



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 2**

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas**

Gestión de trazas en el Sistema para el control Farmacológico del Centro de Informática Médica

**Autores:** Henry Martin Frías

Yanmichel Usatorres Ramirez

**Tutores:** Ing. Yoiler Joaquin Frometa Moreno

Ing. Guillermo Suárez Lorenzo

**Co-Tutor:** Ing. Yunior Pacheco Correa

**La Habana, junio de 2014**

**“Año 56 de la Revolución”**

*El estudiante tiene el deber de permanecer siempre puro, por la dignidad de su misión social, sacrificándolo todo en aras de la verdad moral e intelectual.*

*Julio Antonio Mella*

## **Datos de contacto**

### **Ing. Yoiler Joaquin Frometa Moreno:**

Graduado de Ingeniero en Ciencias Informáticas en la UCI. Pertenece al Centro de Informática Médica (CESIM). Se desempeña como Desarrollador del proyecto Sistemas para el control Farmacológico en el Departamento de Sistemas Especializados en Salud.

Correo electrónico: [yfrometa@uci.cu](mailto:yfrometa@uci.cu)

### **Ing. Guillermo Suárez Lorenzo:**

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2011. Pertenece al Centro de Informática Médica (CESIM). Desarrollador en el Departamento de Sistema de Apoyo a la Salud.

Correo electrónico: [glorenzo@uci.cu](mailto:glorenzo@uci.cu)

### **Ing. Yunior Pacheco Correa:**

Ingeniero en Ciencias Informáticas recién graduado en adiestramiento. Se desempeña como Desarrollador del proyecto Synta en el Departamento Sistemas Especializados de Salud y se tituló con un componente de gestión de trazas para el Sistema Nomenclador de Información.

Correo electrónico: [ypacheco@uci.cu](mailto:ypacheco@uci.cu)

## Agradecimientos

*De Henry*

*Agradezco a mis padres (Tony y Danay) y abuelos (Julia y Germán) por haberme guiado siempre por el buen camino, por haber tenido tanta paciencia conmigo, tanta confianza y por haber creído en mí. En especial a ese MULTIFUNCIONAL ABUELO, amigo, padre y guía. A "Papá".*

*A mis tíos, Yoel y Olaidy. Por brindarme ese apoyo cuando más lo necesite. Y a mis primos Elaine y Yoelvis.*

*Agradezco a mis hermanos, Keren, Kevin, Anthony y Yenny.*

*Agradezco a mi tía Dolores y a mi prima Talya, gracias por brindarme su casa, su comida, sus consejos y la compañía que muchas veces necesite cuando pasaba por momentos difíciles.*

*A mi segunda mamá en la uci, la profesora Zoraida y su hija Lisandra, muchas gracias por cuidarme cada catarro, por bajarme las fiebres y por celebrarme los cumpleaños.*

*A mis grandes amigos, esas personas que siempre han estado ahí a cada momento para empujarme a seguir, para darme un consejo, un regaño y una felicitación. Gracias por secar mis lágrimas cuando yo pensé que no podía continuar y decirme, sigue que tú puedes. Ellos son Alianny, Ramses, Saralys, Anabel, Katherine, Beyda, Yenisbel, Marilyn, Mailin, Remberto, Aramis, Nerbys, Suyen, Evelyn, Yanelis, Neda, Jakelin.*

*A los amigos de la universidad con los que compartí grandes momentos, algunos amargos, otros no tanto y otros de mucha alegría. Estos que entraron conmigo y ya hoy muchos son ingenieros. Gracias Zahily, Manuel Emilio, Mirna, Alejandro, Chuchy, Dainier, Yamila, Heily, Yudy, Rogeidys, Victor y los que faltan de aquel maravilloso grupo 7103.*

*A mis amigos de grupo. Eiler, Beatriz, Mairelis, Dianiset, Juan Miguel, Yordan, Daniel, Mary Nelsa.*

*Agradezco a esas personas que también formaron parte de mi vida en la universidad a lo largo de todos estos años. Esas personas a las cuales les agradezco cada momento compartido. ellos son: Yani, Yunior, Martha Rocio, Wendy, Gabriel Hurtado, Rosali, Maiyara.*

*A mi compañero de tesis, que sin su perseverancia hoy no fuera posible este sueño. Muchas gracias Yanmichel.*

*A todos mis profesores a lo largo de esta carrera. Muchas gracias por confiar en mí y por darme esa oportunidad que necesitaba para demostrar que si podía llegar.*

*A mis tutores Guillermo, Yunior y Yoiler por darme ese empujón que necesitaba para concluir el trabajo de diploma. Gracias al tutor Yoiler por dedicar gran parte de su tiempo libre para revisar el documento, incluso cuidando a sus niños.*

*Al piquete de la FEU especialmente al de la facultad 7 por hacerme crecer como diseñador y como reportero en cada evento.*

*Agradezco mucho a todas las personas que han estado para darme un consejo, un empujón para continuar, un aliento para seguir y a las personas que me criticaron en determinado momento, también muchas gracias pues me ayudaron a superarme a cada instante.*

*De Yanmichel*

*Agradecer principalmente a mis padres y mis abuelos que me han apoyado en todo el transcurso de la carrera en los momentos buenos y malos.*

*A mi familia por apoyarme desde lejos y darme fuerzas para no darme por vencido.*

*Agradecer a mi tía Irma por su ayuda en los consejos que me ha dado sobre las experiencias que ha tenido en el transcurso de su licenciatura.*

*A mi novia Elizabeth y mi suegra Silvia por estar a mi lado y brindarme su apoyo en los momentos buenos y malos que he tenido durante la carrera.*

*Agradecer a todo mis compañeros de clase por ayudarme a prepararme para enfrentar los exámenes.*

*Agradecer a los profesores y especialistas del proyecto por ayudarme cuando he necesitado su ayuda, especialmente a mi co-tutor Yunior Pacheco Correa y a mis tutores los Ingenieros Yoiler Joaquín Frometa y Guillermo Suarez Lorenzo.*

*No por último es menos importante agradezco a mi compañero de tesis, por estar siempre dispuesto y sin su ayuda no hubiese sido posible la realización de este trabajo de diploma.*

## Dedicatoria

*De Henry*

*Dedico mi trabajo de diploma a mis abuelos Julia y Germán (Papá), por darme ese amor incondicional y esa confianza que tanto necesité en todo momento.*

*A mis padres Antonio y Danay por cumplir cada uno de mis caprichos durante toda mi vida.*

*A todos mis hermanos, en especial a ese que espero siga mis pasos. Yenny.*

*De Yanmichel*

*Le dedico mi trabajo de diploma a mi familia, principalmente a mis padres y a Mamá, que aunque no se encuentre físicamente a mi lado ha sido mi guía de inspiración y ejemplo a seguir para conseguir las metas que me he propuesto.*

**Resumen**

La *gestión de trazas* en un sistema informático se emplea para guardar el historial de acciones de los usuarios autenticados. Al Sistema para el control Farmacológico se le adiciona la funcionalidad de gestión de trazas, con el objetivo de monitorizar las operaciones realizadas. Por otra parte, el sistema permite la recuperación de los datos eliminados. Esta aplicación consta de varios *privilegios* de acuerdo a la jerarquía de cada usuario. Los usuarios Administradores solo pueden gestionar las *trazas* de su nivel y del área a la cual pertenece. Los niveles son los siguientes: Nacional, Provincial, Municipal o Unidad de Salud. El sistema controla el acceso de los usuarios a la aplicación comprobando los datos introducidos por el usuario con los almacenados en la base de datos.

Las nuevas funcionalidades adicionadas al Sistema para el control Farmacológico; permiten la gestión de trazas de las acciones realizadas por los usuarios en los *módulos* Administración, Consumo de Medicamentos, Tarjeta de Control, Reacciones Adversas a Medicamentos y Control de Recetas Médicas. Para el desarrollo de la solución se utilizó Symfony (v1.4) como *framework* de desarrollo, este utiliza el patrón de arquitectura de software Modelo Vista Controlador. Como lenguaje de programación del lado del servidor se utilizó PHP (v5.3) y del lado del cliente JavaScript (v1.10). Como herramienta de modelado se empleó el Enterprise Architect (v7.1) y para el diseño el Lenguaje Unificado de Modelado (v2.1). Como gestor de base de datos el MySQL (v5.6) y Doctrine (v1.2), como mapeo de objeto relacional (ORM).

**Palabras clave:** *framework, gestión de trazas, módulos, privilegios, trazas*

---

Índice	
Resumen.....	VI
Introducción.....	1
Capítulo 1: Fundamentación teórica.....	7
1.1 Principales conceptos asociados al dominio del problema .....	7
1.2 Sistemas existentes a nivel internacional para la gestión de trazas .....	9
1.2.1 Sellenne .....	9
1.2.2 Bitácora V5.....	9
1.2.3 Tool Traza ERP.....	9
1.2.4 Trazatec .....	10
1.3 Sistemas existentes a nivel nacional para la gestión de trazas.....	10
1.3.1 Gestión de trazas de eventos para la Minería de Procesos en Sistemas de Información Orientada a Datos .....	10
1.3.2 Sistema de Autenticación, Autorización y Auditoría (SAAA) .....	10
1.3.3 Sistema Nomenclador de Información. ....	11
1.3.4 Componente de Seguridad para aplicaciones del Área Temática Sistema de Apoyo a la Salud. ....	11
1.3.5 Resultados de la investigación. ....	12
1.4 Tecnologías, herramientas y metodología en las que se apoya la solución del problema....	13
1.4.1 Servidor Web Apache 2.2 .....	13
1.4.2 JavaScript 1.10 .....	13
1.4.3 ExtJS 3.3.....	14
1.4.4 PHP 5.3 .....	14
1.4.5 Symfony 1.4 .....	15
1.4.6 NetBeans IDE 7.3 .....	15
1.4.7 MySQL 5.0 .....	15
1.4.8 RUP .....	16
1.4.9 Enterprise Architect 7.1 .....	16
Conclusiones del capítulo .....	16
Capítulo 2: Características de las funcionalidades de gestión de trazas .....	17
2.1 Propuesta de solución .....	17
2.2 Diagrama del Modelo de Dominio .....	18
2.2.1 Conceptos fundamentales .....	18

---

---

2.3	Requerimientos del Software.....	19
2.3.1	Requerimientos Funcionales (RF).....	20
2.3.2	Requerimientos no Funcionales (RNF) .....	21
2.4	Actores del Sistema.....	24
2.5	Casos de Uso.....	24
2.6	Diagrama de Casos de Uso del Sistema .....	25
2.7	Descripción textual de Casos de Uso del Sistema .....	26
	Conclusiones del Capítulo .....	34
	Capítulo 3: Diseño e implementación de las funcionalidades gestionar trazas del Sistema para el control Farmacológico.....	35
3.1	Descripción de la arquitectura .....	35
3.2	Definición de elementos del diseño.....	36
3.3	Diagramas de Clases del diseño.....	37
3.3.1	Descripción de las clases .....	39
3.4	Modelo de datos.....	41
3.4	Descripción de la funcionalidad gestión de trazas.....	44
3.4.1	Gestión de Trazas.....	44
3.5	Diagrama de Despliegue .....	45
3.6	Modelo de implementación.....	45
3.7	Diagrama de componentes.....	45
3.8	Tratamiento de errores .....	50
3.9	Estándares de codificación.....	51
3.9.1	Estándares utilizados en la codificación .....	51
3.9.2	Estilo de codificación utilizado .....	52
	Conclusiones Generales.....	53
	Recomendaciones .....	54
	Referencias Bibliográficas .....	55
	Bibliografía .....	60
	Anexos .....	65
	Anexo 1 Entrevista .....	65
	Anexo 2 Diagrama que representa el patrón MVC en Symfony.....	66
	Anexo 3 Diagrama que muestra el flujo de trabajo de Symfony.....	67

---

## Introducción

El desarrollo científico técnico alcanzado en la época actual en manos de la sociedad, garantiza que el uso de nuevas tecnologías incremente la productividad y el bienestar del hombre, entre las que se destaca la computación, la que ha pasado a ocupar un lugar trascendente en todas las esferas de la vida social. Ello revela la importancia de que todos los miembros de la sociedad estén preparados para su utilización en los diferentes sectores sociales en que se encuentran insertados; y fundamenta la necesidad de crear una cultura informática en la sociedad. (Moreno, 2011)

La informática en los últimos tiempos se ha convertido en parte del sustrato tecnológico del proceso de globalización en el cual está inmerso todo el mundo, lo que implica la necesidad de preparar a las nuevas generaciones para la asimilación y utilización de dicha tecnología. (Inder, 2004)

Son innumerables los beneficios que brinda esta rama de las ciencias, por ejemplo, la rapidez en la obtención de resultados, almacenamiento de grandes volúmenes de información, facilidades para encontrar información adecuada y/o actualizada por parte de científicos, investigadores, profesionales, estudiantes con lo cual se trabaja intensamente en nuestro país para ir incorporando de forma progresiva estos elementos al servicio de toda la población. (Inder, 2004)

En la actividad de seguridad informática en la salud, el principal objetivo es tomar conciencia con respecto a que cada dato a proteger no es simplemente un dato estadístico o una anotación, es un paciente reflejado en un sistema informático con el objetivo de que, apoyándose en las nuevas tecnologías, su estudio y curación sean más rápidos y seguros. Es por ello importante definir la política que ha de seguir cada institución, los principios que la rigen, el análisis y gestión de riesgos en los que se debe trabajar y las medidas que deben prevalecer en cada lugar donde exista un equipo en función de la salud de nuestro país, ya que los propósitos pueden ser múltiples pero cada uno se entrelaza en una misma función. (Vidal Ledo, y otros)

Para lograr un mayor entendimiento de lo que es la seguridad informática a continuación se expone un concepto asociado.

**Seguridad informática:** consiste en aquellas prácticas que se llevan adelante respecto a un determinado sistema de computación a fin de proteger y resguardar su funcionamiento y la información en él contenida. (ABC, 2014)

La seguridad informática también es considerada como políticas para la protección de los datos, establecidas en las empresas por el departamento de seguridad informática o un grupo de expertos en el tema. Dichas políticas consta en la mayoría de las ocasiones de restringir el acceso a las computadoras y software de la empresa, sólo al personal autorizado. Así garantizan que individuos externos o ajenos a la entidad, dañen la integridad, disponibilidad y confidencialidad de la información.

### **¿Qué es una traza?**

Según lo planteado en el diccionario The Free Dicccionario (Diccionario Libre), expresa como concepto de traza: 1-huella, rastro de una persona o cosa a seguir, las trazas de la presa, 2-habilidad de una persona para hacer cierta cosa y 3- plan para realizar un fin, la traza o rastro de un presupuesto. (The Free Diccionario, 2013)

En la rama de la informática, se podría definir la palabra traza como: registro de acciones realizadas por un usuario o una aplicación al autenticarse en un sistema. Una traza nos permite ver el pasado, pudiendo ver qué se ha enviado y qué se ha recibido, por lo que suele ayudar a entender el comportamiento de un proceso. Enfocando este concepto en el Sistema para el control Farmacológico se puede decir que las trazas se generan cuando un usuario realiza acciones como son: crear paciente, modificar población, insertar certificado, entre otras.

### **¿Por qué es importante guardar las trazas en un sistema?**

Los resultados de una traza se pueden guardar en un archivo o en una base de datos. Se pueden utilizar los archivos de traza para realizar lo siguiente:

- Auditar la base de datos.
- Realizar análisis de rendimiento.
- Llevar el control de las acciones de cada usuario.
- Optimizar las consultas.

Como se describe anteriormente, una traza guarda un historial de acciones que va siendo generado una vez que un usuario se autentique en la aplicación. Dicho historial es revisado al

ocurrir determinado incidente en la información, o simplemente porque la empresa o entidad está siendo auditada por los administrativos informáticos.

En caso de eliminación de la información, mediante el estudio de la traza de ese sistema es posible determinar con exactitud la persona que provocó el incidente, así como el lugar, la fecha y otros datos, de manera que queda al descubierto su identidad y el individuo sería sancionado de acuerdo a las regulaciones informáticas establecidas en su empresa.

El Centro de Informática Médica (CESIM) de la Universidad de las Ciencias Informáticas se especializa en la producción de software para el sector de la salud, con alcance nacional e internacional.

Uno de los productos del CESIM es el Sistema para el control Farmacológico (Synta), que tiene como objetivo informatizar algunos procesos ejecutados por el Departamento de Farmacoepidemiología del Ministerio de Salud Pública de Cuba. Dentro de estos procesos se encuentran: el control de la distribución de las recetas médicas en el país, control de las inscripciones al consumo de medicamentos controlados, registro de las reacciones adversas a medicamentos, conformación y ejecución del cuadro básico de medicamentos en Cuba.

Con la futura implementación del sistema, el Ministerio de Salud Pública (MINSAP) tendrá la información relacionada a la distribución de recetas médicas centralizada, permitirá llevar el control del consumo de medicamentos a nivel nacional. Se establecerán comparaciones con años anteriores y posibles pronósticos del consumo del medicamento. Se obtendrá un control eficiente de los pacientes inscritos a los medicamentos controlados así como identificar las Reacciones Adversas a Medicamentos (RAM) ocurridas en un tiempo determinado.

