Universidad de las Ciencias Informáticas

Sistema de Gestión Integral de Seguridad. (SIGIS)

Trabajo de diploma para optar por el título de Ingeniero Informático.

AUTORES

Noel Jesús Rivero Pino

Aquiles Massó Machandi

TUTOR

Oiner Gómez Baryolo

CO-TUTOR

Darién García Tejo

Ciudad de la Habana, Junio, 2009

Año del 50 aniversario del triunfo de la revolución.

DEDICATORIA

Noel Jesus Rivero Pino.

Dedico este trabajo a toda mi familia en especial a mis padres por todo su apoyo y todo lo que han hecho por mí.

Aquiles Massó Machandi.

Dedico este trabajo a toda mi familia por apoyarme siempre y por su confianza, y a los amigos que estuvieron en las malas y en las buenas.

AGRADECIMIENTOS

ElNoE.

Primeramente un beso y un abrazo bien bien bien grande a las dos personas que son mi razón de ser, las dos personas que mas quiero en el mundo, mami y papi. Los quiero mucho mucho. Gracias por estar siempre ahí cuando necesito de su ayuda. Sin la ayuda de ustedes no hubiese sido posible llegar hasta aquí.

*Otro besote grandote a mi bisa mima y a mis abuelitos, especialmente a mi **ueli** linda que siempre me esta mimando por que dice que todavía yo soy su niño chiquito. Te quiero mucho.*

A mis 2 segundas madres, mis tías lindas, un poco loquitas las dos, las quiero mucho, gracias por ayudarme siempre en todo.

Un abrazo grande para mi tío, gracias por preocuparte y ayudarme siempre.

Un besote a mi hermanito Pedri, te quiero mucho sigue estudiando, luchando y esforzándote que vas muy muy bien, y sigue practicando fútbol para ver si me puedes ganar.

A todos mis primos un abrazo grande los quiero.

A mi compañero de tesis Aquiles, un poco cabezón pero bueno se ha comportado muy bien y me ha ayudado mucho. Te quiero chama.

A mis 2 tutores Oiner y Darien, gracias chamas por ayudarme y enseñarme mucho.

A mi oponente José Luis, por la ayuda que nos brindó para refinar y arreglar los errores de la tesis. Muchas gracias.

Al tribunal por su gran ayuda.

Al pikete RS, Bolek, el lisu, Cacharro, el porreto, el gwo, yuri, jonathan, charly, forta, Sergi que compartimos muchas travesuras y alegrías a lo largo de la carrera y nos ayudamos en las buenas y en las malas. Los quiero.

Agradecimientos

A mi grupo lindo por compartir y ayudarnos siempre, en especial a mis amigas la china, Daili y lisin, muaaaaaaa, las quiero.

A mi piquete especial de arquitectura, Javi, Aliexer (Los líderes), Adri eres especial te quiero mucho, Gretty, Akiles, Rosa, Mile, que en este último año hemos consolidado una gran amistad y espero se siga fortaleciendo. Los quiero.

A mis amis lindis luci y liu, gracias por ser especiales, por ayudarme y darme muchos consejos. Las quiero muchísimo.

Al piquete sabroso de las canchas.

Al piquete de la Lenin que muchos empezamos juntos desde la primaria y nos graduamos aquí juntos. Los quiero.

A todos mis amigos que por dejar de mencionarlos no significa que no hayan significado mucho para mí. Los quiero a todos.

Capo.

Primeramente agradecer a mi mamá y a mi papa que bastante me han apoyado durante toda mi carrera.

A mi prima Karelia, a mi primo Arismey y a toda mi familia por creer en mi y apoyarme.

A mis amigos, los de antaño y los de ahora, los que vienen conmigo desde los camilitos,Raul , Diovis,Adrian, ALina ;el pikete del doble Tony,El Duque ,Dariel ,Lore ,Dannys ,Yariek y ha esa tropa extraordinaria de colegas del ERP en especial los de arquitectura,Gretel,Javier,Aliexer,Adriana,Osvaldo y Yannia .

A los tutores Oiner y Darien por siempre estar dispuestos a ayudarnos.

Al jefe de proyecto Rene por aconsejarnos.

A mi compañero de tesis Noel, un chama de lo mejor y muy responsable .Te quiero mano.

A todas las personas que de una forma u otra me ayudaron en la realización de la tesis.

“Tu tiempo está limitado, así que no lo desaproveches viviendo la vida de algún otro. No te dejes arrastrar por los dogmas, que es lo mismo que vivir con los resultados del pensamiento de otras personas. No dejes que el ruido de las opiniones de otros ahoguen completamente tu voz interior. Y más importante, ten el valor de seguir a tu corazón y a tu intuición. Ellos, de algún modo, ya saben en lo que verdaderamente te quieres convertir.Todo lo demás es secundario.”

Steve Jobs.

RESUMEN

La seguridad es un factor fundamental en cualquier institución y los datos son una parte fundamental en las mismas y cualquier pérdida, desvió o mala manipulación de la información ocasionaría daños económicos y sociales irreparables.

Por tanto se requiere de una buena administración y control de la información y medios materiales, así como una adecuada seguridad acorde a los intereses del país.

En la actualidad todas las aplicaciones implantadas en Cuba donde existe manejo de la información implementan su propia seguridad, lo cual ocasiona pérdida de tiempo y gastos innecesarios de cuantiosos recursos humanos y materiales. Además no se realiza una administración centralizada de los sistemas de seguridad, lo que trae consigo que sea muy difícil controlar que no ocurran violaciones en los sistemas y en caso de que ocurra poder detectarlas. Para minimizar estos riesgos surge la necesidad de desarrollar un sistema que gestione la seguridad de forma centralizada en un entorno de varias aplicaciones. De esta forma se estará fortaleciendo una parte muy importante de la arquitectura de seguridad como es la protección de la información almacenada en los sistemas que utilicen sus servicios, disminuyendo así el tiempo de desarrollo de las aplicaciones, el costo total de los proyectos y los riesgos que puedan ser aprovechados para realizar un ataque.

SIGIS brindará sus servicios a todos los sistemas que se suscriban a él. Para ello se gestionarán las conexiones a la base de datos, las funcionalidades asociadas y las acciones que realizan las mismas. Una vez registrada esta información se procederá a la creación de roles a los cuales se les darán los permisos dentro de cada sistema. Luego se crearán los usuarios con el perfil seleccionado y se le asignarán uno o muchos roles en una o muchas entidades respecivamente.

Para aumentar la flexibilidad se garantiza la integración con otros sistemas, a nivel de servicios web, de interfaz de usuario y por medio del componente IOC[1], para este último el sistema que lo necesita debe estar desarrollado sobre el marco de trabajo PACSOFT[2].

[1] **IOC** del inglés Inversion of Control, es un concepto junto a unas técnicas de programación en las que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos (procedure calls) o funciones.

[2] **PACSOFT** (Paquete de componentes Softwares), es un marco de trabajo compuesto por varios componentes desarrollados por el equipo de arquitectura del Centro Integral de Gestión de Entidades.

Índice de contenido.

Índice

Índice

Índice de Tablas

Índice

Índice de Figuras

Índice

Introducción.

A medida que van pasando los años las Tecnologías de la Informática y las Comunicaciones (TIC) se van desarrollando y por ende se van perfeccionando. Cuba, aunque se encuentra fuertemente bloqueado por el imperialismo, hace todo lo posible para estar a la altura de los países desarrollados en esta rama, llevando a cabo la informatización de todos los sectores del país. Para ello por idea del Comandante en Jefe Fidel Castro Ruz surge la Universidad de las Ciencias Informáticas (UCI). Esta institución tiene como misión insertar la isla en el mercado mundial del software aprovechando así la capacidad intelectual e inventiva de los cubanos, produciendo software en este sentido.

Las aplicaciones que se realizan en la UCI requieren de una seguridad estricta y bien concebida que permita controlar y monitorear la información para evitar que las aplicaciones sean atacadas y en caso de que esto ocurra, que el ataque no sea fructífero y se logre atrapar al atacante en el menor tiempo posible. Es por este motivo que los procesos de gestión de seguridad de los sistemas informáticos son muy importantes y constituyen una parte fundamental de los mismos.

Es necesario que los sistemas conciban una política de seguridad adecuada para logar un buen desempeño de sus objetivos. Con el desarrollo de SIGIS se logrará una administración segura y centralizada de todos los sistemas que utilicen sus servicios, disminuyendo así el tiempo de desarrollo de las aplicaciones, el costo total de los proyectos y los riesgos de seguridad.

La UCI en conjunto con la Unidad de Compatibilización Integración y Desarrollo de Software para la Defensa (UCID), se encuentran desarrollando una serie de aplicaciones entre ellas el Sistema de Gestión de Recursos Empresariales Cedrux ó ERP Cuba, como también se le conoce, de vital importancia para el desarrollo del país. En él se manejarán todos los recursos materiales, humanos y financieros para lograr un mejor control de los recursos económicos. Por la importancia de la información que manejará este sistema es preciso desarrollar una aplicación que administre la seguridad de forma centralizada en un entorno de varias aplicaicones. Este sistema no sólo prestará sus servicios al ERP sino a todos los sistemas que requieran de los mismos.

En la UCI cada proyecto implementa una versión diferente para administrar la seguridad de cada producto. El centro UCID no queda exento de esta problemática por lo que también

desarrolló su propio sistema para la administración de la seguridad al cual se le detectaron una serie de aspectos que no se tuvieron en cuenta a la hora de concebir el mismo, se desarrolló en un corto período de tiempo por lo cual carecía de la calidad requerida. Por esto los usuarios planteaban que era muy difícil de configurar, además estaba necesitada de la administración de conexiones a la base de datos, de resolver la problemática usuario multi-entidad para así lograr la compartimentación de la información en cada una de ellas, de controlar los accesos a los servicios entre sistemas y dar la posibilidad al usuario de configurar su perfil. Las interfaces diseñadas presentaron problemas a la hora de realizar las actualizaciones deseadas y presentaban un entorno poco amigable. Tampoco se podía importar o exportar sistemas ya registrados para facilitar el trabajo de los administradores.

Teniendo en cuenta las deficiencias antes mencionadas, partiendo de esta solución y de los nuevos requerimientos a raíz del desarrollo de Cedrux, surge la imperiosa necesidad de crear un sistema capaz de gestionar la seguridad los recursos del país que este manejaría, planteando el siguiente **problema a resolver:** ¿Cómo lograr la administración centralizada de la seguridad en un entorno de varias aplicaciones?

Para ello se tendrá como **Objeto de estudio:** La administración de la seguridad en softwares de gestión.

Teniendo como **campo de acción**: La administración de la seguridad en sistemas web.

Para resolver el problema planteado se ha propuesto como **objetivo general:** Desarrollar un sistema de seguridad que garantice la gestión centralizada de la seguridad en un entorno de varias aplicaciones.

Para dar cumplimiento al objetivo general se han trazado los siguientes **objetivos específicos:**

- Definir y fundamentar los elementos teóricos para el análisis del problema a resolver.
- Diseñar un sistema que administre la seguridad de forma centralizada en un entorno de varias aplicaciones.
- Implementar el sistema para la administración centralizada de la seguridad en un entorno de varias aplicaciones.
- Realizar pruebas al sistema de gestión integral de seguridad.

Introducción

Para llevar a cabo esta investigación y dar cumplimiento a los objetivos propuestos, se planificaron las siguientes **tareas**:

- Estudio de la información obtenida sobre la seguridad en los marcos de trabajo en PHP.
- Estudio de los sistemas existentes que administran la seguridad de forma centralizada en un entorno de varias aplicaciones.
- Diseño del sistema.
- Implementación del sistema.
- Realización de las pruebas a la solución.

Se tiene como **idea a defender** de la presente investigación:

Si se desarrolla SIGIS se logrará la administración centralizada de la seguridad en un entorno de varias aplicaciones.

- V.I: Desarrollo de SIGIS
- V.D: Administración centralizada de la seguridad en un entorno de varias aplicaciones.

El trabajo consta de tres capítulos, el primero trata la fundamentación teórica del trabajo, el mismo incluye seguridad en frameworks PHP, estudio de otras soluciones, elementos de reutilización y resultados esperados u objetivos de la solución propuesta.

El **segundo capítulo** trata sobre los requisitos, prototipos de interfaces, diseño de clases, artefacto de implementación y casos de pruebas.

El tercer y último capítulo se realiza una valoración de la solución, matriz de cubrimiento o matriz de inferencia de indicadores de calidad, métricas que evalúan el diseño, análisis de los resultados y pruebas de concepto.

1

Capítulo

Fundamentación teórica

CAPÍTULO 1: Fundamentación teórica

1.1 Introducción

La gestión centralizada de la seguridad tiene suma importancia para lograr un mayor control y seguridad de la información. Dicho proceso actualmente no se tiene en cuenta en los sistemas ubicados en la UCI y en Cuba, por lo que este trabajo está enmarcado en el desarrollo de una aplicación que gestione la seguridad de forma centralizada en un entorno de varias aplicaciones. En este capítulo se hace un análisis de los problemas que existen actualmente cuando se le implementa usando un framework PHP la seguridad a un sistema web. Además se llevó a cabo un estudio referencial del estado del arte actual de los sistemas que implementaban una seguridad centralizada.

1.2 Seguridad en aplicaciones Web.

El término seguridad informática es una generalización para un conjunto de tecnologías que ejecutan ciertas tareas relativas a la seguridad de los datos. (1)

ISO, en su norma 7498, define la seguridad informática como una serie de mecanismos que minimizan la vulnerabilidad de bienes y recursos, donde un bien se define como algo de valor y la vulnerabilidad se define como la debilidad que se puede explotar para violar un sistema o la información que contiene. El bien más preciado por cualquier institución es la información y de ahí que se han desarrollado protocolos y mecanismos adecuados, para preservar su seguridad. Se puede hablar en este sentido de cinco conceptos principales de la seguridad de los sistemas: *autentificación*, *autorización, auditoría, administración de perfiles y administración de conexiones*. Basándonos en estos conceptos a la hora de implentar la seguridad se lograría cumplir con la *confidencialidad*, *integridad* y el *no-repudio* aspectos fundamentales para cualquier sistema que gestione información.

A la hora de desarrollar una aplicación, generalmente nos centramos más en la funcionalidad que en la seguridad. Lo que trae como consecuencia que los atacantes se aprovechen de esto y atenten contra cualquiera de estos cuatro aspectos. En la seguridad de aplicaciones juegan

un papel fundamental los procesos de autenticación y autorización, ya que permiten un mejor control en el acceso a la información.

La **autenticación** (o **autentificación**) es el proceso de verificar formalmente la identidad de las entidades participantes en una comunicación o intercambio de información. Por entidad se entiende tanto personas, como procesos o computadoras.
Existen varias formas de poder autentificarse:

- Basada en claves.
- Basada en direcciones.
- Criptográfica.

De estas tres posibilidades la más segura es la tercera, pues en el caso de las dos primeras es posible que alguien escuche la información enviada y pueden suplantar la identidad del emisor de información.

Desde otro punto de vista se puede hablar de formas de autentificarse, como puede ser a través de la biometría (huellas digitales, retina del ojo, la voz...), por medio de passwords o claves, y por último utilizando algo que poseamos, como un certificado digital.
Se llama autentificación fuerte a la que utiliza al menos dos de las tres técnicas mencionadas en el párrafo anterior, siendo bastante frecuente el uso de la autentificación biométrica, que como se indicó antes se basa en la identificación de personas por medio de algún atributo físico.

La **autorización** es la parte del sistema que protege los recursos del sistema permitiendo que sólo sean usados por aquellos consumidores a los que se les ha concedido autorización para ello. Los recursos incluyen archivos y otros objetos de dato, programas, dispositivos y funcionalidades provistas por aplicaciones.
El control de la **auditoría** en aplicaciones en una entidad determinada es de vital importancia. Comúnmente, las aplicaciones presentan una gran vulnerabilidad lo que provoca un posible ataque y el mismo no necesariamente depende de la plataforma y tecnologías utilizadas. Estas vulnerabilidades pueden aparecer por un error en la codificación del sistema o simplemente un mal diseño de la aplicación. Para evitar la existencia de vulnerabilidades se implementa la auditoría donde para tener un buen monitoreo de lo que sucede en las aplicaciones de la

empresa es recomendable hacer un resumen o reporte, ya sea diario, semanal, o según el tiempo que se estime, de las acciones realizadas sobre las mismas siempre recogiendo algunos conceptos que son de mucho interés como por ejemplo:

Logs: Son ficheros que almacenan información, generalmente esta información registra el acceso que realiza un usuario a una aplicación.

En estos ficheros se almacena:

- *Páginas visitadas (Qué):* Páginas de la aplicación que son visitadas por un usuario determinado o acción realizada.
- *IP del visitante (Desde):* Dirección de donde proviene la conexión, o sea, dirección de la PC donde está ubicado el usuario que accede a la aplicación.
- *Fecha y hora de la conexión (Cuándo):* Datos del momento de la conexión a la aplicación (fecha y hora, etc.).
- *Sistemas (Dónde):* Aplicación en la cual realizó acciones el usuario.
- *Datos del usuario (Quién):* Esto solo sería en el caso de que el usuario este previamente registrado en la aplicación, se registraría el nombre y otros datos

La siguiente figura ilustra lo anteriormente expresado:

Figura 1 (Conceptos presentes en los Logs de Auditoría)

Fundamentación teórica

Administración de perfil: Es la forma de personalización de las aplicaciones de este dominio a nivel de cada usuario, se define como perfil los datos únicos de cada recurso dentro del sistema que define el comportamiento del mismo ante las entradas emitidas por este recurso y las salidas entregadas por el (los) subsistemas, esto garantizaría un sin número de bondades tanto de usabilidad como de configurabilidad a la solución en cuestión.

La **administración de conexiones** consiste en un grupo de procesos dedicados a la gestión de las conexiones a la base de datos de un sistema determinado ubicado en un servidor de bases de datos definido también como un parámetro configurable, así como el gestor en uso.

La **confidencialidad** es la propiedad de la seguridad que permite mantener en secreto la información y solo los usuarios autorizados pueden manipularla. Igual que antes, los usuarios pueden ser personas, procesos, programas.

Para evitar que personal no autorizado pueda tener acceso a la información transferida y que recorra la Red se utilizan técnicas de encriptación o codificación de datos.

Hay que mantener una cierta coherencia para determinar cuál es el grado de confidencialidad de la información que se está manejando, para así evitar un esfuerzo suplementario a la hora de decodificar una información previamente codificada.

La **integridad** de la información corresponde a lograr que la información transmitida entre dos entidades no sea modificada por un tercero y esto se logra mediante la utilización de firmas digitales.

Mediante una firma digital se codifican los mensajes a transferir, de forma que una función, denominada hash, calcula un resumen de dicho mensaje y se añade al mismo.

La validación de la integridad del mensaje se realiza aplicándole al original la misma función y comparando el resultado con el resumen que se añadió al final del mismo cuando se calculó por primera vez antes de enviarlo.

Mantener la integridad es importante para verificar que en el tiempo de viaje de la información por la Red entre el sitio emisor y receptor el mensaje no ha sido modificado por personal no autorizado.

Los servicios de **no-repudio** ofrecen una prueba al emisor de que la información fue entregada y una prueba al receptor del origen de la información recibida.

Con este aspecto se consigue que una vez que alguien ha mandado un mensaje no pueda renegar de él, es decir, no pueda negar que es el autor del mensaje.

Para la realización de los procesos de un sistema como el ERP es importante ya que garantiza la realización de las transacciones para las entidades o subsistemas participantes.

Se aplica en ambos lados de la comunicación, tanto para no poder rechazar la autoría de un mensaje, como para negar su recepción.

Es necesario identificar la información que debe conocer cada una de las entidades participantes en el proceso y con ello permitir la privacidad a las partes autorizadas para su uso.

1.2.1 Tipos de ataques.

Un ataque no es más que la realización de una amenaza. Se entiende por amenaza a la condición del entorno del sistema de información (persona, máquina, suceso o idea) que, dada una oportunidad, podría dar lugar a que ocurra una violación de la seguridad (confidencialidad, integridad, disponibilidad, autenticidad). Las cuatro categorías generales de amenazas o ataques son las siguientes:

Intercepción: Una entidad no autorizada consigue acceso a un recurso. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa o un ordenador. Ejemplos de este ataque son pinchar una línea para tomar los datos que circulen por la red y la copia ilícita de ficheros o programas (intercepción de datos), o bien la lectura de las cabeceras de paquetes para desvelar la identidad de uno o más de los usuarios implicados en la comunicación observada ilegalmente (intercepción de identidad). (2)

Modificación: Una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad. Ejemplos de este ataque es el cambio de valores en un archivo de datos, alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que están siendo transferidos por la red. (2)

Fundamentación teórica

Interrupción: Un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son la destrucción de un elemento hardware, como un disco duro, cortar una línea de comunicación o deshabilitar el sistema de gestión de ficheros. (2)

Fabricación: Una entidad no autorizada inserta objetos falsificados en el sistema. Este es un ataque contra la autenticidad. Ejemplos de este ataque son la inserción de mensajes ilegítimos en una red o añadir registros a un archivo. Estos ataques se pueden asimismo clasificar de forma útil en términos de ataques pasivos y ataques activos. (2)

Ataques pasivos: En los ataques pasivos el atacante no altera la comunicación, sino que únicamente la escucha o monitoriza, para obtener información que está siendo transmitida. Sus objetivos son la intercepción de datos y el análisis de tráfico, una técnica más sutil para obtener información de la comunicación, que puede consistir en: Obtención del origen y destinatario de la comunicación, leyendo las cabeceras de los paquetes monitorizados. Control del volumen de tráfico intercambiado entre las entidades monitorizadas, obteniendo así información acerca de actividad o inactividad inusuales. Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los períodos de actividad. Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitar su éxito mediante el cifrado de la información y otros mecanismos. (2)

Ataques activos: Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos, pudiendo subdividirse en cuatro categorías:

Suplantación de identidad: El intruso se hace pasar por una entidad diferente. Normalmente incluye alguna de las otras formas de ataque activo. Por ejemplo, secuencias de autenticación pueden ser capturadas y repetidas, permitiendo a una entidad no autorizada acceder a una serie de recursos privilegiados suplantando a la entidad que posee esos privilegios, como al robar la contraseña de acceso a una cuenta. (2)

Reactuación: Uno o varios mensajes legítimos son capturados y repetidos para producir un efecto no deseado, como por ejemplo ingresar dinero repetidas veces en una cuenta dada.

Modificación de mensajes: Una porción del mensaje legítimo es alterada, o los mensajes son retardados o reordenados, para producir un efecto no autorizado. Por ejemplo, el mensaje

"Ingresa un millón de pesos en la cuenta A" podría ser modificado para decir "Ingresa un millón de pesos en la cuenta B". (2)

Degradación fraudulenta del servicio: Impide o inhibe el uso normal o la gestión de recursos informáticos y de comunicaciones. Por ejemplo, el intruso podría suprimir todos los mensajes dirigidos a una determinada entidad o se podría interrumpir el servicio de una red inundándola con mensajes espurios. Entre estos ataques se encuentran los de denegación de servicio, que consisten en paralizar temporalmente el servicio de un servidor de correo, Web, FTP, etc. (2)

1.3 Problemática.

En la actualidad existen muchos frameworks[3] implementados en el lenguaje PHP como principales y más usados mundialmente en el desarrollo de software podemos destacar *Symfony*, *Zend Framework*, *CakePHP*, *Code Igniter* y *Kumbia* entre otros. Estos presentan herramientas que le brindan seguridad al sistema que estemos desarrollando pero ninguno de estos framework implementa una seguridad de forma centralizada en un entorno de varias aplicaciones.

1.3.1 Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, 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. (3)

Symfony cuenta con muchos plugin[4], un grupo de ellos es utilizado para gestionar la *seguridad* de las aplicaciones, dentro de este grupo el más usado, seguro y eficaz es: *sfGuardPlugin.*

[3] Un **framework**, en el desarrollo de software, es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

[4] Un **complemento** (o **plug-in** en inglés) Pequeño programa que añade alguna función a otro programa, habitualmente de mayor tamaño. Un programa puede tener uno o más conectores. Son muy utilizados en los programas navegadores para ampliar sus funcionalidades.

.

1.3.1.1 Sobre sfGuardPlugin

En toda aplicación dinámica, siempre esta presente la gestión de usuarios, permisos y roles. En Symfony esto puede resultar mu sencillo gracias a SfGuard. Este plugin gestiona de manera muy sencilla usuarios, roles, permisos y logins.

Básicamente, la autenticación y la seguridad en una aplicación es limitar el acceso a partes de la misma. Significa que los usuarios tendrán que iniciar una sesión (autenticación) para acceder a ciertas áreas (seguridad). Diferentes usuarios pueden tener diferentes privilegios (autorización). De esta manera aseguras tu aplicación para diferentes tipos de usuarios. SfGuard Plugin te brinda el modelo (usuario, grupo y objetos de permisos).

Está compuesto por cuatro módulos:

- El módulo *sfGuardAuth* es el encargado del login y el acceso restringido.
- El módulo *sfGuardGroup* es el encargado de la gestión de grupos o roles.
- El módulo *sfGuardPermission* es el encargado de la gestión de permisos.
- El módulo *sfGuardUser* es el encargado de la gestión de usuarios.

1.3.2 Zend Framework

En su nivel más simple, *Zend Framework* es una librería de componentes escritos en *PHP5*, para facilitar el desarrollo de sitios web. Como está basada en *PHP5* (5.1.4 es la versión mínima necesaria), eso significa que es completamente *Orientada a Objetos*. (4)

Zend está formado por muchos componentes, estos están divididos en grupos uno de estos grupos se encarga de gestionar la seguridad, los componentes de este grupo son:

Zend_Auth permite chequear y guardar credenciales de usuario de distintas maneras: utilizando la *Base de Datos,* el *método Diguest de Apache,* o *autenticación http simple.* Provee un Adapter de Interfaz para personalizar los mecanismos de autenticación, además almacenamiento automático de identidad para una fácil personalización. Es simple y extensible.

A su vez **Zend_Session** trabaja como un administrador de datos de sesión, al igual que en **PHP,** solo que ofrece algo de valor agregado.

El componente **Zend_ACL** es una implementación de Listas de control de acceso (ACL)[5] en PHP que nos permite asignar roles y permisos a usuarios o grupos de usuarios en la aplicación. Soporta herencia de roles y recursos. También soporta el control de acceso condicional basado en una interfaz de declaraciones.

Finalmente el componente **Zend_Log** que permite realizar un log de acciones en diferentes medios como la consola de *PHP,* archivos y base de datos mediante el anexo de varios ¨writers¨.

La clase maneja diferentes niveles de de criticidad:

- EMERG = 0; //Emergencia: el sistema esta fuera de línea.
- ALERT = 1; //Alerta: acción debe ser atendida.
- CRIT = 2; //Crítica: condición crítica.
- ERR = 3; //Error: condición de error.
- WARN = 4; //Advertencia: condición de advertencia.
- NOTICE = 5; //Notificación: normal pero con una nota.
- INFO = 6; //Información: mensajes de información.
- DEBUG = 7; //Depuración: mensajes de depuración.

1.3.3 Code Igniter

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías. *CodeIgniter* le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada. (5)

CodeIgniter gestiona la seguridad de diferentes formas. Es justamente restrictivo sobre que caracteres permitir en las cadenas URI para ayudar a minimizar la posibilidad de que datos maliciosos puedan ser pasados a su aplicación.

[5] *La **Lista de Control de Acceso** o **ACL*** (del inglés, **Access Control List**) es un concepto de seguridad informática usado para fomentar la separación de privilegios. Es una forma de determinar los permisos de acceso apropiados a un determinado objeto, dependiendo de ciertos aspectos del proceso que hace el pedido.

Las URL sólo pueden contener lo siguiente:

- Texto alfanumérico
- "Tilde": **~**
- Punto: **.**
- Dos puntos: **:**
- Guión bajo: _
- Guión: **-**

Los datos GET son simplemente anulados por *CodeIgniter* ya que el sistema utiliza segmentos URI en vez de las tradicionales query strings de URL (a menos que la opción query string esté habilitada en su archivo config). El arreglo global GET es destruido por la clase de Entrada (input) durante la inicialización del sistema. (5)

Durante la inicialización del sistema, todas las variables globales son destruidas, excepto aquellas encontradas en los arreglos *$_POST* y *$_COOKIE*. La rutina de eliminación es efectivamente lo misma que register_globals = off. (5)

La directiva *magic_quotes_runtime* es apagada durante la inicialización del sistema para que no tenga que remover las barras cuando se recuperen datos de la base de datos.

CodeIgniter viene con un filtro de ***XSS***[6] (Cross Site Scripting). Este filtro busca técnicas comúnmente usadas para embeber Javascript malicioso a sus datos, u otro tipo de código que intente "secuestrar" (hijack) cookies o hacer otra cosa maliciosa. El Filtro XSS es descripto aquí. CodeIgniter tiene una Clase de Validación que le asiste en la validación, filtro y preparación de datos. (5)

1.3.4 KumbiaPHP

KumbiaPHP es un framework para aplicaciones web libre escrito en *PHP5*. Basado en las prácticas de desarrollo web como ***DRY***[7] y el principio ***KISS***[8] para software comercial y educativo.

[6] **XSS**, del inglés ***Cross-site scripting*** es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado.

[7] El principio **No te repitas** (en inglés **Don't Repeat Yourself** o **DRY**, también conocido como **Una vez y sólo una**) es una filosofía de definición de procesos que promueve la reducción de la duplicación especialmente en informática.

Kumbia fomenta la velocidad y eficiencia en la creación y mantenimiento de aplicaciones web, reemplazando tareas de codificación repetitivas por poder, control y placer. (6)

La implementación de la seguridad usando *KumbiaPHP* se lleva a cabo con el uso de de las listas de control de acceso *(ACL).*

1.3.5 CakePHP

CakePHP es un framework para aplicaciones web escrito en PHP que provee una extensa arquitectura para el desarrollo, mantenimiento y despliegue de aplicaciones. CakePHP reduce los costos de desarrollo y ayuda a los desarrolladores a escribir menos código. (7)

Cuenta con un módulo de autenticación de usuario único llamado 'Access Lists', que se puede utilizar para dar acceso a los diferentes usuarios de diferentes partes de su sitio web con CakePHP. También cuenta con un componente de *seguridad* para proteger la aplicación de ataques de tipo Cross Site Request Forgery[9] (CSRF).

1.3 Estudio referencial.

1.4.1 Sistemas de autenticación centralizados.

En el ecosistema de software de las empresas y corporaciones de hoy en día, la seguridad de acceso a las aplicaciones de la organización constituye un creciente desafío. Implementar y poner en marcha un modelo de seguridad de acceso implica navegar múltiples aplicaciones, en la mayoría de los casos, con muy pocas características comunes entre ellas.

La norma hasta el momento ha sido que cada aplicación dentro de una empresa o corporación cuenta con un adecuado sistema de control de accesos y su propio repositorio de datos. En la actualidad dentro del ambiente de tecnología de cualquier empresa o corporación existen diversos métodos de autenticación y autorización de accesos que generan una gran

[8] El principio **Manténgalo Suave** y **Breve** (en ingles, **Keep It Short and Simple** o **KISS**) es aquel que recomienda el desarrollo empleando partes sencillas, comprensibles y con errores de fácil detección y corrección, rechazando lo enrevesado e innecesario en el desarrollo de sistemas complejos en ingeniería.

