Universidad de las Ciencias Informáticas

Facultad 1

Título: Sistema para el control y monitoreo del acceso en la Universidad de las Ciencias Informáticas.

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

Autores:

Jesús Camilo Gámez Díaz.

Marlen del Carmen Ramírez Díaz.

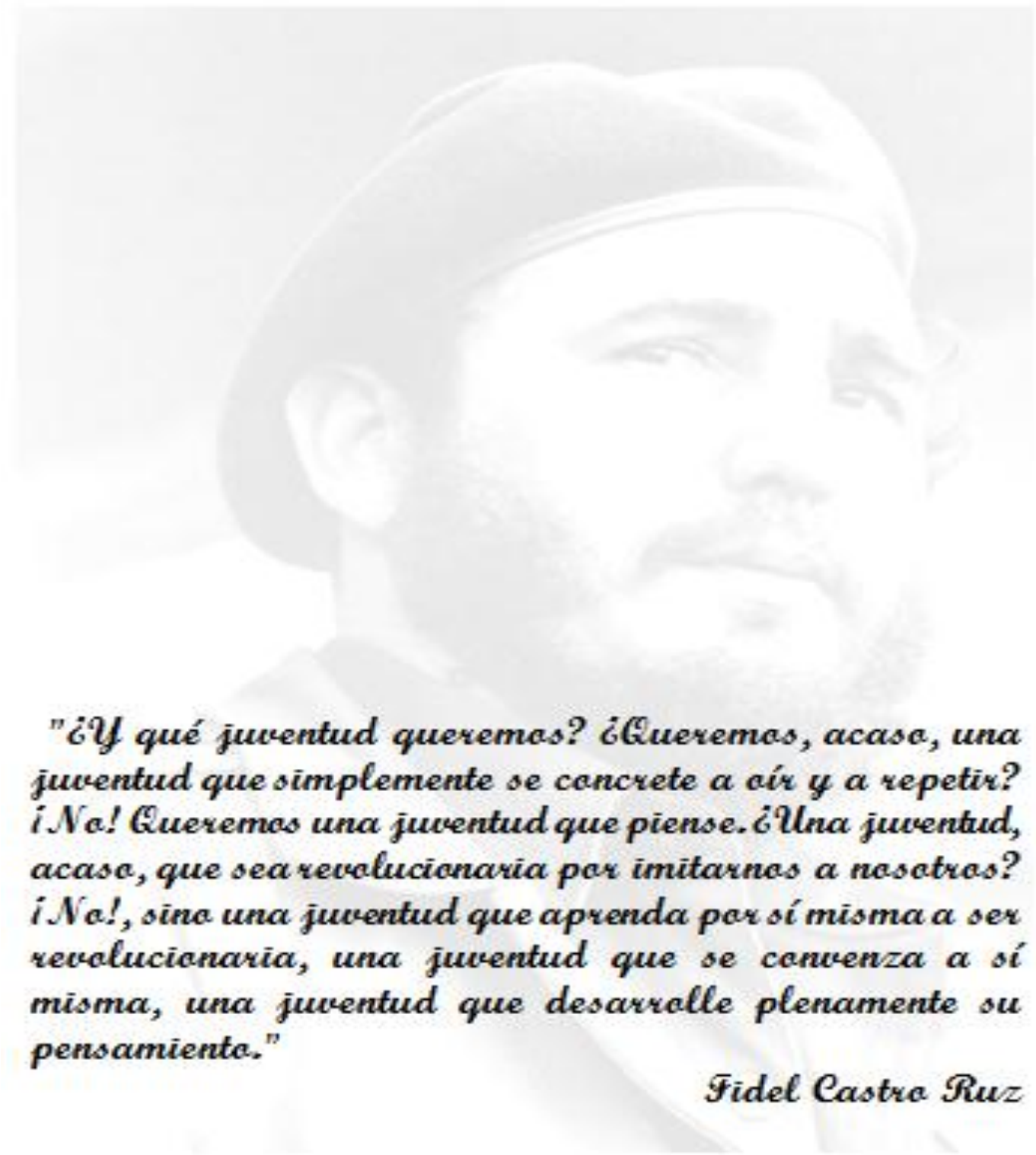
Tutores:

Ing. Rodolfo Venero Noriega.

Ing. Juan Carlos Suárez López.

La Habana, 24 de junio del 2013.

"Año 55 de la Revolución".



"¿Y qué juventud queremos? ¿Queremos, acaso, una juventud que simplemente se concrete a oír y a repetir? ¡No! Queremos una juventud que piense. ¿Una juventud, acaso, que sea revolucionaria por imitarnos a nosotros? ¡No!, sino una juventud que aprenda por sí misma a ser revolucionaria, una juventud que se convenza a sí misma, una juventud que desarrolle plenamente su pensamiento."

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: “Sistema para el control y monitoreo del acceso en la Universidad de las Ciencias Informáticas”, y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y al Centro de Identificación y Seguridad Digital (CISED) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

_____	_____
Firma del Autor	Firma del Autor
Jesús Camilo Gámez Díaz	Marlen del Carmen Ramírez Díaz
_____	_____
Firma del Tutor	Firma del Tutor
Ing. Rodolfo Venero Noriega	Ing. Juan Carlos Suárez López

DATOS DE CONTACTO

Autor: Jesús Camilo Gámez Díaz
Correo electrónico: jcgamez@estudiantes.uci.cu

Autor: Marlen del Carmen Ramírez Díaz
Correo electrónico: mcramirez@estudiantes.uci.cu

Tutor: Ing. Rodolfo Venero Noriega
Ingeniero en Ciencias Informáticas
Años de Experiencia: 3 años.
Correo electrónico: rvenero@uci.cu

Tutor: Ing. Juan Carlos Suárez López
Ingeniero en Ciencias Informáticas.
Años de Experiencia: 5 años.
Correo electrónico: jslopez@uci.cu

DEDICATORIA

Marlen

A mis padres, María Victoria y Juan Francisco por entregarme su vida y su amor desmedido, por estar pendientes de mí, y ser los guías y educadores de mi vida. Les agradezco por todo su apoyo y todo el esfuerzo que han hecho por darme lo que he necesitado y mucho más, por todo esto, mi corazón y mi respeto.

A mi Familia, que de una forma u otra han colaborado porque este momento se hiciera realidad; a mi hermano, Carlos Alberto por ser el punto hasta donde yo siempre he querido llegar y a mi abuela Eda que siempre ha creído en mí y en todo lo que he sido capaz de hacer.

A mi novio Dariel Rojas, que ha sido mi punto de apoyo todo este tiempo en la ausencia de mis progenitores, brindándome su apoyo desmedido y aguantando mis malacrianzas; a su familia, que ya es mi familia, por ser tan cariñosos conmigo.

A mi compañero de tesis Jesús Camilo, por ser tan paciente conmigo, por soportarme todo este tiempo que hemos pasado juntos, a Aliana por ayudarnos en el momento más difícil del trabajo, muchas gracias.

A mis amigos, a mis compañeros de aula que de una forma u otra me acompañaron hasta aquí, a Rocío por ser mi amiga, a Eneris que se ha comportado como mi hermana desde el momento en que nos conocimos, a los conocidos en el deporte, a todas mis amistades en los 5 años de estudios, a todos en general, gracias.

A todos los profesores de la facultad 1 que me educaron desde primero hasta quinto año; a Delly y a Héctor por ser personas incondicionales, en especial a Joel que fue como "mi padrino" y a Alién por siempre darme aliento en cada actividad que realicé y por creer en mí todo este tiempo.

Jesús Camilo

A mi abuelo que aunque ya no esté a mi lado siempre lo tengo presente. A mi mamá y mi abuela que han sido mi sustento durante estos cincos años, a mi tíos por apoyarme cada vez que hizo falta, en fin, a toda mi familia. A mi novia Aliana que ha estado conmigo en todos los momentos de la carrera y que su apoyo ha sido esencial para llegar hasta aquí. A la familia de mi novia que siempre ha estado para ayudarme. En general a todos los que de una forma u otra han estado siempre pendientes de mí en los buenos y malos momentos. Este título es también de ustedes.

AGRADECIMIENTOS

Marlen y Jesús:

Agradecemos a Dios por ser tan misericordioso con nosotros.

A la Revolución por habernos proporcionado todo lo que tenemos y darnos la posibilidad de estar en la Universidad de las Ciencias Informáticas.

A todas las personas que han colaborado con nosotros de alguna u otra manera, en especial a los integrantes del departamento de Identificación, tanto profesores como estudiantes.

A nuestros tutores, al tribunal y oponente, por la ayuda brindada. A la universidad y los profesores que nos formaron.

Muchas gracias a todos por su apoyo incondicional.

RESUMEN

El procedimiento de controlar el acceso de las personas siempre ha sido el objetivo principal de las grandes instituciones. Para ello se crearon los Sistemas de control de acceso, que brindan robustez, comodidad y ayudan a preservar la seguridad de las entidades. La Universidad de las Ciencias Informáticas (UCI) es un ejemplo de este tipo de institutos, que necesitan controlar el flujo de acceso diario del personal y los activos que acceden a la misma.

En el Centro de Identificación y Seguridad Digital (CISED) se lleva a cabo el desarrollo de una Plataforma Modular de Identificación y Control de Acceso (PMICA) con el objetivo de automatizar los procesos de solicitud, impresión y entrega de los documentos de identificación, así como la gestión y supervisión centralizada del acceso del personal a una institución y sus diferentes áreas, enmarcándose fundamentalmente en la UCI.

Con el fin de resolver los problemas existentes en la universidad, se decidió desarrollar los módulos para el control y monitoreo del acceso, los cuales deben contar con las restricciones necesarias para autorizar o denegar el acceso y mostrar las infracciones que ocurren en un tiempo determinado.

Para el desarrollo del trabajo se realizó una profunda investigación de los Sistemas de control de acceso existentes a nivel mundial y nacional, siendo estos guías para la realización de los módulos mencionados. En la presente investigación se describe la metodología, herramientas, tecnologías y artefactos diseñados en el transcurso del desarrollo de la solución.

Palabras claves: activos, artefactos, control de acceso, tecnologías.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

Índice de figuras

ÍNDICE DE TABLAS

INTRODUCCIÓN

La seguridad resulta uno de los componentes fundamentales para las instituciones, puesto que se desea la protección de las distintas áreas que las componen. En la actualidad, son cada vez más las entidades de todas partes del mundo, desde las empresas hasta un sin número de centros educacionales que manejan equipos especiales, dinero y grandes volúmenes de tecnologías, que son empleados para un mejor funcionamiento y desarrollo de su trabajo. Estas entidades necesitan limitar el flujo de personas que por ellas transitan pero se les dificulta controlar la entrada y salida del personal autorizado, puesto que cuentan con una cifra elevada de personas. Por lo que se hace necesaria la implantación de un mecanismo para restringir el acceso de las personas y llevar un registro de las mismas. Con la llegada de las Tecnologías de la Informática y las Comunicaciones (TICs), se han creado soluciones que permiten automatizar el control de acceso. Estas soluciones son los Sistemas de control de acceso que han tenido un gran auge en temas de seguridad, por lo fáciles y prácticos que son a la hora de utilizarlos, ya que controlan y monitorean el acceso de todo el personal y sus activos[1].

En la Universidad de las Ciencias Informáticas (UCI), diariamente circula un elevado número personas que pueden acceder a las distintas áreas de la universidad. La UCI cuenta con un Departamento de seguridad y protección que se encarga del cuidado de cada uno de los medios, áreas y el personal que radica en la misma. Este departamento es el encargado de realizar el proceso de controlar el acceso de las personas que intentan entrar o salir de la universidad. Dicho proceso lo realiza el agente de seguridad que procede a verificar la identificación comprobando solamente que la foto tenga similitud con la persona que solicita el acceso. Como este proceso se hace de forma manual, a través del uso de un documento de identificación, dificulta en muchas ocasiones el reconocimiento por parte de los agentes de seguridad del sujeto en cuestión, puesto que pueden presentar identificaciones plagiadas o con un nivel elevado de deterioro, o incluso, identificaciones de personal que ya esté de baja en la universidad. Por otra parte no se lleva un registro de las personas que en un momento determinado acceden a la institución, ya que solamente se registra en un libro los vehículos que entran y salen, lo que dificulta la búsqueda de información precisa para realizar un análisis ya sea de un hecho en particular o de las estadísticas de entradas y salidas de la universidad.

[1] Activos: Se refiere a vehículos, *laptop*, equipos electrodomésticos, etc.

Introducción

Teniendo en cuenta la problemática antes expuesta se define el siguiente **problema a resolver**: ¿Cómo mejorar el control y monitoreo de la entrada y salida de estudiantes, profesores, trabajadores y activos en la UCI en cuanto a efectividad?

El **objeto de estudio** de este trabajo son los procesos control y monitoreo de la entrada y salida.

Para dar respuesta al problema planteado se traza el siguiente **objetivo general**: desarrollar un sistema para el control y monitoreo de la entrada y salida de estudiantes, trabajadores, profesores y activos en la UCI.

Desglosado en los siguientes **objetivos específicos**:

- Realizar el análisis del estado del arte acerca de los Sistemas de control de acceso tanto en el ámbito nacional como internacional.
- Realizar un estudio de herramientas, tecnologías, metodología y lenguajes a utilizar en la solución.
- Realizar el análisis y diseño de la solución.
- Implementar la solución.
- Validar mediante pruebas unitarias y de caja negra la solución.

Definiéndose como **campo de acción** el control y monitoreo del acceso de estudiantes, profesores, trabajadores y activos en la Universidad de las Ciencias informáticas.

Para el desarrollo de la investigación los **métodos científicos** empleados son:

Métodos teóricos:

- **Histórico-lógico**
  Para el estudio de los antecedentes, la evolución y el desarrollo que han tenido los sistemas de control de acceso, así como para valorar las tendencias actuales de los mismos.

- **Analítico-sintético**
  Para analizar las teorías, documentos y bibliografías en general que permitieron la extracción de los principales elementos y características de los sistemas de control de acceso.

Métodos empíricos:

- **Entrevista**
  Para obtener la información necesaria relacionada con los problemas existentes en el control de acceso, en los cuales se basó la investigación.

Justificación de la investigación:

Los sistemas de control de acceso a nivel mundial se han convertido para cualquier entidad en una herramienta necesaria para mantener asegurados de una forma u otra los recursos tantos humanos como materiales. La relevancia de esta investigación se concentra en la importancia que tiene en la práctica un Sistema de control y monitoreo del acceso para la Universidad de la Ciencias Informáticas, ya que brindará un mejor control de toda la información relacionada con el acceso de las personas y activos permitiendo mejorar la seguridad y protección de la comunidad universitaria en general.

Estructuración del contenido

Capítulo 1: Fundamentación teórica: En este capítulo se puntualizan los principales conceptos relacionados al dominio del problema, se realiza el estudio del estado del arte concerniente a los Sistemas de control de acceso, tanto a nivel internacional como nacional, así como las principales tecnologías, metodologías, herramientas y lenguajes para el desarrollo de la solución propuesta.

Capítulo 2: Propuesta de solución: En este capítulo se presenta un modelo global compuesto por los procesos que informatizará el sistema y el modelo de dominio, un listado de las funcionalidades a desarrollar en la solución, un plan de desarrollo para las iteraciones a realizar. Se muestra la arquitectura definida, se describe cómo será construido el sistema a través de diversos diagramas de clases y de secuencia, se hace un análisis de los patrones de diseño utilizados en cada uno de ellos y se presenta el modelo de datos.

Capítulo 3: Implementación y pruebas del sistema: En este capítulo se hace una descripción del estilo de codificación utilizado, se muestra la distribución física del sistema entre los diferentes nodos a través del diagrama de despliegue, se presenta el diagrama de componentes que permite comprender la estructura del sistema y los resultados de las principales pruebas realizadas a la aplicación para verificar que responda a un correcto funcionamiento de acuerdo a los requerimientos establecidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se realiza una investigación sobre los conceptos fundamentales, los antecedentes y algunos ejemplos existentes en la actualidad de Sistemas de control de acceso, tanto en el ámbito internacional como nacional; así como un estudio sobre los diferentes dispositivos que se utilizan en la UCI para controlar el acceso. También se describen las herramientas, la metodología, tecnologías, plataformas, lenguajes de programación y marcos de trabajo utilizados para el desarrollo de la solución.

1.2 Conceptos fundamentales

Al profundizar en los Sistemas de control de acceso es importante destacar los diversos significados que se le acreditan a los mismos. Entre los conceptos fundamentales están las palabras “control”, “acceso”, “control de acceso” y “monitorizar”. Según el Diccionario de la Real Academia Española, “acceso” es la acción de llegar o acercarse; por su parte “control” es la regulación, manual o automática, sobre un sistema. En términos técnicos o lógicos, el “acceso” es la interacción entre un sujeto y un objeto que resulta un flujo de información de uno al otro. Asimismo el control de acceso es el proceso de conceder permisos a usuarios o grupos para acceder a diferentes lugares y recursos, tales como empresas, objetos e información; además incluye autenticar la identidad de los usuarios o grupos y autorizar el acceso a datos. Dentro de los términos esenciales del control de acceso se aprecian los léxicos “autenticación” y “autorización”. La “autenticación” es la verificación de que el usuario que trata de identificarse es válido. Usualmente se implementa con una contraseña en el momento de iniciar una sesión. Del mismo modo la “autorización” es el procedimiento para determinar si el usuario o proceso previamente identificado y autenticado tiene permitido el acceso a los recursos [1]. Entre otros de los vocablos existentes se encuentra “monitorizar”, que no es más que observar mediante aparatos especiales el curso de uno o varios parámetros fisiológicos o de otra naturaleza para detectar posibles anomalías, según el Diccionario de la Real Academia Española. En su uso más amplio, al verbo “monitorear”, se le puede acarrear el término “vigilancia”, que se refiere a la acción y al resultado de vigilar a algo o a alguien. También se le designa al sistema organizado y capacitado para llevar a cabo la acción de vigilar. De la misma manera consiste en el monitoreo del comportamiento de personas, de objetos o de procesos que se encuentran insertos dentro de un determinado sistema [2]. De acuerdo a los conceptos antes planteados los autores de la presente investigación definen a los Sistemas de control de acceso y monitoreo como: sistemas que

posibilitan autorizar y denegar la entrada y salida de las personas a una determinada entidad, además de verificar las previas acciones realizadas en cierto momento por el personal que ingresó a la misma.

1.3 Evolución y desarrollo de los Sistemas de control de acceso

La evolución es la ley que marca la supervivencia de las especies. Uno de los contextos donde cobra su mayor protagonismo lo representa el universo de la tecnología. En este ámbito, los sistemas de seguridad y extensión, llamados control de accesos están sujetos a cambios permanentes, obedeciendo los objetivos que demanda el mercado en este sector. La tecnología utilizada para el control de accesos ha crecido con los sistemas de información protegidos. De acuerdo a esto comienzan a surgir nuevos modelos de control de acceso como el MAC (control de acceso obligatorio) y DAC (control de acceso discrecional), pero estos al no cubrir las expectativas de la mayor parte de las organizaciones, son renovados. El modelo DAC es débil para controlar el acceso a los recursos de información de forma efectiva, en tanto que el MAC es rígido. A partir de los años 1980 se propone el modelo de control de accesos basado en roles RBAC, como intento de unificar los modelos clásicos DAC y MAC. El modelo RBAC trabaja con roles que establecen un nivel de indirección entre los usuarios y los derechos de acceso, a través de la asignación de roles a usuarios y de permisos y privilegios a roles. 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, representándose de forma natural la estructura de las organizaciones [3]. A partir de los avances de la tecnología, se lograron confeccionar diseños de sistemas que ofrecían mayores niveles de seguridad y que representaban un completo desafío para los intrusos. Primeramente se emplearon las puertas con cerraduras clásicas, las cuales fueron cambiadas a puertas con cerraduras electrónicas. Las puertas con cerraduras electrónicas establecieron un control de acceso efectivo, cómodo y rápido. Los usuarios del sistema de puertas electrónicas contaban con tarjetas, que se deslizaban en un lector y podían acceder a su sitio de trabajo o a información importante. Además, la labor del guardia de seguridad, con estos sistemas, se hizo complementaria, por lo que se contaba con un sistema con plan de respaldo. Al igual que con las cerraduras clásicas, las puertas electrónicas comenzaron a presentar varios problemas. Los malhechores encontraron formas de clonar o robar las tarjetas, por lo que era necesario buscar una nueva estrategia, que fuera a prueba de fraudes. Se requería un sistema que certificara la entrada y salida de las personas autorizadas, que habían ingresado a un sitio determinado, siendo imposible, que una persona sin autorización lo hiciera [4].

En la actualidad con el desarrollo acelerado de la “Era de la Informática y las Comunicaciones”, se confeccionaron nuevos sistemas, los cuales ofrecen:

- Seguridad a las instalaciones.
- Ahorro del tiempo dedicado a la gestión.
- Supervisión y monitoreo de las diferentes áreas de una instalación.

Con el uso de la informática se posibilitó el incremento de diversos sistemas de control de acceso, vinculados a numerosas tecnologías de identificación (tarjetas de proximidad, DNI electrónico[2], sistemas biométricos, etc.) y a dispositivos (tornos, molinetes, lectores de código de barras, etc.) con los que se pueden proteger instalaciones de accesos no deseados, además de la existencia de aplicaciones capaces de gestionar los permisos de los accesos de las personas, de monitorizar y dar seguimiento al personal.

1.4 Soluciones homólogas en la actualidad

A lo largo de los años han ido evolucionando los Sistemas de control de acceso, posibilitando que se garantice una mejor seguridad respecto a las áreas importantes de una instalación determinada. Estos sistemas ayudan a mejorar la gestión del personal autorizado y sus activos, aportando como beneficios, el ahorro de tiempo, la precisión, la confiabilidad y protección de los recursos asociados a la entidad.

1.4.1 Sistemas de control de acceso internacionales

A nivel mundial existe gran diversidad de Sistemas de control de accesos que utilizan disímiles *software* y *hardware* para garantizar la seguridad de las instalaciones. Entre la variedad de este tipo de sistemas se analizaron los siguientes:

Sistema de control de acceso *Easy Way*

El Sistema de control de acceso *Easy Way* creado en el Departamento de Sistemas de la empresa SCSSA[3]. Es un método seguro destinado a controlar el ingreso y egreso de personas a todas las áreas de la empresa (control personal) utilizando la huella dactilar. Además permite configurar el *hardware* desde la computadora, controlar la inclusión de los planos de la instalación donde se implanta, hasta generar informes y elaborar estadísticas. Igualmente tiene los controles de acceso totalmente integrados y en forma modular, también cubre ampliamente las necesidades habituales de distintas empresas para el control de acceso de personal y para el control de acceso a instalaciones o edificios. Asimismo es opcionalmente adaptable a sofisticados requerimientos particulares que puedan llegar a solicitarse, estableciéndose así, una relación personalizada con el cliente y su *software* de control de acceso. Trabaja

[2] Documento emitido por una autoridad oficial para permitir la identificación de la población de forma personal o virtual.
[3] Empresa argentina líder en el desarrollo en sistemas de identificación y de control de acceso.

de forma autónoma tanto para la abertura de las puertas como para accionar barreras, alarmas y otros dispositivos. Estos son adaptables a cualquier sistema de lectura [5].

Software de Control Personal *Wapa*

El *software Wapa* fue creado por la empresa *DokkoGroup*[4]. Es un *software* rápido, práctico y completo para obtener toda la información que se necesita sobre cualquier empleado. Este facilita el monitoreo de las entradas y salidas del personal de la empresa de forma sencilla y segura mediante la utilización de tarjetas o llaveros con códigos únicos e irrepetibles, además utiliza unidades lectoras por radio-frecuencia que permiten que el personal quede registrado con sólo aproximar la tarjeta a la unidad central, evitando de esta manera errores de lectura causados por desgastes o malos usos. Conjuntamente tiene como beneficio el acceso al sistema mediante una computadora con conexión a *Internet*[5] desde cualquier parte del mundo, permitiendo así que se pueda monitorear y administrar el personal de la empresa estando fuera de ésta [6].

Software biométrico EP 300 de la empresa *INBIOSYS*

El sistema EP 300 creado para el control de la entrada y salida de los empleados de la empresa *INBIOSYS*[6] utiliza tecnología biométrica para la autenticación del personal correspondiente a la misma. Al mismo tiempo es un equipo que tiene capacidad de 2000 huellas digitales, almacenando hasta 50000 registros. Cuenta con una pantalla LCD (pantalla delgada de cristal líquido), un teclado, varios puertos USB (estándar industrial desarrollado vinculado con dispositivos electrónicos) y una red que se gestiona mediante el protocolo TCP[7]/IP[8]. Además posee un control de tiempos y asistencia que se puede instalar en las oficinas para un mejor control de los empleados, conjuntamente tiene un completo programa en español que le permite configurar los horarios de los trabajadores, emitir reportes y exportar la información en archivos planos a otros programas de nómina [7].

Software biométrico VF 30 de la empresa *INBIOSYS*

El sistema VF 30 controla el acceso de personas, permite la apertura de puertas y el control de tiempos-asistencias con la huella digital. Fue desarrollado para la seguridad de pequeñas y medianas empresas. Al

[4] Empresa argentina que ofrece diversidades de productos tecnológicos.
[5] Conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP.
[6] Empresa colombiana experta en Ingeniería y elaborar sistemas biométricos.
[7] TCP (por sus siglas en inglés Transmission Control Protocol): Uno de los protocolos fundamentales en *Internet* y garantiza que los datos lleguen a su destino sin errores y en el mismo orden que se transmitieron.
TCP/IP: Familia de protocolos de *Internet*. Conjunto de protocolos de red en los que se basa *Internet* y que permiten la transmisión de datos entre computadoras.

mismo tiempo realiza la integración de la identificación dactilar, de los sistemas RFID[9], las alarmas, los tiempos de atención y las funciones de control de acceso con una elegante apariencia y calidad confiable. Posee una alarma musical de alta calidad, pantalla en varios idiomas, interfaz de usuario amigable y comunicación adecuada para la gestión de datos en diferentes entornos. Incluye un *software* funcional de gestión, compatible con varios tipos de base de datos y con la zona horaria [7].

Software AccPro

Sistema desarrollado por la empresa *LARCON-SIA*[10].Este moderno *software* es una herramienta para tener el control centralizado del estado de todos los accesos de una empresa. Se pueden visualizar en una pantalla, basada en un plano real de una misma planta o de diferentes sucursales todos los movimientos y alarmas en tiempo real; con la posibilidad de activar el video digital de cada acceso para que el personal de vigilancia pueda observar lo que ocurre y actuar de forma inmediata. También integra un módulo de biometría que permite registrar las huellas de usuarios y visitas, transmitiéndolas por la red a cada una de las controladoras de acceso. De la misma forma es fácil de usar, brinda facilidades para el monitoreo, permite compatibilizar diferentes tecnologías de identificación como: huella dactilar, tarjetas de proximidad, tarjetas inteligentes, entre otros. Permite ejecutar múltiples servicios en forma simultánea (en uno o varios servidores) de manera que puede obtener el máximo rendimiento y el menor tiempo de respuesta en todas las comunicaciones con los terminales de fichaje [8].

Control de Acceso *Easy* (AEC)

El sistema *AEC* fue creado por la empresa *Bosch*[11].El mismo brinda la posibilidad de controlar todos los puntos de entrada para otorgar o denegar el acceso a edificios enteros, oficinas individuales, áreas de estacionamiento, ascensores, zonas restringidas y otras áreas específicas. Además es fácil de instalar y utilizar; asimismo integra una serie de características de seguridad, tales como: los CCTV (circuitos cerrados de televisión), video verificación y control de intrusión sin dejar de ser un sistema simple. Igualmente es sencillo de conectar, dándole la posibilidad a cualquier cliente de administrar la base de datos, monitorear la actividad que ocurre en las instalaciones o modificar los dispositivos conectados [9].

RFID (por sus siglas en inglés Radio *Frequency IDentification*): Sistema de almacenamiento y recuperación de datos remotos que usa dispositivos denominados etiquetas, tarjetas, transpondedores o tags RFID.

[10] Empresa Argentina fabricante de la línea de productos ClockCard. Dedicada a brindar soluciones en las áreas de la identificación y el control de acceso**.**

[11] Empresa que diseña soluciones de alta calidad de seguridad.

1.4.2 Sistemas de control de acceso nacionales

Cuba en los últimos años se ha visto inmersa en el tema y el uso de Sistemas de control de acceso a partir de la necesidad que existe de garantizar la seguridad de las instituciones. El desarrollo de las TICs también es uno de los principales objetivos de la nación, logrando así un auge en la construcción de *software*, donde han surgido numerosas entidades dedicadas a la actividad de incrementar la creación de tales sistemas. Un ejemplo de estas es la empresa *DATYS*[12], que produce bienes y servicios informáticos. La misma ha desplegado aplicaciones propias que son utilizadas con el fin de controlar el acceso al personal de las instituciones, como son:

XymaSafeAccess

Sistema de identificación y control de acceso físico, formado por una red coordinada de tarjetas de identificación, lectores electrónicos, bases de datos especializadas, *software* y computadoras diseñadas para monitorear y controlar el tráfico a través de puntos de acceso. También posee 4 módulos y ofrece variantes de configuración en dependencia del nivel de seguridad que se quiera implementar. Del mismo modo ofrece un proceso de registro seguro, que identifica a cada individuo y determina que la persona está autorizada para utilizar los privilegios o servicios que este brinda. Además dispone de procedimientos para emitir tarjetas de identidad con seguridad, garantizando que los documentos de identidad sean formulados solamente por la entidad autorizada; para expedir dichas tarjetas y que solo sean emitidos documentos de identidad para las personas correctas. De igual forma ofrece soluciones para monitorear el uso de la identificación y la autenticación. La autenticación implementa una cadena de confianza previamente establecida, verificando la identidad de los portadores del identificador y la legitimidad de las tarjetas de identidad y las credenciales [10].

XymaSafeVision

Es un sistema de video protección profesional basado en la tecnología IP[13]. El mismo se integra a los diversos equipamientos de cómputo, videocámaras y servidores para cámaras analógicas. También tiene incorporado algoritmos de reconocimiento de patrones que permiten identificar y verificar las matrículas de los autos, definir perímetros para la detección de movimiento o cantidad de objetos, entre otras. Contiene una gran capacidad de manejo de usuarios y cámaras, puede realizar grabaciones de forma manual, programada, continua, por detección de movimiento o por eventos de alarmas predefinidos. Este admite interactuar con los servicios de mapas y planos de las plantas de las instalaciones en apoyo a la

[12]Empresa cubana dedicada a la tecnología y sistemas.

[13] Dispositivo que permite realizar una comunicación utilizando una red IP, ya sea mediante red de área local o a través de *Internet.*

administración y utilización del sistema. Del mismo modo posee herramientas para la gestión del ancho de banda de la red y la calidad de las imágenes a través de la personalización de parámetros de video de cada canal y para cada función [10].

Biomesys

Sistema que aprovecha las bondades de las tecnologías biométricas, para registrar los eventos de asistencia en una organización por medio de la identificación de los empleados y de la autenticación de la identidad mediante un sensor biométrico de huellas dactilares. Asimismo con la captura de identificaciones biométricas únicas, se convierte en un generador de datos altamente confiable por su bajo o casi nulo nivel de vulnerabilidad por la suplantación de identidad. Entre sus diferentes características distintivas se encuentra: el no establecimiento de limitaciones de implementación asociadas al tipo de organización; puede integrar de forma rápida diferentes medios de autenticación como escáneres biométricos, credenciales de bandas magnéticas, tarjetas de códigos de barras y proximidad. También posee un módulo de captura biométrica para el control de las entradas y las salidas del personal [11].

Frontpas

Es una solución integral para la gestión y control de la frontera. El sistema integra el registro, control y vigilancia de pasos fronterizos legales trabajando con estándares de seguridad y calidad internacionales. El mismo provee una plataforma tecnológica que facilita el control y gestión del tránsito de personas en las fronteras legales y los flujos en condiciones de estricta seguridad, permitiendo contrarrestar según el interés nacional, el tráfico ilegal de personas, mercancías y combustibles. Gestiona el proceso de chequeo migratorio en general desde cualquier punto fronterizo; captura y utiliza la información multibiométrica (rostro y huella) para detectar y evitar la suplantación de identidad, además verifica la identidad de la persona contra listas de control (donde se registran aspectos o rasgos de una persona). Permite además el chequeo de listas de pasajeros de vuelos (API[14]) previo a la llegada de los viajeros. Propicia trabajar en un régimen conectado o desconectado del núcleo central, facilitando gran confiabilidad a la solución. También admite configurar los puestos de trabajo, los dispositivos de captura, los roles, usuarios del sistema, las actividades del chequeo, las listas de control propias, los nomencladores y el comportamiento de procesos del sistema. Ayuda a alertar en tiempo real a las autoridades correspondientes a partir de reglas pre-definidas [11].

[14] API(por sus siglas en inglés Advanced Passenger Information System): Intercambio electrónico de datos.

1.4.3 Sistemas de control de acceso en la UCI

En la Universidad de las Ciencias Informáticas se cuenta actualmente con diferentes sistemas de control de acceso en varias áreas, proporcionando una mejor seguridad en las instalaciones. Dentro de los sistemas informatizados relacionados con el control del personal a las áreas limitadas, se encuentran los siguientes:

Sistema de control de acceso a comedores (CONTACC)

Este sistema apoya el proceso de gestión de control de acceso de los estudiantes, trabajadores y profesores a los comedores de los diferentes complejos alimenticios durante las tres sesiones de servicio: desayuno, almuerzo y comida. El mismo se divide en dos partes: el control de acceso y la gestión de comensales. El acceso en cada una de las puertas de los comedores se controla registrando el código de barras, a través de la identificación de cada persona, garantizando que esto tenga lugar una sola vez y por la puerta correspondiente. La gestión de comensales permite a los directivos la asignación correspondiente a cada uno de los comedores y puertas. Actualiza la información del personal que pasa por cada puerta, brindando reportes sobre las que tuvieron acceso, haciendo seguro y confiable el proceso de identificar a dichas personas. Posee una interfaz amigable, se adapta con facilidad a variaciones en la estructura del flujo informativo y permite configurar los tiempos de sincronización de forma manual o automática cada determinado momento [12].

Sistema único de control de acceso para personas vinculadas a la producción en el centro CISED de la Universidad de las Ciencias Informáticas

Los sistemas que controlan el acceso de las personas a los laboratorios destinados a los proyectos productivos, comprueban mediante un documento de identificación que los datos del personal estén en la base de datos correspondiente. Estas identificaciones son verificadas mediante un lector de código de barras, el cual es usado en la universidad. Una de las características más importantes de este sistema es que cada usuario tiene asignado un ip y un solapín, al autenticarse el sistema reconoce que la persona que se ha logueado tiene un ip conocido en el centro [13].

1.5 Análisis de las soluciones homólogas

Con el estudio de los sistemas a nivel nacional e internacional antes mencionados se trazan las siguientes conclusiones:

- Los sistemas *Easy Way*, EP 300, VF 30, *AccPro*, *Biomesys* y el *Frontpas* no son utilizados porque los mismos emplean de una forma u otra tecnología biométrica, además algunos de ellos son

adaptables a cualquier sistema de lectura. Por otra parte el *Easy Way* trabaja de forma autónoma, el EP 300 posee un control de tiempos y asistencias para diferentes oficinas, el VF 30 es compatible con varios tipos de base de datos y con las zonas horarias. Asimismo el *AccPro* ejecuta varios servicios de forma simultánea en uno o varios servidores, el *Biomesys* es un generador de datos altamente confiable y el *Frontpas* trabaja en un régimen conectado y desconectado del núcleo central y encargado para la labor diaria en las fronteras.

