UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

COMPONENTE PARA LA ADMINISTRACIÓN Y CONFIGURACIÓN DEL CONTROL DE ACCESO EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

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

Autores

Yaciel Mendoza Durán

Redecto Rodríguez Castillo

Tutores

Ing. Ronaldo Castro Milán

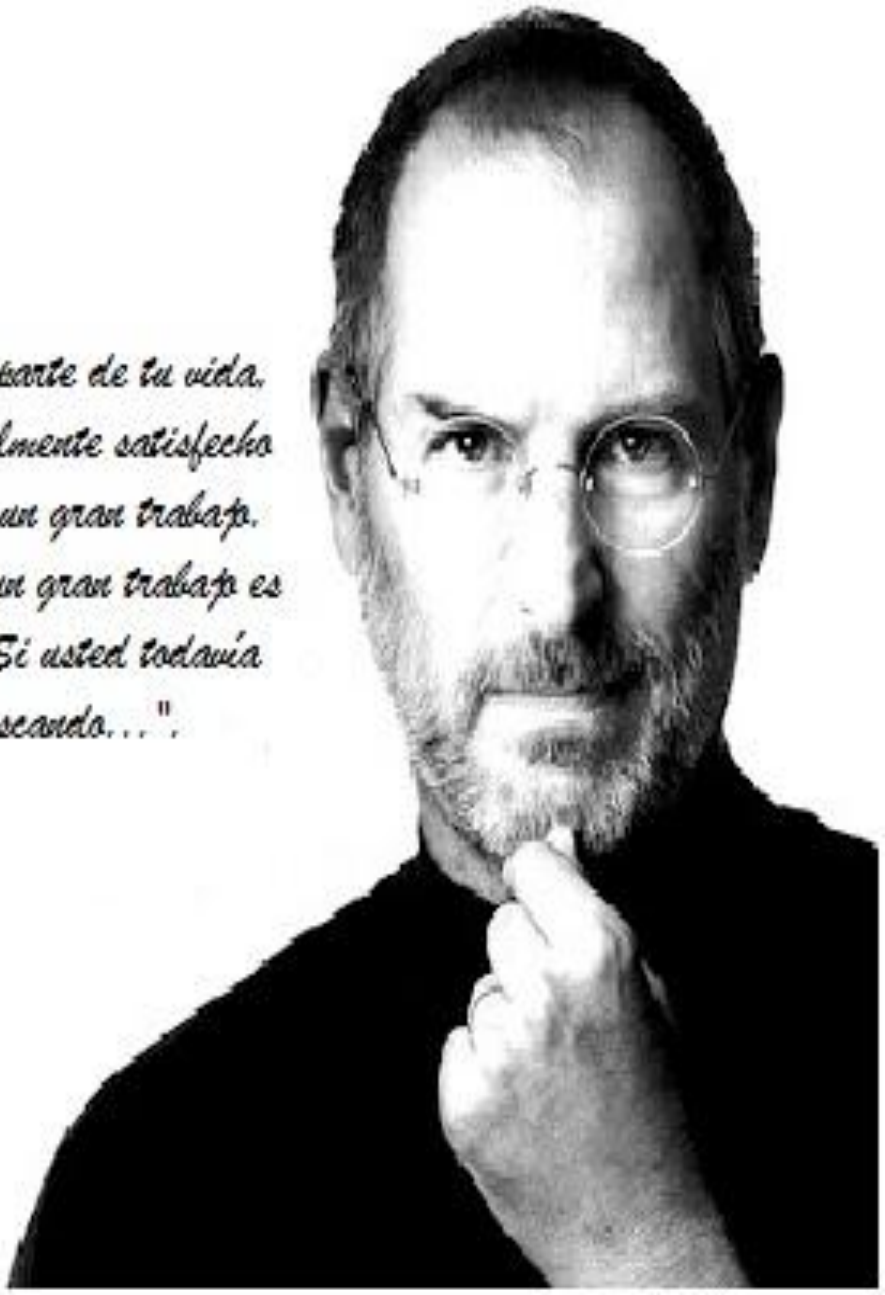
Ing. Yeneidys Alvarez González

LA HABANA, 24 DE JUNIO DE 2013

PENSAMIENTO

"El trabajo va a llenar gran parte de tu vida, y la única forma de estar realmente satisfecho es hacer aquello que creen es un gran trabajo. Y la única manera de hacer un gran trabajo es hacer lo que te gusta hacer. Si usted todavía no lo ha encontrado, sigan buscando...".

Steve Jobs



DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los 24 días del mes de junio del año 2013.

Yaciel Mendoza Durán.

_____

Firma de los Autores

Redecto E. Rodríguez Castillo.

_____

Firma de los Autores

Ing. Ronaldo Castro Milán.

_____

Firma de los Tutores

Ing. Yeneidys Alvarez González.

_____

Firma de los Tutores

DATOS DEL CONTACTO

Autor: Yaciel Mendoza Durán.
Correo electrónico: ymduran@estudiantes.uci.cu

Autor: Redecto E. Rodríguez Castillo.
Correo electrónico: rerodriguez@estudiantes.uci.cu

Tutor: Ronaldo Castro Milán.
Especialidad de graduación: Ciencias Informáticas
Categoría docente: Instructor
Correo electrónico: rcastro@uci.cu

Tutor: Yeneidys Alvarez González.
Especialidad de graduación: Ciencias Informáticas
Categoría docente: no Instructor
Correo electrónico: yalvarezg@uci.cu

AGRADECIMIENTOS

Quisiera agradecer a cada una de las personas que de una forma u otra han influido en mi formación y han estado a mi lado hasta convertirme en lo que hoy soy. A todos aquellos amigos de la infancia que no olvido, a los que vinieron después y se convirtieron en parte importante de mi vida. A mi piquete de la FEEM con el que aprendí a sentir que un grupo unido era capaz de lograr cualquier cosa. A mi gente del Pre que hasta hoy conservan los mejores recuerdos con los que cuento, a José Carlos, Annalay, José Alexis, Yaima, Nachiel, Maylen, Diana y a los demás, les agradezco su entera amistad y confianza. A la gente de Trinidad que compartieron y comparten conmigo en esta universidad, así como a mi grupo y en especial a aquellos que se convirtieron casi en parte de mi familia. A mi hermano Rede, que juntos hemos logrado enfrentar y vencer todas las batallas que esta vida de estudiante nos ha impuesto, a ti hermano por estar de mi lado siempre. A todas mis amistades, gracias.

Gracias a toda mi familia, tanto materna como paterna, a mis tías y tíos, a mis primas y primos, por ser la mejor familia que hubiese podido desear, por acompañarme, por preocuparse, por orientarme, apoyarme, por hacer de mí el Yaciel que soy ahora. Gracias a mis hermanos, Yoán, Kary y Yele, por hacerme sentir la persona más especial de este mundo. Gracias a mi abuela del alma, por ser mí ejemplo de sacrificio, de entrega, de coraje, gracias por ser tú y no otra, gracias por hacer de nuestra familia lo que somos hoy. A mi Adrianita, la personita con el corazón más lindo y cariñoso que he conocido, gracias por tomar mi mano e invitarme a compartir todo contigo, gracias por apoyarme, por quererme, por amarme, gracias por hacer que te quiera hoy, mañana y siempre 3MSC. Muchas gracias a las personas más importantes de mi vida, gracias a mi padre, a Miguel Mendoza, al hombre que me hizo y me dio el apellido por el cual me conoce la mayoría de la gente, al que me demostró lo que es ser un ejemplo de padre a pesar de la distancia y que cualquier sacrificio es insignificante siempre que sea por un hijo, a ese, mi superhéroe, le agradezco la vida; por otro lado mi madre, la mejor madre del mundo, la mujer más capaz, emprendedora y sacrificada que haya conocido, a ti mami te debo la vida y todo lo que soy, gracias por tu amor, sabiduría, comprensión y no acabaría de agradecer jamás todo lo que has hecho por mí, solo deseo tenerte a mi lado por siempre y poder dedicarte todo lo que consiga en mi vida.

Yaciel Mendoza Durán

Agradezco a cada una de las personas que de una forma u otra contribuyeron en mi formación y han estado a mi lado hasta convertirme en lo que hoy soy. A todos aquellos amigos de la infancia que no olvido, a los que vinieron después y se convirtieron en parte importante de mi vida. A mi gente de la Primaria que se formaron desde pequeño junto conmigo los cuales son imposible de olvidar ya que son con los que se comienza a florecer tu árbol de la vida, a mi profesora de esa enseñanza que hace poco lloraba de alegría cuando se enteró que dos alumnos de ella terminaban su carrera como empezaron "juntos", también a mis grandes amigo del barrio que siempre esperaban ansioso mi llegada de la UCI para conversar y tomar mucho ron, que me permitieron saber que las amistades siempre están hay cuando más lo necesitas, a Matute, Ballester, Machete y Ronnei (El poso) a todos ello les agradezco su entera amistad y confianza . A la gente de Trinidad que compartieron y comparten conmigo en esta universidad, así como a mi grupo y en especial a aquellos que se convirtieron casi en parte de mi familia. Y qué decir de mi gran hermano, azabache y manager Yaciel (El sasa), gracias hermano por juntos haber logrado enfrentar y vencer todas las batallas que esta vida de estudiante nos ha impuesto, por estar de mi lado siempre y guiarme siempre en ese camino lleno de dificultades a salir sin mirar a atrás, por esa grata compañía que siempre me brindaste a lo largo de 17 años ñoooo… es casi imposible pero real siempre junto y en la misma mesa desde 1er grado, de verdad muchísimas… gracias. A todas mis amistades, gracias.

Gracias a toda mi familia, tanto materna como paterna, a mis tías y tíos Acela, Marta, Carmen, Maricela, Papo, Yoel a todos ustedes por quererme tanto como si fuera su hijo, a mis primas y primos, por ser la mejor familia que hubiese podido desear, por acompañarme, por preocuparse, por orientarme, apoyarme, por hacer de mí el que soy ahora. Gracias a mis hermanos, Roxana y Javier y Diego, por hacerme sentir la persona más especial de este mundo. Gracias a mis abuelas Petronila y Oristela, por ser mí ejemplo de sacrificio y de entrega, por luchar conmigo desde pequeño para que nada me faltara, gracias por hacer de nuestra familia la más acogedora familia del mundo. A mi LULU (Elisa), la personita con el corazón más lindo y cariñoso que he conocido, gracias por tomar mi mano e invitarme a compartir todo contigo, gracias por apoyarme en los momentos más difíciles de mi vida, por quererme, por amarme, gracias por hacer que te quiera hoy y que sienta mi vida realizada contigo hasta que la muerte nos separe sin más palabras te amare por siempre solo por existir. Muchas gracias a las personas más importantes de mi vida, a mi padre, Andrés Rguez, al hombre que me hizo y supo con firmeza hacer de mí lo que hoy soy , al que me demostró lo que es ser un ejemplo de padre y que cualquier sacrificio es insignificante siempre que sea por un hijo, a ese, mi superhéroe, le agradezco la vida; por otro lado mi madre, la mejor madre del mundo, la mujer más capaz, emprendedora y sacrificada que haya conocido, a ti mami te debo la vida y todo lo que soy, gracias por tu amor, sabiduría, comprensión y no acabaría de agradecer jamás todo lo que has hecho por mí, solo deseo tenerte a mi lado por siempre y poder dedicarte todo lo que consiga en mi vida.

Redecto E. Rodríguez Castillo

DEDICATORIA

Dedico el presente trabajo de diploma a mi abuela, mi padre y mi queridísima madre, por ser mi principal apoyo en la construcción de este sueño.

Yaciel Mendoza Durán

Dedico el presente trabajo de diploma a mis grandes tesoros, que siempre confiaron en mí, me apoyaron sin condición y que sin ellos no lo pudiera haber logrado, mi mama, mi papa y mi linda hermana Roxi, mis abuelas y al amor de mi vida Elisa Gamez López por ser mi principal apoyo en la construcción de este sueño.

Redecto E. Rodríguez Castillo

RESUMEN

Los sistemas de control de acceso son de vital importancia para garantizar la seguridad y preservar los bienes y recursos de cada organización. La Universidad de las Ciencias Informáticas (UCI) es unos de los centros de altos estudios con mayor flujo de personal diario, lo cual representa un riesgo para los medios e instalaciones que la misma posee.

En el Centro de Identificación y Seguridad Digital (CISED) se está llevando a cabo el desarrollo de una Plataforma Modular de Identificación y Control de Acceso (PMICA) con el objetivo de automatizar los procesos de identificación así como la gestión y supervisión centralizada del acceso del personal a una institución.

Surge así la necesidad de desarrollar un componente para la administración y configuración del control de acceso que permita administrar de manera centralizada el acceso a las instalaciones que requieran un mayor nivel de seguridad.

El presente trabajo parte de un estudio realizado de los principales sistemas y modelos de control de acceso, así como la evaluación de los últimos con el objetivo de elegir el adecuado para la implementación de dicho componente. Se incluyen además elementos del análisis y diseño, donde sus resultados servirán de base para la fase de implementación.

Palabras clave: modelo de control de acceso, sistema de control de acceso, rbac, asp.net mvc4.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURA

ÍNDICE DE TABLA

INTRODUCCIÓN

Históricamente la humanidad, en el afán de contar con una mejor organización y cuidado sobre los medios que posee, ha implementado diversos mecanismos para controlar el acceso de las personas tanto a instalaciones como a recursos. Con la llegada de las nuevas tecnologías de la informática y las comunicaciones (NTIC), el control de acceso se ve considerablemente beneficiado gracias a la creación de soluciones que permiten la automatización de dicho proceso de gestión.

En organizaciones donde circulan diariamente un gran cúmulo de personas, resulta complicado mantener un control estricto del flujo de las mismas tanto en las salidas como las entradas a las diferentes áreas o locales de la empresa. Esta situación trae como consecuencia un déficit en la seguridad de la organización por lo que la implantación de un sistema automatizado que permita la gestión de cada una de las políticas de acceso se convierte en medida imprescindible para resguardar la misma.

A partir del proceso de informatización que se desarrolla en Cuba, varias instituciones nacionales han requerido la implantación de sistemas de control de acceso. La Universidad de las Ciencias Informáticas (UCI), es uno de los centros de altos estudios con mayor circulación de personal diario, lo que pone en riesgo la seguridad de la información así como los recursos e instalaciones de la misma.

Actualmente la UCI cuenta con un departamento destinado a la defensa, protección y seguridad de cada uno de los medios, áreas y el personal que radica en ella. Dicho departamento es el encargado de controlar el acceso del personal al interior de la universidad. Este proceso se lleva a cabo a través de la verificación visual de una identificación que tiene carácter personal e intransferible. Por lo que dicho proceso está expuesto a errores humanos debido a que no existe una fuente que certifique que la credencial es válida y que quien la posee es realmente el propietario. Para el chequeo de la identificación no se tiene en cuenta su fecha de vencimiento, puesto que dicho documento no especifica cuando deja de ser admitido. En ocasiones no se observa correctamente la fotografía impresa, ya que la misma cuenta con cierto grado de deterioro; la aglomeración de personal esperando acceder es otra de las causas que impide que se le preste la atención adecuada a la identificación. Producto a que los materiales para la emisión de la identificación no son siempre los mismos, existen algunas con más calidad que otras así como diversos niveles de falsificación, lo que dificulta aún más el trabajo de los agentes de seguridad.

Una de las principales deficiencias del proceso de control de acceso en la UCI está dada por el lento desarrollo al autorizar la entrada de un visitante, ya que tanto la búsqueda en el libro de visitas como la toma de sus datos se realiza de manera manual, lo cual se incrementa al no aparecer en dicho libro y se hace necesario llamar a determinada persona que autorice el acceso.

Por otro lado, dentro de la universidad existen sistemas encargados de controlar el acceso a un grupo de lugares que requieren ciertos niveles de seguridad; como ejemplo se tienen los complejos comedores y

algunos de los laboratorios de producción. Sin embargo se carece de una solución genérica y centralizada que satisfaga todas las necesidades de la institución en lo que respecta a la protección de recursos e información y que a su vez permita realizar dicha labor de forma automática.

Debido a que la protección de los recursos en una institución es un problema latente, la creación de un sistema de control de acceso que permita adaptarse a las condiciones propias de cada empresa con el menor costo posible, resulta una solución factible para las entidades encargadas de ofrecer servicios de seguridad. Por tal motivo en el Centro de Identificación y Seguridad Digital (CISED), se está desarrollando una Plataforma Modular de Identificación y Control de Acceso (PMICA), que permitirá de manera eficaz la gestión de acceso del personal a los diferentes recursos e instalaciones de cada organización.

Para la correcta implementación de esta solución se requiere de un subsistema de control de acceso que sea capaz de gestionar las políticas de acceso dentro de la organización, permitiéndole a la PMICA controlar de manera centralizada quién puede acceder a qué recurso y bajo qué circunstancias.

Teniendo en cuenta la situación expuesta anteriormente, se ha determinado como **problema de la investigación**: ¿Cómo garantizar el cumplimiento de las políticas de control de acceso en la Universidad de las Ciencias Informáticas?

Constituyendo el **objeto de estudio** los procesos de control de acceso.

Para dar respuesta al problema planteado se define como **objetivo general**: Desarrollar un componente para la administración y configuración del control de acceso en la Universidad de las Ciencias Informáticas.

Enmarcado en el **campo de acción**, el proceso de control de acceso en la Universidad de las Ciencias Informáticas.

Con el fin de alcanzar el objetivo general planteado se proponen los siguientes **objetivos específicos**:

- Analizar estándares, metodologías y soluciones existentes relacionadas a los sistemas de control de acceso.
- Realizar análisis y diseño de la solución propuesta.
- Realizar implementación del componente de Administración y Configuración.
- Validar la solución propuesta a través de pruebas de caja negra y caja blanca.

Para lograr el correcto cumplimiento de los objetivos se trazaron las siguientes **tareas de la investigación**:

1. Análisis y selección del modelo de control de acceso a utilizar.
2. Análisis y valoración de los sistemas de control de accesos más relevantes.
3. Realización del levantamiento del negocio.
4. Realización del modelo de dominio.
5. Análisis de los requerimientos del software.

6. Descripción de los requisitos del software.
7. Confección de los diagramas de clases del diseño.
8. Implementación del componente.
9. Realización de las pruebas de unidad y de caja negra al componente.
10. Corrección de errores a partir de las no conformidades.

Para dar cumplimiento al objetivo propuesto se han combinado diferentes métodos y procedimientos teóricos y empíricos de la investigación científica, en la búsqueda y procesamiento de la información. Los fundamentales son:

Métodos teóricos: El método ***hipotético-deductivo*** para proponer líneas de trabajo a partir de resultados parciales obtenidos durante el proceso de investigación. El método ***analítico-sintético*** al descomponer el problema de la investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta; el método ***inducción-deducción*** como vía de la constatación teórica durante el desarrollo de la tesis.

Métodos empíricos: Se utilizó el método de la ***observación*** durante el estudio del funcionamiento de los procesos de control de acceso que se desarrollan en la Universidad de las Ciencias Informáticas a partir de una planificación previa donde se precisaron los elementos que serían objeto de análisis. El método ***experimental*** se utilizó en la comprobación de la utilidad de los resultados obtenidos a partir de la introducción de datos ficticios.

PRINCIPALES APORTES

La presente investigación aporta un componente altamente flexible y escalable, para gestionar de manera sencilla y económica las diversas políticas de control de acceso surgidas en el marco de una institución. Por otro lado el estudio realizado recoge un análisis crítico de los principales conceptos y tendencias asociadas a los sistemas de control de acceso así como su aplicación en la actualidad. Independientemente del software se pone a disposición de quien lo necesite un grupo de artefactos que pueden servir de referencias para próximas investigaciones como son: modelo de dominio, catálogo de requisitos, descripción de requisitos, arquitectura de software, diagramas de clase y estándares de codificación.

ESTRUCTURA DEL CONTENIDO DEL PRESENTE TRABAJO DE DIPLOMA

Capítulo 1: Fundamentación Teórica: Este capítulo contiene una base teórica para entender el problema a solucionar, incluye un estado del arte del control de acceso a nivel internacional, nacional y de la Universidad de las Ciencia Informáticas, además se exponen las tendencias, tecnologías, metodologías y software utilizados en la actualidad; los cuales sirven de base para el desarrollo de la aplicación.

Capítulo 2: Propuesta de Solución: En este capítulo se describe el flujo actual de los procesos de control de acceso en la Universidad de las Ciencias Informáticas profundizando en los que serán objeto de optimización. Se presenta además el modelo de dominio que ilustrará las relaciones entre los principales conceptos del sistema. Por último se lleva a cabo la descripción de la solución propuesta a partir de la definición previa de los requisitos funcionales y no funcionales del sistema.

Capítulo 3: Implementación y Prueba: Este capítulo abarca la fase de implementación y pruebas, comenzando a partir de los resultados obtenidos del modelo de diseño y continúa con la implementación del sistema concluyendo con las pruebas unitarias y de caja negra en pos de garantizar la calidad de la solución. Se obtendrán también los diferentes diagramas que propone la metodología de desarrollo en dicha fase, para facilitar el desarrollo de la aplicación. Se describe además el modelo de datos y los estándares de codificación con el objetivo de lograr una mejor comprensión del código por parte del equipo de desarrollo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

INTRODUCCIÓN

En el desarrollo de las tecnologías de la informática y las comunicaciones (TIC), los sistemas de control de acceso se han visto especialmente beneficiados a partir de que incrementan de manera exponencial los niveles de seguridad y reducen los gastos en recursos humanos. El mundo y su constante transformación sumada al dinamismo que persiste en él, comienza a exigir de sistemas más sofisticados, adaptables y potentes que sean capaces de proteger determinados recursos con un alto grado de seguridad, sin implicar cambios trascendentales en las organizaciones y que funcionen bajo las condiciones más extremas. A nivel mundial existen numerosas soluciones de este tipo, y para el desarrollo del componente propuesto, es necesario llevar a cabo un estudio de los fundamentos teóricos generales relacionados con los sistemas de control de acceso, que permitan una mejor comprensión del negocio en cuestión, así como un análisis de las principales tendencias, tecnologías, herramientas y metodologías más convenientes para ser aplicadas en la presente investigación.

SOLUCIÓN CONCEPTUAL

Los ***sistemas de control de acceso*** es el software encargado de gestionar la entrada y salida de personas o medios a un lugar determinado, registrándose los datos necesarios para garantizar el cumplimiento de una política de permiso que afiance la seguridad del lugar [1]. Estos sistemas se vuelven indispensables debido a que representan la primera barrera de defensa contra personas no deseadas, que pudieran causar algún tipo de perjuicio dentro de la institución.

Para lograr una correcta implementación de los sistemas de control de acceso, se deben tener presentes tres conceptos fundamentales que normalmente suelen mezclarse de manera difusa. Uno de ellos es la ***identificación***, la cual se refiere a la acción por parte de un usuario de presentar su identidad ante un sistema [2]. Se tiene también la ***autenticación***, descrita como el proceso en el que se realiza la verificación de que el usuario que trata de identificarse es válido [3]. Y por último, la ***autorización*** se encargaría de determinar si el usuario previamente identificado y autenticado tiene permisos para acceder a un recurso [4].

Sin embargo la aplicación de los conceptos anteriormente mencionados es intrascendente si no están encaminados a cumplir con una política determinada. Dicha **política** representa el mecanismo mediante el cual una institución regula el correcto cumplimiento de los procesos que se desarrollan en la misma. Para el caso específico del control de acceso, las políticas se refieren a las reglas que se establecen para poder acceder o no a un recurso, es decir, determinan qué individuo puede acceder a determinado recurso y

bajo qué circunstancias se le debe conceder dicho acceso [5]. Para el escenario particular de una universidad, una política podría ser: *se permite la entrada al área de laboratorios, a todos los trabajadores del proyecto X, en los horarios de 8:00 am a 5:00 pm*. De esta manera el personal que coincida con esas características se le concederá el acceso si la solicitud es realizada en el horario especificado.

Con la combinación de los procesos anteriormente mencionados se puede garantizar el control del acceso a los más diversos medios e instalaciones de una organización. De esta manera queda definido el ***control de acceso*** como el conjunto de políticas que definen las normas en todo lo que a un usuario se le permite hacer en un ámbito determinado. Para establecer dichas políticas se hace necesaria la utilización de un modelo de control de acceso que permita especificar cómo se otorgan los permisos y así llevar a cabo el proceso de autorización. A continuación se exponen las características, ventajas y desventajas de los principales modelos de control de acceso existentes en la actualidad.

Control de Acceso Discrecional

El modelo de control de acceso discrecional (*Discretionary Access Control*, DAC), también llamado modelo de seguridad limitada, es un modelo no orientado al control del flujo de información. Todos los sujetos y objetos en el sistema son controlados y se especifican reglas de autorización de acceso para cada sujeto y objeto. Los sujetos pueden ser usuarios, grupos o procesos. Los modelos DAC están basados en la idea de que el propietario de un objeto, su autor, tiene el control sobre los permisos del mismo, así como autorizar la utilización de este por otros usuarios [6]. Como se puede apreciar este modelo está enfocado fundamentalmente a entornos computacionales donde los permisos actúan sobre objetos de carácter digital, dígase archivos, directorios, entre otros.

Control de Acceso Obligatorio

En el modelo de control de acceso obligatorio (*Mandatory Access Control*, MAC) todos los sujetos y objetos son clasificados basándose en niveles predefinidos de seguridad que son usados en el proceso de obtención de los permisos de acceso. Para describir estos niveles de seguridad todos los sujetos y objetos son marcados con etiquetas de seguridad que siguen el modelo de clasificación de la información militar (desde “desclasificado” hasta “alto secreto”), formando lo que se conoce como política de seguridad multinivel. A diferencia de DAC, los modelos MAC proporcionan mecanismos más sólidos para la protección de datos, y tratan con requerimientos de seguridad más específicos, así como, los requerimientos derivados de las políticas de control de los flujos de información. Existen varias implementaciones de este modelo, como el modelo de Bell-LaPadula y el de Biba (Bell) pero en todos los casos el aseguramiento de las políticas es un tema engorroso puesto que una vez que se establecen los niveles de seguridad es muy difícil la asignación de permisos garantizando que se cumpla siempre la jerarquía en el acceso inherente a este método [6].

Listas de Control Acceso

Las listas de control de acceso (*Access Control List*, ACL) son la forma más antigua y la más básica de control de acceso. Estas definen que cada recurso en un sistema cuyo acceso debe ser controlado, referido como un objeto, tiene su propia lista asociada de asignaciones entre el conjunto de entidades que solicitan acceso a estos y el conjunto de acciones que cada entidad puede tener en el recurso. Las ACL son frecuentes en todos los sistemas operativos modernos, por lo que cada organización que hace uso de un sistema operativo es casi seguro que tiene una implementación de ACL por defecto. Sin embargo aunque ampliamente utilizada, ACL tiene sus limitaciones. Las ACL pueden ser difícil de gestionar en un entorno empresarial donde muchas personas necesitan tener diferentes niveles de acceso a distintos recursos. Selectivamente la adición, eliminación y modificación de las ACL en archivos individuales o incluso grupos de archivos, puede llevar mucho tiempo y puede estar propenso a errores [6].

Control de Acceso Basado en Tareas

El control de acceso basado en tareas (*Task Based Access Control*, TBAC) permite controlar el acceso en entornos representados por flujos de trabajo. El modelo TBAC extiende los tradicionales modelos de control de acceso basados en sujetos/objetos incluyendo aspectos que aportan información contextual basada en las actividades o tareas. Este es garantizado por medio de “etapas de autorización”. Las “etapas de autorización” son un concepto abstracto introducido por TBAC para modelar y manejar un sistema de permisos relacionados con el progreso de las tareas o actividades dentro del contexto de un flujo de trabajo.

De esta manera, a la hora de aplicar TBAC en un entorno empresarial, los procesos de control de acceso de dicha institución deben seguir la dinámica que propone este modelo, donde el interés fundamental recae sobre las tareas que desarrollará cada individuo, teniendo estas un carácter cambiante.

Control de Uso

El modelo de control de uso (*Usage Control*, UCON) es un marco conceptual que cubre las áreas de autorización, condición y obligación, de una manera sistemática para proporcionar un marco de propósito general y unificado para la protección de los recursos digitales. UCON no es un sustituto para el control de acceso tradicional, la gestión de confianza, o la gestión de derechos digitales. Más bien abarca estas tres áreas y va más allá en su definición y alcance, además de lograr un control detallado de los recursos digitales.

Los modelos $UCON_{ABC}$ constan de ocho componentes principales: sujetos, atributos de sujetos, objetos, atributos de objetos, los derechos, autorizaciones, obligaciones y condiciones. Las autorizaciones, obligaciones y condiciones son predicados funcionales que tienen que ser evaluados por decisión de empleo. Cada predicado se puede dividir en predicados detallados. Por su parte los sujetos, objetos y

derechos pueden dividirse en varios componentes detallados con diferentes perspectivas. Los controles de acceso tradicionales utilizan sólo las autorizaciones de los proceso de decisión, mientras que las obligaciones y condiciones son nuevos conceptos que se han discutido recientemente para resolver ciertas deficiencias mostradas en los controles de acceso tradicionales [7].

Sin embargo, a pesar de representar una evolución sustancial de los modelos de control de acceso tradicionales, su carácter orientado al uso de determinados recursos se aleja de las necesidades propias de una institución donde los objetos a proteger son áreas físicas y no recursos digitales o lógicos.

Control de Acceso Basado en Políticas

El control de acceso basado en políticas (*Policy Based Access Control*, PBAC) es una armonización y normalización del modelo ABAC a nivel empresarial en apoyo a los objetivos específicos de gobernanza. PBAC combina los atributos de los recursos y el medio ambiente con la información del solicitante sobre el conjunto particular de circunstancias bajo las cuales la solicitud de acceso se lleva a cabo, y utiliza los conjuntos de reglas que especifican si se permite el acceso bajo la política de la organización para esos atributos en esas circunstancias.

Aunque PBAC es una evolución del ABAC, es un modelo mucho más complicado. Partiendo que los atributos tienen que ser mantenidos en toda la empresa, así como es necesario diseñar e implementar sistemas de nivel empresarial para acomodar PBAC. Esto incluye bases de datos, servicios de directorio, y otras aplicaciones de middleware[1] y administración, donde todos deben estar integrados. En contraste con otros modelos de control de acceso, PBAC requiere no sólo un complicado nivel de aplicación lógica para determinar el acceso sobre la base de atributos, sino también un mecanismo para especificar las reglas de políticas en términos inequívocos, así como asegurar que toda la empresa utiliza los mismos atributos para el acceso, además de que provengan de una fuente autorizada [8].

Control de Acceso Basado en Roles

El modelo de control de acceso basado en roles (*Role-Based Access Control*, RBAC) es una tecnología que se ha desarrollado relativamente rápido, en cuanto a control de accesos se refiere, más aún si es comparado con sus sistemas antecesores DAC y MAC. Esto se debe principalmente a la forma jerárquica de funcionar que tiene el control de acceso basado en roles, ya que es más fácil administrar un sistema asignando roles a los usuarios, y así llevar una gestión confiable y de bajo costo. RBAC enfoca su interés principalmente en determinar qué usuarios y qué grupos de usuarios pueden ejecutar qué tipo de operación sobre qué tipo de recurso.

[1] **Middleware**: Se define como la capa de software que se encuentra entre el sistema operativo y las aplicaciones en cada sitio del sistema.

En RBAC, los permisos se encuentran asociados con los roles y los usuarios son miembros de estos, de manera que los roles son los que adquieren permisos, posibilitando que la administración en el control de acceso resulte más sencilla. Los roles son creados en forma centralizada para las distintas funciones de trabajos en una organización. Cada usuario tiene asignado uno o varios roles, por lo que puede ser asignado de un rol a otro y a su vez tener diferentes permisos, dependiendo del rol que esté llevando a cabo [9].

Una vez analizados los modelos anteriores se decide implantar el RBAC a partir de la flexibilidad y neutralidad que posee el mismo. Además ofrece mayor eficiencia que otros sistemas de control de acceso, ya que la forma de trabajar centralizada posibilita eliminar algunas abstracciones que se tienen con otros métodos.

Usuarios, Roles y Permisos en RBAC

Para que un sistema sea confiable se debe de tener en cuenta cuales podrían ser los caminos para llegar a la confiabilidad, los niveles que existen en los usuarios (por ejemplo: usuario y súper usuario), y los permisos o privilegios que dichos usuarios puedan llegar a tener.

Cuando en una organización se cambia un usuario hacia otro puesto, éste tiene que cambiar de permisos y responsabilidades, lo que resulta difícil y llega a ser de alto costo. Estos problemas pueden ser evitados por RBAC debido a que son los roles del usuario los que hacen que se tenga acceso al sistema, en vez de que sea la identificación del usuario.

Por lo tanto, se tiene que un ***usuario*** es la persona que pertenece a una organización, cuya labor básicamente es cumplir una función, en este caso un rol o múltiples roles.

Cada ***rol*** realiza la función que se tiene que desempeñar, y puede tener a uno o más usuarios asignados a su cargo. Los ***permisos*** determinan los recursos que pueden ser accedidos y luego a cada rol se le asignan diferentes privilegios, los cuales son asignados dependiendo de las capacidades que se tengan para ejecutar el rol [9].

Modelos RBAC y su evolución

Con el paso del tiempo el concepto de RBAC ha ido evolucionando. A continuación se muestran algunas de las definiciones al respecto. A medida que los modelos RBAC han progresado, han ido incorporando funcionalidades adicionales (*ver Tabla 1: Modelos RBAC*).

Modelos	Jerarquías	Restricciones
RBAC0	No	No
RBAC1	Si	No
RBAC2	No	Si
RBAC3	Si	Si

Tabla 1: Modelos RBAC

Según el Instituto Nacional de Estándares y Tecnología (*National Institute of Standards & Technology*, NIST) el control de acceso basado en roles es un modelo que está dividido en los siguientes componentes: RBAC centralizado, RBAC jerárquico, relaciones estáticas de separación de cargas y relaciones dinámicas de separación de cargas.

Una vez analizadas las principales características de RBAC se puede concluir que el mismo ofrece diverso beneficios sobre cualquier otro método de control de acceso, como es el caso de: la administración simplificada de sistemas, productividad organizacional mejorada, reducción de tiempo muerto de nuevo empleado, seguridad e integridad mejorada de sistemas y conformidad reguladora simplificada. Por otro lado el uso de roles para la gestión de accesos permite a los administradores de sistemas llevar un mejor control.

RBAC es reconocido como uno de los modelos que más ha revolucionado en los últimos años, ya que su aporte fundamental está en que puede llegar a funcionar con grandes sistemas, y al mismo tiempo brindarle seguridad [9].

En el sistema propuesto se utilizará el modelo de control de acceso basado en roles RBAC, pues al ser la fusión de los otros modelos, se consigue que el proceso para gestionar las políticas de control de acceso se realice de manera centralizada y con un alto grado de flexibilidad. Se garantiza además que el sistema de control de acceso sea adaptable a las condiciones particulares y estructura de la institución.

Principales Sistemas de Control de Acceso en el Mundo

Los sistemas de control de acceso han tomado gran auge a nivel mundial a partir de la necesidad que representan como mecanismo fundamental para proteger los recursos que se posee. Como consecuencia de la demanda existente en nuestros días, muchas son las instituciones que han apostado por desarrollar este tipo de sistemas, generándose una gran competencia por dominar este mercado y lograr establecer cada uno de sus productos. Dentro de los más reconocidos se pueden mencionar los siguientes:

YTime

YTime es un sistema de control de presencia totalmente autónomo potenciado por ByTech[2], que permite a pequeñas y medianas empresas la gestión horaria de sus empleados de una manera absolutamente fácil, precisa y económica. Integra un reloj chequeador con lector de huellas y tarjetas. Posee versiones para 50 y 250 empleados y hasta 4 terminales en la misma instalación, así como portal del empleado para consultas de datos personales. La aplicación es 100% web compatible con cualquier navegador y es capaz de integrarse en el control de accesos Sixdoors y Watch & Mochi V2 potenciados por la misma empresa [10].

[2] **ByTech**: Es una empresa líder en el diseño, fabricación y distribución de dispositivos de Control de Acceso.

Dentro de las principales funcionalidades que provee YTime se tiene la creación de jornadas de trabajo, verificar incidencias, asignar calendarios laborales a una persona o a un departamento completo, establecer períodos de vacaciones, enviar mensajes personalizados a uno o a varios empleados, generar informes a medida y corregir fichajes manualmente. Permite además la generación de informes de manera remota a través de un simple navegador web pudiendo incluso exportarlos a Excel.

CS-Access

CS-Access es un software de control de acceso de personal especialmente diseñado por Grupo SPEC para controlar los accesos y garantizar la seguridad de las instalaciones de las pequeñas y medianas empresas de cualquier sector de actividad.

CS-Access es un completo sistema de seguridad que permite la apertura de puertas e impedimentos de paso mediante tecnología de huella digital, mano, tarjeta de proximidad, tarjeta magnética o PIN, o una combinación de ellas [11].

Software de Control de Acceso Suprema BioStar

BioStar es un software de administración que funciona en PC sobre la plataforma de Microsoft Windows. El sistema de control de acceso BioStar se basa en conectividad de IP y seguridad biométrica. El software habilita los dispositivos de control de acceso Suprema para realizar funciones de control de acceso exhaustivo, incluyendo administración de usuarios, administración de dispositivos, control de puertas, zonas, monitoreo a tiempo real, entre otras [12].

Software de Control de Acceso AMADEUS 5

Amadeus 5 es un sofisticado software de control de acceso y administración de alarmas, que permite de forma centralizada y en tiempo real, controlar el flujo de empleados y visitantes por las dependencias de una empresa, monitorear y reaccionar ante alarma.

Este transforma las instalaciones en un edificio inteligente. Por ejemplo, el pase de una tarjeta en un lector apagará automáticamente las luces y la calefacción en cualquier área designada, permitiendo así el ahorro de energía.

Amadeus 5 permite definir múltiples grupos de acceso; programas diarios, semanales y feriados; los cuales se usan para especificar horas a las que los empleados y visitantes pueden acceder a las diferentes zonas de la empresa, horas a las que los lectores operan en diferentes niveles de seguridad y los horarios en los que las zonas están bajo alarma.

Este software de control de acceso incluye un grupo de características y funcionalidades que refuerzan la gestión de accesos en una institución. Entre las más relevantes se encuentran: el control de visitantes, visualización de eventos y transacciones, definición de múltiples eventos como alarmas, monitoreo gráfico

de alarmas, definición de lectores que necesitan doble validación, generación de informes, administración de estacionamiento de vehículos, control de ascensores, rondas de guardias, entre otras [13].

Softar® ACCESOS

Softar® ACCESOS es una aplicación de propósito general para el control de accesos, idónea para parqueos, recintos feriales, fábricas, obras, centros de congresos, museos y otras instalaciones. Este software permite controlar eficazmente zonas, usuarios y grupos, expedir tarjetas de acceso, así como obtener información sobre permanencias de personas en diferentes zonas.

Dentro de las principales funciones y características que posee se encuentran: fichas de usuarios con multitud de datos personales; permisos definibles por usuarios, por grupos o por tarjetas individuales; control de instalaciones, zonas y puntos de paso; contadores asociados a lectores para aforo[3], disponibilidad de plazas en parqueos e informes de permanencia en zonas [14].

PRINCIPALES SISTEMAS DE CONTROL DE ACCESO EN CUBA

En Cuba el uso de sistemas de control de acceso se ha ido extendiendo en los últimos años a partir de la necesidad que existe de preservar la seguridad de las instituciones. La implantación de los mismos está a cargo principalmente de entidades dedicadas a proveer servicios de seguridad y protección como lo son: Servicios Especializados de Protección, SEPSA y Servicios Especializados Integral, SEISA, donde los productos instalados provienen regularmente de otros países. Por otro lado con el auge que ha alcanzado el desarrollo del software en la isla, han surgido diversas entidades dedicadas a la creación de dichos sistemas, entre la que podemos encontrar la empresa (Tecnologías y Sistemas, DATYS[4]). A continuación se presentan los sistemas de control de acceso más relevantes en Cuba, ya sea por su uso o porque son desarrollos de carácter nacional.

Biomesys

Biomesys control de asistencia es un sistema que aprovecha las bondades de las tecnologías que aplican la biometría, para registrar los eventos de asistencia en una organización por medio de la identificación de los empleados y de la autenticación de su identidad mediante un censor biométrico de huellas dactilares. A partir de la captura de identificaciones biométricas únicas, el sistema se convierte en un generador de datos altamente confiable por su bajo o casi nulo nivel de vulnerabilidad por la suplantación de identidad. De utilidad para los que pretenden apoyarse en un instrumento sencillo con un enfoque flexible y

[3] **Aforo**: Cálculo de una cantidad existente en un depósito.
[4] **DATYS**: Empresa cubana que produce bienes y servicios informáticos, desarrollando el empleo integral de las tecnologías de la información, de las comunicaciones y de seguridad técnica, con alta calidad y eficiencia.

adaptable a la estructura funcional de la organización, que propone herramientas personalizables para el control y la toma de decisiones en el área de los recursos humanos.

