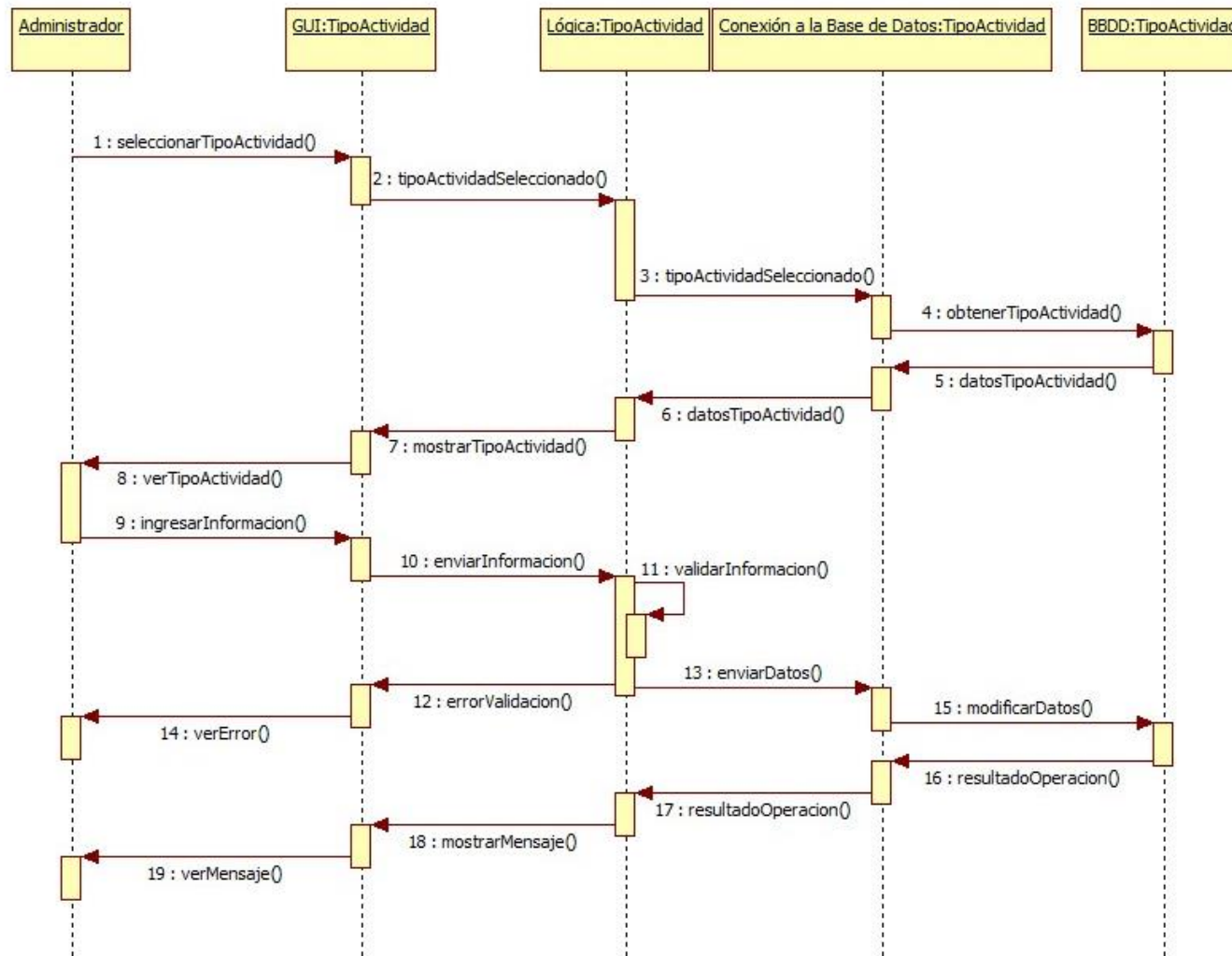
Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar una modificación sobre un tipo de actividades.	2. El sistema muestra el respectivo formulario de modificación de datos.
3. El usuario introduce la información en el formulario.	4. El sistema valida la información suministrada.
	6. El sistema informa el éxito de la operación.

Cursos alternos

6. La validación de la información suministrada en el formulario falla. El sistema informa al usuario y vuelve a la línea 2.

Figura 23. Diagrama de secuencia: modificar tipo de actividad



6.20 ELIMINAR TIPOS DE ACTIVIDAD

Actores: Experto temático (iniciador).

Objetivo: Eliminar algún tipo de actividad del sistema.

Descripción: el experto temático desea eliminar algún tipo de actividad, el sistema pide que seleccione el tipo de actividad, para posteriormente permitirle realizar dicha acción.

Tipo: esencial y primario.

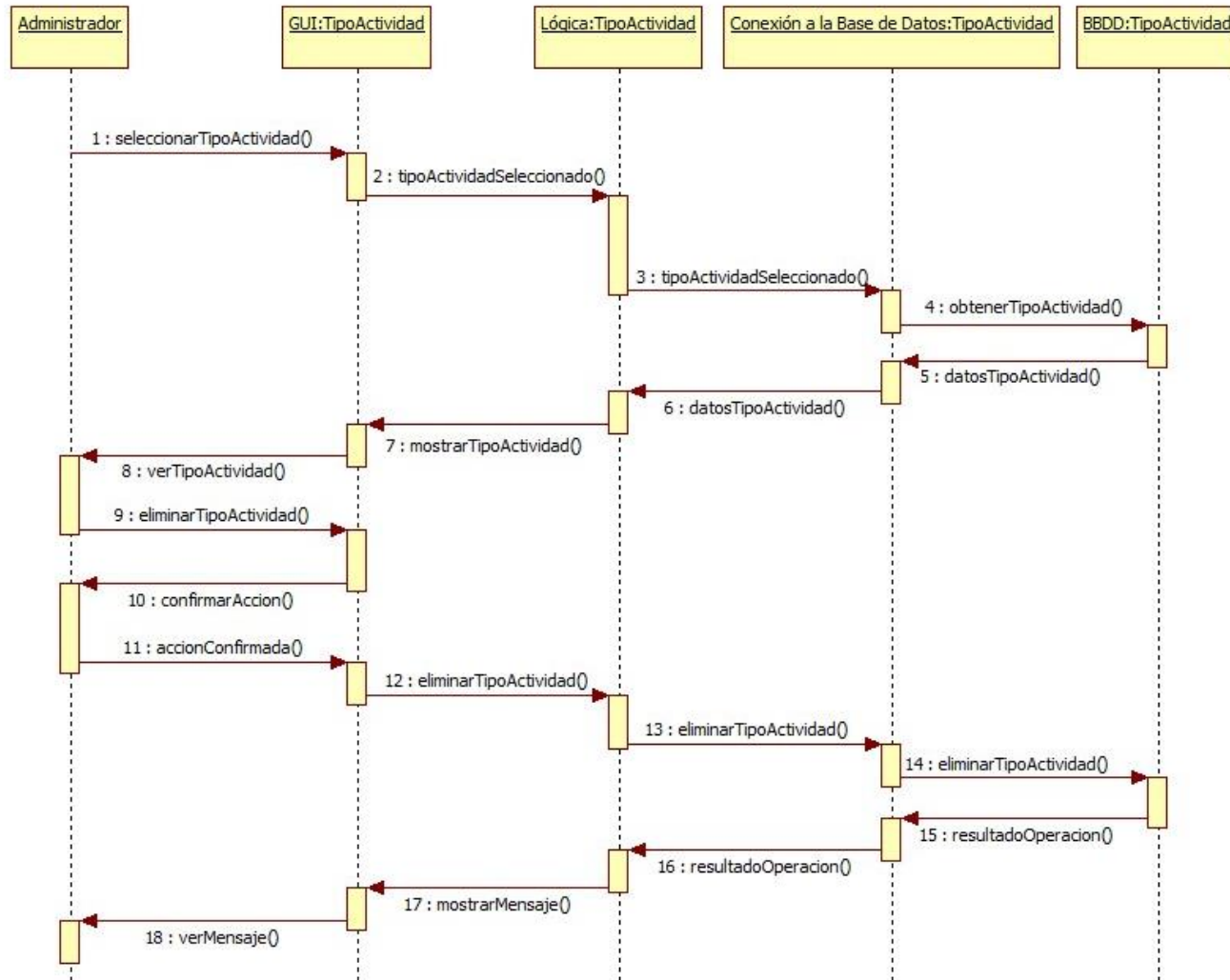
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea eliminar un tipo de actividad.	3. el sistema pide seleccionar el tipo de actividad.
4. El usuario elije.	5. El sistema pide confirmación de la acción a realizar.
6. El usuario confirma.	7. El sistema elimina el tipo de actividad del sistema.
	8. El sistema informa el éxito de la operación.

Cursos alternos:

6. El usuario no confirma la acción a realizar. Vuelve a la línea 3.

Figura 24. Diagrama de secuencia: eliminar tipo de actividad



6.21 CREACIÓN DE RECURSOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Permitir al experto temático y al docente crear un nuevo recurso perteneciente a un eje temático.

Descripción: El experto temático o el docente desean crear un nuevo recurso, el sistema deberá permitirselo e informar el estado final de esta operación.

Tipo: esencial y primario.

