Universidad de las Ciencias Informáticas
Facultad 2

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

Módulo de Administración de Seguridad en el Sistema ABCD 3.0

Autores: Lisetvis Rodríguez Quintana

Yusbiel Hermelo Alfonso

Tutor: Ing. Yadier Mesa Pérez

Co-Tutor: Ing. Yaksel Duran Rivas

“La Habana, 2015”

DECLARACIÓN DE AUTORÍA

Declaramos que Lisetvis Rodríguez Quintana y Yusbiel Hermelo Alfonso somos los únicos autores de este trabajo y autorizamos al Centro de Informatización de Gestión Documental (CIGED) de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmo la presente a los ____ días del mes de ____Junio____ del año ____2015____.

_____	_____
Firma del Autor	Firma del Autor
Lisetvis Rodríguez Quintana	Yusbiel Hermelo Alfonso
_____	_____
Firma del Tutor	Firma del Co-Tutor
Ing. Yadier Mesa Pérez	Ing. Yaksel Duran Rivas

Agradecimientos

Lisetvis Rodríguez Quintana

A mis mamá por todos los sacrificios que ha hecho para convertir este sueño realidad, por su amor, por su entrega, por no dudar nunca de mí, por el apoyo infinito que siempre me brindó, por ser siempre la luz que me guio en todos estos años.

A mi novio Yulierkys, por ayudarme y darme su apoyo en todo este tiempo que hemos estado juntos, por su comprensión, por estar siempre presente a mi lado guiándome, por su amor y por hacer hasta lo imposible por ver realizado nuestros sueños.

A mi papá, a mi segunda mamá, Santica que me adora y me apoyó siempre. A hermano que siempre tuvo un consejo que darme en los momentos tristes y de desesperación, a mi hermana por su cariño. A mis dos sobrinas por el cariño, el amor y aguantar mis peleas, por hacerme reír cada vez que estamos las tres juntas. A mi sobrino ahijado que de todos es el que más regaño recibe por darme su amor.

A mi tutor por todo su empeño y preocupación, por dedicarle parte de su tiempo a la realización de esta investigación, por su constante tenacidad para que hagamos las cosas bien y tengamos buenos resultados.

A mi compañero de tesis por su paciencia, por la ayuda que me ha brindado. A mis compañeros de grupo, principalmente a Dayana,, Ernesto, Bilmarys, Jordan por su amistad. A las nuevas amistades encontradas, Daniel, Yazmin por ser tan pacientes conmigo y tan atentos.

A Osmel, gracias por todo, por hacerme reír, por compartir conmigo en todos los momentos, buenos o malos, por tu apoyo, gracias.

Agradecimientos

Yusbiel Hermelo Alfonso.

A mi mamá por ser la principal causa de esta parte de mi vida, a mis amistades del grupo, Yoel, Jeffrey, por sus consejos. A mi compañera de tesis por su comprensión. A mi tutor por todas las atenciones que mostró durante el desarrollo de este proyecto, A todas las personas involucradas, gracias.

RESUMEN

Los Sistemas de Gestión Bibliotecaria son utilizados para informatizar los procesos que se generan en una biblioteca. El sistema de Automatización de Bibliotecas y Centros de Documentación en su versión 1.2 desarrollado en la Universidad de Ciencias Informáticas, carece de una adecuada arquitectura de software y no implementa políticas de seguridad que garanticen la protección del mismo. En la presente investigación se realizó un análisis de las funcionalidades que fueron desarrolladas y asociadas al sub-módulo de administración de la seguridad en la nueva versión 3.0 del sistema de Automatización de Bibliotecas y Centros de Documentación. Para su desarrollo se utilizaron las herramientas Visual Paradigm 8.0 para el modelado, PostgreSQL para la gestión de los datos en la base de datos del sistema y Virgo como servidor de aplicaciones. Como resultado del desarrollo del sub-módulo de seguridad se tiene un mejor control sobre las operaciones que realizan los usuarios. Se maneja toda la información de los usuarios que acceden al sistema contribuyendo a una mejor práctica de los servicios prestados.

PALABRAS CLAVE

Control, información, protección, seguridad

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El rápido crecimiento de las Tecnologías de la Información y las Comunicaciones (TIC) y las mejoras de seguridad, organización y gestión que ellas traen consigo promueven su uso y necesidad de aplicación en diferentes áreas de la vida del hombre. Las TIC posibilitan el fácil acceso a la información, pueden procesar de forma rápida e íntegra los datos, ostentan una amplia capacidad de almacenamiento y automatización de trabajos, así como gran interactividad con los usuarios que la utilizan.

Como parte de las acciones del gobierno revolucionario para lograr fomentar el uso de las TIC en función de alcanzar un mayor desarrollo de la economía nacional y la sociedad; se crea en el año 2002 la Universidad de las Ciencias Informáticas (UCI) que tiene como misión formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática, producir aplicaciones y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la informática.

En la UCI se encuentra el Centro de Informatización de la Gestión Documental (CIGED), el cual posee varios proyectos encaminados a la informatización de los procesos inherentes a la gestión de la documentación. Uno de estos proyectos es el de Automatización de Bibliotecas y Centros de Documentación (ABCD) que se encarga de desarrollar un sistema web para informatizar los procesos relacionados con las distintas áreas que puedan existir en una biblioteca.

Debido a que los sistemas basados en la Web están siendo cada vez más accedidos a través de las redes por usuarios y sistemas, estos son susceptibles a exponer información de carácter confidencial. Por ello es necesario tener en cuenta la seguridad de la información y la protección de los datos, aspectos fundamentales para lograr la integridad, confidencialidad y disponibilidad de la información.

Para evitar violaciones que puedan afectar el correcto funcionamiento y estado de los sistemas se deben utilizar mecanismos de seguridad que permitan solo el acceso de las personas autorizadas y controlar las actividades que estas realizan, garantizando además la autenticidad y el no repudio. Por tanto es de vital importancia tener en cuenta la Seguridad Informática, la cual según fuentes electrónicas es la disciplina que se relaciona a diversas técnicas, aplicaciones y dispositivos encargados de asegurar la integridad y privacidad de la información para el procesamiento de datos en sistemas informáticos (1).

Producto del trabajo desarrollado en el proyecto ABCD se obtuvo una versión de un sistema destinado a informatizar los procesos que se realizan en las bibliotecas cubanas. Dicho sistema se

denominó ABCD 1.2 y puede decirse que sentó las bases para desarrollar nuevas versiones destinadas a perfeccionar los procesos informatizados por este.

En el sistema ABCD 1.2, no existía una arquitectura de software definida, por lo que los módulos fueron implementados independientemente por los desarrolladores, los que utilizaron arquitecturas distintas según las facilidades que estas le ofrecían para la implementación. Esto presentó como consecuencia que no se lograra una buena integración de los módulos implementados, además que no se tuviera una estructura adecuada del sistema. De esta manera se vio afectado el cumplimiento de los términos del proyecto para con el cliente.

Por otro lado, las políticas de seguridad implementadas no satisfacen la protección del sistema. El mecanismo de autenticación implementado no cuenta con la robustez necesaria para que solo logren el acceso a la aplicación los usuarios autorizados, trayendo consigo que cualquier persona que intente ingresar al sistema lo logre.

Otro de los puntos débiles de la seguridad implementada en el ABCD 1.2 es la protección de los datos y la información. Esta versión del sistema no garantiza su seguridad, pues se centra más en el flujo de información que en su protección en sí. Mientras exista un volumen de información lo suficientemente extenso compartido entre los usuarios, la seguridad pasa a un estado secundario. Como consecuencia de esto, una persona con acceso a los datos del sistema puede realizar cambios sobre estos, viéndose comprometida, la integridad, disponibilidad y confidencialidad de la información, pilares fundamentales para lograr la seguridad de cualquier sistema informático.

Por otra parte, no implementa un buen algoritmo de autenticación que controle el acceso a funcionalidades del sistema, así como a la información sensible que este maneja. No tiene control sobre la gestión de permisos según los roles de los trabajadores, debido a que no se manejan por las responsabilidades que debe tener cada trabajador sino que prima el objetivo de que estén activos los servicios, ya sea responsabilidad del trabajador en cuestión o no.

Actualmente se está implementando una nueva versión del sistema ABCD, para resolver los problemas encontrados en ABCD 1.2. En desarrollo, ABCD 3.0 está estructurado por diferentes módulos, los cuales responden a las necesidades que existen en la biblioteca y al propio proceso de gestión. Uno de estos módulos es el de Administración que, entre otras funciones, tiene a su cargo la seguridad del sistema.

Habiendo expuesto lo anterior se plantea como **problema a resolver**: ¿cómo contribuir con el apoyo de las Tecnologías de la Información y las Comunicaciones a garantizar la seguridad en el Sistema ABCD 3.0?

Se define para ello como **objetivo general**: desarrollar las funcionalidades asociadas a la administración de la seguridad en el Sistema ABCD 3.0. Obteniéndose, como **objeto de estudio**, según el problema identificado anteriormente: la seguridad de la información en los sistemas informáticos; quedando enmarcado el **campo de acción** en: las funcionalidades asociadas a la administración de la seguridad en el Sistema ABCD 3.0.

Para dar solución al objetivo general se trazaron las siguientes **tareas de investigación**:

1. Análisis de los principales conceptos asociados los procesos relacionados con la administración de la seguridad para obtener la base teórica necesaria en el desarrollo de la solución.

2. Análisis de los diferentes modelos de implementación de los sistemas de seguridad existentes para la elaboración del Sub-Módulo Seguridad.

3. Estudio de las técnicas de validación de los sistemas de gestión bibliotecaria para la validación del Módulo de Seguridad.

Para alcanzar los resultados esperados en esta investigación se aplicaron **métodos de investigación científica**. Estos métodos proporcionan una serie de pasos sistemáticos y constituyen instrumentos que conllevan a un conocimiento científico.

Del marco teórico, ha sido seleccionado el método Analítico-Sintético. Con el análisis se divide el fenómeno en sus múltiples relaciones y componentes para facilitar su estudio. En el proceso de síntesis se establece mentalmente la unión entre las partes previamente analizadas y de esta forma se descubren sus características generales y las relaciones esenciales entre ellas. Este método es utilizado en el transcurso de la investigación en las distintas fuentes citadas.

Dentro de los métodos empíricos se ha seleccionado como técnica de recopilación de información la entrevista. Con la confección previa de un guión de preguntas, enmarcadas en el problema objeto de estudio, se logró una mayor comprensión de las necesidades y objetivos del sistema a desarrollar.

Estructuración del trabajo de diploma

El presente documento consta de tres capítulos, estructurados de la siguiente manera:

Capítulo 1: Fundamentos teóricos del desarrollo del componente de seguridad, donde se realiza un estudio preliminar de sistemas y funcionalidades que puedan dar respuesta al problema planteado. Igualmente muestra las tecnologías, metodologías y herramientas que fueron utilizadas

en el desarrollo de la solución propuesta.

Capítulo 2: Características de la solución propuesta, contiene el flujo actual de los procesos, el modelo de dominio, la propuesta de solución, los requisitos de software y el modelo de caso de uso, así como los diagramas de clases del sistema ABCD 3.0.

Capítulo 3: Diseño y desarrollo de la solución propuesta, se centra en la modelación detallada y la construcción de la estructura del componente. Además se implementan las clases y subsistemas en términos de componentes de la solución propuesta.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

Introducción

Como parte de las exigencias de control y seguridad de los nuevos sistemas de gestión de la información, se hace necesaria la creación de estructuras capaces de permitir el manejo y control de acceso de los usuarios a estos. Debido a la importancia que tiene la seguridad de los datos y la información de los sistemas de gestión documental es fundamental concebir un mecanismo que garantice que se desarrolle un sistema informático seguro.

El aprovechamiento de las nuevas características tecnológicas que ha traído consigo el desarrollo de las TIC, agiliza el procesamiento de los datos en los sistemas de información garantizando rapidez en los servicios que brindan. En función de describir las técnicas, metodologías y herramientas utilizadas para implementar un sistema de información seguro a continuación se desarrolla el capítulo 1.

1.1 Conceptos fundamentales.

1.1.1 Sistemas Integrados de Gestión Bibliotecaria.

Los Sistemas Integrados de Gestión Bibliotecaria (SIGB) son herramientas para automatizar los procesos inherentes a una biblioteca (2). Un SIGB, integra en un solo programa informático un conjunto de aplicaciones específicas que se denominan módulos, pensados para facilitar las tareas específicas de este, las cuales están directamente relacionadas unas con otras. Toda la información reunida en esta, se almacena en una misma base de datos que permite el mejor intercambio de la información y el aprovechamiento de los recursos con el menor esfuerzo posible (3).

Debido al volumen de información con que trabajan estos sistemas y al acceso de un gran número de usuarios que interactúan con sus servicios, los SIGB requieren de una estructura de seguridad que garantice la integridad, confidencialidad y disponibilidad de la información, además del control de las operaciones que se realizan en él.

1.1.2 Seguridad Informática.

La seguridad informática se refiere a las características y condiciones en sistemas de procesamiento de datos y su almacenamiento para garantizar confidencialidad, integridad y disponibilidad (4). Se le dice seguridad informática tanto a la investigación como a la ejecución de políticas de protección de datos en ordenadores por parte de un individuo o equipo de expertos en

computación (5). Define además aquellas prácticas que se llevan adelante a fin de proteger y resguardar el funcionamiento y la información contenida en un sistema de cómputo.

La seguridad está finamente ligada a la certeza. Para entender esto, hay que aclarar que no existe seguridad absoluta, más bien, lo que se intenta es minimizar el impacto y/o riesgo. Existen tres pilares fundamentales que hacen que la información se encuentre protegida. Estos pilares se ocupan principalmente de proteger tres aspectos de la información:

- ***Confidencialidad***: la información puede ser accedida únicamente por las personas que tienen autorización para hacerlo.

- ***Disponibilidad***: este término hace referencia al método de precaución contra posibles daños tanto en la información como en el acceso a la misma: ataques, accidentes o simplemente descuidos pueden ser los factores que obligan a diseñar métodos para posibles bloqueos.

- ***Integridad***: garantiza estar totalmente seguros de que la información no ha sido borrada, copiada o alterada, no sólo en su trayecto, sino también desde su origen.

Para lograr que estos tres pilares funcionen correctamente hay que asegurarse de que el mecanismo para proteger la información sea funcional y corresponda a satisfacer las necesidades del sistema, de modo que no se vea afectado el correcto manejo de los datos que se procesan en el mismo.

1.1.3 Mecanismo de seguridad.

Para poder crear un sistema informático, hay que llevar a cabo un análisis profundo y detallado de cuáles podrían ser las principales amenazas que pudiera sufrir dicho sistema. A partir de este análisis, habrá que diseñar una política de seguridad capaz de integrarse a las estrategias del negocio, a su misión y visión, estableciendo un grupo de responsabilidades y reglas a seguir para evitar esas amenazas o minimizar los efectos si se llegan a producir.

Una política de Seguridad es el conjunto de normas y procedimientos establecidos por una organización que mantengan los niveles de seguridad estables, para regular el uso de la información con el fin de mitigar el riesgo de pérdida, deterioro o acceso no autorizado a la misma ante alguna amenaza y en caso de que ocurra tratar de reponerse (6). Para esto se implementan un grupo de mecanismos para la seguridad del sistema, que garanticen la protección del mismo.

Mecanismo de seguridad

Los mecanismos de seguridad son también llamados herramientas de seguridad y son todos aquellos que permiten la protección de los bienes y servicios informáticos (7). Dentro de sus funciones se encuentran el indicar la manera en que se deben ejecutar las acciones que permitan resguardar la seguridad y se eviten vulnerabilidades en la misma. Un mecanismo de seguridad es aquel que está diseñado para detectar, prevenir o recobrarse de un ataque de seguridad. Los mecanismos de seguridad implementan varios servicios básicos de seguridad o combinaciones de estos.

Estos tipos de mecanismos pueden ser clasificados en tres grupos (8):

- Mecanismos de prevención.
- Mecanismos de detección.
- Mecanismos de recuperación.

Mecanismos de prevención

Los mecanismos de prevención son aquellos que aumentan la seguridad de un sistema durante su funcionamiento. Los mecanismos de prevención más habituales en redes son los siguientes:

- Mecanismos de autenticación e identificación: permite identificar la entrada al sistema de una determinada entidad y una vez identificada realiza el proceso de autenticación, de esta forma se garantiza que el usuario que quiere entrar a la aplicación es quien dice ser (9).
- Mecanismos de control de acceso: permite tener un control sobre los servicios u objetos que ofrece el sistema (10).
- Mecanismos de separación: cualquier sistema con diferentes niveles de seguridad ha de implementar mecanismos que permitan separar los objetos dentro de cada nivel, evitando el flujo de información entre objetos y entidades de diferentes niveles siempre que no exista una autorización expresa del mecanismo de control de acceso. Los mecanismos de separación se dividen en cinco grandes grupos, en función de cómo separan a los objetos: separación física, temporal, lógica, criptográfica y fragmentación (11).
- Mecanismos de seguridad en las comunicación: son aquellos que en un sistema protegen la integridad y privacidad de los datos cuando se transmiten a través de la red, la mayoría de estos mecanismos se basan en la criptografía que no es más que un conjunto de proto-

colos y sistemas que se utilizan para proteger la información y dotar de seguridad a las comunicaciones y a las entidades que se comunican (12).

Mecanismos de detección

Son aquellos que se utilizan para detectar violaciones de la seguridad o intentos de violación; ejemplos de estos mecanismos son los programas de auditoría como Tripwire3 (13).

Mecanismos de recuperación

Son aquellos que se aplican cuando ha ocurrido una violación en el sistema y poder retornar al mismo a un funcionamiento correcto (14).

¿Qué tipos de mecanismos utilizar?

De estos mecanismos, los sistemas de información se centran principalmente en los mecanismos de prevención, dado a que se espera garantizar un comportamiento seguro desde el inicio de los procesos, en cuyo caso, la detección y la recuperación constituyen procesos de segundo grado.

Como los SIGB son sistemas en los que el flujo de información y el control y el acceso son importantes mantener sin afectaciones, se deben utilizar mecanismos de prevención que proporcionen el cumplimiento de estos requisitos. Es por ello que para satisfacer sus exigencias se deben implementar un mecanismo de control de acceso y un mecanismo de autenticación, para facilitar una mayor seguridad sobre los objetos del sistema.

La Autenticación.

En redes de equipos públicos y privados el mecanismo de autenticación se lleva a cabo comúnmente a través de contraseñas de inicio de sesión. Todos los clientes deben ser autenticados cuando se conecten a la aplicación, después de eso el cliente es de confianza y a su vez cuando un cliente ejecuta alguna acción, estas son chequeadas contra una política específica, donde cada uno de los controles y acciones son permitidos o denegados para ese usuario. Esta especificidad asume que el usuario esta autenticado tanto tiempo como dure la conexión.

El Control del Acceso.

Los mecanismos de control de acceso se clasifican en tres grupos:

Control de Acceso Discreto (DAC por sus siglas en inglés).

En este mecanismo el creador o propietario del recurso, decide cómo protegerlo estableciendo cómo compartirlo, mediante controles de acceso impuestos por el sistema. Lo esencial es que

el propietario del recurso puede cederlo a un tercero. Esto refleja que en DAC es más importante el flujo de información que su seguridad en sí (15).

Control de Acceso Obligatorio (MAC por sus siglas en inglés)

En este modelo es el sistema quién protege los recursos, comparando las etiquetas del sujeto que accede frente al recurso accedido, o sea, la autorización para que un sujeto acceda a un objeto depende de los niveles de seguridad que tengan, ya que estos indican que permiso de seguridad tiene el sujeto y el nivel de sensibilidad del objeto. Esto viene dado por anillos de confianza en los que solo el administrador del sistema puede otorgar permisos a los usuarios (16).

Control de Acceso Basado en Roles (RBAC por sus siglas en inglés)

En este modelo a los usuarios le son asignados uno o varios roles mientras que los permisos y privilegios se asignan a estos roles. Por tanto, las políticas de control de accesos basado en roles regulan el acceso de los usuarios a la información en términos de sus actividades y funciones de trabajo (roles), representándose así de forma natural la estructura de las organizaciones (17). En una organización, un rol puede ser definido como una función que describe la autoridad y responsabilidad dada a un usuario en un instante determinado. Incluye un conjunto de sesiones donde cada sesión es la relación entre un usuario y un subconjunto de roles que son activados en el momento de establecer dicha sesión. Cada sesión está asociada con un único usuario. Mientras que un usuario puede tener una o más sesiones asociadas. Los permisos disponibles para un usuario son el conjunto de permisos asignados a los roles que están activados en todas las sesiones del usuario, sin tener en cuenta las sesiones establecidas por otros usuarios en el sistema. La arquitectura de RBAC se puede representar como se muestra en la figura 1.

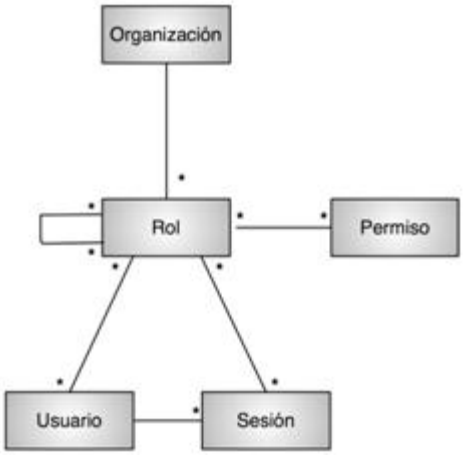


Figura 1: Estructura del mecanismo RBAC

¿Por qué RBAC?

RBAC es reconocido como uno de los modelos que más han evolucionado en los últimos años, ya que fundamentalmente puede llegar a funcionar con grandes sistemas, y al mismo tiempo proporcionarle seguridad. Básicamente, la gran evolución que se está teniendo con RBAC es porque funciona como una mezcla de DAC y MAC, pues contiene de cierta forma la flexibilidad para el Control de Accesos que tiene DAC y la rigidez de MAC (18). Además que dada la alta integración entre los roles y las responsabilidades de los usuarios, se pueden seguir los principios del mínimo privilegio y de la separación de responsabilidades.

Principios a tener en cuenta al implementar RBAC:

- Un usuario tiene acceso a los objetos de acuerdo al rol que tenga asignado: este principio establece que un usuario puede acceder a diversos objetos del sistema según la responsabilidad que posee, por tanto no tendrá la posibilidad de acceder a funcionalidades con las que no está vinculado.

- Los roles son definidos en base a funciones de trabajo: plantea que los roles que desempeñan los usuarios del sistema están estrechamente orientados a las funciones que realiza cada persona, definiendo varios roles a la misma en dependencia de su trabajo.

- Los permisos son definidos en función de la autoridad y responsabilidad que se asume en una función de trabajo: para definir los permisos RBAC le asigna esto a los roles,

pues ellos son los encargados de llevar consigo el peso de cada labor que realizan los usuarios en el sistema.

- Las operaciones sobre un objeto son invocadas de acuerdo a los permisos: esto asegura que para realizar cualquier actividad sobre un objeto se tengan en cuenta los permisos del usuario que está efectuando la operación, ya que solo aquellos que ostentan esa condición serán capaces de efectuar cambios en dichos procedimientos.

1.2 Algoritmos de cifrado.

Muchos son los ataques que puede recibir un software. Por ello, implementar un mecanismo de seguridad que cumpla con los requisitos necesarios puede ser de gran ayuda para obtener el efecto que provee un sistema informático seguro. Lograr que la información que se maneje en los sistemas viaje de forma segura propicia el cumplimiento de los tres pilares fundamentales de la seguridad informática.