Actualmente el Sistema para el control Farmacológico puede controlar el acceso de los usuarios a las funcionalidades, evitando que personas no autorizadas hagan uso de la información, pero no se puede identificar los responsables de la realización de determinados cambios que puedan afectar la veracidad de dicha información. Además, no se puede conocer qué persona modificó, eliminó o insertó un determinado valor, o en qué momento o estación de trabajo lo hizo. Esto provoca que no se puedan tomar medidas administrativas con la persona que intente o dañe la integridad de los datos.

Es imposible llevar a su estado anterior los cambios realizados a la información del sistema, en caso que sea necesario. Ejemplo de esto se puede evidenciar de la siguiente forma: si un determinado usuario altera el plan de entrega de recetas médicas, esto puede provocar el desvío de medicamentos, generando un faltante en las farmacias y centros de salud del país.

A partir de la situación expuesta anteriormente se tiene como **problema a resolver**: el Sistema para el control Farmacológico no registra las acciones ejecutadas por los usuarios, dificultando el control de los cambios de la información y la integridad de la misma.

Se delimita como **objeto de estudio** el proceso de gestión de trazas en aplicaciones web, y como **campo de acción**, procesos de gestión de trazas en el Sistema para el control Farmacológico del Centro de Informática Médica

**Objetivo general:** Desarrollar un conjunto de funcionalidades que garanticen la gestión de trazas y la revocación de la pérdida de datos en el Sistema para el control Farmacológico del Centro de Informática Médica que permita llevar el control de los cambios de la información así como la integridad de la misma.

Para solucionar el objetivo planteado se proponen las siguientes **tareas de la investigación**:

- Analizar los sistemas de gestión de trazas existentes a nivel nacional e internacional estableciendo similitudes con la investigación en curso.
- Analizar los módulos del Sistema para el control Farmacológico, para identificar del flujo de información, qué trazas se van a gestionar.
- Asimilar la arquitectura, tecnologías y herramientas definidas en el departamento para el desarrollo de las funcionalidades de gestión de trazas.
- Desarrollar los artefactos asociados a los flujos de trabajo de “Modelado de Negocio”, “Gestión de requerimientos”, “Diseño” e “Implementación”.
- Implementar funcionalidades que permitan la gestión de trazas de los diferentes módulos del Sistema para el control Farmacológico.

Los **métodos científicos** empleados son los siguientes:

### **Métodos Teóricos**

**Analítico-Sintético:** la utilización de este método permitió realizar un análisis del problema de investigación y desglosarlo en gestión de trazas.

**Histórico-Lógico:** fue preciso emplear este método para conocer sobre sistemas que incluyan la gestión de trazas.

**Inductivo-Deductivo:** se utilizó para obtener los conocimientos necesarios analizando los datos generales de gestión de trazas.

**Modelación:** se empleó este método para representar el problema de la realidad, en diagramas.

### **Métodos Empíricos**

**Entrevista:** Se le realizó la entrevista relacionada en el [Anexo 1](#) al Ing. Alfredo Sánchez Rodríguez, Jefe del grupo de Infraestructura del centro Softel de la Universidad de las Ciencias Informáticas. El tipo de entrevista realizada fue Entrevista a Profundidad.

**Análisis de documentos:** a través del cual se realizó un estudio de la documentación existente que el departamento y en los repositorios, para conocer el proceso de gestión de trazas en el proyecto Synta.

### **Beneficios esperados:**

- La implementación de un mecanismo capaz de gestionar las trazas en el Sistema para el control Farmacológico garantizará el registro de las acciones de los usuarios en los diferentes módulos.
- Se podrá restaurar la información que haya sido eliminada o modificada.
- Se podrá identificar con mayor facilidad y rapidez las acciones realizadas en la aplicación en dependencia de cada rol por usuario.

El desarrollo del trabajo se estructura en tres capítulos que se describen a continuación:

**Capítulo 1: Fundamentación teórica**, en este capítulo se incluye el estudio del estado del arte tanto a nivel nacional como internacional. Se hace un estudio sobre los componentes para garantizar la seguridad del Sistema para el control Farmacológico (Synta). Se describen las herramientas, tecnologías y metodologías a usar en la solución del problema planteado.

**Capítulo 2: Características de las funcionalidades de gestión de trazas del Sistema para el control Farmacológico**, en este capítulo se especifican los requerimientos funcionales y no funcionales, a partir de los cuales se representan los casos de uso del sistema y la descripción de los mismos. Se implementan los mecanismos definidos para la seguridad de la aplicación web y la gestión de trazas de los usuarios.

**Capítulo 3: Diseño e implementación de las funcionalidades de gestión de trazas**, en este capítulo se justifican los patrones a usar en el diseño de las funcionalidades, se define la estructura y elementos del diseño, mostrando los diagramas de clases del diseño e implementación.

## Capítulo 1: Fundamentación teórica

La fundamentación teórica de una tesis, es el estudio de conceptos asociados al tema de investigación. Donde se plantean los principios de los que parte el investigador, y a partir de ahí se desarrolla, analiza y define el asunto o problema de estudio, con base en información acumulada en el transcurso de la indagación bibliográfica.

### 1.1 Principales conceptos asociados al dominio del problema

Con el objetivo de lograr un mejor entendimiento del problema se explican varios conceptos.

#### **Confidencialidad**

Se entiende por confidencialidad el servicio de seguridad, o condición, que asegura que la información no pueda estar disponible o ser descubierta por o para personas, entidades o procesos no autorizados. La confidencialidad, a veces denominada secreto o privacidad, se refiere a la capacidad del sistema para evitar que personas no autorizadas puedan acceder a la información almacenada en él. (Vidal Ledo, y otros)

Algunos de los mecanismos utilizados para salvaguardar la confidencialidad de los datos son, por ejemplo:

- El uso de técnicas de control de acceso a los sistemas.
- El cifrado de la información confidencial o de las comunicaciones.

#### **Integridad**

Se entiende por integridad el servicio de seguridad que garantiza que la información es modificada, incluyendo su creación y borrado, sólo por el personal autorizado. Suelen integrarse varios conceptos análogos en este segundo aspecto de la seguridad (Vidal Ledo, y otros):

- precisión
- integridad
- autenticidad

El concepto de integridad significa que el sistema no debe modificar o corromper la información que almacene, o permitir que alguien no autorizado lo haga. Esta propiedad permite asegurar que no se ha falseado la información. Por ejemplo, que los datos recibidos o recuperados son exactamente los que fueron enviados o almacenados, sin que se haya producido ninguna modificación, adición o borrado. De hecho el problema de la integridad no sólo se refiere a

modificaciones intencionadas, sino también a cambios accidentales o no intencionados. (Vidal Ledo, y otros)

### **Autenticación**

Es el proceso de verificación de la identidad digital de un remitente de una comunicación que hace una petición para conectarse a un sistema. El remitente puede ser una persona que usa una computadora u otro medio electrónico, una computadora por sí misma o un programa. En otras palabras, es un modo de asegurar que los usuarios son realmente quienes dicen ser y que tienen la autorización para realizar funciones en el sistema. (Gómez, et al, 2009)

### **Autorización**

Proceso por el cual se autoriza al usuario identificado a acceder a determinados recursos del sistema, es decir, se comprueba que los usuarios con identidad válida solo tengan acceso a aquellos recursos sobre los cuales tienen privilegios. (Gómez, et al, 2009)

### **Auditoría**

Es la capacidad de un sistema de registrar eventos y rastrear la actividad del usuario mientras accede a los recursos, soportando sistemas de trazas. A través de la auditoría se detectan irregularidades en un sistema (indicando qué ha pasado, sobre qué información y quién ha accedido; formando parte imprescindible de cualquier sistema que pretenda proporcionar seguridad a la información). (RedUsers, 2013)

### **Traza**

Las trazas se gestionan en cualquier tipo de sistema donde se requiera hacer una revisión de los eventos, errores cometidos y las acciones realizadas por los usuarios dentro del mismo. Además permiten crear un registro de sucesos para un dispositivo o aplicación. A través de estos datos, se puede monitorear el funcionamiento de la aplicación y se ofrece una oportunidad para corregir problemas de seguridad. (Gómez, et al, 2009)

### **Trazabilidad**

Es la capacidad que tiene una organización o sistema para rastrear, reconstruir o establecer relaciones entre objetos monitoreados, para identificar y analizar situaciones específicas o generales en los mismos. (Cano, 2014)

## 1.2 Sistemas existentes a nivel internacional para la gestión de trazas

### 1.2.1 Sellenne

El software **Sellenne** es de origen peruano. En este software la trazabilidad no es un módulo, sino que se encuentra integrada en todo su entorno de trabajo participando absolutamente en cualquier movimiento (compras, ventas, inventariado, producción, etc.) de artículos que se realicen en esa empresa. Al gestionar una compra Sellenne genera automáticamente un código de entrada y lo hace por defecto, por ello podrá decidir trabajar su codificación de trazabilidad de la forma que se elija. Sellenne ERP Trazabilidad gestiona y almacena la traza de los productos y las operaciones, por eso es una trazabilidad estructural. (SynerPlus)

### 1.2.2 Bitácora V5

Este sistema da respuesta a las crecientes necesidades de las organizaciones en materia de gestión de la seguridad y de cumplimiento normativo. (Sales, 2010)

**S21sec**, compañía española líder en el sector especializado en servicios y productos de seguridad digital, ha creado “Bitácora V5”, Se trata de una única aplicación capaz de gestionar de manera sencilla, eficaz, automatizada y rápida la seguridad digital completa de la organización. (Sales, 2010)

- Almacena de forma unificada lo ocurrido en una red, realiza análisis forenses y ayuda a cumplir con los requisitos normativos y de auditoría.
- Reduce el tiempo de resolución de incidentes de seguridad.
- Detecta incidentes que otros sistemas no pueden detectar individualmente.

### 1.2.3 Tool Traza ERP

**Tools Traza ERP (Sistema de Planificación de Recursos Empresariales)** es el software de trazabilidad de medicamentos que surge del trabajo conjunto de Softlatam y Klap IT. Este nuevo desarrollo suma a todas las funcionalidades del sistema de gestión Tools ERP, el sistema de trazabilidad de medicamentos, obteniendo como resultado un producto íntegro que permitirá llevar la gestión administrativa, financiera y contable, y registrar la trazabilidad de los medicamentos que ingresan y egresan de la empresa. (SOFLATAM, 2012)

De este modo, la empresa, informando de manera automática al Sistema Nacional de Trazabilidad de Medicamentos y realizando su seguimiento correspondiente. (SOFLATAM, 2012)

#### **1.2.4 Trazatec**

**TRAZATEC** es un software de Trazabilidad de producción y de Gestión de la cadena de suministro para la industria en general y los sectores alimentación y distribución alimentaria en particular. Es totalmente flexible e integrable con bases de datos estándar, permitiendo complementar a cualquier ERP, ya que estos, no cubren particularidades específicas de entornos productivos complejos. (Asintec)

### **1.3 Sistemas existentes a nivel nacional para la gestión de trazas**

#### **1.3.1 Gestión de trazas de eventos para la Minería de Procesos en Sistemas de Información Orientada a Datos**

Sistema de Gestión de trazas de eventos para la Minería de Proceso en Sistemas de Información Orientados a Datos (Universidad de las Ciencias Informáticas, 2014)

El objetivo de esta investigación es desarrollar una aplicación en SW libre para la extracción de registros de eventos. El enfoque propuesto fue implementado mediante el desarrollo de un módulo en Sauxe (marco de trabajo para aplicaciones web de gestión desarrollado en la Universidad de las Ciencias Informáticas). Sauxe es utilizado por varias empresas desarrolladoras de software del país y Cedrux (ERP-Cuba). (Universidad de las Ciencias Informáticas, 2014)

El módulo implementado le permite a Sauxe la generación y exportación de trazas de eventos para la realización de estudios de Minería de Proceso.

Autores: Héctor David Peguero Álvarez, Katia Saria Preval, Damián Pérez Alfonso y Raykenler Yzquierdo Herrera

#### **1.3.2 Sistema de Autenticación, Autorización y Auditoría (SAAA)**

El Sistema SAAA se basa en una infraestructura Grid Security Infraestructure (GSI), es una tecnología compuesta por un conjunto de ordenadores que brinda servicios de seguridad de recursos distribuidos basado en normas de seguridad de certificados digitales x.509. El sistema

incluye funcionalidades que se encargan de gestionar las trazas de las acciones realizadas por los usuarios.

Autor: Empresa Softel.

Entrevistado: Ing. Alfredo Sánchez Rodríguez. Jefe del Grupo de Infraestructura de Softel UCI.

### **1.3.3 Sistema Nomenclador de Información.**

El Sistema Nomenclador de Información es un sistema reutilizable y generalizable, que se puede integrar a cualquier sistema que cuente en su desarrollo con información nomenclada. Para dicha integración es necesario el consumo de los servicios web que este brinda. La gestión de trazas permite el registro de las operaciones que se realizan en el sistema y quiénes las ejecutan. (Sánchez Santos, y otros, 2013)

Las trazas que se registran en este sistema son las siguientes:

- Usuarios modificados
- Usuarios eliminados
- Nomencladores modificados
- Nomencladores eliminados

### **1.3.4 Componente de Seguridad para aplicaciones del Área Temática Sistema de Apoyo a la Salud.**

El componente de Autenticación, Autorización y Auditoría (AAA) para aplicaciones basadas en Servicios Web XML permite una gestión eficiente de usuarios, asignación de roles y privilegios para el acceso a todos los sistemas externos que consuman los servicios proporcionados por este. Por otra parte brinda un eficiente y óptimo proceso de auditoría y trazabilidad, de manera que se lleve un control estricto de las operaciones en que se involucran los usuarios de los sistemas. (Arencibia Morales, y otros, 2013)

Este sistema mediante los procesos de administración proporcionados, permite una gestión eficiente de todos los requerimientos de seguridad para aquellos sistemas externos que consuman sus servicios, lográndose de esta forma la reutilización del código y evitándose que se realicen acciones innecesarias fuera de cada negocio. Esto se logra mediante la publicación de Servicios Web XML debidamente descritos utilizando el Lenguaje de Descripción de Servicios Web (WSDL). (Arencibia Morales, y otros, 2013)

### 1.3.5 Resultados de la investigación.

#### **Sistemas internacionales:**

La investigación sobre la existencia de estos sistemas que gestionan trazas arrojó como resultado, que no cumplen con las políticas para el desarrollo de aplicaciones informáticas en el sector de la salud definidas por el MINSAP, una de ellas define que:

- Todos los productos y servicios se integrarán a la ciberinfraestructura del sector y se realizarán sobre sistemas abiertos, utilizando software libre y de calidad.

Por lo tanto no sería posible la utilización del código fuente para modificarlo o adaptarlo a las necesidades de las funcionalidades de gestión de trazas. Además la gestión de trazas son específicas del negocio de estos sistemas.

#### **Sistemas nacionales:**

El estudio de los sistemas nacionales arrojó como resultado que:

Los sistemas SAAA estudiados ofrecen un mecanismo de seguridad mediante los procesos de Autenticación-Autorización y Auditoría, estrechamente vinculados entre sí y en el que las funcionalidades de gestión de trazas dependen de los procesos anteriores. El Sistema para el control Farmacológico cuenta con un mecanismo de Autenticación y Autorización propia, definidos por el negocio y no puede ser sustituido por un sistema externo, por lo tanto no se podrían aprovechar las funcionalidades de estos.

Como principal inconveniente se tiene que el Componente de Seguridad para aplicaciones del Área Temática Sistema de Apoyo a la Salud solo registran las operaciones realizadas por los usuarios y no es capaz de realizar la recuperación de cambios realizados, requisito indispensable entre las nuevas funcionalidades incorporadas al Sistema para el control Farmacológico, además no existe código fuente de dicha aplicación. En el caso del software Gestión de trazas de eventos para la Minería de Procesos en Sistemas de Información Orientada a Datos, la gestión de trazas es específica del negocio.

En el caso del Sistema Nomenclador de Información no pueden ser utilizadas las funcionalidades de gestión de trazas, pues son específicas de este sistema y de su compleja

estructura. Sin embargo debido a que fue desarrollado en el mismo departamento, se encuentra integrado al Sistema para el control Farmacológico como uno de sus módulos, por lo que comparten la misma arquitectura, por lo que se decidió tomar como guía el proceder utilizado en el sistema nomenclador para el desarrollo de las funcionalidades de gestión de trazas.

## **1.1 Tecnologías, herramientas y metodología en las que se apoya la solución del problema**

Para dar solución al problema en cuestión se utilizaron las herramientas y tecnologías establecidas por el proyecto Synta.