Biomesys tiene como características distintivas las siguientes: no establece limitaciones de implementación asociadas al tipo de organización; se puede integrar de forma rápida con diferentes medios de autenticación como escáneres biométricos, credenciales de bandas magnéticas, tarjetas de códigos de barras y proximidad; posee un módulo de captura biométrica para el control de las entradas y las salidas del personal [15].

XymaSafeAccess

Sistema de identificación y control de acceso físico, formado por una red coordinada de tarjetas de identificación, lectores electrónicos, bases de datos especializadas, software y computadoras diseñadas para monitorear y controlar el tráfico a través de puntos de acceso. Posee cuatro módulos y ofrece variantes de configuración en dependencia del nivel de seguridad que se quiera implementar.

XymaSafeAcces ofrece un proceso de registro seguro, que establece la entidad de cada individuo y determina que la persona está autorizada para utilizar los privilegios o servicios que están siendo brindados; procedimientos para emitir tarjetas de identidad con seguridad y asegurar que los documentos de identidad sean emitidos solamente por la entidad autorizada para expedir dichas tarjetas, y que solamente sean emitidos documentos de identidad para las personas correctas; procedimientos para monitorear el uso de la identificación; un proceso de autenticación que implementa la cadena de confianza previamente establecida, verificando la identidad de los portadores del identificador y la legitimidad de las tarjetas de identidad y sus credenciales [16].

XymaSafeVision

Es un sistema de video protección profesional basado en tecnología IP. Este tiene como características que no está atado a ninguna tecnología específica, se integra a los más diversos equipamientos de cómputo, cámaras y servidores para cámaras analógicas, tiene incorporado algoritmos de reconocimiento de patrones que permiten identificar y verificar matrículas de los autos, definir perímetros para la detección de movimiento o cantidad de objetos, entre otras. Alta compatibilidad con los productos de Axis, Vivotek, Panasonic y Sony, y se trabaja ampliando la gama de marcas y modelos. Capacidad ilimitada de manejo de usuarios y cámaras. Grabaciones de forma manual, programada, continua y por detección de movimiento o por eventos de alarmas predefinidos. Interactúa con servicios de mapas y planos en planta de las instalaciones en apoyo a la administración y utilización del sistema. Posee herramientas para la gestión del ancho de banda de la red y la calidad de las imágenes a través de la personalización de parámetros de video de cada canal y para cada función. Ofrece un buen servicio de salidas estadísticas y

nominales que permiten evaluar el funcionamiento y uso del sistema. El sistema es auditable al conservar todas las transacciones y operaciones que se realizan durante la explotación [17].

Frontpas

Es una solución integral para la gestión y control de la frontera. Es un sistema integrado de registro, control y vigilancia de pasos fronterizos legales, que trabaja con estándares de seguridad y calidad internacionales, en un ambiente amigable y flexible.

La solución provee una plataforma tecnológica que facilita el control y gestión del tránsito de personas en las fronteras legales y sus flujos en condiciones de estricta seguridad permitiendo contrarrestar, según el interés nacional, el tráfico ilegal de personas, mercancías y combustibles.

Frontpas permite gestionar el proceso chequeo migratorio en general desde cualquier punto fronterizo. Capturar y utilizar la información multibiométrica (rostro y huella) para detectar y evitar la suplantación de identidad. Chequear la identidad de la persona contra listas de control y arraigos. Chequeo de listas de pasajeros de vuelos previo a la llegada de los viajeros. Trabajar en régimen conectado o desconectado del núcleo central, lo que le da gran confiabilidad a la solución. Configurar los puestos de trabajo, los dispositivos de captura, los roles y usuarios del sistema, las actividades del chequeo, las listas de control propias, los nomencladores y el comportamiento de procesos del sistema. Alertar en tiempo real a las autoridades correspondientes, a partir de reglas pre definidas [18].

Principales Sistemas de Control de Acceso en la UCI

Con el objetivo de garantizar una mayor seguridad de los recursos que posee la universidad, desde su creación se han estado implantando diversos sistemas que permiten limitar el acceso del personal que por esta circula. La mayoría de las soluciones han sido el resultado de los trabajos de diplomas presentados en el centro que respondían al objeto de estudio en cuestión. Su implantación y puesta en funcionamiento, en muchos de los casos no se ha llevado a cabo totalmente, por lo que actualmente la UCI no cuenta con un sistema centralizado que le permita gestionar las políticas de acceso de manera global sobre cada una de las áreas de la universidad. Por otro lado existen algunos de estos sistemas, que sí se encuentran en funcionamiento y sus principales características se relacionan a continuación.

Sistema de Acreditación

Este sistema brinda un servicio de certificación de identidad a otros sistemas informáticos, como los que se utilizan para el control del acceso. Tiene almacenados los datos de todo el personal que labora y estudia en la Universidad: estudiantes y todo tipo de trabajadores. Lo más importante es que le asigna a cada persona un código único, para su identificación. Este sistema está estructurado por los siguientes módulos; administración, configuración, identificación, detección de rostros y seguridad.

Sistema de Control de Acceso a los Comedores

Mediante este sistema se controla en los comedores de los diferentes Complejos Alimenticios el acceso de los estudiantes, profesores y trabajadores durante las tres sesiones de servicio: desayuno, almuerzo y comida. El mismo se divide en dos partes: el control de acceso y la gestión de comensales. El acceso se controla registrando el código de barras, que se encuentra en la identificación de cada persona, en cada una de las puertas de los comedores. La gestión de comensales permite a los directivos la asignación correspondiente a cada uno de los comedores y puertas, además de ofrecer reportes como cantidad de comensales que han pasado y desglosarlo por puerta o por tipo.

Sistema de Control de Acceso a los Laboratorios de Producción

Este sistema lleva el control de los proyectos que radican en los laboratorios destinados a los procesos productivos y por tanto de las personas que pueden tener acceso a dichos laboratorios. En el sistema se chequea qué personas tienen acceso o no a los laboratorios, verificando que estén en la base de datos correspondiente, mediante el número de la identificación.

De este sistema, en la universidad existen varias implementaciones, cada una de ellas específica para el área productiva donde se encuentra, lo que hace que no exista una base de datos centralizadas con todos los datos referentes a todos los laboratorios de producción, a pesar de que son aplicaciones que no están bien concebidas ni con una amplia documentación. Sin embargo la aplicación a desarrollar es capaz de gestionar toda la información que manejan dichas aplicaciones de forma centralizada y con una mayor eficiencia.

GYES Identity Manager

Es un sistema de autenticación centralizado, que permite, a partir de contratos previos entre el sistema y las entidades de una organización, establecer una autenticación segura y con varios niveles de complejidad. Dichas entidades pueden ser usuarios, dispositivos, sistemas y recursos de una red en general. Gyes además, está basado en estándares para garantizar interoperabilidad con múltiples sistemas; este incluye entre uno de sus componentes el Subsistema de Autorización, encargado de llevar a cabo la gestión de los recursos de una red así como autorizar el acceso a los mismos, utiliza LDAP como repositorio de identidades, específicamente OpenLDAP, aunque la arquitectura permite extender el sistema con la incorporación de otros repositorios de identidades de diferentes índoles. Gyes está pensado para la integración con componentes de autorización y aprovisionamiento, completando así una suite para el manejo identidades. Entre sus características fundamentales figuran que Gyes Authentication Server permite:

Autenticación centralizada: Permite tener varios recursos o aplicaciones autenticados con una misma sesión.

Autenticación flexible: Permite definir múltiples esquemas de autenticación a través de la combinación de diversos factores (biométricos, basados en X.509, dispositivos seguros y frases de acceso), además los datos para la autenticación pueden estar dispersos en diferentes nodos del sistema. **Integración fácil:** La arquitectura permite la interacción con otros servicios de autorización o aprovisionamiento.

VALORACIÓN DE LOS SISTEMAS ESTUDIADOS

Después de haber hecho un estudio de los sistemas de control de acceso tratados, se puede concluir que es necesario el desarrollo de un Sistema de Control de Acceso para la Universidad de las Ciencias Informáticas basándose la afirmación anterior en un grupo de aspectos a tener en consideración. Primero, los sistemas propios de la UCI, ya sea los que están en funcionamiento como los realizados en dicha universidad pero no implantados, no satisfacen las necesidades que se presentan a partir de que están orientados a fines específicos o no cumplen con los requisitos que se exigen para preservar la seguridad. Segundo, los productos existentes en el mercado nacional están orientados fundamentalmente a funciones específicas dentro de los sistemas de control de acceso y sus características no se ajustan a las condiciones de la universidad, por lo que se desecha la propuesta de utilizarlos en la UCI. Tercero, las soluciones estudiadas a nivel internacional, a pesar de integrar las tecnologías más avanzadas, incluir diversas funcionalidades que responden a las exigencias más complejas, adaptarse a la mayoría de los entornos que presentan las instituciones; su aplicación en la UCI resulta inapropiada a partir de los altos costos por concepto de licencia y soporte de las mismas.

Como parte del estudio previo se logró identificar los principales tipos de software de control de acceso existentes, como son los autónomos, off-line y on-line. El uso de cada tipo de solución está en correspondencia con las características de la organización en la que se vaya a desplegar y el nivel de seguridad y control que la misma desee implantar.

Algunas de las funcionalidades y características más comunes de estos software son la gestión de instalaciones, zonas y puntos de acceso o puertas, la gestión de usuarios, grupos de usuarios y roles, gestión y asignación de permisos a los roles y grupos, manejo de registro, recuperación de fichajes, identificación y clasificación del personal, creación de credenciales, entre otras.

Por otro lado muchas de las aplicaciones de control de acceso se extienden con grupo de módulos opcionales que contribuyen a un mejor control de los recursos y personal de la institución. Entre estos módulos se encuentra, la administración de alarmas, vigilancia por CCTV, control de iluminación y ventilación, control de presencia u horaria, administración de estacionamiento de vehículos, control de ascensores, ronda de guardia y sistema de emergencia.

TECNOLOGÍAS DE HARDWARE UTILIZADAS PARA EL CONTROL DE ACCESO

Una amplia gama de tecnologías de hardware brindan una solución efectiva tanto en las grandes empresas que requieren máxima seguridad, robustez y flexibilidad de programación, así como a pequeños comercios que necesitan precios económicos y facilidad de uso. Un software de control de acceso, con sus accesorios, permite llevar el control de horarios a lugares en los cuales antes no era posible por cuestiones operativas (oficinas, obras, personal móvil e inspectores) o de costo. El reto, entonces, es encontrar la solución que garantice una relación costo/beneficio y que requiera la menor cantidad de esfuerzo en su implantación y uso. A continuación se relacionan algunas de las tecnologías más destacadas en el área del control de acceso.

Claves por Teclado: Esta opción es la más económica, pero la menos segura. Hace tiempo que han caído en desuso, por lo que no se han generado hasta el momento nuevas aplicaciones donde puedan resurgir como una opción válida.

Tarjetas de Banda Magnética: Es la tecnología más conocida y por consiguiente difundida, ya que se utiliza en todos los sistemas de tarjetas de crédito y compra. Su uso más habitual está destinado a la identificación rápida y segura de su portador mediante la incorporación de elementos identificativos como ID de usuario, nombre o fotografía así como la aplicación de una tecnología de lectura. Su ventaja prevalece en su popularidad y bajo costo, aunque es de todos los medios de identificación, el más vulnerable. La banda magnética de la tarjeta, debe ser tratada con cierto cuidado para evitar que se raye o sea expuesta a campos magnéticos que la borren, por tales motivos, no son recomendables para usar en ambientes industriales. Solo se recomiendan en oficinas o establecimiento administrativos [19].

Tarjetas de Código de Barras: Actualmente los códigos de barras (lineales y multidimensionales) son la forma más extendida para codificar datos y automatizar su lectura mediante dispositivos destinados a tal efecto. Poseen un bajo costo de producción, rapidez en la lectura y una amplia aceptación. Como desventajas presentan una relativa facilidad de fraude y la imposibilidad de codificar gran cantidad de datos. El código de barras almacena información que puede ser reunida en él de manera rápida y con una gran precisión. Los códigos de barras representan un método simple y fácil para codificación de información de texto que puede ser leída por dispositivos ópticos, los cuales envían dicha información a una computadora como si la información hubiese sido tecleada [20].

Touch Memories: Es una pastilla electrónica, de unos 16 mm de diámetro y encapsulada en acero inoxidable, comúnmente denominada llave electrónica, son memorias de contacto inalterables con la apariencia externa de una pila de calculadora que contienen en su interior un código irrepetible y que sirve para identificar a su portador, además brindan un alto nivel de seguridad, ya que son altamente resistentes

al desgaste, siendo ideales para ambientes industriales aunque no son recomendables para ambientes con alto grado de generación de corriente estática. Se provee con un llavero para su fácil operación [21].

Tarjetas de Proximidad: Son prácticamente imposibles de duplicar debido a su diseño tecnológico, lo que las convierte en una de las tecnologías más modernas y efectivas. Son prácticas y de bajo costo de mantenimiento, ideales en situaciones de máxima seguridad y alta tecnología, incluso pueden ser leídas dentro una cartera.

Las tarjetas de proximidad son cada vez más usadas como portadoras de información ya que el desgaste por rozamiento no existe y su duración es prácticamente indefinida. Su uso resulta cómodo debido a que estas tarjetas plásticas poseen internamente una antena y no necesitan ser insertadas en un lector, el chip de la tarjeta se comunica con el lector por radiofrecuencia (=RF) e identifica al titular (=ID) sin contacto físico. Por ello se llaman también "tarjetas contactless" o "tarjetas RFID". Son utilizadas en aquellos escenarios donde la fluidez y el ahorro de tiempo son factores importantes como en controles de accesos en eventos, peajes o transporte público. Estas tarjetas de proximidad utilizan un chip que trabaja a una frecuencia de 125 KHz con una rápida velocidad de lectura, con un alcance que varía desde 2 cm a 1 m, dependiendo del lector, y capaz de almacenar hasta 1 Kb de información. Las tarjetas de baja frecuencia se suelen utilizar en servicios que no requieren más de un dato como el control de acceso al transporte público, dispensadores automáticos, entre otros. [22]

Sistemas Biométricos: Se entiende por sistema biométrico a un sistema automatizado que realiza labores de biometría. Es decir, un sistema que fundamenta sus decisiones de reconocimiento mediante una característica personal que puede ser reconocida o verificada de manera automatizada. En esta sección son descritas algunas de las características más importantes de estos sistemas. Las características básicas que un sistema biométrico debe cumplir para la identificación del personal pueden expresarse mediante:

- El ***desempeño***, que se refiere a la exactitud, la rapidez y la robustez alcanzada en la identificación.
- La ***aceptabilidad***, que indica el grado en que las personas están dispuestas a aceptar un sistema biométrico en su vida diaria.
- La ***fiabilidad***, que refleja cuán difícil es burlar al sistema. El sistema biométrico debe reconocer características de una persona viva, pues es posible crear dedos de látex, grabaciones digitales de voz, prótesis de ojos, entre otras.

En la actualidad existen sistemas biométricos que basan su acción en el reconocimiento de diversas características, las técnicas biométricas más conocidas son nueve y están basadas en los siguientes indicadores biométricos [23]:

Principales indicadores biométricos:

- Huella Digital (*Fingerprint*).

- Geometría de Mano (*Hand Geometry*).
- Iris.

Otros indicadores biométricos:

- Rostro.
- Voz.
- Firma.
- Patrones de Venas.
- Termograma[5] del rostro.
- Patrones de la retina.

Tecnología utilizada en la UCI: **Código de Barras**

La tecnología de código de barras es muy económica y de fácil implementación, ya que basta con poseer una impresora de calidad para realizar impresos de carnés de identificación con dichos códigos. Existen varios tipos de codificación, a continuación se muestran los más importantes en la actualidad:

- Código 39.
- Código 39 ASCII Total.
- Codabar.
- Intercalado 2 de 5.
- Código 128.
- Código Universal de Producto (*Universal Product Code*, UPC).

En la UCI el más utilizado es el Código 39, debido a que codifica números, letras mayúsculas, y algunas marcas de puntuación, Ej. (Mayúsculas A-Z, números 0-9, "espacios" y símbolos: -, +, /, $, %, *). El Código 39 puede ser variable en longitud, permitiendo hacer códigos de cualquier número de dígitos, convirtiéndose este formato en la norma principal para el gobierno, industria, educación y aplicaciones de negocios.

Ilustración 1: Ejemplo de Código 39

[5] **Termograma**: Técnica que permite medir temperaturas a distancia con exactitud y sin necesidad de contacto físico con el objeto a estudiar. Mediante la captación de la radiación infrarroja del espectro electromagnético, utilizando cámaras termográficas o de termovisión, se puede convertir la energía radiada en información sobre temperatura.

Beneficios del Código de Barras

El código de barras es el mejor sistema de colección de datos mediante identificación automática, y presenta muchos beneficios, a continuación se relacionan los más importantes:

- Virtualmente no hay retrasos desde que se lee la información hasta que puede ser usada.
- Se mejora la exactitud de los datos, hay una mayor precisión de la información.
- Se tienen costos fijos de labor más bajos.
- Se puede tener un mejor control de calidad, mejor servicio al cliente.
- Se puede contar con nuevas categorías de información.
- Se mejora la competitividad.
- Se reducen los errores.
- Se capturan los datos rápidamente.
- Se mejora el control de las entradas y salidas.
- Precisión y contabilidad en la información, por la reducción de errores.
- Eficiencia, debido a la rapidez de la captura de datos.

El incremento de la velocidad y exactitud en la toma de datos, lleva a reducir errores así como a un ahorro de tiempo y dinero [24].

Tipos de lectores de Código de Barras

En los sistemas de control de acceso actuales, los códigos de barras pueden ser leídos de muchas formas, utilizando diferentes dispositivos. Generalmente un sistema de lectura se compone de dos partes: una interfaz, llamada por lo regular "decodificador", y lo que se conoce por el término de "dispositivo de entrada". Existen tres tipos de lectores: batch portátil, portátiles de radiofrecuencia y combinados [24].

Entrada de datos por teclado: Portátiles o montados se conectan a una computadora y transmiten los datos al mismo tiempo que el código es leído.

Lectores portátiles tipo batch: Recolección de datos en campo, son operados con baterías y almacena la información en memoria para después transferirla a una computadora.

Lectores de radiofrecuencia: Almacenan también la información en memoria, sin embargo la información es transmitida a la computadora en tiempo real. Esto permite el acceso instantáneo a toda la información para la toma de decisiones.

TECNOLOGÍAS, LENGUAJES Y HERRAMIENTAS UTILIZADAS EN EL DESARROLLO

.NET Framework

.NET Framework es un componente integral de Windows que admite la compilación y la ejecución de la siguiente generación de aplicaciones y servicios web XML. Este proporciona un entorno coherente de programación orientada a objetos, en el que el código de los objetos se puede almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota, además de un entorno de ejecución de código que reduce al máximo posible la implementación de software y los conflictos de versiones. Ofrece un entorno de ejecución de código que promueve la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza. Proporciona un entorno de ejecución de código que elimina los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos. Este brinda al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en la web. Esta centra toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se pueda integrar con otros tipos de código [25].

.NET Framework es además un entorno de desarrollo distribuido, en el cual se reúnen en conjunto lenguajes y servicios que simplifican el desarrollo y ejecución de aplicaciones .NET. Además de soportar múltiples lenguajes de programación es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes: (*C Sharp*, C#), *Visual Basic*, C++, (*Java Sharp*, J#).

La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar, a la funcionalidad que los servicios web proveen. Es multiplataforma, debido a que posee el Motor Común de Ejecución (*Common Language Runtime*, CLR) por lo que un programa .NET podrá ser compilado y ejecutado en cualquier plataforma que incluya un CLR[6] [26].

Active Server Pages .NET

Active Server Pages, ASP.NET es un modelo de desarrollo web unificado que incluye los servicios necesarios para crear aplicaciones web empresariales con código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones de este se tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el *Common Language Runtime* (CLR), entre ellos *Microsoft Visual Basic* y C#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del CLR, seguridad de tipos, herencia, entre otros [27].

[6] **CLR**: Núcleo del Framework .Net, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes.

ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación web anteriores como son: mejor rendimiento, compatibilidad con herramientas de primer nivel, eficacia y flexibilidad, simplicidad, facilidad de uso, escalabilidad y disponibilidad, posibilidad de personalización y extensibilidad, además de seguridad [28].

Lenguaje C Sharp

C Sharp, C# es el lenguaje orientado a objetos diseñado por Microsoft para su plataforma .NET. Es multiplataforma y combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, *Java, Visual Basic* o *Delphi*. Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programar usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que este carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET. Microsoft ha escrito la mayor parte de la Biblioteca de Clases Base (*Base Class Library*, BCL) usando C#, por lo que su compilador es el más depurado y optimizado de los incluidos en el .NET Framework SDK[7].

El lenguaje proporciona la capacidad de generar componentes de sistemas duraderos en virtud, además de poseer innumerables características como son, su sencillez, modernidad, orientado a objetos, orientado a componentes, gestión automática de memoria, seguridad de tipos, instrucciones seguras, sistema de tipos unificado, extensibilidad de tipos básicos, extensibilidad de operadores, extensibilidad de modificadores, versionable, eficiente y compatible [29].

Servicios Web

Un servicio web (*web service*) es una aplicación identificada por un Identificador Uniforme de Recursos (*Uniform Resource Identifier*, URI), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los servicios web hacen posible la interacción entre “agentes” software utilizando mensajes XML intercambiados mediante protocolos de Internet. Un servicio web es un componente de software que puede ser registrado, descubierto e invocado mediante protocolos estándares de Internet. Permiten exponer y hacer disponibles funcionalidades de los sistemas informáticos de las organizaciones mediante tecnologías y protocolos web estándar. Cada servicio web tiene la responsabilidad de realizar un conjunto de funciones concretas y bien definidas. Los servicios web actúan como componentes independientes que se pueden integrar para formar sistemas distribuidos complejos.

Por otro lado, permiten que las aplicaciones trabajen en conjunto, haciendo uso de funcionalidades brindadas por otras aplicaciones independientemente de cómo se hayan creado, cuál sea el sistema

[7] **Software Development Kit, SDK**: Es un paquete de programación que permite desarrollar aplicaciones para una plataforma específica.

operativo o la plataforma en que se ejecutan y cuáles sean los dispositivos utilizados para obtener acceso a ellos [30]. Aunque los servicios web son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

En la UCI la mayoría de las aplicaciones que se utilizan en la intranet ofrecen o "consumen" servicios web de otras, es decir, existe una interrelación entre los sistemas de la red para lograr la reutilización y la funcionalidad de estas.

Visual Studio .NET 2010

Visual Studio es un Entorno de Desarrollo Integrado (*Integrated Development Environment*, IDE) que permite a los desarrolladores crear aplicaciones (web, escritorio), aplicaciones para teléfonos inteligentes y ordenadores de bolsillo, servicios web y otras utilidades. Este IDE tiene una característica especial, es multilenguaje por lo que soporta varios lenguajes de programación entre los que se encuentran C#, *Visual Basic* y Visual C++ [31].

Visual Studio .NET 2010, aparte de presentar varias novedades trae consigo importantes mejoras para fomentar la colaboración de los equipos multidisciplinarios implicados en los proyectos. En *Team Foundation Server*, TFS se integra toda la información, convirtiéndose en un repositorio no sólo del código, sino también de requisitos, casos de uso, pruebas, incidencias y planes de proyecto, entre otros documentos. La nueva plataforma dispone de toda la potencia del análisis de datos de Microsoft SQL Server 2008 para realizar informes. De esta forma, se alinea la información de negocio con el desarrollo, se reducen los tiempos de entrega y se optimiza el proceso de trabajo, siendo más sencilla la consecución de objetivos para todos los miembros del equipo.

Visual Studio .NET 2010 simplifica el desarrollo de aplicaciones en un ambiente que es una mezcla de lenguajes y lo hace a través de ciertas características como por ejemplo diseñadores visuales para XML, HTML, datos y código del lado del servidor. Además es capaz de proveer este nivel de integración porque cuenta con las facilidades del .NET Framework [32].

Lenguaje de Consulta Estructurado

Lenguaje de Consulta Estructurado (*Structured Query Lenguaje*, SQL) es el lenguaje utilizado para definir, controlar y acceder a los datos almacenados en una base de datos relacional. Como ejemplos de sistemas gestores de bases de datos que utilizan SQL podemos citar DB2, SQL Server, Oracle, MySQL, Sybase, PostgreSQL o Access [33].

El SQL es un lenguaje universal que se emplea en cualquier sistema gestor de bases de datos relacional. Tiene un estándar definido, a partir del cual cada sistema gestor ha desarrollado su versión propia. Este, en principio es orientado únicamente a la definición y al acceso a los datos por lo que no se puede

considerar como un lenguaje de programación como tal ya que no incluye funcionalidades como son estructuras condicionales, bucles, formateo de la salida, entre otras.

Se puede ejecutar directamente en modo interactivo, pero también se suele emplear embebido en programas escritos en lenguajes de programación convencionales. En estos programas se mezclan las instrucciones del propio lenguaje (denominado anfitrión) con llamadas a procedimientos de acceso a la base de datos que utilizan el SQL como lenguaje de acceso. Ejemplo de estos lenguajes se tiene a *Visual Basic*, *Java*, C#, PHP. El SQL proporciona una rica funcionalidad más allá de la simple consulta o recuperación de datos. Asume el papel de lenguaje de definición de datos (LDD), lenguaje de definición de vistas (LDV) y lenguaje de manipulación de datos (LMD). Además permite la concesión y denegación de permisos, la implementación de restricciones de integridad y controles de transacción, y la alteración de esquemas [34].

PostgreSQL

Es un sistema de gestión de base de datos relacionales orientado a objetos, publicado bajo la licencia (*Berkeley Software Distribution*, BSD[8]). Es más completo que MySQL ya que permite métodos almacenados, restricciones de integridad, vistas, entre otros. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada como Grupo Global de Desarrollo de PostgreSQL (*PostgreSQL Global Development Group*, PGDG).

Es el sistema de gestor de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. Utiliza el lenguaje SQL para llevar a cabo sus búsquedas de información, las bases de datos generadas dentro de servidores de SQL son bases de datos relacionales. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando [35].

Visual Paradigm

Visual Paradigm es una herramienta de Ingeniería de Software Asistida por Computación (*Computer Aided Software Engineering*, CASE) la cual propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

[8] **BSD**: es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution).

Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a ciertas necesidades: *Enterprise*, *Professional*, *Community*, *Standard*, *Modeler* y Personal. Además existe una alternativa libre y gratuita de este software. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos [36].

Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el ciclo de desarrollo de un software. UML ofrece un estándar para describir modelos, incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, además de aspectos concretos como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizables. UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, nos es un proceso, sino, que es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Permite modelar, construir y documentar los elementos que forman un sistema orientado a objetos. Tiene como objetivo brindar un material de apoyo que le permita al lector poder definir diagramas propios como también entender diagramas ya existentes [37].

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML se puede usar para modelar distintos tipos de sistemas como son: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

NHibernate

Es un framework de mapeo objeto-relacional (*Object-Relational-Mapping*, ORM[9]) que facilita el mapeo de los atributos entre una base de datos relacional y el modelo de los objetos de una aplicación, que además permite establecer estas relaciones. NHibernate (NH) es la conversión de Hibernate del lenguaje Java a C#, para su integración en la plataforma .NET. Algunas características de la filosofía de diseño de NHibernate han de ser destacada especialmente por su importancia, las cuales son:

- Puede utilizar los objetos definidos por el usuario.
- No utiliza técnicas como generación de código a partir de descriptores del modelo de datos.
- Abre las puertas a utilizar herencia en el modelo de datos.

[9] **ORM**: Framework encargados de adaptar el mundo de objetos al relacional.

NHibenate se encarga, justamente, de relacionar clases con tablas, es decir una tabla se mapea contra una clase y cada columna contra un atributo de dicha clase. Con este, los mapeos pueden escribirse en archivos XML o utilizando anotaciones y así NHibernate se comunicara con la base de datos y realizará las acciones requeridas por los objetos persistentes (inserción, actualización, borrado y selección) [38].

METODOLOGÍA DE DESARROLLO DE SOFTWARE UTILIZADA: FDD

El Desarrollo Manejado por Rasgos (*Feature Driven Development*, FDD) es un enfoque ágil para el desarrollo de sistemas; se basa en iteraciones cortas que entregan funcionalidades tangibles. En el caso del FDD las iteraciones duran dos semanas y transitan por los siguientes procesos:

- Desarrollar un Modelo Global.
- Construir una Lista de los Rasgos.
- Planear por Rasgo.
- Diseñar por Rasgo.
- Construir por Rasgo.

Los primeros tres se ejecutan al principio del proyecto. Los últimos dos se conciben en cada iteración. Cada proceso se divide en tareas y se da un criterio de comprobación. Los desarrolladores entran en dos tipos: dueños de clases y programadores jefe. Los programadores jefe son los desarrolladores más experimentados, lo cual se les asignan rasgos a construir. Sin embargo ellos no los construyen solos, sino que identifican qué clases se involucran en la implantación de un rasgo y juntan a los dueños de dichas clases para que formen un equipo para desarrollar ese rasgo. El programador jefe actúa como el coordinador, diseñador líder y mentor mientras los dueños de clases hacen gran parte de la codificación del rasgo.

Dentro de las principales características de FDD figuran: su preocupación por la calidad, a partir de un monitoreo constante del proyecto, ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallas en el programa o el hecho de entregar menos de lo deseado, propone tener etapas de cierre cada dos semanas obteniéndose resultados periódicos y tangibles, se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorear, define claramente entregas tangibles y formas de evaluación del progreso del proyecto, no hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción [39].

Dicha metodología de desarrollo de software fue seleccionada para la solución que se desea implementar a partir de que la misma está definida para proyectos cortos y con un equipo de desarrollo pequeño, lo cual coincide con características del presente proceso a realizar. Esta metodología se basa en un proceso iterativo, permitiéndole al cliente, recibir después de cada iteración un parte funcional del componente

además de estar informado de la situación del mismo, de manera que si el desarrollo del producto se aleja de sus propósitos, el mismo puede intervenir y valorar estado de este, ya que él forma parte en todo el proceso de desarrollo. La metodología FDD está basada en la planeación e iteración por funcionalidades, que permite identificar los requisitos con mayor prioridad, y diseñarlos a través de un proceso iterativo, facilitando esto el diseño de la solución, la descripción textual y la revisión de los mismos antes de comenzar a implementar el componente.

CONCLUSIONES

En el presente capítulo se presentaron los conceptos fundamentales relacionados con la propuesta de solución al problema en cuestión. Se llevó a cabo un estudio de los principales sistemas de control de acceso existentes tanto a nivel internacional, nacional como en la propia universidad; concluyendo que ninguno resulta factible para solucionar la problemática existente a partir de que los más completos son costosos y otros, solo son aplicables a áreas específicas. Se analizaron los diferentes modelos de control de acceso seleccionando RBAC como el más conveniente para ser utilizado como base en la solución. Se examinaron la principales tendencias, tecnologías, herramientas y metodologías, quedando expuestas las utilizadas en el diseño e implementación de la aplicación propuesta.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

INTRODUCCIÓN

En el presente capítulo se describe el flujo actual de los procesos de control de acceso en la Universidad de las Ciencias Informáticas profundizando en los que serán objeto de informatización. Se presenta además el Modelo de Dominio que ilustrará las relaciones entre los principales conceptos del sistema. Por último se lleva a cabo la descripción de la solución propuesta a partir de la definición previa de los requisitos funcionales y no funcionales del sistema.

DESCRIPCIÓN DE LOS PROCESOS DEL NEGOCIO

Los procesos de negocio son la disposición ordenada de actividades que operan bajo un conjunto de procedimientos con el fin de conseguir un objetivo específico. El análisis de un proceso de negocio determina la interdependencia entre las actividades. Las actividades están relacionadas porque un hecho específico inicia la primera actividad del proceso, la cual a su vez, a través de la salida que produce lanza la actividad subsiguiente y así sucesivamente. El procedimiento consiste en determinar la secuencia de actividades siguiendo el flujo de información. Cuando las entradas y salidas de las actividades individuales están conectadas entre sí, emerge un proceso de negocio [40].

Flujo actual de los procesos del negocio

Actualmente el Departamento de Defensa, Protección y Seguridad es el encargado de efectuar y velar por el correcto funcionamiento del proceso de control de acceso en la UCI. Dicho proceso parte de la previa distribución de agentes de seguridad por cada uno de los puntos de acceso de la universidad, el número de agentes asignados por posta varía en dependencia del nivel de complejidad de la misma.

Para autorizar la entrada del personal al interior de la universidad, la persona que pretende acceder solicita el permiso y el agente de seguridad examina de manera visual la identificación que posee el solicitante. La validez de dicha identificación depende de que esta contenga las características básicas de un documento emitido en la universidad y que en su parte trasera posea el cuño impreso del órgano emisor. El agente además verifica que la foto del documento coincida con el rostro de la persona que desea entrar o salir. Una vez cumplidas las condiciones anteriores se le otorga el acceso al que lo solicita. Si la persona no posee el documento de identificación por diversos motivos (pérdida, robo o deterioro); esta se remite hacia la oficina de control de acceso, donde el técnico de control de acceso busca a la persona a través de la intranet, si se encuentra en la base de datos se le concede el acceso, si no se le

deniega. Además existen personas que tienen libre acceso a la universidad como altos dirigentes de las FAR, MININT y agentes del DTI, los cuales pueden entrar y salir libremente.

En caso de que la persona que solicita el acceso no pertenezca a la institución, se le considera visitante. El agente de seguridad revisa si el visitante figura en el libro de visitas y en caso positivo, procede a registrar el resto de sus datos para otorgarle el acceso luego de entregado un pase de visitante. Si no aparece en el ya mencionado libro establece contacto con el dirigente capacitado para autorizar la entrada, este permite la misma, el agente registra sus datos, entrega el pase y otorga el acceso.

De manera similar ocurre cuando se solicita el acceso para un vehículo. Primero se verifica de forma visual un documento impreso, de manera que tiene que coincidir el nombre de la persona con la chapa del vehículo y este proceso queda plasmado también en el documento, pero si la persona que solicita el acceso de su vehículo no pertenece a la UCI solo puede acceder en los horarios que trabaja en el centro (trabajadores externos que brindan algún servicio, entre otros), donde el técnico de control de acceso les concede un pase de "vía permanente".

Para los equipos electrodoméstico no existe revisión alguna por parte de los agentes de seguridad a la entrada, por lo que si la persona no declara que lleva algún artículo, esta puede acceder con el equipo electrodoméstico; pero si lo declara, la persona se dirige a la oficina de control de acceso, donde el técnico registra el activo en el libro de incidencias, siendo la estancia de este equipo electrodoméstico temporal o no. Para controlar la salida de estos equipos electrodoméstico el agente de seguridad le solicita a la persona el documento de autorizo que le realiza el decano de la facultad o director del centro al que pertenece, que tiene todos los datos de la persona y del equipo electrodoméstico, el agente de seguridad verifica con el solapín y el carnet de identidad si es la persona y solicita el número de serie del activo el cual también debe de coincidir con el que está plasmado en el documento de autorizo, si todos los datos coinciden se le concede el permiso de salida a la persona y a su equipo electrodoméstico.

Descripción del proceso a automatizar

Luego de realizado un análisis del funcionamiento de los procesos de control de acceso en la UCI se decidió automatizar aquellos que influyen directamente en la seguridad de la institución así como en la velocidad del servicio que se presta. A continuación se describen como quedarían dichos procesos automatizados:

Registrar acceso de personal perteneciente a la institución: La persona solicita el acceso, que puede ser la entrada o salida a un área determinada, el agente verifica la identificación a través de un dispositivo lector de código de barras y el sistema certifica si el solicitante tiene permiso para acceder a dicha área. En caso de que el solicitante posea algún activo, el sistema procede a verificar que se encuentre asociado al solicitante y en caso positivo se le concede el acceso.

Registrar acceso de personal ajeno a la institución: Cuando el visitante solicita el acceso por primera vez a la institución, el agente de seguridad busca los datos pertenecientes a la identificación personal en el sistema y de aparecer registrado se procede a tomar los restantes datos de dicho visitante emitiéndose una identificación con los permisos correspondientes a las áreas de su interés. En caso de que el visitante posea un vehículo se buscará también si se encuentra registrado en el sistema y en caso afirmativo se procederá a la captura de sus datos y emisión de la identificación correspondiente. Cuando el visitante ya esté registrado en el sistema podrá acceder a cada una de las áreas permitidas según la clasificación otorgada.

Para una mayor comprensión del proceso anteriormente expuesto se le brinda el modelo de procesos del negocio en el **Anexo 1**: Modelo de Procesos del Negocio**.**

PROPUESTA DE SOLUCIÓN

Para cumplir los objetivos propuestos al inicio de este trabajo se propone como solución una aplicación web desarrollada sobre ASP.NET Modelo-Vista-Controlador versión 4 que estará dividida en cuatro módulos lógicos fundamentales: Control de Acceso, Monitoreo, Administración y Configuración. La presente investigación se enfocará principalmente en los dos últimos.

La parte de control de acceso se encargará de la verificación automatizada de cada una de las identificaciones que soliciten acceder a un área determinada, donde dicha actividad será realizada por un agente de seguridad a través de un lector de código de barras. Este módulo contará además con un mecanismo de autorización el cual será el encargado de dictaminar si se concede el acceso o no.

El módulo de monitoreo servirá de herramienta principal para supervisar el comportamiento del control de acceso en el centro. A través de este se podrán obtener un grupo de reportes que aportarán una visión detallada del estado en el que se encuentra la institución.

Por otro lado la administración será la responsable de gestionar cada uno de los elementos que interactúan en el control de acceso. En este ámbito se podrán definir cuáles serán los recursos a los que se les desea controlar el acceso, así como qué roles o grupos de personas pueden acceder por los mismos. Contará además con la posibilidad de asignarle restricciones a los diferentes recursos de manera tal que a la hora de chequear el acceso se tengan en cuenta otro tipo de condiciones como puede ser la fecha y hora de entrada o salida.

Finalmente el módulo configuración será el escenario en el que se podrá establecer la estructura de la organización a partir de la definición de cada uno de los tipos de recursos que posee la misma, así como los atributos adicionales con que constarán los recursos y las personas. Se podrá gestionar además los dispositivos con los que se llevará a cabo el control de acceso, así como definir la conexión a datos y la gestión de cuentas de usuarios del sistema.

INFORMACIÓN QUE SE MANEJA

A continuación se explicarán los principales conceptos asociados a los procesos del negocio de control de acceso existente en la Universidad de las Ciencias Informáticas:

Acceso: Se refiere a la entrada o salida de determinado personal, vehículo o medio a un área determinada en la institución.

Punto de Acceso: Representa el espacio concerniente a un área determinada en la institución por el cual se produce el acceso.

Permiso: Es la autorización otorgada a un individuo, medio o vehículo para acceder a un área determinada.

Pase: Documento impreso que recoge los datos del visitante.

Libro de Visitas: Volumen que contiene las solicitudes de visitas y su estado.

Solapín: Documento de identificación que recoge los datos de determinada persona perteneciente a la institución.

Modelo de Dominio

Un modelo del domino captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema [41].

Principales Conceptos del Modelo de Dominio

Recurso: Son las diferentes instalaciones en las que se estructura una organización y se les requiere controlar el acceso. Los recursos pueden estar conformados por otros recursos. Ejemplo: Docente 1, Manzana 10, Complejo Comedor 2, Puerta Principal, Aula 1303.

Tipo de Recurso: Nomencladores en los que se clasifican los recursos de una organización. Ejemplo: Zonas, Edificios, Locales, Puertas, Puntos de Acceso.

Atributo: Parámetros adicionales que requiera un recurso o persona para ayudar en su descripción.

Dispositivo: Elementos de hardware utilizados en el control de acceso que están presentes en los recursos de la organización. Ejemplo: Lector de Código de Barras, Lector de Huellas.

Grupo: Representan una estructura lógica en la que se encuentra organizado el personal. Los grupos pueden ser subgrupos de un grupo superior. Ejemplo: Facultad 1, Departamento de Identificación, Grupo 1504.

Rol: Representa las funciones que puede llevar a cabo el personal de la organización. Los roles pueden heredar de otros roles. Ejemplo: Cocinero, Profesor, Estudiante.

Persona: Representa cualquier individuo que forme parte de la organización o esté vinculada a la misma. Ejemplo: Juan, Pedro, María.

Restricción: Representan las reglas que podrán imponerse a los recursos para controlar las condiciones en que los roles y/o grupos pueden acceder a estos. Definen las circunstancias en las que se debe producir un acceso.

Acceso: Representa la evidencia de que una persona determinada entró o salió de cierto recurso.

Activo: Son todos aquellos medios que pueden venir acompañando a una persona en el momento de realizar un acceso.

Infracción: Representa la evidencia de un acceso no autorizado.