Desde tiempos antiguos el hombre ha buscado la manera de enviar mensaje de forma tal que sólo el receptor sea capaz de entender lo que se le ha enviado. Con el pasar del tiempo esta técnica se ha ido perfeccionando y en nuestros días con el uso de las TIC e internet como principal fuente de comunicación en el mundo, se ha hecho necesario desarrollar algoritmos que encripten la información que exponen los usuarios. Algunos de estos algoritmos son:

MD5 (Message Digest Algorithm 5 o Algoritmo de Firma de Mensajes 5): Desarrollado por Ron Rivest, ha sido hasta los últimos años el algoritmo hash más usado. Procesa mensajes de una longitud arbitraria en bloques de 512 bits generando un compendio de 128 bits. Debido a la capacidad de procesamiento actual esos 128 bits son insuficientes, además de que una serie de ataques cripto-analíticos han puesto de manifiesto algunas vulnerabilidades del algoritmo. Puede ser útil para comprobar la integridad de un fichero tras una descarga, por ejemplo, pero ya no es aceptable desde el punto de vista del criptoanálisis (19).

Blowfish: uno de los algoritmos de encriptación en bloque más poderoso, desarrollado por el criptógrafo Bruce Schneider. El tamaño del bloque es de 64 bits; tamaño de la clave hasta 448 bits (20).

GOST: algoritmo soviético creado por el KGB (Komitet Gosudárstvennoj Bezopásnosti o Comité para la Seguridad del Estado) a finales de los 70. Funciona con bloques de 64-bits. Longitud de clave - hasta 256 bits. A pesar de los varios agujeros de seguridad que se encontraron, aún se

considera bastante fiable. Es el estándar de encriptación de la Federación Rusa (20).

Rijndael: algoritmo, desarrollado por Joan Daemen y Vincent Rijmen. Cumple con los estándares de AES (Estándar de Encriptación Avanzado). Utiliza claves de diferentes tamaños (128, 192 y 255 bits) y bloques del mismo tamaño (20).

RC4: algoritmo de encriptación usado en muchos sistemas de seguridad de redes (por ejemplo en el protocolo SSL usado en Netscape y la encriptación de contraseña de Windows NT). Las mayores ventajas de este código son su alta velocidad y tamaño de clave ajustable. Este algoritmo fue desarrollado por RSA por Ronald Rivest. RC significa "Ron´s Code" (Código de Ron) o "Rivest Cipher". Fue propiedad intelectual de RSA hasta 1995 (20).

Con el análisis de los algoritmos de cifrado se llegó a la conclusión de que no son recomendables para la encriptación de la información en el sistema ABCD 3.0. Aunque cada uno ellos en su momento logró satisfacer las necesidades existentes en los software, las fallas que presentan y por las que han sido quebrantados demuestran que no son confiables.

1.2 Sistemas existentes.

Para la gestión de la información generada en las bibliotecas se han desarrollado diferentes aplicaciones que han satisfecho las necesidades que presentan los servicios que estas brindan. Muchos de estos sistemas han utilizado características similares con el fin de obtener mayor colaboración de los usuarios que la utilizan. Mayormente los SIGB tienen la característica de ser sistemas de código libre y/o abierto, pues les proporcionan a los usuarios libertad de uso y el código fuente de estos para poder realizar cambios si se desea.

Actualmente existen en el mundo varias aplicaciones informáticas destinadas a la gestión bibliotecaria, todas ellas con la misión de perfeccionar sus servicios pero no exponen la forma en la que manejan la seguridad de la información de los usuarios que interactúan con el sistema ni la del propio software en sí. Por este motivo la presente investigación centra su estudio en otras aplicaciones informáticas.

Sistema Quarxo Fase 2: utiliza el framework spring security para proporcionar servicios de seguridad con un mecanismo declarativo o independiente del sistema donde se despliegue. El núcleo de la seguridad del componente Onyx se centra en el desarrollo de las funcionalidades brindadas por el framework Spring Security, el mismo provee mecanismos de autenticación y autorización basados en el uso de una serie de filtros de seguridad que configurados debidamente elevan la seguridad de la aplicación. El mecanismo de seguridad delega responsabilidades en un

conjunto de filtros, que aseguran entre todos las funcionalidades del sistema. El componente de seguridad Onyx hace uso de 14 filtros de seguridad, 4 de ellos fueron especialmente creados para cubrir las necesidades propias del sistema y asegurar partes especificas del mismo a las que el framework no era capaz de llegar, estas necesidades van a ser expuestas desde el punto de vista de su solución más adelante, además se adaptó el comportamiento de otros 3 filtros para asegurar la compatibilidad con el sistema y se dejó con el funcionamiento estándar a los restantes 7 filtros (21).

Desarrollo de funcionalidades para fortalecer la seguridad del Módulo Admisión aplicando los perfiles de seguridad IHE: utiliza mecanismos de autenticación y autorización para ello utiliza los perfiles de seguridad EUA y ATNA que proponen crear certificados auto-firmados para avalar que un sistema o nodo sea quien dice ser y se sustituyan los protocolos HTTP y TCP/IP por los HTTPS y SSL respectivamente, para garantizar la confidencialidad de los datos mediante su encriptación. El segundo grupo engloba los elementos asociados al desarrollo de funcionalidades que solucionen las problemáticas de confidencialidad de la información clínica. Dichas funcionalidades estarán orientadas a los datos guardados en el gestor de base de datos, y en los documentos clínicos almacenados en el servidor de aplicaciones. Para solucionar esta situación se encripta y desencripta la información mediante el algoritmo simétrico AES con una llave de 256 bits, para lograr una mayor resistencia a ataques informáticos. De esta forma se asegura que los datos y ficheros se guarden de forma ilegible. Además se crea un servidor VSFTPD para almacenar todas las HCE generadas en el módulo Visor HC como recomienda el perfil de seguridad XDS. Admisión aplicando los perfiles de seguridad IHE (22).

Módulo de administración para la Plataforma de Identificación de la UCI: utiliza el *framework* .Net en su versión 4.0 centrando su seguridad en los mecanismos de autenticación (a través de contraseñas) para ello La primera vez que un usuario perteneciente al sistema se autentique, deberá usar su contraseña del dominio LDAP de la UCI, pues ha sido registrado al sistema con dicha contraseña. Una vez autenticado se le brinda la opción de editar su perfil, cambiando su contraseña por otra que le permita entrar por el dominio local. Luego de esto, ya no podrá acceder más por la contraseña del dominio LDAP. Para el acceso al sistema utiliza el mecanismo de autorización RBACK, asignando los permisos del sistema a los roles definidos para el sistema, utiliza el mecanismo de seguridad de auditorias el cual es Para chequear el correcto comportamiento del sistema, el mismo mostrará un registro de las operaciones realizadas en el sistema a través de un filtrado por fecha, usuario y acción que se desee verifica (23).

Analizando la seguridad de los sistemas descritos anteriormente, se puede llegar a la conclusión de que los mecanismos de seguridad para la autenticación y el control de acceso, satisfacen las necesidades de seguridad de estos sistemas. Aunque cada uno de ellos tiene características y propósitos diferentes esto solo resulta en el aporte de un matiz distinto para la implementación del mecanismo escogido por ellos. Como están basados en otras arquitecturas y utilizan, en algunos casos, herramientas que difieren de las escogidas por el equipo desarrollo del sistema ABCD 3.0 no se recomienda la implementación de los mecanismos de seguridad utilizados según la forma propuesta por estos sistemas.

1.3 Metodología y Entorno de desarrollo

Durante el proceso de desarrollo de software cobran vital importancia las herramientas que se deben utilizar y la forma en que se realiza todo el proceso. Por ello es imprescindible realizar una buena selección de los métodos y útiles que serán utilizados durante la elaboración del sistema informático.

La ingeniería del software es la disciplina que se encarga del análisis, diseño, implementación y despliegue de sistemas informáticos. Para que un software tenga un buen desarrollo debe tener concebida una metodología capaz de solucionar los problemas que pueden ser generados en el proceso de desarrollo del mismo. Al enfrentar cualquier tarea, se debe tener una base conceptual sólida que permita la visión objetiva del problema a solucionar y las vías adecuadas que permitan lograrlo. Para el desarrollo del sistema ABCD 3.0 se parte de una metodología y herramientas previamente definidas.

1.3.1 Metodología y herramientas

Las herramientas utilizadas por el proyecto ABCD para darle solución al software que se desea desarrollar son: como lenguaje de modelado, UML (*Unified Modelling Language*) en su versión 2.1. La herramienta CASE es Visual Paradigm 8.0 para el modelado de clases. Se utilizará además PostgresSQL 9.4 como gestor de bases de datos y Eclipselink para la persistencia de los datos. El desarrollo de la aplicación se hará en la plataforma de Eclipse en su versión 4.3 como IDE (*Integrate Development Environment*) de desarrollo. El sistema será implementado en el lenguaje orientado a objeto, Java, el cual es uno de los lenguajes más utilizados en aplicaciones web. Para el diseño de las interfaces se hará uso de RAP 3.0 (*Remote Aplication Protocol*). OSGI (*Open Service Gateway Initiative*) es el *framework* de desarrollo determinado que implementa los servicios a través de *bundles,* el cual es un componente muy parecido a un .jar tradicional de Java,

con sus interfaces, e implementaciones. Para acceder al marco modular y configuración dinámica de OSGI se utiliza SpringDM, quien además facilita la implementación. El servidor de aplicaciones es Virgo en su versión 3.6.3, pues tiene una gran interacción con el marco de desarrollo definido. Todo esto se desarrollará a través de la metodología del Proceso Unificado de *Rational* (RUP), el cual establece una guía para la arquitectura que es utilizada en el diseño y prueba del sistema.

Conclusiones del capítulo

En este capítulo se hizo un análisis del estudio de los sistemas existentes relacionados con la seguridad de los sistemas existentes en el mundo y en la UCI. Se describen los mecanismos de seguridad existentes y se justifica por qué la utilización del mecanismo escogido para el sistema ABCD 3.0, además se especifica la utilización de las herramientas y tecnología escogida por el proyecto en el sub-módulo de seguridad.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DE LA SOLUCIÓN

Introducción

En respuesta a la problemática expuesta es necesario realizar un análisis de las características que debe tener la solución que se quiere brindar. Para ello es esencial la generación de los artefactos que deben describir, según la metodología utilizada, las fases de análisis y diseño del sub-módulo de seguridad del sistema a construir.

Propuesta de solución.

El desarrollo de un sistema seguro es de primordial importancia para las empresas desarrolladoras de software. La construcción de una herramienta segura proporciona por parte de las entidades involucradas, confiabilidad, mayor satisfacción con el producto desarrollado y aumenta la credibilidad del cliente respecto al equipo de desarrollo.

La implementación del sistema ABCD 3.0 se hará sobre una nueva arquitectura definida por el proyecto, lo que conlleva a un estudio exhaustivo de las herramientas, tecnologías y metodologías a utilizar. El uso de OSGI como *framework* de trabajo propone un entorno de desarrollo de colaboración. Los *bundles*, que son los componentes de OSGI para los desarrolladores, permiten la comunicación a través de servicios bien definidos y ocultan el código interno de otros *bundles*, lo que proporciona más libertad a la hora de realizar cambios, reduciéndose de esta manera la complejidad y facilitando el uso de otros componentes desarrollados por terceros, de modo que sea reutilizable. OSGI presenta otras características como son:

- Servicios: da soporte intrínsecamente a una arquitectura orientada a servicios (SOA). Los paquetes publican servicios en el registro de servicios y otros paquetes pueden descubrir estos servicios a partir del registro de servicios (24).
- Ciclo de Vida: el API de instalar, iniciar, detener, actualizar y desinstalar paquetes.
- Módulos: es la capa que define los conjuntos. Puede importar y exportar código.
- Seguridad: la capa que se encarga de los aspectos de seguridad.
- Entorno de ejecución: define qué métodos y clases están disponibles en una plataforma específica.

Con la implementación de la versión 3.0 del sistema ABCD que se desarrolla, el módulo de

administración de dicho sistema estará compuesto por diferentes sub-módulos, dentro de los cuales se encuentra el sub-módulo de Seguridad. Dicho sub-módulo tiene como objetivo desarrollar nuevas políticas de seguridad, lograr mayor eficiencia en la identificación y autenticación de las personas que acceden al sistema ABCD 3.0 para evitar problemas que puedan afectar el correcto funcionamiento del mismo, así como una buena implementación de los requisitos de seguridad utilizados.

Además de implementar nuevos mecanismos de prevención, también se desea utilizar un mecanismo de encriptación para las contraseñas de los usuarios que accedan al sistema. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema, para las que se exige una adecuada complejidad en su formato y se complementa su protección con un algoritmo de encriptación computacionalmente complejo.

La seguridad del sistema ABCD 3.0 utiliza el algoritmo Sha-1 para la encriptación de las contraseñas, el cual produce una salida resumen de 160 bits (20 bytes) de un mensaje que puede tener un tamaño máximo de 2^{64} bits. Este ha sido examinado muy de cerca por la comunidad criptográfica pública, y no se ha encontrado ningún ataque efectivo (25).

Para lograr los objetivos necesarios que permitan que el sistema ABCD 3.0 cumpla con las políticas de seguridad establecidas, se pretende alcanzar con la utilización de RBAC como mecanismo de control de acceso la seguridad y control a nivel de usuario, permitiendo el acceso de los mismos solo a las funcionalidades establecidas según la función que desempeñan.

Se mantendrá otro nivel de seguridad para las sesiones de trabajo, garantizando solo la ejecución de las aplicaciones que hayan sido definidas para la sesión en cuestión. A las sesiones estarán asociados los roles, los cuales tendrán consigo los permisos asociados a cada usuario según el rol que desempeñan en cada sesión activa a la que tengan acceso.

Para la conexión al servidor del sistema desde el navegador se utilizará el protocolo HTTPS, el cual como está basado en SSL/TLS funciona en la capa más baja de comunicación, cifrando todos los datos de HTTP. Por lo que no protege solo la página web sino también la URL completa, los parámetros enviados, las cookies, etc. Lo único que quedará al descubierto son los datos del paquete TCP: el servidor y el puerto al que se hace la conexión.

Al realizar una petición al servidor, este envía información al cliente web que la está solicitando. En la versión 3.0 de ABCD se utiliza el protocolo SSL (capa de conexión segura) para garantizar que la información viaje encriptada y de forma segura propiciando que disminuya el riesgo de que

terceros usen indebidamente la información.

Es también de suma importancia garantizar que la información almacenada en el servidor sea consistente y se utilizarán validaciones que limiten la entrada de datos irreales y mecanismos de vuelta atrás en procesos críticos que terminen abruptamente y produzcan estados inconsistentes de la información. La información deberá estar disponible a los usuarios en todo momento, limitada solamente por las restricciones que estos tengan de acuerdo a la función que realicen en el sistema.

2.1 Modelo de dominio.

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Representa las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema (26). Por otro lado se extraen las reglas del negocio, las cuales constituyen requisitos de cómo el negocio puede operar. Estas pueden enmarcarse en categorías, por ejemplo, reglas de restricciones, reglas de estímulo y respuesta, reglas de restricción de funcionamiento, etc.

La utilización del modelo de dominio o modelo conceptual brinda una representación gráfica y un mejor entendimiento de cómo interactúan los conceptos fundamentales que intervienen en los procesos a informatizar. Este describe entidades o conceptos del mundo real que están asociados al problema en cuestión. Dicho modelo se utiliza como una base de las abstracciones relevantes en el proceso de construcción de la solución propuesta.

2.1.1 Conceptos del modelo del dominio.

Para un mejor entendimiento del dominio se deben describir y extraer los conceptos relacionados con las entidades, objetos, personas, detectadas en el estudio del sistema que se está desarrollando. A continuación se hace una descripción de cada uno de los conceptos que intervienen en la seguridad de los procesos que se realizan en el sistema.

Tabla 2.1: Conceptos del Modelo de Dominio

Conceptos	Descripción
Actor	El actor es aquel usuario o sistema.
Actor Tangible	Es un tipo específico de actor que está más relacionado con los usuarios.
Persona	Son todos aquellos que pueden que pueden estar vinculados al sistema independientemente del rol que este ejerza.
Account	Se refiere a las cuentas que posee una biblioteca.

Perfil	Son todos aquellos roles que existen en una biblioteca y que se asocian a una cuenta de usuario respondiendo a las responsabilidades del mismo.
Permisos	Los permisos son asociados a los perfiles que presenta una cuenta de usuario en la biblioteca, de igual manera un permisos puede estar asignado a varios perfiles.
Address	Es una lista de direcciones a las que accede el actor en el sistema.
Biblioteca	Las bibliotecas son aquellas que cuentan con una serie de cuentas asociados a los usuarios, además de los perfiles que pueden presentar los mismos.
Usuario Préstamo	Es el usuario que recibe algún servicio externo de la biblioteca como son los pedidos o prestamos de ejemplares.

2.1.2 Diagrama de clases del modelo del dominio

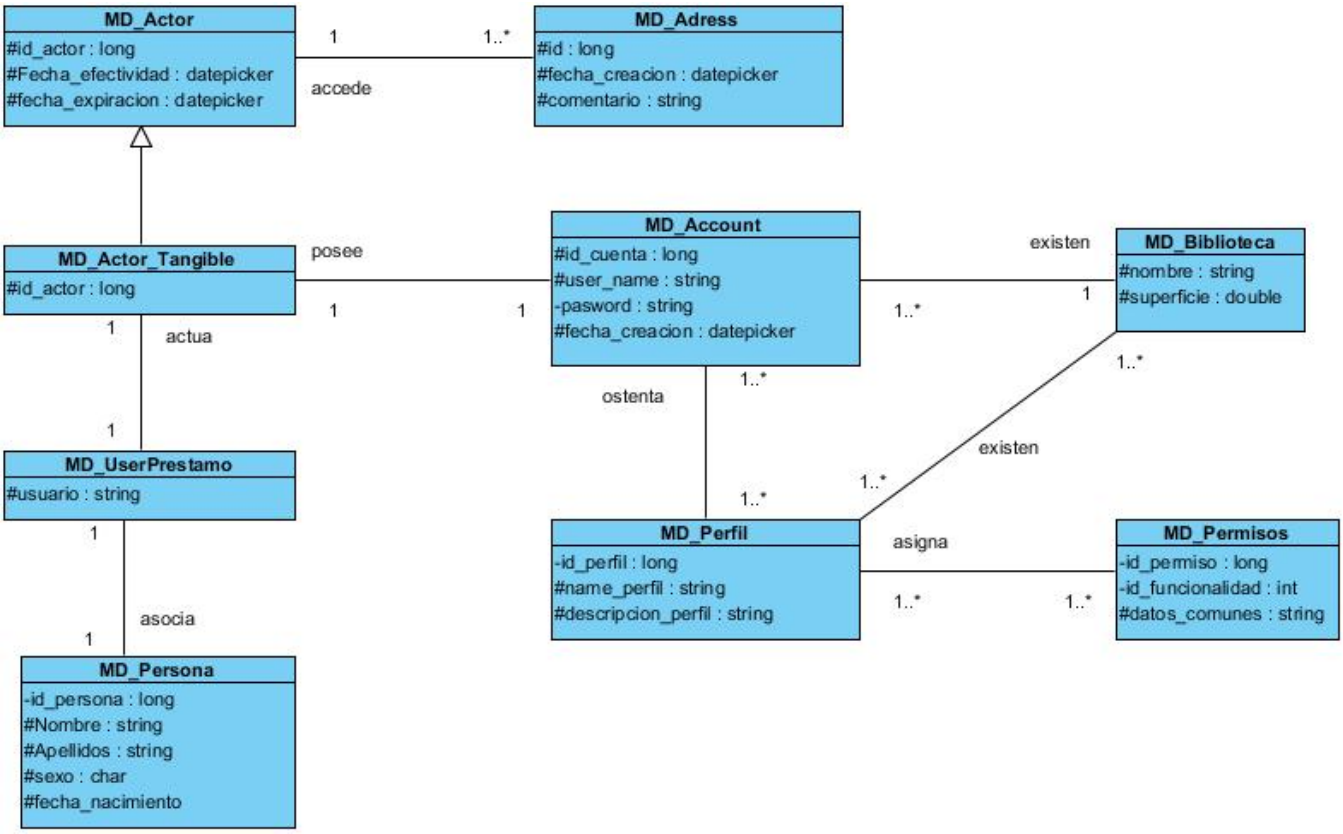


Figura 2.1: Diagrama de clases del Modelo de Dominio.

2.2 Modelos de caso de uso.

Un modelo de caso de uso (CU) es una descripción, en un lenguaje bien definido, de una secuencia de acciones a fin de detallar la interacción que existe entre un actor y las

funcionalidades del sistema que se está construyendo. Ellos contienen en si toda la información relacionada con los requisitos vinculados al software que se quiere desarrollar (27).

Para la creación de casos de usos comúnmente se utilizan patrones. Estos patrones describen soluciones a los problemas más comunes de organización y estructura que se pueden presentar al diseñar los casos de uso.

2.2.1 Patrones de casos de uso.

Los patrones de casos de uso nos permiten modelar y especificar los requerimientos de nuestro sistema. Entre los beneficios de su uso se encuentran que facilitan la planeación de los proyectos y ayudan a llegar a acuerdos con los clientes. Para que los casos de uso sean más claros y sostenibles es importante encontrar patrones y documentarlos. De esta manera, cuando nos encontremos con un problema igual o parecido, se puede resolver en menor tiempo.

El concepto de patrones no es algo que solo es aplicable a la práctica de requerimientos. De hecho, la disciplina de requerimientos copia este concepto de la de análisis y diseño. Lo que se busca con los patrones es reutilizar lo aprendido en los nuevos proyectos y usarlos en la organización como estándares (28).

Para la obtención y diseño de los casos de uso del sistema de seguridad se utilizaron los siguientes patrones de CU:

- **CRUD**: este patrón se utiliza en los casos donde se quiere realizar altas, bajas, cambios y consultas a alguna entidad del sistema. Su nombre es un acrónimo de las palabras en inglés Create, Read, Update, Delete.

- **Extensión**: consiste en dos casos de uso y una relación extendida entre ellos. Este patrón se aplica cuando el flujo de un caso de uso puede extender al flujo de otro, así como ser realizado en sí mismo.

- **Inclusión**: Se incluye una relación del caso de uso base al caso de uso de inclusión. El último puede ser instalado en sí mismo. El caso de uso base puede ser concreto o abstracto.

2.2.2 Descripción de los CU

Tabla 2.2: Descripción de Casos de Uso

Nombre del caso de Uso	Descripción

Gestionar persona	Permite describir los procesos de Registrar, Editar, visualizar y eliminar datos de una persona
Consultar Persona	Describe el proceso de realizar una búsqueda de una persona en el sistema y muestra la lista de posibles coincidencias.
Gestionar Usuario	Permite describir lo procesos de registrar, editar, Visualizar y Eliminar datos de un usuario.
Consultar Usuario	Describe el proceso de realizar la consulta de un usuario en el sistema y muestra la lista de posibles coincidencias.
Gestionar Perfil de Usuario	Permite describir lo procesos de registrar, editar, Visualizar y Eliminar datos de un perfil usuario.
Consultar Perfil de Usuario	Describe el proceso que se realiza para consultar un perfil de usuario en el sistema y muestra la lista de posibles coincidencias.
Iniciar Sesión	Describe el proceso de iniciar sesión de un usuario del sistema.
Mostrar Escritorio de Trabajo	Describe que acciones realiza el sistema para mostrar el escritorio de trabajo de los usuarios autenticados en el mismo.
Cambiar Contraseña	Describe el proceso de cambiar contraseña, para ello el usuario debe haber validado sus datos en el sistema.
Consultar Sesiones Activas	Describe el proceso que se realiza para consultar las sesiones activas de un usuario en el sistema y muestra la lista de posibles coincidencias.
Gestionar Registro de Acceso	Permite describir lo procesos de registrar, editar, Visualizar y Eliminar datos de un registro de acceso.
Consultar Registro de acceso	Describe el proceso que se hace para consultar un registro de acceso y devuelve una lista con las posibles coincidencias encontradas.