- Igualmente el *Wapa* y el *XymaSafeAccess* controlan el personal a través de tarjetas de identidad. Además el *Wapa* monitorea a las personas a través de llaveros únicos e irrepetibles utilizando unidades lectoras por radio-frecuencia y administra el sistema fuera de la empresa con solo tener una computadora conectada a la red.
- De esta manera el *XymaSafeVision* y el AEC integran una serie de características como video de protección profesional, CCTV y video verificación. Del mismo modo el *XymaSafeVision* incluye algoritmos de reconocimiento de patrones, interactúa con servicios de mapas y planos de las instalaciones y realiza grabaciones de forma manual, programadas y continuas por detección de movimiento o por eventos de alarmas predefinidos. También el AEC da la posibilidad de que cualquier cliente administre la base de datos y monitoree la actividad que ocurre o modifique los dispositivos conectados.
- De la misma forma CONTACC y el Sistema único de control de acceso para personas vinculadas a la producción se aplican en áreas específicas (comedores y laboratorios) y usan como dispositivo de identificación el lector de código de barras. El CONTACC además brinda reportes haciendo uso seguro y confiable del proceso de identificar las personas y configura los tiempos de sincronización de forma manual o automática. El Sistema único de control de acceso para personas vinculadas a la producción no cuenta con base de datos centralizada con todos los datos referentes a todos los laboratorios de producción.
- Además los sistemas a nivel internacional antes mencionados no concuerdan con la tecnología que actualmente se aplica en la UCI. También brindan una mayor robustez y adaptabilidad ya que es tecnología de punta la cual es costosa e implica diversas inversiones económicas para implantarlas en la universidad y el país. Los mismos necesitan requerimientos de *hardware* muy avanzados para poder funcionar de forma rápida y eficiente, etc.
- También los sistemas a nivel nacional no concuerdan con la tecnología que actualmente se aplica en la UCI. Además estos sistemas están confeccionados para lugares específicos.

- Los sistemas creados en la UCI no resuelven los problemas que se presentan de identidades activas (baja a estudiantes y trabajadores, graduados) que pertenecen a personas que ya no residen en la entidad accediendo con estas sin ser previamente verificadas y se encuentran en áreas específicas.

Beneficios que aportan a la investigación las soluciones homólogas

Las soluciones estudiadas anteriormente brindan diversos beneficios a la presente investigación como: la incorporación de otros dispositivos de identificación. También garantizan la seguridad de los empleados e información de valor, posibilitan la administración a través de una misma interfaz de usuario y controlan y monitorean el acceso de personas por cada punto de control. Además realizan una o varias tareas representadas a través de módulos y proporcionan el aumento de los requerimientos del sistema.

1.6 Tendencias actuales de los Sistemas de control de acceso

Con la necesidad de reforzar toda seguridad existente en cada local, las tecnologías de *hardware* y *software* se han hecho indispensables para la identificación y autenticación de las personas, convirtiéndose en un recurso importante y fácil de usar para la protección. En el proceso de controlar el acceso de las personas se integran diferentes dispositivos que ayudan a hacer resistente la seguridad de las áreas, ejemplo de estos son: las cerraduras eléctricas, torniquetes electrónicos, lectores de código de barras, los sistemas biométricos y otros. Los mismos son ayudados por un conjunto de técnicas para la identificación, como: tarjetas de radio frecuencia o proximidad, tarjetas de banda magnética, tarjetas de código de barras, entre otras; que proporcionan facilidad y ahorro de tiempo en el momento de acceder a cualquier entidad. Las técnicas anteriormente mencionadas pueden combinarse entre sí, para posibilitar que el nivel de seguridad sea alto. Además es necesario para llevar a cabo el monitoreo del personal que accede a un área determinada, la existencia de métodos tecnológicos que se complementan y ayudan a observar las distintas acciones de los individuos. Ejemplos de estos métodos son: escuchas secretas, escuchas telefónicas, micro cámaras, rastreo GPS, implementación de un circuito cerrado de televisión y otros que facilitan también el nivel de vigilancia tanto en un nivel público como privado. Estas tecnologías aunque son útiles también tienen características, ventajas y desventajas que permiten a las personas escoger la más adecuada para implantarlas en el lugar deseado.

1.7 Tecnologías utilizadas en la solución

Para el desarrollo de la solución, fue seleccionado el código de barras 39, por ser una de las simbologías de código de barras popular y conocida a nivel mundial. También se encuentra entre los códigos que

comúnmente se utilizan para la gestión de inventarios y es un tipo de código que reconoce prácticamente todos los tipos de escáneres. La ventaja de estos códigos reside en que son capaces de codificar información numérica y además letras mayúsculas y algunos signos de puntuación. Igualmente pueden representar códigos de longitud variable [14].

Figura 1. 1 Código de barras 39.

Figura 1. 2 Lector de código de barras *Voyager.*

El dispositivo para la lectura del código de barras antes mencionado es el lector *Voyager* MS9520. El mismo se ha concebido como un lector rápido, con gran profundidad de campo y una velocidad de lectura, siendo casi el doble de la de la serie MS 900.Combina las prestaciones del MS 951 y el MS 961 dentro de un mismo escáner para ofrecer mayor flexibilidad al usuario. El *Voyager* puede operar en el modo “manos libres” cuando se sitúa sobre su soporte. Se necesita solamente la presentación del código, para que el lector realice automáticamente la lectura del documento de identificación. También es programable para lecturas de corto o largo alcance tanto en el modo automático como manual, con lo que incrementa su confiabilidad y productividad [15].

1.8 Ambiente de desarrollo

Para la realización de la aplicación propuesta el proyecto de Plataforma Modular de Identificación y Control de Acceso (PMICA) del departamento de Identificación del centro CISED definió el uso de diferentes tecnologías, herramientas, metodología y otros recursos que son necesarios para el completo desarrollo de la solución propuesta. La utilización de las mismas facilita un rápido aprendizaje y manejo.

1.8.1 Metodología de desarrollo

***Features Driven Development* (FDD)**

Es un enfoque ágil para el desarrollo de sistemas. Se enfoca en iteraciones cortas que entregan funcionalidades tangibles. Fue desarrollada por Jeff De Luca y Peter Coad. Dicho enfoque no hace énfasis en la obtención de los requerimientos, sino en cómo se realizan las fases de diseño y construcción. Sin embargo, fue diseñado para trabajar con otras actividades de desarrollo de *software* y no requiere la utilización de ningún modelo de proceso específico. Además hace alusión en aspectos de calidad durante todo el proceso de desarrollo e incluye un monitoreo permanente del avance del proyecto. FDD afirma ser conveniente para el desarrollo de sistemas críticos [16]. La misma presenta cinco fases secuenciales que son mostradas a continuación en la Figura 1.3 [17].

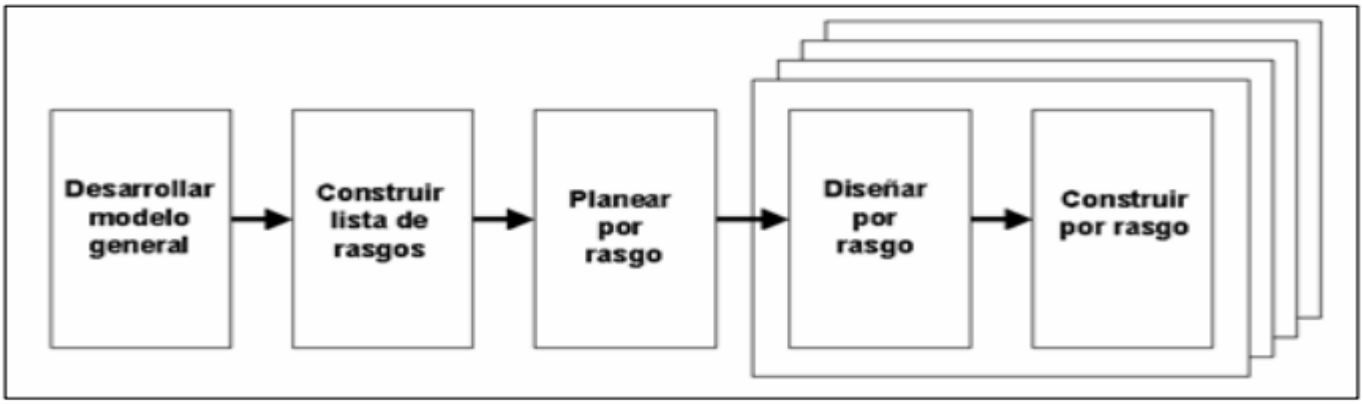


Figura 1. 3 Fases de FDD.

Desarrollar un modelo general:

Construcción del esqueleto del primer modelo del sistema por parte de los expertos del dominio y los desarrolladores del sistema teniendo previamente una idea del contexto y los requerimientos del sistema. Este modelo de dominio global es dividido en distintas áreas y a cada subgrupo se le asigna un área de dominio a desarrollar. Finalmente el equipo de desarrollo discute y decide cual es el modelo de objetos apropiado para cada área del dominio y posteriormente se construye el modelo global en base a todas las alternativas.

Construir lista de rasgos:

El equipo identifica las características, las agrupa, las prioriza y las pondera. Además se divide en subgrupos especializados en áreas relacionadas al dominio. Se confecciona la lista de características la cual es revisada por los usuarios y *sponsors*[15] del sistema para su validación y aprobación.

Planear por rasgo:

A partir de la etapa anterior, se toma la lista de características y se ordena en base a la prioridad y a la dependencia de cada característica, conjuntamente se establecen hitos y se diseña un cronograma de diseño y construcción. También las clases identificadas en la primera etapa son asignadas a cada programador.

Diseñar por rasgo:

Identificación de las clases involucradas, realización del diagrama de secuencia correspondiente y elaboración de las descripciones de las clases y sus métodos.

Construir por rasgo:

Construcción de los métodos para cada una de las características, confección de las pruebas unitarias para cada una de las clases, inspección del código y la realización de las pruebas de integración [18].

1.8.2 Lenguaje de modelado

UML 8.0

UML (*Unified Modeling Language*) es un lenguaje de modelado para visualizar, especificar, construir y documentar los artefactos de un sistema. Proporciona una forma estándar de escribir los planos de un sistema, tales como funciones del sistema, clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de *software* reutilizables. Es la sucesión de una serie de métodos de análisis y diseño orientados a objetos. Está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requisitos y estructuras necesarias para plasmar un sistema de *software* previo al proceso intensivo de escribir código [19]. Este lenguaje presenta varias ventajas y desventajas entre ellas están:

Ventajas:

- Es un lenguaje consolidado.
- Estándar de facto.
- Fácil de aprender.

[15] Patrocinadores o auspiciantes corporativos, personas o empresas que colaboran económicamente con el evento con fines publicitarios.

- Permite una comunicación fluida entre los diversos actores acerca del modelo [20].

Desventajas:

- UML no ha sido diseñado para modelar procesos de negocios, por lo que no está orientado a lo que necesita el experto en el dominio del negocio.
- Predispone un enfoque orientado a objetos lo que puede contradecir un enfoque orientado al negocio.
- UML está más orientado a los arquitectos de sistemas y diseñadores de *software*. Está pensado para un público eminentemente técnico [20].

BPMN

BPMN (*Business Process Modelling Notation*) es un estándar de la BPMI (*Business Process Management Initiative*). Abarca únicamente los procesos de negocio, lo que significa que otro tipo de modelos relacionados (estructura de la organización, recursos, modelos de datos, estrategias, reglas de negocio, etc.) quedan fuera de la especificación [21]. Define un diagrama de procesos de negocio que está basado en la técnica de diagramas de flujo y adaptado para crear modelos gráficos de las operaciones de los procesos de la organización. Está compuesto de un conjunto de elementos gráficos que facilitan el desarrollo de un solo diagrama [22].

1.8.3 Herramienta de modelado

Visual Paradigm for UML 8.0

Visual Paradigm for UML es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML [23].

1.8.4 Acceso a datos

Gestor de Base de Datos *PostgreSQL* 9.1

PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto. Soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Es altamente escalable, tanto en la enorme cantidad de datos que puede manejar y en el número de usuarios concurrentes que puede administrar. Incluye una biblioteca de funciones estándar con cientos de funciones integradas que

van desde las operaciones matemáticas básicas, operaciones con *string* (cadena de caracteres) para criptografía y compatibilidad con Oracle [24]. *PostgreSQL* 9.1 es la última actualización de este sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD[16] [25]. Esta nueva edición trae bastantes novedades como son:

- **Replicación sincrónica:** permitiendo alta disponibilidad con consistencia sobre múltiples servidores.
- **Regionalización por columna:** soportando correctamente el ordenamiento por lenguaje en las bases de datos, tablas o columnas.
- **Tablas *unlogged*[17]*:*** importante incremento del rendimiento para datos efímeros [26].

NHibernate 3.1

NHibernate es un asignador relacional de objetos (ORM), herramienta o marco. Traduce el lenguaje basado en objetos con un lenguaje que entiende la base de datos, es decir, genera las sentencias SQL necesarias para poder insertar, actualizar, eliminar y cargar datos [27]. Tiene como objetivo ser una solución completa al problema de la gestión de los datos persistentes cuando se trabaja con bases de datos relacionales y clases de modelo de dominio. Se esfuerza por llevar a cabo la ardua labor de mediación entre la aplicación y la base de datos [28].

1.8.5 *Frameworks*

***Microsoft .NET Framework* 4.0**

.NET Framework 4.0 proporciona nuevas mejoras y características que lo hacen diferente a sus versiones anteriores, aunque posee la capacidad de funcionar en paralelo con ellas. Además es un componente integral de *Windows* que admite la compilación y la ejecución de las aplicaciones y servicios *web.* Proporciona un entorno de ejecución administrado, un desarrollo e implementación simplificada y la integración con una gran variedad de lenguajes de programación. *.NET Framework* 4.0 incluye un modelo de seguridad mejorado. Este *framework* posee características y mejoras en la implementación como:

- *Client Profile.*
- Ejecución en paralelo en el mismo proceso.
- Biblioteca de clases portable.
- Diagnósticos y rendimiento.
- Globalización.

[16] BSD (por sus siglas en ingles *Berkeley Software Distribution*): licencia de *software* otorgada principalmente para los sistemas BSD.
[17] Tablas especiales que mejoran el desempeño, son más rápidas al escribir datos.

- Recolección de elementos no utilizados.
- Contratos de código.
- Ensamblados de interoperabilidad en tiempo de diseño [29].

ASP.NET MVC 4

ASP.NET MVC es un marco de desarrollo *web* de *Microsoft* que combina la eficacia y la pulcritud de la arquitectura Modelo-Vista-Controlador (MVC), las técnicas de desarrollo ágil y las mejores partes de la plataforma ASP.NET existente [30].

El MVC es un principio de diseño arquitectónico que separa los componentes de una aplicación *web*. Esta separación le da más control sobre las partes individuales de la aplicación, lo que le permite desarrollarlas con mayor facilidad, modificarlas y probarlas. Este implementa una variante moderna del MVC que es especialmente adecuado para aplicaciones *web.*

El *framework* MVC incluye los siguientes componentes:

- **Modelo:** Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- **Vista:** Maneja la visualización de la información.
- **Controlador:** Se encargan de la interacción del usuario, trabajar con el modelo y además seleccionar una vista para hacer que se muestre en la interfaz de usuario. En una aplicación MVC, la vista sólo muestra la información, el controlador es el encargado de responder a la entrada del usuario y la interacción [31].

ASP.NET MVC 4 es un marco de trabajo para crear aplicaciones *web* escalables y basadas en estándares mediante patrones de diseño. Incluye ASP.NET *Web* API el cual es un marco para la creación de servicios HTTP que pueden llegar a una amplia gama de clientes entre ellos navegadores y dispositivos móviles. El mismo presenta varias características principales entre ellas están:

- ASP.NET *Web* API.
- Plantillas de proyecto predeterminadas, renovadas y modernizadas.
- Nueva plantilla de proyecto móvil.
- Soporte de aplicaciones móviles.
- Soporte mejorado para métodos asincrónicos [32].

***JQuery framework* 1.9.1**

JQuery es un *framework* de *JavaScript* que sirve como base para la programación avanzada de aplicaciones aportando una serie de funciones o códigos para la ejecución de tareas habituales. Ofrece una infraestructura de mayor facilidad para la creación de aplicaciones complejas del lado del cliente. De la misma forma es un producto serio, estable, bien documentado, empleado para implementar interfaces de usuario, galerías, con una buena aceptación por parte de los programadores y un grado de penetración en el mercado muy amplio [33]. Del mismo modo facilita la interacción entre un documento HTML, es decir, entre *JavaScript* y DOM[18]. Simplifica el recorrido HTML[19] y manipula documentos, maneja eventos del navegador, animaciones de DOM, interacciones con AJAX, etc [34].

1.8.6 Lenguaje de programación *CSharp*

Microsoft Visual C# es potente lenguaje orientado a componentes de *Microsoft*. C# tiene un papel importante en la arquitectura de la plataforma *Microsoft. NET Framework*, está dirigido principalmente a los desarrolladores para crear aplicaciones mediante el uso de la plataforma *Microsoft. NET Framework.* Hereda muchas de las mejores características de C++ y *Microsoft Visual Basic* [35]. Las numerosas innovaciones en C# permiten el desarrollo rápido de aplicaciones, manteniendo la expresividad y elegancia de los lenguajes estilo C. El mismo presenta varias características fundamentales, entre ellas están:

Sencillez: Elimina elementos que otros lenguajes incluyen y que son innecesarios en *.NET.*

Modernidad: Incorpora al lenguaje elementos que son útiles para el desarrollo de aplicaciones.

Seguridad de tipos: Incluye mecanismos que permiten asegurar que los accesos a tipos de datos se realicen correctamente para evitar que se produzcan errores.

Instrucciones seguras: Evita errores comunes, imponiendo una serie de instrucciones en el uso de las expresiones más usadas (==, =, goto o break, etc.).

Versionable: Incluye una política de versionado que permite crear versiones de tipos [36].

1.8.7 Entorno de desarrollo

***Microsoft Visual Studio* 2010**

Microsoft Visual Studio no es más que un entorno de desarrollo integrado (IDE) para *Windows*. *Visual Studio* es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones *web*

[18] DOM (en sus siglas en inglés *Document Object Model*): Modelo de objetos para la representación de documentos.

[19] HTML (por sus siglas en inglés *Hyper Text Markup Language*): lenguaje de marcado predominante para la elaboración de páginas *web* que se utiliza para describir y traducir la estructura y la información en forma de texto.

ASP.NET, servicios *web*, aplicaciones de escritorio y aplicaciones móviles *Visual Basic* , *Visual* C# y *Visual* C++ utilizan todos el mismo entorno de desarrollo integrado, que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones del *framework .NET*, las cuales ofrecen acceso a tecnologías claves para simplificar el desarrollo de aplicaciones *web* con ASP y servicios *web* [37]. *Visual Studio* 2010 incluye mejoras que aceleran la creación de código y que simplifican la implementación, asimismo incorpora fragmentos de código para controles HTML, ASP.NET y *JavaScript*. También puede empaquetar y publicar aplicaciones *web* con un clic. Conjuntamente ofrece una alternativa para la creación de aplicaciones de *SharePoint*, enlace de datos de tipo arrastrar y colocar para aplicaciones de *Windows Presentation Foundation* (WPF)[20] y mayor compatibilidad con el desarrollo dirigido por pruebas [38].

1.8.8 Tecnologías

JavaScript

Se utiliza principalmente para crear páginas *web* dinámicas e interactivas. Técnicamente *JavaScript* es un lenguaje de programación interpretado, por lo que no es necesario compilar el código para poder ejecutarlo. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador *web* sin necesidad de procesos intermedios. Reduce la cantidad de transacciones que se efectúan a través del protocolo HTML y las posibilidades de que se genere un error durante la inserción de datos. También puede leer y escribir *cookies*[21], una operación que hasta hace poco únicamente podía desarrollar el servidor *web* [39].

CSS 3

CSS (*Cascading Style Sheets* u Hojas de Estilo en Cascada) es la tecnología desarrollada por el *World Wide Web Consortium* (W3C), lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML[22]. CSS es la mejor forma de separar los contenidos y su presentación siendo imprescindible para la creación de páginas *web* complejas. La versión utilizada para la presente solución es CSS3, última de la especificación CSS. El término CSS3 es una referencia a las nuevas características de CSS, pero en el tercer nivel en el progreso de la especificación CSS. Contiene casi todas las características de las versiones anteriores y también agrega

[20] Permite el desarrollo de interfaces de interacción en *Windows* tomando características de aplicaciones de *Windows* y de aplicaciones *web.*
[21] Pequeña información enviada por un sitio *web* y almacenada en el navegador del usuario, de manera que el sitio *web* puede consultar la actividad previa del usuario.
[22] XHTML (por sus siglas del inglés *eXtensible Hypertext Markup Language*): Es HTML expresado como XML válido.

nuevas características para ayudar a los desarrolladores a resolver una serie de problemas sin necesidad de marcado no semántico, *scripting*[23] complejo o imágenes adicionales [40].

AJAX

Es un acrónimo de *Asynchronous JavaScript* + XML[24], que se puede traducir como "*JavaScript* asíncrono + XML". La misma está formada por varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes". Las tecnologías que forman AJAX se muestran a continuación en la Figura 1.4:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT[25] y JSON[26], para el intercambio y la manipulación de información.
- *XMLHttpRequest*[27], para el intercambio asíncrono de información.
- *JavaScript*, para unir todas las demás tecnologías [41].

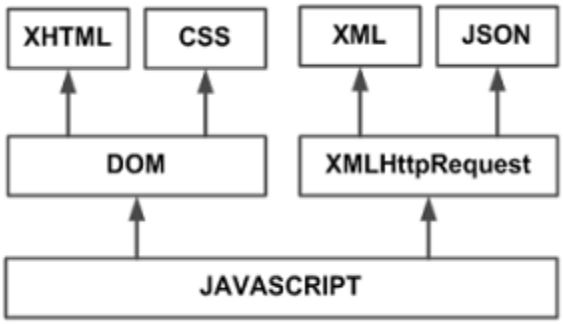


Figura 1. 4 Tecnologías agrupadas bajo el concepto de AJAX.

Por otra parte para desarrollar aplicaciones con AJAX se requiere de un conocimiento avanzado de todas las tecnologías antes mencionadas. Igualmente permite mejorar completamente la interacción del usuario con la aplicación, para evitar recargas de la página, mediante la confección de un elemento intermedio entre el usuario y el servidor el cual agiliza la respuesta de la aplicación [41].

[23] Guión, archivo de órdenes o archivo de procesamiento por lotes.
[24] XML (por sus siglas en inglés *eXtensible Markup Language*): Lenguaje de marcas extensible.
[25] Transformaciones XSL, es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros.
[26] JSON (por sus siglas en inglés *JavaScript Object Notation*): Formato ligero para el intercambio de datos.
[27] XMLHttpRequest (por sus siglas en inglés Extensible Markup Language o *Hypertext Transfer Protocol*): Es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores *web*.

1.8.9 Plataforma de desarrollo *Microsoft .NET*

Microsoft .NET es una plataforma de desarrollo y ejecución de aplicaciones, que facilita el proceso de construcción de programas multipropósito, tanto en entornos cliente, entornos *web* o móviles y está construido sobre una arquitectura abierta. Incluye una amplia gama de productos creados para trabajar con los estándares de XML e *Internet Microsoft*. La misma facilita la rápida puesta en marcha de servicios y soluciones basados en *Internet*, brindando una mejor atención a los usuarios, facilidad de instalación y despliegue de programas de *software.* Fue diseñada explícitamente para dar paso al rápido desarrollo, integración y orquestación de cualquier grupo de servicios y aplicaciones *web* en una solución. La plataforma *Microsoft .NET* propicia una gama de servicios *web* y proporciona un entorno abierto de alta productividad diseñado específicamente para negocios avanzados en *Internet* [42].

1.9 Conclusiones

En el presente capítulo a partir de los conceptos fundamentales se llegó a una mejor comprensión del tema que se trata en la investigación. Con el estudio de los sistemas de control de acceso tanto a nivel nacional o internacional se consiguió identificar las principales características que estos tienen en común y como se realizan los procesos de controlar y monitorear el acceso de manera general. Concluyéndose que no existe ninguno que pueda ser empleado en la solución. Como ambiente de desarrollo se definió por el proyecto la metodología, herramientas tecnologías y lenguajes para la implementación de la solución, cumpliendo con los requisitos determinados y facilitando la creación de un producto con las condiciones y la calidad deseada.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 Introducción

En el presente capítulo se confecciona la propuesta de solución al problema planteado, para ello se realiza un análisis crítico de la situación actual en la UCI. Posteriormente se estudian los procesos dentro del negocio, realizando las mejoras y el modelado de los mismos. A partir de estos modelos se logran precisar las actividades que se informatizarán, las cuales se agrupan y se clasifican, definiéndose un cronograma eficiente para el diseño e implementación de cada una. Asimismo se describen las funcionalidades más importantes y se muestran los requerimientos no funcionales. Se incorpora la vista lógica de la arquitectura, donde quedan reflejados los componentes del proceso y cómo interactúan entre sí. Finalmente se construyen los diagramas de clases y de secuencia y se efectúa un estudio de los patrones de diseño utilizados en cada uno de ellos y se realiza el modelo de datos.

2.2 Análisis crítico de la situación actual

Desde los inicios de la UCI (2002) se ha tenido la necesidad de controlar la entrada y salida del personal, por el flujo de acceso tan grande que se presenta diariamente. Con el objetivo de conceder permisos de acceso a los usuarios autorizados, se aplica un control de acceso de forma manual, siendo supervisada esta actividad por la Oficina de control de acceso. La ejecución del proceso anteriormente mencionado cuenta con diferentes irregularidades que se han ido reflejando en el transcurso de los años debido al número de personas que ya no residen en esta (bajas, graduados) y otras que acceden sin autorizo. Existen distintas acciones que infringen en el funcionamiento del proceso actual y afectan la seguridad, dentro de las trabas se pueden citar:

- Pérdida o deterioro de los documentos emitidos para conceder el acceso de las personas.
- Chequeo del documento de identificación del personal por aproximación visual, provocando suplantación de identidad y la realización de hechos delictivos (robos).
- Dificultad para conocer datos específicos de las personas que realizan alguna infracción.
- Limitación para monitorear el acceso y las infracciones de personas y los activos, ya que de la forma que se ejecuta no se tiene un registro de las mismas.
- Complicación para obtener datos estadísticos de los accesos e infracciones realizados en cierto período.

Por los problemas anteriormente expuestos se realizarán los módulos de Control de acceso y Monitoreo del sistema PMICA, permitiendo responder a los requerimientos necesarios para mejorar las presentes ineficiencias.

2.3 Propuesta de solución

Modelo de proceso

El modelo de procesos de negocio es una abstracción de cómo funciona un negocio y sus detalles, difieren según la perspectiva de la persona que crea el modelo [43]. Después de analizar los procesos actuales para controlar el acceso a la universidad se realiza una propuesta de solución. A continuación se muestra en la Figura 2.1 la modelación de los procesos de control de acceso y en el (ANEXO I) el monitoreo del acceso, para lograr un mejor entendimiento del negocio.

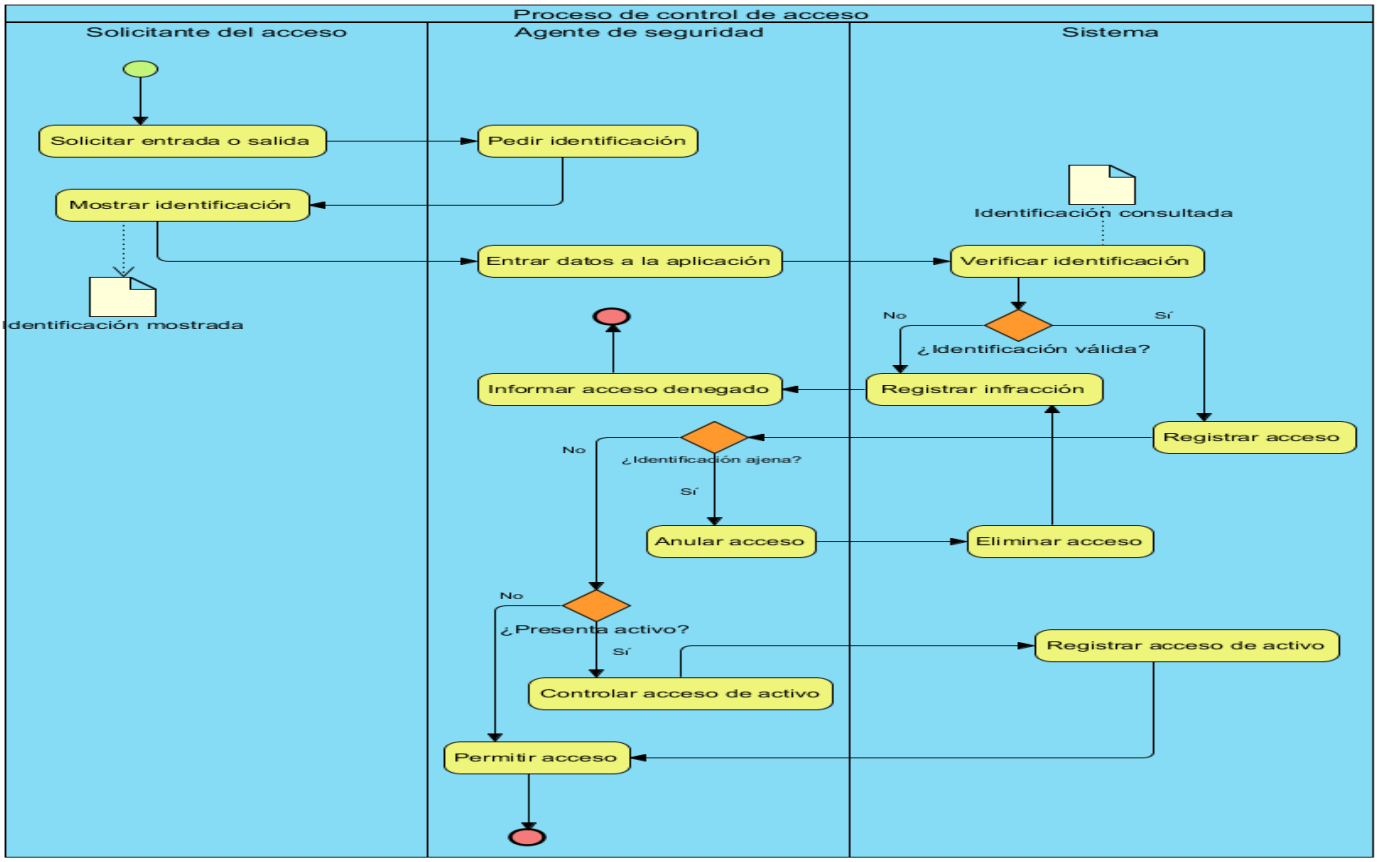


Figura 2. 1 Modelo de proceso de negocio "Control de acceso".

Descripción de las actividades del proceso "Control de acceso"

Para obtener una mejor comprensión de forma clara y detallada del flujo de acciones que se llevan a cabo para el control de acceso de entrada y salida de las personas, se realiza la descripción de cada una de las actividades que conforman el proceso. A continuación se evidencian cada una de estas descripciones en la tabla 2.1 y las actividades del proceso de monitoreo en el (ANEXO II).

Actividad	Descripción	Responsable	Entrada	Salida
Solicitar entrada o salida.	El Solicitante le pide al Agente de seguridad el autorizo para acceder.	Solicitante.	Solicitar acceso.	Acceso solicitado.
Pedir identificación.	El Agente de seguridad exige la presentación de la identificación.	Agente de seguridad.	Exigir identificación.	Identificación exigida.
Mostrar identificación.	El Solicitante presenta la identificación.	Solicitante.	Mostrar identificación.	Identificación presentada.
Entrar datos a la aplicación.	El Agente de seguridad introduce los datos de la identificación en la aplicación.	Agente de seguridad.	Introducir datos en la aplicación.	Datos ingresados al sistema.
Verificar identificación.	El Sistema verifica si la persona solicitante tiene acceso o no al área en cuestión.	Sistema.	Datos ingresados al sistema.	Verificación realizada.
Registrar infracción.	En caso que acurra una infracción el Sistema la registra	Sistema.	Verificación realizada.	Infracción creada.
Informar acceso denegado.	El Agente de seguridad informa el acceso denegado al área en cuestión.	Agente de seguridad.	Verificación realizada.	Información de denegación.
Registrar acceso.	En caso que acurra un acceso el Sistema lo registra.	Sistema.	Verificación realizada.	Acceso creado.
Anular Acceso.	El Agente de seguridad anula el acceso de la persona en caso que la identificación presentada no sea de su propiedad.	Agente de seguridad.	Verificación realizada.	Acceso anulado.
Eliminar acceso.	El Sistema elimina el acceso.	Sistema.	Datos del acceso.	Acceso eliminado.
Permitir acceso.	El Agente de seguridad permite el acceso.	Agente de seguridad.	Verificación realizada.	Entrada o salida del Solicitante.
Controlar acceso de activo.	En caso que el solicitante posea un activo el Agente de seguridad procede a controlar el acceso del mismo.	Agente de seguridad.	Introducir datos del activo.	Datos del activo ingresados en el sistema.
Registrar acceso de activo.	El Sistema registra el acceso del activo.	Sistema.	Datos del activo ingresados en el sistema.	Acceso del activo registrado.

Tabla 2. 1 Descripción de las actividades del proceso "Control de acceso".

Modelo del dominio

El modelo de dominio es una representación de conceptos en el dominio del problema. Según el lenguaje de modelado de sistemas de *software* UML el modelo de dominio se ilustra como un grupo de diagramas de estructura estática donde no se define ninguna operación [44]. Para un mayor conocimiento por parte del equipo de desarrollo quedan plasmados los conceptos fundamentales y una representación de sus relaciones en la Figura 2.2.

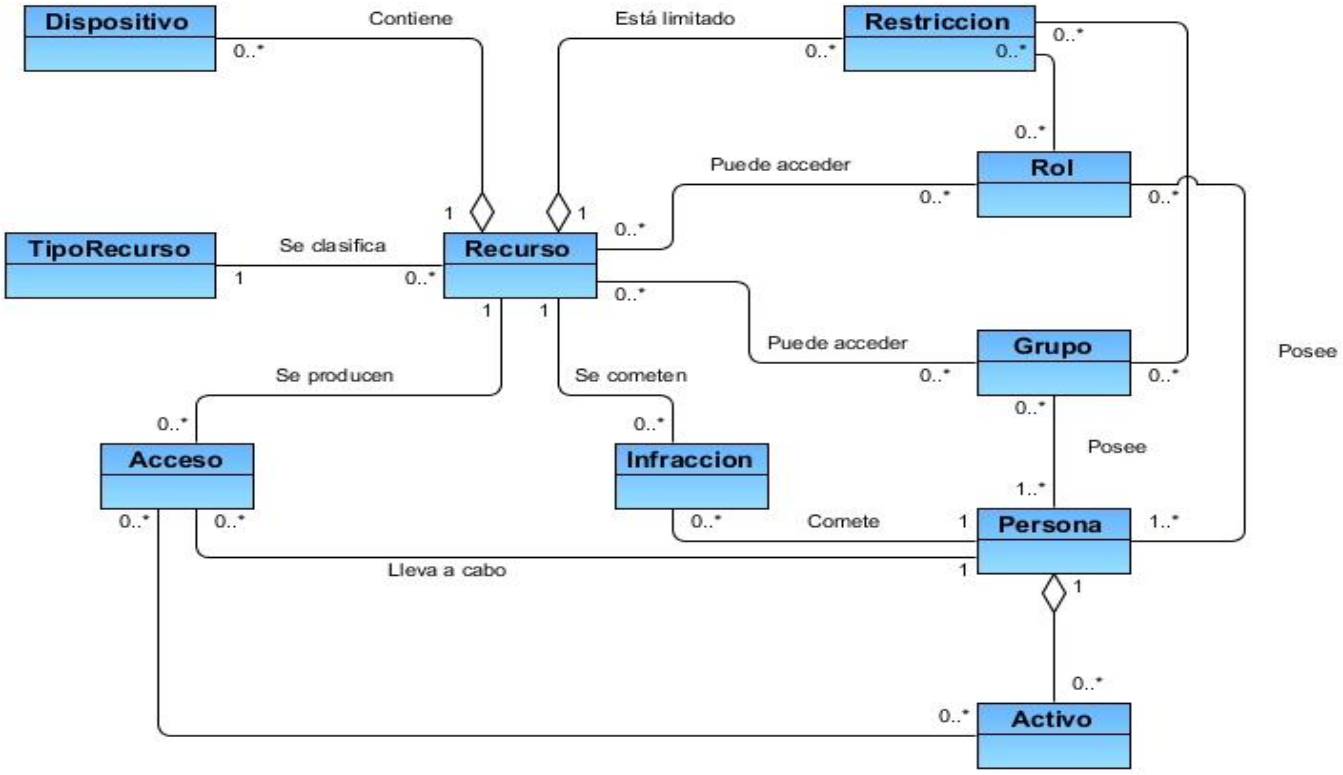


Figura 2. 2 Modelo de dominio.

Principales conceptos asociados a la solución

- **Persona:** Representa cualquier individuo que forme parte de la organización o esté vinculado a la misma (estudiantes, profesores y trabajadores).
- **Activo:** Medios que puede poseer una persona en el momento de realizar un acceso.
- **Recurso:** Instalaciones en las que se estructura una organización y se les requiere controlar el acceso.
- **Tipo de Recurso:** Nomencladores en los que se clasifican los recursos de una organización.