[9] El **CSRF** (del inglés ***Cross-site request forgery*** o falsificación de petición en sitios cruzados) es un tipo de exploit malicioso de un sitio web en el que comandosno autorizados son transmitidos por un usuario en el cual el sitio web confía. Esta vulnerabilidad es conocida también por otros nombres como XSRF, enlace hostil, ataque de un click, cabalgamiento de sesión, y ataque automático.

ineficiencia. Sin embargo, con la implementación de un agente Single Sign-On (SSO) el sistema se encarga de almacenar, en una base datos o directorios protegidos, las credenciales que permiten al usuario acceder a cada una de las aplicaciones o servicios en el momento que lo desee, ya que el proceso de autenticación se realiza de manera transparente para el usuario, una vez que éste ha sido autenticado por medio de la arquitectura SSO. Se puede decir que con dicha implementación el sistema simplificaría y centralizaría el control de accesos a todas las aplicaciones de la empresa o corporación, reduciendo el costo en la administración de seguridad, lo que logra un mejor rendimiento y velocidad de los procesos de autenticación y acceso que facilita a los usuarios la interacción con los sistemas de la empresa, simplificando el manejo de claves y lo fundamental es que aumenta los niveles de seguridad, ya que contará con una plataforma central para el manejo de la seguridad en todos los procesos de autenticación y acceso a sus aplicaciones. (8)

El agente SSO se refiere al acceso a múltiples recursos por medio de un único proceso de ingreso. Gran cantidad de las arquitecturas implementadas en diferentes organizaciones han sido diseñadas con el objeto de dar acceso a los usuarios a múltiples servicios Web y aplicaciones. En la mayoría de los casos se encuentra que cada uno de los servicios o aplicaciones cuenta con su propio componente de seguridad, el cual generalmente compromete la seguridad de todo el sistema, dado el nivel de seguridad del componente más débil, el cual determina el nivel de confianza del sistema en su conjunto. (8)

1.4.1.1 Tipos de sistemas Single Sign-On.

Existen cinco tipos principales de sistemas SSO, también conocidos como Reduced Sign-On Systems (Sistemas de Autenticación Reducida). Los cuales son:

- **Enterprise Single Sign-On (E-SSO)**: también llamado Legacy Single Sign-On, el cual funciona luego de una autenticación primaria, interceptando los requerimientos de autenticación presentados por las aplicaciones secundarias para completarlos con el usuario y la contraseña. Los sistemas E-SSO permiten interactuar con sistemas que pueden deshabilitar la presentación de la pantalla de login. (9)

- **Web Single Sign-On (Web-SSO)**: conocido como Web Access Management (Web-AM), trabaja sólo con aplicaciones y recursos que se acceden vía Web. Los accesos son interceptados con la ayuda de un servidor Proxy o de un componente instalado en

el servidor Web destino. Los usuarios no autenticados que tratan de acceder son redirigidos a un servidor de autenticación y regresan sólo después de haber logrado un acceso exitoso. Se utilizan cookies, para reconocer aquellos usuarios que acceden y su estado de autenticación. (9)

- **Kerberos**: es un método popular de externalizar la autenticación de los usuarios. Los usuarios se registran en el servidor Kerberos y reciben un ticket, que luego utilizan para obtener acceso. (9)
- **Federation**: es una nueva manera de concebir este tema, también para aplicaciones Web. Utiliza protocolos basados en estándares para habilitar que las aplicaciones puedan identificar los clientes sin necesidad de autenticación redundante. (9)
- **OpenID**: es un proceso de SSO distribuido y descentralizado donde la identidad se compila en una URL de forma que cualquier aplicación o servidor puede verificar. (9)

1.4.2 Sistemas de gestión de seguridad utilizados actualmente a nivel mundial.

Actualmente existen soluciones para el control de la seguridad de varias aplicaciones de manera centralizada, o sea, en un entorno de varias aplicaciones controlarlas a todas de igual forma, pero hay que destacar que estas propuestas son incipientes todavía por lo que en muchas ocasiones son miradas con recelos por los clientes. Una de las soluciones que proponen una seguridad centralizada es: AccessMaster IAM (Gestión de Identidades y Acceso) & SSO (Single Sign-On).

Los sistemas de información corporativos incluyen un número creciente de aplicaciones heterogéneas y recursos alojados en diversos sistemas abiertos. Mientras esta variedad y heterogeneidad facilita los procesos de negocio, también constituyen un problema desde el punto de vista de la gestión de la seguridad. Se plantea la dificultad de definir e implantar una política de seguridad única, que sea aplicable a todos esos recursos y aplicaciones. El principio de las soluciones de Gestión de Identidades y Accesos (AccessMaster IAM) es el poder establecer, mediante políticas de control de acceso, qué usuarios pueden acceder a que aplicaciones y recursos, de manera que un usuario no pueda usar aplicaciones o entrar en recursos para los que no está autorizado. Otro valor añadido de esta solución es la gestión centralizada y segura de usuarios y contraseñas para los distintos servicios de la organización. El Single Sign-On (SSO) supone efectuar en lugar del usuario la operación de identificarse

mediante login y password frente a las aplicaciones y recursos corporativos. Los usuarios se autentican una única vez, contra el sistema de IAM y SSO, y después este sistema se encarga de forma transparente de las autenticaciones subsiguientes en su lugar, según se van produciendo los accesos correspondientes.

Otro ejemplo es la plataforma v-Go Single ON, producida por la compañía de Passlogix. Es uno de los sistemas de tipo SSO más populares y óptimos que se han implementado.

1.4.3 Sistemas de gestión de seguridad utilizados actualmente en Cuba y en la UCI.

La investigación en la UCI sobre los softwares de gestión de seguridad arrojó como resultado la existencia del Sistema de Gestión de Sesiones basado en la arquitectura SSO cuyo objetivo es gestionar las sesiones de los usuarios en un único proceso de autenticación invisible a los ojos de los mismos, a través de un sistema con una fachada de servicios web que sea capaz de gestionar la apertura y cierre de sesiones por parte de las personas que trabajan en el dominio **uci.cu**.

Además en la facultad 10 se llevó a cabo el desarrollo de un software de autenticación y control centralizado para la corporación de PDVSA capaz de mapear la credenciales de los usuarios hacia todas las aplicaciones de dicho sistema y otros sistemas heterogéneos, proporcionando una infraestructura para simular un login único para el usuario corporativo frente a una plataforma tecnológica heterogénea dentro del ambiente de aplicativos de integración, en que los usuarios puedan acceder a diferentes aplicaciones con solo un conjunto de credenciales.Esta aplicación trajo como resultado principal la importancia de mapeos entre diferentes entornos que manejan diferentes mecanismos de seguridad.

1.5 Elementos de reutilización.

En este epígrafe se presenta una tabla en la cual se hace una representación de los diferentes elementos que se reutilizaron (Fragmentos de código, ideas, sistemas, componentes) en el desarrollo del sistema.

Proceso	Reutilización	Código, contexto, ideas	Objetivo	%
Construcción de la interfaz visual	Si	Utilización del framework Ext-Js	Utilizar los artefactos y las herramientas que este brinda para una mejor construcción de la interfaz visual.	100
Acceso a datos	Si	Utilización del framework Doctrine	Utilizar las potencialidades que este posee en el manejo de datos.	100
Lógica de negocio	Si	Utilización del Zend Framework	Utilizar las potencialidades que posee este framework en la implementación de la lógica de negocio.	100
Gestión del multilenguaje	Si	Generador de JSON	Utilizar esta herramienta para la creación de etiquetas dinámicas para el manejo del multilenguaje.	100
Mapeo de la base de datos	Si	Mapeador de doctrine	Utilizar esta herramienta para el mapeo de la base de datos.	100

Tabla 1 (Elementos de reutilización utilizados en la construcción del sistema)

1.6 Resultados esperados u objetivos de la solución propuesta.

Como resultados esperados u objetivos de la solución propuesta, se obtiene un sistema de seguridad el cual se divide en 4 módulos, *Configurar nomencladores*, *Configurar sistemas*, *Configurar servidores* y *Configurar usuario*s. El módulo *Configurar nomencladores* permite el

manejo de los dominios, base de datos, gestores de base de datos, esquemas, idiomas, temas, escritorios, expresiones y claves, para el manejo posterior de los servidores, sistemas y usuarios. El módulo *Configurar sistemas* posee las funcionalidades correspondientes al manejo de los sistemas, las funcionalidades, acciones, servicios que brinda ó consume sus funciones y parámetros de las mismas. El módulo *Configurar usuarios* permite la gestión de los usuarios, roles, perfiles de usuario y los campos del perfil de usuario. El módulo *Configurar servidores* permite manejar los datos de los servidores, gestores de base de datos, base de datos y esquemas de base de datos.

Además se brinda un Documento que permite conocer los requisitos funcionales y no funcionales más importantes que brinda dicho sistema, también un Manual de Usuarios para brindar un mejor entendimiento y operabilidad de los casos de usos existentes dentro del sistema.

Constituyendo todos estos elementos una gama de facilidades para garantizar la seguridad de forma centralizada a cualquier sistema al que se le brinde servicios.

1.7 Conclusiones

La investigación realizada sobre la gestión de la seguridad en los frameworks implementados en PHP, arrojó como resultado que cada uno de ellos gestionaba la seguridad de forma diferente, sin garantizar que la misma se administrara de forma centralizada, por lo que la implementación de esta tendría que ser adaptada a las características del sistema a desarrollar.Además se debían configurar algunos ficheros manualmente, si este proceso fuese realizado por desarrolladores sin experiencia provocaría que se dejasen brechas de seguridad.

Los sistemas de seguridad existentes estudiados están basados en la arquitectura SSO, solo garantizan la gestión de sesiones. Los desarrollados fuera del país son muy costosos por ejemplo la plataforma v-Go Single ON, producida por la compañía de Passlogix, tiene precios muy altos, la distribución de esta plataforma en conjunto con IBM tiene un costo de 75 USD por usuario que incluye un valor de licencia única y su mantenimiento por un año, si se compra su versión multilingüe y multiplataforma que incluye el CD-ROM de instalación costará entonces 140 USD por usuario y los servicios de instalación y configuración tendrán un valor de 3.60 USD por usuario.

Fundamentación teórica

En la Universidad de las Ciencias Informáticas los sistemas de seguridad están desarrollados basándose en la arquitectura SSO, por lo que ninguno trataba el tema de la multientidad.

Debido a la no existencia en la UCI de un sistema que gestionase la seguridad de forma centralizada, que tratase el tema de la multientidad, administrase las conexiones a la base de datos, permitiese al usuario la configuración de su perfil y garantizase la integración con otros sistemas, se decidió por parte de la dirección del proyecto Cedrux el desarrollo del Sistema de Gestión Integral de seguridad.

2

Capítulo

Desarrollo de la solución

CAPÍTULO 2: Desarrollo de la solución.

2.1 Introducción

En este capítulo se hace un levantamiento de los requisitos funcionales y no funcionales del componente desarrollado. También una muestra del prototipo de interfaz y el diseño de clases del mismo, y además los artefactos de implementación y diversos casos de prueba realizados. Para lograr una mejor comprensión de la funcionalidad e integralidad del sistema.

2.2 Requisitos

En el (Anexo 1: Requisitos funcionales del Sistema de Gestión Integral de Seguridad (SIGIS)) se muestra una descripción detallada de los requisitos funcionales de SIGIS.

2.2.1 Requisitos funcionales

El sistema está compuesto por cuatro módulos, seguidamente se plantearan los requisitos del sistema correspondientes a cada módulo.

Requisitos funcionales del módulo Configurar Sistemas

R1 Requisito Funcional Gestionar Sistema

- R1.1 Cargar Sistemas
- R1.2 Registrar Sistema
- R1.3 Modificar Sistema
- R1.4 Eliminar Sistema
- R1.5 Importar Sistema
- R1.6 Exportar Sistema

R2 Requisito Funcional Gestionar Funcionalidad

- R2.1 Cargar Sistemas
- R2.2 Registrar Funcionalidad
- R2.3 Modificar Funcionalidad
- R2.4 Eliminar Funcionalidad
- R2.5 Buscar Funcionalidad

R3 Requisito Funcional Gestionar Acciones

- R3.1 Cargar Sistemas
- R3.2 Registrar Acción
- R3.3 Modificar Acción
- R3.4 Eliminar Acción
- R3.5 Buscar Acción

R4 Requisito Funcional Gestionar servicios que presta un sistema

- R4.1 Cargar Sistemas
- R4.2 Registrar Servicio
- R4.3 Modificar Servicio
- R4.4 Eliminar Servicio

R5 Requisito Funcional Gestionar servicios que consume un sistema

- R5.1 Registrar Servicio
- R5.2 Eliminar Servicio

R6 Requisito Funcional Gestionar las funciones de un servicio

- R6.1 Cargar Servicios
- R6.2 Registrar Función
- R6.3 Modificar Función
- R6.4 Eliminar Función

R7 Requisito Funcional Gestionar parámetros de una acción

- R7.1 Cargar Servicios
- R7.2 Registrar Parámetro
- R7.3 Modificar Parámetro
- R7.4 Eliminar Parámetro

Requisitos funcionales del módulo Configurar Servidores

R8 Requisito Funcional Gestionar Servidor

- R8.1 Cargar Servidores
- R8.2 Registrar Dominio
- R8.3 Modificar Servidores
- R8.4 Eliminar Servidores

R9 Requisito Funcional Gestionar gestor de BD

- R9.1 Cargar Servidores
- R9.2 Registrar Gestores de BD

- R9.3 Eliminar Gestores de BD

R10 Requisito Funcional Gestionar Bases de Datos

- R10.1 Cargar servidores existentes
- R10.2 Asignar las BD que usará un gestor de un servidor determinado
- R10.3 Eliminar BD

R11 Requisito Funcional Gestionar Esquemas de bases de datos

- R11.1 Cargar Servidores
- R11.2 Asignar los esquemas que utilizará una BD en un gestor de un servidor determinado
- R11.3 Eliminar Esquema

Requisitos funcionales del módulo Configurar Usuarios

R12 Requisito Funcional Gestionar Roles

- R12.1 Cargar Sistemas
- R12.2 Registrar Rol
- R12.3 Modificar Rol
- R12.4 Regular Acciones
- R12.5 Eliminar Rol

R13 Requisito Funcional Gestionar Usuarios

- R13.1 Cargar Usuarios
- R13.2 Registrar Usuario
- R13.3 Modificar Usuario
- R13.4 Eliminar Usuario
- R13.5 Asignar Roles
- R13.6 Cambiar Contraseña

R14 Requisito Funcional Gestionar Campos del perfil de usuario

- R14.1 Cargar campos del perfil de usuario
- R14.2 Registrar campo de perfil de usuario
- R14.3 Modificar campo de perfil de usuario
- R14.4 Eliminar campo de perfil de usuario

R15 Requisito Funcional Gestionar Perfil de usuario

- R15.1 Registrar perfil de usuario
- R15.2 Modificar perfil de usuario

Requisitos funcionales del módulo Configurar Nomencladores

R16 Requisito Funcional Gestionar nomenclador de dominios

- R16.1 Cargar nomenclador de Dominio
- R16.2 Registrar nomenclador de Dominio
- R16.3 Modificar nomenclador de Dominio
- R16.4 Eliminar nomenclador de Dominio

R17 Requisito Funcional Gestionar nomenclador de expresiones regulares

- R17.1 Cargar nomenclador de Expresiones Regulares
- R17.2 Registrar nomenclador de Expresiones Regulares
- R17.3 Modificar nomenclador de Expresiones Regulares
- R17.4 Eliminar nomenclador de Expresiones Regulares

R18 Requisito Funcional Gestionar nomenclador de claves

- R18.1 Cargar nomenclador de Claves
- R18.2 Registrar nomenclador de Claves
- R18.3 Modificar nomenclador de Claves

R19 Requisito Funcional Gestionar nomenclador de Gestores de BD

- R19.1 Cargar nomenclador de gestores de BD
- R19.2 Registrar nomenclador de Gestores de BD
- R19.3 Modificar nomenclador de Gestores de BD
- R19.4 Eliminar nomenclador de Gestores de BD
- R19.5 Buscar nomenclador de Gestores de BD

R20 Requisito Funcional Gestionar nomenclador de BD

- R20.1 Cargar nomenclador de BD
- R20.2 Registrar nomenclador de BD
- R20.3 Modificar nomenclador de BD
- R20.4 Eliminar nomenclador de BD
- R20.5 Buscar nomenclador de BD

R21 Requisito Funcional Gestionar nomenclador de Esquemas

- R21.1 Cargar nomenclador de esquema
- R21.2 Registrar nomenclador de esquema
- R21.3 Modificar nomenclador de esquema
- R21.4 Eliminar nomenclador de esquema
- R21.5 Buscar nomenclador de esquema

R22 Requisito Funcional Gestionar nomenclador de Idiomas

- R22.1 Cargar nomenclador de idiomas
- R22.2 Registrar nomenclador de idiomas
- R22.3 Modificar nomenclador de idiomas
- R22.4 Eliminar nomenclador de idiomas

R23 Requisito Funcional Gestionar nomenclador de Temas

- R23.1 Cargar nomenclador de temas
- R23.2 Registrar nomenclador de temas
- R23.3 Modificar nomenclador de temas
- R23.4 Eliminar nomenclador de temas

R24 Requisito Funcional Gestionar nomenclador de Escritorios

- R24.1 Cargar nomenclador de escritorios
- R24.2 Registrar nomenclador de escritorios
- R24.3 Modificar nomenclador de escritorios
- R24.4 Eliminar nomenclador de escritorios

Otros requisitos funcionales del sistema

R25 Requisito Funcional brindar servicio para autenticar un usuario en el marco de trabajo utilizando LDAP[10] o según la información registrada en el sistema.

- R25.1 Especificación del requisito recibir petición con parámetros: usuario a autenticar, contraseña.
- R25.2 Especificación del requisito registrar comprobar validez de datos suministrados y caducidad de la contraseña.

R26 Requisito funcional brindar servicio de cargar todos los sistemas a los que un usuario tiene acceso.

- R26.1 Especificación del Requisito Recibir petición con parámetros.
- R26.2 Especificación del requisito comprobar validez y correspondencia de los datos suministrados.

2.2.2 Requisitos no funcionales

Usabilidad (USB)

El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.

[10] **LDAP** (*Lightweight Directory Access Protocol*), (Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Rendimiento (REN)

Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones.

Seguridad (SEG)

Autenticación (Contraseña de acceso). Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. La atención al sistema incluyendo el mantenimiento de las bases de datos así como la salva de la información se realizara de forma centralizada por el administrador.

Portabilidad (POR)

El sistema debe ser multiplataforma, haciendo énfasis en Linux y Windows.

Soporte (SOP)

La aplicación contará antes de su puesta en marcha con un período de pruebas, se le dará mantenimiento, configuración y se brindará el servicio de instalación.

Políticos culturales (CUL)

El sistema solo podrá ser utilizado en territorio cubano y por las entidades autorizadas por el Ministerio de las FAR. El producto no debe contener palabras en otros idiomas. El producto debe respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

Legales (LEG)

El sistema está avalado por los tres documentos rectores emitidos en el país para la certificación y validación de los sistemas contables:

- La Resolución Orden #4 del Ministro de las Fuerzas Armadas Revolucionarias.

Software (SFT)

Para el cliente:

- Navegador Mozilla Firefox.

- Sistema operativo Windows 98 o superior o Linux.
- Para el servidor:
- Sistema operativo Windows Advancer Server (2000 o superior) o Linux en cualquiera de sus distribuciones.
- Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible, este debe estar configurado con la extensión “pgsql” incluida.
- Un servidor de base de datos PostgreSQL 8.0 o superior.

Hardware (HDW)

Para el servidor:

- Requerimientos mínimos: Procesador Pentium III a 1GHz de velocidad de procesamiento y 1Gb de memoria RAM.
- Al menos 40Gb de espacio libre en disco duro.
- Tarjeta de red.

Para el cliente:

- Requerimientos mínimos: Procesador Pentium II a 133Mhz con 128 Mb de memoria RAM.
- Tarjeta de red.

2.3 Prototipo de interfaz de usuario.

En este epígrafe se relacionan los requisitos funcionales establecidos por cada componente en cada escenario.

2.3.1 Requisito funcional R1 Gestionar Sistema.

En el SIGIS una de las actividades más importantes que se realiza es la de gestionar los sistemas ya que este es el encargado de garantizar la seguridad a todos lo sistemas que se suscriban a él.

En este prototipo de interfaz están representados los requisitos R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7

Figura 2 (Prototipo interfaz para requisitos R1.1, R1.2, R1.3, R1.4, R1.5, R1.6, R1.7)

2.3.2 Requisito funcional R2 Gestionar Funcionalidad.

De cada uno de los sistemas que estarán suscritos al SIGIS se controlarán las funcionalidades que estos brindan.

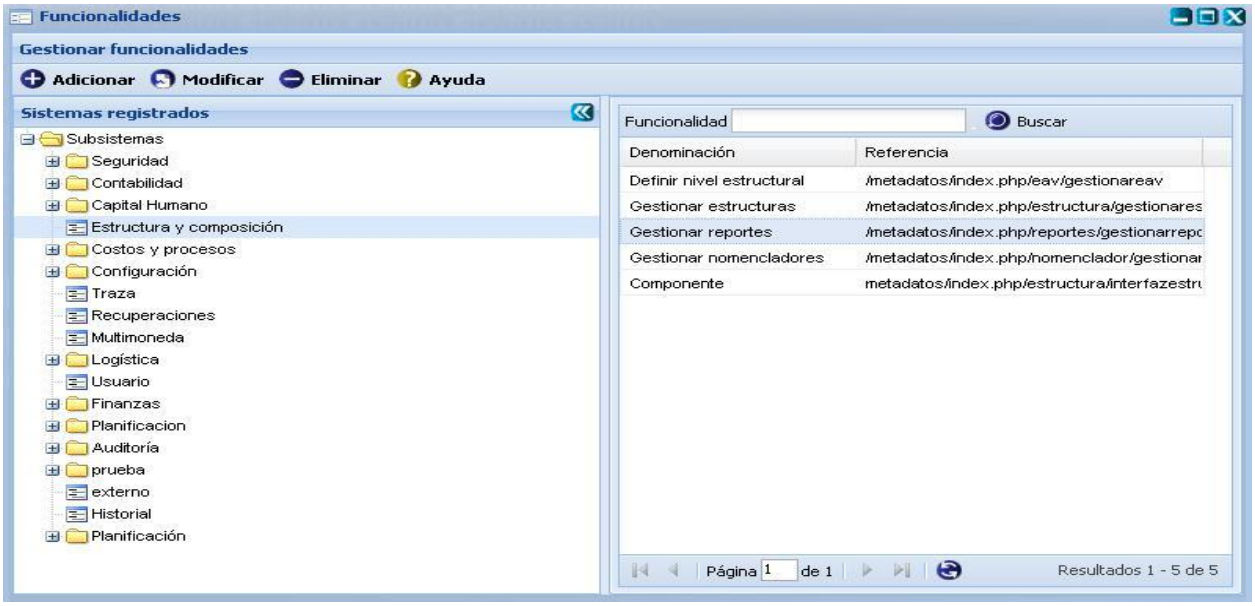

Figura 3 (Prototipo interfaz para requisitos R2.1, R2.2, R2.3, R2.4, R2.5)

2.3.3 Requisito funcional R3 Gestionar Acciones.

Cada una de las funcionalidades que tienen los sistemas puede tener un grupo de acciones registradas.

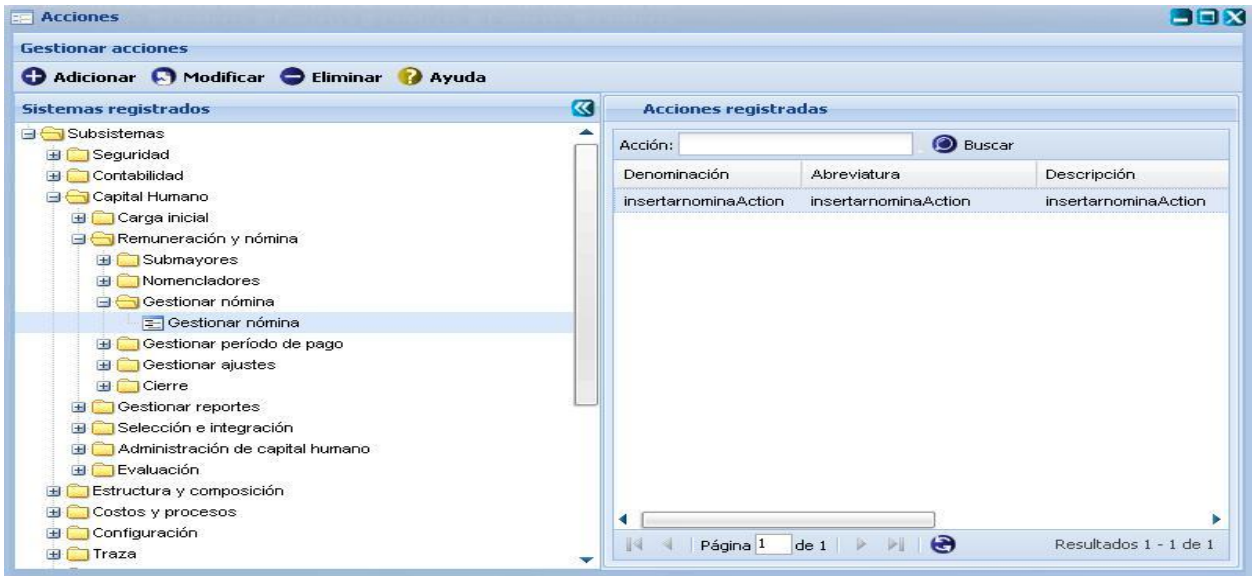

Figura 4 (Prototipo interfaz para requisitos R3.1, R3.2, R3.3, R3.4, R3.5)

2.3.4 Requisito funcional R4 Gestionar Servicios que presta un sistema.

El sistema controla los servicios que son brindados por los sistemas suscritos al mismo.

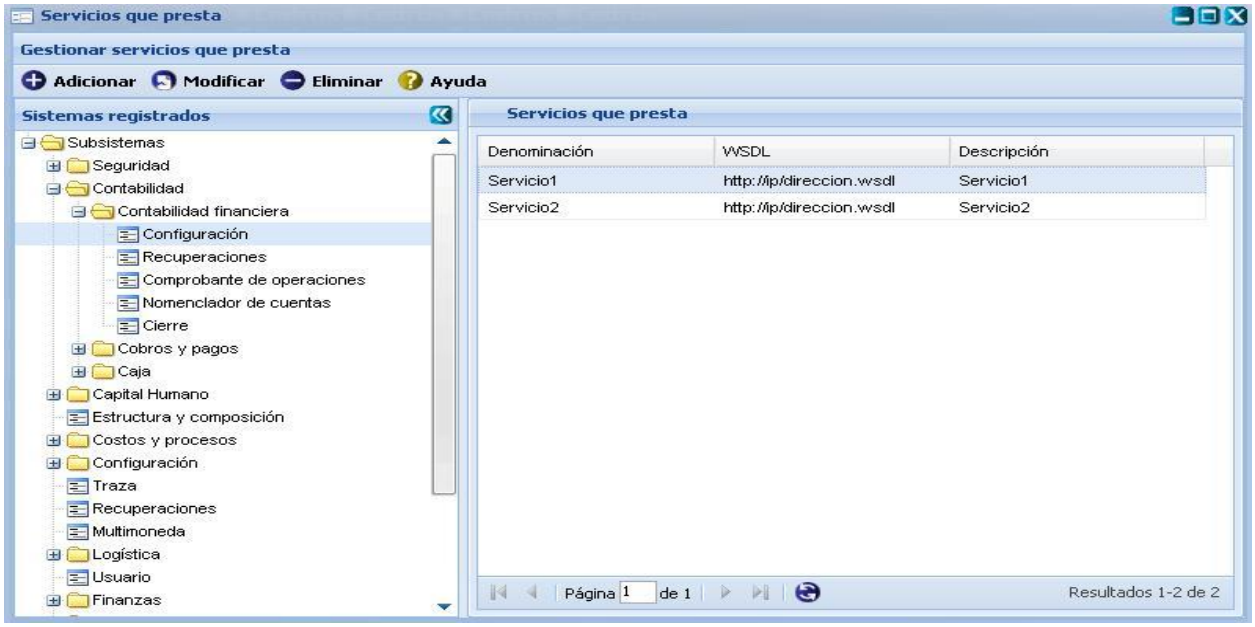

Figura 5 (Prototipo interfaz para requisitos R4.1, R4.2, R4.3, R4.4)

2.3.5 Requisito funcional R5 Gestionar Servicios que consume un sistema.

El sistema controla los servicios que son consumidos por los sistemas suscritos al mismo.

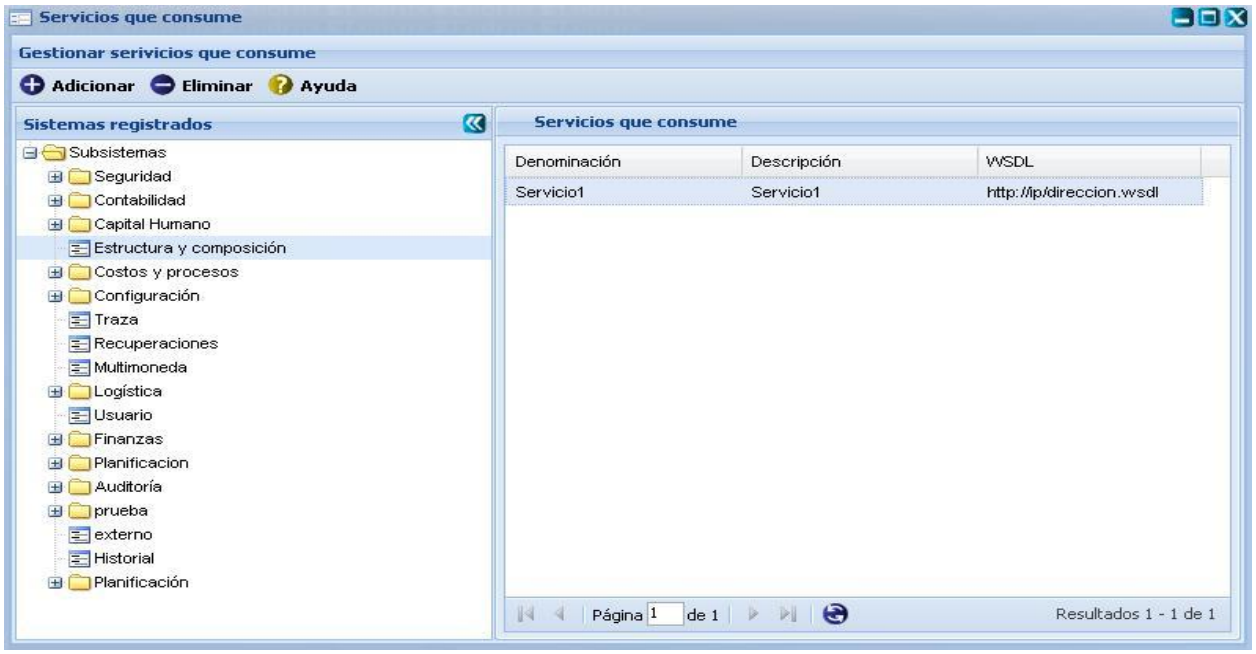

Figura 6 (Prototipo interfaz para requisitos R5.1, R5.2)

2.3.6 Requisito funcional R6 Gestionar funciones de un servicio.

El sistema controla las funciones que tiene cada servicio.

