

Universidad de las Ciencias Informáticas

Facultad 1



Título: “Sistema de Reportes de Accesos de la Universidad de las Ciencias Informáticas.”

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

Autores: Aylín Ricaño Gutiérrez

Lester Anías Rosales

Tutor: Ing. Roberlán Rodríguez Sánchez

Ciudad de La Habana, junio 2010

*Nunca consideres el estudio como una obligación,
sino como una oportunidad para penetrar
en el bello
y maravilloso mundo del saber.*

Albert Einstein.

DECLARACIÓN DE AUTORÍA

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

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

Firma del autor

Firma del autor

Firma del tutor

AGRADECIMIENTOS

Agradezco:

A la revolución por haberme dado la posibilidad de estudiar esta carrera.

A mi mamá Maritza y mi abuela Milagros por haberme ayudado siempre y apoyado en todo, dándome los mejores consejos.

A mi papá Edel por haberse preocupado todo el tiempo por mí.

A mi familia toda, por su cariño y confianza depositada en mí.

A mi novio (Yix) por su inseparable compañía y que tanto me ha ayudado a hacer realidad este sueño.

A todos mis amigos y compañeros de aula por haber compartido estos 5 años de estudio.

A mi tutor Roberlán por la preocupación en cada paso de la tesis.

A todos los que de alguna manera han colaborado con la realización de este trabajo.

Aylin.

Agradezco:

A la revolución y a Fidel por darme la oportunidad de estudiar en una universidad de excelencia.

A mis amistades que durante la elaboración de este trabajo brindaron su apoyo, sin su solidaridad no hubiera sido posible su culminación a tiempo; les agradezco por su ayuda incondicional, especialmente al compañero Yixander Yero por su esfuerzo.

A mis padres por tratar de ayudarme en el transcurso de la carrera, familiares, profesores y amistades que me han apoyado con honradez y desinterés.

Lester.

DEDICATORIA

A mi familia toda por esperar siempre lo mejor de mí, especialmente a:

Mis padres Maritza y Edel por su entrega de toda la vida.

Mi abuela Milagros por toda su devoción y tantas noches de sueño le he robado.

Aylín.

A mi tía Cachá por servirme de ejemplo en mi consagrada labor como estudiante revolucionario.

Lester.

RESUMEN

En la Universidad de las Ciencias Informáticas diariamente ocurren numerosos accesos. Llevar el control de los mismos se hace muy difícil, debido a que el número de personas que frecuenta la universidad es muy grande. En la actualidad toda la información relacionada con los accesos es enviada al Oficial Operativo del Puesto de Mando (PM) UCI, quien es el encargado de elaborar un informe diario para hacerlo llegar al Especialista de Seguridad y Protección, este a su vez conforma el Reporte de Accesos que le es solicitado por el Jefe de Seguridad y Protección en la UCI. Existen diversas dificultades las cuales se manifiestan de forma negativa a la hora del trabajo con los accesos, además de que van en contra del correcto funcionamiento de la Universidad. Dentro de estos problemas, se pueden citar: el número de accesos que tienen lugar constantemente solo son chequeados por aproximación visual, dificultad para conocer si una determinada persona ha incurrido en algún acceso indebido, dificultad para encontrar datos estadísticos de los accesos debido a que ningún proceso está automatizado, entre otros. El Sistema de Reportes de Accesos de la Universidad de las Ciencias Informáticas surge por la necesidad que tiene la universidad de automatizar dichos procesos. Este aportará beneficios como: agilizar el desarrollo de los procesos de control de accesos, posibilitar la fiabilidad de la información, ya que la misma se encontrará digitalizada en la base de datos del sistema y mejorar el control de la seguridad y protección en la Universidad de las Ciencias Informáticas.

Palabras clave: reportes, accesos.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción.....	5
1.2 Sistemas Similares.....	5
1.2.1 Sistemas similares en la Universidad.....	8
1.3 Sistema de Gestión de Contenidos (CMS).....	8
1.3.1 Drupal.....	8
1.4 Lenguaje de programación.....	9
1.4.1 ASP.....	9
1.4.2 PERL.....	10
1.4.3 PHP.....	11
1.5 Sistema gestor de Base de Datos (SGBD).....	12
1.6 Entorno de Desarrollo Integrado (IDE).....	14
1.7 Servidor de Aplicaciones.....	14
1.8 Metodologías de desarrollo de software.....	15
1.8.1 El Proceso Unificado de Rational (RUP).....	15
1.8.2 Programación Extrema.....	16
1.8.3 Desarrollo Guiado por la Funcionalidad (FDD).....	17
1.9 Lenguaje de Modelado.....	18
1.9.1 UML (Unified Modeling Language).....	18
1.10 Herramientas CASE.....	19
1.10.1 Rational Rose.....	19

1.10.2 Visual Paradigm for UML	20
1.11 Conclusiones.....	21
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA	22
2.1 Introducción.....	22
2.2 Descripción de los procesos del negocio.....	22
2.3 Objeto de automatización	22
2.4 Descripción general de la propuesta de sistema.	23
2.5 Modelo de Negocio.....	23
2.5.1 Definición de actores del negocio.....	24
2.5.2 Definición de trabajadores del negocio.....	25
2.6 Especificación de los requisitos de software.....	27
2.6.1 Requisitos funcionales	27
2.6.2 Requisitos no funcionales	28
2.7 Modelo del sistema. Definición de actores y casos de uso del sistema	29
2.7.1 Definición de los actores del sistema	29
2.7.2 Definición de casos de uso del sistema.....	30
2.8 Distribución de casos de uso por ciclos de desarrollo.....	31
2.9 Descripción expandida de los casos de uso del sistema	32
2.10 Conclusiones.....	36
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA	37
3.1 Introducción.....	37
3.2 Arquitectura del Sistema.....	37
3.3 Análisis.....	41

3.4 Diseño	43
3.4.1 Diagrama de Clases de Diseño.....	44
3.4.2 Diagramas de Secuencia.	45
3.4.3 Diseño de la base de datos.....	47
3.4.4 Descripción de las tablas del modelo de datos.....	48
3.5 Conclusiones.....	52
CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA	53
4.1 Introducción.....	53
4.2 Modelo de despliegue	53
4.3 Diagrama de Componentes.....	54
4.4 Tratamiento de errores	55
4.5 Seguridad.....	55
4.6 Prueba.....	55
4.6.1 Pruebas de Caja Negra.....	55
4.6.2 Casos de Prueba	56
4.6.2.1 Caso de Prueba para el Caso de Uso: Realizar Búsqueda.	56
4.7 Análisis de costos y beneficios del proyecto.	58
4.8 Conclusiones.....	63
CONCLUSIONES	64
RECOMENDACIONES.....	65
REFERENCIAS BIBLIOGRÁFICAS.....	66
BIBLIOGRAFÍA.....	68
GLOSARIO DE TÉRMINOS	69

ÍNDICE DE FIGURAS

FIGURA 2.1 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.	24
FIGURA 2.2 DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “ELABORAR INFORME”	25
FIGURA 2.3 DIAGRAMA DE ACTIVIDADES DEL CASO DE USO “GENERAR REPORTE DE ACCESOS”	26
FIGURA 2.4 MODELO DE OBJETOS DEL NEGOCIO.	27
FIGURA 2.5 DIAGRAMA DE CASOS DE USO DEL SISTEMA	31
FIGURA 2.6. PAQUETE DE CASOS DE USO A DESARROLLAR.	31
FIGURA 3.1. REPRESENTACIÓN DEL PATRÓN MVC.	38
FIGURA 3.2. REPRESENTACIÓN DEL PATRÓN MVC EN DRUPAL.	39
FIGURA 3.4. REPRESENTACIÓN DE LA INTEGRACIÓN DE MVC Y 3 CAPAS.	41
FIGURA 3.5. DIAGRAMA DE CLASES DEL ANÁLISIS.	43
FIGURA 3.6. DIAGRAMA DE CLASES DE DISEÑO.	45
FIGURA 3.7 DIAGRAMA DE SECUENCIA DEL CASO DE USO “REALIZAR BÚSQUEDA”.	46
FIGURA 3.8 DIAGRAMA DE SECUENCIA DEL CASO DE USO “GENERAR REPORTE DE ACCESOS”.46	
FIGURA. 4.1 DIAGRAMA DE DESPLIEGUE.	53
FIGURA. 4.2 DIAGRAMA DE COMPONENTES.	54

INTRODUCCIÓN

Una gran cantidad de entradas y salidas, tanto de trabajadores como de estudiantes, tienen lugar diariamente en la Universidad de Ciencias Informáticas (UCI), donde es una tarea engorrosa gestionar el movimiento de todo ese personal. Aunque el centro requiere de un control estricto de la seguridad y protección física, el control de entrada y salida del personal se realiza de forma manual, dando lugar a la siguiente **situación problemática**: por no existir un sistema capaz de gestionar eficientemente todo el flujo de información sobre accesos, estos no son controlados como se debe, propiciando que entren personas no autorizadas al centro entre otras infracciones.

Anteriormente se puso en práctica un Sistema de Control de Accesos desarrollado en .NET que actualmente se está migrando a software libre. Teniendo en cuenta que la Base de Datos está desarrollada en Microsoft SQL Server 2000, y que no hay una forma de mostrar los reportes de accesos con los datos que almacenará el sistema, se hace necesaria la migración de la Base de Datos para lograr la migración total del sistema y desarrollar una aplicación que brinde distintos servicios de reportes a los directivos de Seguridad y Protección, permitiéndoles tener un control sistemático del flujo de entrada y salida de personal a la Universidad de las Ciencias Informáticas.

Por lo antes planteado se tiene el siguiente **problema científico**: ¿Cómo generar reportes para mejorar el proceso del control de la seguridad y protección en la Universidad de las Ciencias Informáticas?

El **objetivo general** que persigue el trabajo es desarrollar una aplicación que brinde distintos servicios de reportes, permitiendo la automatización del control de accesos a la Universidad de las Ciencias Informáticas.

De aquí se derivan los siguientes **objetivos específicos**:

- Definir los requisitos que debe cumplir el sistema.
- Diseñar el sistema informático.
- Implementar el sistema para generar reportes.

El **objeto de estudio** de este trabajo es el control de la seguridad y protección en la Universidad de Ciencias Informáticas.

El **campo de acción** queda enmarcado en el proceso de gestión de reportes de la información resultante del Sistema de Control de Accesos de la Universidad de las Ciencias Informáticas.

Esto hace plantear la siguiente **hipótesis**: Con el desarrollo del sistema de gestión de reportes es posible lograr con rapidez y seguridad el control de accesos a la Universidad de las Ciencias Informáticas.

A partir de la hipótesis se definen las siguientes variables:

Variable independiente: Sistema de gestión de reportes

Variable dependiente: Rapidez y seguridad de la generación de reportes.

Para cumplir los objetivos trazados, se definieron las siguientes **tareas científicas**:

- Realización de un estudio del entorno de trabajo.
- Identificación de las necesidades de la institución.
- Definición de los requisitos que debe cumplir el sistema.
- Descripción de los procesos que se van a implementar en el sistema.
- Selección de las herramientas apropiadas para el análisis, diseño e implementación del sistema informático.

informático.

- Modelación conceptual de las clases que están implicadas en el sistema.
- Desarrollo de los diagramas de actividad.
- Desarrollo de los diagramas que describen el diseño del sistema.
- Descripción de las clases del diseño.
- Migración de la Base de Datos.
- Diseño de las consultas SQL para realizar los reportes.
- Diseño de la interfaz.
- Implementación de la aplicación.

Los métodos científicos que se utilizan en la realización de este trabajo, permiten estudiar las características del objeto de investigación.

➤ **Métodos Teóricos**

- El método Analítico - Sintético facilitó el procesamiento de la información obtenida en la investigación realizada sobre el objeto de estudio.
- El método Análisis histórico Lógico posibilitó el conocimiento del estado de arte del objeto de estudio, así como su evolución y desarrollo.

➤ **Métodos Empíricos**

- El método Observación proporcionó una visión realista y objetiva de los procesos, conceptos y definiciones presentes en el negocio.
- El método Entrevista permitió obtener la información necesaria relacionada con los problemas presentes en los procesos de seguridad y control, mediante entrevistas realizadas a personas que trabajan en la Oficina de Seguridad y Protección específicamente al Especialista de Seguridad y Protección.

Estructuración del contenido

El contenido de este trabajo está distribuido por capítulos, los que a su vez están distribuidos por temáticas. El informe cuenta con cuatro capítulos de los cuales se describe de forma general su contenido:

Capítulo 1: Fundamentación Teórica.

En este capítulo se realiza un estudio sobre los lenguajes de programación, herramientas, gestor de base de datos, metodología de desarrollo de software y lenguaje de modelado que se usarán durante el desarrollo del software.

Capítulo 2: Características del sistema.

En este capítulo se aborda acerca de cómo debe funcionar el sistema, haciendo una descripción general de la propuesta. Se elabora una lista de requisitos funcionales y no funcionales que se deben tener en cuenta para la realización de la herramienta, así como la identificación de los casos de usos del sistema. Además, se realiza la descripción de los procesos que serán objeto de automatización.

Capítulo 3: Análisis y Diseño del sistema.

En este capítulo mediante el diseño se hace una especificación de los requisitos con el propósito de describir como se debe implementar el sistema. Se realizan los diagramas de clases del diseño y diagramas de interacción más relevantes.

Capítulo 4: Implementación y Prueba.

En este capítulo se realiza el diagrama de componentes y se muestra la ubicación física de los nodos de procesamiento a través del diagrama de despliegue. Se realizan pruebas de caja negra a la herramienta para comprobar su correcto funcionamiento.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se describen los fundamentos teóricos de los procesos que se desean automatizar, atendiendo a las tendencias actuales. Se informa sobre la descripción de los principales conceptos asociados al problema que son necesarios para entender el negocio y la propuesta de solución. Se realiza un estudio sobre los sistemas existentes que gestionan reportes y sobre las distintas herramientas que se utilizan en el diseño e implementación de este tipo de aplicaciones. Se justifica la selección de las herramientas, metodologías y lenguajes con las que se elaborará el trabajo.

1.2 Sistemas Similares

La gestión de la información es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma. [1]

Los generadores de reportes son herramientas complementarias de los sistemas de información. Utilizan una especie de lenguaje transparente para el usuario por medio del cual este realiza consultas a la base de datos y obtiene información de ella en forma de reporte.

Permiten a los usuarios obtener con facilidad (pero no dar entrada o modificar) datos de archivos o bases de datos. Se puede obtener el contenido parcial o total de los registros. Pueden manejar datos de cálculos y lógica compleja antes de darles la salida. [2]

Los lenguajes de consulta facilitan la recuperación de datos almacenados sin necesidad de escribir muchas instrucciones orientadas hacia el procedimiento, o especificar el formato de los datos. Permiten al usuario formular preguntas -consultas- en formatos tabulares o parecidos al inglés.

Los sistemas Web, son las aplicaciones a las que los usuarios tienen acceso mediante un servidor Web a través de una intranet o Internet utilizando un navegador.

Un sistema web de gestión de reportes es un conjunto de procesos automatizados que se ejecutan sobre una plataforma digital, orientados a obtener información almacenada en bases de datos, de una determinada entidad, en forma de reporte.

Dichos sistemas contribuyen a que exista una mejor comunicación entre los usuarios y la información. Permiten que el cliente acceda a los datos rápidamente. Brindan privacidad en el acceso, ya que al sistema solo pueden acceder aquellas personas que están autorizadas a trabajar con dicha información, en este caso el Especialista de Seguridad y Protección.

Los sistemas informáticos brindan la posibilidad de gestionar cualquier tipo de información, de manera rápida, eficiente y confiable.

Con la implementación de un sistema para gestionar los reportes de accesos, podrá consultarse la información almacenada en la base de datos desde cualquier ordenador autorizado y con acceso a la red de la universidad. Se podrá manejar de forma clara, sencilla y ordenada, toda la información relacionada con los accesos que tienen lugar en la misma; permitiendo así tener un control riguroso del flujo de entrada y salida del personal, tiempo de estancia, cantidad de accesos, etc.

Diversas son las empresas en el mundo que cuentan con un sistema de gestión de reportes. Infogestión es un sistema de reportes estadísticos de la gestión del SIBUDEC (Sistema de Bibliotecas de la Universidad de Concepción), en Chile. Su principal característica es la generación dinámica y en línea de informes a partir de la definición de sentencias en Lenguaje de Consulta Estándar (SQL) sobre la base de datos corporativos de la Universidad de Concepción. El desarrollo de Infogestión permitió al Sistema de Bibliotecas dar apoyo a la gestión institucional, administrar eficientemente las 12 bibliotecas del sistema, facilitar al personal del SIBUDEC la obtención de reportes estadísticos y contar con un sistema de reportes actualizado diariamente. [3]

El Instituto politécnico comunitario de Nueva Escocia (NSCC, por Nova Scotia Community College) de Canadá, convertido en una institución de administración independiente, implementó un sistema de reportes financieros para llevar a cabo la gestión de sus actividades.

Cuando los usuarios finales pasaron de distribuir los reportes físicamente a poder acceder a ellos (a través de Internet o mediante su distribución por correo electrónico), adquirieron la libertad para usar y manipular los datos. Los datos se generaban justo al terminar el mes, por lo tanto, los usuarios podían acceder a los reportes que deseaban al cerrar el periodo financiero.