### **1.4.1 Servidor Web Apache 2.2**

El servidor HTTP Apache es un servidor web de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Se desarrolla dentro del proyecto HTTP Server (httpd) de la *Apache Software Foundation*. Es altamente configurable, admite bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración. Apache es una aplicación que permite montar un servidor web en cualquier equipo y la gran mayoría de los sistemas operativos. Al contrario que IIS (*Internet Information Server*) que sólo funciona en sistemas operativos de Microsoft. Este soporta PHP como lenguaje de programación y con los módulos adecuados, puede soportar también ASP. (Alonso, 2012)

### **1.4.2 JavaScript 1.10**

Es un lenguaje de programación del lado del cliente, que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como: texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript (JS) es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos con JS se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. JS puede funcionar como un procedimiento y un lenguaje orientado a objetos. Los objetos se crean mediante la programación en este lenguaje, fijando

los métodos y propiedades. Una vez que un objeto se ha construido puede ser utilizado como un modelo (o prototipo) para la creación de objetos similares. (Eguiluz, 2014)

### 1.4.3 ExtJS 3.3

ExtJS es un *framework* JavaScript que permite construir aplicaciones complejas en Internet. Este conjunto de librerías incluye: (Sánchez Rosas, 2008)

- Componentes Interfaz de Usuario (UI) de alta presentación y personalizables
- Modelo de componentes extensibles
- Licencias Open Source (Código abierto) y comerciales

Se ajusta dentro de este esquema como un motor, que permite crear Ricas Interfaces de Aplicaciones (RIA) mediante JavaScript. Posibilita crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de *layout (capas)* similar al que provee Java Swing; gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno. (Sánchez Rosas, 2008)

### 1.4.4 PHP 5.3

PHP es un acrónimo recursivo que significa *Hypertext Pre-processor*(PHP). El lenguaje fue diseñado para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting), pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas, incluyendo aplicaciones con interfaz gráfica. (King, 2010)

Entre sus principales características están: (King, 2010)

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente, por ser es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de Programación Orientada a Objetos.

### 1.4.5 Symfony 1.4

Symfony es un marco de trabajo completo (o *framework* de desarrollo), diseñado para optimizar el desarrollo de las aplicaciones web gracias a sus características. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Es multiplataforma. (Zaninotto, y otros, 2008)

### 1.4.6 NetBeans IDE 7.3

El IDE Netbeans es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto Netbeans está formado por un IDE de código abierto y una plataforma de aplicación, que permite a los desarrolladores crear con rapidez: aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby onRails, Groovy and Grails y C/C++. (NetBeans, 2013)

### 1.4.7 MySQL 5.0

MySQL es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. (Esepe EStudio, 2005)

### **1.4.8 RUP**

El Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés), es una infraestructura flexible de desarrollo de software, que proporciona prácticas recomendadas probadas y una arquitectura configurable (GSInnova, 2011). RUP es una metodología de desarrollo de software que está basado en componentes e interfaces bien definidas, y junto con el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. (RUP, 2011)

### **1.4.9 Enterprise Architect 7.1**

*Enterprise Architect* (EA) es una herramienta de modelado que utiliza el lenguaje UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. Es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a modelar software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. El UML provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos; EA soporta este proceso en un ambiente fácil de usar, rápido y flexible.

### **Conclusiones del capítulo**

El estudio del estado del arte permitió conocer las características de diferentes sistemas nacionales e internacionales que gestionan trazas, determinando si pueden ser utilizados para darle solución al problema planteado, y el estudio de las tecnologías permitió asimilar las herramientas que se emplearon en el desarrollo del Sistema para el control Farmacológico.

## Capítulo 2: Características de las funcionalidades de gestión de trazas

En este capítulo se describen las características del sistema de seguridad y gestión de trazas para el control farmacológico, así como la propuesta de solución para garantizar la optimización del producto.

### 2.1 Propuesta de solución

En el documento: Arquitectura, normas y tecnologías para el desarrollo de aplicaciones informáticas para la Salud Pública en Cuba (Sánchez Rodríguez, y otros, 2007), se especifican los Requerimientos de Seguridad de los sistemas informáticos en el sector de la salud. Los requerimientos relacionados con las trazas, fueron analizados por sus semejanzas con la investigación en curso. A continuación se detallan los mismos:

- Las trazas del sistema contendrán un texto descriptivo de las acciones realizadas así como el día, mes, año, hora, minuto, segundo en que se registra.
- Cada Petición de usuario a un componente, autorizada o no, será registrada en las trazas del sistema.
- Cada componente, podrá registrar la información que considere deba formar parte de la traza del sistema.

Para el desarrollo de la gestión de trazas del Sistema para el control Farmacológico, se propone la implementación de varias funcionalidades que permitan la gestión de trazas de los usuarios que interactúan con el sistema, cumpliendo con los requerimientos de seguridad anteriormente expuestos. La gestión de trazas permitirá deshacer cambios no deseados en el sistema (recuperar información) y realizar controles de las operaciones realizadas. Además se propone el desarrollo de funcionalidades que permitan limitar los permisos a los Administradores al crear o modificar usuarios de acuerdo a su nivel (Nacional, Provincial, Municipal, Unidad de salud).

La información almacenada de acuerdo a las trazas debe estar dividida en dos grupos: en el primer grupo la información que puede ser recuperada (operaciones: modificar población, eliminar usuario, modificación del plan de recetas médicas, eliminar medicamento del cuadro básico, etc.) y en el segundo la información que puede ser consultada (operaciones: crear

usuario, insertar certificado, insertar población, etc.). Al ejecutarse una acción (por el usuario) en el sistema, este debe generar una traza con la información de la operación realizada. Esta traza debe contener información respecto a: usuario que realiza la acción, operación que se realiza, nombre de la operación realizada, objeto al cual se le realizó la operación, dirección IP, fecha de ejecución y hora de ejecución; con el objetivo de ser consultada.

Los Administradores autenticados en el sistema sólo podrán gestionar trazas de acuerdo a su nivel de permisos. De tal forma que un Administrador con nivel Provincial podrá gestionar solamente las trazas generadas por los usuarios de su nivel o niveles inferiores.

## 2.2 Diagrama del Modelo de Dominio

A continuación se muestra el modelo de dominio:

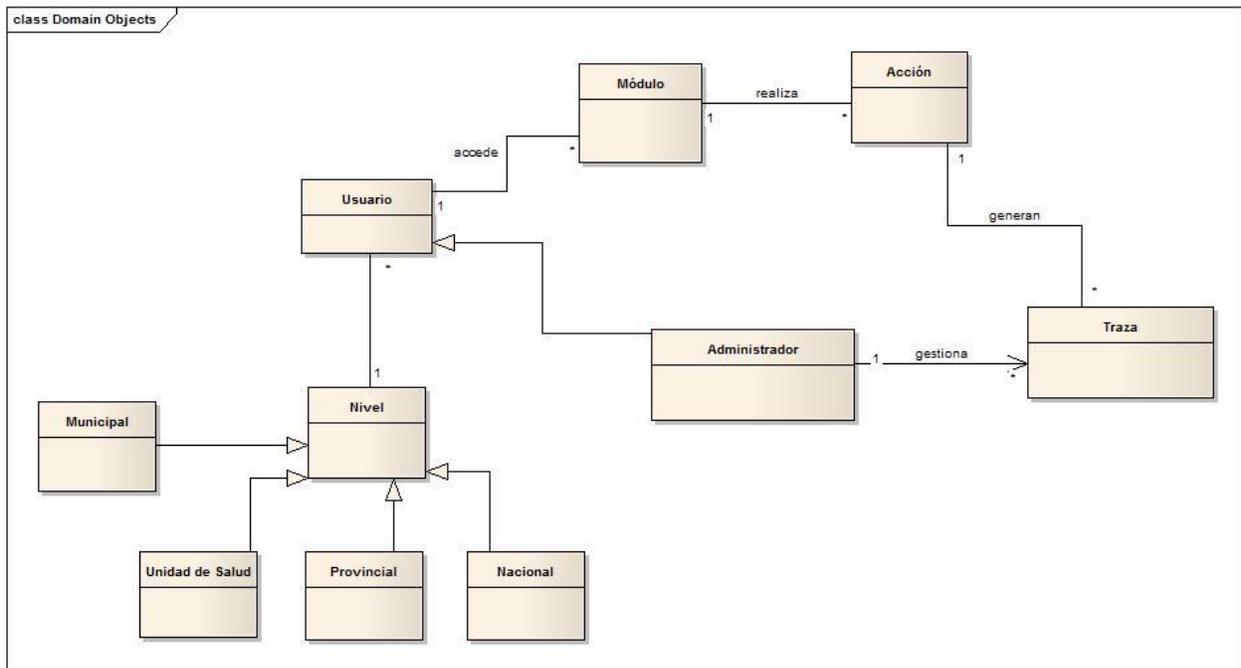


Figura 1. Diagrama de Dominio

### 2.2.1 Conceptos fundamentales

Con el objetivo de lograr un mejor entendimiento del Diagrama de Dominio, se procede a explicar cada uno de los conceptos representados en el negocio.

**Administrador:** representa un usuario autenticado en el sistema que gestiona las trazas.

**Usuario:** representa una persona autenticada en la aplicación que realiza varias acciones en el sistema y esto provoca que se generen trazas.

**Acciones:** son las operaciones que realiza un usuario autenticado en la aplicación.

**Módulo:** Son elementos del sistema que muestran bloques de información en diferentes posiciones o zona de una plantilla.

**Nivel:** son los niveles de dirección administrativa del Sistema de Salud cubano siendo estos Nacional, Provincial, Municipal o Unidad de Salud.

**Nacional:** representa al Ministerio de Salud Pública (MINSAP) como órgano de mando supremo en la nación.

**Provincial:** representa a las Direcciones Provinciales de Salud, se subordinan directamente al nivel Nacional.

**Municipal:** representan las Direcciones Municipales de Salud y su órgano inmediato superior es el Provincial.

**Unidad de Salud:** representa una entidad de salud subordinada a la Dirección Municipal de cada municipio.

**Traza:** representa un registro de las acciones realizadas por los usuarios al autenticarse en la aplicación.

### 2.3 Requerimientos del Software

El proceso que lleva a la construcción de un documento, es conocido como especificación de requerimientos de un software que a su vez es esencial para obtener una solución con la calidad requerida. Los requerimientos del software son una propiedad o restricción que un producto de software debe cumplir. Estos se clasifican en dos grupos: los requisitos funcionales y los requisitos no funcionales.

### 2.3.1 Requerimientos Funcionales (RF)

En la especificación de los requisitos de software, se definen las condiciones o capacidades necesarias para uno o varios usuarios; con el fin de solucionar un problema o conseguir un objetivo. Los requerimientos del software no son más que la propiedad o restricción determinada con precisión, que un producto software debe satisfacer (Sommerville, Ian; 2005) A continuación se representan los diferentes requisitos separados en Requerimientos Funcionales y Requerimientos No Funcionales. Se hace referencia solamente a los requerimientos incorporados al sistema, los demás requerimientos se encuentran descritos en el [expediente del proyecto](#).

Tabla 1. Requerimientos Funcionales de gestión de trazas:

RF1 Recuperar información	RF3 Eliminar traza
RF2 Buscar traza	RF4 Generar traza

A continuación se describen los nuevos requerimientos funcionales:

**RF1 Recuperar información:** permite recuperar información que ha sido eliminada o modificada.

**RF2 Buscar traza:** permite buscar las trazas de las operaciones realizadas en el sistema por un usuario registrado.

**RF3 Eliminar traza:** permite eliminar trazas en el sistema.

**RF4 Generar traza:** se genera una traza en el sistema cuando se ejecuten las operaciones (insertar, modificar, eliminar, crear). Para poder generar las trazas fue necesario modificar algunos requerimientos funcionales de los módulos Tarjeta de Control, Administración, Consumo de Medicamentos, Reacciones Adversas a Medicamentos y Control de Recetas. A continuación se muestra una tabla donde se especifican dichos requerimientos:

Tabla 2. Requerimientos Funcionales del Módulo Administración:

RF Autenticación	RF Eliminar usuario
RF Crear usuario	

RF Modificar usuario	
----------------------	--

Tabla 3. Requerimientos Funcionales del Módulo Control de Recetas:

RF Realizar entrega de recetas	RF Registrar incidencias en la distribución
RF Confirmar recibo	RF Registrar incidencia en farmacia
RF Modificación del plan de entrega de recetas médicas	RF Importar información XML de incidencias

Tabla 4. Requerimientos Funcionales del Módulo Reacciones Adversas a Medicamentos:

RF Registrar planilla por el especialista	RF Insertar paciente
RF Registrar planilla por el paciente	RF Modificar planilla
RF Importar información	

Tabla 5. Requerimientos Funcionales del Módulo Consumo de Medicamentos:

RF Crear cuadro básico	RF Modificar consumos
RF Modificar cuadro básico	RF Importar información
RF Adicionar causas de la adición de medicamentos al cuadro básico	

Tabla 6. Requerimientos Funcionales del Módulo Tarjeta de Control:

RF Eliminar inscripción de medicamento al paciente	RF Insertar población
RF Insertar certificado	RF Importar información XML de certificados
	RF Modificar población

### 2.3.2 Requerimientos no Funcionales (RNF)

Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. (Ingeniería de Software, 2012). A continuación se muestran los RNF:

- **Usabilidad**

**RNF1** Las funcionalidades de gestión de trazas solo podrán ser utilizado por los Administradores definidos, según sus niveles (Nacional, Provincial, Municipal, Unidad de Salud).

**RNF2** El usuario definido como administrador debe tener conocimientos básicos de informática.

- **Seguridad**

Para aplicar las políticas de seguridad se tuvo en cuenta lo expuesto en el documento, Arquitectura, normas y tecnologías para el desarrollo de aplicaciones informáticas para la Salud Pública en Cuba. A continuación se exponen tres requerimientos de dicho documento.

**RNF3** Las trazas del sistema contendrán un texto descriptivo de las acciones realizadas así como el día, mes, año, hora, minuto, segundo en que se registra.

**RNF4** Cada componente, podrá registrar la información que considere deba formar parte de la traza del sistema.

**RNF5** Se harán validaciones de la información tanto en el cliente como en el servidor contra ataques de inyección HTML o SQL.

- **Eficiencia**

**RNF6** El sistema deberá responder de forma rápida ante las solicitudes de los usuarios Administradores, de modo que el tiempo de respuesta deberá ser el menor posible.

- **Software para el servidor**

**RNF7** Se utilizará MySQL Versión 5.0.

**RNF8** Sistema operativo GNU/Linux Debian 4.

**RNF9** Se utilizará el servidor de aplicación Apache en su versión 2.2 o superior para el servidor web.

**RNF10** Para el desarrollo con PHP 5.3 y el *Framework* Symfony 1.4.

**RNF11** Para diseño de las páginas se utilizará CCS, HTML, JavaScript 1.1 y ExtJS 3.3.

- **Software para el cliente**

**RNF12** Sistema operativo Windows o Linux.

**RNF13** Navegador web Mozilla Firefox v.3.6, internet Explorer v. 8.0 o versiones superiores.

- **Interfaz**

**RNF14** Las funcionalidades deben tener una interfaz fácil de usar incluyendo mensajes, cuando la búsqueda no obtenga resultados o cuando existan campos obligatorios vacíos.

**RNF15** Las interfaces de las funcionalidades gestión de trazas deben estar definidas y creadas siguiendo las pautas establecidas por el proyecto Synta. (Ejemplo: íconos, mensajes del sistema, etiquetado).

**Requisitos no Funcionales de Hardware:**

- **Hardware común**

**RNF16** Ordenador Pentium IV.

**RNF17** Se requiere tarjeta de red.

- **Hardware para el cliente**

**RNF18** Computadora con 512 MB de RAM, disco duro de al menos 1 GB libre y tarjeta de red.

- **Hardware para el servidor de base de datos**

**RNF19** Computadora con Microprocesador de velocidad 3.0 GHz, 3 GB de memoria RAM, disco duro de 500 GB y tarjeta de red.

- **Hardware para el servidor de aplicaciones**

**RNF20** Computadora con Microprocesador de velocidad 3.0 GHz, 1 GB de memoria RAM, disco duro de 10 GB y tarjeta de red.

- **Estándares Aplicables**

**RNF21** Para la implementación de las funcionalidades se deberá seguir los estándares de codificación y diseño definidos por el proyecto Synta.

## 2.4 Actores del Sistema

Un actor es una entidad externa al sistema que se modela y que puede interactuar con él. Puede ser una persona o un grupo de personas homogéneas, otro sistema, o una máquina. Los actores son externos al sistema que se va a desarrollar. Por lo tanto, al identificarlos, estamos comenzando a delimitar el sistema y a definir su alcance. (Joooa, 2010)

A continuación se describen los actores del sistema:

Tabla 7. Descripción de los actores del sistema

Actor	Descripción
Administrador	Representa un usuario autenticado en el sistema que gestiona las trazas
Usuario	Usuario del sistema con privilegios para gestionar información según el módulo al que pertenezca

## 2.5 Casos de Uso

A continuación se muestran los casos de uso que generan trazas separados por módulos. La descripción de los mismos se encuentra en el expediente del proyecto. Además se encuentra en la tabla una relación de los casos de uso agregados al sistema.

Tabla 8. Casos de Uso del Sistema

Casos de uso del Sistema	
Módulo Administración	Módulo RAM
CUS1 Autenticar	CUS15 Modificar planillas
CUS2 Gestionar usuario	CUS16 Registrar planillas por el paciente
	CUS17 Importar información

	CUS18 Registrar planilla por el especialista
	CUS19 Insertar paciente
<b>Módulo Control de Recetas</b>	<b>Módulo Consumo de Medicamentos</b>
CUS3 Entrega de recetas	CUS20 Modificar consumo
CUS4 Modificación de plan de entregas	CUS21 Modificar cuadro básico
CUS5 Confirmar recibo	CUS22 Crear cuadro básico
CUS6 Importar información XML de incidencias	CUS23 Gestionar causas de modificaciones
CUS7 Registrar incidencias de distribución	CUS24 Importar información
CUS8 Registrar incidencias en la farmacia	
<b>Módulo Tarjeta de Control</b>	<b>Nuevos Casos de Uso</b>
CUS9 Insertar población	CUS25 Generar traza
CUS10 Modificar población	CUS26 Eliminar traza
CUS11 Buscar medicamentos del paciente	CUS27 Buscar traza
CUS12 Eliminar inscripción del medicamento	CUS28 Recuperar información
CUS13 Importar XML de certificados	
CUS14 Insertar certificados	

## 2.6 Diagrama de Casos de Uso del Sistema

Los diagramas de caso de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. (Cáceres Tello)

A continuación se muestra el diagrama de Casos de Uso. Se describen los casos de uso de la funcionalidad de gestión de trazas en el módulo Administración.