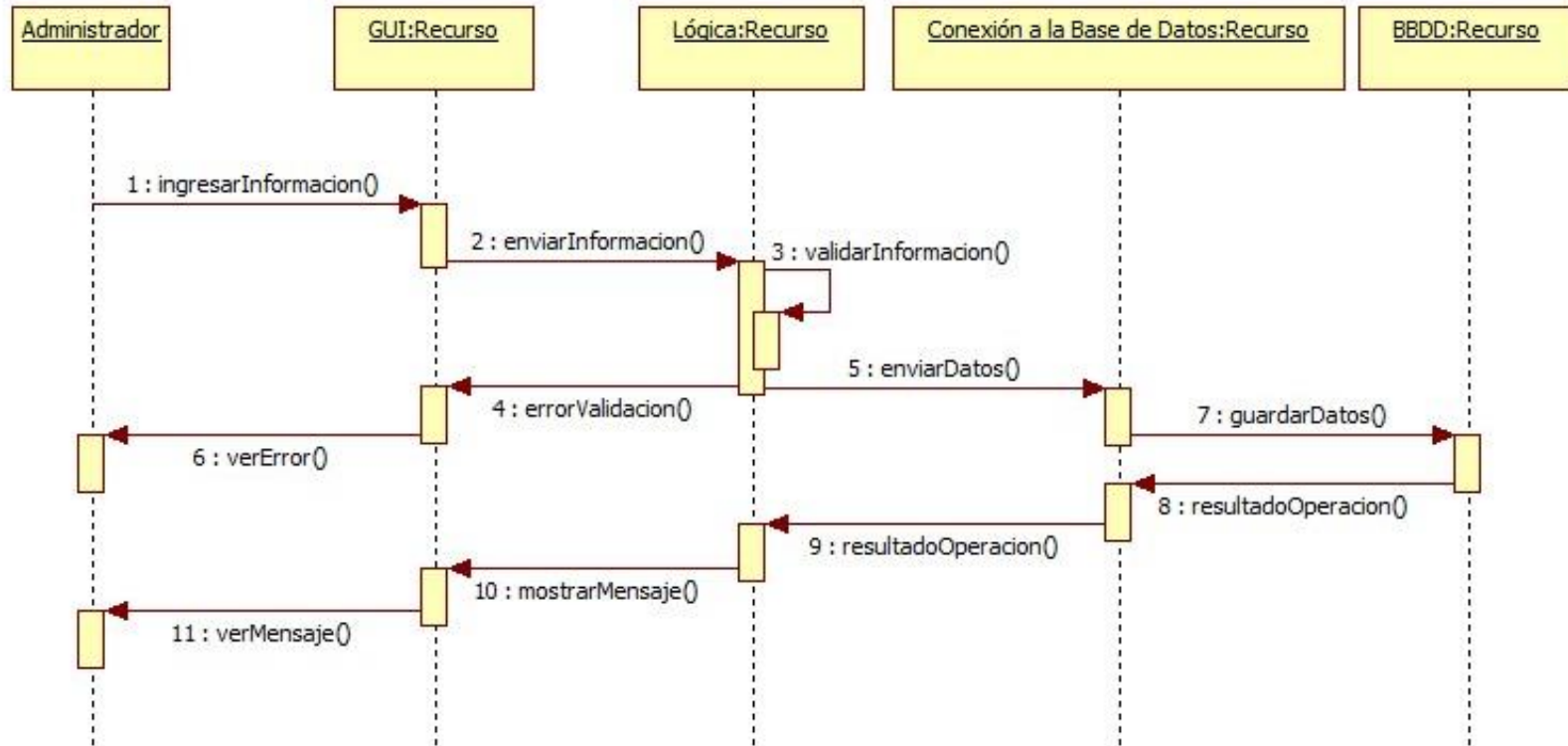
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea crear un nuevo recurso.	3. El sistema despliega el formulario con la información necesaria
4. El usuario llena el formulario.	5. El sistema valida la información suministrada en el formulario
	6. El sistema informa el éxito de la operación.

Cursos alternos:

5. La validación de los datos introducidos en el formulario falló. El sistema informa al usuario y vuelve a la línea 3.

Figura 25. Diagrama de secuencia: crear recurso



6.22 LISTAR RECURSOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Realizar un listado detallado de los recursos de un eje temático.

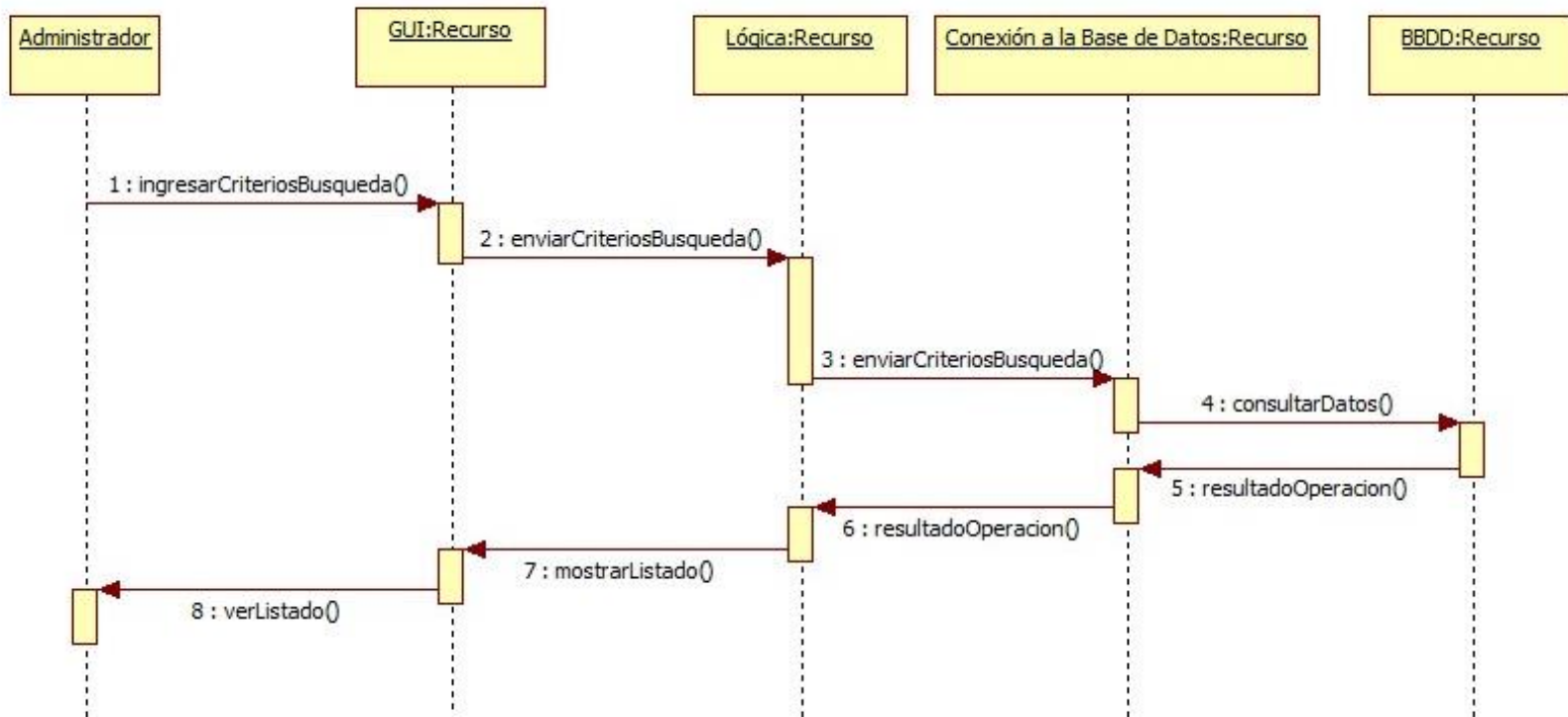
Descripción: El experto temático o docente desea listar los recursos de un eje temático, el sistema responde a esto con el respectivo listado detallado.

Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar un listado de recursos.	2. El sistema le muestra los criterios de búsqueda, entre ellos un eje temático en específico.
3. El usuario elije.	4. El sistema muestra el listado según la elección del usuario.

Figura 26. Diagrama de secuencia: listar recursos



6.23 MODIFICAR RECURSOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Realizar una modificación sobre algún recurso de un eje temático.

Descripción: el experto temático o docente desean modificar los recursos de un eje temático determinado, el sistema pide que seleccione el eje temático, para posteriormente permitirle realizar dicha modificación.

Tipo: esencial y primario.

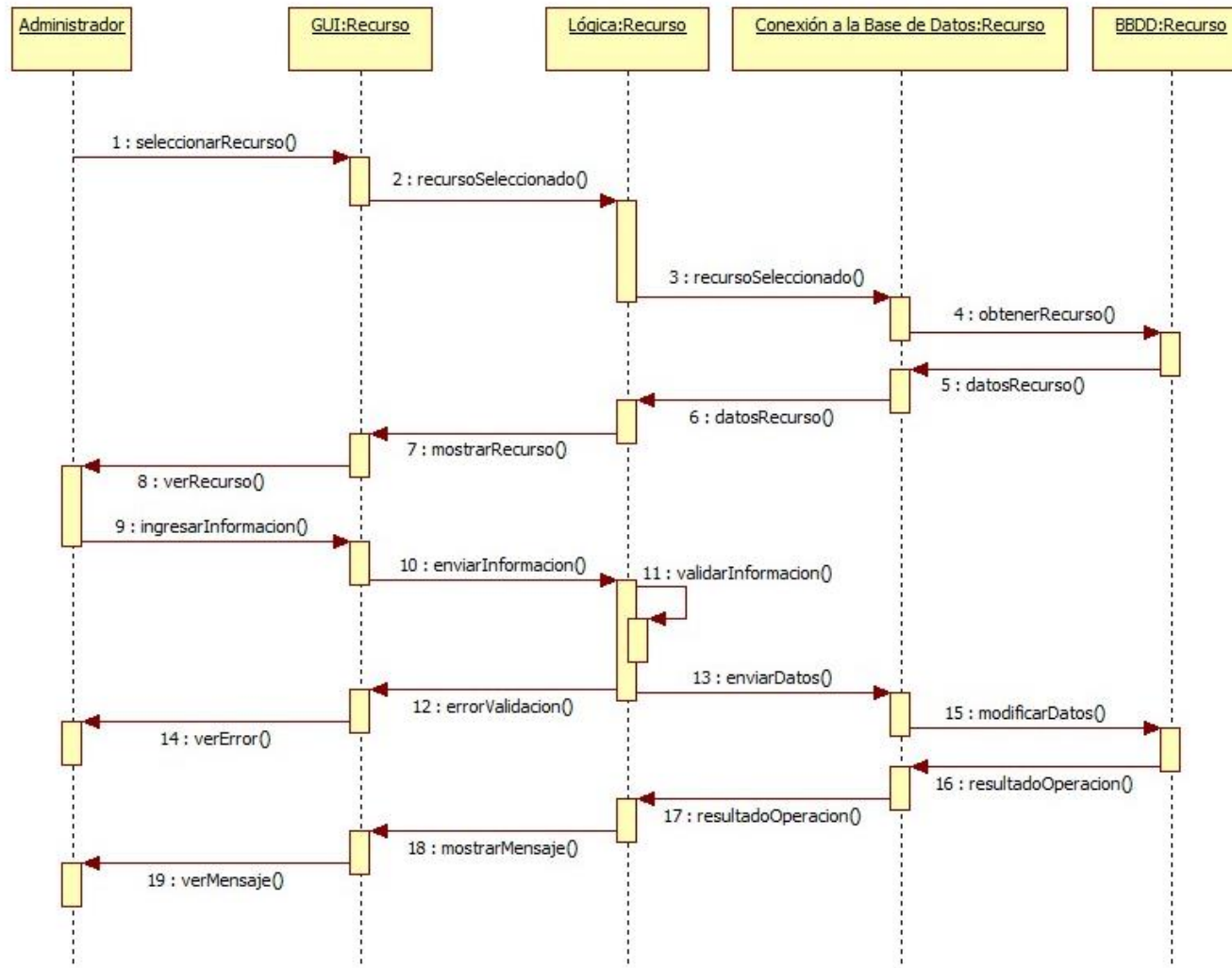
Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea realizar una modificación sobre un recurso.	2. el sistema pide seleccionar el eje temático.
3. El usuario elije.	4. El sistema muestra el respectivo formulario de modificación de datos.
5. El usuario introduce la información en el formulario.	6. El sistema valida la información suministrada.
	7. El sistema informa el éxito de la operación.

