



Universidad Nacional Autónoma de México

Facultad de Ingeniería

**Diseño y elaboración de una
base de datos en Internet sobre
la Producción Científica del
Instituto de Física (PCIF):
1990-2000.**

T E S I S

Que para Obtener el título de

INGENIERO EN COMPUTACIÓN

PRESENTA:

Angélica Altamirano Casillo

Directora de Tesis Dr. Ana María Vázquez Vargas





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicatoria

Este trabajo de tesis se lo dedico a mis padres y hermanas por su comprensión, ayuda y motivación.

Agradecimientos

A mi Directora de Tesis Dr. Ana María Vázquez Vargas

Por su tiempo, apoyo, comprensión y consejos en la elaboración de este proyecto de tesis.

A mi Papá, a mi Mamá y mis Hermanas Evelyn, Nadia, y Tania

Por su paciencia, apoyo y principalmente su comprensión

Al Instituto de Física

A todo el personal de la Biblioteca del Instituto de Física, pero principalmente a la Lic. Lucila Martínez Arellano por su apoyo y confianza, a Carlos Castaños, Jessy y Nepta por su amistad,

A la Facultad de Ingeniería

Por la formación que adquirí durante mi estancia.

A **Daniel, Alejandro**, por su amistad y consejos

Omar, Gris, Mariza, Lili, José Luis, Elizabeth, Alma, por su amistad

Índice general

1. Introducción	1
2. Definición de PostgreSQL	9
2.1. Definición de PostgreSQL	9
2.1.1. Lenguajes de programación que utiliza PostgreSQL	10
2.1.2. Aplicaciones Web con PostgreSQL	10
2.2. Breve Historia de PostgreSQL	10
2.3. Fundamentos de la Arquitectura de PostgreSQL	11
2.4. Características de PostgreSQL	15
2.5. Seguridad en PostgreSQL	17
3. Selección y Evaluación de otras herramientas (Apache, PHP)	20
3.1. Definición de PHP	20
3.1.1. Funcionamiento de PHP	21
3.1.2. Aplicaciones que se pueden hacer en PHP	23
3.1.3. Seguridad en PHP	25
3.2. Definición de Apache	25
4. Diseño de la base de datos con PostgreSQL	
4.1. Diseño de la Base de Datos "PCIF"	28
4.1.1. Planteamiento del problema	28
4.2. El Modelo Entidad-Relación	30
4.2.1. Construcción del modelo Entidad-Relación y las relaciones	31
4.2.1.1. Identificar las entidades	31
4.2.1.2. Identificar las asociaciones	32
4.2.1.3. Identificar los atributos de las entidades y de las asociaciones	33
4.2.1.4. Determinar los dominios de los atributos	35
4.2.1.5. Determinar las llaves primarias y foráneas	36
4.2.1.6. Dibujar el diagrama entidad-relación	37
4.2.1.7. Relaciones que se obtienen a partir del Diagrama E/R	37
4.2.2. Creación de la base de datos	39
4.2.2.1. Instrucción "CREATE TABLE"	39
4.2.2.2. Otorgar Privilegios a la base de datos con "GRANT"	41
4.2.2.3. Insertar datos en las tablas	44
5. Manual de Instalación de PostgreSQL, Apache y PHP	46
5.1. Iniciación de PostgreSQL	46

5.1.1.	Iniciación y arranque del servidor PostgreSQL.....	46
5.1.2.	Notación de las ordenes por los diferentes usuarios.....	47
5.2.	Iniciación de Apache y PHP para su uso en el sistema.....	48
5.2.1.	Actualización del archivo de configuración	48
5.2.2.	Arranque del servidor Apache.....	49
6.	Interfaces Desarrolladas con PHP HTML y Flash.....	50
6.1.	Interfaces para la búsqueda del artículo por investigador o titulo o palabra clave.....	50
6.1.1.	Interfaz principal para la búsqueda de la producción científica del Instituto de Física.....	50
6.1.2.	Interfaz que muestra de la búsqueda por titulo del artículo, el artículo solicitado.....	52
6.1.3.	Interfaces para la búsqueda por investigador.....	55
6.1.4.	Interfaces para la búsqueda por palabra clave.....	58
6.1.5.	Código en PHP de los queries para la búsqueda del artículo por titulo o investigador o palabra clave.....	61
6.1.5.1.	Query de la búsqueda por Título del artículo.....	61
6.1.5.2.	Query de la búsqueda por Investigadores.....	61
6.1.5.3.	Query de la búsqueda por palabra clave del artículo.....	62
6.2.	Interfaces para el administrador.....	62
6.2.1.	Interfaz de autenticación para el administrador de la base de datos en Internet.....	63
6.2.2.	Interfaz para insertar los artículos de la Base de Datos.....	64
6.2.2.1.	Interfaz Insertar Artículos.....	65
6.2.2.1.1.	Código PostgreSQL.....	66
6.2.2.1.2.	Código PHP.....	66
6.2.2.2.	Interfaz Insertar palabras claves.....	68
6.2.2.2.1.	Código PostgreSQL.....	68
6.2.2.2.2.	Código PHP.....	69
6.2.2.3.	Interfaz Insertar editoriales.....	71
6.2.2.3.1.	Código PostgreSQL.....	72
6.2.2.3.2.	Código PHP.....	72
6.2.2.4.	Interfaz Insertar procedencia.....	74
6.2.2.4.1.	Código PostgreSQL.....	75
6.2.2.4.2.	Código PHP.....	75

6.2.2.5.	Interfaz Insertar investigadores.....	77
6.2.2.5.1.	Código PostgreSQL.....	77
6.2.2.5.2.	Código PHP.....	78
6.2.3.	Interfaz para Actualizar.....	80
6.2.3.1.	Actualizar Artículos.....	81
6.2.3.1.1.	Código PostgreSQL.....	81
6.2.3.1.2.	Código PHP.....	81
6.2.3.2.	Actualizar Palabras claves.....	82
6.2.3.2.1.	Código PostgreSQL.....	82
6.2.3.2.2.	Código PHP.....	83
6.2.3.3.	Actualizar editoriales.....	84
6.2.3.3.1.	Código PostgreSQL.....	84
6.2.3.3.2.	Código PHP.....	84
6.2.3.4.	Actualizar procedencias de los investigadores.....	85
6.2.3.4.1.	Código PostgreSQL.....	86
6.2.3.4.2.	Código PHP.....	86
6.2.3.5.	Actualizar investigadores.....	87
6.2.3.5.1.	Código PostgreSQL.....	87
6.2.3.5.2.	Código PHP.....	87
6.2.3.6.	Actualizar procedencias.....	89
6.2.3.6.1.	Código PostgreSQL.....	89
6.2.3.6.2.	Código PHP.....	89
6.2.3.7.	Actualizar abstract.....	90
6.2.3.7.1.	Código PostgreSQL.....	91
6.2.3.7.2.	Código PHP.....	91
6.2.4.	Interfaz para seleccionar un registro de una tabla	93
6.2.4.1.	Buscar Palabra.....	94
6.2.4.1.1.	Código PostgreSQL.....	94
6.2.4.1.2.	Código PHP.....	94
6.2.4.2.	Buscar Editorial.....	96
6.2.4.2.1.	Código PostgreSQL.....	96
6.2.4.2.2.	Código PHP.....	96
6.2.4.3.	Buscar Procedencia.....	98
6.2.4.3.1.	Código PostgreSQL.....	98
6.2.4.3.2.	Código PHP.....	98

6.2.4.4.	Buscar Investigador.....	100
6.2.4.4.1.	Código PostgreSQL.....	100
6.2.4.4.2.	Código PHP.....	101
6.2.5.	Interfaz para Borrar.....	102
6.2.5.1	Código PosgreSQL.....	102
6.2.5.2	Código PHP.....	102
6.2.6.	Interfaz para inicializar una sequencia.....	103
6.2.6.1	Código PosgreSQL.....	103
6.2.6.2	Código PHP.....	103
7.	Conclusiones.....	105
	Glosario.....	107
	Bibliografía.....	119

Introducción

Este proyecto nació de la necesidad de dar difusión a las publicaciones que se generan en el Instituto de Física de la Universidad Nacional Autónoma de México

Actualmente se cuenta en la biblioteca con las publicaciones científicas de los investigadores de dicho instituto, y se requiere que estas publicaciones se tengan en Internet, y debido a los derechos de autor sólo se podrán publicar en la web los abstracts de estas publicaciones.

Para cumplir con este objetivo es necesario construir una base de datos para que sea consultada por Internet.

Por lo tanto en este proyecto de tesis se tuvo que realizar una Base de Datos con las características mencionadas y que funcione en las plataformas de Linux y Windows.

Para solucionar el problema fue necesario sistematizar las publicaciones de los investigadores en una base de datos y para que se encuentre en Internet es necesario diseñar una serie de interfaces para que realice la búsqueda de la información.

Este sistema cuenta con dos secciones, una sección es para el administrador de la base de datos y la otra es para cualquier usuario que consulte la Base de Datos. Esta división se creó para la seguridad de la Base de Datos, dado que solo el administrador de la base de datos es quien puede incrementar, actualizar o borrar los registros, por tal motivo fue necesario una interfaz de autenticación y de esta forma poder validar el campo del login y un password enviado en el formulario HTML y con esta autenticación el administrador podrá realizar estas modificaciones a través de Internet.

En la sección correspondiente al administrador de la Base de datos se le ha diseñado las interfaces en PHP con Flash para insertar, borrar, actualizar y buscar las publicaciones, estas interfaces se pueden realizar en las plataformas Linux como Windows.

En la sección de consulta se tiene la finalidad de realizar tres búsquedas, una de ellas es la **búsqueda por autor** y coautor, la siguiente **búsqueda es por título** del artículo, y la última es por **palabra clave** que el autor y/o los coautores han determinado en la publicación del artículo.

Para la implementación y desarrollo de este trabajo se escogió la Base de Datos PostgreSQL por tener las siguientes ventajas:

1. Es software libre,

2. Es Multiplataforma, PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.
3. Incorpora los cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema y son: **clases, herencia, tipos y funciones,**
4. otras características adicionales que aportan potencia y flexibilidad adicional a la base de datos son: Restricciones (**Constraints**), Disparadores (**triggers**), Reglas (**rules**), e Integridad transaccional. Estas características colocan a PostgreSQL en la categoría de las Bases de Datos identificadas como objeto-relacionales.

Se eligió como lenguaje de programación PHP por contar con las siguientes ventajas:

1. Es gratuito, es software libre,
2. Debido a la sencillez con que se trabaja con las bases de datos,
3. Por tener una sintaxis cómoda ya que no es necesario declarar variables al inicio del programa y se pueden hacer grandes cosas con pocas líneas de código debido a que tiene una amplia variedad de funciones y que son de gran utilidad en el momento en el que se esta programando,
4. Por ser multiplataforma y el disponer de una gran cantidad de librerías que facilitan muchísimo el desarrollo de las aplicaciones.

Las bases de datos son usadas primordialmente para almacenar y recuperar datos y básicamente usadas cuando se tiene grandes volúmenes de datos que se pretenden ser actualizados o seleccionarlos frecuentemente, por lo que en esos casos las bases de datos son la mejor elección.

Otra ventaja que se tiene con las bases de datos es el de permitir ordenar los datos almacenados, recuperar los datos rápidamente, realizar operaciones con los datos.

En los sistemas comerciales actuales, los tipos de datos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres , cantidades monetarias, fechas

Internet es un medio de comunicación y que actualmente se pueden manejar Bases de Datos como MySQL, PostgreSQL, Oracle entre otras;

Internet es un medio para localizar, enviar, recibir información de diversos tipos, aun con las bases de datos. En el ámbito competitivo, es esencial ver las ventajas que esta vía electrónica proporciona para presentar la información, reduciendo costos y el almacenamiento de la información, y aumentando la rapidez de difusión de la misma.

Internet provee de un formato de presentación dinámico para ofrecer campañas y mejorar negocios, además de que permite acceder a cada sitio alrededor del mundo, con lo cual se incrementa el número de personas a las cuales llega la información.

Millones de personas alrededor del mundo hacen uso de Internet, lo cual demuestra el enorme potencial que esta red ha alcanzado, con lo que se puede decir que en un futuro no muy lejano, será el principal medio de comunicación utilizado para distintos fines.

Pero, no sólo es una vía para hacer negocios, sino también una gran fuente de información, siendo éste uno de los principales propósitos con que fue creada.

Una gran porción de dicha información requiere de un manejo especial, y puede ser provista por Bases de datos.

En el pasado, las Bases de datos sólo podían utilizarse al interior de las instituciones o en redes locales, pero actualmente la Web permite acceder a bases de datos desde cualquier parte del mundo. Estas ofrecen, a través de la red, un manejo dinámico y una gran flexibilidad de los datos, como ventajas que no podrían obtenerse a través de otro medio informativo.

Con estos propósitos, los usuarios de Internet pueden obtener un medio que puede adecuarse a sus necesidades de información, con un costo, inversión de tiempo, y recursos mínimos. Asimismo, las Bases de datos serán usadas para permitir el acceso y manejo de la variada información que se encuentra a lo largo de la red.

Definición de Internet

Las redes que se encuentran interconectadas abarcan desde grandes instalaciones gubernamentales, como la NASA, hasta redes universitarias o mantenidas por organizaciones sin fines de lucro, como la Cruz Roja y corporaciones como IBM, Microsoft, etc. Maestros, estudiantes y cualquier persona en particular utilizan Internet cualquiera sea el fin. Pero a pesar de esta diversidad de personas, empresas e instituciones que la utilizan, nadie es su dueño. Simplemente cada usuario paga su conexión para acceder a ella.

La utilización de este medio es tan variada para los propósitos de cada usuario que se puede investigar desde un tema de clase de un estudiante como la música que a éste le interese.

[Referencias en Internet

<http://es.tldp.org/Postgresql-es/web/navegable/admin/security.html>

<http://www.dbinternet.com.ar/imple.htm>

<http://www.virtual.unal.edu.co/cursos/economicas/92211/lecciones/unidad5/internet1.htm>

<http://www.sobl.org/traduccion/practical-postgres/node55.html>

<http://www.sobl.org/traduccion/practical-postgres/node72.html>

La escala de Internet es realmente abrumadora: 40 millones de usuarios, 6 millones de computadoras conectadas... alrededor de 250 Mb de información se agregan a los grupos de noticias cada día... Pero todo esto tuvo un comienzo...

Algunas fechas importantes sobre Internet

Corría el año 1969 cuando un grupo de personas da a inicio a Internet. Nace como un proyecto de la Agencia De Proyectos De Investigación De Defensa Avanzados de EE.UU. Surge de la necesidad de intercambiar información entre los investigadores y científicos militares, ubicados en sitios apartados. Esta red, que en principios se suponía debía soportar hasta un ataque nuclear sin perder la conexión, originariamente recibió el nombre de DARPANET y estaba conformada por cuatro computadoras interconectadas.

En 1972 y luego de que comenzara a utilizarse el protocolo NCP y se haya escrito el primer programa de correo electrónico para una red distribuida, ya se habían conectada 37 computadoras a la red que pasó a denominarse ARPANET debido a un cambio en el nombre de la agencia que la administraba. En los años sucesivos el servicio se extendió hasta Inglaterra y Noruega pero como siempre ocurre en estos casos la aplicación más utilizada no fue la que se había definido originariamente (Investigación) sino un servicio subvalorado en un principio: el correo electrónico o E-Mail.

En 1974 se presenta un proyecto de la red ETHERNET en donde se publica el protocolo TCP usado actualmente en Internet y con diversos programas de comunicación. Así se creó una rama comercial de ARPANET denominada TELENET.

En 1976, en los laboratorios de BELL, de AT&T, se desarrolla el UUCP (Unix to Unix Communication Program), usado en sistema de E-Mail por modem y se crea USENET empleando dicho sistema.

En 1982, ARPANET adopta TCP/IP (IP: Internet protocol) y se funda la red europea EUNET.

En 1984 se crea JUNET que es la NET de Japón. Este mismo año, la Fundación Nacional De Ciencias De EE.UU. (NSF) establece la NSFNET (basado en la tecnología de ARPANET), que se origina a partir de la necesidad de compartir la potencia de grandes Super-Computadoras (eran tan grandes y caros que sólo se construyeron cinco centros), instaladas por dicha fundación, con la comunidad académica. En principio se había previsto utilizar ARPANET pero a esta altura ya se encontraba saturada. NSFNET estaba dividida en regiones y cada una de ellas tenía acceso a por lo menos una Super-Computadora, y el resto conectado en forma directa a través de distribuidores.

[Referencias en Internet

<http://www.virtual.unal.edu.co/cursos/economicas/92211/lecciones/unidad5/internet1.htm>]

En 1987 ya había demasiada gente utilizándola, y no solo para la investigación académica, resultando así insuficiente. En el transcurso de este año se redimensionó totalmente incorporando formas de acceso más rápidas gracias al aumento de la velocidad de modems y computadoras, lo que permitió extender su utilización por parte de investigadores universitarios, agencias gubernamentales e incluso a organizaciones internacionales dedicadas a la investigación, con la única condición de pertenecer a países aliados a EE.UU.

En 1989, COMPUSERVE, uno de los mayores sistemas de datos en línea se vincula a la red brindando sus servicios.

En 1990 la red ARPANET es discontinuada y mediante la caída del Muro de Berlín, la red (o Internet, como se la conoce actualmente) se abre para todo aquel que pudiera conectarse.

En 1991 se crea WAIS (Wide Area Information Service) que facilita el acceso a datos en Internet; junto a este hace su aparición el sistema Gopher de menús.

En 1992 se crea la WWW (World Wide Web) siendo recién en 1993, con el uso masivo de esta herramienta, cuando se descubren las fabulosas capacidades de la red.

1993. El mundo financiero se interna a los servicios de Internet para hacer negocios.

1994. El Instituto de Tecnología de Georgia analizó una encuesta para observar el perfil del usuario de Internet. Se crea centros comerciales dentro de Internet y el número de servidores llega a 2.000.000 distribuido por todo el globo.

1995. La Nacional Science Foundation entrega la espiga dorada de la red de empresas de telecomunicaciones como Spring, American y Pacific Bell. Los servidores suman 4.000.000 y sus usuarios 30.000.000, alrededor del mundo

En Argentina se abre el acceso comercial a Internet en 1995, 26 años después de su aparición.

En México la organización encargada de la administración del nombre de dominio territorial (ccTLD, country code Top Level Domain) es "El Network Information Center". Donde el código asignado a cada país es según el ISO 3166. En México el código que le corresponde es el "mx". Entre sus funciones de esta organización es el de proveer los servicios de información y registro para .MX así como la asignación de direcciones de IP y el mantenimiento de las bases de datos respectivas a cada recurso.

En lo que respecta a México, la historia de Internet comienza a finales de la década de los 80's cuando el ITESM, Campus Monterrey establece conexión directa a Internet. Merit Network, Inc indica Febrero de 1989 como la fecha de conexión de México a NFSNET (Internet). En esos momentos se conecta el primer equipo a Internet bajo el dominio .mx: dns.mty.itesm.mx con la dirección 131.178.1.1.

La máquina para el primer servidor de nombres para el dominio ".mx" utilizada fue una Microvax-II, digital, lo fue hasta el 6 de septiembre de 1993, la sustituyó una Sun SPARC Classic con 48n MB en RAM y 400 MB en disco.

En el año de 1987, el Instituto Tecnológico y de Estudios Superiores de Monterrey, en el campus Monterrey (ITESEM) se conectó a BITNET a través de líneas conmutadas por medio de una línea privada analógica de 4 hilos a 9600 bits por segundo, en 1989 lo hizo a Internet al enlazarse por medio de la Universidad de Texas en San Antonio (UTSA), por la misma línea privada. La Universidad Nacional Autónoma de México accedió a Internet por medio de una conexión vía satélite de 56 Kbps con el Centro Nacional de Investigación Atmosférica de Boulder, Colorado, siendo éste el segundo nodo de Internet en México. Después se interconectaron ambas universidades mexicanas usando líneas privadas analógicas de 9600 bps, velocidad suficiente para proveer correo electrónico, transferencia de archivos y acceso remoto.

Poco a poco se fueron incorporando a Internet otras instituciones educativas mexicanas como son: Universidad de Chapingo en el Estado de México, el Centro de Investigación de Química Aplicada de Saltillo, el Laboratorio Nacional de Informática Avanzada de Jalapa, Veracruz, los cuales se conectaban al ITESEM para salir a Internet. Para este entonces, en México ya existía un organismo llamado RED-MEX, formado por la academia y dirigida por una organización civil, donde se discutían las políticas, estatutos y procedimientos que habrían de regir y dirigir el camino del control de la red de comunicación de datos de México. Tiempo más tarde, surgió otro organismo denominado MEXNET que reunía representantes legales de cada institución, el cual incluía a varias universidades de distintos lugares del país. Dicha organización, en 1992, establece una salida de 56 kbps al Backbone de Internet.

En 1993 la CONACyT se conecta a Internet mediante un enlace satelital al NCAR (Centro Nacional de Investigación Atmosférica) al igual que el ITAM, la UAM, en ese mismo año, se establece como el primer NAP (Network Access Point), al intercambiar tráfico entre dos diferentes redes. A finales de este año en México ya se contaba con distintas redes: MEXnet, Red UNAM, Red ITESEM, RUTyC (desaparece el mismo año), BAJAnet, Red total CONACyT y SIRACyT. Fue en 1994, con la fundación de la Red Tecnológica Nacional (RTN), integrada por MEXnet y CONACyT, que se generó un enlace a 2 Mbps (E1).

En el mismo año, Internet se abre en el ámbito comercial en México, con lo cual se inicia una nueva era de desarrollo para nuestro país que beneficia a todas las

personas, empresas o instituciones que deciden participar en el proyecto desde sus inicios, ya que hasta entonces sólo instituciones educativas y de investigación tenían acceso a la súper carretera de la información.

A finales de 1995 se crea el Centro de Información de Redes de México (NIC - México) el cual se encargó de la coordinación y administración de los recursos de Internet asignados al país, como son la administración y delegación de los nombres de dominio bajo ".mx". En 1996, se registran cerca de 17 enlaces E1 contratados con TELMEX para uso privado, asimismo se consolidan los principales ISP (proveedores de servicios de Internet) en el país, de los casi ya 100 ubicados a lo largo y ancho del territorio nacional. Para el año de 1997 existen más de 150 ISP's, ubicados en los principales centros urbanos: Cd. De México, Guadalajara, Monterrey, Chihuahua, Tijuana, Puebla, Laredo, Saltillo, Oaxaca, entre otros.

Actualmente, Internet es utilizado tanto por instituciones educativas y gubernamentales, empresas privadas y personas de todo el mundo, entre quienes se llevan a cabo intercambios constantes de información dando origen a la llamada globalización de la comunicación. Hasta el día de hoy, gracias a Internet, se puede recibir información al instante de cualquier parte del mundo, agilizando y facilitando de esta forma el proceso comunicativo a distancia. De esta manera hemos llegado a superar los obstáculos de la comunicación a través de los años, los últimos 50 años han sido de gran avance, ahora sólo queda preguntarnos:

[Referencia en internet

http://www.interware.com.mx/tecnologia/tecnologia/iwetecnologia_historia_internet04.html

<http://www.nic.mx/es/NicMexico.Historia>]

El plan de esta tesis es desarrollar:

En el **capítulo 1** presentar la definición del problema y la solución que se le ha dado.

En el **capítulo 2** explicar brevemente lo que es la base de Datos PostgreSQL , sus características y mencionar las ventajas de su uso.

En el **capítulo 3** explicar y detallar las herramientas utilizadas en el desarrollo de este trabajo.

