



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES  
"ARAGÓN"**

**DESARROLLO E IMPLEMENTACIÓN DE  
SISTEMAS CON SOFTWARE LIBRE, CENTRO DE  
INFORMACIÓN TELEFÓNICA.**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE :  
INGENIERO EN COMPUTACIÓN  
P R E S E N T A :  
EDGAR MARTÍNEZ RICO**

**ASESOR:  
ING. JUAN GASTALDI PÉREZ**

**MÉXICO**

**2006**





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.

## **Agradecimientos**

### **A mi Familia**

---

**Con todo mi amor, en agradecimiento por la comprensión y confianza siempre brindados para la realización de mi carrera profesional.**

**Sabiendo que la única forma de agradecerles es no defraudarlos, quiero que sientan que el objetivo logrado también es suyo, y que la fuerza que me ayudó a conseguirlo fue su apoyo.**

## Índice

Introducción	1
1. Sistema Operativo	2
1.1. Software Libre	3
1.2. Sistema Operativo Linux	2
1.2.1. ¿Qué es Linux?	3
1.2.2. Un poco de Historia de Linux	4
1.3. Características de Linux	5
1.3.1. Multitarea	5
1.3.2. Multiusuario	6
1.3.3. Shell Programable	6
1.3.4. El Sistema X Window	7
1.3.5. Soporte TCP/IP	7
1.3.6. Memoria Virtual y Librerías Compartidas	8
1.3.7. Distribuciones	8
1.4. El Kernel de Linux	8
1.5. El Servicio Principal de Linux	10
2. Apache	11
2.1. ¿Qué es un Servidor WEB?	12
2.2. ¿Qué es Apache?	13
2.2.1. ¿Qué es el Apache Group?	13
2.2.2. ¿Qué es el Proyecto Apache?	14
2.3. Características de Apache	15
2.4. ¿Cómo trabaja Apache?	17
2.5. Arquitectura de Apache	18
3. PHP	20
3.1. ¿Qué es una aplicación WEB?	21
3.2. ¿Qué es PHP?	22
3.3. Características de PHP	23
4. MySQL	26
4.1. ¿Qué es MySQL?	27
4.2. Características de MySQL	27
5. Aplicación de Centro de Información Telefónica	32
5.1. ¿Qué es DGSCA?	33
5.2. Organigrama de las Sedes de DGSCA	33
5.3. Enfoque de CIT	34
Conclusiones	37
Bibliografía	38

## Introducción

El objetivo principal de este trabajo es el conocer las herramientas de software libre y sus principales características, empleadas en el desarrollo del sistema Centro de información Telefónica (CIT), realizado en la Dirección General de Sistemas de Cómputo Académico (DGSCA). Así como mostrar la problemática que resuelve dicho sistema, utilizando casos de uso para el planeamiento y descripción del problema.

Aunque existen otras herramientas utilizadas en el desarrollo de este proyecto como Java, Power Designer, Dreamweaver, entre otras. Solo se mencionaran las siguientes:

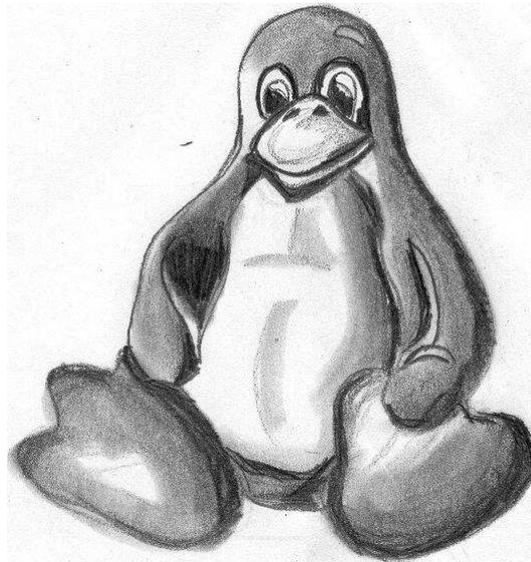
- Linux.
- Apache.
- PHP.
- MySQL

Este trabajo se encuentra constituido por cinco capítulos, donde los primeros cuatro capítulos se encuentran dedicados al conocimiento y características de las herramientas, de tal forma que el capítulo número uno **Linux** habla de las de que es Linux y sus características, el capítulo número dos **Apache** dedicado a dicho servidor Web, el capítulo número tres **PHP** dedicado al lenguaje de programación y el capítulo número cuatro **MySQL** dedicado al Servidor de Base de Datos (propriadamente el RDBMS Relational Database Management System) MySQL.

El capítulo número cinco denominado **Aplicación Centro de Información Telefónica**, plantea la problemática a la cual da solución el proyecto, así como los casos de uso que se emplearon para el desarrollo del sistema **CIT** (*Centro de Información Telefónica*).

## Capítulo 1

### Sistema Operativo Linux



#### Objetivos:

- Conocer que es el software libre.
- Conocer que es Linux, como surge y cuáles son sus principales características.
- Entender qué es el kernel
- Conocer el servicio principal de Linux

En este primer capítulo se explica a que tipo de software se le denomina software libre debido que todas las herramientas empleadas son de este tipo, se enfoca principalmente en conocer qué es Linux, como surge y cuáles son sus principales características.

Además sabrá que el kernel es el corazón de Linux y por ende la parte más importante, no importando con que distribución de Linux se desee trabajar, conocerá cual es el servicio padre de todos los sistemas Linux y cuál es su función en el sistema Linux.

## 1.1. **Software Libre**

El término de Software Libre no se refiere a que este software sea gratis, si no que más bien se refiere a la libertad de su uso es decir, que para que un software sea libre debe estar bajo una licencia que se conoce como GNU.

Las siglas GNU (es un acrónimo de no es UNIX) GPL General Public License muchas veces llamado copy left. Esta licencia fue creada por Free Software Foundation para permitir a los programadores escribir software libre, la palabra libre se refiere a libertad de los siguientes tres aspectos principales:

- 1 Permite que el autor original conserve sus derechos de autor.
- 2 Permite que otros tomen el software y lo modifiquen, o que se a parte de otro programa.
- 3 Permite que otros redistribuyan o revendan el software, o modifiquen versiones del software.

## 1.2. **Sistema Operativo Linux**

### 1.2.1. ¿Qué es Linux?

Linux es una versión gratuita de UNIX desarrollado por Linus Trovalds en la Universidad de Helsinski Finlandia. Linux estuvo disponible en Noviembre de 1991 en su primera versión la 0.10, y en Diciembre de 1991 fue lanzada la versión 0.11.

Linux es un sistema operativo para la plataforma Intel (computadoras personales). El sistema ha sido diseñado y construido por cientos de programadores alrededor del mundo. La meta ha sido crear un clon de UNIX, libre de cualquier licencia comercial (como lo es copy night), que cualquiera en el mundo lo pueda utilizar. Linux inició como un hobby de Linus Trovalds, cuando era estudiante de la Universidad de Helsinski en Finlandia. Su meta era crear un reemplazo del sistema operativo MINIX, el cual es un sistema operativo parecido a UNIX para computadoras personales.

Una de sus mayores ventajas es que es fácilmente portable a diferentes tipos de computadoras, por lo que existen versiones para casi todos los tipos de computadoras, desde **PC** y **Mac** hasta estaciones de trabajo y supercomputadoras. A diferencia de otros sistemas operativos, como por ejemplo **MacOS** (Sistema operativo de los **Apple Macintosh**), Linux no está pensado para ser fácil de emplear, sino para ser sumamente **flexible**. Por lo tanto **Linux** no es en general tan sencillo de emplear como otros sistemas operativos, aunque, se están realizando grandes esfuerzos para facilitar su uso. Pese a todo la enorme flexibilidad de **Linux** y su gran estabilidad (y el bajo coste) han hecho de este sistema operativo una opción a considerar por aquellos usuarios que se dediquen a trabajar a través de redes, naveguen por

**Internet**, o se dediquen a la programación. Además el futuro de **Linux** es brillante y cada vez más y más gente y más y más empresas (entre otras **IBM**, **Intel**, **Corel**) están apoyando este proyecto, con lo que el sistema será cada vez más sencillo de emplear y los programas serán cada vez mejores.

### 1.2.2. Un poco de Historia de Linux

La historia de Linux se encuentra muy apegada a la historia de UNIX. Como ya se mencionó Linux nace a partir de un programa llamado MINIX creado por el científico en cómputo Andrew Tannenbaum. Pero antes de seguir con MINIX veamos un poco de la historia de UNIX.

Cuando Novell Inc. compra el sistema Unix a AT&T (American Telephone and Telegraph Corporation) en 1993, adquieren la marca registrada reconocida como UNIX y, por lo tanto el control de los derechos de quienes usan el nombre de UNIX con propósitos comerciales. Esta es una razón por la cual no se le puede llamar Linux UNIX.

Aunque AT&T fue el creador del sistema operativo UNIX, muchas otras compañías han intentado implementar la idea básica por años. Ken Thompson quien era programador de computadoras, fue quien trabajo para AT&T Bell Laboratories, y en 1969 un grupo de personas bajo la dirección de Ken desarrollaron un sistema operativo que era flexible y completamente compatible con las distintas necesidades de los programadores.

Berkeley Software Distribution (Universidad de Berkeley California) liberaron su propia versión de UNIX, basado en AT&T versión 7, en 1978. BSD UNIX, como es bien conocido en la industria, contiene mejoras desarrolladas por la comunidad académica de Berkeley, diseñado para hacer un UNIX más usuario-amigable.

Las mejoras de usuario-amigable en BSD UNIX fue un intento por hacer un UNIX mas común (amigable). Con un ambiente mas avanzado para los programadores, quienes les gusta la flexibilidad conforme a sus demandas cambiantes. A pesar de su compatibilidad menor al 100% con el UNIX original de AT&T. BSD se ha convertido en el estándar académico Los creadores originales de BSD desde entonces liberado una versión para la plataforma Intel, llamada BSD.

Coherent otro clon de UNIX es distribuido por la compañía de Mark Williams. El producto es llamado Coherent y en su momento fue el clon de UNIX mas barato disponible.

USL, ó UNIX System Laboratories, fue una compañía de AT&T que ha estado desarrollando el sistema operativo UNIX desde 1980. Antes de que este fuera comprado por Novell en 1993. USL produce el código fuente para todos los sistemas UNIX V. derivados en la industria. Sin embargo USL por si solo no es esta a la venta como producto.