El ambiente de generación de reportes de Crystal ha progresado y ahora incluye la simplificación de recursos humanos y la gestión de la administración estudiantil. El NSCC cuenta con más de 1.200 empleados y 25.000 estudiantes, por lo tanto, la generación de reportes empresariales se ha convertido en la norma. El departamento de finanzas sigue siendo quien tiene mejor conocimiento y experiencia con Crystal Reports, y esto se traduce en una mayor adopción y una aceptación más general. [4]

También existen algunos sistemas de control de acceso que tienen incluido un módulo de reportes, por ejemplo:

En Argentina, la empresa IDENTIFIC-AR desarrolló un sistema de control de acceso por medio de huella dactilar combinado con un software para control de horarios de nómina o personal. Este Software, posee un módulo de reportes muy eficiente que analiza automáticamente los ingresos y de acuerdo con los distintos horarios asignados a cada empleado permite conocer las ausencias, incumplimientos de horarios, errores en la cantidad de registros y estancia, entre otros. Además cuenta con la posibilidad de exportar los datos almacenados en una base de datos de Microsoft Access a archivos de texto separados por comas o por tabulaciones, permitiendo de esta forma interactuar con cualquier sistema de liquidación de sueldos y/o jornales. [5]

RC Soluciones Integrales, empresa diseñadora y desarrolladora de soluciones tecnológicas, ubicada en Pereira/Risaralda – Colombia desarrolló Ipass, una poderosa herramienta de control de acceso y asistencia, control de activos (herramienta, equipos, etc.), además de informar el estado del personal en los distintos departamentos de su compañía. [6] Además de las diversas herramientas integradas que posee y las opciones que brinda, entre los módulos con que cuenta se encuentra el generador de reportes. Este módulo permite la generación de información útil para la administración y seguridad de la compañía, reportes de cantidades de visitas, frecuencias de visitas, seguimiento de visitantes, recintos más visitados, horas y fechas de entrada y salida, etc.

1.2.1 Sistemas similares en la Universidad.

En la universidad existe un Sistema de Gestión de Reportes Dinámicos que ofrece una solución de reporte que potencia una redituable inteligencia de negocios, destinada a mejorar la rapidez y calidad de la adopción de decisiones en todos los niveles de una organización. Permite a las organizaciones encargarse de este proceso y les proporciona a los desarrolladores herramientas para confeccionar reportes e implementar soluciones personalizadas de informes para usuarios individuales en toda una organización.

Constituye una de las herramientas que posibilitan la inteligencia de negocios ya que genera vistas agregadas de datos para mantener a la gerencia informada sobre el estado de su negocio, además de que proporciona soporte a todo el ciclo de vida de los reportes, provee accesibilidad a la información emitiendo reportes a través de un navegador Web u otros formatos estándares y está orientada al usuario final, brindando herramientas como el Diseñador de Reportes que permite la confección de informes de forma interactiva en la web.

1.3 Sistema de Gestión de Contenidos (CMS)

Un sistema de gestión de contenidos (en inglés *content management system*, abreviado CMS) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos, principalmente en páginas web. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio y pueden estar perfectamente separadas del diseño.

1.3.1 Drupal

Drupal es un CMS modular muy configurable, de código abierto, con licencia GNU/GPL, y por lo tanto es software libre, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Se puede extender o adaptar según las necesidades de cada cliente. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la Web y un énfasis especial en la usabilidad y consistencia de todo el sistema. El contenido puede ser tratado individualmente de la presentación si así el usuario lo desea. Cuenta con páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores. Usa el *mod_rewrite* de Apache para crear *URLs* (Localizador Uniforme de Recursos) que son manejables por los usuarios y los motores de búsqueda. [7]

Puede trabajar con Apache o Microsoft IIS como servidor Web y en sistemas operativos como Linux, BSD, Solaris, Windows y Mac OS X. La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional. Toda la actividad y los sucesos del sistema se guardan en un 'registro de eventos', que puede ser visualizado por un administrador. Incluye un mecanismo de control de congestión, que permite habilitar y deshabilitar determinados módulos dependiendo de la carga del servidor. Tiene un mecanismo de caché que permite eliminar consultas a la base de datos, incrementando el rendimiento y reduciendo la carga del servidor. El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP. Por las razones expuestas se decidió utilizar Drupal para el desarrollo de la aplicación.

1.4 Lenguaje de programación

La versatilidad de un lenguaje está íntimamente relacionada con su complejidad. Es por ello que a la hora de elegir el lenguaje que queremos utilizar tenemos que saber claramente qué es lo que queremos hacer y si el lenguaje en cuestión nos lo permite o no.

Los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP y PERL. [8] A continuación se destacan características propias de cada uno de ellos.

1.4.1 ASP

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitios web dinámicos. ASP significa en inglés (Active Server Pages), fue liberado por Microsoft en 1996. Para las páginas web desarrolladas bajo este lenguaje es necesario tener instalado Internet Information Server (IIS).

ASP no necesita ser compilado para ejecutarse. Para escribir páginas ASP se utiliza un lenguaje de scripts, que se coloca en la misma página web junto con el código HTML. Comúnmente este lenguaje de scripts es Visual Basic Script, que deriva del conocido Visual Basic, aunque también se pueden escribir los scripts ASP en otro lenguaje: JScript, que deriva a su vez del conocido Javascript. [8]

Con ASP se pueden realizar fácilmente páginas de consulta de bases de datos, funciones sencillas como obtener la fecha y la hora actual del sistema servidor, cálculos matemáticos simples, etc.

Ventajas:

- Usa Visual Basic Script, siendo fácil para los usuarios.
- Comunicación óptima a motores de bases de datos SQL, Access, Oracle, y a cualquier otro con soporte de conexión ODBC.
- Soporta el lenguaje JScript (Javascript de Microsoft).

Desventajas:

- Código desorganizado.
- Tecnología propietaria.
- Hospedaje de sitios web costosos.

1.4.2 PERL

Perl es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo web, programación en red, desarrollo de GUI y más.

Muchas de las características de este lenguaje están basadas en otros lenguajes como C/C++, el lenguaje interpretado *shell*, AWK, Lisp, entre otros. Su estructura está formada principalmente por diferentes tipos de datos, estructuras de control y subrutinas. [9]

Perl es software libre (bajo licencia GNU y licencia artística), lo que significa que no es necesario pagar para obtenerlo. Además, existen distribuciones y adaptaciones para gran cantidad de sistemas operativos como Linux, Unix, Windows, Mac, etc.

La principal desventaja de Perl se encuentra en el tiempo de ejecución de un programa, ya que un programa Perl es compilado cada vez que se ejecuta, por lo que puede resultar más lento que un programa similar escrito en otro lenguaje. [9]

Se creó con la idea de que fuera práctico (facilidad de uso, eficiente, completo) en lugar de hermoso (pequeño, elegante, mínimo). Sus principales características son que es fácil de usar, soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional, tiene

incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles.

1.4.3 PHP

PHP es un lenguaje muy potente, mundialmente utilizado en la programación de aplicaciones Web del lado del servidor. Es similar a otros lenguajes como C o C++, rápido, de sintaxis cómoda y su sencillez contribuye a su rápido aprendizaje. Sus primeras versiones, PHP 3 y PHP 4, sirvieron de gran ayuda a la comunidad de desarrolladores, para convertir a PHP en el lenguaje más utilizado en la construcción de páginas Web dinámicas.

Es un lenguaje interpretado, multiplataforma, se le pueden agregar extensiones fácilmente y dispone de una gran cantidad de librerías. Se puede utilizar como módulo de Apache, lo que lo hace extremadamente veloz. Por estar completamente escrito en C, se ejecuta rápidamente utilizando poca memoria. No hay que pagar licencias, no limita su distribución y se puede ampliar con nuevas funcionalidades si se desea. [10]

PHP 5 es la versión más adelantada, completa e intuitiva. El código es fácil de entender y mantener. Posee una vasta gama de funciones que le permiten adaptarse a cualquier entorno y a cualquier sistema operativo, por lo que es más eficiente.

El objetivo principal de PHP 5 es perfeccionar el lenguaje, utilizando los mecanismos de programación orientada a objetos (POO) que no estaban presentes en las versiones anteriores, parámetro necesario e imprescindible que convierte a PHP en un lenguaje idóneo para todo tipo de aplicaciones y entornos. Es un lenguaje muy asequible, tanto para aquellas personas que tienen experiencia en la programación de sistemas web dinámicos, como para los que no.

Lenguaje seleccionado a utilizar

PHP unido al servidor web Apache, representan las dos herramientas claves para la creación de este tipo de aplicaciones. Las razones antes expuestas además de su excelente forma de trabajar con casi todas las bases de datos, y en especial, con MySQL, hacen que PHP sea el lenguaje perfecto para implementar el sistema de reportes de accesos.

1.5 Sistema gestor de Base de Datos (SGBD)

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información. Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de una organización, para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas.[11]

Todos los sistemas automáticos para la gestión del mantenimiento emplean base de datos, estas son el soporte para almacenar toda la información que manejan. Deben presentar unicidad, consistencia, seguridad e integridad. Existen varios sistemas gestores de base de datos, Access, Oracle, PostgreSQL, SQL, y MySQL son algunos de ellos [11].

Drupal utiliza como gestor de base de datos mysql y postgres, a continuación se muestran características propias de cada uno de ellos.

➤ **MySQL** implementa funcionalidades Web que permiten un acceso a los datos, seguro y fácil. Es uno de los SGBD más populares, desarrollado bajo la filosofía de código abierto.

La licencia GPL de MySQL obliga a distribuir cualquier producto derivado (aplicación) bajo esa misma licencia. Por tanto MySQL tiene sus restricciones, sólo es gratis si se está dispuesto a distribuir la aplicación que se quiere desarrollar bajo esa misma licencia GPL.

MySQL tiene como una de sus principales ventajas la velocidad en la lectura de datos. Se adapta fácilmente a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl, Java y su integración en los diferentes sistemas operativos. Utiliza múltiples tablas para almacenar y organizar la información. Cuenta con una gran cantidad de herramientas que administran cómodamente la base de datos. En las últimas versiones se incorporaron algunas transacciones de las que se carecía al inicio.

Es muy destacable su condición de código abierto, de utilización gratuita, que puede modificarse con total libertad con solo descargar su código fuente.

MySQL se integra perfectamente con Apache como servidor de aplicaciones y PHP como lenguaje de programación Web. La combinación de estos tres elementos es considerada por muchos desarrolladores como la mejor integración para el desarrollo web.

En el caso del CMS Drupal, existe una tendencia extremadamente marcada a desarrollar módulos sobre MySQL.

➤ **PostgreSQL** está considerado como el SGBD de código abierto más avanzado del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales de alto calibre tales como DB2 u Oracle. [12]

Es un SGBD objeto-relacional, aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas, el control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y valores no atómicos (atributos basados en vectores y conjuntos).

Es altamente extensible, soporta operadores y tipos de datos definidos por el usuario. Incluye características avanzadas tales como las uniones (joins) SQL92. Cuenta con una API (del inglés Application Program Interface) flexible lo cual ha permitido dar soporte para el desarrollo con PostgreSQL en diversos lenguajes de programación entre los que se incluyen: Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike. Tiene soporte para lenguajes procedurales internos, incluido un lenguaje nativo denominado PL/pgSQL, el cual es comparable con el lenguaje procedural de Oracle PL/SQL [11].

SGBD seleccionado a utilizar.

Tomando en consideración que en las aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos y que lo que se necesita es un sistema rápido y con pocos usuarios concurrentes, se elige a MySQL como SGBD por reunir las características que más se ajustan al sistema que se quiere implementar.

1.6 Entorno de Desarrollo Integrado (IDE)

Un IDE es un programa que integra varias herramientas que facilitan el desarrollo de software sobre uno o varios lenguajes de programación. Sus características más comunes son: editor de código, herramientas para traceo, consulta a bases de datos, depurador de código y construcción de interfaz gráfica. Entre los IDEs más reconocidos mundialmente se encuentran: Visual Studio .NET, Java para Eclipse y Borland C++. Para desarrollo Web figuran Aptana, Dreamweaver, Zend Studio, entre otros. Estos últimos son herramientas poderosas para el desarrollo de software sobre PHP.

Se decidió optar por Zend Studio para el desarrollo de este trabajo, debido a que integra lo mejor de las prácticas actuales para desarrollo web, permite la integración con Eclipse y simplifica en gran medida las tareas de integración. Posee herramientas para editar, analizar, optimizar y depurar el código generado. Permite trabajar con bases de datos y otros lenguajes relacionados como *JavaScript*, HTML y XML.

1.7 Servidor de Aplicaciones

Se denomina servidor de aplicaciones a un software que proporciona aplicaciones a los equipos o dispositivos cliente, por lo general a través de Internet y utilizando el protocolo *Hyper Text Transfer Protocol* (HTTP). Se distinguen de los servidores Web por el uso extensivo del contenido dinámico y por su frecuente integración con bases de datos. La ventaja principal de un servidor de aplicaciones es la facilidad para desarrollarlas, puesto que estas no necesitan ser programadas y en cambio, se arman a partir de módulos provistos por el servidor de aplicaciones. [13]

El servidor Apache inicialmente se basó en el servidor desarrollado en la *National Center for Supercomputing Applications* (NCSA), al que se le añadieron algunos parches. Estos arreglos hicieron de Apache un servidor más competente que el de la NCSA en menos de un año.

Es un software libre que se desarrolla dentro del proyecto *HTTP Server* de la *Apache Software Foundation*, cuya función principal es atender las peticiones de los clientes y enviarles las páginas Web solicitadas. Presenta entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Es el servidor más empleado en el mundo, actualmente su porcentaje de utilización supera el 60%. Sus ventajas más significativas son:

- Modular

- Código abierto
- Multiplataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte) [14].

Debido a que PHP se puede utilizar como módulo de Apache, lo que hace que la velocidad de las páginas sea mucho mayor, se decidió utilizar Apache como servidor de aplicaciones.

1.8 Metodologías de desarrollo de software

Uno de los temas más comunes en el mundo de la informática es el de las metodologías de desarrollo de software. Una metodología tiene como objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo, por medio de una mayor transparencia y control sobre el proceso; producir lo esperado en el tiempo esperado y con el coste esperado.

Todo sistema a implementar se rige por una metodología. Esta no es más que un conjunto de pasos a seguir en pos del resultado final. En la esfera del desarrollo de software, se busca obtener un producto de alta calidad, que satisfaga las necesidades del usuario. La metodología en la ingeniería de software sería entonces el conjunto ordenado de pasos a seguir para lograr un software de alta calidad que satisfaga las necesidades del usuario, define quién debe hacer, qué, cómo y cuándo debe hacerlo.

Analizaremos a continuación tres de las más conocidas, sus características, ventajas y desventajas.

1.8.1 El Proceso Unificado de Rational (RUP)

RUP son las siglas en inglés de Rational Unified Process. Esta es una de las metodologías más generales de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto, no sólo de software. Es el producto final de treinta años de desarrollo y uso práctico. Se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como lenguaje de modelado.

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes. Cada uno de estos ciclos se divide en cuatro fases (Inicio,

Elaboración, Construcción y Transición) en ellas se llevan a cabo una o varias iteraciones, cada una de las cuales resulta un incremento de la precedente, por lo que se dice que RUP es iterativo e incremental, cuenta con nueve flujos de trabajo (Modelamiento del negocio, Requisitos, Análisis y diseño, Implementación, Prueba, Instalación, Administración del proyecto, Administración de configuración y cambios y Ambiente) [15].

El ciclo de vida de RUP se caracteriza por ser:

- Dirigido por casos de Uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

1.8.2 Programación Extrema

La Programación Extrema, es otra de las metodologías de desarrollo de software que existen en la actualidad. “Mientras que el RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción” [15]

Supone la disposición en todo momento, por parte del equipo de trabajo, de un representante competente del cliente, que debe estar en condiciones de dar una respuesta rápida y correcta a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones.

La base para el desarrollo del software que usa esta metodología son las llamadas Historias de Usuario (User Stories), historias escritas por el cliente en las que describe escenarios sobre el funcionamiento del sistema. Estas Historias de Usuario junto a la arquitectura que se persigue, sirve de base para crear un plan de “entregas de software” entre el equipo de desarrollo y el cliente. Las Historias de Usuario y los casos de pruebas son la base sobre la que se asienta el trabajo del desarrollador.

Esta metodología apuesta por iteraciones cortas que generan software que el cliente puede ver [15].

El objetivo ideal sería que cada integrante del equipo trabaje al menos una vez con cada uno de los demás integrantes y con cada componente del software, de forma que el conocimiento de la aplicación completa lo posea el equipo entero y no unos pocos miembros [15].

1.8.3 Desarrollo Guiado por la Funcionalidad (FDD)

Se podría considerar que FDD (por las siglas en inglés de Feature Driven Development) es una metodología ligera. Está pensada para proyectos con un tiempo de desarrollo relativamente corto (menos de un año). Se basa en un proceso iterativo con iteraciones cortas de aproximadamente dos semanas que producen un software funcional, el cual puede ser examinado por el cliente y la dirección de la empresa. Cada iteración se define en término de funcionalidades que son pequeñas partes del sistema con significado para el cliente [15].

Un proyecto que siga esta metodología estará dividido en cinco fases: desarrollo de un modelo general, construcción de la lista de funcionalidades, plan de entregas sobre la base de las funcionalidades a implementar, diseño basado en las funcionalidades e implementación basada en las funcionalidades. Todo el trabajo se realiza en grupo, aunque siempre hay un responsable, que generalmente tiene mayor experiencia, que dice la última palabra en caso de no llegar a un acuerdo [15].

Las funcionalidades de cada entrega se dividen entre los distintos subgrupos del equipo y se implementan. El código escrito (las clases) tiene propietario, o sea, sólo quien lo crea puede modificarlo. Es por eso que en un subgrupo deben estar todos los propietarios de las clases implicadas, pudiendo un desarrollador pertenecer a varios subgrupos. También se contemplan como parte del proceso de implementación, la preparación y ejecución de pruebas, las revisiones de código y la integración de las partes que componen el software [15].