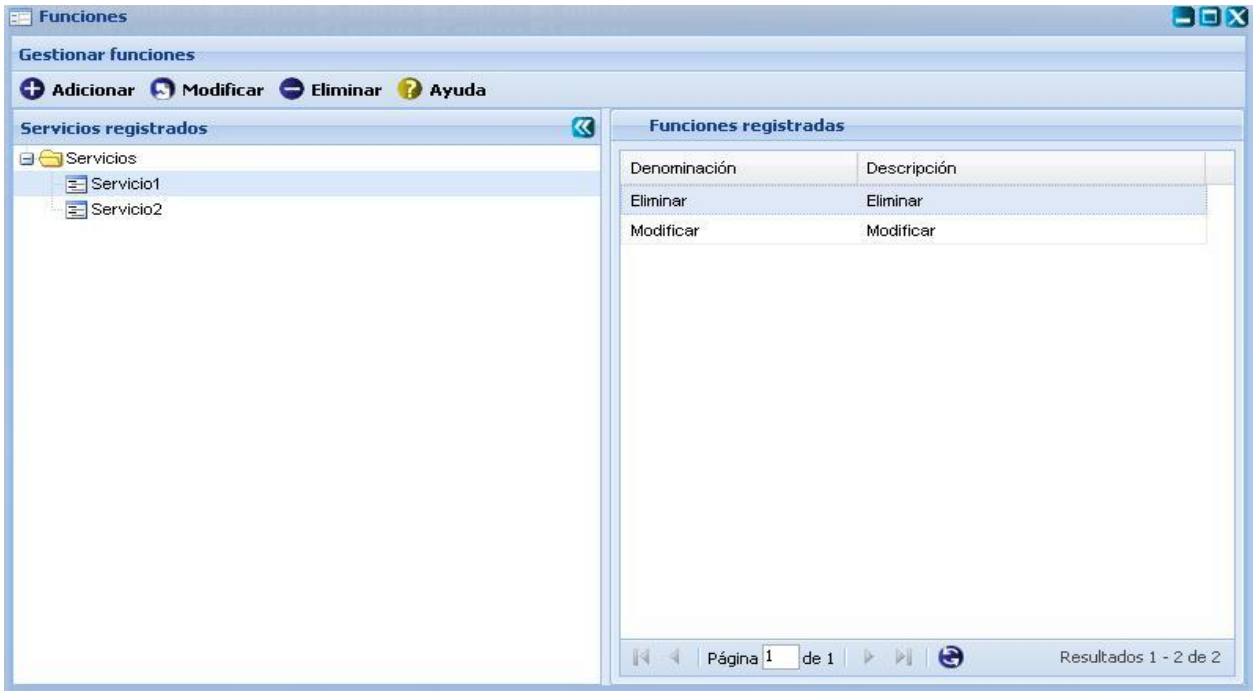

Figura 7 (Prototipo interfaz para requisitos R6.1, R6.2, R6.3, R6.4)

2.3.7 Requisito funcional R7 Gestionar parámetros.

El sistema controla todos los parámetros correspondientes a las funciones de cada servicio.

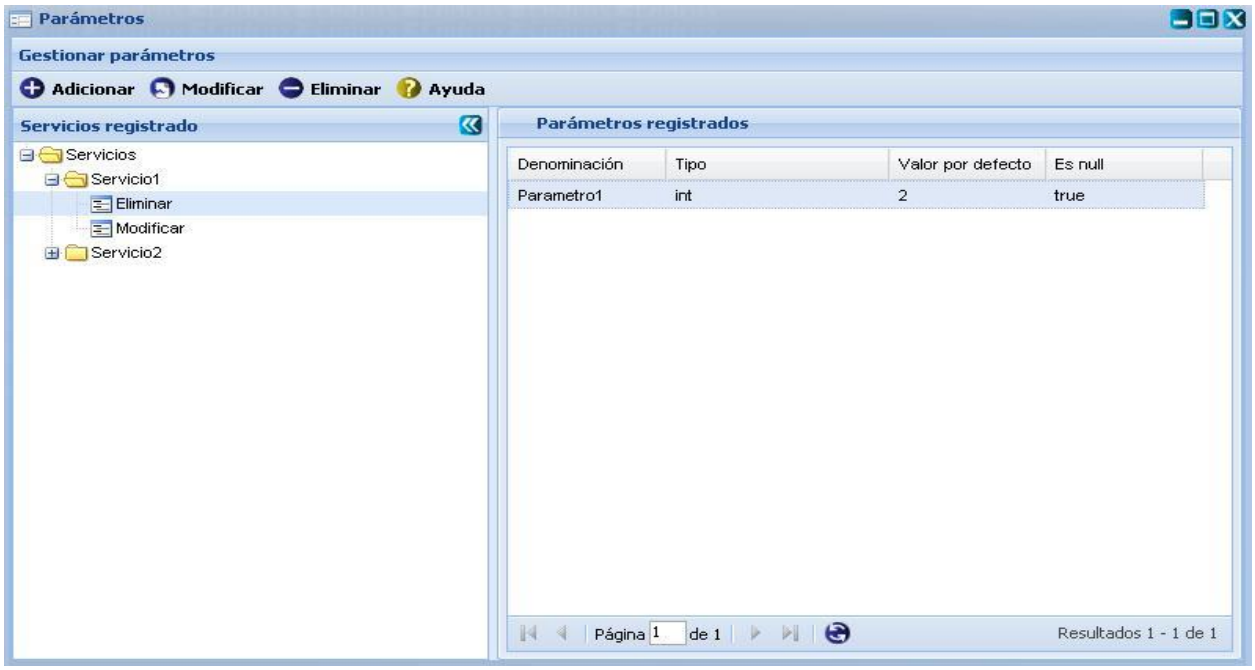

Figura 8 (Prototipo interfaz para requisitos R7.1, R7.2, R7.3, R7.4)

2.3.8 Requisito funcional R8 Gestionar servidores.

El sistema gestiona los servidores de los sistemas suscritos a él.

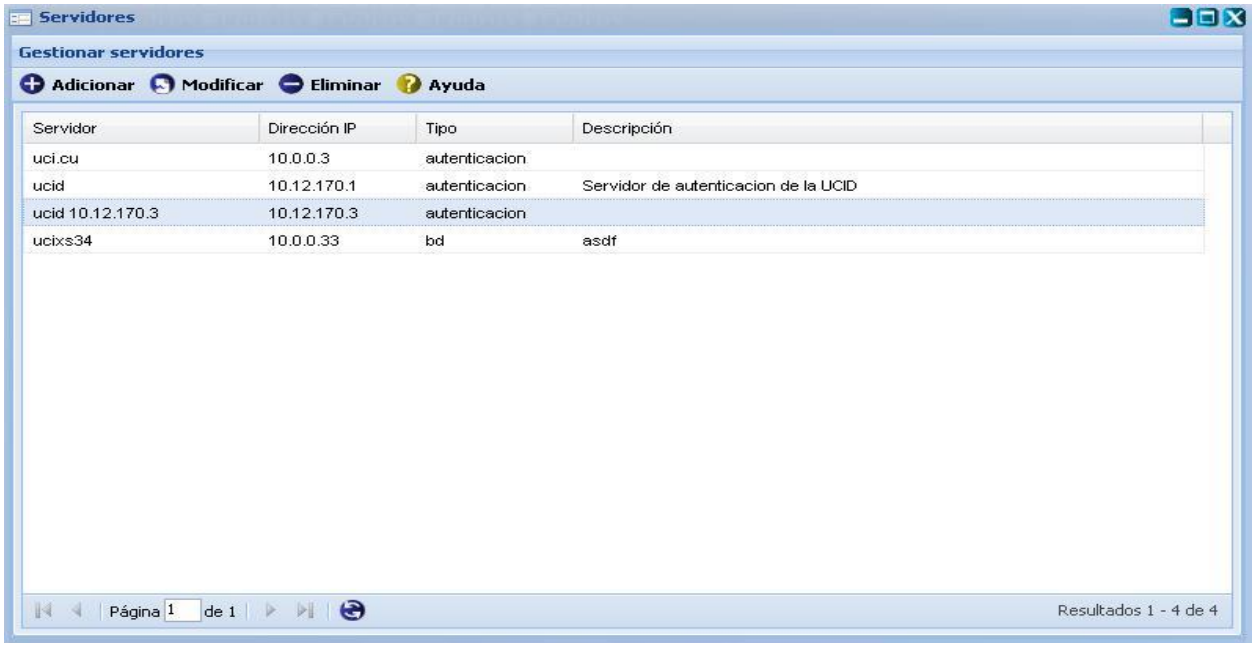

Figura 9 (Prototipo interfaz para requisitos R8.1, R8.2, R8.3, R8.4)

2.3.9 Requisito funcional R9 Gestionar gestores de BD.

El sistema gestiona los gestores de BD que serán utilizados en cada sistema.

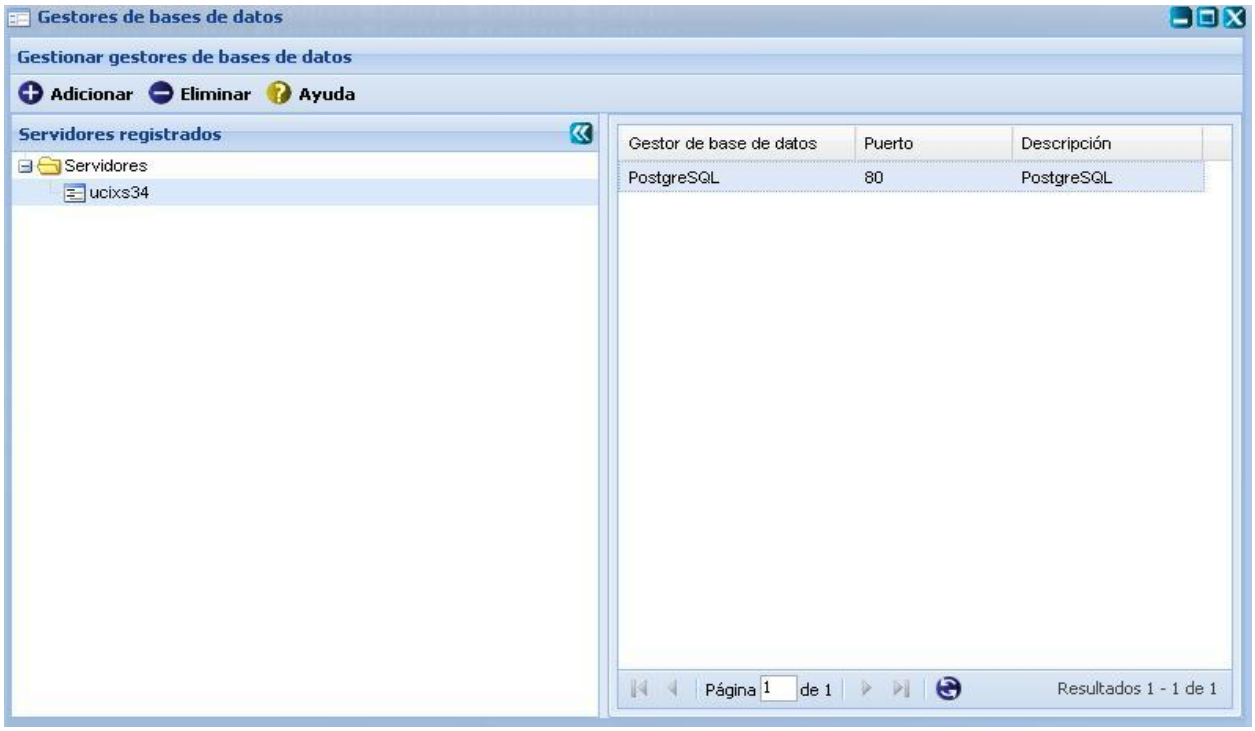

Figura 10 (Prototipo interfaz para requisitos R9.1, R9.2, R9.3)

2.3.10 Requisito funcional R10 Gestionar bases de datos.

El sistema gestiona las bases de datos que serán utilizadas por cada sistema.

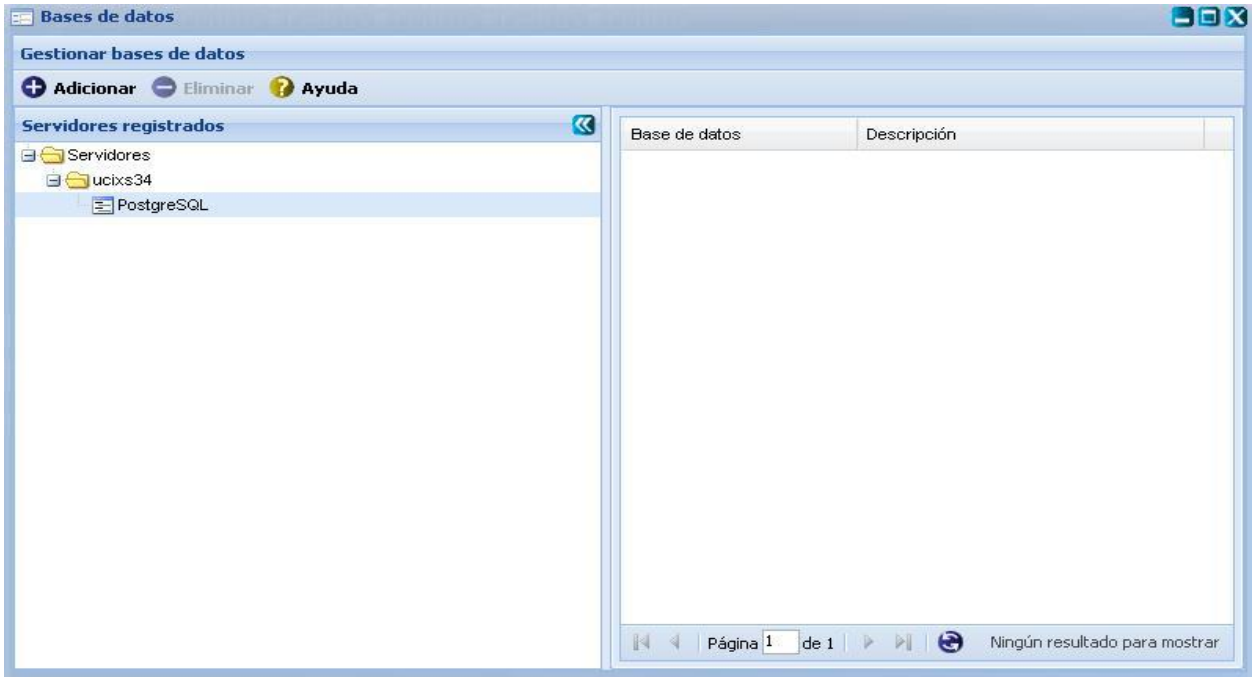

Figura 11 (Prototipo interfaz para requisitos R10.1, R10.2, R10.3)

2.3.11 Requisito funcional R11 Gestionar esquemas de bases de datos.

El sistema gestiona los diferentes esquemas de base de datos utilizados por cada sistema.

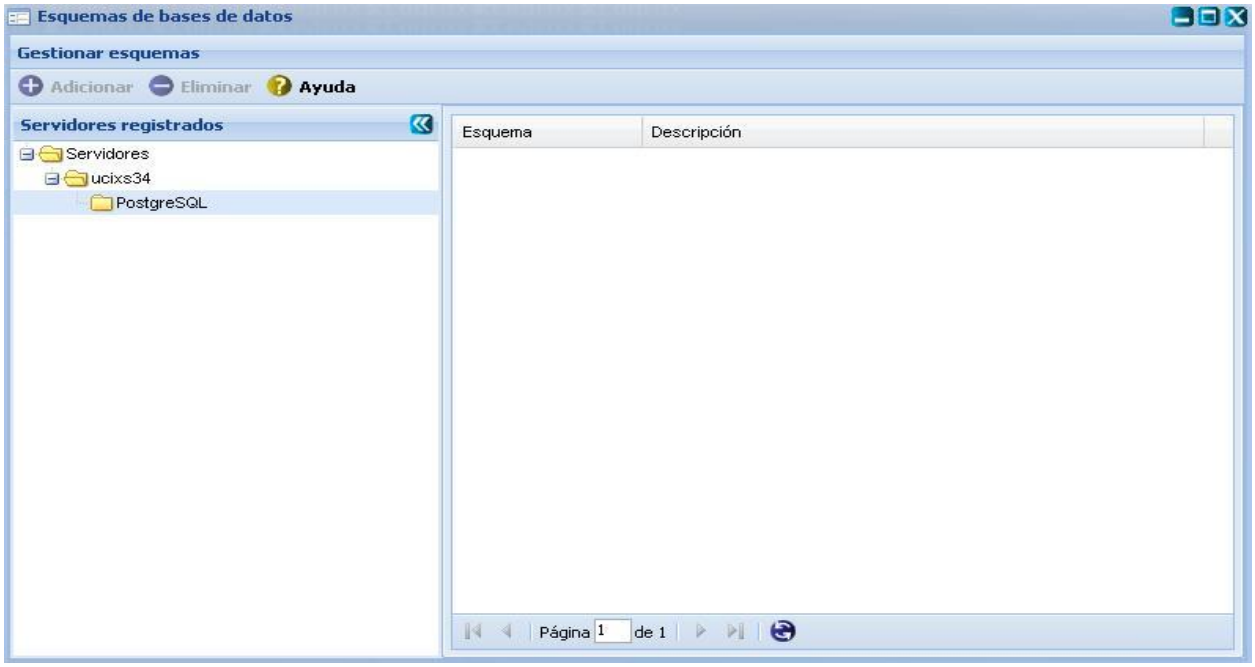

Figura 12 (Prototipo interfaz para requisitos R11.1, R11.2, R11.3)

2.3.12 Requisito funcional R12 Gestionar roles.

El sistema gestiona los roles de usuarios para controlar el acceso de los usuarios a cada uno de los sistemas y funcionalidades.

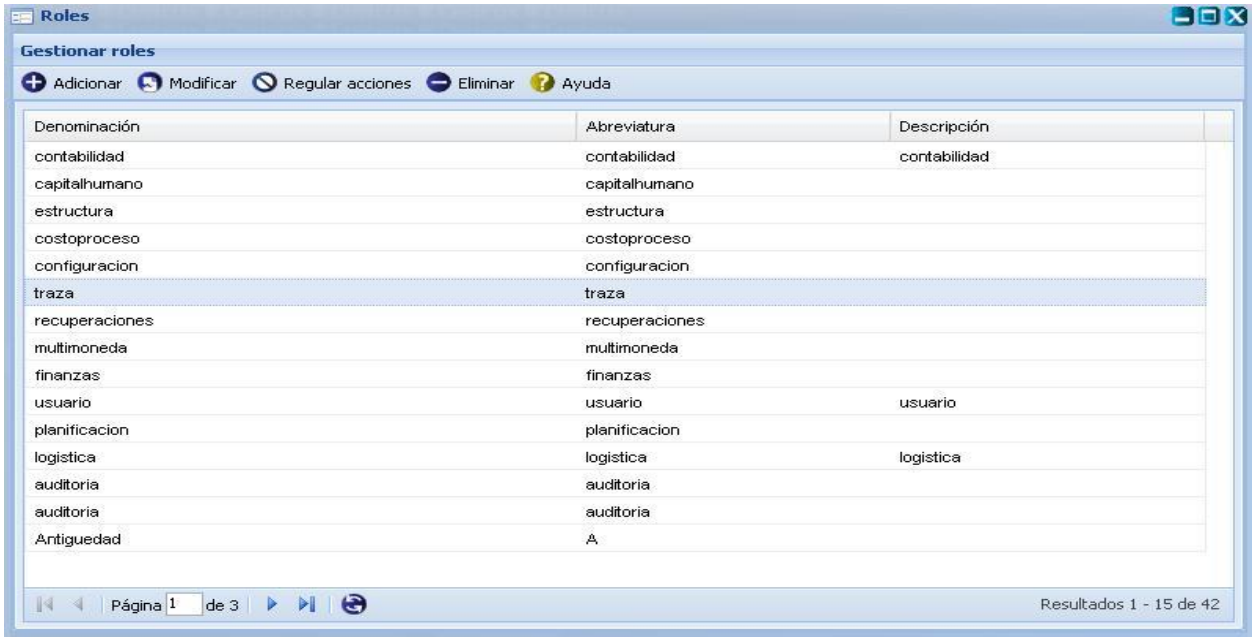

Figura 13 (Prototipo interfaz para requisitos R12.1, R12.2, R12.3, R12.4, R12.5)

2.3.13 Requisito funcional R13 Gestionar usuarios.

El sistema gestiona toda la información de los usuarios de los sistemas suscritos a él.

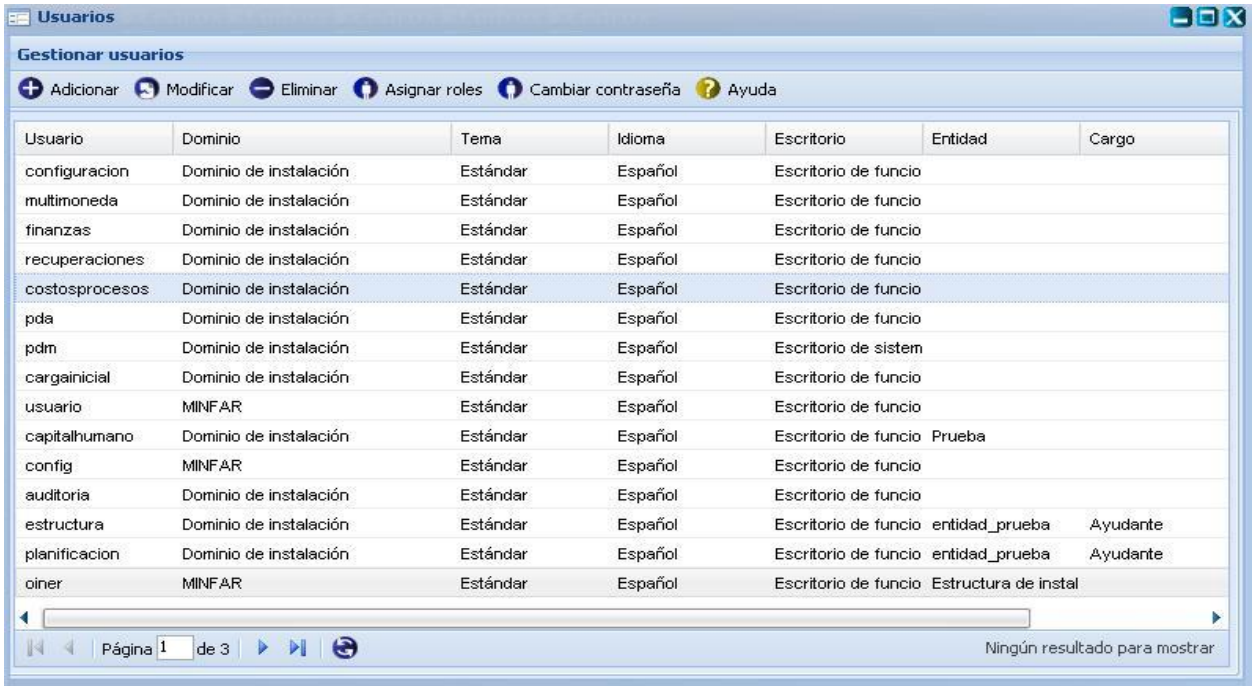

Figura 14 (Prototipo interfaz para requisitos R13.1, R13.2, R13.3, R13.4, R13.5, R13.6)

2.3.14 Requisito funcional R14 Gestionar Campos de perfil de usuario.

El sistema gestiona los campos que se quieran utilizar cuando estemos configurando el perfil de usuario.

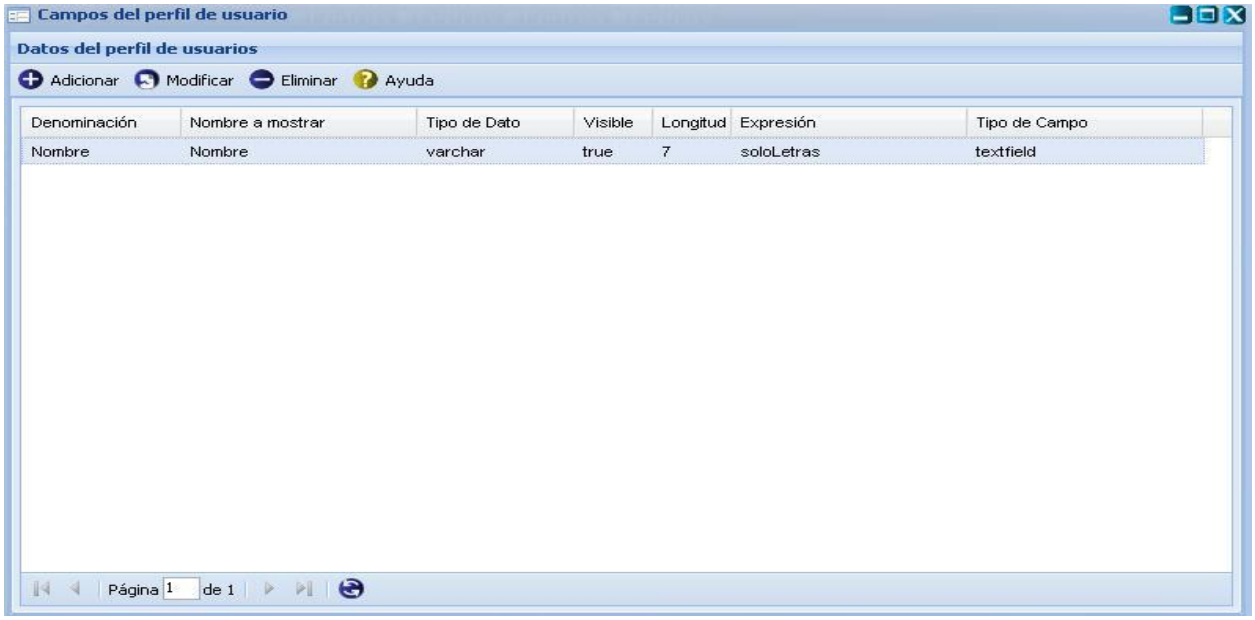

Figura 15 (Prototipo interfaz para requisitos R14.1, R14.2, R14.3, R14.4)

2.3.15 Requisito funcional R15 Gestionar Perfil de usuario.

El sistema gestiona el perfil de usuario de los usuarios pertenecientes a los sistemas suscritos a él.

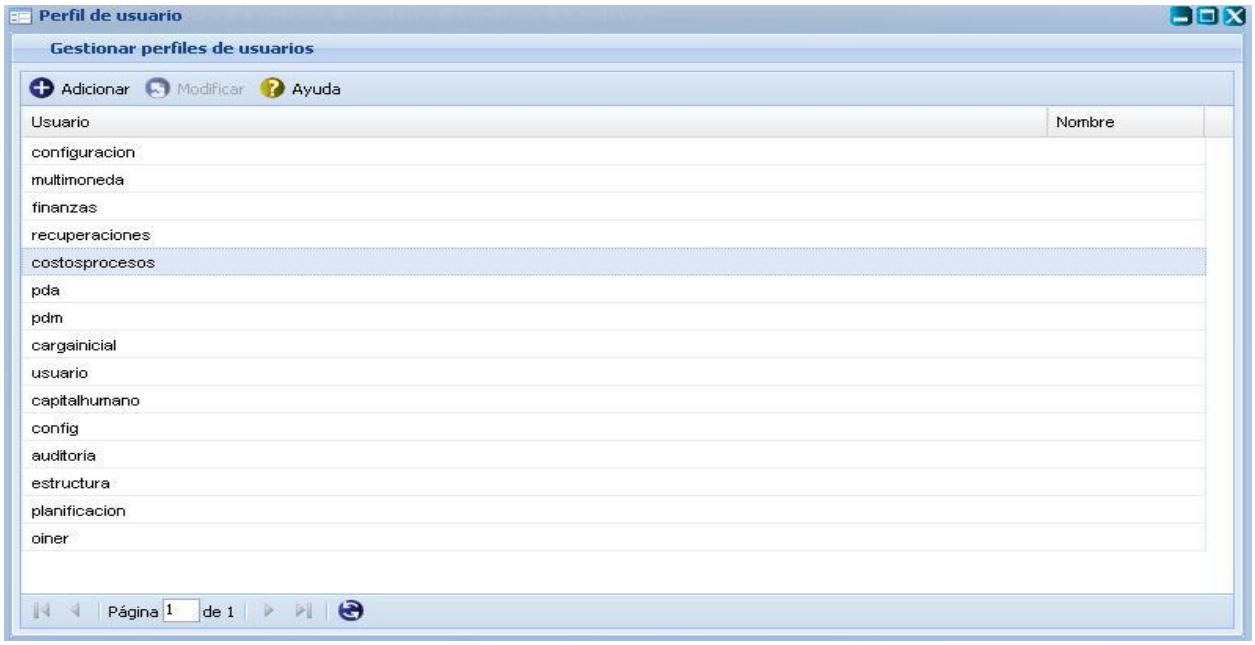

Figura 16 (Prototipo interfaz para requisitos R15.1, R15.2)

2.3.16 Requisito funcional R16 Gestionar Nomenclador de dominios.

El sistema gestiona los dominios compuestos por entidades.

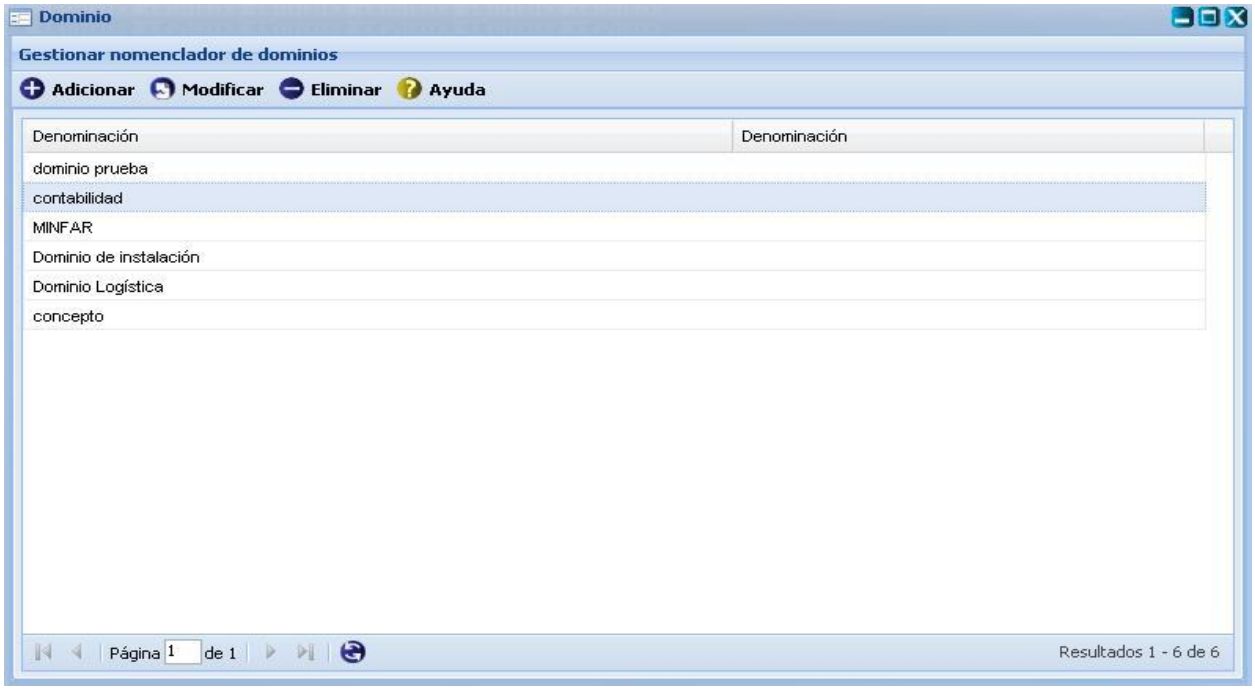

Figura 17 (Prototipo interfaz para requisitos R16.1, R16.2, R16.3, R16.4)

2.3.17 Requisito funcional R17 Gestionar Nomenclador de dominios.

El sistema gestiona las expresiones regulares utilizadas por los sistemas suscritos a él.