- **Atributo:** Parámetros adicionales que requiera un tipo de recurso o persona para ayudar en su descripción.
- **Dispositivo:** Elementos de *hardware* utilizados en el control de acceso que están presentes en los recursos de la organización.
- **Grupo:** Representan una estructura lógica en la que se encuentra organizado el personal. Los grupos pueden ser subgrupos de un grupo superior.
- **Rol:** Representa las funciones que puede llevar a cabo el personal de la organización. Los roles pueden heredar de otros roles.
- **Restricción:** Representan las reglas que podrán imponerse a los recursos para controlar las condiciones en que los roles y/o grupos pueden acceder a estos. Definen las circunstancias en las que se debe producir un acceso.
- **Acceso:** Representa la evidencia de que una persona determinada entró o salió de cierto local.
- **Infracción:** Representa la evidencia de que ocurrió un problema en el momento del acceso.

Reglas del negocio

Las reglas de negocio son una declaración completa y atómica que permite ser expresada de forma inteligible[28] y que al juntarse con las demás reglas, conforman el marco estructural, la política, la estrategia y la operativa de una empresa u organización siendo una parte importante para el desarrollo de los sistemas [45]. Las mismas se clasifican en reglas textuales, reglas del modelo de datos, reglas de relación y reglas de derivación. Para mayor detalle sobre las reglas del negocio de la solución dirigirse al (ANEXO III).

2.4 Especificación de los requisitos de *software*

Los requerimientos del sistema establecen con detalles las funciones, servicios y restricciones operativas del sistema [46]. El término, "especificación de requisitos del *software*" se refiere típicamente a la producción de un documento, o a su equivalente electrónico, que puede estar sistemáticamente repasado, evaluado, y aprobado [47]. La especificación de requisitos de *software* proporciona una base informada para transferir un producto de *software* a los nuevos usuarios o a las máquinas nuevas [48].

Lista de características

La metodología FDD en la fase número 2 especifica la realización de un listado de funcionalidades o características. Del mismo modo luego del análisis del modelo de proceso de negocio fueron definidas las

[28] Inteligible: Comprendido o entendido.

actividades que conforman el objeto de informatización, o sea, la base para definir los requisitos funcionales de la solución propuesta. Las mismas se agruparon por "conjunto de características" y en la tabla 2.2 se listan las características de la solución propuesta que serán informatizadas.

Conjunto de características	Características
Controlar acceso de personas.	C1.Registrar la entrada de personas.
	C2.Registrar la salida de personas.
	C3.Registrar infracción.
	C4.Anular acceso.
Controlar acceso de activos.	C5.Registrar acceso de activo.
	C6. Eliminar acceso de activo.
	C7.Adicionar un nuevo activo.
Monitorear el acceso de personas.	C8. Mostrar datos de accesos de personas en un rango de fecha.
	C9.Mostrar datos de accesos de personas por un punto de control en un rango de fecha.
	C10. Mostrar datos de accesos de una persona en un rango de fecha.
	C11.Mostrar datos de accesos de una persona por un punto de control en un rango de fecha.
	C12. Ver detalles de accesos.
Monitorear el acceso de activos.	C13.Mostrar datos de accesos de activos en un rango de fecha.
	C14. Mostrar datos de accesos de activos por un punto de control en un rango de fecha.
	C15. Mostrar datos de accesos de un activo de una persona en un rango de fecha.
	C16.Mostrar datos de accesos de un activo por un punto de control en un rango de fecha.
Monitorear las infracciones.	C17. Mostrar infracciones en un rango de fecha.
	C18.Mostrar infracciones de una persona en un rango de fecha.
	C19. Mostrar infracciones por un punto de control en un rango de fecha.
	C20.Mostrar infracciones de una persona por un punto de control en un rango de fecha.
	C21. Ver detalles de infracciones.

Monitorear estadísticas gráficas.	C22.Mostrar gráfica con cantidad de accesos e infracciones en el día por intervalos de horas.
	C23.Mostrar gráfica con cantidad de accesos e infracciones en el año por meses.
	C24. Mostrar gráfica con cantidad de accesos e infracciones en un rango de fecha.
	C25. Mostrar gráfica con cantidad de accesos e infracciones de una persona en un rango de fecha.
	C26. Mostrar gráfica con cantidad de accesos e infracciones por un punto de control en un rango de fecha.
	C27. Mostrar gráfica con cantidad de accesos e infracciones de una persona por un punto de control en un rango de fecha.

Tabla 2. 2 Listado de características.

Descripción de las características

Los requerimientos funcionales (características) son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares y también pueden declarar explícitamente lo que el sistema no debe hacer [49]. A continuación se muestra la descripción de la funcionalidad “Registrar acceso de personas” en la tabla 2.3, las demás descripciones verlas en el (0).

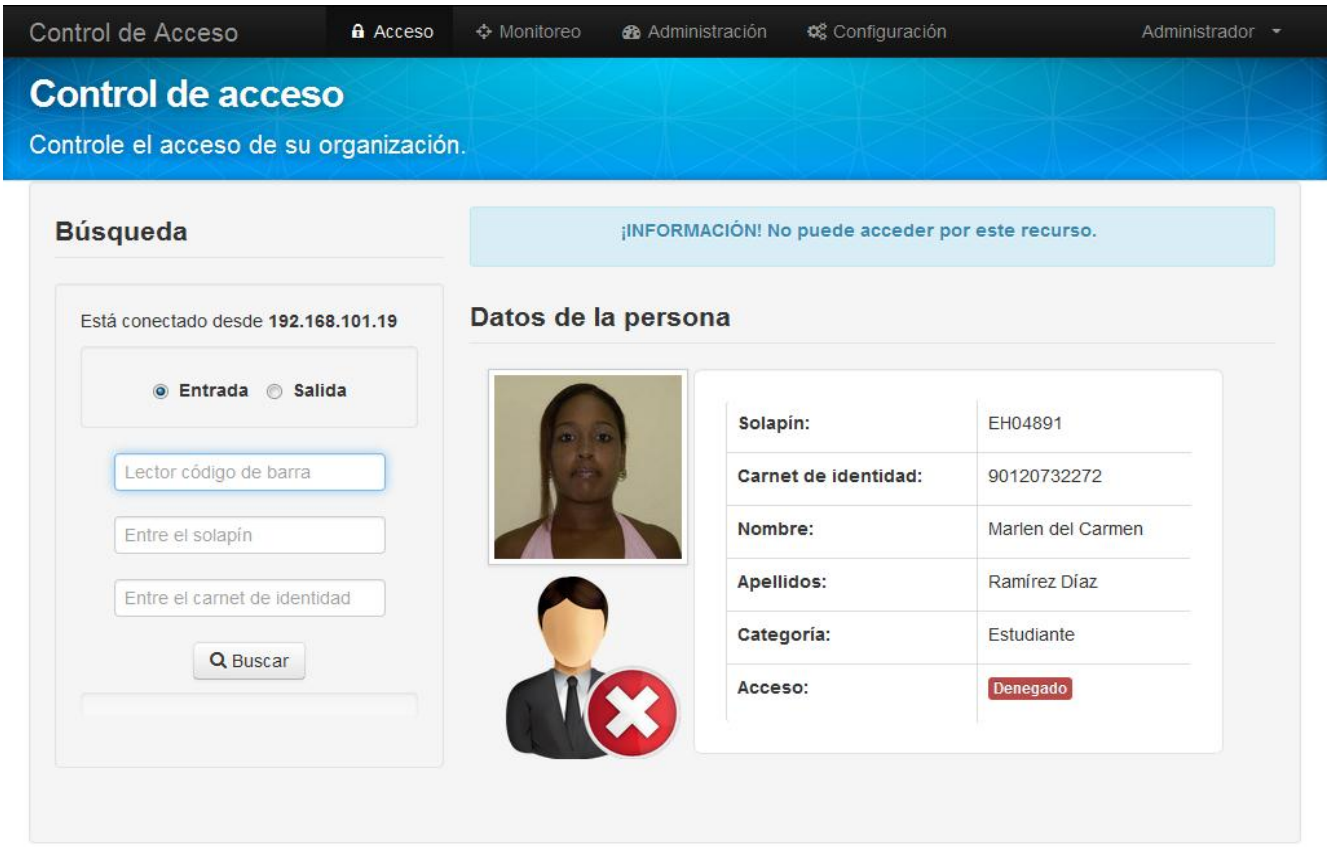
Precondiciones	El usuario debe estar autenticado con el rol Agente de Seguridad.
Funcionalidades asociadas	C1,C2,C3,C4
Conceptos tratados	Usuario, rol, identificación.
Descripción básica	• El sistema muestra la interfaz “Control de acceso”. El usuario autenticado entra el criterio de búsqueda de la persona que intenta acceder, además del tipo de acceso. **Tipo de acceso:** - Entrada o salida **Criterio de búsqueda:** - Código de barras: Se usa un dispositivo para la lectura del código de barra de la identificación. - Número de solapín: filtro de búsqueda por el número de solapín de la persona. - Carnet de identidad: filtro de búsqueda por el número de carnet de

	identidad. - **Al presionar el botón Buscar** Si la persona tiene acceso se muestran sus datos y se registra el acceso. En caso que se detecte que la persona que intenta acceder nos es la propietaria de la identificación se procede a anular el acceso y el sistema crea la infracción. **Datos mostrados:** - Solapín - Carnet de identidad - Nombre - Apellidos - Categoría - Acceso(permitido) - Imagen del usuario - ícono de acceso(permitido) Si la persona no tiene acceso se muestran sus datos **(Ver descripción alterna 1)** y el sistema registra la infracción. Si la persona no se encuentra se muestra un mensaje informando la inexistencia de la misma. **(Ver descripción alterna 2).**
Interfaz	

Descripción alterna 1	
Descripción alterna	**Datos mostrados:**

	- Solapín - Carnet de identidad - Nombre - Apellidos - Categoría - Acceso(denegado) - Imagen del usuario - ícono de acceso(denegado)

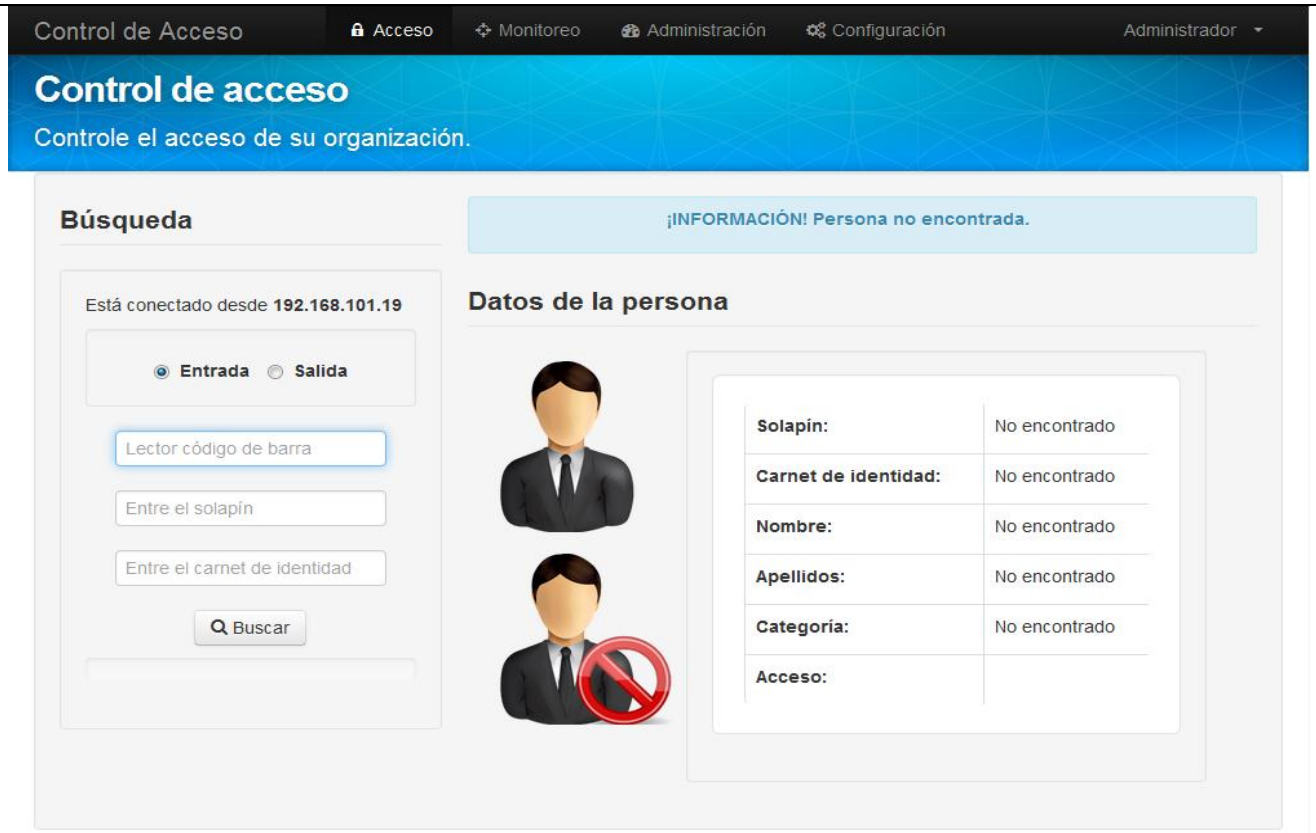
Interfaz



Descripción alterna 2

Descripción alterna	**Mensaje:** ¡INFORMACIÓN! Persona no encontrada.

Interfaz



Validaciones	Si al realizar la búsqueda, los campos se encuentran vacíos se muestra un mensaje de error. El criterio de búsqueda debe tener el formato correcto. Cada vez que se presione el botón “Buscar”, el sistema mostrará un mensaje informando lo ocurrido.
Post-condiciones	Queda registrado o denegado el acceso.
Requisito no funcional	
Servicios	
Componente	

Tabla 2. 3 Especificación del requisito "Registrar acceso de personas".

Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo y estándares [49]. Los siguientes

requisitos no funcionales brindados son de vital importancia para el buen funcionamiento de la aplicación propuesta:

Requisitos de *software*

- Cliente: Navegador *Mozilla Firefox* 17.0 o superior.
- Servidor: Activado el *Internet Information Services (IIS)* 7.5, Sistema Operativo *Windows Server* 2008 R2 con SP2, *Microsoft* .Net *framework* v4.0.

Requisitos de *hardware*

- Cliente: Lector de código de barras, conectado a la red permanentemente, *Pentium* 4 a 1 GHz o superior, mínimo 512 MB de RAM, 40 GB o superior de disco duro.
- Servidor: Conectado a la red permanentemente, Pentium 4 a 2 GHz o superior, mínimo 2 GB de RAM, 250 GB o superior de disco duro.

Restricciones en el diseño y la implementación

- Plataforma de desarrollo .NET utilizando *Visual Studio Ultimate* 2010 SP1.
- Biblioteca *JQuery* 1.9.
- ASP.NET MVC 4 para el desarrollo de la capa de presentación.
- Para el acceso a datos se utilizará el ORM *NHibernate* 3.1.
- *Visual Paradigm for UML* 8.0 *Enterprise Edition* para el modelado
- UML y BPMN como lenguajes de modelado.

Requisitos de apariencia o interfaz externa

La interfaz debe ser amigable con colores poco llamativos, con un diseño sencillo para así tener una mejor interacción con el usuario.

Requisitos de seguridad

Cada usuario realizará operaciones en la aplicación en dependencia de sus privilegios o niveles de acceso.

Requisitos de soporte

Se le debe dar mantenimiento a los servidores de bases de datos en un período de aproximadamente 6 meses, controlando la integridad de la información.

Requisitos de fiabilidad

El sistema debe estar disponible las 24 horas del día, los 7 días de la semana realizándose copias de seguridad semanalmente, tanto de la aplicación como de la base de datos. El tiempo medio entre fallos no debe ser superior a 1 hora.

Requisitos de usabilidad

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

- Conocimientos básicos relativos al uso de una computadora.
- Conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.

El sistema será distribuido en idioma español. Los términos utilizados se establecerán acorde al negocio correspondiente para facilitar la comprensión de la herramienta de trabajo. El sistema poseerá una estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación.

2.5 Planificación

Con el fin de realizar una planificación para el diseño y la implementación de la solución propuesta, en la tabla 2.4 se clasifican las características identificadas anteriormente en críticas y secundarias. Las características críticas son las que tienen mayor importancia porque cubren las principales tareas o funciones que el sistema debe realizar, por lo que deben diseñarse e implementarse en las primeras iteraciones. Las características secundarias aunque tienen un impacto menor, deberán ser diseñadas e implementadas en iteraciones posteriores. También se confeccionó un cronograma para mostrar con detalles las fechas de ejecución de cada característica en las fases de diseño e implementación y del mismo modo la iteración a la que pertenecen reflejadas en el (ANEXO V)**.**

Conjunto de características	Características	Clasificación
Controlar acceso de personas.	C1.Registrar la entrada de personas.	Crítica.
	C2.Registrar la salida de personas.	Crítica.
	C3.Registrar infracción.	Crítica.
	C4.Anular acceso.	Crítica.
Controlar acceso de activos.	C5.Registrar acceso de activo.	Crítica.
	C6. Eliminar acceso de activo.	Crítica.
	C7.Adicionar un nuevo activo.	Crítica.
Monitorear el acceso de	C8. Mostrar datos de accesos de personas en un rango de fecha.	Secundaria.

personas.	C9.Mostrar datos de accesos de personas por un punto de control en un rango de fecha.	Secundaria.
	C10. Mostrar datos de accesos de una persona en un rango de	Secundaria.
	C11.Mostrar datos de accesos de una persona por un punto de control en un rango de fecha.	Secundaria.
	C12. Ver detalles de accesos.	Secundaria.
Monitorear el acceso de activos.	C13.Mostrar datos de accesos de activos en un rango de fecha.	Secundaria.
	C14. Mostrar datos de accesos de activos por un punto de control en un rango de fecha.	Secundaria.
	C15. Mostrar datos de accesos de un activo de una persona en un rango de fecha.	Secundaria.
	C16.Mostrar datos de accesos de un activo por un punto de control en un rango de fecha.	Secundaria.
Monitorear las infracciones.	C17. Mostrar infracciones en un rango de fecha.	Secundaria.
	C18.Mostrar infracciones de una persona en un rango de fecha.	Secundaria.
	C19. Mostrar infracciones por un punto de control en un rango de	Secundaria.
	C20.Mostrar infracciones de una persona por un punto de control en un rango de fecha.	Secundaria.
	C21. Ver detalles de infracciones.	Secundaria.
Monitorear estadísticas gráficas.	C22.Mostrar gráfica con cantidad de accesos e infracciones en el día por intervalos de horas.	Secundaria.
	C23.Mostrar gráfica con cantidad de accesos e infracciones en el año por meses.	Secundaria.
	C24. Mostrar gráfica con cantidad de accesos e infracciones en un rango de fecha.	Secundaria.
	C25. Mostrar gráfica con cantidad de accesos e infracciones de una persona en un rango de fecha.	Secundaria.
	C26. Mostrar gráfica con cantidad de accesos e infracciones por un punto de control en un rango de fecha.	Secundaria.
	C27. Mostrar gráfica con cantidad de accesos e infracciones de una persona por un punto de control en un rango de fecha.	Secundaria.

Tabla 2. 4 Clasificación de las características según su impacto.

2.6 Arquitectura del sistema

La arquitectura de *software* es una forma de representar sistemas complejos mediante el uso de la abstracción. La arquitectura de *software* es importante como disciplina debido a que los sistemas de *software* crecen de forma tal que resulta muy complicado que sean diseñados, especificados y entendidos por un solo individuo [50]. El estilo arquitectural en capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver [51]. Este estilo permite asignar correctamente las funcionalidades a cada capa, pudiéndose reutilizar las capas inferiores que no tengan dependencias con las superiores. El sistema en su vista más abstracta es una solución **Cliente - Servidor** que define la relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a la otra (servidor) y este último le envía las respuestas. La aplicación está compuesta por capas bien definidas y diseñadas, con el objetivo de delegar responsabilidades. En la Figura 2.3 se puede observar una vista de la arquitectura, la cual permite la realización de cambios en las capas sin realizar grandes modificaciones en las demás. Cada capa está integrada por un conjunto de componentes que encapsulan la mayor parte del comportamiento y escenarios en tiempo de desarrollo.

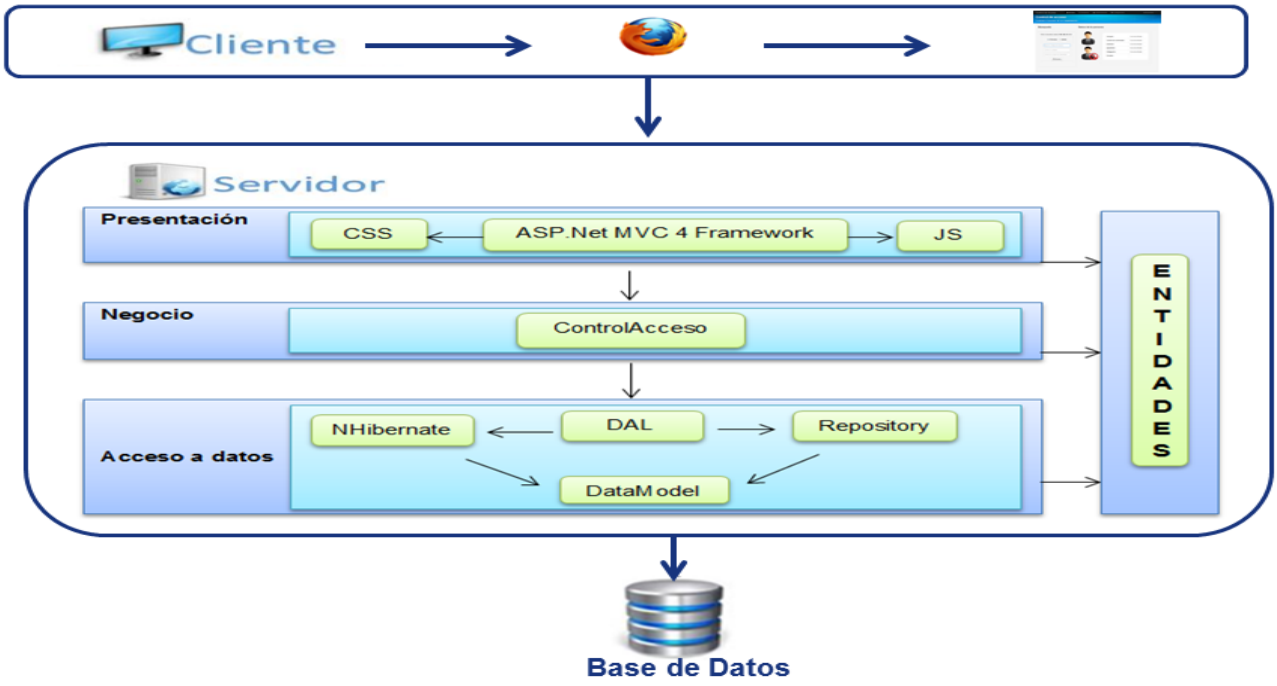


Figura 2. 3 Arquitectura del sistema.

Capa de Presentación:

Se encuentran todas las interfaces que son mostradas al usuario utilizando el patrón arquitectónico MVC donde están presentes los elementos necesarios para su correcto funcionamiento, entre los cuales se

pueden citar: los ficheros de código *JavaScript*, que dan paso a la integración con los componentes de *JQuery*; así como los archivos CSS que contienen estilos para la aplicación. Esta capa de presentación se encuentra representada por el componente ***PMICA.ControlAcceso.WebApp*** de la aplicación y tiene interacción directa con las capas de Negocio y Entidades.

Capa de Negocio:

Está conformado por un conjunto de servicios de negocio que realizan las acciones del negocio. Además mantienen separadas las acciones atómicas del negocio de la definición del proceso. Esta capa se encuentra constituida por el componente ***PMICA.ControlAcceso.Business*** y tiene relación directa con las capas Acceso a Datos y Entidades.

Capa de Acceso a datos:

Es el componente que da soporte a las funcionalidades de la capa de negocio y que se encuentra relacionada con la fuente de datos. La principal función de esta capa es realizar una implementación de las funcionalidades definidas en las interfaces de la capa de negocio y al mismo tiempo trabajar directamente con la fuente de datos. La capa utiliza el *framework NHibernate* para la gestión de los datos persistentes cuando se trabaja con bases de datos relacionales y ayuda a una mejor comunicación entre la aplicación y la base de datos. La capa está constituida por los componentes ***PMICA.ControlAcceso.DAL***, ***PMICA.ControlAcceso.Repository*** y ***PMICA.ControlAcceso.DataModel***.

Capa de Entidades:

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

Modelo Vista Controlador (MVC)

Los patrones son la descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indican la manera de utilizarlos en circunstancias diversas [52]. El patrón Modelo-Vista-Controlador (MVC) es un principio de diseño arquitectónico que separa los componentes de una aplicación *web*. Esta separación ofrece más control sobre las partes individuales de la aplicación, lo cual permite desarrollarlas, modificarlas y probarlas más fácilmente [53]. En la solución planteada se utiliza en la capa de presentación el patrón MVC utilizando específicamente las facilidades que brinda ASP.NET MVC 4.

2.7 Diseño de las funcionalidades

La metodología FDD define que luego de realizar la planificación de las funcionalidades, se comienza por orden de prioridad a diseñar cada una de estas, se identifican las clases involucradas y se describen especificando los métodos que serán implementados en ellas.

Descripción de las clases del sistema

La clase es la unidad básica que encapsula toda la información de un objeto a través de la cual podemos modelar el entorno en estudio [54]. Para mejor comprensión de los diagramas de clases anteriormente expuestos, se ofrece la tabla 2.5, que brinda una breve descripción de la clase *AccesoController* y los métodos correspondientes, siendo una de las más importantes en el sistema. Para poder comprender mejor el funcionamiento de las restantes clases dirigirse al (ANEXO VI).

Clase	Método	Breve descripción
SP_Acceso*Controller*	*Index*()	Muestra la vista correspondiente al control de acceso.
	Index (FormCollection datos)	Se encarga de procesar los datos buscados y mostrar la vista con información referente al acceso.
	AdicionarActivo (carnet, identificador, tipo, *string*, idAcceso)	Adiciona el activo a la persona y se registra su acceso.
	RegistraAccesoActivo (identificador, idAcceso)	Registra el acceso del activo.
	DenegarAccesoActivo(identificador, idAcceso)	Elimina el acceso del activo.
	DenegarAccesoPersona(idAcceso, idInfraccion, carnet)	Elimina el acceso de la persona.

Tabla 2. 5 Descripción de la clase Acceso*Controller.*

Diagramas de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. En un diagrama de clases se pueden distinguir principalmente dos elementos: clases y sus relaciones [54].

Los diagramas de clases pertenecientes al sistema se encuentran organizados por características. En la Figura 2.4 se representa el diagrama del proceso “Control de Acceso”, donde se engloban algunas de las

clases existentes en el sistema desarrollado. Para mayor detalle sobre los diagramas de clases restantes dirigirse al (ANEXO VII).

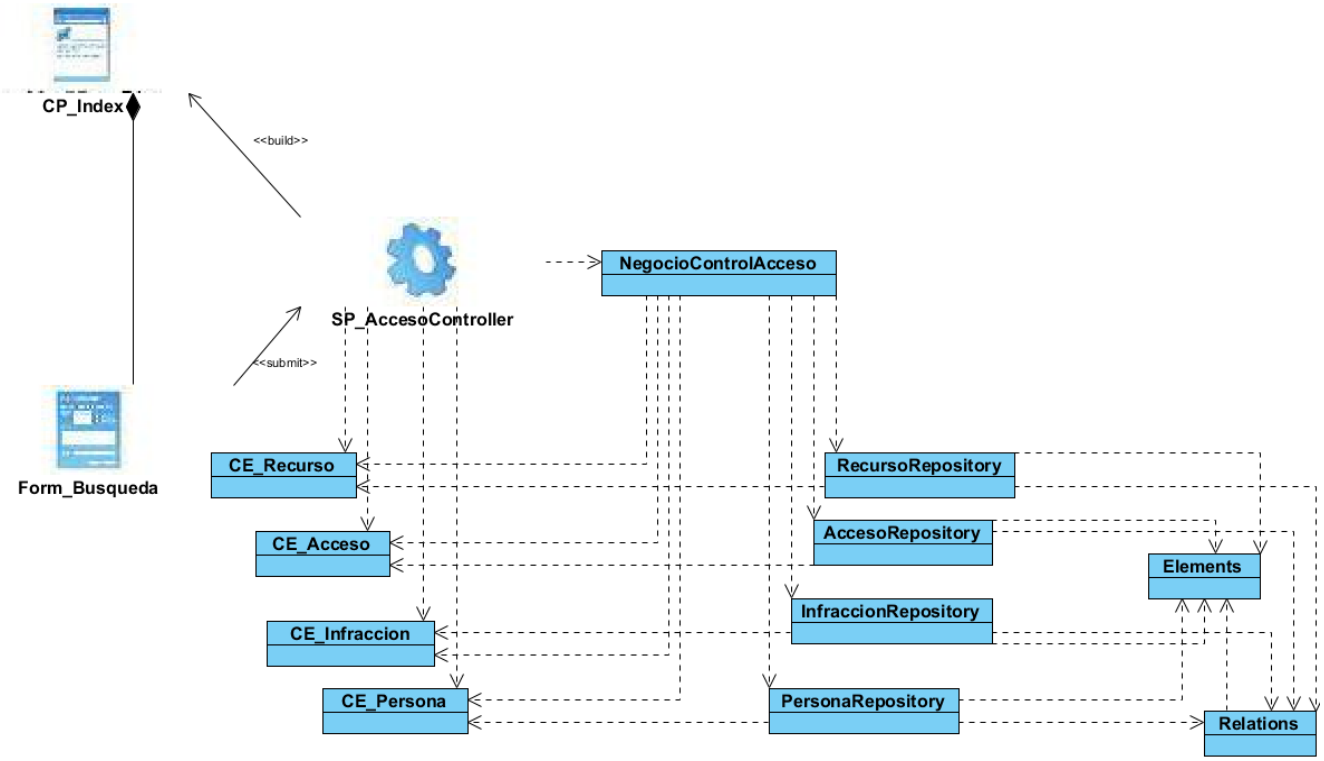


Figura 2. 4 Diagrama de clases "Control de acceso a personas".

Diagramas de secuencia

Un diagrama de interacción explica gráficamente las interacciones existentes entre las instancias (y las clases) del modelo de éstas. El UML define 2 tipos de estos diagramas; ambos sirven para expresar interacciones semejantes o idénticas de mensaje, estos son los diagramas de colaboración y secuencia. Los diagramas de secuencia describen las interacciones en una especie de formato de cerca o muro [55]. En la Figura 2.5 se ofrece el diagrama de secuencia “Control de acceso de activos”, que brinda una mejor visión de cómo funciona este. Para obtener más detalles de otras funciones importantes en la aplicación ver (ANEXO VIII).

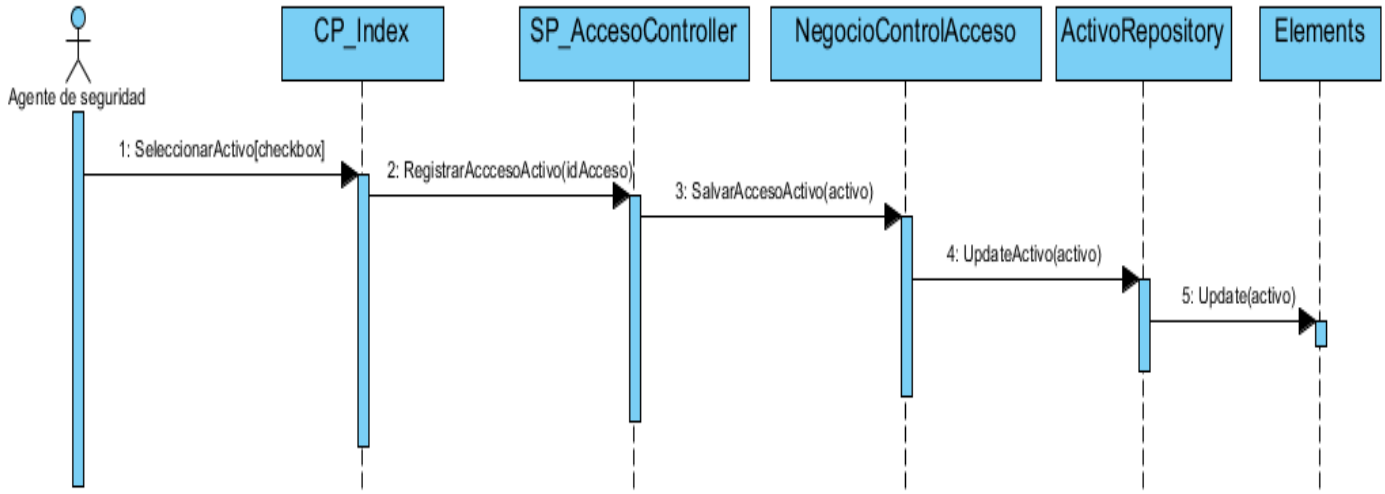


Figura 2. 5 Diagrama de secuencia "Control de acceso de activos".

2.8 Patrones de diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de *software*, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular [56].

Dentro de los patrones de diseño se encuentran dos grupos fundamentales conocidos por Patrones Generales de Asignación de Responsabilidades de *Software* (GRASP por sus siglas en inglés) y Banda de los Cuatro (GOF por sus siglas en inglés). Los Patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones [57].

Los Patrones GOF describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Los patrones de diseño GOF se clasifican en tres categorías:

- **Creacionales:** describen las formas de crear instancias de objetos. El objetivo de estos patrones es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.
- **Estructurales**: describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionados pueden ser incluso objetos simples u objetos compuestos.
- **Comportamiento**: definen la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos [58].

Patrones utilizados

- **Experto:** ¿Quién asumirá la responsabilidad en el caso general?

Perteneciente al grupo de patrones GRASP, consiste en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Con este patrón se pretende que los objetos hagan cosas relacionadas con la información que poseen [59].

```
public class Acceso : Elemento
{
    private TipoAcceso _tipoAcceso;

    public TipoAcceso TipoAcceso
    {
        get
        {
            string aux = _reader.GetAttributeValueBySchema(Data, "Acceso", "tipoAcceso");
            switch (aux)
            {
                case "Entrada":
                    _tipoAcceso = TipoAcceso.Entrada;
                    break;
                case "Salida":
                    _tipoAcceso = TipoAcceso.Salida;
                    break;
            }
            return _tipoAcceso;
        }
```

Figura 2. 6 Ejemplo de patrón Experto.

- **Creador:** ¿Quién crea?

  Perteneciente al grupo de patrones GRASP se basa en asignarle a la clase B la responsabilidad de crear una instancia de la clase A. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos y tiene como propósito fundamental encontrar un creador que se debe conectar con el objeto producido en cualquier evento [59].

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

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

Figura 2. 7 Ejemplo de patrón Creador.

- **Alta cohesión:** ¿Cómo mantener controlable la complejidad?

Perteneciente al grupo de patrones GRASP, se basa en asignar una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las

responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que realicen un trabajo enorme [59].

```
public class ElementsRepository : GenericRepository<Elements>, IElementsRepository
{
    public ElementsRepository(ISession session) : base(session)...

    public Elements GetElementById(Guid idElement)...

    public List<Elements> GetElementsByType(string type)...

    public string GetTypeById(Guid idElement)...

    public string GetDataById(Guid idElement)...

    public string GetElementTagById(Guid id, string tag)...

    public Dictionary<string, Guid> GetElementIdByXmlTag(string elementType, string tag)...

    public List<Guid> GetElementIdList(string elementType)...
```

Figura 2. 8 Ejemplo de patrón Alta cohesión.

- **Controlador:** ¿Quién administra un evento del sistema?

El Patrón Controlador perteneciente al grupo de patrones GRASP se encarga de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa [59].

```
public class AccesoController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
```

Figura 2. 9 Ejemplo de patrón controlador.

- **Bajo acoplamiento:** ¿Cómo dar soporte a poca dependencia y a una mayor reutilización?

Perteneciente al grupo de patrones GRASP consiste en asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras, con las que conoce y con que recurre a ellas. El patrón propone el diseño de clases más independientes, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas [59].

```
public class AccesoController : Controller
{
    private ControlAccesoBusiness controlAccesoNegocio=new ControlAccesoBusiness();
```

Figura 2. 10 Ejemplo de patrón Bajo acoplamiento.