La última versión de USL fue la versión UNIX System V versión 4.2 marco su entrada al mercado, junto con Novell quienes temporalmente crearon una compañía llamada Univel, USL produjo una versión llamada UnixWare. Mas tarde Novell compro a USL quien forma ahora parte de Novell Unix System Group, con la finalidad de producir el código fuente para UnixWare.

Microsoft desarrollo su propia versión de UNIX, llamada XENIX a finales de la década 1970 y principio de la década 1980 en la cumbre de la PC. Cuando la capacidad de procesamiento de las mini computadoras empezó a competir con las PCs y con la llegada de los microprocesadores 80386 de Intel, fue evidente que XENIX el cual había sido desarrollado específicamente para las PCs ya no era necesario. Microsoft y AT&T juntaron XENIX y UNIX en un nuevo sistema operativo llamado System V/386 versión 3.2, el cual podía operar prácticamente en cualquier hardware con una configuración en particular.

Sun Microsystems ha contribuido ampliamente en la venta de UNIX promocionando su sistema SunOS. Sun trabajo junto con AT&T para producir un versión compatible con BSD la cual fue UNIX System V versión 4.0.

IBM se aventuro en el mundo de UNIX con su producto llamado AIX (Advanced Interactive Executive). A pesar de que AIX no era tan conocido como otras versiones de UNIX.

Linux comenzó como un hobby de Linus Trovalds un estudiante de ciencias de la computación. Quien esperaba crear una versión mas robusta de UNIX para los usuarios de MINIX, el cual fue desarrollado por el Dr. Andrew Tannenbaum. MINIX fue desarrollado para demostrar distintos conceptos de ciencias de la computación en el área de sistemas operativos. El Dr. Andrew Tannenbaum incorporo dichos conceptos en un sistema que imitaba a UNIX. El programa estaba disponible extensivamente para todos los estudiantes en el mundo y pronto despertó interés. Linus Trovalds proveyó una mejor plataforma para los usuarios de MINIX la plataforma de Intel 80386.

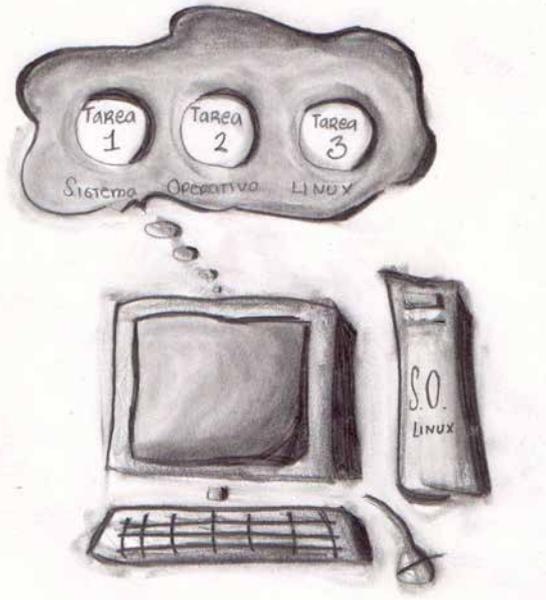
## **1.3. Características de Linux**

### **1.3.1. Multitarea**

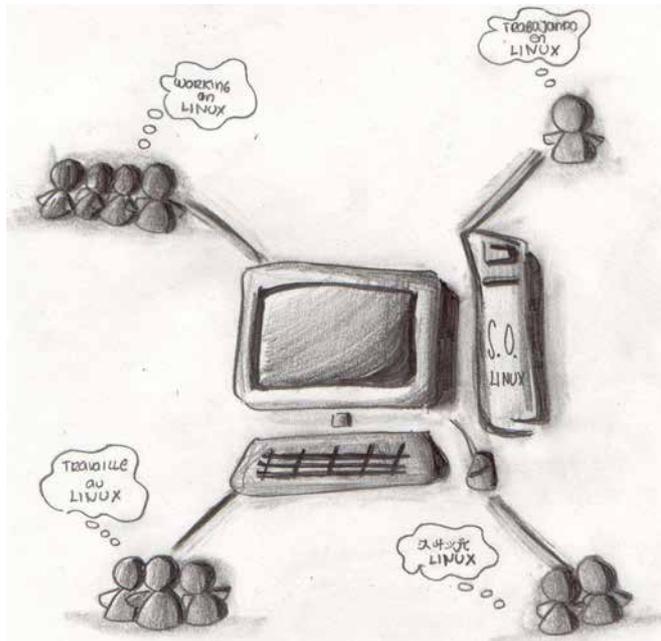
La palabra multitarea describe la capacidad de poder ejecutar múltiples programas al mismo tiempo, sin detener cada aplicación de su ejecución. Es llamado multitarea por que cada aplicación tiene la oportunidad de ejecutarse, cada programa se ejecuta hasta que el sistema operativo permita a otro programa ejecutarse. El microprocesador en la computadora puede ejecutar solo una instrucción a la vez, pero es capaz de ejecutar distintas tareas en periodos de tiempo muy pequeños. Por ejemplo un microprocesador de 25 a 90 MHz, se refiere a que es capaz de transferir de 25 a 90 millones de bits por

segundo. Cuando el procesador completa un grupo de instrucciones, la velocidad es mucho mayor, típicamente 30 nanosegundos. La mente humana no puede detectar la diferencia en periodos de tiempo tan cortos como estos ocurriendo simultáneamente. Por ello da la ilusión de que todas las tareas se realizan al mismo tiempo, aunque no es así.

Una de las principales ventajas de un sistema multitarea es la reducción de los tiempos muertos, quizá la mayor de las ventajas de LINUX es el acceso al código fuente del sistema operativo. Ya que si se poseen los conocimientos suficientes se puede modificar de manera parcial o total de acuerdo a sus necesidades.



### 1.3.2. Multiusuario



Linux es capaz de permitir que el microprocesador asigne tiempo de ejecución a diferentes aplicaciones, así mismo permite a diferentes usuarios, ejecuten una o más aplicaciones en un mismo instante. La verdadera ventaja de que Linux sea multitarea y multiusuario es que permite que más de un usuario pueda trabajar en una versión específica de un programa al mismo tiempo, desde la misma terminal o en terminales separadas.

### 1.3.3. Shell Programable

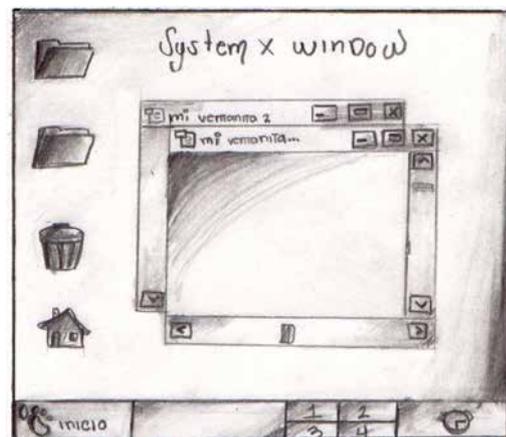
El shell programable es otra de las características que hace a UNIX, y a Linux el sistema operativo más flexible disponible. Aunque muchas versiones de UNIX y Linux incluyen más de un tipo de shell, todos ellos esencialmente realizan lo mismo. Un shell trabaja como un intérprete entre el usuario y el kernel (el cual es el corazón o cerebro de sistema operativo Linux).

Quizás el escenario más simple para ayudarnos a comprender la programación del shell de Linux es el proceso en segundo plano. Por ejemplo, se debe de respaldar todas las partes del sistema en un momento, para evitar accidentes de borrado de archivos indeseados. El proceso de respaldo puede requerir una cantidad de tiempo considerable si los archivos y directorios son considerables en tamaño.

Con un simple programa de shell que consiste en una o quizás dos líneas de comandos, el proceso de respaldo de la información del sistema puede ejecutarse concurrentemente con otra operación que quizá desee realizar (ya que el proceso de respaldo lleva algo de tiempo). Después de programar el shell e iniciar el proceso de respaldo, se puede simplemente ejecutar alguna aplicación deseada. Cuando el proceso es finalizado, Linux envía una señal del que el trabajo ha sido realizado.

### 1.3.4. El Sistema X Window

El sistema *X Window* es la interfaz grafica para las máquinas UNIX. Linux cuenta con su propia versión completa del sistema *X Window* conocida como *XFree86*. La cual es una interfaz grafica muy poderosa, es capaz de soporta muchas aplicaciones. Por ejemplo se puede tener múltiples sesiones de login en diferentes ventanas en la pantalla al mismo tiempo.



### 1.3.5. Soporte TCP/IP



Este es un conjunto de protocolos que millones de universidades y negocios soportan dentro del *World Wide Web* mejor conocida como Internet. Con una conexión Ethernet, se puede tener acceso a Internet o a una red de área local. Utilizando SLIP (serial line protocol) se puede tener acceso a Internet utilizando como medio una línea telefónica.

### 1.3.6. Memoria Virtual y Librerías Compartidas

Linux puede utilizar una porción del disco duro como memoria virtual, expandiendo el monto total de su memoria RAM disponible. Linux también implementa librerías compartidas. Permitiendo a los programas el uso de subrutinas en tiempo de ejecución ubicadas en dichas librerías. Esto permite ahorro de espacio en su sistema. Ya que cada aplicación no necesita almacenar una copia de ellas.

### 1.3.7. Distribuciones

Una de los primeros conceptos que aparecen al iniciarse en **Linux** es el concepto de **distribución**. Una **distribución** es un agrupamiento del núcleo (**kernel**) del sistema operativo **Linux** (la parte desarrollada por **Linus Torvalds**) y otra serie de aplicaciones de uso general o no tan general. En principio las empresas que desarrollan las distribuciones de **Linux** están en su derecho al cobrar una cierta cantidad por el software que ofrecen, aunque en la mayor parte de las ocasiones se pueden conseguir estas distribuciones desde **Internet**, de **revistas** o de **amigos**, siendo todas estas formas **gratuitas** y **legales**.

Las distribuciones más conocidas son **RedHat**, **Debian**, **Slackware**, **SuSE** y **Corel Linux**, todas ellas incluyen el software más reciente y empleado lo cual incluye compiladores de C/C++, editores de texto, juegos, programas para el acceso a **Internet**, así como el entorno gráfico de **Linux: X Window**.

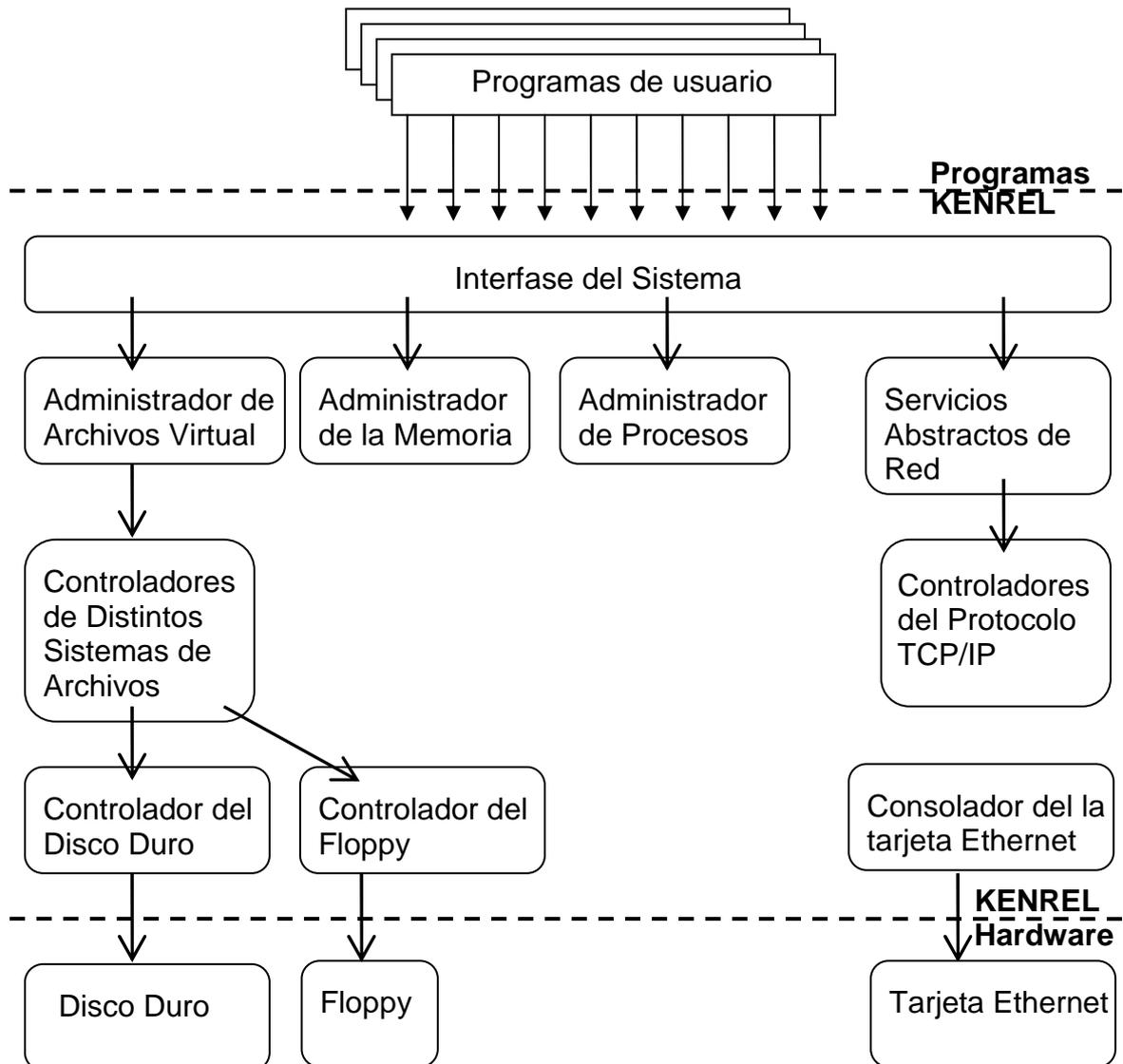
## 1.4. El Kernel de LINUX

Como se mencionó anteriormente el sistema operativo Linux cuenta con diferentes distribuciones, todas ellas difieren en cuanto el software adicional que puedan proporcionar, pero la parte que no cambia de distribución a distribución es el kernel el cual es el corazón del sistema operativo Linux.

El sistema operativo Linux consiste de un **kernel** y de otros programas. El **kernel** es el corazón del sistema operativo Linux, mantiene los archivos en el disco duro, ejecuta los programas y los mantiene en ejecución de manera concurrente. Asigna memoria y otros recursos a los diferentes procesos, recibe y envía paquetes a través de la red, entre otras tareas.

El kernel es relativamente pequeño pero provee todas las herramientas con las cuales cualquier servicio puede ser construido. Además previene a cualquiera del acceso directo al hardware, forzando a cualquiera a utilizar las herramientas que provee. Dichas herramientas pueden utilizarse a través de llamadas al sistema. Algunos servicios de software que provee el kernel tienen propiedades similares y por lo tanto pueden ser abstraídas en clases. Por ejemplo los diferentes protocolos de red han sido abstraídos en una interfase de programación la cual es BSD socket library.

El kernel de Linux se compone de diversas partes importantes como: el administrador de procesos, el administrador de la memoria, los controladores de los dispositivos de hardware, el sistema de archivos, el administrador de red, entre otros, A continuación se muestra algunas partes del kernel:



Probablemente las partes más importantes del kernel son el administrador de memoria y el administrador de procesos. El administrador de memoria se encarga de la asignación de las localidades de memoria así como el intercambio de dichas localidades de memoria con los procesos. El administrador de procesos crea procesos e implementa la capacidad de multitarea a través del intercambio de procesos activos en el procesador.

## 1.5. El Servicio Principal de LINUX

El servicio más importante en un sistema LINUX es proveído por **init**. **Init** es el primer proceso del sistema LINUX, como la última cosa que el núcleo hace cuando arranca. Cuando **init** empieza, continúa la inicialización de los procesos realizando varias tareas iniciales (verificar e instalar el o los sistemas de archivos, poniendo en marcha los distintos demonios, etc.).

La lista exacta de cosas que **init** hace depende del tipo; existen varios tipos a escoger. **Init** provee el concepto de **single user mode**, en el que nadie puede entrar al sistema y *root* usa una consola de shell generalmente. El modo acostumbrado es llamado el **multiuser mode**. A estos tipos se les conoce como **run levels** (niveles de ejecución).

Linux permite hasta 10 **run levels**, 0 - 9, pero generalmente solamente algunos de éstos son definidos de manera general. El **run level 0** lo es definido como **halt system** (Alto de sistema). El **run level 1** es definido como **single user mode** (El usuario modo solo). El **run level 6** es definido como **system reboot** (Reinicio de sistema). Otros **run levels** están en función de cómo su distribución los ha definido, y varían significativamente entre las distribuciones. Si se consulta el archivo **/etc/inittab** se puede tener un idea de cómo los **run levels** se encuentran definidos.

En la operación normal, **init** se asegura de que *getty* esté trabajando (**getty** permite que los usuarios entren al sistema), así como adoptar a los procesos huérfanos (son aquellos procesos cuyo padre se ha muerto; en un sistema UNIX todo proceso debe estar en un árbol; así los procesos huérfanos deben ser adoptados).

Cuando el sistema es cerrado, **init** es el que se encarga de matar a todos los procesos, desinstalar el sistema de archivos y parar el procesador, así como alguna otra tarea que le haya sido asignada a través del archivo de configuración **/etc/inittab**).

## Capítulo 2

### Apache



#### Objetivos:

- Comprender el funcionamiento de un servidor Web.
- Saber que es apache así como quien los desarrolla.
- Conocer sus características principales
- Entender cual es su funcionamiento básico.
- Conocer su Arquitectura.

En este capítulo se pretende dar a conocer que es apache, quien lo desarrolla, conocer sus características principales que lo han convertido en el servidor Web mas utilizado en la Web según las estadísticas proporcionadas por el Netcraft. Se vera su funcionamiento pero de manera básica, además se explicará como se encuentra construido apache. Pero para poder entender que es Apache en primera instancia se explicará que es un servidor Web.

## 2.1. ¿Qué es un Servidor WEB?

Un servidor Web es un software detrás del World Wide Web. Está pendiente de la o las solicitudes de un cliente, como un navegador como Netscape o el Internet Explorer de Microsoft. Cuando consigue una, procesa dicha petición y devuelve algunos datos. Estos datos toman forma de una página organizada con el texto y gráficos generalmente. El navegador representa estos datos de la mejor manera y lo presenta al usuario. Los servidores Web en concepto son programas muy simples. Se encuentran en espera de peticiones y cumple cada una de ellas cuando las recibe.

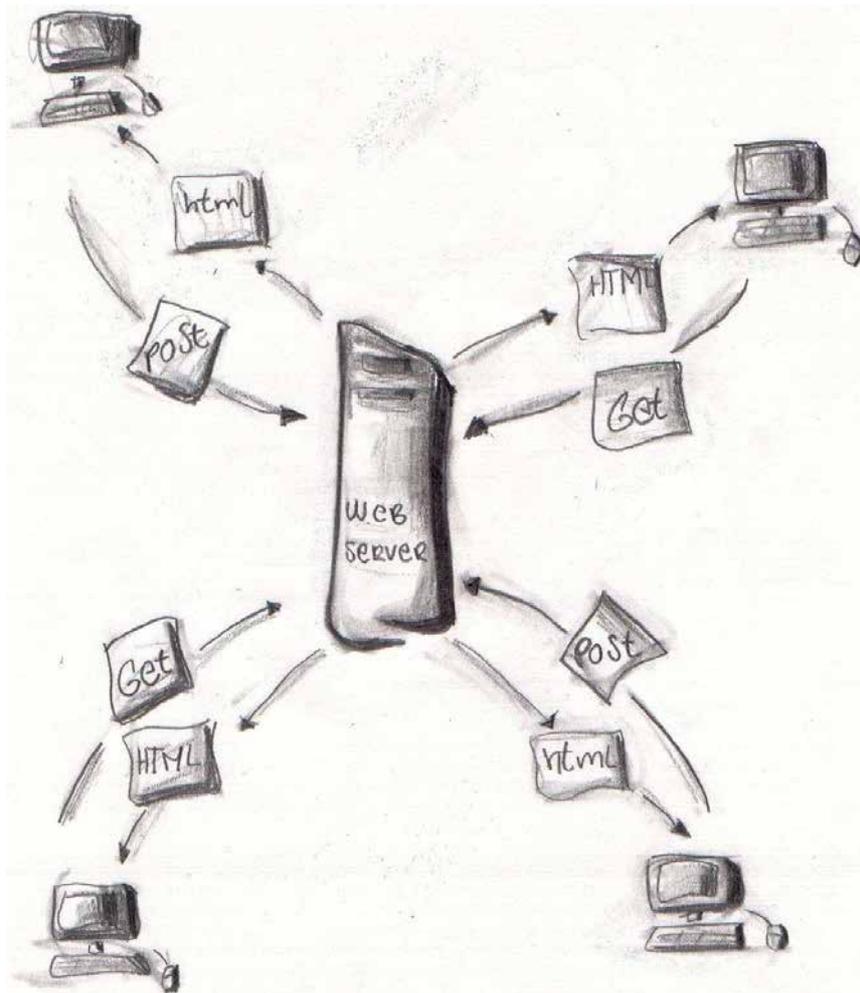
Los servidores de Web se comunican con navegadores u otros clientes que usan el protocolo de transferencia de hipertexto (el HTTP), que es un protocolo simple que normaliza la manera en que las solicitudes son enviadas y procesadas. Esto permite que una gran variedad de clientes se comuniquen con el servidor no importando cual es el distribuidor, evitando los problemas de compatibilidad.

La mayoría de los documentos pedidos son formateados usando lenguaje de marcado de hipertexto (el HTML). El HTML es un subconjunto pequeño de otro lenguaje de marcación llamado lenguaje de marcación estandar general (el SGML por sus siglas en inglés Standar General Markup Language), que se utiliza ampliamente por muchas organizaciones.

El HTML es el alma de la WEB. HTML (por sus siglas en ingles HiperTex Markup Language) es el lenguaje de marcación de hipertexto usado para formatear el texto. Los navegadores interpretan esta información e interpretan dándole la mejor apariencia dependiendo de sus habilidades. La característica más importante de HTML es que permite vincular a documentos diferentes y a otros recursos en la WEB.

El hipertexto permite que un usuario consulte otros documentos guardados en la misma computadora o en una computadora ubicada en una parte diferente del mundo.

El proceso de recuperación de información es totalmente transparente para el usuario; es fácil. Provee una experiencia de exploración perfecta de documentos y servicios. Permite la búsqueda de información fácilmente y lo presenta en una manera que es fácil de digerir. Es gráfico, y puede combinar con sonido y video.



## 2.2. ¿Qué es Apache?

El proyecto apache es un proyecto enfocado al desarrollo de un servidor HTTP por un grupo de personas denominadas como el Apache Group.

### 2.2.1. ¿Qué es el Apache Goup?

Es un grupo de personas enfocadas al desarrollo del servidor Web apache. Uno de sus miembros, Rob Hartill, una vez sarcásticamente describió a Apache Group de la siguiente manera:

***“El Apache Group es: Un grupo personas individuales talentosas que están tratando perfeccionar el arte de nunca terminar algo”***

## 2.2.2. ¿Qué es el Proyecto Apache?

El proyecto Apache es una colaboración para el desarrollo e implementación de un servidor HTTP (WEB). El proyecto es conjuntamente administrado por un grupo de voluntarios en todo el mundo, utilizando el Internet y Web para comunicarse, planear y desarrollar el servidor, así como la documentación relacionada. Estos voluntarios son conocidos como el “**Apache Group**”. Cientos de usuarios han contribuido con ideas, código y documentación acerca del proyecto.

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas UNIX (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPD 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a *patchy server* (un servidor *parcheado*).

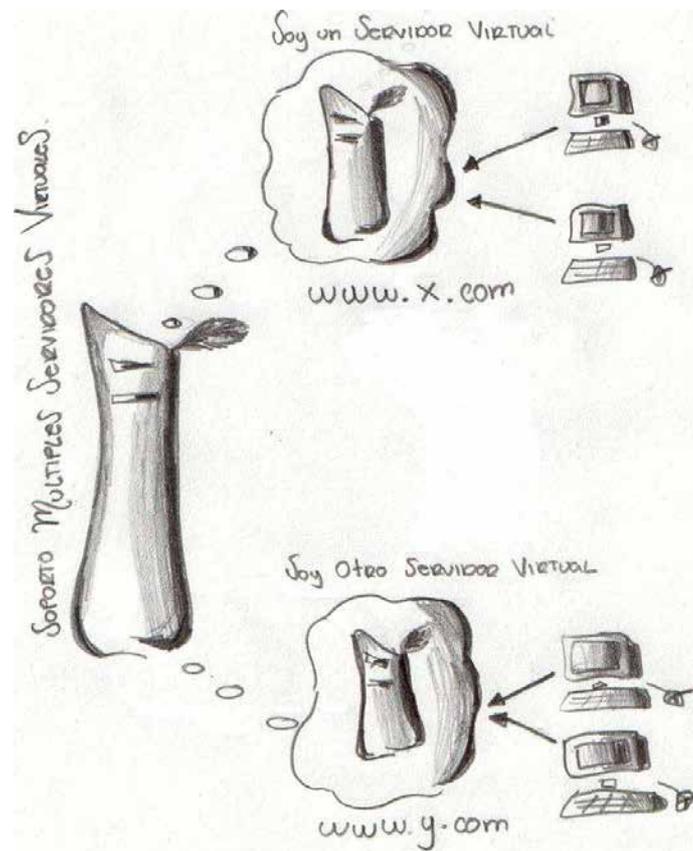
El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor HTTP con el 68% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

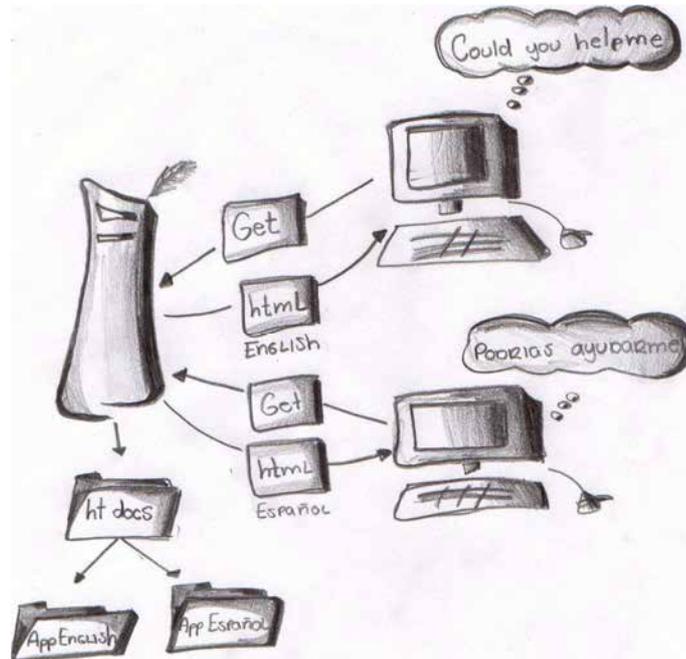
## 2.3. Características de Apache

- El servidor WEB Apache se encuentra disponible para casi cualquier plataforma. Apache se encuentra disponible de dos formas precompilado (binario) Y en código fuente. Apache se encuentra disponible para las siguientes plataformas:
  - AUX
  - BSDI 2.0
  - FreeBSD
  - HP-UX
  - IRIX
  - LINUX
  - NetBSD
  - OS/2
  - Solaris
  - SunOS
  - UnixWare
  - Windows

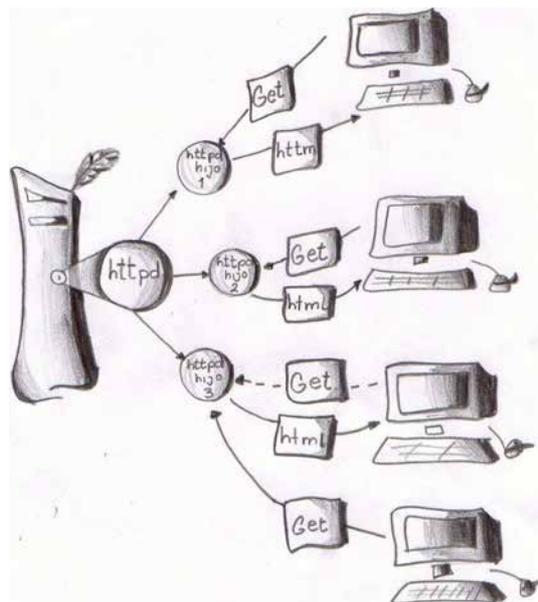
- La configuración de Apache es sencilla. Apache utiliza un solo archivo de configuración global (el cual se llama httpd.conf y se encuentra en la carpeta /conf donde se instaló Apache), el cual se encuentran configurado por defecto. Para poder configurar Apache solo se necesita un editor de texto como vi, emacs o pico en los Sistemas UNIX. El archivo de configuración es procesado cuando se levanta Apache y cada vez que este se reinicia.
- Apache provee de distintos métodos de Autenticación de usuarios, desde un archivo plano, por medio de una base de datos. Módulos de Terceros para permitir la interfase con distintos sistemas de autenticación de red.
- Apache provee servidores virtuales. Un servidor virtual atiende distintas peticiones de diferentes dominios, por ejemplo **http://www.x.com** y **http://www.y.com**. Ambos residen en la misma computadora y son atendidos por el mismo proceso (servidor).



- El servidor Apache puede negociar automáticamente la recuperación del contenido con el navegador y puede proporcionar el contenido más apropiado de varios sistemas de documentos equivalentes. Esto permite que el servidor recupere el documento que tiene la mejor representación que el cliente está dispuesto a aceptar. Por ejemplo cuando apache proporciona documentos en español e Inglés, dependiendo de la configuración del navegador. Apache proporcionara automáticamente la versión más apropiada.



- El servidor Apache cuenta con un modelo de pre-bifurcación (pre-forking) que administra y configura la cantidad de procesos hijos. Un modelo de pre-bifurcación es aquel donde los procesos del servidor se encuentran en espera de peticiones. La mayoría de los servidores inician un proceso solo cuando se ha recibido una petición. Una vez que el servidor ha terminado la petición el proceso muere, y este ciclo se presenta cada vez que existe una petición. Este proceso de bifurcación puede ser muy costoso. Ya que dependiendo de la plataforma este puede tomar demasiado tiempo la creación de un proceso. El modelo de pre-bifurcación con el cuenta apache permite una respuesta mucho más rápida y eficiente.