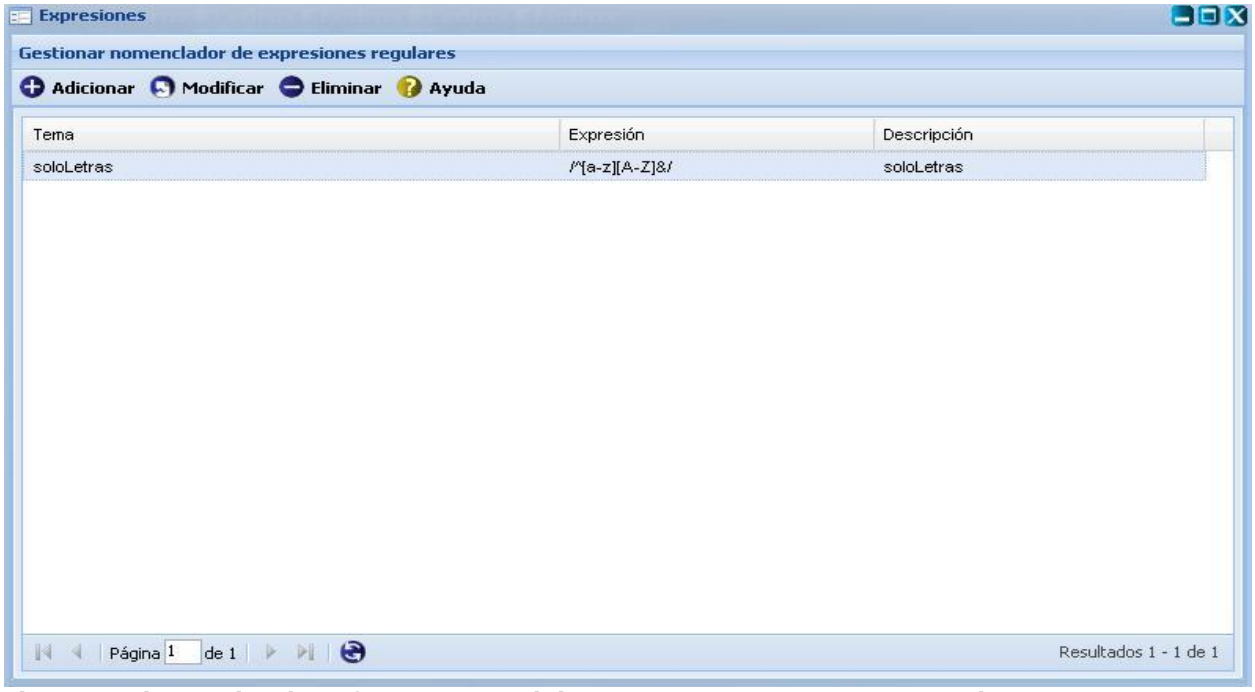

Figura 18 (Prototipo interfaz para requisitos R17.1, R17.2, R17.3, R17.4)

2.3.18 Requisito funcional R18 Gestionar Nomenclador de claves.

El sistema gestiona el tipo de clave que podrá utilizar el usuario.

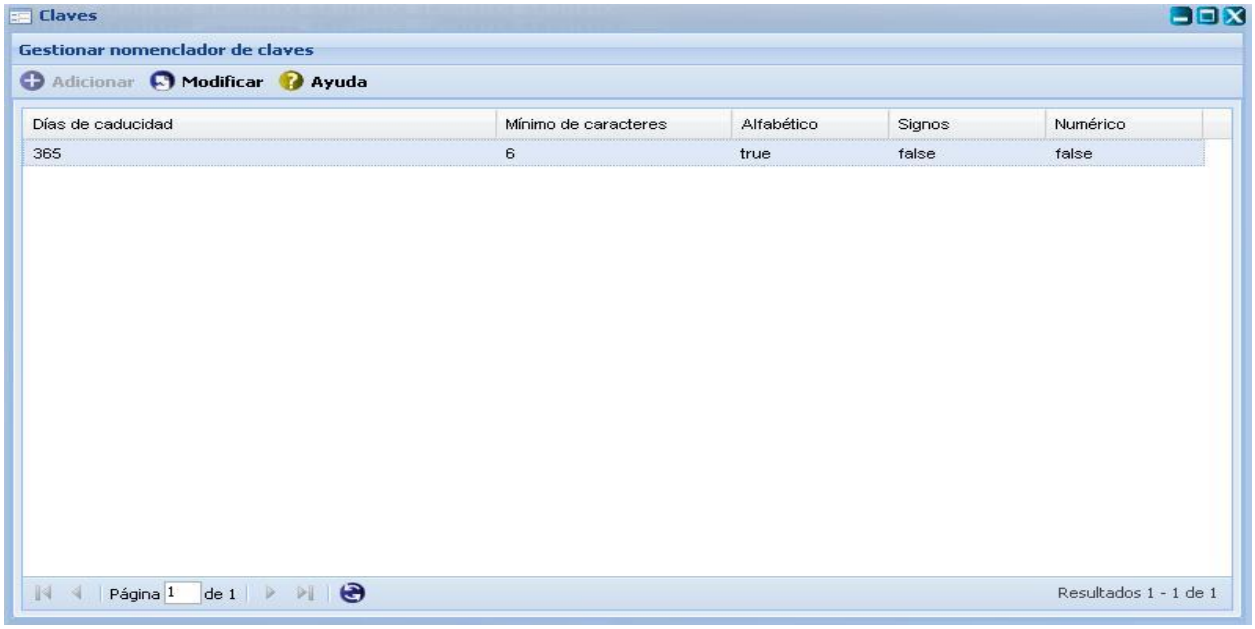

Figura 19 (Prototipo interfaz para requisitos R18.1, R18.2, R18.3)

2.3.19 Requisito funcional R19 Gestionar Nomenclador de gestores de BD.

El sistema gestiona los gestores de BD insertados al sistema que después son asignados a los servidores según las necesidades de los sistemas suscritos a él.

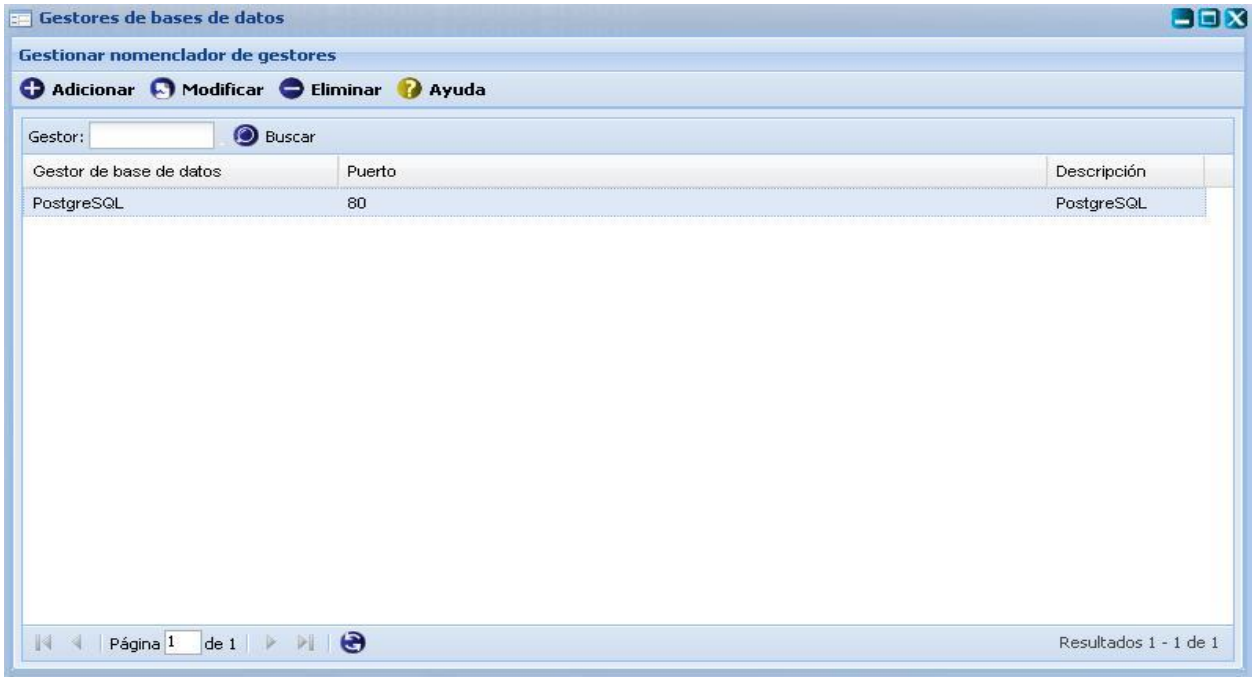

Figura 20 (Prototipo interfaz para requisitos R19.1, R19.2, R19.3, R19.4, R19.5)

2.3.20 Requisito funcional R20 Gestionar Nomenclador de BD.

El sistema gestiona las BD insertadas al sistema que después son asignadas a los gestores de BD según las necesidades de los sistemas suscritos a él.

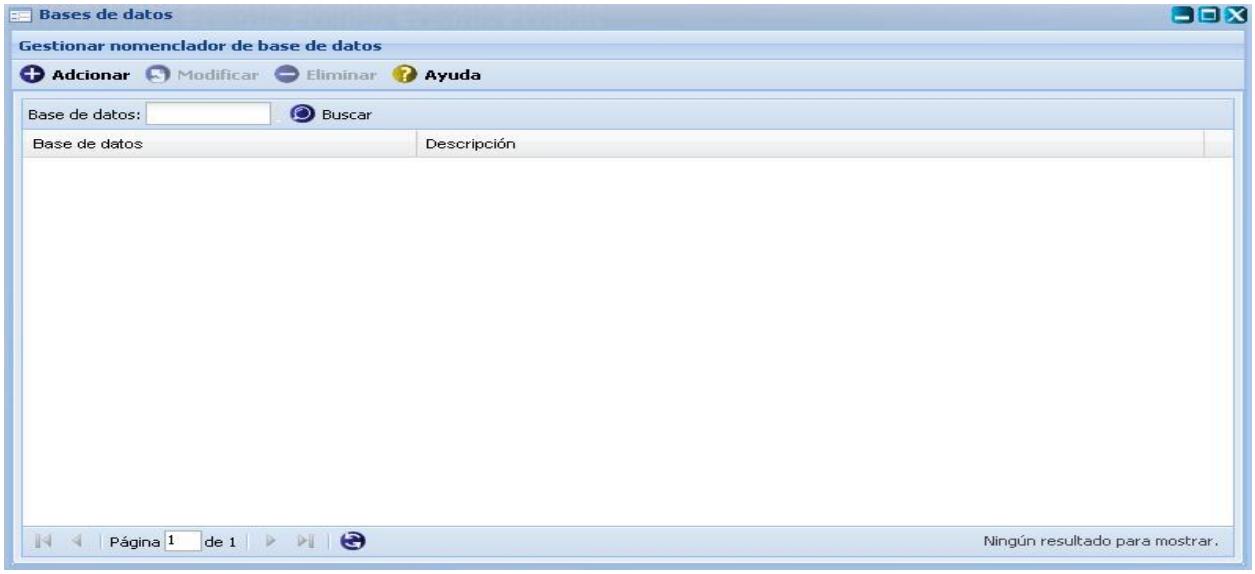

Figura 21 (Prototipo interfaz para requisitos R20.1, R20.2, R20.3, R20.4, R20.5)

2.3.21 Requisito funcional R21 Gestionar esquemas.

El sistema gestiona los esquemas insertados al sistema que después son asignados a las BD según las necesidades de los sistemas suscritos a él.

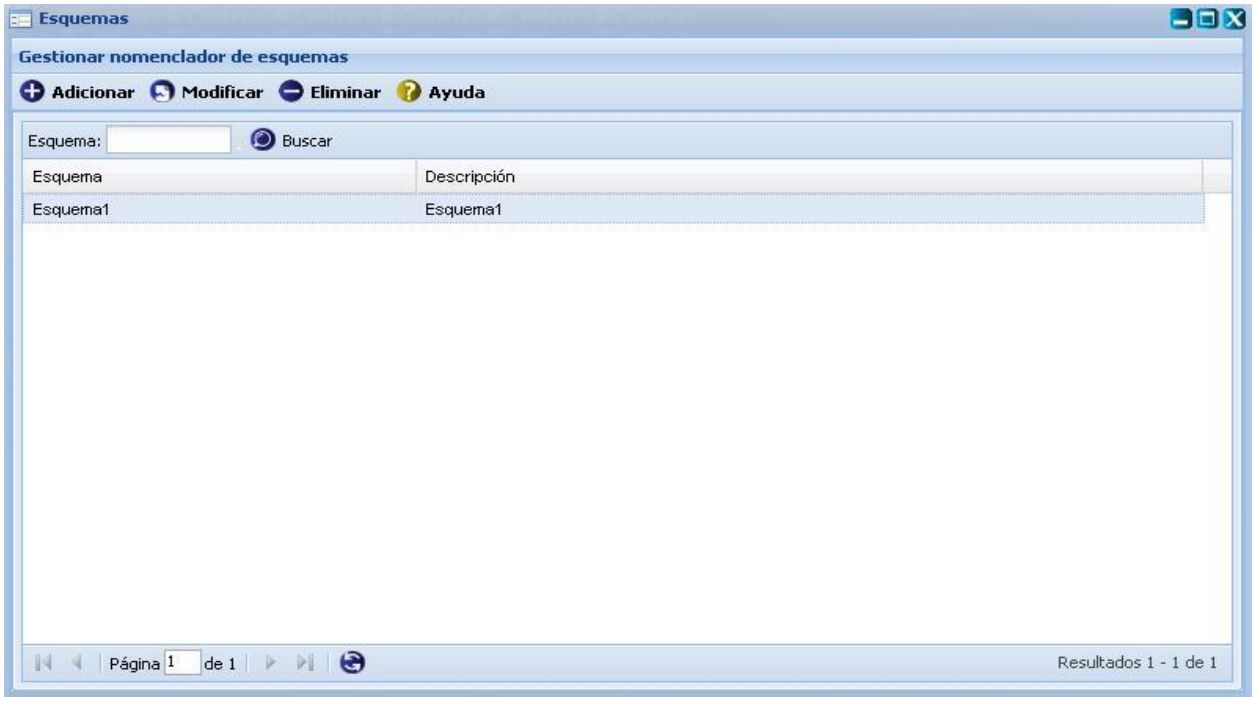

Figura 22 (Prototipo interfaz para requisitos R21.1, R21.2, R21.3, R21.4, R21.5)

2.3.22 Requisito funcional R22 Gestionar idiomas.

El sistema gestiona los idiomas que son utilizados por los sistemas suscritos a él.

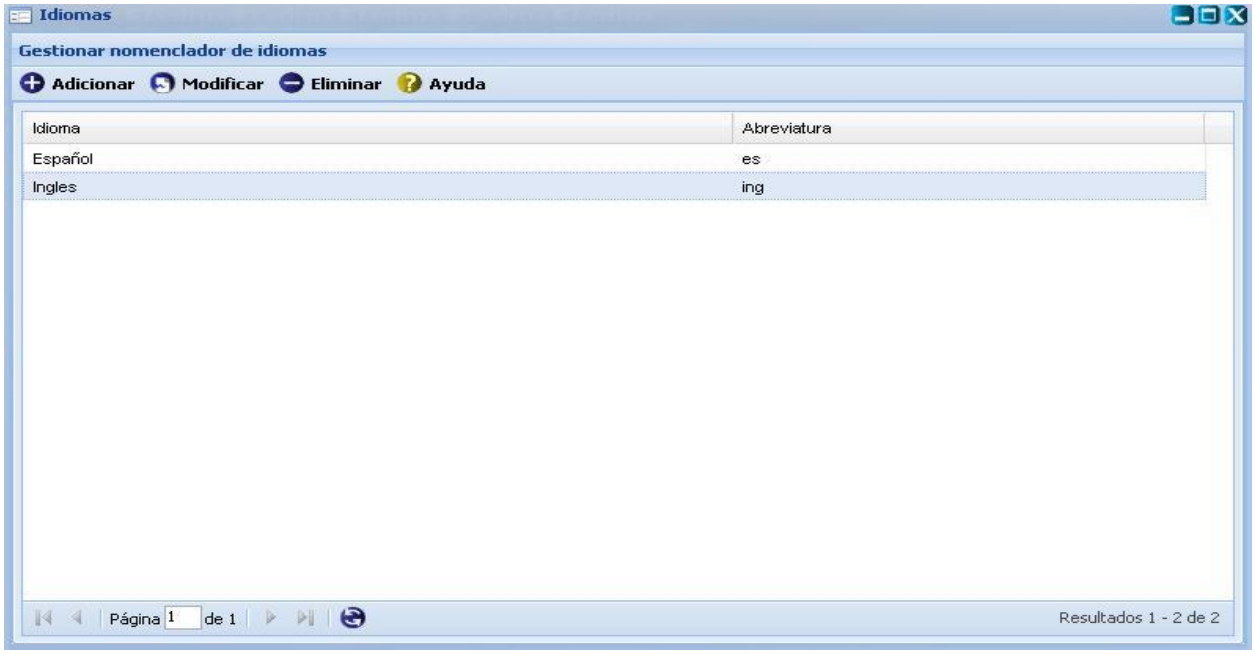

Figura 23 (Prototipo interfaz para requisitos R22.1, R22.2, R22.3, R22.4)

2.3.23 Requisito funcional R23 Gestionar temas.

El sistema gestiona los temas utilizados por los sistemas suscritos a él.

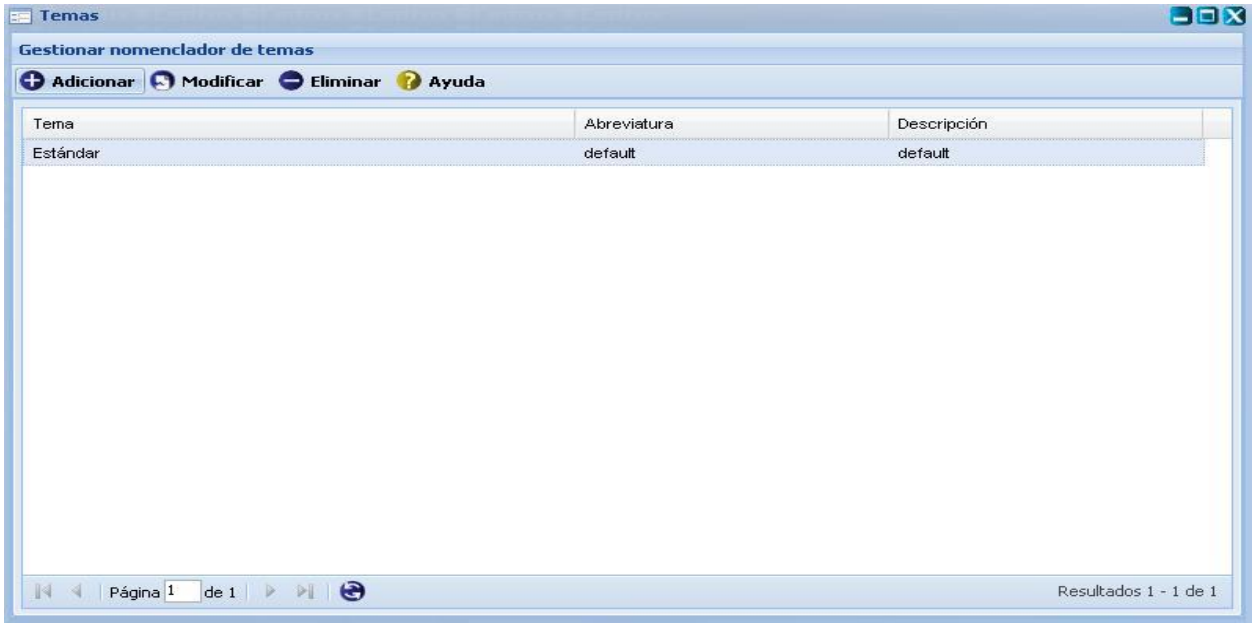

Figura 24 (Prototipo interfaz para requisitos R23.1, R23.2, R23.3, R23.4)

2.3.24 Requisito funcional R24 Gestionar escritorios.

El sistema gestiona los tipos de escritorios de los sistemas suscritos a él.

En este prototipo de interfaz están representados los requisitos R24.1, R24.2, R24.3, R24.4

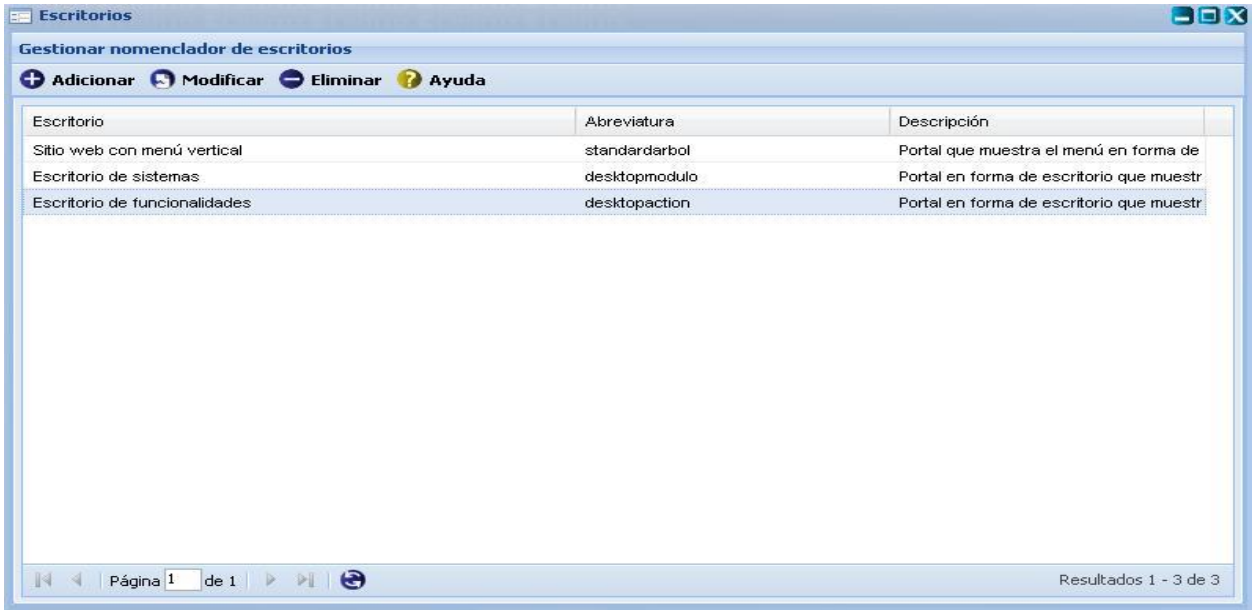

Figura 25 (Prototipo interfaz para requisitos R24.1, R24.2, R24.3, R24.4)

2.4 Diseño de clases.

En este epígrafe presentaremos el diseño de las clases persistentes del sistema, este diagrama se mostrara divido primeramente por las clases correspondientes a cada módulo, y luego se mostrará el diagrama general. El estudio de este diagrama ayuda a un mejor entendimiento de cómo fue diseñada la solución.

2.4.1 Clases pertenecientes al módulo configurar sistema.

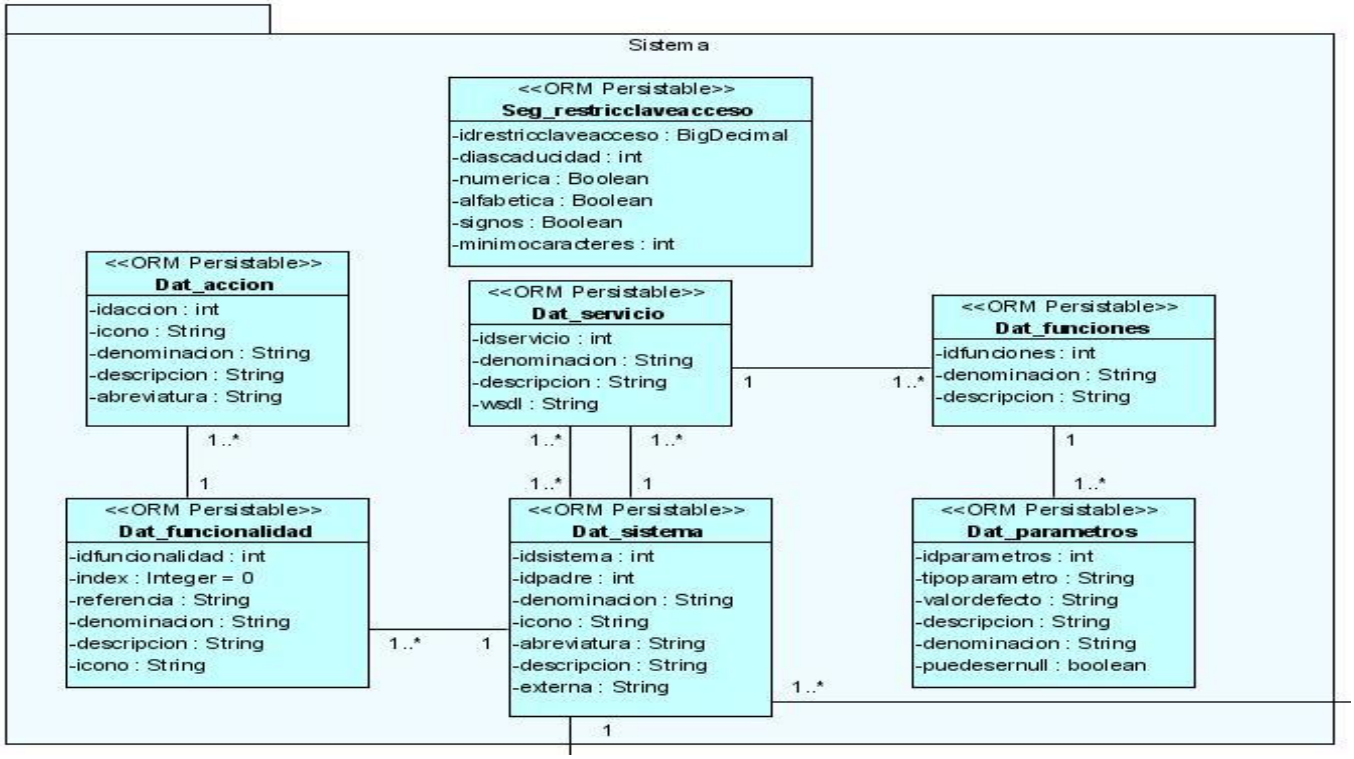

Figura 26 (Diagrama de clases correspondiente al módulo Configurar Sistemas)

2.4.2 Clases pertenecientes al módulo configurar servidores.

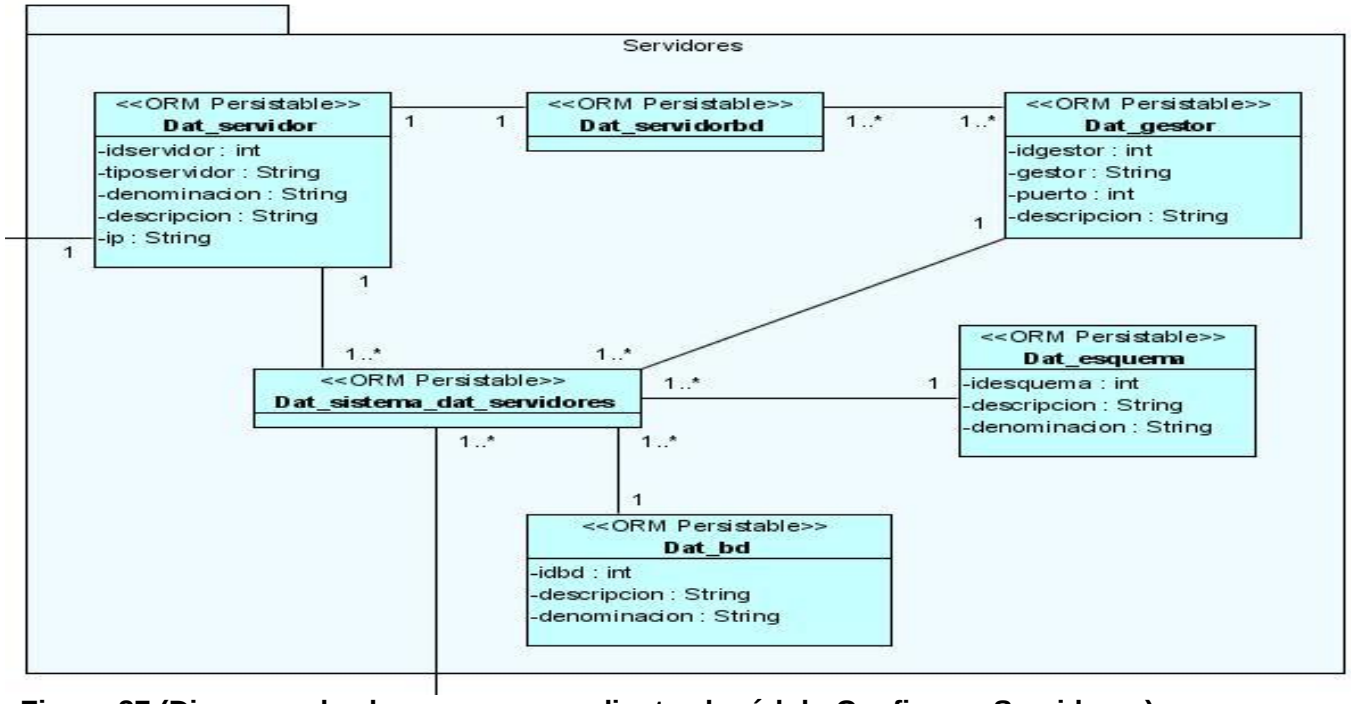

Figura 27 (Diagrama de clases correspondiente al módulo Configurar Servidores)

2.4.3 Clases pertenecientes al módulo configurar usuarios.

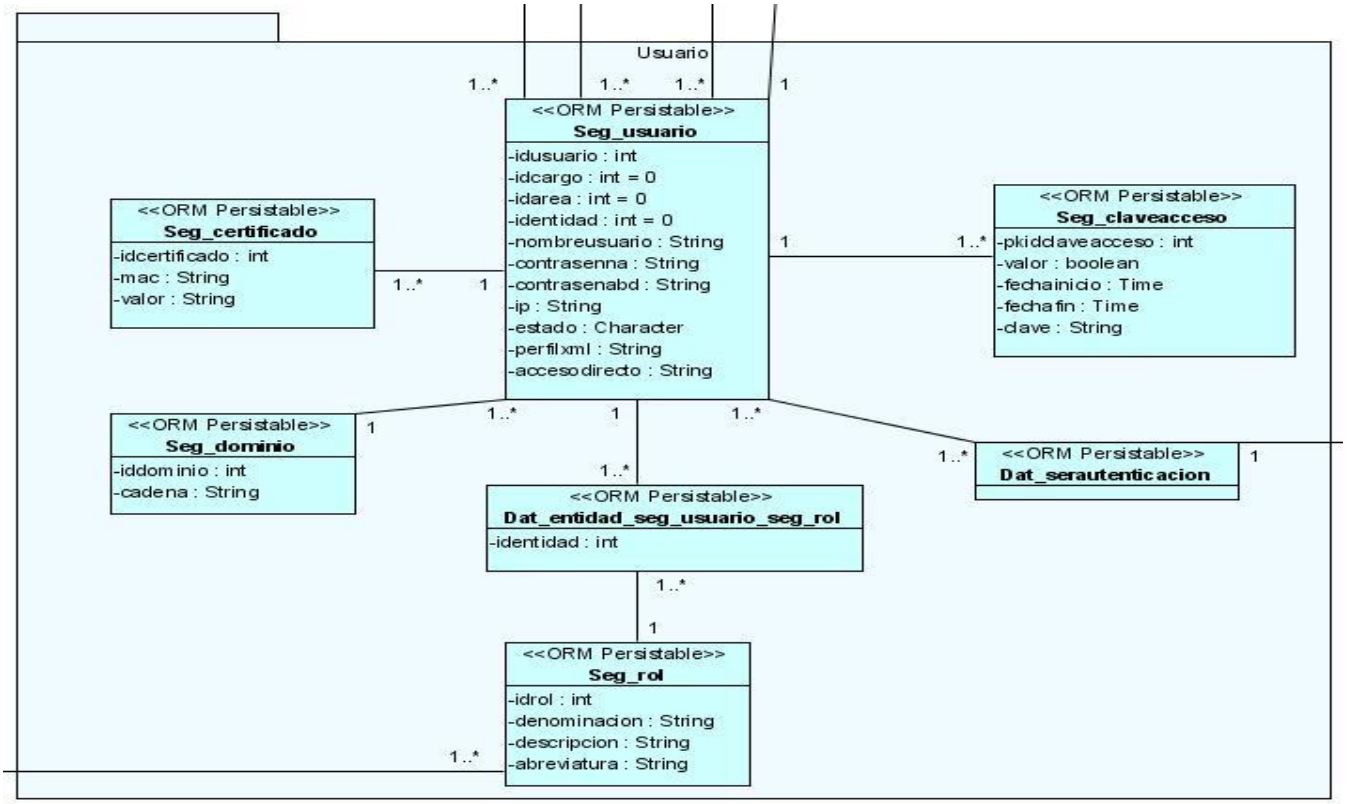

Figura 28 (Diagrama de clases correspondiente al módulo Configurar Usuarios)