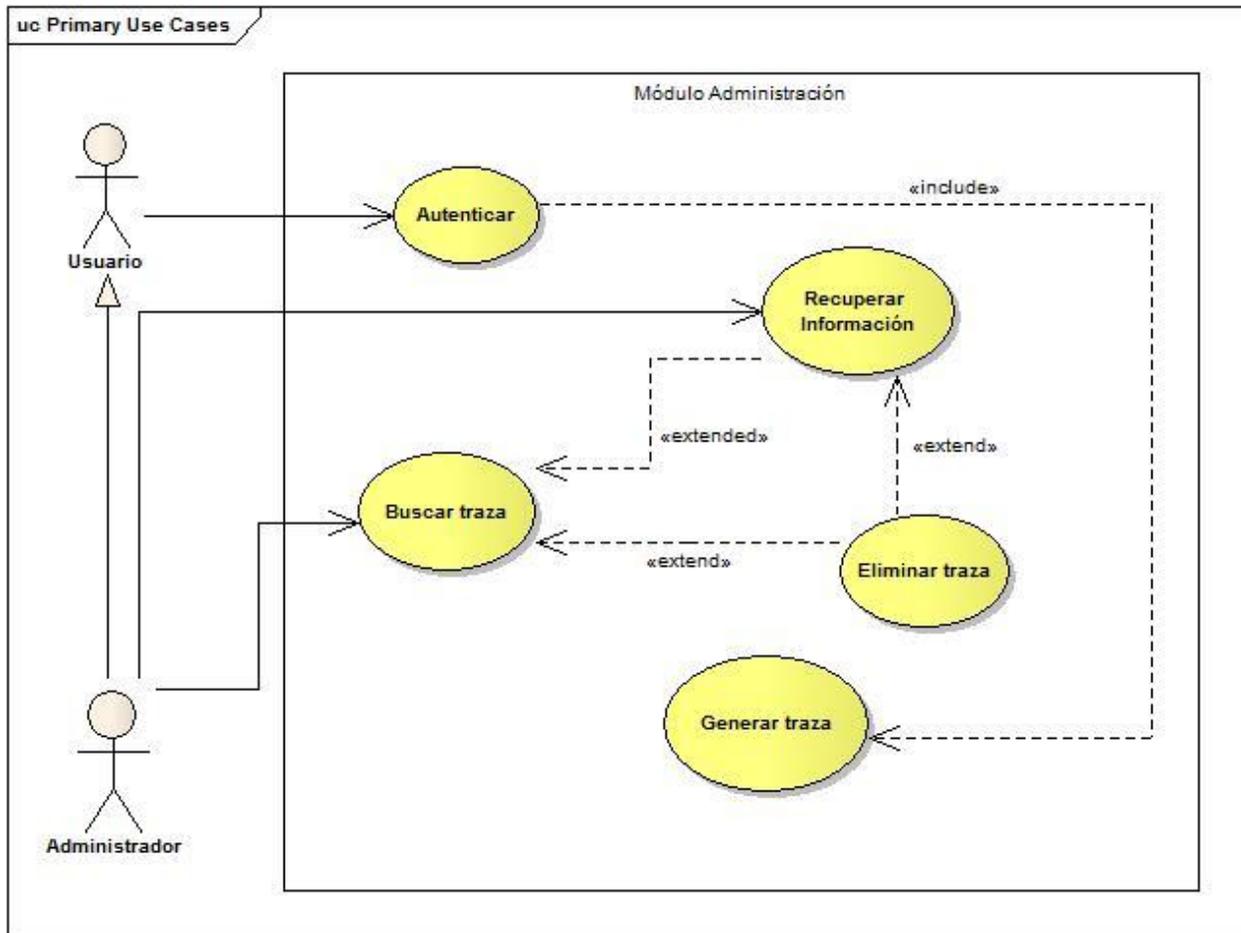


Figura 2. Diagrama de Casos de Uso del Sistema

## 2.7 Descripción textual de Casos de Uso del Sistema

A continuación se realiza la descripción de dos casos de usos. Las descripciones restantes serán mostradas en el expediente de proyecto.

Tabla 9. Descripción textual CUS Recuperar Información

<b>Objetivo</b>	Permite recuperar información referente a una traza en el sistema.
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el actor selecciona una traza y accede a la opción Recuperar, el sistema recupera la información de la traza, el caso de uso termina.
<b>Complejidad</b>	Baja
<b>Prioridad</b>	Alta

<b>Precondiciones</b>	Debe estar creada la traza en el sistema.  Debe seleccionar la traza que se desea recuperar la información asociada.	
<b>Postcondiciones</b>	Información recuperada satisfactoriamente.	
<b>Requisitos Funcionales</b>	RF1	
<b>Flujo de eventos</b>		
<b>Flujo básico Recuperar traza</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El caso de uso inicia cuando el Administrador selecciona la traza y presiona el botón Recuperar.	El sistema muestra el mensaje. “Estás seguro que desea restaurar la información de las trazas seleccionadas.”  Cancelar operación. Ver <b>Alternativa 1.</b>
2.	Selecciona la opción Si	Se recupera la información de las trazas seleccionadas y se eliminan las trazas del sistema.
3.		El caso de uso termina.
<b>Flujos alternos</b>		
<b>Nº 1 Cancelar operación</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción No	El sistema vuelve al estado anterior.
<b>Prototipo no funcional</b>		

👤 Gestionar usuario 📄 Gestionar reportes 📄 Gestionar trazas

🏠 Inicio 🔄 Recuperar información

**Filtros de búsqueda**

Módulo: \* Reacciones adversas a medicamen Operación: \* Modificar planilla RAM Usuario:

Fecha de modificación:

---

**Operaciones realizadas**

Objeto	Fecha	Hora	Usuario
Planilla registrada al paciente:Alberto Fr...	2014-05-16	13:05:50	giset
Planilla registrada al paciente:Caridad E...	2014-05-20	22:05:40	giset
Planilla registrada al paciente:Caridad E...	2014-05-21	08:05:13	giset
Planilla registrada al paciente:Miriam Ra...	2014-05-21	14:05:08	giset
Planilla registrada al paciente:Miriam Ra...	2014-05-21	14:05:54	giset

Página 1 de 1 Mostrando 1 - 5 de 5

---

**Detalles de los campos modificados**

**Nombre del campo:** Sospechoso.Medicamento:DIFENHIDRAMINA 25 MG TABLETAS.  
**Valor antiguo:**No  
**Valor actual:** Si

---

**Nombre del campo:** Medicamento  
**Valor antiguo:**ACICLOVIR 250 MG POLVO ESTERIL.  
**Valor antiguo:**ACICLOVIR 250 MG POLVO ESTERIL.

---

👤 Gestionar usuario 📄 Gestionar reportes 📄 Gestionar trazas

🏠 Inicio 🔄 Recuperar información

Reacciones adversas a medicamen Modificar planilla RAM

Fecha de modificación:

**Información**

¿Está seguro que desea restaurar la información de las trazas seleccionadas?

---

**Operaciones realizadas**

Objeto	Fecha	Hora	Usuario
Planilla registrada al paciente:Alberto Fr...	2014-05-16	13:05:50	giset
Planilla registrada al paciente:Caridad E...	2014-05-20	22:05:40	giset
Planilla registrada al paciente:Caridad E...	2014-05-21	08:05:13	giset
Planilla registrada al paciente:Miriam Ra...	2014-05-21	14:05:08	giset
Planilla registrada al paciente:Miriam Ra...	2014-05-21	14:05:54	giset

Página 1 de 1 Mostrando 1 - 5 de 5

---

**Detalles de los campos modificados**

**Nombre del campo:** Sospechoso.Medicamento:DIFENHIDRAMINA 25 MG TABLETAS.  
**Valor antiguo:**No  
**Valor actual:** Si

---

**Nombre del campo:** Medicamento  
**Valor antiguo:**ACICLOVIR 250 MG POLVO ESTERIL.  
**Valor actual:** null

© 2010-2014 Universidad de las Ciencias Informáticas. Todos los derechos reservados.

<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	CUS26 Eliminar traza
<b>Requisitos funcionales</b>	<b>no</b>	
<b>Asuntos pendientes</b>		

Tabla 10. Descripción textual CUS Buscar Traza

<b>Objetivo</b>	Permitir buscar una traza en el sistema.	
<b>Actores</b>	Administrador	
<b>Resumen</b>	El caso de uso inicia cuando el actor accede a la opción Buscar traza, el sistema brinda la posibilidad de introducir criterios de búsqueda para localizar la traza, el actor introduce los datos que considera como criterios para realizar una búsqueda, el sistema busca y muestra las trazas que cumplen con los criterios de búsqueda, el caso de uso termina.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	Deben existir trazas en el sistema creadas para poder realizar la búsqueda.	
<b>Postcondiciones</b>	Se busca las trazas correctamente.	
<b>Requisitos Funcionales</b>	RF2	
<b>Flujo de eventos</b>		
<b>Flujo básico Buscar Traza</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El caso de uso inicia cuando el Administrador accede a la opción Buscar trazas	Brinda la posibilidad de introducir los criterios elementales de búsqueda: <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Módulo</li> <li>• Operación</li> <li>• Fecha de inicio de la operación</li> <li>• Fecha de fin de la operación</li> </ul>
2.	El usuario introduce los parámetros obligatorios: <ul style="list-style-type: none"> <li>• Módulo</li> </ul>	Busca la traza dado criterios. Si se desea cancelar la operación ver <b>Alternativa 1</b> : “Cancelar operación”  Si el sistema ni tiene valores a mostrar ver <b>Alternativa 2</b> : “La búsqueda no obtuvo

		resultados”  Si el Administrador deja campos vacíos obligatorios ver <b>Alternativa 3</b> : “Campos vacíos”
3.		Muestra un listado de trazas que cumplen con los criterios de búsqueda, mostrando los siguientes atributos: <ul style="list-style-type: none"> <li>• Usuario</li> <li>• Objeto</li> <li>• Operación</li> <li>• Dirección</li> <li>• Fecha de ejecución</li> <li>• Hora de ejecución</li> </ul> Ordenados ascendentemente o descendentemente por el Dato 1, hasta Dato n mostrando la cantidad páginas de información, permitiendo navegar por el resultado.
4.		El caso de uso termina
<b>Flujos alternos</b>		
<b>Nº 1</b> Cancelar operación		
	<b>Actor</b>	<b>Sistema</b>
2.	Presiona el botón Cancelar.	El sistema vuelve al estado anterior.
<b>Nº 2</b> No existe información para la búsqueda		
	<b>Actor</b>	<b>Sistema</b>
2.		El sistema muestra un mensaje: “La búsqueda no obtuvo resultados”
<b>Nº 3</b> Campos vacíos		
	<b>Actor</b>	<b>Sistema</b>
2.	Deja en blanco los valores de los campos obligatorios en el sistema.	El sistema señala los campos vacíos con un * y muestra un mensaje: “Existen campos obligatorios vacíos”.
<b>Prototipo no Funcional</b>		

[Gestionar usuario](#) ▾ [Gestionar reportes](#) ▾ [Gestionar trazas](#) ▾

[Inicio](#) [Buscar operaciones realizadas](#)

**Filtros de búsqueda**

Modulo: \*  Operación:  Usuario:

Fecha de inicio operación:

Fecha de fin operación:

**Información de acciones**

Usuario	Operación	Objeto	Dirección	Fecha de ejecución	Hora de ejecución
giset	Modificar planilla RAM	Planilla registrada al paciente:Alberto Fr...	127.0.0.1	2014-05-16	13:05:50
giset	Modificar planilla RAM	Planilla registrada al paciente:Caridad E...	127.0.0.1	2014-05-20	22:05:40
giset	Modificar planilla RAM	Planilla registrada al paciente:Caridad E...	127.0.0.1	2014-05-21	08:05:13
giset	Modificar planilla RAM	Planilla registrada al paciente:Miriam Ra...	127.0.0.1	2014-05-21	14:05:08
giset	Modificar planilla RAM	Planilla registrada al paciente:Miriam Ra...	127.0.0.1	2014-05-21	14:05:54

Página 1 de 1 Mostrando 1 - 5 de 5

 **Adrian Rodriguez Ronquillo**  
 Módulo Administración MINSAP

[Inicio](#) [Estilo](#) [Clave](#) [Ayuda](#) [Salir](#)

[Gestionar usuario](#) ▾ [Gestionar reportes](#) ▾ [Gestionar trazas](#) ▾

[Inicio](#) [Buscar operaciones realizadas](#)

**Filtros de búsqueda**

Modulo: \*  Operación:  Usuario:

Fecha de inicio operación:

Fecha de fin operación:

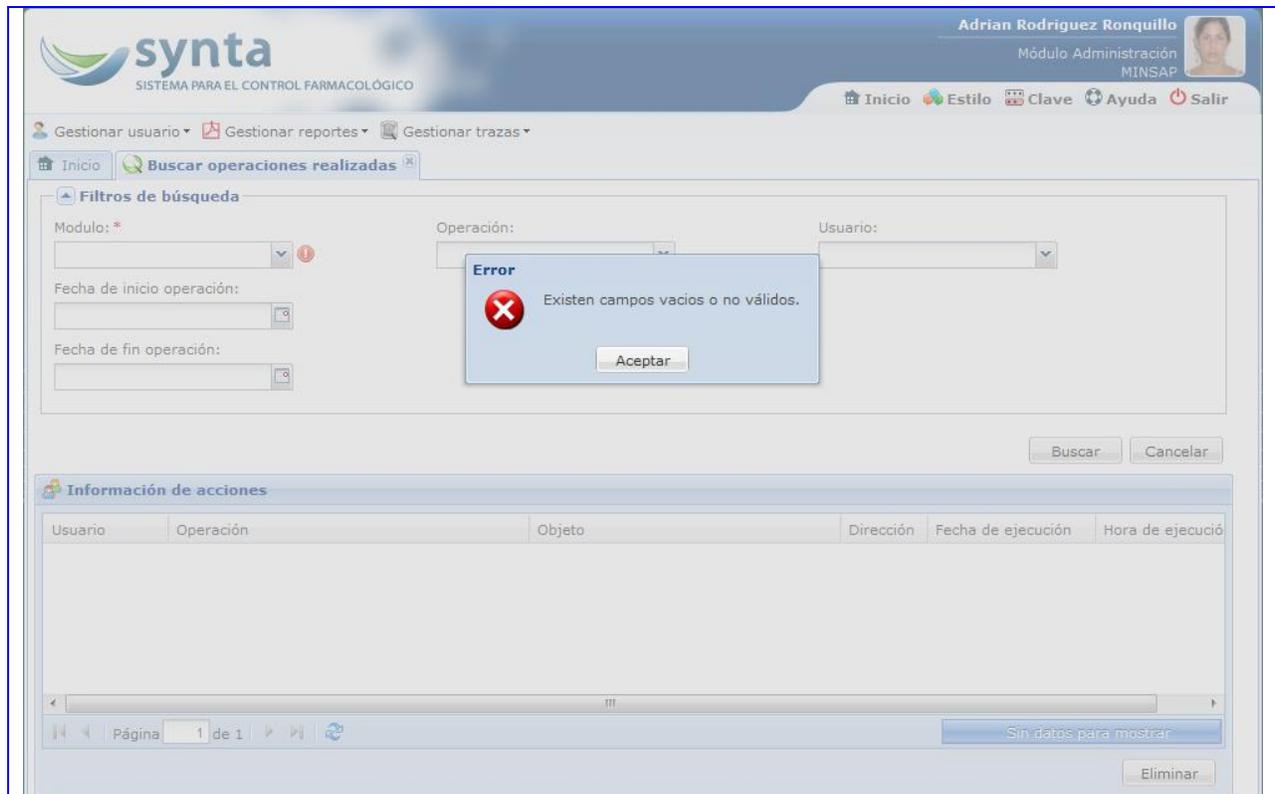
**Información**

La búsqueda no obtuvo resultados.