Metodología seleccionada a utilizar.

Tomando en cuenta las características de cada metodología posible a emplear para la realización de la aplicación, se selecciona RUP como la más conveniente y eficaz para el desarrollo del producto. Además de ser la más utilizada, se basa en casos de uso para describir lo que se espera del software, se documenta lo mejor posible, basándose en UML como lenguaje de modelado y está pensado para adaptarse a cualquier proyecto.

1.9 Lenguaje de Modelado

El lenguaje de modelado es un conjunto estándar de símbolos y de formas que facilita la modelación de un diseño de software.

1.9.1 UML (Unified Modeling Language)

UML son las siglas de Unified Modeling Language (Lenguaje Unificado de Modelado), notación (esquemática en su mayor parte) con que se construyen sistemas por medio de conceptos orientados a objetos. [16].

El Lenguaje Unificado de Modelado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML permite una forma de modelar conceptualmente procesos de negocio y funciones de sistema, además de elementos concretos como lo son esquemas de base de datos, componentes de software reutilizables y escribir clases en un lenguaje determinado.

Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto. Permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Tiene como objetivo brindar un material de apoyo que le permita al lector poder definir diagramas propios como también entender diagramas ya existentes.

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

1.10 Herramientas CASE

Se puede definir herramienta CASE (*Computer Aided Software Engineering* y en su traducción al español: Ingeniería de Software Asistida por Computadora) a la aplicación de métodos y técnicas a través de las cuales las personas pueden modelar o diseñar sistemas por medio de programas, procedimientos y su respectiva documentación. [17]

Objetivos de las herramientas CASE:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto
- Ayudar a la reutilización del software, portabilidad y estandarización de la documentación
- Gestionar las fases de desarrollo del software con una misma herramienta.

Ejemplos de herramientas CASE:

- Umbrello
- ArgoUML
- Gaphor
- System Architect
- Rational Rose
- Visual Paradigm

1.10.1 Rational Rose

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y elaboración del modelo, construcción de los componentes, transición a los usuarios. Permite especificar, analizar, diseñar el sistema antes de codificarlo. Rational Rose establece una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable; facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero comparten un mismo modelo a lo largo de todo el ciclo de vida del proyecto.

Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Rational Rose utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades. Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML. Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

1.10.2 Visual Paradigm for UML

Visual Paradigm for UML es una potente herramienta que permite visualizar y diseñar elementos de software y soporta el ciclo de vida completo del desarrollo de software. Una de sus ventajas sobre las demás herramientas CASE es condición multiplataforma en su edición Community, por lo que tiene la capacidad de ejecutarse sobre diferentes Sistemas Operativos. Es fácil de usar y presenta una agradable interfaz para interactuar con el usuario. Se integra con el Visio para importar imágenes del mismo para realizar los diagramas de despliegue y genera documentación para el proyecto en HTML, MS Word y PDF. Además, importa y exporta diagramas en XML y como imágenes (ya sea con extensiones jpg o png). Otra característica importante es que puede generar código y realizar ingeniería inversa a diferentes lenguajes de programación como son: Java, C++, CORBA IDL, PHP, XML Schema y ADA.

Facilita la interoperabilidad con otras herramientas CASE como el Rational Rose y se integra con las siguientes herramientas Java [18]:

- Eclipse/IBM WebSphere
- Jbuilder
- NetBeans IDE

- Oracle Jdeveloper
- BEA Weblogic.

Herramienta CASE seleccionada a utilizar

La herramienta CASE seleccionada fue Visual Paradigm for UML ya que permite el diseño del producto de forma rápida y con calidad, nos brinda facilidades para el diseño de los diagramas, así como su documentación, puede generar código y realizar ingeniería inversa a diferentes lenguajes de programación. La Universidad cuenta con la licencia para su uso y además tiene la característica de ser multiplataforma.

1.11 Conclusiones

En la actualidad, existen en el mundo diversos sistemas que automatizan la gestión de reportes. Estos sistemas fueron implementados para satisfacer necesidades propias de la institución o empresa a la que pertenecen, por lo que se decidió realizar un sistema que se ajuste a las necesidades específicas del centro.

Se optó por RUP como metodología de desarrollo, pues además de ser la más utilizada, se basa en casos de uso para describir lo que se espera del software, se basa en UML como lenguaje de modelado y está pensado para adaptarse a cualquier proyecto. Entre las herramientas que se escogieron para el desarrollo del sitio Web se encuentra Drupal, por ser un sistema gestor de contenidos libre que cumple con los estándares internacionales de desarrollo Web. El servidor Apache está ubicado a la vanguardia en aplicaciones desarrolladas con software libre, lo que unido a la utilización de PHP y MySQL, garantizan una codificación robusta y rápida.

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

2.1 Introducción

En este capítulo se describen los procesos del negocio que se relacionan con el área de Seguridad y Protección referentes a la forma de manejar la información de los accesos a la Universidad, se enumeran los requisitos funcionales y no funcionales que debe tener el Sistema de Reportes de Accesos que se propone, obteniendo una concepción general del sistema que permite identificar mediante un Diagrama de Casos de Uso las relaciones de los actores que intervienen en el sistema y la secuencia de acciones con las que interactúan.

2.2 Descripción de los procesos del negocio

Los procesos que intervienen en el Control de Acceso de la UCI ocurren de la siguiente manera:

Una vez que las personas pertenecientes a la UCI, llegan a la garita para entrar a la universidad, se identifican mostrando su credencial, se verifica su validez de manera visual por parte de los agentes de SEPCOM. Si alguna persona no porta su credencial en esos momentos, se hace una comprobación mediante una llamada telefónica al directivo del área a la que pertenece y tras obtener la confirmación se le entrega un autorizo de entrada el cual debe ser firmado por el directivo para poder salir. Debe realizarlo por la misma puerta mostrando dicho autorizo. Para salir también se muestra la credencial y se verifica además si porta algún medio y con la correspondiente autorización para sacarlo de la Universidad.

Para permitirles la entrada a los automóviles se verifica si tienen o no el logotipo de la Universidad y las credenciales de las personas que vienen en ellos. En el caso de la salida se verifica además si porta algún medio y con la correspondiente autorización para sacarlo de la UCI. Toda esta información es enviada al Oficial Operativo del Puesto de Mando (PM) UCI, quien es el encargado de elaborar un informe diario para hacerlo llegar al Especialista de Seguridad y Protección. Este a su vez conforma el Reporte de Accesos que le es solicitado por el Jefe de Seguridad y Protección en la UCI.

2.3 Objeto de automatización

Existen diversas dificultades las cuales se manifiestan de forma negativa a la hora del trabajo con los accesos, además de que van en contra del correcto funcionamiento de la Universidad. Dentro de los problemas, se pueden citar:

Capítulo2: Características del Sistema.

- El número de accesos que tienen lugar constantemente solo son chequeados por aproximación visual.
- Dificultad para conocer si una determinada persona ha incurrido en algún acceso indebido.
- Dificultad para encontrar datos estadísticos de los accesos debido a que ningún proceso está automatizado, estos datos se necesitan para la obtención de reportes.

Sobre la base de estas dificultades será que el Sistema de Reportes de Accesos se realice. Los procesos para el desarrollo del Sistema estarán centralizados en la obtención de reportes sobre la información almacenada de los accesos que ocurren continuamente. Para ello el sistema ha de permitir al usuario especificar diferentes criterios de selección para conformar el reporte, además de poder guardarlo e imprimirlo. Además, como toda aplicación Web que maneja datos importantes, tendrá implementado un sistema de autenticación de usuario, permitiendo esto que tengan acceso a la información solo las personas autorizadas.

Este sistema posibilitará el mejoramiento del control de la seguridad y protección en la Universidad de las Ciencias Informáticas.

2.4 Descripción general de la propuesta de sistema.

Se implementará una aplicación Web para gestionar los datos de los accesos a la Universidad. La misma estará programada en lenguaje PHP y contará con una base de datos en MySQL como ya se mencionó. Se manipulará información referente al personal que accede a la universidad, en qué horario lo hacen, por cuál puerta, con qué frecuencia, la cantidad de accesos en un determinado período, etc. La aplicación debe ser agradable y fácil de usar para el usuario.

2.5 Modelo de Negocio

El Modelo de Negocio es el primero de los nueve flujos de trabajos que integran la metodología de RUP. Es la disciplina que se encarga de comprender la estructura y dinámica de la organización en la cual se va a implantar el sistema, de entender los problemas actuales de la organización e identificar las mejoras potenciales. Asegura que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común del funcionamiento de la organización y deriva los requisitos del sistema que va a soportar la organización.

2.5.1 Definición de actores del negocio

Tabla 1. Definición de actores del negocio.

Actores del negocio	Justificación
Oficial Operativo del Puesto de Mando (PM)UCI	Es el responsable de enviar al Especialista de Seguridad y Protección (SP) la información que se conozca de los accesos para este elaborar un informe con los datos registrados diariamente.
Jefe de Seguridad y Protección en la UCI.	Es quien necesita el reporte que se genere.

La figura 2.1 muestra el diagrama de casos de uso del negocio, en el cual aparecen relacionados los actores del negocio con los casos de uso del mismo.

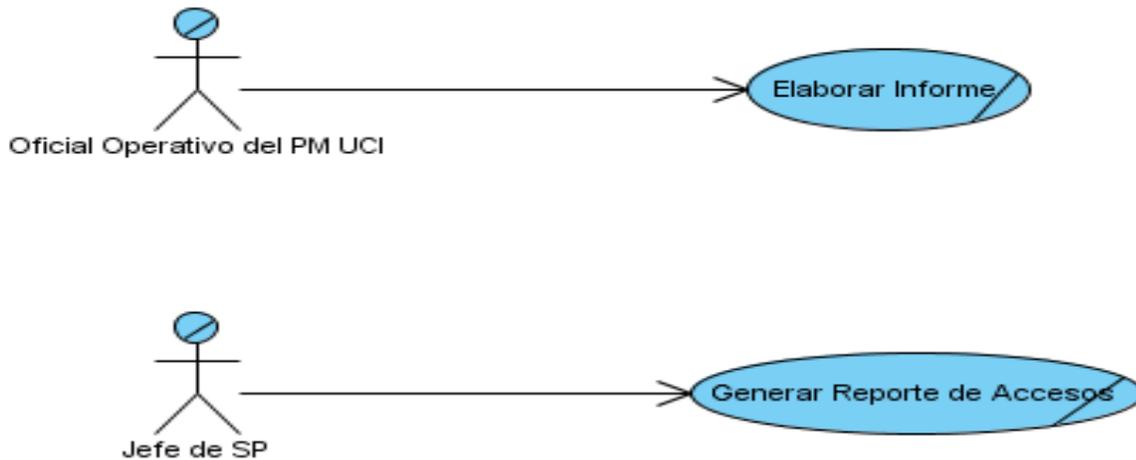


Figura 2.1 Diagrama de casos de uso del negocio.

2.5.2 Definición de trabajadores del negocio

Tabla 2. Definición de trabajadores del negocio

Trabajadores del negocio	Justificación
Especialista de Seguridad y Protección (SP).	Es el responsable de crear y archivar los informes de accesos que le son enviados por el oficial operativo del PM de la UCI, y obtener el reporte con la información solicitada por el Jefe de Seguridad y Protección.

La figura 2.2 y 2.3 muestran los diagramas de actividades para los casos de uso “Elaborar Informe” y “Generar Reporte de Accesos” respectivamente.

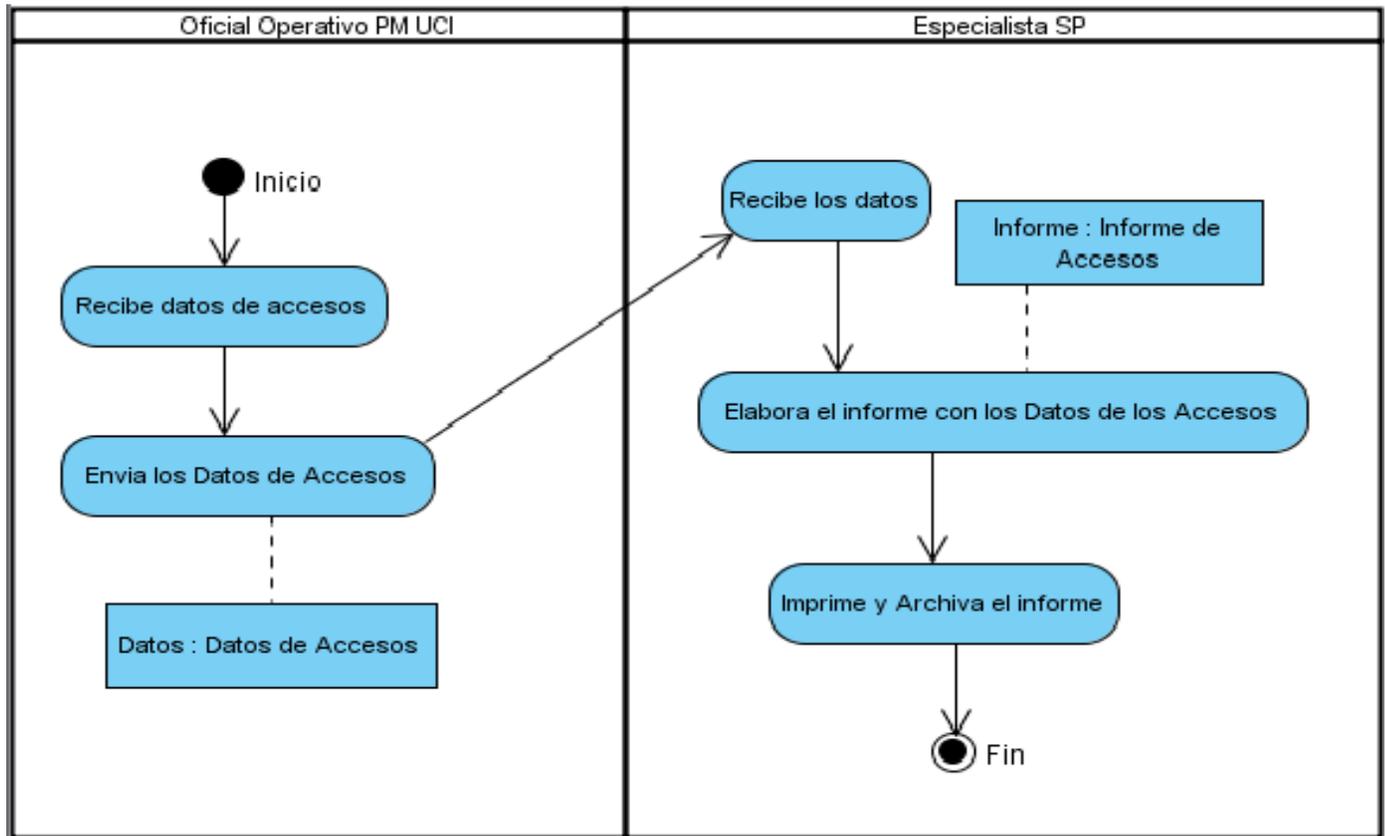


Figura 2.2 Diagrama de Actividades del caso de uso “Elaborar Informe”

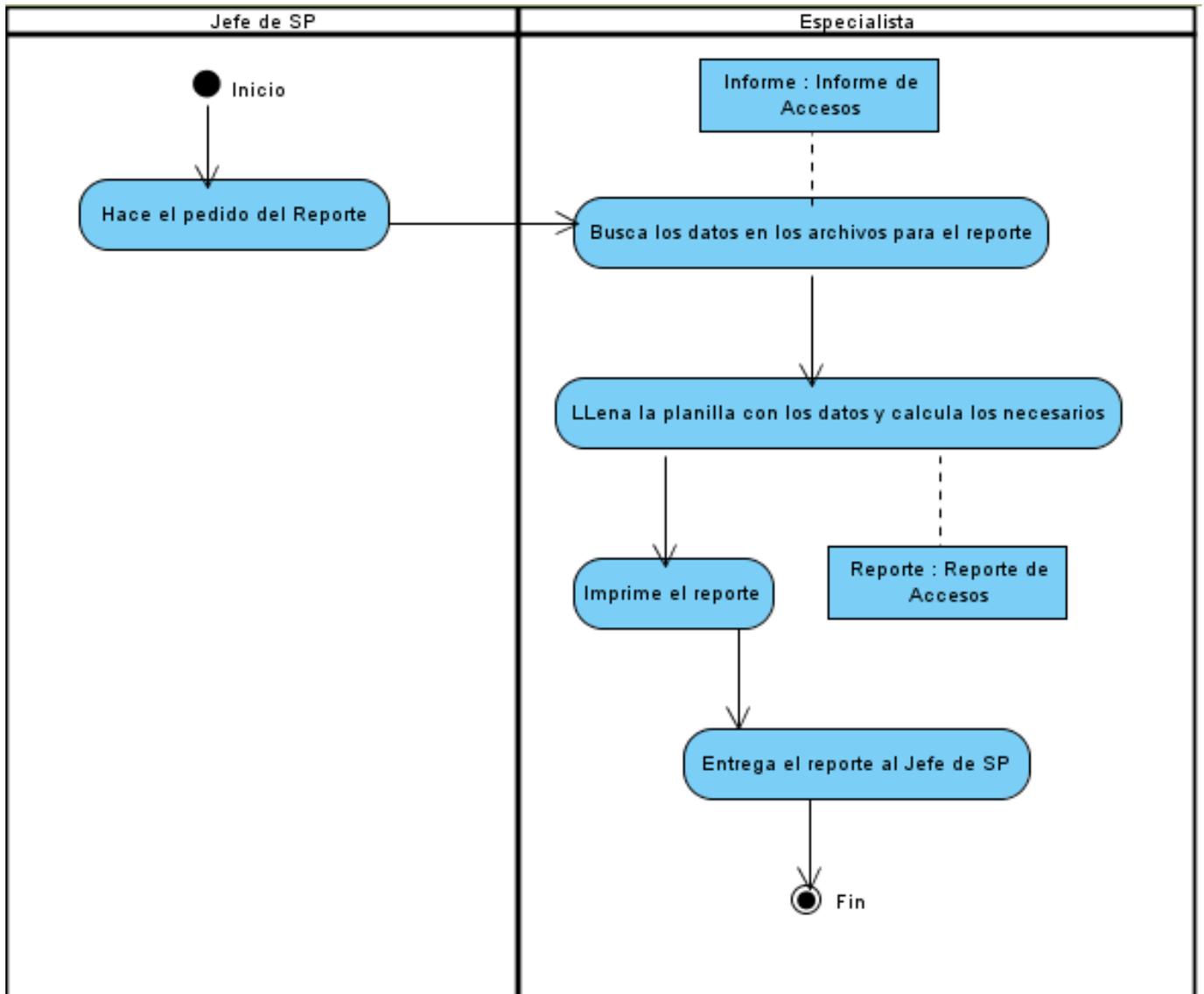


Figura 2.3 Diagrama de Actividades del caso de uso "Generar Reporte de Accesos"

La figura 2.4 muestra el modelo de objetos del negocio.