Cursos alternos

6. La validación de la información suministrada en el formulario falla. . El sistema informa al usuario y vuelve a la línea 4.

Figura 27. Diagrama de secuencia: modificar recurso



6.24 ELIMINAR RECURSOS

Actores: Experto temático (iniciador), Docente (iniciador).

Objetivo: Eliminar algún recurso del sistema.

Descripción: el experto temático o docente desean eliminar algún recurso de un eje temático determinado, el sistema pide que seleccione el eje temático, para posteriormente permitirle realizar dicha acción.

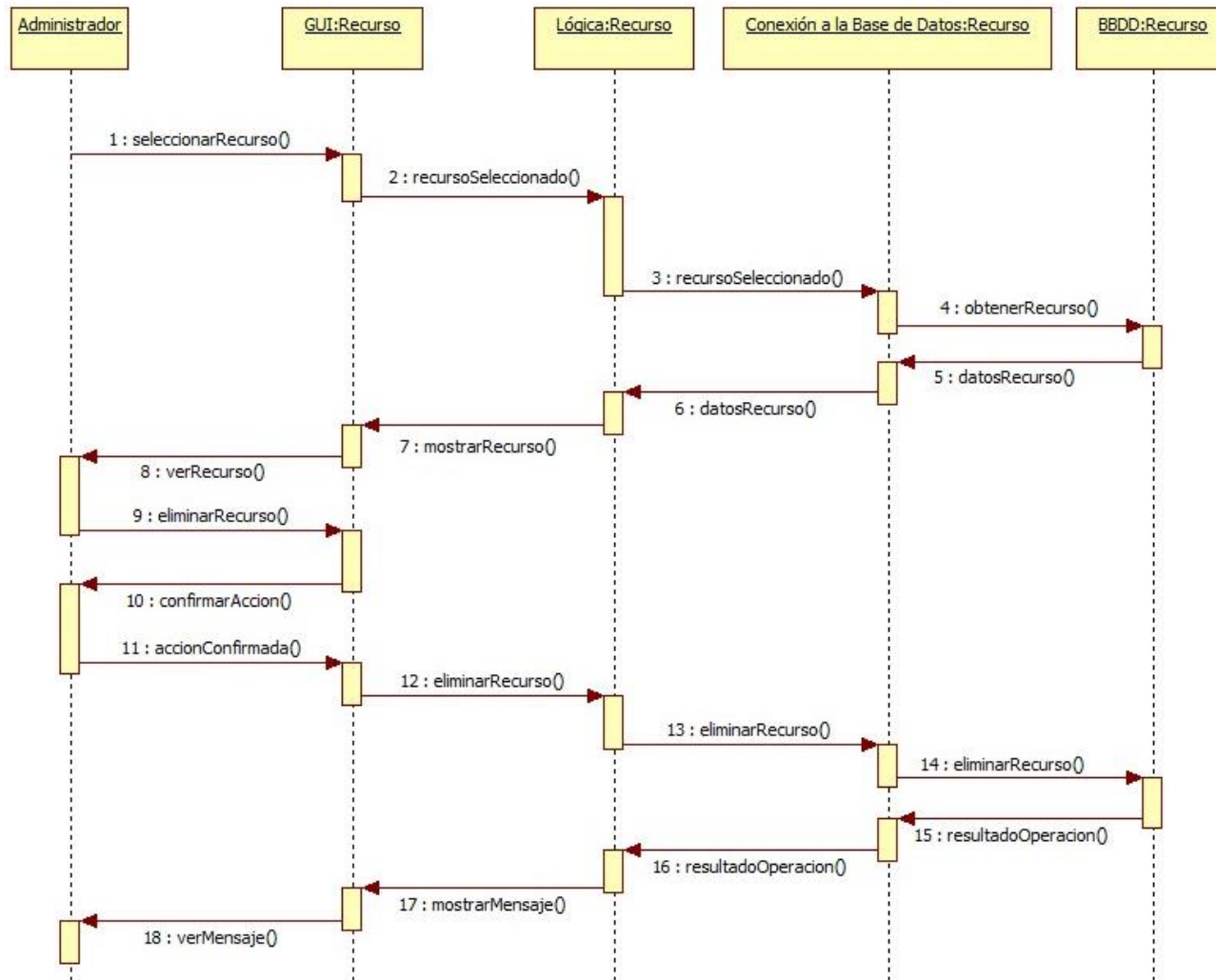
Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea eliminar un recurso.	3. el sistema pide seleccionar el eje tematico.
4. El usuario elije.	5. El sistema pide seleccionar un recurso
6. El usuario elije.	6. El sistema pide confirmación de la acción a realizar.
7. El usuario confirma.	7. El sistema elimina el recurso del sistema.
	9. El sistema informa el éxito de la operación.

Cursos alternos: 6. El usuario no confirma la acción a realizar. Vuelve a la línea 3.

Figura 28. Diagrama de secuencia: eliminar recurso



6.25 REVISAR ETAPA

Actores: Líder de Etapa (iniciador), Experto temático (iniciador).

Objetivo: Aceptar o rechazar una etapa de un módulo.

Descripción: El líder de etapa (Comunicador, Líder Pedagógico, Coordinador) o el Experto temático desea revisar la etapa de un módulo correspondiente a su área, una vez revisada debe decidir si la acepta o la rechaza.

Tipo: esencial y primario.

Curso normal de los eventos

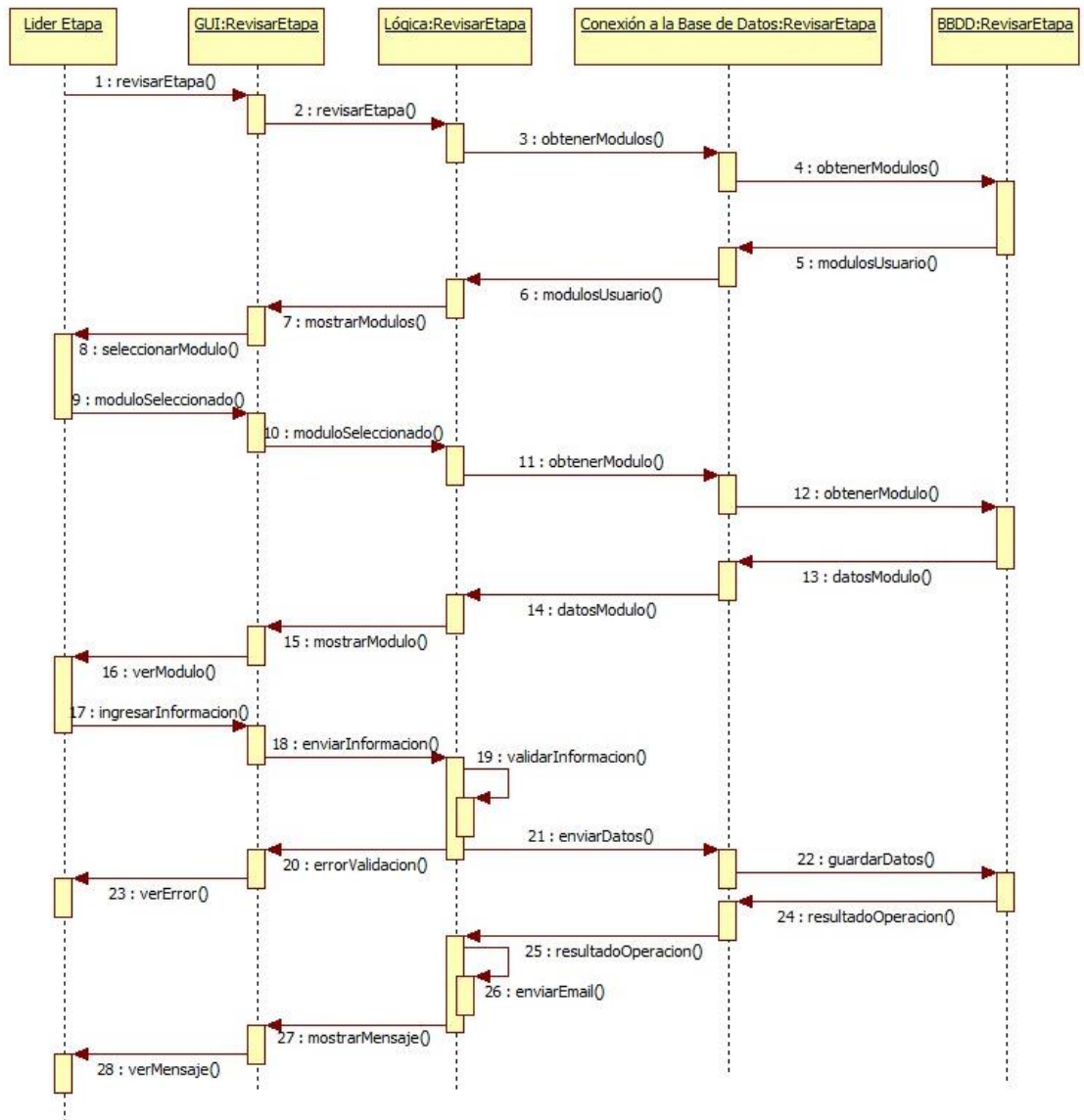
ACTOR	SISTEMA
1. Login (ver sección).	--
2. El usuario desea revisar su etapa.	3. El sistema muestra todos los módulos que tiene a cargo el usuario actual.
4. El usuario elige el módulo en el que desea trabajar.	-- 5. El sistema muestra la información del módulo seleccionado por el usuario, con el respectivo formulario de revisión.
6. El usuario revisa la etapa y decide aceptarla o rechazarla.	-- 6. El sistema valida la información suministrada.
	-- 7. El sistema notifica al Experto

	temático y al Docente encargado del módulo sobre la decisión del usuario (aceptada o rechazada) mediante correo electrónico.
	8. El sistema informa el éxito de la operación.