**Información de acciones**

Usuario	Operación	Objeto	Dirección	Fecha de ejecución	Hora de ejecución
Sin datos para mostrar					

Página 1 de 1 Sin datos para mostrar

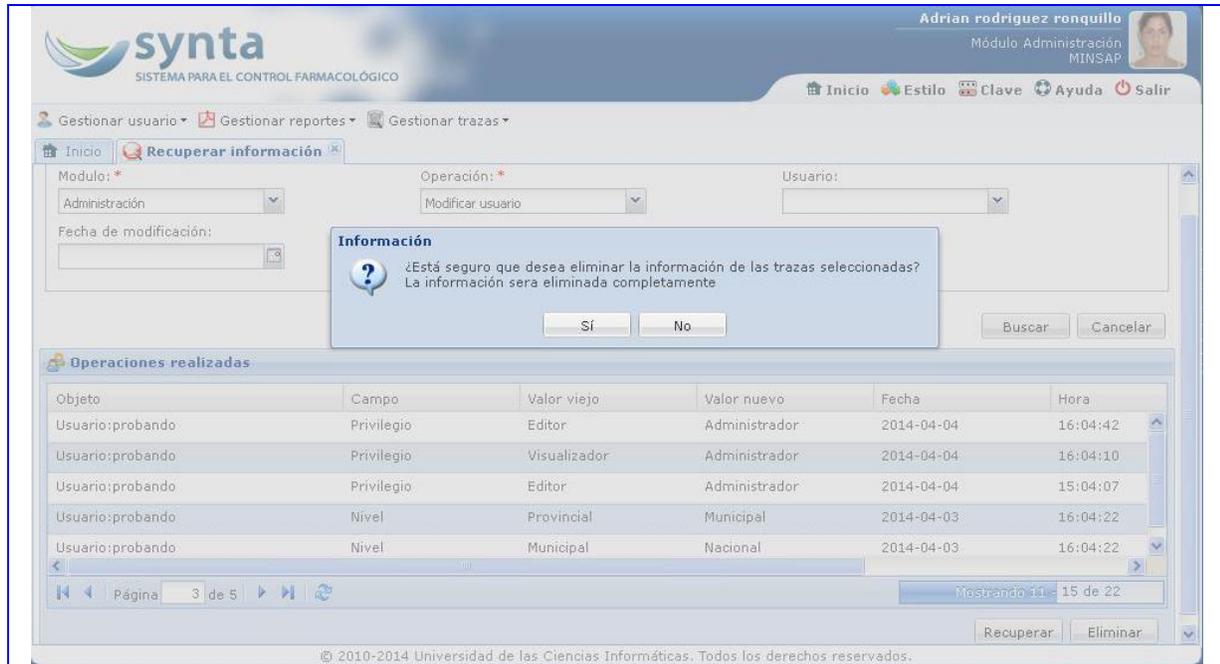


<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	CUS26 Eliminar traza, CUS28 Recuperar Información
<b>Requisitos funcionales</b>	<b>no</b>	
<b>Asuntos pendientes</b>		

Tabla 11. Descripción textual CUS Eliminar Trazas

<b>Objetivo</b>	Permitir eliminar una traza en el sistema
<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el actor selecciona una traza y accede a la opción Eliminar traza, el sistema elimina la traza, el caso de uso termina
<b>Complejidad</b>	Baja
<b>Prioridad</b>	Alta
<b>Precondiciones</b>	Debe estar creada la traza en el sistema.  Debe seleccionar la traza que se desea eliminar.
<b>Postcondiciones</b>	Se elimina la traza correctamente.

<b>Requisitos Funcionales</b>	RF3	
<b>Flujo de eventos</b>		
<b>Flujo básico Eliminar traza</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	El caso de uso inicia cuando el Administrador selecciona la traza y presiona el botón eliminar.	Muestra el mensaje de advertencia “Desea eliminar la información referente a las trazas seleccionadas. La información será eliminada completamente.” y permite presionar el botón: <ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
2.	El Administrador selecciona la opción deseada.	Si desea la operación No, ver <b>Alternativa 1:</b> “Cancelar operación”  Si presiona el botón Si, el sistema elimina la traza.
3.		El caso de uso termina.
<b>Flujos alternos</b>		
<b>Nº 1 Cancelar operación</b>		
	<b>Actor</b>	<b>Sistema</b>
2.	Presiona el botón No.	El sistema vuelve al estado anterior.
<b>Prototipo no funcional</b>		



<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Asuntos pendientes</b>		

### Conclusiones del Capítulo

En este capítulo se realizó la propuesta de solución de gestión de trazas en el Sistema para el control Farmacológico (Synta) y se realizó el modelo de dominio con el objetivo de expresar con conceptos del negocio las relaciones entre las clases. Se definieron los requisitos funcionales y no funcionales, así como la modelación del diagrama de caso de uso del sistema, diferenciando las nuevas funcionalidades.

## **Capítulo 3: Diseño e implementación de las funcionalidades gestionar trazas del Sistema para el control Farmacológico.**

En este capítulo se justifican los patrones a usar en el diseño de las funcionalidades, se define la estructura y elementos del diseño, mostrando los diagramas de clases de diseño e implementación.

### **3.1 Descripción de la arquitectura**

Para el desarrollo de este complemento que gestiona trazas en los módulos Administración, Tarjeta de Control, Consumo de Medicamentos y Reacciones Adversas a Medicamentos del Sistema para el control Farmacológico (Synta) fue necesario asumir pautas, software y estándares predefinidos por el proyecto Synta. Ejemplo de ello es la utilización como marco de desarrollo Symfony en su versión 1.4.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles (Potencier, y otros)

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (Potencier, y otros)

A continuación se muestra la estructura del MVC de Symfony:

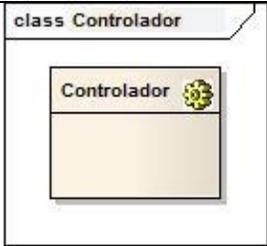
- La capa del Modelo
  - Abstracción de la base de datos
  - Acceso a los datos
- La capa de la Vista
  - Vista
  - Plantilla
  - Layout
- La capa del Controlador
  - Controlador frontal
  - Acción

En el [Anexo 2](#) se muestra un diagrama que representa el patrón MVC en Symfony.

### 3.2 Definición de elementos del diseño

El sistema es una aplicación web que se modeló utilizando elementos del diseño como las clases UML estereotipadas “Client Page”, “Server Page” y “Form”, utilizadas para el código cliente, código servidor y código del formulario, permitiendo representar ficheros contenedores de sentencias script como por ejemplo PHP y JavaScript.

Tabla 12. Clases del Diseño estereotipadas

Estereotipos para las clases del Diseño	
Estereotipo	Descripción
	<b>Server Page:</b> Representa una página web que tiene scripts que son ejecutados por el servidor. Estos scripts interactúan con recursos del servidor como bases de datos, lógica de negocio, sistemas externos y se encarga de construir (build) o generar el resultado HTML.

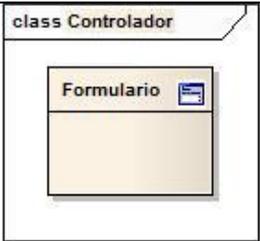
	<p><b>Client Page:</b> Es una página web con formato HTML en la cual se presentan los datos. Las páginas clientes son representadas por los navegadores clientes, y pueden contener scripts que son interpretados por el navegador.</p>
	<p><b>Formulario:</b> Es una colección de campos de entrada que forman parte de una página cliente. Los formularios envían sus datos al servidor para ser procesados los pedidos (submit).</p>

Tabla 13. Estereotipos para la relación entre las clases

Estereotipos para la relación entre las clases	
link	Representa un enlace desde una página cliente (client page) hacia otra página cliente o una página servidora (server page).
submit	Se establece el "Submit" para conectar una página formulario (form) y una página servidora (server page), donde esta procesa los datos que envía la página formulario.
build	Ocurre cuando la página servidora crea una página cliente. La servidora puede crear varias páginas clientes, pero esta última sólo puede ser creada por una página servidora.
redirect	Es una relación que indica que una página web reorienta el procesamiento a otra página.

### 3.3 Diagramas de Clases del diseño

Diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en un aplicación, contiene información como. Clases, asociaciones y atributos, interfaces con sus operaciones y constantes, métodos, navegabilidad, dependencias. (Bravo Llanqui, y otros)

A continuación se muestran algunos de los Diagramas de Clases del Diseño de los casos de uso incorporados al sistema, en el expediente de proyecto se encuentran detalladamente dichos diagramas.

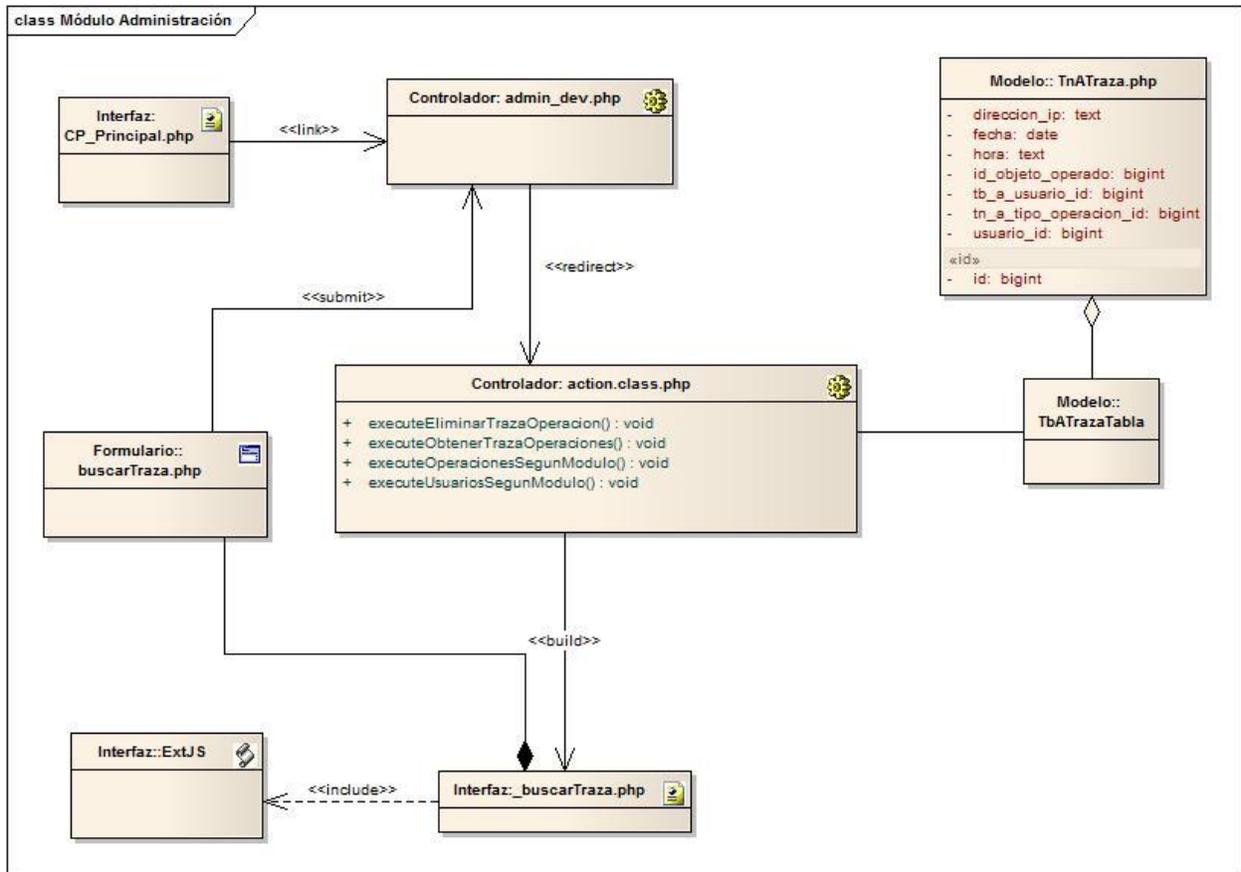


Figura 3. Diagrama de Clases del Diseño: DD\_Buscar Traza

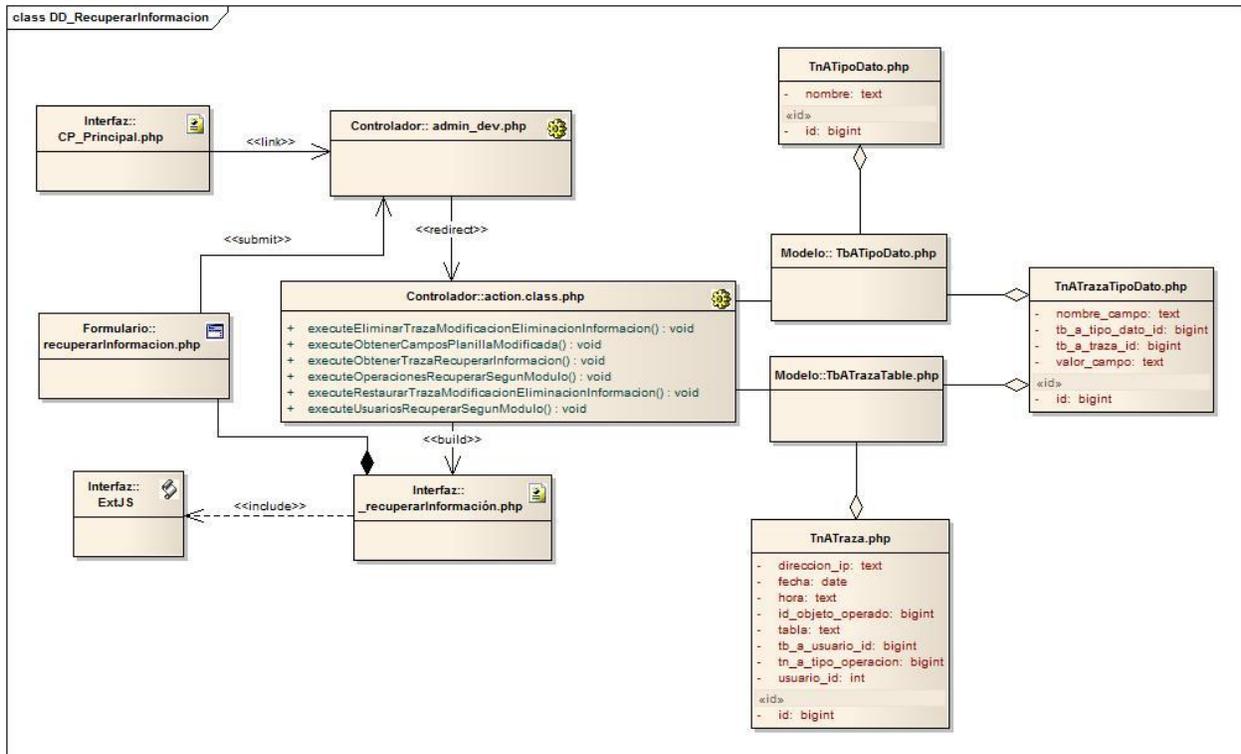


Figura 4. Diagrama de Clases del Diseño: DD\_Recuperar Información

### 3.3.1 Descripción de las clases

A continuación se describen algunas clases que han sido identificadas para la posterior implementación del sistema. De esta manera se poseerá un mejor entendimiento sobre el funcionamiento del mismo.

Tabla 14. Descripción de la clase admin\_dev.php

<b>Nombre</b>	admin_dev.php
<b>Tipo</b>	Página servidora
<b>Descripción</b>	El controlador frontal es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse. Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Tabla 15. Descripción de la clase ExtJS.php

<b>Nombre</b>	ExtJS.php
<b>Tipo</b>	Librería
<b>Descripción</b>	Es utilizada para la creación de las interfaces de usuario. Permite crear interfaces de usuario dinámicas y organizadas, capaces de brindar todas las funcionalidades necesarias para la gestión de la información.

Tabla 16. Descripción de la clase TbNombreTabla.php

<b>Nombre</b>	TbNombreTabla.php
<b>Tipo</b>	Acceso a datos
<b>Descripción</b>	Esta clase hereda todos los métodos de las clases bases y no les afectan las modificaciones en el esquema, permite empezar a programar desde el primer momento. La estructura de archivos creada permite personalizar y evolucionar el modelo. Es una clase objeto que representan un registro de la base de datos. Permiten acceder a las columnas de un registro y a los registros relacionados.

Tabla 17. Descripción de la clase TbNombreTablaTable.php

<b>Nombre</b>	TbNombreTablaTable.php
<b>Tipo</b>	Acceso a datos
<b>Descripción</b>	Esta clase hereda todos los métodos de las clases bases y no les afectan las modificaciones en el esquema, permite empezar a programar desde el primer momento. La estructura de archivos creada permite personalizar y evolucionar el modelo. Tienen métodos estáticos para trabajar con las tablas de la base de datos. Proporcionan los medios necesarios para obtener los registros de las tablas. Sus métodos devuelven normalmente un objeto o una colección de objetos de la clase objeto relacionado.

Tabla 18. Descripción del fichero action.class.php

<b>Nombre</b>	action.class.php
<b>Tipo</b>	Página servidora

<b>Descripción</b>	Este fichero contiene toda la lógica del módulo al que pertenece. Las acciones son métodos de una clase que utilizan el modelo y definen variables para la vista. Cuando se realiza una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición.
--------------------	---

### 3.4 Modelo de datos

Marta E. Zorrilla Pantaleón, profesora de la Universidad de Cantabria expresó el concepto de modelo de datos que se relaciona a continuación: Colección de herramientas conceptuales que se emplean para especificar datos, las relaciones entre ellos, su semántica asociada y las restricciones de integridad. (Zorrilla Pantaleón, 2011)

A continuación se muestra el modelo de datos para lograr mayor entendimiento de las relaciones de las tablas en la base de datos.