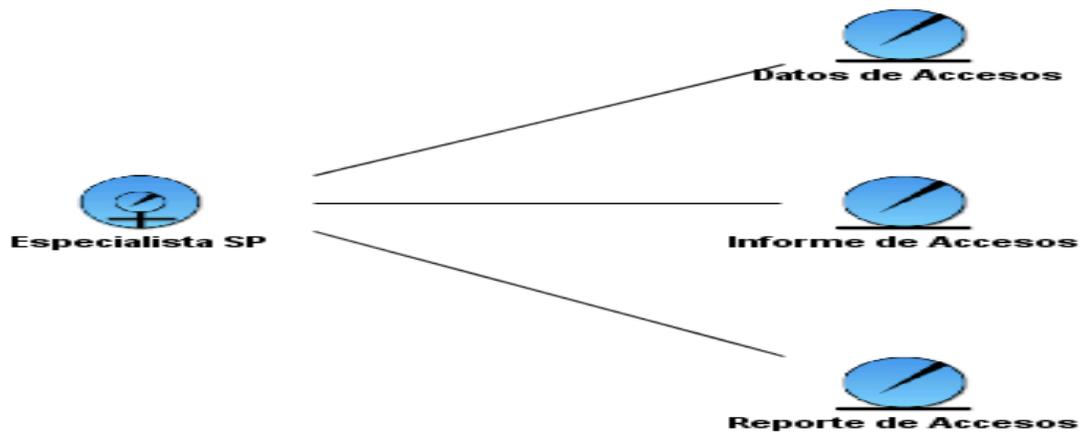


Figura 2.4 Modelo de objetos del negocio.

2.6 Especificación de los requisitos de software

Los requisitos, son condiciones o capacidades que tienen que ser alcanzadas o poseídas por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Son esencialmente las ideas o peticiones que los clientes, usuarios y miembros del equipo de proyecto, tienen o hacen con relación a lo que debe hacer el sistema. Los requisitos se clasifican en: funcionales y no funcionales.

2.6.1 Requisitos funcionales

Los requisitos funcionales representan las funcionalidades del sistema, es decir, son capacidades o condiciones que el sistema debe cumplir. Para este sistema existen las siguientes:

1. Autenticar.
 - 1.1 El sistema debe permitir la autenticación de usuarios por el dominio.
2. Administrar Usuarios.
 - 2.1 Debe permitir al Jefe de Seguridad y Protección:
 - 2.2 Insertar, modificar y elimina usuarios del sistema.
 - 2.3 Asignar roles a los usuarios.
3. Generar reportes de accesos.
 - 3.1 Seleccionar parámetros para generar reportes.

3.2 Visualizar los reportes.

3.3 Guardar los reportes.

3.4 Imprimir los reportes.

2.6.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se podrá determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Para el sistema propuesto se han definido los siguientes requisitos no funcionales:

➤ Requisitos de apariencia o interfaz externa.

- Interfaz con colores poco llamativos, persiguiendo mostrar el tipo de información que gestiona el sistema y el área de la UCI con la que se relaciona (Oficina de Seguridad y Protección).
- Interfaz con un diseño sencillo, con vista a acelerar la velocidad de respuesta hacia el usuario.
- Interfaz debe mostrar información relacionada solamente con las funcionalidades del sistema para no facilitar distracción en el trabajo que se desarrolle.

➤ Requisitos de usabilidad.

- La aplicación tendrá un ambiente sencillo y será de fácil navegación para los usuarios, incluso aquellos que no han tenido mucha experiencia en el trabajo con computadoras o con sistemas informáticos.

➤ Requisitos de rendimiento.

- Estará implementado sobre una tecnología Web, facilitando su uso a través de la red.
- El sistema deberá ser capaz de gestionar toda la información y dar respuesta en el menor tiempo posible.

➤ Requisitos de seguridad.

Capítulo 2: Características del Sistema.

- Solo tendrán acceso al sistema aquellos usuarios que tengan permiso establecido por el Directivo de Seguridad y Protección.
- La autenticación se realizará sobre el dominio UCI.
- Se establecerán los servidores tanto de aplicación como de base de datos en lugares de restringido acceso.

- Requisitos de hardware.
 - Se requiere de una impresora estándar para la impresión de los reportes de accesos.

- Requisitos de software.
 - Apache como servidor de aplicaciones.
 - MySQL como gestor de base de datos.

- Restricciones en el diseño y la implementación.
 - Para separar el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario debe utilizarse el patrón Modelo Vista Controlador.

- Requisitos de portabilidad.
 - El sistema podrá ser ejecutado sobre los sistemas operativos Linux y Windows.

2.7 Modelo del sistema. Definición de actores y casos de uso del sistema

2.7.1 Definición de los actores del sistema

Un actor del sistema son todas aquellas personas o sistemas que intercambien información con la aplicación. Para este trabajo se definen los siguientes actores:

Tabla 3. Definición de actores del sistema.

Actores	Justificación
Directivo de Seguridad y Protección.	Es el responsable de insertar, modificar y eliminar usuarios del sistema, así como asignar roles a los usuarios. También puede seleccionar los parámetros para generar los reportes de accesos, teniendo la posibilidad de guardarlos e imprimirlos.
Usuario.	Es una generalización del Directivo de Seguridad y Protección

2.7.2 Definición de casos de uso del sistema

Tabla 4. Definición del caso de uso Autenticar.

CU-1	Autenticar
Actor	Usuario.
Descripción	El caso de uso comienza cuando el usuario desea entrar al sistema, en dependencia de los permisos establecidos así será el acceso que tenga dentro del mismo. El caso de uso finaliza cuando el usuario logra acceder a la sección del sistema para su rol.
Referencia	Requisito funcional 1.

Tabla 5. Definición del caso de uso Administrar Usuarios.

CU-2	Administrar Usuarios.
Actor	Directivo de Seguridad y Protección
Descripción	El caso de uso comienza cuando el Directivo de Seguridad y Protección selecciona la opción administrar. En esta sección podrá insertar, modificar o eliminar usuarios y asignarle roles.
Referencia	Requisito funcional 2.

Tabla 6. Definición del caso de uso Realizar Búsqueda.

CU-3	Realizar Búsqueda
Actor	Directivo de Seguridad y Protección.
Descripción	El caso de uso inicia cuando el Directivo de SP selecciona la opción de búsqueda, luego entra los parámetros por los cuales realizará la búsqueda, esta se efectúa, el sistema muestra lo encontrado y si el Directivo lo desea puede obtener algún reporte.
Referencia	Requisito funcional 3.

Tabla 7. Definición del caso de uso Generar Reportes de Accesos.

CU-4	Generar Reportes de Accesos
Actor	Usuario
Descripción	El caso de uso inicia cuando el usuario selecciona la opción de realizar el reporte, luego de haber realizado la búsqueda que le permitiera localizar los parámetros sobre los cuales desea

Capítulo 2: Características del Sistema.

	obtener el reporte, se le muestra este último y tiene la opción verlo en formato pdf.
Referencia	Requisito funcional 3.

La figura 2.5 muestra el diagrama de casos de uso del sistema, donde se relacionan los actores del sistema y los casos de uso del mismo.

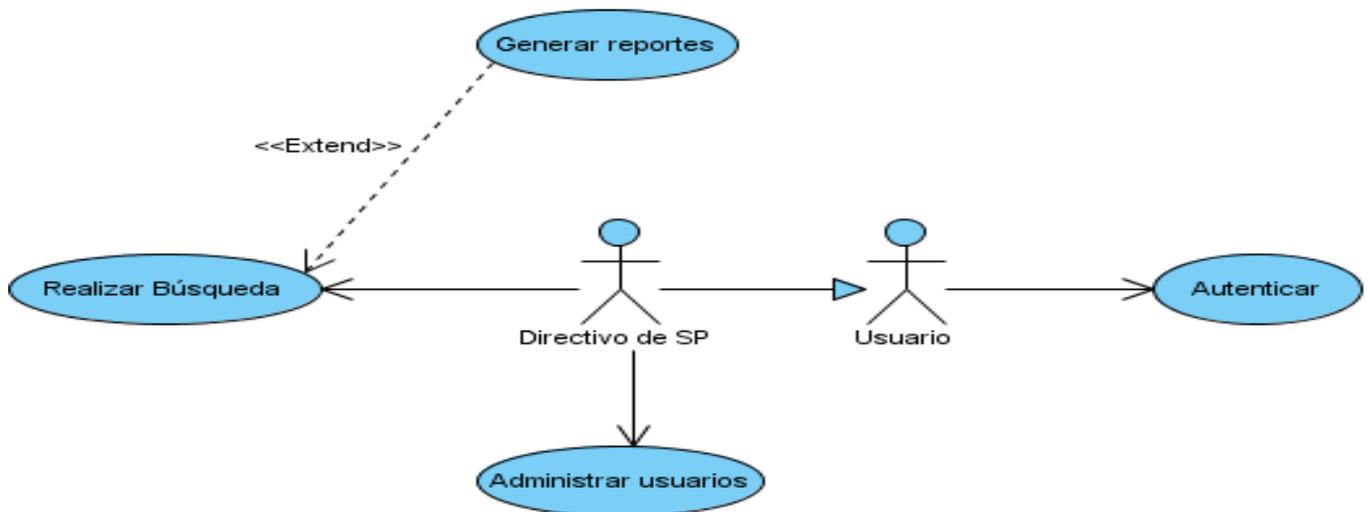


Figura 2.5 Diagrama de casos de uso del sistema

2.8 Distribución de casos de uso por ciclos de desarrollo

Para lograr los objetivos de la aplicación es necesario desarrollar todos los procesos antes planteados. El desarrollo de estos procesos se llevarán hasta la fase de implementación y prueba, quedando conformado de esta forma un producto final. Ver figura 2.6.

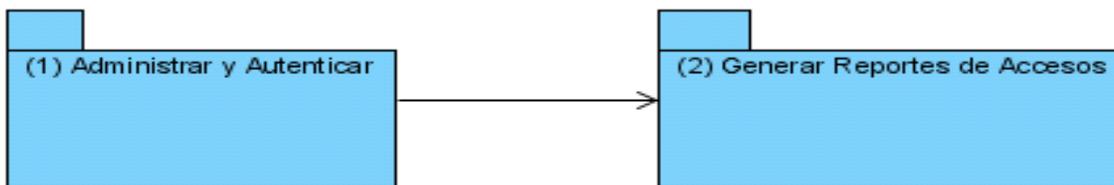


Figura 2.6. Paquete de casos de uso a desarrollar.

Capítulo 2: Características del Sistema.

Tabla 8. Casos de uso a desarrollar en el ciclo de desarrollo.

No. Ciclo	Nombre de caso de uso	Paquete	Justificación de la selección.
1 Módulo Principal	Autenticar. Administrar Usuarios.	1	Estos casos de uso permitirán a los usuarios autenticarse para trabajar en el sistema y al Directivo de Seguridad y Protección poder dar y denegar permisos a los usuarios.
	Realizar Búsqueda Generar Reportes de Accesos	2	Estos casos de uso son de máxima prioridad para la institución, fundamentalmente en la Oficina de Seguridad y Protección para el manejo de la información relacionada con los accesos.

2.9 Descripción expandida de los casos de uso del sistema

Las descripciones de los casos de uso del sistema permitirán conocer las funcionalidades específicas de cada uno de ellos.

Tabla 9. Descripción expandida del caso de uso del sistema Autenticar.

Caso de uso	
CU-1	Autenticar.
Propósito	Permitir acceso al sistema solamente a aquellos usuarios que tengan permiso.
Actores: Usuario (generalización del Directivo de Seguridad y Protección)	
Resumen: El caso de uso comienza cuando el usuario (Directivo de Seguridad y Protección) desea entrar al sistema, en dependencia de los permisos establecidos así será el acceso que tenga dentro del mismo. El caso de uso finaliza cuando el usuario logra acceder a la sección del sistema que le está permitido.	
Referencias	Requisito Funcional 1
Acción del actor	Respuesta del sistema
1. El Usuario accede al sistema para entrar.	2. El sistema muestra la interfaz de entrada donde tendrá que introducir su usuario y contraseña.
3. El Usuario introduce su usuario y contraseña.	

Capítulo 2: Características del Sistema.

4. El Usuario acepta entrar al sistema.	5. El sistema verifica el usuario y contraseña entrada.
	6. El sistema verifica los permisos dentro del sistema para el usuario que solicita la entrada.
	7. El sistema permite la entrada al usuario a la interfaz correspondiente para el tipo de usuario.
Flujos alternativos	
Acción del actor	Respuesta del sistema
4. El usuario cancela entrar a la aplicación.	4.1 Se termina el caso de uso.
	5. Si el sistema verifica que los datos son incorrectos.
	5.1 El sistema envía un mensaje "Datos Incorrectos". Pasa a la acción 2.

Tabla 10. Descripción expandida del caso de uso del sistema Administrar Usuarios.

Caso de uso	
CU-2	Administrar Usuarios.
Propósito	Dar permiso a usuarios para utilizar el sistema.
Actores: Directivo de Seguridad y Protección	
Resumen: El caso de uso comienza cuando el Directivo de Seguridad y Protección selecciona la opción administrar. En esta sección podrá insertar, modificar o eliminar usuarios y asignarle roles.	
Referencias	Requisito Funcional 2.
Acción del actor	Respuesta del sistema
1. El Directivo de Seguridad y Protección después de autenticado selecciona la opción de Administrar Usuarios.	2. El sistema muestra la interfaz para administrar usuarios del sistema. Con las opciones de insertar "Nuevo Usuario", modificar u eliminar para cada usuario del sistema.
3. El Directivo de Seguridad y Protección selecciona una de las opciones: - Insertar Usuario. - Modificar Usuario. - Eliminar Usuario.	
Sección Insertar Usuario	

Capítulo 2: Características del Sistema.

	4. El sistema muestra el formulario para la inserción del usuario.
5. El Directivo de Seguridad y Protección entra los datos del usuario y selecciona "Insertar".	6. El sistema verifica usuario en el dominio UCI.
	7. El sistema adiciona usuario al sistema.
	8. El sistema muestra un mensaje: "Usuario insertado".
Sección Modificar Usuario	
	9. El sistema muestra un formulario los datos almacenados referente al usuario seleccionado.
10. El Directivo de Seguridad y Protección modifica los datos del usuario y selecciona "Insertar".	11. El sistema modifica la información almacenada y muestra el mensaje: "Usuario Modificado".
Sección Eliminar Usuario	
	12. El sistema elimina el usuario seleccionado.
	13. El sistema muestra el mensaje: "Usuario Eliminado".
Flujos alternativos	
Acción del actor	Respuesta del sistema
3. Si el Directivo de Seguridad y Protección selecciona cancelar la opción de Administrar Usuario.	3.1 Finaliza el caso de uso.
5. Si el Directivo de Seguridad y Protección selecciona cancelar inserción.	5.1 Pasa a la acción 3.
	6. Si el sistema verifica que el usuario no es del dominio. Muestra el mensaje "Usuario Incorrecto".
6.1 El usuario acepta el mensaje.	6.2 Pasa a la acción 4.
10. Si el Especialista de SP selecciona cancelar modificar usuario.	10.1 Se pasa a la acción 3.

Tabla 11. Descripción expandida del caso de uso del sistema Realizar Búsqueda.

Caso de uso	
CU-3	Realizar Búsqueda

Capítulo 2: Características del Sistema.

