

81



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería

PUBLICACION MASIVA DE RECURSOS DE APOYO A LA EDUCACION EN INTERNET CON HERRAMIENTAS AMPLIAMENTE DISPONIBLES

PUMASI

TESIS:

Que Para obtener el Titulo de
INGENIERO EN COMPUTACION

Presenta:

DANIEL SOL LLAVEN

DIRECTOR

Ing. Alberto Templos Carbajal



México, D.F.,

278700

2000



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

Agradecimientos

A mi Madre, cuyo esfuerzo dio origen a este proyecto:

Hermila Llaven Nucamendi

A los sinodales, que han enriquecido este trabajo con su visión y dirección:

Ing. Alberto Templos Carbajal
Dra. Ana María Vázquez Vargas
Mtra. Laura Sandoval Montaño
Ing. Juan José Carreón Granados
Ing. Orlando Zaldivar Zamorategui

A la Dirección General de Servicios de Cómputo Académico por su apoyo al proyecto:

Director General: Dr. Alejandro Pisanty Baruch
Director de Cómputo para la Investigación: Dr. Rafael Fernández Flores
Coordinador de Prospección: Ing. Antonio Enrique González Velázquez

A mis colegas en la DGSCA:

Ing. Eva Edith López López
Lic. Itzel Hernández Serra
Manuel Alejandro Alfaro Flores
Luis Alejandro Castillo Vilchis

A quienes desarrollaron el software de uso gratuito, esencial para este proyecto:

Larry Wall y el grupo de usuarios de Perl
Apache Group y la NCSA en la Universidad de Illinois en Urbana Champaign
Hughes Technologies

CAPÍTULO I INTRODUCCIÓN	3
CAPÍTULO II ANTECEDENTES	5
A. EDUCACIÓN E INTERNET	5
B. DESARROLLO HISTÓRICO	6
1. Internet.....	6
2. REDUNAM	9
CAPÍTULO III DEFINICIÓN DEL PROBLEMA	13
A. OBJETIVO.....	13
B. HIPÓTESIS DE SOLUCIÓN.....	13
CAPÍTULO IV MÉTODO	15
A. ANÁLISIS	15
1. Selección de Herramientas	15
2. Arquitectura	17
3. Selección de elementos constitutivos	19
a) Sistema Gestor de Base de Datos (DBMS).....	19
b) Servidor Web.....	21
c) Cliente Web.....	22
d) Lenguaje de Programación	23
4. Arquitectura de Cliente Abierto.....	25
B. CONSIDERACIONES PARA EL DISEÑO.....	27
C. DISEÑO.....	29
1. Base de Datos	30
a) Diagrama Entidad Relación	32
b) Entidades	33
c) Relaciones.....	35
d) Atributos.....	39
2. Interfaz (CGI)	47
a) Diagramas Unified Modeling Language (UML).....	47
◆ Entrada al Sistema.....	48
◆ Creación de Materiales.....	54
◆ Administración de Cursos.....	60
b) Diagramas de Relación de CGI.....	69
◆ Explicación de símbolos	71
◆ Modelo del Sistema.....	72
3. Apariencia del Sistema	94
CAPÍTULO V ANÁLISIS DE RESULTADOS.....	109
A. LIMITACIONES	109

1. <i>Limitado rango de usuarios</i>	107
2. <i>Búsquedas</i>	107
3. <i>Materiales transitorios</i>	107
4. <i>Limitaciones de distribución</i>	107
5. <i>Transplante a plataformas diferentes</i>	107
6. <i>Uso limitado de las capacidades del DBMS</i>	107
B. POTENCIAL	107
1. <i>Mantenimiento</i>	107
2. <i>Mantenimiento a la Base de Datos</i>	107
3. <i>Desarrollos adicionales</i>	107
a) <i>Recursos de Audio y Video</i>	107
b) <i>Modificación del catálogo de materiales</i>	107
c) <i>Integración con ambientes de colaboración</i>	107
d) <i>Módulos para edición de Fórmulas</i>	107
e) <i>Programas de instalación</i>	107
4. <i>Compatibilidad con otros sistemas</i>	107
C. CONCLUSIONES	107
1. <i>Tendencia de la Industria</i>	107
BIBLIOGRAFÍA	107

Capítulo I **Introducción**

Capítulo I **Introducción**

En el presente trabajo se revisa el desarrollo de un sistema que tiene por propósito el facilitar a profesores y alumnos el utilizar la red como un medio para intercambiar material de apoyo a sus tareas académicas.

Este sistema se encarga de controlar a los usuarios y su acceso a los materiales y brinda un mecanismo sencillo para que los profesores o alumnos autorizados generen nuevo material para la comunidad.

La tesis comienza revisando la evolución y avance del cómputo y la Red UNAM en el capítulo II, con lo que se tiene una mejor perspectiva de la situación en la que se utiliza el sistema.

En el capítulo III se define un objetivo específico para el trabajo de desarrollo y se indica la construcción de un sistema como el medio para ayudar a la solución del problema.

En el capítulo IV se resume el proceso de desarrollo del sistema en sus diversas etapas. Iniciando con el análisis y la selección de las herramientas y arquitectura a usar, en la sección A. Revisando las condiciones especiales que estas imponen sobre el desarrollo en la sección B. Y detallando el diseño que se realizó del sistema en la sección C.

El capítulo V es una reflexión sobre lo que se logró con el desarrollo, el trabajo que resta por hacer y las consecuencias que tienen las diversas características del sistema en el marco de un posible uso institucional.

El desarrollo del sistema se realizó en la Coordinación de Prospección e Innovación de la Dirección General de Servicios de Cómputo Académico entre 1997 y 1999. Con el apoyo del Ingeniero Antonio E. González Velázquez (Coordinador), del Dr. Rafael Fernández (Director de Cómputo para la Investigación) y del Doctor Alejandro Pizanti Baruch (Director General de Servicios de Cómputo Académico) quienes con su continuo interés y dirección determinaron el rumbo y utilidad del presente trabajo.

Capítulo II **Antecedentes**

Capítulo II **Antecedentes**

A. Educación e Internet

Uno de tantos usos que ha recibido la Internet desde sus orígenes es la de funcionar como apoyo a la labor educativa, principalmente en instituciones de educación superior y de investigación, fungiendo como un medio de comunicación práctico y eficaz para investigadores alrededor del mundo.

Algunos de los esfuerzos más importantes de la comunidad académica en Internet, aparte de la difusión de incontables de proyectos, es la creación de varias organizaciones dedicadas al desarrollo de la educación que emplean la Internet como punto de contacto de sus miembros y medio de difusión de los resultados obtenidos.

Podemos encontrar a la UNESCO¹ (United Nations Educational, Scientific and Cultural Organization), SEP² (Secretaría de Educación Pública), EDUCOM³ (Consortio de Universidades y colegios de los E.U. dedicados a transformar la educación mediante el uso de la tecnología de comunicaciones) y muchas otras publicando sus esfuerzos y actividades a través de Internet y fomentando que los diversos órganos que las componen hagan lo mismo.

Además de estos, la iniciativa privada ha puesto especial atención en los servicios de red como una herramienta que agilice operaciones comerciales, el flujo interno de información y para capacitar personal mediante presentaciones multimedia y centros de capacitación que pueden emplearse por personas en diversas sedes de la compañía empleando la infraestructura de red. Ahorrando de este modo, costos asociados a la manutención de personal dedicado a capacitación en todas las sedes de la compañía, al establecer puntos centrales de referencia y ayuda. Además, se pueden construir herramientas de capacitación que brindan información oportuna y fidedigna sobre el desempeño y capacidades de los empleados, lo que permite un mejor control del personal.

¹<http://www.education.unesco.org/index.html>

² <http://www.sep.gov.mx>

³ <http://www.educom.edu/>

B. Desarrollo Histórico

1. Internet

Se he escrito mucho sobre el surgimiento y desarrollo de Internet, y mucho más ha quedado sin registrar. A diferencia de las tecnologías desarrolladas exclusivamente por una compañía, consorcio o por encargo de alguna entidad gubernamental, la Internet se ha nutrido del esfuerzo de muchos especialistas de varias especialidades y es regulado por muchas instituciones simultáneamente. Quizá la mayor parte del desarrollo de lo que hoy constituye Internet fue realizado por universidades y personas en instituciones que no pretendían lucrar con ella, lo que ha sido determinante en su conformación y desarrollo.

El término Internet está definido oficialmente desde el 24 de Octubre de 1995 por la FNC (Federal Networking Council)⁴ donde se especifica lo siguiente:

Resolución: El Consejo Federal de Redes (FNC) se manifiesta de acuerdo en que el lenguaje siguiente refleja nuestra definición del término "Internet". "Internet" se refiere al sistema global de información que -- (i) está lógicamente interconectado en una unidad por un espacio único de direcciones basado en el Protocolo de Internet (IP) o sus subsecuentes ampliaciones/versiones; (ii) es capaz de adoptar comunicaciones usando el conjunto de aplicaciones del Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP) o sus subsecuentes ampliaciones/versiones, y/o otros protocolos compatibles con IP; y (iii) provee, usa o hace accesible, ya sea de modo público o privado, servicios de alto nivel que se asientan en las comunicaciones e infraestructura relacionada descrita en la presente.

Sin embargo, la Internet tuvo sus orígenes mucho antes, en la Red conocida como ARPANET.⁵

En 1964 la RAND (Contracción de "Research and Development"), una organización no lucrativa dedicada a mejorar las políticas y toma de decisiones gubernamentales mediante investigaciones y análisis, inventó un nuevo tipo de red orientada hacia la solidez y tolerancia a fallas en el equipo. La propuesta escrita por Paul Baran se basó en la idea de que la nueva estructura de red no tendría ninguna autoridad central. Además, estaba diseñada para operar aún estando desmembrada. Para esto, todos los nodos serían iguales cualitativamente, cada uno podría enviar y recibir mensajes. Todos los mensajes

⁴ La resolución se puede encontrar en: http://www.fnc.gov/Internet_res.html

⁵ Fuente: "The history of Internet and the WWW" por David Mayr
<http://ourworld.compuserve.com/homepages/dmayr/history.htm>

se enviarían en paquetes, cada uno con su propia dirección. Estos paquetes podrían ser enviados a un nodo y llegar a otro. Lo innovador era que la forma en que los paquetes atravesaban la red no era importante. Eso significa que si un nodo era destruido, el resto de los nodos seguirían siendo capaces de comunicarse. Esto es poco eficiente y lento, pero extremadamente confiable. La Internet todavía usa este método hoy día, y solo ha habido una suspensión de servicio colectiva a la fecha. Sin embargo, la estructura actual de Internet ya no está construida con la supervivencia a la destrucción de sus partes sino a una comunicación más eficiente, por lo que es más frágil de lo que su idea original especifica.

La primera red de prueba construida sobre estos principios estuvo en el Laboratorio Nacional de Investigaciones (National Research Laboratory) en la Gran Bretaña en 1968. Poco tiempo después, ARPA, Agencia de Proyectos de Investigación Avanzados (Advanced Research Projects Agency) del Pentágono, quiso instalar una red más avanzada basada en los mismos principios en los Estados Unidos. La red consistió de cuatro computadoras de alto rendimiento. En 1969, se instaló el primer nodo en la UCLA, Universidad de California en Los Angeles (University of California at Los Angeles).

Para 1971 había 23 nodos de ARPANET, que fue el nombre dado a esta red, después del de la UCLA, se establecieron nodos en el Instituto de Investigaciones de Stanford (Stanford Research Institute), en la universidad de Utah y en la UCSB

Universidad de California, Berkeley (University of California Santa Barbara).

ARPANET se construyó porque el tiempo de procesamiento tenía un costo muy alto muy elevado y ARPANET ofreció a los investigadores la posibilidad de emplear a las computadoras de otras instituciones y compartir las propias a distancia.

En Marzo de 1972 Ray Thomlinson de la BBN (Bolt Beranek and Newman, Compañía de cómputo) inventó el primer programa de correo electrónico. Hoy día uno de los principales usos de la Internet es el intercambio de mensajes, y la forma de estos no ha cambiado mucho desde estos primeros intentos.

Los primeros nodos internacionales se establecen en 1973. Estos estaban localizados en Escocia y en Noruega. El crecimiento de la ARPANET fue posible porque podía ser empleado por casi cualquier plataforma, gracias a que el código fue puesto a disposición gratuitamente y se incluía con el sistema operativo UNIX.

En 1974 Vint Van Cerf y Bob Kahn publican "A protocol for Packet Network Internetworking" (Un protocolo para Interconexión de Redes de Paquetes) en el que se especifica el diseño de un TCP (Protocolo de Transferencia de Paquetes).

En 1979 se establece la Usenet, basado en, UUCP Unix to Unix Copy (Copia de Unix a Unix) con fines de intercambio académico de información.

El protocolo TCP/IP se establece para ARPANET en 1982. Este protocolo se estandarizaría el 1 de enero de 1983 (reemplazando al NCP o Network Control Protocol, Protocolo de Control de Red). La palabra "Internet" comienza a utilizarse. Esta transición fue una de las más grandes pruebas a la capacidad de auto organización de la comunidad de Internet, y se llevó a cabo sin que se reportaran incidentes de consideración.

También en 1983 ARPANET se divide en ARPANET y el segmento militar MILNET. MILNET se integra con la red de datos de la defensa (Defense Data Network) creada el año anterior. El nuevo protocolo estándar creado el año anterior y esta división son factores importantes en la demilitarización de la ARPANET.

Este mismo año se desarrolla el servidor de nombres en la universidad de Winsconcin, el que se usa todavía y permite identificar a los nodos de la red por un nombre en vez de una secuencia de números.

En 1984 se rebasan los 1000 nodos de ARPANET y se adopta el servicio de nombres con el servicio llamado DNS Domain Name System (Sistema de Nombres de Dominio).

La Fundación Nacional de Ciencia, NSF (National Science Foundation) decide ligar cinco de sus centros de supercómputo (cómputo de alto rendimiento) para que pudiesen ser empleados en proyectos de investigación, pero la burocracia y falta de personal de ARPANET impide que se integren a esta red. De este modo se construye una nueva red basada en el protocolo IP de ARPANET, la NSF enlaza los cinco centros con líneas de 56 Kbps (56x1024 bits por segundo) pero no establecen las ligas con las universidades por falta de recursos económicos para tender el cableado necesario. Las escuelas e instituciones se comunican en grupos regionales y comunican estos grupos con alguna de las supercomputadoras. La carga de información por estas líneas se incrementó paulatinamente y pronto las líneas fueron insuficientes para manejar la cantidad de datos enviados.

En 1987 la NSF firma un contrato con Merit Networks para incrementar la capacidad de la red. Desde ese entonces las líneas y servidores han sido continuamente mejorados por diversos proveedores. Se rebasan los 10,000 servidores.

Para 1989 se rebasan los 100,000 servidores.

En 1990 ARPANET deja de existir, pero los usuarios difícilmente lo notan por mantenerse los servicios que esta prestaba. De hecho, todavía se conservan.

WAIS (Wide Area Information Service) Servicio de Información de Area Amplia y Gopher (contracción de Go-For y mascota de la universidad de Minnesota donde se desarrollo) se liberan, y pronto se convierten en dos de los servicios principales de Internet.

WWW World Wide Web (Telaraña de amplitud mundial) se libera en 1992 por la CERN "Conseil Europeen pour la Recherche Nucleaire" (Consejo europeo de investigaciones nucleares, llamado actualmente Laboratorio Europeo de Física de Partículas). El número de servidores rebasa el 1,000,000.

Para 1993, el primer programa lector de WWW (browser) llamado Mosaic, se libera. La tasa de crecimiento de Internet se estima en 341% y aunque a un ritmo menor, continúa creciendo.

Con la popularización de la Internet y la nueva presencia comercial en ella, comienzan a hacerse evidentes algunas deficiencias en los protocolos básicos principalmente respecto a ciertos aspectos de seguridad y direccionamiento de servidores, en respuesta a estos problemas, Los Directores del Area de IP de la IETF, (Internet Engineering Task Force) Fuerza de Ingeniería de Internet en su reunión del 25 de julio de 1994⁶. Recomiendan el desarrollo y adopción gradual de un nuevo protocolo, conocido como IPng, (Next Generation Internet Protocol) o IP versión 6, y se lanza su propuesta de estándar ese mismo año. Este evento marca el inicio de la así llamada Internet II (basada en una nueva infraestructura radicalmente más poderosa que la anterior) y la adopción gradual de los nuevos protocolos, en contraste con la adopción del TCP/IP el 1º de enero de 1983.

En 1995 se culmina la privatización de la infraestructura de red a cargo de la NSF planteado desde 1985. Los fondos recabados de la adquisición de derechos por compañías privadas se redistribuyen entre las redes regionales para adquirir enlaces de escala nacional y alto rendimiento con las ahora numerosas compañías que podían proveer ese servicio.

2. REDUNAM⁷

Aún cuando la Internet, y la propia REDUNAM son fenómenos que se han presentado en la presente década, la comunicación electrónica de datos tuvo sus comienzos desde la década de los 60 y 70, con la naciente infraestructura telefónica de la UNAM, la cual se

⁶ RFC 1752

⁷ Fuente: Centro de Información de REDUNAM <http://www.nic.unam.mx/REDUNAM/historia.html>

empleó ampliamente para intercomunicar terminales de diversos tipos por medio de las líneas telefónicas.

En octubre de 1985, la UNAM e IBM de México subscribieron un convenio con el cual se puso en marcha un proyecto conjunto de investigación y desarrollo en el que se contemplaba: La instalación de una red universitaria de cómputo para apoyo a la docencia que permitiera el acceso remoto a los sistemas de procesamiento de datos actuales y futuros de las dependencias de la UNAM y la creación de un laboratorio para el diseño y la manufactura apoyado por computadoras. Dicho convenio fue firmado por el rector de la UNAM, el presidente y director general de IBM y por el secretario de Comercio y Fomento Industrial. A raíz del convenio se integraron dos grupos de especialistas de la UNAM e IBM con objeto de definir detalladamente el plan de trabajo y así cumplir con los objetivos propuestos. El planteamiento original se adaptó a las cambiantes condiciones de las redes de computadoras y se alejó paulatinamente del esquema de redes cerrado de IBM hacia sistemas académicos más abiertos basados en software gratuito y estándares públicos.

En 1987 se establece la primera conexión a la red Académica de C o Bitnet por medio de un enlace telefónico, desde la Ciudad Universitaria hasta el Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM) y de ahí hasta San Antonio, Texas en los EUA.

Bitnet es la abreviatura de "Because it's Time" y fue una de las primeras redes de área amplia del mundo empleada primordialmente por universidades con fines académicos.

En un esfuerzo por consolidar su presencia en Bitnet, se utilizó la recién adquirida computadora IBM 4381 como residencia del correo electrónico y demás servicios de Bitnet. Y mediante terminales IBM con emulación 3270 por un enlace con la red TELEPAC de la SCT se logra dar una cobertura nacional, aunque limitada a este servicio.

No fue sino hasta 1989 cuando la UNAM a través del instituto de Astronomía establece un convenio de enlace con la red NSF de los EUA, el que empleaba al satélite Morelos II para enlazar al instituto de Astronomía de la UNAM con el UCAR-NCAR con residencia en Boulder Colorado y por primera vez en la universidad se comunica usando fibra óptica a las redes locales del Instituto de Astronomía y la Dirección General de Servicios de Cómputo Académico. Siendo inaugurada oficialmente en ese mismo año la REDUNAM por el entonces rector José Sarukhán Kermez.

En este período de tiempo, la UNAM comenzó una autentica revolución en cuanto al manejo de sus recursos de cómputo y comunicaciones que todavía sigue transformando la forma en que los universitarios nos comunicamos entre nosotros. Se inició la

adquisición masiva de computadoras personales, la construcción de grandes centros de cómputo de alto rendimiento y la ampliación y fortalecimiento de la red de comunicaciones de la universidad que estableció el enlace central de fibra óptica y la telefonía digital con que hoy contamos.

En 1990 la UNAM fue la primera institución en Latinoamérica que se incorpora a la red mundial Internet, la que constituya uno de los fenómenos sociales más importantes del siglo originados en avances tecnológicos.

Los servicios de Telnet, FTP (File Transfer Protocol), correo electrónico y listas de correo en Internet comenzaron a operar en 1991⁸. En 1992 se establecen los primeros servidores de Gopher en la universidad, lo que se complementa en 1993 con la creación de servicios electrónicos de Bases de Datos y revistas electrónicas, servicios hemerográficos y publicación de artículos íntegros en 1994. Servicio de traducción automática español-inglés, establecimiento de la red de videoconferencias e inicio de los servidores World Wide Web universitarios en 1995. Bases de datos de imágenes, servicios específicos Web y de bases de datos para las dependencias que los solicitan y almacenamiento masivo de información en 1996. Transmisión de Radio UNAM por Internet, Colaboración como sitio de consulta de los resultados preliminares en las elecciones del 6 de Junio, integración de la red telefónica digital con Red UNAM, Comienzo de la operación de los laboratorios "Fundación UNAM" con una fuerte orientación hacia Internet y se comienza la migración de la red central de fibra óptica de tecnología FDDI (Fiber Distributed Data Interface) a ATM (Asynchronous Transfer Mode).

Actualmente⁹ la infraestructura de REDUNAM transmite indistintamente voz y datos, mediante sistemas digitales basados en las más modernas normas internacionales. Las principales instalaciones de la Universidad están integradas a la Red. Esto significa que a nivel licenciatura, posgrado e investigación, alrededor del 90% de sus miembros se encuentran en instalaciones cubiertas por la Red, independientemente de su ubicación geográfica. El sistema es descentralizado redundante y esta integrado por 31 Nodos de Cómputo y Telecomunicaciones enlazados entre sí por vía fibra óptica. Así mismo tiene una infraestructura instalada para más de 170 redes locales de cómputo. La Red enlaza a cerca de 8000 computadoras en la UNAM entre sí y alrededor de un millón de computadoras en el resto del mundo.

Actualmente la REDUNAM brinda conexiones a 33 dependencias universitarias fuera de la Ciudad Universitaria y a 71 organismos privadas y de gobierno a lo largo y ancho del territorio nacional.

⁸Fuente: Departamento de Operación de la Red

Capítulo III **Definición del
problema**

Capítulo III Definición del Problema

Dada la complejidad de los elementos que intervienen en la vida diaria en la actualidad se requiere de un proceso educativo formal cada vez más significativo en la vida de todas las personas.

En las actividades académicas se requiere de todos los apoyos que se encuentren disponibles para que los profesores puedan conformar una estrategia educativa eficaz, sin embargo, la Internet como un medio de comunicación y de distribución de materiales educativos se ha mantenido casi sin usar. Esto se debe principalmente a los altos costos asociados a la adquisición de equipo, software y salarios del personal especializado necesarios.

La capacitación del personal docente debe ser además una actividad que requiera poco tiempo del profesor y que le permita ver resultados rápidamente de modo que resulte atractiva y no le distraiga de sus tareas primordiales.

A. Objetivo

Crear un mecanismo que permita a los profesores publicar material en red con mínimos requerimientos de software, hardware y de personal especializado de apoyo que sea adecuado al ámbito universitario.

B. Hipótesis de solución

Si en una institución se cuentan con recursos de cómputo para al menos una parte de su población escolar y algunos profesores consideran que se puede emplear la distribución de materiales educativos por Internet como un apoyo a su tarea, solo será necesaria la construcción de un esquema de distribución de los materiales económica en el tiempo de personal capacitado y de profesores, en el software que utiliza y que pueda funcionar en la infraestructura existente para que sea posible comenzar su utilización en apoyo a la educación.

Para esto se plantea construir un sistema que permita la distribución de materiales elaborados por el profesor necesitando un mínimo de apoyo por personal altamente especializado en tecnologías de Internet.

Capítulo IV **Método**

Capítulo IV **Método**

A. Análisis

Reconocemos tres elementos primordiales: Las personas involucradas en el proceso enseñanza aprendizaje, la infraestructura de cómputo y telecomunicaciones y el sistema para elaborar el material educativo y distribuirlo por red.

De estos elementos el presente trabajo solo se encargará de revisar el último, la construcción de un sistema que facilite la elaboración del material a los profesores y brinde un mecanismo de acceso simple pero controlado para los alumnos.

El sistema deberá ser capaz de atender comunidades de al menos 50 usuarios y comúnmente de algunos cientos. Siendo usado por red, lo que implica que deberá estar disponible las 24 horas del día. Es por esto que se consideró conveniente el emplear un servidor UNIX para albergar el sistema con capacidad suficiente para tender varias consultas por segundo.

La infraestructura de red y de terminales está siendo mejorada por el programa "Fundación UNAM" y por un esfuerzo sostenido de las diversas escuelas, facultades e institutos. De modo que gratuitamente o mediante una cuota los estudiantes tienen acceso a recursos de cómputo y de red que les permitan emplear este tipo de apoyos a su educación.

En cuanto a los profesores y alumnos que empleen este sistema, se realiza una continua actividad de promoción de las herramientas disponibles como parte de las tareas de la Dirección General de Servicios de Cómputo Académico de modo que sean utilizadas por aquellos profesores que manifiesten su interés en incluir estas alternativas a sus esquemas de enseñanza.

1. Selección de Herramientas

Enfocaremos el trabajo a realizar en generar una alternativa sencilla de utilizar, que aproveche los recursos que la Universidad está construyendo en materia de cómputo y telecomunicaciones como ayuda a la comunicación e intercambio de materiales académicos entre las personas que tengan intereses similares en la universidad y eventualmente hacia la comunidad universitaria en su conjunto.

El aspecto que presenta más ventajas para los procesos de comunicación es el de las redes de computadoras empleadas en servicios comunes de red, los que ya están siendo utilizados en algunos ámbitos académicos, en especial con fines de investigación y comunicación con colegas de otros países.

Las herramientas que parecen sacar un mejor provecho de los recursos de red, son los así llamados “Servicios de Red”; programas que a lo largo de varios años de operación a nivel mundial han probado su eficacia y han corregido casi todos sus problemas originales de seguridad y funcionamiento, Estas herramientas, se encuentran disponibles por varios productores de software, algunos de ellos, las otorgan gratuitamente para fines académicos, por lo que son muy atractivas para su uso en la Universidad.

La publicación de materiales de toda clase para su distribución en Internet ha pasado por varias etapas, las que han desembocado en un predominio de la World Wide Web (WWW o simplemente “web”) entre los servicios de Internet. Debido principalmente a su libertad al relacionar la información y su capacidad para integrar multimedia superior a la de otros mecanismos como la transferencia de archivos o los paquetes de software tradicional. Su principal problema al emplearlo para publicar materiales de apoyo, es que debido principalmente a la libertad de navegación y a la riqueza gráfica que promueve, la elaboración de materiales suele ser tarea de equipos multidisciplinarios que elaboran toda una imagen a cada material que se publica, esfuerzo que sería redundante y excede a la capacidad de los equipos de trabajo que la UNAM tiene dedicados a este fin si se quisiera aplicar a una gran cantidad de cursos.

Lo que se requiere entonces, es el esfuerzo de integración de estas herramientas de modo que la complejidad de su instalación y administración se vea reducida al mínimo cuando se empleen en un esquema conocido de antemano, y que la tarea de elaborar nuevos materiales sea distribuida en un grupo de trabajo de modo que el docente pueda acceder a ella con un mínimo de esfuerzo adicional al de la propia elaboración del material, en lo que el trabajo de diseñadores y técnicos se vea aprovechado al máximo al ser utilizado por un gran número de materiales.

La gran variedad de variantes y de diversas herramientas disponibles para desarrollar este tipo de servicios, hace forzoso el considerar cuidadosamente las diversas opciones, sus fortalezas, debilidades, compañías o comunidades que las respaldan y el propio desarrollo que han tenido al través del tiempo para identificar aquellas que resulte más conveniente usar.

No se puede hablar de encontrar las mejores herramientas disponibles ya que sus características no siempre son las adecuadas para toda aplicación, además, los detalles de implementación son tantos y sus efectos en la aplicación son tan difíciles de predecir, que en realidad no es conveniente prestar atención a todas las características de estos.

No debemos olvidar que en su gran mayoría, las herramientas están diseñadas para poder atender a una gran variedad de aplicaciones, de modo que el presente trabajo no requerirá aplicar todas sus capacidades.

Hay una gran cantidad de comparaciones técnicas de las herramientas que a continuación se mencionan, y hay una gran cantidad de argumentos que se pueden esgrimir, en este trabajo nos limitaremos a revisar las características principales de las herramientas contraponiéndolas en cuadros comparativos, para seleccionar la que se acerque más a las necesidades presentes y para detectar las alternativas que puedan tenerse aún después de optar por alguna de ellas.

