

Universidad de las Ciencias Informáticas

Facultad 10



**Título: Interfaz de Administración Web para el Sistema de Filtrado Filpacon**

Trabajo de Diploma para optar por el Título de Ingeniería en Ciencias Informáticas

**Autores:** José Ramón Hermosilla Moreno  
Luis Enrique Sánchez Arce

**Tutor:** Ing. Alain Guerrero Enamorado  
**Co-Tutor:** Ing. Siovel Rodríguez Morales

**Ciudad de la Habana, 2 de Julio de 2008**

## Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
José Ramón Hermosilla Moreno  
Autor

\_\_\_\_\_  
Luis Enrique Sánchez Arce  
Autor

\_\_\_\_\_  
Ing. Alain Guerrero Enamorado  
Tutor

\_\_\_\_\_  
Ing. Siovel Rodríguez Morales  
Co-Tutor

## **Agradecimientos**

A mi familia por su apoyo y ejemplo durante la vida.

A mis amigos por estar siempre ahí desinteresadamente.

A los demás integrantes del proyecto Filtrado de Paquetes por Contenidos (Filpacon) por brindarme su ayuda en todo momento.

***Luis Enrique Sánchez Arce***

A Dios, por sus misericordias y por haberme permitido llegar hasta aquí.

A mis padres y hermanos, por su amor y apoyo incondicional.

Al resto de mi familia por su preocupación y cariño.

A mi tía Lidia y su familia, por estar siempre dispuestos a ayudarme en todo.

A todos los que de una forma u otra me han brindado su ayuda.

***José Ramón Hermosilla Moreno***

Agradecemos además a:

La Revolución Cubana, la Universidad de las Ciencias Informáticas (UCI) y en especial la Facultad 10 por darnos la oportunidad de insertarnos en el maravilloso mundo del Software Libre.

Nuestros profesores por su ayuda en nuestra preparación profesional.

Nuestros tutores Alain y Siovel por toda la ayuda que nos han brindado y por sus orientaciones y la confianza que depositaron en nosotros.

## **Dedicatoria**

Dedico este trabajo a mis padres (Rolando y Maritza), por darme la vida y siempre estar ahí cuando los necesito. A mi hermano, mis abuelos, mi cuñada y sobrinita por compartir tantos momentos buenos. A mis amigos y amigas de la Universidad; Dayron, Miguel, Yanni, Kiuver, Pedro, Dovie, Yandry (Ruso) y a todos aquellos que me han ayudado aunque no los mencione aquí. A mi compañero de tesis por permitir la realización de este trabajo de diploma.

.... a todos muchas gracias.

***Luis Enrique Sánchez Arce***

A mis padres, hermanos y toda la familia.

A Carmen.

Al gnomo.

Algún día cuando estés  
Como otro más en la cima  
No recordarás la rima  
De este poema tal vez.  
Van a descansar tus pies  
Protegidos sin andar  
Pero querrás saborear  
Como hacha al hombro en un monte  
Un enano el horizonte  
No se cansa de buscar.

***José Ramón Hermosilla Moreno***

*“La función de un buen software es hacer que lo complejo aparente ser simple.”*

**Grady Booch**

## **Resumen**

El Filtrado de Contenidos es una solución técnica al problema que plantea la presencia de contenidos inadecuados en Internet y es utilizado para regular el acceso de los usuarios en empresas, centros de enseñanza y hogares. Filpacon es un software de este tipo, desarrollado en la UCI y diseñado para instalarse en un servidor dedicado Debian GNU/Linux que no tendrá necesariamente un entorno gráfico de usuario. Cuenta además con una base de datos donde se almacena información que es preciso gestionar para su funcionamiento, y aunque esta tarea puede realizarse utilizando algunas herramientas disponibles, no todos los usuarios cuentan con las habilidades necesarias para utilizarlas. En aras de mejorar dicha interacción este trabajo tiene como propósito el desarrollo de una Interfaz de Administración Web (IAW) aprovechando el resultado de fusionar las facilidades que brindan las Interfaces Gráficas de Usuario (GUIs) con el uso de la Web como medio de aplicación. Para lograr dicho objetivo se utilizó el framework Symfony y se eligió RUP para guiar a través de sus fases y flujos el proceso de desarrollo. Los artefactos generados junto a las facilidades que brindó el framework escogido, hicieron posible obtener una aplicación Web amigable, funcional y eficiente, que permite realizar las tareas administrativas de Filpacon de modo más sencillo.

## **Palabras claves**

Filpacon, Interfaz de Administración Web, framework, Symfony

# Índice general

---

<b>Agradecimientos</b>	<b>I</b>
<b>Dedicatoria</b>	<b>II</b>
<b>Resumen</b>	<b>IV</b>
<b>Introducción</b>	<b>1</b>
<b>1. Fundamento teórico</b>	<b>6</b>
1.1. Trabajos similares . . . . .	6
1.1.1. Ámbito internacional . . . . .	6
1.1.2. Ámbito nacional . . . . .	7
1.2. Interfaces Gráficas de Usuario . . . . .	8
1.3. Aplicaciones Web . . . . .	9
1.3.1. ¿Qué son las aplicaciones Web? . . . . .	9
1.3.2. Historia o evolución de las aplicaciones Web . . . . .	11
1.3.3. Características . . . . .	16
1.3.4. Categorías o tipos . . . . .	16
1.3.5. Arquitectura . . . . .	18
1.4. Tecnologías a utilizar . . . . .	21
1.4.1. Lenguaje y herramienta de programación . . . . .	21
1.4.2. Framework seleccionado . . . . .	23

1.4.3. Herramienta de programación . . . . .	24
1.5. Metodología de desarrollo . . . . .	25
1.5.1. RUP . . . . .	25
1.6. Conclusiones . . . . .	27
<b>2. Características del sistema</b>	<b>28</b>
2.1. Descripción del problema . . . . .	28
2.2. Modelo de Dominio . . . . .	31
2.3. Solución propuesta . . . . .	31
2.4. Requisitos funcionales . . . . .	32
2.5. Requisitos no funcionales . . . . .	33
2.5.1. Relacionados con la calidad . . . . .	33
2.5.2. Relacionados con la Interfaz de Usuario . . . . .	35
2.5.3. Relacionados con el entorno o ambiente . . . . .	36
2.6. Definición de los casos de uso . . . . .	36
2.6.1. Definición de los actores . . . . .	36
2.6.2. Listado de casos de uso . . . . .	37
2.6.3. Diagramas de casos de uso . . . . .	45
2.6.4. Casos de uso expandidos . . . . .	49
2.7. Conclusiones . . . . .	66
<b>3. Análisis y diseño del sistema</b>	<b>67</b>
3.1. Análisis . . . . .	67
3.1.1. Modelo de Clases del Análisis . . . . .	67
3.2. Diseño . . . . .	71
3.2.1. Symphony . . . . .	71
3.2.2. Diagramas de Clases del Diseño . . . . .	81
3.2.3. Diagramas de Interacción . . . . .	85
3.2.4. Descripción de las clases . . . . .	89



3.2.5. Diseño de la Base de Datos . . . . .	89
3.2.6. Interfaz . . . . .	94
3.2.7. Concepción de la ayuda . . . . .	96
3.3. Conclusiones . . . . .	96
<b>4. Implementación y prueba</b>	<b>97</b>
4.1. Implementación . . . . .	97
4.1.1. Diagrama de Despliegue . . . . .	97
4.1.2. Diagramas de Componentes . . . . .	98
4.2. Modelo de prueba . . . . .	102
4.3. Conclusiones . . . . .	103
<b>Conclusiones</b>	<b>104</b>
<b>Recomendaciones</b>	<b>106</b>
<b>Referencia bibliográfica</b>	<b>110</b>
<b>Glosario de términos</b>	<b>111</b>
<b>A. Figuras relacionadas</b>	<b>114</b>
<b>B. Diagramas de Casos de Uso expandido</b>	<b>118</b>
<b>C. Diagramas de clases del Análisis</b>	<b>142</b>
<b>D. Diagramas de clases del Diseño</b>	<b>146</b>
<b>E. Diagramas de Interacción</b>	<b>169</b>
<b>F. Diagramas de Componentes</b>	<b>193</b>

# Índice de figuras

---

1.1. Componentes genéricos de una aplicación Web. . . . .	20
1.2. Patrón de diseño MVC. . . . .	21
2.1. Diagrama Entidad-Relación de la BD. . . . .	29
2.2. Modelo de Dominio del problema. . . . .	31
2.3. Diagrama de Casos de Uso del módulo Usuario. . . . .	46
2.4. Diagrama de Casos de Uso del módulo Dominio de navegación básica. . . . .	47
2.5. Diagrama de Casos de Uso del módulo Categoría. . . . .	47
2.6. Diagrama de Casos de Uso del módulo URL. . . . .	48
2.7. Diagrama de Casos de Uso del módulo Reporte. . . . .	49
3.1. Diagrama de clases del análisis - Casos de Uso insertar, eliminar, actualizar y cambiar contraseña. . . . .	68
3.2. Diagrama de clases del análisis - Casos de Uso cambiar contraseña y modificar mi perfil. . . . .	68
3.3. Diagrama de clases del análisis - Casos de Uso categorizar URL, modificar categoría de URL, eliminar categoría de URL y categorías asociadas de una URL. . . . .	69
3.4. Diagrama de clases del análisis - Casos de Uso adicionar, modificar y eliminar. . . . .	70
3.5. Diagrama de clases del análisis - Casos de Uso adicionar y eliminar. . . . .	70
3.6. Esquema del MVC en Symfony . . . . .	72
3.7. Cadena de filtros. . . . .	74
3.8. Organización de la aplicación . . . . .	75

3.9. Patrón decorador entre el layout y el template . . . . .	76
3.10. Diagrama de flujo del proceso de validación. . . . .	78
3.11. Diagrama de Clases del Diseño - Caso de Uso adicionar usuario. . . . .	81
3.12. Diagrama de Clases del Diseño - Caso de Uso adicionar categoría a una URL. . . . .	82
3.13. Diagrama de Clases del Diseño - Caso de Uso adicionar categoría. . . . .	83
3.14. Diagrama de Clases del Diseño - Caso de Uso adicionar dominio de navegación básica. . . . .	84
3.15. Diagrama de Interacción - Caso de Uso adicionar usuario. . . . .	85
3.16. Diagrama de Interacción - Caso de Uso adicionar categoría a una URL. . . . .	86
3.17. Diagrama de Interacción - Caso de Uso adicionar categoría. . . . .	87
3.18. Diagrama de Interacción - Caso de Uso adicionar dominio de navegación básica. . . . .	88
4.1. Diagrama de Despliegue de la aplicación. . . . .	98
4.2. Diagrama de Componente - Caso de Uso adicionar usuario. . . . .	99
4.3. Diagrama de Componente - Caso de Uso adicionar una categoría a una URL. . . . .	100
4.4. Diagrama de Componente - Caso de Uso adicionar una categoría. . . . .	101
4.5. Diagrama de Componente - Caso de Uso adicionar dominio de navegación básica. . . . .	102
A.1. Evolución de la Web. . . . .	114
A.2. Conceptos alrededor de la Web 2.0. . . . .	115
A.3. Interfaz de Administración de Optenet. . . . .	116
A.4. Interfaz de Administración de PureSight. . . . .	117
C.1. Diagrama de clases del Análisis - Casos de Uso habilitar y deshabilitar navegación. . . . .	142
C.2. Diagrama de clases del Análisis - Casos de Uso login y logout. . . . .	143
C.3. Diagrama de clases del Análisis - Casos de Uso incidencias de usuario, ranking de URLs denegadas, ranking de días de la semana por incidencias, cantidad de incidencias por tipo de usuario, cantidad de sitios por categoría y rankig de usuarios por incidencias. . . . .	144
C.4. Diagrama de clases del Análisis - Casos de Uso regulaciones incorrectas. . . . .	145
C.5. Diagrama de clases del Análisis - Casos de Uso mis incidencias. . . . .	145

D.1. Diagrama de clases del Diseño - Caso de Uso eliminar usuario. . . . .	146
D.2. Diagrama de clases del Diseño - Caso de Uso actualizar usuario por el administrador. . . . .	147
D.3. Diagrama de clases del Diseño - Caso de Uso actualizar usuario por usuario básico. . . . .	148
D.4. Diagrama de clases del Diseño - Caso de Uso cambiar contraseña por el administrador. . . . .	149
D.5. Diagrama de clases del Diseño - Caso de Uso deshabilitar navegación. . . . .	150
D.6. Diagrama de clases del Diseño - Caso de Uso habilitar navegación. . . . .	151
D.7. Diagrama de clases del Diseño - Caso de Uso login. . . . .	152
D.8. Diagrama de clases del Diseño - Caso de Uso logout. . . . .	153
D.9. Diagrama de clases del Diseño - Caso de Uso cambiar contraseña del usuario básico. . . . .	154
D.10. Diagrama de clases del Diseño - Caso de Uso eliminar categoría de una URL. . . . .	155
D.11. Diagrama de clases del Diseño - Caso de Uso actualizar categoría de una URL. . . . .	156
D.12. Diagrama de clases del Diseño - Caso de Uso listar categorías de una URL. . . . .	157
D.13. Diagrama de clases del Diseño - Caso de Uso eliminar categoría. . . . .	158
D.14. Diagrama de clases del Diseño - Caso de Uso actualizar categoría. . . . .	159
D.15. Diagrama de clases del Diseño - Caso de Uso eliminar dominio de navegación básica. . . . .	160
D.16. Diagrama de clases del Diseño - Caso de Uso cantidad de sitios por categoría. . . . .	161
D.17. Diagrama de clases del Diseño - Caso de Uso ranking de días de la semana por incidencias. . . . .	162
D.18. Diagrama de clases del Diseño - Caso de Uso ranking de URLs denegadas. . . . .	163
D.19. Diagrama de clases del Diseño - Caso de Uso incidencia de usuario. . . . .	164
D.20. Diagrama de clases del Diseño - Caso de Uso regulaciones incorrectas. . . . .	165
D.21. Diagrama de clases del Diseño - Caso de Uso ranking incidencias por usuario. . . . .	166
D.22. Diagrama de clases del Diseño - Caso de Uso incidencias por tipo usuario. . . . .	167
D.23. Diagrama de clases del Diseño - Caso de Uso mis incidencias. . . . .	168
E.1. Diagrama de Interacción - Caso de Uso cambiar contraseña administrador a otro usuario. . . . .	169
E.2. Diagrama de Interacción - Caso de Uso administrador cambiar su misma contraseña. . . . .	170
E.3. Diagrama de Interacción - Caso de Uso eliminar usuario. . . . .	171
E.4. Diagrama de Interacción - Caso de Uso deshabilitar navegación. . . . .	172
E.5. Diagrama de Interacción - Caso de Uso habilitar navegación. . . . .	173

E.6. Diagrama de Interacción - Caso de Uso login. . . . .	174
E.7. Diagrama de Interacción - Caso de Uso logout. . . . .	175
E.8. Diagrama de Interacción - Caso de Uso actualizar usuario por administrador. . . . .	176
E.9. Diagrama de Interacción - Caso de Uso actualizar datos por usuario básico. . . . .	177
E.10. Diagrama de Interacción - Caso de Uso cambiar contraseña por usuario básico. . . . .	178
E.11. Diagrama de Interacción - Caso de Uso eliminar categoría de una URL. . . . .	179
E.12. Diagrama de Interacción - Caso de Uso actualizar las categorías de una URL. . . . .	180
E.13. Diagrama de Interacción - Caso de Uso listar las categorías de una URL. . . . .	181
E.14. Diagrama de Interacción - Caso de Uso eliminar categoría. . . . .	182
E.15. Diagrama de Interacción - Caso de Uso actualizar categoría. . . . .	183
E.16. Diagrama de Interacción - Caso de Uso eliminar dominio de navegación básica. . . . .	184
E.17. Diagrama de Interacción - Caso de Uso cantidad de sitios por categoría. . . . .	185
E.18. Diagrama de Interacción - Caso de Uso ranking de días de la semana por incidencias. . . . .	186
E.19. Diagrama de Interacción - Caso de Uso ranking de URLs denegadas. . . . .	187
E.20. Diagrama de Interacción - Caso de Uso incidencia de usuario. . . . .	188
E.21. Diagrama de Interacción - Caso de Uso regulaciones incorrectas. . . . .	189
E.22. Diagrama de Interacción - Caso de Uso ranking incidencias por usuario. . . . .	190
E.23. Diagrama de Interacción - Caso de Uso incidencias por tipo usuario. . . . .	191
E.24. Diagrama de Interacción - Caso de Uso mis incidencias. . . . .	192
F.1. Diagrama de Componente - Caso de Uso cambiar contraseña por el administrador. . . . .	193
F.2. Diagrama de Componente - Caso de Uso eliminar usuario. . . . .	194
F.3. Diagrama de Componente - Caso de Uso deshabilitar navegación. . . . .	195
F.4. Diagrama de Componente - Caso de Uso habilitar navegación. . . . .	196
F.5. Diagrama de Componente - Caso de Uso login. . . . .	197
F.6. Diagrama de Componente - Caso de Uso logout. . . . .	198
F.7. Diagrama de Componente - Caso de Uso actualizar usuario por administrador. . . . .	199
F.8. Diagrama de Componente - Caso de Uso cambiar contraseña por usuario básico. . . . .	200
F.9. Diagrama de Componente - Caso de Uso actualizar datos por usuario básico. . . . .	201

F.10. Diagrama de Componente - Caso de Uso eliminar categorías a una URL. . . . .	202
F.11. Diagrama de Componente - Caso de Uso listar las categoría de una URL. . . . .	203
F.12. Diagrama de Componente - Caso de Uso actualizar las categorías de una URL. . . . .	204
F.13. Diagrama de Componente - Caso de Uso eliminar categoría. . . . .	205
F.14. Diagrama de Componente - Caso de Uso actaulizar categoría categoría. . . . .	206
F.15. Diagrama de Componente - Caso de Uso eliminar dominio de navegación básica. . . . .	207
F.16. Diagrama de Componente - Caso de Uso cantidad de sitios por categoría. . . . .	208
F.17. Diagrama de Componente - Caso de Uso ranking de días de la semana por incidencias. . . . .	209
F.18. Diagrama de Componente - Caso de Uso ranking de URLs denegadas. . . . .	210
F.19. Diagrama de Componente - Caso de Uso incidencia de usuario. . . . .	211
F.20. Diagrama de Componente - Caso de Uso regulaciones incorrectas. . . . .	212
F.21. Diagrama de Componente - Caso de Uso ranking incidencias por usuario. . . . .	213
F.22. Diagrama de Componente - Caso de Uso incidencias por tipo usuario. . . . .	214
F.23. Diagrama de Componente - Caso de Uso mis incidencias. . . . .	215

# Introducción

---

Una Interfaz de Usuario funciona como un “conversor de idiomas” [1] para lograr el entendimiento entre una persona y un sistema computacional. Las GUIs, pertenecientes a la categoría anterior, permiten a través del uso de elementos visuales una interacción amigable con un sistema informático, respondiendo su surgimiento y desarrollo a la necesidad de mejorar el proceso de comunicación entre seres humanos y ordenadores. Aún cuando están estrechamente ligadas a las aplicaciones de escritorio, han ido ocupando un espacio en la Web, tras la evolución de esta, al pasar de un medio puramente informativo a uno de aplicación; debido al desarrollo e influencia de las tecnologías y estándares del Consorcio World Wide Web W3C. A pesar que las aplicaciones Web poseen un menor grado de interactividad con los usuarios finales, han ido mitigando dicha limitante con tendencias actuales como la Web 2.0<sup>1</sup>, y poseen a su favor las siguientes ventajas:

- Necesitan bajos requisitos de hardware y software para los clientes, debido a la practicidad del navegador Web como cliente ligero.
- Su mantenimiento es fácil, y sin gastos asociados a la actualización y distribución en los clientes.
- Son multiplataformas pues funcionan en cualquier medio donde se pueda ejecutar un navegador Web.
- Su desarrollo es económico, sencillo y ágil, cuando se siguen buenas prácticas.

Una interfaz de administración es un caso particular de Interfaz Gráfica de Usuario (GUI), que brinda acceso de forma sencilla y amigable a funcionalidades involucradas en el control, configuración y gestión de la información de un servicio o recurso determinado, según los privilegios que posean los usuarios que

---

<sup>1</sup>Representa la evolución de las aplicaciones tradicionales hacia aplicaciones web enfocadas al usuario final.

hacen uso del mismo. En el marco de un software de Filtrado de Contenidos constituye una herramienta útil para su configuración y gestión. El Filtrado de Contenidos es por su parte, una solución técnica al problema que plantea la presencia de contenidos inadecuados en Internet y es utilizado para regular el acceso de los usuarios en empresas, centros de enseñanza y hogares. Algunos sistemas de filtrado de vanguardia como Optenet<sup>2</sup> y PureSight<sup>3</sup> disponen de una herramienta de administración Web y aunque sus buenos resultados están avalados por la eficiencia del filtrado que realizan, también juega un papel importante la comodidad que representa para los usuarios la presencia de la misma.

Desde el año 2005, en la UCI a solicitud de la Oficina de Seguridad para las Redes Informáticas (OSRI), se desarrolla un software de filtrado que lleva por nombre Filpacon, está diseñado para instalarse en un servidor dedicado Debian GNU/Linux que no tendrá necesariamente un entorno gráfico de usuario. Además cuenta con una base de datos donde se almacena información que es preciso gestionar para su funcionamiento, y aunque esta tarea puede realizarse utilizando algunas herramientas disponibles, no todos los usuarios cuentan con las habilidades necesarias para utilizarlas.

A partir de los planteamientos anteriores y ante las amplias posibilidades de comercialización de Filpacon, se identificó por parte del grupo de proyectos *Filtrado de Contenidos de Internet* perteneciente al polo productivo *Centro de Estudios de Internet*, la necesidad de que los usuarios puedan interactuar con el sistema sin tener conocimientos avanzados, dando lugar al siguiente **problema a resolver**: ¿Cómo mejorar la interacción entre los usuarios y el sistema Filpacon en la realización de las tareas administrativas necesarias para su funcionamiento?

Se puede formular entonces la siguiente **idea a defender**: El desarrollo de una IAW para Filpacon permitirá realizar las tareas administrativas de manera eficiente y sencilla.

La aceptación final de este producto estará ligada en gran medida a la percepción que tengan los usuarios de dicha interfaz, teniendo en cuenta que será el único modo de interacción con el sistema y que el éxito o

---

<sup>2</sup><http://www.optenet.com/>

<sup>3</sup><http://www.icognito.com/>



fracaso de un software está determinado por quienes lo usan, deben considerarse entonces, elementos de calidad y utilidad.

En contraste con el problema y la idea a defender antes planteada, se define como **objeto de estudio** las aplicaciones Web y el **campo de acción** será las aplicaciones Web de administración.

El **objetivo general** puede definirse como: desarrollar una IAW que permita realizar las tareas administrativas de Filpacon de manera sencilla y eficiente. Del mismo se derivan los siguientes objetivos específicos:

- Caracterizar las aplicaciones Web a partir del estado del arte asociado a ellas.
- Analizar y diseñar las funcionalidades de la IAW del sistema Filpacon.
- Implementar las funcionalidades de la IAW del sistema Filpacon.

Para lograr los objetivos propuestos se precisan las siguientes tareas:

- Investigar la existencia de sistemas similares relacionados con el que se va a desarrollar y la importancia que tienen las GUIs en los mismos.
- Definir que se entiende por aplicación Web y analizar su historia, características, categorías y arquitectura.
- Seleccionar las tecnologías y herramientas adecuadas para el desarrollo de la IAW teniendo en cuenta lenguaje de programación, framework a utilizar, Entorno de Desarrollo Integrado (IDE) y metodología de desarrollo.
- Realizar una descripción del problema a resolver y representarlo a través de un diagrama.
- Proponer una estructura para la IAW e identificar, describir y modelar las funcionalidades que esta ha de tener.
- Especificar las pautas de diseño gráfico para la IAW del sistema Filpacon y diseñar la interfaz a partir de las mismas.
- Implementar y probar las funcionalidades especificadas.

En el cumplimiento de las tareas se usarán los siguientes métodos teóricos:

- El **Analítico-Sintético** se aplicará para entender las aplicaciones Web a partir del análisis de las características, categorías y arquitectura que presentan, y para formular conclusiones a través de la síntesis de los conocimientos y resultados obtenidos.
- El **Histórico-Lógico** permitirá una mayor comprensión del estado actual de las aplicaciones Web a partir del análisis de su evolución y las etapas principales por las que han transitado.
- La **Modelación** mediante el lenguaje de modelado UML<sup>4</sup> se utilizará para reflejar la estructura, relaciones internas y características de la solución a través de diagramas.

El presente documento está organizado en cuatro capítulos tal como se describe a continuación.

El primer capítulo está centrado en el estado del arte relacionado con el objeto de estudio. En el se analiza la existencia de soluciones a nivel nacional e internacional que puedan reutilizarse y se abordan temas relacionados con las GUI y las aplicaciones Web. Respecto a esta última se profundiza en aspectos tales como su historia o evolución, características, categorías existentes y arquitectura que poseen. Además se identifican las tecnologías y herramientas, así como la metodología para el desarrollo de la IAW.

En el segundo capítulo se profundiza en el problema a resolver a través de su descripción y se muestra el modelo de dominio generado. Se realiza una propuesta de solución y se identifican los requisitos funcionales y no funcionales que deben tenerse en cuenta. Por último se hace una descripción de los Casos de Uso obtenidos a partir de los requisitos y se presentan los diagramas relacionados con estos.

En el tercer capítulo se presentan los artefactos generados en el flujo de análisis y diseño. Además se exponen las ideas relacionadas con la validación de los datos, el tratamiento de errores y la seguridad, basándose en los mecanismos que el framework Symfony provee para dichas cuestiones y se describen una serie de aspectos relativos a la base de datos, el diseño gráfico de la presentación y la ayuda con que debe contar la aplicación.

---

<sup>4</sup>Es un lenguaje usado para especificar, visualizar y documentar los componentes de un sistema en desarrollo orientado a objetos.

El cuarto y último capítulo se enfoca en la implementación y prueba de la solución propuesta tomando como base los resultados que se presentan en los capítulos anteriores. También ilustra la relación de los principales componentes de la aplicación y como estará desplegada.

---

## Capítulo 1

# Fundamento teórico

---

Las GUIs permiten una interacción amigable entre las personas y los sistemas informáticos. Aunque están estrechamente vinculadas al software convencional, han ido ocupando un lugar en la Web con el surgimiento de las más variadas aplicaciones en este medio. Una IAW constituye un caso particular de aplicación Web, siendo necesario para su comprensión el estudio de aspectos tales como: historia o evolución, características, categorías y arquitectura. En el marco de los Filtros de Contenido la presencia de una IAW permite realizar las tareas administrativas de forma sencilla, precisándose para su desarrollo la selección de tecnologías y herramientas adecuadas.

## 1.1. Trabajos similares

Un Filtro de Contenido es uno o más elementos de software que operan juntos para regular (permitir o denegar) el acceso de los usuarios a determinados materiales que se encuentran en Internet. Filpacon constituye una solución de este tipo y ante la necesidad de una interfaz de administración para el mismo, se precisa el estudio de soluciones similares que puedan existir.

### 1.1.1. Ámbito internacional

En el marco particular de los sistemas de filtrado, el uso de interfaces Web para el proceso de administración está ampliamente extendido. Las figuras A.3 y A.4 muestran las interfaces de administración de los filtros *Optenet* y *PureSight* respectivamente.

Las interfaces de estos productos han influido en gran medida en la aceptación de los mismos, independientemente de la eficiencia del filtrado que realizan, brindando a los usuarios un modo intuitivo para el proceso de administración de sus funcionalidades. Sin embargo *squidGuard*<sup>1</sup>, que es un Sistema de Filtrado libre, carece de una interfaz y requiere que las tareas de configuración y administración se realicen mediante las *shell* o consolas de GNU/Linux, que no todos los usuarios dominan a plenitud.

Las interfaces de los sistemas mencionados, no constituyen una solución que pueda adaptarse a Filpacon. En primer lugar son privativas, provocando que su código no pueda ser reutilizado por terceras personas. Como segundo elemento, debe tenerse en cuenta que constituyen soluciones a la medida que responden a las necesidades particulares de dichos sistemas. En el marco del software libre tampoco existen soluciones reutilizables.

### 1.1.2. **Ámbito nacional**

En el país Filpacon constituye la única solución de su tipo, implicando novedad en cada uno de sus componentes. Dado que es imposible aprovechar alguna solución existente a nivel internacional, es preciso entonces desarrollar una interfaz de administración para suplir la carencia de esta. Una aplicación de escritorio sería una alternativa válida si no constituyera un problema por la pérdida de transparencia y la carga asociada que traería consigo la instalación de una aplicación adicional en cada ordenador desde el que se necesite administrar el sistema junto a los costos asociados al mantenimiento y actualización. Sin embargo, la Web como medio para propiciar la interacción requerida, converge con el uso que hacen de ésta los usuarios al navegar por Internet de modo que una IAW sería la solución adecuada para realizar las tareas asociadas a la administración de Filpacon.

Una IAW puede verse como la intersección de dos entes diferentes, las GUIs y las aplicaciones Web, por lo cual es necesario profundizar en ambos elementos para poder sentar las bases teóricas que permitan llevar a cabo el desarrollo de la misma.

---

<sup>1</sup><http://www.squidguard.org/>

## 1.2. Interfaces Gráficas de Usuario

Una Interfaz de Usuario (UI) es la parte de una aplicación que el usuario ve y con la cual interactúa, puede incluir ventanas, menús y ayuda en línea. Su evolución ha ido en paralelo con los sistemas operativos y actualmente constituye uno de sus principales componentes. Existen dos tipos:

- Interfaces alfanuméricas (intérpretes de mandatos).
- Interfaces Gráficas de Usuario, son aquellas que permiten comunicarse con el ordenador de una forma muy rápida e intuitiva.

El presente trabajo está relacionado con el segundo de estos tipos, por lo que se abordarán con mayor profundidad. En los inicios se desarrollaban pensando solamente en su correcto funcionamiento, luego surgió la necesidad que fueran más usables y después se hizo patente la importancia de la estética en las mismas, momento en que el diseño gráfico entró de lleno en el mundo de las GUIs. No solo deben funcionar, sino también agradar y estar acordes con las inquietudes estéticas que se presenten. Elementos como colores, formas, agrupaciones y comunicación, forman parte indispensable de las mismas [2].

Las GUIs constituyen una de las partes más importante de cualquier programa, a través de ellas se determina cuan fácilmente es posible que el programa haga lo que el usuario desea hacer. Un software muy bueno con una interfaz pobremente elaborada tiene poco valor para un usuario no experto. Lograr un buen diseño exige una gran dedicación, pues generalmente son complejas, difíciles de implementar, depurar y modificar; pero sus características y facilidades han logrado que se conviertan en un estándar en la actualidad [3], algunas de ellas son:

1. Los usuarios pueden ver en la pantalla, gráficos y textos, no complejos códigos.
2. Siguen el paradigma de la interacción objeto-acción permitiendo una relación sencilla con los usuarios y acercándose lo más posible al modelo mental de los mismos.
3. Permiten manipular en la pantalla objetos e información de forma directa.
4. Proporcionan respuesta visual a las acciones del usuario.

## 1.3. Aplicaciones Web

La Web tiene una influencia diaria y permanente en la vida de los seres humanos; es tal que se ha extendido a sectores como la economía, industria, educación, salud, administración pública y entretenimiento. Actualmente es difícil encontrar un área que no halla sido infiltrada por la misma. Uno de los factores existentes para esta omnipresencia descansa en la naturaleza de la Web, que es caracterizada por su disponibilidad global y permanente, así como por el acceso cómodo y uniforme a la gran cantidad de información distribuida y producida por cualquiera en forma de páginas Web [4]. Los criterios anteriores muestran el alcance e impacto de la Web en la sociedad humana y permiten vislumbrar cual es y será su importancia para la vida. Uno de los factores que han impulsado esto es la evolución y desarrollo de las aplicaciones Web.

### 1.3.1. ¿Qué son las aplicaciones Web?

La Web fue diseñada originalmente como un medio puramente informativo pero ha evolucionado hasta convertirse en un medio de aplicación [4], pero antes de analizar cual ha sido su historia y evolución, es importante definir que se entiende como tal.

El primer paso para diferenciar las aplicaciones Web de los tradicionales sitios Web centrados en el contenido consiste, según Bob Baxley [5], en enfocarse en el significado de la palabra aplicación. La mayoría de las definiciones expresan que una aplicación es un software o programa de computadora que tiene como objetivo realizar una tarea específica. A partir de este concepto una aplicación Web puede definirse de forma sencilla como un software al que se accede mediante un navegador Web y cuyo propósito es la realización de una o varias tareas solicitadas por un usuario. La siguiente definición aporta otros elementos importantes a tener en cuenta:

Una aplicación Web es un software basado en tecnologías y estándares de W3C que provee recursos específicos tales como contenidos y servicios a través de de una UI a la que puede accederse utilizando un navegador Web [4].

## Ventajas, desventajas y ejemplos

### Ventajas:

- Son independientes de las tecnologías de hardware y software (sistema operativo) que se utilicen. Una empresa podría realizar un proceso de migración y todo quedaría funcionando correctamente.
- La centralización del código implica que los cambios en el software no tengan riesgos ni incompatibilidades, así como la actualización inmediata a todos los clientes.
- No es necesario descargarlas, instalarlas y configurarlas. Simplemente se acceden a través de un navegador.
- Poseen menos errores ya que todos los clientes utilizan la misma versión y una vez que un error es detectado se corrige rápidamente y todos se benefician.
- Se pueden utilizar diferentes lenguajes de programación del lado del servidor y quedar de modo transparente para los usuarios.

### Desventajas:

- Se precisa de una conexión de red constante, elemento que no está al alcance de todos.
- Los usuarios continúan desconfiando de la seguridad de sus datos personales, relacionados con las compras en línea, cuentas bancarias, entre otras.
- Aún quedan algunas deficiencias frente aplicaciones de escritorio.

En la actualidad existen una serie de alternativas en la Web como reemplazo a aplicaciones de escritorio.

Algunas de las más destacadas son las siguientes: [6]

- Como herramienta ofimática, *Google Docs*<sup>2</sup>.

---

<sup>2</sup><http://docs.google.com/>



- Descompresor de archivos, *Wobzip*<sup>3</sup>.
- Planificador de tareas personales, *Remember The Milk*<sup>4</sup>, *Nozbe*<sup>5</sup> o *Labelia*<sup>6</sup>.
- Gestión de proyectos, *MyQuire*<sup>7</sup>, *Project2manage*<sup>8</sup>.

### 1.3.2. Historia o evolución de las aplicaciones Web

La evolución de las aplicaciones Web ha transformado los sistemas informáticos, rompiendo barreras físicas (debido a la distancia), económicas y lógicas (debido a los diferentes sistemas operativos) y ha abierto un abanico de nuevas posibilidades [7]. En este proceso ha existido una retroalimentación entre el desarrollo de la Web y las aplicaciones que en la misma se pueden construir. Se puede enmarcar su desarrollo en cuatro grandes etapas o generaciones [8] y aunque en la actualidad las cuatro conviven, ya casi nadie utiliza las dos primeras.

La **primera generación** abarca desde el 1992 (nacimiento de la Web) hasta mediados de 1994. En esta etapa se emplea la Web como un medio de comunicación tradicional (libros, revistas, etc.), una simple colección de documentos estáticos. Puede definirse como la Web 1.0 donde las páginas tienen las siguientes características:

- Tiempo de carga rápido. Uso de textos, pocas imágenes y ningún recurso multimedia.
- Navegación poco estructurada.
- Páginas largas que parece que nunca se acaban. Poco uso de enlaces entre páginas.
- Páginas que poseen un contenido educativo o científico.
- Empleo de listas para organizar la información.

---

<sup>3</sup><http://wobzip.filetap.com/>

<sup>4</sup><http://www.rememberthemilk.com/>

<sup>5</sup><http://www.nozbe.com/>

<sup>6</sup><http://labelia.net/>

<sup>7</sup><http://www.myquire.com/>

<sup>8</sup><http://www.project2manage.com/>

- Se pueden visualizar prácticamente en cualquier navegador.

A finales de este período aparece la tecnología *Common Gateway Interface* (CGI), que permite la creación de páginas dinámicas dando lugar a la **segunda generación** que se extiende desde 1995 hasta la actualidad. Su diferencia principal con la primera generación está dada por la incorporación de elementos gráficos:

- Los *iconos* sustituyen las palabras.
- El color de fondo se sustituye por una imagen.
- Los *banners* sustituyen los encabezados de las páginas.

Aunque en esta etapa la mayoría de las páginas continúan siendo estáticas se emplea cada vez más la tecnología CGI, mostrándose mucho más interés en su uso que en el propósito final de la aplicación. La aparición de esta tecnología revolucionó el mundo de las aplicaciones Web y a partir de ese momento comenzarían el proceso de interacción con base de datos para la creación de aplicaciones de entretenimiento y educación, entre otras. Se había evolucionado a una etapa superior, ya se hablaba de la Web 1.5. Las principales características de esta etapa son:

- Tiempo de carga más lento. Empleo excesivo de imágenes y animaciones debido a la novedad de su uso.
- Empleo de tablas, aunque no con el propósito del diseño, sino para mostrar datos tabulados.
- Las páginas poseen una estructura de arriba hacia abajo.
- Prima el uso de tecnologías multimedia (imágenes y sonido), aunque no puedan ser visualizadas por el público correctamente.
- Aún no existe una filosofía de navegación, aunque existe una navegación jerárquica a partir de la página principal.

La **tercera generación** aparece a mediados de 1996. Durante esta etapa ocurre una “explosión” en el número de herramientas relacionadas con la Web. En cuanto a la generación de páginas dinámicas se logró un notable avance por el uso extendido de CGI, pero debido a sus limitantes aparecen nuevas tecnologías. La primera solución fue *Internet Database Connector* (IDC) de *Microsoft*, le siguió *Active Server Page* (ASP) que resultó ser la verdadera revolución de las páginas dinámicas. Después de esto aparecieron otras como: ColdFusion, PHP y Java Server Page (JSP) basada en Java. En esta etapa aún la Web se encuentra en su versión 1.5 pero tuvo a su favor una serie de herramientas que hacían de las aplicaciones Web un fuerte rival frente a las aplicaciones de escritorio. Se utilizaban en la construcción de sistemas empresariales y prácticamente cualquier otro tipo. Las páginas pertenecientes a esta generación son las más comunes en la actualidad. Se caracterizan por:

- Tiempo de carga rápido. Uso de *Cascading Style Sheets* (CSS) y optimización del código *HyperText Markup Language* (HTML).
- Las páginas se limitan para que se puedan visualizar completamente en una pantalla sin tener que realizar desplazamiento (*scroll*).
- Se crean pensando en el usuario final, teniendo como objetivos ofrecer servicios, informar, vender, etc.
- Se tienen en cuenta principios tipográficos y de organización visual de la información.
- Se incorporan los principios de usabilidad y accesibilidad.
- Se comprueba con usuarios reales su funcionamiento.
- Se emplean de forma coherente los colores, las imágenes, tipos de letras, los símbolos e *iconos*, etc.

Esta etapa dejó bases muy sólidas para el desarrollo de la **cuarta generación** que comenzó en 1999 y llega hasta nuestros días, donde la mayoría de las páginas Web se crean a partir de información almacenada en una base de datos. Representa un salto en la forma de ver Internet con la aparición del concepto Web 2.0. Las aplicaciones Web alcanzan su máximo esplendor y son utilizadas tanto en la gestión de la información de las empresas como en sistemas de transferencia de voz y vídeo. Las características de las páginas de esta etapa son:

- Se emplean nuevamente los recursos gráficos.
- HTML evoluciona: se extiende el uso de tecnologías poco empleadas hasta ese momento, como CSS y aparecen nuevas como *Dynamic HTML* (DHTML). Estas tecnologías generan incompatibilidad entre distintos navegadores, a pesar de introducir considerables ventajas.
- Uso de nuevas tecnologías multimedia: Se puede crear un sitio Web sin emplear HTML.
- Crean a los usuarios una experiencia agradable desde que visita la primera página hasta que abandona el sitio.

Aunque el desarrollo actual llega hasta la Web 2.0, se tienen aspiraciones en las cuales ya se incursiona. La Web 3.0 pretende añadirle significado a la Web y promete transformar la experiencia de los usuarios desde todos los puntos de vista, contribuyendo al desarrollo de aplicaciones Web; se asocia en gran medida al término de Web Semántica y aunque no es lo mismo, será la principal característica que se incorpore. Mecanismos con inteligencia artificial serán los encargados de sentar las bases de esta etapa de desarrollo. Hablar de la Web 4.0 (WebOS) puede que suene a ciencia ficción, pero *Raymond Kurzweil*<sup>9</sup>, afirma que para el 2029 las computadoras tendrán la potencia de proceso equivalente al de un cerebro humano. La WebOS como se le atribuye, pretende convertir la Web en un sistema operativo virtual, los usuarios podrán conectarse usando un navegador y trabajar en línea del mismo modo que lo hacen hoy en sus puestos de trabajo. La figura A.1 muestra la evolución de la Web de modo compacto, mostrando en cada etapa las principales características.

## La Web 2.0

La Web 2.0 es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones Web enfocadas al usuario final [9]. Se trata de aplicaciones que generen colaboración y servicios que reemplacen las aplicaciones de escritorio. El término surgió durante una conferencia impartida por Dale Dougherty donde se hablaba del renacimiento y evolución de la Web.

---

<sup>9</sup>Músico, empresario, inventor, escritor y científico de la computación, nacido en algún lugar del estado de Massachusetts.

Web 1.0	Web 2.0	Propósito
DoubleClick	Google AdSense	Servicios Publicidad
Ofoto	Flickr	Comunidades fotográficas
Akamai	BitTorrent	Distribución de contenidos
mp3.com	Napster	Descargas de música
Britannica Online	Wikipedia	Enciclopedias
Sitios personales	Blogs	Páginas personales

Cuadro 1.1: Evolución de aplicaciones hacia la Web 2.0

En la actualidad es un término que está muy de moda y han sido tantas las especulaciones al respecto que han hecho fracasar muchos proyectos, por no comprender realmente su significado. En el año 2007 entró en el mercado de la colaboración y la productividad empresarial para quedarse, ha logrado atrapar la atención de los profesionales encargados de gestionar la información y el conocimiento y derribar las fronteras del espacio de trabajo. Entre las características más relevantes se encuentran:

- Se simula el software de escritorio pero sin la necesidad de instalarlo en la computadora.
- Sitios fáciles, usables y ágiles para localizar la información.
- Colaboración en línea a través de los distintos recursos disponibles.
- Nuevos procedimientos para trabajar, comunicarse y participar en la Web.

Algunos ejemplos demuestran la evolución el proceso de evolución alrededor de la Web 2.0:

Esta tendencia ha estado sustentada en el uso de diferentes tecnologías y estándares como XHTML, la separación del contenido y el diseño a través de las hojas de estilo (CSS) y el uso de Ajax <sup>10</sup> agregando mucha más interactividad con los usuarios. La Web 2.0 no es precisamente una tecnología, sino la actitud con la que se debe trabajar para desarrollar aplicaciones. La figura A.2 muestra la mayoría de los aspectos relacionados con el término Web 2.0.

<sup>10</sup> Acrónimo de Asynchronous JavaScript and XML

### 1.3.3. Características

Las aplicaciones Web poseen una serie de características que las diferencian de las aplicaciones tradicionales. Las que se mencionan a continuación están basadas en las que presenta Roger S. Pressman [10].

A partir de su propia naturaleza son *intensivas de red*, dado que residen en una red (Intranet, Extranet, Internet) y brindan servicios a los usuarios.

En muchos casos una aplicación Web brinda a los usuarios contenidos como textos, gráficos, sonido y vídeo, haciendo que sean *controladas por el contenido*. Junto a esta característica puede identificarse la *estética* y no es de extrañar que el éxito o fracaso de este tipo de aplicaciones esté estrechamente ligado a la apariencia que poseen y la facilidad con que los usuarios puedan interactuar con ellas.

A diferencia del software convencional, que evoluciona con una serie de versiones planificadas y espaciadas en el tiempo, las aplicaciones Web están en *constante evolución*.

Existen otras dos propiedades importantes, la *inmediatez* y la *seguridad*. La primera responde a que el proceso para desarrollar una aplicación Web es generalmente más rápido que el seguido para el software tradicional. La segunda se debe a la necesidad de tener en cuenta los controles de seguridad apropiados para evitar accesos indebidos a la información o funcionalidades que se brinden por parte de los usuarios que acceden a estas aplicaciones.

En sentido general las aplicaciones Web siguen las características anteriores, pero no todas las presentan en el mismo grado. Seguidamente se mencionan cuales son las categorías que existen para este tipo de aplicaciones.

### 1.3.4. Categorías o tipos

Las aplicaciones Web se pueden categorizar de varias maneras y en este aspecto, al igual que en muchos otros, no existe un criterio ampliamente aceptado. En esta sección se presentarán dos nuevos criterios que se

consideran adecuados para lograr una mayor comprensión del tema. El primero, basado en la funcionalidad, es resaltado por algunos autores (Murugesan y Ginige) [11] argumentando su utilidad en la comprensión de los requerimientos necesarios para el desarrollo y despliegue de las mismas. El segundo, está basado en el grado de complejidad o tamaño. Existen además enfoques donde se mezclan los criterios anteriores con la evolución que han tenido este tipo de aplicaciones para realizar una categorización que abarca desde aquellas centradas en el contenido, hasta las que siguen tendencias más actuales en el marco de la Web semántica.

Atendiendo a las funcionalidad que brindan, las aplicaciones Web pueden ser agrupadas en las categorías que se describen a continuación, aunque es importante señalar que algunas pueden clasificarse en varias de ellas.

Las aplicaciones Web *informativas o centradas en el contenido* son precursoras de las actuales. Generalmente las páginas Web que las componen son estáticas y actualizadas a mano mediante herramientas específicas, lo cual puede considerarse un factor de costo cuando son numerosas, esto trae como resultado que usualmente estén desactualizadas. Los principales beneficios son la simplicidad y estabilidad así como los cortos tiempos de respuesta que presentan. Los catálogos de productos y la prensa digital se enmarcan en esta categoría.

El uso de formularios que permiten la realización de peticiones con diversos parámetros y la generación dinámica de enlaces y páginas nuevas de acuerdo con dichas solicitudes, es una de las características de las aplicaciones Web *interactivas*.

Los sistemas bancarios, los de reservación aérea y de los pago, son algunos ejemplos de las aplicaciones *orientadas a transacciones* que permiten a los usuarios una mayor interactividad, teniendo como requisito el uso de bases de datos para el almacenamiento de información y la posibilidad de realizar consultas sobre las mismas de forma eficiente y consistente.

EL manejo de operaciones y procesos son objetivos de las aplicaciones Web *orientadas a flujos de trabajo*. El uso de servicios Web es necesario para poder manejar la interoperabilidad, siendo la necesidad de

flexibilidad y robustez el reto más importante para este tipo de aplicaciones, ejemplificadas en los sistemas B2B<sup>11</sup> de comercio electrónico.

Las aplicaciones Web *colaborativas* están orientadas a grupos de usuarios que colaboran entre sí en la realización de diferentes tareas y que por lo general residen en lugares distantes. Los sistemas de planificación de tareas y las plataformas de aprendizaje electrónico son ejemplos de esta categoría.

En correspondencia con las funcionalidades y objetivos que posea una aplicación Web, será el tamaño y complejidad de la misma. Esto da lugar al segundo criterio antes mencionado, según el cual muchas aplicaciones pertenecen a la categoría de *pequeñas a mediana escala* [12]. Estas utilizan lenguajes *scripts* en los que se mezclan etiquetas HTML, funciones y consultas a las bases de datos que utilizan, además el tiempo de desarrollo es corto y sus requerimientos son bastante sencillos. Sin embargo, si las funcionalidades que se ofrecen son numerosas, se necesita un alto rendimiento y disponibilidad y se trabaja con grandes volúmenes de datos teniendo en cuenta la consistencia de los mismos, entonces puede hablarse de una aplicación a *gran escala*. Los conocimientos y actividades ingenieriles que se requieren son mucho mayores y elementos como la seguridad y usabilidad adquieren más importancia.

En dependencia del tipo de aplicación Web, sus requisitos y complejidad, hay que diseñar la arquitectura a seguir. A continuación se tratan los aspectos que pueden servir de base para ello.

### **1.3.5. Arquitectura**

La calidad final de una aplicación Web está directamente influenciada por la arquitectura de la misma. El pobre rendimiento, la insuficiente mantenibilidad y baja disponibilidad, son algunos de los problemas usualmente causados por arquitecturas defectuosas o mal diseñadas. Cuando se desarrollan aplicaciones Web es preciso tener en cuenta una serie de requerimientos y restricciones que influyen directamente en las decisiones arquitectónicas y es tal la importancia de estas que pueden conducir al éxito o fracaso de la aplicación.

---

<sup>11</sup>Es la abreviatura comercial de la expresión anglosajona *business to business*: comercio electrónico entre empresas.



El término arquitectura es uno de los más polémicos que existen en el ámbito de la ingeniería de software debido al amplio abanico de definiciones que existen del mismo. Según la IEEE [13] la arquitectura de software puede definirse como la organización fundamental de un sistema contenido en sus componentes, las relaciones entre ellos y el ambiente, y los principios que orientan su diseño y evolución.

Es importante señalar que la arquitectura de software de un sistema no puede ser entendida desde un solo punto de vista, por lo que Philippe Kruchten plantea que esta puede modelarse mediante cuatro vistas diferentes y una vista de casos de uso para ilustrar y validar las otras. Este modelo es conocido como 4 + 1 vistas arquitectónicas, las cuales se describen brevemente a continuación.

- Vista de diseño: comprende las clases, interfaces y colaboraciones entre ellas como un reflejo del problema y su solución. Esta vista sustenta los requerimientos funcionales del sistema.
- Vista de procesos: comprende los hilos y procesos que forman el sistema en tiempo de ejecución, teniendo en cuenta los mecanismos de control de concurrencia y sincronización.
- Vista de implementación: está constituida por los componentes y ficheros que componen el software y que son usados para obtener un sistema ejecutable.
- Vista de despliegue: está conformada por los nodos que forman la plataforma de hardware donde el sistema se ejecuta y se enfoca en la distribución, entrega e instalación del mismo.
- Por último, la vista de casos de uso persigue describir el comportamiento del sistema según la visión de los analistas, probadores y usuarios finales. Contiene los Casos de Uso claves para guiar el diseño de la arquitectura en las fases iniciales y para validar las restantes vistas en las fases finales del desarrollo.

### **Componentes genéricos de la arquitectura Web**

Las aplicaciones Web están basadas en varios componentes que se comunican entre sí mediante el principio petición-respuesta. Un componente envía una petición a otro componente y este envía una respuesta a dicha solicitud a través de un canal de comunicación sincrónico. A continuación se muestra una breve descripción de los componentes principales:

- Cliente: generalmente es un navegador Web que es utilizado por un usuario para acceder a la aplicación.
- Servidor Web: es un software específico que soporta varios protocolos entre los cuales están HTTP y HTTPS. Permite además procesar y responder a las peticiones de los clientes.
- Servidor de base de datos: es utilizado para almacenar información persistente.

La figura 1.1 muestra la relación entre ellos.

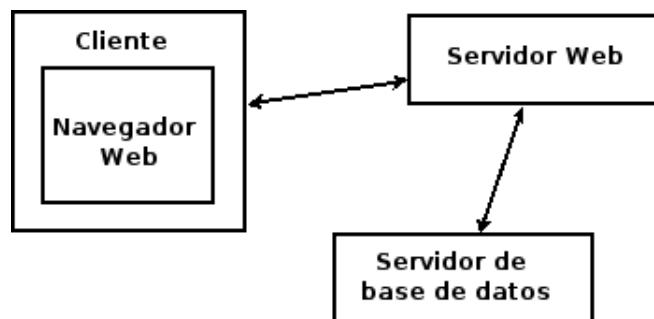


Figura 1.1: Componentes genéricos de una aplicación Web.

### El patrón arquitectónico MVC

Uno de los patrones arquitectónicos más usados en las aplicaciones Web, es el Modelo-Vista-Controlador (MVC), que permite separar la lógica de negocio de la presentación. Este patrón fue desarrollado por Trygve Reenskaug para Smalltalk en los 70's y desde entonces su uso e influencia ha ido creciendo. Cuenta con tres niveles, el *modelo* representa la lógica de negocio de la aplicación y la información con la que se trabaja. La *vista* transforma el modelo en una página Web que permite la interacción del usuario con la aplicación. El *controlador* es el encargado de procesar las interacciones de los usuarios realizando los cambios necesarios en el modelo y mostrando los resultados en la vista.

Cuando un usuario realiza una petición a través de la vista esta es atendida por el controlador, que se comunica con el modelo para realizar las acciones necesarias que permitan darle respuesta. Finalmente el

controlador le da una respuesta al usuario, modificando la vista y mostrando los resultados provenientes del modelo. En la figura 1.2 se muestra el diagrama correspondiente a este patrón.

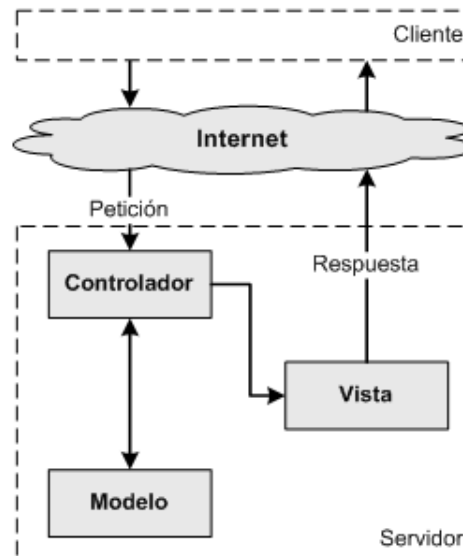


Figura 1.2: Patrón de diseño MVC.

## 1.4. Tecnologías a utilizar

Decidir desde el inicio del desarrollo de un software las tecnologías adecuadas a utilizar, contribuye a evitar el aumento del tiempo requerido y de los costos, así como el descenso de la calidad del producto. Dado que Filpacon es un software que utiliza como base el sistema operativo Debian GNU/Linux la selección de las tecnologías que serán utilizadas para el desarrollo de la IAW ha de hacerse en un marco más estrecho. Analizar las características particulares de la aplicación así como las ventajas que reportaría el uso de cada herramienta son elementos importantes a tener en cuenta en la selección.

### 1.4.1. Lenguaje y herramienta de programación

Para el desarrollo de aplicaciones Web en sistemas Unix/Linux existen diferentes lenguajes de programación, tales como: Perl, Java, PHP, Ruby y Python. La elección de uno de ellos debe sustentarse principalmente en las particularidades de la solución que se proponga y los conocimientos de los desarrolladores de

los mismos. A partir de esto se eligió PHP, teniendo en cuenta sus características, ventajas y desventajas.

#### **Características.** [14]

- Simplicidad: usuarios con experiencia en Perl, C ó C++ podrán utilizarlo rápidamente.
- Velocidad: alta velocidad de ejecución sin introducir demoras en la máquina, bajo consumo de recursos y muy buena integración con Apache.
- Estabilidad: ninguna aplicación está completamente libre de errores (*bugs*), pero con una amplia comunidad de programadores y usuarios corregirlos es mucho más fácil. Posee un sofisticado manejo de variables que lo hacen muy robusto y estable.
- Seguridad: permite la protección contra diversos ataques a través de diferentes niveles de seguridad que pueden ser configurados desde el archivo *.ini*.

#### **Ventajas.** [14]

- PHP es *Open Source* ó código abierto, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no está forzado a pagar actualizaciones anuales para tener una versión que funcione.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, Oracle, Informix, PostgreSQL y otros.
- Actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD.
- PHP corre en “casi cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en aproximadamente 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP,etc) y Macs.

#### **Desventajas** [15].

- No posee un adecuado manejo de la internacionalización y *unicode*.

- Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.
- Por sus características promueve la creación de código desordenado y complejo de mantener.
- Está diseñado especialmente para un modo de hacer aplicaciones Web que es ampliamente considerado problemático y obsoleto (mezclar el código con la creación de la página Web).

Las dos últimas desventajas pueden evitarse mediante la disciplina y el diseño previo de lo que se va a desarrollar. Aunque PHP no obliga a seguir una determinada metodología a la hora de programar, existen formas de obtener código ordenado, estructurado y manejable. Una de ellas es el uso de un framework.

### ¿Porque PHP5?

El éxito y aceptación del lenguaje PHP5 ha sido mayor que el de sus versiones anteriores (PHP3 y PHP4). Los cambios más significativos que introducen están relacionados con la programación orientada a objetos (POO). El uso de modificadores de acceso (*private*, *protected*, *public*) a atributos y métodos, el paso de parámetros por referencia de forma implícita y el soporte para interfaces son algunos de ellos. Además incluye mejoras importantes en el rendimiento y la seguridad. Otro factor clave para su uso fue el anuncio de que PHP4 no tiene soporte desde el 31 de diciembre de 2007 y sólo se corregirán los fallos más importantes relacionados con la seguridad hasta el 8 de agosto de 2008.

#### 1.4.2. Framework seleccionado

Un framework permite reutilizar un diseño para un dominio específico de software rigiendo la arquitectura del mismo. Además define la estructura global de un sistema teniendo en cuenta las clases y objetos, sus responsabilidades claves y como colaboran entre sí, de modo que los diseñadores e implementadores puedan concentrarse en los elementos específicos del mismo [16]. Por último hace la programación más fácil encapsulando operaciones complejas en instrucciones sencillas.

La mayoría de los frameworks para aplicaciones Web escritos en PHP, se basan en el patrón MVC como elemento principal para la arquitectura. Tomando como fuente principal el artículo "Taking a look at ten different PHP frameworks" [17] se han identificado los siguientes: CodeIgniter, ZooP Framework, WACT,

Seagull Framework, CakePHP, Zend Framework, Symfony, Prado, PHP on TRAX y eZ Components. De todos ellos se seleccionó Symfony teniendo en cuenta las características que se mencionan a continuación:

1. Está escrito en PHP5 basándose en el patrón MVC.
2. Cuenta con la versión estable 1.0.
3. El 28 de Octubre de 2006 se anunciaba en el sitio de Symfony que *Yahoo Bookmarks* había lanzado una nueva versión beta de su portal, desarrollado con Symfony, el cual contaba con 12 millones de usuarios y estaba disponible en 12 idiomas.
4. Pueden usarse fácilmente componentes de eZ Components y Zend Framework para proyectos con requisitos cuya solución no está presente en Symfony, lo que brinda junto a otros mecanismos de extensión una gran flexibilidad.
5. Cuenta con una excelente documentación en español que incluye la traducción del libro “The Definitive Guide to Symfony” que posee 19 capítulos en los que se explica todo lo relativo al trabajo con la versión 1.0.
6. Numerosos *plugins* son creados y actualizados constantemente ante las necesidades de los desarrolladores.

### **1.4.3. Herramienta de programación**

Las ventajas principales de un IDE radican en la automatización de algunas tareas asociadas al desarrollo de aplicaciones y al soporte para algunas funcionalidades como el autocompletamiento de código y la depuración, contribuyendo a evitar pérdidas de tiempo. La selección de un IDE para el desarrollo de aplicaciones Web con soporte para PHP se centró en herramientas libres. Aunque existen varios candidatos entre los que están gPHPEdit, Komodo y EclipsePDT, se escogió este último por la experiencia en el trabajo con Eclipse y a partir de las siguientes características [18]:

- Fácil de usar e intuitivo.
- Se adhiere a los estándares de Eclipse.

- Extensibilidad y soporte continuo para el desarrollo de PHP.

Otra característica importante es la integración con subversión como herramienta para el control de versiones.

## 1.5. Metodología de desarrollo

El desarrollo de aplicaciones Web es complejo y presenta más retos que los que generalmente se tienen en cuenta. Para desarrollar y mantener tales sistemas satisfactoriamente se necesita adoptar un proceso de desarrollo disciplinado y una metodología acertada. La ingeniería Web maneja estas necesidades y se puede definir como el uso de principios científicos, ingenieriles y de gestión, junto a métodos sistemáticos, para desarrollar, desplegar y mantener aplicaciones Web de alta calidad. Su esencia es manejar la complejidad y diversidad de los sistemas basados en la Web, evitando fallos potenciales que puedan originar serias implicaciones. Sin embargo, esta disciplina es relativamente joven y no posee sus propios modelos de procesos para desarrollar aplicaciones Web, por tanto muchas personas tratan de adaptar los existentes para el desarrollo de software convencional, proponiendo nuevas metodologías [19]. Algunas de ellas son:

- WAE-Process Conallen (Web Application Extension for UML – Process Conallen).
- OO-Method, con su adaptación a interfaces hipertextuales.
- SOHDM (Scenario-based Object-oriented Hypermedia Design Methodology).
- WebML (Web Modelling Language).
- UML Based Web Engineering (UWE).
- OOWS (Object Oriented Web Solutions).

### 1.5.1. RUP

Teniendo en cuenta la necesidad e importancia de utilizar una metodología, en el caso particular de la Interfaz de Administración de Filpacon se decidió utilizar Rational Unified Process (RUP). La decisión final quedó sustentada por los siguientes elementos:

- Es una de las metodologías más usadas y mayormente probadas a nivel internacional para el desarrollo de software e incluye una extensión para UML que permite el modelado de aplicaciones Web.
- Para mantener un proceso homogéneo entre el desarrollo de la Interfaz de Administración y Filpacon, ya que este usa RUP para la gestión de sus procesos.
- La experiencia del equipo de desarrollo.

