

# Universidad de las Ciencias Informáticas

## Facultad 10



**Título:** Módulo Control de Acceso del proyecto Intranet del Centro Rector de Universidad para Todos

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

**Autores**

Madelyn Saumell Reyes  
Miguel Jaeger Rodríguez Lazo

**Tutor**

Daynel Marmol Lacal

Ciudad de La Habana, Cuba  
Junio, 2007

## ***DECLARACIÓN DE AUTORÍA***

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Autores: Madelyn Saumell Reyes

Miguel Jaeger Rodriguez Lazo

Tutor: Daynel Marmol Lacal

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Tutor

## ***DATOS DE CONTACTO***

Tutor: Lic. Daynel Marmol Lacal (dmarmol@uci.cu)

Graduado de Licenciatura en Ciencias de la Computación en la Universidad Central “Martha Abreu” de Las Villas en julio del 2005. Desde su graduación está trabajando en la UCI, desempeñándose como profesor de las asignaturas Programación II (Estructura de Datos) y Programación IV (Compiladores), desde su incorporación al centro es líder del proyecto para informatizar el Centro Rector de Universidad para Todos.

## **AGRADECIMIENTOS**

Agradecemos a todas aquellas personas que han sido fuente de inspiración para estos entes a lo largo de su carrera, a los que desde un inicio creyeron en nosotros y nos apoyaron en esta travesía, pero sobre todo a los que no confiaron ni pensaron que podíamos llegar a hacer realidad este sueño de convertirnos en Ingenieros en Ciencias Informáticas. A esos, pues fueron los que nos motivaron a esforzarnos cada día más y a crecernos ante las dificultades para poder demostrarles lo errados que estaban.

A nuestros padres pues por ellos es que hoy formamos parte de este mundo.

A nuestros familiares y a todas aquellas personas que nos supieron apoyar en todo momento brindándonos fuerzas para seguir adelante.

A nuestros compañeros de los grupos 10501 y antiguo 10505, amigos de los proyectos Biblioteca Nacional y 23 y B por compartir con nosotros tantas alegrías y pesares.

A Yadi, Ialy, Daymo, Liut, Ailén, Mare, Nadia, Grethel, Kenia, Lisse y Yaumy, Osmany, Ado, Geikelyn, Daniel y a mi amiga Ari por soportarme todo este tiempo.

A Yusliel pues sin su ayuda no lo habríamos logrado.

A todos nuestros profesores por ser la lumbre que iluminaba nuestro camino hacia la meta, en especial a aquellos que nos ayudaron en el desarrollo del trabajo, a Chony y a David, a Carmen y a profesores de ingeniería del software Maidely y Susel, a nuestro tutor Daynel.

A la Revolución Cubana pues de no existir, lo más probable es que ninguno de nosotros hubiera podido llegar a donde hemos llegado.

Los Autores

## **DEDICATORIA**

***“La confianza en sí mismo es el primer secreto del éxito.”***

***Ralph Waldo Emerson***

A ti Mima por haber confiado siempre en mi, y enseñarme a seguir y luchar por mis sueños hasta hacerlos realidad.

Miguel Jaeger

***“A veces sentimos que lo que hacemos es tan solo una gota en el mar, pero el mar sería menos si le faltara esa gota.”***

***Madre Teresa de Calcuta***

A mis padres y mis hermanos....

A mis abuelitas del corazón.....

A una madre que nunca olvidaré: Maité

A mis tías por su apoyo...

A todas estas personas por haberme apoyado y guiado todos estos años de estudio.

Madelyn Saumell

## **RESUMEN**

El Control de Acceso al Centro Rector de Universidad para Todos y Canales Educativos actualmente ocurre de manera manual, lo que trae como consecuencia demoras, ineficiencia y vulnerabilidades en la seguridad. A esta institución en la actualidad accede un gran número de personas, tanto trabajadores, visitantes, como profesores vinculados.

Como parte del proceso de informatización que se lleva a cabo en este Centro se pretende desarrollar un Módulo para el Control de Acceso. En este trabajo se presenta un servicio web que brinda funcionalidades que permiten llevar a cabo el control de acceso de personas a dicho centro y áreas dentro del mismo, así como gestionar información referente a todo este proceso, constituyendo el sistema software a desarrollar.

En este documento se reflejan los resultados del estudio realizado en esta institución para el desarrollo e implementación del servicio web, se explican conceptos relacionados con el trabajo a desarrollar, se hace un análisis de la propuesta del sistema, y se exponen algunas recomendaciones para mejoras futuras.

## ÍNDICE

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1    SISTEMAS DE CONTROL DE ACCESO .....	5
1.1.1    Características Generales de los Sistemas de Control de Acceso.....	8
1.2    TENDENCIAS ACTUALES DE LOS SISTEMAS DE CONTROL DE ACCESO. ....	9
1.3    ANÁLISIS DE LAS TECNOLOGÍAS SOFTWARE .....	11
1.3.1    ¿Qué es un servicio web? .....	11
1.3.2    ¿Qué es XML?.....	11
1.3.3    ¿Qué es SOAP? .....	12
1.3.4    ¿Qué es NuSOAP?.....	12
1.3.5    ¿Qué es WSDL?.....	12
1.3.6    ¿Por qué utilizar servicios web? .....	12
1.3.7    Fundamentación del lenguaje de programación a utilizar .....	13
1.3.8    Fundamentación del Gestor de Bases de Datos a utilizar .....	13
1.3.9    Fundamentación de la metodología a utilizar .....	14
1.3.10    Herramientas a utilizar .....	15
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>16</b>
2.1    CENTRO RECTOR DE UNIVERSIDAD PARA TODOS Y CANALES EDUCATIVOS.....	16
2.1.1    Objetivos estratégicos del DSPF y procesos de negocio que los soportan.....	17
2.1.2    Flujo actual y análisis de los procesos del negocio involucrados en el campo de acción .....	17
2.2    OBJETO DE AUTOMATIZACIÓN. ....	19
2.2.1    Descripción de los procesos a automatizar .....	19
2.2.2    Información que se maneja.....	19
2.3    DESCRIPCIÓN DEL SISTEMA PROPUESTO .....	19
2.4    MODELO DE NEGOCIO .....	21
2.4.1    Actores del negocio .....	21

2.4.2	Trabajadores del negocio .....	21
2.5	DIAGRAMA DE CASOS DE USO DEL NEGOCIO .....	22
2.5	DESCRIPCIÓN DE LOS CASOS DE USOS DEL NEGOCIO.....	22
2.6	DIAGRAMAS DE ACTIVIDADES.....	22
2.7	DIAGRAMA DE CLASES DEL MODELO DE OBJETOS.....	26
2.9	ESPECIFICACIÓN DE REQUISITOS.....	27
2.9.1	Requerimientos Funcionales .....	27
2.9.2	Requerimientos no Funcionales .....	29
2.10	DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA.....	31
2.10.1	Actores del sistema .....	31
2.10.2	Casos de usos del sistema .....	31
2.11	DIAGRAMA DE CASOS DE USO DEL SISTEMA .....	36
2.11.1	Casos de uso por ciclo de desarrollo .....	36
2.11.2	Casos de Uso expandidos .....	37
<b>CAPÍTULO 3: DISEÑO DEL SISTEMA PROPUESTO.....</b>		<b>38</b>
3.1	MODELO DE DISEÑO.....	38
3.2	DIAGRAMAS DE INTERACCIÓN .....	38
3.3	DIAGRAMA DE CLASES .....	38
3.3.1	Diagramas de clases por paquetes .....	40
3.3.2	Descripción de las clases .....	40
3.4	DISEÑO DE LA BASE DE DATOS.....	61
3.4.1	Modelo de Datos.....	61
3.4.2	Descripción de las tablas de la Base de Datos.....	62
<b>CAPÍTULO 4: IMPLEMENTACIÓN DEL SISTEMA.....</b>		<b>66</b>
4.1	MODELO DE DESPLIEGUE.....	66
4.2	MODELO IMPLEMENTACIÓN .....	67
4.2.1	Diagrama de Componentes en Paquetes.....	67
4.2.2	Diagramas de Componentes separados por paquetes .....	68
4.2.3	Descripción de cada uno de los Paquetes.....	70



<b>CONCLUSIONES .....</b>	<b>71</b>
<b>RECOMENDACIONES.....</b>	<b>72</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	
<b>BIBLIOGRAFÍA</b>	
<b>ANEXOS</b>	
<b>GLOSARIO DE TÉRMINOS</b>	

## ***INTRODUCCIÓN***

La seguridad es básicamente la protección de algo valioso para asegurar que no sea perdido, o alterado. En la actualidad, son cada vez más las organizaciones de todas las partes del mundo, desde pequeñas compañías hasta grandes empresas corporativas y gubernamentales que tienen áreas en las cuales se manejan equipos especiales, dinero, grandes volúmenes de tecnologías, en fin, cierta cantidad de recursos que emplean para un mejor funcionamiento y desarrollo de su trabajo, estando al tanto del peligro ante fallas que puedan existir en la seguridad (1).

Para la protección de estos medios utilizan sistemas que garantizan la integridad y el cuidado diario de los mismos; en estos momentos a nivel internacional se emplean sistemas que controlan y regulan el acceso de todo el personal que asista de una forma u otra a una institución determinada. Lo más difícil en esta situación es encontrar el sistema que verdaderamente garantice la protección y sea adecuado a las posibilidades económicas que se tengan, pues no es menos cierto que existen vulnerabilidades que provocan en ocasiones que existan ataques de robo, malversación o maltrato de medios puestos a disposición.

Nuestro país, no se encuentra lejos de esta realidad, ya que existen diversos centros de trabajo, organizaciones, entidades, que emplean sistemas que restringen el acceso de su personal al igual que instituciones similares ubicadas en otras regiones del mundo.

Surge así, como una de las nuevas entidades de nuestro país y ligada a esta realidad, el Centro Rector de Universidad para Todos y Canales Educativos, siendo una infraestructura tecnológica y física, con una serie de recursos tecnológicos de última adquisición, con el objetivo de brindar servicios a los profesores de Universidad para Todos y Canales Educativos, que ha sido el personal dedicado a la formación por TV en Cuba desde hace más de 5 años (2).

Actualmente, acceden a esta institución y a sus recursos disponibles, los trabajadores (directivos, de servicio, de seguridad), invitados por alguna u otra razón, así como profesores, que hoy son pocos pero en un futuro, luego de su inauguración, está previsto que asista un número mayor con el objetivo de prepararse, o enriquecer sus conocimientos para impartir mejores teleclases.

En la actualidad solamente se realiza un pequeño registro de los datos personales de estas personas que entran y salen de la institución en la entrada principal, en la garita, por parte del Agente de

Seguridad, archivando esta información en papel. Sólo se lleva a cabo el control en esta entrada y no en otra zona de la institución, trayendo consigo que no se tenga un dominio total de las áreas a las cuales puede acceder una persona una vez estando dentro, pudiendo ponerse en peligro toda la tecnología que allí se encuentra. Todo este proceso es llevado manualmente, lo que se torna un tanto inseguro y a su vez sería engorroso generar información del mismo, ya que sería complicado mostrar algún reporte de cuántas personas, trabajadores o visitantes entraron un día determinado a la institución teniendo en cuenta la forma en que se desarrolla, esta actividad requiere gasto de tiempo y recursos humanos y la información puede correr el riesgo de ser perdida.

Por tales motivos y debido a la no existencia de un mecanismo eficiente que garantice el acceso de personal, solamente autorizado al Centro Rector de Universidad para Todos y Canales Educativos y áreas dentro del mismo, surgió la idea de automatizar el control de acceso de personas, brindando funcionalidades para controlar de manera eficaz y segura las entradas y salidas de todo el personal, así como las áreas a las cuales tienen acceso, que permita gestionar en poco tiempo toda la información precisa y requerida y garantice que el nivel de seguridad sea confiable en la institución.

Dada la **situación problemática** descrita se plantea como **Problema de Investigación**:

¿Cómo resolver de forma automatizada el control de acceso al Centro Rector de Universidad para Todos y Canales Educativos, y áreas dentro del mismo?

Considerando como **Objeto de Estudio** la seguridad y protección del Centro Rector de Universidad para Todos y Canales Educativos y dentro de éste como: **Campo de Acción** el control de acceso a dicho centro y áreas dentro del mismo.

Como **Objetivo general de la investigación** se trazó:

Desarrollar un servicio web que brinde funcionalidades que permitan llevar a cabo el control de accesos de personas al Centro Rector de Universidad para Todos y Canales Educativos y áreas dentro del mismo, así como gestionar información referente a todo este proceso.

Para el desarrollo de la investigación se proponen las siguientes **preguntas científicas**:

1. ¿Cuáles son los fundamentos teóricos que sustentan los sistemas de control de acceso?
2. ¿Cuáles son las tecnologías software empleadas para el desarrollo de servicios web?
3. ¿Cómo desarrollar un servicio web que responda a las necesidades del Control de Acceso al Centro Rector de Universidad para Todos y Canales Educativos?

Con el objetivo de guiar la investigación se definieron las siguientes tareas:

1. Realizar un estudio de los fundamentos teóricos que sustentan los sistemas de control de acceso.
2. Realizar un estudio de las tecnologías para desarrollar servicios web.
3. Identificar las necesidades del Centro Rector de Universidad para Todos y Canales Educativos en cuanto al proceso de control de acceso.
4. Evaluar las alternativas de implementación de servicios web que se adecuan a las condiciones de la institución.
5. Diseñar e implementar un servicio web que gestione el proceso de control de acceso a esta institución.

Para darle cumplimiento a las tareas de investigación, se utilizaron varios métodos que hicieron posible la misma:

El método **Analítico-Sintético** se empleó para encontrar la relación entre los diferentes conceptos involucrados en los procesos de control de acceso y para realizar una propuesta de solución de acuerdo a las condiciones específicas del campo de acción.

Se utilizó el método empírico de **observación**, para poder determinar los procesos del negocio relacionados con el control de acceso al Centro Rector de Universidad para Todos y Canales Educativos así como las necesidades reales del mismo. La **entrevista** fue empleada como vía esencial para obtener datos de interés acerca de los procesos de entradas y salidas a la institución.

El presente documento se estructura en cuatro capítulos y las correspondientes conclusiones, recomendaciones, referencia bibliográfica, bibliografía, anexos y glosario de términos.

**Capítulo 1:** Fundamentación teórica. Se describen algunos conceptos y características de sistemas de control de acceso que se deben tener en cuenta para comprender los objetivos del trabajo y se realiza un análisis de las tecnologías de software empleadas para desarrollar el sistema.

**Capítulo 2:** Características del Sistema. Se abordan las descripciones iniciales del sistema (servicio web), se hace un estudio del Centro Rector de Universidad para Todos y Canales Educativos y un análisis crítico de los procesos involucrados en el control de acceso, se modela el negocio, se presenta la propuesta del sistema, se definen los requerimientos funcionales y no funcionales y los casos de uso.

**Capítulo 3:** Diseño del sistema propuesto. Se describe el diseño del sistema que se propone. Se representan los diagramas de interacción, los diagramas de clases y el diseño de la base de datos.

**Capítulo 4:** Implementación del sistema. Se describe el sistema que se propone a través de una óptica de programación, basado en diagramas de componentes y diagrama de despliegue.

## 1 FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan características y conceptos importantes relacionados con los sistemas de control de acceso. Se presenta una panorámica general de las tendencias actuales de dichos sistemas de seguridad, y la fundamentación de las tecnologías software a emplear para la implementación, así como la metodología y herramienta para la modelación, teniendo en cuenta la arquitectura general del proyecto al cual pertenece este módulo.

### 1.1 Sistemas de Control de Acceso

Desde los orígenes del tiempo (informático) se ha tratado de controlar el acceso ya sea a recursos o sistemas de información, en la actualidad este tema ha generado gran importancia debido a la revolución tecnológica que ha venido desarrollándose. Es necesario introducir algunos conceptos como:

**Acceso:** es la Entrada o paso (3) a un lugar, dígame institución, compañías, o simplemente a páginas web dentro de un sitio, o a otros servicios informáticos; y el **Control de Acceso** es el mecanismo que en función de la identificación ya autenticada permite acceder a datos o recursos (4); es considerado como el proceso de gestión de los accesos controlados a zonas o áreas determinadas, de personas, medios de transporte, información en caso de ser a un sitio o datos dentro de la red.

Siempre ha sido necesario, por las sociedades o personas, controlar el acceso a ciertas zonas y lugares, porque ha existido el temor a que individuos inescrupulosos puedan robar material valioso de acuerdo a criterios personales, sociales, o comerciales (5) dentro de una institución.

Históricamente, para realizar el control de personas en cuanto a entradas, salidas y privilegios de los mismos a un determinado lugar u horario establecido, ha dependido del uso de tarjetas de identidad

(6), tales como: tarjeta de biblioteca, de crédito o una tarjeta de identificación de trabajador. Las mismas demuestran, ya sea para un guardia o para un dispositivo electrónico, que el portador tiene ciertos derechos sobre el área en la que se encuentra. Teniendo en cuenta esto, se puede afirmar que existen dos formas para el control, como tradicionalmente se realiza: de forma no automatizada a través de la presentación de un objeto de identificación a dicho guardia; y a través de sistemas automatizados, presentando una misma tarjeta o similar pero a un dispositivo electrónico, o sea basado en **Tecnologías de Identificación Automática**.

En la actualidad, motivados por la necesidad de mantener mayor seguridad, protección, combatir problemas que se podrían presentar y producto a los grandes cambios generados a partir de las tecnologías de la información, existen razones y motivos para mantener e implementar sistemas de control de acceso automatizados, sobre zonas que se deseen cuidar, ya que un sistema de control de este tipo es una manera eficiente, segura, rápida y más confiable de restringir el acceso a zonas delicadas, pues permiten registrar automáticamente hacia donde se dirige cada individuo o vehículo, horarios de entradas y salidas, áreas a la cual accedió, así como el tiempo que demoró, permitiendo un acceso controlado, y garantizando el cumplimiento de las políticas de seguridad de la institución donde son puestos en práctica. Como se dice en (7), “básicamente constituyen una de las principales puertas de protección, siendo la primera defensa contra personas no autorizadas que intentasen causar problemas y violar la seguridad”.

El empleo de estos sistemas mejora la productividad y reduce notablemente la ineficiencia y los errores humanos que se pudieran cometer con un sistema no automatizado, aunque hay que tener en cuenta que son implantados por lo general en lugares que cuentan con recursos económicos para pagar su adquisición. Sustituir el método tradicional no es tan fácil en otras zonas del mundo pues restricciones económicas se lo impiden.

Para controlar el acceso se debe tener presente cada uno de los elementos que conforman los sistemas de control de acceso, para implementarlos con fiabilidad, es imprescindible tenerlos claros y precisos. Lo primero que se necesita es definir un método de reconocimiento o sea un sistema de identificación que permita reconocer la identidad de una persona a través de algún dispositivo electrónico, dentro de estos sistemas se encuentran los siguientes:

1. Claves por Teclado
2. Tarjetas de Banda Magnética
3. Tarjetas de Código de Barras
4. Tarjetas de Proximidad o Radio Frecuencia (RF)
5. Biométricos.

Los sistemas de control de acceso con el empleo de los mismos ponen en práctica al menos 4 de los mecanismos complementarios necesarios para garantizar la seguridad, integridad y confidencialidad de las que se hace alusión en este trabajo.

Los mecanismos (ver **Figura 1.1**) que se ponen de manifiesto así como combinaciones entre ellos en dependencia del sistema son los siguientes:



**Figura 1.1** Mecanismos complementarios de un sistema de control de acceso

**Mecanismos:**

- **Identificación de usuario:** La identificación de usuario es un mecanismo que posibilita la identificación de las características propias de un usuario y solo de él. El propósito fundamental que persigue es permitir el acceso a sistemas determinados, a través de la restricción del número de usuarios. Tecnológicamente este mecanismo se implanta mediante la asignación de un nombre de usuario, User ID o el PIN (Personal Identification Number).



- **Autenticación de Usuario:** La autenticación de usuario es un mecanismo basado en la identidad del usuario, o sea saber determinar que un usuario es quien dice ser. Se apoya en tecnologías muy relacionado con la identificación de usuario (User ID). La autenticación se realiza por medio de algo que se conoce, como número de identificación (PIN), login/password, o a través del sistema de identificación que el usuario posea, ya sea: tarjetas inteligentes, magnéticas, códigos de barra, entre otros.
- **Verificación de la autenticación:** La verificación de autenticación es tratada como el mecanismo que facilita probar la autenticidad del usuario, a través de alguna característica propia del usuario como: huellas dactilares, tono de voz, entre otros. Este mecanismo está siendo más popular y práctico en la actualidad, debido a la tecnología que lo apoya.
- **Re-autenticación:** La re-autenticación es lo que posibilita por cualquier motivo reconfirmar, en un momento dado, la autenticidad del usuario. Este mecanismo tiene muy en cuenta el hecho de mantener la autenticación del usuario. Esto se evidencia como ejemplo automáticamente en la interacción de un usuario con un sistema determinado (5).

### **1.1.1 Características Generales de los Sistemas de Control de Acceso**

Los sistemas de control de acceso, de forma general:

- Garantizan la seguridad y control de los recursos que se encuentran a disposición en una entidad o área determinada.
- Permiten a una organización mantener o aumentar la seguridad mientras se ahorra tiempo, logrando una distribución más completa de la información.
- Brindan protección tanto para el acceso físico como lógico de forma simultánea.
- Posibilitan mantener el control de todo el personal que accede de una forma u otra a cualquier área, permitiendo llevar a cabo controles estadísticos de los mismos.
- Actualizan la Base de Datos con los accesos realizados por los usuarios.
- Restringen el acceso a personas no autorizadas ya sea por usos indebidos de recursos informáticos, no estar registrado en la base de datos o por infracciones que se cometen que no permiten el paso.
- Permiten realizar reportes.

- Permiten llevar a cabo tanto un control administrativo centralizado como el análisis de cualquier tentativa de acceso no autorizado.
- Permiten implementar sistemas de alarmas en conjunto con su funcionalidad específica.

## **1.2 Tendencias actuales de los Sistemas de Control de Acceso.**

A nivel internacional existen lugares que ponen en marcha sistemas para que controlen el acceso de personal a las áreas establecidas, con el objetivo de garantizar la seguridad. Existen varios ejemplos de estos sistemas de control electrónicos, que limitan los accesos de acuerdo a criterios propios y específicos del lugar donde son establecidos, se podría citar el sistema de control de acceso (UMGina) en aulas, en la Universidad de Murcia, es utilizado para aulas de libre acceso (ALAs), pues con empleo de tarjetas inteligentes garantizan elementos de seguridad como: autenticación, integridad, confidencialidad y no repudio. En la biblioteca de Alejandría, existe un Sistema que permite el control y el seguimiento de las personas que entran y salen a determinadas zonas de este lugar, basado en tarjetas de la propia instalación. Existen universidades, como la Universidad Panamericana con su sede en Guadalajara y la Universidad Autónoma de Nayarit, que controlan las entradas de personas al Centro de Cómputo correspondientes a las mismas, basados en credenciales con Código de Barras, de Alta Calidad y Durabilidad.

Dentro de este marco se destacan la existencia de software con estos fines como:

- Software de Control de Acceso **JAS WIN**, el mismo soporta los ambientes de Windows NT/2000/XP y Novell y con uso opcional en Linux, posee funciones para controlar días festivos programados por compañías donde es utilizado, puede controlar múltiples unidades, así como llevar a cabo la verificación de cualquier tarjeta de empleado y brindar un fácil manejo de reportes sobre movimientos de personas a través de un edificio.
- Software **ASV\_Gimnasios** para el control de accesos y gestión de gimnasios, este software permite controlar las personas que asisten a gimnasios y centros deportivos, dentro de sus funciones están: registrar los datos necesarios de los servicios que se brindarán en dichos recintos, modificarlos y eliminarlos, registrar los datos de las personas que asisten y consumen dichos servicios, así como actualizarlos en casos pertinentes, permitir conocer en cada momento cuántas veces y cuándo acceden, denegar entradas si es necesario, permitir

obtener listados de incidencias que se producen, restringir los horarios de accesos de personas, prohibir la entrada a alguien a partir de una fecha determinada así como otras funciones de interés.

De forma general de todos estos sistemas sólo se pueden conocer características generales e información que procesan, no cómo fueron desarrollados ni qué tecnología emplearon; sin embargo, esto no es una limitante para la presente investigación, pues sólo se quería conocer principalmente su funcionamiento. Por otro lado, generalmente estos sistemas son caros y a la vez procesan información específica de los lugares para los cuales fueron desarrollados.

En nuestro país prevalecen entidades que emplean sistemas automatizados para controlar el acceso, tal es el caso de: la Corporación Cimex.SA, donde existe un sistema de control basado en tarjetas magnéticas para la entrada de personas a la institución y a cada una de sus oficinas, el mismo registra los horarios de entrada y salida, así como datos, ya sean de trabajadores o visitantes, permite realizar reservaciones y mostrar estadísticas de interés. En la Empresa de Telecomunicaciones de Cuba (ETECSA), entidad encargada de los servicios públicos de telecomunicaciones mediante la operación, instalación, explotación y mantenimiento de las redes públicas de telecomunicaciones en todo el territorio, se controla la entrada y salida de sus trabajadores a cada uno de los locales con un sistema similar al anterior y en los Hoteles 4 y 5 estrellas del país para controlar la entrada y salida de huéspedes a las habitaciones.

En la Universidad de las Ciencias Informáticas se emplea un sistema para el control de entradas a los comedores, el mismo garantiza que las credenciales sean utilizadas una sola vez y verifica que el portador de la misma pertenece a dicho comedor. Fue desarrollado con el empleo de tecnología .Net, como una aplicación de escritorio. Existe también el sistema de control de acceso centralizado en algunos de los edificios donde se encuentran los laboratorios, por lo general de proyectos, dicho sistema verifica si el estudiante que entra, a través del número de solapín<sup>1</sup> pertenece a uno de los laboratorios de dicho bloque y así a uno de los proyectos, registrando la hora de entrada, de salida, quien autoriza la entrada y algunas estadísticas generadas del proceso, éste fue desarrollado con el empleo de PHP y MySQL como Gestor de Base de Datos.

---

<sup>1</sup> Credencial empleada con fotografía y datos de una persona, que permiten identificarla.

### **1.3 Análisis de las tecnologías software**

Para el desarrollo del sistema que se desea implementar con el fin de gestionar la información referente al control de acceso al Centro Rector de Universidad para Todos y Canales Educativos, se tuvo en cuenta el análisis de las tecnologías y la propuesta planteada por la arquitectura del proyecto al cual pertenece este módulo. A continuación se fundamentan las tecnologías software empleadas y algunas definiciones que son importantes conocer.

#### **1.3.1 ¿Qué es un servicio web?**

Un servicio web es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Es un componente de software que se comunica con otras aplicaciones codificando los mensajes en XML y enviando estos mensajes a través de protocolos estándares de Internet tales como el Hypertext Transfer Protocol (HTTP). Intuitivamente, es similar a un sitio web pero no cuenta con una interfaz de usuario ya que da servicios a aplicaciones y no a las personas, pues en vez de obtener solicitudes desde el navegador y retornar páginas web como respuesta, lo que hace es recibir solicitudes a través de un mensaje formateado en XML desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta también formateado en XML (8). Los servicios web exponen funcionalidad útil a los usuarios web mediante un protocolo web estándar. En la mayoría de casos, el protocolo utilizado es SOAP. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar estos servicios para intercambiar datos en redes de ordenadores como Internet.

#### **1.3.2 ¿Qué es XML?**

Es un lenguaje desarrollado por el W3 Consortium para permitir la descripción de información contenida en la red de redes a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red. El formato de XML es muy parecido al del HTML aunque no es una extensión ni un componente de éste (9). Sirve para que muchos programas interpreten bien cualquier tipo de dato y se comuniquen entre ellos sin intervención humana. Constituye el lenguaje base de los Servicios Web.

### **1.3.3 ¿Qué es SOAP?**

Es un protocolo que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje. Está basado en XML y es la base principal de los servicios web (9).

### **1.3.4 ¿Qué es NuSOAP?**

NuSOAP es un conjunto de herramientas (Toolkits) para desarrollar servicios web bajo el lenguaje PHP. Está compuesto por una serie de clases que hacen mucho más fácil el desarrollo de dichos servicios. Provee soporte para el desarrollo de clientes (aquellos que consumen los servicios web) y de servidores (aquellos que los proveen). NuSOAP está basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1 (9).

### **1.3.5 ¿Qué es WSDL?**

Es el lenguaje de descripción de servicios web, es un dialecto basado en XML sobre el esquema que describe un servicio web. Un documento WSDL proporciona la información necesaria al cliente para interactuar con el servicio web. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red, incluyendo SOAP sobre HTTP e incluso protocolos que no se basan en XML como DCOM sobre UDP (10).

### **1.3.6 ¿Por qué utilizar servicios web?**

Se emplea servicios web teniendo en cuenta que la exposición de aplicaciones existentes con estos servicios permite que usuarios creen aplicaciones más potentes y que los utilicen como elementos constituyentes, así como teniendo en cuenta que permiten compartir datos y pueden ser solicitados a través de diferentes plataformas y sistemas operativos, sin tener en cuenta el lenguaje de programación, brindando la posibilidad de gestionar cualquier tipo de información. Proporcionan una solución viable para habilitar la interoperabilidad de datos y sistemas. Además ofrecen funciones muy útiles a usuarios del medio web, ya que emplean un protocolo estándar que, en casi todos los casos, es SOAP y permiten describir sus interfaces con suficiente detalle para que el usuario diseñe una aplicación cliente que permita comunicarse con ellas. Se registran para que sean localizados fácilmente y dicho registro es a través de UDDI (descripción, descubrimiento e integración

universales). Utilizan mensajería basada en XML como medio fundamental de comunicación de datos para contribuir a reducir las diferencias existentes entre diferentes entornos de trabajo.

Un servicio web puede ser implementado con el empleo de cualquier lenguaje de programación, teniendo en cuenta las características y facilidades del utilizado.

Entre los múltiples lenguajes se encuentran: **ASP.Net**, **JSP** (*Java Server Pages*), **Perl**, **PHP**, **Python**, **Perl**, entre otros.

### ***1.3.7 Fundamentación del lenguaje de programación a utilizar***

Para la implementación del servicio web se empleó como lenguaje de programación PHP ya que es uno de los más extendidos en la red de redes y ha sido aceptado precisamente por la simplicidad y potencia que lo caracteriza, ofrece gran variedad de funciones para la explotación de bases de datos sin grandes complicaciones. Es un lenguaje multiplataforma completamente gratuito que puede ser ejecutado en la mayoría de los sistemas operativos tales como UNIX, Windows y Mac OS X, y puede interactuar con los servidores de web más populares, pues existe en versión CGI, módulo para Apache, e ISAPI.

Fue lanzado bajo la licencia BSD, no obliga a entregar el código fuente, pero sí impone la irritable cláusula publicitaria de este tipo de Licencia. Permite la conexión a diferentes tipos de servidores de bases de datos tales como: MySQL, Postgre SQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones web muy robustas (11). Es un lenguaje fácil de aprender y de aplicar, consume pocos recursos y con gran rapidez de ejecución, contiene funciones para trabajar virtualmente con todas las tecnologías para la web existentes hoy, por lo que es muy empleado para el desarrollo de servicios web.