2. Arquitectura

Siguiendo la idea de desarrollar una solución basada en servicios de red, notamos que estos pueden funcionar sobre todas las alternativas de redes de computadoras actuales, por lo que no es necesario elegir alguna en particular. Existen, sin embargo, diversas alternativas de implementación de servicios de Red, de entre las cuales tendremos que elegir alguna basados en sus características. Siendo deseable el tener toda la información almacenada en un repositorio central, disponible desde cualquier punto de la red, las alternativas se reducen solo a servicios de Red y aplicaciones cliente servidor.

Opción	Características	Productos disponibles
Aplicaciones cliente servidor	<ul style="list-style-type: none"> • Almacén central de información • Control sobre los clientes • Desarrollo por plataforma • Desarrollo para cada necesidad 	Oracle, Sybase, PostgreSQL y otras Bases de datos para las que se puede desarrollar en lenguajes como C, C++, Delphi, Visual Basic, Java y otros.
Servicio de red, World Wide Web con bases de datos	<ul style="list-style-type: none"> • Almacén central de información • Libertad de clientes a emplear • Gran libertad de plataformas • Gran flexibilidad en el resultado final 	Bases de datos arriba mencionadas, Servidores WWW como Enterprise Server, Internet Information Server, Apache, y otros, una gran variedad de clientes y lenguajes para programación de interfaces como Perl, C, C++, Java, Phyton, TCL, etc.
Servicio de red, listas de correo electrónico	<ul style="list-style-type: none"> • Completa libertad de plataformas • Gran libertad y facilidad para la participación 	Diversas alternativas para cada SO, y servidores de listas como Mayordomo, List Serve, y otros.
Servicios de	<ul style="list-style-type: none"> • Facilidades de administración 	Las alternativas que han tenido

red, Integraciones comerciales	<ul style="list-style-type: none"> • Almacén central de información • Esquema coherente de empleo • Dependencia de plataforma • Alto costo 	más aceptación son: Lotus Notes, Microsoft Network, Netscape, entre otras.
--------------------------------------	--	--

Como puede notarse, cada una de las alternativas plantea algunas ventajas que la hacen muy atractiva. De hecho, no se considera que la solución a la problemática presente puede darse por una sola de estas alternativas, los problemas de costos asociados a las Integraciones comerciales de servicios de red las hacen prohibitivas para organizaciones que prefieren dedicar el presupuesto a ampliar la infraestructura existente en vez de experimentar con nuevos esquemas de uso, afortunadamente, las características más atractivas de estos sistemas son reproducibles en desarrollos de herramientas con las alternativas anteriores.

Las aplicaciones cliente servidor se han empleado con éxito en la Universidad por aproximadamente 7 años en aplicaciones tales como la administración escolar y administración de inventarios, sin embargo, carecen de la flexibilidad que permita que las personas que las emplean cambien gradualmente de un papel de usuarios a uno de desarrolladores como las cambiantes necesidades del académico requieren, la única solución que cuenta con la flexibilidad necesaria para adaptarse a las necesidades de diversos académicos y sus alumnos parece ser la de Servicios World Wide Web (WWW) con Bases de datos. Se tiene la ventaja de que existen varias alternativas de software libre para estas arquitecturas, de modo que el esfuerzo que se dedique en desarrollo está asegurado no solo por los estándares de los servicios de Red, sino por las comunidades de usuarios en todo el mundo.

La solución basada en un servicio WWW apoyado por bases de datos resuelve con éxito la parte de construcción y distribución unidireccional de materiales, pero requiere del apoyo de la libertad de los servicios de listas de correo electrónico para que haya una resaltada capacidad de interacción entre los docentes y sus alumnos. Sin embargo, se dejará de lado en el presente trabajo el desarrollo de una solución que integre el manejo y control del correo electrónico por la complejidad que encierra el ser compatible con las herramientas existentes.

Resulta recomendable que en los ambientes educativos que apliquen el sistema a desarrollar se utilice el correo electrónico. Como un valioso elemento de comunicación entre los participantes del sistema.

3. Selección de elementos constitutivos

Tomando la alternativa de generar una solución basada en un servidor WWW, resta analizar las principales alternativas para los elementos constitutivos del sistema, los que son: El servidor de Base de datos en el que se mantendrá la información. El servidor WWW también conocido como HTTPD (del inglés: Hyper Text Transfer Protocol Daemon) que atenderá las peticiones de los clientes, Los clientes que se encuentran disponibles, las capacidades y requisitos con que cuentan, el servidor de listas de correo y el lenguaje de desarrollo en el servidor por el que se integrarán las diversas partes en un todo coherente.

Cada una de estas tareas cuenta con varias alternativas, a menudo intercambiables, es decir, muchos servidores WWW pueden emplearse indistintamente en el mismo sistema (sin necesidad de modificar el código de enlace con la base de datos) gracias a que se apegan a algún estándar del servicio, de modo que, aún después de realizado el sistema es muy posible que se cuente con una gran libertad en cuanto a los componentes que lo constituyen y por el contrario, aquellas alternativas que no permitan este tipo de compatibilidad deberán ser detectadas y evaluadas con cuidado.

a) Sistema Gestor de Base de Datos (DBMS)

El sistema Gestor de la Base de Datos, también conocido como Database Management System (DBMS) se encarga de almacenar con una redundancia controlada y en una estructura que facilite la recuperación y acceso una gran cantidad de datos. Los sistemas de bases de datos relacionales han tenido un gran desarrollo en la última década y son en la actualidad la herramienta más usada para mantener grandes colecciones de datos en sistemas de cómputo.

Las principales alternativas de Manejador de Base de Datos con que se cuenta actualmente para desarrollar este tipo de sistemas son:

Opción	Sistemas Operativos	Compatibilidad SQL	Capacidades adicionales
Sybase System 11	Gran variedad de sabores de UNIX, Linux (gratuito)	Compatibilidad completa con SQL 1	Lenguaje de transacciones, cursores y procedimientos almacenados activados por evento
Oracle 8	Gran variedad de sabores de UNIX, Linux (gratuito), Windows NT	Compatibilidad completa con SQL 1	Lenguaje de transacciones, cursores, procedimientos almacenados activados

			por evento y generación de índices especiales
Informix	Gran variedad de sabores de UNIX, Linux, Windows NT	Compatibilidad completa con SQL 1	Lenguaje de transacciones, cursores, procedimientos almacenados activados por evento y generación de índices especiales
Mini SQL	Gran variedad de sabores de UNIX	Compatible con casi todo el SQL 1 (faltan funciones de fechas)	Software gratuito para uso académico
Postgres SQL	Gran variedad de sabores de UNIX, Windows NT	Compatible con el SQL 1	Software libre, Lenguaje de transacciones, cursores, procedimientos almacenados activados por evento y generación de índices especiales

La selección de base de datos a emplear depende por lo regular del número de registros que se desee procesar, pero los sistemas de Bases de Datos mencionados no tienen más limitaciones que los que el sistema les imponga a este respecto, por lo que debemos tomar en cuenta las funciones que se espera que la base de datos tenga, así como el costo que deseamos al momento de elegir alguno de estos. El que tiene un costo menor, tanto en el costo de la licencia, como en los recursos de máquina que consume y por la amplia disponibilidad de personal capacitado en su uso en la UNAM es el Mini SQL, por lo que se desarrolló bajo su versión 2.1. Debe notarse que la decisión no debe considerarse como definitiva, ya que a la larga, la falta de procedimientos almacenados y las peculiaridades en su SQL son razón suficiente para pensar en substituirlo en versiones posteriores y todas las facilidades para este cambio deben tomarse en cuenta. Debe mencionarse que para aplicaciones institucionales, el servicio que se puede obtener en el mantenimiento de los servicios de Bases de Datos es un factor crucial en la decisión, por lo que es deseable que en aquellas instituciones donde se puedan dedicar los recursos a la compra de un sistema de bases de datos con contratos de soporte disponibles (tales como Sybase y Oracle) se pueda adaptar o generar versiones especiales del sistema, que puedan almacenar su información en estos sistemas. Postgres compete en igualdad con las alternativas comerciales, sin embargo, no hay suficiente experiencia en las instituciones educativas actualmente para darle el soporte necesario.

b) Servidor Web

Los servidores Web son sistemas encargados de proporcionar el servicio de HTTP a los clientes que así lo soliciten, usualmente programas residentes con capacidades de filtrado, registro e incluso de identificación de los clientes y cifrado de la comunicación. Su función principal es la de controlar la entrega de materiales que se encuentren en el servidor y en su caso, ejecutar programas locales cuyos resultados funcionen como un material disponible para consulta.

En la parte de Servidor HTTP se encuentran como principales opciones:

Opción	Estándar	Plataformas	Características
Netscape Enterprise Server	Servicio: HTML 3 Administración: Sigue el estilo de administración del NCSA HTTPD.	Solaris, Irix, HP-UX, Aix, Linux, Windows 95/NT, Mac OS	Sistema gráfico de administración, Gratuito para aplicaciones académicas, Contratos de mantenimiento y asesoría disponibles
Internet Information Server de Microsoft	Servicio: HTML 3 Administración: Se configura junto con el ambiente de NT	Windows NT 4.0 Service Pack 3 o superior	Sistema gráfico de administración, Integrado al sistema de usuarios de NT, Contratos de mantenimiento y asesoría disponibles
Apache	Servicio: HTML 3 Administración: Sigue el estilo del NCSA HTTPD.	Solaris, Irix, HP-UX, Aix, Linux, Windows 95/NT, Mac OS	Software libre, Uno de los servidores más populares en el mundo, Contratos de mantenimiento y asesoría disponibles por empresas independientes, Muy buen desempeño.

Adicionalmente a estas alternativas, existe una gran cantidad de servidores WWW disponibles para cada plataforma, sin embargo, no todos tienen el soporte de comunidades de usuarios y de compañías, así como la disponibilidad de administradores con experiencia en su manejo. De estos servicios, el que resulta más conveniente utilizar es el Apache, no solo por la gran disponibilidad de información técnica y por ser software libre, sino por ser uno de los servidores más robustos, disponible para la mayor cantidad de sistemas operativos y con mayor apego a los estándares. HTTP son las sigas del estándar de operaciones y lenguaje de definición de los documentos de hipertexto empleados en WWW.

c) Cliente Web

Los programas comúnmente conocidos como “Browsers” o Navegadores de red, que son los verdaderos clientes de HTTP cuentan con una enorme variedad que no solo se diferencia por sus características sino por la plataforma para la que son desarrollados, a continuación se mencionan varios de ellos no para elegir uno al que se limite el sistema, sino para detectar las características con que se puede contar en la mayoría de ellos de modo que el sistema sea compatible con un máximo de ellos.

Navegador	Java	Tablas	Tamaño de letra	Color de letra	Javascript
Explorer 5.0 Windows	*	*	*	*	*
Explorer 4.0 Windows	*	*	*	*	*
Explorer 3.0 Windows	*	*	*	*	*
Explorer 2.0 Windows		*	*	*	
Explorer 1.0 Windows		*	*	*	
Mosaic 3.0 Windows		*	*		
Mosaic 1.0 Windows					
Navigator 4.0 Windows	*	*	*	*	*
Navigator 3.0 Windows	*	*	*	*	*
Navigator 2.0 Windows	*	*	*	*	*
Navigator 1.0 Windows		*	*		
Opera 3.5 Windows		*	*	*	*
Lynx Windows		*			
Cyberdog 2.0 Macintosh		*	*	*	
Explorer 4.0 Macintosh	*	*	*	*	*
Explorer 3.0 Macintosh	*	*	*	*	*
Explorer 2.0 Macintosh		*	*	*	
Mosaic 3.07 Macintosh		*	*	*	
Mosaic 2.0 Macintosh		*			
Mosaic 1.0 Macintosh					
Navigator 4.5 Macintosh	*	*	*	*	*
Navigator 4.6 Macintosh	*	*	*	*	*
Navigator 3.0 Macintosh	*	*	*	*	*
Navigator 2.0 Macintosh		*	*	*	
Navigator 1.0 Macintosh		*	*		
Explorer 4.1 UNIX	*	*	*	*	*
Mosaic 2.75 UNIX		*			
Mosaic 1.0 UNIX					
Navigator 4.6 UNIX	*	*	*	*	*
Navigator 3.0 UNIX	*	*	*	*	*

Navigator 2.0 UNIX	*	*	*	*	
Navigator 1.1 UNIX		*	*		
Arena 3 UNIX		*	*		
Lynx UNIX		*			
Web TV, Web Pal SetTop Box		*	*	*	*
Navigator 2.2 OS/2		*	*	*	
Web Explorer 1.1 OS/2		*	*	*	
Opera OS/2		*	*	*	*
Lynx OS/2		*			

Las características que se consideran necesarias para el sistema son: Tablas, control sobre el tamaño y control de las letras, y el soporte a formularios y galletas (estas últimas se encuentran disponibles en todos los navegadores por lo que no se incluyeron en la tabla). La mayoría de los navegadores cuentan con estas características por lo que se cumple el objetivo de mantener al sistema compatible con una gran variedad de plataformas.

Resulta conveniente dejar en la mayor libertad posible al usuario de elegir el navegador que resulte más adecuado al equipo con que cuenta, teniendo cuidado de especificar en las características del sistema las normas a las que el navegador que se elija deberá apegarse para operar adecuadamente. Si se logra evitar que el sistema requiera de capacidades adicionales a la norma de algún navegador la libertad de elección de los usuarios estará garantizada. A modo de recomendación y patrón para el desarrollo, se empleará Netscape Navigator por ser uno de los más completos y comunes, que trata de apegarse a los estándares de HTML.

Una de las ventajas que se logran con el estricto apego a los estándares más comunes es el que puede dejarse el mantenimiento del navegador a cargo del propio usuario, con lo que se reducen de manera dramática las necesidades de personal de apoyo técnico a los usuarios del sistema.

d) Lenguaje de Programación

Entre el Sistema Gestor de Bases de Datos (SGBD) y el Servidor HTTP (HTTPD) se requiere de construir una serie de programas que son los que establecen la funcionalidad del sistema, estos programas se encargan de conciliar las peticiones que se reciben por el servidor de Web con los comandos y protocolo de conexiones del SGBD, una vez que se obtienen o se almacenan los datos en el SGBD o se han hecho uso de los recursos de sistema adecuados, el propio CGI genera una respuesta en el formato adecuado para el HTTPD y la entrega a este para su envío al cliente.

Estos programas por su capacidad de adaptar las necesidades de comunicación de diversos servicios entre sí en un solo sistema coherente son llamados: "Common Gateway Interface" (CGI). De hecho, existen desde que se inventaron los primeros servicios en red como Gopher y LPR y se han desarrollado en casi todos los lenguajes de programación disponibles en las plataformas en que se alojaban los servidores.

Para desarrollar los CGI del sistema se requiere elegir un lenguaje de programación con la capacidad necesaria para no requerir de otros lenguajes (simplificando el desarrollo y mantenimiento) y las características que hagan más sencillo el trabajo requerido en la programación y documentación de los programas resultantes.

A continuación se realiza una tabla con las características más importantes de los lenguajes más comúnmente usados para desarrollar CGI para sistemas Web que emplean SGBD.

Opción	Características	Bibliotecas de funciones
C/C++	Lenguaje híbrido que al compilarse genera código directamente ejecutable que puede o no contar con librerías de carga dinámica, su programación se toma como patrón en cuanto a complejidad y dificultad	Proceso de CGI, en todas las bases de programación de conexión por Red
Perl	Lenguaje que se ejecuta en un intérprete o como módulo del HTTPD y que puede emplear las librerías de carga dinámica del sistema, programación sencilla y potentes capacidades de manejo de datos	Proceso de CGI, Interfaz con las principales bases de datos, Interfaz genérica a bases de datos, Programación de comunicaciones por Red, Proceso de correo electrónico
Shell	Es un lenguaje nativo a los sistemas UNIX, tiene un rendimiento inferior y debe apoyarse en programas para sus funciones más avanzadas	No hay
Python	Lenguaje de Scripts que se ejecuta en un intérprete, optimizado para aplicaciones de sistema y CGI	Proceso de CGI, Interfaces con las principales bases de datos, Interfaz para gráficos
Java	Lenguaje que requiere de un ambiente de	Proceso de CGI,

	ejecución (interprete) que en el caso de la programación para servicios de Red puede estar incluido en el programa servidor (Apache, Netscape y IIS), puede generar código empaquetado de modo que su revisión y modificación no sea posible después de distribuido	Interfaz con las principales bases de datos, Interfaz genérica a bases de datos, Programación de comunicaciones por Red
PHP3	Lenguaje que se implementa como módulo en servidores Web Apache, los programas son partes de páginas especiales. Especializado en la interacción con Bases de datos pero tiene severas limitaciones en la interacción con el sistema	Interfaces para las principales bases de datos, especializado en CGI

Se elige a Perl¹⁰ como lenguaje de programación por la facilidad que presta al desarrollo de sistemas tomando en cuenta que se puede emplear el interprete de Perl incluso en la distribución final para las distribuciones universitarias facilitando el mantenimiento del mismo.

Una alternativa muy popular es PHP3, sin embargo sus capacidades de procesamiento de datos son inferiores a las de los demás lenguajes de programación lo que lo hace poco recomendable para sistemas extensos o con grandes posibilidades de expansión y no está diseñado para acceder a los recursos del sistema.

4. Arquitectura de Cliente Abierto

Habiendo revisado las herramientas a utilizar para el desarrollo del sistema, debemos considerar con cuidado las relaciones y tareas que los diversos elementos del sistema deberán guardar para que los resultados obtenidos sean óptimos.

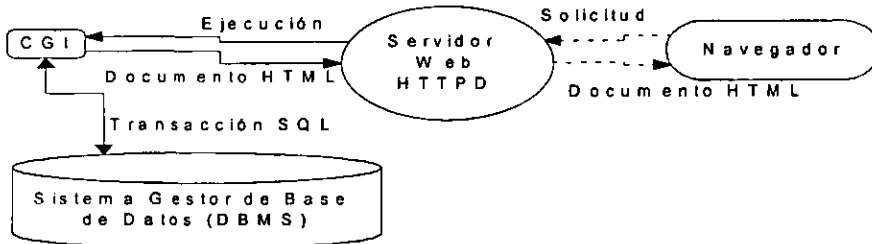
El sistema a construir será un sistema de Cliente abierto basado en WWW con apoyo de base de datos SQL. Resulta conveniente hacer una breve revisión de los conceptos asociados a los componentes en este tipo de sistemas.

La arquitectura de cliente abierto, es una evolución de la estructura cliente/servidor, en la que se incluye una pieza intermedia entre el servidor y el cliente, esta pieza es el servicio de Red, el que actúa como interfaz de comunicación entre programas clientes y

¹⁰ <http://www.perl.com>

servidores, permitiendo al emplear un estándar tal como HTTP que cualquier cliente estándar del servicio de Red funcione como cliente del sistema.

La estructura general de un sistema de cliente abierto es la siguiente:



El Sistema Gestor de Base de Datos (DBMS), es el encargado de almacenar la información con la que el sistema trabaja, es responsable de mantener la integridad y coherencia de la información en todo momento así como de verificar los permisos y características de acceso a la información de cada transacción y a menudo de realizar los respaldos de la información.

Los CGI son diversos programas desarrollados específicamente para el sistema, propiamente dicho, son el sistema junto con las formas HTML y los datos almacenados en la base de datos. Su misión primordial es la de realizar las operaciones necesarias para que el servicio WWW pueda entregar información a la Base de datos y para que los resultados que esta arroje sean presentados de modo adecuado al servicio y de ahí al cliente que los solicitó. Para que funcionen adecuadamente los CGI se deben hacer lo más reducidos que sea posible.

El servidor Web HTTPD, es el programa encargado de atender las peticiones de los clientes, las que se formulan en el estándar propio del servicio que se haya elegido, en este caso, HTTP para World Wide Web. Comparte con los clientes la responsabilidad de llevar a cabo las transacciones por red desde el equipo central hasta la terminal del usuario, y entrega las peticiones de los clientes a los CGI para después entregar el resultado que entregue a los clientes adecuados. Muchos servicios de Red cuentan además con capacidades de autenticación de usuarios, transacciones seguras y modos especiales de entrega de información.

Los Navegadores, son programas hechos específicamente para establecer transacciones con los servicios de Red, gracias a que se basan en un estándar que regula tanto el lenguaje en que han de formularse las peticiones y resultados, como la mecánica de comunicación y el modo de presentar los resultados, puede emplearse cualquier cliente

para presentar la información independientemente del programa que brinda el servicio de red, ya que ambos deben cumplir con el mismo estándar. Como siempre, algunos productos soportan características adicionales a las del estándar, pero por lo regular, esto no impide que mantengan en funcionamiento todo lo que el estándar especifica.

Durante el funcionamiento del sistema se realizan una gran cantidad de peticiones por la red, el ciclo de vida de una de estas peticiones puede ser descrita en general como sigue:

El **Navegador**, al recibir un URL por solicitar o al seguir un vínculo, genera una **Solicitud** que envía por red al **Servidor Web HTTPD**, al interpretar esta solicitud el **HTTPD** ejecuta el **CGI** adecuado y le entrega los parámetros que el **Navegador** especificó en su petición. El **CGI**, en base a su programación realiza una consulta al **Sistema Gestor de Base de Datos** a partir de los parámetros que recibe y con los resultados de la consulta genera un **Documento HTML** para ser enviados por el **Servidor Web** al **Navegador** en el que son presentados al usuario.

En este sistema en particular, se cuenta con una base de datos relativamente débil, por lo que algunas funciones de verificación de la integridad de la información recaen en los CGI, esto incrementa el esfuerzo necesario en el desarrollo del sistema, pero amplía las alternativas de Bases de Datos que se pueden emplear, ya que en caso de que la Base de Datos cuente con todas las funciones de verificación de integridad solo será necesario suspender las actividades de los CGI que resulten redundantes. El servicio WWW actuará primordialmente como interfaz entre los clientes y los CGI resultando invisible casi todo el tiempo, sin embargo, se plantea la construcción de una interfaz basada en WWW para administración del propio sistema, esta estará protegida solo por las capacidades de control de acceso del servicio con su sistema de identificación por contraseña, siendo el administrador el único usuario del que tendrá conocimiento el servicio WWW. Se permitirá el uso de cualquier cliente que soporte HTML 2.0 y "cookies" o galletas, estas últimas son parejas de nombre y valor que permiten identificar a los clientes a lo largo de varias transacciones.

B. Consideraciones para el diseño

Las metodologías tradicionales de diseño de sistemas, se han aplicado con éxito al desarrollo de aplicaciones centralizadas, donde el proceso y el almacenamiento de información se realiza en un solo ordenador. Y en fechas más recientes en sistemas cliente servidor en los que el almacenamiento y proceso de los datos se realiza en una computadora central y el proceso asociado a la presentación e interacción con el usuario

se realiza en un grupo de computadoras más pequeñas que realizan las mismas funciones obteniendo los datos del servidor.

¿Se puede afirmar que las mismas metodologías de diseño de sistemas puedan aplicarse a proyectos como el presente?, Donde los datos y su proceso esta centralizado, como en la arquitectura cliente servidor, pero que difiere de esta en que los programas que brindan la comunicación, y la presentación de resultados no están hechos acorde a las necesidades del sistema, sino para las necesidades en general de un esquema de comunicación, es decir, son servicios estándar de red.

Aún más importante que la distribución de la información, en los sistemas actuales se realizan tareas muy complejas en sistemas que se desarrollan con facilidad gracias a las capacidades especializadas de los lenguajes de desarrollo y de los servicios sobre los que se construye el sistema, de modo que el diseño se ve forzado a ignorar gran parte de la funcionalidad real del sistema o resultaría más complicado que la propia parte que se desarrolla.

Muchos aspectos de la implementación del sistema quedan fuera del diseño en una de estas metodologías, por ejemplo, el orden en el tiempo en que las diversas acciones se realizan o las diversas funciones del sistema que se implementan en un solo programa, esto es en parte una ventaja ya que al no tener un estado bien definido este tipo de sistemas, no se garantiza el orden de las acciones, del mismo modo, se goza de una casi total libertad en cuanto a los parámetros con que las diversas partes del programa serán utilizadas por lo que las posibilidades que tendrían que considerarse en un diseño formal serían demasiadas.

En resumen, la magnitud del sistema es tal, que una metodología de diseño aplicada de modo estricto generaría una cantidad de información excesiva y por lo tanto, de poca utilidad, sin embargo, gracias a los apoyos que las especializaciones de los lenguajes de programación, en particular Perl, podemos depender de las metodologías de desarrollo solo para dar una visión global coherente del sistema dejando los detalles de implementación al propio programador.

El Análisis del sistema se realizó con la metodología UML (Unified Modeling Language) que a pesar de estar orientada a objetos contiene con mucho éxito con conceptos tales como las transacciones y los distintos comportamientos de cada módulo del sistema. La consistencia de la información y por lo tanto la operación queda garantizada porque la Base de Datos y las reglas inherentes se diseñan con una metodología de Entidad Relación Estricta.

Aplicando estos criterios para el diseño es imperiosa la necesidad de que el código sea lo más sencillo posible ya que los futuros desarrollos necesitarán revisarlo para

comprender el auténtico funcionamiento del sistema ya que tanto UML como el diseño de la base de datos no tratan los aspectos funcionales, por lo que a lo largo del desarrollo se fue bastante estricto en criterios de presentación del código como son la indentación, la correcta identificación de los programas y la continua búsqueda de hacer el código mismo lo más comprensible que sea posible sin afectar el rendimiento o la funcionalidad.

C. Diseño

Como ya se discutió en la Selección de elementos constitutivos (página 14), el sistema se implementó en una Arquitectura de Cliente Abierto utilizando los siguientes elementos:

- Servidor Web Apache versión 4.0¹¹
- Sistema Gestor de Base de Datos, MiniSQL versión 3.5¹²
- Perl Versión 5.004 (con módulos para CGI y para conectividad con MiniSQL)

En la programación de los CGI se practicaron una serie de medidas para facilitar su portabilidad y mantenimiento, entre las que destacan:

- Variables de configuración al principio del código para los parámetros que dependen de la instalación del sistema.
- Las secciones de código HTML constantes en los documentos generados por los CGI se leen de archivos estáticos que pueden ser editados como HTML convencional para facilitar las tareas de diseño gráfico y evitar la edición de programas por cuestiones de presentación.
- Para facilitar la navegación en el sistema se emplean parámetros vacíos en el URL de las diversas etapas de las operaciones para generar registros coherentes en los historiales de los navegadores.

El sistema cuenta con tres partes principales, el conjunto de programas CGI, la base de datos en la que se almacena la información y un área de documentos en la que se colocan los materiales creados por el sistema de modo que sean accesibles por de servidor HTTPD.

A continuación se verán los esquemas que representan la estructura y funcionamiento del sistema comenzando por los diagramas Entidad Relación y el reporte de tablas y

¹¹ <http://www.apache.org>

¹² <http://www.huges.com>

atributos que definen a la base de datos para después revisar los Diagramas de Relación de los CGI.

1. Base de Datos

El esquema de la Base de datos se realizó con el apoyo del software ER-Win/ERX 2.5.01 for Power Builder, empleando el esquema para Power Builder.

Como la sintaxis de Power Builder y de MiniSQL tienen ligeras discrepancias se generó un pequeño programa en Perl que realiza las conversiones necesarias y ejecuta los comandos en el SGBD, gracias a esto el diseño corresponde directamente con la base de datos real.

El diseño de la base de datos se realizó en etapas, agregando conforme fueron necesarias las tablas y relaciones que permitían almacenar la información cada vez más compleja requerida por el sistema, por esto el resultado final es aparentemente complejo cuando en realidad es la suma de diversos esquemas relativamente simples. A continuación se da una breve descripción de las necesidades que las diversas tablas cumplen y el modo en que se relacionan.

El control de las sesiones se centra en la tabla "**sesión**", que funciona como referencia del uso que se realiza del sistema, una sesión agrupa un conjunto de acciones realizadas por el usuario, que se almacenan en la tabla "**acción**" y que pueden estar referidas a un material. La sesión funciona también para identificar al usuario, por lo que se relaciona con este.

La información del usuario se centra en la tabla "**usuario**", se consideran usuarios por igual a aquellos que crean material y a quienes solo lo consultan por lo que se requiere de un sistema de permisos que regule las actividades que cada uno pueda realizar. El catálogo de permisos posibles para cada usuario se almacena en la tabla "**permisos**" que se relaciona con los usuarios por dos mecanismos distintos.

Para indicar que permisos son los que el usuario puede ejercer se relacionan en la tabla "**autoriza**".

Como una forma de facilitar la asignación inicial de permisos a un usuario según un perfil predefinido, se creó la tabla "**perfil**" en la que se describen los tipos básicos de usuarios que hay y se indican los permisos recomendados para cada categoría en la tabla "**recomienda**". De este modo, al crear un nuevo usuario solo es necesario seleccionar un perfil para el usuario y luego, si así se desea, otorgar o revocar permisos específicos.

Uno de los elementos centrales del sistema son los materiales que son creados y consultados por los usuarios, la tabla "**material**" contiene la información de catálogo que todo material debe tener y es especializada en las tablas "**hoja**", "**recurso**", "**prueba**" y "**html**" según el tipo de material de que se trate. Por el momento el sistema solo genera materiales tipo "**hoja**" y "**recurso**" que son los materiales HTML creados y las imágenes a incluir en estas respectivamente, pero se solicitó el considerar tipos adicionales para pruebas de rendimiento de los alumnos y para controlar materiales HTML creados fuera del sistema desde el principio por lo que se les incluyo de un modo que no perjudiquen la operación normal del sistema.

Resta el control de acceso de un usuario en particular a un conjunto restringido de materiales, para esto se empleó un esquema de grupos de usuarios almacenados en la tabla "**grupo**" al que los usuarios se relacionan mediante la tabla "**inscrito**" que son asignados a un curso en particular. Los cursos, descritos en la tabla "**curso**" son colecciones de materiales indicadas por la tabla "**emplea**". Cuando el grupo al que pertenece un usuario esta asignado a un curso en particular el usuario tiene acceso a los materiales que pertenecen al curso, con lo que se logra el control deseado.

b) Entidades

Entidad: permiso

Definición: Descripciones de todos los permisos reconocidos por el sistema

Indices: XPKpermiso (valor único) *Atributos:* idprmso (asendente)

Entidad: accion

Definición: Registro de las acciones que los usuarios realizan en el sistema

Indices: XIF16accion *Atributos:* idses (asendente)

XIF17accion *Atributos:* idmat (asendente)

Entidad: autoriza

Definición: Entidad relacional en que se especifican los permisos con los que efectivamente cuenta el usuario

Indices: XIF24autoriza *Atributos:* idprmso (asendente)

XIF25autoriza *Atributos:* idusr (asendente)

Entidad: curso

Definición: Es la definición de las entidades (cursos) bajo las que se agruparán los materiales de modo que se puedan asignar a uno o varios grupos de usuarios

Indices: XIF14curso *Atributos:* idusr (asendente)

XPKcurso (valor único) *Atributos:* idcurso (asendente)

Entidad: emplea

Definición: Entidad relacional que asocia los materiales con el curso correspondiente

Indices: XIF21emplea *Atributos:* idmat (asendente)

XIF22emplea *Atributos:* idcurso (asendente)

Entidad: grupo

Definición: Entidad en la que se asocian los usuarios a un curso

Indices: XIF11grupo *Atributos:* idcurso (asendente)

XPKgrupo (valor único) *Atributos:* idgpo (asendente)

Entidad: hoja

Definición: Detalle de material HTML creado con ayuda del sistema

Indices: XIF2hoja *Atributos:* idmat (asendente)

XPKhoja (valor único) *Atributos:* idmat (asendente)

Entidad: html

Definición: Detalle de los conjuntos de páginas y recursos generados fuera del sistema que solo se van a controlar (no utilizado)

Indices: XIF19html *Atributos:* idmat (asendente)
 XPKhtml (valor único) *Atributos:* idmat (asendente)

Entidad: inscrito

Definición: Entidad asociativa que refleja la operación de inscribir un usuario a un grupo

Indices: XIF10inscrito *Atributos:* idgpo (asendente)
 XIF9inscrito *Atributos:* idusr (asendente)
 XPKinscrito (valor único) *Atributos:* idusr (asendente)

Entidad: material

Definición: Entidad en que se almacena la información de catálogo de un material

Indices: XIF13material *Atributos:* idusr (asendente)
 XIF23material *Atributos:* super (asendente)
 XPKmaterial (valor único) *Atributos:* idmat (asendente)

Entidad: perfil

Definición: Especificación de los permisos a otorgar por defecto a una categoría de usuarios

Indices: XPKperfil (valor único) *Atributos:* categoría (asendente)

Entidad: prueba

Definición: Detalle de un material de tipo prueba (no utilizado)

Indices: XIF6prueba *Atributos:* idmat (asendente)
 XPKprueba (valor único) *Atributos:* idmat (asendente)

Entidad: recurso

Definición: Detalle de un material de tipo imagen o recurso

Indices: XIF4recurso *Atributos:* idmat (asendente)
 XPKrecurso (valor único) *Atributos:* idmat (asendente)

Entidad: sesion

Definición: Elementos que permiten rastrear la actividad del usuario entre transacciones y conservar su autorización de acceso

Indices: XIF15sesion *Atributos:* idusr (asendente)
 XPKsesion (valor único) *Atributos:* idses (asendente)

Entidad: sugiere

Definición: Entidad relacional para especificar los permisos sugeridos para cada categoría

Indices: XIF28sugiere *Atributos:* idprmso (asendente)
 XIF29sugiere *Atributos:* categoría (asendente)

Entidad: usuario

Definición: Información general de los usuarios del sistema

Indices: XIF27usuario *Atributos:* categoría (asendente)

XPKusuario (valor único) *Atributos:* idusr (asendente)

c) Relaciones

Nombre: aplica

Entidad Padre: permiso

Verbo de Relación: aplica

Entidad Hija: autoriza

Llave Foránea: idprms0

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Uno o Muchos
(P)

Regla de Padre a Hijo: Un permiso aplica al menos en una autorización.

Regla de Hijo a Padre: Una autorización se asocia exactamente con un permiso.

Nombre: autoriza

Entidad Padre: usuario

Verbo de Relación: autoriza

Entidad Hija: autoriza

Llave Foránea: idusr

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Uno o Muchos
(P)

Regla de Padre a Hijo: Un usuario autoriza al menos a una autorización.

Regla de Hijo a Padre: Una autorización se asocia con exactamente un usuario.

Nombre: auxilia

Entidad Padre: material

Verbo de Relación: auxilia

Entidad Hija: emplea

Llave Foránea: idmat

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Uno o Muchos
(P)

Regla de Padre a Hijo: Un material auxilia al menos a un emplea.

Regla de Hijo a Padre: Un emplea se asocia con exactamente un material.

Nombre: califica

Entidad Padre: perfil

Verbo de Relación: califica

Entidad Hija: usuario

Llave Foránea: categoría

Tipo de relación: No identifica

Nulos: Nulos Permitidos

Cardinalidad: Cero o Uno, a Cero, Uno o Muchos

Regla de Padre a Hijo: Un perfil califica muchos usuarios.

Regla de Hijo a Padre: Un usuario se asocia con cero o un perfil.

Nombre: contrata

Entidad Padre: usuario

Verbo de Relación: contrata

Entidad Hija: inscrito

Llave Foránea: idusr

Definición: Los usuarios, mediante las inscripciones, adquieren ciertos derechos sobre los diversos cursos, primordialmente el de revisarlos y ser atendidos por el personal de apoyo,

asimismo, el personal de apoyo se inscribe a los cursos con mayores privilegios para poder hacer frente a las necesidades que se le planteen.

Tipo de relación: Identifica

Cardinalidad: Uno, a Uno o Muchos (P)

Regla de Padre a Hijo: Un usuario contrata al menos a una inscripción.

Regla de Hijo a Padre: Una inscripción se asocia con exactamente un usuario.

Nombre: contribuye

Entidad Padre: usuario

Verbo de Relación: contribuye

Entidad Hija: material

Llave Foránea: idusr

Definición: Los materiales son entregados al sistema por algún usuario, el que conserva los privilegios de modificación. Posteriormente puede adjudicarse un material a otro usuario si se desea transferir los privilegios.

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Uno o Muchos (P)

Regla de Padre a Hijo: Un usuario contribuye al menos a un material.

Regla de Hijo a Padre: Un material se asocia con exactamente un usuario.

Nombre: diseña

Entidad Padre: usuario

Verbo de Relación: diseña

Entidad Hija: curso

Llave Foránea: idusr

Definición: Los cursos definen conjuntos de materiales que deben ser revisados. Su diseño se realiza independientemente de la existencia y

naturaleza los propios materiales por lo que tienen un autor como tal.

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Uno o Muchos (P)

Regla de Padre a Hijo: Un usuario diseña al menos a un curso.

Regla de Hijo a Padre: Un curso se asocia con exactamente un usuario.

Nombre: establece

Entidad Padre: inscrito

Verbo de Relación: establece

Entidad Hija: sesion

Llave Foránea: idusr

Definición: Cuando un usuario inscrito emplea sus privilegios de acceso comienza una sesión, la que enmarcará una serie de acciones. La importancia de la sesión está en el análisis del trabajo del usuario.

Tipo de relación: No identifica

Nulos: Nulos Permitidos

Cardinalidad: Cero o Uno, a Cero, Uno o Muchos

Regla de Padre a Hijo: Un inscrito establece muchas sesiones.

Regla de Hijo a Padre: Una sesión se asocia con cero o un inscritos.

Nombre: ocupa

Entidad Padre: curso

Verbo de Relación: ocupa

Entidad Hija: grupo

Llave Foránea: idcurso

Definición: Los diversos grupos solo pueden asociarse a grupos de usuarios para ser empleados mediante grupos, como una medida de garantizar la disponibilidad de instructores y

ayudantes en el desarrollo de las actividades.

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Cero, Uno o Muchos

Regla de Padre a Hijo: Un curso ocupa muchos grupos.

Regla de Hijo a Padre: Un grupo se asocia con exactamente un curso.

Nombre: realiza

Entidad Padre: sesion

Verbo de Relación: realiza

Entidad Hija: accion

Llave Foránea: idses

Definición: Al interactuar con los materiales puestos a su disposición el usuario va construyendo una historia de uso que puede ser muy útil para mejorar su rendimiento y los propios materiales.

Tipo de relación: No identifica

Nulos: Nulos Permitidos

Cardinalidad: Cero o Uno, a Uno o Muchos (P)

Regla de Padre a Hijo: Una sesión realiza al menos a una acción.

Regla de Hijo a Padre: Una acción se asocia con cero o una sesión.

Nombre: recibe

Entidad Padre: grupo

Verbo de Relación: recibe

Entidad Hija: inscrito

Llave Foránea: idgpo

Definición: Los grupos se alimentan de usuarios mediante las inscripciones.

Tipo de relación: No identifica

Nulos: Nulos Permitidos

Cardinalidad: Cero o Uno, a Uno o Muchos (P)

Regla de Padre a Hijo: Un grupo recibe al menos a un inscrito.

Regla de Hijo a Padre: Un inscrito se asocia con cero o un grupos.

Nombre: recomienda

Entidad Padre: permiso

Verbo de Relación: recomienda

Entidad Hija: sugiere

Llave Foránea: idprmso

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Uno o Muchos (P)

Regla de Padre a Hijo: Un permiso es recomendada al menos por un sugiere.

Regla de Hijo a Padre: Una sugerencia se asocia con exactamente un permiso.

Nombre: revisa

Entidad Padre: curso

Verbo de Relación: revisa

Entidad Hija: emplea

Llave Foránea: idcurso

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Uno o Muchos (P)

Regla de Padre a Hijo: Un curso revisa al menos a un emplea.

Regla de Hijo a Padre: Un emplea se asocia con exactamente un curso.

Nombre: subtema

Entidad Padre: material

Verbo de Relación: subtema

Entidad Hija: material

Llave Foránea: super

Tipo de relación: No identifica

Nulos: Nulos Permitidos

Cardinalidad: Cero o Uno, a Cero, Uno o Muchos

Relaciones "es un"

Regla de Padre a Hijo: Un material tiene por subtemas a muchos materiales.

Regla de Hijo a Padre: Un material es subtema de cero o un materiales.

Nombre: sugiere

Entidad Padre: perfil

Verbo de Relación: sugiere

Entidad Hija: sugiere

Llave Foránea: categoría

Tipo de relación: No identifica

Nulos: No Nulos

Cardinalidad: Uno, a Uno o Muchos (P)

Regla de Padre a Hijo: Un perfil sugiere al menos a una sugerencia.

Regla de Hijo a Padre: Una sugerencia se asocia con exactamente un perfil.

Nombre: sustenta

Entidad Padre: material

Verbo de Relación: sustenta

Entidad Hija: accion

Llave Foránea: idmat

Definición: Los materiales son los objetos sobre los que los usuarios realizan las acciones.

Tipo de relación: No identifica

Nulos: Nulos Permitidos

Cardinalidad: Cero o Uno, a Cero Uno o Muchos

Regla de Padre a Hijo: Un material sustenta muchas acciones.

Regla de Hijo a Padre: Una acción se asocia con cero o un materiales.

Entidad Padre: material

Verbo de Relación: es una

Entidad Hija: hoja

Llave Foránea: idmat

Tipo de relación: Subtype

Cardinalidad: Es un (Is a)

Regla de Hijo a Padre: Una hoja es un tipo de material.

Entidad Padre: material

Verbo de Relación: es un

Entidad Hija: html

Llave Foránea: idmat

Tipo de relación: Subtipo

Cardinalidad: Es un (Is a)

Regla de Hijo a Padre: Un html es un tipo de material.

Entidad Padre: material

Verbo de Relación: es un

Entidad Hija: prueba

Llave Foránea: idmat

Tipo de relación: Subtipo

Cardinalidad: Es un (Is a)

Regla de Hijo a Padre: Un prueba es un tipo de material.

Entidad Padre: material

Verbo de Relación: es un

Entidad Hija: recurso

Llave Foránea: idmat

Tipo de relación: Subtipo

Cardinalidad: Es un (Is a)

Regla de Hijo a Padre: Un recurso es un tipo de material.

d) Atributos**Entidad: accion****Atributo: hecho**

Columna: hecho

Definición: Acción realizada

Tipo: Owned Non-key

Tipo de Datos: CHAR(1)

Valores Nulos: NOT NULL

Atributo: idmat

Definición: Identificador del recurso

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: idses

Definición: Identificador de la sesión

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Atributo: momfe

Definición: Fecha en que se realiza la operación

Tipo: Owned Non-key

Tipo de Datos: DATE

Valores Nulos: NOT NULL

Atributo: momho

Definición: Hora en que se realiza la operación

Tipo: Owned Non-key

Tipo de Datos: TIME

Valores Nulos: NOT NULL

Entidad: permiso**Atributo: acciones**

Definición: Especificación de las acciones que se permitirá realizar a los usuarios que tengan este permiso

Tipo: Owned Non-key

Tipo de Datos: CHAR(50)

Valores Nulos: NOT NULL

Atributo: idprmso

Definición: Identificador de un permiso, debe ser una palabra desplegable en los menús de administración

Tipo: Owned Key

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Entidad: material**Atributo: almacén**

Definición: Información de sistema respecto a la ubicación física del material original

Tipo: Owned Non-key

Tipo de Datos: CHAR(60)

Nota: Es la ruta (relativa) en donde se almaceno el archivo que dio origen a este recurso.

Atributo: fecha

Definición: Fecha en que se creó o en su defecto, en que se agregó al sistema

Tipo: Owned Non-key

Tipo de Datos: DATE

Valores Nulos: NOT NULL

Atributo: idmat

Definición: Identificador del recurso

Tipo: Owned Key

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: **super**

Columna: idmat

Definición: Identificador del recurso

Tipo: Owned Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Nota: Es la llave del material padre, en caso de ser nulo se consideran como independientes

Atributo: **idusr**

Definición: identificador único para cada usuario, en el caso de universitarios se recomienda el uso del número de cuenta.

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(9)

Valores Nulos: NOT NULL

Atributo: **ordinal**

Definición: Lugar a ocupar entre los materiales hijos de un mismo material

Tipo: Owned Non-key

Tipo de Datos: INTEGER

Nota: Especifica el orden relativo respecto a los otros materiales hijos de algún otro, en caso de ser igual al de otro se les colocara por orden de identificador

Atributo: **resumen**

Definición: Breve descripción del contenido y propósito del material

Tipo: Owned Non-key

Tipo de Datos: CHAR(250)

Atributo: **titulo**

Definición: Título del recurso

Tipo: Owned Non-key

Tipo de Datos: CHAR(50)

Valores Nulos: NOT NULL

Entidad: **usuario**

Atributo: **alta**

Definición: Fecha desde la cual el usuario forma parte del sistema

Tipo: Owned Non-key

Tipo de Datos: DATE

Valores Nulos: NOT NULL

Atributo: **apmat**

Definición: Apellido materno del usuario

Tipo: Owned Non-key

Tipo de Datos: CHAR(15)

Atributo: **appat**

Definición: Apellido paterno del usuario

Tipo: Owned Non-key

Tipo de Datos: CHAR(15)

Valores Nulos: NOT NULL

Atributo: **categoria**

Definición: Es la palabra que identifica la categoría a signar a los usuarios

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(20)

Atributo: **idusr**

Definición: identificador único para cada usuario, en el caso de universitarios se

recomienda el uso del número de cuenta.

Tipo: Owned Key

PK: (PK)

Tipo de Datos: CHAR(9)

Valores Nulos: NOT NULL

Nota: puede o no implementarse un mecanismo de validación, por ejemplo, en caso de que solo se tengan alumnos de una institución educativa que ya les de matrícula.

Atributo: **institucion**

Definición: Institución de origen del usuario

Tipo: Owned Non-key

Tipo de Datos: CHAR(50)

Atributo: **login**

Definición: Identificador del usuario para sistema de autenticación

Tipo: Owned Non-key

Tipo de Datos: CHAR(8)

Atributo: **nacimiento**

Definición: Fecha de nacimiento

Tipo: Owned Non-key

Tipo de Datos: DATE

Valores Nulos: NOT NULL

Atributo: **nombre**

Definición: Nombre o nombres del usuario

Tipo: Owned Non-key

Tipo de Datos: CHAR(30)

Valores Nulos: NOT NULL

Atributo: **passwd**

Definición: Password del usuario para tener acceso al sistema

Tipo: Owned Non-key

Tipo de Datos: CHAR(8)

Entidad: **html**

Atributo: **archivos**

Definición: Número de archivos colocados, una medida de la magnitud de las páginas

Tipo: Owned Non-key

Tipo de Datos: INTEGER

Valores Nulos: NOT NULL

Atributo: **idmat**

Definición: Identificador del recurso

Tipo: Foreign Key

FK: (FK)

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: **indice**

Definición: Es la página principal del conjunto, usualmente un índice de los demás archivos.

Tipo: Owned Non-key

Tipo de Datos: CHAR(30)

Valores Nulos: NOT NULL

Atributo: **usuario**

Definición: Es el user-id de unix del usuario dueño de los archivos

Tipo: Owned Non-key

Tipo de Datos: CHAR(8)

Entidad: **inscrito**

Atributo: **atributo**

Definición: Especificación de las atribuciones del usuario dentro del grupo

Tipo: Owned Non-key

Tipo de Datos: CHAR(1)

Nota: Por ejemplo: alumno, ayudante, profesor, oyente, etc.

Atributo: **idgpo**

Definición: Identificador del grupo

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Atributo: **idusr**

Definición: identificador único para cada usuario, en el caso de universitarios se recomienda el uso del número de cuenta.

Tipo: Foreign Key

FK: (FK)

PK: (PK)

Tipo de Datos: CHAR(9)

Valores Nulos: NOT NULL

Atributo: **operacion**

Definición: Identificador de la operación por la que se inscribe al alumno

Tipo: Owned Non-key

Tipo de Datos: CHAR(10)

Atributo: **realizada**

Definición: Fecha en la que se inscribe al alumno

Tipo: Owned Non-key

Tipo de Datos: DATE

Entidad: **recurso**

Atributo: **bloques**

Definición: Número de bloques de 1024 bytes en el recurso

Tipo: Owned Non-key

Tipo de Datos: INTEGER

Atributo: **funcion**

Definición: Especifica si el recurso se empleará de uno u otro modo

Tipo: Owned Non-key

Tipo de Datos: CHAR(18)

Nota: Los modos que se tienen actualmente son:

g - general

h - divisor (barra en páginas HTML)

b - fondo

Atributo: **idmat**

Definición: Identificador del recurso

Tipo: Foreign Key

FK: (FK)

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Nota: Tipos permitidos: jpg, gif, ra, rm

Atributo: **tipo**

Definición: Tipo de recurso

Tipo: Owned Non-key

Tipo de Datos: CHAR(10)

Nota: Se piensa emplear la extensión como denominador de los diversos tipos y en caso de controversia agregar un identificador de la aplicación de servidor a emplear para su proceso o presentación.

Entidad: **perfil**

Atributo: **categoría**

Definición: Es la palabra que identifica la categoría a signar a los usuarios

Tipo: Owned Key

PK: (PK)

Tipo de Datos: CHAR(20)

Valores Nulos: NOT NULL

Nota: Cada categoría representa el conjunto de permisos que se sugiere se asignen a un usuario tipo, posteriormente puede afinar el detalle de los permisos que cada usuario tenga de modo que se puedan realizar las combinaciones que resulten más convenientes.

Atributo: **propósito**

Definición: Es la descripción del uso que se dará a la categoría

Tipo: Owned Non-key

Tipo de Datos: CHAR(50)

Valores Nulos: NOT NULL

Nota: Es la descripción de los usos que se le dará a la categoría especificando de modo cualitativo los privilegios que brinda al usuario y a quien debe aplicarse

Entidad: **sugiere**

Atributo: **categoría**

Definición: Es la palabra que identifica la categoría a signar a los usuarios

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(20)

Valores Nulos: NOT NULL

Atributo: **idprmso**

Definición: Identificador de un permiso, debe ser una palabra desplegable en los menús de administración

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Tabla: **grupo**

Atributo: **creacion**

Definición: Fecha en la que se inician labores

Tipo: Owned Non-key

Tipo de Datos: DATE

Valores Nulos: NOT NULL

Atributo: **funcion**

Definición: En caso de que los usuarios tengan un propósito común en el curso, se le especifica en este campo

Tipo: Owned Non-key

Tipo de Datos: CHAR(30)

Atributo: **idcurso**

Definición: Identificador del curso que se imparte a este grupo

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: **idgpo**

Definición: Identificador del grupo

Tipo: Owned Key

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: **termino**

Definición: Fecha en las que las labores relacionadas con el curso se dan por terminadas

Tipo: Owned Non-key

Tipo de Datos: DATE

Valores Nulos: NOT NULL

Entidad: sesion

Atributo: finfe

Definición: Fecha de terminación de la sesión

Tipo: Owned Non-key

Tipo de Datos: DATE

Atributo: finho

Definición: Hora de termino de la sesión

Tipo: Owned Non-key

Tipo de Datos: TIME

Atributo: idses

Definición: Identificador de la sesión

Tipo: Owned Key

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: idusr

Definición: identificador único para cada usuario, en el caso de universitarios se recomienda el uso del número de cuenta.

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(9)

Atributo: iniciofe

Definición: Fecha de inicio de la sesión

Tipo: Owned Non-key

Tipo de Datos: DATE

Valores Nulos: NOT NULL

Atributo: inicioho

Definición: Hora de inicio de la sesión

Tipo: Owned Non-key

Tipo de Datos: TIME

Valores Nulos: NOT NULL

Atributo: ipcli

Definición: Dirección IP del cliente permitido en la sesión

Tipo: Owned Non-key

Tipo de Datos: CHAR(16)

Entidad: curso

Atributo: idcurso

Definición: Identificador del curso

Tipo: Owned Key

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: idusr

Definición: identificador único para cada usuario, en el caso de universitarios se recomienda el uso del número de cuenta.

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(9)

Valores Nulos: NOT NULL

Atributo: objetivo

Definición: Es una descripción breve del propósito del curso

Tipo: Owned Non-key

Tipo de Datos: CHAR(100)

Atributo: titulo

Definición: Es el título del curso en su conjunto

Tipo: Owned Non-key

Tipo de Datos: CHAR(50)

Entidad: emplea

Atributo: idcurso

Definición: Identificador del curso

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: idmat

Definición: Identificador del recurso

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Entidad: hoja

Atributo: idmat

Definición: Identificador del recurso

Tipo: Foreign Key

FK: (FK)

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: lineas

Definición: Número de líneas en el archivo original

Tipo: Owned Non-key

Tipo de Datos: INTEGER

Entidad: prueba

Atributo: idmat

Definición: Identificador del recurso

Tipo: Foreign Key

FK: (FK)

PK: (PK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: mecanica

Definición: Tipo de aplicación del examen

Tipo: Owned Non-key

Tipo de Datos: CHAR(1)

Atributo: preguntas

Definición: Número de preguntas a aplicar en cada prueba

Tipo: Owned Non-key

Tipo de Datos: INTEGER

Entidad: autoriza

Atributo: idprmso

Definición: Identificador de un permiso, debe ser una palabra desplegable en los menús de administración

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(10)

Valores Nulos: NOT NULL

Atributo: idusr

Definición: identificador único para cada usuario, en el caso de universitarios se recomienda el uso del número de cuenta.

Tipo: Foreign Non-key

FK: (FK)

Tipo de Datos: CHAR(9)

Valores Nulos: NOT NULL

2. Interfaz (CGI)

Es en los programas CGI en los que se implementa en realidad la funcionalidad del sistema, ya que los demás elementos solo establecen los mecanismos de comunicación o de almacenamiento y recuperación de la información, por esto la metodología UML que se enfoca al desarrollo de sistemas resultó de especial utilidad en el análisis de las tareas a realizar por cada CGI.

a) Diagramas Unified Modeling Language (UML)

Una de las herramientas para análisis y diseño de sistemas que tienen mejor aceptación en la actualidad es el lenguaje conocido como UML, en parte ha tenido tanto éxito por estar vinculado con la programación orientado a objetos por lo que resulta muy adecuado a los lenguajes actuales, pero su principal virtud consisten en brindar una serie de esquemas sencillos que logran retratar la función de los diversos módulos que componen los grandes sistemas actuales.

Se realizó el diseño del sistema usando la metodología UML logrando una buena documentación de los diversos módulos y operaciones que se requieren, cada uno de los diagramas, como se verá mas adelante, es extremadamente simple, y la complejidad total del sistema puede medirse en función de la cantidad de operaciones que se debieron programar.

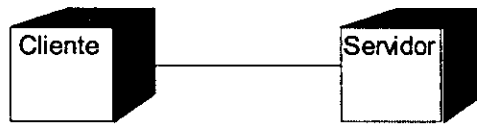
El diseño total del sistema consta de:

- 22 Diagramas de Caso de Uso (Use Case Diagram)
- 26 Diagramas de Secuencia y de Colaboración (Sequence Diagrams y Collaboration Diagrams)
- 17 Diagramas de Clases (Class Diagrams)
- 3 Diagramas de Componentes (Component Diagram)
- 1 Diagrama de Implementación (Deployment Diagram)

Se consideran como los mas representativos del sistema: el Diagrama de Implementación y algunos de los diagramas de Caso de Uso y de Secuencia, ya que son los que reflejan la organización y operaciones que tiene el sistema. En particular se incluyen los diagramas de Caso de Uso y de Secuencia de los módulos de Entrada, Salida, Creación de Materiales y de Administración de Cursos. Los primeros como ejemplo del uso que se

dio a la metodología en el caso particular de diseñar un sistema de cliente abierto en las áreas públicas del sistema, y el segundo para retratar las áreas de administración, ya que probaron estar mas cercanas a las aplicaciones tradicionales de base de datos que los demás módulos del sistema.

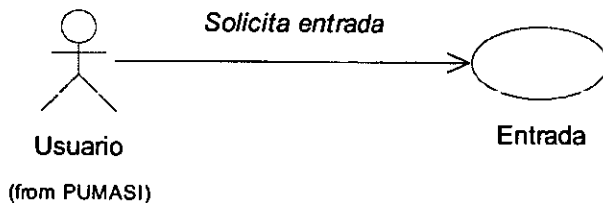
Diagrama de Implementación (Deployment Diagram)



Como se puede apreciar la implementación es bastante sencilla en sí, el cliente solo requiere tener un navegador compatible y el servidor contiene, en un solo equipo, los sistemas de Base de Datos, los CGI y el servidor HTTPD. Toda la comunicación entre ellos se realiza mediante el protocolo HTTP.

