



UNIVERSIDAD DEL AZUAY

Facultad Ciencias de la Administración

ESCUELA DE INGENIERIA EN SISTEMAS

Ventajas y desventajas de la Tecnología RIA (Rich Internet Applications).

Desarrollo de tesis previo a la obtención del título de Ingeniero en Sistemas

Autores:

Carlos Germán Ávila Nieto.

Eliana Fernanda Sangurima Alvarado.

Director:

Ing. Pablo Pintado

Cuenca-Ecuador

2011

DEDICATORIA

Quiero dedicar primeramente a Dios por darme la oportunidad, la capacidad, la fortaleza, salud y sabiduría de realizar la presente tesis; también a un gran hombre Gustavo Sangurima que es mi padre, el cual me ayudo moralmente y económicamente en todas las circunstancias de mi vida, a mi madre Magaly Alvarado y mi hermana Alexandra.

En especial a mi hermano Fabián por sus consejos y sabiduría que supo brindármelo en un momento dado a pesar de estar ausente y a mi tío Pablo Fernando que supo apoyarme desde los inicios de mis estudios.

Eliana.

DEDICATORIA

Dedico este proyecto a mi padre, apoyo, animo, coraje, pasión y perfecta armonía entre disciplina y amor; a mi madre, fuerza, perseverancia, dulzura, comprensión y valentía hacia la vida; mis hermanos, amistad, empuje, aprendizaje, unión, arte, vida; y a Jehová, fuente de amor y poder

Carlos .

AGRADECIMIENTO

Agradecemos en primer lugar a nuestros padres por su comprensión y cariño. De igual manera agradecemos atentamente a nuestro director de tesis Ing. Pablo Pintado por invertir su tiempo en la consecución de este trabajo. También queremos expresar especial agradecimiento a todos los profesores que nos han inculcado valores y enseñanza, no solamente para ser futuros profesionales sino para crecer como seres humanos

INDICE DE CONTENIDO

DEDICATORIA	II
AGRADECIMIENTO	IV
INDICE DE CONTENIDO	V
ÍNDICE DE ILUSTRACIONES Y CUADROS	IX
RESUMEN	XI
ABSTRACT	XII

INTRODUCCION	1
--------------------	---

1.RIA (Rich Internet Applications).....	2
1.1. Introducción.	2
1.2Qué es Aplicación Web?.....	3
1.2.1.Estructura	3
1.2.2. Utilidades	3
1.3 Evolución	4
1.3.1 La Web 1.0 ó Web tradicional.	4
1.3.1.1Características:.....	4
1.3.1.2. Limitaciones:.....	5
1.3.2 La Web 2.0.....	5
1.4. Aplicaciones Ricas de Internet (RIA: Rich Internet Applications.....	7
1.4.1. Características	7
1.4.2. Estructura	9
1.4.3. Ventajas de la Estructura.	11
1.4.4. Ventajas de la aplicación RIA.....	12
1.4.5. Utilidades	12
1.5 Herramientas de diseño RIA	13
1.6 Conclusiones	13
1.7. Recomendaciones	14
2. RIA dentro de Silverlight.....	15
2.1. Introducción	15
2.2 Introducción a Silverlight.	16
2.3 Evolución de Silverlight.	16
2.3.1. Versión 1.0.....	17
2.3.2.Versión 2.0	18
2.3.3.Versión 3.0	18
2.3.4.Versión 4.0.....	19
2.4 Principales funcionalidades de la herramienta.....	19
2.4.1. Experiencia de usuario multiplataforma	20
2.4.2. Modelado de programación flexible con herramientas de colaboración.	20
2.4.3. Multimedia de alta calidad a bajo costo.....	21
2.4.4. Conexión de datos, servidores y servicios.	21
2.5 Ejemplos.	21

2.6 Pruebas de concepto.....	23
2.7 Comparación de Silverlight con otras herramientas RIA.	24
2.7.1.JavaScript/ Ajax	24
2.7.2.Google's GWT Framework	25
2.7.3.Adobe Air.....	26
2.7.4. Adobe Designer	26
2.7.5. Appcelerator.....	27
2.7.6. OpenLaszlo	27
2.7.7. MagicAjax.NET.....	28
2.7.8. Curl 5.0, REBOL 2.6, y Seaside for Smalltalk	28
2.7.9. JavaFX	28
2.7.10. Java	28
2.7.11. NexaWeb.....	29
2.7.12. Macromedia Flex	30
2.7.13. NQ Suite.....	31
2.7.14.Controles ActiveX	31
2.8. Conclusiones.....	32
2.9. Recomendaciones	32
3. Análisis y Diseño del Sistema.....	33
3.1Introducción	33
3.2. Recolección y levantamiento de la información.....	34
3.2.1. Entrevistas y reuniones con el personal administrativo del centro educativo CEBCI...34	
3.2.2. Definición de requisitos del sistema.	36
3.2.2.1. Introducción	36
3.2.2.2. Propósito	36
3.2.2.3. Ámbito del sistema	37
3.2.2.4 Definiciones	37
3.2.2.5. Responsables del documento	37
3.2.2.6. Historia de las Versiones del Documento.....	38
3.2.3. Descripción General.....	38
3.2.3.1. Funciones del Sistema.....	38
3.2.3.2. Suposiciones y Dependencias	38
3.2.4. Requisitos Específicos	39
3.2.4.1. Requisitos Funcionales	39
3.2.4.2. Descripción de actores	39
3.2.4.3. Descripción de casos de uso	46
3.2.4.3.1Administrar Inscripción.	46
3.2.4.3.1.1Ingresar.	47
3.2.4.3.1.2Modificar.....	48
3.2.4.3.1.3.Eliminar.....	49
3.2.4.3.1.4 Consultar.....	50
3.2.4.3.2 Administrar Pre Matrícula.	51
3.2.4.3.2.1. Ingresar.	52
3.2.4.3.2.2 Modificar.....	53

3.2.4.3.2.3 Eliminar.....	54
3.2.4.3.2.4 Consultar.....	55
3.2.4.3.3. Administrar Matrícula.....	56
3.2.4.3.3.1 Ingresar.....	57
3.2.4.3.3.2 Modificar.....	59
3.2.4.3.3.3 Eliminar.....	60
3.2.4.3.3.4 Consultar.....	61
3.2.4.3.4. Administrar Pago de Pensiones.....	62
3.2.4.3.4.1 Ingresar.....	63
3.2.4.3.4.2 Modificar Servicios.....	64
3.2.4.3.5. Administrar Estudiantes.....	65
3.2.4.3.5.1 Modificar.....	66
3.2.4.3.5.2 Consultar.....	67
3.2.4.3.6. Administrar Año Lectivo.....	68
3.2.4.3.6.1 Ingresar.....	69
3.2.4.3.6.2 Modificar.....	70
3.2.4.3.6.3 Eliminar.....	71
3.2.4.3.6.4 Consultar.....	72
3.2.4.3.7. Administrar Rubro.....	73
3.2.4.3.7.1 Ingresar.....	74
3.2.4.3.7.2 Modificar.....	75
3.2.4.3.7.3 Eliminar.....	76
3.2.4.3.7.4 Consultar.....	77
3.2.4.3.8. Administrar Categoría Rubro.....	78
3.2.4.3.8.1 Ingresar.....	79
3.2.4.3.8.2 Modificar.....	80
3.2.4.3.8.3 Eliminar.....	81
3.2.4.3.8.4 Consultar.....	82
3.2.4.3.9. Administrar Nivel Alumno.....	83
3.2.4.3.9.1 Ingresar.....	84
3.2.4.3.9.2 Modificar.....	85
3.2.4.3.9.3 Eliminar.....	86
3.2.4.3.9.4 Consultar.....	87
3.2.4.3.10. Administrar Documentos.....	88
3.2.4.3.10.1 Ingresar.....	89
3.2.4.3.10.2 Modificar.....	90
3.2.4.3.10.3 Eliminar.....	91
3.2.4.3.10.4 Consultar.....	92
3.2.5. Interfaces de usuario.....	93
3.3 Modelado de Análisis.....	94
3.3.1. Modelo de Clases.....	94
3.3.2. Modelo Objeto-Relación.....	95
3.3.3. Modelo de Comportamiento.....	96
3.3.3.1. Modelo de Actividad.....	96

3.3.3.2. Modelo de Secuencia.....	97
3.4. Modelado de Diseño.....	99
3.4.1 Diseño de Objetos.....	99
3.4.1.1. Diccionario de Datos.....	99
3.4.1.2 Seudocódigo.....	103
3.5. Diseño de Subsistemas.....	108
3.6. Interfaz de Usuario.....	109
3.7. Diseño de Gestión de Datos.....	112
3.7.1. PHP.....	112
3.7.2. Silverlight.....	112
3.8. Gestión de Configuración.....	113
3.8.1.Control de versiones PHP.....	113
3.8.2.Control de versiones SILVERLIGHT.....	115
3.9.Conclusiones.....	116
3.10.Recomendaciones.....	116
4.Comparación de tecnologías.....	117
4.1Introducción.....	117
4.2. Pruebas de aplicación.....	117
4.2.1 PHP.....	117
4.2.2.Silverlight.....	118
4.3 Cuadro de comparación entre PHP y Silverlight.....	119
4.4. Conclusiones.....	122
4.5.Recomendaciones.....	122
CONCLUSIONES.....	123
GLOSARIO.....	124
BIBLIOGRAFIA.....	128

ÍNDICE DE ILUSTRACIONES Y CUADROS

Figura 3.1: Caso de uso General.	40
Figura 3.2: Caso de uso Administrar Inscripción.	41
Figura 3.3: Caso de uso Administrar Pre Matrícula.	41
Figura 3.4: Caso de uso Administrar Matrícula.	42
Figura 3.5: Caso de uso Administrar Pago de Pensiones.....	42
Figura 3.6: Caso de uso Administrar Estudiantes.	43
Figura 3.7: Caso de uso Administrar Año Lectivo.	43
Figura 3.8: Caso de uso Administrar Rubro.	44
Figura 3.9: Caso de uso Administrar Categoría Rubro.....	44
Figura 3.10: Caso de uso Administrar Nivel Alumno.	45
Figura 3.11: Caso de uso Administrar Documentos.	45
Gráfico 1.1: Estructura de tres niveles [2].	3
Gráfico 1.2: Estructura de cuatro niveles [2].	9
Gráfico 1.3: Esquema de interacción entre el cliente y servidor [6].....	10
Gráfico 1.4: Esquema de rendimiento del servidor	11
Gráfico 2.1: Representación de un collage en la página HardRock [21].....	22
Gráfico 2.2: Representación de búsqueda de información [23].....	23
Gráfico 2.3: Modelo JavaScript/ Ajax. [15].....	25
Gráfico 2.4: Modelo Adobe Designer. [1]	26
Gráfico 2.5: Modelo OpenLaszlo. [8].....	27
Gráfico 2.6: Modelo NexaWeb. [8]	29
Gráfico 2.7: Modelo Macromedia Flex. [9].....	30
Gráfico 2.8: Modelo NQ Suite. [10]	31
Gráfica 3.1 Página Principal PHP.....	109
Gráfica 3.2 Página Menú Año lectivo PHP	109
Gráfica 3.3 Página Menú Mantenimiento PHP.....	110
Gráfica 3.4 Página Principal Silverlight	110
Gráfica 3.5 Página mensajes de advertencias Silverlight	111
Gráfica 3.6 Página Menú Año Lectivo Silverlight	111
Tabla 2.1: Conexión a la base de datos MySQL.....	23
Tabla 2.2: Vinculación de datos con la aplicación Silverlight.....	24
Tabla 2.3: Compatibilidad de Navegadores.....	24
Tabla 3.1: Descripción de definiciones.....	37
Tabla 3.2: Versiones del documento.....	38
Tabla 4.1: Prueba de conexión PHP.	117
Tabla 4.2: Prueba de Filtración de datos PHP.	118
Tabla 4.3: Prueba generación del menú personalizado PHP.	118
Tabla 4.4: Prueba de conexión Silverlight.....	118
Tabla 4.5: Prueba de Filtración de datos Silverlight.....	119

Tabla 4.6: Prueba de compatibilidad de navegadores Silverlight.....	119
Tabla 4.7: Métricas de fiabilidad	119
Tabla 4.8: Métricas de eficiencia.	119
Tabla 4.9: Métricas de Integridad	120
Tabla 4.10: Métricas de Usabilidad.	120
Tabla 4.11: Métricas de Facilidad de Mantenimiento	120
Tabla 4.12: Métricas de Facilidad de Prueba.....	120
Tabla 4.13: Métricas de Portabilidad.	121
Tabla 4.14: Métricas de Reusabilidad.....	121
Tabla 4.15: Métricas de Interoperabilidad.....	121
Tabla 4.16: Cuadro comparativo PHP y Silverlight.	121

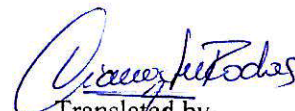
RESUMEN

El propósito principal de este proyecto es analizar las ventajas y desventajas del uso de la tecnología Rich Internet Applications (RIA) para el desarrollo de aplicaciones de escritorio más factibles que las tradicionales aplicaciones Web.

Para lo cual se desarrollo bajo los lenguajes PHP y Javascript, RIA sobre Silverlight una aplicación administrativa para el Centro Educativo Bilingüe Cristiano Israel CEBCI llamado *SISTEMA DE INSCRIPCIONES, PRE MATRICULA, MATRICULA Y PAGO DE PENSIONES*, el cual nos ayudó a determinar mediante un cuadro comparativo bajo métricas de calidad las funcionalidades, eficiencia y limitaciones de la tecnología RIA sobre Silverlight.

ABSTRACT

The main purpose of this project is to analyze the advantages and disadvantages of using (RIA) Rich Internet Applications technology for the development of desktop applications that are more dynamic and intuitive than the traditional Web applications. For this purpose, a management application system for CEBCI (Comunidad Educativa Bilingüe Cristiana Israel) Educational Institution was developed. The system is called INSCRIPTION, PRE-ENROLLMENT, ENROLLMENT AND TUITION PAYMENT SYSTEM, under the PHP/Javascript and RIA/Silverlight languages, which helped us to determine, through a comparative chart of quality measurements, the functionality, efficiency and limitations of RIA technology over Silverlight technology.


UNIVERSIDAD DEL
AZUAY
DPTO. IDIOMAS
Translated by.

Diana Lee Rodas

INTRODUCCION

En el desarrollo de la aplicación web para analizar las ventajas y desventajas del uso de la tecnología RIA mediante Silverlight se realizó una comparativa con PHP las cuales se verán en la implementación y diseño de esta tecnología en los capítulos siguientes.

En el primer capítulo se explica los conceptos para entender las aplicaciones web, para determinar cuáles son las funciones, estructura y los objetivos de tecnología RIA (Rich Internet Applications) lo que ayudará en la comprensión del siguiente capítulo.

En el segundo capítulo se analiza la tecnología RIA dentro de Silverlight y se determinan sus versiones, aplicaciones, ejemplos y comparación con otras tecnologías RIA.

En el tercer capítulo se presenta el análisis y diseño del sistema a desarrollar, el que se determinó con la recolección de la información necesaria para la elaboración de los diagramas de lenguaje unificado UML; además se representa el diseño del sistema.

En el último capítulo se explica la comparación de las tecnologías mediante pruebas de aplicación y calidad del sistema.

CAPÍTULO 1.

RIA (RICH INTERNET APPLICATIONS)

1.1. Introducción.

En la actualidad el uso de los sitios Web se han tornado más complejos debido a la demanda de los usuarios en sus requerimientos en la búsqueda de productos o servicios de una manera eficientes y de alta calidad. Debido a todos estos inconvenientes hicimos un recorrido sobre las aplicaciones Web 1.0 y su evolución a la Web 2.0 presentando sus características, estructuras, limitaciones y utilidades correspondientes.

Sin embargo el estudio se enfoca en la tecnología RIA (Rich Internet Applications) presentando la importancia en cuanto a su manejo y compatibilidad con diferentes sistemas operativos y navegadores Web. De esta manera se pretende evitar la renderización total de la página al cargar nuevos datos en la pantalla cuando el usuario lo requiere y así evitar problemas en el desarrollo total dentro de una aplicación.

1.2 ¿Qué es Aplicación Web?

Es una aplicación ubicada en el servidor Web el cual ofrece un determinado servicio y es accesible desde diferentes redes telemáticas, con la presencia de la más común “Internet” (World Wide Web).

1.2.1. Estructura

Su estructura suele presentar como un modelo de tres niveles.

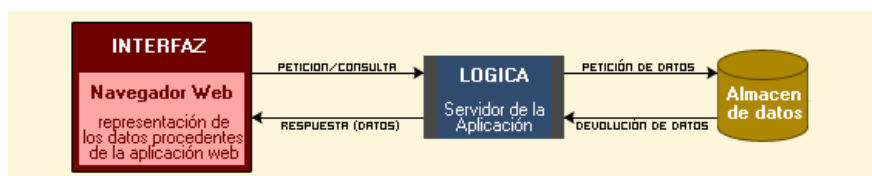


Gráfico 1.1: Estructura de tres niveles [2].

Nivel Interfaz.- Presenta las funciones que limitan el envío de los requerimientos del usuario, obtiene los datos generados por parte de la aplicación y finalmente presenta los datos en el navegador. Es decir, se encarga únicamente de la representación.

Nivel Lógico.- Se encuentra en la parte principal de la aplicación, y está encargado de brindar el contenido dinámico y ejecutarlo por completo en el servidor web.

Ofrece en su contenido lo siguiente:

- Interpretación de las consultas generadas por el cliente.
- Obtención o almacenaje de datos.
- Interacción con otras aplicaciones web.
- Envío de los datos generados al usuario.

Nivel Datos.- Este nivel podría estar basado en ficheros XML o en otro tipo de almacenaje, dependiendo de las necesidades de la aplicación.

1.2.2. Utilidades

Las utilidades de una aplicación web son muy extensas ya que dependen de los usuarios y sus requerimientos. Pero se podría clasificar en tres categorías como:

- Particulares: Web mails, Agendas de contactos, Calendarios, Tareas.
- Empresas: Intranets, Noticias, Facturación, Gestión de proyectos.
- Comunidades: Foros, Chats, Libros de visitas, Portales.

1.3 Evolución

1.3.1 La Web 1.0 ó Web tradicional.

El concepto general de web 1.0 se define como: “Un tipo de web estática con documentos que jamás se actualizaban y los contenidos dirigidos a la navegación (HTML y GIF)” (PUJOL, 2007). En las aplicaciones tradicionales existe una recarga continua de páginas, esto ocurre cada vez que el usuario desea ingresar a una página por defecto. De esta manera se produce un problema de datos entre el cliente y el servidor causando una recarga total dentro de la aplicación.

Entre las desventajas de las aplicaciones tradicionales esta la deficiente capacidad en multimedia, puesto que se debe tener otro programa externo para su reproducción eficiente y de alta calidad.

1.3.1.1 Características:

Las capacidades de las tecnologías cliente-servidor han ido evolucionando tanto en su optimización, estandarización y dinamismo. Como ejemplo tenemos los estudios sobre usuarios de aplicaciones *e-commerce* que demostraron las inconformidades en lo siguiente: (PALACIOS, 2008)

- Dificultad de navegación
- Lo difícil que se hace encontrar un producto sobre un sitio de estas características.
- Las malas experiencias en la compra de productos online.

Hoy en día las aplicaciones desean adaptarse a la forma en que deben visualizarse las cosas que requieren los clientes de acuerdo a sus requerimientos. Intentar representar una funcionalidad interactiva requiere de varios pasos ya que HTML tiene varias limitaciones lo

cual haría más confusa los requerimientos del usuario. Los usuarios han sido convencidos que se debe llevar a cabo una tarea de la siguiente manera: (PALACIOS, 2008)

- Todas las tareas ahora deben ser Multi-pasos.
- Todas las tareas deben ser un flujo de varias páginas.
- Es necesario navegar hacia otra página para ver los resultados de la búsqueda que el usuario seleccionó, en vez de mostrarlos sobre la misma página.
- No indicarnos que un link lleva a una página que no tiene resultados.

1.3.1.2. Limitaciones:

En las aplicaciones tradicionales se presentan las siguientes limitaciones o restricciones:

Personalización.- Existe demanda por la mayoría de las empresas en visualizar sus productos, servicios y combinaciones en el ingreso de nuevos datos.

Diseño pobre de datos.- En las aplicaciones tradicionales lo común es hacer el modelo de datos y sus relaciones al último momento, hace difícil la navegación lo cual conlleva a la malas experiencias y expectativas de los usuarios.

Búsquedas complejas.- La manera de encontrar productos o servicios se torna complejo en la mayoría de las aplicaciones.

Ausencia de retroalimentación (Feedback).- Al necesitar que las aplicaciones Web sean más interactivas para poder dar un cambio en el estado de respuesta de los datos al usuario.

Páginas HTML.- Se basan en Internet con servidores acoplados los que proveen la inteligencia en la estructura de los requerimientos de los usuarios, lo que es un reto en las aplicaciones web.

Serialización.- Dentro de la aplicación solo se consigue mediante la utilización de cookies los cuales no soportan objetos.

Debido a todo lo descrito anteriormente las aplicaciones se han convertido en páginas de difícil manejo en base a todos los requerimientos del usuario, los cuales deben estar plasmados de una manera más confiable y productiva.

1.3.2. La Web 2.0.

La web 2.0 está orientada a la interacción de los contenidos y redes sociales, los cuales pueden explotar todos los efectos de las redes haciéndola más interactiva y funcional. En 2003 Tim O'Reilly definió el concepto de web 2.0 como: "Web 2.0, subrayamos un cambio de paradigma sobre la concepción de Internet y sus funcionalidades, que ahora abandonan su marcada unidireccionalidad y se orientan mas a facilitar la máxima interacción entre los usuarios y el desarrollo de redes sociales (tecnologías sociales) donde puedan expresarse y opinar, buscar y recibir información de interés, colaborar y crear conocimiento (conocimiento social), compartir contenidos." (MARQUES, 2007)

De esta manera se puede entender como: "Todas aquellas utilidades y servicios de Internet que se sustentan en una base de datos, la cual puede ser modificada por los usuarios del servicio, ya sea en su contenido (añadiendo, cambiando o borrando información o asociando metadatos a la información existente), bien en la forma de presentarlos, o en contenido y forma simultáneamente." (RIBES, 2009)

La estructura de la Web 2.0 es compleja y va en evolución, la cual incluye el software de servidor, sindicación de contenidos (RSS), protocolos de mensajes, navegadores estándares y varias aplicaciones para los clientes.

Entre las técnicas tenemos:

- CSS, marcado HTML válido semánticamente y Micro Formatos
- Técnicas de aplicaciones ricas no intrusivas (AJAX)
- Java Web Start
- XUL
- Sindicación/ Agregación de datos en RSS/ ATOM
- URL sencilla y con significado (SEM)
- Soporte para postear en un blog
- XML
- Algunos aspectos de redes sociales

- Mashup (Aplicación Web híbrida).

La web 2.0 tiene como característica principal, la facilidad en el control de datos ya sea para extraer o introducir los mismos sin complicaciones, los usuarios son los únicos dueños de sus datos los cuales pueden ser manejados desde cualquier navegador basado en estándares, como:

Redifusión de contenido.- Este utiliza protocolos estándares que les permite a los usuarios ver los contenidos de la aplicación desde otros contextos (otra Web), o una aplicación de escritorio; éstos se encuentran en RSS, RDF los cuales son derivados de XML.

Servicios Web.- Son protocolos de mensajes bidireccionales como RestFul (indica un tipo de llamada a un servicio web donde el cliente transfiere todos sus estados) y SOAP (depende del servidor para retener la información de estado). Estos son llamados desde un API, este debe estar personalizado en función de las necesidades específicas del sitio web. Pero hay que tener en cuenta que los API's de los servicios estándares como un blog, están muy extendidos; pero pueden existir excepciones.

Software de servidor.- Los servicios web requieren de un soporte de base de datos y un flujo de trabajo más robusto, y tienen una similitud a la funcionalidad de intranet tradicional de un servidor de aplicaciones.

1.4. Aplicaciones Ricas de Internet (RIA: Rich Internet Applications)

RIA es un nuevo tipo de aplicación web cuyo objetivo es incrementar y mejorar las opciones y capacidades de las aplicaciones web tradicionales. Estas aplicaciones son desarrolladas en los lenguajes de marcado propio y ejecutados en servidores propios. Las aplicaciones RIA's se basan principalmente en el usuario, aplicación que contiene en su desarrollo de trabajo la arquitectura para obtener efectos eficientes y acortar la brecha entre las aplicaciones web y las de escritorio. RIA involucra interacción, velocidad y flexibilidad; mediante el entorno de RIA, no se produce recargas de páginas ya que desde el inicio se presenta toda la página y los datos de la aplicación y así determinar lo que desea cargar.

1.4.1. Características

- La aplicación es enlazada desde una nueva página web o una nueva página perteneciente a la misma aplicación.
- La mejor interacción con el usuario, este brinda una reacción inmediata ya que no es necesario cargar toda la página al colocar nuevos datos específicos.
- La aplicación permite operaciones comunes en el usuario como: Drag&Drop (arrastrar y soltar) o animaciones de objetos.
- Mejora la expectativa visual por las aportaciones de nuevos componentes más avanzados.
- Permite crear aplicaciones más atractivas con la utilización de audio, video y gráficos.
- La mayoría de las aplicaciones RIA se basan en el lenguaje de programación XML, tanto para el intercambio de datos y la interfaz gráfica.
- Los servidores donde se ejecutan las aplicaciones son diversos pero los más comunes son los basados en Java.
- Disminuye el ancho de banda, con el uso de estas aplicaciones ya que estas permiten almacenar más información del usuario y reduce el uso de transacciones HTTP en el servidor.
- Requiere de la utilización de un RichClient en el lado del usuario el cual será el motor de la tecnología RIA.
- Visualización y ejecución en múltiples plataformas y dispositivos heterogéneos.
- Desvinculación de la presentación lógica.
- Permite la conectividad online/offline.
- Se posibilita la detección de eventos de los componentes y la actualización sin tener que hacer un refresco completo.
- La programación en RIA es compatible con todos los navegadores y evita implementaciones dentro de los mismos.