### ***1.3.8 Fundamentación del Gestor de Bases de Datos a utilizar***

Sin dudas MySQL desarrollado por MySQL AB, bajo la licencia GNU GPL y multiplataforma. Mediante el uso de determinadas APIs se pueden acceder a las bases de datos MySQL a través de diferentes lenguajes de programación siendo script o no; o a través de cualquier lenguaje que soporte ODBC. Es un sistema multihilo y multiusuario con más de seis millones de instalaciones. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL, es ideal

para este tipo de aplicaciones precisamente porque cuando utiliza el motor no transaccional MyISAM es muy rápido en la lectura. Consta ya de varias versiones y las últimas que han visto la luz en el mercado sólo se les ha corregido errores críticos, no se les han añadido ninguna nueva funcionalidad.

### **1.3.9 Fundamentación de la metodología a utilizar**

En la actualidad existe en el mundo un marcado progreso en el desarrollo de software, pero crearlo con la calidad requerida y que los clientes queden satisfechos con el trabajo logrado no es tarea fácil si no se sigue una metodología que indique qué pasos seguir en cada etapa de desarrollo.

En el mundo, en dependencia del tiempo de vida y la complejidad del proyecto que se vaya a desarrollar se proponen diferentes metodologías, pudiendo ser tradicionales o ágiles, dentro de las ágiles se encuentran: Programación extrema o XP, formulado (Extreme Programming), es la más destacada de las de su tipo; enfatiza mucho más la adaptabilidad que la previsibilidad también está Microsoft Solution Framework entre otras. Dentro de las tradicionales está RUP, que a continuación se fundamenta siendo la que se pretende utilizar en el presente trabajo.

#### **1.3.9.1 RUP**

Constituye la metodología tradicional estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos (12).

Se caracteriza por ser iterativa e incremental, centrada en la arquitectura y guiada por los casos de uso. Divide el proceso de desarrollo en ciclos de vida, obteniendo un producto al final de cada ciclo, los cuales se dividen en fases que deben de terminar con un hito, dentro de estas fases se encuentran: Inicio, Elaboración, Construcción y Transición. Es una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Unifica los mejores elementos de metodologías anteriores, está preparada para desarrollar grandes y complejos proyectos y utiliza el UML como lenguaje de representación visual para preparar todos esquemas de un sistema software.

De forma general proporciona una guía para ordenar las actividades del equipo de trabajo, dirige las tareas de cada desarrollador por separado y del equipo como un todo, especifica los artefactos que deben desarrollarse y ofrece criterios para el control y la medición de los productos y actividades de proyectos.

### 1.3.9.2 *UML*

Es un lenguaje gráfico que permite visualizar, especificar y documentar cada una de las partes que comprende el desarrollo del software. Es un lenguaje que permite modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, cosas concretas como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables, así como distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Dentro de sus objetivos tiene el de brindar un material de apoyo que le permita al lector poder definir diagramas propios como también entender diagramas ya existentes. Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

### 1.3.10 *Herramientas a utilizar*

Las Herramientas CASE (**C**omputer **A**ided **S**oftware **E**ngineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas. Dentro de estas herramientas una de la que más se destaca y se utiliza hoy en día es: Visual Paradigm.

#### 1.3.10.1 *Visual Paradigm*

Es una herramienta que da soporte al modelado visual con UML 2.0 y tiene las siguientes ventajas:

- Presenta un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Usa un lenguaje estándar común a todo el equipo de desarrollo y facilita la comunicación.
- Tiene capacidades de ingeniería directa (versión profesional) e inversa.
- Tiene modelos y códigos que permanecen sincronizados en todo el ciclo de desarrollo
- Presenta disponibilidad de múltiples versiones, para cada necesidad y en múltiples plataformas (13).



## **2 CARACTERÍSTICAS DEL SISTEMA**

En el presente capítulo, se hace un estudio del Centro Rector de Universidad para Todos y Canales Educativos y se representa el flujo actual de los procesos involucrados en el control de acceso con un análisis crítico de su ejecución, se detallan los que serán objetos de automatización, así como la información que se maneja. Además se expone el modelo del negocio relacionado con las descripciones realizadas y la propuesta del sistema (servicio web) que se pretende implementar, los requisitos funcionales y no funcionales y los casos de usos correspondientes. De forma general se abordan las características del sistema dejando muy clara la propuesta final que se pretende.

### **2.1 Centro Rector de Universidad para Todos y Canales Educativos**

El objeto de estudio de este trabajo está enmarcado en la seguridad y protección del Centro Rector de Universidad para Todos y Canales Educativos. Dicha institución, que anteriormente era utilizado como centro escolar, fue destinada por la dirección del país desde el año 2003 para que se estableciera como una infraestructura tecnológica y física, que dentro de su misión estuviera la de ser un centro coordinador de las potencialidades institucionales y profesionales creadas por la Revolución para su mejor utilización en los cursos de Universidad para Todos, en función de elevar la calidad de la educación y la Cultura General Integral de la población cubana, así como tener la responsabilidad de llevar a cabo la gestión y el procesamiento del conocimiento que necesitan los profesores de los Canales Educativos, y Universidad para Todos para su desempeño laboral, a partir del óptimo empleo de la nuevas tecnologías de la información y las comunicaciones instaladas. Teniendo en cuenta la existencia de estos medios tecnológicos, existe un conjunto de personas comprometidas del Departamento de Seguridad y Protección (DSPF) a velar por la seguridad de dicha institución.

### **2.1.1 Objetivos estratégicos del DSPF y procesos de negocio que los soportan**

Dentro de los objetivos estratégicos perseguidos por el Departamento de Protección y Seguridad Física (DSPF) se encuentran:

- Controlar las personas que entran y salen del centro a través del registro de sus datos.
- Garantizar el cumplimiento de las políticas de seguridad y protección del Centro Rector de Universidad para Todos y Canales Educativos y proteger sus recursos disponibles.

Los procesos del negocio que soportan estos objetivos son los de: Entrar y Salir de la Institución, a través de estos se controlan las personas que entran y salen de la institución registrando sus datos en la entrada principal.

### **2.1.2 Flujo actual y análisis de los procesos del negocio involucrados en el campo de acción**

Al Centro Rector de Universidad para Todos y Canales Educativos hoy por hoy acceden personas (trabajadores, profesores y visitantes) que necesitan ser controladas para que no pasen sin autorización, en cuanto a este tema existen ineficiencias ya que se desarrolla todo el proceso de forma manual, tarea un tanto tediosa y poco confiable. En la actualidad llevar el control de acceso, es responsabilidad de un Agente de Seguridad, permitiendo la entrada y la salida a todo el personal autorizado.

El proceso de **Entrar a la Institución** se lleva a cabo cuando una de las personas pretende acceder. En caso de ser un trabajador, al llegar a la garita el Agente de Seguridad le solicita la credencial, verifica los datos y archiva en el Registro de Entradas y Salidas: nombre y apellidos, fecha y hora de entrada.

Esto puede traer como consecuencia que en ocasiones no se chequee correctamente la validez de la credencial, o que simplemente alguien pueda portar otra que no sea la suya, posibilitando que entren a la institución personas ajenas que pudieran poner en peligro la seguridad y protección de los medios tecnológicos de los cuales depende el funcionamiento de la institución.

En caso de llegar un visitante, el Agente de Seguridad verifica si la visita está prevista, siendo así, ya tiene sus datos registrados, él solamente los chequea solicitándole la credencial, para éste es el carnet de identidad (CI) y le autoriza la entrada; en caso de ser imprevista, el Agente de Seguridad se comunica vía teléfono con el Directivo del área, o la persona que va a ser visitada, donde le confirman si puede o no ser atendida, en caso de ser afirmativa la respuesta le recoge los datos necesarios: nombre y apellidos, CI, hora de entrada, lugar al que se dirige, la fecha y los archiva en el documento: Registro de Entradas y Salidas. Teniendo en cuenta la posibilidad de que la persona que vaya a entrar sea un profesor, el Agente de Seguridad contacta vía teléfono con algún directivo del Departamento de Gestión y Servicios de Información (DGSI), donde le confirman la entrada del mismo, le recoge los datos necesarios coincidentes con los de los visitantes y los archiva en el Registro de Entradas y Salidas, el profesor siempre entra a la institución a recibir servicios pero debe esperar por el autorizo.

Todo este proceso de entradas al Centro Rector de Universidad para Todos y Canales Educativos por parte de visitantes y profesores, es una manera ineficiente y poco segura pues una vez estando dentro de la institución, no existe un mecanismo para identificar que solamente accedan al área que le fue autorizada, pudiendo traer como consecuencia que puedan llevar a cabo acciones no establecidas o no autorizadas.

El proceso de **Salir de la Institución** es cuando las personas una vez culminada su actividad dentro de la institución se dirigen a la garita, le muestran al Agente de Seguridad la identificación, éste lo localiza en el Registro de Entradas y Salidas y archiva la hora de salida.

Estos procesos desarrollados de forma manual, se vuelven una actividad engorrosa que puede provocar en ocasiones que toda la información archivada pueda ser extraviada, o deteriorada con el tiempo, trayendo consigo que no se puedan realizar reportes estadísticos.

## **2.2 Objeto de automatización.**

### **2.2.1 Descripción de los procesos a automatizar**

Se decidió como parte del proceso de informatización llevado a cabo en este Centro, gestionar toda la información referente al control de acceso, se pretende:

Automatizar los procesos de **Entrar y Salir de la Institución**

Estos procesos permiten controlar las entradas y salidas de profesores, trabajadores y visitantes que acceden al Centro Rector de Universidad para Todos y Canales Educativos a través del registro de los datos personales de las mismas, horarios de entrada, de salida, fecha y verificación de las credenciales a la entrada y salida.

En estos momentos en la institución no existe ningún software vinculado con el campo de acción correspondiente a este tema, ya que todos los procesos son llevados de forma manual.

### **2.2.2 Información que se maneja.**

La información que se maneja es la relacionada con estos procesos, es precisamente el documento Registro de Entradas y Salidas, donde el Agente de Seguridad registra todos los datos referentes al personal que entra y sale de la Institución, en dicho documento se archivan: nombre y apellidos, fecha, hora de entrada y hora de salida, lugar al cual se dirige, así como algunas observaciones en casos pertinentes. No constituye información de gran complejidad pero sí un tanto tediosa tenerla en papel, ya que es un proceso efectuado diariamente.

## **2.3 Descripción del sistema propuesto**

En la presente investigación, con el propósito de darle cumplimiento al problema científico planteado y de acuerdo a los estudios realizados, se propone el desarrollo de un servicio web con el objetivo de gestionar el control de acceso al Centro Rector de Universidad para Todos y Canales Educativos.

Se plantea esta solución partiendo en primer lugar de la situación actual de la institución, ya que existen constantes cambios en cuanto a las tecnologías de desarrollo de la intranet, a la cual se pretenden integrar cada uno de los módulos desarrollados bajo el lenguaje de programación PHP, en este caso se encuentra el control de acceso y teniendo presente las ventajas de los servicios web

ya que tienen la capacidad de soportar la interoperabilidad máquina - máquina a través de una red logrando independencia de sistemas operativos, lenguajes de programación y microprocesadores previendo así la necesidad de que el software no sufra cambios severos a la hora de tener que hacer una migración, pues solo cambiaría la capa de presentación y no, los demás paquetes y componentes.

Este servicio web expondrá funciones que permitirán registrar las áreas sobre las cuales se tendrá vigilancia a través de bloques de accesos que se definan, garantizando el control del personal, registrar los horarios de entrada y salida, fechas de acceso, nombre y apellidos del que accede, detectar la posible entrada de personas ajenas a la institución, realizar búsquedas de datos que se deseen conocer de personas registradas (trabajadores, visitantes, directivos, profesores), las cantidades por estas categorías que han entrado y las que están dentro del sistema un momento determinado. Además brindará la posibilidad de gestionar los niveles de accesos de personas por diferentes áreas, garantizando un control total de los accesos efectuados.

Para llevar a cabo el desarrollo de esta propuesta, se partió del análisis realizado de los procesos que se desarrollan en la institución referentes al control de acceso y las necesidades que existían, se tuvo en cuenta el estudio de algunos software y sistemas de control de accesos que existen en Cuba y en el Mundo, los cuales fueron reflejados en el capítulo 1, con el objetivo de tener conocimientos de los mismos en cuanto a funcionalidades y características, destacando que dentro de las investigaciones efectuadas no se detectaron servicios web similares al que se desea implementar. El sistema más próximo y del cual se pudo realizar un estudio más profundo, fue el Sistema de Control de Acceso Centralizado empleado en algunos de los laboratorios de proyectos de la Universidad de las Ciencias Informáticas cuyas funcionalidades fueron abordadas en el estudio realizado en el capítulo 1.

De este análisis se concluye que los sistemas de control de acceso estudiados son sistemas monolíticos, construidos en un mismo lenguaje de programación, y que forman una aplicación indivisible. Nuestra propuesta de utilizar un servicio web está basada fundamentalmente en la necesidad de tener estos módulos en un entorno desacoplado, que permita una independencia de lenguaje y sistema operativo, de acuerdo con los requerimientos del sistema

## 2.4 Modelo de negocio

El modelo del negocio da una visión de lo que es necesario hacer para dar respuesta a las solicitudes del cliente.

### 2.4.1 Actores del negocio

Los actores son personas, individuos, grupos o entidades con los que el negocio interactúa. Los involucrados en el descrito son:

**Tabla 2.1** Actores del negocio

Actores del Negocio	Justificación
Profesor	Persona que accede a la institución a recibir servicios con el objetivo de prepararse.
Visitante	Persona que no pertenece a la institución, que pretende visitar a alguien dentro del mismo o algo en específico.
Trabajador	Persona que labora en la institución, tiene una credencial (solapín) con sus datos personales.
Usuario	Es la generalización de trabajador, profesor y visitante, inicia los casos de uso Entrar a la institución y Salir de la institución.

### 2.4.2 Trabajadores del negocio

Los trabajadores son las personas responsables del correcto funcionamiento de los procesos llevados a cabo en el negocio.

**Tabla 1.2** Trabajadores del negocio

Trabajadores del Negocio	Justificación
Agentes de Seguridad	Son los responsables de la seguridad en el Centro Rector de Universidad para Todos y Canales Educativos, los encargados de verificar las credenciales de las personas que entran y salen de la institución llevando a cabo el control de todo este proceso en el punto de entrada y registrando sus datos personales está relacionado con todos los

procesos del negocio, llevando a cabo el desarrollo de los mismos.

## 2.5 Diagrama de Casos de Uso del Negocio

Se representan gráficamente los casos de uso del negocio identificados y su interacción con los actores a través del presente diagrama.

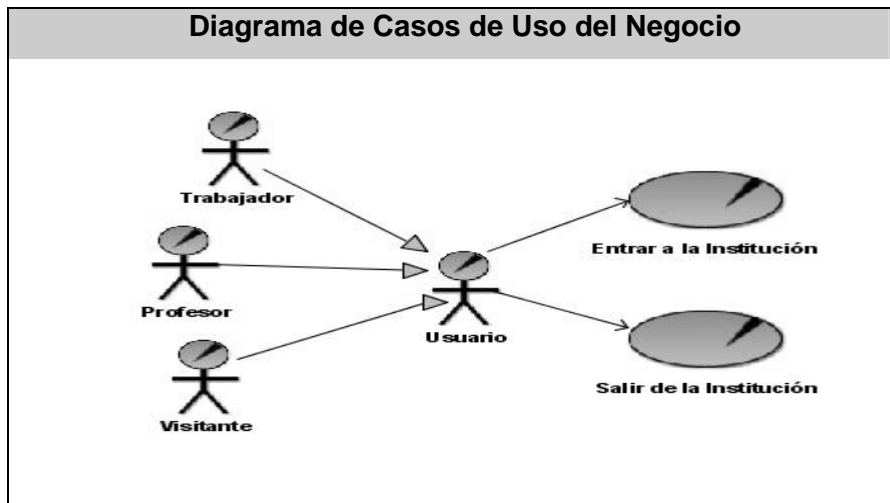


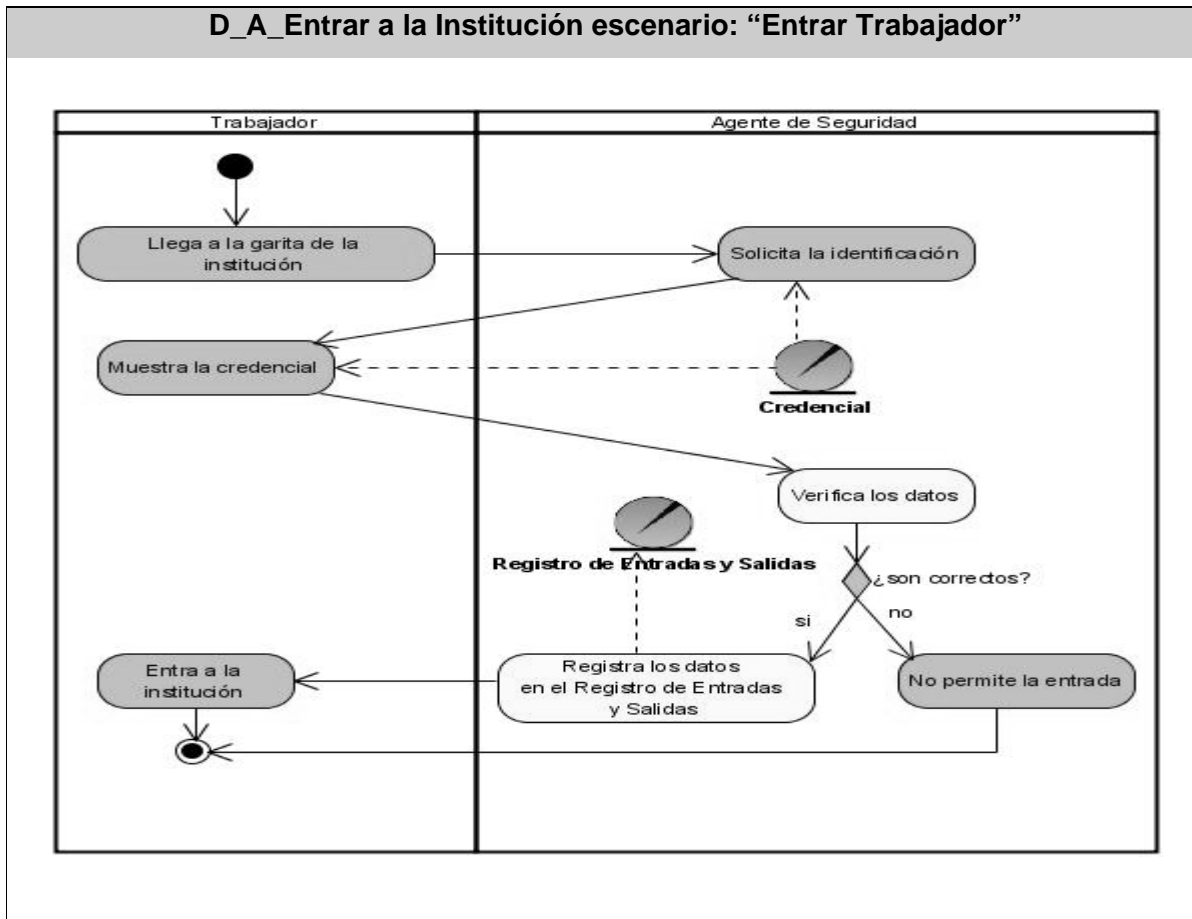
Figura 2.1 Diagrama de Casos de Uso del Negocio

## 2.5 Descripción de los casos de usos del negocio

Para entender el desarrollo de los procesos del negocio reflejados, se realiza una descripción de los mismos, teniendo en cuenta que permiten llegar a obtener una visión común sobre el problema en sí y qué es lo que realmente se necesita, pudiéndose encontrar en el **ANEXO I**.

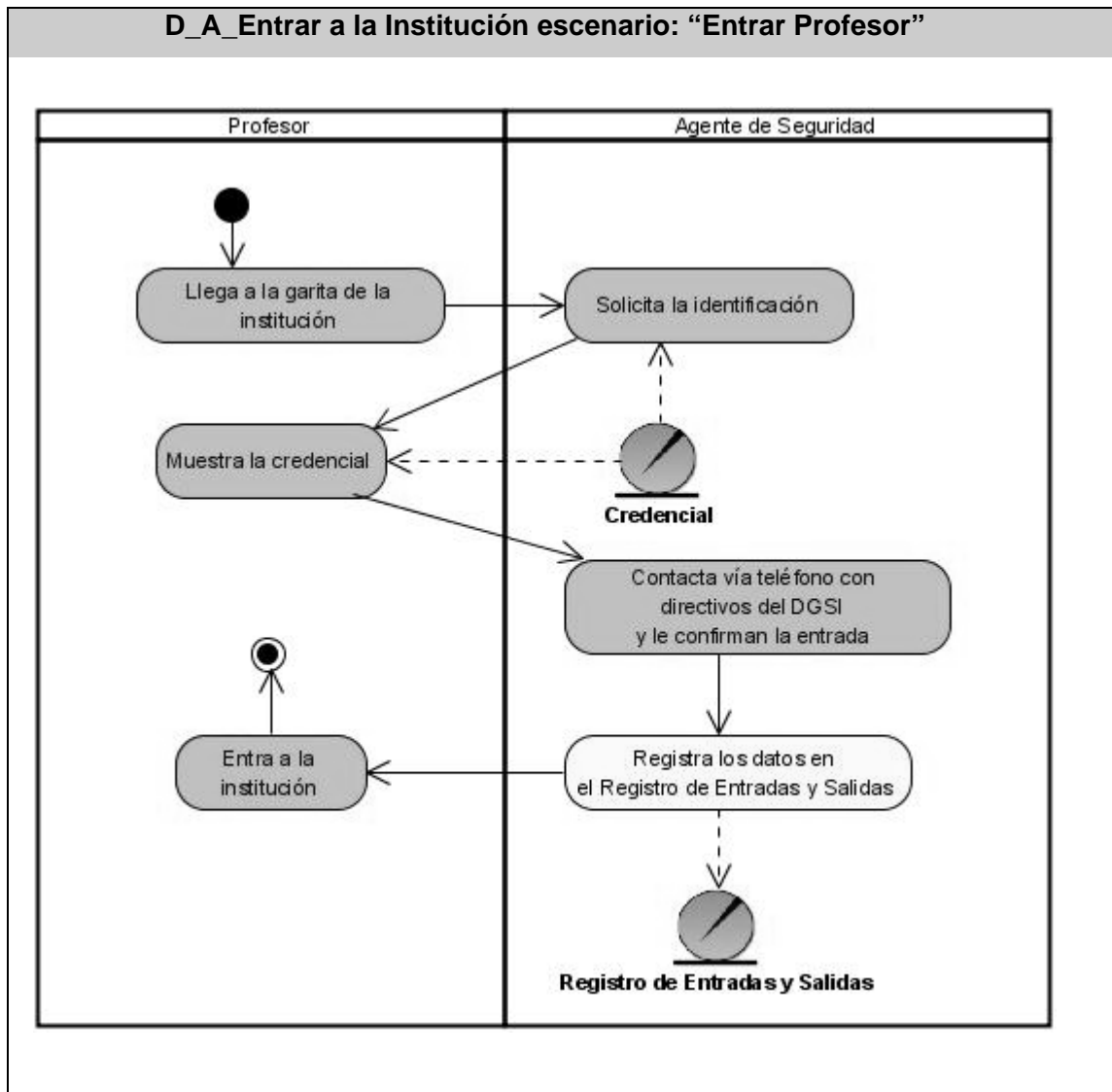
## 2.6 Diagramas de actividades

Se representan los diagramas de actividades de cada uno de los casos de uso del negocio y teniendo en cuenta que el proceso de Entrar a la Institución se describe por escenarios se muestra un diagrama por cada uno de ellos.



**Figura 2.2** Diagrama de actividad escenario: Entrar Trabajador





**Figura 2.3** Diagrama de actividad escenario: Entrar Profesor

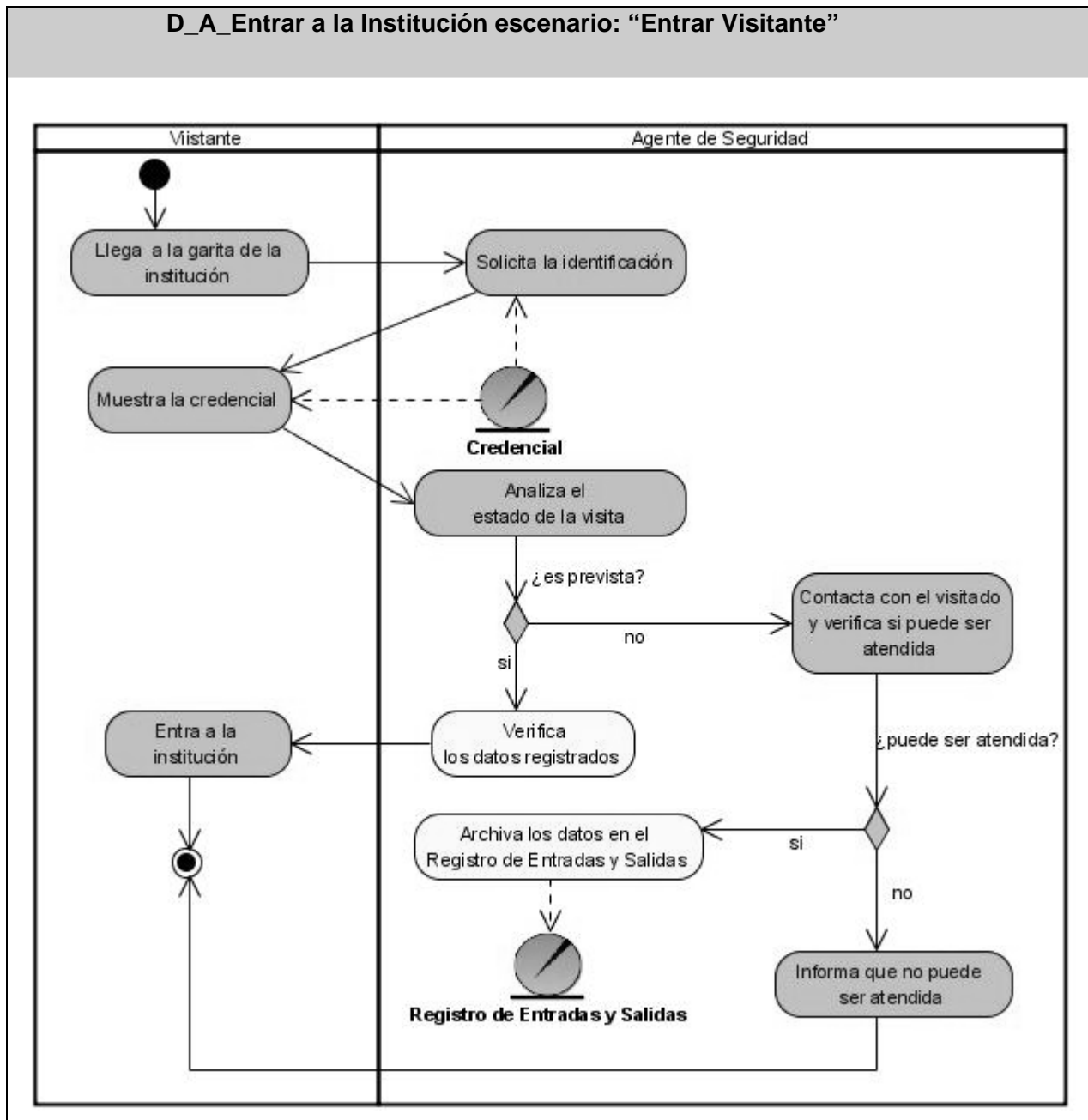
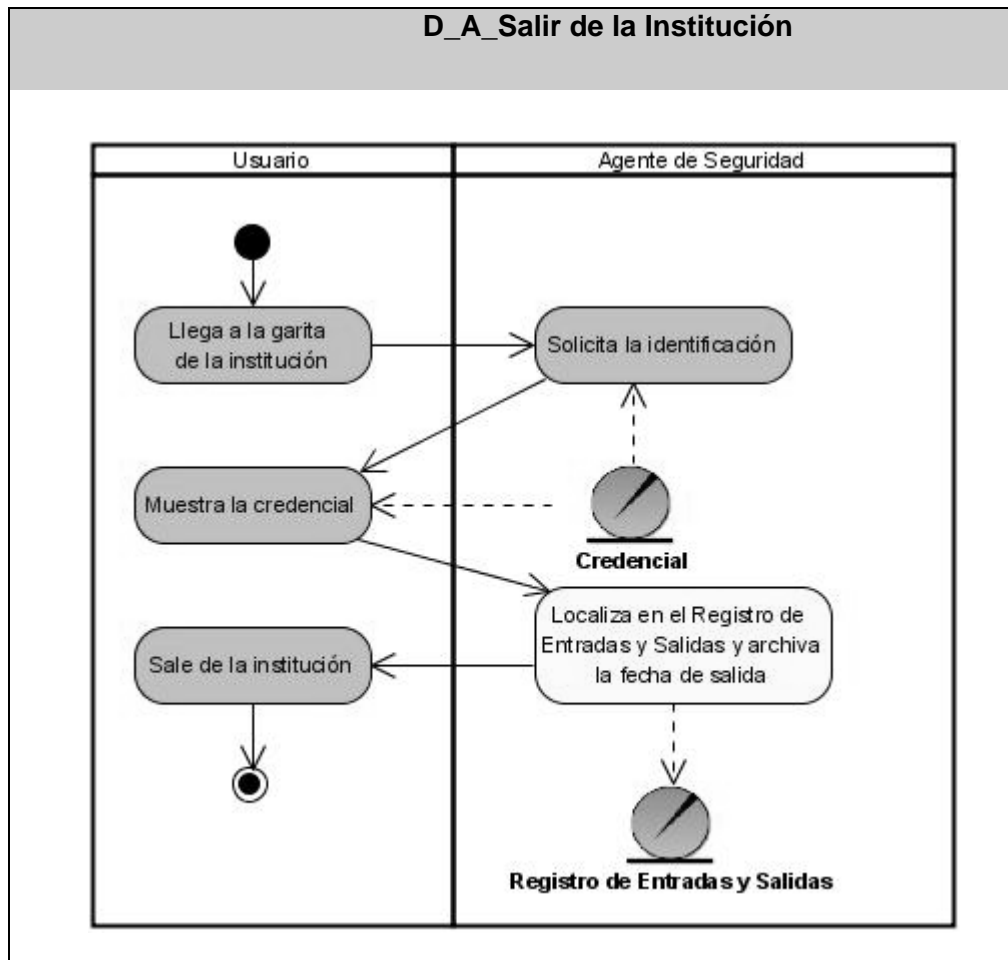


Figura 2.4 Diagrama de actividad escenario: Entrar Visitante



**Figura 2.5** Diagrama de actividad Salir de la Institución

## 2.7 Diagrama de clases del modelo de objetos

El diagrama de clases del modelo de objetos es el correspondiente al negocio que se describe, en él se demuestra cómo colaboran los trabajadores y entidades involucradas.