Por otra parte el hecho de RUP utilizar como lenguaje de modelado UML, que es orientado a objetos, favorece y se encuentra en concordancia con el desarrollo que se pretende realizar, aunque es válido destacar que esta metodología aún posee algunas desventajas en el desarrollo de aplicaciones Web, relacionadas con requerimientos de navegación y diseño gráfico [20]. Un elemento importante que se debe tener presente es el uso de una herramienta CASE<sup>12</sup> de modo que se seleccionó Visual Paradigm<sup>13</sup> por ser la mejor alternativa en entornos libres.

Entre las características fundamentales de esta metodología están las siguientes [21]:

- Dirigido por Casos de Uso: Tiene a los Casos de Uso como el hilo conductor que orienta las actividades de desarrollo. Se centra en la funcionalidad que el sistema debe poseer para satisfacer las necesidades de un usuario (persona, sistema externo, dispositivo) que interactúa con él.
- Centrado en la arquitectura: Propone arquitectura de forma similar a la de un edificio. Es necesario tener varios planos con diferentes aspectos, para tener una imagen completa del edificio antes que comience su construcción, aquí entra a jugar el término Arquitectura de Software, que abarca las diferentes vistas que se mencionaron anteriormente.
- Iterativo e incremental: Propone la descomposición de proyectos grandes en mini-proyectos, cada mini-proyecto es una iteración, y cada iteración debe estar controlada y tratar un determinado grupo de casos de uso.

---

<sup>12</sup>Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador)

<sup>13</sup><http://www.visual-paradigm.com/>



## **1.6. Conclusiones**

En este capítulo se abordaron todos los elementos teóricos que sustentan la solución del problema, a partir del estudio del estado del arte. Partiendo de la necesidad de contar con una interfaz de administración para Filpacon, se analizaron algunos sistemas similares con el objetivo de reutilizarlas, lo cual es imposible por estar diseñadas teniendo en cuenta las especificidades de los mismos. El hecho de que dichas interfaces son aplicaciones Web, influyó junto a otros factores no menos importantes, en la decisión que la interfaz de administración para Filpacon sea Web. Para ello se eligió PHP5 como lenguaje de programación y Symfony como framework de desarrollo. La herramienta de programación y la metodología seleccionada añadirán como resultado la calidad requerida.

---

## Capítulo 2

# Características del sistema

---

El desarrollo de un software parte de comprender la problemática a la que se quiere dar solución. En RUP la fase de elaboración permite lograr lo anterior a través del levantamiento de requisitos y la realización del Modelo de Negocio o Dominio. En el caso particular de la IAW se partirá de la descripción del problema y los conocimientos que se tienen del mismo por parte del equipo de desarrollo para proponer como estará estructurada esta, dando lugar a los requisitos funcionales y no funcionales. Por último se hará una descripción de los casos de usos y se presentarán los diagramas asociados a estos.

## 2.1. Descripción del problema

En el capítulo anterior se presentó la necesidad de contar con una interfaz de administración para Filpacon y se arribó a la conclusión de desarrollarla como una aplicación Web. El objetivo de la misma es permitir la realización de las tareas administrativas de forma eficiente y sencilla, pero puede surgir la siguiente interrogante: ¿cuáles son las tareas administrativas de Filpacon? Para dar respuesta a la misma es preciso profundizar en el funcionamiento de dicho sistema.

Filpacon parte de disponer de una base de datos (BD) en la que se encuentra una gran cantidad de URLs preclasificadas en distintas categorías que pueden pertenecer a su vez a tres tipos de contenidos diferentes (*adecuados, nocivos e ilícitos*) según las características propias del lugar (cultura, normas, regulaciones, leyes vigentes, etc) donde se encuentre instalado. Ante la solicitud de un recurso de Internet por parte de un usuario, el servidor proxy *squid-cache*, que es parte inseparable del sistema, analiza los parámetros

(identificador del usuario, dirección IP de origen y URL) de la misma a través de un programa hijo llamado *redirector* que interactúa con la BD. Según características del usuario como edad y política de navegación, la existencia o no de la URL en la BD y las categorías y tipo de contenido al que pertenezca la URL, se toma la decisión de permitir o denegar el recurso solicitado.

En la BD se almacena además información relacionada con dominios de navegación básica, las incidencias de los usuarios y los reportes que los mismos pueden hacer de posibles regulaciones incorrectas por parte del sistema. La figura 2.1 muestra el diagrama entidad-relación correspondiente a la BD.

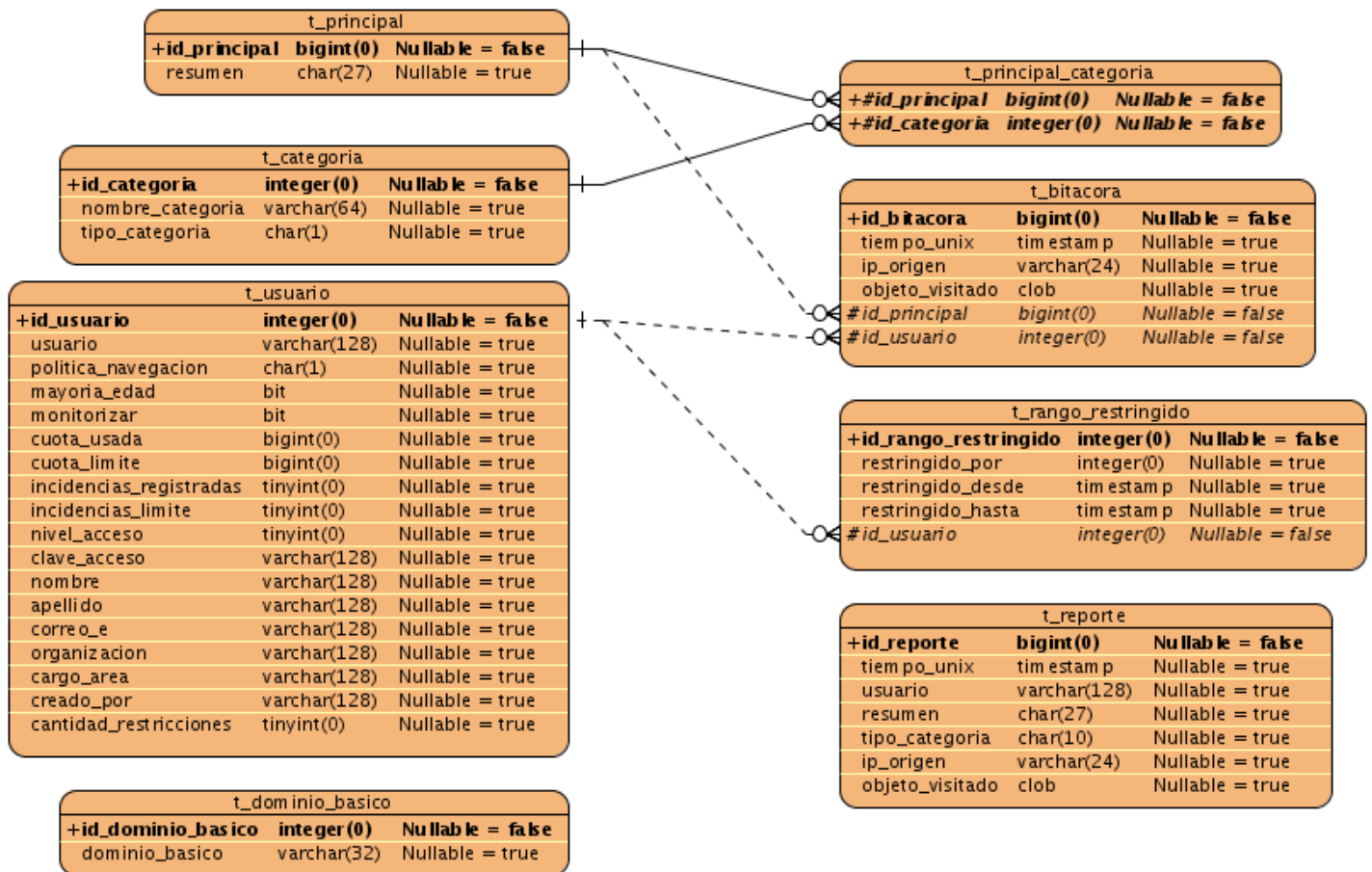


Figura 2.1: Diagrama Entidad-Relación de la BD.

Para que un usuario pueda acceder a Internet a través de Filpacon sus datos deben almacenarse primeramente en la tabla *t\_usuario* de la BD. En este sentido la persona que administre el sistema debe encargarse de adicionar los nuevos usuarios. Podría además modificar los datos y cambiar la contraseña que los mismos utilizan para autenticarse en el proxy, así como eliminar alguno en caso de que sea necesario. Adicionalmente, si un usuario necesita en algún momento consultar algunos de sus datos, modificarlos o cambiar su contraseña, tendría que contactar con dicho administrador para que lo haga. Existen otros dos aspectos importantes relacionados con un usuario que se mencionan a continuación:

- Las incidencias que pueden poseer se almacenan en la tabla *t\_bitacora*, sólo si es monitorizado por el sistema, al acceder a un recurso con contenido no permitido por su política de navegación.
- Las restricciones son intervalos de tiempo que puede estar sin acceso a Internet por superar el umbral especificado por el administrador al adicionarlo al sistema. Estas se almacenan en la tabla *t\_rango\_restringido*.

El administrador del sistema puede además adicionar y eliminar de la tabla *t\_dominio\_basico* los dominios de “navegación básica” que se corresponden con la política que lleva el mismo nombre y que pueden ser usados como “zona segura” para los niños. En cuanto a las categorías a las que puede pertenecer una URL podría realizar las dos tareas anteriores y modificar el tipo de contenido que tienen asignado, las mismas se almacenan en la tabla *t\_categoria*. Además tiene la posibilidad de añadir nuevas URLs al sistema y categorizarlas, sustituir la categoría de una URL por otra, eliminar una categoría o todas las que posee una URL, así como conocer cuales son estas; para ello debería operar con las tablas *t\_principal* y *t\_principal\_categoria*.

Ahora puede darse respuesta a la pregunta planteada, las tareas administrativas de Filpacon son aquellas mencionadas anteriormente. Para realizarlas se puede utilizar la herramienta pgAdminIII, sin embargo, para ello se requieren conocimientos relacionados con SQL, que no necesariamente poseerán todos aquellos que deban administrar el sistema.

## 2.2. Modelo de Dominio

Debido a la relativa simplicidad del entorno donde está enmarcado el sistema y el conocimiento que se posee acerca de su funcionamiento, no es necesario realizar un Modelo de Negocio para comprender la problemática que ha de resolverse, siendo suficiente un Modelo de Dominio (Modelo Conceptual). Este último es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real, no de componentes de software [22]. La figura 2.2 muestra el Modelo de Dominio realizado teniendo en cuenta lo planteado en la descripción del problema.

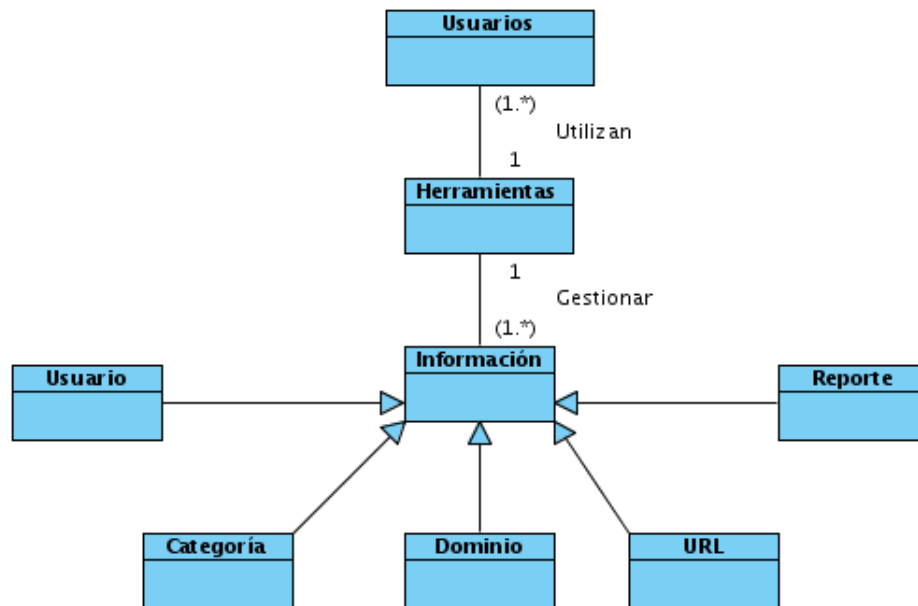


Figura 2.2: Modelo de Dominio del problema.

## 2.3. Solución propuesta

La propuesta para dar solución a la problemática planteada consiste en estructurar la IAW de modo que cuente con cinco módulos, cuatro de ellos relacionados con las tareas que tienen que ver con la gestión de usuarios, dominios básicos de navegación, URLs y las categorías a las que estas pueden pertenecer. El

módulo restante agrupará funcionalidades que permitan obtener reportes a partir de la navegación de los usuarios.

El sistema debe contar con dos roles, un rol de administración que brinde la posibilidad de realizar todas las tareas anteriormente mencionadas y otro básico que permita a cualquier usuario del sistema modificar algunos de sus datos personales, cambiar su contraseña y conocer el historial de accesos a sitios con contenidos inadecuados, que el mismo posea en caso de ser monitorizado. Aunque es un hecho que está implícito, el sistema sólo podrá utilizarse por un usuario si este se ha autenticado, para lo cual se utilizará el mismo identificador y la contraseña que se usa autenticarse en el servidor proxy.

Los requisitos funcionales y no funcionales que se presentan a continuación brindarán más detalles de la solución propuesta.

## **2.4. Requisitos funcionales**

- R1.** Iniciar sesión de usuario.
- R2.** Finalizar sesión de usuario.
- R3.** Insertar nuevo usuario en la Base de Datos.
- R4.** Eliminar usuario existente en la Base de Datos.
- R5.** Actualizar datos de un usuario.
- R6.** Ver perfil del usuario.
- R7.** Cambiar contraseña.
- R8.** Habilitar navegación.
- R9.** Denegar navegación.
- R10.** Insertar categoría de una URL.

- R11. Actualizar categoría de una URL.
- R12. Comprobar categoría de una URL.
- R13. Eliminar categoría de una URL.
- R14. Insertar nueva categoría.
- R15. Eliminar una categoría existente.
- R16. Modificar una categoría existente.
- R17. Establecer dominio de navegación básica.
- R18. Eliminar dominio de navegación básica.
- R19. Mostrar incidencias por usuarios.
- R20. Mostrar ranking de URLs denegadas.
- R21. Mostrar ranking de usuarios por incidencias.
- R22. Mostrar ranking de días de la semana por incidencias.
- R23. Mostrar cantidad de incidencias por tipos de usuarios.
- R23. Mostrar reporte de cantidad de sitios por categorías.
- R25. Mostrar reporte de posibles regulaciones incorrectas.

## **2.5. Requisitos no funcionales**

### **2.5.1. Relacionados con la calidad**

El estándar *ISO/IEC 9126* define un modelo de calidad de software en el que se presentan seis características de calidad que se dividen a su vez en un conjunto de subcaracterísticas. Como uno de los objetivos de esta normativa es ayudar a identificar los requisitos de un software, entonces los elementos de calidad

que ella propone se toman como requisitos no funcionales a cumplir por el sistema, siempre y cuando se mantengan las condiciones especificadas para su uso, ellos son:

1. Debe poseer una funcionabilidad adecuada, o sea, satisfacer los requisitos funcionales declarados e implícitos. Una subcaracterística importante de este requisito es la seguridad que será tratada más adelante.
2. Debe contar con una confiabilidad adecuada, de modo que mantenga el mismo nivel de ejecución a lo largo del tiempo.
3. Debe tener buena usabilidad, de modo que el esfuerzo para usarlo sea mínimo, además debe ser atractivo, útil e intuitivo para los usuarios.
4. La eficiencia debe ser alta, contando con tiempos de respuesta no mayores de 8 segundos. Además el consumo de recursos debe ser mínimo.
5. La mantenibilidad debe ser alta, de modo que pueda adaptarse a condiciones cambiantes del entorno en que se ejecute y las modificaciones puedan hacerse fácilmente.
6. Debe contar con la portabilidad necesaria para poder ser transferido de un ambiente a otro, reemplazado por nuevas versiones y además fácil de instalar y adaptar.

## **Seguridad**

La seguridad es un asunto de interés en el desarrollo de cualquier tipo de aplicación ya que tienen un impacto directo en la funcionabilidad y por tanto en la calidad final. Para las aplicaciones Web este aspecto adquiere dimensiones propias a partir de las características que poseen. Puede descomponerse en los siguientes puntos a tener en cuenta:

- **Confidencialidad:** significa que los datos intercambiados entre el consumidor y el proveedor no deben ser accesibles por terceros.
- **Integridad:** consiste en que nadie puede modificar la información intercambiada.
- **Disponibilidad:** en este marco tiene que ver más con mantener la aplicación accesible todo el tiempo.



- Autenticación: es el proceso mediante el cual se comprueba que alguien o algo es realmente quién dice ser, puede ser persona e incluso otra aplicación.
- Autorización: es el proceso de asignar a un usuario autenticado los privilegios necesarios para acceder a la información, según las características del mismo.

Para poder hacer efectivos los elementos anteriores es preciso cumplir los siguientes requisitos:

1. Deben proveerse mecanismos adecuados para la autenticación y autorización de los usuarios.
2. La aplicación debe contar con mecanismos que prevengan los ataques siguientes: *XSS*, *CSRF*, *SQL Injection* y *Session Fixation*.
3. Debe utilizarse el protocolo HTTPS para garantizar la comunicación segura entre el cliente y el servidor.
4. Las funcionalidades de un administrador no deben ser accedidas por los usuarios comunes y viceversa.

### **2.5.2. Relacionados con la Interfaz de Usuario**

Estos requerimientos definen como los diferentes tipos de usuarios interactuarán con la UI. En el marco de las aplicaciones Web aspectos importantes son la estructura de navegación y la presentación. Mientras estos dos detalles pueden definirse en fases posteriores, las decisiones iniciales acerca de la estrategia a seguir con la interfaz deben tomarse durante la captura de requisitos. Los requisitos siguientes responden al planteamiento anterior.

- Debe diseñarse para 800x600, centrándose en los navegadores que posean una resolución mayor.
- Deben usarse colores apropiados y preferentemente claros, de acuerdo al objetivo que persiguen los Sistemas de Filtrado y acorde al propósito de la interfaz.
- Debe utilizarse un patrón de navegación que permita estructurarla y hacerla intuitiva.
- La interfaz debe tener el mismo diseño para todos los usuarios.

### 2.5.3. Relacionados con el entorno o ambiente

Describen como el sistema estará relacionado con su entorno y las requisitos mínimos para que pueda ser usado.

- El servidor debe poseer las versiones del servidor Web Apache2 y PHP5 iguales o superiores a la 2.2 y la 5.2 respectivamente. Además debe contar con el módulo *libapache2-mod-php5* para permitir el soporte de PHP5 en el servidor Web y con las extensiones *php5-pgsql* y *php5-json* para este lenguaje.
- El servidor Web mencionado anteriormente debe tener habilitado el soporte para SSL.
- El navegador que se utilice debe tener activado el soporte para *cookies* y *javascript*.

## 2.6. Definición de los casos de uso

Los Casos de Uso son la base para la implementación de las fases y disciplinas del RUP. Un Caso de Uso es una secuencia de pasos a seguir para la realización de un fin o propósito, y se relaciona directamente con los requerimientos, ya que un Caso de Uso es la secuencia de pasos que conlleva la realización e implementación de un requerimiento planteado.

### 2.6.1. Definición de los actores

Actores	Descripción
Usuario	Persona que tendrá acceso al sistema como un usuario básico.
Administrador	Podrá ejecutar todas las funcionalidades disponibles relacionadas con el rol de administración.

Tabla 2.1: Definición de los actores

## 2.6.2. Listado de casos de uso

### Módulo Usuario

Caso de Uso:	Iniciar sesión de usuario
Actores:	Administrador, Usuario
Resumen:	El Caso de Uso se inicia cuando el actor introduce los datos para acceder a la aplicación, estos se verifican y finaliza dándole los permisos que tiene y habilitándole o denegándole la entrada al sistema.
Referencia:	R1

Tabla 2.2: CU Iniciar sesión de usuario

Caso de Uso:	Finalizar sesión de usuario
Actores:	Administrador, Usuario
Resumen:	El Caso de Uso se inicia cuando el actor decide salir del sistema.
Referencia:	R2

Tabla 2.3: CU Finalizar sesión de usuario

Caso de Uso:	Insertar usuario
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide insertar un nuevo usuario.
Referencia:	R3

Continúa en la próxima página

Tabla 2.4: CU Insertar usuario

Caso de Uso:	Eliminar usuario
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide eliminar un usuario de la Base de Datos.
Referencia:	R4

Tabla 2.5: CU Eliminar usuario

Caso de Uso:	Actualizar datos de usuario
Actores:	Administrador, Usuario
Resumen:	El Caso de Uso se inicia cuando el Administrador o un Usuario Común deciden cambiar algún dato del perfil de un usuario.
Referencia:	R5

Tabla 2.6: CU Actualizar datos de usuario

Caso de Uso:	Cambiar contraseña
Actores:	Administrador, Usuario
Resumen:	El Caso de Uso se inicia cuando el actor decide cambiar su contraseña.
Continúa en la próxima página	

Referencia:	R7
-------------	----

Tabla 2.7: CU Cambiar contraseña

Caso de Uso:	Habilitar navegación
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide habilitar la navegación de un Usuario que previamente le fue restringida.
Referencia:	R8

Tabla 2.8: CU Habilitar navegación

Caso de Uso:	Denegar navegación
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide restringir la navegación de un Usuario.
Referencia:	R9

Tabla 2.9: CU Denegar navegación

**Módulo URL**

Caso de Uso:	Actualizar categoría de URL
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide actualizar una categoría de una URL.
Referencia:	R11

Tabla 2.10: CU Actualizar categoría de URL

Caso de Uso:	Comprobar categoría de URL
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide determinar si una URL se encuentra categorizada en la Base de Datos y a qué categoría pertenece.
Referencia:	R12

Tabla 2.11: CU Comprobar categoría de URL

Caso de Uso:	Eliminar categoría de una URL
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide eliminar una o todas las categorías a las que pertenece una URL.
Referencia:	R13

Tabla 2.12: CU Eliminar categoría de una URL

**Módulo Categoría**

Caso de Uso:	Insertar categoría
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema con el objetivo de insertar una nueva categoría, que no existe en la Base de Datos.
Referencia:	R14

Tabla 2.13: CU Insertar categoría

Caso de Uso:	Eliminar categoría
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema para realizar la acción de eliminar una categoría de la Base de Datos.
Referencia:	R15

Tabla 2.14: CU Eliminar categoría

Caso de Uso:	Modificar categoría
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema para modificar el contenido correspondiente a una categoría.
Referencia:	R16
Continúa en la próxima página	

Tabla 2.15: CU Modificar categoría

**Módulo Dominios de Navegación Básica**

Caso de Uso:	Establecer dominio de navegación básica
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide establecer dominio de navegación básica.
Referencia:	R17

Tabla 2.16: CU Establecer dominio de navegación básica

Caso de Uso:	Eliminar dominio de navegación básica
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide eliminar un dominio de navegación básica.
Referencia:	R18

Tabla 2.17: CU Eliminar dominio de navegación básica

**Módulo Reportes**



Caso de Uso:	Mostrar las incidencias por usuarios
Actores:	Administrador, usuario
Resumen:	El Caso de Uso se inicia cuando los actores desean obtener el reporte de incidencias por usuarios.
Referencia:	R19

Tabla 2.18: Mostrar las incidencias por usuarios

Caso de Uso:	Mostrar ranking de URLs denegadas
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema para obtener un reporte con el ranking de URLs denegadas.
Referencia:	R20

Tabla 2.19: Mostrar ranking de URLs denegadas

Caso de Uso:	Mostrar ranking de usuarios por incidencias
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte con el ranking de usuarios por incidencias.
Referencia:	R21

Tabla 2.20: Mostrar ranking de usuarios por incidencias

Caso de Uso:	Mostrar ranking de días de la semana por incidencias
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte con el ranking de días de la semana por incidencias.
Referencia:	R22

Tabla 2.21: Mostrar ranking de días de la semana por incidencias

Caso de Uso:	Mostrar cantidad de incidencias por tipos de usuarios
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte de la cantidad de incidencias por tipos de usuarios.
Referencia:	R23

Tabla 2.22: Mostrar cantidad de incidencias por tipos de usuarios

Caso de Uso:	Mostrar reporte de cantidad de sitios por categorías
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte de la cantidad de sitios por categorías.
Referencia:	R24

Continúa en la próxima página

Tabla 2.23: Mostrar reporte de cantidad de sitios por categorías

Caso de Uso:	Mostrar reportes de posibles regulaciones incorrectas
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte de las posibles regulaciones incorrectas.
Referencia:	R25

Tabla 2.24: Mostrar reportes de posibles regulaciones incorrectas.

### 2.6.3. Diagramas de casos de uso

Los diagramas de Casos de Uso para los módulos: Usuario, Dominio de navegación básica, Categoría, Url y Reportes se muestran en las figuras 2.3, 2.4, 2.5, 2.6 y 2.7 respectivamente.

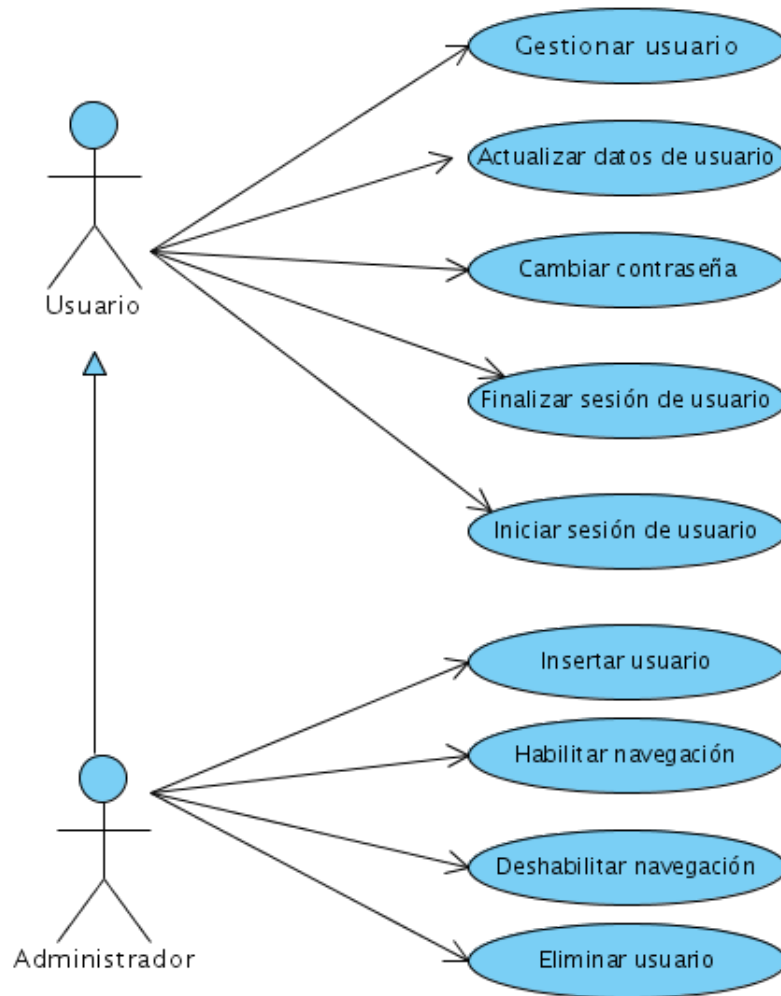


Figura 2.3: Diagrama de Casos de Uso del módulo Usuario.

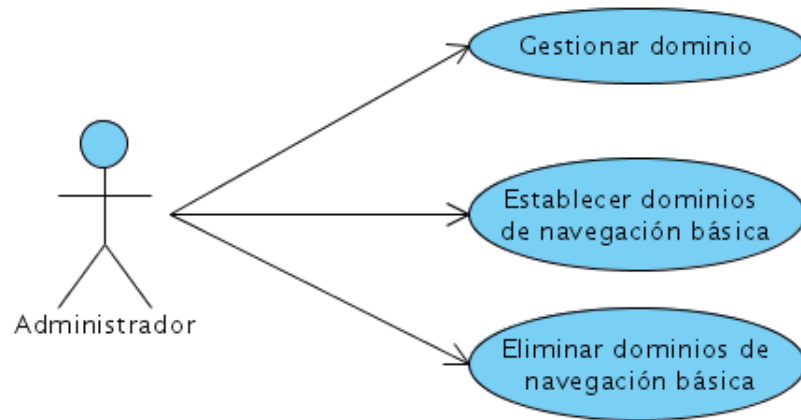


Figura 2.4: Diagrama de Casos de Uso del módulo Dominio de navegación básica.

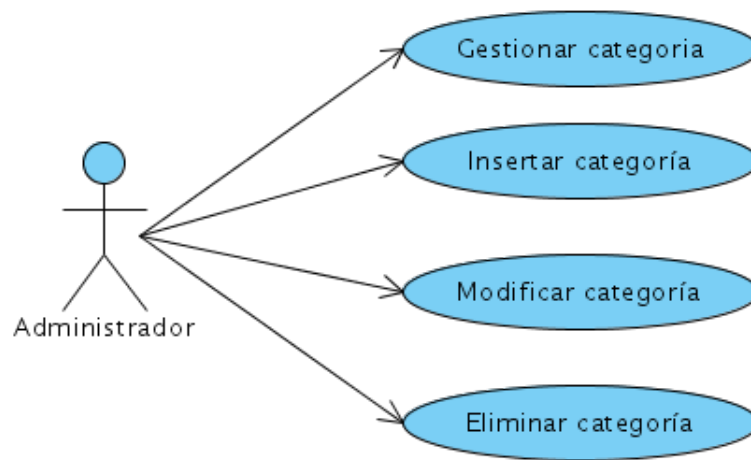


Figura 2.5: Diagrama de Casos de Uso del módulo Categoría.

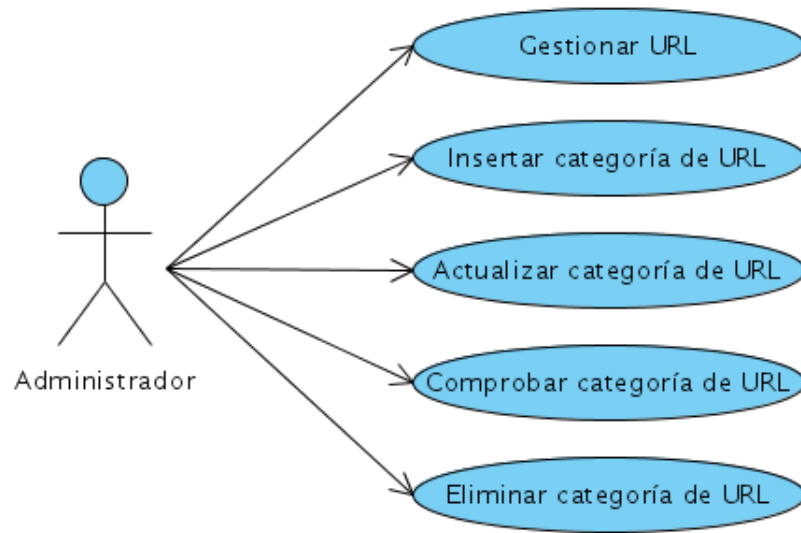


Figura 2.6: Diagrama de Casos de Uso del módulo URL.

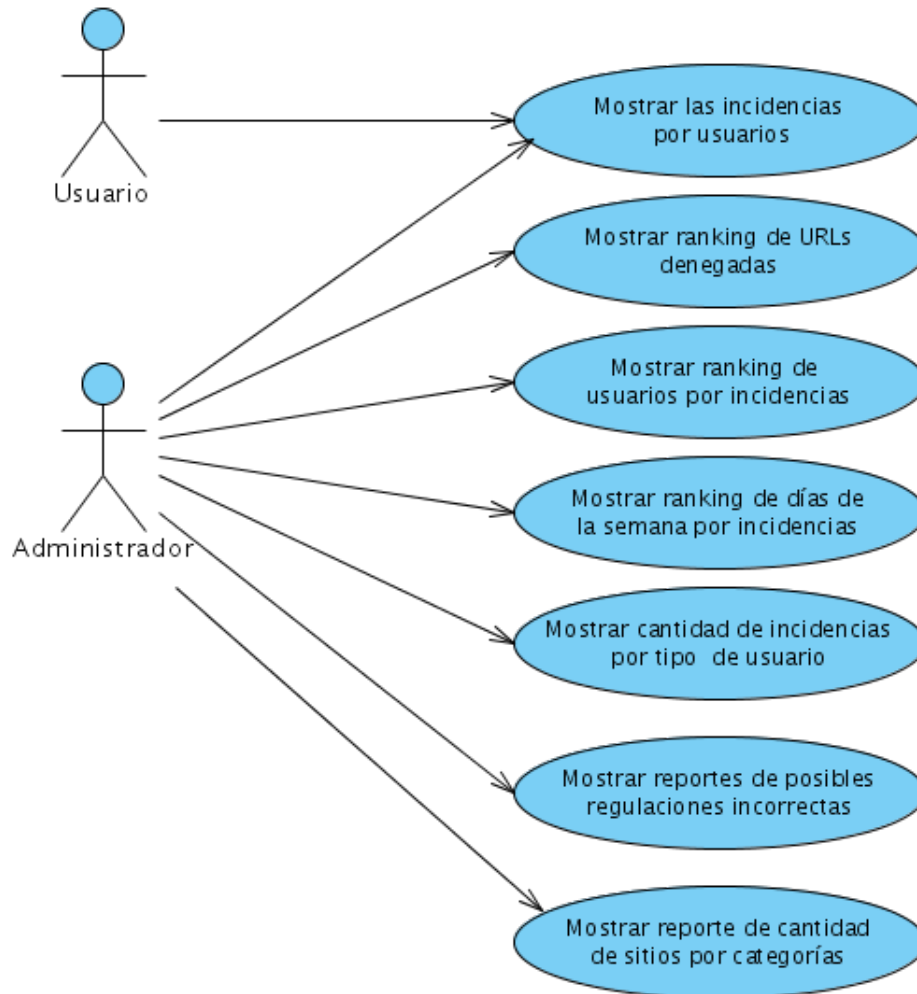


Figura 2.7: Diagrama de Casos de Uso del módulo Reporte.

## 2.6.4. Casos de uso expandidos

### Módulo Usuario

Caso de Uso:	Insertar usuario
Actores:	Administrador
Continúa en la próxima página	

Resumen:	El Caso de Uso se inicia cuando el Administrador decide insertar un nuevo usuario.	
Referencia:	R3	
CU asociados:		
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “nombre del escenario”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<ol style="list-style-type: none"> <li>1. El Administrador ejecuta la opción de insertar un usuario.</li> <li>2. El Administrador entra la información del usuario: Identificador de usuario, Nombre, Apellidos, Correo electrónico, Organización, Cargo área, Contraseña, Grupo (Administrador, Usuario Común), Política de navegación (Ninguna, Plena, Básica, Restrictiva, Permisiva), Cuota límite, Límite de incidencias, Mayoría de edad, Monitorizar.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema muestra el formulario de entrada de datos.</li> <li>2.1 El sistema valida los datos.</li> <li>2.2 El sistema verifica que el identificador no coincide con el de otro usuario.</li> <li>2.3 El sistema inserta el usuario.</li> </ol>	
<b>Flujo alterno</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
Continúa en la próxima página		



	<p>2.1 En caso de faltar algún campo por completar, se muestra un mensaje de advertencia y regresa al paso 2.</p> <p>2.2 En caso de coincidir el identificador con el de otro usuario, se muestra un mensaje de advertencia y regresa al paso 2.</p>
Poscondiciones:	El usuario finalmente es insertado en la Base de Datos del sistema.
Prioridad:	Crítico
Especificaciones:	
Complementaria:	

Caso de Uso:	Eliminar usuario
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide eliminar un usuario de la Base de Datos.
Referencia:	R4
CU asociados:	
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<ol style="list-style-type: none"> <li>1. El Administrador solicita la opción de eliminar un usuario.</li> <li>2. El Administrador entra el identificador del usuario.</li> <li>3. El Administrador da click en el botón eliminar.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema muestra una interfaz para que entre el identificador del usuario.</li> <li>2.1 El sistema busca el Usuario en la Base de Datos dado el parámetro.</li> <li>3.1 El sistema elimina el usuario.</li> </ol>
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<ol style="list-style-type: none"> <li>2.1 Si el Usuario no es encontrado en la Base de Datos, se muestra un mensaje de advertencia y regresa al paso 2.</li> </ol>
Poscondiciones:	El usuario finalmente es eliminado de la Base de Datos del sistema.
Prioridad:	Crítico
Especificaciones:	
Complementaria:	

Caso de Uso:	Actualizar datos de usuario
Actores:	Administrador, Usuario
Resumen:	El Caso de Uso se inicia cuando el Administrador o un Usuario Común deciden cambiar algún dato del perfil de un usuario.
Referencia:	R5
Continúa en la próxima página	

CU asociados:	
Precondiciones:	
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El Actor solicita la opción de actualizar datos de un Usuario.</p>	<p>1.1 Si el rol del que realiza la acción es Administrador, ir a la sección “<b>Actualizar usuario por un Administrador</b>”.</p> <p>1.2 Si el rol que realiza la acción es de un Usuario Común, ir a la sección “<b>Actualizar usuario por un Usuario Común</b>”.</p>
<b>Sección “Actualizar usuario por un Administrador”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<p>2. El Administrador introduce el identificador del usuario al cual desea modificar (puede modificar su propio perfil).</p> <p>3. El Administrador modifica los datos (excepto el usuario y la contraseña, puede modificar cualquier dato).</p>	<p>2.1 El sistema busca los datos del Usuario.</p> <p>2.2 El sistema muestra el perfil relacionado con el Usuario a modificar.</p> <p>3.1 El sistema valida los datos entrados.</p> <p>3.2 El sistema guarda los cambios realizados en la Base de Datos.</p>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>2.1 Si el sistema no encuentra el usuario solicitado por el Administrador, notifica al Administrador y brinda la posibilidad de volver a introducir el identificador del usuario.</p> <p>3.1 En caso de error notifica al Administrador y brinda la posibilidad de volver a introducir el identificador del usuario.</p>
<b>Sección “Actualizar usuario por un Usuario Común”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<p>2. El Usuario solicita la acción de ver su perfil.</p> <p>3. El Usuario modifica los datos permitidos, estos son los relacionados con el Nombre, Apellidos, Correo Electrónico, Organización y Cargo o Área.</p>	<p>2.1 El sistema busca los datos del Usuario.</p> <p>2.2 El sistema muestra el perfil relacionado con el Usuario.</p> <p>3.1 El sistema valida los datos entrados.</p> <p>3.2 El sistema guarda los cambios realizados en la Base de Datos.</p>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>3.1 En caso de error notifica al Usuario y brinda la posibilidad de volver a introducir el identificador del usuario.</p>
Poscondiciones:	Los datos de uno o varios usuarios finalmente son modificados y guardados en la Base de Datos del sistema.
Prioridad:	Secundario
Especificaciones:	
Complementaria:	

**Módulo URL**

Caso de Uso:	Insertar categoría de URL
Continúa en la próxima página	

Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide insertar una categoría de una URL.
Referencia:	R10
CU asociados:	
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El Administrador solicita la opción de insertar la categoría de una URL.</li> <li>2. El Administrador introduce la URL.</li> <li>3. El Administrador selecciona la categoría.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema pide la URL a la cual se le va a insertar una nueva categoría.</li> <li>2.1 El sistema muestra las categorías que existen en la Base de Datos, que no han sido asignadas a dicha URL y le pide en cual de las categorías presentadas desea ubicar la URL.</li> <li>3.1 El sistema inserta la categoría de la URL.</li> </ol>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

	1.1 Si el Administrador no selecciona una URL, se regresa al paso 1.
Poscondiciones:	La categoría de URL finalmente es insertada en la Base de Datos del sistema.
Prioridad:	Crítico
Especificaciones:	
Complementaria:	

Caso de Uso:	Actualizar categoría de URL
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide actualizar una categoría de una URL.
Referencia:	R11
CU asociados:	
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<ol style="list-style-type: none"> <li>1. El Administrador solicita la opción de actualizar la categoría de una URL.</li> <li>2. El Administrador introduce la URL a actualizar.</li> <li>3. El Administrador selecciona la categoría (no mas de una) que desea cambiar y la categoría (no mas de una) que va a ser cambiada por ella.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema pide la URL que desea procesar.</li> <li>2.1 El sistema verifica la existencia de la URL en la Base de Datos.</li> <li>2.2 El sistema muestra las categorías que posee dicha URL.</li> <li>2.3 El sistema muestra las categorías existentes en la Base de Datos por las que el Administrador puede sustituir la categoría que selecciona.</li> <li>3.1 El sistema actualiza la categoría de la URL.</li> </ol>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<ol style="list-style-type: none"> <li>1.1 Si el Administrador no selecciona una URL, se regresa al paso 1.</li> <li>2.1 Si no existe la URL en la Base de Datos, el sistema muestra un mensaje de advertencia y regresa al paso 2.</li> </ol>
<b>Poscondiciones:</b>	La categoría de URL finalmente es actualizada y guardada en la Base de Datos del sistema.
Continúa en la próxima página	



Prioridad:	Crítico
Especificaciones:	
Complementaria:	

Caso de Uso:	Eliminar categoría de una URL
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide eliminar una o todas las categorías a las que pertenece una URL.
Referencia:	R13
CU asociados:	
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.

**Flujo Normal de Eventos**

**Sección “nombre del escenario”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El Administrador solicita la opción de eliminar categoría de una URL.</li> <li>2. El Administrador entra la URL.</li> <li>3. El Administrador selecciona la categoría que desea eliminar.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema pide la URL para mostrar las categorías a las cuales pertenece.</li> <li>2.1 El sistema muestra la(s) categoría(s) a la cuál pertenece dicha URL.</li> <li>3.1 El sistema elimina la categoría seleccionada.</li> </ol>

**Flujo alterno**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

	<p>3.1 Si la URL a ser eliminada tiene asociada una sola categoría el sistema muestra un mensaje preguntando si desea eliminar dicha URL de la Base de Datos al mismo tiempo.</p>
Poscondiciones:	La categoría de URL finalmente es eliminada de la Base de Datos del sistema.
Prioridad:	Crítico
Especificaciones:	
Complementaria:	

### Módulo Categoría

Caso de Uso:	Insertar categoría
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema con el objetivo de insertar una nueva categoría, que no existe en la Base de Datos.
Referencia:	R14
CU asociados:	
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<ol style="list-style-type: none"> <li>1. El Administrador entra al sistema para insertar una nueva categoría.</li> <li>2. El Administrador inserta los datos correspondientes a la categoría, estos son: Nombre de la categoría y Tipo de contenido (Ilícito, Nocivo, Adecuado o Deshabilitada).</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema le muestra el formulario para insertar los datos de la categoría.</li> <li>2.1 El sistema valida la información entrada por el Administrador.</li> <li>2.2 El sistema verifica que esta categoría no exista en la Base de Datos.</li> <li>2.3 El sistema inserta la nueva categoría en la Base de Datos.</li> </ol>
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<ol style="list-style-type: none"> <li>2.1 Si la información no es correcta, el sistema envía un mensaje de advertencia y regresa al paso 2.</li> <li>2.2 Si la categoría ya existe en la Base de Datos y el sistema muestra un mensaje de error y regresa al paso 2.</li> </ol>
<b>Poscondiciones:</b>	La categoría finalmente es insertada en la Base de Datos del sistema y clasificada según el contenido de la misma.
<b>Prioridad:</b>	Crítico
Continúa en la próxima página	

Especificaciones:	
Complementaria:	

Caso de Uso:	Eliminar categoría
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema para realizar la acción de eliminar una categoría de la Base de Datos.
Referencia:	R15
CU asociados:	
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.

**Flujo Normal de Eventos**

**Sección “nombre del escenario”**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El Administrador entra al sistema para eliminar una categoría de la Base de Datos.</li> <li>2. El Administrador selecciona la categoría a eliminar.</li> <li>3. El Administrador confirma la acción de eliminar la categoría seleccionada anteriormente.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema muestra las categorías existentes en la Base de Datos.</li> <li>2.1 El sistema muestra un mensaje advirtiendo al Administrador que al eliminar la categoría seleccionada se eliminarán todas las URL clasificadas como tal.</li> <li>3.1 El sistema elimina de la Base de Datos la categoría seleccionada por el Administrador.</li> </ol>

**Flujo alterno**

Continúa en la próxima página

Acción del Actor		Respuesta del Sistema
Poscondiciones:	La categoría es eliminada de la Base de Datos.	
Prioridad:	Crítico	
Especificaciones:		
Complementaria:		

### Módulo Dominios de Navegación Básica

Caso de Uso:	Establecer dominio de navegación básica	
Actores:	Administrador	
Resumen:	El Caso de Uso se inicia cuando el Administrador decide establecer dominio de navegación básica.	
Referencia:	R17	
CU asociados:		
Precondiciones:	Que el usuario se haya autenticado como Administrador del sistema.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “nombre del escenario”</b>		
Acción del Actor		Respuesta del Sistema
		Continúa en la próxima página

<ol style="list-style-type: none"> <li>1. El Administrador selecciona la opción de añadir dominio de navegación básica.</li> <li>2. El Administrador inserta el nuevo dominio.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema muestra un formulario para que el Administrador inserte el nuevo dominio.</li> <li>2.1 El sistema valida la entrada de los datos.</li> <li>2.2 El sistema verifica si el nuevo dominio pertenece a algún subconjunto existente en la Base de Datos.</li> <li>2.3 El sistema inserta el nuevo dominio en la Base de Datos.</li> </ol>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<ol style="list-style-type: none"> <li>2.1 Si los datos no son correctos regresa al paso 2.</li> <li>2.2 Si el nuevo dominio es un subconjunto de otro ya existente en la Base de Datos, el sistema muestra un mensaje de confirmación para saber si aún se desea insertar el dominio, si la acción es confirmada se elimina el dominio sino se regresa al paso 2.</li> </ol>
<b>Poscondiciones:</b>	Los dominios de navegación básica finalmente son establecidos.
Continúa en la próxima página	

Prioridad:	Secundario
Especificaciones:	
Complementaria:	

Caso de Uso:	Eliminar dominio de navegación básica
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide eliminar un dominio de navegación básica.
Referencia:	R18
CU asociados:	
Precondiciones:	Que el usuario se haya autenticado como Administrador del sistema.

**Flujo Normal de Eventos**

**Sección “nombre del escenario”**

Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> <li>1. El Administrador selecciona la opción de eliminar un dominio de navegación básica.</li> <li>2. El Administrador selecciona el dominio que desea eliminar.</li> <li>3. El Administrador confirma la acción de eliminar el dominio seleccionado.</li> </ol>	<ol style="list-style-type: none"> <li>1.2 El sistema le muestra una interfaz para que el Administrador seleccione el dominio que desea eliminar.</li> <li>2.1 El sistema muestra un mensaje pidiéndole confirmación al Adminsitrador para realizar la acción.</li> <li>3.1 El sistema elimina el dominio de la Base de Datos.</li> </ol>

Continúa en la próxima página

Flujo alterno	
Acción del Actor	Respuesta del Sistema
3. El Administrador no confirma la acción de eliminar el dominio.	3.1 El sistema no elimina el dominio de la Base de Datos y regresa al paso 1.2.
Poscondiciones:	El dominio de navegación básica seleccionado por el Administrador es eliminado de la Base de Datos del Sistema.
Prioridad:	Secundario
Especificaciones:	
Complementaria:	

La descripción de los restantes Casos de Uso se encuentran en el anexo B.

## 2.7. Conclusiones

Aunque existen dos formas de representar un problema en la fase de elaboración (Modelo de Negocio y Modelo de Dominio), se pudo reafirmar que la realización del Modelo de Dominio es adecuado cuando hay una familiarización con los conceptos y procesos principales por parte de las personas involucradas en el desarrollo de la solución. De ese modo a partir de la descripción del problema se hizo posible elaborar una propuesta de solución e identificar los requisitos funcionales y no funcionales. Además puede concluirse que la descripción de Casos de Uso elaborada servirá como base para las fases posteriores y será un pilar para el éxito de la solución.



---

## Capítulo 3

# Análisis y diseño del sistema

---

El análisis y diseño de una aplicación mediante RUP, aunque se mantiene durante sus cuatro fases, alcanza una mayor dimensión en la de elaboración. En este sentido el uso de UML como lenguaje de modelado ofrece la notación gráfica necesaria para representar los sucesivos modelos que se obtienen en el proceso de refinamiento. Asociado a estos flujos de trabajo existen varios artefactos que por su importancia constituyen la base para las fases posteriores, aunque la generación de los mismos está motivada por las necesidades y alcances de cada proyecto. Además cuando se diseña una aplicación Web utilizando un framework, es preciso basar las decisiones que se tomen y los artefactos que se generen en la arquitectura que este propone y en sus características principales.

## 3.1. Análisis

### 3.1.1. Modelo de Clases del Análisis

El Modelo de Análisis ofrece una especificación más precisa de los requisitos con el objetivo de refinarlos y estructurarlos. Uno de los aspectos importantes de este, es que se describe en el lenguaje de los desarrolladores y proporciona una visión general del sistema mediante los diagramas de Clases del Análisis.

**Módulo Usuario**

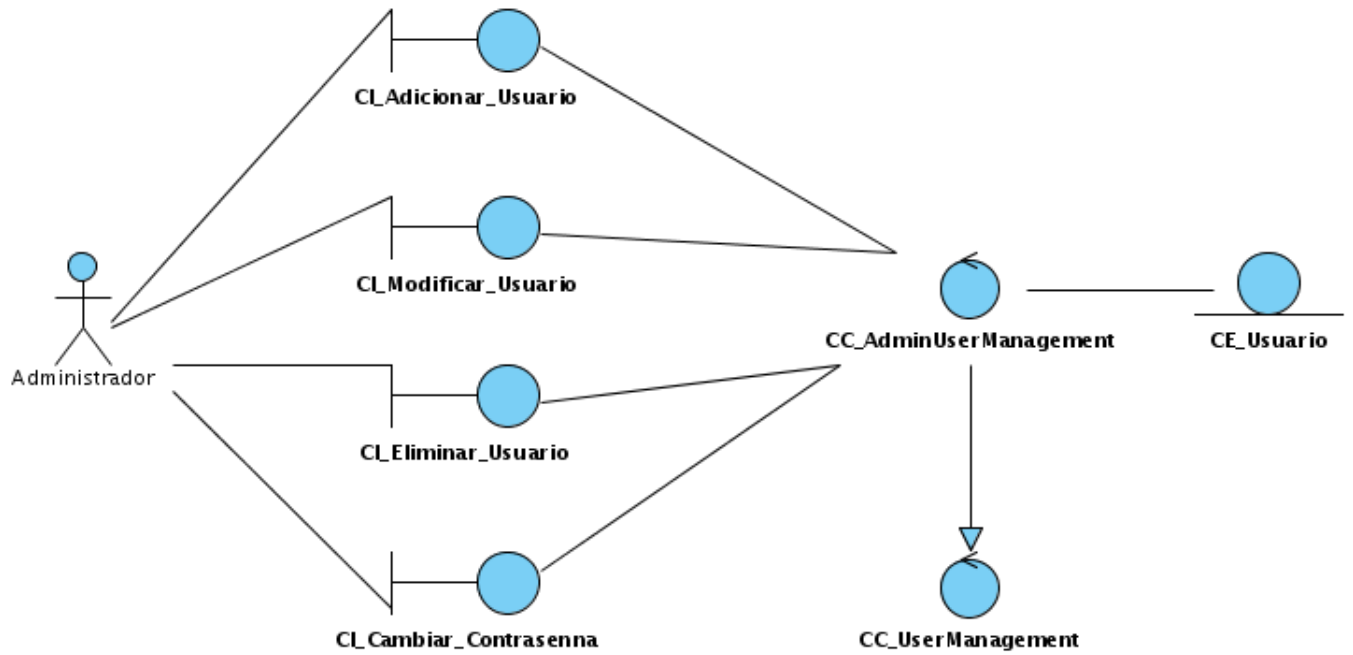


Figura 3.1: Diagrama de clases del análisis - Casos de Uso insertar, eliminar, actualizar y cambiar contraseña.

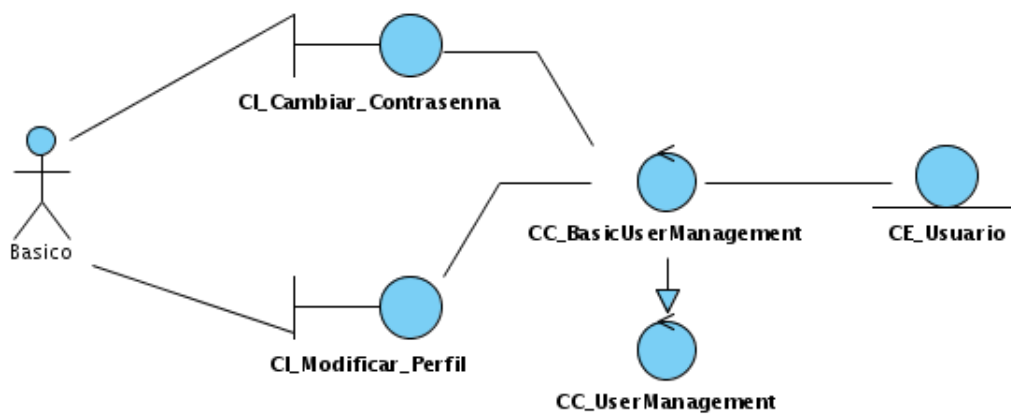


Figura 3.2: Diagrama de clases del análisis - Casos de Uso cambiar contraseña y modificar mi perfil.

**Módulo URL**

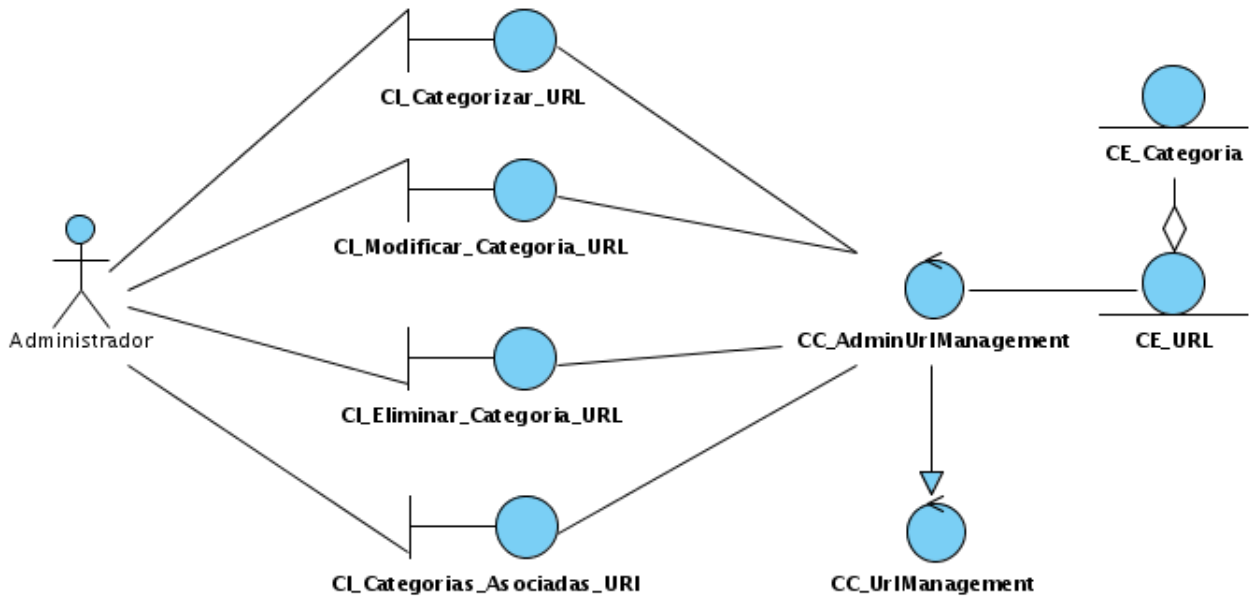


Figura 3.3: Diagrama de clases del análisis - Casos de Uso categorizar URL, modificar categoría de URL, eliminar categoría de URL y categorías asociadas de una URL.

**Módulo Categoría**

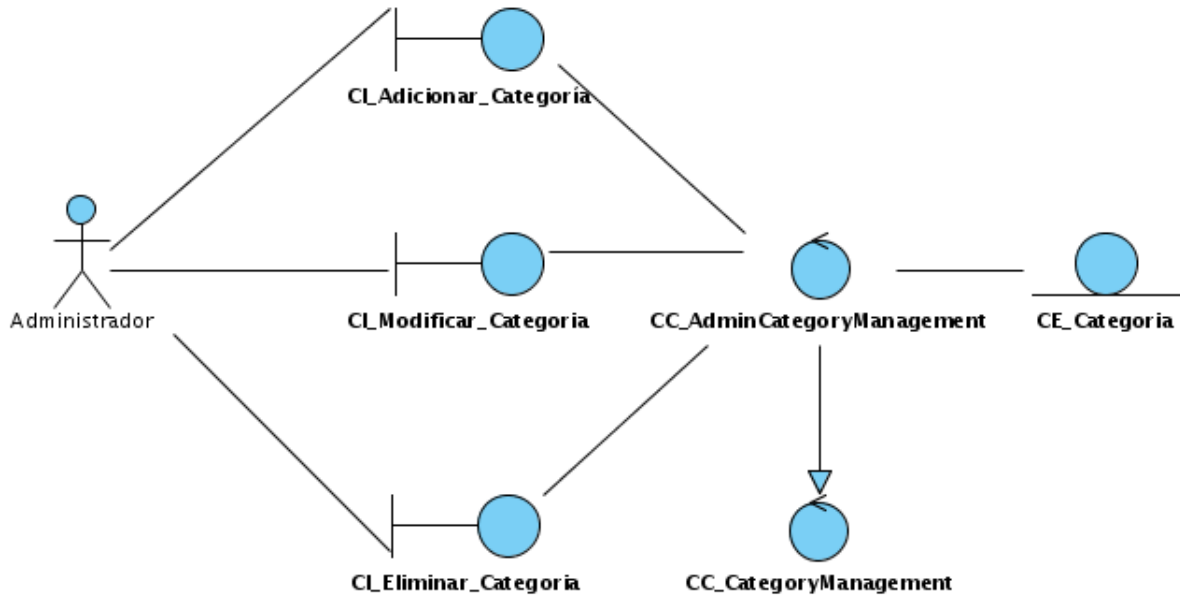


Figura 3.4: Diagrama de clases del análisis - Casos de Uso adicionar, modificar y eliminar.

**Módulo Dominios de Navegación Básica**

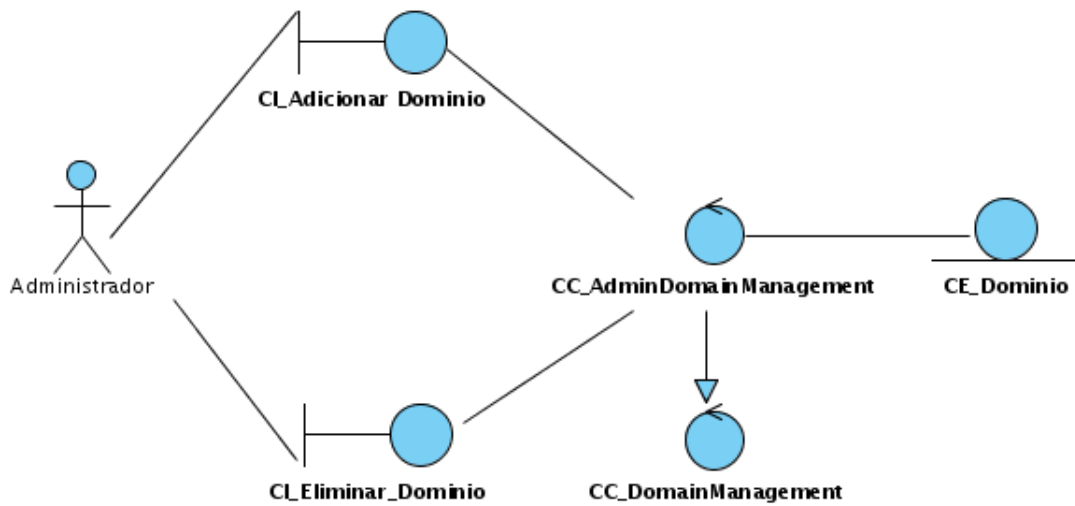


Figura 3.5: Diagrama de clases del análisis - Casos de Uso adicionar y eliminar.

Los restantes diagramas de clases del análisis se encuentran en el anexo C.

## **3.2. Diseño**

El Modelo de Diseño, es un modelo de objetos que describe la realización física de los Casos de Uso, centrándose en como los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar. El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción.

