

**UNIVERSIDAD NACIONAL DEL CENTRO DE LA PROVINCIA DE
BUENOS AIRES – TANDIL**

FACULTAD DE CIENCIAS EXACTAS

**INGENIERIA EN INFORMATICA Y SISTEMAS COMPUTACIONALES
(CONVENIO UNICEN – UTC)**



TEMA DE TESIS

“Sistema de Gestión de Comercios Basados en Arquitectura Web”

Nombre:

Edwin Edison Quinatoa

Tutor:

Ing. Hugo Curtí

Fecha: 18 de Julio del 2007

**TANDIL – ARGENTINA
2007**

Del contenido de la presente tesis se responsabiliza el autor:

A handwritten signature in black ink, consisting of several loops and a horizontal line across the top, positioned above a dotted line.

Edwin E. Quinatoa Arequipa

RESUMEN

En estos últimos años, los inconvenientes que surgen en el área administrativa comercial, por dificultades de organización y manejo generado manualmente, han ocasionado una excesiva pérdida de tiempo al usuario, en el proceso de la alta del registro (entrada) de los artículos, y su venta (salida).

Analizando todo esto, y con el adelanto de nuevas herramientas tecnológicas como el Internet y el Software libre, se crea un “Sistema de Gestión de Comercios, basados en Arquitectura Web” para la Ferretería. “EL SIGLO” en la Ciudad de Tandil. El cual permitirá llevar un mejor registro de sus artículos, por lo que la idea fundamental es lograr que mediante la construcción de una aplicación Web, se pueda controlar eficazmente el acceso a toda la información que se requiera.

Para ello se utilizó el Sistema Operativo GNU/Linux (Ubuntu), el servidor de base de datos PostgreSQL, servidor Web Apache, y el generador dinámico de páginas PHP. Ya que el objetivo es utilizar plataformas de Software libre y Arquitectura Web con habilidad; y operando estas herramientas se crea este Sistema, el cual va a permitir las mejoras en tareas, del ingreso de datos en cada una de las entidades como son: proveedores, artículos, marcas, categorías, y a la vez permitiendo visualizar todo lo integrado, ya sea de forma general o individualmente, y los remitos que se hayan creado.

El control de cada identidad, se realiza a través de la búsqueda por código o nombre, permitiendo tener una mejor organización y uso del tiempo. Con este trabajo realizado permitirá dar mayor flexibilidad y un mejor flujo de la información, a través de su diseño y las restricciones en la base de datos, generaran mayor seguridad en la Información para las personas que utilicen el Sistema.

ABSTRACT

In the last few years the inconveniences occurred in the administrative commercial area, by the organization difficulties and manually generated management, have caused an excessive waste of time to the user in the registering process entrance and exit(sale)of items.

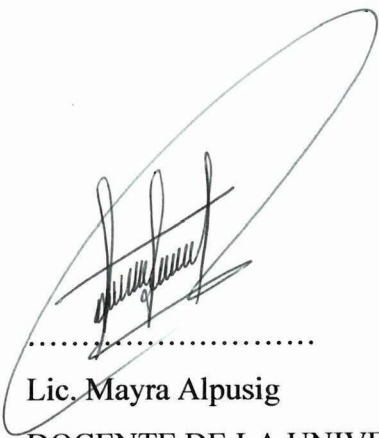
Analyzing all and with the development of new technological tools like internet and free software, a “Bussiness Management System, based in Web Architecture” created for the Hardware shop “El Siglo” in the Tandil city. This system will allow a better registration of the merchandise. Thus, the fundamental idea is to achieve through the web application constriction we can control efficiently the access to all information that we need.

The operating system GNU/Linux (Ubuntu), the database server PostgreSQL, the webserver Apache and the dynamic page generator PHP were used to produce the “Bussiness Management System”. The objective is to use platforms of free software and Web Architecture skillfully, operating these tools the system is created, which will improve the entering tasks data one of the entities such as: suppliers, items, trade marks, categories. Besides, it will let to visualize everything that is integrated, either generally or individually, and the issued remits.

The control of each entity is carried out by searching by code or name, allowing a better organization and use of time. Once this job is done, it will let a better flexibility and better information flow. Through its design and data base restrictions will provide a better security in the information for the system users.

CERTIFICACIÓN

Después de haber revisado y corregido la traducción de inglés realizada por el señor Edwin E. Quinatoa Arequipa, certifico que la misma utiliza los principios gramaticales y sintácticos apropiados, a su vez su contenido está en concordancia con el resumen realizado en lengua castellana

A handwritten signature in black ink, enclosed within a large, hand-drawn oval. The signature is cursive and appears to read 'Mayra Alpusig'. Below the signature is a horizontal dotted line.

Lic. Mayra Alpusig

DOCENTE DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI

DEDICATORIA

A Dios por haberme dado el privilegio de terminar mi carrera, dándome paz, fortaleza y llenando de bendiciones todos los días de mi vida.

A mis padres Abelardo y Etelvina por apoyarme incondicionalmente en todas las áreas y etapas de mi recorrido por este mundo, educándome con disciplina y buenos principios para enfrentar en estos días el duro camino de la vida, a mis hermanas Nelly y Jenny por brindarme todo el cariño de siempre.

AGRADECIMIENTO

Al Ing. Hugo Curtí quien fue mi tutor, por brindarme su ayuda y apoyo permanente en todo el transcurso de este trabajo, a las Universidades Técnica de Cotopaxi (Ecuador) y Nacional del Centro de la Provincia de Buenos Aires (Argentina), por haberme permitido venir a la Argentina a realizar el trabajo final de mi carrera.

A mis amigos y amigas, por darme ánimos de continuar en los momentos que más lo necesite y a toda mi familia, personas que me han ayudado directa o indirectamente y hecho posible la realización de este trabajo.