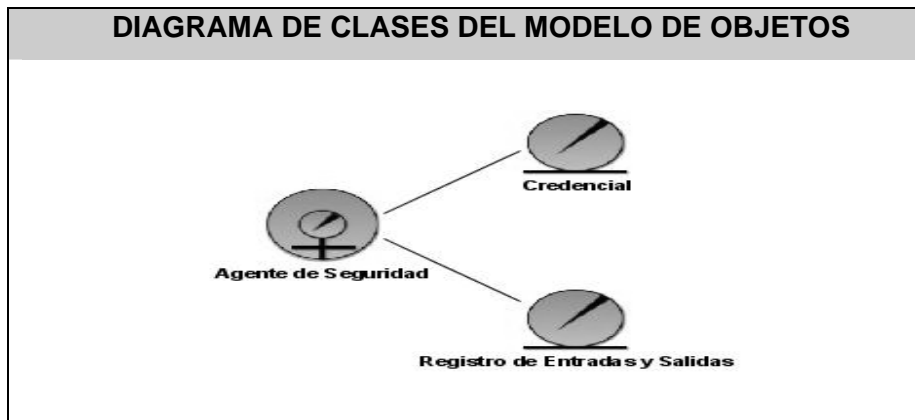


Figura 2.6 Diagrama de Clases del Modelo de Objetos

## 2.9 Especificación de Requisitos.

A continuación se enumeran los requisitos funcionales y no funcionales correspondientes al sistema propuesto.

### 2.9.1 Requerimientos Funcionales

Los requerimientos funcionales son capacidades o condiciones que un sistema determinado debe cumplir. El sistema debe brindar funciones capaces de:

**RF\_1** Registrar las entradas y salidas de personas que acceden al Centro Rector de Universidad para Todos y Canales Educativos, dados los siguientes datos: Solapín, CI, dirección IP del bloque de acceso, Fecha, Hora, quien autoriza.

**RF\_1.1** Verificar la credencial de la persona que entra al pasar por los bloques de acceso, solo registrando los datos anteriores al pasar por el bloque principal.

**RF\_1.2** Manejo de anti\_pass back.

**RF\_2** Registrar áreas de acceso dado el siguiente dato: Descripción área (nombre área).

**RF\_3** Modificar áreas de acceso dada: Descripción área.

**RF\_4** Eliminar las áreas de acceso dado el Id del área de la que se desea eliminar.

**RF\_5** Registrar bloques de acceso, dados los siguientes datos: Descripción, arreglo de áreas asignadas, la dirección IP del bloque de acceso.

**RF\_6** Modificar los Bloques de Acceso dados los siguientes datos: Descripción, arreglo de áreas asignadas, la dirección IP del bloque de acceso.

**RF\_7** Eliminar Bloques de Acceso dado el id del bloque de acceso que se desee eliminar.

**RF\_8** Asignar nivel de acceso por áreas al personal permanente que accede a la institución (trabajadores y profesores) dados los siguientes datos: carnet de identidad (CI), Solapín, Categoría, áreas a las cuales va a acceder y área principal.

**RF\_9** Modificar el nivel acceso teniendo en cuenta que la persona no haya entrado al sistema, permitiendo modificar datos como: CI, Solapín, Categoría, áreas a las cuales va a acceder y área principal.

**RF\_10** Eliminar el nivel de acceso de una persona dado su CI y Solapín, teniendo en cuenta que no haya entrado al sistema.

**RF\_11** Registrar visitantes dados los siguientes datos: CI, Solapín, Nombre, Apellido, quien autoriza, Categoría (invitado), áreas a las cuales va a tener acceso y área principal.

**RF\_12** Actualizar los datos de los visitantes dados: CI, Solapín, Nombre, Apellido, quien autoriza, Categoría (invitado), áreas a las cuales va a tener acceso y área principal.

**RF\_13** Eliminar datos de visitantes dado como datos el Solapín y CI.

**RF\_14** Buscar áreas dado como dato: Descripción área.

**RF\_15** Buscar bloques de acceso dados como dato su: Descripción.

**RF\_16** Buscar visitantes registrados dado como datos: el carnet de identidad (CI), Nombre y Solapín.

**RF\_17** Buscar datos de personas que hayan sido registradas, dado su Solapín como parámetro de búsqueda.

**RF\_18** Mostrar la cantidad de personas por categorías (Profesores, Trabajadores, Directivos, Visitantes) que hay dentro del sistema un momento determinado.

**RF\_18.1** Mostrar cantidad de Profesores

**RF\_18.2** Mostrar cantidad de Trabajadores

**RF\_18.3** Mostrar cantidad de Directivos

**RF\_18.4** Mostrar cantidad de Visitantes

**RF\_19** Realizar búsquedas de personas que han accedido a la institución un momento determinado por: fecha y /o categorías o combinaciones de ellas.

### 2.9.2 Requerimientos no Funcionales

Los requerimientos no funcionales son propiedades que el sistema debe poseer, se encuentran separados por categorías.

#### 1. Software:

- Los servicios web están desarrollados sobre el lenguaje de programación PHP 4.0 y PHP 5.0.
- Los servicios web utilizarán como base de datos MySQL.
- Se utilizará como servidor Web Apache 2.0.
- La intranet será montada sobre el CMS Drupal.

#### 2. Seguridad y Privacidad:

- El acceso será controlado con nombres de usuario y contraseñas. Solo los usuarios con derechos de administrador podrán acceder a las funciones administrativas, los usuarios normales no podrán.
- Se establecerá una llave privada entre el cliente y el proveedor del servicio web, para evitar acceso no autorizado a los servicios.

#### 3. Portabilidad:

- La portabilidad se observa en los servicios web, los cuales podrán ser accedidos desde cualquier protocolo de transporte que sea capaz de transportar texto, como por ejemplo (http, https, SMTP, etc.). El sistema utilizará http sobre TCP en el puerto 80. Esto se hace debido a que en esta organización se protegen las redes mediante firewalls y se cierran casi todos los puertos TCP salvo el 80 que nunca es bloqueado porque es el que utilizan los navegadores.
- El diseño del servicio web permite que el mismo pueda funcionar tanto en sistemas operativos libres como propietarios.

#### **4. Restricciones en el diseño y la implementación:**

- Se usa el lenguaje de programación PHP tanto para los servicios como para la aplicación que los consume.
- Se usa la librería NuSoap para la creación de los servicios web.
- Se usa la librería ADOdb para la abstracción a datos.

#### **5. Usabilidad:**

- El uso de servicios web permite que la aplicación pueda consumir estos aunque esté escrita en una plataforma de desarrollo diferente a ellos, pues su protocolo de comunicación está basado en XML y es compatible con todas las plataformas.
- El uso de los servicios web es muy práctico ya que pueden aportar gran independencia entre la aplicación que usa el servicio web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro.

#### **6. Legales:**

- La aplicación con todos sus módulos y toda la documentación generada pertenecen al proyecto del Centro Rector de Universidad para Todos y a la Universidad de las Ciencias Informáticas.

#### **7. Confiabilidad:**

- La aplicación debe estar disponible las 24 horas de forma tal que se pueda acceder a todos sus servicios, y así aprovechar todas sus funcionalidades.

#### **8. Importación y exportación de datos:**

- El sistema deberá almacenar todos los datos en una base de datos MySQL Server, donde puedan ser accedidos por los servicios.

#### **9. Hardware:**

- El servidor donde estarán montados los servicios requiere como mínimo de RAM 1 GB.
- El servidor de Base de Datos requiere como mínimo 5 GB para almacenar los mismos.

## 2.10 Definición de los casos de uso del sistema

### 2.10.1 Actores del sistema

**Tabla 2.3** Actores del Sistema

Actores	Justificación
Intranet	La Intranet es el actor que inicia los casos de uso invocando los servicios que permiten ejecutarlos beneficiándose de ellos.

### 2.10.2 Casos de usos del sistema

Los casos de uso son fragmentos de funcionalidad que un sistema ofrece. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con el sistema, en este caso el servicio web, los mismos fueron determinados de acuerdo a funciones que pudiera brindar.

**Tabla 2.4** CU del Sistema. Registrar Accesos

CU_1	Registrar Accesos
Actor	Intranet
Descripción	Se inicia cuando el actor necesita verificar si la persona tiene acceso al área solicitada, para ello invoca el servicio registrar accesos enviando los datos pertinentes, verifica que la persona tenga acceso al área y se registra el acceso si es el bloque principal, finalizando el caso de uso.
Referencia	RF_1, RF_1.1, RF_1.2



**Tabla 2.5** CU del Sistema. Gestionar áreas

CU_2	Gestionar áreas
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia cuando el actor desea realizar algunas de las siguientes operaciones: registrar un área nueva de acceso, modificar alguna que ya esté registrada o eliminar, en caso de registrar invoca el servicio insertar área, envía los datos pertinentes, ejecuta la acción y retorna un identificador del área (id área), en caso de modificar o eliminar invoca los servicios actualizar área o eliminar área dado los datos enviados actualizando la base de datos, finalizando el caso de uso.
<b>Referencia</b>	RF_2 RF_3 RF_4

**Tabla 2.6** CU del Sistema. Buscar áreas

CU_3	Buscar áreas
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia cuando el actor desea buscar un área determinada e invoca el servicio buscar áreas enviando la descripción del área y devuelve el área perteneciente a este dato, finalizando el caso de uso.
<b>Referencia</b>	RF_14

**Tabla 2.7** CU del Sistema. Gestionar Bloques de Acceso

CU_4	Gestionar Bloques de Acceso
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia cuando el actor desea realizar alguna de las siguientes operaciones: registrar, modificar o eliminar bloques de acceso, si decide registrar, invoca el servicio insertar bloques de acceso enviando los datos pertinentes, verifica que los mismos sean correctos y los inserta en la BDCA retornando su identificador (id bloque acceso), en caso de invocar los servicios actualizar bloque de acceso o eliminar bloque de acceso, de acuerdo al dato enviado actualiza la Base de Datos finalizando el caso de uso.

<b>Referencia</b>	RF_5 RF_6 RF_7
-------------------	----------------

**Tabla 2.8** CU del Sistema. Buscar bloques de acceso

<b>CU_5</b>	<b>Buscar bloques de acceso</b>
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia por el actor cuando va a buscar un bloque de acceso determinado e invoca el servicio buscar bloques de acceso dada la Descripción del bloque enviado y se devuelve, finalizando el caso de uso.
<b>Referencia</b>	RF_15

**Tabla 2.9** CU del Sistema. Gestionar niveles de acceso

<b>CU_6</b>	<b>Gestionar niveles de acceso</b>
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia cuando el actor desea realizar alguna de las siguientes operaciones : asignar nivel de acceso por área al personal (trabajadores y profesores) permanentes de la institución, modificar el nivel acceso, o eliminarlo, en caso de asignar invoca el servicio insertar acceso persona y dado los datos enviados de la persona asigna el nivel de acceso, en caso de modificar o eliminar, invoca los servicios actualizar acceso persona o eliminar acceso persona en caso de no haber sido verificada su entrada al sistema y en dependencia de los datos enviados por el actor actualiza o elimina de la BDCA, finalizando el caso de uso.
<b>Referencia</b>	RF_8 RF_9 RF_10

**Tabla 2.102** CU del Sistema. Buscar datos personales

<b>CU_7</b>	<b>Buscar datos personales</b>
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia por el actor cuando va a buscar los datos de una persona para verificar el acceso o no, invocando el servicio buscar datos personales, determinando la categoría de la persona en cuestión, y realizando la búsqueda con dicha categoría, si es trabajador, visitante o profesor a través del

	servicio <b>“busqueda_simple”</b> del Servicio web Gestion_Usuario y en la Base de datos Control de acceso en caso de la categoría ser un visitante.
<b>Referencia</b>	RF_17

**Tabla 2.11** CU del Sistema. Gestionar datos visitantes

<b>CU_8</b>	<b>Gestionar datos visitantes</b>
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia cuando el actor desea realizar algunas de las siguientes operaciones: registrar datos de visitante, modificarlos o eliminarlos, en caso de registrar invoca el servicio insertar visitante y dado los datos enviados se registraría en la base de datos, en caso de querer modificar o eliminar invoca los servicios actualizar visitante o eliminar visitante verificando previamente que el visitante no haya entrado al sistema y dado estos datos, actualiza en la base de datos las modificaciones o elimina, finalizando de esta forma el caso de uso.
<b>Referencia</b>	RF_11 RF_12 RF_13

**Tabla 2.12** CU del Sistema. Buscar visitantes registrados

<b>CU_9</b>	<b>Buscar visitantes registrados</b>
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia por el actor cuando va a buscar un visitante determinado e invoca el servicio buscar visitantes registrados enviando los datos pertinentes, se realiza la búsqueda en la base de datos control de acceso, se devuelven los datos esperados, finalizando el caso de uso.
<b>Referencia</b>	RF_16

**Tabla 2.13** CU del Sistema. Buscar accesos

<b>CU_10</b>	<b>Buscar accesos</b>
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia cuando el actor desea buscar las personas que han accedido a la

	institución un momento determinado, la intranet invoca el servicio buscar accesos enviando los datos de búsquedas, ya sea por fecha, categorías o combinaciones de ellos, se devuelven los datos pertinentes finalizando el caso de uso.
<b>Referencia</b>	RF_19

**Tabla 2.14** CU del Sistema. Mostrar estadísticas personas\_ categorías activa

<b>CU_11</b>	<b>Mostrar estadísticas personas_ categorías activa</b>
<b>Actor</b>	Intranet
<b>Descripción</b>	Se inicia cuando el actor desea observar la cantidad de personas de las diferentes categorías (Directivos, profesor, invitados, trabajador) que hay dentro del sistema en un momento determinado, invocando el servicio Mostrar estadísticas personas_ categorías activa devolviendo la cantidad de personas por las diferentes categorías que han entrado al sistema finalizando el caso de uso.
<b>Referencia</b>	RF_18 RF_18.1 RF_18.2 RF_18.3 RF_18.4

## 2.11 Diagrama de casos de uso del sistema

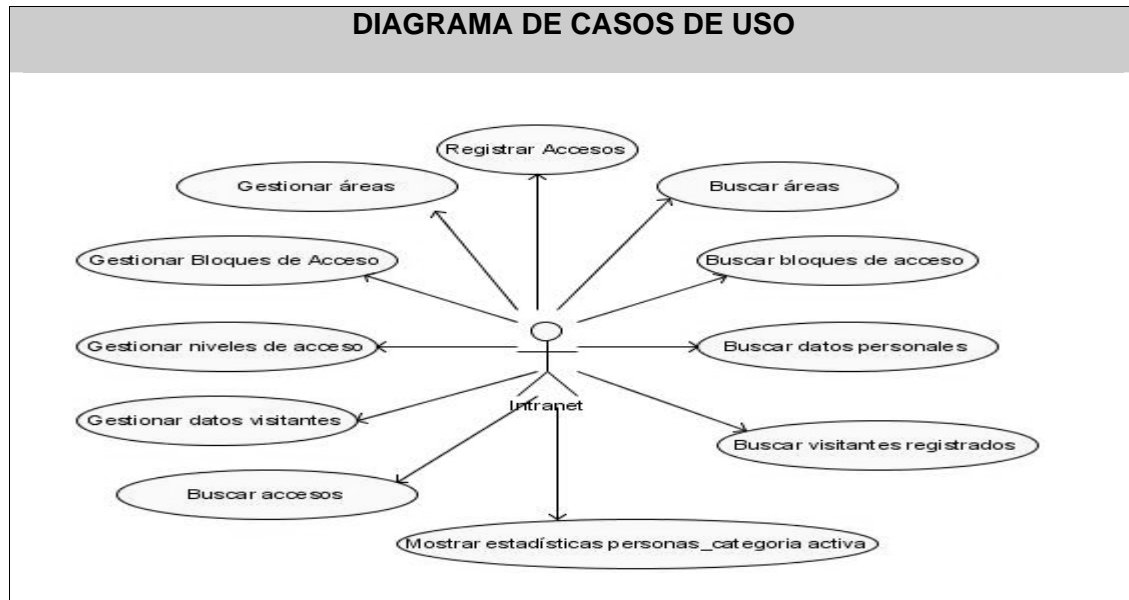


Figura 2.7 Diagrama de casos de uso del sistema

### 2.11.1 Casos de uso por ciclo de desarrollo

Tabla 2.3 Casos de uso por ciclo de desarrollo

Código	Nombre de caso de uso	Justificación de la selección.
CU_1	Registrar Accesos	Casos de usos prioritarios que constituyen mecanismos básicos de funcionamiento del servicio web.
CU_2	Gestionar áreas	
CU_3	Buscar áreas	
CU_4	Gestionar Bloques de Acceso	
CU_5	Buscar bloques de acceso	
CU_6	Gestionar niveles de acceso	

CU_7	Buscar datos personales	
CU_8	Gestionar datos visitantes	
CU_9	Buscar visitantes registrados	
CU_10	Buscar accesos	Casos de usos secundarios que no influyen en el funcionamiento básico del servicio web.
CU_11	Mostrar estadísticas personas_ categorías activa	

### 2.11.2 Casos de Uso expandidos

En el **ANEXO II** se encuentran las descripciones de los casos de uso del sistema pertenecientes al ciclo de desarrollo especificado.

### **3 DISEÑO DEL SISTEMA PROPUESTO**

El presente capítulo se centra fundamentalmente en el diseño del sistema propuesto. Se detallan los casos de usos identificados permitiendo reflejar una vista interna de lo que sería el sistema y se determinan las clases necesarias para llevar a cabo funcionalidades contenidas en ellos. Se representan los diagramas de secuencia, los diagramas de clases del diseño, con las descripciones de las clases identificadas y el diseño de la base de datos a través del Modelo de Datos, con la descripción de las tablas correspondientes.

#### **3.1 Modelo de diseño**

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

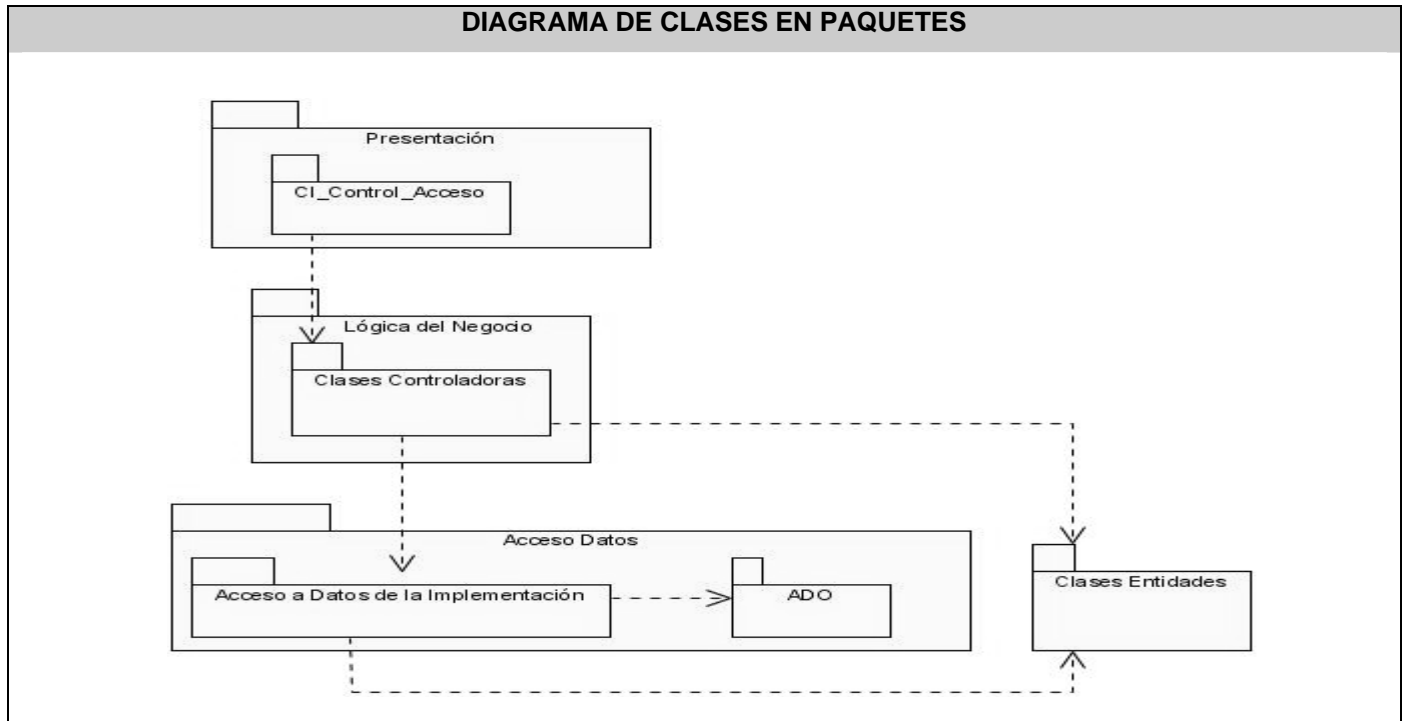
#### **3.2 Diagramas de interacción**

Los diagramas de interacción modelan los aspectos dinámicos del sistema, mostrando las interacciones entre objetos mediante transferencia de mensajes entre ellos. Dentro de estos diagramas están los de secuencia y los de colaboración. En el presente trabajo se emplearon los **diagramas de secuencia** teniendo en cuenta que muestran las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto, los mismos se pueden encontrar en el **ANEXO III**.

#### **3.3 Diagrama de clases**

Con el fin de comprender mejor el diagrama de clases, se han separado las mismas en 4 paquetes atendiendo a su funcionalidad, dichos paquetes se han dividido a su vez en otros subpaquetes para una

mayor organización y legibilidad, así como para lograr un mayor desacoplamiento y reutilización de los diagramas.



**Figura 3.1** Diagrama de clases en paquetes

El paquete **Acceso Datos** contiene las clases para hacer posible la persistencia y recuperación de objetos. Está dividido en dos subpaquetes: el subpaquete **Acceso a Datos de la Implementación**, que contiene las clases encargadas de acceder a la base de datos para manipular la persistencia de las entidades, y el subpaquete **ADO**, utilizado por el paquete anterior, que es donde se encuentran un conjunto de clases que permiten conectarse a diferentes base de datos. El Paquete *AccesoDatos* en general permite a la aplicación abstraerse del origen de los datos y de la lógica de su persistencia, logrando un bajo acoplamiento entre sus componentes.

El paquete **Lógica del Negocio** contiene la lógica de negocio del servicio web. Contiene un paquete **Clases Controladoras**: que contiene las clases controladoras de las entidades relacionadas en el negocio, estas clases permiten diversas funciones como la validación de datos.



El paquete **Entidades** contiene clases que no tienen comportamiento, sólo propiedades y son representaciones de entidades reales del dominio, la mayoría son clases persistentes que son accedidas por las clases de los paquetes **Lógica del Negocio** y **AccesoDatos**.

Por último, el paquete **Presentación** contiene la clase interfaz del servicio web, donde están registrados todos los servicios que se exponen en WSDL (lenguaje de descripción del servicio web).

### 3.3.1 Diagramas de clases por paquetes

En el **ANEXO IV** se muestran cada uno de los diagramas de clases correspondientes a cada paquete del diagrama general representado anteriormente.

### 3.3.2 Descripción de las clases

**Tabla 3.1** Clase Interfaz. CI\_Control de Acceso

<b>Nombre: CI_Control_Acceso</b>	
<b>Tipo de clase: Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
---	---
<b>Responsabilidades</b>	
insertar_area (Descripcion: xsd:string, Key: xsd:string)	
<b>Descripción:</b> Servicio encargado de insertar un área dado los datos enviados.	
actualizar_area (Id_Area: xsd:int, Descripcion: xsd:string, Key: xsd:string)	
<b>Descripción:</b> Servicio encargado de actualizar un área dado los datos enviados.	
eliminar_area (Id_Area: xsd:int, Key: xsd:string)	
<b>Descripción:</b> Servicio encargado de eliminar un área dado los datos enviados.	
get_all_areas ()	
<b>Descripción:</b> Servicio encargado de devolver todos los datos de las áreas registradas.	
buscar_area(Descripcion: xsd:string)	
<b>Descripción:</b> Servicio encargado de devolver las áreas dado los datos enviados.	
insertar_bloque_acceso( Ip_Bloque_Acceso:xsd:string, Descripcion:xsd:string, Arreglo_Area: tns:Arreglo_Areas, Key: xsd:string)	
<b>Descripción:</b> Servicio encargado de insertar un bloque de acceso dado los datos enviados.	

<p>actualizar_bloque_acceso(Id_Bloque_Acceso: xsd:int, Ip_Bloque_Acceso: xsd:string, Descripcion_Bloque: xsd:string, Arreglo_Area: tns:Arreglo_Areas, Key: xsd:string)</p>
<p><b>Descripción:</b> Servicio encargado de actualizar un bloque de acceso dado los datos enviados.</p>
<p>eliminar_bloque_acceso( Id_Bloque_Acceso: xsd:int, Key: xsd:string)</p>
<p><b>Descripción:</b> Servicio encargado de eliminar un bloque de acceso dado los datos enviados.</p>
<p>buscar_bloque_acceso (Descripcion: xsd:string)</p>
<p><b>Descripción:</b> Servicio encargado de devolver un bloque de acceso dado los datos enviados.</p>
<p>get_categoria_trabajadores()</p>
<p><b>Descripción:</b> Servicio encargado de devolver todas las categorías de las personas registradas en la institución.</p>
<p>get_categoria_visitantes()</p>
<p><b>Descripción:</b> Servicio encargado de devolver todas las categorías de los visitantes.</p>
<p>insertar_visitante(CI: xsd:string, Solapin: xsd:string, Nombre1: xsd:string, Nombre2: xsd:string, Apellido1: xsd:string, Apellido2: xsd:string, Autoriza: xsd:string, Categoria: xsd:int, Arreglo_Area: tns:Arreglo_Areas, Area_Principal: xsd:int, Key: xsd:string)</p>
<p><b>Descripción:</b> Servicio encargado de insertar un visitante dado los datos enviados.</p>
<p>actualizar_visitante(CI: xsd:string, Solapin: xsd:string, Nombre1: xsd:string, Nombre2: xsd:string, Apellido1: xsd:string, Apellido2: xsd:string, Autoriza: xsd:string, Categoria: xsd:int, Arreglo_Area: tns:Arreglo_Areas, Area_Principal: xsd:int, CI_ant: xsd:string Solapin_ant: xsd:string Key: xsd:string)</p>
<p><b>Descripción:</b> Servicio encargado de actualizar un visitante dado los datos enviados.</p>
<p>eliminar_visitante(CI: xsd:string, Solapin: xsd:string, Key: xsd:string)</p>
<p><b>Descripción:</b> Servicio encargado de eliminar un visitante dado los datos enviados.</p>
<p>buscar_visitante(Id: xsd:int, CI: xsd:string, Solapin: xsd:string, Nombre: xsd:string)</p>
<p><b>Descripción:</b> Servicio encargado de devolver los datos del visitante dado los parámetros enviados.</p>
<p>insertar_persona( CI: xsd:string, Solapin: xsd:string, Categoria: xsd:int, Arreglo_Area: tns:Arreglo_Areas, Area_Principal: xsd:int, Key: xsd:string)</p>
<p><b>Descripción:</b> Servicio encargado de insertar una persona dado los datos enviados.</p>
<p>actualizar_persona(CI: xsd:string, Solapin: xsd:string, Categoria: xsd:int, Estado: xsd:int)</p>

Arreglo_Area: tns:Arreglo_Areas, Area_Principal: xsd:int, Key: xsd:string)
<b>Descripción:</b> Servicio encargado de actualizar una persona dado los datos enviados.
eliminar_persona(CI: xsd:string, Solapin: xsd:string,Key: xsd:string)
<b>Descripción:</b> Servicio encargado de eliminar una persona dado los datos enviados.
get_datos_personales(CI: xsd:string, Solapin: xsd:string)
<b>Descripción:</b> Servicio encargado de devolver los datos de una persona dado los datos enviados.
registra_acceso(Solapin: xsd:string, CI: xsd:string, Autoriza: xsd:string, Fecha: xsd:string Ip_Acceso: xsd:string, Key: xsd:string)
<b>Descripción:</b> Servicio encargado de registrar los accesos de las personas dado los datos enviados.
get_activos( )
<b>Descripción:</b> Servicio encargado de devolver cantidad de personas que están activas en el sistema.
buscar_registro(Fecha: xsd:string, Categoria: xsd:int)
<b>Descripción:</b> Servicio encargado de devolver los datos del registro de una persona dado los parámetros enviados.

Tabla 3.2 Clase Entidad. CE\_area

<b>Nombre: CE_area</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_area	var
descrip	var
<b>Responsabilidades</b>	
get_id_area()	
<b>Descripción:</b> Responsable de devolver el id de un área.	
get_descripcion()	
<b>Descripción:</b> Responsable de devolver la descripción de un área.	

**Tabla 3.3** Clase Entidad. CE\_bloque\_acceso

<b>Nombre: CE_bloque_acceso</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_bloque_acceso	var
ip_acceso	var
descripcion	var
Arreglo_areas_Acceso	var
<b>Responsabilidades</b>	
get_id_bloque()	
<b>Descripción:</b> Responsable de devolver el id de un bloque de acceso.	
get_ip_acceso()	
<b>Descripción:</b> Responsable de devolver el numero IP de un bloque de acceso.	
get_descrip()	
<b>Descripción:</b> Responsable de devolver la descripción de un bloque de acceso.	
get_arreglo_areas()	
<b>Descripción:</b> Responsable de devolver las áreas que tiene un bloque de acceso.	

**Tabla 3.4** Clase Entidad. CE\_persona

<b>Nombre: CE_persona</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
ID_Persona	var
CI	var
Solapin	var
Arreglo_areas_Acceso	var
Obj_Nom_Categoria_Persona	var

Obj_Nom_estado	var
Arreglo_areas	var
Area_Principal	var
CI_ant	var
Solapin_ant	var
<b>Responsabilidades</b>	
get_id()	
<b>Descripción:</b> Responsable de devolver el id de una persona.	
get_ci()	
<b>Descripción:</b> Responsable de devolver el carnet de identidad de una persona.	
get_solapin()	
<b>Descripción:</b> Responsable de devolver el numero de solapín de una persona.	
get_nom_categoria()	
<b>Descripción:</b> Responsable de devolver la categoría de una persona, si es visitante, trabajador, profesor o directivo.	
get_nom_estado()	
<b>Descripción:</b> Responsable de devolver el estado de una persona, activo o inactivo.	
get_arreglo_areas()	
<b>Descripción:</b> Responsable de devolver las áreas a las que tiene acceso una persona.	
get_area_principal()	
<b>Descripción:</b> Responsable de devolver el área principal de una persona.	
get_ci_ant()	
<b>Descripción:</b> Responsable de devolver el carnet de identidad anterior de una persona.	
get_solapin_ant()	
<b>Descripción:</b> Responsable de devolver el número de solapín anterior de una persona.	