- **Recuerdo:**

Recuerdo es un patrón de comportamiento perteneciente al grupo de patrones GOF que posibilita volver a estados anteriores del sistema [60]. Este patrón es muy utilizado en la solución, un ejemplo claro puede ser que en el momento en que se intenta crear un acceso ocurre un fallo, el sistema revierte todo lo que se realizó y vuelve al estado anterior.

- **Repositorio:**

El patrón Repositorio es un patrón estructural perteneciente a la familia de los patrones GOF, es una construcción común para evitar la duplicación de la lógica de acceso a datos a través de la aplicación [61]. Un repositorio realiza las tareas de intermediario entre las capas de modelo de dominio y mapeo de datos. Conceptualmente, un repositorio encapsula a un conjunto de objetos almacenados en la base de datos y las operaciones que sobre ellos pueden realizarse [62]. Este patrón se usa en la solución ya que se tiene un repositorio por cada clase entidad para así tener un mejor manejo de las mismas.

2.9 Modelo de datos del sistema

Un modelo de datos es una definición lógica, independiente y abstracta de los objetos, operadores y demás que en conjunto constituyen la máquina abstracta con la que interactúan los usuarios. Los objetos nos permiten modelar la estructura de los datos. Los operadores nos permiten modelar su comportamiento [63]. El modelo de datos de la solución, Figura 2.11, está formado por dos tablas, *Element y Relations*. Los elementos están compuestos por un identificador, un tipo y los datos. Los tipos de elementos así como los datos contenidos en un XML, están esquematizados por un XSD. Por otra parte, la tabla *Relations* contiene las relaciones entre elementos y está compuesta por un identificador para la relación, el identificador del titular de la relación, el identificador del elemento relacionado con el titular y el tipo de relación que estos poseen. Este modelo es una variante de un patrón de diseño conocido como EVA (*Entity Value Attribute*). El cual permite tener un dominio detallado sobre todos los atributos que se le asignan a cualquier elemento que es almacenado. Las principales características de este patrón y que a su vez se convierten en sus principales ventajas, son la gran flexibilidad en el almacenamiento de datos y la posibilidad de ampliar el conjunto de atributos sin cambiar la estructura de la tabla.

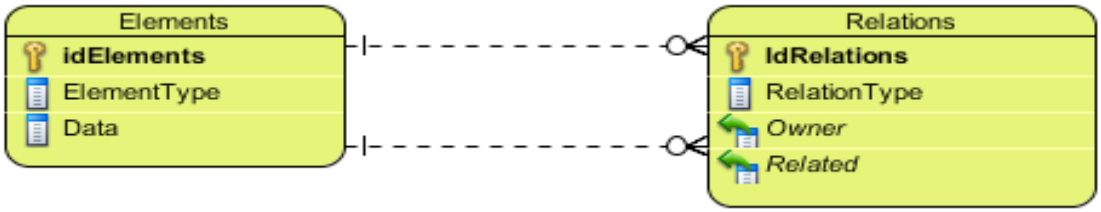


Figura 2. 11 Modelo de datos.

2.10 Conclusiones

La descripción detallada de los procesos posibilitó definir las entradas y salidas de todas las actividades. La creación del Modelo del dominio, posibilitó identificar conceptos asociados con el entorno en que se desarrolla la solución, además las relaciones entre ellos. La creación del listado de características permitió definir de forma concreta las funcionalidades que automatiza la solución. Con la realización de los diagramas de clases y secuencia se mostró cómo puede ser construido el sistema, sirviendo esto como abstracción del modelo de implementación y el código fuente. Se definió la arquitectura, donde se utiliza la solución Cliente-Servidor con el estilo de N-capas, especificando cuáles capas formará parte de la solución, así como la interacción entre ellas. El estudio realizado en el capítulo facilitó un entendimiento de las particularidades del sistema, contribuyendo a una mejor comprensión del problema inicialmente planteado.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

3.1 Introducción

En este capítulo se abordan aspectos importantes de la implementación y pruebas del sistema. Primeramente se describe el estilo de codificación utilizado. Luego se muestra la distribución física del sistema entre los diferentes nodos de cómputo a través del diagrama de despliegue. Se presenta el diagrama de componentes que permite comprender la estructura del sistema y se muestran los resultados de los casos de prueba realizados.

3.2 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del *software* y para obtener un buen rendimiento [64]. Seguir un estándar en la codificación es importante, ya que mejora la lectura del *software*, permitiendo entender el código rápidamente.

3.3 Reglas de codificación

Las técnicas de codificación incorporan muchos aspectos del desarrollo del *software*. Aunque generalmente no afectan a la funcionalidad de la aplicación, sí contribuyen a una mejor compresión del código fuente [65]. En la presente investigación se ofrecen las reglas de codificación que se determinaron, siendo factibles para desarrollar un código estándar y flexible para la aplicación. Las mismas son:

- Evitar escribir métodos de más de 25 líneas.
- No usar directamente números o cadenas como constantes en el código. Declarar las constantes en la parte superior de los ficheros.
- Convertir las cadenas a mayúsculas o minúsculas antes de compararlas.
- Usar "*enum*" siempre que sea requerido. No usar números o cadenas para indicar valores discretos.
- Si un valor errado es encontrado en los ficheros de configuración la aplicación debe mostrar a usuario los valores correctos.
- No programar más de una clase por cada fichero.
- Evitar tener ficheros muy largos. Se deben dividir en dos o más clases.
- Evitar el paso de demasiados parámetros a los métodos, tratar de que no haya métodos de más de 3 parámetros.

- Organizar los ficheros en carpetas apropiadas. Usar dos niveles de profundidad y no más de 10 carpetas en la raíz y no más de 5 subcarpetas en cada carpeta hija.
- Se debe seguir las convenciones y reglas de nombre mostradas en la tabla 3.1.

Los términos Convención Pascal y Convención Camel son usados en la implementación del sistema.

- **Convención Pascal:** el primer caracter de cada palabra es en mayúscula y el resto en minúscula. Ejemplo: *BackColor*
- **Convención Camel:** el primer caracter de cada palabra es en mayúscula (excepto la primera palabra) y el resto en minúscula.

  Ejemplo: *backColor*

Tipos de identificadores	Regla y/o Convención	Ejemplos
Clases.	-Pascal.	*public class* NegocioControlAcceso
Interfaces.	-Prefijo "I". -Pascal.	*public interface IAccesoRepository*
Métodos.	-El nombre del método debe decir que hace. Cada método debe cumplir solamente una función. No se deben combinar más de una función por método.	EliminarAcceso(Acceso acceso)
Variables y parámetros.	-Usar el significado de las palabras para el nombre de las variables. Todas las variables miembros de las clases deben ser prefijadas con underscores (_) para ser diferenciadas de otras variables. Las variables booleanas y las propiedades se les pueden poner como prefijos "*is*". Siempre vigila parámetros no esperados.	*private bool _isHabilitado;* *if* (persona.Habilitado) { *return true;* } *else if* (!persona.Habilitado) { *return false;* }
Namespace.	Deben seguir el siguiente patrón estándar: < product name >.<top level module>.<bottom level module>	PMICA.ControlAcceso.WebApp.*Controllers*.ControlAcceso
Comentarios.	Comentarios deben estar en el mismo nivel del código.	//Devolver la vista *return View*();

Llaves.	Las llaves se deben poner al mismo nivel del código que las contiene.	*public ActionResult Index*() { *return View*(); }
#region.	Se usa para agrupar código.	*#region* Vistas *#endregion*

Tabla 3. 1 Reglas y convenciones para la codificación.

3.4 Tratamiento de errores

Para garantizar la mayor integridad y confiabilidad posible en los datos que utiliza el sistema, se adoptó la siguiente estrategia para el tratamiento de errores:

- El código implementado se encuentra encerrado dentro de bloques *try-catch*, de forma tal que pueda capturarse cada excepción que sea lanzada por el sistema en tiempo de ejecución.
- Los mensajes de error que emitirá el sistema se mostrarán en un lenguaje de fácil comprensión para los usuarios.
- Dado el caso que la información introducida en un formulario sea incorrecta o incompleta, se informará al usuario con una breve descripción.
- Los campos son validados tanto en el cliente como en el servidor.

3.5 Diagrama de componentes

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces[66]. Además modela la vista de implementación estática de un sistema y los elementos físicos que residen en un nodo, tales como ejecutables, tablas, librerías, archivos y documentos [67]

En la Figura 3.1 se muestra el diagrama de componentes de la aplicación, donde se muestra como está estructurada la solución propuesta; cómo interactúan los componentes y las relaciones entre ellos, dando una mejor comprensión de la estructura de la aplicación. De igual forma en la tabla 3.2 se define el propósito de cada componente desarrollado.

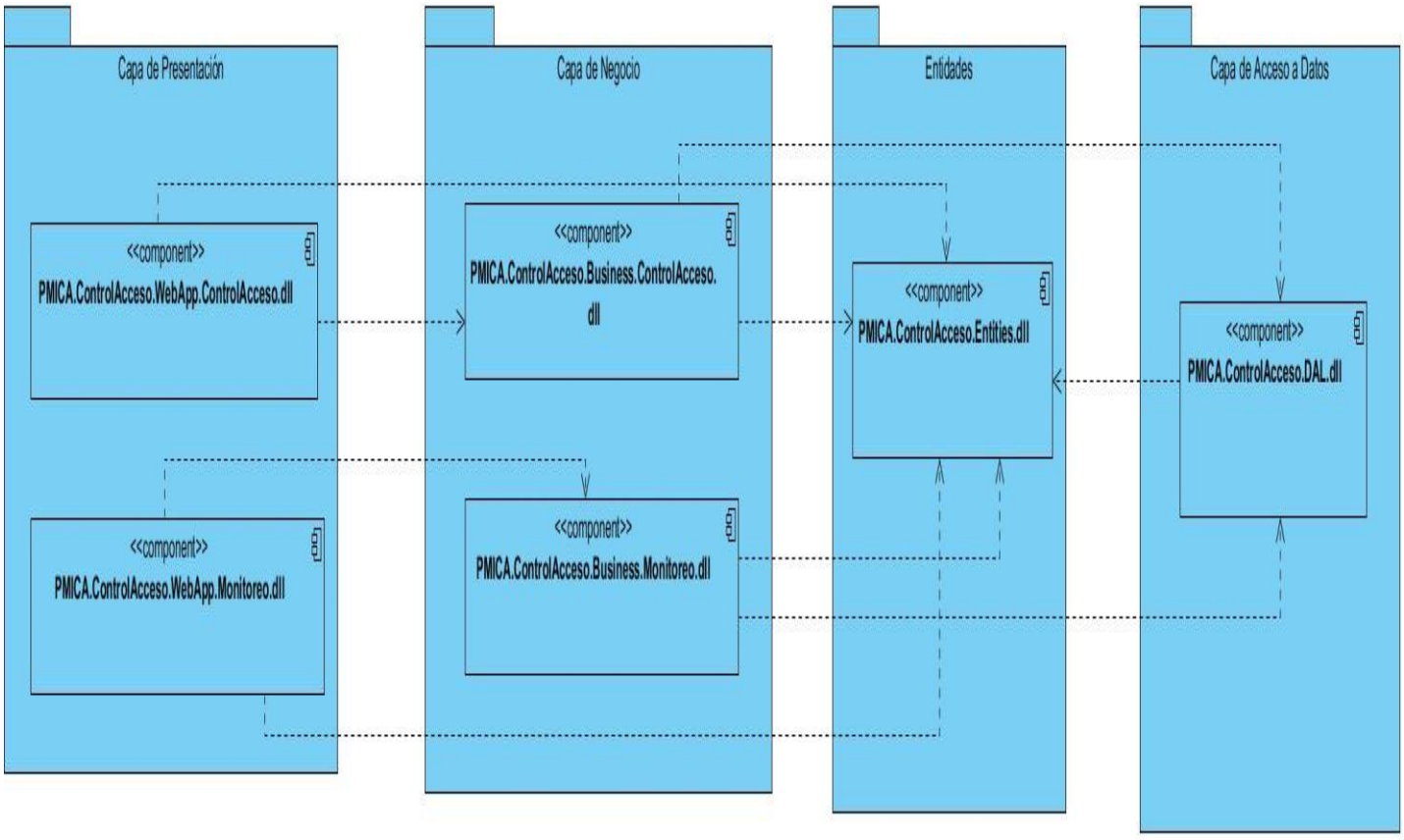


Figura 3. 1 Diagrama de Componentes.

Componente	Capa a la que pertenece	Descripción
PMICA.ControlAcceso.*WebApp*.ControlAcceso	Presentación.	En este se encuentran las interfaces de usuario que están relacionadas con el control de acceso.
PMICA.ControlAcceso.*WebApp*.Monitoreo	Presentación.	En este se encuentran las interfaces de usuario que están relacionadas con el monitoreo.
PMICA.ControlAcceso.*Business*.ControlAcceso	Negocio.	En este se encuentran los métodos relacionados con el control de acceso.
PMICA.ControlAcceso.*Business*.Monitoreo	Negocio.	En este se encuentran los métodos relacionados con el monitoreo.
PMICA.ControlAcceso.*Entities*	Entidad.	En este se encuentran todas las

		clases entidades del sistema.
PMICA.ControlAcceso.*DAL*	Acceso a Datos.	En este se encuentran las interfaces de usuario relacionadas con el acceso a datos.

Tabla 3. 2 Descripción de los componentes.

3.6 Diagrama de despliegue

En el diagrama de despliegue se indica la situación física de los componentes lógicos desarrollados. Es decir, se sitúa el *software* en el *hardware* que lo contiene. Un nodo es un elemento donde se ejecutan los componentes, representan el despliegue físico de estos componentes[68].

En el diagrama que se muestra en la Figura 3.2, el nodo **"Computadora punto de control"** representa la computadora que se encuentra en los puntos de control de acceso, que utiliza un lector de código de barras conectado por el puerto USB representado por el nodo **"Lector código de barra"** que servirá para la lectura del código de barra de la identificación de las personas que soliciten el acceso; desde esta computadora se podrá acceder al servidor de aplicaciones esquematizado por el nodo **"Servidor Aplicaciones(IIS)"** donde se encuentra la aplicación que se encarga de manejar toda la lógica utilizando el servidor **LDAP**(nodo "ldap.uci.cu") para la autenticación de los usuarios y el servidor ***Photostore*** (nodo "photostore.uci.cu") para cargar la foto de las personas. El nodo **"Computadora administrador"** representa la computadora de donde el administrador del sistema podrá conectarse a la aplicación y por último el nodo **"Servidor de base de datos"** donde se encuentra la base de datos asociada a la aplicación.

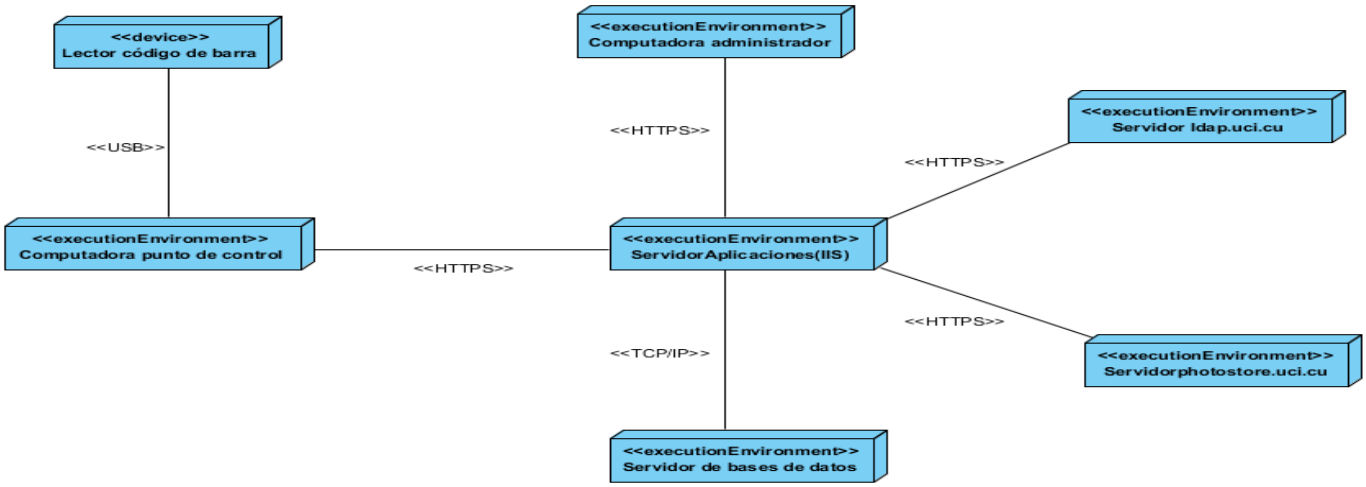


Figura 3. 2 Diagrama de despliegue.

3.7 Interfaces de la solución propuesta

El diseño de las interfaces es un paso importante en la realización del *software*, con las mismas se puede lograr que el usuario observe la apariencia de la aplicación desarrollada. Las interfaces pertenecientes al sistema de control de acceso se muestran a continuación en la Figura 3.3. Para más detalles sobre otras interfaces de la aplicación dirigirse al (ANEXO IX).

Figura 3. 3 Interfaz "Control de Acceso".

3.8 Pruebas de la propuesta de solución

Las pruebas de *software* implican ejecutar una implementación del *software* con datos de pruebas. Son una técnica dinámica de verificación y validación [69]. Las pruebas son el último bastión para la evaluación de la calidad y el descubrimiento de errores [70]. Con la necesidad de comprobar la calidad, usabilidad, posibles fallos de implementación y otras funcionalidades, se realizaron las pruebas necesarias al sistema. Esta etapa se llevó a cabo para evaluar la calidad del producto desarrollado, garantizando que este funcione de acuerdo a las especificaciones del cliente a través de requerimientos funcionales. Con tales fines, se realizaron pruebas de unidad y pruebas de caja negra.

Pruebas unitarias

En la programación, una prueba unitaria o de unidad es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada una de las partes que integran la aplicación

funcione correctamente por separado [71]. Para realizar estas pruebas se utilizó el entorno de desarrollo correspondiente (*Visual Studio* 2010) que puede generar este tipo de pruebas. A continuación se muestra una de las pruebas unitarias realizadas a la funcionalidad del sistema "TieneAccesoPersonaRecurso" en la tabla 3.3. Para observar las restantes pruebas unitarias ir al (ANEXO X)**.**

<table>
<tr><th colspan="3">Prueba de unidad</th></tr>
<tr><td colspan="3">Nombre: TieneAccesoPersonaRecursoTest.</td></tr>
<tr><td>Estado: Satisfactoria.</td><td>Tipo: Caja Blanca.</td><td>Última Ejecución: 14/05/2013.</td></tr>
<tr><td colspan="2">Ejecutado por: Jesús Camilo Gámez.</td><td>Verificado por: Marlen del Carmen Ramírez.</td></tr>
<tr><td colspan="3">Descripción: Para la ejecución de esta prueba se debe entrar la persona y el recurso.</td></tr>
<tr><td colspan="3">Entrada: Persona persona, Recurso recurso.</td></tr>
<tr><td colspan="3">Criterio de aceptación: Verificar si una persona tiene acceso a un recurso.</td></tr>
<tr><td colspan="3">Resultado:
Ejecución de pruebas completado Resultados: 1/1 correctas; Elementos comprobados: 0
Resultado | Nombre de la prueba | Proyecto | Mensaje de error
Pasada | TieneAccesoPersonaRecursoTest | PMICA.ControlAcceso.UnitTest |</td></tr>
</table>

Tabla 3. 3 Descripción de la prueba unitaria "TieneAccesoPersonaRecursoTest".

En la siguiente tabla (tabla 3.4) se representan los resultados alcanzados en la aplicación de las pruebas de unidad realizadas durante las 4 iteraciones de desarrollo.

Iteraciones	Funcionalidades	
	Funcionalidades con errores	Funcionalidades correctas
1	3	2
2	3	3
3	1	3
4	0	6

Tabla 3. 4 Iteraciones de las pruebas de unidad.

En el siguiente gráfico (Figura 3.4) se encuentran representados los resultados de las pruebas unitarias que se han realizado en cada una de las iteraciones hasta el momento, donde se detallan las iteraciones y el número de funcionalidades que fueron probadas y aceptadas en cada una.

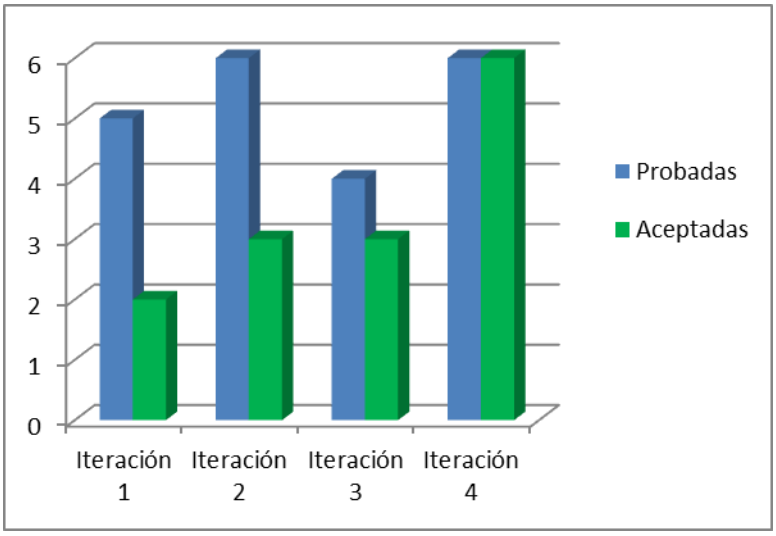


Figura 3. 4 Gráfico del resultado por iteraciones de las pruebas de unidad.

Pruebas de caja negra

Las pruebas de caja negra son las que se aplican sobre las interfaces del *software*. Una prueba de este tipo examina algún aspecto funcional de un sistema [72]. De igual manera verifica que los datos de entrada válidos sean aceptados y los inválidos denegados, así como que la integridad de la información externa se mantenga. Para validar cada uno de los requisitos funcionales del sistema se aplicaron diferentes casos de pruebas (tabla 3.5) a los requisitos definidos previamente en los dos módulos correspondientes del sistema(Control de acceso y Monitoreo). Para observar más detalles de las pruebas realizadas dirigirse al (ANEXO XI).

Escenario	Descripción	Texto	Respuesta del sistema	Flujo central
Esc. 1.1: Registrar acceso de persona mediante criterio.	La persona muestra su documento de identificación para que el código de barras sea leído por el lector o se escoja cualquier criterio de búsqueda.	Código de barras(válido) Solapín(válido) Carnet de identidad (válido).	El sistema devolverá los datos de la persona correspondiente.	1. Opción Acceso. 2. Búsqueda.
		Código de barras (inválido).	El sistema mostrará un mensaje y un ícono de acceso denegado.	
		Solapín (inválido).	El sistema mostrará el mensaje: Persona no encontrada.	
		Carnet de identidad (inválido.)	El sistema mostrará un	

			mensaje notificando que el carnet debe tener 11 dígitos o debe ser numérico solamente.	

Tabla 3. 5 Caso de prueba: Módulo "Control de acceso".

En la siguiente tabla (tabla 3.6) se exponen las principales variables utilizadas en los casos de pruebas:

No.	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Código de barras	campo de texto	Sí	Este campo nunca estará vacío mientras se utilice el lector de código de barras a no ser que se imposibilite la lectura de la identificación.
2	Solapín	campo de texto	Sí	Debe de insertarse un código con letras y números detrás. Este campo puede o no quedar vacío si no se utiliza ese criterio de búsqueda de la persona.
3	Carnet de Identidad	campo de texto	Sí	Debe insertarse un número y que sea de sólo 11 dígitos. Este campo puede o no quedar vacío si no se utiliza ese criterio de búsqueda de la persona.
4	Serie	campo de texto	No	Puede insertarse un número seguido de letras, en dependencia del código que posea el activo. Este campo no puede quedar vacío.
5	Tipo	campo de texto	No	Debe insertarse el tipo del activo (laptop, vehículo, otros equipos). Este campo no puede quedar vacío.
6	Descripción	campo de texto	Sí	Se introducen otros datos del activo. Este campo puede o no quedar en blanco.
7	Fecha inicial	Campo de texto para introducir la fecha o calendario para seleccionar la fecha	No	En este campo se puede insertar o elegir la fecha. Este campo no debe quedar en blanco.
8	Fecha final	Campo de texto para introducir la fecha o calendario para seleccionar la fecha.	No	En este campo se puede insertar o elegir la fecha. Este campo no debe quedar en blanco.

Tabla 3. 6 Descripción de las variables de los Casos de Pruebas.

En la siguiente gráfica (Figura 3.5) se encuentran representados los resultados de las pruebas de validación que se han realizado en cada una de las iteraciones hasta el momento:

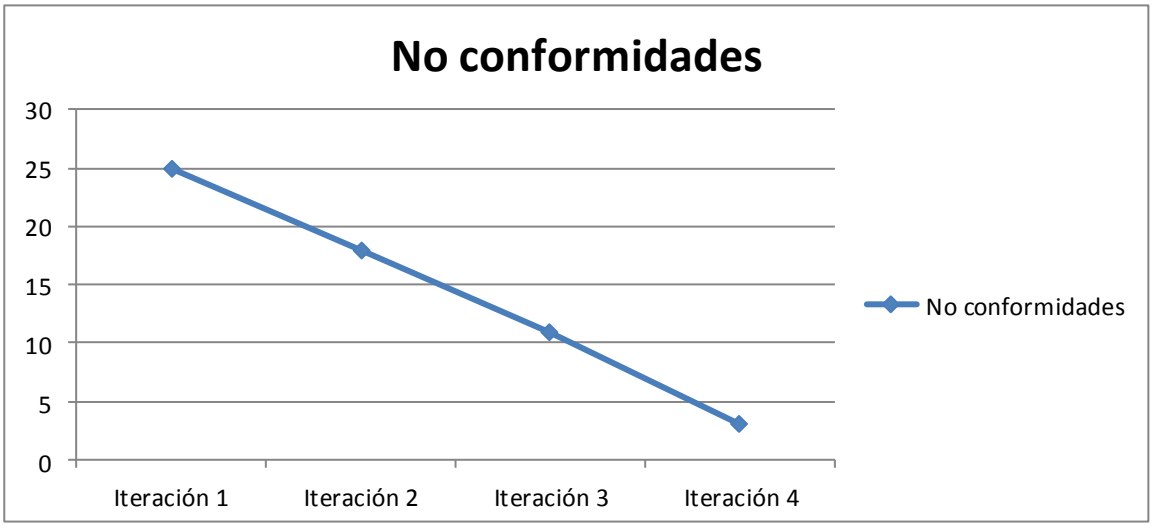


Figura 3. 5 Gráfico del resultado de las pruebas de caja negra.

3.9 Beneficios de la solución propuesta

- El sistema eliminará el proceso manual que se lleva a cabo en la institución, posibilitando que el control de acceso y el monitoreo se hagan de forma digital.
- Además de controlar a las personas, el sistema puede controlar el acceso de todo tipo de activo.
- El sistema es capaz de registrar accesos e infracciones a través de las restricciones especificadas por el usuario.
- Además de controlar el acceso a la UCI, se puede realizar el control de cada una la áreas con que cuenta la universidad.

3.10 Conclusiones

En este capítulo se definió el estilo de codificación utilizado en la implementación del sistema, lo que permitió mantener una uniformidad en la codificación y mayor claridad a la hora de leer o agregar código. La construcción del diagrama de despliegue permitió dejar claro la distribución física del sistema entre los diferentes nodos de cómputo. La implementación de los procesos de Control de acceso y Monitoreo, logró satisfacer los requisitos definidos. La ejecución de las pruebas permitió comprobar el buen funcionamiento del sistema y la calidad del producto.

CONCLUSIONES GENERALES

Una vez finalizada la implementación de los módulos de control de acceso y monitoreo, se pudo comprobar el cumplimiento de los objetivos trazados y se llegan a las conclusiones siguientes:

- A partir del estudio de las soluciones existentes en el ámbito nacional e internacional se demostró que estas no pueden ser utilizadas en la UCI, por lo que se determinó la necesidad de un sistema que cumpla los requerimientos necesarios para ser utilizado en la universidad.
- Las herramientas concebidas por el proyecto fueron estudiadas y cumplieron con todos los requisitos para ser empleadas en la solución propuesta.
- Un estudio detallado del negocio evidenció que el flujo actual de los procesos que se realizaban en el control de acceso en la UCI, no se efectuaba de la mejor manera, debido a la realización del trabajo de forma manual.
- A través de los estándares especificados se obtuvo un código limpio y legible, además se mantuvo un correcto manejo de errores.
- Con la realización de las pruebas se identificaron las no conformidades que fueron corregidas en cada iteración planificada.

RECOMENDACIONES

A los interesados en mejorar o continuar esta investigación se les recomienda:

- Agregar nuevas funcionalidades al módulo de monitoreo según las necesidades que vayan surgiendo a la Oficina de seguridad y protección.
- Realizar un profundo estudio sobre el control de acceso de los visitantes para su posterior informatización.

REFERENCIAS BIBLIOGRÁFICAS

1. *Ecured.Sistemas de control de acceso.* [cited 2013 28 de enero]; Available from: http://www.ecured.cu/index.php/Sistemas_de_control_de_acceso.
2. *Definicion abc.Vigilancia.* [cited 2013 23 de abril]; Available from: http://www.definicionabc.com/general/vigilancia.php..
3. Sierra., Ó.C. (2007) *EVOLUCIÓN Y RETOS DE LOS SISTEMAS DE CONTROL DE ACCESOS.*
4. *Articulos.Historia de los sistemas biométricos para el control de acceso.* 2013; Available from: http://www.articuloz.com/seguridad-articulos/historia-de-los-sistemas-biometricos-para-el-control-de-acceso-4423152.html.
5. *SCSSA.Control de acceso.* [cited 2013 6 de febrero]; Available from: http://www.scssa.com.ar/control-de-acceso.html.
6. *DokkoGroup*. [cited 2013 6 de febrero]; Available from: http://www.dokkogroup.com.ar/wapa.php.
7. *INBIOSYS.Sistema de control de acceso del personal de la empresa.* [cited 2013 6 de febrero]; Available from: http://inbiosys.wordpress.com/nuestros-productos/sistema-de-control-de-acceso-del-personal-de-la-empresa/.
8. *LARCON-SIA.Control de acceso.* [cited 2013 5 de febrero]; Available from: http://www.larconsia.com/05_controldeacceso_accpro.asp.
9. *Bosch*. [cited 2013 6 de febrero]; Available from: http://www.boschsecurity.us/en-us/aec.
10. *Datys*. [cited 2013 6 de febrero]; Available from: http://www.datys.cu/wpversolucion.
11. *Datys*. [cited 2013 6 de febrero]; Available from: http://www.datys.cu/wpinfproducto.
12. Prieto., A.F.C.e.I.R., *Sistema control de acceso a Comedores en la Universidad de las Ciencias Informáticas.* 2007, Universidad de las Ciencias Informáticas.: Ciudad de La Habana,Cuba.
13. González, S.L.S., *Sistema único de control de acceso para personas vinculadas a la producción en el centro CISED de la Universidad de la Ciencias Informáticas.* 2012: Ciudad de La Habana,Cuba.
14. *barrasybarras.Uso interno.* . [cited 2013 6 de febrero]; Available from: http://www.barrasybarras.com/uso-interno.
15. *CGI.Productos lectores-código.* [cited 2013 6 de febrero]; Available from: http://www.insumoscgi.com.ar/productos-lectores-codigo.htm.
16. *Ingenieria de Software.*, Universidad de Union Bolivariana.: México.
17. Jorge Carlos Valverde Rebaza, S.D.y.A.C. (2007) *Metodologías ágiles.*
18. Calabria., L. (2003) *Metodología FDD*.
19. ., R.P., *An introduction to UML.* 2007.

20. Alfonso Rodríguez, E.F.M.y.M.P. (2005) *3er Congreso Iberoamericano de Seguridad Informática.Hacia la definición de Procesos de Negocios basados en una Arquitectura Dirigida por Modelos.*
21. Pérez., J.D., *Notaciones y lenguajes de procesos. Una visión global.*

, in *Computer Languages and Systems.*, Universidad de Sevilla.: Sevilla,España. p. 109.

22. Ruiz, F., *Tecnología para la Gestión de Procesos de Negocio.* 2006, Universidad de Castilla-La Mancha.

: España.

23. Schmuller., J., *Aprendiendo UML en 24 horas.*, ed. P. EDUCACION. 2000, México. 448.
24. *Microbuffer.* . [cited 2013 17 de marzo]; Available from: http://microbuffer.wordpress.com/2011/05/04/que-es-postgresql/.
25. *Desarrolloweb.Actualidad de postgresql-9-1.* [cited 2013 17 de marzo]; Available from: http://www.desarrolloweb.com/actualidad/postgresql-9-1-disponible-5841.html.
26. *Piensa en Binario.Lo nuevo en postgresql-9-1.* 2011 [cited 2013 22 de mayo]; Available from: http://www.piensaenbinario.com/2011/09/lo-nuevo-en-postgresql-91.html.
27. Schenker., A.C.a.D.G.N., *NHibernate 3 Beginner's Guide.* Second edition. ed. 2011: Mallik Neha y Llewellyn F. Rozario. .
28. PIERRE HENRI KUATÉ, T.H., CHRISTIAN BAUER AND GAVIN KING., *NHibernate in Action.* 2009, United States of America.: Cynthia Kane y Tiffany Taylor. 385.
29. 4., M.L.n.e.N.F. 2007 [cited 2013 30 de mayo]; Available from: http://msdn.microsoft.com/es-es/library/ms171868%28v=vs.100%29.aspx.
30. Feeman, A.y.S., Sanderson. , *Pro ASP.NET MVC Framework.* . Third Edition ed. 2011, New York,EUA.: Paul Manning. 837.
31. Jon Galloway, P.H., Brad Wilson and K. Scott Allen. , *PROFESSIONAL ASP.NET MVC .* ed. I. John Wiley & Sons. 2011, Indianapolis,EUA.: Pau Reese, Maureen Spears y Eilon Lipton. 433.
32. *MSDN.ASP.NET MVC 4.* 2013; Available from: http://social.msdn.microsoft.com/Search/en-US?query=asp.net%20mvc%204&ac=4.
33. Alvarez., M.A. *Manual de JQuery.*, 62.
34. *JQuery Community Experts. JQuery Cookbook. Solutions & Examples for JQuery Developers.* , ed. C. Lindley. 2010, United States of America.
35. Sharp, J., *Microsoft Visual C# 2010.Step by Step.* 2010, Redmond, Washington,United States of America.: Ben Ryan,Devon Musgrave y Rosemary Caperton.

36. Seco, J.A.G., *El lenguaje de programación C#.*
37. *MSDN.Visual Studio.* [cited 2013 7 de abril]; Available from: http://msdn.microsoft.com/es-es/library/vstudio/fx6bk1f4%28v=vs.100%29.aspx.
38. *MSDN.Microsoft Visual Studio 2010.*; Available from: http://social.msdn.microsoft.com/Search/en-US?query=visual%20studio%202010&ac=2.
39. Pérez, J.E., *LibrosWeb.Introducción a JavaScript.*
40. Estelle Weyl, L.L.a.A.G., *HTML5 & CSS3 for the Real World.* First Edition ed. 2011, Collingwood, Australia: Kelly Steele y Louis Simoneau. 377.
41. Pérez, J.E., *LibrosWeb.Introducción a Ajax.*
42. Martín, J.G.R.y.F.G., *Programación Orientada a Objetos.INTRODUCCIÓN A LA PLATAFORMA .NET DE MICROSOFT*

U.d.S. Departamento de Informática y Automática, Editor: Salamanca,España.