2.3 Diseño de la solución.

En los inicios de la informática la programación se consideraba un arte y se desarrollaba como tal, debido a la dificultad que entrañaba para la mayoría de las personas. Con el tiempo se han ido

descubriendo y desarrollando formas y guías generales que facilitan la solución de muchos de los problemas de estructura y diseño de software. A estas soluciones, se les ha denominado Arquitectura de Software, porque, a semejanza de los planos de un edificio o construcción, estas indican la estructura, funcionamiento e interacción entre las partes del software.

La Arquitectura se refiere a la estructuración de los sistemas que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificación), y servir como guía en el desarrollo (29). Se hace énfasis en “la estructura” del sistema, compuesto de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.

2.3.1 Patrón de arquitectura de la solución propuesta

Cada software, para su desarrollo, amerita una buena arquitectura que le provea al programador la estructura básica de lo que se quiere implementar. En el desarrollo del sistema ABCD el equipo del proyecto optó por utilizar una arquitectura de capas orientada a servicios. La arquitectura n-capas es un estilo más que existe para definir la estructura de los sistemas y es importante en muchos aspectos. Esta permite, en el desarrollo de una aplicación, separar la lógica de Acceso a Datos (Capa de Datos) de la lógica del Negocio (Capa de Negocio) y a su vez de la lógica de Diseño (Capa de Presentación) (30).

El trabajo en el sistema ABCD 3.0 será distribuido en tres capas las que a su vez están especializadas con distintas responsabilidades. Estas capas son: presentación, negocio y acceso a datos. Para una mayor comprensión de la arquitectura a continuación se muestra el diagrama donde se refleja cada una de las capas y sus puntos de interacción.

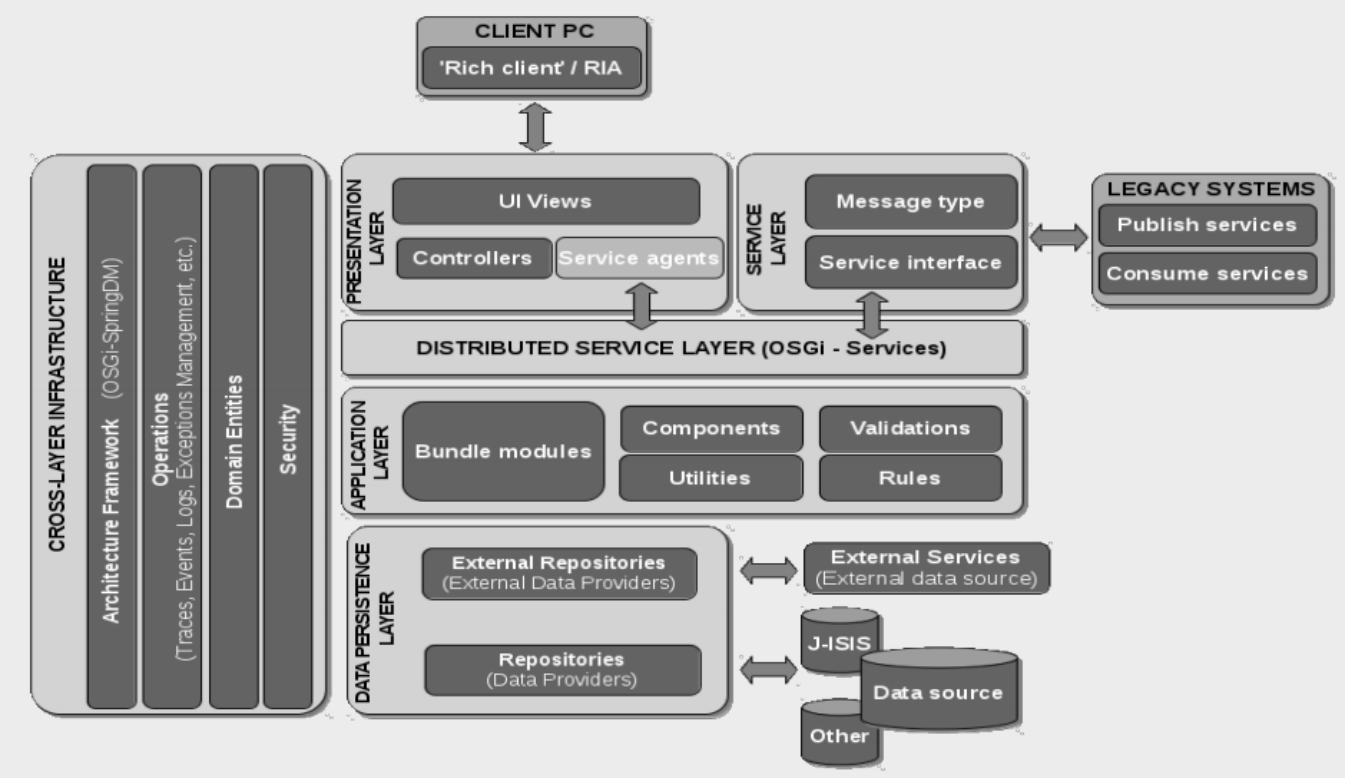


Figura 2.3: Representación de la arquitectura del sistema. **(31)**

Capas de la arquitectura

Presentación: la capa de presentación es aquella con la que interactúan los usuarios que acceden al sistema. Presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de procedimientos. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con las capas de negocio y de seguridad.

Negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos en él. Además brinda seguridad a todas las funcionalidades que lo precisan. También se consideran aquí los programas de aplicación.

Acceso a datos: es donde residen los datos y es la encargada de acceder a los mismos. Con el gestor de bases de datos PostgreSQL realiza todo el almacenamiento de datos, el cual recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Seguridad: es un sub-módulo que se encarga de proteger toda la información que se envía en las capas. Provee de un mecanismo de autenticación para el acceso a la capa de presentación, implementa un mecanismo de control de acceso para una mejor protección en la capa de negocio, así como un algoritmo de autorización para manejar los datos en la capa de datos del sistema.

Por otro lado la orientación a servicios que presenta el sistema está dada a través de un servidor de servicios OSGI. Este es un contenedor de aplicaciones orientado a servicios que proporciona funcionalidades avanzadas para la gestión del ciclo de vida de las aplicaciones desplegadas en el mismo, reflejado en la capa de negocio.

En OSGI las aplicaciones reciben el nombre de *bundle*. Los *bundles* pueden ser cargados, iniciados y detenidos en el entorno de forma dinámica, pudiendo proporcionar uno o más servicios al mismo, de forma que éstos puedan ser utilizados por otras aplicaciones.

2.3.2 Patrones de Diseño utilizados.

Los patrones de diseño expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software (32). Estos muestran las características que deben cumplir las funcionalidades de un sistema de modo que evitan la complejidad y redundancia de los mismos, así como fallas en su funcionamiento y seguridad.

GRASP

Es el sistema orientado a objetos que se compone de objetos que envían mensajes a otros objetos para que lleven a cabo las operaciones requeridas. Los patrones GRASP representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones (33). GRASP es el acrónimo para General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades).

Los patrones GRASP utilizados para el desarrollo del sub-módulo de seguridad son:

Experto: es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con la utilización de este patrón se facilita el encapsulamiento de funcionalidades, ya que los objetos se valen de su propia información para hacer lo que se les pide. Una de las clases en las que se utilizó el patrón experto es en la clase ProfileViewController pues cuenta con los elementos necesarios para el manejo de los perfiles y permisos que estos tienen asociados.

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto

producido en cualquier evento. Brinda un soporte a un bajo acoplamiento, lo que supone menos dependencias respecto al mantenimiento y mejores oportunidades de reutilización. Este patrón es se evidencia en la clase Usuario que crea una instancia de la clase Persona.

Bajo acoplamiento: soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecienten la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades. La clase AccesRecordViewController es un ejemplo donde se aplica el patrón bajo acoplamiento pues esta clase tiene la responsabilidad de controlar los accesos de los usuarios en el sistema y no depende de otras clases por lo que los cambios en ella no le afectan.

Alta cohesión: el patrón Alta Cohesión es la meta principal que ha de tenerse en cuenta en cada momento en todas las decisiones de diseño. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande. Este patrón es importante aplicarlo a cada una de las clases pues lo que se quiere lograr es que las clases contenga solo la información necesaria que les permita realizar la responsabilidad que contiene.

Controlador: asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Este patrón ofrece una guía para tomar decisiones apropiadas que generalmente se aceptan. La misma clase controlador debería utilizarse con todos los eventos sistémicos de un caso de uso, de modo que se pueda conservar la información referente al estado del caso. Este patrón se evidencias en la clase AllManagerViewController.

2.3.3 Diagrama de clases del diseño.

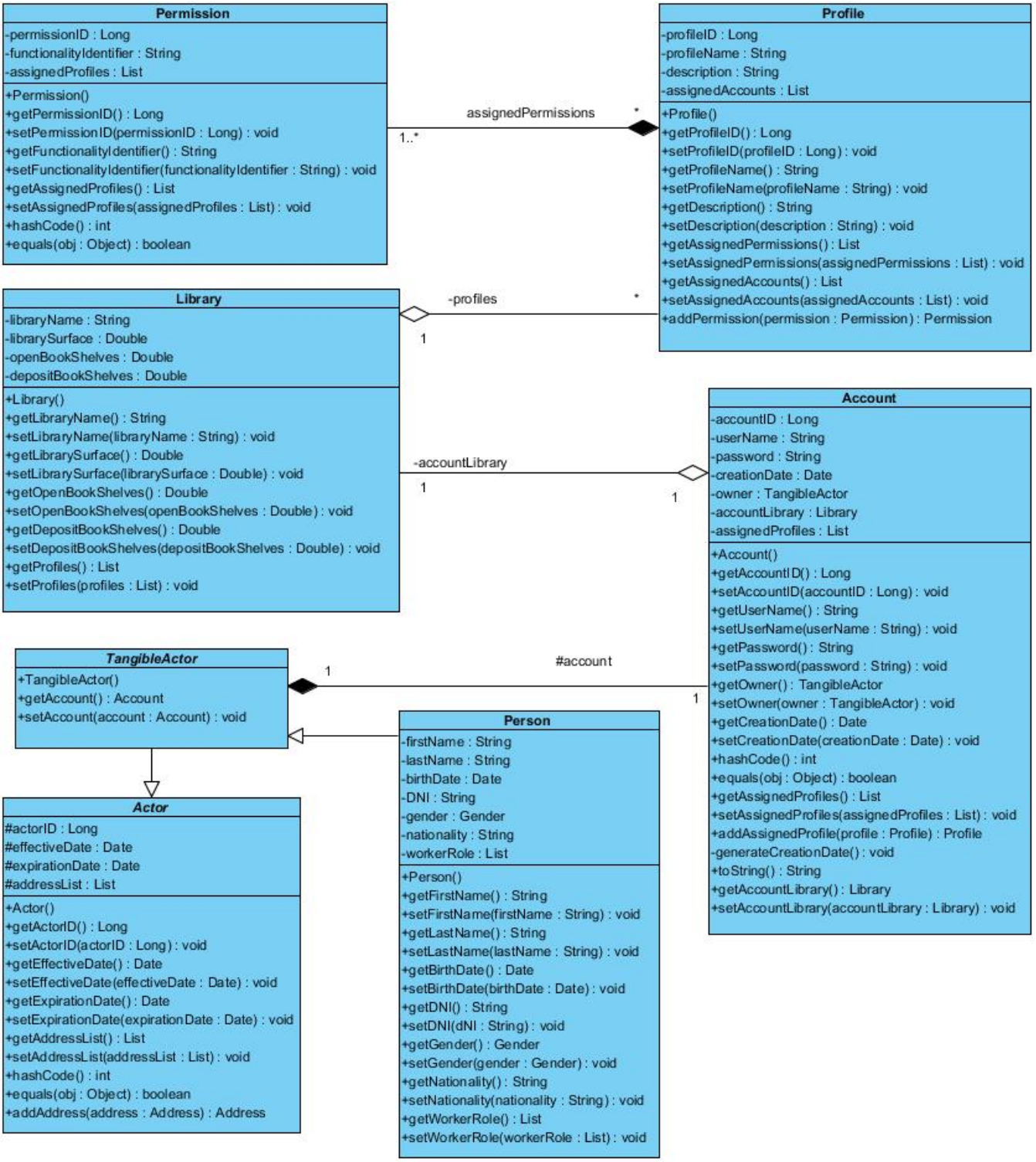


Figura 2.4: Diagrama de Clases del Dominio.

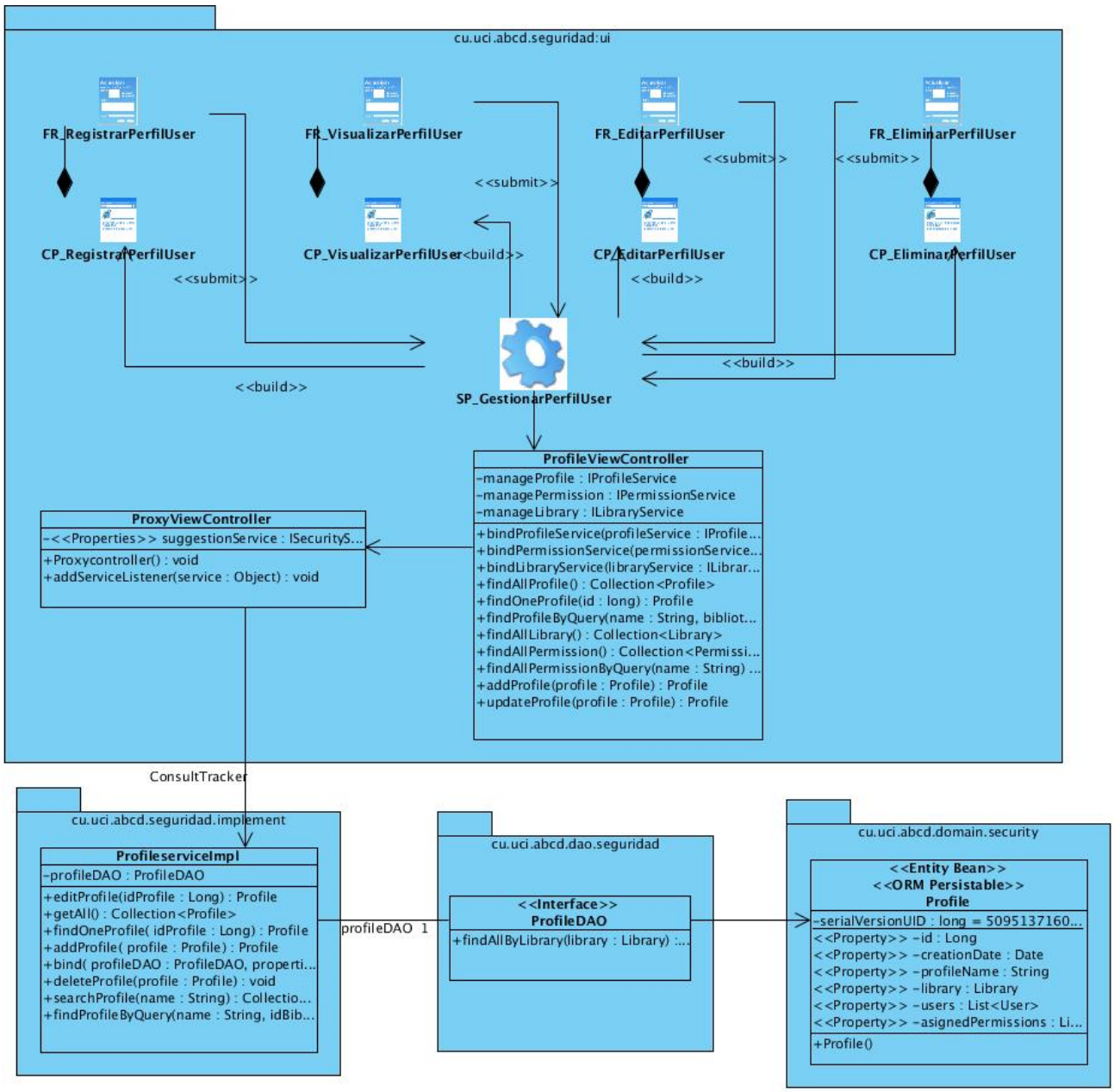


Figura 2.5: Diagrama de Clases del Diseño "*Gestionar Perfil de Usuario".*

2.3.4 Diagramas de Interacción.

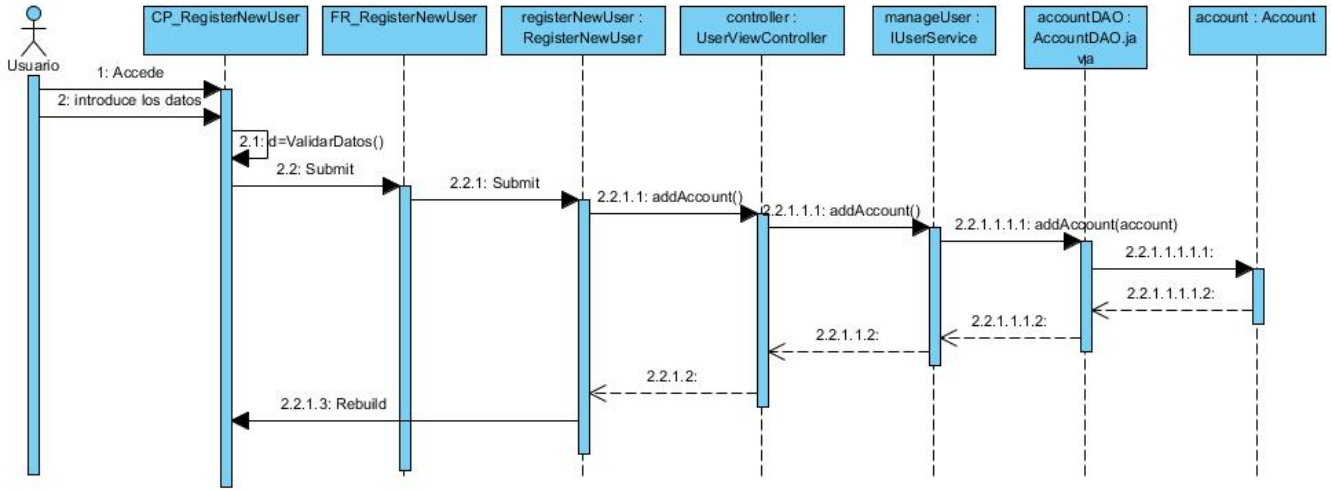


Figura 2.6: Diagrama de Secuencia "*Registrar Usuario*".

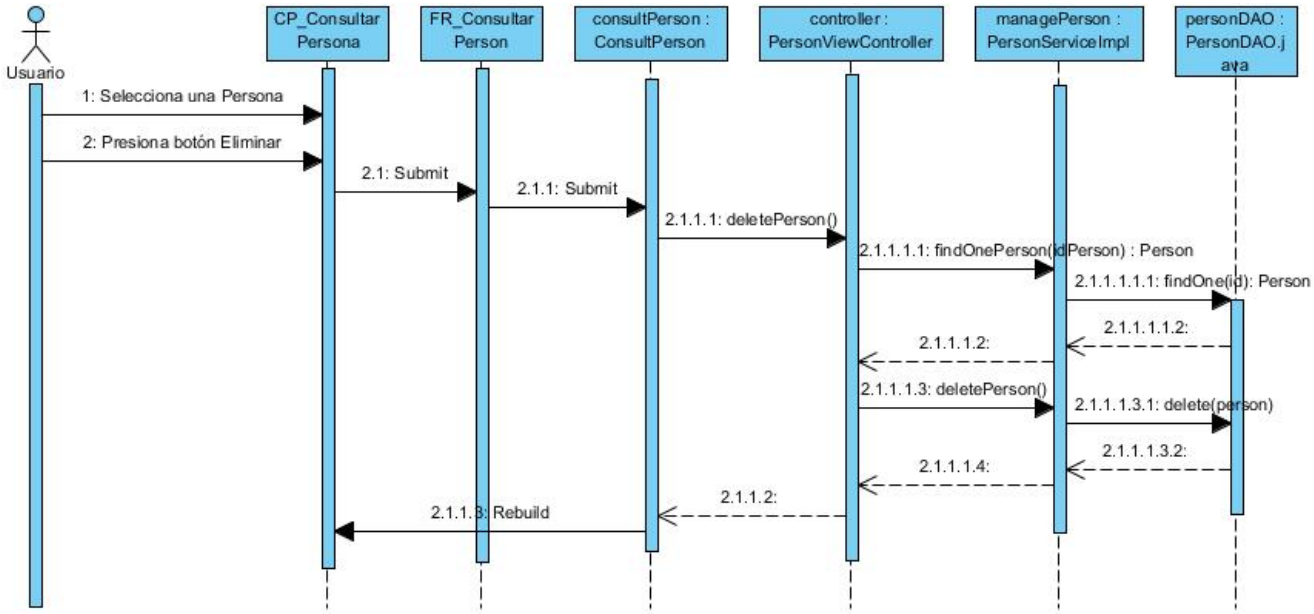


Figura 2.7: Diagrama de Secuencia "*Eliminar Persona*".

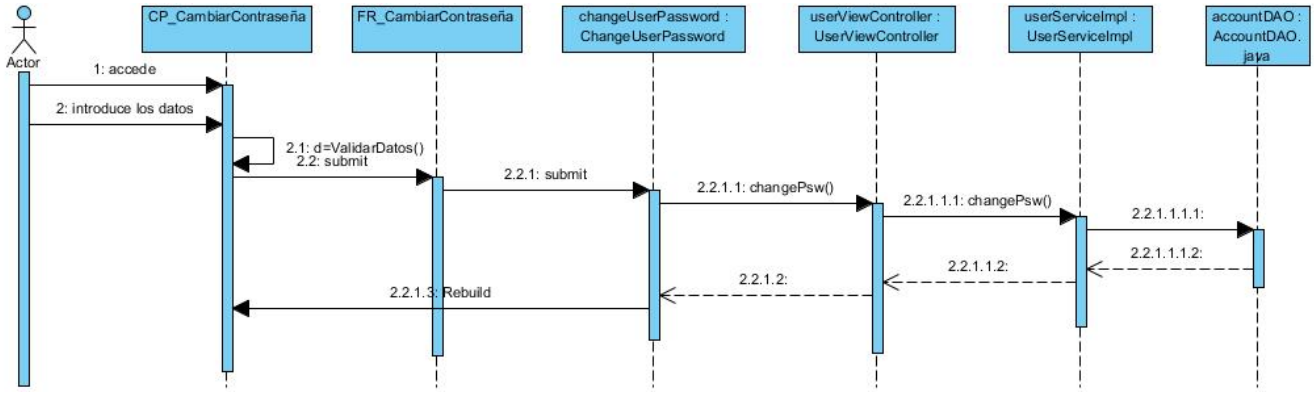


Figura 2.8: Diagrama de Secuencia "*Cambiar Contraseña*".