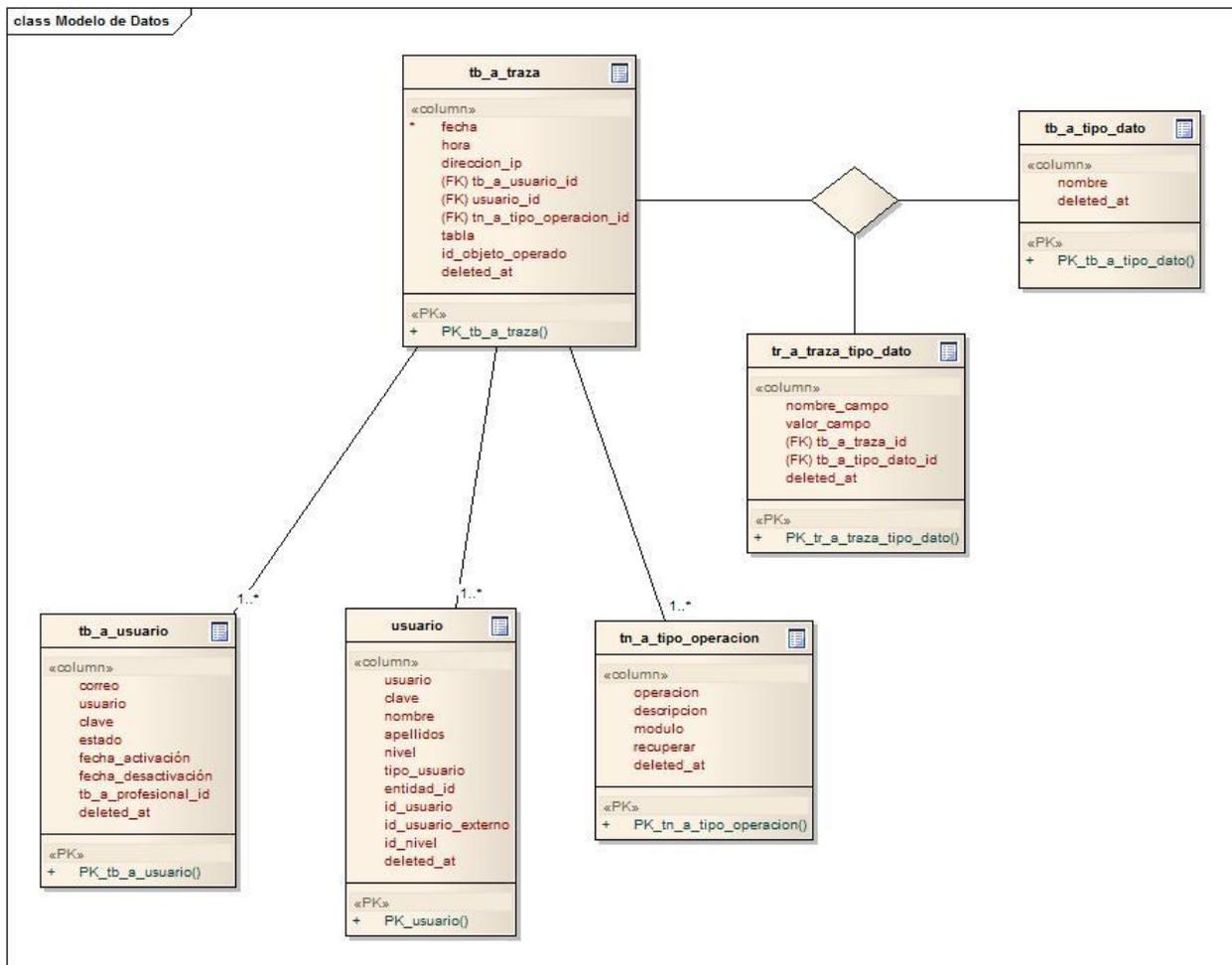


Figura 5. Diagrama de modelo de datos

A continuación se describen algunas tablas de la base de datos:

Tabla 19. Descripción de tb\_a\_traza

Nombre: tb_a_traza			
<b>Descripción:</b> Almacena información referente a los acciones que se realizan en el sistema y los usuarios involucrados, manteniendo un registro para posibles auditorías.			
Atributo	Tipo	Nulo	Descripción
id_traza (PK)	integer	No	Identificador único para la traza
fecha	date	Si	Fecha en que se realizó la acción
hora	character(variable)	Si	Hora en que se realizó la acción
dirección_ip	character(variable)	No	Dirección IP del usuario que realizó la acción
tb_a_usuario_id(FK)	integer	Si	Identificador asociado al usuario que realiza la acción, perteneciendo este usuario a los módulos RAM, Tarjeta de Control, Administración y Consumo de Medicamentos, referente a la tabla tb_a_usuario del esquema
usuario_id (FK)	integer	Si	Identificador asociado al usuario que realiza la acción, perteneciendo este usuario al módulo Control de Recetas, referente a la tabla usuario del esquema
tn_a_tipo_operacion_id(FK)	integer	No	Identificador para representar el tipo de operación, referente a la tabla tn_a_tipo_operacion del esquema
tabla	character(variable)	No	Especifica la tabla del esquema donde se realizó la operación
id_objeto_operado	integer	No	Especifica el id del objeto al cual se le

			realiza la acción
--	--	--	-------------------

Tabla 20. Descripción de tr\_a\_traza\_tipo\_dato

<b>Nombre: tr_a_traza_tipo_dato</b>			
<b>Descripción:</b> Almacena información que permite recopilar los registros de las trazas sobre las que se realizan recuperación de información de los módulos. Registra información descriptiva como el nombre_campo y valor modificado.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_traza_tipo_dato(PK)	integer	No	Identificador único para la traza_tipo_dato
nombre_campo	character(variable)	No	Nombre del campo de la tabla donde se realizó la modificación
valor_campo	character(variable)	No	Valor del campo modificado
tb_a_tipo_dato_id (FK)	integer	No	Tipo de dato del valor modificado, referente a la tabla tb_a_tipo_dato del esquema
tb_a_traza_id	integer	No	Identificador asociado a la traza, referente a la tabla tb_a_traza del esquema

Tabla 21. Descripción de tn\_a\_tipo\_operacion

<b>Nombre: tn_a_tipo_operacion</b>			
<b>Descripción:</b> Almacena información referente a la operación realizada por el usuario.			
<b>Atributo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Descripción</b>
id_tn_a_tipo_operacion(PK)	integer	No	Identificador único para tn_a_tipo_operacion

operación	character(variable)	No	Nombre de la operación realizada
descripción	character(variable)	No	Descripción de la operación realizada
módulo	character(variable)	No	Módulo donde se realizó la operación
recuperar	bool	No	Describe si la operación es de modificación o eliminación

### 3.1 Descripción de la funcionalidad gestión de trazas.

#### 3.4.1 Gestión de Trazas

La gestión de trazas permite la recuperación de la información modificada (acciones: modificar población, modificar permisos, etc.) y eliminada (acciones: eliminar usuario, eliminar inscripción de medicamentos, etc.). Para poder implementar esta funcionalidad en el sistema, los desarrolladores se basan en el comportamiento SoftDelete del ORM Doctrine.

En la implementación de la solución para recuperar información eliminada se agrega la columna (`delete_at`) y almacena la fecha en que se realiza la eliminación. Este procedimiento es aplicado en todas las tablas relacionadas con la información eliminada, de esta forma al eliminar un elemento en el sistema este no se elimina físicamente sino que la columna (`delete_at`) que se encontraba con valor null se actualiza con la fecha en que se realizó la operación.

En la traza de la información eliminada se registra el identificador del elemento eliminado, además del proceso de recuperación de información que consiste en restaurar el valor del campo (`delete_at`) a "null", en la tupla que se corresponde con el identificador del elemento.

En la implementación de la solución para recuperar información modificada, los campos del elemento modificado son (o pueden ser) restaurados ya que en la traza de información modificada se registra el identificador del elemento transformado, el campo de la tabla y el valor que se modificó, donde el proceso de recuperación consiste en obtener dicho elemento y restaurar su valor.

### 3.2 Diagrama de Despliegue

A continuación se muestra el Diagrama de Despliegue

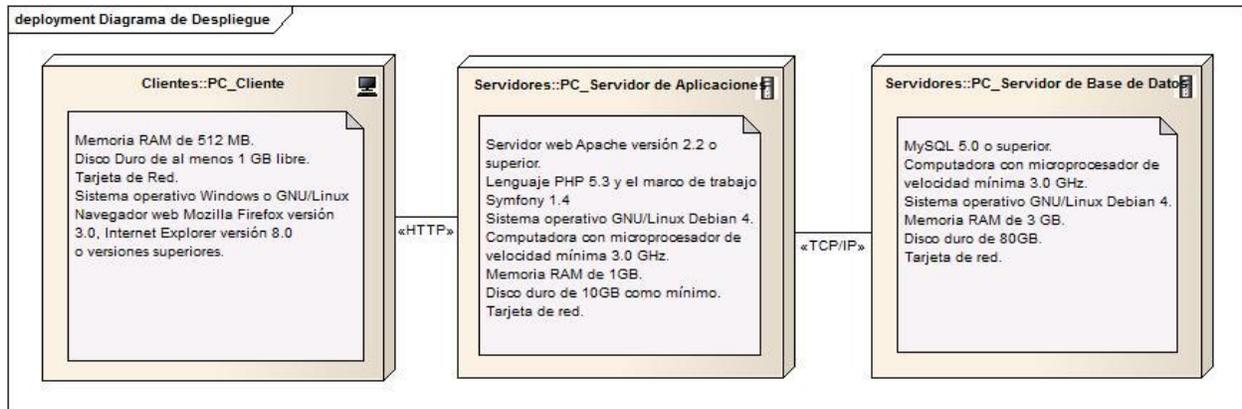


Figura 6: Diagrama de despliegue

### 3.3 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, se implementan en términos de componentes, como ficheros de códigos de fuentes y ejecutables. El modelo de implementación describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes uso de otros. (Jacobson, 2004)

### 3.4 Diagrama de componentes

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. (Microsoft, 2014)

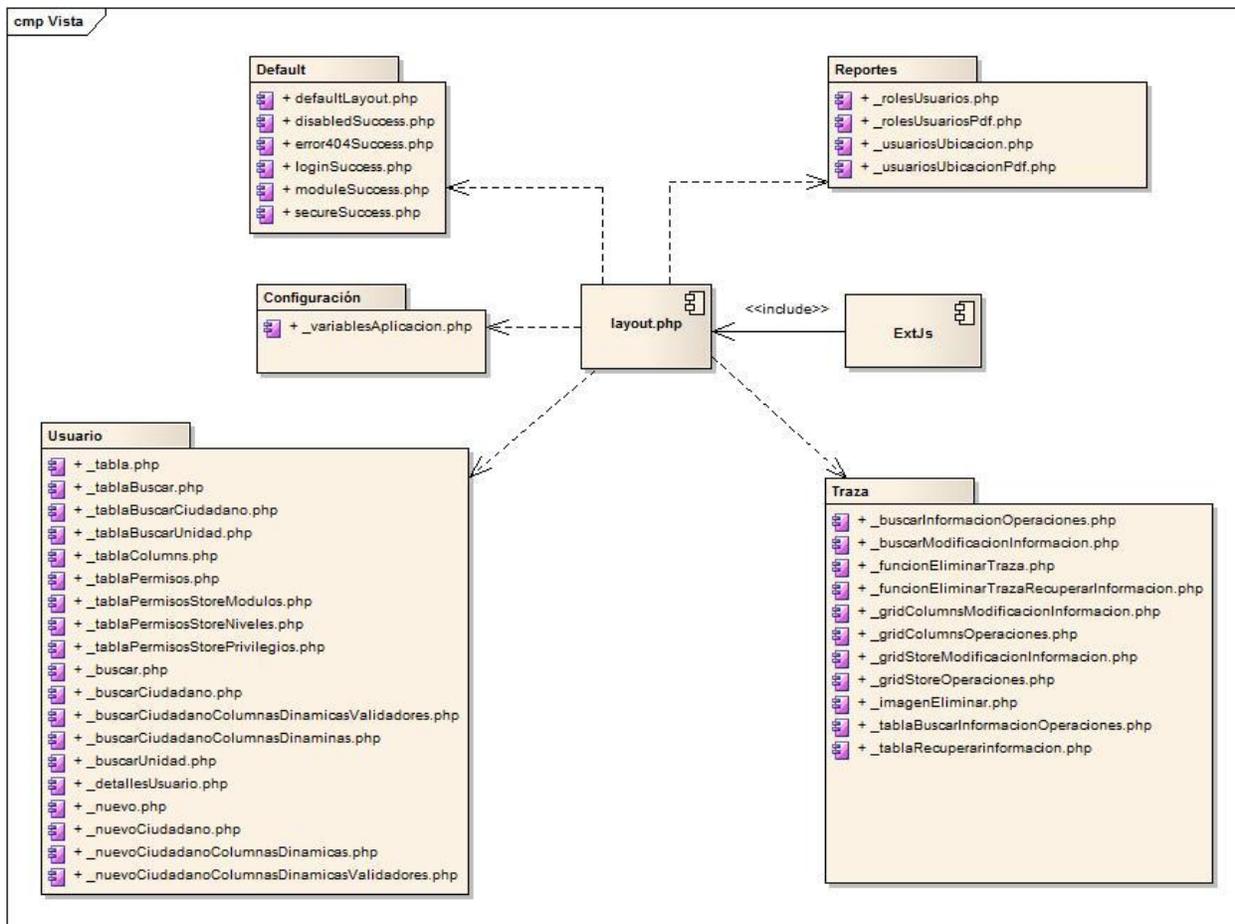


Figura 7. Diagrama de componentes de la vista

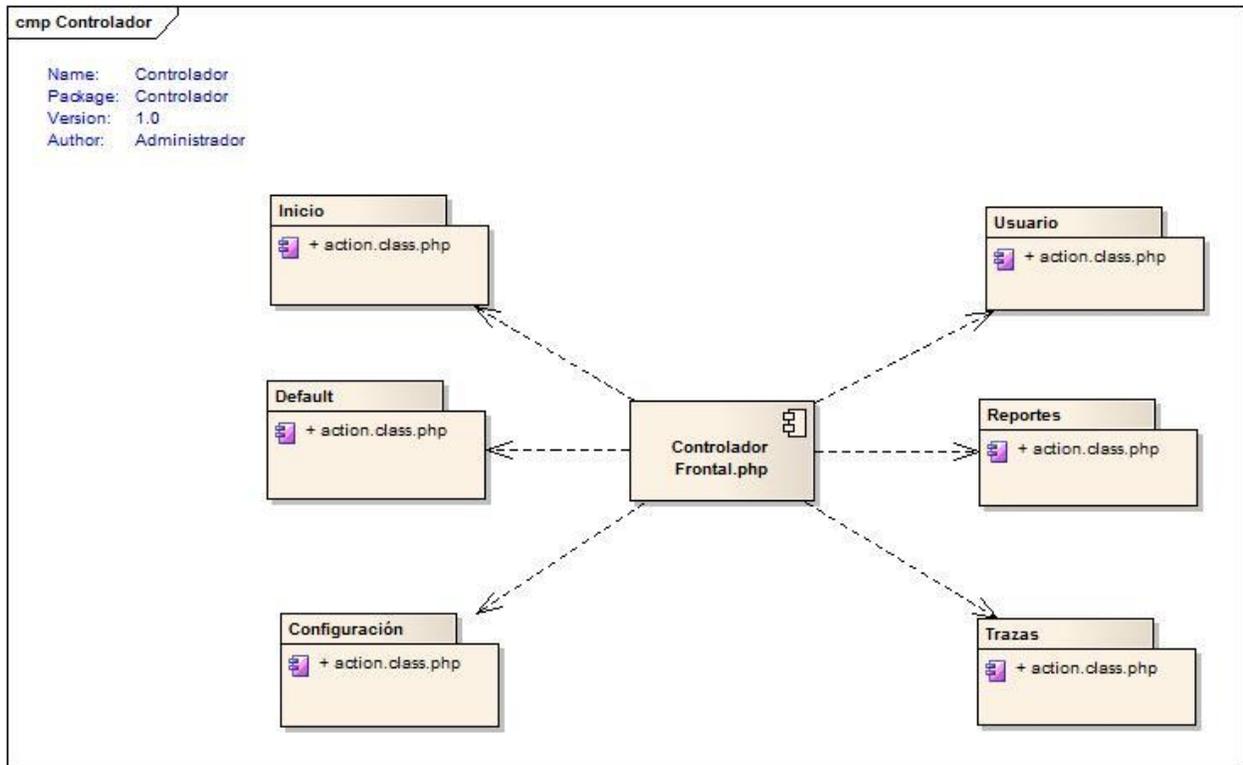


Figura 8. Diagrama de componentes del controlador

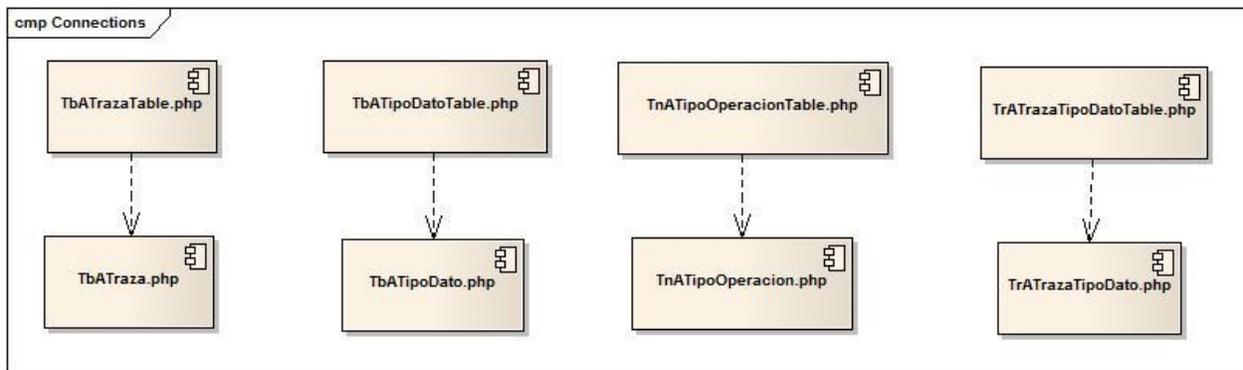


Figura 9. Diagrama de componentes del modelo

A continuación serán expuestas algunas de las clases más importantes adicionadas al sistema, para lograr un mejor entendimiento sobre el funcionamiento del mismo.

