



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

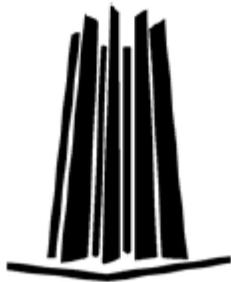
**“SISTEMA AUTOMATIZADO DE CONTROL
PARA UNA SUPERVISIÓN ESCOLAR”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

**P R E S E N T A:
JULIO CESAR CRUZ CRUZ**

**ASESOR DE TESIS:
ING. ERNESTO RAMIREZ MONTALVO**



MÉXICO, 2007.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradezco:

A mi Padre Celestial por darme la oportunidad de cumplir una más de mis metas.

A las tres mujeres más importantes de mi vida: a mi mamá, por darme la vida y por haberme enseñado principios correctos, a mi esposa, por impulsarme a seguir adelante y ser mi ayuda idónea y a mi hija por ser esa chispa que mueve mi vida.

A mi padre que no pudo estar presente hasta esta etapa de mi vida, pero me dio las bases para poder llegar.

A mis hermanos Raúl, Vero, Ana, Andy, Marco y Paco que aportaron su grano de arena.

A la Universidad Nacional Autónoma de México por albergarme en sus aulas durante todos estos años y ser la formadora de mi futuro profesional.

A mis buenos profesores por el conocimiento transmitido.

A mi asesor, Ing. Ernesto Ramírez Montalvo, por confiar en mí y arriesgarse conmigo en este proyecto.

Al M. en C. Marcelo Pérez Medel y al M. en C. Felipe de Jesús Gutiérrez López por el gran apoyo ofrecido

A todos los que de una u otra manera, formaron parte de esta obra.

INTRODUCCIÓN	4
CAPÍTULO 1	
APLICACIONES WEB	5
1.1 Sistemas de Información	6
1.1.1 Modelo General de un sistema	6
1.1.2 Componentes estructurales del sistema de información.	7
1.1.3 Tipos de sistemas de información	7
1.2 Aplicaciones Web	9
1.3 Arquitectura Cliente Servidor	10
1.3.1 Arquitecturas de software de dos capas	11
1.3.2 Arquitecturas de software de tres capas	11
1.4 Arquitecturas de los servidores de aplicaciones Web	12
1.4.1 CGI	13
1.4.2 Modelo de co-servidor	14
1.4.3 Intérprete incrustado	14
1.5 El protocolo HTTP	15
1.5.1 Mecanismo para el manejo de estados en el HTTP	16
1.6 Los lenguajes de marcado	16
1.6.1 Las hojas de estilo CSS	17
1.6.2 Scripts de lado del cliente	18
1.7 El lenguaje SQL	19
1.8 Cifrado de información	20
CAPÍTULO 2	
COMPONENTES DE APLICACIONES WEB CON USO DE BASES DE DATOS.	22
2.2.1 CGI nativo	24
2.1.2 Servlets	24
2.1.3 Módulos del servidor Web	25
2.1.4 Código incrustado en el HTML	25
2.2 Mecanismos para conocer el estado del usuario	26
2.2.1 Validación de datos	26
2.2.2 Autenticación, autorización de usuarios y Sesiones	28
2.2.3 Despachador de mensajes	30
2.3 El Proceso Unificado	31

2.4 UML	34
2.4.1 Modelos y Diagramas	35
2.4.2 Casos de Uso	35
2.4.3 Diagrama de Clases	36
2.4.4 Diagramas de Interacción	37
2.4.5 Diagrama de Estados	39
2.5 Bases de Datos	40
2.5.1 Bases de datos para la Web	40
2.5.2 Modelo de Datos	41
2.5.3 DBMS	43
2.5.4 Conexiones a Bases de Datos	44
2.6 Capa de Reglas de negocio (lógica de la aplicación)	44
2.7 Capa de presentación	46
2.7.1 Presentación interactiva	46
CAPÍTULO 3	
ANÁLISIS	48
3.1 Análisis preliminar del problema	49
3.1.1 Situación Actual	50
3.2 Alcances	53
3.3 Estudio de Factibilidad	55
3.3.1 Factibilidad Técnica	55
3.3.2 Factibilidad Económica	58
3.3.3 Factibilidad Operacional	61
3.4 Determinación de requerimientos	62
3.4.1 Requerimientos de Funcionalidad	62
3.4.2 Requerimientos de uso	67
3.4.3 Requerimientos de confiabilidad.	67
3.4.4 Requerimientos de rendimiento.	68
3.4.5 Requerimientos de soporte	68
3.4.6 Restricciones de diseño	69
3.4.7 Requerimientos de documentación.	69
3.4.8 Interfaces	69
3.5 Reglas de negocio	70
3.6 Casos de Uso	71

CAPÍTULO 4

DISEÑO DEL SISTEMA 82

4.1 Propuesta de solución 82

4.2 Diseño Conceptual 82

4.2.1 Diagramas de secuencia 82

4.2.2 Diagramas de clases conceptuales del sistema 82

4.3 Diseño Físico 82

4.3.2 Capa de datos 82

4.3.3 Capa de proceso 82

4.3.4 Capa de Presentación 82

CONCLUSIONES 82

GLOSARIO 82

CITAS Y REFERENCIAS 82

BIBLIOGRAFÍA 82

Introducción

En el presente trabajo se profundizó en los elementos de las aplicaciones para la Web y las metodologías que se utilizan para el análisis y el diseño de las mismas, también se muestra de manera práctica como aplicar los conocimientos teóricos a un problema real.

Durante los dos últimos ciclos escolares (2004-2005 y 2005-2006) en la supervisión escolar P095 de Naucalpan de Juárez, en el Estado de México, las escuelas particulares y públicas pertenecientes a esta, han presentado una diferencia importante en los tiempos de entrega de la documentación y estadísticas de la matrícula escolar, a causa de conflictos con los sistemas de control enviados por las autoridades educativas, por lo que cada escuela debe llevar a cabo como puede este trabajo.

Esta situación, provoca un problema mayor al final del ciclo escolar: error en los datos con la consecuente falta de documentos oficiales.

La anterior descripción permite plantear el objetivo de usar la tecnología para la optimización de los recursos con que se cuenta en la supervisión y automatización de los procesos de registro, validación, consulta y exportación de datos que proporcione de manera integral la información de los alumnos y docentes.

A través de esta la investigación, se revisan los conceptos fundamentales de un sistema de información computacional, las tecnologías, arquitecturas y protocolos involucrados en el desarrollo de aplicaciones distribuidas y principalmente en aplicaciones Web con uso de bases de datos, esta información se desarrolla en el primer capítulo.

En el segundo capítulo se mencionan algunos enfoques que han surgido para la programación de aplicaciones Web con uso de bases de datos, los cuales tienen diferentes mecanismos y metodologías para realizar este tipo de desarrollos. Se toman algunos de los problemas o situaciones con los que tiene que tratar el desarrollador de aplicaciones Web.

La definición de requisitos y una minuciosa revisión de las características más importantes de estos, la podemos encontrar en la fase del análisis que se encuentra en el tercer capítulo. También en este apartado, se revisan las representaciones de los diagramas UML para el dominio del proyecto y los casos de uso.

El estudio referente al diseño en capas para una aplicación en Web, se hace en el cuarto capítulo al igual que la consecuente utilización de las herramientas para la creación del ambiente de desarrollo y en nuestro caso, del prototipo, con lo que se concluye la presente obra.



Capítulo 1

APLICACIONES WEB

Este capítulo se enfoca al estudio de las tecnologías y protocolos involucrados en el desarrollo de aplicaciones Web. Inicialmente se revisan los conceptos de sistema de información, al igual que en los diferentes tipos en los que se puede clasificar, arquitectura Cliente-Servidor de los sistemas de información, con lo cual se tiene una mejor comprensión de cómo funcionan las aplicaciones Web. Se analizan algunos de los lenguajes y protocolos que son el fundamento de las aplicaciones Web: el HTML, y su medio de transporte: el HTTP. También se habla acerca del lenguaje SQL, el cual es imprescindible si es que se desea desarrollar aplicaciones que usen bases de datos.



1.1 Sistemas de información

El termino Sistemas de información ha sido definido de diferentes modos, según la perspectiva de cada autor, Un autor lo define como “el mecanismo por el cual se generará información”[1], otro como “Es un conjunto de reglas, prácticas y procedimientos mediante las cuales, personas, equipos o ambos, operan para procesar determinadas entradas de datos y producir información que satisface las necesidades de ciertos individuos en determinadas situaciones de la empresa”.

Tomando como base lo anterior se puede definir a un sistema de información como un conjunto de elementos como lo son: datos, personas, tecnología y procesos, para cumplir un objetivo específico o favorecer la ventaja competitiva de una organización.

El sistema de información tiene entradas (datos e instrucciones) y salidas (reportes, resultados de cálculos, etc.), procesa las entradas generando las salidas que serán enviadas al usuario o a otro sistema o subsistema. Además de un proceso de retroalimentación que controle la operación del mismo

1.1.1 Modelo general de un sistema

El conjunto de elementos que interactúan dentro del sistema son de naturaleza diversa y normalmente incluyen:

- a) Entrada de información. Proceso mediante el cual el sistema de información toma o solicita los datos requeridos para procesarlos y convertirlos en información, siendo esta manual o automática.
- b) Almacenamiento de información. Es una de las propiedades más importantes en la computadora, así que el sistema a través de esta propiedad puede recuperar la información guardada. La información es almacenada generalmente en estructuras conocidas como archivos.
- c) Procesamiento de información. Es la capacidad de sistema para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecidas. Las operaciones pueden realizarse con datos introducidos recientemente o almacenados en el sistema. Es importante destacar que una transformación de los datos fuente a información es una de las principales utilidades de los sistemas en las organizaciones
- d) Salida de información. Es la capacidad para mostrar la información procesada o los datos de entrada al exterior. La salida de un sistema de información puede constituir la entrada a otro sistema o modulo

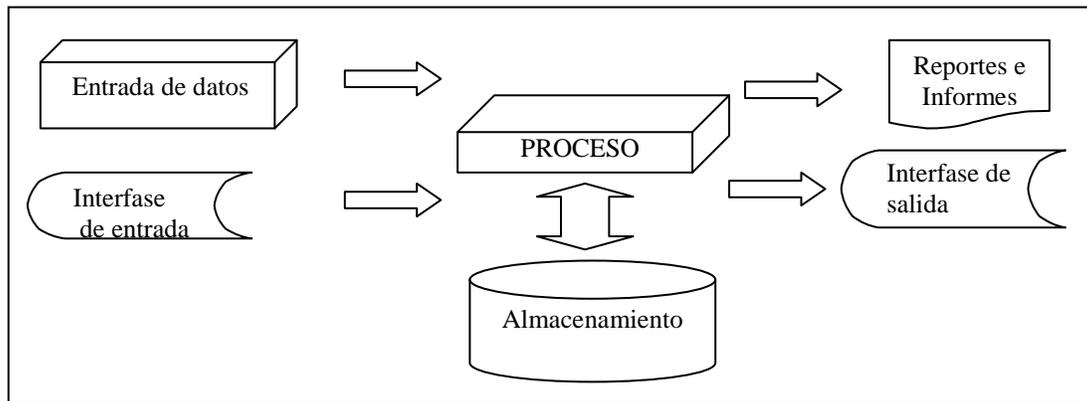


Figura 1.1. Diseño conceptual de un sistema de información.

1.1.2 Componentes estructurales del sistema de información.

Los componentes estructurales de un sistema son simplemente las partes que pueden ser identificables en el, estos componentes poseen propiedades o características que influyen en la operación del sistema, confiabilidad, capacidad y muchos otros aspectos.

Los sistemas de información sin importar la forma en que se desarrollen y diseñen, generalmente están formados por los siguientes componentes estructurales[2]: entrada, modelos, salida, tecnología, base de datos y controles. La forma en la que se acoplen unos con otros y los valores que vayan obteniendo recae en la responsabilidad del diseñador, quien en la medida en que mejor saque provecho de estos, desarrollará mejor los sistemas.

1.1.3 Tipos de sistemas de información

La clasificación de los sistemas de información se puede hacer a través de los tres principales objetivos que cumplen dentro de la organización:

1. Automatizar los procesos operativos (transaccional).
2. Proporcionar información de apoyo a la toma de decisiones.
3. Conseguir ventajas competitivas a través de su uso (estratégicos).

A continuación se presentan las principales características de estos tipos de sistemas de información.

1.1.3.1 Sistema Transaccional

Los sistemas transaccionales se caracterizan por:

- La automatización que favorece a las tareas operativas de la organización.



- Frecuentemente son los primeros sistemas de información en implementarse en la organización, apoyando primeramente las tareas de los niveles de operativos y posteriormente los niveles medios y directivos.
- La información requerida a la entrada de estos sistemas es extensa al igual que la de salida y el proceso efectuado es simple o poco complejo.
- Generalmente son los principales recolectores, almacenadores e integradores de la información que se maneja en la organización.
- La justificación de estos sistemas es fácil, debido a que las ventajas y resultados de su utilización es palpable.
- La adaptabilidad de paquetes de aplicaciones comerciales es fácil porque automatizan generalmente los procesos similares en las organizaciones.

1.1.3.2 Sistemas para la toma de decisiones.

Las características que se pueden resaltar en estos sistemas son:

- El ingreso de estos sistemas a la organización se hace después de los sistemas transaccionales, los cuales les sirven de plataforma.
- Los mandos intermedios y administrativos son los más beneficiados al ser implementados estos sistemas en la organización.
- Las entradas y salidas de información son escasas, sin embargo los cálculos son intensivos.
- Su justificación económica no es sencilla
- La interacción con los usuarios y las interfases amigables están presentes en estos sistemas.
- La utilización de estos sistemas es relevante en el proceso de apoyo a la toma de decisiones.

1.1.3.3 Sistemas estratégicos.

Las características que se pueden resaltar en estos sistemas son:

- La especialización de estos sistemas no es primordialmente para automatizar procesos ni para el apoyo en la toma de decisiones, pero las pueden llevar a cabo en caso de ser necesario.



- El desarrollo de estos sistemas es *in house*¹.
- Las funciones en este tipo de sistemas se incrementa, empezando con un proceso en particular y de ahí se van incrementando nuevos.
- El principal objetivo que persiguen los sistemas estratégicos es el de otorgar ventajas competitivas a la organización
- Las ventajas competitivas obtenidas con este tipo de sistemas con el tiempo tienden a convertirse en requisitos de infraestructura tecnológica.

1.2 Aplicaciones Web

El desarrollo de la Web², que es el medio que permite la transferencia de información de una forma fácil y rápida, por una parte y la evolución de las antiguas aplicaciones distribuidas en CD's a aplicaciones dinámicas, de constante actualización capaces de adaptarse a diferentes tipos de usuarios, dieron lugar al desarrollo de las aplicaciones Web.

Una aplicación Web es un sistema donde la navegación a través del sitio y las entradas de datos por parte del usuario tienen efecto sobre el estado del negocio. En esencia una aplicación Web toma un sitio Web como entrada (Front-end) a una aplicación convencional.

“...Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación Web.”[3]

Las aplicaciones Web son aplicaciones basadas en la arquitectura de software cliente/servidor.

En la computación cliente-servidor “tradicional”, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, requería en la mayoría de las veces una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal.

Con la aparición de las aplicaciones Web este problema ha sido ampliamente superado, permitiendo ejecutar la aplicación desde el servidor mismo, sin tener la necesidad de instalar nada en el equipo local del usuario.

¹ Generalmente no se pueden adaptar a paquetes disponibles en el mercado, creados a la medida de necesidades específicas.

² El World Wide Web Consortium (W3C) define el término *Web* como: “*el universo de información accesible a través de la red*”.



Las páginas Web son el componente principal de una aplicación o sitio Web. Los navegadores piden páginas (almacenadas o creadas dinámicamente) con información a los servidores Web.

La información mostrada en las páginas está típicamente almacenada en archivos. Sin embargo, muchas veces esta información está almacenada en una base de datos, y las páginas son creadas dinámicamente.

1.3 Arquitectura Cliente-Servidor

A finales de la década de los ochentas surge el termino cliente-servidor como resultado de las limitaciones de las arquitecturas *mainframe*³ y *file sharing*⁴, con esta arquitectura se aprovechó el creciente poder de las computadoras personales y fue complementado con los beneficios de los mainframes, ya que estos si podían compartir información de manera simultanea entre diferentes usuarios del sistema.

Con esta arquitectura se introdujo un servidor de base de datos para sustituir al servidor de archivos. La arquitectura cliente-servidor redujo el tráfico en la red proporcionando solo la respuesta de la consulta en vez de transferir todo el archivo y haciendo que el usuario solo se conecte cuando es necesario, obtiene los datos que necesita y se desconecta, dejando la red libre.

Se denomina cliente al solicitante de servicios y servidor al proveedor de servicios. Una sola máquina puede ser tanto un cliente como un servidor dependiendo de la configuración del software. La arquitectura cliente-servidor permitió facilidad de uso, flexibilidad, interoperabilidad y escalabilidad.

Esta arquitectura ha evolucionado para solucionar distintos problemas, entre los modelos de esta arquitectura de software más comunes y en orden de complejidad se pueden encontrar:

- Arquitecturas de dos capas
- Arquitecturas de tres capas
- Arquitectura de tres capas con tecnología de monitoreado en el procesamiento de transacciones
- Tres capas con servidor de mensajes
- Tres capas con un servidor de aplicaciones
- Tres capas con arquitectura ORB⁵
- Arquitectura de empresa distribuida/colaborativa

³ Esta arquitectura basaba toda la operación en una computadora central

⁴ El servidor descarga archivos de la localización compartida al ambiente de escritorio

⁵ Object Request Broker (Corredor de Peticiones de Objetos).



Para aplicaciones Web se utiliza desde las aplicaciones de dos capas en adelante, sin embargo es más conveniente utilizar la de tres capas por las razones que en los párrafos siguientes se expondrá.

1.3.1 Arquitecturas de software de Dos Capas

La arquitectura de dos capas consiste en tres componentes distribuidos en dos capas: cliente (solicitante de servicios) y servidor (proveedor de servicios). Los tres componentes son:

1. **Interfaz de usuario al sistema.** Tales como una sesión, entradas de texto, desplegado de menús, etc.
2. **Administración de procesamiento.** Tales como la ejecución de procesos, el monitoreo de los mismos y servicios de procesamiento de recursos.
3. **Administración de bases de datos.** Tales como los servicios de acceso a datos y archivos.

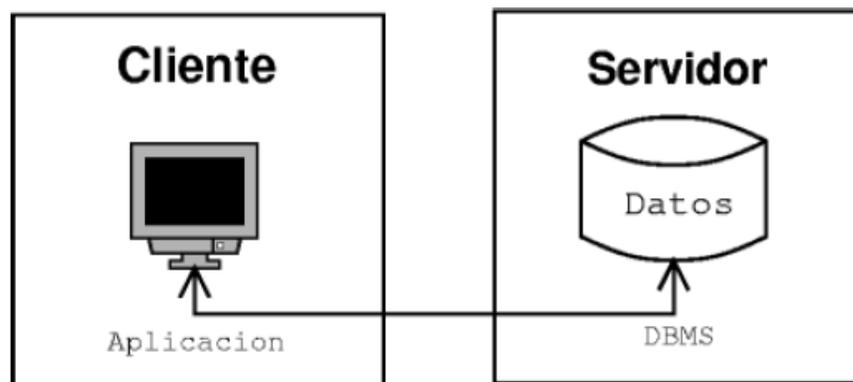


Figura 1.2: Aplicaciones de dos capas.

En este diseño se coloca la interfaz del usuario exclusivamente en el cliente. Coloca la administración de base de datos en el servidor y divide la administración de procesos entre el cliente y/o el servidor, creando únicamente dos capas.

1.3.2 Arquitecturas de software de Tres Capas

La arquitectura de tres capas surgió para solventar las limitaciones de la arquitectura de dos capas. La tercera capa (capa de servicios) se localiza entre la interfaz de usuarios (cliente) y el administrador de datos (servidor). Esta capa intermedia provee de servicios para la administración de procesos (tal como el desarrollo, monitoreo y alimentación de procesos) que son compartidos por múltiples aplicaciones.



Figura 1.3: Aplicaciones de tres capas

La capa intermedia (también conocida como servidor de aplicaciones) centraliza la lógica de las aplicaciones, haciendo que la administración de cambios sea más sencilla. En arquitecturas más simples, cualquier cambio en la lógica, implica una mayor afectación en las aplicaciones que dependan de ésta. En algunas ocasiones, -como en las aplicaciones Web- el uso de esta capa intermedia se divide en dos o más unidades con diferentes funciones, en estos casos la arquitectura es referida como multinivel. En la figura 1.3 se muestra este caso.

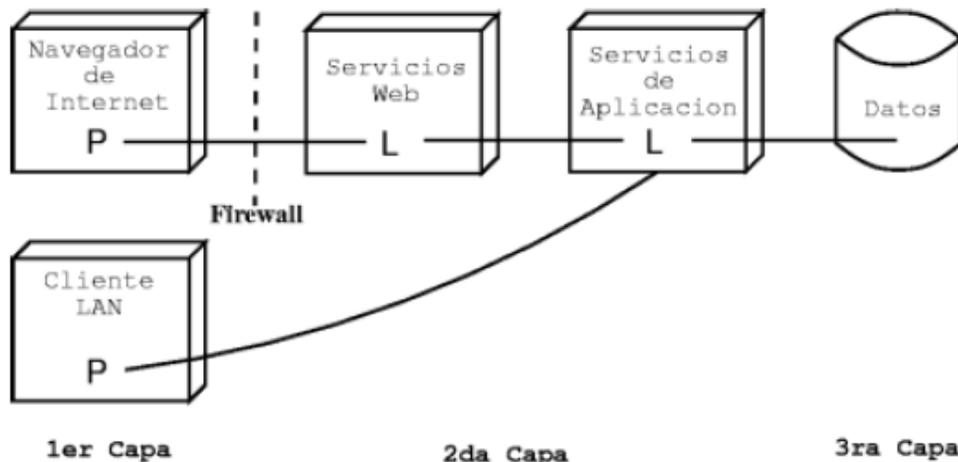


Figura 1.4: Arquitectura multinivel

1.4 Arquitecturas de los servidores de aplicaciones Web

Esta sección describe, las diferentes tecnologías que se han adoptado para que un servidor HTTP ejecute la aplicación Web, cada una con sus ventajas y desventajas. Se comenzará con el estándar Common Gateway Interface (CGI), que es fundamento de las siguientes; hablaremos sobre el modelo de co-servidor, donde un nuevo servicio se levanta al igual que el servidor HTTP y se comunica con este para transmitirse tanto las peticiones como las respuestas. Finalmente se verá el modelo de un lenguaje intérprete incrustado de manera persistente en el mismo servidor HTTP, por lo tanto el mismo proceso de servidor HTTP se encarga de ejecutar la aplicación Web.



1.4.1 CGI

El CGI (Interfaz Común de Puerta de Enlace) es el mecanismo original para crear interfaces entre aplicaciones externas y servidores de información, tales como los servidores HTTP. Un documento HTML plano enviado por un servidor HTTP a un cliente es estático, el texto que contiene esta en un estado constante, no cambia. Por el contrario, un programa CGI se ejecuta en tiempo real, por lo tanto puede recibir información del usuario, procesarla y arrojar datos formateados en HTML, texto, o algún otro formato plano o binario con base a esta información. De ahí que los CGI pueden producir información de manera dinámica.

El CGI provee de una manera consistente para pasar datos a partir de la solicitud de un agente del usuario a un programa que puede ser localizado y ejecutado por el servidor de HTTP.

- Un programa CGI puede ser escrito en cualquier lenguaje que se pueda compilar y/o ejecutar en el sistema donde se implemente, generalmente en C, Perl, Tcl o Algún Shell de Unix.

Un CGI puede adquirir los datos de entrada por dos medios: Por medio de variables de entorno (insertadas por el servidor HTTP al momento de generar el nuevo proceso que ejecutará el programa); y por medio de la entrada estándar, este flujo de información puede estar formateada en un estilo MIME⁶ y debe ser analizada sintácticamente para dividirla en sus distintas variables. Cuando los datos se reciben por la entrada estándar, es por que la información se envió por un método POST o PUT del HTTP.

Como los CGI son programas ejecutables, para que cumplan su función, necesitamos permitir que se ejecuten en el servidor, lo cual no es muy seguro, ya que cualquier error en el CGI podría desembocar en un agujero de seguridad⁷.

Por otro lado cada ejecución de un CGI implica que el servidor HTTP genere un nuevo proceso en el sistema, el cual consume recursos del sistema. El número de solicitudes a un CGI puede llegar a ser tal que consuma todos los recursos y puede desde ralentizar sensiblemente el desempeño del sistema, hasta bloquearlo por completo.

Por esta última desventaja es principalmente por lo que se han desarrollado alternativas al modelo del CGI, para mejorar el desempeño y la escalabilidad de las aplicaciones Web.

⁶ Multi-Purpose Internet Mail Extensions (Extensiones Multipropósito para el Correo de Internet) es una extensión al protocolo original de correo por Internet, que permite usar el protocolo para intercambiar diferentes tipos de archivos de datos: audio, vídeo, imágenes, programas de aplicación, etc.

⁷ Los agujeros de seguridad se pueden definir como puntos vulnerables de un sistema de cómputo, que pueden ser explotables para ganar accesos no autorizados al sistema.



Esta API generalmente se encuentra en C/C++, para muchos programadores no resulta productivo desarrollar programas de aplicación en tales lenguajes, ya que son difíciles y tardados de programar y depurar, así como la carencia de herramientas intuitivas para un desarrollo rápido. Es por ello que se han desarrollado capas intermedias de software donde la API del servidor HTTP se mapea a otro lenguaje para que sea más sencilla su utilización. Estos lenguajes suelen ser lenguajes interpretados tales como Visual Basic, Perl, Python, Tcl, etc. Aunque también existe, en estas capas intermedias de software, mapeos a lenguajes con un mayor grado de abstracción, que permite el procesamiento de plantillas con código incrustado dentro del HTML, tales como el PHP o el ASP.

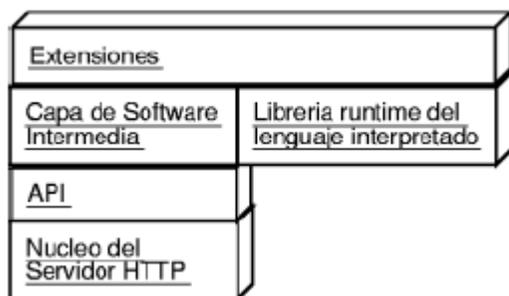


Figura 1.6: Estructura del Intérprete Incrustado

1.5 El protocolo HTTP

El Protocolo de Transferencia de Hipertexto¹⁰ (HyperText Transfer Protocol) es el protocolo base de la Web. Es un protocolo orientado a conexión, basado en el esquema solicitud/respuesta. El servidor envía una respuesta al cliente basado en la solicitud hecha por éste, cerrando inmediatamente la conexión entre ambos; por ello se le conoce como un protocolo sin manejo de estados¹¹

Se dice que está orientado a conexión porque emplea para su funcionamiento un protocolo de comunicaciones TCP (Transport Control Protocol) por el que pasa el flujo de bytes que constituyen los datos que hay que transferir.

Este protocolo puede ser utilizado para varias tareas más allá de su uso para la transferencia de hipertexto, tal como sistemas para la administración de objetos distribuidos y colaborativos de hipermedios¹².

¹⁰ Hipertexto es la organización de unidades de información en asociaciones conectadas que un usuario puede escoger.

¹¹ Se dice que un protocolo de red puede manejar un estado si mantiene un conjunto de variables entre las múltiples conexiones de un mismo cliente, tal como el FTP.

¹² Liga de hipertexto para incluir a un conjunto de objetos multimedia, incluyendo sonido, vídeo y realidad virtual.



En el HTTP las comunicaciones se inician generalmente por un Agente del Usuario¹³ y básicamente consiste en que el usuario establece la comunicación con el servidor y envía un mensaje con los datos de la solicitud, el servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. En el caso más simple, puede ser realizada a través de una sola conexión entre el usuario y el servidor origen. Una situación más complicada ocurre cuando uno o más intermediarios están presentes en la cadena de solicitud/ respuesta.

1.5.1 Mecanismo para el manejo de estados en el HTTP

Cuando se hace una solicitud a un servidor HTTP, este responde a cada solicitud del cliente sin relacionar tal solicitud con alguna previa o subsecuente. Para proveer al protocolo http de un “estado” la empresa Netscape introdujo el concepto de *cookie*, el cual es un grupo de cabeceras que se agrega a las solicitudes de los clientes y a las respuestas de los servidores.

Este mecanismo permite que clientes y servidores puedan intercambiar información sobre el estado y colocar las solicitudes y respuestas del HTTP dentro de un contexto más largo, lo cual llamaremos *sesión*. El diseño del mecanismo para el manejo de estados a través del intercambio de *cookies* tiene los siguientes atributos:

1. Cada sesión tiene un inicio y un fin.
2. Cada sesión tiene un tiempo de vida relativamente corto.
3. Ya sea el agente del usuario o el servidor original puede terminar una sesión.
4. La sesión está implícita en el intercambio de la información del estado.

1.6 Los lenguajes de marcado

Los lenguajes de marcado surgen de la necesidad de crear documentos estandarizados, es decir, poder visualizar un documento sin importar la plataforma que se usará, así nace, a principios de los ochentas, el GML (Generalized Markup Lenguaje) el cual define que cada etiqueta¹⁴ sirve para describir el formato e indicar el contenido, es decir, las etiquetas son un conjunto reglas que definen las partes que conforman un documento, pero que no pertenecen al texto en sí, una etiqueta solo indica el principio de un elemento y otra el final del mismo.

Ante el buen comienzo de este lenguaje, aparece SGML¹⁵ que es una norma ISO de estandarización derivada del GML, este parecía la respuesta al problema de estandarización de documentos, pero no se popularizó debido a que era muy complicado para su manejo,

¹³ Agente del usuario: Programa de computadora o dispositivo electrónico que es usado para acceder a los servicios ofrecidos por una aplicación Web. Éste puede ser un navegador de Internet, un teléfono celular habilitado para acceso a Internet, un software convertidor de texto en audio, entre otros.

¹⁴ Una etiqueta (tag) es un término genérico que designa a un elemento descriptor de un lenguaje.

¹⁵ Standard Generalized Markup Language (Lenguaje Estándar de Marcado Generalizado)



ocasionando que las herramientas para la creación de SGML fueran caras y no tuviera gran aceptación

El HTML creado por Tim Berners-Lee, nace con el objetivo de ser un lenguaje para publicar información científica y distribuirla globalmente, reduce considerablemente el problema de la complejidad del SGML, especificando un pequeño conjunto de etiquetas estructurales y semánticas para escribir documentos relativamente simples.

Con el tiempo se notó que HTML no estaba preparado para soportar las nuevas tecnologías¹⁶ que iban apareciendo en la Web, así, en 1998 aparece el XML¹⁷ por la necesidad de una Web realmente interactiva, para hacer frente a las nuevas problemáticas que vinieron con las nuevas tecnologías de Internet y de la intención de recuperar el poder y la flexibilidad del SGML sin la mayor parte de su complejidad, el XML preserva la mayor parte de su poder y riqueza, y sigue manteniendo todas las características más comúnmente utilizadas.

Una de las principales innovaciones de XML, a diferencia de HTML es que no hay etiquetas predefinidas, cada usuario puede crear sus propias etiquetas del lenguaje según sus necesidades, otra diferencia es que XML es puramente estructural para dar formato a los datos se debe utilizar un auxiliar como CSS o XLS.

Pronto XML resultó complicado para los usuarios comunes y también se encontró con problemas de compatibilidad con los navegadores, por tanto, se decidió crear un lenguaje de transición que ofreciera a los usuarios acercarse al nuevo lenguaje y por otra parte aprovechar las noblezas de HTML, fue entonces cuando aparece el XHTML, que es una familia de tipos de documentos actuales y futuros, módulos que reproducen, subconjuntan y extienden al HTML 4. Los tipos de documentos de la familia XHTML están basados en XML¹⁸, y diseñados para trabajar en conjunto con agentes de usuario basados en XML.

El XHTML es una reformulación de los tres tipos de documentos definidos por HTML 4 (strict, loose y frameset) como aplicaciones XML. Tiene la intención de ser usado como un lenguaje para contenido que se ajuste al XML y que opere en un agente de usuario que cumpla el HTML 4.

1.6.1 Las hojas de estilo CSS

La naturaleza que derivó al SGML hasta llegar al XHTML es que la información descrita en éste lenguaje pudiera ser desplegada de manera independiente del dispositivo y plataforma sin que se perdiera su estructura original, sin embargo con el éxito alcanzado a mediados de los noventas se comenzó a integrar nuevas características al lenguaje de

¹⁶ Terminales Braille o UNIX, teléfonos móviles, WebTV, navegadores de texto o de voz, etc.

¹⁷ Extensible Markup Language (Lenguaje de Marcado Extensible).

¹⁸ Se dice que XHTML esta basado en XML porque XML es un metalenguaje, es decir, un lenguaje hecho para poder construir otros lenguajes con él



manera que los autores de documentos HTML pudieran explotar las capacidades gráficas y multimedia de los navegadores.

Esto hizo que se comenzara a perder el enfoque inicial del HTML, hasta que el World Wide Web Consortium (W3C) desarrolló el concepto de Hoja de Estilo en Cascada (CSS por sus siglas en inglés). Este lenguaje es un mecanismo de definición de estilo, que permite a los autores de documentos HTML y XML adjuntar un estilo a éstos; por ejemplo, tipo de letra, color, espaciado, etc.

Así los autores pueden adjuntar su hoja de estilo preferido, mientras que los lectores pueden tener su hoja de estilo personal que se ajuste a su discapacidad técnica o humana.

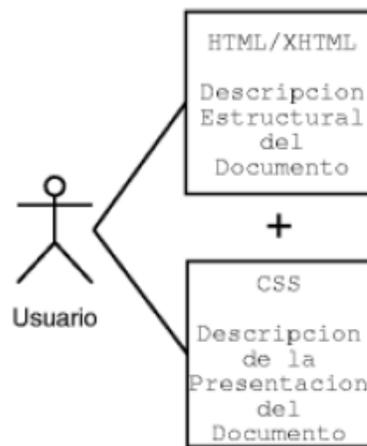


Figura 1.7: HTML + CSS

Otra característica interesante de las CSS es que puede definir objetos que indican comportamientos genéricos, que pueden ser aplicados a distintas estructuras del HTML/XHTML. Además estos objetos pueden ser sobrecargados y heredados, siguiendo un modelo orientado a objetos.

1.6.2 Scripts de lado del cliente

Los scripts del lado del cliente son programas que acompañan a un documento HTML o esta escrito dentro de el directamente¹⁹. El programa se ejecuta en la máquina del cliente cuando el documento se carga, o en algún otro momento cuando la liga es activada. HTML soporta el manejo de scripts independientemente del lenguaje (JavaScript, VBScript, TCL, etc.)

Los scripts ofrecen a los autores una forma de extender los documentos HTML en formas altamente activas e interactivas:

¹⁹ Esto no es muy recomendable por cuestiones de limpieza, mantenimiento y actualizaciones en el código



- Los scripts puede modificar el contenido de documento de manera dinámica.
- Pueden procesar el contenido de una forma HTML.
- Pueden ser ejecutados al dispararse un evento específico tal como la carga y descarga de elementos, movimientos del ratón, etc.
- Pueden ligar controles de formas (botones por ejemplo) para producir elementos de la Interfaz Gráfica de Usuario (GUI por sus siglas en inglés)

Debido a que cada empresa que tenia su navegador creo un script los creadores de los navegadores crearon lenguajes de script diferentes esto llevo a provocar un problema de incompatibilidad de lenguajes soportados por los navegadores, ECMA²⁰ desarrolló el estándar ECMAScript, basado en los lenguajes JavaScript y JScript.

El ECMAScript es un lenguaje de programación orientado a objetos para realizar cálculos y manipular objetos computacionales dentro de un ambiente anfitrión (Navegador). El ECMAScript no tiene la intención de ser computacionalmente autosuficiente; depende de los objetos provistos por el sistema anfitrión.

1.7 El lenguaje SQL

El Structured Query Language (SQL) es un lenguaje de consultas estándar para obtener y actualizar información de una base de datos, además es útil para manipulación de los datos sobre los cuales se trabajará.

SQL fue antecedido por un sublenguaje de acceso a los datos basado en el cálculo de predicados desarrollado en la década de los setenta por investigador de la IBM, E. F. Codd al cual se le llamo SEQUEL (Structured English QUery Language). Tomando esta idea a mediados de los ochentas tanto el American National Standards Institute (ANSI) como la International Organization for Standarization (ISO) desarrollaron un lenguaje de base de datos relacional estándar, basándose en los conceptos del SQL. Actualmente la última revisión del estándar se conoce como SQL3, ocurrida en 1999.

A pesar de estos esfuerzos de estandarización, diversos manejadores de base de datos soportan SQL pero añaden extensiones propietarias al lenguaje por lo que las migraciones entre manejadores distintos suelen ser difíciles.

En SQL las consultas toman la forma de un lenguaje de comandos en idioma inglés, que permiten realizar selecciones, inserciones, actualizaciones y búsquedas de datos.

²⁰ ECMA es una asociación industrial internacional dedicada a la estandarización de sistemas de información y comunicación., Actualmente está aceptado como el estándar ISO 16262.



1.8 Cifrado de información

Todas las comunicaciones que se establecen a través de Internet utilizan el protocolo TCP/IP²¹, este permite que la información sea enviada de una computadora a otra a través de una serie de computadoras intermedias y de redes distintas antes de alcanzar su destino.

La gran flexibilidad del TCP/IP ha conducido a su aceptación como el protocolo de comunicación básico para Internet e Intranets. Pero tiene la desventaja que por el hecho de permitir el paso de información a través de computadoras intermedias, se hace posible que terceras personas interfieran en las comunicaciones.

Para proteger la información durante este viaje entre computadoras se han desarrollado algunas soluciones dentro de las que destacan las siguientes:

- **Algoritmos criptográficos:** Existen dos clases de algoritmos de cifrado basados en el uso de llaves: algoritmos **simétricos** (o de llave secreta) y **asimétricos** (o de llave pública). La diferencia es que los algoritmos simétricos usan la misma llave tanto para el encriptado²² como para el desencriptado en tanto que los algoritmos asimétricos usan una llave diferente para el encriptado y desencriptado. Los métodos de cifrado asimétricos (también llamados algoritmos de llave pública o generalmente criptografía de llave pública) permiten que la llave de encriptado sea pública (puede ser publicada incluso en un periódico), permitiendo que cualquiera encripte con esa llave, donde únicamente el propio receptor (quien conoce la llave privada de desencriptación) puede desencriptar el mensaje. La llave de encriptado también se le llama llave pública y la llave de desencriptado llave privada o llave secreta.

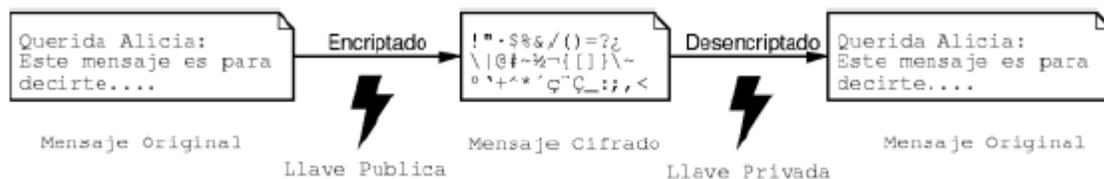


Figura 1.8: Cifrado por llave pública

- **Firmas digitales:** Una firma digital es una pequeña cantidad de información que fue creada usando alguna llave secreta. Las firmas digitales son usadas para verificar que un mensaje realmente proviene del remitente declarado (asumiendo que el emisor conoce la correspondiente llave privada para su llave pública).

²¹ Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisiones/Protocolo de Internet).

²² aunque no existe propiamente en español la palabra encriptar, en informática se ha convertido en sinónimo de la palabra cifrar.



También las firmas digitales pueden ser utilizadas para certificar que una llave pública pertenece a una persona en particular. Esto se logra usando la combinación de la llave y la información sobre su dueño por una llave que sea de confianza. La firma digital por una tercera persona (dueño de la llave de confianza), la llave pública y la información sobre el dueño de la llave pública son llamados **certificados**.

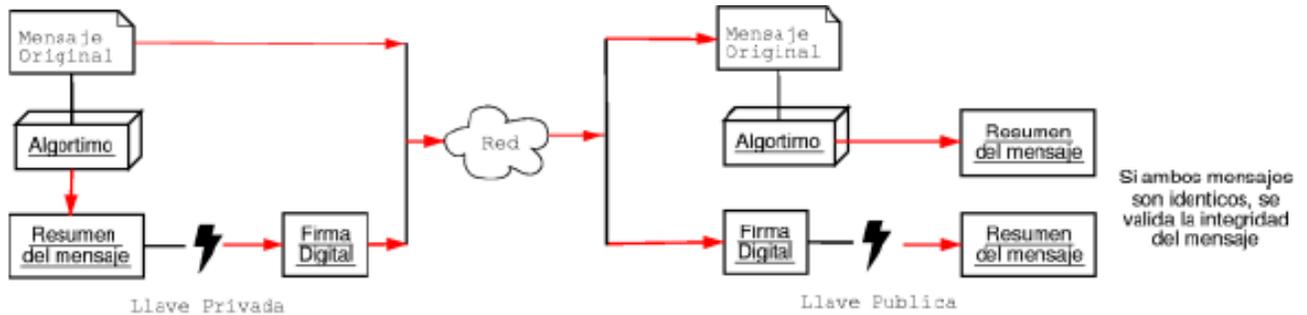


Figura 1.9: Uso de una forma digital para verificar la integridad de los datos

- **El protocolo SSL** Trabaja sobre el TCP/IP y debajo de los protocolos de alto nivel tales como el HTTP o el IMAP²³. Usa el TCP/IP en nombre de los protocolos de alto nivel, y en el proceso permite a un servidor SSL confirmar la identidad por sí mismo de un cliente SSL, permite al cliente autenticar por sí mismo al servidor, y permite a ambas máquinas establecer una conexión encriptada protegida con un mecanismo para detectar si la información fue alterada durante su tránsito en la red.

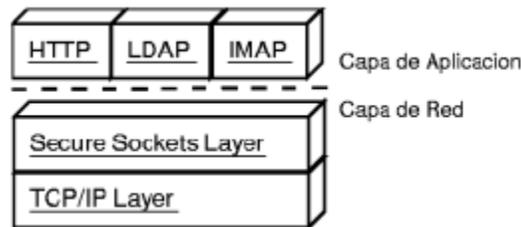


Figura 1.10: Protocolo SSL

²³ (Internet Message Access Protocol). Protocolo de red de acceso a mensajes electrónicos almacenados en un servidor



Capítulo 2

Componentes de aplicaciones Web con uso de bases de datos.

En este capítulo se enumeran algunos enfoques que han surgido para la programación para Web, los cuales tienen diferentes mecanismos y metodologías para realizar este tipo de desarrollos.

Se toman algunos de los problemas o situaciones con los que tiene que tratar el desarrollador de aplicaciones Web. Pero antes es necesario conocer cómo es que funciona una aplicación Web. En la figura 2.1 se observa un diagrama que muestra la estructura general de una aplicación Web. En primera instancia, lo que ve un usuario al ingresar a un sitio que alberga una aplicación Web es la página de contenido inicial, ésta es la página de bienvenida, generalmente en ella existe un espacio dedicado a la autenticación²⁴ del usuario para su ingreso a áreas restringidas del sistema.

Una vez que el usuario es autenticado, la aplicación Web debe de poner en marcha un mecanismo que le permita determinar, en todo momento, cuál es el estatus del usuario, es decir, en qué parte de la aplicación se encuentra, qué variables se han establecido, etc. Lo anterior se logra con el manejo de sesión. A partir de este punto, todas las acciones o peticiones del usuario serán enviadas al despachador de mensajes, el cual se encarga de validarlas y, si aplica, mandar ejecutar el procedimiento correspondiente al mensaje de solicitud y al estado encontrado.

²⁴ Proceso para determinar si algo o alguien es realmente quien declara que es. Este proceso es comúnmente implementado con el uso de un nombre de usuario y una contraseña secreta. Si la entidad que se desea autenticar conoce la contraseña, se asume que es el realmente quien dice ser y se le otorga acceso a los recursos del sistema definidos por el administrador.



Las operaciones que se realizan en una aplicación Web (el almacenamiento de los correos de un usuario o la lista de artículos de un pedido a un almacén por ejemplo) requieren de un medio de almacenamiento persistente, comúnmente esto es realizado por medio de un Sistema Manejador de Bases de Datos (DBMS²⁵); la aplicación Web debe de tener un medio para comunicarse con el DBMS con el fin de leer y escribir la información en la base de datos.

Un aspecto importante en el proceso de desarrollo de software es la forma en la que se ataca el problema, para ello, el analista del sistema debe conocer los métodos y técnicas que le servirán para llevar el procedimiento actual a una mejora, por esta razón se menciona el Lenguaje Unificado de Modelado (UML²⁶) que es considerado como el estándar en modelado de sistemas de información.

Por otra parte esta el diseño en el cual una de las partes más importantes de una aplicación Web es la presentación. Las plantillas son una serie de archivos que definen la apariencia de una aplicación Web, de esta manera, si se desea cambiar la presentación de la aplicación, sólo se tiene que modificar las plantillas sin necesidad de modificar el código que le da funcionalidad a la aplicación.

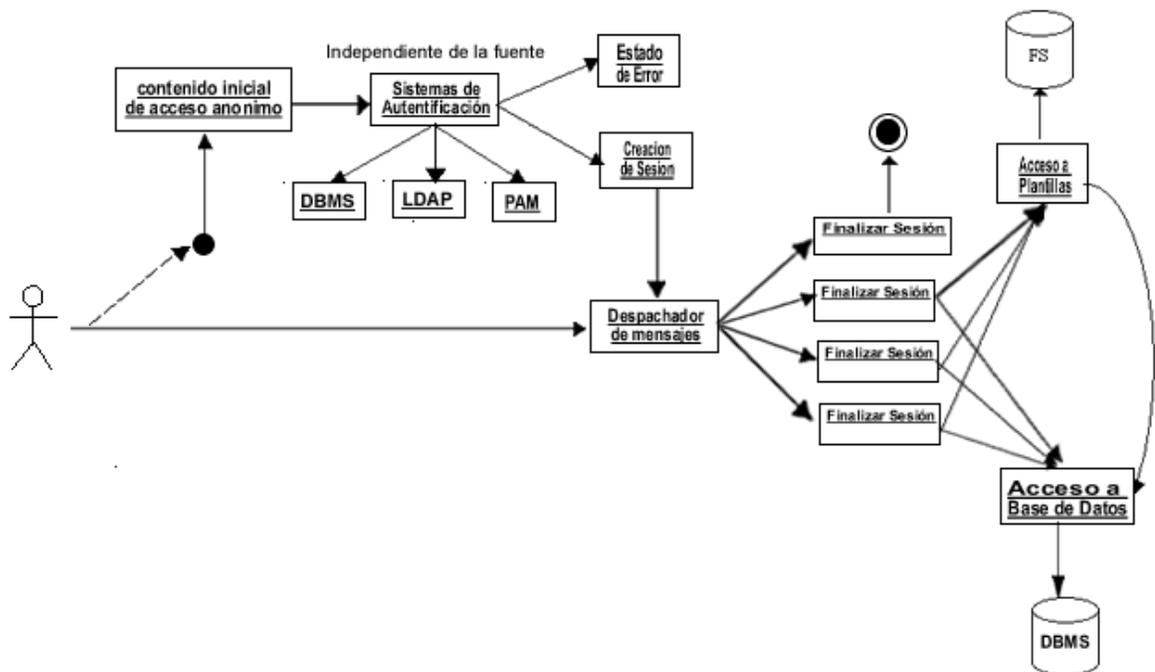


Figura 2.1: Estructura General de una Aplicación Web

2.1 Modelos para la programación de aplicaciones Web

²⁵ Data Base Managment System

²⁶ Unified Modeling Language



Existen varios modelos de programación para aplicaciones Web, de los cuales a continuación haremos un breve análisis comparativo de las características más importantes de cada uno de ellos e incluyo algunos ejemplos para que se pueda apreciar gráficamente

2.2.1 CGI nativo

Comúnmente cuando se refiere a un CGI, se hace referencia a un programa que utiliza la interfaz propuesta por el estándar CGI. Estos programas normalmente son pequeños y son resultado de una solución rápida a algún planteamiento de automatización, ya que como interactúan con el estándar a bajo nivel, cada programa debe implementar sus propias herramientas para el procesamiento tanto de los datos de entrada como de salida.

Algorithm 1 Programa CGI en lenguaje C

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char **argv)
{
    char *your_add;
    your_add = getenv("REMOTE_ADDR");
    printf("Content-type: text/html%c%c", 10, 10);
    printf("<html>");
    printf("<head>");
    printf("<title>Hola Mundo</title>");
    printf("</head>");
    printf("<body bgcolor=\`#FFFFFF\`>");
    printf("<h1 align=\`center\`>Hola Mundo</h1>");
    printf("<p align=\`center\`>Tu IP es: %s", your_add);
    printf("</body>");
    printf("</html>");
}
```

2.1.2 Servlets

Los Servlets son módulos de código, normalmente en lenguaje Java, pero pueden encontrarse en algún otro como Python, que se ejecutan dentro de un servidor de aplicaciones (motor de Servlets) para responder a las peticiones de los sistemas cliente.

Algorithm 2 Servlet en lenguaje Java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloClientServlet extends HttpServlet
{
    protected void doGet(HttpServletRequest req,
        HttpServletResponse res)
        throws ServletException, IOException
    {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.println("<HTML><HEAD><TITLE>Hello Client!</TITLE>" +
            "</HEAD><BODY>Hello Client!</BODY></HTML>");
        out.close();
    }
    public String getServletInfo() {
        return "HelloClientServlet 1.0";
    }
}
```



2.1.3 Módulos del servidor Web

Existen módulos que utilizan el API de los servidores Web para extender su funcionamiento. Estos módulos pueden hacer uso de los manejadores de contenido del servidor HTTP y convertirse así en aplicaciones Web.

Cómo ya se comentó existen integraciones entre lenguajes interpretados y servidores HTTP para simplificar la programación de estos módulos.

Algorithm 3 Módulo para Apache en mod_perl

```
Package HolaMundo;
Use Apache::Constants ':common';
Sub handler {
    my $r = shift;
    $r->content_type('text/html');
    $r->no_cache(1);
    $r->send_http_header;
    $r->print("<HTML>");
    $r->print("<HEAD>");
    $r->print("<TITLE>Hola Mundo</TITLE>");
    $r->print("</HEAD>");
    $r->print("<BODY><H1>Hola Mundo</H1>");
    my $c = $r->connection;
    $r->print("<P>Tu IP es: $c->remote_ip</P>");
    $r->print("</BODY>");
    $r->print("</HTML>");
    return OK;
}
```

Las desventajas en este modelo está en la complejidad en el desarrollo de extensiones directas al servidor HTTP: nos comunicamos directamente con éste, y por consecuencia el nivel de abstracción que tenemos es de muy bajo nivel, en comparación con el siguiente modelo.

2.1.4 Código incrustado en el HTML

En este modelo el código ejecutable está dentro de un documento HTML, de esta manera, el servidor, al recibir una petición, ejecuta el código incrustado y produce un documento en código HTML y es enviado al agente del usuario que realizó la petición. A continuación se muestra un breve ejemplo de como se incrusta código ejecutable dentro de un documento HTML. Dicho de otra manera, en los modelos anteriores, el desarrollador de la aplicación Web escribía código en lenguaje de programación, que como salida, producía HTML. Esta tarea llega a ser laboriosa y muchas veces aburrida. Por lo cual se definió un nuevo modelo, donde el desarrollador no se fija de manera única en el código en algún lenguaje de programación, sino que se dedica a escribir HTML y únicamente donde se deben colocar elementos dinámicos, el desarrollador utiliza marcas que definen el sistema del intérprete incrustado utilizado, y escribe código.

Algorithm 4 Ejemplo de un ASP



```
<%@Language = VBScript %>
<html>
<body>
<% For i = 3 to 7 %>
  <FONT SIZE = <% = i %> >
  Hola Mundo! <br>
<% Next %>
</body>
</html>
```

Es posible generar páginas de código incrustado en el HTML con varios lenguajes:

- VBScript (las Active Server Pages -ASP- de Microsoft)
- Perl (Apache: ASP, Mason, ePerl, etc.)
- Java (JSP)
- PHP

Las ventajas de este modelo de desarrollo radican en que no se necesita mucha formación técnica para escribir una aplicación Web. De la misma manera en que se escribe una página Web, se desarrolla una aplicación Web. Actualmente los lenguajes y ambientes de desarrollo para este modelo proveen de componentes reutilizables muy potentes que ofrecen mayor productividad al programador.

Sin embargo, por su misma naturaleza, su uso se limita a la presentación de documentos basados en texto (HTML, XML, etc.), además de la total flexibilidad que nos ofrecería un lenguaje de programación de propósito general.

2.2 Mecanismos para conocer el estado del usuario

2.2.1 Validación de datos

En el momento que un programa interactúa con un cliente conectado a través de una red, existe la posibilidad de que ese cliente ataque el programa para obtener un acceso no autorizado al sistema. Hasta el más inocente programa puede ser muy peligroso para la integridad del sistema.

Una aplicación Web, por su naturaleza, permite la lectura de archivos, la adquisición de accesos a programas del sistema, etc. Estos accesos pueden ser utilizados de manera no autorizada explotando las:

- Suposiciones del programa
- Debilidades en la configuración del servidor
- Debilidades en otros programas y llamadas al sistema



Muchos autores concedores de la seguridad en Internet afirman que la principal debilidad en las aplicaciones Web es la insuficiente validación de los datos de entrada.

Esto quiere decir que para cada dato que el usuario a mandado a la aplicación Web, antes de procesarla debemos tener control sobre el; es decir, sanear la aplicación de datos no autorizados. Jamás se deben hacer suposiciones acerca de esta información.

El CERN²⁷ aconseja seguir la regla número uno en la construcción de *firewalls* pero aplicada a la programación: “lo que no está expresamente permitido, está prohibido”. Es decir, definir un dominio de valores aceptables por cada elemento de información recibida (variable) y bajo esta verificación reemplazar cualquier caracter no aceptable por algún otro de control, como el guión bajo (“_”) o su completa eliminación. No se debe hacer al revés (identificando lo ilegal y escribir código para rechazar dichos casos) porque es muy probable olvidar manejar algún caso importante de entrada ilegal.

Este saneado de los datos de entrada debe ser tanto léxico como sintáctico, haciendo que ésta se ajuste a lo que se debe procesar. El uso de expresiones regulares²⁸ se vuelve fundamental en esta tarea, ya que éstas nos permiten crear esqueletos con los cuales deben coincidir los datos a procesar, en caso contrario se rechazan o se sanear. Además estas pueden ser reutilizables tanto en el lado del servidor como en el lado de cliente ya que son independientes del lenguaje²⁹. Así mismo es importante limitar la longitud máxima (y si es posible mínima) de cada variable.

La validación puede ser hecha tanto de lado del agente del usuario (si y sólo si éste provee de herramientas para ello como lenguajes script de lado del cliente) y naturalmente en el lado del servidor. Es en este punto donde viene la pregunta ¿dónde debemos de validar?

La validación del lado del servidor asegura que, independientemente del agente del usuario utilizado por el cliente, los datos recibidos serán aceptados únicamente si pasan por el filtro de validación, además de los provistos por los DBMS cuando los datos son serializados. Por el otro lado, la validación en el lado del cliente hace que la aplicación sea más amigable para el usuario, y no tenga que esperar la respuesta del servidor con mensaje de error, para volver a reescribir los datos; sin embargo no se puede saber qué agente del usuario utilizará quien acceda al sistema, y por lo tanto no sabremos con qué capacidades de procesamiento contará.

²⁷ Originalmente de las siglas Computer Emergency Response Team (Equipo para la Respuesta de Emergencias Computacionales). Actualmente es el mayor informador de problemas de seguridad en Internet.

²⁸ Una expresión regular es una manera por la cual un usuario puede decirle a un programa de computadora cómo buscar determinado patrón en una cadena de texto. El usuario puede determinar que hacer cuando determinado patrón se encuentre, tal como eliminarlo o sustituirlo por algún otro.

²⁹ Varios lenguajes de programación proveen de procesadores de expresiones regulares: Perl, JavaScript, Python, VBScript, entre otros.



Es por las anteriores razones que creemos que se debe procurar realizar la validación en ambos lados, para asegurar la confiabilidad de los datos procesados, sin sacrificar la facilidad y amigabilidad de la aplicación. Aunque esto implica la posibilidad de programar más. Se acepta que por norma el 80% del código escrito tiene como objetivo la validación de datos y el manejo de estados de excepción, mientras que el 20% restante maneja el funcionamiento de la aplicación.

2.2.2 Autenticación, Autorización de usuarios y Sesiones

La manera predeterminada de trabajar del WWW es anónima, lo único que se puede saber, y no siempre con toda seguridad, es el número IP que accede a un determinado recurso. Casi siempre, y debido a características de personalización, políticas de restricción, o ambas, las aplicaciones Web deben saber qué usuario las están utilizando; es por ello que se debe solicitar la identidad del usuario, comúnmente por medio de un nombre de usuario y una comprobación de la misma, a través de una palabra o frase secreta. Este proceso de identificación se le conoce como autenticación.

Por otro lado está la autorización, donde la aplicación que ha identificado al usuario que desea acceder a ella, ahora lo reconoce como usuario válido, así como sus restricciones en su uso. Para realizar la autorización y autenticación del usuario, la aplicación Web requiere de una fuente de datos que contenga la lista de usuarios y sus restricciones de uso. Esta información se encuentra normalmente en una base de datos y requiere que se accede a ella a través de algún método específico:

- PAM³⁰
- Radius/Tacacs
- SQL, ODBC, DBI, etc.
- API del sistema operativo
- etcétera

Existen múltiples esquemas para la autenticación de usuarios al momento de querer acceder a un recurso Web. El protocolo HTTP provee, de manera interna, un mecanismo para solicitar la autenticación de un usuario para su acceso al recurso. Finalmente los agentes del usuario o navegadores, se encargan de manejar esa petición HTTP, presentando un cuadro de diálogo solicitando el nombre de usuario y la contraseña. Sin embargo este enfoque presenta varias desventajas, entre ellas están la incapacidad de terminar la sesión hasta que se termine el proceso del agente del usuario, así como la incapacidad de modificar la presentación al usuario que solicita la autenticación, ya que este formulario es manejado por el agente del usuario autónomamente. Además, si se necesita utilizar algún otro método que no sea la utilización de archivos planos (htpasswd), es imprescindible la utilización de módulos externos agregados al servidor HTTP.

³⁰ Pluggable Authentication Modules (Módulos Enchufables de Autenticación)



La otra alternativa es que la aplicación Web se haga cargo de la autenticación, integrándose a la autorización del usuario y el mecanismo de sesiones. Así la presentación del formulario para la tarea se hace a través de formularios HTML, otorgando más flexibilidad para modificar el método de autenticación cuando se necesite.

El desarrollador tiene que establecer desde la etapa de diseño el método de autenticación que va a utilizar y que le permita la escalabilidad y flexibilidad necesaria.

Como ya se estableció en el punto 1.5, el protocolo HTTP no maneja un estado de cada conexión realizada por un agente del usuario, por tanto se deben establecer mecanismos complementarios al servidor de HTTP para llevar el control de la sesión.

Lo anterior se logra mediante el uso de algún esquema de almacenamiento persistente de una estructura de datos por cada cliente concurrente; este almacenamiento puede ser tanto en:

- Memoria RAM compartida
- Archivo plano o con registros del sistema de archivos
- Tabla en la base de datos utilizada
- Registros de DNS dinámico
- *Cookies*
- Controles HTML ocultos

Cada uno de los anteriores tiene sus ventajas y desventajas. Por ejemplo, la opción de utilizar memoria RAM es muy compleja; además la memoria RAM es un recurso muy valioso y muy utilizado por las aplicaciones Web, ya que cada acceso concurrente tendrá sus propias instanciaciones de algunos procesos y variables, por lo cual el consumo de memoria RAM puede incrementarse altamente; el manejo de archivos para el seguimiento de sesiones complica la confiabilidad de los datos ya que el sistema de archivos no provee de los mecanismos apropiados para asegurar la integridad de los datos y el rápido acceso a éstos; y así sucesivamente. Las *cookies* pueden ser leídas por terceros lo cual compromete la seguridad de la aplicación y los controles HTML ocultos se vuelven más complicados de mantener cuando la información persistente crece en tamaño. Pocas aplicaciones Web hacen uso exclusivo de un tipo de almacenamiento persistente, generalmente se prefiere mezclarlos, obteniendo las ventajas de cada uno y tratando de evitar sus desventajas.

Para que la aplicación Web identifique cada petición HTTP, las peticiones deberán contener de manera intrínseca un identificador. Esto se logra pasando ese identificador a través de cualquiera de los siguientes métodos:

- URI
- Parámetro por método GET ó POST
- *Cookie*



De esta manera se evita que el usuario se autentique y autorice en cada petición, así como la retransmisión constante de su nombre de usuario y su contraseña, lo cual es potencialmente peligroso. Debe recalarse que los identificadores de la sesión deben de ser únicos y difíciles de adivinar, ya que puede existir la posibilidad de que agentes externos quieran simular accesos válidos y entrar de manera fraudulenta al sistema. Es por ello que se debe de valer de algún mecanismo que provea de identificadores aleatorios y con un gran periodo en su repetición, tales como generadores de cadenas aleatorias (MD5, DES, etc.) o la utilización del dispositivo de entropía existente en algunos sistemas operativos.

La estructura persistente que se encuentra en el lado del servidor, deberá almacenar únicamente la información necesaria para llevar el seguimiento de la sesión, como identificador de la sesión, identificador del usuario en sesión, tiempo de expiración de la sesión, dirección donde se encuentra localizado el cliente³¹, y otras variables que deberán mantenerse durante la sesión así como variables temporales que posteriormente se serializarán en la base de datos en una transacción atómica.

Una última alternativa, aplicada en los modelos de co-servidor, consiste en generar un servicio alternativo al del servidor HTTP y que pueda comunicarse con él; este proceso alternativo generará ya sea un proceso o un hilo para cada cliente concurrente y será el encargado de controlar su sesión.

Así mismo hay que hacer notar que por la misma naturaleza del HTTP es imposible asegurar la existencia o la ausencia de una sesión, por lo tanto debemos de establecer un proceso que revise periódicamente los tiempos de expiración de cada proceso y los elimine de la base de datos si ya excedió de lo establecido.

2.2.3 Despachador de mensajes

Desarrollar una aplicación a través de una conexión sin estado con el usuario va en contra del paradigma habitual del desarrollo de aplicaciones, ya que el programador está acostumbrado a que el proceso que se crea, al correr alguna aplicación, se mantiene en el sistema hasta que el usuario decida terminar o suceda algún evento de excepción que no se puede controlar (como un fallo en el sistema operativo, en el hardware, etc.). Cualquier programa puede visualizarse como una máquina de estado finito, donde el procesador brinca de una línea de código a otra, dependiendo de su lógica. En una aplicación Web, después que se generó una respuesta a la petición del cliente, el servidor HTTP pierde la identificación del cliente hasta su siguiente petición, es por ello que resulta complicado conocer su estado actual en el sistema.

³¹ Esta variable resulta controvertida ya que actualmente, por la escasez de direcciones IP en Internet, se han construido NAT (Network Address Translators) y servidores Proxy, los cuales comparten su dirección IP común con la LAN a la que sirven, por lo tanto varios clientes pueden estar en una misma dirección y nuestro esquema podría fallar si tomamos esta variable como discriminante.



Por ello la manera habitual con la cual se lleva el control de la máquina de estado finito es a través de la transmisión de mensajes entre el cliente y el servidor, donde el cliente avisa al servidor en qué estado se encuentra y el servidor le indica a qué estado ha pasado. Por lo tanto, a cada solicitud HTTP el cliente envía un mensaje al servidor y el servidor le responde con una actualización o con un seguimiento de su estado actual.

Es por lo anterior que en la arquitectura que recomendamos se debe revisar, a cada solicitud HTTP, los mensajes que envía el cliente, los valide la aplicación Web, y en caso de no hallar excepciones, mande ejecutar los procedimientos correctos de su sesión. Esto se cristaliza en la codificación en un *despachador de mensajes*, que deberá ejecutarse a cada petición HTTP que llegue. Este *despachador* tiene la obligación de recibir los mensajes del cliente, ejecutar operaciones de validación y autenticación y posteriormente mandar ejecutar el procedimiento de acuerdo al mensaje solicitado y al estado encontrado.

La buena operación de este *despachador* es vital para el correcto funcionamiento de la aplicación Web, y cualquier falla o mala implantación del mismo, repercutiría en la estabilidad del sistema o hasta en la presencia de agujeros de seguridad que llevarían desde la corrupción de la integridad en la base de datos, hasta el mal uso y/o destrucción de la información.

2.3 El Proceso Unificado

El proceso unificado para el desarrollo de sistemas fue propuesto por Rational Software Corporation en 1998 después de tres décadas de desarrollo, adoptó el UML como lenguaje de modelado estándar.

Unified Process (UP) es un proceso de ingeniería de programación que busca asegurar la producción de software de alta calidad, satisfaciendo las necesidades del cliente, y con arreglo a un plan y presupuesto predecibles.

Sus características más importantes nos indican que es un proceso conducido por casos de uso, se centra en la arquitectura y es un proceso iterativo e incremental.

Los casos de uso constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas.

La arquitectura involucra los elementos más significativos del sistema, los casos de uso guían el desarrollo de la arquitectura y le permiten conceptualizar, gestionar y desarrollar adecuadamente el sistema.

El dividir en ciclos un proyecto es una manera de hacerlo más manejable, permitiendo para cada ciclo establecer fases de referencia, cada una de estas fases se considera como un mini



Los componentes del proceso de desarrollo establecen *qué* hay que hacer, *quién* debe hacerlo y *cómo* hacerlo. Cada componente lo podemos describir de la siguiente manera:

- Artefacto (*artifacts*), que representan cualquier tipo de información generada, modificada o utilizada en el desarrollo del sistema.
- Trabajadores (*workers*), que corresponden a los roles (una misma persona puede desempeñar varios roles) que intervienen en el componente.
- Flujos de trabajo (*workflows*) y actividades, que deben ser adelantadas por los trabajadores para obtener los artefactos del componente.

Los componentes del proceso de ingeniería son:

- **Modelado de la Organización.** Consiste en la identificación y documentación de la estructura y funcionamiento de la organización en la cual operará la aplicación a desarrollar. Su objetivo es brindar un entendimiento a clientes y desarrolladores sobre *cuál es el problema* de la organización, identificar mejoras potenciales y establecer el impacto que la aplicación a desarrollar tendría sobre la organización.
- **Captura de Requisitos.** Su propósito es obtener la descripción de *para qué sirve el sistema*, y lograr un acuerdo entre el equipo de desarrollo y el cliente en este aspecto.
- **Análisis.** En este componente se define la estructura (clases, paquetes, etc.) y comportamiento del sistema. Su propósito es obtener una descripción de *cómo funciona* el sistema.
- **Diseño.** Mientras que Análisis se ha centrado en establecer la funcionalidad del sistema, el componente de Diseño se enfoca a lograr que esa funcionalidad se haga posible sobre una arquitectura física (computadores, redes, etc.) y un entorno de implementación (sistemas operativos, lenguajes de programación, etc.) dados. Su propósito es obtener una descripción de *cómo se construye* el sistema.
- **Implementación.** Construcción del sistema obteniendo los *archivos* ejecutables, de configuración, librerías, etc.
- **Pruebas.** Se verifican los modelos, prototipos y demás artefactos ejecutables del sistema bajo desarrollo.
- **Puesta en Servicio.** En este componente se realizan las actividades requeridas para poner en funcionamiento el producto en las instalaciones del cliente.



Por otra parte, los componentes de soporte son tres:

- **Gestión de configuración y cambios.** Lleva control sobre la evolución iterativa del sistema, registrando las modificaciones de sus partes y las configuraciones que dan lugar a los prototipos operacionales.
- **Gestión del proyecto.** Define los aspectos específicos de un proceso de desarrollo iterativo. Para ello brinda un marco de razonamiento para la gerencia de proyectos intensivos en programación, junto con guías prácticas para la planificación, constitución de equipos de trabajo, ejecución y supervisión de proyectos, y criterios para el manejo de riesgos.
- **Entorno.** Su propósito es establecer la organización del entorno de desarrollo de programación (procesos y herramientas) requerida por el equipo de desarrollo.

La organización en el tiempo establece la dinámica del proceso de desarrollo, está expresada en términos de Ciclos, Fases, Iteraciones e Hitos:

- **Ciclo:** Desarrollo de una nueva versión del producto. Corresponde al ámbito de un proyecto.
- **Fases:** Etapas en el desarrollo de una versión (proyecto). Cada fase termina en un Hito y puede estar dividida en Iteraciones.
- **Hito:** Punto de control, donde generalmente se revisan los resultados del proceso y se decide si se avanza a la siguiente Fase o Iteración.
- **Iteración:** Unidad de desarrollo del producto, en la que se obtienen o refinan uno o más artefactos del sistema.

2.4 UML

UML es un lenguaje para modelar, construir y documentar los elementos que forman un software orientado a objetos. Se ha convertido en un estándar en la industria del software, debido al prestigio de sus creadores y a que incorpora las principales ventajas de cada uno de los métodos en los que se basa: Booch, OMT y OOSE³². Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado otras empresas como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

UML ha puesto fin a lo que algunos autores llaman “la guerra de métodos”, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás.

³² los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh



UML es una herramienta compartida entre los ingenieros software que trabajan en el desarrollo orientado a objetos en donde se fusiona la notación de estas técnicas

El objetivo principal de UML inicialmente era posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema

Algo muy importante a tener en cuenta es que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación.

2.4.1 Modelos y Diagramas

Un modelo representa a una aplicación software desde una perspectiva específica, es decir, nos muestran la misma figura vista desde distintos ángulos, cada modelo nos permite fijarnos en un aspecto distinto del sistema.

Un diagrama es una representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos

Los modelos de UML con los que vamos a trabajar son los siguientes:

- Diagrama de Casos de Uso.
- Diagrama de Secuencia.
- Diagrama de Colaboración.
- Diagrama de Estados.
- Diagrama de Clases.

Existen algunos libros que se refieren a Diagramas de Paquetes, Diagramas de Subsistemas y Diagramas de Modelos. Sin embargo, éstos corresponden a casos particulares de los diagramas arriba mencionados, cuando en éstos sólo se incluye paquetes (o subsistemas, o modelos, respectivamente).

2.4.2 Casos de Uso

El modelado de Casos de Uso es una técnica que nos ayuda a realizar el análisis de los requisitos del sistema desde la perspectiva del usuario quien es el centro de interés, es la técnica más fácil y efectiva para este objetivo. Los Casos de Uso se utilizan para modelar cómo un sistema o negocio funciona actualmente, o cómo los usuarios desean que funcione, es una forma de modelar procesos. Los casos de uso son generalmente el punto de partida del análisis orientado a objetos con UML.



El diagrama de casos de uso consiste en actores y casos de uso. Los actores representan usuarios y otros sistemas que interactúan con el sistema. Los casos de uso representan el comportamiento del Sistema.

Cada Caso de Uso puede estar definido por:

- Texto que lo describe
- Secuencia de pasos (flujo de eventos) ejecutados dentro del caso de uso
- Precondiciones y poscondiciones para que el caso de uso comience o termine
- Mezclando las anteriores

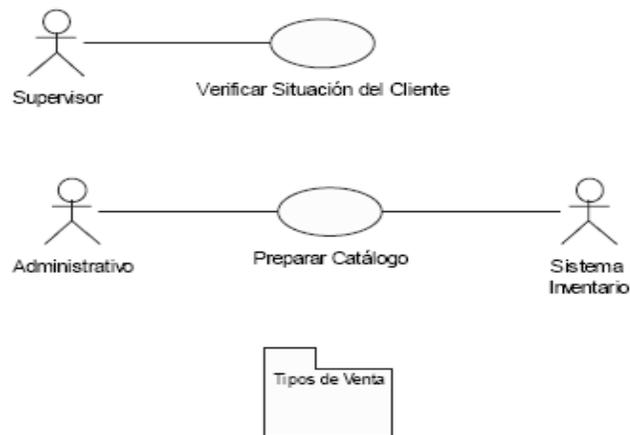


Figura 2.3 Diagrama de casos de Uso

2.4.3 Diagrama de Clases

Este tipo de diagramas es el principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia

Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones. Una clase puede representarse de forma esquemática (plegada), con los detalles como atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase.

El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

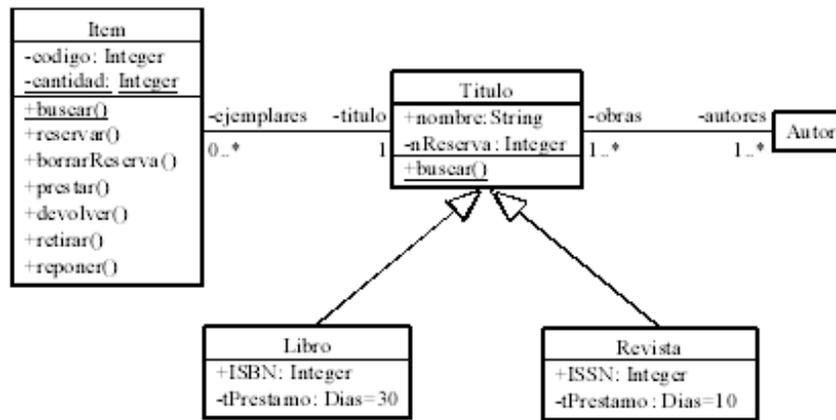


Figura 2.4 Diagrama de Clases

2.4.4 Diagramas de Interacción

En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración.

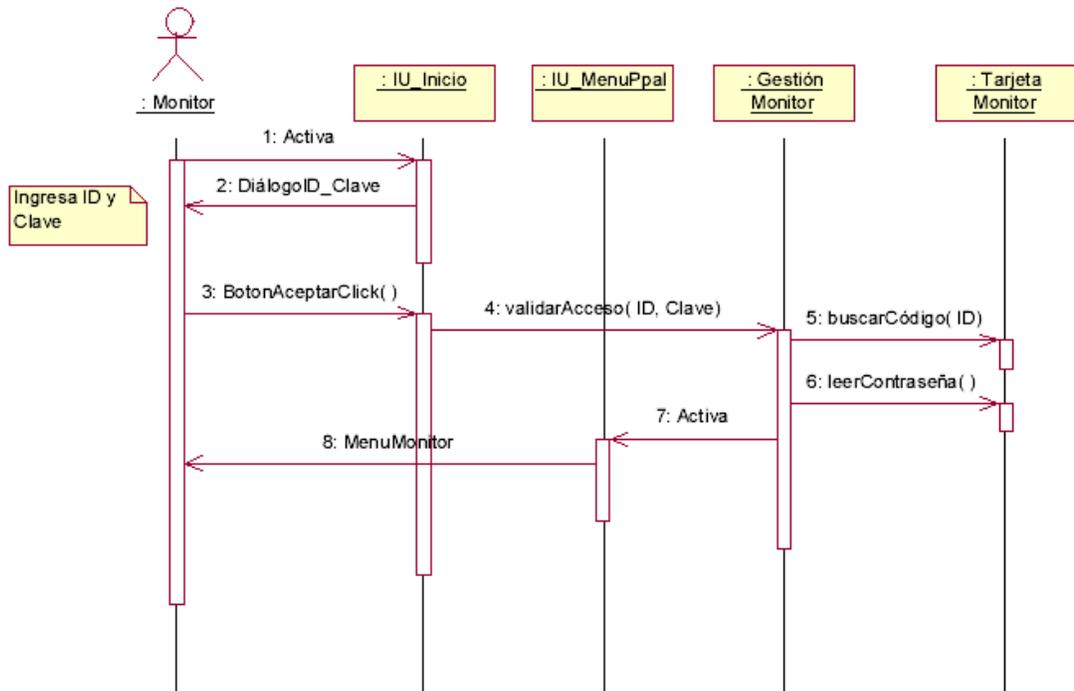


Figura 2.5 Diagrama de Secuencia



2.4.4.1 Diagrama de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba hacia abajo.

2.4.4.2 Diagrama de Colaboración

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

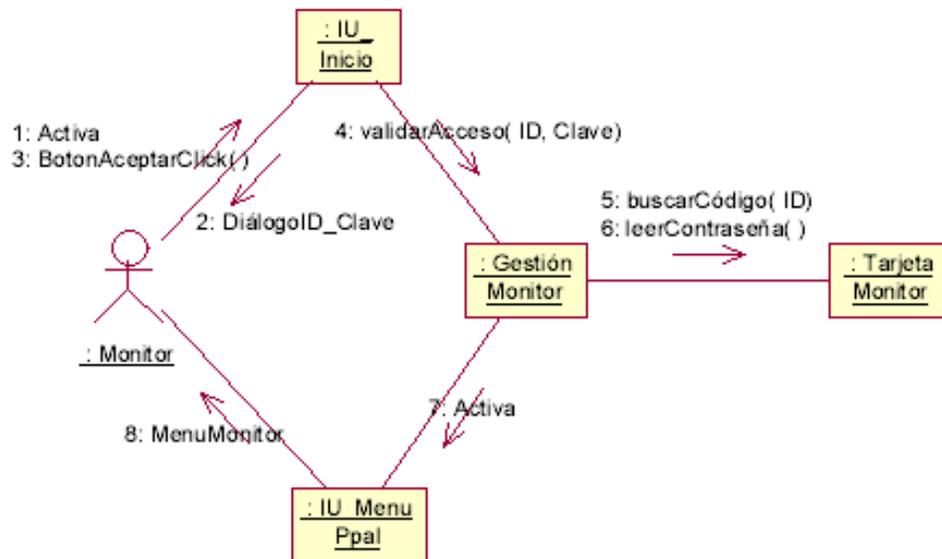


Figura 2.6 Diagrama de Colaboración.

En cuanto a la representación, un Diagrama de Colaboración muestra una serie de objetos con los enlaces entre los mismos, y con los mensajes que se intercambian dichos objetos. Los mensajes son flechas que van junto al enlace por el que “circulan”, y con el nombre del mensaje y los parámetros (si los tiene) entre paréntesis.



Cada mensaje lleva un número de secuencia que denota cuál es el mensaje que le precede, excepto el mensaje que inicia el diagrama, que no lleva número de secuencia. Se pueden indicar alternativas con condiciones entre corchetes.

2.4.5 Diagrama de Estados

Un Diagrama de Estados muestra la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos. Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa como una flecha desde el estado origen al estado destino.

La caja de un estado puede tener 1 o 2 compartimientos. En el primer compartimiento aparece el nombre del estado. El segundo compartimiento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas.

Una acción de entrada aparece en la forma entrada/acción_asociada donde acción_asociada es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta.

Una acción de salida aparece en la forma salida/acción_asociada. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta.

Una acción interna es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma nombre_de_evento/acción_asociada.

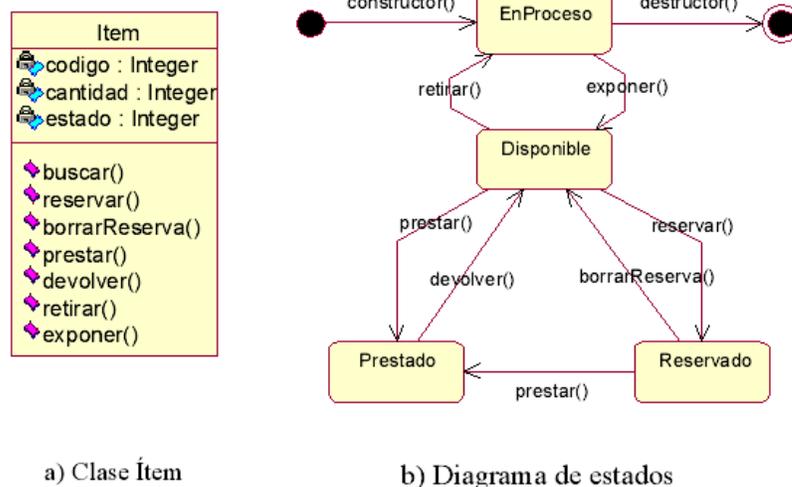


Figura 2.7 Diagrama de Estados



2.5 Bases de Datos

Una base de datos es una estructura de datos formados por 2 o más archivos que tienen relación entre sí.

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas. Por este motivo es importante conocer la forma en que están estructurados las bases de datos y su manejo.

2.5.1 Bases de datos para la Web

La mayoría de las aplicaciones Web sirven como interfaz hacia un almacén de datos, ya que permiten obtener y escribir datos de una manera intuitiva. Para aplicaciones pequeñas que no requieran de gran cantidad de procesamiento de datos, es viable almacenarlos en estructuras específicas que se pueden bloquear y desbloquear según las necesidades de la aplicación, sin embargo, para aplicaciones que requieren manipulación compleja de los datos, control de acceso concurrente de usuarios, mayor estabilidad, control de integridad de datos, entre otros, es imprescindible el uso de una base de datos.

Dadas las claras ventajas competitivas provistas por el acceso a bases de datos Web, es tentador enfocarse del lado de la ecuación Web-Base de Datos. No obstante, es importante darse cuenta de que una interfaz Web que interactúa con una base de datos mal diseñada es un campo fértil para que ocurran desastres de bases de datos a una enorme escala. Por otra parte, un buen diseño y ejecución de bases de datos, acoplados con una buena técnica de desarrollo Web, brinda incontables ventajas de negocio tácticas y estratégicas, oportunidades profesionales virtualmente ilimitadas.

Los sistemas de bases de datos para Internet implican más de solo el desarrollo de aplicaciones Web habilitadas para bases de datos. Además existen temas que deben ser abordados cuando se utilizan interfaces Web como entrada a bases de datos comparativas e interinstitucionales. Por ejemplo, existen retos de seguridad de los datos, de manejo de transacciones, de validación de datos de lado del cliente y muchos otros retos operativos y administrativos. El sistema de base de datos debe existir en un ambiente seguro que mantenga el acceso a los datos bien monitoreado y protegido, un robusto manejo de transacciones enfocado en el mantenimiento de la integridad de los datos, y una sólida recuperación de los datos. Por último, desde el punto de vista del gerente del negocio y del usuario, la base de datos no es particularmente útil a menos que su extremo frontal sea fácil de utilizar, capaz de proporcionar información y que cuente con aplicaciones de usuario que proporcionen transacciones útiles.



El acceso concurrente a bases de datos por múltiples clientes afecta la manera en la que las transacciones se definen y manejan. La existencia de soporte de múltiples fuentes y tipos de datos, el advenimiento de plataformas cada vez más independientes y portátiles, la distribución y escalabilidad y estándares abiertos tienen un efecto importante en como se desarrollan, instalan y manejan las aplicaciones.

Se argumenta que la Web afecta mucho el desarrollo de aplicaciones de bases de datos. Las características de Internet –y en particular su servicio Web- cambian de una manera fundamental el funcionamiento de las aplicaciones. La naturaleza sin estado de la Web tiene un impacto importante en la presentación y ejecución de las consultas. Solo hay que pensar en el modelo de solicitud-respuesta de la Web y en lo diferente que es de la idea del programador convencional.

Para que los sistemas de base de datos sean desarrollados y manejados inteligentemente, el DBA de hoy debe entender el ambiente de negocio basado en la Web para abordar con éxito los temas que propician el desarrollo, uso y manejo de las interfaces Web a bases de datos.

2.5.2 Modelo de Datos

El modelo de datos, es el enfoque utilizado para la representación de las entidades y sus características dentro de la base de datos, y puede ser dividido en tres grandes tipos:

2.5.2.1 Modelos lógicos basados en objetos

El modelo orientado a objetos se basa en objetos, los cuales contienen valores y métodos, entendidos como órdenes que actúan sobre los valores, en niveles de anidamiento. Los objetos se agrupan en clases, relacionándose mediante el envío de mensajes. Algunos autores definen estos modelos como "modelos semánticos".

2.5.2.2 Modelos lógicos basados en registros

Estos modelos se usan para especificar la estructura lógica global de la base de datos, estructurada en registros de formato fijo de varios tipos. El modelo jerárquico representa a registros en una estructura arborescente, el tipo de estructura implica que una entidad no puede tener mas de un sujeto que la pasee, por lo tanto, esta compuesta de muchas asociaciones (1:M)³³ o (1:1)³⁴

³³ uno a muchos

³⁴ Uno a uno



Una entidad superior se conecta de manera lógica con una inferior en una relación “padre-hijo” que puede tener mas de un hijo pero solo puede tener un padre, el nodo que no tiene padres se le conoce como raíz y a los nodos que no tienen hijos se les conoce como hojas.

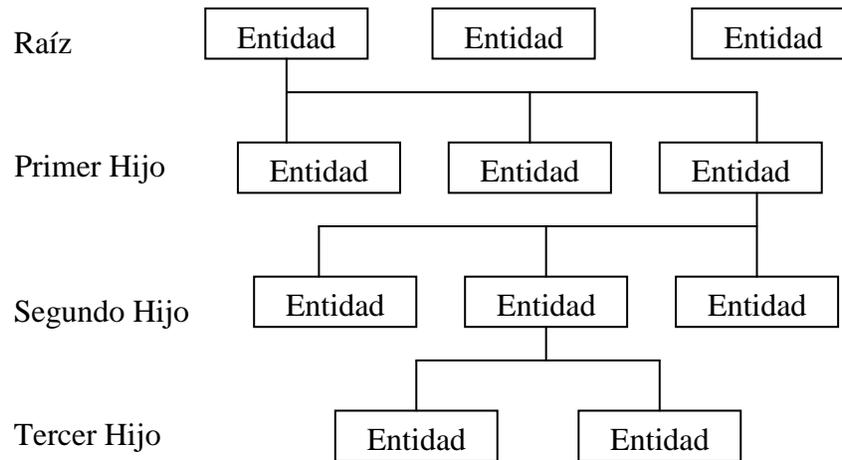


Figura 2.8 Modelo Jerárquico

El modelo de red está es una variación del modelo jerárquico. Permite a una entidad tener cualquier cantidad de subordinados o de superiores, se comunica usando enlaces de red que son conceptos de datos comunes a ambas, conectados en una relación (M:M)³⁵.

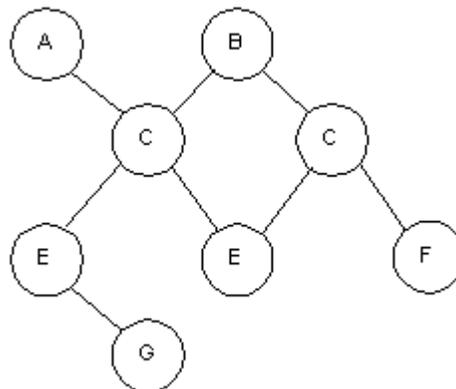


Figura 2.9 Modelo de red de datos

El modelo relacional representa los datos y sus relaciones mediante tablas bidimensionales, que contienen datos tomados de los dominios correspondientes. En este modelo, el lugar y la forma en que se almacenan los datos no tiene relevancia (a diferencia de los dos

³⁵ Muchos a muchos



anteriores). Esto tiene la considerable ventaja de que es más fácil de entender para un usuario.

Atributo 1	Atributo 2	Atributo n	
XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	Tupla 1
XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	Tupla 2
XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	.
XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	.
XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	XXXXXXXXX	Tupla n

Figura 2.10 Modelo Relacional de datos

2.5.3 DBMS

Según la definición de Elmasri y Navathe, un DBMS es “un sistema de software de propósito general que facilita el proceso de definir, construir, y manipular bases de datos para diferentes aplicaciones” [4].

El DBMS empleado debe tener las siguientes características:

- **Los datos se guardan en relaciones (o tablas).** Estas tablas deben estar organizadas de manera que cada una de sus columnas pueda ser identificada por su nombre, la manera en la que los renglones estén organizados no es relevante.
- **Las operaciones deben ser relacionales.** Las operaciones que el sistema provea, deben de generar nuevas tablas a partir de otras existentes. Por ejemplo: *SELECT nombre, id FROM customer* debe de crear una nueva tabla como resultado (la nueva tabla no tiene nombre) y ésta es un subconjunto de la tabla existente.
- **Debe de soportar uniones.** Una base de datos relacional debe soportar operaciones de unión, es decir que permita la obtención de datos relacionados de diferentes tablas y ponerlos en una misma tabla de resultados.

Independientemente de qué DBMS se elija y de cómo trabaje, lo que se busca es que el manejador provea de una API independiente del manejador. Entre las APIs independientes de manejador más comunes están ODBC, JDBC y DBI.

El papel de una API independiente del manejador es crear una interfase común para los diferentes manejadores. De esta manera, una aplicación programada que usa una API independiente para comunicarse con la base de datos, puede migrarse fácilmente a otro manejador, siempre y cuando éste cuente con soporte para la API independiente para la que fue creada la aplicación.

Recomendamos la utilización de un lenguaje de programación que provea de mecanismos de conexión a manejadores de base de datos en un grado de abstracción que permita una



fácil migración y se adhiera a los estándares de la industria. Esto se logra por medio de la librería DBI

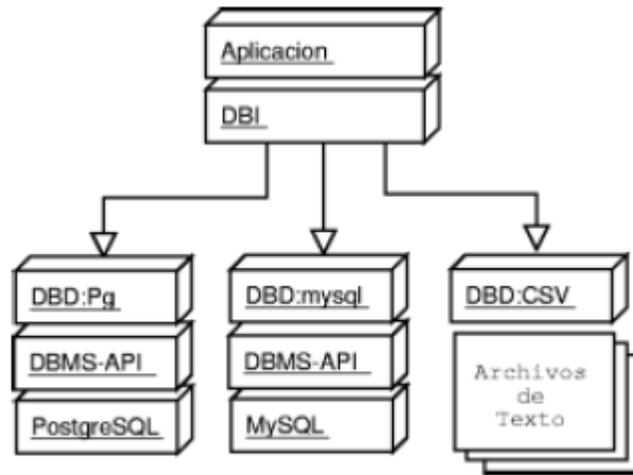


Figura 2.11: Acceso a datos con independencia del DBMS

2.5.4 Conexiones a Bases de Datos

Un problema muy importante que se presenta en el desarrollo de aplicaciones Web es la persistencia a la conexión. Las conexiones HTTP terminan cuando el servidor responde al cliente, por tanto si antes de esa respuesta se ejecuta un proceso que acceda a una base de datos, este proceso debe abrir una conexión con el DBMS para finalmente cerrarla al momento de terminar el proceso. Este esquema produce una sobrecarga de procesamiento cuando el acceso simultáneo de usuarios crece (*overhead*) y realentiza sensiblemente el procesamiento del servidor y la atención a las solicitudes de los clientes. Por ello se debe exigir que la conexión a la base de datos sea persistente y segura. Esta característica debe ser provista por el marco de trabajo y se basa en la creación de una zona de memoria compartida donde se establecerán las variables y procedimientos que accederán a la base de datos, y los procesos que atiendan a las solicitudes Web deberán interactuar con esta zona de memoria compartida y no directamente con el DBMS. Así mismo está la alternativa de crear un proceso alternativo al servidor de HTTP que se encargue de la conexión persistente y provea de una comunicación confiable con el servidor de HTTP, sin embargo, como ya se explicó anteriormente, desarrollar este servicio es una tarea compleja, aunque más flexible.

2.6 Capa de reglas de negocio (lógica de la aplicación)

Toda aplicación trata de reflejar parte del funcionamiento real de una organización, para automatizar tareas que de otro modo serían llevadas a cabo de modo más ineficiente, o bien no podrían realizarse. Para ello, es necesario que cada aplicación refleje las restricciones que existen en el negocio dado, de modo que nunca sea posible llevar a cabo acciones no



válidas. A las reglas que debe seguir la aplicación para garantizar esto se les llama reglas de negocio o bussiness rules.

Podemos clasificar las reglas de negocio en varios grupos:

El primer grupo de reglas de negocio engloba todas aquellas reglas que se encargan de controlar que la información básica almacenada para cada atributo o propiedad de una entidad u objeto es válida: no hay precios de artículos negativos, el sexo de una persona solo puede ser masculino o femenino, una fecha siempre debe ser una fecha válida, etc. A estas reglas se les conoce como reglas del modelo de datos.

Otro grupo importante de reglas incluye todas aquellas reglas que controlan las relaciones entre los datos. Estas reglas especifican, por ejemplo, que todo pedido debe ser realizado por un cliente, y que el mismo debe estar dado de alta en nuestro sistema: además, una vez que un cliente haya hecho algún pedido, se deberá garantizar que no es posible eliminarlo, a menos que previamente se eliminen todos sus pedidos. Estas reglas constituyen las reglas de relación.

Es frecuente que a partir de cierta información se pueda derivar otra: por ejemplo, el total de un pedido se puede calcular a partir de las distintas líneas que lo componen, mientras que el total de cada línea se puede calcular a partir del número de unidades vendidas y el precio por unidad. Al conjunto de reglas que especifican y controlan la obtención de información que se puede calcular a partir de la ya existente se las llama reglas de derivación.

Otro grupo de reglas de negocio es el compuesto por las reglas de restricción, que restringen los datos que el sistema puede contener. Nótese que este grupo de reglas se solapa en cierto modo con las reglas del modelo de datos, dado que aquellas también impiden la introducción de datos erróneos, como se vio anteriormente. La diferencia estriba en que las reglas de restricción restringen el valor de los atributos o propiedades de una entidad más allá de las restricciones básicas que sobre las mismas existen: por ejemplo, para un saldo existe una regla básica (regla del modelo de datos) que indica que éste debe ser un número, pero además puede haber una regla que indique que el saldo nunca puede ser menor que cierta cantidad tope establecida para cierto tipo de clientes. Esta sería lo que aquí denominamos una regla de restricción, y la diferencia fundamental estriba en el hecho de que este tipo de reglas requiere para su verificación del acceso a otros fragmentos de información, algo que no sucede con las reglas del modelo de datos.

El último grupo de reglas de negocio incluye aquellas reglas que determinan y limitan cómo fluye la información a través de un sistema. Por ejemplo, un cliente puede hacer una petición de análisis a un laboratorio, que anota un encargado, hecho esto, se genera un parte para uno o más analistas, estos realizan las mediciones correspondientes y devuelven las partes con el resultado pertinente, a partir de la cuál se genera un informe de análisis, que será un análisis válido solo cuando sea firmado por los responsables de garantizar su



corrección. A las reglas que indican qué camino recorre la información y obligan a que se sigan solo los caminos válidos se las llama reglas de flujo.

2.7 Capa de presentación

En el desarrollo de aplicaciones Web hay una gran tendencia a separar la presentación, la información en HTML/XHTML y la lógica de la aplicación. Esto se hace con el fin de delegar roles de trabajo, dentro del desarrollo del sistema: los diseñadores gráficos que se darán a la tarea de elaborar la capa de presentación de la aplicación utilizando HTML/XHTML y hasta técnicas de HTML dinámico (DHTML) en aras de la facilidad de uso. Los programadores y administradores de las bases de datos escribirán la lógica de la aplicación y las rutinas de acceso a datos.

En este punto hay que escoger entre dos alternativas: se le otorga el control de la lógica a los documentos en HTML y utilizando el modelo de desarrollo de *código incrustado en el HTML*, o se emplea el concepto de plantillas en la programación. Una plantilla es un documento completo en el cual existen partes que se pueden llenar con información variable para adaptarlo a la situación necesitada. Estas plantillas son leídas y procesadas por la aplicación para luego ser enviadas al agente del usuario. Las plantillas pueden ser constantemente modificadas por los diseñadores gráficos sin el temor de corromper el funcionamiento de la aplicación.

La diferencia entre la programación de código incrustado y el uso de plantillas radica en el programa procesador de documentos. Si éste interactúa directamente con el servidor HTTP y ofrece gran flexibilidad y así como una API completa y acceso a una gran colección de objetos, entonces estaremos bajo el modelo de código incrustado. Por el contrario, si el programa procesador de documentos es limitado, simple y puede ser manipulado desde cualquier programa de aplicación, entonces estaremos trabajando con el esquema de plantillas.

2.7.1 Presentación interactiva

El HTML dinámico puede permitir a los documentos Web tener una apariencia y actuar como una aplicación de escritorio o una producción multimedia. Ejemplos de lo anterior puede citarse el cambio del formato de un texto cuando el usuario pase el ratón sobre él; o permitir al usuario hacer “arrastra y suelta” de una imagen en otro lugar de la página Web.

El HTML dinámico no es más que la conjunción creativa del HTML, CSS, un lenguaje del lado de cliente como ECMA y un agente de usuario que soporte lo anterior, además de la especificación DOM³⁶. Alternativamente puede utilizar tecnología como Flash de Macromedia, o Applets Java o componentes ActiveX, pero están fuera del espectro del presente trabajo.

³⁶ Document Object Model (Modelo de Documento Objeto)



La capacidad de cambiar el contenido de un documento, de acuerdo a la interacción con el usuario, es muy importante para que el usuario se sienta contento utilizando la aplicación; se le puede proveer de retroalimentación inmediata, sin necesidad de esperar una respuesta del servidor. Al ofrecerle una interfaz parecida a las de las aplicaciones de escritorio, el usuario se sentirá más cómodo utilizando la aplicación Web.

No obstante, se corre el riesgo de hacer crecer los documentos HTML, y hay que tener en cuenta que el ancho de banda es un recurso muy limitado y la paciencia de los usuarios poca, por lo que es recomendable programar con sobriedad pero sin perder la elegancia. Además hay que tener en mente que no todos los agentes de los usuarios soportan las mismas características, y que por lo tanto se debe asegurar siempre que la aplicación Web se despliegue correctamente en cada agente posible del usuario, ya que no existe normalmente la capacidad de exigirle alguno en específico al usuario.



Capítulo 3

Análisis

Para que una institución educativa funcione de manera eficiente, es necesario que todos los elementos trabajen en conjunto para obtener los resultados que se plantean dentro de la misma institución y que son avalados por las autoridades educativas.

Dentro de estos elementos se encuentra el departamento de control escolar, el cual es conveniente que ofrezca respaldo y confianza. Para esto es recomendable contar con herramientas capaces de solventar la mayoría de los requerimientos establecidos.

Es por eso que los sistemas automatizados de control escolar participan de manera muy importante en las actividades y procesos del control escolar. A fin de lograr esto, se hace uso de la tecnología tanto de software como de hardware y comunicaciones en este caso, para la propuesta de una herramienta que sea capaz de llenar las expectativas de las instituciones educativas y disminuir los problemas de formato, de prontitud, de certidumbre, de duplicidad y de acceso a la información así como de gasto de recursos innecesarios por parte de la supervisión y de las escuelas adscritas a esta, problemas que causan conflictos de diversa índole entre los que destaca por su gravedad, el no cumplir con los tiempo de entrega de la información para los diversos tramites en beneficio de lo que hoy es la causa del planteamiento del presente proyecto: la matricula estudiantil.



3.1 Análisis preliminar del problema

En la supervisión escolar P095 de Naucalpan de Juárez, en el Estado de México, las escuelas particulares y públicas pertenecientes a esta (ver tabla 1), durante el ciclo escolar 2005-2006 han presentado un alto porcentaje en las variaciones de tiempo de entrega de la documentación y estadísticas de inscripción, cambios, bajas, etc. en el padrón escolar, a causa de conflictos con los sistemas de control enviados por las autoridades educativas, mientras algunas instituciones aun no hallaban como instalar el sistema, otras aun no entendían como manejarlo (no existen manuales de apoyo y los que hay parecen no ser claros).

Escuela	Clave Centro de Trabajo
Supervisión Escolar	P-095
Lic. Juan Fernández Albarrán	15EPR0613W
Justo Sierra	15EPR0601R
Gral. Vicente Guerrero	15EPR0888K
Jaime Nunó (turno Matutino)	15EPR0596W
Jaime Nunó (turno Vespertino)	15EPR0952V
Libertador Miguel Hidalgo (turno Matutino)	15EPR1794C
Libertador Miguel Hidalgo (turno Vespertino)	15EPR1916X
Colegio Inverness (particular)	15PPR2198R
Colegio José Lancaster (particular)	15PPR3301U
Colegio Eureka (particular)	15PPR3176M

Tabla 3.1.- Escuelas pertenecientes a la supervisión escolar P095

En el ciclo actual se esta presentando una situación similar, pues el "nuevo" sistema parece poco funcional, por lo que cada escuela debe llevar a cabo como puede todo este trabajo.

Es aquí donde las instituciones públicas quedan en desventaja con respecto a las instituciones privadas, debido a que las primeras no cuenta con personal dedicado a esta actividad, por tanto el personal asignado no tiene la suficiente capacitación en esta tarea, aunado a que no cuentan con un presupuesto para adquirir un software comercial dedicado a esta actividad, teniendo como consecuencia que algunas escuelas no pueda llevar al día su control de información de alumnos y personal.

Toda la problemática descrita anteriormente conlleva a un problema mayor al final del ciclo escolar: error en los datos con la consecuente falta de documentos oficiales (boletas, certificados, etc.)

Considerando estas situaciones, el problema a ser estudiado se basa en la siguiente pregunta:



¿Cómo consideran los directivos de las instituciones de la zona escolar P095 de educación primaria el sistema para captura y el sistema de control que reciben de parte de las autoridades de la SEP?

Y con base a ello:

¿Al utilizar un sistema de interfaz grafica se resolverán los conflictos en la captura y control de datos de alumnos, directivos y docentes?

3.1.1 Situación actual

Uno de los grandes desafíos dentro del departamento de control escolar de toda institución educativa, es lograr el mejoramiento de la calidad de los servicios que se ofrecen a los alumnos e institución en general.

El personal encargado de llevar esta tarea, requiere que los sistemas de captura y control de información propios (internos) u oficiales provistos por la Secretaria de Educación Pública (SEP) sean actualizados y sean amigables para que puedan ser mas útiles en las labores de la administración escolar, puesto que hoy día es muy poco el personal que se arriesga a trabajar con un sistema de captura y control escolar, pues que según dicen los directivos consultados, el problema comienza desde la instalación del sistema y continua durante todo el proceso de captura (fallas del sistema al hacer algún proceso) e impresión (conflictos de comunicación sistema-impresora), por citar algunos y finaliza con la no menos difícil o a veces imposible tarea de tratar de reparar el daño del sistema.

Otro problema conocido, es que la mayoría de los sistemas de control están diseñados con interfaces poco amigables de línea de comandos, lo cual hace que el capturista, secretaria, etc. tenga poco interés en entrar o utilizar el sistema, ya que como no hay "botoncitos" para dar clic, comienzan a hacerse a la idea de que el sistema es aburrido y difícil.

También se menciona que existen conflictos a la hora de la captura en el sentido de que una palabra se teclea de diferentes formas, por lo que se presenta la situación de no saber "cual es la buena", ya que no existen posibles alternativas de las que el usuario pueda elegir, por ejemplo la palabra "Tlalnepantla" en ocasiones se teclea "Tlanepantla".

Todos estos conflictos y fallas del sistema generan problemas como:

- Retrazo en la captura y entrega de los reportes requeridos por las autoridades educativas,
- Conflictos por no entregar dichos reportes como son requeridos, esto también se debe a que no existen formatos preestablecidos para la entrega de información, generalmente toman como formato el trabajo de la primera escuela que entrega completa la información,



- Confusión por no existir apoyo por parte de departamento de sistemas de la SEP, o la que existe es muy precaria y el contacto tarda aproximadamente una semana en responder y dar soporte a la institución, etc.

La duplicación de información debido a los constantes requerimientos de parte de la supervisión escolar, por lo menos se requiere en 3 ocasiones al año y en formatos distintos la misma información.

La no optimización en el uso de recursos humanos y materiales por parte de las escuelas pues si no esta la información como supuestamente se indicó se tendrá que volver a procesar e imprimir.

La incompatibilidad del sistema Programa de Captura de Educación Primaria (PRIM) con el Sistema Operativo Windows XP pues hay algunas opciones que no funcionan, por ejemplo la impresión de archivos o en algunas ocasiones la conexión con la base de datos.

Las problemáticas anteriormente mencionadas son el resultado de un estudio en el que se emplearon los siguientes métodos de recolección:

Entrevistas: fueron realizadas a los directivos y personal encargado de la información de las 10 escuelas y al personal de la supervisión escolar.

Cuestionarios: Estos se usaron para obtener respuestas muy específicas en aspectos operativos

En todos los casos, los docentes entrevistados reconocieron la necesidad de disminuir el volumen de información y/o automatizarlo ya que muchos de los formatos entregados contienen la misma información, en su mayoría tienen que entregarse en más de una ocasión durante el año y cada vez que es requerido hay que procesarla. Los formatos entregados cada año son los siguientes:

- **Forma 911:** Las escuelas reportan a la SEP (Forma 911), al principio y al fin de cada curso, las cifras de aprobación y reprobación, y a partir de estos datos se integra la estadística educativa oficial del sistema educativo nacional en la que se presentan estimaciones y coeficientes de cobertura, reprobación, deserción, eficiencia terminal y otros.

Al inicio del curso se reportan los alumnos de nuevo ingreso y los alumnos reprobados de cada grado separados por edad y sexo.

Al final del año se presenta esta misma forma en donde se reportan los aprobados, los reprobados y la inscripción total al final de año.



- **Libro de inscripción:** Es un concentrado de los formatos de inscripción de cada grupo, se llena por los profesores y se coloca en una carpeta junto con las copias de la documentación, estos formatos son procesados en una hoja de cálculo para formar lo que se denomina libro de inscripción. Este libro se cierra el 30 de septiembre pues después de esta fecha se tienen que reportar altas y bajas con respecto del libro de inscripción del año anterior, mensualmente se hace un reporte de las altas y bajas según se vayan presentando y al final del curso se presenta el libro de inscripción final y un documento con el concentrado de las altas, bajas y cambios ocurridos durante el año.
- **Plantilla del personal docente:** Dos veces al año se entrega esta plantilla con la información de cada uno de los profesores, tanto el personal docente frente a grupo como del personal docente de clases especiales y promotores.
- **Indicadores:** Se entregan varias relaciones con los indicadores de aprovechamiento cada bimestre:
 - Aprovechamiento de primero y segundo grado, en este se reporta el aprovechamiento de cada una de las asignaturas.
 - Aprovechamiento de tercer grado hasta sexto, igual que el anterior se reporta el aprovechamiento, es importante señalar que se separan los grados debido a que las asignaturas que se cursan en primero y segundo son diferentes a las que se cursan en años posteriores.
 - Indicadores de eficiencia de las materias: Español y Matemáticas de primero a sexto grado
 - Indicador de eficiencia terminal de sexto grado: para este formato se revisa el libro de inscripciones de la generación que se encuentra terminando sexto grado y obtiene el porcentaje de bajas y termino de los alumnos que se inscribieron en primer grado 6 años atrás.
- **PRIM:** Es el programa que envía la SEP para la captura de información de los alumnos, existen dos modalidades de este:

Captura: En octubre se reporta la inscripción a primer grado y se hacen las modificaciones de altas y bajas de segundo a sexto grado.

En enero se hace una revisión y se reportan solo altas y bajas.

Calificaciones: Al término del ciclo escolar se reportan las calificaciones, altas, bajas y estadísticas finales.

- **Sistema Automático de Inscripción y Distribución (SAID):** Es el sistema de preinscripción a primer grado, este programa se ejecuta solo en el mes de febrero, la escuela se queda con la documentación anexa y la información se envía a la



supervisión y de ahí se manda la información de toda la supervisión a las oficinas de SAID y es en este lugar en donde se asignan a los alumnos en cada escuela.

- **Fotorama:** Es una plantilla con el nombre y la foto de los alumnos de sexto grado, de aquí se obtiene la relación de certificados.
- **Relación de Certificados:** Esta relación la proporcionan las autoridades escolares de la Subsecretaría de Educación Básica y Normal, en ella vienen los datos del certificado: Folio, Nombre del alumno, Clave Única de Registro de Población (CURP), se reportan las calificaciones a mano, el alumno acusa de que la información es correcta y se regresa a la Subsecretaría de Educación Básica y Normal para que se obtengan los certificados.
- **Libro de Certificados:** Cuando ya se tienen los certificados en cada escuela, se llena este libro a mano, contiene la siguiente información: Número progresivo, Folio del certificado, Nombre del alumno, CURP, Fecha de expedición, fecha de recibo, firma del alumno.
- **Boleta de Calificaciones:** Existen 2 tipos de boletas, unas que son provisionales, que se entregan a cada bimestre y las boletas oficiales que se entregan al final del curso, las primeras son llenadas a mano y las segundas son llenadas en maquina de escribir.

Con este antecedente en la Supervisión Escolar P095 se decidió elaborar una aplicación informática que le ayude al personal encargado de la supervisión, de la dirección y control escolar a facilitar el trabajo además que se propusieron algunas utilidades con las que actualmente no se cuenta.

3.2 Alcances

Basados en las necesidades antes mencionadas nos marcan las siguientes metas a perseguir en el desarrollo de la propuesta del sistema:

- Obtener un interfaz para capturar la información de alumnos que están inscritos en las escuelas pertenecientes a la Supervisión escolar P095 (matricula, nombre completo, CURP, dirección completa, familiares de contacto, datos académicos) en un contexto cliente servidor que este disponible desde la Internet bajo los siguientes criterios:
 - Alta
 - Baja
 - Cambio



- Obtener un interfaz para capturar la información del personal docente frente a grupo y directivo que están adscritos en las escuelas pertenecientes a la Supervisión escolar P095 en un contexto cliente servidor que este disponible desde la Internet.
- Permitir el acceso vía Internet a profesores frente a grupo con el fin de asentar las calificaciones parciales y finales de cada uno de los alumnos de su grupo asignado en el curso escolar.
- Permitir la consulta vía Internet al personal administrativo de la supervisión escolar con el fin de cotejar contra documentos los datos de los alumnos registrados por los directores en cada una de las escuelas y permitirle modificar la información en caso de existir un error, en caso de que no exista error, validar la información.
- Permitir la consulta vía Internet a los padres de familia para consultar las calificaciones parciales y finales de los alumnos.
- Permitir la impresión de la información de alumnos y profesores en papelería oficial a los directivos y permitir a los padres de familia imprimir la información académica de los alumnos.
- Tener un apartado para generación y consulta de reportes.
- Disponer de un espacio para la publicación de documentos y comunicados

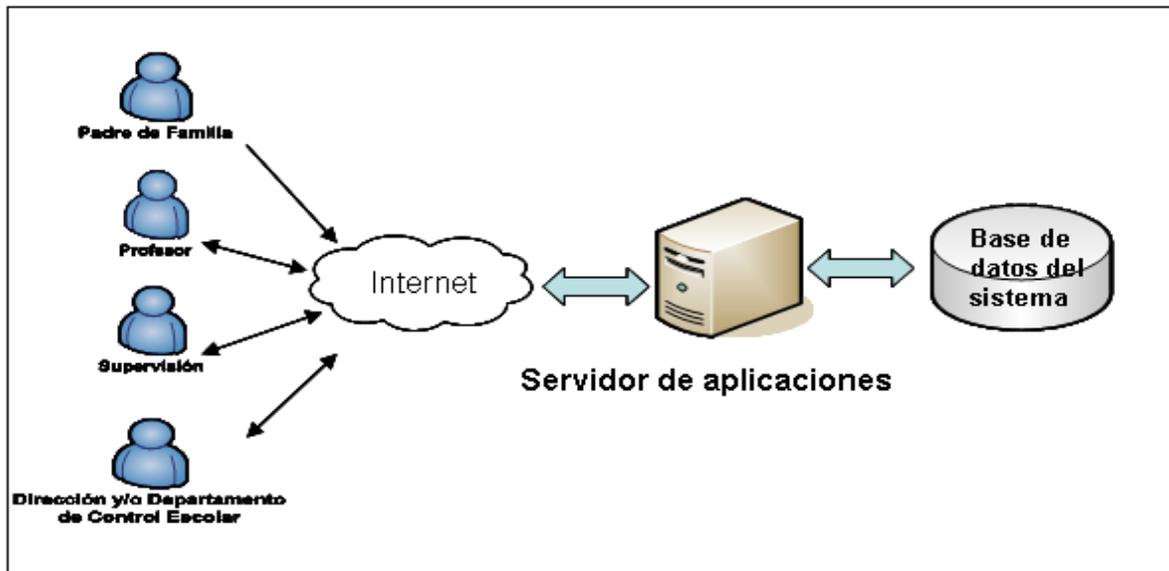


Figura 3.1.- Diagrama que muestra el funcionamiento del sistema interactuando con sus principales actores



3.3 Estudio de factibilidad

El sistema tendrá que interactuar con dos sistemas enviados por la SEP:

El primero es el sistema SAID el cual es utilizado para inscribir a los alumnos de primer ingreso, dicho programa cuenta con una base de datos, la cual esta homologada con el sistema PRIM, que es el sistema de control escolar oficial de la SEP, por tanto el sistema propuesto, el *Sistema Automatizado de Control* (SAC), tendrá que estar también homologada con este sistema para poder enviar los datos a la Subdirección de Control Escolar en el formato correcto.

Estos tres sistemas formarán el Sistema de Registro y Control Escolar de la Supervisión Escolar P095, es importante mencionar que no es la suma sino mas bien la integración de los tres sistemas, de tal forma que homologados y con la reducción de procedimientos sea más efectiva la administración de la matrícula escolar.

El SAC es el sistema que se analiza y diseña dentro de este trabajo de tesis, el cual no tiene antecedente de forma oficial aunque algunas escuelas particulares de la Supervisión tienen implementado algún sistema que realiza funciones parecidas como el *Control Administrativo Escolar* de la empresa *Softwgroup*, sin embargo, debido al nulo presupuesto por parte de las autoridades oficiales destinado a la adquisición de tecnología a nivel local, es difícil adquirir un Software propietario destinado para esta actividad.

3.3.1 Factibilidad técnica

La descripción de los procesos revisados en la parte del análisis preliminar al inicio de este capítulo y la naturaleza de la aplicación, permitieron visualizar que los procesos pueden ser automatizados porque son repetitivos y computables. Para llevar a cabo el proyecto se tienen los siguientes elementos:

3.3.1.1 Hardware

Los equipos de cómputo que se propone ocupar son los que se encuentran en la dirección de cada plantel los cuales tienen las siguientes características:



Escuela	Modelo de Equipo	Procesador	Memoria RAM
Supervisión Escolar	3 AcerPower SK20	Pentium 4 2.8 GHz	512 Mb
Juan Fernández Albarrán	AcerPower Sn	Pentium 4 2.4 GHz	512 Mb
Justo Sierra	AcerPower SK20	Pentium 4 2.8 GHz	512 Mb
Vicente Guerrero	AcerPower SK20	Pentium 4 2.8 GHz	512 Mb
Jaime Nunó Matutino	AcerPower SK20	Pentium 4 3.0 GHz	512 Mb
Jaime Nunó Vespertino	AcerPower SK20	Pentium 4 3.0 GHz	512 Mb
Libertador Miguel Hidalgo Matutino	HP E-PC42	Pentium III 1.7 GHz	256 Mb
Libertador Miguel Hidalgo Vespertino	AcerPower Sn	Pentium 4 a 3.0 GHz	512 Mb
Colegio Inverness (particular)	Optiplex GX620	Pentium 4 a 3.0 GHz	512 Mb
Colegio José Lancaster (particular)	AcerPower SK20	Pentium 4 a 3.0 GHz	512 Mb
Colegio Eureka (particular)	AcerPower SK20	Pentium 4 a 3.0 GHz	512 Mb

Tabla 3.2 Características de los equipos de cómputo de las escuelas primarias

El equipo en el que se propone desarrollar la aplicación tiene las siguientes características:

Modelo	HP a1300la
Procesador	Intel® Celeron D 346 (2.8 GHz, Front Side Bus de 533 MHz, 256KB Cache L2)
Disco Duro	80 GB 7200rpm Serial ATA
Disco Óptico	DVD±R/RW soporta Doble Capa (16X)
Memoria	256MB (1 x 256) PC3200 DDR2

Tabla 3.3 Características del equipo de desarrollo

Tanto las escuelas como la supervisión cuentan con impresora propia.

3.3.1.2 Software

Para el desarrollo y consulta de la estructura de datos así como para el desarrollo de las interfaces propuestas en este trabajo de tesis es requerido el siguiente software:

- Sistema Operativo.
- Navegador de Internet.
- Servidor HTTP para poder visualizar en forma local los resultados de lo que se esta programando y para poder publicar la información a Internet.
- Aunque no es necesario, pues se puede usar cualquier editor de texto, se recomienda utilizar un editor del lenguaje PHP.
- Software que interpreta las instrucciones del lenguaje PHP
- Se requerirá de un Sistema Manejador de Base de Datos (DBMS).



Se propone utilizar el equipo de la Escuela Primaria Juan Fernández Albarrán el cual tiene Windows XP Profesional SP2.

Para el desarrollo de la aplicación se optó por utilizar Software Libre que cuenta con la licencia GPL[5] y como se ha mencionado antes, el presupuesto no está destinado para adquirir licencias de software propietario, y se pretende mejor invertirlo en el sistema que motiva esta tesis, además de que el software libre se ha desarrollado tanto que ofrece muchas de las características que ofrece el software propietario e incluso en algunos casos las soluciones son mejores en aspectos importantes como son seguridad y estabilidad, la deficiencia que se podría tener es el soporte que ofrecen el software propietario pero eso se compensa en gran manera con el gran número de sitios de ayuda que existe en Internet.

Del software libre se propuso utilizar el siguiente:

Como servidor HTTP Apache por su estabilidad, solidez y rendimiento superiores a IIS que es el servidor Web que está incluido en Windows, Apache se puede descargar del sitio oficial <http://www.apache.org/>

Como DBMS se propuso utilizar MySQL que fue desarrollado para manejar bases de datos mucho más rápidamente que las soluciones existentes y ha sido usado exitosamente, es de fácil uso, rápido, confiable, robusto y se puede obtener desde el sitio oficial del proyecto www.mysql.com

Se propuso utilizar PHP como lenguaje de programación.

PHP es un lenguaje de scripts del lado del servidor para generar páginas de forma dinámica, aunque existen otras opciones como CGI, ASP y JSP, se optó por PHP debido a que tiene algunas ventajas como que es software libre, abierto, seguro, multiplataforma, funcional, es procedural pero a partir de la versión 5 tiene más soporte a programación orientada a objetos, tiene soporte para varios servidores Web, soporte nativo para prácticamente cualquier base de datos, buena documentación, simplicidad para aprenderse y utilizarse. El software para que el Servidor Web pueda interpretar los archivos con extensión .php, se recomienda bajarlo del sitio oficial www.php.net

Para desarrollar los scripts PHP, aunque se podría realizar en un editor de texto, se propone utilizar un editor de PHP como el *Maguma Studio* o *PHP Edit* por las ayudas que le podrían proporcionar al programador.

Como cliente para visualizar la aplicación nosotros utilizamos Internet Explorer, Opera, Firefox y Netscape.

Es importante mencionar que para el servidor de aplicación y el servidor de base de datos se recomendó contratar el servicio de hospedaje con Linux/Apache/MySQL/PHP.



En las computadoras de las 10 escuelas primarias 3 tienen Windows XP Profesional, 3 tienen Windows XP Home y 5 tienen Windows 2003 Estándar Edition, mientras que en la supervisión los 3 equipos cuentan con Windows XP Profesional, cabe señalar que todos los equipos cuentan con el navegador Internet Explorer pero es factible utilizar cualquier otro de los antes mencionados descargándolo del sitio oficial.

3.3.1.3 Comunicaciones y conectividad.

Para que pueda funcionar la aplicación, es necesario tener conectividad a Internet en la supervisión escolar y actualmente se cuenta con una red de datos a la cual es necesario fortalecer en la seguridad informática.

En cada una de las escuelas se cuenta con conexión a Internet, sin embargo este servicio se encuentra en la mayoría de las escuelas solo en el aula de medios y la red se encuentra solo en esta aula, por lo que tendrán que ampliar la red para facilitar el acceso a la dirección escolar o área de control escolar.

3.3.2 Factibilidad económica

Por la parte del hardware se puede deducir lo siguiente

Los costos de inversión en equipo nuevo son nulos puesto que ya se tiene todo el equipo, tanto de la supervisión como de las escuelas.

Como se menciono antes, se recomienda utilizar un servicio de hospedaje³⁷ para el servidor de la aplicación y de bases de datos. Existen en el mercado ofertas bastante accesibles como las ofrecidas por las siguientes empresas:

Empresa	Servicios Ofrecidos
www.dattatec.com	Paquete Promopack en Linux por \$319.00 anuales
www.aehosting.com.mx	Paquete Básico en Linux por US\$30.00
www.geniusys.com	Paquete Básico en Linux por US\$3.5 mensuales

Tabla 3.4 Paquetes de hospedaje ofrecidos por 3 empresas diferentes
(Paquetes cotizados el 15 de diciembre de 2006)

Es interesante resaltar que el costo de hospedaje en un servidor es mucho menor que la inversión que se tendría que hacer en acondicionamiento del lugar en donde se tendría el servidor, gastos como electricidad, administración del servidor, aire acondicionado,

³⁷ es el servicio que provee a los usuarios espacio de un servidor para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía Web.



mobiliario para el colocar el servidor, en su caso contratación de un servicio dedicado de Internet para poder tener una IP fija para poder publicar hacia Internet, etc.

En la parte de software se analizaron las 2 mejores opciones de control escolar ya existente:

Grupo Ges:

Ofrecen un sistema para el control escolar bastante completo, incluso incluyen algunos módulos que no se utilizarían en las escuelas de la federación por su carácter de gratuidad, se presentan 2 propuestas:

- La primera de estas es que cada una de las escuelas tendría que comprar una licencia del software que al momento del estudio estaba en \$1,800 dólares +IVA y aparte la supervisión tendría que adquirir una licencia más para poder administrar a las escuelas, la parte Web del sistema sólo es para consulta
- El segundo escenario es que se comprase una licencia corporativa, en donde se podrían administrar todas las escuelas pertenecientes a la zona escolar pero de una forma centralizada, es decir, desde un solo lugar para todas, estratégicamente plantean la supervisión escolar, por lo cual los directivos de cada una de las escuelas tendrían que desplazarse a realizar las modificaciones necesarias cada vez que fuesen necesarias y desde sus escuelas vía Internet podrían hacer consultas.

Éxito Software:

Tienen la versión SE (Servicios Escolares) orientado a resolver las necesidades de información de Control Escolar en un plantel educativo, cuenta con un modulo para pagos Tiene tres modalidades:

- Monousuario 1,500 dólares + IVA, se tendría que comprar una para cada escuela y una para la supervisión.
- Versión WAN, se necesitaría un servidor potente dedicado, enlace dedicado, cuando menos 2 megas reales para cada escuela y el software tiene un precio aproximado de 80,000 dólares.

Dado que ninguna de las soluciones anteriores satisfacen las necesidades del proyecto se confirma que es necesario desarrollar una solución hecha a la medida de las necesidades de la supervisión.

Para la conectividad solamente habría que ampliar la red de las salas de cómputo hacia la dirección escolar, en todos los casos es viable hacerlo por medio de cable UTP pues las distancias son menores a 100 metros y existe por lo menos un puerto disponible en el switch lo cual implica un costo aproximado de 50 dólares cada extensión.



En la supervisión si se tendrá que contratar el servicio de Internet lo que implicaría un costo aproximado de 380 dólares anuales.

Los Costos para el personal que llevara el proyecto a cabo incluyen 100 horas de analista a 75 dólares, 300 horas de programador a 50 dólares y 20 horas de instalación y capacitación a 75 dólares. Los costos de operación y mantenimiento anual implican 7200 dólares en salarios y 800 en servicios y consumibles.

Para este proyecto es importante mencionar que independientemente de los beneficios económicos que genera, los principales y los más importantes en este proyecto son intangibles, es decir, beneficios de impacto social, porque están más ligados a la misión de la institución, podemos mencionar los siguientes:

- Reducción en los tiempos de entrega de la información, esto se logrará al establecer un calendario de actividades en las que se podrán realizar cada una de las actividades en que se tenga que utilizar la aplicación.
- Estandarización de los formatos de entrega. Como será una sola aplicación que genera los reportes, todos se obtendrán con el mismo formato.
- Centralización y fácil acceso a la información. Este es uno de los grandes beneficios que se desea obtener pues al estar toda la información en un solo sitio se podrá consultar de manera más fácil y se tendrá la certeza de que es la información más actual.
- Aprovechamiento de tiempo en otras actividades. Los directivos ya no tendrán que desplazarse a la subdirección para entrega y recepción del programa, reportes ni documentación y lo podrán dedicar a otras actividades como atención de otros asuntos de la escuela, planeación de actividades, mejoramiento en alguna actividad o servicio o atención de padres de familia, etc.

También se obtendrán algunos beneficios tangibles entre los que podemos mencionar los siguientes.

- Se reducirían en promedio 10 horas bimestrales por directivo y 5 por profesor, que en promedio ganan 10 dólares la hora, considerando que son 11 escuelas nos arrojan la suma de 38,500 dólares.
- Disminución o eliminación del gasto en disquetes en un promedio cada escuela y la supervisión se gastan 2 cajas al año con un costo aproximado de 70 dólares
- Disminución en la inversión de traslado de la escuela a la supervisión y su respectivo regreso 700 dólares anuales.
- Disminución en la inversión anual que se hace de papelería y suministros para la impresora 800 dólares.



Pensando que este proyecto se pondría a punto hasta la mitad del próximo año, se tienen los siguientes montos estimados:

Concepto	Año				
	2007	2008	2009	2010	2011
Beneficios					
Reducción de costos en consumibles y papelería	1,570	770	770	770	770
Ahorros de costos operativos	38500	38,500	38,500	38,500	38,500
Beneficios Totales	40,070	39,270	39,270	39,270	39,270
Costos					
Análisis	7,500				
Diseño	15,000				
Instalación	1,500				
Capacitación	1,500				
Hardware	0				
Software	0				
Comunicaciones	974	424	424	424	424
Operación y Mantenimiento		8,000	8,000	8,000	8,000
Costos Totales	26,474	8,424	8,424	8,424	8,424
Diferencia	13,596	30,846	30,846	30,846	30,846
Total de ahorro en la vida del proyecto					136,980

Tabla 3.5 Resumen del análisis costo/beneficio (precios estimados en dólares)

3.3.3 Factibilidad operacional

El 90% del personal docente directivo y auxiliar de las escuelas pertenecientes a la Supervisión están familiarizados con Internet, y tienen conocimientos suficientes para poder operar la aplicación con una capacitación previa.

El 70% personal docente frente a grupo de esta supervisión esta relacionado con Internet y conoce el funcionamiento de un navegador de Internet por lo que podría sin problemas operar la aplicación para la captura de su currículo y asentamiento de calificaciones parciales y finales .

Las 2 profesoras asesoras de la supervisión que están encargadas de la elaboración y concatenación de la información proveniente de las 10 escuelas tienen un amplio conocimiento del uso de por lo menos un navegador de Internet por lo tanto no tendrán problemas para el uso de la aplicación previa una capacitación.

El 10% restante de los directivos y el 30% de los profesores frente a grupo que no esta familiarizado con el ambiente Web, por lo menos tiene conocimientos básicos de computación, por lo que se tendría que dar una capacitación al respecto, sin embargo habrá



que poner más atención a los directivos puesto que ellos son los que tendrán más interacción con la aplicación.

La aplicación permitirá que los directivos obtengan la información con mejor calidad y de una manera más fácil, permitirá que sean más productivos y que dediquen menos tiempo a las tareas administrativas de control escolar.

También permitirá que los profesores frente a grupo puedan asentar mas fácilmente las calificaciones y no tendrán que hacer el trabajo 2 o 3 veces en el año, considerando que cada profesor tiene en promedio 50 alumnos, al ahorrarles por lo menos 2 veces el tener que realizar esta actividad, les deja mas tiempo para que realicen otras actividades.

3.4 Determinación de requerimientos

A continuación se presentan de manera clara todos los requerimientos que se deben cubrir en el Sistema de Administración Escolar:

Se identifican 5 módulos principales que son:

1. Alumnos: Administra la información relacionada con los alumnos que se encuentran inscritos en las escuelas primarias que pertenecen a la Supervisión Escolar P095.
2. Profesores: Administra la información relacionada con los profesores que se encuentran inscritos en las escuelas primarias que pertenecen a la Supervisión Escolar P095.
3. Supervisión: consulta la información relacionada con los alumnos y administra la información de los profesores pertenecientes a la supervisión escolar, aunado a la generación de los resúmenes y graficas de la información de los alumnos y su aprovechamiento escolar requeridos por las autoridades escolares.
4. Dirección: Administra la información relacionada con los alumnos y profesores pertenecientes a una escuela en particular, así como también se encarga de generar los resúmenes y graficas de la información de los alumnos y su aprovechamiento escolar requeridos por las autoridades escolares.
5. Administración: Permite la gestión de los empleados administrativos, de los usuarios del sistema, del estado de las terminales, bitácoras de eventos y ciclos escolares.

3.4.1 Requerimientos de funcionalidad

Autenticación Inicial/ Inicialización de sesión



Enviar al nodo central la información de la sesión: usuario(directivo, profesor) y escuela, al inicio de las operaciones de captura.

Captura

- Permitir la captura de los siguientes datos de los alumnos: Clave Alumno, Turno, Grado, Grupo, Movimiento, Movimiento anterior, Fecha de nacimiento (aaaa/mm/dd), Entidad de nacimiento, Sexo, Apellido Paterno, Apellido Materno, Nombre(s), CURP, Repetidor, Dirección, Domicilio, Teléfono, Datos del padre o tutor.
- Modulo que permita asentar la calificación obtenidas por cada alumno en las respectivas asignaturas en los 5 bimestres.
- Permitir la captura de los siguientes datos de los profesores: Clave de ISSEMYM, Apellido Paterno, Apellido Materno, Nombre(s), Fecha de nacimiento (aaaa/mm/dd), Años de servicio, Función, Número Escalafonario, CURP, Categoría de pago, Preparación Profesional, Domicilio, Estado civil, Grupo a cargo, numero de alumnos a cargo.
- Permitir la corrección a los registros que fueron capturadas de manera errónea.
- Permitir dar de baja un registro de alumno o profesor previamente capturado.
- Al momento de la captura no se realizaran las validación de la información, esa validación se hace por parte de la supervisión.
- Permitir al personal de la supervisión la corrección de los registros que fueron capturados de manera errónea y validar la captura.
- Permitir la impresión desde las terminales de comprobantes de información procesada para las operaciones de ALTA, CORRECCION, BAJA.
- Permitir al operador consultar el estatus de captura en la sesión, el cual deberá incluir información referente al número de registros capturados, transmitidos al aplicativo central y de aquellos de los cuales se hayan impreso.

Administración de Grupos

- Se requiere contar con un modulo que permita la administración de los grupos.

Envío de transacciones



- Firmar electrónicamente el envío de cada transacción.
- Las transacciones que se envíen deben de contener el ID_TERMINAL (dirección MAC), FECHA_HORA, ID_FIRMANTES, FIRMAS, DATOS y TIPO_OPERACION.

Sincronización/HB

- Cada terminal envía un mensaje Heartbeat³⁸ (HB) cada 5 minutos al nodo central para informar de su estado (registros capturados, enviados, impresos).
- Cada Terminal debe responder al proceso de sincronización con el envío de las transacciones que el módulo central solicite después de verificar el contenido de la información de los HB.

Configuración

- Por la naturaleza de la aplicación no se necesita una configuración de las estaciones de captura/consulta y su arquitectura permite mantener actualizada la versión de la aplicación.

Monitoreo

- Se debe contar con un programa de Reimpresión y Monitoreo de terminales ejecutándose desde una PC, la cual se encontrará en la supervisión.
- La Reimpresión desde la PC comprende la información de las operaciones realizadas por cualquiera de las terminales
- El programa de monitoreo en la PC reporta la información capturada, enviada e impresa; procesada en cada Terminal correspondiente.
- El programa de monitoreo en la PC reporta el estado de las terminal correspondiente (Activo/Inactivo).

Consulta

³⁸ Es un fragmento de software que monitoriza la disponibilidad de nodos, envía mensajes a los nodos que quiere monitorizar, y en caso de que uno no conteste, lo considera muerto



- Se requiere un modulo que permita la consulta de la información académica e incidencias de cada uno de los alumnos.
- Se requiere un modulo que permita la consulta de los alumnos por grupo y la generación de las listas de los mismos.

Generación de Reporte Concentrado.

- Resumen Global Principal que muestra una tabla que contiene: El número de grupos que tiene la escuela, de cada grado se desglosa: numero de alumnos inscritos en cada grado, el número de alumnos inscritos en cada grupo, porcentaje de hombres y mujeres de cada grupo, profesor asignado y número de alumnos que conforman la población estudiantil.
- Resumen Global Principal, también muestra una tabla que contiene: Número de profesores frente a grupo, grupo asignado, Número de profesores de clases especiales y asignatura que imparten (música, inglés, computación, danza), Número de promotores educativos y actividades que realizan.

Generación de Reportes detallados y estadísticas

- Detalles de Alumnos dados de alta, desglosado por grado, grupo, fecha del movimiento y persona que autoriza el movimiento.
- Detalles de Alumnos que causaron baja de la escuela que incluye una breve explicación de los motivos, fecha del movimiento y persona que autoriza el movimiento y en su caso persona que lo solicita
- Detalles de los alumnos que cambiaron de grupo que incluye una breve explicación del motivo por el cual se efectúa el movimiento, fecha, persona que autoriza y en su caso persona que lo solicita.
- Detalles de Alumnos que están inscritos a cada uno de los grados desglosados por grupo, incluye el porcentaje de hombres y mujeres.
- Detalle de estadísticas finales en las que se reporta:
 - Aprobación (bimestral y anual)
 - Reprobación (bimestral y anual)
 - Retención
 - Calificaciones de Español y Matemáticas
 - Promedio de edad de cada grado
 - Para los alumnos de sexto grado se calcula la eficiencia terminal.



- Gráficos de calificaciones por grupo, alumno y materia
- Detalles de un alumno determinado, incluye datos generales del alumno, calificaciones parciales, y observaciones realizadas por el profesor o directivo.
- Detalle de información académica de cada alumno para impresión de boletas escolares al final del curso

Conexión/Desconexión.

- Se requiere contar con los mecanismos que permitan proveer al módulo con las repuestas a los mensajes de solicitud de conexión en caso de ser necesario y registrando siempre los mensajes en la bitácora.
- Se debe incluir un módulo que registre en las bitácoras todas las conexiones y desconexiones del sistema.
- Es necesario integrar un módulo que permita registrar, en un medio diferente a la base de datos, las desconexiones debidas a fallas en la base de datos.

Inicio/Finalización de la aplicación.

- Se requiere que la consola cuente con la opción para que los usuarios con los privilegios necesarios puedan arrancar el aplicativo.
- Se requiere que la consola cuente con la opción para detener el aplicativo siguiendo el orden especificado. Sólo a el (los) usuario(s) autorizados se le(s) permitirá realizar esta tarea.

Monitoreo del aplicativo.

- Se requiere contar con mecanismos que permitan conocer el estado del aplicativo, número de conexiones establecidas a él y operación ejecutándose en cada conexión

Respaldo de la base de datos.

- Se requiere contar con mecanismos que permitan realizar el respaldo de la base de datos

Administración de Usuarios



- Se requiere contar con mecanismos que permitan realizar la administración de los usuarios que tendrán acceso a la consola de monitoreo del aplicativo.
- Se requiere contar con mecanismos que permitan realizar la administración de los usuarios administrativos que tendrán acceso a la aplicación.

3.4.2 Requerimientos de uso

3.4.2.1 Para la captura.

- Se requiere capacitar a los profesores y directivos encargados de los datos y calificaciones de los alumnos.
- Se requiere capacitar al usuario o usuarios en la operación de la consola (PC) para captura de usuarios ubicada en la supervisión.

3.4.2.2 Para las consultas

- Se requiere capacitar a los profesores, directivos encargados de los datos y calificaciones de los alumnos, también es pertinente crear un mecanismo para facilitar el uso a los padres de familia
- Se requiere especificar los requerimientos de la computadora en hardware y software a utilizar para la correcta visualización del programa.
- Se requiere monitorear el uso de la aplicación

3.4.2.3 Para la administración de la aplicación.

- Se requiere capacitar al usuario o usuarios en la operación de la consola (PC) para administración ubicada en la supervisión.

3.4.3 Requerimientos de confiabilidad.

3.4.3.1 Para la captura.

- Las terminales de captura deben operar de manera ininterrumpida por lo menos durante 2 horas.
- La transmisión de datos se debe realizar sobre una red de datos segura con protocolo TCP/IP.



- Se debe de poder recuperar de una perdida de datos en la base de datos.
- Los mensajes de comunicación del aplicativo central a las terminales son de carácter informativo, no se permite modificar datos desde el aplicativo central.
- Se requiere incorporar estándares de seguridad informática.
- Solo el usuario asignado dentro de cada escuela y en la supervisión podrá imprimir la información correspondiente a los listados y reportes del sistema
- La información que se registre en la base de datos debe provenir de una transacción firmada por las personas que intervienen en el proceso de captura, si no es así se debe de rechazar.

3.4.3.2 Para las consultas

- Se requiere validar la integridad de la información distribuida.
- Se requiere un mecanismo de replicación de la información de las bases de datos transparente a los usuarios.

3.4.4 Requerimientos de rendimiento.

- Se debe de poder procesar 1000 transacciones concurrentes.
- Se requiere contar con un ancho de banda que garantice la comunicación de 1000 equipos hacia el aplicativo central.

3.4.5 Requerimientos de soporte

- Se debe de comentar el código generado.
- Se debe de establecer una nomenclatura a utilizar en los diferentes módulos o componentes desarrollados.
- Casos de uso definidos para el sistema.
- Se debe almacenar toda la documentación y código que forme parte del Sistema de Administración Escolar
- Se debe generar un documento de arquitectura de la solución
- Se debe generar un diagrama de componentes de la aplicación.



- Se debe generar un diccionario de datos de la base del SIADE
- Se debe generar un diagrama entidad relación de la base de datos del SIADE.
- Se debe generar un glosario con los términos empleados en el desarrollo de la aplicación.

3.4.6 Restricciones de diseño

- El lenguaje de programación que se propone es PHP.
- El protocolo de comunicación a utilizar entre las terminales y el aplicativo central es TCP/IP
- Se deben de establecer conexiones permanentes entre los equipos remotos y el módulo de comunicación central.
- Se propone el uso de la base de datos MySQL en la versión 5.0
- El Sistema Operativo a emplearse es Linux.
- Se requiere conocer el tamaño de paquetes permitidos en la transmisión de información de las terminales.

3.4.7 Requerimientos de documentación.

Manual de usuario para el SAC

.

Manual de operación de las consola de monitoreo.

Manual de contingencias.

Manual técnico

Documentación del sistema (diagramas UML)

3.4.8 Interfaces

Interfaz de usuario

Pantalla de Autenticación



Pantalla de presentación de datos 100 por ciento HTML

Pantalla de captura de datos de profesores

Pantalla de captura de datos del alumno

Módulo con documentos y comunicados de la supervisión.

Pantalla de reportes y gráficos solicitados

Consola de Administración y monitoreo de procesos.

Pantallas de consulta de datos

Interfaz de comunicaciones

Protocolo TCP/IP.

Estándares

PHP 5.0 o posterior

Plantillas de RUP.

Especificaciones UML 2.0

Estándares de seguridad (MD5 y SSL)

3.5 Reglas de negocio

Las reglas de negocio son las siguientes:

- El servicio es para uso exclusivo del personal docente administrativo, directivo, personal docente frente a grupo y padres de familia de las escuelas pertenecientes a la supervisión escolar P095.
- La consulta de los datos académicos de alumnos solo se podrán realizar 15 días después de la fecha de firma de boletas en la escuela primaria, esto con el fin de no propiciar el ausentismo en las juntas bimestrales entre padres de familia con los docentes frente a grupo y este documento no tendrá validez oficial a menos que este sellado por la escuela y la supervisión escolar.
- El supervisor solo podrá consultar los datos con el fin de validar y concatenar información enviada de las diferentes escuelas.



- El supervisor solo podrá modificar la información en caso de existir algún error en la captura y para poder corregir se necesita la firma electrónica de 2 docentes administrativos de la supervisión.
- Los reportes solo podrán ser consultados para fines estadísticos y por el personal autorizado, estos son los profesores frente a grupo, directivos para cada escuela y el personal administrativo de la supervisión para los reportes globales.
- Toda alta fuera del periodo de inscripciones estará sujeta a disponibilidad de la escuela y tendrá que ser avalada por el director escolar.
- La captura de calificaciones escolares se realizara 2 días antes de la entrega de calificaciones parciales a los padres de familia.
- La baja de un alumno tendrá que ir acompañada de una justificación por parte del profesor, esta baja será reportada por el directivo a la supervisión a través del sistema.
- Los trámites continuaran siendo gratuitos.
- La información de los alumnos tendrá el carácter de confidencial y no se le podrá dar otro uso.
- La información del personal docente tendrá el carácter de confidencial y no se le podrá dar otro uso.

3.6 Casos de uso

El principal propósito de los casos de uso para el análisis del Sistema de Administración Escolar de los alumnos de las 11 primarias pertenecientes a la Supervisión escolar P095 es ayudar a visualizar los requerimientos funcionales, incluyendo la relación de los actores que interactúan y los procesos esenciales para desarrollar los objetivos, los actores identificados son:

Actor	Descripción
Director	Vigilar que el personal se presente en tiempo y forma, verificar que la estructura del edificio este en optimas condiciones, verificar que se lleven a cabo los planes y programas de estudio, vigilar que las evaluaciones bimestrales se realicen en los tiempos establecidos, vigilar los recursos materiales con que cuenta la institución y que estos se utilicen adecuadamente, entregar documentación oficial en el tiempo establecido, atender las problemáticas de los alumnos y padres de familia y tratar de dar solución, asistir y coordinar las Reuniones de



	Consejo Técnico
Supervisor	Gestionar el mejoramiento de infraestructura de las escuelas pertenecientes a la zona, revisión periódica para verificar que se realicen las actividades planeadas, entrega de documentación oficial a directores y recopilación de la misma ya que fue procesada para enviarla al departamento regional o al departamento estatal, concatenación y validación de la información recibida de las escuelas, vigilar que los talleres generales se lleven a cabo en tiempo y en forma, atender problemáticas que no se puedan solucionar en la escuela con directivos, profesores y padres de familia.
Profesor	Ejecutar los planes y programas de estudio, evaluar a los alumnos y reportar las calificaciones a padres de familia y directivos, realizar acciones para el fortalecimiento del conocimiento, asistir a las reuniones del Consejo Técnico.
Padre de Familia	Consulta periódicamente las evaluaciones y aprovechamiento de su hijo o hijos que a su vez son alumnos de alguna escuela perteneciente a la supervisión

Alumnos:

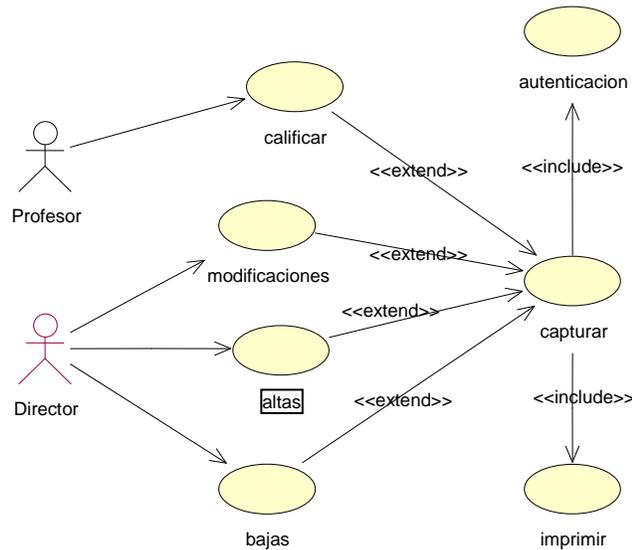
Caso de Uso	Capturar				
Id. del CU	UC_Capturar	Prioridad	Alta	Estado	En elaboración
Actores Participantes e intereses	Director: Busca registrar, eliminar o modificar los datos de los alumnos Profesor: Registra las calificaciones los alumnos.				
Breve Descripción	Permite capturar un registro en la base de datos de alumnos de la escuela.				
Precondiciones	El Usuario debe ser personal docente y debe haberse autenticado en el sistema.				
Post condiciones	Al finalizar se deben imprimir los movimientos realizados.				
Flujo Principal	El docente se conecta a la pagina de la aplicación Web en la Internet El docente se autentifica e ingresa al sistema El sistema le muestra el menú de acciones que puede realizar El docente elige la actividad a realizar (alta, baja, cambio) El sistema presenta la información perteneciente a la actividad El docente captura la información en los formularios				
Flujos Alternos	FA-1 Alta El docente selecciona la opción de ALTA El sistema muestra el formulario para ser llenado El docente llena el formulario El sistema le muestra al usuario la información que digitó como confirmación del correcto registro El sistema informa si se desea realizar otra alta o salir de la opción El docente decide entre las dos opciones anteriores. Si se decide dar otra alta se reinicia el flujo alternativo Si se decide salir de la opción, el caso de uso termina. FA-2 Baja				



	<p>El docente selecciona la opción BAJA El sistema muestra el campo para realizar la búsqueda del registro. El docente llena los campos y realiza la baja El sistema le muestra al usuario la información que digitó como confirmación del correcto registro. Tras la confirmación, el sistema marca el registro como baja pero se conserva en la base de datos. El sistema informa si se desea realizar otra baja o salir de la opción El docente decide entre las dos opciones anteriores. Si se decide realizar otra baja se reinicia el flujo alterno Si se decide salir de la opción, el caso de uso termina.</p> <p><u>FA-3 Modificaciones</u> El docente selecciona la opción MODIFICACIONES El sistema muestra las opciones que puede elegir el usuario El docente elige en el sistema la opción con la que desea trabajar El sistema muestra el campo para realizar la búsqueda del registro a modificar El docente modifica los campos deseados y realiza el movimiento El sistema le muestra al usuario la información que digitó como confirmación del correcto registro. El docente confirma si la información es correcta El sistema solicita la firma electrónica del docente y de una persona autorizada más El sistema pregunta si se desea realizar otro movimiento o salir de la opción El usuario decide entre las dos opciones anteriores. Si se decide realizar otra modificación se reinicia el flujo alterno. Si se decide salir de la opción, el caso de uso termina. El sistema regresa a la página principal del usuario</p> <p><u>FA-4 Calificar</u> El profesor se conecta a la página de la aplicación Web El profesor se autentica e ingresa en el sistema El sistema le muestra las operaciones que puede realizar El profesor selecciona en el sistema la acción capturar calificaciones El sistema muestra el formulario para continuar con la operación seleccionada El profesor llena los campos y valida la operación a realizar El sistema muestra la información de la operación y la correcta ejecución de la misma El profesor solicita salir del sistema. El sistema se desconecta de la aplicación y se conecta a la página de inicio del SACSE.</p>
Flujos de excepción	<p><u>E-2 No se ingresar al sistema</u> El sistema notifica en un lenguaje no técnico la razón por el usuario no pudo acceder al sistema.</p> <p><u>E-1 No se puede elegir opción</u> El sistema notifica en un lenguaje no técnico la razón por la cuál no se puede elegir determinada opción del sistema.</p>
Reglas de negocio relacionadas	<p>Toda alta fuera del periodo de inscripciones estará sujeta a disponibilidad de la escuela y tendrá que ser avalada por el supervisor y el director escolar. La captura de calificaciones escolares se realizara 2 días antes de la entrega de calificaciones parciales a los padres de familia.</p>
Autor	Julio Cesar Cruz Cruz
Fecha	09/10/06



Diagrama de caso de uso



Caso de Uso	Consultar				
Id. del CU	UC Consultar	Prioridad	Alta	Estado	En elaboración
Actores Participantes e intereses	Director: Busca consultar los datos de los profesores de su institución Supervisor: Busca consultar los datos de los profesores, validar la captura Profesor: Busca consultar su información laboral Padre de Familia: Busca consultar la información académica de un alumno				
Breve Descripción	Permite consultar un registro en la base de datos de alumnos inscritos en alguna escuela perteneciente a la supervisión escolar.				
Precondiciones	El Usuario debe haberse autenticado en el sistema.				
Post condiciones	Se realiza la consulta de los registros solicitados				
Flujo Principal	El docente se conecta a la página de la aplicación. El docente se autentifica e ingresa al sistema El sistema le muestra la operación de consulta que puede realizar El usuario ejecuta la operación El sistema le muestra los resultados de la consulta				
Flujos Alternos	FA-1 Alumnos El usuario selecciona la opción de consultar ALUMNO El sistema muestra las formas de consulta del alumno El usuario llena el formulario El sistema le muestra al usuario la información solicitada E-2 El sistema muestra la opción de realizar otra consulta o salir El usuario decide entre las dos opciones anteriores. Si se decide realizar otra consulta se reinicia el flujo alternativo Si se decide salir de la opción, el flujo alternativo termina. FA-2 Profesores El usuario selecciona la opción consultar PROFESORES El sistema muestra el campo para realizar la búsqueda del registro El sistema muestra las formas de consulta del profesor				



	<p>El usuario llena el formulario El sistema le muestra al usuario la información solicitada E-2 El sistema muestra la opción de realizar otra consulta o salir El usuario decide entre las dos opciones anteriores. Si se decide realizar otra consulta se reinicia el flujo alterno Si se decide salir de la opción, el flujo alterno termina. FA-3 Validar El supervisor selecciona la opción de validar información El sistema muestra las opciones de escuela, turno y grupo a validar El supervisor elige la información a validar. Si existe algún error en la captura, el supervisor marca el registro como incorrecto y escribe un comentario relacionado al error. El usuario termina la revisión, imprime el reporte y sale de la opción. El sistema pregunta si se desea salir o realizar una nueva validación. Si se decide realizar otra consulta se reinicia el flujo alterno Si se decide salir de la opción, el flujo alterno termina.</p>
<p>Flujos de excepción</p>	<p>E-1 No se puede elegir opción El sistema notifica en un lenguaje no técnico la razón por la cuál no se puede elegir determinada opción del sistema. E-2 No se ingresar al sistema El sistema notifica en un lenguaje no técnico la razón por el usuario no pudo acceder al sistema.</p>
<p>Reglas de negocio relacionadas</p>	<p>El supervisor solo podrá modificar la información en caso de existir algún error en la captura y para poder corregir se necesita la firma electrónica de 2 personas del personal administrativas de la supervisión.</p>
<p>Autor</p>	<p>Julio Cesar Cruz Cruz</p>
<p>Fecha</p>	<p>09/10/06</p>
<p>Diagrama de caso de uso</p>	<pre>graph TD S((Supervisor)) --> V((validar)) D((Director)) --> A((alumnos)) P((Profesor)) --> A PF((Padre de Familia)) --> A V --> C((consultar)) A --> C C --> I((imprimir)) C --> AU((autenticacion)) AU --> C C -.-> <<extend>> V A -.-> <<extend>> C C -.-> <<include>> I AU -.-> <<include>> C</pre>



Módulo Profesores

Caso de Uso		Capturar			
Id. del CU	UC Capturar	Prioridad	Alta	Estado	En elaboración
Actores Participantes e intereses	Director: Busca registrar, eliminar o modificar los datos del personal docente (clases especiales) Supervisor: Busca capturar la información del personal administrativo de la supervisión personal docente de cada una de las escuelas.				
Breve Descripción	Permite capturar un registro en la base de datos del personal docente de las escuelas y la supervisión.				
Precondiciones	El Usuario debe ser personal docente y debe haberse autenticado en el sistema.				
Post condiciones	Al finalizar se deben imprimir los movimientos realizados.				
Flujo Principal	<p>El docente se conecta a la pagina de la aplicación Web en la Internet</p> <p>El docente se autentifica e ingresa al sistema</p> <p>El sistema le muestra el menú de acciones que puede realizar</p> <p>El docente elige la actividad a realizar (alta, baja, cambio)</p> <p>El sistema presenta la información perteneciente a la actividad</p> <p>El docente captura la información en los formularios</p>				
Flujos Alternos	<p><u>FA-1 Alta</u></p> <p>El docente selecciona la opción de ALTA</p> <p>El sistema muestra el formulario para ser llenado</p> <p>El docente llena el formulario</p> <p>El sistema le muestra al usuario la información que digitó como confirmación del correcto registro.</p> <p>El sistema informa si se desea realizar otra alta o salir de la opción</p> <p>El docente decide entre las dos opciones anteriores.</p> <p>Si se decide dar otra alta se reinicia el flujo alterno</p> <p>Si se decide salir de la opción, el caso de uso termina.</p> <p><u>FA-2 Baja</u></p> <p>El docente selecciona la opción BAJA</p> <p>El sistema muestra el campo para realizar la búsqueda del registro.</p> <p>El docente llena los campos y realiza la baja</p> <p>El sistema le muestra al usuario la información que digitó como confirmación del correcto registro.</p> <p>Tras la confirmación, el sistema marca el registro como baja pero se conserva en la base de datos.</p> <p>El sistema informa si se desea realizar otra baja o salir de la opción</p> <p>El docente decide entre las dos opciones anteriores.</p> <p>Si se decide realizar otra baja se reinicia el flujo alterno</p> <p>Si se decide salir de la opción, el caso de uso termina.</p> <p><u>FA-3 Modificaciones</u></p> <p>El docente selecciona la opción MODIFICACIONES</p> <p>El sistema muestra las opciones que puede elegir el usuario</p> <p>El docente elige en el sistema la opción con la que desea trabajar</p> <p>El sistema muestra el campo para realizar la búsqueda del registro a modificar</p> <p>El docente modifica los campos deseados y realiza el movimiento</p> <p>El sistema le muestra al usuario la información que digitó como confirmación del correcto registro.</p> <p>El docente confirma si la información es correcta</p> <p>El sistema solicita la firma electrónica del docente y de una persona autorizada mas</p> <p>El sistema pregunta si se desea realizar otro movimiento o salir de la opción</p> <p>El usuario decide entre las dos opciones anteriores.</p>				

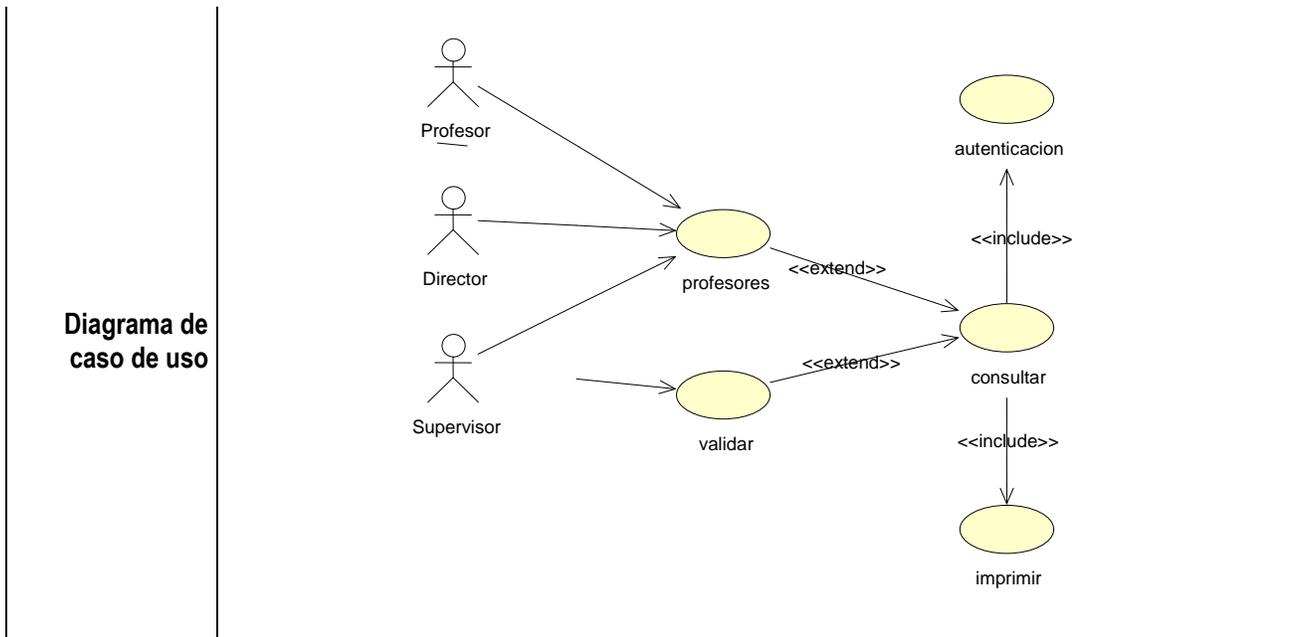


	<p>Si se decide realizar otra modificación se reinicia el flujo alterno. Si se decide salir de la opción, el caso de uso termina. El sistema regresa a la página principal del usuario.</p>
Flujos de excepción	<p>E-2 No se ingresar al sistema El sistema notifica en un lenguaje no técnico la razón por el usuario no pudo acceder al sistema. E-1 No se puede elegir opción El sistema notifica en un lenguaje no técnico la razón por la cuál no se puede elegir determinada opción del sistema.</p>
Reglas de negocio relacionadas	<p>La información del personal docente tendrá el carácter de confidencial y no se le podrá dar otro uso.</p>
Autor	<p>Julio Cesar Cruz Cruz</p>
Fecha	<p>09/10/06</p>
Diagrama de caso de uso	<pre> graph TD Director((Director)) Supervisor((Supervisor)) modificaciones((modificaciones)) altas((altas)) bajas((bajas)) capturar((capturar)) autentificación((autentificación)) imprimir((imprimir)) Director --> modificaciones Director --> altas Director --> bajas Supervisor --> bajas modificaciones --> <<extend>> capturar altas --> <<extend>> capturar bajas --> <<extend>> capturar capturar --> <<include>> autentificación capturar --> <<include>> imprimir </pre>

Caso de Uso	Consultar				
Id. del CU	UC Consultar	Prioridad	Alta	Estado	En elaboración
Actores Participantes e intereses	Director: Busca consultar los datos de los profesores de su institución Supervisor: Busca consultar los datos de los profesores y validar la captura Profesor: Busca consultar su información laboral				
Breve Descripción	Permite consultar un registro en la base de datos de profesores de la supervisión.				
Precondiciones	El Usuario debe haberse autentificado en el sistema.				
Post condiciones	Se realiza la consulta de los registros solicitados				

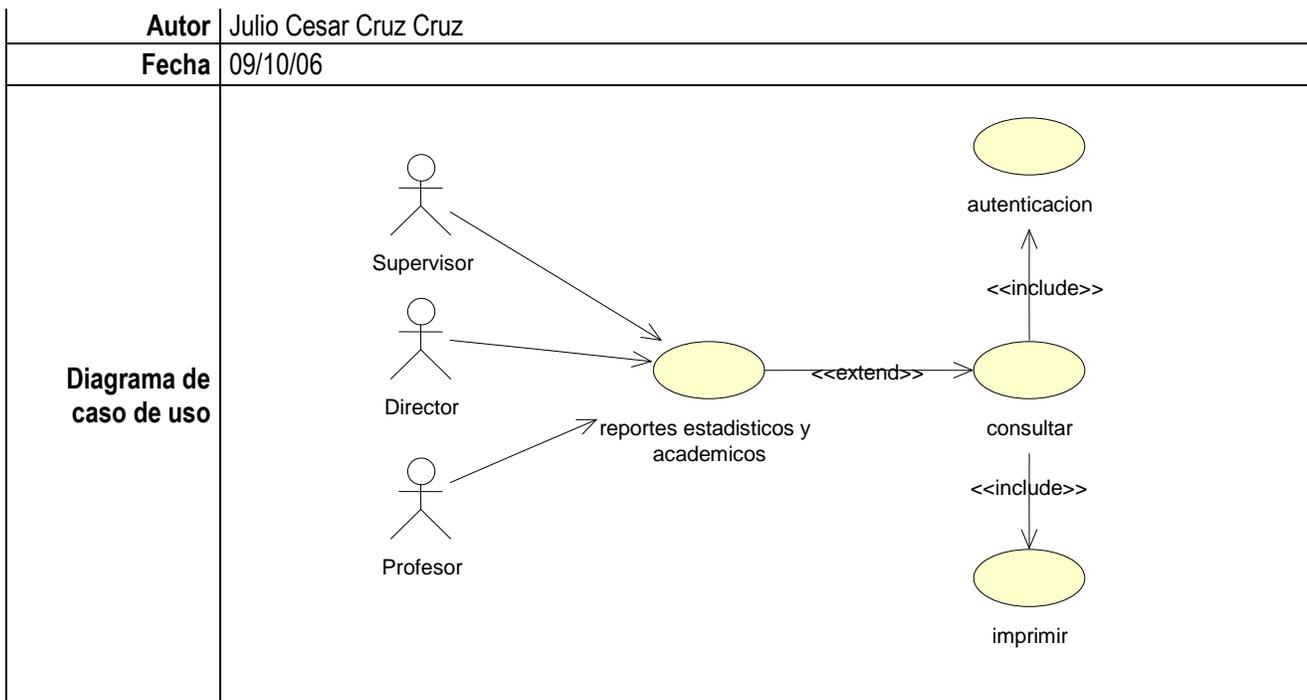


Flujo Principal	<p>El docente se conecta a la página de la aplicación. El docente se autentifica e ingresa al sistema El sistema le muestra la operación de consulta que puede realizar El usuario ejecuta la operación El sistema le muestra los resultados de la consulta</p>
Flujos Alternos	<p><u>FA-1 Alumnos</u> El usuario selecciona la opción de consultar ALUMNO El sistema muestra las formas de consulta del alumno El usuario llena el formulario El sistema le muestra al usuario la información solicitada E-2 El sistema muestra la opción de realizar otra consulta o salir El usuario decide entre las dos opciones anteriores. Si se decide realizar otra consulta se reinicia el flujo alterno Si se decide salir de la opción, el flujo alterno termina.</p> <p><u>FA-2 Profesores</u> El usuario selecciona la opción consultar PROFESORES El sistema muestra el campo para realizar la búsqueda del registro El sistema muestra las formas de consulta del profesor El usuario llena el formulario El sistema le muestra al usuario la información solicitada E-2 El sistema muestra la opción de realizar otra consulta o salir El usuario decide entre las dos opciones anteriores. Si se decide realizar otra consulta se reinicia el flujo alterno Si se decide salir de la opción, el flujo alterno termina.</p> <p><u>FA-3 Validar</u> El supervisor selecciona la opción de validar información El sistema muestra las opciones de escuela, turno y grupo a validar El supervisor elige la información a validar. Si existe algún error en la captura, el supervisor marca el registro como incorrecto y escribe un comentario relacionado al error. El usuario termina la revisión, imprime el reporte y sale de la opción. El sistema pregunta si se desea salir o realizar una nueva validación. Si se decide realizar otra consulta se reinicia el flujo alterno Si se decide salir de la opción, el flujo alterno termina.</p>
Flujos de excepción	<p><u>E-1 No se puede elegir opción</u> El sistema notifica en un lenguaje no técnico la razón por la cuál no se puede elegir determinada opción del sistema.</p> <p><u>E-2 No se ingresar al sistema</u> El sistema notifica en un lenguaje no técnico la razón por el usuario no pudo acceder al sistema.</p>
Reglas de negocio relacionadas	<p>El supervisor solo podrá modificar la información en caso de existir algún error en la captura y para poder corregir se necesita la firma electrónica de 2 personas del personal administrativas de la supervisión.</p>
Autor	<p>Julio Cesar Cruz Cruz</p>
Fecha	<p>09/10/06</p>



Módulo Reportes

Caso de Uso	Reportes				
Id. del CU	UC_REPORTES	Prioridad	<u>Alta</u>	Estado	En elaboración
Actor principal	Supervisor, Director, Profesor.				
Actores participantes	Supervisor: Busca obtener reportes estadísticos y de aprovechamiento académico de la supervisión, y concatenar la información de las escuelas para enviarla a la oficinas regionales. Director: Busca obtener reportes estadísticos y de aprovechamiento académico de una escuela Profesor: Busca obtener reportes estadísticos y de aprovechamiento académico de un grupo.				
Objetivo	Obtener información estadística y académica de la supervisión escolar				
Precondiciones	El usuario se autentica para poder ingresar al sistema y poder obtener los reportes.				
Flujo Principal	El usuario se autentica e ingresa en el sistema El sistema le muestra el menú de reportes que puede obtener El usuario selecciona en el sistema la acción a realizar El sistema muestra la información de la operación El usuario solicita salir del sistema. El sistema pregunta si desea salir del sistema o consultar otra información. Si el usuario decide				
Flujos de excepción	E-1 No se pudo obtener el reporte El sistema notifica en un lenguaje no técnico la razón por la cuál no se pudo realizar la consulta de algún alumno o grupo de alumnos. E-2 No se ingresar al sistema El sistema notifica en un lenguaje no técnico la razón por el usuario no pudo acceder al sistema.				
Reglas de negocio relacionadas	- Los reportes solo podrán ser consultados para fines estadísticos y por el personal autorizado, estos son los profesores frente a grupo, directivos para cada escuela y el personal administrativo de la supervisión para los reportes globales. - La información de los alumnos tendrá el carácter de confidencial y no se le podrá dar otro uso.				



Módulo Administración

Caso de Uso	Gestión de Usuarios				
Id. del CU	UC_GEST_USR	Prioridad	<u>Alta</u>	Estado	En elaboración
Actor principal	Administrador del sistema				
Actores participantes	Administrador del sistema: Busca darle mantenimiento al catalogo de usuarios; registrando, actualizando o borrando los datos y privilegios de los usuarios en el sistema.				
Objetivo	Administra los usuarios de la consola de administración del sistema				
Breve Descripción	Administra los usuarios para la el uso de la aplicación y para el monitoreo de la aplicación, asigna privilegios				
Precondiciones	El administrador del sistema se autentica para poder interesarse al sistema y darle mantenimiento al catalogo de usuarios Las páginas de este caso de uso deberán contar con seguridad de acceso mas restringido a las de los otros casos de uso				
Flujo Principal	<p>El administrador del sistema se autentica e ingresa en el sistema</p> <p>El sistema le muestra el menú de operaciones que puede realizar</p> <p>El usuario selecciona en el sistema la acción a realizar</p> <p>El sistema muestra el formulario para continuar con la operación seleccionada</p> <p>El administrador llena el formulario y valida la operación a realizar</p> <p>El sistema muestra la información de la operación y la correcta ejecución de la misma</p> <p>El sistema envía una solicitud de salida o de otra operación</p> <p>El sistema se desconecta de la aplicación y regresa a la pagina de inicio del SACSE o reinicia el caso de uso</p>				



<p>Flujos de excepción</p>	<p><u>E-1 No se pudo realizar el movimiento administrativo</u> El sistema notifica en un lenguaje no técnico la razón por la cuál no se pudo realizar la petición en el sistema.</p>
<p>Reglas de negocio relacionadas</p>	
<p>Autor</p>	<p>Julio Cesar Cruz Cruz</p>
<p>Fecha</p>	<p>09/10/06</p>
<p>Diagrama de caso de uso</p>	

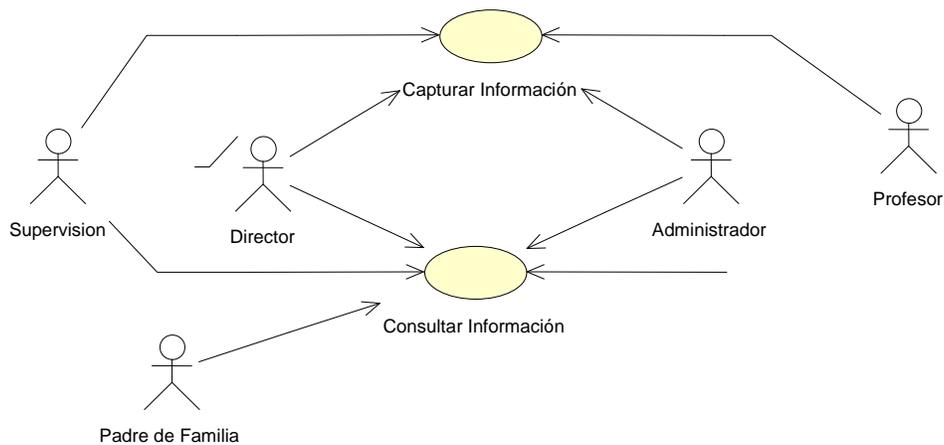


Figura 3.2.- Diagrama General de Casos de Uso



Capítulo 4

Diseño del sistema

Una vez que se ha finalizado la fase del análisis, el siguiente paso es el diseño del sistema. En este diseño del sistema se aplican ciertas técnicas y principios con el propósito de definir procesos, con suficiente detalle como para permitir su interpretación y realización física, es la etapa donde se va a buscar la forma de que los requerimientos, métodos, procesos y toda la información que se recabo en el análisis sea implementada buscando la mejor forma de conjuntar los elementos, como el análisis, el diseño también cuenta con diferentes metodologías, la estructurada, la orientada a objetos entre las principales que se utilizan para llevar a cabo esta fase, dentro de este capítulo se explicarán los pasos del diseño con UML.



4.1 Propuesta de solución

Se propuso un sistema Web que automatice y sea capaz de gestionar toda la información generada en las áreas pertenecientes al control escolar, principalmente deberá proporcionar una reducción significativa de tiempo y recursos materiales, financieros y humanos, también tiene como propósito que la información se obtenga de manera casi inmediata y libre de errores, que esté siempre disponible para todo aquel que este interesado, que se mantenga la confidencialidad de los resultados y que solamente sea proporcionada la información a las personas autorizadas, esto con el fin de tener un buen desempeño y funcionamiento de la Institución.

A continuación se muestra en la figura el diagrama de flujo del sistema Automatizado de Control para una Supervisión Escolar.

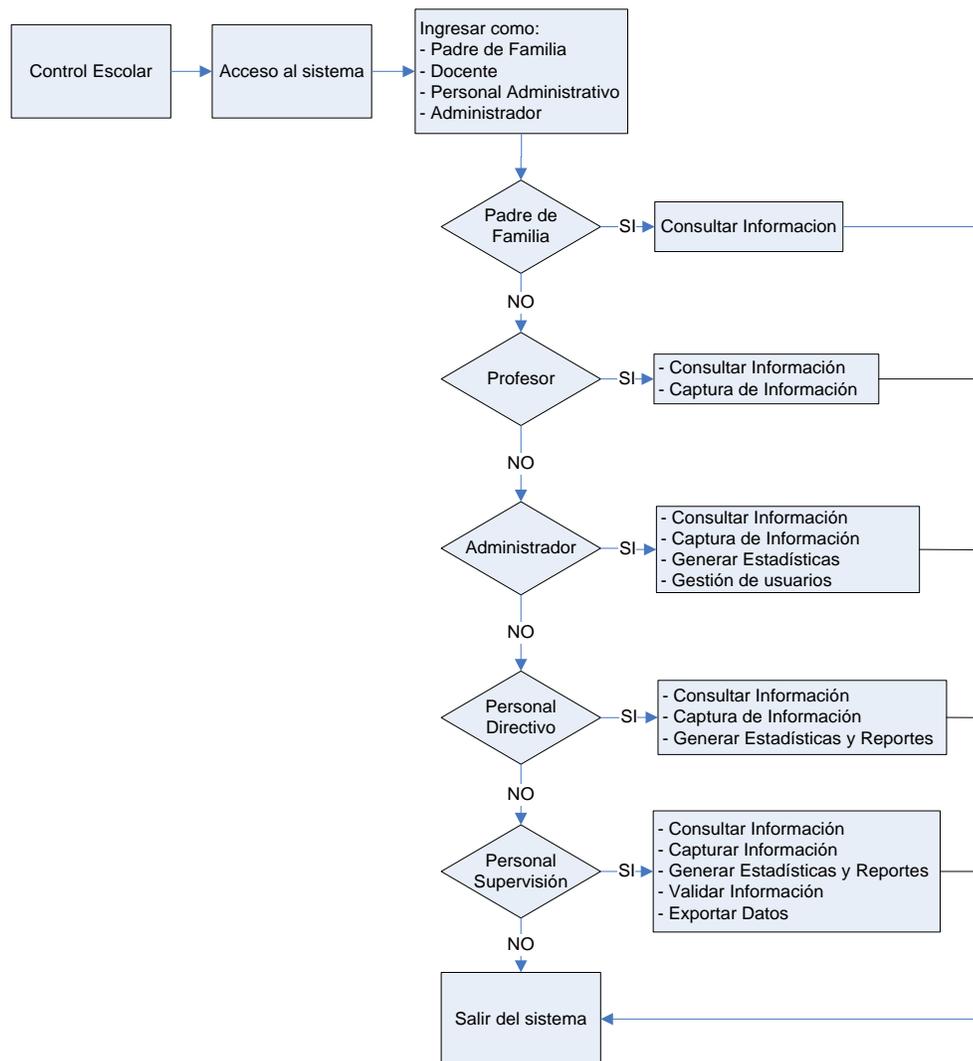


Figura 4.1.- Diagrama de flujo del sistema Automatizado de Control para una Supervisión Escolar



4.2 Diseño conceptual

En el diseño conceptual se modelan los requerimientos del usuario, este es completamente independiente de los aspectos de implementación, como puede ser el SGBD que se vaya a usar, los programas de aplicación, los lenguajes de programación, el hardware disponible o cualquier otra consideración física.

En esta fase del diseño se debe dar solución a los problemas que plantean las especificaciones y proponer un modelo de producto global que realice las funciones necesarias para dar servicio al usuario.

El diseño conceptual trata el producto en forma global, es decir que no se pretende entrar en el diseño en detalle de los subsistemas y componentes del producto, sino que nos debemos centrar en la forma y las funciones principales. No obstante en ocasiones y con el fin de seleccionar alternativas se puede hacer necesario la realización detallada de alguno de los subsistemas.

De cualquier forma, las propuestas que surgen tras la etapa de diseño conceptual debe definir el producto completamente, aunque no sea en detalle. Se deben definir en la medida de lo posible los subsistemas que incorpora, las funciones que realiza, su apariencia general y posibles soluciones a aquellas características novedosas.

4.2.1 Diagramas de clases conceptuales del sistema

Un buen modelo conceptual capta las abstracciones esenciales y la información indispensable para comprender el dominio dentro del contexto de los requerimientos actuales; nos facilita además conocer el dominio: sus conceptos, su terminología y sus relaciones, por ello, el modelo conceptual abarca también la descripción de clases significativas en un dominio del problema.

La utilización de clases conceptuales de especificación corresponde a la necesidad de conceptualizar objetos del modelo del dominio, para que sean descripciones o especificaciones de los elementos que se manejan en el problema el cual se le trata de dar solución, en nuestro caso, la sistematización del control escolar para una supervisión.

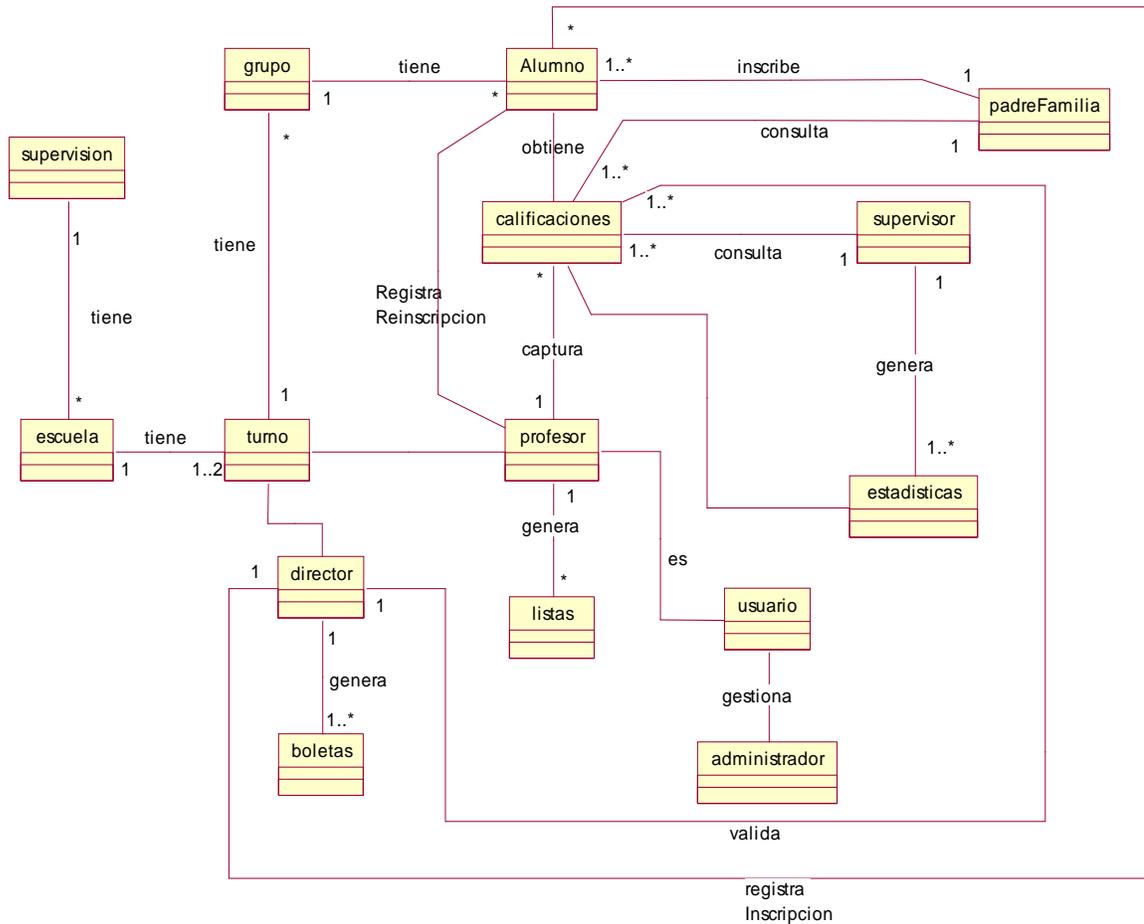


Figura 4.2.- Diagrama que muestra las clases conceptuales del sistema.

4.2.2 Diagramas de secuencia

El uso del diagrama de secuencias para el sistema, es una forma conveniente de estudiar y definir el comportamiento de este como una caja negra, tal comportamiento es una descripción de qué es lo que se hace, sin explicar como lo hace.

La creación de este diagrama es una manera rápida y fácil que muestra los eventos de entrada y salida en el orden en que los generan los actores del sistema, relacionados con el escenario de un evento específico descrito por los casos de uso que se elaboraron anteriormente.

A continuación se describen los siguientes diagramas de secuencia:



- Gestión de usuarios
- Altas, Bajas, Modificaciones
- Consulta de información
- Calificar
- Reportes

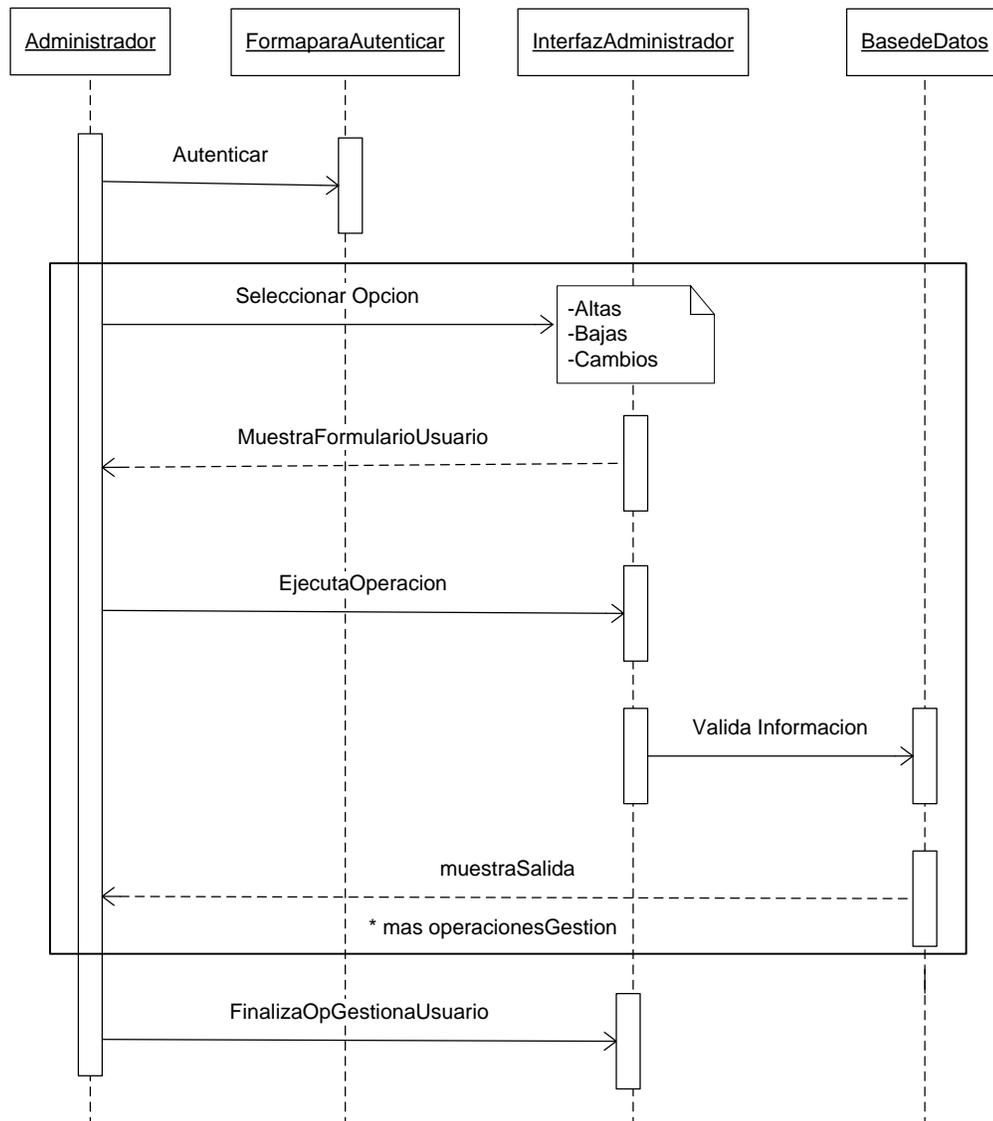


Figura 4.3.- Diagrama de secuencia que muestra el caso de uso Gestión de usuarios por parte del Administrador de Sistemas

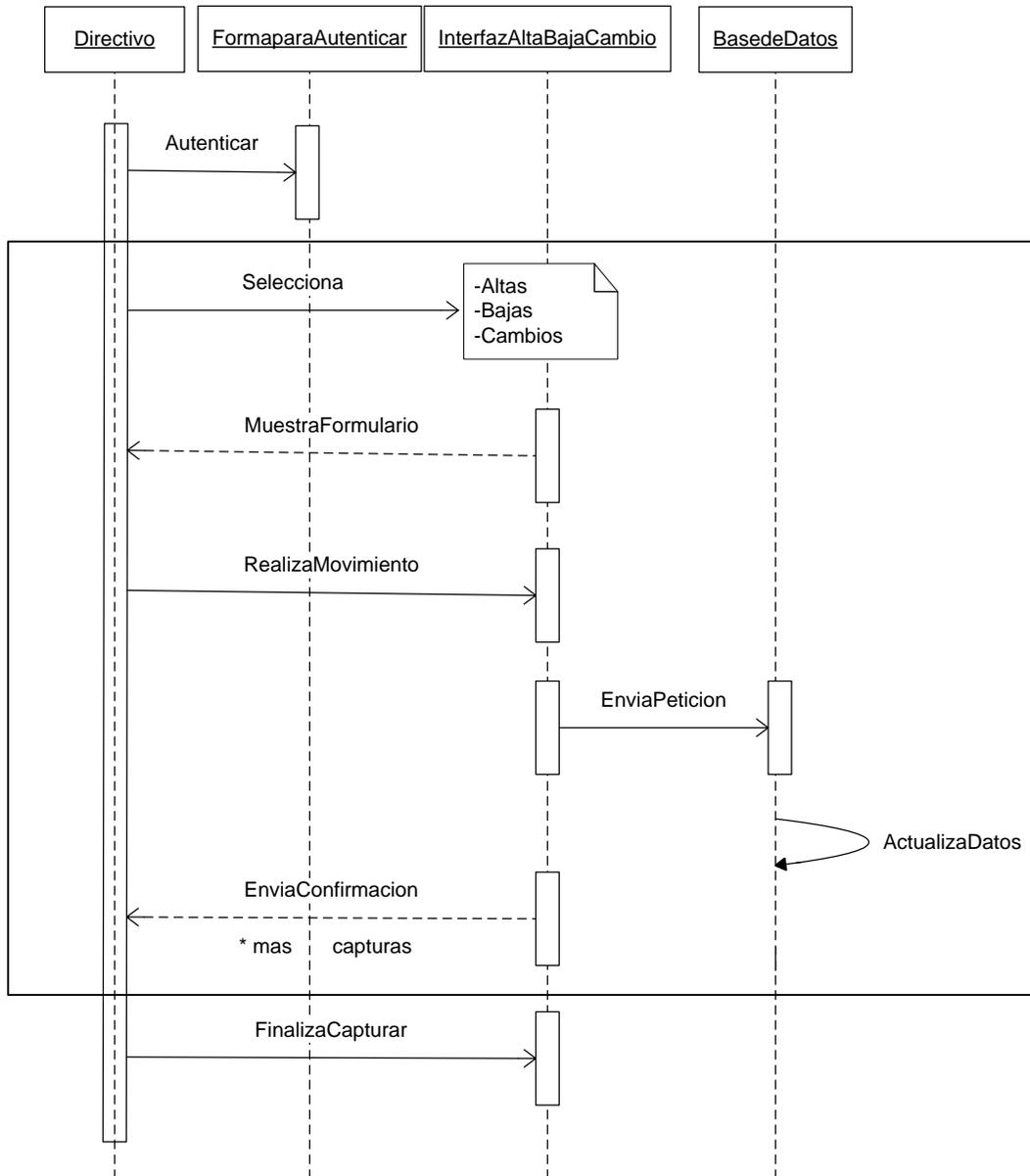


Figura 4.4.- Diagrama de secuencia que muestra el caso de uso Capturar por parte de un personal Directivo.