Modelo de Datos del Sistema

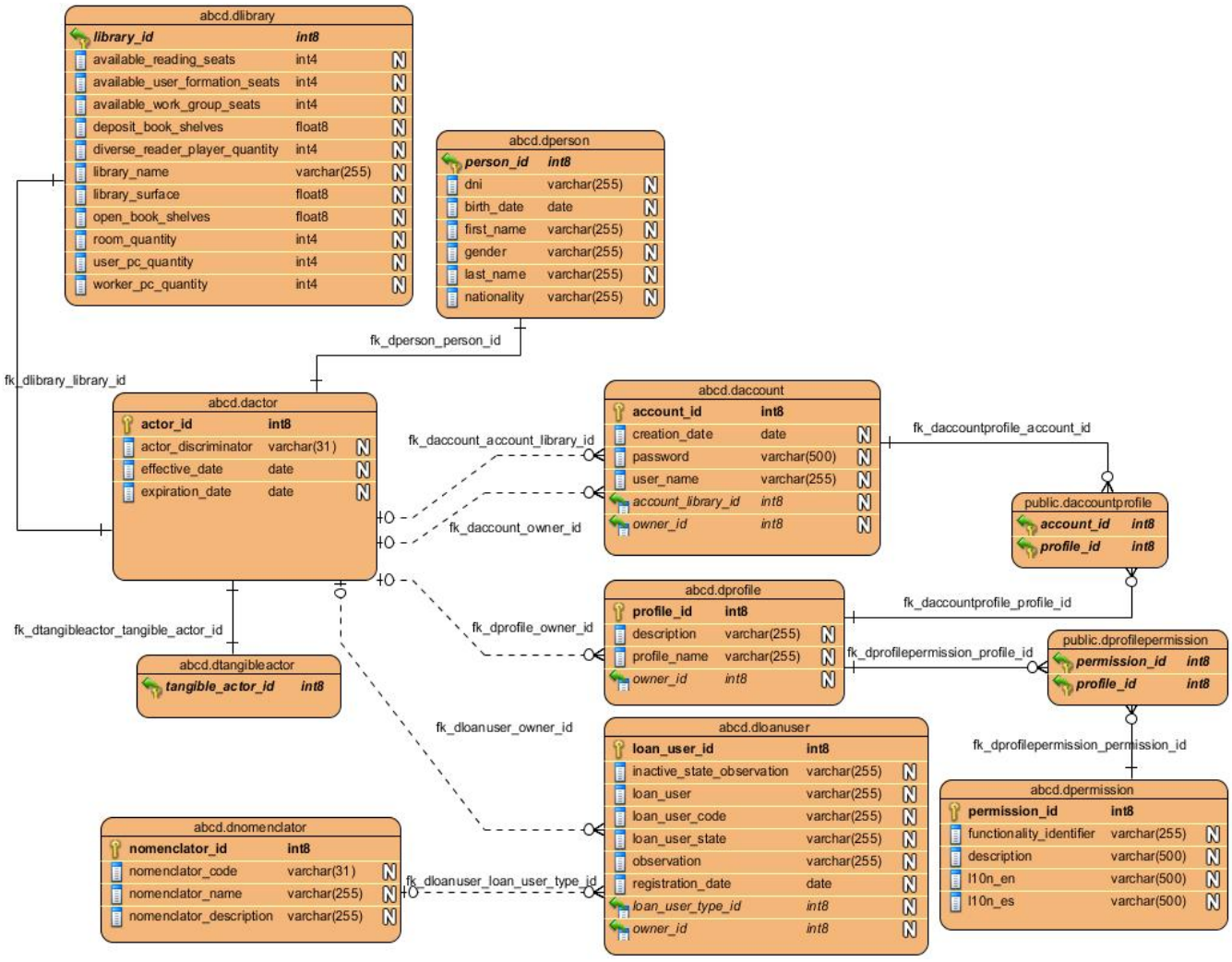


Figura 2.9: Modelo de Datos del sub-sistema de seguridad.

Conclusiones de Capítulo

En el transcurso de este capítulo se brinda una propuesta de solución donde se conciben las características que debe tener el sub-módulo de seguridad para el sistema ABCD 3.0. Además se obtuvo el diseño de las clases que intervienen en las distintas capas de dicha arquitectura, a través de los patrones de diseño escogidos, minimizando el riesgo de implementar incorrectamente la solución propuesta. También se logró el diseño del modelo datos, paso esencial para la construcción de la base de datos del sistema.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE LA SOLUCIÓN

Introducción

En la implementación de los sistemas informáticos es donde se pone en práctica todos los requerimientos que se obtienen del proceso de análisis y diseño del software. Desarrollar las necesidades planteadas para satisfacer lo que las partes involucradas desean, a pesar de tener una base sólida no suele ser algo sencillo. A continuación se describe el proceso de implementación del sub-módulo de seguridad del sistema ABCD 3.0.

3.1 Implementación

Para la implementación del sub-módulo de seguridad del sistema ABCD en su versión 3.0, se definió un conjunto de clases que son las encargadas de contener los elementos necesarios para lograr que el sistema sea seguro. Estas clases están divididas en varios grupos en dependencia de la función que realiza cada una de ellas.

3.1.1 Clases controladoras

Las clases controladoras son aquellos que tienen la responsabilidad de los objetos, pues ellas contienen las acciones que se realizan sobre ellos. Para implementar la seguridad del ABCD 3.0 se utilizaron tres clases controladoras las que se describen a continuación.

PersonViewControllers: esta clase además de tener las acciones básicas a realizar sobre los objetos personas (adicionar, buscar, modificar, eliminar) permite también obtener la cuenta de usuario correspondiente a la persona registrada.

ProfileViewController: esta clase controla las acciones sobre los perfiles de los usuarios del sistema, (adicionar, buscar, modificar, eliminar). Aquí también se manejan los permisos asociados a los perfiles, pues cada perfil tiene una serie de permisos que definen las funciones de los usuarios con determinado rol.

UserViewController: controla todo lo relacionado con los usuarios, como las personas, los perfiles y las cuentas que posee para cada uno de ellos, además de las funciones básicas realizadas sobre el objeto usuario (crear, consultar, eliminar, modificar). Esta clase controladora es muy importante en el sistema por todas las actividades que puede realizar un usuario una vez autenticado. Los usuarios de acuerdo a los perfiles que tienen asociados son los que brindan los

servicios relacionadas con sus funciones.

3.1.2 Clases servicio

Las clases de servicios son aquellas que contienen las diferentes acciones que se hacen sobre los objetos que ellas manejan. Una clase servicio contiene en su código las acciones que se realizan a los objetos además de brindar servicios útiles a la hora de programar ya que pueden contener la información necesaria y los métodos para lograr una buena implementación de los mismos. Los servicios implementados para el sub-módulo de seguridad son los siguientes.

PersonServiceImpl: esta brinda los servicios que se pueden solicitar de las personas. Su objetivo es contener el código de los servicios, visualizar, editar, eliminar, buscar y listar las personas existentes en el sistema.

UserServiceImpl: esta brinda los servicios relacionadas con las cuentas de los usuarios e implementa la interfaz IUserService, que es la interfaz de servicios de usuarios. Además de los servicios de editar, ver, eliminar y adicionar, la clase UserServiceImpl cuenta con un método que permite cambiar la contraseña de una cuenta de usuario y devuelve como resultado si se cambió la contraseña o no.

ProfileServiceImpl: esta clase brinda los servicios relacionadas con los perfiles de los usuarios e implementa la interfaz IProfileService, que es la interfaz de servicios de perfiles. Además de los servicios de editar, ver, eliminar y adicionar, la clase ProfileServiceImpl cuenta con un método que permite obtener los perfiles asociados a una determinada biblioteca y devuelve como resultado una lista de los mismos.

PermissionServiceImpl: esta clase brinda los servicios relacionados con los permisos de los perfiles de usuarios e implementa la interfaz IPermissionService, que es la interfaz de servicios de los permisos. Además de los servicios de editar, ver, eliminar y adicionar, la clase PermissionServiceImpl cuenta con un método que permite obtener los permisos asociados a los perfiles de usuarios, devolviendo como resultado una lista de permisos con sus perfiles.

LibraryServiceImpl: esta clase brinda los servicios que ofrecen las librerías utilizadas para el desarrollo del sistema. Implementa la interfaz ILibraryService y contiene los servicios necesarios para su implementación.

3.1.3 Clases de acceso a datos

Capítulo 3: Implementación y Prueba de la Solución

Las clases de acceso datos son aquellas que contienen la información que se maneja en la base de datos de los sistemas. Estas son las encargadas de manejar los datos de la base de datos y responder a las consultas realizadas por el usuario al sistema.

Account: es la clase que contiene las propiedades de las cuentas de los usuarios registrados en el sistema.

AccountDAO: es la clase que se encarga de realizar las operaciones consecuentes con la entidad *Account* en la base de datos. Provee de un mecanismo para convertir a objeto cada usuario contenido en la base de datos.

Person: es la clase que contiene los datos de las personas en el sistema.

PersonDAO: es la clase que se encarga de realizar las operaciones consecuentes con la entidad *Person*en la base de datos. Provee de un mecanismo para convertir a objeto cada persona contenido en la base de datos.

Profile: es la clase que contiene los datos de los perfiles de usuario en la base de datos del sistema.

ProfileDAO: es la clase que se encarga de realizar las operaciones consecuentes con la entidad *Profile* en la base de datos. Provee de un mecanismo para convertir a objeto cada perfil contenido en la base de datos.

Permission: es la clase que contiene los datos de los permisos de usuario en la base de datos del sistema.

PermissionDAO: es la clase que se encarga de realizar las operaciones consecuentes con la entidad *Permission* en la base de datos. Provee de un mecanismo para convertir a objeto cada permiso contenido en la base de datos.

AccesRegistry: es la clase que contiene los datos de los registros de acceso de los usuarios en la base de datos del sistema.

AccesRegistryDAO: es la clase que se encarga de realizar las operaciones consecuentes con la entidad *AccesRegistry* en la base de datos. Provee de un mecanismo para convertir a objeto cada registro de acceso contenido en la base de datos.

Library: es la clase que contiene los datos de la librería en la base de datos del sistema.

LibraryDAO: es la clase que se encarga de realizar las operaciones consecuentes con la entidad *Library* en la base de datos. Provee de un mecanismo para convertir a objeto la librería contenida en la base de datos.

3.2 Diagramas de componentes

El diagrama de componentes muestra cada uno de los componentes físicos del sistema desarrollado y sus relaciones con otros componentes. Para un mejor entendimiento de la vista de componentes se decidió agruparlos en diferentes paquetes donde cada uno de ellos representa una capa de la arquitectura planteada.

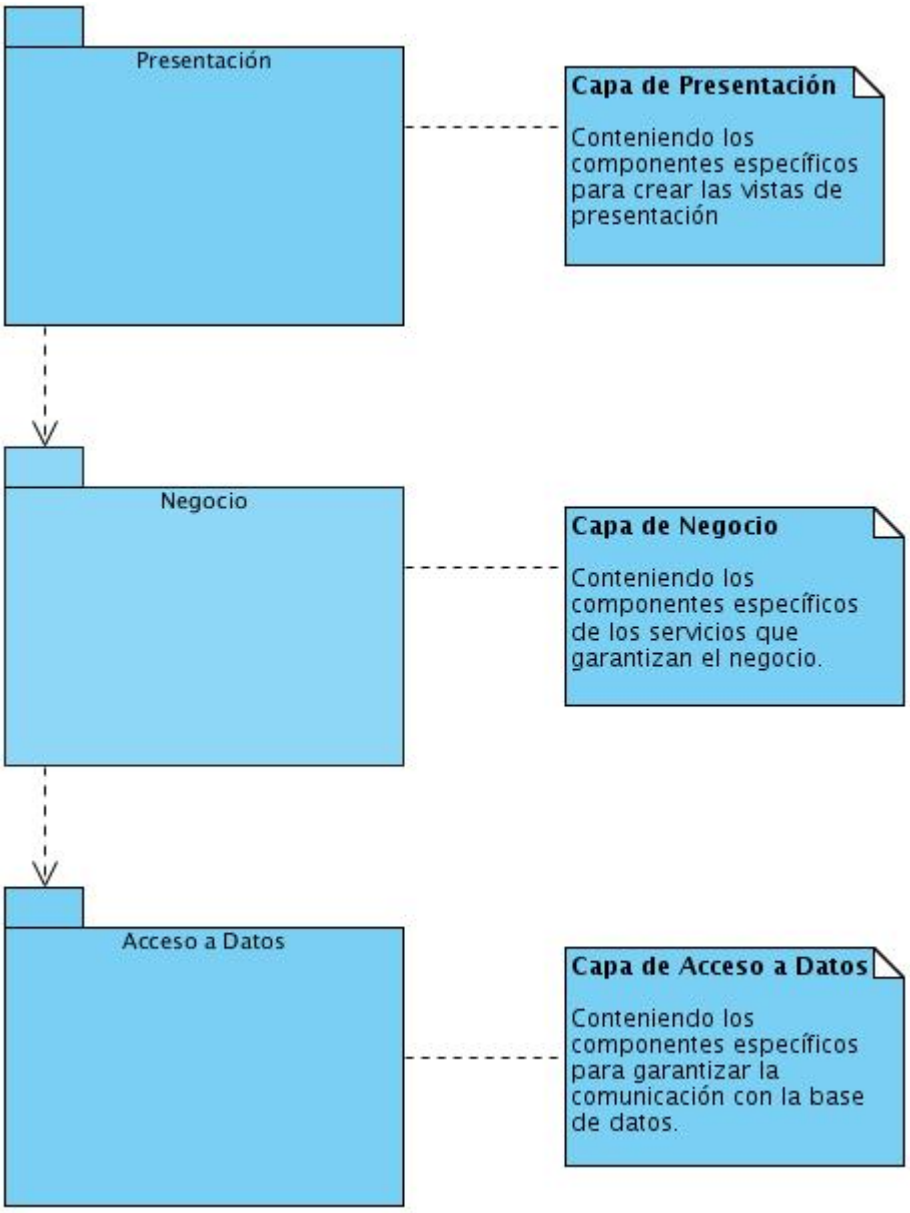


Figura 3.1. Diagrama de paquetes de componentes del sistema**.**

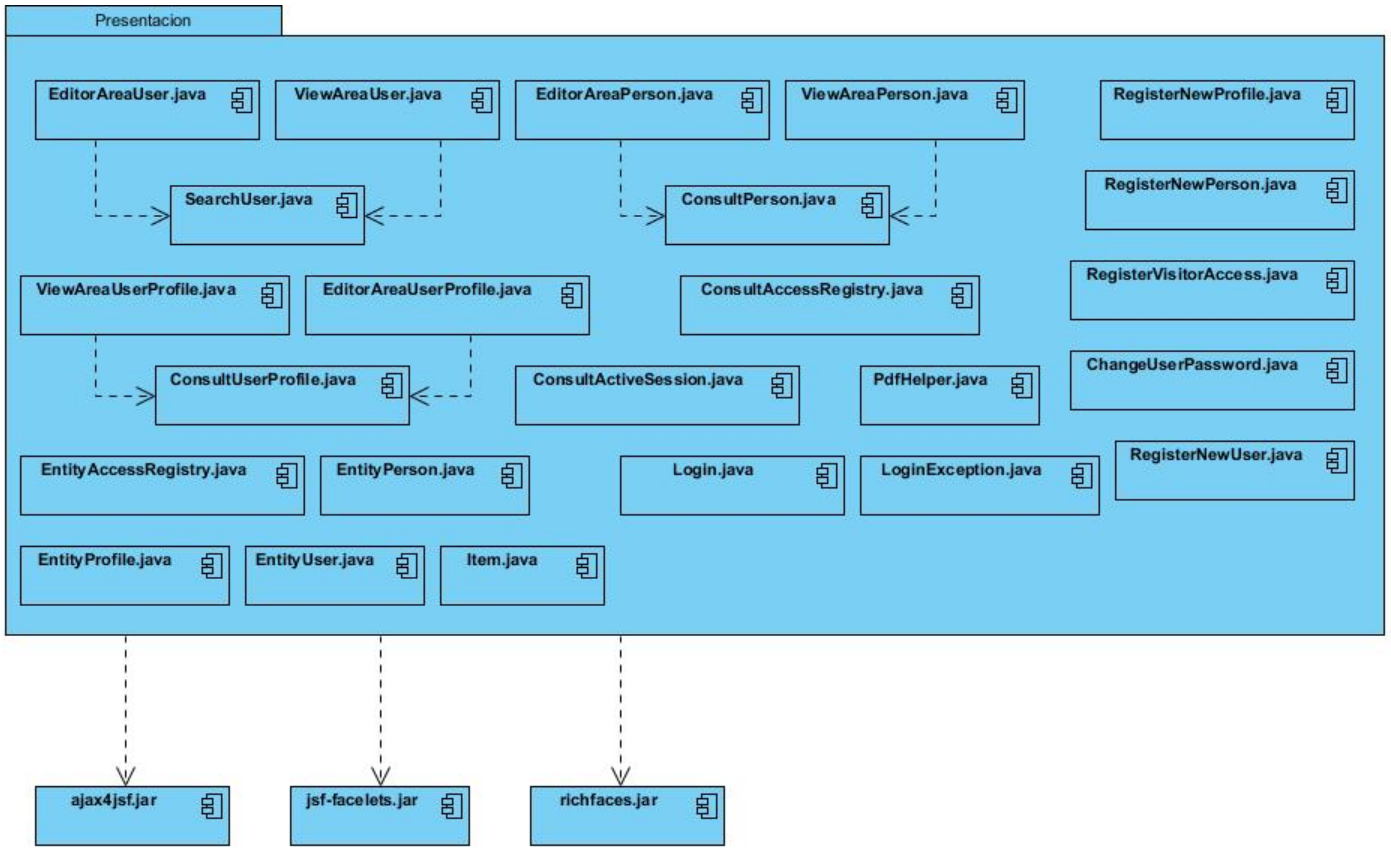


Figura: 3.2. Diagrama de componentes de la Capa de Presentación.

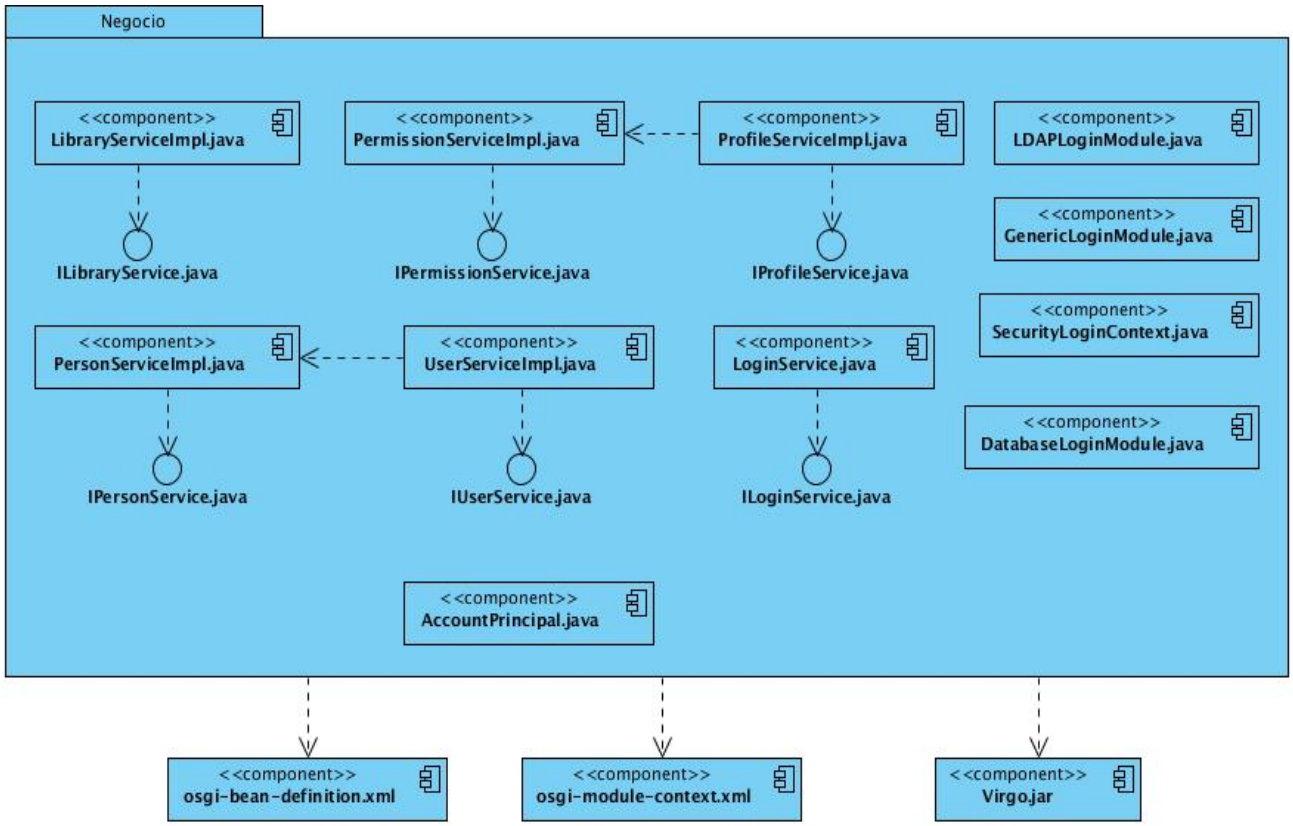


Figura: 3.3. Diagrama de componentes de la Capa de Negocio.

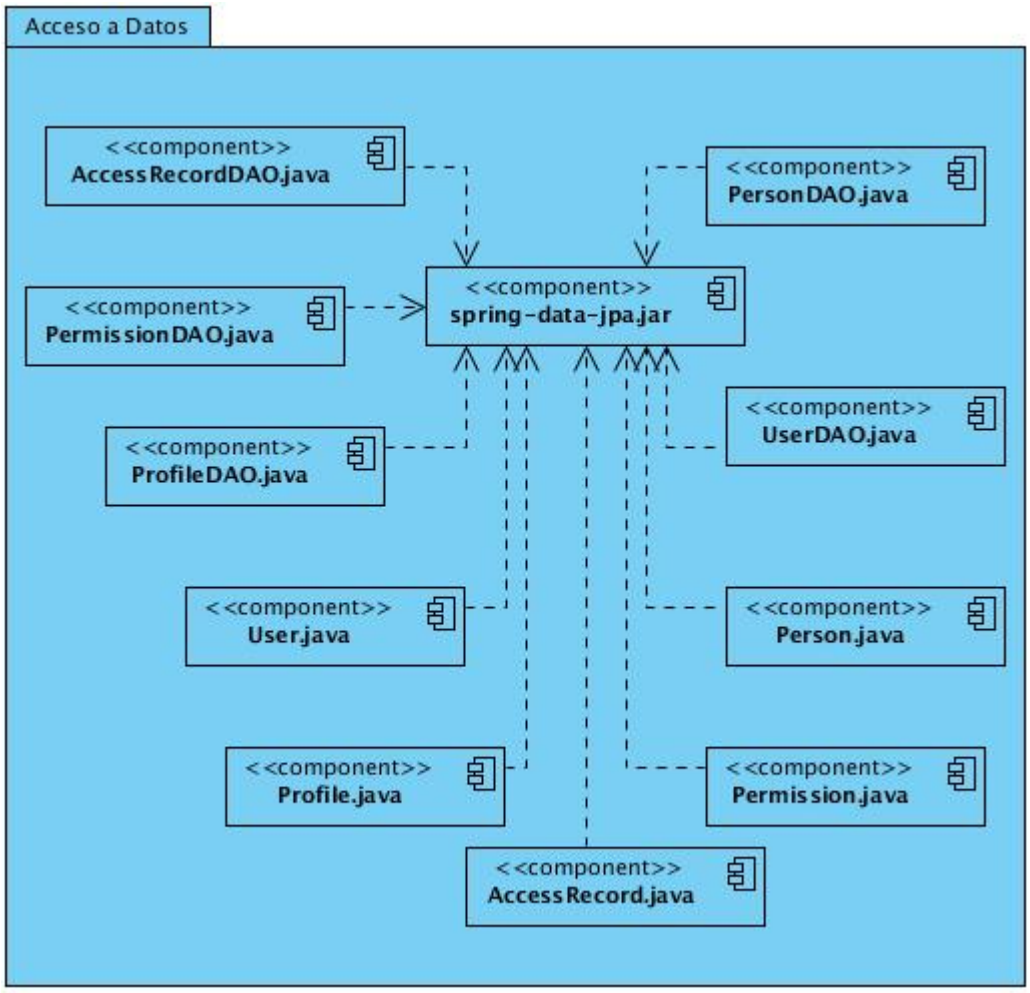


Figura: 3.4. Diagrama de componentes de la Capa de Acceso a Datos.