Cursos alternos:

6. La validación de la información suministrada en el formulario falla. El sistema informa al usuario y vuelve a la línea 5. 7. El sistema no puede notificar al Experto temático y al Docente. El sistema informa al usuario y continúa en la línea 8.

Figura 29. Diagrama de secuencia: revisar etapa



6.26 REVISAR MÓDULO

Actores: Experto temático (iniciador).

Objetivo: Aceptar o rechazar un módulo.

Descripción: El Experto temático desea revisar un módulo, una vez revisado debe decidir si la acepta o la rechaza.

Tipo: esencial y primario.

CURSO NORMAL DE LOS EVENTOS

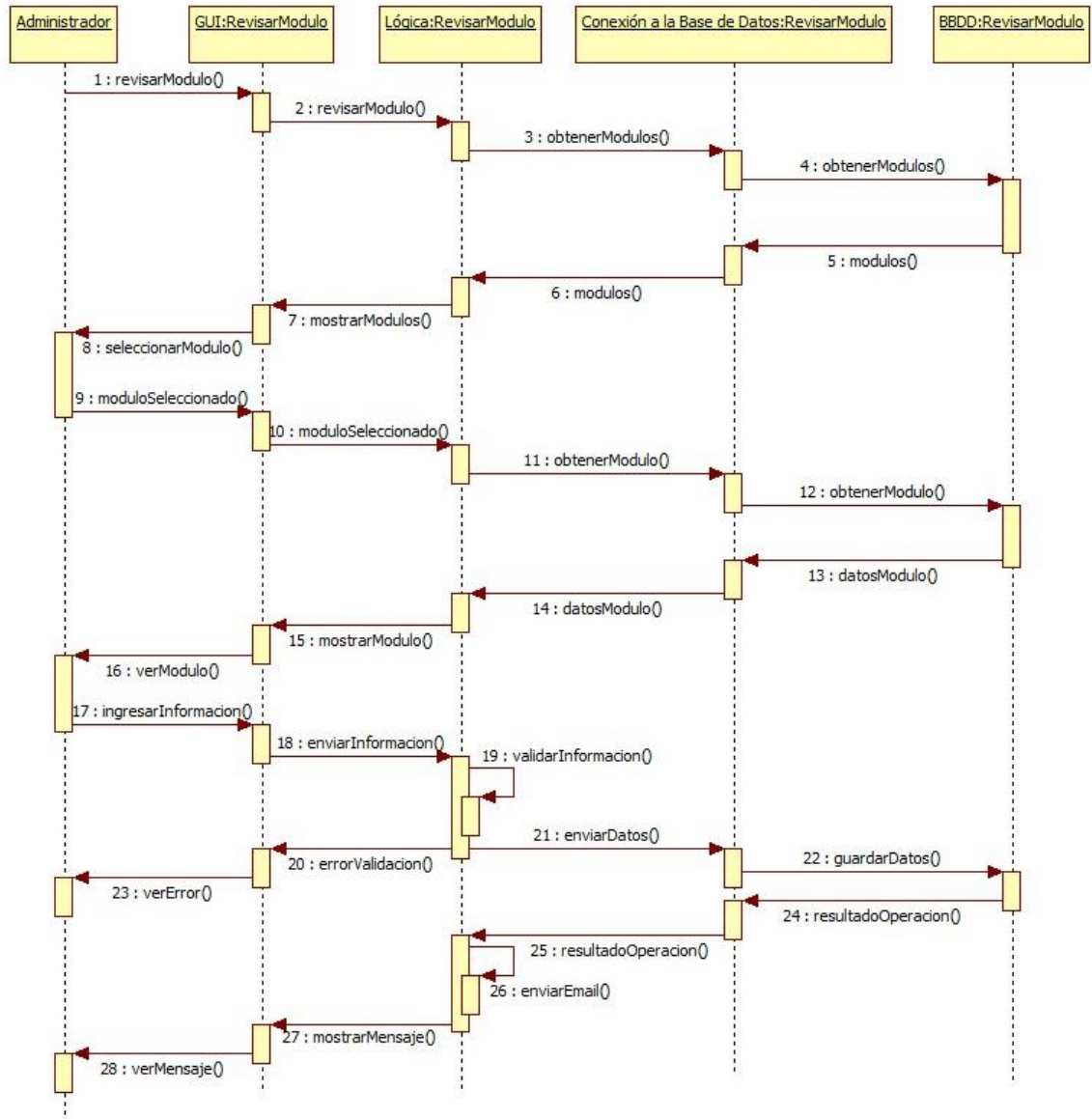
ACTOR	SISTEMA
1. Login (ver sección).
2. El usuario desea revisar un módulo.	3. El sistema muestra todos los módulos creados en el sistema.
4. El usuario elige el módulo que desea revisar.	5. El sistema muestra la información del módulo seleccionado por el usuario, con el respectivo formulario de revisión.
6. El usuario revisa el módulo y decide aceptarla o rechazarla.	6. El sistema valida la información suministrada.
	7. El sistema notifica al

	<p>Docente y al Líder de etapa pertenecientes al módulo sobre la decisión del usuario (aceptada o rechazada) mediante correo electrónico.</p>
	<p>8. El sistema informa el éxito de la operación.</p>

Cursos alternos:

6. La validación de la información suministrada en el formulario falla. El sistema informa al usuario y vuelve a la línea 5. 7. El sistema no puede notificar al Líder de Etapa y al Docente. El sistema informa al usuario y continúa en la línea 8.

Figura 30. Diagrama de secuencia: revisar módulo



6.27 VER TRAZABILIDAD

Actores: Todos los usuarios (iniciador).

Objetivo: Permitirle a los usuario ver todo el proceso de un módulo, desde su creación, hasta el estado actual en que se encuentra.

Descripción: El usuario desea ver la trazabilidad de un módulo, con posibilidad de exportarla.

Tipo: esencial y primario.

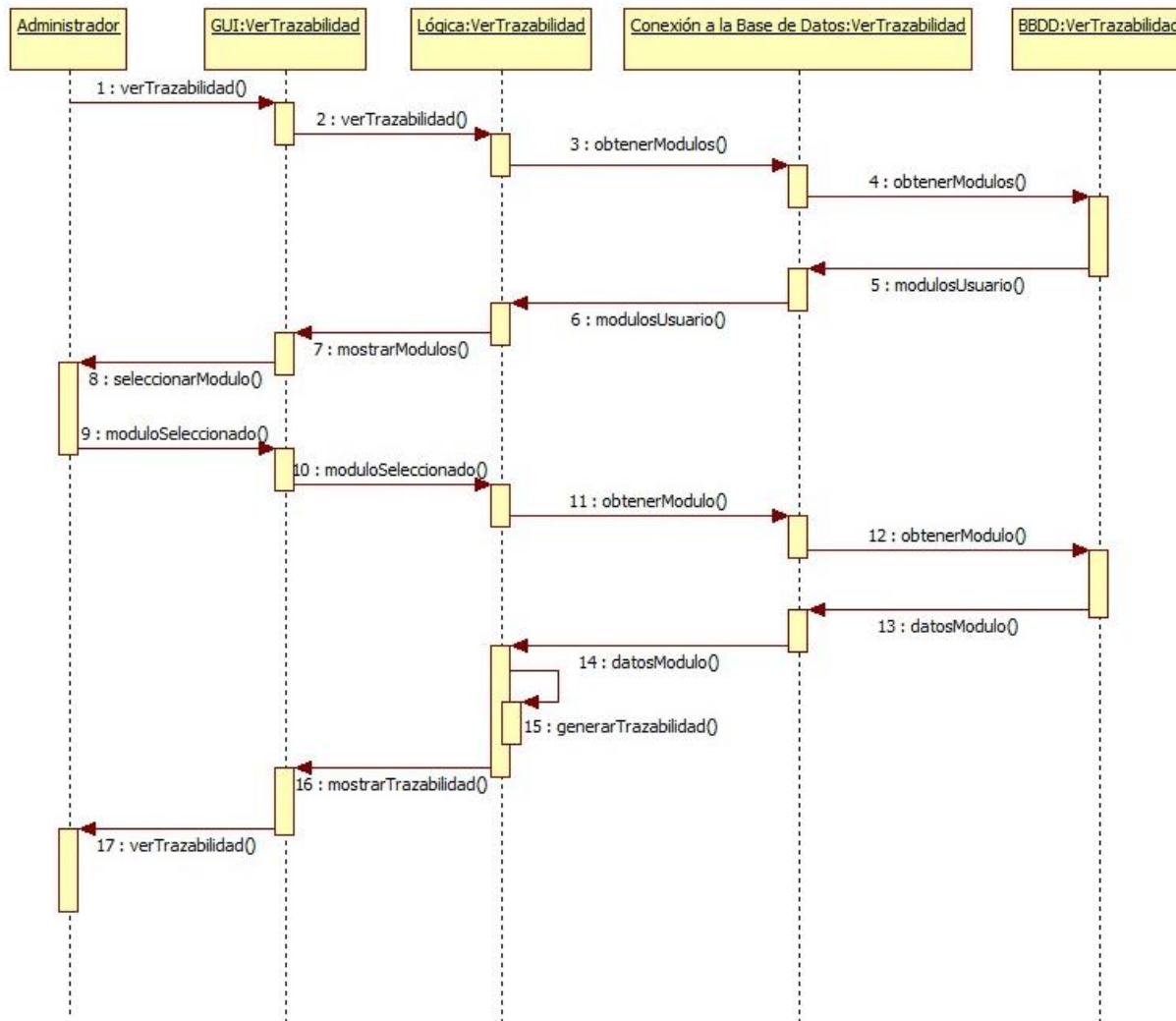
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea ver la trazabilidad de un módulo.	3. El sistema muestra todos los módulos creados y que tiene acceso el usuario en el sistema.
4. El usuario elige el módulo que desea ver.	5. El sistema genera toda la trazabilidad de éste módulo, con opción de exportar.
..	6. El sistema informa el éxito de la operación.