### **3.2.1. Symfony**

Tal como se había mencionado Symfony es un framework diseñado para agilizar el desarrollo de aplicaciones Web a partir de sus características claves. Separa la lógica de negocio, la lógica del servidor y la presentación de la aplicación. Contiene numerosas herramientas y clases con el objetivo de acortar el tiempo de desarrollo de una aplicación compleja y le permite al desarrollador centrarse en los aspectos específicos de la aplicación al automatizar las tareas comunes. El resultado de estas ventajas es que no se necesita reinventar la rueda cada vez que se construye una nueva aplicación.

#### **Implementación del MVC por Symfony**

Symfony ha sido escrito completamente en PHP5 con el objetivo de aprovechar todas las ventajas de esta versión del lenguaje y se basa en el patrón arquitectónico MVC, implementándolo de modo que el desarrollo de aplicaciones sea rápido y sencillo. Además se hace una separación en capas más allá del MVC tal como se muestra en la figura 3.6.

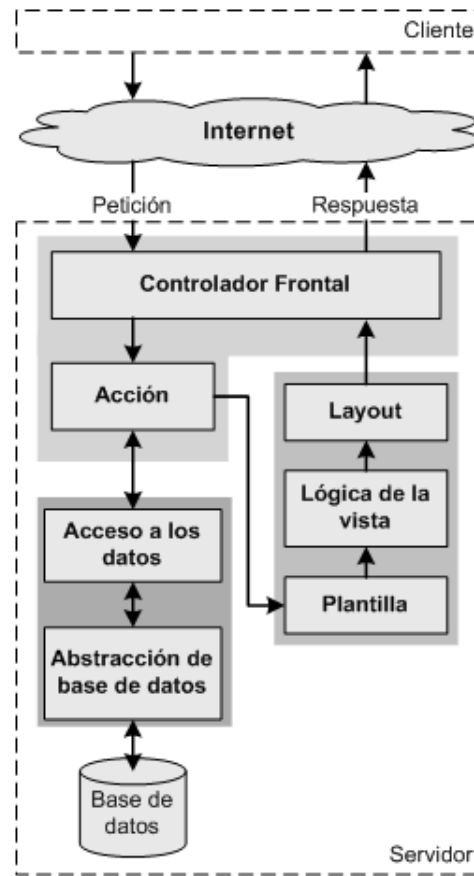


Figura 3.6: Esquema del MVC en Symfony

En esta variante el *controlador* que contiene el código que liga la lógica de negocio con la presentación, está compuesto por el controlador frontal que es el único punto de entrada a la aplicación y las acciones que se encargan de verificar la validez de las peticiones y preparar los datos para la *vista*. Esta última está formada por el *layout* que posee el código común a todas las acciones, las *plantillas* que contienen el código asociado a cada acción en particular y la *lógica* que se puede manipular a través de un fichero de configuración sencillo sin necesidad de programarla. Por último, el *modelo* está formado por la capa de acceso a datos cuyas clases son generadas automáticamente por *Propel* (ORM) en función de la estructura de datos de la aplicación y por la capa de abstracción de la base de datos *Creole*, que es completamente transparente al programador; así si se cambia el sistema gestor de bases de datos en cualquier momento, no

se debe reescribir ni una línea de código, siendo necesario solamente modificar un parámetro en un archivo de configuración.

### **Tareas del controlador frontal**

El controlador frontal es el encargado de despachar las peticiones de los usuarios, pero ello implica algo más que detectar cual es la acción que se ejecuta. A continuación se muestran cuales son las tareas que realiza:

1. Define las constantes del núcleo.
2. Localiza las librerías de Symfony.
3. Carga e inicializa las clases del núcleo del framework.
4. Carga la configuración.
5. Decodifica la URL de la petición para determinar la acción a ejecutar y los parámetros de la petición.
6. Si la acción no existe, redirecciona a la acción *error 404*.
7. Activa los filtros.
8. Ejecuta los filtros, primera pasada.
9. Ejecuta la acción y produce la vista.
10. Ejecuta los filtros, segunda pasada.
11. Muestra la respuesta.

### **Filtros**

Los filtros son un mecanismo que Symfony usa para realizar las tareas que son comunes a toda la aplicación, tales como la validación y la seguridad. De hecho cada petición se procesa como una cadena de filtros que son ejecutados de forma sucesiva. A continuación se mencionan cuales son los filtros que se utilizarán en la IAW de Filpacon y en la figura 3.7 se ilustra la idea a través de un diagrama de secuencia.

1. *sfRenderingFilter*: encargado de renderizar la vista.
2. *sfBasicSecurityFilter*: chequea la seguridad de cada petición.
3. *sfCacheFilter*: controla el mecanismo de cache del framework.
4. *sfCommonFilter*: añade los ficheros *javascript* y *css* a la respuesta.
5. *sfFlashFilter*: elimina las variables *flash* de la sesión.
6. *sfExecutionFilter*: se encarga de validar los parámetros de la petición, la ejecución de la acción y de la vista.

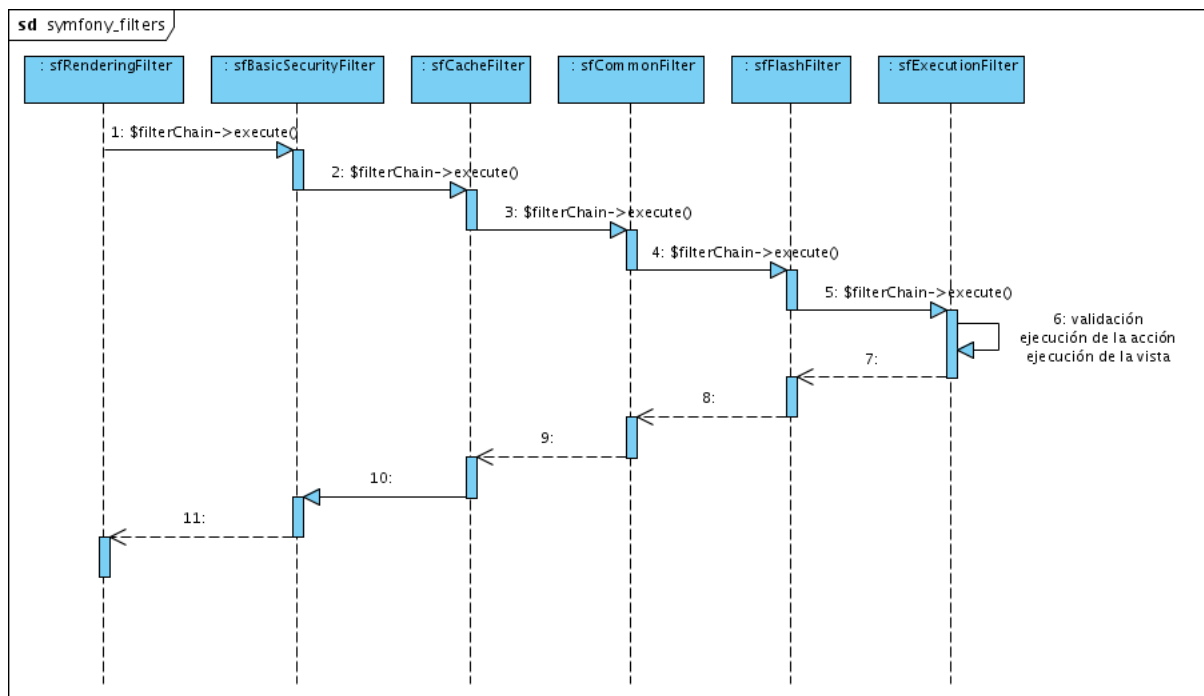


Figura 3.7: Cadena de filtros.

### Organización de la aplicación

Symfony organiza el código fuente en una estructura de tipo proyecto y almacena los archivos del proyecto en una estructura estandarizada de tipo árbol.



En Symfony un proyecto se considera como “un conjunto de servicios y operaciones disponibles bajo un determinado nombre de dominio y que comparten el mismo modelo de objetos”. Dentro de un proyecto, las operaciones se agrupan de forma lógica en aplicaciones y cada aplicación está formada por uno o más módulos. Un módulo normalmente representa a una página Web o un grupo de páginas con un propósito relacionado y almacenan las acciones, que representan cada una de las operaciones que se pueden realizar en un módulo. La IAW seguirá una estructura aproximada a la que se muestra en la figura 3.8.

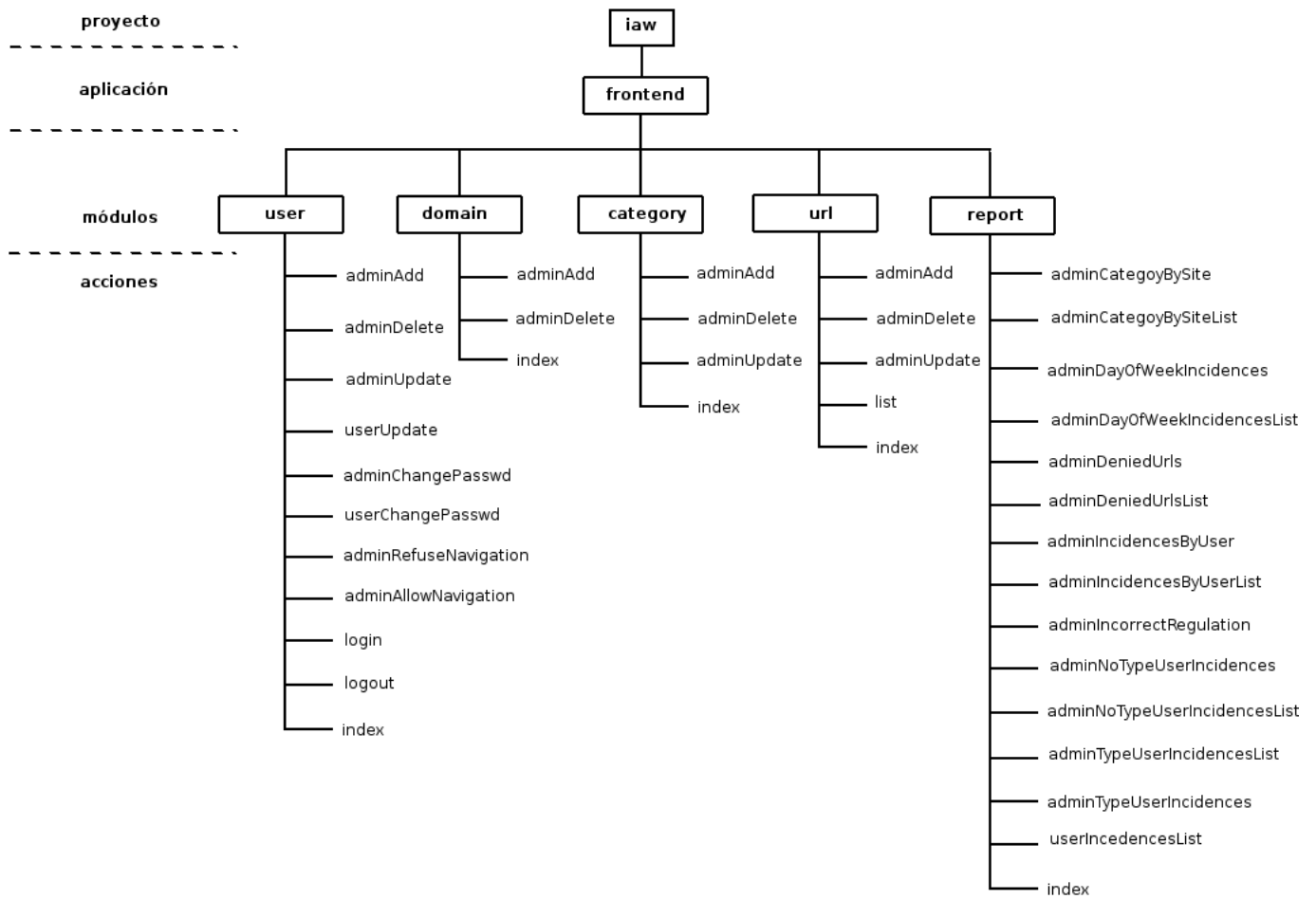


Figura 3.8: Organización de la aplicación

### Patrones de diseño que se usan

Symfony sigue las mejores prácticas y patrones de diseño para el desarrollo de aplicaciones Web. Seguidamente se mencionan cuales son los patrones de diseño más significativos que se utilizan.

Las clases que conforman el núcleo están basadas en el patrón *Factory Method* permitiendo la creación fácil de los objetos y su extensión de forma sencilla.

El patrón *Decorator* es usado para decorar el *layout* de la aplicación con el *template* asociado a cada acción, dando como resultado la vista que es mostrada al usuario. Esto se ilustra en la figura 3.9.

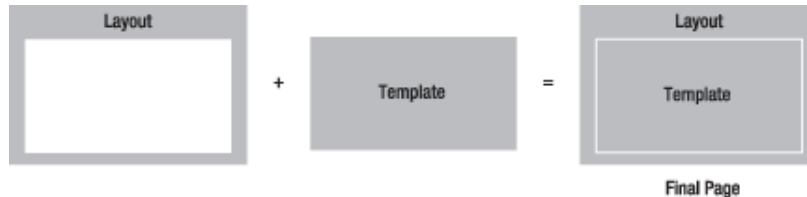


Figura 3.9: Patrón decorator entre el layout y el template

El patrón *Singleton* es usado en Symfony para permitir el acceso desde cualquier lugar de la aplicación a los objetos relacionados con el núcleo del framework.

La ejecución de la secuencia de filtros se basa en el patrón *Chain of Responsibility* que permite que más de un objeto pueda manejar una petición.

Los patrones anteriores pertenecen a los conocidos como GoF<sup>1</sup>. Además de ellos se utilizan otros descritos por Martin Fowler en [23], entre los que están *Front controller*, *Row Data Gateway* y *Table Data Gateway*. Los dos últimos son usados en las clases de acceso a datos y permiten realizar operaciones sobre un registro de una tabla y un conjunto de estos respectivamente.

<sup>1</sup>Gan of Four. Patrones de diseño orientado a objetos

## Validación y tratamiento de errores

La validación y el tratamiento de errores en la aplicación se han diseñado en torno a los mecanismos que Symfony provee.

Posee un mecanismo de validación en el servidor a través del uso de ficheros de validación y una serie de validadores que permite hacer este proceso más fácil y extensible, sin la consecuente promoción de errores que puede traer consigo la programación relacionada con este aspecto. No obstante hay casos especiales donde es necesario hacer la validación manualmente o combinar ambos métodos. Una característica importante de Symfony en este sentido es que permite el relleno automático de datos (*repopulation*), lo que asegura la obtención de datos correctos y mejora la experiencia de los usuarios. Tal como se mencionó anteriormente la validación se realiza como parte de la ejecución del filtro *sfExecutionFilter*. La figura 3.10 muestra como se realiza este proceso.

Para el tratamiento de excepciones se utilizará el mecanismo provisto por PHP y las clases que para este fin Symfony posee.

## Seguridad

### Autenticación y autorización

Cuando se diseña una aplicación Web es importante definir los aspectos relacionados con la seguridad de la misma. En este contexto es importante prestar atención a la autenticación y autorización de los usuarios. Symfony provee todo lo necesario para que ambas cosas puedan realizarse de forma sencilla, mejorando el mecanismo de sesiones que utiliza PHP y haciéndolo más configurable y fácil de usar. Una vez que un usuario se autentica, se le asignan los privilegios necesarios (*credenciales*) para que pueda utilizar las funcionalidades que brinda la aplicación según el rol que desempeñe.

A partir de los requisitos funcionales y no funcionales, y tal como se había planteado, la IAW poseerá dos tipos de usuarios diferentes, un administrador y un usuario común o básico. De este modo solo será necesario definir la seguridad y la *credencial* o *credenciales* que se requieran para la ejecución de cada una de

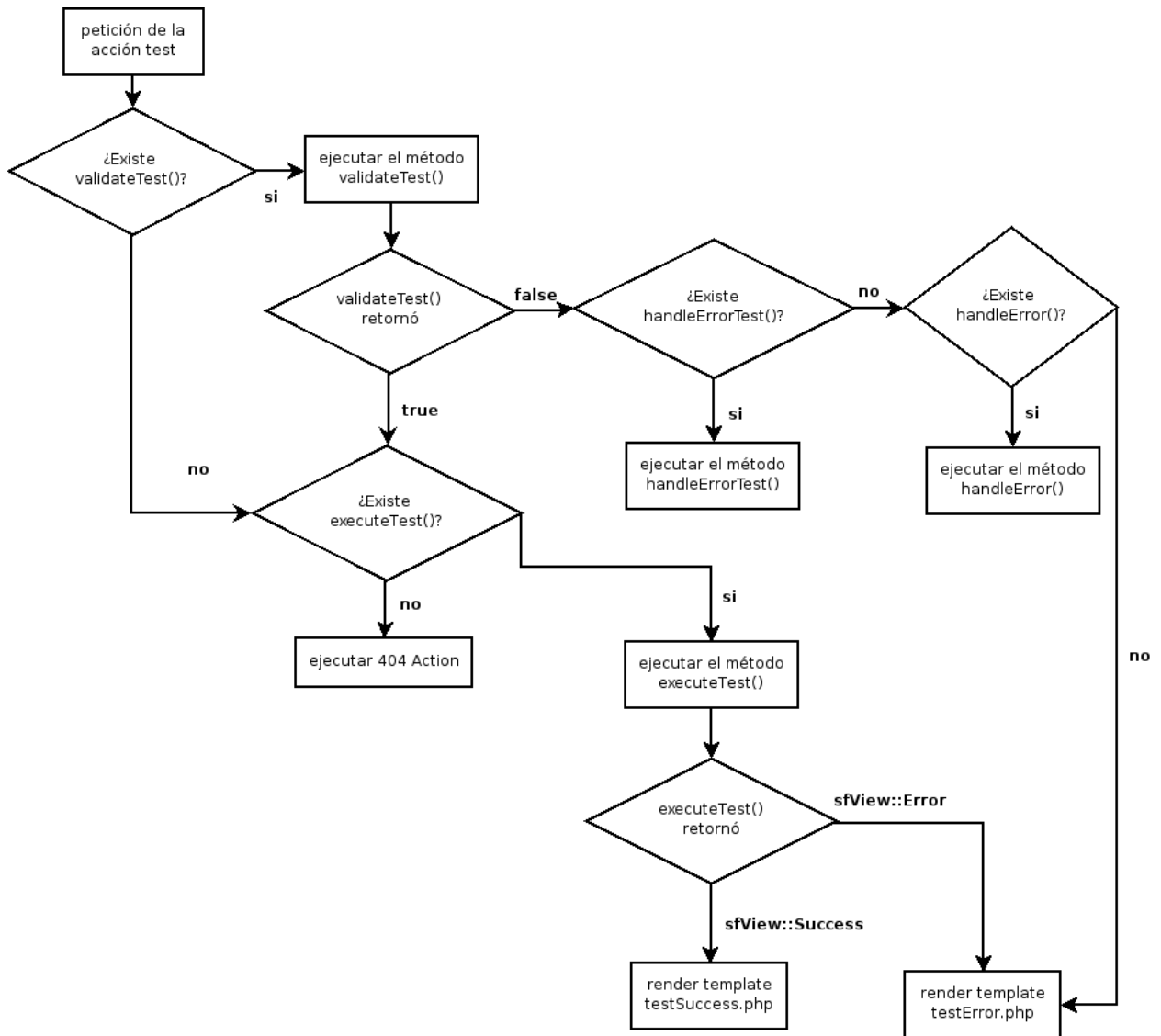


Figura 3.10: Diagrama de flujo del proceso de validación.

las acciones, pues la responsabilidad de chequear que un usuario esté correctamente autenticado y posea los privilegios necesarios será del filtro *sfBasicSecurityFilter*.

## Defensa contra ataques

Existen numerosos ataques informáticos que pueden afectar a las aplicaciones Web. Cuando se produce alguno de ellos es porque se explota alguna vulnerabilidad existente relacionada con los principios de diseño e implementación y con los mecanismos de control o defensa con que se cuentan. A continuación se describen brevemente los principales ataques que podrían afectar la seguridad de la aplicación y por tanto la integridad de los datos que se manejan, así como los principios que regirán la prevención de los mismos en el contexto de Symfony.

*Cross Site Scripting* (XSS) es un tipo de ataque que aprovecha la confianza que un usuario posee en un sitio determinado, se basa en la inyección de códigos hechos en *javascript* que pueden impedir el uso normal de una aplicación. Para su prevención Symfony posee un mecanismo de escape que puede ser activado mediante un parámetro en un fichero de configuración.

*Cross Site Request Forgery* (CSRF) aunque parezca parecido, es contrario al XSS ya que este se basa en aprovechar la confianza que un sitio posee en un usuario para realizar peticiones, al mismo tiempo que puede comprometer su seguridad. Aunque en la versión que se utiliza no posee defensa contra el mismo, existe una extensión que permite añadirlo al framework como un filtro, que se insertará en la secuencia de filtros después del *sfBasicSecurityFilter*.

*SQL injection* es un tipo de ataque que permite realizar acciones con el objetivo de corromper la información de un sistema o ganar acceso al mismo través de la entrada de sentencias SQL maliciosas por medio de formularios o como parámetros de la petición en la URL. EL uso de *Propel* y *Creole* ayuda a prevenirlo, ya que ambos poseen los mecanismos de escape necesarios para este tipo de ataque.

La mayor preocupación relacionada con el uso de sesiones está relacionada con el identificador de la sesión. Generalmente para obtener un identificador válido, un atacante puede utilizar tres variantes, la pre-

dicción, la captura y una tercera conocida en inglés como *Session Fixation*. La primera y la segunda pueden ser mitigadas por el carácter altamente aleatorio que posee la generación de identificadores de sesión en PHP y por el uso de SSL para la transmisión de datos entre cliente y servidor. Sin embargo, en la última el atacante puede lograr que la víctima use un identificador de sesión elegido por él, para así obtener un identificador válido. Como medio para prevenirlo no se utilizará la URL para enviar el identificador de sesión y se regenerará uno nuevo para cada usuario una vez que se autentique en la aplicación.

Tal como se mencionó anteriormente el uso de SSL para la transmisión de los datos contribuye a la integridad de los mismos y a las vez fortalece la defensa prevista contra los ataques anteriores. Por ello la aplicación utilizará el protocolo HTTPS para todas sus funcionalidades.

Otros ataques que pueden afectar la aplicación es la denegación de servicios (DoS) a partir del envío masivo de peticiones al servidor, pueden ser prevenidos a través de reglas en el cortafuegos que utiliza el sistema Filpacon y con las configuraciones del servidor Web relacionadas con la cantidad de usuarios permitidos de forma concurrente, en relación con el consumo de la aplicación y los recursos que este posea.

La protección contra el acceso físico al servidor recae en la institución donde se utilice el Sistema de Filtrado y en los controles de seguridad que ella posea.

### 3.2.2. Diagramas de Clases del Diseño

#### Módulo Usuario

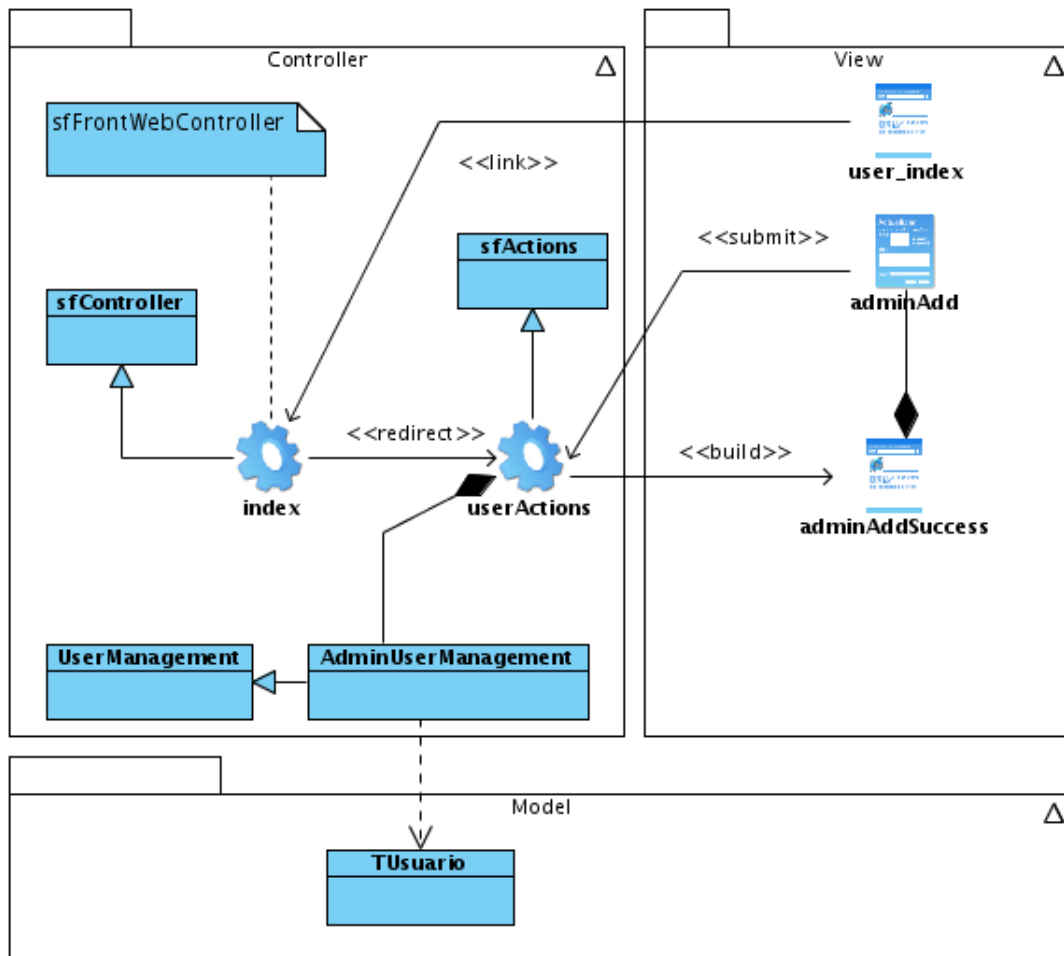


Figura 3.11: Diagrama de Clases del Diseño - Caso de Uso adicionar usuario.

Módulo URL

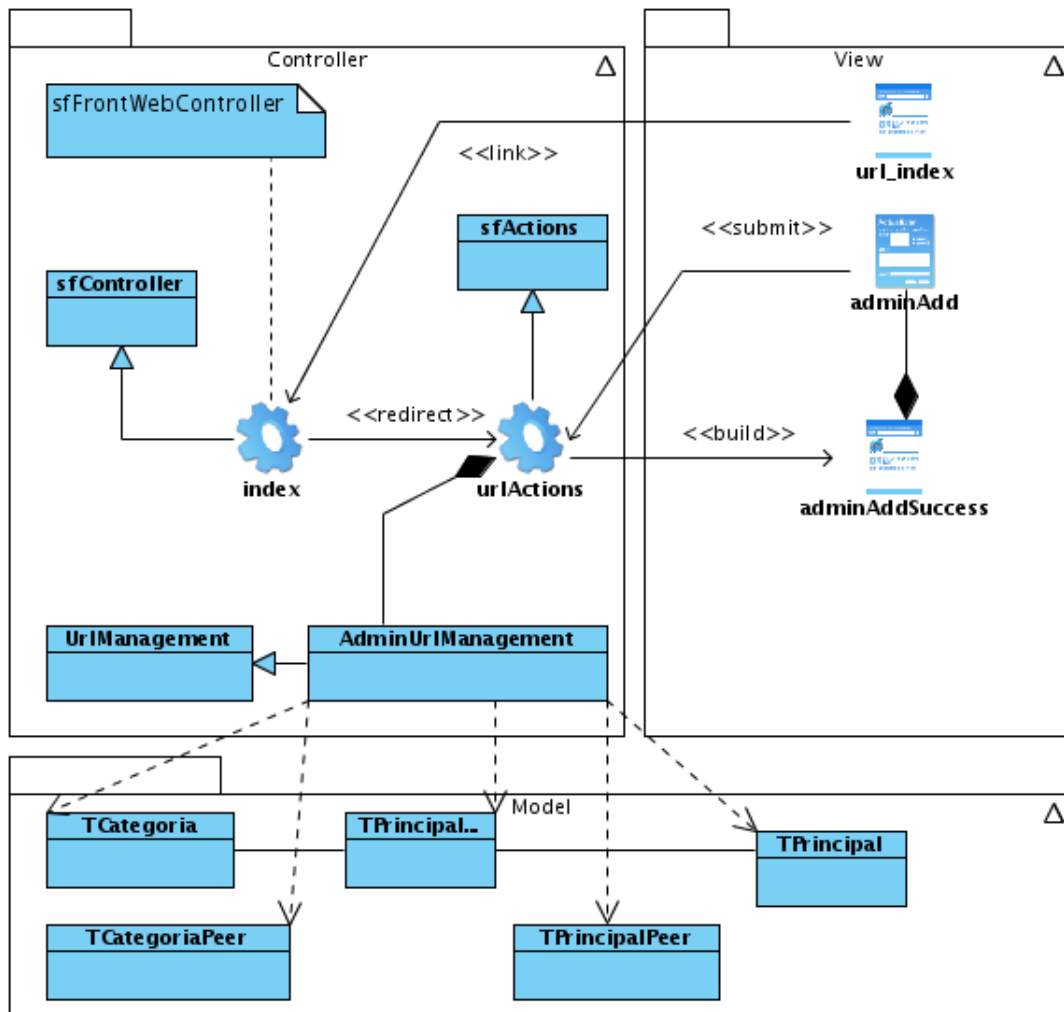


Figura 3.12: Diagrama de Clases del Diseño - Caso de Uso adicionar categoría a una URL.



### Módulo Categoría

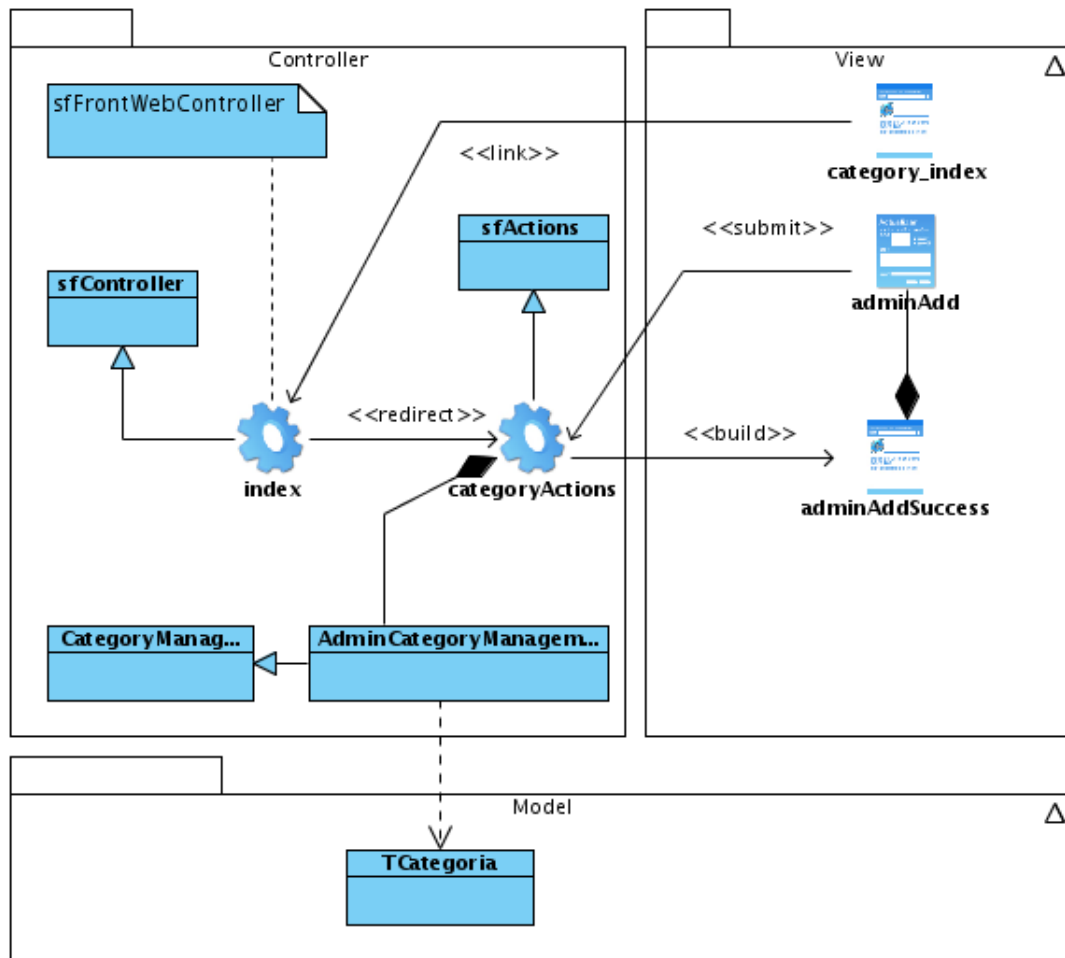


Figura 3.13: Diagrama de Clases del Diseño - Caso de Uso adicionar categoría.

Módulo Dominios de Navegación Básica

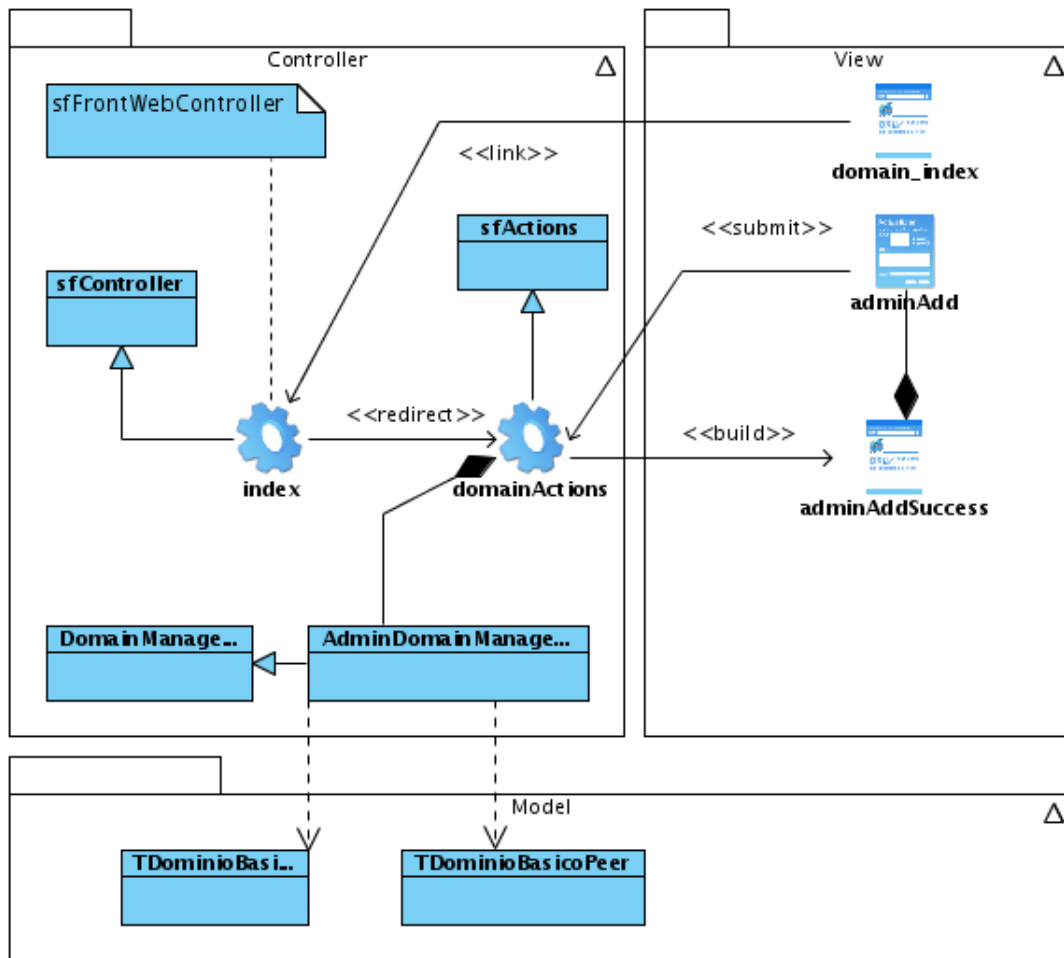


Figura 3.14: Diagrama de Clases del Diseño - Caso de Uso adicionar dominio de navegación básica.

Los restantes diagramas de Clases del Diseño se encuentran en el anexo D.

### 3.2.3. Diagramas de Interacción

#### Módulo Usuario

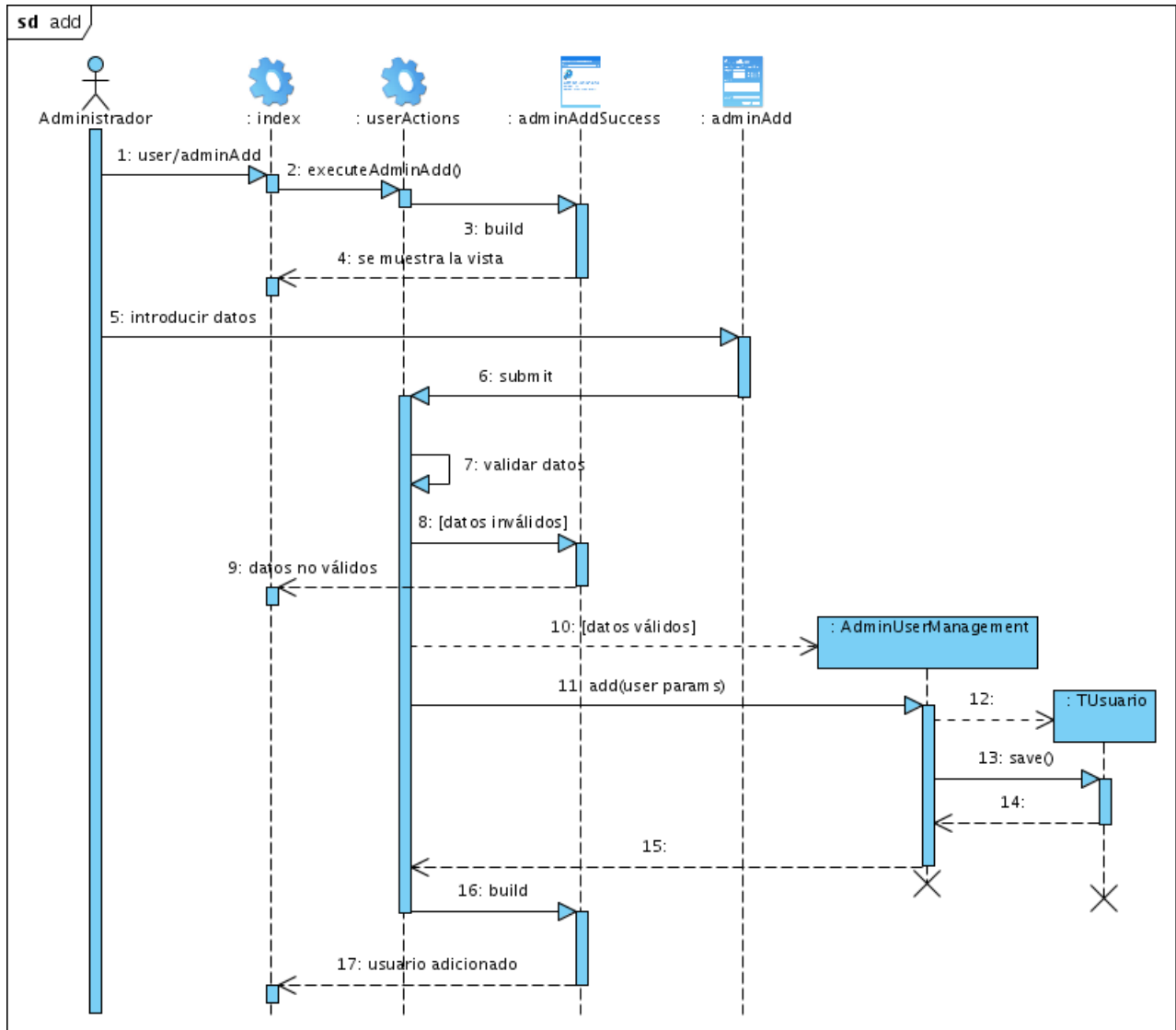


Figura 3.15: Diagrama de Interacción - Caso de Uso adicionar usuario.



Módulo Categoría

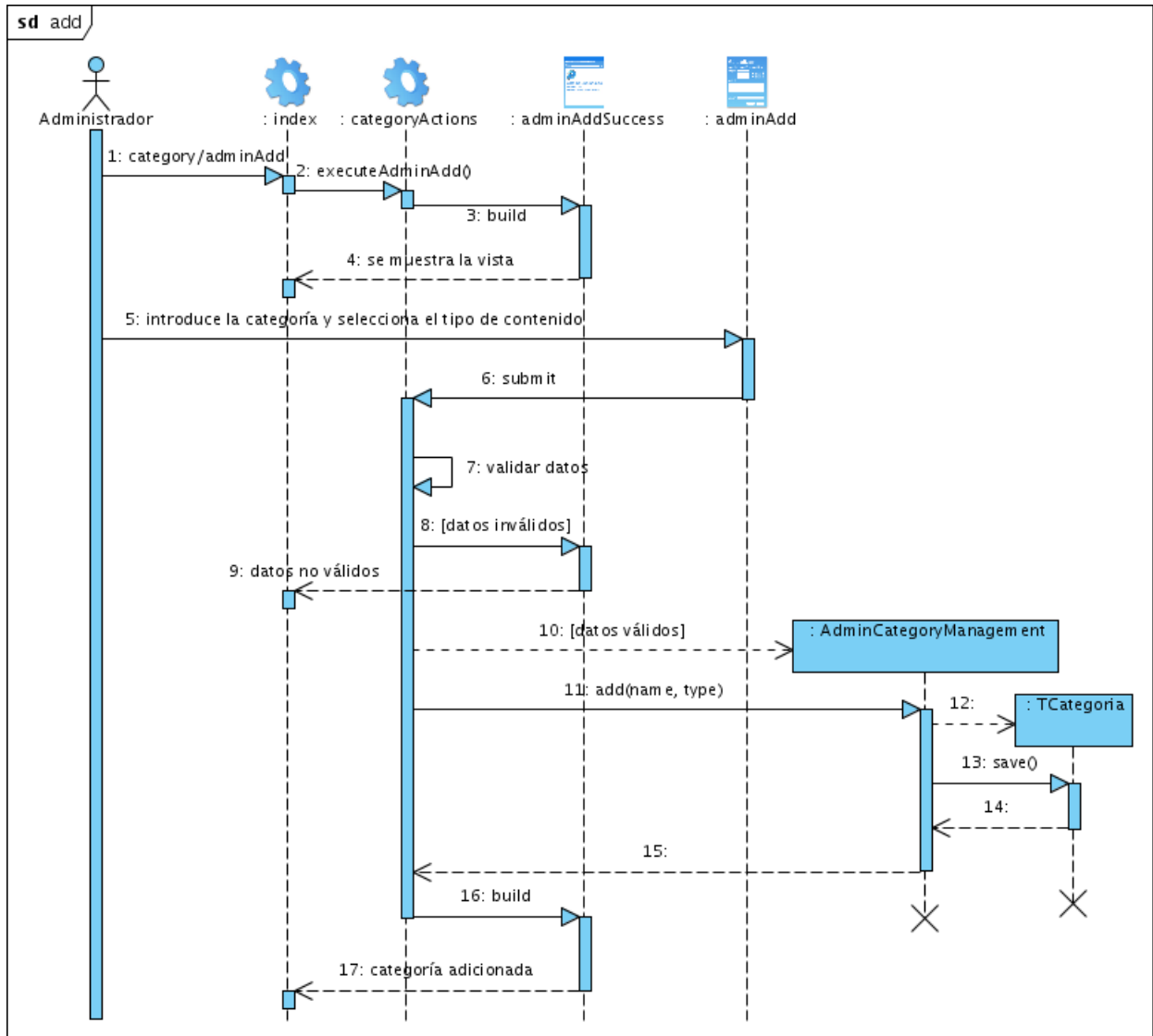


Figura 3.17: Diagrama de Interacción - Caso de Uso adicionar categoría.

Módulo Dominios de Navegación Básica

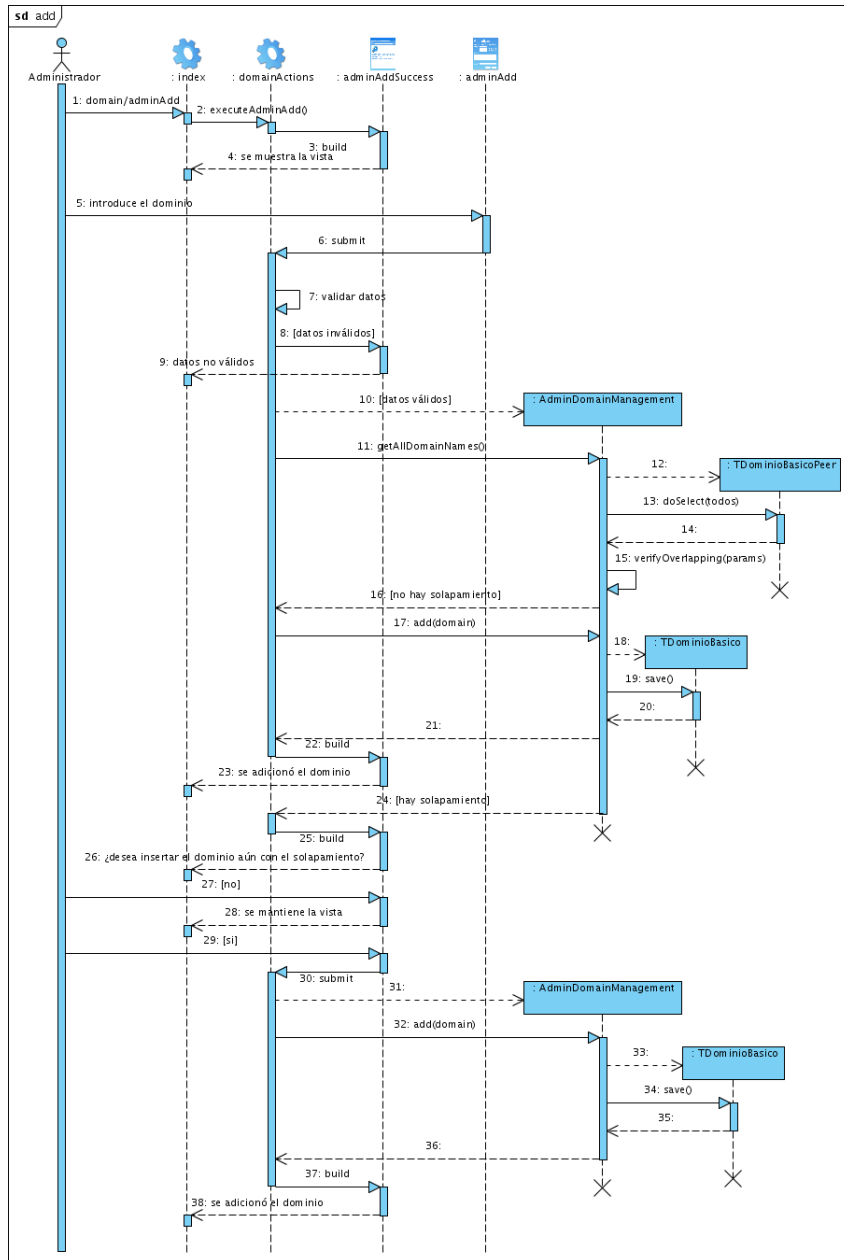


Figura 3.18: Diagrama de Interacción - Caso de Uso adicionar dominio de navegación básica.

Los restantes diagramas de Interacción se encuentran en el anexo E.

### 3.2.4. Descripción de las clases

La descripción de las clases se encuentra en el documento adjunto *diagrama\_clases.pdf*.

### 3.2.5. Diseño de la Base de Datos

Teniendo en cuenta que la base de datos que se utiliza en la IAW es el elemento principal de Filpacon, puede verse como un componente heredado que no se necesita modificar, por lo que sólo se hará una descripción de las tablas que esta posee y sus atributos. A pesar de esto se ha presentado anteriormente el Diagrama Entidad Relación (DER) 2.1 para una mejor comprensión del sistema Filpacon y su BD.

#### Descripción de las tablas

<b>Nombre:</b> t_principal		
<b>Descripción:</b> Almacena todas las URLs que tiene en cuenta el sistema para el proceso de filtrado.		
Atributo	Tipo	Descripción
resumen	char(27)	Almacena las URLs cifradas con el algoritmo sha1_base64.
id_principal	bigint	Campo autoincremental.

<b>Nombre:</b> t_categoria		
<b>Descripción:</b> Contiene las categorías y tipos de contenido (Ilícito, Nocivo o Adecuado) que serán utilizados en el proceso de filtrado. Además permitirá que el sistema sea adaptable al entorno de su instalación.		
Atributo	Tipo	Descripción
nombre_categoria	varchar(64)	Almacena el nombre de todas las categorías que serán analizadas por el sistema.
Continúa en la próxima página		

tipo_categoria	char(1)	Almacena un valor numérico del 0 al 3. 0 = Deshabilitada, 1 = Adecuada, 2 = Nociva y 3 = Ilícita.
id_categoria	integer	Campo autoincremental.

<b>Nombre:</b> t_usuario		
<b>Descripción:</b> Almacena toda la información relacionada con los datos de los usuarios.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
usuario	varchar(128)	Almacenar el identificador del usuario utilizado en el proceso de navegación.
politica_navegación	char(1)	Almacena valores del 0 al 4. 0 = Ninguna, 1= Restrictiva, 2 = Permisiva, 3 = Básica, 4 = Plena.
mayoria_edad	bit	Almacena solo 2 valores true (mayor de edad) o false (no es mayor).
monitorizar	bit	almacena valores true (se dará seguimiento a la navegación del usuario) o false (no se dará seguimiento a la navegación del usuario).
cuota_usada	bigint	Almacena (en bytes) el consumo de navegación del usuario.
cuota_limite	bigint	Almacena (en bytes) el valor máximo permitido en el consumo de navegación del usuario.
incidencias_registradas	int(2)	Almacena la cantidad de incidencias cometidas por el usuario, lo cual permite tomar decisiones con respecto a su navegación.
Continúa en la próxima página		



incidencias_límite	int(2)	Almacena el valor máximo permitido de incidencias para un usuario.
nivel_acceso	int(2)	Almacena el nivel de acceso de los usuarios en la interfaz de administración Web. En caso que sea un administrador es 0 y en caso de un usuario común es 1.
clave_acceso	varchar(128)	Almacena la contraseña de los usuarios cifrada con MD5, será la clave requerida para la navegación tanto para Internet como para IAW.
nombre	varchar(128)	Almacena el nombre del usuario.
apellido	varchar(128)	Almacena los apellidos del usuario.
correo_e	varchar(128)	Almacena el correo electrónico del usuario.
organizacion	varchar(128)	Almacena la organización donde trabaja o radica el usuario.
cargo_area	varchar(128)	Almacena el cargo o el área donde trabaja el usuario.
creado_por	varchar(128)	Almacena quien creó el usuario.
cantidad_restricciones	int(2)	Almacena la cantidad de veces que un usuario ha sido restringido.
id_usuario	integer	Campo autoincremental.

<b>Nombre:</b> t_dominio_basico		
<b>Descripción:</b> Almacena los dominios básicos de navegación que no son más que zonas de confianza que conoce la institución.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Continúa en la próxima página		

dominio_basico	varchar(32)	Almacena todos los dominios básicos definidos por los administradores. Es usado para aquellos usuarios que su tipo de navegación es básica.
id_dominio_básico	integer	Campo autoincremental.

<b>Nombre:</b> t_principal_categoria		
<b>Descripción:</b> Almacena todas las asociaciones entre URLs y Categorías, de ese modo se pueden cambiar las categorías de las URLs, atendiendo el dinamismo de Internet.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_principal	bigint	Llave foránea de la tabla <b>t_principal</b> .
id_categoria	integer	Llave foránea de la tabla <b>t_categoria</b> .

<b>Nombre:</b> t_bitacora		
<b>Descripción:</b> Almacena las URLs que los usuarios han visitado. Solo se hace si está activado el atributo monitorización, dándole seguimiento a su navegación, permitiendo así generar reportes estadísticos a partir de dicha información		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
tiempo_unix	timestamp	Almacena el momento en que un usuario trató de acceder a un sitio con contenido ilícito o nocivo.
ip_origen	varchar(24)	Almacena la dirección ip de origen desde le cual un usuario trató de acceder a un tipo de contenido Ilícito o Nocivo definido por el administrador del sistema.
Continúa en la próxima página		

objeto_visitado	text	Almacena las URLs visitadas por los usuarios en texto plano. Se guardan de este modo para poder hacer estadísticas a partir de las URLs.
id_principal	bigint	Llave foránea de la tabla <b>t_principal</b> .
id_usuario	integer	Llave foránea de la tabla <b>t_usuario</b> .
id_bitacora	bigint	Campo autoincremental.

<b>Nombre:</b> t_rango_restringido		
<b>Descripción:</b> Almacena los rangos de tiempo en los cuales se han denegado la navegación a diferentes usuarios. Esta tabla permite habilitar y deshabilitar la navegación de los usuarios en un determinado período.		
Atributo	Tipo	Descripción
restringido_por	integer	Almacena el identificador del usuario que aplicó la restricción a otro usuario.
restringido_desde	timestamp	Almacena de una restricción cual es la fecha de su comienzo.
restringido_hasta	timestamp	Almacena de una restricción cual es su fecha final.
id_usuario	integer	Llave foránea de la tabla <b>t_usuario</b> .
id_rango_restringido	integer	Campo autoincremental.

<b>Nombre:</b> t_reporte
Continúa en la próxima página

<p><b>Descripción:</b> Almacena posibles inconformidades de los usuarios que naveguen a través del sistema. Constituye un método de retroalimentación para corregir posibles URLs mal clasificadas por parte del sistema.</p>		
Atributo	Tipo	Descripción
tiempo_unix	timestamp	Almacena la fecha que un usuario hizo un reporte de inconformidad con relación a la clasificación de una determinada URL.
usuario	varchar(128)	Almacena que usuario que realizó el reporte de inconformidad.
resumen	char(27)	Almacenar la URL cifrada con el algoritmo sha1_base64.
tipo_categoria	char(10)	Almacena el tipo de contenido que tiene la URL que fue reportada por el usuario.
ip_origen	varchar(24)	Almacena la dirección ip desde la cual el usuario manifestó su inconformidad con respecto a la decisión del sistema.
objeto_visitado	text	Almacena la URL reportada en texto plano.
id_reporte	bigint	Campo autoincremental

### 3.2.6. Interfaz

Para el diseño de la presentación de la IAW es preciso tener conocimientos relacionados con arquitectura de la información y diseño gráfico, por lo cual se consultó a un diseñador de la Infraestructura Productiva (IP) de la UCI para que generara las pautas que han de regir el diseño de la aplicación, estas aparecen en el documento adjunto *pautas\_diseño.pdf*.

## Patrones de diseño gráfico

Los patrones de diseño gráfico contribuyen a la generación de modelos mentales respecto a elementos específicos o incluso a tipos de sitios en general. La importancia de estos patrones radica en la automatización del uso de las interfaces e interacciones, simplificando la realización de tareas y dando al usuario la posibilidad de centrarse en los aspectos más importantes sin necesidad de que resuelva los problemas más básicos para lograr completar correctamente una acción específica. La lógica detrás de dichos patrones permiten identificar y reconocer fácilmente aquellos elementos que se han aprendido y asociado a través de la experiencia y en la medida que este proceso se concreta se eliminan una serie de factores innecesarios asociados a procesos cognitivos, y los usuarios ahorran tiempo y esfuerzo al usar una aplicación [24].

El uso de algunos patrones de diseño gráfico [25, 26] en la IAW para Filpacon permitirán mejorar la experiencia de los usuarios al navegar a través de ella y en la realización de las tareas que brindará la misma. A continuación se muestran los patrones que se utilizarán agrupados según el concepto con que se relacionan.

- Navegación
  - Main Navigation (inverted L menu).
- Búsqueda
  - Search Pagination
- Selección
  - Auto Complete
  - Calendar Picker
- Ayuda o interacción
  - Tool Tip Invitation
- Entrada de datos

- Input Error Message
- Constraint Input

### 3.2.7. Concepción de la ayuda

Un elemento importante de cualquier aplicación es la ayuda que los usuarios necesitan para hacer uso de la misma. En este sentido se proponen varios elementos que junto a la redacción de un manual de usuario (ver documento adjunto *manual\_usuario\_filpacon.pdf*) contribuyan a facilitar el uso de la aplicación, estos son:

- Informar en cada uno de los módulos las acciones que pueden realizarse.
- Utilizar *tooltips* en los formularios para facilitar la entrada de datos.
- Informar a los usuarios el resultado obtenido tras ejecutar cualquier funcionalidad.

## 3.3. Conclusiones

La generación de algunos artefactos relacionados con el flujo de análisis y diseño, teniendo en cuenta la arquitectura MVC que Symfony establece, permitió obtener una mayor comprensión de la aplicación y definir los principios que guiarán la implementación y organización de la misma. Además se analizaron las estrategias a seguir para dar respuesta a los requisitos no funcionales relacionados con la seguridad, se obtuvo un prototipo de interfaz y las pautas de diseño para la misma, así como la concepción de la ayuda de la aplicación.

---

## Capítulo 4

# Implementación y prueba

---

El flujo de implementación y prueba es el más trascendente de la fase de construcción en RUP. Como resultado del mismo, deben traducirse los modelos generados en las fases previas en componentes que formen la aplicación final, reflejando cuales son, su relación y como habrán de distribuirse en los nodos físicos a través de nuevos diagramas. La realización de pruebas también adquiere alta importancia pues permite vislumbrar la existencia de defectos en la implementación realizada.

## 4.1. Implementación

### 4.1.1. Diagrama de Despliegue

El diagrama de Despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Es un grafo de nodos unidos por conexiones de comunicación donde cada nodo puede contener instancias de componentes. En general un nodo puede ser una unidad de computación de algún tipo, desde un sensor hasta un *mainframe* y las instancias de componentes de software pueden estar unidas por relaciones de dependencia [27].

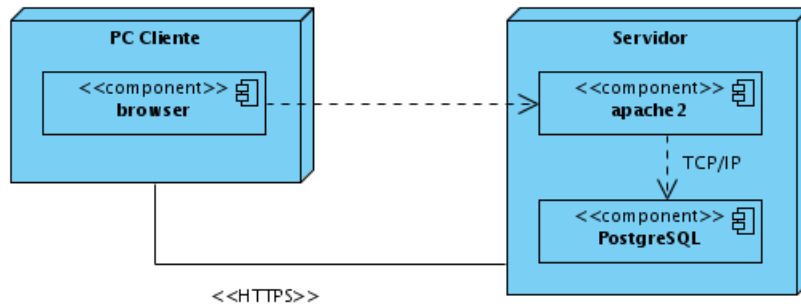


Figura 4.1: Diagrama de Despliegue de la aplicación.

#### 4.1.2. Diagramas de Componentes

El diagrama de Componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean éstos ficheros de código fuente, binarios o ejecutables. Los elementos de modelado que lo conforman son los componentes y paquetes y muestran la estructura del sistema en términos de implementación a un alto nivel. [28].



Módulo Usuario

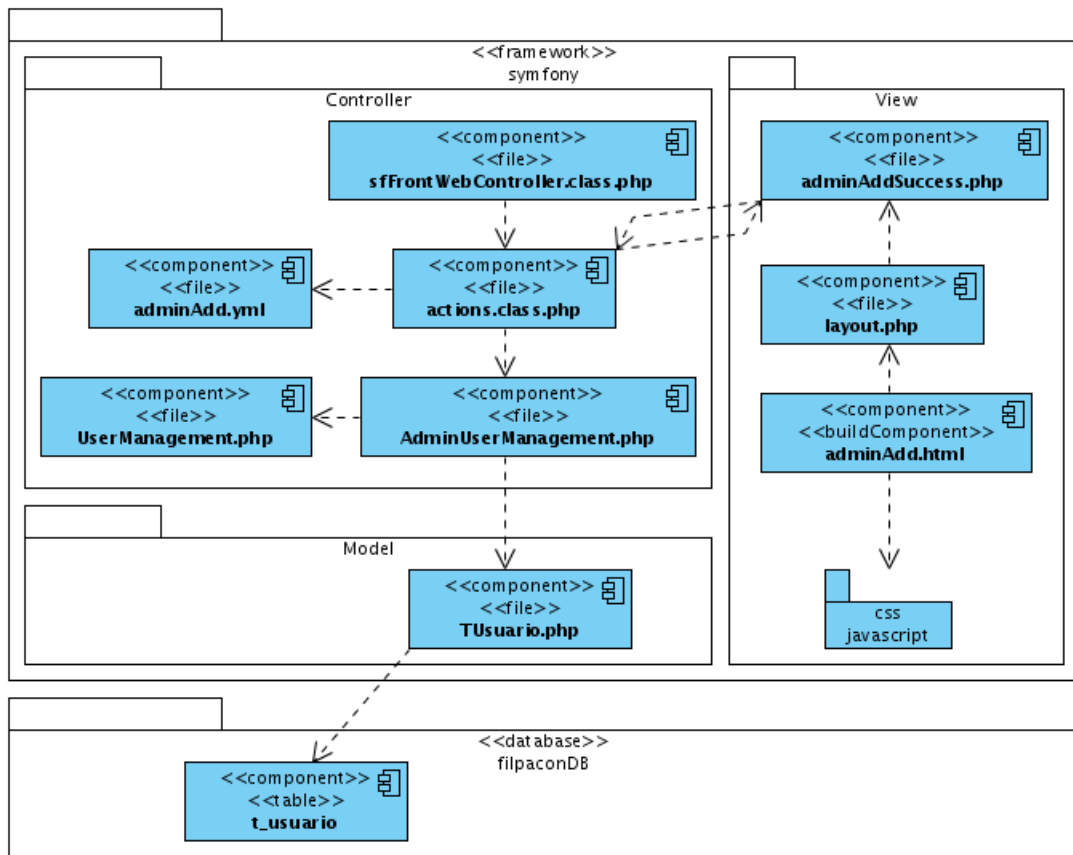


Figura 4.2: Diagrama de Componente - Caso de Uso adicionar usuario.