43. Penker, H.-E.E.a.M., *Business Modeling with UML: Business Patterns at Work*. 2000, Canada: Robert Ipsen. 459.
44. Larman, C., *Fase de Analisis 1*, in *UML Y PATRONES.INTRODUCCIÓN AL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS.* , P.E.R. Vázquez, Editor. 1999: PRENTICE HALL, MÉXICO. p. 536.
45. Trilles, J.J. *Reglas de Negocio (BR) y Gestión por Procesos de Negocio (BPM)*. Grupo AuraPortal 2.
46. Sommerville, I., *Ingeniería del Software*. Séptima ed. 2005, Madrid,España. 712.
47. Vincenti, W.G., *Analytical Studies form Aeronautical History*. What Engineers Know and How They Know It. 1990.: John Hopkins University Press.
48. Robertson, S.R.a.J., *Mastering the Requirements Process*. 1999: Addison-Wesley.
49. Sommerville, I., *Requerimientos.Requerimientos del software.Requerimientos funcionles y no funcionales*, in *Ingeniería se Software*. 2005: Madrid,España. p. 712.
50. Kazman, R., *Tool Support for Architecture Analysis and Design*, in *Joint Proceedings of the ACM SIGSOFT '96 Workshops*, U.o.W. Department of Computer Science, Editor. 1996: San Francisco, CA.
51. César de la Torre Llorente, U.Z.C., Miguel Angel Ramos Barros y Javier Calvarro Nelson., *Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0(Beta)*. 2010, España. 433.
52. Larman, C., *GRASP: PATRONES PARA ASIGNAR RESPONSABILIDADES*. Segunda Edición ed. UML y Patrones.Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2002. 507.

53. MSDN. *Modelo Vista Controlador (MVC)*. [cited 2013. 20 de mayo]; Available from: http://msdn.microsoft.com/es-es/library/gg416514%28v=vs.98%29.aspx.
54. Zuñiga, M.J.J.F.C.y.I.C.B. *DIAGRAMA DE CLASES EN UML*. 6.
55. Larman, C., *Fase del diseño(I)*, in *UML Y PATRONES. INTRODUCCION AL ANALISIS Y DISEÑO ORIENTADO A OBJETOS*. 2005. p. 126.
56. Parra, E.M.y.J.D., *Guía de Patrones, Prácticas y Arquitectura .NET*, M. Corporation, Editor. 2008: United States of America.
57. Larman, C., *Fase del Diseño(1).GRASP: patrones de los principios generales para asignar responsabilidades* in *UML y Patrones: Introducción al análisis y diseño orientado a objetos*. 2004. p. 126.
58. The "Gang of Four".Erich Gamma, R.H., Ralph Johnson, John Vlissides *Design Patterns: Elements of Reusable Object-Oriented Software*. 1995, USA: Addison-Wesley. 431.
59. Larman, C., *RASP: DISEÑO DE OBJETOS CON RESPONSABILIDADES.Patrones* in *UML Y PATRONES.INTRODUCCIÓN AL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS.* . 1999: PRENTICE HALL, MÉXICO. p. 520.
60. The "Gang of Four".Erich Gamma, R.H., Ralph Johnson, John Vlissides *Memento*, D.P.E.o.R.O.-O. Software, Editor, Addison-Wesley: USA. p. 431.
61. (13 de abril de 2012) *The Repository Pattern Example in C#*.
62. César de la Torre Llorente, U.Z.C., Miguel Angel Ramos Barros y Javier Calvarro Nelson., *Sub-Capa de Repositorio (Repository pattern).Patrón Repository*, in *Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0(Beta)*. 2010: España. p. 433.
63. Date, C.J., *INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS*. SÉPTIMA ed. 2001, Naucalpan de Juárez, México: PEARSON EDUCACIÓN. 960.
64. *MSDN.Revisiones de código y estándares de codificación*. 2007; Available from: http://msdn.microsoft.com/es-es/library/aa291591%28v=VS.71%29.aspx(msdn).
65. *MSDN.Técnicas de codificación*. 2007; Available from: http://msdn.microsoft.com/es-es/library/aa291593%28v=VS.71%29.aspx.
66. *MSDN.Diagramas de componentes de UML*. 2007; Available from: http://msdn.microsoft.com/es-es/library/dd409390.aspx.
67. Daniele, I.M., *Teoría 11.El arte de modelar Año* 2007: UNRC.
68. *Programación en castellano.Introducción a UML. Diagramas de despliegue*. [cited 2010 27 de agosto]; Available from: http://www.programacion.com/articulo/introduccion_a_uml_181/7.

69. Sommerville, I., *Verificación y Validación*, in *Ingeniería del Software*. 2005: Madrid,España. p. 712.

70. Pressman, R.S., *Estrategias de Prueba del Software*, in *Ingeniería de SW. Un enfoque práctico*, McGraw-Hill, Editor. 2007: Nueva York,EUA. p. 507.

71. Benítez, C. (2011) *QUnit, testeando nuestras aplicaciones Javascript.Concepto de prueba unitaria*.

72. Pressman, R.S., *Técnicas de Prueba Parte1*, in *Ingeniería de SW. Un enfoque práctico*, McGraw-Hill, Editor. 2007. p. 507.

BIBLIOGRFÍA CONSULTADA

- Real Academia Española. [Disponible en: http://www.rare.es/rae.html.]
- Sistemas de Control de Acceso Grupo G.A.N.B. [Disponible en: http://muysimple.com/ganb].
- Tecisa 74.Soluciones.Control de accesos peatonal. [Disponible en: http://www.tecisa74.com/soluciones/ccaa_peatonal].
- Fernández, Orama, de Jesús, M. Sistema de Control de Acceso. Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba, junio 2004.
- Cruz, Oña, Lilia, T. y Zurita, Pérez, P. Sistema control de acceso a la Universidad de las Ciencias Informáticas. Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba, junio 2006.
- Rodríguez, Guzmán, E. y Viña, Fuentes, I. Módulo de Visitas para el sistema de Control de Acceso. Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba, junio 2009.
- Chiu, Blaya, G. y Fernández, Ortíz, Medellín, E. Desarrollo de un sistema para la Gestión del control de acceso la Universidad de las Ciencias Informáticas. Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba, junio 2010.
- PFLEEGER, Charles P. Security in computing. 2006. ISBN: 978-0-13-239077-4.
- Control de Acceso en Edificios. [Disponible en: http://www.scssa.com.ar/control-de-acceso-en-edificios].
- Tarjeta con código de barras. Sistema para control de asistencia y accesos LARCON-SIA. [Disponible en: http://www.larconsia.com/Tarjetas de código de barras].
- Mühlbauer Group. [Disponible en: http://www.muehlbauer.de].
- Tarjetas Personales. MasterCard. [Disponible en:http://www.mastercard.com/es/tarjetas_personales/innovacion/chip].
- Tipos de código de barras y sus ventajas. [Disponible en: http://www.informatica-hoy.com.ar/informatica-tecnologia-empresas/Tipos-de-codigo-de-barras-y-sus-ventajas].
- ¿Cómo funciona? Tarjetas identificadoras sin contacto RFID. [Disponible en: http://www.ecojoven.com/dos/03/RFID.html].
- MS9520 Voyager. [Disponible en: http://www.metrologicmexico.com/productos1/lectores_manuales/ms9520_voyager].
- IRIS CORPORATION BERHAD. [Disponible en: http://www.iris.com.my/corporate_info].
- Kimaldi. [Disponible en: http://www.kimaldi.com/].
- CSS. [Disponible en: http://www.librosweb.es/css/capitulo_1.html.

- Business-Process-Modeling-Notation. [Disponible en: http://searchcio.techtarget.com/definition/Business-Process-Modeling-Notation
- ASP.NET MVC 3, 2012. [Disponible en: http://msdn.microsoft.com/es-es/library/gg416514%28v=vs.98%29.aspx]
- Sql structured query language. [Disponible en: http://www.buenastareas.com/ensayos/Sql-Structured-Query-Language/25518185.html].

GLOSARIO DE TÉRMINOS

Acoplamiento: Dependencia entre elementos (generalmente tipos, clase y subsistemas), normalmente debido a la colaboración entre ellos para prestar un servicio. Permite mejorar la programación y el diseño de sistemas informáticos y aplicaciones.

Álgebra relacional: Es un conjunto de operaciones que describen paso a paso como computar una respuesta sobre las relaciones, tal y como éstas son definidas en el modelo relacional, es de tipo procedimental.

APIs: Sistema Avanzado de Información sobre pasajeros. Regula la prestación de un número limitado de elementos de datos (datos de identificación del pasaporte y de información de vuelo básico) de aerolíneas comerciales y operadores de embarcaciones en el sistema informático del estado de destino.

Base de Datos: Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Cálculo relacional: Es un lenguaje de consulta que describe la respuesta deseada sobre una Base de datos sin especificar como obtenerla, es de tipo declarativo.

Consulta de base de datos: Es el método para acceder a los datos en las bases de datos. Con las consultas se puede modificar, borrar, mostrar y agregar datos en una base de datos.

Dispositivo: Aparato o mecanismo que desarrolla determinadas acciones.

Efímero: Que dura poco tiempo.

Encapsular: Ocultamiento del estado, es decir, de los datos miembros de un objeto de manera que sólo se pueda cambiar mediante las operaciones definidas para ese objeto. La interacción con un objeto se realiza a través de una interfaz pública de las operaciones.

Estándar: Acuerdos (normas) documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados consistentemente como reglas, guías, o definiciones de características, para asegurar que los materiales productos, procesos y servicios se ajusten a su propósito.

Internet: Conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP.

Inteligible: Que puede ser entendido. Que se oye clara y distintamente.

Lenguaje declarativo: Es el lenguaje que especifica qué es lo que se quiere y no cómo conseguirlo.

Metodología: Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevos software.

Multifuncional: Que puede desempeñar varias funciones.

Programación declarativa: Basado en el desarrollo de programas especificando o "declarando" un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el problema y detallan su solución.

Relacional: El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos.

Software libre: Es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido el mismo puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas.

Tecnología de punta: Hace referencia a toda tecnología que fue desarrollada muy recientemente y que es de avanzada, es decir, que supone un adelanto o algo innovador respecto a los productos ya existentes.

USB: Define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores, periféricos y dispositivos electrónicos.

ANEXOS

ANEXO I Modelo de proceso de negocio "Monitoreo".

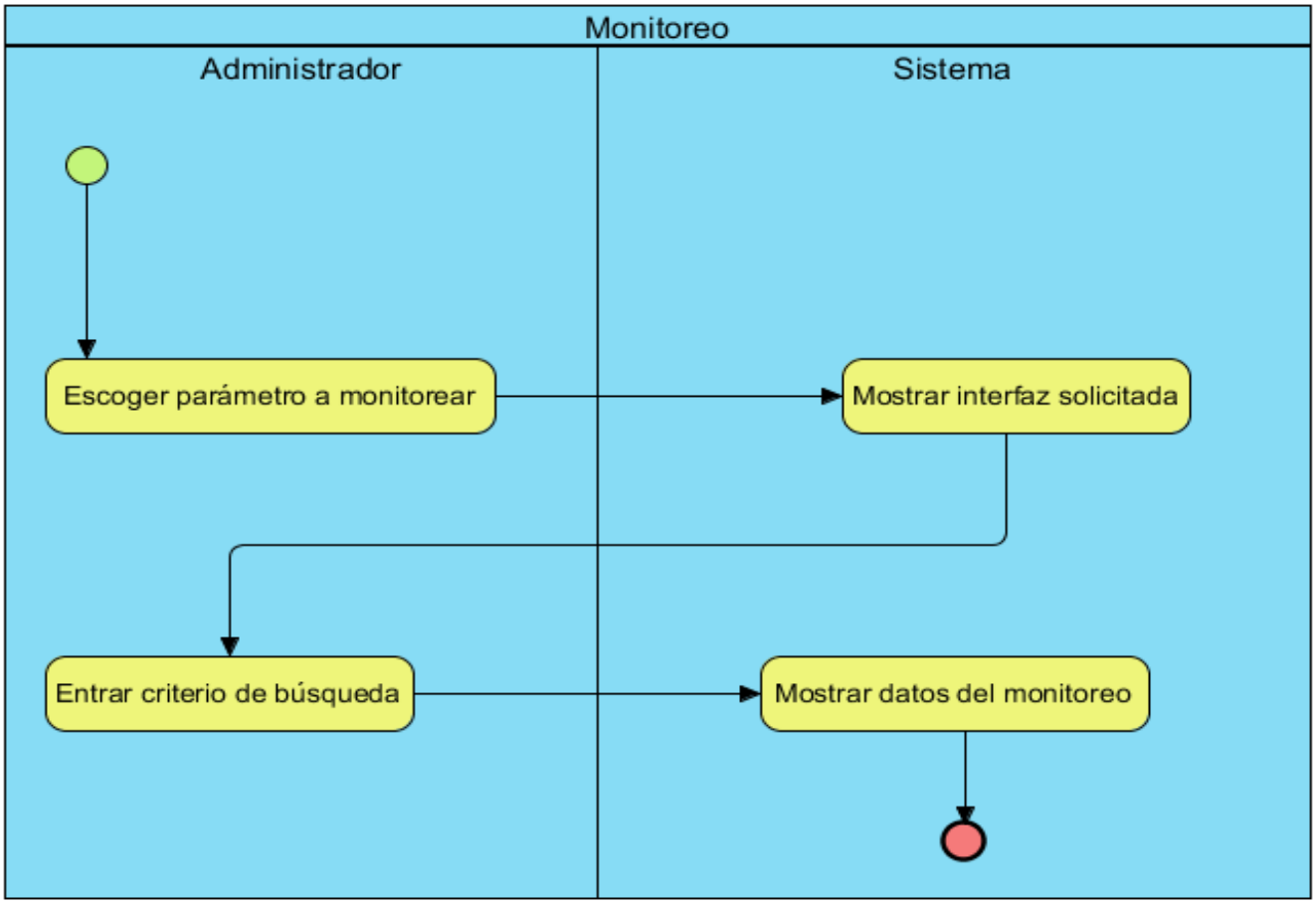


Figura A. 1Modelo de proceso de negocio "Monitoreo".

ANEXO II Descripción de las actividades del proceso Monitoreo.

Actividad	Descripción	Responsable	Entrada	Salida
Escoger parámetro a monitorear.	El Administrador escoge el parámetro que desea monitorear.	Administrador	Solicitud de monitoreo de un parámetro específico.	Solicitud de monitoreo realizada.
Mostrar interfaz solicitada.	El Sistema muestra la interfaz solicitada.	Sistema.	Solicitud de monitoreo realizada.	Interfaz mostrada.
Entrar criterio de búsqueda.	El Administrador entra el criterio de búsqueda.	Administrador	Criterio de búsqueda.	Criterio de búsqueda entrado.
Mostrar datos del monitoreo.	El Sistema muestra la información	Sistema	Criterio de búsqueda entrado.	Información mostrada.

	correspondiente al criterio de búsqueda especificado.			

Tabla A. 1 Descripción de las actividades del proceso Monitoreo.

ANEXO III Reglas del negocio.

No	Tipo	Nombre	Descripción
1	Textual.	El Agente de Seguridad deberá tener previa autorización para acceder al sistema.	El Agente de Seguridad deberá tener previa autorización para acceder al sistema.
2	Textual.	El Administrador deberá tener previa autorización para acceder al sistema.	El Administrador deberá tener previa autorización para acceder al sistema.
3	Textual.	Todo personal de la Universidad de las Ciencias Informáticas deberá portar el solapín para tener derecho a acceder a la organización.	Todo personal de la Universidad de las Ciencias Informáticas deberá portar el solapín para tener derecho a acceder a la organización.
4	Textual.	El Agente de Seguridad deberá solicitar el solapín que identifica al usuario para acceder a la organización (UCI).	El Agente de Seguridad deberá solicitar el solapín que identifica al usuario para acceder a la organización (UCI).
5	Textual.	El Agente de Seguridad solo permitirá la Entrada/Salida al solicitante cuando reciba la autorización del sistema.	El Agente de Seguridad solo permitirá la Entrada/Salida al solicitante cuando reciba la autorización del sistema.
6	Textual.	El Agente de Seguridad será responsable de cualquier acceso no autorizado al área en cuestión así como de los delitos que en consecuencia se puedan efectuar.	El Agente de Seguridad será responsable de cualquier acceso no autorizado al área en cuestión así como de los delitos que en consecuencia se puedan efectuar.
7	Textual.	El Agente de Seguridad no podrá sin previa autorización movilizarse dentro de las oficinas que componen la institución.	El Agente de Seguridad no podrá sin previa autorización movilizarse dentro de las oficinas que componen la institución.
8	Textual.	La persona que se marca como entrada a la institución, la próxima vez	La persona que se marca como entrada a la institución, la próxima vez solo se puede marcar

		solo se puede marcar como de salida.	como de salida.
9	Textual.	La persona que se marca como de salida a la institución, la próxima vez solo se puede marcar como de entrada.	La persona que se marca como de salida a la institución, la próxima vez solo se puede marcar como de entrada.
10	Textual.	Solo se podrá registrar un activo a un usuario válido de la institución (entiéndase persona con identificación propia, no robada).	Solo se podrá registrar un activo a un usuario válido de la institución (entiéndase persona con identificación propia, no robada).
11	Textual.	Los activos serán registrados y verificados a la entrada.	Los activos transportados por los usuarios de la institución (estudiantes, trabajadores y profesores) serán notificados en el momento en que este quiera entrar a la institución.
12	Textual.	El activo que se marca como entrada a la institución, la próxima vez solo se puede marcar como de salida.	El activo que se marca como entrada a la institución, la próxima vez solo se puede marcar como de salida.
13	Textual.	El activo que se marca como de salida a la institución, la próxima vez solo se puede marcar como de entrada.	El activo que se marca como de salida a la institución, la próxima vez solo se puede marcar como de entrada.
14	Relación.	Se prohíbe a los usuarios prestar su solapín de identificación a otros usuarios de la organización (UCI).	Se prohíbe a los usuarios prestar su solapín de identificación a otros usuarios de la organización (UCI).

Tabla A. 2 Reglas del negocio.

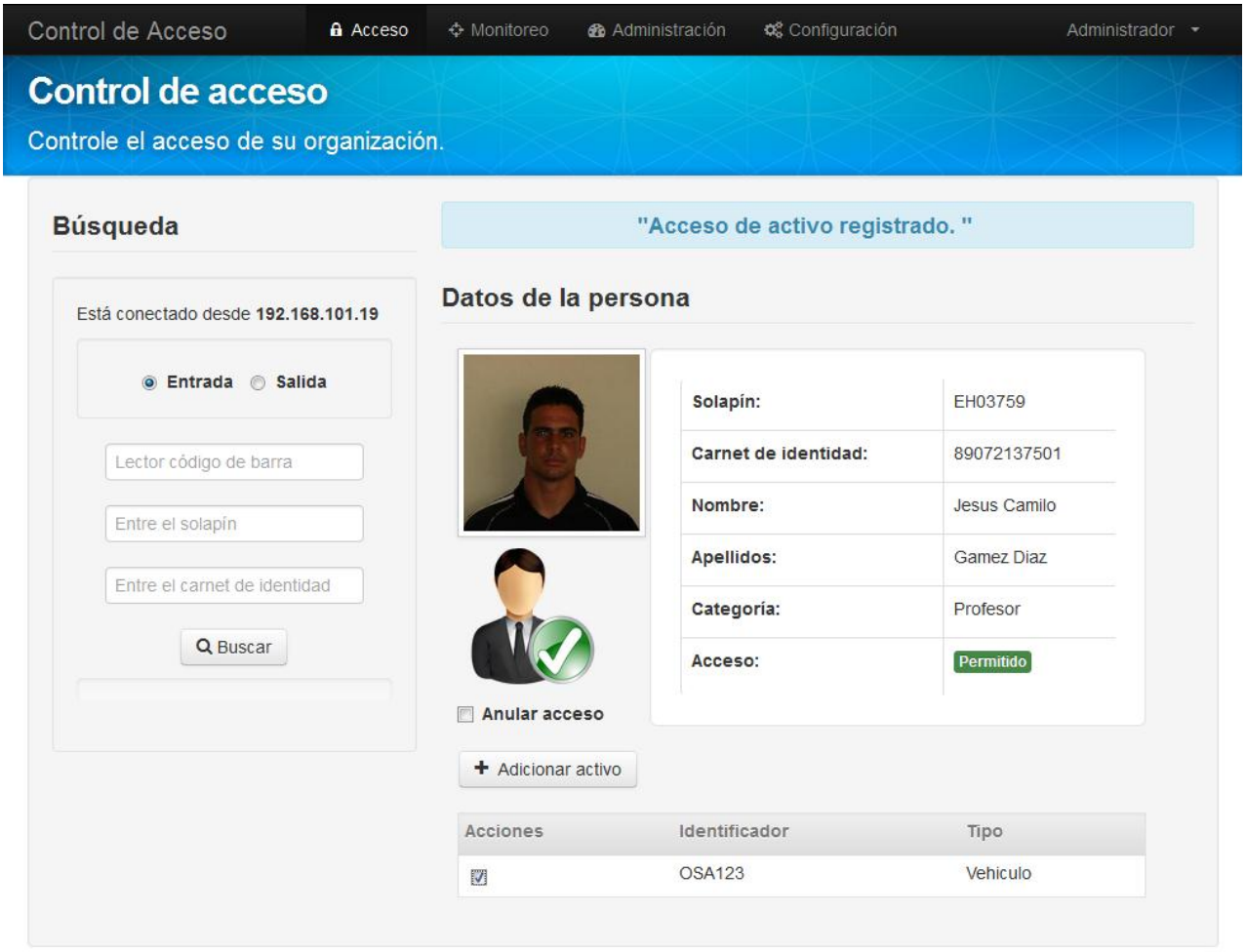
ANEXO IV Descripción de las funcionalidades.

Controlar acceso de activos

Precondiciones	El usuario debe estar autenticado con el rol Agente de Seguridad.
Funcionalidades asociadas	C5, C6,C7
Conceptos tratados	Acceso, Activos
Descripción básica	1. Si la persona tiene acceso y posee activos, entonces se procede al control de los mismos. El sistema muestra los activos

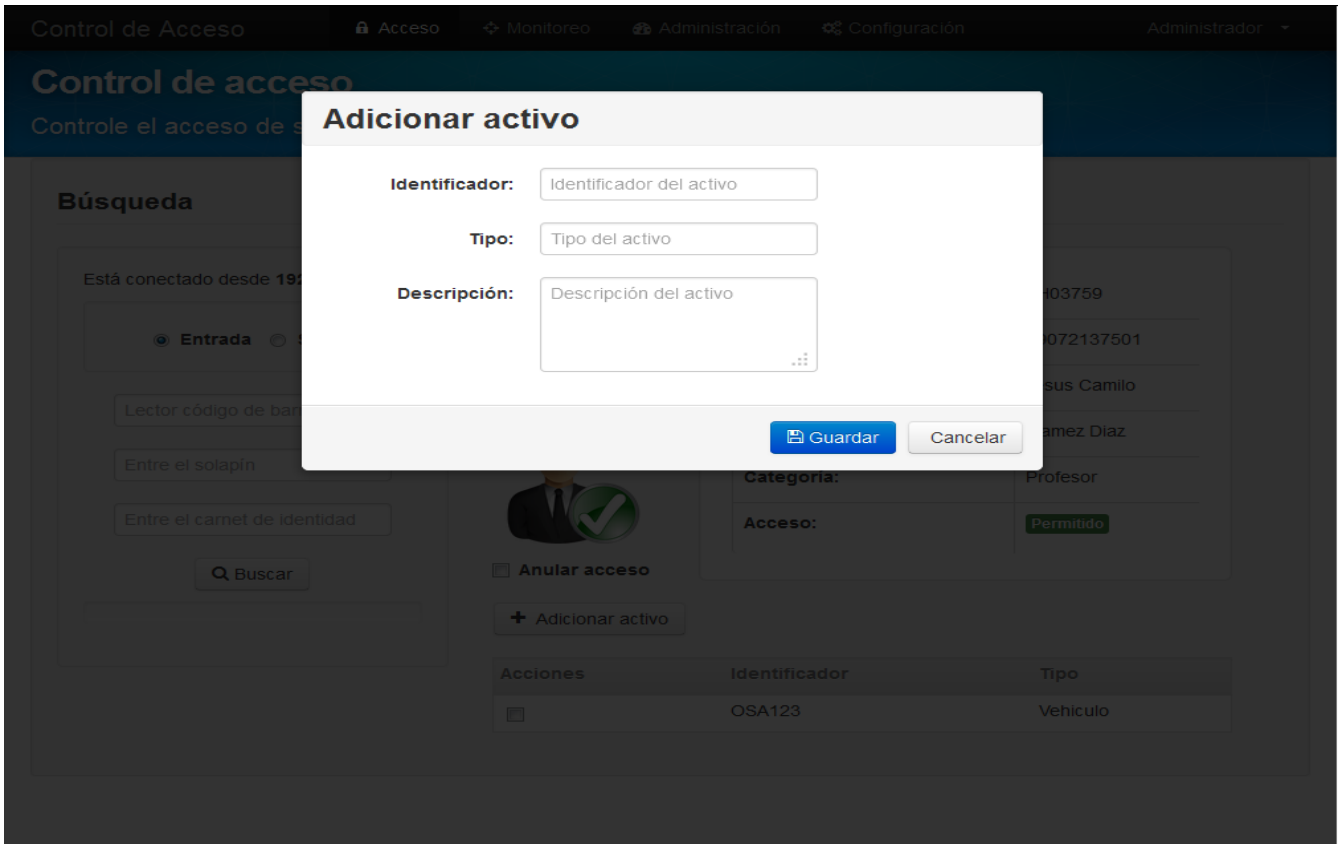
	asociados con la persona y se seleccionarán los que posea a la hora de acceder. 2. En caso que la persona esté efectuando un acceso de entrada y posee un activo que no esté registrado en el sistema *(Ver descripción alterna 1).*

Prototipos



Descripción alterna 1

Descripción alterna	Se procede a registrar el activo. Se presiona el botón Adicionar activo donde se muestra una venta con los datos necesarios para crear un activo

	nuevo. **Datos a mostrar:** Identificador Tipo Descripción
Prototipos 	
Validaciones	Solo se registra el activo a la entrada. Los datos del activo deben tener el formato correcto.
Postcondiciones	Queda registrado el acceso del activo.

Requisito no funcional	-
Servicios	-
Componente	-

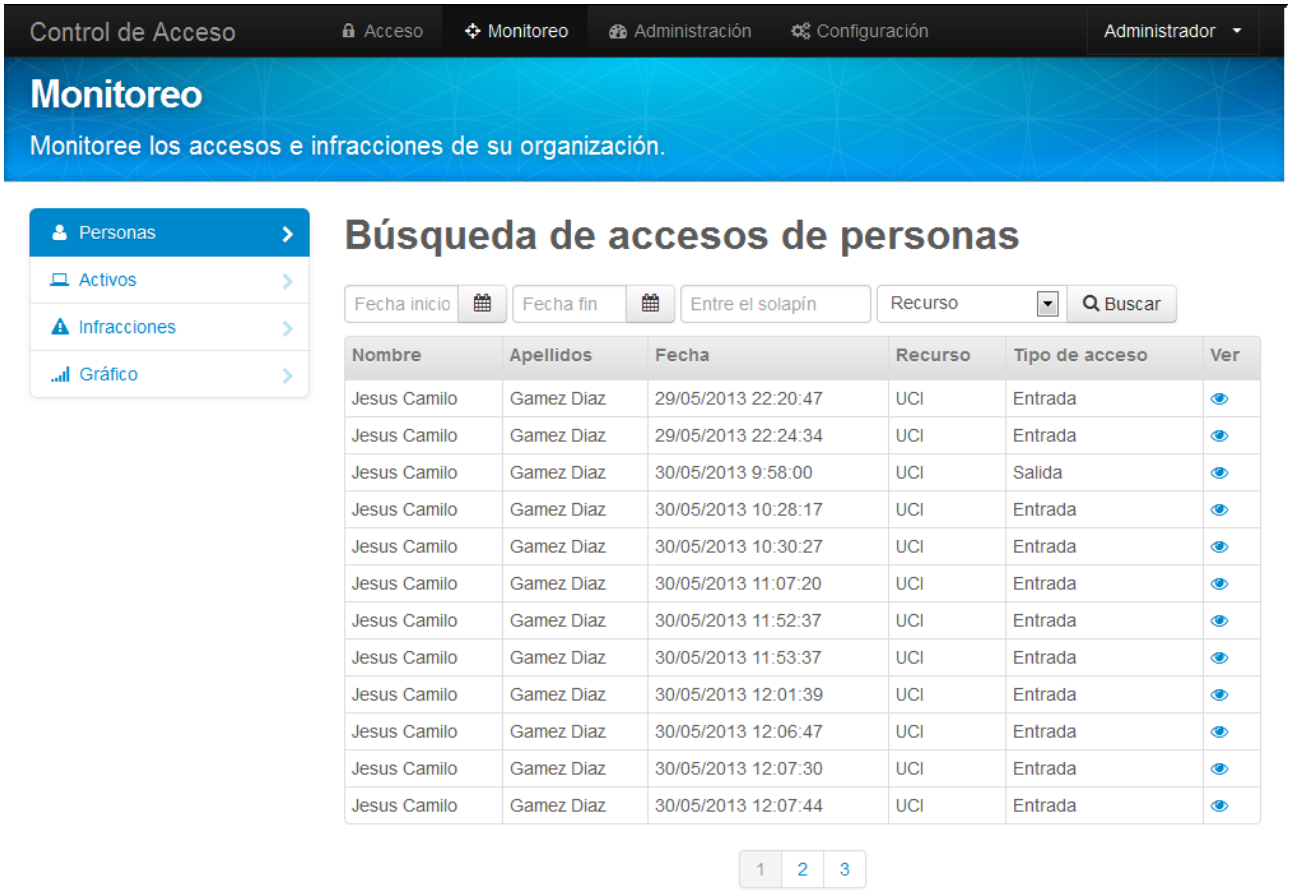
Tabla A. 3 Descripción de las funcionalidades.

Monitorear el acceso de personas

Precondiciones	El usuario debe estar autenticado con el rol Administrador.
Funcionalidades asociadas	C8,C9,C10,C11,C12
Conceptos tratados	Monitoreo, Accesos
Descripción básica	1. Si el usuario autenticado selecciona la opción Monitoreo y luego del menú lateral la opción Personas, el sistema muestra una interfaz que contiene opciones para entrar el criterio de búsqueda: • Criterio de búsqueda: - Fecha inicial (rango inicial para realizar la búsqueda de los accesos). - Fecha final (rango final para realizar la búsqueda de los accesos). - Solapín (opción donde se introduce el número del solapín de la persona para su búsqueda). - Recurso (área por la cual la persona accede). 2. El usuario presiona el botón Buscar y se muestra una tabla que contendrá los siguientes datos: • Nombre • Apellidos • Fecha • Recurso

	• Tipo de acceso(entrada, salida) • Opción Ver (muestra detalles del acceso). *(Ver descripción alterna 1)*

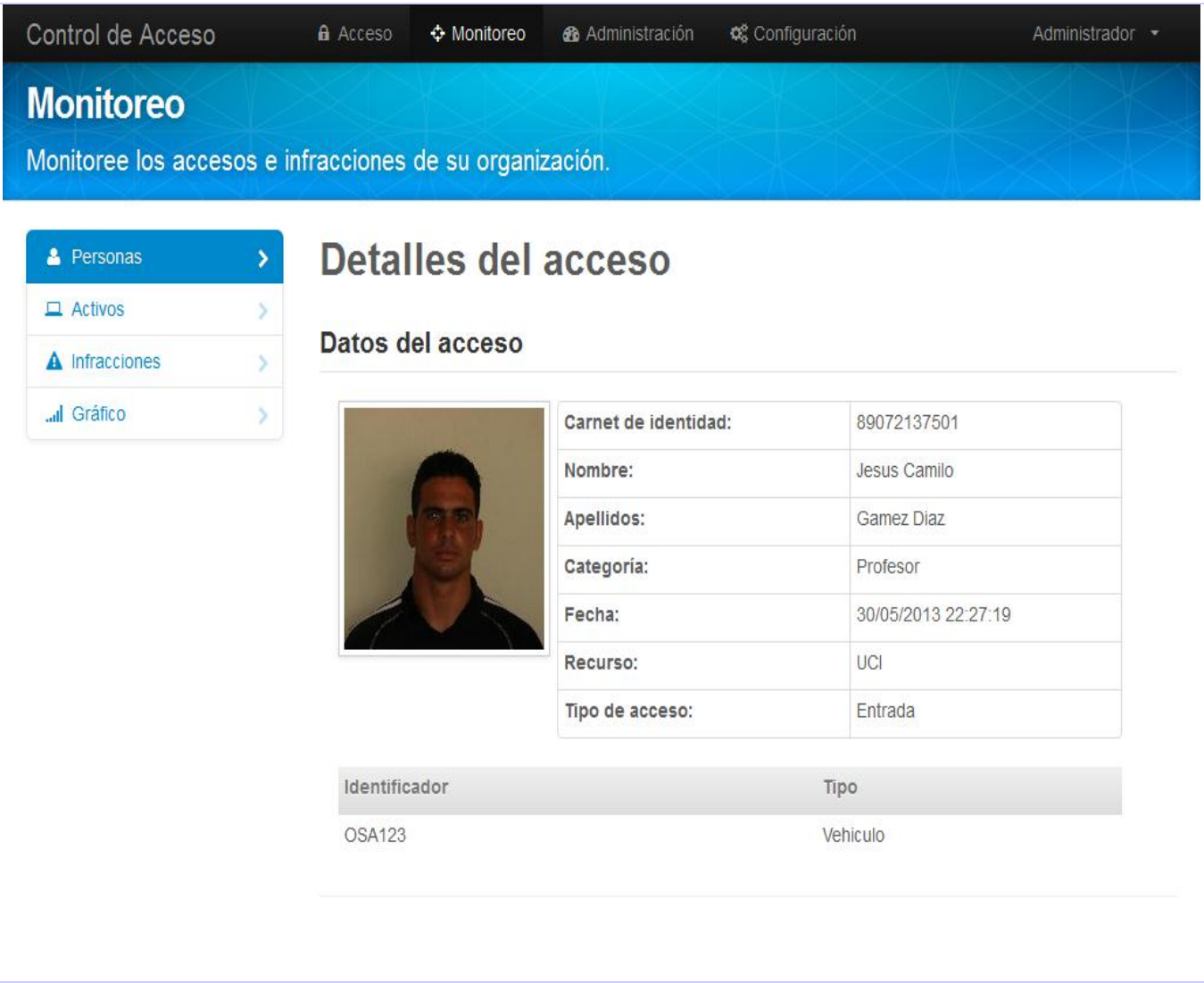
Prototipos



Descripción alterna 1

Descripción alterna	Si el usuario selecciona la opción ver detalles del acceso se muestran los datos del mismo. **Datos mostrados:**

	Foto de la persona que realizo el acceso. Carnet de identidad. Nombre Apellidos Categoría Fecha del acceso Recurso por donde ocurrió el acceso Tipo de acceso(entrada o salida) Tabla con los activos de la persona que realizó el acceso (Se muestra el identificador y el tipo de activo).
Prototipos	

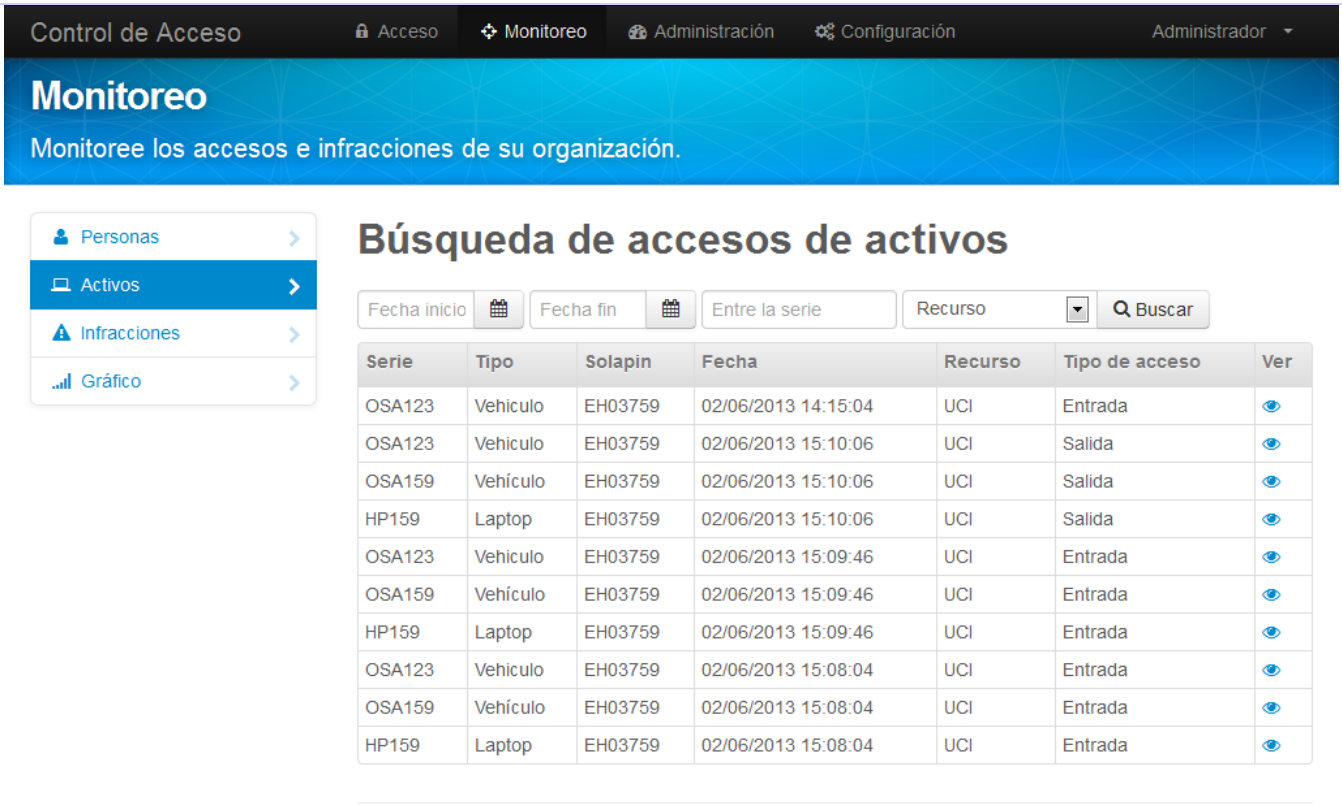


Validaciones	Para buscar los accesos se tienen que definir la fecha inicial y fecha final. La fecha inicial tiene que ser menor que la fecha final. El criterio de búsqueda debe tener el formato correcto.
Post-condiciones	Se muestran los accesos de las personas correspondientes al criterio de búsqueda.
Requisito no funcional	-
Servicios	-
Componente	-