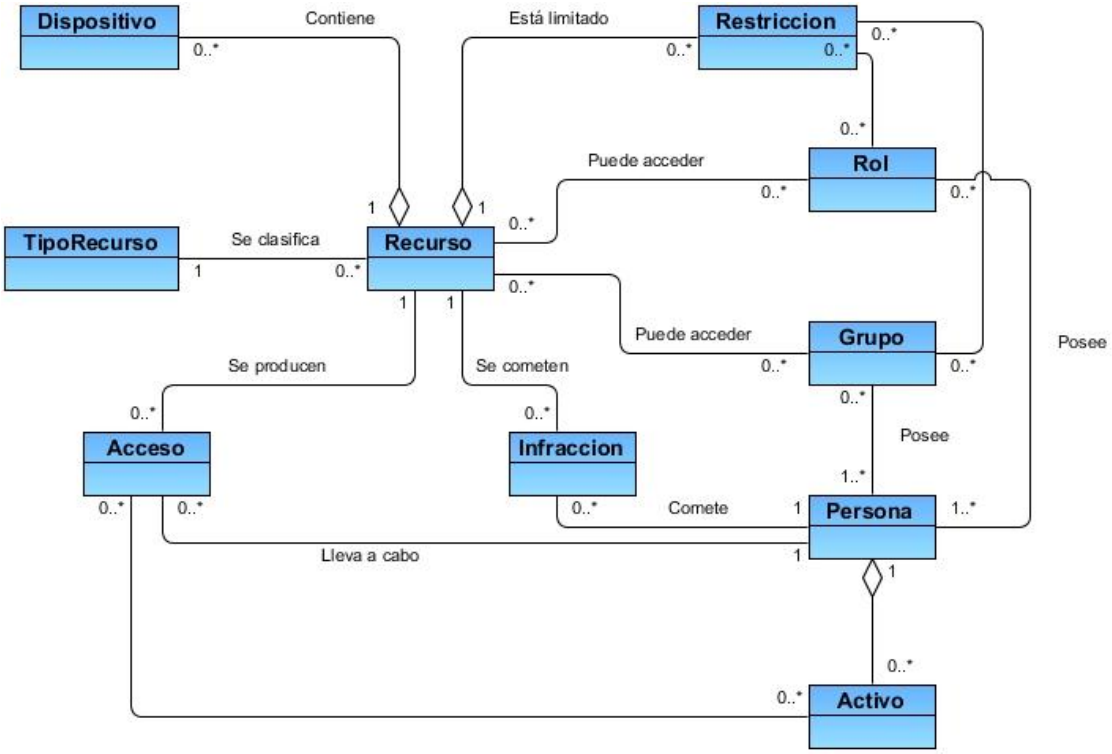


Ilustración 2: Modelo de Dominio

ESPECIFICACIÓN DE LOS REQUISITOS DE SOFTWARE

Un requisito es una condición o capacidad necesaria para que un usuario resuelva un problema o alcance un objetivo [42]. A continuación se describen los requisitos funcionales y no funcionales del sistema.

Requisitos Funcionales

Los requisitos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema. Para cumplir con los objetivos propuestos se plantean las siguientes funcionalidades:

Módulo Configuración

RF-1. **Autenticar Usuario del Sistema.**

RF-2. **Gestionar Cuenta Usuario del Sistema.**

2.1. Adicionar Cuenta Usuario del Sistema.

2.2. Eliminar Cuenta Usuario del Sistema.

2.3. Modificar Cuenta Usuario del Sistema.

2.4. Mostrar Cuenta Usuario del Sistema.

2.5. Listar Cuenta Usuario del Sistema.

2.6. Activar/Desactivar Cuenta Usuario del Sistema.

RF-3. **Definir Conexión de Acceso a Datos.**

RF-4. **Gestionar Dispositivo.**

4.1. Adicionar Dispositivo.

4.2. Eliminar Dispositivo.

4.3. Modificar Dispositivo.

4.4. Mostrar Dispositivo.

4.5. Listar Dispositivo.

4.6. Activar/Desactivar Dispositivo.

RF-5. **Gestionar Nomenclador Tipo de Recurso.**

5.1. Adicionar Nomenclador Tipo de Recurso.

5.2. Eliminar Nomenclador Tipo de Recurso.

5.3. Modificar Nomenclador Tipo de Recurso.

5.4. Mostrar Nomenclador Tipo de Recurso.

5.5. Listar Nomenclador Tipo de Recurso.

5.6. Activar/Desactivar Nomenclador Tipo de Recurso.

RF-6. **Gestionar Nomenclador Atributo.**

6.1. Adicionar Nomenclador Atributo.

6.2. Eliminar Nomenclador Atributo.

6.3. Modificar Nomenclador Atributo.

6.4. Mostrar Nomenclador Atributo.

6.5. Listar Nomenclador Atributo.

6.6. Activar/Desactivar Nomenclador Atributo.

Módulo Administración

RF-7. **Gestionar Recurso.**

7.1. Adicionar Recurso.

7.2. Eliminar Recurso.

7.3. Modificar Recurso.

7.4. Mostrar Recurso.

7.5. Listar Recurso.

7.6. Asignar/Desasignar Restricciones.

7.7. Asignar/Desasignar Dispositivos.

7.8. Activar/Desactivar Recurso.

RF-8. **Gestionar Rol.**

8.1. Adicionar Rol.

8.2. Eliminar Rol.

8.3. Modificar Rol.

8.4. Mostrar Rol.

8.5. Listar Rol.

8.6. Asignar/Desasignar Recursos.

8.7. Activar/Desactivar Rol.

RF-9. **Gestionar Grupo.**

9.1. Adicionar Grupo.

9.2. Eliminar Grupo.

9.3. Modificar Grupo.

9.4. Mostrar Grupo.

9.5. Listar Grupo.

9.6. Asignar/Desasignar Recursos.

9.7. Activar/Desactivar Grupo.

RF-10. **Gestionar Persona.**

10.1. Adicionar Persona.

10.2. Eliminar Persona.

10.3. Modificar Persona.

10.4. Mostrar Persona.

10.5. Listar Persona.

10.6. Asignar/Desasignar Roles.

10.7. Asignar/Desasignar Grupos.

10.8. Asignar/Desasignar Atributos.

10.9. Activar/Desactivar Persona.

RF-11. **Gestionar Restricción.**

11.1. Adicionar Restricción.

11.2. Eliminar Restricción.

11.3. Modificar Restricción.

11.4. Mostrar Restricción.

11.5. Listar Restricción.

11.6. Asignar/Desasignar Roles.

11.7. Asignar/Desasignar Grupos.

11.8. Activar/Desactivar Restricción.

RF-12. **Gestionar Activo.**

12.1. Adicionar Activo.

12.2. Eliminar Activo.

12.3. Modificar Activo.

12.4. Mostrar Activo.

12.5. Listar Activo.

12.6. Activar/Desactivar Activo.

Generales

RF-13. **Buscar Elemento Actual.**

RF-14. **Filtrar Elemento Actual.**

RF-15. **Ordenar Elemento Actual.**

Descripción de Funcionalidades

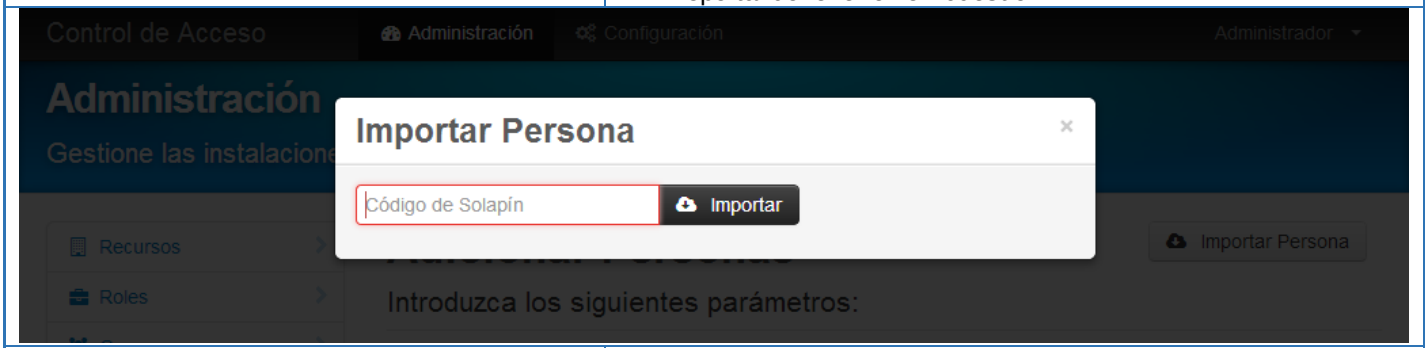
La metodología de desarrollo FDD utilizada para la realización del sistema, plantea que los requisitos funcionales deben ser divididos en subconjuntos, en dependencia de la relación que exista entre ellos, y luego se describen detalladamente para lograr un mejor entendimiento por parte del equipo de desarrollo. La Tabla 2 posee la descripción de la funcionalidad Adicionar Persona perteneciente al requisito funcional Gestionar Persona. El resto de las descripciones están recogidas en el artefacto propuesto por la metodología FDD, llamado Descripción de Requisitos de Software.

PRECONDICIONES	**El usuario debe estar previamente autenticado en el sistema.**
FUNCIONALIDADES ASOCIADAS	RF10
CONCEPTOS TRATADOS	Persona

DESCRIPCIÓN BÁSICA	1. Si el usuario autenticado selecciona la opción Adicionar Persona, el sistema muestra una interfaz que contiene los parámetros necesarios para adicionar una persona. **Parámetros necesarios:** • Nombre. • Apellidos. • Carnet de Identidad. • Solapín. • URL de la foto. **Opciones:** • Adicionar. • Importar Persona. • Cancelar. 1.1. Si el usuario autenticado presiona la opción Adicionar, el sistema valida que cada uno de los parámetros requeridos están escritos correctamente. 1.1.1. Si los datos introducidos son correctos, el sistema adiciona la nueva persona y muestra un mensaje de operación exitosa. (En caso contrario ver Descripción Alterna 1). 1.2. Si el usuario autenticado presiona la opción Importar Persona, el sistema muestra una ventana que contiene un campo necesario para llevar a cabo esta acción. **Campo necesario:** • Solapín. 1.2.1. Si el usuario autenticado presiona la opción Importar, el sistema valida la correcta escritura del Solapín y procede a importar la persona desde una fuente externa, mostrando un mensaje de operación exitosa. (En caso contrario ver Descripción Alterna 2). 1.3. Si el usuario autenticado presiona la opción Cancelar, el sistema desase cualquier cambio y redirecciona a la interfaz principal.

DESCRIPCIÓN ALTERNA	
Descripción alterna 1:	1.1. Si existen campos obligatorios vacíos el sistema muestra un mensaje reportando que los parámetros son obligatorios. 1.2. Si los datos introducidos contienen errores el sistema muestra un mensaje reportando el error en cuestión. 1.3. Si el proceso de adición no es completado correctamente, el sistema muestra un mensaje de error con la causa correspondiente.

Descripción alterna 2:	2.1. Si el campo se encuentra vacío, el sistema indica al usuario que el mismo es requerido. 2.2. Si los datos son incorrectos, el sistema muestra un mensaje reportando el error en cuestión.



VALIDACIONES	
	• Todos los parámetros son obligatorios. • El parámetro “Nombre” debe contener solo letras y su longitud debe estar entre 2 y 25 caracteres. • El parámetro “Apellidos” debe contener solo letras y su longitud debe estar entre 5 y 50 caracteres. • El parámetro “Carnet de Identidad” debe contener solo números y su longitud debe ser de 11 caracteres. • El parámetro “Solapín” debe contener solo números y letras en mayúscula, y su longitud debe estar entre 5 y 7

	caracteres. • El parámetro "URL de la foto" debe poseer la estructura de una URL.
POSTCONDICIONES	La persona adicionada de quedar registrada en la base de datos y aparecer en la lista de personas.

Tabla 2: Descripción de la funcionalidad Adicionar Persona, Gestionar Persona

Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Estas características son las que permiten al producto ser atractivo, usable, rápido, confiable, entre otras.

Requisitos de Software para estaciones clientes

Sistema Operativo Windows XP SP3 o superior con antivirus actualizado, navegador web Mozilla Firefox v17.0 o superior, Google Chrome v20.0 o superior.

Requisitos de Hardware para estaciones clientes.

PC Pentium 4 a 1 GHz o superior, mínimo 512 MB de RAM, 40 GB o superior de disco duro.

Otras: Local climatizado de ser posible.

Requisitos de Software para estaciones servidores.

Sistema Operativo Windows Server 2008 R2 con SP2, antivirus actualizado, Microsoft .Net Framework v4.0. Internet Information Server 7.5 y PostgreSQL 9.1.

Requisitos de Hardware para estaciones servidores.

PC Pentium 4 a 2 GHz o superior, mínimo 2 GB de RAM, 250 GB o superior de disco duro.

Otras: Local climatizado obligatoriamente.

Requisitos de Usabilidad.

RnF-1: El sistema podrá ser utilizado por cualquier usuario con las siguientes características:

a) Conocimientos básicos relativos al uso de una computadora.
b) Conocimientos básicos del sistema operativo Windows.
c) Conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.

RnF-2: El sistema será distribuido en idioma español.

RnF-3: Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo.

RnF-4: El sistema poseerá estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación.

a) Menús laterales y desplegables que permitan el acceso rápido a la información.
b) Menú de soporte que facilite el acceso a herramientas utilitarias, notificaciones del sistema y ayuda integrada.

Requisitos de Fiabilidad.

RnF-1: El sistema debe estar disponible las 24 horas del día los 7 días de la semana realizándose copias de seguridad diariamente tanto de la aplicación como de la base de datos.

RnF-2: El tiempo medio entre fallos no debe ser superior a 1 hora.

RnF-3: El sistema estará 1 hora fuera de operación luego de haber fallado 3 veces.

Requisitos de Eficiencia.

RnF-1: El sistema debe ser capaz de soportar una cantidad escalable de peticiones y dar respuestas efectivas y rápidas.

RnF-2: El sistema debe hacer un uso eficiente de los recursos de hardware donde se encuentre desplegado.

Requisitos de Soporte.

RnF-1: El soporte de las aplicaciones y componentes utilizados en el desarrollo del sistema será responsabilidad de los proveedores de los mismos.

RnF-2: El sistema dispondrá de un año de soporte técnico donde se corregirán las fallas que atenten contra el buen funcionamiento del sistema.

Restricciones de diseño.

RnF-1: Plataforma de desarrollo .NET 4.0 utilizando Visual Studio Ultimate 2010.

RnF-2: Biblioteca jQuery.

RnF-3: Para el acceso a datos se utilizará el ORM NHibernate 3.1.0.

Requisitos de Interfaz.

RnF-1: El sistema dispondrá de un diseño ameno e intuitivo acorde las características de la institución.

Clasificación de Funcionalidades

Las funcionalidades definidas y organizadas por módulos de acuerdo a su propósito se clasifican en críticas y secundarias, siendo las funcionalidades críticas las que constituyen mayor impacto en el negocio por lo que son las primeras en ser diseñadas y construidas en cada iteración de desarrollo. Luego se desarrollan las funcionalidades secundarias las cuales sirven al flujo principal del proceso. En la siguiente tabla se muestra la clasificación de las funcionalidades de los módulos configuración y administración, así como las funcionalidades generales.

Módulo	Funcionalidad	Clasificación
CONFIGURACIÓN	Autenticar Usuario del Sistema.	Crítica
	Gestionar Cuenta de Usuario del Sistema.	Crítica
	Definir Conexión de Acceso a Datos.	Crítica
	Gestionar Dispositivos.	Secundaria
	Gestionar Nomenclador Tipo de Recurso.	Crítica
	Gestionar Nomenclador Atributo.	Crítica
ADMINISTRACIÓN	Gestionar Recurso.	Crítica
	Gestionar Activo.	Secundaria
	Gestionar Rol.	Secundaria

	Gestionar Grupo.	Secundaria
	Gestionar Persona.	Crítica
	Gestionar Restricción.	Crítica
GENERALES	Buscar Elemento Actual.	Secundaria
	Filtrar Elemento Actual.	Secundaria
	Ordenar Elemento Actual.	Secundaria

Tabla 3: Clasificación de Funcionalidades

Planeación de las Iteraciones

A partir de la lista de funcionalidades anteriormente clasificadas, se confecciona un cronograma de diseño y construcción. La siguiente tabla muestra el cronograma para el diseño y desarrollo de las funcionalidades vinculadas al proceso de negocio.

Iteración	Módulo	Funcionalidad	Fecha de ejecución	
			Diseño	Construcción
1	CONFIGURACIÓN	Autenticar Usuario del Sistema.	05/01/13--15/01/13	16/01/13--21/01/13
		Gestionar Cuenta de Usuario del Sistema.	05/01/13--15/01/13	21/01/13--26/01/13
		Definir Conexión de Acceso a Datos.	05/01/13--15/01/13	26/01/13--31/01/13
		Gestionar Dispositivos.	05/01/13--15/01/13	01/03/13--06/03/13
		Gestionar Nomenclador Tipo de Recurso.	05/01/13--15/01/13	01/02/13--5/02/13
		Gestionar Nomenclador Atributo.	05/01/13--15/01/13	06/02/13--11/02/13
2	ADMINISTRACIÓN	Gestionar Recurso.	11/02/13--21/02/13	12/02/13--17/02/13
		Gestionar Activo.	11/02/13--21/02/13	07/03/13--12/03/13
		Gestionar Rol.	11/02/13--21/02/13	13/03/13--18/03/13
		Gestionar Grupo.	11/02/13--21/02/13	19/02/13--24/03/13
		Gestionar Persona.	11/02/13--21/02/13	18/02/13--23/02/13
		Gestionar Restricción.	11/02/13--21/02/13	24/02/13--01/03/13
3	GENERALES	Buscar Elemento Actual	16/03/13--21/03/13	25/03/13--30/03/13
		Filtrar Elemento Actual	16/03/13--21/03/13	31/03/13--4/04/13
		Ordenar Elemento Actual	16/03/13--21/03/13	05/04/13--10/04/13

Tabla 4: Cronograma de Diseño y Construcción

ARQUITECTURA PROPUESTA

La arquitectura de software, se define como la organización fundamental de un sistema representada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. La misma involucra un conjunto de decisiones significativas acerca de la organización del

sistema, selecciona sus elementos estructurales y sus interfaces, así como su comportamiento. También involucra funcionalidad, usabilidad, tolerancia a cambios, rendimiento, reutilización y aspectos estéticos.
Los Patrones Arquitectónicos son los que definen la estructura de un sistema, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes, con el objetivo de facilitar la tarea del diseño de dicho sistema [43].

Especificación de la Arquitectura

El sistema a desarrollar sobre la plataforma .Net, en su versión 4.0, es una solución cliente - servidor que define la relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a la otra (servidor) y este último le envía las respuestas. Los principales componentes del sistema serian: un conjunto de servidores locales que brindan servicios a otras aplicaciones, un conjunto de clientes que realizan peticiones a los servidores y una red que permite la conexión entre servidores y clientes.
Se hace uso del estilo arquitectónico en capas, el cual se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Este estilo permite asignar correctamente las funcionalidades a cada capa, pudiéndose reutilizar las capas inferiores que no tengan dependencias con las superiores, proporcionando un desacople entre las capas ya que la comunicación entre las mismas está basada en la abstracción, evitando que los cambios en una de ellas afecte directamente al resto.

Patrón de arquitectura Modelo Vista Controlador

El patrón de arquitectura de software Modelo Vista Controlador (*Model View Controller*, MVC) separa los datos de una aplicación, la interfaz de usuario y la lógica de control. Este patrón de llamada y retorno se puede apreciar en aplicaciones donde las peticiones que realiza el usuario son atendidas por un controlador, el cual define las vistas a mostrar y esta a su vez posee los mecanismos para visualizar el estado del modelo. Ver **Ilustración 3**.
MVC separa la interfaz de usuario de una aplicación en tres aspectos principales:

- El ***Modelo:*** un conjunto de clases que describen los datos con los que trabaja, así como las reglas de negocio sobre cómo los datos pueden ser cambiados y manipulados.
- La ***Vista:*** define cómo la interfaz de usuario se mostrará.
- El ***Controlador:*** un conjunto de clases que maneja la comunicación por parte del usuario, el flujo general de la aplicación y la lógica específica de la misma [44].

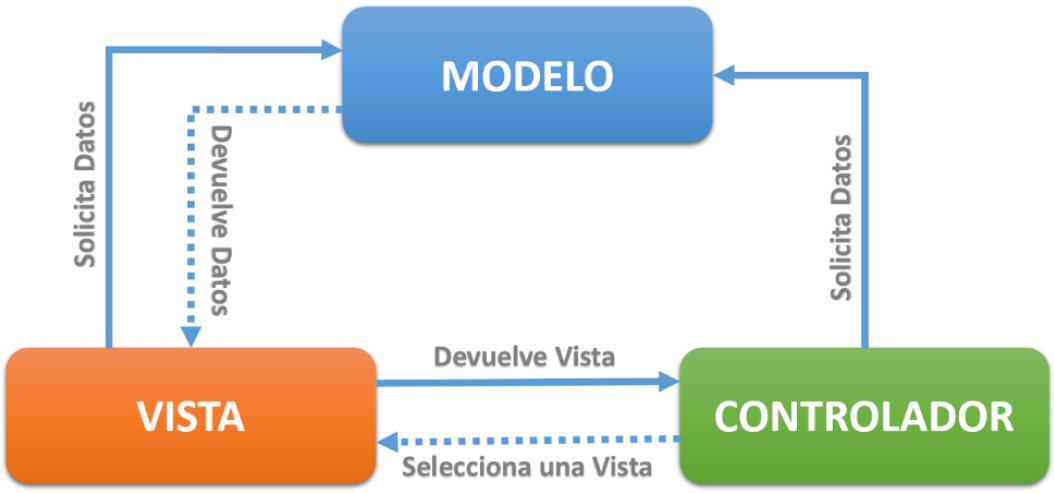


Ilustración 3: Patrón Arquitectónico Modelo-Vista-Controlador

En la solución propuesta se utiliza dicho patrón, permitiendo así la definición de la estructura de la capa de presentación en la arquitectura utilizada y administrando la manera en que los datos son mostrados al usuario. En el caso de dicha solución el modelo lo representan las clases que contienen todas las validaciones necesarias para que los datos pasados a través de las vistas sean aprobados, de esta manera se utilizan las facilidades brindadas por ASP.Net MVC 4 para validaciones llamadas DataAnnotations[10], referentes a validaciones predefinidas por el framework y además posibilita la inserción o implementación de anotaciones nuevas. Por otra parte las vistas son las interfaces que se muestran al usuario, dentro de las cuales se utilizan técnicas para mostrar interfaces a partir de objetos adquiridos desde librerías. El controlador simplemente se encarga de manejar todo el flujo de información entre el modelo y las vistas.

Distribución lógica del sistema

El sistema se encuentra lógicamente dividido en cinco capas definiendo claramente las responsabilidades de cada una, tal como se muestra en la **Ilustración 4**. De esta manera se reduce el acoplamiento y se aumentar la reutilización de las mismas. Esta estructura permite la realización de cambios en las capas sin realizar grandes modificaciones en las demás. La comunicación entre las capas se realizará a nivel de interfaces permitiendo trabajar de manera transparente a las instancias reales.

[10] **DataAnnotations**: Es un espacio de nombres que proporciona las clases de atributos que se utilizan para definir los metadatos de los controles de datos de ASP.NET MVC y ASP.NET.

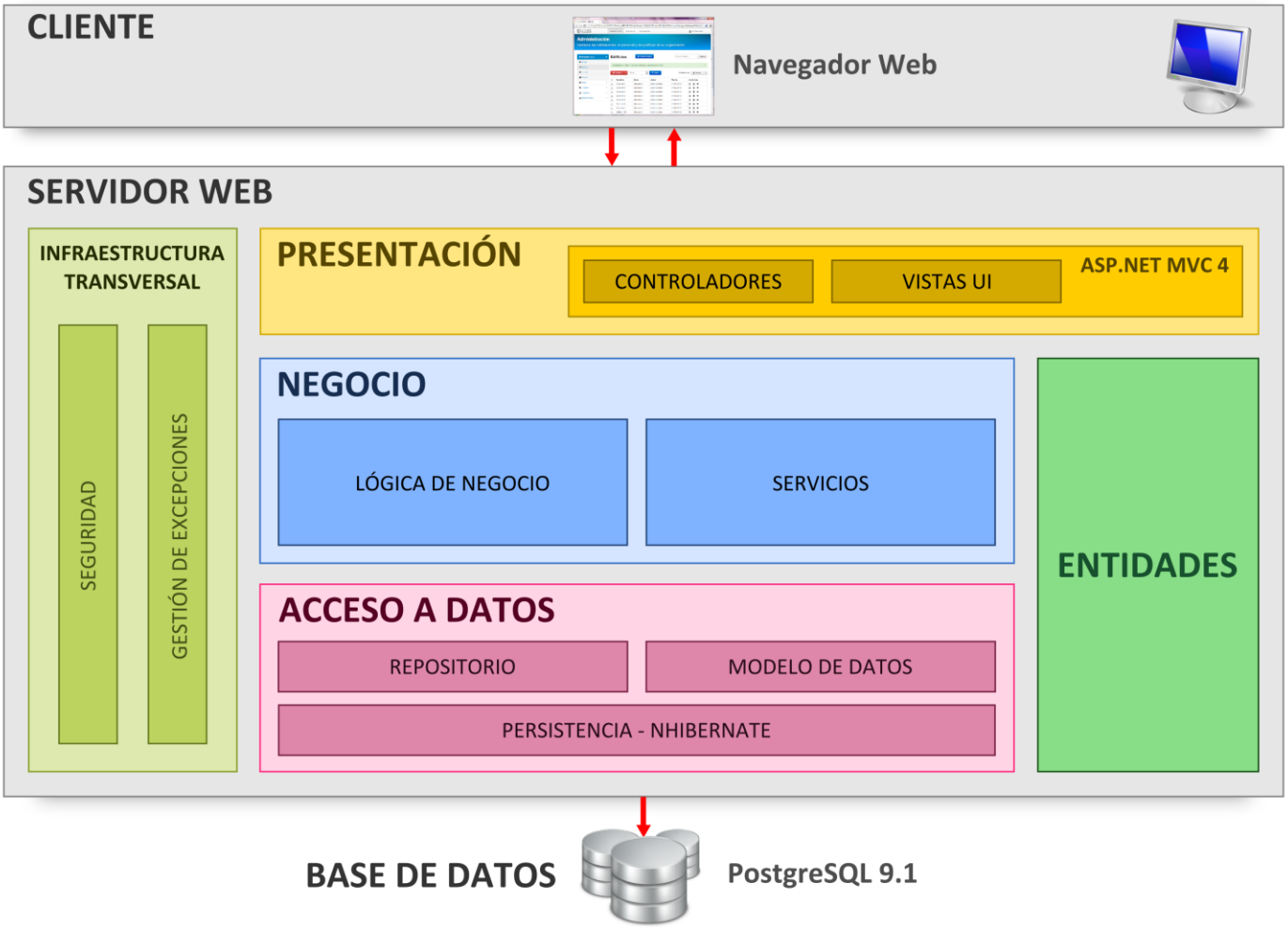


Ilustración 4: Vista Lógica de la Arquitectura del Software

Capa de Presentación

Es la capa donde el sistema interactúa con el usuario, haciendo uso de varias tecnologías para la validación de los datos de entrada así como el uso de componentes. En esta capa se encuentran todas las interfaces que serán mostradas a los usuarios utilizando el patrón arquitectónico MVC donde están presentes los elementos necesarios para su correcto funcionamiento, entre los cuales se pueden citar: los ficheros de código JavaScript, que dan paso a la integración con los componentes de JQuery; así como los archivos CSS[11] que contienen los estilos de la aplicación. Es típico encontrar MVC a través de tres capas pero en este caso se encuentra solo en la capa de presentación donde los controladores representan la lógica de la presentación, las vistas representan las interfaces de usuario y el modelo representa los datos de la presentación que son el resultado del procesamiento en la capa de negocio.

[11] **CSS**: Cascading Style Sheets, es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML.

Esta capa de presentación se encuentra representada por el componente *PMICA.ControlAcceso.WepApp* de la aplicación y tiene una interacción directa con la capa de negocio.

Capa de Negocio

En esta capa se recogen todas las funcionalidades necesarias para darle solución a los requerimientos del negocio. Las funcionalidades se encuentran definidas según el contexto en el que se desenvuelven. Tienen la responsabilidad de manejar todas las operaciones sobre una entidad en específico, así como todas las entidades que por conceptos de composición se encuentran relacionadas con esta. La capa de negocio está constituida por el componente *PMICA.ControlAcceso.Business* y tiene relación directa con las capas: Acceso a Datos y Entidades.

Capa de Acceso a Datos

Es el componente que da soporte a las funcionalidades de la capa de negocio que se encuentran relacionadas con la fuente de datos. En esta capa se encuentra incluido el ORM NHibernate utilizado para la generación del modelo de datos y se encarga de la manipulación de la información en la base de datos. Las facilidades que brinda este modelo son aprovechadas por la implementación del Patrón Repositorio que a su vez se abstrae de las dependencias del NHibernate de la base de datos y posibilita un mejor y fácil trabajo con los datos. La principal función de esta capa es realizar una implementación de las funcionalidades definidas en las interfaces de la capa de negocio y al mismo tiempo trabajar directamente con la fuente de datos. Esta capa está constituida por los siguientes componentes, además de tener relación con la capa de Entidades y la Base de Datos. *PMICA.ControlAcceso.DAL*, *PMICA.ControlAcceso.Repository, PMICA.ControlAcceso.NHibernate, PMICA.ControlAcceso.DataModel.*

Capa de Entidades

Contiene las clases entidades del componente que se gestionan en la aplicación, persisten en la base de datos y se muestran en la presentación. Esta capa se encuentra constituida por el componente *PMICA.ControlAcceso.Entities.*

Capa Infraestructura Transversal

Es la capa que agrupa las funcionalidades que necesitan estar presentes a todo lo largo de la aplicación y su lógica debe ser reutilizable. Específicamente será la encargada de garantizar la seguridad del sistema y llevar a cabo la gestión de excepciones.

Base de Datos

La fuente de datos está constituida por las tablas que permiten el almacenamiento de la información recolectada y procesada utilizando como sistema gestor PostgreSQL 9.1.

DIAGRAMAS DE CLASES DEL DISEÑO

Este diagrama muestra la estructura de las clases que luego serán descritas en un lenguaje de programación las cuales satisfacen los detalles de la implementación. En la **Ilustración 5** se muestra el diagrama de clases del diseño correspondiente al requisito funcional gestionar recurso. El resto de las clases que interactúan en el diseño del sistema podrán ser vistas en cada una de sus capas correspondientes en el **Anexo 2**: Diagramas de Clases del Diseño**.**

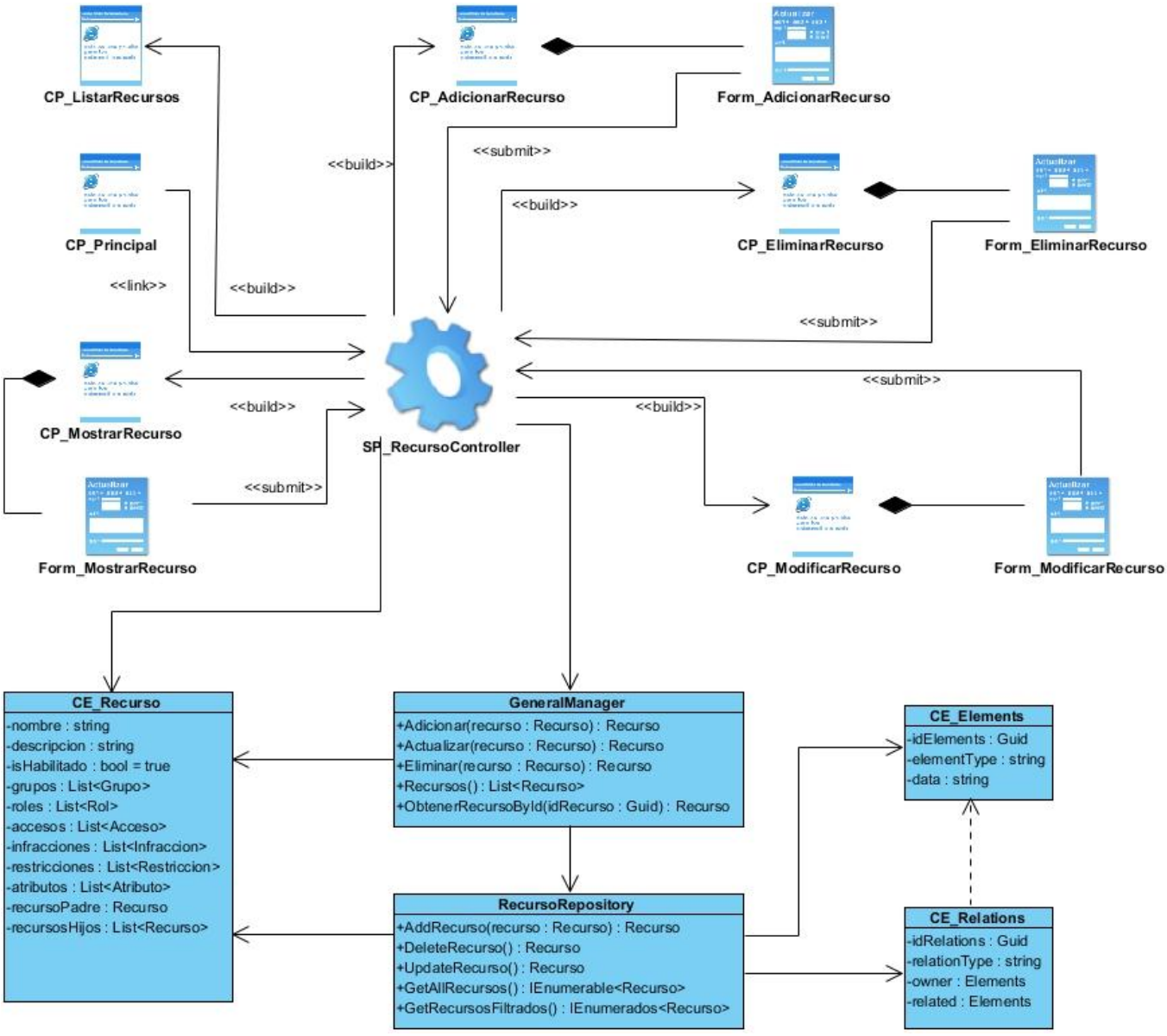


Ilustración 5: Diagrama de Clases del Diseño para la funcionalidad Gestionar Recurso

Diagramas de Secuencia

Los diagramas de secuencia muestran gráficamente las interacciones del actor y de las operaciones a que dan origen, estos se elaboran durante la fase de análisis de un ciclo de desarrollo. Su creación depende de la formulación previa de los casos de uso, el comportamiento del sistema es una descripción de lo que hace, y no cómo lo que hace. De manera general dichos diagramas muestran un determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema [45].

En la **Ilustración 6** se muestra el diagrama de secuencia del requisito funcional gestionar recurso específicamente el escenario modificar recurso. El resto de los diagramas podrán ser vistos en el **Anexo 3**: Diagramas de Secuencia.

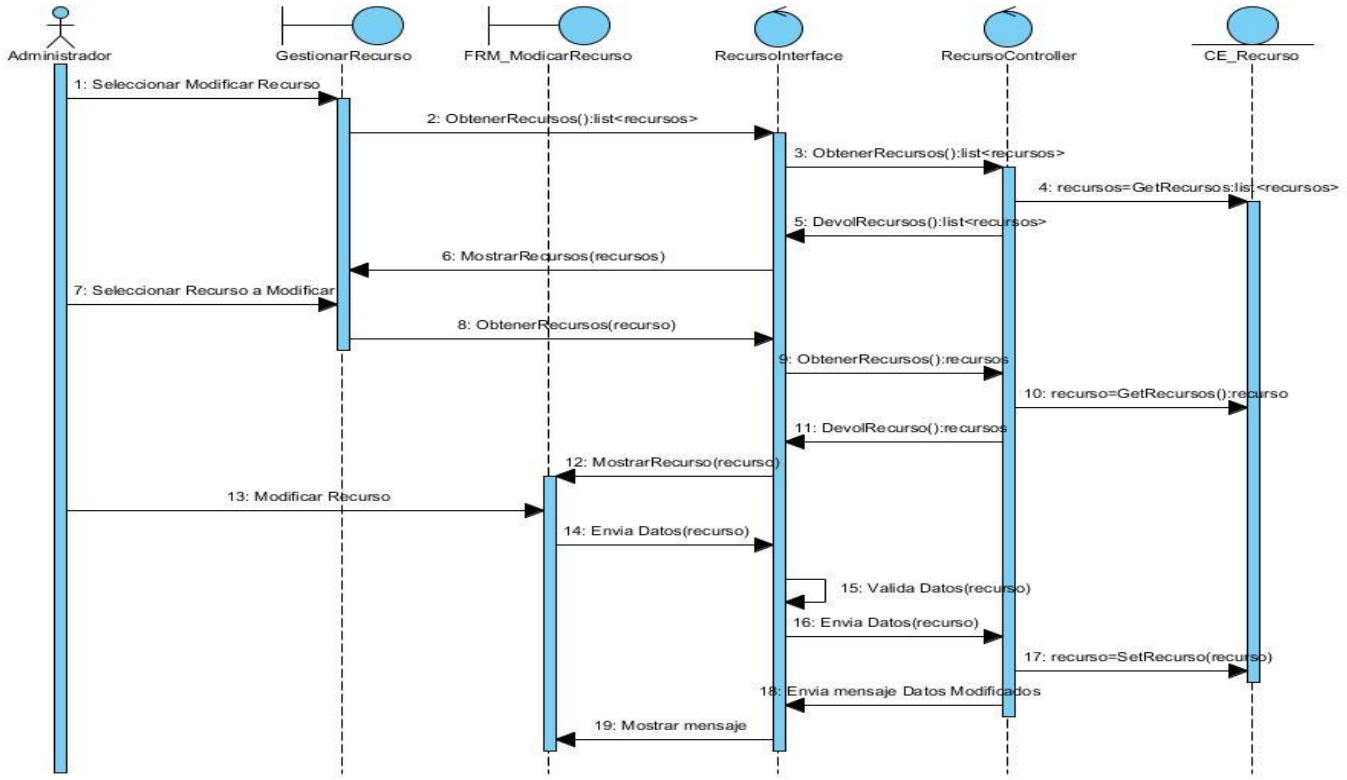


Ilustración 6: Diagrama de Secuencia Modificar Recurso

Patrones de Diseño

Un patrón de diseño es una descripción de un problema y su solución que recibe un nombre y se puede emplear en otros contextos. En teoría indica la manera de utilizarlo en circunstancias diversas. Dentro de los patrones de diseño se encuentran dos grupos fundamentales conocidos por Patrones Generales de Asignación de Responsabilidades de Software (*General Responsibility Assignment Software Patterns*, GRASP) y Banda de los Cuatro (*Gang of Four*, GOF).

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos. Los patrones GOF describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos para ayudar a realizar tareas complejas [46]. A continuación se analizará cada uno de los patrones utilizados para la implementación de la solución y un ejemplo de su utilización en la aplicación.

