



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

**DESARROLLO DE UN SISTEMA PARA
LA GESTIÓN DEL CONTROL DE
ACCESO EN LA UNIVERSIDAD DE LA
CIENCIAS INFORMÁTICAS**

**AUTORES: ELENA MAYDELIN ORTÍZ FERNÁNDEZ
GUILLERMO BLAYA CHIU**

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

**TUTOR: ING.ALIEN GONGORA RODRÍGUEZ
CO-TUTOR: ING.YANAY VIERA LORENZO**

Ciudad de la Habana, Junio, 2010

© 2010, Universidad de las Ciencias Informáticas

DECLARACIÓN DE AUTORÍA

Declaro que somos los únicos autores de este trabajo y autorizo a la Facultad 15 y a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con el mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del 2010.

Firma del Autor
Guillermo Blaya Chiu

Firma del Autor
Elena Maydelín Ortiz Fernández

Firma del Tutor
Alien Góngora Rodríguez

Firma del Co-Tutor
Yanay Viera Lorenzo

DATOS DE CONTACTO

Ing. Yanay Viera Lorenzo

Profesor Adiestrado

Graduado en el 2007 de Ingeniero en Ciencias Informáticas en la Universidad de Ciencias Informáticas (UCI), Título de Oro

Ha impartido asignaturas del departamento Ciencias Básicas, Matemática Discreta y Álgebra Lineal

Cumplimiento de misiones en Venezuela en tareas de un proyecto de software

E-mail: yviera@uci.cu

Ing. Alien Góngora Rodríguez

Instructor Recién Graduado

Graduado en el 2008 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI).

Ha impartido clases sobre las siguientes asignaturas; Programación II, Programación IV y varios cursos optativos.

Cumplimiento de misiones en Venezuela en tareas de un proyecto de software.

E-mail: agrodriguez@uci.cu

DEDICATORIA

A mi mamá por confiar en mí, y formar parte indispensable en mi vida. A mi abuela por el amor incondicional brindado. A mi tía titinga, que la adoro, a Edismel por ser un padre para mí, a mi tío Kiko por brindarme su ayuda cada vez que lo he necesitado, a mi hermanita Leydi, eres muy importante en mi vida, en fin a toda mi familia. A mi novio Joel por su confianza y el amor brindado en estos años. Te amo nene. A mi compañero de tesis y amigo Guillermo Blaya. A todas mis amistades de la UCI, las amistades de Mzlllo, gracias por apoyarme en mis locuras, los quiero a todos.

De Elena

Ya que este trabajo ha sido una de las cosas más importantes que he hecho en mi vida quiero dedicárselo a mis padres y a mi hermanito por cuidarme siempre, por ser mi guía, mi horizonte, mi límite ante los excesos, mis amigos y mis mejores consejeros y sobre todo por darme la oportunidad de ser su hijo. A toda mi familia y en especial a mi abuela por estar siempre a mi lado y depositar toda su confianza en mí. A mi amiga y compañera de tesis Elena por estar desde hace mucho tiempo guiándome, por hacer de mí una mejor persona y ser una amiga indispensable. A todas mis amistades, que aunque la gran mayoría las he conocido en estos cinco años y pronto nos distanciaremos se que las grandes amistades siempre perduran.

De Guille

AGRADECIMIENTOS

Quiero empezar agradeciendo a todas las personas que de una forma u otra han puesto su granito de arena para llegar donde estoy. A toda mi familia que siempre me han apoyado, a mi mamá, mami gracias por apoyarme siempre y confiar en mí, se que han sido dolorosos estos 5 años, pero vez tienes una hija ingeniera, a mi tía titinga que siempre me dijo "sal adelante que tu puedes", a mi tío Edismel que me apoyó como un padre, a mi tío Kiko, que siempre ha estado presente para lo q yo he necesitado, a todos mis primos Sergio y Reynier por ser tan buenos conmigo, a mi hermanita leydy que sabe q la quiero con la vida, a todas mis amistades de Mzlllo. A mi novio Joel por ser el mayor apoyo que he tenido, gracias por estos 3 años, 1 mes y 3 días llenos de felicidad, te amo nene. A todos mis compañeros de aula, de cuarto, a todos mis amigo, gracias por soportarme. A mi amiga Lidiarys por apoyarme y ayudarme siempre que he necesitado su ayuda, a su familia a Eliza y Benito por ser unos padres más para mí aquí en la habana. A mi compañero de tesis y amigo Guillermo, gracias por tu apoyo, eres el mejor. A mis tutores Alien y Yanay, gracias por su ayuda. Por último agradecer a la persona más importante en el mundo para mi, a mi abuelita querida Maribel, gracias abuelita por confiar en mí, por brindarme tu apoyo y amor incondicional, te quiero con la vida.

De Elena

Quiero agradecer por haber hecho posible que yo pudiera alcanzar este logro a mi mamá que no dejó de llamarme ni un solo día en estos cinco años; gracias por estar siempre a mi lado a pesar de la distancia, te quiero mucho. A mi papá que siempre estuvo pendiente de mi y ha sido un ejemplo a seguir; aunque no nos entendamos en ciertos puntos de vista sabes que te quiero. A mi hermanito por siempre preguntarme cuando estaría a su lado en la casa; eres una de las cosas que más quiero en este mundo, espero que sigas este ejemplo que estás viendo. A toda mi familia que siempre me recibió con mucho afecto y cariño cuando me tenían en sus brazos. A mi compañera de tesis y amiga Elena por dejarme estar a su lado y poder compartir este sueño entre los dos. A mis tutores Alien y Yanay; gracias por haber compartido sus conocimientos, han sido muy provechosos. A mis hermanos Yanesita y Arturo por estar de manera incondicional en estos nueve años de hermandad. A todas mis amistades pero en especial al Tike y Amaris; por preocuparse todo el tiempo por mi y compartir todos sus conocimientos y experiencias de la vida, a Alberton, Joel, Geovanis, Cristian, Kirenia y Lidiaris por brindarme siempre todo su apoyo y poder contar con ellos en todo momento. Y a todas las personas que de una forma u otra han contribuido en mi formación profesional.

De Guille

RESUMEN

En las instituciones donde existe gran afluencia de personal, es complejo mantener un control estricto de la entrada y salida del personal autorizado, trayendo como consecuencia la falta de seguridad. Por lo que se hace necesaria la creación de sistemas de control del acceso, para así limitar la entrada del personal a lugares de la institución con acceso restringido y llevar registros de los mismos.

En la Universidad de las Ciencias Informáticas el acceso se controla mediante la verificación manual de una identificación personal intransferible, sin embargo no existe un sistema de control del acceso que permita regular el acceso en la misma. Con el objetivo de mantener un control estricto del personal se automatizarán las actividades que componen el proceso de control de acceso.

El software que se desarrolle debe cumplir con las necesidades reales de los clientes y usuarios finales, por lo que este trabajo abarca el análisis, diseño, implementación y pruebas de la solución informática a desarrollar.

Para realizar estas funciones fue necesario identificar la metodología de desarrollo de software a utilizar y las herramientas necesarias para generar los artefactos correspondientes. Se hizo un estudio de algunas soluciones de software que contribuyeron a la definición de la solución. Finalmente, se obtuvo una solución de software que satisface las necesidades de los clientes y los usuarios finales.

PALABRAS CLAVE

Sistema de Control de Acceso, Solución Informática, Análisis, Diseño, Implementación, Metodología.

ÍNDICE

Índice de contenido

INTRODUCCIÓN	1
CAPÍTULO 1.....	4
FUNDAMENTACIÓN TEÓRICA	4
INTRODUCCIÓN	4
1.1 ORÍGENES DE LOS SISTEMAS DE CONTROL DE ACCESO.....	4
1.1.1 NECESIDAD DE LA EXISTENCIA DE ÁREAS DE ACCESO LIMITADO.....	5
1.2 TECNOLOGÍAS APLICADAS PARA EL CONTROL DE ACCESO EN EL MUNDO ACTUAL.....	6
1.2.1 TECNOLOGÍAS DE HARDWARE.....	7
1.3 PRINCIPALES SISTEMAS DE CONTROL DE ACCESO EN EL MUNDO.....	7
1.3.1 SISTEMA PARA EDIFICIOS INTELIGENTES.....	7
1.3.2 CONTROL DE ACCESO A UNIVERSIDADES.....	8
1.3.3 SISTEMA BIOACCESS.....	8
1.3.4 VALORACIÓN DE LOS SISTEMAS	9
1.4 PRINCIPALES SISTEMAS DE CONTROL DE ACCESO EN CUBA.....	9
1.4.1 SISTEMA DE SALUD PÚBLICA DE CUBA.....	9
1.5 PRINCIPALES SISTEMAS DE CONTROL DE ACCESO EN LA UCI.....	10
1.5.1 SISTEMA DE IDENTIFICACIÓN.....	10
1.5.2 SISTEMA DE CONTROL DE ACCESO A LOS COMEDORES.....	10
1.5.3 SISTEMA DE CONTROL DE ACCESO A LOS LABORATORIOS DE PRODUCCIÓN.....	11
1.5.4 VALORACIÓN DE LOS SISTEMAS.....	11
1.6 TECNOLOGÍA UTILIZADA EN LA UCI.....	12
1.7 TECNOLOGÍAS DE SOFTWARE UTILIZADAS EN EL DESARROLLO DEL SISTEMA.....	12
1.7.1 TECNOLOGÍA .NET.....	13
1.7.1.1 Lenguaje C Sharp.....	13
1.7.1.2 Visual Studio .Net 2005.....	14
1.7.1.3 Spring Framework .Net.....	15
1.7.2 EXTENSIBLE MARKUP LANGUAGE (XML).....	15

1.7.4 SISTEMA DE GESTIÓN DE BASES DE DATOS UTILIZADO	15
1.7.4.1 Lenguaje SQL	16
1.7.4.2 Gestor	16
1.7.4.1.1 PostgreSQL 8.3	16
1.7.4.1.2 DB4O	16
1.7.4.2 Manager	17
1.7.4.2.1 pgAdmin III 1.10	17
1.7.5 CONTROL DE VERSIONES	17
1.7.5.1 Servidor Subversion 1.4.5	18
1.7.5.2 Cliente Ankh 2.1.7756	18
1.7.6 HERRAMIENTAS DE MODELADO	19
1.7.6.1 Visual Paradigm for UML 6.4	19
1.7.6.2 Unified Modeling Language 2.1 (UML)	19
1.8 METODOLOGÍA A UTILIZAR	20
1.8.1 RATIONAL UNIFIED PROCESS (RUP)	20
CONCLUSIONES	21
CAPÍTULO 2.....	22
CARACTERÍSTICAS DEL SISTEMA	22
INTRODUCCIÓN	22
2.1 OBJETO DE AUTOMATIZACIÓN	22
2.2 PROPUESTA DEL SISTEMA	22
2.3 MODELO DE NEGOCIO	23
2.3.1 Definición de los actores del negocio	23
2.3.2 Definición de los trabajadores del negocio	23
2.3.3 Diagrama de Casos de Usos del Negocio	23
2.3.4 Diagrama de Actividades	24
2.3.5 Descripción de los casos de uso del negocio.....	26
2.4 ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA	30
2.4.1 Requisitos funcionales	30
2.4.2 Requisitos no funcionales	32
2.5 DEFINICIÓN DE LOS ACTORES DEL SISTEMA A AUTOMATIZAR	32

2.6. DISTRIBUCIÓN DE LOS CASOS DE USO POR PAQUETES	33
2.7 DESCRIPCIÓN EXPANDIDA DEL CASO DE ESTUDIO REGISTRAR ACCESO	37
CONCLUSIONES	38
CAPÍTULO 3.....	39
ANÁLISIS Y DISEÑO DEL SISTEMA.....	39
INTRODUCCIÓN	39
3.1 DIAGRAMAS DE CLASES DEL ANÁLISIS	39
3.2 DIAGRAMAS DE INTERACCIÓN	40
3.3 DISEÑO	41
3.3.1 Diagramas de clases del diseño.....	41
3.4 DISEÑO DE LA BASE DE DATOS.....	41
3.4.1 Diagrama de Clases Persistentes.....	42
3.4.2 Modelo de Datos	44
3.4.3 Descripción de las tablas	45
3.5 TRATAMIENTO DE ERRORES.....	46
3.6 ESTÁNDARES DE CODIFICACIÓN.....	47
CONCLUSIONES	48
CAPÍTULO 4.....	49
IMPLEMENTACIÓN Y PRUEBA.....	49
INTRODUCCIÓN	49
4.1 DIAGRAMA DE DESPLIEGUE	49
4.2 DIAGRAMAS DE COMPONENTES	50
4.3 ARQUITECTURA Y PATRONES UTILIZADOS	52
4.3.1 Arquitectura	52
4.3.2 Patrones utilizados	53
4.4 MODELOS DE PRUEBAS.....	54
CONCLUSIONES	60
CONCLUSIONES GENERALES	61
RECOMENDACIONES.....	62
REFERENCIAS BIBLIOGRÁFICAS	63
GLOSARIO DE TÉRMINOS	66

Índice de figuras

Figura 1 Diagrama de CU Negocio	23
Figura 2 Diagrama de Actividades Registrar Acceso	24
Figura 3 Diagrama de Actividades Registrar Vehículo	25
Figura 4 Diagrama de actividades Registrar Visita.....	26
Figura 5 Diagrama de CU del Sistema: Módulo Administración	35
Figura 6 Diagrama de CU del Sistema: Módulo Control de Acceso	35
Figura 7 Diagrama de CU del Sistema: Módulo Visitas.....	36
Figura 8 Diagrama de CU del Sistema: Módulo Seguridad	36
Figura 9 Diagrama de CU del Sistema: Módulo Configuración	36
Figura 10 Diagrama de CU del Sistema: Módulo Reportes	37
Figura 11 Diagrama de clases del análisis CU Registrar Acceso.....	39
Figura 12 Diagrama de Secuencia CU Registrar Acceso.....	40
Figura 13 Diagrama de Clases del Diseño del CU Registrar Acceso	41
Figura 14 Diagramas de Clases Persistentes.....	43
Figura 15 Modelo de Datos	44
Figura 16 Mensaje de error	46
Figura 17 Mensaje de notificación.....	47
Figura 18 Diagrama de despliegue	49
Figura 19 Diagrama de Componentes Módulo Administración.....	51
Figura 20 Arquitectura por capas	53

Índice de tablas

Tabla 1 Actores del Negocio	23
Tabla 2 Trabajadores del Negocio	23
Tabla 3 Descripción del CUN Registrar Acceso	27
Tabla 4 Descripción del CUN Registrar Vehículo	28
Tabla 5 Descripción del CUN Registrar Visita.....	30
Tabla 6 Actores del sistema.....	33
Tabla 7 Descripción resumida del CU Registrar Acceso.....	33
Tabla 8 Descripción del CU Registrar Acceso.....	38
Tabla 9 Descripción de la tabla tb_acceso.....	45
Tabla 10 Descripción de la tabla tb_dcliente.....	46
Tabla 11 CU Registrar Acceso a probar en el sistema.....	55
Tabla 12 Las no conformidades y la solución	59

Introducción

En las instituciones donde existe una gran afluencia de personal, es complicado mantener un control estricto de la entrada y salida del personal autorizado, trayendo como consecuencia la falta de seguridad. Por lo que se hace necesaria la creación de sistemas de control del acceso, para así limitar la entrada del personal a lugares de la institución con acceso restringido y llevar registros de los mismos.

Los sistemas de control del acceso, alcanzan gran popularidad por la seguridad que brindan a través del uso de una identificación que regula el acceso del personal a las instituciones. La eficiencia de un sistema de control del acceso está dada por el diseño de seguridad de la institución en el que fue basada su construcción. Para lograr un buen sistema es necesario que ese diseño responda a las necesidades de la institución. Un sistema de control de acceso fácil y rentable le proporciona a la institución la seguridad necesaria.