### Clase controladora perteneciente a la clase Traza

Tabla 22. Descripción de la clase actions.class.php de la clase traza

**Nombre: actions.class.php**

**Descripción:** Esta clase es la encargada de gestionar toda la información relacionada con las trazas. Entre sus tareas se encuentra mostrar los diferentes tipos de trazas y realizar la recuperación o eliminación de la información referente a cada traza. Esta clase tiene gran importancia ya que a través de la gestión de trazas, se logra mantener un registro de las acciones realizadas en el sistema y en caso necesario la recuperación de esta información.

Al modelo se adicionan dos clases auxiliares que contienen las funcionalidades para la gestión de trazas y a través de las cuales las clases controladoras de cada módulo pueden, en dependencia de la acción realizada por el usuario, guardar un registro en la base de datos. Estos registros (trazas) son los que luego la clase Traza gestiona, valiéndose de las funcionalidades que brindan estas clases. Estas son: AdminTrazaAdicionar.class.php y AdminTrazaRecuperar.class.php, a continuación se describen estas clases y algunas de sus funcionalidades más importantes.

**Clase AdminTrazaAdicionar.class.php**

Tabla 23. Descripción de la clase AdminTrazaAdicionar.class.php del modelo

<b>Nombre:</b> AdminTrazaAdicionar.class.php	
<b>Descripción:</b> Esta clase cuenta con las funcionalidades necesarias para adicionar, según la operación realizada por el usuario, la traza con la información necesaria. Entre sus funciones se encuentran (AdicionarTraza, AdicionarCampoValorModificado). A continuación se muestran las funcionalidades anteriores así como sus parámetros de entrada y salida.	
<b>Nombre de la función:</b> AdicionarTraza()	
<b>Parámetros de entrada</b>	
<b>Descripción</b>	<b>Tipo de Dato</b>
\$objeto_tipo_operacion	TnATipoOperacionTable
\$tabla	string
\$obj_id	int
<b>Parámetros de salida</b>	
<b>Descripción</b>	<b>Tipo de Dato</b>
respuesta	bool
<b>Descripción de la funcionalidad</b>	
Luego de realizar cualquier acción de las que sea necesario mantener un registro, la función	

adiciona una traza con la información necesaria. Estos datos incluyen: usuario, dirección ip, tabla donde se realiza la acción, entre otros.

Tabla 24. Descripción de la función AdicionarCampoValorModificado del modelo

<b>Nombre de la función:</b> AdicionarCampoValorModificado()	
<b>Parámetros de entrada</b>	
<b>Descripción</b>	<b>Tipo de Dato</b>
\$traza_id	int
\$nombre	string
\$valor	string
\$tipo_dato_id	int
<b>Parámetros de salida</b>	
<b>Descripción</b>	<b>Tipo de Dato</b>
respuesta	bool
<b>Descripción de la funcionalidad</b>	
Luego de realizar una acción de modificación en una tabla de la base de datos, la función adiciona el campo y el valor modificado del objeto operado, así como el tipo de dato del valor que ha sido modificado y el id de la traza con el que se relaciona.	

Tabla 25. Descripción de la clase AdminTrazaRecuperar.class.php del modelo

<b>Nombre:</b> AdminTrazaRecuperar.class.php	
<b>Descripción:</b> Esta clase cuenta con las funcionalidades necesarias para recuperar o eliminar la información de las trazas según desee el usuario. Entre sus funciones se encuentran (recuperarInformacionModificada, recuperarObjetoEliminado, entre otras). A continuación se muestran dos de las funcionalidades anteriores así como sus parámetros de entrada y salida.	
<b>Nombre de la función:</b> recuperarInformacionModificada()	
<b>Parámetros de entrada</b>	
<b>Descripción</b>	<b>Tipo de dato</b>
\$id_traza_modificada	int
<b>Parámetros de salida</b>	
<b>Descripción</b>	<b>Tipo de dato</b>
respuesta	bool

Descripción de la funcionalidad
La función restaura en el sistema la información guardada en la traza referente a la información modificada antes de ser realizada la acción por el usuario. En caso de que la información esté eliminada, la función retorna 'false' indicando que se debe restaurar primero el objeto eliminado y luego proceder a restaurar los valores modificados.

Tabla 26. Descripción de la función recuperarObjetoEliminado del modelo

Nombre de la función: recuperarObjetoEliminado()	
Parámetros de entrada	
Descripción	Tipo de dato
\$id_traza_eliminada	int
Parámetros de salida	
Descripción	Tipo de dato
respuesta	bool
Descripción de la funcionalidad	
La función restaura en el sistema la información de la traza referente a los datos eliminados.	

### 3.5 Tratamiento de errores

Una excepción sucede cuando ocurre una exclusión de algo que se aparta de una regla común, o una situación inesperada. Esta requiere un tratamiento especial.

En el desarrollo del componente, se definen todas aquellas excepciones que pudieran ocasionar un mal funcionamiento del sistema. Teniendo en cuenta que ya existen algunas implementadas en el marco de trabajo.

Para el tratamiento de errores, se le informará al usuario mediante mensajes, en el momento que ocurra la equivocación. Estos mensajes se emitirán en caso que se queden campos vacíos requeridos.

A continuación se explica un tratamiento de excepción, mediante los bloques de código **try** y **catch**. Para la detección de un error se utiliza el **try** y para el manejo de la misma se emplea el **catch**. La figura muestra el método para recuperar información eliminada, el cual funciona de la siguiente manera:

Primeramente se establece una transacción, que tiene como objetivo preservar la integridad de los datos que se están manejando en ese instante en la base de datos. De esta forma si se

produce algún error en el momento de la recuperación se deshacen los cambios realizados y no se corre el riesgo de perder la información. En el caso de que todo ocurra de manera satisfactoria, se cierra la transacción y son llevados a cabo todos los cambios realizados.

En la figura 10 se muestra un fragmento de código donde se realiza el tratamiento de errores en una transacción.

```
//Método auxiliar Recuperar un Objeto eliminado//
public function RecuperarObjetoEliminado($id_traza_eliminado)
{
    $conn = Doctrine_Manager :: connection ();
    try{
        $conn -> beginTransaction ();
        $obj_traza_eliminado = TbATrazaTable::getInstance()->find($id_traza_eliminado);
        $this->RestaurarObjetoEliminado($obj_traza_eliminado);
        $this->EliminarTrazaObjetoRestaurado($obj_traza_eliminado);
        $conn ->commit ();
    }
    return true;
}
catch(Doctrine_Exception $e)
{
    $conn ->rollback ();
    return false ;
}
}
```

Figura 10. Tratamiento de una excepción

### 3.6 Estándares de codificación

Los estándares de codificación ayudan a garantizar que todos los usuarios tengan acceso a la información que el sistema brinda a sus visitantes. Son pautas de programación que no están enfocadas a la lógica del programa, sino a la estructura y apariencia física del código. Es de gran importancia para lograr la calidad del software tener bien definidos estos estándares ya que convierte al software en un sistema fácil de comprender y de mantener.

#### 3.9.1 Estándares utilizados en la codificación

**Notación PascalCasing:** los identificadores y nombres de variables, métodos y funciones están compuestos por varias palabras juntas, cuya letra inicial es mayúscula.

Ejemplo: *RestaurarObjetoEliminado(\$obj\_traza\_eliminado)*

**Notación CamelCase:** es parecido al PascalCasing con la excepción que la letra inicial del identificador no debe estar en mayúscula.

Ejemplo: *restaurarInformacionModificada(\$obj\_traza\_modificada,\$campo\_modificado)*

### 3.9.2 Estilo de codificación utilizado

A continuación se relacionan los estilos de codificación utilizados:

- Se añaden comentarios descriptivos en el código de tipo aclaratorios, sólo cuando es necesario.
- Todos los nombres de las variables declaradas se escriben con letra minúscula, si poseen dos o más palabras se separan mediante un guión bajo.
- Los bloques de código siempre deben estar encerrado entre llaves, incluso si solo constan de una línea de código.
- Los identificadores de variables y métodos deben tener un nombre significativo para que por su simple lectura, pueda conocerse su función.
- La llave “{” que comienza un bloque de código, no debe aparecer a continuación de la declaración.
- La llave que cierra el bloque de código debe estar sola en una línea en correspondencia a la llave “{” que abre el bloque.

## Conclusiones Generales

Con la realización del presente trabajo de diploma se ha cumplido con el objetivo general propuesto, así como con las tareas de la investigación obteniéndose las siguientes conclusiones:

- El estudio de los principales sistemas que gestionan trazas y seguridad encontrados a nivel internacional y nacional evidenció, que estos integran en su solución, la información proveniente del registro de las acciones realizadas por los usuarios en forma de logs; centrando su fuerza en la vinculación traza-seguridad.
- El análisis de los módulos del Sistema para el control Farmacológico, ayudó a identificar la información de las trazas que se desean gestionar.
- Se asimilaron y analizaron las herramientas y tecnologías propuestas por el departamento, para el desarrollo de la solución, concluyendo que este conjunto de técnicas responden al entorno de desarrollo del proyecto, viabilizando las soluciones y el objeto de trabajo.
- Se implementaron funcionalidades que gestionan las trazas, posibilitando la realización de auditorías, análisis de rendimiento y controles por personal autorizado.

## **Recomendaciones**

Para contribuir a la continuidad de la investigación, se hace la siguiente recomendación:

Realizar un estudio de las funcionalidades de gestión de trazas determinando si es posible incluir nuevas funciones que permitan realizar análisis de rendimiento al sistema.

## Referencias Bibliográficas

**ALEGSA. 2014.** ALEGSA. [En línea] 2014. [Citado el: 4 de Abril de 2014.] <http://www.alegsa.com.ar/Dic/herramienta%20de%20modelado.php>.

**Alonso, Javier Pérez. 2012.** Ajpdsoft. [En línea] 2012. [Citado el: 20 de Enero de 2014.] <http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=820%20Proyecto%20AjpdSoft>.

**Arencibia Morales, Annia , y otros. 2013.** *Sistema de autenticación, autorización y auditoría (AAA) para aplicaciones basadas en servicios Web XML*. Habana : s.n., 2013.

**Arencibia Morales, Annia, y otros. 2013.** SLD137 SISTEMA INTEGRAL PARA EL CONTROL FARMACOLÓGICO. [En línea] 2013. [Citado el: 24 de 5 de 2014.] <http://www.informatica2013.sld.cu/index.php/informicasalud/2013/paper/view/296/216>.

**Asintec.** Asintec. *Asintec*. [En línea] [Citado el: 17 de 3 de 2014.] <http://www.asintec.es/industria/trazabilidad/software-de-trazabilidad/all-pages>.

**Bravo Llanqui, Miguel Angel, Soruco Challada, Ronald Ricardo y Arce Aguilar, Daliana Alejandra.** Universidad Salesiana. [En línea] [Citado el: 3 de 4 de 2014.]

**Cáceres Tello, Jesús.** Universidad de Alcalá. [En línea] [Citado el: 1 de 4 de 2014.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.

**Diccionario de Informática. 2014.** Diccionario de Informática. [En línea] 2014. [Citado el: 3 de 4 de 2014.] <http://www.alegsa.com.ar/Dic/active%20server%20page.php>.

**2013.** Dirección de Informática y Comunicaciones del MINSAP. *Dirección de Informática y Comunicaciones del MINSAP*. [En línea] 2013. [Citado el: 24 de Mayo de 2014.] <http://www.di.sld.cu/cms/wp-content/uploads/2013/12/Normas-para-el-desarrollo-de-aplicaciones-informaticas-para-la-Salud-en-Cuba.pdf>.

**Eguiluz, Javier. 2014.** Introducción a AJAX. [En línea] 1 de 11 de 2014. [Citado el: 27 de 2 de 2014.] <http://www.librosweb.es/ajax/capitulo1.html>.

**Esepe EStudio. 2005.** Esepe EStudio. [En línea] 16 de 8 de 2005. [Citado el: 10 de 3 de 2014.] <http://www.espestudio.com/noticias/que-es-mysql>.

**Foro Web. 2012.** Foro Web. [En línea] Abril de 2012. [Citado el: 10 de Febrero de 2014.] <http://www.forosdelweb.com/f181/problemas-con-symfony-985125/>.

**Grokbase. 2014.** Grokbase. [En línea] 24 de Marzo de 2014. [Citado el: 20 de Abril de 2014.] <http://grokbase.com/t/gg/symfony-es/143rn24pkb/proyecto-con-symfony2-angularjs-mongodb-%C2%BFque-opin%C3%A1is>.

**GSInnova. 2011.** rational. [En línea] 4 de 2 de 2011. [Citado el: 20 de 1 de 2014.] <http://www.rational.com.ar/herramientas/rup.html>.

**Herrera López, Adriana. 2011.** lacoctelera. [En línea] 7 de 2011. [Citado el: 26 de 2 de 2014.] <http://ady-info-design.lacoctelera.net/post/2011/07/16/la-fundamentacion-teorica-una-tesis>.

**Inder. 2004.** Portal Inder. [En línea] 03 de 11 de 2004. [Citado el: 23 de 02 de 2014.] <http://portal.inder.cu/index.php/recursos-informacionales/alerta-informativa/115-la-informatica-y-su-impacto-social>.

**Ingeniería de Software. 2012.** Ingeniería de Software. [En línea] 22 de 09 de 2012. [Citado el: 03 de 04 de 2014.] <http://ingenieriadesoftware.bligoo.com.mx/requerimientos-funcionales-y-no-funcionales-rf-rnf>.

Instituto tecnológico de la Laguna. <http://laguna.snit.mx/index.htm>. [En línea] [Citado el: 21 de febrero de 2014.] [http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=16&cad=rja&ved=0CEYQFjAFOAo&url=http%3A%2F%2Fwww.itlalaguna.edu.mx%2Facademico%2Fcarreras%2Fsistemas%2Fprograweb%2FUnidad5.doc&ei=uL4HU4fzNJOgkQeejIGwAQ&usg=AFQjCNHrKDZ1mH-2n\\_JXzyRzyMH75aGmcA](http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=16&cad=rja&ved=0CEYQFjAFOAo&url=http%3A%2F%2Fwww.itlalaguna.edu.mx%2Facademico%2Fcarreras%2Fsistemas%2Fprograweb%2FUnidad5.doc&ei=uL4HU4fzNJOgkQeejIGwAQ&usg=AFQjCNHrKDZ1mH-2n_JXzyRzyMH75aGmcA).

**Instituto Tecnológico Superior Escárcega. 2010.** *Reporte de Intalación de Apache*. Escárcega, Campeche : s.n., 2010.

**iOCárnicas. 2012.** iOCárnicas. [En línea] 2012. [Citado el: 17 de Enero de 2014.] <http://www.iocarnicas.com/index.php/en/traceability/traceability-software.html>.

**Jacobson, James. 2004.** *El proceso unificado del desarrollo de software*. España : Addison Wesley, 2004.

**Jojoa. 2010.** *Jojoa. Tecnología y marketing*. 2010.

**King, Hannibal. 2010.** ProgramaciónWeb.net. [En línea] 11 de 09 de 2010. [Citado el: 20 de 1 de 2014.]

**LibrosWeb.es. 2014.** LibrosWeb.es. [En línea] 2014. [Citado el: 20 de Enero de 2014.] [http://librosweb.es/javascript/capitulo\\_1/glosario\\_basico.html](http://librosweb.es/javascript/capitulo_1/glosario_basico.html).

**Microsoft. 2014.** Microsoft. [En línea] 2014. [Citado el: 08 de 05 de 2014.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.

**Moral, Jose A. del.** gananzia. *gananzia*. [En línea] [Citado el: 20 de Enero de 2014.] <http://gananzia.com/s21sec-ofrece-comercialmente-una-aplicacion-de-gestion-de-logs-que-desarrollo-para-bankinter>.

**Moreno, Eglis Maylin Lamour. 2011.** Universidad de Málaga. [En línea] julio de 2011. [Citado el: 24 de febrero de 2014.] <http://www.eumed.net/rev/cccss/13/emlm.htm>.

**NetBeans. 2013.** NetBeans. [En línea] 2013. [Citado el: 28 de 2 de 2014.] [https://netbeans.org/community/releases/69/index\\_es.html](https://netbeans.org/community/releases/69/index_es.html).

**postgresql. 2010.** postgresql. [En línea] 10 de 10 de 2010. [Citado el: 20 de 1 de 2014.] <http://www.postgresql.org/about/>.