2.4.4 Clases pertenecientes al módulo configurar usuarios.

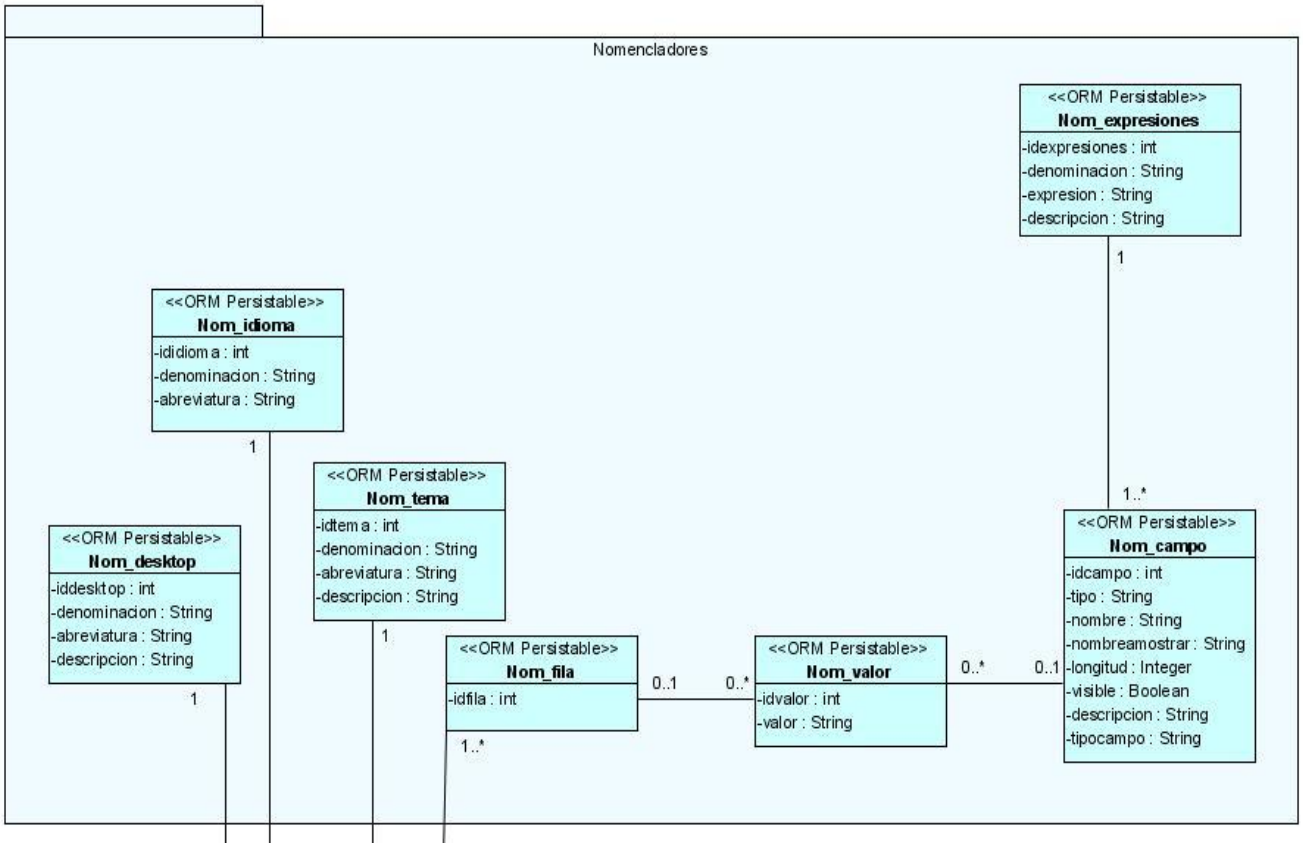

Figura 29 (Diagrama de clases correspondiente al módulo Configurar Nomencladores)

2.4.5 Diagrama de clases general.

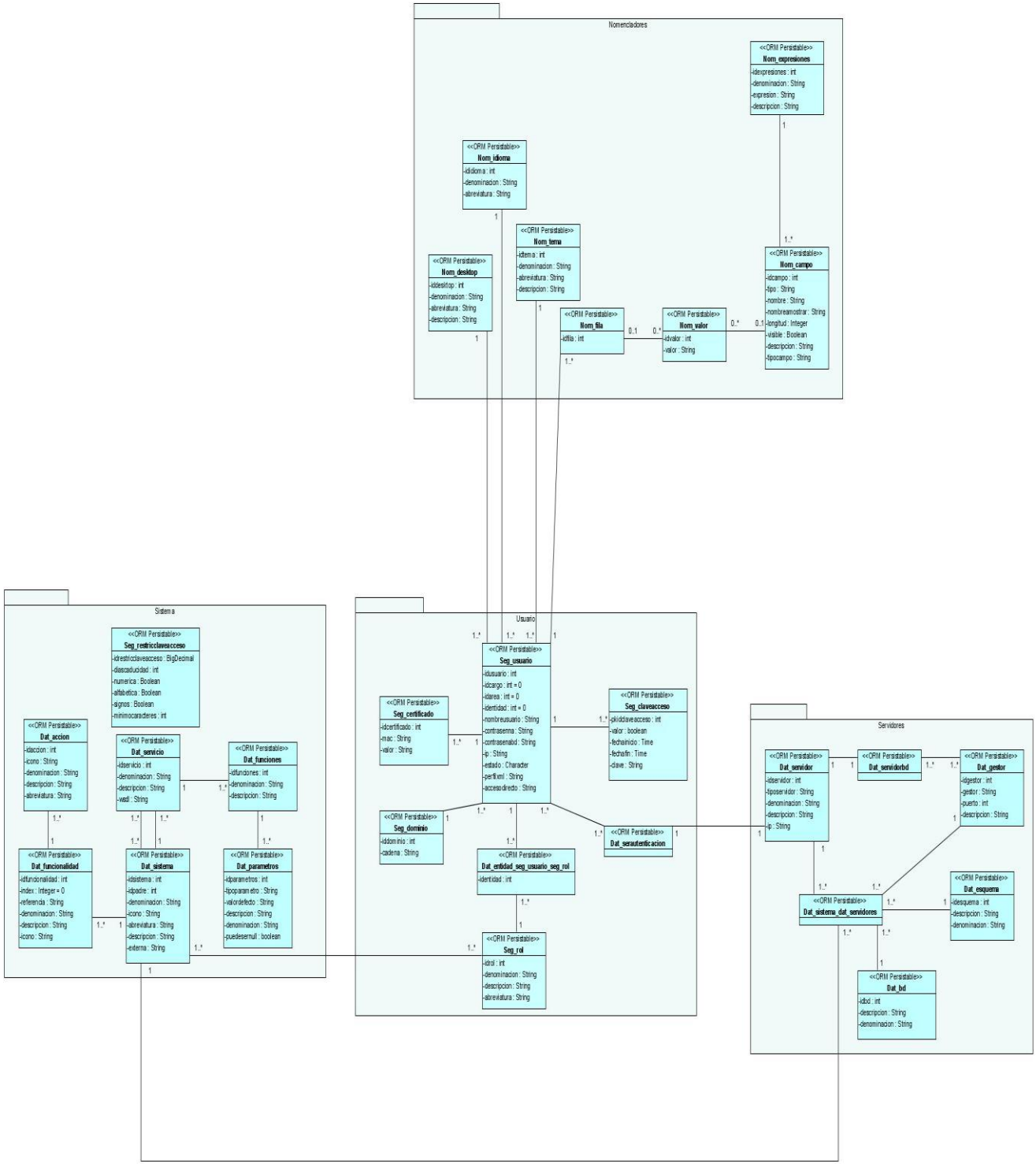

Figura 30 (Diagrama de clases general)

2.5 Artefactos de implementación.

En este epígrafe se verá todo lo referente a los artefactos de implementación utilizados para llevar a cabo el desarrollo de la aplicación. Dentro de estos artefactos se encuentran los estándares de codificación que no son más que pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Constituyen una guía para el desarrollo en la producción, desde el punto de vista arquitectónico, con el propósito de lograr una estandarización del código.

Los estándares de codificación permiten una mejor integración en la producción y se establecen las pautas que conlleven a lograr un código más legible y reutilizable, de tal forma que se pueda aumentar las formas de dar mantenimiento en un futuro.

2.5.1 Estándares de Nomenclatura

2.5.1.1 Nomenclatura de las clases.

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación *PascalCasing*[11]. Con sólo leerlo se reconoce el propósito de la misma.

Ejemplo: GestionarUsuario

Nomenclatura según el tipo de clases.

Clases controladores.

Las clases controladoras después del nombre llevan la palabra:”Controller”.

Ejemplo:

GestionarUsuarioController

[11] **Pascal Casing** es un término muy utilizado en programación, en especial por programadores JAVA y hace referencia a un método de anotación o nomenclatura para definición de variables, procesamiento de strings. Cada variable o string debe comenzar con letra mayúscula.

Clases de los modelos.

Business (Negocio)

Las clases que se encuentran dentro de Business después del nombre llevan la palabra: “Model”.

Ejemplo:

DatAccionModel

Domain (Dominio)

Las clases que se encuentran dentro de Domain el nombre que reciben es el de la tabla en la Base de Datos.

Ejemplo:

DatAccion

Generated (Dominio bases)

Las clases que se encuentran dentro de Generated el nombre comienza con la palabra: “Base” y seguido el nombre de la tabla en la Base de Datos.

Ejemplo:

BaseDatAccion

Clases del framework.

Como parte del marco de desarrollo de Zend existe el Zend_Loader (Cargador) que, amen del cumplimiento de ciertas normas para la nomenclatura de las clases garantiza que, a partir de una ruta de inclusión, este sea el responsable de la inclusión de los recursos requeridos en el proceso.

Ejemplo: Los nombres de las clases contienen la dirección donde se encuentran, de la siguiente forma, si tenemos una clase llamada Condado en la siguiente estructura Cuba/VC/SantaClara/Condado.php entonces un identificador de la misma debe ser

Cuba_VC_SantaClara_Condado lo que garantiza que a través de una casuística particular el cargador localice estos recursos.

Nomenclatura de las funciones.

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing*[12], y con sólo leerlo se reconoce el propósito de la misma.

Ejemplo:

insertarAccion

En caso de ser una acción de la clase controladora se le pone el nombre y seguida la palabra:"Action"

Ejemplo:

insertarAccionAction

Nomenclatura de las variables.

El nombre a emplear para las variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing,* y comenzando con un prefijo según el tipo de datos.

Ejemplo: arrFuncionalidades

Prefijos para los tipos de datos.

Los prefijos a utilizar en la creación de variables serán los siguientes:

Tipos de Datos	Prefijos
Arreglos	arr
Objetos	obj
Enteros	int

[12] CamelCase es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre CamelCase se podría traducir como *Mayúsculas/Minúsculas Camello*, aunque no es correcto en todos los contextos ya que la palabra inglesa *Case* no tiene traducción literal. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello.

Cadena	str
float	flt
Boolean	Boo

Tabla 2 (Prefijos para los tipos de datos)

Nomenclatura de las constantes.

El nombre a emplear para las constantes se escribe con todas las letras en mayúscula.

Ejemplo: USUARIO.

Nomenclatura de los atributos.

El nombre a emplear para los atributos se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing*.* Además en caso de ser un objeto se comienza con:"_" y después se escribe el nombre.

Ejemplo: intMoneda = dinero

objMoneda =_dinero

2.5.2 Normas de comentariado.

Es una necesidad comentar todo lo que se haga dentro del desarrollo, es decir, establecer las pautas que conlleven a lograr un código más legible y reutilizable, de manera que se pueda aumentar su mantenibilidad a lo largo del tiempo.

Nomenclatura de los comentarios.

Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se está desarrollando.

En las clases.

Antes de la declaración de una clase se escribe una breve descripción donde se explique el propósito de la misma. Que se escribe de la siguiente forma:

/**

* Nombre de la clase *

* Descripcion *

* @author *

* @package *(módulo)

* @subpackage *(sub módulo)

* @copyright *

* @version (versión - parche) */

En las funciones.

Antes de la declaración de la función se escribe una breve descripción donde se explique el propósito de la misma. Que se escribe de la siguiente forma:

/**

* Nombre de la función *

* Descripcion *

* @author * (en caso de que no sea el autor de la clase)

* @param *(los parámetros que se le pasan a la función con su descripción)

* @throws *(en caso de que dispare una excepción)

* @return *(se pone lo que devuelve la función y un comentario)/*

En las llaves

Cada vez que se vaya a cerrar una llave se escribe en forma de comentario lo siguiente: Fin y el nombre de lo que se este cerrando.

2.5.3 Estándar de código base de datos.

2.5.3.1 Estándares de nomenclatura.

Nomenclatura de la base de datos.

Los nombres de las Bases de Datos comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación *PascalCasing**. Con sólo leerlo se reconoce el propósito de la misma.

Ejemplo: ContMaterial

Apariencia de los esquemas.

El nombre a emplear para los esquemas se escribe con todas las letras en minúscula, comenzando por el prefijo mod, a continuación el símbolo "_", y por último el nombre del módulo.

Ejemplo: create schema 'mod_seguridad';

Nombre de las tablas.

El nombre empleado, debe escribirse con todas las letras en minúscula para evitar problemas con el *Case Sensitive* del gestor y con solo leerlo se reconoce el propósito de la misma.

Ejemplo: create table 'nom_producto';

Prefijos a utilizar en la creación de tablas.

Los prefijos a utilizar en la creación de tablas serán los siguientes:

- **dat**_ Prefijo utilizado en tablas que almacenan la mayor cantidad de características de una entidad.
- **nom**_ Prefijo utilizado en tablas nomencladoras.
- **seg**_ Prefijo utilizado en tablas que almacenan control de acceso, usuarios y opciones de acceso de uno o varios sistemas. (Tablas de Seguridad)
- **conf**_ Prefijo utilizado en tablas que almacenan parámetros de configuración del sistema**.** (Tablas de Configuración)
- **tmp**_ Prefijo utilizado para tablas que almacenan datos transitorios. (Tablas Temporales)
- **his**_ Prefijo utilizado para tablas que almacenan datos por largos períodos de tiempo y que solo son utilizados para análisis esporádicos. (Tablas Históricos)
- **res**_ Prefijo utilizado para las tablas resúmenes, empleadas en los reportes.

Ejemplo:

nom_producto (Nomeclador).

seg_usuarios (de seguridad).

conf_almacen (de configuración).

Apariencia de los campos.

El nombre a emplear para los campos se escribe con todas las letras en minúscula, con solo leerlo se reconoce el propósito del mismo y debe incluir un comentario con su descripción. Si el campo es un identificador debe empezar con id.

Ejemplo:

add field ‘idusuario’;

cantemb: cantidad de embalajes.

Nombre de las llaves primarias.

El nombre de las restricciones se escribe con minúscula. Comienza con el identificador id seguido el nombre de la tabla todo junto y en minúscula.

Ejemplo: idusuario (Llave primaria de la tabla dat_usuario).

Nombre de las llaves foráneas.

El nombre de las llaves foráneas se escribe con minúscula y el nombre de la llave primaria de la tabla donde pertenece.

Ejemplo: idusuario. (Llave foránea de la tabla “dat_usuario”)

Nombres de las funciones, triggers, tipos de datos y vistas.

El nombre empleado permite con sólo leerlo reconocer el propósito del mismo. Se utilizan los prefijos siguientes para la denominación de funciones, triggers, tipos de datos y vistas.

Estos son:

- **f** Funciones (Procedimientos Almacenados.)
- **ft** Funciones de triggers
- **t** Triggers
- **td** Tipo de datos

- **V** Vistas

Nombre de los tablespace.

El nombre se definirá todo en minúscula, empezando con la letra tbs, seguido un guión bajo y el nombre del módulo junto en minúscula.

Ejemplo:

Tbs_modfinanza (tablespace del módulo de finanza)

Tbs_modestructura (tablespace del módulo de estructura)

Nombre de los dominios.

El nombre se definirá todo en minúscula, empezando con la letra d, posteriormente el nombre del módulo al que pertenece y por último el nombre del dominio, separados por punto.

Ejemplo:

d_modfinanza (dominio que pertenece al módulo finanza)

d_modestructura (dominio que pertenece al módulo estructura)

Nombre de las secuencias.

El nombre de una secuencia se definirá todo en minúscula empezando con la letra sec posteriormente un guión bajo y nombre de la tabla donde pertenece.

Ejemplo: sec_datpersona_seq

2.5.4 Diagrama de componentes del sistema de seguridad SIGIS

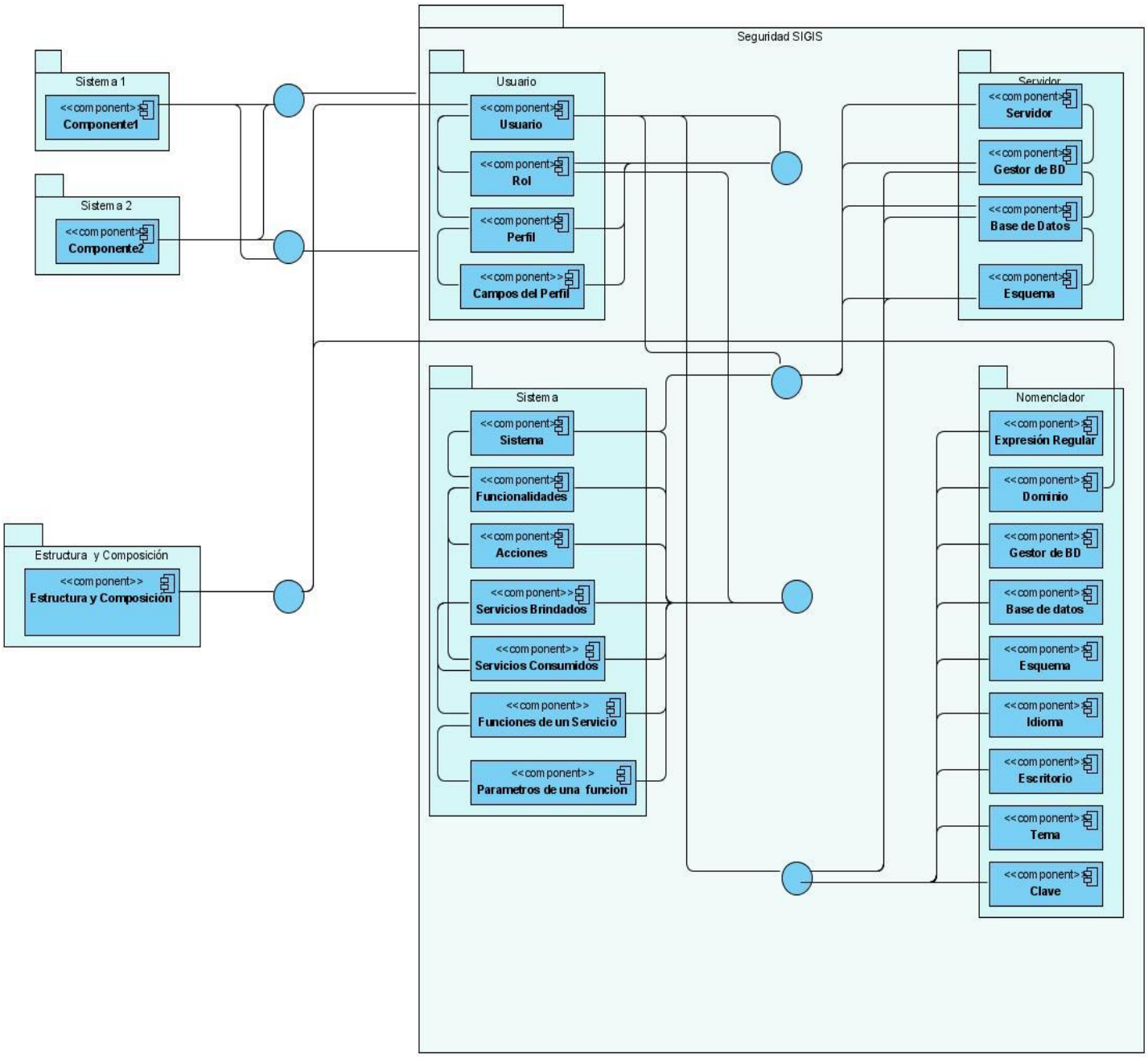

Figura 31 (Diagrama de componentes)

2.6 Casos de prueba.

La realización de los casos de pruebas tiene como objetivo demostrar al cliente la reacción que corresponderá por parte del sistema luego de realizar alguna acción en el mismo. Para un mejor entendimiento de las respuestas o posibles funcionalidades que brindará el sistema, según la necesidad del usuario. La descripción de los casos de prueba del sistema se realizó por escenarios. A continuación se muestra una descripción de los más importantes que se corresponden con: R1 requisito funcional gestionar sistemas y R13 requisito funcional gestionar usuarios, pertenecientes a los módulos Configurar Sistemas y Usuarios respectivamente. En este epígrafe solo se mostrará del R1 los casos de prueba para el R1.2 Registrar Sistemas, los demás R1.3, R1.4, R1.5, R1.6 junto con los correspondientes al R13 se podrán encontrar en el (**¡Error! No se encuentra el origen de la referencia.** de Gestión Integral de Seguridad).

Caso de prueba para R1.2 Registrar Sistemas.

Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el subsistema Seguridad, en el módulo Configurar sistemas, en la interfaz Sistemas.
- El sistema que se desea adicionar no ha sido adicionado antes al sistema.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Registrar sistema	Se adiciona un nuevo sistema.	EP 1.1: Registrar sistema	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar.** – Se insertan todos los datos. – Se presiona el botón **Aceptar**.
		EP 1.2: Registrar sistema dejando campos requeridos en blanco.	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar**. – Se introducen los datos dejando campos requeridos en blanco. – Se presiona el botón **Aceptar**.
		EP 1.3: Registrar sistema introduciendo error en los datos.	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar**. – Se introducen los datos del sistema que se desea adicionar en el formulario introduciendo errores en los datos. – Se presiona el botón **Aceptar**.

		EP 1.4: Aplicar	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar**. – Se introducen los datos del sistema que se desea adicionar en el formulario. – Se presiona el botón **Aplicar**. – Se presiona el botón **Aceptar**.
		EP 1.5: Cancelar	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar**. – Se introducen o no los datos en el formulario. – Se presiona el botón **Cancelar**.

Tabla 3 (Requisitos a probar para caso de prueba del R1.2)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación	Campo de texto	No	Combinación de letras.
2	Abreviatura	Campo de texto	No	Combinación de letras.
3	Icono	Campo de texto	Si	Combinación de letras.
4	Servidores	Treepanel	No	Lista desplegable de los servidores que existen.
5	Descripción	Campo de texto	Si	Combinación de letras, números y caracteres especiales.
6	Externo	Check box	Si	Cuadro de selección

7	Servidor web	Campo de texto	Si	Combinación de letras.

Tabla 4 (Descripción de variables para caso de prueba del R1.2)

Juegos de datos a probar.

Id del escena-rio	Escenario	Denominación	Abreviatura	Icono	Servidores	Descripción	Externo	Servidor web	Respuesta del sistema	Resultado de la prueba
EP1.1	Registrar sistema.	V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se adiciona el sistema y se guarda la información adicionada en el módulo de configurar sistemas.	
EP1.2	Registrar sistema dejando campos requeridos en blanco.	I(vacío)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se muestra el campo de texto en rojo que indica que no se pueden dejar campos en blanco y se mantiene en la ventana	
		V(sist seguridad)	I(vacío)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	I(vacío)	V(seguridad)	V(publict)	V(Ldap)		

		I(Vacio)	I(Vacio)	I(Vacio)	I(Vacio)	I(Vacio)	I(Vacio)	I(vacio)		
EP 1.3	Registrar sistema introduciendo error en los datos.	I(-*/)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se muestra el campo de texto en rojo que indica que no se pueden entrar caracteres inválidos y se mantiene en la ventana Adicionar sistema.	
		V(sist seguridad)	I(-*/)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	I(-*/)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	I(seg34)	V(120)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	I(/*-)		
EP1.4	Aplicar	V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se adiciona el sistema y se guarda la información adicionada en el módulo de configurar sistemas.	

		I(vacío)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se muestra el campo de texto en rojo que indica que no se pueden dejar campos en blanco y se mantiene en la ventana Adicionar sistema.	
		V(sist seguridad)	I(vacío)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	I(vacío)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	I(vacío)	V(publict)	V(Ldap)		
		I(-*/)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se muestra el campo de texto en rojo que indica que no se pueden entrar caracteres inválidos y se mantiene en	
		V(sist seguridad)	I(-*/)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	I(-*/)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		

		V(sist seguridad)	V(seg)	I(vacío)	V(120)	V(seguridad)	V(publict)	V(Ldap)	la ventana Adicionar sistema.	
		V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	I(/*-)		
EP1.5	Cancelar	NA	NA	NA	NA	NA	NA		Se cancela la operación.	

Tabla 5 (Juego de datos a probar para caso de prueba del R1.2)

2.7 Modelo de datos.

En este epígrafe se muestra el modelo de datos. Este estará divido en 4 partes que se corresponden con cada módulo del sistema, *Configurar Sistemas*, *Configurar Servidores, Configurar Usuarios, Configurar Nomencladores.* Con este diagrama podemos comprender como está diseñada la Base de Datos.

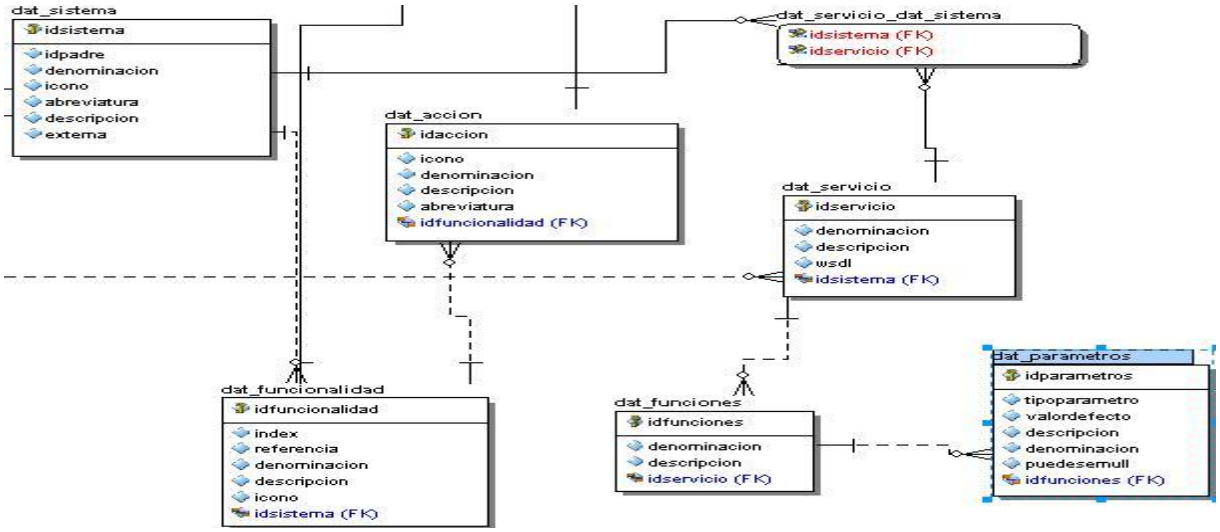

Figura 32 (Modelo de datos para el módulo Configurar Sistemas)

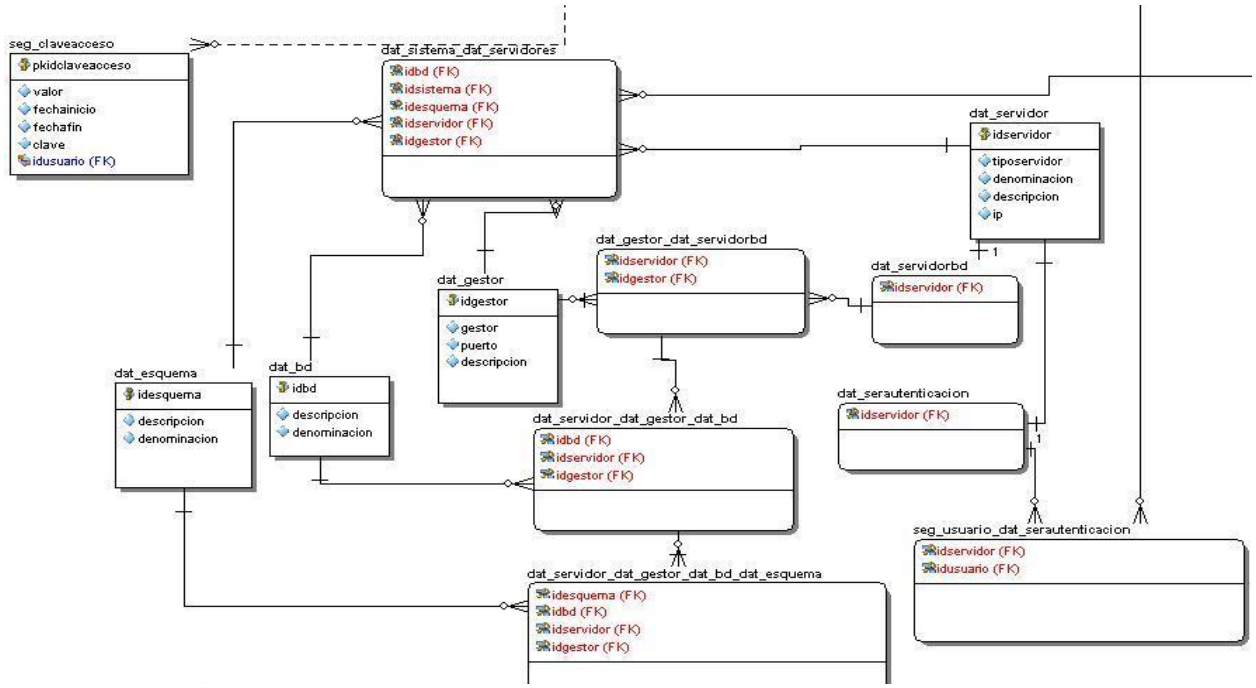

Figura 33 (Modelo de datos para el módulo Configurar Servidores)