**Tabla 3.5** Clase Entidad. CE\_visitante

<b>Nombre: CE_visitante</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_visitante	var
ci	var
solapín	var
nombre1	var
nombre2	var
apellido1	var
apellido2	var
autoriza	var
Obj_Nom_Categoria_Persona	var
Obj_Nom_Estado	var
Arreglo_areas_Acceso	var
Area_Principal	var
CI_ant	var
Solapin_ant	var
<b>Responsabilidades</b>	
get_id_visitante()	
<b>Descripción:</b> Responsable de devolver el id de un visitante.	
get_ci()	
<b>Descripción:</b> Responsable de devolver el carnet de identidad de un visitante.	
get_solapin()	
<b>Descripción:</b> Responsable de devolver el numero de solapín de un visitante.	
get_nombre1()	

<b>Descripción:</b> Responsable de devolver el primer nombre del visitante.
get_nombre2()
<b>Descripción:</b> Responsable de devolver el segundo nombre del visitante.
get_apellido1()
<b>Descripción:</b> Responsable de devolver el primer apellido del visitante.
get_apellido2()
<b>Descripción:</b> Responsable de devolver el segundo apellido del visitante.
get_autoriza()
<b>Descripción:</b> Responsable de devolver el nombre de quien autoriza el acceso.
get_objeto_categoria()
<b>Descripción:</b> Responsable de devolver un objeto con la categoría de la persona.
get_estado()
<b>Descripción:</b> Responsable de devolver un objeto con el estado de la persona.
get_arreglo_areas()
<b>Descripción:</b> Responsable de devolver un arreglo de áreas a las que tiene acceso.
get_area_principal()
<b>Descripción:</b> Responsable de devolver el área principal.
get_ci_ant()
<b>Descripción:</b> Responsable de devolver el carnet de identidad anterior de un visitante.
get_solapin_ant()
<b>Descripción:</b> Responsable de devolver el número de solapín anterior de un visitante.

**Tabla 3.6** Clase Entidad. CE\_registro

<b>Nombre: CE_registro</b>	
<b>Tipo de clase: Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_Registros	var
Objeto_Persona	var
Descripcion	var

Autoriza	var
Fecha	var
Ip_bloque	var
<b>Responsabilidades</b>	
get_id()	
<b>Descripción:</b> Responsable de devolver el id de un registro.	
get_objeto_persona()	
<b>Descripción:</b> Responsable de devolver un objeto con el id de la persona registrada.	
get_descripcion()	
<b>Descripción:</b> Responsable de devolver nombre del área principal de la persona registrada.	
get_autoriza()	
<b>Descripción:</b> Responsable de devolver el nombre de la persona que autoriza el acceso.	
get_fecha()	
<b>Descripción:</b> Responsable de devolver la fecha en que se realiza el acceso.	
get_ip()	
<b>Descripción:</b> Responsable de devolver la dirección IP del bloque desde donde se accede.	

Tabla 3.7 Clases Controladoras. dao\_area

<b>Nombre: dao_area</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
---	---
<b>Responsabilidades</b>	
Inserta_Area(area: var)	
<b>Descripción:</b> Responsable de insertar todos los datos correspondientes a un área.	
Actualiza_Area(area: var)	
<b>Descripción:</b> Responsable de actualizar todos los datos correspondientes a un área.	
Elimina_Area(id:var)	
<b>Descripción:</b> Responsable de eliminar un área determinada dado su id.	



get_existe_area(area: var)
<b>Descripción:</b> Responsable de verificar si existe la descripción de un área determinada.
get_existe_id_area(area: var)
<b>Descripción:</b> Responsable de verificar si existe el id de un área determinada pasándole como parámetro un objeto de tipo área.
get_existe_id(id: var)
<b>Descripción:</b> Responsable de verificar si existe el id de un área determinada pasándole como parámetro el id de un área.
get_all_areas()
<b>Descripción:</b> Responsable de devolver todas las áreas existentes.
get_area(descripción: var)
<b>Descripción:</b> Responsable de devolver las áreas dado una descripción.

**Tabla 3.8** Clases Controladoras. dao\_bloque\_acceso

<b>Nombre: dao_bloque_acceso</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
-----	-----
<b>Responsabilidades</b>	
Inserta_Bloque_Acceso(bloque_acceso:var)	
<b>Descripción:</b> Responsable de insertar un bloque de acceso, dado los parámetros especificados.	
Actualiza_Bloque_Acceso(bloque_acceso:var)	
<b>Descripción:</b> Responsable de actualizar un bloque de acceso, dado los parámetros especificados.	
Elimina_Bloque_Acceso(id:var)	
<b>Descripción:</b> Responsable de eliminar un bloque de acceso dado el id pasado como parámetro.	
get_existe_ip(bloque_acceso:var)	
<b>Descripción:</b> Responsable de verificar si existe una dirección Ip dado un bloque de acceso pasado como parámetro.	
get_existe_descripcion(bloque_acceso:var)	

<b>Descripción:</b> Responsable de verificar si existe la descripción de un bloque de acceso dado.
get_existe_id_obj(bloque_acceso:var)
<b>Descripción:</b> Responsable de verificar si existe el id del bloque pasando como parámetro un objeto de tipo bloque de acceso.
get_existe_id(id:var)
<b>Descripción:</b> Responsable de verificar si existe el id del bloque de acceso pasando como parámetro el id del mismo.
get_Descripcion(id: var)
<b>Descripción:</b> Responsable de devolver la descripción de un bloque de acceso pasando como parámetro el id del mismo.
get_all_datos(descrip : var)
<b>Descripción:</b> Responsable de devolver todos los bloques de acceso pasando como parámetro la descripción del mismo.
get_ip_principal()
<b>Descripción:</b> Responsable de devolver la dirección IP del bloque de Acceso Principal.

**Tabla 3.9** Clases Controladoras. dao\_persona

<b>Nombre: dao_persona</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
---	---
<b>Responsabilidades</b>	
Inserta_Persona(persona:var)	
<b>Descripción:</b> Responsable de insertar una persona dado el objeto persona entrado como parámetro.	
Actualiza_Persona(persona: var)	
<b>Descripción:</b> Responsable de actualizar una persona, dado el objeto de ese tipo entrado como parámetro.	
Cambia_Estado(id:var)	
<b>Descripción:</b> Responsable de cambiar el estado de una persona dado el id de la misma entrado como	

parámetro.
Elimina_Persona(solapin: var, ci: var)
<b>Descripción:</b> Responsable de eliminar una persona dado su número de solapín y su carnet de identidad.
Existe_Solapin(solapin: var)
<b>Descripción:</b> Responsable de verificar si existe el solapín entrado como parámetro.
Existe_ID(id: var)
<b>Descripción:</b> Responsable de verificar si existe el id de la persona entrado como parámetro.
Existe_Solapin_per(persona: var)
<b>Descripción:</b> Responsable de verificar si existe el solapín de la persona entrado como parámetro.
get_id(solapin: var ,ci: var)
<b>Descripción:</b> Responsable de devolver el id de la persona dado un número de solapín y un carnet de identidad entrados como parámetros.
Get_Solapin_CI(id: var)
<b>Descripción:</b> Responsable de devolver el número de solapín y el carnet de identidad de la persona dado el id entrado como parámetro.
Get_Solapin(persona: var)
<b>Descripción:</b> Responsable de devolver el número de solapín de la persona dado un objeto de ese mismo tipo entrado como parámetro.
Get_Id_Estado(id: var)
<b>Descripción:</b> Responsable de devolver el id del estado de la persona dado un id entrado como parámetro.
Get_Categoria(solapin: var, ci: var)
<b>Descripción:</b> Responsable de devolver la categoría de una persona dado el número de solapín y el carnet de identidad entrados como parámetros.
Get_Id_Categoria_ID(id: var)
<b>Descripción:</b> Responsable de devolver el id de la categoría de una persona dado su id entrado como parámetro.

**Tabla 3.10** Clases Controladoras. dao\_visitante

<b>Nombre: dao_visitante</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
---	---
<b>Responsabilidades</b>	
Inserta_Visitante(visitante: var)	
<b>Descripción:</b> Responsable de insertar un visitante dado el objeto del mismo tipo entrado como parámetro.	
Actualiza_Visitante(visitante: var)	
<b>Descripción:</b> Responsable de actualizar un visitante dado el objeto del mismo tipo entrado como parámetro	
Elimina_Visitante(solapin: var, ci: var)	
<b>Descripción:</b> Responsable de eliminar un visitante dado el solapín y el carnet de identidad entrado como parámetro.	
Busca_Datos_Visitante(id: var, ci: var, solapin: var, nombre: var)	
<b>Descripción:</b> Responsable de devolver todos los datos de un visitante dada cualquier combinación de los parámetros entrados como datos.	
Existe_Nombre(nombre: var)	
<b>Descripción:</b> Responsable de verificar que exista el nombre entrado como parámetro.	
Existe_Solapin(solapin: var)	
<b>Descripción:</b> Responsable de verificar que exista el solapín entrado como parámetro.	
Existe_Id(visitante: var)	
<b>Descripción:</b> Responsable de verificar que exista el id del visitante dado un objeto del mismo tipo entrado como parámetro.	
Existe_Id_visitante(solapin: var, ci: var)	
<b>Descripción:</b> Responsable de verificar que exista el id del visitante dado un número de solapín y un carnet de identidad entrados como parámetros.	
Existe_visitante_Id(id: var)	

<b>Descripción:</b> Responsable de verificar que exista el id del visitante dado un id entrado como parámetro.
Exista_CI(ci: var)
<b>Descripción:</b> Responsable de verificar que exista el carnet de identidad del visitante dado un carnet de identidad entrado como parámetro.
get_id(solapin: var, ci: var)
<b>Descripción:</b> Responsable de devolver el id del visitante dado un número de solapín y un carnet de identidad entrados como parámetros.

**Tabla 3.11** Clases Controladoras. dao\_registro

<b>Nombre: dao_registro</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
---	---
<b>Responsabilidades</b>	
Inserta_Registro(registro: var)	
<b>Descripción:</b> Responsable de insertar un registro dado un objeto del mismo tipo entrado como parámetro.	
Actualiza_Registro(registro: var)	
<b>Descripción:</b> Responsable de actualizar un registro dado un objeto del mismo tipo entrado como parámetro.	
Elimina Registró(id_registro: var)	
<b>Descripción:</b> Responsable de eliminar un registro dado un id entrado como parámetro.	
get_fecha_entrada(registro: var)	
<b>Descripción:</b> Responsable de devolver la fecha de entrada a la institución pasando como parámetro un objeto de tipo registro.	
get_fecha_salida(registro: var)	
<b>Descripción:</b> Responsable de devolver la fecha de salida de la institución pasando como parámetro un objeto de tipo registro.	
get_existe_persona(id: var)	
<b>Descripción:</b> Responsable de verificar si existe la persona cuyo id es pasado como parámetro.	

**Tabla 3.12** Clases Controladoras. dao\_acceso

<b>Nombre: dao_acceso</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
---	---
<b>Responsabilidades</b>	
Obten_Ip_Bloque_Acceso(id: var)	
<b>Descripción:</b> Responsable de devolver las direcciones Ip de los bloques a los cuales tiene acceso una persona pasando como parámetro su Id.	
Verificar_Acceso(id: var, ip_bloque: var)	
<b>Descripción:</b> Responsable de verificar si una persona tiene acceso al área solicitada pasando como parámetros su id y la dirección Ip del bloque desde el cual se está verificando.	

**Tabla 3.13** Clases Controladoras. dao\_select

<b>Nombre: dao_select</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
---	---
<b>Responsabilidades</b>	
get_categoria_invitado()	
<b>Descripción:</b> Responsable de devolver la categoría de un visitante.	
get_categoria_trabajadores()	
<b>Descripción:</b> Responsable de devolver la categoría de las personas registradas en la institución (trabajadores, profesores y directivos).	

**Tabla 3.14** Clases Controladoras. dao\_estadisticas

<b>Nombre: dao_estadisticas</b>	
<b>Tipo de clase: Controladora</b>	

Atributo	Tipo
--	----
<b>Responsabilidades</b>	
get_cant_directivos()	
<b>Descripción:</b> Responsable de devolver la cantidad de directivos que hay dentro de la institución un momento dado.	
get_cant_profesores()	
<b>Descripción:</b> Responsable de devolver la cantidad de profesores que hay dentro de la institución un momento dado.	
get_cant_trabajadores()	
<b>Descripción:</b> Responsable de devolver la cantidad de trabajadores que hay dentro de la institución un momento dado.	
get_cant_visitantes()	
<b>Descripción:</b> Responsable de devolver la cantidad de visitantes que hay dentro de la institución un momento dado.	
get_datos_fecha_categoria(fecha: var,categoría: var)	
<b>Descripción:</b> Responsable de devolver todos los datos del registro de una persona pasando como parámetros cualquier combinación de la fecha y la categoría de la misma.	

**Tabla 3.15** Clases Controladoras. dao\_busca\_datos\_visitante

<b>Nombre: dao_busca_datos_visitante</b>	
<b>Tipo de clase: Controladora</b>	
Atributo	Tipo
---	---
<b>Responsabilidades</b>	
busca_datos_visitante(ci: var,solapin: var,nombre: var)	
<b>Descripción:</b> Responsable de devolver todos los datos de un visitante dada cualquier combinación de los datos pasados como parámetros.	
existeNombre(nombre:var)	

**Descripción:** Responsable de verificar si existe el nombre del visitante pasado como parámetro.

**Tabla 3.16** Clases Controladoras. CC\_area

<b>Nombre: CC_area</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
----	-----
<b>Responsabilidades</b>	
Inserta_Area(area:var)	
<b>Descripción:</b> Responsable de insertar un área si los datos enviados son correctos.	
Actualiza_Area(area:var)	
<b>Descripción:</b> Responsable de actualizar un área si los datos enviados son correctos.	
Elimina_areas(id:var)	
<b>Descripción:</b> Responsable de eliminar un área si los datos enviados son correctos.	
Get_All_Areas()	
<b>Descripción:</b> Responsable de devolver todos los datos correspondientes al área.	
Get_Area(descrip:var)	
<b>Descripción:</b> Responsable de devolver todos los datos correspondientes a un área dada su descripción si los datos enviados con correctos.	
Reglas_Negocio_Area(area:var)	
<b>Descripción:</b> Responsable de validar que los datos para insertar un área sean correctos.	
Reglas_Negocio_Actualiza(area:var)	
<b>Descripción:</b> Responsable de validar que los datos para actualizar un área sean correctos.	
Reglas_Elimina(id:var)	
<b>Descripción:</b> Responsable de validar que los datos para actualizar un área sean correctos.	
Reglas_Descripcion(descrip:var)	
<b>Descripción:</b> Responsable de validar que los datos para buscar un área.	

**Tabla 3.17** Clases Controladoras. CC\_bloque\_acceso

<b>Nombre: CC_bloque_acceso</b>
---------------------------------



<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
----	-----
<b>Responsabilidades</b>	
Inserta_Bloque_Acceso(bloque_acceso:var)	
<b>Descripción:</b> Responsable de insertar un bloque de acceso si los datos enviados son correctos.	
Actualiza_Bloque_Acceso(bloque_acceso:var)	
<b>Descripción:</b> Responsable de actualizar un bloque de acceso si los datos enviados son correctos.	
Elimina_Bloque_Acceso(id:var)	
<b>Descripción:</b> Responsable de eliminar un bloque de acceso si los datos enviados son correctos.	
Buscar_Bloque_Acceso(descripcion:var)	
<b>Descripción:</b> Responsable de devolver todos los datos correspondientes al bloque de acceso si los datos enviados con correctos.	
Get_Descripcion (id:var)	
<b>Descripción:</b> Responsable de devolver todos los datos correspondientes a un bloque de acceso dada su descripción si los datos enviados con correctos.	
Reglas_Negocio_Inserta_Bloque_Acceso(bloque_acceso:var)	
<b>Descripción:</b> Responsable de validar que los datos para insertar un bloque de acceso sean correctos.	
Reglas_Negocio_Actualiza_Bloque(bloque_acceso:var)	
<b>Descripción:</b> Responsable de validar que los datos para actualizar un bloque de acceso sean correctos.	
Reglas_Negocio_Elimina_Bloque(id:var)	
<b>Descripción:</b> Responsable de validar que los datos para eliminar un bloque de acceso sean correctos.	
Reglas_Negocio_Descripcion(descrip:var)	
<b>Descripción:</b> Responsable de validar que los datos para buscar un bloque de acceso dada su descripción sean correctos.	
Reglas_Negocio_Descripcion_Id(id:var)	
<b>Descripción:</b> Responsable de validar que el identificador del bloque de acceso sea correcto.	

**Tabla 3.18** Clases Controladoras. CC\_persona

<b>Nombre: CC_persona</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
----	-----
<b>Responsabilidades</b>	
Insertar_Persona(persona:var)	
<b>Descripción:</b> Responsable de insertar una persona si los datos enviados son correctos.	
Actualiza_Persona(persona:var)	
<b>Descripción:</b> Responsable de actualizar una persona si los datos enviados son correctos.	
Elimina_Persona(solapin:var,ci:var)	
<b>Descripción:</b> Responsable de eliminar una persona si los datos enviados son correctos.	
Get_Id_Persona(solapin:var,ci:var)	
<b>Descripción:</b> Responsable de devolver el identificador de una persona si los datos enviados son correctos.	
Get_Categoria(solapin:var,ci:var)	
<b>Descripción:</b> Responsable de devolver la categoría de una persona si los datos enviados son correctos.	
Reglas_Negocio_Persona(persona:var)	
<b>Descripción:</b> Responsable de validar que los datos para insertar una persona sean correctos.	
Reglas_Negocio_Actualiza_Persona(persona:var)	
<b>Descripción:</b> Responsable de validar que los datos para actualizar una persona sean correctos.	
Reglas_Negocio_Elimina_Persona(solapin:var,ci:var)	
<b>Descripción:</b> Responsable de validar que los datos para eliminar una persona sean correctos.	
Reglas_Negocio_Id_Persona(solapin:var, ci:var)	
<b>Descripción:</b> Responsable de validar que los datos para devolver el identificador de una persona sean correctos.	
Reglas_Negocio_Get_Categoria(solapin:var,ci:var)	
<b>Descripción:</b> Responsable de validar que los datos para devolver la categoría de una persona sean correctos.	

**Tabla 3.19** Clases Controladoras. CC\_visitante

<b>Nombre: CC_visitante</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
----	-----
<b>Responsabilidades</b>	
Inserta_Visitante(visitante:var)	
<b>Descripción:</b> Responsable de insertar un visitante si los datos enviados son correctos.	
Actualiza_Visitante(visitante:var)	
<b>Descripción:</b> Responsable de actualizar un visitante si los datos enviados son correctos.	
Elimina_visitante(solapin:var,ci:var)	
<b>Descripción:</b> Responsable de eliminar un visitante si los datos enviados son correctos.	
Busca_Vistante(id:var,ci:var,solapin:var,nombre:var)	
<b>Descripción:</b> Responsable de buscar un visitante si los datos enviados son correctos.	
Reglas_Negocio_Visitante(visitante:var)	
<b>Descripción:</b> Responsable de validar que los datos para insertar un visitante sean correctos.	
Reglas_Negocio_Actualiza_Visitante(visitante:var)	
<b>Descripción:</b> Responsable de validar que los datos para actualizar un visitante sean correctos.	
Reglas_Negocio_Elimina_Visitante(solapin:var,ci:var)	
<b>Descripción:</b> Responsable de validar que los datos para eliminar un visitante sean correctos.	
Reglas_Negocio_Buscar(id:var,ci:var,solapin:var,nombre:var)	
<b>Descripción:</b> Responsable de validar que los datos para buscar un visitante sean correctos.	

**Tabla 3.20** Clases Controladoras. CC\_registro

<b>Nombre: CC_registro</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
----	-----

<b>Responsabilidades:</b>
Inserta_Registro(registro:var)
<b>Descripción:</b> Responsable de insertar un registro si los datos enviados son correctos.
Actualiza_Registro(registro:var)
<b>Descripción:</b> Responsable de actualizar un registro si los datos enviados son correctos.
Get_Fecha_Entrada(registro:var)
<b>Descripción:</b> Responsable de devolver la fecha de entrada al sistema si los datos enviados son correctos.
Get_Fecha_Salida(registro:var)
<b>Descripción:</b> Responsable de devolver la fecha de salida del sistema si los datos enviados son correctos.
Reglas_Negocio_Registro(registro:var)
<b>Descripción:</b> Responsable de validar que los datos del registro sean correctos.

**Tabla 3.21** Clases Controladoras. CC\_estadisticas

<b>Nombre: CC_estadisticas</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
----	-----
<b>Responsabilidades:</b>	
get_activos()	
<b>Descripción:</b> Responsable de devolver los las cantidades de usuarios que están activos en el sistema.	
Buscar_Accesos(fecha:var, categoria:var)	
<b>Descripción:</b> Responsable de devolver todos los accesos de las personas si los datos enviados son correctos.	
Reglas_Negocio_Estadistica(fecha:var, categoria:var)	
<b>Descripción:</b> Responsable de validar que los datos para buscar las estadísticas sean correctos.	

**Tabla 3.22** Clases Controladoras. CC\_busca\_datos

<b>Nombre: CC_busca_datos</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
----	-----
<b>Responsabilidades:</b>	
Busca_Datos(solapin:var,ci:var)	
<b>Descripción:</b> Esta función determina dados los datos enviados, la categoría de la persona en cuestión, si es visitante busca los datos en la base de datos Control de acceso, sino invoca la función <b>busqueda_simple</b> del servicio web Gestion_Usuario devolviendo los datos de la misma.	
Reglas_Negocio_Busca_Datos(solapin:var, ci:var)	
<b>Descripción:</b> Responsable de validar que los datos para buscar una persona sean correctos.	

**Tabla 3.23** Clases Controladoras. CC\_acceso

<b>Nombre: CC_acceso</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
----	-----
<b>Responsabilidades:</b>	
Registra_Acceso(registro:var)	
<b>Descripción:</b> Responsable de registrar los accesos de las personas si los datos enviados son correctos.	

Las clases de tipo controladoras descritas anteriormente no tienen atributos solo responsabilidades, porque son las encargadas del envío de mensajes y no de almacenar datos como tal. La clase interfaz descrita es la correspondiente a la interfaz del servicio web implementado, donde están expuestos todos los servicios que brinda. Existen clases como: CE\_area\_acceso, CE\_bloque\_area, dao\_area\_acceso, dao\_bloque\_area, que no han sido descritas, aunque se utilizan en la implementación del mismo pues no son invocadas directamente sino a través de otras funciones.

### 3.4 Diseño de la Base de Datos

Una Base de Datos es un conjunto de información de forma correlativa y organizada de la cual posteriormente se puede extraer uno o varios datos necesarios. Uno de los modelos más usados hoy en día es el relacional, con el que se puede, como bien dice su nombre, relacionar unos datos con otros para facilitar el trabajo al programador y también a la hora de realizar una búsqueda. Para la construcción de la base de datos se tuvo en cuenta que fuese de este tipo y se empleó el modelo de datos teniendo en cuenta que constituye la representación física de la misma.

#### 3.4.1 Modelo de Datos

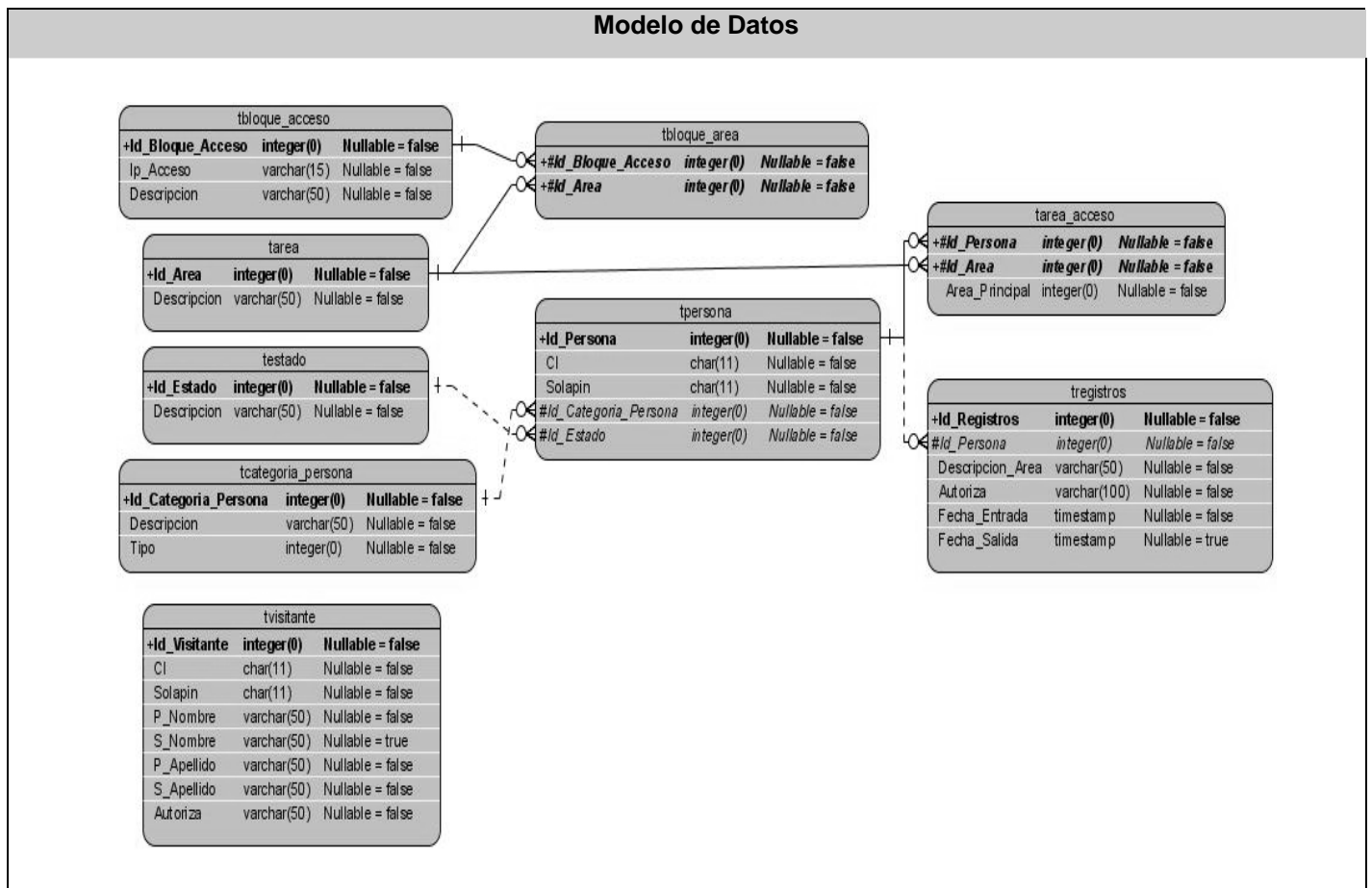


Figura 3.2 Modelo de Datos

### 3.4.2 Descripción de las tablas de la Base de Datos

**Tabla 3.24** Tabla tpersona

<b>Nombre: tpersona</b>		
<b>Descripción:</b> Tabla que almacena los datos de las personas que se registran y acceden a la institución.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Persona	integer (0)	Identificador de la tabla
Cl	char (11)	Carnet de identidad de las personas que son registradas.
Solapín	char (11)	Un identificador para cada persona, en caso de los visitantes es uno temporal y para el resto este numero coincide con el carnet de identidad,
Id_Categoria_Persona	integer (0)	Identificador de la tabla tcategoria_persona (donde se almacenan todas las categorías de las personas registradas).
Id_Estado	integer (0)	Identificador de la tabla estado (donde se almacena el estado de las personas registradas).

**Tabla 3.25** Tabla testado

<b>Nombre: testado</b>		
<b>Descripción:</b> Tabla que almacena los estados de las personas registradas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Estado	integer (0)	Identificador de la tabla.
Descripcion	varchar (50)	Especifica el estado de la persona (activo o inactivo).

**Tabla 3.26** Tabla tcategoria\_persona

<b>Nombre: tcategoria_persona</b>		
<b>Descripción:</b> Tabla que almacena las categorías de las personas registradas con un identificador que especifica a que grupo pertenecen si a visitantes o personas permanentes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Categoria_Persona	integer (0)	Identificador de la tabla.

Descripcion	varchar (50)	Especifica la categoría de la persona (directivo, visitante, trabajador, profesor).
Tipo	integer (0)	Identificador del subgrupo al que pertenece cada categoría (visitantes y personas permanentes).

**Tabla 3.27** Tabla tarea

<b>Nombre: tarea</b>		
<b>Descripción:</b> Tabla que almacena la descripción de las áreas existentes en la institución.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Area	integer (0)	Identificador de la tabla.
Descripcion	varchar (50)	Nombre del área que se registra.

**Tabla 3.28** Tabla tbloque\_acceso

<b>Nombre: tbloque_acceso</b>		
<b>Descripción:</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Bloque_Acceso	integer (0)	Identificador de la tabla.
Ip_Acceso	varchar (15)	Dirección IP de la PC desde la cual se controlará el acceso a las áreas.
Descripcion	varchar (50)	Nombre del bloque de acceso registrado.

**Tabla 3.29** Tabla tbloque\_area

<b>Nombre: tbloque_area</b>		
<b>Descripción:</b> Tabla que almacena el identificador de las áreas que controla un bloque de acceso determinado y el identificador de este.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Bloque_Acceso	integer (0)	Identificador de la tabla tbloque_acceso (donde se almacenan los datos de los bloques de accesos registrados).
Id_Area	integer (0)	Identificador de la tabla tarea (donde se almacenan los



		datos de las áreas registradas).
--	--	----------------------------------

**Tabla 3.30** Tabla tarea\_acceso

<b>Nombre: tarea_acceso</b>		
<b>Descripción:</b> Tabla que almacena el identificador de las personas, las áreas a las que tienen acceso, almacenando su identificador y el Area_Principal.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Persona	integer (0)	Identificador de la tabla
Id_Area	integer (0)	Identificador de la tabla tarea (donde se almacenan los datos de las áreas registradas).
Area_Principal	integer (0)	

**Tabla 3.31** Tabla registros

<b>Nombre: registros</b>		
<b>Descripción:</b> Tabla principal que registra los datos de la personas que acceden. Almacena el identificador de la persona (Id_Persona), el nombre del área principal, quien autoriza y las fechas de entradas y salidas del sistema.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Registros	integer(0)	Identificador de la tabla.
Id_Persona	integer (0)	Identificador de la tabla tpersona (donde se almacenan los datos de las personas registradas que acceden a la institución).
Descripcion_Area	varchar (50)	Nombre del área principal que la persona tiene asignada.
Autoriza	varchar (100)	Usuario que esta logueado en el sistema donde se controla el acceso.
Fecha_Entrada	timestam p(0)	Fecha de entrada al sistema, tomado por defecto incluye la hora.
Fecha_Salida	timestam p(0)	Fecha de salida al sistema, tomado por defecto incluye la hora.