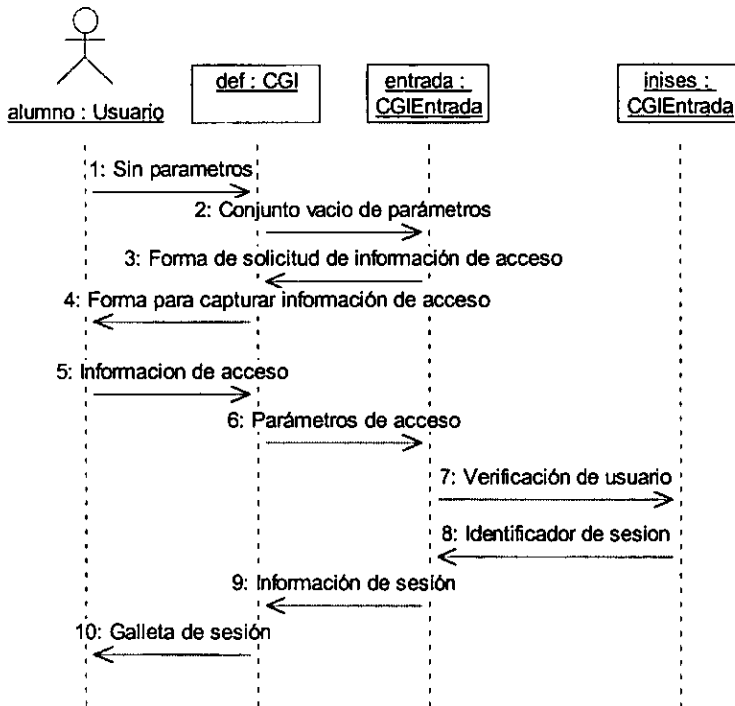
◆ **Entrada al Sistema**

Diagrama de Caso de Uso de la Entrada de un Usuario al Sistema



Esta es por lo regular la primera operación que un usuario realiza en el sistema cuando tiene acceso a él por primera vez o luego de que ha caducado una sesión, para revisar las operaciones necesarias para que la sesión sea autorizada y se pueda continuar usando el sistema tenemos el siguiente diagrama:

Diagrama de Secuencia de la Entrada de un Usuario al Sistema

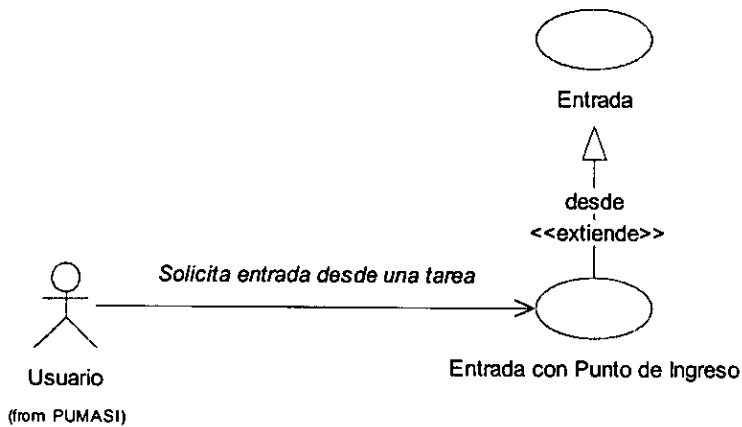


En este se retratan dos operaciones HTTP, una en la que el usuario recibe la forma en la que se le solicita identificarse y una vez que proporciona la información que lo identifica como un usuario válido del sistema, se le entrega la forma con las alternativas principales y la galleta con la que se permitirá que haga uso del sistema.

Uno de los aspectos que deben tomarse en cuenta cuando se desarrolla un sistema para clientes abiertos y en especial para Web, es que los clientes tienen total libertad de solicitar cualquier operación del sistema en cualquier momento, estas solo deben permitirse cuando estén dentro de un esquema coherente o se pondría en riesgo la integridad de la información, de modo que se deben prever avenidas de uso del sistema que no son estrictamente correctas y que deben ser encaminadas al funcionamiento normal del sistema.

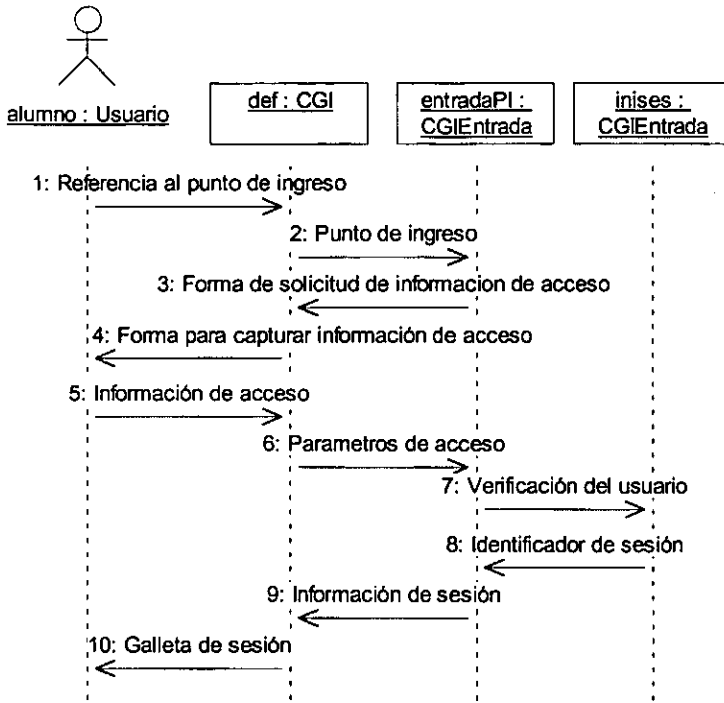
En el caso de la entrada al sistema, tenemos dos casos particulares de uso “no convencional”, uno de ellos es aquel en el que el usuario trata de acceder a una operación del sistema sin haberse identificado primero, lo que debe remitirlo a la entrada y permitirle continuar con la operación que deseaba.

Diagrama de Caso de Uso de la Entrada Solicitando una Operación



Para el diseño UML la entrada en la que el usuario solicita una operación del sistema es un caso particular de la entrada común, sin embargo, se tiene un parámetro adicional a tomar en cuenta que es la operación que el usuario solicitó originalmente. Revisando las operaciones de este caso de entrada tenemos:

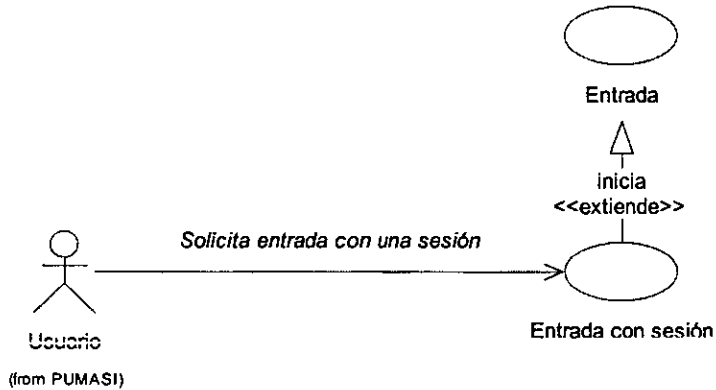
Diagrama de Secuencia de la Entrada Solicitando una Operación



Aún cuando los pasos a seguir son esencialmente los mismos, se puede distinguir que los atributos que se manejan en las operaciones son otros, ya que se desea que una vez que el usuario complete su identificación pueda pasar directamente a realizar la operación que solicitó en primer lugar.

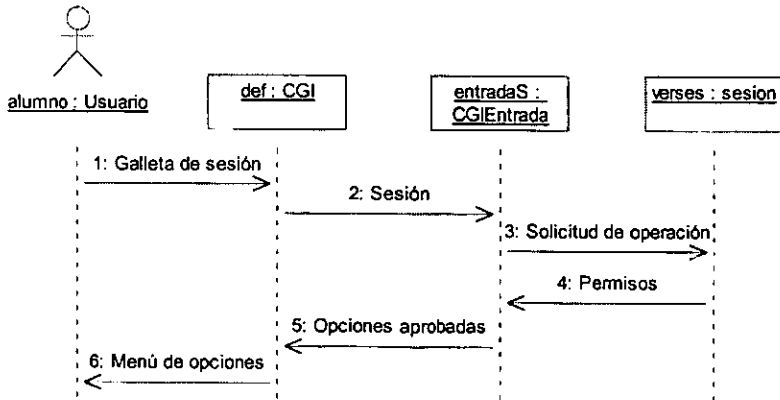
Otro caso particular de la entrada, es aquel en el que el usuario todavía cuenta con una galleta de sesión válida (recordemos que estas tienen una validez de varias horas, y que perduran incluso si el navegador es terminado y reiniciado en la misma máquina cliente siempre cuando no sean explícitamente terminadas).

Diagrama de Caso de Uso de la Entrada con una Sesión Válida



Este caso de uso también es un caso particular de la entrada, aunque la secuencia de operaciones se reduce notablemente.

Diagrama de Secuencia de la Entrada con una Sesión Válida



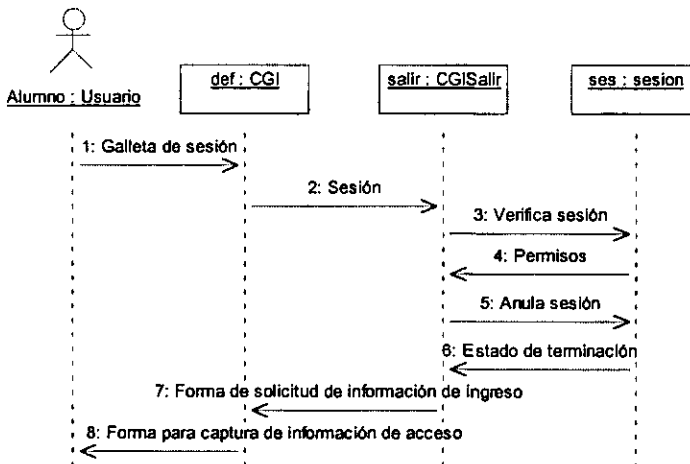
Cuando el usuario presenta una galleta de sesión válida, el sistema presenta las opciones que el usuario tiene como si terminara de identificarse, pero no renueva la sesión de modo que el tiempo para la caducidad de la misma sigue corriendo.

La contraparte de la entrada, y en la que se tiene de modo más directo los elementos mínimos de cada caso de uso y por lo tanto de cada interfaz CGI es la salida del sistema, en este se termina la sesión que el usuario presenta y se le remite a la entrada.

Diagrama de Caso de Uso de la Salida del Sistema



Diagrama de Secuencia de la Salida del Sistema



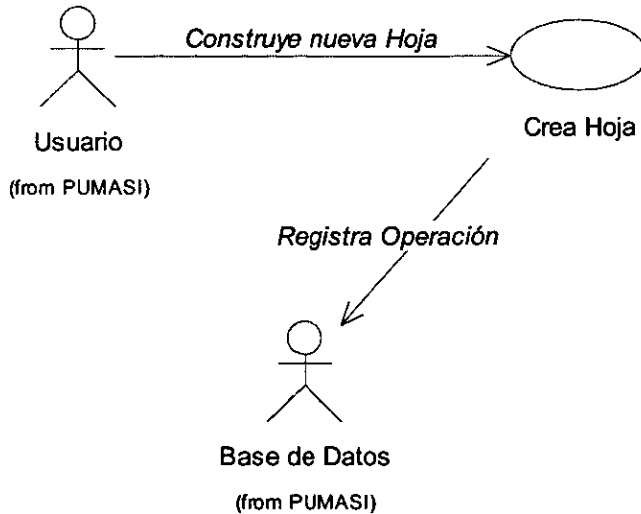
Uno de los aspectos en los que los sistemas de cliente abierto difieren de los Cliente Servidor es que el navegador, quien se encarga de la interfaz con el usuario, no solo es utilizado para acceder al sistema en cuestión, por lo que aún la indicación expresa de que no se desea continuar usando el sistema no es razón para terminar la aplicación, ya que el usuario puede emplearla para acceder a otros recursos en la red.

Ejemplificando el resto del diseño de Casos de Uso del sistema, se colocan la Creación de Materiales y la Administración de Cursos, en la primera tenemos la interacción con los archivos a ser usados por el servidor HTTPD y en la segunda se van las tareas comunes de los sistemas de bases de datos.

Creación de Materiales

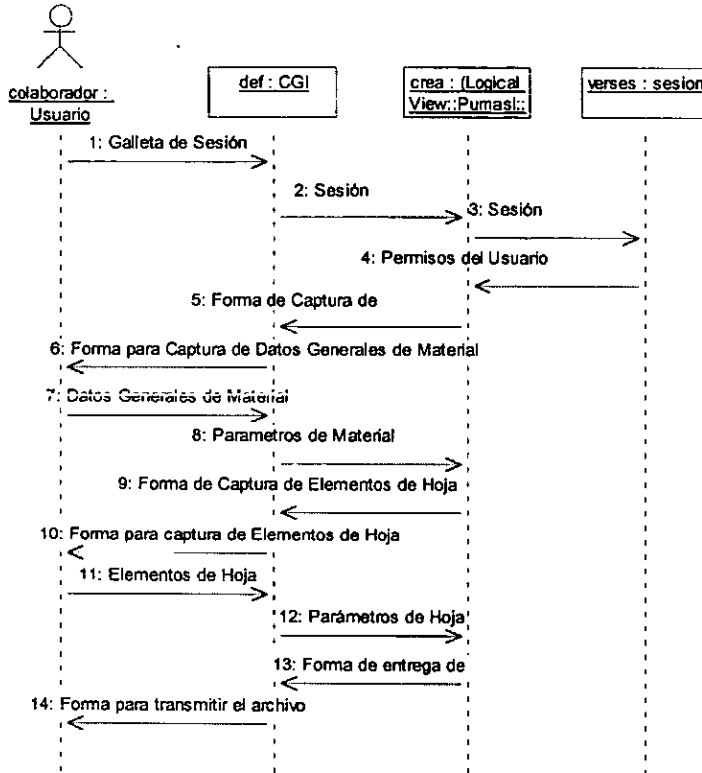
Se tienen dos casos particulares para la creación de materiales, el primero, en el que se desea crear una página a partir de un archivo de texto, lo que conocemos como “Hoja” y el segundo, en el que se desea crear un recurso para ser utilizado por las páginas, un “Recurso” como por ejemplo, una gráfica.

Diagrama de Caso de Uso de Creación de una Hoja



El proceso de creación de un material esta dividido en dos partes, la que se conoce como Creación que estamos revisando se encarga de recopilar la información necesaria para posteriormente recibir los archivos necesarios y efectivamente crear el archivo en un proceso distinto conocido como recepción. Es por esto que en el caso de uso de la creación no se tiene interacción con los elementos del servidor HTTPD.

Diagrama de Secuencia de Creación de una Hoja



El proceso de recopilar la información de la Hoja, está dividido en tres operaciones distintas de modo que en cada paso haya una noción del propósito de la información que se otorga y para que se puedan revisar los elementos de diseño que se emplearán en la hoja final.

Diagrama de Caso de Uso de Creación de Recurso

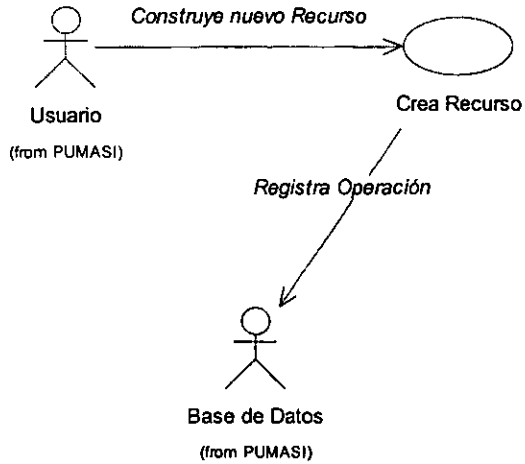


Diagrama de Secuencia de Creación de Recurso

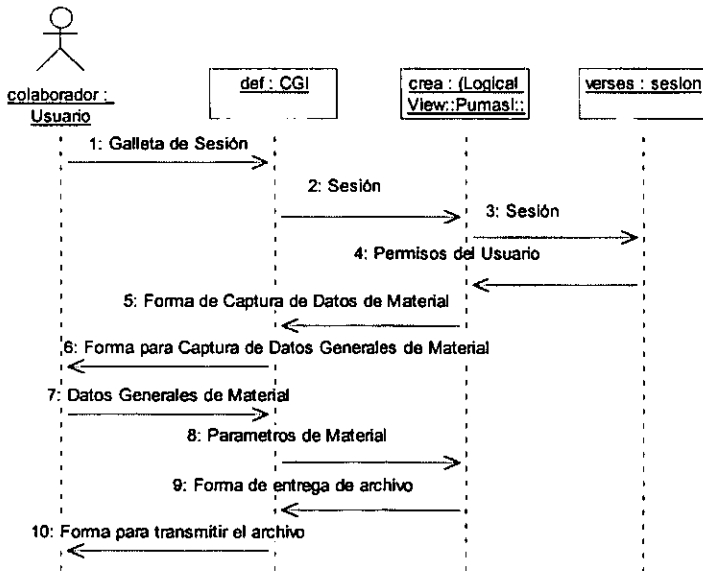


Diagrama de Caso de Uso de Recepción

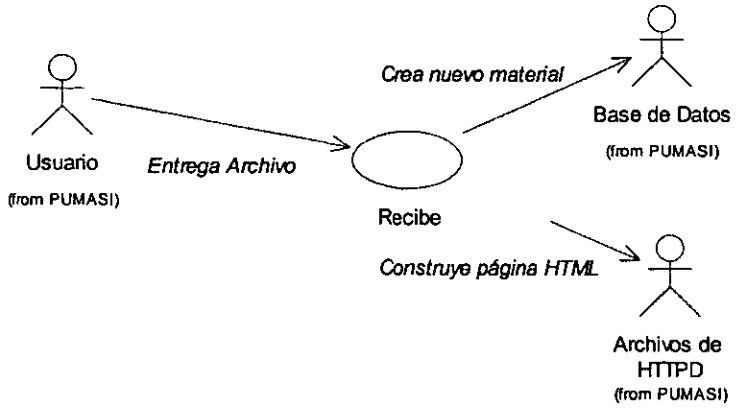
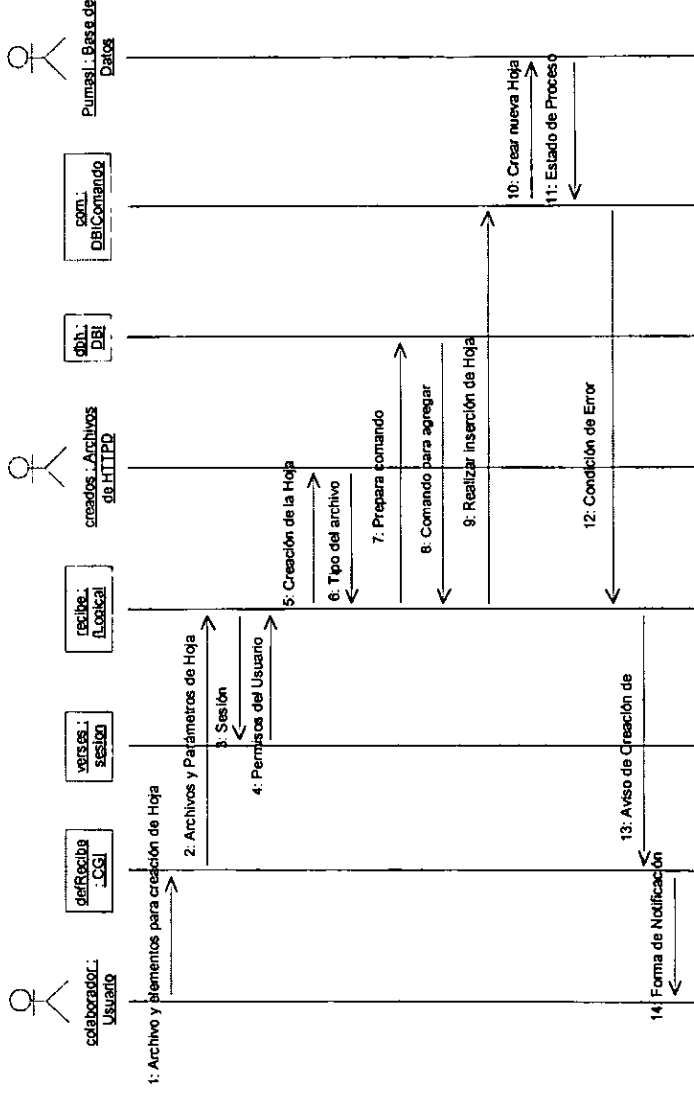


Diagrama de Secuencia de Recepción

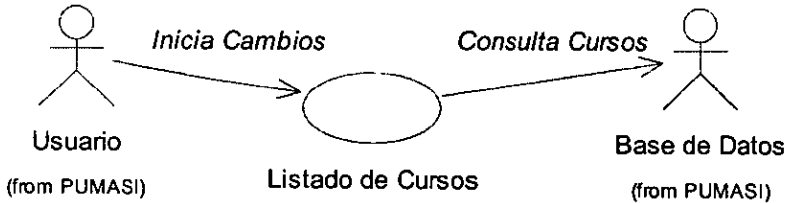


Como se puede apreciar en el diagrama este tipo de procesos son relativamente simples pero involucran a muchas entidades en el sistema, lo que las hace sensibles a las modificaciones que se realicen y que involucren la estructura o funcionamiento de los materiales creados.

◆ **Administración de Cursos**

Toda la sección de administración realiza las tareas clásicas de los sistemas de bases de datos, que son las altas, las bajas y los cambios de registros sobre tablas bien definidas. Adicionalmente realizan la asociación de los registros entre ellas, el funcionamiento de ellos sigue la misma estructura por lo que solo se mostrará el diseño de uno de ellos, el de administración de cursos, y las funciones que realizan ejemplifican las tareas que el resto del sistema realiza sobre la base de datos.

Diagrama de Caso de Uso de Listado de Cursos



La actividad central del control de cursos es el obtener un listado de los cursos que se tienen creados y la información básica de estos de modo que se pueda elegir que operación realizar sobre ellos.

Diagrama de Secuencia de Listado de Cursos

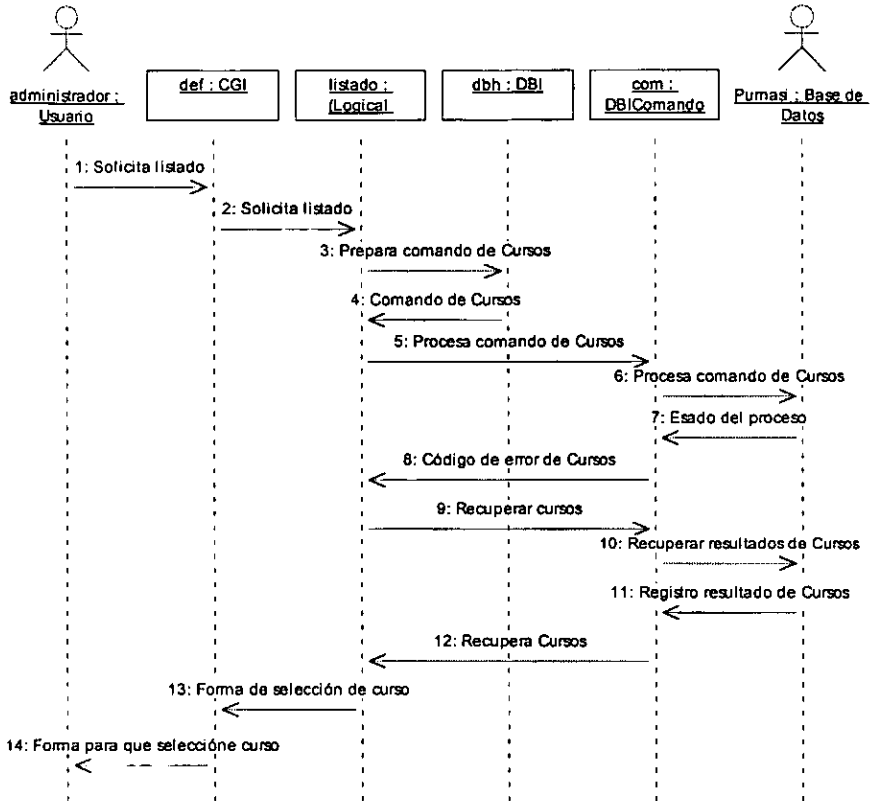


Diagrama de Caso de Uso de Captura de datos de un nuevo curso

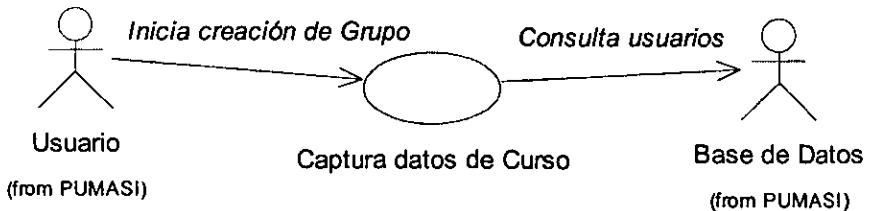


Diagrama de Secuencia de Captura de datos de un nuevo curso

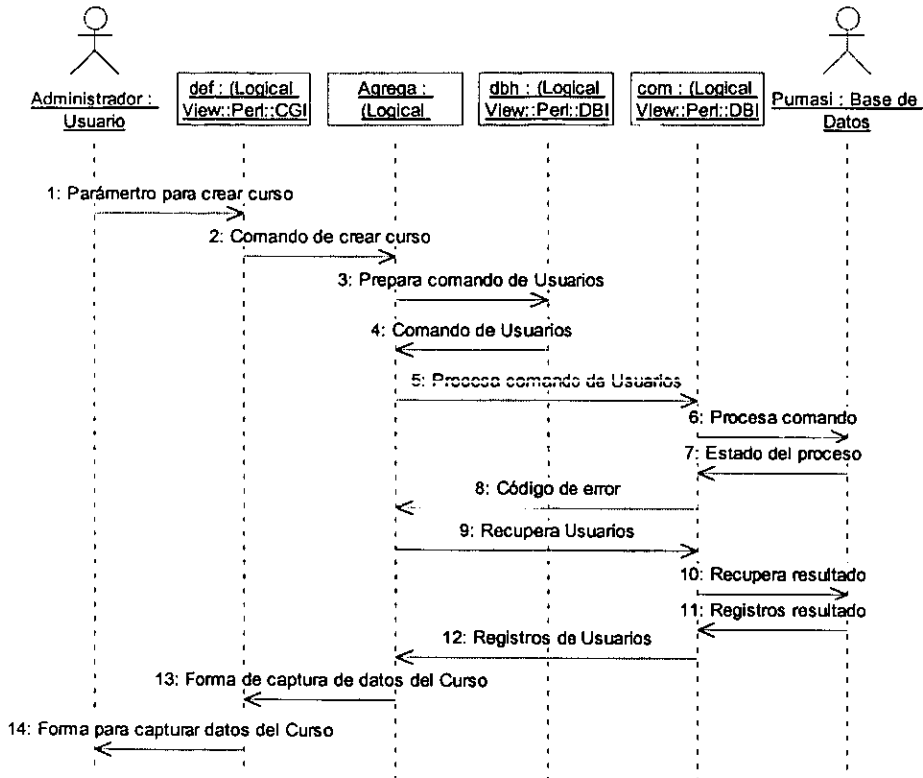
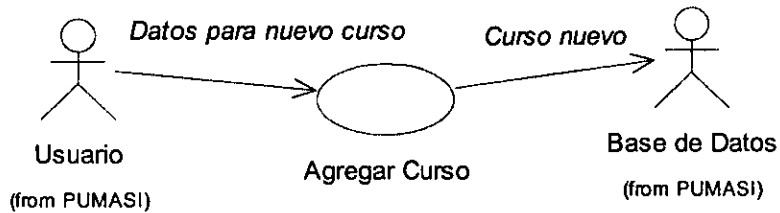
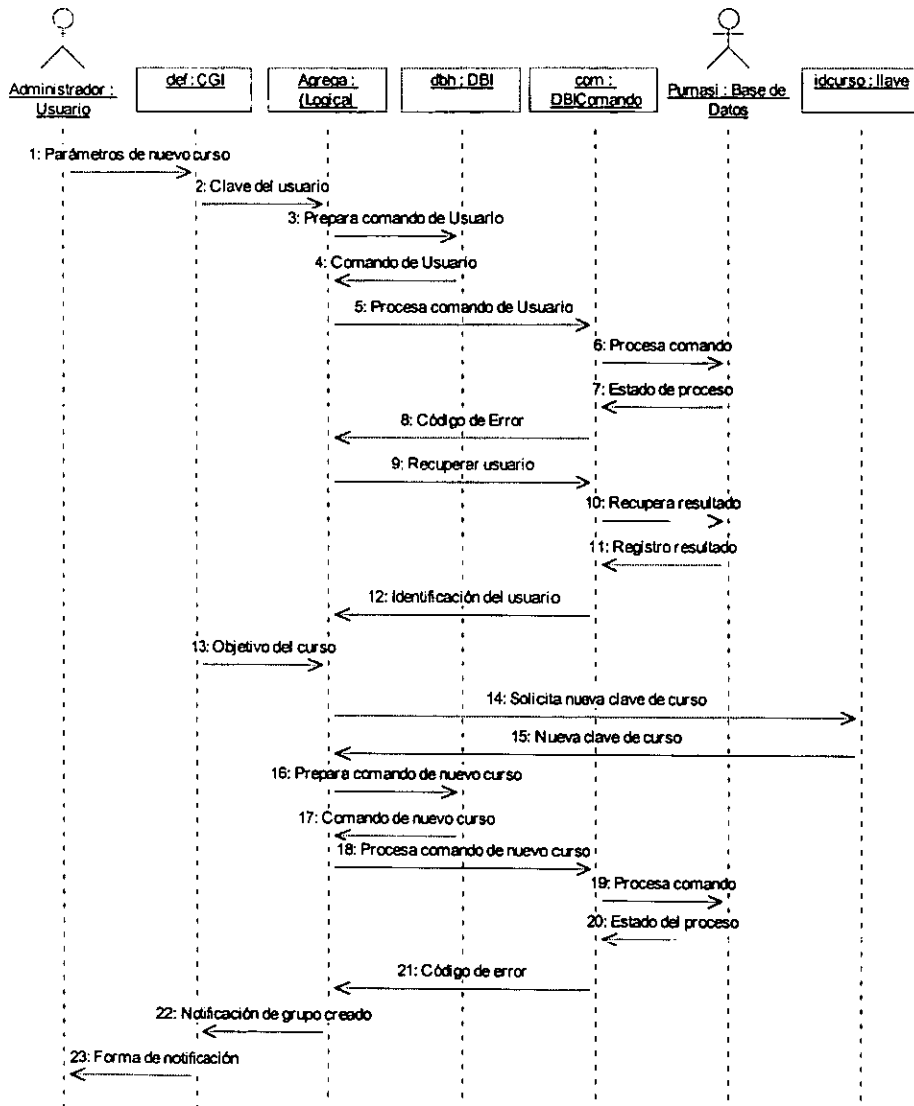


Diagrama de Caso de Uso de Creación de Cursos

La creación de cursos, como otros procesos relacionados con la base de datos, involucran a pocos actores del sistema pero requieren de varias operaciones entre ellos, de modo que el diagrama de Caso de uso puede ser engañosamente sencillo al no reflejar la naturaleza de las relaciones entre los actores.

Diagrama de Secuencia de Creación de Cursos



En este diagrama se puede apreciar las operaciones que se necesitan, la principal razón para un proceso tan largo, es que se requiere verificar la información que se recibe para

crear al nuevo usuario, no debemos olvidar que al recibirse de formas Web, un usuario capacitado podría intentar de modificar los valores a entregar perturbando la operación del sistema, para evitar esto, se limpian los valores y se verifica la validez de los valores recibidos respecto a los privilegios del usuario, evitando así cualquier uso no autorizado.

Diagrama de Caso de Uso de Modificación de un Curso

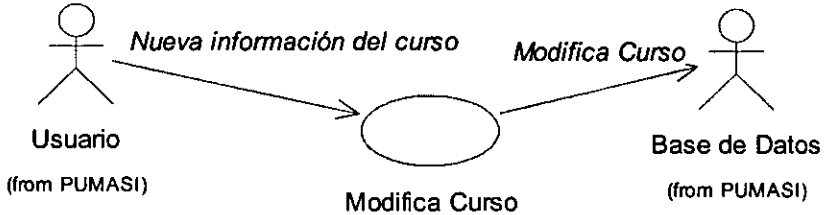


Diagrama de Secuencia de Modificación de un Curso

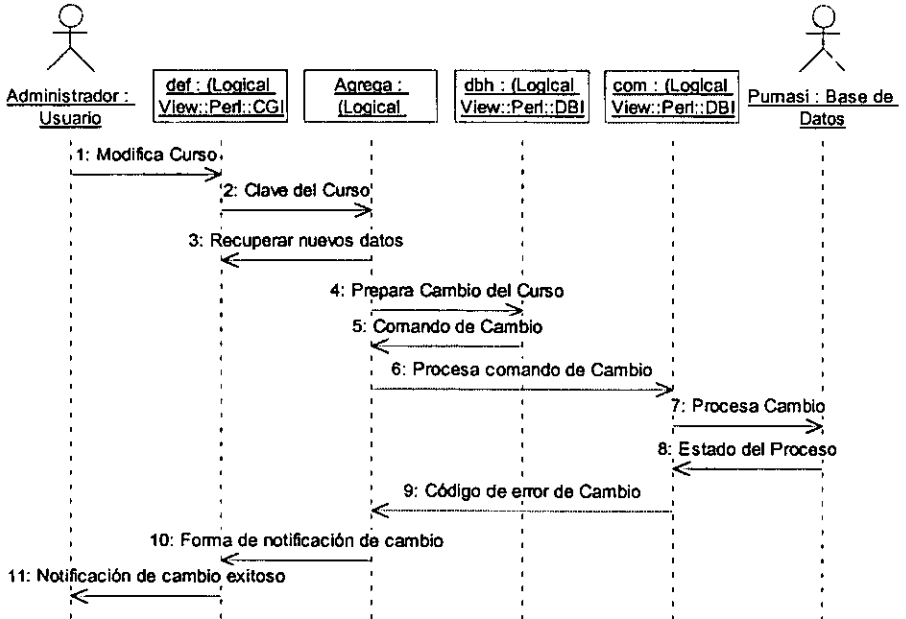


Diagrama de Caso de Uso de Eliminación de un Curso

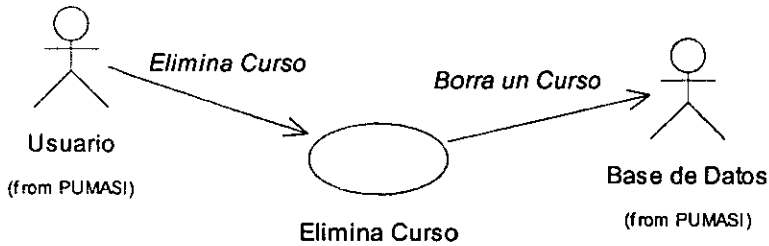
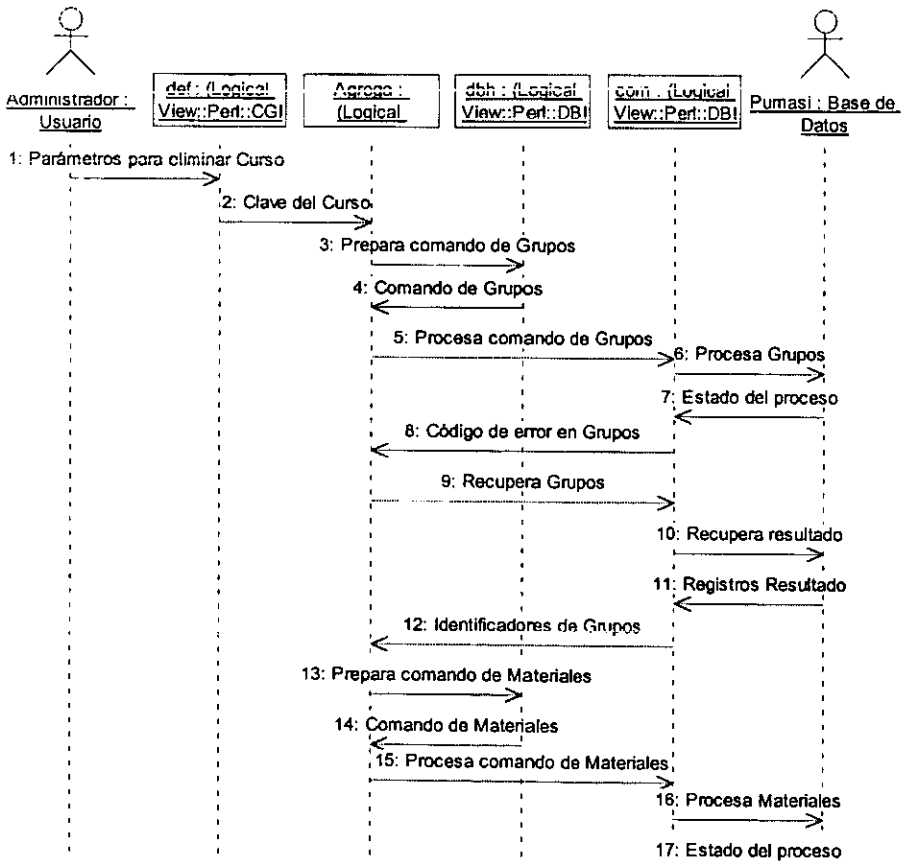


Diagrama de Secuencia de Eliminación de un Curso



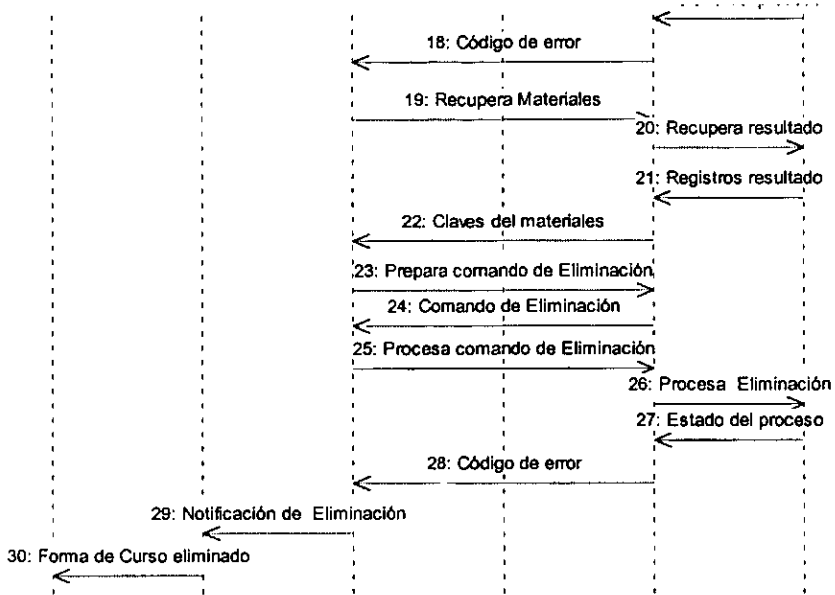


Diagrama de Caso de Uso de Asociar un Responsable

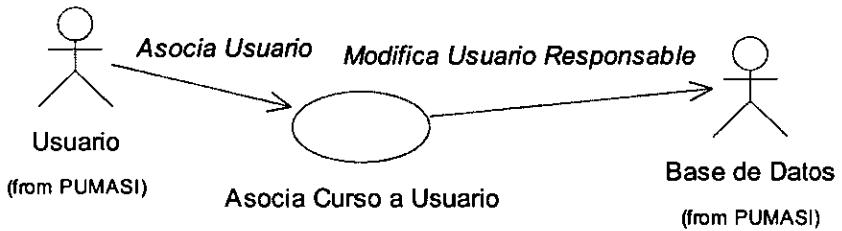
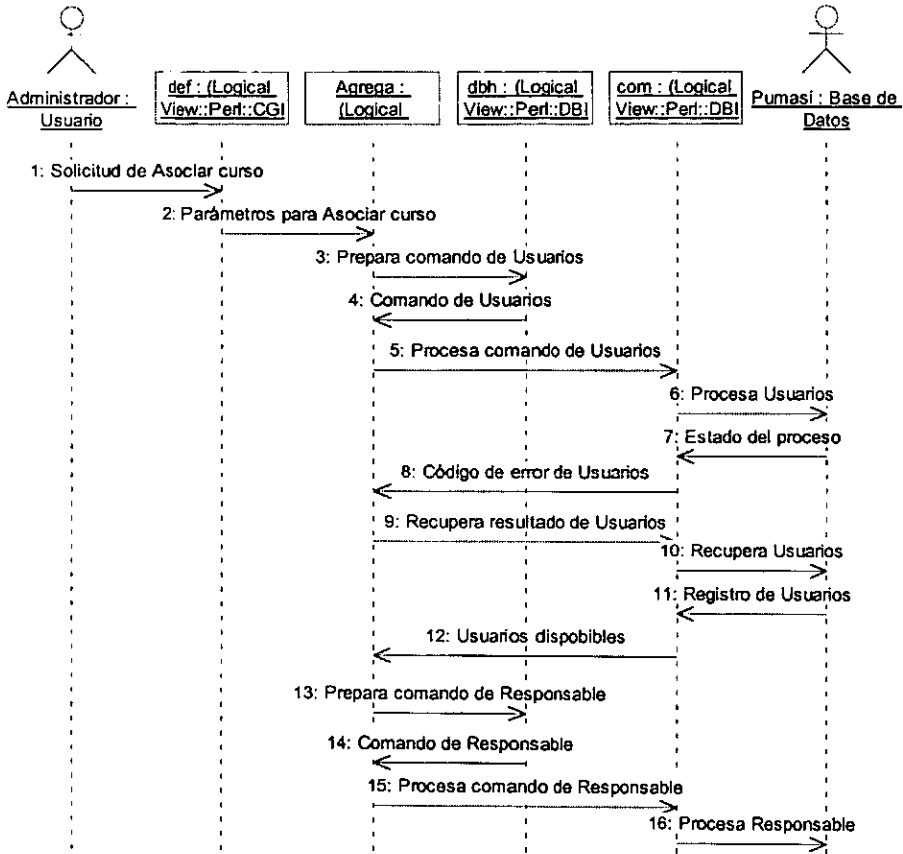
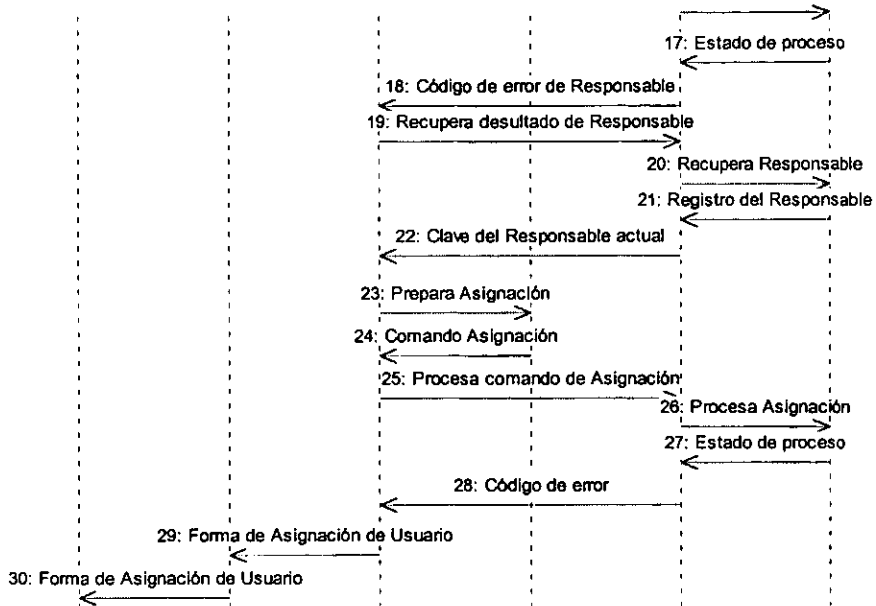


Diagrama de Secuencia de Asociar un Responsable





En el diseño empleando UML se omitieron los diagramas de cambio de estados, debido a que por tratarse de un sistema Web, el usuario puede modificar el estado de las operaciones e incluso interrumpir el flujo planeado del sistema, de modo que los diagramas de cambio de estado debería contemplar la posibilidad de que cualquier estado del sistema fuera seguido por otro cualquiera. Si el usuario proporciona los atributos adecuados a una operación no hay una razón válida para rechazarla ni para distinguirla de una que sigue el flujo planeado y aún en caso de no presentar todos los atributos, debe generarse un estado válido, lo mas cercano a la operación deseada, sin permitir por ello operaciones inválidas.

b) Diagramas de Relación de CGI

Como un complemento al diseño del sistema con la metodología UML, se han desarrollado una serie de diagramas que pretenden ilustrar los elementos que participan en las diversas transacciones que ocurren a lo largo del uso del sistema. Como ya se mencionó, el sistema está compuesto de 13 programas CGI.

En estos diagramas se pretende retratar de forma sencilla los principales elementos que intervienen para determinar el comportamiento de los diversos elementos, recordemos que al tratarse de un sistema basado en Web todas las operaciones se realizan de modo individual y el comportamiento del sistema debe determinarse solo por los parámetros

que el navegador entrega, el CGI que es invocado y la información contenida en la base de datos y materiales generados.

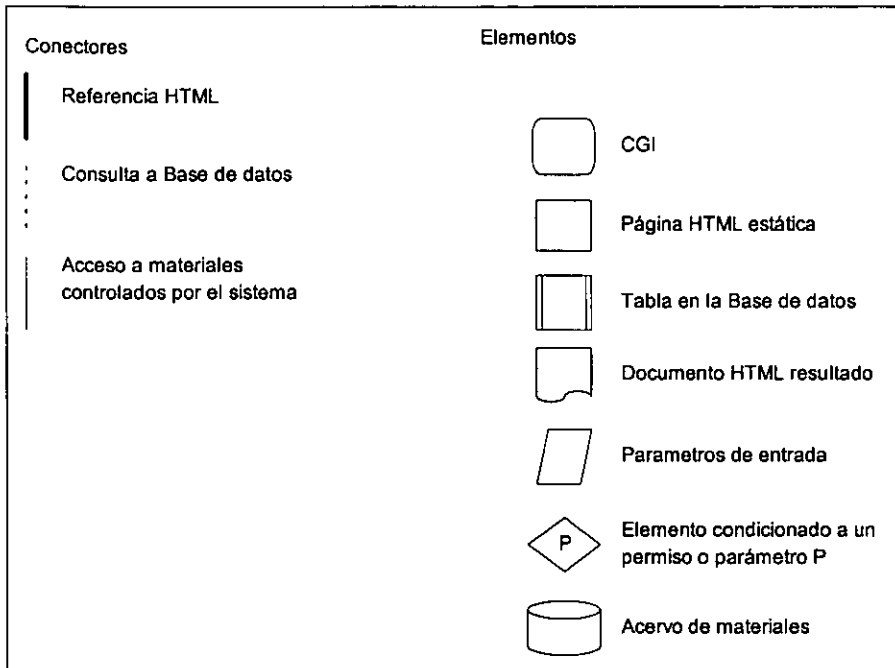
Por lo regular, el comportamiento de un CGI ante un conjunto de parámetros es modificado por la información que se encuentre disponible en la Base de Datos o en los Archivos de apoyo del sistema, por lo que en ocasiones fue necesario indicar a grandes rasgos las condiciones que determinarán el comportamiento de los CGI.

En los diagramas se pone especial énfasis en los parámetros que es necesario entregar a un CGI para que este manifieste un comportamiento dado, el que generará una forma en la que se hará referencia al mismo o a otros CGI del sistema y los parámetros que se emplean en esta nueva forma.

Se ha llamado a estos diagramas "Diagramas de Relación de CGI" porque detallan los parámetros de cada operación y los parámetros del resultado, los que llevan siempre a otro CGI del sistema. El resultado de una operación siempre guiará al usuario a realizar otra operación dentro del sistema mediante ligas al mismo o a otro CGI en las formas resultado.

A continuación se explican los símbolos que se usan en los Diagramas de Relación de CGI que componen esta sección.

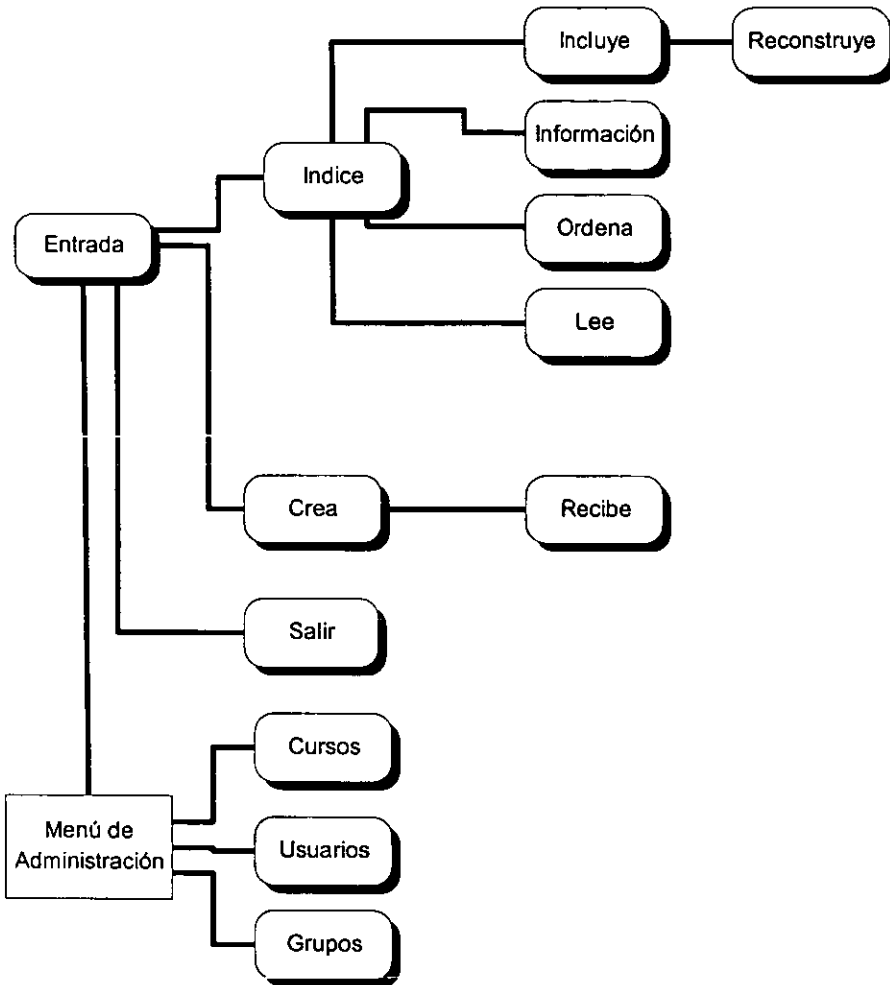
◆ Explicación de símbolos



Debe hacerse notar que todos los CGI implementan un mecanismo de verificación de sesiones en cada operación, de modo que se garantice la validez del usuario, sin embargo, para evitar abultar los diagramas esta operación no se incluye en cada uno de ellos sino que se ha puesto por separado en el diagrama que lleva por título "Relaciones Verses".

En este, el primero de los diagramas se explica a grandes rasgos cual es la relación que estos CGI guardan entre si en materia de referencias. Las relaciones que se ilustran solo pretenden ubicar a cada uno de ellos en el marco de los demás de modo que más adelante sean más fáciles de comprender las funciones que cada uno desempeña.

◆ Modelo del Sistema

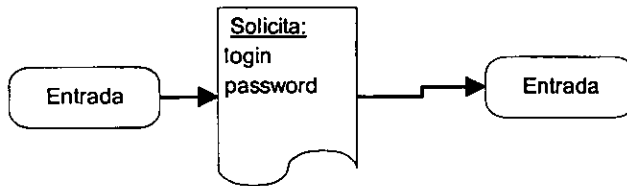


Para cada CGI se tienen diversas posibilidades de comportamiento que van en función de los parámetros entregados, el valor de los parámetros y los resultados que se tenga de las consultas a la base de datos, estos tres elementos constituyen el estado en el que se desarrolla la transacción.

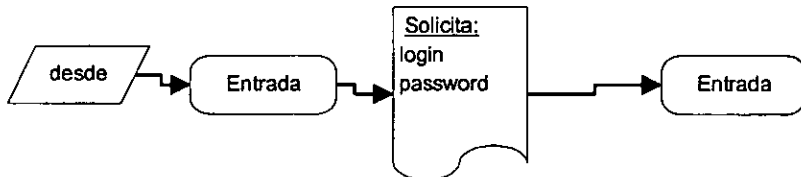
Comenzando por el CGI entrada se tienen diversos esquemas de relación con otros CGI y con él mismo, a cada diagrama se le pone un título que refleja la acción que se está realizando y el CGI sobre el que se desarrolla la acción.

El CGI de entrada es el responsable de generar la galleta que permite al navegador acceder al sistema verificando el login y el password del usuario una sola vez, en caso de que el usuario haya tratado de acceder al sistema desde otro CGI sin una galleta de sesión válida es enviado a este CGI con el parámetro “desde” en el que se indica el CGI a donde trató de acceder originalmente.

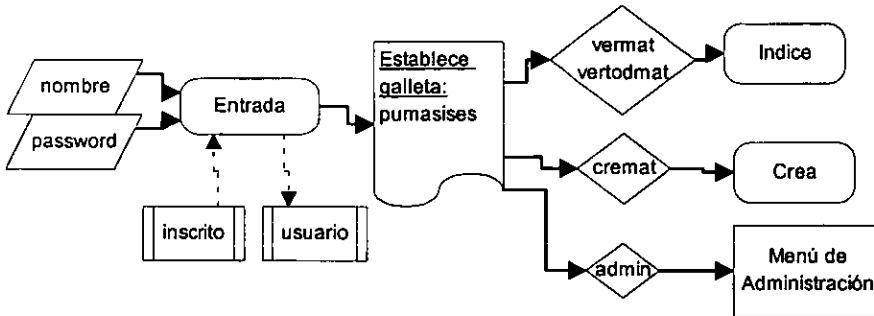
Entrada. Recepción de Usuario



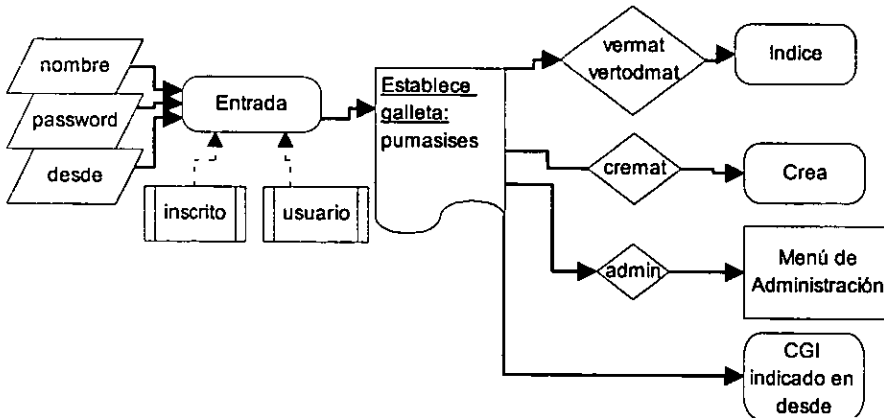
Entrada. Recepción de Usuario conservando punto de Ingreso



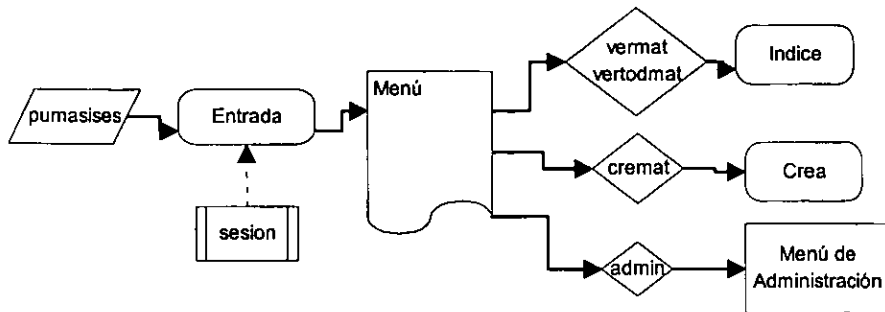
Entrada. Inicio de sesión en el sistema



Entrada. Inicio de sesión en el sistema conservando punto de Ingreso



Entrada. Inicio de sesión en el sistema para administrador

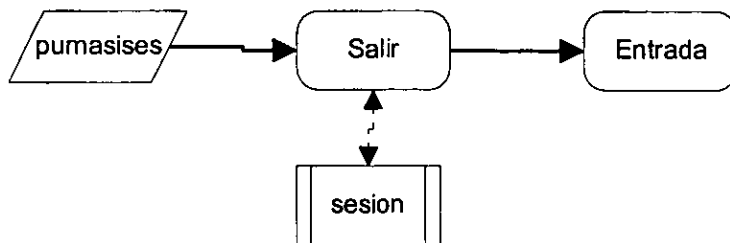


El parámetro que se denota "admin" en este diagrama es en realidad un permiso de entre cualquiera de: revcurso, crecurso, elicurso, asocurso, revgpo, cregpo, eligpo, asogpo, revusr, creusr, eliusr y asousr.

El CGI de salida se encarga de dar por terminadas las sesiones a petición expresa del usuario, por lo regular las sesiones se dan por terminadas cuando termina el tiempo preestablecido de vida de la sesión o cuando el usuario entra al sistema desde otro navegador o cliente.

El parámetro "pumasises" contiene el identificador de una sesión del sistema, funciona como una autorización semi-permanente que permite que el usuario realice varias operaciones sin necesidad de identificarse para cada una de ellas. Este parámetro se implementó con galletas (cookies) con una duración de 6 horas, de modo que cada usuario necesitará identificarse al menos una vez al día pero podrá trabajar por largos periodos de tiempo sin repetir el proceso de entrada.