Propósito	Obtener información sobre el flujo de accesos a la Universidad, mantener un control sistemático sobre el personal que accede a la misma.	
Actores: Directivo de Seguridad y Protección (SP).		
Resumen: El caso de uso inicia cuando el Directivo de SP selecciona la opción de búsqueda, luego entra los parámetros por los cuales realizará la búsqueda, esta se efectúa, el sistema muestra lo encontrado y si el Directivo lo desea puede obtener algún reporte.		
Referencias	Requisito funcional 3.	
Acción del actor		Respuesta del sistema
1. El Directivo de SP accede al sistema y selecciona la opción que desea consultar: -Accesos -Personas/Último Acceso -Accesos por Intervalo de Fecha		2. El sistema muestra el formulario para la selección y entrada de los parámetros de búsqueda.
3. El Directivo de SP selecciona o entra los parámetros por los cuales desea realizar la búsqueda de los accesos.		4. El sistema realiza la búsqueda de los accesos y muestra los encontrados.
5. El Directivo de SP selecciona uno de los accesos para consultarlo.		6. El sistema le muestra en otra interfaz la información del acceso seleccionado.
7. El Directivo de SP selecciona cerrar la información del acceso.		8. El sistema cierra la interfaz que contiene la información del acceso. Se pasa a la acción 5.
Flujos alternativos		
Acción del actor		Respuesta del sistema
3. Si el Directivo desea abandonar la búsqueda		3.1 Finaliza el caso de uso.

Tabla 12. Descripción expandida del caso de uso del sistema Generar Reportes de Accesos.

Caso de uso	
CU-4	Generar Reportes de Accesos.
Propósito	Obtener información sobre el flujo de accesos a la Universidad, mantener un control sistemático sobre el personal que accede a la misma.
Actores: Directivo de Seguridad y Protección	

Capítulo 2: Características del Sistema.

Resumen El caso de uso inicia cuando el usuario selecciona la opción de realizar el reporte, luego de haber realizado la búsqueda que le permitiera localizar los parámetros sobre los cuales desea obtener el reporte, se le muestra este último y tiene la opción verlo en formato pdf.	
Referencias	Requisito funcional 3, se extiende del CU-3 Realizar Búsqueda.
Acción del actor	Respuesta del sistema
1. El Directivo de Seguridad y Protección accede al sistema y selecciona la opción del acceso que desea obtener.	2. El sistema muestra la información obtenida según los parámetros que solicite en la realización del reporte.
3. El Directivo de Seguridad y Protección selecciona la opción Versión en PDF.	4. El sistema muestra el reporte con los datos en formato PDF.
5. El Directivo de Seguridad y Protección tiene la posibilidad de imprimir o guardar el reporte.	6. El sistema imprime o guarda el reporte.
Flujos alternativos	
Acción del actor	Respuesta del sistema
3. Si el Directivo de Seguridad y Protección cancela la operación de generar el reporte.	3.1 Finaliza el caso de uso.

2.10 Conclusiones

En este capítulo se describieron los procesos del negocio referentes al trabajo con los accesos a la Universidad de Ciencias Informáticas. Esto permitió un mejor entendimiento de dichos procesos y permitió identificar las dificultades presentes en su desarrollo. Además se plantean los requisitos funcionales.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

El objetivo de estos flujos de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema. Tienen como objetivos: transformar los requisitos al diseño del futuro sistema, desarrollar una arquitectura para el sistema y adaptar el diseño para que sea consistente con el entorno de implementación.

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales.

3.2 Arquitectura del Sistema.

Arquitectura del CMS Drupal.

Para el diseño del sistema se utilizó el CMS Drupal, el cual emplea el patrón arquitectónico Modelo Vista Controlador (Model-View-Controller-MVC), utilizado para diseñar aplicaciones con sofisticadas interfaces donde la lógica de interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de la aplicación, lo cual permite implementarlos por separado.
- Hay un API muy bien definido, cualquiera que lo use podrá reemplazar el Modelo, la Vista o el Controlador sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica, se produce en tiempo de ejecución. [19]

Este estilo se basa en separar el diseño de las aplicaciones en 3 capas diferentes:

Modelo: Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nueva información.

Vista: Presenta el modelo en un formato adecuado para interactuar, usualmente las páginas HTML y el código que provee de datos dinámicos a las mismas.

Controlador: Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. [19]

A continuación se muestra una representación básica de las relaciones entre las capas del patrón MVC:

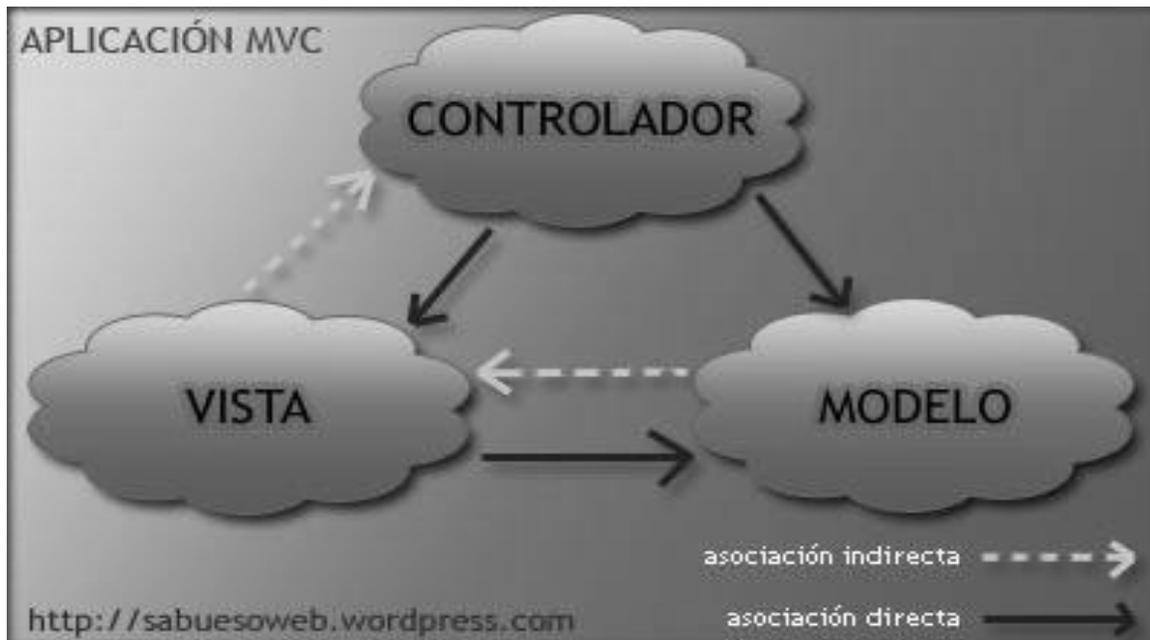


Figura 3.1. Representación del patrón MVC.

El objetivo de este patrón es realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

La siguiente figura muestra la arquitectura del CMS dividida en las 3 capas correspondientes:

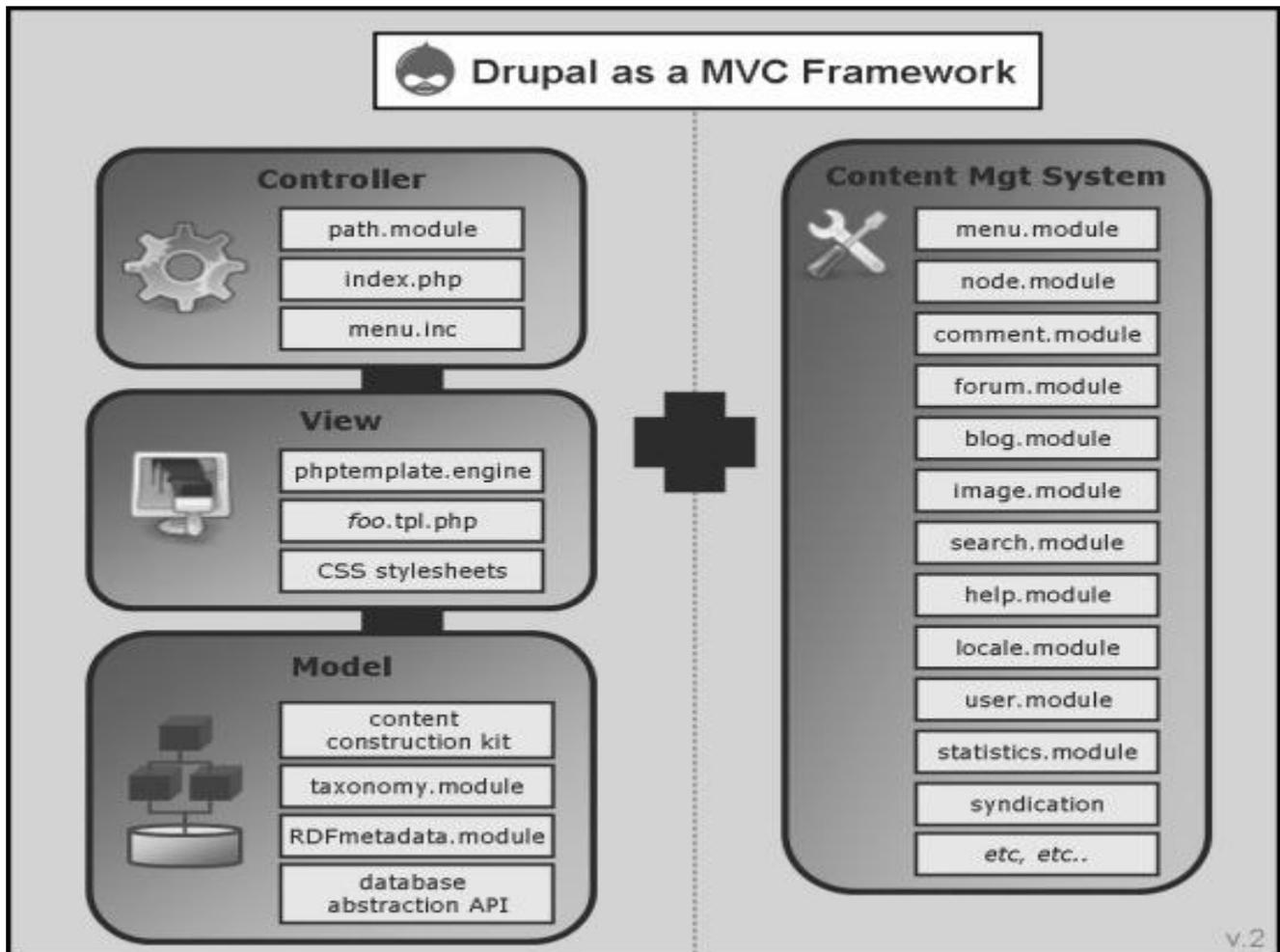


Figura 3.2. Representación del patrón MVC en Drupal.

Para dar cumplimiento a un grupo de funcionalidades requeridas por los clientes no incluidas por el CMS Drupal, se le integra a la aplicación un componente implementado en lenguaje PHP. El componente utiliza como patrón arquitectónico 3 capas.

Capa de presentación o interfaz de usuario: Está formada por los elementos que intervienen en la presentación visual de información al usuario, es la capa que interactúa con el mismo.

Capa de negocio: Esta capa está formada por las entidades que representan objetos que van a ser manejados por toda la aplicación. En este caso, están representados por las clases, funciones, etc.

Capa de acceso a datos: Es donde residen los datos y es la capa encargada de acceder a los mismos. Está formada por uno o más gestores de base de datos que realizan todo el almacenamiento de datos y reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. [20]

La siguiente figura representa la estructura básica del patrón arquitectónico 3 capas.

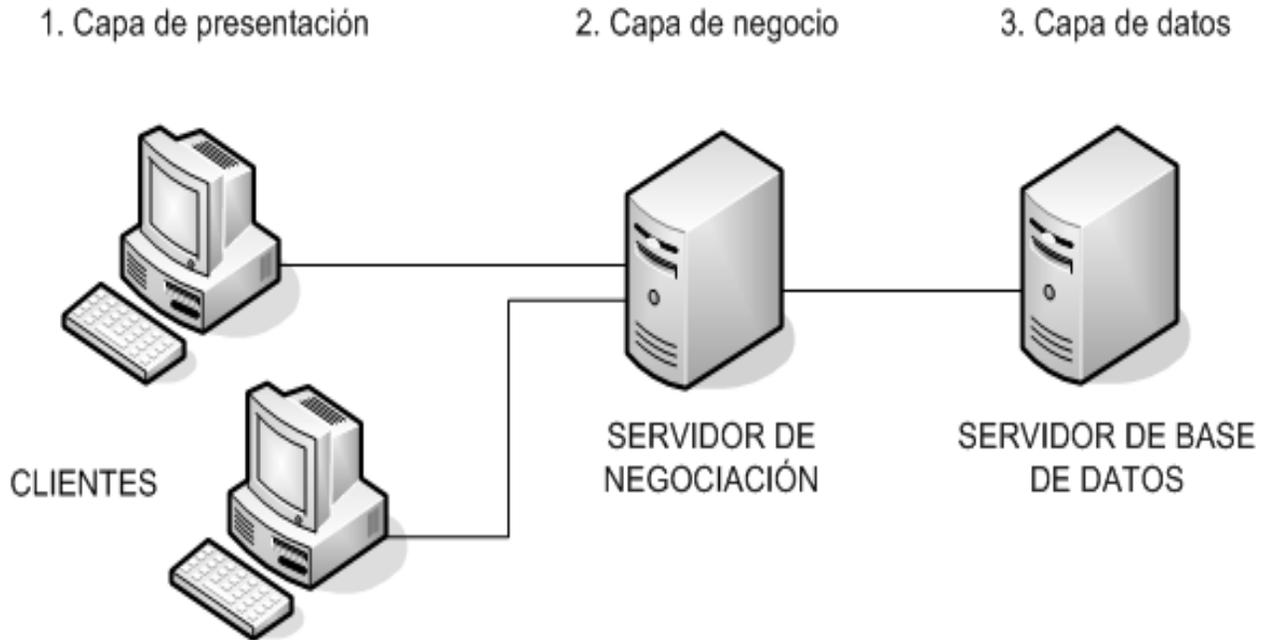


Figura 3.3. Representación del patrón 3 Capas.

¿Cómo se integran las arquitecturas MVC-3 Capas en el sistema?

La integración del componente al CMS se hace a través de los nodos (páginas) del CMS, que representan clases de la capa controlador, o sea, en las páginas (nodo X) se implementan instrucciones en lenguaje PHP que llaman a las clases (formulario.php, session.php, entre otras) pertenecientes a la capa de lógica del negocio en el componente y estas son las encargadas de realizar determinadas operaciones, acceder a los datos (si es necesario) y finalmente mostrar la información a través de la capa "Vista" del CMS. En la figura que se muestra a continuación, se refleja gráficamente la integración de las dos arquitecturas.

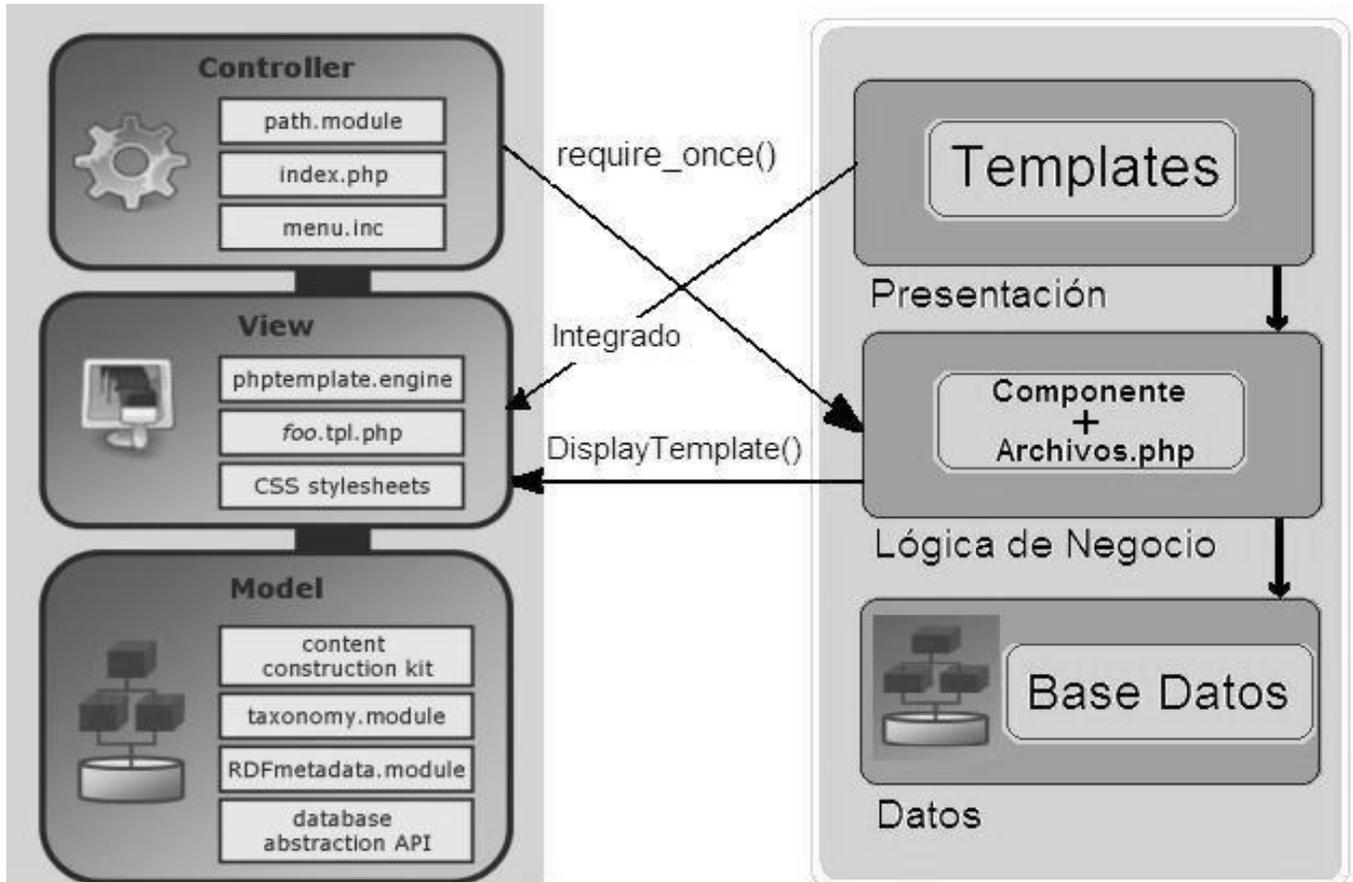


Figura 3.4. Representación de la integración de MVC y 3 Capas.