## 2.4. ¿Cómo Trabaja Apache?

Apache es un programa que se ejecuta con un sistema operativo multitarea apropiado. El proceso de apache se llama **httpd** y es el proceso que debe estar presente para que el servidor pueda empezar a atender peticiones, normalmente la peticiones se realizan en el puerto 80 que es el puerto por defecto. Un sitio web se compone principalmente de cuatro directorios.

*conf:*

El directorio *conf* contiene el archivo de configuración *httpd*.

*htdocs:*

Contiene los scripts de HTML que serán proporcionados a los clientes del sitio. Este directorio y aquellos debajo de él, son accesibles para alguien en la Red

*logs:*

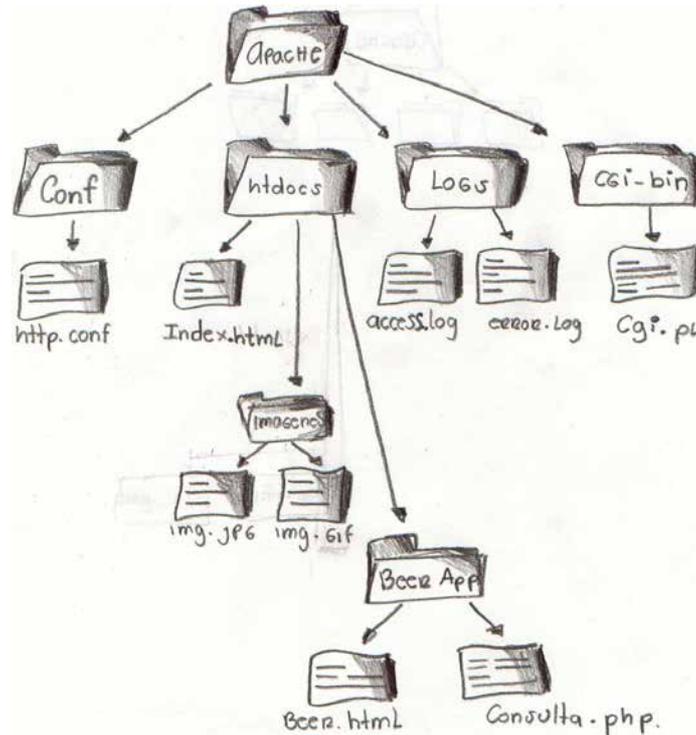
Contiene las bitácoras tanto de acceso como la de errores.

*cgi-bin:*

Contiene los scripts de CGI. Éstos son programas o bombardear scripts escritos por o para el administrador de la web que puede ser ejecutado por Apache de parte de sus clientes.

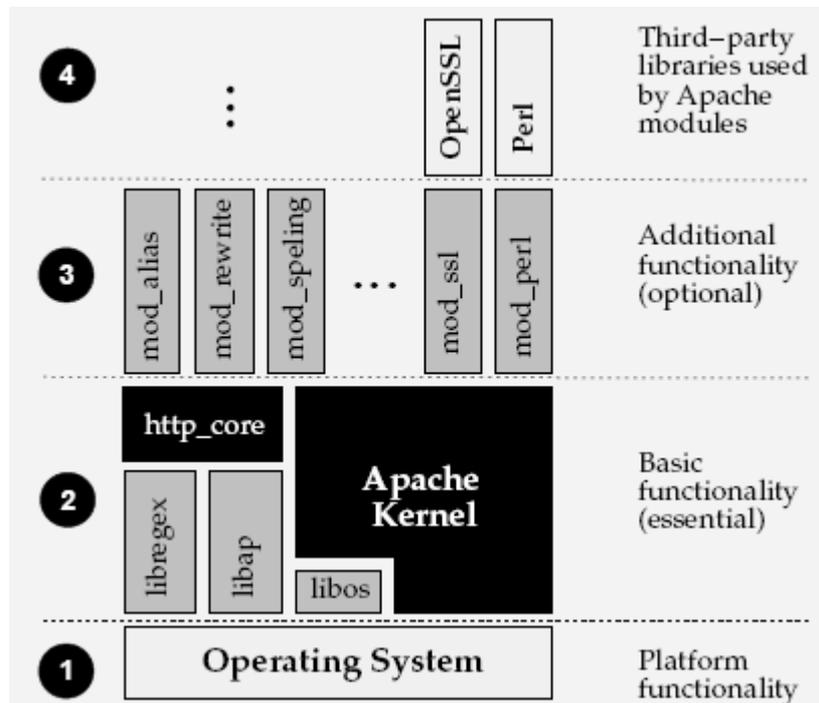
El proceso de Apache (**httpd**) se encuentra en espera escuchando el puerto 80 (normalmente aunque se puede especificar otro en el archivo de configuración *httpd.conf*). Cuando una petición aparece en el puerto, recibe la petición *HTTP* y analiza el contenido de la cabecera. Entonces aplica las reglas que se encuentra en el archivo de configuración (*httpd.conf*) y realiza las acciones apropiadas.

Las reglas y configuraciones del servidor apache que se encuentran en el archivo *httpd.conf* se llevan acabo mediante directivas, apache cuenta con alrededor de 200 directivas para su administración.



## 2.5. Arquitectura de Apache

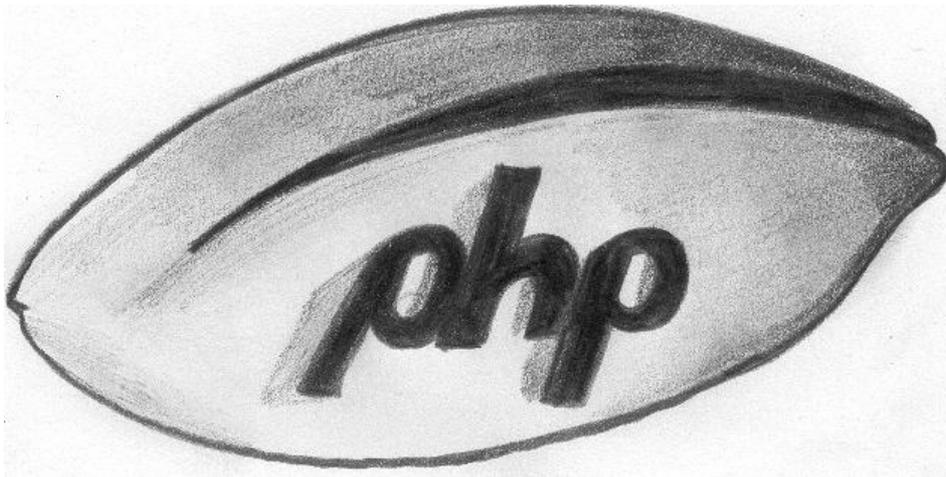
Apache se encuentra construido en una arquitectura de cuatro capas como se muestra continuación:



- 1. Sistema operativo:** La funcionalidad básica es proveída por el sistema operativo subyacente. Para Apache, este sistema operativo es típicamente UNIX, pero él también puede ser Win32, OS/2, el MacOS, o incluso el sistema POSIX de un mainframe.
- 2. Apache Kernel, Módulo de Núcleo, y Bibliotecas de Núcleo:** La capa 2 es el programa principal de Apache, constando de un núcleo de Apache, Un módulo de núcleo, y algunas bibliotecas usuales. El núcleo de Apache, con el módulo de núcleo especial (httpd\_core), implementa la funcionalidad del servidor HTTP básica y provee la (API) interfaz de programación de aplicaciones de Apache a la capa de módulo. Esta capa también contiene una biblioteca de la clave genérica, reutilizable (libap), una biblioteca que implementa expresiones regulares que analiza gramaticalmente y combina (libregex), una biblioteca de abstracción de sistema operativo pequeña (libos).
- 3. Módulos de Apache:** La grandiosa funcionalidad usuario-visible hace a Apache único entre los servidores Web existentes es provista por muchos módulos de Apache sobre la capa 3. Generalmente cada módulo implementa una funcionalidad evidentemente separada. En realidad, ningún módulo es requerido. Operar un servidor de Web mínimo capaz solamente de proporcionar documentos estáticos en un área de documento configurada es posible incluso sin cualquier modulo.
- 4. Bibliotecas de Terceros:** Para los módulos usuales (aquellos encontrados en la distribución de Apache oficial), esta capa está generalmente se encuentra vacía. Los módulos adicionales, como **mod\_ssl** y **mod\_perl**, usan bibliotecas de terceras partes externas, sin embargo; estas bibliotecas pueden encontrarse sobre esta capa de la arquitectura de Apache.

## Capítulo 3

### PHP



#### Objetivos:

- Entender el funcionamiento de una aplicación Web.
- Conocer que es PHP
- Conocer sus principales características.

En este capítulo conocerá que es PHP, cuáles son sus características y entenderá por qué es tan empleado a pesar de que no es un lenguaje puramente orientado a objetos y a pesar de que no es tan robusto como la plataforma J2EE de Sun Microsystems.

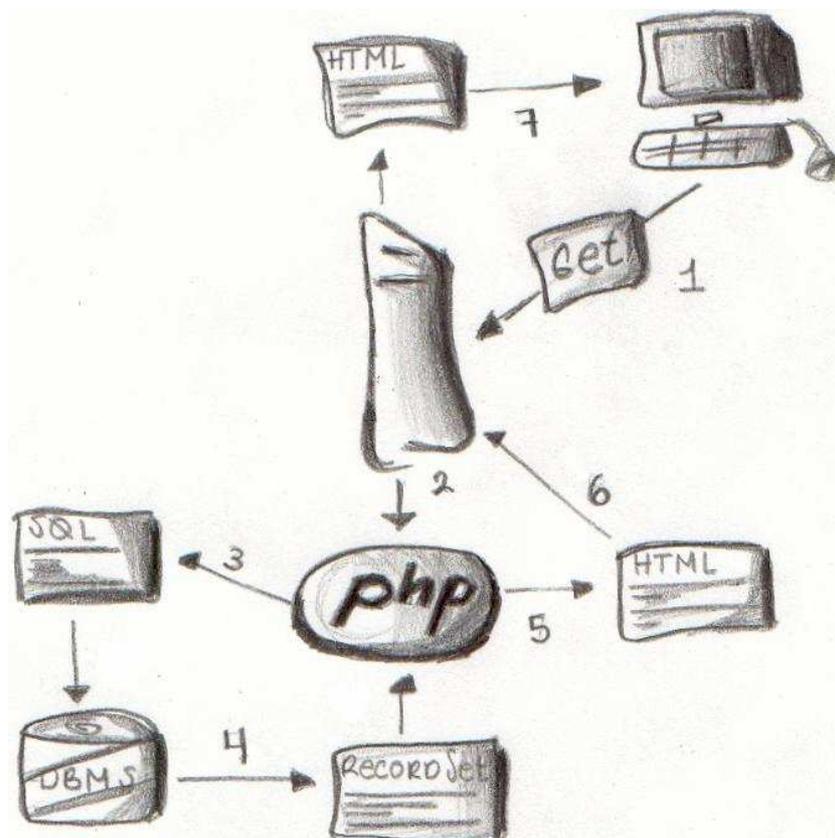
Para poder entender que es PHP primero se verá que es una aplicación Web

### 3.1. ¿Qué es una aplicación Web?

Una aplicación Web consiste básicamente en generar documentos HTML dinámicos, por dinámicos se debe entender a que dichos documentos de HTML no existen como tal en el servidor Web si no que se generan en el momento en que son solicitados.

El concepto en si es sencillo, el procedimiento es el siguiente el servidor Web está en espera de una petición, que el cliente genera por medio de un navegador Web, una vez que se presenta el servidor Web se encarga de procesar la información y de generar un documento de HTML, y devolver el documento de HTML al navegador.

Como podemos ver el procedimiento es sencillo pero falta entrar en detalle acerca del proceso que se realiza dentro del Servidor Web, una vez que existe una petición en el servidor Web este debe contar con un intérprete, el cual es el que realmente se encarga de atender a la petición del cliente, este intérprete es el encargado de generar el documento de HTML; el intérprete se encarga de realizar consultas a un servidor de Base de Datos (DBMS), una vez que el DBMS atiende a la consulta del intérprete, el DBMS se encarga de procesar la consulta y una vez que realiza este proceso, el DBMS devuelve un record set que no es más que un área de memoria donde coloca los resultados de la consulta. Una vez que el intérprete cuenta con el record set ahora sí continúa con el proceso de generación de la pagina Web, una vez que el intérprete termina la elaboración del documento de HTML, este se la entrega al servidor Web, el cual se encarga de enviarla al cliente que realizo dicha petición.



1. El cliente a través del navegador (como Internet Explorer, Netscape, Mozilla Fire Fox, Opera) genera una petición.
2. El servidor Web (como por ejemplo Apache, IIS, JBoss), invoca al intérprete el cual se encarga de generar la página de HTML dinámica.
3. El intérprete para poder generar la página de HTML, es capaz de realizar consultas a un DBMS (como SQL Server, MySQL, PostgreSQL).
4. El DBMS envía el resultado de la petición al intérprete en un *record set* (se debe entender como un conjunto de registros en un arreglo bidimensional, que contiene los campos como las columnas y los registros como los renglones del arreglo).
5. Una vez que el intérprete obtiene el resultado de la consulta de SQL continúa generando la página dinámica de HTML.
6. Una vez que el intérprete termina de generar la página de HTML, esta es proporcionada al servidor Web.
7. Ya que el servidor Web cuenta con la página de HTML esta es proporcionada al cliente que realizó la petición.

### 3.2. ¿Qué es PHP?

*PHP* (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor.

Una respuesta corta y concisa, pero, ¿qué significa realmente? A continuación se muestra un ejemplo:

```
<html>
<head>
  <title>Ejemplo</title>
</head>
<body>
  <?php
  echo "Hola ; soy un script PHP!";
  ?>
</body>
</html>
```

Puede apreciarse que no es lo mismo que un script escrito en otro lenguaje de programación como Perl o C -- En vez de escribir un programa con muchos comandos para crear una salida en HTML, escribimos el código HTML con cierto código PHP embebido (incluido) en el mismo, que producirá cierta salida (en nuestro ejemplo, producirá un texto). El código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un script similar al de nuestro ejemplo en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los archivos HTML con PHP.

Lo mejor de usar PHP es que es extremadamente simple, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales.

Aunque el desarrollo de PHP está concentrado en la programación de scripts en el lado del servidor.

### **3.3. Características de 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 enviar y recibir cookies. Y esto no es todo, se puede hacer mucho más.

Existen tres campos en los que se usan scripts escritos en PHP. Scripts del lado del servidor. Este es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto funcione. El intérprete PHP (CGI ó módulo), un servidor web y un navegador. Es necesario correr el servidor web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectándose con el servidor web.

Scripts en la línea de comandos. Puede crear un script PHP y correrlo sin ningún servidor web o navegador. Solamente necesita el intérprete PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en \*nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesamiento de texto.

Escribir aplicaciones de interfaz gráfica. Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, pero si conoce bien PHP, y quisiera utilizar algunas características avanzadas en programas clientes, puede utilizar PHP-GTK para escribir dichos programas. También es posible

escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.

PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, O'Reilly 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.

De modo que, con PHP tiene la libertad de elegir el sistema operativo y el servidor de su gusto. También tiene la posibilidad de usar programación procedimental o programación orientada a objetos. Aunque no todas las características estándar de la programación orientada a objetos están implementadas en la versión actual de PHP, muchas bibliotecas y aplicaciones grandes (incluyendo la biblioteca PEAR) están escritas íntegramente usando programación orientada a objetos.

Con PHP no se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF y películas Flash (usando libswf y Ming) sobre la marcha. También puede presentar otros resultados, como XHTML y archivos XML. PHP puede auto generar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla.

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 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
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

También contamos con una extensión DBX de abstracción de base de datos que permite usar de forma transparente cualquier base de datos soportada por la extensión. Adicionalmente, PHP soporta ODBC (el Estándar Abierto de Conexión con Bases de Datos), así que puede conectarse a cualquier base de datos que soporte tal estándar.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets puros. PHP soporta WDDX para el intercambio de datos entre lenguajes de programación en Web. Y hablando de interconexión, PHP puede utilizar objetos Java de forma transparente como objetos PHP Y la extensión de CORBA puede ser utilizada para acceder a objetos remotos.

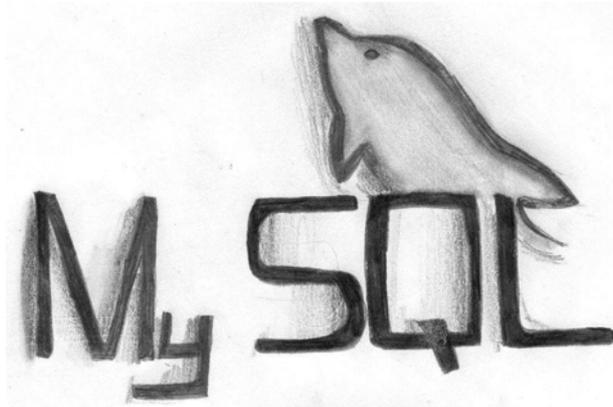
PHP tiene unas características muy útiles para el procesamiento de texto, desde expresiones regulares POSIX extendidas o tipo Perl hasta procesadores de documentos XML. Para procesar y acceder a documentos XML, soportamos los estándares SAX y DOM. Puede utilizar la extensión XSLT para transformar documentos XML.

Si usa PHP en el campo del comercio electrónico, encontrará muy útiles las funciones Cybercash, CyberMUT, VeriSign Payflow Pro y C CVS para sus programas de pago.

Para terminar, contamos con muchas otras extensiones muy interesantes, las funciones del motor de búsquedas mnoGoSearch, funciones para pasarelas de IRC, utilidades de compresión (gzip, bz2), conversión de calendarios, traducción.

## Capítulo 4

### MYSQL



#### Objetivos:

- Conocerá que es MySQL.
- Conocerá sus principales características.

En este capítulo conocerá que es MySQL así como sus principales características que provee y entender por que este Sistema Manejador de Bases de Datos Relacional se ha vuelto tan popular y ampliamente utilizado.

#### 4.1. ¿Qué es MySQL?

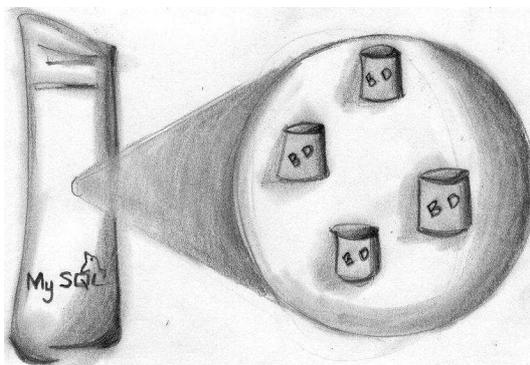
MySQL, pronunciado "mai Ess Cu Ei," es código abierto (open source), es un software de Nivel Empresarial, un Sistema Manejador de Base de Datos Relacional, multi-hilos. Esta es la mejor definición de MySQL.

MySQL fue desarrollado por una firma consultora en Suecia llamada TcX. Ellos tenían la necesidad de un sistema de la base de datos que fuera extremadamente rápido y flexible. Desafortunadamente, no encontraron ningún **Sistema Manejador de Bases de Datos (SMBD)** en el mercado que hiciera lo que ellos deseaban. Así pues, crearon MySQL, que se basa otro en otro **SMDB** llamado **mSQL**. El producto que crearon es rápido, confiable, y extremadamente flexible. Se utiliza en muchos lugares a través del mundo. Los principales usuarios de MySQL son Universidades, proveedores de servicios de Internet y organizaciones no lucrativas, principalmente por que su precio es muy bajo por no decir que es libre.

La razón del crecimiento de MySQL es el advenimiento del movimiento de código abierto (open source) en la industria de cómputo. El movimiento de código abierto, es el resultado de varios vendedores de software que proporcionan no solamente el producto sino también el código fuente de su producto. Esto permite que los usuarios vean cómo su programa funciona y que puedan modificarlo donde requieran un ajuste. Esto aunado a la popularidad de Linux, ha dado generado un crecimiento en el uso de los productos de código abierto en el mundo de negocios.

#### 4.2. Características de MySQL

- **MySQL es un Software Enterprise (Empresarial):**



Un software de nivel empresarial en el área de los negocios, es un software capaz de interactuar con distintos sistemas con la finalidad de conseguir una meta en común. Algunas aplicaciones que se encuentran en este nivel de negocio son SAP, el servidor de Microsoft SQL, el de Oracle 10g, y el Adaptive Server de

Sybase. Las aplicaciones que existen a este nivel de negocio tienden a tener ciertas características. Son generalmente multiusuario, es decir varios usuarios pueden utilizar al mismo tiempo dicha aplicación. Otra

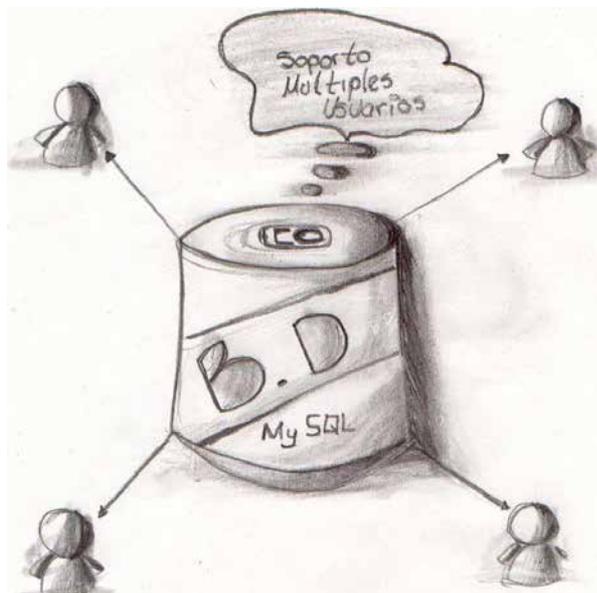
característica es que proporcionan una ciertos mecanismos de seguridad. La característica principal es que ser muy flexibles.

La primera característica de un software de nivel Empresa es que puede ser utilizada por más de una persona a la vez. Esto es un requisito a este nivel del negocio. Más de una persona puede necesitar tener acceso a la información del negocio en un momento dado. Esto es crítico para que el negocio funcione con éxito. MySQL resuelve este requisito. Puede tener hasta 101 conexiones simultáneas. Esto no significa que solamente 101 personas pueden utilizar este uso. Significa que puede tener 101 conexiones de manera concurrente que es un poco diferente. Una conexión es el tiempo que toma para que un usuario reciba los datos que ha solicitado.

La característica siguiente que debe tener es seguridad. Al ocuparse de la información misión crítica, solamente la gente con autorización debe ser permitida acceder a la información. La seguridad mantiene a gente malévolos fuera de la aplicación; sin ella, los desastres pueden suceder. MySQL conoce este requisito. Muy pocas bases de datos en la misma clase que MySQL pueden comparar al nivel de la seguridad que MySQL proporciona. Ya que MySQL tiene la característica de que un usuario puede acceder a una base de datos o a ciertas tablas de una base de datos desde una dirección IP específica, o desde cualquier dirección IP.

- **MySQL es un Sistema Manejador de Bases de Datos:**

MySQL es más que una base de datos. Es un Sistema Manejador de Bases de Datos (SMBD), ya que un Sistema Manejador de Bases de Datos soporta múltiples usuarios realizando peticiones de manera concurrente, así como múltiples bases de datos en un mismo servidor, controla quién puede utilizarlo y como puede manipularlo. Lleva un bitácora de las acciones realizadas y se ejecuta continuamente como proceso en segundo plano (como servicio).

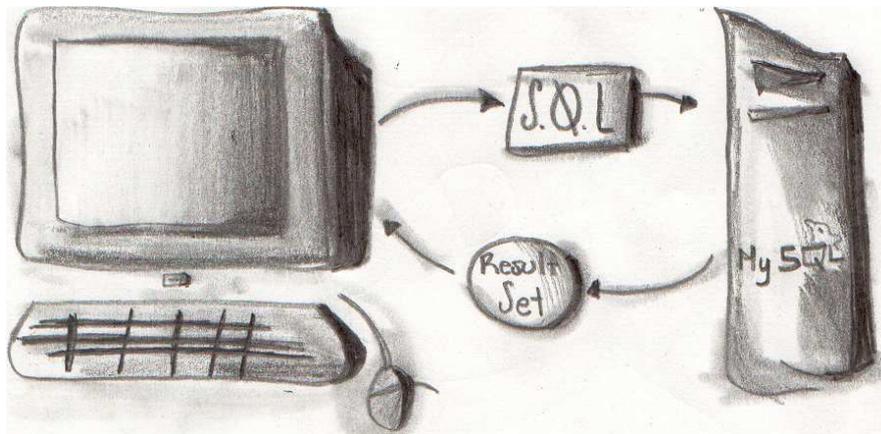


- **MySQL es un Sistema Manejador de Bases de Datos Relacional (SMDBR):**

Un Sistema de Base de Datos Relacional almacena datos en tablas separadas, en lugar de poner todos los datos en una sola tabla. Esto agrega velocidad y flexibilidad. Es decir que MySQL es capaz de soportar lo **constraints** de tipo **foreign key**, es decir validaciones de llave foránea.

- **MySQL soporta el Lenguaje Estructurado de Consultas (SQL por sus siglas en ingles Structured Query Language):**

El nombre de MySQL se compone de SQL debido a que hace referencia al soporte de SQL. SQL es el lenguaje estandarizado usado para tener acceso a bases de datos y es definido por el estándar de ANSI/ISO SQL. El estándar de SQL se ha estado desarrollando desde 1986 y existen varias versiones.



- **MySQL es un software Open Source (código abierto):**

Código abierto (Open Source) significa que es posible que cualquier persona pueda utilizar y modificar el software. Que cualquiera pueda descargar el software de MySQL del Internet y utilizarlo sin pagar. Que cualquiera que desee, pueda estudiar el código fuente y cambiarlo para satisfacer sus necesidades. El software de MySQL se encuentra bajo la licencia **GPL**.

- **MySQL trabaja en una arquitectura Cliente/Servidor:**

MySQL trabaja en un arquitectura cliente/servidor que consiste en múltiples hilos de SQL en el servidor, que atienden a diferentes clientes, programas, librerías, herramientas de administración.

- **MySQL cuenta con diferentes APIs (interfaces de programación de aplicaciones):**

MySQL cuenta con APIs para Perl, TCL, python, C/C++, Java (JDBC), ODBC y .NET.

- **MySQL es Multiplataforma:**

Otra característica de MySQL es su portabilidad. Ya que existe una versión para casi cada plataforma.

- **MySQL soporta diferentes tipos de tablas:**

A diferencia de otros manejadores de bases de datos MySQL permite que al momento de crear una tabla el usuario elija el tipo de tabla que desea utilizar. Cada tipo de tabla tiene diferentes características. A continuación se muestran los tipos de tablas más importantes.

- Mylsam-Static:** Se utilizan cuando todas las columnas tienen su tamaño predeterminado. Son demasiado rápidas para realizar INSERT, UPDATE y DELETE. Por otra parte la seguridad de datos es absolutamente alta

- Mylsam-Dynamic:** Se utilizan cuando alguna de las columnas posee alguno de los siguientes tipos Varchar, xxxText, or xxxBlob, los campos no se encuentran secuenciados, esto incrementa los tiempos de acceso.

- Mylsam-Compresed:** este tipo de tablas se obtienen después de haber utilizado el programa *mysamchk* ocupan menos de la mitad.

- BDB (Berkley Data Base):** Este tipo de tablas permiten el manejo de transacciones es decir los comandos de BEGIN, COMMIT y

ROLLBACK. Además deben contar con una llave primaria obligatoria en cada tabla.

- **InnoDB:** Al igual que BDB soportan transacciones además del row level locking que soporta Oracle y en la versión 4.1 son las únicas tablas que soportan integridad referencial.
  
- **Merge:** Son una unión de una o mas tablas de tipo MyISAM se utilizan cuando el sistema manejador de archivos del sistema operativo es pequeño.
  
- **HEAP:** Este tipo de tablas solo existen en RAM, este tipo de tablas no soportan los tipos text, blob, las búsquedas se realizan con operadores relacionales, no soportan AUTO\_INCREMENT y se borran cuando se da de baja el servidor.

## Capítulo 5

### Aplicación de Centro de Información Telefónica



#### Objetivos:

- Conocer qué es DGSCA.
- Conocer el organigrama de las sede de DGSCA.
- Entender la problemática a la que da solución del CIT.

En este capítulo se plantea la problemática a la cual da solución el Centro de Información Telefónica.

Para poder comprender la problemática es necesario conocer el organigrama de las sedes de DGSCA; para entender la problemática que soluciona CIT. Para así poder plantear los principales casos de uso que se emplearon en el desarrollo de CIT.

## 5.1. ¿Qué es DGSCA?

La Dirección General de Servicios de Cómputo Académico de la UNAM es la entidad universitaria encargada de la operación de los sistemas centrales de cómputo académico y de las telecomunicaciones de la institución; su esfuerzo más amplio es la capacitación en tecnología de la información, de prospección e innovación y de asimilación de estas tecnologías en beneficio de la Universidad y de la sociedad en general. Para poder llevar a cabo su objetivo DGSCA se compone de ocho sedes:

- Dirección de Sistemas.
- Ciudad Universitaria.
- Centro de Estudios Multidisciplinario Polanco.
- Centro Mascarones.
- Centro Coapa.
- Centro Nuevo León.
- Centro San Agustín.
- Sede Palacio de la Autonomía.

## 5.2. Organigrama de las Sedes de DGSCA

El organigrama de las sedes de DGSCA se compone en primera instancia del coordinador de la sede que es el encargado de conseguir contratos con los diferentes clientes como el gobierno del Distrito Federal, el ISSTE, etc.; para llevar a cabo la capacitación de su personal.

Debajo del Coordinador del centro se encuentran tres departamentos los cuales son:

- Informes:

Este departamento se encarga de la difusión de los diferentes cursos que DGSCA imparte, brindar informes a las personas interesadas en dichos cursos. Brinda informes al público interesado en inscribirse en un curso acerca de los trámites que se tienen que realizar. Brinda informes al público acerca de los cursos programados en el calendario trimestral.

- Control Escolar.

Se encarga de la contratación, pago a los profesores para los diferentes cursos. Se encarga también de emitir las constancias tanto para el profesor como para los alumnos.

- Infraestructura.

Se encarga de proveer las aulas con las computadoras suficientes, dispositivos y con el software requerido para los diferentes cursos que se imparten.