Patrón Experto

El Patrón Experto perteneciente al grupo de patrones GRASP consiste en asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Con este patrón se pretende que los objetos hagan cosas relacionadas con la información que poseen.

```
public class Persona : Elemento
{
    private string _nombre;

    [Required]
    [StringLength(25, ErrorMessage = "La longitud de la cadena de estar entre 2 y 25 caracteres.", MinimumLength = 2)]
    [Display(Name = "Nombre:")]
    [DataType(DataType.Text)]
    [RegularExpression("[^0-9]*", ErrorMessage = "Solo se permiten letras.")]
    public string Nombre
    {
        get
        {
            _nombre = _reader.GetAttributeValueBySchema(Data, "Persona", "nombre");
            return _nombre;
        }
        set
        {
            _nombre = value;
            Data = _writer.AddAttributeValueBySchema(Data, "Persona", "nombre", value);
        }
    }
```

Ilustración 7: Ejemplo Patrón Experto

Patrón Creador

El Patrón Creador perteneciente al grupo de patrones GRASP se basa en asignarle a la clase B la responsabilidad de crear una instancia de la clase A. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos y tiene como propósito fundamental encontrar un creador que se debe conectar con el objeto producido en cualquier evento [46].

```
public abstract class ManagerRepository
{
    private ElementsRepository _elements;
    private RelationsRepository _relations;

    protected ManagerRepository(ISession session)
    {
        _elements = new ElementsRepository(session);
        _relations = new RelationsRepository(session);
    }
```

Ilustración 8: Ejemplo Patrón Creador

Patrón Controlador

El Patrón Controlador perteneciente al grupo de patrones GRASP se encarga de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. El patrón propone el diseño de clases con la responsabilidad de controlar el flujo de eventos del sistema a clases específicas [46].

```
public class PersonaController : Controller
{
    private readonly GeneralManager _generalManager = new GeneralManager();

    public ActionResult Index(int page = 1)...

    public ActionResult Details(Guid id)...

    public ActionResult Create()...

    [HttpPost]
    public ActionResult Create(FormCollection collection)...

    public ActionResult Edit(Guid id)...

    [HttpPost]
    public ActionResult Edit(Guid id, FormCollection collection)...

    public ActionResult Delete(Guid id, string returnurl)...

    [HttpPost]
    public ActionResult Delete(Guid id, FormCollection collection)...
```

Ilustración 9: Ejemplo Patrón Controlador

Patrón Alta Cohesión

El Patrón Alta Cohesión perteneciente al grupo de patrones GRASP se basa en asignar una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que realicen un trabajo enorme. El patrón propone el diseño de clases con responsabilidades moderadas en su área funcional y que colabore con las otras para llevar a cabo una tarea [46].

```
public class GenericRepository<TEntity> : Interfaces.IRepository<TEntity> where TEntity : class, new()
{
    public readonly ISession _session;

    public IQueryable<TEntity> GetAll()...

    public virtual IEnumerable<TEntity> Update(IEnumerable<TEntity> items)...

    public virtual TEntity Update(TEntity item)...

    public virtual IEnumerable<TEntity> Delete(IEnumerable<TEntity> items)...

    public virtual TEntity Delete(TEntity item)...

    public virtual IEnumerable<TEntity> Add(IEnumerable<TEntity> items)...

    public virtual TEntity Add(TEntity item)...
```

Ilustración 10: Ejemplo Patrón Alta Cohesión

Patrón Bajo Acoplamiento

El Patrón Bajo Acoplamiento perteneciente al grupo de patrones GRASP consiste en asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras, con las que conoce y con que recurre a ellas. El patrón propone el

diseño de clases más independientes, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas [46].

Patrón Repositorio

El Patrón Repositorio es un patrón estructural perteneciente al grupo de patrones GOF que permite de una forma sencilla, hacer que las capas de datos se puedan probar y trabajar de una forma más simétrica a la orientación a objetos con los modelos relacionales. Un repositorio realiza las tareas de intermediario entre las capas de modelo de dominio y mapeo de datos. Los objetos clientes construyen de forma declarativa consultas y las envían a los repositorios para que las satisfagan. Conceptualmente, un repositorio encapsula a un conjunto de objetos almacenados en la base de datos y las operaciones que sobre ellos pueden realizarse. Para cada tipo de objeto lógico que necesite acceso global, se debe crear un objeto (Repositorio) que proporcione la apariencia de una colección en memoria de todos los objetos de ese tipo. Se debe establecer el acceso mediante una interfaz bien conocida, proporcionar métodos para añadir y eliminar objetos, que realmente encapsularán la inserción o eliminación de datos en el almacén de datos. Proporcionar métodos que seleccionen objetos basándose en ciertos criterios de selección y devuelvan objetos o colecciones de objetos instanciados (entidades del dominio) con los valores de dicho criterio, de forma que encapsule el almacén real (base de datos) y la tecnología base de consulta.

```
namespace PMICA.ControlAcceso.Repository.Implements
{
    public class ElementsRepository : GenericRepository<Elements>, IElementsRepository
    {
        public ElementsRepository(ISession session) : base(session)...
        public Elements GetElementById(Guid idElement)...
        public List<Elements> GetElementsByType(string type)...
        public string GetTypeById(Guid idElement)...
        public string GetDataById(Guid idElement)...
        public string GetElementTagById(Guid id, string tag)...
```

Ilustración 11: Ejemplo del Patrón Repositorio

Patrón Fachada

El Patrón Fachada es un patrón estructural perteneciente al grupo de patrones GOF el cual provee una interfaz unificada y sencilla que funciona de intermediaria entre un cliente y una interfaz o grupos de interfaces más complejas. Este se caracteriza por simplificar el acceso a un conjunto de clases proporcionando una única clase que todos utilizan para comunicarse con dicho conjunto de clases, además de reducir la complejidad y minimizar dependencias. Fachada conoce cuales clases de un subsistema son responsables de una petición y delega las peticiones de los clientes en los objetos del subsistema [47].

```
public interface IPersonaRepository
{
    Persona AddPersona(Persona persona);
    IEnumerable<Persona> AddPersona(IEnumerable<Persona> persona);
    Persona DeletePersona(Persona persona);
    IEnumerable<Persona> DeletePersona(IEnumerable<Persona> persona);
    Persona UpdatePersona(Persona persona);
    IEnumerable<Persona> UpdatePersona(IEnumerable<Persona> persona);
    IEnumerable<Persona> GetAllPersonas();
    IEnumerable<Persona> GetPersonasFiltrados(Func<Persona, bool> filter);
}
```

Ilustración 12: Ejemplo Patrón Fachada

Patrón Singleton

El Patrón Singleton es quizás el más sencillo de los patrones GOF y a su vez uno de los más conocidos y utilizados. Este patrón bien conocido como instancia única está diseñado para garantizar que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. La utilización de este patrón trae como consecuencia: acceso controlado a la única instancia ya que puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia. El Patrón Singleton es una mejora sobre las variables globales. [48].

```
public class NHibernateSessionManager
{
    private static ISessionFactory _sessionFactory;
    private static NHibernateSessionManager _instance;
    public static NHibernateSessionManager Instance
    {
        get { return _instance ?? (_instance = new NHibernateSessionManager()); }
    }
```

Ilustración 13: Ejemplo Patrón Singleton

Patrón Unidad de Trabajo

El Patrón Unidad de Trabajo (*Unit of Work*, UoW) mantiene un monitoreo de los objetos afectados por una transacción empresarial a través de una lista y coordina la escritura fuera de los cambios y la resolución de problemas de simultaneidad. Para una correcta comprensión del mismo, el UoW se ocupa de controlar los cambios que sufren las entidades, a partir de todas las modificaciones que se producen en las bases de datos luego de realizar una acción en las mismas [49].

```
public Elemento Adicionar(Acceso acceso)
{
    var unidadtrabajo = new UnitOfWork(NHibernateSessionManager.Instance.GetSessionFactory);
    var target = new AccesoRepository(unidadtrabajo);
    acceso = target.AddAcceso(acceso);
    if (acceso != null)
    {
        unidadtrabajo.Commit();
        return acceso;
    }
    return null;
}
```

Ilustración 14: Ejemplo Patrón Unidad de Trabajo

CONCLUSIONES

Durante la realización del capítulo que recién concluye se efectuó un profundo análisis del flujo actual de los procesos del negocio, el cual arrojó la situación en la que se encuentra la universidad a la hora de acceder a la misma, proporcionando así dicho análisis un amplio panorama de qué funcionalidades se deben desarrollar para lograr la satisfacción del cliente. Se describieron detalladamente los procesos que facilitaron definir las entradas y salidas de todas las actividades, así como los roles involucrados en la misma y un modelo de dominio que muestra la relación de todas las clases conceptuales necesarias para el sistema, en pos de lograr un correcto entendimiento para la etapa de desarrollo. También se definieron los requerimientos de software que darán respuesta a la problemática planteada, además de la arquitectura y patrones a utilizar con el fin de facilitar la implementación de dicha solución.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

INTRODUCCIÓN

El presente capítulo abarca la fase de implementación y pruebas que comienza con los resultados obtenidos en el diseño, continúa con la implementación del sistema en términos de componentes para terminar en la ejecución de las pruebas unitarias y de caja negra en pos de garantizar la calidad del mismo. Se obtendrán también los diferentes diagramas que propone la metodología de desarrollo en la fase de construcción, para facilitar el desarrollo de la aplicación. Se describe el modelo de datos y los estándares de codificación con el objetivo de lograr una mejor comprensión del código por parte del equipo de desarrollo.

MODELO DE DATOS

Un modelo de datos es un conjunto de conceptos que permiten describir los datos, las relaciones que existen entre ellos, la semántica y las restricciones de consistencia. Además es una representación lógica y física de los datos persistentes usados por la aplicación, permite describir:

- **Las Estructuras de Datos:** los tipos de datos y la forma en que se relacionan.
- **Las Restricciones de Integridad:** condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.
- **Operaciones de Manipulación de los Datos:** operaciones sobre la información en la base de datos.

El modelo de datos del sistema está formado por dos tablas, *Elements* (Elementos) y *Relations* (Relaciones) tal como se muestra en la **Ilustración 15**. Los elementos están compuestos por un identificador, un tipo y los datos. Los tipos de elementos así como los datos contenidos en un XML, están definidos por un XSD[12]. Por otra parte, la tabla *Relations* contiene las relaciones entre elementos; los tipos de relaciones están definidas en un XSD y está compuesta por un identificador para la relación, el identificador del titular de la relación, el identificador del elemento relacionado con el titular y el tipo de relación que estos poseen.

Este modelo es una variante de un patrón de diseño conocido como Entidad-Atributo-Valor (*Entity Value Attribute*, EVA). El cual permite tener un dominio detallado sobre todos los atributos que se le asignan a cualquier elemento que es almacenado. Las principales características de EVA, que a su vez se

[12] XML Schema Definition, XSD: es un formato para definir la correcta estructura de los elementos del documento XML.

convierten en sus principales ventajas son la gran flexibilidad en el almacenamiento de datos y la posibilidad de ampliar el conjunto de atributos sin cambiar la estructura de la tabla [50].

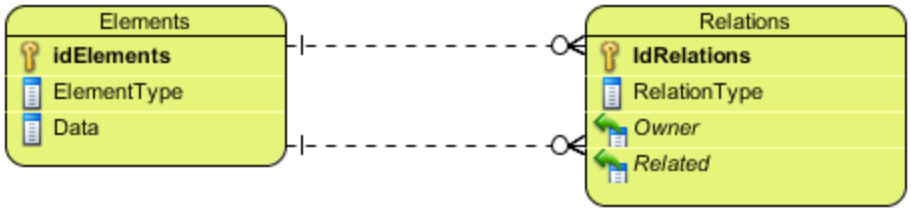


Ilustración 15: Modelo de Datos

DIAGRAMA DE DESPLIEGUE

Es un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Gráficamente, un diagrama de despliegue es una colección de nodos y arcos. La relación ente un nodo y el componente de despliegue puede mostrarse con una relación de dependencia, generalización o asociación de nodos desplegados en un compartimiento adicional dentro del nodo [51]. El diagrama mostrado en la **Ilustración 16** representa el despliegue del sistema en la Universidad, es decir la distribución de los nodos físicos a utilizar en el sistema de identificación y control de acceso de la UCI y la comunicación entre los mismos.

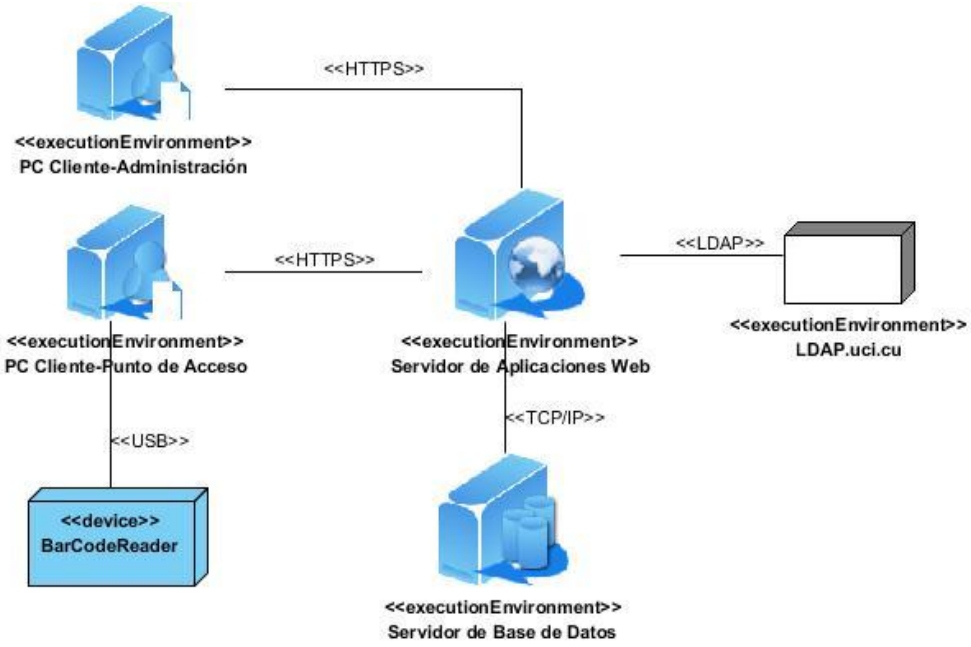


Ilustración 16: Diagrama de Despliegue

Para el **despliegue** del sistema se prevé contar con dos tipos de PC Cliente, una destinada para llevar a cabo las tareas de administración y otra ubicada en cada uno de los puntos de acceso, la cual tendrá

asociado un lector de código de barras por medio de una conexión USB para la lectura de un número de identificación único de cada usuario que desee acceder a la institución. Dichos clientes tienen acceso al servidor de aplicaciones que funciona sobre *Internet Information Server* al cual se le harán peticiones mediante el protocolo de comunicación HTTPS, y este a su vez mediante el protocolo de comunicación TCP/IP establece el vínculo con el servidor de base de datos con PostgreSQL 9.1 para realizar las consultas pertinentes a la identificación, autenticándose con ayuda del servidor LDAP.uci.cu, estableciéndose una conexión con el mismo de tipo LDAP.

DIAGRAMA DE COMPONENTE

Un diagrama de componentes se representa como un grafo de componentes unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. Cada componente representa una parte modular del sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos [52].

Estos diagramas son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos. En la **Ilustración 17** se muestra el diagrama de componentes del sistema.

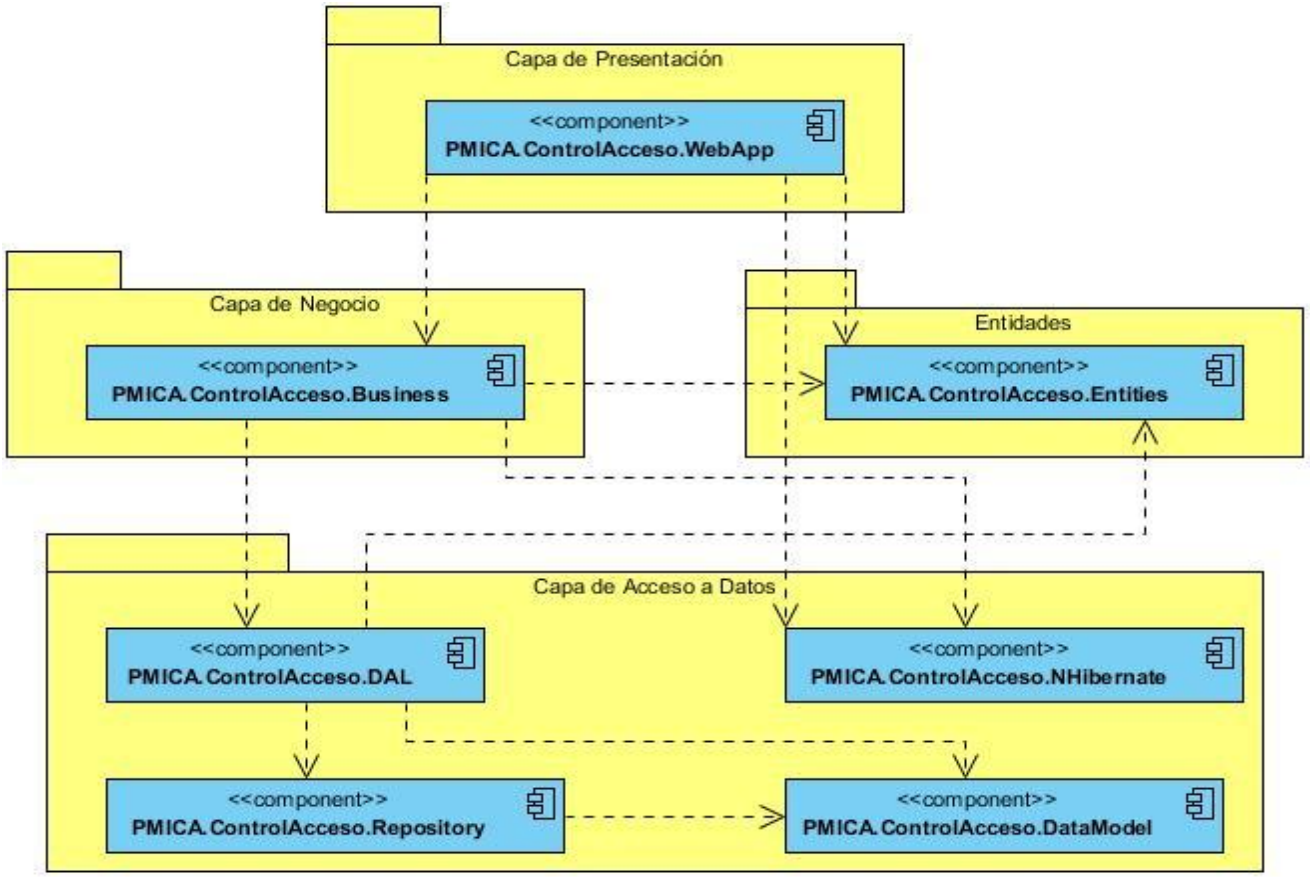


Ilustración 17: Diagrama de Componentes

TRATAMIENTO DE ERRORES

Para el tratamiento de errores se validan los datos que son introducidos por los usuarios al sistema. Se valida que el formato de los datos sea el esperado y que no se omita información de importancia para el procesamiento de solicitudes. Para esto al usuario, sólo se le brindan las opciones mínimas necesarias, a la hora de efectuar cualquier operación, por ejemplo, se deshabilitan ciertos botones si el usuario a partir del rol que ocupa no tiene que utilizarlos en ese momento, si se escriben datos incorrectos se le muestra un mensaje de error y se elimina lo escrito, ver **Ilustración 18**.

Por otra parte, mediante una combinación de validación en el lado del cliente y en el lado del servidor, se garantiza que los datos suministrados por los usuarios, se almacenen íntegros y no existan inconsistencias. Se verifican los campos obligatorios, y se revisa el tipo de dato. Todos los mensajes de error se muestran en color rojo para que resalten.

La manipulación de las excepciones que podrían ocurrir en tiempo de corrida de la aplicación se realiza a través de la sintaxis try {….} catch (*Exception* ex) {…} de tal forma que el sistema almacena los errores producidos y muestra al usuario mensajes sencillos para su comprensión.

Ilustración 18: Ejemplo de Tratamiento de Errores

ESTÁNDARES DE CODIFICACIÓN

Un estándar de codificación es un conjunto de directrices, normas y reglamentos sobre la forma de escribir el código de un programa, cuyo objetivo fundamental es lograr un mayor entendimiento entre todas las personas que trabajan directamente sobre el código. La coherencia y uniformidad lograda a través del uso

de los estándares de codificación constituyen un factor clave para lograr el éxito en el mantenimiento de programas.

El lenguaje de programación seleccionado para el desarrollo del sistema fue C#, a continuación se detallan algunos estándares establecidos a la hora de escribir el código en dicho lenguaje. Para ello conoceremos los términos usados a lo largo de toda la implementación, los cuales se basan en varias convenciones y guías propuestas por Microsoft.

- Convención **Pascal**– el primer carácter de cada palabra es en mayúscula y el resto en minúscula.

Ejemplo: **BackColor**

- Convención **Camel–** el primer carácter de cada palabra es en mayúscula (excepto la primera palabra) y el resto en minúscula.

Ejemplo: **backColor**

Para una correcta comprensión de la implementación del componente se debe conocer las buenas prácticas de programación las cuales son:

- Evitar escribir métodos de más de 25 líneas.
- El nombre del método debe decir que hace. No se deben usar abreviaturas.
- Cada método debe cumplir solamente una función. No se deben combinar más de una función por método tratar además que las funcionalidades sean los más atómicas posibles.
- Siempre vigilar parámetros no esperados. Por ejemplo, si se utiliza un parámetro con dos posibles valores, nunca se debe asumir que si no es el primero sea el segundo, por lo que se debe comprobar efectivamente cada caso.
- No usar directamente números o cadenas como constantes en el código. Declarar las constantes en la parte superior de los ficheros.
- Se debe evitar declarar constantes en todos los ficheros pues es recomendable que exista un fichero de configuración o una base de datos para las mismas.
- Convertir las cadenas a mayúsculas o a minúsculas antes de compararlas.
- Usar ***String.Empty*** en lugar de “”.
- Usar ***enum*** siempre que sea requerido. No usar números o cadenas para indicar valores discretos.
- Nunca asumir que el programa siempre corre desde una torre específica.
- No programar más de una clase por cada fichero.
- Evitar tener ficheros muy largos. Se deben dividir en dos o más clases.
- Evitar el paso de demasiados parámetros a los métodos, tratar de que no haya métodos de más de 3 parámetros.
- Especificar en el archivo ***AssemblyInfo*** la información del número de versión, descripción y desarrollador.

- Organizar los ficheros en carpetas apropiadas. Usar dos niveles de profundidad y no más de 10 carpetas en la raíz, así como no más de 5 subcarpetas en cada carpeta hija.
- Todas las conexiones a bases de datos, sockets y ficheros siempre son cerradas en el bloque ***finally***.

A continuación se muestra una tabla con los estándares de codificación utilizados para la implementación del sistema.

Tipos de Identificadores	Regla y Tipo de Convención	Ejemplo
Clases	Pascal	public class **HelloWorld**
Interfaces	**Pascal**→ Prefijo "I"	IModificar
Métodos	**Pascal**→ El nombre del método debe decir que hace. No se deben usar abreviaturas. Cada método debe cumplir solamente una función. No se deben combinar más de una función por método, tratar además que las funcionalidades sean los más atómicas posibles.	public void **VoyHacerEsto**

Variables y parámetros	**Camel**→ Usar el significado de las palabras para el nombre de las variables, no usar abreviaturas. No usar caracteres simples para el nombre de las variables excepto en los ciclos. No usar underscores (_) para los nombres de las variables locales. Todas las variables miembros de las clases deben ser prefijadas con underscore (_) para ser diferenciadas de otras variables. No usar nombres de variables que coincidan con palabras reservadas. Las variables booleanas y las propiedades se les pueden poner como prefijos "is". Siempre vigilar parámetros no esperados. Por ejemplo, si se usa un parámetro con dos posibles valores, no asumir nunca que si no es el primero sea el segundo, comprobar efectivamente cada caso.	private bool **_isFinalizado**;
		
		void DiceMundo (string name)
		{
		int contadorTotal=0;
		for(int i=0;i< count; i++)
		}

Namespace	Deben seguir el siguiente patrón estándar < product name >.<top level module>.<bottom level module>	PMICA.Workflow.Runtime
Comentarios	Comentarios deben estar en el mismo nivel del código. Usar el mismo nivel de identación.	// Format a message string message = "Hello" ;

Llaves	Las llaves se deben poner al mismo nivel del código que las contiene.	void Method (string name)
		{
		Method…
		}

#region	Se usa para agrupar código.	**#region** Métodos %Métodos% #endregion

Tabla 5: Estándares de Codificación.

PRUEBAS DE CAJA BLANCA

Se basa fundamentalmente en desarrollar pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se hayan probado de forma adecuada. Las pruebas de caja blanca conocidas también como pruebas de caja de cristal o pruebas

estructurales se centran en los detalles procedimentales del software, por lo que su diseño está de gran manera ligado al código fuente. Para la realización de este tipo de pruebas, el probador escoge diferentes valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y cerciorarse de que se devuelven los valores de salida adecuados. Aunque las pruebas de caja blanca son aplicables a varios niveles como son unidad, integración y sistema, habitualmente se aplican a las unidades de software [53].

PRUEBAS UNITARIAS

Luego de especificar el concepto de pruebas de caja blanca, es importante abordar acerca de las pruebas de unidad. En la programación, una prueba unitaria o de unidad es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada una de las partes que integran la aplicación funcione correctamente por separado. Las pruebas unitarias se realizan para controlar el funcionamiento de pequeñas porciones de código como son subprogramas (en la programación estructurada) o métodos (en la programación orientada a objeto, POO). Generalmente son realizadas por los mismos programadores puesto que al conocer con mayor detalle el código, se les simplifica la tarea de elaborar conjuntos de datos de prueba para probarlo [54].

El IDE Visual Studio 2010 permite la realización de pruebas de unidad a los métodos y a continuación se muestran las pruebas unitarias realizadas a algunas de las funcionalidades más importantes dentro de la Gestión de Recurso. También se pueden encontrar pruebas unitarias realizadas a otras funcionalidades en el **Anexo 4**: Pruebas Unitarias**.**.

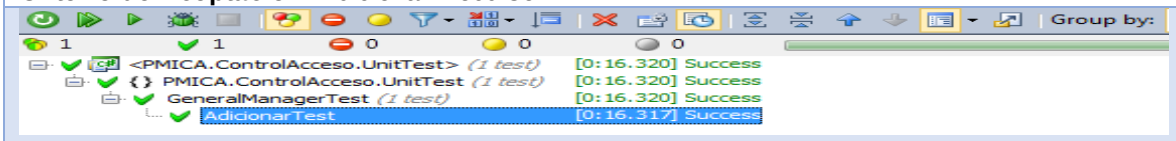
Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza.	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar de la interfaz principal del requisito gestionar recurso la opción "Adicionar Recurso", luego se llenan todos los datos que describen dicho recurso quedando registrado en el sistema.		
Entrada: Recurso recurso		
Criterio de Aceptación: Adicionar Recurso		
		

Tabla 6: Descripción de la prueba unitaria AdicionarTest

Prueba de Unidad		
Nombre Prueba: ObtenerRecursoByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza.	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el recurso de la lista de recursos y acceder a la opción "Mostrar Recurso", luego se le muestra todos los detalles del		

recurso así como la opción para activar o desactivar el mismo.
Entrada: Guid idRecurso
Criterio de Aceptación: Mostrar Recurso

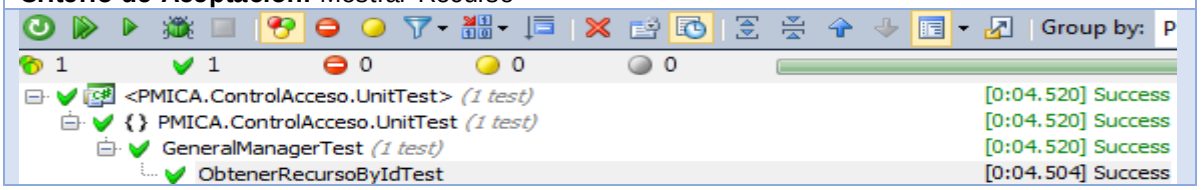


Tabla 7: Descripción de la prueba unitaria ObtenerRecursoByIdTest

Resultado de las pruebas unitarias

Durante la realización de las pruebas unitarias a cada una de las funcionalidades implementadas en el sistema, se pudo apreciar como al inicio de cada etapa los resultados obtenidos eran poco favorables y en la medida que avanzaban las iteraciones se les daba solución a los errores encontrados.

Las pruebas unitarias estuvieron divididas en tres etapas, una para cada módulo (administración y configuración), además de los requerimientos generales que el sistema debe cumplir (filtrar, ordenar y buscar). A continuación se muestra una tabla con los resultados obtenidos en dichas pruebas.

Módulo	Iteración	Funcionalidades Con errores	Funcionalidades Correctas	Funcionalidades No implementadas
ADMINISTRACIÓN	1	12	10	23
	2	15	25	5
	3	0	45	0
CONFIGURACIÓN	1	7	8	11
	2	6	15	5
	3	0	26	0
GENERALES	1	6	6	18
	2	10	15	5
	3	0	30	0

Tabla 8: Resultados de pruebas unitarias por etapas.

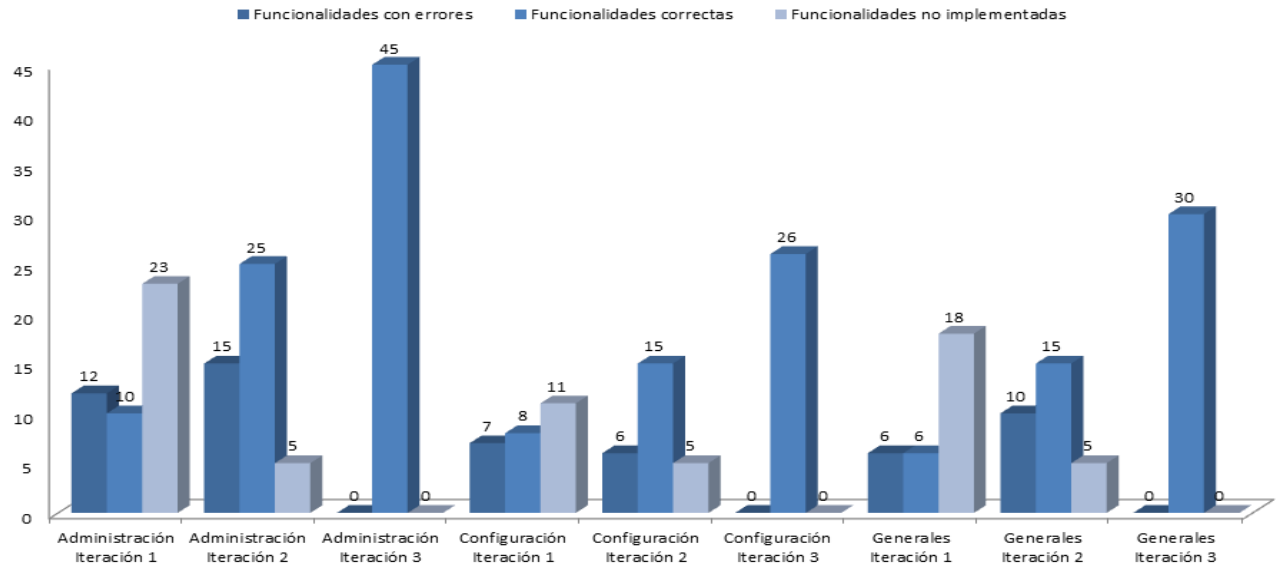


Ilustración 19: Gráfico con resultados de las pruebas unitarias.

PRUEBAS DE CAJA NEGRA

Esta técnica consiste básicamente en realizar pruebas de forma tal que se compruebe que cada función es operativa. Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Los casos de prueba de caja negra pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma correcta, que se produce una salida correcta y que la integridad de la información externa se mantiene [50]. Estos están constituidos por un conjunto de variables a través de las cuales se determina si el requisito de una aplicación es parcial o completamente satisfactorio. Los casos de prueba, fueron realizados y se encuentran registrados en la documentación oficial del proyecto PMICA.

A continuación se muestran algunas tablas con las variables y el caso de prueba perteneciente a la funcionalidad gestionar recurso, particularmente el escenario adicionar recurso.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Representa el nombre que identifica al recurso.
2	Tipo de Recurso	Lista Desplegable	No	Representa la categoría a la que pertenece el recurso.
3	Recurso Padre	Lista Desplegable	Si	Representa el recurso al que está suscrito.
4	Descripción	Área de texto	No	Detalles en lenguaje formal del recurso.

Tabla 9: Descripción de Variables para el caso de prueba Gestionar Recurso

Escenario	Descripción	Nombre	Tipo de Recurso	Recurso Padre	Descripción	Respuesta del sistema	Flujo central
Adicionar Recurso	El usuario selecciona la opción adicionar recurso en la pantalla principal.	V	V	V	V	El recurso se ha adicionado satisfactoriamente.	El usuario selecciona en la pantalla principal la opción Adicionar Recurso, introduce los datos necesarios y el sistema valida que los mismos sean correctos, en caso de no existir errores los datos son registrados.
		I	V	V	V	Los datos son incorrectos.	Si los datos son incorrectos, el sistema muestra los datos erróneos señalados.
		V	I	V	V	Los datos son incorrectos.	Si los datos son incorrectos, el sistema muestra los datos erróneos señalados.
		V	V	I	V	Los datos son incorrectos.	Si los datos son incorrectos, el sistema muestra los datos erróneos señalados.
		I	I	I	I	Los datos son incorrectos.	Si los datos son incorrectos, el sistema muestra los datos erróneos señalados.

Tabla 10: caso de prueba Gestiona Recurso escenario Adicionar Recurso

Resultados de las pruebas de caja negra.

Durante la realización de las pruebas caja negra a cada funcionalidad se observaron algunas no conformidades las cuales fueron solucionadas en las etapas posteriores. A continuación se muestra una tabla con los resultados obtenidos al concluir las iteraciones.

Iteración	Módulos		
	Administración	Configuración	Generales
1	25	12	23
2	16	7	14
3	7	3	6
4	0	0	0

Tabla 11: Resultados de las pruebas de caja negra.

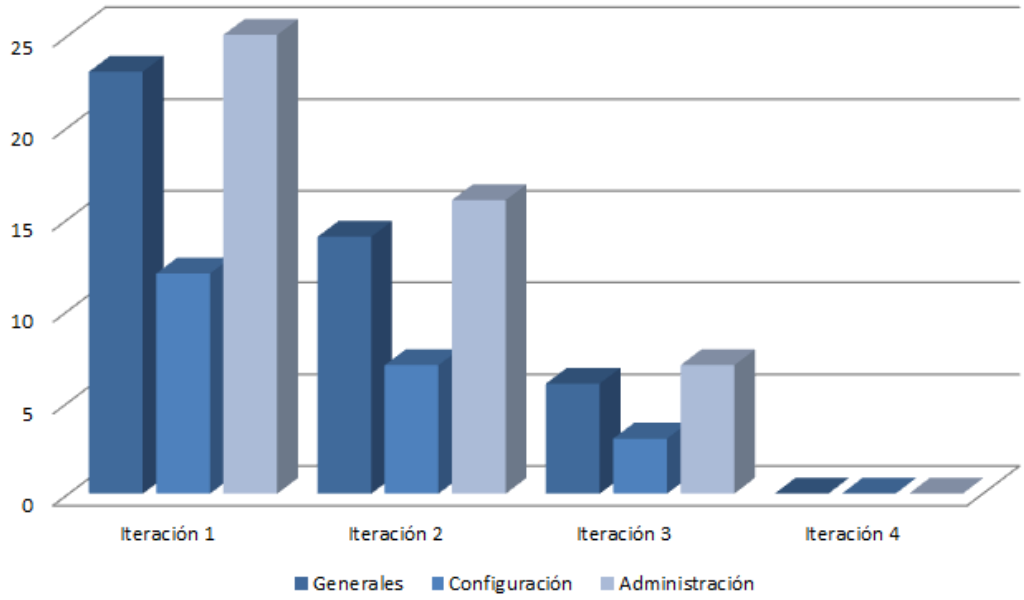


Ilustración 20: Grafico de no conformidades por etapas

CONCLUSIONES

En este capítulo fue presentado como está construido el sistema a partir del modelo de datos, el cual describe los datos y la relación que existen entre ellos. Además se construyó el diagrama de componentes, el que representa las dependencias entre los componentes software y de despliegue, para ilustrar los nodos que serán usados para la implantación de la aplicación y para cada uno de éstos el protocolo de comunicación. Se analizó como se tratan los errores de la aplicación para que brinde un mejor funcionamiento al usuario, así como los estándares de codificación para una correcta comprensión de la misma por los desarrolladores. Además se comprobó el correcto funcionamiento del sistema mediante las pruebas unitarias y de caja negra realizadas.

CONCLUSIONES GENERALES

En la presente investigación se arribaron a las siguientes conclusiones:

- El análisis referente a los modelos de control de acceso más utilizados a nivel mundial, arrojó que el Control de Acceso Basado en Roles es el más adecuado para ser empleado como base de la solución propuesta, a partir de la flexibilidad y neutralidad que posee.
- El estudio de los principales sistemas de control de acceso existentes en la actualidad determinaron que su implantación en la universidad resulta poco factible producto de que algunos están orientados a fines específicos y otros son de difícil adquisición, sin embargo sirvieron de ayuda para entender los principales procesos, módulos y funcionalidades que estos deben cumplir.
- La identificación de las herramientas, lenguajes y tecnologías posibilitó la creación de un marco de trabajo para llevar a cabo el proceso de desarrollo del software.
- La definición de FDD como metodología de desarrollo de software contribuyó a la realización de una correcta planificación, diseño y construcción de la solución propuesta.
- La descripción de los requisitos de software presentes en la fase de construcción de la lista de funcionalidades definida por FDD, permitió el conocimiento de cada una de las funcionalidades a implementar en el componente.
- El plan de iteraciones facilitó la organización del tiempo y el trabajo con el fin de culminar el desarrollo del componente en el período establecido.
- Las pruebas realizadas permitieron comprobar las funcionalidades del componente, arrojando resultados satisfactorios que validan la investigación.
- El componente desarrollado constituye una solución para la Universidad de las Ciencias Informáticas que elimina la adquisición de software costoso para controlar el acceso a sus instalaciones, además de posibilitar el aumento de nuevas funcionalidades que mejoren su desempeño.

RECOMENDACIONES

Los objetivos generales de este trabajo han sido logrados, pero a lo largo de su desarrollo, han surgido nuevas ideas que podrían implementarse en un futuro, de forma tal que se logre una la aplicación más completa y potente; para lo cual se recomienda:

- Implementar nuevas versiones de este componente que incluyan la incorporación de funcionalidades como la gestión de visitantes y la gestión de comensales, así como la adición de un módulo de control de presencia (gestión horaria), logrando que el sistema satisfaga cada una de las necesidades más comunes de las organizaciones.
- Integrar el componente con el subsistema de identificación de la Plataforma Modular de Identificación y Control de Acceso (PMICA).
- Elaborar un manual de usuario del componente con el objetivo de acelerar el aprendizaje del personal que lo utilizará.

REFERENCIAS BIBLIOGRÁFICAS

[1]. **Definicion.de.** [En línea] 14 de Febrero de 2012. El control de acceso. [Citado el: 10 de Diciembre de 2012.] http://www.umanick.com/index.php/soluciones/soluciones-por-aplicacion/control-de-acceso-logico

[2]. **Definicion.de.** [En línea] 10 de Noviembre de 2010. Definición de identificación. [Citado el: 22 de Diciembre de 2012.] http://definicion.de/identificacion/

[3]. **msdn.** [En línea] 18 de Mayo de 2010. Definición de autenticación. [Citado el: 15 de Febrero de 2013.] http://msdn.microsoft.com/es-es/library/syf5yeat.aspx

[4]. **msdn.** [En línea] 18 de Mayo de 2010. Definición de autorización[Citado el: 15 de Febrero de 2013.] http://msdn.microsoft.com/es-es/library/h2ds7dy5.aspx

[5]. **Definicion.de.** [En línea] 20 de Marzo de 2012. Definicion de Política. [Citado el: 10 de Diciembre de 2012.] http://definicion.de/politica/

[6]. **David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli**. *Role-Based Access Control.* USA : s.n., (2010).

[7]. **Jaehong Park, Ravi Sandhu**. The $UCON_{ABC}$ Usage Control Model: s.n., (2010).

[8]. **Brad J.Cox**, Ph.D. PBAC for diverse DoD Security Domains: s.n., (2011).

[9]. **Hyldeé M. Ibarra Naranjo, José A. Mañas Argemí.** RBAC: Alternativa actual para la realización de Control de Accesos a gran escala. España : s.n., (2012).