**Tabla 3.32** Tabla tvisitante

<b>Nombre: tvisitante</b>		
<b>Descripción:</b> Tabla que almacena los datos de los visitantes que acceden a la institución.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_Visitante	integer (0)	Identificador de la tabla.
CI	varchar(11)	Carnet de identidad de los visitantes que son registradas.
Solapin	varchar(11)	Un identificador eventual para cada visitante.
P_Nombre	varchar(50)	Primer nombre del visitante.
S_Nombre	varchar(50)	Segundo nombre del visitante.
P_Apellido	varchar(50)	Primer apellido del visitante.
S_Apellido	varchar(50)	Segundo apellido del visitante.
Autoriza	varchar(50)	Usuario que esta logueado en el sistema donde se controla el acceso.

## 4 IMPLEMENTACIÓN DEL SISTEMA

En el presente capítulo se explica cómo está estructurado físicamente el sistema implementado mediante los modelos de despliegue y de componentes, con los diagramas correspondientes estructurados por paquetes con sus respectivas descripciones.

### 4.1 Modelo de despliegue

El modelo de despliegue contiene los nodos que conforman la topología de hardware sobre la cual se ejecuta el sistema software implementado (servicio web).

En la figura 4.1 se muestra el Diagrama de despliegue del sistema desarrollado. El mismo cuenta con dos nodos: uno de ellos es un servidor “Servicio Web “ donde se va a encontrar toda la lógica del negocio, o sea el servicio web como tal y finalmente la base de datos Control\_Acceso

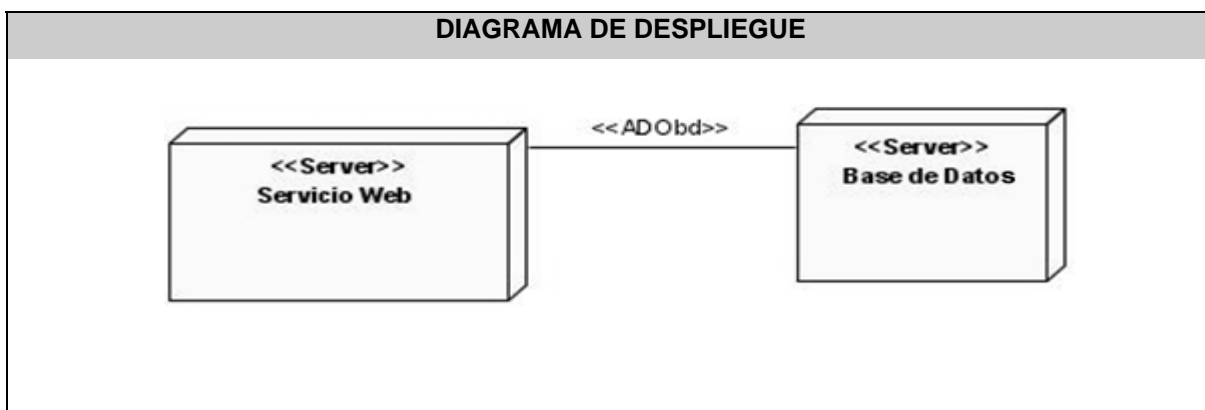


Figura 4.1 Diagrama de Despliegue

## 4.2 Modelo Implementación

El modelo de implementación describe cómo se implementan los elementos del modelo de diseño en términos de componentes. Describe también como se organizan y se relacionan unos con otros, definiendo un componente como el empaquetamiento físico de los elementos de un modelo, como es el caso de las clases del modelo de diseño.

### 4.2.1 Diagrama de Componentes en Paquetes

Los Diagramas de Componentes representan la separación del sistema software en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes, etc.). Se usan para modelar la estructura del software, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables.

A continuación se representa el diagrama de componentes estructurado por paquetes correspondientes al sistema implementado.

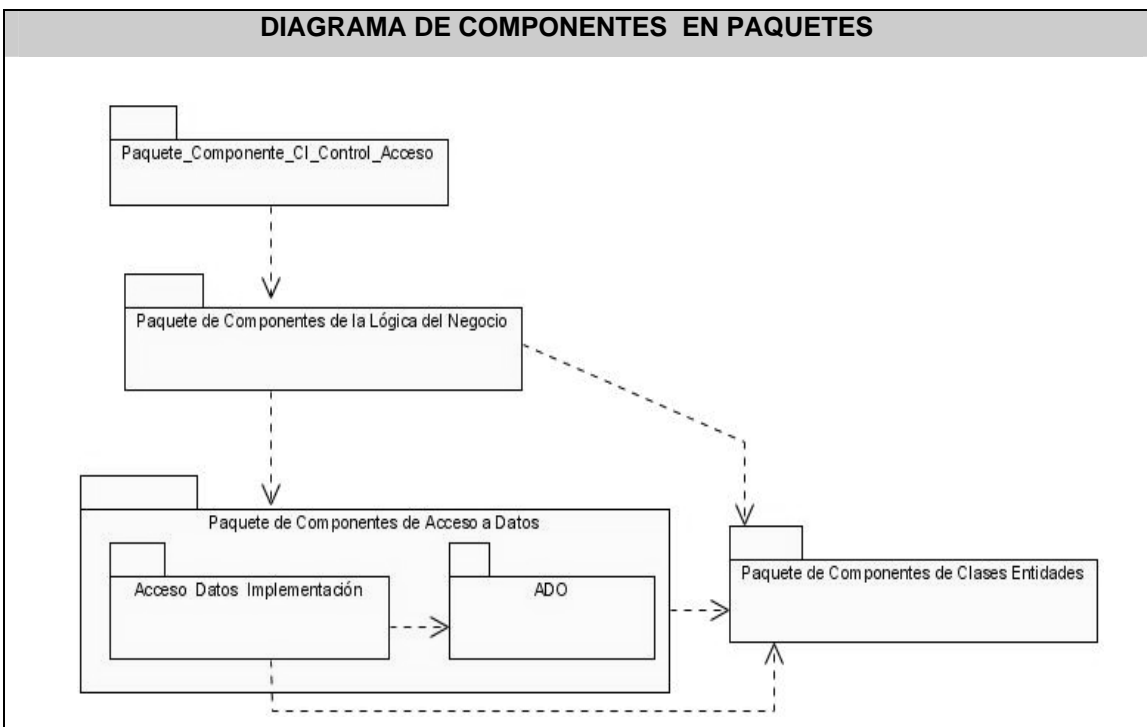


Figura 4.2 Diagrama de componentes en paquetes

#### 4.2.2 Diagramas de Componentes separados por paquetes

Los siguientes diagramas son los correspondientes a cada uno de los paquetes expuestos anteriormente. Con el objetivo de lograr una mayor limpieza y rapidez al encontrar las clases, se implementó cada una en un fichero separado, con el mismo nombre, por tanto, cada componente representado en estos diagramas tiene el mismo nombre de la clase que contiene.

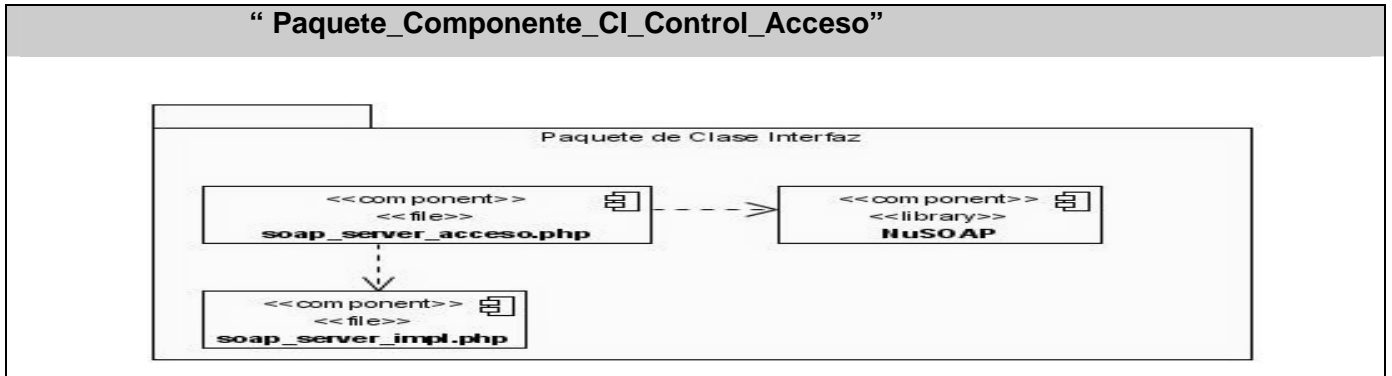


Figura 4.3 Paquete\_Componente\_CI\_Control\_Acceso

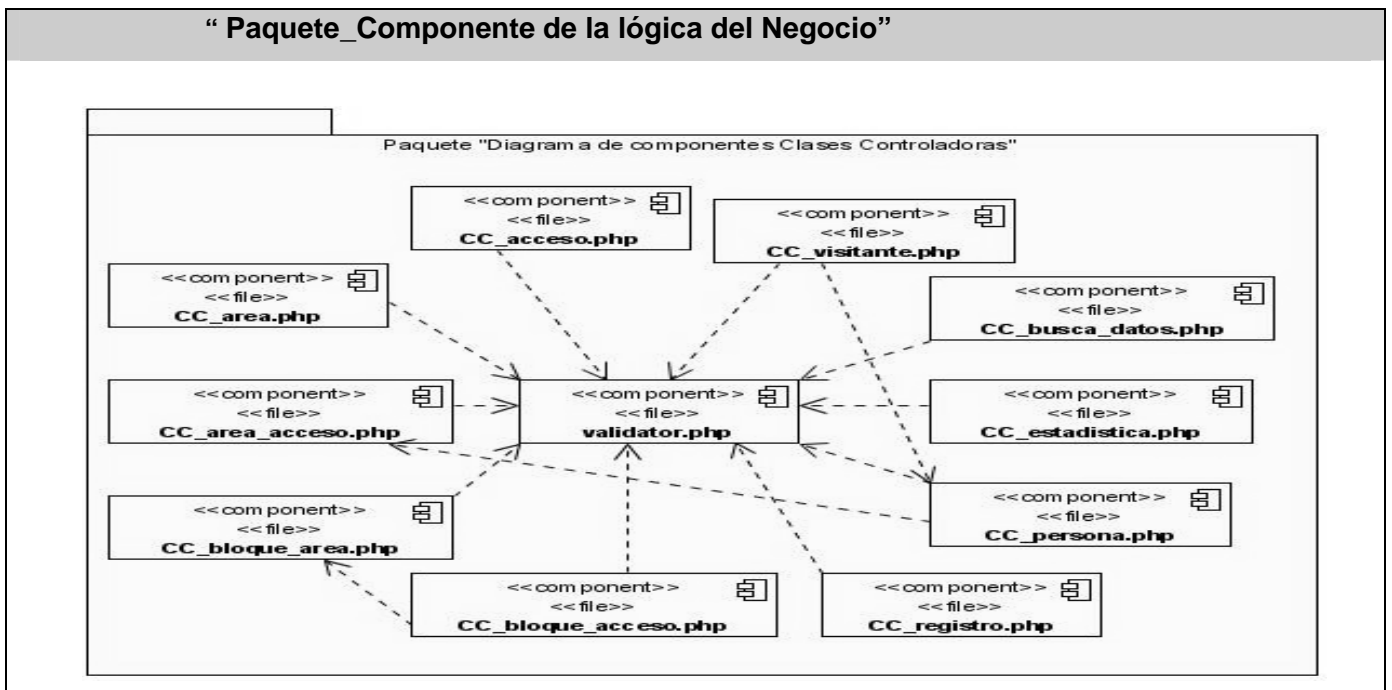


Figura 4.4 Paquete\_Componente de la lógica del Negocio

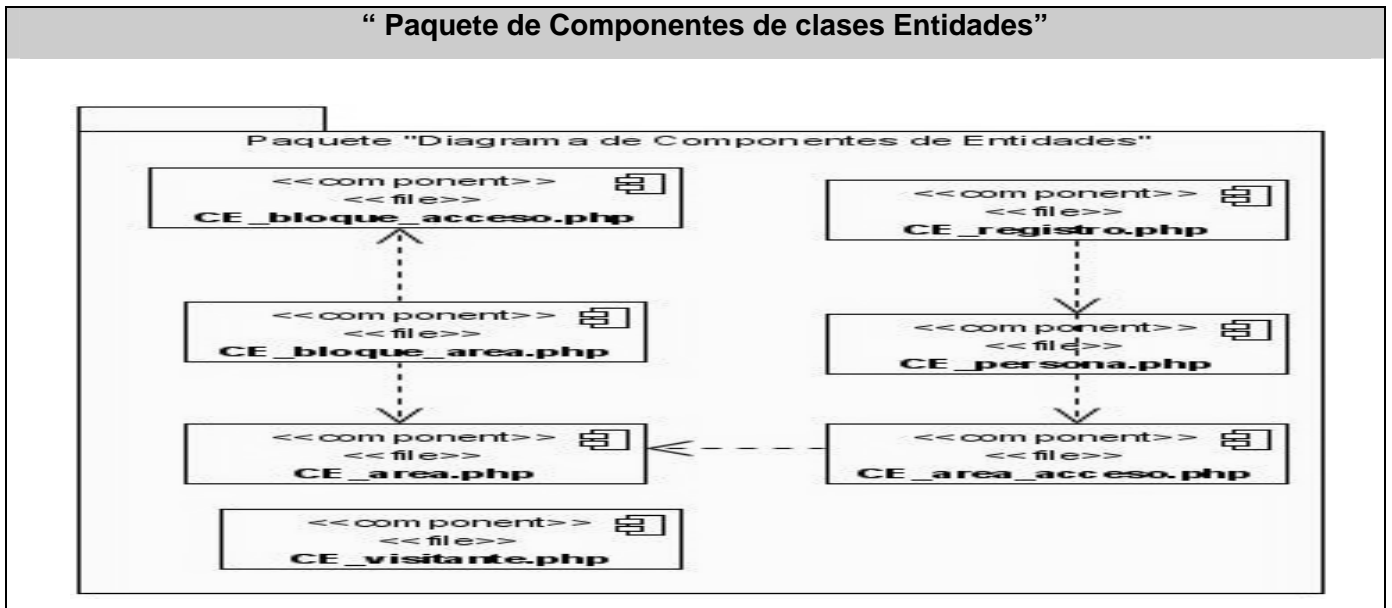


Figura 4.5 Paquete de Componentes de clases Entidades

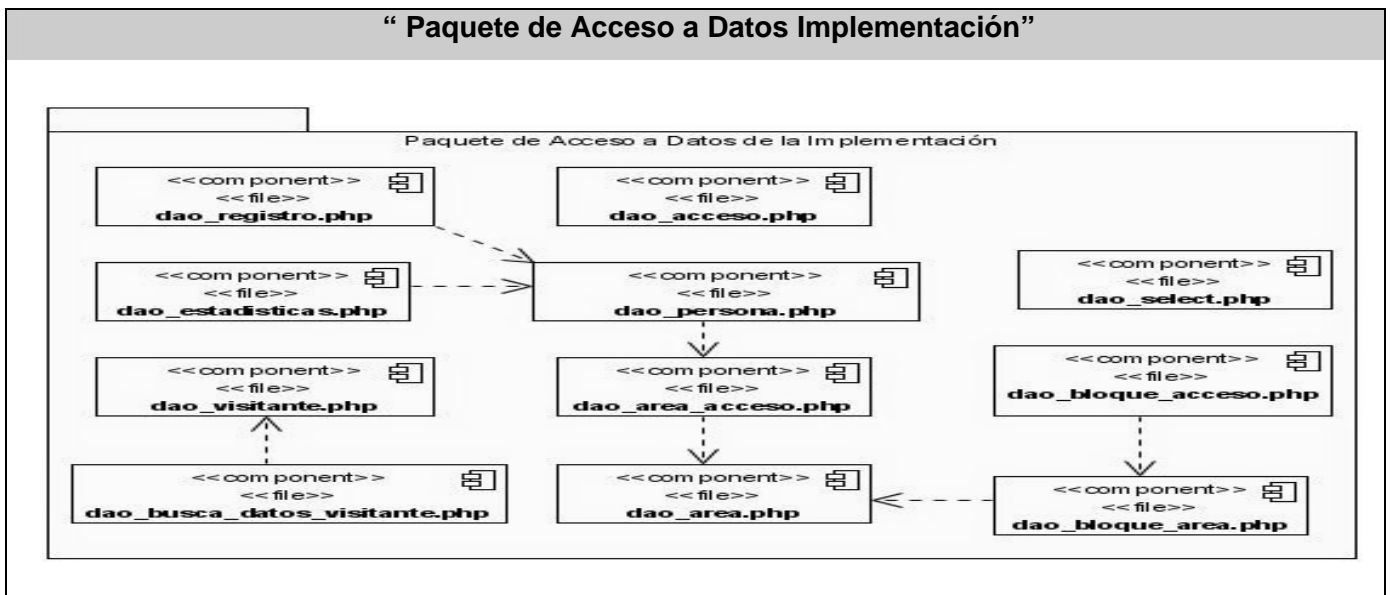


Figura 4.6 Paquete de Acceso a Datos Implementación

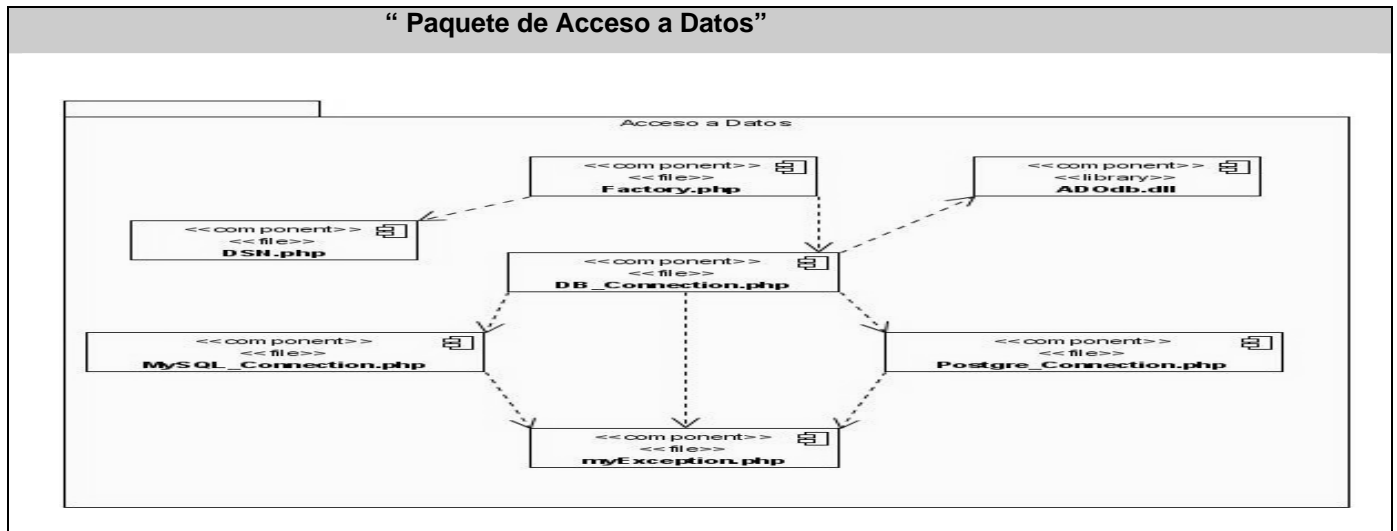


Figura 4.7 Paquete de Acceso a Datos

### 4.2.3 Descripción de cada uno de los Paquetes

- **Paquete\_Componente\_CI\_Control\_Acceso:** En este paquete se encuentran los componentes que contienen las implementaciones de las clases interfaz del Servicio Web, donde se exponen todas las funcionalidades que brinda el mismo, así como la librería NuSOAP que posibilitó su creación.
- **Paquete de Componentes de clases Entidades:** En este paquete se encuentran los componentes que contienen las implementaciones de las clases que representan las entidades del sistema.
- **Paquete\_Componente de la lógica del Negocio:** En este paquete se encuentran los componentes que contienen las implementaciones de clases controladoras de entidades.
- **Paquete de Acceso a Datos Implementación:** En este paquete se encuentran los componentes que contienen las implementaciones de clases que se ocupan de la persistencia en base de datos de las entidades.
- **Paquete de Acceso a Datos:** En este paquete se encuentran los componentes que contienen las implementaciones de clases y librerías que permiten conectarnos a diferentes bases de datos.

## CONCLUSIONES

1. Con el progreso de la presente investigación se demostró la necesidad de desarrollar e implementar en el Centro Rector de Universidad par Todos y Canales Educativos un servicio web para gestionar todo el proceso de control de acceso a dicho centro.
2. Los sistemas de control de accesos estudiados no dan soporte ni cumplen con los requisitos necesarios para desarrollar la aplicación.
3. Se implementó un servicio web que brinda funcionalidades que permiten llevar a cabo el control de accesos de personas al Centro Rector de Universidad para Todos y Canales Educativos y áreas dentro del mismo, así como gestionar información referente a todo este proceso.



## ***RECOMENDACIONES***

1. Se recomienda que este trabajo sea tomado como material de consulta por profesionales que se enfrenten al desarrollo de servicios web.
2. Extender el servicio web adicionándole nuevas funcionalidades de manera que pueda ser utilizado por aplicaciones no sólo en el Centro Rector de Universidad para Todos y Canales Educativos, sino en cualquier empresa que requiera de un control de accesos.

## REFERENCIAS BIBLIOGRÁFICAS

1. *Uso de Tarjetas Inteligentes para un Control de Acceso Físico Seguro. Publicado el: enero 2007 de última actualización: enero 2007. Disponible en: [www.smartcardalliance.org](http://www.smartcardalliance.org). ISBN ID-03003.*
  2. *UCI. Carpeta de documentación del proyecto Informatización del Centro Rector de Universidad para Todos.*
  3. *Acceso. [Consultado el: enero 2007]. Disponible en: [http://www.atrea.com/ATREA/public/UCAGlosario.jsp?marca\\_blanca=terra&letra=a](http://www.atrea.com/ATREA/public/UCAGlosario.jsp?marca_blanca=terra&letra=a).*
  4. *ANEXO I: GLOSARIO DE TÉRMINOS publicado el: enero 2007 de última actualización: enero 2007. Disponible en: [www.garadat.com/auditorias/anexos\\_lopd/anexo1.html](http://www.garadat.com/auditorias/anexos_lopd/anexo1.html).*
  5. *ESPIÑEIRA, S. Mecanismos de control complementarios de un Sistema de Control de Acceso. publicado el: enero 2007 de última actualización: enero 2007. Disponible en: <http://www.pc-news.com/detalle.asp?ida=1379&sid=&id=11>.*
  6. *Informe de la Smart Card Alliance Latin America. publicado el: enero 2007 de última actualización: enero 2007. Disponible en: [www.smartcardalliance.org](http://www.smartcardalliance.org).*
  7. *¿Qué es el control de acceso? publicado el: enero 2007 de última actualización: enero 2007. Disponible en: <http://www.adteurope.com/company/product/accesscontrol.shtml?language=es>.*
  8. *WOLTER, R. Aspectos básicos de XML Web Services. publicado el: febrero 2007 de última actualización: febrero 2007. Disponible en: <http://www.microsoft.com/spanish/msdn/articulos/archivo/280202/voices/webservbasics.asp>.*
  9. *Servicios Web con PHP (NuSOAP) [Consultado el: febrero de 2007]. Disponible en: <http://www.nociondigital.com/webmasters/php-tutorial-servicios-web-con-php-nusoap-detalle-168.html>.*
  10. *WSDL para la documentación de Servicios Web. vol. 2007, Disponible en: <http://www.desarrolloweb.com/articulos/1581.php>.*
  11. *Lenguajes de Programación para la Web. publicado el: febrero 2007 de última actualización: febrero 2007. Disponible en: <http://www.tejedoresdelweb.com/307/article-1883.html>.*
  12. *Proceso Unificado de Rational. publicado el: febrero 2007 de última actualización: febrero 2007. Disponible en: [http://es.wikipedia.org/wiki/Rational\\_Unified\\_Process](http://es.wikipedia.org/wiki/Rational_Unified_Process).*
  13. *CABALLERO, I. Práctica de Ingeniería de Software 3 [Consultado el: febrero de 2007]. Disponible en: [http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1\\_VP.pdf](http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf).*
-

## BIBLIOGRAFÍA

1. A GÓMEZ SKARMETA, J. G. R., G. MARTÍNEZ, . *UMGina: Sistema de control de acceso en aulas basado en tarjetas inteligentes de la Universidad de Murcia*. publicado el: enero 2007 de última actualización: enero 2007. Disponible en:  
<http://www.rediris.es/rediris/boletin/50-51/ponencia19.html#1>.
  2. *Proceso Unificado de Rational*. publicado el: febrero 2007 de última actualización: febrero 2007. Disponible en: [http://es.wikipedia.org/wiki/Rational\\_Unified\\_Process](http://es.wikipedia.org/wiki/Rational_Unified_Process).
  3. *Controles de Acceso de Personal / Identificación /Control de Tiempos y Asistencia / Biometría* [Consultado el: febrero de 2007].  
Disponible en: <http://www.equiposysoluciones.com/AreasE1.asp?aid=124>.
  4. *Control de Acceso*. publicado el: febrero2007 de última actualización: febrero2007. Disponible en: [http://www.kimaldi.com/kimaldi/area\\_de\\_conocimiento/control\\_de\\_acceso\\_y\\_presencia/control\\_de\\_acceso](http://www.kimaldi.com/kimaldi/area_de_conocimiento/control_de_acceso_y_presencia/control_de_acceso).
  5. *Metodología de Desarrollo de Software (MDS)* Última actualización: 2005. [Consultado el: febrero de 2007]. Disponible en: <http://www.reynox.com.ar/sap/metodologia.php>.
  6. *Control de Acceso* Disponible en: [http://www.karnet.com.ve/control\\_acceso.html](http://www.karnet.com.ve/control_acceso.html).
  7. ORAMA FERNÁNDEZ, M. D. J. *Sistema para el Control de Visita*. Instituto Superior Politécnico “José Antonio Echeverría”, 2004.
  8. LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Ciudad de la Habana: 2004. Vol. 1,
-

9. RUMBAUGH, J. J., I. Y BOOCH, G. *El proceso unificado de desarrollo de software*. 2000. vol. 1,
  10. DEPORTES, I. *SuperGym* [Consultado el: junio de 2007]. Disponible en: <http://www.supergympro.com.ar/>.
  11. LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Ciudad de la Habana: vol. 2,
  12. PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico*. Ciudad de la Habana: 2005. Vol. 1,
  13. *Practical UML A Hands-On Introduction for Developers*. Última actualización: marzo, 2004. [Consultado el: abril de 2007]. Disponible en: [http://info.borland.com/techpubs/together/together\\_guides/umlonlinecourse/#component-and-deployment-diagrams](http://info.borland.com/techpubs/together/together_guides/umlonlinecourse/#component-and-deployment-diagrams).
  14. MCCARTHY, J. *Dynamics of Software Development*.
  15. *Practical UML. A Hands-On Introduction for Developers*. Última actualización: marzo 2004. [Consultado el: marzo de 2007]. Disponible en: [http://info.borland.com/techpubs/together/together\\_guides/umlonlinecourse/#component-and-deployment-diagrams](http://info.borland.com/techpubs/together/together_guides/umlonlinecourse/#component-and-deployment-diagrams).
  16. GIL MARTÍN, M. A. *Sistema de Control de Acceso*. Universidad de las Ciencias Informáticas, 2005.
  17. ELECTRONIC, J. *Control de Acceso Basado en la PC* [Consultado el: mayo de 2007]. Disponible en: <http://www.jantek.com/Spanish/jaspcbased.htm>.
-

## ANEXOS

### Anexo I Descripción de los Casos de Uso del Negocio

Tabla I.1 Caso Uso “Entrar la institución”

<b>Nombre del Caso de Uso</b>	Entrar a la institución	
<b>Propósito</b>	El propósito de este caso de uso es permitir al actor entrar al centro.	
<b>Actores:</b>	Usuario	
<b>Resumen:</b>	Se inicia cuando un usuario llega hasta la garita de la institución con el objetivo de entrar, si dicho usuario es un trabajador de la institución, el Agente de Seguridad le pide el solapín y si son correctos los datos le permite la entrada, de lo contrario no lo deja acceder, en caso de ser un profesor el agente contacta vía teléfono con directivos del Departamento de Gestión y Servicios de Información (DGSÍ) del cual recibe confirmación, le registra los datos personales y le permite la entrada, en caso de ser un visitante chequea si la visita es prevista o no, en caso de serlo verifica los datos que están registrados permitiéndole entrar, de lo contrario confirma su entrada llamando por teléfono al visitado y registra sus datos, de no ser así se la niega, el caso de uso finaliza cuando el usuario se retira finalizando el caso finalizando de esta forma el caso de uso.	
<b>Curso Normal de los eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del proceso del negocio</b>	
1 El usuario llega hasta la garita de la institución para entrar:	2 El Agente de Seguridad le solicita la identificación:  a) En caso del usuario ser trabajador ir a la sección “Entrar Trabajador”.  b) En caso de ser Profesor ir a la sección “Entrar Profesor”.  c) En caso de ser Visitante ir a la sección “Entrar Visitante”.	
<b>Sección “Entrar Trabajador ”</b>		
<b>Acción del actor</b>	<b>Respuesta del proceso del negocio</b>	

<p>3 El trabajador muestra la credencial (solapín).</p> <p>6 El trabajador entra finalizando el caso de uso</p>	<p>4 El Agente de Seguridad verifica que los datos sean correctos.</p> <p>5 Archiva en el Registro de Entradas y Salidas los datos: nombre y apellidos, hora de entrada y fecha.</p>
<b>Curso Alternativo de los Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso del negocio</b>
	<p>4 Si el Agente de Seguridad verifica que los datos no son correctos no le permite entrar finalizando el caso de uso</p>
<b>Sección “Entrar Profesor”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>3 El profesor muestra la credencial (CI)</p> <p>6 El profesor entra a la institución finalizando el caso de uso.</p>	<p>4 El Agente de Seguridad contacta vía teléfono con directivos del DGSI y le confirman la entrada al profesor.</p> <p>5 Archiva los datos: nombre y apellidos, CI, hora de entrada y lugar al cual se dirige y fecha en el Registro de Entradas y Salidas.</p>
<b>Sección “Entrar Visitante”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>3 El visitante muestra la credencial (CI).</p> <p>5 El visitante entra finalizando el caso de uso.</p>	<p>4 El Agente de Seguridad verifica en el documento Registro de Entradas y Salidas si la visita está prevista o es imprevista.</p> <p>a) Si la visita es prevista verifica los datos registrados: nombre y apellidos, hora de entrada, lugar al cual se dirige, fecha y le confirma la entrada.</p> <p>b) En caso de la visita ser imprevista (Se pasa al paso</p>

	<p>6)</p> <p>6. El Agente de Seguridad contacta vía teléfono con la persona o el área que va a ser visitada (visitado) confirmando que puede ser atendida.</p> <p>7. El Agente de Seguridad registra en el Registro de Entrada y Salidas los datos personales del visitante, nombre y apellidos, hora de entrada, lugar al cual se dirige y fecha. (Se pasa al paso 5)</p>
<b>Curso Alternativo de los Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso del negocio</b>
	<p>6 El Agente de Seguridad contacta vía teléfono con la persona o el área que va a ser visitada (visitado) confirmando que no puede ser atendida finalizando el caso de uso.</p>
<b>Prioridad</b>	Crítico
<b>Mejoras</b>	<p>Con el proceso de entradas de personas a la institución automatizado permitirá eficiencia, eliminar cualquier problema con el atraso en la misma, cualquier intento de de accesos de personas ajenas con solapines falsos garantizando un control más estricto y confiable del proceso.</p>

**Tabla I.2** Caso Uso “Salir de la institución”

<b>Nombre del Caso de Uso</b>	Salir de la institución
<b>Actores</b>	Usuario
<b>Propósito</b>	Permitir al usuario salir de la institución.
<b>Resumen</b>	El caso de uso se inicia cuando el usuario llega hasta la garita de la institución con el objetivo de salir, el Agente de Seguridad archiva en el registro de Entradas y Salidas la fecha de salida finalizando el caso de uso.
<b>Curso Normal de los eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del proceso de negocio</b>
<ol style="list-style-type: none"> <li>1. El usuario llega hasta la garita de la institución con el objetivo de salir.</li> <li>2. Muestra la credencial</li> <li>3. El usuario sale de la institución finalizando el caso de uso.</li> </ol>	<ol style="list-style-type: none"> <li>2. El Agente de Seguridad localiza al usuario en el documento Registro de Entradas y Salidas y archiva el horario de salida.</li> </ol>
<b>Prioridad</b>	Crítico
Mejoras	Con la automatización de este proceso se permitirá registrar automáticamente el horario de salida de las personas, garantizando efectividad y brindando la posibilidad de generar automáticamente información.