### **5.3. Enfoque de CIT**

La Aplicación Centro de Información Telefónica se enfoca principalmente al departamento de Informes con la finalidad de apoyar en algunas de sus funciones que este realiza.

Para comprender mejor la problemática que resuelve la aplicación veamos a detalle cuales son las funciones principales que realiza el departamento de Informes.

Como se menciona anteriormente la función principal del departamento es la difusión de los cursos que DGSCA imparte, los cuales se programan cada trimestre y se publican en un calendario. El calendario de cursos varia de centro a centro de tal manera que el calendario que se publica en el centro Nuevo León, es diferente del calendario publicado en el centro San agustín. El calendario cuenta con la siguiente información:

- Costo del curso
- Duración en numero de horas.
- Nombre del Curso
- Fecha de Inicio
- Fecha de Fin
- Hora de Inicio del Curso.
- Hora de Fin del Curso.
- Días que se imparte el curso.

Para tener una idea mas clara acerca de esta información a continuación se muestra una imagen del calendario que se publica y que se puede obtener cuando una persona acude al departamento de informes de cualquier centro de DGSCA.

# CURSOS DE CÓMPUTO 2006



INSCRIPCIÓN: De lunes a viernes de 9 a 15 y de 16 a 19 horas.  
 INFORMES: De lunes a viernes 9 a 19 horas y sábados de 9 a 14 horas.  
 Informes@autonomia.dgsca.unam.mx

Tels: 55223348 y 55223485

AUTONOMÍA

Código	No. hrs.	CURSOS	PRIMER TRIMESTRE						LUNES A VIERNES						SABATINOS						
			CURSOS						Fecha de inicio de curso						Fecha de inicio de curso						
			ENE 19	ENE 30	FEB 13	FEB 27	MAR 13	MAR 27	ENE 14	ENE 28	FEB 11	FEB 25	MAR 11	MAR 25	ENE 14	ENE 28	FEB 11	FEB 25	MAR 11	MAR 25	
<b>NOVEDADES</b>																					
\$ 5,528	20	Programación avanzada con JAVA																			
\$ 5,528	20	Programación avanzada con Visual Basic .NET																			
\$ 4,480	30	Diseño arquitectónico avanzado con ArchiCAD																			
\$ 3,084	20	Sistema integral para el registro contable COI avanzado																			
\$ 3,084	30	Elaboración y registro de nómina NOI avanzado																			
\$ 2,490	30	Sistema de administración empresarial SAE																			
\$ 4,580	30	Sistema de administración empresarial SAE avanzado																			
\$ 1,677	20	Taller de aplicaciones financieras con EXCEL																			
\$ 1,735	30	Introducción a la computación e Internet con WINDOWS																			
\$ 1,735	30	Introducción a la computación e Internet con LINUX																			
\$ 1,677	20	Manejo del asistente digital personal - PDA *																			
\$ 4,580	30	Sistema operativo tipo LINUX																			
\$ 3,084	20	Utilerías de LINUX																			
\$ 4,580	30	Administración del Sistema Operativo UNIX																			
\$ 1,677	20	Ambiente gráfico LINUX																			
\$ 1,677	20	Ambiente gráfico WINDOWS																			
\$ 3,084	20	Taller de mantenimiento preventivo a Pc's de escritorio y portátiles																			
\$ 3,084	20	Introducción a las telecomunicaciones																			
\$ 3,084	20	Redes de computadoras																			
\$ 3,084	20	Servicios de red en LINUX																			
\$ 3,084	20	Introducción al conjunto de protocolos de Internet TCP / IP																			
\$ 1,999	30	Introducción al diseño de bases de datos																			
\$ 3,084	20	SQL estándar básico																			
\$ 3,084	20	SQL estándar avanzado *																			
\$ 3,084	20	Uso del manejador de bases de datos VISUAL FOXPRO																			
\$ 3,084	20	Desarrollo de aplicaciones con VISUAL FOXPRO																			
\$ 1,677	20	Uso del manejador de bases de datos ACCESS																			
\$ 3,084	20	Desarrollo de aplicaciones con ACCESS																			
\$ 4,580	30	Uso del manejador de bases de datos SQL SERVER																			
\$ 1,999	30	Introducción a la programación																			
\$ 6,106	40	Lenguaje de programación JAVA																			
\$ 6,106	40	Lenguaje de programación PHP																			
\$ 6,106	40	Lenguaje de programación Visual Basic .NET																			
\$ 6,106	40	Lenguaje de programación C																			
\$ 6,106	40	Lenguaje de programación C ++																			
\$ 4,580	30	Lenguaje de programación Perl																			
\$ 4,580	30	Análisis y diseño orientado a objetos con UML																			
\$ 2,490	30	Elaboración de gráficos avanzados con CORELDRAW																			
\$ 3,084	20	Introducción a la edición de formatos de imágenes digitales																			
\$ 1,677	20	Manipulación de imágenes con PHOTOSHOP																			
\$ 1,677	20	Manipulación avanzada de imágenes con PHOTOSHOP																			
\$ 3,084	20	Creación de imágenes con ILLUSTRATOR																			
\$ 1,677	20	Elaboración de presentaciones con POWERPOINT																			
\$ 1,677	20	Presentaciones avanzadas con POWERPOINT																			
\$ 839	10	Elaboración y edición de archivos con ACROBAT																			
\$ 4,580	30	Animación en tres dimensiones mediante 3D STUDIOMAX																			
\$ 3,084	20	Efectos digitales para video, multimedia y Web con AFTER EFFECTS *																			
\$ 4,580	30	Creación de animación con FLASH																			
\$ 3,084	20	Taller de digitalización de medios																			
\$ 3,084	20	Elaboración de páginas Web con HTML																			
\$ 5,528	20	Creación de páginas Web dinámicas con XML																			
\$ 3,084	20	Creación de páginas Web con DREAMWEAVER																			
\$ 2,490	30	Edición de imágenes con FIREWORKS																			
\$ 4,580	30	Diseño en dos dimensiones con AUTOCAD																			
\$ 3,084	20	Diseño en tres dimensiones con AUTOCAD																			
\$ 4,580	30	Diseño arquitectónico con ArchiCAD																			
\$ 1,677	20	Sistema integral para el registro contable COI																			
\$ 1,677	20	Elaboración y registro de nómina NOI																			
\$ 1,677	20	Hoja de cálculo EXCEL																			
\$ 1,677	20	Técnicas avanzadas con EXCEL																			
\$ 1,677	20	Procesador de texto WORD																			
\$ 1,677	20	Técnicas avanzadas con WORD																			
<b>MÓDULOS DE ACTUALIZACIÓN PROFESIONAL</b>																					
\$ 4,981	60	Módulo en cómputo, Word, Excel, PowerPoint e Internet																			
<b>PROGRAMA DE CAPACITACIÓN PARA EL TRABAJO</b>																					
\$ 10,251	130	Línea de capacitación en cómputo para el trabajo en la oficina																			

■ Horarios 9-11, 11-13, 13-15, 15-17 y 17-19  
 \* Consultar fechas y horarios

¡Si no está programado el curso que quieres **Arma tu propio grupo!**

Con cinco participantes o más, de Lunes a Viernes entre las 11:00 y 19:00 hrs. asistiendo 2 hrs. diarias.  
**¡Preguntanos cómo!**

El hecho de que un curso se encuentre en el calendario no garantiza que se impartirá. Ya que para que un curso se imparta este debe contar con un mínimo de participantes de 5 personas.

Además de la difusión de los cursos publicados en el calendario, el departamento de informes se encarga de llevar un registro de todas aquellas personas interesadas en inscribirse en algún o algunos de los cursos. Con la finalidad de que si en algún momento se imparte, se tengan sus datos necesarios y la posibilidad de avisarle por teléfono, para conocer si dicha persona se interesa en tomar el curso.

El proyecto de CIT surge como un proyecto, para la creación de un Call Center, es decir un centro de Información Telefónica único para todo DGSCA, el cual brinde la información necesaria sobre todos los calendario publicados en los diferentes centros, de manera que solo se tenga un área de informes para todo DGSCA.

## Conclusiones

Actualmente existen un gran número de aplicaciones de Software libre y un gran número de empresas que respaldan este tipo de aplicaciones, las aplicaciones que se mencionan en este trabajo solo son un ejemplo de las muchas que existen en el mercado.

Dichas aplicaciones son bastante robustas y cada vez más utilizadas y aceptadas por corporaciones lucrativas y no lucrativas. Podemos mencionar a empresas como Sun Microsystems y ORACLE las cuales recientemente a entrado en este ámbito. En el caso particular del gobierno del DF cuentan con una gran variedad de soluciones, las cuales se encuentran desarrolladas con aplicaciones de Software Libre.

Podemos ver un futuro donde quizás la mayoría del Software sea libre y donde quizás el negocio ya no sea el desarrollo del Software, sino más bien los servicios que estos puedan producir, tales como el mantenimiento, documentación, capacitación, etc.

## Bibliografía

Steve Suehring, *MySQL Bible*, New York EUA, Wiley Publishing, 2002, 686 pp.

Ralf S. Engelschall, *Apache Desktop Reference*, EUA, Addison-Wesley, 2000, 182 pp.

Ben Laurie and Peter Laurie, *Apache the Definitive Guide*, EUA, O'Reilly, 1999, 370 pp.

Michael Glass, Yann Le Scouarnec, *Beginning PHP, Apache, MySQL Web Development*, EUA, Wiley Publishing, 2004, 700 pp.

Janet Valade, *PHP & MySQL for Dummies*, EUA, Wiley Publishing, 2004, 440 pp.

Jay Greenspan and Brad Bulger, *MySQL/PHP Database Applications*, M&T Books, 2001, 575 pp.

Mark Maslakowski, *Sam's Teach yourself MySQL in 21 days*, EUA, SAMS Publishing, 2000, 532 pp.

Tim Converse and Joyce Park with Clark Morgan, *PHP5 AND MySQL*, EUA, Wiley Publishing, 2004, 1042 pp.

Kamran Husuin, Timothy Parker, *Linux Unleashed*, EUA, Sams Publishing, 1995, 700 pp.

Jack Tackett, David Gunter, *Special Edition Using Linux*, EUA, QUE Corporation, 1995, 700 pp.

*Apache Server*, EUA, Sams Publishing, 1996, 400 pp.

Craig Larman, *UML y Patrones Segunda Edición*, Madrid, Pearson Educación, 2003, 624 pp.