La creciente evolución informática en nuestro país, permite a varias instituciones nacionales la implantación de sistemas de control del acceso. La Universidad de las Ciencias Informáticas (UCI) es la mayor de las universidades del país, en extensión y en personal. Lo que exige a la universidad el uso de un sistema de control del acceso, para proteger la tecnología física existente y salvaguardar la información interna sensible.

En la UCI el acceso se controla mediante la verificación manual de una identificación personal e intransferible, sin embargo no existe un sistema de control del acceso que permita regular el acceso en la misma. Debido al gran flujo de personas que transitan por la Universidad, se hace necesario el uso de un sistema informático que verifique el acceso del personal.

En la actualidad, a la universidad accede personal ajeno sin la autorización correspondiente por lo que se carece de información precisa de quien accede a la misma, ni se puede identificar si las credenciales presentadas por el personal son válidas y si tienen acceso al área que solicitan. No se cuenta con una organización adecuada del proceso de solicitud de acceso del personal ajeno a la institución, este proceso es llevado de forma manual por lo que se hace complicado detectar anomalías a menos que se revise exhaustivamente los registros donde se encuentra almacenada dicha información que puede ser voluminosa y casi imposible de verificar. Además, los sistemas auxiliares que pueden ser utilizados para la

verificación del personal no están destinados para realizar dicha función de forma confiable, pues dichos sistemas no están concebidos con este fin.

Teniendo en cuenta esta problemática el trabajo de diploma concibe un sistema para darle solución al siguiente **problema**: ¿Cómo facilitar el control del acceso del personal en la Universidad de la Ciencias Informáticas?

El **objeto de estudio** se enmarca en los procesos de control del acceso y se define como **campo de acción** el proceso de control del acceso del personal de la Universidad de las Ciencias Informáticas.

Para dar solución al problema planteado se define como **objetivo general** del trabajo de diploma, desarrollar un sistema que permita gestionar el control de acceso en la Universidad de las Ciencias Informáticas.

Para darle cumplimiento al objetivo general del trabajo, quedan definidas las siguientes tareas de la investigación:

- Estudio de los sistemas de control de acceso a nivel mundial y de la UCI.
- Identificación de las herramientas que se utilizarán en el desarrollo del sistema de control de acceso.
- Modelado del negocio de los procesos de control de acceso en la UCI.
- Especificación de los requisitos de la solución de la gestión del control del acceso.
- Diseño de la solución de software para la gestión del control del acceso.
- Implementación de la solución para la gestión del control del acceso.
- Validación de la solución de software.

El presente trabajo de diploma está formado por cuatro capítulos:

Capítulo 1: Fundamentación Teórica.

Este capítulo contiene una base teórica para entender el problema a solucionar, conceptos fundamentales, técnicas, tendencias, metodologías y herramientas utilizadas.

Capítulo 2: Características del sistema.

En este capítulo se describen los procesos¹ necesarios a automatizar. Se hace una descripción del modelo de negocio y la descripción de los requerimientos funcionales y no funcionales.

Capítulo 3: Análisis y Diseño del sistema.

Se muestran los diagramas de clases del análisis y del diseño, con sus correspondientes diagramas de interacción por cada caso de uso del sistema. Se incluye el diseño de la base de datos, las definiciones de diseño y la forma en que se tratan los errores.

Capítulo 4: Implementación y Prueba.

En este capítulo se describe la implementación del sistema, se incluye el diagrama de componentes, el diagrama de despliegue, se definen los componentes a implementar, el modelo de implementación en el cual se describen las partes de la aplicación desde el punto de vista arquitectónico y se diseñan un conjunto de casos de pruebas para validar la solución.

¹**Proceso:** Es un conjunto de uno o más procedimientos o actividades directamente ligadas, que colectivamente realizan un objetivo del negocio normalmente dentro del contexto de una estructura organizacional que define roles funcionales y relaciones entre los mismos. (Workflow Management Coalition)

FUNDAMENTACIÓN TEÓRICA

Introducción

Para garantizar la seguridad de las instituciones se hizo necesaria la creación de los sistemas de control de acceso, los sistemas de control de acceso garantizan la seguridad absoluta a través de dispositivos destinados para tal fin, alarmas, sensores, cámaras de vigilancia, detectores de movimientos, infrarrojos, entre otros.

En este capítulo se realiza una investigación sobre los antecedentes de los sistemas de control de acceso, así como algunos de los principales sistemas a nivel internacional y nacional. Se describen las principales características de las herramientas y frameworks² a utilizar y se explica cómo se concibe el desarrollo la solución.

1.1 Orígenes de los Sistemas de Control de Acceso.

Desde los orígenes de la humanidad se ha tratado de controlar el acceso de las personas a lugares restringidos. El acelerado desarrollo de la informática y las tecnologías trajo consigo una notable evolución en los sistemas del control de acceso. En un mundo desarrollado, es de suma importancia proteger el acceso a los recursos y evitar una mala manipulación de los mismos, que podría causar pérdidas considerables. Surgiendo problemas, los cuales llevan consigo dos conceptos que muy a menudo se

²**Frameworks:** Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

mezclan de manera difusa: la autenticación (pretende establecer quién eres) y la autorización (establece que podemos hacer). Estos conceptos parecen ir ligados de forma indisoluble pero no siempre ocurre de esta manera, por lo que hay que tener claro estos conceptos antes mencionados, quiénes somos, qué puedo hacer para evitar catástrofes. Para impedir pérdida de recursos, los sistemas de control tienen soluciones las cuales llevan consigo la utilización de los tipos básicos de control de acceso, que tienen filosofías diametralmente opuestas.(Historia de los sistemas de control de acceso).

1.1.1 Necesidad de la existencia de áreas de acceso limitado.

Con el de cursar de los años, la humanidad ha adquirido y modificado una cultura de seguridad de sus bienes que ha ido evolucionando desde las pertenencias ligadas directamente con sus necesidades fisiológicas básicas hasta adquirir conciencia de pertenencia social y proteger información e instituciones, que están relacionadas con necesidades de orden superior, como necesidades de aceptación social, de autoestima y de autorrealización.

Establecer recomendaciones y lineamientos generales de seguridad encaminadas a la creación y fomento de una cultura de seguridad preventiva³ orientados a la protección de los recursos humanos, información, bienes, muebles e inmuebles de los Centros Públicos, constituye una buena práctica orientada a la preservación de los bienes y al desarrollo.

Las principales amenazas podemos clasificarlas en dos grandes grupos, las provocadas por situaciones internas y las provocadas por situaciones externas, ambas tienen características equivalentes que se mencionan a continuación:

- Robo de documentos, mercancías, equipos, efectivo, valores, dependiendo de la naturaleza del organismo, cometido por:
 - Personal laboral.
 - Proveedores.
 - Clientes.

³**Seguridad Preventiva:** Es la protección de la vida, integridad física, bienes y posesiones, a través de un análisis de riesgos, acciones y medidas precautorias pertinentes, que tienen como finalidad la reducción de contingencias o eventualidades en el mayor grado posible ante hechos de carácter natural o de conductas mal intencionadas o antisociales.

- Visitantes.
- Personas ajenas al organismo.
- Espionaje
- Sabotaje.
- Actividades no autorizadas.
- Violación y vulnerabilidad de alamas y seguridad.
- Vulnerabilidad en el control de llaves y claves.
- Vulnerabilidad del personal.
- Control de la información.
- Vulnerabilidad informática.

Existen además otras amenazas provocadas por las situaciones externas, ellas son:

- Violación de la seguridad perimetral.
- Terrorismo.
- Ataque.
- Desorden público.

Con el propósito de brindar la mayor seguridad posible a las instalaciones y potenciar el adecuado funcionamiento de organizaciones a lo largo de la historia encontramos la aparición de áreas (instituciones, centros, compañías o empresas) de limitado acceso. Garantizar sólo el acceso del personal autorizado de dichas áreas es una medida indispensable para la seguridad y el adecuado funcionamiento de las mismas.

1.2 Tecnologías aplicadas para el control de acceso en el mundo actual.

En la actualidad el desarrollo de las tecnologías de hardware y software ha posibilitado brindar soluciones efectivas tanto para grandes empresas, que requieren máxima seguridad y robustez según sus características, como para empresas de mediana o menor escala que requieren precios económicos y facilidad de uso. Un software de control de acceso correctamente desarrollado permitirá llevar un buen

control de horarios, ingreso y egresos de personas, que garantice además la rapidez, fiabilidad, accesibilidad, seguridad y eficiencia.

1.2.1 Tecnologías de Hardware.

Después de haber hecho un análisis de las tecnologías de hardware que se pueden utilizar en el desarrollo de los sistemas de control de acceso, se decide utilizar la tecnología de las tarjetas de código de barra por las funcionalidades que mencionan a continuación y por ser la tecnología de menor costo.

Tarjetas de Código Barra: es una tecnología de identificación automática. Un código de barras consiste en una serie de barras adyacentes, paralelas y separadas por pequeños espacios, que permite recoger datos con precisión y rapidez. Los diseños predeterminados de anchura son usados para codificar datos en el código. Para leer información en un símbolo de código de barras, un dispositivo de lectura se desliza a través del símbolo de un lado a otro, mediante dicha operación, la anchura de barras y los espacios son analizados por el decodificador del lector, recuperándose así los datos originales. (TrajeSol, 2005).

1.3 Principales Sistemas de Control de Acceso en el mundo.

En el mundo existen numerosos sistemas relacionados con el control de acceso, pues se han convertido en una necesidad. Dentro de los más reconocidos se pueden mencionar los siguientes:

1.3.1 Sistema para Edificios Inteligentes.

Dorlet⁴ ofrece un **sistema de control de acceso para los edificios**, ahora llamados edificios inteligentes, esta empresa hizo la integración de algunos sistemas de seguridad como los de incendios o los circuitos cerrados de televisión, que son monopolio de los japoneses que estaban ya suficientemente cubiertos, al realizar esta integración además de otros aspectos este sistema es capaz de cuando una persona entra en un edificio inteligente, todo el sistema de seguridad central se pone a funcionar para evitar el acceso a zonas prohibidas, robos y accidentes y controlar al milímetro los movimientos de esa persona.(Dorlet).

⁴**Dorlet:** La empresa guipuzcoana que se dedica a desarrollar tecnología de control de acceso a edificios.

1.3.2 Control de Acceso a Universidades.

Kimaldi⁵, ofrece un **sistema de control de acceso a Universidades**. El sistema incorpora un potente programa que, abarca todo el control específico de todos los tipos de registros para las entradas y/o salidas, gestiona el acceso por régimen jerárquico, por franjas horarias dentro de un calendario anual, el registro del histórico de los accesos no autorizados y los intentos de uso fraudulento. La actualización de la base de datos no exige conocimientos técnicos específicos, por tanto esta actividad resulta ágil y de fácil manejo. Aplicación múltiple de control de accesos y alarmas mediante radiofrecuencia. El sistema de seguridad y control visual son proporcionados por los mismos terminales de acceso que son capaces de activar puntualmente las cámaras. El control de alarmas de las puertas de salida de emergencia del edificio principal, permiten visualizar por pantalla desde el centro de gestión cualquier incidencia que se produzca en cualquier salida de emergencia, así como el estado en que puede quedar (abierta o cerrada), y todos los registros de incidencias.(Kimaldi, 2007).

Los productos que se utilizan para esto son:

- Tarjeta KimaldiNdcan Max
- Lector proximidad Kimaldi RD125K
- Concentrador Kimaldi KMD905 v2

1.3.3 Sistema bioAccess.

Es un sistema de control de acceso de personas que utiliza una fuente de identificación confiable basada en la lectura de huellas digitales, diseñado para controlar y administrar las entradas, salidas y movimientos dentro de la institución. Es confiable, rápido y práctico, mejora el control de personas mediante identificación biométrica y está especialmente diseñado para una moderna y segura administración. Posee una efectividad comprobada a 30.000 huellas por segundo, con conexión con cualquier tipo de dispositivo de apertura y es fácilmente integrable a sistemas existentes.(Universal Control & Security, 2003).

⁵**Kimaldi:** Kimaldi Electronics es una empresa dedicada a la fabricación, diseño y comercialización de productos dentro del mundo de la Identificación de personas.

1.3.4 Valoración de los sistemas

Todos estos sistemas de control de acceso no se pueden utilizar pues están desarrollados única y exclusivamente para instalaciones específicas, además la eficiencia de un sistema de control del acceso está dada por el diseño de seguridad de la institución en el que fue basada su construcción. Utilizan tecnología de punta, lo cual implica grandes inversiones económicas, y la UCI no puede hacer estas inversiones, además de que sus funcionamientos no se basan en la tecnología de código de barras, principal características del sistema a desarrollar.

1.4 Principales Sistemas de Control de Acceso en Cuba.

1.4.1 Sistema de Salud Pública de Cuba.

El Sistema de Salud Pública de Cuba, en respuesta a la necesidad de contar con soporte técnico adecuado y moderno en cuanto a tecnología, para el manejo de informaciones, cuenta con un **Registro Electrónico de Pacientes**. El mismo, tiene un alcance amplio en cuanto al registro de datos, y en términos de seguridad posee implementado varios módulos:

Control del acceso: Con palabras claves (password) seguras o aún mejor con sistemas de control de acceso del habla o biométrico que estén relacionados con un sistema encargado de autenticar el personal de salud, el cual clasificaría a los usuarios acorde a su autorización a acceder a determinada información y a desarrollar ciertas funciones (privilegios médicos y otras).

Firmas electrónicas: Implementación de un sistema que permita a los originadores (personal de salud o su instrumento) añadir firmas electrónicas para las entradas y detectar si alguna entrada ha sido alterada.

Integridad de los datos: Después del proceso de edición ninguna información debe perderse o alterarse en ningún sentido, las correcciones son hechas como mejoras.

Auditorías: Control del acceso de los usuarios a un registro y de las ediciones hechas a ese registro, años atrás esto era considerado una demanda excesiva, ya que la información que se gestiona crecer voluminosamente. Sin embargo, esta consideración ha perdido su valor debido al bajo costo del almacenaje en las computadoras.

Este sistema, además de ser para propósitos específicos, resulta de gran dificultad para su implantación tanto que, de acuerdo a las facilidades que brinda y las necesidades detectadas, se han definido 5 niveles para su aplicación satisfactoria. En estos momentos Cuba no tiene condiciones económicas para alcanzar el nivel 5, sin embargo se ha llegado a un estado intermedio entre los niveles 4 y 5 que garantizarán alcanzar un nivel significativo de computarización de la información de las historias clínicas, con muy pequeñas inversiones. (Registro Electrónico de Pacientes., 1999).

El sistema a desarrollar al igual que la aplicación antes mencionada está estructurado por módulos, pero éste maneja información específica para las entidades del ministerio de salud pública por lo que no sería factible para la UCI.

1.5 Principales Sistemas de Control de Acceso en la UCI.

Dentro de los sistemas automatizados relacionados con el tema de estudio que se presenta, los más conocidos e importantes son:

1.5.1 Sistema de Identificación.

Este sistema brinda un servicio de certificación de identidad a otros sistemas informáticos, como los que son para el control del acceso. Tiene almacenados los datos de todo el personal que labora y estudia en la Universidad: estudiantes y todo tipo de trabajadores. Lo más importante es que le asigna a cada persona un código único, para su identificación.

Este sistema está estructurado por los siguientes módulos; administración, configuración, identificación, detección de rostros y seguridad. Además utiliza frameworks tales como SpringFramework .Net, posee una arquitectura por capas, dichas características hacen que dicha aplicación sea reutilizable, lo que brinda la facilidad de utilizar módulos tales como seguridad y configuración en la aplicación a desarrollar.