3.3 Análisis

El análisis proporciona una visión general del sistema que puede ser más difícil de obtener mediante los resultados del diseño y la implementación, debido a que contienen demasiados detalles. Una visión general como esta puede ser muy valiosa para recién llegados al sistema o para desarrolladores que en general mantienen el sistema.

Las clases del análisis representan una abstracción de una o varias clases y/o subsistemas del diseño del sistema. Esta abstracción posee las siguientes características:

Capítulo 3: Análisis y Diseño del Sistema.

- Una clase de análisis se centra en el tratamiento de los requisitos funcionales y pospone los no funcionales, denominándolos requisitos especiales, hasta llegar a las actividades de diseño e implementación subsiguientes.
- Una clase de análisis raramente define u ofrece una interfaz en términos de operaciones y de sus firmas. En cambio su comportamiento se define mediante responsabilidades en un nivel más alto y menos formal. Una responsabilidad es una descripción textual de un conjunto cohesivo del comportamiento de una clase.
- Una clase de análisis define atributos, aunque esos atributos también son de un nivel bastante alto. Normalmente los tipos de esos atributos son conceptuales y reconocibles en el dominio del problema, mientras que los tipos de los atributos en las clases de diseño y la implementación suelen ser tipos de lenguajes de programación.
- Una clase del análisis participa en relaciones, aunque esas relaciones son más conceptuales que sus compartidas de diseño e implementación.
- Las clases de análisis siempre encaja en tres estereotipos básicos: de interfaz, de control o de entidad. Cada estereotipo implica una semántica específica, lo cual constituye un método potente y consistente de identificar y describir las clases de análisis y contribuye a la creación de un modelo de objetos y una arquitectura robusta.

Las clases interfaz se utilizan para modelar la interacción entre el sistema y sus actores, es decir, usuarios y sistemas externos. Las clases entidad se utilizan para modelar información que posee una vida larga y que es a menudo persistente. Las clases control representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. [21]

Se puede apreciar en la Figura 3.1 el diagrama de clases del análisis para la aplicación, donde se ilustran todas las clases de este flujo de trabajo y sus relaciones con otras clases.

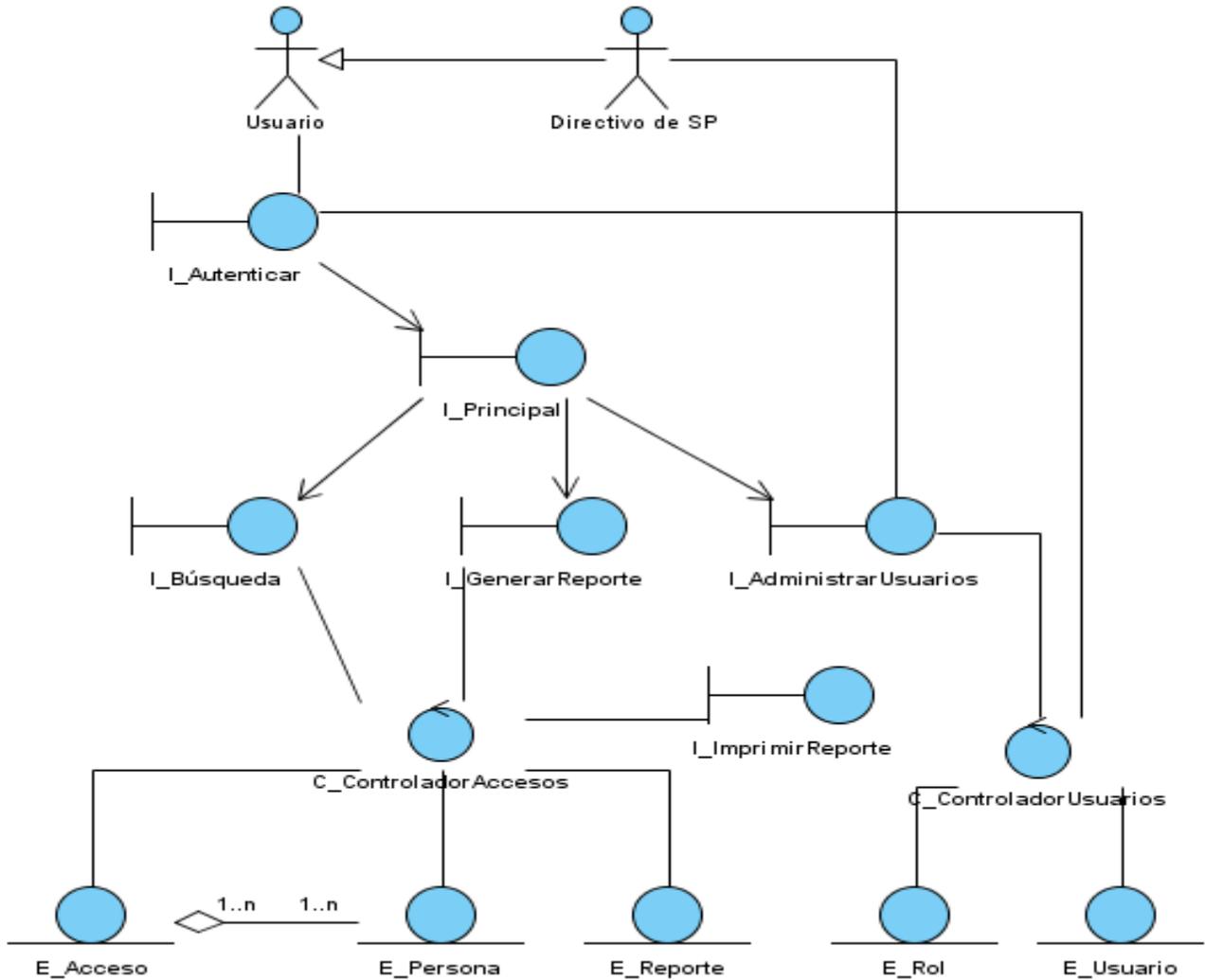


Figura 3.5. Diagrama de clases del análisis.

3.4 Diseño

En este acápite se diseña el sistema propuesto para que sea implementado luego. Es en esta etapa constructiva donde se definen los datos estructuralmente. Se definen clases, entidades, sus relaciones. También se abordan algunos aspectos relacionados con el tratamiento de errores.

3.4.1 Diagrama de Clases de Diseño

El modelo de diseño se crea tomando el modelo de análisis como entrada principal pero adaptada al entorno de implementación elegido, por tanto, mientras el modelo de análisis sirve como una primera aproximación al modelo de diseño, el modelo de diseño funciona como esquema para la implementación. En el diseño de la solución que este trabajo propone se seleccionó Drupal como CMS para llevar a cabo la implementación en materia de codificación y construcción. Debido a la utilización de esta herramienta, se explica de forma general el funcionamiento de la misma y se pasa a describir el diagrama de clases que usa.

Para entender el funcionamiento estructural de Drupal puede verse en forma de paquetes que contienen agrupados los ficheros que implementan las funcionalidades que componen el CMS. A continuación se muestran las secciones en las que se divide Drupal.

Themes: Contiene las plantillas del sistema.

Includes: Contiene los ficheros indispensables para el funcionamiento como por ejemplo: *Database*, que es el que provee las funcionalidades de acceso a la base de datos.

Modules: Incluye los módulos que facilitan las funcionalidades de la aplicación.

Script: Contiene los ficheros correspondientes al manejo visual del sistema, CSS y java script.

La solución propuesta en este trabajo tendría un Diagrama de Clases de Diseño Web en concordancia con el diagrama del propio CMS Drupal y que corresponde al siguiente diagrama:

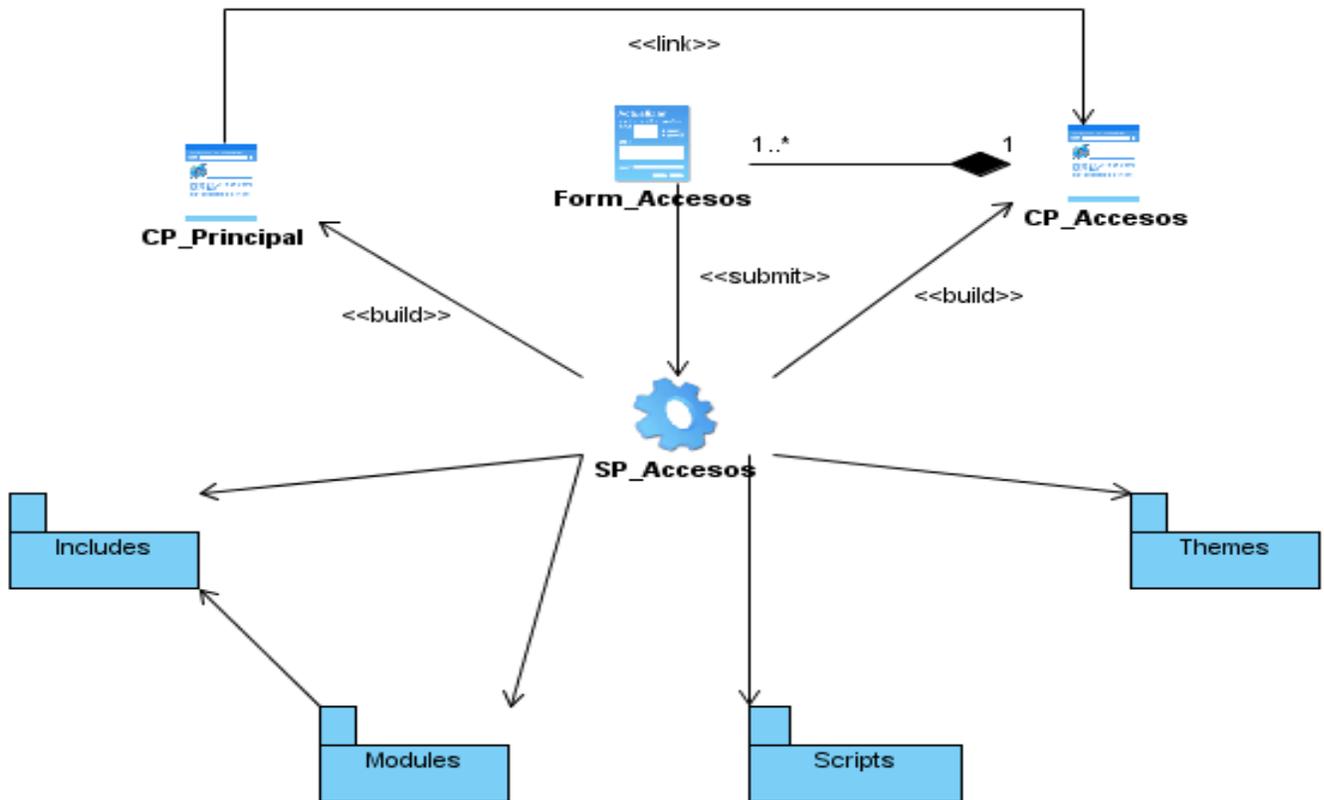


Figura 3.6. Diagrama de Clases de Diseño.

3.4.2 Diagramas de Secuencia.

En este epígrafe se muestra el Diagrama de Secuencia correspondiente al Caso de Uso Realizar Búsqueda y Generar Reporte de Accesos.

Diagrama de Secuencia del CU Realizar Búsqueda.

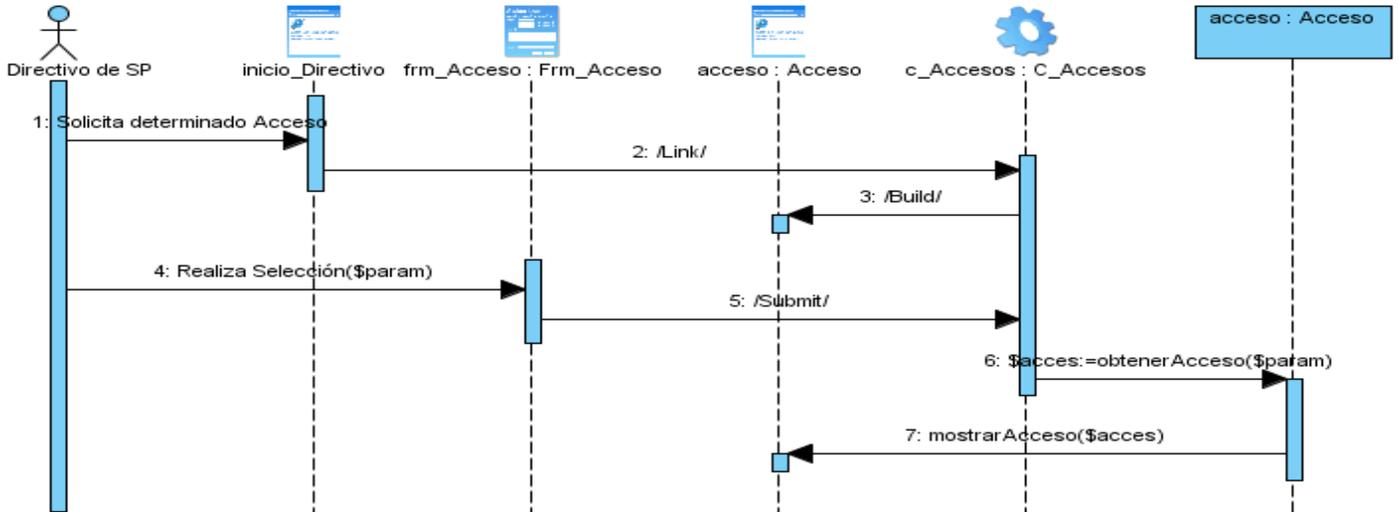


Figura 3.7 Diagrama de secuencia del caso de uso “Realizar Búsqueda”.

Diagrama de Secuencia del CU Generar Reporte de Accesos.

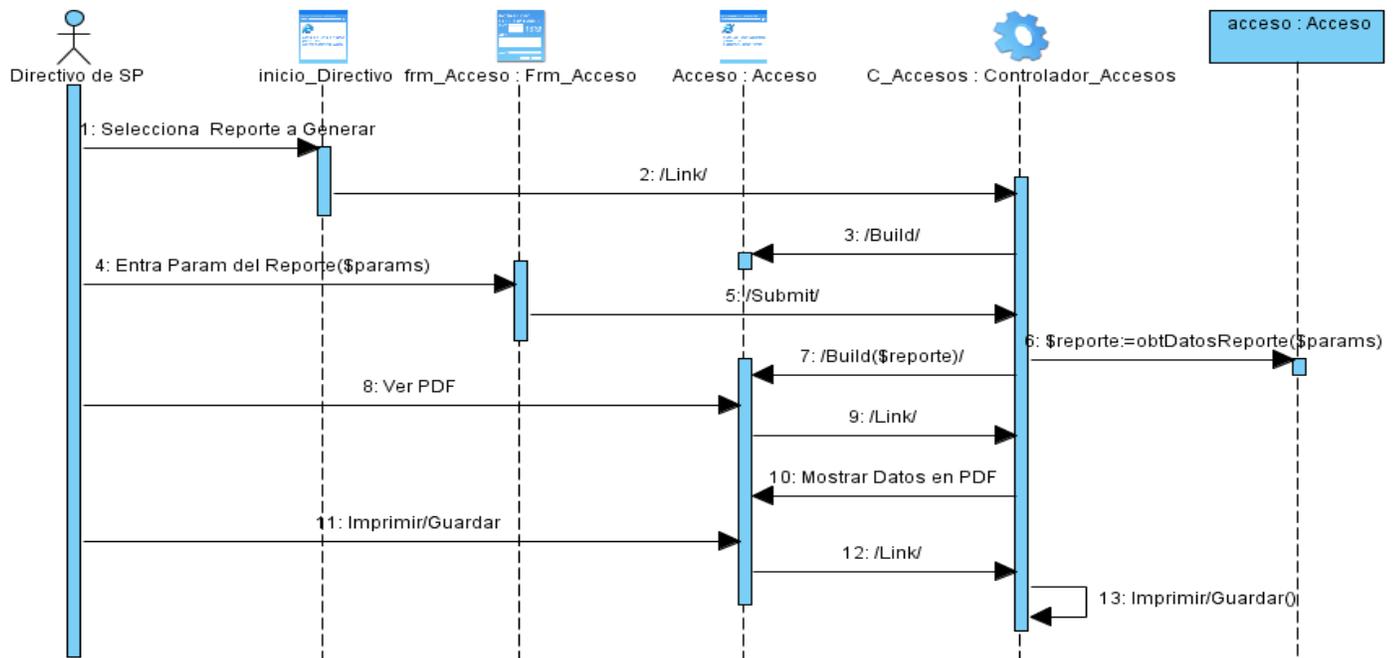


Figura 3.8 Diagrama de secuencia del caso de uso “Generar Reporte de Accesos”.

3.4.3 Diseño de la base de datos.

La figura 3.5 muestra el diseño de la base de datos en la herramienta ERStudio.