Módulo URL

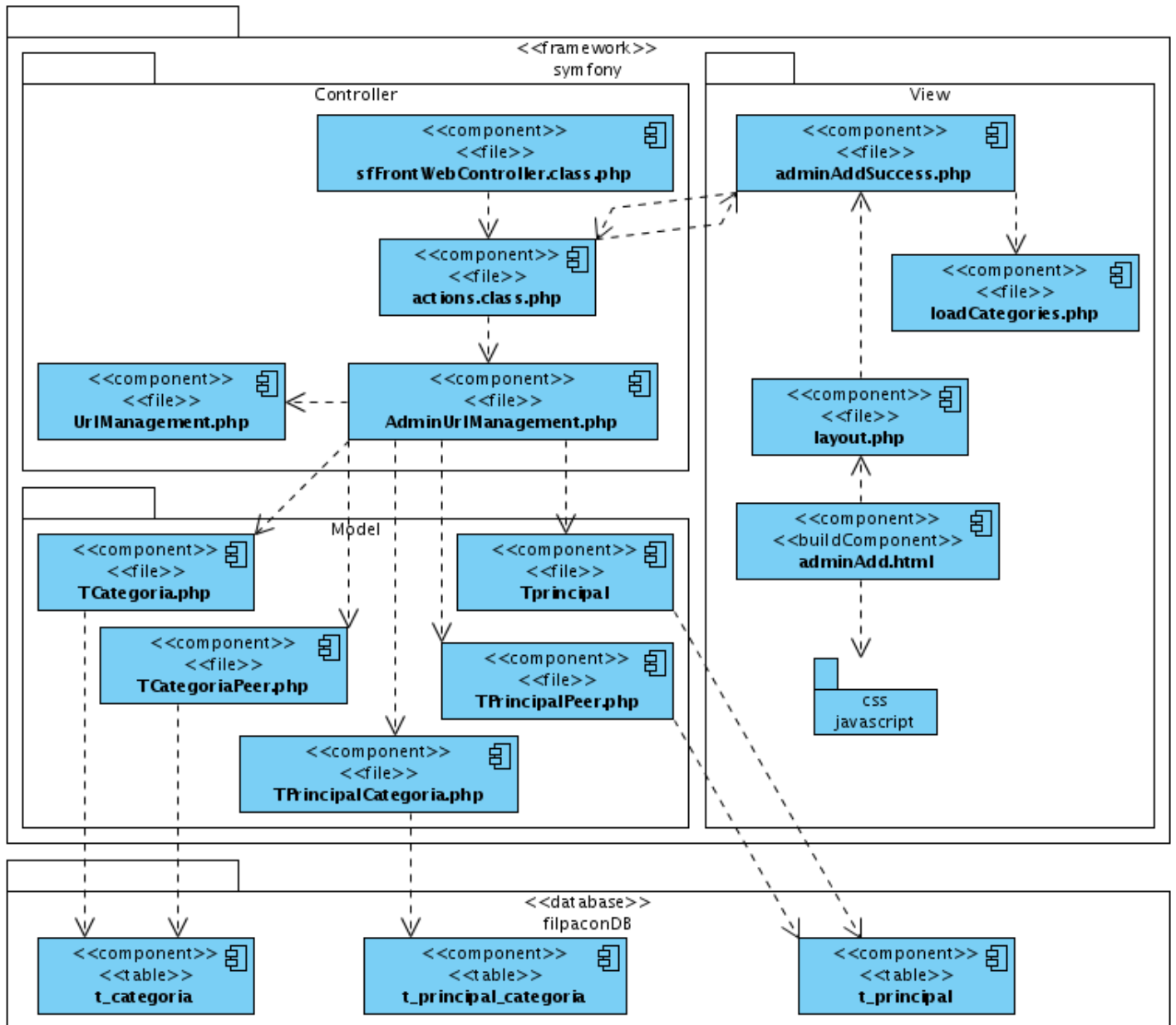


Figura 4.3: Diagrama de Componente - Caso de Uso adicionar una categoría a una URL.

Módulo Categoría

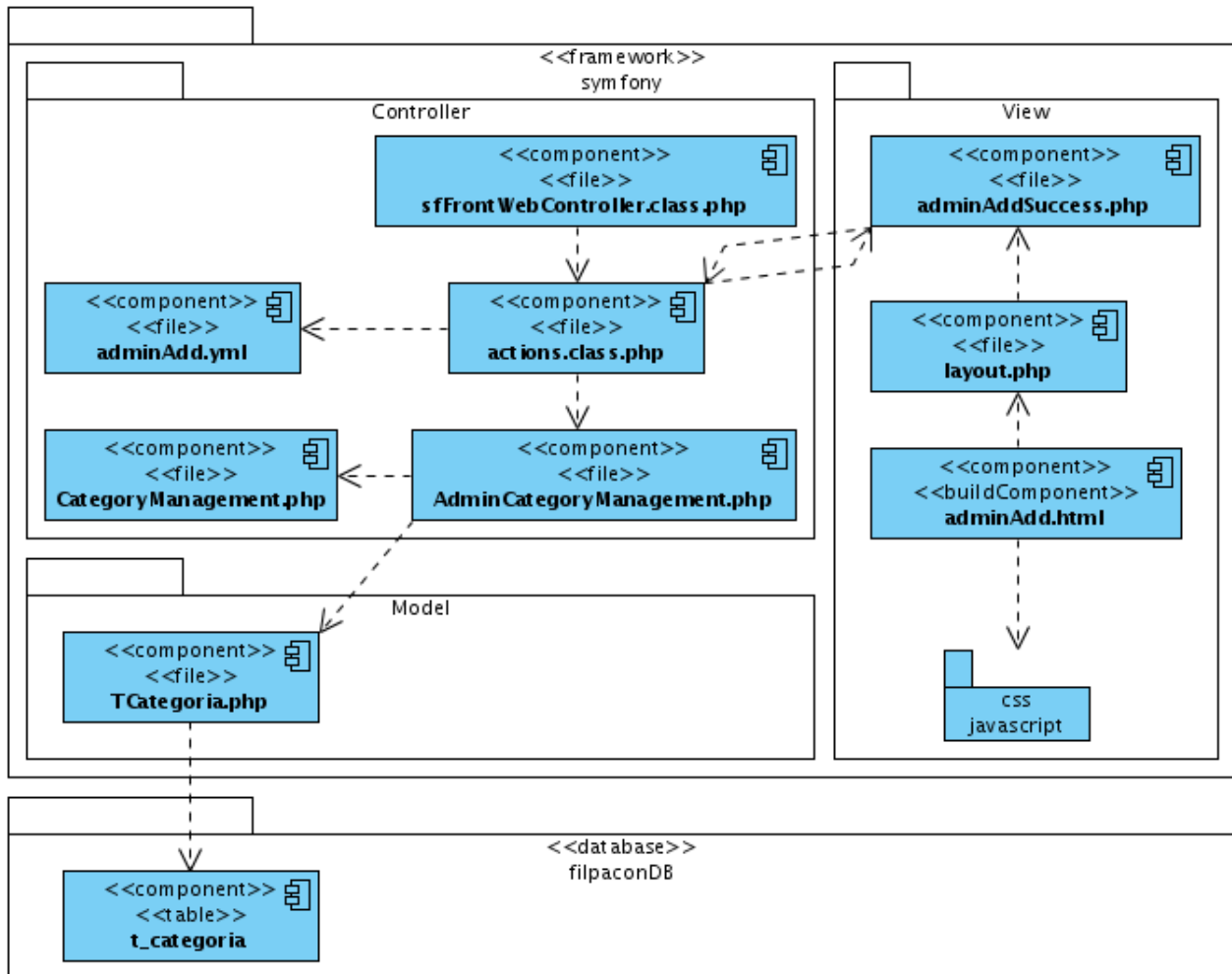


Figura 4.4: Diagrama de Componente - Caso de Uso adicionar una categoría.

**Módulo Dominios de Navegación Básica**

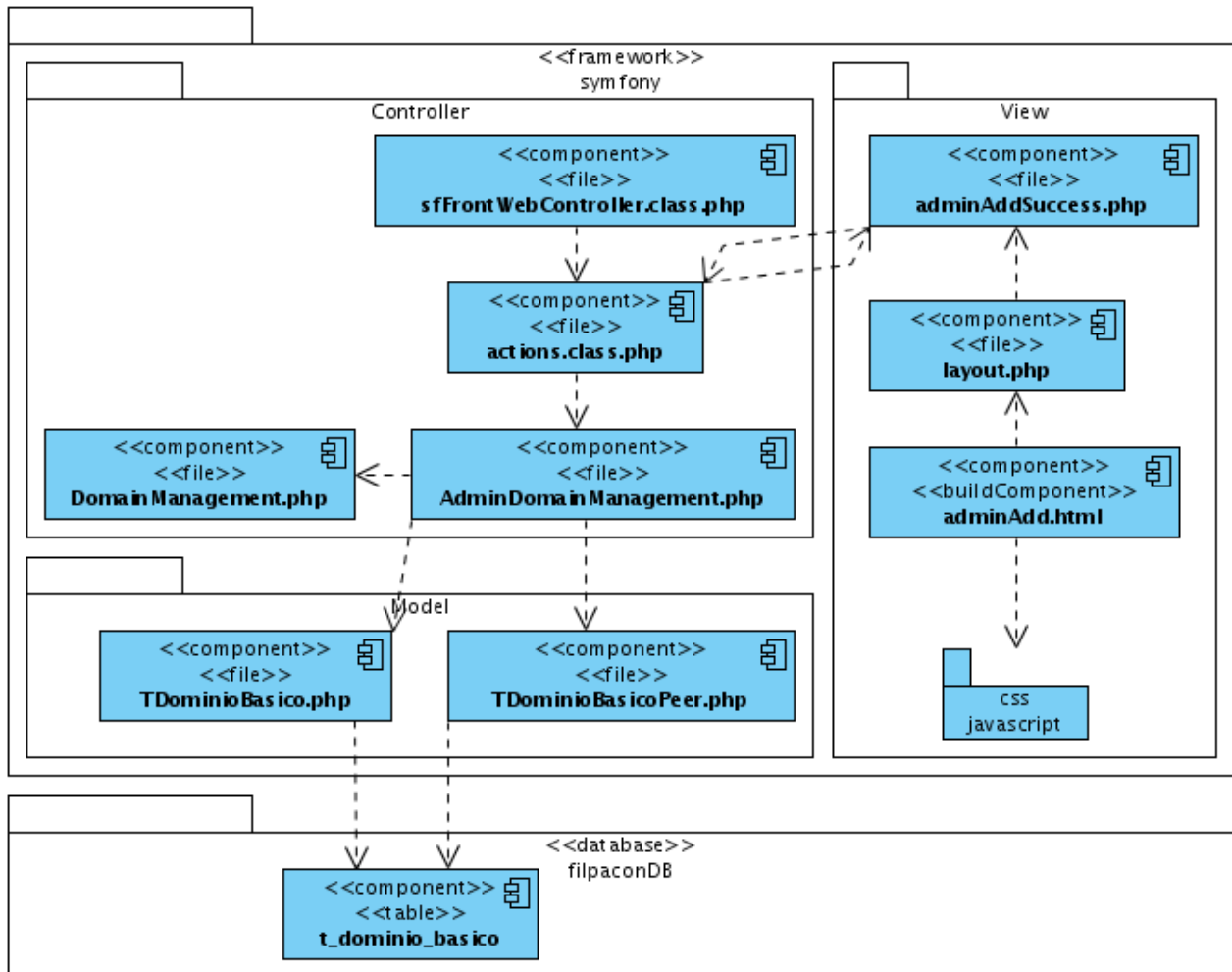


Figura 4.5: Diagrama de Componente - Caso de Uso adicionar dominio de navegación básica.

Los restantes diagramas de Componentes se encuentran en el anexo F.

**4.2. Modelo de prueba**

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún

aspecto del sistema o componente. Las pruebas a la IAW se realizaron por la dirección de calidad de la IP en el marco de las realizadas a Filpacon y la evaluación final puede consultarse en el documento adjunto *pruebas.pdf* 7.

### **4.3. Conclusiones**

Tras el flujo de implementación y prueba el sistema quedó desarrollado. En los diagramas generados pudo ilustrarse la relación entre los principales componentes del sistema y como estará distribuido este. Las pruebas realizadas arrojaron algunos defectos que fueron corregidos y permitieron aumentar la calidad final de la solución. La aplicación final aprovechó las ventajas de Symfony y quedó lista para integrarse con Filpacon.

# Conclusiones

---

Luego de analizar la posibilidad de utilizar alguna solución existente a nivel nacional e internacional para dar respuesta a la problemática planteada, se decidió desarrollar una IAW para el sistema Filpacon. Al finalizar el desarrollo de la misma se pueden plantear las siguientes conclusiones:

- El estudio del estado del arte relacionado con las GUI y las aplicaciones Web permitió contar con los elementos teóricos necesarios y elegir las herramientas, tecnologías y la metodología para guiar el proceso de desarrollo.
- El levantamiento de los requisitos funcionales y no funcionales y la descripción de los casos de uso derivados de estos, sirvieron como guía para los flujos y fases posteriores.
- Los artefactos generados en el flujo de análisis y diseño teniendo en cuenta las características y arquitectura del framework symfony permitieron implementar de la aplicación de forma correcta.
- El uso de symfony permitió centrarse solamente en la lógica de negocio de la aplicación, ahorrando tiempo y esfuerzos que hubiesen tenido que dedicarse a las tareas comunes que hay que realizar cuando se desarrolla una aplicación Web. Además la realización de las pruebas y la corrección de los defectos detectados fue mucho más simple.
- El diseño gráfico de la interfaz, el patrón de navegación utilizado y los elementos de usabilidad incorporados junto a la funcionalidad y eficiencia de la IAW han hecho más sencilla la interacción de los administradores y usuarios con el sistema Filpacon.

Ante los elementos anteriores puede afirmarse que se cumplieron los objetivos propuestos y se verificó la validez de la idea a defender, lo cual se materializa en la aplicación obtenida. Actualmente está implantada en dos Infocentros<sup>1</sup> de la República Bolivariana de Venezuela y en la Facultad 10 de la UCI como parte de Filpacon.

---

<sup>1</sup>Equivalentes a los Joven Club de Computación y Electrónica en Cuba

# Recomendaciones

---

Es importante tener en cuenta que por naturaleza la mayoría de las aplicaciones necesitan evolucionar, ya sea por el surgimiento de nuevos requisitos o por la necesidad de mejorar las características existentes. Ante esta situación, a la que la IAW no es ajena, se proponen las siguientes recomendaciones para futuras versiones de la misma o para incorporarlas en la versión existente.

- Como un elemento relacionado con la seguridad de la aplicación y en especial con la auditoría de las actividades de los usuarios que posean rol de administración deben registrarse todas las actividades realizadas por estos.
- Utilizar el plugin de symfony *sfGuardPlugin* para hacer más flexible el proceso de autenticación y autorización de los usuarios a través de la gestión de roles y grupos.
- Valorar el uso de otras metodologías de desarrollo o incorporar algunos elementos de estas a RUP de modo que el proceso de desarrollo sea más ágil y acorde con la naturaleza cambiante de las aplicaciones Web.
- Automatizar la realización de las pruebas mediante el framework interno de pruebas *Lime* que symfony posee para este propósito.
- Utilizar los mecanismos de internacionalización de symfony para aumentar las posibilidades de uso de la IAW por parte de los usuarios no hispanohablantes y con ello las de comercialización del sistema Filpacon.



Además de estas recomendaciones relacionadas directamente con el objetivo de este trabajo, se propone que las aplicaciones de gestión complejas que hallan de desarrollarse en una plataforma similar a la utilizada usen symfony como framework de desarrollo.

# Referencia bibliográfica

---

- [1] Dulce Carolina Córdova Cruz (2005). *Origen del Diseño de la Interfaz Gráfica* [Internet]. Disponible en: <[http://www.mati.unam.mx/index.php?option=com\\_content&task=view&id=92&Itemid=35](http://www.mati.unam.mx/index.php?option=com_content&task=view&id=92&Itemid=35)> [Citado 15 Mayo 2008].
- [2] *Componentes de una interfaz web. Interfaces* [Internet]. Disponible en: <<http://www.desarrolloweb.com/articulos/2171.php>> [Citado 15 Mayo 2008].
- [3] Cuauhtémoc Rivera Loiza (2000). *Interfaz de usuario* [Internet]. Disponible en: <<http://www.fismat.umich.mx/~crivera/tesis/node6.html>> [Citado 15 Mayo 2008].
- [4] Gerti Kappel et al. *Web Engineering: The Discipline of Systematic Development of Web Applications*, chapter An Introduction to Web Engineering, pages 1–22. John Wiley & Sons Ltd., 2006.
- [5] Bob Baxley (2003). *What is a Web Application?* [Internet]. Disponible en: <[http://www.boxesandarrows.com/view/what\\_is\\_a\\_web\\_application\\_](http://www.boxesandarrows.com/view/what_is_a_web_application_)> [Citado 13 Mayo 2008].
- [6] Daniel Lewis. *Evolución de las interfaces Web* [Internet]. Disponible en: <<http://www.acm.org/crossroads/espanol/xrds13-1/web20.html>> [Citado 15 Mayo 2008].
- [7] Sergio Luján Mora. *Programación de aplicaciones Web: historia, principios básicos y clientes Web*, pages III–IV. Editorial Club Universitario, 2002.
- [8] Sergio Luján Mora. *Programación de aplicaciones Web: historia, principios básicos y clientes Web*, pages 25–37. Editorial Club Universitario, 2002.

- 
- [9] Christian Van Der Henst S. (2005). *¿Qué es la Web 2.0?* [Internet]. Disponible en: <<http://www.maestrosdelweb.com/editorial/web2/>> [Citado 12 Mayo 2008].
- [10] Roger S. Pressman. *Ingeniería del software. Un enfoque práctico*, chapter Ingeniería Web, pages 521–540. McGraw-Hill, 2002.
- [11] San Murugesan y Athula Ginige. *Web Engineering*, chapter Web Engineering: Introduction and Perspectives, pages 1–30. Springer Berlin / Heidelberg, 2001.
- [12] Andrew van der Stock et al. *What are web applications?* [Internet]. Disponible en: <[http://www.owasp.org/index.php/What\\_are\\_web\\_applications%3F](http://www.owasp.org/index.php/What_are_web_applications%3F)> [Citado 13 Mayo 2008].
- [13] *Software Architecture for Software-Intensive Systems* [Internet]. Disponible en: <[http://www.sei.cmu.edu/architecture/published\\_definitions.html](http://www.sei.cmu.edu/architecture/published_definitions.html)> [Citado 16 Mayo 2008].
- [14] Agustín Dondo. *¿Por qué elegir PHP?* [Internet]. Disponible en: <<http://www.programacion.com/php/articulo/porquephp/>> [Citado 13 Mayo 2008].
- [15] *PHP* [Internet]. Disponible en: <<http://es.wikipedia.org/wiki/.php#Desventajas>> [Citado 22 Mayo 2008].
- [16] Erich Gamma et al (1994). *Design Patterns: Elements of Reusable Object-Oriented Software* [Internet]. Disponible en: <<http://www.exciton.cs.rice.edu/JavaResources/DesignPatterns/book/hires/chap1fso.htm>> [Citado 22 Mayo 2008].
- [17] Dennis Pallett. *Taking a look at ten different PHP frameworks* [Internet]. Disponible en: <<http://www.phpit.net/article/ten-different-php-frameworks>> [Citado 16 Mayo 2008].
- [18] *PDT Project* [Internet]. Disponible en: <<http://www.eclipse.org/pdt/>> [Citado 20 Mayo 2008].
- [19] Gregor Engels et al. *Web Engineering: The Discipline of Systematic Development of Web Applications*, chapter The Web Application Development Process, pages 197–218. John Wiley & Sons Ltd., 2006.
- [20] Efraín Villanueva y Diego Fernando Molina. *Aporte a Rup para el Desarrollo de Aplicaciones Web Transaccionales-Requerimientos no Funcionales y Administración de Riesgos* [Internet]. Disponible en: <<http://www.javeriana.edu.co/biblos/tesis/ingenieria/Tesis205.pdf>> [Citado 18 Mayo 2008].

- [21] Ivar Jacobson et al. *El Proceso Unificado de Desarrollo de Software*, chapter El Proceso Unificado: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental, pages 1–12. Addison Wesley, 2000.
- [22] *Modelo de Dominio* [Internet]. Disponible en: <[http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos\\_clase2.pdf](http://iie.fing.edu.uy/ense/assign/desasoft/practico/hoja8/ejemplos_clase2.pdf)> [Citado 20 Mayo 2008].
- [23] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2002.
- [24] Nelson Rodríguez Peña (2004). *Patrones y Diseño Web* [Internet]. Disponible en: <<http://www.webstudio.cl/blog/patrones-y-diseno-web/>> [Citado 16 junio 2008].
- [25] *Design Pattern Library* [Internet]. Disponible en: <<http://developer.yahoo.com/ypatterns/>> [Citado 16 junio 2008].
- [26] *Pattern library* [Internet]. Disponible en: <<http://www.welie.com/patterns/>> [Citado 16 junio 2008].
- [27] Ana Fernandez Vilas (2001). *Diagrama de Despliegue* [Internet]. Disponible en: <<http://www-gris.det.uvigo.es/~avilas/UML/node50.html>> [Citado 4 junio 2008].
- [28] Ana Fernandez Vilas (2001). *Diagrama de Componentes* [Internet]. Disponible en: <<http://www-gris.det.uvigo.es/~avilas/UML/node49.html>> [Citado 4 junio 2008].

# Glosario de términos

---

- UCI** Son las siglas de Universidad de las Ciencias Informáticas la cual fue creada, por el comandante en jefe de la Revolución cubana, al calor de la batalla de ideas; en el año 2002.
- Filpacon** Filtrado de Paquetes por Contenidos
- OSRI** Oficina de Seguridad para las Redes Informáticas
- GUI** Interfaz Gráfica de Usuario
- GUIs** Interfaces Gráficas de Usuario
- UI** Interfaz de Usuario
- IAW** Interfaz de Administración Web
- IDE** Entorno de Desarrollo Integrado
- RUP** Rational Unified Process
- IP** Infraestructura Productiva
- CGI** *Common Gateway Interface*. Estándar que permite el intercambio de información entre el servidor y un programa externo al servidor. Un programa CGI es un programa preparado para recibir y enviar datos desde y hacia un servidor Web según este estándar. Normalmente se programan en C o en Perl, aunque se puede usar cualquier lenguaje de propósito general.

- 
- JSP** *Java Server Page.* Tecnología de *Sun Microsystem* que permite crear páginas web dinámicas en el servidor. Equivale a la tecnología ASP de *Microsoft*. Se programan en *Java*.
- ASP** *Active Server Pages.* Tecnología propietaria de MICROSOFT que permite crear páginas Web dinámicas en el Servidor. Desarrollada con el objetivo de sustituir la tecnología CGI, ofrece una serie de características que facilitan la programación de aplicaciones Web.
- IDC** *Internet Database Connector.* Conector de bases de datos de Internet. Tecnología propietaria de MICROSOFT que permite generar páginas Web dinámicas a partir de la información almacenada en la base de datos. Es el precursor de ASP.
- CSS** *Cascading Style Sheets.* Tecnología empleada en la creación de páginas Web, que permite un mayor control sobre el lenguaje HTML. Permite hojas de estilo que definen como cada elemento, se tiene que mostrar. CSS ha sido desarrollado por W3C.
- HTML** *HyperText Markup Language.* Lenguaje compuesto de una serie de etiquetas o marcas que permiten definir el contenido y la apariencia de las páginas Web. Aunque se basa en el estándar SGML, no se puede considerar que sea un subconjunto de él. Existen cientos de etiquetas con diferentes atributos. W3C se encarga de su estandarización. El futuro sustituto de HTML es XHTML.
- DHTML** *Dynamic HTML.* Conjunto de extensiones a HTML que permiten modificar el contenido de una página Web en el cliente sin necesidad de establecer una conexión con el servidor. Se basa en el uso de DOM para acceder al contenido de la página.
- DOM** *Document Object Model.* Especificación que define como se puede acceder a los objetos de un documento HTML (ventanas, imágenes y formularios) a través de un lenguaje script. Básicamente define una jerarquía de objetos.
- SGML** *Standard Generalized Markup Language.* Lenguaje que permite organizar y etiquetar los distintos elementos que componen un documento. Se emplea para manejar grandes documentos que sufren constantes revisiones y se imprimen en distintos formatos de idioma.

- 
- XHTML** *Extensible HyperText Markup Language*. HTML escrito según las normas que marca XML. Por tanto, se trata de una aplicación concreta de XML y no tienen que confundirse entre sí.
- XML** *Extensible Markup Language*. Metalenguaje de etiquetado en SGML. Diseñado específicamente para la WWW por W3C. Permite que un usuario diseñe sus propias etiquetas, con sus atributos y reglas de construcción de documentos.
- WWW** *World Wide Web*. También conocida como “la Web” ó “la Red”. Sistema mundial de servidores Web conectados a Internet (No todos los ordenadores conectados a Internet forman parte de la WWW).
- W3C** *World Wide Web Consortium*. Consorcio internacional de compañías y organizaciones involucradas en el desarrollo de Internet y en especial de la WWW. Su propósito es desarrollar estándares y “poner orden” en Internet.
- HTTP** *HyperText Transfer Protocol*. Es el protocolo que emplea la WWW. Define como se tienen que crear y enviar los mensajes y que opciones debe tener el servidor y el navegador en respuesta a un comando.

## Anexo A

# Figuras relacionadas

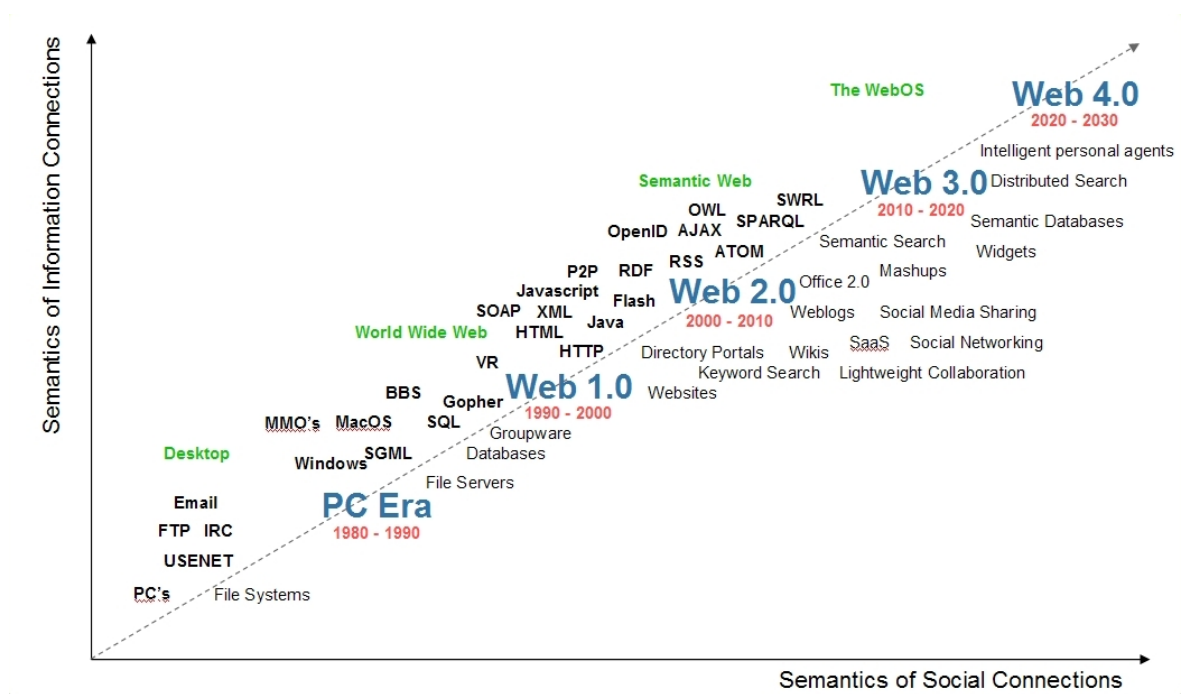


Figura A.1: Evolución de la Web.





Figura A.2: Conceptos alrededor de la Web 2.0.

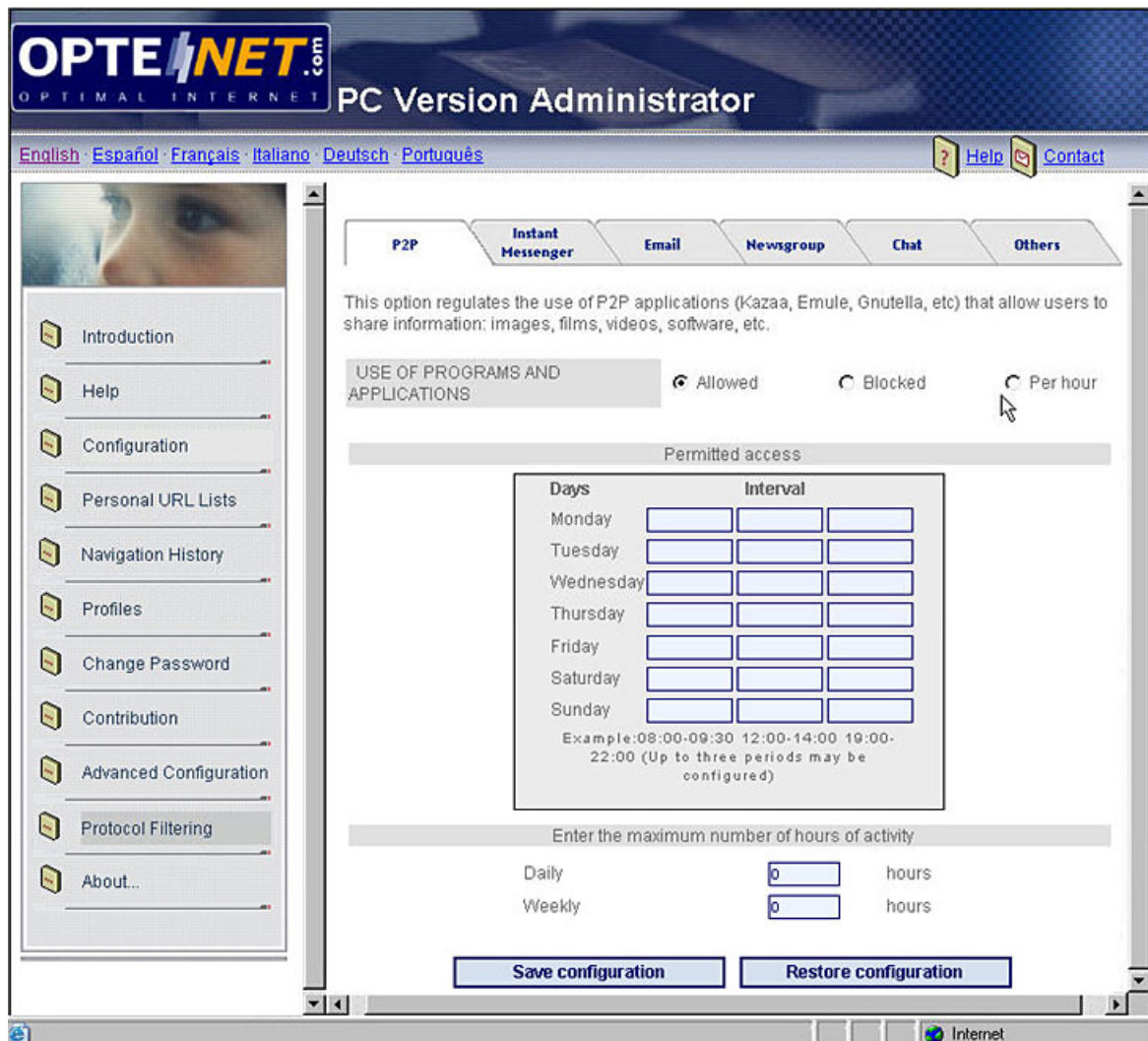


Figura A.3: Interfaz de Administración de Optenet.

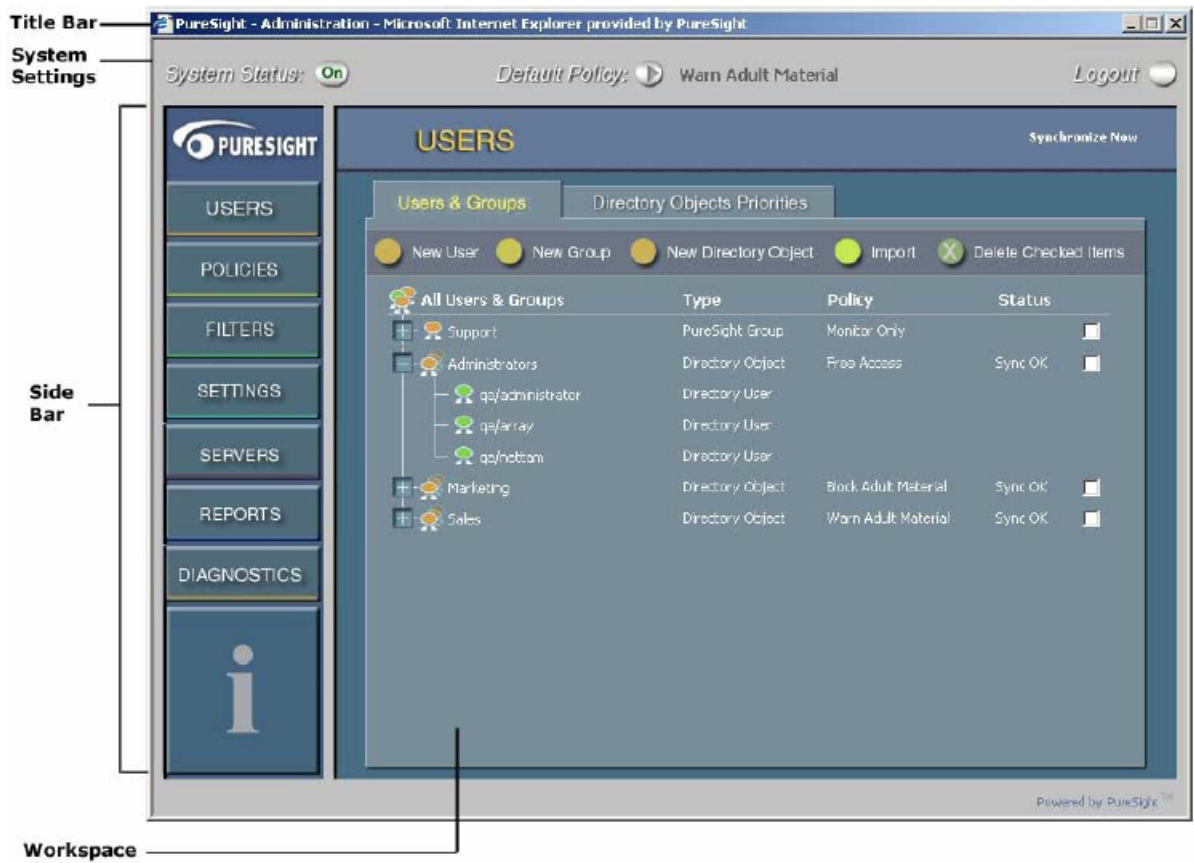


Figura A.4: Interfaz de Administración de PureSight.

---

## Anexo B

# Diagramas de Casos de Uso expandido

---

### Caso de Uso: Iniciar sesión de usuario

Caso de Uso:	Iniciar sesión de usuario
Actores:	Administrador, Usuario
Resumen:	El Caso de Uso se inicia cuando el actor introduce los datos para acceder a la aplicación, estos se verifican y finaliza dándole los permisos que tiene y habilitándole o denegándole la entrada al sistema.
Referencia:	R1
CU asociados:	
Precondiciones:	Que el actor previamente se la haya creado una cuenta en el Sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<p>1. El actor entra su identificador y la contraseña.</p>	<p>1.1 El sistema chequea si existe el identificador en la Base de Datos.</p> <p>1.2 El sistema encripta la contraseña.</p> <p>1.3 El sistema verifica los permisos del Usuario.</p> <p>1.4 El sistema crea la credencial (<i>token</i>).</p> <p>1.5 El sistema asigna los niveles de acceso al Usuario.</p> <p>1.6 El sistema carga el módulo en dependencia de los niveles de acceso.</p>
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>1.1 Si no existe el identificador del actor, volver al paso 1.</p>
Poscondiciones:	La sesión finalmente es iniciada, con todos los permisos que le corresponden.
Prioridad:	Crítico
Especificaciones:	
Complementaria:	

**Caso de Uso: Finalizar sesión de usuario**

Caso de Uso:	Finalizar sesión de usuario
Actores:	Administrador, Usuario
Resumen:	El Caso de Uso se inicia cuando el actor decide salir del sistema.
Referencia:	R2
CU asociados:	
Precondiciones:	Que el actor este autenticado en el sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El actor solicita la opción de terminar sesión.</li> <li>2. El actor confirma la solicitud.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema pide confirmación por la opción solicitada.</li> <li>2.1 El sistema destruye la sesión del actor.</li> <li>2.2 El sistema retorna a la página principal.</li> </ol>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Poscondiciones:	La sesión finalmente es terminada, y se sale del sistema.
Prioridad:	Crítico
Especificaciones:	
Complementaria:	

**Caso de Uso: Cambiar contraseña**

Caso de Uso:	Cambiar contraseña
Actores:	Administrador, Usuario
Resumen:	El Caso de Uso se inicia cuando el actor decide cambiar su contraseña.
Referencia:	R7
CU asociados:	
Precondiciones:	Que el actor se haya autenticado en el sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El actor solicita cambiar la contraseña.	<p>1.1 Si el actor que está autenticado es un Administrador, ir a la sección “<b>Cambiar contraseña por Administrador</b>”.</p> <p>1.2 Si el actor que está autenticado es un Usuario, ir a la sección “<b>Cambiar contraseña por Usuario</b>”.</p>
<b>Sección “Cambiar contraseña por Administrador”</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<ol style="list-style-type: none"> <li>1. El Administrador escoge la opción de cambiar la contraseña de un Usuario y no la opción de cambiar su propia contraseña.</li> <li>2. El Administrador entra el identificador del Usuario.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema le muestra un formulario para para que indique el Usuario al que le desea cambiar la contraseña.</li> <li>2.1 El sistema busca el Usuario en la Base de Datos.</li> <li>2.2 El sistema le muestra el mismo formulario con los campos para la contraseña y la confirmación de esta.</li> <li>3.1 El sistema valida la información.</li> <li>3.2 El sistema actualiza la nueva contraseña del Usuario en la Base de Datos.</li> </ol>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	



<ol style="list-style-type: none"> <li>1. El Administrador escoge la opción de cambiar su propia contraseña.</li> <li>2. El Administrador indica los datos para realizar el cambio de la contraseña.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema le muestra un formulario para que entre la contraseña actual, la nueva y confirme esta última.</li> <li>2.1 Si el Usuario no se encuentra en la Base de Datos, se envía un mensaje de advertencia y retorna al paso 2.</li> <li>2.1 (responde al FA 2.) El sistema cambia la contraseña del Administrador.</li> <li>3.1 Si la contraseña entrada tiene errores, se envía un mensaje de advertencia y regresa al paso 3.</li> </ol>
---	--

**Sección “Cambiar contraseña por Usuario”**

**Flujo Normal de Eventos**

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>2. El Usuario inserta y confirma su nueva contraseña.</li> </ol>	<ol style="list-style-type: none"> <li>2.1 El sistema pide que inserte y confirme la nueva contraseña.</li> <li>2.1 El sistema valida la información.</li> <li>2.2 El sistema actualiza la contraseña del Usuario en la Base de Datos.</li> </ol>

Continúa en la próxima página

Flujo alternativo	
Acción del Actor	Respuesta del Sistema
	2.1 El sistema envía un mensaje en caso de tener problema la contraseña entrada por el Usuario y retorna al paso 2.
Poscondiciones:	Los contraseña del actor finalmente es modificada y guardada en la Base de Datos del sistema.
Prioridad:	Secundario
Especificaciones:	
Complementaria:	

### Caso de Uso: Habilitar navegación

Caso de Uso:	Habilitar navegación
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador decide habilitar la navegación de un Usuario que previamente le fue restringida.
Referencia:	R8
CU asociados:	
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.
Flujo Normal de Eventos	
Sección "nombre del escenario"	
Acción del Actor	Respuesta del Sistema
	Continúa en la próxima página

<ol style="list-style-type: none"> <li>1. El Administrador solicita la opción de habilitar navegación.</li> <li>2. El Administrador entra el identificador del Usuario.</li> <li>3. El Administrador selecciona la restricción que contiene la fecha actual e indica que esta sea eliminada (puede eliminar una, varias o todas).</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema pide el identificador del Usuario.</li> <li>2.1 El sistema busca el Usuario dado su identificador.</li> <li>2.2 El sistema muestra todas las restricciones que impiden la navegación del Usuario.</li> <li>2.3 El sistema diferencia de esas restricciones la que contiene la fecha actual.</li> <li>3.1 El sistema elimina las restricciones seleccionadas por el Administrador.</li> <li>3.2 El sistema restablece la navegación del Usuario seleccionado.</li> </ol>
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<ol style="list-style-type: none"> <li>2.1 Si el identificador no es encontrado en la Base de Datos se muestra un mensaje de advertencia y regresa al paso 2.</li> </ol>
Poscondiciones:	La navegación del usuario finalmente es habilitada.
Prioridad:	Secundario
Especificaciones:	
Continúa en la próxima página	

Complementaria:	
-----------------	--

**Caso de Uso: Denegar navegación**

Caso de Uso:	Denegar navegación	
Actores:	Administrador	
Resumen:	El Caso de Uso se inicia cuando el Administrador decide restringir la navegación de un Usuario.	
Referencia:	R9	
CU asociados:		
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “nombre del escenario”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
Continúa en la próxima página		

<ol style="list-style-type: none"> <li>1. El Administrador solicita la opción de denegar navegación.</li> <li>2. El Administrador entra el identificador del Usuario.</li> <li>3. El Administrador entra el rango de fecha de la sanción.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema pide el identificador del Usuario.</li> <li>2.1 El sistema busca el Usuario a modificar.</li> <li>2.2 El sistema permite que seleccione el rango de fecha para la cual la navegación será denegada.</li> <li>3.1 El sistema valida el rango de fecha entrado por el Administrador.</li> <li>3.2 El sistema actualiza el estado de la navegación en la Base de Datos.</li> </ol>
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<ol style="list-style-type: none"> <li>2.1 Si el identificador no es encontrado en la Base de Datos se muestra un mensaje de advertencia y regresa al paso 2.</li> <li>3.1 Si la fecha no se encuentra en el formato correcto, se envía un mensaje de advertencia y regresa al paso 3.</li> </ol>
Poscondiciones:	La navegación del usuario finalmente denegada.
Prioridad:	Secundario
Continúa en la próxima página	

Especificaciones:	
Complementaria:	

### Caso de Uso: Comprobar categoría de URL

Caso de Uso:	Comprobar categoría de URL	
Actores:	Administrador	
Resumen:	El Caso de Uso se inicia cuando el Administrador decide determinar si una URL se encuentra categorizada en la Base de Datos y a qué categoría pertenece.	
Referencia:	R12	
CU asociados:		
Precondiciones:	Que el actor se haya autenticado como Administrador del sistema.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “nombre del escenario”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<ol style="list-style-type: none"> <li>1. El Administrador solicita la opción de comprobar la categoría de una URL.</li> <li>2. El Administrador entra la URL.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema pide la URL que desea verificar.</li> <li>2.1 El sistema busca la URL.</li> <li>2.2 El sistema muestra la(s) categoría(s) asociada(s) a la URL.</li> </ol>	
<b>Flujo alternativo</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
Continúa en la próxima página		

	2.1 Si la URL no se encuentra en la Base de Datos, se envía un mensaje de advertencia y regresa al paso 2.
Poscondiciones:	La categoría de URL finalmente es comprobada.
Prioridad:	Secundario
Especificaciones:	
Complementaria:	

### Caso de Uso: Modificar categoría

Caso de Uso:	Modificar categoría
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema para modificar el contenido correspondiente a una categoría.
Referencia:	R16
CU asociados:	
Precondiciones:	Que el usuario se haya autenticado como Administrador del sistema.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<ol style="list-style-type: none"> <li>1. El Administrador entra al sistema para modificar el contenido de una categoría.</li> <li>2. El Administrador selecciona la categoría a modificar.</li> <li>3. El Administrador realiza las modificaciones en los datos de la categoría.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema le muestra las categorías existentes en la Base de Datos.</li> <li>2.1 El sistema muestra el tipo de contenido al que pertenece la categoría a modificar (Ilícito, No-civo, Adecuado o Deshabilitada).</li> <li>3.1 El sistema modifica la categoría en la Base de Datos.</li> </ol>
<b>Flujo alternativo</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Poscondiciones:	La categoría es modificada en la Base de Datos del Sistema.
Prioridad:	Crítico
Especificaciones:	
Complementaria:	

### Caso de Uso: Mostrar las incidencias por usuarios

Caso de Uso:	Mostrar las incidencias por usuarios
Actores:	Administrador, usuario
Resumen:	El Caso de Uso se inicia cuando los actores desean obtener el reporte de incidencias por usuarios.
Referencia:	R19
CU asociados:	Gestionar reporte (extensión).
Continúa en la próxima página	



Precondiciones:	Que los actores se hayan autenticado.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
2. El Administrador introduce el identificador del usuario para el cual desea se le muestren las incidencias.	<p>1.5 El sistema valida que el actor sea un Administrador.</p> <p>1.6 El sistema le pide el usuario para el cual desea que se le muestre el reporte de incidencias.</p> <p>2.1 El sistema muestra los datos relacionados con las incidencias del usuario seleccionado (fecha, hora, ip de la conexión y la URL visitada).</p>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

	<p>1.5 Si el actor es un usuario común, este le muestra la opción de ver exclusivamente sus incidencias.</p> <p>2.1 Si el sistema no encuentra ningún resultado correspondiente a la búsqueda, le muestra un mensaje notificando que el Usuario no tiene incidencias registradas.</p>
Poscondiciones:	El Administrador y el Usuario obtienen su reporte.
Prioridad:	Auxiliar
Especificaciones:	
Complementaria:	

### Caso de Uso: Mostrar ranking de URLs denegadas

Caso de Uso:	Mostrar ranking de URLs denegadas
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema para obtener un reporte con el ranking de URLs denegadas.
Referencia:	R20
CU asociados:	Gestionar reporte (extensión).
Precondiciones:	Que el actor se autentique como Administrador.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<p>2. El Administrador selecciona el rango de fecha.</p> <p>3. El Administrador selecciona el número de URLs que desea que le sean mostradas.</p>	<p>1.5 El sistema solicita el rango de fecha.</p> <p>2.1 El sistema solicita el número de URLs que desea que le sean mostradas para este reporte.</p> <p>3.1 El sistema valida que el Administrador haya llenado los datos correctamente.</p> <p>3.2 El sistema realiza la búsqueda en la Base de Datos.</p> <p>3.3 El sistema muestra los resultados encontrados, ordenándolas descendientemente por las URLs más denegadas.</p>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

	<p>3.1a Si el Administrador no llenó los datos correctamente, el sistema muestra un mensaje de advertencia y regresa al paso 2.</p> <p>3.3 Si el sistema no encuentra ningún resultado correspondiente en la búsqueda, le muestra un mensaje al Administrador y regresa al paso 2.</p>
Poscondiciones:	El Administrador obtiene su reporte.
Prioridad:	Auxiliar
Especificaciones:	
Complementaria:	

### Caso de Uso: Mostrar ranking de usuarios por incidencias

Caso de Uso:	Mostrar ranking de usuarios por incidencias
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte con el ranking de usuarios por incidencias.
Referencia:	R21
CU asociados:	Gestionar reporte (extensión).
Precondiciones:	Que el actor se autentique como Administrador.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
Continúa en la próxima página	

<p>2. El Administrador entra el rango de fecha.</p>	<p>1.5 El sistema solicita el rango de fecha.</p> <p>2.1 El sistema valida la información entrada por el Administrador.</p> <p>2.2 El sistema realiza la búsqueda en la Base de Datos.</p> <p>2.3 El sistema muestra los resultados ordenados descendientemente por la cantidad de incidencias y mostrando el nombre de usuario denegadas.</p>
<p><b>Flujo alterno</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
	<p>2.1 Si el Administrador no entra los datos de la fecha, el sistema le muestra un mensaje de advertencia y se le permite regresar al paso 2.</p> <p>2.2 Si el sistema no encuentra ningún resultado correspondiente a la búsqueda, le muestra un mensaje al Administrador y puede regresar al paso 2.</p>
<p>Continúa en la próxima página</p>	

Poscondiciones:	El Administrador obtiene su reporte.
Prioridad:	Auxiliar
Especificaciones:	
Complementaria:	

**Caso de Uso: Mostrar ranking de días de la semana por incidencias**

Caso de Uso:	Mostrar ranking de días de la semana por incidencias	
Actores:	Administrador	
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte con el ranking de días de la semana por incidencias.	
Referencia:	R22	
CU asociados:	Gestionar reporte (extensión).	
Precondiciones:	Que el actor se autentique como Administrador.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “nombre del escenario”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
Continúa en la próxima página		

<p>2. El Administrador entra el rango de fecha.</p>	<p>1.5 El sistema solicita el rango de fecha.</p> <p>2.1 El sistema valida la información entrada por el Administrador.</p> <p>2.2 El sistema realiza la búsqueda en la Base de Datos.</p> <p>2.3 El sistema muestra los resultados encontrados ordenados descendientemente; con el día de la semana, la fecha y la cantidad de incidencias correspondientes a dicho día.</p>
<b>Flujo alterno</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>2.2 Si el sistema no encuentra ningún resultado correspondiente a la búsqueda, le muestra un mensaje al Administrador y puede regresar al paso 2.</p>
Poscondiciones:	El Administrador obtiene su reporte.
Prioridad:	Auxiliar
Especificaciones:	
Complementaria:	

**Caso de Uso: Mostrar cantidad de incidencias por tipos de usuarios**

Caso de Uso:	Mostrar cantidad de incidencias por tipos de usuarios
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte de la cantidad de incidencias por tipos de usuarios.
Referencia:	R23
CU asociados:	Gestionar reporte (extensión).
Precondiciones:	Que el actor se autentique como Administrador.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>2. El Administrador selecciona el rango de fecha.</p> <p>3. El Administrador selecciona el tipo de usuario.</p>	<p>1.5 El sistema solicita el rango de fecha.</p> <p>2.1 El sistema valida la información entrada por el Administrador.</p> <p>2.2 El sistema solicita el tipo de usuario (Mayor de edad, Menor de edad).</p> <p>3.1 El sistema realiza la búsqueda en la Base de Datos.</p> <p>3.2 El sistema muestra los resultados encontrados ordenados descendientemente, con el Nombre de los usuarios.</p>
<b>Flujo alterno</b>	
Continúa en la próxima página	



Acción del Actor	Respuesta del Sistema
	<p>2.1 Si la información no está correcta, el sistema muestra un mensaje de advertencia y regresa al paso 2.</p> <p>3.1 Si el sistema no encuentra ningún resultado correspondiente a la búsqueda, le muestra un mensaje al Administrador y puede regresar al paso 2.</p>
Poscondiciones:	El Administrador obtiene su reporte.
Prioridad:	Auxiliar
Especificaciones:	
Complementaria:	

**Caso de Uso: Mostrar reporte de cantidad de sitios por categorías**

Caso de Uso:	Mostrar reporte de cantidad de sitios por categorías
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte de la cantidad de sitios por categorías.
Referencia:	R24
CU asociados:	Gestionar reporte (extensión).
Precondiciones:	Que el actor se autentique como Administrador.
<b>Flujo Normal de Eventos</b>	
<b>Sección “nombre del escenario”</b>	
Continúa en la próxima página	

Acción del Actor		Respuesta del Sistema
2. El Administrador selecciona la categoría para la cual desea ver la cantidad de sitios clasificados (puede seleccionar una, varias o todas las categorías existentes).		<p>1.5 El sistema le brinda la opción de seleccionar la categoría.</p> <p>2.1 El sistema realiza la búsqueda en la Base de Datos.</p> <p>2.2 El sistema muestra el nombre de la(s) categoría(s) y la cantidad de URL clasificadas con dicha(s) categoría(s).</p>
<b>Flujo alterno</b>		
Acción del Actor		Respuesta del Sistema
Poscondiciones:	El Administrador obtiene su reporte.	
Prioridad:	Auxiliar	
Especificaciones:		
Complementaria:		

**Caso de Uso: Mostrar reportes de posibles regulaciones incorrectas**

Caso de Uso:	Mostrar reportes de posibles regulaciones incorrectas
Actores:	Administrador
Resumen:	El Caso de Uso se inicia cuando el Administrador entra al sistema y desea obtener un reporte de las posibles regulaciones incorrectas.
Referencia:	R25
Continúa en la próxima página	

CU asociados:	Gestionar reporte (extensión).	
Precondiciones:	Que el actor se autentique como Administrador.	
<b>Flujo Normal de Eventos</b>		
<b>Sección “nombre del escenario”</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<b>Flujo alterno</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	1.5 El sistema le muestra el reporte de las regulaciones registradas por el Usuario.	
Poscondiciones:	El Administrador obtiene su reporte.	
Prioridad:	Auxiliar	
Especificaciones:		
Complementaria:		

---

## Anexo C

# Diagramas de clases del Análisis

---

### Módulo Usuario

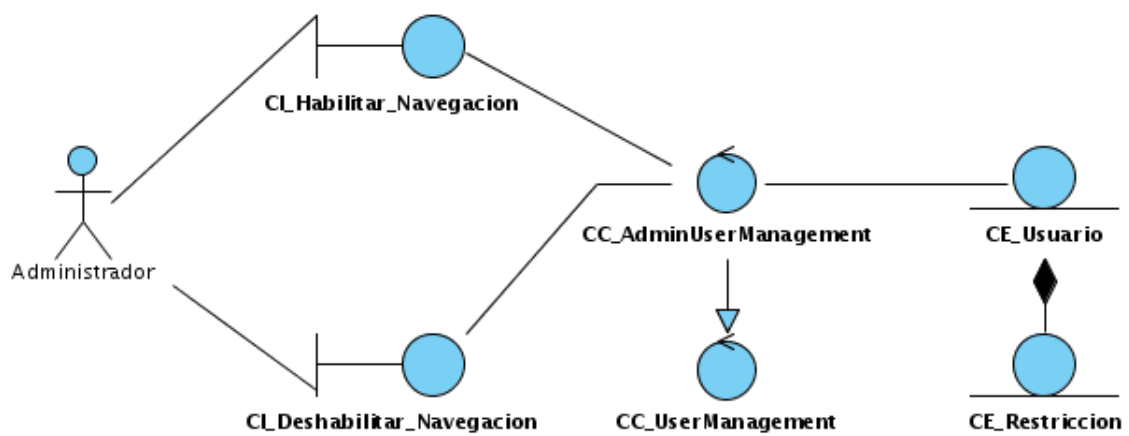


Figura C.1: Diagrama de clases del Análisis - Casos de Uso habilitar y deshabilitar navegación.

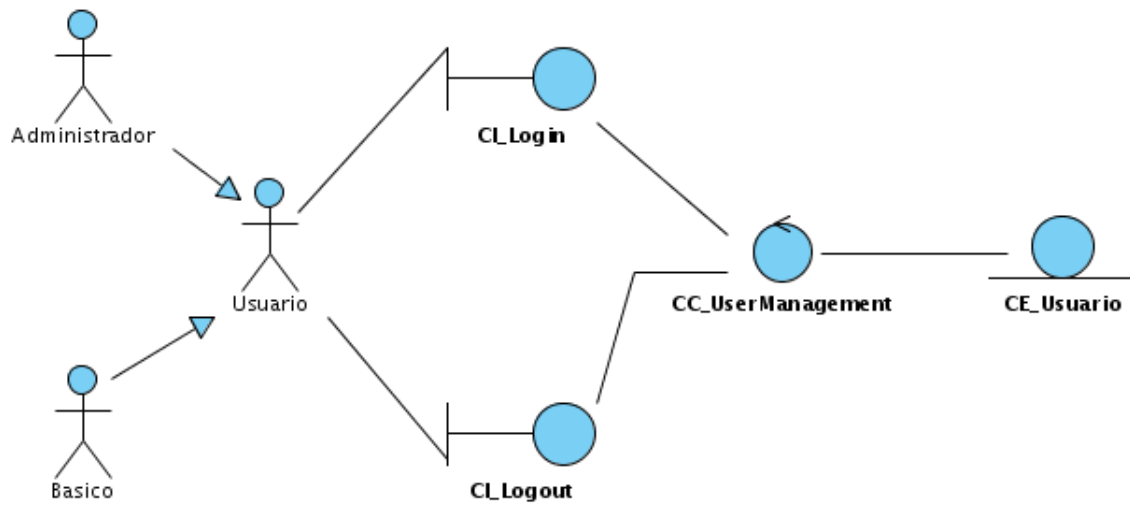


Figura C.2: Diagrama de clases del Análisis - Casos de Uso login y logout.

## Módulo Reporte

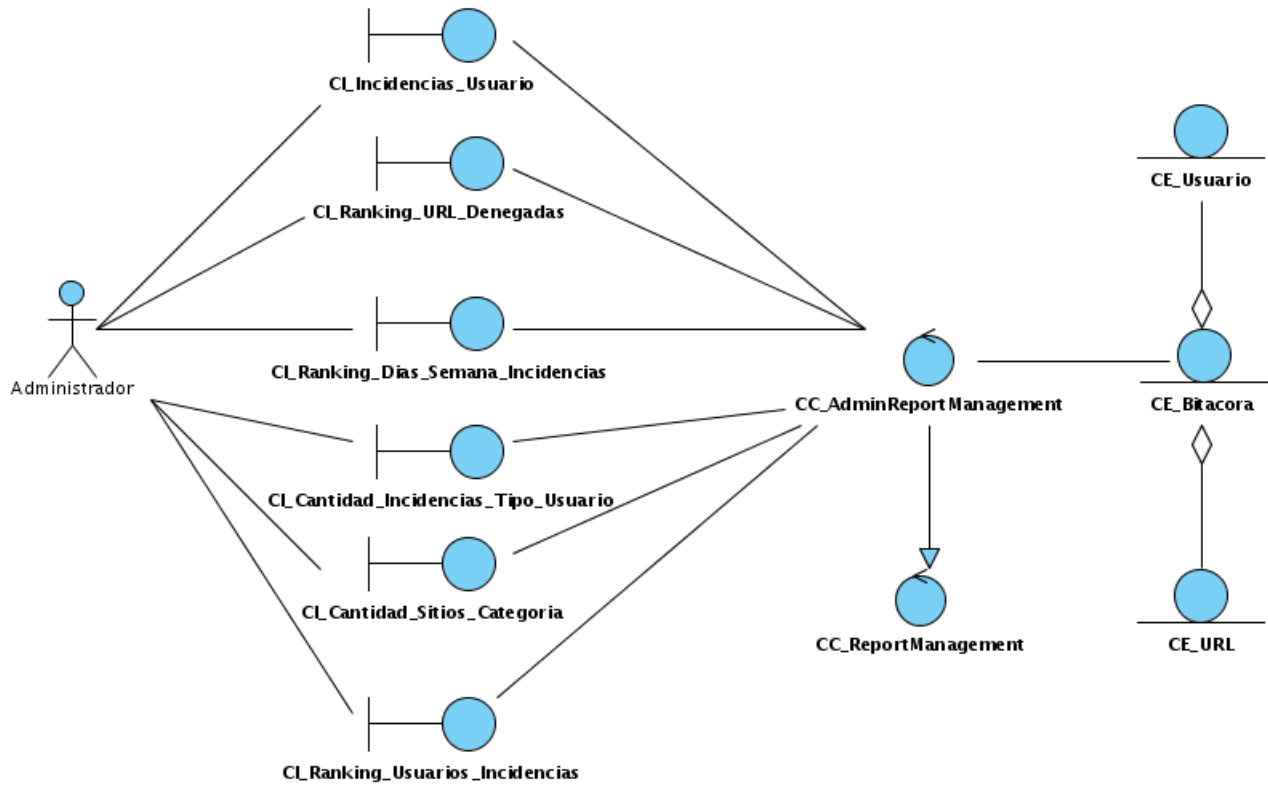


Figura C.3: Diagrama de clases del Análisis - Casos de Uso incidencias de usuario, ranking de URLs dene-gadas, ranking de días de la semana por incidencias, cantidad de incidencias por tipo de usuario, cantidad de sitios por categoría y rankig de usuarios por incidencias.