## Anexo II Casos de usos expandidos

Tabla II.1 Caso de Uso\_ Registrar Accesos

Caso de uso	
CU_1	Registrar Accesos
<b>Propósito</b>	Permite verificar que las personas que entran a las áreas de la institución y pasan por cada uno de los bloques de acceso tienen acceso y se registra la incidencia en dependencia del bloque por el cual esté accediendo.
<b>Actores</b>	Intranet
<b>Resumen:</b>	Se inicia cuando el actor necesita verificar si la persona tiene acceso al área solicitada, para ello invoca el servicio registrar accesos enviando los datos pertinentes, verifica que la persona tenga acceso al área y se registra el acceso si es el bloque principal, finalizando el caso de uso.
<b>Referencias</b>	RF_1, RF_1.1, RF_1.2
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1 La Intranet invoca el servicio registrar accesos enviando los datos referenciados en RF_1	2 Verifica que dentro de los datos enviados el solapín esté activo y que tenga acceso al área solicitada.  3 Verifica si es el bloque principal.  4 Verifica que no haya entrado al sistema y registra el acceso con los datos pertinentes, finalizando el caso de uso.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

	<p>2 Si verifica que el solapín enviado no tiene acceso al área solicitada o esta inhabilitado muestra un mensaje de error “No tiene acceso al área solicitada o usuario inhabilitado”.</p> <p>3. Si verifica que no es el bloque principal, verifica que tenga acceso al bloque y muestra un mensaje “Tiene acceso al área o no tiene acceso”.</p> <p>4 a) Si verifica que ha entrado al sistema y la fecha de salida es nula, actualiza el registro insertando la fecha de salida, finalizando el caso de uso.</p> <p>4 a) Si verifica que ha entrado al sistema y la fecha de salida no es nula, realiza nuevamente el registro insertando los datos pertinentes, finalizando el caso de uso.</p>
--	--

**Tabla II.2** Caso de Uso\_ Gestionar áreas

Caso de uso	
CU_2	Gestionar áreas
<b>Propósito</b>	El propósito de este caso de uso es permitir al actor registrar, modificar o eliminar un área determinada.
<b>Actores</b>	Intranet
<b>Resumen:</b>	Se inicia cuando el actor desea realizar algunas de las siguientes operaciones: registrar un área nueva de acceso, modificar alguna que ya esté registrada o eliminar, en caso de registrar invoca el servicio insertar área, envía los datos pertinentes, ejecuta la acción y retorna un identificador del área (id área), en caso de modificar o eliminar invoca los servicios actualizar área o eliminar área dado los datos enviados actualizando la base de datos, finalizando el caso de uso.
<b>Referencias</b>	RF_2 RF_3 RF_4
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

<p>1. La Intranet puede realizar algunas de las siguientes acciones: registrar, modificar o eliminar áreas.</p> <p>a) Si decide registrar un área ir a la sección “Registrar área de acceso”.</p> <p>b) Si decide modificar un área ir a la sección “Modificar área de acceso”.</p> <p>c) Si decide eliminar un área ir a la sección “Eliminar área de acceso”.</p>	
<b>Sección “Registrar área de acceso”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1 a) La intranet invoca el servicio Insertar área enviando los datos pertinentes:</p> <p>Descripción área</p>	<p>2 Verifica que los datos para insertar el área hayan sido enviados.</p> <p>3 Verifica que los datos enviados sean correctos.</p> <p>4 Verifica que no existan en la Base de Datos Control de Acceso (BDCA).</p> <p>5 Inserta la nueva área en la BDCA retornando el Id del área y finalizando el caso de uso.</p>
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2 Si verifica que los datos no han sido enviados retorna un mensaje de error.</p>

	<p>3 Si verifica que los datos enviados no son correctos retorna un mensaje de error.</p> <p>4 Si verifica que el área suministrada existe en la Base de Datos Control de Acceso (BDCA) retorna un mensaje de error “Esa área ya existe”.</p>
<b>Sección “Modificar área de acceso”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1 b) La intranet invoca el servicio actualizar área enviando los datos pertinentes para las modificaciones:</p> <p>Descripción área</p>	<p>2 Verifica que los datos para insertar el área hayan sido enviados.</p> <p>3 Verifica que los datos enviados para las modificaciones estén correctos.</p> <p>4 Verifica que los datos enviados para las modificaciones no existan en la BDCA.</p> <p>5 Se actualizan las modificaciones efectuadas en la BD Control de Acceso finalizando el caso de uso.</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2 Si verifica que los datos no han sido enviados retorna un mensaje de error.</p> <p>3 Si los datos enviados no son correctos emite un mensaje de error indicando que los datos están incorrectos.</p> <p>4 Si verifica que los datos suministrados para las modificaciones existe en la BDCA retorna un mensaje de error “Esos datos ya existen”.</p>
<b>Sección “Eliminar área de acceso”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1 c) La intranet invoca el servicio eliminar área	2 Elimina el área de la BDCA dado su identificador.  3 Se actualizan los cambios en la BDCA finalizando el caso de uso.
---	---

**Tabla II.3.Caso de Uso\_ Buscar áreas**

<b>Caso de uso</b>	
CU_3	Buscar áreas
<b>Propósito</b>	Permite buscar un área determinada dada su descripción.
<b>Actores</b>	Intranet
<b>Resumen:</b> Se inicia cuando el actor desea buscar un área determinada e invoca el servicio buscar áreas enviando la descripción del área y devuelve el área perteneciente a este dato, finalizando el caso de uso.	
<b>Referencias</b>	RF_14
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1 Invoca el servicio buscar áreas enviando los datos necesarios del área:  a) Descripción área	2. Verifica que el dato de búsqueda haya sido enviado.  3. Verifica que el dato enviado esté correcto.  4 Verifica que el área dado el dato enviado existe en la BD Control de Acceso.  5 Retorna el área finalizando el caso de uso.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2 Si verifica que dato de búsqueda no fue enviado emite un mensaje de error.  3 Si verifica que el dato enviado no es correcto emite un mensaje de error indicando que el dato está incorrecto.  4 Si verifica que el área con ese dato no existe en la BD Control de Acceso emite un mensaje de error "Esa área no existe".

**Tabla II.4** Caso de Uso\_ Gestionar Bloques de Acceso

Caso de uso	
CU_4	Gestionar Bloques de Acceso
<b>Propósito</b>	El propósito de este caso de uso es permitir al actor registrar, modificar o eliminar un bloque de acceso determinado.
<b>Actores</b>	Intranet
<b>Resumen:</b> Se inicia cuando el actor desea realizar alguna de las siguientes operaciones: registrar, modificar o eliminar bloques de acceso, si decide registrar, invoca el servicio insertar bloques de accesos enviando los datos pertinentes, verifica que los mismos sean correctos y los inserta en la BDCA retornando su identificador(id bloque acceso), en caso de invocar los servicios actualizar bloque de acceso o eliminar bloque de acceso , de acuerdo al dato enviado actualiza la Base de Datos finalizando el caso de uso.	
<b>Referencias</b>	RF_5 RF_6 RF_7
Acción del actor	Respuesta del sistema
<p>1 La Intranet puede realizar algunas de las siguientes acciones: registrar, modificar o eliminar bloques de acceso</p> <p>a) Si decide Registrar un bloque de acceso ir a la sección “Registrar bloques de acceso”.</p> <p>b) Si decide modificar un bloque de acceso ir a la sección “Modificar bloques de acceso”.</p> <p>c) Si decide eliminar un bloque de acceso ir a la sección “Eliminar bloque de acceso”.</p>	
<b>Sección “Registrar bloques de acceso”</b>	

Acción del actor	Respuesta del sistema
1 a) La intranet invoca el servicio insertar bloques de acceso enviando los datos especificados en RF_ 5	2. Verifica que los datos hayan sido enviados. 3. Verifica que los datos enviados estén correctos. 4 Verifica que los datos enviados no existan en la Base de Datos Control de Acceso (BDCA). 5 Inserta el bloque de acceso en la BDCA y retorna un identificador (id bloque acceso) finalizando el caso de uso.
<b>Flujo Alternativo</b>	
Acción del actor	Respuesta del sistema
	2. Si verifica que los datos no han sido enviados emite un mensaje de error “Debe enviar todos los datos”. 3 Si verifica que los datos no son correctos emite un mensaje de error “Los datos están incorrectos”. 4 Si verifica que los datos enviados existen en la Base de Datos Control de Acceso (BDCA) retorna un mensaje de error “Ese bloque ya existe en la Base de Datos”.
<b>Sección “Modificar bloques de acceso”</b>	
Acción del actor	Respuesta del sistema
1 b) La intranet invoca el servicio actualizar bloque de acceso enviando los datos especificados en RF_6	2. Verifica que los datos hayan sido enviados. 3 Verifica que los datos enviados para realizar las modificaciones estén correctos. 4 Verifica que los datos enviados no existan en la BDCA. 5 Se actualizan las modificaciones efectuadas en la BD Control de Acceso dado los datos enviados para efectuarlas, finalizando el

	caso de uso.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2. Si verifica que los datos no han sido enviados emite u mensaje de error Debe enviar todos los datos”.</p> <p>3 Si los datos enviados no son correctos se emite un mensaje de error “Los datos están incorrectos”.</p> <p>4 Si verifica que los datos enviados para las modificaciones existen en la BDCA retorna un mensaje de error “Ese bloque de acceso ya esta registrado”.</p>
<b>Sección “Eliminar área de acceso”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1 c) La intranet invoca el servicio eliminar bloque de acceso	<p>2 Elimina el bloque de acceso de la Base Datos Control de Acceso (BDCA) dado su identificador (id_bloque acceso).</p> <p>3 Se actualizan los cambios en la base de datos.</p>

**Tabla II.5.Caso de Uso\_ Buscar bloques de acceso**

<b>Caso de uso</b>	
CU_5	Buscar bloques de acceso
<b>Propósito</b>	Permite buscar un bloque de acceso determinado dada su Descripción
<b>Actores</b>	Intranet
<b>Resumen:</b> Se inicia por el actor cuando va a buscar un bloque de acceso determinado e invoca el servicio buscar bloques de acceso dada la Descripción del bloque enviado y se devuelve, finalizando el caso de uso.	
<b>Referencias</b>	RF_15
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1.La Intranet invoca el servicio Buscar bloques de acceso	2 Verifica que el dato de búsqueda haya sido enviado.



<p>enviando los datos necesarios para realizar la búsqueda:</p> <p>a) Descripción</p>	<p>3 Verifica que el dato enviado esté correcto.</p> <p>4 Verifica que el bloque con ese dato existe en la BD Control de Acceso.</p> <p>5 Retorna el bloque de acceso con sus datos correspondientes finalizando el caso de uso.</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2. Si verifica que el dato de búsqueda no ha sido enviado emite un mensaje de error “Debe enviar el dato de búsqueda”.</p> <p>3. Si verifica que el dato enviado no está correcto emite un mensaje de error “El dato enviado está incorrecto “.</p> <p>4 Si verifica que el bloque de acceso con ese dato no existe en la BD Control de Acceso emite un mensaje de error “Ese bloque no existe”.</p>

**Tabla II.6** Caso de Uso\_ Gestionar niveles de acceso

<b>Caso de uso</b>	
CU_6	Gestionar niveles de acceso
<b>Propósito</b>	El propósito de este caso de uso es permitir al actor asignar niveles de accesos por áreas a personas permanentes de la institución (trabajadores y profesores), modificar o eliminar dichos niveles de acceso.
<b>Actores</b> Intranet	
<p><b>Resumen:</b> Se inicia cuando el actor desea realizar alguna de las siguientes operaciones : asignar nivel de acceso por área al personal (trabajadores y profesores) permanentes de la institución, modificar el nivel acceso, o eliminarlo, en caso de asignar invoca el servicio insertar acceso persona y dado los datos enviados de la persona asigna el nivel de acceso, en caso de modificar o eliminar, invoca los servicios actualizar acceso persona o eliminar acceso persona en caso de no haber sido verificada su entrada al</p>	

sistema y en dependencia de los datos enviados por el actor actualiza o elimina de la BDCA, finalizando el caso de uso.	
<b>Referencias</b>	RF_8 RF_9 RF_10
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1 La Intranet puede realizar algunas de las siguientes acciones: Asignar nivel de acceso, modificar nivel de acceso o eliminar nivel de acceso</p> <p>a) Si decide asignar nivel de acceso ir a la sección “Asignar nivel de acceso”.</p> <p>b) Si decide modificar nivel de acceso ir a la sección “Modificar nivel de acceso”.</p> <p>c) Si decide modificar nivel de acceso ir a la sección “Modificar nivel de acceso”.</p>	
<b>Sección “Asignar nivel de acceso”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1 a) La intranet invoca el servicio insertar acceso persona enviando los datos especificados en RF_8</p>	<p>2 Verifica que los datos hayan sido enviados.</p> <p>3 Verifica que dentro de los datos el solapín enviado no tenga asignado ningún nivel de acceso.</p> <p>4 Le asigna el nivel de acceso a las áreas a las cuales tiene acceso y se almacenan en la BD Control de Acceso (BDCA), finalizando el caso de uso.</p>
<b>Flujo Alternativo</b>	

Acción del actor	Respuesta del sistema
	<p>2 Si verifica que los datos no han sido enviados emite un mensaje de error “Debe enviar los datos de búsqueda”,</p> <p>3 Si verifica que el solapín obtenido de esa persona tiene asignado algún nivel de acceso se emite un mensaje de error “Ese solapín ya tiene asignado un nivel de acceso a una determinada área para realizar otra asignación debe modificar”,</p>
<b>Sección “Modificar nivel de acceso”.</b>	
Acción del actor	Respuesta del sistema
<p>1 b) La intranet invoca el servicio actualizar acceso persona enviando los datos especificados en RF_9</p>	<p>2 Verifica que los datos hayan sido enviados.</p> <p>3 Verifica que la persona no haya entrado al sistema para poder efectuar dichas modificaciones.</p> <p>4 Se actualizan las modificaciones de la asignación de los accesos efectuados a personas por áreas en la BD Control de Acceso (BDCA) de acuerdo a los datos enviados finalizando el caso de uso.</p>
<b>Flujo alternativo</b>	
Acción del actor	Respuesta del sistema
	<p>2 Si verifica que los datos no han sido enviados emite un mensaje de error “Debe enviar los parámetros de búsqueda “.</p> <p>3 Si verifica que la persona esta dentro del sistema envía un mensaje de error “No puede realizar modificaciones en estos momentos, la persona debe salir del sistema”.</p>
<b>Sección “Eliminar nivel de acceso</b>	
Acción del actor	Respuesta del sistema
<p>1 c) La intranet invoca el servicio eliminar acceso persona enviando los datos pertinentes: solapín y/o</p>	<p>2 Verifica que los datos hayan sido enviados.</p>

Cl.	<p>3. Verifica que la persona con esos datos no este dentro del sistema</p> <p>4. Elimina los accesos a las áreas asignadas, de acuerdo a los datos enviados finalizando el caso de uso</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2 Si verifica que los datos no han sido enviados, emite un mensaje de error "Debe enviar los parámetros de búsqueda. "</p> <p>3 Si Verifica que la persona está dentro del sistema envía un mensaje de error "No puede eliminar los accesos asignados en estos momentos, la persona debe salir del sistema".</p>

**Tabla II.7** Caso de Uso\_ Buscar datos personales

<b>Caso de uso</b>	
CU_7	Buscar datos personales
<b>Propósito</b>	Permite buscar los datos de las personas para verificar su acceso o no al centro, a través de un parámetro de búsqueda, el solapín.
<b>Actores</b>	Intranet
<b>Resumen:</b> Se inicia por el actor cuando va a buscar los datos de una persona para verificar el acceso o no, invocando el servicio buscar datos personales, determinando la categoría de la persona en cuestión, y realizando la búsqueda con dicha categoría, si es trabajador, visitante o profesor a través del servicio "busqueda_simple" del Servicio Web Gestion_Usuario y en la Base de datos Control de acceso en caso de la categoría ser un visitante.	
<b>Referencias</b>	RF_17
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1 La Intranet invoca el servicio: buscar datos personales enviando los datos necesarios para realizar la	<p>2. Verifica que el dato de búsqueda haya sido enviado.</p> <p>3. Verifica que el dato enviado esté correcto.</p> <p>4. Obtiene la categoría de la persona en cuestión. (Trabajador,</p>

búsqueda: Solapín.	<p>directivo, profesor o visitante), dado este dato enviado.</p> <p>5 Dada esta categoría, si es trabajador, profesor o directivo invoca el servicio “Búsqueda simple de usuario” del WS Gestionar Usuario, realiza la búsqueda en la base de datos Ciudadano.</p> <p>6 Retorna la persona con sus datos correspondientes finalizando el caso de uso.</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2. Si verifica que el dato de búsqueda no ha sido enviado emite un mensaje de error “Debe enviar el parámetro de búsqueda “.</p> <p>3. Si verifica que el dato enviado no es correcto emite un mensaje de error indicando que los datos están incorrectos.</p> <p>5 Si es visitante realiza la búsqueda en la base de datos Control_Acceso, y devuelve los datos requeridos de la persona.</p>

**Tabla II.8** Caso de Uso\_ Gestionar datos visitantes

<b>Caso de uso</b>	
CU_8	Gestionar datos visitantes
<b>Propósito</b>	El propósito de este caso de uso es permitir al actor registrar, modificar o eliminar datos de los visitantes que acceden al solapín.
<b>Actores</b>	Intranet
<b>Resumen:</b> Se inicia cuando el actor desea realizar algunas de las siguientes operaciones: registrar datos de visitante, modificarlos o eliminarlos, en caso de registrar invoca el servicio insertar visitante y dado los datos enviados se registraría en la base de datos, en caso de querer modificar o eliminar invoca los servicios actualizar visitante o eliminar visitante verificando previamente que el visitante no haya entrado al sistema y dado estos datos, actualiza en la base de datos las modificaciones o elimina, finalizando de esta forma el caso de uso.	
<b>Referencias</b>	RF_11 RF_12 RF_13
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

<p>1 La Intranet puede realizar algunas de las siguientes acciones: Registrar visitantes, modificar datos o Eliminar datos de visitantes.</p> <p>a) Si decide Registrar visitantes ir a la sección “Registrar de datos visitantes”.</p> <p>b) Si decide modificar datos ir a la sección “Modificar datos de visitantes”.</p> <p>c) Si decide eliminar datos ir a la sección “Eliminar datos de visitantes”.</p>	
<b>Sección “Registrar datos de visitantes”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1 a) La intranet invoca el servicio insertar visitante enviando los datos especificados en RF_11</p>	<p>2 Verifica que los datos hayan sido enviados.</p> <p>3 Verifica que los datos enviados estén correctos.</p> <p>4 Verifica que los datos enviados no estén registrados en la base de datos control de acceso (BDCA).</p> <p>5 Inserta el visitante con los datos enviados en la base de datos (BDCA), finalizando el caso de uso.</p>
<b>Flujo Alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2 Si verifica que los datos no han sido enviados emite un mensaje</p>

	<p>de error “Debe enviar los datos de búsqueda“.</p> <p>3 Si verifica que los datos enviados no están correctos emite un mensaje de error, indicando que los datos están incorrectos.</p> <p>4 Si verifica que los datos enviados del visitante están registrados en la base de datos (BDCA) se emite un mensaje de error “Ese visitante ya está registrado”.</p>
<b>Sección “Modificar datos de visitantes”.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1 b) La intranet invoca el servicio actualizar visitante enviando los datos especificados en RF_12.	<p>2 Verifica que los datos hayan sido enviados.</p> <p>3 Verifica que los datos enviados para las modificaciones estén correctos.</p> <p>4 Verifica que el visitante no haya entrado al sistema.</p> <p>5 Se actualizan las modificaciones efectuadas con los datos enviados en la BD Control de Acceso finalizando el caso de uso.</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2 Si verifica que los datos no han sido enviados emite un mensaje de error “Debe enviar los datos de búsqueda “.</p> <p>3 Si verifica que los datos enviados no son correctos emite un mensaje de error informando que los datos están incorrectos.</p> <p>4 Si verifica que el visitante ha entrado al sistema emite un mensaje de error “No pude modificar los datos del visitante”.</p>
<b>Sección “Eliminar datos de visitantes”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

<p>1 c) La intranet invoca el servicio eliminar visitante enviando los siguientes datos:</p> <p>Solapín</p> <p>CI</p>	<p>2. Verifica que los datos hayan sido enviados.</p> <p>3. Verifica que los datos enviados para eliminar estén correctos.</p> <p>4. Verifica que el visitante no haya entrado al sistema y elimina sus datos, finalizando el caso de uso.</p>
<p><b>Flujo alternativo</b></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
	<p>2. Si verifica que los datos no han sido enviados emite un mensaje de error.</p> <p>3. Si verifica que los datos enviados para eliminar no están correctos emite un mensaje de error “Los datos están incorrectos”.</p> <p>4 Si verifica que el visitante ha entrado al sistema muestra un mensaje de error “No puede eliminar los datos”.</p>

**Tabla II.9** Caso de Uso\_ Buscar visitantes registrados

<p><b>Caso de uso</b></p>	
<p>CU_9</p>	<p>Buscar visitantes registrados</p>
<p><b>Propósito</b></p>	<p>Permite buscar visitantes registrados en la base de datos Control de Acceso.</p>
<p><b>Actores</b> Intranet</p>	
<p><b>Resumen:</b> Se inicia por el actor cuando va a buscar un visitante determinado e invoca el servicio buscar_ visitante enviando los datos pertinentes, se realiza la búsqueda en la base de datos control de acceso, se devuelven los datos esperados, finalizando el caso de uso.</p>	
<p><b>Referencias</b></p>	<p>RF_16</p>
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
<p>1 La Intranet invoca el servicio: buscar visitante registrados enviando los datos necesarios</p>	<p>2 Verifica que los datos hayan sido enviados.</p>



para realizar la búsqueda: Nombre o carnet de identidad (CI)	<p>3 Verifica que los datos enviados estén correctos.</p> <p>4 Verifica que esos datos existan en la base de datos Control de Acceso.</p> <p>5 Retorna el visitante con sus datos correspondientes finalizando el caso de uso.</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2 Si verifica que los datos no han sido enviados emite un mensaje de error.</p> <p>3 Si verifica que los datos enviados no son correctos emite un mensaje de error indicando que están incorrectos.</p> <p>4 Si verifica que los datos enviados no existen en la base de datos ciudadano emite un mensaje de error "Ese visitante con esos datos no existe".</p>

**Tabla II.10** Caso de Uso\_ Buscar accesos

<b>Caso de uso</b>	
CU_10	Buscar accesos
<b>Propósito</b>	Permite buscar accesos efectuados a la institución.
<b>Actores</b>	Intranet
<b>Resumen:</b> Se inicia cuando el actor desea buscar las personas que han accedido a la institución en un momento determinado, la intranet invoca el servicio buscar accesos enviando los datos de búsquedas, ya sea por fecha, categorías o combinaciones de ellos, se devuelven los datos pertinentes finalizando el caso de uso.	
<b>Referencias</b>	RF_19
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1 La Intranet invoca el servicio:	2 Verifica que los datos hayan sido enviados.

<p>buscar accesos enviando los datos necesarios para realizar la búsqueda: fecha /o categorías o ambos.</p>	<p>3 Verifica que esos datos existan en la base de datos Control de Acceso.</p> <p>4 Devuelve un registro con todos los datos de las personas que han accedido al centro finalizando el caso de uso.</p>
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2 Si verifica que los datos no han sido enviados emite un mensaje de error.</p> <p>3 Si verifica que los datos enviados no existen en la base de datos emite un mensaje de error “Esos datos de búsquedas no existen en la base de datos”.</p>

**Tabla II.11** Caso de Uso\_ Mostrar estadísticas personas\_ categorías activa

<b>Caso de uso</b>	
CU_11	Mostrar estadísticas personas_ categorías activa
<b>Propósito</b>	Permite mostrar la cantidad de personas registradas por categorías que hay dentro del sistema un momento determinado.
<b>Actores</b> Intranet	
<p><b>Resumen:</b> Se inicia cuando el actor desea observar la cantidad de personas de las diferentes categorías (Directivos, profesor, invitados, trabajador) que hay dentro del sistema en un momento determinado, invocando el servicio Mostrar estadísticas personas_ categorías activa devolviendo la cantidad de personas por las diferentes categorías que han entrado al sistema finalizando el caso de uso.</p>	
<b>Referencias</b>	RF_18 RF_18.1 RF_18.2 RF_18.3 RF_18.4
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1 La Intranet invoca el servicio Mostrar estadísticas personas_ categorías activa	2 Devuelve un listado con la cantidad de personas que hay dentro del sistema un momento determinado por categorías, finalizando el caso de uso.