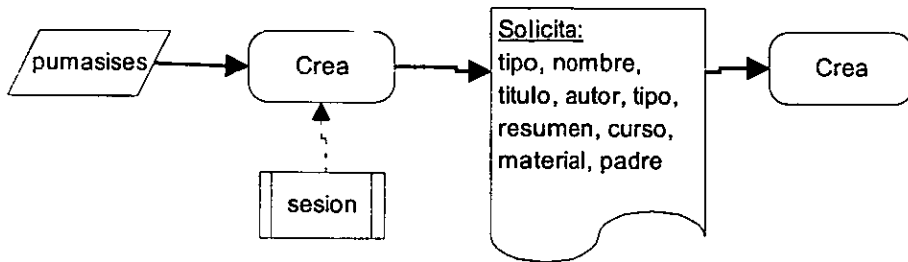
Salida. Termina la sesión



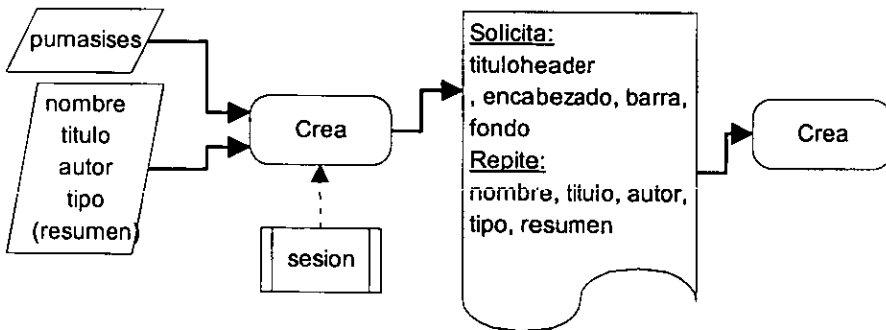
El CGI “crea” genera una serie de formas HTML en las que se entregan los diversos elementos para que se pueda construir un nuevo material en el sistema, de hecho es “recibe” quien crea el material, pero crea permitió hacer el proceso más sencillo al dividir el proceso en varios pasos que dependen del tipo de material que se desee crear, además brinda la posibilidad al usuario de utilizar los materiales gráficos que ya estén dados de alta en el sistema para enriquecer el material que cree y especificar a cual curso pertenecerá y que posición guardará dentro de la jerarquía temática.

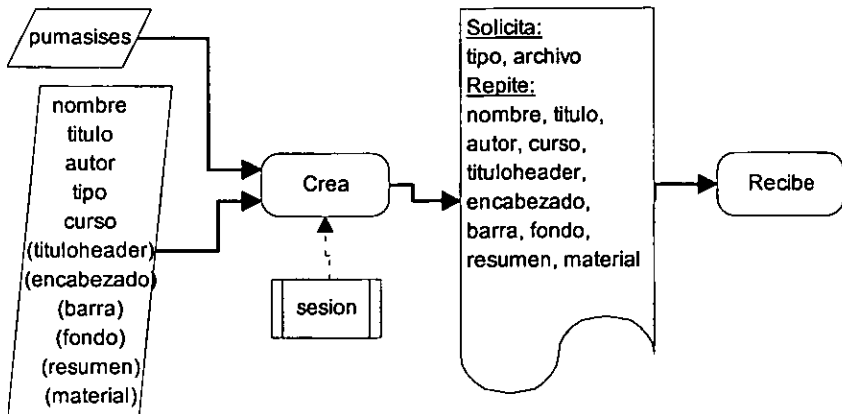
En el sistema se les llama “hoja” a los materiales que se construyen a partir de texto y que serán convertidos a documentos HTML y “recurso” a las gráficas que se usan ya sea como fondo, barra de separación o como ilustración dentro de los materiales “hoja”.

Crea. Información de Catálogo



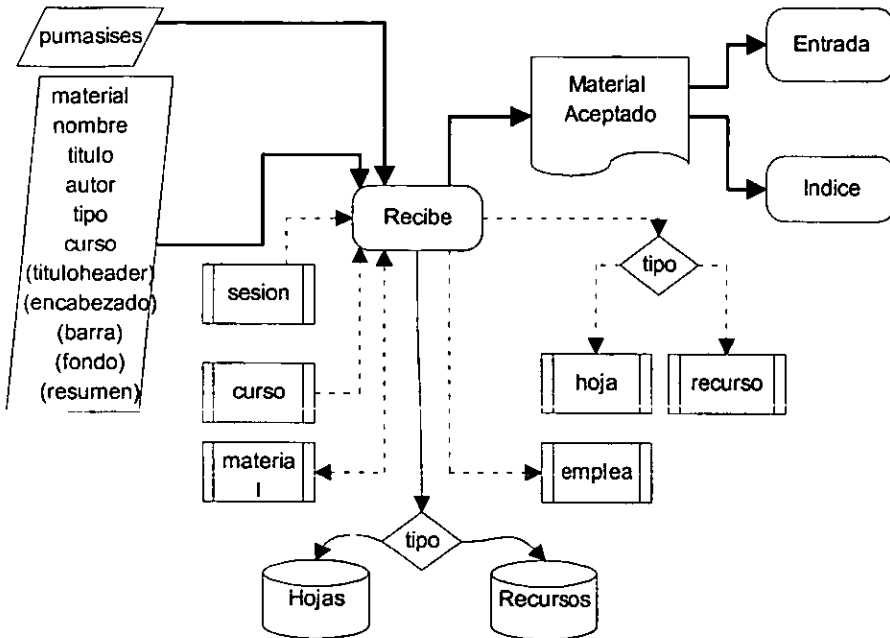
Crea. Datos para un material tipo Hoja



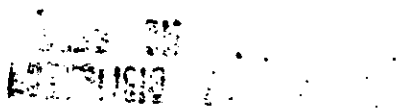
Crea. Petición del Archivo

Recibe es el CGI encargado de crear los materiales y recibir los archivos enviados por el servidor HTTPD, gracias a Crea solo tiene un modo de operación cuando recibe todos los parámetros necesarios según el tipo de material a crear, realiza las operaciones necesarias no solo para dar de alta el registro en la base de datos sino para formar el documento HTML o colocar el recurso entregado en las áreas del servidor HTTPD adecuadas para que puedan usarse por los demás materiales.

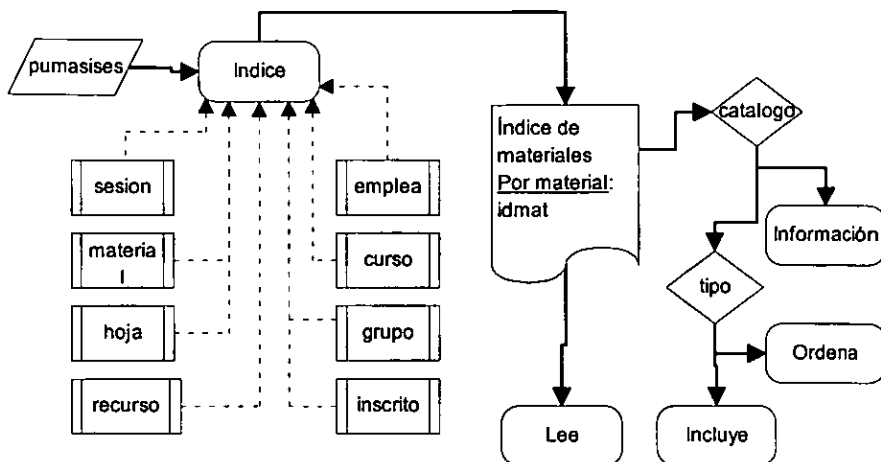
Recibe. Creación de un material



El punto central de la revisión de los materiales del sistema está en el CGI "Indice", es en este que se revisan los catálogos de materiales contra las autorizaciones del usuario para determinar cuales materiales puede ver, generando un listado de estos con las alternativas que el usuario tenga autorizadas, tales como modificar el material o su posición respecto a otros temas.



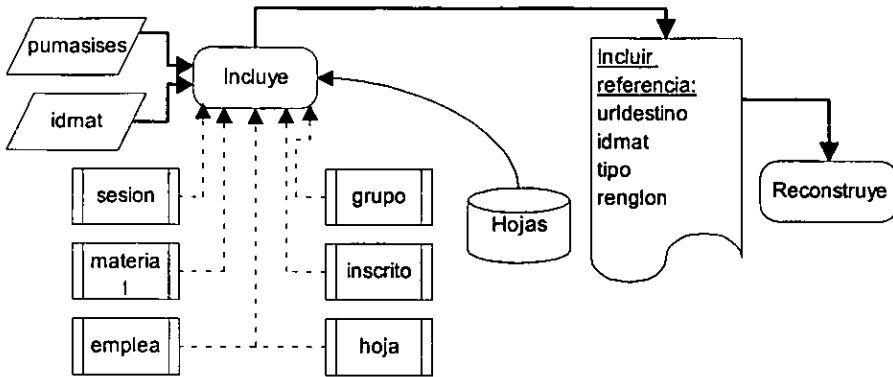
Índice. Listado de materiales



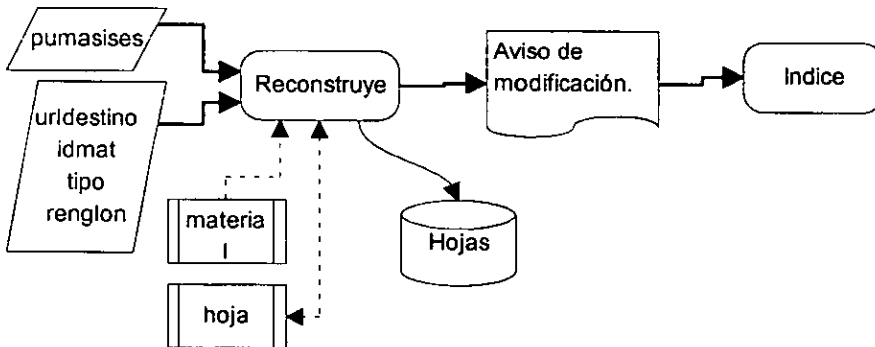
Una de las operaciones que se pueden realizar sobre los materiales tipo Hoja, es el especificar a uno de los renglones del archivo como un elemento activo, ya sea como una referencia a un URL adentro o fuera del sistema o como el punto para insertar una ilustración, el CGI que se encarga de dar al usuario las alternativas de creación de la liga es “incluye” quien entrega la información al CGI “reconstruye” para que genere la nueva hoja.

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

Incluye. Construye una liga

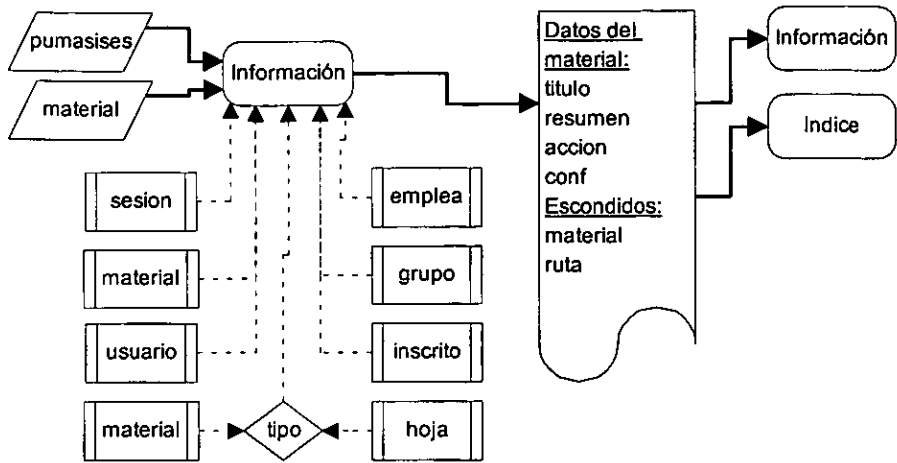


Reconstruye. Establece una Liga

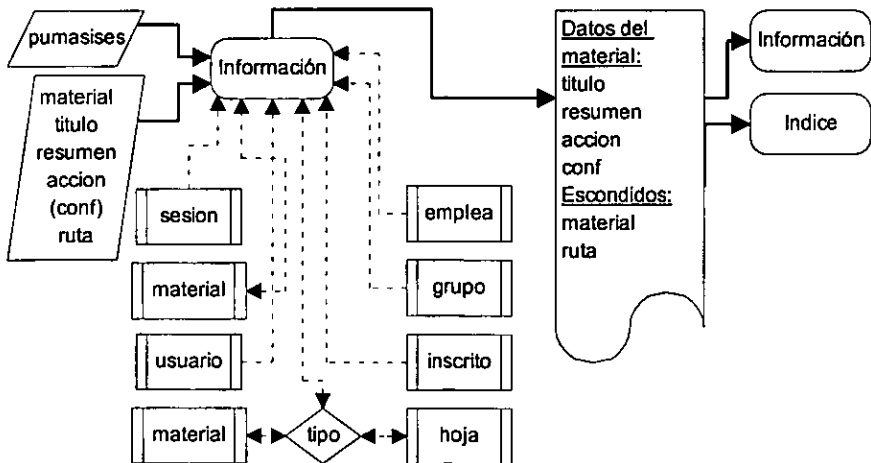


Otra de las alternativas de modificación de los materiales ya existentes es el cambiar la información de catálogo con los que se crearon, como el título o el resumen del material, esta modificación puede hacerse tanto en Hojas como en Recursos ya que todos los materiales del sistema tienen información de catálogo.

Información. Presentación de los datos



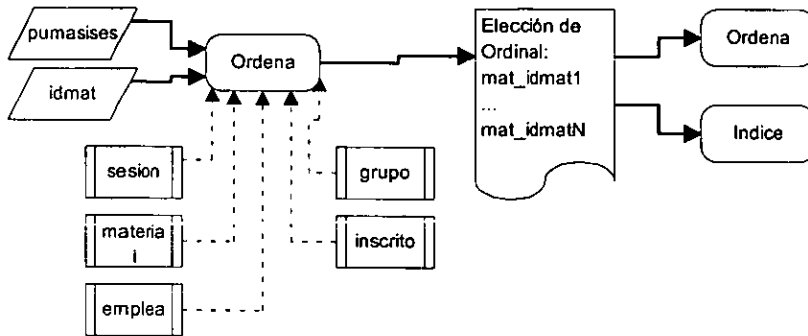
Información. Modificación de los datos



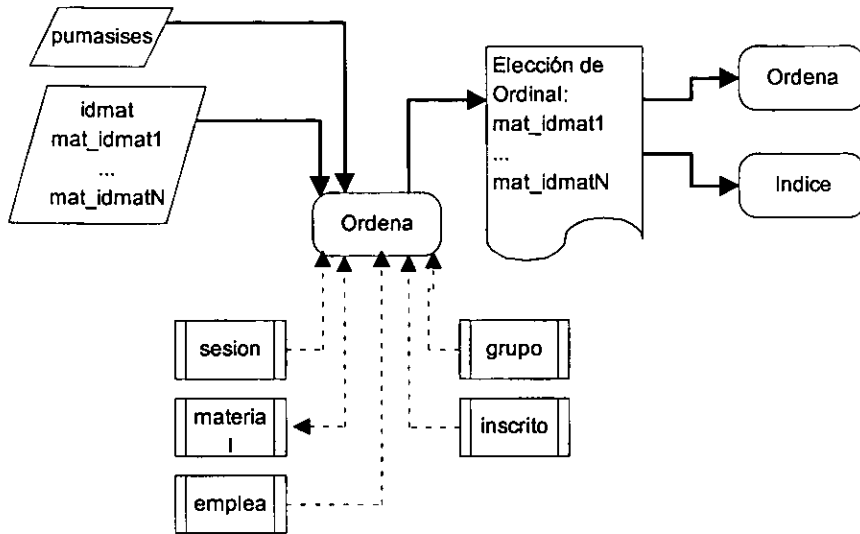
La última de las modificaciones de un material es el cambiar el orden en el que se despliega respecto a los demás materiales que se encuentran en el mismo nivel. Esto se

realiza con el CGI “ordena” que presenta alternativas de números ordinales para todos los materiales del mismo nivel que el material que se quiere modificar. En caso de que se especifique el mismo número ordinal para varios materiales se desplegarán en el orden en que fueron creados. Debe notarse que esto no afecta la relación de pertenencia a sus materiales padre ni a los materiales hijos que tengan.

ordena. *Genera opciones de ordinal*



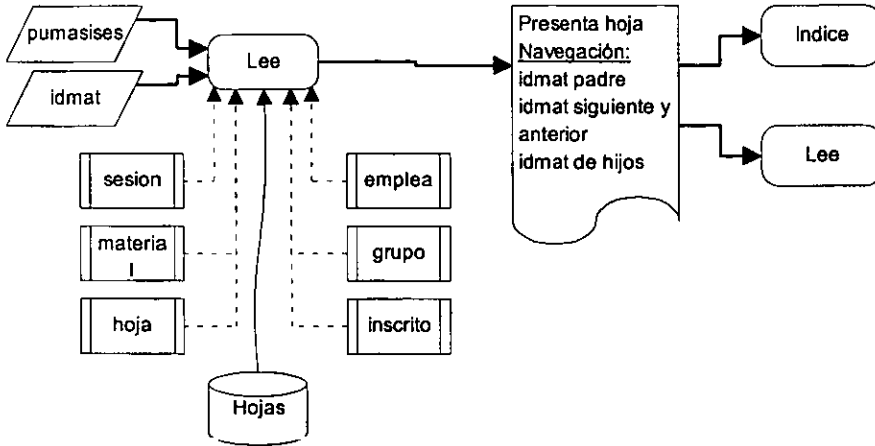
Ordena. Modifica los ordinales de un grupo de materiales



Todos los usuarios del sistema tendrán autorización de leer al menos algunos materiales, para tener un mejor registro de las actividades de los usuarios y para generar opciones de navegación entre los materiales se implementó un CGI que presenta los materiales y revisa la estructura jerárquica de estos para indicar el tema al que pertenece un material, cual es el material siguiente, el anterior y los subtemas que tenga.

El CGI encargado de estas tareas es "lee" y tiene un único modo de operación basado en la identificación del material.

Lee. Presenta material

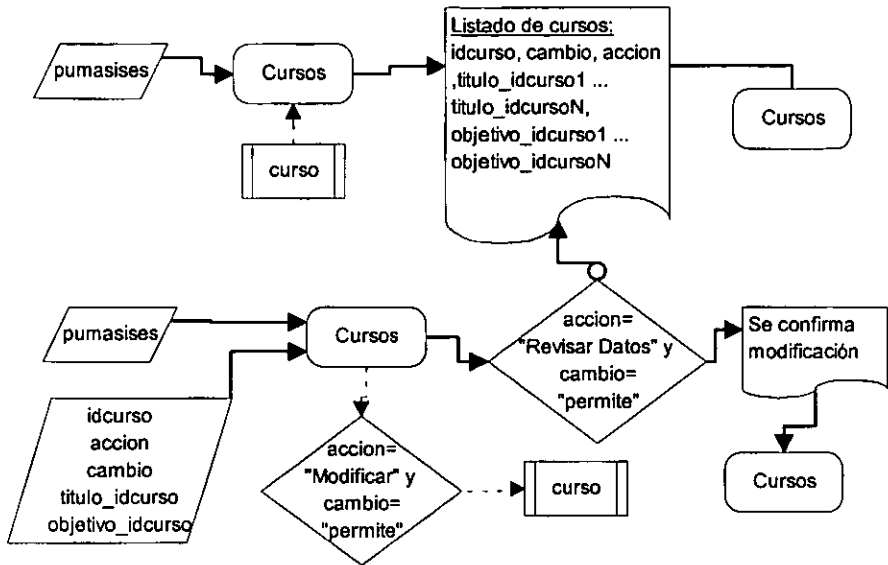


Sección de Administración.

Hasta ahora se han visto los CGI que se encargan del funcionamiento habitual del sistema, sin embargo, una gran parte del desarrollo estuvo enfocado a implementar mecanismos sencillos para que se manejara la información de los usuarios y de acceso a los cursos de modo que el acceso a los materiales pudiera ser regulado docentes sin necesidad de que adquirieran conocimientos técnicos para ello.

La administración del sistema se divide en tres bloques, el control de los cursos, en los que se agrupan los diversos materiales y constituyen la unidad por la que se puede otorgar o negar permiso a un usuario de llegar a los materiales, el control de los usuarios que es implementado en el CGI "usuarios" y otro que maneja los grupos de usuarios a los que se otorga permiso de ver un curso implementado en el CGI "grupos".

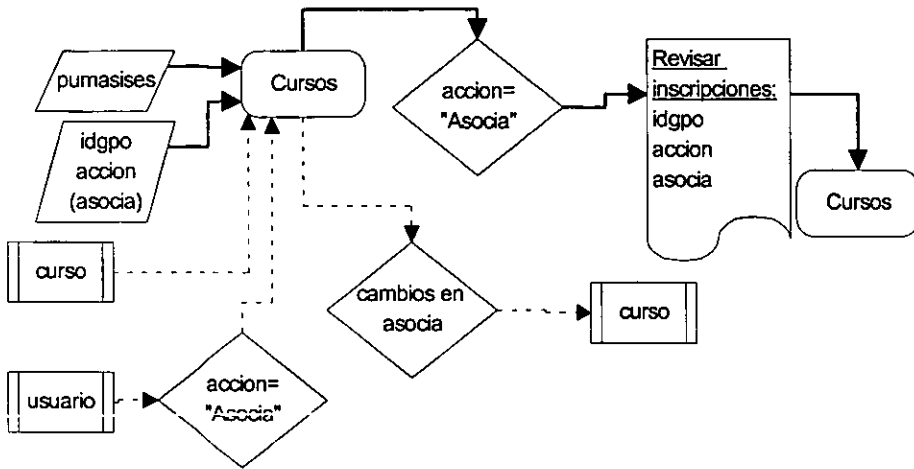
Cursos. Revisión de Información de los Cursos



A diferencia de la mayoría de los diagramas en este se tienen dos operaciones representadas, esto con el fin de ilustrar que en caso de que no se haya activado el parámetro “permite” en la operación, no se modificarán los datos del curso y se presentará de nueva cuenta la forma para verificar los mismos. Para indicar el flujo en caso de que la condición no se cumpla se colocó un círculo en el origen del flujo.

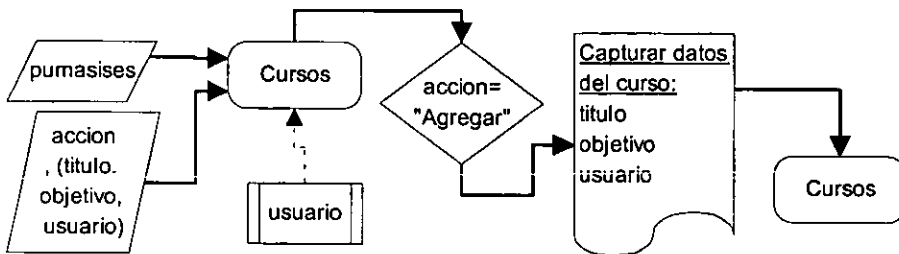
Los materiales se agrupan en los cursos de modo que se tenga un mecanismo rápido de dar acceso a ellos a los usuarios, de modo que los cursos se asocian a los grupos de usuarios para que tengan autorización de ver los materiales. Cada grupo solo tiene asociado un curso, ya que los usuarios y los cursos pueden pertenecer a tantos grupos se requiera.

Curso. Asociar un grupo al curso

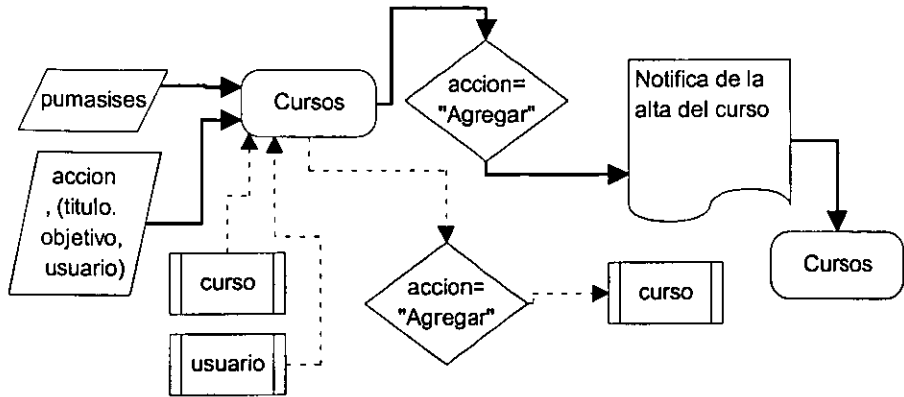


Para crear un nuevo curso, se tiene un proceso de dos pasos, primero se especifica la acción de crear un curso lo que genera la forma para entregar los datos pertinentes y luego se entregan estos datos para generar el nuevo curso.

Curso. Captura Información de nuevo curso

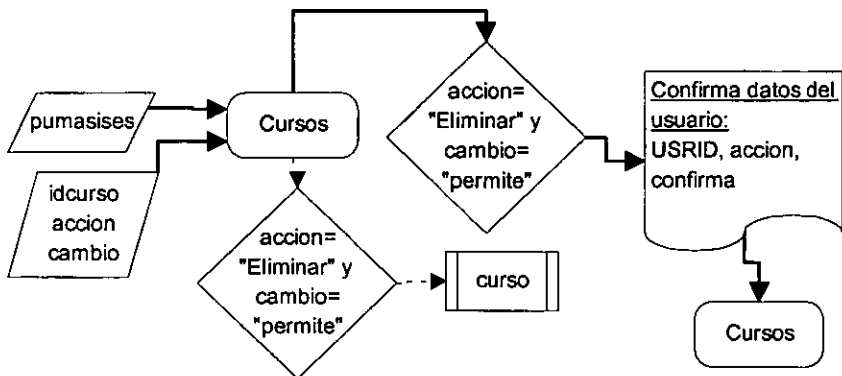


Curso. Crea nuevo curso



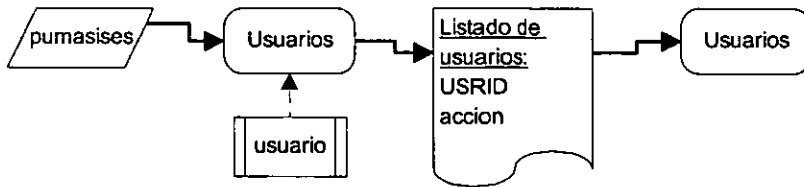
También se tiene la alternativa de eliminar un curso, para lo cual se selecciona en el listado de cursos el que se desea eliminar y se especifica la acción junto con el campo de confirmación de borrado de curso, este campo tiene el propósito de evitar que una selección accidental de acción pudiera eliminar un curso útil.

Curso. Elimina un curso



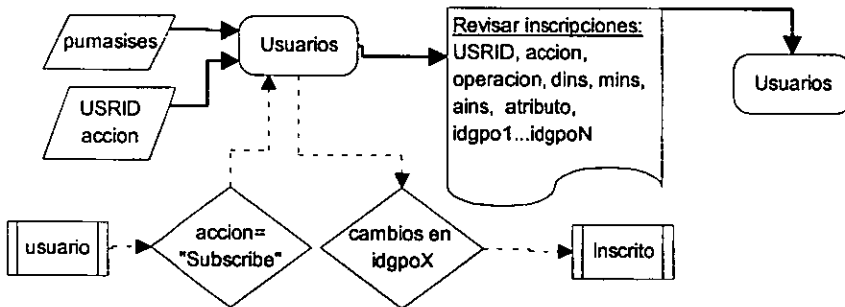
De modo análogo se tiene la administración de usuarios, la acción básica es generar el listado de usuarios con las alternativas de acción.

Usuarios. Listado de usuarios



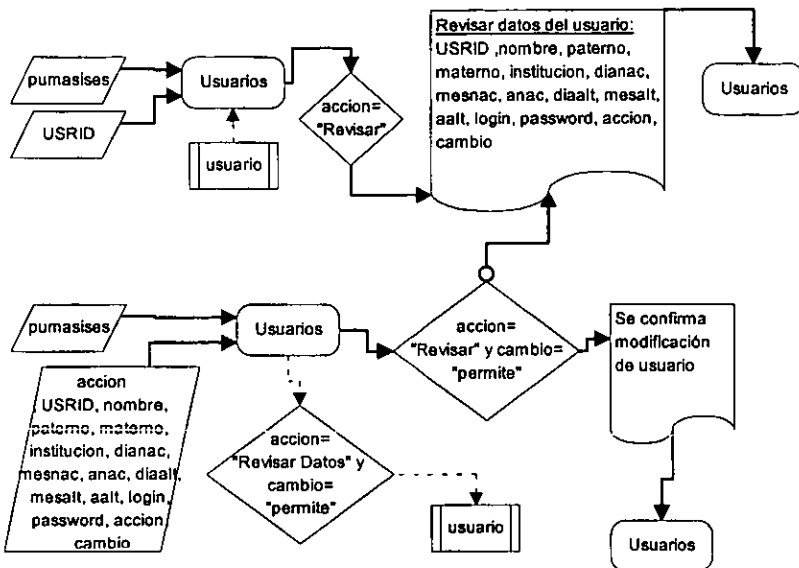
Una de las acciones a realizar es la suscripción del usuario a grupos de modo que tenga acceso a los materiales.

Usuarios. Suscripción a grupos



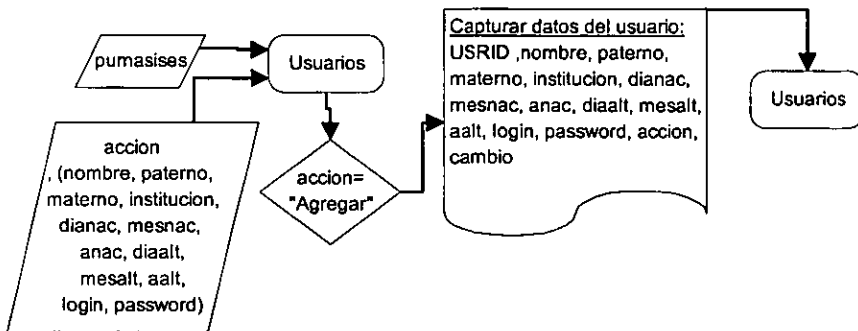
La revisión de un nuevo usuario es de las operaciones que requieren de más parámetros y al igual que el cambio de cursos retorna a la forma de verificación de parámetros si detecta que los campos no están completos o no se verifica que se desea modificar la información.

Usuarios. Cambio de Información del usuario

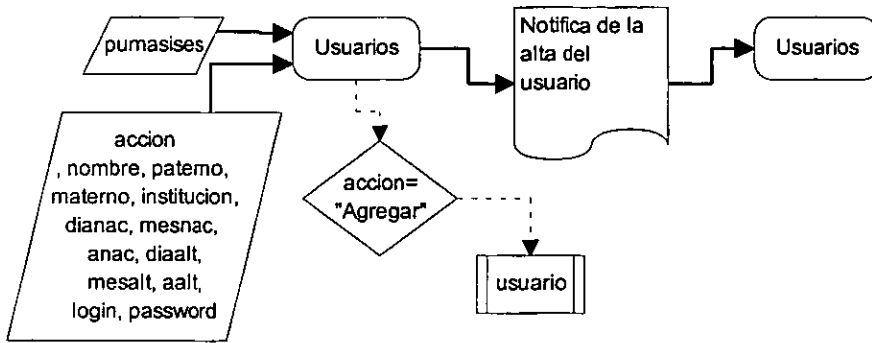


Para agregar usuario se genera una forma en la que se pueden alimentar todos los datos necesarios los que se entregan al mismo CGI de modo que pueda generarlo.

Usuarios. Capturar datos de un nuevo usuario

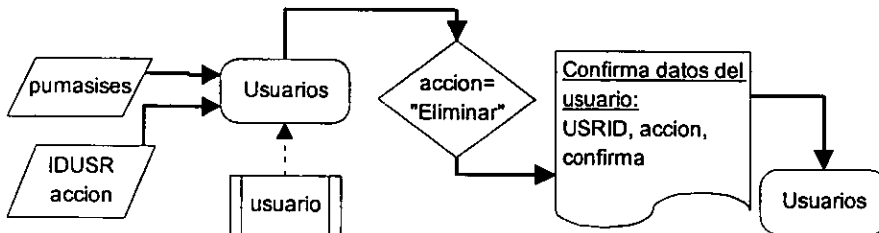


Usuarios. Generar nuevo usuario

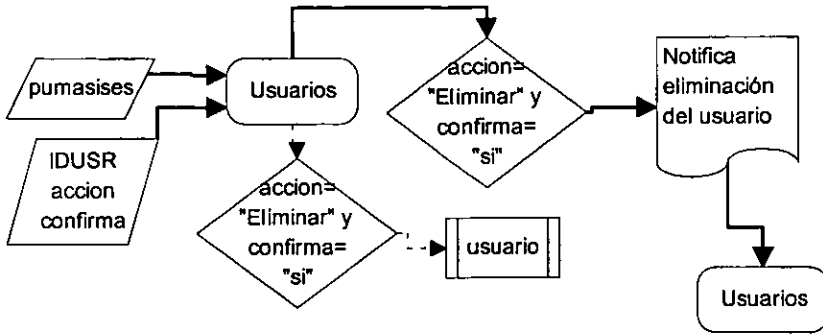


Para eliminar a un usuario también se sigue un proceso de dos pasos de modo que se verifique la información del usuario con el fin de que no haya equivocaciones respecto al usuario a eliminar.

Usuarios. Confirmar datos del usuario a eliminar

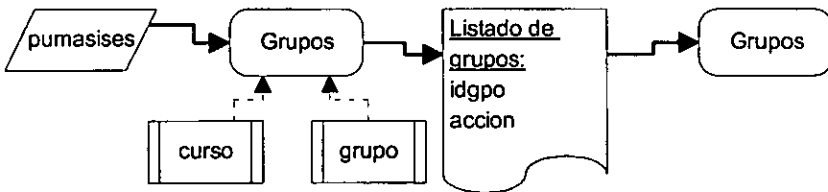


Usuarios. Eliminar usuario



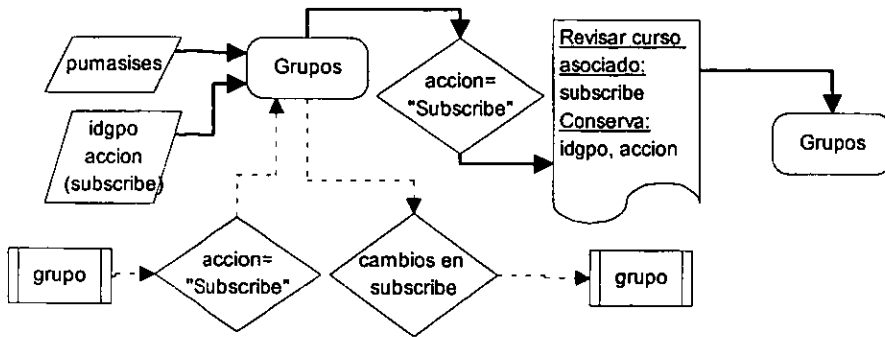
Por último está la interfaz para controlar los grupos, su tarea básica también es generar el listado de los grupos sobre los que se puede trabajar.

Grupos. Listado de Grupos



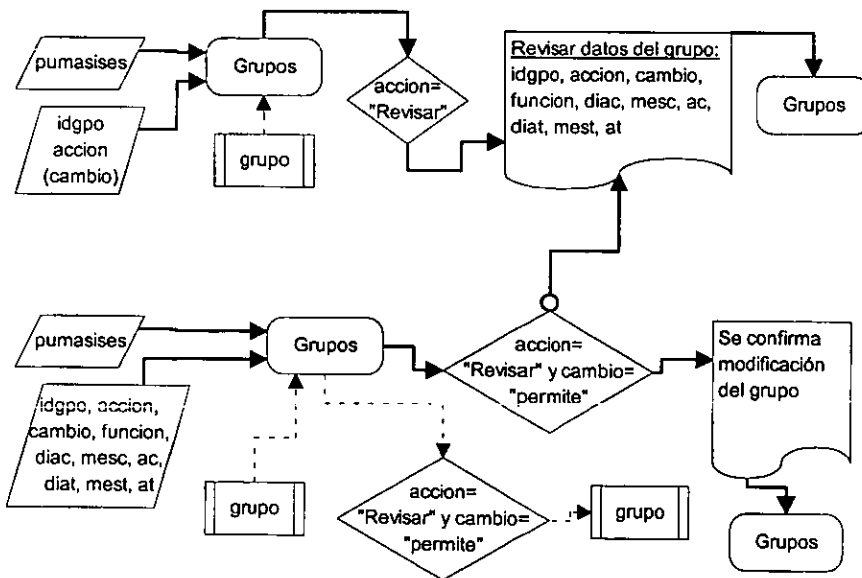
Se puede modificar el curso al que un grupo está asociado de modo que no sea necesario reinscribir a todos los usuarios.

Grupos. *Subscribe a un curso diferente*



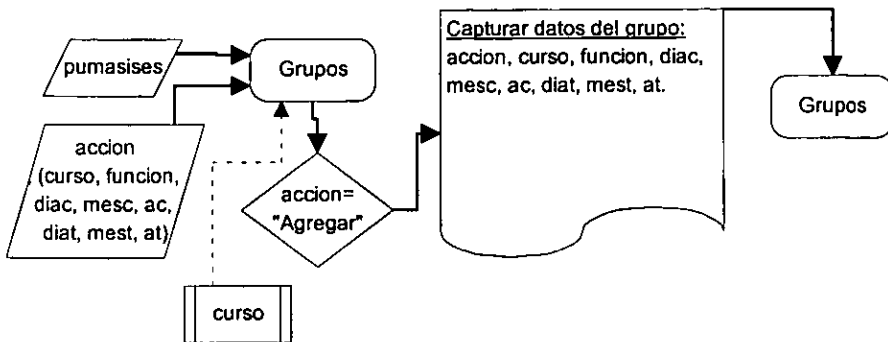
Para modificar la información de un grupo se tiene un proceso de dos pasos.

Grupo. *Modificar grupo*

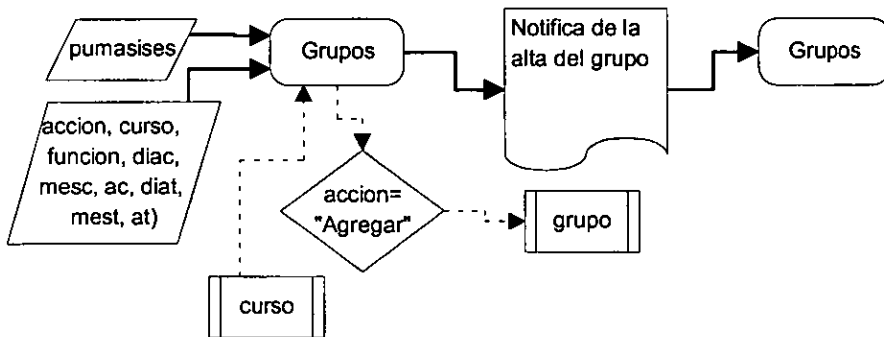


Cuando se indica crear un nuevo grupo primero se capturan los datos de este y una vez que se tienen estos se le crea.

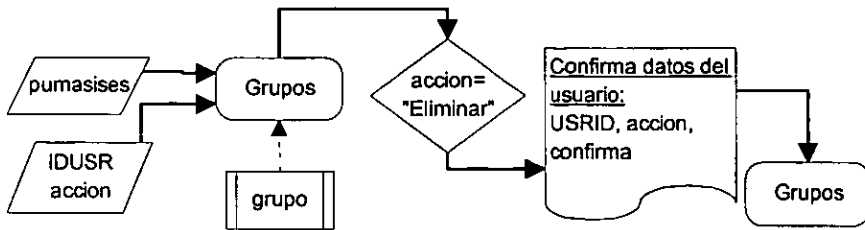
Grupos. Capturar datos de nuevo grupo



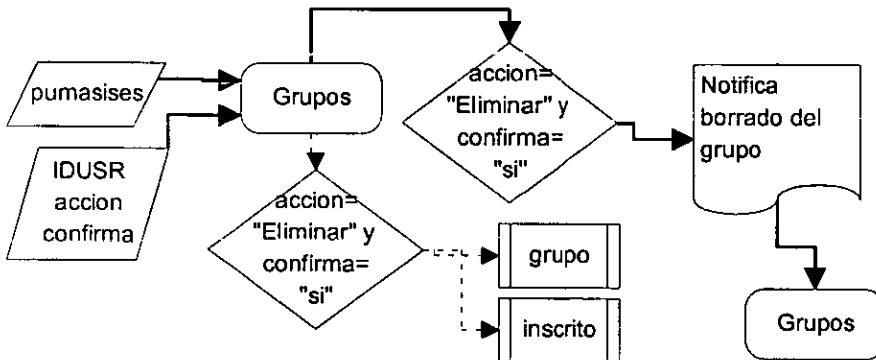
Grupos. Generar nuevo grupo



Grupo. Confirmación para eliminar un grupo



Grupo. Eliminar un grupo



3. Apariencia del Sistema

Los CGI presentan sus resultados como formas HTML que toman de archivos externos a ellos, este modo de implementación nos da la ventaja de poder modificar la apariencia del sistema modificando tan solo archivos HTML lo que puede hacerse por personal de diseño gráfico sin conocimientos de programación y con herramientas especializadas para después reemplazar los archivos usados originalmente logrando el cambio en la apariencia del sistema sin ningún impacto sobre su código o funcionalidad.

Para establecer la imagen actual de las formas del sistema se contó con el apoyo del personal de Diseño de la Coordinación de Prospección de la DGSCA, en especial el de la Lic. Itzel Hernández Serra.

A continuación se incluyen ver algunas pantallas tal como las ve un usuario que emplee Netscape Navigator como cliente del sistema.

Entrada del sistema

Forma de entrada al sistema PUMA SI - Netscape

File Edit View Go Communicator Help

Back Reload Home Search Netscape Print Security

Location: <http://tron.dgpa.unam.mx/cgi-bin/pumasi/entrada.cgi>

Instant Message WebMail Contact People Yellow Pages Download Find Sites Channels

Bienvenido al Sistema de publicación masiva en Internet

PUMASI

En este sistema se han integrado diversas herramientas para permitir al personal docente la creación de materiales de apoyo en Internet.

Con el fin de poder colaborar al mejor aprovechamiento de los recursos, se identifica a cada usuario, para ello, se requiere de un identificador personal y una contraseña que se debe solicitar a la [administración](#).

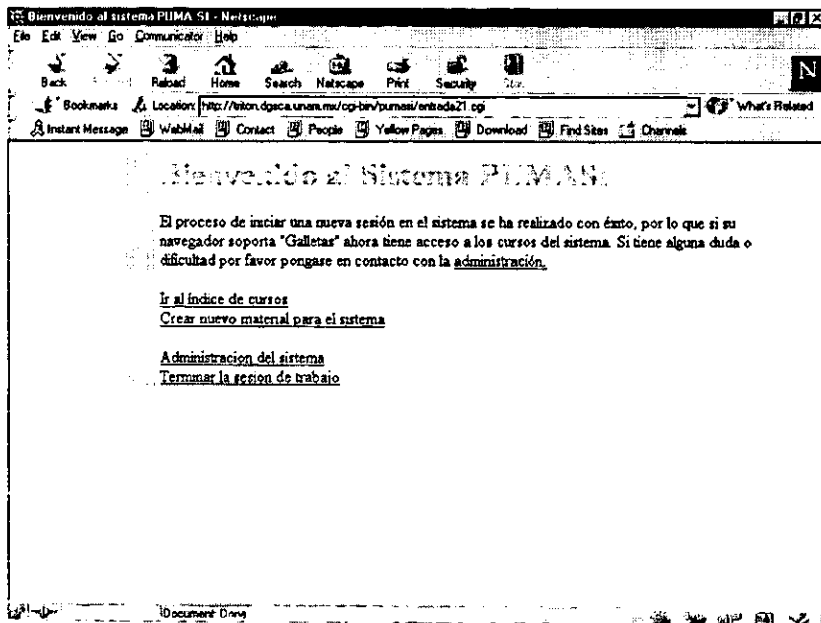
Identificador de usuario

Contraseña

Oprima el botón "Adelante" para entrar al sistema con el identificador y la contraseña que proporcione o "Reinicio" para limpiar los valores. No olvide activar las "Cookies" en su navegador, o no podrá ingresar al sistema.

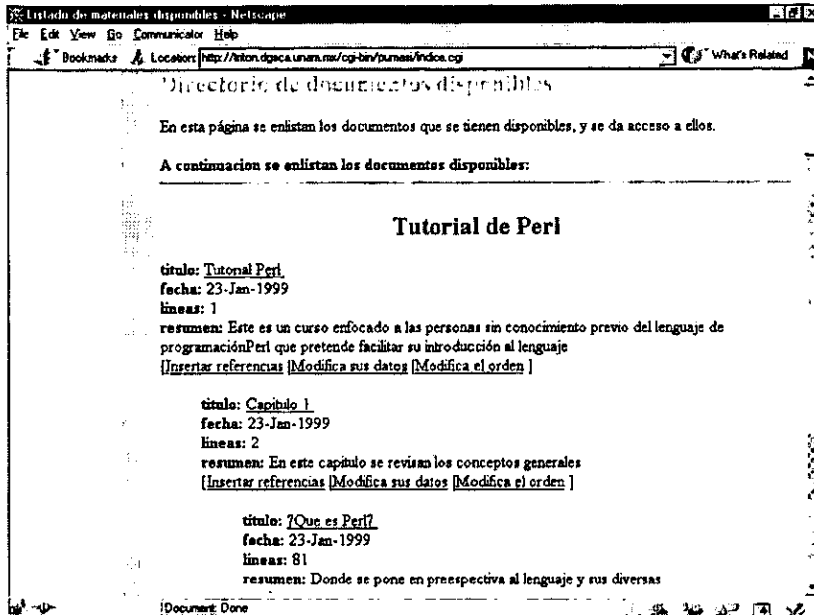
Document Done

Esta es la forma que se presenta al usuario siempre que no presente una sesión válida y le permite dar su información de identificación de modo que se le entregue la galleta de sesión y pueda continuar usando el sistema, de hecho, cualquier CGI es capaz de generar esta forma si detecta que el usuario no entrega una galleta válida.

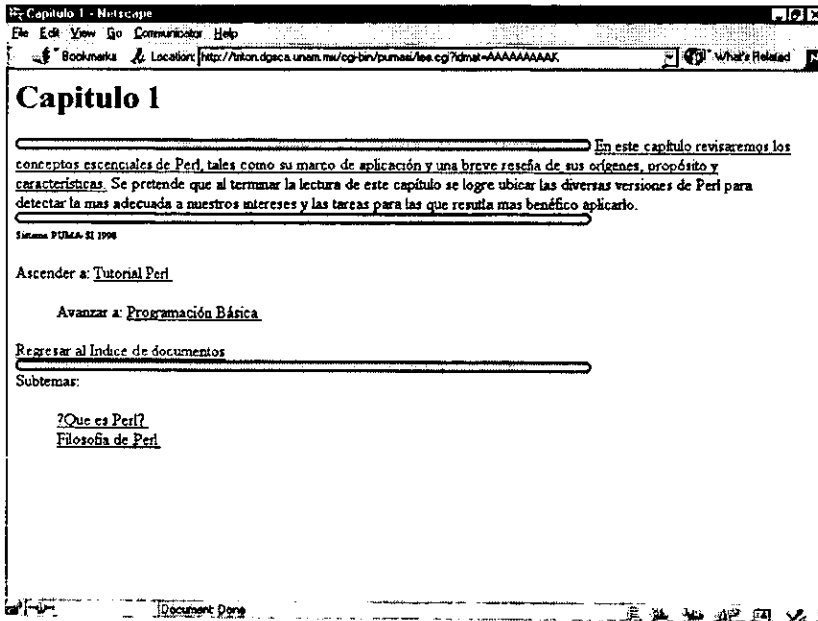
Bienvenida al usuario

Una vez que se realiza el proceso de entrada al sistema, se presentan al usuario las alternativas principales dependiendo de los permisos con que cuente, en este caso se tienen todos los permisos por lo que vemos las alternativas completas:

- Ver el Índice de documentos
- Crear un material nuevo
- Ir al área de administración
- Terminar la sesión actual

Índice de Documentos

En el índice de documentos se presenta un listado de todos los materiales disponibles para el usuario indicando sus relaciones de dependencia a un curso y entre ellos y presentando parte de la información de catálogo, de este unto el usuario elige cual material desea ver y puede regresar para elegir otro o continuar revisando los materiales basado en las ligas en el propio material.

Presentación de un material

Los elementos gráficos que se incluyen en todo material son el fondo, las barras de separación del título y el contenido, el título de la ventana y el título de la página.

Debajo del material hay ligas a los materiales más cercanos al que se presenta; en este caso se puede ver el acceso al "Tutorial Perl" del que depende el "Capítulo 1" que estamos viendo, que el siguiente tema del capítulo es "Programación Básica" y el tema que vemos tiene dos subtemas "¿Qué es Perl?" y "Filosofía de Perl".

Adicionalmente se coloca una liga para que se pueda regresar al índice en cualquier momento.

Creación de un material

Creación de documentos PUMASI - Netscape

File Edit View Go Communicator Help

Bookmarks Location: <http://titon.dgpa.unam.mx/og-brv/pumasi/crea.cgi> What's Related

Formulario de creación de documentos

Datos generales del recurso

Tipo de recurso a crear:

Texto plano Gráfica o recurso binario

Ficha de resumen:

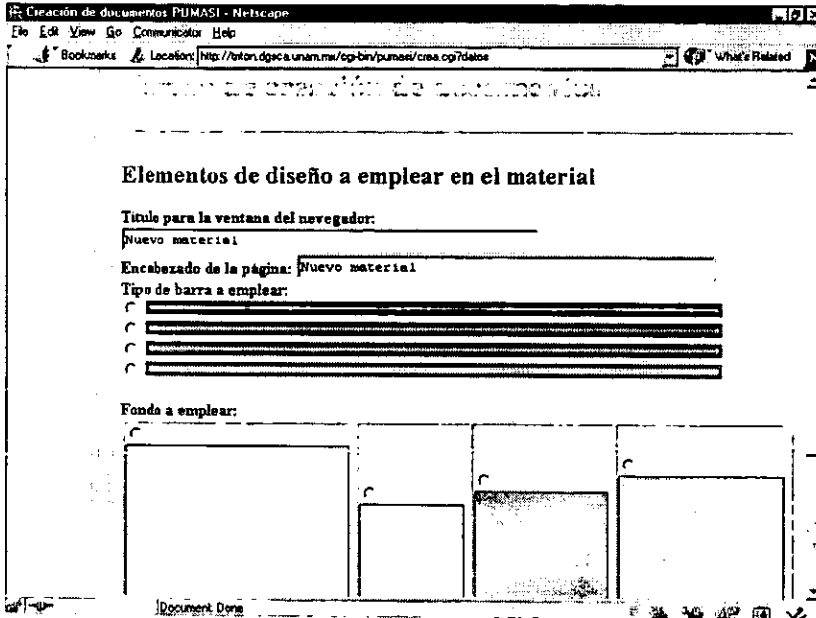
Título:

Autor:

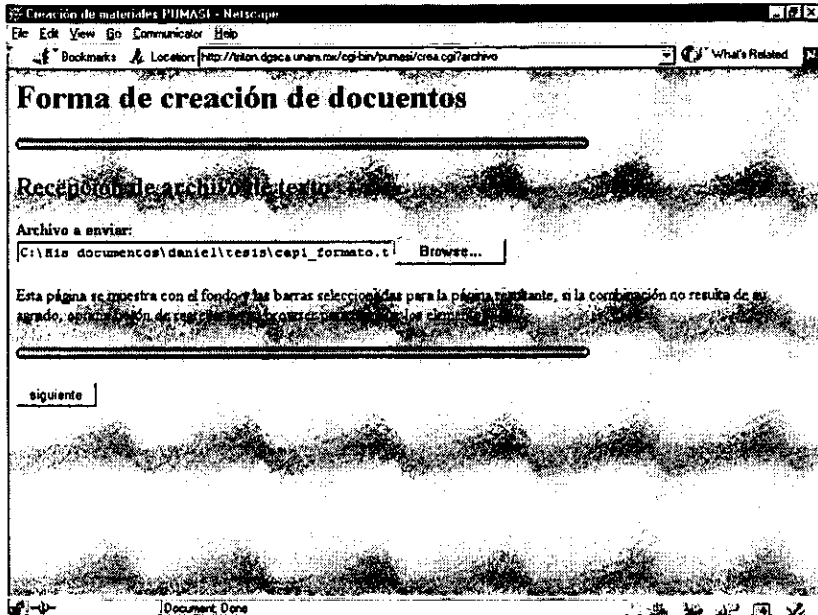
Resumen o descripción:

Document Done

Para crear un material se pasa por una serie de alternativas que inician con la entrega de la información de catálogo del material, es también en esta forma en que se especifica el tipo de material a crear, ya sea una "hoja" o un "recurso" y se especifica el material del que se esté generando un subtema.

Diseño de una nueva "Hoja"

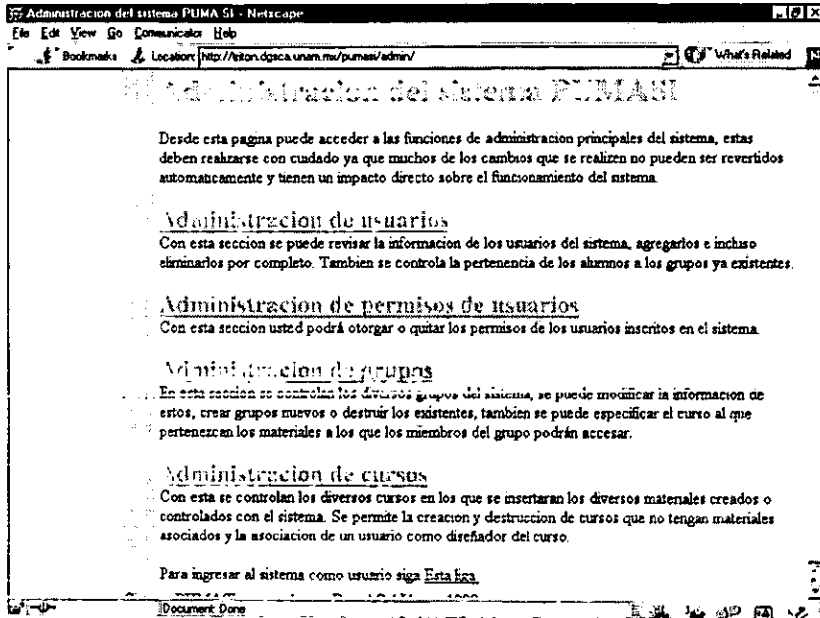
A continuación, se indican cuales elementos se deben emplear en la construcción de la nueva hoja, se elige de entre una colección especial de materiales tipo "recurso" que hayan sido indicados como separadores o como fondos para la creación de nuevos documentos de modo que cada colaborador pueda definir su propio estilo.

Entrega del material a construir

En base a las elecciones de elementos se presenta al usuario una forma en el estilo elegido, para que verifique que sea de su agrado, en el que se le solicita especificar cual es el archivo de texto de su máquina cliente que deberá ser transmitido al servidor y procesado para construir el nuevo material.

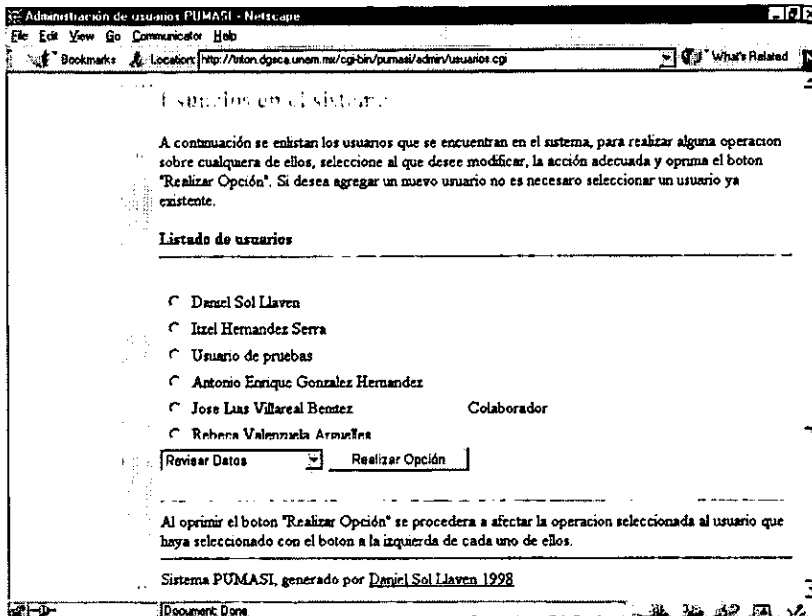
Una vez que se entrega el archivo el sistema genera una nueva hoja y la agrega al catálogo de materiales con lo que aparecerá en el Índice de documentos y será accesible desde sus materiales cercanos en la estructura de subtemas.

Menú de administración



Esta página dá al usuario las alternativas básicas de administración, el control de usuarios, grupos y cursos en los que se determina principalmente la creación, eliminación, modificación de datos y la asociación de estos y una alternativa mas que es el control de permisos de un usuario.

Aún cuando la administración de cada uno de ellos es diferente, se considera que la mas representativa es la de usuarios y la de permisos de estos.