1.5.2 Sistema de Control de Acceso a los Comedores.

Mediante este sistema se controla en los comedores de los diferentes Complejos Alimenticios el acceso de los estudiantes, profesores y trabajadores durante las tres sesiones de servicio: desayuno, almuerzo y comida. El mismo se divide en dos partes: el control de acceso y la gestión de comensales. El acceso se

controla registrando el código de barras, que se encuentra en la identificación de cada persona, en cada una de las puertas de los comedores. La gestión de comensales permite a los directivos la asignación de los comedores y puertas a los mismos, además de ofrecer reportes como cantidad de comensales que han pasado y desglosarlo por puerta o por tipo.

Esta aplicación, a pesar de incorporar tecnologías para el acceso a datos como DB4O, tecnología a utilizar en el sistema que se desea desarrollar, no es de gran utilidad debido a que la misma maneja información enfocada a la gestión del proceso logístico alimenticio, información que no es relevante para el sistema a desarrollar.

1.5.3 Sistema de Control de Acceso a los Laboratorios de Producción.

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

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

1.5.4 Valoración de los sistemas.

Después de haber hecho un estudio de los sistemas de control de acceso comentados, se han logrado identificar algunas de las funcionalidades y características más comunes de estos tipos de software, como gestión de puertas, control de horarios y permisos, manejo de registro, recuperación de fichajes, identificación y clasificación del personal, creación de credenciales. Estos sistemas tienen en común algunas de las características principales que debería tener el sistema a desarrollar, interfaz amigable y sencilla, fácil utilización y mantenimiento de la base de datos, estas dos características son de gran importancia pues el facilitaría al usuario la interacción con el sistema, formas de trabajo on-line y off-line,

para evitar que cuando no estén conectados ocurra pérdida de la información. Coincide con el dispositivo de control de acceso a utilizar, lector de código de barras.

Todas estas funcionalidades y características de los sistemas analizados se van a desarrollar en el sistema, no se van a utilizar ninguno de estos sistemas pues se realizaron con un objetivo específico, no son sistemas generales que se puedan implantar con otro objetivo, que no sea para el cual se desarrolló.

1.6 Tecnología utilizada en la UCI.

- **Lector de barras Voyager**

En la UCI se decidió utilizar el código de barras Código 3 de 9 o Código 39, que es el más utilizado, pues codifica todos los caracteres ASCII (números, letras, símbolos). Puede ser variable en longitud (permitir cualquier número de dígitos) y ofrece gran seguridad a la lectura. Es de tipo discreto (cada carácter es representado por un grupo específico de barras y espacios) y es usado en aplicaciones de mensajería y paquetería, estados de cuenta, números de servicios y credenciales para control de acceso.

El Voyager 9520 se ha concebido como un lector muy agresivo con una mayor profundidad de campo y una velocidad de lectura que es casi el doble de la serie MS 900 y combina las prestaciones del MS 951 y el MS 961 dentro de un mismo escáner para ofrecer mayor flexibilidad al usuario. Por supuesto este producto mantiene el exclusivo sensor infrarrojo patentado por Metrologic y un sistema de control que permite una activación totalmente automática. (IDENTIFIC-AR, 2006).

El sistema a desarrollar utilizará este dispositivo para el registro de los accesos ya sean de entrada o de salida, pues el mismo presenta las funcionalidades necesarias para que se integre con la aplicación y es la solución más económica.

1.7 Tecnologías de software utilizadas en el desarrollo del sistema.

Teniendo en cuenta las políticas de la Dirección de Informatización para el desarrollo de aplicaciones de escritorio se utilizó como lenguaje de programación C#, el entorno de desarrollo integrado Visual Studio .Net 2005, gestor de base de datos relacional PostgreSQL 8.3, gestor de base de datos de objetos DB4O en su variante embebido, framework arquitectónico SpringFramework .Net, UML para el modelado y RUP como metodología para el desarrollo de la aplicación.

1.7.1 Tecnología .Net

El Framework de .NET es un entorno de desarrollo distribuido, en el cual se reúnen en conjunto lenguajes y servicios que simplifican el desarrollo y ejecución de aplicaciones .NET. Soporta múltiples lenguajes de programación, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes: C# (C Sharp), Visual Basic, C++, J# (Java #), Jscript. Es además, un entorno abierto, adaptable o receptivo a nuevos lenguajes de programación y tecnologías.

La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar a la funcionalidad que los Servicios Web proveen.

Es multiplataforma, debido a que posee el CLR⁶ (Common Language Runtime, es decir, el Motor Común de Ejecución). Un programa .NET podrá ser compilado y ejecutado en cualquier plataforma que incluya un CLR.(Microsoft Corporation, 2010).

1.7.1.1 Lenguaje C Sharp

C# es un lenguaje de programación orientado a objetos, desarrollado como parte de la plataforma .NET. Incluye mejoras derivadas de otros lenguajes como C y C++. Utiliza muchas de las características de C++ en las áreas de instrucciones, expresiones, operadores y fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

C# presenta capacidad de control de versiones para facilitar la administración y la implementación de aplicaciones, plena interoperabilidad por medio de los servicios de COM+ 1.0 y .NET Framework con un acceso limitado basado en bibliotecas y compatibilidad con XML para la interacción con componentes basados en tecnología Web. El lenguaje proporciona la capacidad de generar componentes de sistema duraderos debido a que tiene total compatibilidad entre el modelo de componentes (COM) y la plataforma para la integración de código existente. Además de presentar gran robustez y seguridad en el tratamiento

⁶**CLR:** Es el centro neurálgico del Framework de .NET encargado de gestionar la ejecución de las aplicaciones, aplicar parámetros de seguridad y ejecutar el denominado recolector de basuras.(Microsoft Corporation, 2010)

de tipos. Posee plena compatibilidad con conceptos de metadatos extensibles y es posible interactuar con otros lenguajes, entre plataformas distintas, y con datos heredados.

C# es un lenguaje potente, flexible, moderno y seguro. Sin embargo el mayor sentido práctico de este lenguaje se encuentra en Visual Studio .NET debido a la estrecha integración entre dicho entorno y C#. Gracias a esta integración, los diseñadores de Visual Studio .NET son capaces de generar código C# al crear interfaces de usuario basadas en formularios Windows y formularios Web, producir adaptadores de datos, guiones embebidos en documentos ASP.NET. (Microsoft Corporation, 2010).

1.7.1.2 Visual Studio .Net 2005

Visual Studio .NET es la herramienta que Microsoft ha publicado con el fin de crear aplicaciones para la plataforma .NET desde una interfaz visual basada en ventanas. Este software tiene un rasgo particular al constituir un solo ambiente de desarrollo de aplicaciones para todos los lenguajes. Visual Studio .NET soporta los nuevos lenguajes .NET: C#, Visual Basic .NET y Manager C++, además de C++. Visual Studio .NET puede utilizarse para construir aplicaciones dirigidas a Windows (utilizando Windows Forms), Web (usando ASP.NET y Servicios Web) y dispositivos portátiles (utilizando .NETCompact Framework).(Microsoft Corporation, 2010).

Estas son algunas de las principales características del Visual Studio .NET y sus ventajas:

- Lista de Tareas: organiza las tareas y administra los errores y las advertencias en un solo lugar. Las tareas se leen de los comentarios especializados del código fuente y se presentan en un formato tabular, donde se le hace un *doble-clic* para saltar a la sección del código fuente donde está el error o advertencia.
- Explorador de la Solución: brinda una vista jerárquica de una solución dividida en proyectos. Permite la administración de los proyectos involucrados en una solución.
- Explorador de Servidores: administra su computador y otros que están en red, incluyendo recursos como el Servidor de SQL, colas de mensajes, servicios.
- Multi-Monitor: permite un mejor uso del espacio disponible en la pantalla.
- Ayuda Dinámica: hace referencias a la documentación de la ayuda disponible de acuerdo a la materia sobre la cual se esté trabajando.

1.7.1.3 Spring Framework .Net

En el sistema de control de acceso se va a utilizar este framework pues es un marco de aplicaciones de código abierto que permite la construcción de aplicaciones empresariales en .Net. Proporciona componentes basados en patrones de diseño probando que puede ser integrado en todos los niveles de la arquitectura de una aplicación. SpringFramework .Net ayuda a aumentar la productividad en el desarrollo y mejorar la calidad de las aplicaciones y el rendimiento.

Su principal novedad está en la inversión de control o inyección de dependencia que permite independizar las diferentes capas de una aplicación.(Spring Source Community, 2010).

1.7.2 Extensible Markup Language (XML)

XML es un estándar para el intercambio de información estructurada entre diferentes plataformas. Es una tecnología sencilla que tiene a otras tecnologías que la complementan y la hacen mucho más extensible y compatible. Además, permite al programador dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de datos o el recorrido de las estructuras corren a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

Esta tecnología será utilizada en el sistema para gestionar la configuración inicial de la aplicación para la inversión de control⁷ y para la carga de los datos.(W3C, 2003).

1.7.4 Sistema de gestión de Bases de Datos utilizado

Los sistemas de gestión de bases de datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Generalmente los SGBD manejan de manera sencilla y ordena un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

⁷ **Inversión de control:** (**IoC** del inglés *Inversion of Control*) es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos o funciones.

1.7.4.1 Lenguaje SQL

SQL (Lenguaje de Consulta Estructurado, o Structured Query Language) es un lenguaje declarativo y estándar de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos, declarativo de alto nivel por su fuerte base teórica y su orientación al manejo de conjuntos de registros permite una alta productividad en codificación. Hablamos por tanto de un lenguaje normalizado que nos permite trabajar con cualquier tipo de lenguaje en combinación con cualquier tipo de base de datos (MS Access, SQLServer, MySQL, PostgreSQL, SAP Database, Firebird, entre otros). En el sistema se va a utilizar PostgreSQL. El hecho de que sea estándar no quiere decir que sea idéntico para cada base de datos.

En efecto, determinadas bases de datos implementan funciones específicas. Una de sus principales virtudes es su sencillez de uso y versatilidad ya que se puede manejar tanto con unas pocas instrucciones como con estructuras mucho más complejas.(OmegaSphere, 2005).

1.7.4.2 Gestor.

1.7.4.1.1 PostgreSQL 8.3

Es un Sistema Gestor de Base Datos relacional de software libre y orientado a objetos, liberado bajo la licencia BSD. Posee las características de los más potentes sistemas comerciales como Oracle o SQL Server, con la ventaja de que su licencia es gratuita. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales. Dicha comunidad es denominada PostgreSQL Global Development Group (PGDG). Tiene más de 15 años de desarrollo activo y se ha ganado la reputación de ser confiable y mantener la integridad de los datos. Se ejecuta en la mayoría de los sistemas operativos más utilizados en el mundo como Linux, varias versiones de UNIX y Windows.

Este sistema gestor de base datos entre sus ventajas principales se encuentran; alta concurrencia, amplia variedad de tipos nativos, multiplataforma, extensible, estabilidad y confiabilidad e instalación ilimitada.

1.7.4.1.2 DB4O

Es un novedoso motor de base de datos orientado a objetos. Sus siglas se corresponden con la expresión "DataBase4 (for) Objects", que a su vez es el nombre de la compañía que lo desarrolla: *db4objects, Inc.*

Las claves innovadoras de este producto son su alto rendimiento (sobre todo en modo embebido) y el modelo de desarrollo que proporciona a las aplicaciones para su capa de acceso a datos, el cual propugna un abandono completo del paradigma relacional de las bases de datos tradicionales.

La mayor clave del éxito que está teniendo este motor de base de datos frente a otros competidores que han desarrollado tecnologías similares, es que se ha optado por un modelo de licenciamiento idéntico al utilizado por empresas como MySQL: licencia dual GPL/comercial. Es decir, si se quiere desarrollar software libre con esta librería, su uso no conlleva ningún coste por licencia; sin embargo si se desea aplicar a un software privativo, se aplica otro modelo de licenciamiento concreto. (VERSANT CORP, 2010).

1.7.4.2 Manager

1.7.4.2.1 pgAdmin III 1.10

Herramienta muy popular y completa de código abierto para administración y desarrollo en la plataforma PostgreSQL, la más avanzada base de datos de código abierto en el mundo. Puede ser usado en Linux, FreeBSD, Solaris, Mac OSX y Windows para administrar PostgreSQL8.3 y anteriores, así como comerciales y las versiones derivadas de PostgreSQL como Postgres Plus Advanced Server y Greenplumdatabase.

Está diseñado para responder a las necesidades de todos los usuarios, de la escritura simples consultas SQL hasta para el desarrollo de bases de datos complejas. La interfaz gráfica soporta todas las características y hace fácil administración. Es desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo y está disponible en más de una docena de idiomas. Es un software libre publicado bajo la licencia BSD⁸.(pgAdmin PostgreSQL Tools, 2010).

1.7.5 Control de versiones

Una versión, revisión o edición de un producto, es el estado en el que se encuentra en un momento dado en su desarrollo o modificación. Se llama **control de versiones** a la gestión de los diversos cambios que

⁸**Licencia BSD:** licencia de software libre permisiva, permite el uso del código fuente en software no libre.

se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico).

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente. Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web. (Wingerd, 2005).

1.7.5.1 Servidor Subversion 1.4.5

Subversion es un software de control de versiones diseñado específicamente para reemplazar al popular CVS⁹, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como SVN por ser ese el nombre de la herramienta de línea de comandos. Una característica importante es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. (CollabNet, 2010).

1.7.5.2 Cliente Ankh 2.1.7756

Ankh es un cliente para el control de versiones para el Visual Studio. El software le permite realizar las operaciones de control más comunes directamente desde el interior de la versión de Microsoft Visual Studio IDE. Con AnkhSVN que ya no necesita salir de su IDE para realizar tareas como ver el estado de su código fuente, la actualización de su copia de trabajo y confirmar los cambios. Incluso puedes buscar en tu repositorio y usted puede conectar su herramienta de diferencias favorita.

AnkhSVN es un proyecto de código abierto con. La última versión estable es 2.1.7756. (CollabNet, 2010).

⁹**CVS** (del inglés Concurrent Versions System): es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren. (March Hare Pty Ltd & CVSNT Project, 2004).

A pesar de que el equipo de desarrollo de la aplicación a desarrollarse es pequeño se hace necesario la utilización de una herramienta de control de versiones, pues la finalidad de las mismas en el caso en cuestión, es para un respaldo tanto de la aplicación como los demás artefactos que se van generando durante el ciclo vida de la aplicación en el caso de que exista alguna pérdida de datos.

1.7.6 Herramientas de modelado.

1.7.6.1 Visual Paradigm for UML 6.4

Visual Paradigm para UML es una herramienta de modelado profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Esta herramienta a pesar de todas las características antes mencionadas es factible para el desarrollo de la aplicación puesto que la UCI adquirió una licencia de la misma.

1.7.6.2 Unified Modeling Language 2.1 (UML)

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

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

1.8 Metodología a utilizar

1.8.1 Rational Unified Process (RUP)

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El Proceso Unificado utiliza el *Lenguaje Unificado de Modelado* (UML del inglés Unified Modeling Language), para preparar todos los esquemas de un sistema software. De hecho, UML, es una parte esencial del Proceso Unificado sus desarrollos fueron paralelos.(BOOCH, y otros, 2004).

No obstante, los verdaderos aspectos que definen el ciclo de vida del Proceso Unificado de Desarrollo se resumen en tres características fundamentales:

- Dirigido por casos de uso: Los casos de uso reflejan lo que desean y necesitan los usuarios futuros, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí, los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo, describe los elementos del modelo que son importantes para su construcción y los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- Iterativo e incremental: RUP propone que cada fase se desarrolle en iteraciones, estas irán creciendo incrementalmente en cada iteración puesto que se realizarán actividades dentro de los flujos de trabajo que antes no fueron realizadas. Es práctico dividir el trabajo en partes más pequeñas o mini-proyectos. Cada mini-proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del

producto. Las iteraciones deben estar controladas; deben seleccionarse y ejecutarse de una forma planificada.