Tabla A. 4 Monitorear el acceso de personas

Monitorear el acceso de activos

Precondiciones	El usuario debe estar autenticado con el rol Administrador
Funcionalidades asociadas	C13,C14,C15,C16
Conceptos tratados	Monitoreo, Accesos, Activos
Descripción básica	1. Si el usuario autenticado selecciona la opción Monitoreo y luego del menú lateral la opción Activos, el sistema muestra una interfaz que contiene opciones para entrar el criterio de búsqueda: • Criterio de búsqueda: - Fecha inicial (rango inicial para realizar la búsqueda de los accesos). - Fecha final (rango final para realizar la búsqueda de los accesos). - Serie (opción donde se introduce el número de serie del activo para su búsqueda). - Recurso (área por la cual el activo accede). 2. El usuario presiona el botón Buscar y se muestra una tabla que contendrá los siguientes datos: • Serie • Tipo • Solapín • Fecha • Recurso • Tipo de acceso(entrada, salida) • Opción Ver(muestra detalles del acceso)
Prototipos	



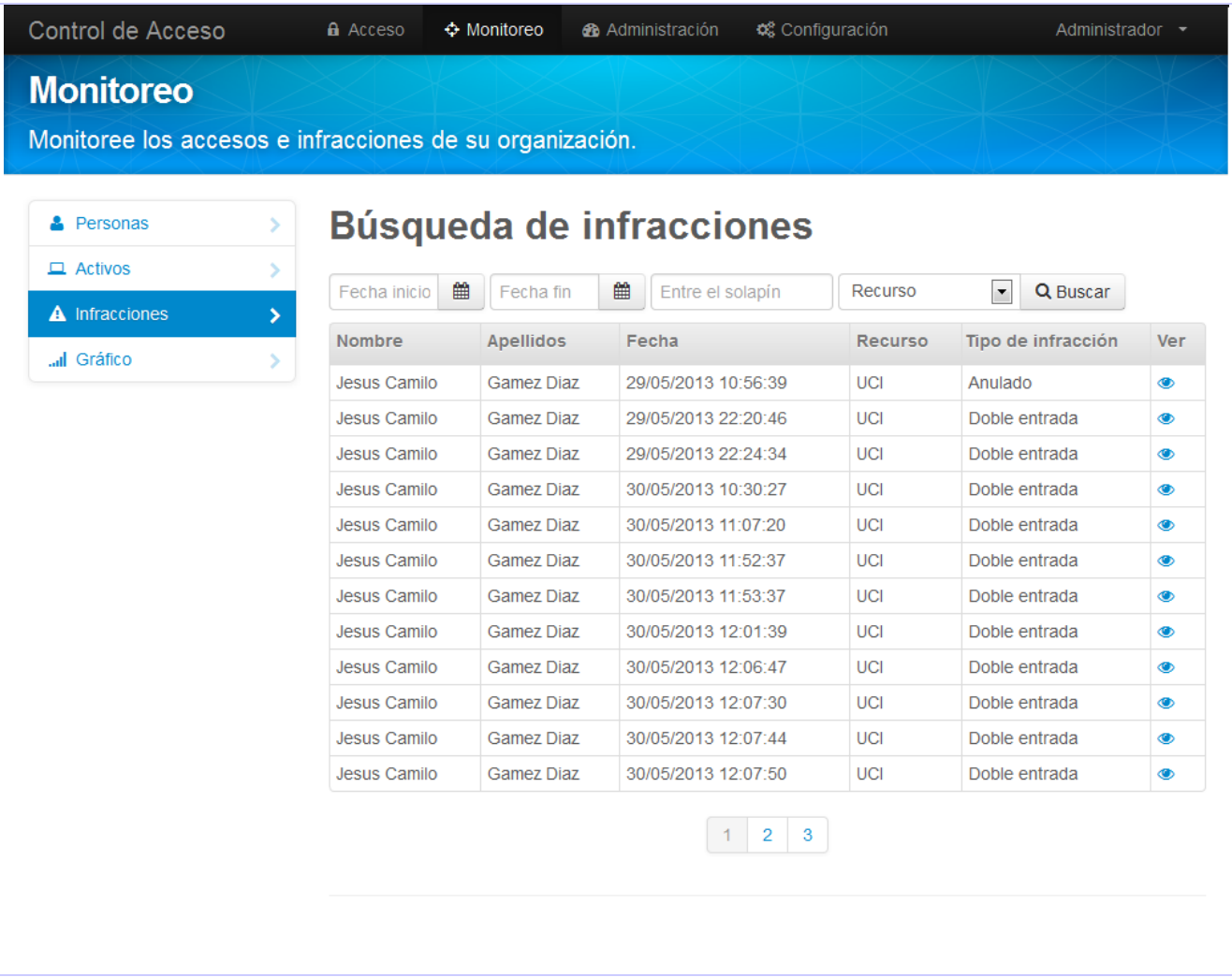
Validaciones	Para buscar los accesos se tienen que definir la fecha inicial y fecha final. La fecha inicial tiene que ser menor que la fecha final. El criterio de búsqueda debe tener el formato correcto.
Postcondiciones	Se muestran los accesos de los activos correspondientes al criterio de búsqueda.
Requisito no funcional	-
Servicios	-
Componente	-

Tabla A. 5 Monitorear el acceso de activos.

Monitorear las infracciones

Precondiciones	El usuario debe estar autenticado con el rol Administrador.
Funcionalidades asociadas	C17,C18,C19,C20,C21

Conceptos tratados	
Descripción básica	1. Si el usuario autenticado selecciona la opción Monitoreo y luego del menú lateral la opción Infracciones, el sistema muestra una interfaz que contiene opciones para entrar el criterio de búsqueda: • Criterio de búsqueda: • Fecha inicial (rango inicial para realizar la búsqueda de las infracciones). • Fecha final (rango final para realizar la búsqueda de las infracciones). • Solapín (opción donde se introduce el número del solapín de la persona para su búsqueda). • Recurso (área por la cual la persona accede) 2. El usuario presiona el botón Buscar y se muestra una tabla que contendrá los siguientes datos: • Nombre • Apellidos • Fecha • Recurso • Tipo de infracción(doble entrada, doble salida, denegado, anulado, acceso violado) • Opción Ver(muestra detalles de la infracción) *(Ver descripción alterna 1)*
Prototipos	



Descripción alterna 1

Descripción alterna	Si el usuario selecciona la opción ver detalles de la infracción se muestran los datos de la misma. **Datos mostrados:** Foto de la persona que realizo la infracción. Carnet de identidad. Nombre Apellidos Categoría

	Fecha de la infracción Recurso por donde ocurrió la infracción Tipo de acceso(entrada o salida) Tabla con los activos de la persona que realizó el acceso (muestra el identificador y el tipo de activo).
Prototipos	
Validaciones	Para buscar las infracciones se tienen que definir la fecha inicial y fecha final. La fecha inicial tiene que ser menor que la fecha final.

	El criterio de búsqueda debe tener el formato correcto.
Post-condiciones	Se muestran las infracciones asociadas al criterio de búsqueda.
Requisito no funcional	-
Servicios	-
Componente	-

Tabla A. 6 Monitorear las infracciones.

Monitorear estadísticas gráficas

Precondiciones	El usuario debe estar autenticado con el rol Administrador.
Funcionalidades asociadas	C22,C23,C24,C25,C26,C27
Conceptos tratados	Accesos, Infracciones, Gráfico, Estadísticas
Descripción básica	1. Si el usuario autenticado selecciona la opción Monitoreo y luego del menú lateral la opción Gráfico, el sistema muestra una interfaz que contiene opciones para entrar el criterio de búsqueda: • Criterio de búsqueda: - Fecha inicial (rango inicial para realizar la búsqueda). - Fecha final (rango final para realizar la búsqueda). - Solapín (opción donde se introduce el número de solapín de la persona para su búsqueda). - Recurso (área por la cual la persona accede) 2. El usuario presiona el botón Buscar y se muestra una gráfica que contendrá los siguientes datos: • Cantidad de accesos. • Cantidad de infracciones. 3. El usuario escoge la opción Ver gráfico de accesos e

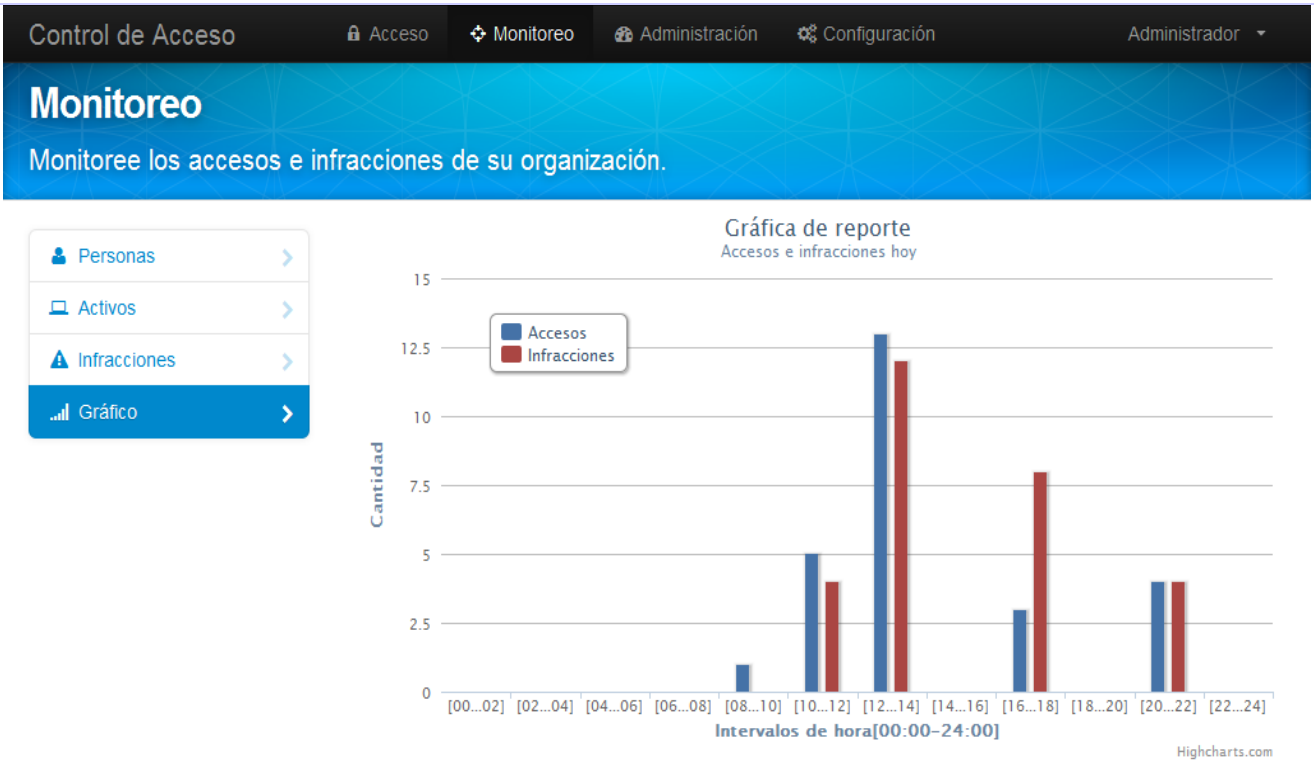
	infracciones de hoy *(Ver descripción alterna 1).* 4. El usuario escoge la opción Ver gráfico de accesos e infracciones de este año *(Ver descripción alterna 2).*

Prototipos

Descripción alterna 1

Descripción alterna	Se muestra una gráfica con los accesos e infracciones en el día por intervalos de tiempo.

Prototipos



Descripción alterna 2	
Descripción alterna	Se muestra una gráfica con los accesos e infracciones en el año por meses.
Prototipo	

Validaciones	Para buscar los datos se tienen que definir la fecha inicial y fecha final. La fecha inicial tiene que ser menor que la fecha final. El criterio de búsqueda debe tener el formato correcto.
Post-condiciones	Se muestran la cantidad accesos e infracciones asociadas al criterio de búsqueda.
Requisito no funcional	-
Servicios	-
Componente	-

Tabla A. 7 Monitorear estadísticas gráficas.

ANEXO V Planificación de la solución.

Iteración	Módulo	Característica	Diseño	Implementación
1	Control de acceso.	C1. Registrar acceso de personas.	29/10/2012-1/11/2012	2/11/2012-6/11/2012
		C2. Adicionar activo.	7/11/2012-9/11/2012	10/11/2012-14/11/2012
		C3.Anular acceso.	15/11/2012-17/11/2012	18/11/2012-22/11/2012
		C4.Controlar acceso de activo.	23/11/2012-25/11/2012	26/11/2012-30/11/2012
2	Monitoreo.	C5.Mostrar datos de accesos de personas en un rango de fecha.	1/12/2012-4/12/2012	5/12/2012-8/12/2012
		C6.Mostrar datos de accesos de personas por un punto de control en un rango de fecha.	9/12/2012-13/12/2012	14/12/2012-17/12/2012
		C7.Mostrar datos de accesos de una persona en un rango de fecha.	18/12/2013-20/12/2013	21/12/2012-23/12/2012
		C8.Mostrar datos de accesos de una persona por un punto de control en un rango de fecha.	3/1/2013-5/1/2013	6/1/2013-9/1/2013
		C9.Mostrar datos de accesos de activos en un rango de fecha.	10/1/2013-13/1/2013	14/1/2013-17/1/2013
		C10. Mostrar datos de accesos de activos por un punto de control en un rango de fecha.	18/1/2013-20/1/2013	21/1/2013-24/1/2013
		C11.Mostrar datos de accesos de activos de una persona en un	25/1/2013-	28/1/2013-2/2/2013

		rango de fecha.	27/1/2013	
		C12.Mostrar datos de accesos de un activo por un punto de control en un rango de fecha.	3/2/2013-6/2/2013	7/2/2013-10/2/2013
		C13.Mostrar infracciones en un rango de fecha.	11/2/2013-14/2/2013	15/2/2013-17/2/2013
		C14.Mostrar infracciones para una persona en un rango de fecha.	18/2/2013-20/2/2013	21/2/2013-24/2/2013
		C15.Mostrar infracciones por un punto de control en un rango de fecha.	25/2/2013-28/2/2013	29/3/2013-2/3/2013
		C16.Mostrar infracciones para una persona por un punto de control en un rango de fecha.	3/3/2013-5/3/20313	6/3/2013-9/3/2013
		C17.Mostrar gráfica con cantidad de accesos e infracciones en el día.	10/3/20313-14/3/2013	15/3/20313-19/3/2013
		C19.Mostrar gráfica con cantidad de accesos e infracciones por criterio.	28/3/20313-31/3/2013	1/4/2013-4/4/2013
		C20.Ver detalles de accesos.	5/4/2013-8/4/2013	10/4/2013-14/4/2013
		C21.Ver detalles de infracciones.	15/4/2013-20/4/2013	21/4/2013- 27/4/2013

Tabla A. 8 Planificación de la solución.

ANEXO VI Descripción de las clases del sistema.

Clase	Método	Breve descripción
ActivoRepository	*Add*Activo(activo)	Adiciona el activo a la base de datos.

	*Delete*Activo(activo)	Elimina el activo de la base de datos.
	*Update*Activo(activo)	Actualiza el activo de la base de datos.
	*GetAll*Activos()	Se obtienen todos los activos de la base de datos.
	AsignarAccesos(activo)	Se actualizan los accesos del activo .
	AsignarPersona(activo)	Se actualiza la persona poseedora del activo.
	CrearRelacionActivoAcceso(activo, acceso)	Se crea la relación del activo con el acceso.
	CrearRelacionActivoPersona(activo, persona)	Se crea la relación del activo con la persona.

Tabla A. 9 Descripción de las clases del sistema.

Clase	Método	Breve descripción
RecursoRepository	AddRecurso(recurso)	Adiciona el recurso a la base de datos.
	DeleteRecurso (recurso)	Elimina el recurso de la base de datos.
	UpdateRecurso (recurso)	Actualiza el recurso de la base de datos.
	GetAllRecursos()	Se obtienen todos los recursos de la base de datos.
	AsignarAccesos(recurso)	Se actualizan los accesos del recurso.

	AsignarInfracciones(recurso)	Se actualiza las infracciones del recurso.
	AsignarRestricciones(recurso)	Se actualiza las restricciones del recurso.
	AsignarGrupos(recurso)	Se actualiza los grupos del recurso.
	AsignarRoles(recurso)	Se actualiza los roles del recurso.
	AsignarRecursosHijos(recurso)	Se actualiza los recursos hijos del recurso.
	AsignarRecursoPadre(recurso)	Se actualiza el recurso padre del recurso.
	AsignarTipoRecurso(recurso)	Se actualiza el tipo de recurso del recurso.
	CrearRelacionRecursoInfraccion(recurso, infraccion)	Se crea la relación del recurso con la infracción.
	CrearRelacionRecursoAcceso(recurso, acceso)	Se crea la relación del recurso con el acceso.
	CrearRelacionRecursoRestriccion(recurso, restriccion)	Se crea la relación del recurso con la restricción.
	CrearRelacionRecursoGrupo(recurso, grupo)	Se crea la relación del recurso con el grupo.
	CrearRelacionRecursoRol(recurso, rol)	Se crea la relación del recurso con el rol.

	CrearRelacionRecursoPadreHijo(recurso, recursoHijo)	Se crea la relación del recurso con el recurso hijo.
	CrearRelacionRecursoHijoPadre(recurso, recursoPadre)	Se crea la relación del recurso con el recurso padre.
	CrearRelacionRecursoTipoRecurso(recurso, tipoRecurso)	Se crea la relación del recurso con el tipo de recurso.
	CrearRelacionRecursoDispositivo(recurso, dispositivo)	Se crea la relación del recurso con el dispositivo.
	AsignarDispositivo(recurso)	Se actualizan los dispositivos del recurso.

Tabla A. 10 Descripción de la clase "RecursoRepository".

Clase	Método	Breve descripción
AccesoRepository	AddAcceso(acceso)	Adiciona el acceso a la base de datos.
	DeleteAcceso(acceso)	Elimina el acceso de la base de datos.
	UpdateAcceso(acceso)	Actualiza el acceso de la base de datos.
	GetAllAccesos()	Se obtienen todos los accesos de la base de datos.
	AsignarActivos(acceso)	Se actualizan los activos del acceso.
	AsignarPersona(acceso)	Se actualiza la persona correspondiente al acceso.
	AsignarRecurso(acceso)	Se actualiza el recurso del acceso.

	CrearRelacionAccesoRecurso(acceso, recurso)	Se crea la relación del acceso con el recurso.
	CrearRelacionAccesoActivo(acceso, activo)	Se crea la relación del acceso con el activo.
	CrearRelacionAccesoPersona(acceso, persona)	Se crea la relación del acceso con la persona.

Tabla A. 11 Descripción de la clase "AccesoRepository".

Clase	Método	Breve descripción
InfraccionRepository	AddInfraccion(infraccion)	Adiciona la infracción a la base de datos.
	DeleteInfraccion(infraccion)	Elimina la infracción de la base de datos.
	UpdateInfraccion(infraccion)	Actualiza la infracción de la base de datos.
	GetAllInfracciones()	Se obtienen todas las infracciones de la base de datos.
	AsignarPersona(infraccion)	Se actualiza la persona correspondiente a la infracción.
	AsignarRecurso(infraccion)	Se actualiza el recurso de la infracción.
	CrearRelacionInfraccionRecurso(infraccion, recurso)	Se crea la relación de la infracción con el recurso.
	CrearRelacionInfraccionPersona(infraccion, persona)	Se crea la relación de la infracción con la persona.

Tabla A. 12 Descripción de la clase "InfraccionRepository".

Clase	Método	Breve descripción
PersonaRepository	AddPersona(persona)	Adiciona la persona a la base de datos.
	DeletePersona(persona)	Elimina la persona de la base de datos.
	UpdatePersona(persona)	Actualiza la persona de la base de datos.
	GetAllPersonas()	Se obtienen todas las personas de la base de datos.
	AsignarGrupos(persona)	Se actualizan los grupos correspondiente a la persona.
	AsignarAtributos(persona)	Se actualiza los atributos de la persona.
	AsignarAccesos(persona)	Se actualiza los accesos de la persona.
	AsignarActivos(persona)	Se actualiza los activos de la persona.
	AsignarInfracciones(persona)	Se actualiza las infracciones de la persona.
	AsignarRoles(persona)	Se actualiza los roles de la persona.
	CrearRelacionPersonaGrupo(persona, grupo)	Se crea la relación de la persona con el grupo.
	CrearRelacionPersonaAtributo(persona, atributo)	Se crea la relación de la persona con el atributo.
	CrearRelacionPersonaAcceso(persona, acceso)	Se crea la relación de la persona con el acceso.

	CrearRelacionPersonaActivo(persona, activo)	Se crea la relación de la persona con el activo.
	CrearRelacionPersonaInfraccion(persona, infraccion)	Se crea la relación de la persona con la infracción.
	CrearRelacionPersonaRol(persona, rol)	Se crea la relación de la persona con el rol.

Tabla A. 13 Descripción de la clase "PersonaRepository".

Clase	Método	Breve descripción
NegocioMonitoreo	BuscarAccesoPersonasFechas(fechaInicio, fechaFinal)	Busca accesos por fechas
	BuscarAccesoPersonasFechasSolapin(fechaInicio, fechaFinal, solapin)	Busca accesos por fechas y solapin
	BuscarAccesoPersonasFechaSolapinRecurso(fechaInicio, fechaFinal,solapin, nombreRecurso)	Busca accesos por fechas , solapin y recurso
	BuscarAccesoPersonasFechasRecurso(fechaInicio, fechaFinal, nombreRecurso)	Busca accesos por fechas y recurso
	BuscarInfraccionesFechas(fechaInicio,fechaFinal)	Busca infracciones por fechas
	BuscarInfraccionesFechasSolapin(fechaInicio, fechaFinal, solapin)	Busca infracciones por fechas y solapin
	BuscarInfraccionesFechasSolapinRecurso(fechaInicio, fechaFinal, solapin, nombreRecurso)	Busca infracciones por fechas, solapin y recurso
	BuscarInfraccionesFechasRecurso(fechaInicio, fechaFinal, nombreRecurso)	Busca infracciones por recurso
	BuscarAccesosActivosFechasIdentificador(fechaInicio, fechaFinal, idActivo)	Busca accesos por fechas y serie de activo

	BuscarAccesosActivosFechasRecursosIdentificador(fechaInicio, fechaFinal, idActivo, nombreRecurso)	Busca accesos por fechas , serie de activo y recurso
	BuscarAccesosActivosFechas(fechaInicio, fechaFinal)	Busca accesos que contengan activos por fechas y recurso.
	BuscarAccesoID(idAcceso)	Busca un acceso por el identificador
	BuscarInfraccionesID(idAcceso)	Busca una infracción por el identificador
	InfraccionesHoyHoras()	Devuelve un arreglo con la cantidad de infracciones en el día.
	AccesosHoyHoras()	Devuelve un arreglo con la cantidad de accesos en el día.

Tabla A. 14 Descripción de la clase "NegocioMonitoreo".

Clase	Método	Breve descripción
SP_MonitoreoActivosController	MonitoreoActivos()	Muestra la vista correspondiente al control de acceso de activo.
	BusquedaAccesosActivos(fechaInicio, fechaFin, string serie, string recurso, page = 1)	Se utiliza para el paginado.
	BusquedaAccesosActivos(datos, page = 1)	Se encarga de procesar los datos buscados y mostrar la vista con información referente al acceso de los activos.

Tabla A. 15 Descripción de la clase "SP_MonitoreoActivosController".

Clase	Método	Breve descripción
SP_MonitoreoInfraccionesController	MonitoreoInfracciones()	Muestra la vista correspondiente a las infracciones.
	BuscarInfracciones(fechaInicio, fechaFin, solapin, recurso, page=1)	Se utiliza para el paginado.

	BuscarInfracciones(FormCollection datos, int page=1)	Se encarga de procesar los datos buscados y mostrar la vista con información referente a las infracciones.

Tabla A. 16 Descripción de la clase "SP_MonitoreoInfraccionesController".

Clase	Método	Breve descripción
SP_MonitoreoPersonasController	MonitoreoPersonas()	Muestra la vista correspondiente a los accesos de las personas.
	BusquedaAccesosPersonas(fechaInicio, string fechaFin, solapin, recurso, page=1)	Se utiliza para el paginado.
	BusquedaAccesosPersonas(datos, page=1)	Se encarga de procesar los datos buscados y mostrar la vista con información referente a los accesos de las personas.

Tabla A. 17 Descripción de la clase "SP_MonitoreoPersonasController".

Clase	Método	Breve descripción
SP_GraficoController	GraficoHoy()	Muestra la vista correspondiente a los accesos e infracciones de hoy.
	GraficoAnno()	Muestra la vista correspondiente a los accesos e infracciones del año.
	ReporteGrafico()	
	ReporteGraficoAccesos(fechaInicio, fechaFin, solapin, recurso)	Se encarga de procesar los datos buscados y mostrar la vista con información referente a los accesos e infracciones según los datos entrados.

Tabla A. 18 Descripción de la clase "SP_GraficoController".

Clase	Método	Breve descripción
SP_DetallesController	DetallesAcceso(id)	Muestra los detalles correspondientes al acceso.
	DetallesInfracciones(id)	Muestra los detalles correspondientes a la infracción.

Tabla A. 19 Descripción de la clase "SP_DetallesController".

Clase	Atributos
Infracción	tipoInfraccion, fecha, habilitado, persona, recurso
Persona	Nombre, apellidos, ci, código, habilitado, estado, foto, codigoBarra, activos, infracciones, accesos, atributos, roles, grupos
Recurso	Nombre, descripcion, habilitado, tipoRecurso, grupos, roles, dispositivos, restricciones, accesos, infracciones, recursoPadre, recursosHijos
Activo	Descripcion, accesos, habilitado, serie, tipoActivo, persona
Acceso	tipoAcceso, fecha, activos, habilitado, idAcceso, persona, recurso
Relations	IdRelations, RelationType, Owner, Related
Elements	IdElements, ElementType, Data

Tabla A. 20 Descripción de los atributos de las clases.

ANEXO VII Diagramas de clases.

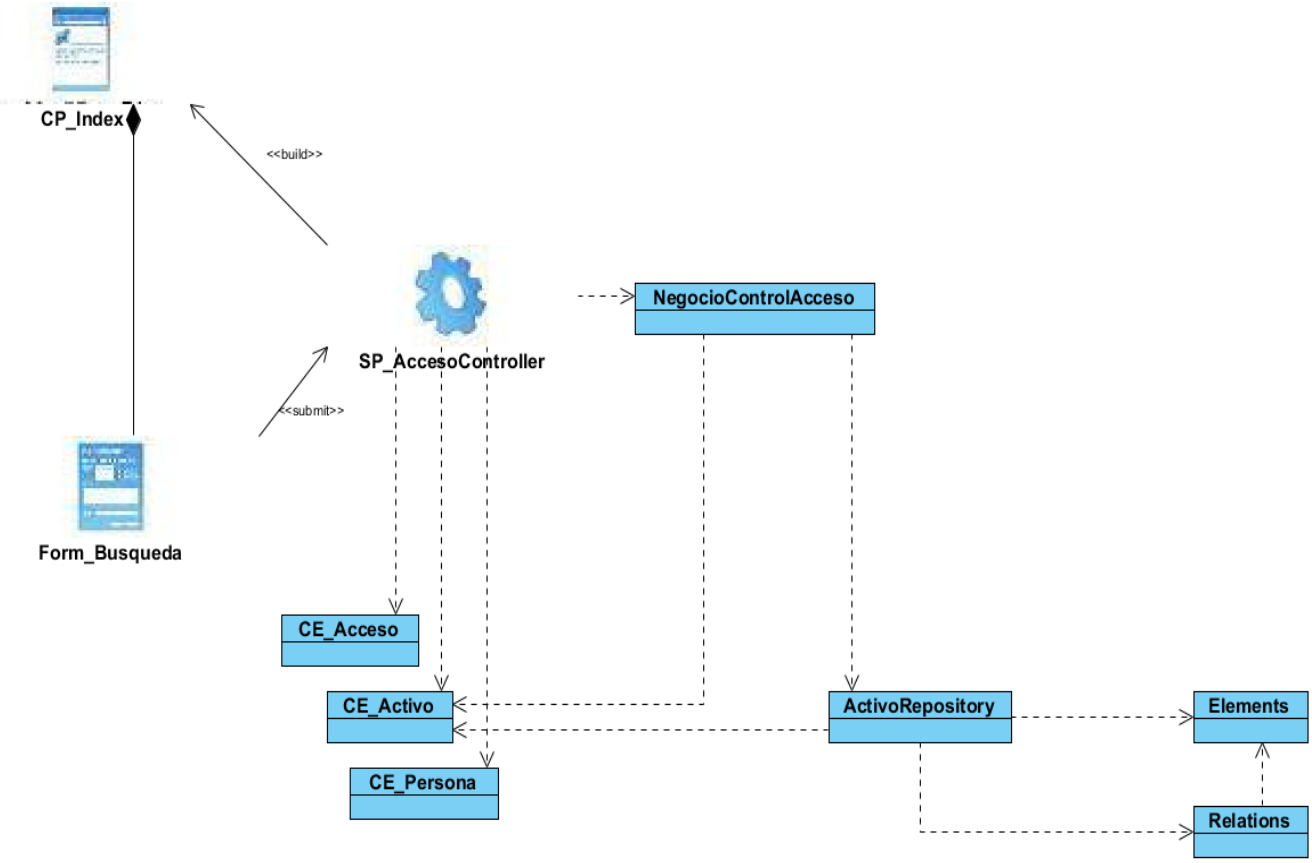


Figura A. 2 Control de acceso de activos.

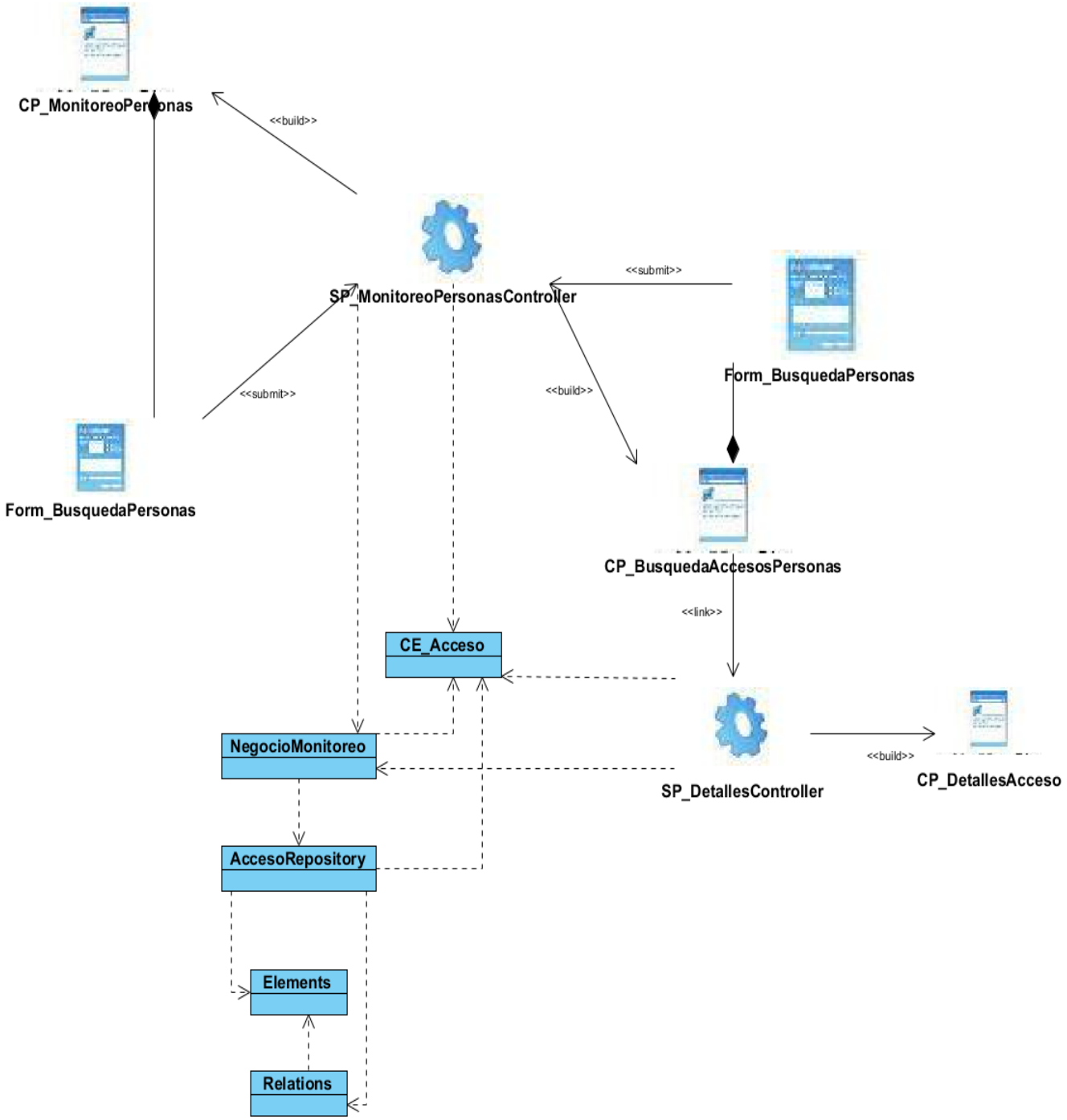


Figura A. 3 Monitoreo de personas.