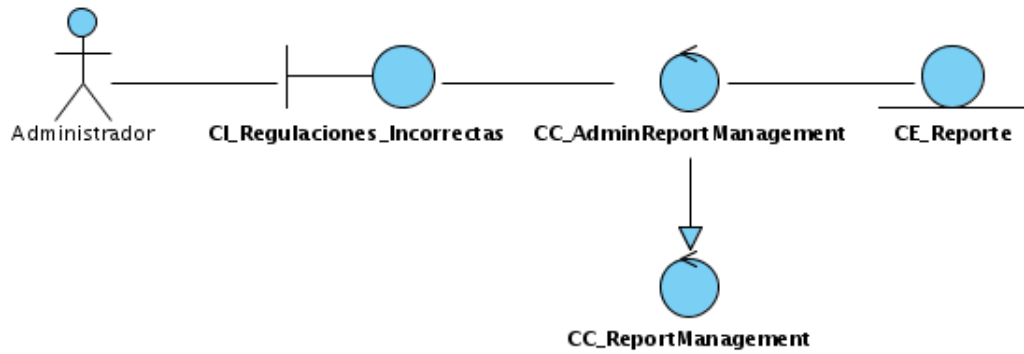


Figura C.4: Diagrama de clases del Análisis - Casos de Uso regulaciones incorrectas.

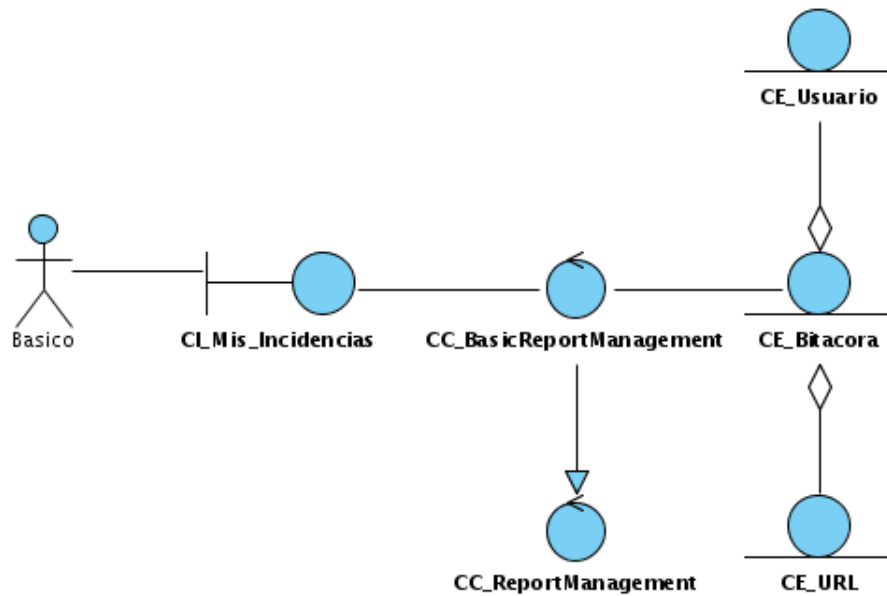


Figura C.5: Diagrama de clases del Análisis - Casos de Uso mis incidencias.

## Anexo D

# Diagramas de clases del Diseño

### Módulo Usuario

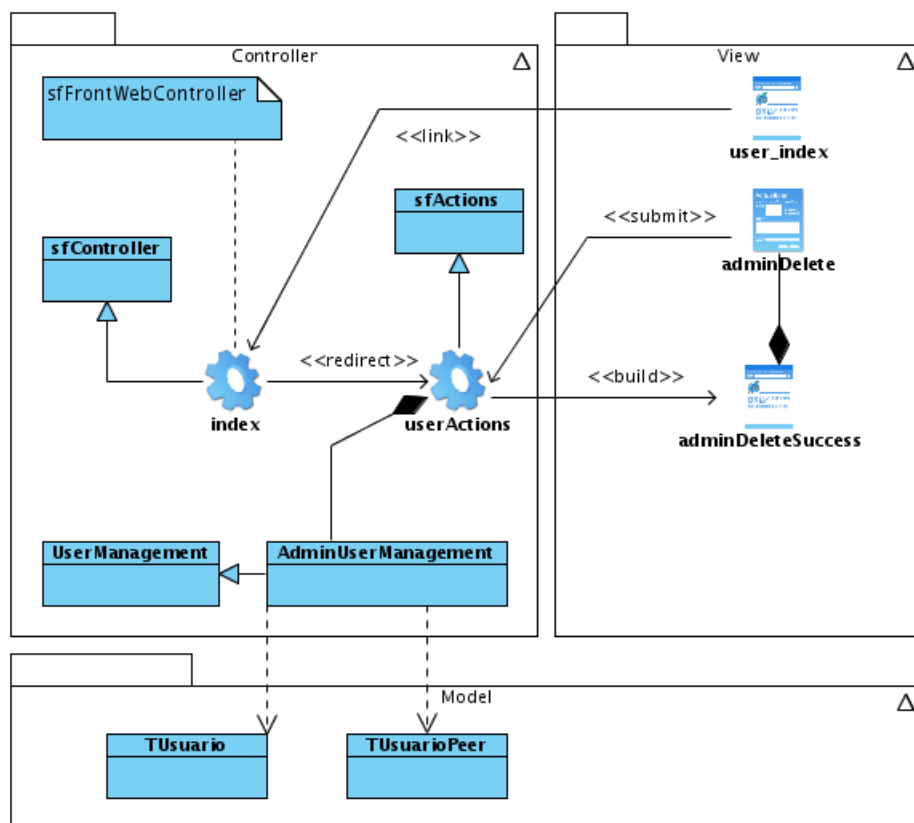


Figura D.1: Diagrama de clases del Diseño - Caso de Uso eliminar usuario.



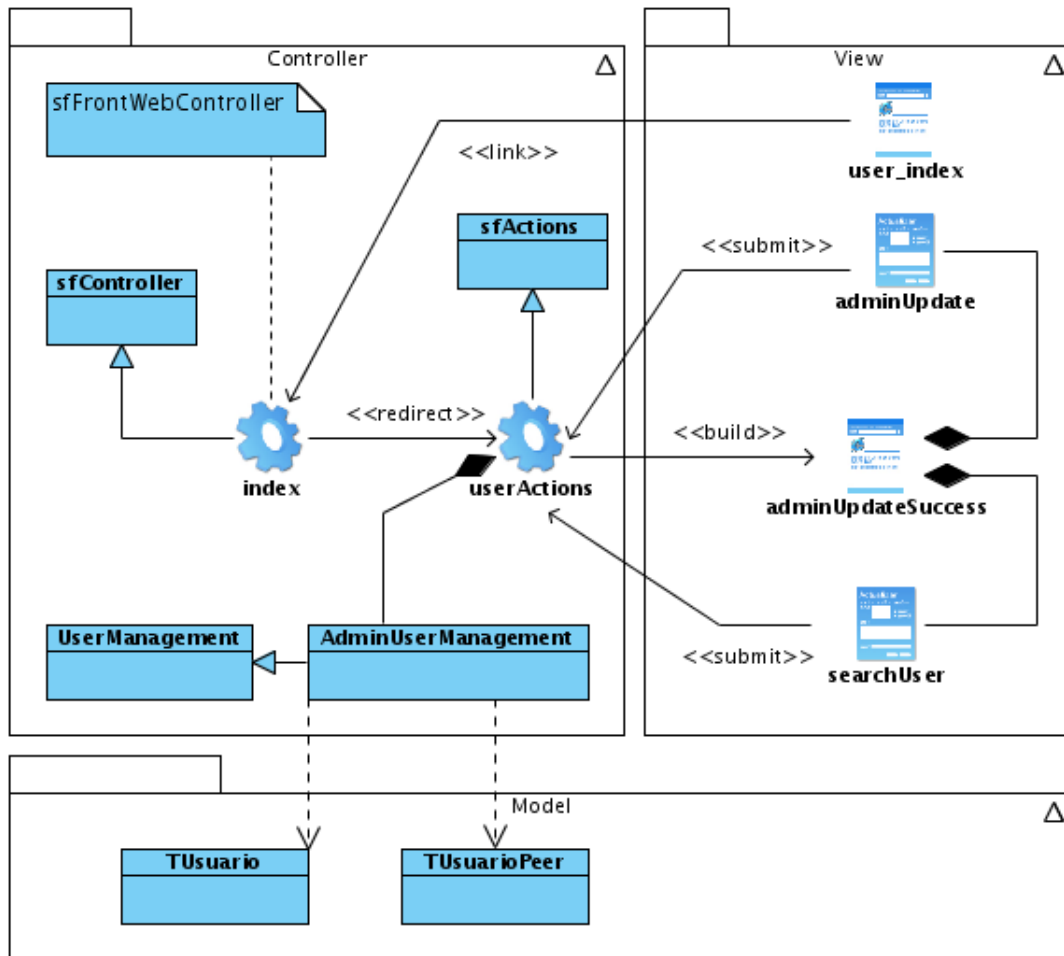


Figura D.2: Diagrama de clases del Diseño - Caso de Uso actualizar usuario por el administrador.

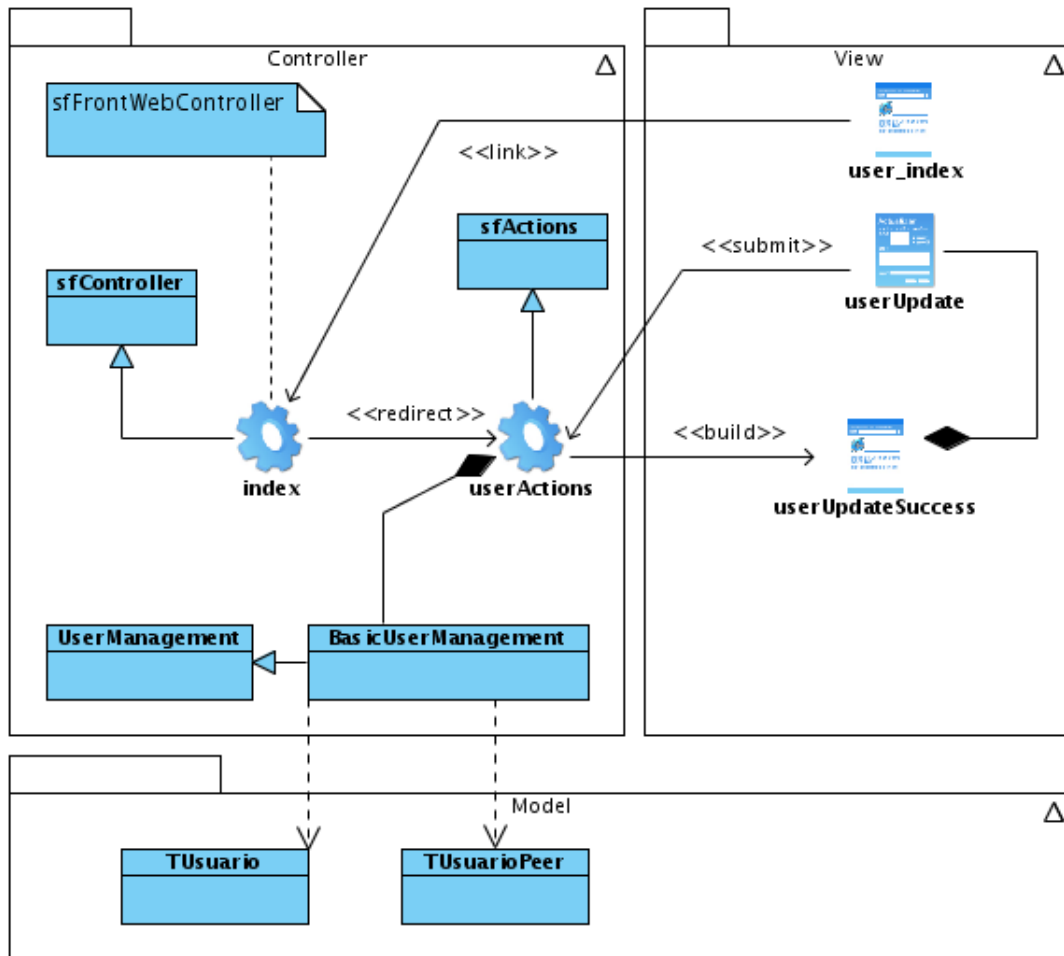


Figura D.3: Diagrama de clases del Diseño - Caso de Uso actualizar usuario por usuario básico.

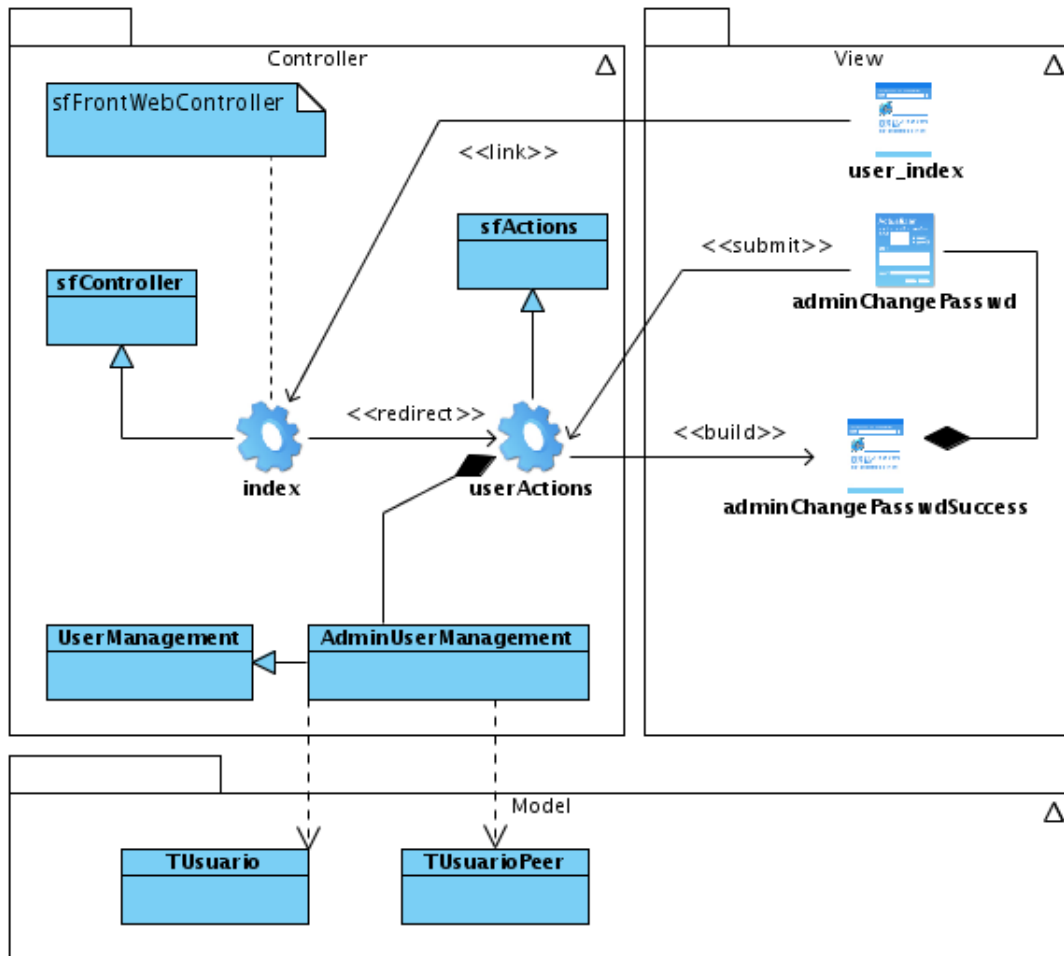


Figura D.4: Diagrama de clases del Diseño - Caso de Uso cambiar contraseña por el administrador.

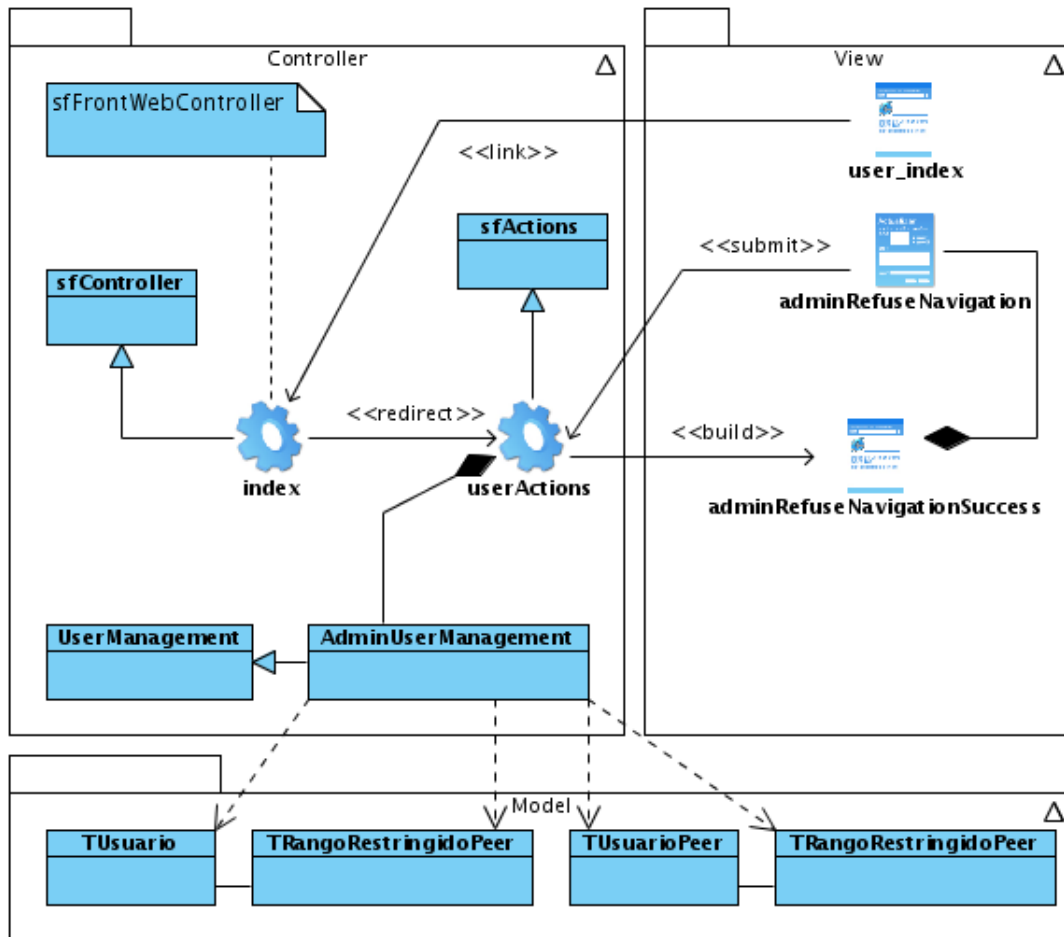


Figura D.5: Diagrama de clases del Diseño - Caso de Uso deshabilitar navegación.

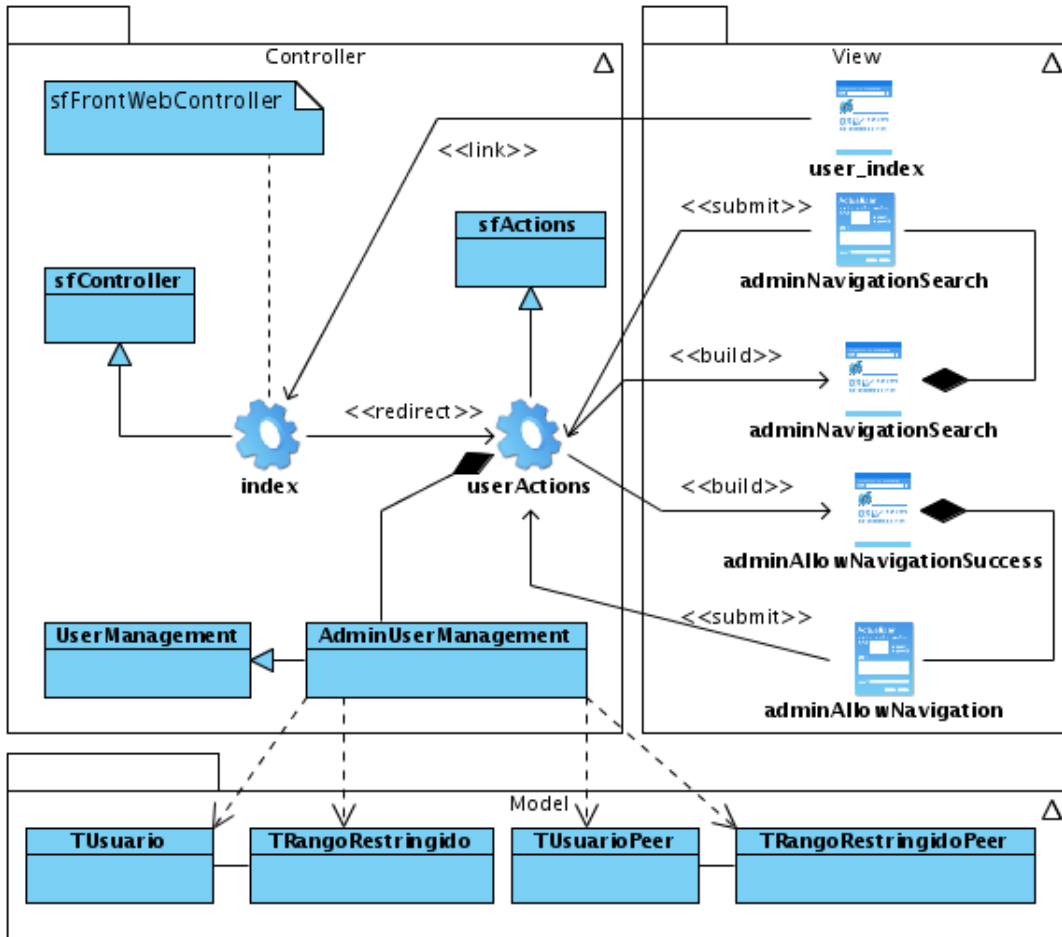


Figura D.6: Diagrama de clases del Diseño - Caso de Uso habilitar navegación.

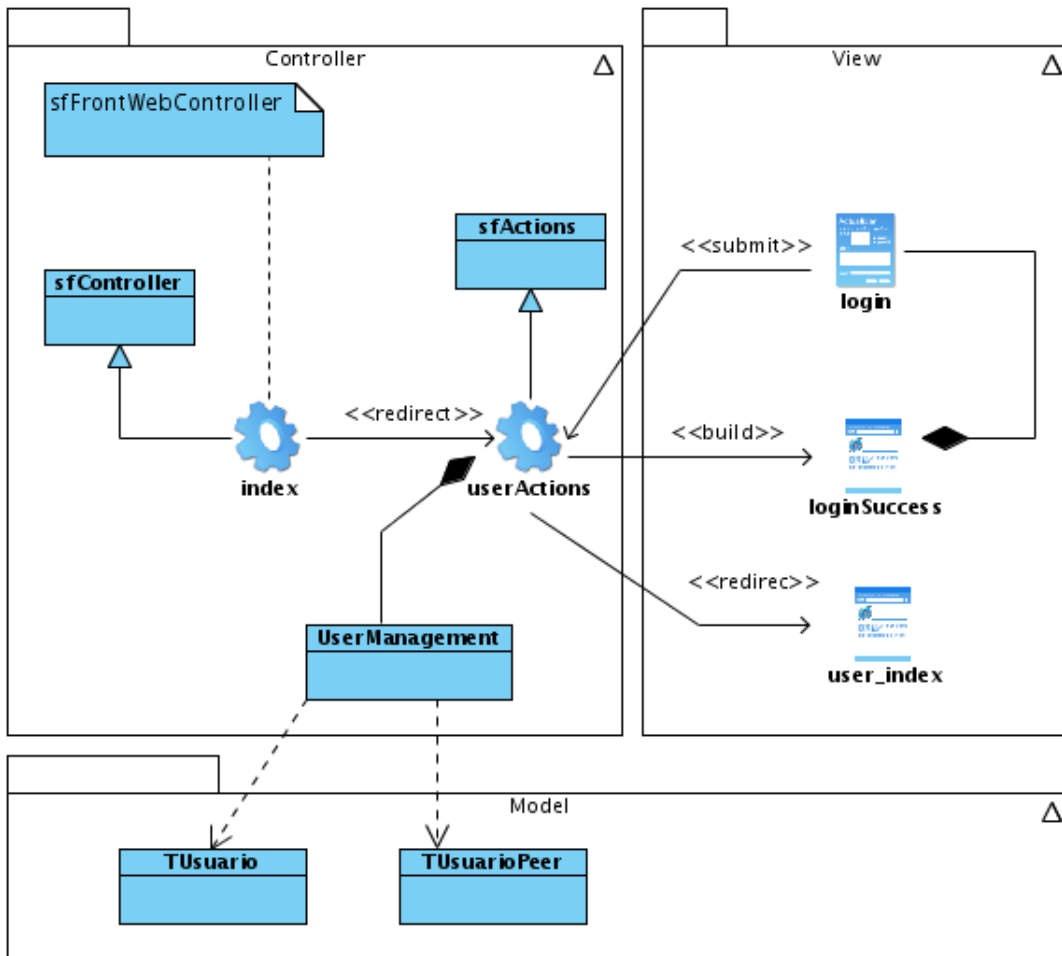


Figura D.7: Diagrama de clases del Diseño - Caso de Uso login.

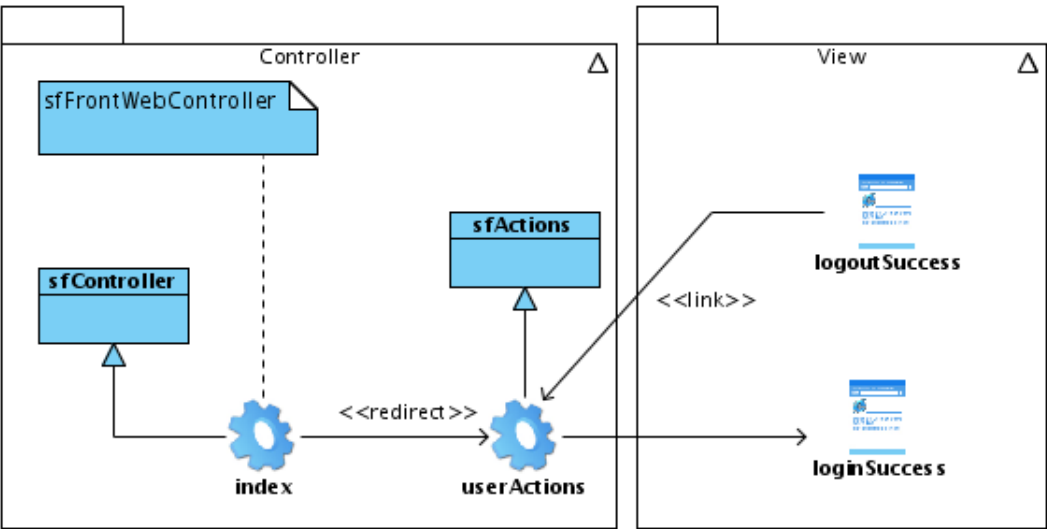


Figura D.8: Diagrama de clases del Diseño - Caso de Uso logout.

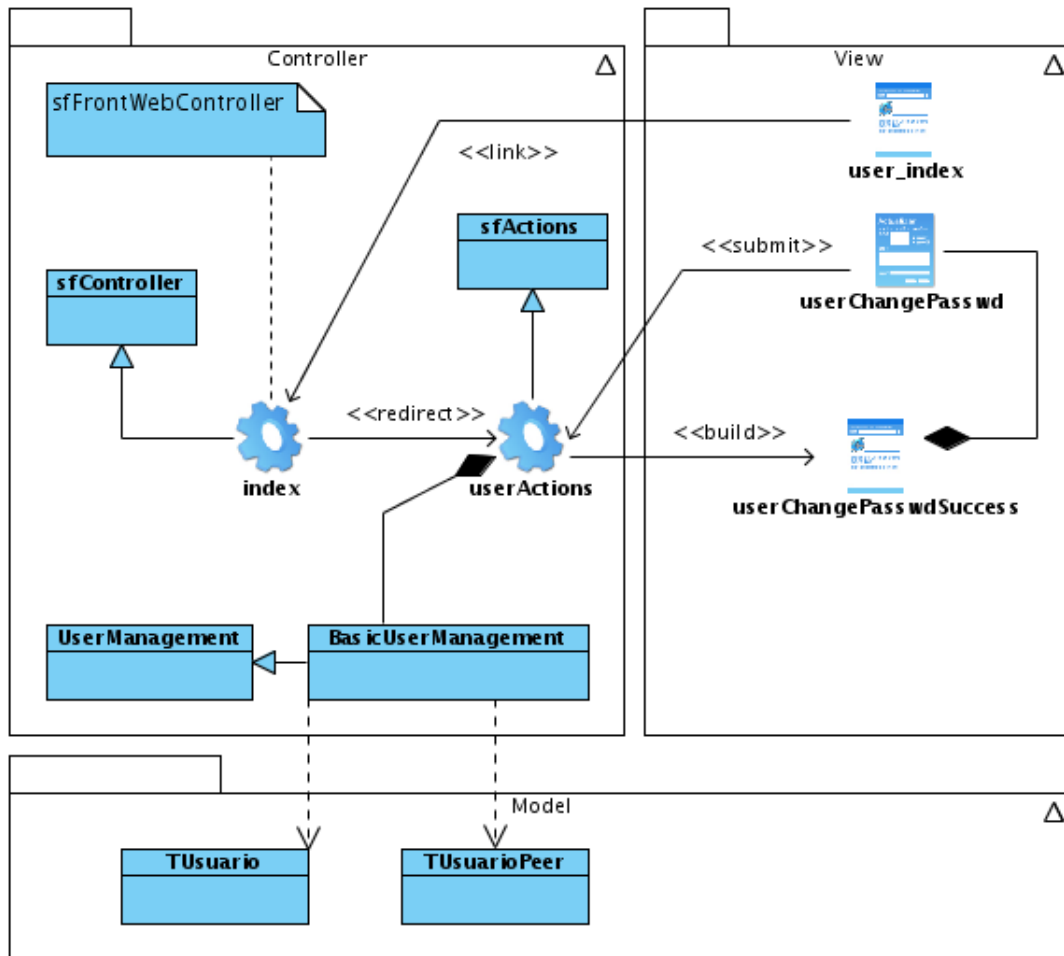


Figura D.9: Diagrama de clases del Diseño - Caso de Uso cambiar contraseña del usuario básico.



## Módulo URL

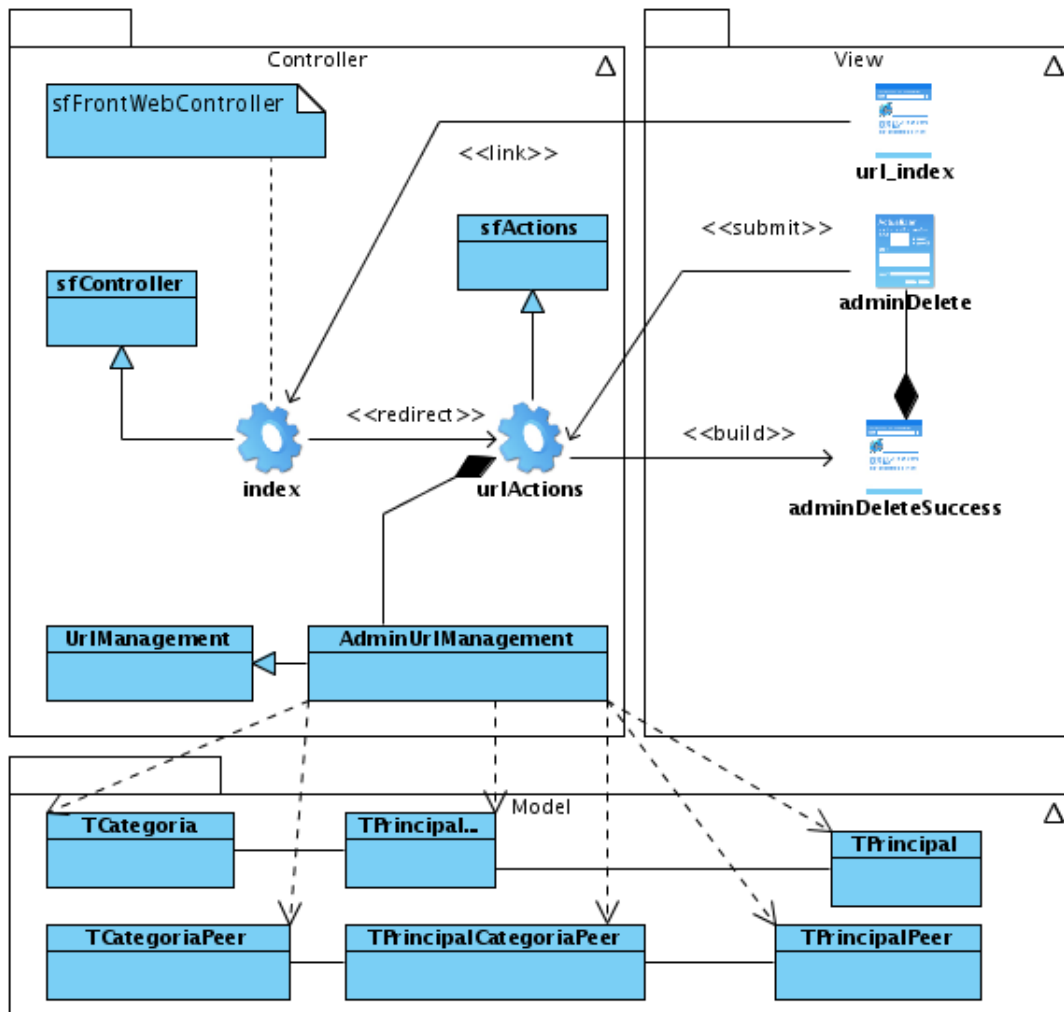


Figura D.10: Diagrama de clases del Diseño - Caso de Uso eliminar categoría de una URL.

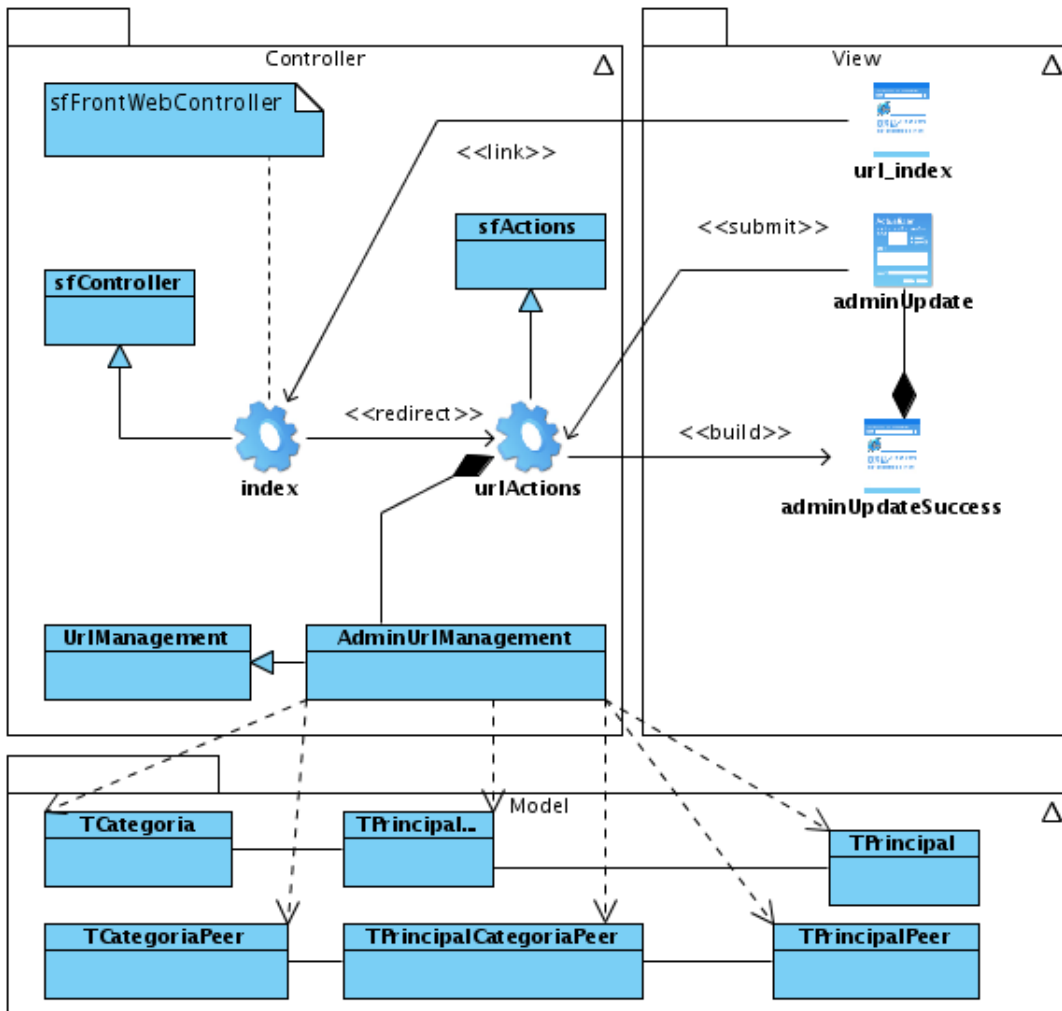


Figura D.11: Diagrama de clases del Diseño - Caso de Uso actualizar categoría de una URL.

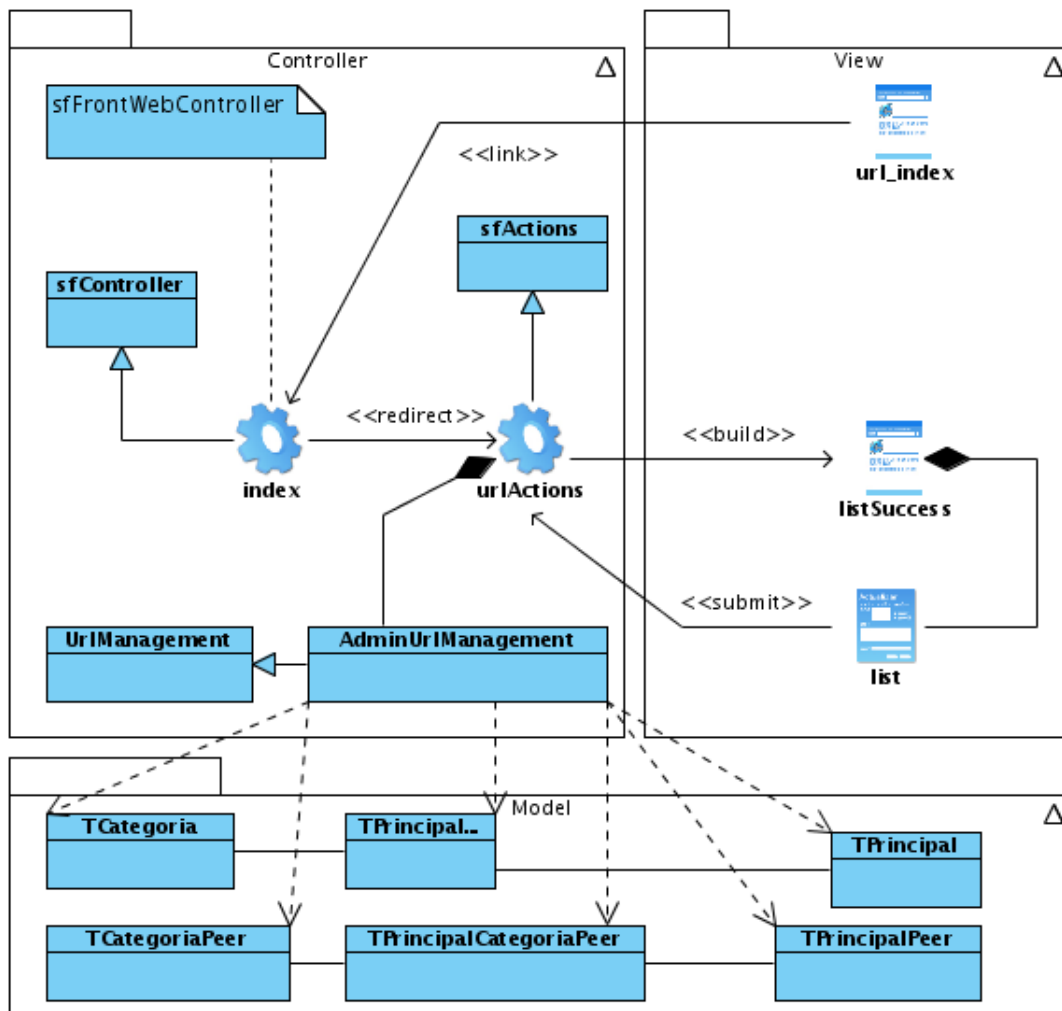


Figura D.12: Diagrama de clases del Diseño - Caso de Uso listar categorías de una URL.

## Módulo Categoría

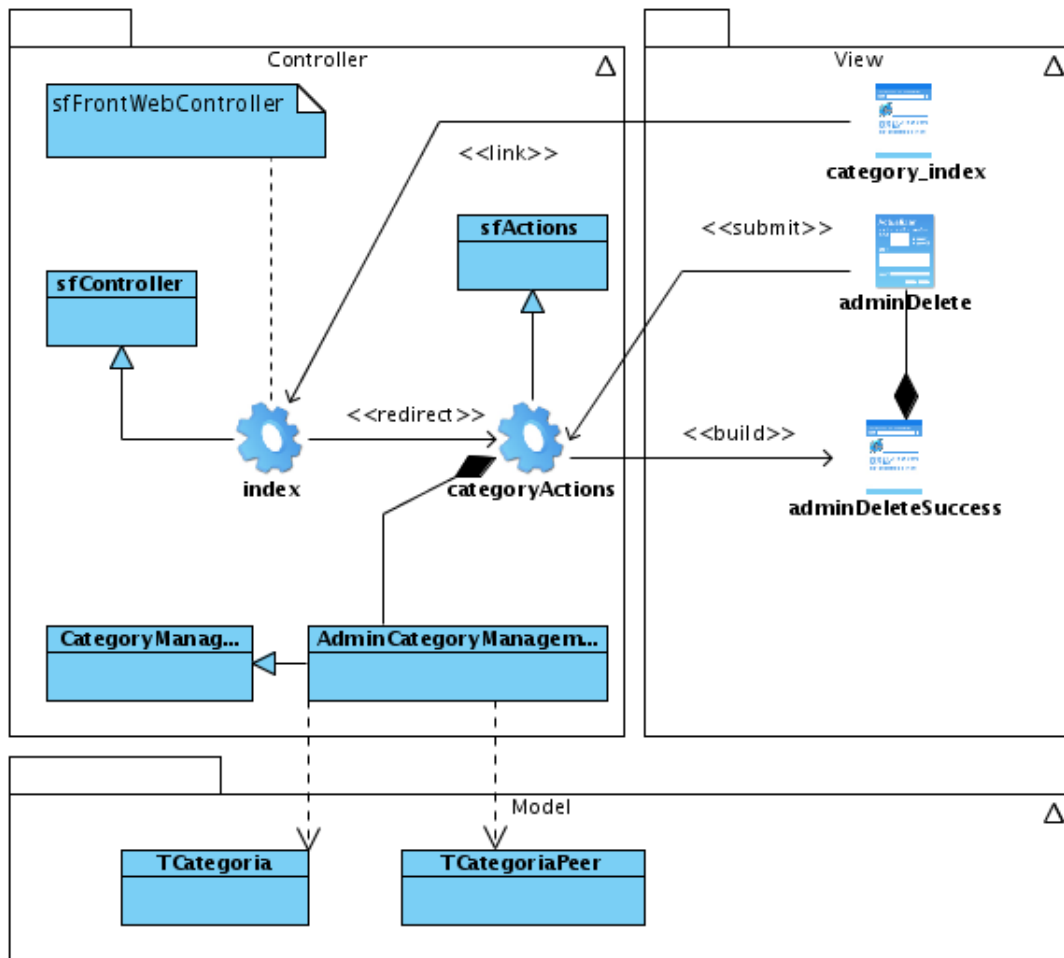


Figura D.13: Diagrama de clases del Diseño - Caso de Uso eliminar categoría.

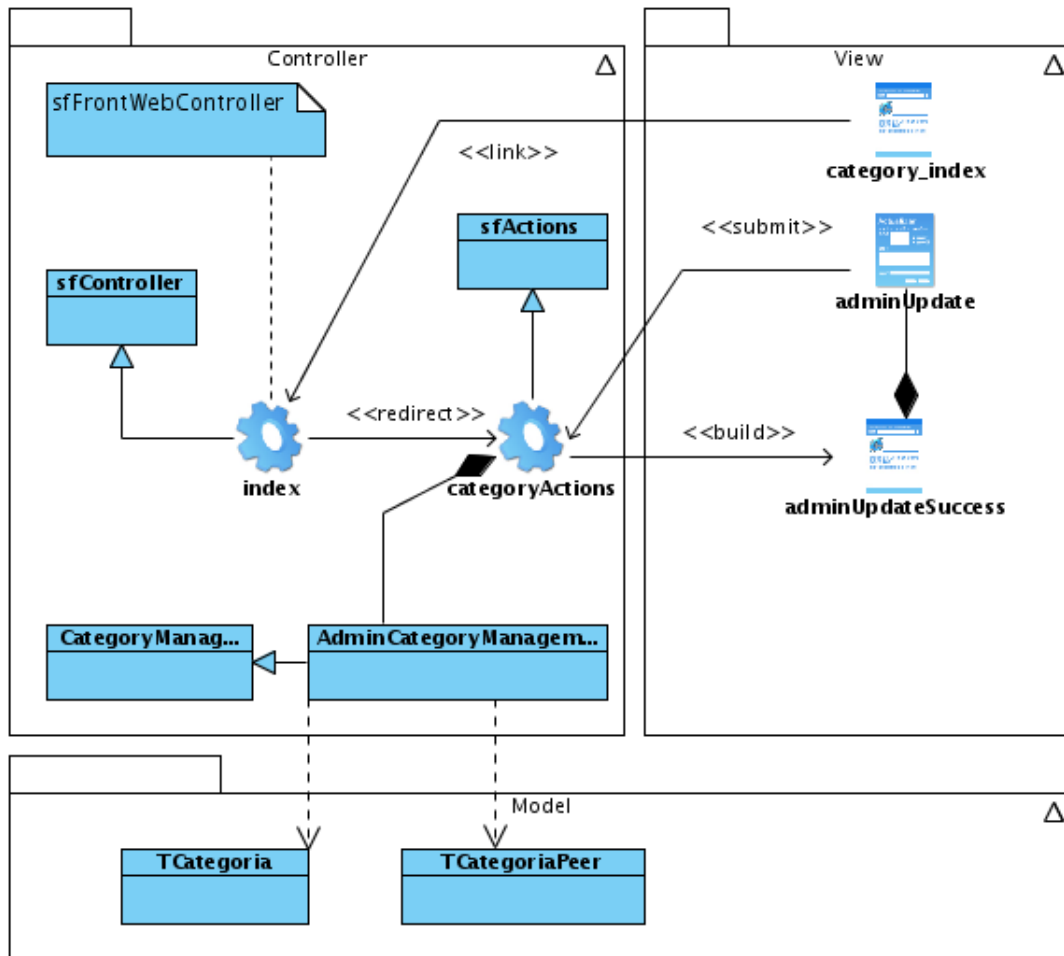


Figura D.14: Diagrama de clases del Diseño - Caso de Uso actualizar categoría.

## Módulo Categoría

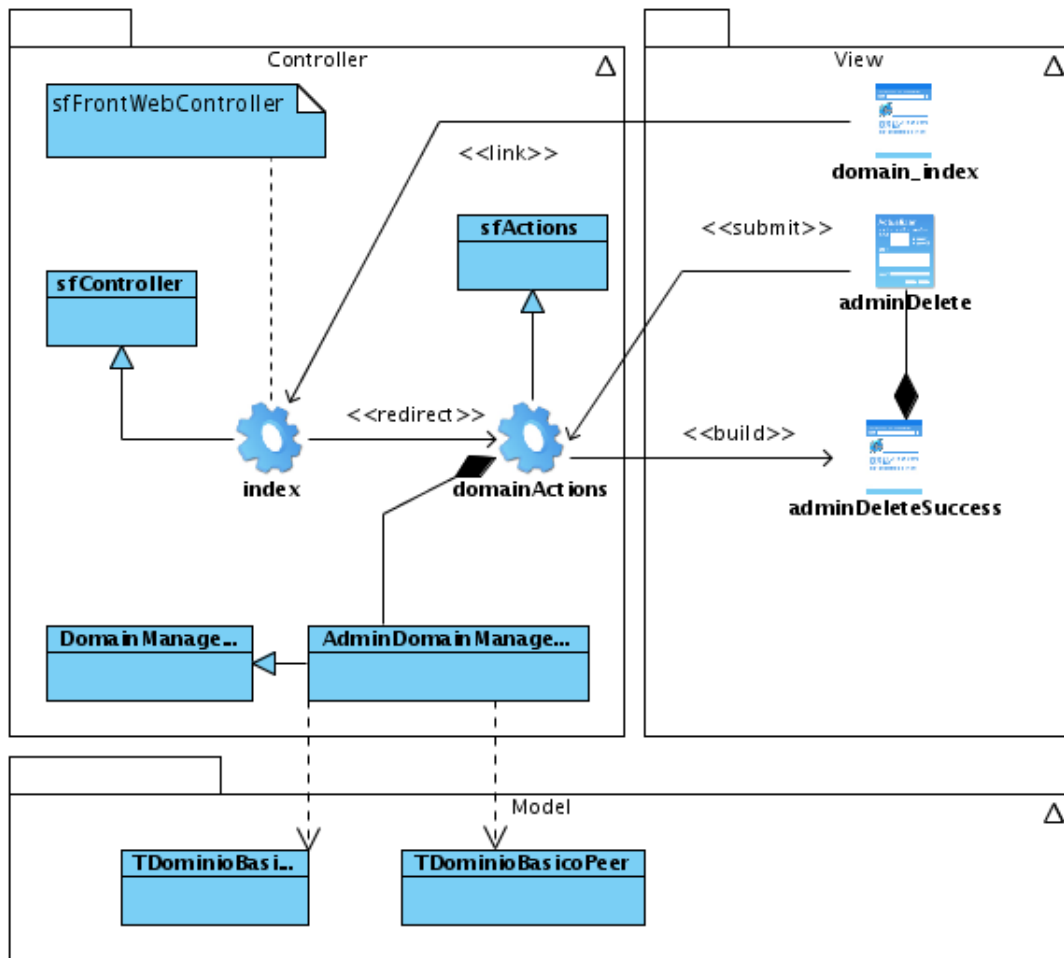


Figura D.15: Diagrama de clases del Diseño - Caso de Uso eliminar dominio de navegación básica.

## Módulo Reporte

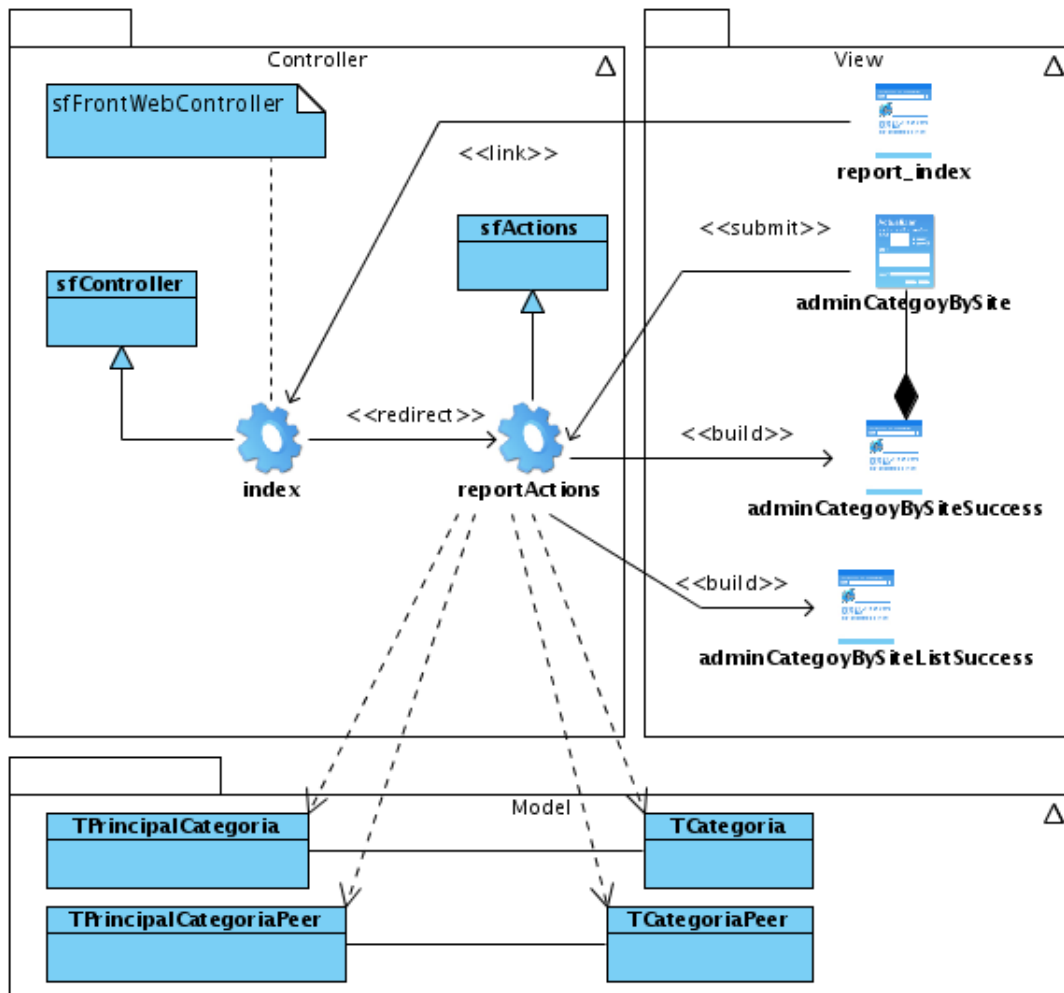


Figura D.16: Diagrama de clases del Diseño - Caso de Uso cantidad de sitios por categoría.

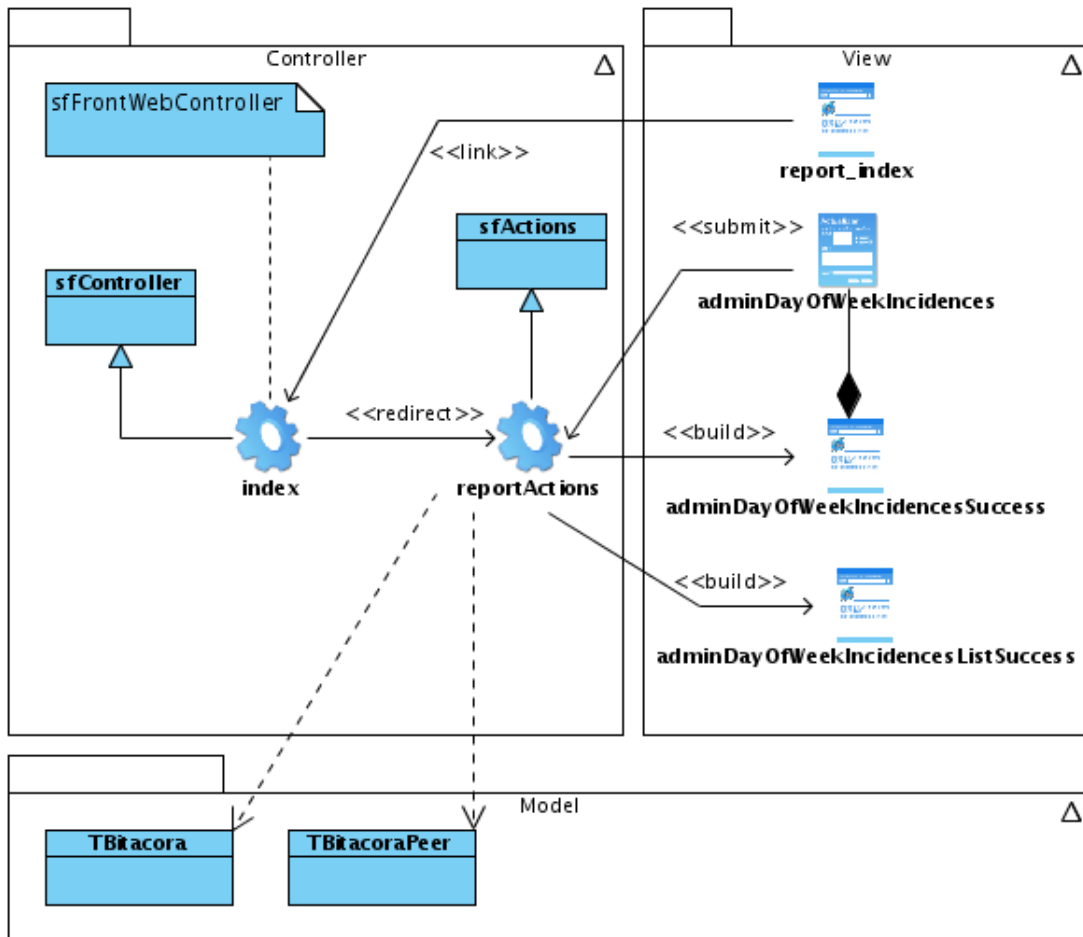


Figura D.17: Diagrama de clases del Diseño - Caso de Uso ranking de días de la semana por incidencias.



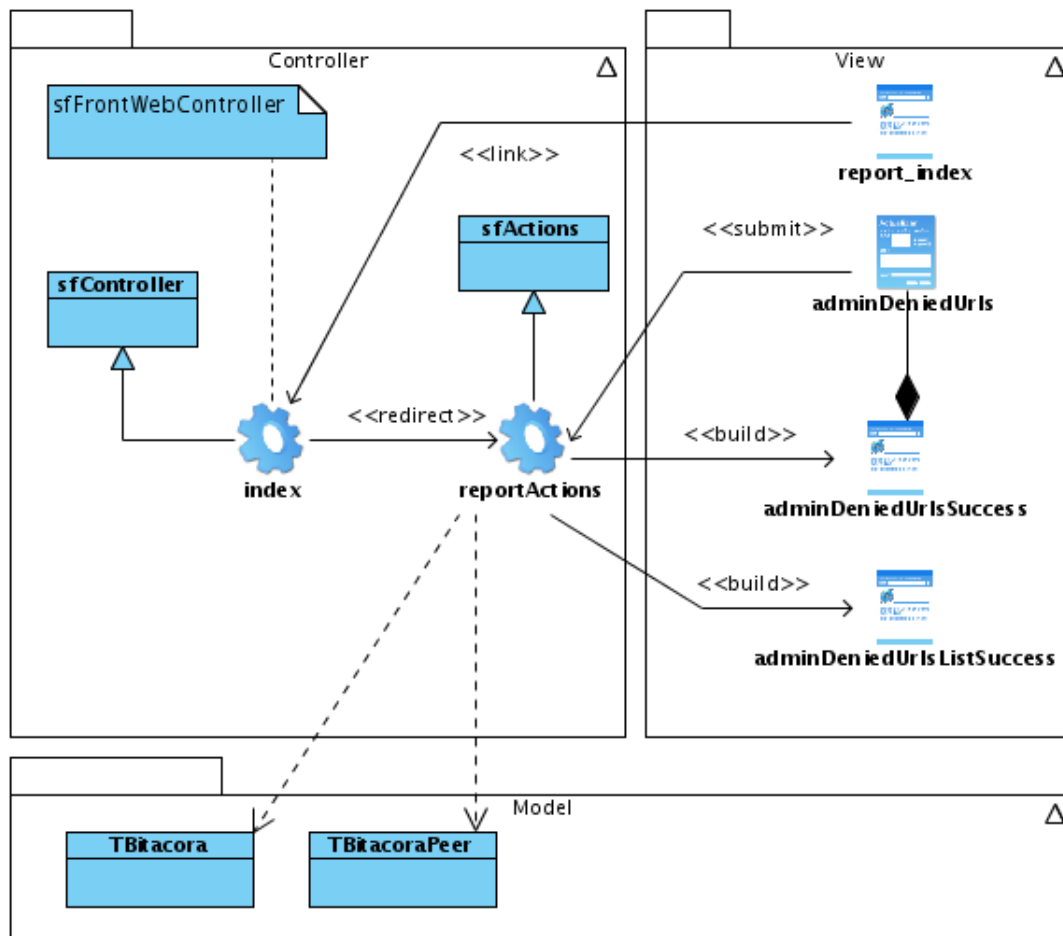


Figura D.18: Diagrama de clases del Diseño - Caso de Uso ranking de URLs denegadas.