[10]. **Ytime:** [En línea] 2010. [Citado el: 15 de Enero de 2013.] http://www.by.com.es/ytime/

[11]. **Grupo SPEC**. [En línea] 2010. [Citado el: 16 de Diciembre de 2012.] http://www.grupospec.com/productos/singular-tech/cs-access-software-de-control-de-acceso-para-pymes

[12]. **Actum.** [En línea] 2013. [Citado el: 13 de Enero de 2013.] http://www.actum.es/soluciones-actum/control-de-acceso-biometrico/terminales-suprema/software-de-control-de-acceso-biostar

[13]. **Accesor.** [En línea] 2012. [Citado el: 11 de Enero de 2013.] http://www.accesor.com/esp/art2_query.php?fam=1&sfam=4

[14]. **Inditar. Tarjetas con soluciones.** [En línea] 2012. [Citado el: 16 de Enero de 2013.] http://www.inditar.com/control-accesos/software-para-edificios.php

[15]. **DATYS.** [En línea] 2011. [Citado el: 20 de Diciembre de 2012.] http://www.datys.cu/wpinfproducto.aspx?42

[16]. **XimaSafeAccess.** [En línea] 2011. [Citado el: 8 de Enero de 2013.] http://www.datys.cu/wpinfproducto.aspx?21

[17]. **DATYS.** [En línea] 2011. [Citado el: 20 de Enero de 2012.] http://www.datys.cu/WPInfNoticias.aspx?47,N

[18]. **DATYS.** [En línea] 2011. [Citado el: 20 de Enero de 2012.] http://www.datys.cu/wpinfproducto.aspx?50

[19]. **Ideatarjetas.** [En línea] 10 de Junio de 2009. Tarjetas de Banda Magnéticas [Citado el: 15 de Febrero de 2013.] http://www.ideatarjetas.es/como-codificar-tarjetas-de-banda-magnetica/

[20]. **quaronline.** [En línea] 10 de Junio de 2011. Tarjetas de Código de Barras [Citado el: 15 de Febrero de 2013.] http://www.quaronline.com/tarjetas-de-codigo-de-barras.php

[21]. **quaronline.** [En línea] 21 de Diciembre de 2010. Touch Memories [Citado el: 24 de Febrero de 2013.] http://www.wilcox.com.ar/productos.php?q=ctrlacceso

[22]. **Tarjetas y Credenciales plásticas de identificación.** [En línea] 21 de Septiembre de 2012. [Citado el: 13 de Diciembre de 2012.] http://www.tu-id.com/tarjetas-de-proximidad2.php

[23]. **Seguridad Informatica.** [En línea] 2 de Agosto de 2011. [Citado el: 18 de Diciembre de 2012.] http://seguridadinformatica-final.blogspot.com/2011/08/sistemas-biometricos.html

[24]. **ConbotasSucias.** [En línea] 27 de Noviembre de 2012. [Citado el: 14 de Enero de 2013.] http://conbotassucias.wordpress.com/2012/11/27/lector-de-codigos-de-barras/

[25]. **Aprende .NET con Juvega.** [En línea] 9 de Marzo de 2011. [Citado el: 2 de Febrero de 2013.] http://juvega.wordpress.com/2011/03/09/introduccin-a-net/

[26]. **msdn.** [En línea] 23 de Dicembre de 2012. [Citado el: 14 de Enero de 2013.] http://msdn.microsoft.com/en-us/library/zw4w595w.aspx

[27]. **msdn.** [En línea] 23 de Dicembre de 2012. [Citado el: 14 de Enero de 2013.] http://msdn.microsoft.com/es-es/library/4w3ex9c2.aspx

[28]. **msdn.** [En línea] 23 de Dicembre de 2012. [Citado el: 14 de Enero de 2013.] http://msdn.microsoft.com/es-es/library/fa1h9d0d%28v=VS.80%29.aspx

[29]. **Introduction to c#.** [En línea] Universidad de Alicante, 8 de Mayo de 2012. [Citado el: 20 de Enero de 2013.] http://si.ua.es/en/documentacion/csharp/documentos/masterpages/modulo1.pdf

[30]. **ecured.** [En línea] 23 de Dicembre de 2012. [Citado el: 14 de Enero de 2013.] http://www.ecured.cu/index.php/Servicios_Web

[31]. **Visual Studio.** [En línea] 25 de Mayo de 2010. [Citado el: 10 de Febrero de 2013.] http://msdn.microsoft.com/es es/library/vstudio/w0x726c2%28v=vs.100%29.aspx

[32]. **Visual Studio.** [En línea] 2012. [Citado el: 10 de Febrero de 2013.] http://msdn.microsoft.com/en-us/vstudio/aa718325.aspx

[33]. **EcuRed.** [En línea] 18 de Noviembre de 2009. [Citado el: 13 de Diciembre de 2012.] http://www.ecured.cu/index.php/SQL

[34]. **Aula Clic.** [En línea] Febrero de 2010. [Citado el: 20 de Diciembre de 2012.] http://www.aulaclic.es/sqlserver/t_2_1.htm

[35]. **Scribd.** [En línea] 29 de Agosto de 2010. [Citado el: 19 de Diciembre de 2012.] http://es.scribd.com/doc/36570462/postgreSQL-investigacion

[36]. **Visual Paradigm.** [En línea] 21 de Enero de 2013. [Citado el: 7 de Febrero de 2013.] http://www.visual-paradigm.com/product/vpuml/

[37]. **UML.** [En línea] 1 de Abril de 2010. [Citado el: 15 de Enero de 2013.] http://www.uml.org/

[38]. **Palacios, Damian.** Motor de persistencia nhibernate. [En línea] [Citado el: 20 de noviembre de 2012.] http://www.slideshare.net/DamianPalacios/motor-de-persistencia-nhibernate

[39]. **Haileen Alicia R. S., Lissette V. García.** Metodologías de Desarrollo de Software. FDD: s.n., 2012.

[40]. **Articulo.org.** [En línea] 15 de Enero de 2013. [Citado el: 14 de Enero de 2013.] http://www.articulo.org/articulo/17916/diccionario_de_procesos.html

[41]. **Dailyn Sosa López.** Sistema para el control del uso de los software educativo. : s.n., 2012.

[42]. **IEEE.** [En línea] 2013. [Citado el: 17 de Febrero de 2013.] http://www.ieee.org/index.html

[43]. **Alonso. G., Casati. F., Kuno H., Machiraju, V**. Web Services Concepts, Architectures and Applications. s.n., 2010.

[44]. Hasheado. [En línea] 8 de Septiembre de 2010. [Citado el: 10 de Enero de 2013.] http://www.hasheado.com/mvc-patron-de-diseno.html

[45]. **SILVIA, Aramúndiz**. DIAGRAMA DE SECUENCIA, Comportamiento de los sistemas, 2012.

[46]. **Larman, Craig**. UML y Patrones: s.n., 2010.

[47]. **Patrones de Diseño.** [En línea] Mayo de 2009. [Citado el: 27 de Febrero de 2013.] http://patronesdediseno.blogspot.com/2009/05/patron-facade.html

[48]. **msdn, Patron Singleton.** [En línea] 2013. [Citado el: 1 de Marzo de 2013.] http://msdn.microsoft.com/es-es/library/bb972272.aspx

[49]. **Manuel.** Unit of Work y los ORMs. [En línea] 9 de julio de 2012. [Citado el: 30 de marzo de 2013.] http://www.nocompila.com/2012/07/unit-of-work-y-los-orms.html

[50]. **José C. Correa Bautista, Yendry Machado García.** Subsistema de Aprovisionamiento de Usuarios para el Sistema de Administración de Identidades, 2012.

[51]. **Rubiano, Martha.** UML Diagramas de despliegue. [En línea] 9 de octubre de 2009. [Citado el: 30 de marzo de 2013.] http://es.scribd.com/doc/19611312/diagramas-de-despliegue-2222

[52]. **Slideshare Present Yourself.** *DiagramasUML: Componentes y Despliegue .* [En línea] 26 de septiembre de 2010. [Citado el: 30 de marzo de 2013.] http://www.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue

[53]. **MANUEL, T. G**. Tecnicas de Prueba, 2012.

[54]. **CARLOS, B**. QUnit, testeando nuestras aplicaciones, 2011.

BIBLIOGRAFÍA CONSULTADA

[1]. **Fowler, Martin.** Patterns of Enterprise Application Architecture. s.l. : Addison-wesley.

[2]. **Elaboración, Fase de.** *Flujo de Trabajo de Análisis y Diseño.* Ciudad de la Habana : s.n., 2010

[3]. **Karen Scarfone, Vincent C. Hu.** Guidelines for Access Control System Evaluation Metrics. USA : s.n., 2012.

[4]. **Pierangela Samarati, Sabrina De Capitani di Vimercati.** *Access Control: Policies, Models, and Mechanism.* Italia : s.n., 2012.

[5]. **Jaehong Park, Ravi Sandhu.** Towards Usage Control Models: Beyond Traditional Access Control. USA : s.n., 2012.

[6]. **Ram Krishnan, Xin Jin**. A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC: s.n., (2010).

[7]. IBM. [En línea] 28 de Agosto de 2012. [Citado el: 9 de Enero de 2013.] http://www-01.ibm.com/software/rational/uml/

[8]. IBM developerWorks. [En línea] 5 de Mayo de 2010. [Citado el: 10 de Enero de 2013.] http://www.ibm.com/developerworks/

[9]. Ingeniería del Software. [En línea] 2013. [Citado el: 18 de Enero de 2013.] http://eva.uci.cu/

[10]. Materiales disponiblenes en el Entorno Virtual de Aprendizaje. [En línea] 17 de Mayo de 2011. [Citado el: 11 de febrero de 2013.] http://eva.uci.cu/mod/resource/view.php?id=9400&subdir=/UML_y_Patrones

[11]. msdn, Patrones y Antipatrones: una Introducción. [En línea] 2013. [Citado el: 1 de Marzo de 2013.] http://msdn.microsoft.com/es-es/library/bb972251.aspx

[12]. Security Artwor. [En línea] 12 de Marzo de 2010. [Citado el: 29 de Noviembre de 2012.] http://www.securityartwork.es/2010/03/12/sistemas-de-control-de-acceso-mac-y-dac/

[13]. subinet. [En línea] 14 de Septiembre de 2010. [Citado el: 17 de Diciembre de 2012.] http://www.subinet.es/guias-y-tips/guias-y-tips-seguridad/%C2%BFcuales-son-los-modelos-de-control-de-acceso/

[14]. The official Microsoft ASP.NET Site. [En línea] 2013. [Citado el: 4 de Febrero de 2013.] http://www.asp.net/

GLOSARIO DE TÉRMINOS

- ✓ **Aforo:** es el cálculo de una cantidad existente en un depósito.
- ✓ **ASP:** es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.
- ✓ **Base de Datos**: conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina, accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.
- ✓ **ByTech:** Es una empresa líder en el diseño, fabricación y distribución de dispositivos de Control de Accesos
- ✓ **Common Language Runtime (CLR):** entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. El CLR es el encargado de compilar una forma de código intermedio llamada *Common Intermediate Language*, al código de máquina nativo, mediante un compilador en tiempo de ejecución.
- ✓ **DATYS:** Empresa cubana que produce bienes y servicios informáticos, desarrollando el empleo integral de las tecnologías de la información, de las comunicaciones y de seguridad técnica, con alta calidad y eficiencia.
- ✓ **Framework:** estructura de soporte definida en 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.
- ✓ **IDE:** software compuesto por un conjunto de herramientas de programación. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).
- ✓ **LDAP:** 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.
- ✓ **Módulo:** componente auto controlado de un sistema que posee una interfaz bien definida hacia otros componentes.
- ✓ **PYMES:** Acrónimo de Pequeñas y medianas Empresas
- ✓ **Servidor:** un servidor es una computadora que maneja peticiones de datos, correo, servicios de redes y transferencia de archivos de otras computadoras (clientes). También puede referirse a un

software específico. Una computadora puede tener distintos software de servidor, proporcionando muchos servidores a clientes en la red.

- ✓ **SQL:** lenguaje declarativo de acceso a bases de datos (BD) relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar información de interés de una BD.
- ✓ **Termograma:** técnica que permite medir temperaturas a distancia con exactitud y sin necesidad de contacto físico con el objeto a estudiar. Mediante la captación de la radiación infrarroja del espectro electromagnético, utilizando cámaras termográficas o de termovisión, se puede convertir la energía radiada en información sobre temperatura.
- ✓ **Usuario:** persona que tiene una cuenta en una determinada computadora por medio de la cual puede acceder a los recursos y servicios que ofrece una red.
- ✓ **XML (Extensible Markup Language):** lenguaje de marcado extensible, es un metalenguaje extensible de etiquetas desarrollado por *el World Wide Web Consortium* (W3C).
- ✓ **XSD:** XML Schema es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML.

ANEXOS

ANEXO 1: MODELO DE PROCESOS DEL NEGOCIO

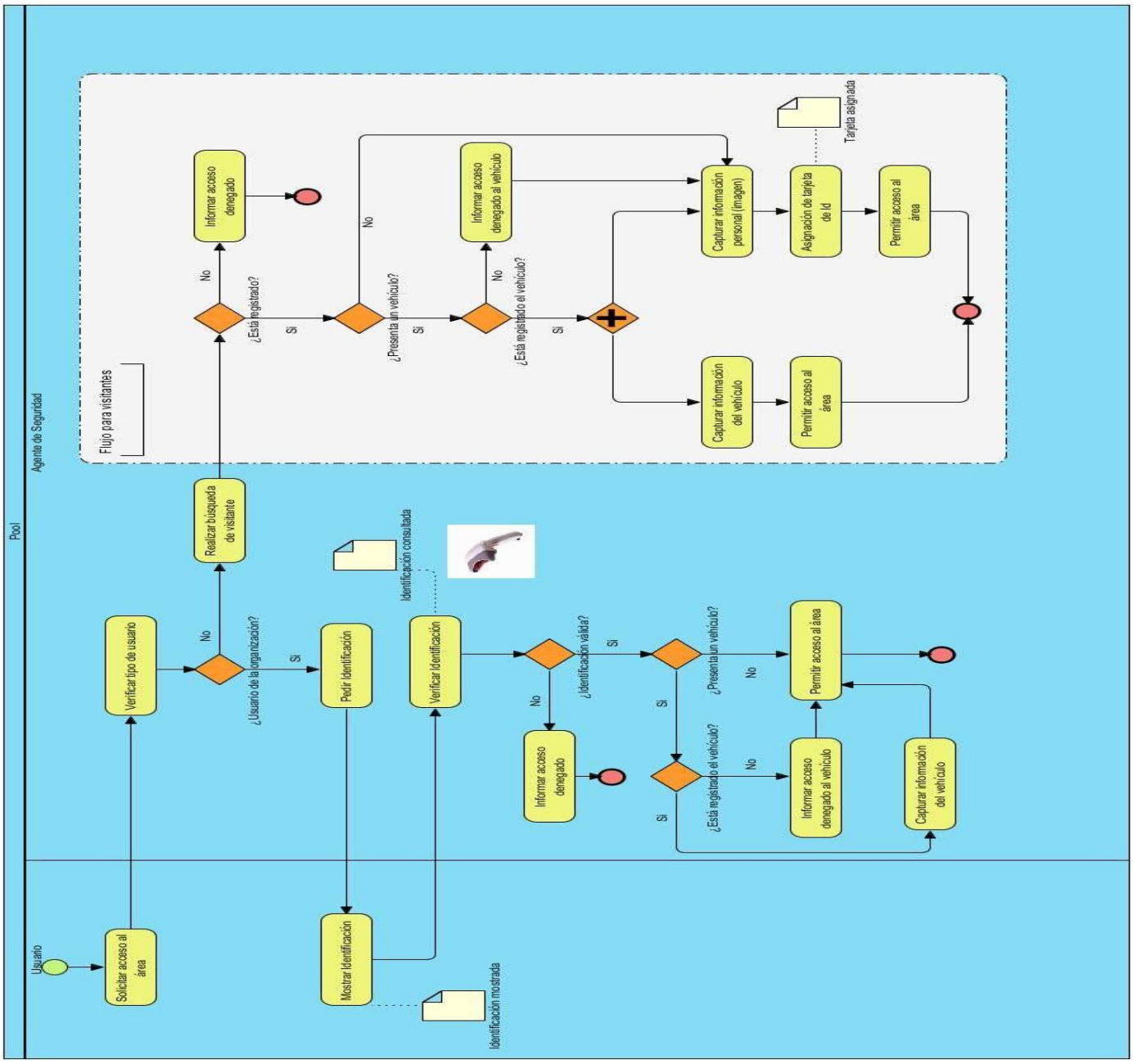


Ilustración 21: Modelo de Procesos del Negocio

ANEXO 2: DIAGRAMAS DE CLASES DEL DISEÑO

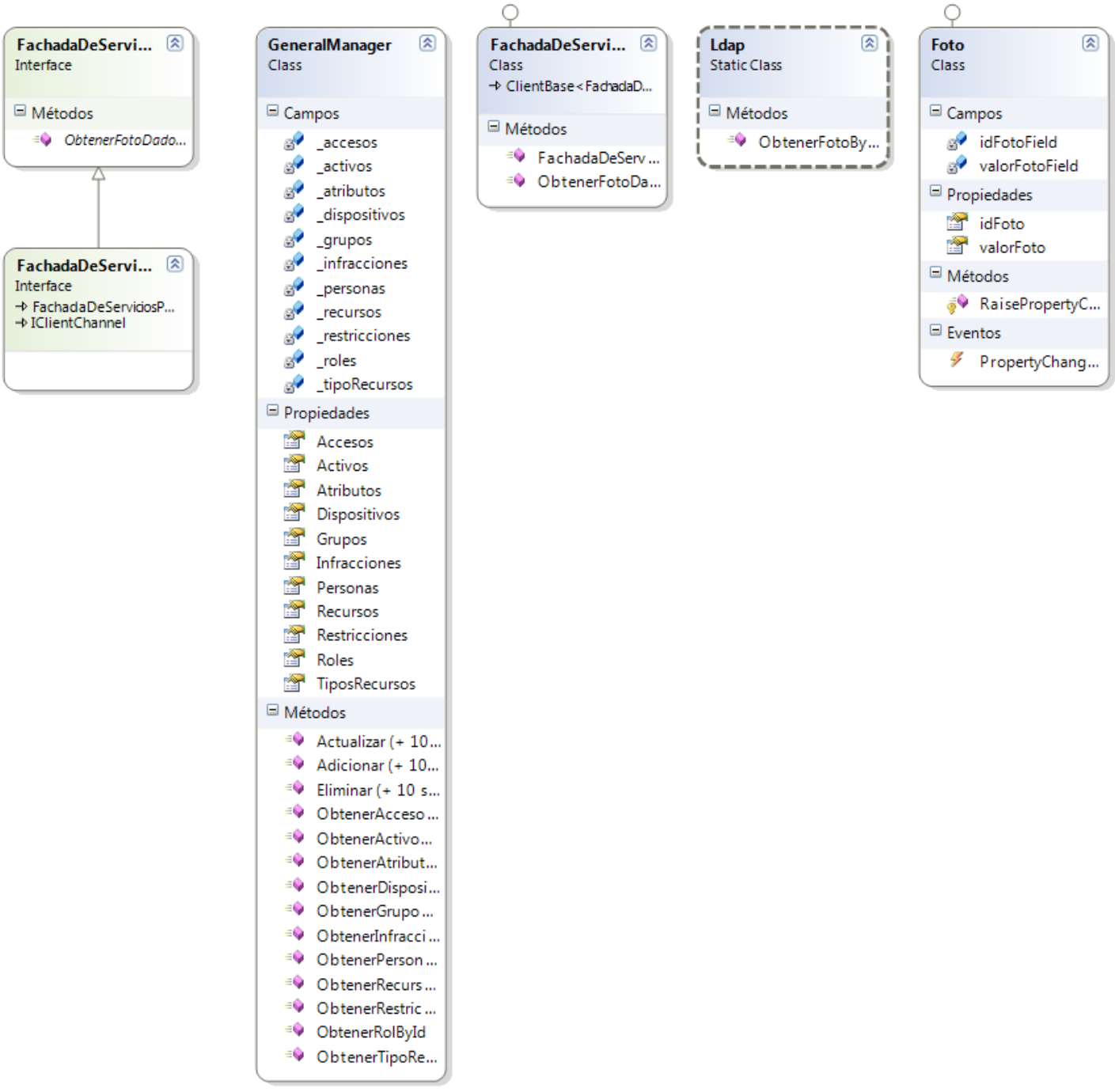


Ilustración 22: Capa Negocio- Lógica de Negocio

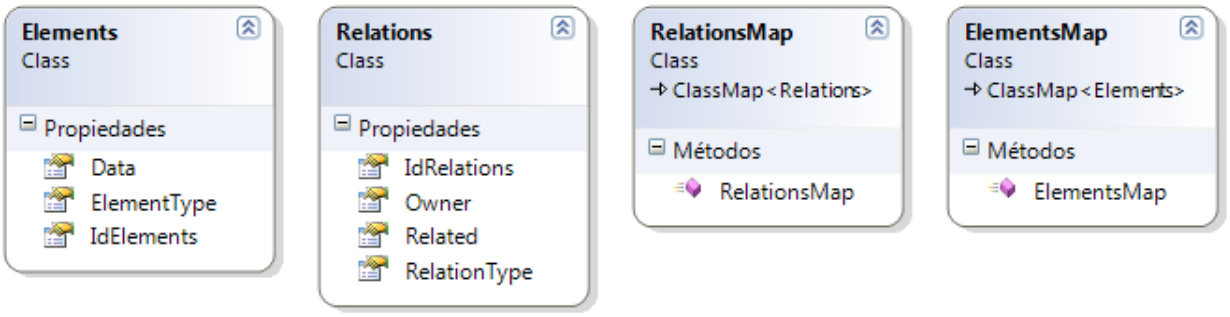


Ilustración 23: Capa Acceso a Datos-Modelo Datos

Ilustración 24: Capa Acceso a Datos-NHibernate

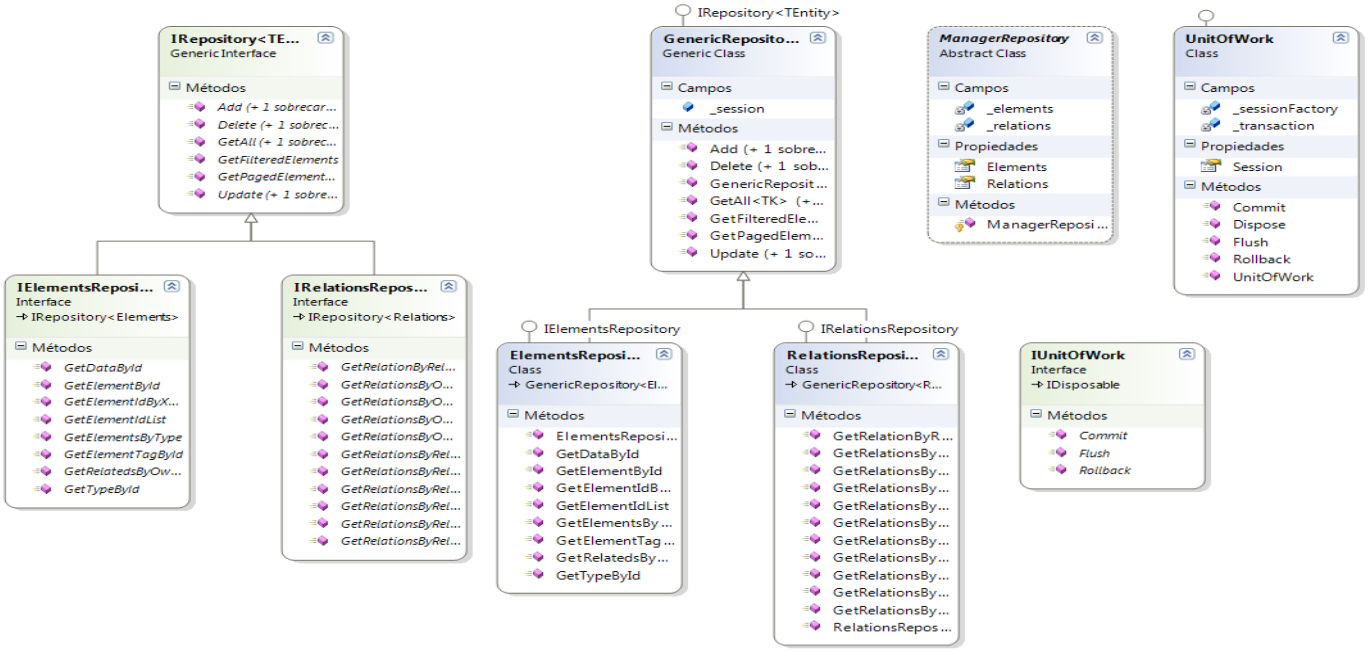
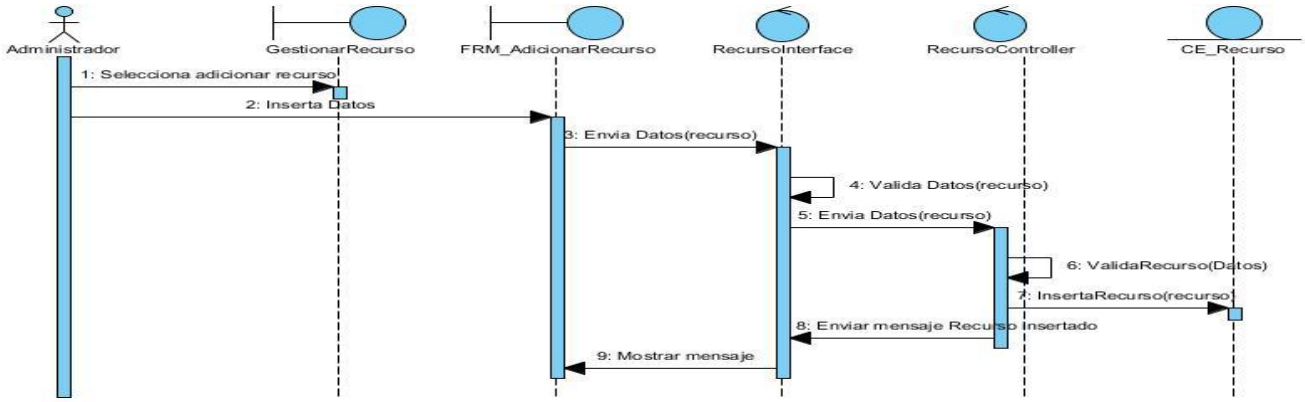


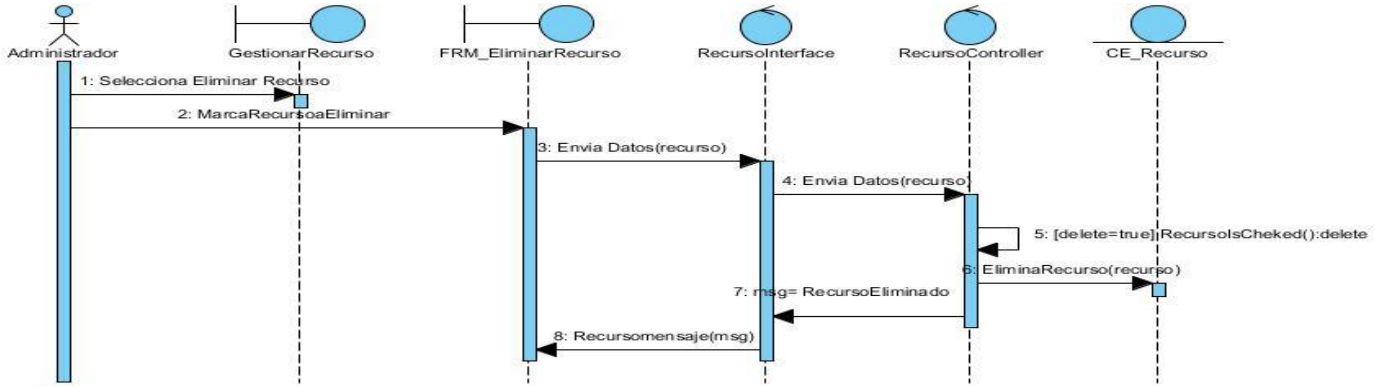
Ilustración 25: Capa Acceso a Datos-Repositorio