Estas características satisfacen la mayoría de los objetivos de las páginas web que deseamos en la actualidad y las que en años anteriores no se podían lograr. Con las aplicaciones RIA el cliente tiende a tener más responsabilidad, deja de ser solo el dibujante sino más bien pasa a ser el responsable de los datos que pida al servidor, el cual se hace más explícito en el envío y recepción de datos a modo asíncronico a un requerimiento del usuario.

RIA, acerca al cliente a las aplicaciones de escritorio y así dejar de ser un simple cliente-servidor. Esto se logra con la agregación de la funcionalidad sobre la arquitectura Multi- capa. Con todo esto se puede solucionar la mayoría de las limitaciones que presenta la web 1.0, además permite desarrollar la aplicación con requerimientos complejos reduciendo el costo del mismo.

1.4.2. Estructura

La estructura es similar a las aplicaciones web 1.0

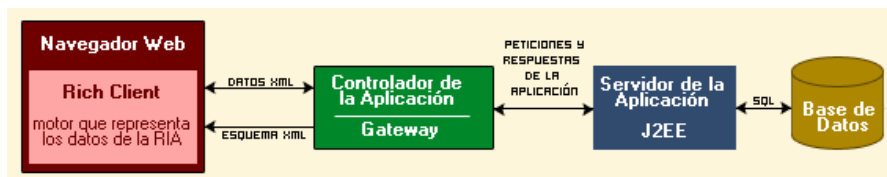


Gráfico 1.2: Estructura de cuatro niveles [2].

Primer Nivel.- Se tiene un navegador web que contiene un Plugin (RichClient) para interpretar los datos que ingresan a la aplicación como ficheros, videos, texto, audio, etc. Este almacena la parte gráfica para evitar la renderización total de una nueva página cada vez que se desea visualizar un nuevo dato en la aplicación.

Segundo Nivel.- Se tiene el contador de la aplicación y el Gateway. Esta es la parte de la aplicación que se encarga de interactuar con el cliente y viceversa; este utilizará otras partes para el ofrecimiento de servicios. Gateway se encarga de transformar los datos de la aplicación a un formato el cual el servidor lo pudiera entender.

Tercer Nivel.- Es el servidor donde se encuentra la aplicación, este se encarga de la compilación y permite la comunicación de la aplicación RIA con otras aplicaciones Servlets y de obtener los datos necesarios de la base de datos.

Cuarto Nivel.- Se tiene el almacenaje de la información, los cuales se dan en la base de datos relacionada con el servidor, la que ofrece la aplicación. Además esta parte no puede ser muy necesaria, ya que existen algunas aplicaciones que no necesitan de servidor para su ejecución.

Es importante tener en cuenta que la arquitectura RIA, en la parte correspondiente al cliente este contiene toda la parte gráfica de la aplicación, de esta manera se limita la comunicación entre cliente-servidor al traspaso de los datos. Existen 3 aspectos muy importantes que influyen de forma directa en el rendimiento de una aplicación.

- El diseño de la aplicación y el contexto de uso.
- El comportamiento y los requerimientos del usuario (lo que desea obtener).
- El comportamiento de la aplicación cuando se está utilizando.

Se debe tener en cuenta desde el instante que el usuario realiza la petición en el URL hasta que la respuesta sea enviada inmediatamente, presentamos los siguientes pasos con los cuales se obtendrá una serie de procesos.

Interacción entre el cliente y el servidor.- El usuario accede link del navegador, éste envía una solicitud al servidor, el cual debe responder inmediatamente a los clientes y determinar la cantidad adecuada de los contenidos pedidos por el usuario; el navegador lo visualizará, el usuario deberá esperar un determinado tiempo hasta que se complete el proceso.

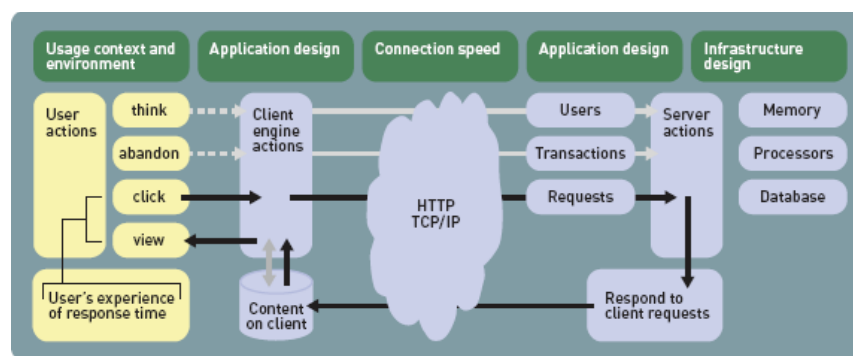


Gráfico 1.3: Esquema de interacción entre el cliente y servidor [6].

Comportamiento del servidor.- Los servidores deben dar respuesta óptima a las peticiones concurrentes del usuario, este consume una parte de los recursos del mismo como:

- Memoria.
- Ciclos de procesador.
- Recursos de motores de base de datos.

Los servidores web pueden responder inmediatamente a los requerimientos del usuario, creando caches de navegación, haciendo más eficientes las respuestas. Pero las acciones que involucran cambios en los datos consumen los recursos del servidor como por ejemplo agregar o quitar los servicios de los estudiantes dentro del pago de pensiones. En el siguiente grafico se presenta el rendimiento del servidor en función de la concurrencia de los usuarios y las transacciones. Todas estas funcionan hasta un cierto nivel de concurrencia llamada punto de inflexión, la cual se va degradando y se convierte en un cuello de botella.

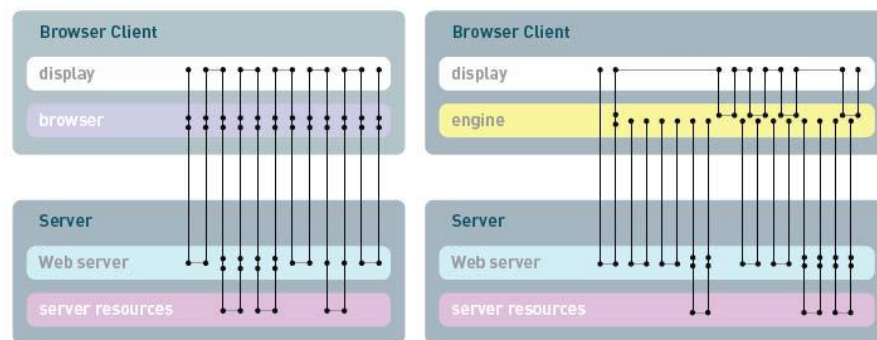


Gráfico 1.4: Esquema de rendimiento del servidor
(Web tradicional y aplicación RIA)[6].

1.4.3. Ventajas de la Estructura.

Las aplicaciones RIA tienen como características “Look and Feel” la cual se aproxima a una aplicación de escritorio, pero tiene tiempo de respuesta parecidos en base a la estructura que la defina.

Entre las que podemos presentar son las siguientes:

Balance de carga entre el cliente y el servidor.- La demanda entre el cliente y el servidor ya se encuentra más balanceada, de tal manera que los servidores web no necesitan ser grandes servidores como las que necesitan las aplicaciones web tradicionales, gracias a esto quedan más recursos disponibles y el servidor puede manejar más sesiones de usuarios concurrentes.

Comunicación asíncrona.- Al tener un “Motor cliente” este puede interactuar con el servidor sin la necesidad de completar una acción en la interfaz. Todo esto le permite al usuario ver e interactuar con la página asincrónicamente desde el “Motor cliente” con el

servidor. RIA permite mover datos entre el cliente y el servidor sin hacer esperar a los usuarios.

Eficiencia en el tráfico de red.- El tráfico puede ser disminuido ya que el “Motor cliente” de una aplicación puede ser más inteligente que un cliente común. Todo esto puede acelerar la respuesta ya que son menos los datos que se intercambian en cada interacción y reduce la carga de red.

1.4.4. Ventajas de la aplicación RIA.

Las aplicaciones RIA tienen la capacidad principal que funcionan en la mayoría de los navegadores web.

- No requieren instalación- actualizar y distribuir la aplicación instantáneamente.
- Las actualizaciones a nuevas versiones son automáticamente.
- Los usuarios pueden utilizar la aplicación con conexión a internet sin importar que sistema operativo se encuentre instalado en el computador.
- Las aplicaciones RIA son menos propensas a la infección de virus ya que estas manejan las interacciones con el usuario a través de un “Motor Cliente”.
- Se puede dar comportamiento a las interfaces de usuarios tales cosas que no se podían dar con Widgets HTM. Mediante su funcionalidad se puede implementar en la tecnología para el uso del usuario, utilizando un componente deslizante que permita cambiar los ítems, ejecutar cálculos del cliente sin necesidad de enviar una petición al servidor.
- Son aplicaciones más ágiles mediante estándares sucesivos de interacción con el servidor.
- Evita la problemática del uso de diferentes navegadores al abstraerse de ellos a través de un Framework.

1.4.5. Utilidades

En cuanto a sus utilidades se han incrementado en Web mails, foros, libros de visitas, agendas, etc.

En la actualidad existen empresas que desarrollan herramientas para la elaboración de aplicaciones RIA entre las que podemos mencionar son:

- Flash y Flex de Adobe

- AJAX
- OpenLaszlo (Herramienta Open Source)
- Silverlight de Microsoft
- JavaFx Script de Sun Microsystems
- NexaWeb, entre otros.

Son aplicaciones que abarcan amplio espectro de áreas y bajo costo de desarrollo, provee la mejor interactividad con propiedades de multimedia disponible. Estas combinan los estándares, tecnologías, con el objetivo de mejorar la calidad de las aplicaciones web las cuales describimos en el siguiente capítulo.

1.5. Herramientas de diseño RIA

Las aplicaciones RIA se encuentran en una etapa de desarrollo y esperando la opinión de los usuarios. Las restricciones y requerimientos en cuanto a las herramientas de desarrollo son las siguientes:

Browser modernos.- Las aplicaciones web requieren browser modernos para su buen funcionamiento, además debe tener un motor JavaScript para el manejo de los XMLHttpRequest en la comunicación cliente-servidor. Deberá adoptar los estándares DOM y manejo de hojas de estilo CSS.

Estándares web.- El problema de escribir aplicaciones con las funcionalidades de JavaScript, es que no todos los browser tienen los mismos estándares, eso hace que se deba escribir código adicional para el funcionamiento en distintos browser.

1.6 Conclusiones

Este capítulo presentó todas las ventajas y desventajas entre la web tradicional y RIA; las cuales nos dan su funcionalidad entre el cliente y servidor. Sin embargo cuyas limitaciones se basan en el manejo de la información a presentar; RIA nos da la facilidad de acceso una sola vez al servidor y así evitar que se sobrecargue la página cada vez que se

desea hacer una consulta mientras que la web tradicional necesita de grandes servidores para su acceso a la información pues esta tiene grandes recargas de página.

1.7. Recomendaciones

La utilización de las tecnologías web dependerá de la facilidad de manejo de la información y los requerimientos del usuario. Puesto que la web tradicional tiene inconvenientes con formatos multimedia lo que se requiere en la actualidad.

CAPÍTULO 2.

RIA DENTRO DE SILVERLIGHT.

2.1. Introducción

Este capítulo presenta una visión general de Silverlight que es la herramienta RIA desarrollada por Microsoft dentro de su marco de trabajo .NET. Esta herramienta ha experimentado varios cambios a través de la historia evolucionando así hasta su versión actual 4.0 y muy cerca de llegar a la 5.0. En cada una de sus evoluciones ha generado cambios concretos mejorando así su presentación, interoperabilidad, librería de controles, entre otros.

A continuación, el capítulo enfoca las principales funcionalidades de la herramienta revisando tareas básicas que permiten a los desarrolladores generar todas las aplicaciones para la red. Luego podremos ver el impacto de Silverlight en el mercado de aplicaciones web durante los últimos años. Para ello se examinan algunos ejemplos de aplicaciones ya en producción, que han sido desarrolladas bajo esta herramienta y analizaremos algunas características interesantes. Finalmente tendremos una revisión de las tecnologías RIA más importantes que hoy por hoy compiten en el mercado junto a Silverlight.

2.2 Introducción a Silverlight.

Las empresas de software para desarrollo saben que la Web es la plataforma que hospedará el futuro de las aplicaciones y servicios informáticos. Con eso en mente, han venido desarrollando una serie de herramientas que permitan crear aplicativos dinámicos, intuitivos, funcionales y enriquecidos con imágenes, video, música, etc.

Una de las herramientas de mayor impacto en los últimos años es Silverlight. Su posición de importancia no ha sido impuesta por el hecho de ser un producto Microsoft, sino que su potencia, alcance, operatividad y funcionalidad le han otorgado dicho puesto por mérito propio. Su capacidad de adaptarse a los entornos de desarrollo, su acoplamiento a varios exploradores y sistemas operativos, su continuo crecimiento e innovación, hacen que varios desarrolladores elijan esta herramienta para generar sus aplicaciones Web.

Silverlight es una herramienta que no ha detenido su evolución desde sus inicios en el mercado RIA, lo que le ha permitido adaptarse a nuevas tendencias informáticas, facilitar el diseño visual de sus aplicaciones mediante el uso de herramientas como ExpressionBlend, mejorar su compatibilidad con nuevos exploradores y agregar funcionalidad para optimizar el rendimiento. Una de las razones de peso para el uso de la herramienta sobre otras del mismo tipo es la integración del lenguaje declarativo XAML dentro del marco de trabajo .NET. XAML es el corazón del desarrollo de Silverlight y permite crear interfaces de usuario asombrosas de una forma muy sencilla a comparación de anteriores tecnologías como JavaScript (DAYLEY, 2008).

Microsoft ha puesto mucho esfuerzo en desarrollar y mantener operativo a Silverlight. Muestra de ello es que en sus 5 años en el mercado ya se han desarrollado 4 versiones de la misma. Esto demuestra el compromiso de Microsoft con Silverlight y nos permite crear gran expectativa sobre su prometedor futuro (BEADLE & KRISHNAN, 2010).

2.3 Evolución de Silverlight.

Desde sus inicios alrededor del año 2007, Silverlight ha venido experimentando una constante evolución, la cual ha permitido a los desarrolladores experimentar mejoras en la funcionalidad, mayor interoperabilidad, facilidad de diseño visual y mejoras en el entorno de desarrollo, gracias al soporte que posee dentro del marco de trabajo .NET.

Para entender el porqué de Silverlight, es necesario tener conocimiento sobre una herramienta previa creada por Microsoft llamada WPF (Windows PresentationFoundation). Este Framework permite a los desarrolladores crear y gestionar interfaces de usuario visualmente atractivas de forma unificada para entornos de escritorio, web y dispositivos móviles (WindowsCliente.NET, 2010).

Microsoft se vio en la necesidad de mejorar la experiencia de usuario para Aplicaciones Web con una herramienta que permita crear y gestionar interfaces de usuario enriquecidas, potencializar las ventajas de WPF a la Web y estar a la altura del concepto de Web 2.0. Es así que en marzo del 2007, realizando un refinamiento a WPF, se crea WPF/E (Windows PresentationFoundation/Everywhere), que meses después sería conocido como la versión 1.0 de Silverlight (SCANLON, 2008). Esta versión introdujo el uso de un Plug-In en el lado del cliente para la ejecución de las aplicaciones Silverlight, pero presentaba una librería de controles muy pobre, imposibilitando a los desarrolladores generar aplicaciones de gran alcance.

2.3.1. Versión 1.0

La versión 1.0 implementó simples pero significativas características, donde la de mayor importancia fue la implementación de CLR (CommonLanguageRuntime) que permitió ejecutar aplicaciones bajo diferentes plataformas gracias a la adaptación de Silverlight dentro de .NET, con lo que los desarrolladores tuvieron la libertad de usar lenguajes como C# y VB. Entre las otras características tenemos:

- Códec embebido que da soporte a los videos VC-1 y WMV, además permite reproducir audio MP3 y WMA dentro del navegador.
- Se facilita la descarga progresiva y reproducción de contenido multimedia de cualquier Servidor Web.
- Permite opcionalmente la distribución de audio y video en internet (media streaming).

- Permite crear interfaces de usuario enriquecidas además de gráficos vectoriales con HTML.
 - Hace sencilla la creación de reproductores de video web.
- (ProgrammerWorld.NET, 2009)

2.3.2. Versión 2.0

La versión 2.0 fue presentada pocos meses después. Entre las mejoras se incluyó una librería de controles mucho más amplia que permitía ahorrar recursos usualmente perdidos en la generación de código JavaScript. Ahora los desarrolladores podían crear y manipular formularios y controles complejos sin mayor esfuerzo. Además se incluyó un sistema de control de eventos que facilitó la creación, manipulación y control de acciones dentro del entorno de desarrollo de la herramienta. Se introdujo el paquete XAP que contiene todos los archivos y recursos necesarios utilizados por las aplicaciones Silverlight (DAYLEY, 2008). Sus principales características son:

- Un motor CLR interno que genera un ambiente de alto desempeño para los navegadores. Silverlight usa el mismo motor central CLR que se utiliza dentro del marco de trabajo .NET.
- Se incluyó una librería enriquecida de clases que pueden usarse para desarrollar aplicaciones de navegador.
- Provee un API para controlar HTML y DOM y permite programar el HTML de navegador usando cualquier lenguaje de .NET.
- No requiere que ASP.NET sea usado como lenguaje de servidor, puede usarse PHP u otros. (ProgrammerWorld.NET, 2009)

2.3.3. Versión 3.0

La versión 3.0 fue lanzada en septiembre del 2009. Fue una extensión de Silverlight 2 y entre sus avances presentaba una capacidad grafica enriquecida, más controles e integración con la herramienta de desarrollo y diseño WEB ExpressionBlend. La mejora de mayor importancia en esta versión fue la capacidad de ubicar las aplicaciones Silverlight directamente en el computador, ya sea en el escritorio o en un directorio determinado, sin

necesidad de instalar ningún Plug-In al explorador (BEADLE & KRISHNAN, 2010). En esta versión podemos apreciar:

- Soporte para gráficos 3D permitiendo a los desarrolladores aplicar contenido 2D a un plano 3D transformando la perspectiva. Se puede simular rotación aplicando la perspectiva apropiada a los elementos XAML. Los desarrolladores no tienen que escribir ni una línea de código ya que todo esto se puede generar desde la herramienta ExpressionBlend.
- Se puede generar animaciones asombrosas utilizando diferentes funciones que vienen por defecto. Además da la oportunidad de crear efectos personalizados y animarlos utilizando nuestras propias formulas.
- Incluye un gestor de media mejorado que soporta flujo multimedia de alta calidad.
- Da soporte para nuevos formatos como (AAC)/MP4 y la nueva línea de audio/video RAW que da soporte a códec de terceros, brindando la oportunidad de desarrollar un rango de formatos amplio.
- El SDK de Silverlight 3 provee controles adicionales para crear aplicaciones en un ambiente de desarrollo rápido. (Ghoda, 2010)

2.3.4. Versión 4.0

La versión 4.0 salió al mercado en marzo del 2010. Incluyó soporte para el uso de micrófono y cámaras web para la captura de voz y video en el equipo cliente, además permitió la compatibilidad con nuevos exploradores como Google Chrome. A todo esto se agregó soporte a varios formatos de video y se optimizó el rendimiento de las aplicaciones permitiendo que se ejecute mucho más rápido (Microsoft, 2010).

2.4 Principales funcionalidades de la herramienta.

El objetivo final del desarrollo con Silverlight es obtener aplicaciones funcionales, visualmente atractivas ambientadas en Internet y sin tanto esfuerzo en programación. Con esto en mente analizaremos un conjunto de funcionalidades que definen su comportamiento y le dan forma como herramienta

2.4.1. Experiencia de usuario multiplataforma

- Silverlight genera una experiencia única de navegación mediante la implementación de aplicaciones de internet enriquecidas con video, animaciones, interactividad e interfaces de usuario impresionantes.
- Su instalación es increíblemente simple debido a plug-in que pesa menos de 2 megabytes, instalado bajo demanda y que funciona con todos los navegadores de importancia en estos días.
- Genera experiencias consistentes en Windows y MacOS sin ningún requerimiento adicional de instalación.
- Permite crear una experiencia Web impresionante al tomar ventaja del desempeño del cliente que es el que ejecutará toda la aplicación, liberando así la carga al servidor e incrementando el desempeño general del sistema. Impresionantes gráficas basadas en vectores, media, textos, animaciones, entre otros además de facilitar la integración de efectos en cualquier aplicación Web existente. Mejora las aplicaciones estándar basadas en AJAX con graficas mejoradas y utilizando todo el potencial de Silverlight.

2.4.2. Modelado de programación flexible con herramientas de colaboración.

- Basado en el marco de trabajo .NET, Silverlight permite a los desarrolladores y diseñadores el uso de herramientas existentes para generar experiencias multimedia enriquecidas para la Web.
- La opción de elegir desarrollar la aplicación entre varios lenguajes de programación tales como AJAX, Visual Basic, .NET, C#, Python y Ruby otorga a los desarrolladores y diseñadores la flexibilidad de usar sus habilidades sin la necesidad de aprender un nuevo lenguaje. Silverlight se integra fácilmente con cualquier tecnología de servidor. Se puede vincular fácilmente a cualquier infraestructura existente incluyendo Apache y PHP en el servidor o JavaScript y XHTML en el cliente. Herramientas específicas para diseñadores y desarrolladores que aprovechan los estándares Web y el software de Microsoft .NET.

2.4.3. Multimedia de alta calidad a bajo costo

- Formatos de media unificados desde HD a WMV (Windows Media Video), la implementación de Microsoft del estándar de video VC-1 de SMPTE (Society of Motion Picture and Television Engineers), así como soporte para WMA (Windows Media Audio) y MP3.
- Soluciones flexibles de inserción de video y animaciones incluyendo la capacidad de distribuir en la red de manera fluida y sin pérdida de fidelidad visual o calidad de movimiento.
- Costo bajo de distribución multimedia con las tecnologías de Windows Media que disminuyen significativamente el costo de envío de datos. Se puede reducir aún más los costos con el paquete Media de Microsoft IIS (Internet Information Services).

2.4.4. Conexión de datos, servidores y servicios.

- Creación de mash-ups incorporando datos y servicios desde cualquier lugar en la web mediante el soporte de Silverlight para LINQ. Acceso a los datos mediante protocolos comunes como JSON, RSS, POX y REST.
 - Incrementa la capacidad de indexar y encontrar contenido RIA en internet, gracias al formato basado en texto XAML.
- (Ghoda, 2010)

2.5 Ejemplos.

Silverlight ha venido ganando terreno en varios campos de negocio y producción. En la actualidad existen varias empresas que utilizan esta tecnología como base de desarrollo para sus aplicaciones y aprovechan sus características para generar una experiencia interactiva, intuitiva, atractiva y funcional a todos los usuarios que se permiten usar sus aplicaciones Web. A continuación presentaremos algunos sitios Web que utilizan Silverlight describiendo sus características y desempeño.

El sitio web de “Hard Rock Café Memorabilia” ubicado en la dirección <http://memorabilia.hardrock.com/> presenta una portada diseñada en tonalidades oscuras. Un menú principal sencillo sin opción a desplegar, y un menú secundario a modo lista. Sin duda alguna la parte significativa y atrayente de la página es su sección central donde muestra un collage con imágenes de todos los productos que ofertan.

Si uno realiza un acercamiento lo suficientemente amplio como para centrar en pantalla un solo producto, un cuadro de información del producto aparece. Además cuando se especifica una búsqueda, los resultados sales del mismo collage de imágenes pero se organizan mediante un efecto muy simpático que permite crear un collage más pequeño pero solo con los productos buscados. Sin duda alguna es un ejemplo muy bueno de las capacidades de Silverlight.

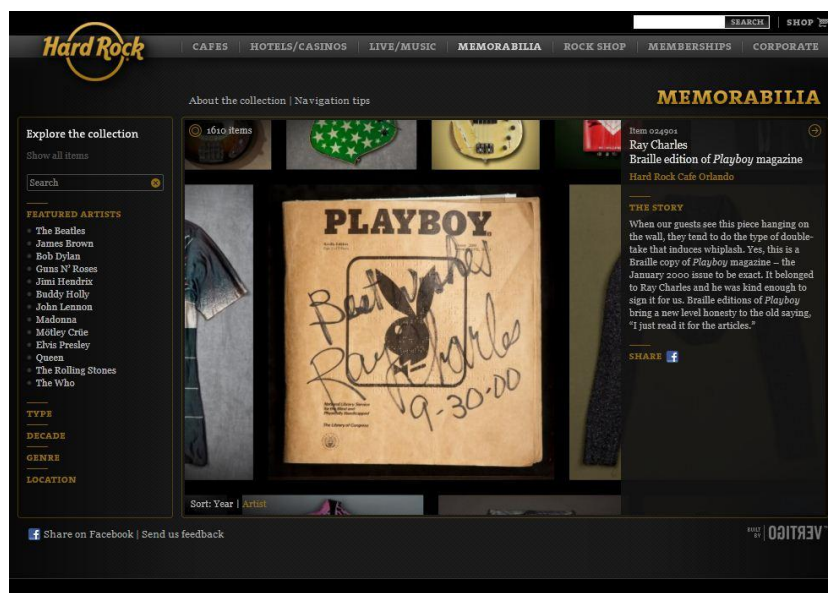


Gráfico 2.1: Representación de un collage en la página HardRock [21].

Otro sitio de mucho interés es Microsoft HealthCommonUser Interface con dirección en <http://pjd.mscai.net/default.htm>. En este sitio primeramente seleccionaremos la opción que nos convenga. Luego no llevara a distintas pantallas donde luego de esperar por un momento podremos interactuar con su contenido y podremos ver que la experiencia es prácticamente como ejecutar una aplicación de escritorio.

La página muestra distintos paneles que se pueden mover y ajustar a nuestra medida. Además una herramienta de búsqueda de pacientes que permite elegir la manera en que se

muestran los datos encontrados, aumentando o disminuyendo la información que se muestra por paciente. Una vista de línea de tiempo con información relevante a donde y cuando se atendió al paciente y así entre otras cosas.



Gráfico 2.2: Representación de búsqueda de información [23].

2.6 Pruebas de concepto.

Las pruebas de concepto nos permitieron descubrir ciertas habilidades de la herramienta cuando se desenvuelve en escenarios de aplicación específicos. A continuación se muestra algunas tablas con las pruebas efectuadas describiendo la expectativa que se generó, el resultado obtenido sobre dicha expectativa y algunas observaciones relevantes a la prueba desarrollada.

Prueba	Conexión a la base de datos de MySQL.
Expectativa	La aplicación a nivel de servidor puede conectarse al gestor de base de datos MySQL mediante un conector ODBC.
Resultado	Se cumplió la expectativa.
Observaciones	La conexión se logró luego de instalar el Conector MySQL de .net y MySQL ODBC. Luego de una configuración rápida del conector recientemente instalado fue necesario crear una clase cuya función sería efectuar la conexión y así se logró tener acceso a la base de datos desde Silverlight.

Tabla 2.1: Conexión a la base de datos MySQL.

Prueba	Vincular los datos de la base con la aplicación Silverlight en el equipo cliente.
---------------	---

Expectativa	La aplicación a nivel de cliente puede acceder a los datos de la base para efectuar todas las operaciones de mantenimiento (insertar, modificar, eliminar).
Resultado	Se cumplió la expectativa.
Observaciones	Aplicando Servicios Web se logró establecer un contrato donde el programa cliente tendría acceso a información específica de la base de datos mediante solicitudes preestablecidas.

Tabla 2.2: Vinculación de datos con la aplicación Silverlight.

Prueba	Compatibilidad con los navegadores más importantes (Internet Explorer, Mozilla Firefox y Google Chrome).
Expectativa	La aplicación se ejecuta en los navegadores más importantes sin presentar variación de comportamiento en absoluto.
Resultado	Se cumplió la expectativa.
Observaciones	El programa fue revisado en cada uno de los navegadores ya mencionados en busca de incompatibilidades funcionales o visuales, pero no se encontró ninguna variación.

Tabla 2.3: Compatibilidad de Navegadores.

2.7 Comparación de Silverlight con otras herramientas RIA.

El mercado de RIA ha tenido un gran impacto en las empresas desarrolladoras de herramientas de programación, lo que compromete cada vez más a ingenieros y diseñadores para elaborar mejoras funcionales a sus herramientas y así puedan sobresalir sobre sus competidoras. Entre las herramientas más importantes tenemos:

2.7.1. JavaScript/ Ajax

Java (Asynchronous JavaScript) es una técnica, no tecnología para el desarrollo de aplicaciones RIA. Se basan en:

- Presentación basada en estándares XHTML y CSS.
- Cambios dinámicos en la presentación del contenido y el control de eventos del usuario a través del DOM.
- Intercambio de datos y manipulación mediante el uso de XML y XSLT.

- Obtención de datos de forma asíncrona a través de XMLHttpRequest JavaScript para la unión de todo.

- El DHTML ó LAMP (glosario) no constituyen una tecnología en sí, pero trabajan conjuntamente. El hecho que trabajen en el intercambio de datos para obtener una aplicación más transparente con el servidor.

Ajax es un término nuevo el cual esta combinado y está siendo usado por Google para los proyectos Gmail y Google Maps. Varios Frameworks Open Source han sido desarrollados así como Frameworks comerciales.

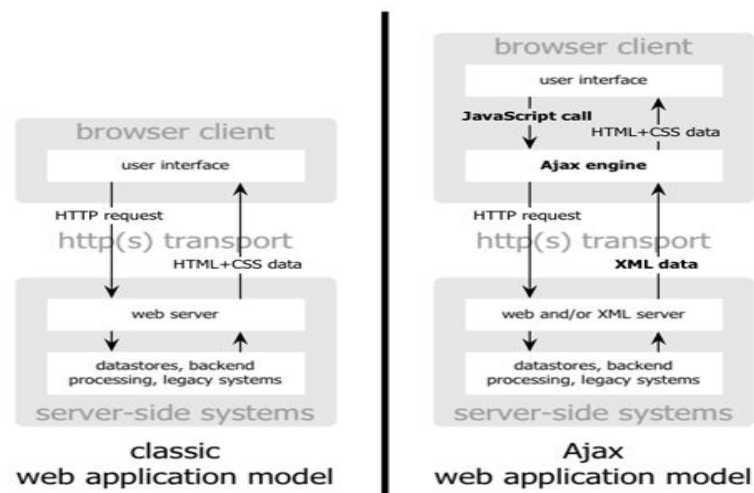


Gráfico 2.3: Modelo JavaScript/ Ajax. [15]

2.7.2. Google's GWT Framework

Google fue una de las primeras aplicaciones publicada en el 2006, el cual lanzo una herramienta para el desarrollo de las aplicaciones RIA “Google Web Toolkit” llamado GWT.

Este permite desarrollar aplicaciones RIA utilizando el lenguaje de programación Java. El paradigma de la programación GWT se centra en codificar la lógica de la interfaz de usuario en Java (similar a Swing/AWT) y ejecuta el compilador para que se pueda ejecutar en los browser compatibles con JavaScript.

Este Frameworks fue diseñado solo para los programadores java, el cual permite la programación, debug, testeo de aplicaciones RIA, todo esto se logra utilizando herramientas adecuadas como Eclipse y lo más importante de esta herramienta es que se la puede realizar sin tener conocimientos del lenguaje de JavaScript o DOM de los navegadores web.

2.7.3. Adobe Air

Usa las mismas tecnologías que utiliza la red como HTML, Ajax, Flash, Flex para el desarrollo de aplicaciones de escritorio. Entre las características más destacadas que es multiplataforma la cual puede funcionar en distintos sistemas operativos. Se basa en WebKit, permite la posibilidad de acceder a ficheros locales y a una base de datos.

2.7.4. Adobe Designer

Estas se basan al PDF para la presentación de información y XML para el procesamiento de los datos, este permite crear formularios en los que se combinan una presentación de alta fidelidad con la gestión de datos XML. Su interfaz gráfica es de fácil funcionalidad cual permite crear formularios, plantillas y visualizarlos antes de implantarlos como PDF o HTML. También mejora la seguridad de los documentos a sus diseños de plantilla con las firmas digitales.

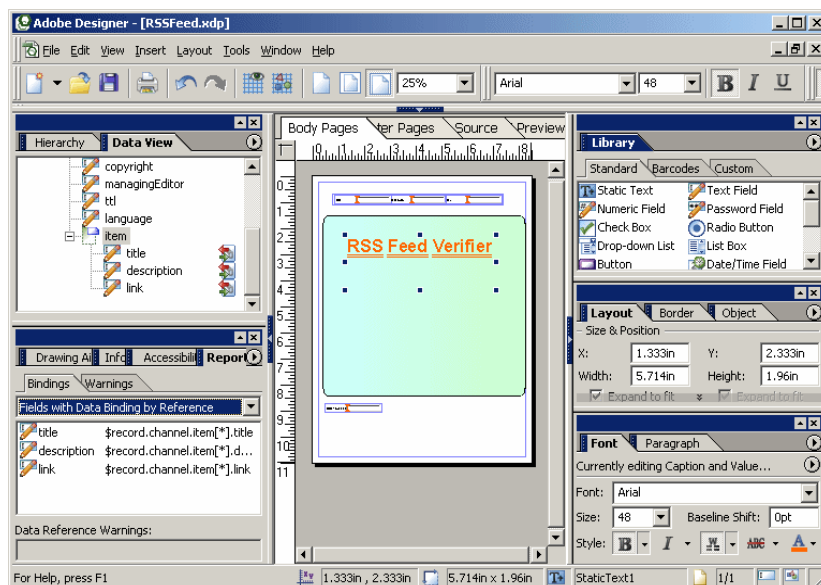


Gráfico 2.4: Modelo Adobe Designer. [1]

2.7.5. Appcelerator

Es una plataforma Open Source para el desarrollo de aplicaciones RIA. Utiliza un servicio basado a la arquitectura y estándares tales como HTML, CSS, y JavaScript. Estas pueden ser integradas automáticamente en distintas capas de servicios de desarrollo con Java, PHP, Python, .NET y Perl. A la vez se puede utilizar para la creación de Widgets y unirlos con RIA.

2.7.6. OpenLaszlo

Es una plataforma de código abierto creado por LaszloSystem Inc. El compilador OpenLaszlo compila los programas de escritorio en LZX y LaszloPresentation.

LZX: es un lenguaje descrito en XML y JavaScript que ese encarga de la capa de presentación para las aplicaciones RIA, ofrece una definición de interfaces simples y extensas dentro de DHTML los cuales exportan Flash 7 y Flash 8.

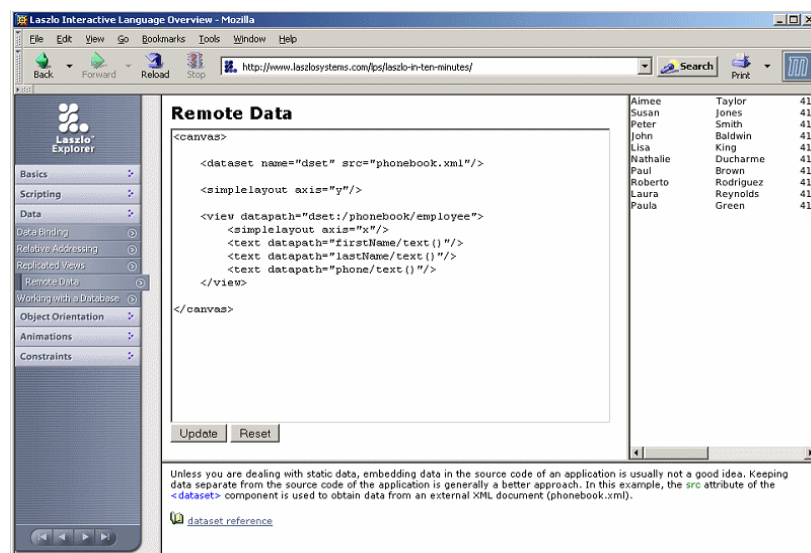


Gráfico 2.5: ModeloOpenLaszlo. [8]

LaszloPresentation: Es un Servlets que compila las aplicaciones LZX en ficheros SWF. Además cachea las aplicaciones para ser ejecutados por cualquier navegador web.

Con OpenLaszlo se podrá realizar lo siguiente:

Desarrollar aplicaciones RIA en lenguajes de programación estándar XML y JavaScript.

Desplegar las aplicaciones en cualquier navegador basado en J2EE, en las plataformas Linux, Unix, Windows o MAC.

2.7.7. MagicAjax.NET

Es un Framework Open Source, diseñado para la agilidad en el desarrollo de los programadores el cual permite agregar Ajax a las páginas web de una manera más intuitiva, se puede seguir utilizando los controles ASP.NET y no necesitamos código JavaScript.

2.7.8. Curl 5.0, REBOL 2.6, y Seaside for Smalltalk

Existen aplicaciones RIA en Java las cuales incluyen máquinas virtuales para Curl, Rebol y Smalltalk.

Curl: No facilita la constancia de los datos del usuario. Rebol: No requiere de browser. Smalltalk: Utiliza la versión reducida del lenguaje para una mejor expectativa web más rica.

2.7.9. JavaFX

Sun Microsystems presentó a JavaFX entre uno de los productos pertenecientes a su familia basada en la tecnología Java con el cual se pretendía proveer una variedad de dispositivos incluidas las de escritorio como Applets.

La plataforma JavaFX inicia como consistente de JavaFX script y JavaFX Mobile. El primero permite un desarrollo rápido de interfaces en 2D usando SVG (Scalable Vector Graphics). El segundo es un producto comercial disponible para los fabricantes de celulares.

2.7.10. Java

Las aplicaciones RIA escritas en Java pueden ser ejecutadas en un navegador como cualquier otra aplicación. Estas tienen la ventaja de usar todo su potencial de la plataforma Java para darle riqueza a las aplicaciones ya sean en 2D o 3D.

Java es una plataforma que tiene un amplio rango de librerías comerciales y abiertas, incluyendo soportes para cualquier sistema como JNI o JNA. Hoy en día existen Frameworks para el desarrollo con Java como XUI, Swixml, Cnoos y UltraLightClient.

2.7.11. NexaWeb

Es una plataforma de desarrollo RIA que utiliza usuarios basado en Java (Applets) para la ejecución y recepción de instrucciones en XML precedentes de un Servlets. El usuario puede ser ejecutado en varios navegadores web y puede funcionar en la mayoría de los sistemas operativos. Ofrece IDE para el desarrollo de las aplicaciones la cual se debe instalar una máquina virtual la que posee de un editor que ejecuta a través de la web la cual se realiza en online.

Entre las características están:

- Es extensible y escalable, su arquitectura está directamente con Internet.
- El cliente no necesita instalar ningún software adicional.
- Tiene una interfaz gráfica potente y funcional.
- Los datos se actualizan en tiempo real.
- El cliente es multiplataforma y soporta varios navegadores Web.
- La comunicación cliente/servidor no son muy concurrentes y la actualización se la realiza sin refrescos.

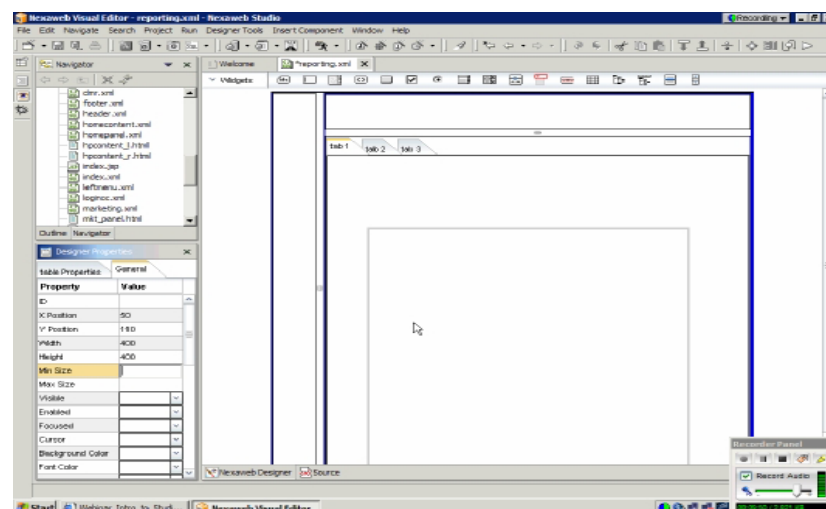


Gráfico 2.6: Modelo NexaWeb. [8]

- Soporta comunicaciones seguras mediante protocolos SSL sobre HTTP.
- Permite hacer clusterización de servidores.
- Soporta la interacción con Web Services.

2.7.12. Macromedia Flex

Es un servidor de presentación y estructura que permite crear aplicaciones dinámicas con la utilización de herramientas de diseño actual. Antiguamente llamado “Royale” este ofrece una metodología de programación declarativa, basada en estándares y servidores runtime de servidor el cual ofrece interfaces de usuarios inteligentes a través de Macromedia Flash.

Está construido para cubrir las expectativas y necesidades prácticas de las aplicaciones con el uso de las herramientas más óptimas.

Flex define las interfaces de usuario dinámico utilizando un lenguaje basado en XML, cuyo servidor traduce las aplicaciones de cliente que se ejecutan sobre Flash Player.

La estructura combina una sencilla sintaxis de programación, una librería de clases y potentes servicios runtime para la conexión de datos, despliegue y gestión de expectativas futuras.

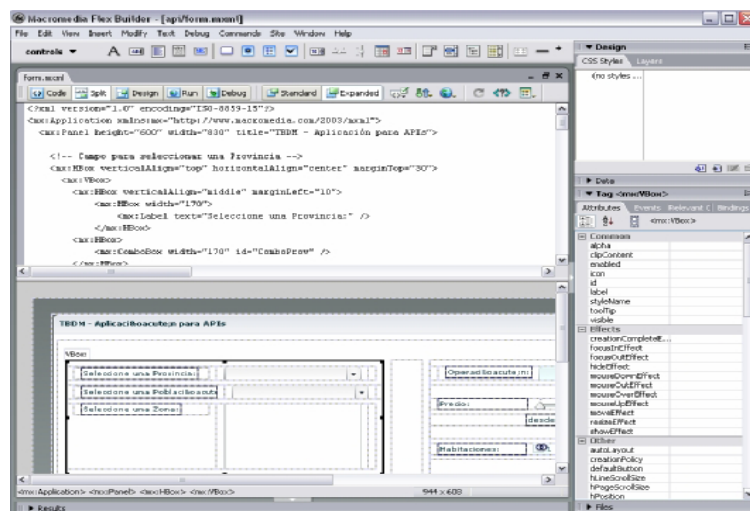


Gráfico 2.7: Modelo Macromedia Flex. [9]

2.7.13. NQ Suite

Es un producto creado por la empresa JackBe, este optimiza la ejecución de las aplicaciones RIA y apta para redes privadas de empresas con sucursales geográficamente dispersas. Su plataforma funciona con los navegadores estándares sin necesidad de instalar software en el cliente.

JackBe mejora la comunicación entre el cliente/servidor de manera que el usuario puede tener una experiencia similar al funcionamiento del ancho de banda. Ofrece un editor fácil de utilizar, en él se puede diseñar aplicaciones arrastrando los componentes al panel principal este posee una librería de herramientas, objetos y componentes que ayudan a su desarrollo.

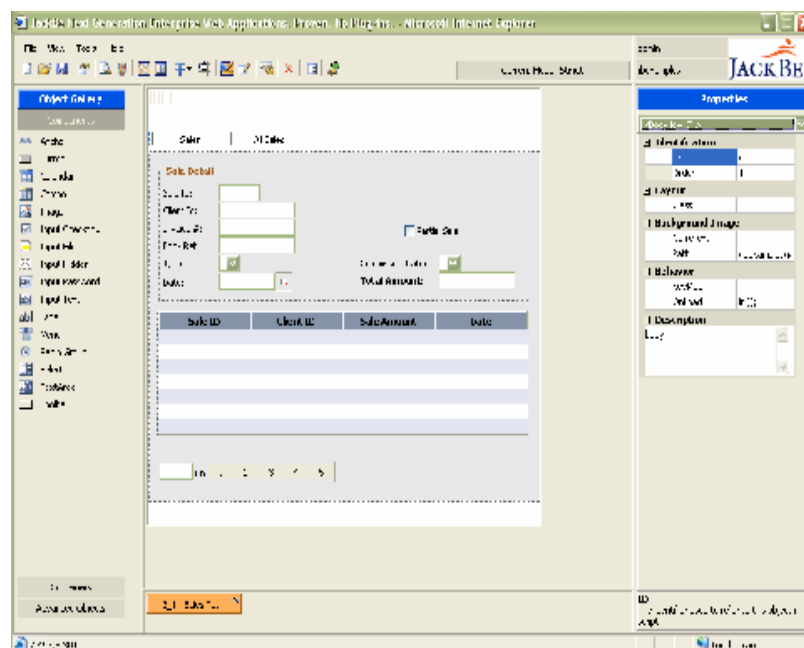


Gráfico 2.8: Modelo NQ Suite. [10]

2.7.14. Controles ActiveX

Al usar controles dentro de HTML es muy eficaz para el desarrollo de aplicaciones RIA, pero entre las desventajas es que su buen funcionamiento se efectúa solo con el navegador web de Internet Explorer. Se debe tener en cuenta que se presenta la existencia de más virus y Malware bajo esta técnica y al activarlas estamos más propensos a estos males.

El uso de esta técnica es una buena opción para las empresas que tiene a este navegador web como principal.

2.8. Conclusiones

El desarrollo de aplicaciones Silverlight para la red, son más intuitivos, dinámicos en multimedia, mediante ExpressionBlend la cual ayuda en la programación sin tener mucho esfuerzo. Además Silverlight es una herramienta de actualidad la cual ha evolucionado hasta la versión 4.0 y cerca de la 5.0 cada versión con mayores cambios y manejos de los objetos. Además de la comparación de Silverlight con otras herramientas nos da como resultado la facilidad de uso las herramientas y servicios que nos brinda.

2.9. Recomendaciones

Para el desarrollo dinámico se debe realizar prácticas en ExpressionBlend la cual ayuda en la parte de diseño en la inserción de imágenes, efectos, multimedia en sí; además se puede exportar documentos realizados en Flash o Adobe.

CAPÍTULO 3

ANÁLISIS Y DISEÑO DEL SISTEMA.

3.1. Introducción

Este capítulo presenta el análisis y diseño del Sistema de Inscripciones, pre matrícula, matrícula y pago de pensiones (SIPMP); el cual fue tomado como base para el desarrollo, y demostración de cualidades, características y utilidades en los dos lenguajes tanto en PHP MySql y Silverlight El desarrollo inicia con el levantamiento de la información mediante entrevistas con el personal administrativo del Centro educativo CEBCI luego de esto se determinó los modelos de análisis, casos de uso con sus respectivas especificaciones, Clases, Objeto Relación, Comportamiento (Actividad, Secuencia).

Seguidamente se presenta el modelo de diseño en el cual se especifica el diccionario de datos proveniente de la Base de Datos (Mysql); además del pseudocódigo que visualiza el flujo de codificación de uno de los módulos a desarrollar. Igualmente se presenta la interfaz gráfica de los dos lenguajes de una forma más intuitiva para el uso por parte del usuario. Al finalizar este capítulo presentamos la gestión de base de datos la cual nos ayuda a determinar las versiones de cambio que se presentan durante su desarrollo.

3.2. Recolección y levantamiento de la información.

3.2.1. Entrevistas y reuniones con el personal administrativo del centro educativo CEBCI.

La entrevista se realizó el día 27 de julio del 2010 en las instalaciones del centro educativo CEBCI con la presencia de la Lcda. Marielisa Ávila, contadora de la institución, Srta. Carla Zalamea, colectora, y la Srta. Cristina Montaña, secretaria.

La metodología utilizada fue netamente abierta, con un dialogo exhaustivo en el que se explicó en forma detallada el funcionamiento de cada uno de los departamentos de la institución, así como el manejo de los procesos de inscripciones, pre matriculas, matrículas y pago de pensiones. La Lcda. Ávila nos brindó documentos impresos utilizados en la administración del centro educativo, ya que actualmente carece de un sistema informático. Mediante las entrevistas se obtuvo una descripción detallada del sistema actual, la cual se relata a continuación.

- Inscripción: Este proceso se realiza solo para estudiantes nuevos que aspiran el ingreso a la institución educativa. Empieza 2 meses antes del periodo de matriculación. El proceso inicia con el pago de un rubro de \$20 que le otorga un cupo al interesado; se toma los datos del representante y números telefónicos para informar sobre cualquier cambio que pueda ocurrir o para recordar las fechas de matriculación, de forma que no pierda su cupo. Esta inscripción tiene un periodo de validez que vence la primera semana de clases; si en ese tiempo el interesado no se ha matriculado, pierde su cupo y su dinero no es reembolsado.

- Pre matrícula: Este proceso se realiza para estudiantes que han culminado el año en la institución educativa y desean continuar en esta. Empieza en la misma fecha de las inscripciones descritas anteriormente. Consiste en realizar un abono a la matrícula de \$30 mínimo con lo que asegura su cupo. El resto del dinero de la matricula será cancelado en el momento de la matriculación.

-Matrícula: Este proceso inicia con el pago completo de la matrícula en caso de inscripción, o el pago restante luego del abono en caso de pre matrícula. Una vez realizado este pago el proceso continua en secretaria, donde se efectúa la recolección de datos y documentos legales para su matriculación. Entre los documentos solicitados se listan los siguientes:

- Escuela.
 - Libreta de calificaciones del año anterior.
 - Partida de Nacimiento.
 - Dos fotos.
 - Exámenes de Sangre y Coproparasitorio.

- Colegio.
 - Certificado de promoción.
 - Copia a color de la cedula de identidad.
 - Dos fotos.
 - Exámenes de sangre y Coproparasitorio.

Además de estos documentos se requieren datos del representante para la elaboración del acta de compromiso, cuyo modelo está registrado como una plantilla en Word (Anexo 1). Esta acta se imprime y se la hace firmar al representante para su posterior almacenamiento en el libro de matrículas. Los datos solicitados se listan a continuación:

- Fecha de nacimiento.
- Lugar de nacimiento
- Nombre de la escuela que procede
- Numero de cedula
- Nombres del Padres del alumno o apoderado.
- Representante
- Domicilio
- Teléfonos (casa, celular, oficina)
- Ocupación del padre
- Ocupación de la madre
- Servicios (refrigerio, almuerzo, talleres, transporte).

Una vez obtenido todos estos datos, se los registra en una hoja de Excel para la elaboración de listas de grados y cursos que serán entregados a cada profesor al inicio de clases.

-Pago de pensiones y otros rubros: Este proceso se realiza de forma mensual a partir del inicio del año lectivo hasta su finalización. Consiste en el cobro de \$65 correspondientes a la pensión. Además de este cobro, los alumnos tienen la posibilidad de adquirir los servicios de transporte, refrigerio, almuerzo, clases de nivelación, control de tareas, entre otras. Cada uno tiene su respectivo costo y la duración del servicio depende del representante.

El costo de transporte lo determina el transportista según la distancia de la residencia del alumno. Los datos que entrega el transportista para el cobro del servicio correspondiente a cada alumno serán comprobados en colecturía al momento del cobro. El contrato del mismo puede ser medio mes o mes completo y transporte completo o parcial (solo ida o solo regreso). Adicional a los rubros normales, existen gastos adicionales que se cargan al estado de cuenta individual de cada alumno, por ejemplo: derechos de grado, copias, paseos, entre otros.

3.2.2. Definición de requisitos del sistema.

Utilizamos la metodología ERS para los requisitos del sistema.

SISTEMA DE INSCRIPCIONES, PRE MATRICULA, MATRICULA Y PAGO DE PENSIONES

3.2.2.1. Introducción

Este documento es una Especificación de Requisitos de Software para el Sistema de inscripciones, pre matrícula, matrícula y pago de pensiones. Todo su contenido será elaborado con la colaboración de los directivos y personal administrativo de la institución educativa CEBCI.

3.2.2.2. Propósito

Este documento va dirigido a los directivos y personal administrativo del sistema; será el medio de comunicación entre las partes involucradas en el desarrollo del Sistema de Inscripciones, pre matricula, matrícula y pago de pensiones.

3.2.2.3 Ámbito del sistema

El Sistema contempla lo siguiente:

- Automatizar los procesos de inscripciones y pre matriculas.
- Generar reportes y listado de los alumnos matriculados en la institución
- Facilitar el manejo contable de las cuentas individuales de cada alumno.
- Minimizar el tiempo del proceso de matriculación.
- Eliminar la repetición de actividades de recepción de datos.

El objetivo del proyecto es construir un sistema que permita la automatización en el proceso de Inscripciones, pre matricula, matrícula y pago de pensiones.

3.2.2.4 Definiciones

SIPMP	Sistema de Inscripciones, pre matricula, matrícula y pago de pensiones.
INTERFAZ WEB	Es el mecanismo, entorno o herramienta que hace posible la conexión física y funcional para la comunicación entre el usuario y el sistema.
BASE DE DATOS	Es un conjunto de datos almacenados y organizados sistemáticamente en formato digital.
CASOS DE USO	Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir el objetivo específico.

Tabla 3.1: Descripción de definiciones.

3.2.2.5 Responsables del documento

1. Personal administrativo de la institución.
2. Reglamento de matriculación provista por la dirección de educación.

3.2.2.6. Historia de las Versiones del Documento

CUANDO	VERSION	QUE SE CAMBIO
20100728	1.0	Versión original del documento
20100729	1.1	Cambios pequeños en redacción. Se agrega requerimientos computacionales Se agrega el caso de uso

Tabla 3.2: Versiones del documento.

3.2.3. Descripción General

SIPMP (Sistema de Inscripciones, pre matricula, matrícula y pago de pensiones.) se ha planteado bajo la necesidad del personal administrativo para automatizar y optimizar el proceso de inscripciones, pre matriculas, matrículas y pago de pensiones.

SIPMP dispondrá procesos automatizados, los que permitirán la fácil y rápida adaptación por parte del personal que trabaja en la institución educativa.

3.2.3.1. Funciones del Sistema

El espacio de aplicación corresponde a la Institución Educativa CEBCI, su objetivo es la optimización del servicio en las aéreas de colecturía y secretaria en los procesos de inscripciones, pre matricula, matrícula y pago de pensiones, las actividades a realizar es la revisión de documentos, toma de datos del representante, elaboración de las actas de compromiso, emisión de factura de los respectivos rubros.

3.2.3.2 Suposiciones y Dependencias

Se asume que los requisitos descritos en este documento no van a variar luego de que hayan sido aprobados por los líderes del proyecto.

3.2.4 Requisitos Específicos

Esta es la sección más larga e importante del documento por cuanto aquí se especifican detalladamente los requisitos. Esto permite que los diseñadores realicen un sistema que satisfaga completamente los requisitos del personal administrativo y que se planifiquen las pruebas de aceptación para la satisfacción de los directivos.

3.2.4.1. Requisitos Funcionales

Los requisitos funcionales serán agrupados en casos de uso. Estos casos de uso dan una perspectiva del sistema desde el punto de vista de los actores, permiten dar una visión general, los participantes, y el alcance del sistema.

3.2.4.2. Descripción de actores

Representante: Es la persona responsable del alumno.

Colectora: Encargada del cobro del rubro de inscripción, pre matricula, matrícula y pago de pensiones.

Secretaria: Encargada de la recepción de documentos para la matriculación y la elaboración de las listas correspondientes de cada grado o curso.

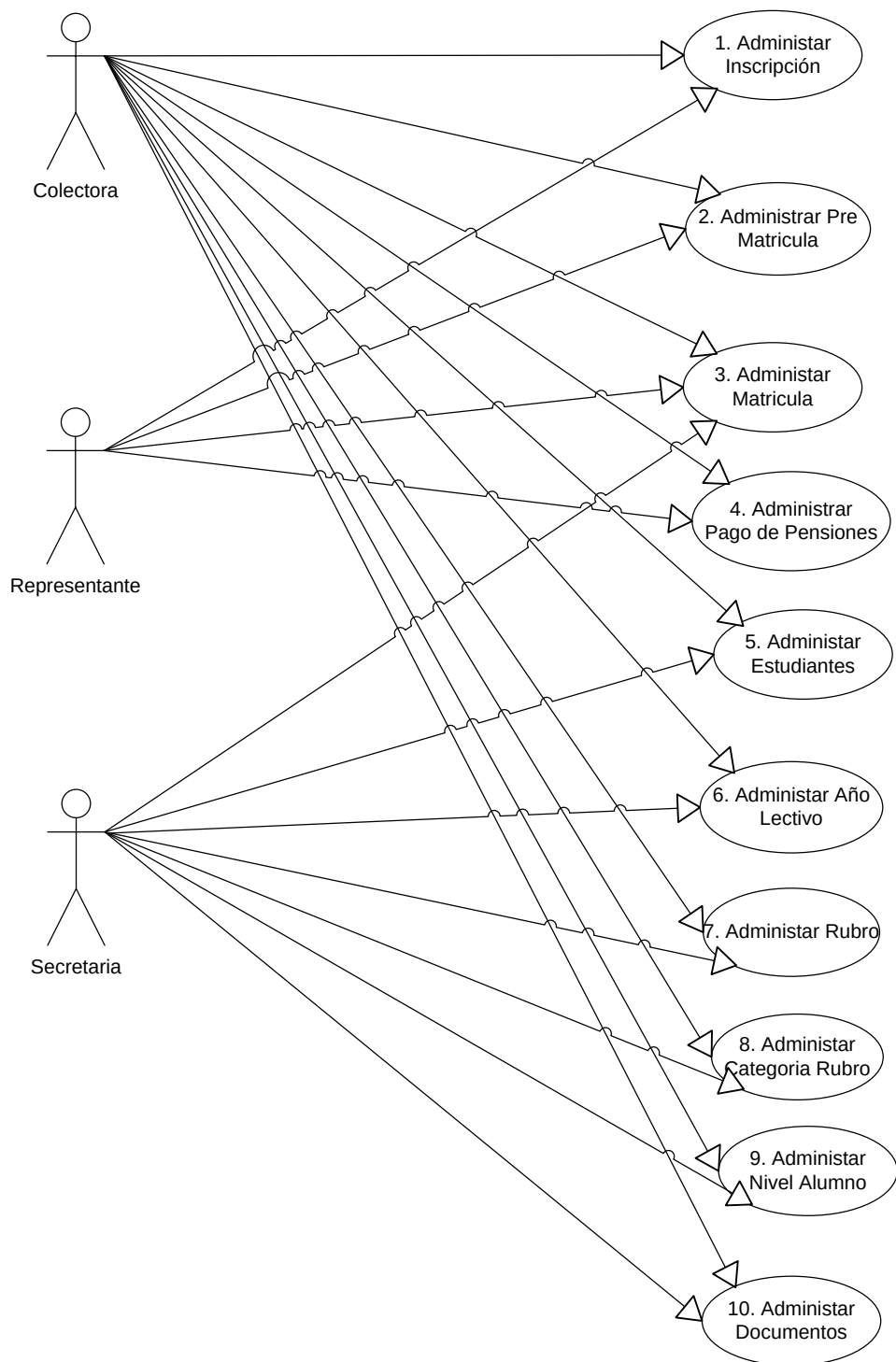


Figura 3.1: Caso de uso General.

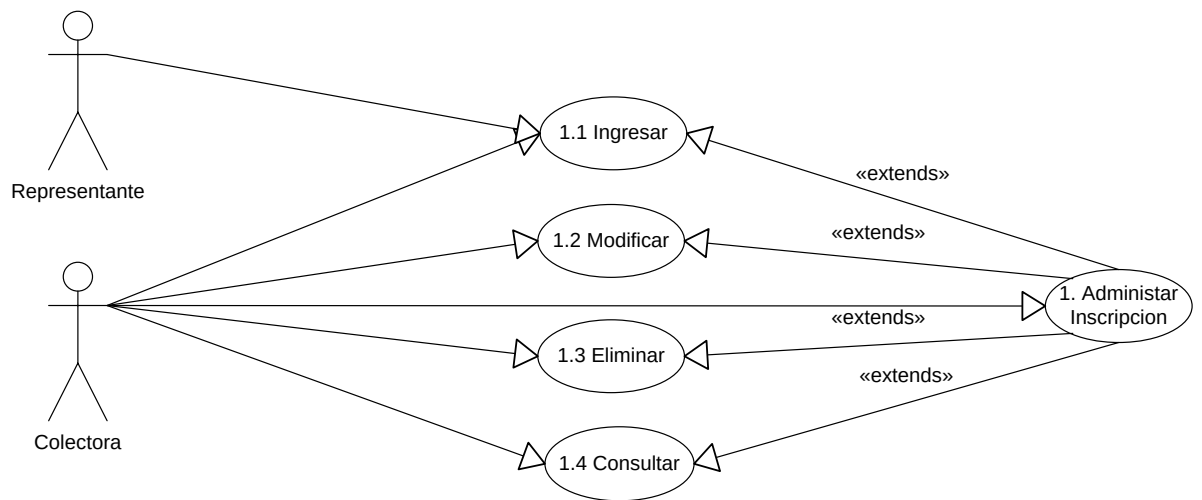


Figura 3.2: Caso de uso Administrar Inscripción.

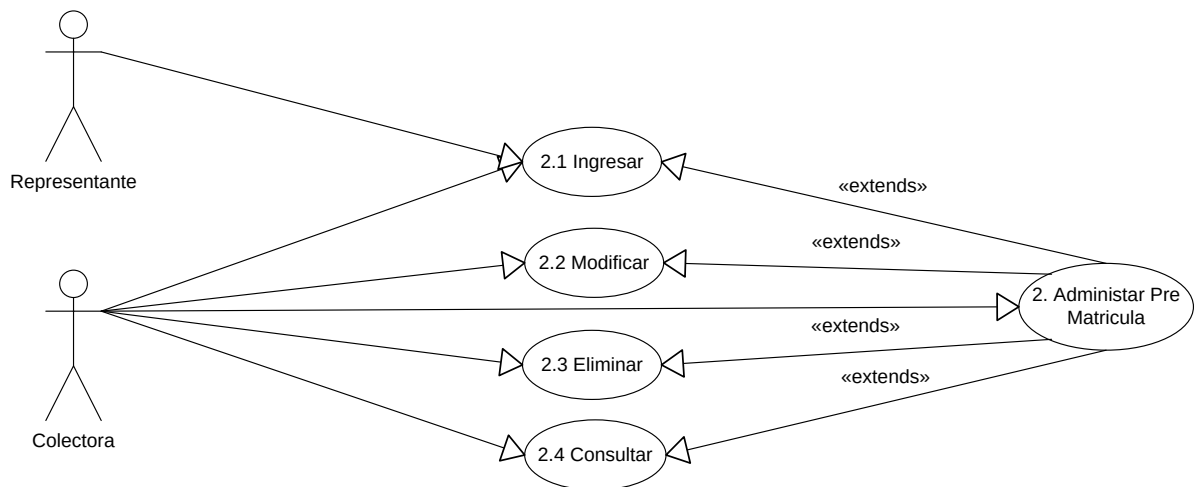


Figura 3.3: Caso de uso Administrar Pre Matrícula.

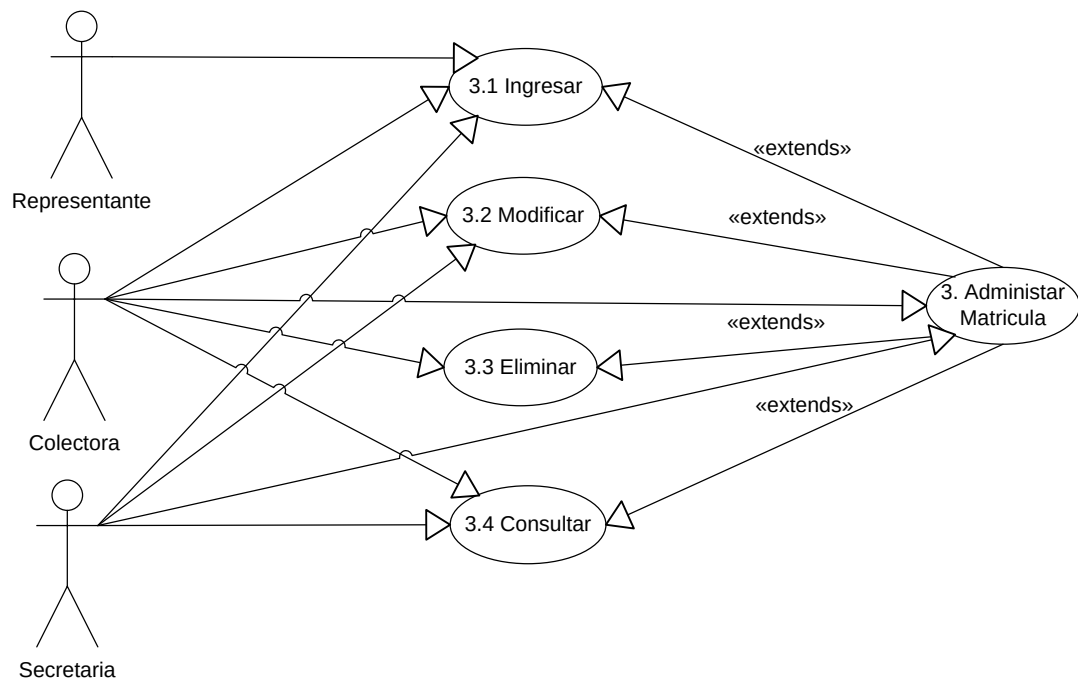


Figura 3.4: Caso de uso Administrar Matrícula.

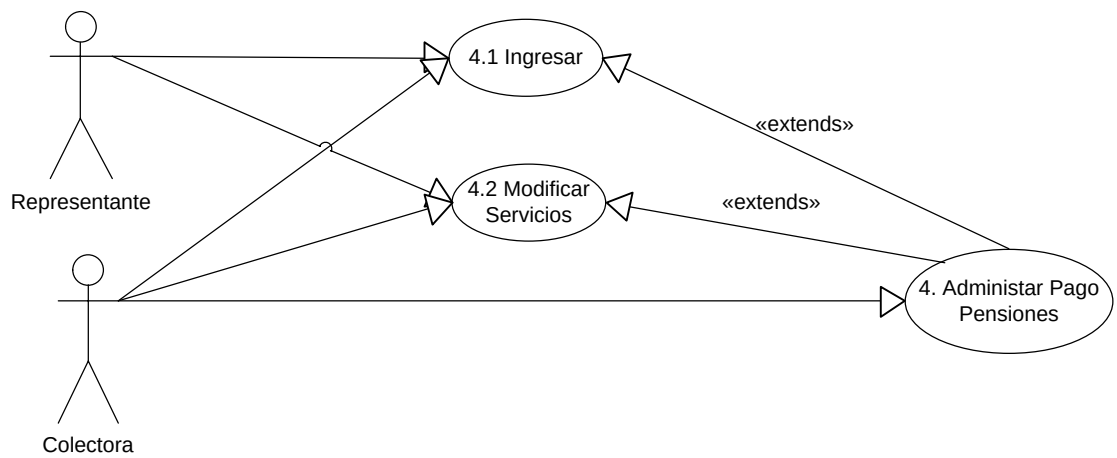


Figura 3.5: Caso de uso Administrar Pago de Pensiones.

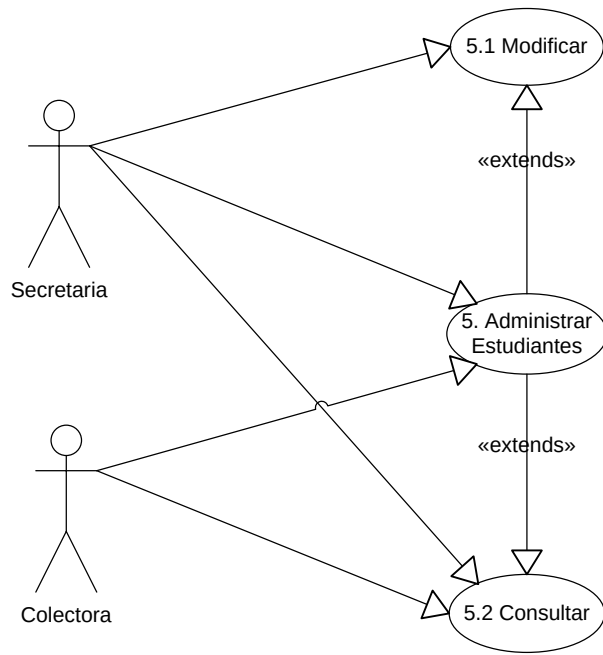


Figura 3.6: Caso de uso Administrar Estudiantes.

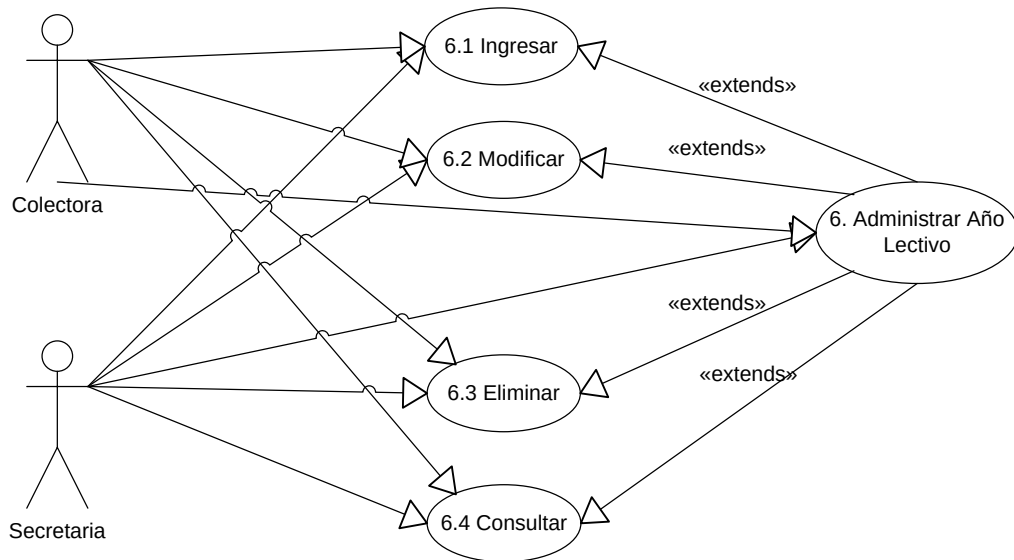


Figura 3.7: Caso de uso Administrar Año Lectivo.

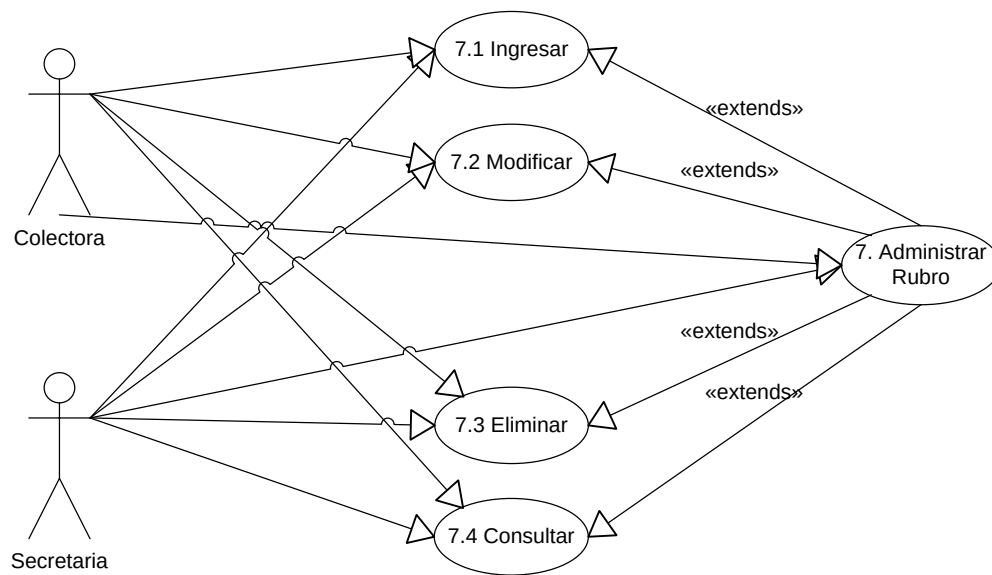


Figura 3.8: Caso de uso Administrar Rubro.

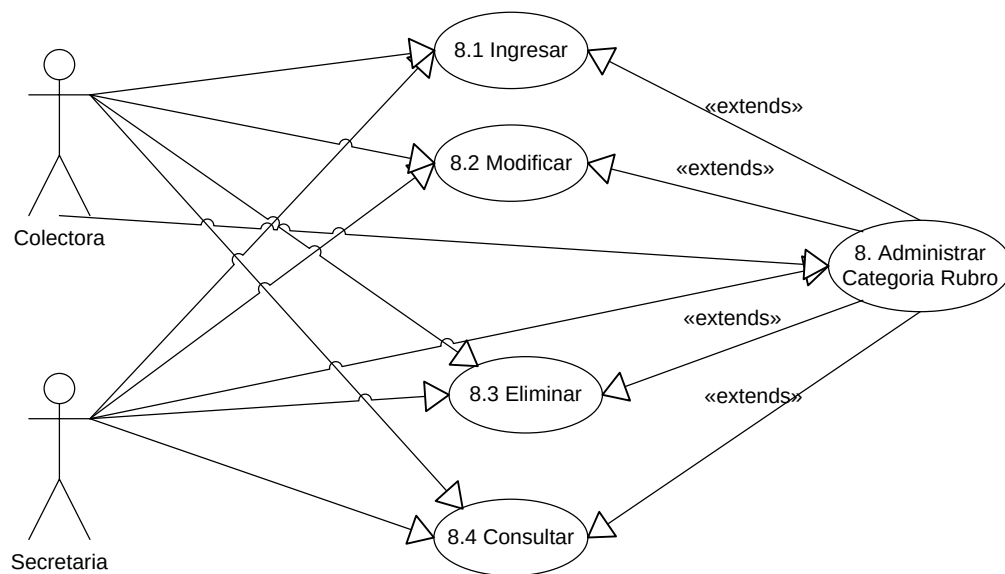


Figura 3.9: Caso de uso Administrar Categoría Rubro.

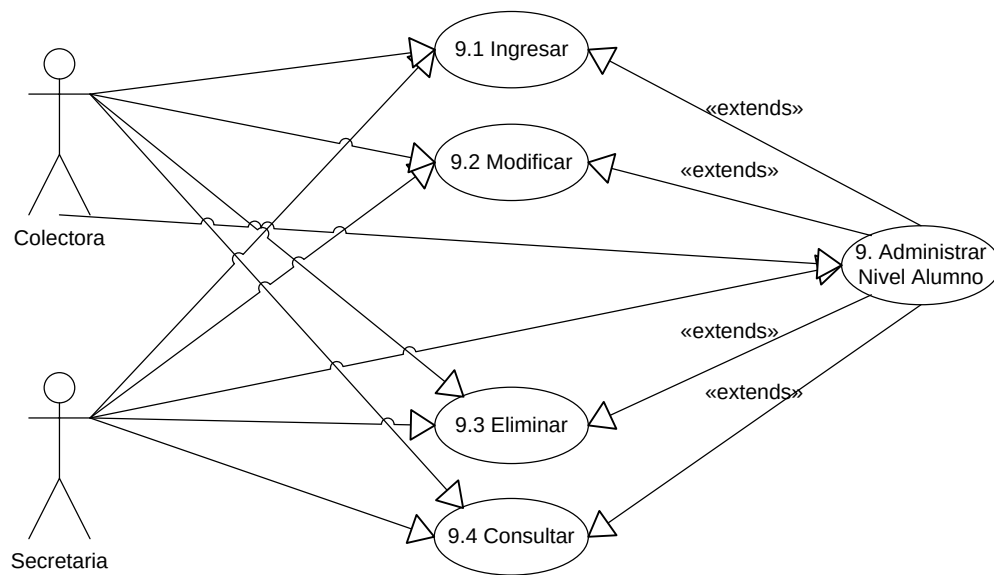


Figura 3.10: Caso de uso Administrar Nivel Alumno.

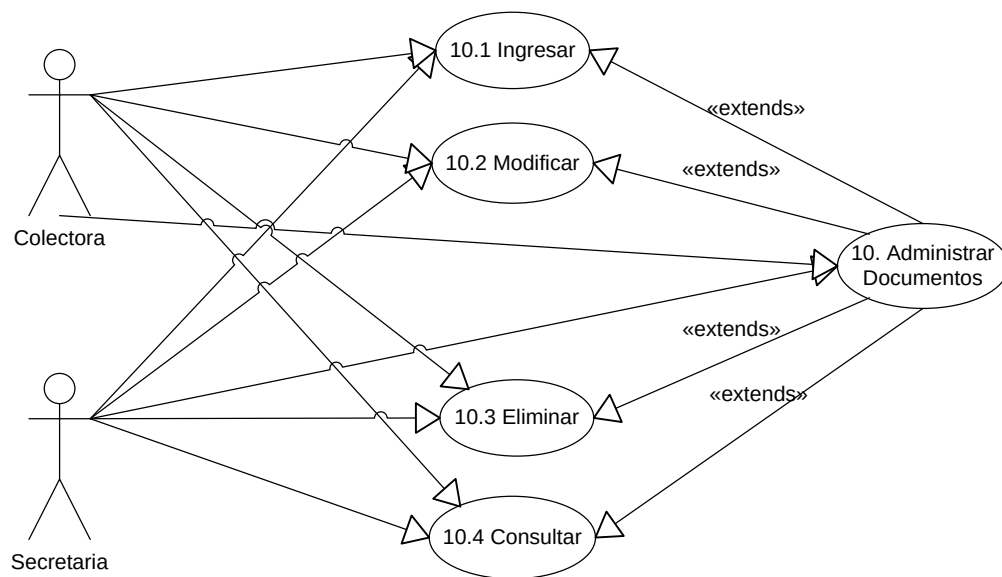


Figura 3.11: Caso de uso Administrar Documentos.

3.2.4.3. Descripción de Casos de Uso

Caso de Uso	3.2.4.3.1 Administrar Inscripción.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar las inscripciones.
Actores	Colectora y Representante.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona la opción de Administrar Inscripción. 3. El sistema presenta el menú con las diferentes opciones para la administración de Inscripciones.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.1.1 Ingresar.
Descripción	Este proceso implica la creación de un nuevo registro del alumno a ingresar en la institución con la finalidad de obtener un cupo.
Actores	Colectora y Representante.
Pre Condiciones	• Que el alumno no pertenezca a la institución.
Post Condiciones	• El estudiante se encuentra registrado en el sistema.
Curso normal de eventos 1. El representante solicita a la colectora la inscripción de su representado. 2. La colectora solicita los datos del representante siendo estos los nombres y números de teléfono del mismo. 3. La colectora ingresa los datos al sistema. 4. La colectora solicita al representante la cancelación de los \$20 correspondientes a este rubro. 5. La colectora procede a la impresión de la factura, la cual es entrega al representante como único comprobante.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.1.2 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos que se encuentren inconsistentes en el registro de Inscripción.
Actores	Colectora.
Pre Condiciones	• Que la inscripción este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona el archivo de Inscripción a modificar. 3. La colectora modifica los datos inconsistentes dentro del archivo. 4. La colectora procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.1.3. Eliminar.
Descripción	Este proceso implica la eliminación de un archivo de inscripción que pertenece al sistema.
Actores	Colectora.
Pre Condiciones	• Que la inscripción este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona el registro de Inscripción a eliminar. 3. La colectora elimina el registro.	
Excepciones El momento que el alumno ya ha realizado la matricula, la inscripción relacionada a dicha matricula no podrá ser eliminada.	
Notas Ninguno	

Caso de Uso	3.2.4.3.1.4 Consultar.
Descripción	Este proceso implica la consulta de un registro de inscripción que pertenezca al sistema.
Actores	Colectora.
Pre Condiciones	• Que la inscripción este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona el archivo de Inscripción a consultar. 3. La colectora accede a la información del archivo seleccionado.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.2 Administrar Pre Matrícula.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar las Pre Matrículas.
Actores	Colectora y Representante.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona la opción de Administrar Pre Matrícula. 3. El sistema presenta el menú con las diferentes opciones para la administración de Pre Matrículas.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.2.1. Ingresar.
Descripción	Este proceso implica la creación de un estado de cuenta individual de cada estudiante. Se realiza un abono de la matrícula con la finalidad de reservar un cupo al siguiente año.
Actores	Colectora y Representante.
Pre Condiciones	• Que el alumno este cursando sus estudios en la institución.
Post Condiciones	• El alumno se encuentra Pre Matriculado.
Curso normal de eventos 1. El representante solicita a la colectora la pre matricula de su representado. 2. La colectora busca los datos del alumno en el sistema. 3. La colectora crea una nueva cuenta individual del alumno con el valor del abono de la matrícula. 4. La colectora solicita al representante un abono del valor total de la matrícula. 5. La colectora procede a la impresión del comprobante de pago.	
Excepciones Si el representante no cancela el pago de un abono mínimo de \$30 del valor total de la matrícula, el proceso se cancela y su representado no podrá ser pre matriculado.	
Notas Ninguno	

Caso de Uso	3.2.4.3.2.2 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos que se encuentren inconsistentes en el registro de Pre Matrícula.
Actores	Colectora.
Pre Condiciones	• Que la Pre Matrícula este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona el archivo de Pre Matrícula a modificar. 3. La colectora modifica los datos inconsistentes dentro del archivo. 4. La colectora procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.2.3 Eliminar.
Descripción	Este proceso implica la eliminación de un archivo de Pre Matrícula que pertenece al sistema.
Actores	Colectora.
Pre Condiciones	• Que la Pre Matrícula esté correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona el registro de Pre Matrícula a eliminar. 3. La colectora elimina el registro. 4. La colectora devuelve el valor del abono que se haya entregado por parte del representante menos el valor del rubro Matricula.	
Excepciones El momento que el alumno ya ha realizado la matrícula, la Pre Matrícula relacionada a dicha Matrícula no podrá ser eliminado.	
Notas Ninguno	

Caso de Uso	3.2.4.3.2.4 Consultar.
Descripción	Este proceso implica la consulta de un registro de Pre Matrícula que pertenezca al sistema.
Actores	Colectora.
Pre Condiciones	• Que la Pre Matrícula este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona el archivo de Pre Matrícula a consultar. 3. La colectora accede a la información del archivo seleccionado.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.3. Administrar Matrícula.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar las Matrículas.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora o secretaria selecciona la opción de Administrar Matrícula. 3. El sistema presenta el menú con las diferentes opciones para la administración de Matrículas.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.3.1 Ingresar.
Descripción	Este proceso implica la creación de una ficha y el acta de compromiso que vincula al estudiante a la institución educativa.
Actores	Colectora, Representante y Secretaria.
Pre Condiciones	• Si es alumno nuevo, haber realizado la Inscripción. • Si es alumno perteneciente a la institución, haber realizado la Pre Matrícula.
Post Condiciones	• El alumno se encuentra matriculado.
Curso normal de eventos 1. El representante solicita a la colectora la matrícula de su representado. 2. La colectora busca los datos del alumno en el sistema verificando las pre condiciones. 3. La colectora solicita al representante el pago del valor total o el saldo pendiente de la matricula dependiendo el caso. 4. El representante solicita a la secretaria la creación de la una nueva ficha de matrícula. 5. La secretaria verifica en el sistema que se haya cancelado el valor de la matrícula en su totalidad. 6. La secretaria pide los datos del representante los cuales son ingresados al sistema como: - Fecha de nacimiento. - Lugar de nacimiento - Nombre de la escuela que procede - Numero de cedula - Nombres del Padres del alumno o apoderado. - Representante - Domicilio	

- Teléfonos (casa, celular, oficina)
- Ocupación del padre
- Ocupación de la madre
- Servicios (refrigerio, almuerzo, talleres, transporte).

7. La secretaria hace la verificación de los documentos solicitados tales como:

Caso escuela:

- Libreta de calificaciones del año anterior.
- Partida de Nacimiento.
- Dos fotos.
- Exámenes de Sangre y Coproparasitorio

Caso colegio:

- Certificado de promoción.
- Copia a color de la cedula de identidad.
- Dos fotos.
- Exámenes de sangre y Coproparasitorio.

8. La secretaria realiza la impresión del acta de compromiso que debe ser firmada por el representante.

9. La secretaria almacena la ficha en el libro de matrículas requerido por la dirección de educación.

Excepciones

Si el alumno no realizó el proceso de Inscripción y existe disponibilidad de cupo este podrá realizar la Matricula siempre cuando realice primero el proceso de Inscripción.

Si el alumno no realizo el proceso de Pre Matrícula y existe disponibilidad de cupo este podrá realizar la Matricula siempre cuando realice primero el proceso de Pre Matrícula.

Notas

Ninguno

Caso de Uso	3.2.4.3.3.2 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos que se encuentren inconsistentes en el registro de Matrícula.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que la Matrícula este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora o secretaria selecciona el registro de Matrícula a modificar. 3. La colectora o secretaria además tiene las opciones de agregar documentos que no fueron entregados al momento de crear la matricula. 4. La colectora o secretaria modifica los datos inconsistentes dentro del registro. 5. La colectora procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.3.3 Eliminar.
Descripción	Este proceso implica la eliminación de un archivo de Matrícula que pertenece al sistema.
Actores	Colectora.
Pre Condiciones	• Que la Matrícula esté correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona el registro de Matrícula a eliminar. 3. La colectora elimina el registro. 4. La colectora procede a la devolución del dinero menos el valor perteneciente al rubro Matrícula.	
Excepciones Los registros de matrícula pueden ser eliminados únicamente mientras no haya empezado el año lectivo.	
Notas Ninguno	

Caso de Uso	3.2.4.3.3.4 Consultar.
Descripción	Este proceso implica la consulta de un registro de Matrícula que pertenezca al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que la Matrícula este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora o secretaria selecciona el registro de Matrícula a consultar. 3. La colectora o secretaria accede a la información del registro seleccionado.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.4. Administrar Pago de Pensiones.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar el Pago de Pensiones.
Actores	Colectora.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora ingresa al sistema. 2. La colectora selecciona la opción de Administrar Pago de Pensiones. 3. El sistema presenta el menú con las diferentes opciones para la administración del Pago de Pensiones.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.4.1 Ingresar.
Descripción	Este proceso implica el pago de los servicios utilizados mensualmente por parte del alumno.
Actores	Colectora y Representante.
Pre Condiciones	• Que el alumno este cursando sus estudios en la institución.
Post Condiciones	Ninguna
Curso normal de eventos 1. El representante solicita a la colectora el pago de la pensión de su representado. 2. La colectora busca el estado de cuenta individual del alumno en el sistema. 3. La colectora selecciona el pago de los rubros que desea cancelar el representante en ese instante. 4. La colectora entrega el comprobante de pago al representante.	
Excepciones El representante tiene la posibilidad de cancelar ciertos rubros pertenecientes a la pensión de un mes determinado. Los rubros restantes se añadirán al estado de cuenta individual del alumno para el siguiente mes. El representante tiene los 10 primeros días del mes para cancelar el valor de la pensión de ese mes, caso contrario se le cobrará una multa.	
Notas Ninguno	

Caso de Uso	3.2.4.3.4.2 Modificar Servicios.
Descripción	Este proceso implica la administración de los servicios de los que hacen uso los alumnos.
Actores	Colectora y Representante.
Pre Condiciones	• Que el alumno este cursando sus estudios en la institución.
Post Condiciones	Ninguna
Curso normal de eventos 6. La colectora ingresa al sistema. 7. La colectora selecciona la hoja de servicios perteneciente al alumno requerido. 8. El representante solicita la activación o cancelación de los servicios que desee. 9. La colectora realiza los cambios solicitados. 10. La colectora procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas • Los servicios pueden ser “Medio Mes” o “Mes Completo”. • Existen servicios especiales se los puede catalogar como “Medio” y “Completo”.	

Caso de Uso	3.2.4.3.5. Administrar Estudiantes.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar el registro de Estudiantes.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora o secretaria selecciona la opción de Administrar Estudiantes. 3. El sistema presenta el menú con las diferentes opciones para la administración de Estudiantes.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.5.1 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos que se encuentren inconsistentes en el registro de Alumnos del sistema.
Actores	Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 1. La secretaria ingresa al sistema. 2. La secretaria selecciona el registro del Alumno a modificar. 3. La colectora modifica los datos inconsistentes dentro del registro. 4. La colectora procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.5.2 Consultar.
Descripción	Este proceso implica la consulta de un registro de Estudiante que pertenezca al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora o secretaria selecciona el registro de Estudiante a consultar. 3. La colectora o secretaria accede a la información del registro seleccionado.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.6. Administrar Año Lectivo.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar el Año Lectivo.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora o secretaria selecciona la opción de Administrar Año Lectivo. 3. El sistema presenta el menú con las diferentes opciones para la Administración de Año Lectivo	
Excepciones	Ninguno
Notas	Ninguno

Caso de Uso	3.2.4.3.6.1 Ingresar.
Descripción	Este proceso implica el ingreso del año lectivo vigente con sus respectivas fechas de inicialización y finalización.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna.
Post Condiciones	Ninguna
Curso normal de eventos 1. La secretaria o colectora selecciona la opción a ingresar nuevo año. 2. La secretaria o colectora ingresa los datos correspondientes al nuevo año lectivo vigente. 3. El sistema visualiza los datos ingresados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.6.2 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos previamente ingresados los cuales que se encuentren inconsistentes
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Año Lectivo estén bien ingresados.
	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora o secretaria selecciona el Año lectivo a modificar. 3. La colectora o secretaria realiza los cambios solicitados. 4. La colectora o secretaria procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.6.3 Eliminar.
Descripción	Este proceso implica la eliminación de un archivo de Año Lectivo que pertenece al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Año Lectivo estén bien ingresados.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora selecciona el registro de Año Lectivo a eliminar. 3. La colectora o secretaria elimina el registro.	
Excepciones Ninguno.	
Notas Ninguno	

Caso de Uso	3.2.4.3.6.4 Consultar.
Descripción	Este proceso implica la consulta de un registro de Año Lectivo que pertenezca al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que el Año Lectivo este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 1. La colectora o secretaria ingresa al sistema. 2. La colectora o secretaria selecciona el archivo de Año Lectivo a consultar. 3. La colectora o secretaria accede a la información del archivo seleccionado.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.7. Administrar Rubro.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar de Rubros que van a estar vigentes en un Año Lectivo.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 4. La colectora o secretaria ingresa al sistema. 5. La colectora o secretaria selecciona la opción de Administrar Rubro. 6. El sistema presenta el menú con las diferentes opciones para la Administración de Rubro.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.7.1 Ingresar.
Descripción	Este proceso implica el ingreso de un nuevo rubro dentro del sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna.
Post Condiciones	Ninguna
Curso normal de eventos 4. La secretaria o colectora selecciona la opción a ingresar nuevo rubro. 5. La secretaria o colectora ingresa los datos correspondientes al nuevo rubro. 6. El sistema visualiza los datos ingresados.	
Excepciones El rubro debe pertenecer a una categoría la cual debe estar previamente creada para su efecto.	
Notas Ninguno	

Caso de Uso	3.2.4.3.7.2 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos previamente ingresados los cuales que se encuentren inconsistentes
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Rubro estén bien ingresados.
	Ninguna
Curso normal de eventos 5. La colectora o secretaria ingresa al sistema. 6. La colectora o secretaria selecciona el Rubro a modificar. 7. La colectora o secretaria realiza los cambios solicitados. 8. La colectora o secretaria procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.7.3 Eliminar.
Descripción	Este proceso implica la eliminación de un archivo de Rubro que pertenece al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Rubro estén bien ingresados.
Post Condiciones	Ninguna
Curso normal de eventos 4. La colectora o secretaria ingresa al sistema. 5. La colectora selecciona el registro de Rubro a eliminar. 6. La colectora o secretaria elimina el registro.	
Excepciones	Ninguno.
Notas	Ninguno

Caso de Uso	3.2.4.3.7.4 Consultar.
Descripción	Este proceso implica la consulta de un registro de Rubro que pertenezca al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que el Rubro este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 4. La colectora o secretaria ingresa al sistema. 5. La colectora o secretaria selecciona el archivo de Rubro a consultar. 6. La colectora o secretaria accede a la información del archivo seleccionado.	
Excepciones	Ninguno
Notas	Ninguno

Caso de Uso	3.2.4.3.8. Administrar Categoría Rubro.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar Categoría de Rubros que van a estar vigentes en un Año Lectivo.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 7. La colectora o secretaria ingresa al sistema. 8. La colectora o secretaria selecciona la opción de Administrar Categoría Rubro. 9. El sistema presenta el menú con las diferentes opciones para la Administración de Categoría Rubro.	
Excepciones	Ninguno
Notas	Ninguno

Caso de Uso	3.2.4.3.8.1 Ingresar.
Descripción	Este proceso implica el ingreso de una nueva categoría rubro dentro del sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna.
Post Condiciones	Ninguna
Curso normal de eventos 7. La secretaria o colectora selecciona la opción a ingresar nueva categoría rubro. 8. La secretaria o colectora ingresa los datos correspondientes a la nueva categoría rubro. 9. El sistema visualiza los datos ingresados.	
Excepciones Ninguno.	
Notas Ninguno	

Caso de Uso	3.2.4.3.8.2 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos previamente ingresados los cuales que se encuentren inconsistentes
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Categoría Rubro estén bien ingresados.
	Ninguna
Curso normal de eventos 9. La colectora o secretaria ingresa al sistema. 10. La colectora o secretaria selecciona la Categoría a modificar. 11. La colectora o secretaria realiza los cambios solicitados. 12. La colectora o secretaria procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.8.3 Eliminar.
Descripción	Este proceso implica la eliminación de un archivo de Categoría Rubro que pertenece al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Categoría Rubro estén bien ingresados.
Post Condiciones	Ninguna
Curso normal de eventos 7. La colectora o secretaria ingresa al sistema. 8. La colectora selecciona el registro de Categoría a eliminar. 9. La colectora o secretaria elimina el registro.	
Excepciones Ninguno.	
Notas Ninguno	

Caso de Uso	3.2.4.3.8.4 Consultar.
Descripción	Este proceso implica la consulta de un registro de Categoría Rubro que pertenezca al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que la Categoría este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 7. La colectora o secretaria ingresa al sistema. 8. La colectora o secretaria selecciona el archivo de Categoría a consultar. 9. La colectora o secretaria accede a la información del archivo seleccionado.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.9. Administrar Nivel Alumno.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar Nivel de alumno que van a estar vigentes en un Año Lectivo.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 10. La colectora o secretaria ingresa al sistema. 11. La colectora o secretaria selecciona la opción de Administrar Nivel Alumno. 12. El sistema presenta el menú con las diferentes opciones para la Administración de Nivel Alumno.	
Excepciones	Ninguno
Notas	Ninguno

Caso de Uso	3.2.4.3.9.1 Ingresar.
Descripción	Este proceso implica el ingreso de un nuevo Nivel dentro del sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna.
Post Condiciones	Ninguna
Curso normal de eventos 10. La secretaria o colectora selecciona la opción a ingresar nuevo Nivel. 11. La secretaria o colectora ingresa los datos correspondientes al nuevo Nivel. 12. El sistema visualiza los datos ingresados.	
Excepciones Ninguno.	
Notas Ninguno	

Caso de Uso	3.2.4.3.9.2 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos previamente ingresados los cuales que se encuentren inconsistentes
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Nivel estén bien ingresados.
	Ninguna
Curso normal de eventos 13. La colectora o secretaria ingresa al sistema. 14. La colectora o secretaria selecciona el Nivel a modificar. 15. La colectora o secretaria realiza los cambios solicitados. 16. La colectora o secretaria procede a guardar los cambios efectuados.	
Excepciones	Ninguno
Notas	Ninguno

Caso de Uso	3.2.4.3.9.3 Eliminar.
Descripción	Este proceso implica la eliminación de un archivo de Nivel que pertenece al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Nivel estén bien ingresados.
Post Condiciones	Ninguna
Curso normal de eventos 10. La colectora o secretaria ingresa al sistema. 11. La colectora selecciona el registro de Nivel a eliminar. 12. La colectora o secretaria elimina el registro.	
Excepciones Ninguno.	
Notas Ninguno	

Caso de Uso	3.2.4.3.9.4 Consultar.
Descripción	Este proceso implica la consulta de un registro de Nivel que pertenezca al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que el Nivel este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 10. La colectora o secretaria ingresa al sistema. 11. La colectora o secretaria selecciona el archivo de Nivel a consultar. 12. La colectora o secretaria accede a la información del archivo seleccionado.	
Excepciones	Ninguno
Notas	Ninguno

Caso de Uso	3.2.4.3.10. Administrar Documentos.
Descripción	Este proceso implica la generación de un menú de opciones que presenta las operaciones para administrar Documentos que van a estar vigentes en un Año Lectivo.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna
Post Condiciones	Ninguna
Curso normal de eventos 13. La colectora o secretaria ingresa al sistema. 14. La colectora o secretaria selecciona la opción de Administrar Documentos. 15. El sistema presenta el menú con las diferentes opciones para la Administración de Documentos.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.10.1 Ingresar.
Descripción	Este proceso implica el ingreso de un nuevo Documento dentro del sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	Ninguna.
Post Condiciones	Ninguna
Curso normal de eventos 13. La secretaria o colectora selecciona la opción a ingresar nuevo Documento. 14. La secretaria o colectora ingresa los datos correspondientes al nuevo Documento. 15. El sistema visualiza los datos ingresados.	
Excepciones Ninguno.	
Notas Ninguno	

Caso de Uso	3.2.4.3.10.2 Modificar.
Descripción	Este proceso implica la modificación de uno o varios datos previamente ingresados los cuales que se encuentren inconsistentes
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Documento estén bien ingresados.
	Ninguna
Curso normal de eventos 17. La colectora o secretaria ingresa al sistema. 18. La colectora o secretaria selecciona el Documento a modificar. 19. La colectora o secretaria realiza los cambios solicitados. 20. La colectora o secretaria procede a guardar los cambios efectuados.	
Excepciones Ninguno	
Notas Ninguno	

Caso de Uso	3.2.4.3.10.3 Eliminar.
Descripción	Este proceso implica la eliminación de un archivo de Documento que pertenece al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que los datos de Documento estén bien ingresados.
Post Condiciones	Ninguna
Curso normal de eventos 13. La colectora o secretaria ingresa al sistema. 14. La colectora selecciona el registro de Documento a eliminar. 15. La colectora o secretaria elimina el registro.	
Excepciones Ninguno.	
Notas Ninguno	

Caso de Uso	3.2.4.3.10.4 Consultar.
Descripción	Este proceso implica la consulta de un registro de Documento que pertenezca al sistema.
Actores	Colectora y Secretaria.
Pre Condiciones	• Que el Documento este correctamente ingresada al sistema.
Post Condiciones	Ninguna
Curso normal de eventos 13. La colectora o secretaria ingresa al sistema. 14. La colectora o secretaria selecciona el archivo de Documento a consultar. 15. La colectora o secretaria accede a la información del archivo seleccionado.	
Excepciones Ninguno	
Notas Ninguno	

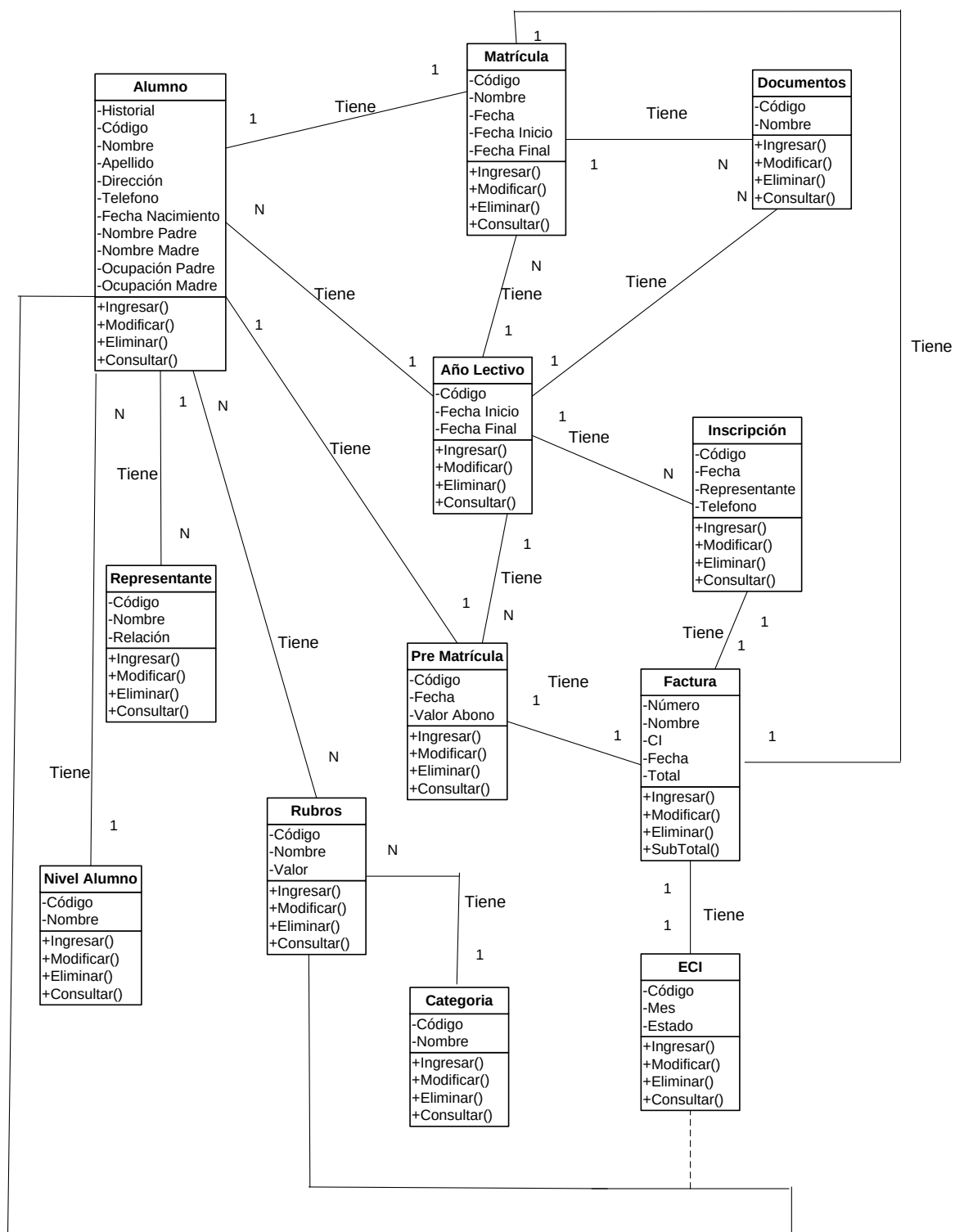
3.2.5. Interfaces de usuario

El sistema ofrecer una interfaz de usuario intuitivo, fácil de aprender y sencillo de manejar. Se pretende que el usuario se familiarice con el sistema en cuestión de pocas horas.

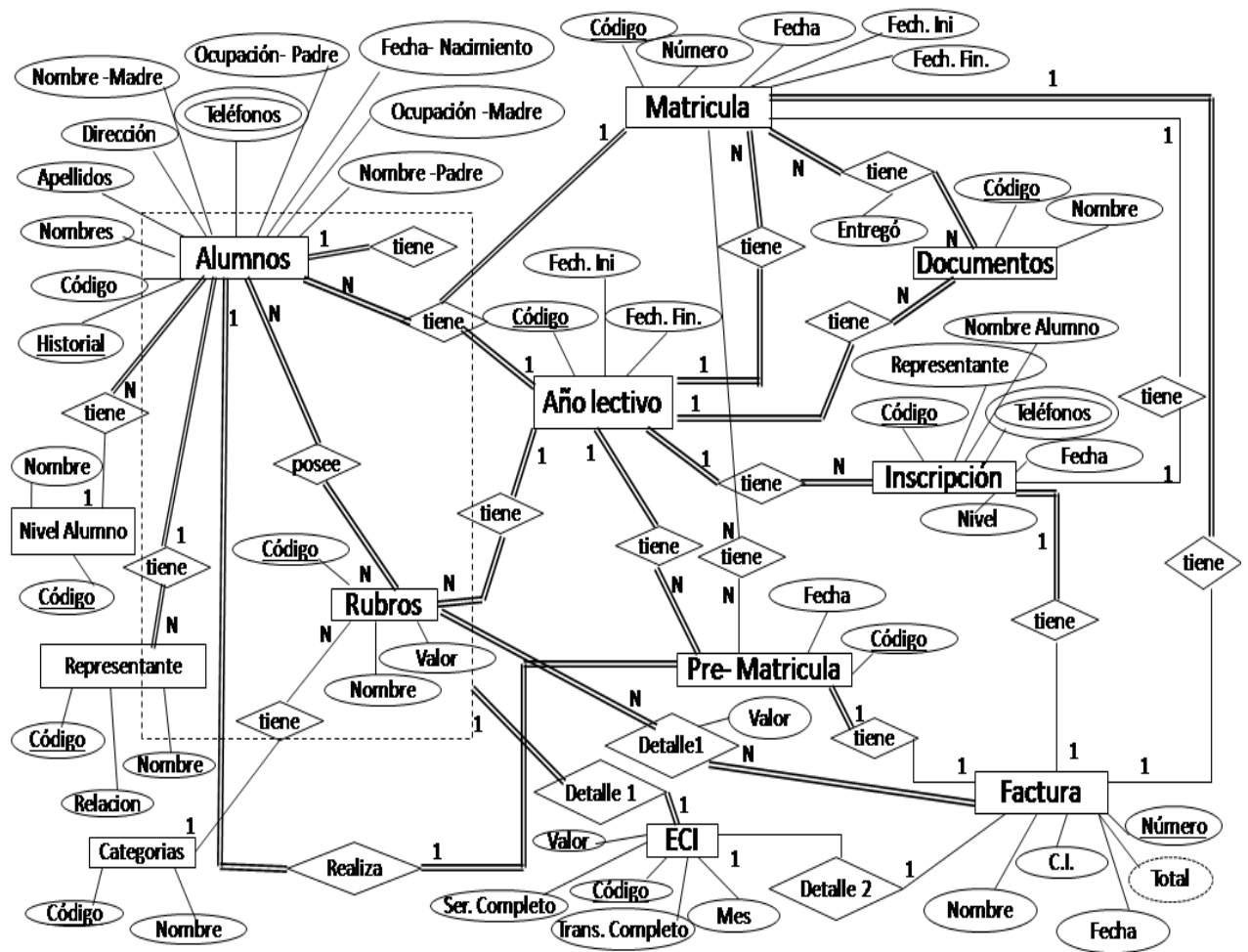
Las interfaces de usuario deben ser orientadas a la Web y seguirán los estándares planteados por los desarrolladores.

3.3. Modelado de Análisis.

3.3.1. Modelo de Clases.

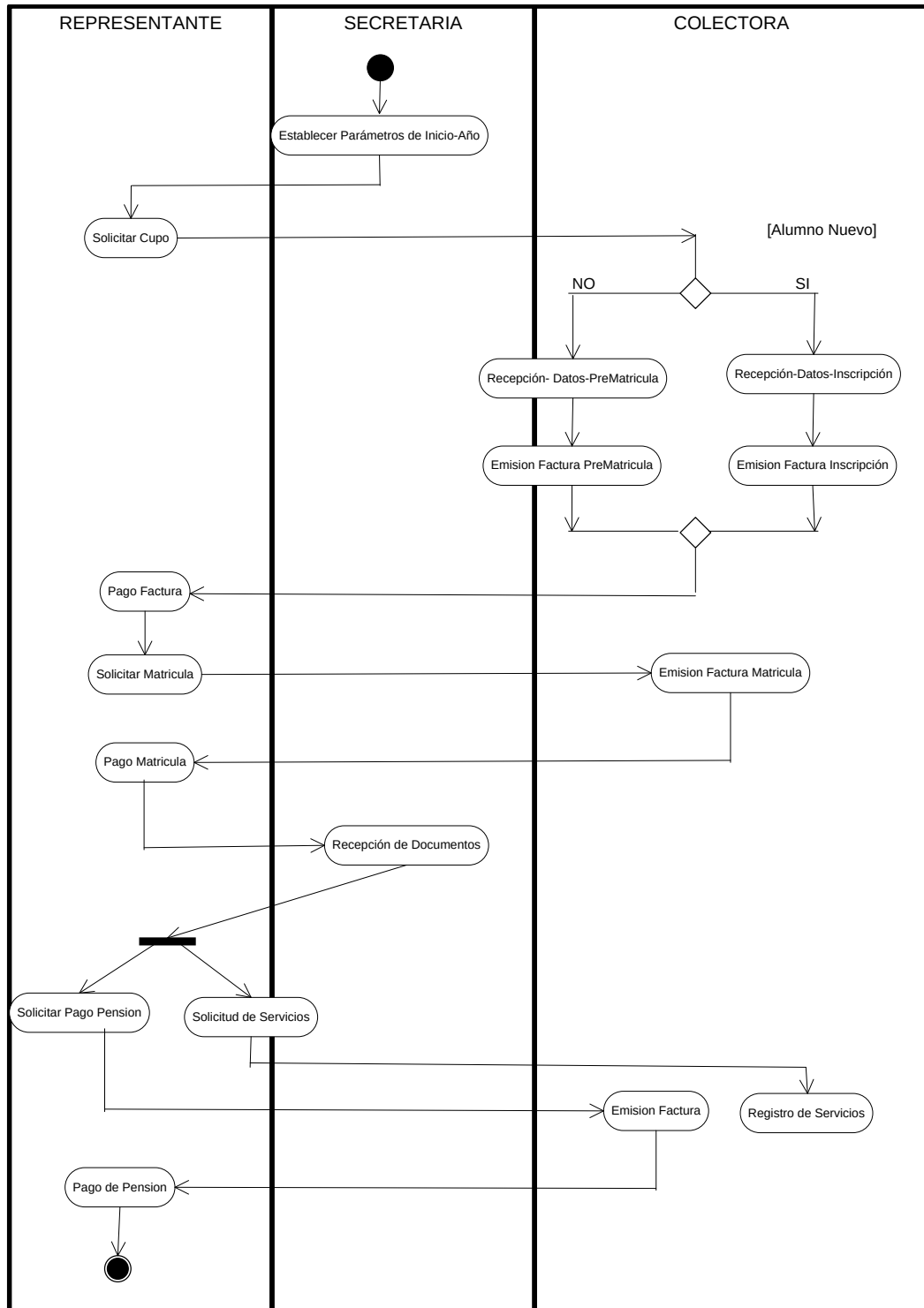


3.3.2. Modelo Objeto-Relación.



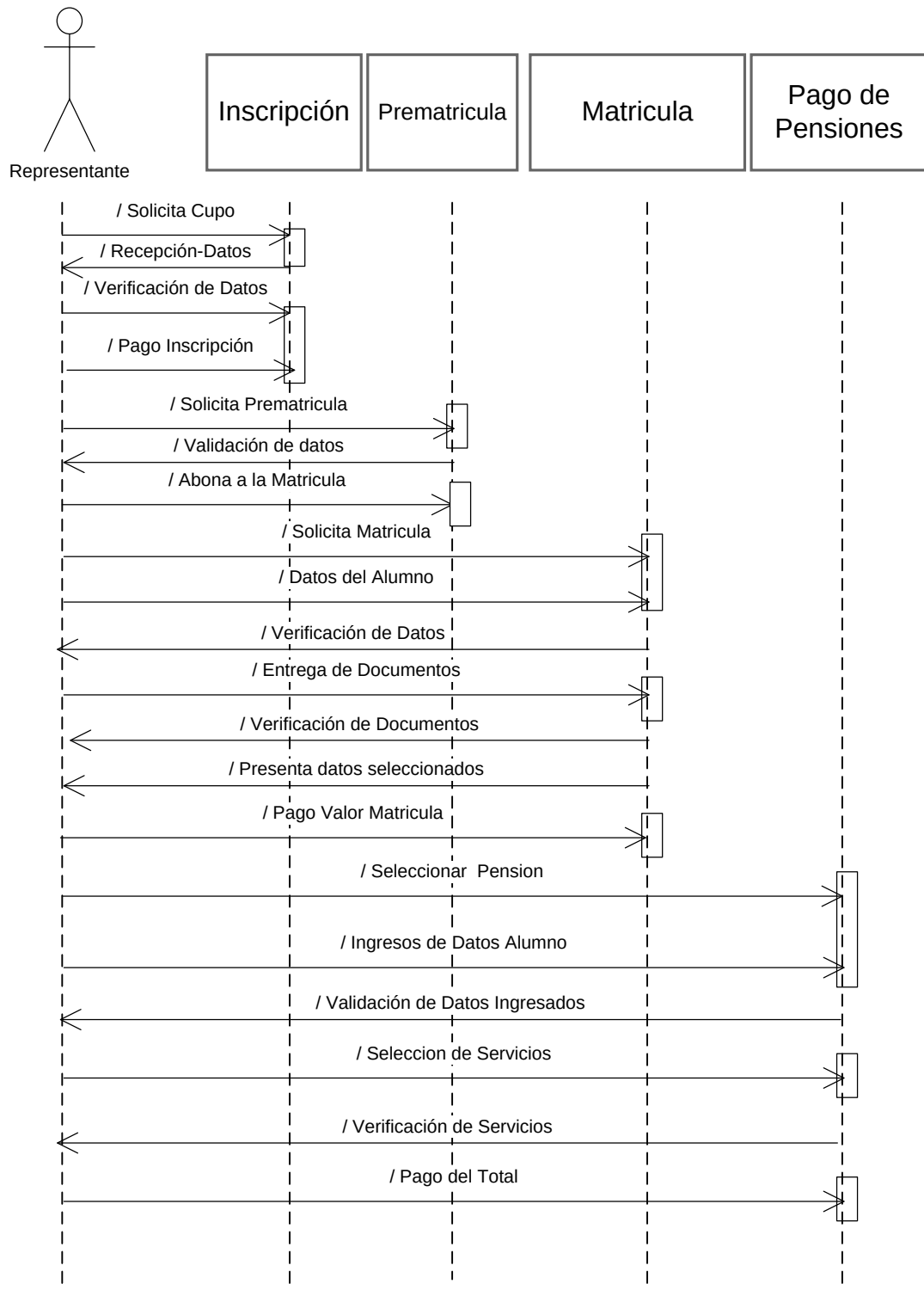
3.3.3. Modelo de Comportamiento.

3.3.3.1. Modelo de Actividad.

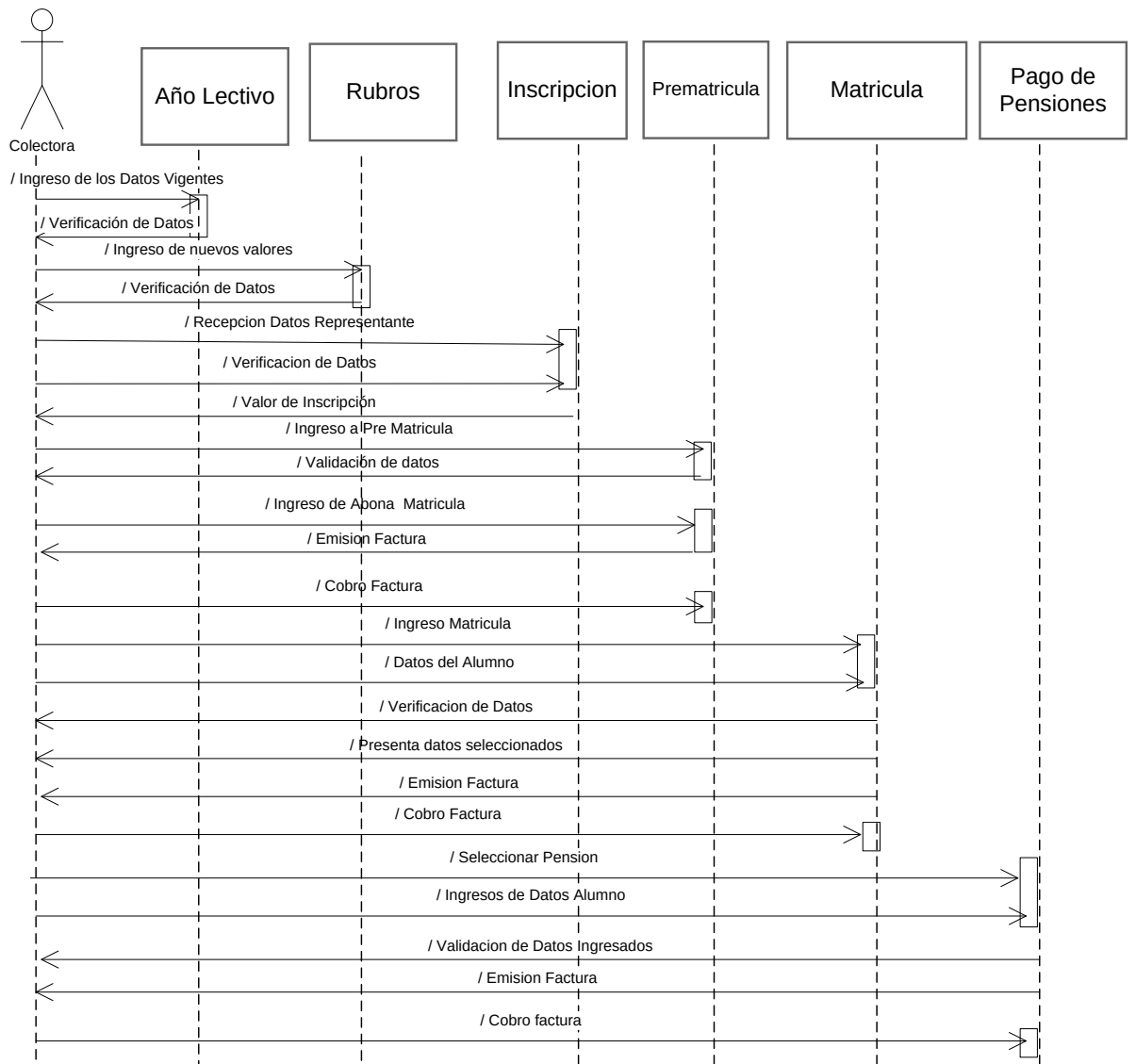


3.3.3.2. Modelo de Secuencia.

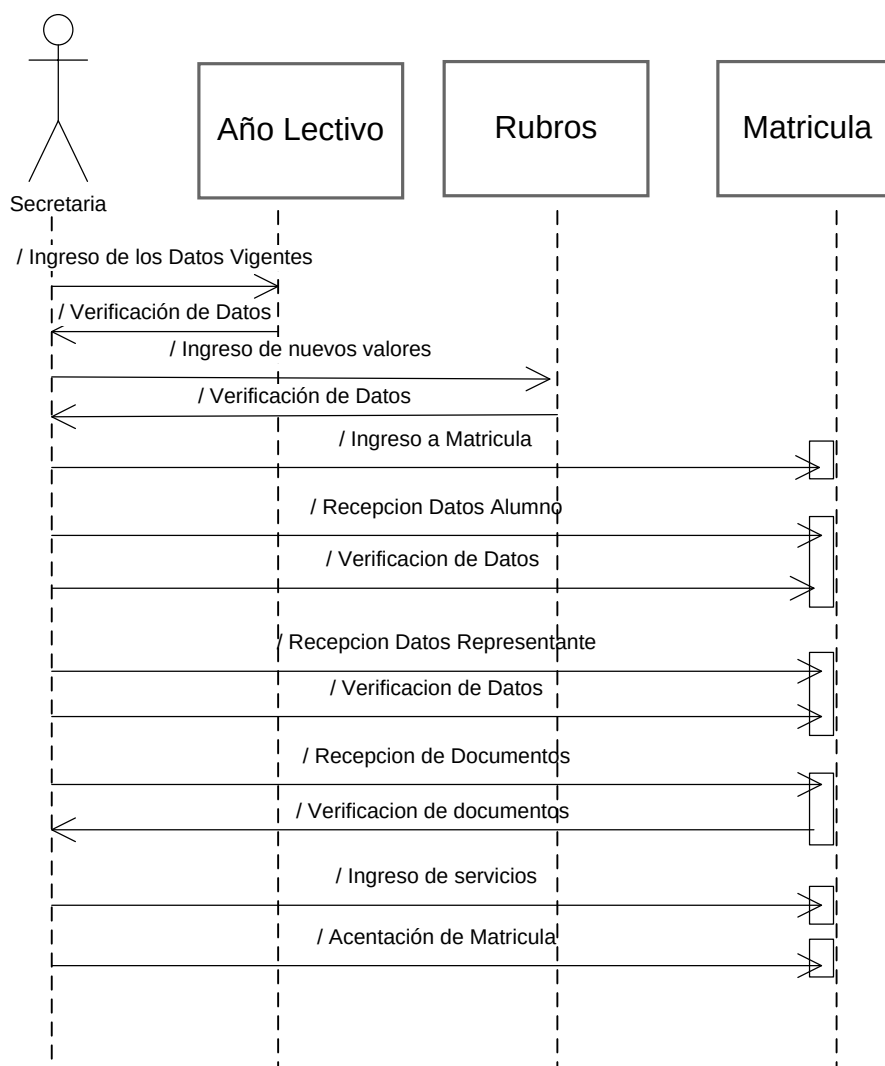
Representante



Colectora



Secretaría



3.4 Modelado de Diseño.

3.4.1 Diseño de Objetos.

3.4.1.1. Diccionario de Datos

telefono_inscripcion

Campo	Tipo	Nulo	Predeterminado	Comentarios
insc_codigo	int(11)	No	0	Llave foranea de la tabla "inscripcion".
tele_insc_numero	varchar(10)	No		Número telefónico.

telefono_alumno

Campo	Tipo	Nulo	Predeterminado	Comentarios
alum_historial	int(11)	No	0	Llave foranea de la tabla "alumno".
tele_alum_numero	varchar(10)	No	0	Número telefónico.

rubro_factura

Comentarios de la tabla: Almacena el detalle de la factura

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>rubr_codigo</u>	int(11)	No		Llave foranea de la tabla "rubro"
<u>fact_numero</u>	int(11)	No		Llave foranea de la tabla "factura"
rubr_fact_valor	double(10,2)	Sí	NULL	Es campo determina el valor de abono para la prematricula

rubro_alumno

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>rubr_codigo</u>	int(11)	No	0	Llave foranea de la tabla "rubro"
<u>alum_historial</u>	int(11)	No	0	Llave foranea de la tabla "alumno".

rubro

Comentarios de la tabla: Registra los rubros de la institucion y otros.

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>rubr_codigo</u>	int(11)	No		Identificador de cada rubro
rubr_nombre	varchar(50)	No		Nombre del rubro.
rubr_valor	double(10,2)	No	0.00	Costo del rubro.
anio_rubr	varchar(10)	No		Llave foranea de "anio_lectivo"
cate_rubr	int(11)	No	0	Llave foranea de "categoria"

representante

Comentarios de la tabla: Tabla que almacena el nombre del representante con la relaci

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>repr_codigo</u>	int(11)	No		Llave principal de la tabla "Representante"
repr_nombre	varchar(50)	No		Campo que almacena el nombre de la persona que representa al alumno
repr_relacion	varchar(50)	No		Campo que almacena la relacion que tiene el representante con el alumno
alum_repr	int(11)	No		Llave foranea de la tabla "Alumno"

prematricula_matricula

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>prem_codigo</u>	int(11)	No	0	Llave foranea de la tabla "prematricula".
<u>matr_codigo</u>	int(11)	No	0	Llave foranea de la tabla "matricula".

prematricula

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>prem_codigo</u>	int(11)	No		Llave principal de la tabla "prematricula".
prem_fecha	date	No		Fecha en la que se realizó la prematricula.
anio_prem	varchar(10)	No		Llave foranea de la tabla "anio_lectivo".
fact_prem	int(11)	No	0	Llave foranea de la tabla "factura".
alum_prem	int(11)	No	0	Llave foranea de la tabla "alumno".

nivel

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>nivel_codigo</u>	int(11)	No		Llave primaria de la tabla "nivel".
nivel_descripcion	varchar(50)	No		Descripcion o nombre del nivel.

matricula

Comentarios de la tabla: Registro de matrículas.

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>matr_codigo</u>	int(11)	No		Llave primaria de la tabla "matriculas"
matr_numero	int(11)	Sí	NULL	Numero de la matrícula emitida.
matr_fecha	date	Sí	NULL	Fecha en la que se matriculó el alumno.
matr_fecha_inicio	date	Sí	NULL	Fecha de inicio del alumno en casos especiales.
matr_fecha_fin	date	Sí	NULL	Fecha de salida del alumno en casos especiales.
anio_matr	varchar(10)	No		Llave foranea de la tabla "anio_lectivo".
fact_matr	int(11)	No	0	Llave foranea de la tabla "factura"

inscripcion_matricula

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>insc_codigo</u>	int(11)	No	0	Llave foranea de la tabla "inscripcion".
<u>matr_codigo</u>	int(11)	No	0	Llave foranea de la tabla "matricula".

inscripcion

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>insc_codigo</u>	int(11)	No		Codigo de la tabla "inscripcion"
insc_representante	varchar(50)	No		Nombre de la persona que inscribió al alumno.
insc_fecha	date	No		Fecha en la que se realizó la inscripcion.
insc_nivel	varchar(50)	No		Nivel al que aspira ingresar el alumno.
insc_nombre_alumno	varchar(50)	No		Nombre del alumno inscrito.
anio_insc	varchar(10)	No		Llave foranea de "anio_lectivo"
fact_insc	int(11)	No	0	Llave foranea de "factura"

factura

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>fact_numero</u>	int(11)	No		Numero de la factura.
fact_Cl	varchar(10)	No		Cedula de Identidad del cliente.
fact_nombre	varchar(50)	Sí	NULL	Nombre del cliente.
fact_fecha	date	Sí	NULL	Fecha de la factura.
fact_total	double(10,2)	Sí	NULL	Total de la factura.

eci_factura

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>fact_numero</u>	int(11)	No	0	Llave foranea de la tabla "factura".
<u>eci_codigo</u>	int(11)	No	0	Llave foranea de la tabla "eci".

eci

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>eci_codigo</u>	int(11)	No		Llave primaria de la tabla "eci".
eci_mes	varchar(15)	No		Describe el mes correspondiente al eci.
eci_valor	double(10,2)	Sí	NULL	Campo que determina el valor de los rubros-alumno
eci_servicio_completo	tinyint(1)	Sí	NULL	Campo que almacena si ha aceptado este servicio el alumno
eci_transporte_completo	tinyint(1)	Sí	NULL	Campo que almacena la aceptacion de este servicio por parte del alumno
rubr_codigo	int(11)	No	0	Llave foranea de la tabla "rubro_alumno"
alum_historial	int(11)	No	0	Llave foranea de la tabla "rubro_alumno"

documento_matricula

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>docu_codigo</u>	int(11)	No	0	Llave foranea de la tabla "documento" que conforma la llave.
<u>matr_codigo</u>	int(11)	No	0	Llave foranea de la tabla "matricula" que conforma la llave
docu_matr_entrego	varchar(1)	No	0	Campo que registra si se ha entregando o no el documento.

documentos

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>docu_codigo</u>	int(11)	No		Llave primaria de la tabla.
docu_nombre	varchar(100)	No		Nombre del documento.
anio_docu	varchar(10)	No		Llave foranea de la tabla "anio_lectivo".

categoria

Comentarios de la tabla: Categoría de los rubros

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>cate_codigo</u>	int(11)	No		Codigo de la categoría
cate_nombre	varchar(50)	No		Nombre de la categoría.

anio_lectivo

Campo	Tipo	Nulo	Predeterminado	Comentarios
<u>anio_codigo</u>	varchar(10)	No		Codigo del año lectivo. Ejem: "2009-2010"
anio_fecha_inicio	date	No		Fecha prevista para inicio de actividades estudiantiles.
anio_fecha_fin	date	No		Fecha prevista para final de actividades estudiantiles.

alumno

Comentarios de la tabla: Registra los datos del alumno.

Campo	Tipo	Nulo	Predeterminado	Comentarios
alum_historial	int(11)	No		Llave primaria de la tabla "alumnos"
alum_codigo	int(11)	No	0	Codigo del alumno dentro de la institución.
alum_nombre	varchar(50)	No		Nombre del alumno.
alum_apellido	varchar(50)	No		Apellido del alumno
alum_direccion	varchar(50)	Sí	NULL	Dirección del alumno.
alum_nombre_padre	varchar(50)	Sí	NULL	Nombre y apellido del padre del alumno.
alum_nombre_madre	varchar(50)	Sí	NULL	Nombre y apellido de la madre del alumno.
alum_ocupacion_padre	varchar(30)	Sí	NULL	Ocupacion del padre del alumno.
alum_ocupacion_madre	varchar(30)	Sí	NULL	Ocupación de la madre del alumno
alum_fecha_nacimiento	date	Sí	NULL	Fecha de nacimiento del alumno.
alum_nivel	int(11)	Sí	NULL	Nivel en el que se encuentra el alumno.
anio_alum	varchar(10)	Sí	NULL	Llave foranea de la tabla "Anio_lectivo"
matr_alum	int(11)	No		Llave foranea de la tabla "Matricula"

3.4.1.2 Seudocódigo.

Generada por la tabla ECI para el pago de pensiones

Funcion `listar_eci (alum_historial, rubr_codigo)` como `Listadeeci`

Variable `lista_eci` como `Lista(Deeci)`

Variable `comando` Como `ComandoMySQL`

Variable `resultado` Como `LectorDeDatosMySQL`

Variable `query` Como `Cadena`

Variable `mes_inicio, mes_fin` Como `Cadena`

Variable `i, mes` Como `Entero`

Variable `rubr_valor` Como `Cadena`

Variable `lista_anio()` Como `Cadena` = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"}

`conectar()`

`query = "SELECT rubr_valor FROM rubro WHERE rubr_codigo = 'rubr_codigo' "`

`comando = NuevoComandoMySQL(query, Conexion)`

`resultado = comando.RealizarConsulta`

`Leer resultado`

`rubr_valor = resultado("rubr_valor")`

Cerrar resultado

```
query = "SELECT anio_fecha_inicio, anio_fecha_fin FROM anio_lectivo, alumno WHERE  
anio_codigo = anio_alum and alum_historial = 'alum_historial '"
```

```
comando = NuevoComandoMySQL(query, Conexion)
```

```
resultado = comando.RealizarConsulta
```

Leer resultado

```
mes_inicio = resultado("anio_fecha_inicio").SubCadena(3, 2)
```

```
mes_fin = resultado("anio_fecha_fin").SubCadena(3, 2)
```

Cerrar resultado

```
Para i = Convert.ToInt16(mes_inicio) hasta mes_fin + 12
```

```
Si i > 12 Entonces
```

```
mes = i - 13
```

```
Caso contrario
```

```
mes = i - 1
```

```
FinSi
```

```
Variable objeto_eci Como Nuevo eci
```

```
objeto_eci.propiedad_eci_habilitado = Verdadero
```

```
objeto_eci.propiedad_eci_mes = lista_anio(mes)
```

```
objeto_eci.propiedad_eci_mes_contratado = Falso
```

```
objeto_eci.propiedad_eci_valor = rubr_valor
```

```
objeto_eci.propiedad_eci_servicio_completo = Verdadero
```

```
objeto_eci.propiedad_eci_transporte_completo = Verdadero
```

```
lista_eci.Add(objeto_eci)
```

```
Siguiente i
```

Intentar

'Hago la consulta para que me regrese todos los eci que YA han sido cancelados

```
query = "SELECT * FROM eci WHERE alum_historial = 'alum_historial ' and rubr_codigo =  
'rubr_codigo ' and eci_codigo in (SELECT eci_codigo FROM eci_factura)"
```

```
comando = NuevoComandoMySQL(query, Conexion)
```

```
resultado = comando.RealizarConsulta
```


'Ahora les ingreso al arreglo pero con la propiedad "eci_habilitado" puesta a "FALSO" pues no deben habilitarse a modificación al estar cancelados ya.

HacerMientrasresultado.Read()

Para i = 0 To Contar (lista_ec) - 1

Si lista_ec(i).propiedad_eci_mes = Convert.ToCadena(resultado("eci_mes")) Entonces

lista_ec(i).propiedad_eci_habilitado = Falso

lista_ec(i).propiedad_eci_codigo = Convert.ToCadena(resultado("eci_codigo"))

lista_ec(i).propiedad_eci_mes = Convert.ToCadena(resultado("eci_mes"))

lista_ec(i).propiedad_eci_mes_contratado = Verdadero

lista_ec(i).propiedad_eci_valor = Convert.ToCadena(resultado("eci_valor"))

lista_ec(i).propiedad_eci_servicio_completo =

Convert.ToCadena(resultado("eci_servicio_completo"))

lista_ec(i).propiedad_eci_transporte_completo =

Convert.ToCadena(resultado("eci_transporte_completo"))

i = Contar (lista_ec)

FinSi

Siguiente i

Repetir

Cerrar resultado

'Hago lo mismo pero ahora consulto para que me regrese todos los eci que NO han sido cancelados aun

query = Cadena.Paramato("SELECT * FROM eci WHERE alum_historial = '{0}' and
rubr_codigo = '{1}' and eci_codigo NOT IN (SELECT eci_codigo FROM eci_factura)",
alum_historial, rubr_codigo)

comando = NuevoComandoMySQL(query, Conexion)

resultado = comando.RealizarConsulta

'Ahora les ingreso al arreglo pero con la propiedad "eci_habilitado" puesta a
"VERDADERO" pues deben habilitarse a modificación al no estar cancelados aun.

HacerMientrasresultado.Read()

Para i = 0 To Contar (lista_ec) - 1

Si lista_ec(i).propiedad_eci_mes = resultado("eci_mes"))Entonces

```

lista_eci(i).propiedad_eci_habilitado = Verdadero
lista_eci(i).propiedad_eci_codigo = resultado("eci_codigo")
lista_eci(i).propiedad_eci_mes = resultado("eci_mes")
lista_eci(i).propiedad_eci_mes_contratado = Verdadero
lista_eci(i).propiedad_eci_valor = resultado("eci_valor")
lista_eci(i).propiedad_eci_servicio_completo = resultado("eci_servicio_completo")
lista_eci(i).propiedad_eci_transporte_completo = resultado("eci_transporte_completo")

```

```

i = Contar (lista_ec)

```

FinSi

Siguiente i

Repetir

Cerrar resultado

desconectar()

Capturar ex ComoExcepcion

'Hago la consulta para que me regrese todos los eci que YA han sido cancelados

conectar_alterno()

```

query = Cadena.Paramato("SELECT * FROM eci WHERE alum_historial = '{0}' and
rubr_codigo = '{1}' and eci_codigo in (SELECT eci_codigo FROM eci_factura)",
alum_historial, rubr_codigo)

```

```

comando = NuevoComandoMySQL(query, Conexion_alterno)

```

```

resultado = comando.RealizarConsulta

```

'Ahora les ingreso al arreglo pero con la propiedad "eci_habilitado" puesta a "FALSO" pues no deben habilitarse a modificación al estar cancelados ya.

```

HacerMientrasresultado.Read()

```

```

Para i = 0 To Contar (lista_ec) - 1

```

```

Silista_eci(i).propiedad_eci_mes = resultado("eci_mes") Entonces

```

```

lista_eci(i).propiedad_eci_habilitado = Falso

```

```

lista_eci(i).propiedad_eci_codigo = resultado("eci_codigo")

```

```

lista_eci(i).propiedad_eci_mes = resultado("eci_mes")

```

```

lista_eci(i).propiedad_eci_valor = resultado("eci_valor")

```

```

Siresultado("eci_servicio_completo") = "1"Entonces
lista_eci(i).propiedad_eci_servicio_completo = Verdadero
Caso contrario
lista_eci(i).propiedad_eci_servicio_completo = Falso
FinSi
Siresultado("eci_transporte_completo") = "1"Entonces
lista_eci(i).propiedad_eci_transporte_completo = Verdadero
Caso contrario
lista_eci(i).propiedad_eci_transporte_completo = Falso
FinSi
i = Contar (lista_ec)
FinSi
Siguiente i
Repetir
Cerrar resultado

```

'Hago lo mismo pero ahora consulto para que me regrese todos los eci que NO han sido cancelados aun

```

query = "SELECT * FROM eci WHERE alum_historial = 'alum_historial ' and rubr_codigo = 'rubr_codigo ' and eci_codigo NOT IN (SELECT eci_codigo FROM eci_factura)"
comando = NuevoComandoMySQL(query, Conexion_alterno)
resultado = comando.RealizarConsulta

```

'Ahora les ingreso al arreglo pero con la propiedad "eci_habilitado" puesta a "VERDADERO" pues debe habilitarse a modificación.

```

HacerMientrasresultado.Read()
Para i = 0 To Contar (lista_ec) - 1
Si lista_eci(i).propiedad_eci_mes = resultado("eci_mes")) Entonces
lista_eci(i).propiedad_eci_habilitado = Verdadero
lista_eci(i).propiedad_eci_codigo = resultado("eci_codigo")
lista_eci(i).propiedad_eci_mes = resultado("eci_mes")
lista_eci(i).propiedad_eci_valor = resultado("eci_valor")
Siresultado("eci_servicio_completo") = "1"Entonces
lista_eci(i).propiedad_eci_servicio_completo = Verdadero

```

Caso contrario

lista_eci(i).propiedad_eci_servicio_completo = Falso

FinSi

Siresultado("eci_transporte_completo") = "1"Entonces

lista_eci(i).propiedad_eci_transporte_completo = Verdadero

Caso contrario

lista_eci(i).propiedad_eci_transporte_completo = Falso

FinSi

i = Contar (lista_ec)

FinSi

Siguiente i

Repetir

Cerrar resultado

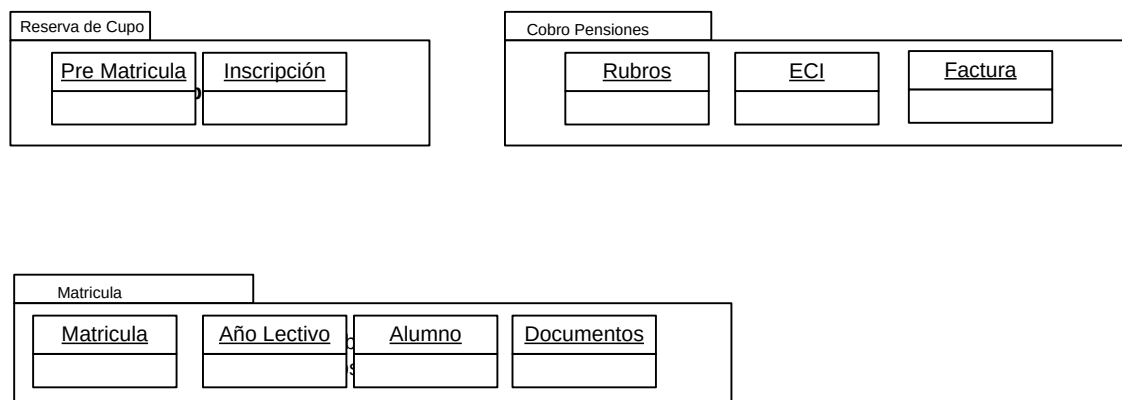
desconectar_alterno()

FinIntentar

Return lista_eci

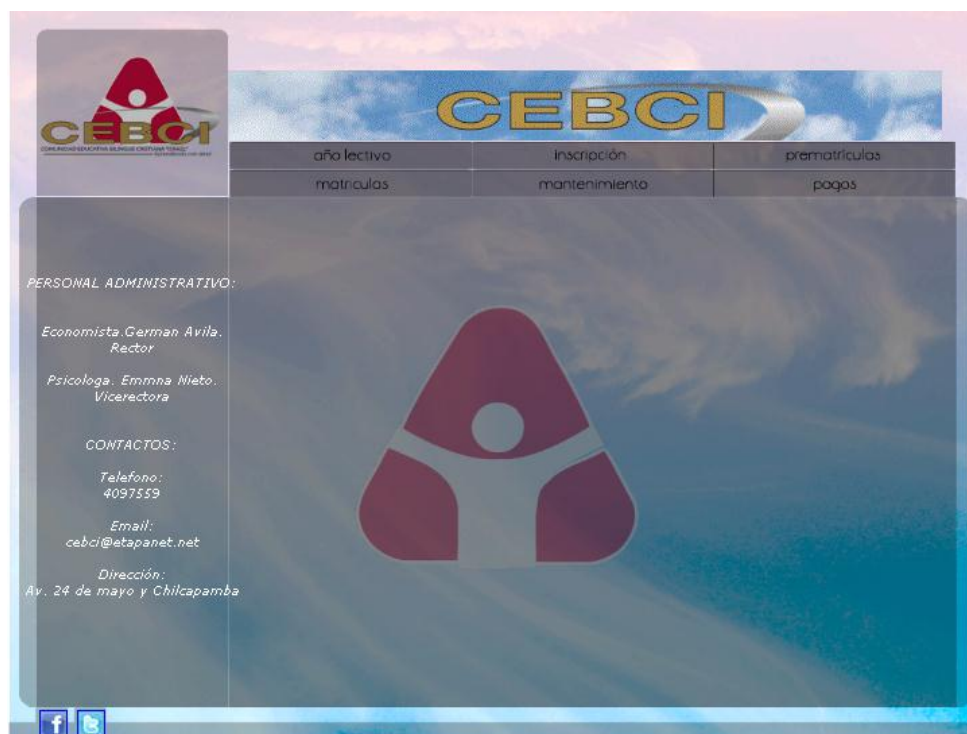
FinFuncion

3.5 Diseño de Subsistemas.

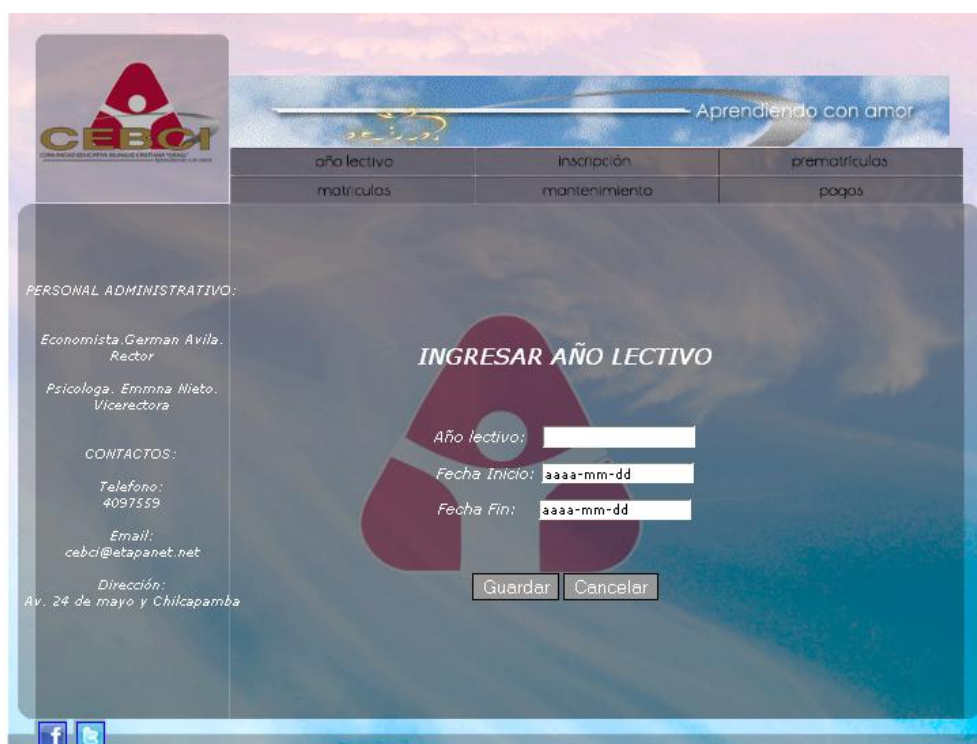


3.6 Interfaz de Usuario

3.6.1. MySql y PHP



Gráfica 3.1 Página Principal PHP



Gráfica 3.2 Página Menú Año lectivo PHP

CEBCI
COMUNIDAD EDUCATIVA BILINGÜE CHILCAPAMBA

Aprendiendo con amor

año lectivo	inscripción	prematriculas
matriculas	mantenimiento	pagos

PERSONAL ADMINISTRATIVO:

Economista: German Avila.
Rector

Psicóloga: Emma Nieto.
Vicerectora

CONTACTOS:

Teléfono: 4097559

Email: cebci@etapanet.net

Dirección: Av. 24 de mayo y Chilcapamba

DATOS

Año Lectivo: 2009-2010

Código Matricula: 1 Historial: 1

Código del Alumno: 777

Nombres: Jaime Fabian

Apellidos: Sangurima Alvarado

Dirección: Cdla. Tomebamba

Fecha de Nacimiento: aaaa-mm-dd

Nombre del Padre: Gustavo Sangurima Ocupación: Joyero

Nombre de la Madre: Magaly Alvarado Ocupación: Ama de casa

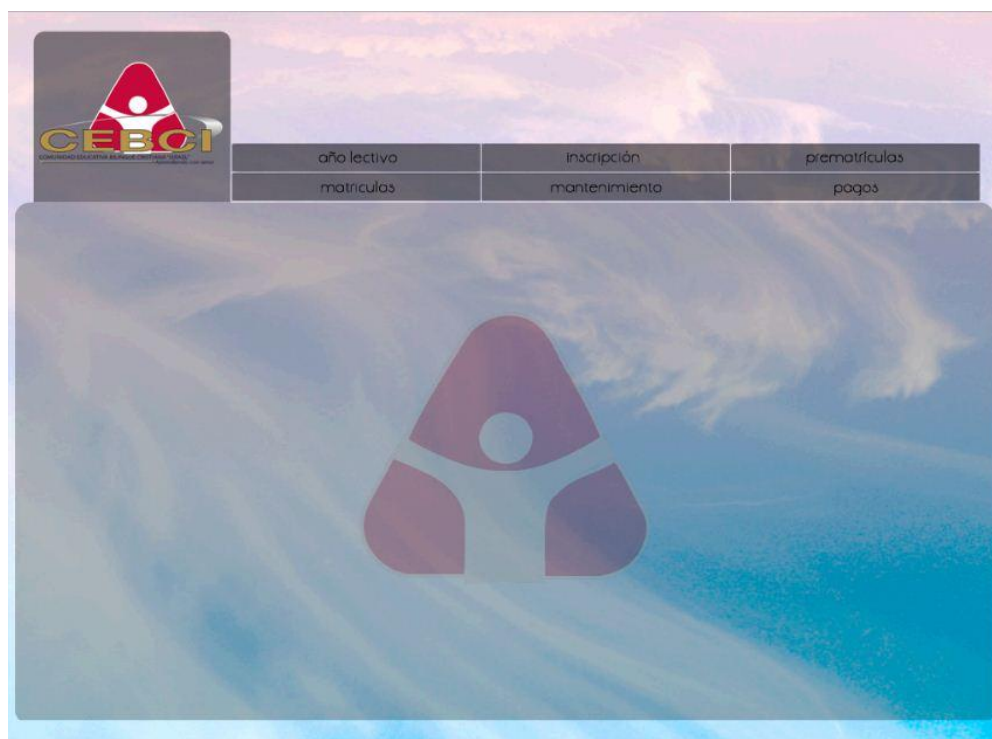
Nivel: 1 **Seleccionar**

Representante: Magaly Alvarado Relacion: Madre

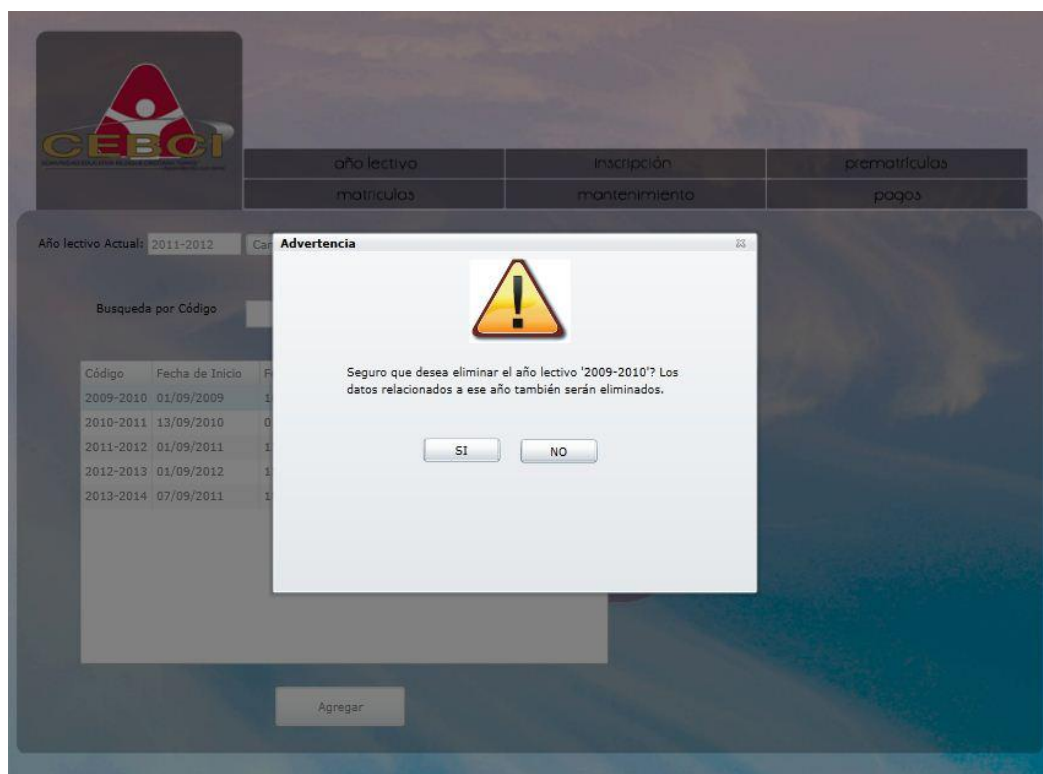
Guardar **Cancelar**

Gráfica 3.3 Página Menú Mantenimiento PHP

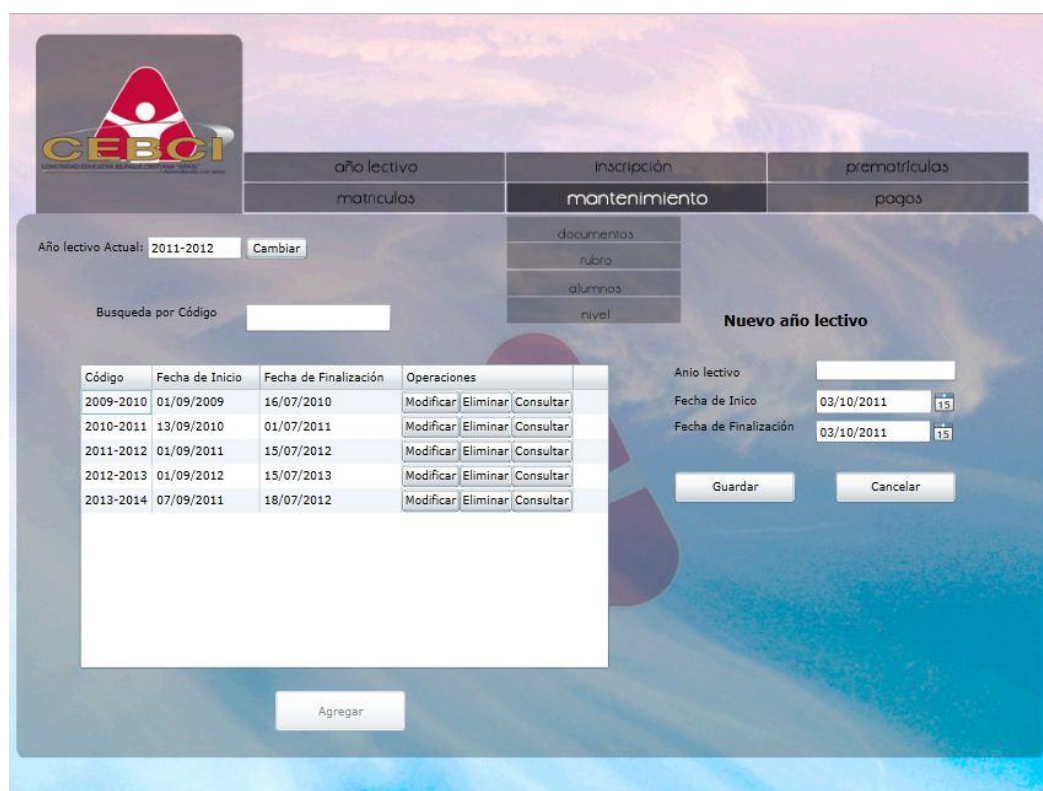
3.6.2. Silverlight



Gráfica 3.4 Página Principal Silverlight



Gráfica 3.5 Página mensajes de advertencias Silverlight



Gráfica 3.6 Página Menú Año Lectivo Silverlight

3.7. Diseño de Gestión de Datos.

3.7.1. PHP

A continuación explicaremos como se presenta la conexión de php.

Antes de nada debe estar instalado el servidor apache, este se lo puede realizar ejecutando el archivo `apache_1.3.23-win32-x86-no_src.exe`

Para instalar php se desempaqueta el archivo `php-4.1.2-Win32.zip` en algún directorio de Windows puede ser `c:\php` se busca el archivo `PHP4TS.DLL` y se copia este en el directorio `c:\windows\system`, luego copie todos los archivos que están en `c:\php\php-4.1.2-Win32\dlls` en la carpeta `c:\windows\system`. Si utiliza Windows XP copiarlos también a la carpeta `c:\windows\system32`. Se crea la carpeta `C:\Archivos de programa\Apache Group\Apache\php\` y se copia todo lo que está en `C:\php\php-4.1.2-Win32\` incluido las carpetas a `C:\Archivos de programa\Apache Group\Apache\php\`

En el directorio `c:\php\php-4.1.2-Win32\` existen dos archivos el `php.ini-recommended` y el `php.ini-dist` el primer archivo es la configuración de php con algunos cambios para optimizar su ejecución el segundo archivo es la configuración de php con las opciones por default, se va a escoger el primer archivo y se lo copiara en el directorio `c:\windows` cambiándolo de nombre a `php.ini`.