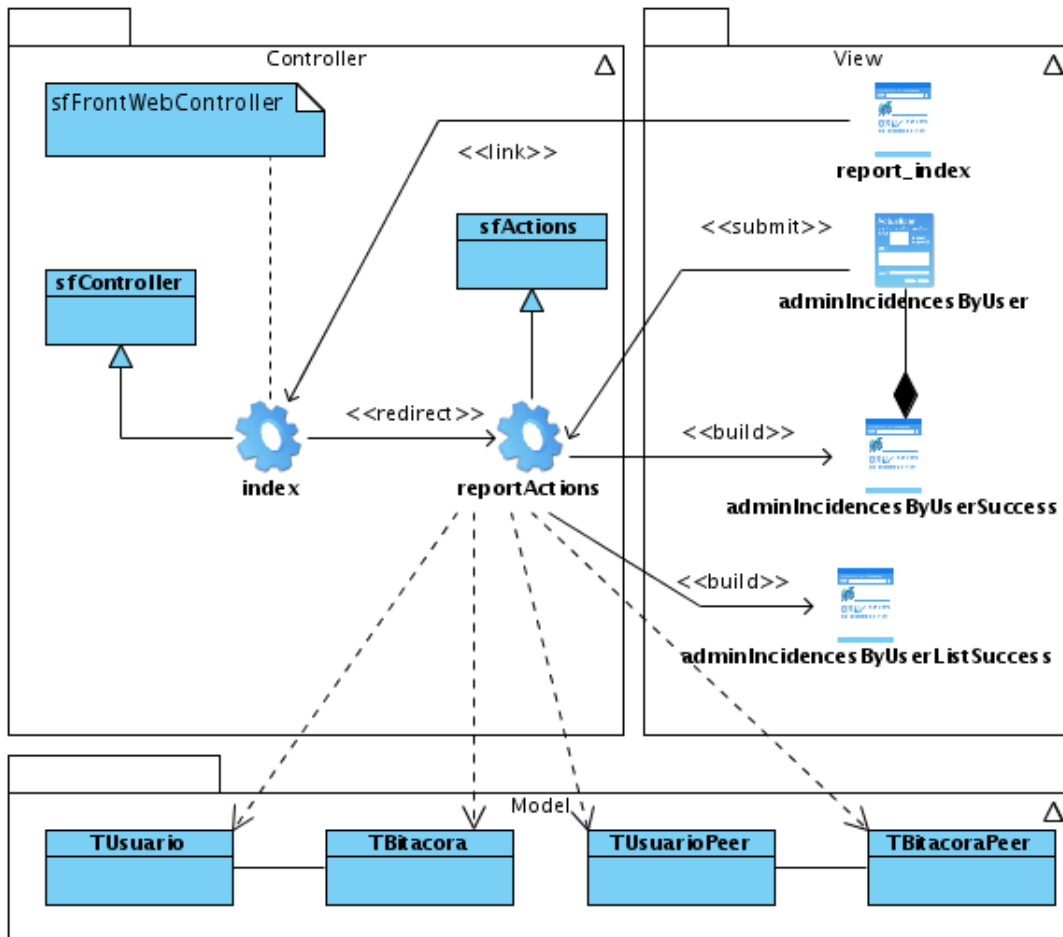


Figura D.19: Diagrama de clases del Diseño - Caso de Uso incidencia de usuario.

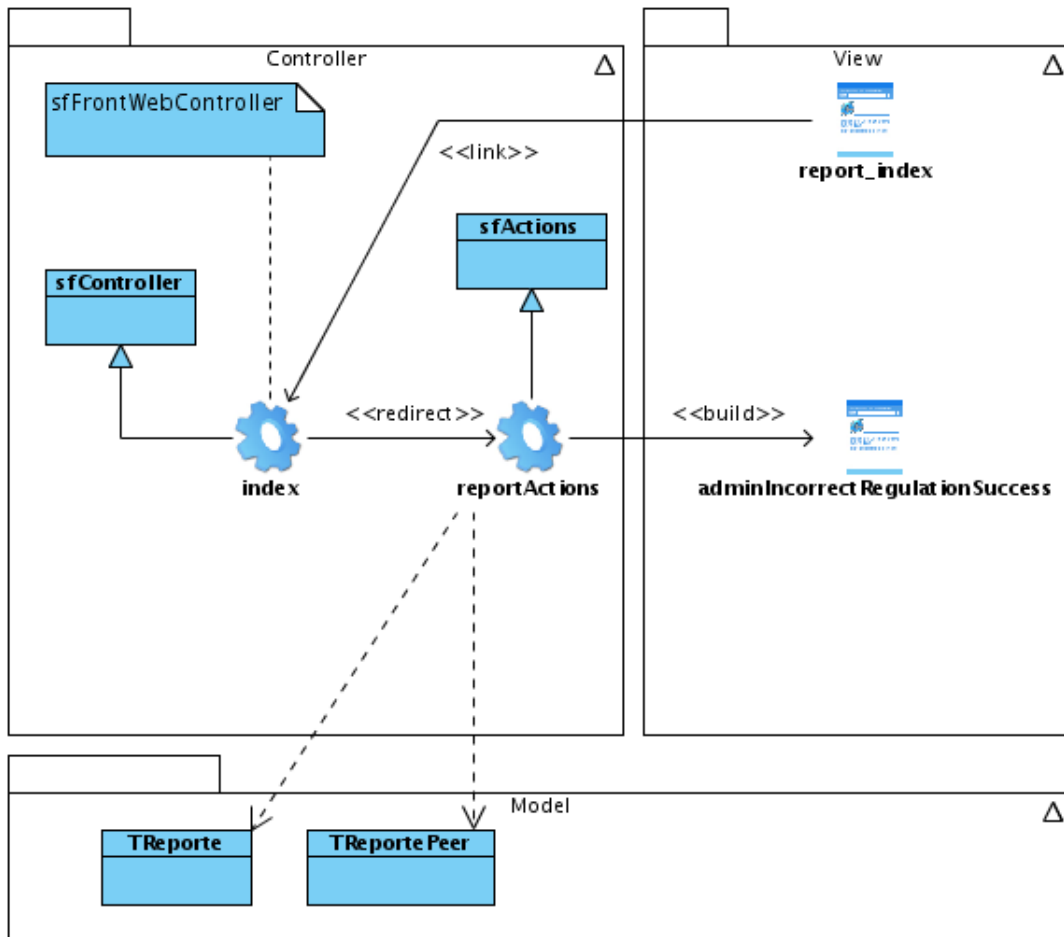


Figura D.20: Diagrama de clases del Diseño - Caso de Uso regulaciones incorrectas.

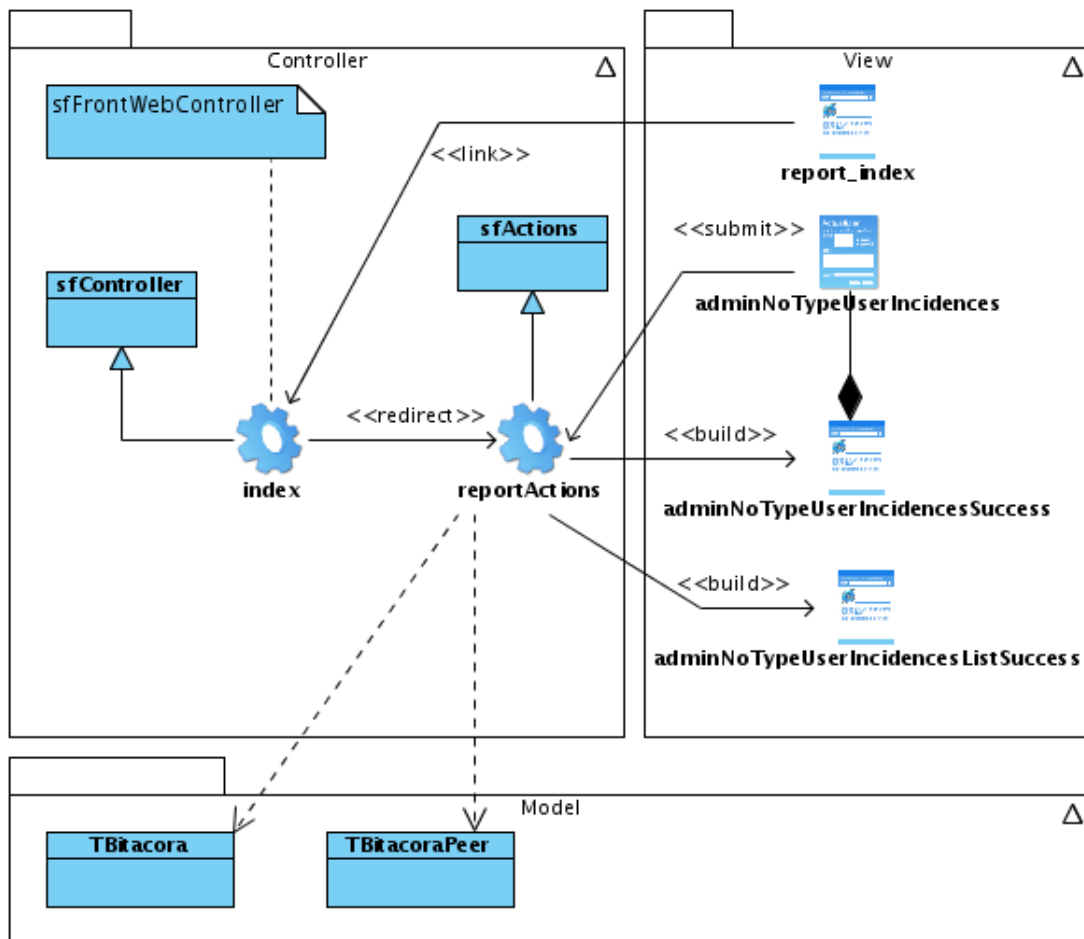


Figura D.21: Diagrama de clases del Diseño - Caso de Uso ranking incidencias por usuario.

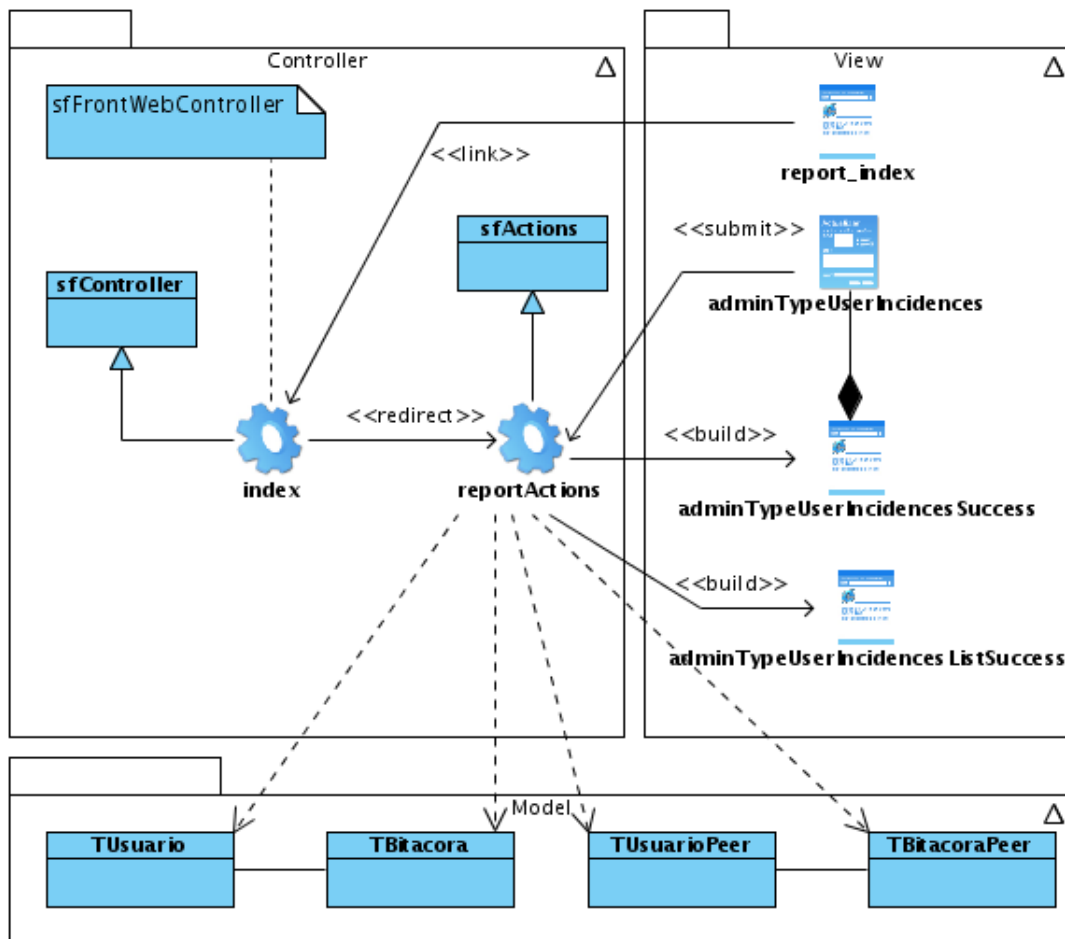


Figura D.22: Diagrama de clases del Diseño - Caso de Uso incidencias por tipo usuario.

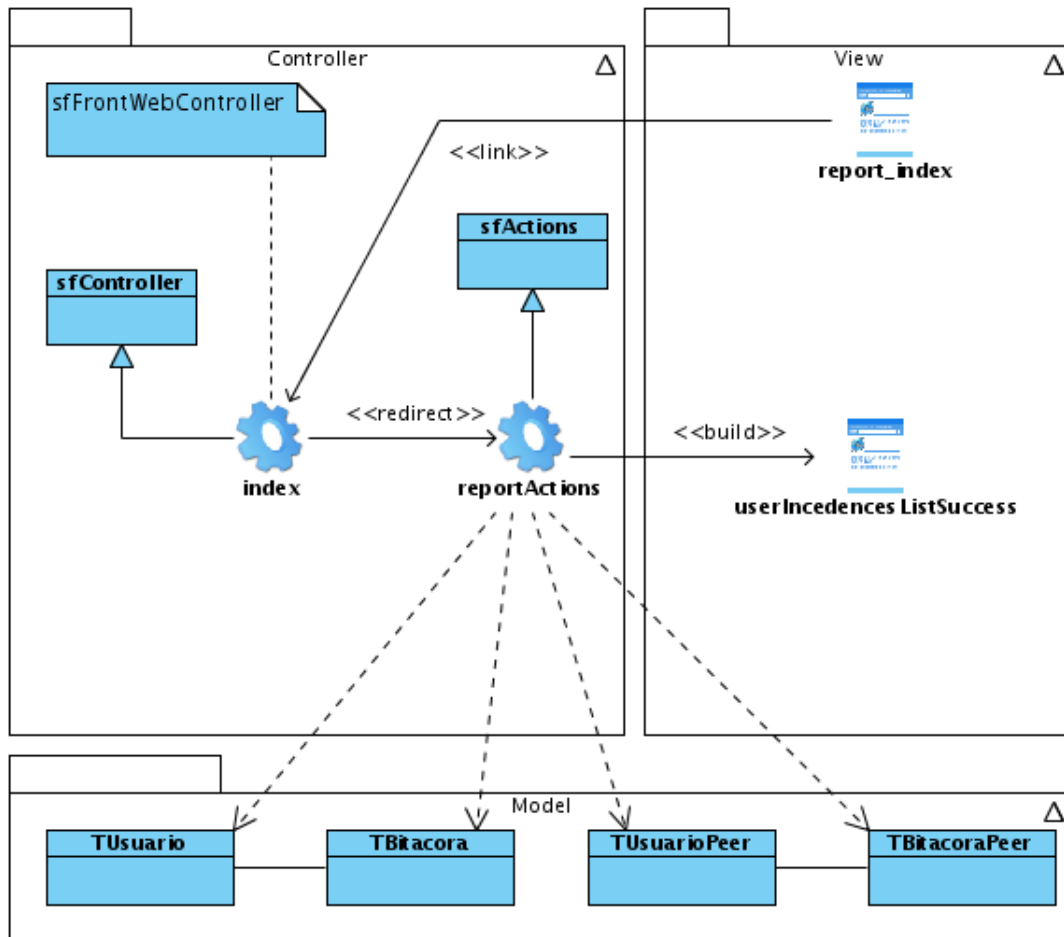


Figura D.23: Diagrama de clases del Diseño - Caso de Uso mis incidencias.

## Anexo E

# Diagramas de Interacción

### Módulo Usuario

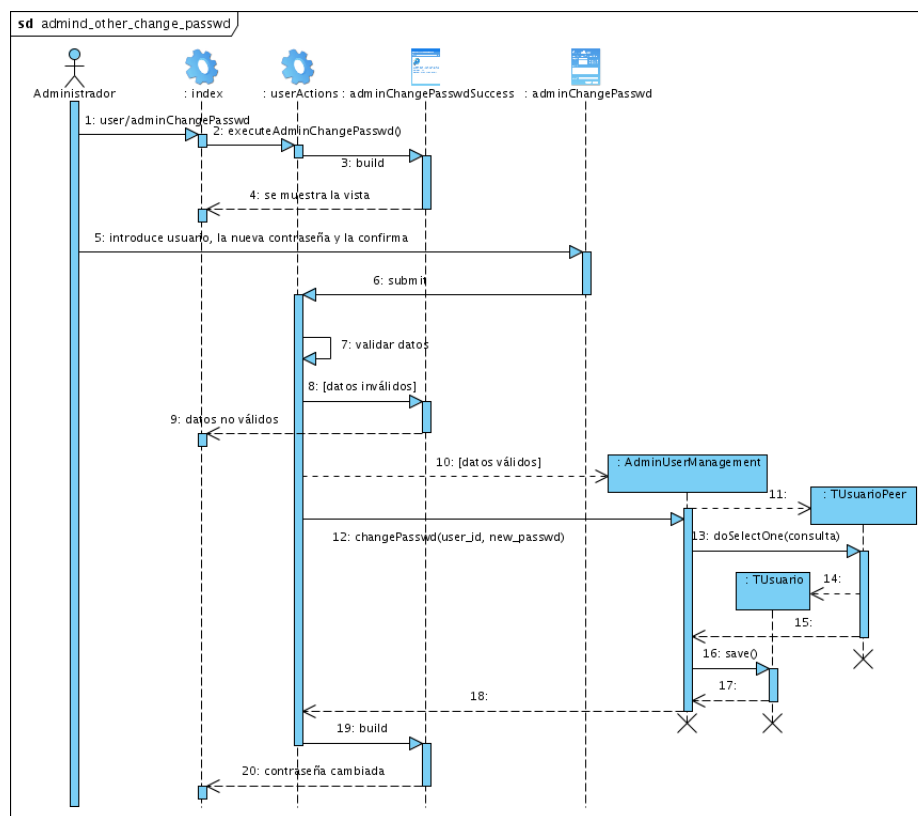


Figura E.1: Diagrama de Interacción - Caso de Uso cambiar contraseña administrador a otro usuario.

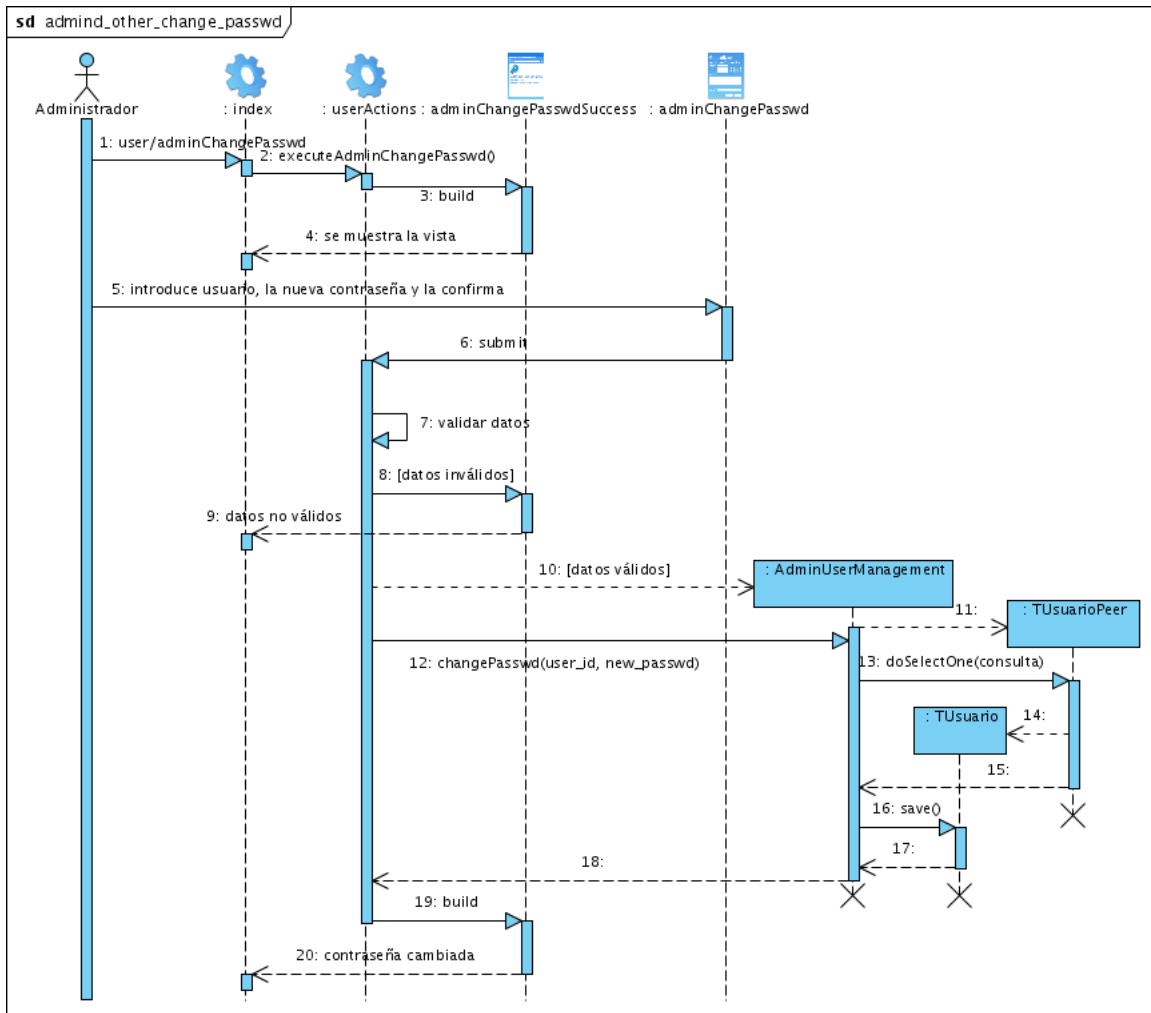


Figura E.2: Diagrama de Interacción - Caso de Uso administrador cambiar su misma contraseña.



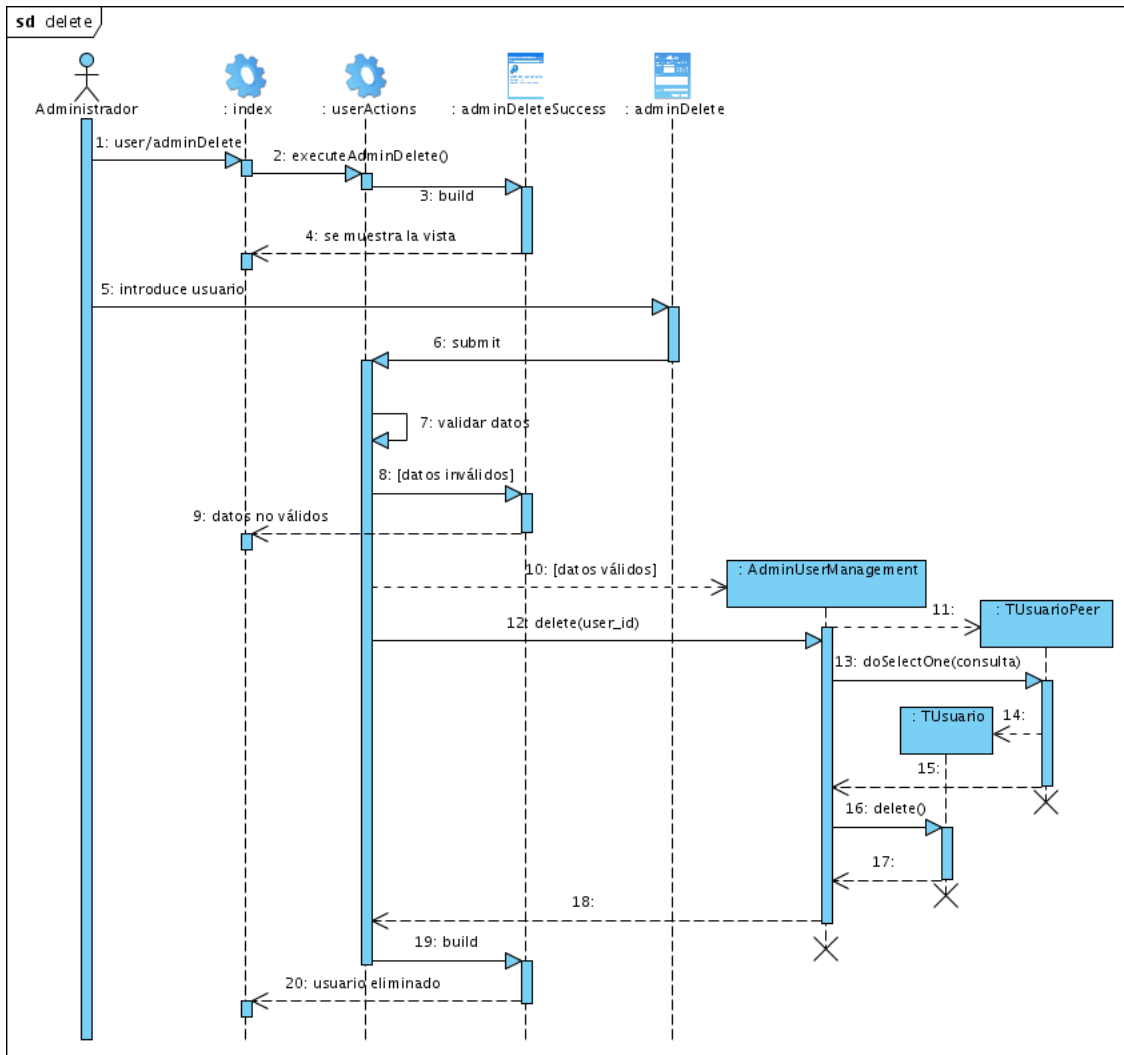


Figura E.3: Diagrama de Interacción - Caso de Uso eliminar usuario.

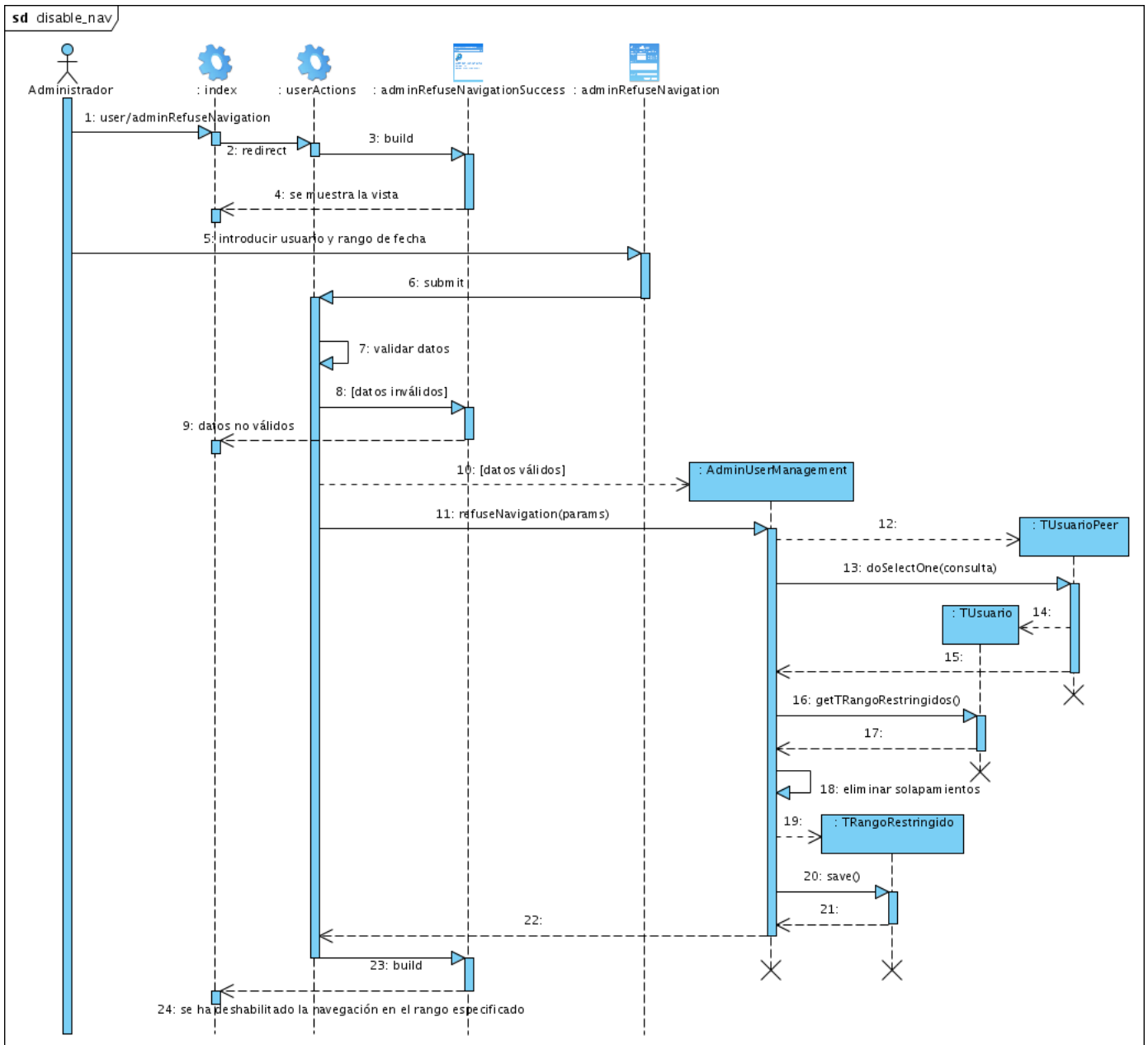


Figura E.4: Diagrama de Interacción - Caso de Uso deshabilitar navegación.

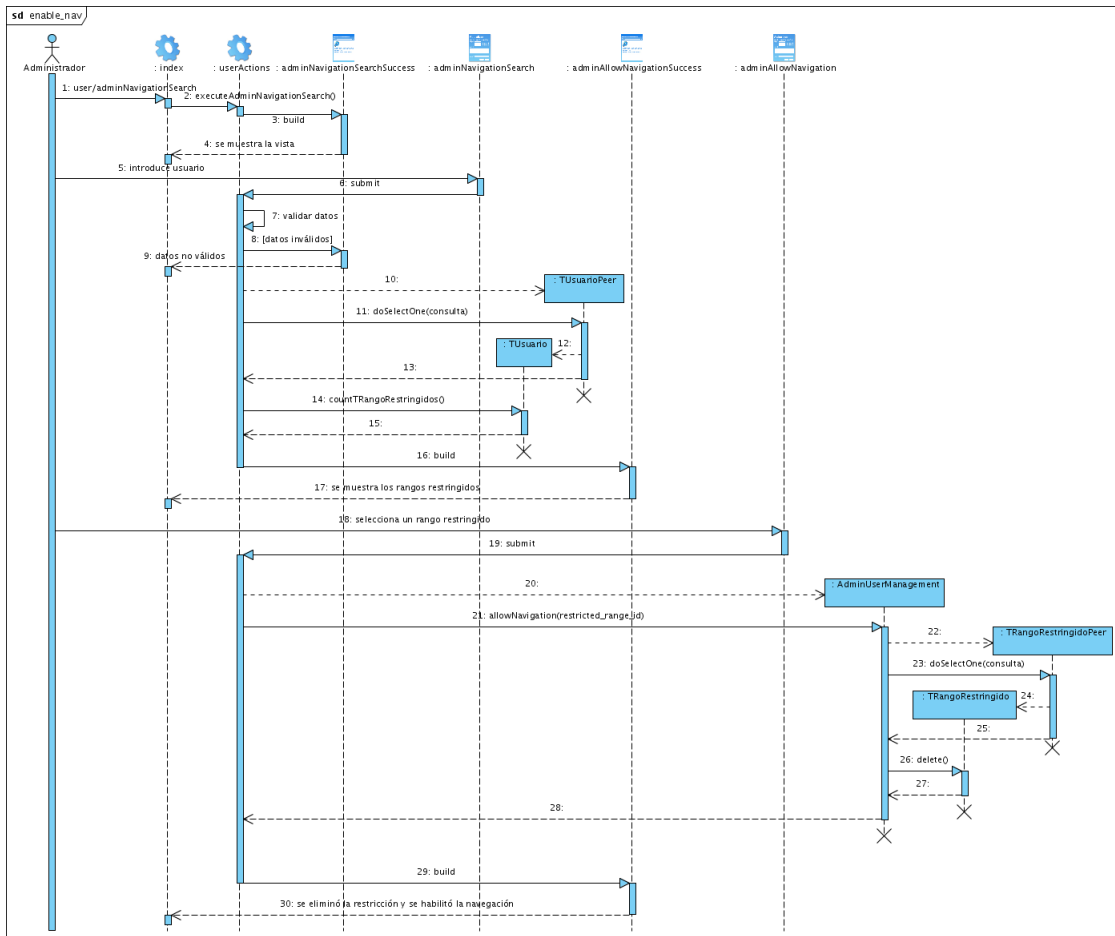


Figura E.5: Diagrama de Interacción - Caso de Uso habilitar navegación.

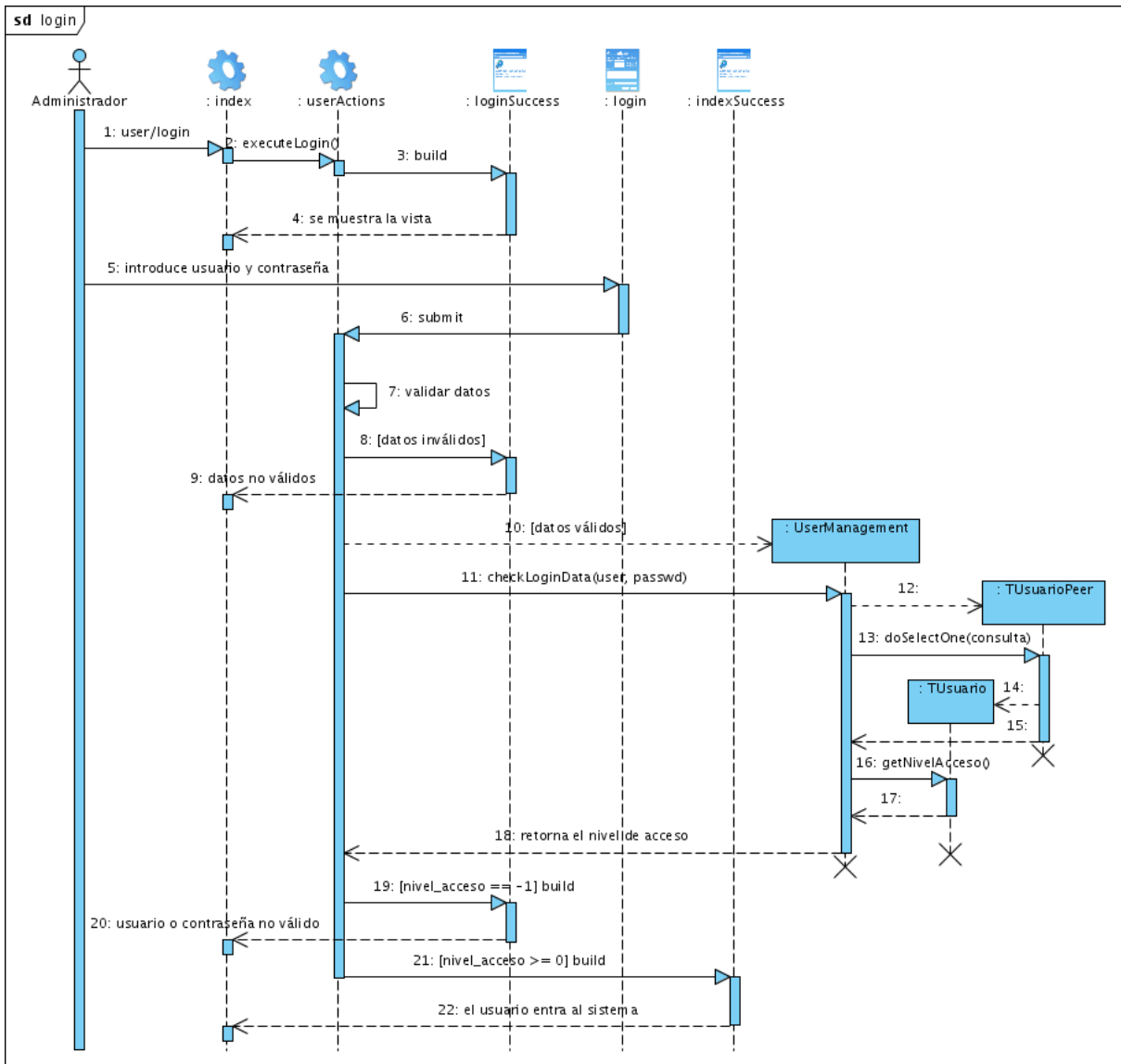


Figura E.6: Diagrama de Interacción - Caso de Uso login.

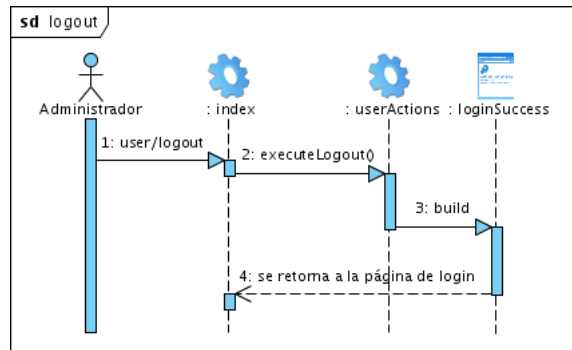


Figura E.7: Diagrama de Interacción - Caso de Uso logout.

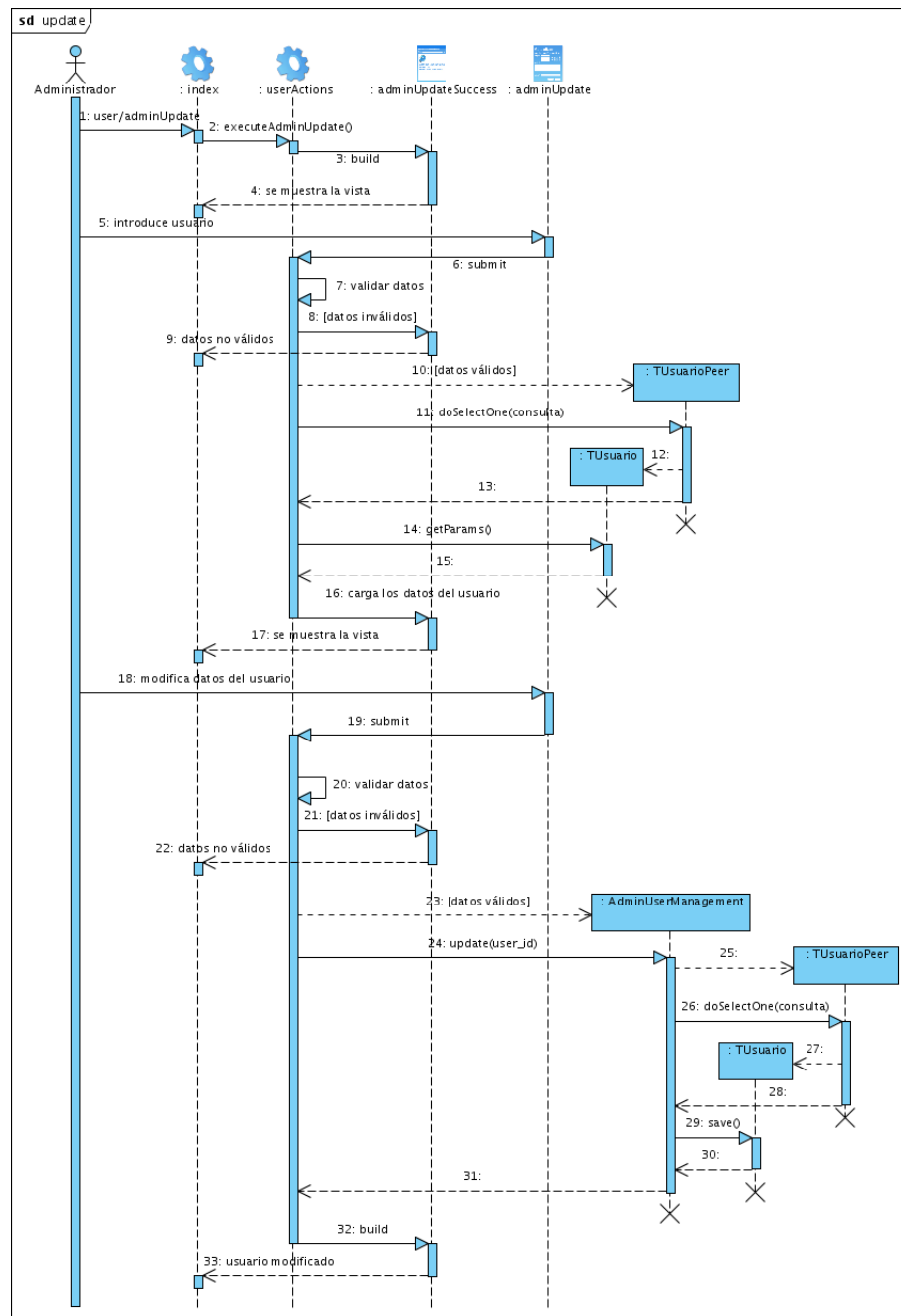


Figura E.8: Diagrama de Interacción - Caso de Uso actualizar usuario por administrador.

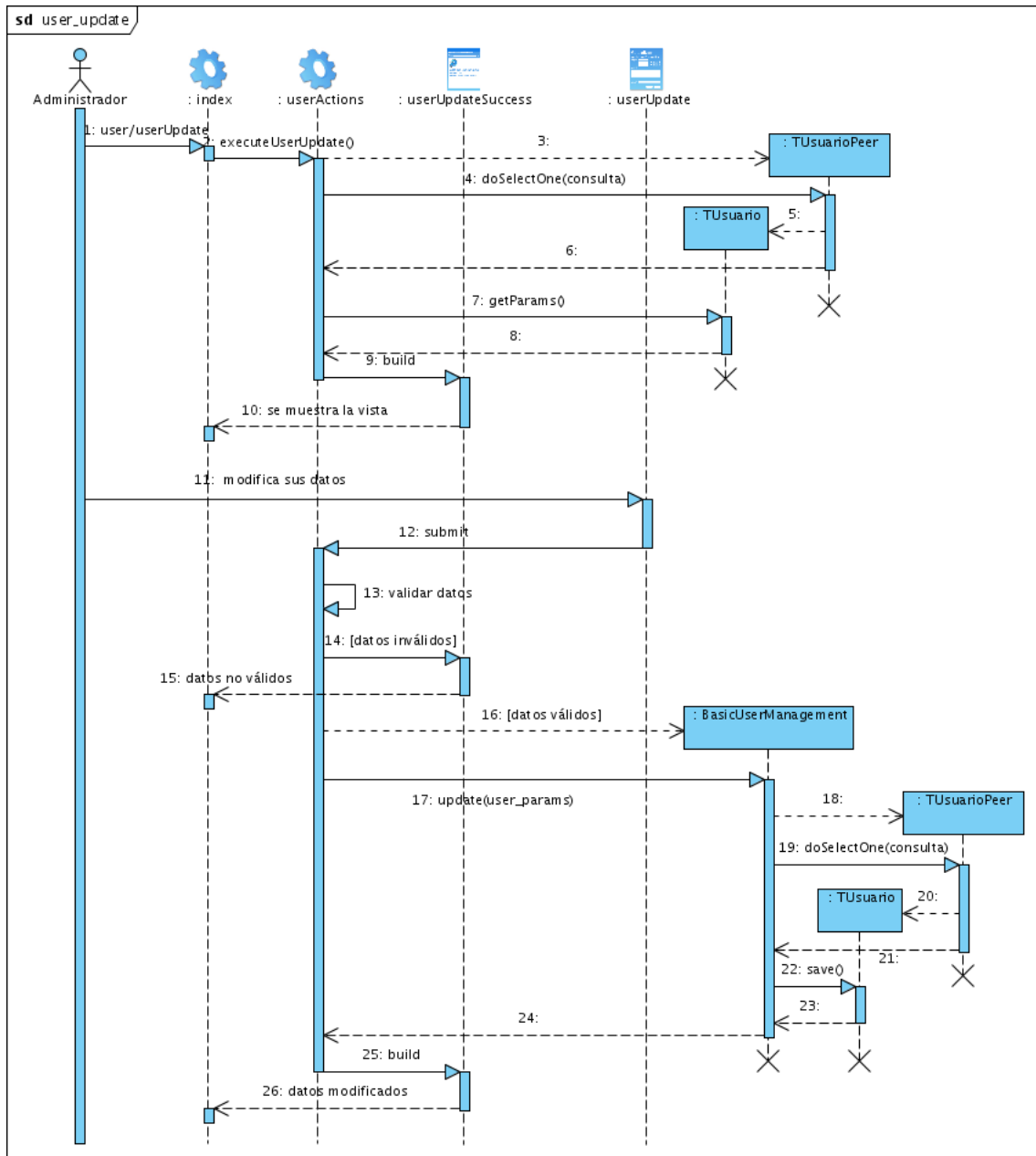


Figura E.9: Diagrama de Interacción - Caso de Uso actualizar datos por usuario básico.

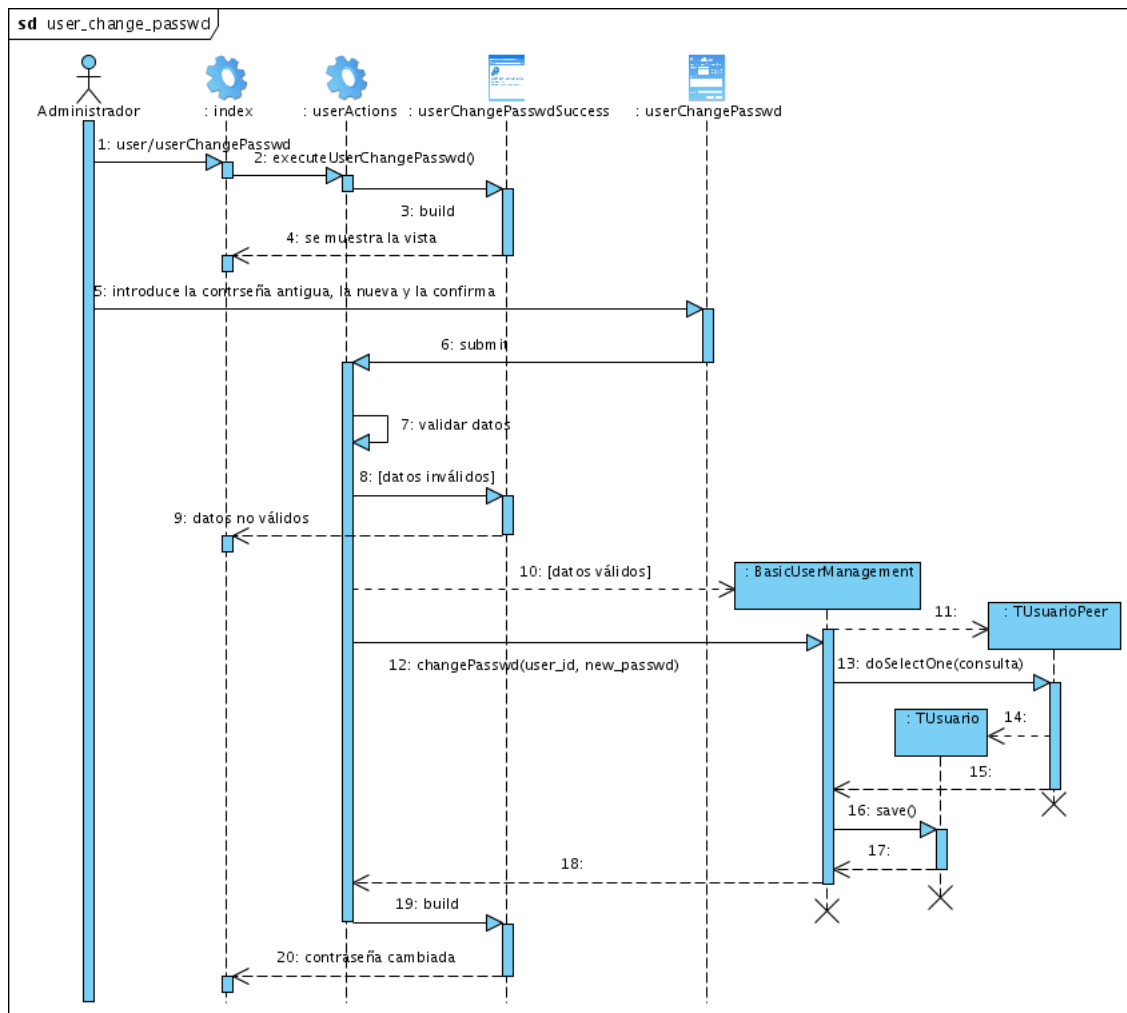


Figura E.10: Diagrama de Interacción - Caso de Uso cambiar contraseña por usuario básico.



## Módulo URL

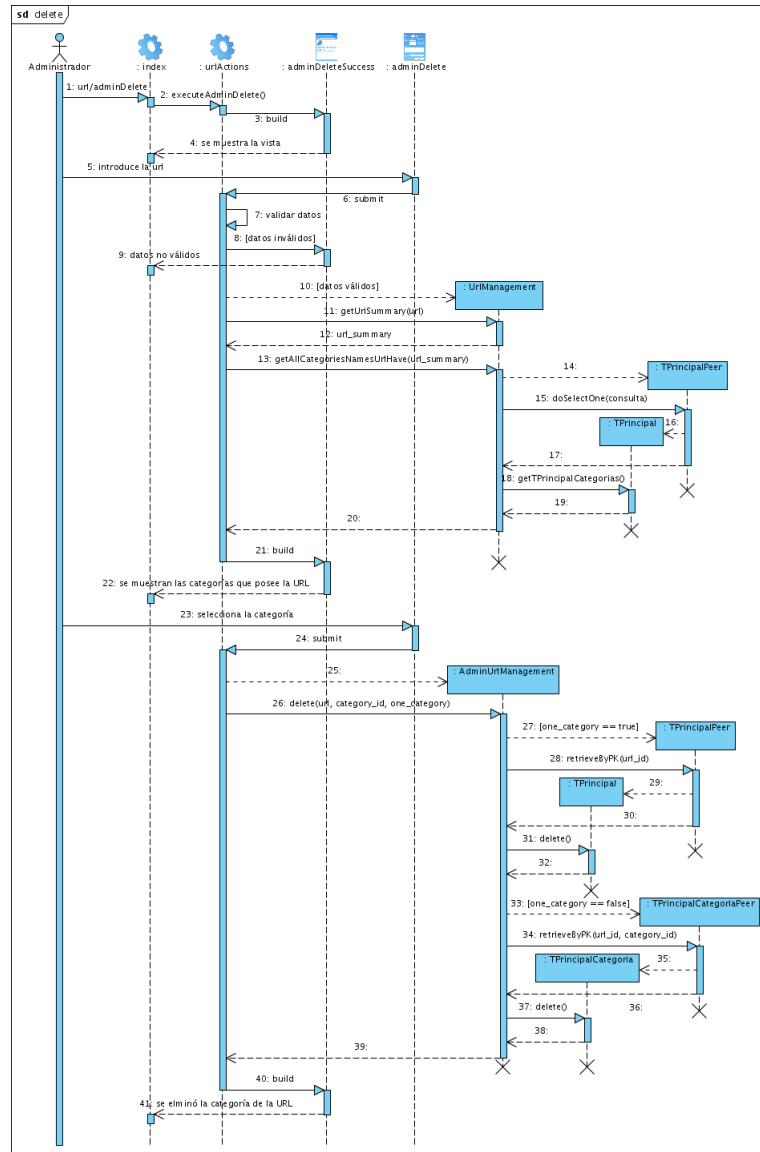


Figura E.11: Diagrama de Interacción - Caso de Uso eliminar categoría de una URL.

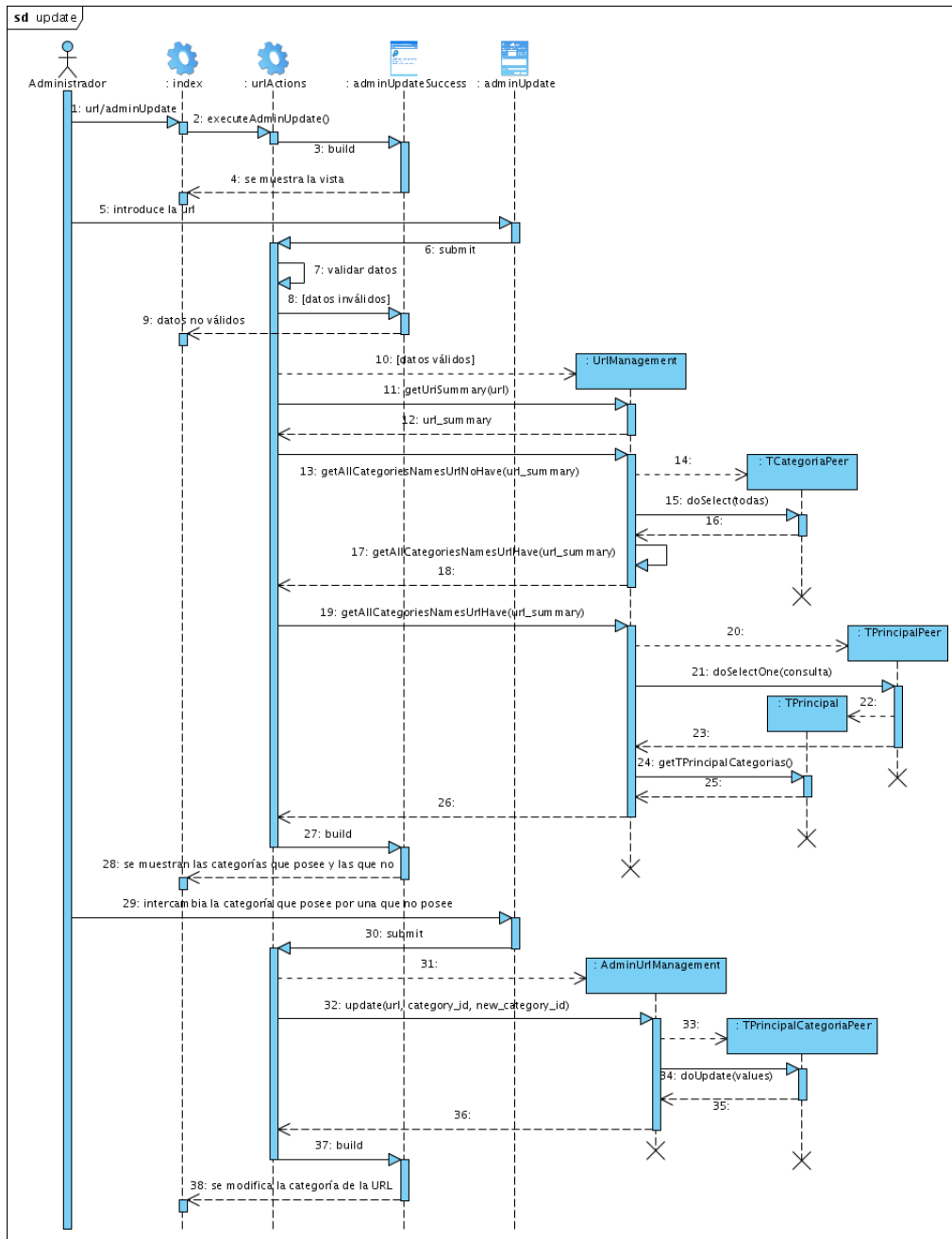


Figura E.12: Diagrama de Interacción - Caso de Uso actualizar las categorías de una URL.

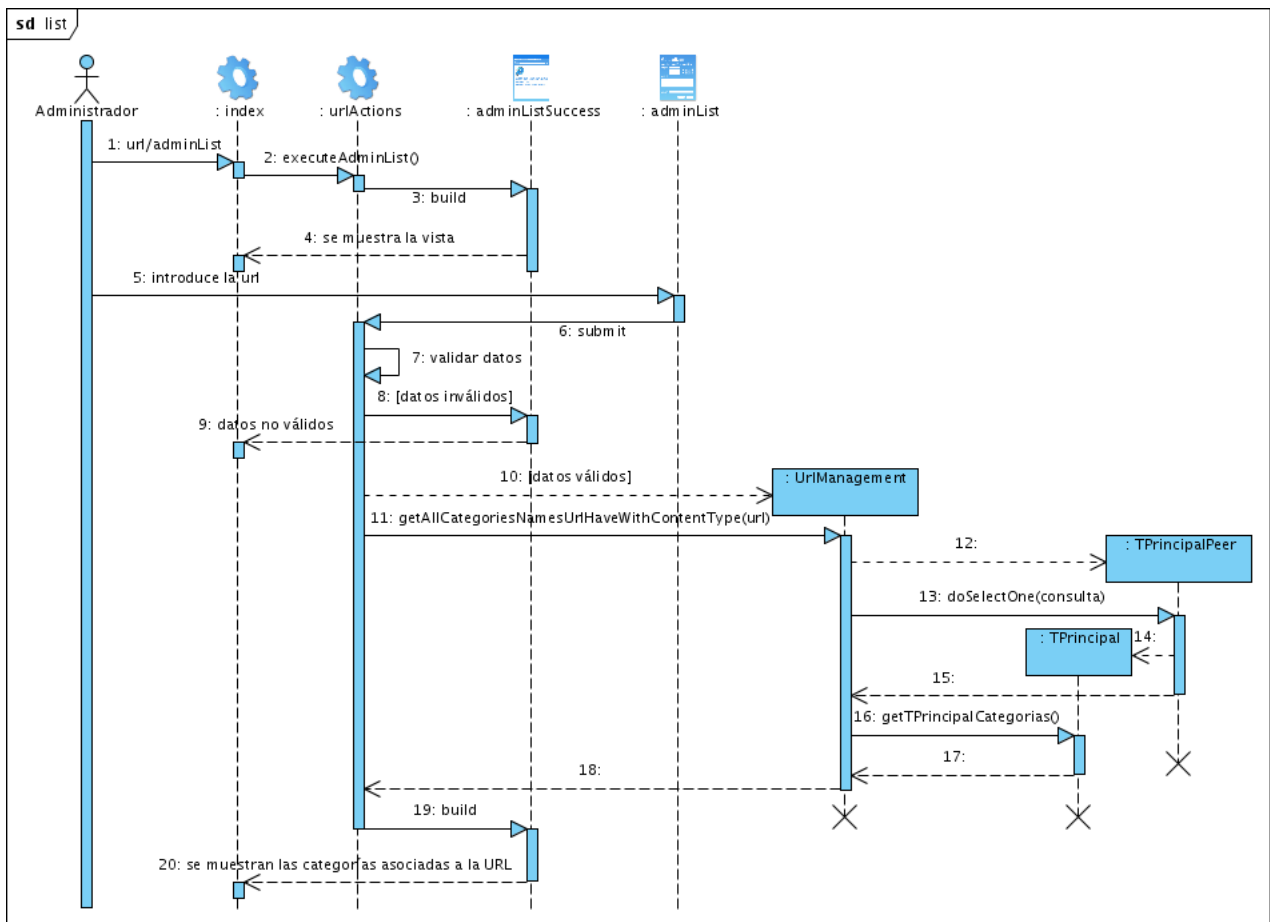


Figura E.13: Diagrama de Interacción - Caso de Uso listar las categorías de una URL.

## Módulo Categoría

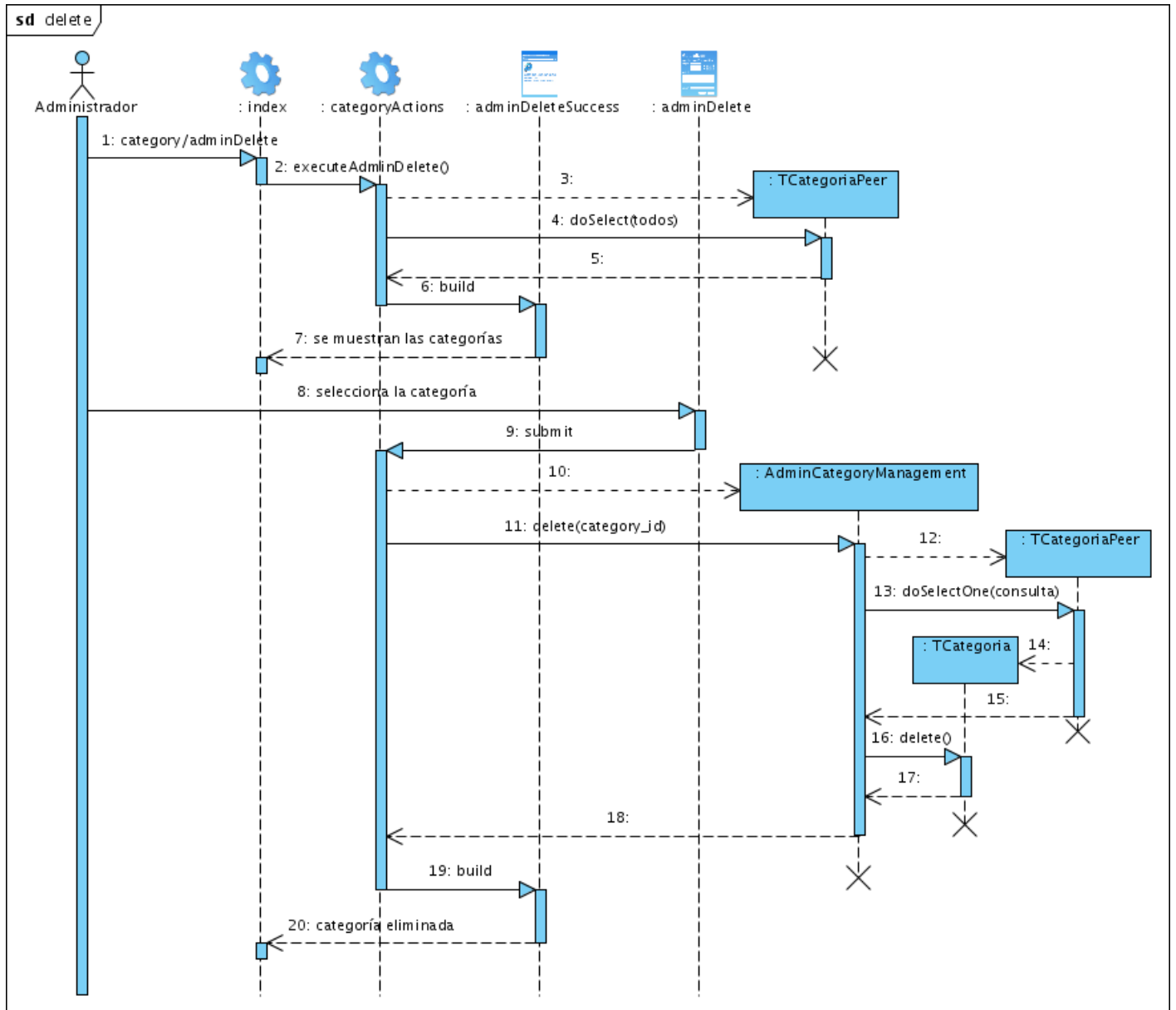


Figura E.14: Diagrama de Interacción - Caso de Uso eliminar categoría.