ÍNDICE

INTRODUCCIÓN

1 MARCO TEÓRICO

1.1 SISTEMAS DE GESTIÓN

1.1.1	Introducción.....	1-1
1.1.2	Sistemas de Gestión.....	1-2
1.1.3	Sistemas de Gestión de Base de Datos.....	1-3
1.1.4	Sistemas Gestor de Base de Datos Relacional Orientado a Objetos.....	1-3
1.1.5	Sistema de Gestión de Contenidos.....	1-4
1.1.6	Sistema de Gestión Integral.....	1-5
1.1.7	Sistema de Gestión de Clientes.....	1-6
1.1.8	Sistema de Gestión Comercial.....	1-6
1.1.9	Sistema de Gestión de Proveedores.....	1-7
1.1.10	Sistema de Gestión de Almacén.....	1-8
1.1.11	Sistema de Información.....	1-9

1.2 SOFTWARE LIBRE

1.2.1	Introducción.....	1-9
1.2.2	Historia de Software Libre.....	1-11
1.2.3	Definición de Software Libre.....	1-13
1.2.3.1	Software Libre.....	1-15
1.2.3.2	Software de Código Abierto (Open Source).....	1-16
1.2.3.3	Software de Dominio Público.....	1-16
1.2.3.4	Copyleft.....	1-17

1.2.3.5	Software Protegido con Copyleft.....	1-18
1.2.3.6	Software Libre no Protegido con Copyleft.....	1-19
1.2.3.7	Software Semilibre.....	1-19
1.2.3.8	Software Privativo.....	1-20
1.2.3.9	Software Privado.....	1-20
1.2.3.10	Software Comercial.....	1-20
1.2.4	Qué es la Fundación para el Software Libre.....	1-21
1.2.4.1	Fundación Software Libre América Latina (FSFLA).....	1-22
1.2.4.2	Objetivos de la Fundación de Software Libre de América Latina.....	1-23
1.2.5	Licencias.....	1-24
1.2.5.1	Licencia Pública General de GNU.....	1-24
1.2.5.2	Licencias de Software Libre Compatibles con GLP.....	1-26
1.2.5.3	Licencias de Software Libre Incompatibles con la GLP..	1-27
1.2.5.4	Licencias de Software no Libre.....	1-28

2 PLATAFORMA

2.1 PHP

2.1.1	Introducción.....	2-1
2.1.2	Historia de PHP.....	2-3
2.1.2.1	PHP/FI.....	2-3
2.1.2.2	PHP 3.....	2-4
2.1.2.3	PHP 4.....	2-5
2.1.2.4	PHP 5.....	2-6
2.1.3	Qué es PHP.....	2-6
2.1.4	Qué se Puede Hacer con PHP.....	2-7
2.1.5	Características.....	2-9

2.2 POSTGRESQL

2.2.1	Introducción.....	2-10
2.2.2	Historia y Evolución.....	2-11
2.2.3	Qué es Postgresql.....	2-14
2.2.4	Características.....	2-14
2.2.5	Versión 8.2.....	2-16

2.2.5.1 Características Adicionales.....	2-17
2.2.6 Licencia.....	2-18
2.2.7 Plataformas que Soporta Postgresql.....	2-18
2.3 APACHE	
2.3.1 Introducción.....	2-19
2.3.2 Servidor Apache.....	2-21
2.3.3 Historia.....	2-21
2.3.4 Apache 2.0.....	2-23
2.3.4.1 Cambios hacia Apache 2.0.....	2-23
2.3.4.2 Funcionalidades.....	2-24
2.3.5 Módulos de Multiprocesamiento (MPMS).....	2-27
2.3.6 Características.....	2-28
2.3.7 Licencia.....	2-30
2.4 GNU/Linux	
2.4.1 Introducción.....	2-30
2.4.2 Historia del Proyecto GNU.....	2-31
2.4.3 Linux y el Proyecto GNU.....	2-33
2.4.3.1 El Sistema GNU.....	2-36
2.4.3.2 El Software de GNU.....	2-37
2.4.4 La Historia de Ubuntu.....	2-38
2.4.4.1 Versiones de Ubuntu.....	2-39
2.4.5 Ubuntu.....	2-46
2.4.6 Características.....	2-49

3 IMPLANTACIÓN DEL SISTEMA

3.1 Introducción.....	3-1
3.2 Diagrama Del Modelo Físico de la Base De Datos.....	3-1
3.3 Funciones y Procedimiento.....	3-4
3.4 Aplicación.....	3-9

4 CONCLUSIONES

4.1 Conclusiones.....	4-1
-----------------------	-----

5 TRABAJOS FUTUROS

5.1 Trabajos Futuros.....	5-1
---------------------------	-----

6 BIBLIOGRAFÍA Y REFERENCIAS

6.1 Bibliografía y Referencias.....	6-1
-------------------------------------	-----

GLOSARIO

G.1 GLOSARIO.....	G-1
-------------------	-----

APÉNDICES

A.1 CERTIFICACIÓN PARA LOS SISTEMAS DE GESTIÓN.....	A-1
---	-----

A.2 PGADMIN III.....	A-2
----------------------	-----

ÍNDICE DE FIGURAS

Figura: 1-1. Categorías de Software.....	1-13
---	------

Figura: 1-2. Software Libre definido por la FSF.....	1-22
---	------

Figura: 2-1. Como funciona PHP.....	2-9
--	-----

Figura: 2-2. Evolución de PostgreSQL.....	2-13
--	------

ÍNDICE DE TABLAS

Tabla: 2-1. Base de Datos soportadas por PHP.....	2-8
--	-----

Tabla: 2-2. Versiones de Ubuntu.....	2-40
---	------

INTRODUCCIÓN

Hace algunos años en diferentes Instituciones Educativas, Empresariales, Comerciales, Financieras etc. era común observar que la información generada dentro de ellas se procesaba en forma manual y en archiveros que sobrepasaban su límite, pero con la llegada de los ordenadores, y el avance de nuevas herramientas ha permitido que todo eso cambie potenciando el trabajo corporativo mediante el uso de Sistemas de Gestión. Los mismos han permitido optimizar recursos de toda índole; de esa forma evolucionando en todo el mundo el procesamiento de información.

En los últimos años los sistemas operativos y las nuevas herramientas informáticas están cambiando profundamente la sociedad. Una de las tecnologías mas evolucionadas de hoy en día es el Software Libre que esta disponible públicamente, y que brinda la libertad de ejecutarse, copiarse, distribuirse, estudiarse, cambiarse y mejorarse.

La rápida expansión de Internet y el uso de intranets corporativas ha permitido una transformación en las necesidades de información de las empresas como accesibilidad desde cualquier lugar e incluso desde el exterior, permitiendo un rápido flujo de información, y todo esto ha provocado un movimiento creciente de cambio de las aplicaciones de escritorio a aplicaciones Web que se han constituido en complejos sistemas con interfaces de usuario, dando servicio a procesos de negocio y estableciéndose sobre ellas requisitos estrictos de accesibilidad y respuesta.

Esto ha generado una mejor arquitectura y técnicas de diseño más adecuadas que permiten interactuar con el usuario a través del navegador con aplicaciones

Web: como consecuencia de la actividad del usuario, se envían peticiones al servidor donde se aloja la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con las misma.

Los Sistemas Informáticos son un excelente medio para obtener información de una manera rápida, organizada y concisa, y entre las áreas que más se utilizan estos sistemas es la contabilidad por medio de control de stock, debido a que esta información contable requiere ser procesada de una manera integra y eficiente para que las empresas puedan tener un control oportuno de la mercadería existente que permita un desenvolvimiento rápido y una oportuna toma de decisiones.