### Anexo III Diagramas de secuencias por escenarios de Casos de Usos

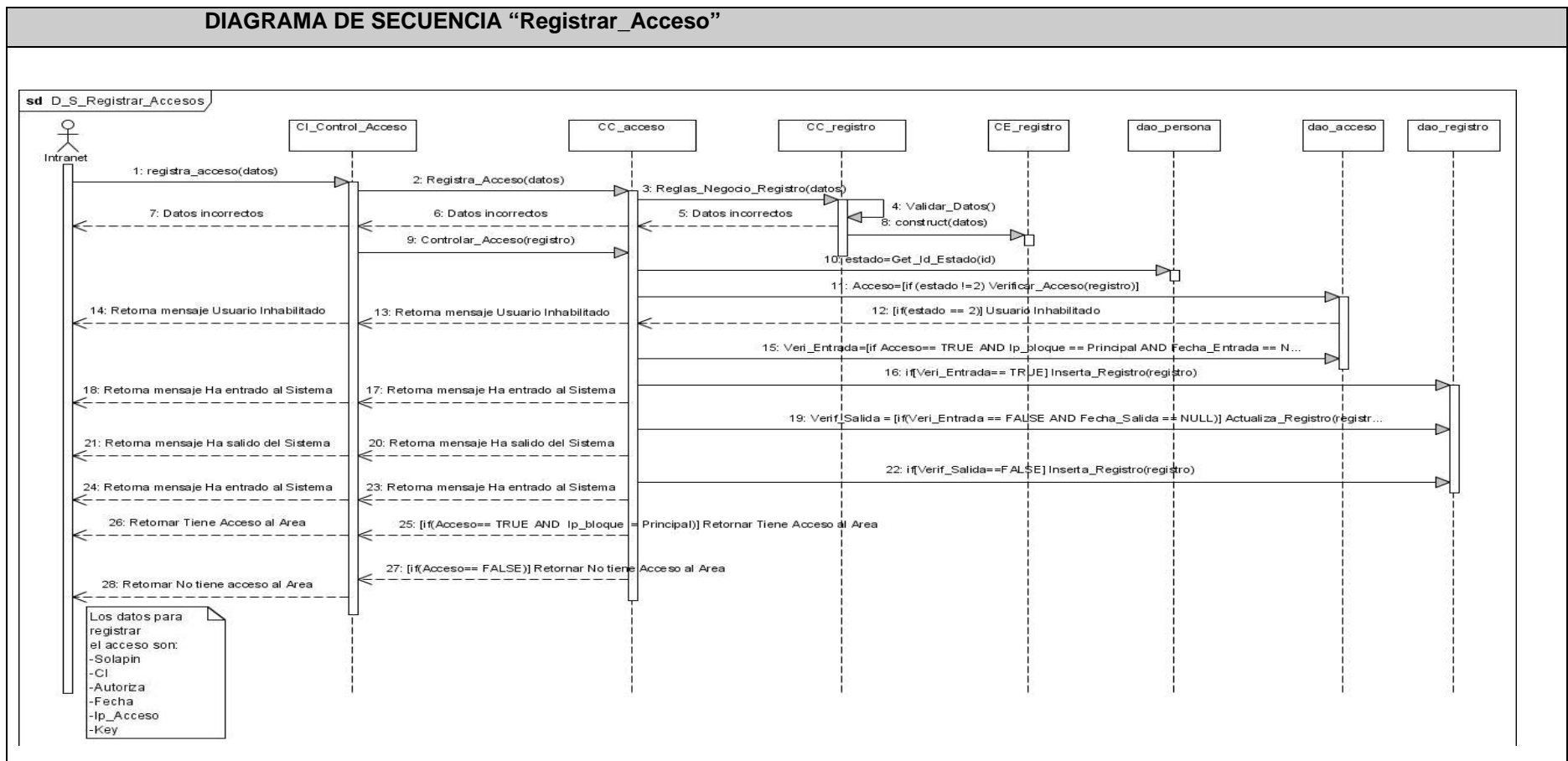


Figura III.1 Diagrama de Secuencia\_ Registrar\_Accesos

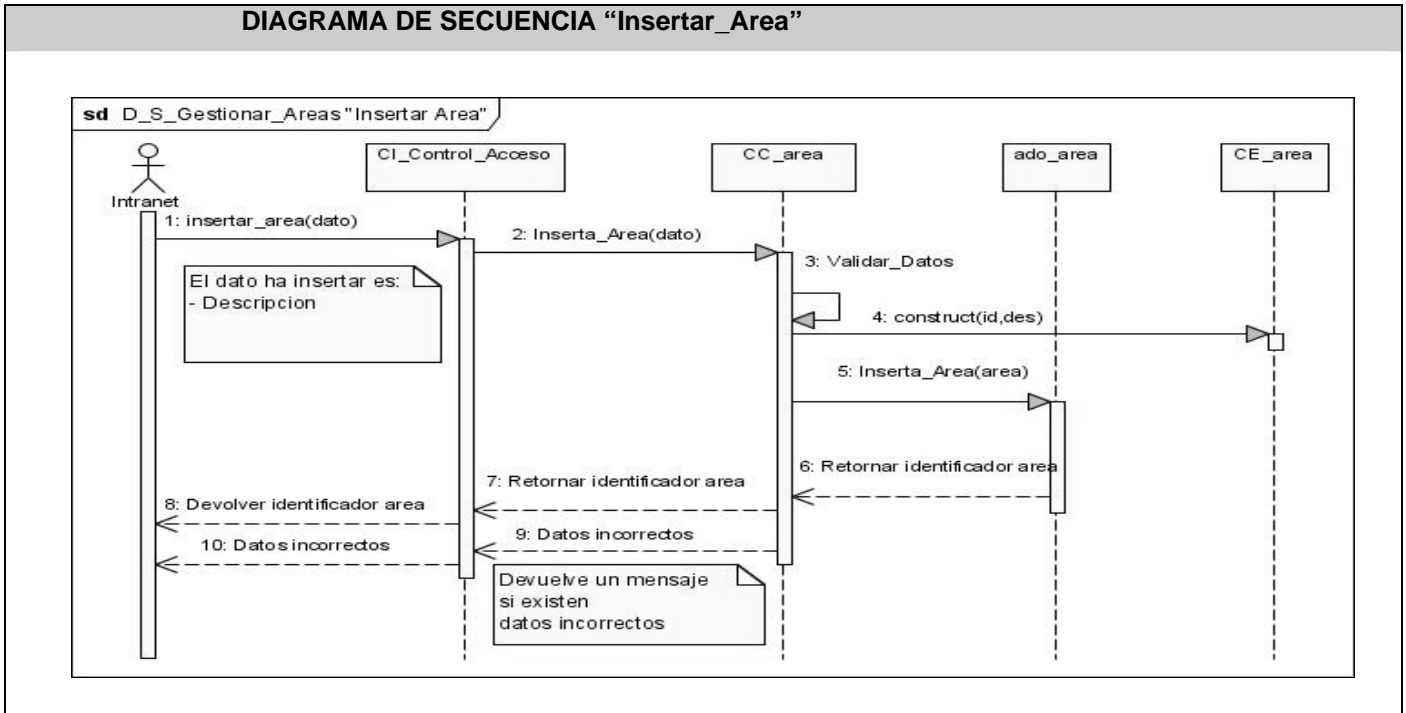


Figura III.2 Diagrama de Secuencia\_ Insertar\_Area

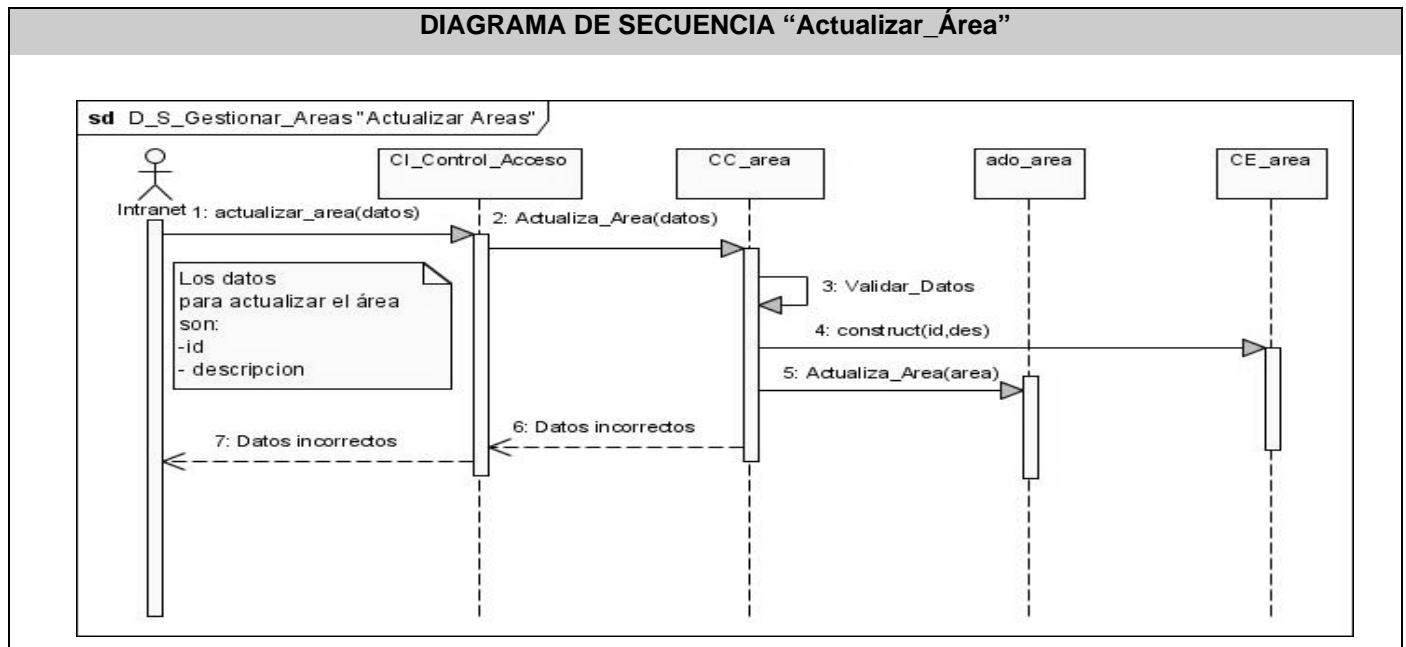


Figura III.3 Diagrama de Secuencia\_ Actualizar\_Area

DIAGRAMA DE SECUENCIA "Eliminar\_Area"

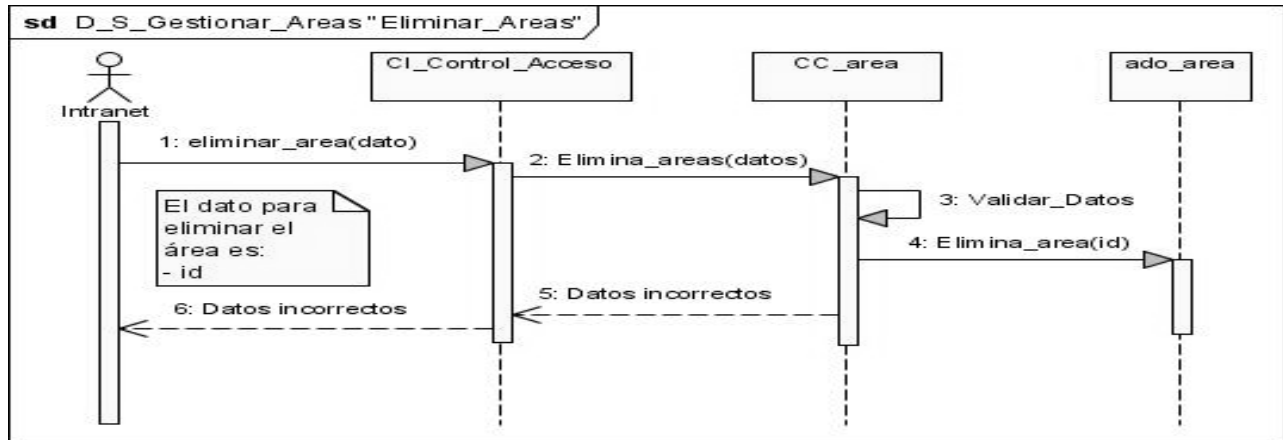


Figura III.4 Diagrama de Secuencia\_Eliminar\_Area

DIAGRAMA DE SECUENCIA "Buscar\_Areas"

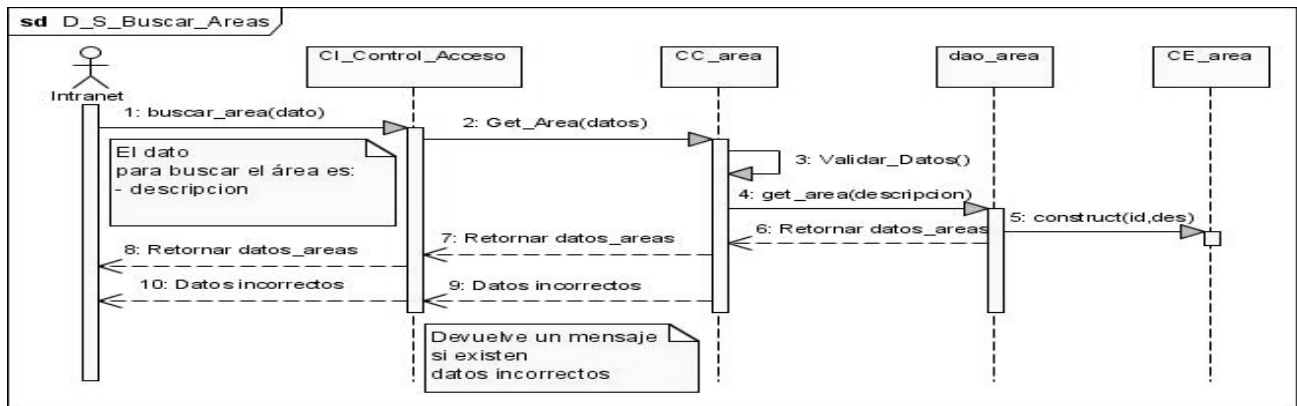


Figura III.5 Diagrama de Secuencia\_Buscar\_Areas

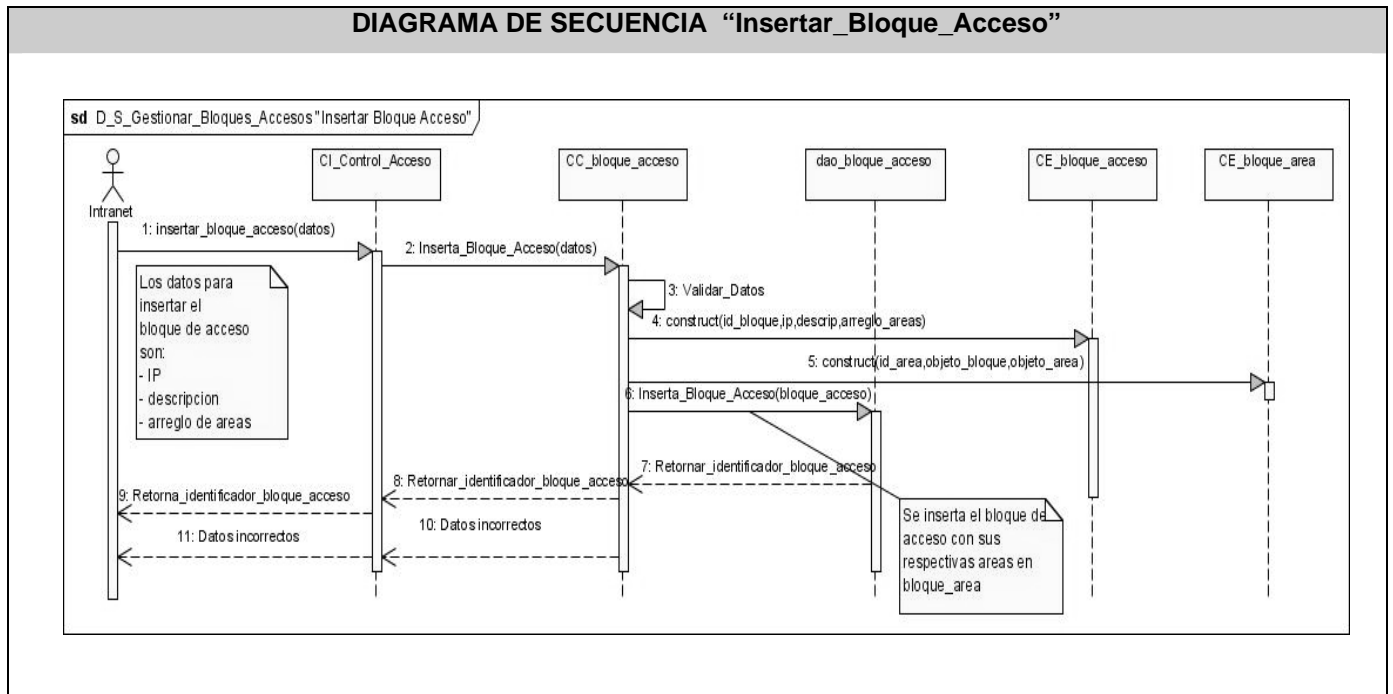


Figura III.6 Diagrama de Secuencia\_ Insertar\_Bloque\_Acceso

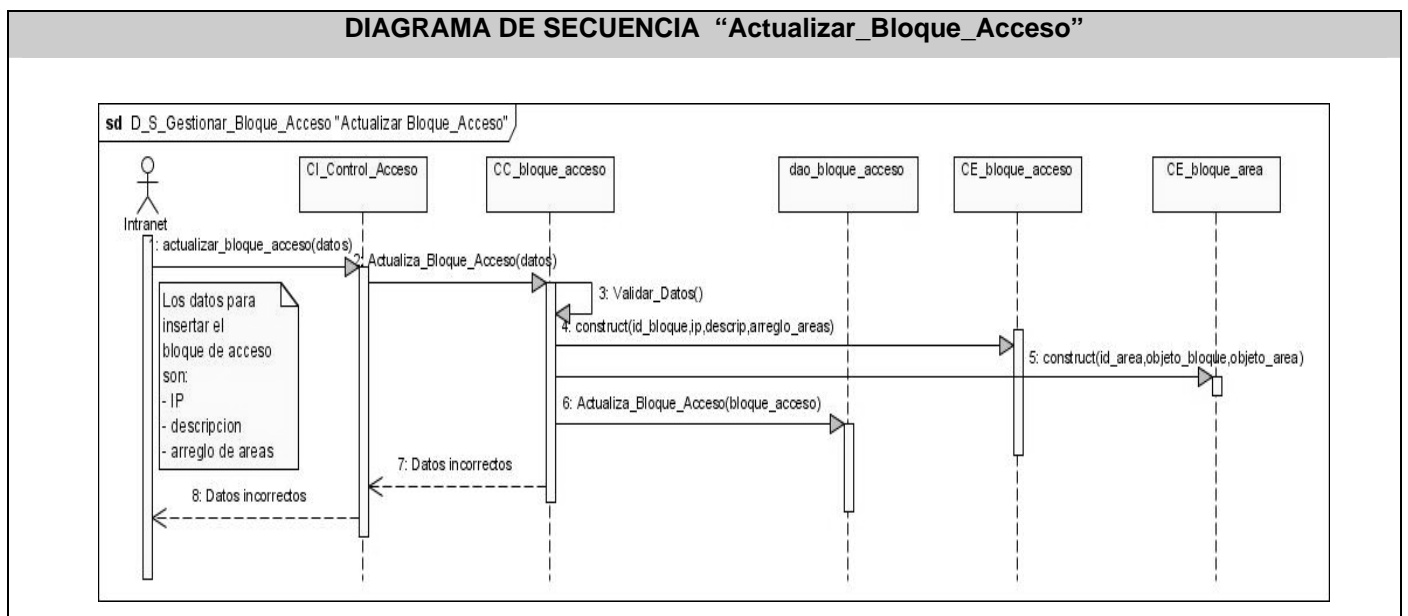


Figura III.7 Diagrama de Secuencia\_ Actualizar\_Bloque\_Acceso

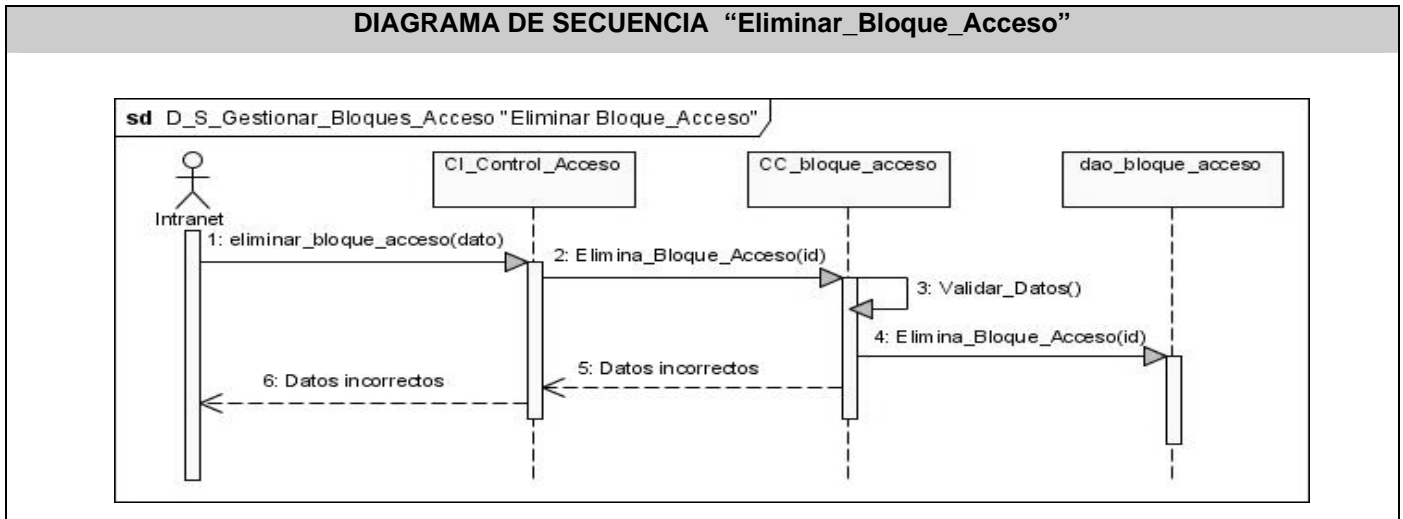


Figura III.8 Diagrama de Secuencia\_ Eliminar\_Bloque\_Acceso

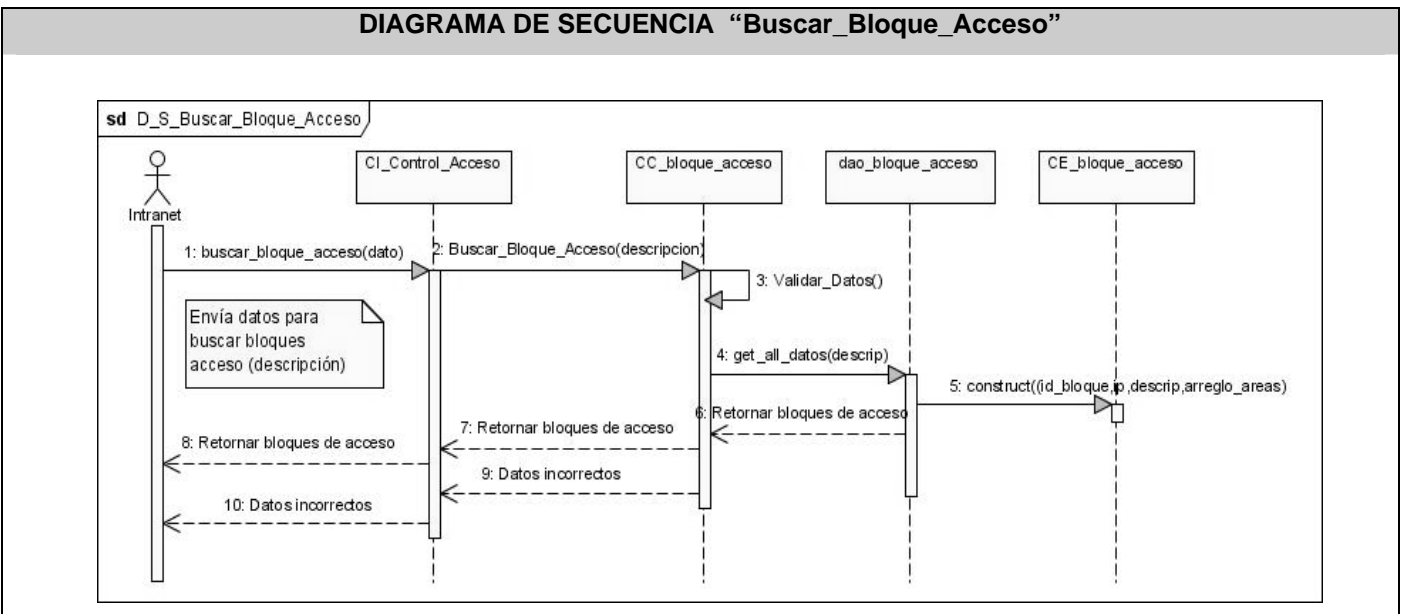


Figura III.9 Diagrama de Secuencia\_ Buscar\_Bloque\_Acceso

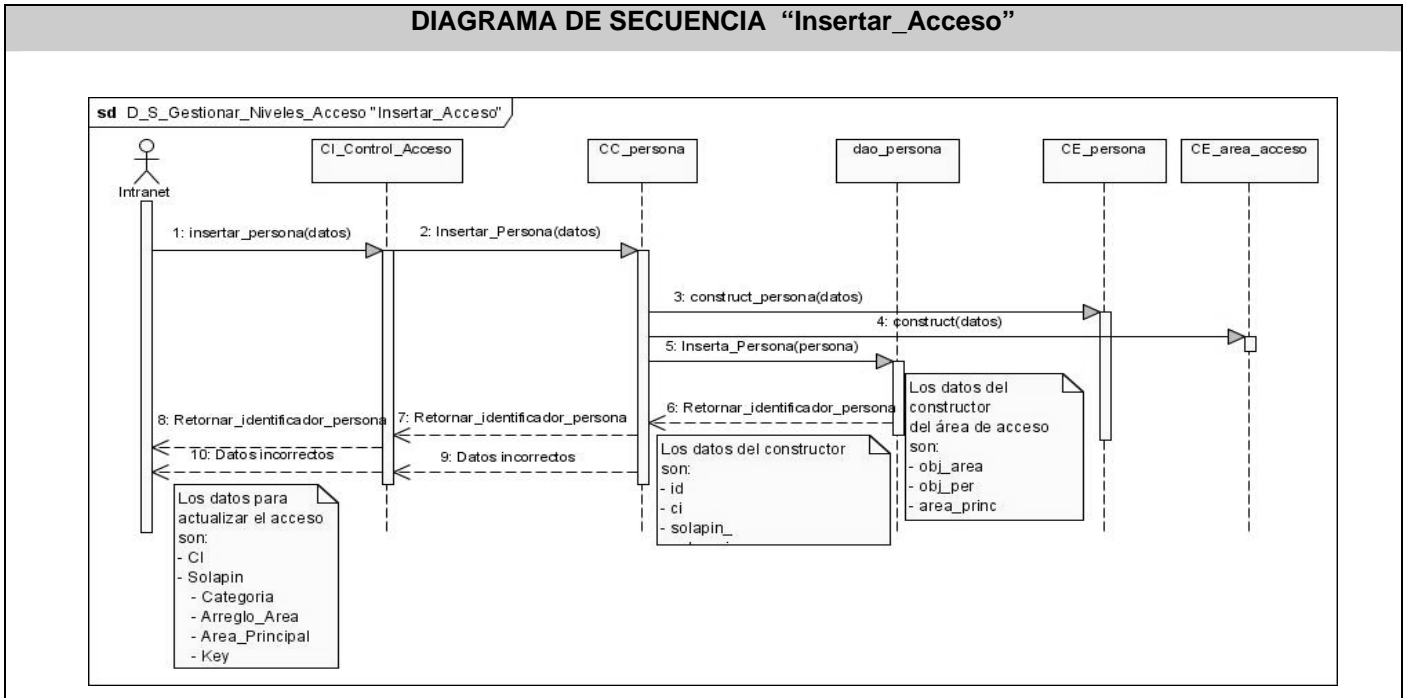


Figura III.10 Diagrama de Secuencia\_ Insertar\_Acceso

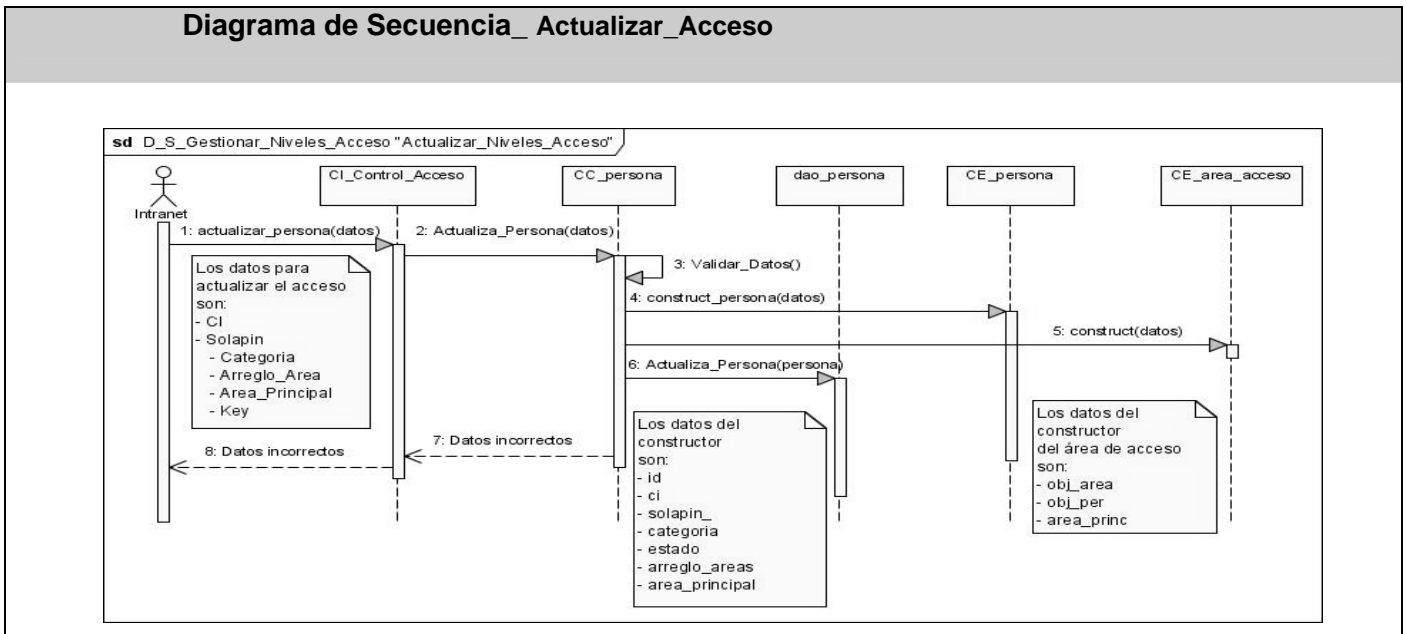


Figura III.11 Diagrama de Secuencia\_ Actualizar\_Acceso



DIAGRAMA DE SECUENCIA "Eliminar\_Accesos"

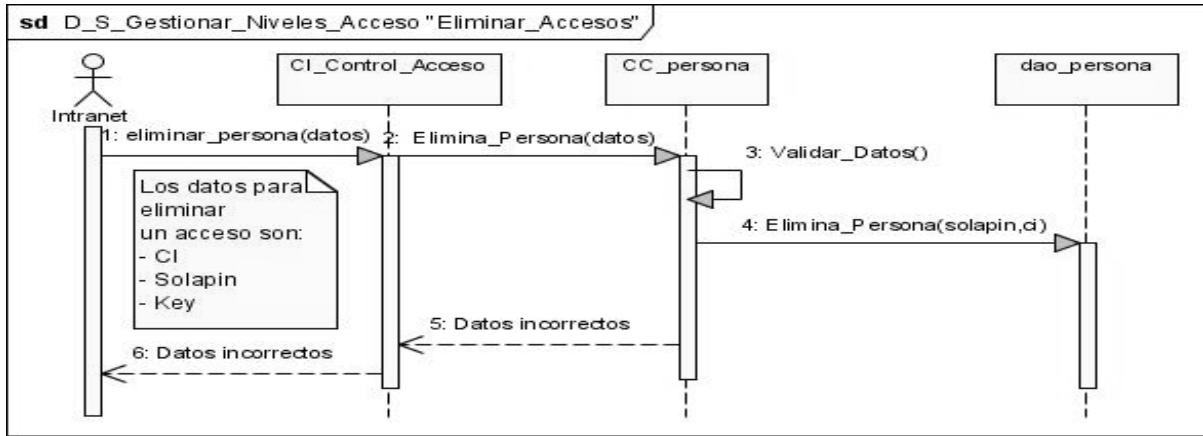


Figura III.12 Diagrama de Secuencia\_Eliminar\_Acceso

DIAGRAMA DE SECUENCIA "Insertar\_Visitante"