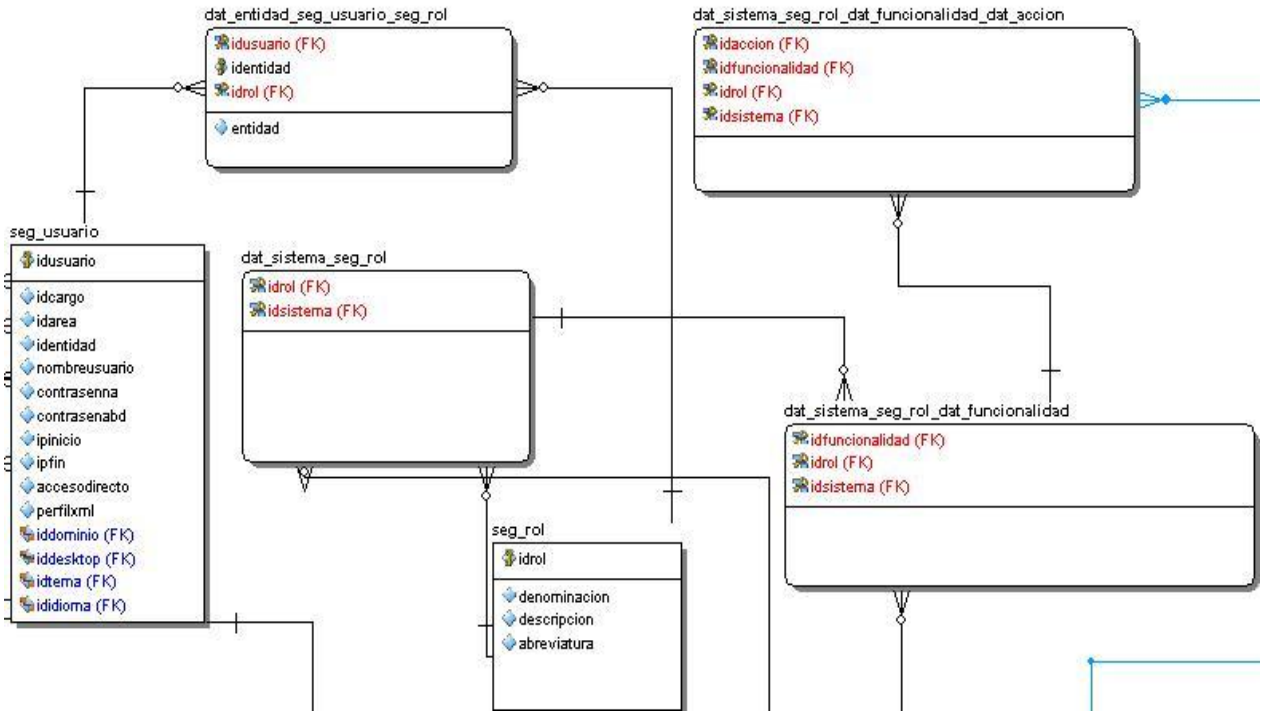

Figura 34 (Modelo de datos para el módulo Configurar Usuarios)

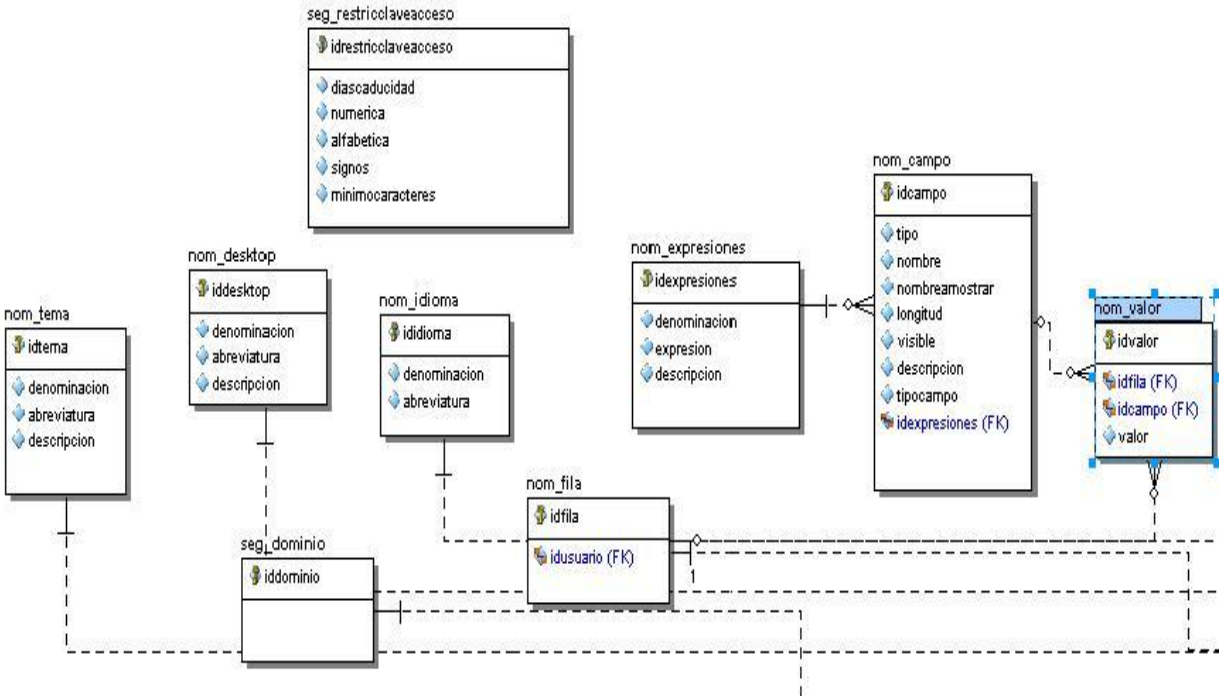

Figura 35 (Modelo de datos para el módulo Configurar Nomencladores)

2.8 Conclusiones

Durante el transcurso del capítulo se argumentaron los fundamentales aspectos que se llevan a cabo durante el proceso de análisis, comenzando por el levantamiento de los requisitos funcionales que contienen cada unos de los subsistemas integrados en el componente principal y de los requisitos no funcionales para garantizar una ejecución eficaz de la aplicación. También se reflejaron los distintos prototipos de interfaz que se relacionan con los requisitos levantados, se plasmó el diseño de clases para una mejor comprensión de cómo están estructuradas las clases que conforman toda la capa del negocio de los componentes. Se expusieron además los distintos artefactos de implementación generados: Estándar de codificación utilizado tanto para la base de datos como para el desarrollo, diagrama de componentes y el modelo de datos, y se describieron los casos de pruebas de todos los componentes para comprobar si las funcionalidades implementadas en cada uno cumple con los requisitos argumentados en el proceso de desarrollo.

3

Capítulo

Evaluación de la solución.

CAPÍTULO 3: Evaluación de la solución.

3.1 Introducción.

En el presente capítulo se muestran algunas métricas que se aplican en la actualidad para validar la calidad en el diseño de software y se definen cuales se aplicaron al diseño de cada uno de los componentes que integran la solución propuesta. Las mismas ayudan a comprender todo el proceso técnico que se utiliza para desarrollar un producto, así como el propio producto. También brinda una valoración de la solución del componente y la matriz de inferencia de indicadores de calidad.

3.2 Valoración de la solución.

Confirmación por inspección y provisión de evidencia objetiva de que los requerimientos particulares para un uso específico son alcanzados. En diseño y desarrollo, la validación está relacionada con el proceso de reexaminación de un producto para determinar la conformidad con las necesidades del usuario. La validación es realizada normalmente sobre el producto final bajo condiciones operacionales definidas. La valoración de la solución es el paso final en el proceso de evaluación del software. Donde una solución puede ser valorada de diversas maneras, debido a que existen distintas tipos de valoraciones como son:

- Valoración cualitativa.
- Valoración indirecta.
- Valoración directa.

Una valoración se realiza mediante una métrica para asignar uno de los valores de una escala (el mismo puede ser número o categoría) al atributo de la entidad (sistema, subsistemas o componentes).

Valoración cualitativa: Es una evaluación sistemática del grado o capacidad de una entidad para satisfacer necesidades o requerimientos específicos. Además se emplean categorías,

como algunos de los atributos más importantes de una entidad, ejemplo: el lenguaje de desarrollo del programa (C, C ++, C #, PHP, JAVA).

Valoración indirecta: Es la valoración de un atributo derivada del valor de uno o más atributos diferentes. Es la valoración externa de un atributo de un sistema, ejemplo: el tiempo de respuesta a la información alimentada por el usuario, es una valoración indirecta de los atributos del software, debido a que esta medida se verá influenciada por los atributos externos del sistema, así como los propios internos.

Valoración directa: Es una valoración del producto, de forma indirecta o directa. Ejemplo: El número de líneas de código, las valoraciones de la complejidad, el número de fallas encontradas durante el proceso y el índice de señales o alertas, son todas las valoraciones internas propias del producto en sí.

3.2.1 Valoraciones aplicadas a la solución propuesta.

Dentro de los distintos tipos de validación del software existentes se escogieron para la solución propuesta las de tipo cualitativa y directa.

Debido a los problemas actuales que existen con las licencias de software el sistema se realizó sobre plataforma libre implementándose en el lenguaje PHP utilizando el Zend Framework por decisión del grupo de arquitectura del Proyecto Cedrux donde fue desarrollado el sistema. Se empleó como gestor de base de datos la herramienta PostgreSQL debido a que es una herramienta con características importantes como: presenta interfaz amigable y ágil de manipular.Para el acceso a datos se utilizó el framework de acceso a datos Doctrine. Se utilizó el framework Ext-JS para el desarrollo de la interfaz visual. Todas estas herramientas fueron una decisión del grupo de arquitectura del proyecto Cedrux.

El sistema contiene 20601 líneas de código, es totalmente seguro y eficaz, presenta una interfaz amigable para el usuario y fácil de manipular, así como una conexión segura con la base de datos y con los ficheros a gestionar, al igual que con el estandarizado de los mismos.

3.3 Métricas de software.

Un aspecto importante a tener en cuenta en la fase de evaluación de la calidad del diseño ha sido la creación de métricas básicas inspiradas en el estudio de la calidad del diseño orientado

a objeto referenciadas por Pressman en; teniendo en cuenta que este estudio brinda un esquema sencillo de implementar y que a la vez cubre los principales atributos de calidad de software. Siendo esto la principal razón de la concepción de las métricas inspiradas en lo propuesto por Pressman.

Atributos de calidad que se abarcan:

- ***Responsabilidad.*** Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- ***Complejidad de implementación.*** Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
- ***Reutilización.*** Consiste en el grado de reutilización de presente en una clase o estructura de clase, dentro de un diseño de software.
- ***Acoplamiento.*** Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, esta muy ligada a la característica de Reutilización.
- ***Complejidad del mantenimiento.*** Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.
- ***Cantidad de pruebas.*** Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, modulo, clase, conjunto de clases, etc.) diseñado.

Las métricas concebidas como instrumento para evaluar la calidad del diseño del Sistema de Gestión Integral de Seguridad y su relación con los atributos de calidad definidos en este trabajo son las siguientes:

Tamaño operacional de clase (TOC): Está dado por el número de métodos asignados a una clase.

Atributo de calidad	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Tabla 6 (Tamaño operacional de clase (TOC))

Relaciones entre clases (RC): Esta dado por el número de relaciones de uso de una clase con otra.

Atributo de calidad	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 7 (Relaciones entre clases (RC))

3.3.1 Resultados del instrumento de evaluación de la métrica Tamaño operacional de clase (TOC).

Ver instrumentos y tabla de resultados en (Anexo 3: Instrumento de medición de la métrica Tamaño operacional de clase (TOC)).

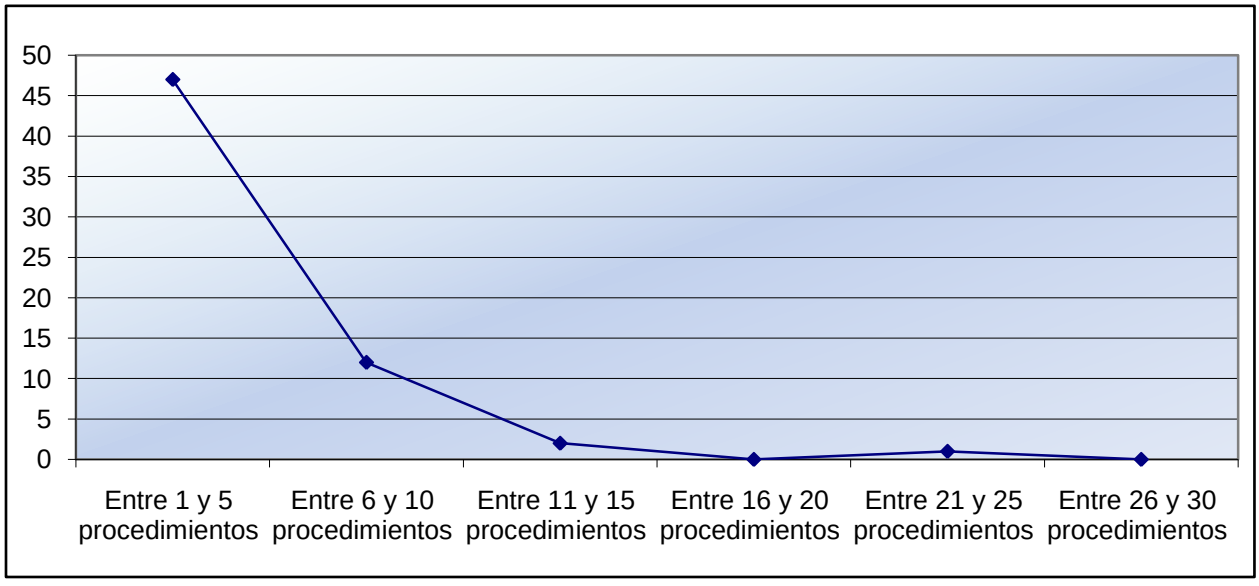

Figura 36 (Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos)

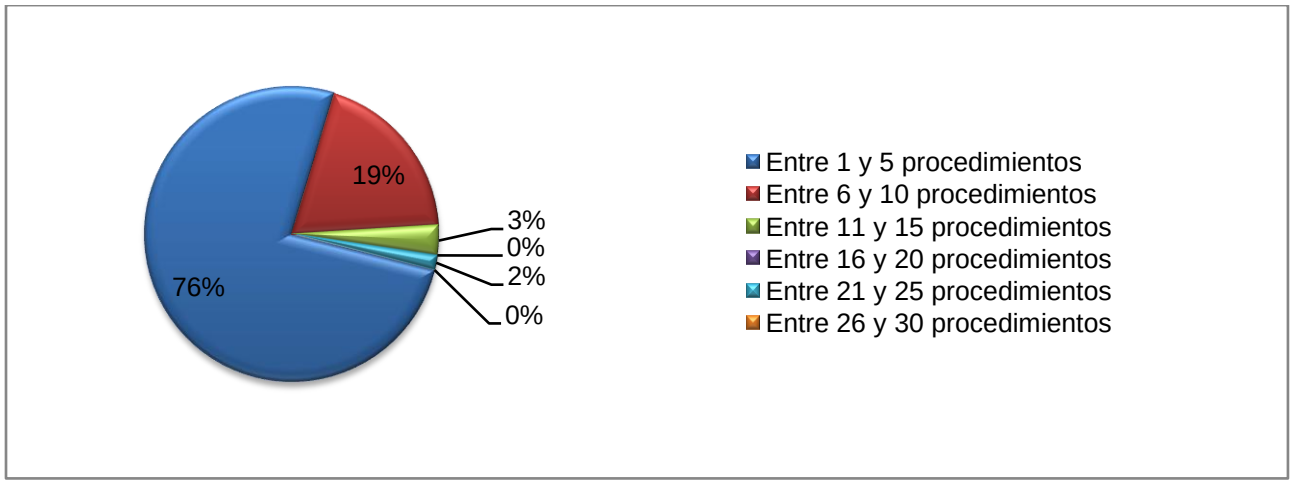

Figura 37 (Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos)

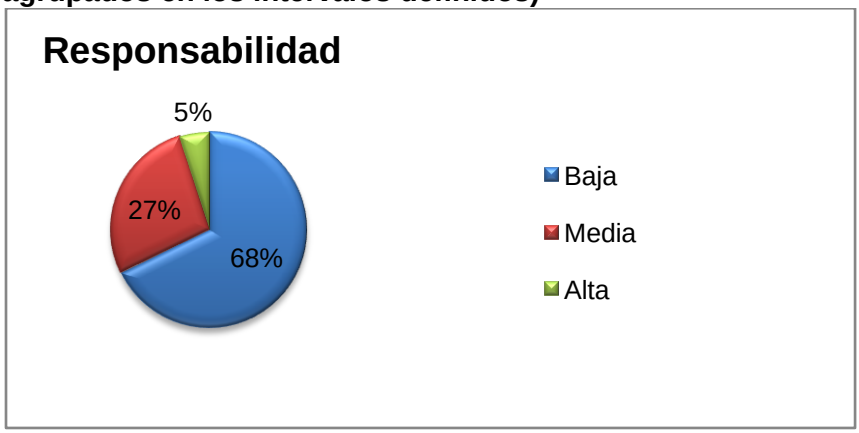

Figura 38 (Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo responsabilidad)

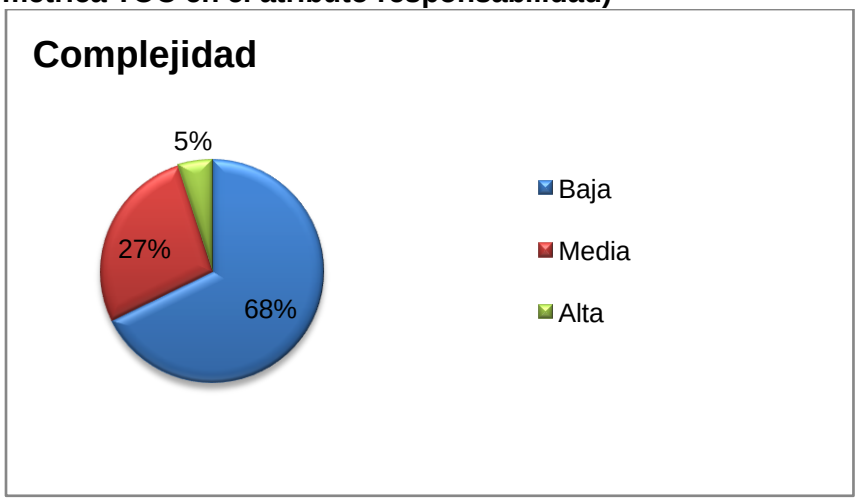

Figura 39 (Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación)

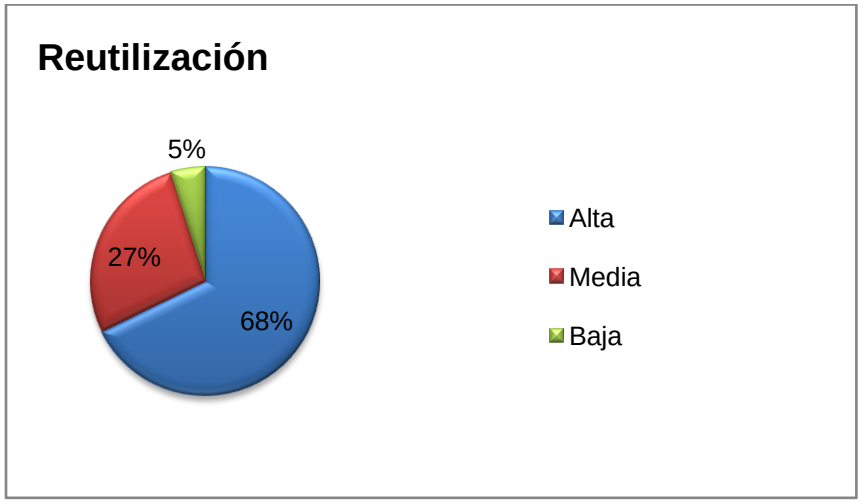

Figura 40 (Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización)

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica TOC, se puede concluir que el diseño del Sistema de Gestión Integral de Seguridad tiene una calidad aceptable teniendo en cuenta que el 95 % de las clases incluidas en este sistema posee menos cantidad de operaciones que la mitad del valor máximo registrado en las mediciones. Además el 95% de las clases poseen evaluaciones positivas en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

3.3.2 Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC).

Ver instrumentos y tabla de resultados en (Anexo 4: Instrumento de medición de la métrica Relaciones entre clases (RC)).

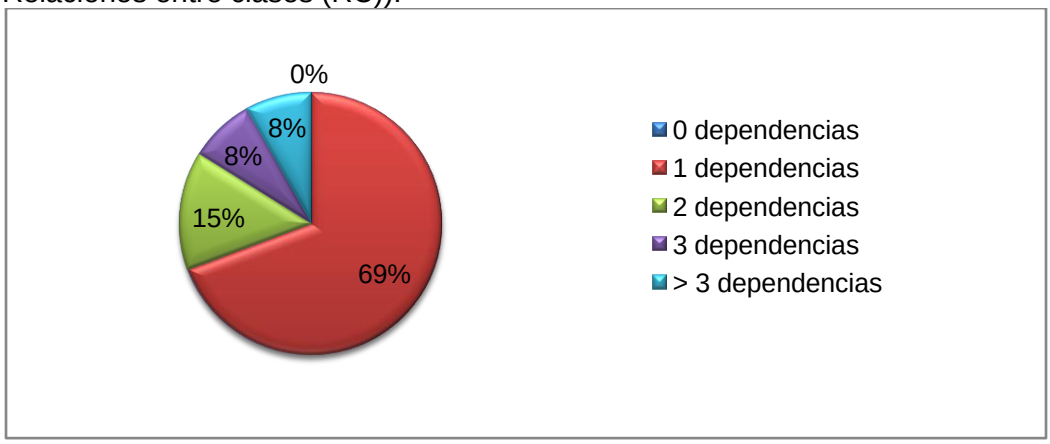

Figura 41 (Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos)

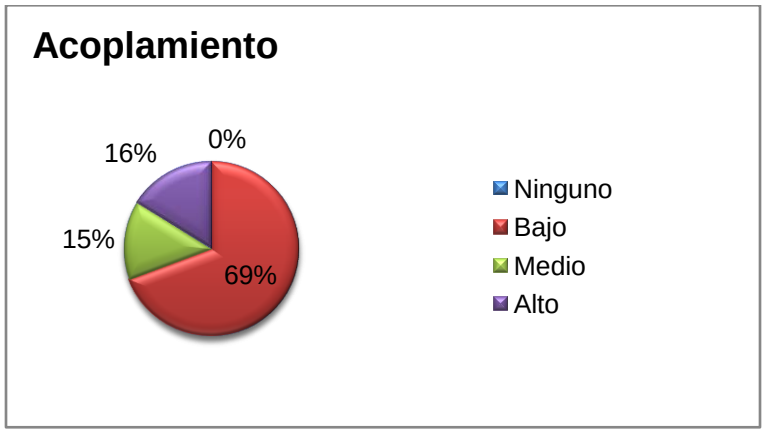

Figura 42 (Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento)

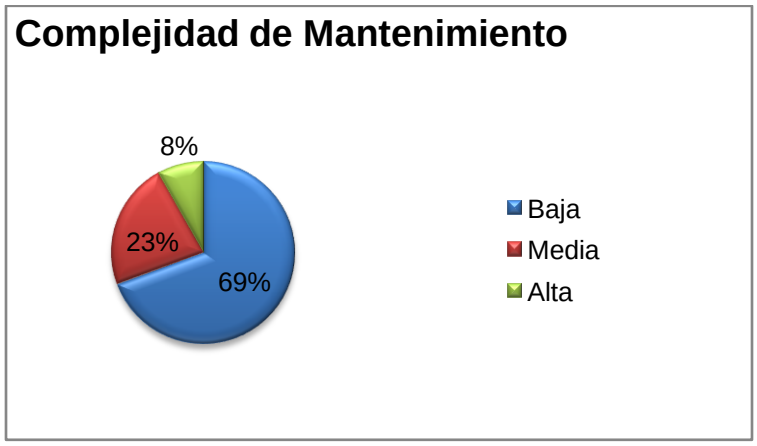

Figura 43 (Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento)

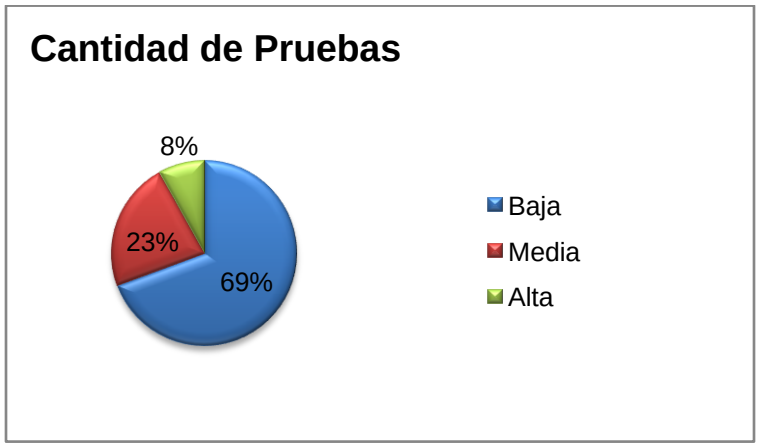

Figura 44 (Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas)

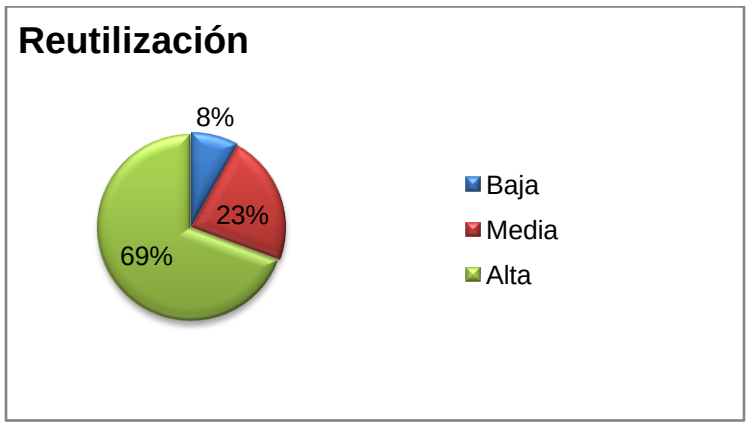

Figura 45 (Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización)

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica RC, se puede concluir que el diseño del Sistema de Gestión Integral de Seguridad tiene una calidad aceptable teniendo en cuenta que el 90 % de las clases incluidas en estos subsistemas posee menos de 3 dependencias de otras clases. Además el 95% de las clases posee índices aceptables en cuanto a Acoplamiento. Así mismo los atributos de calidad Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización se comportan satisfactoriamente en un 96 % de las clases.

3.4 Matriz de cubrimiento o matriz de inferencia de indicadores de calidad.

La matriz de cubrimiento o matriz inferencia de indicadores de calidad es una representación estructurada de los atributos de calidad y métricas utilizadas en el capitulo anterior para evaluar la calidad del diseño de los componentes que integran la solución propuesta. La misma permite conocer si el resultado obtenido de la relación atributo/métricas para cada componente es positivo o negativo. Llevando estos resultados a una escalabilidad numérica donde, si los resultados son positivos tendrá un valor de 1, si son negativos de 0 y si no existe relación alguna se tomará como nula (-). Una vez completado los datos de dicha relación se realiza un cálculo donde se promedia la sumatoria de los valores obtenidos de un atributo por cada métrica evaluada, y la división de dicha sumatoria por la cantidad de métricas evaluadas (solo se promedian las que arrojan un resultado, las nulas no). Este valor es el que va a tener el atributo dentro de una tabla que medirá si los atributos fueron buenos, regulares o malos. Se lograron los siguientes resultados:

Evaluación de la solución

Atributos/Métricas	TOC	RPC	Promedio
Responsabilidad	1	-	1
Complejidad de Implementación	1	-	1
Reutilización	1	1	1
Acoplamiento	-	1	1
Complejidad de Mantenimiento	-	1	1
Cantidad de pruebas	-	1	1

Tabla 8 (Resultados evaluados de la relación Atributos/Métricas por cada componente que integran la solución)

Categoría	Rango de valores
Malo	<= 0.4
Regular	>0.4 y <=0.7
Bueno	>0.7

Tabla 9 (Rango de valores para la evaluación técnica de los atributos de calidad evaluados por cada métrica)

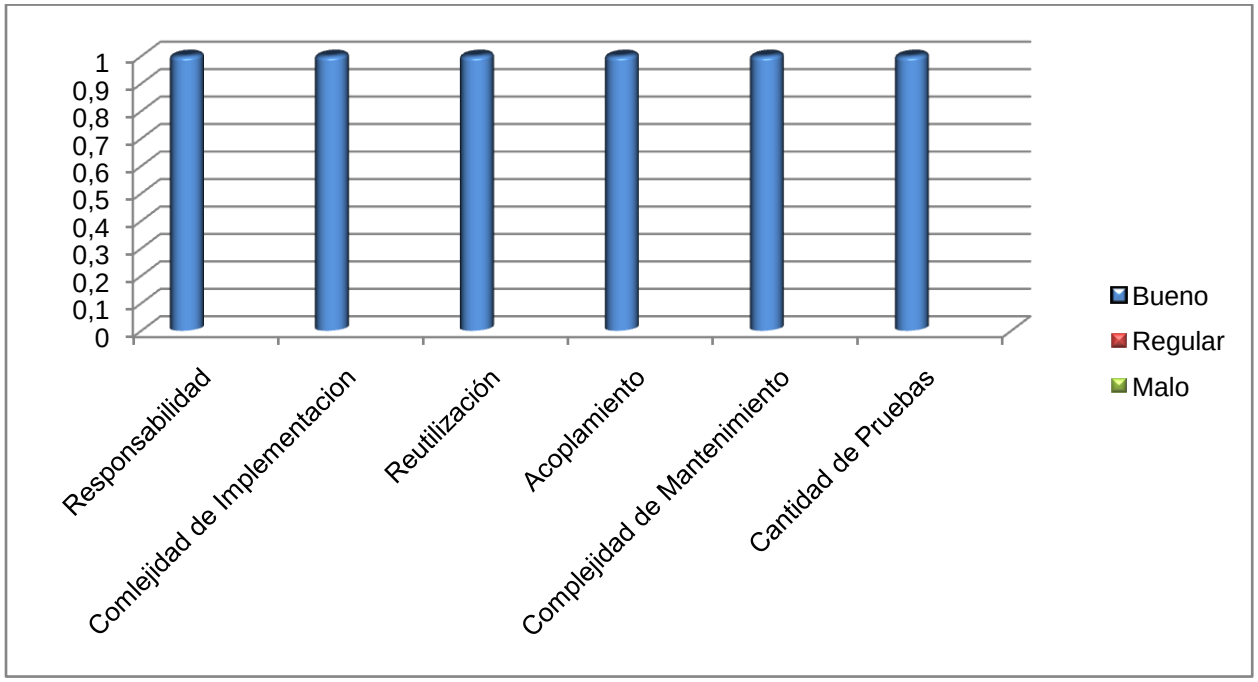

Figura 46 (Gráfica de los resultados obtenidos de los atributos de calidad evaluados en las métricas)

3.5 Conclusiones

En este capítulo se utilizaron los principales métodos para obtener el nivel de calidad de la solución propuesta. Mediante instrumentos inspirados en métricas para la calidad del diseño y basándose sobre los atributos de calidad trazados para llevar a cabo una medición estricta de cada uno de los componentes, los cuales permitieron afirmar que el diseño realizado se puede valorar de muy bueno debido a las métricas aplicadas que arrojaron valores de medición excelentes.

Conclusiones

Al realizar un estudio del estado del arte de los sistemas de gestión de seguridad tanto dentro como fuera del país, arrojó como resultado que en la forma en que estos implementaban la seguridad no cumplían con las necesidades del proyecto.

Para darle solución a esta problemática se desarrolló SIGIS el cual gestiona la seguridad de forma centralizada, el tema de multi-entidad tan importante para la compartimentación de la información, la administración de conexiones, la configuración de perfil del usuario y la integración con otros sistemas.El sistema forma parte de las políticas definidas en la arquitectura de seguridad.

La solución fue probada por la línea de Calidad del proyecto ERP Cuba donde se verificó que se cumplieran todos los requisitos funcionales. Además se realizó una validación del diseño mediante la utilización de instrumentos de medición que se inspiraron en métricas para la calidad del diseño. Los resultados arrojados permitieron concluir que el diseño presentaba valores positivos en indicadores de calidad tales como Reutilización, Facilidad de Mantenimiento, Complejidad del Diseño, Complejidad de Implementación, Acoplamiento, Cantidad de pruebas, entre otros. Esto apoya la afirmación de que el diseño desarrollado se puede considerar como excelente.

Actualmente el sistema se encuentra brindando servicios en 11 sistemas de la Universidad como Akademos 2.0, Cedrux, Aduana, Sistema de supervisión y control de los PSI (Hoyo), Sistema General PATDSI, Sistema de Gestión Estadística (CENTALD), Fuerza de Trabajo Calificada, LiberGIS, Minería de datos (CENTALD), Generador de reportes (CENTALD), Sistema informático para la gestión de auditoria y control (SIGAC) y Sistema de Mando y Estado Mayor (SIMEM).