Administración de usuarios

Se presenta un listado de los usuarios del sistema para que se elija a aquel que se vaya a modificar y las operaciones que se pueden realizar, que son:

- Revisar la información de un usuario
- Crear un nuevo usuario
- Eliminar a un usuario
- Inscribir al usuario a un grupo

Creación de un nuevo usuario

Administración de usuarios (RUMASI) - Netscape

File Edit View Go Communicator Help

Location: <http://172.16.1.100/cgi-bin/rumasi/admin/usuarios.cgi>

Información del usuario

En esta forma puede usted revisar todos los datos del usuario, en caso de que desee modificar alguno, cambie el valor que se encuentra en la forma, seleccione el control de "Permite cambios en la información" y oprima el botón "Realiza los cambios". Si no desea realizar ningún cambio, no active la opción de permitir cambios y oprima el botón de regresar en su navegador.

Información del usuario

Nombre, Apellido Paterno y Materno

Fecha de Nacimiento (dd/mm/aa)

Institucion de procedencia

Categoria de usuario

Fecha de alta al sistema

Login al sistema

Password de entrada

Al oprimir el boton "Realizar los cambios", se procedera a cambiar la informacion del usuario siempre y cuando se haya seleccionado la opcion de permitir cambios y oprima el boton de regresar en su navegador.

Se entregan los datos generales del usuario a crear, incluyendo el login y el password que va a utilizar, una vez que se termine se acciona el botón de "Agregar al usuario" y se obtiene una confirmación de que el usuario se haya creado con éxito para de ahí regresar al listado de usuarios.

Modificación de un usuario

Administración de usuarios PUMASI

File Edit View Go Communicator Help

Location: http://biton.dgsc.a.unon.mx/cgi-bin/pumasi/admin/usuarios.cgi

Información del usuario

En esta forma puede usted revisar todos los datos del usuario, en caso de que desee modificar alguno, cambie el valor que se encuentra en la forma, seleccione el control de "Permite cambios en la información" y oprima el botón "Realiza los cambios". Si no desea realizar ningún cambio, no active la opción de permitir cambios y oprima el botón de regresar en su navegador.

Información del usuario

Nombre, Apellido Paterno y Materno: Daniel Sol Llavén

Fecha de Nacimiento (dd/mm/aa): 03/1/19

Institución de procedencia: DGSCA

Categoría de usuario: Administrador

Fecha de alta al sistema: 16/2/19

Login al sistema: lacertus

Password de entrada: *****

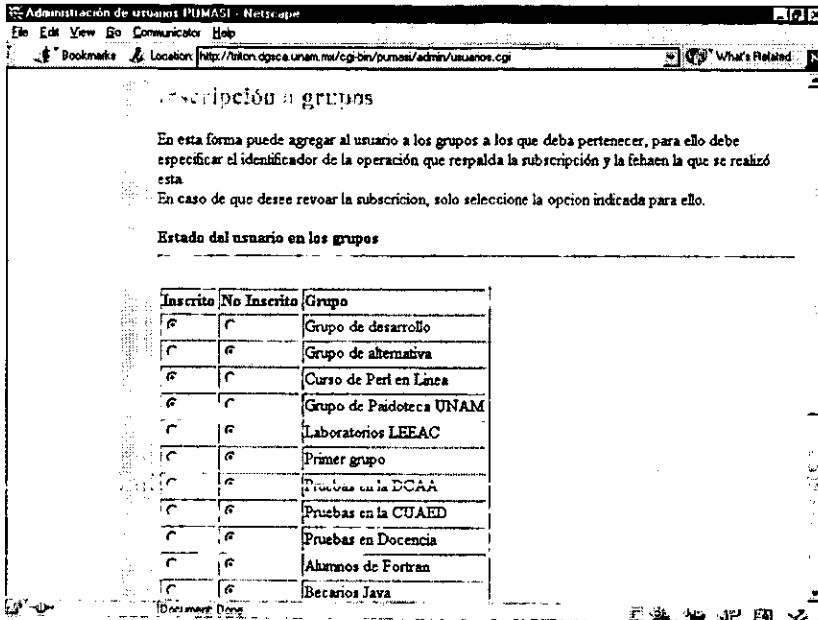
Permite cambios en la Información

Al oprimir el boton "Realizar los cambios, se procedera a cambiar la información del usuario siempre y cuando haya seleccionado la opción para permitir cambios en la información.

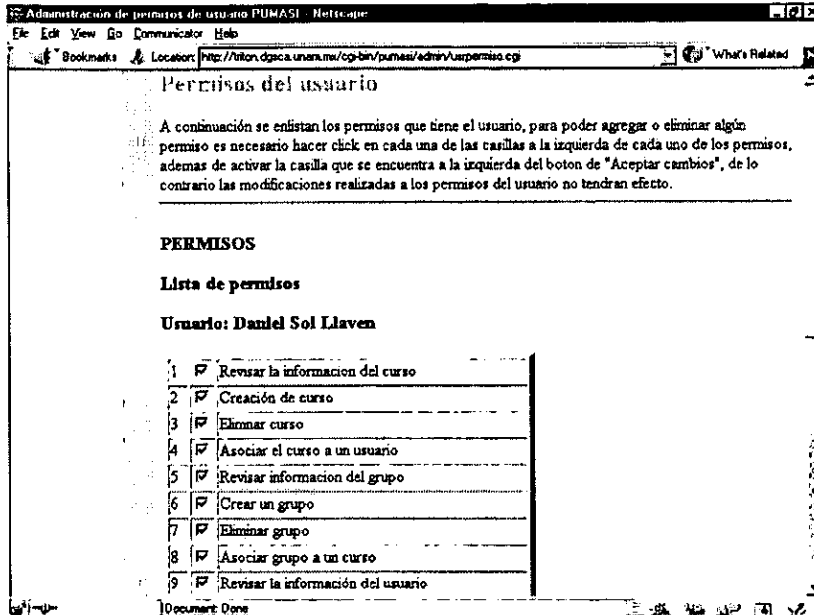
Document Done

Al solicitar la modificación de un usuario se presenta una forma de captura con la información que actualmente se tiene de modo que se pueda revisar, para evitar cambios accidentales se colocó el campo "Permitir cambios en el material" que está inactivo por defecto y que debe estar activo para que se realicen cambios en la información.

El proceso de eliminar a un usuario es muy similar, con la diferencia de que en el botón se aclara que se estará eliminando al usuario en lugar de modificar su información.

Subscripciones

Una parte de la administración consiste en establecer los vínculos entre los usuarios y los grupos, los grupos y los cursos y los cursos y sus responsables. En la pantalla de Inscripción a grupos podemos controlar a cuales grupos y por consecuencia a que materiales tendrá acceso un usuario, basta con elegir si deseamos que el usuario elegido pertenezca o no a cada uno de los grupos (que aparecen con el estado actual de inscripciones) y solicitar la actualización de la información.

Permisos del usuario

Después de elegir a cual usuario deseamos modificar, del mismo modo que en la administración de usuarios, se nos presenta la lista de todos los permisos reconocidos por el sistema de modo que se elija cuales se desea otorgar al usuario, una vez que se realizan las modificaciones deseadas se solicita la actualización de la información y se confirma con una nueva forma, igual a la anterior, pero con los nuevos permisos otorgados al usuario.

Capítulo V **Análisis de
Resultados**

Capítulo V **Análisis de Resultados**

La primera etapa de desarrollo del sistema se ha concluido con éxito y aunque restan algunos módulos por implementar se está comenzando a utilizar el sistema para generar algunos materiales educativos:

- Apoyo a la generación de material para universidad abierta
- Publicación regular de artículos de opinión sobre nuevas tecnologías en la coordinación de prospección

En ambas actividades es crítico el reducir los tiempos en que se genera el material, en el primero por los elevados objetivos que se han planteado que serán aún más apremiantes una vez que se reanuden las actividades normales en escuelas y facultades.

En lo referente a la publicación de artículos de opinión se puede utilizar a pesar de que el personal que lo ocuparía es capaz de generar HTML por si mismo como un mecanismo de uniformar el diseño de los artículos generados, para minimizar el tiempo que esto ocupe y para dar un mejor control de la generación cotidiana de materiales por un equipo de aproximadamente 20 personas.

A. Limitaciones

1. Limitado rango de usuarios

Para simplificar el mecanismo de publicación de un material en el sistema se restringió en gran medida el potencial de los materiales en Internet que se puede obtener, de modo que los usuarios que tengan conocimientos de publicación HTML y de Internet se sentirán seguramente encasillados por la estrecha funcionalidad disponible.

Esto restringe a los usuarios del sistema tan solo a aquellos que no cuentan todavía con conocimientos de publicación HTML o que no desean dedicar sino un mínimo de tiempo a los materiales a publicar.

2. Búsquedas

Aunque el Sistema Gestor de Base de Datos podría soportarlo, no se han implementado búsquedas de palabras clave ni en las fichas de catálogo ni en el contenido de los materiales.

3. Materiales transitorios

Actualmente se observa una gran proliferación de sitios que tienen por principal tarea el entregar las noticias correspondientes a áreas de interés especificadas por el usuario cada vez que el usuario revise la información disponible, el sistema desarrollado no contempla la posibilidad de elegir un subconjunto de los materiales disponibles para el usuario y presentárselos de modo aleatorio o basado en criterios de tema.

4. Limitaciones de distribución

Al estar el sistema desarrollado en Perl no es conveniente elaborar versiones binarias del mismo, para distribuirlo se debe entregar el código del programa. Existe una alternativa a entregar el código del programa, que consiste en cifrar el código usando un módulo especial de pre-procesamiento de código Perl que lo extraiga en base a una llave previamente establecida para su ejecución evitando así la reproducción no autorizada, modificación o uso indebido del mismo. Sin embargo, esta es una característica de Perl y no estrictamente del sistema elaborado.

5. Transplante a plataformas diferentes

A pesar de que los elementos del sistema se encuentran disponibles para las principales variaciones de UNIX el sistema no puede ser transplantado a plataformas diferentes con facilidad, en especial cuando se pretende cambiar de Sistema Gestor de Base de Datos ya que las rutinas de interfaz con la base de datos no manejan los tipos de datos de modo transparente a la aplicación, de modo que la información de fechas y de generación de llaves deberá ser adaptada al nuevo SGBD.

El proceso de instalación del sistema involucra una configuración de los programas que aunque sencilla no se ha automatizado y debe realizarse a mano por personal especializado.

6. Uso limitado de las capacidades del DBMS

Como se decidió emplear un SGBD mínimo, en este caso el Mini SQL, no se emplearon capacidades comunes de los SGBD modernos como son los procedimientos almacenados y los procedimientos para salvaguardar la integridad referencial (Triggers), y muchas de estas funciones son cumplidas por el sistema.

Se necesita también de un mantenimiento a la Base de Datos externo a las funciones del sistema, en lo particular en lo referente a la generación de llaves.

Los registros de operaciones del sistema se almacenan como registros en la tabla de "acción" de la Base de datos y cada una recibe un identificador único, la base de datos solo tiene capacidad para generara una cantidad finita de tales identificadores, pero más importante aún, no implementa ningún mecanismo para eliminar los registros que ya no

sean útiles por lo que eventualmente saturarán al sistema, se estima que se requiere de una purga de los registros de operaciones cada mes en el peor de los casos. (Estimado para una carga de una operación por segundo si no se desea exceder de 1 GB de almacenamiento empleado en el registro de las operaciones).

B. Potencial

1. Mantenimiento

Estando en la primera etapa de pruebas del sistema por personas fuera del equipo de desarrollo se está realizando también la experiencia de capacitación de personal nuevo en el mantenimiento del sistema. Al ser este un desarrollo realizado en el marco de mis actividades en la Coordinación de Prospección de la Dirección General de Servicios de Computo Académico se planea su continua aplicación y ampliación por un equipo de trabajo en constante cambio, por lo que es necesario que pueda ser dominado en corto tiempo por personal adecuadamente capacitado.

Hasta el momento se han obtenido resultados satisfactorios en cuanto a la capacitación de nuevos equipos de desarrollo, ya que después de un mes de tener el primer contacto con el sistema ya se cuenta con un equipo de tres becarios capaces de desarrollar nuevos módulos en concordancia con las normas que emplee para el desarrollo original.

Para evaluar la importancia de este resultado, debe tomarse en consideración que el sistema cuenta con seis mil líneas de código en 14 CGI y una librería y 52 páginas HTMLP.

En particular, los becarios de la DGSCA-DCAA: Manuel Alejandro Alfaro Florez y Luis Alejandro Castillo Vilchis. Demostraron la rápida comprensión del sistema al desarrollar uno de los módulos, el de asignación de permisos en la sección de administración, apegándose a las normas y formas en que se desarrollo el resto del sistema.

Los elementos que han resultado de más valía para la eficiente comprensión del sistema son el correcto diseño de la Base de Datos y la aparente simplicidad del código en Perl. La disponibilidad de información de diseño de la base de datos provee una visión totalizadora del sistema y permite ubicar las actividades de los CGI en función de sus interacciones con la base de datos. En lo que el código Perl permite que al ver el código se comprendan las acciones que los CGI toman a lo largo de sus ejecuciones.

Diversas técnicas y criterios empleados en el desarrollo del sistema se han adoptado como normas para el desarrollo, evitando en lo posible se evite que el sistema diverja en

diversas estrategias de implementación. Se detallan algunas de estas técnicas en el anexo A donde se revisa un ejemplo del código del sistema.

2. Mantenimiento a la Base de Datos

Gracias a las capacidades del software para diseño de bases de datos las revisiones que se ha requerido hacer a la base de datos se han podido resolver casi sin impacto para la operación del sistema, ya que al contar con generación automática de scripts de creación de las tablas e índices a modificar solo se requiere hacer las adecuaciones a los datos a restablecer en las tablas. Afortunadamente no ha habido cambios mayores en la base de datos por lo que hasta el momento ha habido muy poca necesidad de este tipo de cambios.

3. Desarrollos adicionales

La accesible capacitación de nuevo personal de desarrollo que emplee las mismas herramientas y siga los mismos lineamientos de desarrollo aunado a una adecuada documentación del sistema hace factible el desarrollo sostenido de nuevas capacidades conforme se presenten las necesidades específicas, algunas de las que consideramos pueden ser incluidas en un futuro cercano son:

a) Recursos de Audio y Video

Con la creciente disponibilidad de material de audio y video en la red se está considerando el ampliar las capacidades de los recursos de modo que también puedan incluir los flujos de estos medios en la Red, incluir recursos como Real Video, Real Audio y todo tipo de recursos que basen su reproducción en Plugins es relativamente sencillo por lo que puede ser implementado en poco tiempo.

b) Modificación del catálogo de materiales

Existen varias propuestas de catálogos para materiales electrónicos que están siendo evaluadas por la Dirección General de Bibliotecas, una vez que se establezca una norma universitaria para catálogos de materiales educativos será una prioridad el adecuar este sistema y los que le sigan de modo que usen los campos establecidos en esta norma, ganando con ello no solo la solidez de un catálogo formalmente establecido, sino la capacidad de pertenecer a bases de datos que se establezcan a nivel universidad.

c) Integración con ambientes de colaboración

Paralelamente al desarrollo de este sistema se han realizado en la DGSCA proyectos orientados al apoyo de la colaboración de grupos de trabajo que solo han tocado la superficie de la generación y catalogación del materiales, se planea integrar este sistema con uno de estos desarrollos para lograr construir una estructura completa de comunidades virtuales.

d) Módulos para edición de Fórmulas

Uno de los elementos más comúnmente requeridos en el material educativo para las escuelas y facultades de la UNAM son las fórmulas matemáticas de cierta complejidad. Para elaborarlas en un formato compatible con los navegadores se requiere de una herramienta gráfica de edición y del uso de formatos estándar.

La edición de este tipo de recursos va más allá de lo que se puede lograr con CGI, por lo que se cree que lo más accesible sin sacrificar la libertad de formato y la compatibilidad con diversas plataformas sería realizar un editor gráfico de fórmulas en Java que genere archivos en XML que sean interpretados en gráficas estáticas o en applets de presentación.

e) Programas de instalación

Aún no se cuenta con un esquema coherente de programas que puedan adaptar las variables de configuración del sistema para que pueda ser distribuido sin apoyo del equipo de desarrollo, antes de que se comience a distribuir como producto de software si se decide esta forma de distribución, será necesario generar los programas necesarios para adaptar el sistema a nuevos ambientes.

4. Compatibilidad con otros sistemas

Como los CGI son en realidad clientes de un Sistema Gestor de Base de Datos convencional, no hay ningún impedimento para que otros sistemas se basen en la misma base de datos e incluso la amplíen siembre y cuando respeten los requisitos de PUMASI y se eviten condiciones de carrera, como el sistema se hizo en mente para ser usado por muchos usuarios simultáneamente son muy pocos los puntos en los que debería tenerse cuidado en el uso de la base de datos por varios sistemas.

Los materiales que PUMASI genera son reflejados en documentos HTML convencionales por lo que puede continuarse con su desarrollo por fuera del sistema, incluso sin retirarlos del él, esto nos da la ventaja de que si se cuenta con personal especializado en realización de páginas y se desea enriquecer el material generado automáticamente no es necesario renunciar a las ventajas en cuanto a control de usuarios que da el sistema.

Gracias a los módulos que se emplearon para la construcción del sistema, se cuenta con alguna libertad en cuanto al Sistema Gestor de Base de Datos a emplear ya que se empleó la librería DBI de Perl que permite emplear las mismas funciones para operar con diversas bases de datos. Sin embargo, el manejo de fechas y la generación de identificadores únicos para los índices son implementado de modos diferentes, en previsión a esto se aislaron las funciones que se encargan de convertir estos datos a un formato interno, por lo que se espera que el impacto de un cambio sea mínimo, sin embargo no se cuenta aún con las rutinas para otros SGBD por lo que se tendría que pasar por una etapa de desarrollo y de pruebas antes de liberar PUMASI para otro SGBD.

C. Conclusiones

La experiencia adquirida con el desarrollo de este sistema nos lleva a tres conclusiones principales:

- Se requiere de herramientas capaces de integrar coherentemente los recursos de red en una sola herramienta sensible a las necesidades del usuario mucho más allá de los alcances de este sistema.
- La adecuada catalogación y existencia de mecanismos de acceso a la información de apoyo a la educación debe ser una prioridad para poder sacar cabal provecho de los esfuerzos invertidos en realizarlos.
- El uso de arquitecturas de cliente abierto permite un fácil acceso a nuevos sistemas ya que centralizan la administración y minimizan la necesidad de soporte técnico a los usuarios.

La necesidad de integrar los recursos de información apoyados por las redes de computadoras surge del gran contraste que hay entre la enorme cantidad de materiales individuales que existen a los que se tiene acceso y la extrema dificultad de encontrar materiales realizados por personas que se desarrollen en el mismo esquema de trabajo que nosotros. El propósito principal de este sistema y los demás de su tipo es el de

atraer a las personas que están involucradas en una línea de trabajo y facilitarles la comunicación entre ellos, de modo que sus opiniones y colaboraciones a la comunidad sean cabalmente aprovechadas y que en el grupo encuentre un valioso elemento de retroalimentación.

La catalogación es un elemento que desde hace ya mucho tiempo se ha identificado como indispensable cuando se integran grandes acervos de cualquier tipo de información, de lo contrario no es posible acceder a la información que se necesita y el gran volumen de información se convierte en un impedimento en lugar de ser una ventaja. Gracias al interés que hay por las bibliotecas de ampliar sus acervos con recursos electrónicos hay muy serias iniciativas para sistemas de catalogación de recursos en la red y de cómputo basados en los empleados en las bibliotecas que son muy adecuados a las necesidades de la información a pesar de las grandes diferencias en el medio.

La arquitectura de Cliente Abierto que se inicia con sistemas de Gopher y se populariza con el Web es hoy día la solución preferida en la industria por su rápido desarrollo, implementación e inmediata distribución. Al compartir un cliente estándar como es el navegador Web se elimina la necesidad de dar asistencia técnica en la instalación de un nuevo cliente por cada aplicación como era el caso de la arquitectura Cliente Servidor, mas aún, la producción y soporte de los navegadores queda a cargo de compañías especializadas y el proceso de instalación a cargo del propio usuario. Con la aparición de tecnologías como los Applets de Java se puede incluso brindar toda la funcionalidad de los clientes tradicionales sin sacrificar la compatibilidad y facilidades de los Clientes Abiertos. Debe mencionarse que lo que ha hecho posible el desarrollo de la arquitectura de Clientes Abiertos es la amplia difusión y apego a estándares, lo que permite que desarrollos completamente independientes sean capaces de utilizar los mismos recursos.

La convergencia de estas necesidades delimitan lo que se está conociendo como Portales o Sistemas de Manejo Global de Información.

1. Tendencia de la Industria

El principal requerimiento que se ha detectado en el tiempo de pruebas y que debe ser la principal prioridad para la próxima etapa de desarrollo es la integración de la estructura existente de control de usuarios y de materiales a un Ambiente de Manejo Global de Información.

El objetivo de los ambientes de manejo global de información es el servir no solo de repositorio y punto de distribución de información, sino ser capaces de detectar las necesidades del usuario y reaccionar acorde a estas necesidades. Es el esquema que se

está popularizando en los así llamados “Portales” que a partir de un perfil de usuario generan índices de materiales con los materiales disponibles más relevantes para los intereses expresados, además, se constituyen comunidades virtuales con las personas que comparten ciertos intereses, y se apoya a estas comunidades con la vigilancia y participación de personal especializado en el tema de modo que el desarrollo de las discusiones resulte en un máximo provecho para los usuarios.

Este tipo de sistemas requieren de una gran integración entre todos los servicios que Internet ofrece, de un equipo de personas que vigilen y apoyen el desarrollo de las comunidades y de un sistema de catalogación de la información con el suficiente detalle como para ser empleado en la predicción del interés de los usuarios según los intereses que se tengan registrados de ellos.

Una de las tendencias que se está dando en la universidad es que aunque cada institución se encarga de generar sus propios materiales para los temas y con las técnicas que le resultan más convenientes, se están generando una gran cantidad de acuerdos entre instituciones para construir acervos conjuntos y para acordar en los mecanismos de catalogación a emplear para ellos.

Bibliografia

Bibliografía

Object Oriented Analysis and Design Using the Unified Modeling Language (UML), Student Guide, Andy Olsen y Des Ward, Oracle Corporation México D.F., 1997, 230p

Decline & Fall of the American Programmer, Eduard Yourdon, Prentice-Hall Inc., Englewood Cliffs, New Jersey USA, 1992, 352p

Ingeniería de Software, Richard E. Fairley, traducción Antonio Sánchez Aguilar, Pedro Luis Flores Suárez, McGraw-Hill S.A. de C.V. Edo. de México, México, 1987, 390p

Perl Reference Manual, versión 5.002beta1g, Larry Wall y otros, documentación distribuida por la Comprehensive Perl Archive Network, EUA, 1987, 320p

Computación II, Pascal Lenguaje de programación estructurada y modular, Dirección General de Proyectos Académicos, Antologías para la actualización de los profesores de enseñanza media superior, Ciudad Universitaria D.F. México, 1987, 115p

**Anexo A Muestra de
Código**

Anexo A Muestra de Código

En el desarrollo del sistema se siguieron varios lineamientos para la elaboración de los diversos programas que facilitan su mantenimiento y la rápida comprensión por nuevo personal que se asimile al equipo de trabajo.

A continuación se revisa uno de los programas del sistema haciendo hincapié en los elementos que se utilizan y la función que tienen.

El programa elegido para la muestra; "salida.cgi", se compone casi exclusivamente de los elementos que todos los CGI requieren ya que el código para desempeñar su función es mínimo.

```
#!/usr/bin/perl

# Programa terminador de sesiones 1.0
# Daniel Sol Llaven
# 25 de Mayo de 1997
# PUMA SI

#Librerías
use CGI;
use rutinas;
use DBI;

# Variables de configuración

#Mi propio URL para autoreferencia
$mi_url="http://triton.dgsca.unam.mx/cgi-bin/pumasi/salir.cgi";
# Base de los CGI de pumasi para limitar galleta
$baseCGI="/cgi-bin/pumasi";
#URL para el CGI de entrada
$URLentrada="http://triton.dgsca.unam.mx/cgi-bin/pumasi/entrada.cgi";
#Localización del HTMLP para ir a la entrada:
$htmlpentrada="/usr/users/pumasi/docs/entrada.htmlp";
```

1 El propósito del programa, versión, autor y fecha de elaboración son obligatorios en todo programa.

2 Rutas a archivos, y ligas se colocan en variables al inicio.

```
#Correo electrónico de la administración
$emailadmin="pumasi\@triton.dgsca.unam.mx";
#Manejador de base de datos de DBI
$dbi='dbi:mysql:';
#Base de datos a usar
$dbd='PUMASI';
```

3 Parámetros de conexión a la base de datos.

```
$accion='salida';
$c='s';
```

4 Como se registra el proceso en la bitácora de acciones.

```
#Inicialización del CGI
$def=new CGI;
```

```
#Conexión a base de datos
$dbh=DBI->connect($dbi.$dbd, 'pumasi') || error(5);
```

5 El proceso de verificación de la sesión es casi idéntico en todo el sistema, en caso de que no se apruebe la sesión se genera la forma para que el usuario inicie una nueva.

```
#Verificar que se presente una galleta de sesión valida
$sides=$def->cookie('pumasises');
if(!(%permisos=verses($dbh,$sides,$accion,$c))
{
    #Sesión reprobada por verses o fallo en la verificación,
    #remitir a entrada.
    print $def->redirect($URLentrada);
    #En caso de cambio de URL se debe editar el .htmlp
    open(AE,"<$htmlpentrada") || error 0;
    while(<AE>)
    {
        #s/<!-- Opcional -->/$sides<br>$c1<br>/; #error
        explicado
        print;
    }
    close AE;
}
else
{
    #sesión valida, terminarla y avisar.
    if(finses($dbh,$sides))
    {
        #sesión terminada, remitir a entrada.
```

6 Las formas a generar están en archivos tipo htmlp.

```

print $def->redirect($URLEntrada);
#En caso de cambio de URL se debe editar el .htmlp
open(AE,"<$htmlpentrada") || error 0;
while(<AE>)
{
    #s/<!-- Opcional -->/$idses<br>$cl<br>/; #error
    explicado
    print;
}
close AE;
}
else
{
    #no se termino con éxito la sesión
    error(1,"No se pudo terminar la sesión");
}
}

```

7 Los archivos htmlp cuentan con campos que se pueden usar para desplegar información generada en el proceso y que no son visibles si no se emplean.

8 El reporte de errores se realiza por una rutina común a todo el sistema que no requiere de formas htmlp ni de encabezados hechos por el programa.

Como se observa en el código hay diversos elementos que se consideran comunes a todos los CGI del sistema, cada uno de estos elementos se constituyó con una función específica.

1. Estos datos son considerados como la información mínima para identificar un programa y se les considera obligatorios ya que permiten, incluso después de un largo periodo de tiempo, ubicar la función del programa, el sistema para el que se realizó y las personas responsables, información que suele ser la diferencia entre rescatar o rehacer un sistema.
2. A lo largo del desarrollo y en los procesos de migración o instalación de los programas suelen modificarse las rutas a los archivos y las ligas a otros elementos Web, el colocar estos valores en variables al principio del código y no en los puntos donde se utilizan evitan que se requiera revisar todo el programa para realizar estas modificaciones y abre la posibilidad de automatizar el proceso de instalación del sistema.

3. Al emplear DBI como mecanismo de acceso al Sistema Gestor de Base de Datos se requiere especificar el nombre de la base de datos, el tipo de sistema a emplear y para algunos SGBD el login y password, para simplificar un posible proceso de cambio de base de datos se dejan estos valores en variables y no mezcladas en el código.
4. El sistema genera una bitácora en la que se registran todas las acciones solicitadas, como cada CGI debe especificar una acción diferente se considera parte de los datos específicos sujetos a cambio del programa.
5. Para verificar la validez de las sesiones se emplea una rutina llamada "verses" y cada CGI puede determinar su conducta en caso de que la sesión sea aceptada o rechazada, sin embargo, casi todos los CGI se limitan a generar la forma de entrada al sistema en caso de que la sesión no sea aceptada.
6. La salida de los CGI debe estar dada en un formato reconocido por el navegador, usualmente en HTML, como el diseño de la presentación del sistema es realizado por el equipo de diseño gráfico y de páginas HTML que no tiene conocimientos de programación, se emplearon archivos independientes que siguen la misma sintaxis que el HTML convencional con comentarios que indican donde deben insertarse en la página los resultados de la operación del programa, de este modo se logra integrar con facilidad el trabajo de programación con el de diseño.

Además se recomienda que todos los programas estén indentados de modo que su estructura lógica resulte mas clara cuando se le dé mantenimiento, así como acompañarlos de un archivo de texto donde se explique a detalle la función y los parámetros de entrada y de salida que tenga. Sin embargo el contenido de dicho archivo y el estilo de indentación que se utilice son dejados al criterio del programador.