En el área del comercio, la Ferretería “*EL SIGLO*”, es uno de los almacenes que presta servicios de venta de artículos para la construcción, carpintería, etc., y para uso general, en la Ciudad de Tandil la cual pertenece a la Provincia de Buenos Aires de la Republica de la Argentina. Esta Ferretería se encuentra ubicada en la Avenida Marconi 1541. Todo su control de entrada, salida y stock de artículos lo llevan manualmente y a través de hojas realizadas en planillas de cálculo, lo cual no le permite a las personas que trabajan en la ferretería llevar un control específico y a la vez les ocasiona pérdida de tiempo al momento de ingresar los artículos. Otro de los inconvenientes al momento de la venta de sus artículos hacia los diferentes clientes que requieren de un servicio rápido y oportuno.

Analizando todos estos inconvenientes y teniendo acceso a nuevas tecnologías informáticas, se ha visto la necesidad de desarrollar un Sistema de Gestión de Comercios basado en Arquitectura Web, el cual permitirá llevar los controles de stock de la Ferretería, por lo que la idea fundamental es lograr que mediante la construcción de una aplicación Web, se pueda tener acceso a la información que se requiera y de esa manera permitir controlar eficazmente el ingreso de cada una de las entidades como son la de los proveedores, artículos y marcas permitiéndolas borrar, actualizar, modificar cada entrada que se realice y

clasificándolas a través de categorías admitiéndola tener una mejor distribución, y el control de stock para tener un registro de los abastecimientos de los artículos tanto para salida como para entrada de sus ventas y asimismo para la creación de remitos que sirven de comprobante de todos los artículos comprados que son entregados a los diversos clientes, y a su vez el sistema permitirá visualizar a todos los proveedores, artículos, marcas, categorías, remitos que se hayan ingresado y creado, ya sea de forma general o individualmente y para un mejor control de cada uno de ellos a través de búsquedas permitiendo tener una mejor organización y disminución de tiempo.

MARCO TEÓRICO

1.1 SISTEMA DE GESTIÓN

1.1.1 INTRODUCCIÓN

[DNV] manifiesta que los Sistemas de Gestión ayudan a que las empresas se enfrenten a los desafíos cambiantes del mercado global del siglo XXI, ya que a través de la implantación y certificación permiten ayudar a lograr su mejora continúa.

Los Sistemas de Gestión de Base de Datos son un conjunto de programas que permiten a los usuarios crear y mantener una base de datos lo cual permite su manipulación para obtener datos específicos y generar informes a partir de los datos que se obtenga; y los Sistema Gestores de Base de Datos Relacionales Orientados a Objetos tiene la capacidad de relacionarse y de soportar aplicaciones para grandes transacciones de datos.

Los Sistemas de Gestión de Contenidos son utilizados para la creación y administración de contenidos específicamente para paginas Web, lo cual permite

a las empresas tener una mayor rapidez en la gestión de paginas nuevas una mayor flexibilidad y capacidad de crecimiento en el creciente sitio Web. Los Sistemas de Gestión Integral son desarrollados para labores de cada una de las áreas de la empresa ya que la principal característica es la mecanización que permite reducir notablemente la carga de trabajo.

Los Sistemas de Gestión de Clientes permiten gestionar los datos de los clientes de negocio, empresa o comercio. El Sistema de Gestión Comercial planifica diversas actividades y posee mecanismos automáticos que permiten diseñar y supervisar eficientemente los planes comerciales. Los Sistemas de Proveedores permiten gestionar la información de los productos de los diversos proveedores con los que se trabaja y se consideran factores como precios, descuentos, etc.

Los Sistemas de Gestión de Almacén tienen como objetivo principal llevar el control de mapa de almacén el cual permite gestionar los stocks y optimizar los recursos físicos y humanos. Los Sistemas de Gestión de Información sirven de apoyo para las actividades y procesos de la empresa a través de implementación y mantenimiento de sistemas informáticos.

1.1.2 SISTEMAS DE GESTIÓN

[DNV] Analiza que las empresas que operan en el siglo XXI independientemente de su tamaño se enfrentan a numerosos desafíos respecto a rentabilidad, competitividad, globalización, calidad, tecnología y desarrollo sostenible, por tal razón un Sistema de Gestión eficiente, diseñado a la medida de los procesos comerciales, puede ayudar a enfrentar los desafíos del cambiante mercado global de hoy.

Una implantación y certificación de un sistema de gestión permite ayudar a las empresas a lograr la mejora continua de su actividad, y permite centrar, organizar sistematizar los procesos para poder comparar los resultados con las mejores prácticas nacionales e internacionales.

El proceso de certificación se adapta a las necesidades particulares de cada compañía, de forma que proporcione la mejor información a la alta Dirección sobre la capacidad de la organización para alcanzar sus objetivos estratégicos.

1.1.3 SISTEMAS DE GESTIÓN DE BASE DE DATOS

[EN97] Analiza que los Sistemas de Gestión de base de datos (“*SGBD o DBMS*”), o “*Database Management System*”, son un conjunto de programas que permiten a los usuarios crear y mantener una base de datos, por lo tanto, el *SGBD* es un sistema de software de propósito general que facilita el proceso de definir, construir y manipular bases de datos para diversas aplicaciones.

Para definir una base de datos hay que especificar los tipos de datos, las estructuras y las restricciones de los datos que se almacenan en ella, y construir un proceso para guardar los datos en algún medio controlado y que permita la manipulación de funciones como consultar la base de datos para obtener datos específicos, actualizar la base de datos para reflejar cambios en el momento y generar informes a partir de los datos.

CARACTERISITCAS DEL SISTEMA DE GESTION DE BASE DE DATOS

- Control de la redundancia
- Restricción de los accesos no autorizados
- Almacenamiento persistente de objetos y estructuras de datos de programas
- Inferencia en la base de datos mediante reglas de deducción
- Suministro de múltiples interfaces de los usuarios
- Representación de vínculos complejos entre los datos
- Cumplimiento de las restricciones de integridad
- Respaldo y recuperación

1.1.4 SISTEMA GESTOR DE BASE DE DATOS RELACIONAL ORIENTADO A OBJETOS

El Sistema Gestor de Base de Datos Relacional Orientado a Objeto (“*ORDBMS*”), o “*Object-Relational Database Management System*”, tiene la capacidad de relacionarse y de soportar aplicaciones para grandes transacciones de datos y puede usar lenguajes de interrogación estándar como el Lenguaje de Consulta Estructurado (“*SQL*”), o “*Structured Query Language*”. Además como sistemas orientados a objetos son capaces de gestionar relaciones muy complejas de los componentes y al mismo tiempo adaptarse más fácilmente a nuevos tipos y formatos de datos. También proporcionan medios para ampliar las posibilidades de una base de datos a funciones y tipos de datos definidos por los usuarios.

1.1.5 SISTEMA DE GESTIÓN DE CONTENIDOS

Un Sistema de Gestión de Contenido (“*CMS*”), o “*Content Management System*”, permite la creación y administración de contenidos principalmente en páginas Web.

Un *CMS* consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público.

Un Sistema de Gestión de Contenidos controla la creación, la gestión, distribución, publicación, y mantenimiento de la información corporativa. Cubre el ciclo de vida de las páginas de un sitio Web, proporcionando herramientas simples para crear el contenido, a través de gestión de archivos y posee la capacidad de manejar la estructura de un determinado sitio.

VENTAJAS DEL USO DE *CMS*:

- Mayor rapidez en la gestión de nuevas páginas y cambios en las mismas.
- Mayor consistencia de contenidos.
- Navegación mejorada del sitio Web.
- Flexibilidad creciente del sitio Web.
- Ayuda a una descentralización del sitio Web.
- Mayor seguridad del sitio Web.
- Reducción de la duplicación de información.
- Mayor capacidad del crecimiento.
- Reducción de los costos de mantenimiento.

1.1.6 SISTEMAS DE GESTIÓN INTEGRAL

Son aplicaciones diseñadas específicamente para desarrollar las labores de gestión de cada una de las áreas de la empresa, e integrarlas entre si, se las denomina Planificación de Recursos Empresariales (“ERP”), o “*Enterprise Resource Planning*” y se presentan generalmente en módulos para ajustarse a las necesidades reales de una empresa.

La principal característica es la mecanización de la mayor parte de las funciones dentro de la cobertura proporcionada, reduciendo notablemente la carga de trabajo y que conllevan además la automatización de la mayor parte de los flujos de información dentro de una empresa.

Estos Sistemas permiten que se integre la información de la empresa y se realice un análisis más exhaustivo de la misma, por lo que al reducir la necesidad de introducir datos manualmente, reduce la carga de trabajo y elimina una posible fuente de errores.

Los sistemas de gestión integral en lo que se refiere a la situación de mercado están en uno de los momentos álgidos de su historia, con crecimientos espectaculares en la facturación de los principales proveedores de sistemas ERP. Pero su costo económico por lo general es bastante alto y su rentabilidad también, y su complejidad es muy elevada, principalmente por el gran número de funciones que incorporan y el volumen de los datos diferentes que gestionan.

1.1.7 SISTEMAS DE GESTIÓN DE CLIENTES

Los Sistemas de Gestión permiten gestionar toda la información acerca de los clientes, su relación con la empresa, los pedidos realizados y el estado actual de los mismos y su interacción con otros sistemas, como puede ser la planificación de la producción, aumentándolo aún más los beneficios obtenidos.

La Ventaja de los Sistemas de Gestión es que permiten un seguimiento completo de los pedidos y la integración de las diferentes fases del proceso reduciendo el trabajo repetitivo y la probabilidad de error Humano. Pero el inconveniente se presenta por la problemática de cada empresa, en especial lo relacionado con la facturación.

Dentro del mercado estos tipos de sistemas son muy extendidos, aunque es previsible una futura adaptación a las nuevas tendencias del mercado, incorporando nuevas tecnologías ya que, el costo depende fundamentalmente del grado de cobertura de la aplicación.

1.1.8 SISTEMAS DE GESTIÓN COMERCIAL

Este tipo de aplicación es muy específica, generalmente no incluida en sistemas de gestiones mayores, que permite planificar las actividades comerciales, realizar seguimientos de las mismas e incluso coordinar las actividades del equipo comercial. Incluyen además mecanismos automáticos de gran utilidad, los cuales permiten diseñar y supervisar eficientemente los planes comerciales, y automatizar un buen número de actividades comerciales, pero su manejo suele ser un tanto complejo.

Dentro del mercado de carácter cada vez más competitivo aparece la necesidad de realizar un gran número de actividades comerciales, que han propiciado un fuerte crecimiento de estos sistemas en los últimos años, y su costo varía

dependiendo de las funciones que incorporen, siendo en general medio-bajo, y los beneficios que, proporcionan rentabilizan en la inversión a corto plazo.

La labor de la informática es agilizar y mejorar las tareas habituales en una oficina, y aplicadas a la mejora de las actividades comerciales, permiten preparar material comercial de forma atractiva, automatizada. Las principales herramientas ofimáticas aplicables en esta área son: Procesador de texto, generación de mailing (correo múltiple), hoja de cálculo, mecanismos de representación gráfica de los datos, bases de datos personales, y herramientas que simplifican enormemente la creación de etiquetas, portadas, carátulas y otros elementos. Estos producen resultados muy atractivos con un esfuerzo mínimo pero requieren de una formación específica.

1.1.9 SISTEMAS DE GESTIÓN DE PROVEEDORES

Permiten gestionar toda la información relacionada con los proveedores que se considere necesaria en los tiempos de respuesta. Paralelamente gestionan información sobre los productos de los diferentes proveedores con los que se trabaje, considerando factores como características, precios, descuentos, etc.

Por otro lado la integración con otros Sistemas como la Gestión de Stocks y la planificación de la producción, permiten adecuar las compras de acuerdo a las necesidades que se requiera. Mediante la utilización adecuada de esta información se puede diseñar una política de compras que facilite optimizar los costos de adquisición y almacenamiento de materias primas, pero el inconveniente es que dependen principalmente del número de proveedores existente para cada tipo de producto de manera simultánea, y las pequeñas empresas tienden a trabajar con un número muy reducido. La situación de mercado no está generalizada salvo en el caso de las grandes empresas ya que su costo depende de las características que tenga.

A través del Intercambio Electrónico de Datos una empresa puede intercambiar documentos con sus proveedores de un modo rápido y eficaz como son: pedidos,

ofertas, presupuestos, confirmaciones, etc. pero necesitan estar integrados con el Sistema de Gestión de compras para resultar efectivos y por otra parte los documentos generados por el sistema de compras son automáticamente enviados al proveedor correspondiente, permitiendo el intercambio fluido de información.

1.1.10 SISTEMAS DE GESTIÓN DE ALMACÉN

El Sistema de Gestión de Almacén se incluye en los sistemas logísticos y el objetivo principal es llevar el control del mapa de almacén para:

- Gestionar los stocks y planificar necesidades
- Evitar que los productos queden obsoletos
- Reducir la dependencia de los operarios del almacén
- Disminuir el inmovilizado
- Optimizar los recursos físicos y humanos.