Recomendaciones

El equipo de trabajo que realiza la presente investigación considera importante distinguir las siguientes recomendaciones:

- Los sistemas web de gestión desarrollados en la UCI y en Cuba se suscriban a SIGIS, para así estos logren obtener una buena seguridad.
- Trabajar en nuevas versiones del sistema para fortalecer la gestión de la seguridad brindada por este.

Aval

Por la presente damos constancia de que la “Solución para la gestión integral de sistemas en entorno multientidad” se está aplicando en el proyecto Cedrux, ubicado en la Universidad de las Ciencias Informáticas.

El Sistema de Gestión Integral de Seguridad SIGIS esta concebido para garantizar la seguridad en un entorno de varias aplicaciones. Esta desarrollado sobre el marco de trabajo del Sistema de Gestión de Recursos Empresariales Cedrux, lo que permite que todos los proyectos que se desarrollen sobre este marco de trabajo no tengan que preocuparse por la seguridad de sus aplicaciones ya que SIGIS es un componente más de él.

Con la reutilización de SIGIS se ahorran cuantiosos recurso humanos y materiales puesto que adquirir un sistema de este tipo en el mercado internacional resultaría costoso y si se decide implementar un sistema que garantice la seguridad de cada una de las aplicaciones que se desarrollen gastaríamos millones de dólares innecesariamente y nunca se lograría estandarizar este proceso. Este sistema se encuentra debidamente documentado permitiendo que usuarios y desarrolladores dominen a fondo todas las funcionalidades y puedan adaptarlo a sus necesidades.

SIGIS forma parte de una arquitectura de seguridad que no solo garantiza la seguridad de las aplicaciones sino que tienen en cuenta la seguridad durante todas las fases del software. Por lo que utilizar este sistema permitiría la reutilización de esta arquitectura de seguridad que ya se encuentra formalizada y aprobada para su puesta en práctica con su expediente arquitectónico, listas de chequeo, normas y configuraciones seguras.

Por todas las facilidades anteriormente expuestas, SIGIS junto al marco de trabajo de Cedrux, han sido reutilizados en 11 proyectos de la universidad en los cuales ha tenido una gran aceptación ya que estudiantes y profesionales se han logrado familiarizar con el mismo en un corto periodo de tiempo.

Y para constancia de ello en mi condición de Directora de Producción, firmo la presente a los 13 días del mes de mayo del 2009.

_____

Yadenis Piñero
Directora de Producción
Centro de soluciones Generales de Gestión

Referencias Bibliográfícas

1. **Microsoft.** Microsoft.com. *Microsoft Technet Seguridad.* [Online] Microsoft Corporation, 2009. http://www.microsoft.com/latam/technet/seguridad/articulos/bpsegcorp.mspx..

2. **DelitosInformáticos.com.** DelitosInformáticos.com. *Seguridad.* [Online] Marzo 25, 2001. [Cited: Enero 9, 2009.] http://www.delitosinformaticos.com/seguridad/clasificacion.shtml.

3. **Potencier, Fabier and Zaninotto, Francois.** *The Definitive Guide to Symfony.* s.l. : Apress, 2008. 978-1590597866.

4. **Rigazzi, Pablo.** Zend Framework Playground. [Online] WordPress, Septiembre 9, 2008. [Cited: Febrero 3, 2009.] http://spanish.zendfw.com/.

5. **inc, EllisLab.** CodeIgniter. [Online] EllisLab, 2009. [Cited: Febreo 6, 2009.] http://www.codeigniter.com.

6. **Tejeda, Deivinson.** KumbiaPHP Framework. [Online] KumbiaPHP Framework, 2007. [Cited: Febrero 8, 2009.] http://www.kumbiaphp.com/blog/.

7. **Corporation, Cake Development.** Cake Development Corporation. [Online] Cake Software Fundation Inc, 2007 - 2009. [Cited: Febrero 12, 2009.] http://www.cakedc.com/.

8. **Marín Mártires, Lázaro Antonio and Alvarado Capriles, Luis Ramón.** *Sistema de autenticación y autorización centralizado.* Ciudad Habana : UCI, 2008.

9. **Eliurkis.** DeepinPHP. [Online] DeepinPHP, 2006. [Cited: Febrero 15, 2009.]

Bibliografía

Microsoft Corporation. Microsoft TechNet Seguridad. [Online] 2009. http://www.microsoft.com/latam/technet/seguridad/articulos/bpsegcorp.mspx.

Delitosinformáticos.com. Delitosinformáticos.com. Seguridad. [Online] Marzo 25, 2001. [Cited: Febrero 12, 2009.] http://www.delitosinformaticos.com/seguridad/clasificacion.shtml.

Potencier, Fabier and Francois, Zaninotto. The Definitive Guide to Symfony. s.l.: Apress, 2008. ISBN-13: 978-1590597866.

Zend Framework Playground. Zend Framework Playground. [Online] WordPress, Septiembre 2008, 2008. [Cited: Abril 21, 2009.] http://spanish.zendfw.com/.

EllisLab, inc. CodeIgniter. [Online] EllisLab, 2001. http://www.codeigniter.com.

KumbiaPHP Framework. KumbiaPHP Framework. [Online] 2007. http://www.kumbiaphp.com.

Cake Development Corporation. Cake Development Corporation. CakePHP. [Online] Cake Development Corporation, 2007. http://www.cakedc.com/.

Marín Mártires, Lázaro Antonio and Capriles Alvarado, Luis Ramón. Sistema de autenticación y autorización centralizado. Ciudad Habana: Universidad de las Ciencias Informáticas., 2008.

Eliurkis. DeepinPHP. [Online] 2006. [Cited: Mayo 20, 2009.] http://www.deepinphp.com/2007/09/01/single-sign-on-sistema-de-autenticacion-unico.

UCI. Teleformación. [Online] Universidad de las Ciencias Informáticas. http://teleformacion.uci.cu.

UCI. Biblioteca. [Online] Universidad de las Ciencias Informáticas. http://biblioteca.uci.cu/.

Symfony.es. Symfony.es[Online] 2009. [Cited: Febrero 12, 2009.] http://www.symfony.es.

SICV. SeguridadInformatica.es. [Online] Seguridad Informática y Creación Visual, 2009. http://www.seguridadinformatica.es/.

SeguInfoNews. SeguInfoNews. [Online] Enero 29, 2009. [Cited: Mayo 20, 2009.] http://blog.segu-info.com.ar/2009/01/ataque-que-es-cross-site-request.html.

Ramos, Kenyie Araya. Desarrolloweb. [Online] 2006. [Cited: Febrero 10, 2009.] http://www.desarrolloweb.com/articulos/introduccion-cross-site-scripting.html.

Pressman, R. Ingenería del software: Un enfoque práctico. McGraw Hill: s.n., 2000.

PHP, Comunidad. Comunidad_PHP. [Online] Universidad de las Ciencias Informáticas. http://php.uci.cu/.

Bibliografía

Lazo Ochoa, Rene and Yzquierdo Herrera, Raykenler. El modelo de diseño del sistema HyperWeb.Módulos de Tratamiento Farmacológico y Configuración. Ciudad Habana: Universidad de las Ciencias Informáticas, 2007.

Fundation, OpenLDAP. OpenLDAP. [Online] net Boolean, Marzo 3, 2008. http://www.openldap.org.

Fundation, Ext. Ext JS. [Online] EXT; LLC, 2006. http://www.extjs.com/.

FormatoWeb. FormatoWeb. XSS. [Online] 2007. http://www.formatoweb.com.ar/blog/2007/10/10/ataques-xss-el-peligro-de-hacer-echo-get-var/.

Abrams, Brad. Brad Abrams. [Online] Microsoft Corporation, 2009. http://blogs.msdn.com/brada/archive/2004/02/03/67024.aspx.

Symfony. Open Source PHP Web Framework. [Online] Symfony & Sensio Labs, 2009. http://www.symfony-project.org/.

Microsoft Corporation. Microsoft Solution Framework. 2006.

Equipo Arquitectura del ERP Cedrux. Especificación Técnica para el marco de la arquitectura. 2008.

Anexos

Anexo1: Requisitos funcionales del Sistema de Gestión Integral de Seguridad (SIGIS)

R1 Requisito Funcional Gestionar Sistema

Especificación del R1.1 Cargar sistemas

Conceptos tratados	**Conceptos**	**Atributos**
	Sistema	Nombre del sistema, abreviatura, servidor, descripción, gestor de BD, BD, esquema.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener sistemas registrados.	Registrar Sistema
Descripción	Mostrar sistemas existentes.	
Validaciones	No procede	
Post - Condiciones	Mostrar los sistemas registrados	
Post - Requisito	No procede	

Tabla 10 (Especificación del requisito R1.1 cargar sistemas)

Especificación del R1.2 Registrar sistemas

Conceptos tratados	**Conceptos**	**Atributos**
	Sistema	Nombre del sistema, abreviatura, servidor, descripción, gestor de BD, BD, esquema.
	Servidor	Nombre del servidor, dirección ip, tipo de servidor, descripción.
	Gestor de BD	Nombre de gestor de BD
	BD	Nombre de la BD
	Esquema	Nombre de esquema de BD
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener servidores registrados.	Registrar Servidor.
	Tener gestores de BD registrador.	Registrar Gestor de BD.
	Tener registradas BD.	Registrar BD

	Tener esquemas registrados.	Registrar Esquema
Descripción	Para registrar un nuevo sistema se debe entrar los datos del sistema que son obligatorios como el nombre del sistema y la abreviatura y seleccionar el servidor, el gestor de BD, la BD y el esquema que utilizará estos datos pueden ser nulos. Notificar datos erróneos que se puedan haber insertado y permitir la corrección de los mismos. Aplicar registro. Cancelar registro.	
Validaciones	El sistema valida los datos según lo descrito en ERP-ARQ Modelo conceptual v1.0.	
Post-condiciones	Se ha registrado un nuevo sistema.	
Post-requisito	No procede.	

Tabla 11 (Especificación del requisito R1.2 registrar sistemas)

Especificación del R1.3 Modificar sistemas

Conceptos tratados	**Conceptos**	**Atributos**
	Sistema	Nombre del sistema, abreviatura, servidor, descripción, gestor de BD, BD, esquema.
	Servidor	Nombre del servidor, dirección ip, tipo de servidor, descripción.
	Gestor de BD	Nombre de gestor de BD
	BD	Nombre de la BD
	Esquema	Nombre de esquema de BD
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener sistemas registrados.	Registrar Sistema.
	Tener gestores de BD registrador.	Registrar Gestor de BD.
	Tener registradas BD.	Registrar BD
	Tener esquemas registrados.	Registrar Esquema
Descripción	Seleccionar el sistema que será modificado. Modificar sistema. Notificar datos erróneos que se puedan haber insertado y permitir la corrección de los mismos. Aplicar Modificación.	

	Cancelar Modificación.
Validaciones	El sistema valida los datos según lo descrito en ERP-ARQ Modelo conceptual v1.0.
Post-condiciones	Se han modificado los datos del sistema.
Post-requisito	No procede.

Tabla 12 (Especificación del requisito R1.3 modificar sistemas)

Especificación del R1.4 Eliminar sistemas

Conceptos tratados	**Conceptos**	**Atributos**
	Sistema	Nombre del sistema, abreviatura, servidor, descripción, gestor de BD, BD, esquema.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener sistemas registrados.	Registrar Sistema.
Descripción	Seleccionar el sistema que será eliminado. Mostrar mensaje de confirmación.	
Validaciones	No procede	
Post-condiciones	Se ha eliminado el sistema.	
Post-requisito	No procede.	

Tabla 13 (Especificación del requisito R1.4 eliminar sistemas)

Especificación del R1.5 Importar sistemas.

Conceptos tratados	**Conceptos**	**Atributos**
	Sistema	Nombre del sistema, abreviatura, servidor, descripción, gestor de BD, BD, esquema.
Precondiciones	**Precondiciones**	**Pre-requisito**
	No procede	No procede.
Descripción	Seleccionar el sistema que será importado desde una dirección externa. Mostrar mensaje de confirmación.	
Validaciones	No procede	
Post-condiciones	Se ha importado el sistema.	
Post-requisito	No procede.	

Tabla 14 (Especificación del requisito R1.5 importar sistemas)

Especificación del R1.6 Exportar sistemas.

Conceptos tratados	**Conceptos**	**Atributos**
	Sistema	Nombre del sistema, abreviatura, servidor, descripción, gestor de BD, BD, esquema.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener sistemas registrados.	Registrar Sistema.
Descripción	Seleccionar el sistema que será exportado. Mostrar mensaje de confirmación.	
Validaciones	No procede	
Post-condiciones	Se ha exportado el sistema.	
Post-requisito	No procede.	

Tabla 15 (Especificación del requisito R1.6 exportar sistemas)

R13 Requisito Funcional Gestionar Usuarios

Especificación del R13.1 Cargar Usuarios

Conceptos tratados	**Conceptos**	**Atributos**
	Usuario	Nombre del usuario, dominio, tema, idioma, escritorio, rango ip, contraseña, cargo, área, entidad.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener usuarios registrados.	Registrar usuario
Descripción	Mostrar usuarios existentes	
Validaciones	No Procede	
Post-condiciones	Mostrar los usuarios registrados.	
Post-requisito	No procede.	

Tabla 16 (Especificación del requisito R13.1 cargar usuarios)

Especificación del R13.2 Registrar Usuario

Conceptos tratados	**Conceptos**	**Atributos**
	Usuario	Nombre del usuario, dominio, tema, idioma, escritorio, rango ip, contraseña, cargo, área, entidad.
	Dominio	Denominación, descripción y estructura.
	Tema	Denominación, abreviatura y descripción.
	Idioma	Denominación y abreviatura.
	Escritorio	Denominación, abreviatura y descripción.
	Servidor	Nombre del servidor, dirección ip, tipo de servidor,

		descripción.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener registrados sistemas.	Registrar sistema.
	Tener servidores registrados	Registrar servidor.
	Tener dominios registrados	Registrar dominio.
	Tener temas registrados	Registrar tema.
	Tener idiomas registrados	Registrar idioma.
	Tener escritorios registrados	Registrar escritorio.
Descripción	Solicitar datos de usario Registrar usuario. Notificar datos erróneos que se puedan haber insertado y permitir la corrección de los mismos. Aplicar registro. Cancelar registro.	
Validaciones	El sistema valida los datos según lo descrito en ERP-ARQ Modelo conceptual v1.0.	
Post-condiciones	Se ha registrado un nuevo usuario. Mostrar mensaje de registro realizado.	
Post-requisito	No procede.	

Tabla 17 (Especificación del requisito R13.2 registrar usuario)

Especificación del R13.3 Modificar Usuario

Conceptos tratados	**Conceptos**	**Atributos**
	Usuario	Nombre del usuario, dominio, tema, idioma, escritorio, rango ip, contraseña, cargo, área, entidad.
	Dominio	Denominación, descripción y estructura.
	Tema	Denominación, abreviatura y descripción.
	Idioma	Denominación y abreviatura.
	Escritorio	Denominación, abreviatura y descripción.
	Servidor	Nombre del servidor, dirección ip, tipo de servidor, descripción.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener registrados sistemas.	Registrar sistema.
	Tener servidores registrados	Registrar servidor.

	Tener dominios registrados	Registrar dominio.
	Tener temas registrados	Registrar tema.
	Tener idiomas registrados	Registrar idioma.
	Tener escritorios registrados	Registrar escritorio.
Descripción	Seleccionar el usuario que será modificado. Modificar Usuario. Notificar datos erróneos que se puedan haber insertado y Permitir la corrección de los mismos. Cancelar Modificación.	
Validaciones	El sistema valida los datos según lo descrito en ERP-ARQ Modelo conceptual v1.0.	
Post-condiciones	Se ha registrado un nuevo usuario. Mostrar mensaje de registro realizado.	
Post-requisito	No procede.	

Tabla 18 (Especificación del requisito R13.3 modificar usuarios)

Especificación del R13.4 Eliminar Usuario

Conceptos tratados	**Conceptos**	**Atributos**
	Usuario	Nombre del usuario, dominio, tema, idioma, escritorio, rango ip, contraseña, cargo, área, entidad.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener usuarios registrados.	Registrar usuario.
Descripción	Se selecciona el usuario que se desea eliminar.	
Validaciones	No procede.	
Post-condiciones	Se ha eliminado el usuario. Mostrar mensaje de confirmación.	
Post-requisito	No procede.	

Tabla 19 (Especificación del requisito R13.4 eliminar usuarios)

Especificación del R13.5 Asignar Rol

Conceptos tratados	**Conceptos**	**Atributos**
	Usuario	Nombre del usuario, dominio, tema, idioma, escritorio, rango ip, contraseña, cargo, área, entidad.
	Rol	Nombre de la rol, abreviatura, descripción.

Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener registrados usuarios.	Registrar usuario
	Tener registrados roles.	Registrar rol.
Descripción	Seleccionar usuario al cual se le asignará un rol Seleccionar el o los roles que se asignarán y por cada uno seleccionar las entidades en las que lo tendrá. Aplicar restricción. Cancelar restricción.	
Validaciones	Un usuario puede tener muchos roles pero solo puede tener un rol en una entidad determinada.	
Post-condiciones	Se le han asignado roles a un usuario.	
Post-requisito	No procede.	

Tabla 20 (Especificación del requisito R13.5 asignar rol)

Especificación del R13.6 Cambiar Contraseña

Conceptos tratados	**Conceptos**	**Atributos**
	Usuario	Nombre del usuario, dominio, tema, idioma, escritorio, rango ip, contraseña, cargo, área, entidad.
Precondiciones	**Precondiciones**	**Pre-requisito**
	Tener registrados usuarios.	Registrar usuario
Descripción	Seleccionar usuario al cual se le cambiara la contraseña Aplicar cambio de contraseña. Cancelar cambio de contraseña.	
Validaciones	No procede	
Post-condiciones	Se ha cambiado la contraseña. Mostrar mensaje de cambio realizado.	
Post-requisito	No procede.	

Tabla 21 (Especificación del requisito R13.6 cambiar contraseña)

Anex2: Diseño de casos de prueba del Sistema Gestión Integral de Seguridad.

Caso de prueba para el requisito R1 Gestionar Sistema.

Caso de prueba para R1.2 Registrar Sistemas

Condiciones de ejecución

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el subsistema Seguridad, en el módulo Configurar sistemas, en la interfaz Sistemas.
- El sistema que se desea adicionar no ha sido adicionado antes al sistema.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Registrar sistema	Se adiciona un nuevo sistema.	EP 1.1: Registrar sistema	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar.** – Se insertan todos los datos. – Se presiona el botón **Aceptar**.
		EP 1.2: Registrar sistema dejando campos requeridos en blanco.	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar**. – Se introducen los datos dejando campos requeridos en blanco. – Se presiona el botón **Aceptar**.

		EP 1.3: Registrar sistema introduciendo error en los datos.	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar**. – Se introducen los datos del sistema que se desea adicionar en el formulario introduciendo errores en los datos. – Se presiona el botón **Aceptar**.
		EP 1.4: Aplicar	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar**. – Se introducen los datos del sistema que se desea adicionar en el formulario. – Se presiona el botón **Aplicar**. – Se presiona el botón **Aceptar**.
		EP 1.5: Cancelar	– Escoger el subsistema al que se le desea agregar el subsistema. – Se presiona el botón **Adicionar**. – Se introducen o no los datos en el formulario. – Se presiona el botón **Cancelar**.

Tabla 22 (Requisitos a probar para caso de prueba del R1.2)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación	Campo de texto	No	Combinación de letras.
2	Abreviatura	Campo de texto	No	Combinación de letras.
3	Icono	Campo de texto	Si	Combinación de letras.
4	Servidores	Treepanel	No	Lista desplegable de los servidores que existen.
5	Descripción	Campo de texto	Si	Combinación de letras, números y caracteres especiales.
6	Externo	Check box	Si	Cuadro de selección.
7	Servidor web	Campo de texto	Si	Combinación de letras.

Tabla 23 (Descripción de variables para caso de prueba del R1.2)

Juegos de datos a probar.

Id del escenario	Escenario	Denominación	Abreviatura	Icono	Servidores	Descripción	Externo	Servidor web	Respuesta del sistema	Resultado de la prueba
EP1.1	Registrar sistema.	V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se adiciona el sistema y se guarda la información	
EP1.2	Registrar sistema dejando campos requeridos en blanco.	I(vacío)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se muestra el campo de texto en rojo que indica que no se pueden dejar	
		V(sist seguridad)	I(vacío)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	I(vacío)	V(seguridad)	V(publict)	V(Ldap)		
		I(Vacio)	I(Vacio)	I(Vacio)	I(Vacio)	I(Vacio)	I(Vacio)	I(vacio)		

EP 1.3	Registrar sistema introduciendo error en los datos.	I(-*/)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se muestra el campo de texto en rojo que indica que no se pueden entrar caracteres	
		V(sist seguridad)	I(-*/)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	I(-*/)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	I(seg34)	V(120)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	I(/*-)		
EP1.4	Aplicar	V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se adiciona el sistema y se guarda la información adicionada en el módulo	

									de configurar sistemas.	
		I(vacío)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se muestra el campo de texto en rojo que indica que no se pueden dejar campos en blanco y se mantiene en la ventana Adicionar	
		V(sist seguridad)	I(vacío)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	I(vacío)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	I(vacío)	V(publict)	V(Ldap)		

									sistema.	
		I(-*/)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)	Se muestra el campo de texto en rojo que indica que no se pueden entrar caracteres inválidos y se mantiene en la ventana Adicionar sistema.	
		V(sist seguridad)	I(-*/)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	I(-*/)	V(Ldap)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	I(vacío)	V(120)	V(seguridad)	V(publict)	V(Ldap)		
		V(sist seguridad)	V(seg)	V(seg34)	V(Ldap)	V(seguridad)	V(publict)	I(/*-)		

EP1.5	Cancelar	NA	NA	NA	NA	NA	NA		Se cancela	

Tabla 24 (Juego de datos a probar para caso de prueba del R1.2)

Caso de prueba para R1.3 Modificar Sistemas

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el subsistema Seguridad, en el módulo Configurar sistemas, en la interfaz Sistemas.
- El sistema que se desea modificar ha sido adicionado anteriormente al sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Modificar sistema	Se adiciona un nuevo sistema.	EP 1.1: Modificar sistema	– Escoger el subsistema al que se le desea modificar el subsistema. – Se presiona el botón **Modificar.** – Se insertan todos los datos. – Se presiona el botón **Aceptar**.
		EP 1.2: Modificar sistema dejando campos requeridos en blanco.	– Escoger el subsistema al que se le desea modificar el subsistema. – Se presiona el botón **Modificar**. – Se introducen los datos dejando campos requeridos en blanco. – Se presiona el botón **Aceptar**.

		EP 1.3: Modificar sistema introduciendo error en los datos.	– Escoger el subsistema al que se le desea modificar el subsistema. – Se presiona el botón **Modificar**. – Se introducen los datos del sistema que se desea modificar en el formulario introduciendo errores en los datos. – Se presiona el botón **Aceptar**.
		EP 1.4: Cancelar	– Escoger el subsistema al que se le desea modificar el subsistema. – Se presiona el botón **Modificar**. – Se introducen o no los datos en el formulario. – Se presiona el botón **Cancelar**.

Tabla 25 (Requisitos a probar para caso de prueba del R1.3)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Denominación	Campo de texto	No	Combinación de letras.
2	Abreviatura	Campo de texto	No	Combinación de letras.
3	Icono	Campo de texto	Si	Combinación de letras.
4	Servidores	Árbol	No	Lista desplegable de los servidores que existen.
5	Descripción	Campo de texto	Si	Combinación de letras, números y caracteres especiales.

6	Externo	Check box	Si	Cuadro de selección
7	Servidor web	Campo de texto	Si	Combinación de letras.

Tabla 26 (Descripción de variables para caso de prueba del R1.3)

Juegos de datos a probar.

Id del escenario	Escenario	Denominación	Abreviatura	Icono	Servidores	Descripción	Externo	Servidor web	Respuesta del sistema	Resultado de la prueba
EP1.1	Modificar sistema.	V(sist seguridad)	V(seg)	V(seg34)	V(LDAP)	V(seguridad)	V(publict)	V(LDAP)	Se modifica el sistema y se guarda la información modificada en el módulo de	
EP1.2	Modificar sistema dejando campos requeridos en blanco.	I(vacío)	V(seg)	V(seg34)	V(LDAP)	V(seguridad)	V(publict)	V(LDAP)	Se muestra el campo de texto en rojo que indica que no se pueden dejar	
		V(sist seguridad)	I(vacío)	V(seg34)	V(LDAP)	V(seguridad)	V(publict)	V(LDAP)		
		V(sist seguridad)	V(seg)	I(vacío)	V(LDAP)	V(seguridad)	V(publict)	V(LDAP)		

		V(sist seguridad)	V(seg)	V(seg34)	I(vacío)	V(seguridad)	V(publict)	V(LDAP)	campos en blanco y se mantiene en la ventana Modificar sistema.	
		V(sist seguridad)	V(seg)	V(seg34)	V(LDAP)	I(vacío)	V(publict)	V(LDAP)		
		V(sist seguridad)	V(seg)	V(seg34)	V(LDAP)	V(seguridad)	I(vacío)	V(LDAP)		
		V(sist seguridad)	V(seg)	V(seg34)	V(LDAP)	V(seguridad)	V(publict)	I(vacío)		
EP 1.3	Modificar sistema introduciendo error en los datos.	I(-*/)	V(seg)	V(seg34)	V(LDAP)	V(seguridad)	V(publict)	V(LDAP)	Se muestra el campo de texto en rojo que indica que no se pueden entrar caracteres inválidos y se mantiene en la ventana	
		V(sist seguridad)	I(-*/)	V(seg34)	V(LDAP)	V(seguridad)	V(publict)	V(LDAP)		
		V(sist seguridad)	V(seg)	I(-*/)	V(LDAP)	V(seguridad)	V(publict)	V(LDAP)		
		V(sist seguridad)	V(seg)	I(vacío)	V(120)	V(seguridad)	V(publict)	V(LDAP)		
		V(sist seguridad)	V(seg)	V(seg34)	V(LDAP)	I(/*-)	V(publict)	V(LDAP)		

		V(sist seguridad)	V(seg)	V(seg34)	V(LDAP)	V(seguridad)	I(250)	V(LDAP)	Modificar sistema.	
		V(sist seguridad)	V(seg)	V(seg34)	V(LDAP)	V(seguridad)	V(publict)	I(/*-)		
EP1.4	Cancelar	NA	NA	NA	NA	NA	NA		Se cancela la operación.	

Tabla 27 (Juegos de datos a probar para caso de prueba del R1.3)

Caso de prueba para R1.4 Eliminar Sistema

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el subsistema Seguridad, en el módulo Configurar sistemas, en la interfaz Sistemas.
- El sistema que se desea eliminar se encuentra registrado en el sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Eliminar sistema.	Se elimina el sistema seleccionado.	EP1.1: Eliminar sistema.	- Se selecciona el sistema que se desea eliminar. - Se presiona el botón **Eliminar** perteneciente a Gestionar sistemas. - Se presiona el botón **Aceptar**.

		EP1.2: Cancelar.	- Se selecciona el sistema que se desea eliminar. - Se presiona el botón **Eliminar** perteneciente a Gestionar sistemas. - Se presiona el botón **Cancelar**.

Tabla 28 (Requisitos a probar para caso de prueba del R1.4)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

Tabla 29 (Descripción de variables para caso de prueba del R1.4)

Juego de datos a probar.

Id del escenario	Escenario	Sistema	Respuesta del sistema	Resultado de la prueba
EP1.1	Eliminar sistema	NA	Se selecciona el sistema que se desea eliminar, se efectúa la operación y el mismo queda eliminado satisfactoriamente.	
EP1.2	Cancelar	NA	Se cancela la operación.	

Tabla 30 (Juego de datos a probar para caso de prueba del R1.4)

Caso de prueba para R1.5 Importar Sistema

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el subsistema Seguridad, en el módulo Configurar sistemas, en la interfaz Sistemas.
- El sistema que se desea exportar se encuentra registrado en el sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Importar sistema.	Se importa el sistema.	EP1.1: Importar sistema.	- Se selecciona el sistema adonde se desea importar el sistema. - Se presiona el botón **Importar** perteneciente a Gestionar sistemas. - Se selecciona el fichero. - Se presiona el botón **Aceptar**.
		EP1.2: Cancelar.	- Se selecciona el sistema que se desea importar. - Se presiona el botón **Importar** perteneciente a Gestionar sistemas. - Se presiona el botón **Cancelar**.

Tabla 31 (Requisitos a probar para caso de prueba del R1.5)

Descripción de variables

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

Tabla 32 (Descripción de variable para caso de prueba del R1.5)

Juego de datos a probar.

Id del escenario	Escenario	Sistema	Respuesta del sistema	Resultado de la prueba
EP1.1	Importar sistema	NA	Se selecciona el sistema que se desea importar, se efectúa la operación y sistema se importa en un sistema	
EP1.2	Cancelar	NA	Se cancela la operación.	

Tabla 33 (Juego de datos a probar para caso de prueba del R1.5)

Caso de prueba para R1.6 Exportar Sistema

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el subsistema Seguridad, en el módulo Configurar sistemas, en la interfaz Sistemas.
- El sistema que se desea exportar se encuentra registrado en el sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Exportar sistema.	Se exporta el sistema seleccionado.	EP1.1: Exportar sistema.	- Se selecciona el sistema que se desea exportar - Se presiona el botón **Exportar** perteneciente a Gestionar sistemas. - Se selección donde se va a guardar el fichero. - Se presiona el botón **Guardar**.
		EP1.2: Cancelar.	- Se selecciona el sistema que se desea exportar. - Se presiona el botón **Exportar** perteneciente a Gestionar sistemas. - Se presiona el botón **Cancelar**.

Tabla 34 (Requisitos a probar para caso de prueba del R1.6)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

Tabla 35 (Descripción de variables para caso de prueba del R1.6)

Juego de datos a probar.

Id del escenario	Escenario	Sistema	Respuesta del sistema	Resultado de la prueba
EP1.1	Exportar sistema	NA	Se selecciona el sistema que se desea exportar, se efectúa la operación y sistema se exporta en un fichero XML.	
EP1.2	Cancelar	NA	Se cancela la operación.	

Tabla 36 (Juego de datos a probar para caso de prueba del R1.6)

Casos de prueba para el requisito R13 Gestionar Usuarios.

Caso de prueba para R13.2 Registrar Usuario.

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el módulo Seguridad, en el submódulo Configurar usuarios en la interfaz Usuarios
- El usuario que se desea adicionar no ha sido adicionado antes al sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar usuario	Se adiciona un nuevo usuario al sistema	EP 1.1: Adicionar un nuevo usuario al sistema.	– Se presiona el botón **Adicionar**. – Se introducen los datos del usuario que se desea adicionar en el formulario. – Se presiona el botón **Aceptar**.
		EP 1.2: Adicionar un usuario dejando campos requeridos en blanco.	– Se presiona el botón **Adicionar**. – Se introducen los datos del usuario que se desea adicionar en el formulario dejando al menos un campo requerido en blanco. – Se presiona el botón **Aceptar**.
		EP 1.3: Adicionar un usuario introduciendo errores en los datos.	– Se presiona el botón **Adicionar**. – Se introducen los datos del usuario que se desea adicionar en el formulario introduciendo errores en los datos. – Se presiona el botón **Aceptar**.
		EP 1.4: Aplicar.	– Se presiona el botón **Adicionar**. – Se introducen los datos del usuario que se desea adicionar en el formulario. – Se presiona el botón **Aplicar**.
		EP 1.5: Cancelar.	– Se presiona el botón **Adicionar**. – Se introducen o no los datos en el formulario. – Se presiona el botón **Cancelar**.