En el **capítulo 4** En este capítulo se explica en breve como está construida la base de datos del sistema.

En el **capítulo 5**

Se explican que procedimientos se tiene que seguir para levantar los servicios de PostgreSQL, Apache y PHP en el servidor

En el **capítulo 6**

Se muestran las interfaces desarrolladas en el lenguaje PHP con Flash

En el **capítulo 7** se presentan las conclusiones de este proyecto de tesis.

CAPÍTULO 2

Base de Datos con PostgreSQL

2.1 Definición De PostgreSQL

PostgreSQL

PostgreSQL es la base de datos Open Source (gratuita) más avanzada del mundo. Muchas organizaciones, incluyendo grandes corporaciones, instituciones gubernamentales y pequeños negocios en línea usan PostgreSQL para manejar sus datos más valiosos y aplicaciones de misión crítica ya que aprovechan las ventajas de:

- ✍ Poder instalar un número ilimitado de veces sin temor de sobrepasar la cantidad de licencias, la principal preocupación de muchos proveedores de bases de datos comerciales,
- ✍ Velocidad y rendimiento excepcionales,
- ✍ Confiabilidad a toda prueba,
- ✍ Seguridad de primera clase,
- ✍ Flexibilidad para extenderse según se requiera,
- ✍ Diseño altamente escalable,
- ✍ Red mundial de Proveedores Independientes de Software (ISV),
- ✍ Muchas opciones de soporte,
- ✍ Mínimos requerimientos de administración,
- ✍ Bajo Costo Total de Operación (TCO),
- ✍ Conforman a Estándares ANSI

Con una historia de desarrollo de mas de 16 años, una de las comunidades de desarrollo más fuertes del mundo y una reputación global de una alta calidad de ingeniería de software, PostgreSQL hace su trabajo sin problemas.

2.1.1 Lenguajes de programación compatibles con PostgreSQL

C

C++

Java

Perl

PHP

Python

[Referencias en Internet <http://www.sobl.org/traducciones/practical-postgres/node12.html>
<http://www.postgresd.org.mx/#Cat>]

TCL

En Unix se puede utilizar Perl, PHP, C.

2.1.2 Aplicaciones WEB con PostgreSQL

Se puede crear aplicaciones para una interfaz en la web, que genere páginas web, con la posibilidad no solo de consultas sino con altas, bajas y modificaciones, con la alternativa de actualizar a través de la web, agregados de seguridad a los mecanismos estándar,

La combinación puede ser PHP-PostgreSQL (aunque también es Perl-PostgreSQL, Java-PostgreSQL...).

2.2 Breve historia de PostgreSQL

El Sistema administrador de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL (y anteriormente llamado Postgres95) está derivado del paquete Postgres escrito en Berkeley, PostgreSQL es el manejador de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, Tcl y Python. ...).

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado *Ingres* en la Universidad Berkeley de California. *Ingres* fue más tarde desarrollado comercialmente por la Relational Technologies/*Ingres Corporation*.

En 1986 otro equipo dirigido por *Michael Stonebraker* de Berkeley continuó el desarrollo del código de *Ingres* para crear un sistema de bases de datos objeto-relacionales llamado *Postgres*. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, *Postgres* fue renombrado a *PostgreSQL*, tras un breve tiempo como *Postgres95*. El proyecto *PostgreSQL* sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

[Referencias en Internet

<http://es.tldp.org/Postgresql-es/web/navegable/tutorial/x56.html>

<http://www.postgresql.org.mx/>]

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

PostgreSQL es un administrador de Bases de Datos basado en el modelo relacional, aunque incorpora algunos conceptos del modelo Orientado a Objetos, tales como la herencia. Usa un subconjunto amplio de comandos SQL como lenguaje de consulta y está implementado siguiendo la arquitectura cliente-servidor. PostgreSQL ofrece gran variedad de herramientas y librerías para acceder a las Bases de Datos. En su versión 6.3.2, existen interfaces para los siguientes lenguajes:

- C
- C++
- Java (a través de JDBC)
- Perl
- Python
- TCL

2.3 Fundamentos de la Arquitectura del Sistema PostgreSQL

PostgreSQL utiliza el modelo cliente - servidor conocido como proceso por usuario, una sesión en postgresQL que consiste en los siguientes procesos.

- ? Un proceso demonio supervisor (**postmaster**)
- ? La aplicación sobre la que trabaja el usuario (frontend por ejemplo el programa psql)
- ? Uno o más servidores de base de datos en segundo plano (backend el mismo proceso postgres)

Un proceso en el servidor, el cual maneja archivos de la base de datos, acepta conexiones a la base de datos de aplicaciones ejecutadas por el cliente, realizando instrucciones sobre la base de datos hechas por el

cliente. El programa del servidor de la base de datos es llamado **postmaster**.

El **postmaster** es el proceso que actúa como una puerta de control para las peticiones al sistema PostgreSQL. Las aplicaciones frontend se conectan al **postmaster**, que mantiene registros de los errores del sistema y de la comunicación entre los procesos backend. El **postmaster** puede aceptar varios argumentos desde la línea de órdenes para poner a punto su comportamiento. Sin embargo, el proporcionar argumentos es necesario sólo si se intenta trabajar con varios sitios o con uno que no se ejecuta a la manera por defecto.

Un único **postmaster** controla una colección de bases de datos dadas en un único host. Debido a esto una colección de bases de datos se suele llamar un sitio. Las aplicaciones de frontend que quieren acceder a una determinada base de datos dentro de una instalación hacen llamadas a la librería. La librería envía peticiones de usuario a través de la red al **postmaster**, el cual en respuesta inicia un nuevo proceso en el servidor (backend)

Una aplicación cliente usuario realiza una operación en la base de datos, siendo esta aplicación muy diversa pudiendo ser una herramienta de texto, una aplicación gráfica o una conexión al servidor web de la base de datos para poder de esta aplicación desplegar paginas web.

[Referencia en Internet

<http://www.phpfreaks.com/postgresqlmanual/page/tutorial-arch.html>

<http://es.tldp.org/Postgresql-es/web/navegable/tutorial/arch.html>

<http://developer.postgresql.org/pdf/transactions.pdf>]

PostgreSQL system overview

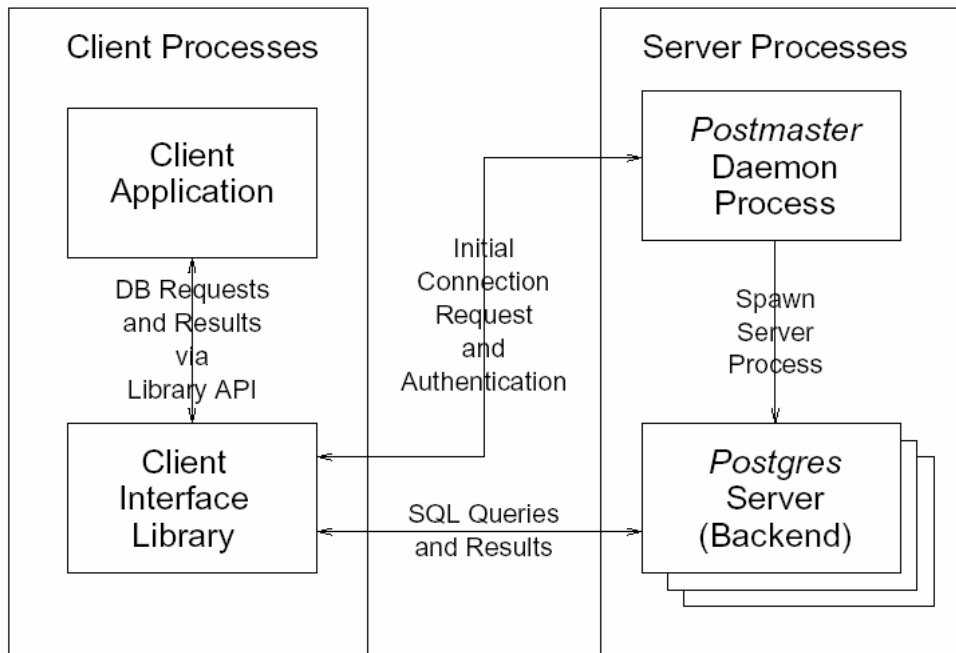


Tabla 2.1 Presentación de la arquitectura del sistema PostgreSQL.

En una típica aplicación cliente/servidor, el cliente y el servidor pueden estar en diferentes hosts, en ese caso la comunicación puede realizarse sobre una conexión de red TCP/IP.

El servidor PostgreSQL puede manejar múltiples conexiones concurrentes de los usuarios. Por esa razón empieza un nuevo proceso por cada conexión que se ejecute, en ese punto el cliente y el nuevo proceso de comunicación en el servidor están sin intervención por el proceso **Postmaster**, Por lo tanto el postmaster esta siempre corriendo, esperando las conexiones de los clientes, mientras que el cliente y proceso del servidor van y viene.

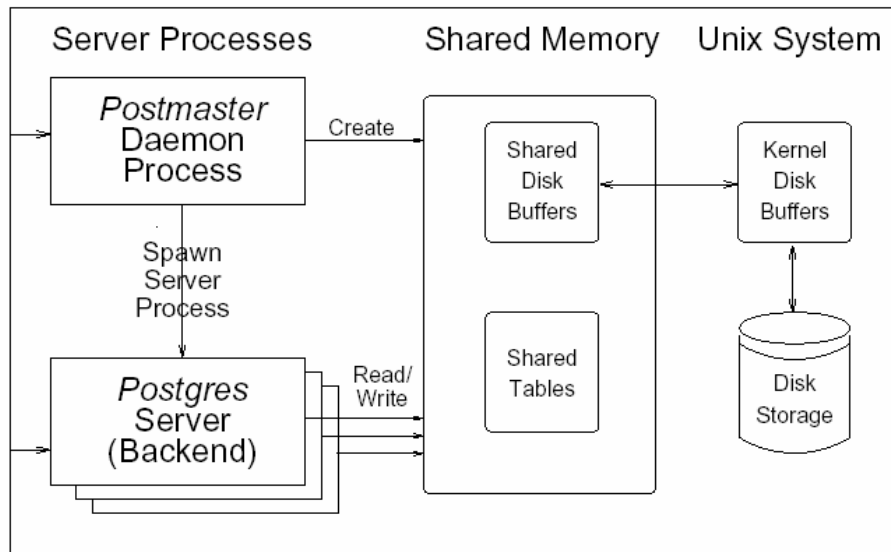


Tabla 2.2 Presentación general del sistema PostgreSQL.

Referencias de la tablas en Internet

<http://developer.postgresql.org/pdf/transactions.pdf>

2.4 Características de PostgreSQL

PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo, además proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente lista da un amplio panorama de las características de las bases de datos.

DBMS Objeto-Relacional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arrays.

Altamente Extensible

PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.

Soporte SQL Comprensivo

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

Integridad Referencial

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

API Flexible

La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

Lenguajes Procedurales

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

MVCC

MVCC, o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Si alguna vez ha usado algún DBMS con capacidades SQL, tal como MySQL o Access, probablemente habrá notado que hay ocasiones en las que una lectura tiene que esperar para acceder a información de la base de datos. La espera está provocada por usuarios que están escribiendo en la base de datos. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros.

Mediante el uso de MVCC, PostgreSQL evita este problema por completo. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

Cliente / servidor

PostgreSQL usa una arquitectura proceso-por-usuario cliente / servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Write Ahead Logging (WAL)

La característica de PostgreSQL conocida como *Write Ahead Logging* incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.

[Referencia en Internet

<http://www.sobl.org/traduccion/practical-postgres/node19.html>]

2.5 Seguridad en PostgreSQL

La evaluación de este punto es uno de los más importantes en la interconexión de la Web con Bases de datos. A nivel de una red local, se puede permitir o impedir, a diferentes usuarios el acceso a cierta información, pero en la red mundial de Internet se necesita de controles más efectivos en este sentido, ante posible espionaje, copia de datos, manipulación de éstos, etc.

La identificación del usuario es una de las formas de guardar la seguridad. Las identidades y permisos de usuarios están definidas en los Archivos de Control de Acceso.

Pero la seguridad e integridad total de los datos puede conservarse, permitiendo el acceso a distintos campos de una base de datos, solamente a usuarios autorizados para ello.

En este sentido, los datos pueden ser presentados a través de la Web de una forma segura, y con mayor impacto en todos los usuarios de la red mundial.

La seguridad de la base de datos está implementada en varios niveles:

- ? Protección de los archivos de la base de datos. Al estar protegidos contra escritura los archivos almacenados en la base de datos, solo el superusuario postgres será el único que puede modificarlos
- ? Las conexiones de los clientes al servidor de la base de datos están permitidas, por defecto, únicamente mediante sockets Unix locales y no mediante sockets TCP/IP. Ha de arrancarse el demonio con la opción `-i` para permitir la conexión de clientes no locales.
- ? Las conexiones de los clientes se pueden restringir por dirección IP y/o por nombre de usuario mediante el archivo `pg_hba.conf` situado en `PG_DATA`.
- ? Las conexiones de los clientes pueden ser autenticadas mediante otros paquetes externos.
- ? A cada usuario de PostgreSQL se le asigna un nombre de usuario y (opcionalmente) una contraseña. Por defecto, los usuarios no tienen permiso de escritura a bases de datos que no hayan creado.
- ? Los usuarios pueden ser incluidos en *grupos*, y el acceso a las tablas puede restringirse en base a esos grupos.

Autenticación de Usuarios

Autenticación es el proceso mediante el cual el servidor de la base de datos y el postmaster se aseguran de que el usuario que está solicitando acceso a la base de datos es en realidad quien dice ser. Todos los usuarios que quieren utilizar Postgres se comprueban en la tabla `pg_user` para asegurarse que están

autorizados a hacerlo. Actualmente, la verificación de la identidad del usuario se realiza de distintas formas:

Desde el shell del usuario

Un demonio que se lanza desde la shell del usuario anota el id original del usuario antes de realizar un `setuid` al id del usuario `postgres`. El id original del usuario se emplea como base para todo tipo de comprobaciones.

Desde la red

Si Postgres se instala como distribuido, el acceso al puerto TCP del `postmaster` está disponible para todo el mundo. El Administrador de la Base de Datos configura el archivo `pg_hba.conf` situado en el directorio `PG_DATA` especificando el sistema de autenticación a utilizar en base al equipo que realiza la conexión y la base de datos a la que se conecta. La autenticación basada en equipos no es perfecta incluso en los sistemas Unix. Es posible, para determinados intrusos, enmascarar el equipo de origen.

Para la integración de Bases de datos con la Web es necesario contar con una interfaz que realice las conexiones, extraiga la información de la base de datos, le dé un formato adecuado de tal manera que puede ser visualizada desde un browser de la Web, y permita lograr sesiones interactivas entre ambos, dejando que el usuario haga elecciones de la información que requiere.

Control de Acceso

Postgres proporciona mecanismos para permitir a los usuarios limitar el acceso que otros usuarios tendrán a sus datos.

Super Usuario de la Base de Datos

El Super Usuario de la base de datos (es quien tienen el campo `pg_user.usesuper` activado) ignoran todos los controles de acceso descritos anteriormente con dos excepciones: las actualizaciones del catálogo del sistema no están permitidas si el usuario no tiene el campo `pg_user.secatupd` activado, y nunca se permite la destrucción del catálogo del sistema (o la modificación de sus estructuras).

Privilegios de acceso

El uso de los privilegios de acceso para limitar la lectura, escritura y la

[Referencias en Internet

<http://es.tldp.org/Postgresql-es/web/navegable/admin/security.html>]

puesta de reglas a las clases se trata en *grant / revoke()*.

Borrado de clases y modificación de estructuras.

Los comandos que borran o modifican la estructura de una clase, como **alter**, **drop table**, y **drop index**, solo funcionan con el propietario de la clase. Como hemos dicho antes, estas operaciones no están permitidas *nunca* en los catálogos del sistema.

CAPÍTULO 3

Selección y Evaluación de herramientas

3.1 PHP

PHP fue originalmente llamado (Personal Home Page) y creado en 1994 por *Rasmus Lerdorf* como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. Así su utilidad y capacidad creció y comenzó a utilizarse para situaciones profesionales que su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI

La siguiente contribución al lenguaje fue hecha a mediados de 1997 cuando se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3. Actualmente PHP se encuentra en su versión 4, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su rapidez gracias a que primero se compila y luego se ejecuta, y antes se ejecutaba mientras se interpretaba el código-, su mayor independencia del servidor web creando versiones de PHP nativas para En el último año, el número de servidores que utilizan PHP se ha disparado, logrando situarse cerca de los 5 millones de sitios y 800.000 direcciones IP, lo que le ha convertido a PHP en una tecnología popular. Esto es debido, entre otras razones, a que PHP es el complemento ideal para que Linux-Apache sea compatible con la programación del lado del servidor de sitios web.

Referencia en Internet

[<http://www.desarrolloweb.com/articulos/436.php?manual=12>
<http://cr2.php.net/history>.]

Bibliografía

Larry Ullman
PHP and MySQL for Dynamic Web Sites Visual QuickPro Guide
Editorial Peachpit Press

Luke Welling, Laura Thompson
PHP and MySQL Web Development
Segunda Edición, Editorial Sams Publishing

PHP es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como C o C++, es rápido a pesar de ser interpretado, multiplataforma y dispone de una gran cantidad de librerías que facilitan muchísimo el desarrollo de las aplicaciones.

Decir que PHP está embebido dentro de HTML significa que PHP puede ser interpretado dentro del código HTML el cual hace sitios dinámicos Web mas accesibles

PHP se ejecuta en el servidor y no en la maquina del usuario, es multiplataforma esto quiere decir que PHP corre en muchos sistemas operativos tales como Unix, Linux, Macintosh, Windows, etc.

Es un lenguaje de programación que está muy orientado al desarrollo de páginas web dinámicas, una página web dinámica es un documento que se regenera en cada momento que la página es visitada. El funcionamiento de PHP es de la siguiente manera, cuando pedimos a nuestro servidor web una página PHP, que no es más que un programa PHP embebido dentro del código HTML, antes de enviar dicha página al cliente se la pasa al interprete de PHP. Este la interpreta y es el resultado de esta interpretación del programa PHP, contenido en la página PHP, lo que termina llegando al cliente.

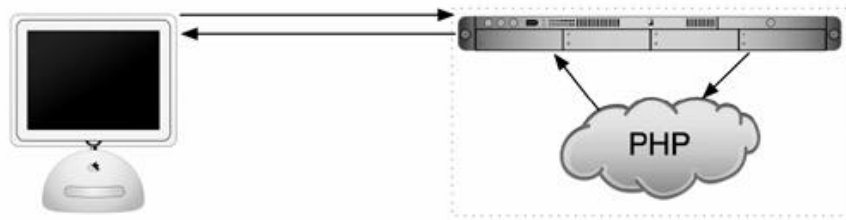
3.1.1 *Funcionamiento de PHP*

PHP es un lenguaje del lado del servidor, lo que significa que el código escrito en PHP reside sobre el host de una computadora que da servicio a la petición de los visitantes a las paginas Web

Cuando un visitante va al sitio Web escrito en PHP el servidor lee el código PHP y entonces se procesa de acuerdo a las direcciones escritas.

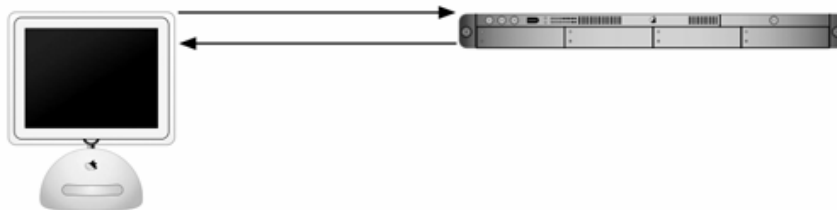
En la siguiente figura se muestra como el código PHP le dice al servidor las instrucciones para que éste envíe los datos apropiados del código HTML al Web browser. Como PHP ajusta dentro del modelo cliente servidor cuando hay una petición de un usuario a una pagina Web

Figure 3.1 How PHP fits into the client-server model when a user requests a Web page.



La diferencia de una página estática HTML es que cuando una petición es hecha el servidor simplemente envía los datos de la pagina HTML al Web browser y ahí el servidor no interpreta lo ocurrido. La figura 3.2 muestra el proceso en el que se ven involucrados con una petición de una página estática HTML

Figure 3.2 The process involved with a request for a static HTML page.



[Referencias bibliograficas de las figura 3.1 y 3.2
Larry Ullman
PHP and MySQL for Dynamic Web Sites QuickPro Guide]

FUNCIONAMIENTO DE PHP CON EL SERVIDOR

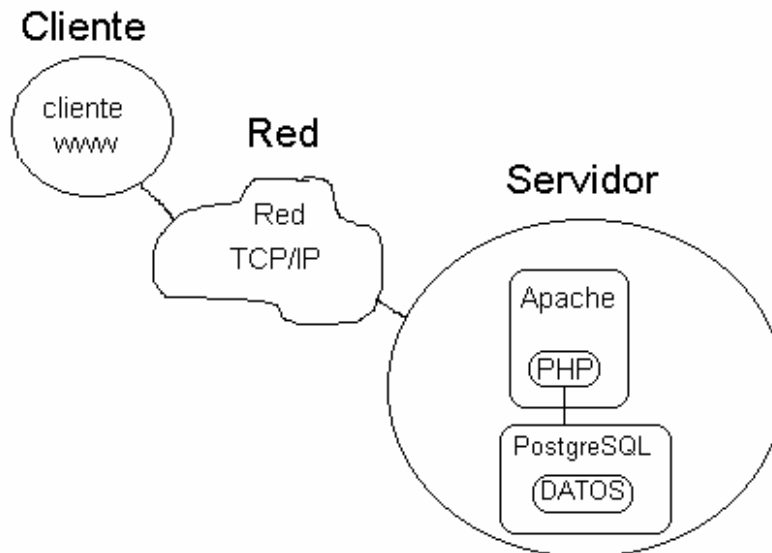


Fig 3.1 Interacción entre PHP el cliente y el servidor web

3.1.2 Definición de PHP

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje "open source" interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. El código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP.

3.1.3 Lo que se puede realizar con PHP

PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies.

PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros. PHP tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía web para una base de

[Referencia de la figura en Internet
<http://www.angelfire.com/al/acs/bd.html>]

datos es una tarea simple con PHP. Las siguientes bases de datos están soportadas actualmente:

- ? Adabas D
- ? Ingres
- ? Oracle (OCI7 and OCI8)
- ? dBase
- ? InterBase
- ? Ovrimos
- ? FrontBase
- ? **PostgreSQL**
- ? mSQL
- ? Hyperwave
- ? Direct MS-SQL
- ? Sybase
- ? MySQL
- ? Velocis
- ? Informix
- ? ODBC
- ? Unix dbm

PHP también tiene soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros.

PHP es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como C o C++, es rápido a pesar de ser interpretado, multiplataforma y dispone de una gran cantidad de librerías que facilitan muchísimo el desarrollo de las aplicaciones.

PHP es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar ni licencias, ni estamos limitados en su distribución y, podemos ampliarlo con nuevas funcionalidades si así se quiera.

La serie está orientada al uso conjunto de Apache y PHP, aunque PHP se puede compilar como un ejecutable que puede ser utilizado como CGI-bin. De esta forma, cualquier servidor de web que soporte CGI podría ejecutar las páginas PHP. La ventaja que tiene Apache frente a estos es que podemos compilar el interprete PHP como un módulo de Apache, por lo que la velocidad de ejecución de estas páginas PHP es muy superior, así como el consumo de recursos al cargarse el interprete de PHP una sola vez en memoria.

[Referencia en Internet
<http://php.caos.cl/manual/es/introduction.php>]

PHP es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como C o C++, es rápido a pesar de ser interpretado, multiplataforma y dispone de una gran cantidad de librerías que facilitan muchísimo el desarrollo de las aplicaciones.

3.1.4 Seguridad que se tiene con PHP

PHP es un potente lenguaje y el interprete, tanto incluido en el servidor web como modulo o ejecutado como un binario CGI, puede acceder a archivos, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor web sea inseguro por defecto. PHP ha sido diseñado específicamente, para ser un lenguaje mas seguro para escribir programas CGI, que Perl o C y con la correcta selección de las opciones de configuración del tiempo de compilación y ejecución se consigue la exacta combinación de libertad y seguridad que se necesita.

Ya que existen diferentes modos de utilizar PHP, existen multitud de opciones de configuración que permiten controlar su funcionamiento. Una gran selección de opciones garantiza que se pueda usar PHP para diferentes usos, pero también significa que existen combinaciones de estas opciones y configuraciones del servidor que producen instalaciones inseguras.

3.2 Apache

Apache surgió a partir del servidor de HTTP más famoso desarrollado por Rob McCool En el Centro Nacional de Aplicaciones para Súper cómputo (NCSA) de la Universidad de Illinois. Desde entonces se convirtió en un poderoso rival de todos los servidores Unix utilizados hasta la fecha por su eficiencia, funcionalidad y rapidez. Es por ello que se conoce como el rey de los servidores *Web*. Se desarrolla de forma estable y segura gracias a la cooperación y los esfuerzos de un grupo de personas conocidas como grupo Apache (*Apache Group*), los cuales se comunican a través de Internet y del *Web*. Juntos se dedican a perfeccionar el servidor y su documentación regidos por la ASF (*Apache Software Foundation*).

Las características principales del servidor Apache son:

- ? Es un servidor *Web* potente, flexible y ajustado al HTTP/1.1
- ? Es altamente configurable y extensible.
- ? Puede ser ajustado a través de la definición de módulos empleando su propio API (*Application Programming Interface*).

- ? Provee todo su código fuente de forma libre y se distribuye sin restricciones de licencia.
- ? Corre en diversas plataformas tales como: Windows 9x/NT, Macintosh, Novell NetWare, OS/2, Linux y la mayoría de los Unix existentes: IRIX, Solaris, HPUX, SCO, FreeBSD, NetBSD, AIX, Digital Unix, etc.
- ? Activamente se esta desarrollando. Se desarrolla de forma acelerada estimulando la retroalimentación desde sus usuarios a través de nuevas ideas, reportes de errores y parches.
- ? Apache significa "*A PAtCHy sErver*", o sea se basa en un código y un conjunto de ficheros "parches". Otros desarrolladores relacionan su nombre con el de las tribus nativas americanas de Apaches.
- ? Implementa muchas posibilidades frecuentemente demandadas, tales como:
 - o **Bases de datos DBM para autenticación.**
Permiten establecer fácilmente la protección de documentos a través de *passwords* para una gran cantidad de usuarios sin dañar el funcionamiento del servidor.
 - o **Respuestas adaptables a los errores o problemas.**
Se pueden definir ficheros o *scripts* de tipo CGI los cuales son regresados por el servidor en respuesta a errores internos o en las solicitudes realizadas.
 - o **Directiva para definir múltiples índices.**
Permite que se le diga que Directorio index regrese: index.html, index.cgi, index.php O default.html, donde el servidor enviará el que se le dijo.
 - o **Ilimitadas y flexibles posibilidades de redireccionamiento y definición de alias para los URLs.**
Apache no tiene un límite establecido para definir alias y redireccionamientos que pueden ser declarados en sus ficheros de configuración.
 - o **Negociación del contenido de las respuestas.**
Apache es capaz de ofrecer la mejor representación de la información accedida de acuerdo con las capacidades del cliente solicitante.
 - o **Soporte de *hosts* virtuales.**
Es la habilidad del servidor de distinguir entre los pedidos hechos a diferentes direcciones IP o nombres de dominio definidos en la misma máquina.
 - o **Configuración flexible de las trazas generadas.**
Es posible adaptar el formato de las trazas obtenidas así como redireccionarlas a través de tuberías (Unix) en aras de filtrarlas. De esta forma se puede lograr por ejemplo dividir dinámicamente las trazas de los *hosts* virtuales en distintos ficheros.

El paquete de la distribución Red Hat que contiene la implementación del servidor Apache para Linux se nombra `apache`. También se dispone de un manual del mismo en el paquete `apache-manual`. Existe una aplicación con interfaz gráfica en el

paquete `apacheconf` que permite configurar al Apache con las limitaciones propias de dichas interfaces.

El Apache en Red Hat se ejecuta a través de un *daemon* (demonio) llamado `httpd` que se manipula utilizando el *script* de inicio del mismo nombre en `/etc/rc.d/init.d/`. Por tanto la forma más sencilla de iniciar, detener, conocer el estado o indicar que relea su configuración al *daemon* es como se muestra en los ejemplos:

```
# service httpd stop
Stopping httpd:          [ OK ]

# service httpd start
Starting httpd:         [ OK ]

# service httpd status
httpd (pid 6973 6972 6971 6970 6969 6968 6967 6966 6963) is running...

# service httpd reload
Reloading httpd:      [ OK ]
```

Bibliografía

Korry Douglas, Susan Douglas
PostgreSQL
Editorial Sams Publishing Pub 13 de Febrero del 2003
Pages: 816

Larry Ullman
PHP and MySQL for Dynamic Web Sites Visual QuickPro Guide
Editorial Peachpit Press

Luke Welling, Laura Thompson
PHP and MySQL Web Development,
Segunda Edición, Editorial Sams Publishing

CAPITULO 4

Diseño de una Base de Datos con PostgreSQL

4.1 DISEÑO DE LA BASE DE DATOS “PCIF”

Para el diseño de la base de datos en PostgreSQL hay que considerar los siguientes puntos para su elaboración.

- ✍ Encontrar las entidades importante
- ✍ Identificar las asociaciones entre las entidades.
- ✍ Identificar los atributos de las entidades y asociaciones.
- ✍ Determinar los dominios de los atributos.
- ✍ Determinar las llaves primarias y foráneas.
- ✍ Dibujar el diagrama entidad-relación.

Para el diseño de la base de datos, en los que se involucran los pasos anteriores nos lleva al esquema estructural lógico de la base de datos que es el Diagrama Entidad-Relación.

Para la construcción de la base de datos de la biblioteca, se tiene los siguientes elementos:

- ✍ Artículo
- ✍ Autor y coautor
- ✍ Palabras clave del artículo
- ✍ Editorial de publicación
- ✍ Procedencia del investigador

4.1.1 PLANTEAMIENTO DEL PROBLEMA

El problema es de que no se cuenta con una base de datos para la producción científica de Instituto de Física y por ello se desea crear la base de datos y que se consulte a través de Internet, y debido a los recursos proporcionados se implementara este sistema en la base de datos **PostgreSQL** utilizando el lenguaje de programación **PHP** para la construcción de las interfaces que se mostraran en Internet.

Para darnos una idea de cómo resolver el problema, hay que analizar paso a paso las necesidades que se pidieron para la realización de este sistema. Para ello hay que considerar la producción científica de los investigadores del Instituto de Física se buscarán por:

1. autor,
2. titulo del artículo, o

3. por palabra clave,

La primera interfaz del sistema ver fig 6.1.1.a mostrará una página común y corriente de HTML que mostrará un formulario con tres campos para poder escribir la petición estos campos son etiquetas `<input type=text>` con sus respectivas variables para poder realizar la búsqueda del artículo esperado.

Un formulario no es más que un cuestionario insertado en la página web mediante el cual la persona que accede a la misma puede enviar la información que es necesaria para las tres búsquedas que nos lleva a la información del artículo solicitado.

La estructura general de un formulario es:

```
<FORM ACTION="programa.php" METHOD="POST" ENCTYPE="TEXT/PLAIN">  
...  
cuerpo del formulario.  
...  
Botones de envío y borrado.  
</FORM>
```

Atributo Parámetros Descripción

ACTION= "programa.php" Indica la acción a efectuar; en este caso, que los datos se envíen al programa que esta hecho en PHP.

METHOD= "POST" Indica que los datos sean enviados nada más pulsar el botón de envío.

ENCTYPE= "TEXT/PLAIN" Indica que los datos se reciban como fichero de texto, legibles y sin codificar.

El formato general de la etiqueta que permite la introducción de datos es el siguiente:

? **<INPUT TYPE="tipo_datos" NAME="campo" VALUE="texto">**

Atributo Parámetros Descripción

TYPE= "tipo_datos" Indica el tipo de datos que se van a introducir.

NAME= "campo" Indica el nombre del campo que recibe los datos.

VALUE= "texto" Indica el nombre asociado a un elemento.

Para el programa de PHP se utilizó el siguiente atributo:

Se empleó la etiqueta:

? **<INPUT TYPE="TEXT" NAME="nombre">**

En la parte inferior de esta interfaz se tiene los **BOTONES DE ENVÍO Y BORRADO**

Son los botones que permiten enviar los datos del formulario o borrarlos, respectivamente. Para ello se emplean las siguientes etiquetas:

? **<INPUT TYPE="SUBMIT" VALUE="texto">**

Crea el botón de envío, dentro del cual aparece el texto indicado en texto. Al pulsarle, los datos son enviados.

```
<INPUT TYPE="SUBMIT" VALUE="Enviar">
```

<INPUT TYPE="RESET" VALUE="texto">

Crea el botón de borrado, dentro del cual aparece el texto indicado en texto. Al pulsarle, se borran los datos introducidos, apareciendo los valores establecidos por defecto al definir el formulario.

```
<INPUT TYPE="RESET" VALUE="Borrar">
```

Para la elaboración del Diagrama Entidad Relación E-R hay que especificar las partes que lo conforman con los datos con los que se cuenta es decir la información que se tiene para la elaboración del sistema de la biblioteca del Instituto de Física de la UNAM, ya que las partes que conforman el diagrama son: entidades, atributos, relaciones.

4.2 El modelo entidad-relación

El modelo entidad-relación es una herramienta de ayuda utilizado para construir el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976.[CHE,1976] El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y semánticas.

El modelo de datos entidad-relación (E-R) está basado en una percepción de un mundo real que consta de una serie de objetos reales llamados *entidades*, y de relaciones entre entidades. Este esquema representa la estructura lógica general de la base de datos.

Una **entidad** es un objeto que existe y que puede distinguirse de otros objetos. La distinción se logra relacionando cada objeto con una serie de **atributos** que lo describen. Y una **relación** es una asociación entre varias entidades.

La *cardinalidad* con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es *obligatoria (total)* si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es *opcional (parcial)*.

Los *atributos* representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante elipses que cuelgan de las entidades o relaciones a las que pertenecen. Ver figura 4.6

4.2.1 Construcción del modelo Entidad-Relación y las relaciones

4.2.1.2 Identificar las entidades

Una *entidad* es un objeto que existe y puede distinguirse de otros objetos. Para identificar las entidades de la base de datos de la biblioteca, es necesario tomar en cuenta la información que se tiene y los requisitos solicitados, y al tener presente que la búsqueda tiene que ser por investigador, artículo y palabra del abstract de una revista publicada, entonces hay que buscar las entidades que van a identificar de manera única una información determinada.

Se cuenta con la siguiente información para construir la base de datos de las revistas publicadas por investigadores del Instituto de Física.

Los artículos publicados cuya identificación es su título, volumen, año de publicación número, paginas donde se encuentra delimitado el artículo, su palabra clave, su editorial, sus escritores de la publicación,
Recuperando la información por entidades:

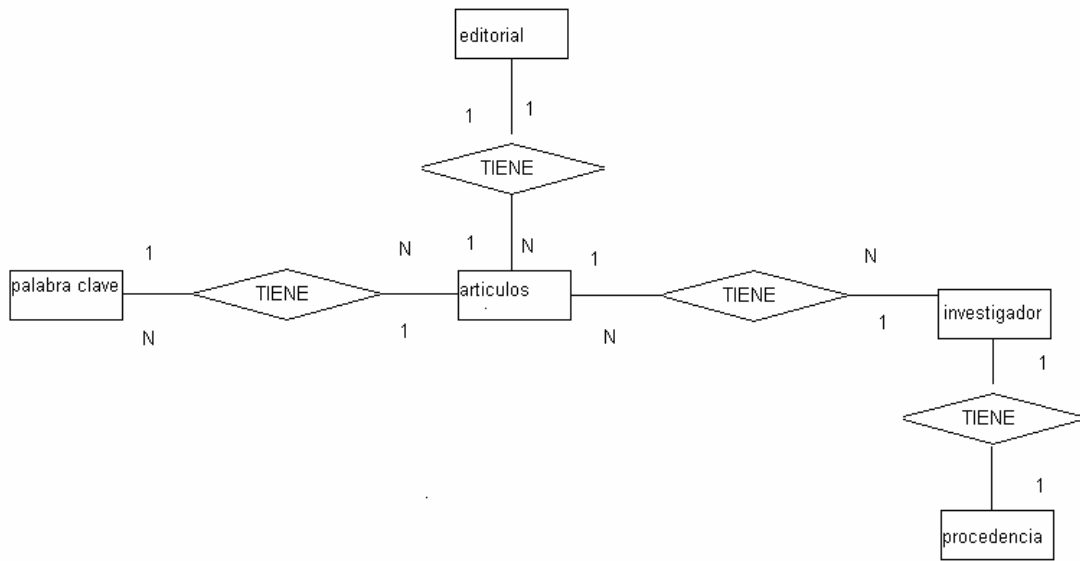


Figura 4.1 Entidades y asociaciones de la base de datos

Entidades:

- ARTÍCULOS
- PALABRA CLAVE.
- INVESTIGADOR
- PROCEDENCIA DEL INVESTIGADOR
- EDITORIAL DE LA REVISTA

Atributos

Artículos:: nombre del artículo, número, volumen, páginas, año de publicación, editorial, abstract.

Investigador :: nombre del investigador del artículo, procedencia.

palabra : palabra clave del artículo

editorial : nombre de la editorial.

procedencia: procedencia

4.2.1.3 Identificar las asociaciones entre las entidades

Las asociaciones son:

1. **artículo_investigador::**

- ✍ Un Investigador escribe varios artículos
- ✍ Un articulo tiene varios Investigadores



Figura 4.2 Asociación palabra clave con los artículos

2. palabra_artículo ::

- ✍ Un artículo tiene varias palabras claves.
- ✍ Una palabra clave tiene varios artículos publicados.



Figura 4.3 Asociación de investigador con los artículos

3. artículo editorial



Figura 4.4 Asociación palabra clave con los artículos

4. procedencia_investigador



Figura 4. Asociación palabra clave con los artículos

4.2.1.4 Identificar los atributos de las entidades y de las relaciones

Conforme se van identificando los atributos, se les asignan nombres que tengan significado. De cada atributo se debe anotar la siguiente información:

- * Nombre y descripción del atributo.
- * Tipo de dato y longitud.
- * Valores por defecto del atributo (si se especifican).
- * Si el atributo siempre va a tener un valor (si admite o no nulos).
- * Si el atributo es compuesto y, en su caso, qué atributos simples lo forman.

ARTICULOS

Column	Type	Modifiers
<u>ide artículo</u>	integer	not null
<u>nombre_artículo</u>	character varying(700)	not null
<u>número</u>	character varying(20)	
<u>volumen</u>	character varying(20)	
<u>páginas</u>	character varying(20)	
<u>año</u>	integer	
<u>ideditorial</u>	integer	not null
<u>abstract</u>	integer	not null
Primary key: <u>ide_articulo</u>		

INVESTIGADOR

Column	Type	Modifiers
<u>idinvestigador</u>	integer	not null
<u>nombre_investigador</u>	character varying(300)	
<u>idprocedencia</u>	integer	not null

Primary key: | |
 idinvestigador

EDITORIAL

Column	Type	Modifiers
<u>ideditorial</u>	integer	not null
Nombre_editorial	character varying(300)	
Primary key: ideditorial		

PALABRA

Column	Type	Modifiers
<u>idpalabra</u>	integer	not null
palabra	character varying(100)	
Primary key: idpalabra		

PROCEDENCIA

Column	Type	Modifiers
<u>idprocedencia</u>	integer	not null
procedencia	character varying(500)	
Primary key: idprocedencia		

Se quiere el acceso directo por investigador, titulo de la publicación y por palabra clave del artículo, además solo se puede mostrar el abstract del articulo, por lo que con esta petición se encuentra el atributo abstract que pertenece a la entidad artículo.

4.2.1.5 Determinar los dominios de los atributos.

El dominio de un atributo es el conjunto de valores que puede tomar el atributo. Por ejemplo el dominio de los artículos son las secuencias de los caracteres permitidos.

Cada atributo tiene un conjunto de valores asociados denominado *dominio*. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio

4.2.1.6 Determinar las llaves primarias y foráneas

La generalización es el resultado de la unión de dos o más conjuntos de entidades (de bajo nivel) para producir un conjunto de entidades de más alto nivel.

Al clasificar estas búsquedas con artículo palabra y con artículo investigador.

Artículo_palabra se describe mediante un conjunto de atributos que incluyen todos los atributos del conjunto de entidades artículo y otros adicionales, las entidades artículo_palabra se describen además con el atributo idart_pal.

4.2.1.7 Dibujar el diagrama entidad-relación.

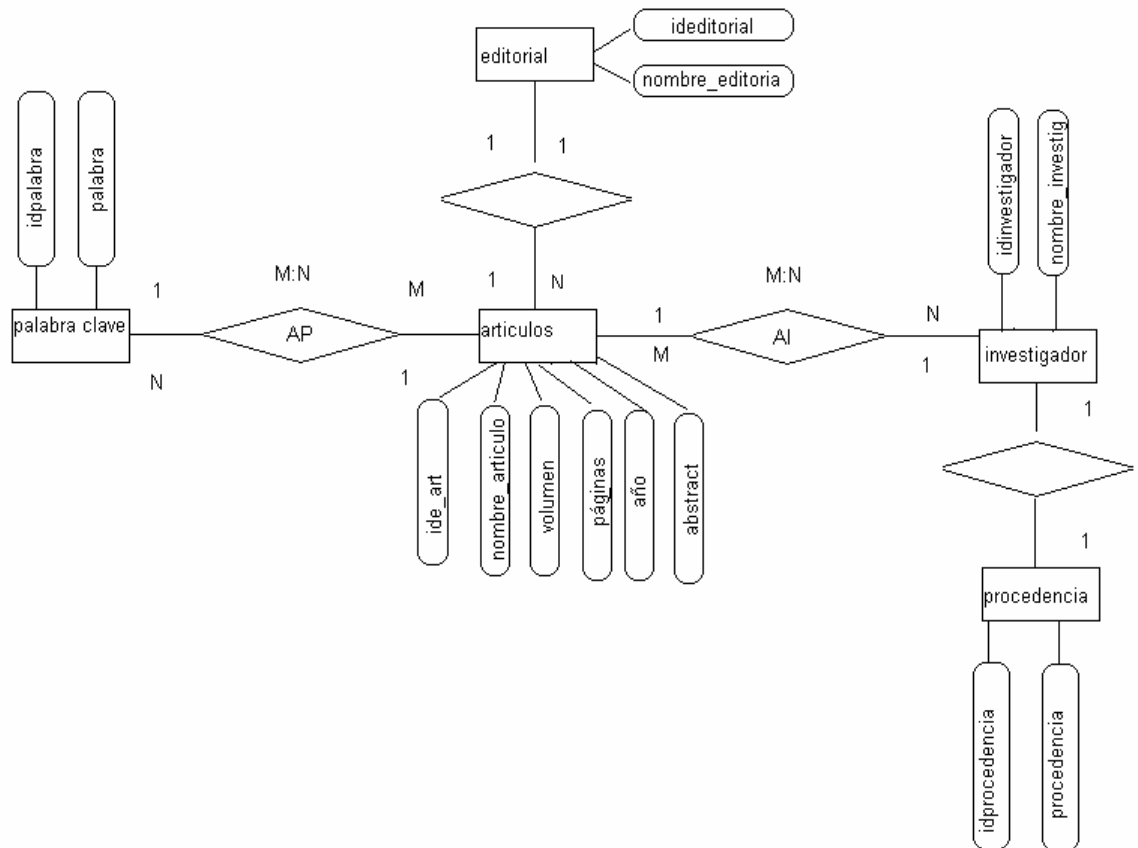
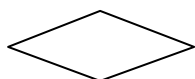
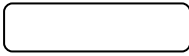


Figura 4.6. Diagrama Entidad-Relación del Sistema de Base de Datos para el Instituto de Física de la UNAM.

NOTACIÓN DEL DIAGRAMA ENTIDAD- RELACION.

 Representa la entidad

 Representa la asociación



Representa los atributos de la entidad.

Una vez identificados todos los conceptos, se puede dibujar el diagrama entidad-relación correspondiente. Se obtiene así un esquema conceptual local.

Las relaciones obtenidas a partir del M E/R son:

R_{articulo} (#ide_articulo, nombre_articulo, volumen, número, páginas, año, ideditorial, abstract)

R_{editorial} (#ideditorial, nombre_editorial)

R_{investigador} (#idinvestigador, nombre_investigador, #idprocedencia)

R_{procedencia} (#idprocedencia, procedencia)

R_{palabra_clave} (#idpalabra, palabra)

R_{AI} (#investigador,#artículo)

R_{AP} (#artículo,#palabra clave)

REFERENCIAS

<http://www3.uji.es/~mmarques/f47/apun/node84.html>

<http://techdocs.postgresql.org/tutorials/lesson1.php>

<http://www.dbinternet.com.ar/metodo.htm>

4.2.2 Creación de la base de datos

Para crear una base de datos en PostgreSQL, el que debe crear la base es el usuario postgres en nuestro servidor pero no es una regla ya que si el administrador de la base de datos quiere dar permisos a un usuario específico de crear bases de datos se lo dará al momento de que el usuario postgres crea el usuario se le va preguntado si este nuevo usuario puede crear bases, y por seguridad a los usuarios que se crearon para este sistema no se les otorgo esos permisos, debido a que el administrador de la base de datos es el superusuario postgres es éste el usuario que da permisos a otro usuario para que un usuario x tenga el privilegio de crear bases de datos.

Lo primero que hay que realizar es con la creación de nuestra base de datos que será ejecutada por el superusuario **postgres**.