Cursos alternos

5. No fue posible generar la trazabilidad. El sistema informa al usuario y vuelve a la línea 3.

Figura 31. Diagrama de secuencia: ver trazabilidad



6.28 EXPORTAR TRAZABILIDAD

Actores: Todos los usuarios (iniciador).

Objetivo: Permitirle a los usuarios exportar la trazabilidad de un módulo.

Descripción: El usuario después de generar la trazabilidad de un módulo, desea exportarlo, el sistema se lo debe permitir, informando al final el éxito o no de la operación.

Tipo: esencial y primario.

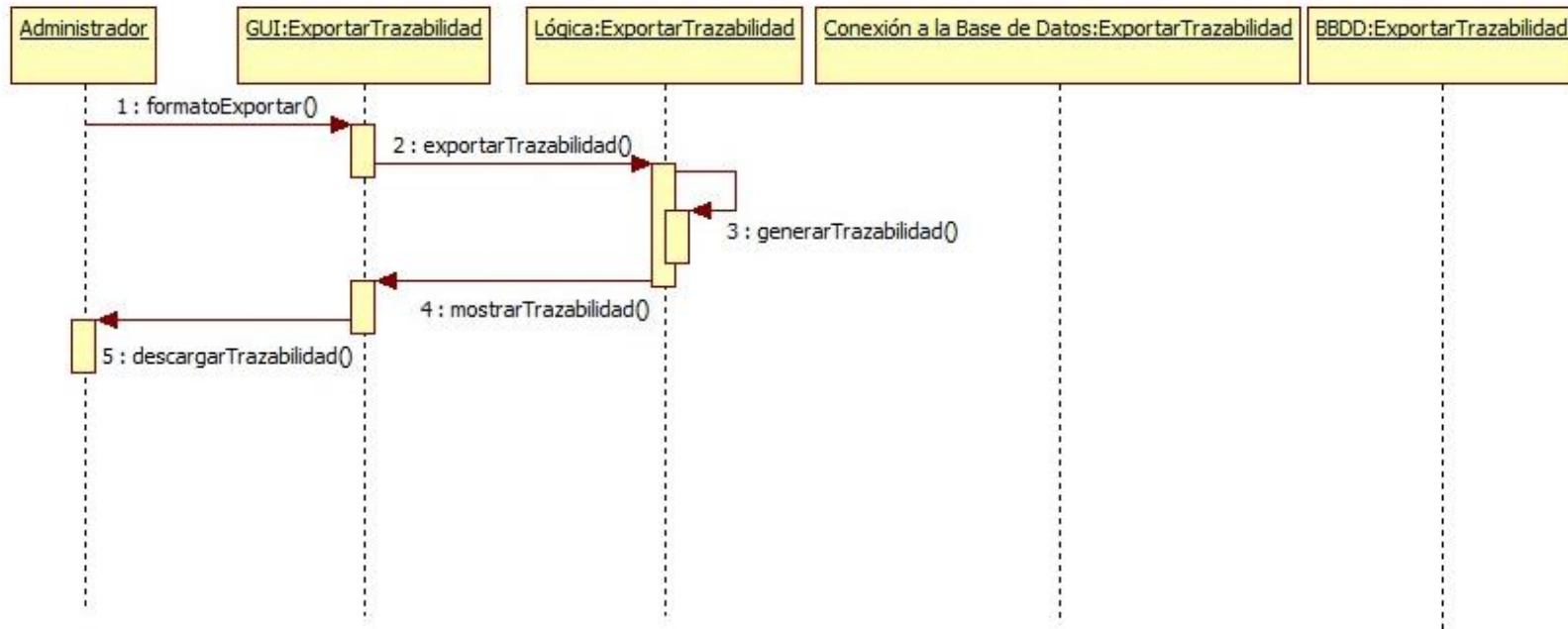
Curso normal de los eventos

ACTOR	SISTEMA
1. Login (ver sección).	
2. El usuario desea exportar la trazabilidad de un módulo.	3 El sistema despliega los formatos con posibilidad de exportación.
4. El usuario elige el formato a generar.	5. El sistema genera el reporte de trazabilidad del módulo en el formato elegido por el usuario.
..	6. El sistema informa el éxito de la operación.

Cursos alternos

5. No fue posible generar le reporte de trazabilidad a exportar. El sistema informa al usuario y vuelve a la línea 3.

Figura 32. Diagrama de secuencia: exportar trazabilidad



6.29 SECCION – LOGIN

Actores: Todos los usuarios (iniciador).

Objetivo: Permitir el acceso al sistema.

Descripción: Un usuario desea acceder al sistema, para esto ingresara sus datos de cuenta y la aplicación los validara.

Tipo: esencial y primario.

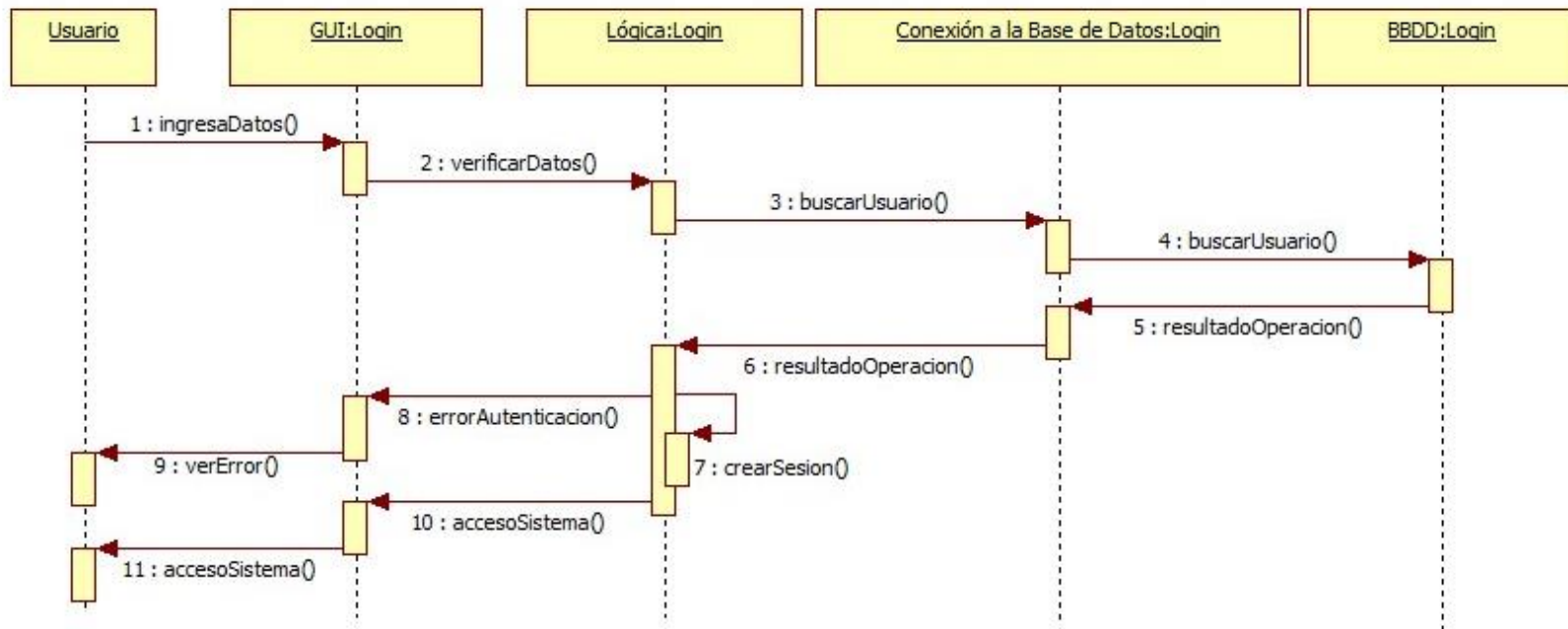
Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea ingresar al sistema	2. El sistema le muestra un formulario de ingreso de datos.
3. El usuario ingresa sus datos de cuenta.	4. El sistema valida los datos.
	5. El sistema permite el acceso al usuario.

Cursos alternos

4. la validación no es correcta. El sistema informa al usuario y vuelve a la línea 2.

Figura 33. Diagrama de secuencia: sección - login



6.30 SECCION – LOGOUT

Actores: Todos los usuarios (iniciador).

Objetivo: Permitir salir de manera segura del sistema.

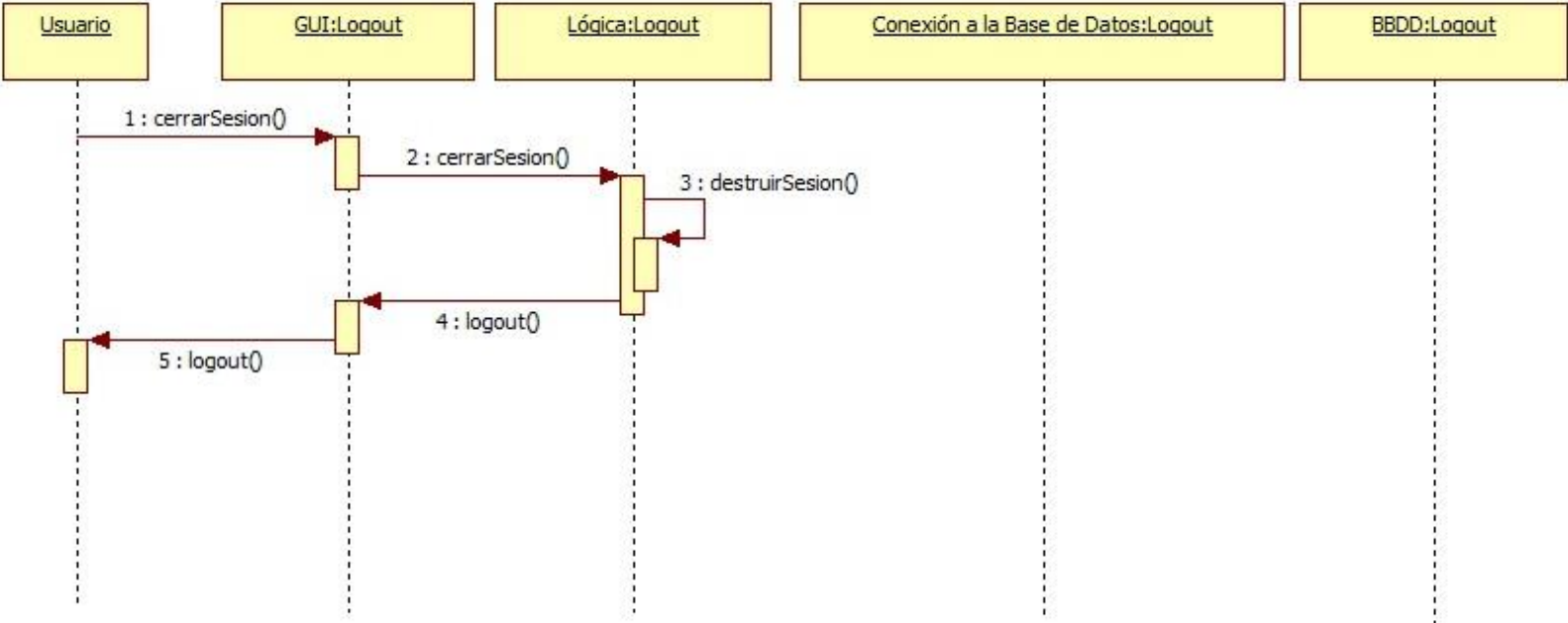
Descripción: Un usuario desea salir del sistema, para esto cerrara sesión del sistema.

Tipo: esencial y primario.

Curso normal de los eventos

ACTOR	SISTEMA
1. El usuario desea salir del sistema	
3. El usuario cierra sesión desde el menú del sistema.	4. El sistema cierra la sesión del usuario actual.
	5. El sistema re direcciona al usuario a la página de ingreso.

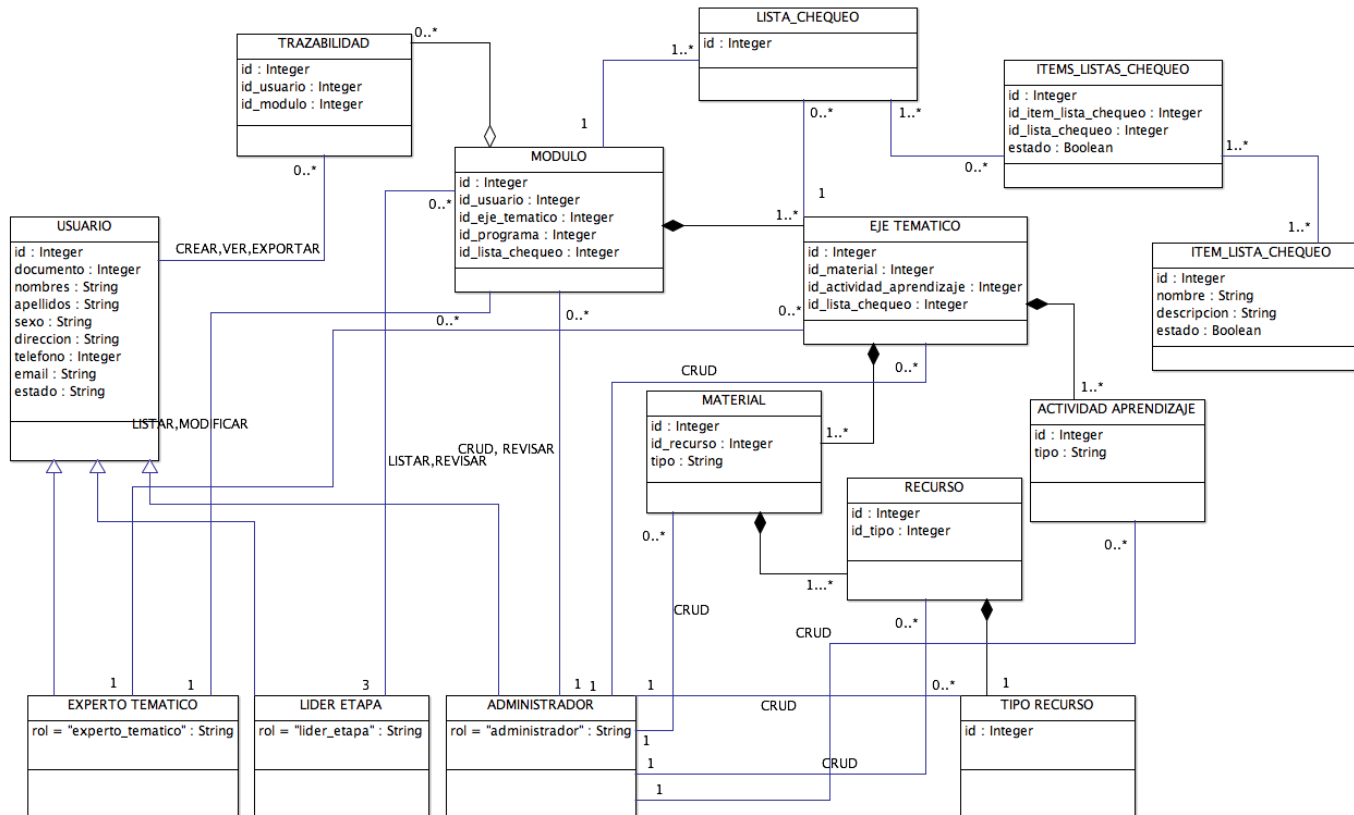
Figura 34. Diagrama de secuencia: sección - logout



7. DIAGRAMAS UML

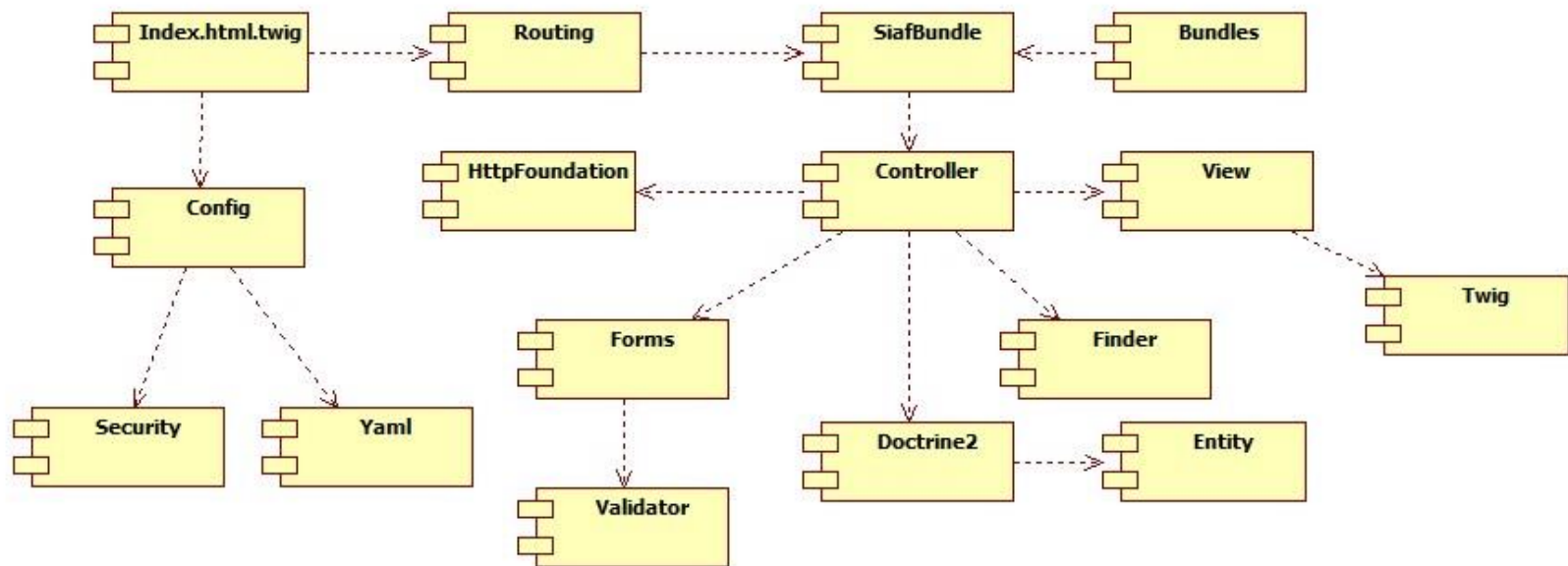
7.1 DIAGRAMA DE CLASES

Figura 35. Diagrama de clases



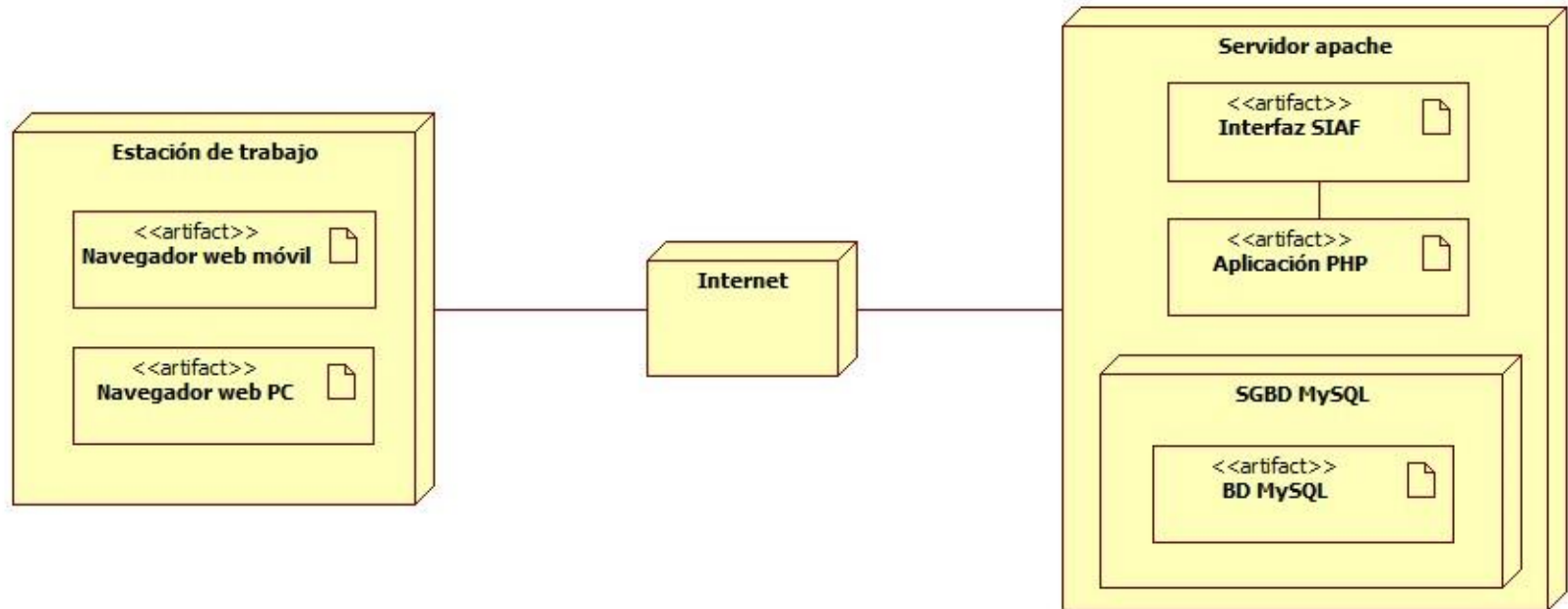
7.2 DIAGRAMA DE COMPONENTES

Figura 36. Diagrama de componentes



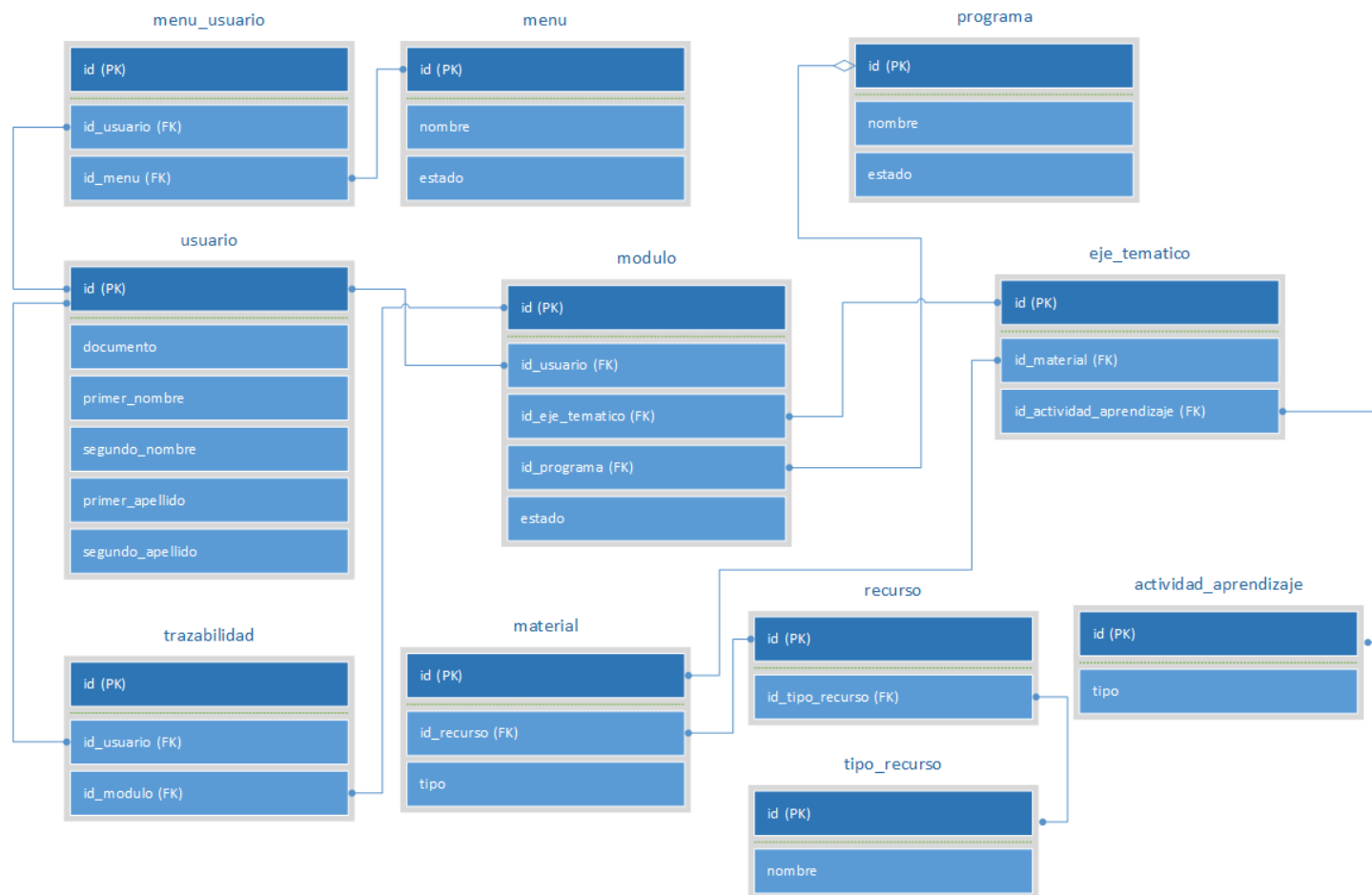
7.3 DIAGRAMA DE DESPLIEGUE

Figura 37. Diagrama de despliegue



7.4 DIAGRAMA ENTIDAD - RELACIÓN

Figura 38. Diagrama entidad - relación



8. CASOS DE PRUEBA

8.1 ALCANCE

Este documento cubre el conjunto de pruebas realizadas sobre los casos de uso “Listar módulos”, “Creación de módulo”, “Modificación de módulo”, “Eliminar módulo”, “Revisar módulo”, “Ver trazabilidad” y “Log in” con el fin de garantizar que en la ejecución de cada caso de uso que se evalúe, se obtenga el resultado esperado cuando las variables ingresadas son correctas y el mensaje de error adecuado en caso contrario.

8.2 ESTRATEGIA

Se utilizará la técnica de diseño de pruebas de caja negra, ya que esta no considera el detalle procedimental plasmado en el código fuente, si no su compatibilidad con la especificación detallada en los casos de uso.

8.3 CASOS DE PRUEBA

8.3.1 Listar módulos.

Basado en el flujo de eventos descritos en la documentación de este caso de uso se deduce la siguiente matriz de decisión:

Tabla 1. Caso de prueba: listar módulos

ID	Escenario / Condición	Sesión de usuario	Resultado esperado
CP1	Escenario 1/Listar módulos	(C)	Mostrar los módulos que tiene el usuario a cargo.
CP2	Escenario 2/Listar módulos	(I)	Redireccionar al usuario a la sección de log in.

Donde:

C = Valor Correcto, **I** = Valor Incorrecto

8.3.2 Crear módulos.

Basado en el flujo de eventos descritos en la documentación de este caso de uso se deduce la siguiente matriz de decisión:

Tabla 2. Caso de prueba: crear módulos

ID	Escenario/Condición	Sesión de usuario	Campos obligatorios	Resultado esperado
CP1	Escenario 1/Crear módulo	(I)	(I)	Redireccionar al usuario a la sección de autenticación del sistema.
CP2	Escenario 2/Crear módulo	(I)	(C)	Redireccionar al usuario a la sección de autenticación del sistema.
CP3	Escenario 3/Crear módulo	(C)	(I)	No crear módulo e informar al usuario cuales son los campos obligatorios.
CP4	Escenario 4/Crear módulo	(C)	(N)	No crear módulo e informar al usuario cuales son los campos obligatorios.
CP5	Escenario 5/Crear módulo	(C)	(C)	Crear el módulo e informar al usuario.

Donde:

C = Valor Correcto, **I** = Valor Incorrecto

8.3.3 Modificar módulos.

Basado en el flujo de eventos descritos en la documentación de este caso de uso se deduce la siguiente matriz de decisión:

Tabla 3. Caso de prueba: modificar módulos

ID	Escenario/Condición	Sesión de usuario	Módulo	Campos obligatorios	Resultado esperado
CP1	Escenario 1/ Listar módulos	(I)	(C)	N/A	Redireccionar al usuario a la sección de autenticación del sistema.
CP2	Escenario 2/ Listar módulos	(C)	(N)	N/A	Mostrar los módulos que tiene el usuario a cargo.
CP3	Escenario 3/ Modificar módulo	(C)	(I)	N/A	Muestra al usuario mensaje que el módulo seleccionado es inválido.
CP4	Escenario 4/ Modificar módulo	(I)	(I)	N/A	Redireccionar al usuario a la sección de autenticación del sistema.
CP5	Escenario 5/Modificar módulo	(C)	(C)	(I)	No guardar módulo e informar al usuario cuales son los campos obligatorios.
CP6	Escenario 6/Modificar módulo	(C)	(C)	(N)	No guardar módulo e informar al usuario los

ID	Escenario/Condición	Sesión de usuario	Módulo	Campos obligatorios	Resultado esperado
					campos obligatorios.
CP7	Escenario 7/Modificar módulo	(C)	(C)	(C)	Guardar módulo e informar al usuario.

Donde:

C = Valor Correcto, **I** = Valor Incorrecto, **N** = Ningún valor

8.3.4 Eliminar módulos.

Basado en el flujo de eventos descritos en la documentación de este caso de uso se deduce la siguiente matriz de decisión:

Tabla 4. Caso de prueba: eliminar módulos

ID	Escenario/Condición	Sesión de usuario	Módulo	Resultado esperado
CP1	Escenario 1/Listar módulos	(I)	N/A	Redireccionar al usuario a la sección de autenticación del sistema.
CP2	Escenario 2/Listar módulos	(C)	N/A	Mostrar los módulos que tiene el usuario a cargo.
CP3	Escenario 3/Eliminar módulo	(C)	(I)	Muestra al usuario mensaje que el módulo seleccionado es inválido.
CP4	Escenario 4/Eliminar módulo	(I)	(I)	Redireccionar al usuario a la sección de autenticación del sistema.
CP5	Escenario 5/Eliminar módulo	(C)	(C)	Eliminar el módulo del sistema e informar al usuario.

Donde:

C = Valor Correcto, **I** = Valor Incorrecto

8.3.5 Revisar módulo.

Basado en el flujo de eventos descritos en la documentación de este caso de uso se deduce la siguiente matriz de decisión:

Tabla 5. Caso de prueba: revisar módulo

ID	Escenario/Condición	Sesión de usuario	Módulo	Campos obligatorios	Resultado esperado
CP1	Escenario 1/Listar módulo	(C)	N/A	N/A	Mostrar los módulos que tiene el usuario a cargo.
CP2	Escenario 2/Listar módulo	(I)	N/A	N/A	Redireccionar a la sección de autenticación del sistema.
CP3	Escenario 3/Seleccionar módulo	(C)	(C)	N/A	Mostrar el formulario de revisión del módulo seleccionado.
CP4	Escenario 4/Seleccionar módulo	(I)	(I)	N/A	Redireccionar a la sección de autenticación del sistema.
CP5	Escenario 5/Seleccionar módulo	(I)	(C)	N/A	Redireccionar a la sección de autenticación del sistema.
CP6	Escenario 6/Seleccionar módulo	(C)	(I)	N/A	Muestra al usuario mensaje que el módulo seleccionado es inválido.
CP7	Escenario 8/Aceptar	(C)	(C)	(C)	Aceptar el

ID	Escenario/Condición	Sesión de usuario	Módulo	Campos obligatorios	Resultado esperado
	módulo				módulo para el usuario actual y enviar correo de notificación.
CP8	Escenario 9/Aceptar módulo	(I)	(I)	(I)	Redireccionar a la sección de autenticación del sistema.
CP9	Escenario 10/Aceptar módulo	(I)	(I)	(C)	Redireccionar a la sección de autenticación del sistema.
CP10	Escenario 11/Aceptar módulo	(I)	(C)	(C)	Redireccionar a la sección de autenticación del sistema.
CP11	Escenario 11/Aceptar módulo	(C)	(I)	(I)	Muestra al usuario mensaje que el módulo seleccionado es inválido.
CP12	Escenario 12/Aceptar módulo	(C)	(C)	(I)	No aceptar módulo e informar al usuario cuales son los campos obligatorios.
CP13	Escenario 13/Rechazar módulo	(C)	(C)	(C)	Rechazar el módulo para el usuario actual.
CP14	Escenario 14/Rechazar módulo	(I)	(I)	(I)	Redireccionar a la sección de

ID	Escenario/Condición	Sesión de usuario	Módulo	Campos obligatorios	Resultado esperado
					autenticación del sistema.
CP15	Escenario 15/Rechazar módulo	(I)	(I)	(C)	Redireccionar a la sección de autenticación del sistema.
CP16	Escenario 16/Rechazar módulo	(I)	(C)	(C)	Redireccionar a la sección de autenticación del sistema.
CP17	Escenario 17/Rechazar módulo	(C)	(I)	(I)	Muestra al usuario mensaje que el módulo seleccionado es inválido.
CP18	Escenario 18/Rechazar módulo	(C)	(C)	(I)	No rechazar módulo e informar al usuario cuales son los campos obligatorios.

Donde:

C = Valor Correcto, **I** = Valor Incorrecto

8.3.6 Ver trazabilidad.

Basado en el flujo de eventos descritos en la documentación de este caso de uso se deduce la siguiente matriz de decisión:

Tabla 6. Caso de prueba: ver trazabilidad

ID	Escenario/Condición	Sesión de usuario	Módulo	Resultado esperado
CP1	Escenario 1/Seleccionar módulo	(C)	N/A	Mostrar los módulos que tiene el usuario a cargo.
CP2	Escenario 2/Seleccionar módulo	(I)	N/A	Redireccionar a la sección de autenticación del sistema.
CP3	Escenario 3/Seleccionar módulo	(C)	(C)	Mostrar el formulario de revisión del módulo seleccionado.
CP4	Escenario 4/Seleccionar módulo	(I)	(I)	Redireccionar a la sección de autenticación del sistema.
CP5	Escenario 5/Seleccionar módulo	(I)	(C)	Redireccionar a la sección de autenticación

ID	Escenario/Condición	Sesión de usuario	Módulo	Resultado esperado
				del sistema.
CP6	Escenario 6/Seleccionar módulo	(C)	(I)	Muestra al usuario mensaje que el módulo seleccionado es inválido.
CP7	Escenario 8/Generar Trazabilidad	(C)	(C)	Generar toda la trazabilidad del módulo. desde su creación, hasta el estado actual.
CP8	Escenario 9/Generar Trazabilidad	(I)	(I)	Redireccionar a la sección de autenticación del sistema.
CP9	Escenario 10/Generar Trazabilidad	(I)	(C)	Redireccionar a la sección de autenticación del sistema.
CP10	Escenario 11/Generar Trazabilidad	(C)	(I)	Muestra al usuario mensaje que el módulo seleccionado es inválido.

Donde: **C** = Valor Correcto, **I** = Valor Incorrecto

8.3.7 Log in.

Basado en el flujo de eventos descritos en la documentación de este caso de uso se deduce la siguiente matriz de decisión:

Tabla 7. Caso de prueba: login

ID	Escenario/Condición	usuario	contraseña	Resultado esperado
CP1	Escenario 1/Log in	(C)	(N)	Mostrar al usuario mensaje de credenciales inválidas.
CP2	Escenario 2/Log in	(N)	(C)	Mostrar al usuario mensaje de credenciales inválidas.
CP3	Escenario 3/Log in	(N)	(N)	Mostrar al usuario mensaje de credenciales inválidas.
CP4	Escenario 4/Log in	(C)	(I)	Mostrar al usuario mensaje de credenciales inválidas.

ID	Escenario/Condición	usuario	contraseña	Resultado esperado
CP5	Escenario 5/Log in	(I)	(C)	Mostrar al usuario mensaje de credenciales inválidas.
CP6	Escenario 6/Log in	(I)	(I)	Mostrar al usuario mensaje de credenciales inválidas.
CP7	Escenario 7/Log in	(C)	(C)	Permitir acceso a la plataforma.

Donde:

C = Valor Correcto, **I** = Valor Incorrecto, **N** = Ningún valor

8.3.8 Log out.

Basado en el flujo de eventos descritos en la documentación de este caso de uso se deduce la siguiente matriz de decisión:

Tabla 8. Caso de prueba: logout

ID	Escenario/Condición	Sesión de usuario	Resultado esperado
CP1	Escenario 1/Log out	(I)	Redireccionar a la sección de autenticación del sistema.
CP2	Escenario 2/Log out	(C)	Eliminar la sesión del usuario y redireccionarlo a la sección de autenticación del sistema.

Donde:

C = Valor Correcto, **I** = Valor Incorrecto, **N** = Ningún valor

CONCLUSIONES

- Se concluye que implementar el modelo en cascada como metodología para el diseño del sistema de información, hace que la planificación del proyecto sea más sencilla y de fácil comprensión, debido al claro orden en que se encuentran cada una de sus etapas.
- Hacer uso del modelo iterativo como complemento en la fase de recolección de requerimientos, permite realizar una mejor proyección y estructuración del trabajo; así mismo como obtener resultados más precisos.
- Realizar el diseño de una aplicación implementando el patrón de diseño Modelo Vista Controlador (MVC), proporciona un buen manejo de estandarización, fácil acoplamiento al proyecto y mantenibilidad del código.
- Implementar un marco de trabajo como Symfony PHP Framework y usar adecuadamente las herramientas que este proporciona, permite disminuir el tiempo y optimizar el desarrollo de los proyectos de software.
- Establecer y realizar casos de prueba proporciona una alta probabilidad de obtener un desarrollo de buena calidad y de esta manera disminuir en gran medida todo tipo de errores y fallos que el cliente final pudiera llegar a experimentar.

RECOMENDACIONES

- Se considera importante implementar (además de los casos de prueba) pruebas unitarias y funcionales si se quiere aumentar aún más la calidad del software. Además, es relevante mencionar que si se combinan las pruebas unitarias y funcionales con otra buena práctica de ingeniería de software como TDD⁵ (desarrollo guiado por pruebas) y BDD⁶ (desarrollo guiado por comportamiento), no sólo se aumentará enormemente la calidad, si no que también se reducirán los costos en gran medida.
- El desarrollo de software basado en marcos de trabajo que automatizan tareas comunes, permiten que el equipo desarrollador se enfoque por completo en las características específicas del proyecto y como consecuencia reducir tiempo y costos.
- La documentación en la etapa de codificación de un desarrollo de software es indispensable a la hora de mantener y escalar la aplicación. Un desarrollo de software de buena calidad también va de la mano con una buena documentación. De esta manera se evitará invertir mucho más tiempo en investigar, analizar y hacer ingeniería inversa.

⁵ *Test-driven development. Más información:* <<http://agiledata.org/essays/tdd.html>>

⁶ *Behavior-driven development. Mas información:* <<http://guide.agilealliance.org/guide/bdd.html>>

BIBLIOGRAFIA

- [citado en 22 septiembre de 2015] Disponible en Internet: <http://astreo.ii.uam.es/~jlara/TACCII/8_Patrones.pdf>
- [citado en 22 septiembre de 2015] Disponible en Internet: <http://es.wikipedia.org/wiki/Desarrollo_en_cascada>
- [citado en 22 septiembre de 2015] Disponible en Internet: <[http://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado#Tipos de Diagramas de UML](http://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado#Tipos_de_Diagramas_de_UML)>
- [citado en 22 septiembre de 2015] Disponible en Internet: <<http://es.wikipedia.org/wiki/Modelo%20%93vista%20%93controlador>>
- [citado en 22 septiembre de 2015] Disponible en Internet: <<http://guide.agilealliance.org/guide/bdd.html>>
- [citado en 22 septiembre de 2015] Disponible en Internet: <<http://lscauml.wordpress.com/2012/01/05/disciplinas/>>
- [citado en 22 septiembre de 2015] Disponible en Internet: <<http://www.agiledata.org/essays/tdd.html>>
- [citado en 22 septiembre de 2015] Disponible en Internet: <<http://www.eli-mariamoliner.dyndns.org/~fperal/proy/ingenieriaSW.pdf>>
- [citado en 22 septiembre de 2015] Disponible en Internet: <<http://www.desarrolloweb.com/articulos/que-es-mvc.html>>
- [citado en 22 septiembre de 2015] Disponible en Internet: <<http://www.fdi.ucm.es/profesor/jpavon/poo/2.14pdoo.pdf>>
- [citado en 22 septiembre de 2015] Disponible en Internet: <<http://www.jordisan.net/blog/2006/que-es-un-framework/>>
- [citado en 22 septiembre de 2015] Disponible en Internet: <http://www.symfony.com/legacy/doc/jobee/1_4/es/04?orm=Propel>