Tabla 37 (Requisitos a probar para caso de prueba del R13.2)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Rango IP	Campo de texto	No	Números
2	Tipo de escritorio	Combobox	No	Lista desplegable escritorios existentes.
3	Dominio	Combobox	No	Lista desplegable dominios existentes.
4	Idioma	Combobox	No	Lista desplegable idiomas existentes.
5	Tema	Combobox	No	Lista desplegable temas existentes.
6	Servidores	Combobox	No	Lista desplegable servidores existentes.
7	Entidad	Campo de texto	No	Lista desplegable entidades existentes.
8	Cargo	Campo de texto	No	Lista desplegable cargos existentes.
9	Area	Campo de texto	No	Lista desplegable aéreas existentes.
10	Usuario	Campo de texto	No	Letras
11	Contraseña	Campo de texto	No	Letras, números y caracteres.
12	Confirmar Contraseña	Campo de texto	No	Letras, números y caracteres.

Tabla 38 (Descripción de variables para caso de prueba del R13.2)

Juego de datos a probar.

Id del escenario	Escenario	Rango IP	Tipo de escritorio	Dominio	Idioma	Tema	Servidores	Entidad	Cargo	Usuario	Contraseña	Confirmar Contraseña	Respuesta del sistema
EP 1.1.	Adicionar un nuevo usuario al sistema.	V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok)	V(ok)	Se adiciona el usuario y se guarda la información adicionada.
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(12)	V(12)	
EP 1.2.	Adicionar un usuario dejando campos requeridos en blanco	I(vacio)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	Se muestra el campo de texto en rojo que indica que no se pueden dejar campos en blanco y se mantiene en la ventana Adicionar usuario.
		V(10.1.1.1)	I(vacio)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	I(vacio)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	I(vacio)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	I(vacio)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	I(vacio)	V(UCI)	V(Jefe personal)	I(vacio)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	I(vacio)	V(Jefe personal)	V(rosa)	I(vacio)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	I(vacio)	V(rosa)	V(ok12/*)	I(vacio)	
		I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	Se muestran los campos de textos en rojo que indica que no se pueden

													dejar campos en blanco y se mantiene la interfaz Adicionar funcionalidad.
EP 1.3.	Adicionar un usuario introduciendo errores en los datos.	I(ho*/)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	Se muestra el campo de texto en rojo que indica que no se pueden entrar caracteres inválidos y se mantiene en la ventana Adicionar usuario.
		V(10.1.1.1)	V(estándar	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	I(/*-)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(120)	V(ok12/*)	
EP 1.4.	Aplicar	V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok)	V(ok)	Se adiciona el usuario y se guarda la información adicionada
		I(vacio)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	Se muestra el campo de texto en rojo que indica que no se pueden dejar campos en blanco y se mantiene en la ventana Adicionar usuario.
		V(10.1.1.1)	I(vacio)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	I(vacio)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	I(vacio)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	

		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	I(vacio)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	I(vacio)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	I(vacio)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	I(vacio)	V(rosa)	V(ok12/*)	I(vacio)	
		I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	Se muestran los campos de textos en rojo que indica que no se pueden dejar campos en blanco y se mantiene la interfaz Adicionar funcionalidad.
		I(ho*/)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	I(ho*/)	V(ok12/*)	V(ok12/*)	Se muestra el campo de texto en rojo que indica que no se pueden entrar caracteres inválidos y se mantiene en la ventana Adicionar usuario.
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	I(ho*/)	V(ok12/*)	
		V(10.1.1.1)	I(123/*)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	I(ho*/)	
EP 1.5.	Cancelar	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Se cancela la operación.

Tabla 39 (Juego de datos a probar para caso de prueba del R13.2)

Caso de prueba para R13.3 Modificar Usuario.

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el módulo Seguridad, en el submódulo Configurar usuarios en la interfaz Usuario.
- El usuario que se desea modificar ha sido adicionado antes al sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Modificar usuario.	Se modifica un usuario del sistema	EP 1.1: Modificar un usuario del sistema.	– Se escoge el usuario que se va a modificar – Se presiona el botón **Modificar**. – Se introducen los datos del usuario que se desea modificar en el formulario. – Se presiona el botón **Aceptar**.
		EP 1.2: Modificar un usuario dejando campos requeridos en blanco.	– Se escoge el usuario que se va a modificar – Se presiona el botón **Modificar**. – Se introducen los datos del usuario que se desea modificar en el formulario dejando al menos un campo requerido en blanco. – Se presiona el botón **Aceptar**.

		EP 1.3: Modificar un usuario entrando errores en los datos.	– Se escoge el usuario que se va a modificar – Se presiona el botón **Modificar**. – Se introducen los datos del usuario que se desea modificar en el formulario introduciendo errores en los datos. – Se presiona el botón **Aceptar**.
		EP 1.4: Cancelar.	– Se escoge el usuario que se va a modificar – Se presiona el botón **Modificar**. – Se introducen o no los datos del usuario que se desea modificar en el formulario. – Se presiona el botón **Cancelar**.

Tabla 40 (Requisitos a probar para caso de prueba del R13.3)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Rango IP	Campo de texto	No	Números
2	Tipo de escritorio	Combobox	No	Lista desplegable escritorios existentes.
3	Dominio	Combobox	No	Lista desplegable dominios existentes.
4	Idioma	Combobox	No	Lista desplegable idiomas existentes.
5	Tema	Combobox	No	Lista desplegable temas existentes.
6	Servidores	Combobox	No	Lista desplegable servidores existentes.
7	Entidad	Campo de texto	No	Lista desplegable entidades existentes.
8	Cargo	Campo de texto	No	Lista desplegable cargos existentes.
9	Área	Campo de texto	No	Lista desplegable areas existentes.

10	Usuario	Campo de texto	No	Letras
11	Contraseña	Campo de texto	No	Letras, números y caracteres.
12	Confirmar Contraseña	Campo de texto	No	Letras, números y caracteres.

Tabla 41 (Descripción de variables para caso de prueba del R13.3)

Juego de datos a probar.

Id del escenario	Escenario	Rango IP	Tipo de escritorio	Dominio	Idioma	Tema	Servidores	Entidad	Cargo	Usuario	Contraseña	Confirmar Contraseña	Respuesta del sistema
EP 1.1.	Modificar un nuevo usuario al sistema.	V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok)	V(ok)	Se adiciona el usuario y se guarda la información adicionada.
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(12)	V(12)	
EP 1.2.	Modificar un usuario dejando campos requeridos en blanco	I(vacio)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	Se muestra el campo de texto en rojo que indica que no se pueden dejar campos en blanco y se mantiene en la ventana Modificar usuario.
		V(10.1.1.1)	I(vacio)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	I(vacio)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	I(vacio)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	I(vacio)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	I(vacio)	V(UCI)	V(Jefe personal)	I(vacio)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	I(vacio)	V(Jefe personal)	V(rosa)	I(vacio)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	I(vacio)	V(rosa)	V(ok12/*)	I(vacio)	

		I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	I(vacio)	Se muestran los campos de textos en rojo que indica que no se pueden dejar campos en blanco y se mantiene la interfaz Modificar funcionalidad.
		V(10.1.1.1)	V(estándar	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	I(/*-)	V(ok12/*)	V(ok12/*)	
		V(10.1.1.1)	V(estándar)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(120)	V(ok12/*)	
		V(10.1.1.1)	I(vacio)	V(uci)	V(español)	V(rojo)	V(uci.cu)	V(UCI)	V(Jefe personal)	V(rosa)	V(ok12/*)	V(ok12/*)	
EP 1.5.	Cancelar	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	Se cancela la operación.

Tabla 42 (Juego de datos a probar para caso de prueba del R13.3)

Caso de prueba para R13.4 Eliminar Usuario.

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el módulo Seguridad, en el submódulo Configurar usuarios en la interfaz Usuario.
- El usuario que se desea eliminar ha sido adicionado antes al sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Eliminar usuario	Se elimina el usuario del sistema.	EP 1.1. Eliminar un usuario del sistema.	– Se escoge el usuario que se va a eliminar. – Se presiona el botón **Eliminar**. – Se presiona el botón **Aceptar**.
		EP 1.2. Cancelar.	– Se escoge el usuario que se va a eliminar. – Se presiona el botón **Eliminar**. – Se presiona el botón **Cancelar**

Tabla 43 (Requisitos a probar para caso de prueba del R13.4)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
NA	NA	NA	NA	NA

Tabla 44 (Descripción de variables para caso de prueba del R13.4)

Juego de datos a probar.

Id del escenario	Escenario	Id inicio	Respuesta del sistema	Resultado de la prueba
EP 1.1	Eliminar un usuario del sistema	NA	Se selecciona el usuario que se desea eliminar, se efectúa la operación y el mismo queda eliminado satisfactoriamente.	
EP 1.2	Cancelar	NA	Se cancela la operación.	

Tabla 45 (Juego de datos a probar para caso de prueba del R13.4)

Caso de prueba para R13.5 Asignar Roles.

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el módulo Seguridad, en el submódulo Configurar usuarios en la interfaz Usuario.
- El usuario al que se le van a regular las acciones debe estar registrado en el sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Asignar roles	Se le asignan o se le quitan permisos al usuario.	EP 1.1: Asignar roles	– Se escoge el usuario al que se le va asignar el rol – Se presiona el botón **Asignar roles**. – Se escoge en el sistema el rol que tiene el usuario. – Se escoge la entidad que se va asignar. – Se presiona el botón **Aceptar**.
		EP 1.3: Cancelar	– Se escoge el usuario al que se le va asignar el rol – Se presiona el botón **Asignar roles**. – Se escoge en el sistema el rol que tiene el usuario. – Se escoge la entidad que se va asignar. – Se presiona el botón **Cancelar.**

Tabla 46 (Requisitos a probar para caso de prueba del R13.5)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	NA	NA	NA	NA

Tabla 47 (Descripción de variables para caso de prueba del R13.5)

Juego de datos a probar.

Id del escenario	Escenario	Usuario	Respuesta del sistema	Resultado de la prueba
EP 1.1	EP 1.1: Asignar roles	NA	Se asigna el rol satisfactoriamente.	
EP 1.3	Cancelar	NA	El sistema cancela la operación.	

Tabla 48 (Juego de datos a probar para caso de prueba del R13.5)

Caso de prueba para R13.6 Cambiar Contraseña.

Condiciones de ejecución.

- Se tienen los permisos necesarios para realizar esta operación.
- El usuario se debe encontrar en el módulo Seguridad, en el submódulo Configurar usuarios en la interfaz Usuarios.
- La contraseña que se desea cambiar debe de ser de un usuario registrado en el sistema.

Requisitos a probar.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1-Cambiar contraseña	Se cambia la contraseña del usuario del sistema.	EP 1.1: Cambiar la contraseña.	-Se presiona el botón **Cambiar contraseña.** -Se introducen los datos para cambiar la contraseña. -Se presiona el botón Aceptar
		EP 1.2: Cambiar la contraseña del usuario dejando campos requeridos en blanco.	Se presiona el botón **Cambiar contraseña.** Se introducen los datos del perfil de usuario que se desea adicionar en el formulario dejando al menos un campo requerido en blanco. - Se presiona el botón **Aceptar.**
		EP 1.3: Cancelar.	- Se presiona el botón **Cambiar contraseña.** – Se introducen o no los datos del perfil de usuario que se desea adicionar en el formulario. - Se presiona el botón **Cancelar.**

Tabla 49 (Requisitos a probar para caso de prueba del R13.6)

Descripción de variables.

No	Nombre de campo	Clasificación	Puede ser nulo	Descripción
1	Usuario	Campo de texto	No	Letras
2	Contraseña anterior	Campo de texto	No	Letras, números y caracteres especiales.
3	Nueva contraseña	Campo de texto	No	Letras, números y caracteres especiales.
4	Confirmar nueva contraseña	Campo de texto	No	Letras, números y caracteres especiales.

Tabla 50 (Descripción de variables para caso de prueba del R13.6)

Juego de datos a probar.

Id del escenario	Escenario	Usuario	Contraseña anterior	Nueva contraseña	Confirmar nueva contraseña	Respuesta del sistema	Resultado de la prueba
EP 1.1	Cambiar la contraseña.	V(seguridad)	V (seguridaderp)	V(erp123/)	V(erp123/)	Se cambia la contraseña del usuario y se guarda la información.	
EP 1.2	Cambiar la contraseña del usuario dejando campos requeridos en blanco.	V(seguridad)	I(vacio)	V(erp123/)	V(erp123/)	Se muestra el campo de texto en rojo que indica que no se pueden dejar campos en blanco y se mantiene en la ventana Cambiar contraseña.	
		V(seguridad)	V (seguridaderp)	I(vacio)	V(erp123/)		

		V(seguridad)	V (seguridaderp)	V(erp123/)	I(vacio)		
EP 1.3	Cancelar	NA	NA	NA	NA	Se cancela la operación.	

Tabla 51 (Juego de datos a probar para caso de prueba del R13.6)

Anexo 3: Instrumento de medición de la métrica Tamaño Operacional de Clase (TOC).

Atributo	Categoria	Criterio
Responsabilidad	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Complejidad implementación	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	<= Prom.

Tabla 52 (Rango de valores de para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC)

Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
DatAccionModel	4	Baja	Baja	Alta
DatFuncionesModel	4	Baja	Baja	Alta
DatServPrestaModel	4	Baja	Baja	Alta
DatFuncionalidadModel	4	Baja	Baja	Alta
DatParametrosModel	4	Baja	Baja	Alta
DatSistemaModel	3	Baja	Baja	Alta
DatSistemaSegUsuarioModel	2	Baja	Baja	Alta
DatServicioDatSistemaModel	3	Baja	Baja	Alta
DatAccion	7	Media	Media	Media
DatFunciones	3	Baja	Baja	Alta
DatFuncionalidad	9	Media	Media	Media
DatParametros	3	Baja	Baja	Alta
DatSistema	12	Alta	Alta	Baja
DatSistemaSegRolDatFuncionalidadDatAccion	6	Media	Media	Media
DatSistemaDatServidores	5	Media	Media	Media
DatServicio	4	Baja	Baja	Alta
DatServicioDatSistema	5	Media	Media	Media
DatSistemaSegRol	6	Media	Media	Media
DatSistemaSegRolDatFuncionalidad	6	Media	Media	Media
DatSerautenticacionModel	2	Baja	Baja	Alta
DatServidorDatGestorDatBdModel	2	Baja	Baja	Alta
DatBdModel	4	Baja	Baja	Alta
DatGestorDatServidorbdModel	2	Baja	Baja	Alta
DatServidorModel	5	Media	Media	Media
DatEsquemaModel	4	Baja	Baja	Alta
DatGestorModel	4	Baja	Baja	Alta
DatServidorbdModel	2	Baja	Baja	Alta

DatServidorDatGestorDatBdDatEsquemaModel	2	Baja	Baja	Alta
DatEsquema	10	Media	Media	Media
DatGestorDatServidorbd	4	Baja	Baja	Alta
DatServidorDatGestorDatBdDatEsquema	5	Media	Media	Media
DatServidor	7	Media	Media	Media
DatBd	10	Media	Media	Media
DatSerautenticacion	1	Baja	Baja	Alta
DatServidorbd	2	Baja	Baja	Alta
DatGestor	12	Alta	Alta	Baja
DatServidorDatGestorDatBd	4	Baja	Baja	Alta
SegRestricclaveaccesoModel	4	Baja	Baja	Alta
SegRolModel	5	Media	Media	Media
SegCertificadoModel	4	Baja	Baja	Alta
SegUsuarioModel	7	Media	Media	Media
SegDominioModel	4	Baja	Baja	Alta
SegUsuarioSegRolModel	2	Baja	Baja	Alta
SegRestricclaveacceso	3	Baja	Baja	Alta
SegCertificado	4	Baja	Baja	Alta
SegRol	6	Media	Media	Media
SegClaveacceso	1	Baja	Baja	Alta
SegUsuario	21	Alta	Alta	Baja
SegDominio	7	Media	Media	Media
SegUsuarioDatSerautenticacion	2	Baja	Baja	Alta
NomIdiomaModel	4	Baja	Baja	Alta
NomCampoModel	4	Baja	Baja	Alta
NomTemaModel	4	Baja	Baja	Alta
NomDesktopModel	4	Baja	Baja	Alta
NomExpresionesModel	4	Baja	Baja	Alta
NomExpresiones	3	Baja	Baja	Alta
NomValor	6	Media	Media	Media
NomFila	3	Baja	Baja	Alta
NomCampo	4	Baja	Baja	Alta
NomIdioma	4	Baja	Baja	Alta
NomDesktop	4	Baja	Baja	Alta
NomTema	4	Baja	Baja	Alta

Tabla 53 (Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización))

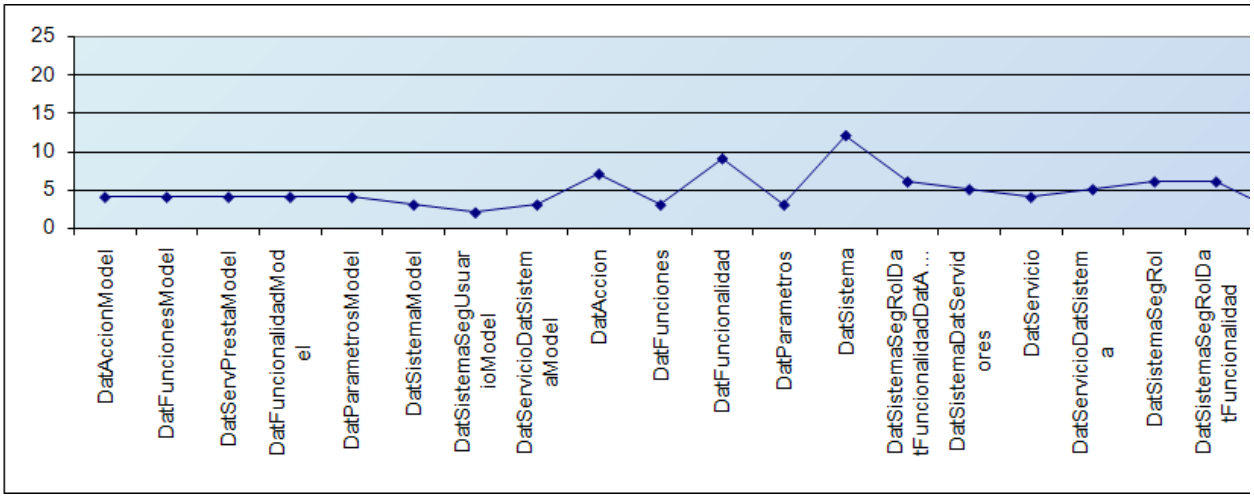

Figura 47 (Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización), parte 1)

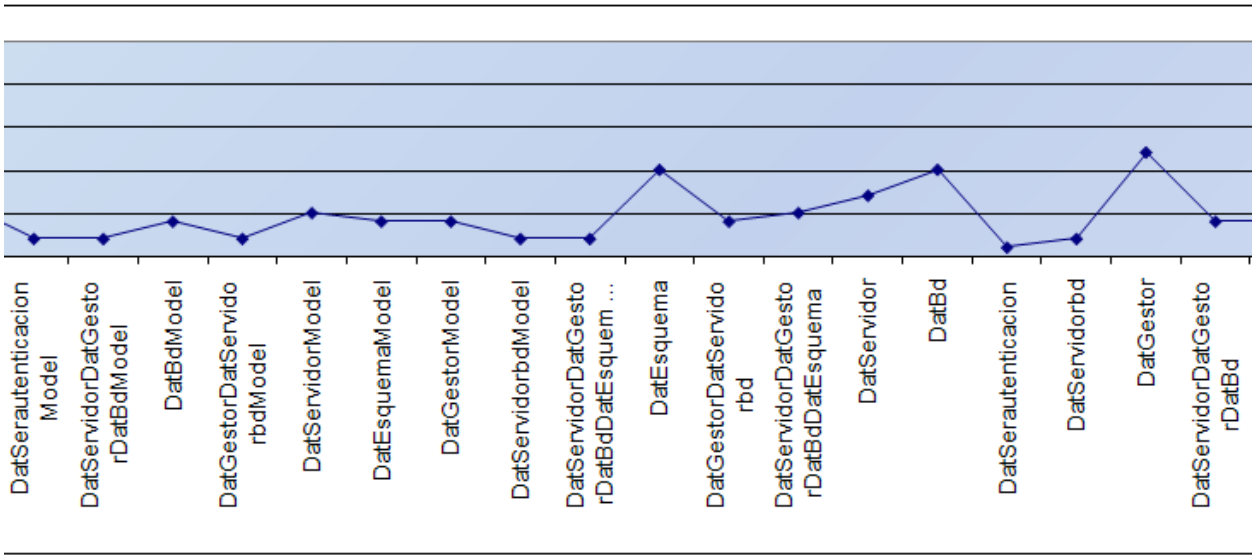

Figura 48 (Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización), parte 2)

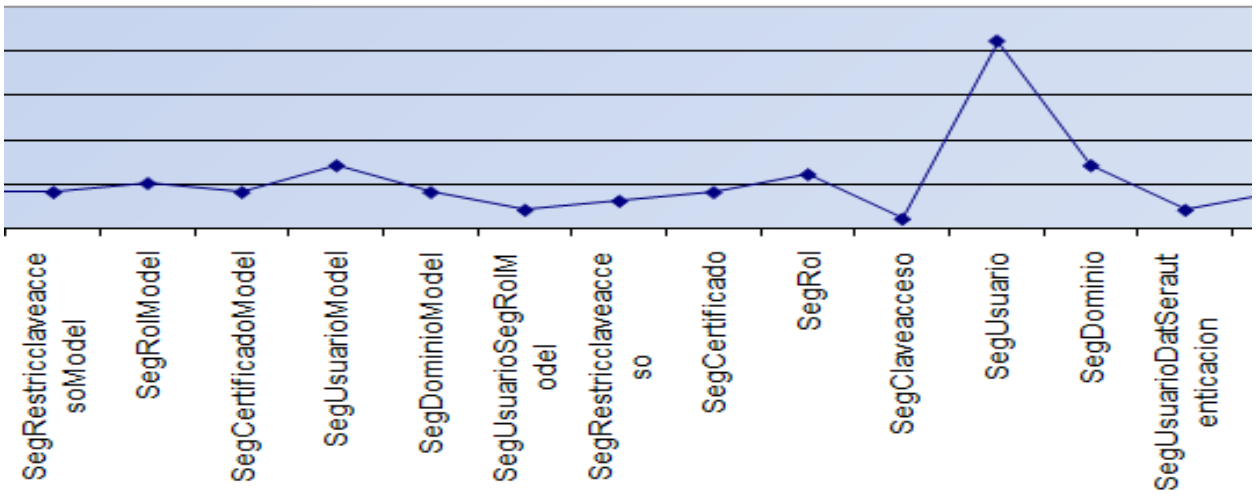

Figura 49 (Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización), parte 3)

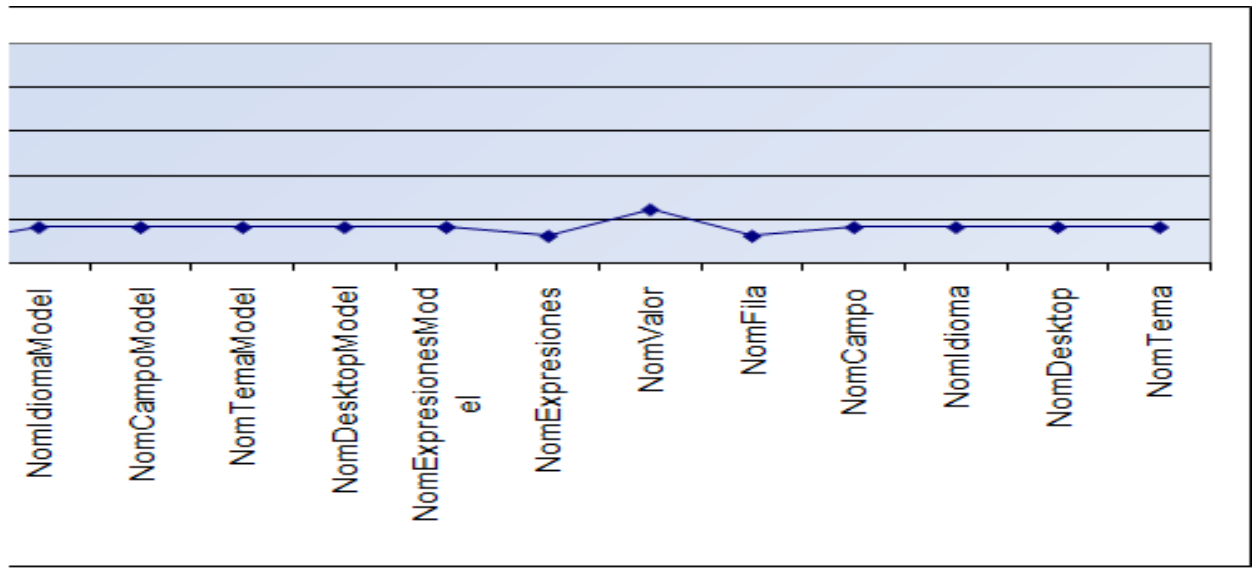

Figura 50 (Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización), parte 3)

Anexo 4: Instrumento de medición de la métrica Relaciones entre Clases (RC).

Atributos	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad Mant.	Baja	<= Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	> 2*Prom.
Reutilización	Baja	>2* Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	<= Prom.
Cantidad de Pruebas	Baja	<= Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	> 2*Prom.

Tabla 54 (Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC.)

Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización	Cantidad de Pruebas
DatAccionModel	1	Bajo	Baja	Alta	Baja
DatFuncionesModel	1	Bajo	Baja	Alta	Baja
DatServPrestaModel	1	Bajo	Baja	Alta	Baja
DatFuncionalidadModel	1	Bajo	Baja	Alta	Baja
DatParametrosModel	1	Bajo	Baja	Alta	Baja
DatSistemaModel	1	Bajo	Baja	Alta	Baja
DatSistemaSegUsuarioModel	1	Bajo	Baja	Alta	Baja
DatServicioDatSistemaModel	1	Bajo	Baja	Alta	Baja
DatAccion	2	Medio	Media	Media	Media
DatFunciones	1	Bajo	Baja	Alta	Baja
DatFuncionalidad	3	Alto	Media	Media	Media
DatParametros	2	Medio	Media	Media	Media
DatSistema	2	Medio	Media	Media	Media
DatSistemaSegRolDatFuncionalidadDatAccion	2	Medio	Media	Media	Media
DatSistemaDatServidores	6	Alto	Alta	Baja	Alta
DatServicio	4	Alto	Alta	Baja	Alta
DatServicioDatSistema	3	Alto	Media	Media	Media
DatSistemaSegRol	2	Medio	Media	Media	Media
DatSistemaSegRolDatFuncionalidad	2	Medio	Media	Media	Media
DatSerautenticacionModel	1	Bajo	Baja	Alta	Baja
DatServidorDatGestorDatBdModel	1	Bajo	Baja	Alta	Baja

DatBdModel	1	Bajo	Baja	Alta	Baja
DatGestorDatServidorbdModel	1	Bajo	Baja	Alta	Baja
DatServidorModel	1	Bajo	Baja	Alta	Baja
DatEsquemaModel	1	Bajo	Baja	Alta	Baja
DatGestorModel	1	Bajo	Baja	Alta	Baja
DatServidorbdModel	1	Bajo	Baja	Alta	Baja
DatServidorDatGestorDatBdDatEsquemaModel	1	Bajo	Baja	Alta	Baja
DatEsquema	3	Alto	Media	Media	Media
DatGestorDatServidorbd	2	Medio	Media	Media	Media
DatServidorDatGestorDatBdDatEsquema	1	Bajo	Baja	Alta	Baja
DatServidor	3	Alto	Media	Media	Media
DatBd	3	Alto	Media	Media	Media
DatSerautenticacion	1	Bajo	Baja	Alta	Baja
DatServidorbd	2	Medio	Media	Media	Media
DatGestor	4	Alto	Alta	Baja	Alta
DatServidorDatGestorDatBd	1	Bajo	Baja	Alta	Baja
SegRestricclaveaccesoModel	1	Bajo	Baja	Alta	Baja
SegRolModel	1	Bajo	Baja	Alta	Baja
SegCertificadoModel	1	Bajo	Baja	Alta	Baja
SegUsuarioModel	1	Bajo	Baja	Alta	Baja
SegDominioModel	1	Bajo	Baja	Alta	Baja
SegUsuarioSegRolModel	1	Bajo	Baja	Alta	Baja
SegRestricclaveacceso	1	Bajo	Baja	Alta	Baja
SegCertificado	1	Bajo	Baja	Alta	Baja
SegRol	2	Medio	Media	Media	Media
SegClaveacceso	1	Bajo	Baja	Alta	Baja
SegUsuario	10	Alto	Alta	Baja	Alta
SegDominio	1	Bajo	Baja	Alta	Baja
SegUsuarioDatSerautenticacion	1	Bajo	Baja	Alta	Baja
NomIdiomaModel	1	Bajo	Baja	Alta	Baja
NomCampoModel	1	Bajo	Baja	Alta	Baja
NomTemaModel	1	Bajo	Baja	Alta	Baja
NomDesktopModel	1	Bajo	Baja	Alta	Baja
NomExpresionesModel	1	Bajo	Baja	Alta	Baja
NomExpresiones	1	Bajo	Baja	Alta	Baja
NomValor	1	Bajo	Baja	Alta	Baja
NomFila	1	Bajo	Baja	Alta	Baja
NomCampo	4	Alto	Alta	Baja	Alta
NomIdioma	1	Bajo	Baja	Alta	Baja
NomDesktop	1	Bajo	Baja	Alta	Baja
NomTema	1	Bajo	Baja	Alta	Baja

Tabla 55 (Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas))

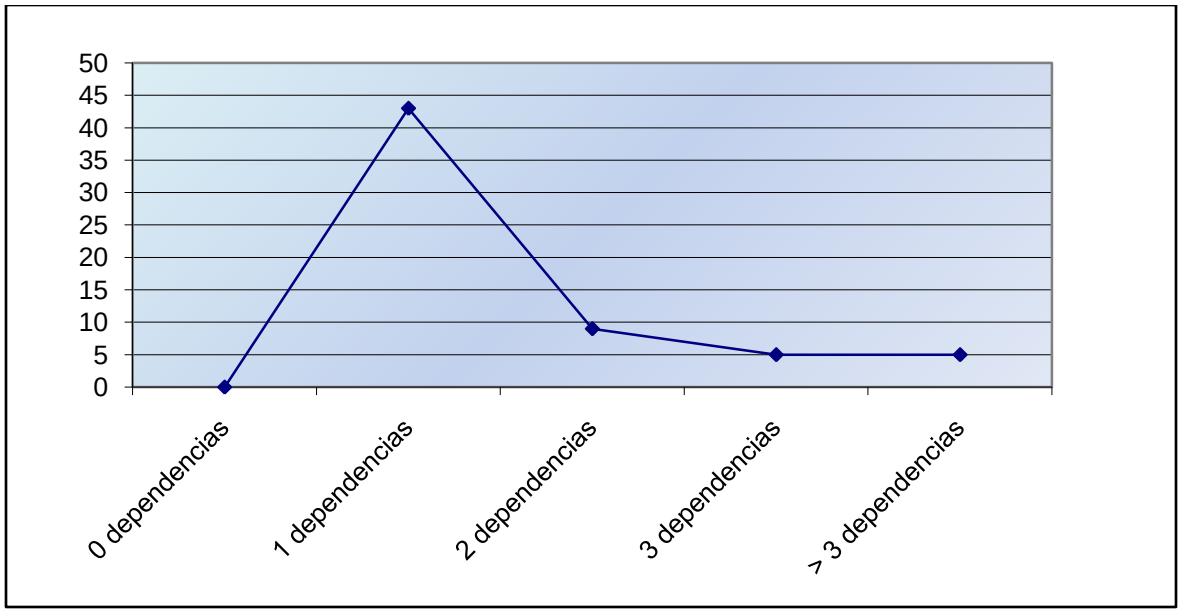

Figura 51 (Gráfica de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores)