

**DESARROLLO DE UNA APLICACIÓN WEB PARA EL REGISTRO ACADÉMICO
DE LA INSTITUCIÓN EDUCATIVA MARÍA ANTONIA RUIZ**

**ÓSCAR ADRIÁN PARRA
LUIS GERMÁN TORO PAREJA**

**CÓD. 0766887
CÓD. 0767394**



**UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
INGENIERÍA DE SISTEMAS
TULUÁ – VALLE
2013**

**DESARROLLO DE UNA APLICACIÓN WEB PARA EL REGISTRO ACADÉMICO
DE LA INSTITUCIÓN EDUCATIVA MARÍA ANTONIA RUIZ**

**ÓSCAR ADRIÁN PARRA
LUIS GERMÁN TORO PAREJA**

**CÓD. 0766887
CÓD. 0767394**

**DIRECTORA
INGENIERA DE SISTEMAS
CLAUDIA PATRICIA MORENO**



**UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
INGENIERÍA DE SISTEMAS
TULUÁ – VALLE
2013**

RESUMEN

La Institución Educativa María Antonia Ruiz realiza sus procesos de registro académico¹ de forma manual, lo que dificulta el manejo excesivo de la papelería de cada estudiante, que conlleva en algunos casos a la pérdida y deterioro de los mismos. Tomando en cuenta el avance de la tecnología y el Internet se busca cubrir sus necesidades a través de una aplicación web que agilice los procesos descritos de forma eficiente y segura.

El diseño e implementación de la aplicación está dirigido al área administrativa y secretaría académica, profesores, alumnos y acudientes. Donde la secretaria puede realizar la matrícula académica, los estudiantes visualizan sus notas, descargar el certificado de estudio y el carnet estudiantil. Los docentes de la Institución pueden registrar las notas de las materias de cada estudiante.

Para dar solución al problema, se utilizará la metodología ICONIX para desarrollo web basado en procesos ágiles. Es un marco de trabajo conceptual de la ingeniería de web que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto.

¹ Se entiende por Registro Académico, los procesos de matrícula académica, registro de notas y generación de reportes (fichas de matrícula, listado de notas, certificado de estudio, carnet estudiantil).

ABSTRACT

Educational Institution Maria Antonia Ruiz conducts academic registration processes manually, which excessive difficulty handling the paperwork of each student, which in some cases leads to the loss and degradation of these. Given the advancement of technology and the Internet are looking to fill their needs through a web application to streamline the processes described in an efficient and safe.

The design and implementation of the application is directed to the administrative and academic secretary, teachers, students and guardians. Where the secretary can make academic enrollment, students displayed their notes, download the certificate of study and student card. Institution Teachers can record notes of every student matters.

To solve the problem, we will use the ICONIX methodology based web development agile processes. A conceptual framework of web engineering that promotes development iterations throughout the project life cycle.

CONTENIDO

RESUMEN.....	3
1. INTRODUCCIÓN.....	10
1.1 DESCRIPCIÓN GENERAL.....	10
1.2 PLANTEAMIENTO DEL PROBLEMA.....	11
1.2.1 Descripción del problema	11
1.2.2 Formulación del Problema	11
1.3 OBJETIVOS	12
1.3.1 Objetivo General.....	12
1.3.2 Objetivos Específicos	12
1.4 ALCANCE DEL PROYECTO.....	13
1.5 ESTRUCTURA DEL DOCUMENTO	13
2. MARCO DE REFERENCIA.....	14
2.1 MARCO TEÓRICO.....	14
2.1.1 Ingeniería Web	14
2.1.2 Proceso de ingeniería web	14
2.1.3 Aplicación Web.....	15
2.1.3.1 Servidor web.....	16
2.1.3.2 Navegador web o browser	16
2.1.4 Sistema de Información	17
2.1.5 PHP.....	17
2.1.6 MYSQL.....	18
2.1.7 Framework	18
2.1.8 UML.....	18
2.1.9 Modelo vista controlador.....	19
2.2 MARCO CONTEXTUAL	20
2.3 MARCO CONCEPTUAL.....	22
2.4 MARCO LEGAL.....	24
3. DESARROLLO DEL PROYECTO.....	28
3.1 ANÁLISIS DE REQUISITOS.....	31
3.1.1 Análisis de Usuarios y Fases.....	32
3.1.2 Detalles del Sistema	33
3.1.3 Requerimientos	35
3.1.3.1 Requerimientos Funcionales.....	35
3.1.3.2 Requerimientos No Funcionales	37
3.1.4 Modelo de Dominio.....	39
3.2 ANÁLISIS Y DISEÑO PRELIMINAR	40
3.2.1 Casos de Uso.....	40
3.2.1.1 Diagramas de Casos de Uso:	40

3.2.1.2	Descripción de Casos de Uso	41
3.2.2	DIAGRAMA DE PAQUETES	43
3.3	DISEÑO.....	44
3.3.1	Modelo Entidad-Relación.....	44
3.3.2	Diagramas de Secuencia.....	46
3.3.3	Diagrama de Clases	48
3.4	IMPLEMENTACIÓN	49
3.4.1	Aplicaciones y Herramientas	49
3.4.2	Estructura de la Implementación	52
3.4.3	Diagrama de Componentes	57
4.	PRUEBAS Y RESULTADOS	58
4.1	Casos de Prueba.....	59
	CASO DE PRUEBA: GENERAR MATRÍCULA ACADÉMICA	60
4.2	Resultados	61
5.	CONCLUSIONES Y TRABAJOS FUTUROS	62
5.1	Trabajos Futuros	62
5.2	Conclusiones	63
	BIBLIOGRAFÍA.....	64

LISTA DE FIGURAS

	Pág.
Figura 1. Elementos de un Sistema de Información	17
Figura 2. Modelo Vista Controlador (MVC)	19
Figura 3. Metodología ICONIX.....	30
Figura 4. Formato Registro de Notas	31
Figura 5. Jerarquía de los Usuarios del Sistema.....	34
Figura 6. Modelo de Dominio	39
Figura 7. Caso de Uso Gestionar Matrícula Académica.....	40
Figura 8. Caso de Uso Gestionar Grupos	40
Figura 9. Diagrama de Paquetes	43
Figura 10. Modelo Entidad – Relación	45
Figura 11. Diagrama de Secuencia: Generar Matrícula Académica	46
Figura 12. Diagrama de Secuencia: Modificar Grupo.....	47
Figura 13. Diagrama de Clases	48
Figura 14. Proceso de compilación y publicación de una aplicación Flex	50
Figura 15. Elementos Protocolo AMFPHP	52
Figura 16. Diagrama de Componentes	57

LISTA DE TABLAS

	Pág.
Tabla 1. Requerimientos Módulo Usuarios	35
Tabla 2. Requerimientos Módulo Parámetros/Configuración	36
Tabla 3. Requerimientos Módulo Registro académico	37
Tabla 4. Requerimientos Módulo Reportes	37
Tabla 5. Requerimientos de Interfaz de Usuario	37
Tabla 6. Requerimientos de Software	37
Tabla 7. Requerimientos de Hardware	38
Tabla 8. Requerimientos de Plataforma	38
Tabla 9. Requerimientos de Seguridad	38
Tabla 10. Requerimientos de Confiabilidad y Disponibilidad	38
Tabla 11. Caso de Uso Generar Matrícula	42
Tabla 12. Caso de Uso Modificar Grupo	42
Tabla 13. Caso de Prueba: Ingresar al Sistema	59
Tabla 14. Caso de Prueba: Generar Matrícula Académica	60

ANEXOS

ANEXO A: Documento de Formatos y Plantillas

ANEXO B: Especificación de Requerimientos de Software

ANEXO C: Documento de Diseño del Sistema

ANEXO D: Documento de Pruebas del Sistema

1. INTRODUCCIÓN

1.1 DESCRIPCIÓN GENERAL

Actualmente, en el municipio de Tuluá las instituciones educativas de carácter público están en un constante crecimiento de su población estudiantil, lo cual genera una gran cantidad y variedad de información, que es necesaria almacenar y procesar bajo los criterios establecidos por el ministerio de educación, la secretaría de educación municipal y el pacto de convivencia de la Institución.

Los Sistemas de Información han sido de gran aporte para satisfacer las necesidades requeridas en cuanto al manejo y procesamiento de la información. Han ayudado a las instituciones de educación en los procesos de matrícula académica, registro de notas, gestión de estudiantes, permitiendo tener la información organizada y estructurada, de tal forma que siempre esté disponible, y se pueda acceder de forma rápida y convergente.

En este documento se plantea el desarrollo de una aplicación web, para agilizar los diferentes procesos de registro académico que se llevan a cabo en la Institución Educativa María Antonia Ruiz y para gestionar la información generada, aplicando las técnicas de la ingeniería web.

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 Descripción del problema

Actualmente la Institución pública educativa, María Antonia Ruiz de Tuluá (Valle del Cauca), realiza los procesos de matrícula académica, administración de notas, certificado de estudio y reportes de forma manual, causando diversos problemas en el área de Secretaria académica. Se hace muy dispendioso manejar, procesar y analizar la gran cantidad de archivos físicos que se generan, ocasionando posibles equivocaciones, pérdidas y daño de los archivos. Esta forma de realizar los procesos, ocasiona demora en las matrículas, en la consulta del registro académico de los estudiantes, retraso en la entrega de notas, boletines, y además propicia a la obtención y entrega de información errónea o desactualizada en los documentos. Estos problemas provocan inconvenientes con los padres de familia y los estudiantes.

1.2.2 Formulación del Problema

¿Cómo se solucionan las dificultades del registro académico que presenta la Institución Educativa María Antonia Ruiz a través de un producto de software?

1.3 OBJETIVOS

1.3.1 Objetivo General

Desarrollar una aplicación web para la generación del registro académico de la Institución Educativa María Antonia Ruiz, que permita optimizar y mejorar los procesos desarrollados durante el año lectivo.

1.3.2 Objetivos Específicos

- ❖ Definir las actividades y elementos que componen el proceso de registro académico que maneja la Institución, en el área de Matrícula académica, estudiantes, docentes, registro de notas, cursos y asignaturas.
- ❖ Diseñar e implementar los módulos del sistema que permitan el manejo de los siguientes procesos:
 - Gestión de usuarios.
 - Matrícula académica.
 - Gestión de estudiantes.
 - Registro de notas.
 - Administrar cursos y asignaturas.
 - Gestión de docentes.
- ❖ Sistematizar los procedimientos y formatos de registro académico de la Institución (ficha de matrícula, listado de notas, certificado de estudio y carnet estudiantil) con el fin de que garantice el esquema corporativo que manejan actualmente.
- ❖ Realizar un módulo de reportes, que permita generar las Fichas de matrícula de la Institución, los listados de Notas, certificados de estudio y carnet estudiantil.

1.4 ALCANCE DEL PROYECTO

El presente trabajo tiene en cuenta lo siguiente para definir los límites:

- La aplicación se realizará para la sede Central de la Institución Educativa María Antonia Ruiz.
- El sistema solo permitirá realizar el registro académico de la Institución, matrícula académica, registro de notas, generación de reportes (ficha de matrícula, listado de notas por materia, certificado de estudio y carnet estudiantil).
- No nos haremos responsables de la documentación física que almacena la Institución como folders, archivos, documentos, recibos o actas.
- Los parámetros de configuración del sistema se implementarán según lo estipulado en el Pacto de Convivencia de la Institución, por ejemplo las notas, los criterios de evaluación, las asignaturas y áreas.
- Se realizará un Plan de Pruebas para verificar el correcto funcionamiento de los Módulos de la aplicación.
- No se realizará la implantación del sistema en la Institución Educativa María Antonia Ruiz, ya que la institución se encuentra en el proceso de adquirir un servidor propio mediante el Ministerio de Educación.

1.5 ESTRUCTURA DEL DOCUMENTO

El resto del presente documento está organizado de la siguiente manera:

En el capítulo 2, se describe el marco de referencia, donde se presenta los conceptos relacionados a la aplicación y una descripción de la Institución Educativa.

En el capítulo 3 se describe la fase de desarrollo del proyecto, donde se presenta las etapas de análisis, diseño e implementación del producto de software.

En el capítulo 4 se presenta las pruebas del sistema y los resultados logrados.

El documento finaliza con el capítulo 5, donde se exponen las conclusiones obtenidas del proyecto y los trabajos futuros.

2. MARCO DE REFERENCIA

2.1 MARCO TEÓRICO

2.1.1 Ingeniería Web

En la última década se ha asistido al crecimiento vertiginoso del desarrollo y uso de aplicaciones y sistemas Web cada vez más complejos y sofisticados. Desafortunadamente, dicha complejidad no parece estar acompañada de los mecanismos adecuados que garanticen la calidad de unos sistemas de los que cada día tenemos mayor dependencia a nivel social, funcional y económico. Esto ha venido generando una preocupación creciente entre la comunidad científica y técnica involucrada en el desarrollo Web.

De esta forma surge el nacimiento de una disciplina denominada Ingeniería Web que es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad [1]. Esta breve definición nos lleva a abordar un aspecto clave de cualquier proyecto como es: determinar qué tipo de proceso es más adecuado en función de las características del mismo [2]. La aplicación de principios de ingeniería pueden evitar el caos potencial, y poner bajo control el desarrollo de las aplicaciones Web, minimizando riesgos y mejorando el mantenimiento y calidad.

La ingeniería de la Web es una rama de la ingeniería del software que afronta temas específicamente relacionados con el diseño y desarrollo a gran escala de aplicaciones Web.

2.1.2 Proceso de ingeniería web

Un proceso incremental y evolutivo en el desarrollo de aplicaciones Web, permite que el usuario se involucre activamente, facilitando el desarrollo de productos que se ajustan mucho lo que éste busca y necesita, las actividades que formarían parte del marco serían aplicables a cualquier aplicación Web, independientemente del tamaño y complejidad de la misma. [3]

Las actividades que forman parte del proceso son: formulación, planificación, análisis, modelado, generación de páginas, test y evaluación del cliente.

La **Formulación** identifica objetivos y establece el alcance de la primera entrega.

La **Planificación** genera la estimación del coste general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.

El **Análisis** especifica los requerimientos e identifica el contenido.

La **Modelización** se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario.