3.3 Pruebas

Para verificar que un sistema cumpla con las especificaciones expuestas desde el inicio de desarrollo de un software tanto por el analista de un proyecto como por el cliente, se hace necesario realizarle pruebas a los sistemas con el objetivo de cumplir lo más estricto posible con lo establecido desde el principio. Las pruebas de software son las investigaciones empíricas y técnicas cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto (34). Esta actividad forma parte del proceso de control de calidad global. Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software y dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento del proceso de desarrollo.

3.3.1 Tipos de prueba

La correcta selección de cuál tipo de prueba realizar cuando se está desarrollando un software es muy importante, ya que las técnicas y métodos utilizados para garantizar una buena satisfacción con el sistema realizado deben ser capaces de detectar las posibles incongruencias o errores del software. Existen disímiles tipos de pruebas, las cuales están en correspondencia a la metodología utilizada en el desarrollo de un sistema informático. Algunos tipos de prueba son:

- Pruebas de Regresión: se realizan principalmente sobre una aplicación que no se tiene documentación o ningún tipo de pruebas, por lo que estas pruebas se deben realizar antes de hacerle una modificación a la aplicación. Por lo tanto, solo hay que hacer pruebas de funcionamiento básico y no exhaustivo preferentemente (35).

- Pruebas Funcionales: son pruebas similares a las de aceptación con la diferencia que sí son técnicas y por lo tanto deberán incluir cada uno de los requerimientos funcionales (35).

- Pruebas de Calidad de Código: este tipo de pruebas sirve para garantizar que la calidad del código es realmente óptima y que la probabilidad de tener errores en la codificación es mínima (35).

Los niveles de pruebas también son tipos de pruebas que se le pueden realizar a los sistemas de software.

3.3.2 Método de prueba

Los métodos de prueba del software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo. Los tipos de prueba que se realizan a los software están compuestos por diferentes métodos que estas realizan para lograr un correcto resultado al aplicarse. A continuación se mencionan los métodos utilizados por algunas de las pruebas de software más utilizadas (36).

Tabla 3.1 Métodos de Prueba

Pruebas	**Métodos**	Descripción

Caja Blanca	Prueba del Camino Básico.	Es un indicador del número de caminos independientes que existen en un grafo.
	Prueba de Condición.	Esta prueba utiliza dos estrategias para su uso, las Pruebas de Ramificaciones y las Pruebas de Dominio
	Prueba de Flujo de Datos.	Selecciona caminos de un programa de acuerdo a las definiciones y uso de las variables.
	Prueba de Bucles.	Este método de prueba se divide en otros tipos de prueba para facilitar su uso. (Bucles simples, Bucles anidados, bucles concatenados, bucles no estructurados)
Caja Negra	Métodos de prueba basados en grafos.	Define una serie de pruebas que verifican que "todos los objetos tienen entre ellos la relaciones esperadas.
	Partición Equivalente.	Divide el dominio de entrada de un programa en clases de datos. El diseño de casos de prueba para la partición equivalente se basa en la evaluación de las clases de equivalencia.

	Análisis de valores límite.	Nos lleva a elegir las pruebas que nos ejecuten los valores límite, con esta técnica se complementa la partición equivalente.
	Prueba de Comparación.	Esta técnica consiste en la comparación de salidas de un mismo software pero de sus diferentes versiones.
	Conjetura de Errores.	Enumera una lista de equivocaciones que pueden cometer los desarrolladores. Genera los casos de prueba en base a dicha lista y esta generación de casos se obtiene en base a la intuición o la experiencia.
Otras Pruebas	Prueba de la Caja de Cristal.	Consiste en abstenerse de realizar pruebas de depurar bastante bien un proyecto; se deja al cliente que lo ensaye y acepte. El resultado es una bomba de tiempo.

	Pruebas Aleatorias	Se simula los posibles datos de entrada en la secuencia y frecuencia que pueden aparecer en la práctica. Si el proceso de generación se ha realizado correctamente, se crearán eventualmente todas las posibles entradas del programa en todas las posibles combinaciones y permutaciones

3.3.3 Estrategia de Prueba seguida

La estrategia de prueba utilizada para verificar la calidad del sub-módulo de seguridad del sistema ABCD 3.0 es la de pruebas funcionales de las que se utilizó el tipo de prueba caja negra. Estas pruebas fueron realizadas de manera estricta a partir de los casos de prueba elaborados. Para su implementación se utilizó la técnica partición de equivalencia que permitió realizar los diseños de casos de prueba, evaluando el conjunto de entradas y salidas sobre las que se trabajó, teniendo en cuenta que en todo programa hay un grupo de entradas que causan un comportamiento erróneo en los sistemas, y como consecuencia producen una serie de salidas que revelan la presencia de defectos.

Realizar las pruebas de caja negra permitió detectar, documentar y solucionar los errores existentes en el sistema implementado. Se validó que la herramienta creada satisface los requisitos identificados. En la primera iteración se encontraron cuatro no conformidades y ocho recomendaciones. Para la segunda iteración de las pruebas y utilizando la técnica de regresión no se encontraron deficiencias en el sistema.

Tabla 3.2: Resultados de las pruebas

No Conformidades	Recomendaciones

1- No aparecen deshabilitadas en las vistas las funcionalidades a las que no tienen acceso los usuarios.	1- Deshabilitar en las vistas las funcionalidades a las que un usuario no tiene acceso o eliminarlas del listado de funcionalidades a las que tiene acceso. 2- Realizar un filtro de las funcionalidades cuando el usuario se está autenticando en el sistema
2- El sistema muestra el mensaje de que existen campos obligatorios vacíos pero no aparece un indicador sobre estos campos.	3- Poner un indicador sobre los campos que son obligatorios cuando el usuario no le inserte datos. 4- Tener una opción que marque los campos que son obligatorios insertarles datos.
3- Hay campos que tienen diferentes formatos, por ejemplo la edad está como listbox y campo de texto.	5- Utilizar el formato de entrada de texto para los campos que tienen que ver directamente con las características del usuario que se autentica en el sistema. 6- Utilizar el listbox para aquellos campos que muestren variedades de opciones que permite elegir el sistema.
4- Cuando se solicita el registro de acceso de un determinado usuario del sistema aparece un mensaje de error aun mostrándose el resultado de la petición hecha al sistema.	7- Verificar los mensajes de respuestas del sistema. 8- Deshabilitar las opciones de modificar y eliminar cuando se muestra la lista de acceso de un usuario.

3.3.4 Diseño de caso de prueba o implementación de pruebas

Tabla 3.3: Caso de Prueba "*Iniciar Sesión*"

Escenario	Descripción	Usuario	Contraseña	Biblioteca	Respuesta del Sistema	Flujo Central
Esc1.1Iniciar Sesión	Se consulta con todos los datos de un usuario registrado en el sistema	V	V	V		
		lquintana	3art45*1	José Martí	Valida los datos. Permite el acceso al sistema. Muestra el nombre de usuario de la persona que se encuentra autenticada y la interfaz correspondiente al escritorio de trabajo del usuario en cuestión. Ver CP: Mostrar Escritorio de Trabajo.	El usuario accede a la opción que le permite iniciar sesión. Introduce y selecciona los datos del usuario. Accede al botón que le permite iniciar sesión: “Iniciar Sesión”
Esc 1.2 salir	El actor	NA	NA	NA	Permite	El usuario

de la vista actual	desea salir de la vista actual.				regresar a la vista anterior.	accede a la opción que le permite iniciar sesión. Introduce y selecciona los datos del usuario. Accede al botón que le permite iniciar sesión: “Iniciar Sesión”. Accede a la opción que le permite salir de la vista actual: “Salir”.
Esc 1.3 El	El usuario	NA	NA	NA	Cierra la	Accede a

usuario no usa el sistema en un período de 10 minutos	no utiliza la sesión en un período de 10 minutos				sesión. El sistema guarda el estado de los datos. Muestra una vista de inicio de sesión. Regresa al escenario EC1.1 paso 1.	la opción que le permite iniciar sesión. Accede al botón que le permite iniciar una sesión de usuario: "Iniciar Sesión". El usuario no utiliza la sesión por un período de 10 minutos.
Esc1.4Datos	El usuario	I	I	I	"Muestra un	Accede a

incorrectos	introduce datos incorrectos	la29*/	3se3	l509l7- */hg	mensaje de error: “El usuario o contraseña introducida es incorrecta”. Regresa al escenario Esc1.1 paso 3."	la opción que le permite iniciar sesión. Introduce datos no válidos. Accede al botón que le permite iniciar sesión de usuario: ""Iniciar Sesión"".
Esc1.5Datos	Se intenta	I(Vacío)	I(Vacío)	I(Vacío)	"Muestra un	Accede a

Vacíos	consultar una usuario pero existen campos vacíos que son obligatorios				mensaje de error: “Existen campos vacíos que son obligatorios, por favor complete estos campos”. Regresa al escenario EC1.1 paso 3.	la opción que le permite consultar un usuario. Accede al botón que le permite consultar un usuario: "Consultar" . "
Esc 1.6El	Se desea	V	V	V	"Muestra el	Accede a

usuario introduce datos erróneos por tres intentos consecutivos.	introducir los datos de un usuario.	lrquintana	r4t2*s	Máximo Gómez	mensaje de error “Ha sobrepasado los intentos permitidos para iniciar sesión, para mayor seguridad su perfil será bloqueado. Para obtener acceso nuevamente espere el tiempo establecido o contacte con el administrador del sistema”. Inhabilita el usuario. Regresa al escenario EC1.1 paso 2."	la opción que le permite iniciar sesión. Introduce y/o selecciona los datos del usuario. Accede al botón que le permite iniciar sesión: "Iniciar Sesión".
Esc 1.7El	Se teclea	V	V	V	Muestra un	Accede a

usuario no se encuentra registrado.	un nombre de usuario que no existe en el sistema	yvaldez	yt69	Hermanos Saiz	mensaje de información: “No se encontraron coincidencias”. Regresa al escenario EC1.1 paso 3.	la opción que le permite iniciar sesión. Introduce y selecciona los datos del usuario. Accede al botón que le permite iniciar sesión: "Iniciar Sesión". (Inicia sesión con un usuario que no existe)

Conclusiones del capítulo

Con el desarrollo del presente capítulo dedicado a la implementación y las pruebas del sistema ABCD 3.0 se describieron las clases del sistema que responden a las necesidades del negocio, lográndose un mejor entendimiento del mismo. Se obtuvo el diagrama de componentes físico del sistema, modelado en tres capas para un mejor entendimiento de los cambios que pueden producirse en el sistema La realización de las pruebas determinó las deficiencias del sistema

desarrollado y propició la solución satisfactoriamente de las mismas.

Recomendaciones

CONCLUSIONES

Una vez finalizada la presente investigación y después de haber cumplido con el objetivo y las tareas propuestas, se arribaron a las siguientes conclusiones:

1. El análisis a los sistemas informáticos relacionados con el campo de acción contribuyeron a identificar una solución a la problemática planteada, así como a determinar un mecanismo eficiente para implementar la seguridad del sistema.

2. El modelado de los artefactos necesarios, según la metodología de software asumida, correspondientes a las fases Análisis, diseño e Implementación, permitió el desarrollo de las funcionalidades del módulo administración de la seguridad. Obteniéndose resultados concretos durante el proceso de desarrollo del software.

3. La implementación se basó en tecnologías de desarrollo disponibles y que aseguran el cumplimiento de los requerimientos y la construcción de funcionalidades completamente integradas al módulo de administración de la seguridad.

RECOMENDACIONES

Se recomienda para próximas versiones del sub-módulo de administración de la seguridad del sistema ABCD desarrollado en la UCI: implementar un nuevo algoritmo de encriptación para las contraseñas pues aunque el algoritmo utilizado es lo suficientemente robusto por ahora, con el desarrollo de software hackers más inteligente este pudiera ser quebrantado.

REFERENCIAS BIBLIOGRÁFICAS

1. *Seguridad en redes de ordenadores: Control de Acceso.* [pdf] 2012. ISBN.

2. Sistemas Integrados de Gestión Bibliotecaria: una visión general. *Sistemas Integrados de Gestión Bibliotecaria: una visión general.* [En línea] Doknos, 2 de marzo de 2010. [Citado el: 8 de febrero de 2015.] http://www.doknos.com/node/127. ISBN.

3. **Cuozzo, Lic. Gabriela.** academia.edu. *academia.edu.* [En línea] 12 de junio de 2013. [Citado el: 24 de febrero de 2015.] http://www.academia.edu/8337474/_Los_Sistemas_integrados_de_gesti%C3%B3n_bibliotecarias_SIGB_oportunidades_y_o_desventajas_que_ofrecen_hoy_. ISBN.

4. **ERB, Markus.** Gestion de Riesgos en la Seguridad Informática. *Gestion de Riesgos en la Seguridad Informática.* [En línea] WordPress.com, 27 de septiembre de 2011. [Citado el: 14 de marzo de 2015.] https://protejete.wordpress.com/gdr_principal/definicion_si/. ISBN.

5. **Campos, Ilse Cuevas.** Prezi.com. *Prezi.com.* [En línea] 30 de septiembre de 2013. [Citado el: 12 de febrero de 2015.] https://prezi.com/qo57opyclkwv/seguridad-informatica/. ISBN.

6. **Copyright.** Protege tu informacion.com. *Protege tu informacion.com.* [En línea] ISMS Forum Spain, 23 de marzo de 2011. [Citado el: 19 de febrero de 2015.] http://www.protegetuinformacion.com/perfil_tema.php?id_perfil=6&id_tema=56. ISBN.

7. **Urgiles, Johana.** blogspot.com. *blogspot.com.* [En línea] 22 de mayo de 2013. [Citado el: 21 de febrero de 2015.] http://materiadeinformatica3q.blogspot.com/2013/05/mecanismos-de-seguridad-informatica.html. ISBN.

8. **Mifsud, Elvira.** Mecanismnos de seguridad. *Mecanismnos de seguridad.* [En línea] NIPO, 26 de marzo de 2012. [Citado el: 26 de mayo de 2015.] http://recursostic.educacion.es/observatorio/web/en/software/software-general/1040-introduccion-a-la-seguridad-informatica?start=4. ISBN.

9. **Kioskea.** *es.kioskea.net.* [pdf] s.l. : Creative Commons, 10 de junio de 2014. ISBN.

10. **López, Antonio.** Mecanismos Básicos de Control de Acceso. *Mecanismos Básicos de Control de Acceso.* [En línea] 27 de noviembre de 2014. [Citado el: 11 de marzo de 2015.] https://www.incibe.es/blogs/post/Seguridad/BlogSeguridad/Articulo_y_comentarios/control_acceso. ISBN.

11. **Mifsud, Elvira.** Observatorio Tecnológico. *Observatorio Tecnológico.* [En línea] 26 de marzo de 2012. [Citado el: 3 de marzo de 2015.] http://recursostic.educacion.es/observatorio/web/es/component/content/article/1040-introduccion-a-la-seguridad-informatica?start=4. ISBN.

12. **NAVA, JAVIER GIMENO.** spi1.nisu.org/. *spi1.nisu.org/.* [En línea] 16 de octubre de 2014. [Citado el: 5 de marzo de 2015.] http://spi1.nisu.org/recop/al01/javier/biblo.html. ISBN.

13. **Pérez, Jorge.** Prezi.com. *Prezi.com.* [En línea] Prezi, 3 de noviembre de 2012. [Citado el: 14 de marzo de 2015.] https://prezi.com/z429_zxqbm5g/mecanismos-para-la-deteccion-de-ataques-e-intrusiones/. ISBN.

14. **planeacioncentrosinformatica.blogspot.com.** planeacioncentrosinformatica.blogspot.com. *planeacioncentrosinformatica.blogspot.com.* [En línea] Arhesi, 20 de marzo de 2010. [Citado el: 14

de marzo de 2015.] http://planeacioncentrosinformatica.blogspot.com/2010_03_01_archive.html. ISBN.

15. *Unidad 4: Seguridad en Redes y Sistemas Operativos.* **Aboy, Dr. Juan José Aranda.** Quito : Universidad de Las Américas, 2006. ISBN.

16. **Diago, Javier Vela.** Security ArtWork. *Security ArtWork.* [En línea] S2 Grupo, 12 de marzo de 2010. [Citado el: 24 de abril de 2015.] http://www.securityartwork.es/2010/03/12/sistemas-de-control-de-acceso-mac-y-dac/. ISBN.

17. **InterSoft S.A.** mastersolutions.com.ar. *mastersolutions.com.ar.* [En línea] 7 de mayo de 2004. [Citado el: 24 de abril de 2015.] http://mastersolutions.com.ar/ideafix/Parte%20V/Sistema%20de%20Control%20de%20Accesos/Seccion%201-Seguridad%20Informatica/HTML-PDF/Capitulo-01.html. ISBN.

18. **Britos, José Manuel.** Scribd.com. *Scribd.com.* [En línea] 1 de septiembre de 2010. [Citado el: 16 de marzo de 2015.] http://es.scribd.com/doc/125508422/Deteccion-de-Intrusines#scribd. ISBN.

19. **Seguridad en Redes.** Seguridad en Redes. *Seguridad en Redes.* [En línea] 14 de octubre de 2009. [Citado el: 12 de junio de 2015.] http://seguridadredesmedina.blogspot.com/2009/10/md5-definicion-y-aplicaciones.html. ISBN.

20. **CP-Lab.com.** CP-Lab.com. *CP-Lab.com.* [En línea] 1 de febrero de 2015. [Citado el: 12 de junio de 2015.] http://www.cp-lab.com/es/cryptography.html. ISBN.

21. **Maceo, Jorge Enriquez Ricardo.** [En línea] 24 de junio de 2013. [Citado el: 12 de junio de 2015.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/8689/1/TD_06884_13.pdf. ISBN.

22. **cusa, Yazmin Galvez.** [En línea] 23 de junio de 2013. [Citado el: 17 de junio de 2015.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/8388/1/TD_06387_13.pdf. ISBN.

23. **Millent, Dailen Barrera.** Biblioteca.uci.cu. *Biblioteca.uci.cu.* [En línea] 26 de junio de 2013. [Citado el: 14 de junio de 2015.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/8180/1/TD_06449_13.pdf. ISBN.

24. IBM Center Knowledge. *IBM Center Knowledge.* [En línea] 22 de abril de 2014. [Citado el: 16 de abril de 2015.] http://www-01.ibm.com/support/knowledgecenter/SS8PJ7_9.1.0/com.ibm.aries.osgi.doc/topics/cosgiarchitecture.html?lang=es. ISBN.

25. **Domingo, José.** Seguridad en Redes. *Seguridad en Redes.* [En línea] WordPress, 18 de abril de 2013. [Citado el: 4 de mayo de 2015.] https://statusexcessu.wordpress.com/2013/04/18/sha-1/. ISBN.

26. **Botta, Adrián.** Scribd.com. *Scribd.com.* [En línea] Sistemasutn, 13 de marzo de 2012. [Citado el: 12 de mayo de 2015.] http://es.scribd.com/doc/85235594/Diseno-de-Sistemas-Resumen-Completo-v1-1#scribd. ISBN.

27. **Vega, Miguel.** *Casos de Uso.* [slideshare] Granada : UGR, 2010. ISBN.

28. **SG Buzz.** SG Buzz. *SG Buzz.* [En línea] 3 de marzo de 2011. [Citado el: 17 de febrero de 2015.] http://sg.com.mx/content/view/510. ISBN.

29. *Arquitectura de Software.* **Cervantes, humberto.** 27, Madrid : SG, 2010, Vol. Arquitectura. ISBN.

30. **Cardona, Juan David Nicholls.** Tips de Desarrollo Web. *Tips de Desarrollo Web.* [En línea] BlogEngine.NET, 9 de julio de 2013. [Citado el: 13 de mayo de 2015.] http://www.nicholls.co/blog/post/Arquitectura-N-Capas-y-LinqToSQL. ISBN.

31. **ABCD.** *Presentación del Colectivo Técniso ABCD.* La Habana : s.n., 2014. ISBN.

32. **Jimenez, Carlos josé Requena.** NewBloggerThemes.com#sthash.EcCCOFqG.dpuf. *NewBloggerThemes.com#sthash.EcCCOFqG.dpuf.* [En línea] Web2feel, 25 de junio de 2012. [Citado el: 7 de mayo de 2015.] http://carlosjoserequena.blogspot.com/2013/01/arquitectura-de-software-y-patrones-de_14.html. ISBN.

33. **Mora, Roberto Canales.** AdictosAlTrabajo.com. *AdictosAlTrabajo.com.* [En línea] Autentia, 22 de diciembre de 2009. [Citado el: 15 de mayo de 2015.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=grasp. ISBN.

34. **it Mentor.** *Pruebas de Software.* [pdf] 2012. ISBN.

35. **Montoya, Eliu.** EliuMM. *EliuMM.* [En línea] Java & Arch Blog, 11 de septiembre de 2012. [Citado el: 28 de mayo de 2015.] http://javablog.eliumontoya.com/home/tipodepruebasparadesarrollodesoftware. ISBN.

36. **utm.mx.** utm.mx. *utm.mx.* [En línea] 5 de junio de 2010. [Citado el: 28 de mayo de 2015.] http://www.utm.mx/~dtorres/cursos/ingsw/tema5.pdf. ISBN.

Bibliografía

1. *Seguridad en redes de ordenadores: Control de Acceso.* [pdf] 2012. ISBN.

2. Sistemas Integrados de Gestión Bibliotecaria: una visión general. *Sistemas Integrados de Gestión Bibliotecaria: una visión general.* [En línea] Doknos, 2 de marzo de 2010. [Citado el: 8 de febrero de 2015.] http://www.doknos.com/node/127. ISBN.

3. **Cuozzo, Lic. Gabriela.** academia.edu. *academia.edu.* [En línea] 12 de junio de 2013. [Citado el: 24 de febrero de 2015.] http://www.academia.edu/8337474/_Los_Sistemas_integrados_de_gesti%C3%B3n_bibliotecarias_SIGB_oportunidades_y_o_desventajas_que_ofrecen_hoy_. ISBN.

4. **ERB, Markus.** Gestion de Riesgos en la Seguridad Informática. *Gestion de Riesgos en la Seguridad Informática.* [En línea] WordPress.com, 27 de septiembre de 2011. [Citado el: 14 de marzo de 2015.] https://protejete.wordpress.com/gdr_principal/definicion_si/. ISBN.

5. **Campos, Ilse Cuevas.** Prezi.com. *Prezi.com.* [En línea] 30 de septiembre de 2013. [Citado el: 12 de febrero de 2015.] https://prezi.com/qo57opyclkwv/seguridad-informatica/. ISBN.

6. **Copyright.** Protege tu informacion.com. *Protege tu informacion.com.* [En línea] ISMS Forum Spain, 23 de marzo de 2011. [Citado el: 19 de febrero de 2015.] http://www.protegetuinformacion.com/perfil_tema.php?id_perfil=6&id_tema=56. ISBN.

7. **Urgiles, Johana.** blogspot.com. *blogspot.com.* [En línea] 22 de mayo de 2013. [Citado el: 21 de febrero de 2015.] http://materiadeinformatica3q.blogspot.com/2013/05/mecanismos-de-seguridad-informatica.html. ISBN.

8. **Mifsud, Elvira.** Mecanismnos de seguridad. *Mecanismnos de seguridad.* [En línea] NIPO, 26 de marzo de 2012. [Citado el: 26 de mayo de 2015.] http://recursostic.educacion.es/observatorio/web/en/software/software-general/1040-introduccion-a-la-seguridad-informatica?start=4. ISBN.

9. **Kioskea.** *es.kioskea.net.* [pdf] s.l. : Creative Commons, 10 de junio de 2014. ISBN.

10. **López, Antonio.** Mecanismos Básicos de Control de Acceso. *Mecanismos Básicos de Control de Acceso.* [En línea] 27 de noviembre de 2014. [Citado el: 11 de marzo de 2015.] https://www.incibe.es/blogs/post/Seguridad/BlogSeguridad/Articulo_y_comentarios/control_acceso. ISBN.