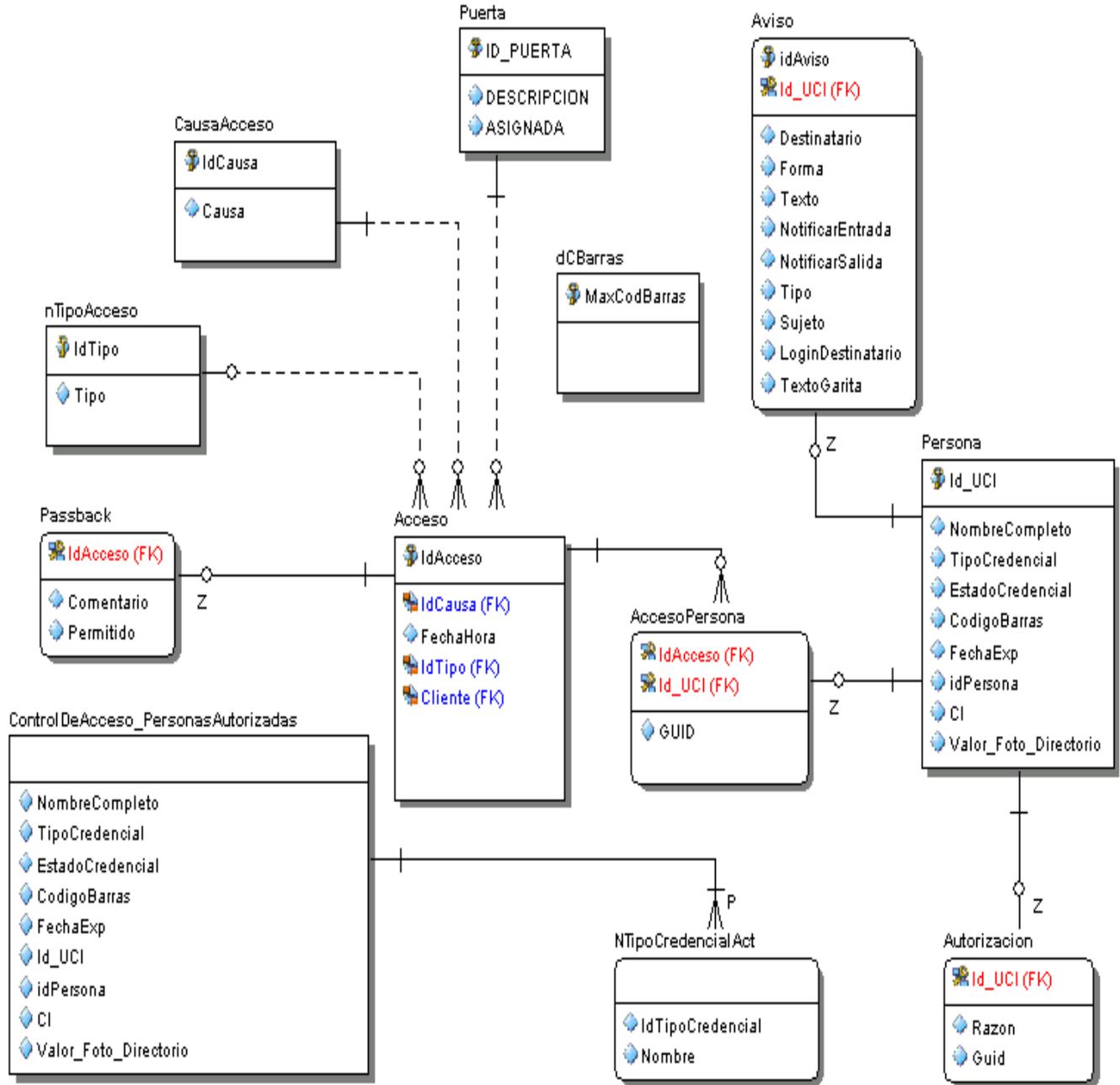


Figura 3.9 Diseño de la base de datos.

3.4.4 Descripción de las tablas del modelo de datos.

Tabla 13: Descripción de la tabla persona.

Nombre: persona		
Descripción: Esta tabla almacena los datos referentes a las personas		
Atributo	Tipo	Descripción
Nombre Completo	varchar(123)	Se almacena el nombre y los apellidos de la persona
Tipo Credencial	nvarchar(50)	Se almacena el tipo de credencial que posee la persona
Estado Credencial	nvarchar(50)	Se almacena el estado en que se encuentra la credencial
Código Barras	nvarchar(10)	Almacena el identificador que le da validez a la credencial
Fecha_Exp	datetime	Almacena la fecha en que expira la credencial
Id_UCI	integer	Es una llave primaria, almacena el identificador de la persona perteneciente a la UCI
Id_Persona	nvarchar (36)	Almacena el identificador de la persona
CI	char(11)	Almacena el Carné de Identidad de la persona
Valor_Foto_Directorio	varchar(50)	Almacena el valor de la foto del directorio

Tabla 14: Descripción de la tabla Puerta.

Nombre: Puerta		
Descripción: Esta tabla almacena los datos de la puerta por la que tuvo lugar el acceso		
Atributo	Tipo	Descripción
IdPuerta	varchar(50)	Es la llave primaria y almacena el identificador de la puerta
Descripción	varchar(150)	Almacena la descripción de la puerta por la que tuvo lugar el acceso
Asignada	bit	Almacena si la puerta fue asignada o no a determinado acceso

Capítulo 3: Análisis y Diseño del Sistema.

Tabla 15: Descripción de la tabla NTipoCredencialAct.

Nombre: NTipoCredencialAct		
Descripción: Esta tabla almacena los datos referentes a las credenciales que están activas		
Atributo	Tipo	Descripción
IdTipoCredencial	integer	Almacena el identificador del tipo de credencial
Nombre	nvarchar(50)	Almacena el nombre de la credencial

Tabla 16: Descripción de la tabla dCBarras.

Nombre: dCBarras		
Descripción: Esta tabla almacena el código de barras de cada credencial activa		
Atributo	Tipo	Descripción
MaxCodBarras	varchar(10)	Es la llave primaria y almacena el código de barras de cada credencial activa

Tabla 17: Descripción de la tabla Acceso.

Nombre: Acceso		
Descripción: Esta tabla almacena los datos relacionados con los accesos		
Atributo	Tipo	Descripción
IdAcceso	integer	Es la llave primaria, almacena el identificador del acceso.
IdCausa	integer	Es una llave extranjera y almacena el identificador de la causa del acceso.
IdTipo	integer	Es una llave extranjera y almacena el identificador del tipo de acceso.
FechaHora	datetime	Almacena la fecha y la hora en que tuvo lugar el acceso.
IdPuerta	Varchar(50)	Es una llave extranjera y almacena el identificador de la puerta del acceso.

Capítulo 3: Análisis y Diseño del Sistema.

Tabla 18: Descripción de la tabla nTipoAcceso.

Nombre: nTipoAcceso		
Descripción: Esta tabla almacena los datos relacionados con los tipos de accesos		
Atributo	Tipo	Descripción
IdTipo	integer	Es la llave primaria y almacena el identificador del tipo de acceso
Tipo	varchar(40)	Almacena el tipo de acceso

Tabla 19: Descripción de la tabla Autorización.

Nombre: Autorización		
Descripción: Esta tabla almacena		
Atributo	Tipo	Descripción
Id_UCI		Es una llave extranjera, forma parte de la llave primaria de la tabla y almacena el identificador de la persona en la UCI
Razón	varchar(50)	Almacena la razón de la autorización
Guid	varchar(36)	(Globally Unique Identifier) Es un número pseudoaleatorio que identifica la persona.

Tabla 20: Descripción de la tabla PassBack.

Nombre: PassBack		
Descripción: Esta tabla almacena		
Atributo	Tipo	Descripción
IdAcceso	integer	Es una llave extranjera, forma parte de la llave primaria de la tabla y almacena el identificador del acceso
Comentario	nvarchar(100)	Almacena un comentario del acceso.
Permitido	integer	Almacena si fue permitido o no el acceso.

Capítulo 3: Análisis y Diseño del Sistema.

Tabla 21: Descripción de la tabla CausaAcceso.

Nombre: CausaAcceso		
Descripción: Esta tabla almacena los datos relacionados con las causas de los accesos		
Atributo	Tipo	Descripción
IdCausa	integer	Es la llave primaria y almacena el identificador de la causa del acceso
Causa	varchar(50)	Almacena la causa del acceso

Tabla 22: Descripción de la tabla Acceso_Persona.

Nombre: Acceso_Persona		
Descripción: Esta tabla almacena los datos relacionados con los accesos de las personas		
Atributo	Tipo	Descripción
IdAcceso	integer	Es una llave extranjera, forma parte de la llave primaria de la tabla y almacena el identificador del acceso
Id_UCI	integer	Es una llave extranjera, forma parte de la llave primaria de la tabla y almacena el identificador de la persona en la UCI
GUID	varchar(50)	(Globally Unique Identifier) Es un número pseudoaleatorio que identifica la persona.

Tabla 23: Descripción de la tabla Aviso.

Nombre: Aviso		
Descripción: Esta tabla almacena los datos relacionados con los avisos		
Atributo	Tipo	Descripción
IdAviso	integer	Es la llave primaria y almacena el identificador del aviso
Id_UCI	integer	Es una llave extranjera y almacena el identificador de la persona en la UCI

Capítulo 3: Análisis y Diseño del Sistema.

Forma	integer	Forma del Mensaje
Texto	varchar(100)	Almacena el texto del aviso
Notificar Entrada	bit	Almacena la notificación de la entrada
Notificar Salida	bit	Almacena la notificación de la salida
Tipo	integer	Almacena el tipo de aviso
Sujeto	varchar(50)	Sujeto del mensaje
Login Destinatario	varchar(50)	Usuario del Destinatario
Texto Garita	char(50)	Almacena texto complementario introducido por el oficial de guardia.

Tabla 24: Descripción de la tabla ControlDeAcceso_PersonasAutorizadas.

Nombre: ControlDeAcceso_PersonasAutorizadas		
Descripción: Esta tabla almacena los datos referentes a las personas autorizadas		
Atributo	Tipo	Descripción
Nombre Completo	varchar(123)	Se almacena el nombre y los apellidos de la persona
Tipo Credencial	nvarchar(50)	Se almacena el tipo de credencial que posee la persona
Estado Credencial	nvarchar(50)	Se almacena el estado en que se encuentra la credencial
Código Barras	nvarchar(10)	Almacena el identificador que le da validez a la credencial
Fecha_Exp	datetime	Almacena la fecha en que expira la credencial
Id_UCI	integer	Almacena el identificador de la persona perteneciente a la UCI
Id_Persona	nvarchar (36)	Almacena el identificador de la persona
CI	char(11)	Almacena el Carné de Identidad de la persona
Valor_Foto_Directorio	varchar(50)	Almacena el valor de la foto del directorio

3.5 Conclusiones

A partir de los procesos descritos en el capítulo anterior se ha desarrollado el análisis y diseño de la aplicación. Se ha obtenido el diagrama de clases del diseño y diagramas de secuencia por caso de uso. Se mostró el modelo entidad relación de la base de datos y las descripciones de las tablas en el mismo. Además se describió la arquitectura del sistema.

CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

En el presente capítulo se describe cómo se ha de implementar la aplicación a desarrollar. Partiendo del resultado de la etapa del diseño, las clases refinadas y sus relaciones se puede comenzar la implementación del sistema. Este nivel en la construcción del sistema debe ser en todo momento consistente con el diseño, es ahora que se construye el sistema en términos de componentes: ejecutables, ficheros de código fuente, scripts, entre otros. El objetivo principal es desarrollar la arquitectura y definir la organización del código. Se modela el sistema en términos de componentes.

4.2 Modelo de despliegue

El modelo de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos hardware sobre los cuales pueden ejecutarse los elementos software. La figura 4.1 presenta el Modelo de Despliegue propuesto.

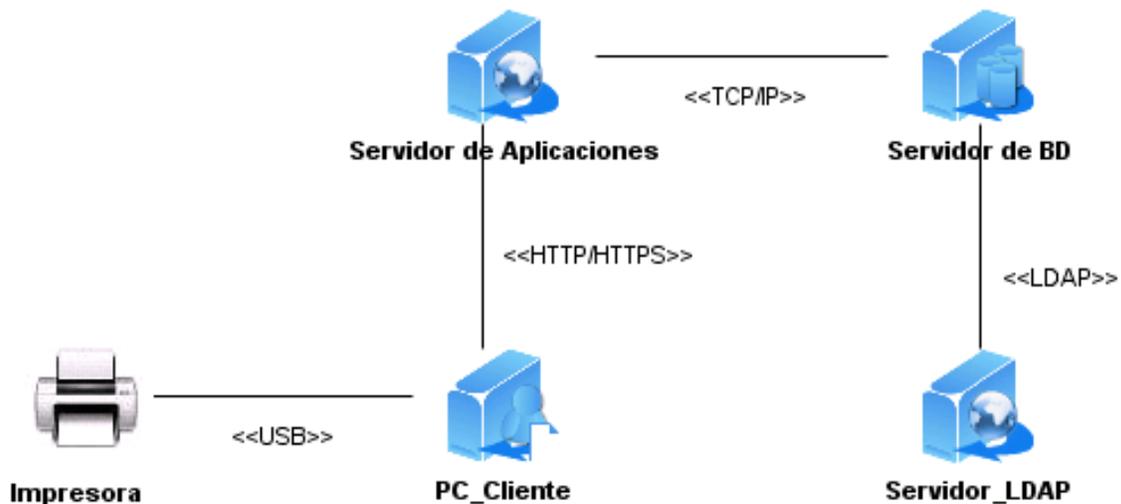


Figura. 4.1 Diagrama de despliegue.

4.3 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

Estereotipos de los componentes:

- Ejecutable: Es un programa que se puede ejecutar en un nodo.
- Biblioteca: Es una biblioteca de objetos estática o dinámica.
- Tabla: Es una tabla de una BD.
- Archivo: Es un fichero que contiene código fuente.
- Documento: Es un documento.
- Página Web: Es una página que se obtiene de la ejecución del sistema.

La figura 4.2 muestra el diagrama de componentes para el sistema propuesto.

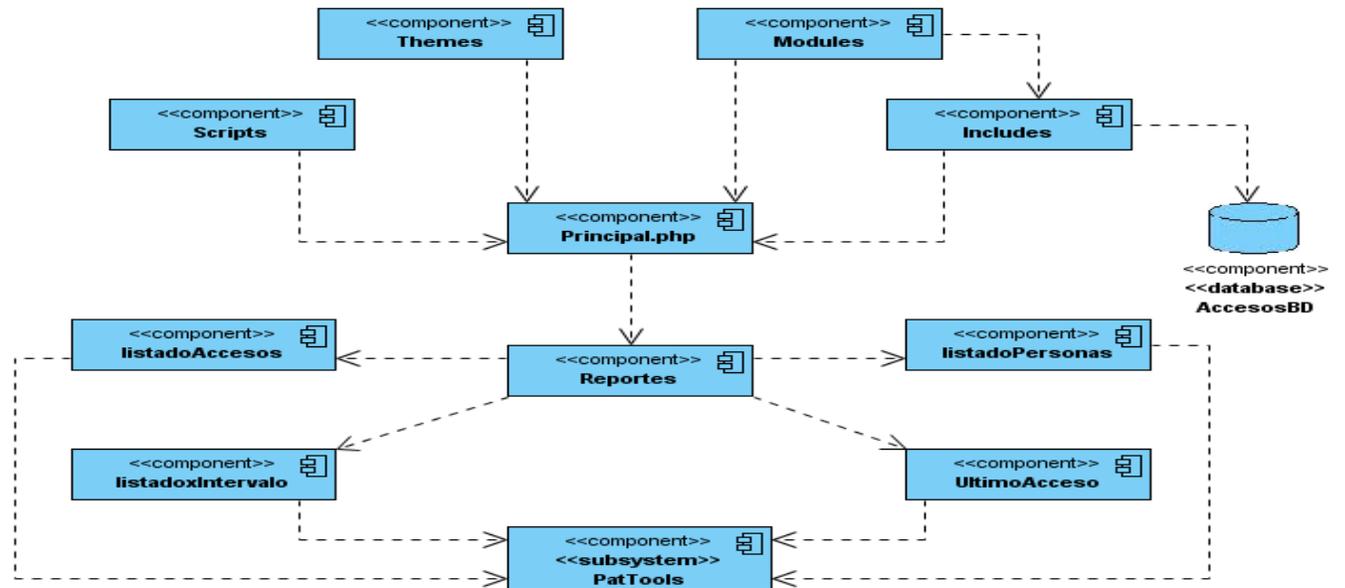


Figura. 4.2 Diagrama de componentes.

4.4 Tratamiento de errores

La aplicación diseñada tiene un proceso de tratamiento de errores bastante robusto a partir de los mecanismos ya implementados en el mismo CMS que se usa para su implementación. Logra monitorear la introducción de datos inválidos y la existencia de aquellos que son obligatorios. Además, Drupal mantiene permanentemente informado al usuario sobre sus acciones sobre la aplicación y como valor agregado, muestra información útil para el usuario que le permite intuir o corregir su interacción con el sistema.

4.5 Seguridad

La seguridad en el sitio está implementada a través del servidor de base de datos MySQL y el uso de variables de sesión para restringir el acceso de los usuarios a determinadas páginas, existen varios niveles de seguridad, definidos como tipos de usuarios, teniendo en cuenta los roles que desempeñan los usuarios que interactúan con el sistema. El sistema posee un módulo de autenticación que verifica que el usuario está registrado y tiene acceso a la aplicación.

4.6 Prueba

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requisitos específicos, los resultados son observados y registrados, donde se evalúa algún aspecto del sistema o componente.

Las pruebas tienen como objetivos:

- Encontrar los defectos que puedan afectar la calidad del software.
- Validar que el software trabaje como fue diseñado.
- Validar los requisitos que debe cumplir el software.
- Validar que los requisitos fueron implementados correctamente.