Es importante destacar la importancia del diseño de la interfaz. Independientemente del valor del contenido y servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos.

En la **Generación de páginas** se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de las aplicaciones, las páginas.

El **Test** busca errores a todos los niveles: contenido, funcional, navegación, rendimiento.

2.1.3 Aplicación Web

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. [4]

Ventajas: Proporcionan movilidad, dado que puedes ejecutarlas desde cualquier computador con conexión a Internet. La información que manejan se accede a través de Internet, motivo por el cual son especialmente interesantes para desarrollar aplicaciones multiusuario basadas en la compartición de información.

Desventajas: La comunicación constante con el servidor que ejecuta la aplicación establece una dependencia con una buena conexión a Internet. Además, el servidor debe tener las prestaciones necesarias para ejecutar la aplicación de manera eficiente, no sólo para un usuario sino para todos los que la utilicen de forma concurrente.

Aspectos Generales en Arquitectura Web

- Escalabilidad.
- Separación de responsabilidades.
- Portabilidad.
- Utilización de componentes en los servicios de infraestructura.
- Gestión de la sesión del usuario.
- Aplicación de patrones de diseño.

2.1.3.1 Servidor web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS. Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle: [5]

Un servidor Web puede procesar:

- Procesamiento de páginas Web estáticas.
- Procesamiento de páginas dinámicas.
- Acceso a una base de datos.

2.1.3.2 Navegador web o browser

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

2.1.4 Sistema de Información

Es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo). Dichos elementos formarán parte de alguna de estas categorías:

- Personas
- Datos
- Actividades o técnicas de trabajo
- Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todos estos elementos interactúan entre sí para procesar los datos (incluyendo procesos manuales y automáticos) dando lugar a información más elaborada y distribuyéndola de la manera más adecuada posible en una determinada organización en función de sus objetivos. [6]

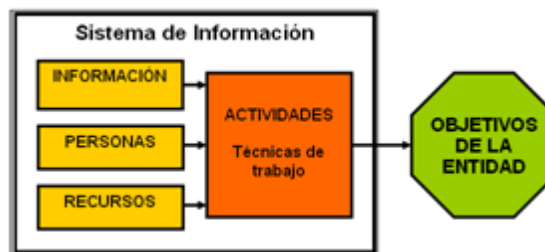


Figura 1. Elementos de un Sistema de Información

2.1.5 PHP

PHP (acrónimo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML.

Php es un lenguaje de programación del lado del servidor, es independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación., esto hace que cualquier sistema pueda ser compatible con el lenguaje. El fácil uso y la similitud con los lenguajes más comunes de programación estructurada permiten a la mayoría de los programadores crear aplicaciones complejas. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas. [7]

2.1.6 MYSQL

Es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL /GNU (Licencia Pública General). Está basado en un lenguaje de consultas estructurado (SQL), el cual es usado para agregar, eliminar y modificar información de la base de datos.

Este gestor de bases de datos es probablemente el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración en los servidores web. [8]

2.1.7 Framework

Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. [9]

2.1.8 UML

El Lenguaje de Modelado Unificado es una de las herramientas de mayor ayuda en el proceso de diseño de un sistema, ya que a través de este se elaboran modelos fácilmente comprensibles tanto para analistas y desarrolladores como para clientes. Un modelo UML indica que es lo que supuestamente hará el sistema.

Cuenta con un conjunto de elementos gráficos, que son combinados entre sí para dar paso a los diagramas, los cuales tienen como finalidad la presentación de diversos modelos o perspectivas de un sistema. [10]

2.1.9 Modelo vista controlador

La lógica Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página y el código que provee de datos dinámicos a la página, el modelo es el sistema de gestión de base de datos y el controlador representa los eventos de la aplicación y la comunicación entre la Vista y el Modelo. [11]

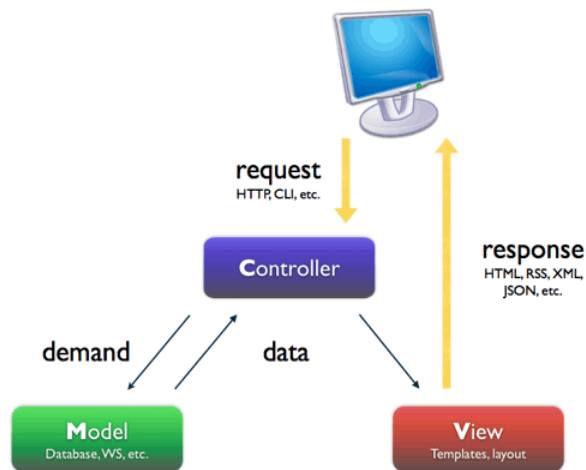


Figura 2. Modelo Vista Controlador (MVC)

2.2 MARCO CONTEXTUAL

“La Institución Educativa “María Antonia Ruiz” fue creada en el año 1939 por decreto N°. 1939 del MEN (Ministerio de Educación nacional) como escuela oficial en la parte urbana del Municipio de Tuluá (Valle del Cauca) de niñas en el barrio Victoria (de 1º a 5º de primaria).

Actualmente la Institución Educativa “María Antonia Ruiz “ofrece los servicios de básica primaria, secundaria, media técnica, con especialidad en informática.

Está localizada en la Calle 29 # 31-51 del barrio Victoria del municipio de Tuluá (Valle del Cauca).

La Institución está consolidada por siete sedes llenas de leyendas y tradición educativa.

- ✓ El Jazmín
- ✓ Francisco José de Caldas
- ✓ San Bautista de la Salle
- ✓ Escuela de Formación Deportiva Osmiro Colonia Rodríguez
- ✓ Jesús Maestro
- ✓ Laboratorio Integrado de Ciencias
- ✓ Sede Central María Antonia Ruiz.²

Actualmente la Institución cuenta con una Misión y Visión que representa el compromiso con la comunidad educativa.

² Tomado de la revista del institución

MISIÓN

“La Institución Educativa, tiene como misión formar niños y jóvenes competentes y felices a través del desarrollo de competencias básicas para asumir su proyecto de vida y contribuir a una sociedad mejor.

Formar personas competentes y felices a través del desarrollo de conocimientos, habilidades, aptitudes y valores necesarios para posibilitar la realización armónica y coherente de su propia personalidad; que les permita desempeñarse con excelencia en la vida productiva, como seres honestos y autónomos que aportan a la construcción de un mundo mejor.”³

VISIÓN

“La Institución Educativa para el año 2.020 habrá logrado consolidar todos sus procesos mediante el mejoramiento de la calidad educativa en los siguientes aspectos:

Con relación a los resultados académicos: Logrará obtener un desempeño alto en las pruebas externas: pruebas SABER y pruebas de ESTADO.

Con relación a la ciencia y la tecnología: Fortalecería los semilleros de investigación como un medio para acceder al conocimiento en forma crítica y científica.

Utilizará las tecnologías de información como herramientas para acceder a los conocimientos productivos pro la humanidad y la utilización adecuada de estos.

Con relación a la promoción: Promoverá a la mayoría de los estudiantes dotándolos de las competencias necesarias para asumir las etapas educativas posteriores que le corresponda.

Con relación a la convivencia: Logrará un buen nivel de convivencia entre sus miembros, constituyéndola en objeto del proceso educativo.”⁴

³ Tomado de la revista del institución

⁴ Tomado de la revista del institución

2.3 MARCO CONCEPTUAL

▪ REGISTRO ACADÉMICO

Tiene la finalidad de permitir el ingreso, procesamiento y recuperación de datos de los estudiantes que se manejan en cada curso o grado de la Institución Educativa María Antonia Ruiz, con el propósito de mecanizar el procesamiento de la matrícula, notas, reportes, boletines, certificado de estudio, carnet de una manera efectiva y eficiente.

▪ MATRÍCULA ACADÉMICA

La matrícula es el acto voluntario de una persona natural mediante el cual adquiere la calidad de estudiante de la Institución, previo el cumplimiento de todos los requisitos señalados por la misma. A través de este acto, la Institución otorga al estudiante matriculado, el derecho a cursar el grado para el respectivo período y el estudiante se compromete a cumplir el presente reglamento y las demás normas establecidas por la Institución.

▪ ASIGNACIÓN ACADÉMICA

Es el proceso en el cual se asignan las materias que los docentes dictarán dependiendo del curso al que pertenezcan y el periodo académico

▪ PERIODO ACADÉMICO

Conjunto sucesivo de semanas que de manera autónoma define y planea la Institución Educativa, en el que se desarrolla un ciclo completo de los procesos académicos y administrativos. Al inicio de este los estudiantes deben revalidar su matrícula y al final se evalúa, confirma y actualiza la condición y el estado académico de sus estudiantes.

▪ LOGRO:

El logro representa el resultado que debe alcanzar el estudiante al finalizar la asignatura, el resultado anticipado por supuesto, las aspiraciones, propósitos, metas, los aprendizajes esperados en los estudiantes, el estado deseado, el modelo a alcanzar, tanto desde el punto de vista cognitivo como personal y social –

motivacional (el aprender y aprender a hacer, el aprender a ser y aprender a convivir).

- **CARNET**

Documento de carácter personal que indica la identidad del estudiante, la cual contiene información básica (Nombre de la institución, foto, año lectivo, nombre del estudiante, identificación, RH, grado, domicilio, teléfono, ocupación, fecha vencimiento).

- **SEDE**

Lugar donde está situada la Institución Educativa, en la cual se desarrollan las actividades académicas y administrativas.

2.4 MARCO LEGAL

Luego de conocer los procesos y formatos de registro académico mediante reuniones con el personal administrativo de la institución educativa María Antonia Ruiz, apoyados bajo el Pacto de convivencia aprobado por resolución 1684 de 10 Octubre de 2001 y fusionado según resolución N° 1841 5 de Septiembre de 2002. Resolución N° 310-056.314 de Junio 18 de 2008 S.E.M (Secretaría de Educación Municipal), se pudo conocer normas y reglamentos internos para llevar a cabalidad el correcto funcionamiento de los procesos.

Después de conocer la estructura del pacto de convivencia, se va a mencionar los artículos que aportaran en el desarrollo del proceso de registro académico.

Título I.

La institución Educativa; criterios y políticas institucionales

Capítulo II. Condiciones de ingreso, permanencia y retiro de estudiantes.

Artículo 4 LAS CONDICIONES DE INGRESO a la institución educativa están regidas por el acuerdo de prestación de servicio educativo entre la institución y el padre de familia o acudiente e beneficio del menor, condicionado por el numeral C del fallo de la corte constitucional tutela 534 de 1994 que define " el derecho a la educación no es absoluto, es derecho- deber y el derecho-deber exigido a todos los DEBERES" el estudiante debe cumplir las siguientes condiciones para que sea estudiado su ingreso:

- a) Inscribirse de acuerdo a las condiciones preestablecidas por la institución
- b) Presentar los documentos autenticados y legales que exige la institución y no incurrir fraudes para diligenciar o resolver una cuestión escolar.
- c) Presentar entrevista con la Comisión de Admisión.
- d) Estar respaldado con un acudiente para asistirlo en las situaciones que lo requiera o cuando la institución lo solicite.

Artículo 8 La matrícula es el acto que formaliza la vinculación del estudiante al servicio educativo, esta se realiza por una sola vez en la institución a través del SIMAT y la firma del documento de matrícula debe hacerse por el estudiante, padre de familia y el rector del establecimiento educativo, se pueden hacer renovaciones para cada año lectivo como lo indica el artículo 95 de la ley 115.

Artículo 9 La matrícula es un acuerdo que obliga a los padres de familia, acudientes y estudiantes a dar cumplimiento con todas las normas vigentes en este PACTO DE CONVIVENCIA y la ley 1098 de 2006 nuevo código de la infancia y la adolescencia.

Artículo 10 PROCEDIMIENTO PARA REALIZAR LA MATRÍCULA: La matrícula es el acto por el cual el aspirante admitido adquiere la calidad de estudiante de la institución educativa, esta se legaliza con la ficha de matrícula y su respectivo registro en el SIMAT la cual debe llevar las firmas de la Rectora, del padre de familia o acudiente y del estudiante. Serán requerimientos del procedimiento para realizar la matrícula los siguientes aspectos.

1. Entregar diligenciada la documentación y todos los requisitos solicitados en el proceso de inscripción y admisión establecidas en el pacto de convivencia.
2. Conocer y aceptar en su totalidad el presente manual o pacto de convivencia por parte de los padres de familia o acudientes de los estudiantes.
3. No estar reportado en otra institución educativa según el SIMAT como estudiante activo.
4. Estar inscrito previamente durante las fechas establecidas para dicho acto de pre-matrícula.
5. Presentarse al acto de matrícula de manera personal con su acudiente.
6. Presentar buena convivencia social en el observador del estudiante de donde procesa, en caso contrario deberá firmar el documento de matrícula en observación y cumplir con las exigencias de la institución.
7. Estar a paz y salvo académico, económico y con los bienes de la institución o de la que proceda.

Título III.

Sistema Educativo, Académico, Pedagógico Y Curricular

Capítulo I Aspectos Académicos del Sistema Educativo.

Artículo 57 La institución Educativa programa 40 semanas lectivas efectivas de estudio como lo ordenan las autoridades educativas en las resoluciones departamentales o municipales y como lo exige el decreto 1850.

Artículo 58 El año lectivo estará dividido en 4 periodos escolares de 10 semanas hábiles cada uno y se entregará a los padres de familia cinco informes descriptivos de valoración o boletines durante el año.

Artículo 60 Los estándares básicos de competencia han clasificado los niveles escolares de educación en grupos de grados para articular y desarrollar los respectivos planes de estudios así:

- a) Grupo de grados 1º, 2º, 3º de educación básica primaria.
- b) Grupo de grados 4º, 5º de educación básica primaria.
- c) Grupo de grados 6º, 7º de educación básica secundaria.
- d) Grupo de grados 8º, 9º de educación básica secundaria.
- e) Grupo de grados 10º, 11º de educación media.

Capítulo II Criterios de Valoración y Evaluación Integral Institucional.