El Logro de estos objetivos varía en función de la filosofía de trabajo de la empresa, ya que básicamente, desde el punto de vista de la gestión informática se distinguen tres grandes líneas:

- Sistemas de Gestión de Almacén Tradicional: El trabajo de los operarios del almacén se basa en hojas de pedido y la actualización de los stocks no se hace en tiempo real.
- Sistemas de Gestión en Tiempo Real: Los operarios reciben las órdenes de entrada y salida de materiales a través de un Terminal móvil.
- Sistemas de Gestión Automáticos: Los almacenes tienen su propio sistema informático que admite tener el control sin intervención manual.

Se obtiene como beneficio un aumento del control del stock, reduciendo las necesidades de almacenamiento, las roturas de stocks y la dependencia de los operarios, y en lo que respecta al mercado están en la mira de varias empresas por su gran potencial de ahorro y mejora de atención al cliente. Su costo depende del sistema que se elija y para determinar su complejidad se realiza un

estudio previo para evaluar cada caso lo cual permite que haga posteriormente su manejo sencillo.

1.1.11 SISTEMAS DE INFORMACIÓN

Los Sistemas de Información acceden servir de apoyo en las actividades y procesos de la empresa a través de implementación y mantenimiento de los sistemas informáticos.

- **Implantación de sistemas:** Permite que las tareas que se realizan en el proceso de implantación de sistemas informáticos ya sea estudios de necesidad, análisis, desarrollo, generalmente sean destinados a cubrir las necesidades de otras áreas.
- **Mantenimiento:** Permite que las actividades se aseguren del correcto funcionamiento de los sistemas informáticos de una empresa.
- **Atención a usuarios:** Organiza el servicio de atención técnica a los diferentes usuarios de los sistemas informáticos.

La gestión de los sistemas de información de la empresa es posiblemente el área en el que más opciones existen a la hora de implantar sistemas de apoyo, ya que existe un sinnúmero de tecnologías y técnicas disponibles para ello como son: mantenimiento remoto, analizadores de tráfico de red, etc. Las pequeñas empresas se distinguen por disponer de una infraestructura de sistemas de información bastante sencilla, a veces incluso escasa o inexistente.

1.2 SOFTWARE LIBRE

1.2.1 INTRODUCCION

La historia de Software libre manifiesta que en aquel tiempo las labores informáticas, programas, maquinas estaban íntimamente ligados, ya que las personas que ejecutaban programas tenían altos conocimientos de programación,

por lo que el software libre se origina desde inicios de 1969, con la creación de la red ARPANET quien disponía de la información y los archivos fuentes del programa.

Con el pasar de los tiempos Stallman crea el proyecto de Licencia Publica General (“*GNU o GLP*”), o “*General Public License*” y en 1985 se crea la Fundación de Software Libre, y en los 90 Torvalds, crea Linux un **Kernel o núcleo**¹ y en combinación con Linux y *GNU* crean el sistema GNU/Linux y desde entonces varios hechos significativos han evolucionado de software libre que ha sido creado como alternativa del Software propietario.

La definición de Software libre manifiesta que, el “Software Libre” es un asunto de libertad, no de precio, ya que se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software, para lo cual se refieren a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades.
- La libertad de distribuir copias, con lo que se puede ayudar al vecino.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

El código abierto se puede referirse regularmente, a la misma categoría de software libre pero con la diferencia que software libre se refiere a libertad, algo que la expresión de código abierto no lo hace.

Se tiene diversos tipos de software como el Software de Dominio Publico, es el software que no esta protegido por los derechos de autor; **Copyleft**², expresa que cualquiera que redistribuye el software, con o sin cambios, debe dar la libertad

¹Kernel, o núcleo es la parte fundamental de un Sistema Operativo.

²Copyleft, o copia permitida que admite que un programa sea libre, permitiendo compartir programas y sus mejoras para la gente.

de copiarlo y modificarlo; Software protegido con *Copyleft*, es software libre cuyas expresiones de distribución no admiten agregar ninguna restricción adicional cuando redistribuyen o modifican el software; Software libre no protegido con *copyleft*, incluye la autorización del autor para redistribuir y modificar el software y permiso para añadirle restricciones adicionales.

El software semilibre es mejor prácticamente que el software privativo, ya que es un software que no es libre, pero permite para que particulares lo usen, copien, distribuyan y lo modifiquen sin propósito lucrativo; El Software Privativo es un software que no es libre ni semilibre, y por lo que su uso, redistribución o modificación están prohibidos, y requieren que se solicite una autorización; Software Privado, es el software desarrollado para un usuario generalmente en una organización o compañía, y el usuario lo tiene en su poder y lo utiliza, y no lo libera al público ni como código fuente ni como binario; Software Comercial es aquel desarrollado por un negocio que pretende obtener dinero de su utilización.

La Fundación de Software Libre es la principal organización que patrocina el proyecto *GNU*, y la Fundación de Software Libre de América Latina, es una organización no gubernamental sin fines de lucro, el objetivo es defender y promover el uso y desarrollo de software libre y los derechos de los usuarios y programadores en relación al software.

Las Licencias permiten que se publique software libre por lo regular utilizan *GNU* la cual pretende garantizar la libertad de compartir y modificar software libre y a la vez admitan distribuir copias y protege los derechos de los usuarios al poner bajo *copyright*³ y si se desea incorporar partes del programa en otros programas libres cuyas condiciones son diferentes se pide autorización al autor, y se dividen en Licencias de Software Libre Compatibles con *GNU*, Licencias de Software Libre Incompatibles con la *GNU* y Licencias de Software no Libre.

1.2.2 HISTORIA DE SOFTWARE LIBRE

³ Copyright, es el derecho de autor, es un conjunto de normas y principios que regulan los derechos morales que la ley concede a los autores.

[GF05a] Difunde que las labores de la informática, los programas y las máquinas que los ejecutaban estaban íntimamente ligados y no existía el concepto de programa.

Tampoco existían usuarios domésticos, sino que las personas que ejecutaban los programas solían tener muchos conocimientos de programación y por lo general eran científicos e ingenieros por lo cual eran usuarios expertos; y lo normal era intercambiar y mejorar los programas, compartiendo sus modificaciones, que a veces recibían el nombre de (*hacks*). Ellos fueron los primeros (*hackers*) informáticos, lejos de la equiparación habitual de hacker con delincuente, en su concepción originaria se refiere a "una persona que disfruta del desafío intelectual de crear cosas y sobrepasar las limitaciones de algo".

En un principio no eran concebibles las restricciones de las licencias de software actuales, como limitar el número de máquinas en las que se puede correr un programa o el número de copias realizables del mismo ya que el software era libre y los programas se intercambiaban; este espíritu perduró en la comunidad de programadores durante años como algo natural, hasta que las restricciones de los fabricantes y la comercialización de licencias de uso hizo necesario distinguir entre el software que era libre y el que no.

Realizando un recorrido por la historia de Software Libre se da inicio en 1969 con la creación de la red ARPANET, la precursora de Internet. La red ARPANET fue un proyecto público que colocó a la disposición la información y los archivos fuentes de los programas y en 1970 los Laboratorios Bell desarrollan el sistema operativo UNIX, quién poseía los derechos de autor de UNIX, distribuyó los archivos fuentes del sistema operativo.

En 1979 la Universidad de California en Berkeley comienza a desarrollar su propia versión del sistema operativo UNIX *BSD*, lo que propicia el desarrollo de nuevas variantes UNIX comerciales basadas en parte en este sistema y en vista de las tendencias comerciales y por razones puramente morales, Richard Stallman decide en crear el proyecto *GNU*. En 1985 se funda la Free Software Foundation con él a la cabeza; y lo más importante, se establece la Licencia

Publica General, que llega a ser un modelo de licencia para diversos proyectos de software libre.

En los 90, Linus Torvalds, crea Linux, un nuevo *kernel* compatible con UNIX y en 1992 Linux y GNU se combinan para producir el sistema GNU/Linux, quien origino que algunos programas libres hicieran su entrada en las empresas y por otra parte el desarrollo del Internet.

En 1998 fue el año del despegue definitivo de Linux. Desde entonces varios hechos significativos en la evolución del software libre se han presentado y se siguen presentando, es por ello que hablar del futuro no es nada fácil. Muchos manifiestan que GNU/Linux es el futuro, para que Linux se consolide hasta llegar al usuario final estiman que pueden transcurrir entre 5 y 20 años.

El Software Libre ha sido creado como una alternativa al Software Propietario, por lo que hoy en día el Software Libre ha llegado para quedarse y ser el principal competidor de las empresas del software propietario, que están, en estos momentos, en posición dominante.

La Figura 1-1. Muestra las diferentes Categorías de Software.

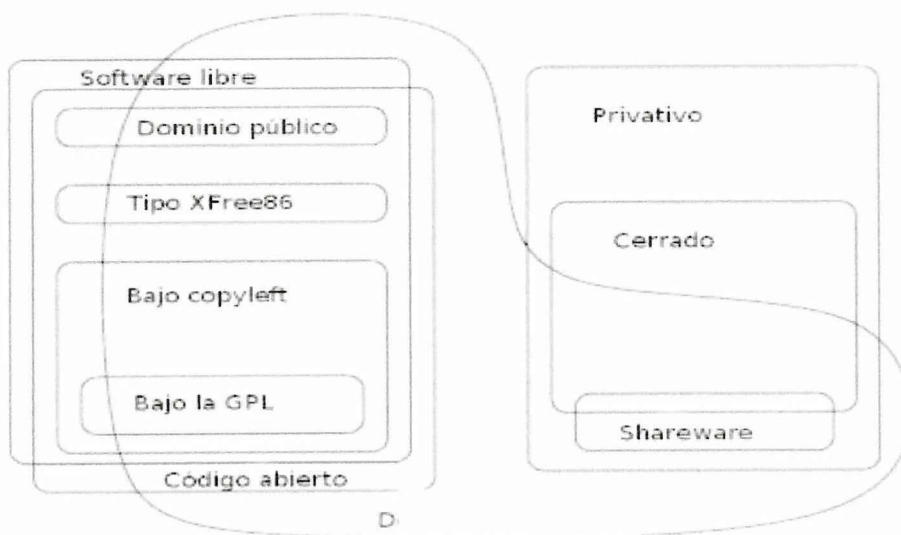


Figura: 1-1. Categorías de Software.

1.2.3 DEFINICION DE SOFTWARE LIBRE

[PM06], manifiesta que, el “Software Libre” es un asunto de libertad, no de precio. Ya que para definir el concepto, se debe pensar en “libre” como en “libertad de expresión”.

Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software, para lo cual se refieren a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que se puede ayudar al vecino (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre y si se tiene todas esas libertades, define [PM06] que se debería tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, el ser libre representa que no se tiene que pedir o pagar permisos.

Y se debería tener la libertad de hacer modificaciones y utilizarlas, y si se publican cambios, no se tiene por qué avisar a nadie en particular, y al utilizarlo un programa por cualquier persona u organización de cualquier tipo de sistema informático, para diversas clases de trabajo, no tener la obligación de comunicárselo al desarrollador.

La libertad de distribuir copias debe incluir tanto las formas binarias o ejecutables del programa como su código fuente, ya sean versiones modificadas o sin modificar.

Para realizar modificaciones y publicar versiones mejoradas, se debe tener acceso al código fuente del programa, por lo cual acceder al código fuente es una condición necesaria para el software libre.

Son aceptables ciertos tipos de reglas sobre la manera de distribuir software libre, mientras no se entre en conflicto con las libertades centrales.

“Software libre” no significa “no comercial”. Un programa libre debe estar disponible para uso, desarrollo y distribución comercial.

El proyecto *GNU*, utiliza *copyleft* para proteger de modo legal estas libertades para todos. Pero el software libre sin *copyleft* existe también ya que se cree que existen razones importantes para lo que es mejor *copyleft*, pero si los programas son software libre sin ser *copyleft*, se lo puede utilizar de todos modos.

A veces las normas de control de exportación del gobierno y las sanciones mercantiles pueden restringir la libertad de distribuir copias de los programas a nivel internacional, y los desarrolladores de software no tienen el poder de eliminar o sobrepasar estas restricciones, pero lo que pueden y deben hacer es rehusar el imponerlas como condiciones de uso del programa.

Cuando se habla de software libre, es mejor evitar términos como: “regalar” o “gratis”, porque esos términos implican que lo importante es el precio, y no la libertad.

Para decidir si una licencia de software concreta es una licencia de software libre, se basan en estos criterios para determinar si tanto su espíritu como su letra en particular los cumplen. Si una licencia incluye restricciones contrarias a la ética de software libre, se rechaza, aun cuando no hubiera previsto el problema en estos criterios.

1.2.3.1 SOFTWARE LIBRE

[RC07] difunde que Software libre es software con autorización para usarlo, copiarlo y distribuirlo por cualquiera que lo desee, ya sea con o sin modificaciones, gratuitamente o mediante una retribución, esto significa que el código fuente debe estar disponible, si no es fuente, no es software.