Con los pasos anteriores ya queda en funcionamiento la conexión con php.

3.7.2 SILVERLIGHT

Explicaremos la conexión de la base de datos MySQL con el conector ODBC.

1.- Primero descargamos lo siguiente:

- Conector MySQL. NET
- Conector MySQL ODBC

2.- Debemos tener ya creada nuestra base de datos e instalar correctamente los conectores antes descritos.

3.- Vamos al panel de control e ingresamos a las herramientas Administrativa, damos clic en orígenes de datos ODBC

- En la pestaña DNS del sistema damos clic en el botón agregar.
- Crear nuevo origen de datos buscando la opción MySQL ODBC driver

-Llenamos los datos de la siguiente manera:

- Data sourcename: nombre del conector de la base
- Descripción: describir brevemente al conector
- Server: colocamos LOCALHOST
- User: colocamos el nombre del usuario.
- Password: colocamos la clave de la base de datos
- DataBase: buscamos la base con la cual vamos a trabajar.

-Damos clic en aceptar.

Con todos estos pasos ya queda conectada la base de datos por medio del conector ODBC.

3.8. Gestión de Configuración.

Para el desarrollo de la gestión de configuración determinamos la línea base de cada módulo para su respectivo cambio durante su desarrollo

A continuación presentamos los informes de cambio.

PROYECTO: SIPMP

Sistema de Inscripciones, pre matricula, matrícula y pago de pensiones

CLIENTE: CEBCI (centro educativo bilingüe cristiano Israel)

ÁREA: Administrativa

CÓDIGO: SIPMP_001

VERSIÓN: 1.00

FECHA: 04/10/2010

3.8.1. Control de versiones PHP

Versión	Apartados afectados (módulos)	Descripción del cambio	Fecha del cambio
1.0	Conexión Año lectivo Documento	Nuevos módulos	04/10/2010
1.1	Nivel alumno Rubro Categoría	Nuevos módulos Modificación del módulo nivel	07/10/2010

1.2	Inscripción Pre matrícula	Nuevos módulos	10/10/2010
1.3	Facturación Inscripción	Nuevo modulo	12/10/2010
1.4	Inscripción	Adición del módulo de facturación inscripción	15/10/2010
1.5	Facturación Pre matrícula	Nuevo módulo	17/10/2010
1.6	Pre matrícula	Adición del módulo de facturación	21/10/2010
1.7	Matrícula	Nuevo módulo	25/10/2010
1.8	Facturación Matrícula	Nuevo módulo	27/10/2010
1.9	Matrícula	Adición módulo de facturación matricula	30/10/2010
1.10	Matrícula	Adición del módulo documento	05/11/2010
1.11	Matrícula	Adición del módulo de rubros	07/11/2010
1.12	Alumno	Nuevo documento	10/11/2010
1.13	Matrícula	Adición del módulo de alumno	14/11/2010
1..14	Pago Pensiones	Nuevo documento	18/11/2010
1.15	Pago Pensiones	Adición del módulo de rubros	21/11/2010
1.16	Rubros	Modificación de servicios	27/11/2010
1.17	Pago de pensiones	Adición del módulo de factura	03/01/2011
1.18	Menú	Creación de un menú dinámico	10/01/2011

3.8.2. Control de versiones SILVERLIGHT

Versión	Apartados afectados (módulos)	Descripción del cambio	Fecha del cambio
1.0	Conexión	Nuevo módulo	15/01/2011
1.1	Generación de servicios	Creación del módulo estándar de servicio	18/01/2011
1.2	Año lectivo	Nuevo módulo	25/01/2011
1.3	Nivel alumno	Nuevo módulo	26/01/2011
1.4	Rubro Categoría	Nuevo módulo	27/01/2011
1.5	Documento Alumno	Nuevo módulo	03/02/2011
1.6	Inscripción	Nuevo módulo	05/02/2011
1.7	Pre matrícula	Nuevo módulo	07/02/2011
1.8	Inscripción	Adición de factura	10/02/2011
1.9	Pre matricula	Adición de factura	12/02/2011
1.10	Matricula	Nuevo módulo	15/02/2011
1.11	Matricula	Adición de alumno, nivel	23/02/2011
1.12	Matricula	Adición de rubros, documentos	25/02/2011
1.13	Matricula	Adición de alumno	27/02/2011
1.14	Pago pensiones	Nuevo modulo	04/03/2011
1.15	Pago pensiones	Adición de rubros	06/03/2011
1.16	Pago pensiones	Adición de factura	09/03/2011
1.17	Menú	Creación de un menú dinámico	15/03/2011

3.9. Conclusiones

En el desarrollo de este capítulo se presentó el análisis y diseño para el desarrollo de la aplicación tanto para PHP y Silverlight, con la presentación de la información recolectada mediante entrevista a las personas involucradas pertenecientes al departamento administrativo del centro educativo CEBCI. Sin embargo la determinación de los diagramas y diccionario de datos nos facilita el entendimiento de las relaciones de la informaron recolectada con anterioridad. Además se realizo las conexiones correspondientes con la base de datos y la presentación de versiones para un control adecuado de cambios.

3.10. Recomendaciones

Tener en cuenta los requerimientos previos dados en la entrevista ya que esta ayuda a tener una visión más clara de que información se debe presentar en la aplicación. Además tener cuidado en el manejo del flujo de datos en los diagramas.

En Silverlight con ExpressionBlend tener en cuenta la funcionalidad de los objetos, estados, para el manejo correcto de los efectos dinámicos en la parte de diseño.

CAPÍTULO 4.

COMPARACIÓN DE TECNOLOGÍAS.

4.1.Introducción

Este capítulo presenta la comparación de tecnologías PHP vs Silverlight para el desarrollo del mismo se realizó pruebas de aplicación tomando en cuenta la conexión con la base de datos, la interacción con el cliente y compatibilidad de navegadores de internet para el funcionamiento adecuado.

A continuación se presenta un cuadro de comparación describiendo la cualidad, funcionalidad y eficiencia de cada tecnología.

4.2.Pruebas de aplicación.

4.2.1 PHP

Prueba	Conexión a la base de datos de MySQL.
Expectativa	La aplicación a nivel de servidor puede conectarse al gestor de base de datos MySQL mediante un módulo de extensión
Resultado	Se cumplió la expectativa.
Observaciones	La conexión se logró luego de instalar la aplicación WampServer luego se creó una página de prueba para determinar la conexión y así lograr con éxito el acceso a la base de datos desde PHP.

Tabla 4.1: Prueba de conexión PHP.

Prueba	Filtración de datos.
Expectativa	La aplicación a nivel de cliente puede utilizar la filtración de datos provenientes de la base de datos para su respectiva consulta.

Resultado	Se cumplió la expectativa.
Observaciones	La filtración de datos ayuda al cliente a tener una búsqueda más eficiente de la información requerida.

Tabla 4.2: Prueba de Filtración de datos PHP.

Prueba	Generación del menú personalizado.
Expectativa	A nivel de cliente se pretende implementar un menú con imágenes preestablecidas dentro de un diseño personalizado y manteniendo la funcionalidad de la aplicación.
Resultado	Se cumplió la expectativa.
Observaciones	El menú resalta el aspecto grafico colaborando en la elaboración de una aplicación intuitiva.

Tabla 4.3: Prueba generación del menú personalizado PHP.

4.2.2. Silverlight

Prueba	Conexión a la base de datos de MySQL.
Expectativa	La aplicación a nivel de servidor puede conectarse al gestor de base de datos MySQL mediante un conector ODBC.
Resultado	Se cumplió la expectativa.
Observaciones	La conexión se logró luego de instalar el Conector MySQL de .net y MySQL ODBC. Luego de una configuración rápida del conector recientemente instalado fue necesario crear una clase cuya función sería efectuar la conexión y así se logró tener acceso a la base de datos desde Silverlight.

Tabla 4.4: Prueba de conexión Silverlight.

Prueba	Vincular los datos de la base con la aplicación Silverlight en el equipo cliente.
Expectativa	La aplicación a nivel de cliente puede acceder a los datos de la base para efectuar todas las operaciones de mantenimiento (insertar, modificar, eliminar).
Resultado	Se cumplió la expectativa.

Observaciones	Aplicando Servicios Web se logró establecer un contrato donde el programa cliente tendría acceso a información específica de la base de datos mediante solicitudes preestablecidas.
----------------------	---

Tabla 4.5: Prueba de Filtración de datos Silverlight.

Prueba	Compatibilidad con los navegadores más importantes (Internet Explorer, Mozilla Firefox y Google Chrome).
Expectativa	La aplicación se ejecuta en los navegadores más importantes sin presentar variación de comportamiento en absoluto.
Resultado	Se cumplió la expectativa.
Observaciones	El programa fue revisado en cada uno de los navegadores ya mencionados en busca de incompatibilidades funcionales o visuales, pero no se encontró ninguna variación.

Tabla 4.6: Prueba de compatibilidad de navegadores Silverlight.

4.3 Cuadro de comparación entre PHP y Silverlight.

Para las comparaciones respectivas consideramos las siguientes métricas de calidad

Métricas:

Fiabilidad: hasta donde se puede esperar que un programa lleve a cabo su función con exactitud.

Valor	Descripción
1	No cumple sus funciones correctamente.
2	Presenta varias falencias en cumplir sus funciones.
3	Término medio.
4	Cumple con exactitud significativa.
5	Cumple sus funciones con exactitud.

Tabla 4.7: Métricas de fiabilidad.

Eficiencia: la cantidad de recursos informáticos y código necesario para que un programa realice su función.

Valor	Descripción
1	Requiere demasiados recursos para cumplir sus funciones.
2	Utiliza algunos recursos para su funcionamiento.
3	Término medio.
4	Optimiza significativamente los recursos.
5	Requiere recursos mínimos para cumplir sus funciones.

Tabla 4.8: Métricas de eficiencia.

Integridad: hasta donde se puede controlar accesos no autorizados al software o a los datos.

Valor	Descripción
1	Deplorable control de accesos a los datos.
2	Limitado control de acceso a los datos.
3	Término medio.
4	Control moderado en el acceso a los datos.
5	Control exhaustivo en el acceso a los datos.

Tabla 4.9: Métricas de Integridad.

Usabilidad (facilidad de manejo): el esfuerzo necesario para aprender, operar, preparar las entradas, interpretar las salidas de un programa.

Valor	Descripción
1	Complejidad en manejar interfaz y controles.
2	Interfaz poca intuitiva
3	Término medio.
4	Interfaz entendible y controles simples.
5	Facilidad en operar interfaz y controles.

Tabla 4.10: Métricas de Usabilidad.

Facilidad de mantenimiento: el esfuerzo necesario para localizar y arreglar un error del programa.

Valor	Descripción
1	Requiere gran esfuerzo de mantenimiento.
2	Complejidad moderada del mantenimiento.
3	Término medio.
4	Relativa facilidad del mantenimiento.
5	Facilidad de mantenimiento.

Tabla 4.11: Métricas de Facilidad de Mantenimiento.

Facilidad de prueba: el esfuerzo necesario para probar un programa.

Valor	Descripción
1	Mayor requerimiento de recursos en sus funciones de prueba.
2	Disponibilidad limitada en sus funciones de prueba.
3	Término medio.
4	Disponibilidad moderada en sus funciones de prueba.
5	Facilidad de en sus funciones.

Tabla 4.12: Métricas de Facilidad de Prueba.

Portabilidad: el esfuerzo necesario para transferir el programa de un entorno de hardware/software a otro.

Valor	Descripción
1	Mayor requerimiento de componentes de compatibilidad.
2	Requiere elevado esfuerzo para su implementación en otro ambiente.
3	Término medio.

4	Compatibilidad en la mayoría de sus componentes para su implementación en otro ambiente.
5	Esfuerzo mínimo de implementación en otro entorno.

Tabla 4.13: Métricas de Portabilidad.

Reusabilidad (capacidad de reutilización): hasta donde se puede volver a emplear partes o todo el programa en otras aplicaciones.

Valor	Descripción
1	Ningún componente puede ser reutilizado.
2	Algunos componentes pueden ser reutilizados
3	Término medio.
4	Presenta funciones y componentes de uso general.
5	Todas sus funciones y componentes son reutilizados.

Tabla 4.14: Métricas de Reusabilidad.

Interoperabilidad: el esfuerzo necesario para acoplar un sistema con otro.

Valor	Descripción
1	Se requiere demasiado esfuerzo de acoplamiento.
2	Requiere generar componentes externos para acoplarse a otro sistema.
3	Término medio.
4	Se requiere esfuerzo mínimo de acoplamiento con otro sistema
5	No se requiere de ningún componente externos para acoplarse.

Tabla 4.15: Métricas de Interoperabilidad.

Descripción	PHP	Silverlight
Fiabilidad	4	4
Eficiencia	5	3
Integridad	4	5
Usabilidad	3	5
Facilidad de Mantenimiento	2	4
Facilidad de prueba	4	4
Portabilidad	5	4
Reusabilidad	3	3
Interoperabilidad	2	5

Tabla 4.16: Cuadro comparativo PHP y Silverlight.

4.4. Conclusiones

Se presentó la comparativa entre los dos lenguajes PHP y Silverlight mediante pruebas de aplicación las cuales en si fueron funcionales en las aplicaciones.

En relación al cuadro de comparación se determinó mediante las métricas de calidad tomando en cuenta su corrección, fiabilidad, usabilidad, portabilidad, entre otras cuyas características nos ayudaron a determinar la eficiencia y facilidad de uso entre los dos lenguajes. En resumen se determinó que Silverlight en general tiene mayor versatilidad para generar sistemas adaptables a varios entornos según el requerimiento de los usuarios.

4.5 Recomendaciones

Tener en cuenta las pruebas de aplicación y los estándares de calidad para su respectiva comparación.

CONCLUSIONES

Al inicio del estudio investigamos y explicamos los conceptos para entender de mejor manera las aplicaciones web tradicionales y RIA; las cuales presentaron sus funcionalidades del manejo cliente - servidor, además de las limitaciones en el manejo de la información; sin embargo RIA presenta gran facilidad de acceso, una sola vez al servidor y así evitar la sobrecarga de la página como lo hace la web tradicional.

Luego analizamos RIA dentro de Silverlight, las cuales son aplicaciones para la red, más intuitivas, dinámicas en multimedia con la utilización de la herramienta ExpressionBlend, igualmente Silverlight es una herramienta de actualidad la cual ha ido evolucionando constantemente presentando mayores cambios dentro de sus objetos. La comparación de Silverlight con otras herramientas nos dio como resultado la facilidad de uso y el manejo de servicio que nos brinda.

Posteriormente en el siguiente capítulo se determinó la información del análisis y diseño del sistema para el desarrollo de la aplicación tanto en PHP y Silverlight, mediante la información adquirida por el personal del departamento administrativo del centro educativo CEBCI. Con la presentación de los diagramas y diccionario de datos nos ayudaron en el desarrollo a determinar las dificultades y virtudes de ambas tecnologías en cuanto a su implementación y diseño.

Finalmente se presento la comparativa entre los dos lenguajes PHP y Silverlight mediante las pruebas de aplicación. Así mismos se presentó un cuadro comparativo mediante las métricas de calidad, cuyas características nos ayudaron a determinar la eficiencia y facilidad de uso entre los lenguajes.

En si podemos concluir con la perspectiva más clara sobre la tecnología RIA la cual nos ayuda en el desarrollo en aplicaciones de escritorio que hoy en día son más requeridas.

GLOSARIO.

APPLETS: Los Applets de Java están programados en Java y pre compilados, es por ello que la manera de trabajar de éstos varía un poco con respecto a los lenguajes de Script como JavaScript. Los Applets son más difíciles de programar que los scripts en JavaScript y requerirán unos conocimientos básicos o medios del lenguaje Java.

ASP (Active Server Pages): Tecnología creada por Microsoft para generar páginas Web en forma dinámica. Se ejecuta en el lado del servidor y se ejecuta mediante IIS (Internet Information Services, servidor desarrollado por Microsoft).

CLR (Common Language Runtime): Es la herramienta de programación utilizada en .NET que permite a los desarrolladores elaborar código en varios lenguajes de programación (C++, Visual Basic, etc.), y tomarlo para generar un código unificado nativo de máquina.

COOKIE: Es un fragmento de información que se almacena en el disco duro del computador por medio del servidor, el cual puede ser recuperado cuando este lo requiera en futuras visitas.

DHTML: El HTML Dinámico o DHTML (DynamicHTML) designa el conjunto de técnicas que permiten crear sitios Web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript) y el lenguaje de Hojas de estilo en cascada (CSS).

DOM: El DOM es una jerarquía de objetos predefinidos que describen los elementos de una página web que está mostrando el navegador, así como otras características del proceso de navegación como son el historial, el tamaño de la ventana de navegación.

ECLIPSE: Eclipse es una gran estructura formada por un núcleo y muchos plug-ins que van conformando la funcionalidad final. La forma en que los plug-ins interactúan es mediante interfaces o puntos de extensión.

EXPRESSION BLEND: Es una herramienta de diseño creada y soportada por Microsoft. Permite aprovechar al máximo las herramientas gráficas de .NET y Silverlight gracias a su interfaz intuitiva y sus controles que ayudan a la generación de efectos visuales con facilidad.

HTTP: (Hyper Text Transfer Protocol) es un estándar para el intercambio de archivos (texto, gráficos y multimedia) a través de Internet. Es un protocolo sin estado de tipo cliente/servidor.

IDE: Son las siglas de IntegratedDevelopmentEnvironment, es decir, un entorno integrado de desarrollo.

JAVASCRIPT: Lenguaje de programación interpretado, orientado a objetos que usualmente se ejecuta en el lado del cliente mediante un navegador de internet y genera procedimientos dinámicos que enriquecen la parte visual y aportan a la funcionalidad de las páginas WEB.

LAMP: El término LAMP se origina a finales del 2000 en Alemania para describir a las aplicaciones Web creadas utilizando la siguiente combinación de herramientas: Linux, Apache, Mysql y Perl/PHP/Python.

.NET: Marco de trabajo desarrollado y soportado por Microsoft. Es un grupo de herramientas que permiten elaborar aplicaciones tanto de escritorio así como WEB, permite programar en varios lenguajes y soporta varias tecnologías de desarrollo.

PLUGIN: Es un conjunto de componentes de software que agrega funcionalidades de otra aplicación más grande, el cual permite personalizar una aplicación a sus necesidades.

PYTHON: Lenguaje de programación de código abierto conocido como multiparadigma ya que es capaz de manejar programación orientada a objetos así como programación estructurada o imperativa.

RIA: Aplicaciones de Internet Enriquecida (Rich Internet Application). Es la nueva tendencia informática que busca mejorar la experiencia del usuario mediante el aprovechamiento de las aplicaciones web, agregando la robustez de las aplicaciones de escritorio, evitando así el refrescado permanente y disminuyendo el tráfico entre cliente y servidor.

RSS: Es una tecnología que está basada en XML, el cual usa “Sindicar”, los usuarios obtienen los contenidos mediante lectores o agregadores.

SERIALIZACIÓN: La serialización de datos es el proceso en el que se toman objetos y se convierte su información de estado en un formato que permita su almacenamiento. La acción inversa consiste en deshacer la serialización y dejar el objeto con la misma forma que tenía inicialmente.

SERVLETS: Los Servlets son objetos que corren dentro del contexto de un servidor de y extienden su funcionalidad.

SISTEMA HETEROGÉNEO: Está compuesto por hardware y software con características distintas pero los cuales se pueden comunicar utilizando medios comunes.

URL: UniformResourceLocator (Localizador Uniforme de Recursos). Forma de organizar información en la aplicación web. Es la dirección que permite acceder a un archivo como HTML, PHP, ASP, entre otros.

WEBKIT: El motor de renderización con el que funcionan navegadores como Konqueror y Safari, además de Adobe Flash y Flex, ofrece una serie de API's para facilitar el desarrollo de aplicaciones de escritorio. WebKit es un producto de Apple, creado a partir de las librerías KHTML y KJS de KDE. KDE es un entorno de escritorio para Linux, que dispone de librerías para la interpretación de código HTML (KHTML) y JavaScript (KJS), así que WebKit partió de estas librerías para su creación, portándolas al sistema operativo Mac OS X.

WIDGETS: Es una pequeña aplicación presentado en archivos que son ejecutados por un motor de Widgets o WidgetsEngine. Entre sus objetivos están los de dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.

XAML: Lenguaje Extensible de Formato para Aplicaciones (eXtensibleApplicationMarkupLanguage). Es el lenguaje de marcas creado por Microsoft para definir interfaces gráficamente enriquecida. Es utilizado por las herramientas WPF y Silverlight dentro de .NET.

XML: Extensible MarkupLanguage (lenguaje de marcado ampliable extensible) desarrollado por el World Wide Web Consortium (W3C).

XMLHttpRequest: Este puede ser utilizado por las secuencias de comandos para la determinada conexión con el servidor mediante programación a través de HTTP

XLST: (Extensible StylesheetLanguageTransformations) es un lenguaje declarativo, basado en XML utilizado para la transformación de documentos XML en otros documentos XML.

BIBLIOGRAFÍA.

- [1] ADOBE, S. I. (14 de 7 de 2009). *ADOBE AIR*. Recuperado el 18 de 9 de 2010, de Deliver Engaging Application: <http://www.adobe.com/products/air/>
- [2] ALVAREZ, M. (15 de 7 de 2008). *WebKit*. Recuperado el 18 de 9 de 2010, de Desarrolladores Web.com: <http://www.desarrolloweb.com/articulos/que-es-webkit.html>
- [3] ANTONIO. (14 de 4 de 2009). *Definición de URL*. Recuperado el 18 de 9 de 2010, de ALEGSA.com: <http://www.alegsa.com.ar/Dic/url.php>
- [4] BELLIDO, A. (11 de 8 de 2004). *RSS sindicando contenido*. Recuperado el 6 de 9 de 2010, de Maestros del Web: <http://www.maestrosdelweb.com/editorial/sindicando/>
- [5] GOOGLE, C. (5 de 7 de 2009). *Compile & Debug*. Recuperado el 18 de 9 de 2010, de Google.com: <http://code.google.com/webtoolkit/doc/latest/DevGuideCompilingAndDebugging.html>
- [6] ITISMO. (4 de 7 de 2008). *Sistemas Heterogeneos*. Recuperado el 17 de 9 de 2010, de Sistemas Distribuidos: http://www.itistmo.edu.mx/Pag%20Informatica/APUNTES_archivos/page0003.htm
- [7] KESTEREN, A. (8 de 3 de 2009). *XMLHttpRequest*. Recuperado el 18 de 9 de 2010, de W3C XMLHttpRequest: <http://www.w3.org/TR/XMLHttpRequest/#the-xmlhttprequest-interface>
- [8] LOOSLEY, C. (4 de 7 de 2006). *Rich Internet Applications*. Recuperado el 18 de 9 de 2010, de Scribd.com: <http://www.scribd.com/doc/263488/KeyNoteSystem-Rich-Internet-Applications-Design->
- [9] MARQUES, P. (20 de 08 de 2007). *La Web 2.0 y sus aplicaciones didácticas*. Recuperado el 5 de 9 de 2010, de peremarques.net: <http://www.peremarques.net/web20.htm>
- [10] MICROSOFT, M. (14 de 5 de 2008). *Silverlight*. Recuperado el 18 de 9 de 2010, de MSDN Magazine: <http://msdn.microsoft.com/es-ec/magazine/cc163404.aspx>
- [11] NIETO, I. (23 de 8 de 2008). *El Modelo de Objetos DOM*. Recuperado el 18 de 9 de 2010, de Java Avanzado codigo.com: <http://www.elcodigo.com/tutoriales/jsavanzado/jsavanzado7.html>
- [12] PALACIOS, E. (20 de 12 de 2008). *Trabajo de Grado*. Recuperado el 5 de 9 de 2010, de Aplicaciones Ricas en Internet (RIA): <http://revista2.linti.unlp.edu.ar/tesinas/tesis44.pdf>
- [13] PUJOL, M. (30 de 11 de 2007). *Que es Web 1.0*. Recuperado el 5 de 9 de 2010, de Documentacion Informatica: <http://www.scribd.com/doc/550264/Que-es-web-10>

- [14] RIBES, X. (4 de 10 de 2009). *TELOS*. Recuperado el 16 de 8 de 2010, de La Web 2.0 El VAlor de los metadatos y de la inteligencia colectiva: <http://sociedadinformacion.fundacion.telefonica.com/telos/articuloperspectiva.asp@idarticulo%3D2&rev%3D73.htm>
- [15] VICTORAZZI, N. (8 de 5 de 2008). *Serializacion de datos*. Recuperado el 18 de 9 de 2010, de Proyectos y Desarrollos: http://147.96.80.209/SI_G2_04_05/src/proyecto/Serializacion.htm
- [16] WESLEY, A., OPDYKE, W., & BECK, k. (2007). *Refactoring Improving the Design of Existing Code*.
- [17] BEADLE, P., & KRISHNAN, M. (2010). *Silverlight 4 for dummies*. Indianapolis: Wiley Publishing Inc.
- [18] DAYLEY, B. (2008). *Silverlight 2*. Indianapolis: Wiley Publishing Inc.
- [19] Ghoda, A. (2010). *Introducing Silverlight 4*. Apress.
- [20] Microsoft. (2010). *Microsoft Silverlight*. Recuperado el 19 de 09 de 2010, de <http://www.silverlight.net/getstarted/silverlight-4/>
- [21] ProgrammerWorld.NET. (2009). *ProgrammerWorld.NET*. Recuperado el 30 de 08 de 2011, de <http://faq.programmerworld.net/programming/silverlight.html>
- [22] SCANLON, J. (2008). *Accelerated Silverlight 2*. Apress.
- [23] WindowsCliente.NET. (2010). *WindowsClient.NET*. Recuperado el 12 de 09 de 2010, de <http://windowsclient.net/wpf/>