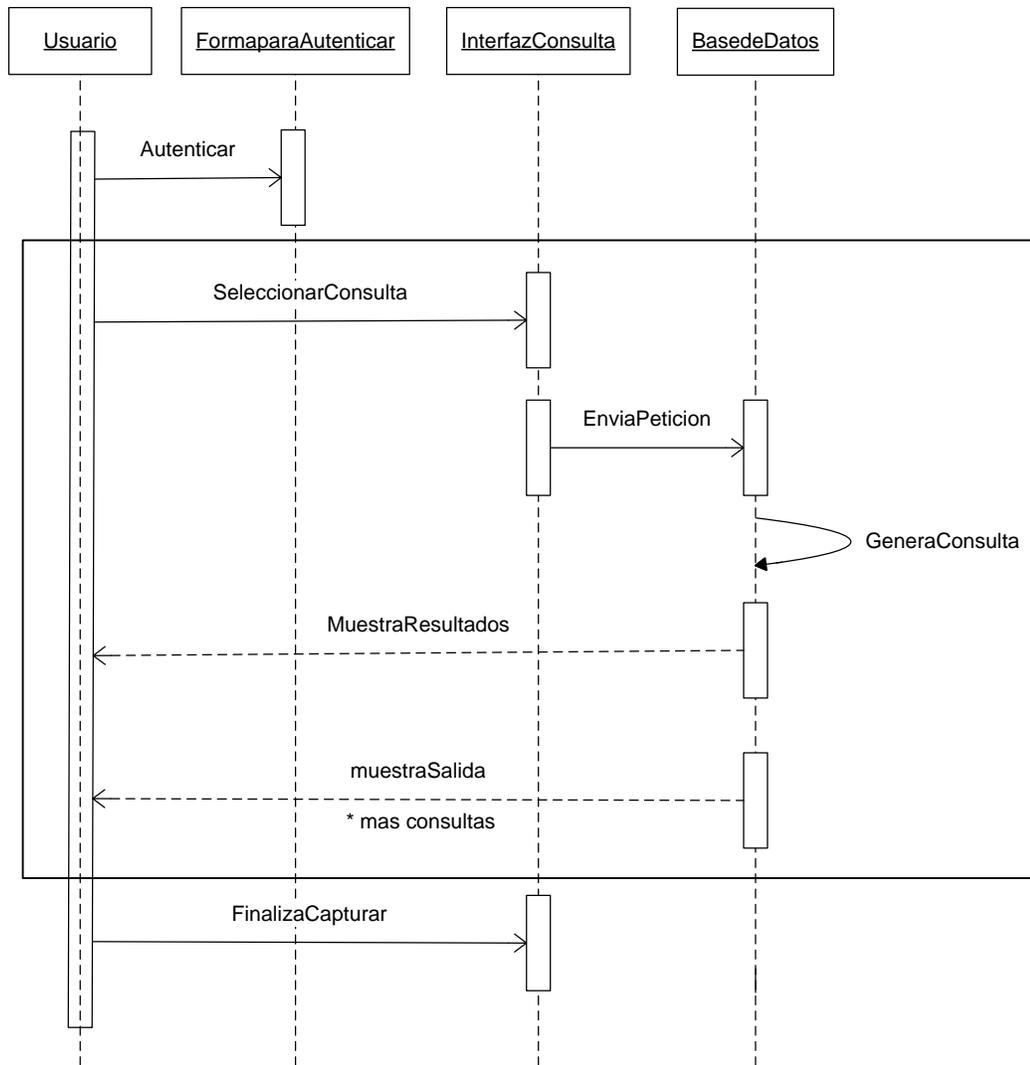


Figura 4.5.- Diagrama de secuencia que muestra el caso de uso Consulta.

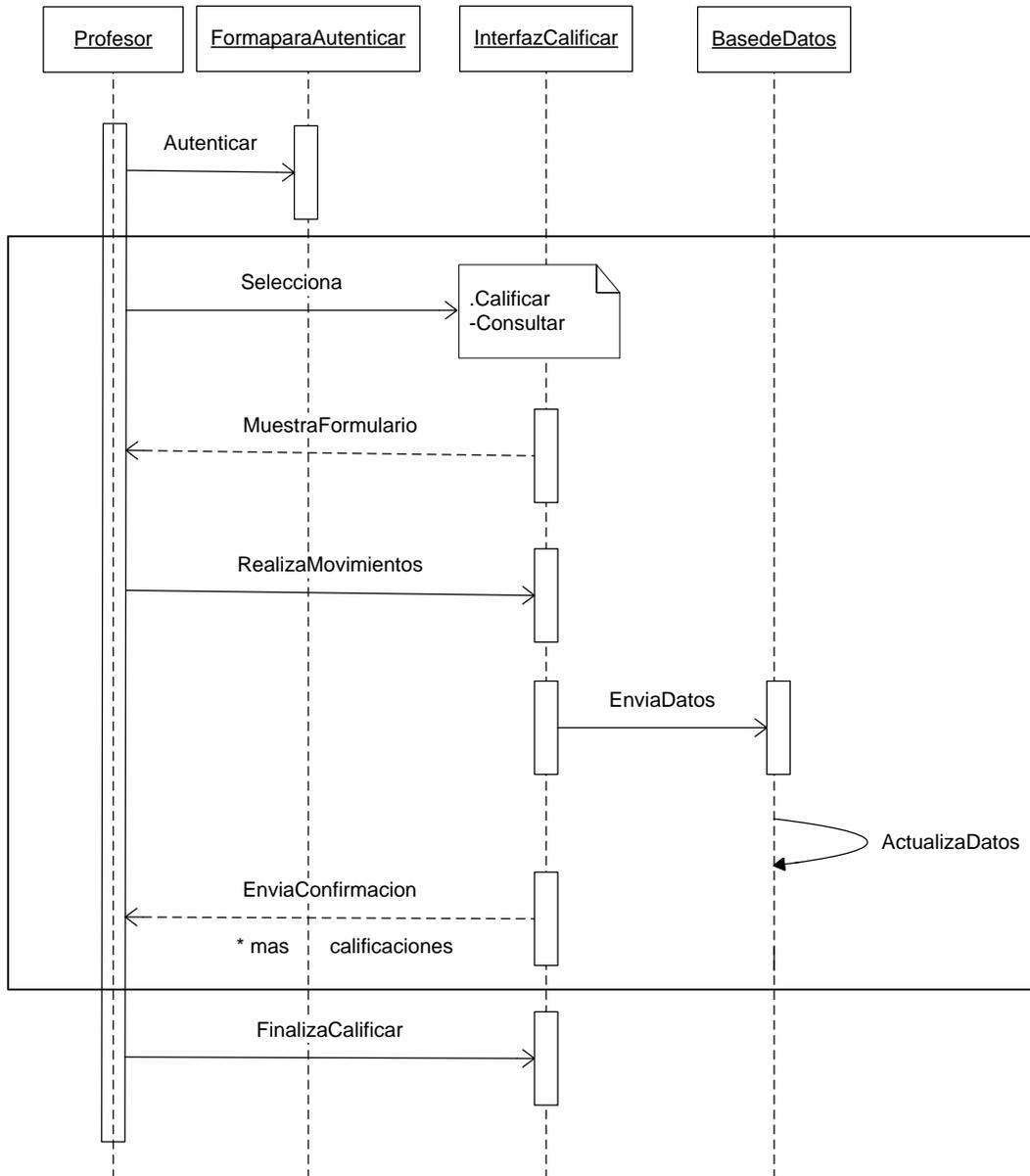


Figura 4.6.- Diagrama de secuencia que muestra el caso de uso Calificar por parte de un Profesor.

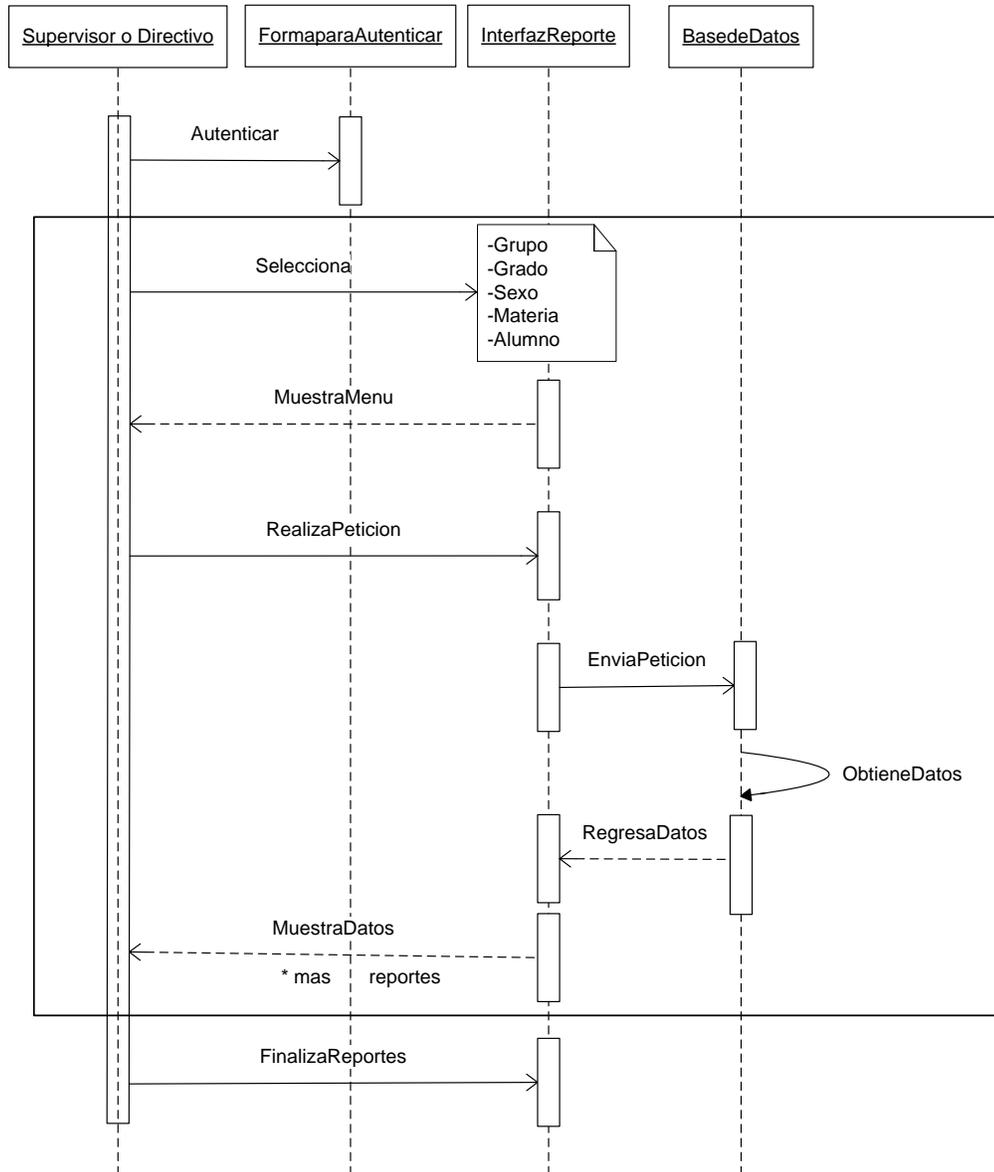


Figura 4.7.- Diagrama de secuencia que muestra el caso de uso Reportes.



4.3.2 Capa de datos

La capa de datos tiene como misión la administración de la información que maneja el sistema, esto incluye el almacenamiento, la actualización y la recuperación de todos los datos contenidos en el sistema. En la práctica, esta capa es esencialmente un DBMS y el esquema de datos propio de la aplicación.

4.3.2.1 DBMS

MySQL es un manejador de base de datos relacional, muy rápido (preferentemente con tablas menores a 1,000,000 de registros), confiable y fácil de usar, es software de fuente abierta (*Open Source*³⁹), aunque también se puede comprar una versión comercial licenciada.

Su conectividad, velocidad y seguridad hacen a MySQL altamente satisfactorio para acceder bases de datos en Internet.

MySQL soporta un amplio subconjunto de ANSI SQL 99, y varias extensiones más.

4.3.2.2 Modelo Entidad-Relación para la base de datos del sistema.

El uso de un modelo Entidad-Relación es necesario debido a que a través de este, se pueden identificar las entidades, atributos y las relaciones que aparecen en el sistema, así como identificar las llaves primarias y foráneas.

³⁹ Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo, cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License)

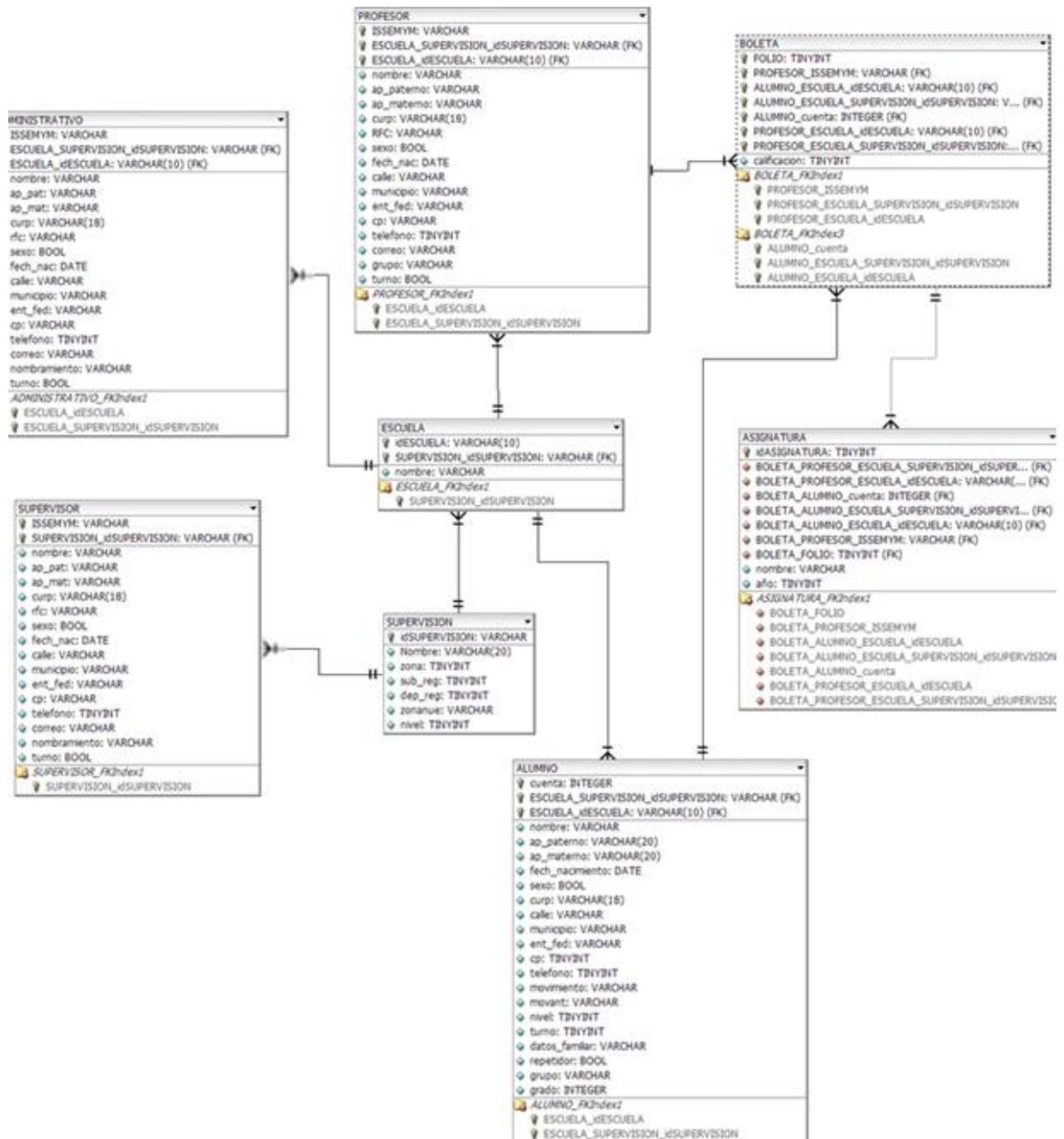


Figura 4.9.- Diagrama Entidad-Relación.



4.3.3 Capa de proceso

En la capa de proceso o de lógica del negocio se modela el comportamiento del sistema, basándose en los datos provistos por la capa de datos y actualizándolos según sea necesario. Esta capa describe los distintos procesos de negocio que tienen lugar en la organización.

El comportamiento de la aplicación esta definida por los componentes que modelan la lógica del negocio. Estos componentes reciben las acciones a realizar a través de la capa de presentación, y llevan a cabo las tareas necesarias utilizando la capa de datos para manipular la información del sistema.

Se dice que esta capa es el “corazón” de cualquier aplicación, es aquí donde se codifican las consultas concretas para manejar los datos y los pasa a la capa de datos para el procesamiento del motor de la base de datos. Después de que la capa de datos ha resuelto una petición, este es entregado a la capa de presentación. En cuanto el usuario elija una acción, se devuelve el identificador de la acción con los datos correspondientes a la capa de proceso. Antes de que la acción sea procesada se realiza la validación de los datos con las reglas de negocio, se hace algún pre-procesamiento y se envía nuevamente la consulta a la capa de datos.

A partir del modelo de datos que presenta los objetos del sistema y las relaciones existentes entre estos, se procede con una especificación funcional, es decir, con el establecimiento de que deberá hacer el sistema.

4.3.3.1 Autenticación

En toda aplicación y más en una aplicación Web como lo es nuestro caso de estudio, resulta fundamental la autenticación en la que se desee restringir lo que los usuarios pueden hacer. Esta autenticación permite conocer la identidad de quienes se conectan al servidor, la autorización permite verificar los privilegios que tiene asignado cada usuario.

Una vez detectada la necesidad de autenticar a los usuarios, surge la necesidad de decidir como se llevara a cabo esta actividad. Al menos se dispone de dos métodos:

1. Mediante las tablas de usuario de la base de datos, para mayor nivel de seguridad no guardaremos la clave o firma digital, sino que guardaremos un valor HASH⁴⁰ que nos servirá para comprobar si la clave es "buena". Para este caso, necesitaremos una pequeña utilidad para generar ese valor HASH.

⁴⁰ La firma digital se transforma a través de una función hash en un número único que es decodificado utilizando el mismo algoritmo y posteriormente verificado con el agente autenticador que almacena estos datos.



2. Construyendo una tabla propia de perfiles de usuario, para un buen nivel de seguridad, se recomienda que las claves se guarden como valores HASH, no guardadas directamente, el problema en este caso es que ni nosotros podremos saber la clave del usuario, salvo que en lugar de dejar que se genere automáticamente usemos una utilidad como la indicada en el punto anterior.

Se recomienda cualquiera de los dos anteriores, pero se hace énfasis en que el valor de la contraseña se cifre con un valor HASH.

4.3.3.2 Aplicar Reglas de Validación.

Es importante hacer validaciones tanto para la captura como para la consulta de los datos, por lo que se sugiere:

1. Captura: La información de los alumnos y personal docente es lo más importante, así que la información relacionada con estos, debe estar actualizada y debe de estar completa, por lo que si no tiene los datos mínimos necesarios para la consulta, se levantará una excepción que no permitirá que pueda ser registrado, con lo que evitaremos tener registros incompletos.

La captura de los alumnos y personal docente corre a cargo principalmente de dos figuras: el Director y el Supervisor. Una de las principales reglas que se deben tener en cuenta, es que los datos mínimos necesarios de los formularios estén debidamente requisitados.

La validación con respecto a que información puede registrar cada uno de los usuarios, tiene las siguientes reglas:

- El registro de los alumnos corre a cargo de los directores de la escuela y no por parte del personal de la supervisión.
 - Los registro de los profesores frente a clase solo lo podrá realizar el personal de la supervisión escolar, a los profesores de clases especiales los podrá registrar el director.
2. Validación: En la validación solo se analizan los formularios si los usuarios que van a validar la información aplican cambios a la información en el momento de validarla.
 3. Consulta: dependiendo del usuario tendrá permisos y privilegios para consultar determinados apartados, con las siguientes reglas:
 - El personal de supervisión puede consultar la información de todas las escuelas pertenecientes a la zona escolar
 - El personal directivo podrá consultar la información de alumnos y profesores de una escuela determinada.



- El padre de familia solo puede consultar la información de su o sus hijos inscritos en alguna escuela perteneciente a la supervisión escolar.

4.3.3.3 Apache, Javascript y PHP.

El servidor de aplicaciones Apache proporciona varias opciones para el desarrollo de aplicaciones Web, haciendo uso de varios lenguajes de programación y protocolos de comunicación, principalmente el HTTP.

A través de la publicación de la información, utilizando páginas de contenido estático y páginas de contenido dinámico es como se construyen las aplicaciones.

Las solicitudes a través de páginas dinámicas se ejecutan del lado del servidor y las respuestas son enviadas del lado del cliente en el navegador. Apache soporta una gran variedad de tecnologías para emplear y desarrollar páginas dinámicas incluyendo las aplicaciones CGI, Perl Scripts, PHP, Python, etc. Por esta razón y por usabilidad se recomendó para la implementación de este proyecto.

Para programar la lógica de la aplicación se recomienda de PHP, que es un lenguaje Open Source interpretado de alto nivel, fue creado para desarrollar aplicaciones para Internet, y llegó a ser muy popular debido a que el PHP puede ser incrustado en páginas HTML y tiene una sintaxis fácil de aprender.

Con el paso del tiempo, fueron surgiendo nuevas necesidades y requerimientos; como el crear aplicaciones más robustas y complejas. Fue entonces que la idea de mezclar el PHP con HTML ya no era suficiente. Por otro lado, la programación orientada a objetos llegó a tomar protagonismo y se mostró como la solución a muchos problemas en la programación

Java, como lenguaje símbolo en la programación orientada a objetos, tomó fuerza en los últimos años, pero aún parece que la complejidad de Java no se justifica en muchas aplicaciones Web. Pues ya PHP4, con un modelo orientado a objetos bastante limitado, ofrecía lo necesario para programar usando algunas de las ventajas de los objetos. Pero el mayor cambio, sin lugar a duda, lo trae PHP5. Con una orientación a objetos bastante parecida al Java, PHP5 presenta mejoras significativas y un ambiente de programación orientada a objetos mucho más completo que permitirán que PHP proporcione un alto rendimiento a las aplicaciones Web empresariales al nivel de las plataformas J2EE y .NET.

Finalmente, como los programas o scripts de JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos, etc., se convierte en el lenguaje ideal para realizar las validaciones en los formularios de nuestra aplicación.



4.3.4 Capa de Presentación

Esta capa contiene todos los elementos que constituyen la interfaz del usuario. Esta capa contiene todo aquello con lo que el usuario puede interactuar, por ejemplo las ventanas, los cuadros de dialogo, el modelo de navegación del sistema y los adaptadores para el modelo de acceso para el cliente como lo es el navegador.

4.3.4.1 Modelo de navegación con respecto a perfiles de usuario

La navegación es un paso crítico en la implementación de la capa de presentación. El diseño navegacional o mapa de navegación nos ayudará a visualizar los casos específicos o subprocesos así como la secuencia de la aplicación y actividades que se realizaran en cada uno de los casos.

La ventaja de diseñar la navegación con respecto a los perfiles de usuario es la personalización de la aplicación para establecer la funcionalidad o secciones de una aplicación que utilizará un usuario en específico.

La definición de los perfiles de usuario se hace con base en la descripción de la función de cada actor o usuario.

En la siguiente figura, se muestra el mapa de navegación general del Sistema Automatizado de Control para una Supervisión Escolar. Se puede observar que inicia con la pantalla de bienvenida en donde se presentan las opciones de seleccionar el tipo de usuario, una vez que se selecciono el tipo de usuario, este tendrá que autenticarse para poder acceder a los servicios a que tiene derecho, posteriormente que se ha autenticado, se muestra una nueva pantalla que tiene diferentes opciones a elegir.

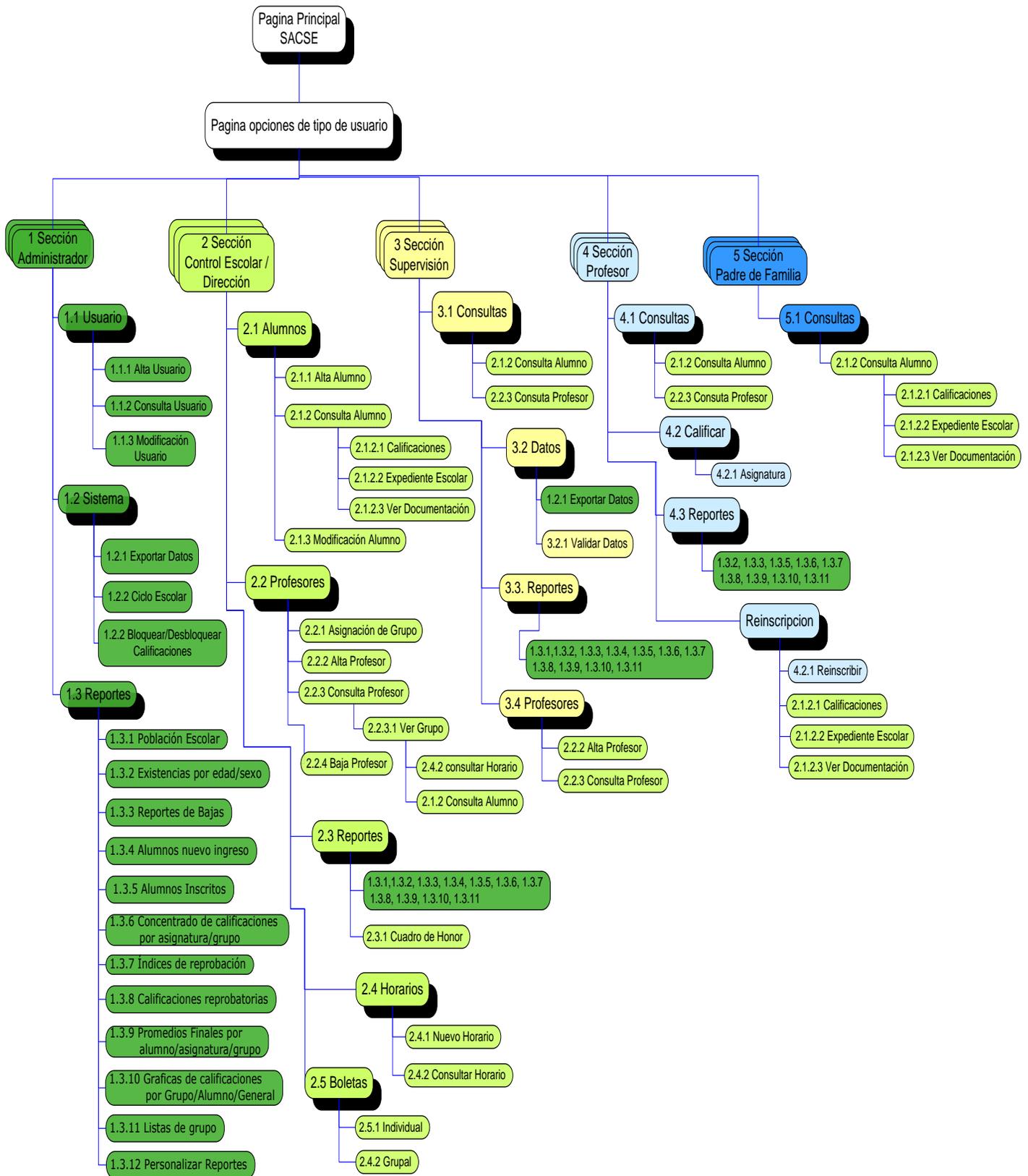


Figura 4.10.- Mapa de navegación general del Sistema Automatizado de Control



4.3.4.2 Interfaz de usuario

La interfaz es el medio a través del cual el usuario tiene acceso a la información y puede interactuar con los datos, esta interfaz será presentada en el navegador por medio de páginas elaboradas dinámicamente en lenguaje HTML.

Es importante mencionar la importante participación del usuario final en esta etapa ya que la interfaz se diseñará con base a las necesidades y tareas del usuario, de tal forma que resulte amigable y de fácil manejo. Para lograr lo anterior es necesario tener en cuenta las siguientes recomendaciones[6]:

- Usabilidad: esto es la facilidad de uso
- Accesibilidad: Se refiere a la posibilidad de acceso, es decir, que posibilite el acceso a todos sus potenciales usuarios
- Arquitectura de la información: la usabilidad de la aplicación depende no sólo del diseño del interfaz, sino también de su arquitectura, estructura y organización , en otras palabras, del componente no visible del diseño, aunque es importante señalar que es el diseño el que permite integrar las estructuras de datos.

Una vez que se ha decidido el modelo navegacional de la aplicación Web, una buena práctica para la interfaz de usuario es utilizar prototipos de la interfaz que son documentos, diseños o sistemas que simulan o tienen implementadas partes del sistema final a desarrollar. Los prototipos son cruciales para diseñar un buen sitio Web, facilitan la planificación del proceso de creación, reducen el coste de las evaluaciones, aumentan su efectividad y evitan graves errores en el diseño.

El solo propósito de crear estos prototipos es dar la oportunidad de evaluar el diseño prematuramente. El objetivo es producir estos prototipos rápidamente y evaluarlos eficazmente para que puedan ser refinados, elaborados y reevaluados antes del producto final.

Se utilizaron los storyboard que ayudaron a visualizar las relaciones entre la aplicación con las páginas Web con las cuales interactúa, para la creación de los prototipos de la interfaz del usuario se utilizó la herramienta de Macromedia Dreamweaver, ya que es un programa flexible, de fácil manejo, compatible y con gran potencial, se puede elaborar una página Web de forma rápida sin necesidad de escribir código, pero principalmente por la facilidad de crear aplicaciones dinámicas con bases de datos.