Conclusiones

En este capítulo se hizo un breve análisis de los inicios de los sistemas de control de acceso, por qué se hizo necesario su surgimiento y qué problemas resolverían. Se analizaron los sistemas de control del acceso, en el ámbito internacional, los cuales no los podemos utilizar pues incorporan tecnología de punta y no se tiene acceso a su documentación, de los sistemas nacionales se analizaron los que están en la UCI y de todos los analizados se decidió reutilizar del sistema de identificación, los módulos de configuración y seguridad y el framework Sprint Framework.Net, estos sistemas todos coinciden con la tecnología del lector de código barras, por ser este el que se utiliza en la universidad. Además, se identificaron la metodología y las herramientas a utilizar para el desarrollo del sistema.

Características del sistema

Introducción

En este capítulo se realiza el Modelo de Negocio. Se detalla la propuesta de sistema así como los requerimientos que debe cumplir el sistema. Estos requerimientos permiten obtener una concepción general del sistema. Y se presenta una propuesta del sistema.

2.1 Objeto de Automatización

El objeto de automatización es el control del personal que accede a la UCI así como a las diferentes áreas dentro de la Universidad. Con la automatización, se persigue tener un mejor y mayor control del personal debido a la gran afluencia que existe actualmente.

2.2 Propuesta del sistema

El sistema que se desarrollará tiene como objetivo servir de herramienta para controlar el acceso a la UCI y a sus diferentes áreas.

Con el fin de cumplir los objetivos del trabajo, el sistema debe poseer dos niveles de acceso: el administrador del sistema que sería el usuario encargado de gestionar toda la información referente a los módulos de Seguridad, Administración y Configuración; y el agente de seguridad que se traduce en el usuario que gestiona la información concerniente a los módulos de Visita y Control de Acceso.

2.3 Modelo de Negocio

2.3.1 Definición de los actores del negocio

Actor	Justificación
Persona	Es un trabajador, profesor o estudiante de la UCI que solicita la entrada a la misma, así como a sus diferentes áreas.
Visitante	Es una persona ajena a la UCI que solicita la entrada a la misma.

Tabla 1 Actores del Negocio

2.3.2 Definición de los trabajadores del negocio

Trabajador	Justificación
Agente de seguridad	Es el encargado de validar la entrada a la UCI o sus áreas

Tabla 2 Trabajadores del Negocio

2.3.3 Diagrama de Casos de Usos del Negocio

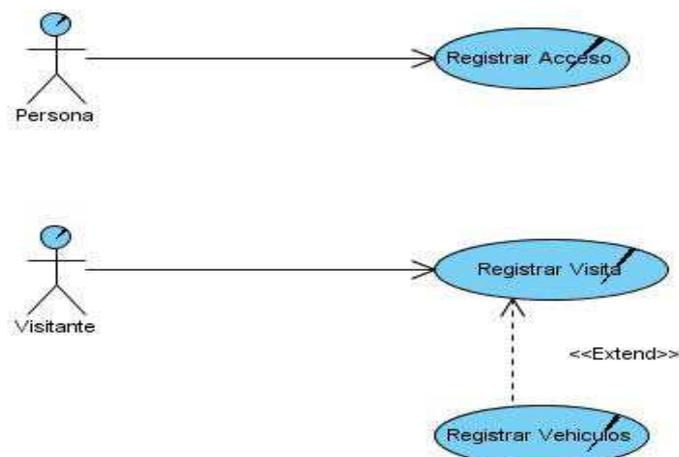


Figura 1 Diagrama de CU Negocio

2.3.4 Diagrama de Actividades

- Caso de Uso Registrar Acceso

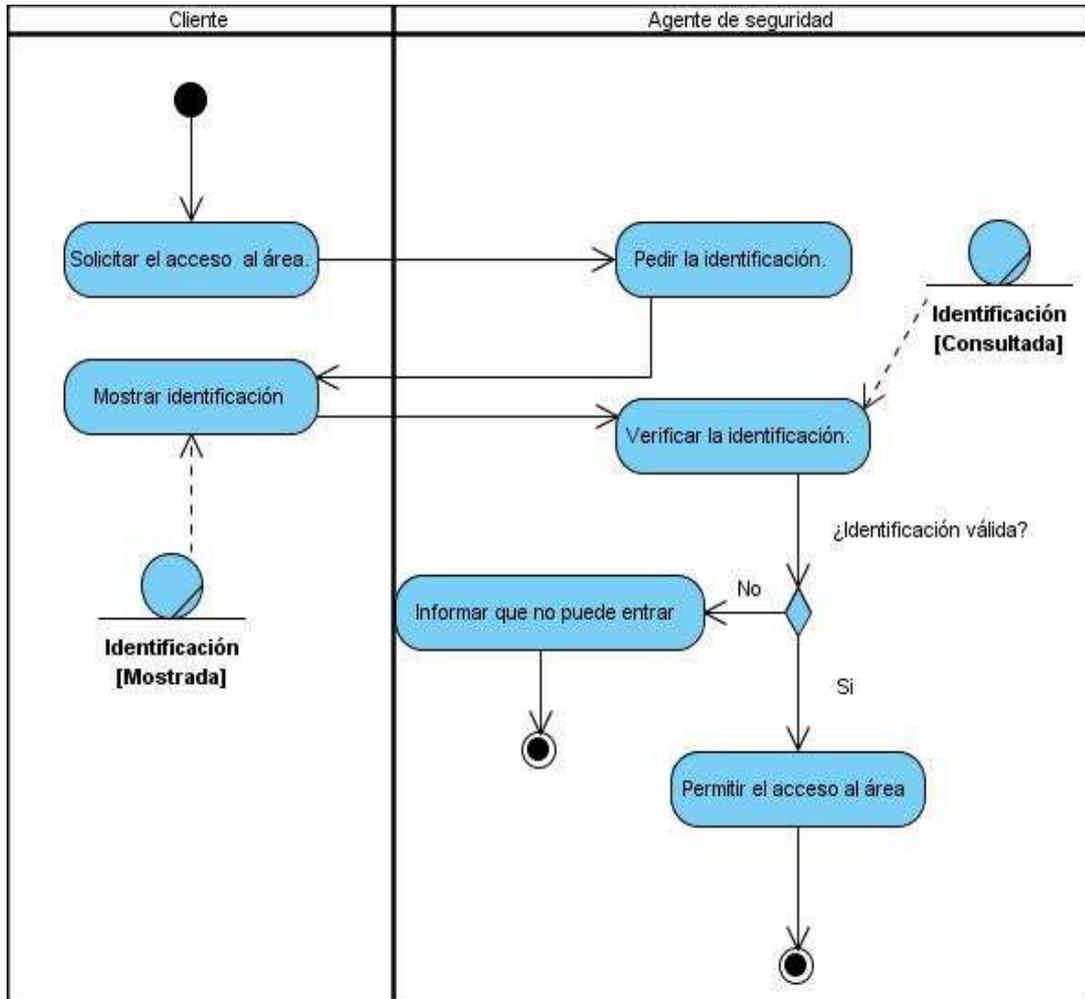


Figura 2 Diagrama de Actividades Registrar Acceso

• Caso de Uso Registrar Vehículo

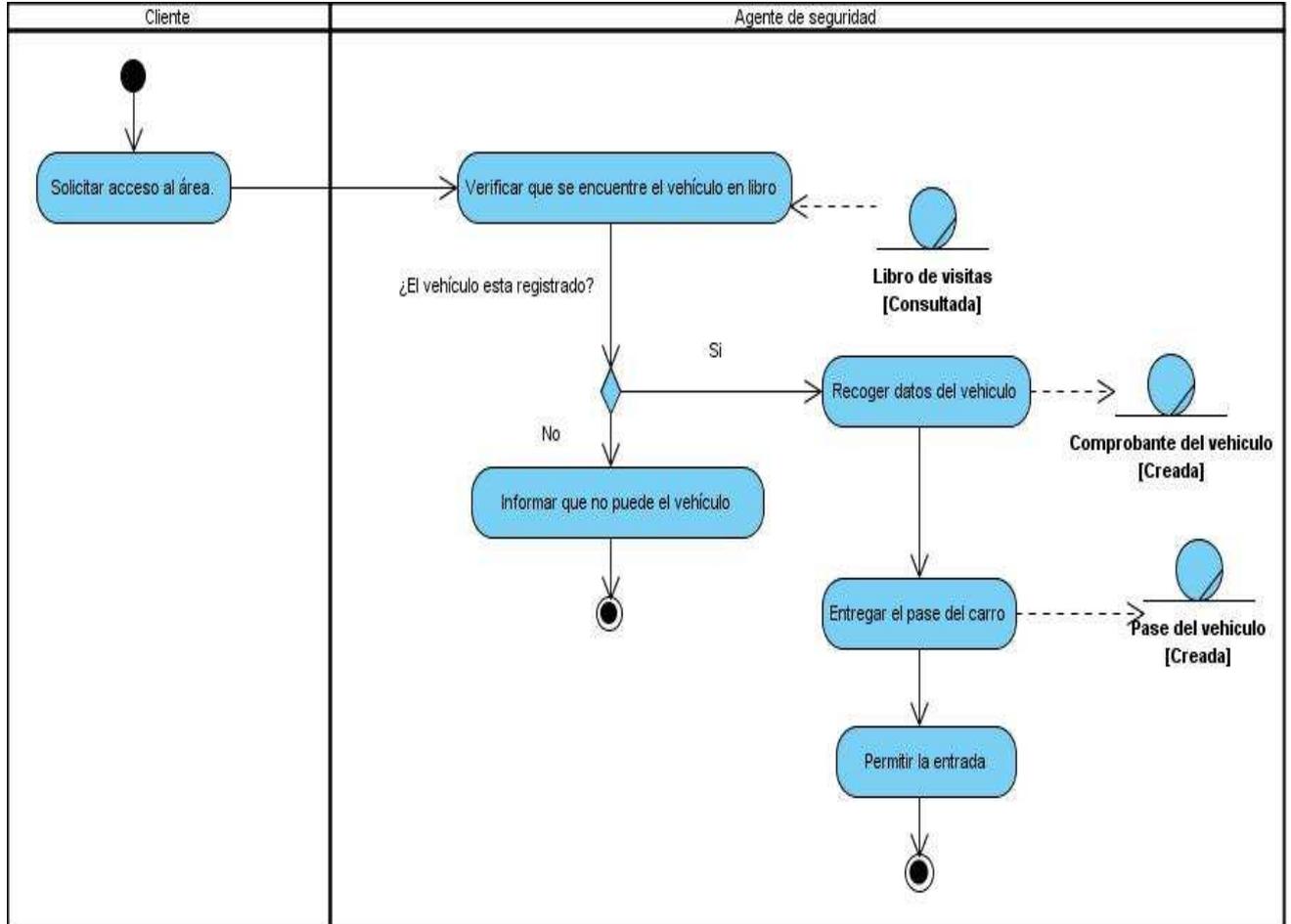


Figura 3 Diagrama de Actividades Registrar Vehículo

- **Caso de Uso Registrar Visita**

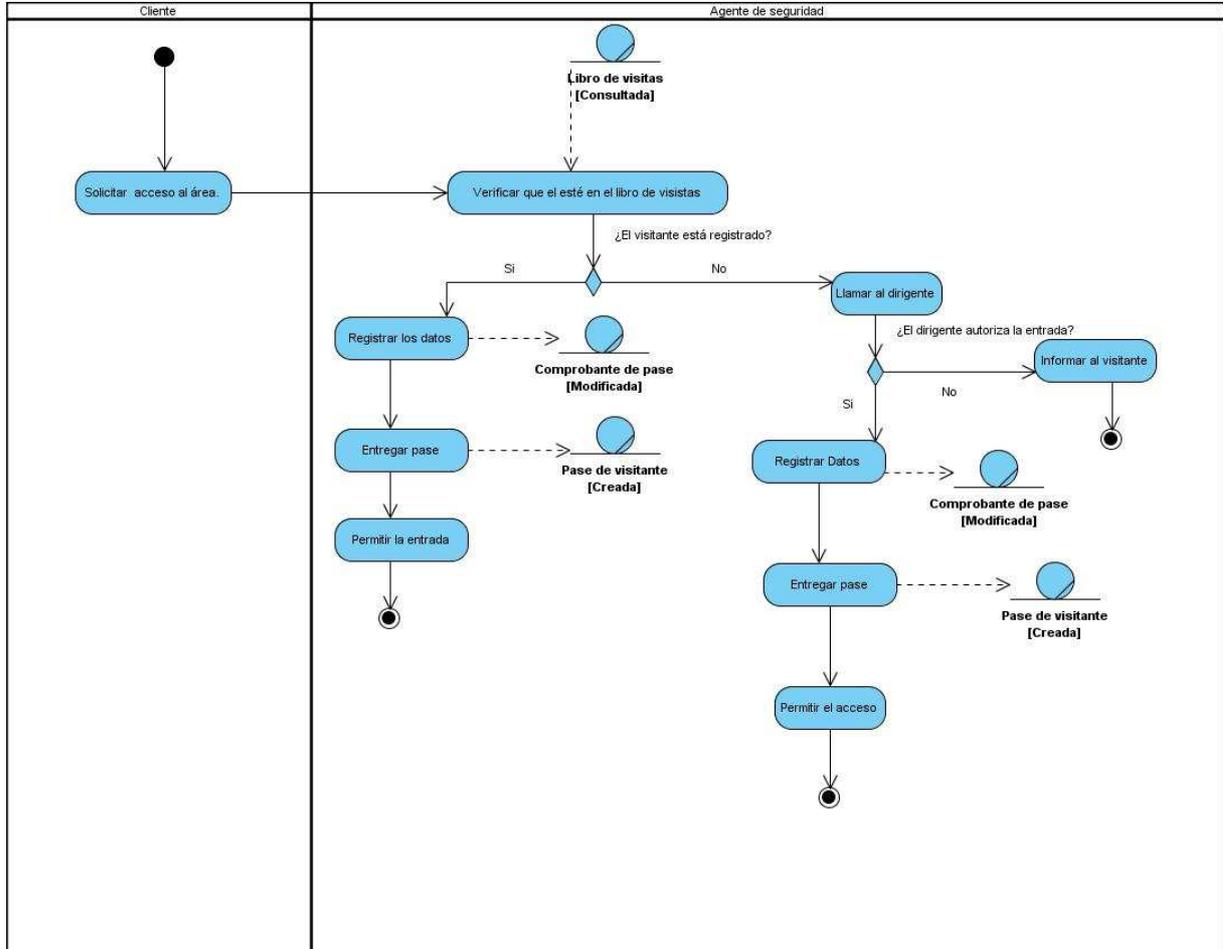


Figura 4 Diagrama de actividades Registrar Visita

2.3.5 Descripción de los casos de uso del negocio

- **Caso de Uso Registrar Acceso**

Caso de Uso:	Registrar Acceso
Actores:	Cliente
Trabajadores:	Agente de seguridad

Capítulo 2. Características del sistema

Resumen:	Este caso de uso se inicia cuando un estudiante, trabajador, profesor intenta ingresar a una determinada área en la Universidad de las Ciencias Informáticas (UCI). El agente le solicita la identificación y el local la muestra y se le permite el acceso.	
Precondiciones:		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El cliente le solicita el acceso al agente de seguridad a una determinada área.		
	2. El agente de seguridad le pide su identificación.	
3. El cliente muestra su identificación.		
	4. Verifica la identificación.	
	5. Si la identificación presentada es válida, permitir el acceso al área solicitada. Si la identificación presentada no es válida, ver el flujo alternativo 5.1	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
	5.1 Denegar el acceso al área solicitada	
Precondiciones	El cliente accede al área.	