Artículo 73 La escala de expectativas de la Institución Educativa, con respeto al decreto 1290, surge del debate y acuerdo realizado con toda la comunidad educativa a través de la evaluación que facilita el desarrollo de las competencias y desempeños cognitivos, personales y sociales así:

Su. → Desempeño Superior
 Al. → Desempeño Alto
 Bs. → Desempeño Básico
 Bj. → Desempeño Bajo

Artículo 74 Las valoraciones o desempeños descriptivos que la Institución educativa María Antonia Ruiz utilizará para valorar el desempeño de los educandos serán clasificados en FORTALEZAS y DIFICULTADES, estas categorías de valoración sean definidas así a partir de los desempeños:

FORTALEZAS ⇒ Su - Al - Bs

DIFICULTADES ⇒ Bj

ESCALA NACIONAL	ESCALA INSTITUCIONAL	ESCALA EN RANGOS	
		Literales	Rangos
Desempeño Superior	SUPERIOR	Su	4.6 – 5.0
Desempeño Alto	ALTO	Al	4.0 – 4.5
Desempeño Básico	BÁSICO	Bs	3.0 – 3.9
Desempeño Bajo	BAJO	Bj	1.0 – 2.9

Artículo 76 Para determinar la evaluación en el alcance de cualquiera de los logros previstos en el plan de estudios, todos los docentes manejarán los mismos instrumentos de evaluación o planillas aplicando para su evaluación integral los términos que propone el Decreto 1290:

1. Evaluarán los desempeños cognitivos o académicos
2. Evaluarán los desempeños personales
3. Evaluarán los desempeños sociales

Artículo 77 La evaluación de los estudiantes será continua e integral como lo exige el decreto 1290 del 2009 y se hará con referencia a cuatro periodos de igual duración en los que se dividirá el año escolar.

Capítulo III Criterios de Promoción Institucional

Artículo 104 Realizarán estrategias pedagógicas de apoyo al estudiante, aquellos casos que presenten debilidades en el proceso formativo y cuyas áreas obtengan niveles de desempeño BAJO en cualquiera de los periodos escolares programados o al finalizar el año lectivo.

3. DESARROLLO DEL PROYECTO

METODOLOGÍA DE DESARROLLO

De acuerdo al análisis de las distintas metodologías de desarrollo de software, se concluyó que para la realización del producto se debe utilizar una metodología de Desarrollo Ágil, entre las cuales se encuentran las metodologías XP, SCRUM, ICONIX, DSDM.

Para este proyecto en particular, se utilizó la metodología ICONIX, ya que es la mejor opción para el desarrollo de aplicaciones web y se adapta a los recursos y limitaciones de tiempo con que se cuenta.

A continuación se describe las fases de desarrollo de ICONIX que se aplicaron en el proyecto.

ICONIX [12]

Se define como un “proceso” de desarrollo de software práctico. ICONIX está entre la complejidad del RUP (Rational Unified Processes) y la simplicidad y pragmatismo del XP (Extreme Programming), sin eliminar las tareas de análisis y de diseño que XP no contempla.

Las tres características fundamentales de ICONIX son:

- ✓ **Iterativo e incremental:** varias iteraciones ocurren entre el desarrollo del modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.
- ✓ **Trazabilidad:** cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos.
- ✓ **Dinámica del UML:** La metodología ofrece un uso “dinámico del UML” como los diagramas del caso de uso, diagramas de secuencia y de colaboración.

El método cubre las fases del ciclo del desarrollo del software.

1. Análisis de Requisitos

- a) Identificar en el “mundo real” los objetos y todas las relaciones de agregación y generalización entre ellos. Utilizar un diagrama de clases de alto nivel definido como **modelo de dominio**.
- b) Presentar, si es posible, un **prototipo** rápido de las interfaces del sistema, los diagramas de navegación, etc., de forma que los clientes puedan comprender mejor el sistema propuesto.
- c) Identificar los casos de uso del sistema mostrando los actores involucrados. Utilizar para representarlo el **modelo de casos de uso**.
- d) Organizar los casos de uso en grupos, o sea, utilizar los **diagramas de paquetes**.
- e) Asociar los requisitos funcionales con los casos de uso y con los objetos del dominio (trazabilidad).

2. Análisis y Diseño Preliminar

- a) Describir los casos de uso, como un flujo principal de acciones, pudiendo contener los flujos alternativos y los flujos de excepción. La principal sugerencia de ICONIX, en esta actividad, es que no se debe perder mucho tiempo con la descripción textual. Debería usarse un estilo consistente que sea adecuado al contexto del proyecto.

3. Diseño

- a) Especificar el comportamiento a través del **diagrama de secuencia**. Para cada caso de uso identificar los mensajes entre los diferentes objetos. Es necesario utilizar los diagramas de colaboración para representar la interacción entre los objetos.
- b) Terminar el modelo estático, adicionando los detalles del diseño en el **diagrama de clases**.
- c) Verificar si el diseño satisface todos los requisitos identificados.

4. Implementación

- a) Utilizar el diagrama de componentes, para apoyar el desarrollo. Es decir, mostrar la distribución física de los elementos que componen la estructura interna del sistema.
- b) Escribir/ Generar el código.
- c) Realizar pruebas. Test de unidades, de casos, datos y resultados. Test de integración con los usuarios para verificar la aceptación de los resultados.

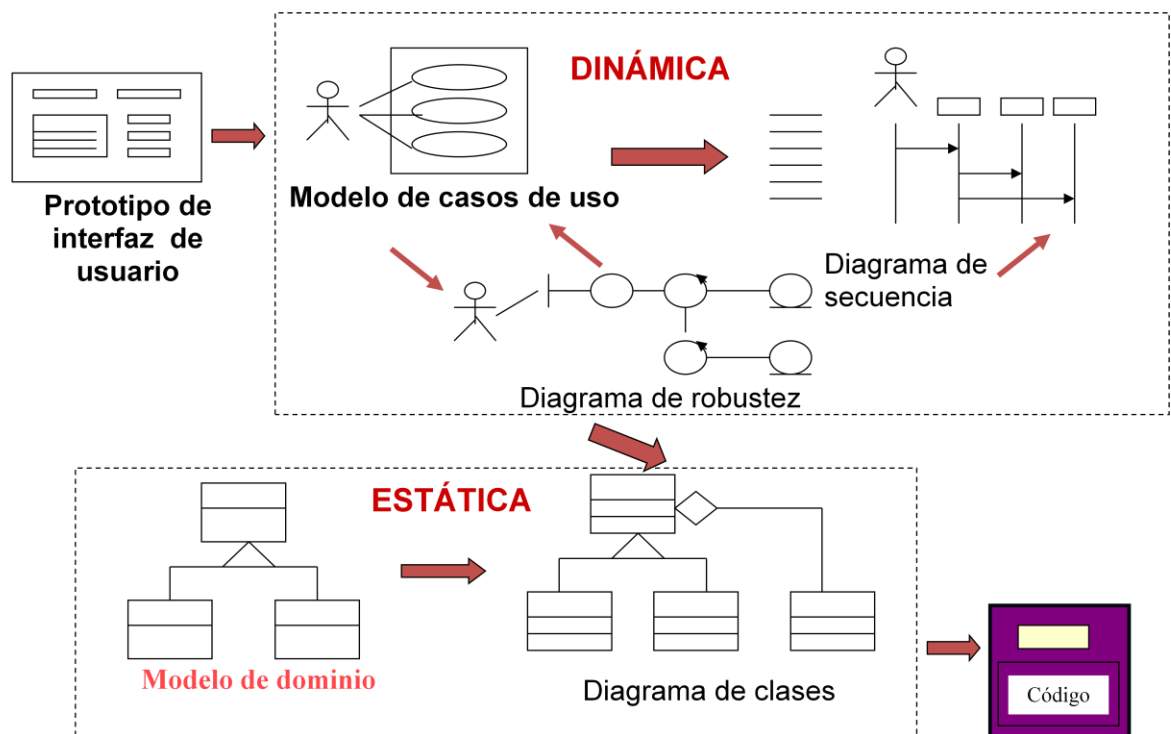


Figura 3. Metodología ICONIX

3.1 ANÁLISIS DE REQUISITOS

En esta etapa se realizó la evaluación de la problemática, la definición de requerimientos, se planteó una solución de software para el problema, se definió el Alcance Funcional de la aplicación.

Todo esto se realizó por medio de actividades conjuntas con el cliente (entrevistas, presentaciones), además de una investigación del proceso de Registro académico que se ejecuta en la institución educativa, las normas que lo rigen, los formatos y plantillas que se emplean para el registro.

La información obtenida se seleccionó y se documentó en este capítulo, en el Marco de Referencia Capítulo 2, página 14 y los formatos y plantillas se registraron en el documento **Anexo A**. Formatos y Plantillas entre los cuales se encuentra el formato de Registro de Notas.

PLANILLA DE EVALUACION INTEGRAL.

INSTITUCION EDUCATIVA:

MARIA ANTONIA RUIZ

Sede:

ESCUELA DEPORTIVA OSMIRO COLONIA

Docente:

HECTOR GARCIA RAMIREZ

Area:

CIENCIAS NATURALES Y EDUCACION AMBIENTAL

Asignatura:

CIENCIAS NATURALES Y EDUCACIÓN AMBIENTAL

Grupo:

NOVENO 5

Año lectivo:

2011-02-07 - 2011-11-27

Periodo:

2

CRITERIOS DE VALORACION INSTITUCIONAL

ESCALA INSTITUCIONAL	RANGO
Su SUPERIOR	4.6 - 5
Al ALTO	4 - 4.5
Bs BASICO	3 - 3.9
Bj BAJO	1 - 2.9

CRITERIOS DE EVALUACION INSTITUCIONAL

COGNITIVO	PERSONAL	SOCIAL
AZTDE01 Apropiación de Ap	AZTDE08 Asistencia.	AZTDE15 Tolerancia.
AZTDE02 Solución de Probl	AZTDE09 Calidad Ética.	AZTDE16 Solidaridad.
AZTDE03 Dominio de las TI	AZTDE10 Responsabilidad	AZTDE17 Equidad e Inclusi
AZTDE04 Atención en Clase	AZTDE11 Respeto a la dife	AZTDE18 Convivencia Pacíf
AZTDE05 Tareas, talleres	AZTDE12 Identidad Cultural	AZTDE19 Trabajo en equipo
AZTDE06 Participación y c	AZTDE13 Presentación Pers	AZTDE20 Mediación Escolar
AZTDE07 Apoyo y gu	AZTDE14 Sentido de	AZTDE21 Conservación del

ESTUDIANTES

	VALORACION INTEGRAL DE DESEMPEÑOS POR PERIODO										TOTAL AUSENCIAS	VALORACION INTEGRAL DEFINITIVA																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
	COGNITIVO			PERSONAL	SOCIAL	CONTROL DE AUSENCIAS				Periodo Anterior		Valoración Actual																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
	APRENDER	APRENDER	APRENDER A HACER	APRENDER A SER	APRENDER A CONVIVIR	S1	S2	S3	S4	S5		S6	S7	S8	S9	S10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
1	AGUDELO ESPITTA JEFFREY ALEXANDER																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															

3.1.1 Análisis de Usuarios y Fases

Propósito:

El presente documento tiene como objetivo definir y describir las funcionalidades que requiere la **Aplicación Web para el Registro Académico de la Institución Educativa María Antonia Ruiz**. De igual manera en este capítulo se pretende explicar claramente los atributos y las características necesarias que debe tener la aplicación para realizar adecuadamente dichos procesos.

Alcance:

El producto a desarrollar en este proyecto se identifica con el nombre de “Sistema de Información María Antonia Ruiz” (**SIMAR**)

SIMAR será una aplicación web basada en la arquitectura Cliente/Servidor, que permitirá registrar y realizar de una manera organizada y controlada el registro académico de la Institución.

Dentro de las pretensiones del proyecto, se espera que la aplicación permita cumplir con las siguientes especificaciones:

- El sistema solo permitirá realizar el registro académico de la sede Central de la Institución, parámetros/configuración, matrícula académica, registro de notas, generación de reportes (fichas de matrícula, listado de notas, certificado de estudio, carnet estudiantil).
- Gestionar la información de los Usuarios que ingresarán al sistema. Crear, modificar, eliminar y visualizar los registros de usuario.
- Gestionar la información de los Grupos o cursos de la institución. Crear, modificar, eliminar y visualizar.
- Gestionar la información de los Docentes. Crear, modificar, eliminar y visualizar los registros.
- Gestionar la información de las Áreas de conocimiento en las que se clasifican las materias. Crear, visualizar, modificar y eliminar.
- Gestionar la información de las Materias de los grupos. Crear, visualizar, modificar y eliminar.

- Gestionar los Logros Académicos de las materias por grado para cada uno de los periodos académicos.
- Configurar la Asignación académica de los Docentes y de los Cursos de la Institución.
- Realizar la Matrícula académica de los Estudiantes.
- Gestionar la información de los Estudiantes. visualizar, modificar y eliminar los registros de los estudiantes.
- Registrar las Notas de las materias de los estudiantes por cada periodo académico.
- Generar Reportes con base en la información del Registro académico (Ficha de matrícula, listado de notas por materia, certificado de estudio y carnet).
- Permitir descargar los Reportes en formato digital (.pdf).

3.1.2 Detalles del Sistema

Perspectiva:

Debido a la naturaleza de este Proyecto de Ingeniería de Software, **SIMAR** es un sistema independiente de otros productos, ya que se convertirá en la primera implementación sistematizada que refiere al registro académico de la Institución Educativa María Antonia.

Sin duda, este proyecto modificará positivamente el proceso administrativo y de gestión del registro académico que se realiza actualmente en Institución Educativa, pues con la implementación de **SIMAR** se podrá reducir algunos gastos de papelería, se mejorará la atención a los usuarios, se reduce el tiempo de ejecución de los procesos y estará disponible para toda la comunidad educativa.

Características del Usuario:

Los usuarios dentro del sistema se organizan jerárquicamente de la siguiente manera:



Figura 5. Jerarquía de los Usuarios del Sistema

Como puede observarse en la figura anterior:

El Administrador es el tipo de usuario de mayor importancia, ya que está autorizado para acceder a todas las funciones del sistema, siendo las más relevantes: Gestionar los usuarios y los parámetros de configuración del sistema. Este tipo de Usuario sería la Rectora y el jefe de sistemas de la institución.

La Secretaria Académica es el segundo tipo de usuario de mayor relevancia, ya que está autorizado para acceder a funciones del Registro académico, siendo las más relevantes: realizar la matricula académica, gestionar la información de los estudiantes y generar certificado de estudio y carnet estudiantil.