11. **Mifsud, Elvira.** Observatorio Tecnológico. *Observatorio Tecnológico.* [En línea] 26 de marzo de 2012. [Citado el: 3 de marzo de 2015.] http://recursostic.educacion.es/observatorio/web/es/component/content/article/1040-introduccion-a-la-seguridad-informatica?start=4. ISBN.

12. **NAVA, JAVIER GIMENO.** spi1.nisu.org/. *spi1.nisu.org/.* [En línea] 16 de octubre de 2014. [Citado el: 5 de marzo de 2015.] http://spi1.nisu.org/recop/al01/javier/biblo.html. ISBN.

13. **Pérez, Jorge.** Prezi.com. *Prezi.com.* [En línea] Prezi, 3 de noviembre de 2012. [Citado el: 14 de marzo de 2015.] https://prezi.com/z429_zxqbm5g/mecanismos-para-la-deteccion-de-ataques-e-intrusiones/. ISBN.

14. **planeacioncentrosinformatica.blogspot.com.** planeacioncentrosinformatica.blogspot.com. *planeacioncentrosinformatica.blogspot.com.* [En línea] Arhesi, 20 de marzo de 2010. [Citado el: 14 de marzo de 2015.] http://planeacioncentrosinformatica.blogspot.com/2010_03_01_archive.html. ISBN.

Bibliografía

15. *Unidad 4: Seguridad en Redes y Sistemas Operativos.* **Aboy, Dr. Juan José Aranda.** Quito : Universidad de Las Américas, 2006. ISBN.

16. **Diago, Javier Vela.** Security ArtWork. *Security ArtWork.* [En línea] S2 Grupo, 12 de marzo de 2010. [Citado el: 24 de abril de 2015.] http://www.securityartwork.es/2010/03/12/sistemas-de-control-de-acceso-mac-y-dac/. ISBN.

17. **InterSoft S.A.** mastersolutions.com.ar. *mastersolutions.com.ar.* [En línea] 7 de mayo de 2004. [Citado el: 24 de abril de 2015.] http://mastersolutions.com.ar/ideafix/Parte%20V/Sistema%20de%20Control%20de%20Accesos/Seccion%201-Seguridad%20Informatica/HTML-PDF/Capitulo-01.html. ISBN.

18. **Britos, José Manuel.** Scribd.com. *Scribd.com.* [En línea] 1 de septiembre de 2010. [Citado el: 16 de marzo de 2015.] http://es.scribd.com/doc/125508422/Deteccion-de-Intrusines#scribd. ISBN.

19. **Seguridad en Redes.** Seguridad en Redes. *Seguridad en Redes.* [En línea] 14 de octubre de 2009. [Citado el: 12 de junio de 2015.] http://seguridadredesmedina.blogspot.com/2009/10/md5-definicion-y-aplicaciones.html. ISBN.

20. **CP-Lab.com.** CP-Lab.com. *CP-Lab.com.* [En línea] 1 de febrero de 2015. [Citado el: 12 de junio de 2015.] http://www.cp-lab.com/es/cryptography.html. ISBN.

21. **Maceo, Jorge Enriquez Ricardo.** [En línea] 24 de junio de 2013. [Citado el: 12 de junio de 2015.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/8689/1/TD_06884_13.pdf. ISBN.

22. **cusa, Yazmin Galvez.** [En línea] 23 de junio de 2013. [Citado el: 17 de junio de 2015.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/8388/1/TD_06387_13.pdf. ISBN.

23. **Millent, Dailen Barrera.** Biblioteca.uci.cu. *Biblioteca.uci.cu.* [En línea] 26 de junio de 2013. [Citado el: 14 de junio de 2015.] http://repositorio_institucional.uci.cu/jspui/bitstream/ident/8180/1/TD_06449_13.pdf. ISBN.

24. IBM Center Knowledge. *IBM Center Knowledge.* [En línea] 22 de abril de 2014. [Citado el: 16 de abril de 2015.] http://www-01.ibm.com/support/knowledgecenter/SS8PJ7_9.1.0/com.ibm.aries.osgi.doc/topics/cosgiarchitecture.html?lang=es. ISBN.

25. **Domingo, José.** Seguridad en Redes. *Seguridad en Redes.* [En línea] WordPress, 18 de abril de 2013. [Citado el: 4 de mayo de 2015.] https://statusexcessu.wordpress.com/2013/04/18/sha-1/. ISBN.

26. **Botta, Adrián.** Scribd.com. *Scribd.com.* [En línea] Sistemasutn, 13 de marzo de 2012. [Citado el: 12 de mayo de 2015.] http://es.scribd.com/doc/85235594/Diseno-de-Sistemas-Resumen-Completo-v1-1#scribd. ISBN.

27. **Vega, Miguel.** *Casos de Uso.* [slideshare] Granada : UGR, 2010. ISBN.

28. **SG Buzz.** SG Buzz. *SG Buzz.* [En línea] 3 de marzo de 2011. [Citado el: 17 de febrero de 2015.] http://sg.com.mx/content/view/510. ISBN.

29. *Arquitectura de Software.* **Cervantes, humberto.** 27, Madrid : SG, 2010, Vol. Arquitectura. ISBN.

30. **Cardona, Juan David Nicholls.** Tips de Desarrollo Web. *Tips de Desarrollo Web.* [En línea] BlogEngine.NET, 9 de julio de 2013. [Citado el: 13 de mayo de 2015.] http://www.nicholls.co/blog/post/Arquitectura-N-Capas-y-LinqToSQL. ISBN.

31. **ABCD.** *Presentación del Colectivo Técniso ABCD.* La Habana : s.n., 2014. ISBN.

32. **Jimenez, Carlos josé Requena.** NewBloggerThemes.com#sthash.EcCCOFqG.dpuf. *NewBloggerThemes.com#sthash.EcCCOFqG.dpuf.* [En línea] Web2feel, 25 de junio de 2012. [Citado el: 7 de mayo de 2015.] http://carlosjoserequena.blogspot.com/2013/01/arquitectura-de-software-y-patrones-de_14.html. ISBN.

33. **Mora, Roberto Canales.** AdictosAlTrabajo.com. *AdictosAlTrabajo.com.* [En línea] Autentia, 22 de diciembre de 2009. [Citado el: 15 de mayo de 2015.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=grasp. ISBN.

34. **it Mentor.** *Pruebas de Software.* [pdf] 2012. ISBN.

35. **Montoya, Eliu.** EliuMM. *EliuMM.* [En línea] Java & Arch Blog, 11 de septiembre de 2012. [Citado el: 28 de mayo de 2015.] http://javablog.eliumontoya.com/home/tipodepruebasparadesarrollodesoftware. ISBN.

36. **utm.mx.** utm.mx. *utm.mx.* [En línea] 5 de junio de 2010. [Citado el: 28 de mayo de 2015.] http://www.utm.mx/~dtorres/cursos/ingsw/tema5.pdf. ISBN.

Tabla 1: Entrevista

Preguntas	Respuestas
¿Qué responsabilidad tiene como trabajador de la biblioteca y a qué servicios tiene acceso en el sistema ABCD 1.2?	
¿Cuándo se autentica en el sistema posee permisos de administrador sin tener altas responsabilidades?	
¿Una vez en el sistema, sin privilegios de administrador puede realizar cambios en la base de datos local de personas?	
¿Se tiene registros de las actividades de los usuarios en el sistema? En caso que sea cierto explique cómo se hace.	