**Pressman, Roger. 2001.** *Ingeniería de Software. Un enfoque práctico*. 2001.

**Sales, Sergi. 2010.** UPC NET. [En línea] 3 de 10 de 2010. [Citado el: 20 de 2 de 2014.] <https://intranet.upcnet.es/innovacio/innovasaurus/vigilancia-tecnologica/la-competencia-del-sigvi-bitacora-v5-de-s21sec>.

**Sánchez Rodríguez, Alfredo y Pompa Sourd, Frank. 2007.** Dirección de Informática y Comunicaciones. *Dirección de Informática y Comunicaciones*. [En línea] Junio de 2007. [Citado

el: 24 de Mayo de 2014.] <http://www.di.sld.cu/cms/wp-content/uploads/2013/12/Normas-para-el-desarrollo-de-aplicaciones-informaticas-para-la-Salud-en-Cuba.pdf>.

**Sánchez Rosas, Juan Eladio. 2008.** Desarrollo en Web. [En línea] 12 de 11 de 2008. [Citado el: 20 de Enero de 2014.]

**Sánchez Santos, Arletys y Pacheco Correa, Yunior. 2013.** *Mecanismo de control de acceso en los servicios web y gestión de trazas en el Sistema Nomenclador de Información.* La Habana : s.n., 2013.

Sellenne ERP Trazabilidad. *Sellenne ERP Trazabilidad.* [En línea] [Citado el: 18 de Enero de 2014.] <http://www.sellenne-software-erp.pe/ERP-Funcionalidades/ERP-con-Trazabilidad/19.html>.

**SOFLATAM. 2012.** SOFLATAM. [En línea] Buenos Aires, Argentina, 2012. [Citado el: 17 de 3 de 2014.] <http://www.softlatam.com.ar/producto-traza.html>.

**Sommerville, Ian;. 2005.** *Ingeniería de software.* Madrid : s.n., 2005.

**SynerPlus.** SynerPlus. [En línea] [Citado el: 27 de 2 de 2014.] <http://www.synerplus.es/Sellenne-ERP/ERP-Software-Con-Trazabilidad/46.html>.

**The Free Dictionary. 2013.** The Free Dictionary. [En línea] 2013. [Citado el: 20 de 5 de 2014.] <http://es.thefreedictionary.com/traza>.

**Universidad de las Ciencias Informáticas. 2014.** Uciencia. *Uciencia.* [En línea] 06 de 2 de 2014. [Citado el: 17 de 3 de 2014.] <http://uciencia.uci.cu/?q=node/2541>.

**Vidal Ledo, María , García Pierrot, Gonzalo y Cazes, Guillermo.** *Revista cubana de informática médica.* [En línea] [Citado el: 20 de 2 de 2014.] [http://www.rcim.sld.cu/revista\\_7/articulo\\_htm/segurinfsalud.htm](http://www.rcim.sld.cu/revista_7/articulo_htm/segurinfsalud.htm).

**Zaninotto, Frncois y Fabien, Potencier. 2008.** *Sympony la guía definitiva.* 2008.

**Zorrilla Pantaleón, Mata E. 2011.** Universidad de Cantabria. [En línea] 2011. [Citado el: 24 de 5 de 2014.] <http://personales.unican.es/zorrillm/BasesDatos/02%20-%20Modelos%20de%20datos%20ER-UML-relacional.pdf>.

---

## Bibliografía

**Potencier, Fabien y Zaninotto, François.** *Symfony 1.4, la guía definitiva.*

**ALEGSA. 2014.** ALEGSA. [En línea] 2014. [Citado el: 4 de Abril de 2014.] <http://www.alegsa.com.ar/Dic/herramienta%20de%20modelado.php>.

**Alonso, Javier Pérez. 2012.** Ajpdsoft. [En línea] 2012. [Citado el: 20 de Enero de 2014.] <http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=820%20Proyecto%20AjpdSoft>.

**Arencibia Morales, Annia , y otros. 2013.** *Sistema de autenticación, autorización y auditoría (AAA) para aplicaciones basadas en servicios Web XML.* Habana : s.n., 2013.

**Arencibia Morales, Annia, y otros. 2013.** SLD137 SISTEMA INTEGRAL PARA EL CONTROL FARMACOLÓGICO. [En línea] 2013. [Citado el: 24 de 5 de 2014.] <http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/view/296/216>.

**Asintec.** Asintec. *Asintec.* [En línea] [Citado el: 17 de 3 de 2014.] <http://www.asintec.es/industria/trazabilidad/software-de-trazabilidad/all-pages>.

**Bravo Llanqui, Miguel Angel, Soruco Challada, Ronald Ricardo y Arce Aguilar, Daliana Alejandra.** Universidad Salesiana. [En línea] [Citado el: 3 de 4 de 2014.]

**Cáceres Tello, Jesús.** Universidad de Alcalá. [En línea] [Citado el: 1 de 4 de 2014.] <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>.

**Caldwell, Ben , Cooper, Michael y Guarino Reid, Loretta . 2008.** Pautas de Accesibilidad de Contenido Web 2.0. [En línea] 11 de Diciembre de 2008. [Citado el: 15 de Mayo de 2014.] <http://www.codexemplar.org/traduccion/pautas-accesibilidad-contenido-web-2.0.htm>.

**Diccionario de Informática. 2014.** Diccionario de Informática. [En línea] 2014. [Citado el: 3 de 4 de 2014.] <http://www.alegsa.com.ar/Dic/active%20server%20page.php>.

**2013.** Dirección de Informática y Comunicaciones del MINSAP. *Dirección de Informática y Comunicaciones del MINSAP.* [En línea] 2013. [Citado el: 24 de Mayo de 2014.]

---

<http://www.di.sld.cu/cms/wp-content/uploads/2013/12/Normas-para-el-desarrollo-de-aplicaciones-informaticas-para-la-Salud-en-Cuba.pdf>.

**Eguiluz, Javier. 2014.** Introducción a AJAX. [En línea] 1 de 11 de 2014. [Citado el: 27 de 2 de 2014.] <http://www.librosweb.es/ajax/capitulo1.html>.

**Esepe EStudio. 2005.** Esepe EStudio. [En línea] 16 de 8 de 2005. [Citado el: 10 de 3 de 2014.] <http://www.espeestudio.com/noticias/que-es-mysql>.

**Foro Web. 2012.** Foro Web. [En línea] Abril de 2012. [Citado el: 10 de Febrero de 2014.] <http://www.forosdelweb.com/f181/problemas-con-symfony-985125/>.

**Grokbase. 2014.** Grokbase. [En línea] 24 de Marzo de 2014. [Citado el: 20 de Abril de 2014.] <http://grokbase.com/t/gg/symfony-es/143rn24pkb/proyecto-con-symfony2-angularjs-mongodb-%C2%BFque-opin%C3%A1is>.

**GSInnova. 2011.** rational. [En línea] 4 de 2 de 2011. [Citado el: 20 de 1 de 2014.] <http://www.rational.com.ar/herramientas/rup.html>.

**Herrera López, Adriana. 2011.** lacoctelera. [En línea] 7 de 2011. [Citado el: 26 de 2 de 2014.] <http://ady-info-design.lacoctelera.net/post/2011/07/16/la-fundamentacion-teorica-una-tesis>.

**Inder. 2004.** Portal Inder. [En línea] 03 de 11 de 2004. [Citado el: 23 de 02 de 2014.] <http://portal.inder.cu/index.php/recursos-informacionales/alerta-informativa/115-la-informatica-y-su-impacto-social>.

**Ingeniería de Software. 2012.** Ingeniería de Software. [En línea] 22 de 09 de 2012. [Citado el: 03 de 04 de 2014.] <http://ingenieriadesoftware.bligoo.com.mx/requerimientos-funcionales-y-no-funcionales-rf-rnf>.

Instituto tecnológico de la Laguna. <http://laguna.snit.mx/index.htm>. [En línea] [Citado el: 21 de febrero de 2014.] [http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=16&cad=rja&ved=0CEYQFjAFOAo&url=http%3A%2F%2Fwww.itlalaguna.edu.mx%2Facademico%2Fcarreras%2Fsistemas%2Fprograweb%2FUnidad5.doc&ei=uL4HU4fzNJOgkQeejIGwAQ&usg=AFQjCNHrKDZ1mH-2n\\_JXzyRzyMH75aGmcA](http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=16&cad=rja&ved=0CEYQFjAFOAo&url=http%3A%2F%2Fwww.itlalaguna.edu.mx%2Facademico%2Fcarreras%2Fsistemas%2Fprograweb%2FUnidad5.doc&ei=uL4HU4fzNJOgkQeejIGwAQ&usg=AFQjCNHrKDZ1mH-2n_JXzyRzyMH75aGmcA).

**Instituto Tecnológico Superior Escárcega. 2010.** *Reporte de Intalación de Apache.* Escárcega, Campeche : s.n., 2010.

**iOCárnicas. 2012.** iOCárnicas. [En línea] 2012. [Citado el: 17 de Enero de 2014.] <http://www.iocarnicas.com/index.php/en/traceability/traceability-software.html>.

**Jacobson, James. 2004.** *El proceso unificado del desarrollo de software.* España : Addison Wesley, 2004.

**Jojooa. 2010.** *Jojooa. Tecnología y marketing.* 2010.

**King, Hannibal. 2010.** ProgramaciónWeb.net. [En línea] 11 de 09 de 2010. [Citado el: 20 de 1 de 2014.]

**LibrosWeb.es. 2014.** LibrosWeb.es. [En línea] 2014. [Citado el: 20 de Enero de 2014.] [http://librosweb.es/javascript/capitulo\\_1/glosario\\_basico.html](http://librosweb.es/javascript/capitulo_1/glosario_basico.html).

**Microsoft. 2014.** Microsoft. [En línea] 2014. [Citado el: 08 de 05 de 2014.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.

**Moral, Jose A. del.** gananzia. *gananzia.* [En línea] [Citado el: 20 de Enero de 2014.] <http://gananzia.com/s21sec-ofrece-comercialmente-una-aplicacion-de-gestion-de-logs-que-desarrollo-para-bankinter>.

**Moreno, Eglis Maylin Lamour. 2011.** Universidad de Málaga. [En línea] julio de 2011. [Citado el: 24 de febrero de 2014.] <http://www.eumed.net/rev/cccss/13/emlm.htm>.

**NetBeans. 2013.** NetBeans. [En línea] 2013. [Citado el: 28 de 2 de 2014.] [https://netbeans.org/community/releases/69/index\\_es.html](https://netbeans.org/community/releases/69/index_es.html).

**postgresql. 2010.** postgresql. [En línea] 10 de 10 de 2010. [Citado el: 20 de 1 de 2014.] <http://www.postgresql.org/about/>.

**Pressman, Roger. 2001.** *Ingeniería de Software. Un enfoque práctico.* 2001.

---

**RUP. 2011.** Ecured. [En línea] 04 de 02 de 2011. [Citado el: 20 de 01 de 2014.] <http://www.ecured.cu/index.php/RUP>.

**Sales, Sergi. 2010.** UPC NET. [En línea] 3 de 10 de 2010. [Citado el: 20 de 2 de 2014.] <https://intranet.upcnet.es/innovacio/innovasaurus/vigilancia-tecnologica/la-competencia-del-sigvi-bitacora-v5-de-s21sec>.

**Sánchez Rodríguez, Alfredo y Pompa Sourd, Frank. 2007.** Dirección de Informática y Comunicaciones. *Dirección de Informática y Comunicaciones*. [En línea] Junio de 2007. [Citado el: 24 de Mayo de 2014.] <http://www.di.sld.cu/cms/wp-content/uploads/2013/12/Normas-para-el-desarrollo-de-aplicaciones-informaticas-para-la-Salud-en-Cuba.pdf>.

**Sánchez Rosas, Juan Eladio. 2008.** Desarrollo en Web. [En línea] 12 de 11 de 2008. [Citado el: 20 de Enero de 2014.]

**Sánchez Santos, Arletys y Pacheco Correa, Yunior. 2013.** *Mecanismo de control de acceso en los servicios web y gestión de trazas en el Sistema Nomenclador de Información*. La Habana : s.n., 2013.

Selenne ERP Trazabilidad. *Selenne ERP Trazabilidad*. [En línea] [Citado el: 18 de Enero de 2014.] <http://www.selenne-software-erp.pe/ERP-Funcionalidades/ERP-con-Trazabilidad/19.html>.

**SOFLATAM. 2012.** SOFLATAM. [En línea] Buenos Aires, Argentina, 2012. [Citado el: 17 de 3 de 2014.] <http://www.softlatam.com.ar/producto-traza.html>.

**Sommerville, Ian;. 2005.** *Ingeniería de software*. Madrid : s.n., 2005.

**Symfony para Todos.** [En línea] <http://symfony.cubava.cu/introduccion/symfony-2/>.

**SynerPlus.** SynerPlus. [En línea] [Citado el: 27 de 2 de 2014.] <http://www.synerplus.es/Selenne-ERP/ERP-Software-Con-Trazabilidad/46.html>.

**The Free Dictionary. 2013.** The Free Dictionary. [En línea] 2013. [Citado el: 20 de 5 de 2014.] <http://es.thefreedictionary.com/traza>.

**Universidad de las Ciencias Informáticas. 2014.** Uciencia. *Uciencia*. [En línea] 06 de 2 de 2014. [Citado el: 17 de 3 de 2014.] <http://uciencia.uci.cu/?q=node/2541>.

**Vidal Ledo, María , García Pierrot, Gonzalo y Cazes, Guillermo.** Revisa cubana de informática médica. [En línea] [Citado el: 20 de 2 de 2014.] [http://www.rcim.sld.cu/revista\\_7/articulo\\_htm/segurinsalud.htm](http://www.rcim.sld.cu/revista_7/articulo_htm/segurinsalud.htm).

**W3c. 2013.** Pautas de Accesibilidad al Contenido en la Web 1.0. [En línea] 13 de diciembre de 2013. [Citado el: 10 de Mayo de 2014.] [http://www.usableyaccesible.com/recurso\\_misvalidadores.php#accesibilidadwcag2](http://www.usableyaccesible.com/recurso_misvalidadores.php#accesibilidadwcag2).

**Zaninotto, Frncois y Fabien, Potencier. 2008.** *Sympony la guía definitiva*. 2008.

**Zorrilla Pantaleón, Mata E. 2011.** Universidad de Cantabria. [En línea] 2011. [Citado el: 24 de 5 de 2014.] <http://personales.unican.es/zorrillm/BasesDatos/02%20-%20Modelos%20de%20datos%20ER-UML-relacional.pdf>.

## Anexos

### Anexo 1 Entrevista

Entrevista realizada al Lic. Alfredo Sánchez Rodríguez. Jefe del Grupo de Infraestructuras de SOFTEL UCI.

#### **¿Qué es el sistema SAAA y en que tecnología se basa?**

El sistema SAAA es un Sistema de Autenticación, Autorización y Auditoría que está basado en la tecnología GSI (Grid Security Infrastructure), tecnología compuesta por un conjunto de ordenadores que brinda servicios de seguridad de recursos distribuidos, la autenticación y autorización de dichos recursos se realizan mediante el uso de certificados digitales estándar X509.

También puede ser denominado un sistema paralelo y distribuido que permite: compartir, seleccionar, añadir recursos distribuidos a través de múltiples dominios administrativos.

Basados en requerimientos de:

- Disponibilidad
- Capacidad
- Desempeño
- Costo y calidad de servicio

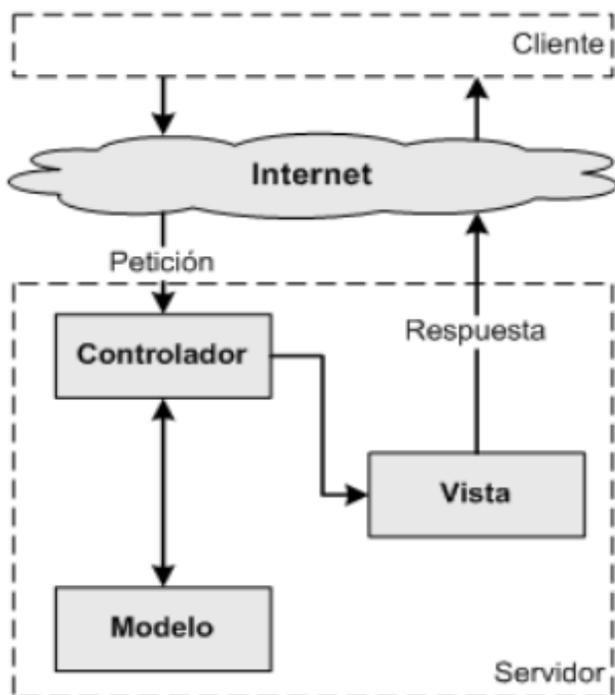
#### **¿Cómo se realiza la gestión de trazas en el sistema?**

La Gestión de trazas se llevan a cabo sobre las acciones que realizan los usuarios en los recursos, guardando la hora, fecha, usuario y operación realizada. Estas solo pueden ser gestionadas por los administradores del sistema, desde los administradores Nacionales, Provinciales, Municipales y de Unidad de Salud.

#### **¿Se puede revocar la pérdida de datos en el sistema?**

Si, existen informaciones que por su importancia es necesario poder recuperarla en caso de que estas sean modificadas o eliminadas. No se puede decir que informaciones son, por la seguridad del sistema.

**Anexo 2** Diagrama que representa el patrón MVC en Symfony.



**Anexo 3** Diagrama que muestra el flujo de trabajo de Symfony.