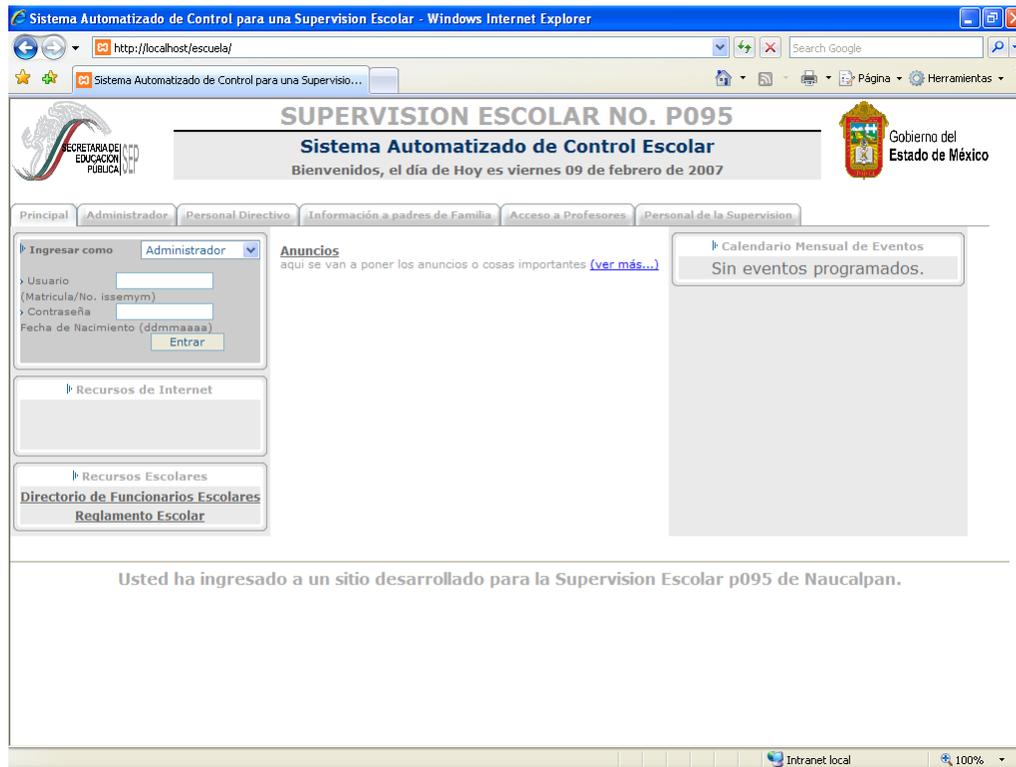


Figura 4.11.- Página de Inicio del Sistema Automatizado de Control

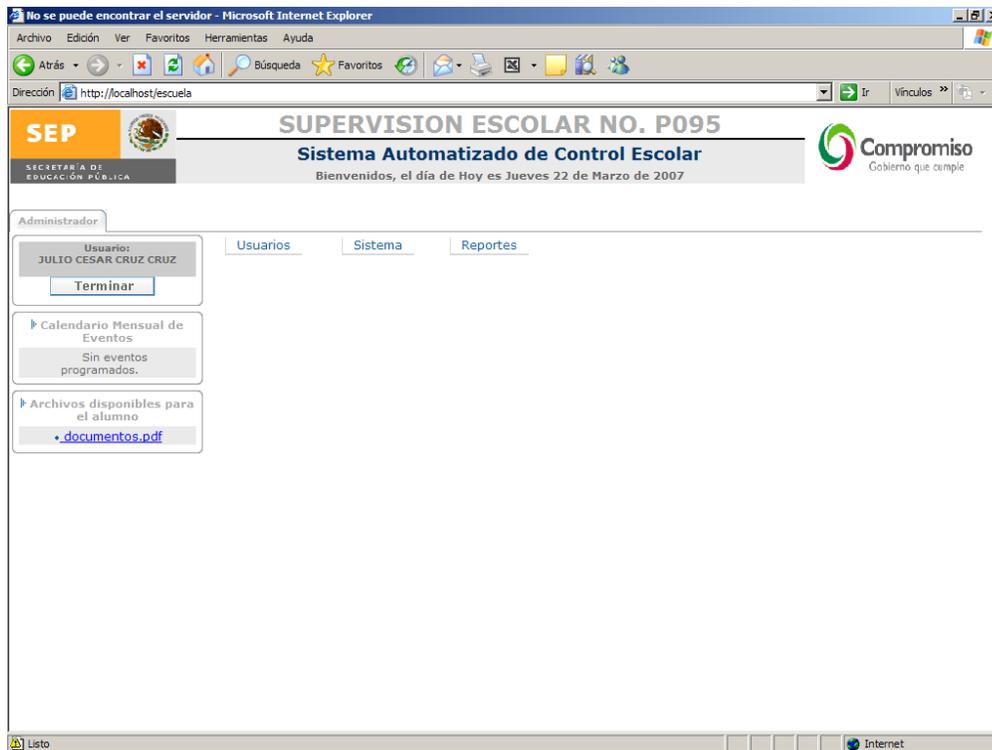


Figura 4.12.- Página de inicio del modulo de Administrador

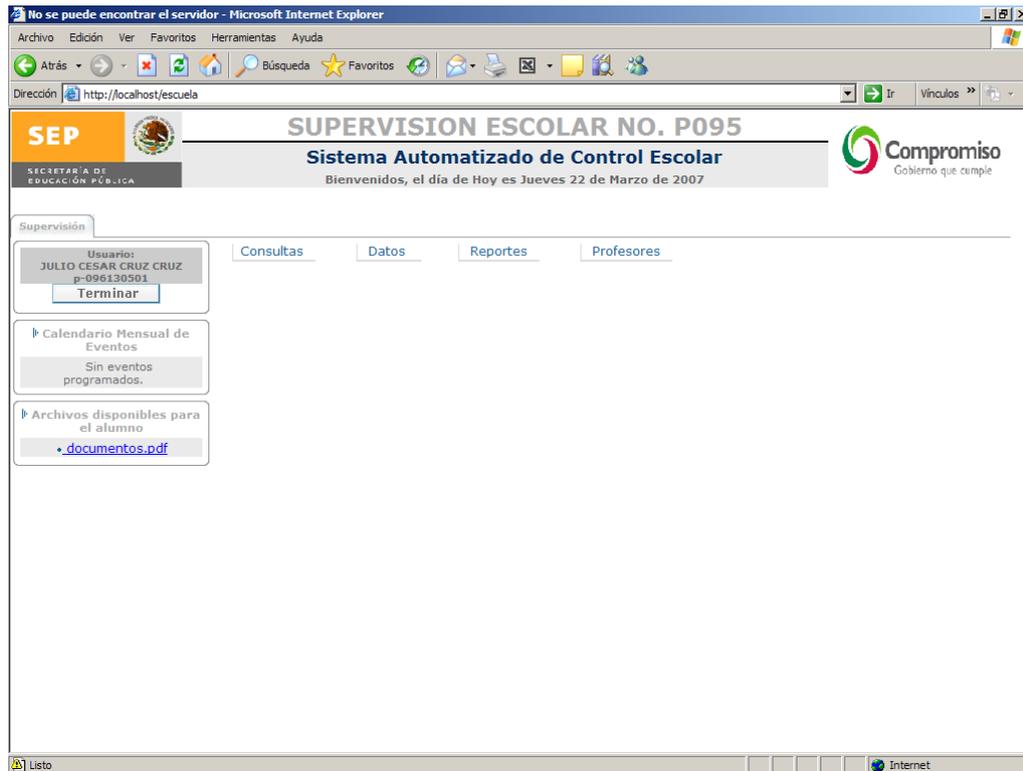


Figura 4.13.- Página de inicio del modulo de la Supervisión

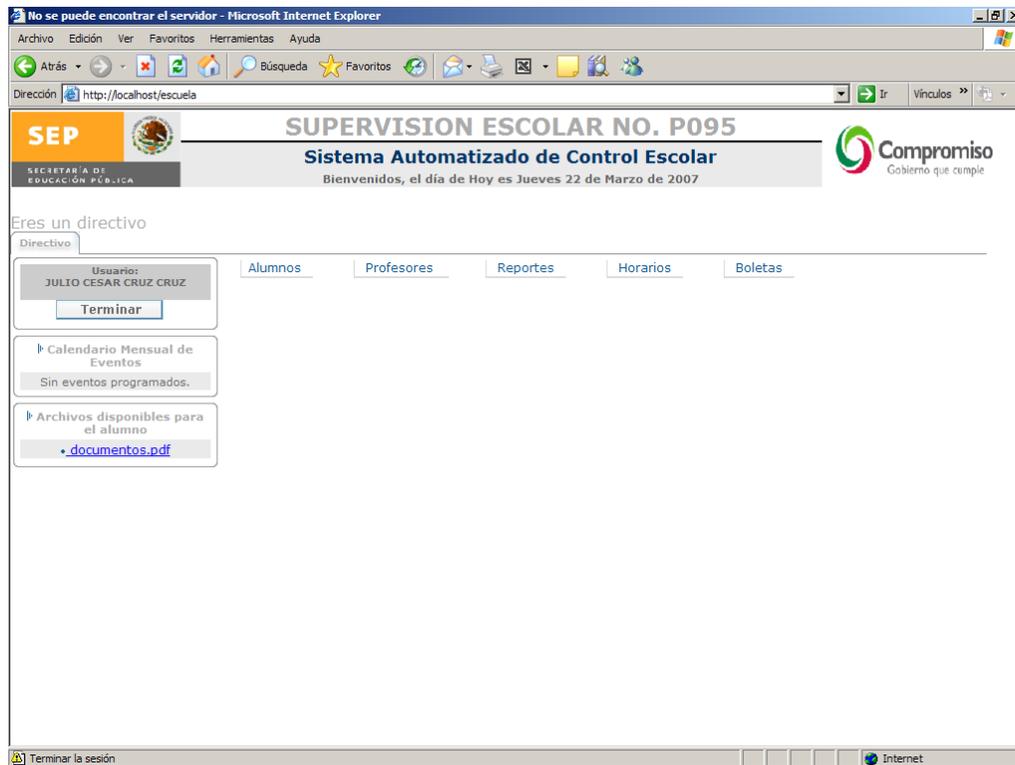


Figura 4.14.- Página de inicio del modulo del personal directivo o de control escolar

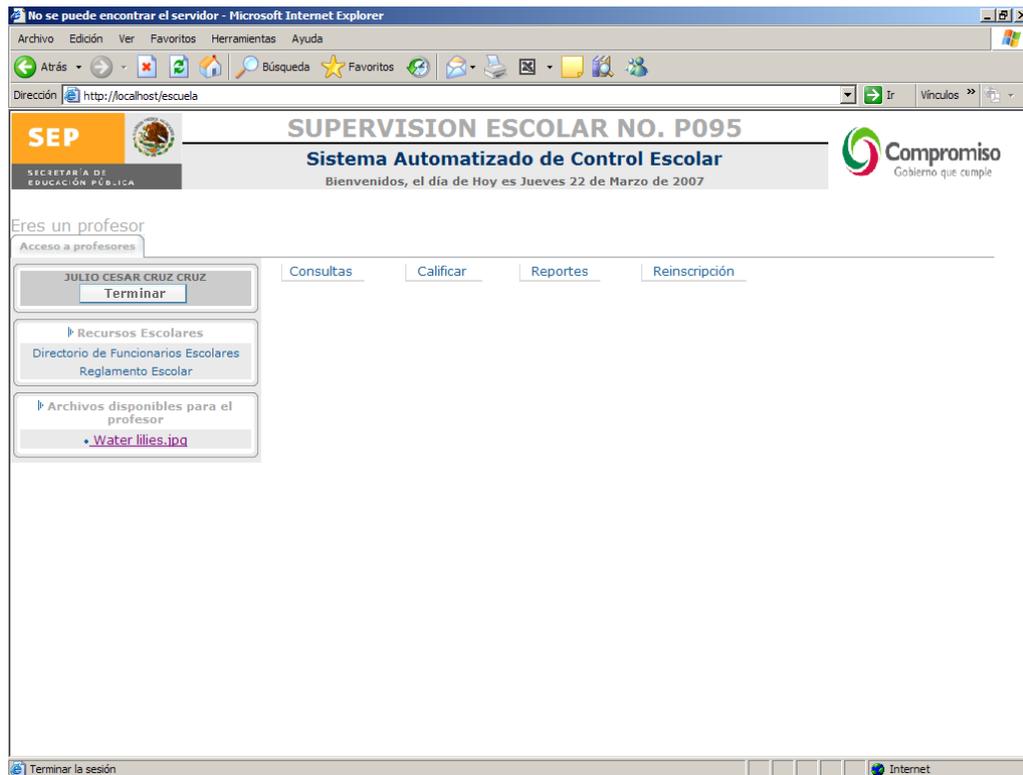


Figura 4.18.- Página de inicio del modulo de profesores

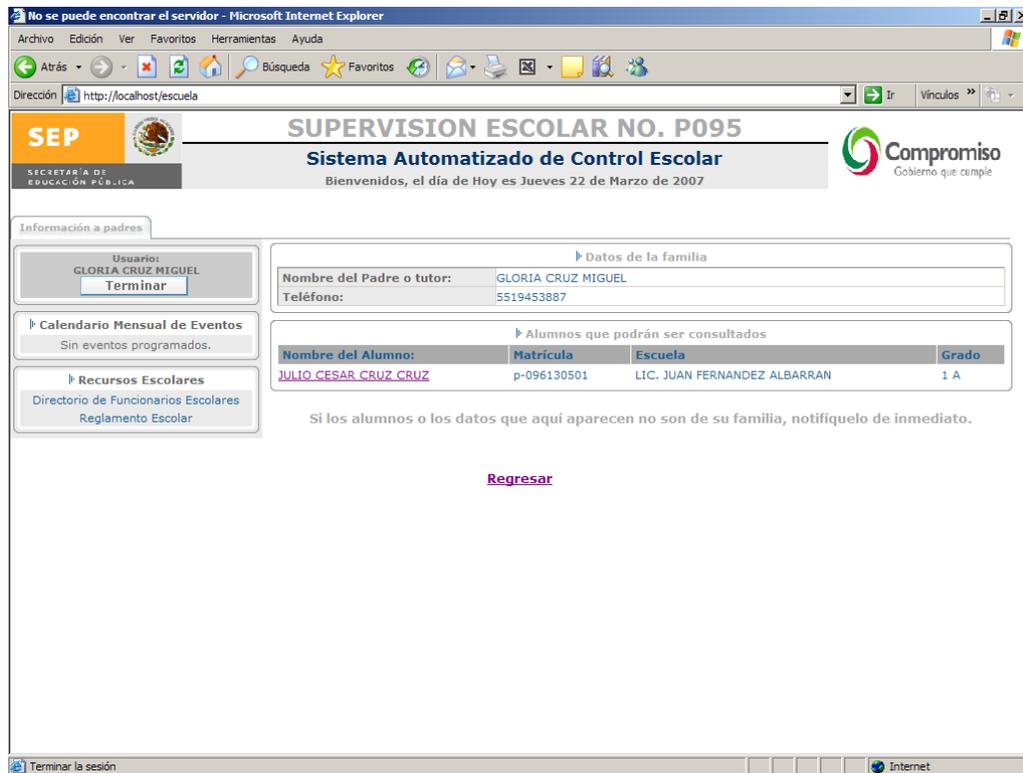


Figura 4.15.- Página de inicio del modulo de acceso a información de alumnos



The screenshot shows a web browser window with the following content:

- Browser Title:** No se puede encontrar el servidor - Microsoft Internet Explorer
- Address Bar:** http://localhost/escuela
- Page Header:** SEP SECRETARÍA DE EDUCACIÓN PÚBLICA, SUPERVISION ESCOLAR NO. P095, Sistema Automatizado de Control Escolar, Bienvenidos, el día de Hoy es Jueves 22 de Marzo de 2007, and Compromiso Gobierno que cumple.
- Navigation:** Alumnos, Calificaciones, Expediente Escolar, Ver Documentación
- User Information:** Usuario: JULIO CESAR CRUZ CRUZ, p-096130501, with a Terminar button.
- Calendar:** Calendario Mensual de Eventos, Sin eventos programados.
- Archives:** Archivos disponibles para el alumno, documentos.pdf
- Student Identification Table:**

Identificación del Alumno	
Nombre Completo:	JULIO CESAR CRUZ CRUZ
Año:	Grado
Inscrito en el grupo:	Grupo
Matricula:	p-096130501
C.U.R.P.:	CUCJ800915327
Status Actual:	Inscrito
Familiar de Contacto :	GLORIA CRUZ MIGUEL
- Buttons:** Regresar
- Footer:** Terminar la sesión, Internet

Figura 4.16.- Página con la información del alumno

Conclusiones

El uso de aplicaciones automatizadas optimiza los procesos repetitivos y computables y a la vez ayuda a eficientar el uso de recursos humanos, materiales y económicos.

El análisis y el desarrollo del sistema a través de la metodología de Proceso Unificado ofreció las siguientes ventajas: permitió integrar todas las fases de desarrollo de la aplicación pues provee un marco de proceso adaptable a las necesidades y características de cada proyecto, adicionalmente que proveyó de una especificación entendible en el proceso para los participantes tanto técnicos como no técnicos. Por otra parte, el uso del Proceso Unificado, permitió la integración de todas las etapas del proceso de desarrollo en una metodología, con el plus de que se establecen claramente las fases.

La utilización de UML me permitió constatar la gran capacidad de estas herramientas para el desarrollo de aplicaciones, entre las capacidades que puedo destacar son que: UML me permitió identificar las fortalezas y debilidades del proyecto, además de que al hacer modificaciones, el uso de UML hace más fácil el proceso y que una de las causas por las cuales el software es poco utilizado o no utilizado es que los analistas y desarrolladores no toman en cuenta a los usuarios finales.

Pude concluir que es conveniente invertir una mayor cantidad de recursos en la realización de un buen análisis y de un buen diseño, esto nos traerá como beneficio que posteriormente no se tendrá que invertir en mantenimientos costosos o no obtener resultados por falta de inversión en las etapas iniciales.

El proyecto me dio a conocer que el desarrollo de sistemas automatizados empleando aplicaciones Web incrementa los alcances pues no se necesita una infraestructura rígida para su funcionamiento y son de alta disponibilidad en tiempo real.

El desarrollo de cualquier aplicación Web se involucra la utilización de la arquitectura en capas, la cual permite definir correctamente las necesidades. Además de que esta arquitectura permite generar una plataforma en la cual se puede continuar con la implementación de nuevas funcionalidades.

Puedo afirmar que el análisis y diseño de la aplicación sugerida será vigente durante un periodo largo de tiempo, el suficiente para que pueda ser tomado por las autoridades educativas para su desarrollo y sea puesto en marcha en uno o dos ciclos escolares próximos a partir del ciclo escolar presente 2006-2007.

Por otra parte, corrobore que el uso de software libre es óptimo para el desarrollo de aplicaciones funcionales y ofrece resultados similares y en algunos casos mejores que el software comercial

En el presente caso fue más adecuado un desarrollo propio que la compra de una aplicación comercial

El Sistema Automatizado de Control para una Supervisión Escolar es una aplicación Web que permitirá realizar el proceso de captura, modificación y consulta y exportación de los datos de los alumnos y docentes de la supervisión escolar P-095 de Naucalpan de Juárez, la principal característica de este sistema es su funcionalidad requerida para los procesos, ya que estos se pueden realizar sin necesidad de que los interesados tengan que trasladarse hacia otras instituciones.

El objetivo de tener una aplicación altamente disponible y amigable hacia los usuarios se cumplirá teniendo esta aplicación en línea, accesibles mediante Internet, ya que es posible tener la información de manera inmediata y con la certeza de que es la información mas reciente, además de que los horarios para algunos servicios prestados por las instituciones educativas se pueden aumentar considerablemente sin que esto represente un gasto para las instituciones ni para los solicitantes.

Glosario

Aprovechamiento Escolar: Forma en que repercute el aprendizaje en el alumno en cuanto a un beneficio en las diferentes áreas como actitudes, habilidades y conocimientos; que le posibiliten relaciones de mayor calidad con la sociedad, con el entorno y consigo mismo.

Asignatura: Unidad básica de un plan de estudios, consistente en un conjunto de temas de una disciplina o de un área de especialización. Suele corresponderle un valor determinado en créditos.

Boleta de calificaciones: Papeleta que se da a un alumno donde consta la calificación que obtuvo en un curso o asignatura.

Categoría de pago: Las categorías del escalafón.

Concatenación: une varios elementos en uno sólo, en este caso, se unen las bases de datos de las escuelas en una para ser entregada a la Coordinación Regional de la SEP.

Coefficiente de cobertura: Porcentaje de alumnos en un ciclo educativo o en todo el sistema, calculado respecto al número de personas en edad de estudiar dicho ciclo. Se aplica a todo el país o a estados, regiones o ciudades.

Clave ISSEMYM: Clave de 9 dígitos asignada por el Instituto de Seguridad Social del Estado de México y Municipios (ISSEMYM) al servidor publico docente.

Deserción: Se define como el abandono que hace el alumno de los cursos o carrera a los que se ha inscrito, dejando de asistir a las clases y de cumplir las obligaciones fijadas, lo cual afecta la eficiencia terminal del conjunto. Es un indicador, ya que si se toma en cuenta el total de las deserciones de los alumnos, permite apreciar el comportamiento del flujo escolar en una generación.

Educación Básica: Comprende los niveles escolares de preescolar, primaria y secundaria

Eficiencia Terminal: Relación cuantitativa entre los alumnos que ingresan y los que egresan de una determinada cohorte. Se determina al dividir el total de alumnos que se inscribieron al primer grado y los que concluyen los estudios en el periodo establecido por el plan de estudios.

Estadística educativa oficial: ofrece información sobre la actividad educativa de los Centros de Trabajo de la educación Básica, sus recursos humanos, las características del alumnado, los resultados académicos y datos de Servicios Complementarios.

Estimaciones: Atribuir a algo un valor o calcular éste aproximadamente, en este caso se refiere a las estimaciones de eficiencia Terminal, deserción, reprobación que se tienen que reportar en el formato 911

Grado: Cada uno de los periodos anuales de los que consta un plan de estudios

Grupo: Conjunto de alumnos que recibe, en una misma aula, con igual horario y con los mismos profesores, las materias o cursos establecidos en un plan de estudios correspondiente a un ciclo escolar.

Generación: Grupo de personas que inician sus estudios en un programa educativo al mismo tiempo, es decir, en el mismo periodo escolar.

Numero escalafonario: Numero en el escalafón del Sindicato de Maestros al Servicio del Estado de México (SMSEM) asignado a cada docente, este numero cambia en relación al cambio de categoría, a las jubilaciones y los decesos.

Personal docente: Personal académico que se encarga del funcionamiento y mantenimiento de un centro educativo. Comprende a los directivos, bibliotecarios, laboratorios y docentes que imparten clases al alumnado.

Personal docente frente a grupo: Personas que orientan la enseñanza. Guían y estimulan los aprendizajes necesarios para la formación de los alumnos.

Personal docente de clases especiales: Es el personal académico contratado para impartir determinadas asignaturas o materias lo que implica la dedicación de un cierto número de horas/semana

Promotores: En el Estado de México, La Secretaría de Educación, Cultura y Bienestar Social ofrece y coordina los servicios de promotoría en educación artística, docencia de arte, promotoría en educación física y promotoría en educación para la salud, principalmente en escuelas ubicadas en cabeceras municipales y zonas rurales. La promotoría es un apoyo fundamental para el docente de preescolar y primaria en el desarrollo de los contenidos programáticos del área, trabajando con los alumnos y en algunas acciones con los padres de familia. Asimismo, los promotores coordinan y desarrollan actividades y eventos artísticos, deportivos y de fomento para el cuidado de la salud.

Reprobación: Es una decisión que toma un profesor o un jurado con respecto al trabajo escolar de un alumno, curso o examen, y por la cual no se le conceden los créditos correspondientes debiendo, en consecuencia, repetir el curso o el examen. Se presenta cuando la población escolar sujeta a un programa curricular no cumple los requisitos académicos exigidos en el plan de estudio y por tanto no están en condiciones escolares de ser promovidos al grado inmediato superior.

Retención: Indicador que expresa el número de alumnos que permanecen dentro del sector educativo, en un grado o nivel, durante todo un ciclo escolar y que continúan en el ciclo escolar siguiente.

Sistema Educativo Nacional (SEN): En sentido amplio, engloba al conjunto de los dispositivos de formación inicial y continua, así como los de educación formal, informal y no formal. En sentido restringido, se refiere al conjunto de actores, estructuras y dispositivos de formación inicial y continúa de la Educación Nacional y del sector privado concertado[7]

El propósito del SEN es ofrecer servicios de educación escolarizada, no escolarizada o mixta a toda la población según lo establezca la Constitución Política. La estructura del sistema educativo se refiere a diversos tipos, niveles y modalidades. Así. La Ley General distingue la educación básica, la media superior, la superior y la de adultos.

El SEN esta formado por[8]:

- Educandas, educandos, educadores y educadoras
- Autoridades educativas
- Planes, programas, métodos y materiales
- Instituciones educativas y organismos descentralizados
- Instituciones particulares con autorización o reconocimiento de validez oficial
- Instituciones de educación superior autónomas

Supervisión Escolar: tiene como propósito prioritario observar el cumplimiento de la reglamentación administrativa oficial y verificar el correcto desempeño de las labores docentes, integración y control de los recursos humanos, materiales y financieros necesarios para la prestación del servicio educativo.

Citas y Referencias

- [1] COHEN Karen, Daniel. Sistemas de información para los negocios: Un enfoque de toma de decisiones. 3ª edición. México: McGraw-Hill, 2000.
- [2] BURCH, John G & STRATER, Felix R. Sistemas de información: Teoría y práctica. México: Limusa, 1981.
- [3] CONALLEN, James. *Modeling Web Application Architectures with UML* [en línea]. Communications of the ACM, USA. Vol. 42, No. 10, October 1999 [Consulta: 10 enero 2006]. Disponible en: <http://www.uml.org.cn/UMLApplication/pdf/webapps.pdf>.
- [4] ELMASRI, R. y NAVATHE, S. B. Fundamentos de Sistemas de Bases de Datos, 3ª edición. Addison Wesley Iberoamericana. 2002.
- [5] FREE Software Foundation. *GNU Licenses* [en línea], actualización 6 febrero 2006 [Consulta: 7 Septiembre 2006]. Disponible en <http://www.gnu.org/licenses/licenses.html>.
- [6] YUSEF Hassan & Francisco J. Martín Fernández & Ghzala Iazza; *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información*, [en línea]; "Hipertext.net", núm. 2, 2004 [Consulta: 7 febrero 2007]. Disponible en: <http://www.hipertext.net>.
- [7] ALAIN, Michel; La conducción de un sistema complejo: la educación nacional; Revista Iberoamericana de Educación, No. 10, enero a abril 1996, OEI, Evaluación de la Calidad de la Educación.
- [8] SEP; Perfil de la educación en México; SEP, 3ª. Edición, México, 2000.

Bibliografía

- BOOCH, Grady, RUMBAUGH, James. y JACOBSON, Ivar. El Lenguaje Unificado de Modelado. España: Addison Wesley, 2000.
- BOOCH, Grady, RUMBAUGH, James. y JACOBSON, Ivar. El Proceso Unificado de Desarrollo de Software. España: Addison Wesley, 2000.
- ECMA. *ECMAScript Language Specification. Standard ECMA-262* [en línea]. 3ª edición Diciembre 1999 [consulta: 2 febrero 2006]. Disponible en: <http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf>.

FERRÉ Grau, Xavier y SÁNCHEZ Segura, María Isabel. *Desarrollo Orientado a Objetos con UML* [en línea]. Facultad de Informática – Universidad Politécnica de Madrid. [Consulta: 3 marzo 2006]. Disponible en: www.itox.mx/Comunidad/Librero/umlTotal.pdf.

FIELDING, R., GETTYS J., et all. *Hypertext Transfer Protocol –HTTP/1.1* [en línea]. Junio 1999 [consulta: 12 enero 2006]. Disponible en: <http://www.ietf.org/rfc/rfc2616.txt>

FOWLER, Martin y SCOTT, Kendall. *UML Gota a Gota*. Addison-Wesley – Pearson, 1999

JCC Consulting, Inc. *JCC's SQL Std. Page* [en línea]. 26 Octubre 2006 [consulta: 2 febrero 2006]. Disponible en: <http://www.jcc.com/sql.htm>.

KRISTOL David M. y MONTULLI Lou. *HTTP State Management Mechanism* [en línea]. Febrero 1997 [consulta: 2 enero 2006]. Disponible en: <http://www.ietf.org/rfc/rfc2109.txt>.

KROLL, Per, KRUCHTEN, Philippe. *Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison Wesley, 8 abril 2003

LARMAN, Craig. *UML y Patrones: Introducción al análisis y diseño orientado a objetos*. 1ª edición. México: Prentice Hall, 1999.

LÓPEZ Santoyo, Saúl, JAQUEZ Leal, Víctor Manuel. *Desarrollo de Aplicaciones Web Utilizando Software Libre* [en línea]. 16 marzo 2001. [Consulta: 5 diciembre 2005]. Disponible en: <http://red.coral.com.mx/ceyusa/tesis/titulacion.pdf>.

NCSA HTTPd Development Team. *Common Gateway Interface* [en línea]. 21 enero 1998 [consulta: 1 Marzo 2001]. Disponible en: <http://hoohoo.ncsa.uiuc.edu/cgi/intro.html>.

MATEU, Carles. *Software Libre. Desarrollo de Aplicaciones Web* [en línea]. Fundació per a la Universitat Oberta de Catalunya .1ª Edición marzo 2004. [Consulta: 12 febrero 2006]. Disponible en: http://www.uoc.edu/posgrado/matricula_abierta/img/873.pdf.

SADOSKI, Darleen. *Three Tier Software Architecture* [en línea]. 16 febrero 2000 [consulta: 7 enero 2006]. Disponible en: <http://www.sei.cmu.edu/str/descriptions/threetier.html>.

SADOSKI, Darleen. *Client/Server Software Architectures -An Overview* [en línea]. 2 Agosto 1997 [consulta: 7 enero 2006]. Disponible en: <http://www.sei.cmu.edu/str/descriptions/clientserver.html>

SILVA, Darío Andrés y MERCERAT, Bárbara. *Construyendo aplicaciones Web con una metodología de diseño orientada a objetos* [en línea]. 28 febrero 2002 [Consulta: 12 marzo 2006]. Disponible en: http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_r.pdf.

RAGGETT, Dave, LE HORS, Arnaud y JACOBS, Ian, traducción Juan R. Pozo. *HTML 4.01 Specification* [en línea]. 24 diciembre 1999 [consulta: 30 diciembre 2005]. Disponible en: <http://html.conclase.net/w3c/html401-es/cover.html>.

W3C working group. *XHTML 1.0 The Extensible HyperText Markup Language (second edition)* [en línea]. 26 Enero 2000, revisión agosto 2002 [consulta: 30 diciembre 2005]. Disponible en: <http://www.w3.org/TR/xhtml1/>.

W3C working group, Traducción Carlos Benavides. *Cascading Style Sheets, level 2 revision 1CSS2 Specification* [en línea]. 25 febrero 2004 [consulta: 2 enero 2006]. Disponible en: http://www.w3.org/Style/css21-updates/css2.1_spa.pdf.

YERGEAU, Francois. *Extensible Markup Language (XML) 1.0 (Third Edition)* [en línea]. 4 febrero 2004 [consulta: 30 diciembre 2005]. Disponible en: <http://www.w3.org/TR/2004/REC-xml-20040204/>.