El Docente puede acceder a la función Registro de notas de las asignaturas que dicta por cada periodo.

El Estudiante es el tipo de usuario de más bajo nivel, pues interactúa únicamente con el sistema, para visualizar las notas de las materias del curso, generar su certificado de estudio y carnet estudiantil.

3.1.3 Requerimientos

Los requerimientos del proyecto se dividen en Requerimientos funcionales y no funcionales.

Para ver la información completa sobre los requerimientos, remítase al **Anexo B. Especificación de Requerimientos de Software** (Página 10).

3.1.3.1 Requerimientos Funcionales

Los requerimientos funcionales se han dividido por cada módulo del sistema. Esto con el fin de que tanto el usuario como el equipo de desarrollo tengan una mejor comprensión de los requerimientos del sistema. Se presenta un resumen de los requerimientos:

Módulo: Usuarios

<i>Requerimientos Funcionales Usuarios</i>	
Req.1.1.1	El sistema deberá permitir el ingreso a la aplicación, sólo a los usuarios que estén registrados, para poder acceder a los Módulos que le ofrece el sistema.
Req.1.2.1	El sistema deberá permitir crear y clasificar los usuarios según el perfil (Administrador, Secretaria académica, Docente, Estudiante) que accederán a la aplicación.
Req.1.2.2	El sistema deberá permitir consultar y visualizar la información de los usuarios que están registrados, por: código o todos.
Req.1.2.3	El sistema deberá permitir actualizar o modificar la contraseña de usuario para ingresar a la aplicación.
Req.1.2.4	El sistema deberá permitir eliminar los usuarios que están registrados en la aplicación.

Tabla 1. Requerimientos Módulo Usuarios

Módulo: Parámetros/Configuración

<i>Requerimientos Funcionales Configuración</i>	
Req.2.1.1	El sistema deberá permitir configurar el año lectivo que iniciará la institución, con los respectivos periodos académicos.
Req.2.2.1	El sistema deberá permitir crear los grados o niveles escolares que ofrece la institución estipulados en el pacto de convivencia.
Req.2.2.2	El sistema deberá permitir consultar y visualizar la información de los grados.
Req.2.2.3	El sistema deberá permitir actualizar o modificar la información los grados cuando se requiera.
Req.2.2.4	El sistema deberá permitir eliminar los grados que están registrados.

Req.2.3.1	El sistema deberá permitir crear o registrar los grupos o cursos que tiene la institución.
Req.2.3.2	El sistema deberá permitir consultar y visualizar la información de los grupos que estén registrados.
Req.2.3.3	El sistema deberá permitir actualizar o modificar algunos datos de los grupos.
Req.2.3.4	El sistema deberá permitir eliminar los grupos que están registrados.
Req.2.4.1	El sistema deberá permitir crear o registrar la información de los docentes que trabajan en la institución y asignarle las materias que dictarán.
Req.2.4.2	El sistema deberá permitir consultar y visualizar la información de los docentes que estén registrados, por: código, nombre, número del documento o todos.
Req.2.4.3	El sistema deberá permitir actualizar o modificar la información de los docentes cuando se requiera. Y además se podrá modificar las materias que dicta el docente.
Req.2.4.4	El sistema deberá permitir eliminar la información de los docentes que están registrados.
Req.2.5.1	El sistema deberá permitir crear o registrar las áreas del conocimiento, que permite clasificar las materias.
Req.2.5.2	El sistema deberá permitir consultar y visualizar la información de las áreas registradas.
Req.2.5.3	El sistema deberá permitir actualizar o modificar los datos de las áreas cuando se requiera.
Req.2.5.4	El sistema deberá permitir eliminar las áreas que están registrados.
Req.2.6.1	El sistema deberá permitir crear o registrar los Materias que se dictan en cada grupo o curso, y clasificarlas según el área de conocimiento.
Req.2.6.2	El sistema deberá permitir consultar y visualizar la información de las materias que estén registrados, por: nombre, código, curso o todas.
Req.2.6.3	El sistema deberá permitir actualizar o modificar algunos datos de las materias cuando se requiera, estos datos son: nombre, grupo o descripción.
Req.2.6.4	El sistema deberá permitir eliminar las materias que están registrados.

Tabla 2. Requerimientos Módulo Parámetros/Configuración

Módulo: Registro Académico

<i>Requerimientos Funcionales Registro académico</i>	
Req.3.1.1	El sistema deberá permitir realizar la matrícula académica, para registrar un nuevo estudiante y matricularlo en un curso.
Req.3.1.2	El sistema deberá permitir consultar y visualizar la información de las matrículas generadas.
Req.3.2.1	El sistema deberá permitir crear o registrar un estudiante en la aplicación, a través de la matrícula académica
Req.3.2.2	El sistema deberá permitir consultar y visualizar la información de los estudiantes que estén registrados, por: nombre, código, número del documento o cursos.
Req.3.2.3	El sistema deberá permitir actualizar o modificar la información de los estudiantes cuando se requiera.
Req.3.2.4	El sistema deberá permitir eliminar los estudiantes que están registrados.

Req.3.3.1	El sistema deberá permitir en cada periodo del año lectivo registrar o crear las Notas de las materias que cursan los estudiantes en cada grupo, cumpliendo con las condiciones estipuladas en el Pacto de Convivencia de la Institución (criterios de evaluación, rango, número de ausencias, nota final y nota de recuperación). Para las notas, se tendrá 3 criterios de evaluación donde cada uno equivale al 33,3%, académico o cognitivo, social y personal.
Req.3.3.2	El sistema deberá permitir consultar y visualizar las Notas, por materias.
Req.3.3.3	El sistema deberá permitir actualizar o modificar los valores de las Notas de las materias, cuando sea necesario.

Tabla 3. Requerimientos Módulo Registro académico

Módulo: Reportes

Requerimientos Funcionales Reportes	
Req.4.1.1	El sistema deberá permitir generar y descargar los Reportes de Fichas de matrícula, listados de Notas por materia de los Estudiantes, certificado de estudio y el carnet estudiantil en un archivo en formato digital (.PDF).

Tabla 4. Requerimientos Módulo Reportes

3.1.3.2 Requerimientos No Funcionales

1. Requerimientos de Interfaz de Usuario	
R 1.1	La aplicación deberá ofrecer las opciones de Cerrar Sesión y Regresar a la página o menú principal.
R 1.2	La aplicación deberá validar las entradas de valores numéricos o alfanuméricos por parte del usuario durante el diligenciamiento de formularios.
R 1.3	La aplicación deberá proporcionar títulos claros y descriptivos en todas las páginas para referenciar los procesos que el usuario este realizando.
R 1.4	La aplicación deberá desplegar mensajes de confirmación, éxito o fracaso para mantener al usuario enterado de cualquier proceso que se realice.

Tabla 5. Requerimientos de Interfaz de Usuario

2. Requerimientos Tecnológicos	
2.1 Software	
R 2.1.1	La aplicación debe ser desarrollada con el motor de base de datos MySQL.
R 2.1.2	La aplicación se desarrollará mediante el framework Adobe Flex y el lenguaje PHP
R 2.1.3	Para la ejecución de la aplicación es necesario un servidor Web con soporte para PHP (Apache).

Tabla 6. Requerimientos de Software

2.2 Hardware	
R 2.2.1 Desarrollador - Implementación	R 2.2.2 Cliente - Servidor
Fabricante <i>Equipo: HP Pro-Book 4420s</i>	Fabricante: <i>COMPAQ CQ50-103LA</i>
Procesador: Intel Core i3	Procesador: <i>AMD Dual Core QL60 1.90Ghz</i>
Disco Duro: 500GB	Disco Duro: 160 GB
Memoria RAM: 3.00 GB	Memoria RAM: 2.00 GB
Tarjeta de Red: <i>Ethernet 10/100Mbps</i>	Tarjeta Red: <i>Ethernet 10/100Mbps</i>
Wireless: LAN 802.11b/g/n	Wireless: LAN 802.11n
R 2.2.3 Desarrollador – Pruebas	R 2.2.4 Cliente - Usuario
Fabricante <i>Equipo: COMPAQ CQ50-103LA</i>	Fabricante: <i>COMPAQ CQ50-103LA</i>
Procesador: <i>AMD Dual Core QL60 1.90Ghz</i>	Procesador: <i>AMD Dual Core QL60 1.90Ghz</i>
Disco Duro: 160 GB	Disco Duro: 160 GB
Memoria RAM: 2.00 GB	Memoria RAM: 2.00 GB
Tarjeta de Red: <i>Ethernet 10/100Mbps</i>	Tarjeta Red: <i>Ethernet 10/100Mbps</i>
Wireless: LAN 802.11n	Wireless: LAN 802.11n
R 2.2.5 Otros	
Internet Banda Ancha	
Impresora	

Tabla 7. Requerimientos de Hardware

3. Requerimientos de Plataforma	
R 3.1	La aplicación debe funcionar en cualquier Sistema operativo (Windows, Linux o Mac OS) y en cualquier navegador Web (Google Chrome, Mozilla Firefox, Internet Explorer, etc.), que soporte la plataforma Adobe flash player.

Tabla 8. Requerimientos de Plataforma

4. Requerimientos de Seguridad	
R 4.1	La aplicación deberá permitir al usuario ingresar al sistema por medio de una cuenta de acceso (código y contraseña).
R 4.2	La aplicación deberá encriptar la contraseña de los usuarios.
R 4.3	La aplicación debe permitir solo el ingreso de usuarios que estén registrados en el sistema.

Tabla 9. Requerimientos de Seguridad

5. Requerimientos Confiabilidad y Disponibilidad	
R 5.1	La aplicación será administrada por la rectora, jefe de sistemas y la secretaria académica de la Institución.
R 5.2	Las funciones básicas de los formularios serán validadas.
R 5.3	La aplicación posee una Base de datos robusta que le permite al usuario almacenar, modificar, consultar o eliminar los datos correctamente.

Tabla 10. Requerimientos de Confiabilidad y Disponibilidad

3.1.4 Modelo de Dominio

Con el modelo de dominio se identificaron los objetos y entidades que intervienen en el sistema.

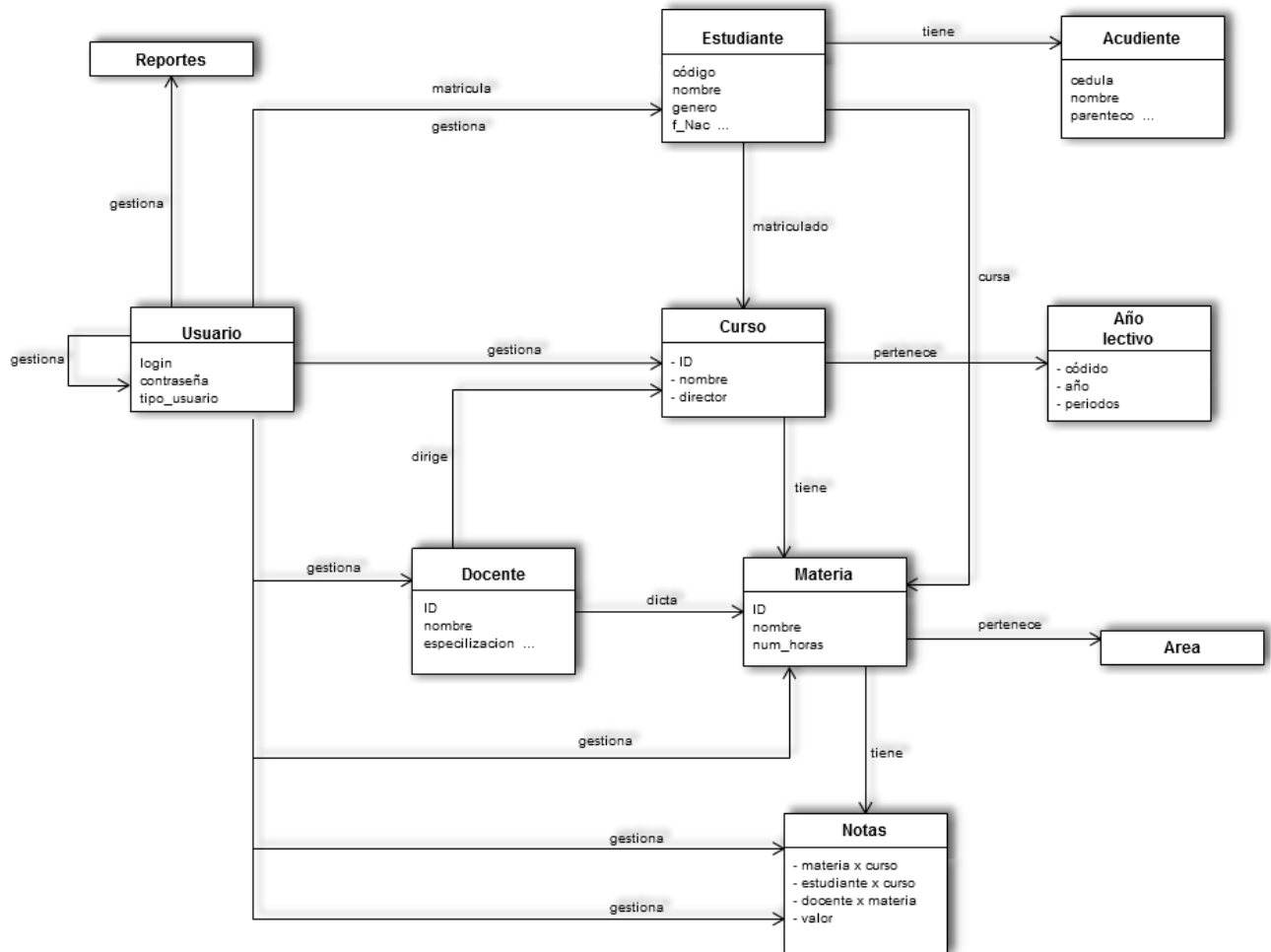


Figura 6. Modelo de Dominio

3.2 ANÁLISIS Y DISEÑO PRELIMINAR

En esta Fase se especificó detalladamente los escenarios y las acciones que realizan los usuarios dentro del sistema

3.2.1 Casos de Uso

3.2.1.1 Diagramas de Casos de Uso:

Para ver los diagramas completos de los Casos de Uso, remitirse al **Anexo B. Especificación de Requerimientos de Software** (Página 29).