ANEXO 3: DIAGRAMAS DE SECUENCIA

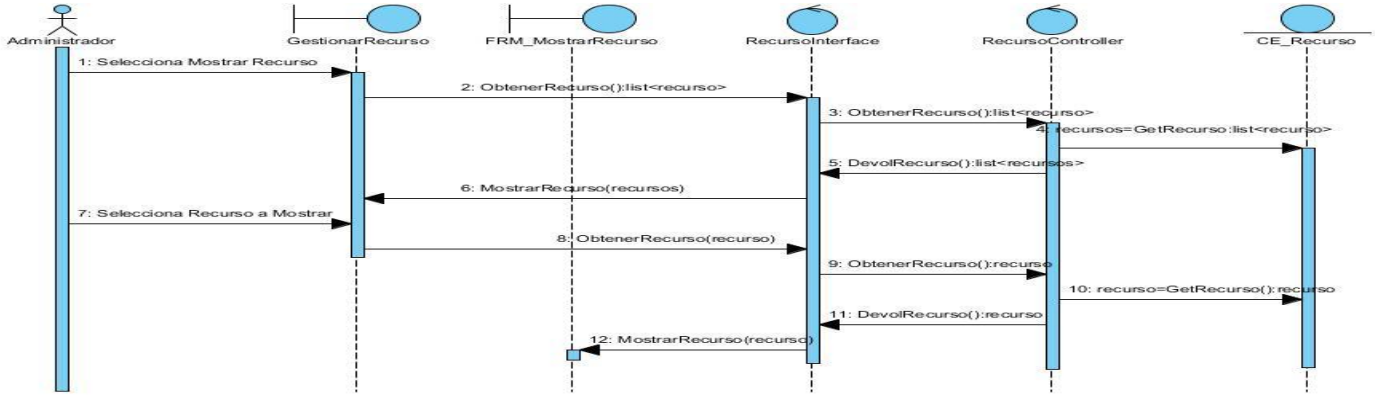
Módulo de Administración



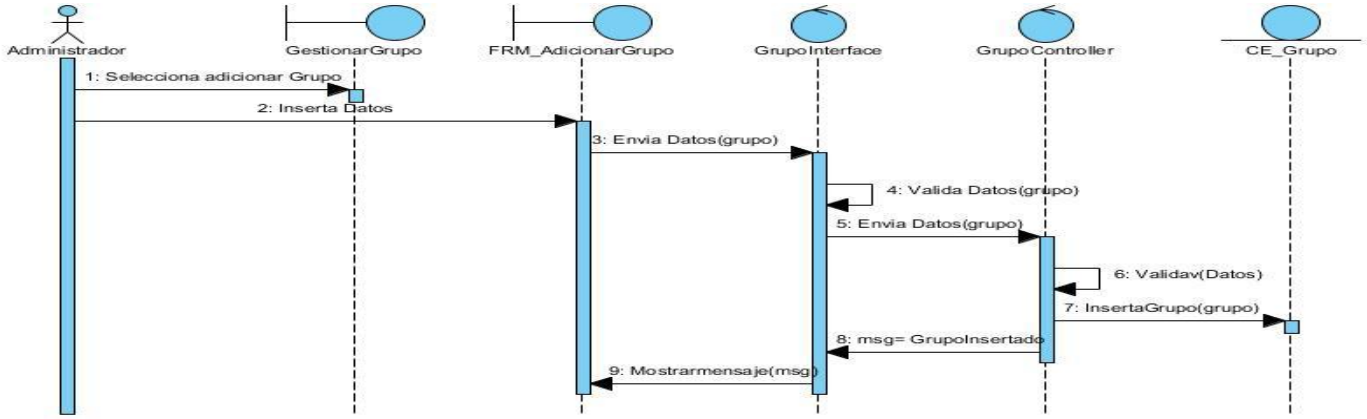
Adicionar Recurso



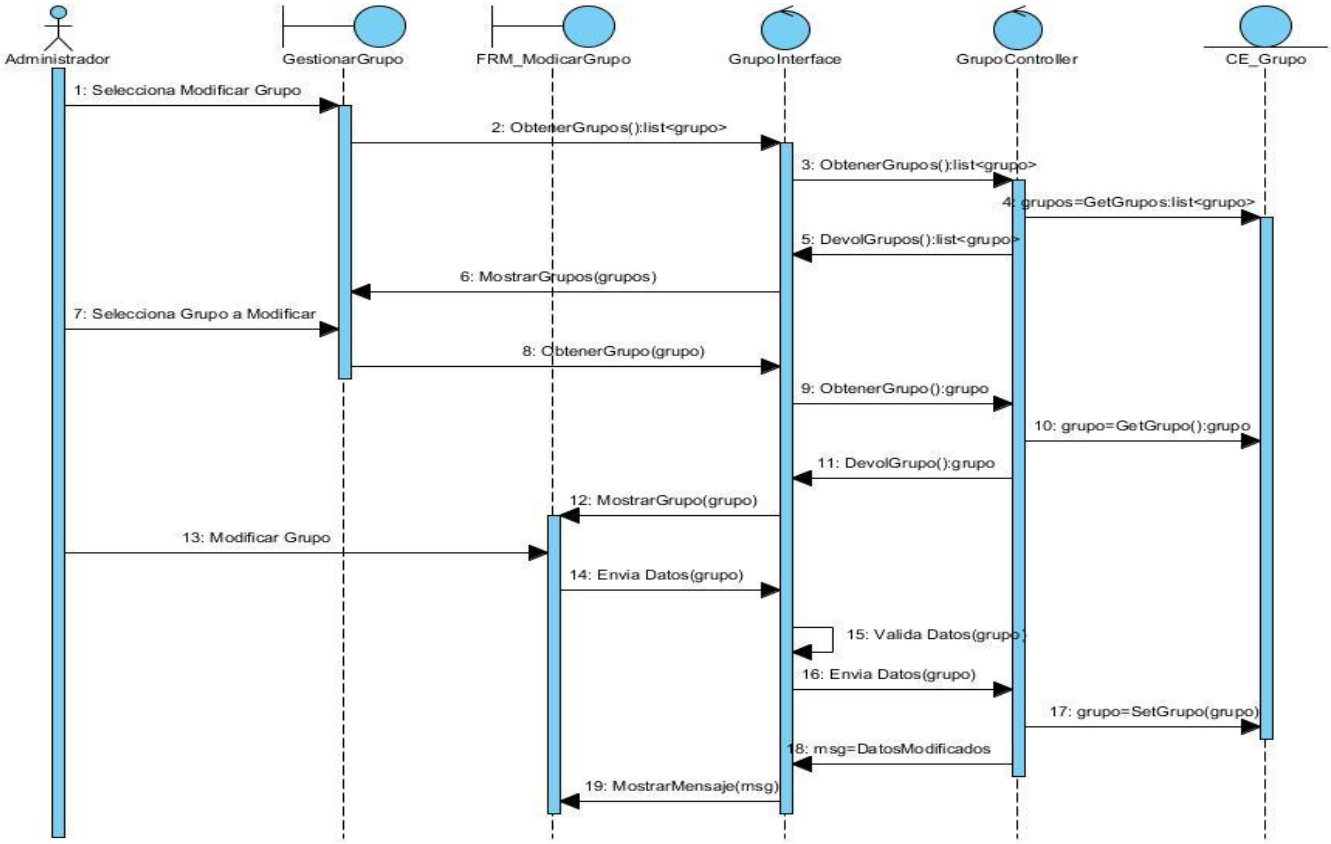
Eliminar Recurso



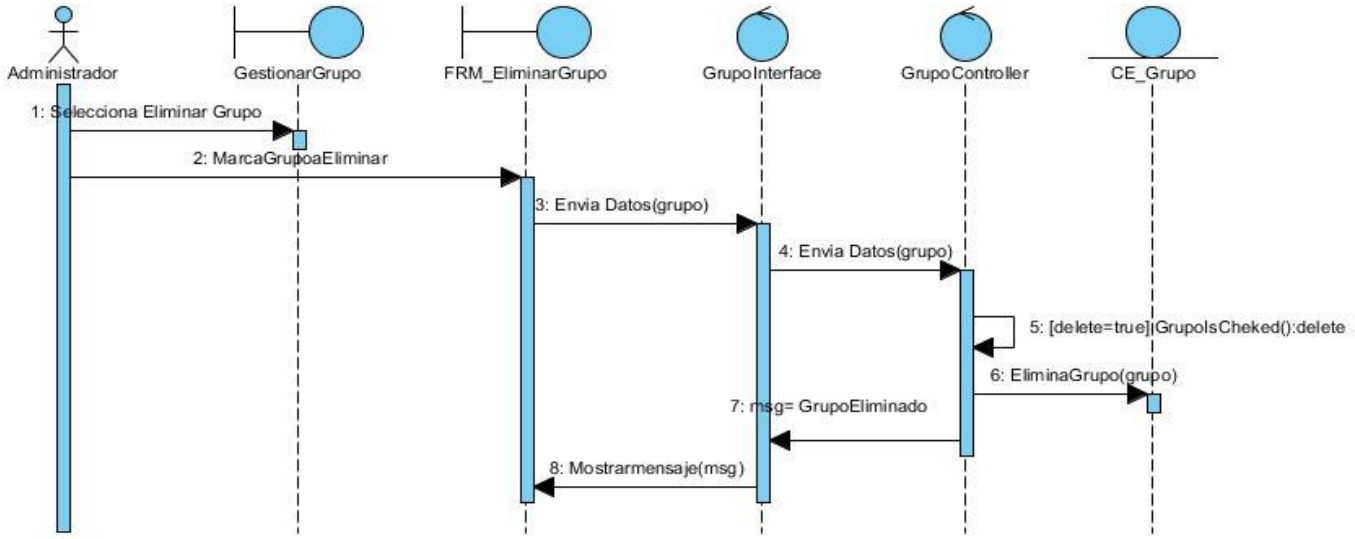
Mostrar Recurso



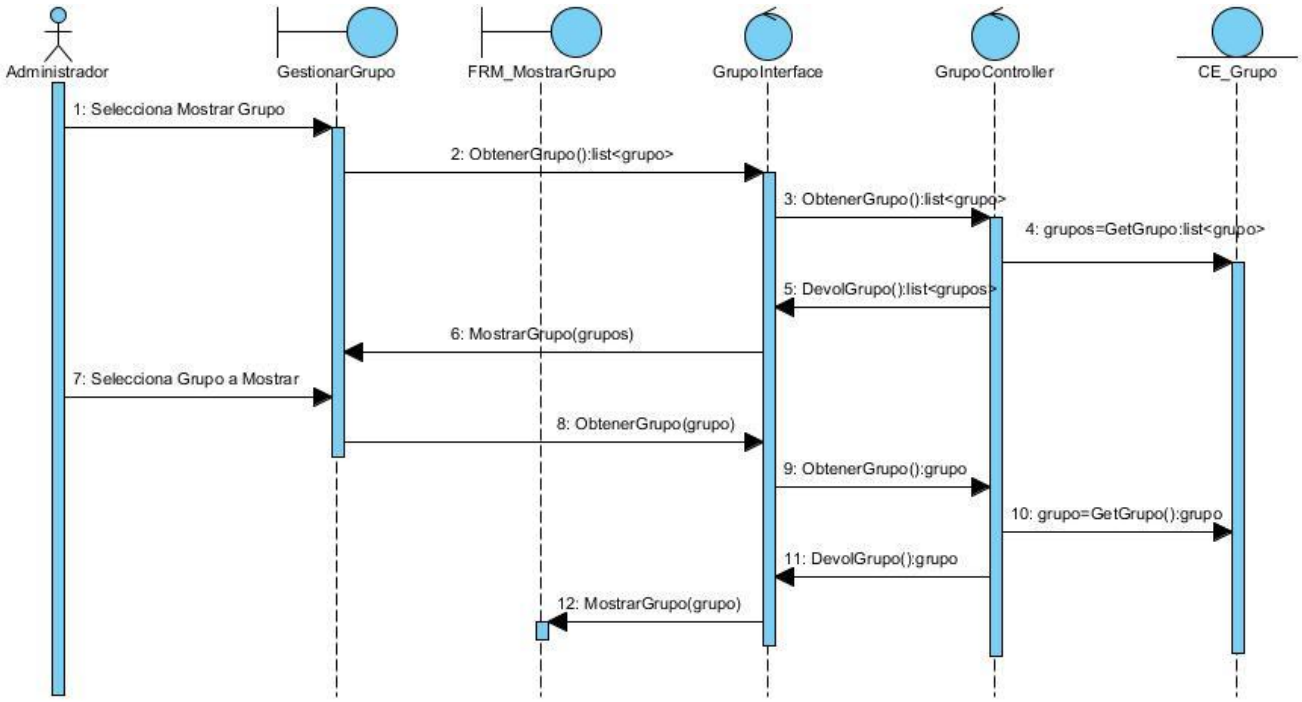
Adicionar Grupo



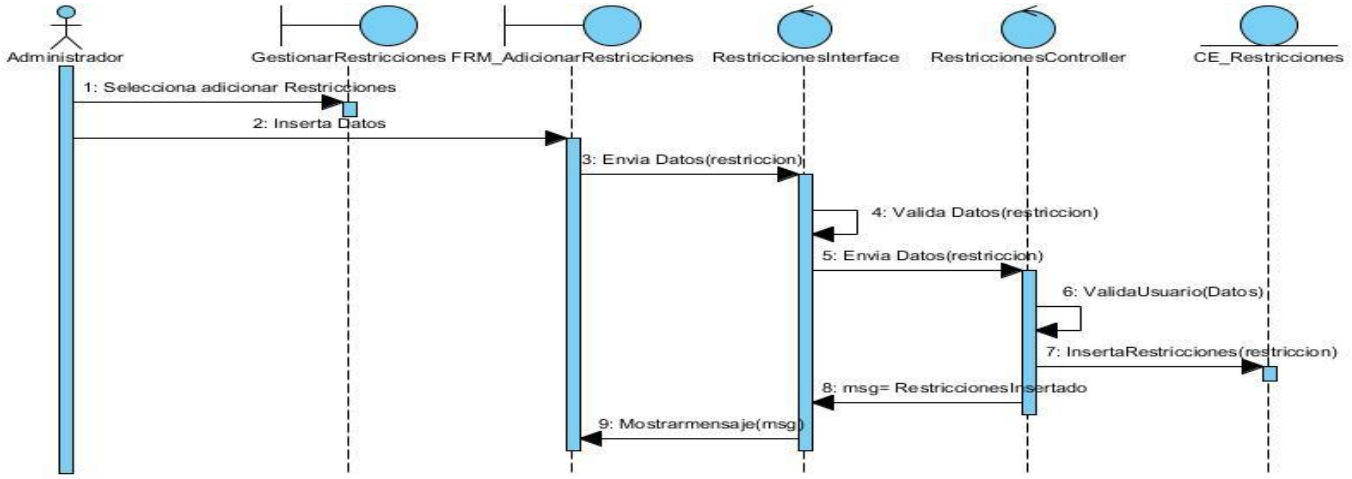
Modificar Grupo



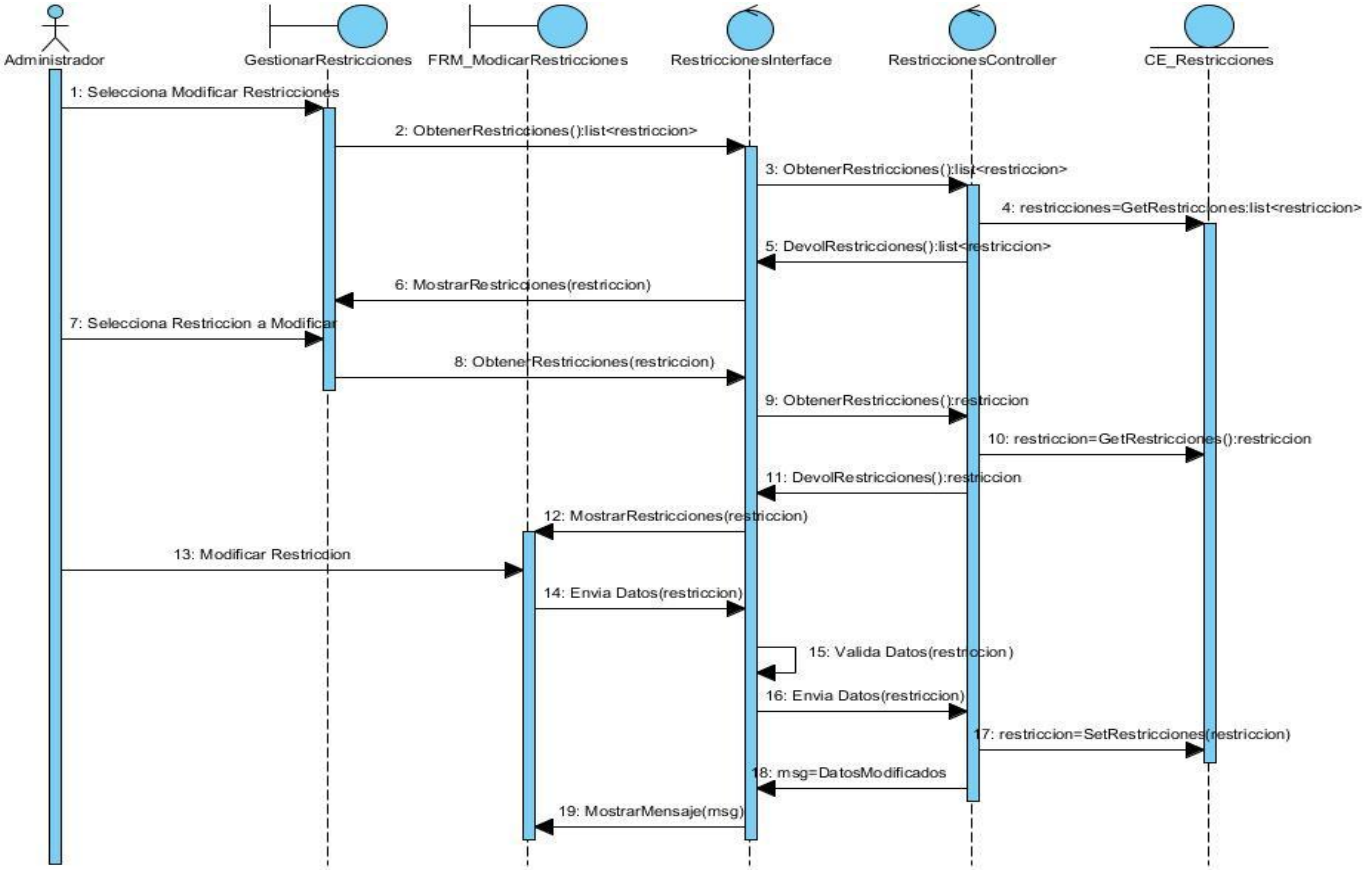
Eliminar Grupo



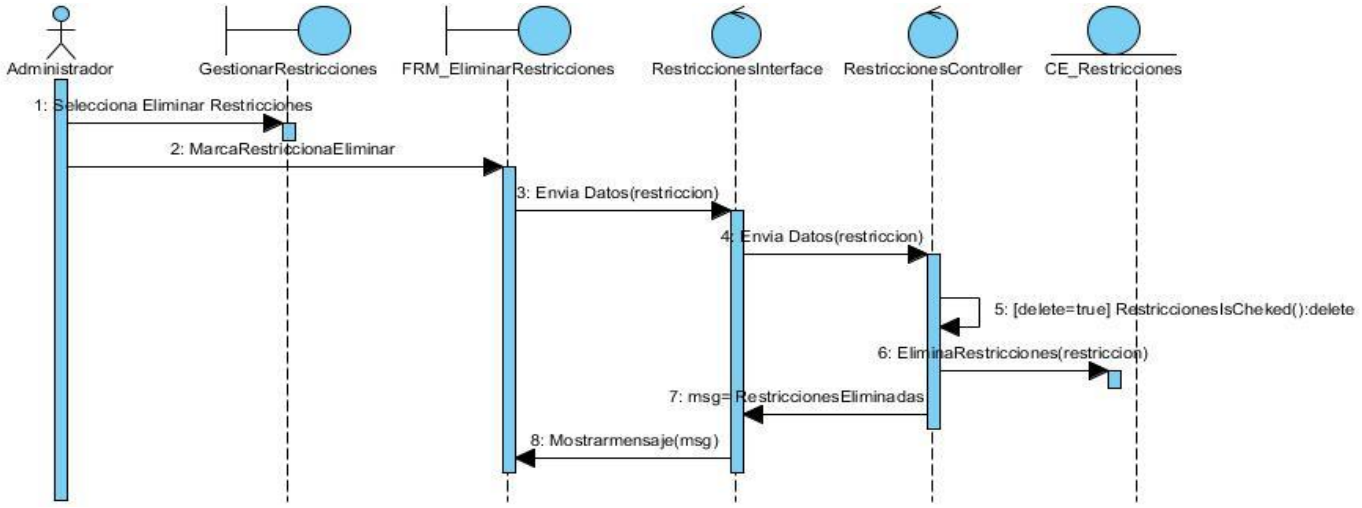
Mostrar Grupo



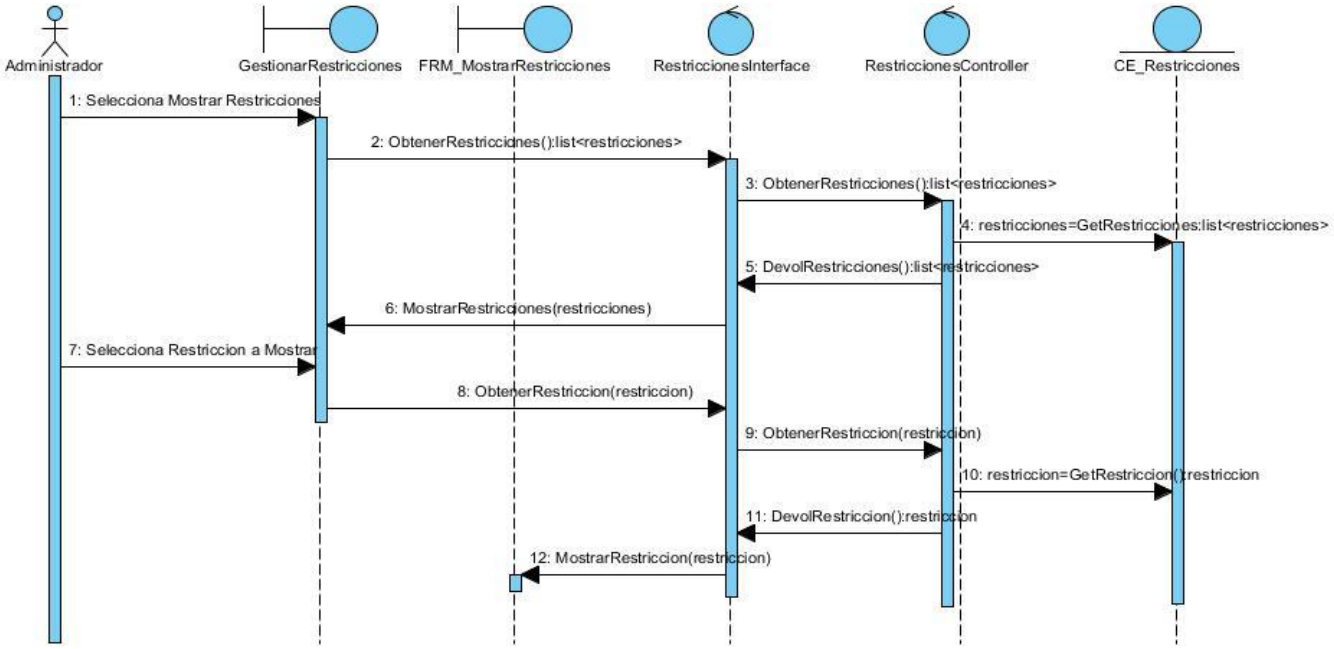
Adicionar Restricciones



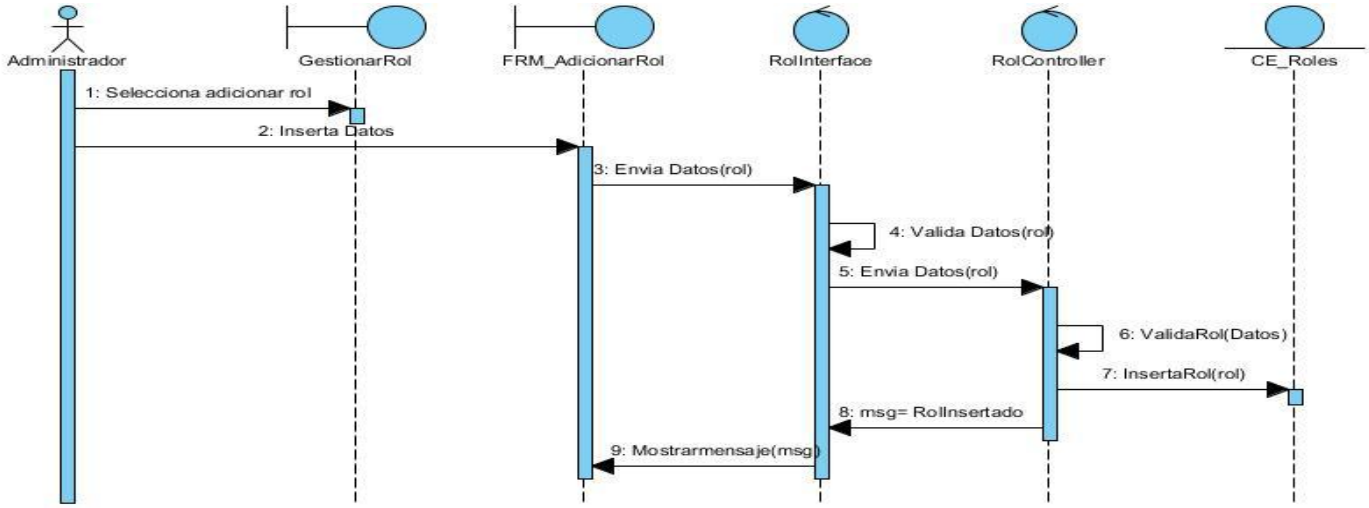
Modificar Restricciones



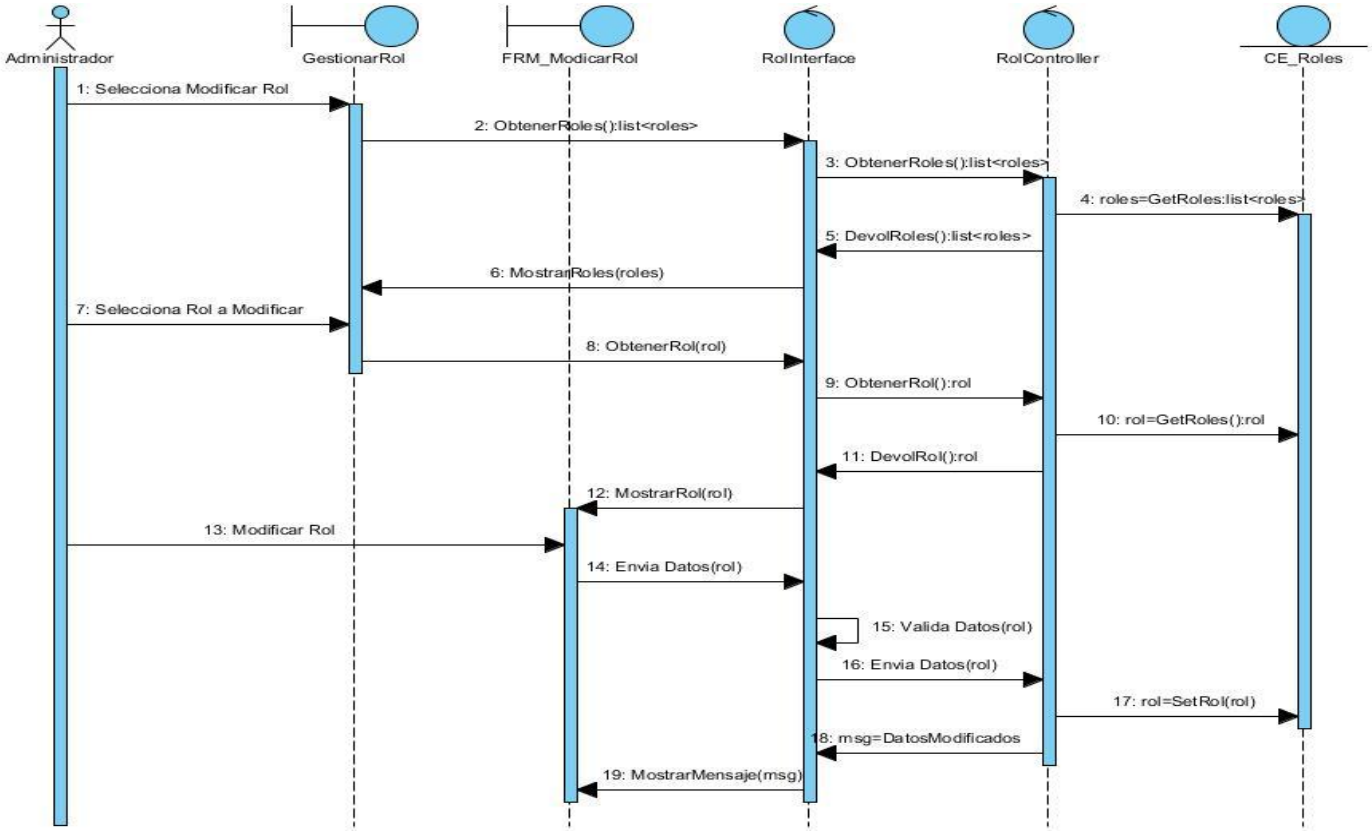
Eliminar Restricciones



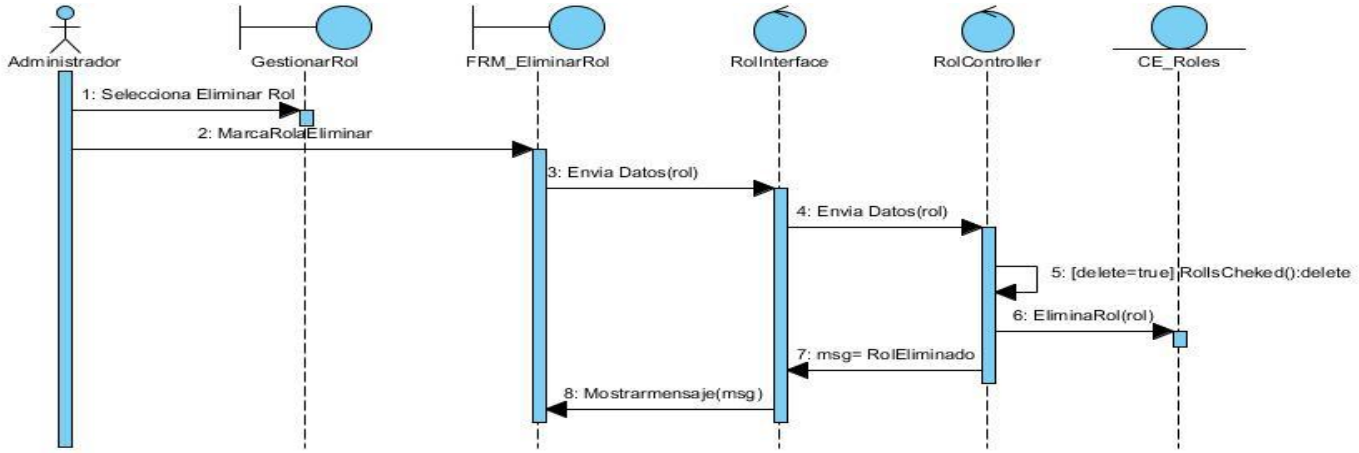
Mostrar Restricciones



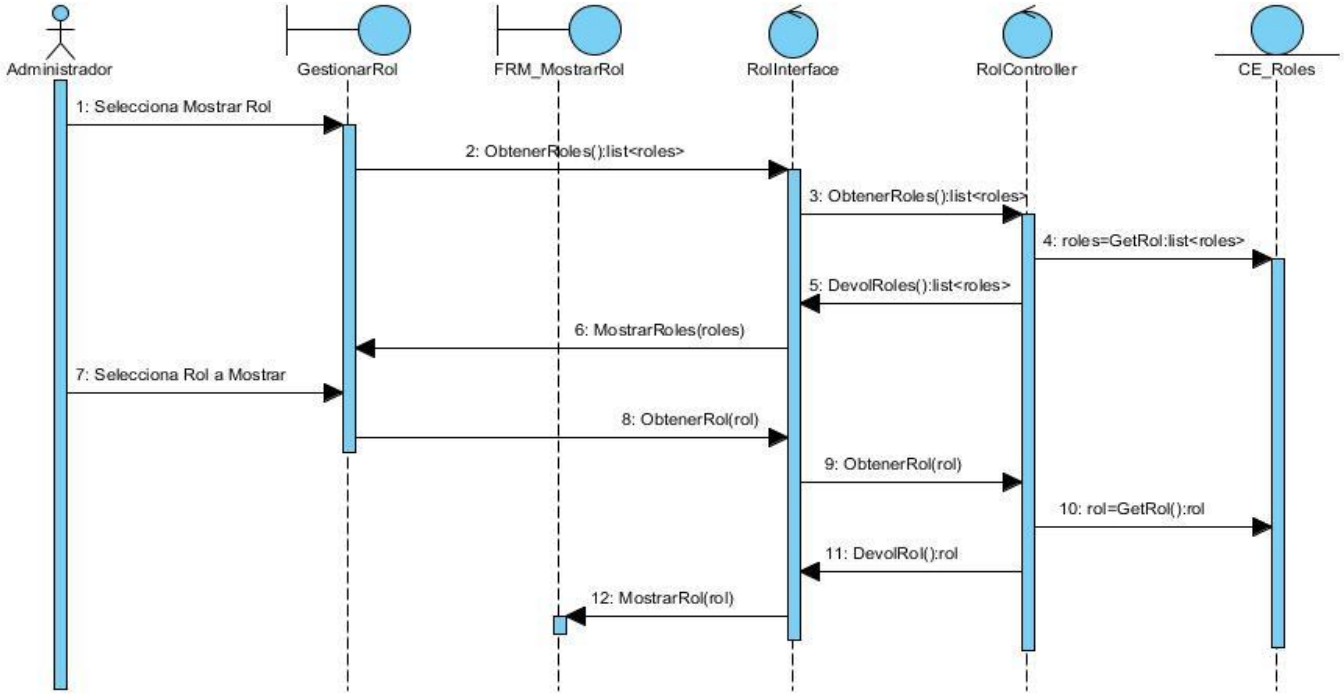
Adicionar Rol



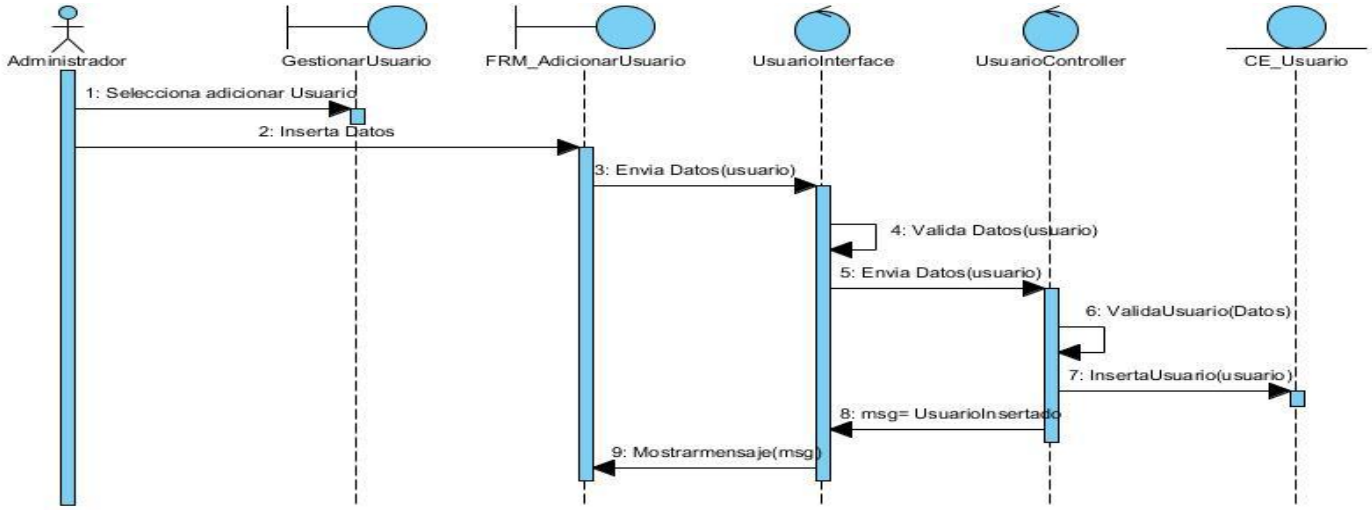
Modificar Rol



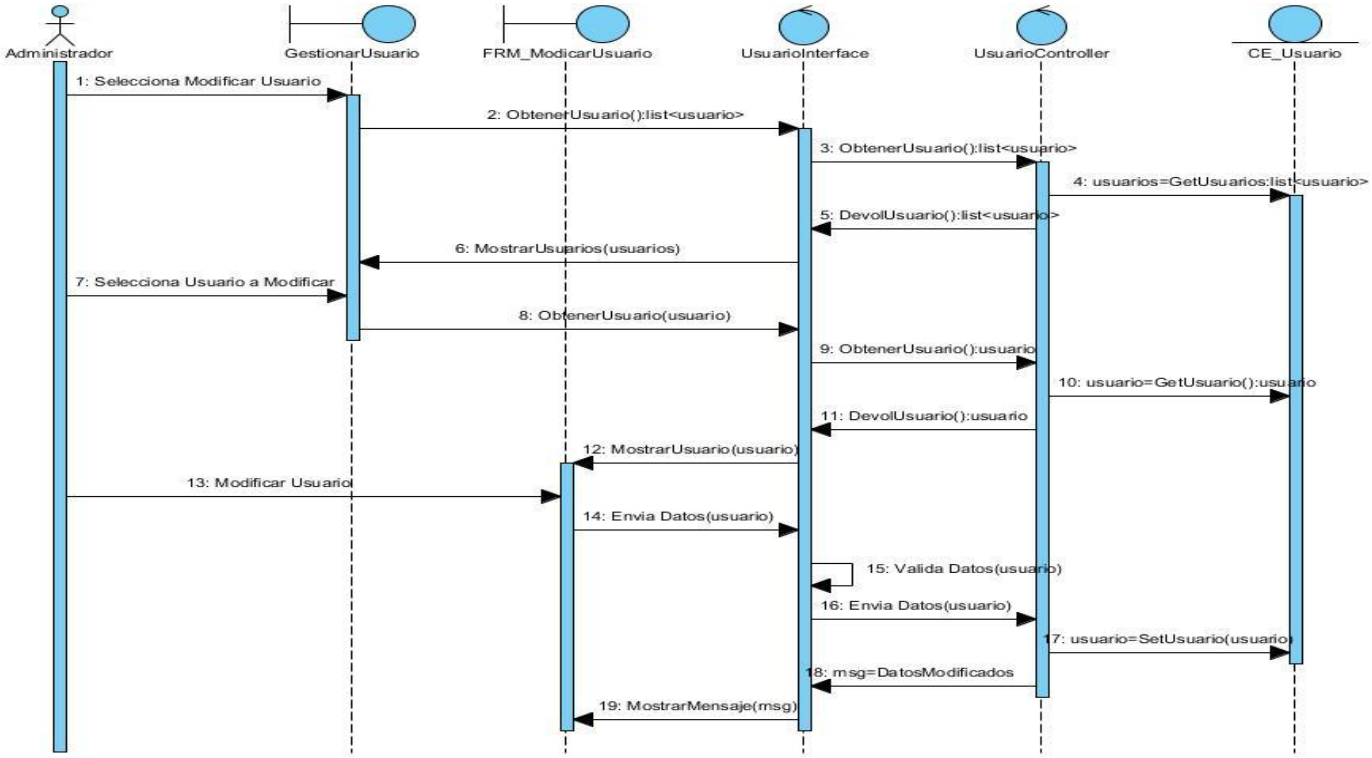
Eliminar Rol



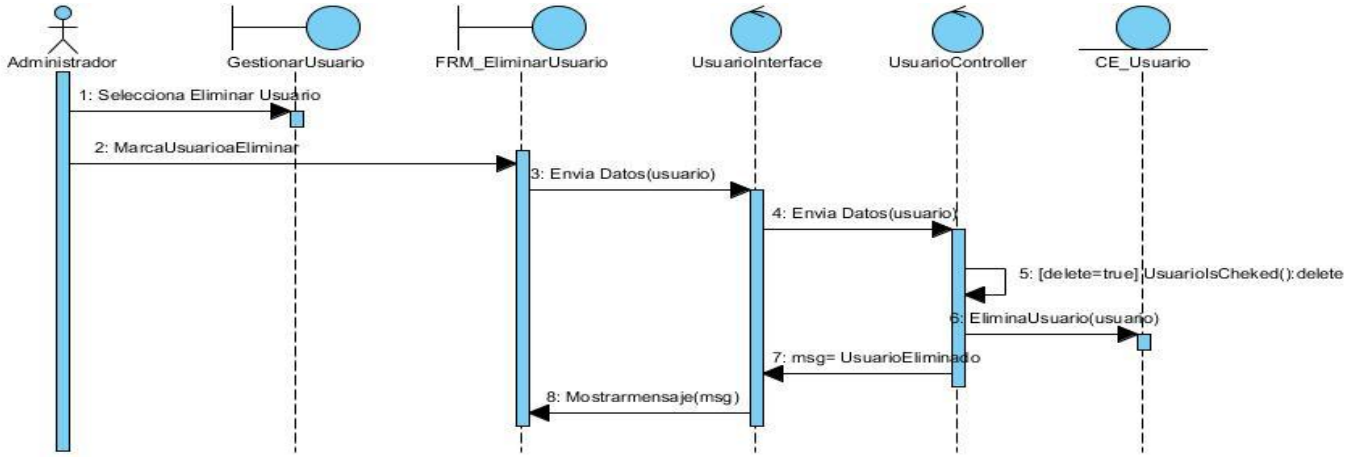
Mostrar Rol



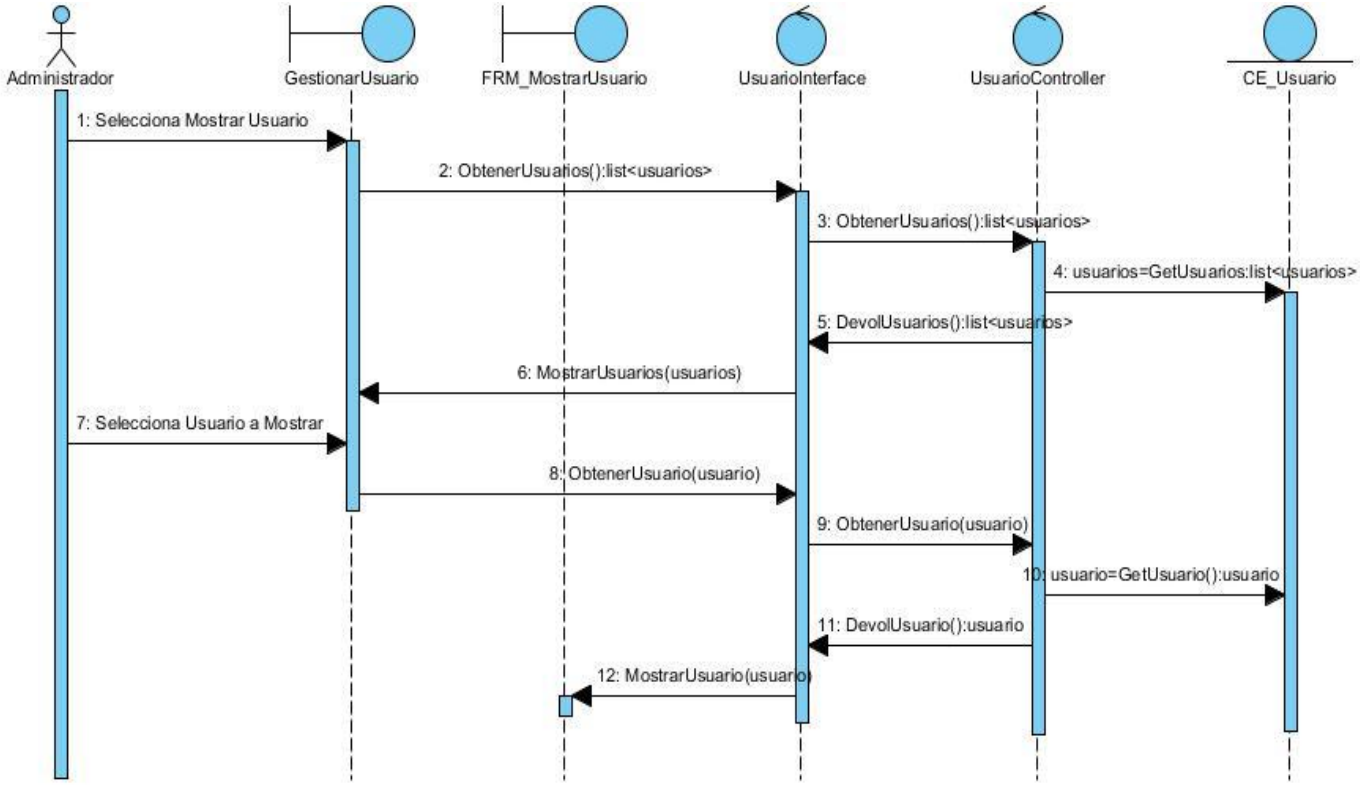
Adicionar Persona



Modificar Persona

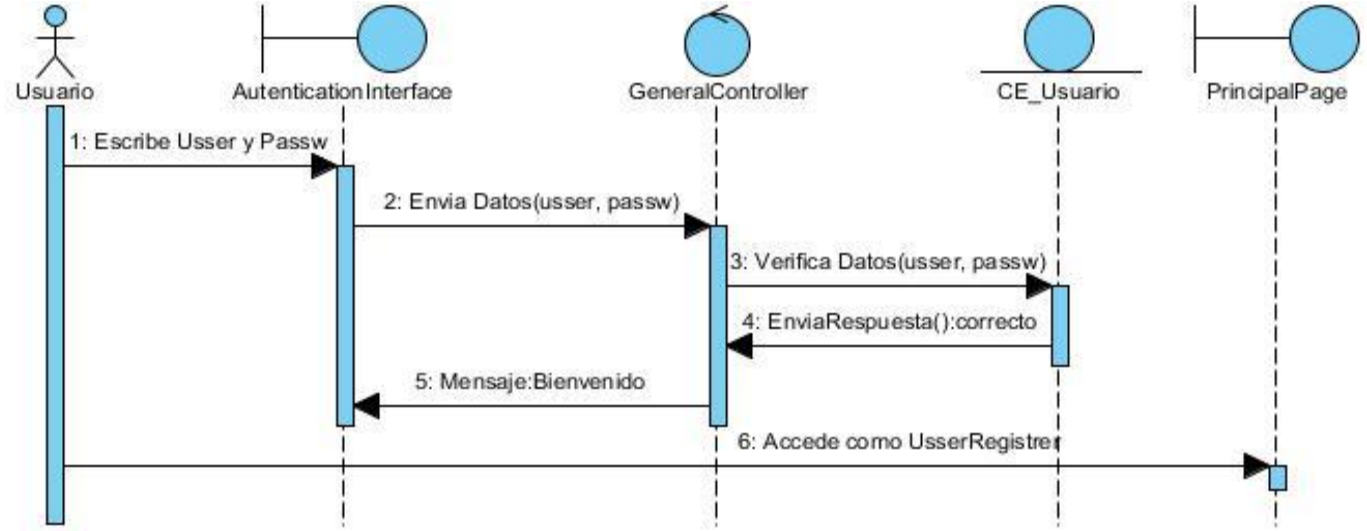


Eliminar Persona

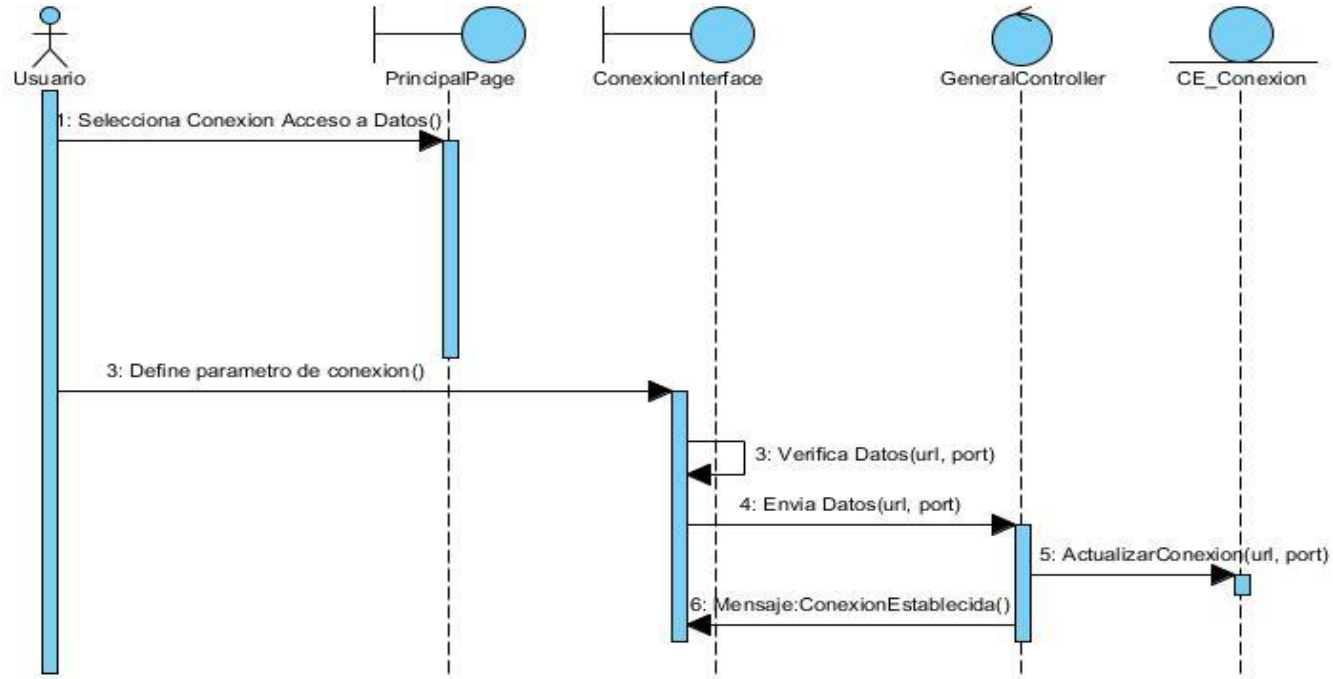


Mostrar Persona