Diagrama de secuencia: Registrar Persona

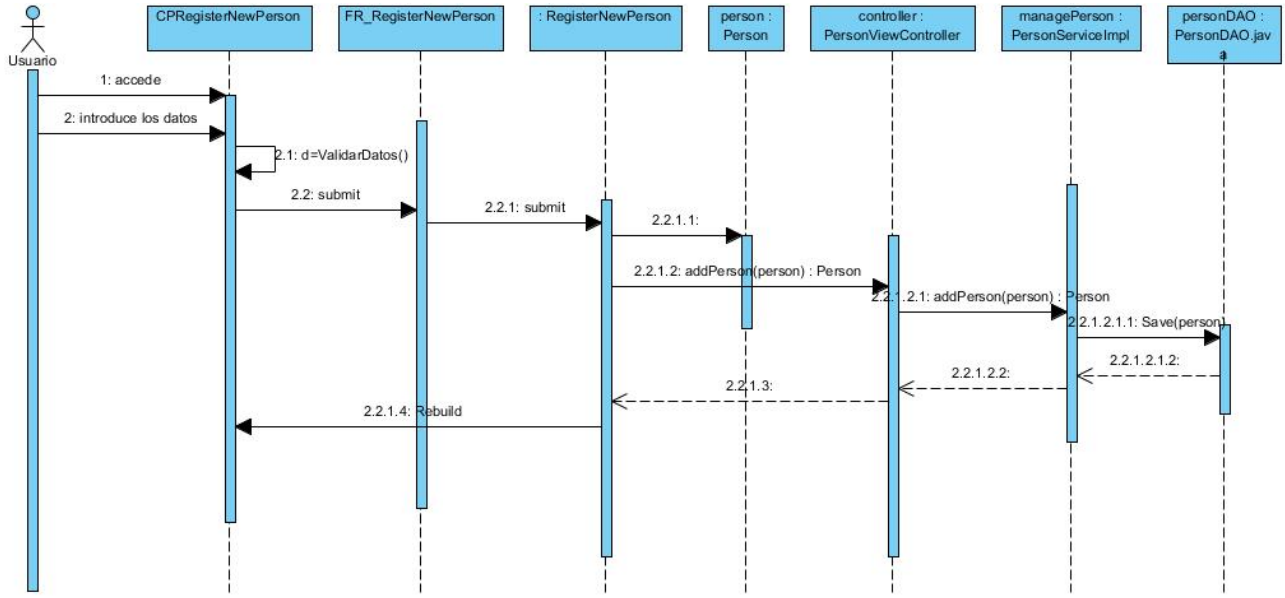


Diagrama de secuencia: Editar Persona

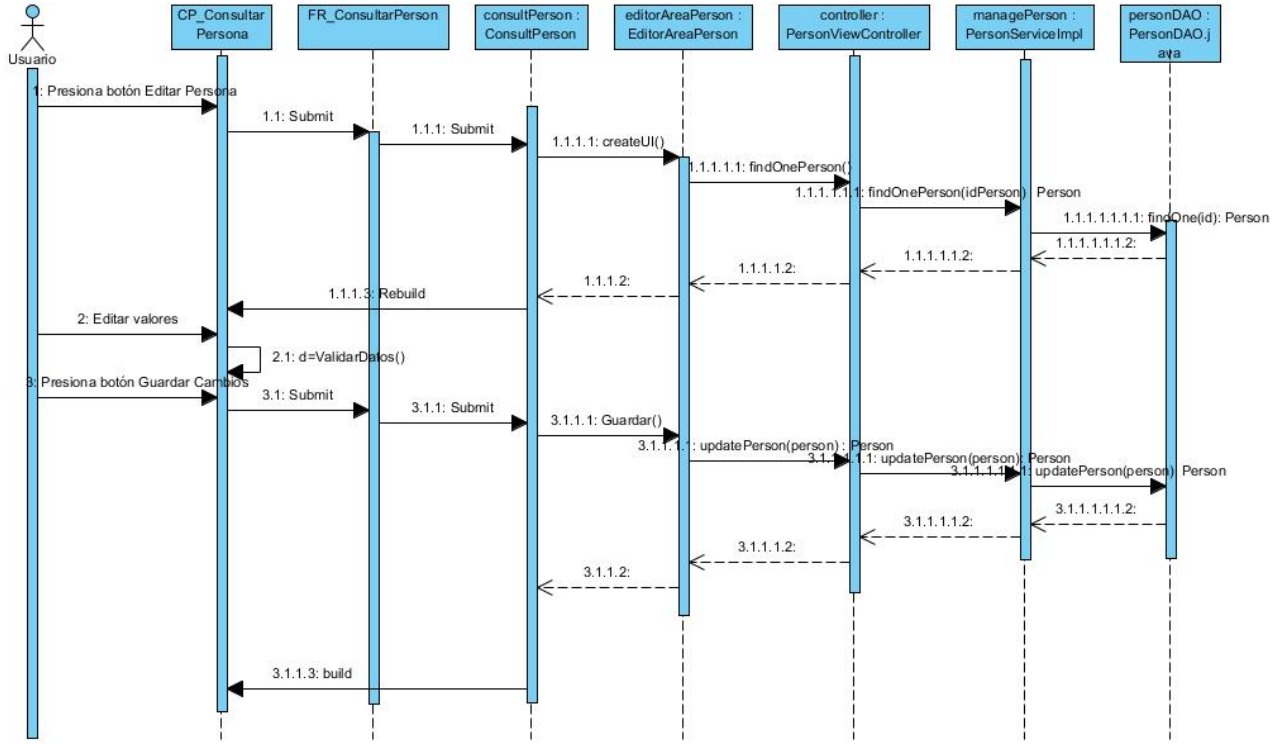


Diagrama de secuencia: Visualizar Persona

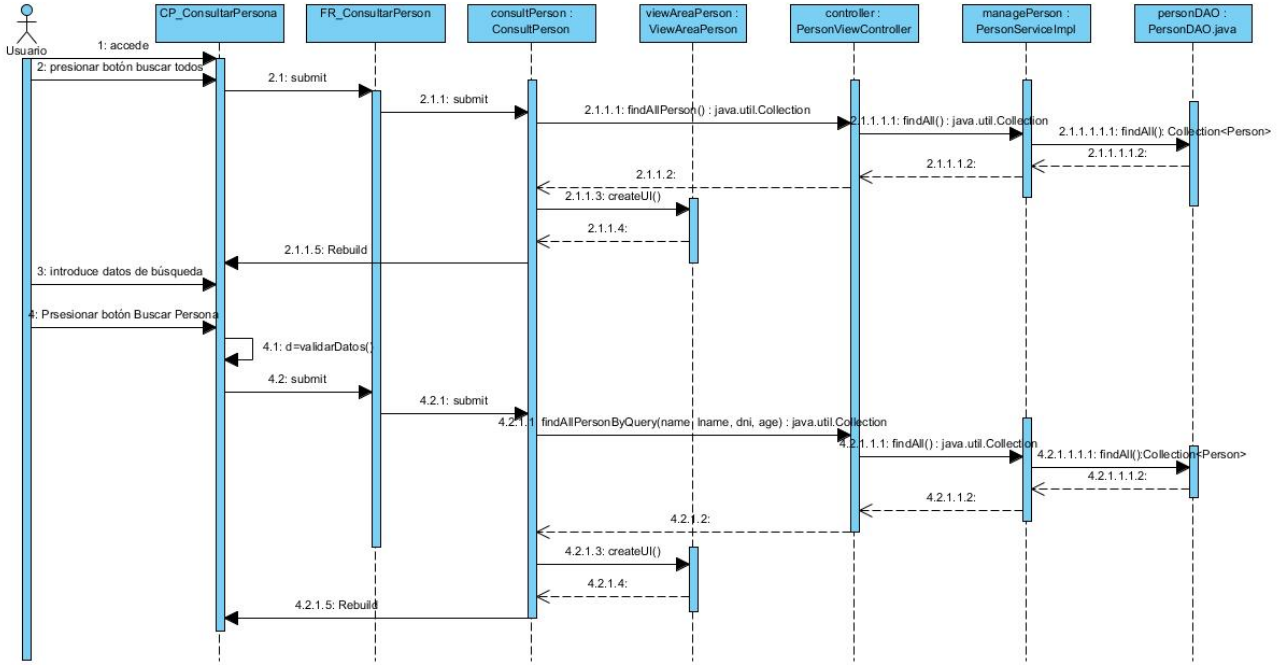


Diagrama de secuencia: Eliminar Persona

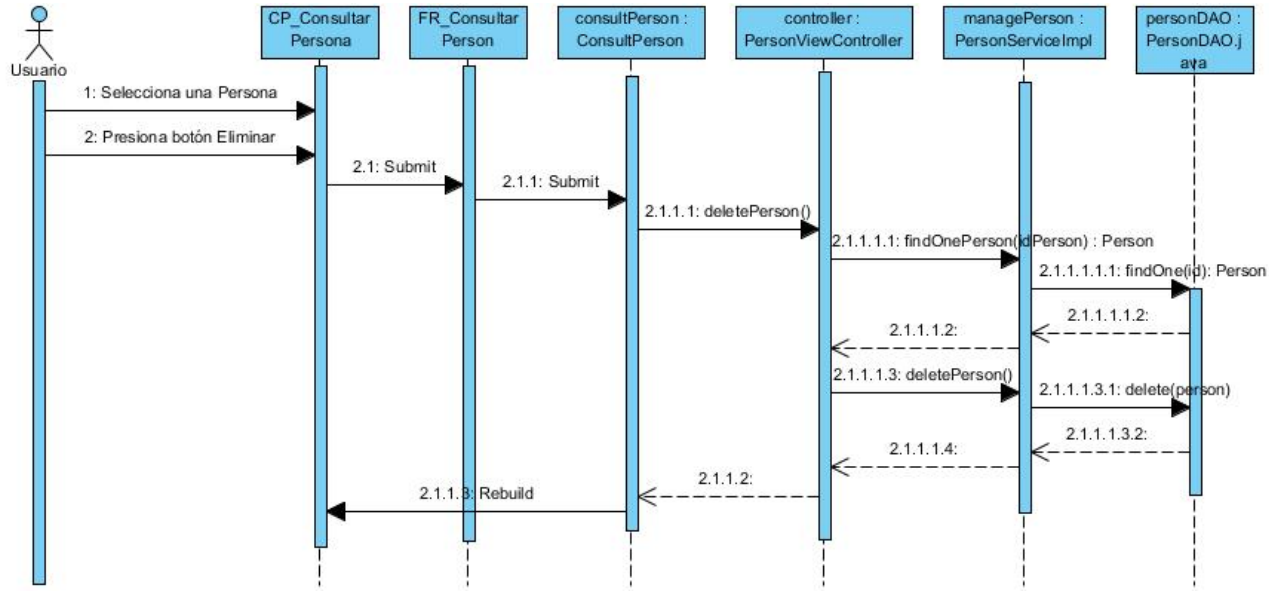


Diagrama de secuencia: Visualizar Usuario

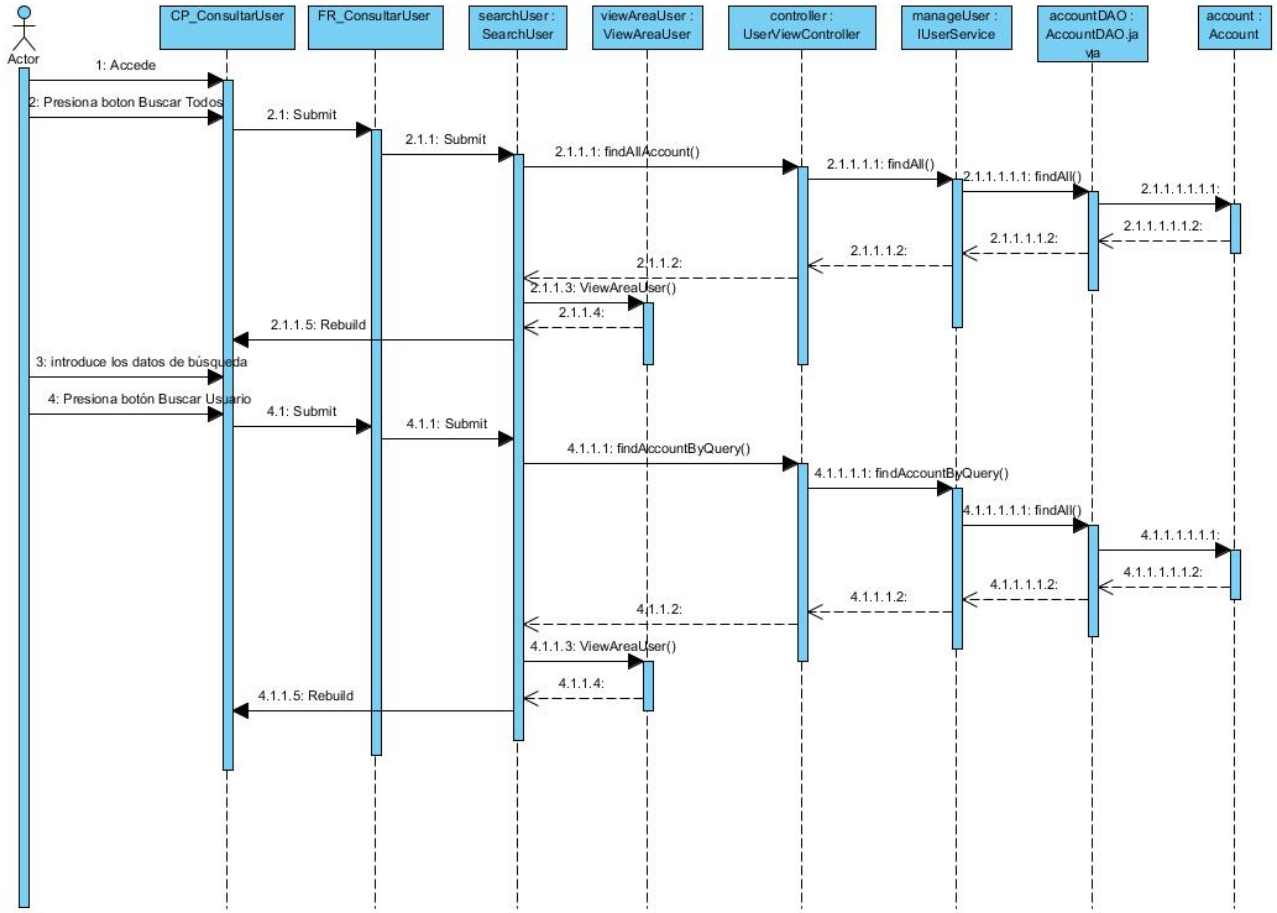


Diagrama de secuencia: Editar Usuario

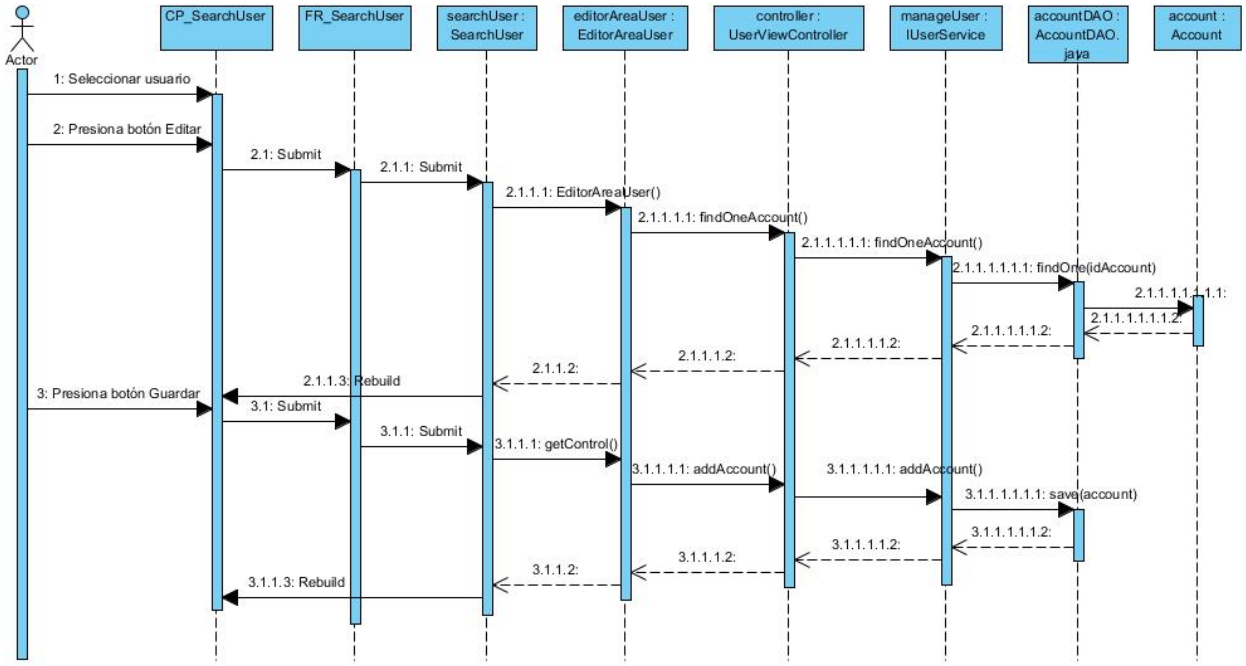


Diagrama de secuencia: Eliminar Usuario

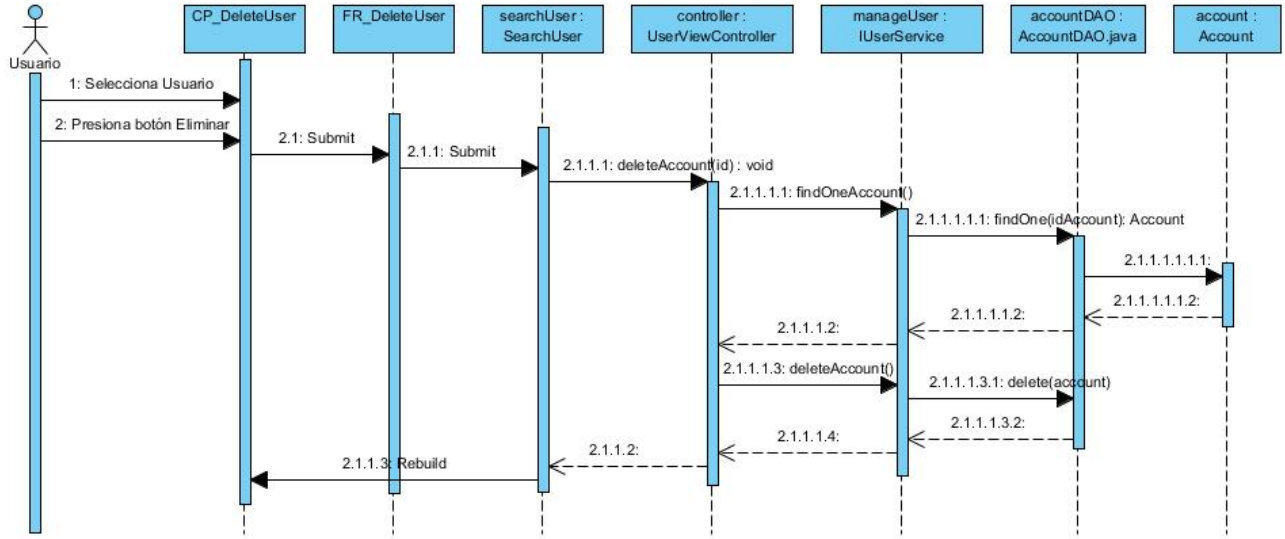


Diagrama de secuencia: Cambiar Contraseña

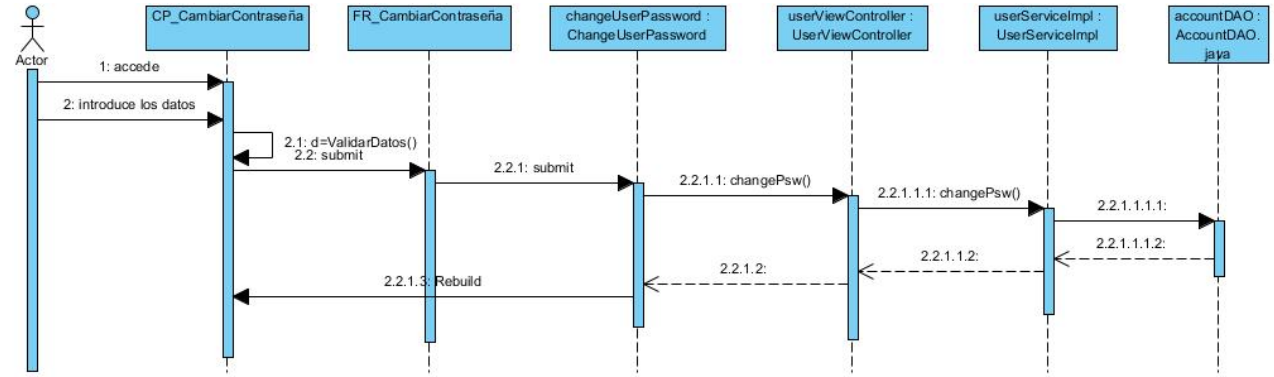


Diagrama de secuencia: Registrar Perfil de Usuario

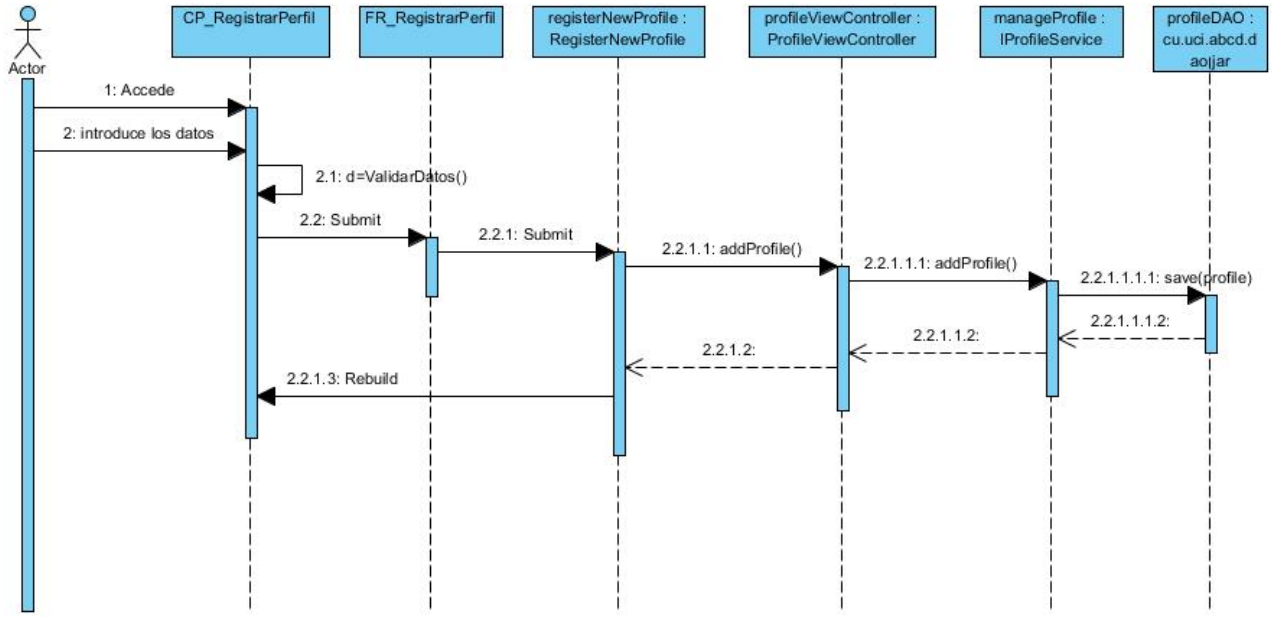


Diagrama de clases del diseño: Cambiar Contraseña

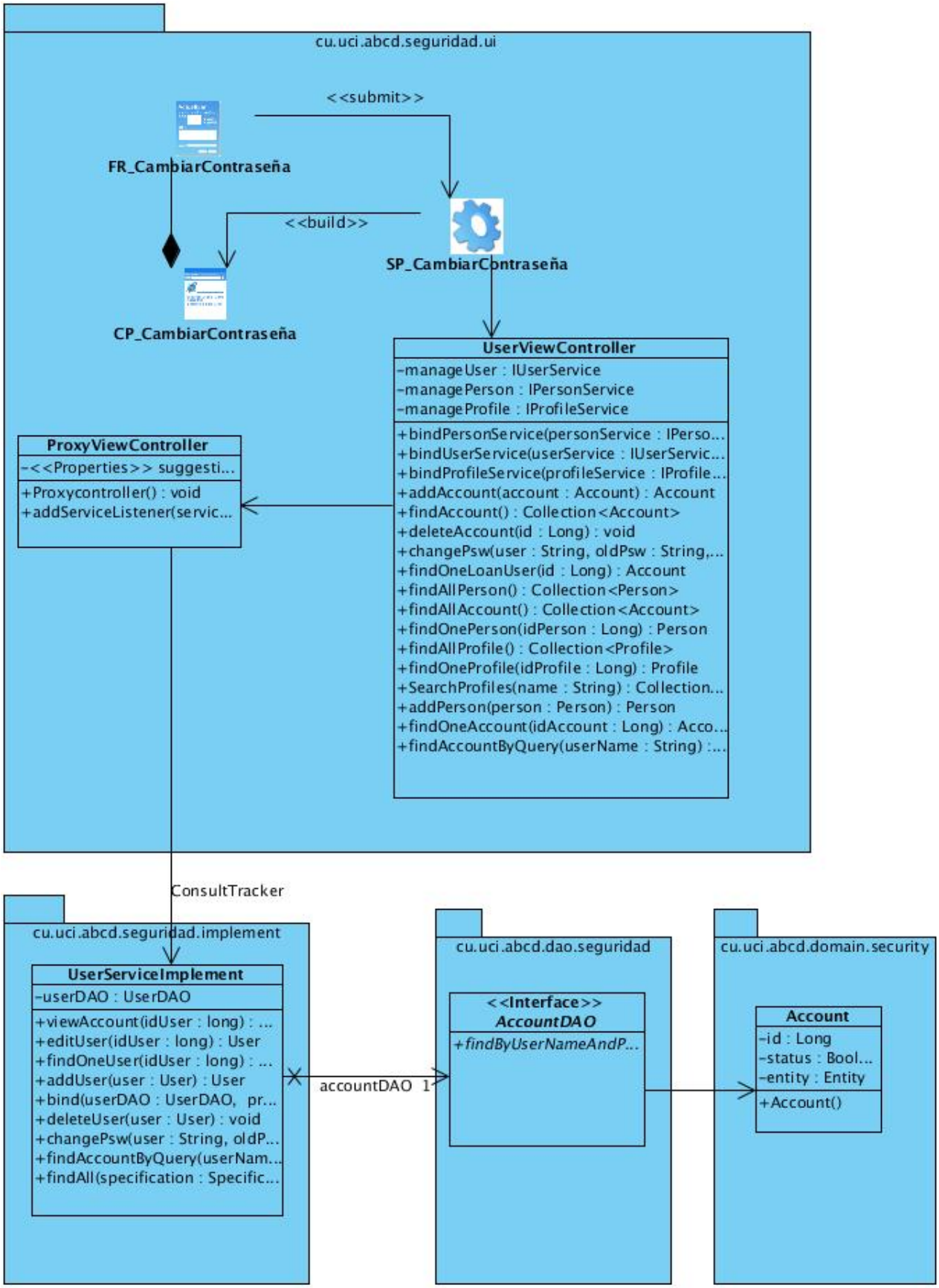

cu.uci.abcd.seguridad.ui
<<submit>>
FR_CambiarContraseña
<<build>>
SP_CambiarContraseña
CP_CambiarContraseña
UserViewController
-manageUser : IUserService
-managePerson : IPersonService
-manageProfile : IProfileService
+bindPersonService(personService : IPerso...
+bindUserService(userService : IUserServic...
+bindProfileService(profileService : IProfile...
+addAccount(account : Account) : Account
+findAccount() : Collection<Account>
+deleteAccount(id : Long) : void
+changePsw(user : String, oldPsw : String,...
+findOneLoanUser(id : Long) : Account
+findAllPerson() : Collection<Person>
+findAllAccount() : Collection<Account>
+findOnePerson(idPerson : Long) : Person
+findAllProfile() : Collection<Profile>
+findOneProfile(idProfile : Long) : Profile
+SearchProfiles(name : String) : Collection...
+addPerson(person : Person) : Person
+findOneAccount(idAccount : Long) : Acco...
+findAccountByQuery(userName : String) :...
ProxyViewController
-<<Properties>> suggesti...
+Proxycontroller() : void
+addServiceListener(servic...
ConsultTracker
cu.uci.abcd.seguridad.implement
UserServiceImplement
-userDAO : UserDAO
+viewAccount(idUser : long) : ...
+editUser(idUser : long) : User
+findOneUser(idUser : long) : ...
+addUser(user : User) : User
+bind(userDAO : UserDAO, pr...
+deleteUser(user : User) : void
+changePsw(user : String, oldP...
+findAccountByQuery(userNam...
+findAll(specification : Specific...
accountDAO 1
cu.uci.abcd.dao.seguridad
<<Interface>>
AccountDAO
+findByUserNameAndP...
cu.uci.abcd.domain.security
Account
-id : Long
-status : Bool...
-entity : Entity
+Account()