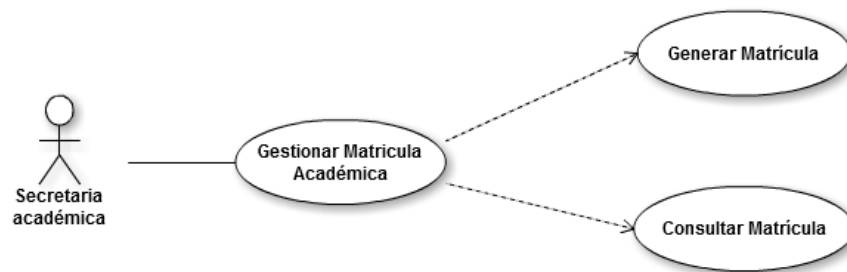


Figura 7. Caso de Uso Gestionar Matrícula Académica

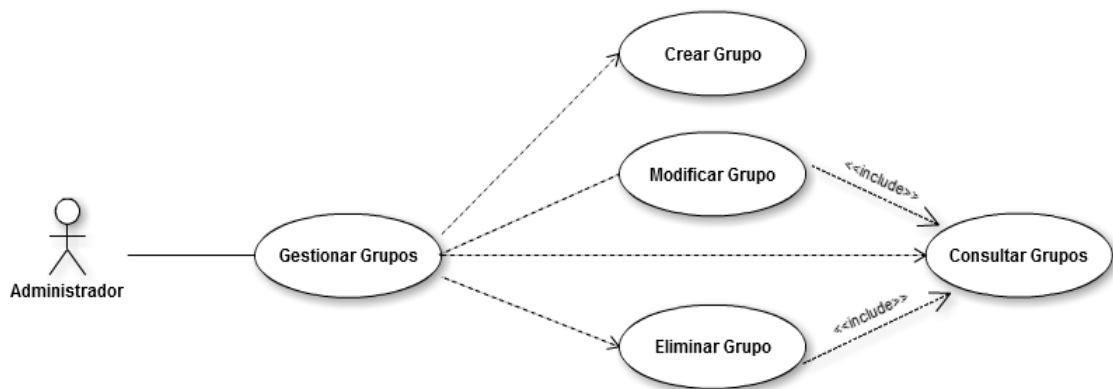



Figura 8. Caso de Uso Gestionar Grupos

3.2.1.2 Descripción de Casos de Uso

Para ver los escenarios de uso completos del sistema, remitirse al **Anexo B**. Especificación de Requerimientos de Software (Página 33).

	SIMAR	Id: CU-034
Caso de Uso: Generar Matrícula		Página #: 63/75 Fecha: 2012-08-20

Documentos de Referencia

Id	Descripción	Formato
	Ficha de Matrícula	

INFORMACIÓN GENERAL


Actores:	<ul style="list-style-type: none"> Secretaria académica
Propósito:	El sistema permitirá registrar un nuevo Estudiante y lo matriculará en un Grupo
Precondiciones:	<ul style="list-style-type: none"> Ingresar al sistema. CU-001
Postcodiciones:	Se guarda la información en una Ficha de Matrícula
Referencias Cruzadas:	R.3.1.1

Flujo Normal de los Eventos	
Acción del Actor	Respuesta del sistema
1. Se dirige al módulo Registro Académico y selecciona la opción "Matricula Académica".	
	2. Despliega el formato de matrícula con los campos: Número de ficha, fecha, información del Estudiante (CU-032) y curso a ingresar.
3. Ingresar los datos correspondientes	
4. Selecciona la opción: Registrar matrícula	
	5. Verifica que los datos hayan sido ingresados y valida que estos datos sean correctos.
	6. Verifica que la Ficha de matrícula no esté ya registrada
	7. Almacena el registro en la Base de Datos
	8. Muestra un mensaje indicando que la Matrícula se registró satisfactoriamente.

Flujos de eventos Excepcionales
5. No se ingresa un dato, o algún dato es incorrecto. 5.1 El sistema muestra un mensaje de error e indica cual campo falta o es incorrecto 5.2 Se deben ingresar los datos faltantes o editar los incorrectos.
6. El código de la ficha de matrícula coincide con un registro actual de la base de datos. 6.1 El sistema muestra un mensaje indicando que esa ficha ya se encuentra registrada

7. No se puedo registrar la matrícula por un fallo en la conexión con la base de datos 7.1 El sistema muestra un mensaje indicando la falla, e intenta reanudar la conexión; si no se puede reanudar, la operación se debe iniciar nuevamente
--

Tabla 11. Caso de Uso Generar Matrícula

	SIMAR	Id: CU-016
Caso de Uso: Modificar Grupos		Página #: 54/85
		Fecha: 2012-08-03

INFORMACIÓN GENERAL

Actores:	<ul style="list-style-type: none"> Administrador
Propósito:	El usuario podrá actualizar la información de los Grupos
Precondiciones:	<ul style="list-style-type: none"> Ingresar al sistema. CU-001 Consultar Grupos CU-015
Postcodiciones:	Se actualiza la información de los grupos
Referencias Cruzadas:	R.2.3.3

Flujo Normal de los Eventos	
Acción del Actor	Respuesta del sistema
1. Ingresar al menú de Grupos y selecciona el grupo que desea Modificar	
	2. Muestra el Grupo con los campos que puede modificar: Nombre del grupo, grado, jornada, director.
3. Modifica los campos necesarios	
4. Selecciona la opción Actualizar grupo.	
	5. Verifica que los datos hayan sido ingresados y valida que estos datos sean correctos.
	6. Almacena los cambios en la Base de Datos
	7. Muestra un mensaje indicando que el Grupo se actualizó satisfactoriamente.

Flujos de eventos Excepcionales
5. No se ingresa un dato, o algún dato es incorrecto. 5.1 El sistema muestra un mensaje de error e indica cual campo falta o es incorrecto
7. No se logro actualizar los datos del grupo por un fallo en la conexión con la base de datos 7.1 El sistema muestra un mensaje indicando la falla, e intenta reanudar la conexión; si no se puede reanudar, la operación se debe iniciar nuevamente

Tabla 12. Caso de Uso Modificar Grupo

3.2.2 DIAGRAMA DE PAQUETES

Por medio del Diagrama de Paquetes se obtuvo una visión más clara del sistema de información orientado a objetos. Permitió organizar y agrupar los elementos (Casos de uso, objetos) en subsistemas o módulos.

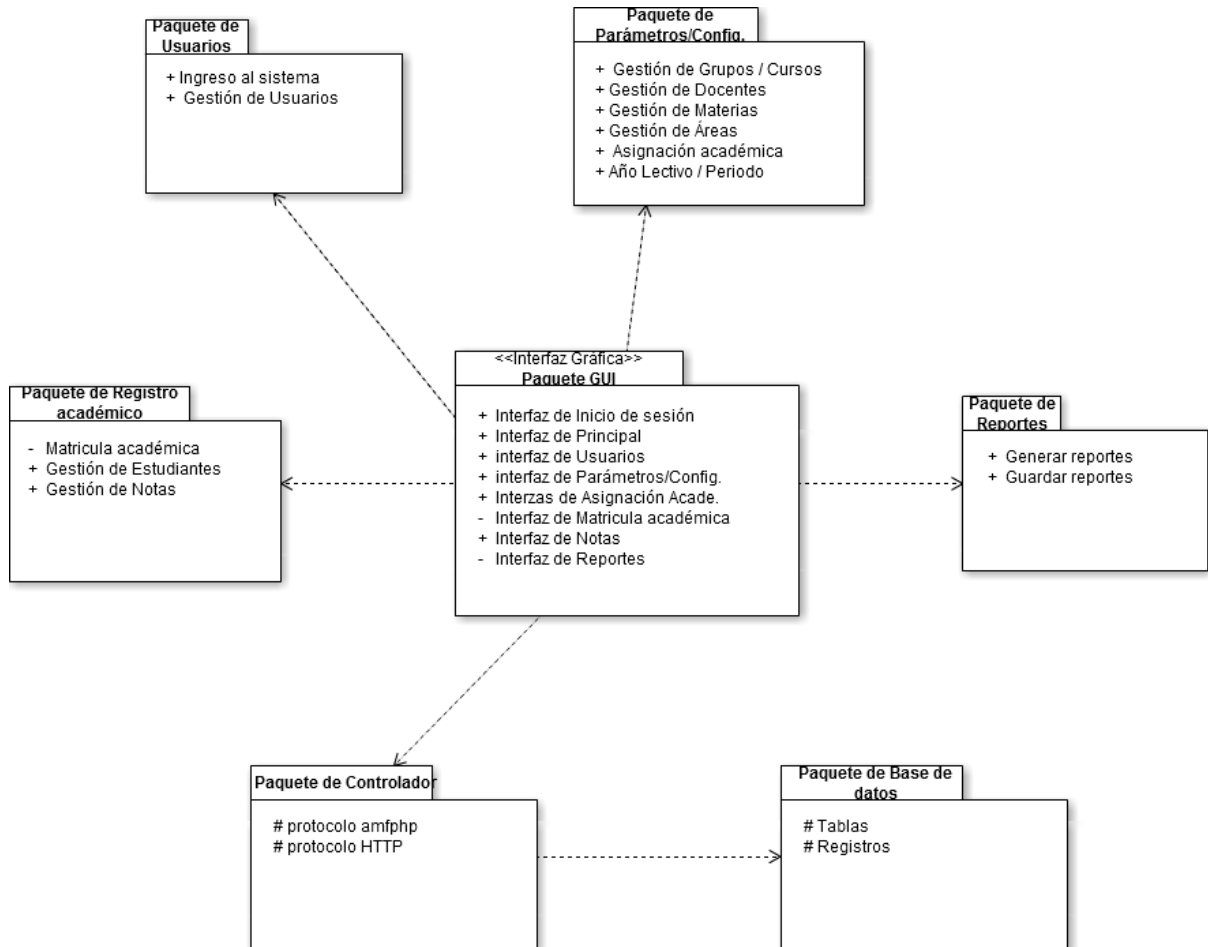


Figura 9. Diagrama de Paquetes

3.3 DISEÑO

Para ver el Diseño completo del sistema (Modelo Entidad Relación, Diagramas de Secuencia y Diagramas de Clases), remitirse al **Anexo C**. Documento de Diseño del Sistema.

3.3.1 Modelo Entidad-Relación

Para el correcto funcionamiento y persistencia de datos, la aplicación para el Registro Académico cuenta con una Base de Datos llamada “simar”.

A continuación se muestra el MER (Modelo Entidad Relación) donde se muestra las tablas que conforman la base de datos y la relación entre ellas.