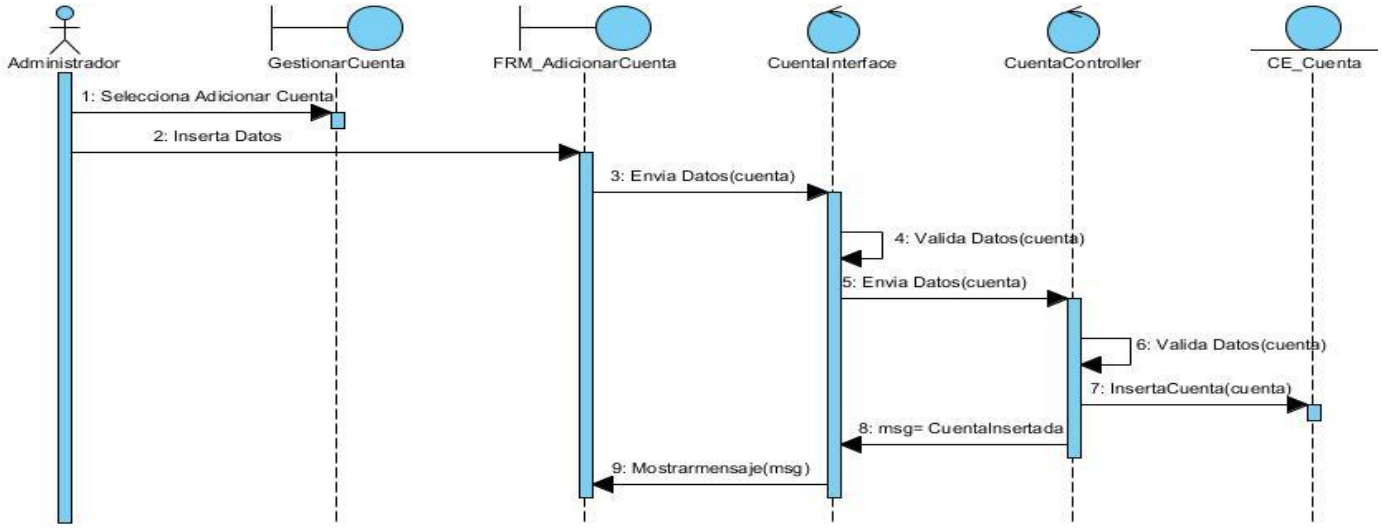
Módulo de Configuración



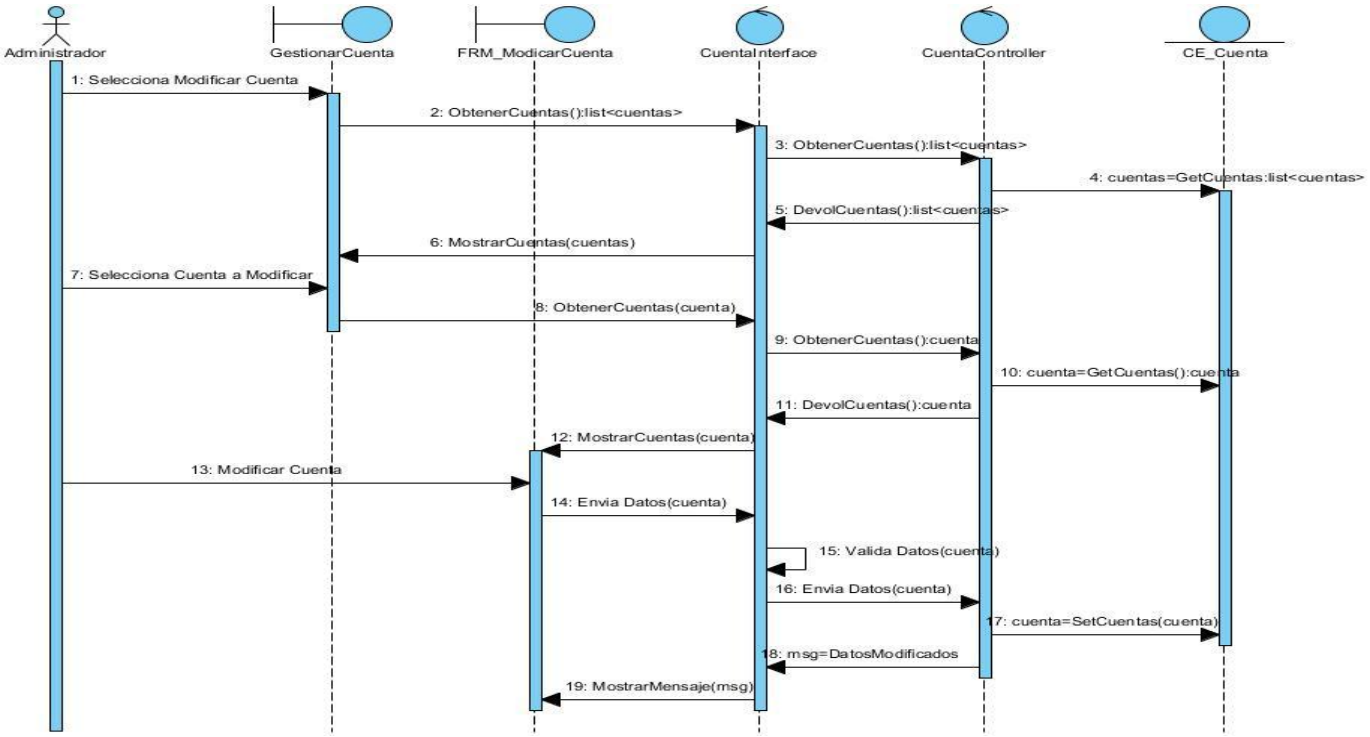
Autenticar Usuario del Sistema



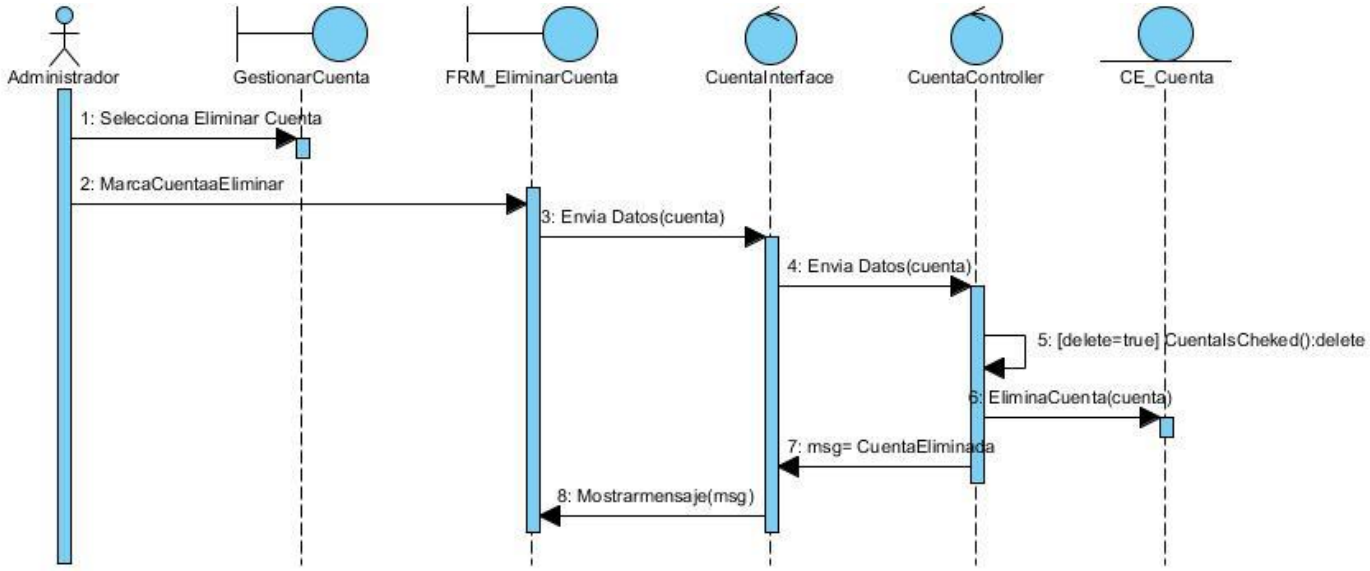
Definir Conexión de Acceso a Datos



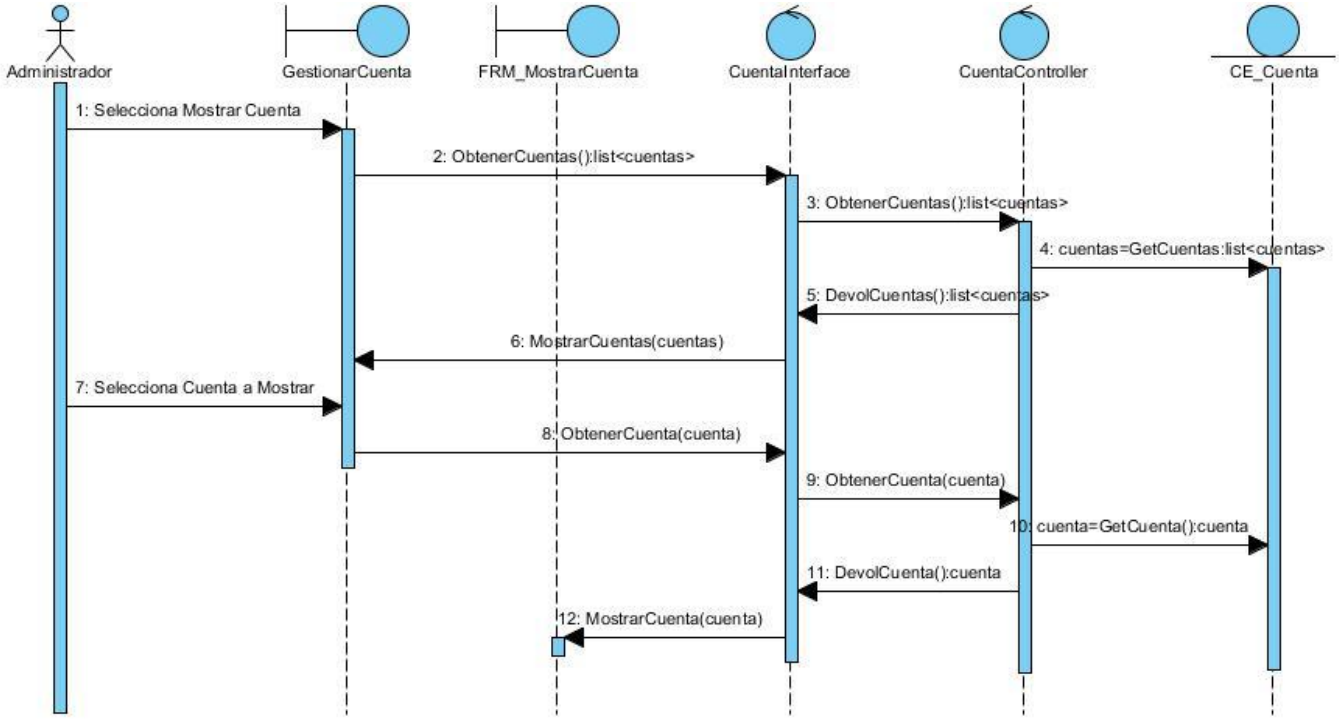
Adicionar Cuenta



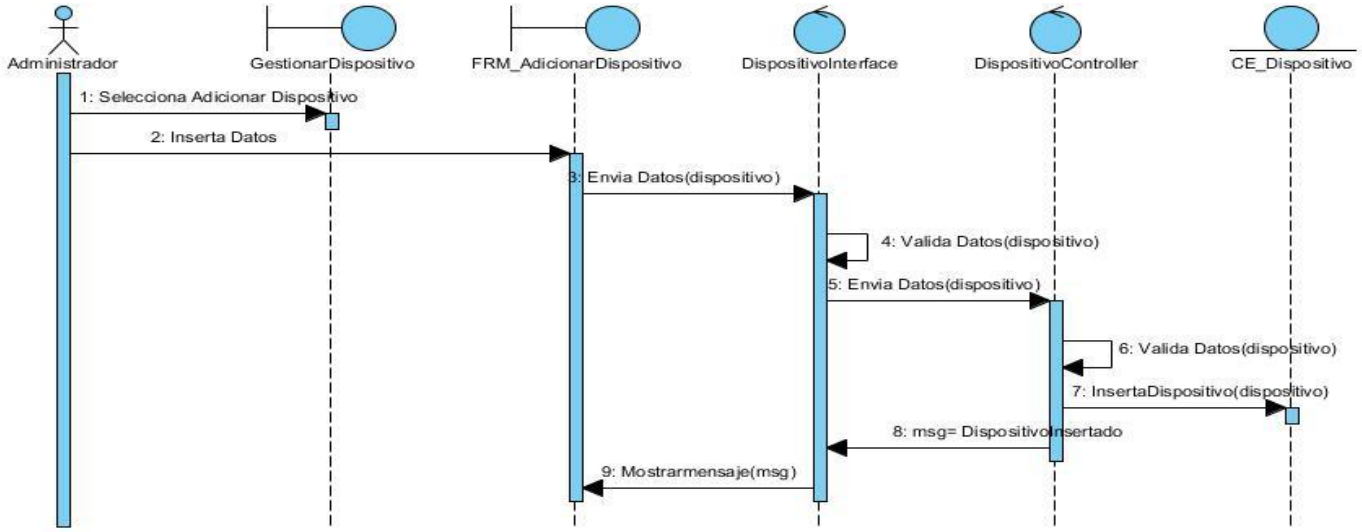
Modificar Cuenta



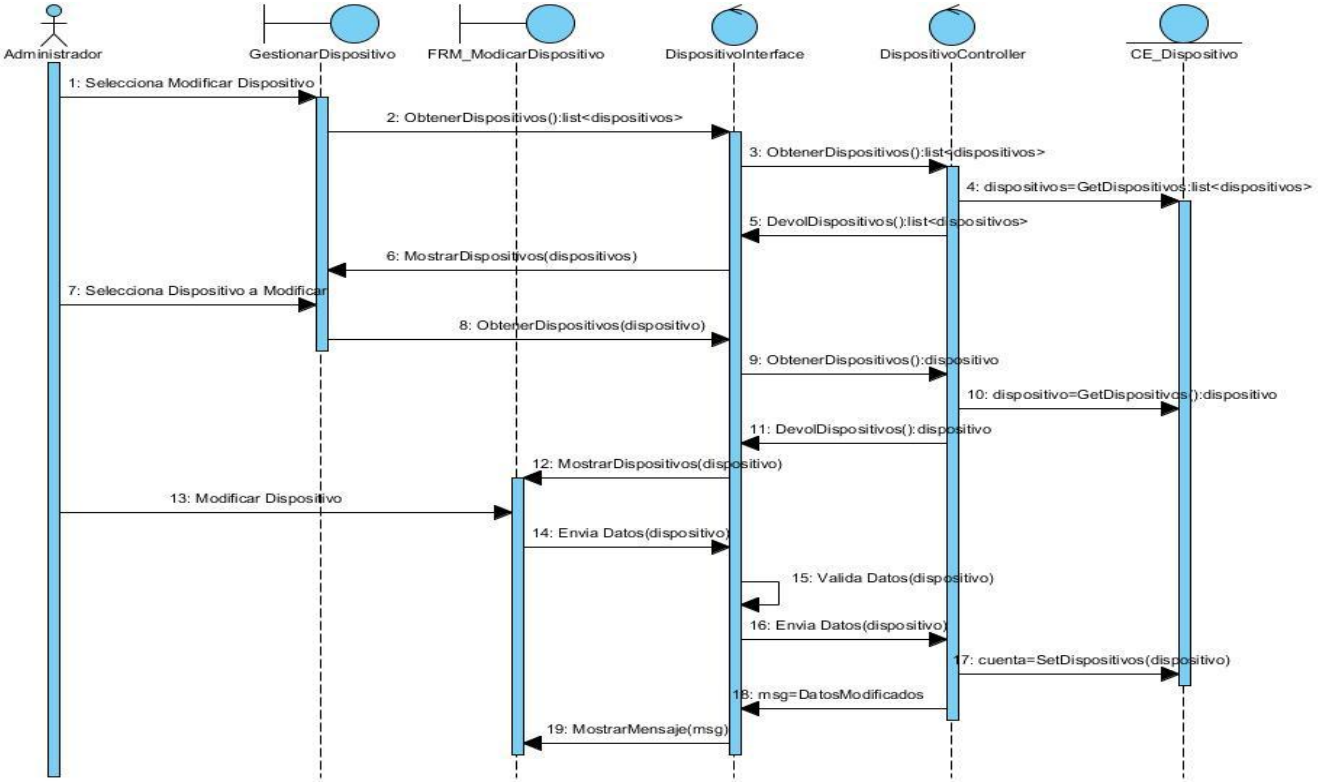
Eliminar Cuenta



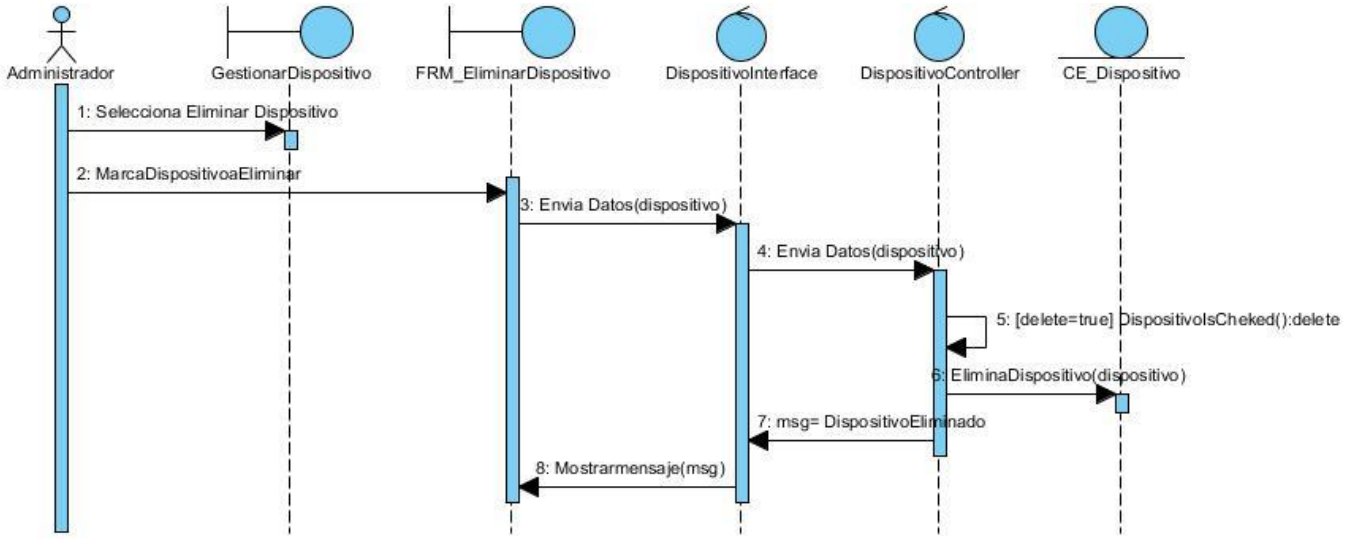
Mostrar Cuenta



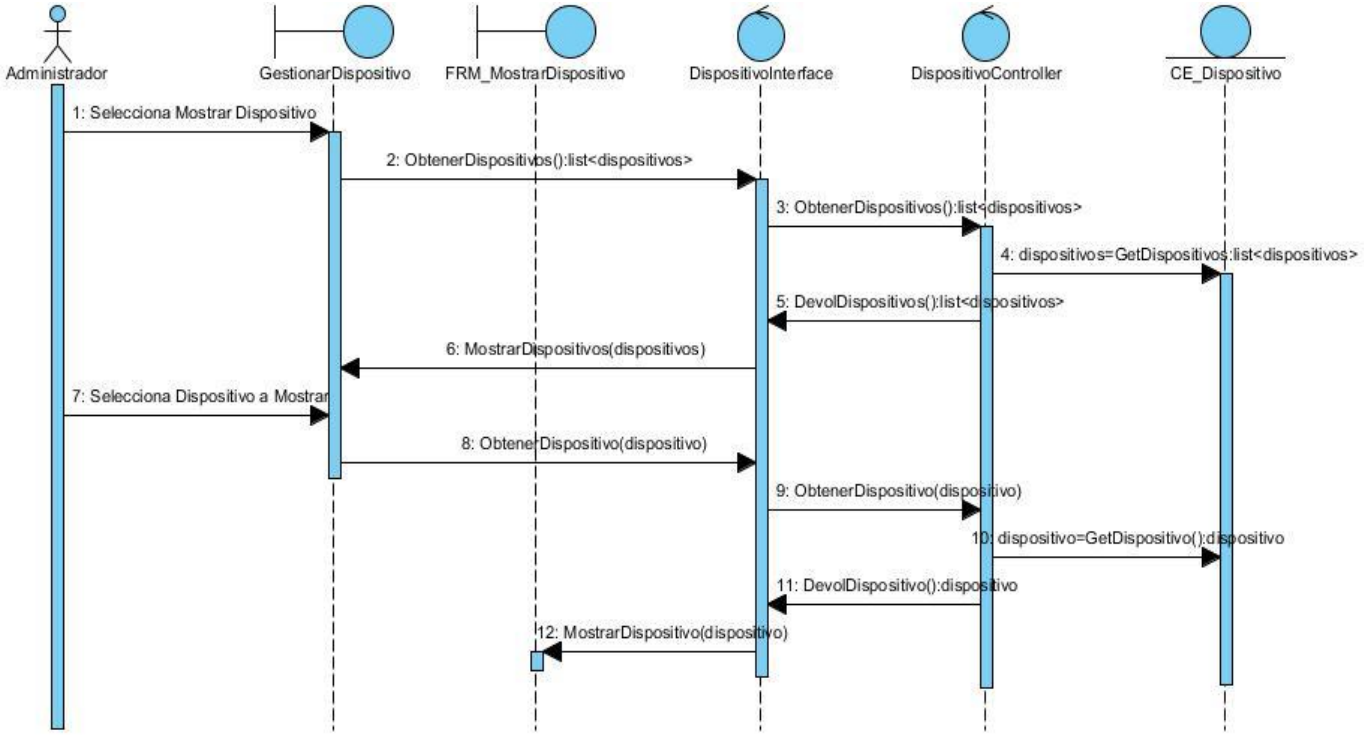
Adicionar Dispositivo



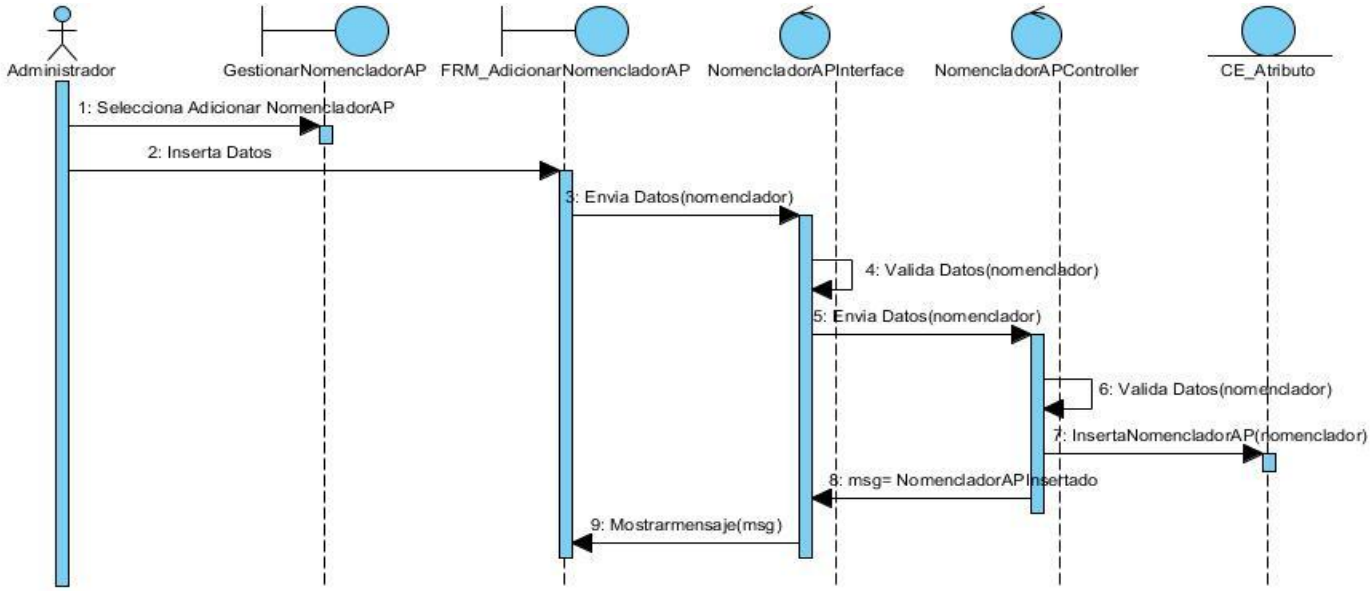
Modificar Dispositivo



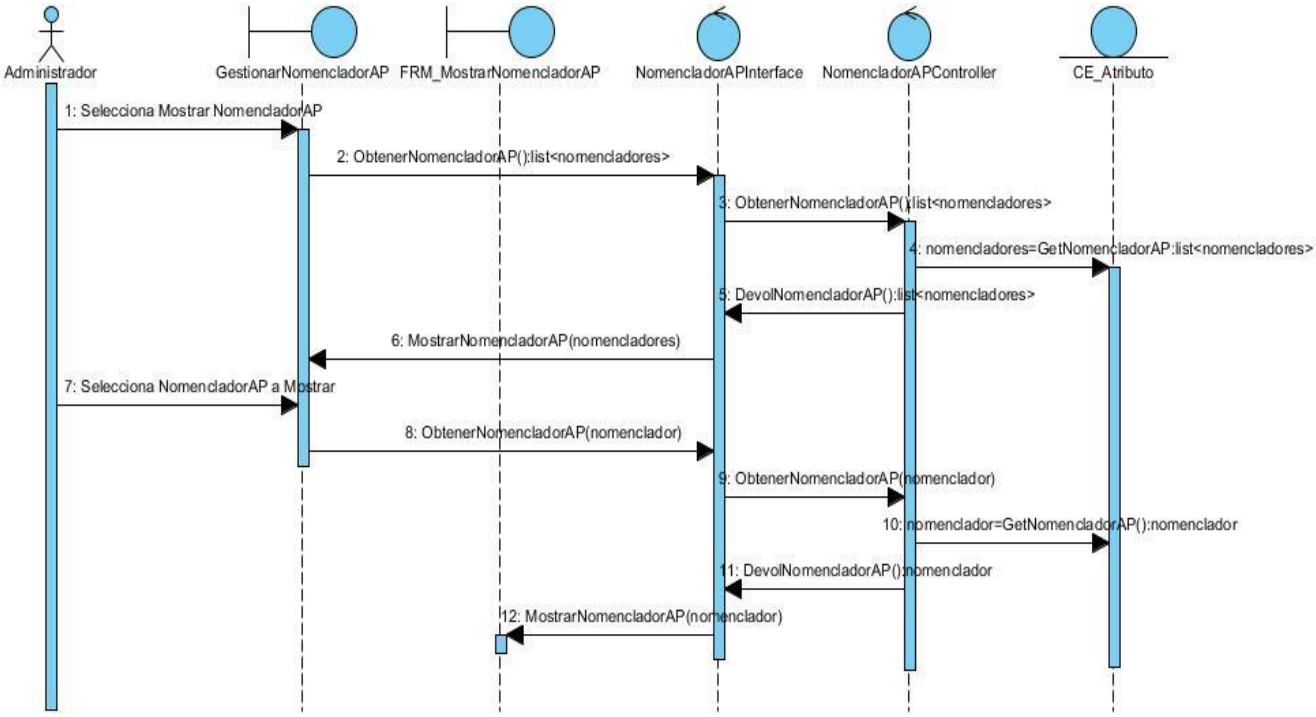
Eliminar Dispositivo



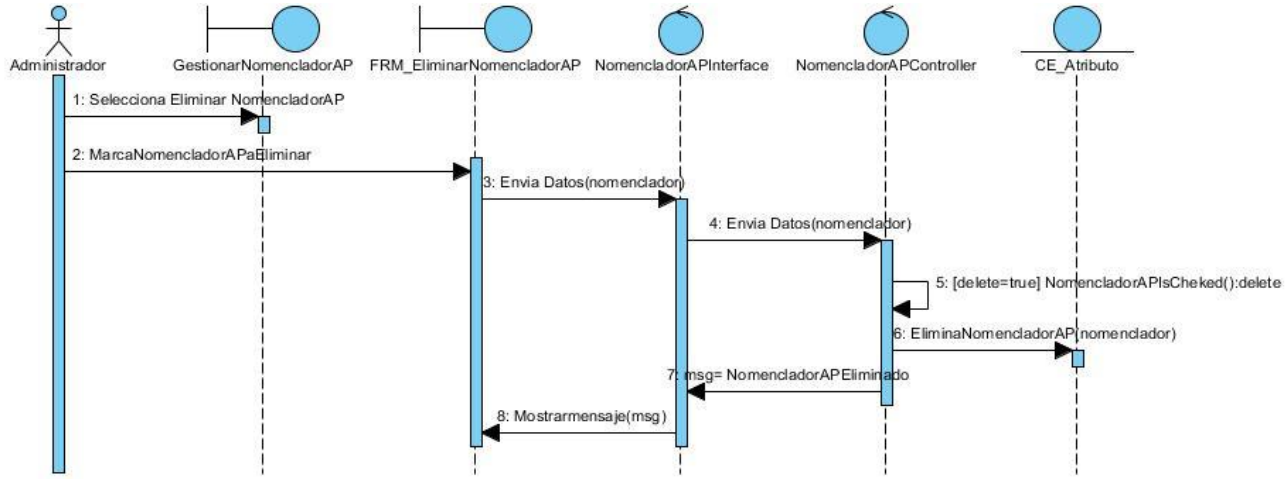
Mostrar Dispositivo



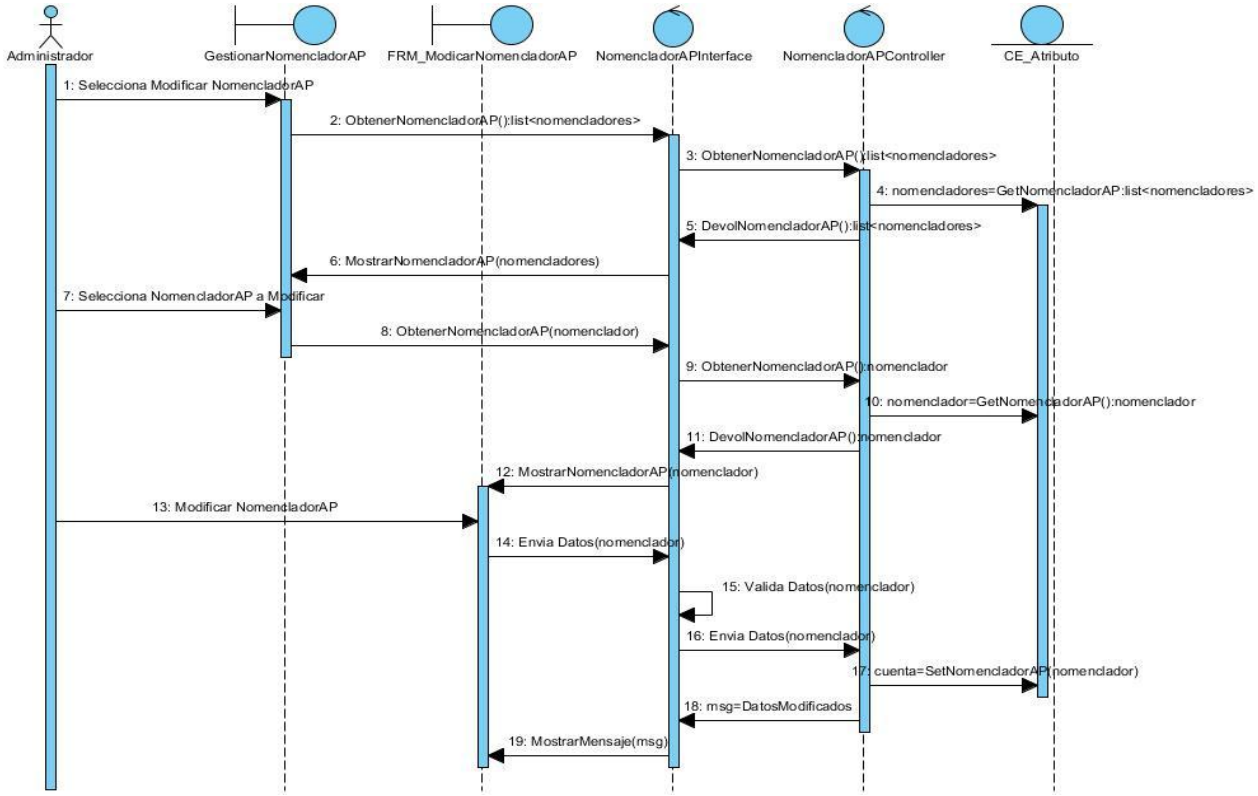
Adicionar Nomenclador Atributo



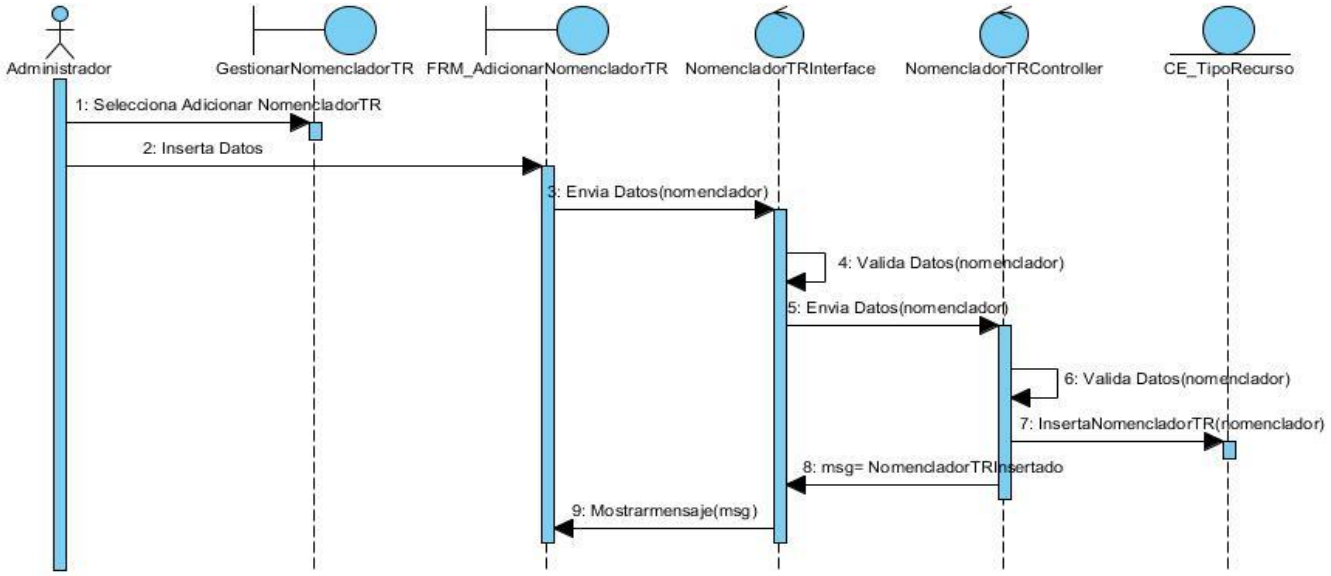
Mostrar Nomenclador Atributo



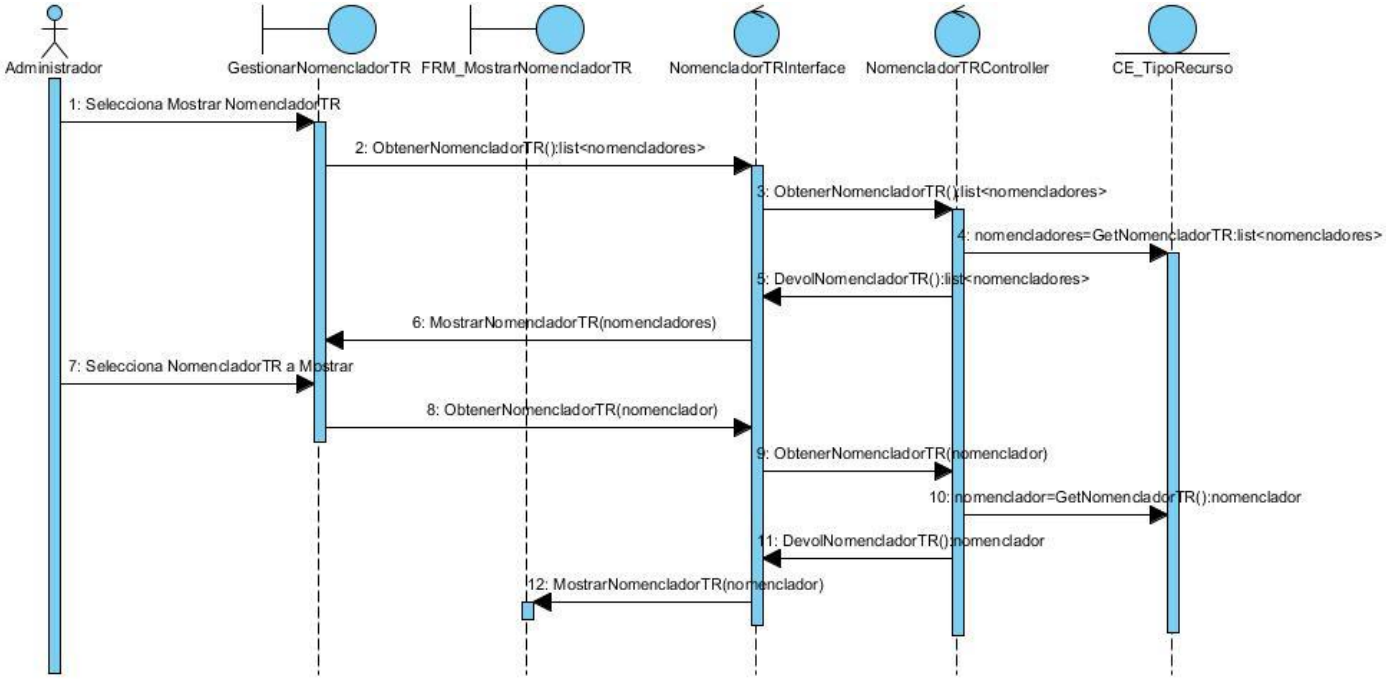
Eliminar Nomenclador Atributo



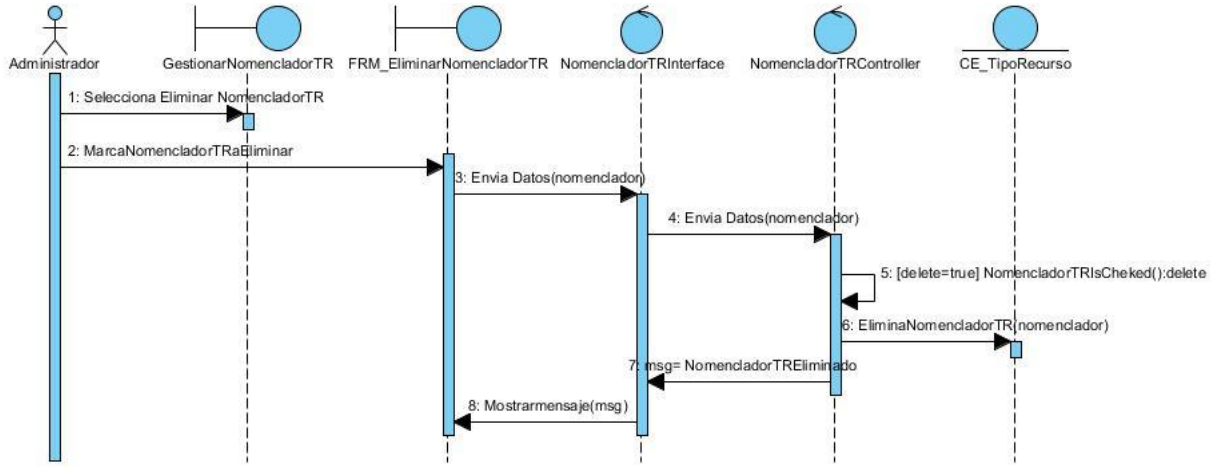
Modificar Nomenclador Atributo



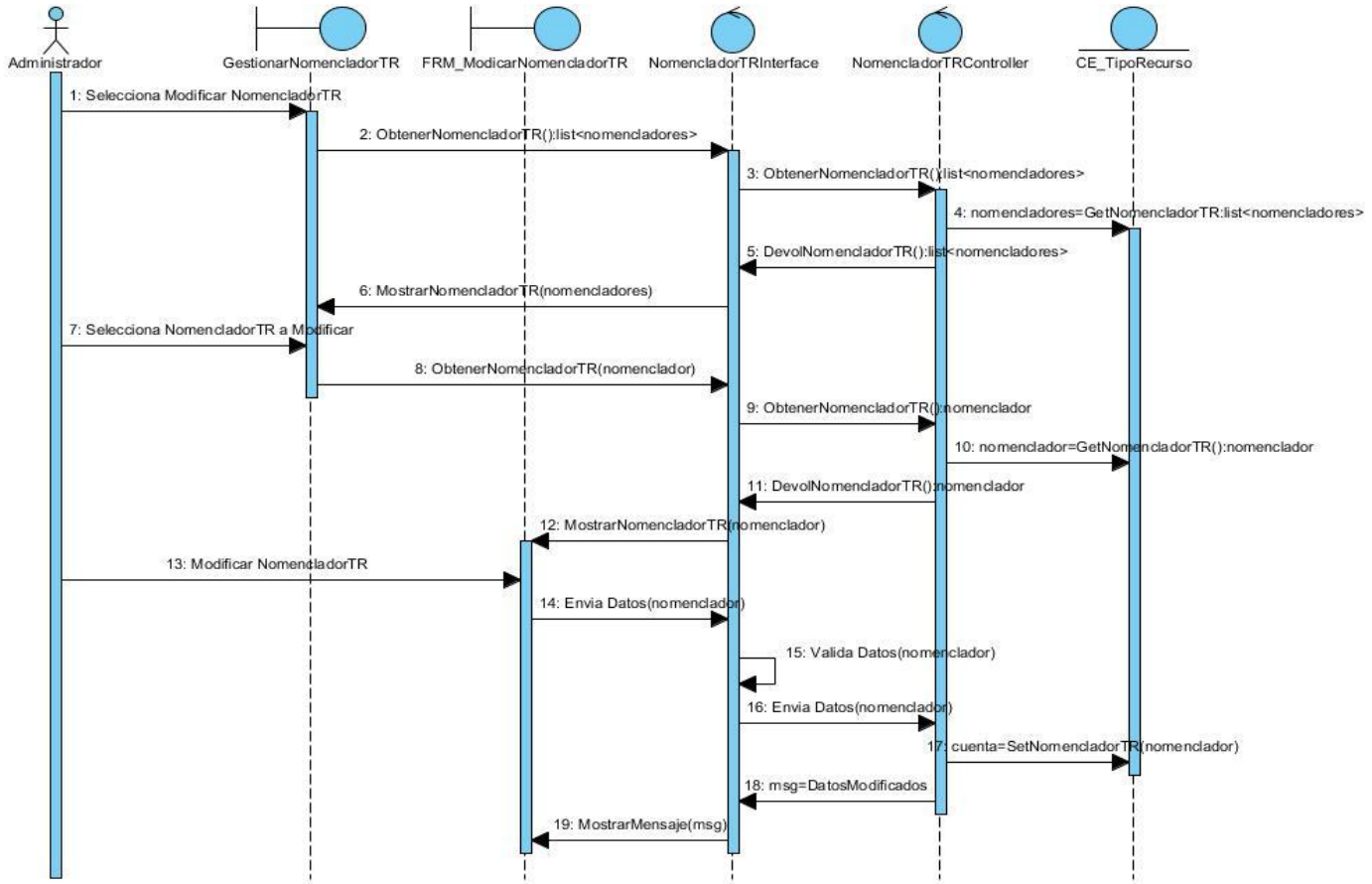
Adicionar Tipo de Recurso



Mostrar Tipo de Recurso



Eliminar Tipo de Recurso



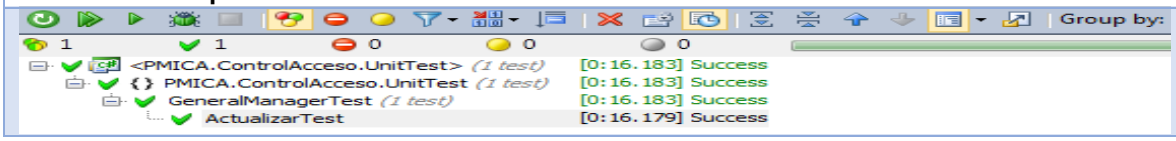
Modificar Tipo de Recurso

ANEXO 4: PRUEBAS UNITARIAS.