4.6.1 Pruebas de Caja Negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. 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 un resultado correcto.

Capítulo 4: Implementación y Prueba.

Pruebas de caja negra:

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- Es hecha sin el conocimiento interno del producto.
- No validan funciones ocultas (por ejemplo funciones implementadas pero no descritas en las especificaciones funcionales del diseño) por tanto los errores asociados a ellas no serán encontrados).

4.6.2 Casos de Prueba

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada.

4.6.2.1 Caso de Prueba para el Caso de Uso: Realizar Búsqueda.

Nombre de la sección	Escenarios de la sección	Descripción de la Funcionalidad	Flujo Central
SC 1: Flujo Normal de Eventos	EC 1.1: Accesos.	Permite acceder a una lista con todos los accesos.	El usuario hace clic en el vínculo "Accesos" en el área de reportes. El sistema muestra una página conteniendo una tabla con la lista del contenido solicitado.
	EC 1.2: Listar Personas.	Permite acceder a una lista con todas las personas.	El usuario hace clic en el vínculo "Listar Personas-Último Acceso" en el área de reportes. El sistema muestra una página conteniendo una tabla con la lista del contenido solicitado.

Capítulo 4: Implementación y Prueba.

	EC 1.3: Último Acceso.	Permite acceder al último acceso de cada una de las personas.	El usuario hace clic en el vínculo “Último Acceso” de una determinada persona. El sistema muestra una página con el contenido solicitado.
	EC 1.4: Accesos por Intervalo de Fecha.	Permite acceder a una lista con todos los accesos en el intervalo de fecha solicitado.	El usuario hace clic en el vínculo “Accesos por Intervalo de Fecha” en el área de reportes. El sistema muestra una página conteniendo una tabla con la lista del contenido solicitado.

Sección en Prueba: SC 1 Flujo Normal de Eventos

ID del escenario	Escenario	Respuesta del sistema	Resultado de la Prueba
EC 1.1	Accesos.	El sistema debe mostrar el contenido solicitado.	Satisfactoria
EC 1.2	Listar Personas.		El sistema muestra correctamente el contenido solicitado.
EC 1.3	Último Acceso.		
EC 1.4	Accesos por Intervalo de Fecha.		

4.7 Análisis de costos y beneficios del proyecto.

La estimación de costos y esfuerzos es una de las tareas más difíciles en la gestión de un proyecto de software. En la actualidad existen técnicas que permiten realizar esta labor tales como: COCOMO II y Puntos de Casos de Uso. La especificación de este último ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema, por lo que se ha seleccionado este método para desarrollar el cálculo de esfuerzo para el proyecto.

Cálculo de esfuerzo

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor, se calcula a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

El UAW representa la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema. [22]

Tabla 25: Tipos de actores y sus pesos.

Actores del sistema	Tipo de actor	Factor de Peso
Usuario	Complejo	3
Directivo de Seguridad y Protección	Complejo	3

UAW es igual a la suma de la multiplicación del actor por su factor de peso.

$$UAW = 3 \times 1 + 3 \times 1$$

$$UAW = 6$$

Capítulo 4: Implementación y Prueba.

El UUCW Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

Tabla 26: Tipos de Casos de Uso y sus pesos.

Casos de uso	Tipo de Caso de Uso	Factor de Peso
Autenticar	Simple	5
Administrar Usuarios	Medio	10
Generar Reporte de Accesos	Medio	10
Realizar Búsqueda	Medio	10

UUCW es igual a la suma de la multiplicación del Caso de Uso por su factor de peso.

$$UUCW = 3 \times 10 + 1 \times 5$$

$$UUCW = 35$$

Al ser UAW=6 y UUCW=35 finalmente, los Puntos de Casos de Uso sin ajustar resultan:

UUCP = UAW + UUCW sustituyendo tenemos:

$$UUCP = 35 + 6$$

$$UUCP = 41$$

Cálculo de Puntos de Casos de Uso ajustados

$$UCP = UUCP \times TCF \times EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Capítulo 4: Implementación y Prueba.

El TCF es el coeficiente que se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Tabla 27: Factor de complejidad técnica.

Factor	Descripción	Peso	Valor asignado
T1	Sistema distribuido	2	1
T2	Objetivos de performance o tiempo de respuesta	1	1
T3	Eficiencia del usuario final	1	2
T4	Procesamiento interno complejo	1	2
T5	El código debe ser reutilizable	1	0
T6	Facilidad de instalación	0.5	0
T7	Facilidad de uso	0.5	2
T8	Portabilidad	2	1
T9	Facilidad de cambio	1	2
T10	Concurrencia	1	1
T11	Incluye objetivos especiales de seguridad	1	1
T12	Provee acceso directo a terceras partes	1	0
T13	Se requieren facilidades especiales de entrenamiento	1	0

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$TCF = 0.72$$

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente (EF). El cálculo del mismo se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Capítulo 4: Implementación y Prueba.

Tabla 28: Factor de ambiente.

Factor	Descripción	Peso	Valor asignado
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4
E2	Experiencia en la aplicación	0.5	3
E3	Experiencia en trabajo orientado a objetos	1	4
E4	Capacidad del analista líder	0.5	4
E5	Motivación	1	5
E6	Estabilidad de los requisitos	2	5
E7	Personal a tiempo completo	-1	0
E8	Dificultad del lenguaje de programación	-1	0

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$EF = 0.545$$

Al ser UUCP = 41, TCF = 0.74, EF = 0.545 y

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 41 \times 0.74 \times 0.545$$

$$UCP = 16.5353$$

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde:

E: Esfuerzo

UCP: Puntos de Casos de Uso ajustados (calculado anteriormente).

CF: Factor de conversión (para este tipo de proyecto 20 horas-hombre/Punto de Casos de Uso).

Capítulo 4: Implementación y Prueba.

Al tener UCP = 16.5353, CF = 20 horas-hombre/Punto de Casos de Uso y

$$E = UCP \times CF$$

$$E = 16.5353 \times 20 \text{ horas-hombre/Punto de Casos de Uso}$$

$$E = 330.706 \text{ horas-hombre}$$

Tabla 29: Esfuerzo por flujo de trabajo.

Actividad	Porcentaje	Esfuerzo
Análisis	10%	82.6765
Diseño	20%	165.353
Implementación	40%	330.706
Prueba	15%	124.014
Sobrecarga	15%	124.014
Total	100%	826.765

El proyecto requiere de 826.765 horas-hombre para su desarrollo. Trabajando 8 horas diarias se obtiene aproximadamente 103 días para el desarrollo del proyecto, es decir, para una persona trabajando 26 días al mes el proyecto tiene una duración de 4 meses aproximadamente.

Costos y beneficios

Varios de los proyectos que se desarrollan en la Universidad se realizan para mejorar procesos que en ella existen. Este proyecto es uno de ellos, ya que proporciona mejoras a los procesos que tienen lugar en la Oficina de Seguridad y Protección, relacionados con el trabajo de los accesos a la Universidad de las Ciencias Informáticas.

Por lo antes planteado los beneficios que se obtienen con el sistema son beneficios intangibles. Los beneficios intangibles son aquellos que son difíciles de cuantificar, pero no por ello dejan de ser importantes. Algunos de ellos son:

- Aumento de la precisión cuando se realiza una búsqueda de persona o acceso.
- Mejora y agiliza los procesos de negocio.
- Incrementa la satisfacción de los usuarios del sistema al eliminar el trabajo tedioso.

El proyecto no incluye costos pues no ha sido necesario realizar gastos para adquirir equipos, como computadoras, ni en salarios para desarrolladores. Esto se debe a que la fuerza de trabajo con que se cuenta consiste en estudiantes que no son remunerados y el equipamiento es proporcionado por la Universidad.

4.8 Conclusiones

En este capítulo se definió la arquitectura física del sistema mediante el diagrama de despliegue, y el diagrama de componentes con el objetivo de brindar una idea de cómo se implementó el software en término de componentes. Se trató la validación del software mediante los casos de prueba de caja negra para verificar el correcto funcionamiento de la aplicación comprobándose que no existen errores en las funciones operativas del software. Además se realizó el análisis de costos y beneficios del proyecto.

CONCLUSIONES

Con el desarrollo del presente trabajo se ha logrado:

- Estudiar el flujo de información relacionada con el control de accesos de la Universidad de las Ciencias Informáticas.
- Realizar un estudio de estrategias y herramientas usadas para facilitar el manejo de esta información.
- Estudiar herramientas y metodologías adecuadas para resolver las problemáticas planteadas.
- Diseñar e implementar una solución informática factible, que resuelva las dificultades planteadas.
- Realizar pruebas para comprobar la funcionalidad de la solución propuesta e implementada en este trabajo.

De esta forma se ha cumplido con los objetivos planteados en la presentación de este trabajo de diploma, pues se ha logrado de forma eficiente la implementación de un sistema que posibilita la obtención de reportes que permiten una correcta manipulación de la información generada de los accesos a la Universidad de las Ciencias Informáticas.

RECOMENDACIONES

Al mismo tiempo que los objetivos planteados en el desarrollo de este trabajo se cumplieron con éxito, se hacen las siguientes recomendaciones:

- Garantizar al sistema implementado la debida atención y actualización para mantener de forma confiable su fuente de información y servicios de calidad relacionados con el control de accesos.
- Realizar un manual de usuario para garantizar la preparación del personal que trabaje con el sistema implementado.
- Continuar estudiando el proceso de control de accesos y las necesidades de la Universidad para mejorar las funcionalidades del sistema implementado (Visualizar reportes mediante gráficas por ejemplo).
- Se recomienda este trabajo para ser usado como material de estudio en la realización de alguna aplicación similar.

REFERENCIAS BIBLIOGRÁFICAS

- [1]. INFORMATION MANAGEMENT. ¿Qué es la gestión de la información? [En línea] 2006. [Disponible en: <http://informationmanagement.wordpress.com/2006/11/28/%c2%bfque-es-la-gestion-de-la-informacion-3-de-4/>)]
- [2]. Cosaslibres.com. Software. En línea. [Disponible en: <http://www.cosaslibres.com/software.html>]
- [3]. Silva Picón, Carla. Fierro Caro, Luis. Sistema de reportes estadísticos de la gestión del SIBUDEEC, a partir de datos generados en el Sistema ALEPH.2007. [Disponible en: <http://www.sibudec.cl/quam/ponencias/infogestion.pdf>]
- [4]. mkm Publicaciones. Implementación de un sistema de reportes financieros.2007. [Disponible en: <http://www.mkm-pi.com>]
- [5]. IDENTIFIC-AR. Control Horario, acceso huella dactilar, Tiempo y Asistencia. [En línea] [Disponible en: <http://identific-ar.com.ar/acceso.htm>]
- [6]. RC Soluciones Integrales. Ipass - herramienta de control de acceso empresarial.2007. [Disponible en: <http://www.rcsolucionesintegrales.com/docs/FichaTecnicalpass.pdf>]
- [7]. Reyero, José A. Sobre Drupal. [En línea] 16 de Marzo de 2006. [Disponible en: <http://drupal.org.es/drupal>]
- [8]. Álvarez, Rubén. Introducción a la programación en ASP. [En línea] [Disponible en: <http://www.desarrolloweb.com/articulos/244.php>]
- [9]. Pérez Hidalgo, Andrey F. DESCRIPCIÓN DE LOS ASPECTOS FUNDAMENTALES DEL LENGUAJE DE PROGRAMACIÓN PERL, 2007. [Disponible en: <http://www.dimare.com/adolfo/cursos/2007-1/pp-Perl.pdf>]
- [10]. Castillo, Alvaro del. Webs dinámicos con PHP. [En línea] 2006. [Disponible en: <http://www.programacion.net/php/tutorial/php4/2/>]

- [11]. Multimedia. 2009. [Disponible en: http://es.encarta.msn.com/encyclopedia_761562178/multimedia.html].
- [12]. PostgreSQL Global Development Group PostgreSQL. [En línea] [Disponible en: <http://www.postgresql.org>].
- [13]. Romero, Diego F. ¿Qué Es Un Servidor De Aplicaciones? [En línea] 2007. [Disponible en: <http://www.editum.org/Que-Es-Un-Servidor-De-Aplicaciones-p-473.html>].
- [14]. Foundation, Apache Software. Apache. [En línea] [Disponible en: <http://www.apache.org>].
- [15]. Molpeceres, Alberto. Procesos de desarrollo: RUP, XP y FDD. 2003. [Disponible en: <http://www.willydev.net/descargas/Articulos/General/cualxpfdrup.PDF>].
- [16]. Booch, G, Rumbaugh, J y Jacobson, I. El Lenguaje Unificado de Modelado. 2000.
- [17]. Périssé, M C. Una Metodología Simplificada. Buenos Aires: s.n., 2001.
- [18]. Free Download Manager. Visual Paradigm for UML. [En línea] [Disponible en: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/].
- [19]. Catalani, Exequiel. ARQUITECTURA Modelo/Vista/Controlador. [En línea] [Disponible en: <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>].
- [20]. Slideshare. Arquitectura 3 Capas. [En línea] [Disponible en: <http://www.slideshare.net/Decimo/arquitectura-3-capas>].
- [21]. JACOBSON, IVAR. El Proceso Unificado de Desarrollo de Software. Ciudad de La Habana, Félix Varela, 2004, p.165-253.
- [22]. PERALTA, MARIO. Estimación del esfuerzo basada en Casos de Uso. [Disponible en: <http://www.itba.edu.ar/capis/rtis/rtis-6-1/estimaci%F3n-del-esfuerzo-basada-en-casos-de-usos.pdf>].

BIBLIOGRAFÍA

1. Apache Software Foundation. Apache. [En línea] [Disponible en: <http://www.apache.org>]
2. Desarrolloweb.com. Zend Studio. 2008 [En línea] [Disponible en: <http://www.desarrolloweb.com/articulos/1178.php> .]
3. Drupal Hispano. [En línea] [Disponible en: <http://drupal.org.es/>]
4. Jacobson, Booch, Rumbaugh. El Proceso Unificado de Desarrollo de Software. Madrid, Pearson educación, S.A, 2000. P.464.
5. La Web del Programador. [En línea] [Disponible en: <http://www.lawebdelprogramador.com/>]
6. Booch, Grady. 1996. Análisis y Diseño Orientado a Objetos. 2da edición. Ed. Addison-Wesley / Díaz de Santos.
7. Pressman, Robert. 1998. Ingeniería de Software.
8. Web oficial de PHP. [En línea] [Disponible en: <http://www.php.net/>]
9. Molpeceres, Alberto. Procesos de desarrollo: RUP, XP y FDD. 2003. [Disponible en: <http://www.willydev.net/descargas/Articulos/General/cualxpfdrrup.PDF>].
10. Booch, G, Rumbaugh, J y Jacobson, I. El Lenguaje Unificado de Modelado. 2000.

GLOSARIO DE TÉRMINOS

API: Application Programming Interface (Interfaz de Programación de Aplicaciones).

CMS: Content Management System (Sistema de gestión de contenidos) es un programa que se refiere usualmente a sitios web que tienen sistemas de administración que permiten la fácil creación y edición de contenidos, como páginas nuevas, noticias, etc.

Tomado de: Glosario de Informática e Internet. <http://www.panamacom.com/glosario/letra-c.html>

HTML: HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Tomado de: Glosario de Informática e Internet. <http://www.panamacom.com/glosario/letra-h.html>

HTTP: HyperText Transfer Protocol Protocolo de Transferencia de Hipertexto. HTTP es un protocolo con la ligereza y velocidad necesaria para distribuir y manejar sistemas de información hipermedia.

Tomado de: Glosario de Informática e Internet. <http://www.panamacom.com/glosario/letra-h.html>

IDE: Integrated Development Environment (Entorno de desarrollo integrado) es un programa compuesto por un conjunto de herramientas para un programador.

Tomado de: Babylon. http://www.babylon.com/definition/entorno_de_desarrollo/Spanish.

JavaScript: Lenguaje desarrollado por Netscape y aunque es parecido a Java se diferencia de él en que los programas están incorporados en el archivo HTML.

Tomado de: Glosario de Informática e Internet. <http://www.panamacom.com/glosario/letra-j.html>

NCSA: National Center for Supercomputing Applications (Centro Nacional de Aplicaciones de Supercomputación) es un organismo estadounidense relacionado con la investigación en el campo de la Informática y las Telecomunicaciones.

Tomado de: <http://www.glosarium.com/term/443,4,xhtml>

Proceso: En informática se manejan varias definiciones que aluden a diversos elementos: puede ser simplemente una operación o conjunto combinado de operaciones con datos, o bien una secuencia de acontecimientos definida única y delimitada, que obedece a una intención operacional en condiciones predeterminadas. También se denomina proceso a una función que se está ejecutando.

Tomado de: Glosario de Informática e Internet.

http://www.casarramona.com/mt/programador/diccionario/dic_p.html

RUP: Rational Unified Process (Proceso Unificado de Desarrollo) es la metodología de IBM Rational para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología.

Tomado de: Jacobson, Booch, Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. Madrid, Pearson educación, S.A, 2000. P.464.

SGBD: DataBase Management System (Sistemas de Gestión de Base de Datos) 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.

Tomado de: Babylon.

http://www.babylon.com/definicion/sistema_de_gesti%C3%B3n_de_base_de_datos/Spanish

UML: Unified Modeling Language (Lenguaje Unificado de Modelado) 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.

Tomado de: Popkin Software and Systems, Modelado de Sistemas con UML. P.532.

XML: Extensible Markup Language (lenguaje de marcas) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

Tomado de: Glosario de Informática e Internet. <http://www.panamacom.com/glosario/letra-x.html>