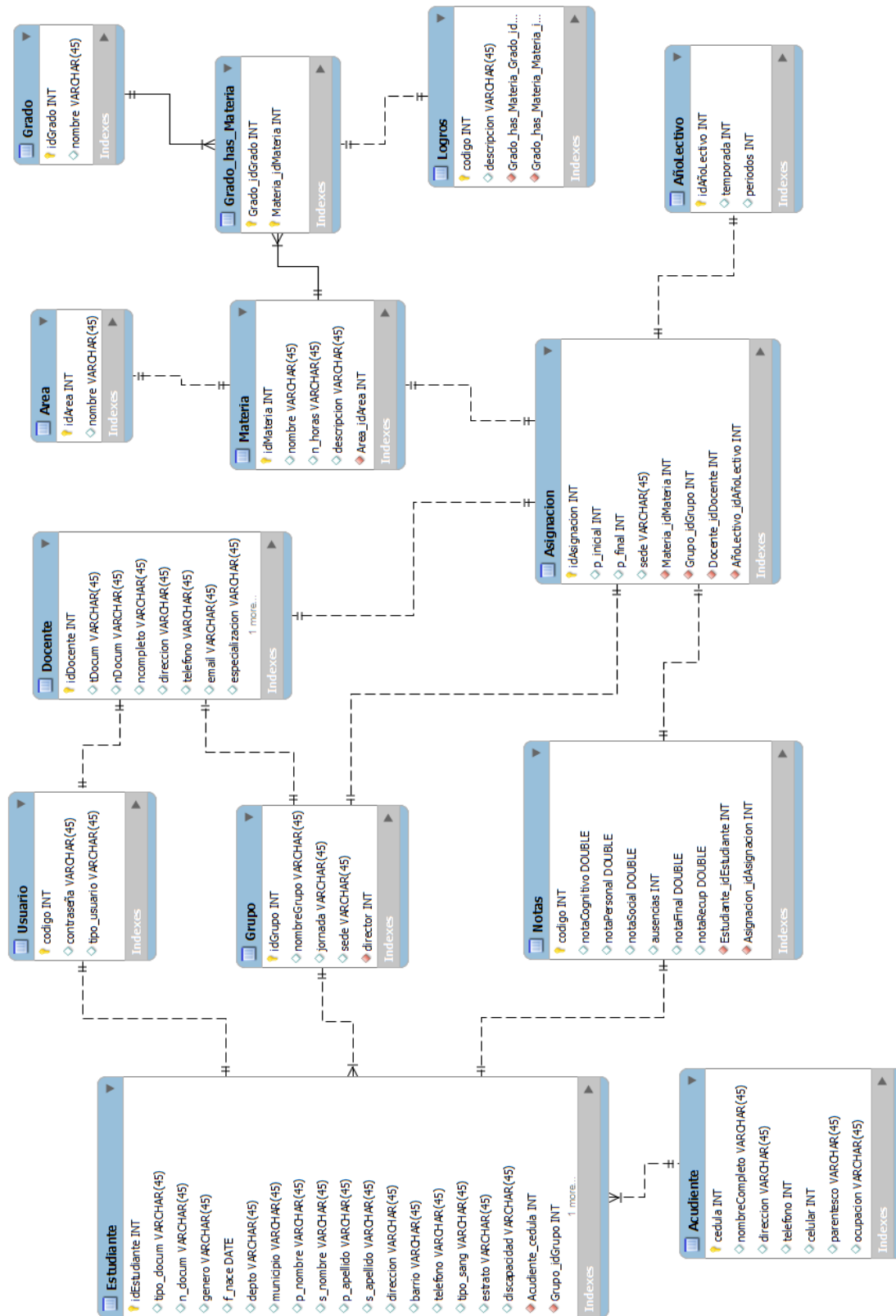


Figura 10. Modelo Entidad – Relación

3.3.2 Diagramas de Secuencia

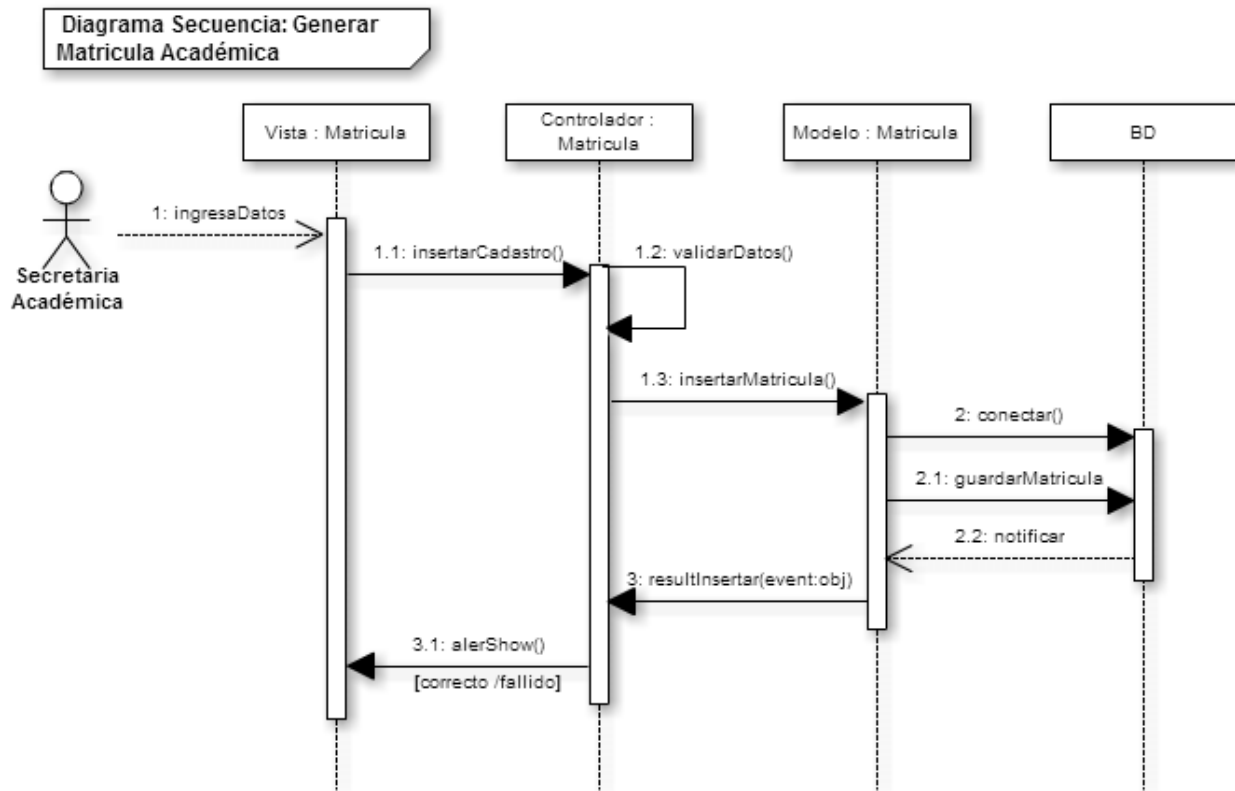


Figura 11. Diagrama de Secuencia: Generar Matrícula Académica

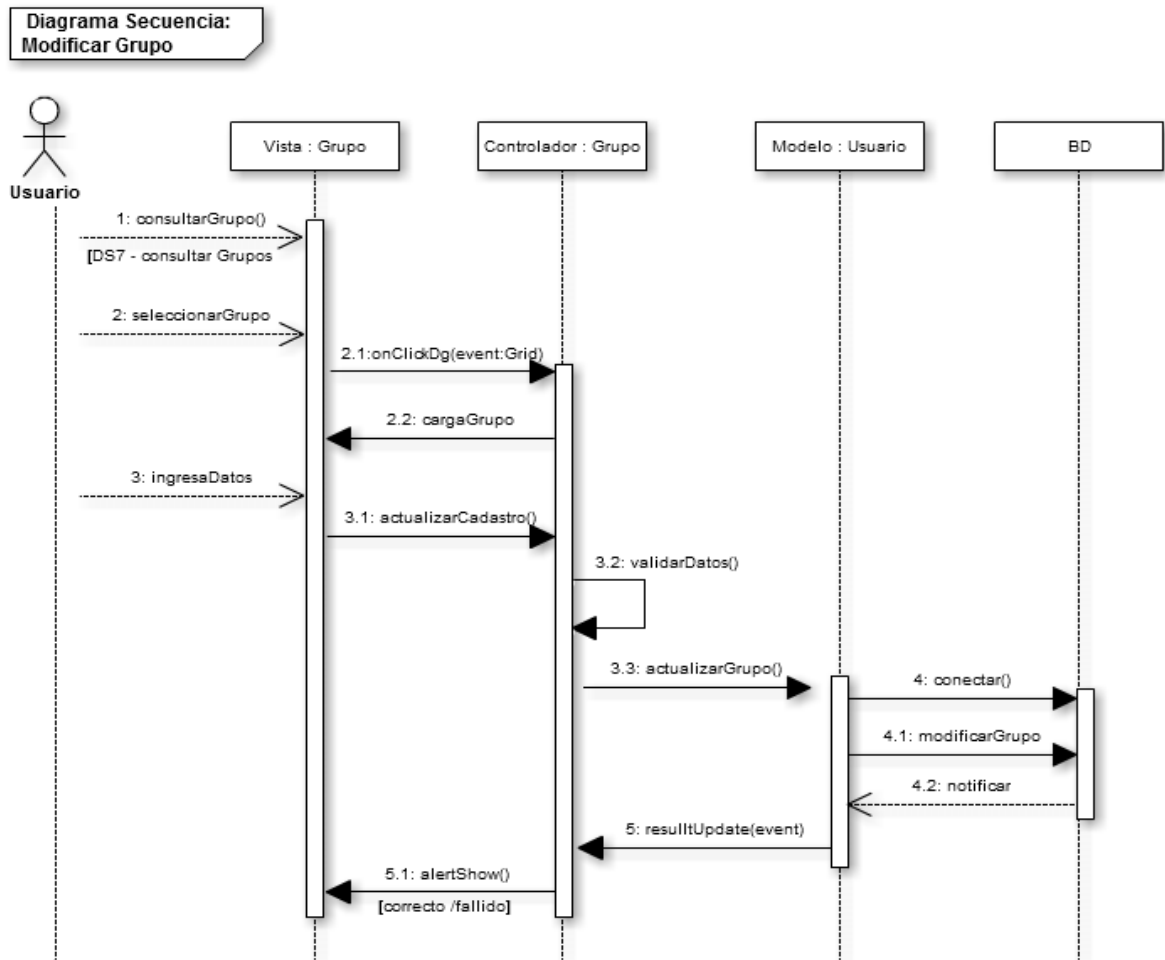


Figura 12. Diagrama de Secuencia: Modificar Grupo

3.3.3 Diagrama de Clases

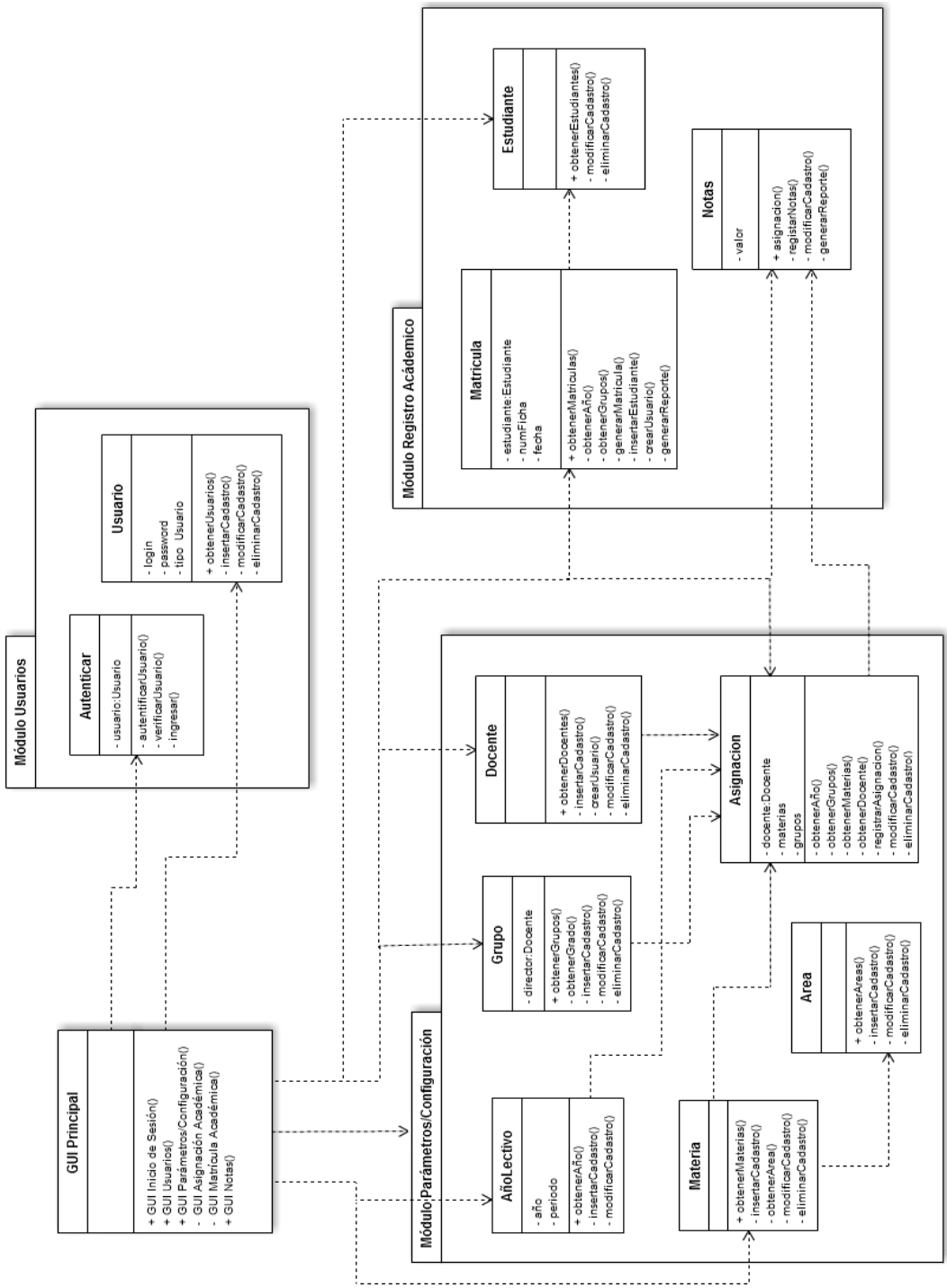


Figura 13. Diagrama de Clases

3.4 IMPLEMENTACIÓN

En esta última fase del desarrollo, se generó el código fuente de la aplicación (interfaces, clases, métodos, relaciones, scripts), se realizó el diagrama de componentes y las pruebas del sistema.

3.4.1 Aplicaciones y Herramientas

- **Framework Adobe Flex [13]**

El sistema de información 'SIMAR' fue implementado mediante el Framework Adobe Flex, que permite crear aplicaciones enriquecidas de Internet (RIA), basadas en el formato de Adobe Flash (.swf). La aplicación puede ejecutarse en cualquier navegador Web que soporte Adobe Flash Player.

Flex no es un lenguaje de programación, pero sí ejecuta el proceso compilación, en el cual el código que utilice el Framework se traduce al lenguaje ActionScript para luego formar un archivo binario de extensión SWF que puede ser reproducido con Flash Player. Esto hace posible ejecutar el mismo archivo en Windows, Mac o Linux sin mayores complicaciones, ya que el reproductor es quien se encarga de la compatibilidad para cada una de las plataformas.

¿Qué no es Flex?

Para comprender de una manera más adecuada qué es Flex, se lista aquellos elementos que no es y que normalmente generan confusión en el ámbito profesional, universitario y en el mercado.

- No es un lenguaje de programación.
- No es un entorno de trabajo (IDE).
- No es un lenguaje de marcado.
- No es una plataforma de servidor.
- No es competencia para Flash

El Kit de desarrollo de Flex está compuesto por un conjunto de clases de lenguaje ActionScript 3.0, una completa librería de componentes visuales como botones, grillas de datos, ventanas, paneles, entre otros, que pueden ser manipulados mediante un lenguaje de marcado, denominado MXML; también incluye el compilador necesario para exportar las aplicaciones desarrolladas con Flex a formato SWF; y un depurador que nos permite probar y detectar errores en el código mientras se está desarrollando la aplicación.

MXML: Es un lenguaje de marcado declarativo basado en XML, describe interfaces de usuario, crea modelos de datos y tiene acceso a los recursos del servidor. Tiene una estructura en base a etiquetas con una sintaxis menos ambigua.

ActionScript 3: Es un lenguaje de programación orientado a objetos de la Plataforma Adobe Flash, muy similar a JavaScript, permite una programación mas interactiva.