Diagrama de clase del diseño: Consultar Perfil de Usuario

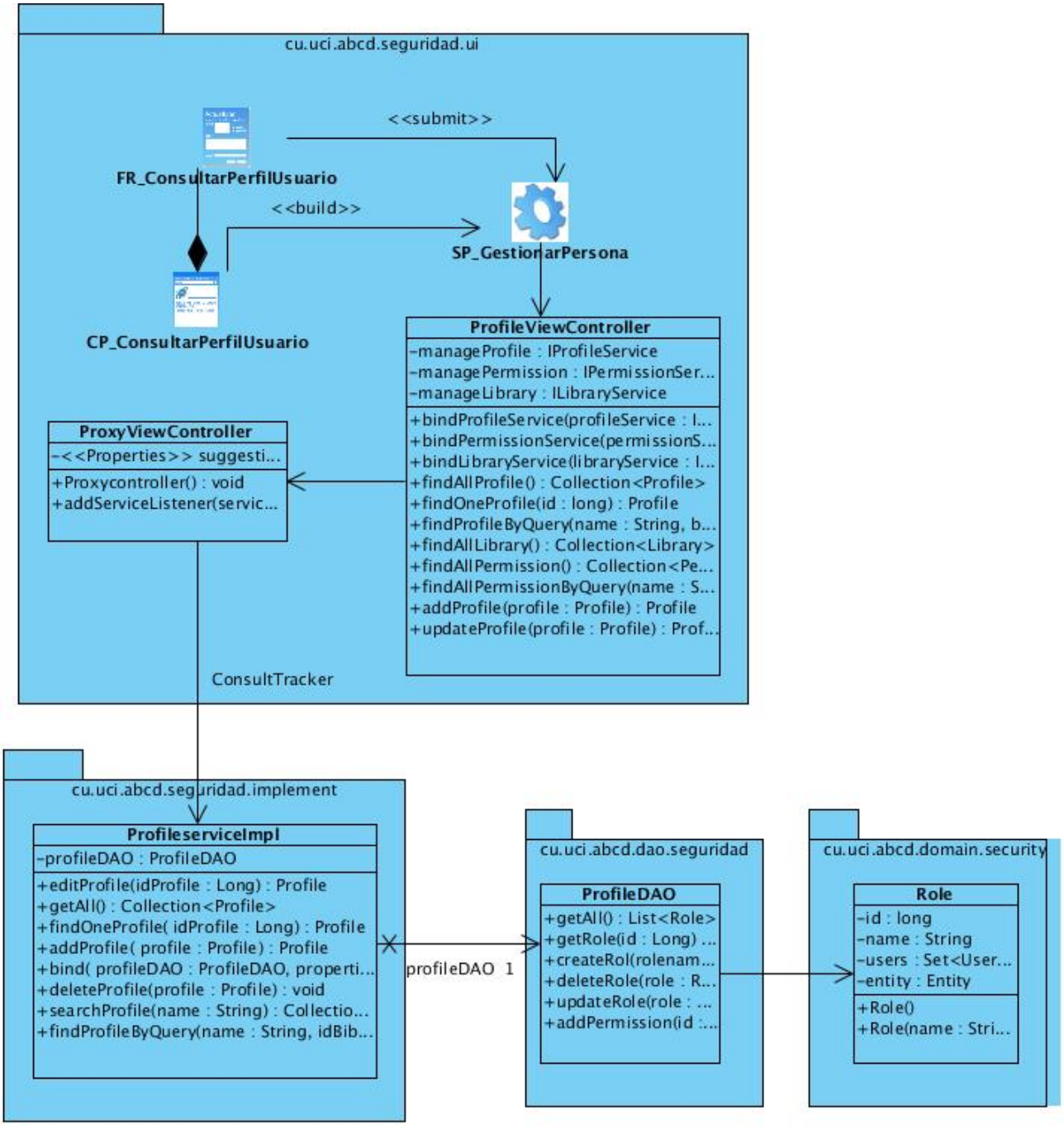


Diagrama de clase del diseño: Consultar Persona

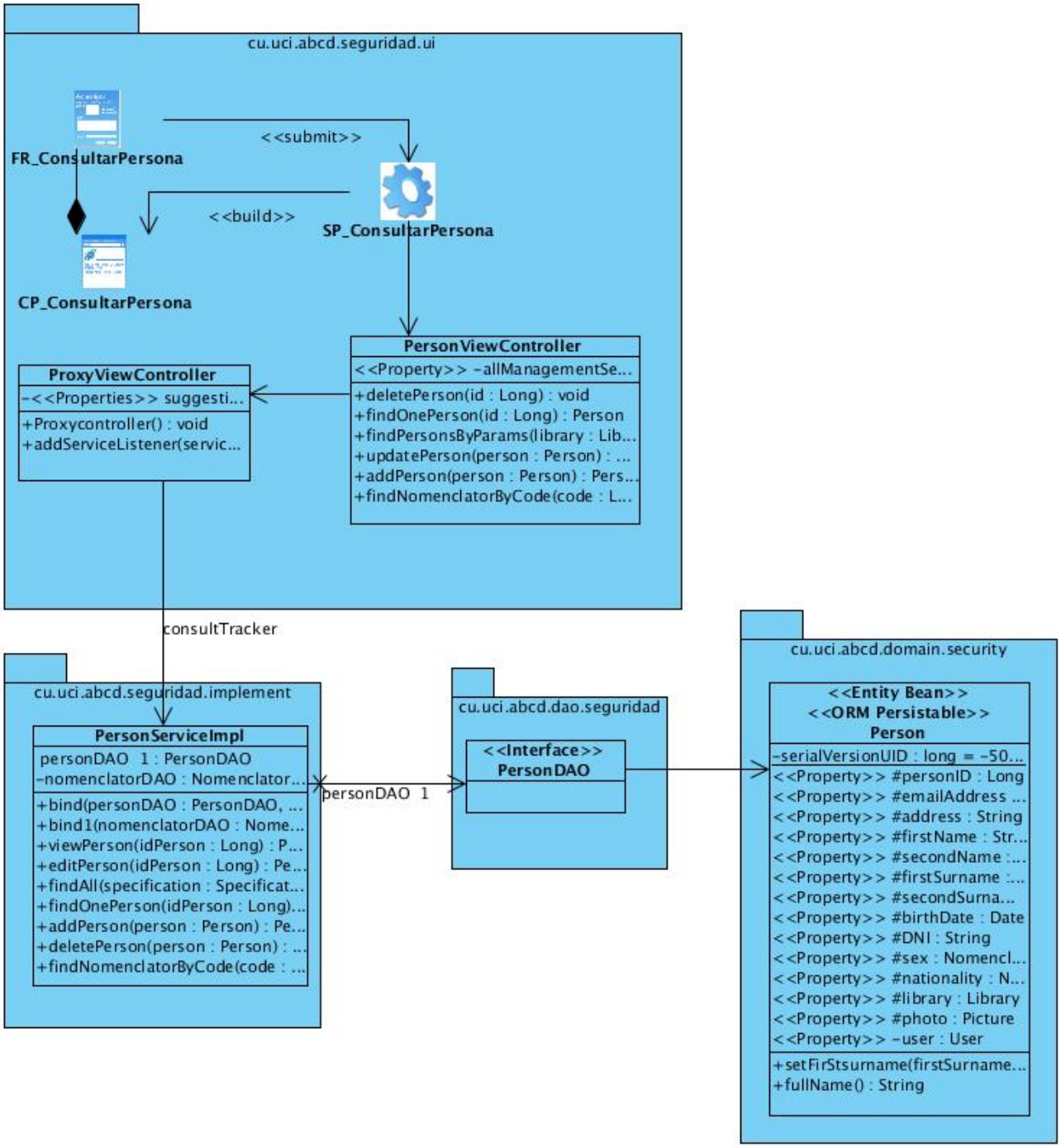


Diagrama de clases del diseño: Consultar Usuario

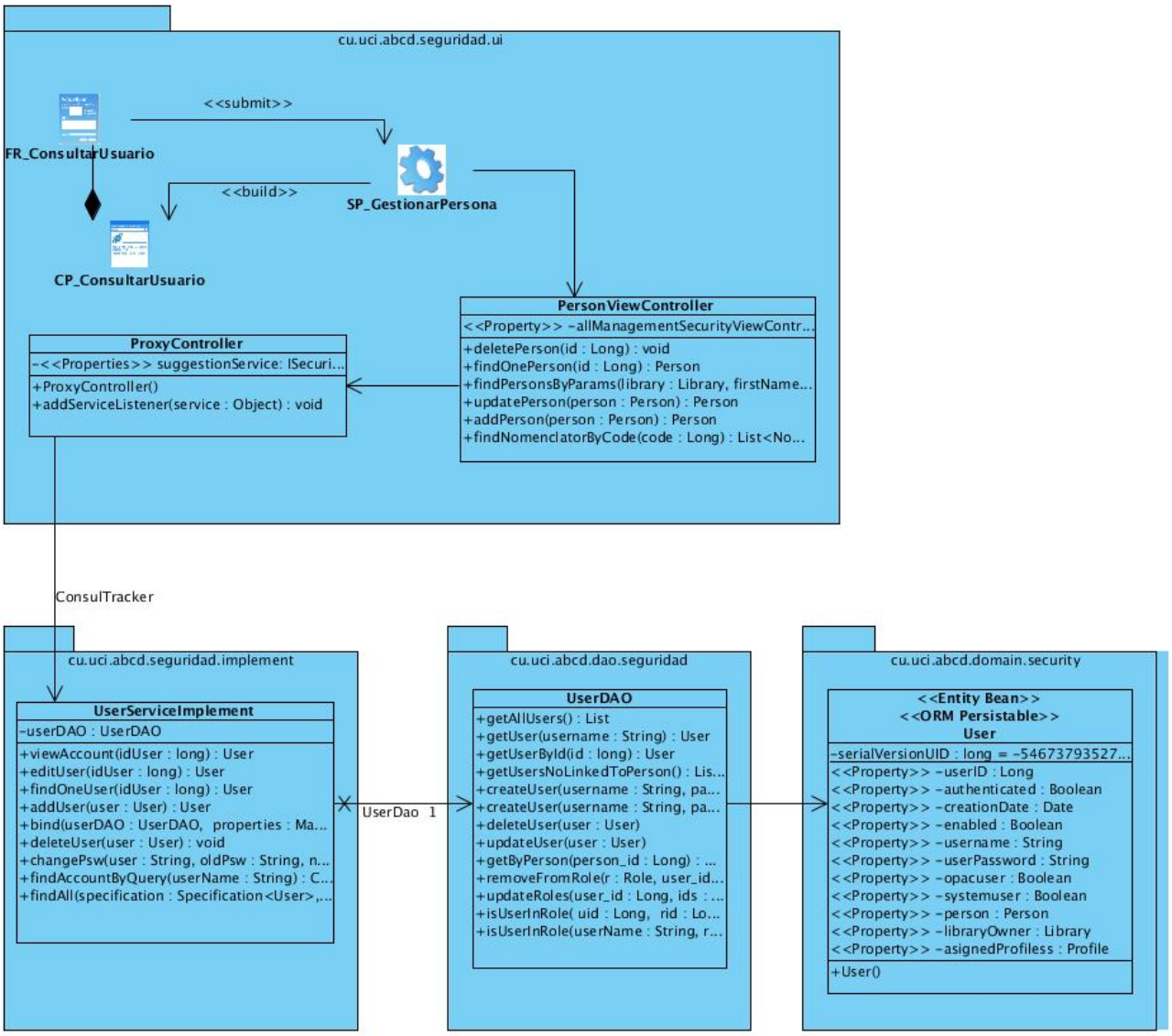


Diagrama de clase del diseño: Consultar Acceso

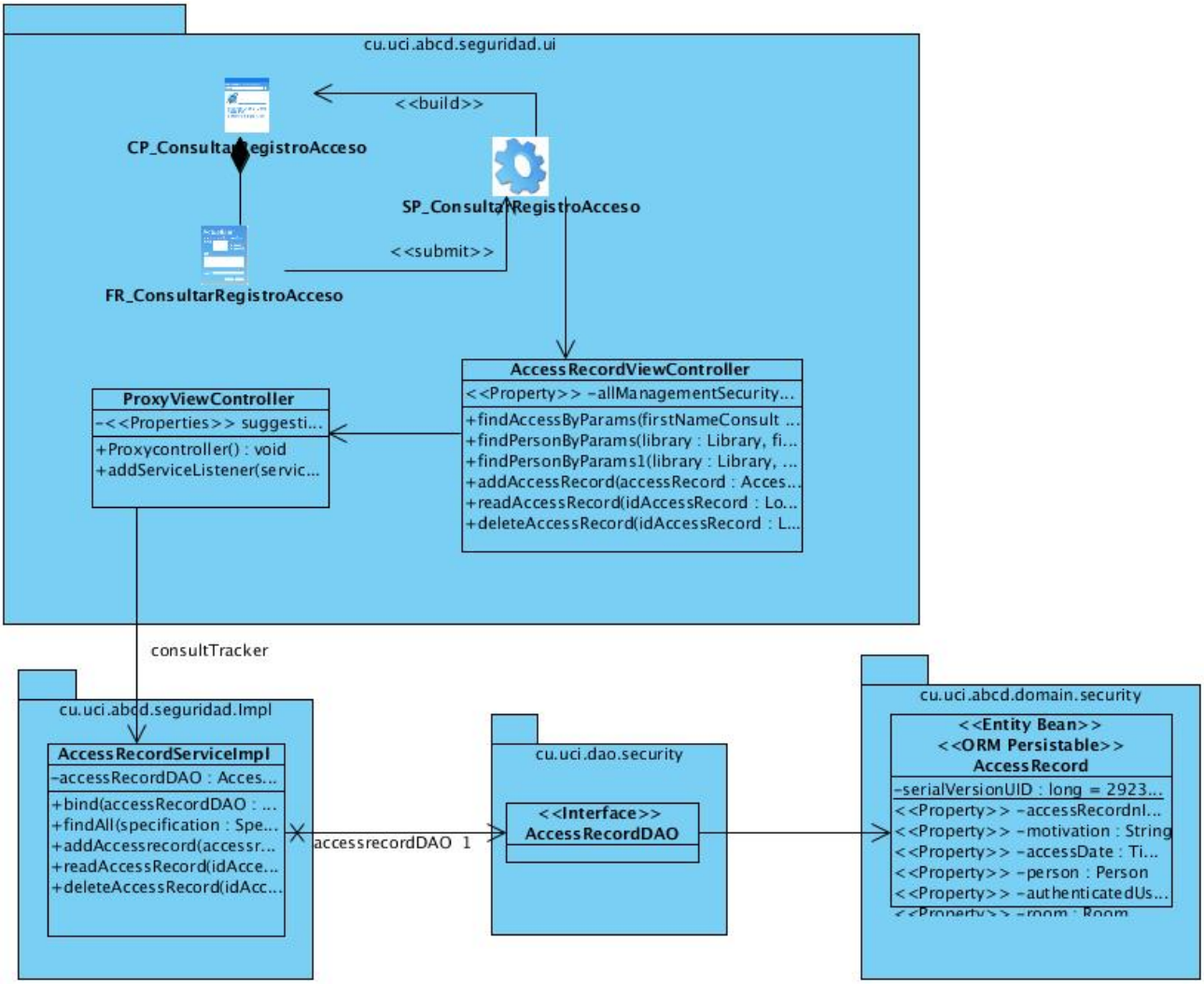


Diagrama de clase del diseño: Gestionar Perfil de Usuario

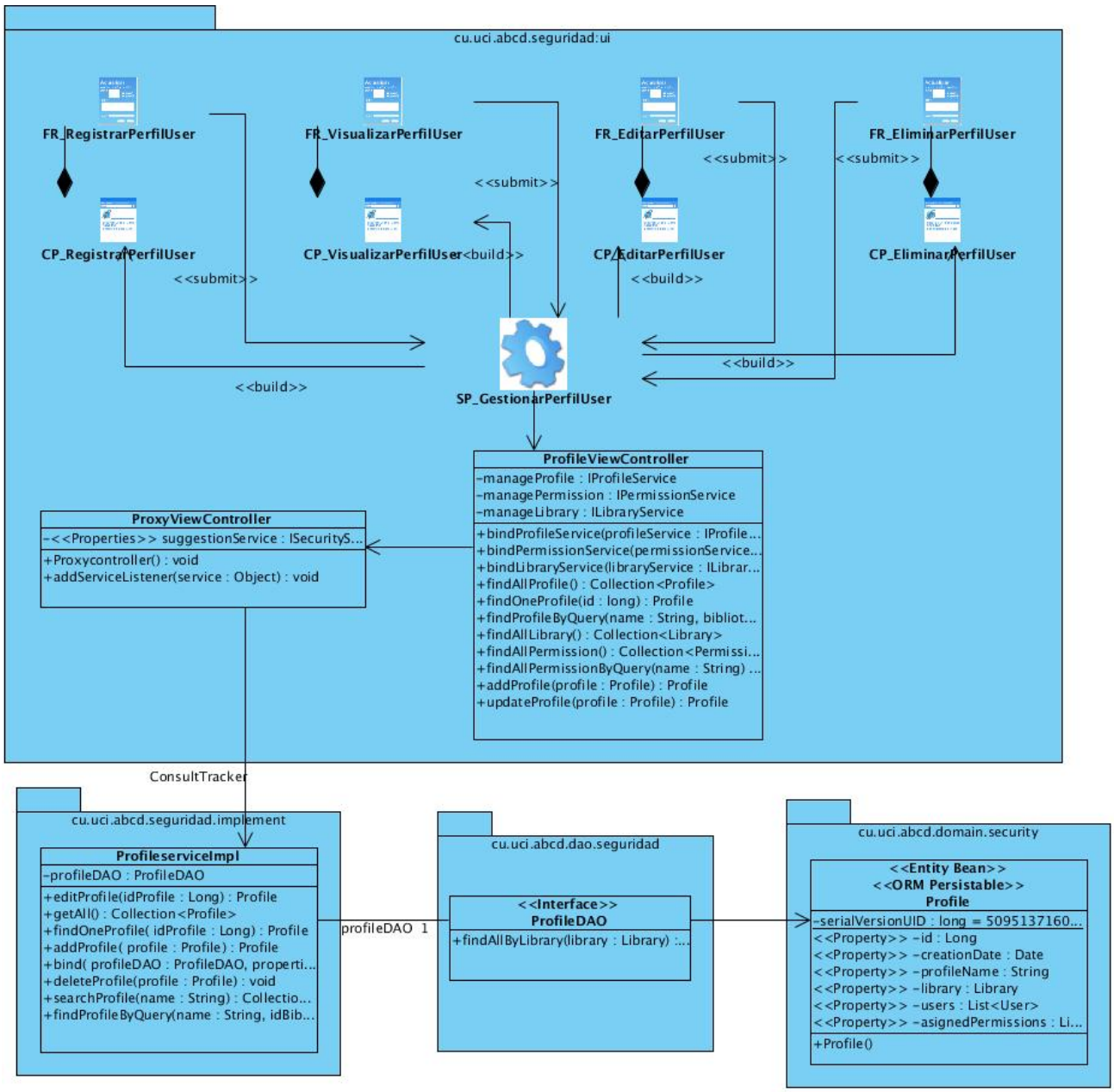


Diagrama de clases del diseño: Gestionar Usuario

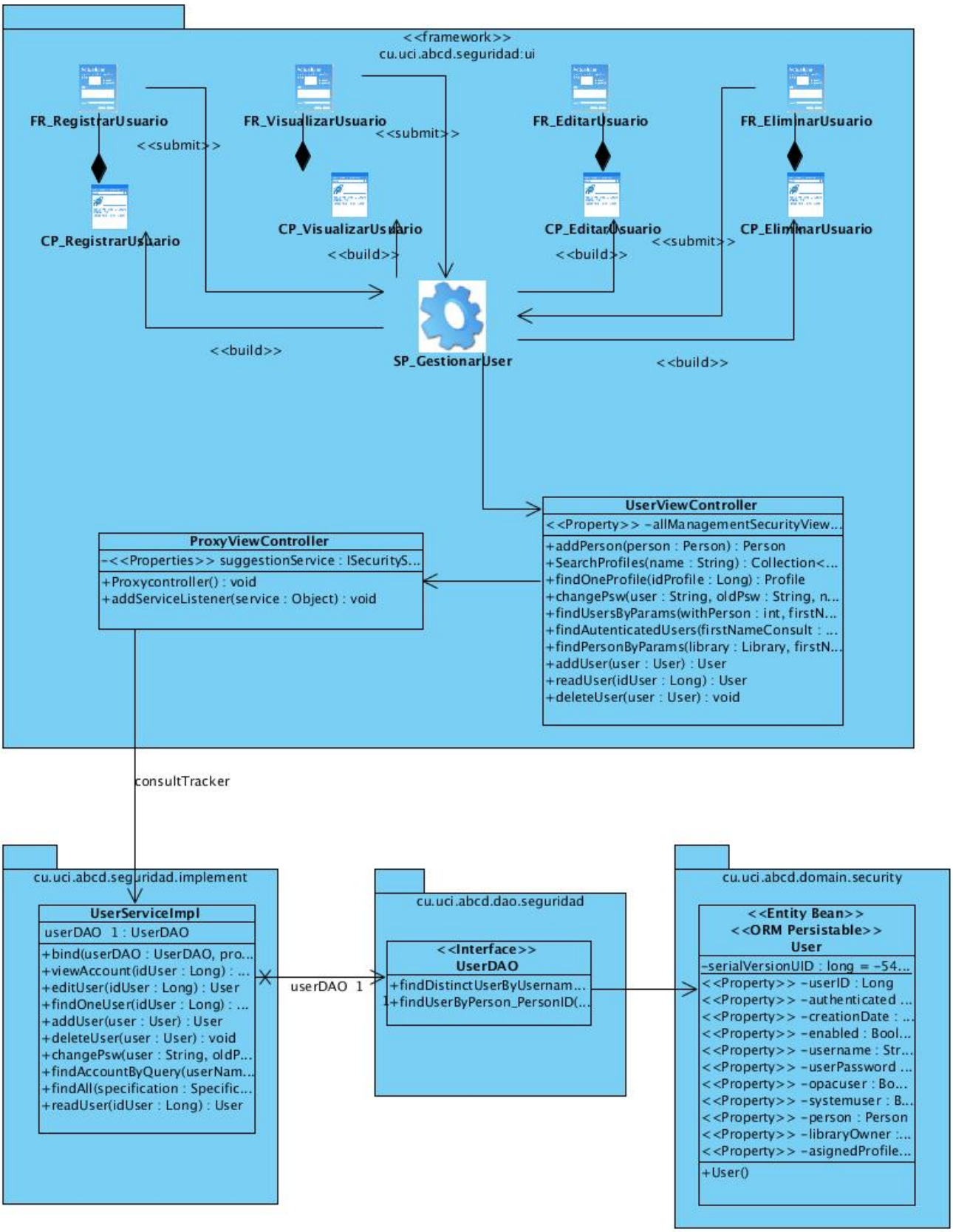

<<framework>>
cu.uci.abcd.seguridad:ui
FR_RegistrarUsuario
FR_VisualizarUsuario
FR_EditarUsuario
FR_EliminarUsuario
<<submit>>
<<submit>>
CP_RegistrarUsuario
CP_VisualizarUsuario
CP_EditarUsuario
CP_EliminarUsuario
<<build>>
<<build>>
<<submit>>
<<build>>
SP_GestionarUser
<<build>>
UserViewController
<<Property>> -allManagementSecurityView...
+addPerson(person : Person) : Person
+SearchProfiles(name : String) : Collection<...
+findOneProfile(idProfile : Long) : Profile
+changePsw(user : String, oldPsw : String, n...
+findUsersByParams(withPerson : int, firstN...
+findAutenticatedUsers(firstNameConsult : ...
+findPersonByParams(library : Library, firstN...
+addUser(user : User) : User
+readUser(idUser : Long) : User
+deleteUser(user : User) : void
ProxyViewController
-<<Properties>> suggestionService : ISecurityS...
+Proxycontroller() : void
+addServiceListener(service : Object) : void
consultTracker
cu.uci.abcd.seguridad.implement
UserServiceImpl
userDAO 1 : UserDAO
+bind(userDAO : UserDAO, pro...
+viewAccount(idUser : Long) : ...
+editUser(idUser : Long) : User
+findOneUser(idUser : Long) : ...
+addUser(user : User) : User
+deleteUser(user : User) : void
+changePsw(user : String, oldP...
+findAccountByQuery(userNam...
+findAll(specification : Specific...
+readUser(idUser : Long) : User
userDAO 1
cu.uci.abcd.dao.seguridad
<<Interface>>
UserDAO
+findDistinctUserByUsernam...
+findUserByPerson_PersonID(...
cu.uci.abcd.domain.security
<<Entity Bean>>
<<ORM Persistable>>
User
-serialVersionUID : long = -54...
<<Property>> -userID : Long
<<Property>> -authenticated ...
<<Property>> -creationDate : ...
<<Property>> -enabled : Bool...
<<Property>> -username : Str...
<<Property>> -userPassword ...
<<Property>> -opacuser : Bo...
<<Property>> -systemuser : B...
<<Property>> -person : Person
<<Property>> -libraryOwner :...
<<Property>> -asignedProfile...
+User()

Diagrama de clase del diseño: Gestionar Persona

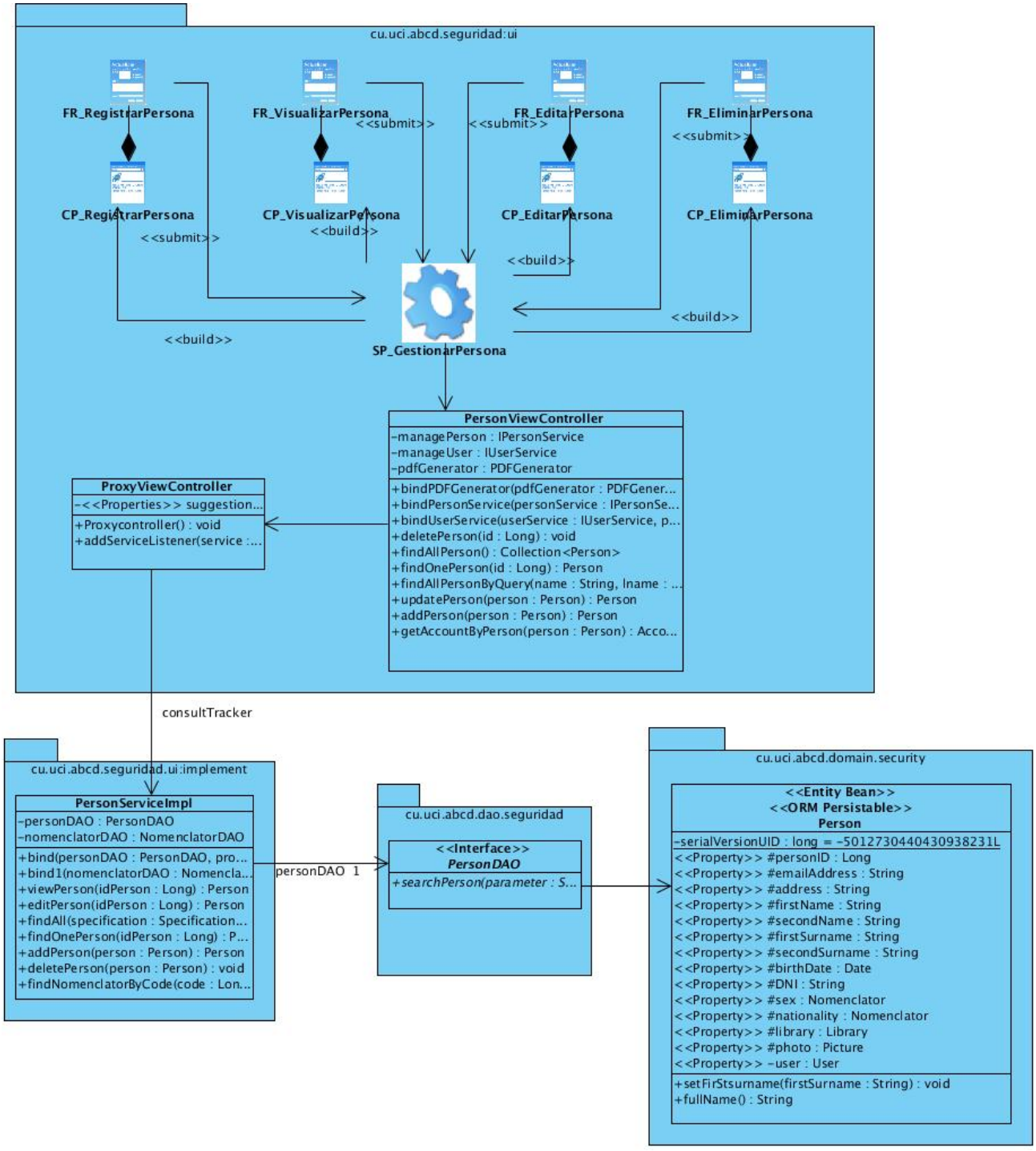
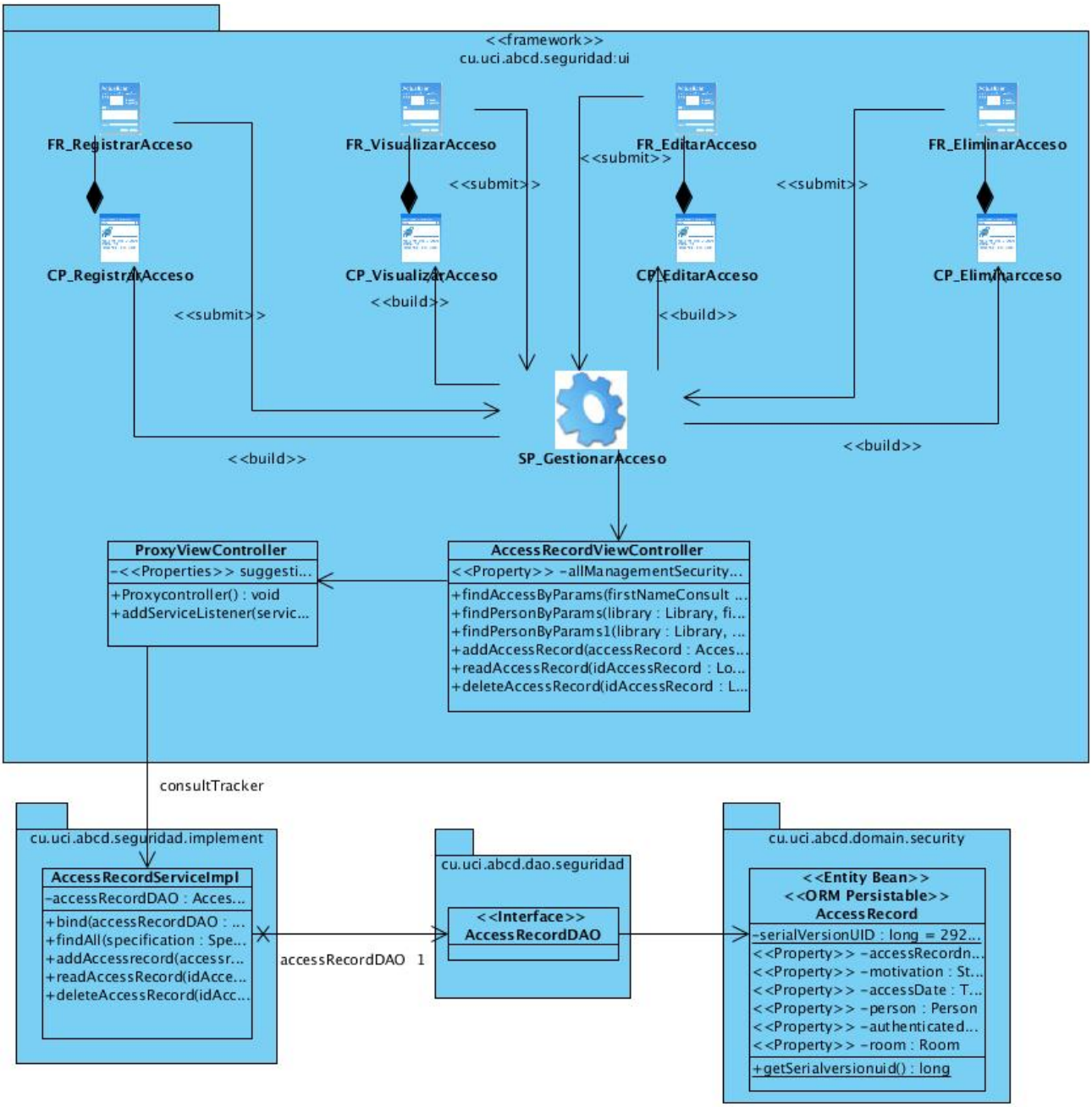


Diagrama de clase del diseño: Gestionar Acceso



Glosario

El glosario de términos estará estructurado de la siguiente manera: primero el término que se menciona en el documento, segundo su significado entre comilla simple y después el número de la página donde aparece.

- **ABCD**: ‘Automatización de Bibliotecas y Centros de Documentación’ 1.
- **API**: ‘Interfaz de Programas de Aplicación’ 18
- **CASE**: ‘Ingeniería de Software Asistida por Ordenador’ 15.
- **CIGED**: ‘Centro de Informatización de Gestión Documental’ 1**.**
- **CU**: ‘Casos de Uso’ 26.
- **DAC**: ‘Control de Acceso Discreto’ 8.
- **GOF**: ‘Grupo de Cuatro’ 33.
- **GRASP**: Patrones Generales de Software para Asignar Responsabilidades’ 32.
- **IDE**: ‘Entorno de Desarrollo Integrado’ 17.
- **JAAS**: ‘Servicio de Autenticación y Autorización de Java’ 19.
- **MAC**: ‘Control de Acceso Obligatorio’ 9.
- **OSGI**: ‘Iniciativa de Entrada para Servicios Abiertos’ 15.
- **PMB**: ‘PHPMyBiblio’ 11.
- **RAP**: ‘Plataforma Remota de Aplicación’ 20.
- **RBAC**: ‘Control de Acceso Basado en Roles’ 9.
- **RUP**: ‘Proceso Unificado de Rational’ 13.
- **SGBD**: ‘Sistema Gestor de Bases de Datos’ 17
- **SIGB**: ‘Sistemas Integrados de Gestión Bibliotecaria’ 5.
- **SQL**: ‘Lenguaje de Consultas Estructurado’ 18
- **SSL**: ‘Capa de Conexión Segura’ 19.

Glosario

- **TIC**: ‘Tecnologías de la información y las Comunicaciones’ 1.
- **UML**: ‘Lenguaje de Modelado Unificado’ 14.