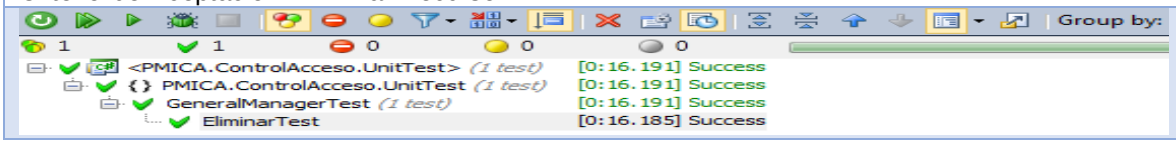
Módulo de Administración

Gestionar Recurso

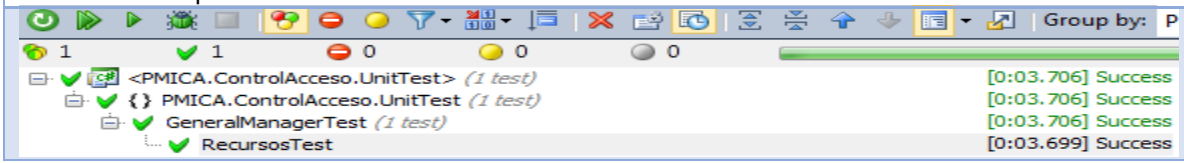
Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar el recurso de la lista de recursos y acceder a la opción "Modificar Recurso", luego se modifican los datos del mismo.		
Entrada: Recurso recurso		
Criterio de Aceptación: Modificar Recurso		



Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el recurso de la lista de recursos y acceder a la opción "Eliminar Recurso", luego se elimina el recurso del sistema.		
Entrada: Recurso recurso		
Criterio de Aceptación: Eliminar Recurso		

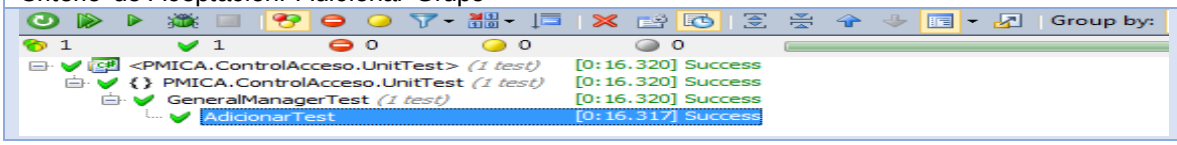


Prueba de Unidad		
Nombre Prueba: Recursos		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la opción "Listar Recurso", luego se le muestra todos los recursos existentes en el sistema.		
Entrada:		
Criterio de Aceptación: Listar Recurso		

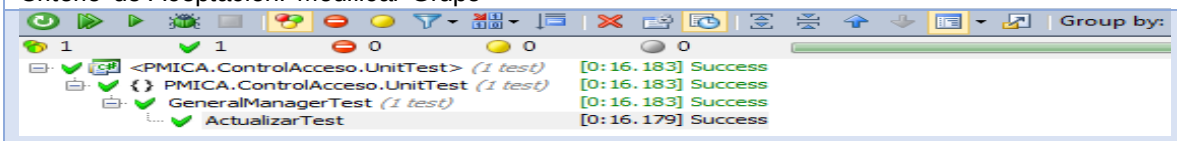


Gestionar Grupo

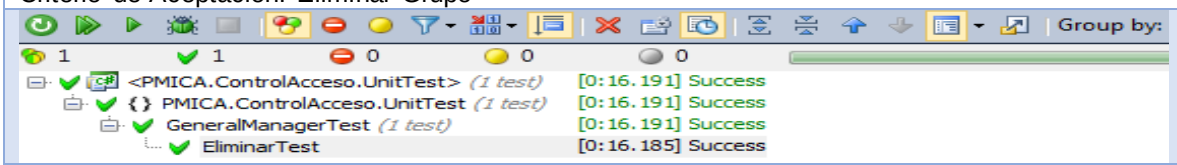
Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de grupos la opción "Adicionar Grupo", luego se llenan todos los datos que describen dicho grupo y queda el grupo existente en el sistema así como la opción para activar o desactivar el mismo.		
Entrada: Grupo grupo		
Criterio de Aceptación: Adicionar Grupo		



Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar el grupo de la lista de grupos y acceder a la opción "Modificar Grupo", luego se modifican los datos del mismo.		
Entrada: Grupo grupo		
Criterio de Aceptación: Modificar Grupo		

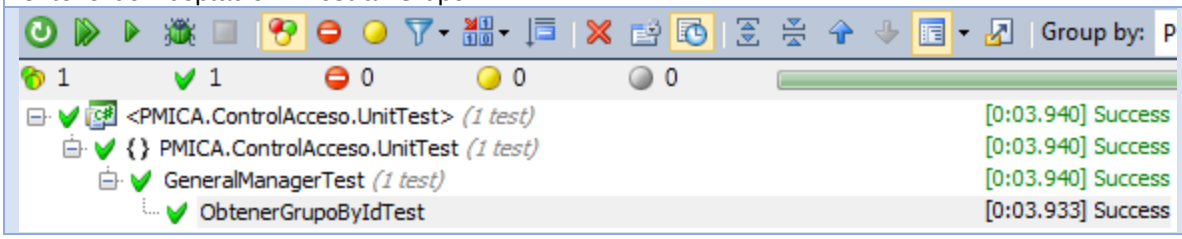


Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el grupo de la lista de grupos y acceder a la opción "Eliminar Grupo", luego se elimina el grupo del sistema.		
Entrada: Grupo grupo		
Criterio de Aceptación: Eliminar Grupo		

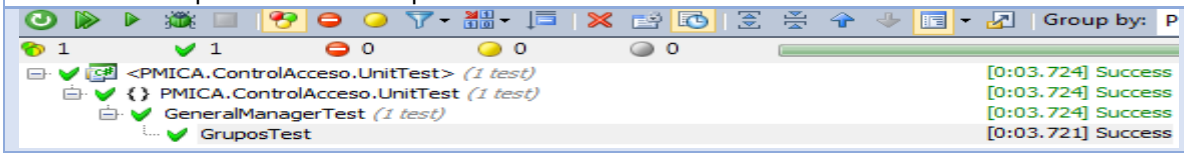


Prueba de Unidad		
Nombre Prueba: ObtenerGrupoByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013

Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán
Descripción: Para la realización de esta prueba se debe seleccionar el grupo de la lista de grupos y acceder a la opción "Mostrar Grupo", luego se le muestra todos los detalles del grupo.	
Entrada: Guid idGrupo	
Criterio de Aceptación: Mostrar Grupo	

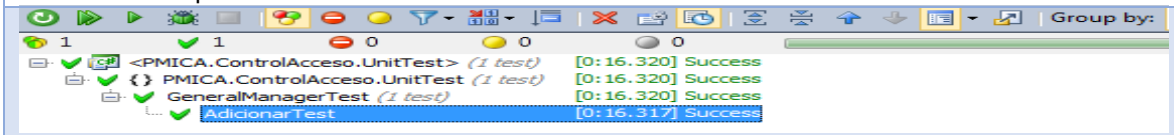


Prueba de Unidad		
Nombre Prueba: Grupos		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la opción "Listar Grupo", luego se le muestra todos los grupos existentes en el sistema.		
Entrada:		
Criterio de Aceptación: Listar Grupo		



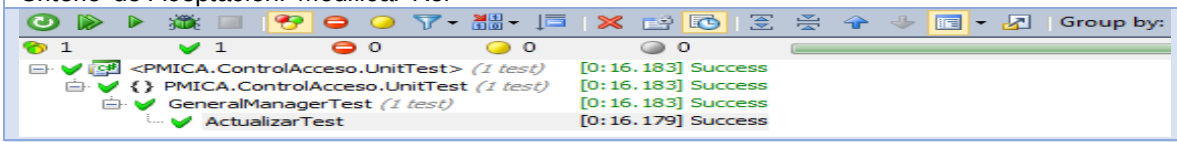
Gestionar Rol

Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de roles la opción "Adicionar Rol", luego se llenan todos los datos que describen dicho rol y queda el rol existente en el sistema así como la opción para activar o desactivar el mismo.		
Entrada: Rol rol		
Criterio de Aceptación: Adicionar Rol		

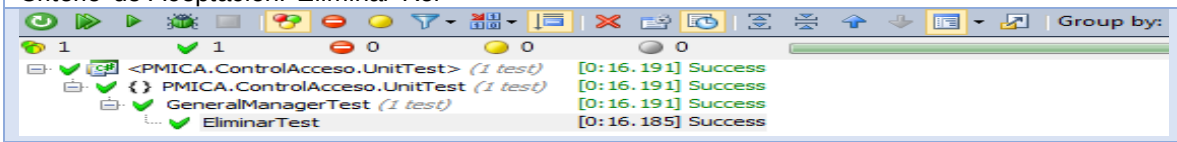


Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar el rol de la lista de roles y acceder a la opción "Modificar Rol", luego se modifican los datos del mismo.		
Entrada: Rol rol		

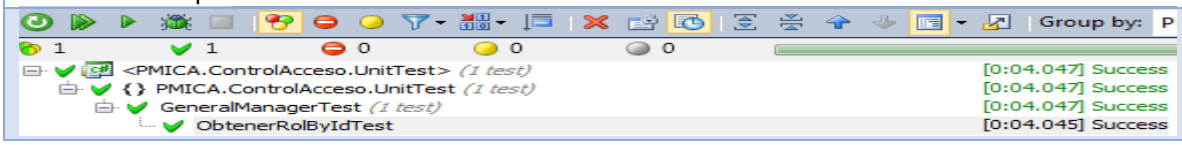
Criterio de Aceptación: Modificar Rol



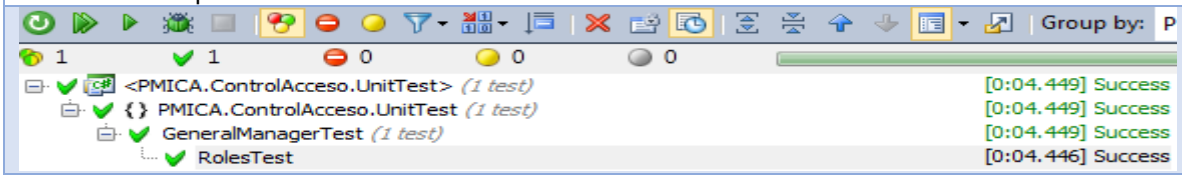
Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el rol de la lista de roles y acceder a la opción "Eliminar Rol", luego se elimina el rol del sistema.		
Entrada: Rol rol		
Criterio de Aceptación: Eliminar Rol		



Prueba de Unidad		
Nombre Prueba: ObtenerRolByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el rol de la lista de roles y acceder a la opción "Mostrar Rol", luego se le muestra todos los detalles del rol.		
Entrada: Guid idRol		
Criterio de Aceptación: Mostrar Rol		

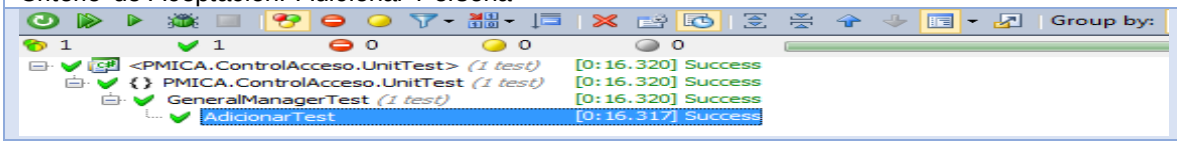


Prueba de Unidad		
Nombre Prueba: Roles		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la opción "Listar Roles", luego se le muestra todos los roles existentes en el sistema.		
Entrada:		
Criterio de Aceptación: Listar Roles		

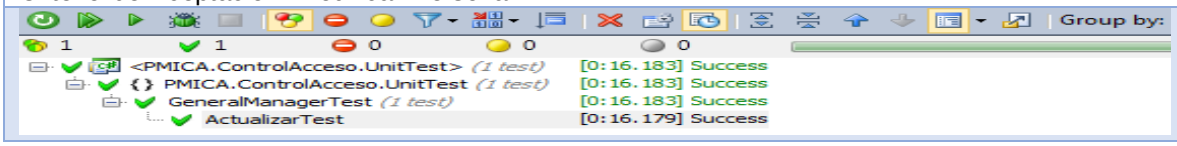


Gestionar Persona

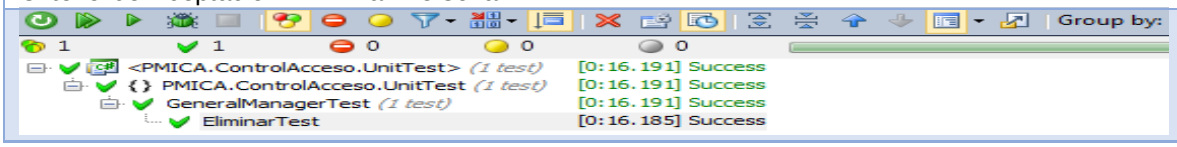
Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de personas la opción "Adicionar Persona", luego se llenan todos los datos que describen dicha persona quedando registrada en el sistema.		
Entrada: Persona persona		
Criterio de Aceptación: Adicionar Persona		



Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar la persona de la lista de personas y acceder a la opción "Modificar Persona", luego se modifican los datos de la misma.		
Entrada: Persona persona		
Criterio de Aceptación: Modificar Persona		

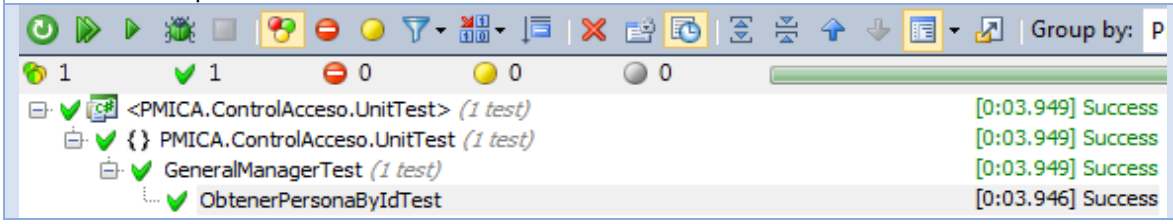


Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la persona de la lista de personas y acceder a la opción "Eliminar Persona", luego se elimina la persona del sistema.		
Entrada: Persona persona		
Criterio de Aceptación: Eliminar Persona		

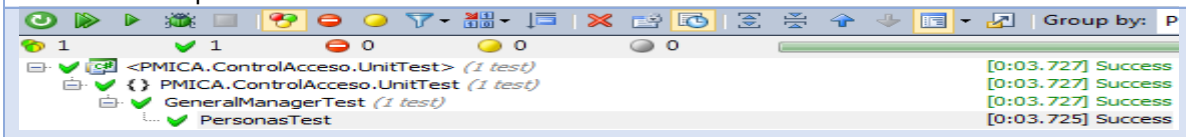


Prueba de Unidad		
Nombre Prueba: ObtenerPersonaByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el usuario de la lista de personas y acceder a la opción "Mostrar Persona", luego se le muestra todos los detalles de la		

persona.
Entrada: Guid idPersona
Criterio de Aceptación: Mostrar Persona

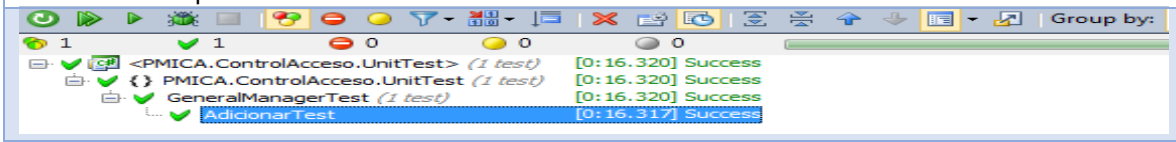


Prueba de Unidad		
Nombre Prueba: Personas		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la opción "Listar Personas", luego se le muestra todas las personas existentes en el sistema.		
Entrada:		
Criterio de Aceptación: Listar Personas		



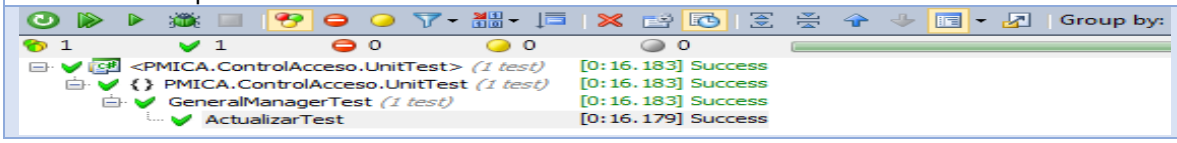
Gestionar Restricción

Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de restricciones la opción "Adicionar Restricción", luego se llenan todos los datos que describen dicha restricción y queda la restricción existente en el sistema así como la opción para activar o desactivar la misma.		
Entrada: Restriccion restriccion		
Criterio de Aceptación: Adicionar Restricción		

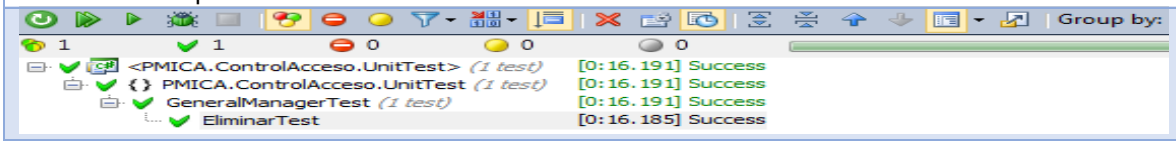


Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar la restricción de la lista de restricciones y acceder a la opción "Modificar Restricción", luego se modifican los datos del mismo.		

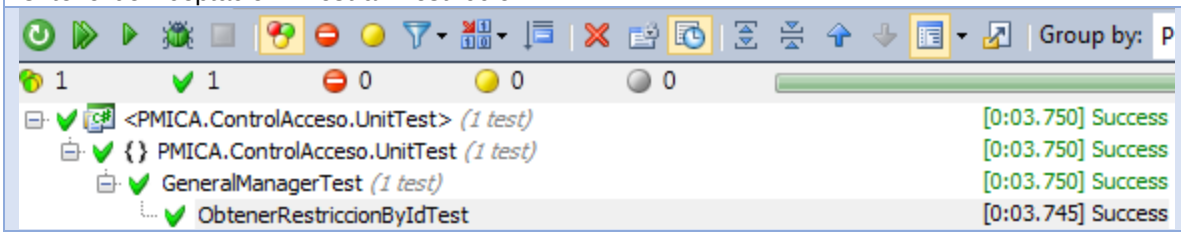
Entrada: Restriccion restriccion
Criterio de Aceptación: Modificar Restricción



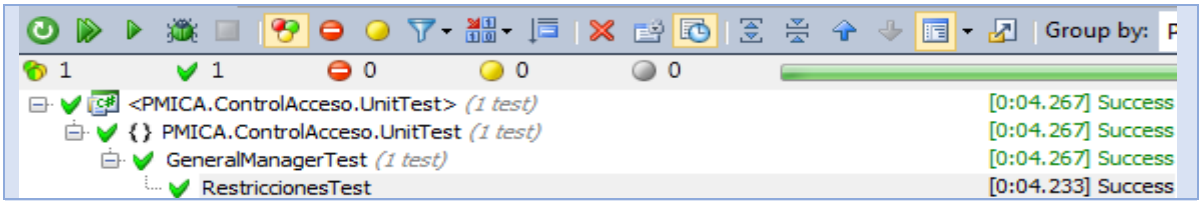
Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la restricción de la lista de restricciones y acceder a la opción "Eliminar Restricción", luego se elimina la restricción del sistema.		
Entrada: Restriccion restriccion		
Criterio de Aceptación: Eliminar Restricción		



Prueba de Unidad		
Nombre Prueba: ObtenerRestriccionByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la restricción de la lista de restricciones y acceder a la opción "Mostrar Restricción", luego se le muestra todos los detalles de la restricción.		
Entrada: Guid idRestriccion		
Criterio de Aceptación: Mostrar Restricción		

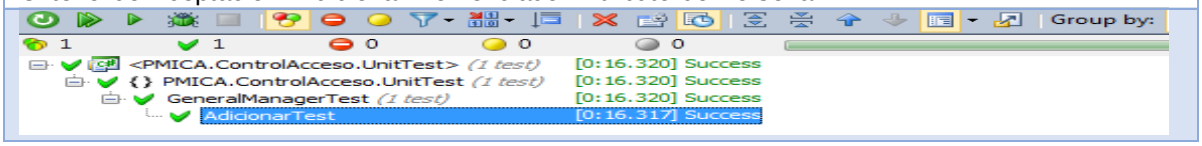


Prueba de Unidad		
Nombre Prueba: Restricciones		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la opción "Listar Restricciones", luego se le muestra todas las restricciones existentes en el sistema.		
Entrada:		
Criterio de Aceptación: Listar Restricciones		

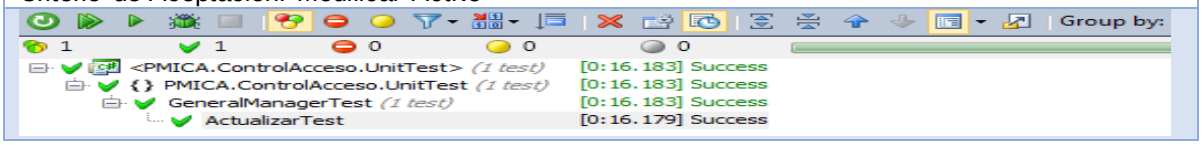


Gestionar Activo

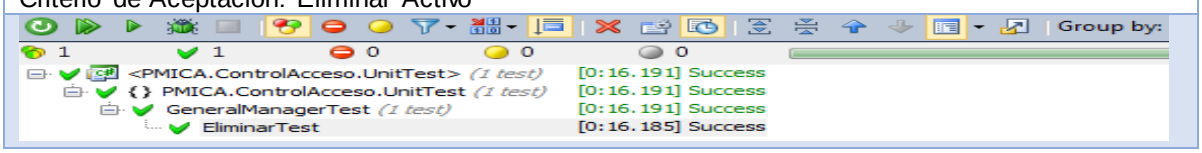
Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de Activos la opción "Adicionar Activo", luego se llenan todos los datos que describen dicho activo y queda el activo existente en el sistema así como la opción para activar o desactivar el mismo.		
Entrada: Activo activo		
Criterio de Aceptación: Adicionar Nomenclador Atributo de Persona		



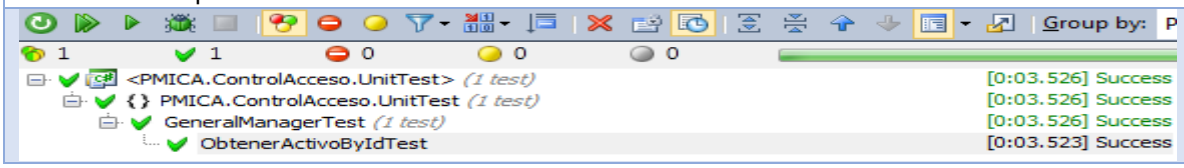
Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar el activo que desea modificar y acceder a la opción "Modificar Activo ", luego se modifican los datos del mismo.		
Entrada: Activo activo		
Criterio de Aceptación: Modificar Activo		



Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el activo que desea eliminar y acceder a la opción "Eliminar Activo ", luego se elimina el activo del sistema.		
Entrada: Activo activo		
Criterio de Aceptación: Eliminar Activo		



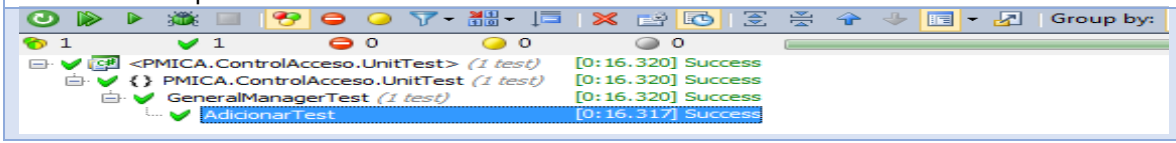
Prueba de Unidad		
Nombre Prueba: ObtenerActivoByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el activo que desea detallar y acceder a la opción "Mostrar Activo ", luego se le muestra todos los detalles del activo.		
Entrada: Guid idActivo		
Criterio de Aceptación: Mostrar Activo		



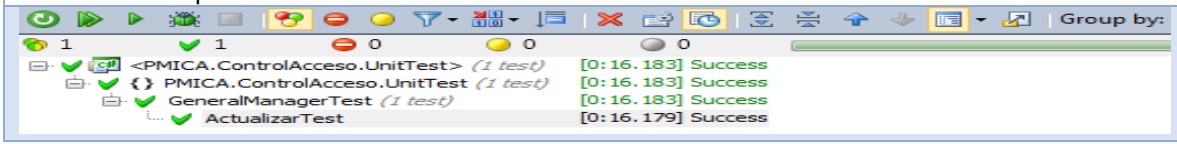
Módulo de Configuración

Gestionar Cuenta

Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de cuentas la opción "Adicionar Cuenta", luego se llenan todos los datos que describen dicha cuenta y queda la cuenta existente en el sistema así como la opción para activar o desactivar la misma.		
Entrada: Cuenta cuenta		
Criterio de Aceptación: Adicionar Cuenta		

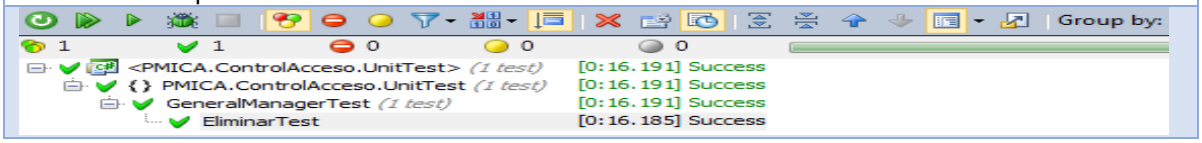


Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar la cuenta de la lista de cuentas y acceder a la opción "Modificar Cuenta", luego se modifican los datos de la misma.		
Entrada: Cuenta cuenta		
Criterio de Aceptación: Modificar Cuenta		

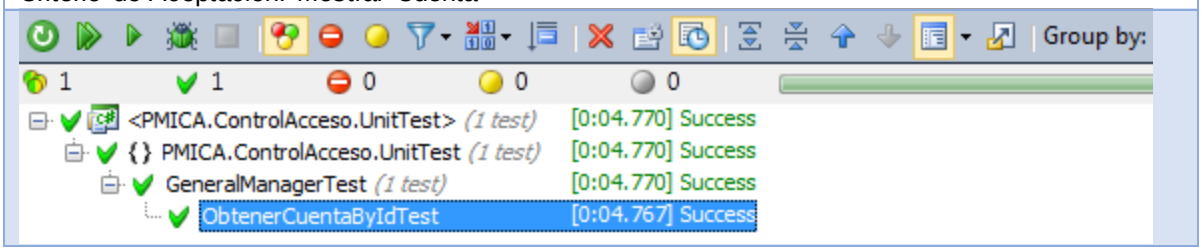


Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la cuenta de la lista de		

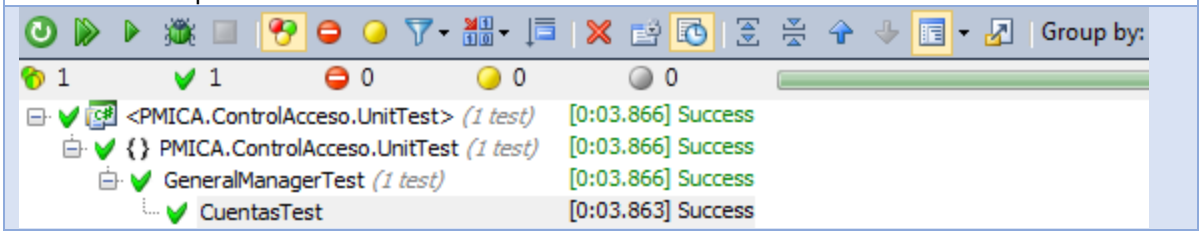
cuentas y acceder a la opción "Eliminar Cuenta", luego se elimina la cuenta del sistema.
Entrada: Cuenta cuenta
Criterio de Aceptación: Eliminar Cuenta



Prueba de Unidad		
Nombre Prueba: ObtenerCuentaByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la cuenta de la lista de cuentas y acceder a la opción "Mostrar Cuenta", luego se le muestra todos los detalles de la cuenta.		
Entrada: Guid idCuenta		
Criterio de Aceptación: Mostrar Cuenta		



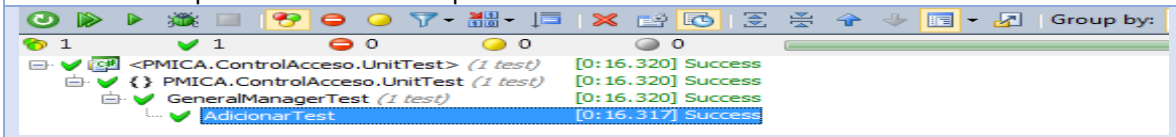
Prueba de Unidad		
Nombre Prueba: Cuentas		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la opción "Listar Cuentas", luego se le muestra todas las cuentas existentes en el sistema.		
Entrada:		
Criterio de Aceptación: Listar Cuenta		



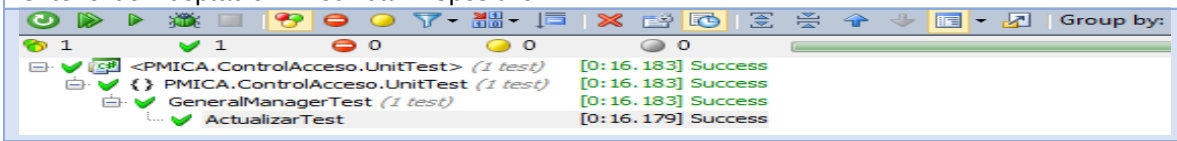
Gestionar Dispositivos

Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de dispositivos la opción "Adicionar Dispositivo", luego se llenan todos los datos que describen dicho dispositivo quedando registrado en el sistema.		
Entrada:		

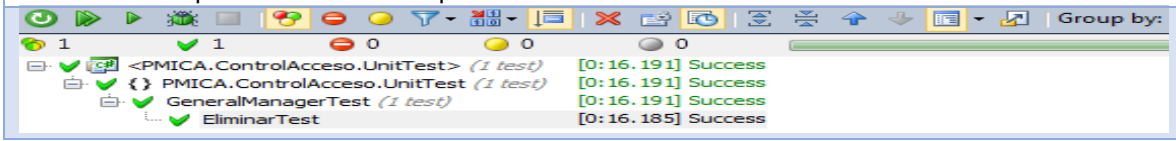
Criterio de Aceptación: Adicionar Dispositivo



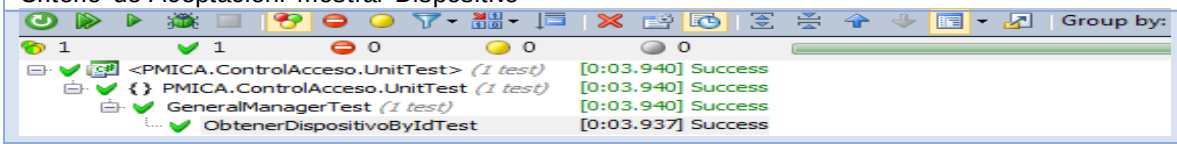
Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	Ultima Ejecución:15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar el dispositivo de la lista de dispositivos que desea modificar y acceder a la opción "Modificar Dispositivo", luego se modifican los datos del mismo.		
Entrada:		
Criterio de Aceptación: Modificar Dispositivo		



Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el dispositivo de la lista de dispositivos que desea eliminar y acceder a la opción "Eliminar Dispositivo", luego se elimina el dispositivo del sistema.		
Entrada:		
Criterio de Aceptación: Eliminar Dispositivo		

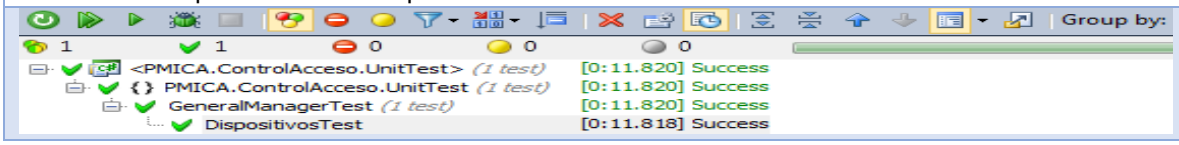


Prueba de Unidad		
Nombre Prueba: ObtenerDispositivoByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el dispositivo de la lista de dispositivos que desea mostrar y acceder a la opción "Mostrar Dispositivo", luego se le muestra todos los detalles del dispositivo.		
Entrada:		
Criterio de Aceptación: Mostrar Dispositivo		



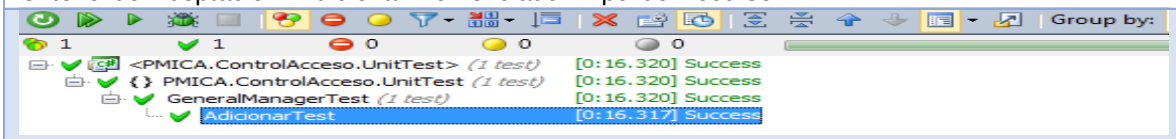
Prueba de Unidad

Nombre Prueba: Dispositivos		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la opción "Listar Dispositivo", luego se le muestran todos los dispositivos existentes en el sistema.		
Entrada:		
Criterio de Aceptación: Listar Dispositivo		

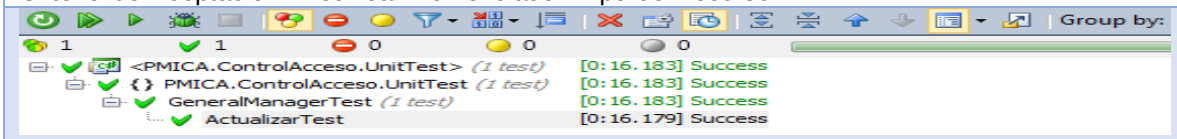


Gestionar Nomenclador Tipo de Recurso

Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de Nomencladores Tipo de Recurso la opción "Adicionar NomencladorTR", luego se llenan todos los datos que describen dicho nomenclador y queda el nomenclador existente en el sistema así como la opción para activar o desactivar el mismo.		
Entrada: TipoRecurso tipoRecurso		
Criterio de Aceptación: Adicionar Nomenclador Tipo de Recurso		

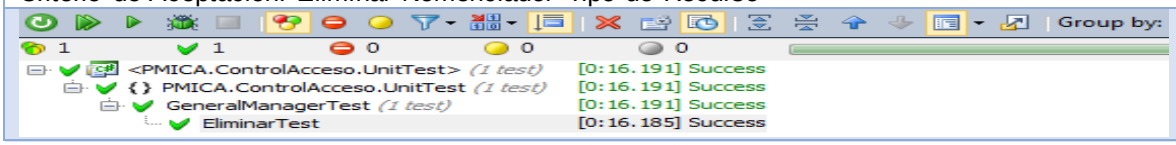


Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar el nomenclador que desea modificar y acceder a la opción "Modificar NomencladorTR", luego se modifican los datos del mismo.		
Entrada: TipoRecurso tipoRecurso		
Criterio de Aceptación: Modificar Nomenclador Tipo de Recurso		

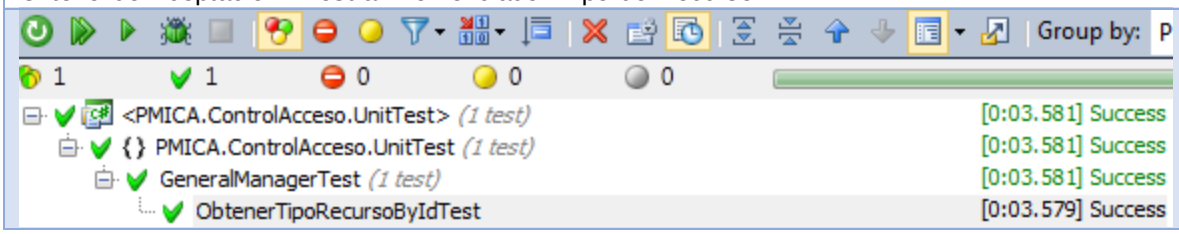


Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el nomenclador que desea eliminar y acceder a la opción "Eliminar NomencladorTR", luego se elimina el nomenclador del sistema.		
Entrada: TipoRecurso tipoRecurso		

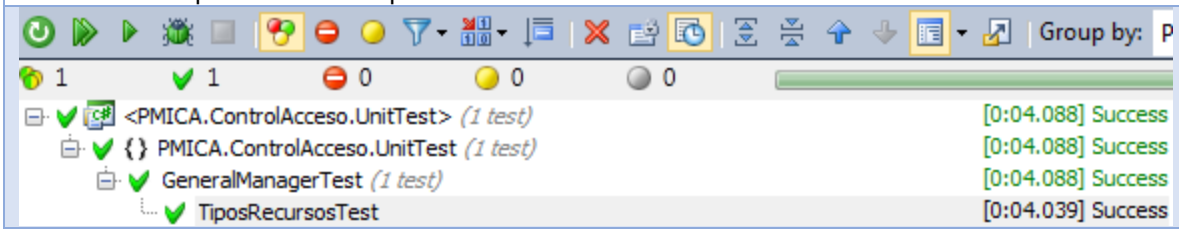
Criterio de Aceptación: Eliminar Nomenclador Tipo de Recurso



Prueba de Unidad		
Nombre Prueba: ObtenerTipoRecursoByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el nomenclador que desea detallar y acceder a la opción "Mostrar NomencladorTR", luego se le muestra todos los detalles del nomenclador.		
Entrada: Guid idTipoRecurso		
Criterio de Aceptación: Mostrar Nomenclador Tipo de Recurso		

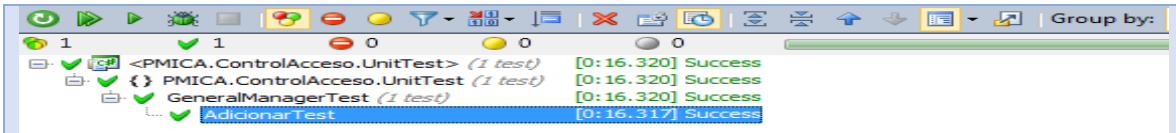


Prueba de Unidad		
Nombre Prueba: Tipos de Recursos		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar la opción "Listar Tipos de Recursos", luego se le muestra todos los tipos de recursos existentes en el sistema.		
Entrada:		
Criterio de Aceptación: Listar Tipos de Recursos		

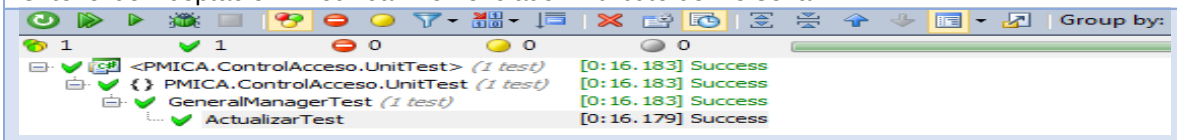


Gestionar Nomenclador Atributo de Persona

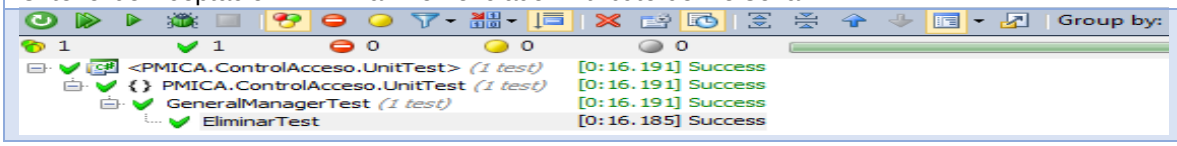
Prueba de Unidad		
Nombre Prueba: Adicionar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar en la interfaz para la gestión de Nomencladores Atributo de Persona la opción "Adicionar NomencladorAP", luego se llenan todos los datos que describen dicho nomenclador y queda el nomenclador existente en el sistema así como la opción para activar o desactivar el mismo.		
Entrada: Atributo atributo		
Criterio de Aceptación: Adicionar Nomenclador Atributo de Persona		



Prueba de Unidad		
Nombre Prueba: Actualizar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la ejecución de esta prueba se debe seleccionar el nomenclador que desea modificar y acceder a la opción "Modificar NomencladorAP", luego se modifican los datos del mismo.		
Entrada: Atributo atributo		
Criterio de Aceptación: Modificar Nomenclador Atributo de Persona		



Prueba de Unidad		
Nombre Prueba: Eliminar		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el nomenclador que desea eliminar y acceder a la opción "Eliminar NomencladorAP", luego se elimina el nomenclador del sistema.		
Entrada: Atributo atributo		
Criterio de Aceptación: Eliminar Nomenclador Atributo de Persona		



Prueba de Unidad		
Nombre Prueba: ObtenerAtributoByID		
Estado: Satisfactoria	**Tipo:** Caja Blanca	**Ultima Ejecución:**15/052013
Ejecutado por: Yaciel Mendoza	**Verificado por:** Ronaldo Castro Milán	
Descripción: Para la realización de esta prueba se debe seleccionar el nomenclador que desea detallar y acceder a la opción "Mostrar NomencladoAP", luego se le muestra todos los detalles del nomenclador.		
Entrada: Guid idAtributo		
Criterio de Aceptación: Mostrar Nomenclador Atributo de Persona