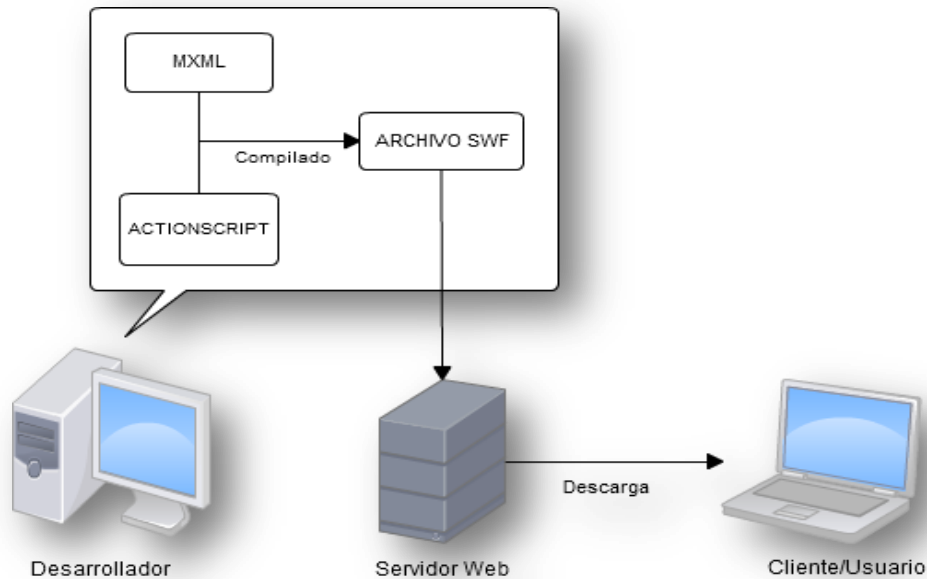


Figura 14. Proceso de compilación y publicación de una aplicación Flex

▪ Servidor XAMPP [14]

Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

Las características de la versión XAMPP win32-1.7.4-VC6

- Apache 2.2.17
- MySQL 5.0.7
- PHP 5.3.5
- phpMyAdmin3.3.9

▪ **Protocolo de Comunicación AMF [15]**

Es muy común que la aplicación Flex se comunique con el servidor Web para enviar o recibir datos. Esta comunicación puede establecerse utilizando formatos estándares como XML o JSON.

AMF (Action Message Format) es un protocolo de intercambio de datos en formato binario que nos brinda muchas ventajas sobre los demás protocolos a la hora de comunicación con el servidor de ActionScript.

Antes de enviar los datos, AMF se encarga de serializar los objetos por enviar. La serialización es un proceso de codificación a un formato capaz de ser transmitido o almacenado; este proceso también puede ser comprimido. Llegados a destino, los datos son deserializados para obtener un clon de objeto existente del otro lado.

La serialización de objetos no sólo se utiliza para transmitir la información, también puede utilizarse para almacenar objetos en una sesión en el servidor, o directamente en una base de datos, manteniendo intactas sus propiedades.

Integración con el Servidor

Desde ActionScript se utiliza el AMF como protocolo de comunicación; se implementa mediante una Llamada a Procedimiento Remoto (Comúnmente denominado RPC). RPC es un protocolo que permite invocar la ejecución de código en una máquina remota (como el servidor Web) y recibir la respuesta de forma totalmente transparente al programador, quien no tiene que ocuparse por la transmisión de los datos, ya que quedan encapsulados dentro del protocolo.

▪ **AMFPHP**

El papel de AMFPHP es proporcionar una manera muy simple de conectar un cliente con un servidor en la nube, empleando lenguaje PHP. Este se encargará de procesar la petición, accediendo a bases de datos (MySQL) e interactuando con sus propios formatos, y luego nos enviará una respuesta con su propio formato, sin necesidad de conversiones.

En el interior de la carpeta AMFPHP se encuentran las siguientes sub-carpetas.

Core: Allí se encuentran los archivos que conforman el núcleo de AMFPHP, necesarios para la ejecución de la librería.

Services: Dentro de esta carpeta debemos incluir nuestros archivos de php para exponerlos como “servicios” en el protocolo AMF que se genera.

Browser: contiene un explorador de servicios que mediante una interfaz web nos permite visualizar los servicios disponibles en nuestro AMFPHP.

Para utilizar AMFPHP en el proyecto Flex se debe ubicar la carpeta amfphp en nuestro servidor Xampp y se alojan las clases PHP en la carpeta Services.

Los servicios son clases PHP que representan a cada entidad disponible en el servidor, generalmente tendremos un clase asociada con cada fuente de datos que se accede desde Flex.

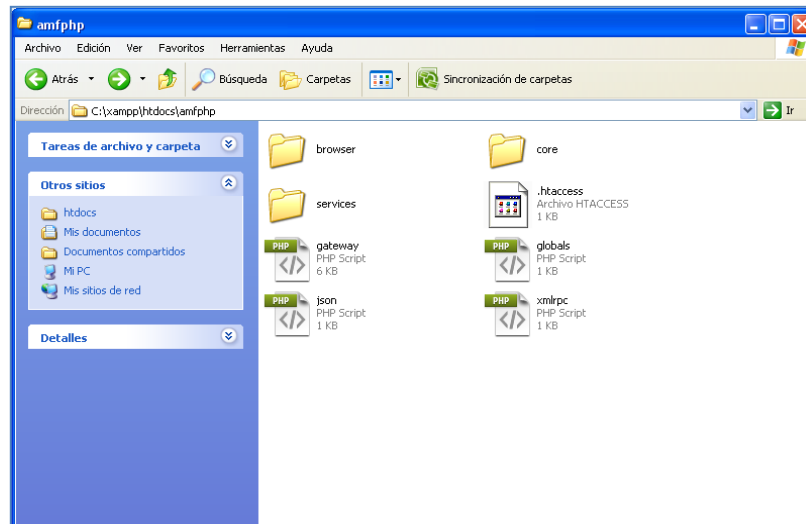


Figura 15. Elementos Protocolo AMFPHP

3.4.2 Estructura de la Implementación

Modelo MVC

Para la implementación de la aplicación se utilizó el paradigma de programación Orientado a Objetos, utilizando el patrón de diseño Modelo Vista Controlador (MVC), que nos ayudo a identificar fácilmente los objetos involucrados, para dar solución a la problemática planteada.

En el momento de programar bajo ActionScript se separa la lógica y los eventos de la parte gráfica y visual.

El modelo MVC se aplicó en las clases de la aplicación de la siguiente manera:

- **Modelo:**

Representa los datos que serán utilizados en la aplicación y la inteligencia de negocio.

En este caso, son los datos que se muestran a través de los componentes **Model o ArraysCollection** y las clases PHP que contienen los métodos referentes a consultar, insertar, modificar y eliminar información de la Base de Datos (queries).

A continuación se presenta como ejemplo el código perteneciente al Modelo de la clase “*grupo.php*”

```
<?php

Class grupo
{

function obtenerGrupo()
{
    $Query = " SELECT * FROM grupos ORDER BY codigo ASC";
    $Result = mysql_query( $Query );

    while ($row = mysql_fetch_object($Result))
    {
        $return[] = $row;
    }
    return ( $return );
}

function insertarGrupo($id_c,$grupo,$grado,$jornada,$director,$sede)
{
    $sql= "INSERT INTO grupos (codigo, nombreGrupo, grado, jornada,
        director, sede)
        VALUES('$id_c','$grupo','$grado','$jornada','$director','$sede')";

    $Result = mysql_query($sql);

    if ($Result >= 1)
    {
        return "insertado";
    } else
    {
        return "falla";
    }
}

function eliminarGrupo ($id)
{
    $sqlDelete= "DELETE FROM grupos WHERE codigo= $id";
    $Result = mysql_query( $sqlDelete );
    return "Eliminar";
}
```

```

function actualizarGrupo($id_c,$grupo,$grado,$jornada,$director, $sede)
{
    $sql = "UPDATE grupos SET nombreGrupo='$grupo', grado='$grado',
            jornada='$jornada', director='$director', sede='$sede'
            WHERE codigo= $id_c";

    $Result = mysql_query( $sql );
    return "Actualizar";
}
?>

```

▪ Vista:

Es la Interfaz Gráfica de la aplicación, la capa visible por el usuario; se utilizaron componentes visuales que permiten mostrar los datos en pantalla, como **DataGrid**, **Tree y lists**. También se aplicaron componentes que le facilitan la comprensión, el aprendizaje y el uso de la aplicación, como son: **image, banner, BorderContainer, panel, label, forms, tabs, RadioButton, textInput, Button, Link**, entre otros.

A continuación se presenta como ejemplo el código perteneciente a la Vista de la clase "*Materia.mxml*"

```

<s:BorderContainer x="6" y="38" width="450" height="214"
    borderColor="#B8B3B3">

    <s:Form x="2" y="4" width="445" height="207">

        <s:FormItem width="366" label="ID" fontWeight="bold"
            required="true">
            <s:TextInput width="113" id="txtId" restrict="0-9"/>
        </s:FormItem>

        <s:FormItem width="445" label="Nombre" fontWeight="bold"
            required="true">
            <s:TextInput width="279" id="txtNombre" restrict="A-Z"/>
        </s:FormItem>

        <s:FormItem width="365" label="Intensidad Horaria"
            fontWeight="bold">
            <s:TextInput width="112" id="txtHoras" restrict="0-9"/>
        </s:FormItem>

        <s:FormItem width="330" label="Grado">
            <s:ComboBox id="txt_grado" width="190"/>
        </s:FormItem>

    </s:Form>

</s:BorderContainer>

```

- **Controlador:**

Se encarga de los eventos de la aplicación y la comunicación entre la Vista y el Modelo, en este caso se hizo uso del servicio **RemoteObject**, el cual se comunica mediante el protocolo AMF, para la conexión con el modelo.

El servicio **RemoteObject** permite la comunicación entre las clases actionScript y las clases PHP, para recibir los resultados de las consultas de la base de datos.

Se muestra la configuración del servicio *RemoteObject* de la clase “*Materia.mxml*”

```
<fx:Declarations>

<s:RemoteObject id="Materia" destination="amfphp" source="materia"
    showBusyCursor="true">

    <s:method name="insertarMaterias"
        result="resultInsertar(event) "
        fault="Alert.show(event.fault.faultString) ">

        <s:arguments>
            <id>{txtId.text}</id>
            <nombre>{txtNombre.text}</nombre>
            <horas>{txtHoras.text}</horas>
            <descripcion>{txtDescri.text}</descripcion>
            <areas>{Obtarea.selectedItem.nombrel}</areas>
        </s:arguments>

    </s:method>

    <s:method name="actualizarMaterias"
        result="resultUpdate(event) "
        fault="Alert.show(event.fault.faultString) ">

        <s:arguments>
            <id>{txtId.text}</id>
            <nombre>{txtNombre.text}</nombre>
            <horas>{txtHoras.text}</horas>
            <descripcion>{txtDescri.text}</descripcion>
            <areas>{Obtarea.selectedItem}</areas>
        </s:arguments>

    </s:method>

    <s:method name="eliminarMaterias"
        result="resultUpdate(event) "
        fault="Alert.show(event.fault.faultString) ">

        <s:arguments> <id>{txtId.text}</id> </s:arguments>

    </s:method>

</s:RemoteObject>

</fx:Declarations>
```

Se muestra el manejo de los eventos de la clase “Grupo.mxml”

```
<fx:Script>

<![CDATA[

import mx.collections.ArrayCollection;
import mx.controls.Alert;
import spark.events.GridEvent;

private function insertarCadaastro():void
{
    if (txtNombre.text == "" && txtId.text == "")
    {
        Alert.show("Es necesario digitar un Código o nombre", "ALERTA")
    }
    else
    {
        Materia.insertarMaterias.send();
    }
}

private function resultInsertar(event:Object):void
{
    var resultado:String=String(event.result);
    if (resultado == 'insertado')
    {
        Alert.show('Funcionario Insertado Exitosamente', 'Mensaje')
        Materia.obtenerMaterias.send();
        limpiarCampos();
    }
    else
    {
        Alert.show('Error al insertar', 'ERROR...');
    }
}

[Bindable]

private var selectDg:Object;
private function onClickDg(event:GridEvent):void
{
    selectDg=event.itemRenderer.data;
    txtId.text=selectDg.id_mater;
    txtNombre.text=selectDg.nombre;
    txtHoras.text=selectDg.n_horas;
    Obtarea.selectedItem=selectDg.nombrel;

    if (txtId.text == selectDg.codigo)
    {
        txtId.enabled=false;
    }
}

]]>

</fx:Script>
```


3.4.3 Diagrama de Componentes

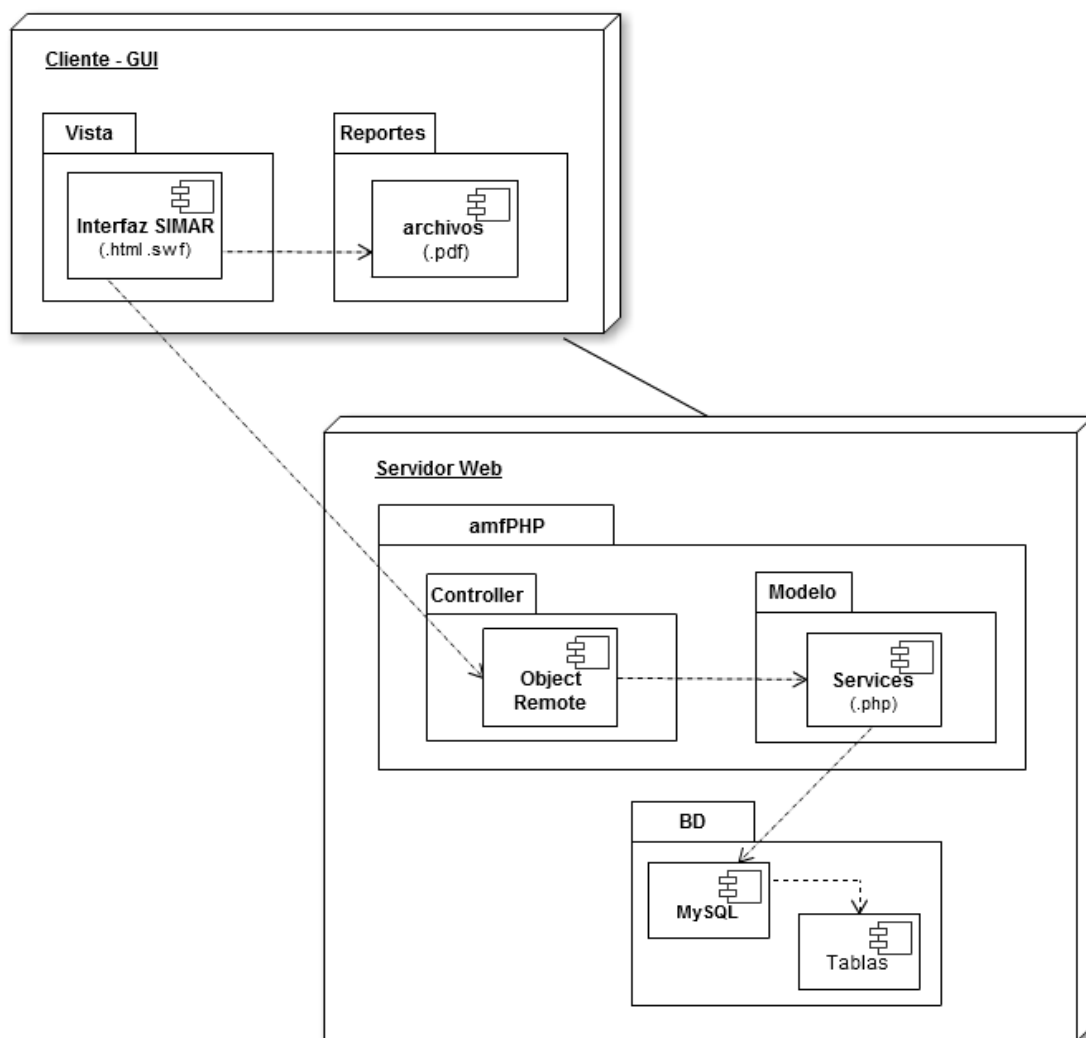


Figura 16. Diagrama de Componentes

4. PRUEBAS Y RESULTADOS

Como se describió en el alcance del proyecto, se realizó un Plan de Pruebas donde se probó la comunicación y el funcionamiento de cada uno de los módulos de la Aplicación Web para el Registro Académico de la Institución Educativa María Antonia Ruiz (**SIMAR**). Este plan de pruebas también nos permitió identificar algunos errores que se presentaron en la funcionalidad de la aplicación.