Tabla 3 Descripción del CUN Registrar Acceso

- **Caso de Uso Registrar Vehículo**

Caso de Uso:	Registrar Vehículo	
Actores:	Cliente	
Trabajadores:	Agente de seguridad	
Resumen:	Este caso de uso se inicia cuando una persona ajena a la institución de la Universidad de las Ciencias Informáticas (UCI) solicita al acceso a la misma. El agente de seguridad revisa en el libro de visitas que esté registrado el vehículo y le permite el acceso.	
Precondiciones:	En el Libro de visitas debe constatar que va a acceder un vehículo a la Universidad.	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Negocio
	1. El cliente le solicita el acceso al agente de seguridad a una determinada área con un vehículo.	
		2. El agente de seguridad verifica que este autorizado a entrar con el vehículo.
		3. Si está autorizado, recoger datos del vehículo y entregarle un pase del carro. Sino no está autorizado ver el flujo alterno 3.1.
Flujos Alternos		
	Acción del Actor	Respuesta del Negocio
		3.1 Denegar el acceso del vehículo a la institución.
Precondiciones	Se registra el acceso del vehículo a la Universidad.	

Tabla 4 Descripción del CUN Registrar Vehículo

- **Caso de Uso Registrar Visita**

Caso de Uso:	Registrar Visita	
Actores:	Cliente	
Trabajadores:	Agente de seguridad	
Resumen:	Este caso de uso se inicia cuando una persona ajena a la institución de la Universidad de las Ciencias Informáticas (UCI) solicita al acceso a la misma. El Agente de seguridad verifica en el Libro de visitas y le permite el acceso.	
Precondiciones:	En el Libro de visitas debe constatar que un visitante va a acceder a la Universidad.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El cliente le solicita el acceso al agente de seguridad a una determinada área.		
	2. El agente de seguridad verifica en el libro de visitas que el cliente este registrado.	
	3. Si esta en el listado de los visitantes registran los datos en el comprobante de pase y se le entrega un pase de visitante sino está registrado ver el flujo alterno 3.1.	
	4. Si el cliente posee vehículo realizar el caso de uso extendido Registrar Vehículo.	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	

	3.1 Llamam al dirigente que autoriza la entrada y si este lo autoriza se realiza el paso 4, sino no le permiten la entrada.
Precondiciones	Se registra el acceso de un visitante.

2.4 Especificación de los requisitos del sistema

Tabla 5 Descripción del CUN Registrar Visita

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto.

2.4.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. (BOOCH, y otros, 2004).

1. Autenticar Usuario
2. Gestionar Usuario
 - 2.1. Registrar usuario
 - 2.2. Modificar usuario
 - 2.3. Eliminar usuario
3. Registrar Acceso
4. Registrar Visita
5. Registrar Vehículo
6. Gestionar Configuración del cliente
7. Registrar Sede
8. Registrar recursos de almacenamiento de fotos
9. Cantidad de Acceso por persona
10. Cantidad de Acceso por Puntos de Control
11. Cantidad de Personas que accedieron por un evento
12. Puntos de control a los que ha accedido una persona en un día
13. Asignar eventos a puntos de control

14. Gestionar Grupo

14.1 Modificar Grupo

14.2 Eliminar Grupo

14.3 Adicionar Grupo

15. Registrar puntos de control

16. Modificar puntos de control

17. Bloquear puntos de control

18. Gestionar Evento

18.1 Adicionar Evento

18.2 Modificar Evento

18.3 Eliminar Evento

19. Gestionar Rol

19.1 Adicionar rol

19.2 Bloquear rol

19.3 Modificar rol

20. Gestionar Dirigente que autoriza

20.1 Adicionar Dirigente que autoriza

20.2 Eliminar Dirigente que autoriza

20.3 Modificar Dirigente que autoriza

21. Gestionar Nomencladores Tipo Acceso

21.1 Adicionar Nomencladores tipo acceso

21.2 Modificar Nomencladores tipo acceso

22. Gestionar Nomencladores Atributo de Persona

22.1 Adicionar Nomencladores atributo de persona

22.2 Modificar Nomencladores atributo de persona

23. Gestionar Nomencladores Lugar a Visitar

23.1 Adicionar Nomencladores lugar a visitar

23.2 Modificar Nomencladores lugar a visitar

24. Registrar Estructuras

25. Modificar Estructuras

26. Bloquear Estructuras

2.4.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Son fundamentales en el éxito del producto y normalmente están vinculados a requisitos funcionales. (BOOCH, y otros, 2004).

Los requisitos no funcionales de la aplicación han sido agrupados por categorías y se muestran a continuación.

Apariencia o interfaz externa

El sistema posee una interfaz amigable al usuario.

Seguridad

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

Las personas que estén registrados como dirigentes son las únicas que van a dar el permiso para el acceso de visitantes a la UCI.

Rendimiento

Todas las salidas del sistema tienen que tener el 100% de veracidad y precisión.

El sistema deberá ser capaz de recuperarse rápidamente de las fallas, en un tiempo promedio de 2 minutos aproximadamente.

Portabilidad

El sistema deberá tener un 100% de disponibilidad para usarlo efectivamente.

Se adapta fácilmente en cualquier lugar que requiere un sistema de control de acceso.

Confiabilidad

El sistema cuenta con una Base de Datos embebidas para cuando no tengamos acceso a la red, el sistema siga funcionando.

2.5 Definición de los actores del sistema a automatizar

Actor	Justificación
-------	---------------

Administrador	Es el encargado de gestionar los usuarios, los permisos de estos, los grupos, los eventos, los puntos de control, los roles.
Agente de seguridad	Es el encargado de registrar todo tipo de acceso en el sistema, es el que se relaciona con el personal de la UCI y las personas ajenas que soliciten la entrada a la misma.

Tabla 6 Actores del sistema

A continuación se presenta el caso de uso resumido que fue seleccionado para darle seguimiento al desarrollo del trabajo, para satisfacer los requerimientos funcionales del sistema.

CU-1	Registrar Acceso
Actor(es)	Agente de seguridad.
Descripción	El caso de uso se inicia cuando un agente de seguridad introduce el código de una identificación. Luego de verificar el código le permite la entrada a la persona que la esté solicitando.
Referencia	RF3

Tabla 7 Descripción resumida del CU Registrar Acceso

2.6. Distribución de los casos de uso por paquetes

Los casos de uso del sistema se agruparon en varios paquetes, los cuales agrupan las funcionalidades que realiza cada actor. Los paquetes son los módulos en los que se dividió nuestro sistema, los módulos correspondientes son:

- **Módulo Administración**

Gestionar Grupo

Gestionar Eventos

Asignar Eventos a Puntos de Control

Gestionar Dirigente que Autoriza

Registrar Puntos de Control

Modificar Puntos de Control

Bloquear Puntos de Control

Registrar Estructuras

Modificar Estructuras

Bloquear Estructuras

- **Módulo Control de Acceso**

Configuración del Cliente

Registrar Acceso

- **Módulo Visitas**

Registrar Visita

Registrar Vehículo

- **Módulo Seguridad**

Gestionar Roles

Gestionar Usuarios

- **Módulo Configuración**

Registrar Sede

Gestionar Nomenclador

Registrar recursos de almacenamientos de fotos

- **Módulo Consultar**

Consultar Cantidad de Acceso por Persona

Consultar Cantidad de Acceso por Punto de Control

Consultar Cantidad de Personas que Accedieron por Evento

Las imágenes que se muestran a continuación representan los Diagramas de Casos de Uso del Sistema de cada módulo.

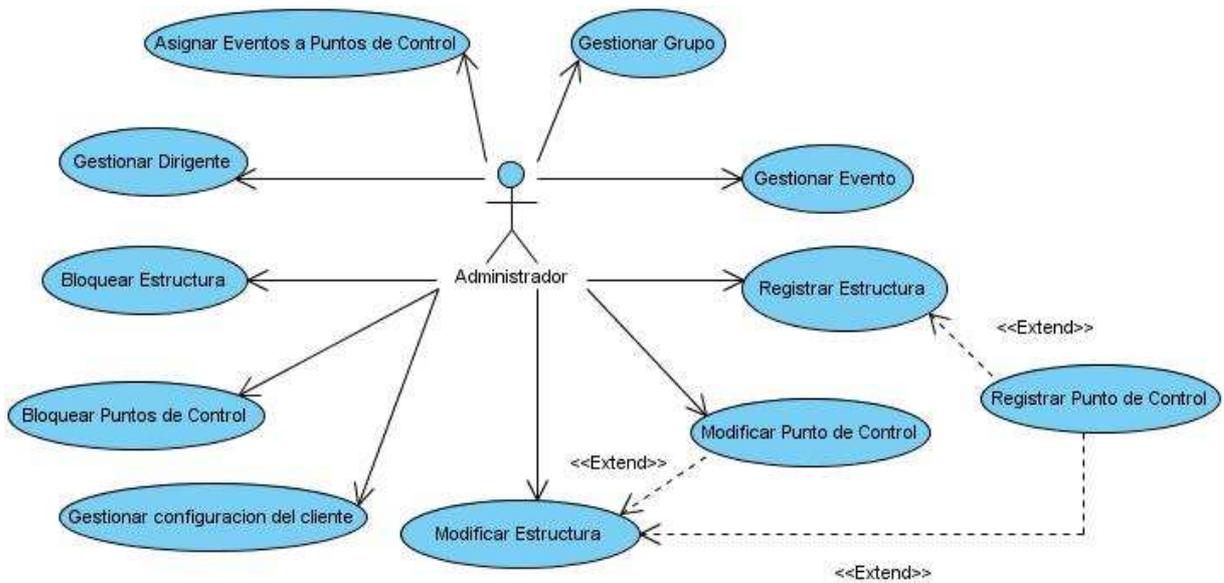


Figura 5 Diagrama de CU del Sistema: Módulo Administración



Figura 6 Diagrama de CU del Sistema: Módulo Control de Acceso



Figura 7 Diagrama de CU del Sistema: Módulo Visitas

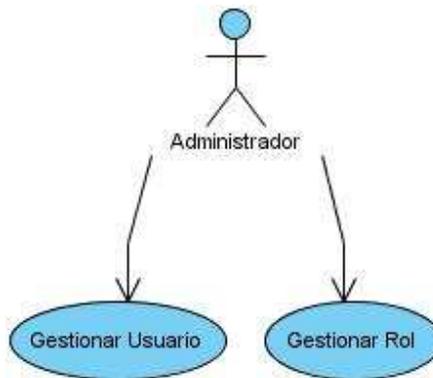


Figura 8 Diagrama de CU del Sistema: Módulo Seguridad

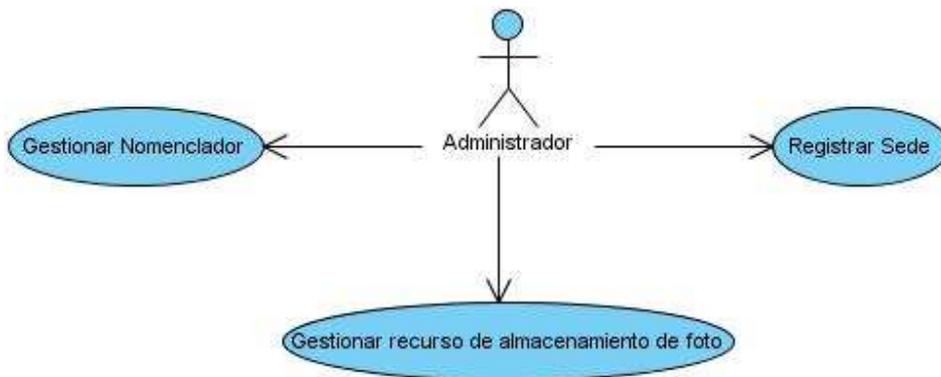


Figura 9 Diagrama de CU del Sistema: Módulo Configuración

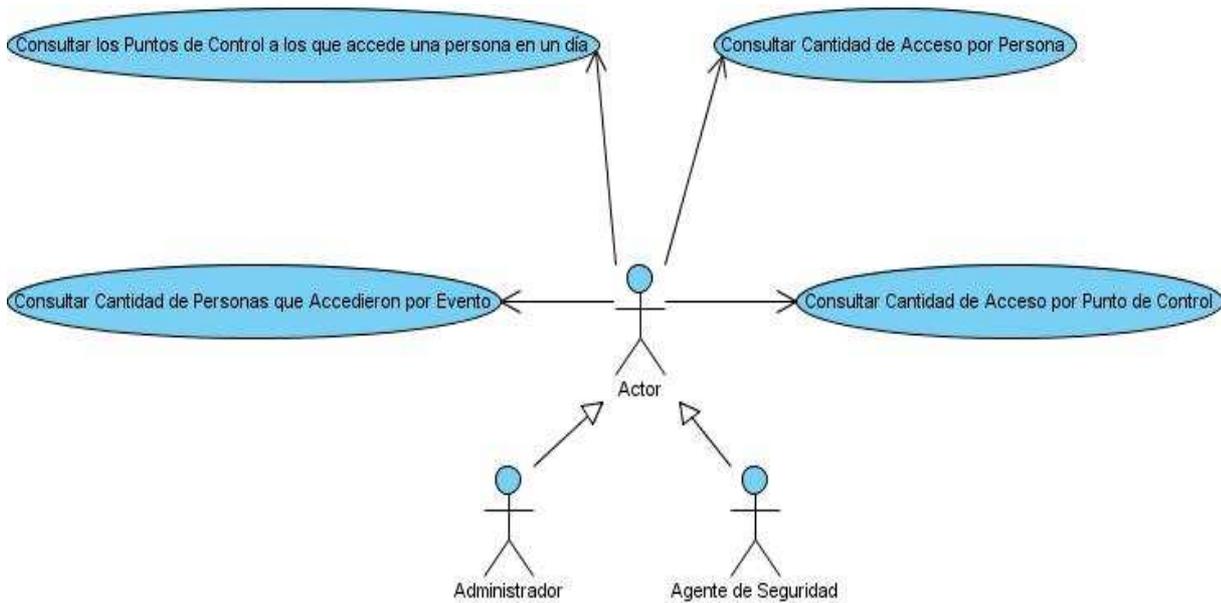


Figura 10 Diagrama de CU del Sistema: Módulo Reportes

2.7 Descripción expandida del caso de estudio Registrar Acceso.

Caso de Uso:	Registrar Acceso	
Actores:	Agente de Seguridad	
Resumen:	El caso de uso se inicia cuando un agente de seguridad introduce el código de una identificación, si no es correcto el sistema deberá alertar al agente de seguridad.	
Precondiciones:		
Flujo Normal de Eventos		
1. El agente de seguridad entra el código de acceso de la identificación.	2. El sistema comprueba que la identificación sea válida. Si la identificación no es válida ver el	

Capítulo 2. Características del sistema

	flujo alternativo 2.1
	3. El sistema muestra un mensaje “La identificación es válida”.
Flujos Alternos	
	2.1 En caso de no ser correctos el código de la identificación el sistema muestra el mensaje “La identificación no es válida”.
Poscondiciones	

Tabla 8 Descripción del CU Registrar Acceso

Para ver las restantes descripciones de los casos de usos ver la versión expandida del trabajo de diploma.

Conclusiones

En este capítulo se desarrolló la propuesta de solución del sistema atendiendo a las características que debe tener. A partir del modelo del negocio y las propias necesidades del cliente, se identificaron los requisitos funcionales del sistema los que fueron agrupados en casos del uso del sistema, se definieron los actores y se modeló el sistema, asumiendo una organización por módulos. Fueron identificados, además, los requisitos no funcionales del sistema. Los casos de uso fueron descritos de forma breve y tres de los casos de uso fueron descritos detalladamente.

Análisis y Diseño del sistema

Introducción

En este capítulo se realizan los diagramas de clases de análisis, los diagramas de interacción, así como los diagramas de clases del diseño para cada uno de los casos de uso determinados en la captura de los requisitos.