Por lo tanto la instrucción que da el usuario postgres para la creación de una base de datos es: **createdb PCIF**, y tendrá como respuesta **CREATE DATABASE**, como se muestra.

```
[angie@motomorphosys angie]$su - postgres
Password:
bash-2.05a$ createdb PCIF
CREATE DATABASE
```

Y al crear la base de datos hay que crear las tablas:

```
CREATE TABLE procedencia
          (idprocedencia integer not null PRIMARY KEY,
          procedencia varchar(700));
```

```
CREATE INDEX procedencia_procedencia on procedencia (procedencia);
```

```
CREATE TABLE investigador
          (id integer not null PRIMARY KEY,
          nombre varchar(400),
          idprocedencia integer not null REFERENCES
          procedencia);
```

```

CREATE INDEX investigador_nombre on investigador (nombre);

CREATE TABLE editorial
        (ideditorial integer not null PRIMARY KEY,
        editorial varchar(350));
CREATE INDEX editorial_editorial ON editorial (titulo);

CREATE TABLE palabra
        (idpalabra integer not null PRIMARY KEY,
        palabra varchar(300));
CREATE INDEX palabra_palabra on palabra(palabra);

CREATE TABLE artículos
        (ide integer not null PRIMARY KEY,
        nombre _ articulo varchar(1000) not null,
        numero varchar(20),
        volumen varchar(20),
        abstract text);
CREATE INDEX artículos_nombre_artículo ON artículos (nombre_articulo);

CREATE TABLE art_palabra
        (ide integer not null REFERENCES artículos,
        idpalabra integer not null REFERENCES palabra);

CREATE TABLE art_invest
        (ide integer not null REFERENCES artículos,
        id integer not null REFERENCES investigador);

CREATE SEQUENCE seq_artículo;
CREATE SEQUENCE seq_investigador;
CREATE SEQUENCE seq_palabra;
CREATE SEQUENCE seq_editorial;
CREATE SEQUENCE seq_procedencia;
CREATE SEQUENCE seq_abstract;

```


Otorgar privilegios a la Base de Datos "PCIF"

La base de datos se llama
PCIF, con las siguientes tablas
abstract, art_invest, art_palabra,
artículos, investigador, palabra,
procedencia, editorial

Para dar privilegios a la base de datos se necesita crear al menos un usuario para que acceda a mi base de datos, y como el usuario al que le daré permisos es al usuario apache ya que es el servidor web del sistema Linux así que lo creo con el nombre de apache por supuesto siendo el usuario postgres Debido a que postgres es el administrador de la base de datos, entonces para crear un usuario a nuestra base de datos hay que ejecutar la siguiente instrucción "createuser". Como se muestra a continuación.

```
angie@motomorphosys angie$ su - postgres
```

```
Password:
```

```
bash-2.05a$ createuser apache
```

```
Shall the new user be allowed to create databases? (y/n) n
```

```
Shall the new user be allowed to create more new users? (y/n) n
```

```
CREATE USER
```

De esta forma ya tengo el usuario apache que no tiene permisos de crear base de datos ni tampoco de que pueda crear usuarios. Pero como el usuario que corre el servidor web es "apache" entonces es al que se le da privilegios para que se pueda acceder a la base de datos REVISTAS en el momento que el script corra en el servidor web y por este motivo se la da permisos con la instrucción "grant" como se aprecia en las siguientes líneas.

```
angie@motomorphosys angie$ su - postgres
```

```
Password:
```

```
bash-2.05a$
```

```
bash-2.05a$ psql revistas
```

```
revistas => GRANT SELECT ON investigador
```

```
revistas => TO apache;
```

```
revistas => \z
```

```
revistas => \q
```

```
bash-2.05a$ exit
```

Y al estar como el superusuario de la base de datos, y para crear tablas, ejecutar inserts, es necesario saber como trabajar con el monitor psql, ya que es este permite interactuar con los comandos de SQL en la base de datos "Revistas".

monitor de Postgresql

Y de esta manera al tener los permisos el usuario apache con la tabla investigador, hay que darle permisos a las demás tablas.

Para que en nuestro servidor tenga apache permisos de ejecución de todas las tablas en lo personal escribe las instrucciones para todas las tabla en un archivo.

El archivo que tiene las instrucciones GRANT:

```
GRANT ALL ON abstract TO APACHE;  
GRANT ALL ON procedencia TO APACHE;  
GRANT ALL ON art_invest TO APACHE;  
GRANT ALL ON art_palabra TO APACHE;  
GRANT ALL ON articulos TO APACHE;  
GRANT ALL ON investigador TO APACHE;  
GRANT ALL ON palabra TO APACHE;  
GRANT ALL ON publicacion TO APACHE;  
GRANT ALL ON seq_articulo TO APACHE;  
GRANT ALL ON seq_investigador TO APACHE;  
GRANT ALL ON seq_palabra TO APACHE;  
GRANT ALL ON seq_publicacion TO APACHE;  
GRANT ALL ON seq_procedencia TO APACHE;  
GRANT ALL ON seq_abstract TO APACHE;
```

Para el caso de la inserción de los datos en las interfaces hay que dar permisos a las secuencias que son las que se incrementan con el comando "**nextval**".

Y ahora que ya se tiene el archivo grant, hay que ejecutar la instrucción

```
bash-2.05a$ psql -d revistas -f grant
```

```
GRANT  
GRANT  
GRANT  
GRANT  
GRANT
```

GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT

Esta instrucción significa conectándose a la base de datos **revistas** (-d revistas) lee lo que tiene el archivo **grant** (-f grant) y ejecuta la instrucción que tiene, de esta manera se ejecutan los permisos de las tablas al usuario apache.

Así la instrucción psql -d (nombre de la base de datos) -f (nombre del archivo donde se tienen las instrucciones que la base de datos postgresQL reconoce). Por lo que en los archivos de inserción, de actualización, solo llene las instrucciones correspondientes a las tablas con los VALUES que tienen y escribe la instrucción.

Insertando registros a la base de datos

Insertando las palabras claves en la tabla "palabras claves"

```
bash-2.05a$ psql PCIF
```

```
Welcome to psql, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
```

```
\h for help with SQL commands
```

```
\? for help on internal slash commands
```

```
\g or terminate with semicolon to execute query
```

```
\q to quit
```

```
PCIF=# INSERT INTO palabra
```

```
VALUES(nextval('seq_palabra'),'palabra clave')
```

insertando las procedencias de los investigadores en la tabla procedencia

```
PCIF=# INSERT INTO procedencia
```

```
VALUES(nextval('seq_procedencia'),'procedencia')
```

Insertando los investigadores de las revistas en la tabla "investigadores ":

```
PCIF=# INSERT INTO investigador
```

```
VALUES(nextval('seq_investigador'),'investigador')
```

Insertando las editoriales en la tabla "editorial"

```
PCIF=# INSERT INTO editorial
```

```
VALUES(nextval('seq_editorial'),'editorial')
```

Insertando los articulos en la tabla "articulos"

```
PCIF=# INSERT INTO articulos
```

```
VALUES(nextval('seq_articulos '),'articulos')
```

Definición de psql

psql es una aplicación hecha en C, que sirve como cliente, monitor y front-end hacia el motor de Postgresql. Le permite teclear comandos SQL interactivamente, enviarlas a Postgres y visualizar los resultados de la consulta. Las sentencias SQL pueden leerse desde un archivo, además de nos provee de varios metacomandos y varias características similares a un shell, lo cual nos permite automatizar una gran variedad de tareas

Capítulo 5

MANUAL DE INSTALACION DE POSTGRESQL, APACHE Y PHP

El servidor del Instituto de Física tiene el sistema operativo Linux corriendo la versión (Red Hat 7.3) teniendo el software PHP, PostgreSQL y Apache, por lo que únicamente se tuvo que levantar sus correspondientes servicios en el servidor.

Debido a que en el servidor **motomorphosys** de la biblioteca tiene el software que se necesita hay que inicializarlos para su uso.

Por ello al levantar los servicios de PostgreSQL, y Apache, hay que dar las siguientes instrucciones para estos dos servicios.

5.1 Iniciación de PostgreSQL para su uso en el sistema.

Ahora tenemos que decirle al sistema donde poder encontrar las librerías necesarias, para ello actualizamos el archivo `/etc/ld.so.conf`:

Actualización del archivo de configuración

```
angie@motomorphosys angie$ su - postgres
bash-2.05a$ echo /usr/local/pgsql/lib >> /etc/ld.so.conf
bash-2.05a$ /sbin/ldconfig.
bash-2.05a$ exit
```

Una vez que hemos terminado de configurar nuestro sistema, tenemos que inicializarlo y arrancarlo:

5.1.1 Iniciación y arranque de la Base de Datos.

Para inicializar nuestra base de datos PostgreSQL hay que conectarse como el usuario postgres (que es a quien es el administrador DBA) y arranca el demonio de postgres con el comando

```
"nohup postmaster >logfile 2>&1 </dev/null &".
```

```
angie@motomorphosys angie$ su - postgres
Password:
```

```
bash-2.05a$ initdb
bash-2.05a$ cd
bash-2.05a$ $ nohup postmaster -i > pgserver.log 2>&1 &
bash-2.05a$ $ exit
```

Y debido a que en el servidor, ya se tiene la base de datos PostgreSQL, instalada . Pues lo que sigue es hacer un par de ajustes en la configuración para que funcione y de esta manera se pueda acceder a PostgreSQL vía PHP/web.

El *superusuario* de Postgres es el usuario llamado *postgres* que es dueño de los archivos de la bases de datos y binarios de Postgres. Como superusuario de la base de datos, puede acceder a cualquiera de los datos de forma arbitraria. Además, al superusuario de Postgres se le permite ejecutar programas de soporte que generalmente no están disponibles para todos los usuarios. Considerar que el superusuario de Postgres **no** es el mismo que el superusuario de Linux (que es conocido como *root*). El superusuario debería tener un identificador de usuario (*UID*) distinto de cero por razones de seguridad.

El *administrador de la base de datos (database administrator)* o DBA, es la persona responsable de instalar Postgres con mecanismos para hacer cumplir una política de seguridad para el servidor. El DBA puede añadir nuevos usuarios (como se explico en la página 41) y mantener un conjunto de bases de datos plantilla para usar con createdb.

El postmaster es el proceso que actúa como una puerta de control (clearing-house) para las peticiones al sistema Postgres. Las aplicaciones frontend se conectan al postmaster, que mantiene registros de los errores del sistema y de la comunicación entre los procesos backend. El postmaster puede aceptar varios argumentos desde la línea de órdenes para poner a punto su comportamiento. Sin embargo, el proporcionar argumentos es necesario sólo si se intenta trabajar con varios sitios o con uno que no se ejecuta a la manera por defecto.

[Referencia

[PostgreSQL](#)

<http://es.tldp.org/Postgresql-es/web/navegable/tutorial/terminology.html>

<http://www.sobl.org/traduccion/practical-postgres/node19.html>]

El backend de Postgres (el programa ejecutable postgres real) lo puede ejecutar el superusuario directamente desde el intérprete de órdenes de usuario de Postgres (con el nombre de la base de datos como un argumento). Sin embargo, hacer esto elimina el buffer pool compartido y bloquea la tabla asociada con un postmaster/sitio, por ello esto no está recomendado en un sitio multiusuario.

5.1.2 Notación de las ordenes por los diferentes usuarios.

Las órdenes ejecutadas desde la cuenta del root estarán precedidas por ">". Las órdenes ejecutadas desde la cuenta del superusuario de Postgres estarán precedidas por "%", mientras que las órdenes ejecutadas desde la cuenta de un usuario sin privilegios estarán precedidas por "\$". Las órdenes de SQL estarán precedidas por "=>" o no estarán precedidas por ningún prompt, dependiendo del contexto.

Al pedirse que el sistema de la Biblioteca estuviera en Internet, por esta razón hay que dar privilegios de acceso en la base de datos/tablas al usuario "Apache" para que pueda tomar los datos de PostgreSQL (Apache es el usuario que ejecuta el servidor Apache por defecto). Por lo tanto con estas instrucciones tendremos permiso de navegación y con ello la interfaz en php cuando se ejecutan los queries de la búsqueda, ya tiene permiso apache y no hay problema.

5.2 Iniciación de Apache y PHP para su uso en el sistema.

Ya tenemos apache instalado y PHP como módulo del mismo. Ahora tenemos que hacer unos cuantos ajustes en la configuración para que todo funcione. Tenemos que editar el archivo **httpd.conf** que esta en la ruta /usr/local/apache/conf/ y añadirle lo siguiente:

5.2.1 Actualización del archivo de configuración

```
AddType application/x-httpd-php3 .php  
DirectoryIndex index.html index.php
```

Estas dos líneas son las únicas necesarias para que Apache sepa que hacer con un archivo que contenga código PHP.

Y para no escribir las instrucciones de Iniciación cada vez que se inicia la maquina hay que decirle al servidor, cuales servicios levantará al iniciarse la computadora y en cuanto respecta al sistema de la biblioteca hay que indicarle tanto **postgreSQL** como **http** .

Para ello hay que configurar "serviceconf" : al escribirlo aparecerá una ventana y en está hay que marcar postgreSQLy http.

5.2.2 Arranque del servidor

Levantar el servicio Apache

```
root@motomorphosys root$ su -
root@motomorphosys root$ /etc/init.d/apache start
root@motomorphosys root$ exit
root@motomorphosys root$ su -
root@motomorphosys root$ /usr/local/apache/bin/httpd
-f /usr/local/apache/conf/httpd.conf
root@motomorphosys root$ exit
```

[REFERENCIA

[Apache](#)

<http://httpd.apache.org/docs/misc/FAQ.html#what>

<http://programacion.com/articulo/apache>

<http://httpd.apache.org/docs/>

<http://programacion.com/articulo/apache>

[PHP](#)

<http://www.openresources.com/es/magazine/tutoriales/php/e1/>

<http://www.php.net/manual/es/security.php>

CAPITULO 6

6. Interfaces Desarrolladas con PHP HTML y Flash

6.1. Interfaces para la búsqueda del artículo por investigador o título o palabra clave

6.1.1. Interfaz principal para la búsqueda de la producción científica del Instituto de Física

Se muestra en esta interfaz las tres búsquedas que se tiene en el sistema, las búsquedas son simples es decir es solo **por** :

1. **investigador,**
2. **por título o**
3. **por palabra clave del artículo.**

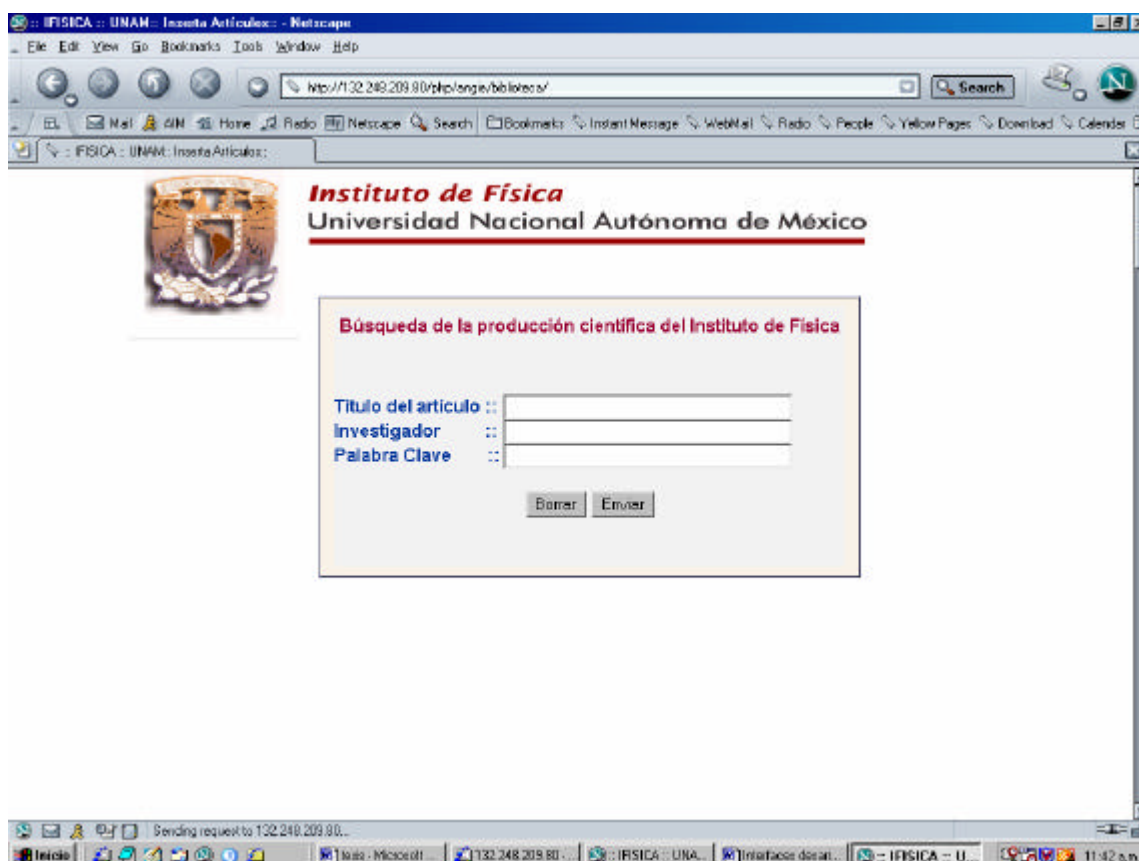


Figura 6.1.1.a **Interfaz principal** para la búsqueda de la producción científica del Instituto de Física.

Pero si se llegara hacer una búsqueda por dos o por tres de los campos se mostrará una interfaz como se ve en la figura 4.2 donde se dice las condiciones para la búsqueda correcta del artículo



Figura 6.1.1.b Interfaz obtenida al hacer la búsqueda por dos o tres campos de la interfaz principal..

Pero si la Búsqueda se hace por un campo, mostrará el artículo según sea la petición en uno de los tres campos que se tiene en la interfaz principal.

6.1.2. Interfaz que muestra de la búsqueda por título del artículo, el artículo solicitado



Figura 4.1.2.a Interfaz principal seleccionando el **título del artículo**.

6.1.2.1. Interfaz final de la búsqueda de la producción científica del Instituto de Física, donde se muestra el abstract del artículo solicitado.

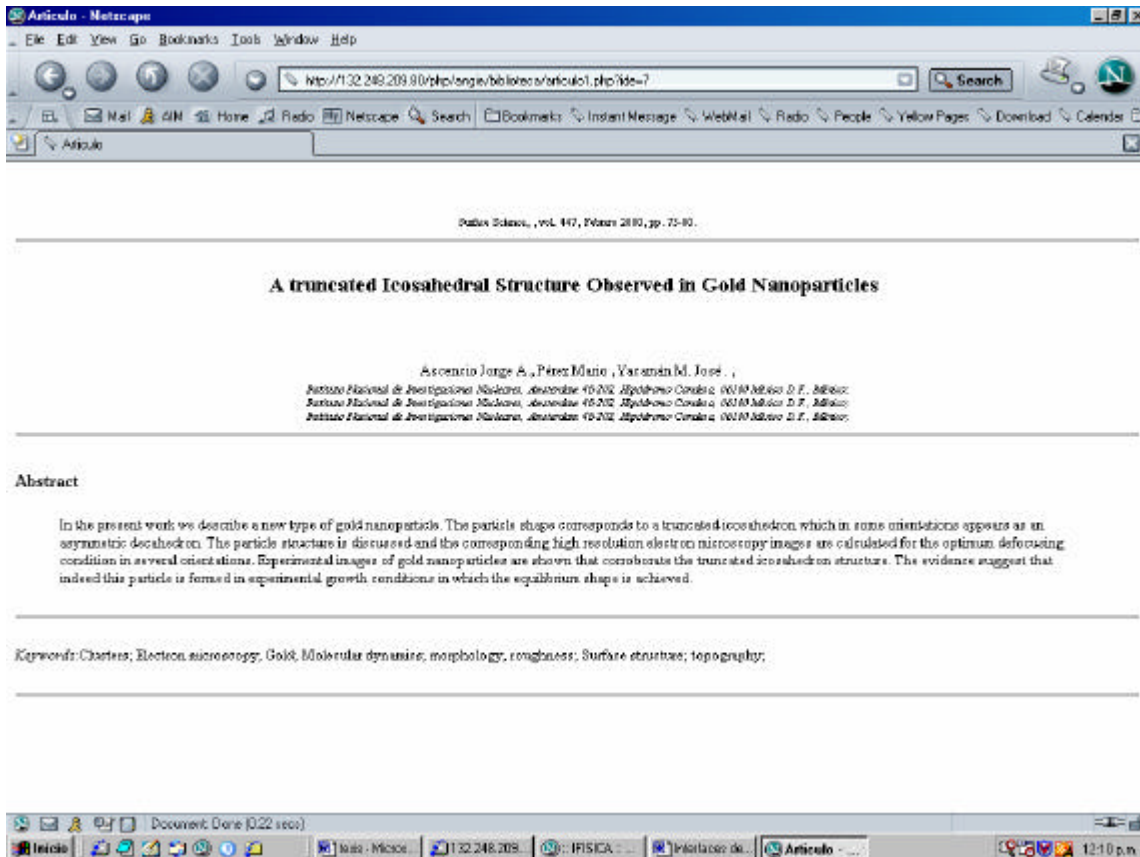


Figura 4.1.2.c Interfaz final, muestra el abstract del artículo.

Código PostgreSQL para esta interfaz sera la siguiente:
"SELECT

investigador.nombre,
editorial.nombre_editorial,
articulos.nombre_articulo,
procedencia.procedencia,
articulos.numero,
articulos.volumen,
articulos.año,
articulos.paginas,
articulos.abstract

FROM articulos, procedencia, investigador, art_invest, editorial
WHERE investigador.id=art_invest.id and art_invest.ide=articulos.ide
and procedencia.idprocedencia=investigador.idprocedencia and
articulos.ideditorial=editorial.ideditorial and articulos.ide='\$ide' order
by investigador.nombre"

Esta interfaz es el resultado final de la búsqueda del artículo ya haya sido por **titulo, investigador, palabra clave**. En esta página se aprecia en el lado superior el nombre de la editorial el volumen, el número, la fecha de publicación y las páginas; posteriormente se tiene con letras negritas el titulo del artículo, a continuación se tiene los investigadores con sus procedencias según sea el orden. En la parte inferior se tiene el abstract y las palabras claves.

6.1.3. Interfaces para la búsqueda por investigador.

Por Investigador “Ascencio Jorge A.”

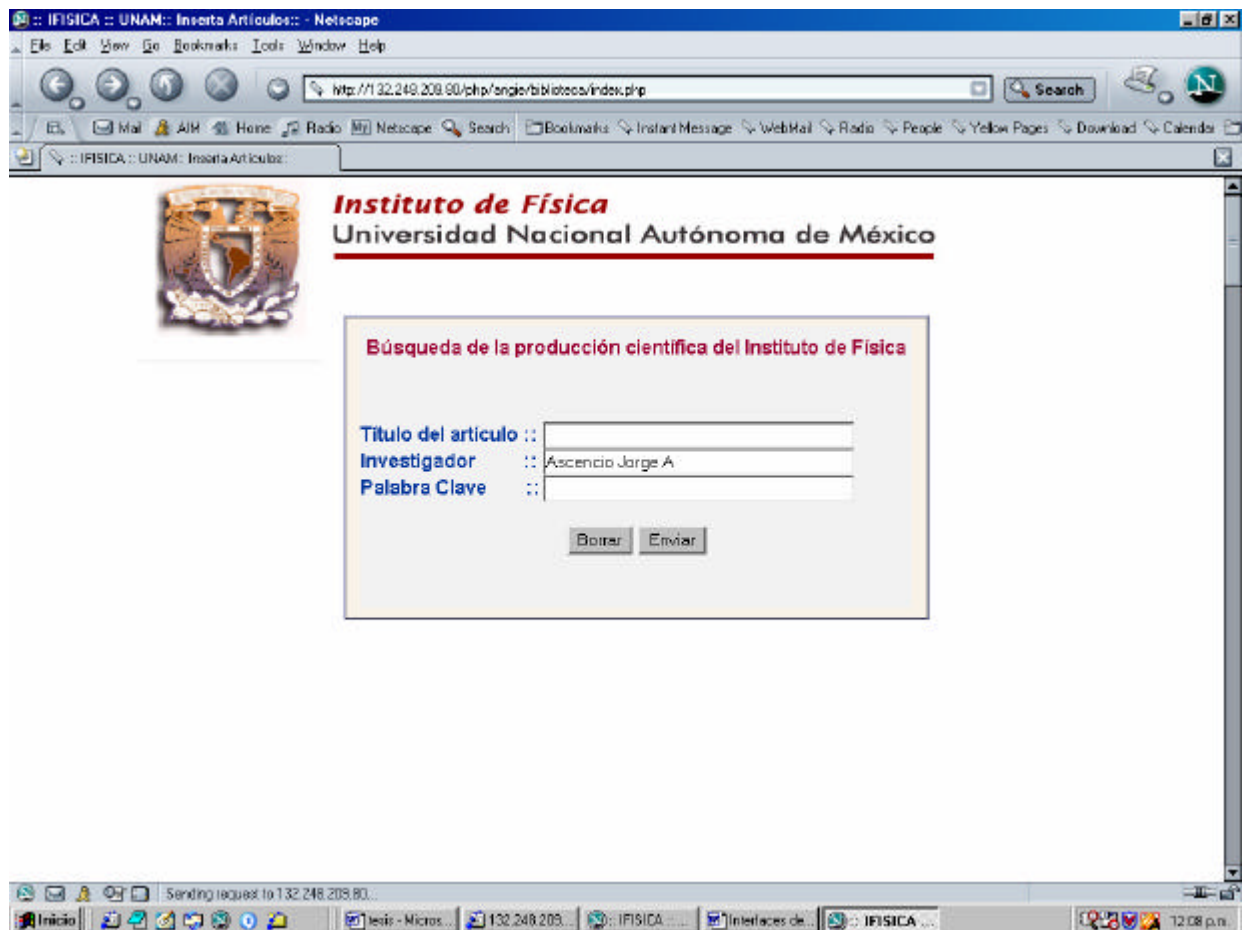


Figura 6.1.3.a Interfaz principal seleccionando el **Investigador**.

De esta interfaz donde se envía la petición de la búsqueda por el Investigador Ascencio Jorge A., mostrará la siguiente interfaz



Figura 6.1.3.b Interfaz obtenida al hacer la búsqueda por título del artículo.

En este caso no se mostrará la pantalla final debido a que el autor puede tener varios artículos escritos y por ende se muestra la lista de los artículos escritos por este autor.

Código PostgreSQL para esta interfaz sera la siguiente:

```
SELECT editorial.nombre_editorial,articulos.nombre_articulo,
investigador.nombre,
articulos.ide FROM articulos, investigador,art_invest, editorial
WHERE
investigador.id=art_invest.id and art_invest.ide=articulos.ide and
editorial.ideditorial=articulos.ideditorial and
UPPER(investigador.nombre)
LIKE UPPER('$autor%') order by investigador.nombre
```

De estas ligas se debe seleccionar el nombre del articulo que tiene este investigador, por ejemplo la segunda.

Interfaz final de la búsqueda de la producción científica del Instituto de Física, donde se muestra el abstract del artículo elegido de la figura 4.3.b



Figura 6.1.3.c Interfaz final, muestra el abstract del artículo.

Este es el artículo del autor solicitado de las ligas anteriores. Por lo que podemos apreciar el abstract del artículo seleccionado de la lista que se mostró al enviar el investigador Ascencio Jorge A.

Código PostgreSQL para esta interfaz será la siguiente:
 “SELECT

```

investigador.nombre,
editorial.nombre_editorial,
articulos.nombre_articulo,
procedencia.procedencia,
articulos.numero,
articulos.volumen,
articulos.año,
articulos.paginas,
articulos.abstract
  
```

FROM articulos, procedencia, investigador, art_invest, editorial
WHERE investigador.id=art_invest.id and art_invest.id=articulos.id
and procedencia.idprocedencia=investigador.idprocedencia and
articulos.ideditorial=editorial.ideditorial and articulos.id='&id' order
by investigador.nombre”

6.1.4. Interfaces para la búsqueda por palabra clave

En esta caso la palabra clave elegida es: “*lattice relaxation*”



Figura 4.4.a Interfaz principal seleccionando la palabra clave

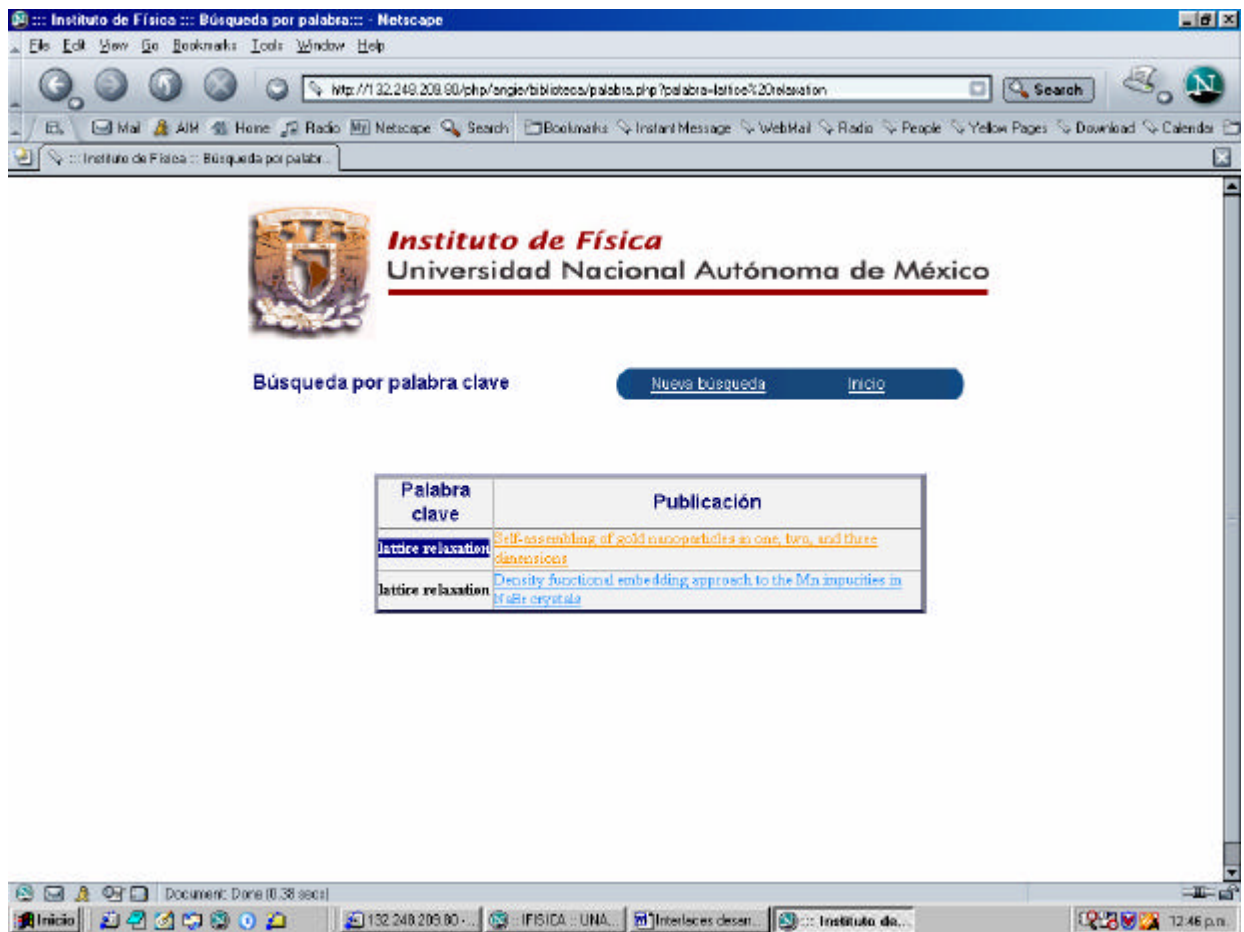


Figura 4.4.b Interfaz obtenida al hacer la búsqueda por palabra clave.

Código PostgreSQL para esta interfaz sera la siguiente:

```

"SELECT
editorial.nombre_editorial,articulos.nombre_articulo,palabra.palabra,
articulos.id
FROM articulos, palabra, art_palabra, editorial
WHERE      palabra.idpalabra=art_palabra.idpalabra      and
art_palabra.id=articulos.id
articulos.ideditorial=editorial.ideditorial  and  UPPER(palabra.palabra)  LIKE
UPPER('$palabra%') order by editorial.nombre_editorial"

```

De esta interfaz al Seleccionar la segunda liga, tenemos:

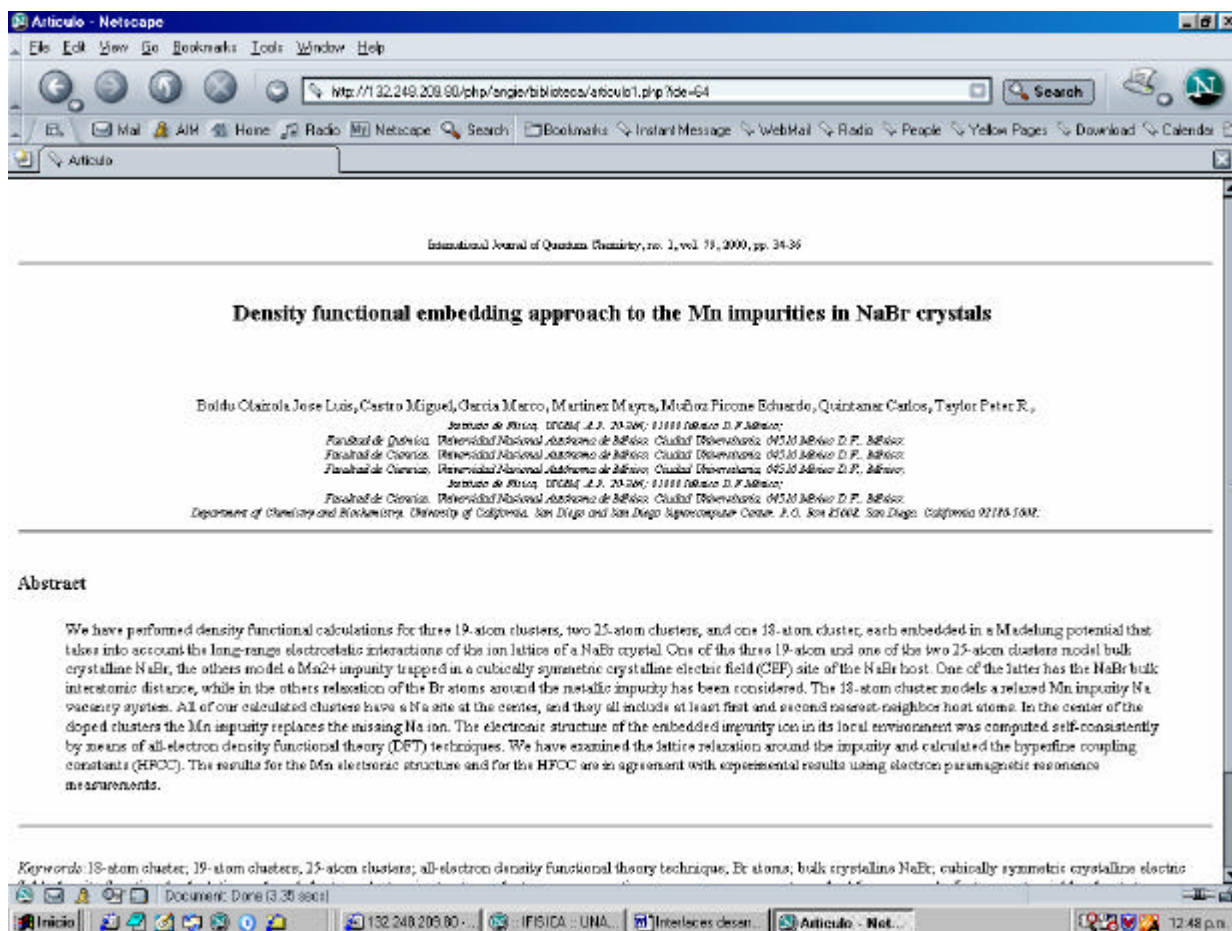


Figura 4.4.c Interfaz final, muestra el abstract del artículo.

Interfaz que muestra el artículo de la búsqueda solicitada.

Código PostgreSQL para esta interfaz sera la siguiente:

“SELECT

investigador.nombre,
 editorial.nombre_editorial,
 articulos.nombre_articulo,
 procedencia.procedencia,
 articulos.numero,
 articulos.volumen,
 articulos.año,
 articulos.paginas,
 articulos.abstract

FROM articulos, procedencia, investigador, art_invest, editorial
 WHERE investigador.id=art_invest.id and art_invest.ide=articulos.ide
 and procedencia.idprocedencia=investigador.idprocedencia and
 articulos.ideditorial=editorial.ideditorial and articulos.ide='\$ide' order
 by investigador.nombre”

6.1.5. Código en PHP de los queries para la búsqueda del artículo por título o investigador o palabra clave

6.1.5.1. Query de la búsqueda por Título del artículo

```
<?
// los parámetros para el acceso a la base de datos
$db = "revistas";
// conectarse a la base de datos
$connection = pg_connect ("dbname=$db");
if (!$connection)
{
    die("No ha sido posible establecer la conexión con la base de datos.");
}
// generar y ejecutar el query
$query = "SELECT
editorial.nombre_editorial,articulos.nombre_articulo,articulos.ide
FROM articulos,editorial
WHERE articulos.ideditorial=editorial.ideditorial and
UPPER(articulos.nombre_articulo) LIKE UPPER('$articulo%')";
$result = pg_exec($connection, $query) or die("Error en el query: $query. " .
pg_last_error($connection));
mostrar_direcciones($result);
// cerrar la conexión con la base de datos
pg_close($connection);
?>
```

6.1.5.2. Query de la búsqueda por Investigadores

```
<?php
// los parámetros para el acceso a la base de datos
$db = "revistas";
// conectarse a la base de datos
$connection = pg_connect ("dbname=$db");
if (!$connection)
{
    die("No ha sido posible establecer la conexión con la base de datos.");
}
// generar y ejecutar el query
$query = "SELECT editorial.nombre_editorial,articulos.nombre_articulo,
investigador.nombre,
articulos.ide FROM articulos, investigador,art_invest, editorial WHERE
investigador.id=art_invest.id and art_invest.ide=articulos.ide and
editorial.ideditorial=articulos.ideditorial and UPPER(investigador.nombre)
LIKE UPPER('$autor%') order by investigador.nombre";

$result = pg_exec($connection, $query) or die("Error en el query: $query. " .
pg_last_error($connection));
mostrar_direcciones($result);
// cerrar la conexión con la base de datos
pg_close($connection);
?>
```

6.1.5.3. Query de la búsqueda por palabras clave

```
<?
// los parámetros para el acceso a la base de datos
$db = "revistas";
// conectarse a la base de datos
$connection = pg_connect ("dbname=$db");
if (!$connection)
{
    die("No ha sido posible establecer la conexión con la base de datos.");
}
// generar y ejecutar el query
$query = "SELECT
editorial.nombre_editorial,articulos.nombre_articulo,palabra.palabra,articulos.ide
FROM articulos, palabra, art_palabra, editorial
WHERE palabra.idpalabra=art_palabra.idpalabra and
art_palabra.ide=articulos.ide and
articulos.ideditorial=editorial.ideditorial and UPPER(palabra.palabra)
LIKE
UPPER('$palabra%') order by editorial.nombre_editorial";

$result = pg_exec($connection, $query) or die("Error en el query: $query. " .
pg_last_error($connection));
mostrar_direcciones($result);
// cerrar la conexión con la base de datos
pg_close($connection);
?>
```

6.2. Interfaces para el administrador

6.2.1. Interfaz de autenticación para el administrador de la base de datos en Internet

Los permisos que se dieron en la base de datos son:

- ✍ **Update**, actualiza los campos de la base de datos según las condiciones establecidas
- ✍ **Delete**, borra registros de la base de datos.
- ✍ **Insert**, inserta registros en la base de datos.
- ✍ **Select**, selecciona registros
- ✍ **Setval** inicializa una secuencia según el número que se le da a la secuencia seleccionada.

Por seguridad a la base de datos revistas se creo un usuario que tiene su login y su password que el administrador es la única persona que contara con estos datos debido a que el es el encargado de usar estos permisos sobre la base de datos.

Interfaz de autenticación para el administrador de la base de datos en Internet



Figura 6.2.1 Interfaz de autenticación para el administrador de la base de datos.

Después de proporcionar estos datos se tendrá acceso a la interfaz correspondiente según sea la dirección URL.

Código PostgreSQL

```
“SELECT name, password from usuario where name='$usuario' and password='$passwd”
```

6.2.2. Interfaz para insertar los artículos de la Base de Datos



figura 6.2.2 Interfaz insertar principal

Esta interfaz muestra 5 inserciones que se pueden realizar al seleccionar esa liga, para Insertar los Artículos, Palabras claves, editoriales, procedencias de los investigadores, investigadores.

El menú que aparece en la parte superior esta hecha en flash ya que al haber proporcionado los datos de autenticación el administrador puede seleccionar estos menús.

6.2.2.1. Interfaz Insertar Artículos

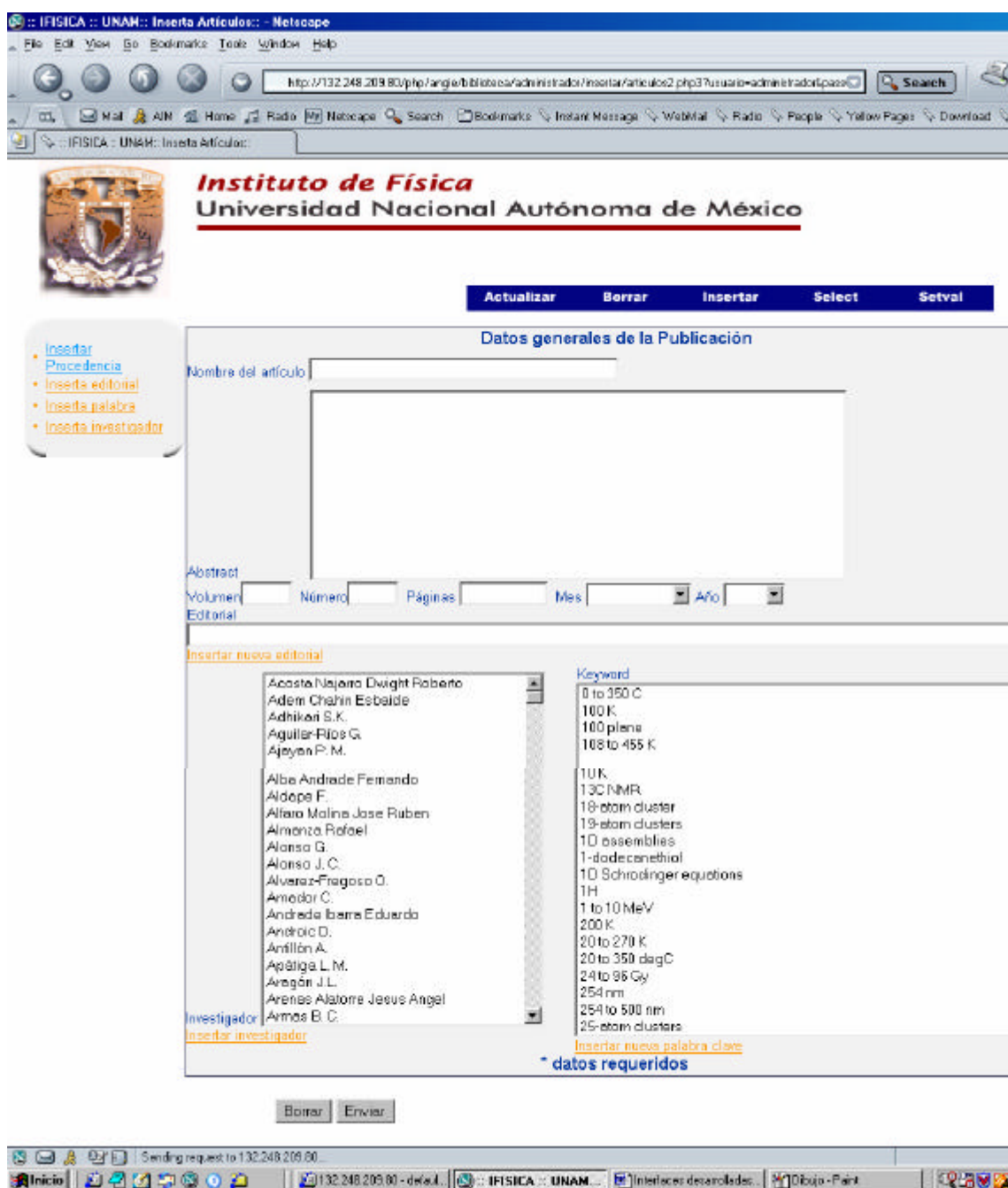


Figura 6.2.2.1 Interfaz inserta artículos (articulos2.php3).

Esta interfaz al pedir en sus campos el título, el abstract, el volumen, número, fecha, editorial, investigador o investigadores (pues es un multiselect), palabras claves (que también están en multiselect).

Código PostgreSQL

Para la Editorial

```
"SELECT * from editorial order by nombre_editorial asc"; //query SQL
```

Para las Palabras Claves

```
"SELECT * from palabra order by palabra asc";
```

Para los investigadores-

```
"SELECT * from investigador order by nombre asc";
```

Código PostgreSQL para insertar el artículo

```
INSERT INTO articulos  
VALUES(nextval('seq_articulo'),'$art','$vn$sp$num','$v$sp$vol','$pg$sp$pag','$select$sp$year','$ed  
itorial','$abstract')
```

Código PHP

En la interfaz hay tres queries a la base de datos que se realizaron para las tres etiquetas <select> de HTML, Las etiquetas <select> que corresponden en la interfaz son:

? Editorial
? Palabra Clave
? Investigador

- Para la Editorial-

```
<select name=editorial>  
  <option></option>  
<?php  
$conn=pg_connect("dbname=revistas"); //conexion a la base de datos  
$query="SELECT * from editorial order by nombre_editorial asc"; //query SQL  
$result=pg_exec($query); //ejecuta el query  
$rows = pg_numrows($result); //cuenta el numero de filas que se tiene al ejecutar  
el query  
for($i=0; $i<$rows; $i++){  
  $row = pg_fetch_row($result, $i);  
  echo "<option value=$row[0]>$row[1]</option>";  
}  
pg_close($conn);  
?>  
  </select>
```

-Para las Palabras Claves-

```
<select name=investigador[] multiple>;  
  
<?php  
$conn=pg_connect("dbname=revistas");  
  
  $query="SELECT * from investigador order by nombre  
asc";  
$result=pg_exec($query);  
$rows = pg_numrows($result);
```

```

for($i=0; $i<$rows; $i++){
$row = pg_fetch_row($result, $i);
echo"<option value=$row[0]>$row[1]</option>";
}
pg_close($conn);
?>

</select>

```

-Para los investigadores-

```
<select name=investigador[] multiple>;
```

```
$conn=pg_connect("dbname=revistas");
```

```

                $query="SELECT
* from investigador order by nombre asc";
$result=pg_exec($query);
$rows = pg_numrows($result);
for($i=0; $i<$rows; $i++){
$row = pg_fetch_row($result, $i);
echo"<option value=$row[0]>$row[1]</option>";
}
pg_close($conn);

```

```
echo"</select>
```

Cabe señalar que las ultimas etiquetas son multiselect ya que en la base de datos estas asociaciones son N:M

De está interfaz estos datos serán enviados a la siguiente página que se insertarán con la instrucción SQL correspondiente. Si los datos corresponden al tipo de dato establecido en la base de datos o si los campos no están vacios se insertará.

Código PHP para Insertar el articulo

```

<?php
$conn=pg_connect("dbname=revistas");
$query = "INSERT INTO articulos
VALUES(nextval('seq_articulo'),'$art','$vn$sp$num','$v$sp$vol','$pg$sp$pag','$select$sp$year',
'$editorial','$abstract')";
$result = pg_exec($conn, $query);
$qcurrval="SELECT currval('seq_articulo')";
$rcurrval=pg_exec($qcurrval);
for($i=0; $i<1; $i++){
    $row = pg_fetch_row($rcurrval, $i);
    $sequence=$row[0];

```

```

}
for($i=0; $i<=$nokw; $i++){
    $q="Insert into art_palabra values('$sequence','$kw[$i]');
    $result = pg_exec($conn, $q);
}
}
$noin=count($investigador);
for($j=0; $j<=$noin; $j++){
    $qaut="Insert into art_invest values('$sequence','$investigador[$j]');

    $raut = pg_exec($conn, $qaut);
}
pg_close($conn);
?>

```

6.2.2.2. Interfaz Insertar Palabras claves



Figura 6.2.2.2 Interfaz inserta palabra clave del artículo.

En esta interfaz se muestra una tabla que es un SELECT a la tabla palabra clave de la base de datos "Revistas"

Código PostgreSQL

La tabla que se muestra en la interfaz es hecha por un select de postgresQL y su código es:

```
SELECT * FROM palabra order by palabra
```

Código PostgreSQL para insertar la palabra clave

La interfaz 6.2.2.2 envía la palabra escrita en el formulario pero para que no se inserte una palabra repetida se ejecuta la siguiente instrucción y de esta forma no se insertará si se encontró en la base de datos pues quita los espacios, convierte a mayúsculas para poder realizar la comparación.

```
select idpalabra, translate(palabra, '[ ]+', '') from palabra where UPPER(palabra) LIKE UPPER('$p%')
```

Este INSERT se realiza después de haber hecho la comparación anterior.

```
INSERT INTO palabra VALUES(nextval('seq_palabra'),' $p')
```

Código PHP

```
//Muestra el identificador y la palabra clave
$conn = pg_connect ("dbname=revistas");
$query = "SELECT * FROM palabra order by palabra";

$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"

<TABLE BORDER=1 CELLSPACING=1 CELLPADDING=1>
<tr>
<td height=4></td>
<td></td>
</tr>
<TR>
<TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>palabra</b></font></TD>
</TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
$myrow = pg_fetch_row ($result,$i);
printf ("
<tr>
<td><font color=#003399>%s</font></td>
<td><font color=#003399>%s</font></td>
</tr>
\n", $myrow[0], $myrow[1]); }
echo"

</table>";
pg_close($conn);
```

Si el campo no se encuentra vacío o no esta repetido en la base de datos se insertará el registro

En está interfaz se insertará pero no se necesita poner el numero del registro correspondiente pues se incrementara al tener la instrucción nextval.

6.2.2.3. Interfaz Insertar editorial



Figura 6.2.2.3 Interfaz inserta editorial del artículo.

En esta interfaz se escribirá la editorial en el campo <input=text...> y de esta forma se enviará para insertar si es que no se tiene repetido en la base de Datos. También nos muestra las editoriales insertadas anteriormente.

Código PostgreSQL

Debido a la instrucción que tiene en el código de postgresQL que es la función “**nextval**” se incrementará su valor por cada inserción que se realice. Y su código postgresQL de esta interfaz es:

Para mostrar la tabla de la interfaz:

```
“SELECT * FROM editorial order by ideditorial desc “
```

Código PostgreSQL al insertar la editorial

La función “translate” elimina los espacios y me sirve para buscar dentro de la base la editorial y no se inserte repetida, y debido que en PHP envió esta variable también sin espacios

La siguiente consulta a la base de datos se necesita para buscar la editorial y no se repita esta variable enviada si es que se encuentra en la base de datos

```
select ideditorial, translate(nombre_editorial, '[ ]+', '') from editorial where  
UPPER(nombre_editorial) LIKEUPPER ('$edit%')
```

Pero si esa variable enviada al realizar la instrucción anterior y al no haberse encontrado en la base se ejecutará el siguiente INSERT

```
INSERT INTO editorial VALUES(nextval('seq_editorial'),'edit')
```

Código PHP para la interfaz 6.2.2.3

En está interfaz se muestra una tabla que es un SELECT a la tabla editorial de la base de datos “*Revistas*”

```
<?php  
//Muestra el identificador y el nombre de la editorial  
$conn = pg_connect ("dbname=revistas");  
$query = "SELECT * FROM editorial order by ideditorial desc";  
$result = pg_Exec($conn, $query);  
if (!$result) {printf (ERROR); exit;}  
echo"<TABLE BORDER=1 CELLSPACING=1 CELLPADDING=1>  
<TR><TD>&nbsp;<B><font color=#003399>id</B></font></TD>  
<TD>&nbsp;<font color=#003399><B>editorial</B></font></TD></TR>";  
$rows = pg_numrows($result);  
for ($i=0; $i < $rows; $i++){  
    $myrow = pg_fetch_row ($result,$i);  
    echo"<tr><td><font color=#003399>$myrow[0]</font></td><td><font  
color=#003399>$myrow[1]</font></td></tr>";  
    }  
echo"</table>";
```



```

else{
$edit=chop($edit);

$edit=ltrim($edit);
$edit=str_replace(" ;", ";", $edit);
$edit=str_replace(" .", ".", $edit);
$edit=str_replace(" ", " ", $edit);
$edit=str_replace(" ", " ", $edit);
$edit=str_replace(" ", " ", $edit);

$query = "INSERT INTO editorial VALUES(nextval('seq_editorial'),' $edit)";
$result = pg_exec($conn, $query);

```

6.2.2.4. Interfaz Insertar procedencia



Figura 6.2.2.4 Interfaz inserta procedencia del investigador.

En esta interfaz se muestra una tabla que es un SELECT a la tabla procedencia del investigador de la base de datos "Revistas"

Código PostgreSQL

```
SELECT * FROM procedencia order by procedencia
```

Código PostgreSQL para insertar la variable

```
select idprocedencia, translate(procedencia, '[ ]+', '') from procedencia where  
UPPER(procedencia) LIKE UPPER ('$p%') ";
```

Código PHP

```
/**MUESTRA DE LA BASE DE DATOS POR MEDIO DE UN SELECT**/  
$conn = pg_connect ("dbname=revistas");  
$query = "SELECT * FROM procedencia order by procedencia ";  
$result = pg_Exec($conn, $query);  
if (!$result) {printf (ERROR); exit;}  
echo "<TABLE BORDER=1 CELLSPACING=1 CELLPADDING=1>  
<TR><TD>&nbsp;<font face=Arial, Helvetica, sans-serif color=#003399  
>id</B></font></TD>  
<TD>&nbsp;<font color=#003399 face=Arial, Helvetica, sans-serif  
><B>procedencia</B></font></TD></TR>";  
$rows = pg_numrows($result);  
for ($i=0; $i < $rows; $i++){  
    $myrow = pg_fetch_row ($result,$i);  
    echo "<tr><td><font color=#003399 size=1 face=Arial,Helvetica, sans-  
serif>$myrow[0]</font></td>  
    <td><font face=Arial, Helvetica, sans-serif size=1  
color=#003399>$myrow[1]</font></td></tr>";  
}  
echo "</table>";  
pg_close($conn);  
/**FIN DEL SELECT
```

Si el campo no se encuentra vacío o no está repetido en la base de datos se insertará el registro

En esta interfaz se insertará pero no se necesita poner el número del registro correspondiente pues se incrementará al tener la instrucción nextval.

Código PHP para Insertar la procedencia del investigador

Este primer query busca en la base de datos que no este repetido la solicitud enviada por el usuario.

```
$conn = pg_connect ("dbname=revistas");
```

```
$q="select idprocedencia, translate(procedencia, '[ ]+', '') from procedencia where  
UPPER(procedencia) LIKE UPPER  
('$p%') ";  
$resultado=pg_exec($conn, $q);
```

```

$rows = pg_numrows($resultado);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($resultado,$i);
    $procedencia=$myrow[1];
}
$scad = $procedencia;
$scad = strtoupper ($scad);

$bcad=str_replace(" ","",$p);
$bcad=strtoupper($p);
$scad = $procedencia;
$scad = strtoupper($scad);
$bcad= $p;
$bcad=str_replace(" ","",$bcad);
$bcad=strtoupper($bcad);
if($scad==$bcad){ echo"<tr>
                <td align= center bgcolor= #F2F2F2 height= 11 >
                    <font face= Arial, Helvetica, sans-serif color=
#003399><b>$p</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</font><br>
                    <font face= Arial,Helvetica, sans-serif color=
#990000 ><b>ya est&aacute; en la base";
}

```

Este query insertará lo enviado por el usuario

```

else{
    $p=chop($p);
    $p=ltrim($p);
    $p=str_replace(" ,",",",$p);
    $p=str_replace(" .",".",$p);
    $p=str_replace(" ",",$p);
    $p=str_replace(" ",",$p);
    $p=str_replace(" ",",$p);
    $query = "INSERT INTO procedencia VALUES(nextval('seq_procedencia'),'$p)";
    $result = pg_exec($conn, $query);
}

```

6.2.2.5. Interfaz Insertar investigadores

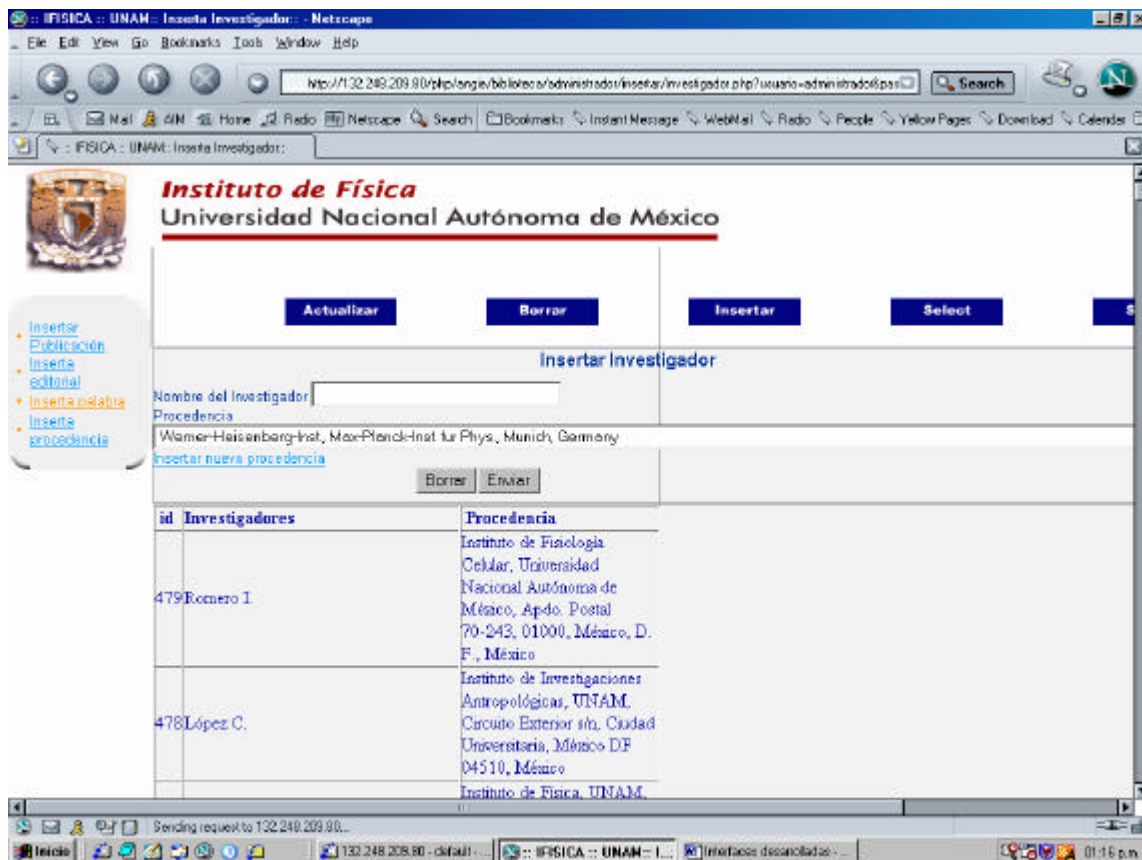


Figura 4.10 Interfaz inserta investigador del artículo.

En esta interfaz se tiene que seleccionar la procedencia del investigador a través de un select.

Código PostgreSQL

La interfaz necesita la siguiente instrucción para poder mostrar la etiqueta <select> de HTML

```
SELECT * from procedencia order by procedencia desc
```

Para mostrar la tabla de la interfaz se necesita esta instrucción:

```
SELECT investigador.id,
investigador.nombre, procedencia.procedencia
FROM investigador,procedencia WHERE
procedencia.idprocedencia=investigador.idprocedencia
order by investigador.id desc
```

Código PostgreSQL para insertar el investigador

```
"select id, translate(nombre, '[ ]+', '')
FROM investigador
WHERE UPPER(nombre) LIKE UPPER('$name%')
```

Código PHP

En esta interfaz se muestra una tabla que es un SELECT a la tabla investigador y a la tabla procedencia de la base de datos "Revistas", mostrando en esta interfaz al investigador con su correspondiente procedencia.

Por lo que se tiene 2 Queries:

Query que muestra la procedencia en la etiqueta <select> de HTML <select name=procedencia>;

```
<select name=procedencia>
$conn=pg_connect("dbname=revistas");
$query="SELECT * from procedencia order by procedencia desc";
$result=pg_exec($query);
$rows = pg_numrows($result);
for($i=0; $i<$rows; $i++){
    $row = pg_fetch_row($result, $i);
    echo "
                                <option value=$row[0]>
                                $row[1]</option>";
}
pg_close($conn);
```

El siguiente query es para que se pueda ver el nombre del investigador su identificador y su procedencia.

```
/**MUESTRA DE LA BASE DE DATOS POR MEDIO DE UN SELECT**/
$conn= pg_connect ("dbname=revistas");
$query = "SELECT investigador.id,
            investigador.nombre, procedencia.procedencia
            FROM investigador,procedencia WHERE
            procedencia.idprocedencia=investigador.idprocedencia
            order by investigador.id desc";
$result = pg_Exec($conn, $query);
if (!$result) {printf (ERROR); exit;}
echo "
<TABLE BORDER=1 CELSPACING=0 CELLPADDING=0>
<TR>
    <TD width=16><font face=\Arial, Helvetica, sans-serif\
color=\#003399>&nbsp;<B>id</B></font></TD>
    <TD width=107><font color=\#003399\ face=\Arial, Helvetica,
sans-serif><B>Investigadores</B></font></TD>
    <TD width=1379><font face=\Arial, Helvetica, sans-serif\
color=\#003399>&nbsp;<B>Procedencia</B></font></TD>
</TR>
    $rows =pg_numrows($result);
    for ($i=0; $i < $rows; $i++){
        $myrow = pg_fetch_row ($result,$i);
        echo "
        <tr>
            <td width=16> <font color=\#003399\ size=1\ face=\Arial,
Helvetica, sans-serif>$myrow[0]
            </font></td>
            <td width=107><font size=2\ color=\#003399\ face=\Arial,
```


Este query insertará lo enviado por el usuario

```
else{
    $name=chop($name);
    $name=ltrim($name);
    $name=str_replace(" .", "", $name);
    $name=str_replace(" ", "", $name);
    $name=str_replace(" ", "", $name);
    $name=str_replace(" ", "", $name);
    $name=str_replace(" ", "", $name);
    $query = "INSERT INTO investigador
VALUES(nextval('seq_investigador'),'<u>$name</u>','$procedencia')";
    $result = pg_exec($conn, $query);
}
```

6.2.3. Interfaz para Actualizar



figura 6.2.3 Interfaz principal, donde se escogerá que tabla se desea actualizar.

Esta interfaz muestra 7 actualizaciones que se pueden realizar al seleccionar esa liga, para Insertar los Artículos, Palabras claves, editoriales, procedencias de los investigadores, investigadores, procedencias y abstract.

6.2.3.1. Actualizar Artículos



figura 6.2.3.1 Interfaz que actualiza el artículo de la base de datos

En esta interfaz, se actualiza al seleccionar el checkbox y modificar los datos pues los datos que aparecen en el type=text son los que se tiene almacenados en la base de datos, por ello hay que escribir al que se desea actualizar.

Código PostgreSQL

Para mostrar la tabla que se muestra en esta interfaz se requiere el siguiente código.

```
SELECT ide, nombre_articulo, numero, volumen, paginas, año, ideditorial FROM articulos order by ide
```

Código PostgreSQL para actualizar el artículo.

```
UPDATE articulos SET
numero='$num[$j]',volumen='$vol[$j]',paginas='$pag[$j]',ideditorial='$edit[$j]',año='$year[$j]'
where ide='$id[0]
```

Código PHP

Se muestra el identificador, el nombre, número, el volumen, las paginas, el año de publicación, la editorial del artículo.

```
<?
$conn = pg_connect ("dbname=revistas");
```

```
$query = "SELECT ide, nombre_articulo, numero, volumen, paginas, año, ideditorial FROM
articulos order by ide";
$result = pg_Exec($conn, $query);
```

Código PHP que actualiza el artículo

```
<?
    $conn = pg_connect ("db name=revistas");
    $query="UPDATE articulos SET
numero=$num[$j],volumen=$vol[$j],paginas=$pag[$j],ideditorial=$edit[$j],año=$year[$j] where
ide=$id[0]";
    $result=pg_exec($query) or die ('The query update abstract failed! ');
    pg_close($conn);
?>
```

6.2.3.2. Actualizar Palabras claves



figura 6.2.3.2 Interfaz actualiza la palabra clave del artículo

Se modificará la palabra clave al escribir la palabra que se desee y el identificador.

Código PostgreSQL

```
SELECT * FROM palabra order by palabra
```

Código PostgreSQL para actualizar la palabra clave

```
UPDATE palabra SET palabra='$ac' where idpalabra='$id'
```

Código PHP

```
<?
$conn = pg_connect ("dbname=revistas");
$query = "SELECT * FROM palabra order by palabra";

$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>palabra</b></font></TD></TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    printf ("<tr><td><font color=#003399>%s</font></td><td><font
color=#003399>%s</font></td></tr>\n", $myrow[0],
$myrow[1]);
}
echo"</table>";
pg_close($conn);
?>
```

Código PHP que actualiza la palabra clave al enviar este formulario

```
<?
$conn = pg_connect ("dbname=revistas");
echo"$ac,$id";
$query="UPDATE palabra SET palabra='$ac' where idpalabra='$id';
$result=pg_exec($query)or die ('The query update palabra failed! ');
pg_close($conn);
```

6.2.3.3. Actualizar editoriales



figura 4.14 Interfaz actualiza la editorial del articulo

En esta interfaz se actualiza la editorial al escribir en el campo la nueva editorial pero escribiendo el identificador para que se puede realizar la actualización.

Código PostgreSQL

Este query es para que se pueda ver en la tabla de la interfaz el identificador y el nombre de la editorial

```
SELECT * FROM editorial order by ideditorial
```

Código PostgreSQL para actualizar la editorial

```
UPDATE editorial SET nombre_editorial='$ac' where ideditorial='$id
```

Código PHP

```
<?
$conn = pg_connect ("dbname=revistas");
$query = "SELECT * FROM editorial order by ideditorial";

$result = pg_exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
```

```

<TD>&nbsp;<font color=#003399><b>editorial</b></font></TD></TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    printf ("<tr><td><font color=#003399>%s</font></td><td><font
color=#003399>%s</font></td></tr>\n", $myrow[0],
$myrow[1]);
}
echo"</table>";
pg_close($conn);

```

?>

Código PHP que actualiza la editorial al enviar este formulario

```

<?
$conn = pg_connect ("dbname=revistas");
echo"$ac,$id";
$query="UPDATE editorial SET nombre_editorial='$ac' where ideditorial='$id'";
$result=pg_exec($query)or die ('The query update editorial failed! ');
pg_close($conn);

```

?>

6.2.3.4. Actualizar procedencias de los investigadores

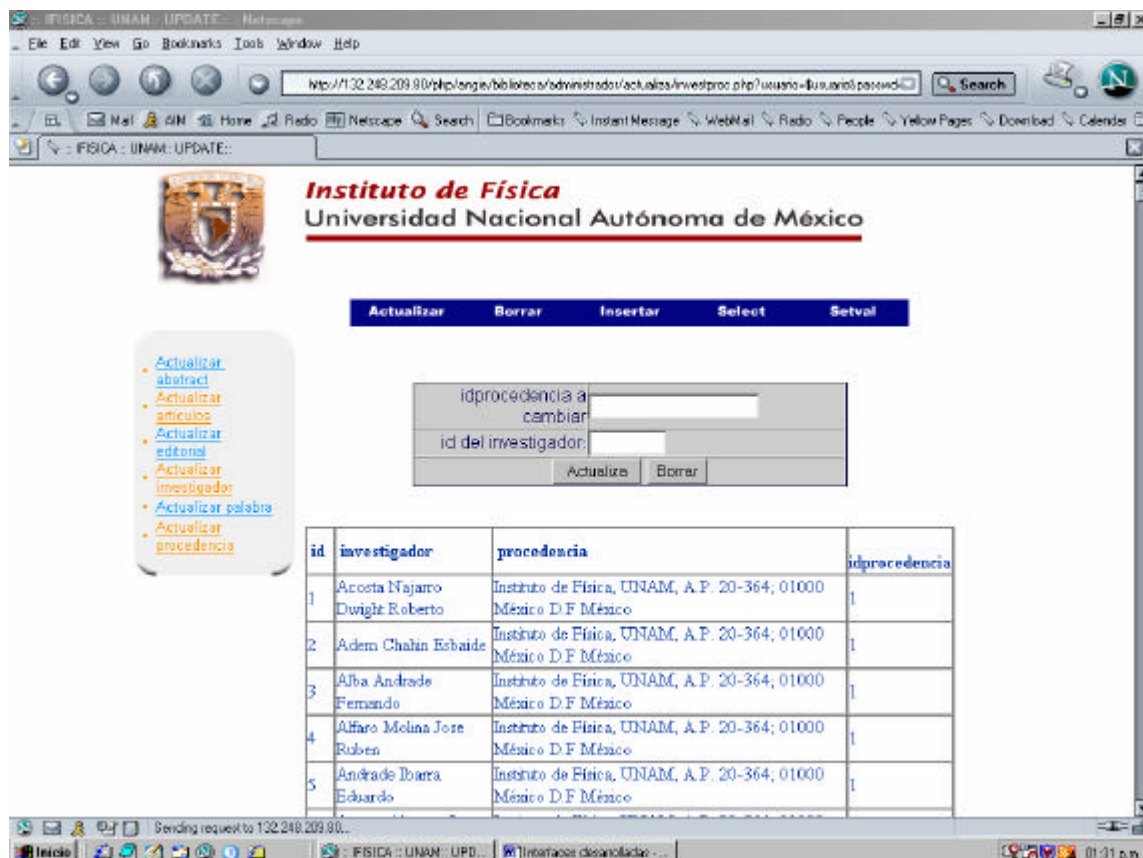


figura 4.15 Interfaz que actualiza la procedencia del investigador

Buscar el identificador de la procedencia para que se escriba en el primer campo solicitado y en el siguiente campo escribir el identificador que le corresponde al investigador que se quiere.

Código PostgreSQL

```
SELECT investigador.id,
investigador.nombre,procedencia.procedencia,procedencia.idprocedencia
FROM investigador,procedencia
WHERE procedencia.idprocedencia=investigador.idprocedencia order by investigador.id
```

Código PostgreSQL *actualiza la procedencia del investigador*

```
UPDATE investigador SET idprocedencia='$ac' where id='$id'
```

Código PHP

Este query es para que se pueda ver en la tabla de la interfaz el identificador, el nombre de la editorial, la procedencia y el identificador de la procedencia

```
<?
    $conn = pg_connect ("dbname=revistas");
    $query = "SELECT investigador.id,
investigador.nombre,procedencia.procedencia,procedencia.idprocedencia FROM
investigador,procedencia WHERE procedencia.idprocedencia=investigador.idprocedencia order by
investigador.id ";

    $result = pg_Exec($conn, $query);
    if (!$result) {printf ("ERROR"); exit;}
    echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>investigador</b></font></TD>
<TD>&nbsp;<font color=#003399><b>procedencia</b></font></TD>
<TD>&nbsp;<font color=#003399><b>idprocedencia</b></font></TD></TR>";

    $rows = pg_numrows($result);
    for ($i=0; $i < $rows; $i++){
        $myrow = pg_fetch_row ($result,$i);
        echo"<tr><td><font color=#003399>$myrow[0]</font></td><td><font
color=#003399>$myrow[1]</font></td><td><font color=#003399>$myrow[2]</font></td><td><font
color=#003399>$myrow[3]</font></td></tr>";
    }
    echo"</table>";
    pg_close($conn);
?>
```

Código PHP que actualiza la procedencia del investigador al enviar este formulario

```
<?
    $conn = pg_connect ("dbname=revistas");
    echo"$ac,$id";
    $query="UPDATE investigador SET idprocedencia='$ac' where id='$id'";
    $result=pg_exec($query)or die ('The query update investigador failed! ');
    pg_close($conn);
?>
```

6.2.3.5. Actualizar investigadores



figura 4.16 Interfaz actualiza articulo

Escribir el nombre correcto del investigador y poner el identificador que le corresponde.

Código PostgreSQL

```
SELECT investigador.id,  
investigador.nombre,procedencia.procedencia,procedencia.idprocedencia FROM  
investigador,procedencia WHERE  
procedencia.idprocedencia=investigador.idprocedencia order by investigador.id
```

Código PostgreSQL para "actualiza investigador"

```
UPDATE investigador SET nombre='$ac' where id='$id'
```

Este query es para que se pueda ver en la tabla de la interfaz el identificador y el nombre del investigador

Código PHP

<?

```
$conn = pg_connect ("dbname=revistas");
```

```
$query = "SELECT investigador.id,
investigador.nombre,procedencia.procedencia,procedencia.idprocedencia FROM
investigador,procedencia WHERE procedencia.idprocedencia=investigador.idprocedencia order by
investigador.id ";
```

```
$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>investigador</b></font></TD>
</TR>";
```

```
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    echo"<tr><td><font color=#003399>$myrow[0]</font></td><td><font
color=#003399>$myrow[1]</font></td></tr>";
}
echo"</table>";
pg_close($conn);
?>
```

Código PHP que actualiza al investigador al enviar este formulario

```
<?
$conn = pg_connect ("dbname=revistas");
echo"$ac,$id";
$query="UPDATE investigador SET nombre='$ac' where id='$id";
$result=pg_exec($query)or die ('The query update investigador failed! ');
pg_close($conn);
```

?>

6.2.3.6. Actualizar procedencias



Figura 4.17 Interfaz actualiza procedencia.

Se necesita escribir en el primer campo el nombre correcto de la procedencia y en el siguiente campo escribir su identificador correspondiente a esa procedencia.

Código PostgreSQL

```
SELECT * FROM procedencia order by idprocedencia
```

Código PostgreSQL para actualizar procedencia

```
UPDATE procedencia SET procedencia='$ac' where idprocedencia='$id'
```

Código PHP

Este query es para que se pueda ver en la tabla de la interfaz el identificador y el nombre de la procedencia

```

<?
$conn = pg_connect ("dbname=revistas");
$query = "SELECT * FROM procedencia order by idprocedencia";

$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>procedencia</b></font></TD></TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    printf ("<tr><td><font color=#003399>%s</font></td><td><font
color=#003399>%s</font></td></tr>\n", $myrow[0],
$myrow[1]);
}
echo"</table>";
pg_close($conn);

```

?>

Código PHP que actualiza la procedencia de la tabla procedencia de la base de datos al enviar este formulario

```

<?
$conn = pg_connect ("dbname=revistas");
echo"$ac,$id";
$query="UPDATE procedencia SET procedencia='$ac' where idprocedencia='$id'";
$result=pg_exec($query)or die ('The query update procedencia failed! ');
pg_close($conn);

```

?>

6.2.3.7. Actualizar abstract



Figura 4.18 Interfaz actualiza el abstract del articulo

Se tiene que seleccionar el checkbox correspondiente al articulo del abstract que se desea cambiar en el campo textarea modificarlo y enviar esta petición para que se ejecute el query que tiene el programa de actualizar el abstract.

Código PostgreSQL

```
SELECT ide,abstract, nombre_articulo FROM articulos order by ide
```

Código PostgreSQL para actualizar abstract

```
UPDATE articulos SET abstract='$ac[$j]' where ide='$id[0]
```

Código PHP

Este query es para ver en la tabla de la interfaz el identificador con un checkbox, el nombre del artículo, y el abstract al que podemos modificar

```

<?
                                $conn = pg_connect ("dbname=revistas");
$query = "SELECT ide,abstract, nombre_articulo FROM articulos order by ide";

$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>titulo del<br> articulo</b></font></TD>
<TD>&nbsp;<font color=#003399><b>abstract </b></font></TD>
</TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    echo"<tr align=\"center\">

        <td align=\"left\" valign=\"middle\" width=\"70\">

            <font size=1 face=\"Arial\" ><input type=checkbox value=$myrow[0]
name=id[]>$myrow[0]</font></td>

        <td align=\"left\" valign=\"middle\" width=\"200\">

            <font size=1 face=\"Arial\">$myrow[2]</font></td>

        <td align=\"left\" valign=\"middle\" width=\"70\">

            <font size=1 face=\"Arial\"><textarea name=ac[] cols=50 rows=6>
$myrow[1]</textarea>

            </font></td>
        </tr>";
    }
echo"</table>";
pg_close($conn);
?>

```

Código PHP que actualiza el abstract al enviar este formulario

```

<?
$j=$id[0]-1;
echo"$id[0],$ac[$j]";

                                $conn = pg_connect ("dbname=revistas");
                                $query="UPDATE articulos SET abstract='$ac[$j]' where ide='$id[0]'";
                                $result=pg_exec ($query) or die ('The query update abstract failed! ');
pg_close($conn);

```

6.2.4. Interfaz para seleccionar un registro de una tabla

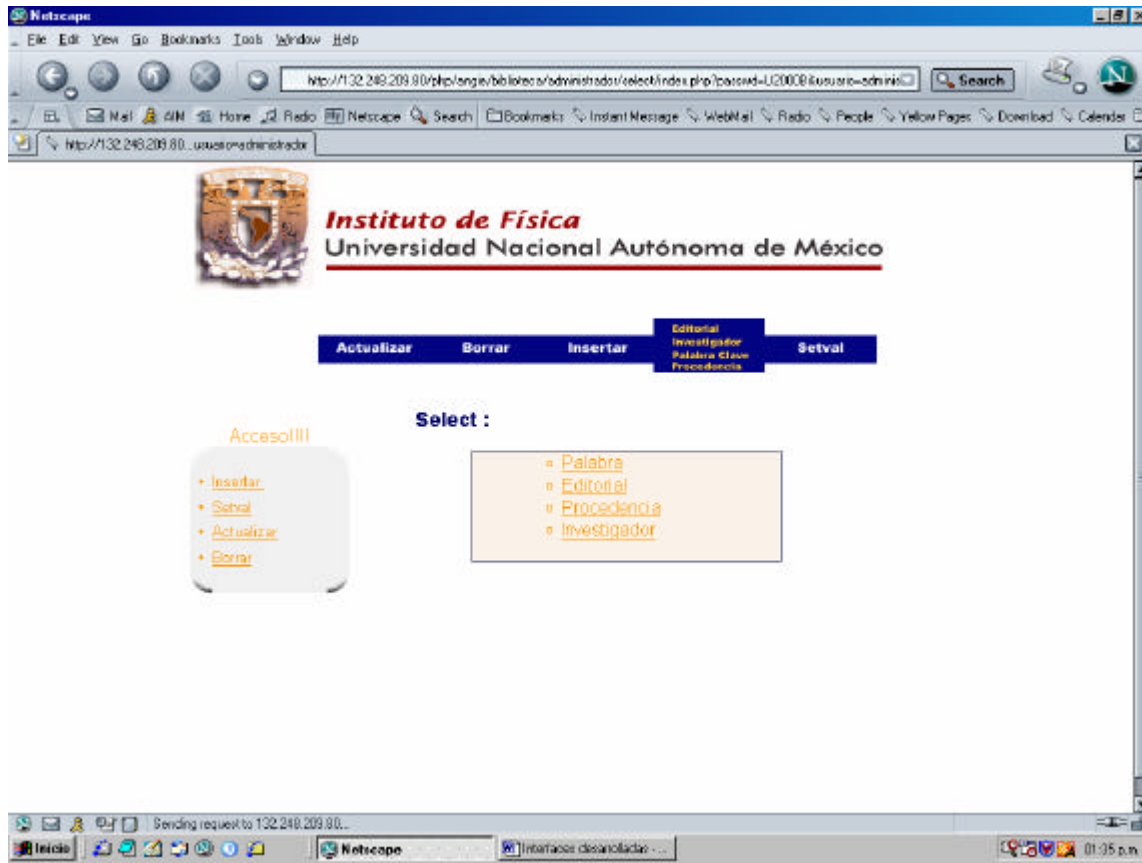


Figura 4.19 Interfaz principal de seleccionar.

Se tiene 4 ligas donde se buscará en las tablas correspondientes según sean la liga elegida.

6.2.4.1. Buscar Palabra



Figura 4.20 Interfaz que busca una palabra

Buscará en la base de datos la palabra que se escriba en el campo, al enviar se mostrará en la pantalla si se encontró esa palabra u otras parecidas.

Código PostgreSQL

```
SELECT * FROM palabra order by idpalabra
```

Código PostgreSQL para buscar la palabra clave

```
SELECT * from palabra where upper(palabra) like upper('%$ac%')
```

Código PHP

En esta interfaz se muestra el identificador y el nombre de la palabra clave, con el siguiente query.

<?

```
$conn = pg_connect ("dbname=revistas");  
$query = "SELECT * FROM palabra order by idpalabra";
```

```

$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>palabra</b></font></TD></TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    printf ("<tr><td><font color=#003399>%s</font></td><td><font
color=#003399>%s</font></td></tr>\n", $myrow[0],
$myrow[1]);
}
echo"</table>";
pg_close($conn);
?>

```

Código PHP que busca la palabra clave enviada

<?

```

$ac=chop($ac);
$ac=ltrim($ac);
$ac=str_replace(" ", "", $ac);
$ac=str_replace(" .", ".", $ac);
$ac=str_replace(" ", "", $ac);
$ac=str_replace(" ", "", $ac);
$ac=str_replace(" ", "", $ac);

```

```

$conn=pg_connect("dbname=revistas");
$query="SELECT * from palabra where upper(palabra) like upper('%$ac%')";
$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo" <font color=#003366\" face=\"Arial, Helvetica, sans-serif\" >
<table border=\"1\" bordercolor=\"#000066\"
bgcolor=\"#CCCCCC\"
cellpadding=\"0\" cellspacing=\"0\" >
<TR bgcolor=#000066>
<TD height=\"10\"><font color=#FFFFFF><b>id</b></font></TD>
<TD height=\"10\"><font color=#FFFFFF><b>palabra</b></font></TD>
</TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    echo"<tr bgcolor=#F2F2F2><td><font
color=#000066 >$myrow[0]</font></td><td><font
color=#000066 >$myrow[1]</font></td></tr>";
}
echo"</table></font>";
pg_close($conn);
?>

```

6.2.4.2. Buscar Editorial



Figura 4.21 Interfaz que busca editorial

Buscará en la base de datos la palabra que se escriba en el campo, al enviar se mostrará en la pantalla si se encontró esa palabra u otras parecidas en la tabla editorial.

Código PostgreSQL

```
SELECT * FROM editorial order by ideditorial
```

Código PostgreSQL para la búsqueda de la editorial

```
SELECT * from editorial where UPPER(nombre_editorial) like  
UPPER('%$ac%')
```

Código PHP

En esta interfaz se muestra el identificador y el nombre de la editorial, con el siguiente query.


```

<?
                                $conn = pg_connect ("dbname=revistas");
$query = "SELECT * FROM editorial order by ideditorial";

$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>editorial</b></font></TD></TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    printf ("<tr><td><font color=#003399>%s</font></td><td><font
color=#003399>%s</font></td></tr>\n", $myrow[0],
$myrow[1]);
}
echo"</table>";
pg_close($conn);
?>

```

Código PHP que busca la editorial enviada por el usuario

```

<?
$ac=chop($ac);
                                $ac=ltrim($ac);
                                $ac=str_replace(" ", "", $ac);
                                $ac=str_replace(" .", ".", $ac);
                                $ac=str_replace(" ", "", $ac);
                                $ac=str_replace(" ", "", $ac);
                                $ac=str_replace(" ", "", $ac);
                                $ac=str_replace(" ", "", $ac);

$conn=pg_connect("dbname=revistas");
$query="SELECT * from editorial where UPPER(nombre_editorial) like
UPPER('%$ac%')";
$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo" <font color=#003366" face="Arial, Helvetica, sans-serif">
<table border="1" bordercolor="#000066"
bgcolor="#CCCCCC"
cellpadding="0" cellspacing="0" >
<TR bgcolor=#000066>
<TD height="10"><font color=FFFFFF><b>id</b></font></TD>
<TD height="10"><font color=FFFFFF><b>editorial</b></font></TD>
</TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    echo"<tr bgcolor=#F2F2F2><td><font
color=#000066 >$myrow[0]</font></td><td><font
color=#000066 >$myrow[1]</font></td></tr>";
}
echo"</table></font>";
pg_close($conn);
?>

```

6.2.4.3. Buscar Procedencia



Figura 4.22 Interfaz que busca procedencia

Código PostgreSQL

Para esta interfaz es el siguiente:

```
SELECT * FROM editorial order by ideditorial
```

Código PostgreSQL para buscar procedencia

```
SELECT * from procedencia where UPPER(procedencia)  
like UPPER('%$ac%')
```

Código PHP

Buscará en la base de datos la palabra que se escriba en el campo, al enviar se mostrará en la pantalla si se encontró esa palabra u otras parecidas en la tabla procedencia.

En esta interfaz se muestra el identificador y el nombre de la procedencia, con el siguiente query.

```
<?
                                $conn = pg_connect ("dbname=revistas");
    $query = "SELECT * FROM editorial order by ideditorial";

    $result = pg_Exec($conn, $query);
    if (!$result) {printf ("ERROR"); exit;}
    echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
    <TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
    <TD>&nbsp;<font color=#003399><b>editorial</b></font></TD></TR>";
    $rows = pg_numrows($result);
    for ($i=0; $i < $rows; $i++){
        $myrow = pg_fetch_row ($result,$i);
        printf ("<tr><td><font color=#003399>%s</font></td><td><font
    color=#003399>%s</font></td></tr>\n", $myrow[0],
    $myrow[1]);
    }
    echo"</table>";
    pg_close($conn);
?>
```

Código PHP que busca la procedencia enviada por el usuario

```
<?
    $ac=chop($ac);
                                $ac=ltrim($ac);
                                $ac=str_replace(" ", "", $ac);
                                $ac=str_replace(" .", "", $ac);
                                $ac=str_replace(" ;", "", $ac);
                                $ac=str_replace(" ", "", $ac);
                                $ac=str_replace(" ", "", $ac);

    $conn=pg_connect("dbname=revistas");
    $query="SELECT * from procedencia where UPPER(procedencia)
    like UPPER('%$ac%')";
    $result = pg_Exec($conn, $query);
    if (!$result) {printf ("ERROR"); exit;}
    echo " <font color='\#003366\' face='\Arial, Helvetica, sans-serif\'>
    <table border='\1\' bordercolor='\#000066\'
    bgcolor='\#CCCCCC\'
    cellpadding='\0\' cellspacing='\0\' >
    <TR bgcolor=#000066>
    <TD height='\10\'><font color=#FFFFFF><b>id</b></font></TD>
    <TD height='\10\'><font color=#FFFFFF><b>procedencia</b></font></TD>
    </TR>";
    $rows = pg_numrows($result);
    for ($i=0; $i < $rows; $i++){
        $myrow = pg_fetch_row ($result,$i);
        echo"<tr bgcolor=#F2F2F2><td><font
    color=#000066 >$myrow[0]</font></td><td><font
    color=#000066 >$myrow[1]</font></td></tr>";
    }
    echo"</table></font>";
    pg_close($conn);
?>
```

6.2.4.4. Buscar Investigador



Figura 4.23 Interfaz que busca investigador

Buscará en la base de datos la palabra que se escriba en el campo, al enviar se mostrará la pantalla si se encontró esa palabra u otras parecidas en la tabla investigador.

Código PostgreSQL

```
SELECT * FROM investigador order by nombre
```

Código PostgreSQL busca investigador

```
SELECT investigador.id, investigador.nombre,  
procedencia.procedencia, procedencia.idprocedencia from  
procedencia, investigador where  
procedencia.idprocedencia=investigador.idprocedencia and  
UPPER(investigador.nombre) like UPPER('%$ac%')
```

Código PHP

En esta interfaz se muestra el identificador y el nombre del investigador, con el siguiente query.

```
<?
$conn = pg_connect ("dbname=revistas");
$query = "SELECT * FROM investigador order by nombre";

$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo"<TABLE BORDER=1 CELLSPACING=0 CELLPADDING=0 >
<TR><TD>&nbsp;<font color=#003399><b>id</b></font></TD>
<TD>&nbsp;<font color=#003399><b>investigador</b></font></TD></TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    printf ("<tr><td><font color=#003399>%s</font></td><td><font
color=#003399>%s</font></td></tr>\n", $myrow[0],
$myrow[1]);
}
echo"</table>";
pg_close($conn);
?>
?>
```

Código PHP que busca la procedencia enviada por el usuario

```
<?
$conn=pg_connect("dbname=revistas");
$query="SELECT investigador.id, investigador.nombre,
procedencia.procedencia, procedencia.idprocedencia from
procedencia,investigador where
procedencia.idprocedencia=investigador.idprocedencia and
UPPER(investigador.nombre) like UPPER('%$ac%')";
$result = pg_Exec($conn, $query);
if (!$result) {printf ("ERROR"); exit;}
echo" <font color=#003366\ face="Arial, Helvetica, sans-serif">
<table border="1" bordercolor=#000066"
bgcolor=#CCCCCC"
cellpadding="0" cellspacing="0" >
<TR bgcolor=#000066>
<TD height="10"><font color=#FFFFFF><b>id</b></font></TD>
<TD height="10"><font color=#FFFFFF><b>investigador</b></font></TD>
<TD height="10"><font color=#FFFFFF><b>procedencia</b></font></TD>
<TD height="10"><font
color=#FFFFFF><b>idprocedencia</b></font></TD></TR>";
$rows = pg_numrows($result);
for ($i=0; $i < $rows; $i++){
    $myrow = pg_fetch_row ($result,$i);
    echo"<tr bgcolor=#F2F2F2><td><font
color=#000066>$myrow[0]</font></td><td><font
color=#000066>$myrow[1]</font></td><td><font
color=#000066>$myrow[2]</font></td><td><font
color=#000066>$myrow[3]</font></td></tr>";
```

```

}
echo"</table></font>";
pg_close($conn);
?>

```

6.2.5. Interfaz para Borrar



Figura 4.24 Interfaz que borra un registro de una tabla

Al seleccionar del select la tabla a la que se quiere borrar un registro en el siguiente campo aparece el número que le corresponde a ese registro.

Código PostgreSQL para borrar

```
DELETE from $table where $id='$num'
```

Código PHP

El query que borra el registro de la tabla seleccionada

```

<?
    $id=strstr($cad,',');//ID
    $id=substr($id,1);
    $com=",";
    $table=strrpos($cad,$com);//TABLE

```

```

$table=substr($cad,0,$table);

$con=pg_connect("dbname=revistas");
$query="DELETE from $table where $id='$num'";
$result=pg_exec($query);
pg_close($con);
?>

```

?>

6.2.6. Interfaz para inicializar una secuencia



Figura 4.25 Interfaz que inicializa una secuencia

Se selecciona la secuencia que se desea modificar su valor y en el siguiente campo se pide el número al que se modificará.

Código PostgreSQL

```
select setval('$cad', '$num');
```

Código PHP

<?

```
$con=pg_connect("dbname=revistas");
```

```
$query="select setval('$cad', $num)";  
$result=pg_exec($query);  
pg_close($con);
```

?>

Conclusiones

En esta tesis se presentó el desarrollo y la implementación del sistema que permite realizar la búsqueda, las altas y bajas de los abstracts de la Producción Científica del Instituto de Física en Internet.

La aplicación requiere de un sistema multiplataforma que funcione tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server), de acuerdo con el análisis, recursos del problema se eligieron las herramientas de tipo software libre.

Por lo tanto el sistematizar la producción científica para su uso en Internet se ahorra muchos recursos, por lo que se mencionan las siguientes ventajas en el sistema.

- ✍ Realiza la búsqueda de la Producción Científica del Instituto de Física (PCIF) por medio de tres diferentes campos que son, búsqueda por autor, por título del artículo o por palabra clave del artículo.
- ✍ Se tiene la visualización de los abstracts de la PCIF a través de Internet
- ✍ Además de lograr el almacenamiento de la Producción Científica del Instituto de Física (PCIF) en una base de datos relacional. Esta inserción la puede realizar cualquier persona que tenga el login y el password correspondientes y sin saber las instrucciones SQL para llevarse a cabo esta ejecución, pues al estar en Internet solo se muestra los campos que se deben llenar para su inserción.

Al evaluar las herramientas de trabajo PostgreSQL, PHP y Apache con sus características descritas tenemos un sistema que realiza búsquedas rápidas, eficientes, precisas y claras para los usuarios que la utilizan a través de Internet

Otra característica relevante del sistema es el ser dinámico, pero que a la interfaz de autenticación para el administrador en Internet, esta persona es quien puede realizar las modificaciones necesarias a la base de datos así como también realizar altas y bajas de la producción científica del Instituto de Física de la UNAM y aunque se tiene esta producción de 10 años ha sido diseñada para su incrementación de los siguientes años.

En conclusión, el sistema tiene las siguientes características:

- ? Utiliza muchas herramientas (Apache, PHP, PostgreSQL, Linux) siendo estás **gratuitas**.
- ? Es un sistema fácil de consultar.
- ? Es fácil para insertar, borrar, actualizar nueva información
- ? Cualquier persona puede hacer la consulta vía Internet sin necesidad de estar físicamente en la Biblioteca del Instituto.
- ? Se ahorra tiempo al ser practico la búsqueda del abstract.

Glosario

Adabas D Es una base de datos relacional, esta disponible para plataformas Unix así como para Windows.

ANSI (American National Standards Institute) Organismo americano dedicado a la normalización. Por lo que nos ocupa, desarrolló un standard para representación de información en la pantalla, en el terminal. Una BBS en modo ANSI o modo terminal, sería en modo texto.

ASCII American Standard Code for Information Interchange. Estandar Americano para Intercambio de Información. La tabla básica de caracteres ASCII está compuesta por 128 caracteres incluyendo símbolos y caracteres de control. Existe una versión extendida de 256

ARPA (Agencia de Proyectos de Investigación Avanzada) En la actualidad es el nombre del organismo militar norteamericano que antes se llamaba DARPA.

ARPANET **A**dvanced **R**esearch **P**rojects **A**gency **N**etwork (Red de la Agencia de Proyectos de Investigación Avanzada) Una de las primeras redes de computadoras interconectadas a través de líneas telefónicas. Este proyecto fue financiado por DARPA y sin duda constituye uno de los proyectos piloto que sirvió de base para el desarrollo de Internet.

C El lenguaje C se conoce como un lenguaje compilado. Existen dos tipos de lenguaje: interpretados y compilados. Los interpretados son aquellos que necesitan del código fuente para funcionar. Los compilados convierten el código fuente en un fichero objeto y éste en un fichero ejecutable. Podemos decir que el lenguaje C es un lenguaje de nivel medio, ya que combina elementos de lenguaje de alto nivel con la funcionalidad del lenguaje ensamblador. Es un lenguaje estructurado, ya que permite crear procedimientos en bloques dentro de otros procedimientos. Hay que destacar que el C es un lenguaje portable, ya que permite utilizar el mismo código en diferentes equipos y sistemas informáticos: el lenguaje es independiente de la arquitectura de cualquier máquina en particular.

[Referencia

<http://uebb.cs.tu-berlin.de/~krischan/adabasd/LinuxHowTo.html>

<http://dgenp.unam.mx/planteles/p4/p4lenguajec/1int-c.htm#intro>]

C++ es desarrollado por AT&T como extensión de ANSI C. Fue una necesidad de incorporar mecanismos de programación orientada a objetos para controlar la complejidad derivada del tamaño que alcanza los programas escritos en C. C++ hereda de C la obsesión por la búsqueda de eficiencia

CGI Common Gateway Interface . Interface de Acceso Común. Programas usados para hacer llamadas a rutinas o controlar otros programas o bases de datos desde una página Web. También pueden generar directamente HTML.

Cliente Un cliente es un proceso que se ejecuta en el mismo o en diferente nodo y que realiza peticiones al servidor. Estas peticiones son producidas por la necesidad del cliente de acceder a un recurso gestionado por el servidor.

DARPA Defense Advanced Research Projects Agency (Agencia de Proyectos de Investigación Avanzada para la Defensa) Esta organización depende del Departamento de Defensa norteamericano (DoD) y entre otras funciones se encarga de investigar y desarrollar dentro del ámbito militar, jugó un papel fundamental en el desarrollo de Internet.

Datagrama Entidad de datos autocontenida e independiente que transporta información suficiente para ser encaminada desde su **ordenador** de origen a su **ordenador** de destino sin tener que depender de que se haya producido anteriormente tráfico alguno entre ambos y la red de transporte.

DB2 Pertenece a la familia RDBMS de IBM funcionando para diferentes plataformas de sistemas operativos.

[Referencia en internet

<http://www.darpa.mil/>

<http://www.monografiass.com/monografiass/EpyVlplpAZdDhGGDCa.php#Datagrama>

<http://informatica.comunalia.com/modules/glossaire/glossaire-aff.php?lettre=C>

]

DBMS Son las siglas en inglés para los Sistemas de Gestión de Bases de Datos (SGBD). Bajo este nombre se conoce a productos de fabricantes como Oracle, Sybase, Informix, Ingres, Borland, Microsoft, IBM, etc. Sistema de administración de bases de datos. Software que controla la organización, el almacenamiento, la recuperación, la seguridad e integridad de los datos en una base de datos. Acepta solicitudes de la aplicación y ordena al sistema operativo transferir los datos apropiados.

Demonio Los demonios son los encargados de supervisar los procesos, además de garantizar el acceso. En los sistemas UNIX, es una aplicación que está siempre en funcionamiento en un servidor Internet, en segundo plano, para realizar determinadas tareas como, por ejemplo, enviar un mensaje de correo electrónico o servir una página web.

Es una Aplicación UNIX que está alerta permanentemente en un servidor para realizar determinadas tareas como, por ejemplo, enviar un mensaje de correo electrónico o servir una página web.

ETHERNET Es una red cuyo cableado se rige con un grupo de especificaciones que cubre las capas 1 y 2 del modelo OSI. Es una red de banda base, o sea que provee un único canal de comunicación sobre el medio físico (cable), de forma que solo puede usarlo un dispositivo a la vez.

FrontBase Esta Base de datos fue creada para satisfacer las necesidades de un servidor robusto, escalable y no requiriendo administración. Esta base es rápida tiene una amplia gama de características. Es la única base de datos relacional para sistemas operativos Mac.

FTP (File Transfer Protocol). Método utilizado para transferir múltiples archivos entre dos sitios de Internet. Existen muchos sitios que permiten a los usuarios ingresar libremente a sus máquinas para obtener programas shareware u otro tipo de información.

[Referencia

<http://www.frontbase.com/cgi-bin/WebObjects/FrontBase>

<http://aspn.activestate.com/ASPN/docs/PHP/ref.fbsql.html>

http://searchdatabase.techtarget.com/sDefinition/0..sid13_gci213669.00.html

BIBLIOGRAFÍA

Tanenbaum S. Andrew

Sistemas Operativos Modernos

Editorial Pearson Educación

2001]

HTML: (Hypertext Markup Lenguaje). Lenguaje de Marcas de Hypertexto.

Lenguaje para elaborar paginas Web.Fue desarrollado en el CERN.

HTTP (HyperText Transfer Protocol). Protocolo de Transferencia de Hypertexto. Protocolo usado en la WWW para transmitir páginas Web.

Host: (sistema central) Computadora que permite a los usuarios comunicarse con otros sistemas centrales de una red. Los usuarios se comunican utilizando programas de aplicación, tales como el correo electrónico, Telnet, WWW y FTP.

Hyperware Hyperwave es un sistema de información similar a una base de datos, se enfoca en el almacenamiento y manejo de los documentos pudiendo el documento ser posibles piezas de datos que pueden también almacenarse en un archivo, Hyperware fue desarrollada por IICM (Institute for Information processing and Computer supported new Media). Iniciando con el nombre Hyper-G cambiando su nombre por Hyperware para su comercialización en 1996. Hyperware no es software libre,

IMAP **Internet Mail Access Protocol.** Este protocolo representa un mecanismo de comunicación para que los clientes de mail interactúen con servidores, y manipulen buzones

Informix Familia del DBMS de Informix Software, Inc., Menlo Park, CA, que incluye un DBMS relacional basado en SQL, un lenguaje de cuarta generación y juegos de herramientas para la inclusión del SQL en programas de aplicación.

Ingres Es una base de datos que no soporta las instrucciones SQL pero que tiene sus propias instrucciones de bases de datos con el lenguaje QUEL(QUEL(Query Language) es un **lenguaje** comercial que se desarrolló para INGRES, está basado en el cálculo relacional orientado a tuplas) tiene una interactiva interfaz C y los comandos Ingres.

InterBase es una base de datos Multiplataforma cuya seguridad esta amplia para desarrollar y desplegar aplicaciones con bases de datos distribuidas para Internet y dispositivos móviles.

IP es un protocolo para el intercambio de datagramas en modo no conectado. Esto no garantiza la llegada de mensajes (cosa que se hará con el TCP).

[Referencia

<http://www.hyperwave.com/e/news/pr25.html>

<http://www.phpfreaks.com/phpmanual/page/ref.hwapi.html>

<http://www.monografiass.com/monografiass/EpyuVuVFIATWsMTCSL.php>

JAVA Java es un lenguaje orientado a objetos y desarrollado por Sun Microsystems. Comparte similitudes con C, C++ y *Objective C*. Basándose en otros lenguajes orientados al objeto, Java recoge lo mejor de todos ellos y elimina sus puntos más conflictivos.

El principal objetivo de JAVA fue hacer un lenguaje que fuera capaz de ser ejecutado de una forma segura a través de Internet (aunque el código fuera escrito de forma maliciosa). Esta característica requiere la eliminación de muchas construcciones y usos de C y C++. El más importante, es que no existen punteros. Java no puede acceder arbitrariamente a direcciones de memoria.

Java es un lenguaje compilado en un código llamado "*codigo-byte*" (*byte-code*). Este código es interpretado "*en vuelo*" por el interprete Java. Java fue diseñado también para escribir código libre de *bugs*, esto se consigue en gran parte, eliminando las operaciones de localización de memoria del lenguaje C.

JDBC se utiliza comúnmente para conectar un programa del usuario con una base de datos por "detrás de la escena", sin importar qué software de administración o manejo de base de datos se utilice para controlarlo.

Lenguajes procedurales son aquellos en los cuales el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos con el fin de calcular el resultado deseado.

Login Nombre o identificativo de usuario en un sistema remoto.

LINUX es un sistema operativo de 32 bits de libre distribución, desarrollado originalmente por Linus Torvalds, un estudiante de la Universidad Finlandesa de Helsinki, quien en 1991 se abocó a la tarea de reemplazar a Minix, un clon de Unix de pequeñas proporciones y finalidad académica desarrollado años antes por Andrew Tannenbaun. Linux tiene direccionamiento lineal de 32 bits, memoria virtual, es multitarea real, comparte librerías, los módulos del kernel son cargables on-demand, soporta TCP/IP (incluyendo SLIP, PPP, NFS, etc.), y entorno gráfico X-Windows y es distribuido bajo la Licencia General Pública de GNU

LDAP LDAP("Lightweight Directory Acces Protocol", Protocolo Ligero de Acceso a Directorios).

Es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio. Se usó inicialmente como un Front-end o interfaz final para x.500, pero también puede usarse con servidores de directorio únicos y con otros tipos de servidores de directorio.

[Referencia

<http://java.sun.com/products/jdbc/>

<http://www.monografias.com/trabajos13/mabas/mabas.shtml>]

Mainframe macrocomputadora: Una computadora grande. A mediados de los años 60, las épocas antiguas de las computadoras, todas las computadoras eran mainframes (literalmente "bastidor principal"), ya que el término se refería al gabinete que contenía la CPU. Aunque mainframe aún significa gabinete principal, usualmente se refiere a un gran sistema de computación y toda la experiencia asociada que va con él.

modelo OSI Es una arquitectura de red basada en la descomposición en siete niveles de abstracción de red, cada uno de los cuales se apoya en los **servicios** que lo ofrece el nivel inmediatamente inferior. Cada nivel tiene una función bien definida. El conjunto de reglas que rigen el diálogo (virtual) entre un nivel de un sistema y su homólogo en otro sistema se conoce como el **protocolo** de ese nivel. El modelo OSI publicó estándares para cada uno de estos niveles.

mSQL mSQL, o mini SQL, es un motor ligero de la base de datos la cual soporta un subconjunto ANSI SQL. Esto proporciona un rápido acceso a los datos almacenados requiriendo baja memoria mSQL corre en una amplia variedad de sistemas UNIX. mSQL Soporta múltiples índices para la misma tabla, longitud variable de los campos, arquitectura cliente/servidor, múltiples procesos "back-end" y una serie de herramientas para acceder a la información y ejecutar rutinas, incluyendo su *Lite*, que es un pequeño SQL propietario para crear las rutinas.

Fue desarrollado por David J. Hughes en la Universidad Bond en Australia.

MS-SQL Base de datos que utiliza el JDBC.

MS SQL es una bases de datos relacional (RDBMS) para la familia de sistemas operativos Microsoft, esta adecuada para un amplio espectro de clientes corporativos y fabricantes independientes de *software* (ISV).

multi-usuario es cuando se permita a varios usuarios acceder al mismo tiempo a través de terminales, y que distribuya los recursos disponibles entre todos. Así mismo, el sistema debería proporcionar la posibilidad de que más de un usuario pudiera trabajar con la misma versión de un mismo programa al mismo tiempo, y actualizar inmediatamente cualquier cambio que se produjese en la base de datos, quedando reflejado para todos.

MySQL Es un sistema manejador de base de datos relacionales es muy rápido, confiable y fácil de usar. Ofrece una buena y útil conjunto de funciones. Su rápida conectividad y seguridad hace que el servidor MYSQL sea altamente confiable para acceder bases de datos en Internet. El software de MYSQL usa el GPL (General Public License) para definir a través de esta licencia lo que se puede realizar o no con este software.

[Referencia

<http://www.linti.unlp.edu.ar/trabajos/tesisDeGrado/tutorial/redes/modosi.htm>

<http://www.mainframes.com/whatis.htm>

http://edocs.bea.com/wls/docs81/jdbc_drivers/mssqlserver.html

<http://www.cs.umbc.edu/~finin/461/mssql/>

<http://www.mysql.com/doc/en/What-is.html>

<http://www.mysql.com/doc/en/Introduction.html>

]

NCP : Network Control Protocol: Programa para Control de Redes. Un programa que controla el tráfico entre múltiples terminales y una mini o macroordenador. Típicamente reside en un procesador frontal y ejecuta operaciones tales como el censo de las terminales.

NNTP Protocolo de transferencia de Noticias.

Object Pascal, Es un lenguaje compilado de alto nivel robusto que soporta la estructura orientado a objetos, teniendo como beneficios el tener una fácil lectura de código, una rápida compilación y utiliza múltiples unidades de archivos para la programación modular.

ODBC Open Data Base Connection (Conexión abierta a base de datos) . Es una forma de conexión a veces de datos, independientemente del lenguaje o programa que estemos utilizando. Cada fabricante provee su propia librería(DLL) con las características de conexión a las bases de datos.

Oracle (OCI7 and OCI8) Es manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Es el conjunto de datos que proporciona la capacidad de almacenar y acude a estos de forma consecuente con un modelo definido como relacional. Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo, soporta unos 17 idiomas, corre automáticamente en más de 80 arquitectura de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos. Su desventaja es el no ser open source.

ORDBMS Object-Relational Database Management System. Esto significa que una base de datos relacional con extensiones objeto, para que los objetos se puedan almacenar en la base de datos su método puede ser llamado directamente del lenguaje query.

Ovrimos Ovrimos SQL Server, is a client/server, transactional RDBMS combined with Web capabilities and fast transactions.

[Referencia

<http://mindprod.com/jgloss/ordbms.html>

<http://www.monografiass.com/monografiass/EpyuVuVFIATWsMTCSL.php>

<http://aspn.activestate.com/ASPN/docs/PHP/ref.ovrimos.html>

]

Perl, Es un lenguaje interpretado que tiene varias utilidades, pero está principalmente orientado a la búsqueda, extracción y formateado de ficheros de tipo texto. También es muy usado para manejo y gestión de procesos.

pg_hba.conf este archivo permite la autenticación de cliente entre el servidor PostgreSQL y la aplicación cliente. Este archivo consiste en una serie de entradas, las cuales verificará los permisos de una máquina para poder conectar a la base de datos solicitada.

PHP, es un lenguaje "open source" interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Pike. Es un lenguaje de programación orientado a objetos, es interpretado, rápido, dinámico, de alto nivel, este lenguaje tiene el *garbage collector* que es un reciclador de memoria dinámica, tiene avanzadas funciones de cadenas, avanzados tipos de datos tales como los arrays.

POP3 Siglas de Post Office Protocol. Protocolo usado por ordenadores personales para administrar el correo electrónico.

PostgreSQL es la base de datos Open Source más avanzada del mundo, es el Sistema administrador de Bases de Datos Relacionales Orientadas a Objetos, está basado en el modelo relacional, aunque incorpora algunos conceptos del modelo Orientado a Objetos, tales como la herencia, usa un subconjunto amplio de comandos SQL como lenguaje de consulta y está implementado siguiendo la arquitectura cliente-servidor. PostgreSQL ofrece gran variedad de herramientas y librerías para acceder a las Bases de Datos.

Postmaster es el proceso que actúa en el servidor como una puerta de control para las peticiones al sistema PostgreSQL.

[Referencia

http://www.geocities.com/SunsetStrip/Backstage/6023/CGI_perl_01.html#Que%20es%20PERL

<http://php.caos.cl/manual/es/introduction.php>

http://fredrik.hubbe.net/pike/tutorial/tutorial_split_introduction.html#preface

<http://www.sobl.org/traduccion/practical-postgres/node40.html>

BIBLIOGRAFÍA

Tanenbaum S. Andrew
Sistemas Operativos Modernos
Editorial Pearson Educación
2001
]

Portabilidad en Bases de datos El diseño deber permitir la implementación del modelo físico en diferentes gestores de bases de datos.

Proceso es un programa en ejecución que consta del programa ejecutable, sus datos y pila, contador, registros, además de toda la información necesaria para ejecutar el programa.

Protocolo El [protocolo](#) es un conjunto de reglas que el navegador ha de cumplir para el proceso de obtener el recurso indicado en el resto del URL. Hay varios tipos de recursos en Internet, y cada uno de ellos tiene su propio protocolo. Un navegador no tiene porqué tener soporte para todos los protocolos, con lo cual hay que conocer de antemano qué recursos es capaz de obtener nuestro navegador.

Python, es denominado lenguaje de 'scripting' en forma similar a Perl, con mayor funcionalidad que el shell. Será el lenguaje para scripting en Koffice. Elementos de C, Lisp, Modula-3.

Ruby, es un lenguaje interpretado para su rápida y fácil programación orientada a objetos. Este lenguaje es simple robusto y extensible.

servidor Un servidor es un proceso que se está ejecutando en un nodo de la red y que gestiona el acceso a un determinado recurso.

Servidores concurrentes: El servidor recoge cada una de las peticiones de servicio y crea otros procesos para que se encarguen de atenderlas. De esta manera el servidor puede estar recogiendo las peticiones de los clientes con mayor velocidad.

Shell es un proceso que lee, interpreta y activa los comandos de una terminal, el shell es la interfaz entre el Kernel y el usuario, es la interfaz cuando el usuario esta sentado frente a una terminal del sistema operativo. Es también un lenguaje de programación y como tal permite el usar variables, estructuras sintácticas

[Referencia

<http://www.ruby-lang.org/en/man-1.4/preface.html>

BIBLIOGRAFÍA

Tanenbaum S. Andrew
Sistemas Operativos Modernos
Editorial Pearson Educación
2001

]

SNMP Es el protocolo simple de administración de redes (SNMP) es un estándar de administración de redes utilizado en redes TCP/IP y, más recientemente, en redes de Intercambio de paquetes entre redes (IPX). SNMP proporciona un método de administración de hosts de redes como concentradores, puentes, enrutadores y equipos de servidor o estaciones de trabajo desde un equipo central que ejecuta un software de administración de redes. SNMP realiza servicios de administración mediante una arquitectura distribuida de sistemas de administración y agentes

sockets Los sockets constituyen la interfaz de comunicaciones en la red. Van a permitir la comunicación entre procesos que se están ejecutando en máquinas remotas, pero integradas en una red. Los sockets corresponden al nivel 4 (transporte) del modelo OSI. Son puntos o mecanismos de comunicación entre procesos que permiten que un proceso hable (emita o reciba información) con otro proceso incluso estando estos procesos en distintas máquinas.

Spam Correo electrónico no solicitado. Se utiliza generalmente para promover productos y/o servicios

SQL: (Structured Query Language) Lenguaje Estructurado de Consulta. Lenguaje utilizado para interrogar y procesar datos en una base de datos relacional. Desarrollado originalmente por IBM para sus macrocomputadoras, han habido muchas implementaciones creadas para aplicaciones de base de datos en mini y microcomputadoras. Las órdenes (mandatos) de SQL se pueden utilizar para trabajar interactivamente con una base de datos, o pueden incluirse en un lenguaje de programación para servir de interfaz a una base de datos

SQL92 es una versión desarrollada por el SQL para acceder bases de datos realizado en 1992

SQL99 se refiere a la realización estándar hecha en 1999 para el estructurado lenguaje query para acceder bases de datos

Sybase Tiene un alto funcionamiento para almacenar los datos. Es un rápido, confiable manejador de base de datos (DBMS), además de ser muy eficiente en términos de utilización de los recursos del sistema operativo pero no es open source.

TCL, Es un lenguaje de programación de comandos muy popular para el desarrollo de pequeñas aplicaciones en entornos Unix, permite programar de forma rápida y sencilla aplicaciones no demasiadas complejas, su velocidad de ejecución no es muy elevada debido a que es un lenguaje interpretado y no es un lenguaje compilado.

TCL/Tk librería gráfica para desarrollar aplicaciones portables sencillas

[Referencia

<http://www.monografiass.com/monografiass/EpyuVuVFIATWsMTCSL.php>

<http://tejo.usal.es/~nines/d.alumnos/sockets3/index.html#DEFINICION>

http://www.sybase.com/about_sybase

http://searchdatabase.techtarget.com/sDefinition/0..sid13_qci561000.00.html]

TCP es un protocolo **orientado a conexión** que transporta de forma segura grupos de octetos (segmentos) **modo duplex** (en los dos sentidos).

TCP/IP (Transmission Control Protocol/Internet Protocol)

Familia de protocolos que hacen posible la interconexión y tráfico de red en Internet

TELENET. es una aplicación que permite la usuario conectarse a un equipo remoto.

USENET Red de acceso público en la Internet que al usuario le proporciona noticias y correo electrónico. Es una placa de anuncios dispersa y gigante absolutamente voluntaria para aquéllos que quieran proporcionar noticias y correo a otros nodos. Comenzó en 1979 como tal entre dos universidades de Carolina del Norte.

UDP: (User Datagram Protocol) Protocolo de Datagrama para Usuario Protocolo TCP/IP que permite que una aplicación envíe un mensaje a una o varias aplicaciones ejecutándose en la máquina destino. La aplicación es responsable de un envío fiable.

UUCP (Unix to Unix Copy Program) Protocolo de AT&T para transferir archivos entre máquinas UNIX. Muy utilizado para copiar news y correo electrónico, aunque tiende a desaparecer y queda reservado a uso local.

Velocis es un poderoso servidor de datos cross-plataforma, es utilizada para aplicaciones industriales web y otras aplicaciones con requisitos de funcionamiento exigentes. Velocis es robusta para desarrollo de aplicaciones.

[Referencia

<http://www-gris.det.uvigo.es/~sha/tcl.pdf>

<http://www.uucp.org/info.shtml>

<http://www.uco.es/ccc/glosario/glosario.html>

<http://www.faqs.org/faqs/usetnet/what-is/part1/>

<http://sal.kachinatech.com/H/1/VELOCIS.html>

WAIS o [Wide Area Information Service](#) es una red de sistemas de edición, diseñados para distribuir datos a cualquier tipo de cliente. Este sistema ha sido creado exclusivamente para las páginas Web, y está pensado para preparar índices de grandes bases de datos, que faciliten la búsqueda de información.

WWW Siglas de World Wide Web. La WWW provee una manera de enlazar las computadoras en Internet a través del código html y usando hipervínculos que le permiten avanzar de un sitio a otro en la Web.

<http://www.uco.es/ccg/glosario/glosario.html>

Bibliografía

PostgreSQL

Momjian, Bruce
PostgreSQL : introduction and concepts
Boston, Massachusetts : Addison-Wesley, 2001
462 p.

PHP

Welling Luke, Thomson Laura
PHP and MySQL Web development
867 p

Greenspan Jay, Bulger Brad
MySQL-PHP database applications
596 p

Hilton Craig, Willis Jeff
Building database applications on the web using PHP3
597 p

Rasmus Lerdorf and Kevin Tatroe ; with Bob Kaehms and Ric McGredy
Programming PHP
507 p
2002

Otros

Tanenbaum S. Andrew
Sistemas Operativos Modernos
Editorial Pearson Educación