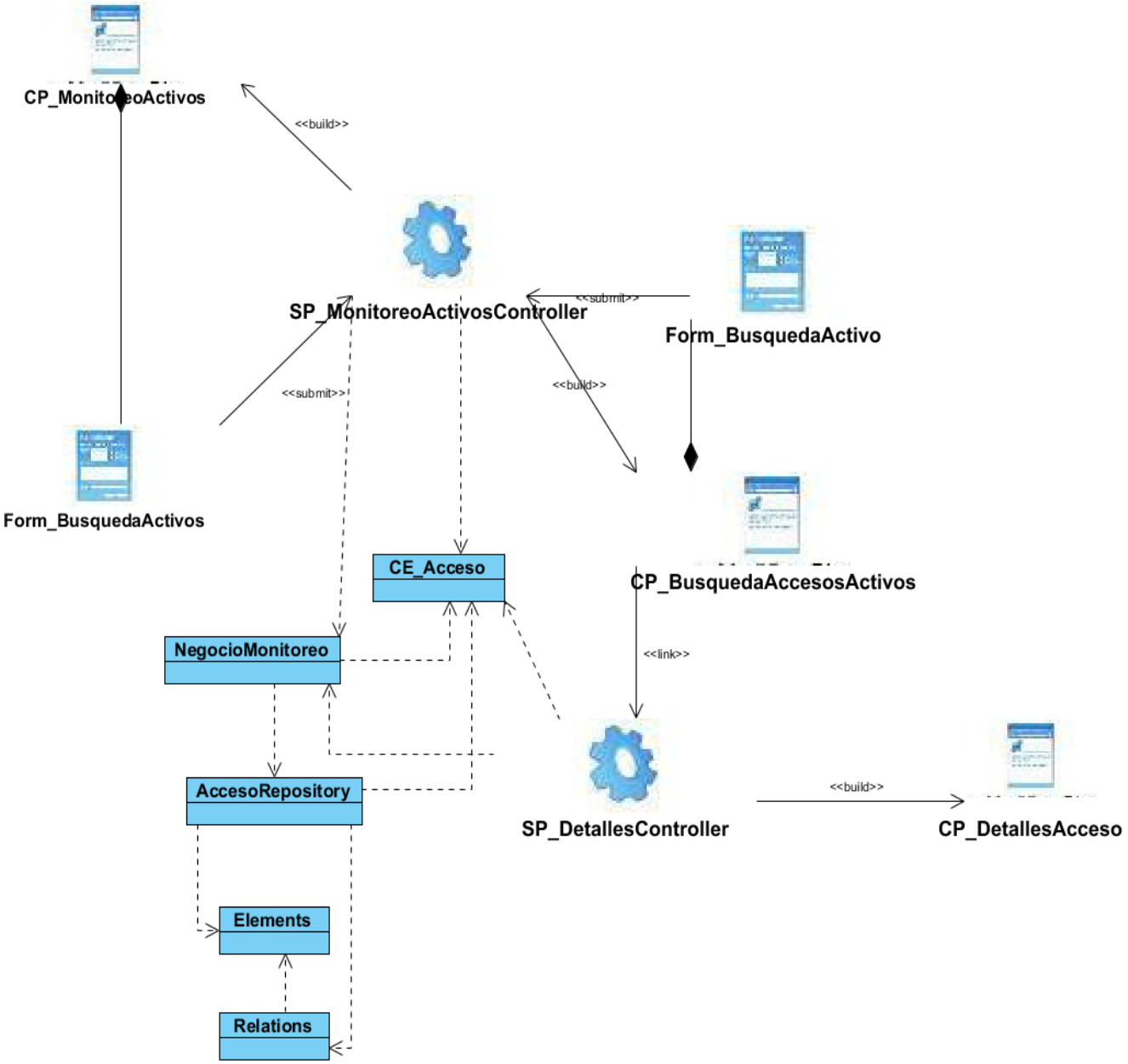


Figura A. 4 Monitoreo de activos.

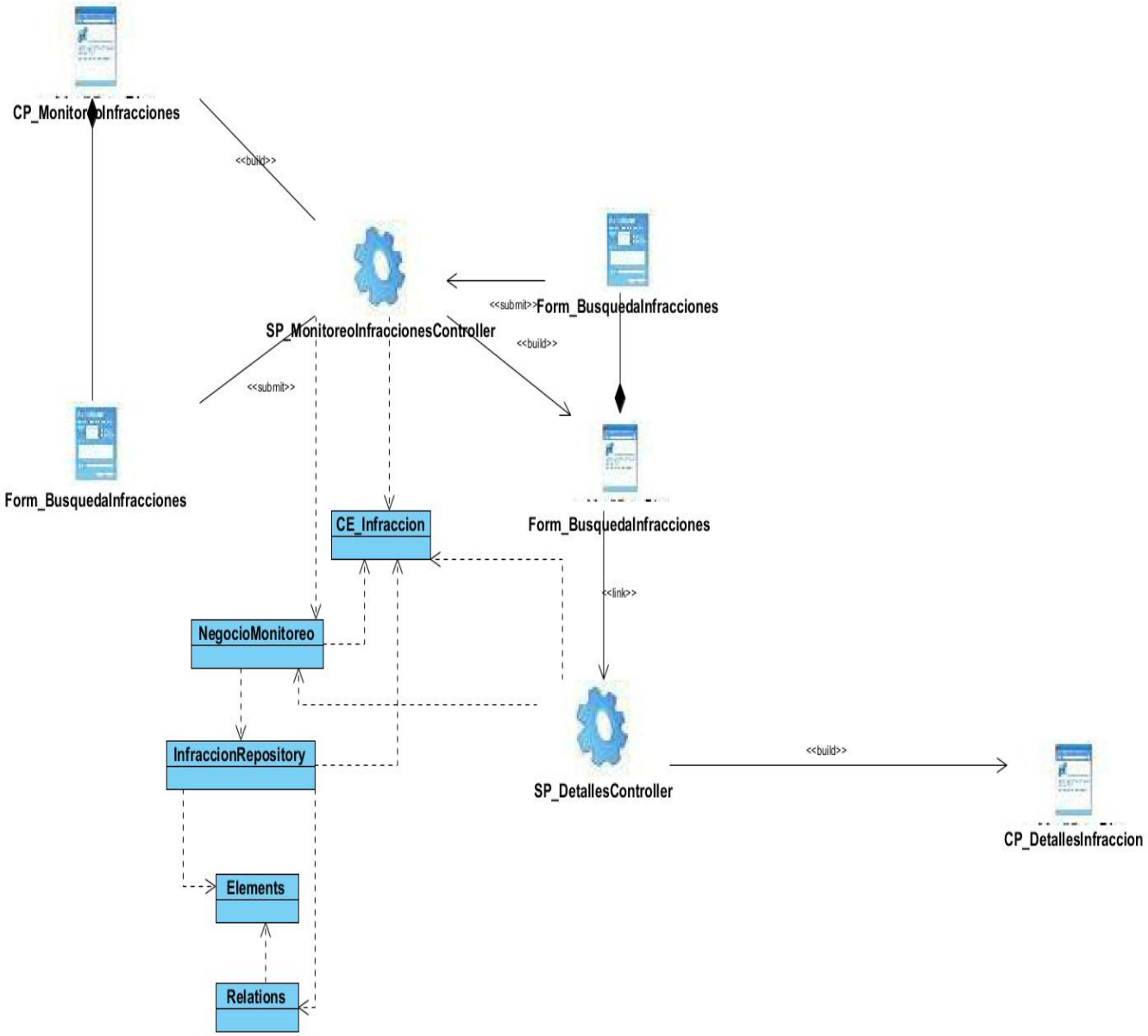


Figura A. 5 Monitoreo de infracciones.

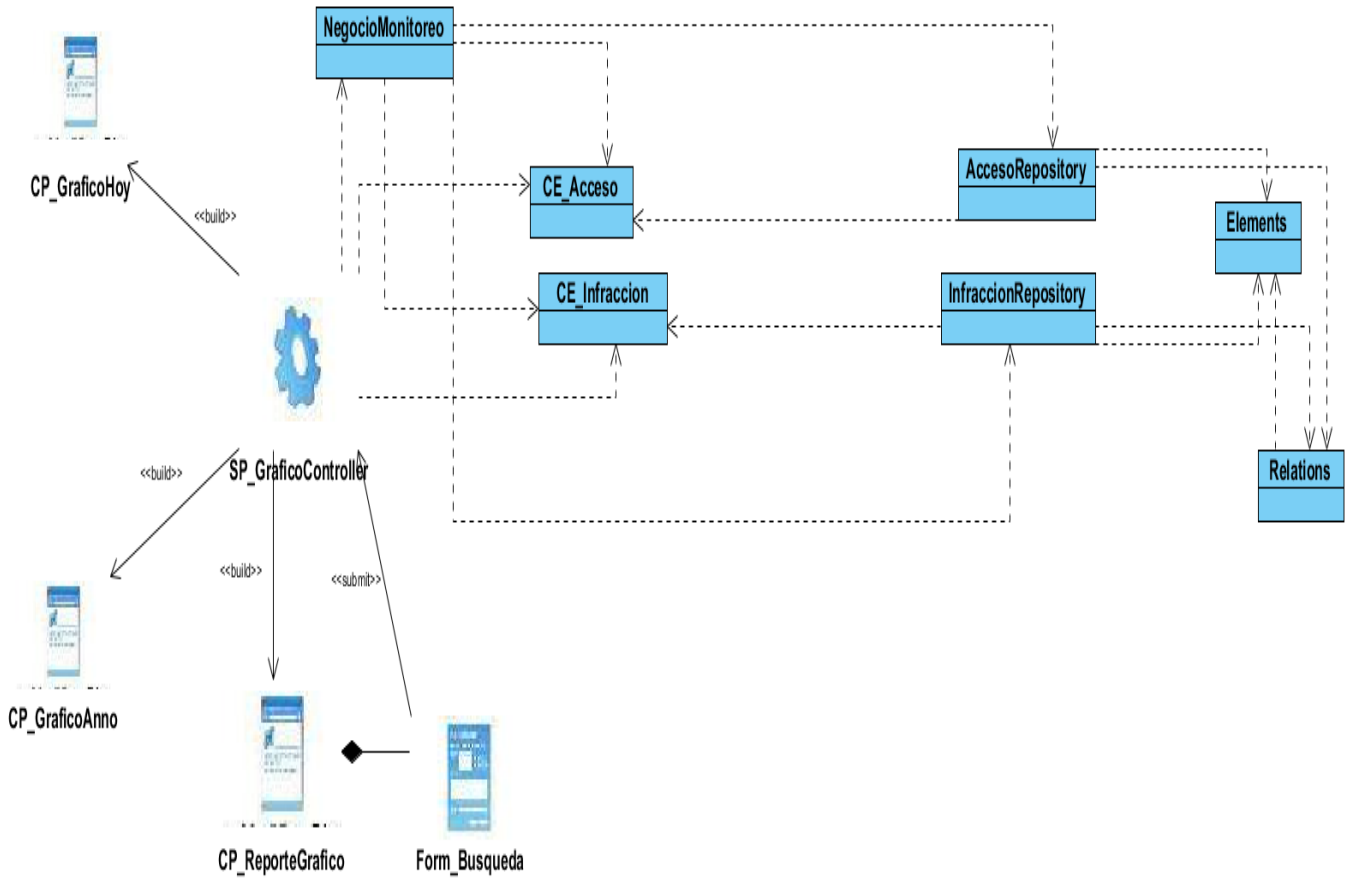


Figura A. 6 Reporte gráfico.

ANEXO VIII Diagramas de Secuencia.

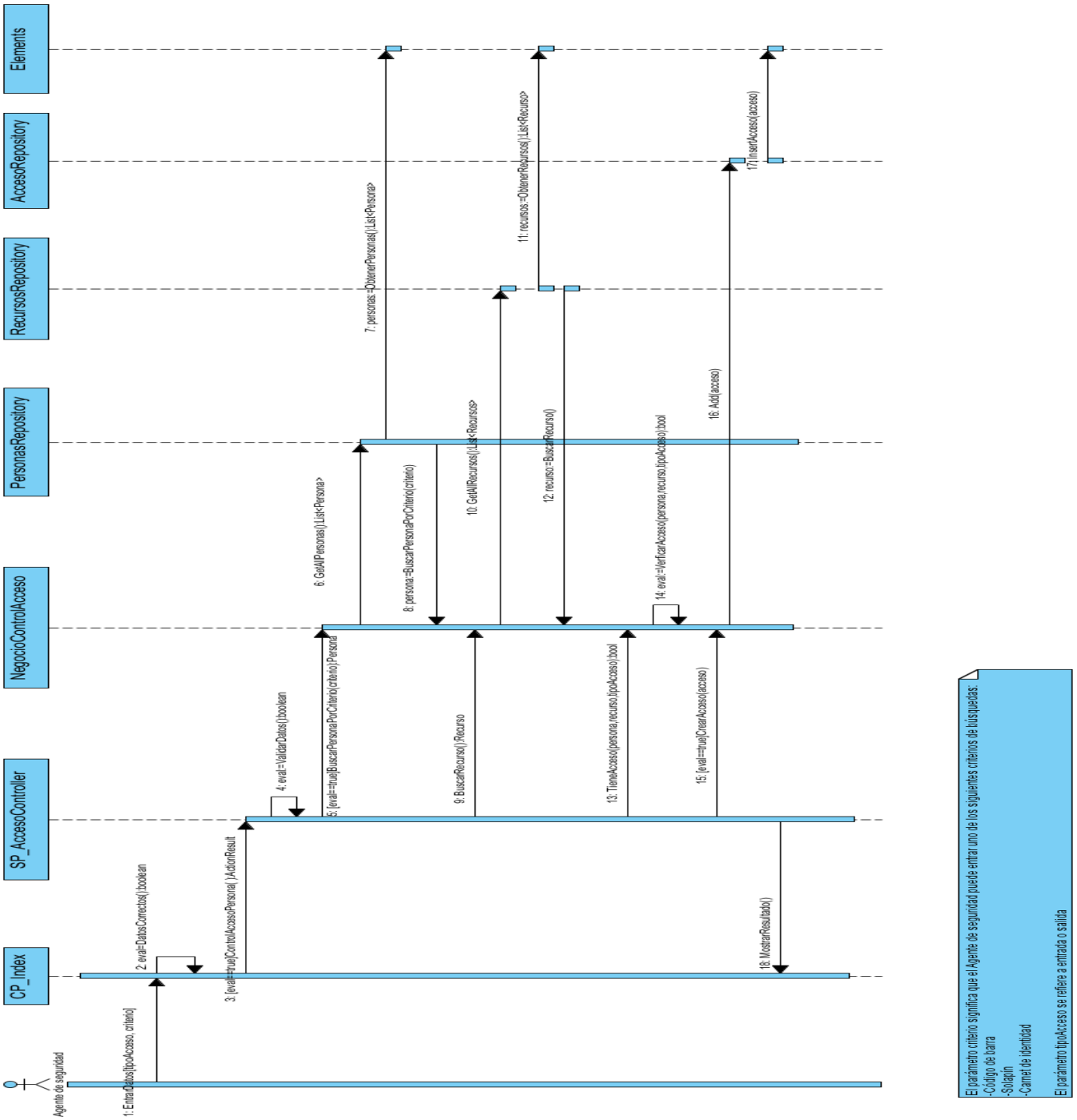


Figura A. 7 Control de acceso a personas.

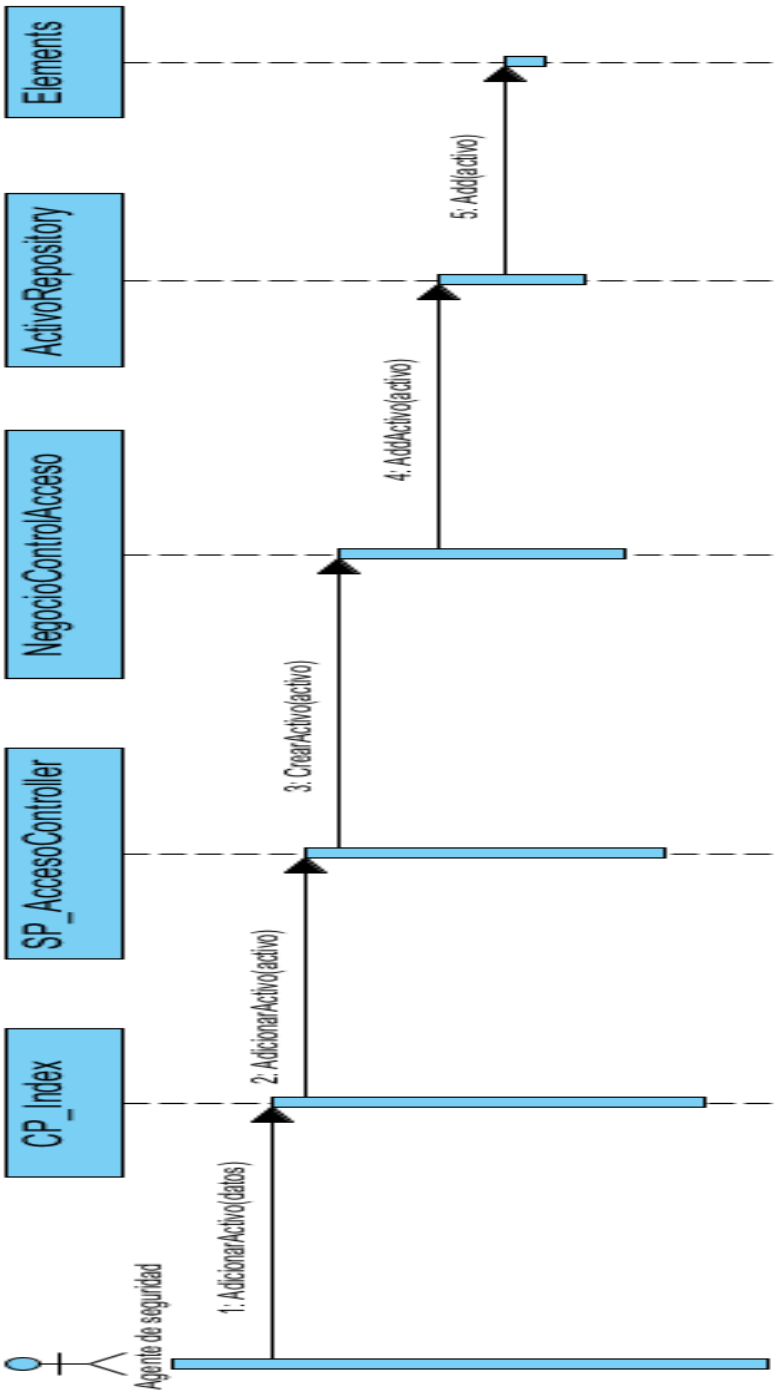


Figura A. 8 Adicionar activo.

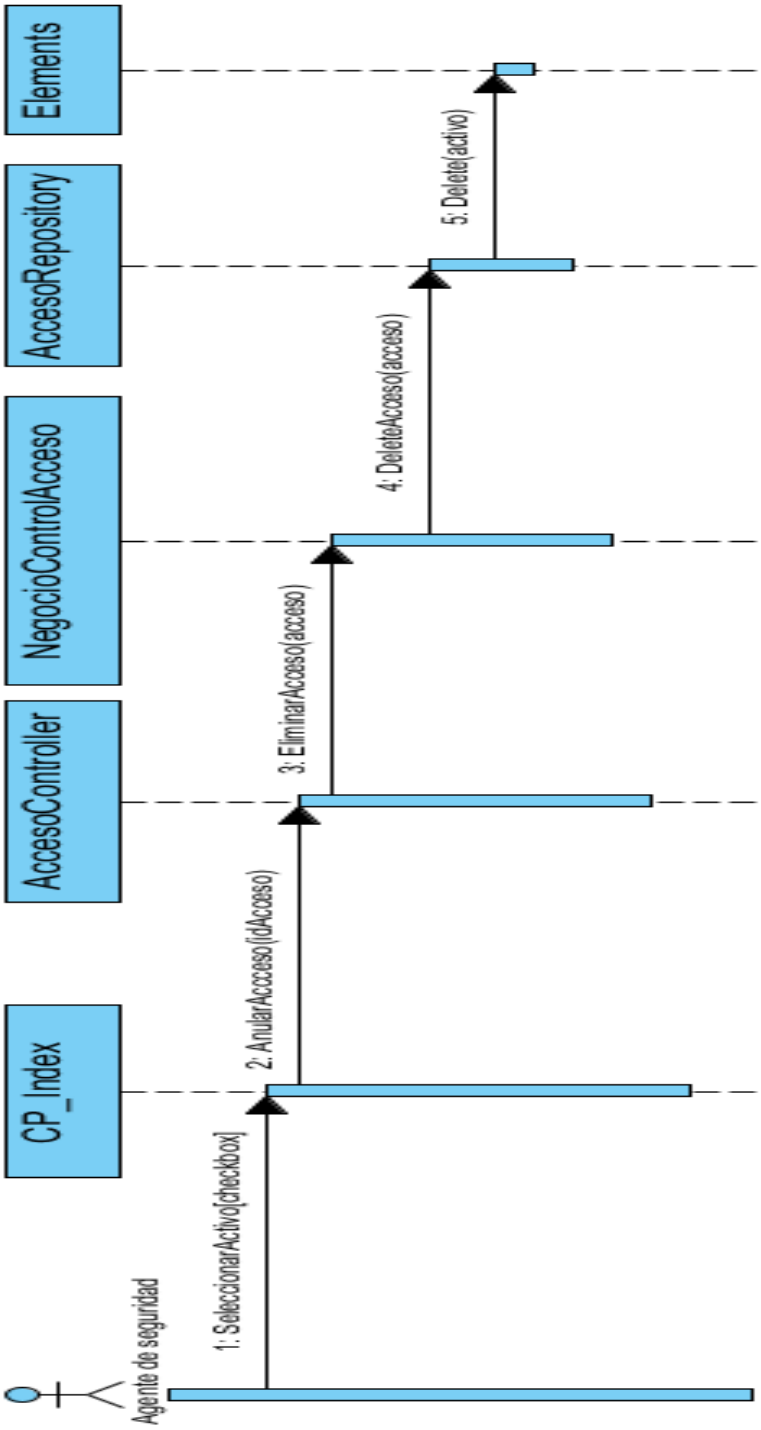


Figura A. 9 Anular acceso.

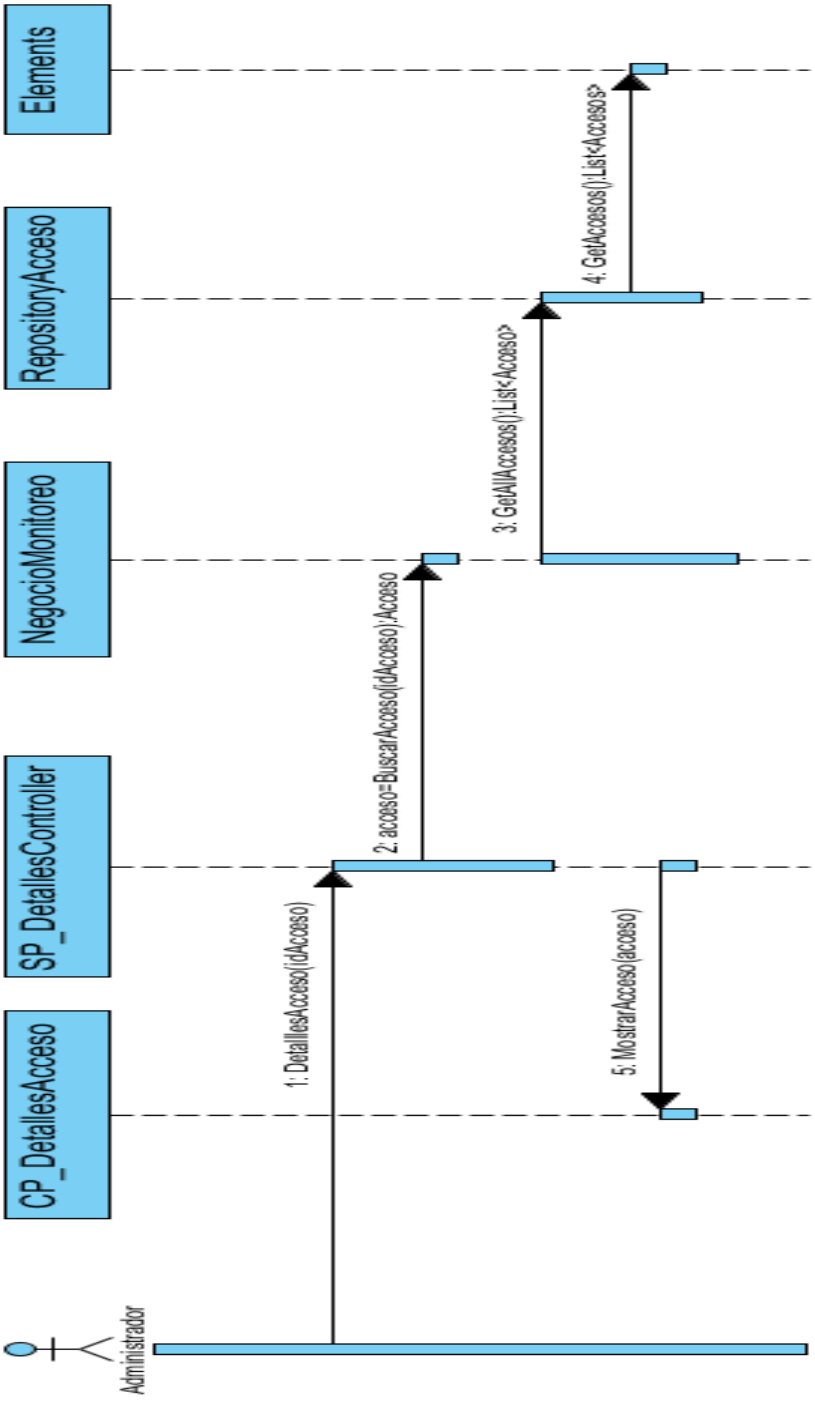


Figura A. 10 Detalles del acceso.

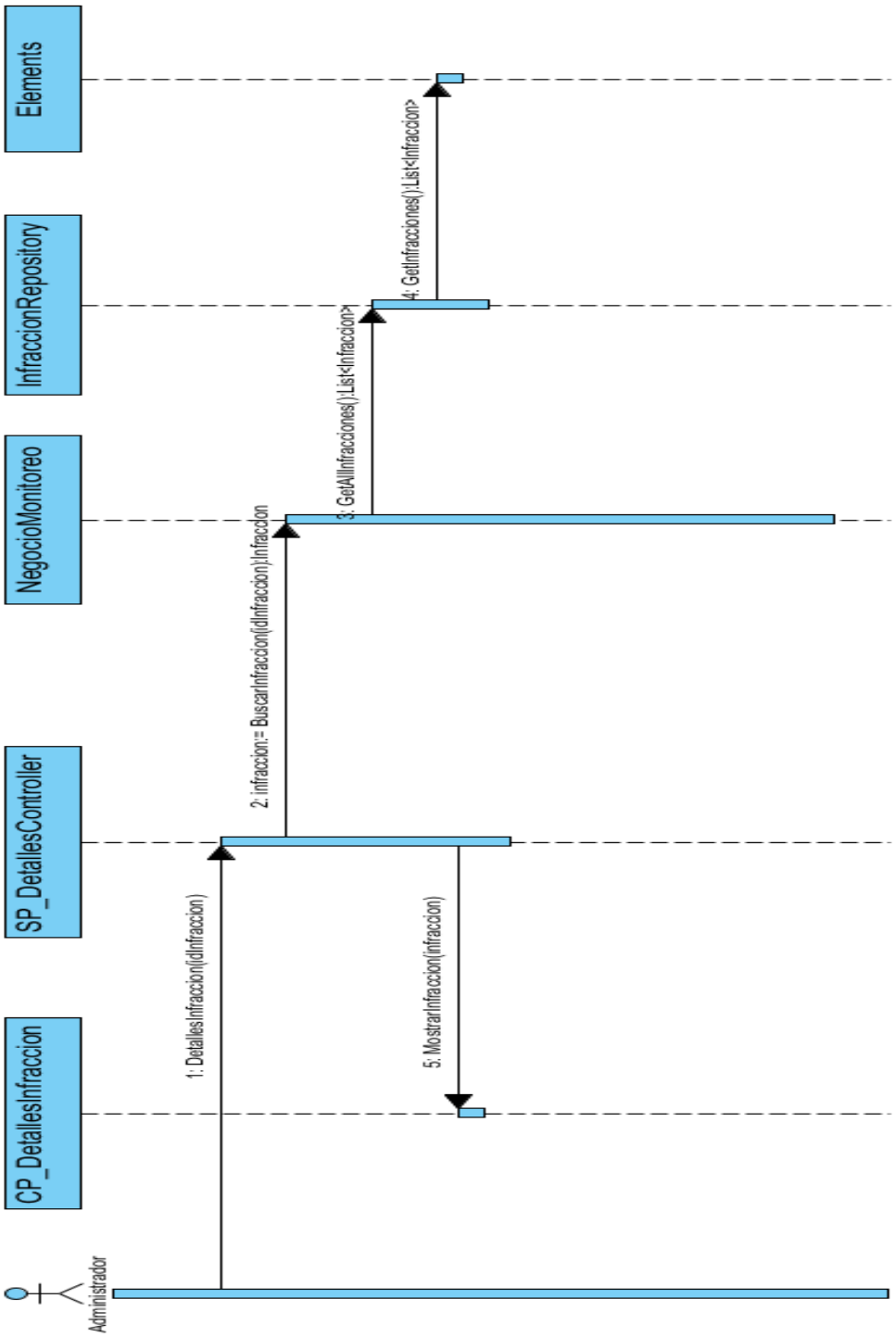


Figura A. 11 Detalles de la infracción.

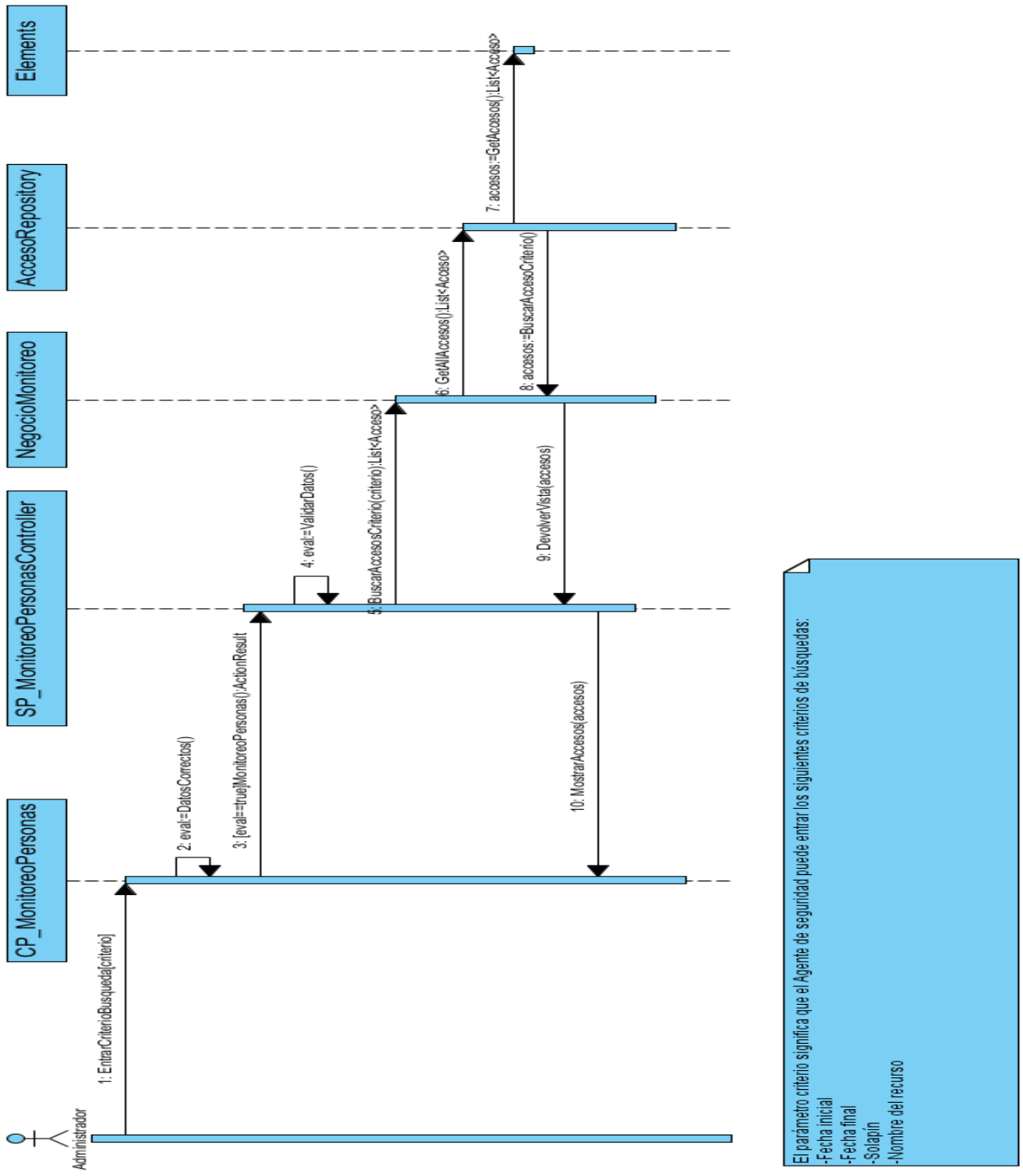


Figura A. 12 Monitoreo del acceso de personas.

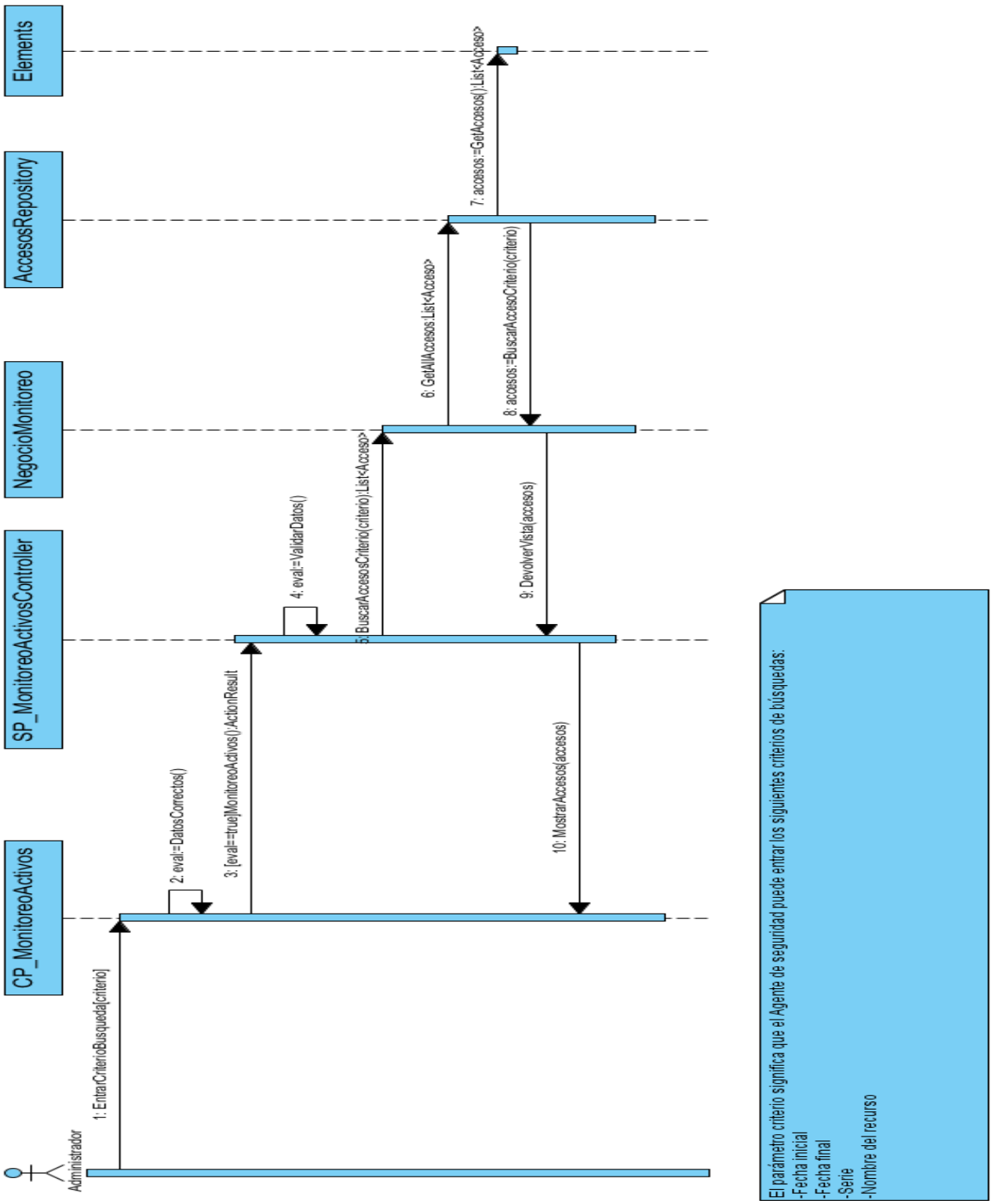


Figura A. 13 Monitoreo del acceso de activos.