3.1 Diagramas de Clases del Análisis

El diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema.

A continuación se refleja el diagrama de clases de análisis correspondiente al caso de estudio Registrar Acceso.

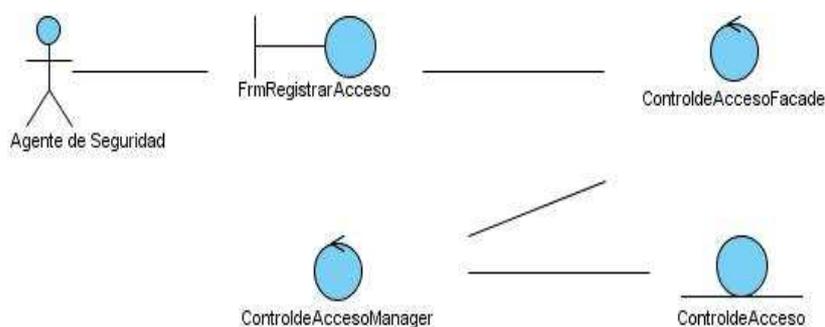


Figura 11 Diagrama de clases del análisis CU Registrar Acceso

Para ver los restantes diagramas de clases del análisis de los casos de usos ver la versión expandida del documento del trabajo propuesto.

3.2 Diagramas de Interacción

Los diagramas de interacción muestran cómo se comunican los objetos en una interacción. Existen dos tipos de diagramas de interacción:

- Diagramas de Colaboración: Resaltan la organización de los objetos que participan en una interacción.
- Diagramas de Secuencia: Resaltan el orden temporal de los mensajes.

Para modelar los diagramas de interacción de los casos de uso del Sistema de Control de Acceso, se escoge el tipo de diagrama secuencia debido a que los diagramas de secuencia son más adecuados para observar la perspectiva cronológica de las interacciones.

La figura que se muestran a continuación es el diagrama de interacción del caso de uso Registrar Acceso.

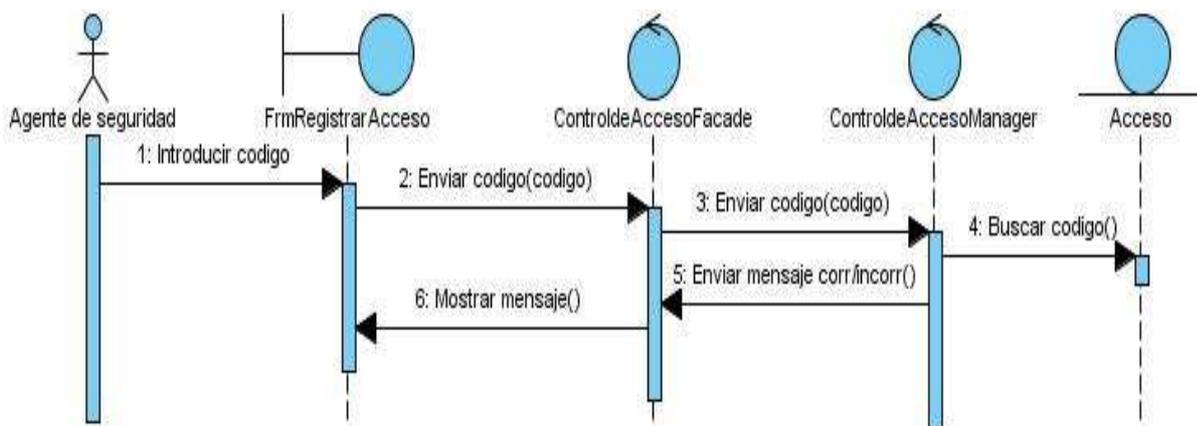


Figura 12 Diagrama de Secuencia CU Registrar Acceso

Para ver los restantes diagramas de interacción de los casos de usos ver la versión expandida del trabajo de diploma.

3.3 Diseño

En el diseño se modela el sistema, incluyendo la arquitectura, de forma que soporte todos los requisitos, el diseño no es más que el refinamiento del análisis, la forma como cumple el sistema sus objetivos. Es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Para lograr un sistema implementado sin ambigüedades es necesario un diseño eficiente.

3.3.1 Diagramas de clases del diseño

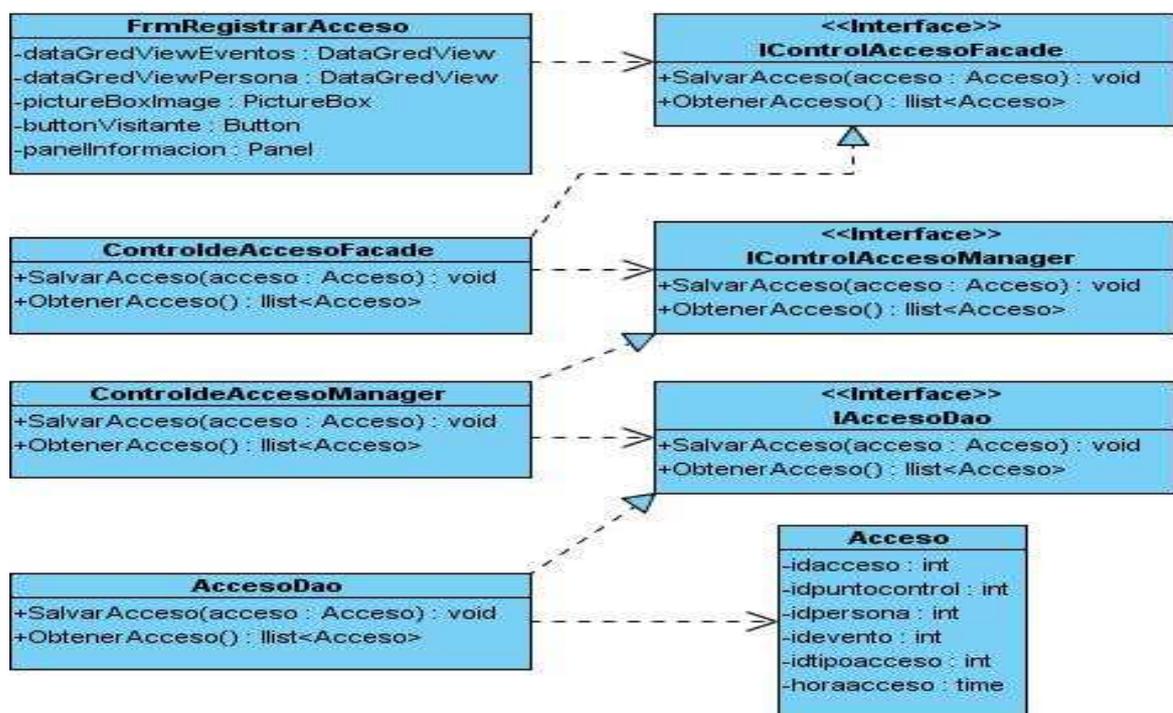


Figura 13 Diagrama de Clases del Diseño del CU Registrar Acceso

Para ver los restantes diagramas de clases del diseño de los casos de usos ver la versión expandida del trabajo de diploma.

3.4 Diseño de la Base de Datos

El diseño de la base de datos es un aspecto importante, teniendo en cuenta que debe brindar la persistencia al modelo que se ha descrito con anterioridad en los epígrafes.

3.4.1 Diagrama de Clases Persistentes

Todas las clases que se identifican en el dominio del análisis no son persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Es importante tener en cuenta que es responsabilidad del diseñador definir cuáles son las que deben ser clases persistentes.

La figura que se muestra a continuación presenta el Diagrama de Clases Persistentes.

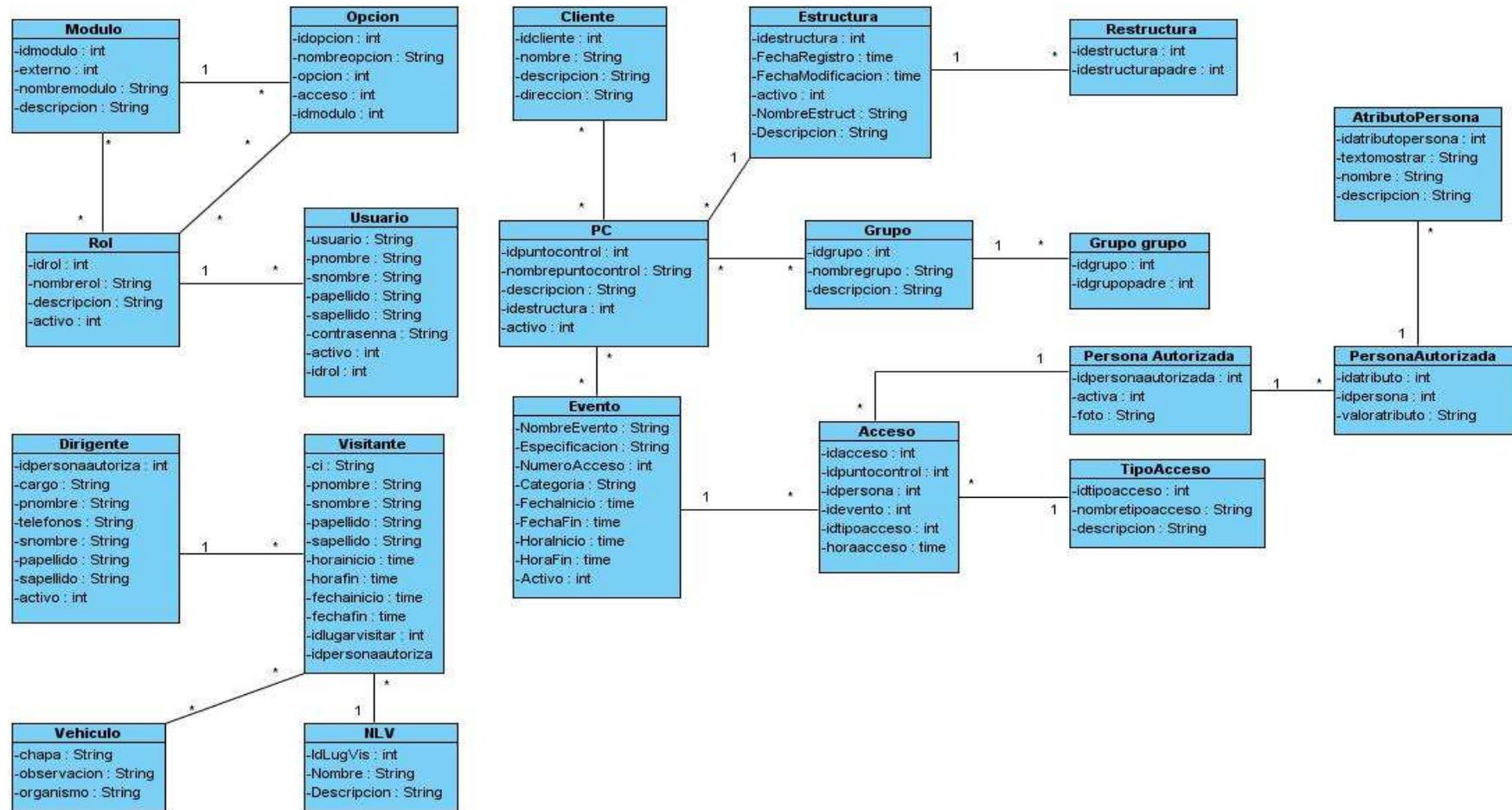


Figura 14 Diagramas de Clases Persistentes

3.4.2 Modelo de Datos

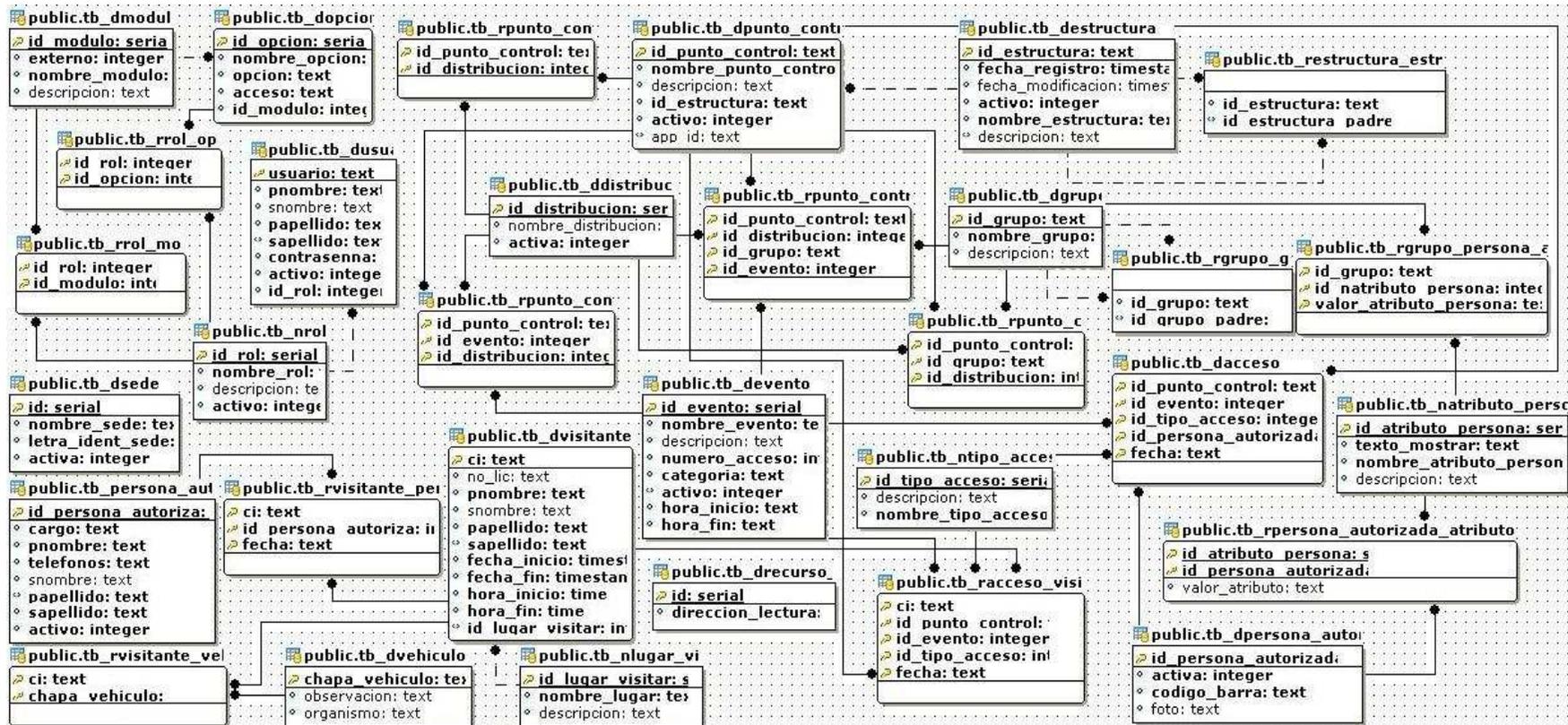


Figura 15 Modelo de Datos

3.4.3 Descripción de las tablas

Nombre: tb_acceso		
Descripción: Almacena el identificador del punto de control, de la persona autorizada, del evento así como el tipo de acceso cuando se produce un acceso.		
Atributo	Tipo	Descripción
id_punto_control	text	Identificador del punto de control
id_persona_autorizada	text	Identificador de la persona autorizada
id_evento	integer	Identificador del evento
id_tipo_acceso	integer	Identificador del tipo de acceso
hora_acceso	timestamp	Hora en que ocurre el acceso
id_acceso	serial	Identificador del acceso

Tabla 9 Descripción de la tabla tb_acceso

Nombre: tb_dcliente		
Descripción: Almacena todos los datos del cliente		
Atributo	Tipo	Descripción
id_cliente	text	Identificador del cliente
nombre	text	Nombre del cliente
descripcion	text	Breve descripción del cliente

direccion	text	Dirección del cliente
-----------	------	-----------------------

Tabla 10 Descripción de la tabla tb_dcliente

Para ver las restantes descripciones de las tablas de la base de datos ver la versión expandida del trabajo de diploma.