Si un programa es libre, puede ser potencialmente incluido en un sistema operativo libre tal como *GNU*, pero existen diferentes maneras de hacer libre un programa, numerosas cuestiones a decidir que en muchos casos son matices, y que pueden resolverse de varias formas sin que por ello el programa deje de ser libre.

El calificativo libre del software libre se refiere a (libertad), no a gratuidad, pero sin embargo las campañas de software privativo emplean en ocasiones el término software libre para referirse al precio.

Debido a esta potencial confusión, cuando una compañía de software dice que su producto es software libre, es conveniente verificar siempre los términos de esa distribución para conocer si los usuarios tienen todas las libertades que implica el software libre, ya que a veces es verdadero software libre y en otras no lo son.

Diversos idiomas tienen dos palabras diferentes para (free) libertad y (free) precio cero, ya que el software libre es con frecuencia más confiable que el software no libre.

1.2.3.2 SOFTWARE DE CÓDIGO ABIERTO (OPEN SOURCE)

Numerosas personas utilizan la expresión software de (código abierto) para referirse, más o menos, a la misma categoría a la que pertenece el software libre, pero sin embargo, no son exactamente el mismo tipo de software, ya que código abierto aceptan algunas licencias que software libre considera demasiado restrictivas, y hay licencias de software libre que código abierto no aceptan. Sin embargo, las diferencias entre lo que abarcan ambas categorías son pocas, ya que casi todo el software libre es de código abierto, y casi todo el software de código abierto es libre.

La expresión (*software libre*) se refiere a libertad, algo que la expresión (*código abierto*) no lo hace.

1.2.3.3 SOFTWARE DE DOMINIO PÚBLICO

Es el software que no está protegido por derechos de autor, lo que presenta que algunas copias o versiones modificadas pueden no ser completamente libres.

Existen algunos casos en el que un programa ejecutable puede ser de dominio público sin que su código fuente esté disponible, ya que este software no es software libre, porque para que lo sea es preciso que el código fuente sea accesible. Por otro lado, la mayoría del software libre no es software de dominio público; está protegido por derechos de autor, y los poseedores de estos han dado permiso legal para que cualquiera lo emplee libremente usando una licencia de software libre.

En ocasiones se utiliza el término (*dominio público*) de una manera imprecisa queriendo decir (*libre*) o (*disponible gratuitamente*), pero dominio público, es un término legal y significa de manera precisa (*sin derechos de autor*).

Bajo la convención de Berna, que ha sido firmada por la mayoría de los países, manifiesta que a cualquier escrito nuevo se le aplica automáticamente derechos de autor, lo cual incluye a los programas, es decir que si se quiere que un programa que usted haya escrito sea de dominio público, debe cumplir ciertos procedimientos legales y de esa manera evitar que automáticamente, le sean añadidos los derechos de autor.

1.2.3.4 COPYLEFT

Para hacer que un programa sea libre no es la mejor opción ponerlo en el dominio público, sin derechos reservados, porque por un lado se permite compartir el programa y sus mejoras a la gente, si así se desea; pero también permite a la gente no cooperativa convertir el programa en software privativo, haciéndolo cambios, muchos o pocos, y distribuyendo el resultado como un producto privativo.

El proyecto *GNU*, tiene como objetivo el dar a todo usuario la libertad de redistribuir y cambiar software *GNU*. Si los intermediarios pudieran quitar esa libertad, se tendría muchos usuarios, pero esos usuarios no tendrían libertad. Por lo que en vez de poner software *GNU* en el dominio público, se le protege con “*copyleft*”.

Copyleft expresa que cualquiera que redistribuye el software, con o sin cambios, debe dar la libertad de copiarlo y modificarlo más, por lo que garantiza que cada usuario tiene libertad y igualmente provee de incentivos para que otros programadores aporten al software libre, y ayuda a un programador que quiere contribuir mejoras al software libre obteniendo permiso para hacerlo.

Para cubrir un programa con ‘*copyleft*’, primero reserva los derechos; luego se le añade los términos de distribución, lo cual permite dar derechos de utilizar, modificar, y redistribuir el código del programa, siempre y cuando los términos de distribución no sean cambiados, permitiendo así que el código y las libertades se hagan legalmente inseparables.

1.2.3.5 SOFTWARE PROTEGIDO CON *COPYLEFT*

Es software libre cuyas expresiones de distribución no admiten a los redistribuidores agregar ninguna restricción adicional cuando estos redistribuyen o modifican el software.

En Proyecto *GNU*, protege mediante *copyleft* casi todo el software que se escribe, porque el objetivo es dar a cada usuario las libertades que el término (software libre) implica.

Para proteger un programa con *copyleft*, se necesita usar un conjunto específico de términos de distribución, existen diversas maneras para escribir términos de distribución *copyleft*, por lo que implica que pueden existir muchas licencias de software libre *copyleft*, en lo práctico casi todo el software *copyleft* usa la Licencia Pública General de *GNU*.

1.2.3.6 SOFTWARE LIBRE NO PROTEGIDO CON *COPYLEFT*

Incluye la autorización del autor para redistribuir y modificar el software, así como el permiso para añadirle restricciones adicionales.

El que un programa sea libre pero no está protegido con *copyleft*, implica que algunas copias o versiones modificadas del mismo pueden no ser completamente libres. Una compañía de software podría compilar el programa, con o sin modificaciones, y distribuir el archivo ejecutable como un producto de software privativo.

1.2.3.7 SOFTWARE SEMILIBRE

Es un software que no es libre, pero incluye autorización para que los particulares lo usen, lo copien, lo distribuyan y lo modifiquen (*incluyendo las versiones modificadas*) sin propósitos lucrativos. El software semilibre es mejor prácticamente que el software privativo.

Es imposible incluir software semilibre en un sistema operativo libre, ya que incluye que los términos de distribución para el sistema operativo libre como un todo son la conjunción de los términos de distribución de todos los programas que lo componen. Si se agrega un programa semilibre al sistema hará al sistema (*como un todo*) sólo semilibre.

Existen dos razones para que no suceda esto:

- Se acepta que el software libre deberá ser para todos, incluyendo empresas, no solamente para escuelas o para actividades de ocio particular. .
- La distribución comercial de sistemas operativos libres, incluyendo sistemas GNU/Linux, es muy importante, y los usuarios aprecian el poder disponer de distribuciones comerciales en CD-ROM.

La Fundación para el Software Libre no es una entidad comercial, y por lo tanto podría utilizar internamente un programa semilibre de forma

legal, pero no lo realiza, porque repercutiría negativamente en los esfuerzos realizados para obtener programas que pudieran incluir en el sistema *GNU*.