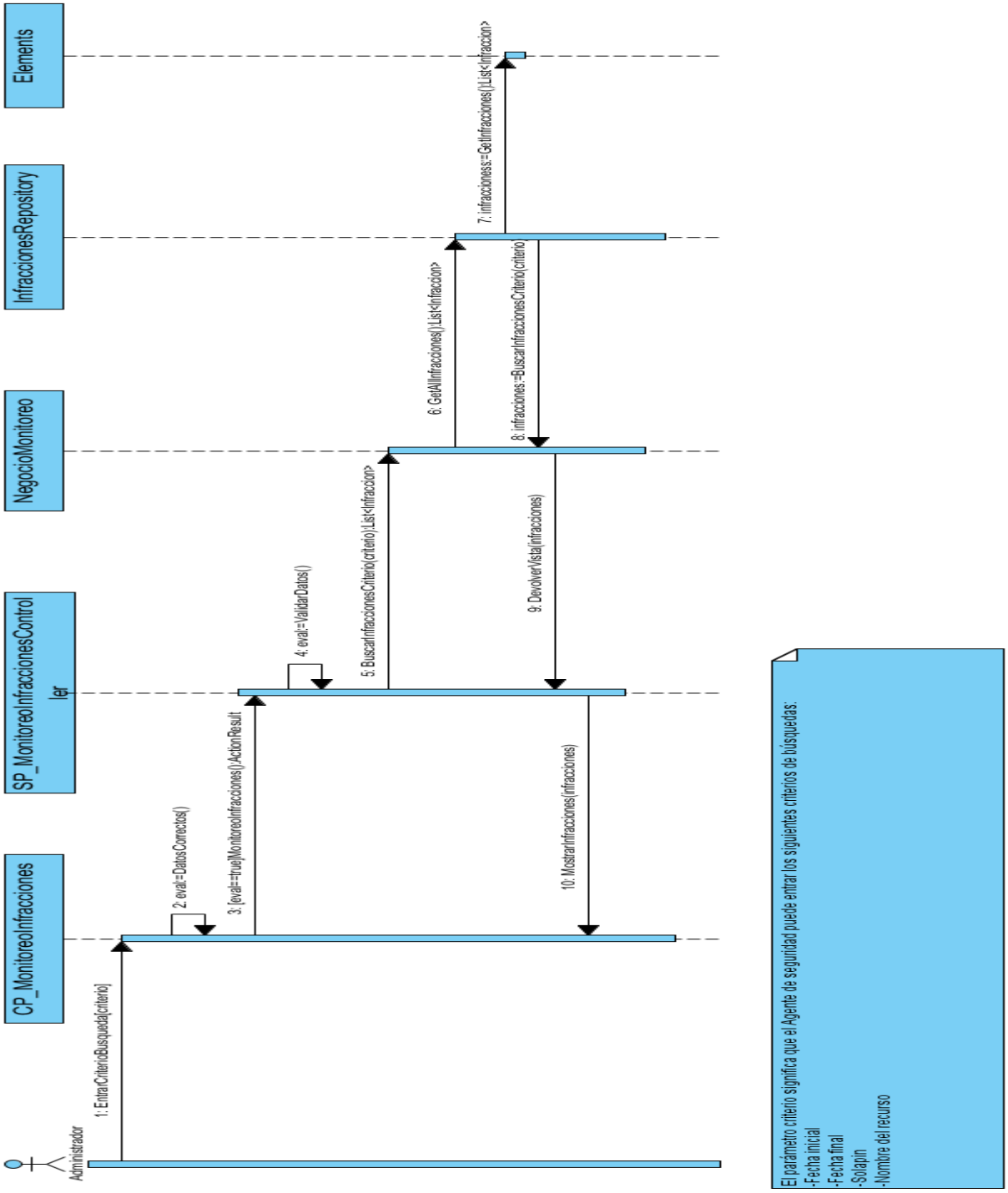


Figura A. 14 Monitoreo de infracciones.

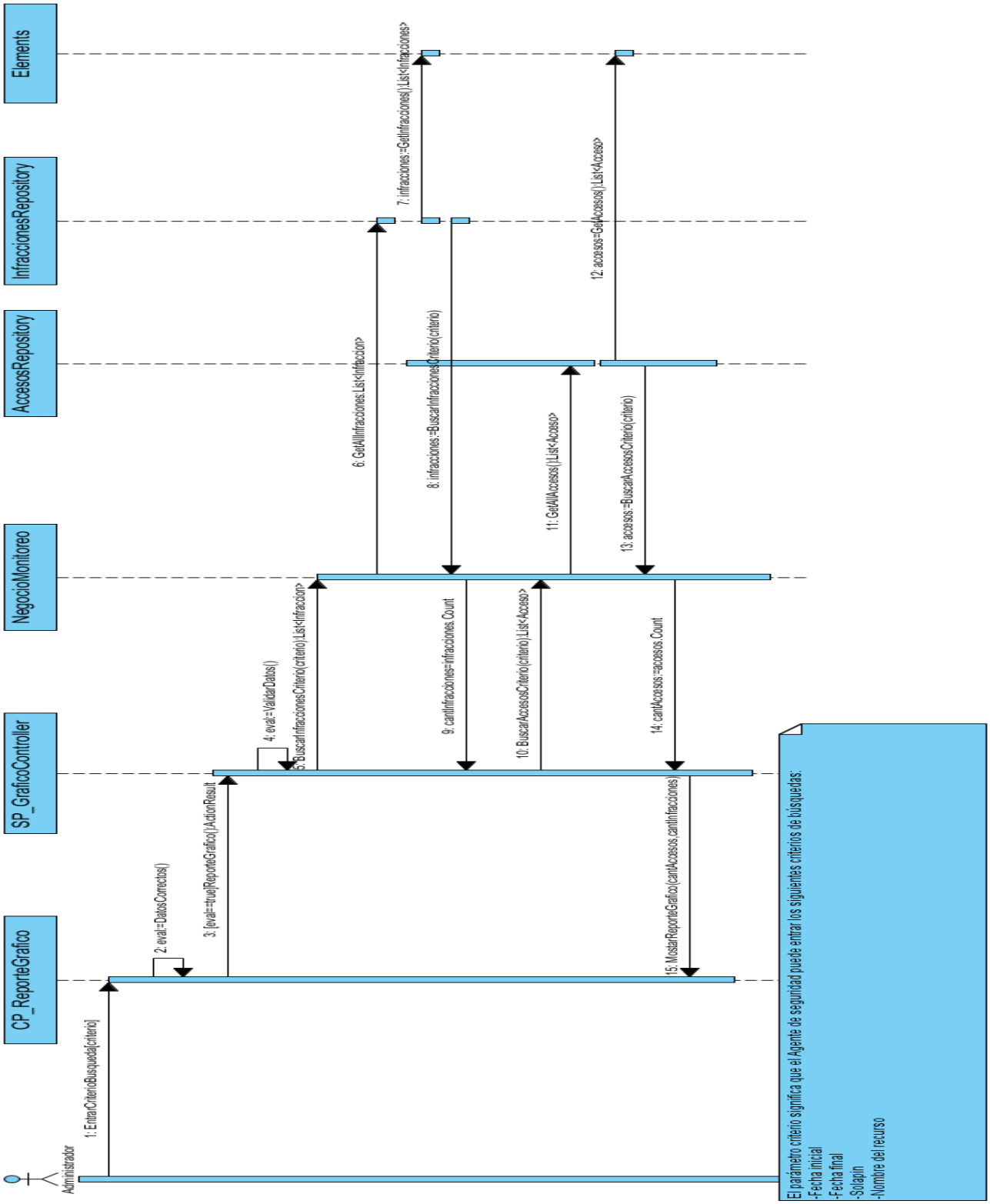


Figura A. 15 Reporte gráfico.

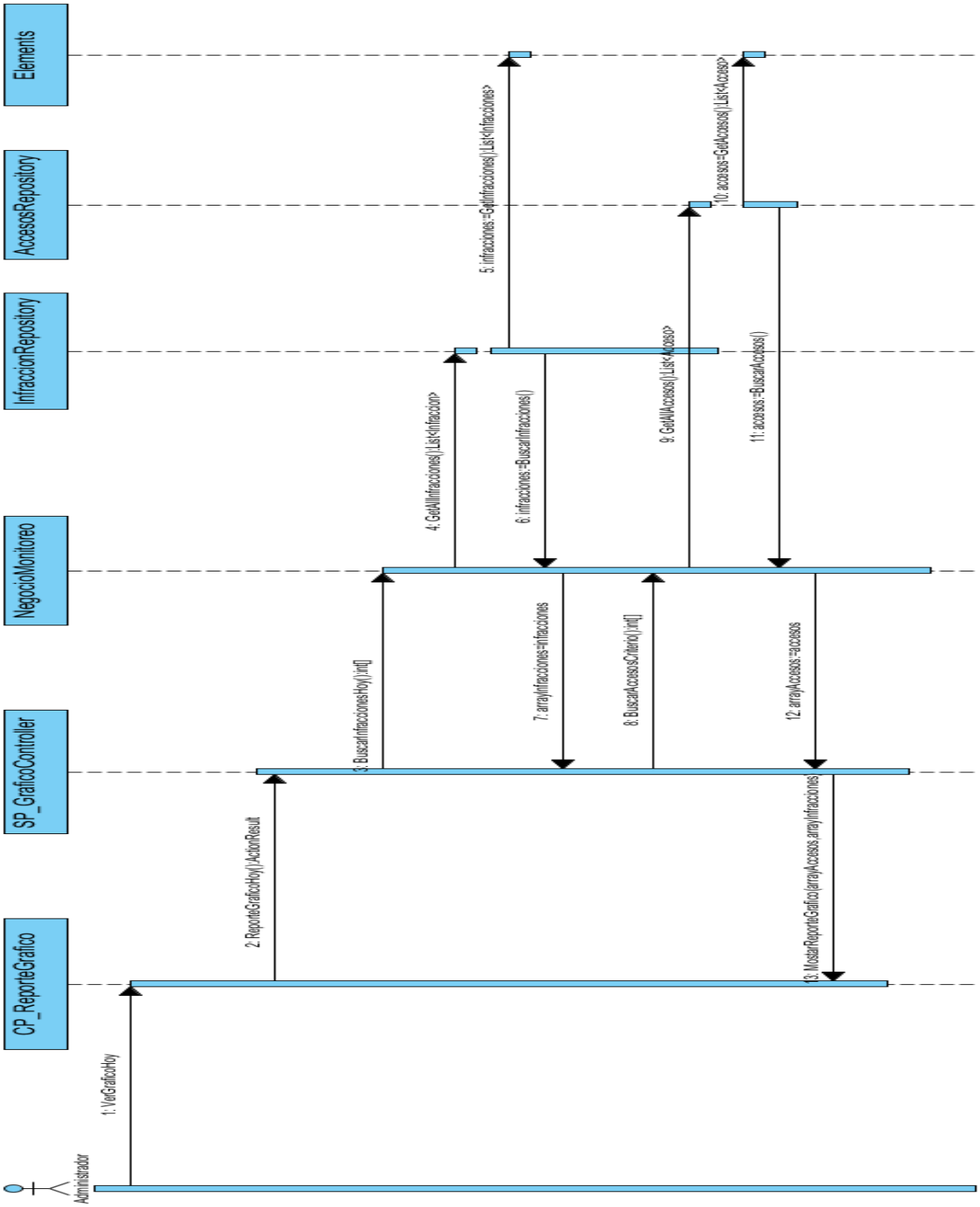


Figura A. 16 Reporte gráfico diario.

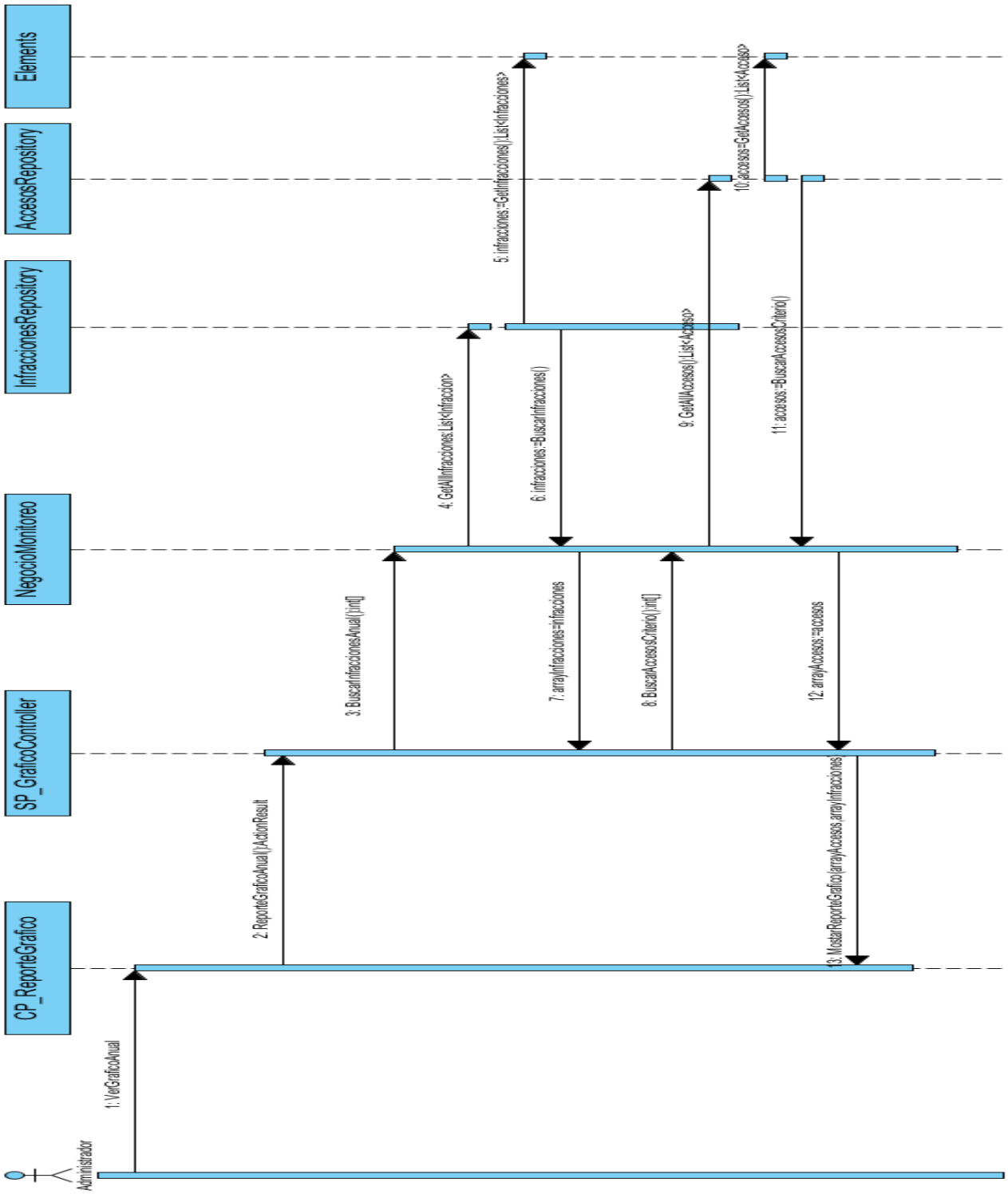


Figura A. 17 Reporte gráfico anual.

ANEXO IX Interfaces del sistema.

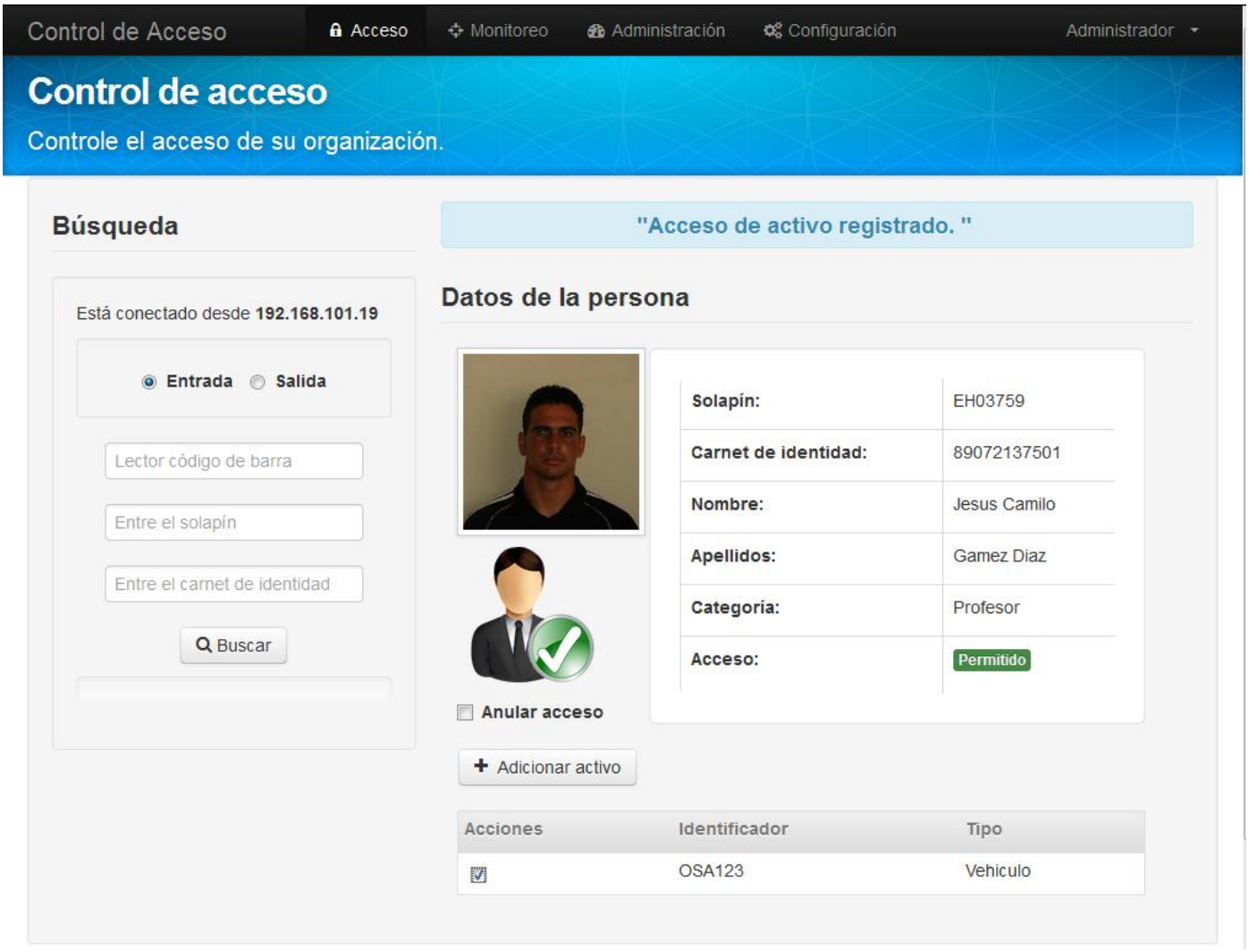


Figura A. 18 Interfaz "Datos de la persona".

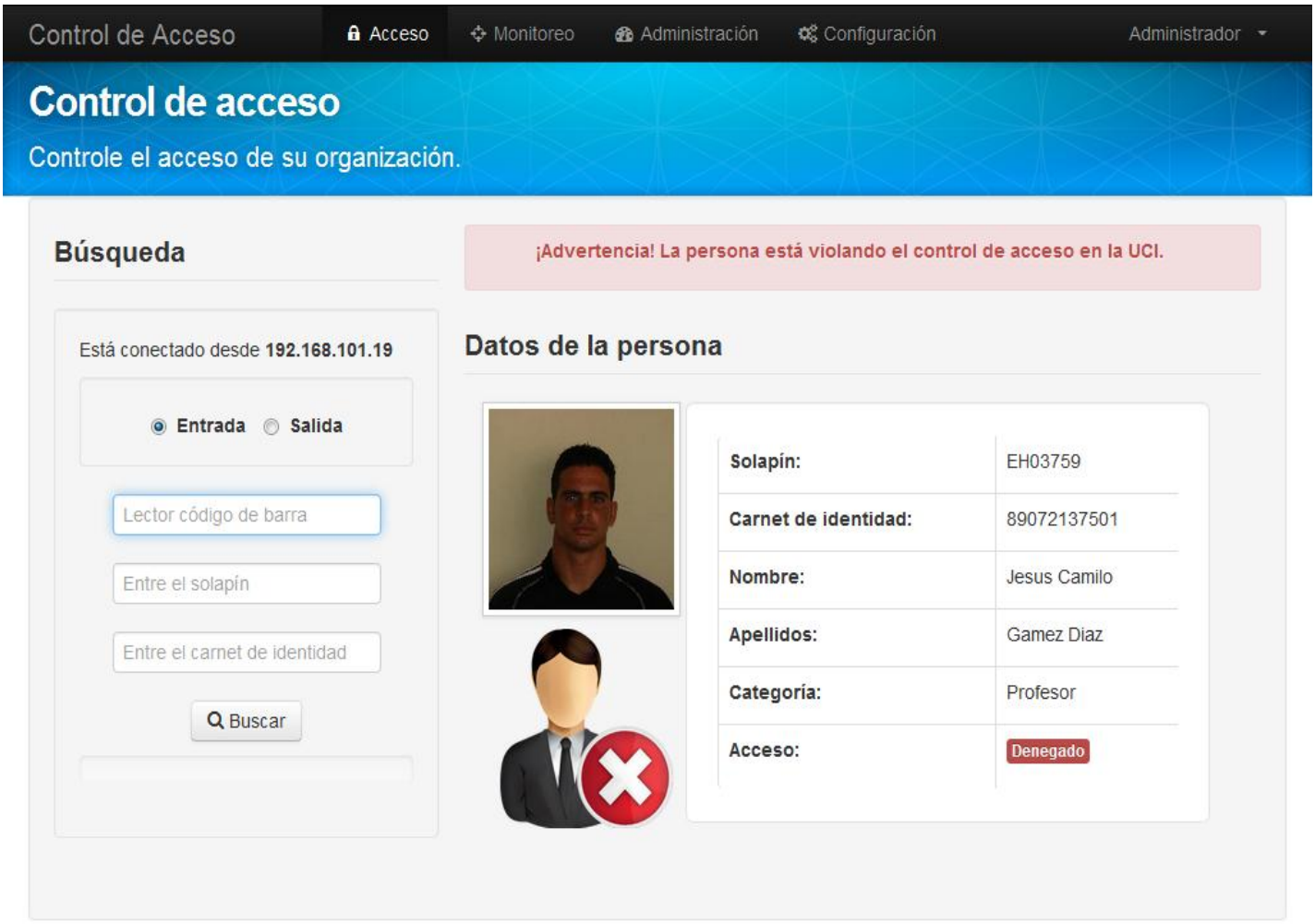


Figura A. 19 Interfaz "Datos de la persona".

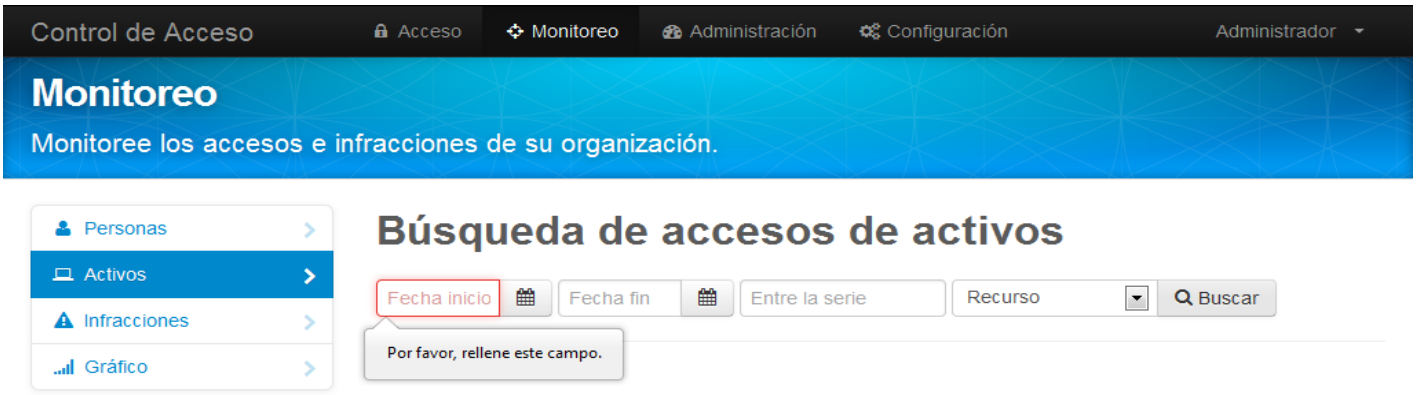


Figura A. 20 Interfaz "Búsqueda de accesos de activos".

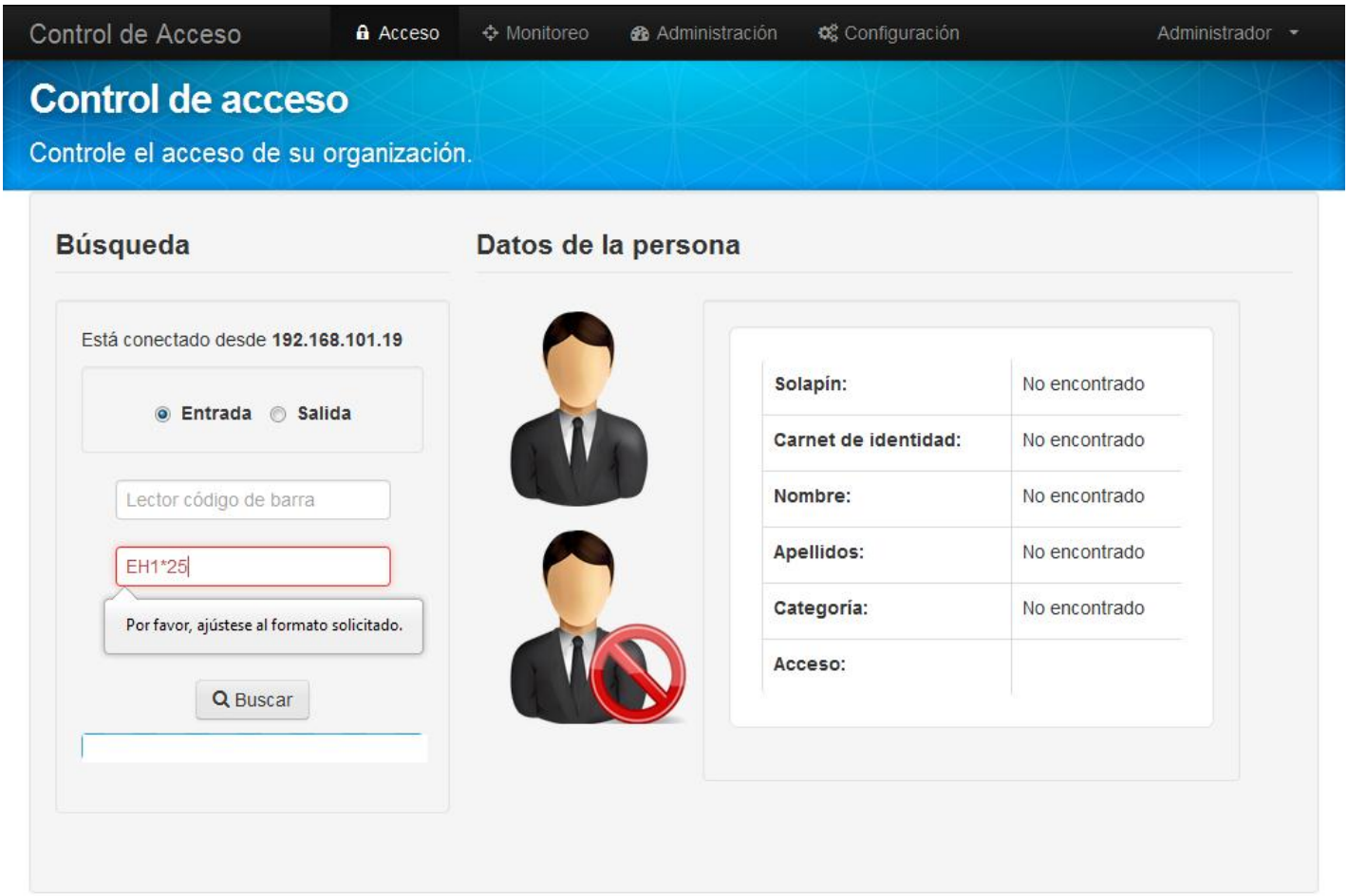


Figura A. 21 Interfaz "Datos de la persona".

Figura A. 22 Interfaz "Datos de la persona".

Figura A. 23 Interfaz "Datos de la persona".

ANEXO X Pruebas unitarias.

CrearAccesoActivoTest

Prueba de unidad		
Nombre: CrearAccesoActivoTest.		
Estado: Satisfactoria.	**Tipo:** Caja Blanca.	**Última Ejecución:** 14/05/2013.
Ejecutado por: Jesús Camilo Gámez.	**Verificado por:** Marlen del Carmen Ramírez.	
Descripción: Para la ejecución de esta prueba se debe entrar el activo.		
Entrada: Activo activo.		
Criterio de aceptación: Registra el acceso del activo.		

Resultado:

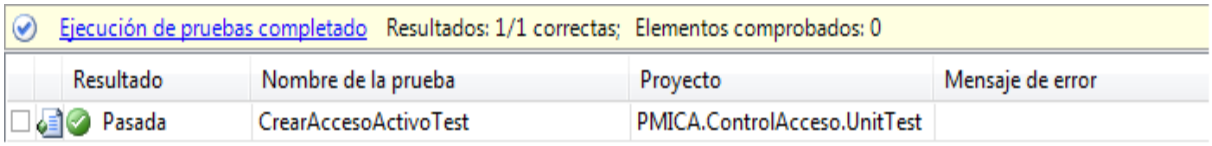


Figura A. 24 Resultados de las pruebas unitarias "CrearAccesoActivoTest".

Tabla A. 21 Pruebas unitarias "CrearAccesoActivoTest".

BuscarInfraccionesFechasRecursoTest

Prueba de unidad		
Nombre: BuscarInfraccionesFechasRecursoTest.		
Estado: Satisfactoria.	**Tipo:** Caja Blanca.	**Última Ejecución:** 14/05/2013.
Ejecutado por: Jesús Camilo Gámez.	**Verificado por:** Marlen del Carmen Ramírez.	
Descripción: Para la ejecución de esta prueba se debe entrar la fecha inicial, la fecha final, el solapín y nombre del recurso.		
Entrada: DateTime fechaInicio, DateTime fechaFin, string solapín, string recurso.		
Criterio de aceptación: Busca las infracciones por el criterio especificado.		
Resultado:		

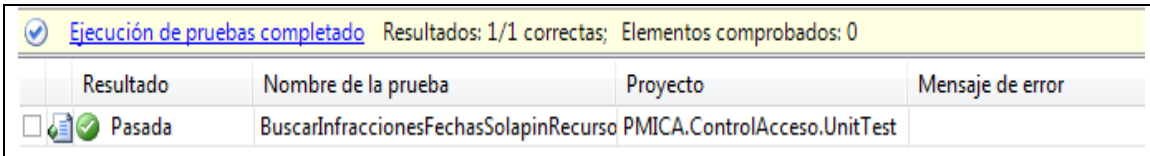


Figura A. 25 Resultados de las pruebas unitarias "BuscarInfraccionesFechasSolapinRecursoTest".

Tabla A. 22 Pruebas unitarias "BuscarInfraccionesFechasRecursoTest".

BuscarPersonaSolapinTest

Prueba de unidad		
Nombre: BuscarPersonaSolapinTest.		
Estado: Satisfactoria.	**Tipo:** Caja Blanca.	**Última Ejecución:** 14/05/2013.
Ejecutado por: Jesús Camilo Gámez.	**Verificado por:** Marlen del Carmen Ramírez.	
Descripción: Para la ejecución de esta prueba se debe entrar el solapín de la persona.		
Entrada: string solapin		
Criterio de aceptación: Buscar una persona.		
Resultado:		

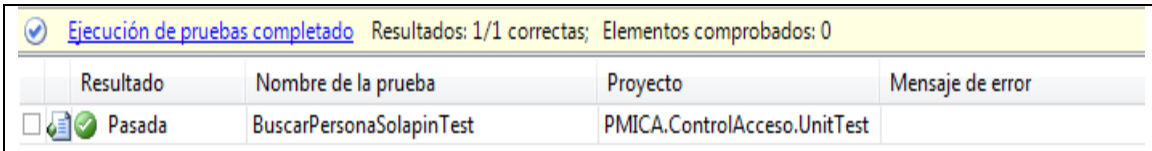


Figura A. 26 Resultados de las pruebas unitarias "BuscarPersonaSolapinTest".

Tabla A. 23 Pruebas unitarias "BuscarPersonaSolapinTest".

CrearAccesoTest

Prueba de unidad		
Nombre: CrearAccesoTest.		
Estado: Satisfactoria.	**Tipo:** Caja Blanca.	**Última Ejecución:** 14/05/2013.
Ejecutado por: Jesús Camilo Gámez.	**Verificado por:** Marlen del Carmen Ramírez.	
Descripción: Para la ejecución de esta prueba se debe entrar la persona, el recurso y el tipo de acceso.		
Entrada: Persona persona, Recurso recurso, TipoAcceso tipoAcceso.		
Criterio de aceptación: Crear un acceso.		
Resultado:		

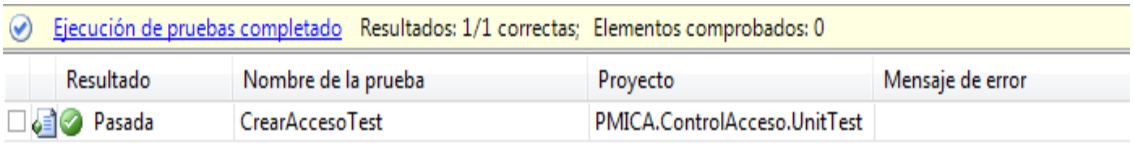


Figura A. 27 Resultados de las pruebas unitarias "CrearAccesoTest".

Tabla A. 24 Pruebas unitarias "CrearAccesoTest".

BuscarAccesoPersonasFechasRecurso

Prueba de unidad		
Nombre: BuscarAccesoPersonasFechasRecursoTest.		
Estado: Satisfactoria.	**Tipo:** Caja Blanca.	**Última Ejecución:** 14/05/2013.
Ejecutado por: Jesús Camilo Gámez.	**Verificado por:** Marlen del Carmen Ramírez.	
Descripción: Para la ejecución de esta prueba se debe ingresar la fecha inicial, la fecha final, el solapín y nombre del recurso.		
Entrada: DateTime fechaInicio, DateTime fechaFin, string solapín, string recurso.		
Criterio de aceptación: Busca los accesos por el criterio especificado.		

Resultado:

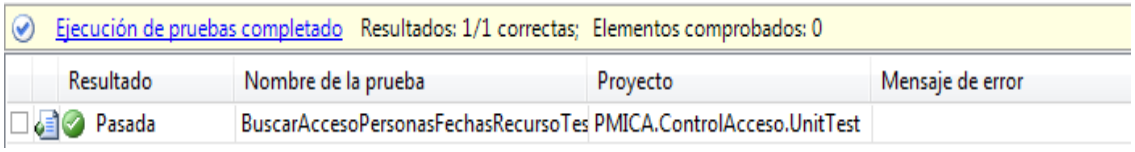


Figura A. 28 Resultados de las pruebas unitarias "BuscarAccesoPersonasFechasRecursoTest".

Tabla A. 25 Pruebas unitarias "BuscarAccesoPersonasFechasRecursoTest".

ANEXO XI Pruebas de caja negra.

Escenario	Descripción	Texto	Respuesta del Sistema	Flujo central
Esc. 2.1: Adicionar activo.	Para registrar activos de manera automática se debe de llenar los campos necesarios.	Serie (válido). Tipo (válido). Descripción (válido).	El sistema registra la información introducida por el usuario y mostrará el nuevo activo en la tabla correspondiente a estos.	1. Opción Acceso. 2. Datos de acceso de la persona. 3. Adicionar activo. 4. Guardar.
		Serie (inválido). Tipo (inválido) Descripción (inválido).	El sistema devolverá un mensaje de entrar los datos del activo correctamente.	

Tabla A. 26 Diseño de Caso de prueba: Módulo Control de Acceso.

Caso de prueba: Módulo Monitoreo

Escenario	Descripción	Texto	Respuesta del Sistema	Flujo central
Esc. 3.1: Mostrar acceso de personas por criterio de búsquedas.	Para observar los accesos de las personas se realizan filtros de búsquedas mediante las fechas y el solapín.	Fecha inicial (válido). Fecha fina l(válido) Solapín (válido).	El sistema mostrará los datos correspondientes del acceso.	1. Opción Monitoreo personas.
		Fecha inicial (inválido). Fecha final (inválido).	El sistema mostrará mensajes: Nos e pueden dejar fechas vacías o Cadena no admitida.	
		Solapín (inválido).	El sistema mostrara un	

			mensaje de sólo introducir letras y números.	

Tabla A. 27 Diseño de Caso de prueba: Módulo Monitoreo.