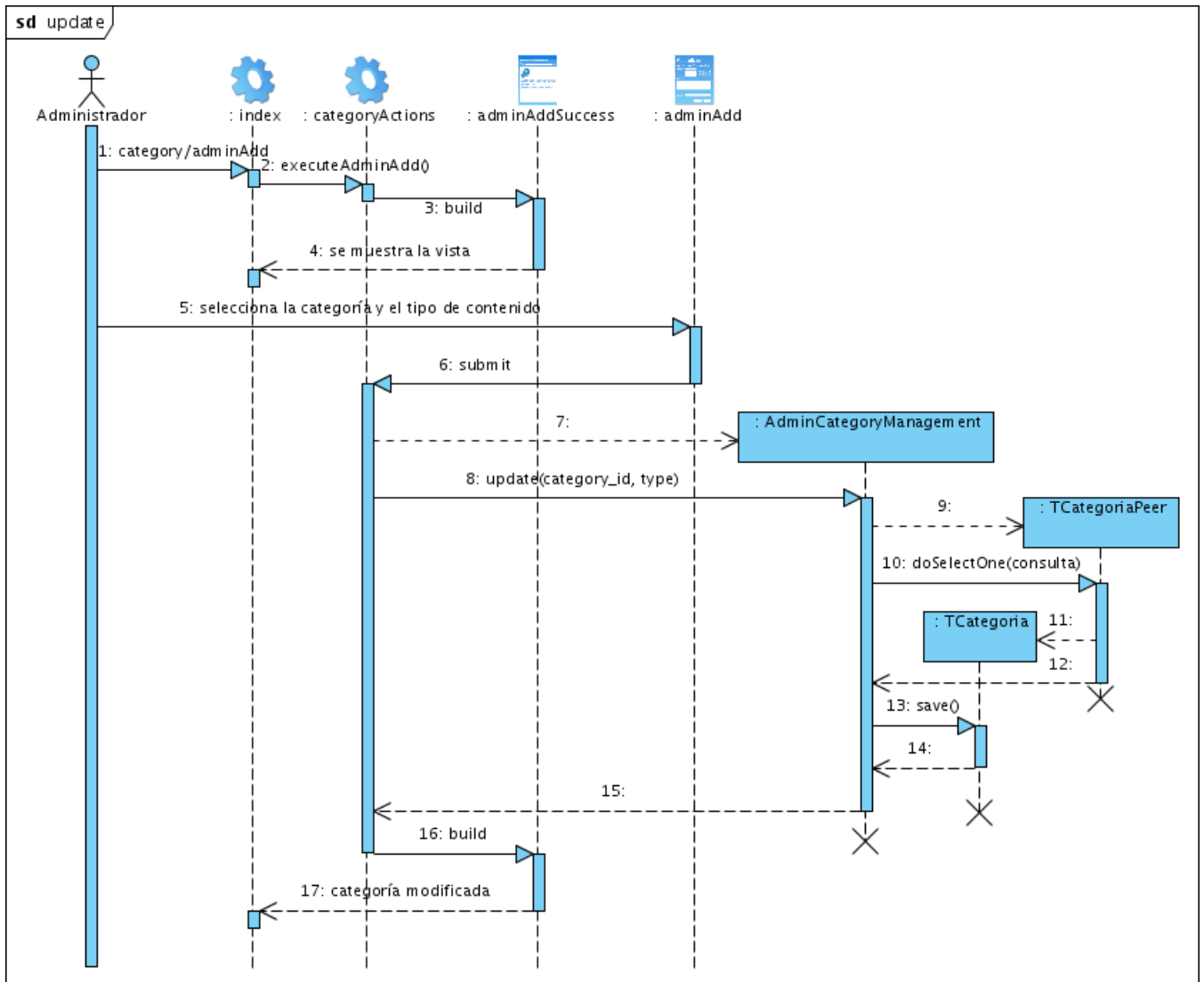


Figura E.15: Diagrama de Interacción - Caso de Uso actualizar categoría.

## Módulo Dominios de Navegación Básica

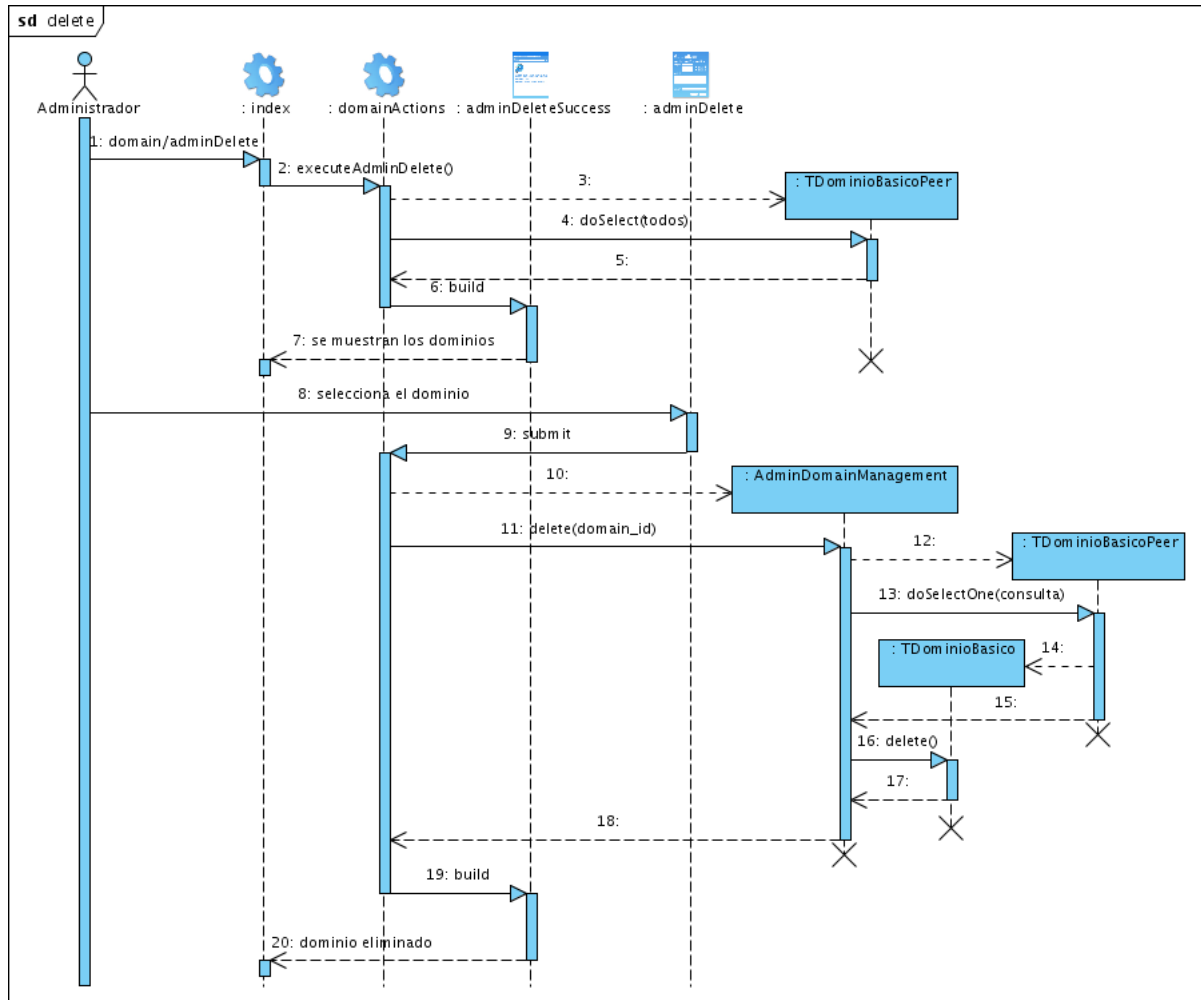


Figura E.16: Diagrama de Interacción - Caso de Uso eliminar dominio de navegación básica.

## Módulo Reporte

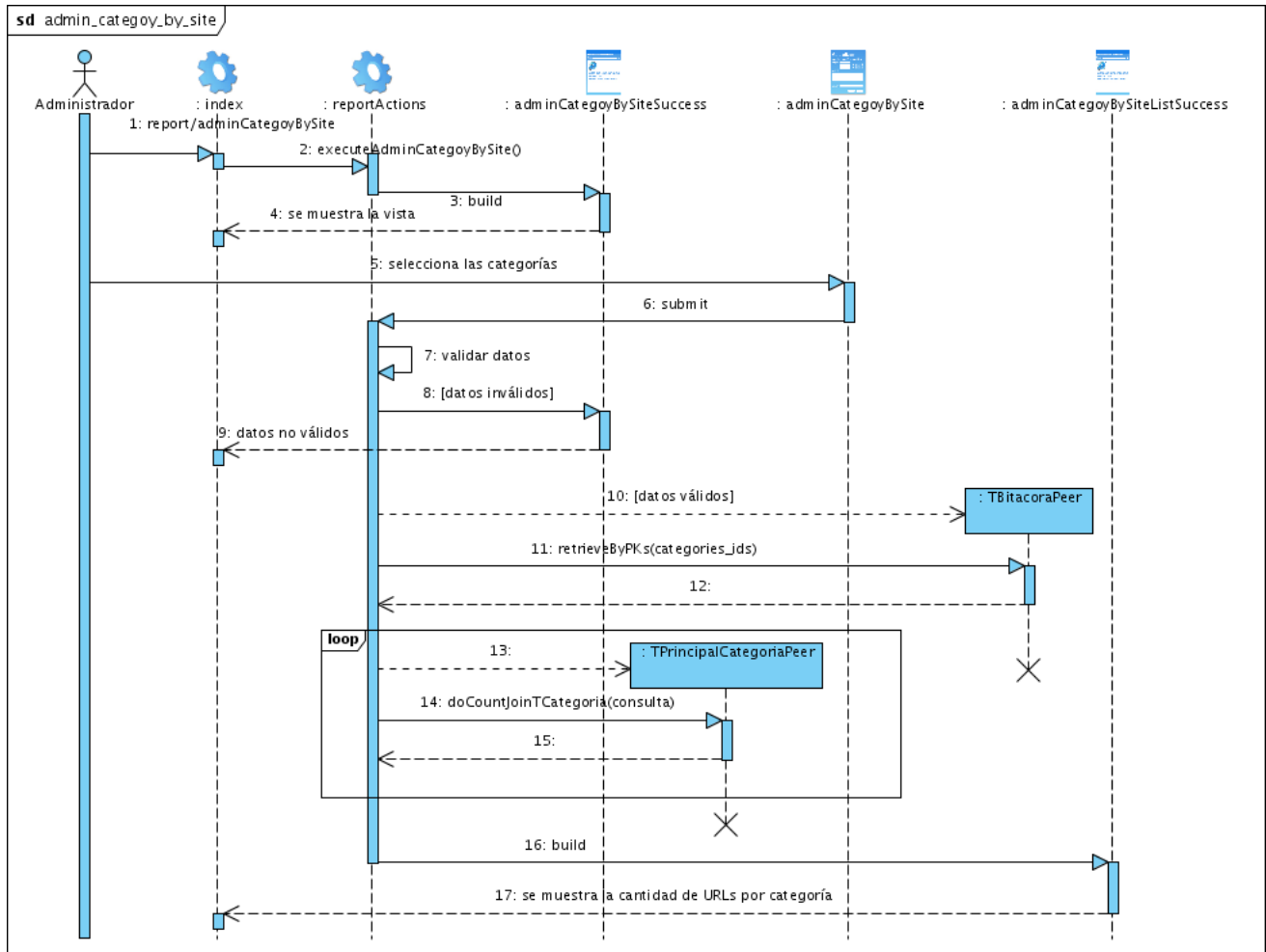


Figura E.17: Diagrama de Interacción - Caso de Uso cantidad de sitios por categoría.

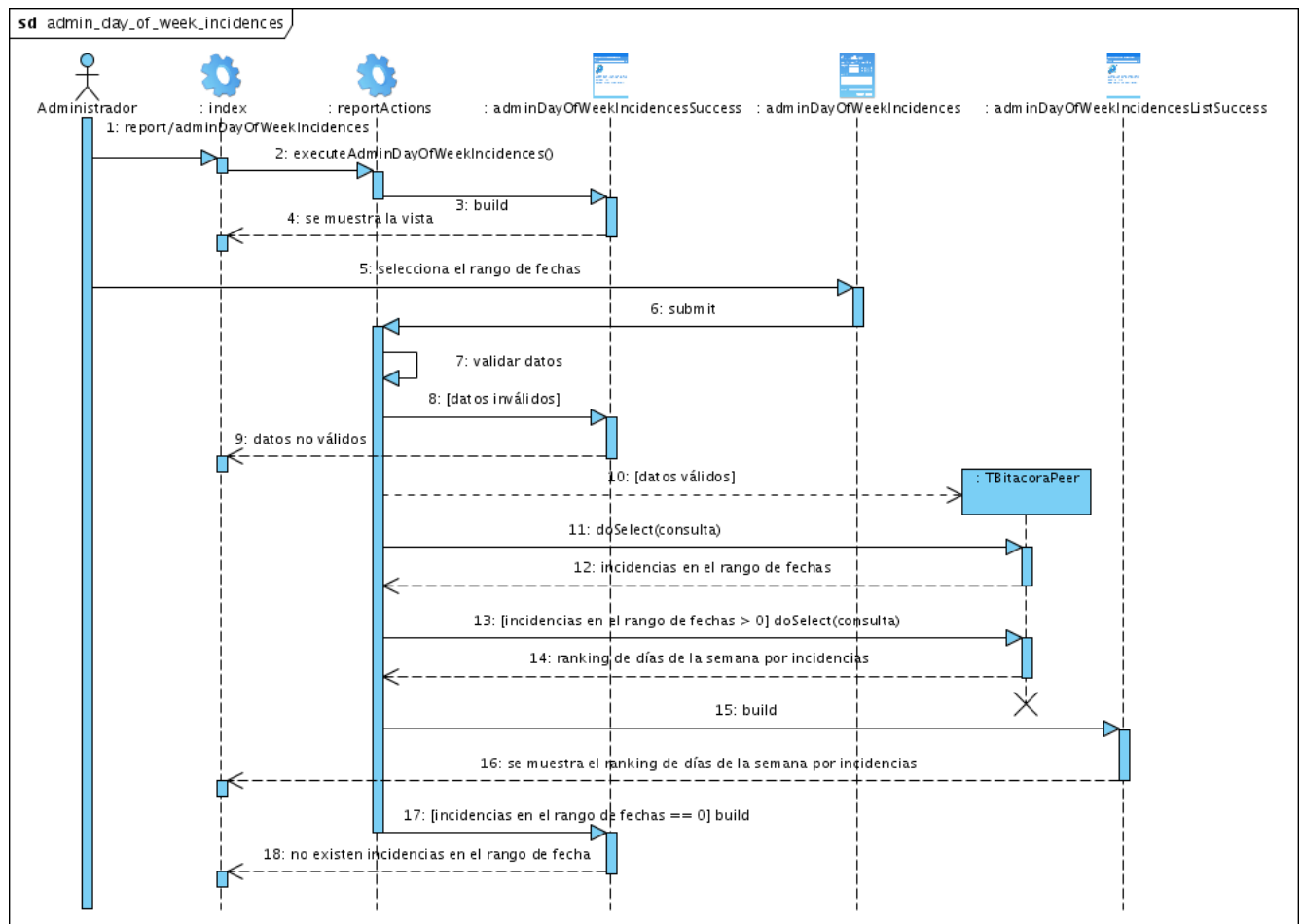


Figura E.18: Diagrama de Interacción - Caso de Uso ranking de días de la semana por incidencias.



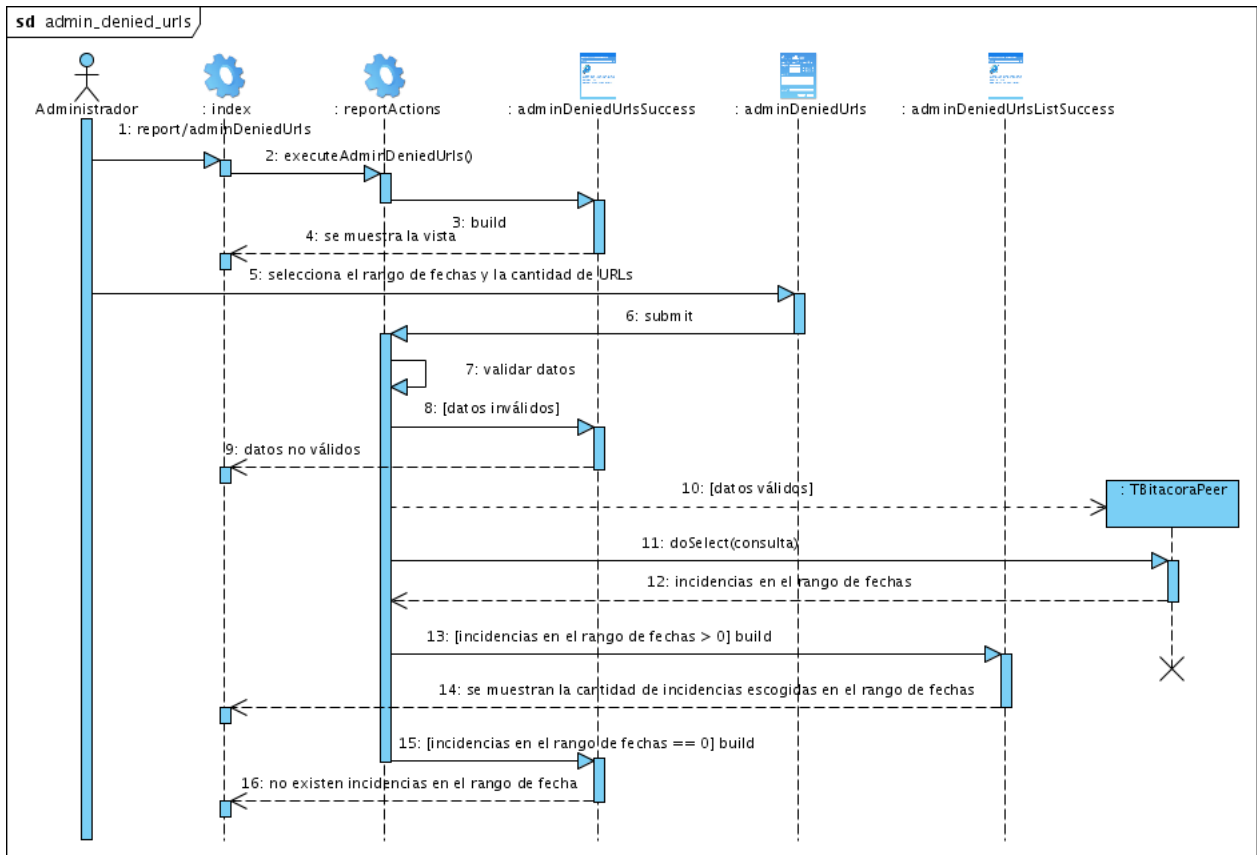


Figura E.19: Diagrama de Interacción - Caso de Uso ranking de URLs denegadas.

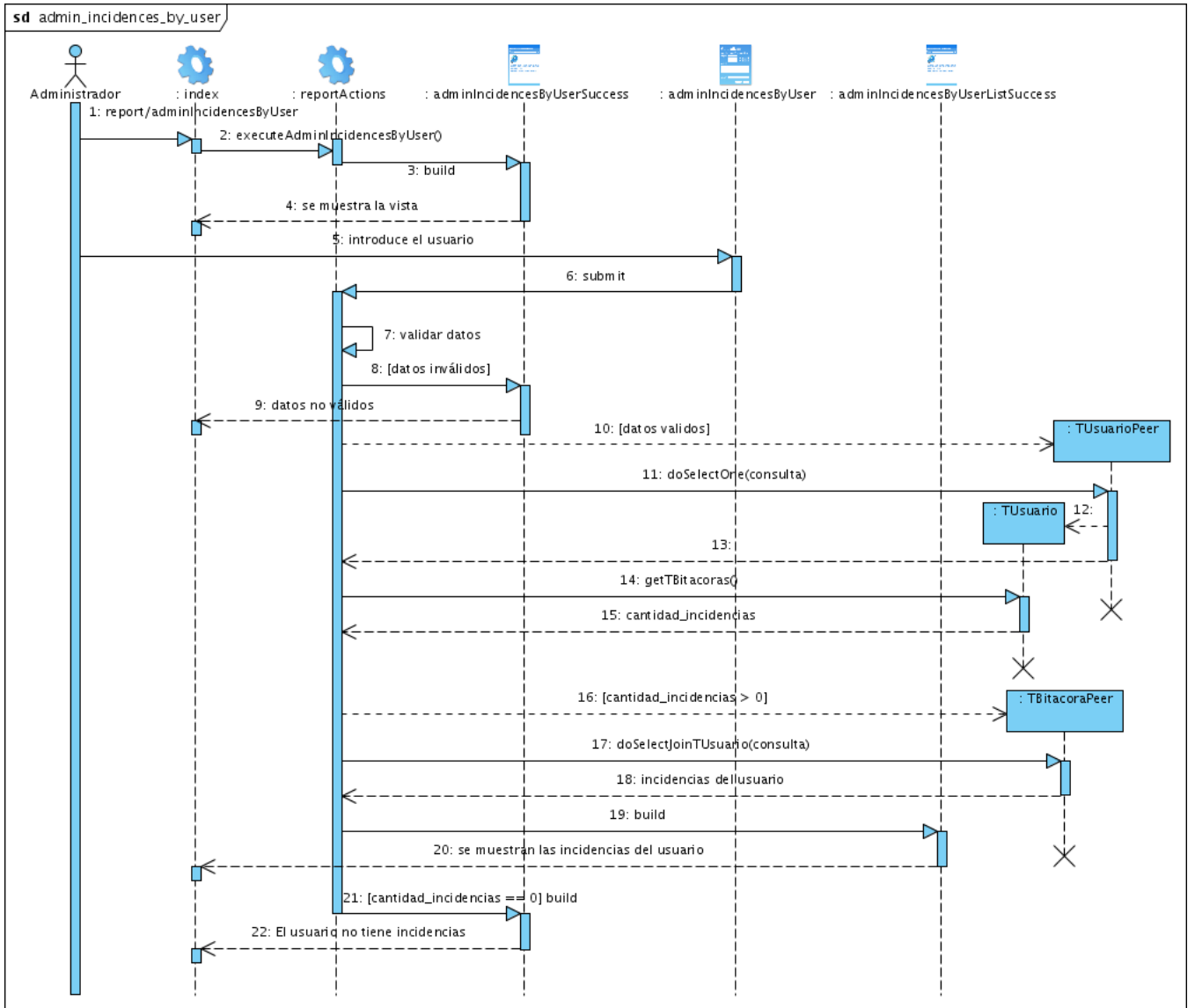


Figura E.20: Diagrama de Interacción - Caso de Uso incidencia de usuario.

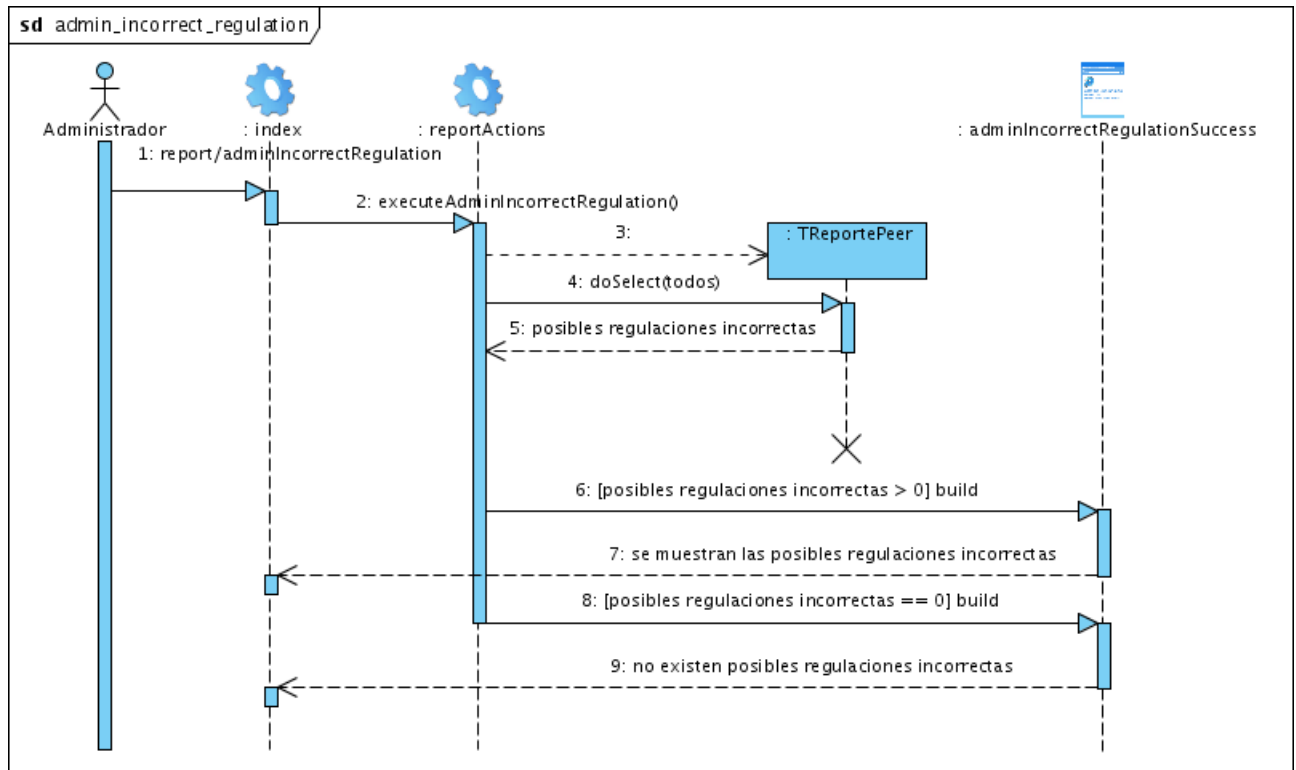


Figura E.21: Diagrama de Interacción - Caso de Uso regulaciones incorrectas.

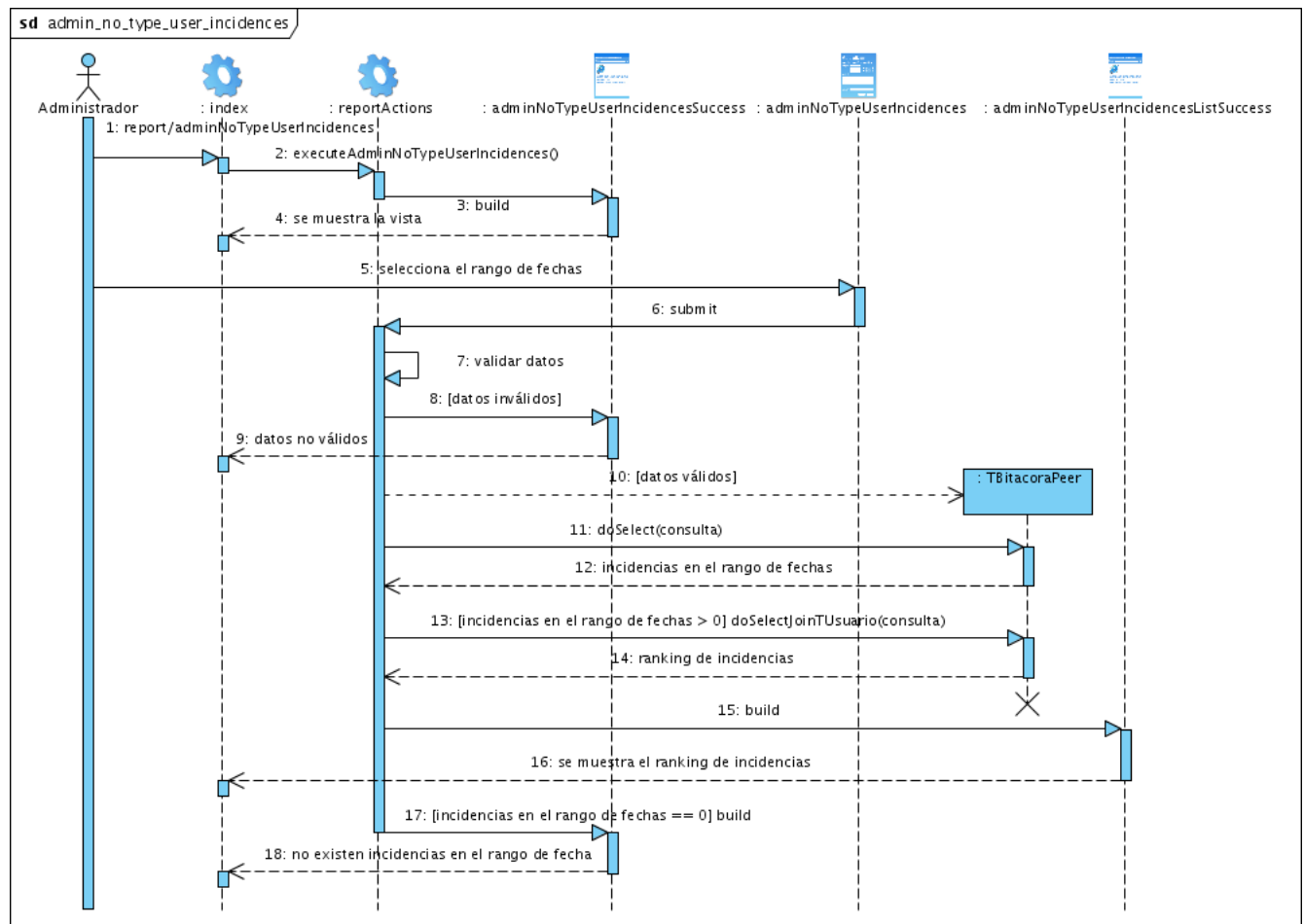


Figura E.22: Diagrama de Interacción - Caso de Uso ranking incidencias por usuario.

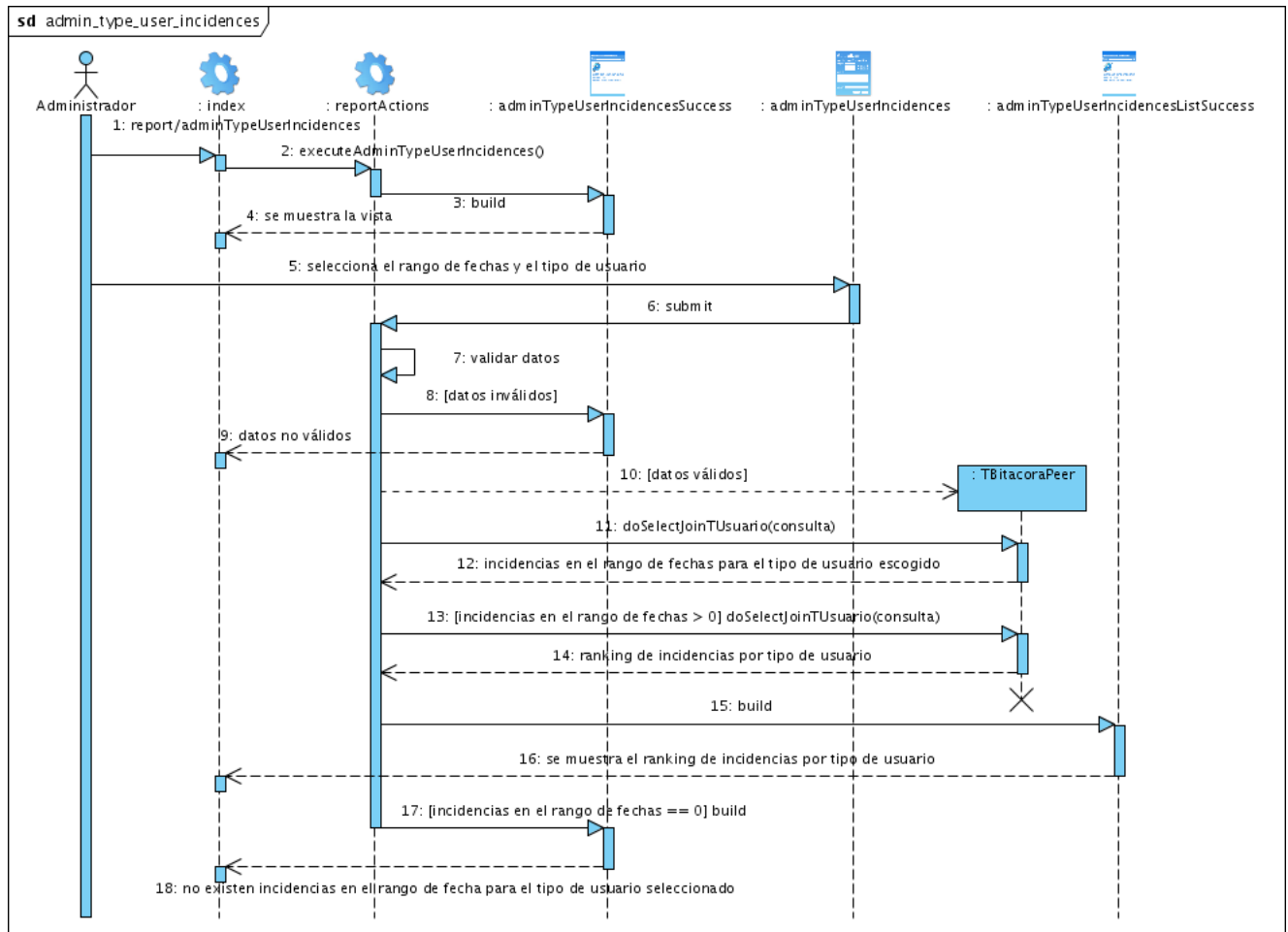


Figura E.23: Diagrama de Interacción - Caso de Uso incidencias por tipo usuario.

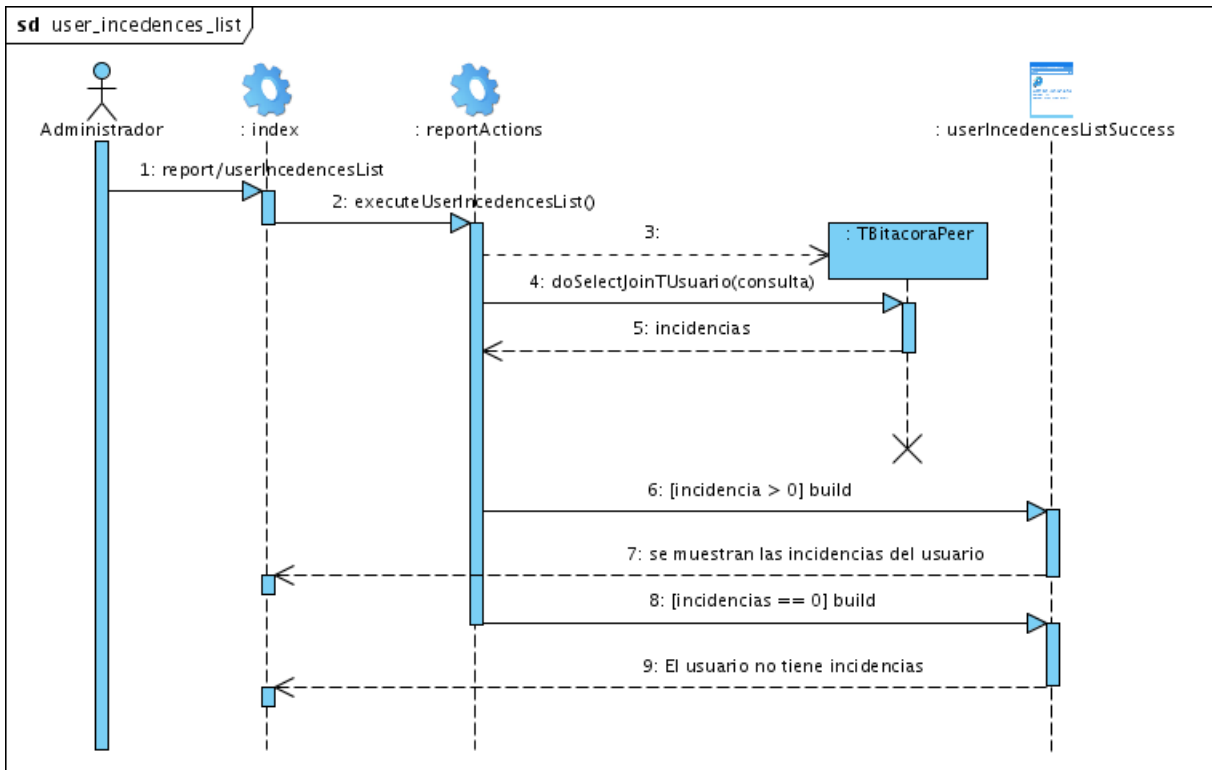


Figura E.24: Diagrama de Interacción - Caso de Uso mis incidencias.

## Anexo F

# Diagramas de Componentes

### Módulo Usuario

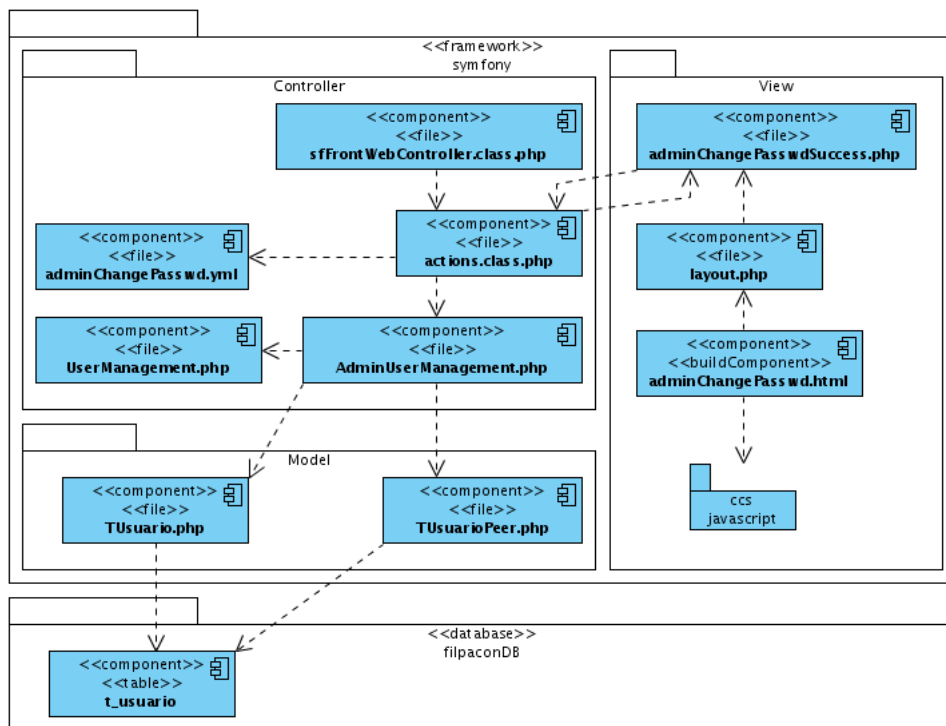


Figura F.1: Diagrama de Componente - Caso de Uso cambiar contraseña por el administrador.

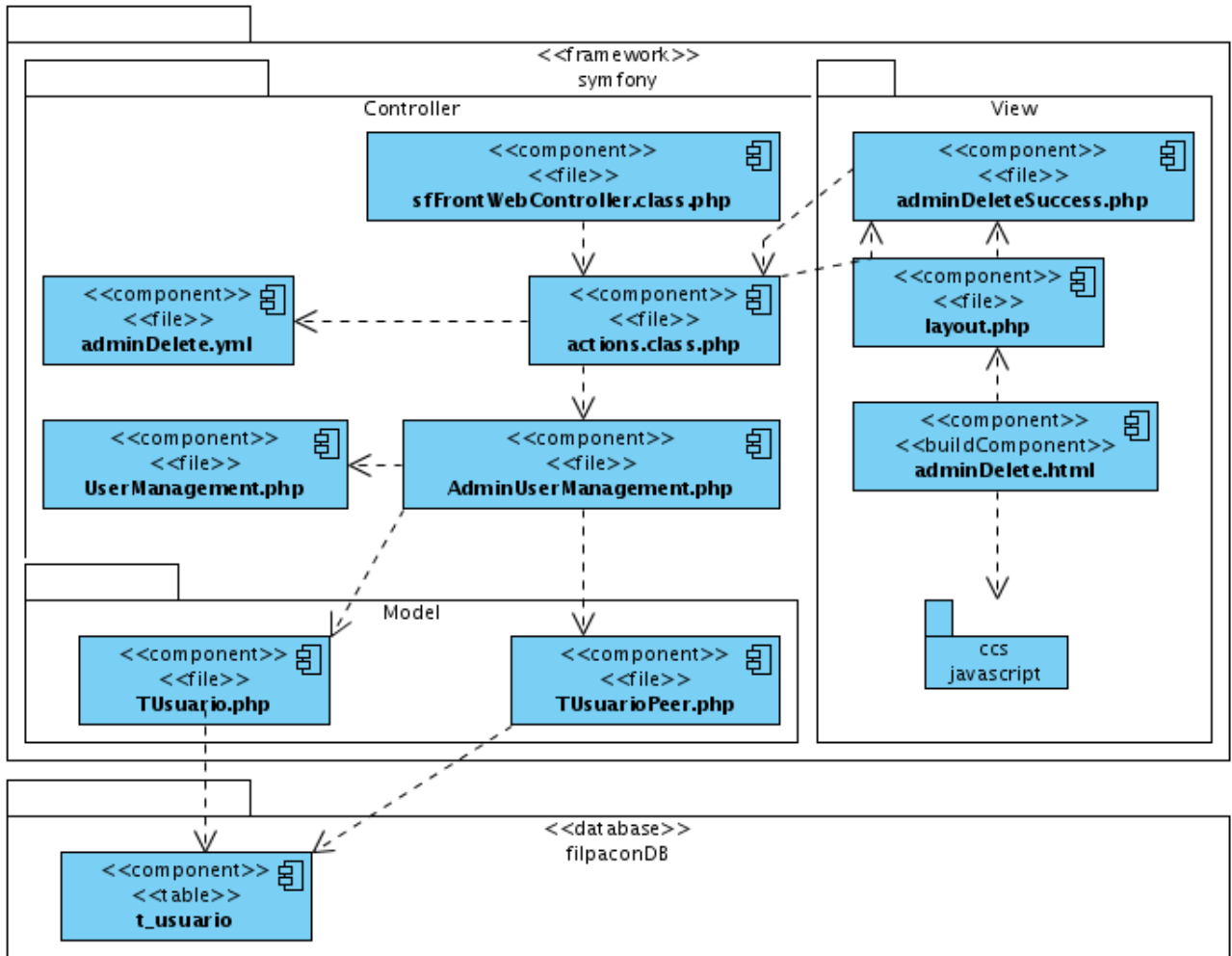


Figura F.2: Diagrama de Componente - Caso de Uso eliminar usuario.



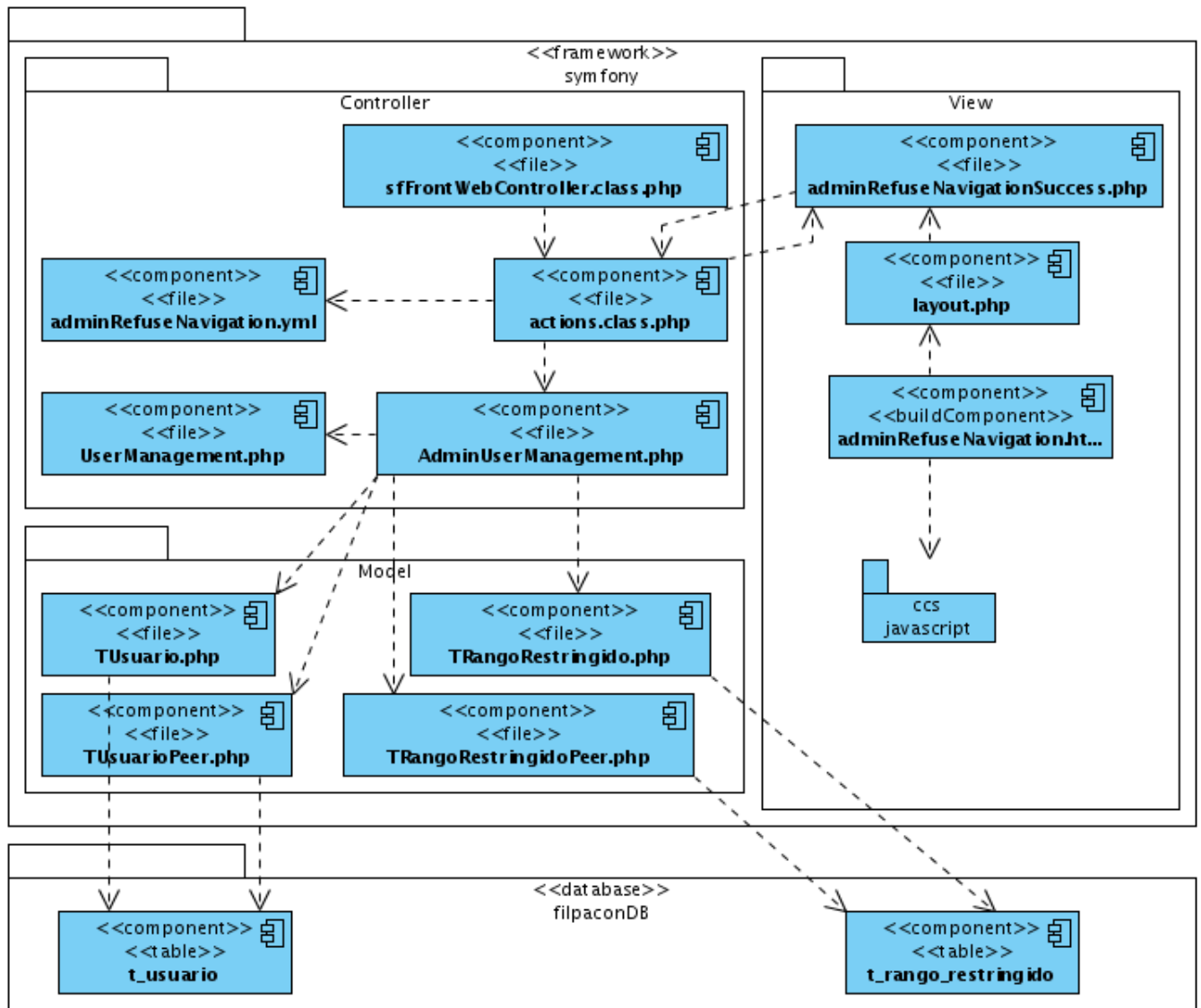


Figura F.3: Diagrama de Componente - Caso de Uso deshabilitar navegación.

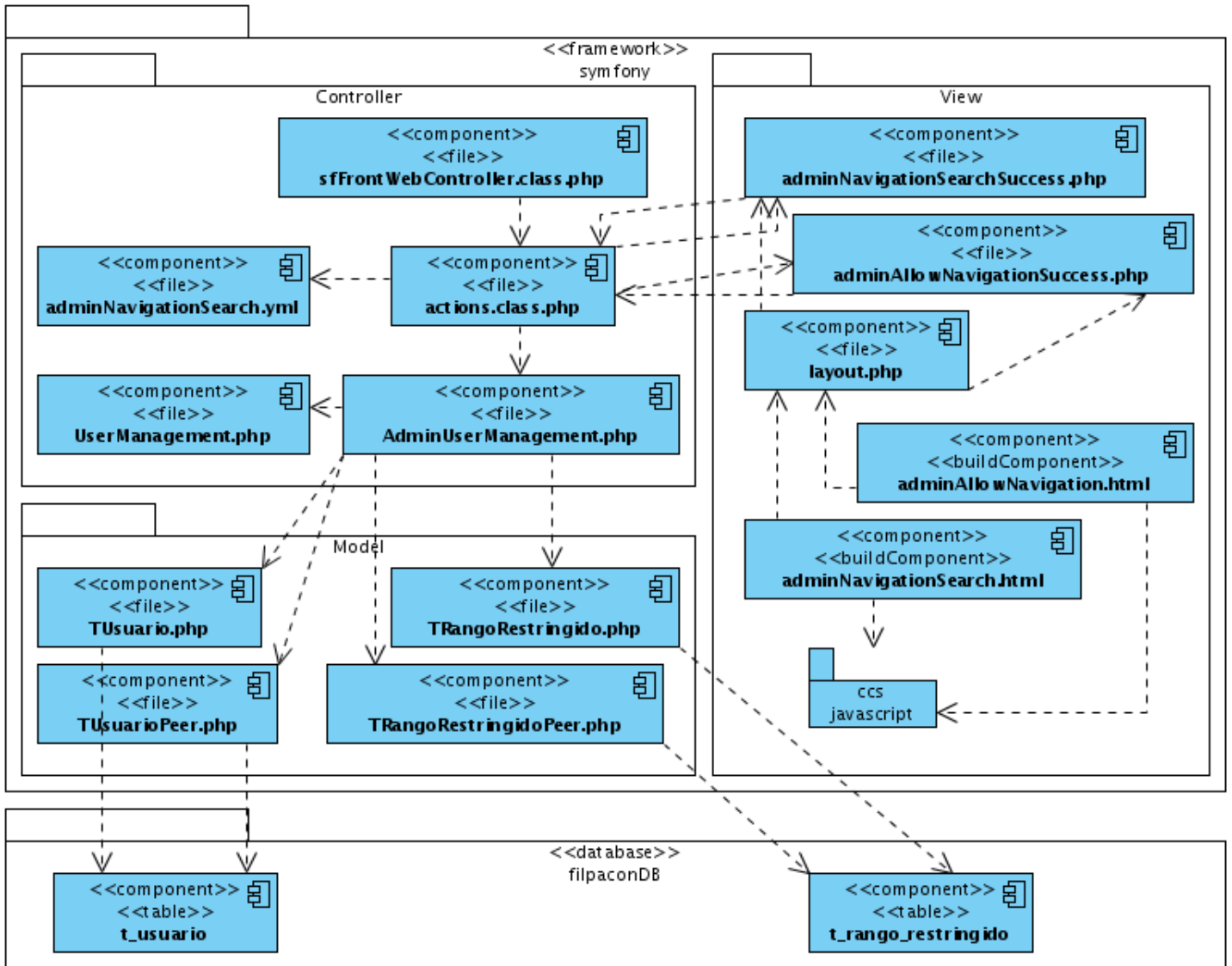


Figura F.4: Diagrama de Componente - Caso de Uso habilitar navegación.

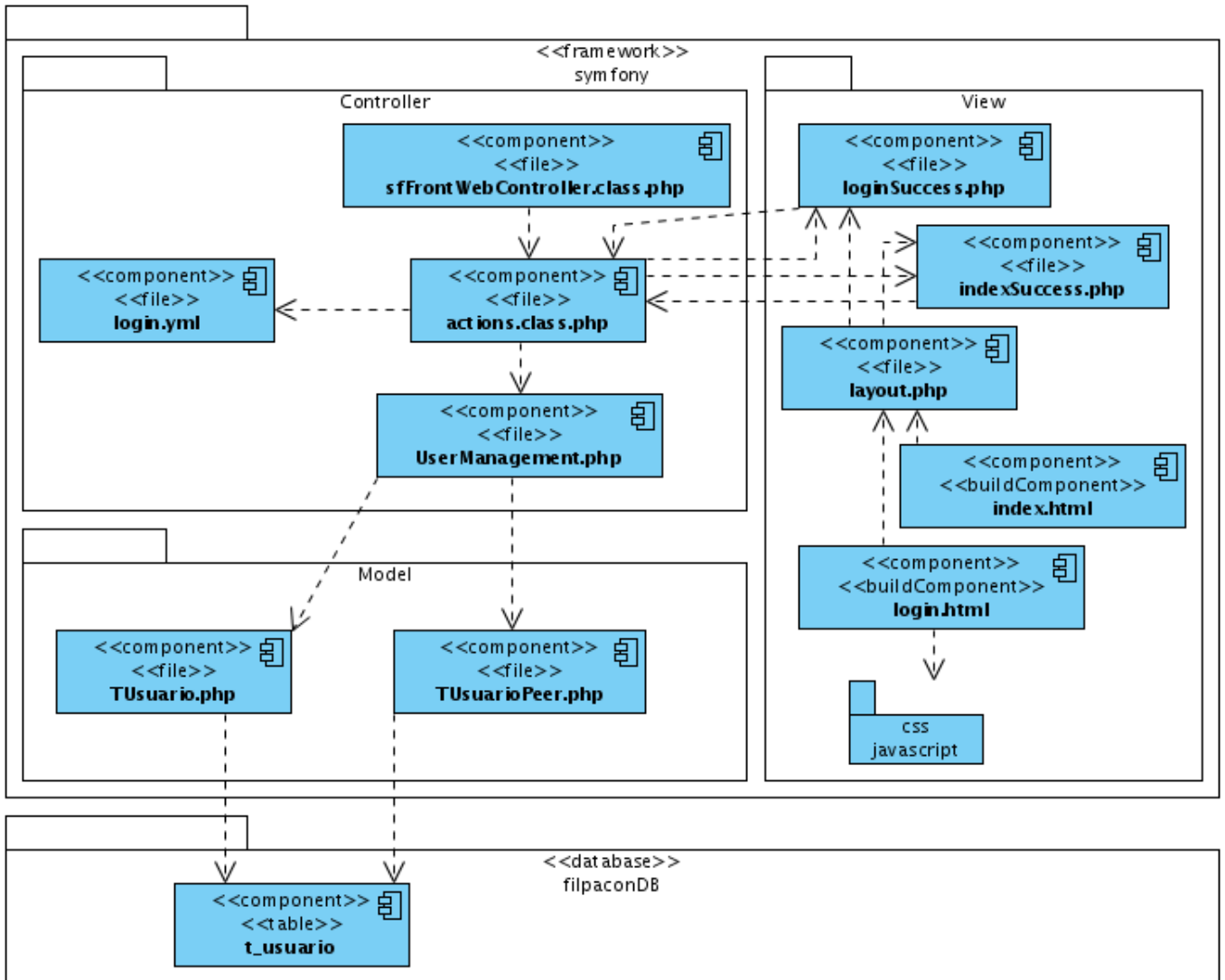


Figura F.5: Diagrama de Componente - Caso de Uso login.

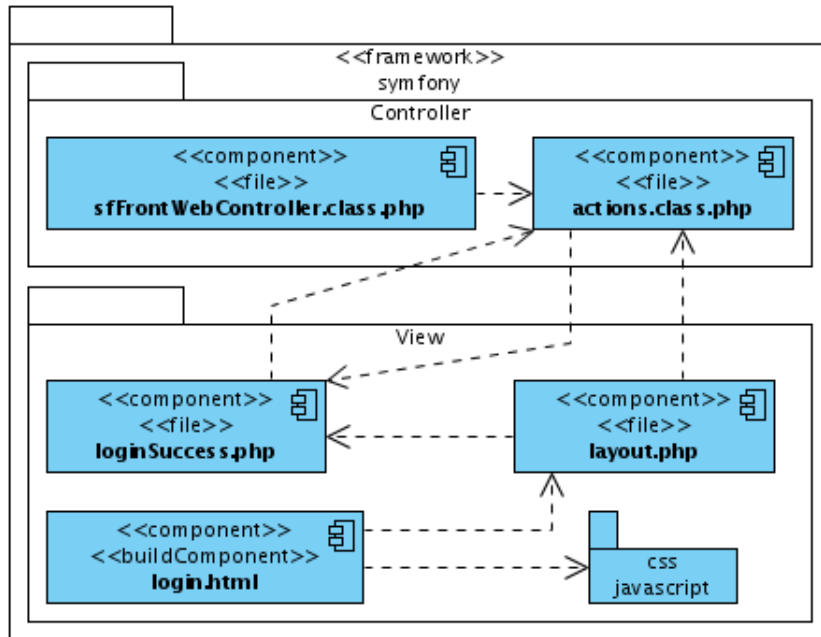


Figura F.6: Diagrama de Componente - Caso de Uso logout.

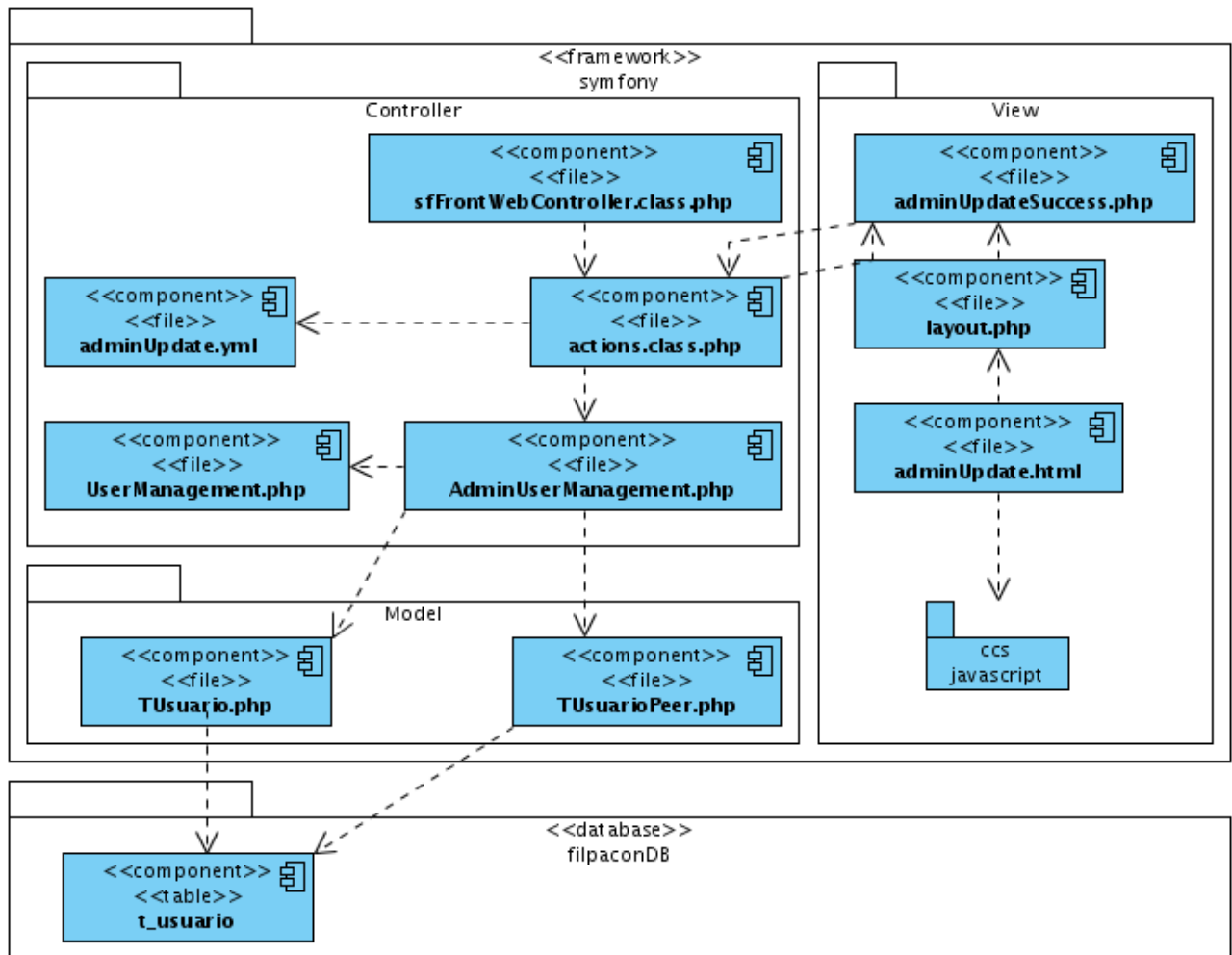


Figura F.7: Diagrama de Componente - Caso de Uso actualizar usuario por administrador.