Se realizaron casos de prueba donde se detalla a que módulo y a cuales funcionalidades se le aplicó la prueba, también se verifica que la aplicación cumpla con los requerimientos funcionales y no funcionales y con los casos de uso del sistema.

Pruebas de Integridad de Datos y Base de Datos:

- Verificar el acceso a la Base de Datos de la aplicación.
- Verificar conexiones al servidor.
- Verificar el correcto retorno de los datos actualizados en la base de datos.

Pruebas del Sistema:

- Verificar Ingreso al Sistema.
- Verificar Módulo Usuarios.
- Verificar Módulo Parámetros/Configuración.
- Verificar Asignación Académica.
- Verificar Generar Matrícula Académica.
- Verificar Registrar Notas.

Pruebas de Unidad:


Se realizó las Pruebas de Unidad por medio del Framework de Adobe Flex, que nos ofrece una herramienta para realizar pruebas de unidad de clases o proyectos llamada **FlexUnit Test**.

FlexUnitTest genera un archivo donde muestra el resultado de la prueba detallando cualquier falla que se presente en la estructura y codificación de las clases.

Para ver el Plan de Pruebas del sistema completo, remitirse al **Anexo D**. Documento de Pruebas del Sistema.

4.1 Casos de Prueba

A continuación se presenta los casos de prueba Ingresar al Sistema y Generar Matrícula Académica:

	SIMAR	Id: CP-001
CASO DE PRUEBA: INGRESAR AL SISTEMA		Página #: 5/10 Fecha: 2012-10-27


Documentos de Referencia

Id	Descripción	Formato
ERS	Documento de Especificación de Requerimientos del Sistema	.pdf

INFORMACIÓN GENERAL

Objetivo	Verificar el ingreso al sistema de un Usuario
Fecha ejecución	Octubre 27 de 2012
Tiempo Estimado	50 segundos
Tiempo Real	30 segundos
Hardware Requerido	Estación de Trabajo con 2 GB de RAM y Procesador AMD Athlon X2
Software Requerido	Sistema Operativo Windows 7 y Windows XP
Proceso	Ejecutar la aplicación Ingresar los campos de inicio de sesión (Código y contraseña) Seleccionar la opción Iniciar Sesión Ingresar a la Interfaz principal de la aplicación
Valores de prueba	Usuario de tipo Administrador Código: 1116251014 Contraseña: 12345
Resultado Esperado	Una vez de que el Usuario ingrese a la aplicación podrá acceder a las funcionalidades que requiera.
Prueba Exitosa	Si <u> X </u> No <u> </u>
Observaciones	El usuario ya esta previamente registrado en el sistema

Tabla 13. Caso de Prueba: Ingresar al Sistema

	SIMAR	Id: CP-006
CASO DE PRUEBA: GENERAR MATRÍCULA ACADÉMICA		Página #: 9/10
		Fecha: 2012-11-13

Documentos de Referencia

Id	Descripción	Formato
ERS	Documento de Especificación de Requerimientos del Sistema	.pdf

Información General

Objetivo	Verificar el registro de la Matrícula académica de un estudiante.	
Fecha ejecución	Noviembre 13 de 2012	
Tiempo Estimado	5 minutos	
Tiempo Real	3.5 minutos	
Hardware Requerido	Estación de Trabajo con 2 GB de RAM y Procesador AMD Athlon X2	
Software Requerido	Sistema Operativo Windows 7 y Windows XP	
Proceso	Ingresar al sistema CP-001 Ingresar al Módulo Registro Académico Acceder a la opción Matrícula académica	
Valores de prueba	Estudiante: Código: 201201 Tipo Identificación: T.I Número Identificación: 99090306240 Sexo: Masculino Fecha Nacimiento: 1999-09-03 Lugar Nacimiento: Tuluá Primer Nombre: Marcelo Segundo Nombre: Sebastián Primer Apellido: Parra Segundo Apellido: Reyes Municipio Residencia: Tuluá Dirección: cra 30 #7-50 Barrio: Morales Teléfono: 2262461 Tipo de Sangre: O+ EPS: Nueva EPS Nivel Sisben: 1	Acudiente: Número Identificación: 1116251014 Nombre Completo: Ancizar Parra R. Dirección: cra 30 #7-50 Teléfono: 2262461 Celular: 3173323890 Parentesco: Padre Ocupación: Vendedor
		Ficha de Matrícula: Número Ficha: 12 Fecha: 2012-11-13 Grupo: Septimo-1
Resultado Esperado	La Ficha de matricula correspondiente y un Estudiante matriculado en un Grupo.	
Prueba Exitosa	Si <u> X </u> No <u> </u>	
Observaciones	<ul style="list-style-type: none"> El usuario que realiza la Matrícula es la Secretaria académica. La funcionalidad Consultar Ficha de matrícula es un poco lenta porque tiene muchos campos y consulta todos los estudiantes matriculados, a la hora de mostrar en pantalla se demora. 	

Tabla 14. Caso de Prueba: Generar Matrícula Académica

4.2 Resultados

- ✓ Se realizó un estudio detallado del sistema actual para evaluar y describir las características del sistema propuesto, durante este análisis se logró identificar las necesidades y requerimientos. Se obtuvo información importante para el desarrollo del sistema: Formato de matrícula, formato de registro de notas, ficha de docente, pacto de convivencia y revista de la institución.
- ✓ Se sistematizó la configuración, los procesos y formatos del registro académico que emplea actualmente la Institución, siguiendo las normas y lineamientos estipulados en el Pacto de Convivencia.
- ✓ Se implementó la aplicación web 'SIMAR' (Sistema de Información María Antonia Ruiz) que permite la ejecución de los procesos del registro académico de la Institución, no tiene un código cerrado lo que implica que se pueden seguir realizando modificaciones, mejoras y nuevas implementaciones. También permite adaptarse a otras instituciones educativas que requieran sistematizar su proceso de Registro Académico.
- ✓ SIMAR está compuesta por los módulos de software:
 - **Módulo Usuarios:** permite la gestión de Usuarios
 - **Módulo Parámetros /Configuración:** permite la gestión del Año lectivo, de los grupos, de los docentes, de las materias, los logros académicos y de la asignación académica.
 - **Módulo Registro Académico:** permite generar la matrícula académica de los estudiantes, registrar las notas de las materias.
 - **Módulo Reportes:** permite generar los reportes de las fichas de matrícula y listado de notas.
- ✓ Se diseñó una Base de Datos robusta que permite almacenar la información generada por los usuarios de la aplicación.
- ✓ Se ejecutó un Plan de Pruebas del Sistema donde se verificó la funcionalidad de los módulos de la aplicación, que permitió corregir errores que se presentaron en la implementación.

5. CONCLUSIONES Y TRABAJOS FUTUROS

5.1 Trabajos Futuros

Con el fin de mejorar el proyecto y ampliar los alcances del mismo, se hace interesante la realización de las siguientes propuestas:

- Permitir gestionar la información de los empleados de la institución (Rectora, secretarias, coordinadoras, tesorera, vigilantes).
- Ampliar el Módulo de Reportes, incluir gráficas y reportes de todos los módulos para que permita conocer las estadísticas finales del año lectivo y de cada periodo sobre el registro académico de la institución, que ayuden en la toma de decisiones y a corregir errores que se presenten.
- Desarrollar los módulos necesarios para adaptar el registro académicos de las demás sedes en la Aplicación.
- Se deberá adquirir un motor de base de datos licenciado para mejorar la capacidad de almacenamiento de registro, para incorporar las demás sedes de la Institución.
- Implantar la aplicación en un servidor y dominio propio de la Institución.
- Adaptar la aplicación para que pueda ser comercializada a otras Instituciones educativas de la ciudad.

5.2 Conclusiones

Con el desarrollo de este proyecto se puede concluir:

La metodología *ICONIX* permitió un desarrollo práctico e iterativo, donde se interactuó con el cliente durante todo el proceso, lo que brindo ir probando, corrigiendo y mejorando la aplicación durante el desarrollo. Sólo plantea realizar los artefactos UML necesarios para las etapas de análisis y diseño, para que la mayor parte del tiempo se le dedique a la etapa de implementación.

Adobe Flex es una buena alternativa de framework de desarrollo, ofrece una completa librería de herramientas que permite desarrollar aplicaciones de forma ágil y unificada, su principal ventaja es que las aplicaciones desarrolladas se pueden ejecutar en cualquier sistema operativo y en el cualquier navegador web que ejecute el estándar '*flash*' (.swf). Sin embargo fue muy costoso el aprendizaje de *Adobe Flex* porque era una nueva tecnología y no se tenía la documentación necesaria para manejarlo, además el *IDE* de desarrollo de *Adobe Flex* es licenciado y no se tenía una versión estable.

Debido a la investigación y al levantamiento de información se adquirieron conocimientos sobre el proceso de Registro de Académico de las instituciones educativas oficiales de la ciudad de Tuluá, las normas establecidas por el Ministerio de Educación Nacional y la Secretaría de Educación Municipal.

Al aplicar el patrón de diseño *MVC (Modelo Vista Controlador)* permitió estructurar y organizar el código fuente de la aplicación, para que sea estándar y cualquier desarrollador pueda mejorar o incorporar nuevas funcionalidades a la aplicación.

Por otra parte, las funcionalidades *core* de la aplicación (*Registro de Notas, Registro y Consulta de las Matrículas Académica*) se tardan un poco en ejecutarse por la cantidad de datos que procesa y realiza muchos accesos a la base de datos. Lo que se recomienda que la aplicación se ejecute en un equipo de cómputo con buena capacidad de memoria, almacenamiento y procesamiento. Y además optimizar las *consultas SQL*.

Finalmente se puede concluir que con la puesta en funcionamiento de la aplicación web *SIMAR* se mejorará la gestión del proceso de Registro Académico de la Institución Educativa María Antonia Ruiz, en cuanto a tiempo, integración y organización de la información.

BIBLIOGRAFÍA

[1] UNIVERSIDAD DEL VALLE. Escuela de Ingeniería de Sistemas y computación. 2010, Aplicaciones Web y Redes Inalámbricas; Ingeniería web. [Online] (Disponible en: <http://eisc.univalle.edu.co/cursos/web/material/750087M/1/Clase2www-2010.pdf>)

[2] S MURUGESAN, Y. DESHPANDE, S. HANSEN, A. GINIGE.; "Web Engineering: A New Discipline for Development of Web-Based Systems". Lecture Notes in Computer Science 2016 Springer 2001. Pág. 3.

[3] R. PRESSMAN, Software Engineering; A Practitioner's Approach. 5th Edition, Mc Graw-Hill 2000. Chapter 29, "Web Engineering," Pág. 813 – 843.

[4] WIKIPEDIA, The free encyclopedia. Aplicación web. [Online]. (Disponible en: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web . Consultado el: 20 de abril de 2011).

[5] CIBERTIA. Servidor web. [Online]. (Disponible en: http://www.cibertia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php. Consultado el: 29 de abril de 2011).

[6] WIKIPEDIA, The free encyclopedia. Sistema de Información web. [Online]. (Disponible en: http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n Consultado el: 29 de abril de 2011).

[7] PHP GROUP. Lenguaje Php. [Online]. (Disponible en: <http://php.net/manual/es/intro-what-is.php>. Consultado el: 20 de abril de 2011).

[8] MySQL. [Online]. (Disponible en: <http://www.techterms.com/definition/mysql> . Consultado el: 20 de abril de 2011).

[9] WIKIPEDIA, Framework. [Online]. (Disponible en <http://es.wikipedia.org/wiki/Framework> . Consultado el: 29 de abril de 2011).

[10] SHUMELLER, JOSEPH. 2000. *Aprendiendo uml en 24 horas*. s.l. : Prentice Hall.

[11] MARSTON, TONY. 2004. The Model-View-Controller (MVC). [Online]. (Disponible en: <http://www.tonymarston.net/php-mysql/model-view-controller.html>. Consultado el: 20 de abril de 2011).

[12] CARLA REBECA PATRICIA DE SAN MARTIN OLIVA, Metodología ICONIX [Online] (Disponible en: <http://www.portalhuarpe.com.ar/Seminario09/archivos/MetodologiaICONIX.pdf>).

[13] MARIANO CARRIZO, MAXIMILIANO FIRTMAN; “Flex 4.5: Plataforma Flash para Profesionales; Alfaomega 2011. Capitulo 1, “Que es Flex,” Pág. 13 – 60.

[14] XAMPP. Servidor Xampp. [Online].
(Disponible en: <http://www.apachefriends.org/es/xampp.html>).

[15] MARIANO CARRIZO, MAXIMILIANO FIRTMAN; “Flex 4.5: Plataforma Flash para Profesionales; Alfaomega 2011. Capitulo 2, “Comunicación con el Servidor,” Pág. 61 – 90.