3.5 Tratamiento de Errores

En el sistema se redefinió el sistema de tratamiento de errores que trae implementado el framework .NET para adecuarlo a los posibles errores que puede arrojar el sistema. Cuando ocurre una excepción, el usuario es notificado mediante un mensaje que muestra el error correspondiente (ver Figura 16), por otra parte se utilizan mensajes de confirmación para acciones como inserción de datos (ver Figura 17).

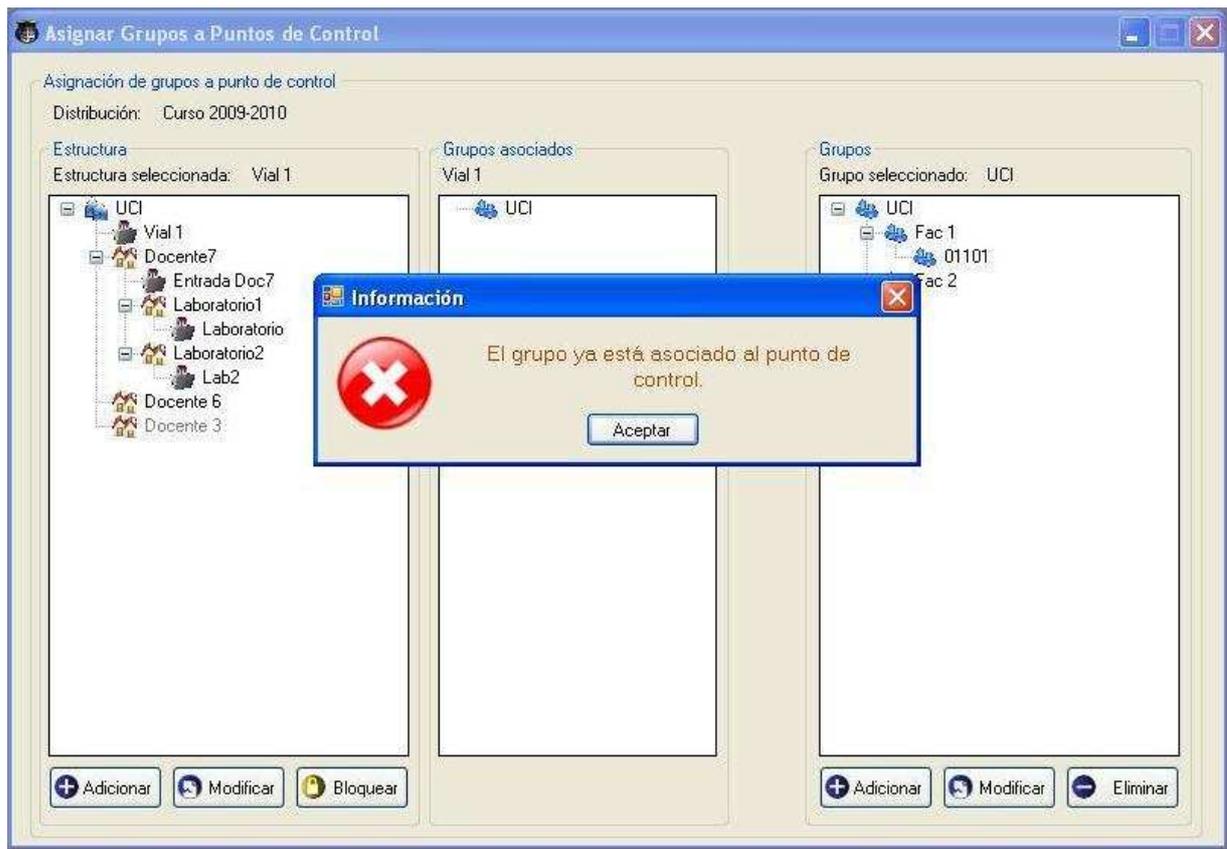


Figura 16 Mensaje de error

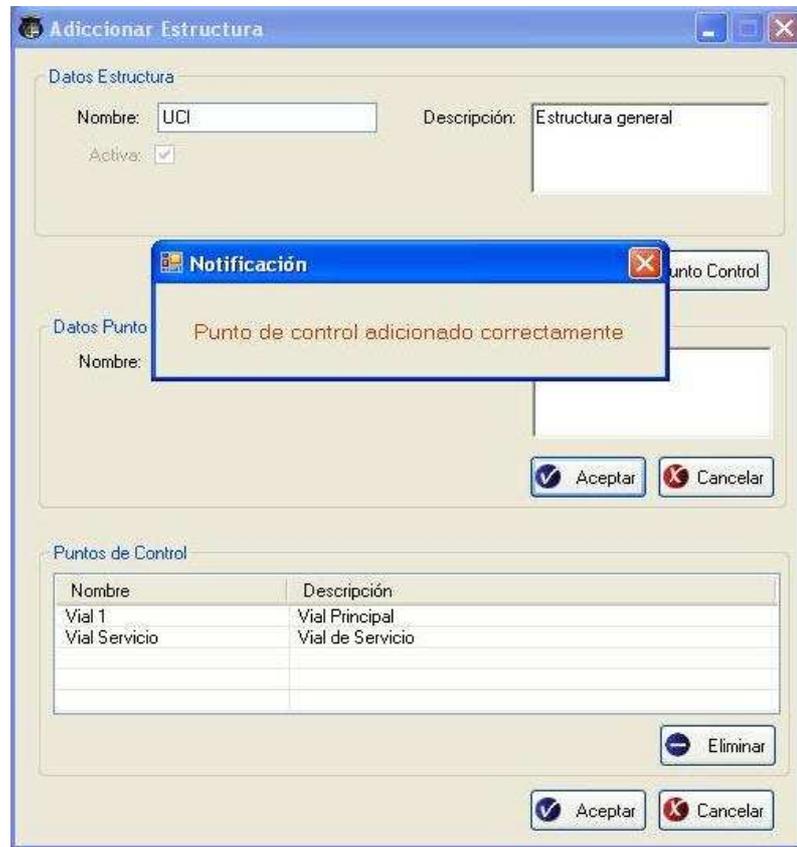


Figura 17 Mensaje de notificación

3.6 Estándares de Codificación

En el caso de las declaraciones, para declarar una clase se escribió el nombre de la clase con la primera letra en mayúscula y el resto del nombre en minúscula. Los métodos de una clase tienen una forma encamellada se capitaliza todas las palabras que lo conforman. Las variables se declararon explícitamente sin usar abreviaturas y las constantes se declararon en mayúscula.

Los comentarios se definen comenzando con los caracteres `/*` y terminando con `*/` Ejemplo: `/* esto es un comentario */`, siempre que se quiera comentar grandes instrucciones de códigos. Si solo se desea comentar una sola instrucción se utiliza `//`. Cuando se implementa un método se pone un comentario en el encabezado que nos permita visualizar al mismo de forma más rápida.

La indentación es a tres espacios en blanco.

Entre los operadores aritméticos y sus operandos así como entre los operadores lógicos aritméticos y sus operandos se dejó un espacio en blanco ejemplo:

```
if (personas[i].Nombre == 'A' &&personas[i].Grupo == '15510')
```

Las llaves para abrir y cerrar una clase, un método o un bloque de control de flujo se sitúan al mismo nivel del bloque al que pertenecen, ejemplo:

```
if (personas[i].Nombre == 'A' &&personas[i].Grupo == '15510')
{
    return true;
}
```

Conclusiones

En el presente capítulo se realizaron los diagramas de clases del análisis, diagramas de interacción, diagramas de clases del diseño así como el diseño de la base de datos, teniendo en cuenta el diagrama de clases persistentes y el modelo de datos, todo esto dando continuación al desarrollo de la propuesta de solución.

Implementación y Prueba

Introducción

En este capítulo se desarrollan los flujos de implementación y prueba, se describe cómo los elementos del modelo de diseño son implementados en términos de componentes y cómo se organizan en el modelo de despliegue. Además se exponen los diferentes casos de pruebas para cada caso de uso, siguiendo específicamente el método de pruebas de caja negra.

4.1 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos los cuales están unidos por conexiones de comunicación.

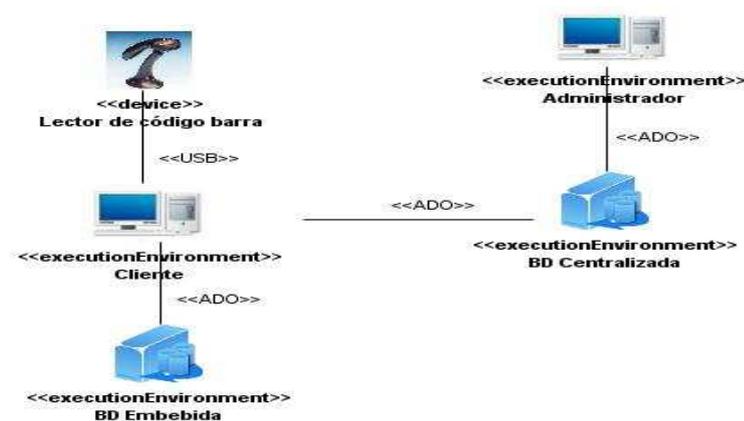


Figura 18 Diagrama de despliegue

4.2 Diagramas de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes, además de ayudar a entender mejor el modelo de implementación. Los diagramas de componentes contienen componentes, interfaces y relaciones entre ellos. También puede contener paquetes utilizados para agrupar elementos del modelo.

Se van a presentar los diagramas de componentes, cada diagrama es un módulo del sistema.

A continuación se muestra el diagrama de componente del módulo administración para ver los restantes diagramas ver el documento extendido del trabajo de diploma.

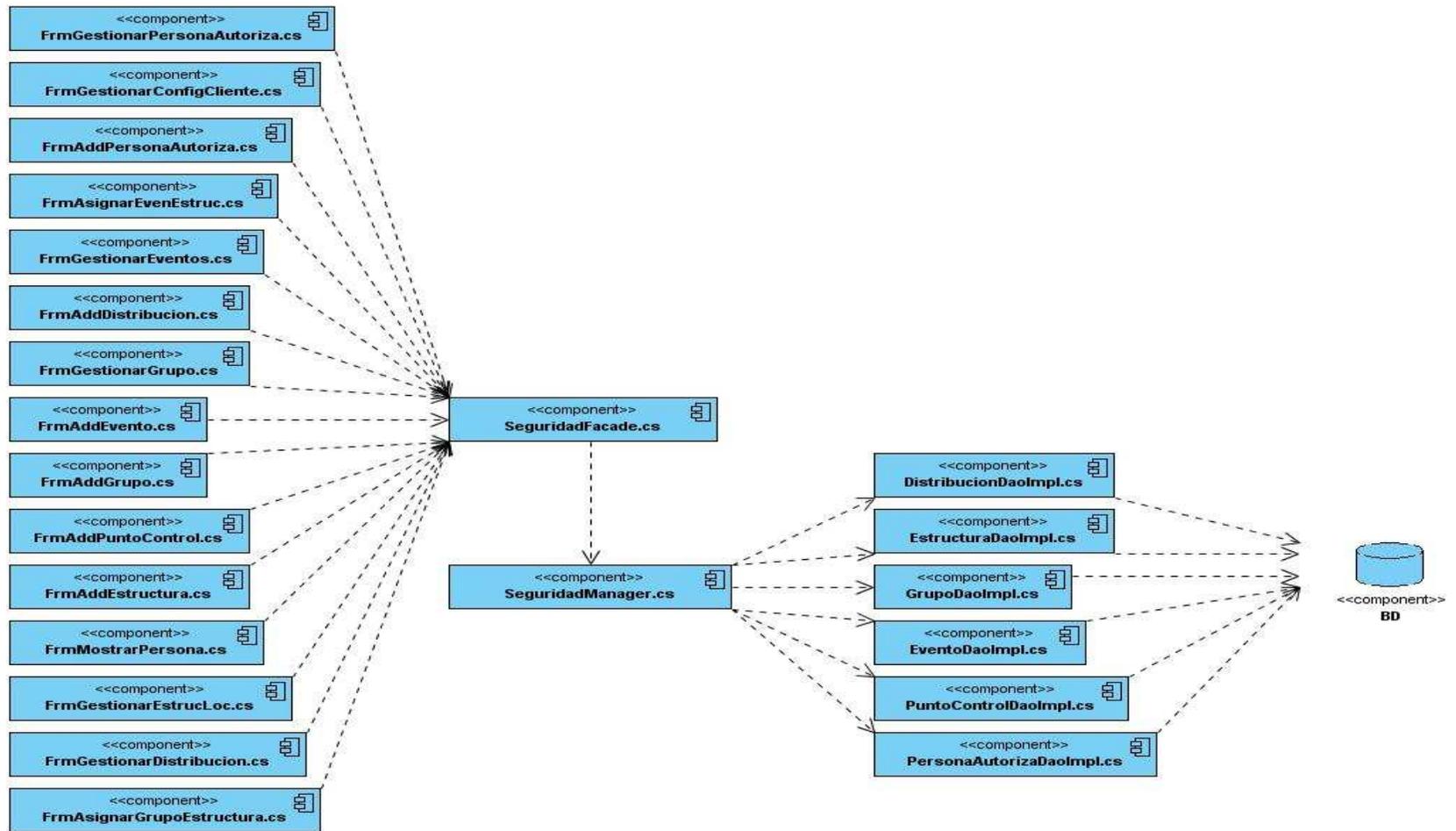


Figura 19 Diagrama de Componentes Módulo Administración

4.3 Arquitectura y Patrones utilizados

4.3.1 Arquitectura

Las técnicas metodológicas desarrolladas con el fin de facilitar la programación se engloban dentro de la llamada Arquitectura de Software o Arquitectura Lógica. Se refiere a un grupo de abstracciones y patrones que nos brindan un esquema de referencia útil para guiarnos en el desarrollo de software dentro de un sistema informático. Así, los programadores, diseñadores, ingenieros y analistas pueden trabajar bajo una línea común que les posibilite la compatibilidad necesaria para lograr el objetivo deseado. (BOOCH, y otros, 2004).

El sistema desarrollado está basado en una arquitectura por capas, la cual es una especificación de la arquitectura cliente-servidor¹⁰. La arquitectura por capas brinda un estilo de programación donde el objetivo primordial es la separación de la lógica de negocio de la lógica de diseño. En el caso específico del sistema desarrollado la arquitectura está compuesta por tres capas; una capa de presentación la cual es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio, en la capa de lógica de negocio se reciben las peticiones del usuario y se envían las respuestas tras el proceso además de establecer todas las reglas que deben cumplirse. La capa de lógica de negocio se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para almacenar o recuperar datos de la base de datos. La capa de acceso es donde residen los datos y es la encargada de acceder a los mismos. (Ver Figura 20 Arquitectura por capas).

¹⁰**Cliente-servidor:** Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta.



Figura 20 Arquitectura por capas

Este tipo de arquitectura permite que el desarrollo del sistema se pueda llevar a cabo en varios niveles y en caso que se produzca algún cambio, sólo se ataca el nivel requerido sin tener que revisar entre código mezclado, además permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles.

4.3.2 Patrones utilizados

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan nos brindan una estructura conocida para todos los programadores, de manera que la forma de trabajar no resulte distinta entre los mismos, permiten tener una estructura de código común, ahorran grandes cantidades de tiempo en la construcción de un software, son fáciles de comprender, mantener y extender además de dar una mejor imagen de profesionalidad y calidad.