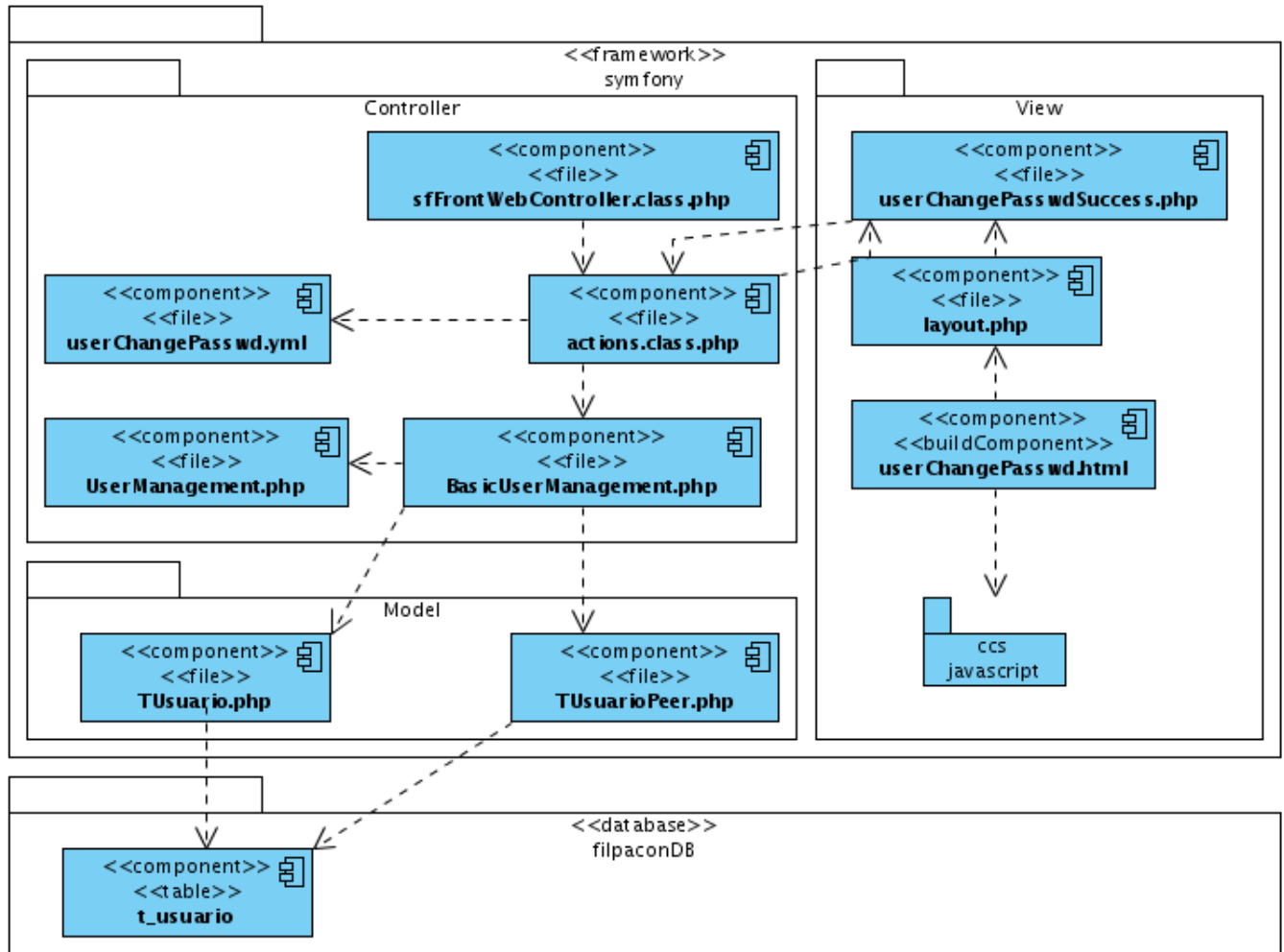


Figura F.8: Diagrama de Componente - Caso de Uso cambiar contraseña por usuario básico.

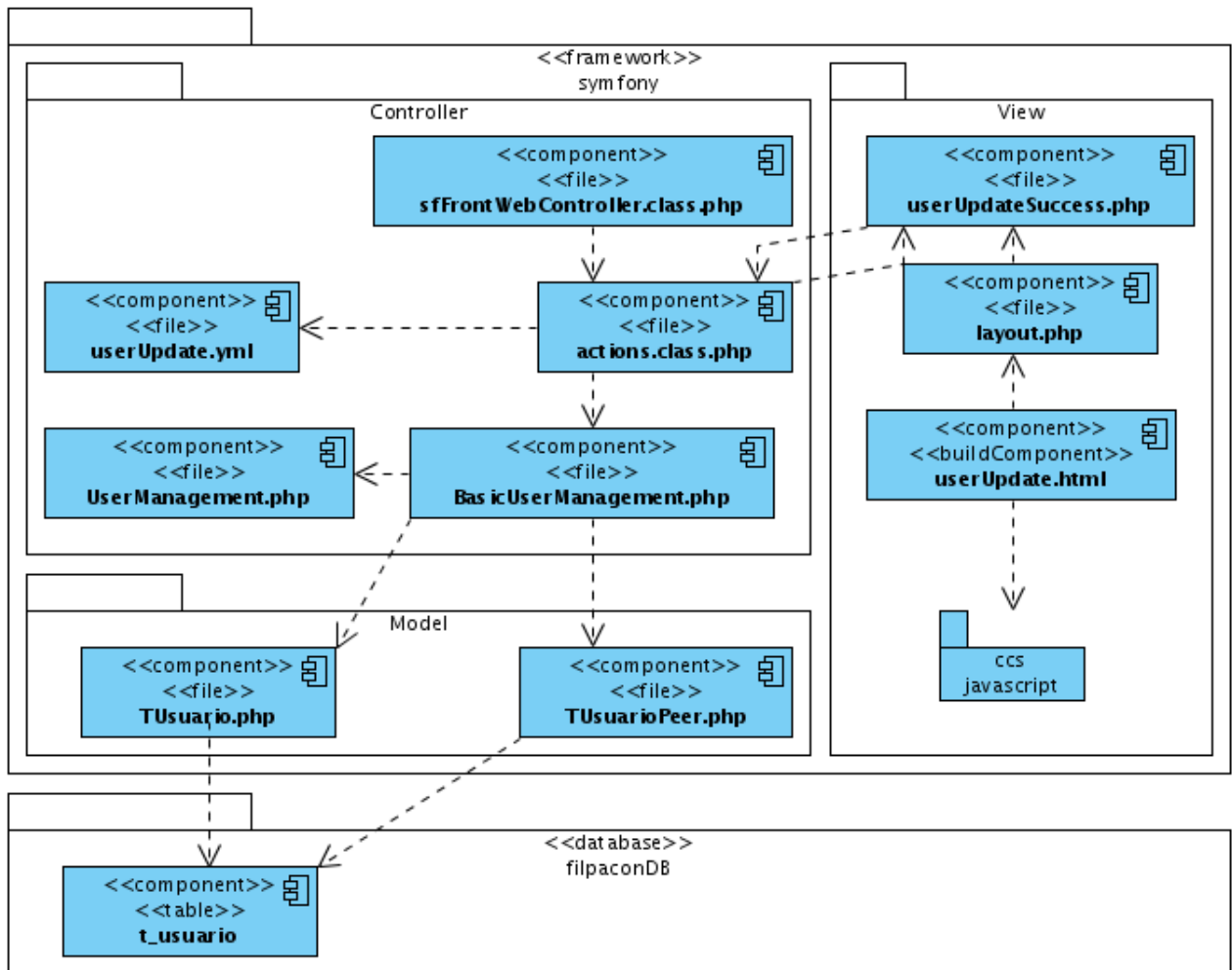


Figura F.9: Diagrama de Componente - Caso de Uso actualizar datos por usuario básico.

## Módulo URL

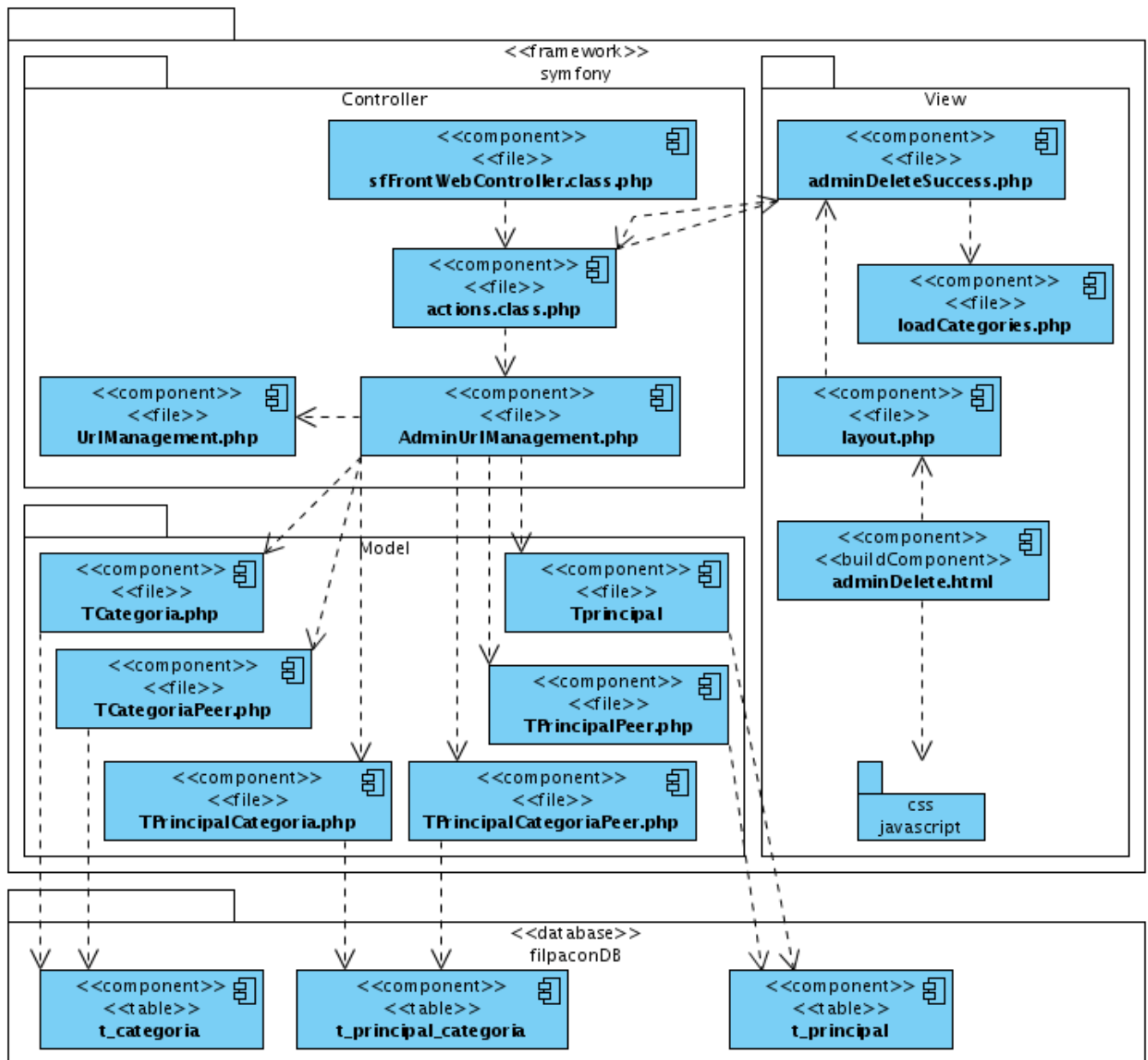


Figura F.10: Diagrama de Componente - Caso de Uso eliminar categorías a una URL.



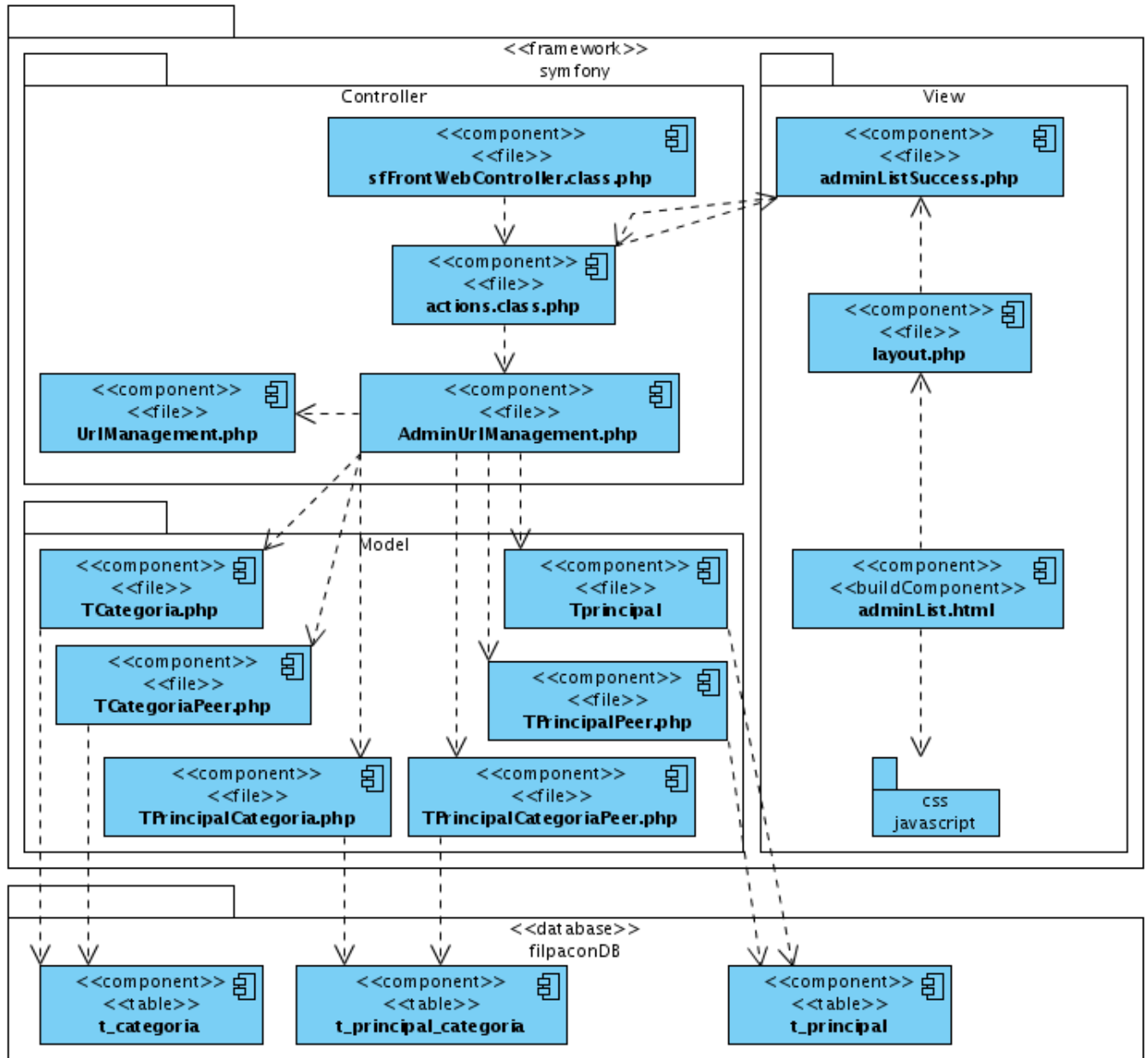


Figura F.11: Diagrama de Componente - Caso de Uso listar las categoría de una URL.

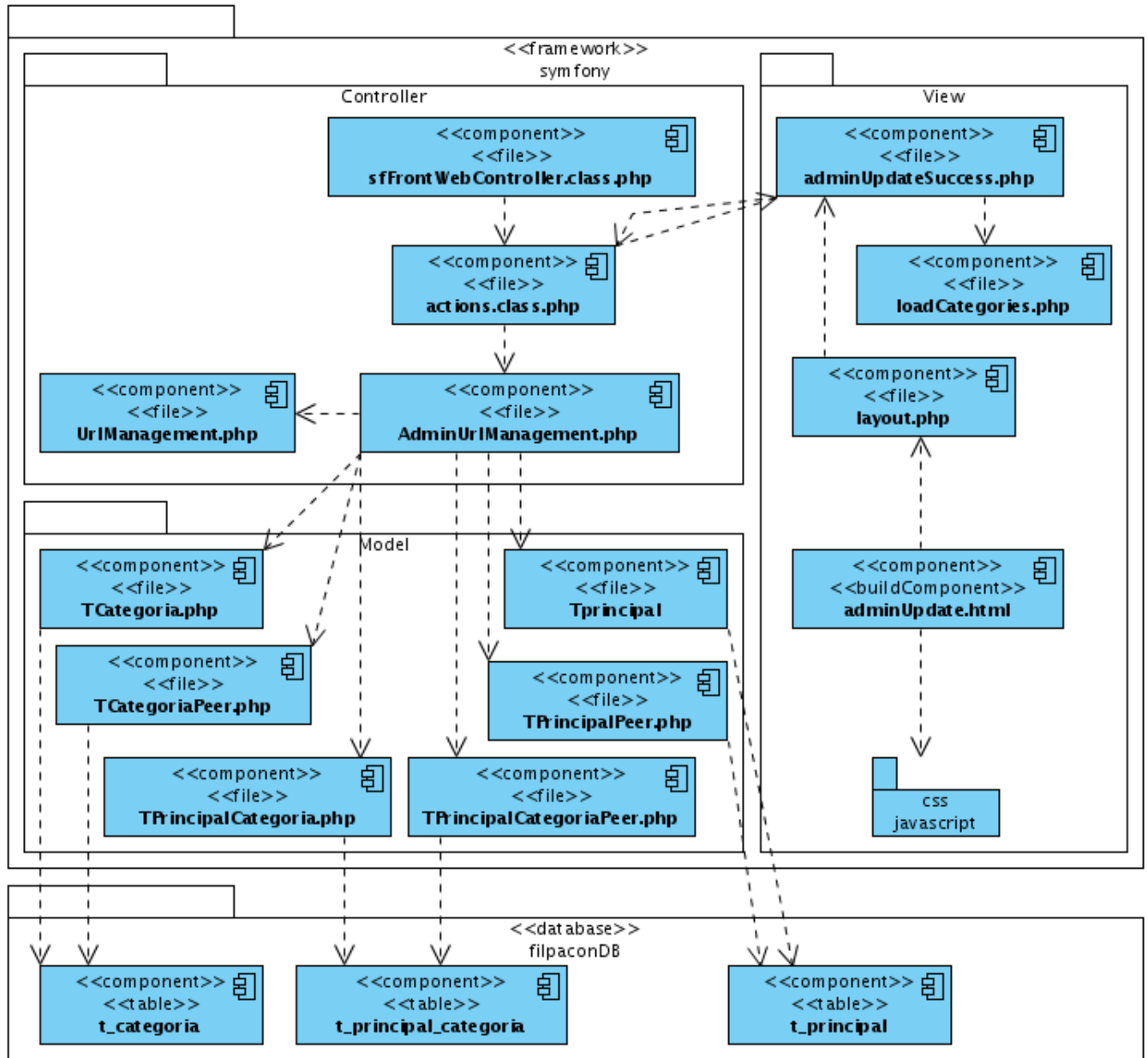


Figura F.12: Diagrama de Componente - Caso de Uso actualizar las categorías de una URL.

## Módulo Categoría

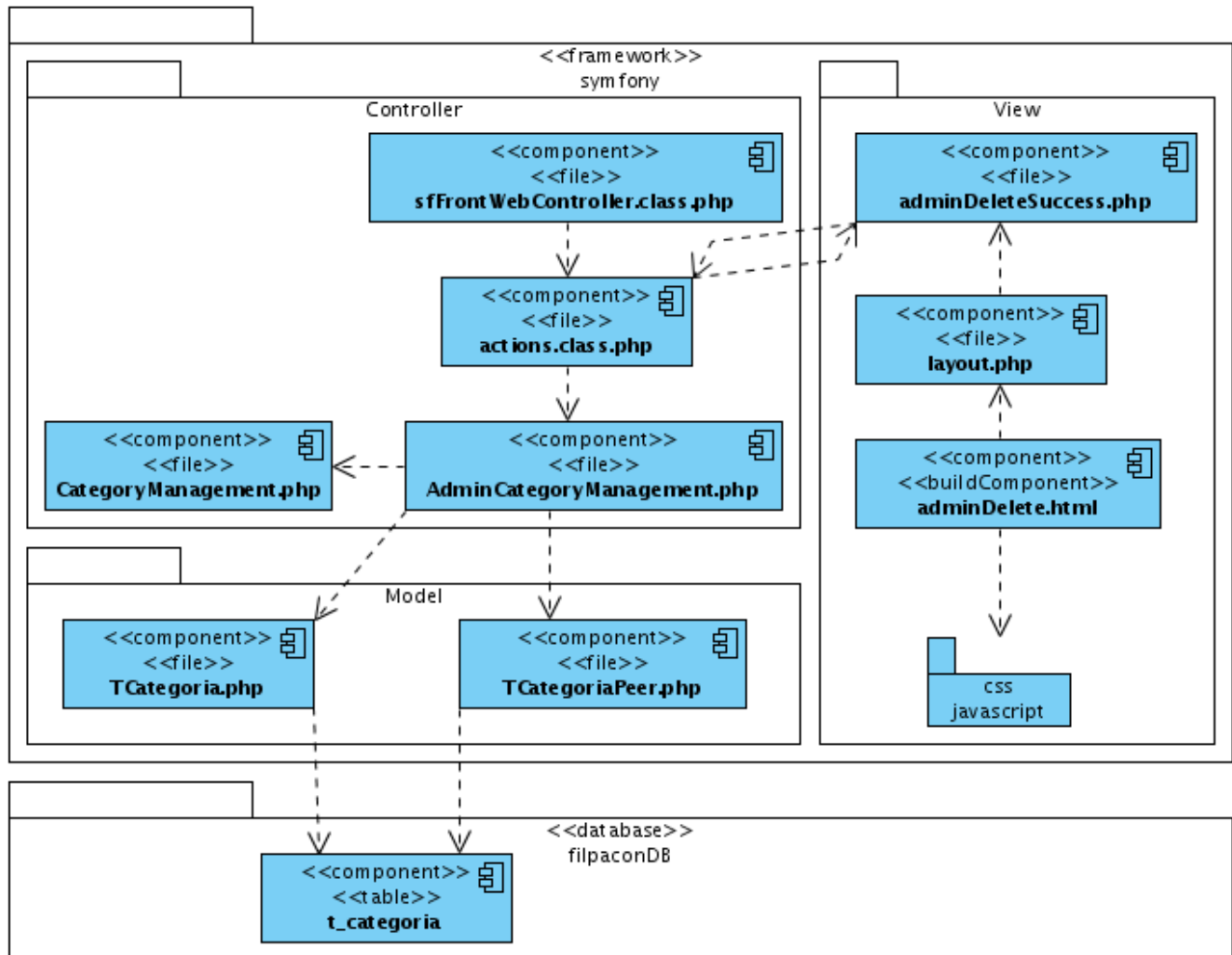


Figura F.13: Diagrama de Componente - Caso de Uso eliminar categoría.

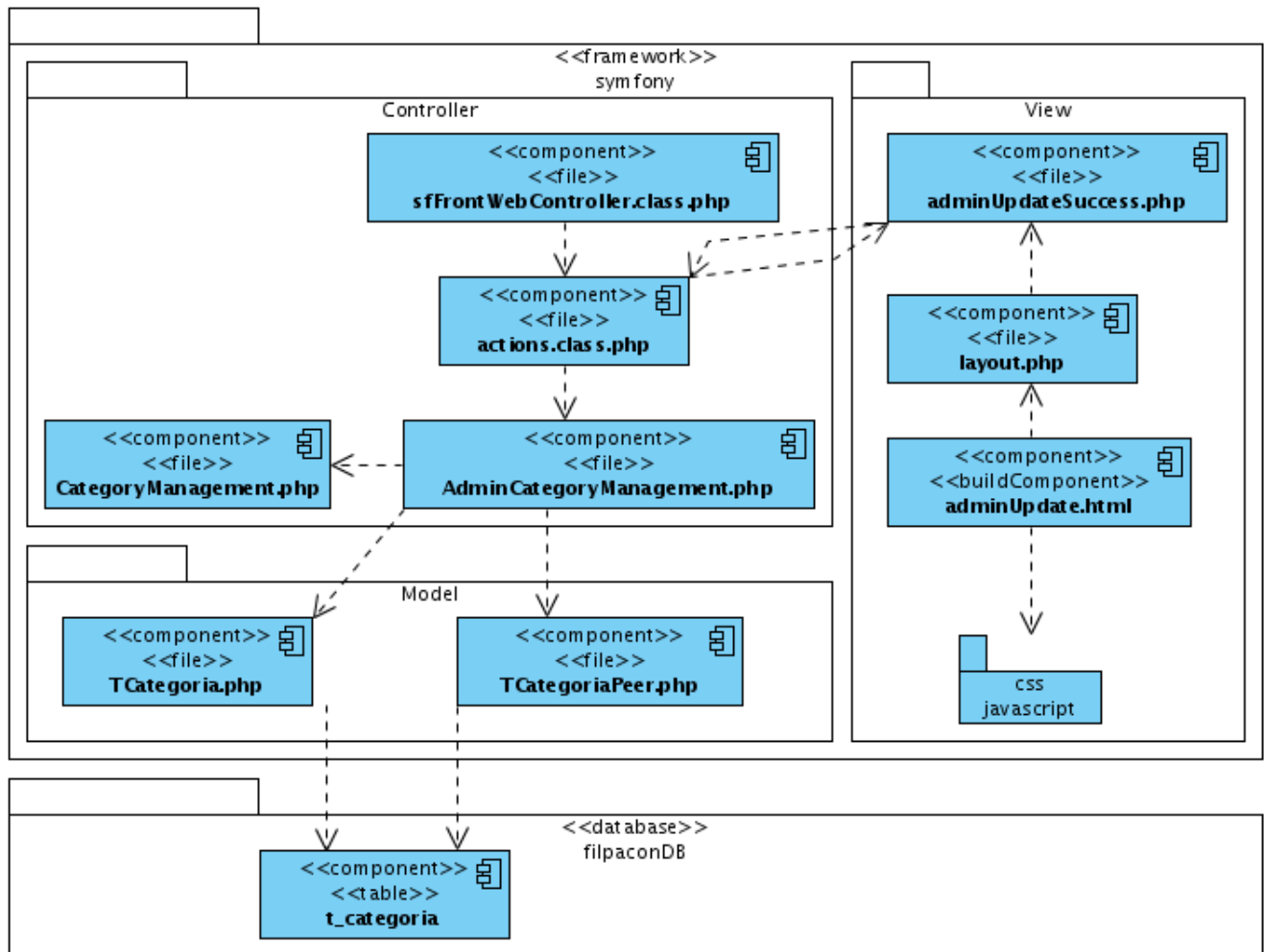


Figura F.14: Diagrama de Componente - Caso de Uso actualizar categoría categoría.

## Módulo Dominios de Navegación Básica

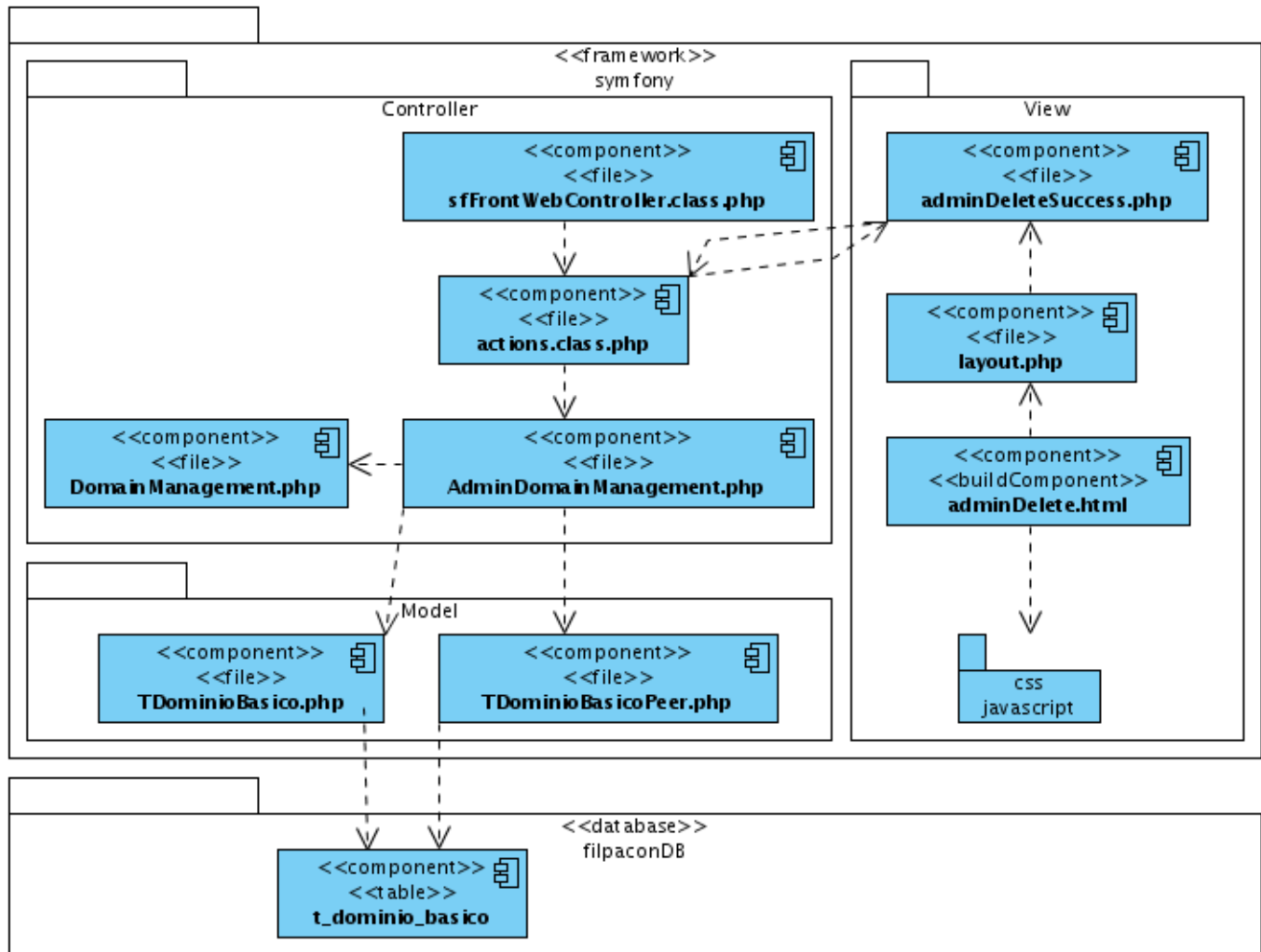


Figura F.15: Diagrama de Componente - Caso de Uso eliminar dominio de navegación básica.

## Módulo Reporte

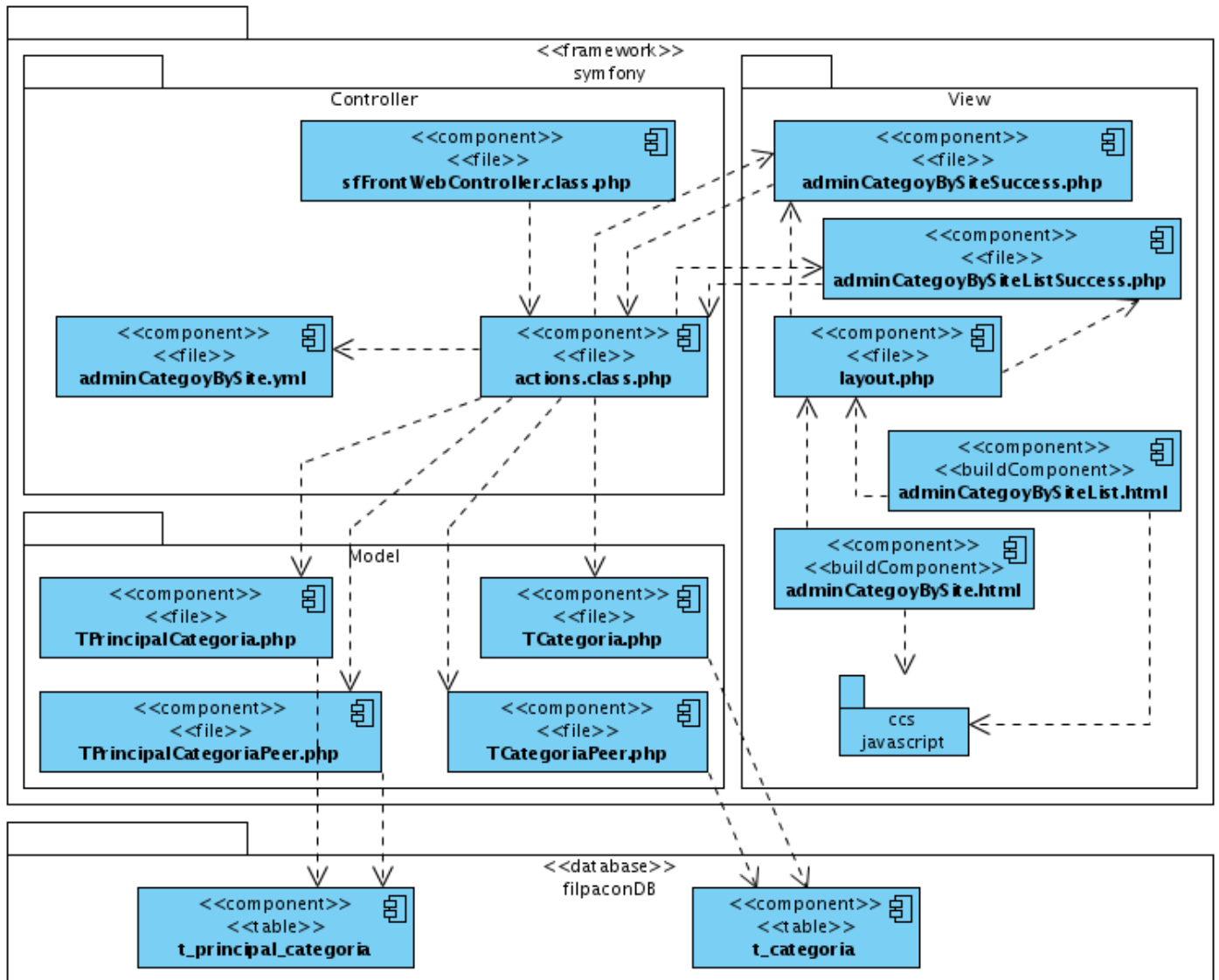


Figura F.16: Diagrama de Componente - Caso de Uso cantidad de sitios por categoría.

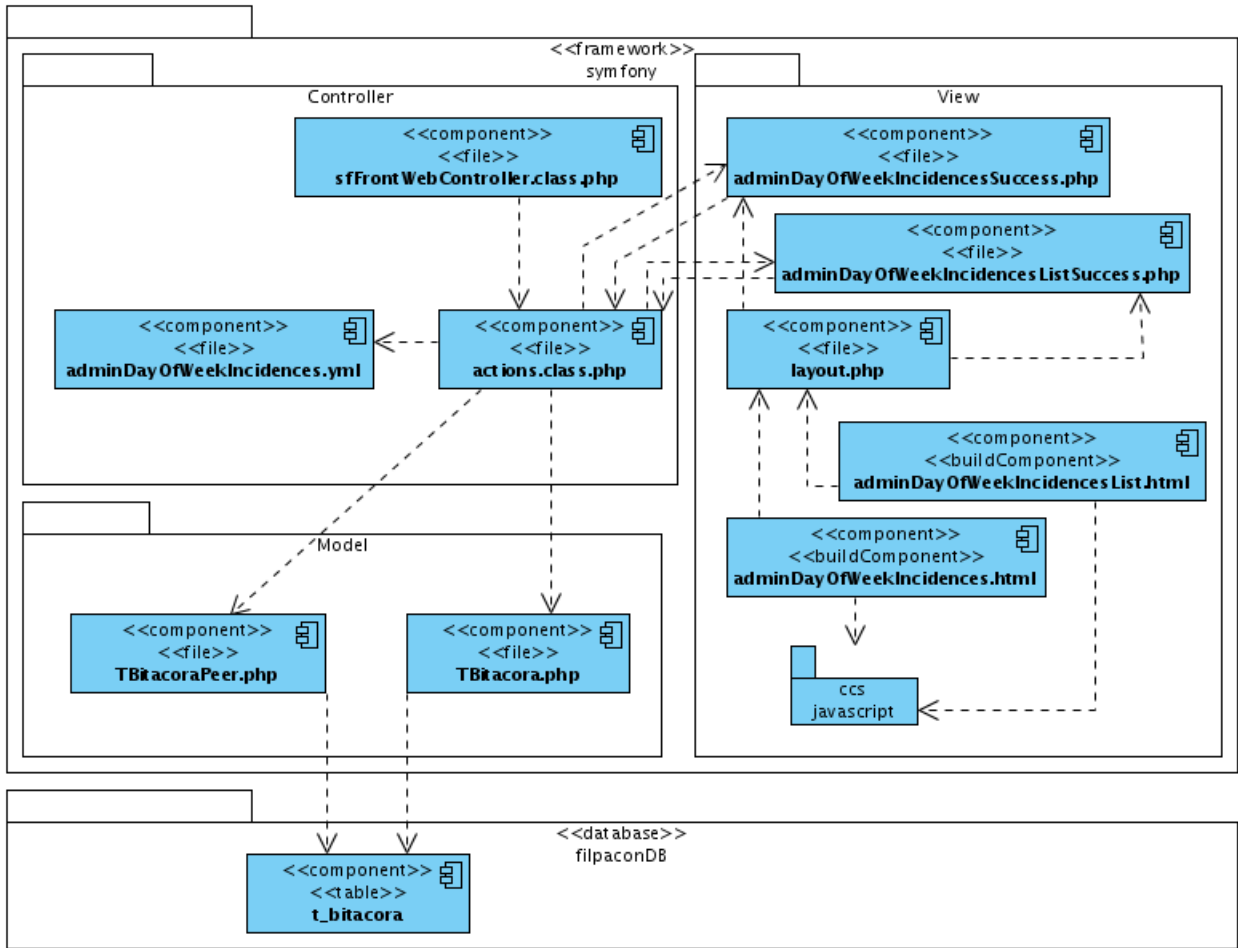


Figura F.17: Diagrama de Componente - Caso de Uso ranking de días de la semana por incidencias.

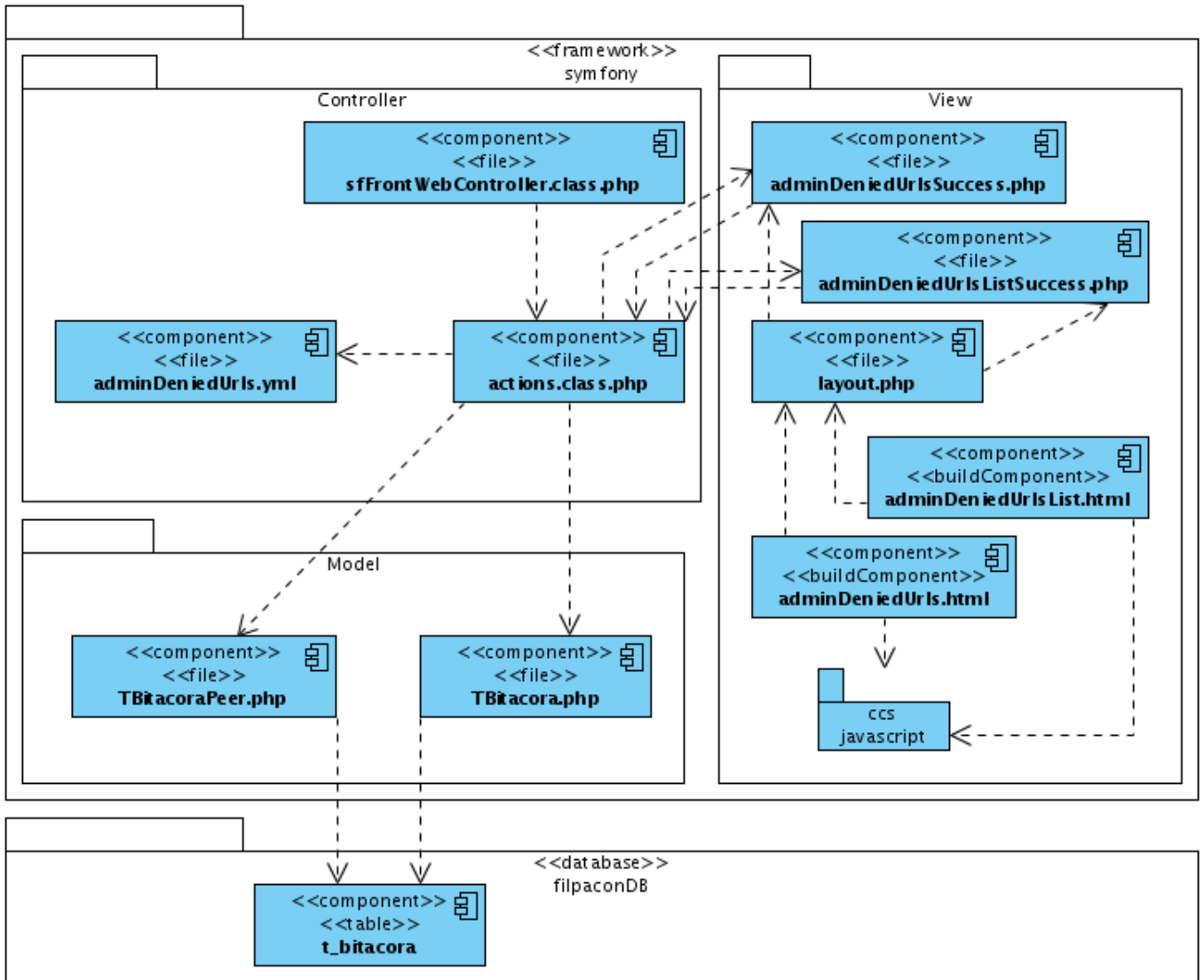


Figura F.18: Diagrama de Componente - Caso de Uso ranking de URLs denegadas.



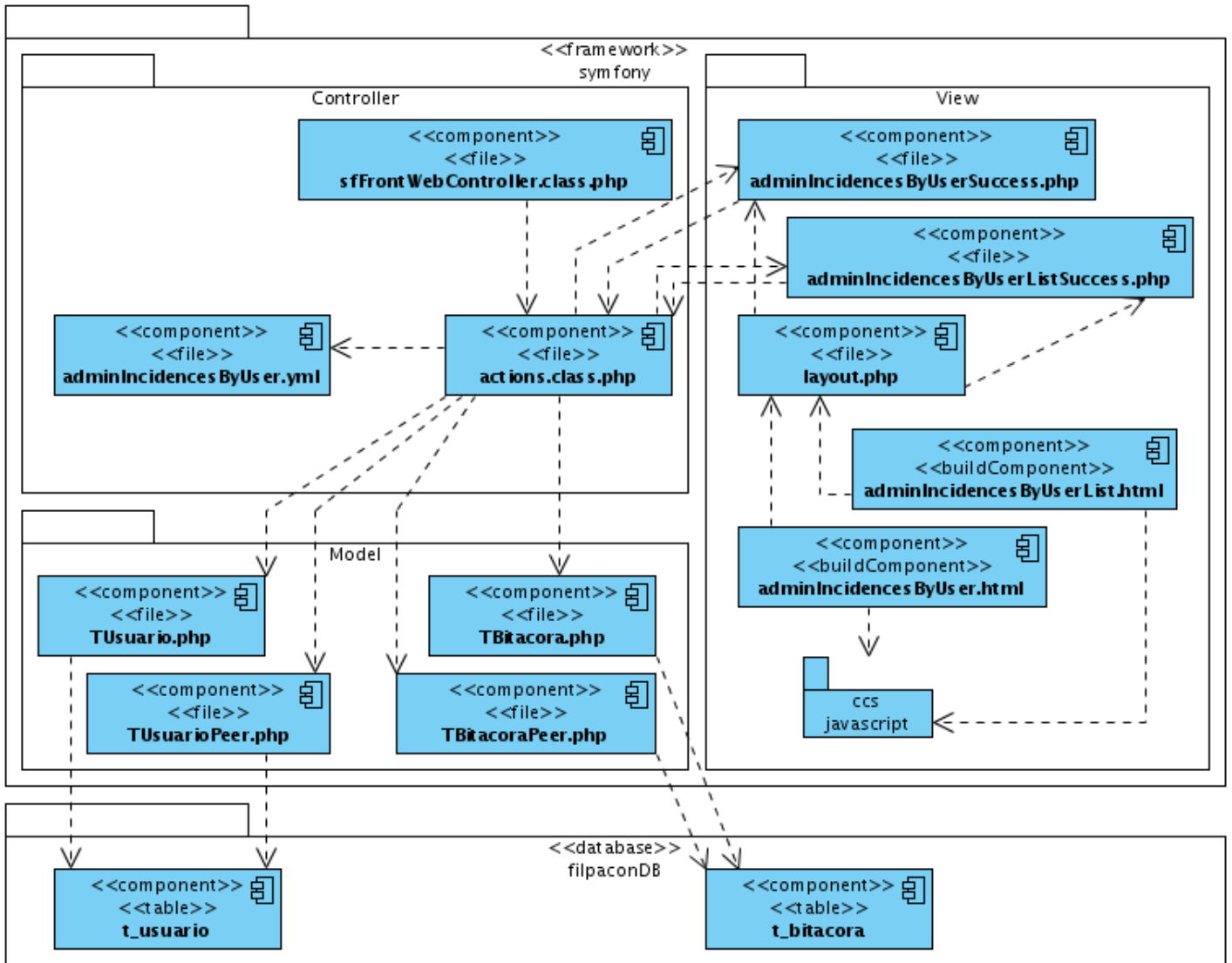


Figura F.19: Diagrama de Componente - Caso de Uso incidencia de usuario.

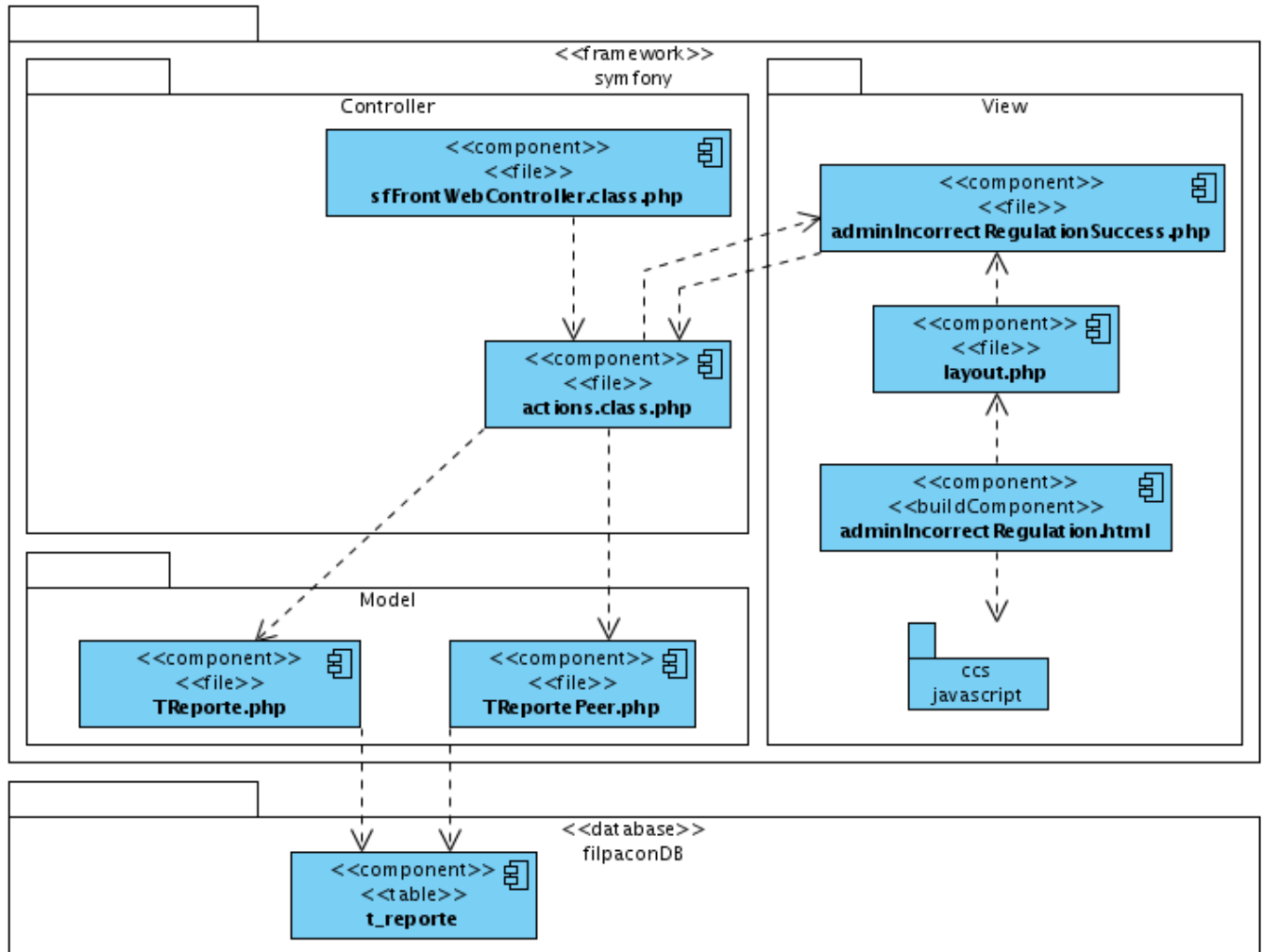


Figura F.20: Diagrama de Componente - Caso de Uso regulaciones incorrectas.

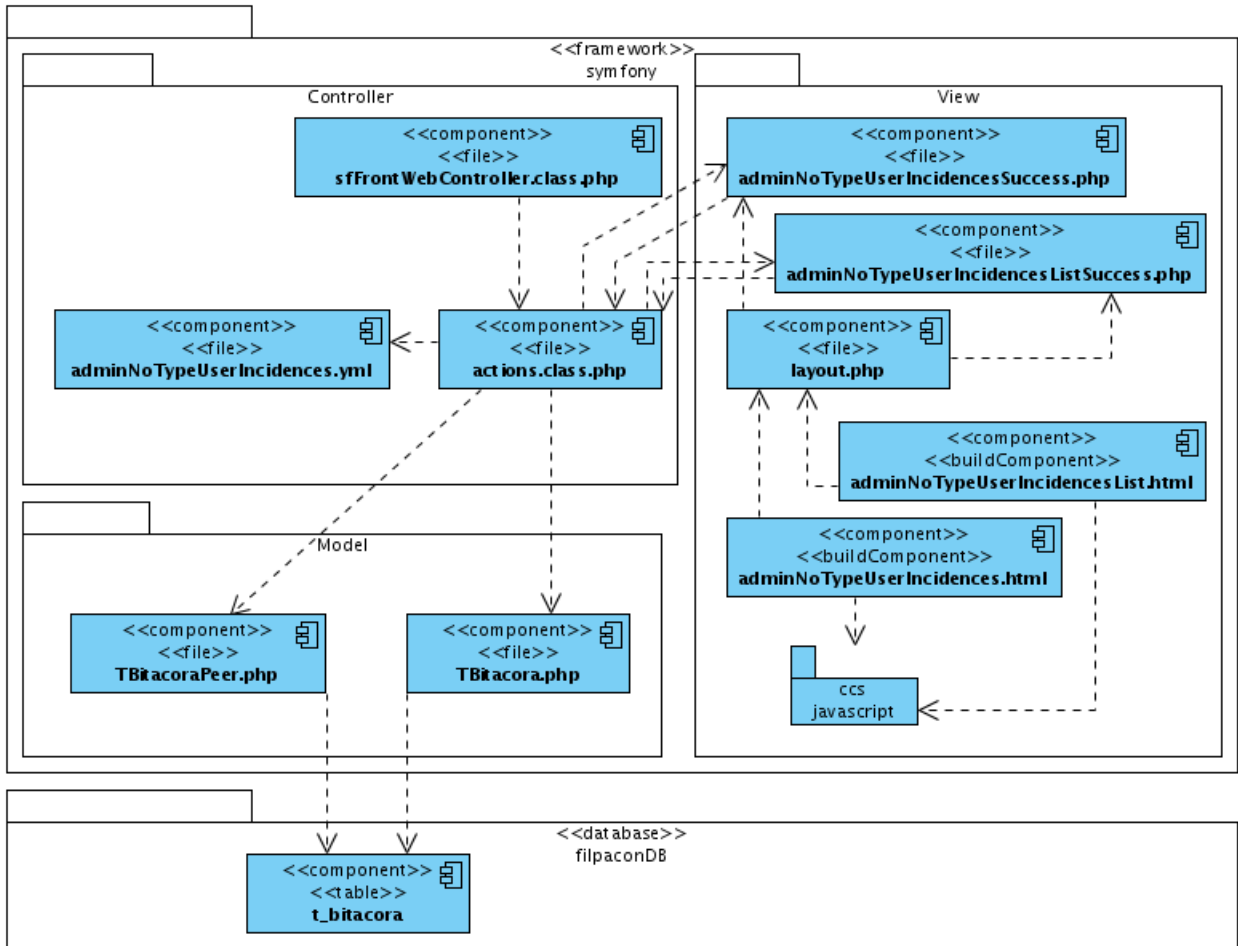


Figura F.21: Diagrama de Componente - Caso de Uso ranking incidencias por usuario.

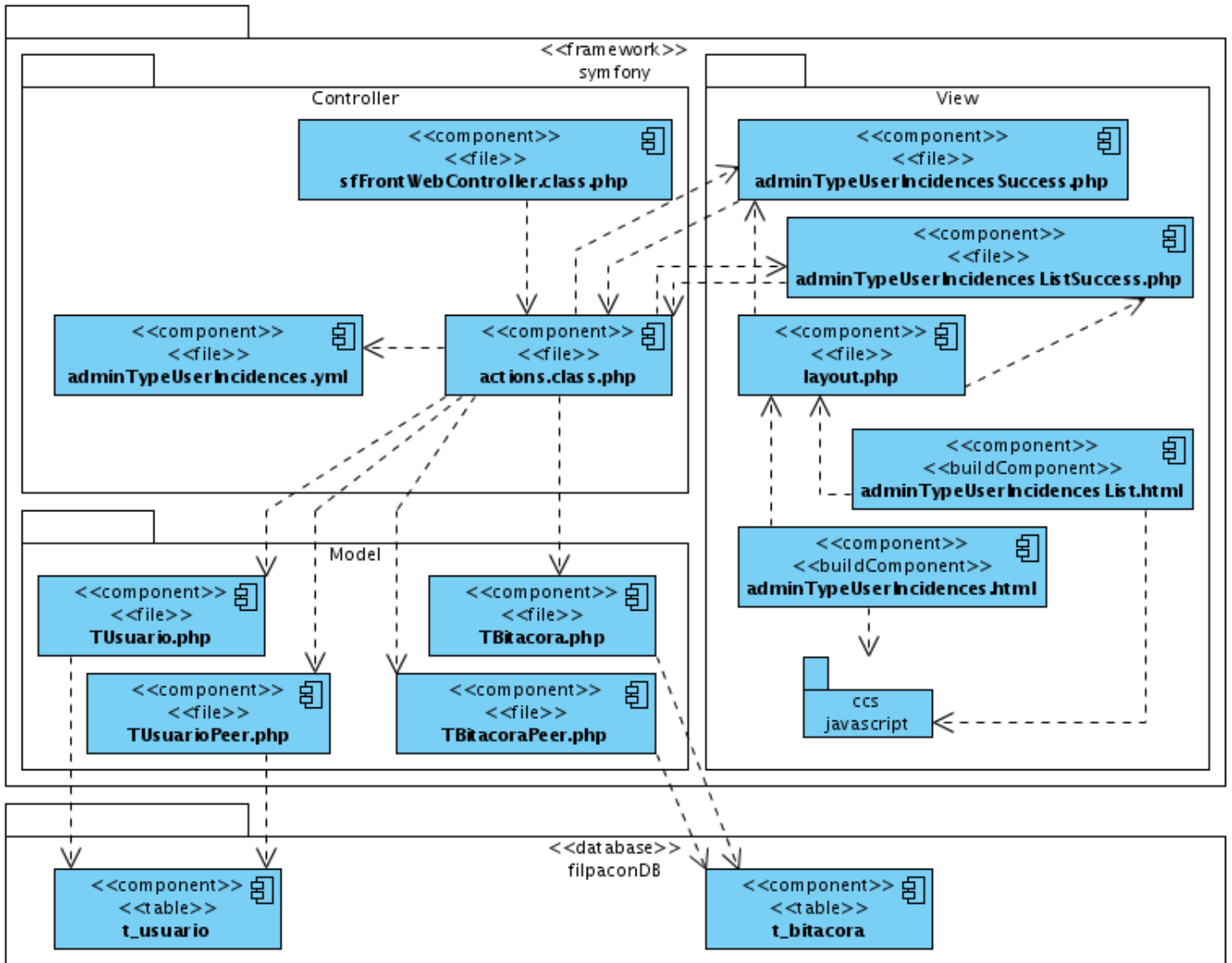


Figura F.22: Diagrama de Componente - Caso de Uso incidencias por tipo usuario.

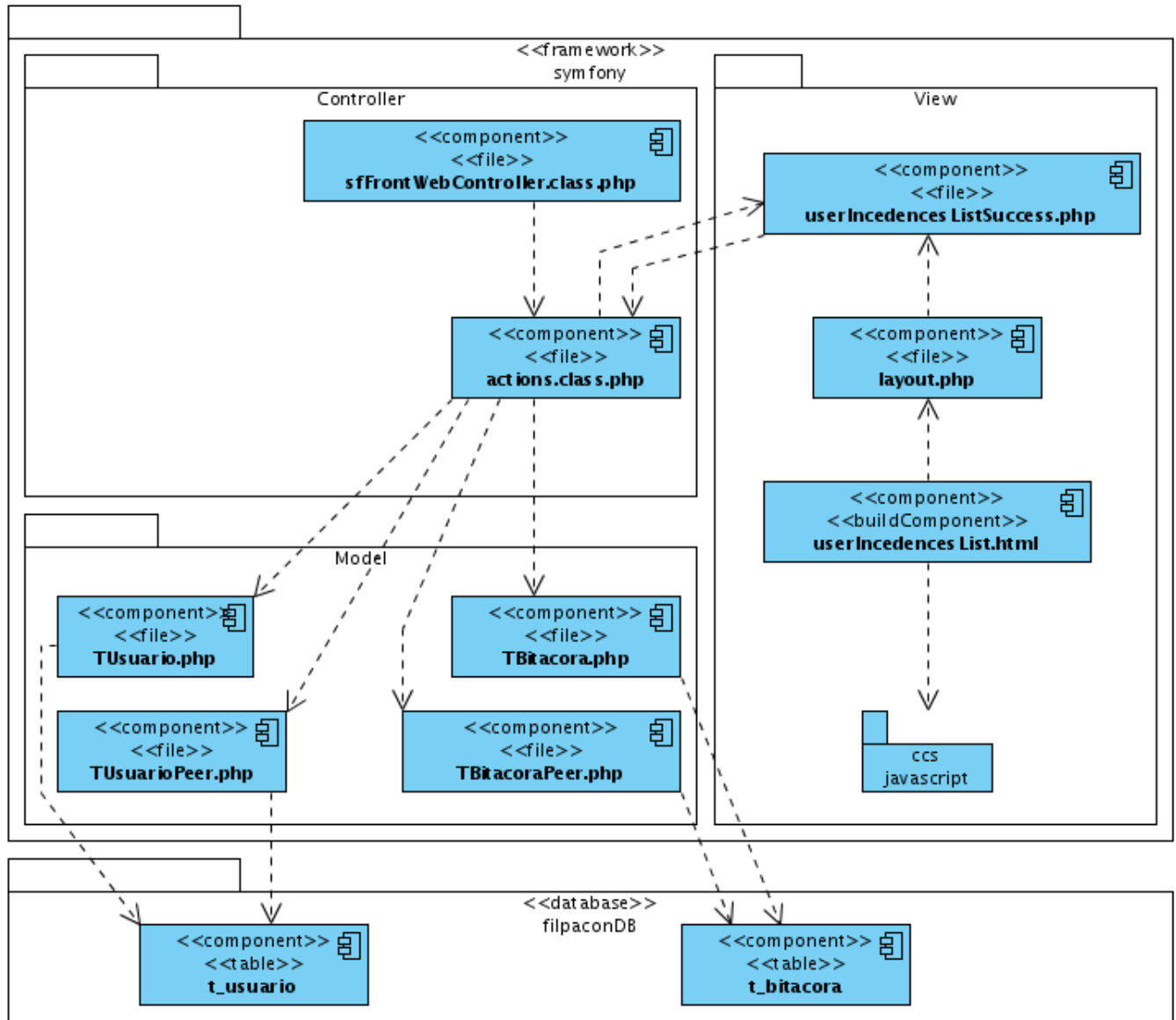


Figura F.23: Diagrama de Componente - Caso de Uso mis incidencias.