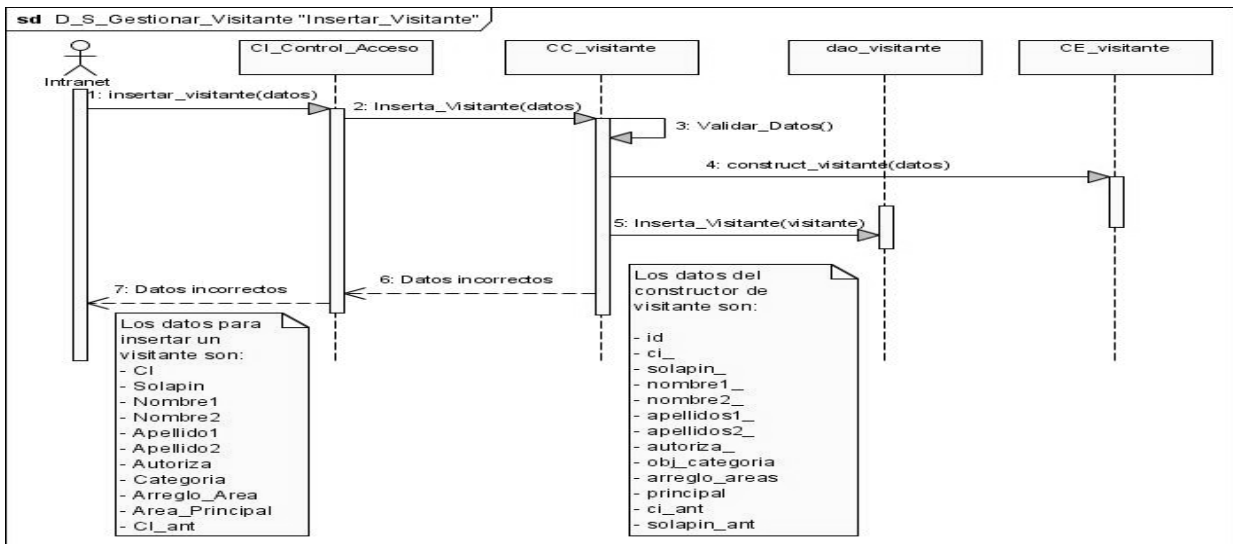


Figura III.13 Diagrama de Secuencia\_Insertar\_Visitante

**DIAGRAMA DE SECUENCIA "Actualizar\_Visitante"**

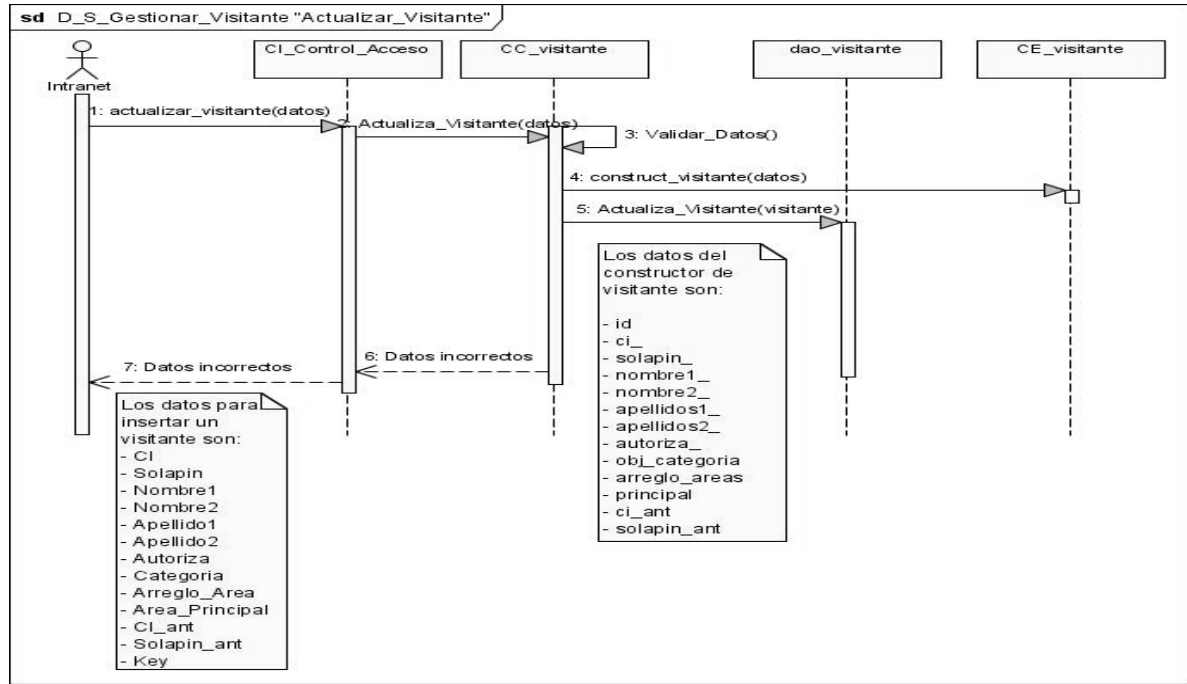


Figura III.14 Diagrama de Secuencia\_ Actualizar\_Visitante

**DIAGRAMA DE SECUENCIA "Eliminar\_Visitante"**

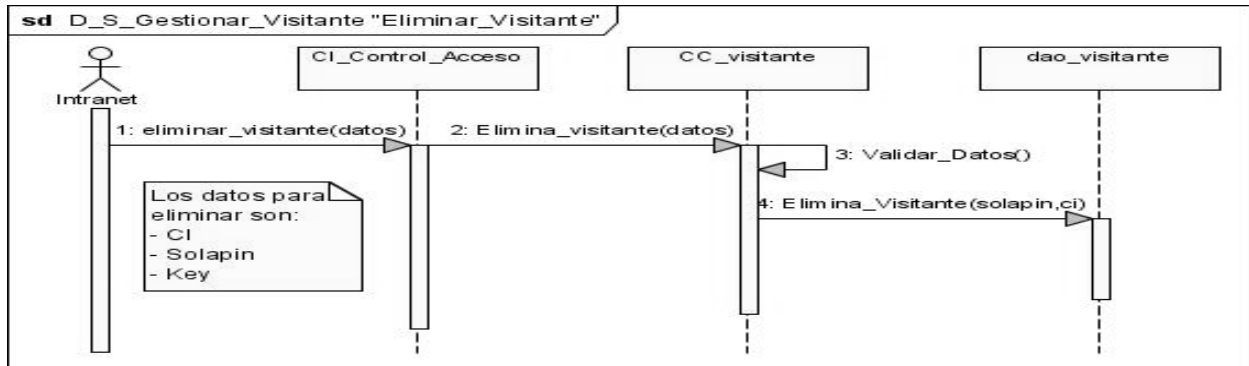


Figura III.15 Diagrama de Secuencia\_ Eliminar\_Visitante

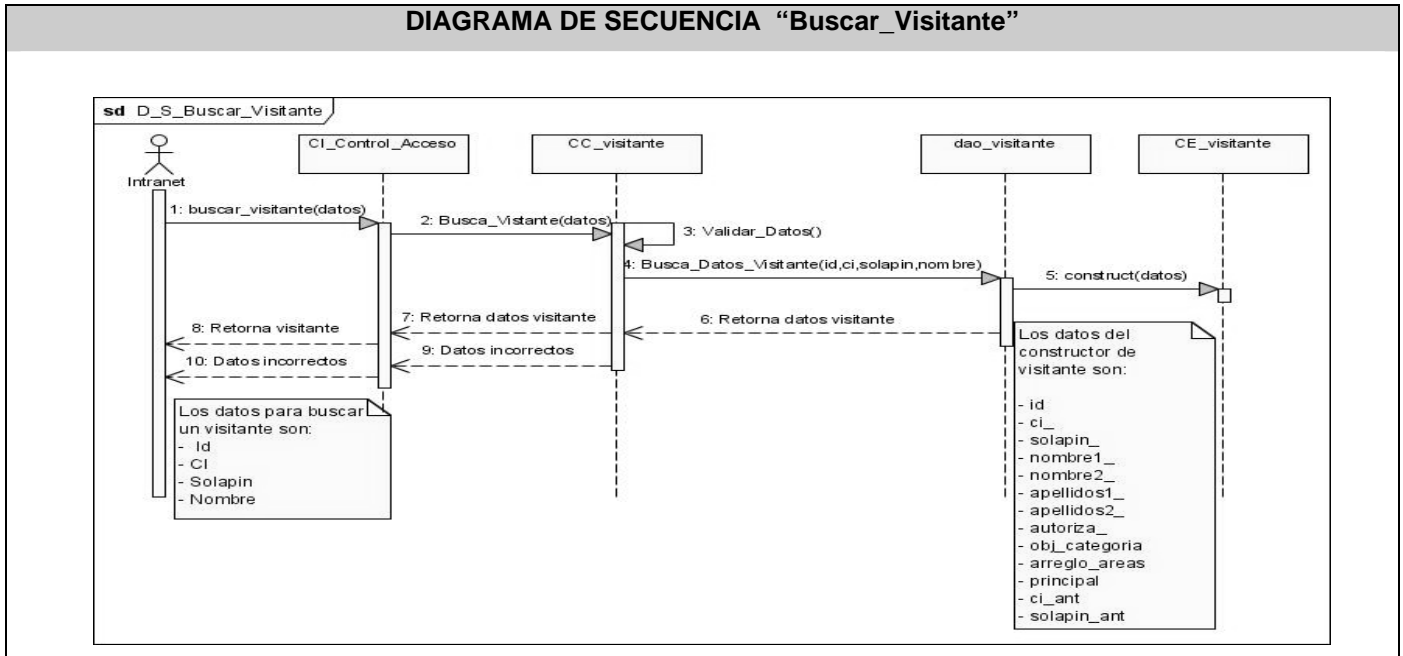


Figura III.16 Diagrama de Secuencia\_Buscar\_Visitante

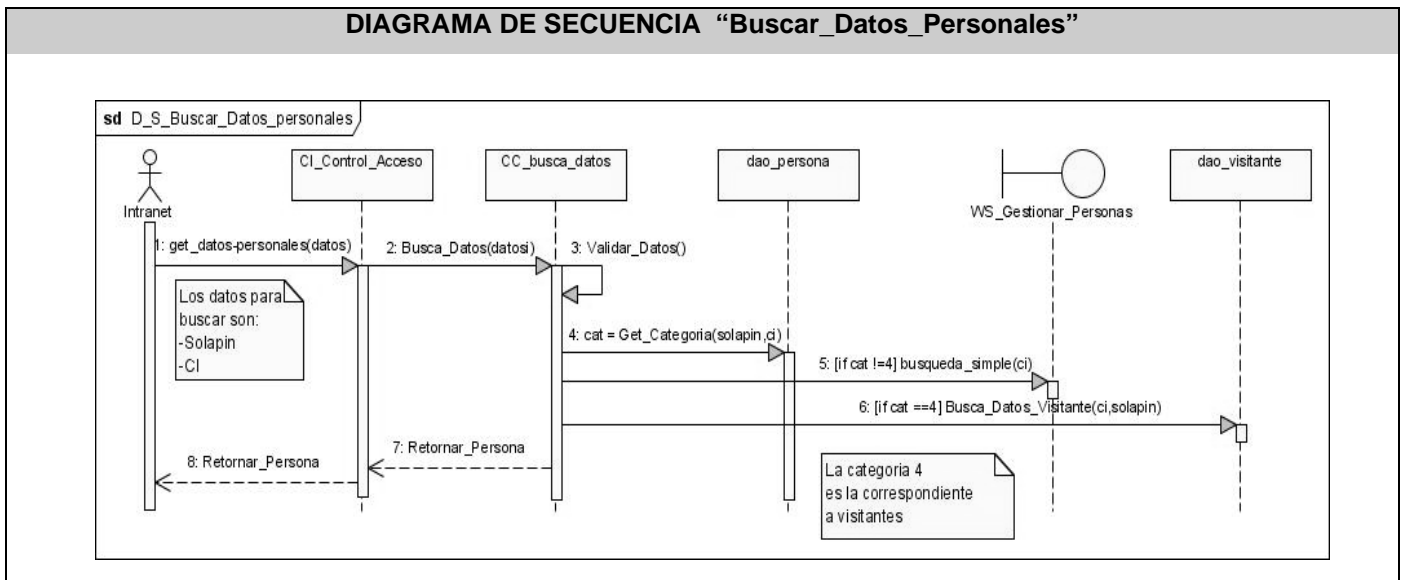


Figura III.17 Diagrama de Secuencia\_Buscar\_Datos\_Personales

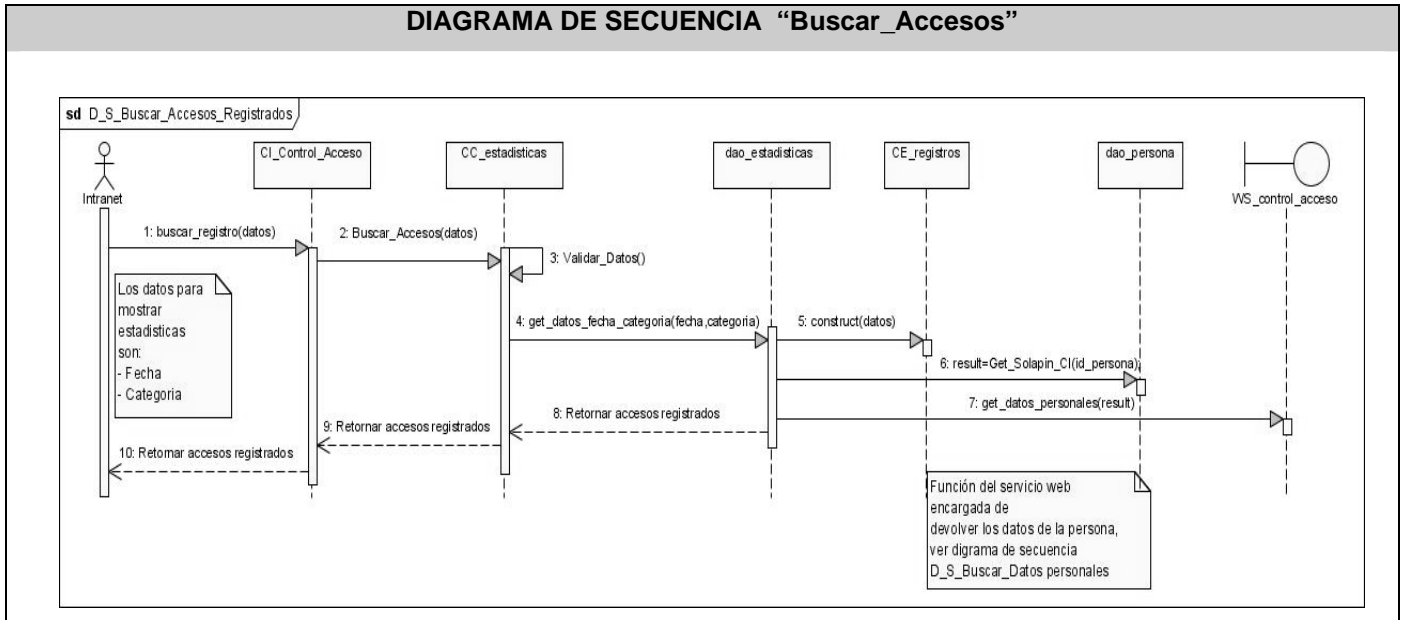


Figura III.18 Diagrama de Secuencia\_ Buscar\_Accesos

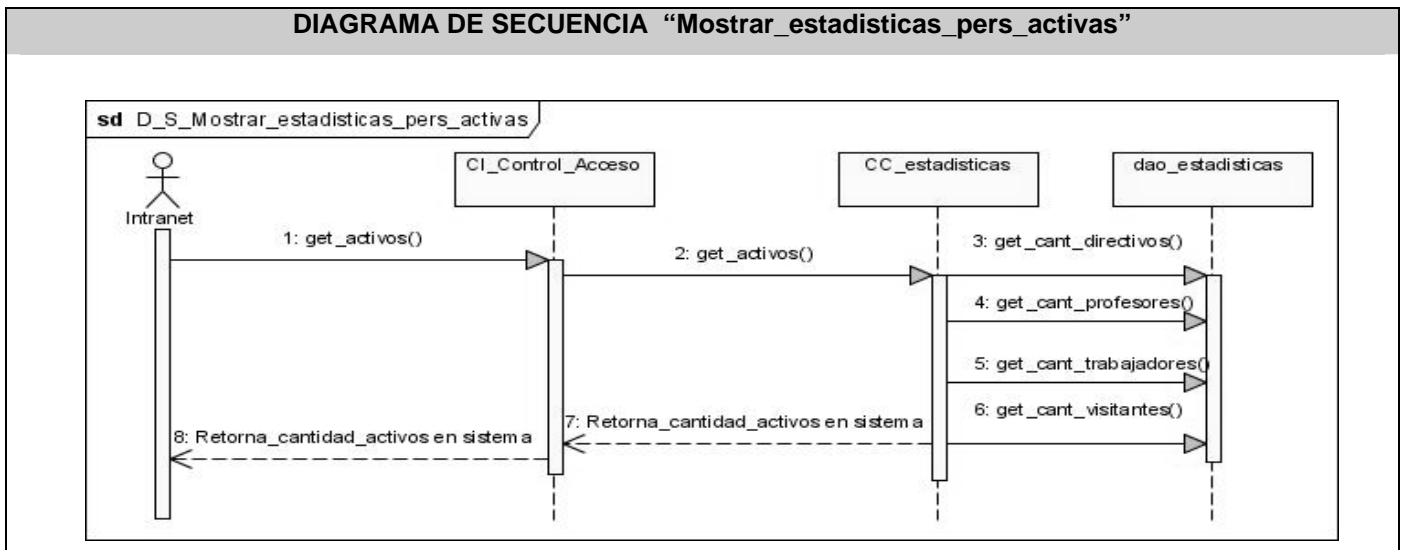


Figura III.19 Diagrama de Secuencia\_ Mostrar\_estadisticas\_pers\_activas

## Anexo IV Diagrama de clases del diseño por paquetes

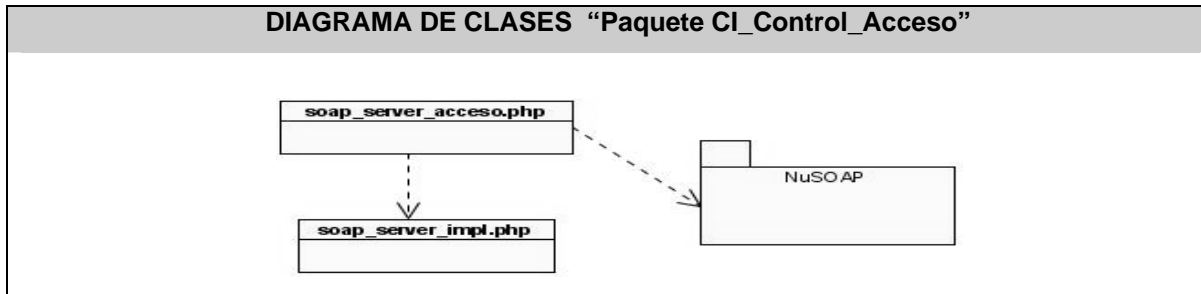


Figura IV.1 Diagrama de Clases Paquete Interfaz

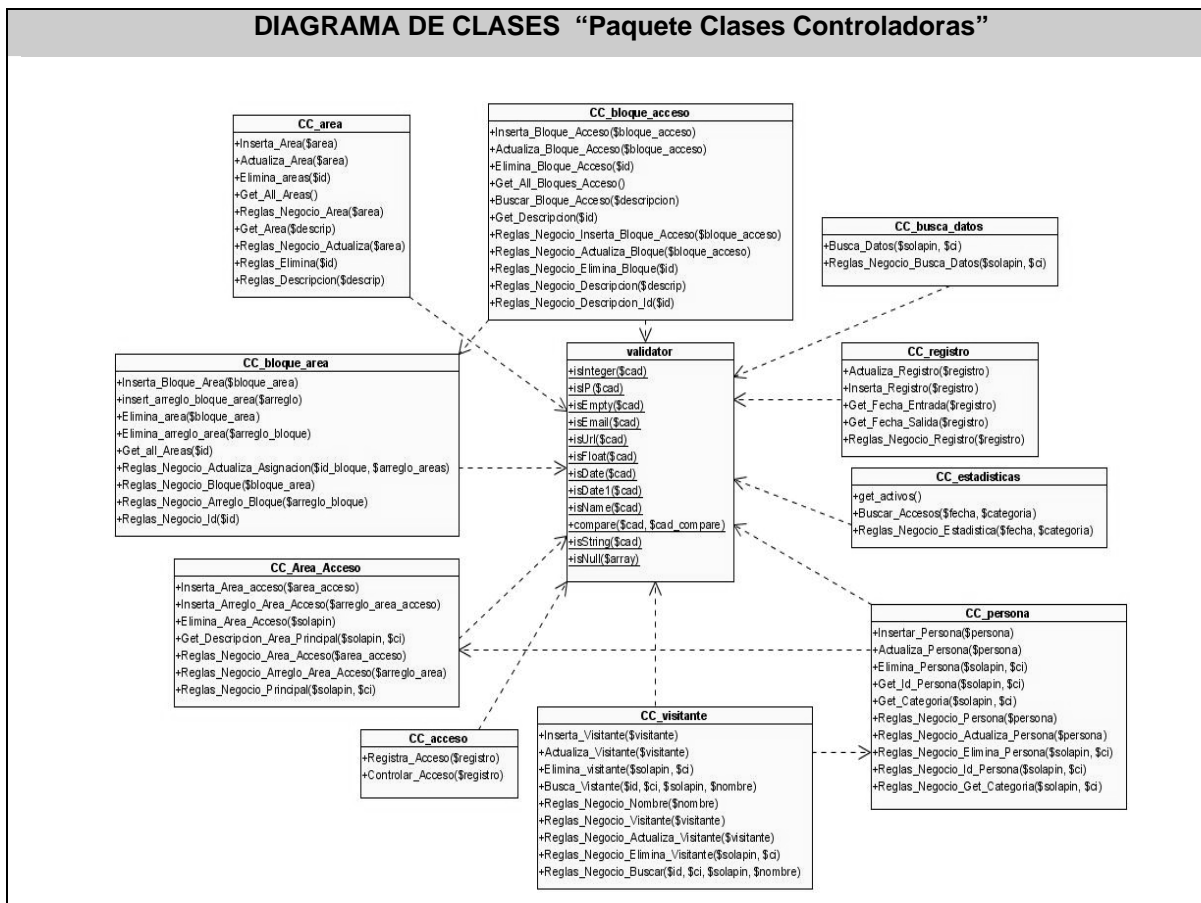


Figura IV.2 Diagrama de Clases Controladoras

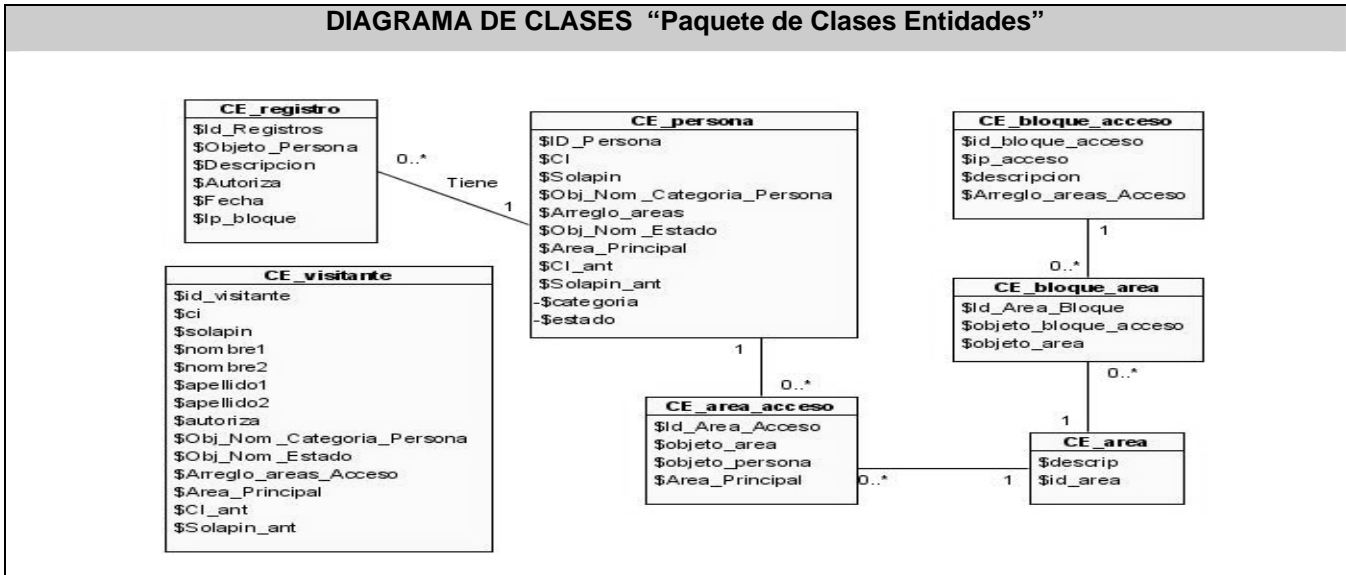


Figura IV.3 Diagrama de Clases Paquete Clases Entidades

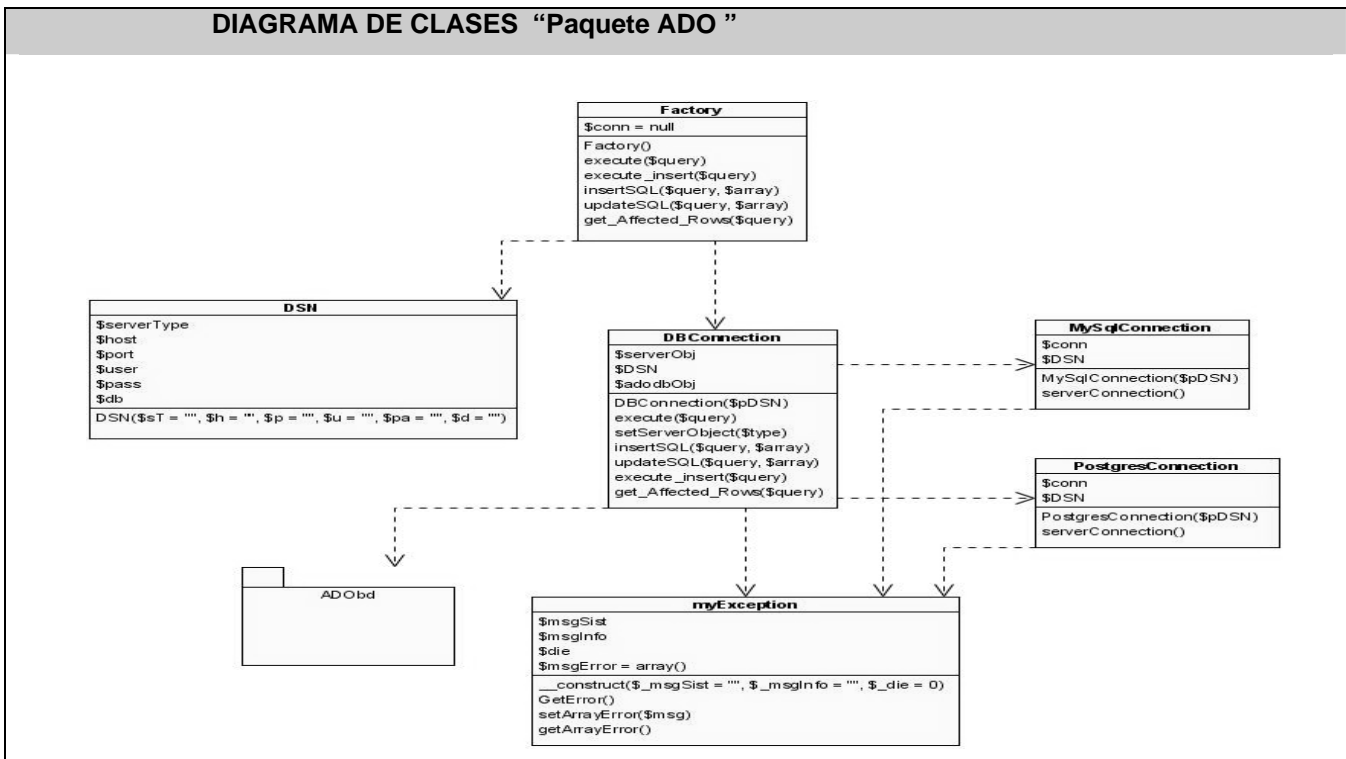


Figura IV.4 Diagrama de Clases Paquete Acceso a Datos

DIAGRAMA DE CLASES “Paquete de Clases Acceso a Datos Implementación”

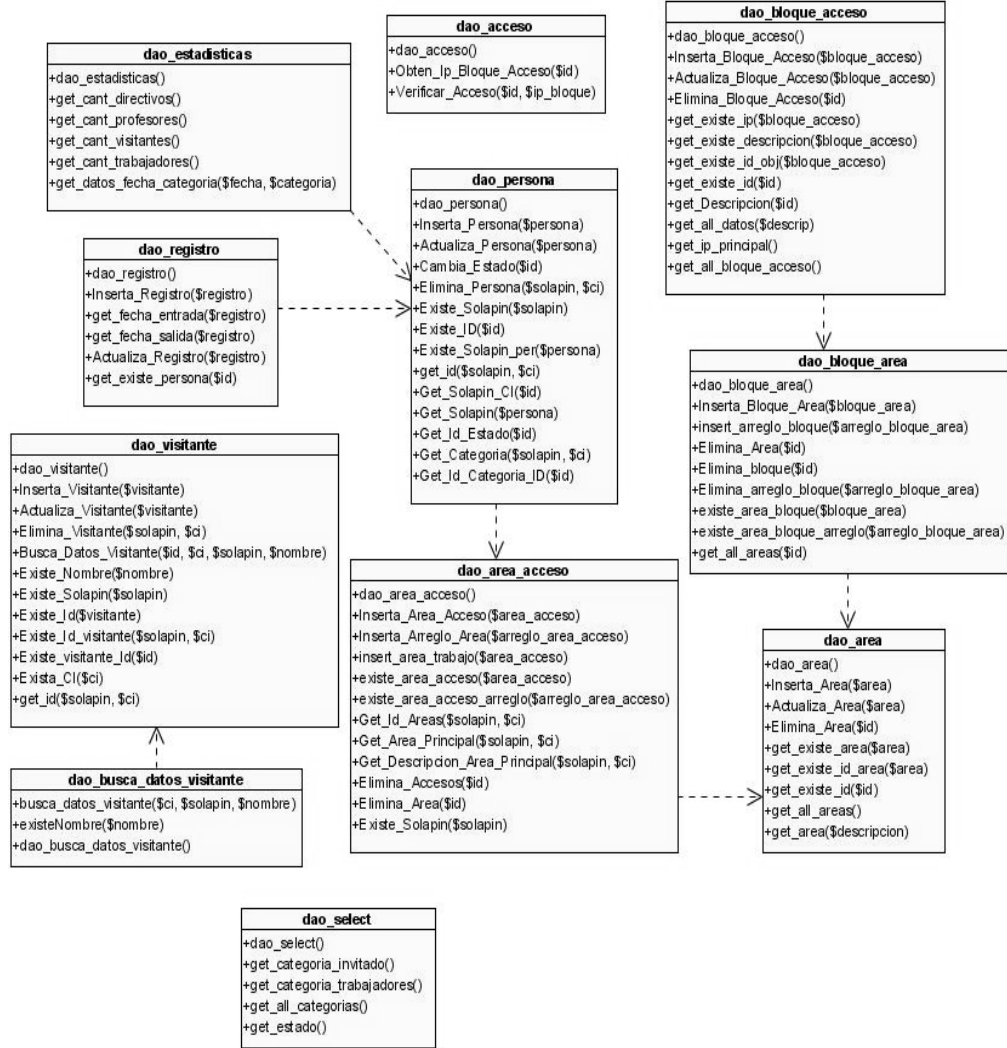


Figura IV.5 Diagrama de Clases Paquete Acceso a Datos Implementación

## **GLOSARIO DE TÉRMINOS**

**SOAP:** Es un protocolo elaborado para facilitar la llamada remota a funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web

**UML:** Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

**XML:** Es el acrónimo de eXtensible Markup Language (lenguaje de marcado extensible) desarrollado por el World Wide Web Consortium (W3C).

**CGI:** Common Gateway Interface (en inglés «Pasarela de Interfaz Común», abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa.

**ISAPI:** Internet Server API. Es una API para el servidor web IIS (Internet Information Server) de Microsoft. La ISAPI le permite a los programadores desarrollar aplicaciones basadas en Web que corran mucho más velozmente que los programas CGI convencionales porque están más íntimamente integrados con el servidor web. Además del IIS, varios servidores Web de otros fabricantes soportan la ISAPI

**ODBC:** Conectividad abierta de bases de datos (Open Database Connectivity). Una interfaz de programación de aplicaciones (API), que permite a las aplicaciones tener acceso a bases de datos multiplataforma desde diversas especificaciones estándar de orígenes de datos

**CASE:** **Acrónimo** inglés de Computer Aided Software Engineering, que viene a significar Ingeniería de Software Asistida por Ordenador.

**Passback:** Evento que ocurre cuando se detectan dos entradas o salidas consecutivas de una misma credencial.

---