Existen varios tipos de patrones entre los que se encuentran los patrones estructurales, patrones de creación y los patrones de asignación de responsabilidad. En el sistema desarrollado se utilizaron los siguientes patrones:

Patrones estructurales:

- Fachada: el cual provee una interfaz unificada y sencilla que funciona de intermediaria entre un cliente y una interfaz o grupos de interfaces más complejas.

- Inversión de control: es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos. Este patrón es el concepto fundamental del framework Spring.NET, ya que, implementa un contenedor que se encarga de gestionar las instancias (así como sus creaciones y destrucciones) de los objetos del usuario.

Patrones de creación:

- Singleton: garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

Patrones de asignación de responsabilidad:

- Alta cohesión: dice que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase.
- Bajo acoplamiento: es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.
- Polimorfismo: se lleva a cabo siempre que una responsabilidad o comportamiento relacionados varían dependiendo del tipo (Clase).

4.4 Modelos de pruebas

Se puede llamar pruebas de software al proceso de ejercitar un programa con la intención específica de encontrar errores previos a la entrega al usuario final. Las pruebas son además una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. El objetivo de la prueba de software es descubrir errores. En todas las fases del desarrollo del proyecto hay que probar el software que se va construyendo. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

Para que el desarrollo de las pruebas de software tenga éxito hay que tener en cuenta los siguientes elementos:

- Estrategia de Prueba
- Niveles de Prueba
- Tipo de Prueba
- Método de Prueba

Existen distintos métodos de pruebas, entre los cuales se encuentra Prueba de Caja Negra; prueba que se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

A continuación se muestra el caso de prueba del caso de estudio Registrar Acceso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
Registrar Acceso	Registrar Acceso de forma correcta.	El sistema permite registrar un acceso.	El agente de seguridad verifica el código de la identificación. El sistema registra el acceso.
	Registrar Acceso incorrectamente.	El sistema muestra un mensaje de error.	El agente de seguridad verifica el código de la identificación y no está correcto. El sistema muestra un mensaje de error.

Tabla 11 CU Registrar Acceso a probar en el sistema

Capítulo 4. Implementación y Prueba

Para ver los restantes caso de prueba ver la versión expandida del trabajo de diploma.

Como resultado de las pruebas se detectaron 10 no conformidades, de las cuales todas procedieron al cambio. En la tabla que se muestra a continuación se detallan las no conformidades y la solución de las mismas.

Elemento	No	Pedido de cambio	Áreas	Solución	Tiempo de Sol.(d)	Proced e	Import ancia
Bloquear Punto de Control	1	Deficiencia al no seleccionar el punto de control que se quiere bloquear no muestra un mensaje de error.	Módulo Administración	Se va a lanzar un mensaje de error cada vez que ocurra esto.	Tres días para darle la solución.	Sí	Baja
Gestionar Rol	2	Deficiencia a la hora de insertar un nuevo rol, al dejar campos en blanco y no mostrar un mensaje de error.	Módulo Seguridad	Se va a validar que los campos no pueden quedar vacíos.	Tres días para darle la solución.	Sí	Baja
Gestionar Usuario	3	Falta de ortografía en el mensaje de error al crear un usuario que ya estaba registrado.	Módulo Seguridad	Se va a corregir la falta de ortografía.	Tres días para darle la solución.	Sí	Baja

Capítulo 4. Implementación y Prueba

Elemento	No	Pedido de cambio	Áreas	Solución	Tiempo de Sol.(d)	Proced e	Import ancia
Autenticación	4	Petición de cambio de forma de autenticarse	Autenticación	Se le van a cambiar los nombres de los botones a la hora de autenticarse.	Tres días para darle la solución.	Sí	Baja
Gestionar Grupo	5	Cuando se va a realizar la búsqueda de una persona para en el CU Gestionar Grupo por el concepto del nombre de la persona, existe dificultad a la hora de mostrar los datos de esa persona, pues algunos datos no son reales.	Módulo Administración	Modificar la implementación.	Tres días para darle la solución.	Sí	Baja
Gestionar Evento	6	No se puede eliminar un evento que no se va a realizar más.	Módulo Evento	Modificar la implementación.	Tres días para darle la solución.	Sí	Baja

Capítulo 4. Implementación y Prueba

Elemento	No	Pedido de cambio	Áreas	Solución	Tiempo de Sol.(d)	Proced e	Import ancia
Gestionar Dirigente que	7	Cuando se bloquea una persona no se muestra un mensaje que se hizo, por tanto no se sabe si se bloquea o no.	Módulo Administración	Mostrar el mensaje si se realiza la acción.	Tres días para darle la solución.	Sí	Baja
Registrar Sede	8	Botón sin funcionamiento	Módulo Configuración	Modificar la implementación.	Tres días para darle la solución.	Sí	Baja
Registrar Sede		Cambiarle el nombre a la interfaz Gestionar Sede por Registrar Sede porque en realidad es lo que se hace.	Módulo Configuración				

Elemento	No	Pedido de cambio	Áreas	Solución	Tiempo de Sol.(d)	Proced e	Import ancia
Asignar Grupos a Puntos de Control	9	Cuando se va a realizar la asignación de grupos a puntos de control existe una deficiencia pues si se asigna primero un grupo de una facultad y luego se asigna la facultad completa, el sistema no reconoce que un grupo de esa facultad está asignado.	Módulo Administración	Modificar la implementación.	Tres días para darle la solución.	Sí	Baja
Gestionar NAP	10	Arreglar el mensaje de error al insertar un nomenclador atributo de persona dejando algunos campos vacíos.	Módulo Configuración	Se va a arreglar el mensaje mostrado.	Tres días para darle la solución.	Sí	Baja

Tabla 12 Las no conformidades y la solución

Las no conformidades que se encontraron son de importancia baja, de las 10 que se encontraron todas tuvieron solución un tiempo máximo de tres días, lo que implica que el sistema desarrollado presenta buena calidad.

Conclusiones

En este capítulo fue presentado cómo está construido el sistema a partir del diagrama de componentes, el cual representa las dependencias entre los componentes del software y de despliegue, el cual ilustra los nodos que serán usados para la implantación de la aplicación y para cada uno de éstos el protocolo de comunicación. Además se comprobó el correcto funcionamiento del Sistema de Control de Acceso mediante las pruebas de Caja Negra realizadas.

CONCLUSIONES GENERALES

En el presente trabajo se arribó a las siguientes conclusiones:

- ✓ Los Sistemas de Control de Acceso permiten llevar un control específico de un lugar determinado, favorecen a los usuarios brindándole información actualizada.
- ✓ Los Sistemas de Control de Acceso poseen características que son comunes:
 - Brindan información específica.
 - Permiten llevar un mejor control del lugar en el que esté implantado el sistema.
- ✓ Las Herramientas óptimas seleccionadas para desarrollar el Sistema de Control de Acceso fueron:
 - C# como lenguaje de programación
 - PostgreSQL 8.3 como Gestor de Base de Datos Relacional.
 - DB4O como Gestor de Base de Datos de objetos
 - Spring.Net.
- ✓ El Sistema desarrollado:
 - Garantiza una mejor comunicación entre las personas encargadas de llevar el control del acceso a algún lugar y la persona que solicite la entrada.
 - Permite estructurar los diferentes puntos de control que va a poseer la entidad donde se va a ejecutar dicha aplicación así como los grupos que van a tener acceso a estos puntos de control y los eventos que se van a ocurrir en dicho punto de control.
 - Brinda funcionalidades que permiten un control de todos los accesos que se realizan en todos los lugares en la universidad, así como brindar reportes de ello.

De esta forma se le dio cumplimiento al objetivo planteado en la Introducción:

- ✓ Desarrollar un sistema que permita gestionar el control de acceso a la Universidad de las Ciencias Informáticas.

RECOMENDACIONES

Se recomienda:

- Incorporar nuevos reportes que permitan realizar un análisis más completo de la información gestionada referente a los accesos.
- Desarrollar un manual de usuario que brinde una mejor forma de comprender en qué orden y como se debe de realizar cada proceso.

REFERENCIAS BIBLIOGRÁFICAS

AMTEL. 2005. [En línea] 2005. [Citado el: 1 de Febrero de 2010.] <http://www.amtel-security.com/spanish/products/ACS.html>.

Amtel Security System. 2005. Sistema de Control de Entrada Vehicular de AMTEL. *Sistema de Control de Entrada Vehicular de AMTEL*. [En línea] 2005. [Citado el: 3 de Febrero de 2010.] <http://www.amtel-security.com/Spanish>.

BOOCH, G., RUMBAUGH, J. y JACOBSON, I. 2004. *El Proceso Unificado de Desarrollo de Software*. La Habana : Félix Varela, 2004.

CollabNet. 2010. AnkhSVN - Subversion Support for Visual Studio. *AnkhSVN - Subversion Support for Visual Studio*. [En línea] 2010. [Citado el: 6 de Febrero de 2010.] <http://ankhsvn.open.collab.net/>.

—. **2010.** The Forrester Wave for SCM. *The Forrester Wave for SCM*. [En línea] 2010. [Citado el: 5 de Febrero de 2010.] http://www.collab.net/forrester_wave_report/index.html.

Dorlet. SECTOR_SEGURIDAD_CONTROL_EDIFICIOS_INTELIGENTES. [En línea] [Citado el: 29 de Enero de 2010.] http://www.elpais.com/articulo/pais/vasco/DORLET_/SECTOR_SEGURIDAD/CONTROL/EDIFICIOS/INTELIGENTES/.

Historia de los sistemas de control de acceso. **Rodríguez Berzosa, Luis.**

IDENTIFIC-AR. 2006. Lector de código de barras láser. *Lector de código de barras láser*. [En línea] 2006. <http://identific-ar.com.ar/ms9520.htm>.

Intelektron. 2008. Intelektron Solutions. *Intelektron Solutions*. [En línea] 2008. [Citado el: 25 de Enero de 2010.] <http://www.intelektron.com/sitio/identifi/touch.htm>.

Kimaldi. 2007. kimaldi.com. [En línea] 2007. [Citado el: 1 de Febrero de 2010.] http://www.kimaldi.com/sectores/universidades/control_de_accesos_universidades.

March Hare Pty Ltd & CVSNT Project. 2004. CVS Soporte Profesional y CVSNT en Español. *CVS Soporte Profesional y CVSNT en Español*. [En línea] 2004. [Citado el: 5 de Febrero de 2010.] <http://www.march-hare.com/cvspro/es.asp>.

Microsoft Corporation. 2010. MSDN. *MSDN*. [En línea] 2010. [Citado el: 3 de Febrero de 2010.] <http://msdn.microsoft.com/es-es/default.aspx>.

Microsoft Corporation. msdn.microsoft.com. [En línea] [Citado el: 28 de Enero de 2010.] <http://msdn.microsoft.com/es-es/library/aa291305%28VS.71%29.aspx>.

Ninatec S.A. 2006. Tarjetas Magnéticas - Software & Hardware Solutions. *Tarjetas Magnéticas - Software & Hardware Solutions*. [En línea] 2006. [Citado el: 23 de Enero de 2010.] <http://www.ninatec.com/>.

OmegaSphere. 2005. SQL Database References. *SQL Database References*. [En línea] 2005. [Citado el: 4 de Febrero de 2010.] <http://www.sql.org/>.

pgAdmin PostgreSQL Tools. 2010. PostgreSQL administration and management tools. *PostgreSQL administration and management tools*. [En línea] 2010. [Citado el: 4 de Febrero de 2010.] <http://www.pgadmin.org/>.

Real Academia Española. 2007. *Diccionario de la lengua española*. España : s.n., 2007.

Registro Electrónico de Pacientes. 1999), **DR. ATHOS A. SÁNCHEZ MANSOLO. 1999.** 1999.

Spring Source Community. 2010. Spring .Net Applications Framework. *Spring .Net Applications Framework*. [En línea] 2010. [Citado el: 5 de Febrero de 2010.] <http://www.springframework.net/>.

SQL Manager.net. 2010. EMS Database Management Solutions. *EMS Database Management Solutions*. [En línea] 2010. [Citado el: 4 de Febrero de 2010.] <http://sqlmanager.net/products/postgresql/manager>.

Telvent. www.telvent.com. [En línea] [Citado el: 29 de Enero de 2010.] http://www.abengoa.com/corp/web/es/noticias_y_publicaciones/noticias/historico/noticias.

TrajeSol. 2005. Tarjetas con código de barras | Fabricación y distribución. *Tarjetas con código de barras | Fabricación y distribución*. [En línea] 2005. [Citado el: 23 de Enero de 2010.] <http://www.tarjetas-tarjesol.com/tarjetas-pvc/tarjetas-con-codigo-de-barras.php>.

Universal Control & Security. 2003. Bioaccess. *Bioaccess*. [En línea] 2003. [Citado el: 3 de Febrero de 2010.] <http://www.univcontrol.com/centro2.htm>.

VERSANT CORP. 2010. Java & .Net Object Database. *Java & .Net Object Database*. [En línea] 2010. [Citado el: 5 de Febrero de 2010.] <http://www.db4o.com/espanol/>.

W3C. 2003. Extensible Markup Language. *Extensible Markup Language*. [En línea] 2003. [Citado el: 4 de Febrero de 2010.] <http://www.w3.org/XML/>.

Wingerd, Laura. 2005. *Practical Perforce*. s.l. : O'Reilly, 2005. ISBN 0-596-10185-6.

Workflow Management Coalition. WfMC. [En línea] [Citado el: 20 de 12 de 2009.] <http://www.wfmc.org>.

GLOSARIO DE TÉRMINOS

ACLs: Identifican a los usuarios y a los grupos a los que se le asignan o deniegan los diferentes tipos de acceso al objeto o recurso asegurable.

AMTEL: Ofrece una amplia gama de soluciones para todos tipos de sistemas de portería vehicular con guardia y sistemas universales de porterías vehicular desatendida. (AMTEL, 2005).

CLR: Es el centro neurálgico del Framework de .NET encargado de gestionar la ejecución de las aplicaciones, aplicar parámetros de seguridad y ejecutar el denominado recolector de basuras. (Microsoft Corporation, 2010).

CVS (del inglés Concurrent Versions System): Es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren. (March Hare Pty Ltd & CVSNT Project, 2004).

Dorlet: La empresa guipuzcoana que se dedica a desarrollar tecnología de control de acceso a edificios.

Frameworks: Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Inversión de control: **Inversión de control (IoC** *del ingles Inversion of Control* en inglés,) es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos o funciones.

Licencia BSD: Licencia de software libre permisiva, permite el uso del código fuente en software no libre.

Mínimo privilegio: Permiso exclusivo sobre el menor recurso, es otorgado por una persona con autoridad. (Real Academia Española, 2007).

Proceso: Es un conjunto de uno o más procedimientos o actividades directamente ligadas, que colectivamente realizan un objetivo del negocio normalmente dentro del contexto de una estructura organizacional que define roles funcionales y relaciones entre los mismos. (Workflow Management Coalition).

Redes: Una red es una red de enlaces y nodos ordenados para la comunicación a distancia, donde los mensajes pueden pasarse de una parte a otra de la red sobre múltiples enlaces y a través de varios nodos.

Seguridad Preventiva: Es la protección de la vida, integridad física, bienes y posesiones, a través de un análisis de riesgos, acciones y medidas precautorias pertinentes, que tienen como finalidad la reducción de contingencias o eventualidades en el mayor grado posible ante hechos de carácter natural o de conductas mal intencionadas o antisociales.

Telvent (Nasdaq: TLVT), la compañía global de tecnologías de la información en tiempo real, es especialista en soluciones de alto valor añadido en cuatro sectores industriales (Energía, Tráfico, Transporte y Medio Ambiente).

Kimaldi: Kimaldi Electronics es una empresa dedicada a la fabricación, diseño y comercialización de productos dentro del mundo de la Identificación de personas.