1.2.3.8 SOFTWARE PRIVATIVO

Es un software que no es libre ni semilibre. Por lo que el uso, redistribución o modificación están prohibidos, y requieren que se solicite una autorización, o está tan restringido que de hecho no puede hacerlo libremente.

La Fundación para el Software Libre sigue la norma de que únicamente se puede instalar un programa privativo en un ordenador, cuando se hace temporalmente y con el propósito de escribir una alternativa libre a ese mismo programa.

1.2.3.9 SOFTWARE PRIVADO

Es el software desarrollado para un usuario generalmente en una organización o compañía. Por lo que el usuario lo tiene en su poder y lo utiliza, y no lo libera al público ni como código fuente ni como binario.

Un programa privado es software libre en un sentido trivial y su único usuario tiene plenos derechos sobre él.

No se lo considera un error que cuando se desarrolla un programa no se lo libere. Existen ocasiones en las que un programa es tan útil que acapararlo para sí constituye un perjuicio para la humanidad, y casi toda la contratación de programadores tiene por objeto el desarrollo de software a medida; por lo tanto, la mayoría de los trabajos de programación son, o podrían ser, desarrollados de un modo compatible con el movimiento de software libre.

1.2.3.10 SOFTWARE COMERCIAL

El software comercial es aquel desarrollado por un negocio que pretende obtener dinero de su utilización, ya que comercial y privativo no son lo mismo; la mayoría del software comercial es privativo, pero hay software libre comercial, y software no libre no comercial.

Ada de *GNU* es un compilador comercial; con la particularidad de que además es software libre.

Dentro del Proyecto *GNU*, el énfasis se hace a otro nivel y lo importante es que Ada de *GNU* es un compilador comercial que posee software libre; el que sea comercial no se refiere a una cuestión crucial ya que el desarrollo adicional de Ada de *GNU* ha hecho de que sea comercial por lo cual es definitivamente, beneficioso.

1.2.4 QUÉ ES LA FUNDACIÓN PARA EL SOFTWARE LIBRE

La Fundación para el Software Libre (“*FSF*”), o “*Free Software Foundation*”, es la principal organización que patrocina el proyecto *GNU* [GA07]. Pero recibe muy pocos fondos de corporaciones o fundaciones, ya que depende de personas como nosotros para poder defender la misión de la *FSF* y así preservar, proteger y promover los derechos de los usuarios; y la libertad de usar, estudiar, copiar, modificar y redistribuir software, también apoya la libertad de expresión, prensa y asociación en Internet, el derecho a usar software criptográfico en comunicaciones privadas, y el derecho a escribir software sin los impedimentos del monopolio.

La Figura 1-2. Se muestra al Software Libre como esta definido por *FSF*.

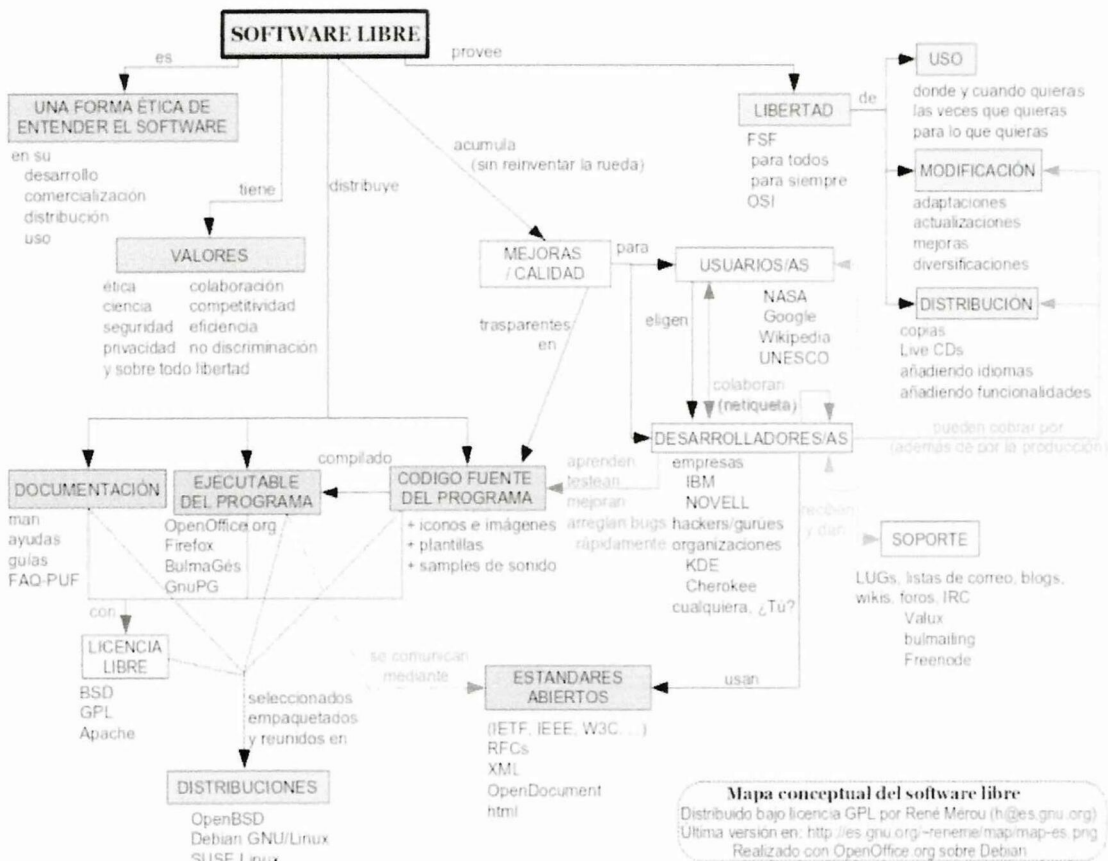


Figura: 1-2. Software Libre definido por la *FSF*.

1.2.4.1 FUNDACIÓN SOFTWARE LIBRE AMÉRICA LATINA (FSFLA)

A [ANT] La *FSFLA* es una organización no gubernamental sin fines de lucro, fundada el 23 de noviembre de 2005, establecida en la ciudad de Buenos Aires, Argentina, el objetivo es promover y defender el uso y desarrollo de Software Libre, y el derecho de las personas a usar, estudiar, copiar, modificar y distribuir software.

FSFLA está formada por un Consejo de Administración, responsable de la toma de decisiones de la organización.

FSFLA es una organización hermana de la Free Software Foundation (*FSF*), la Free Software Foundation Europa (*FSFE*) y la Free Software Foundation India (*FSFI*).

1.2.4.2 OBJETIVOS DE LA FUNDACIÓN DEL SOFTWARE LIBRE DE AMERICA LATINA

[GF05b] Manifiesta que el objetivo principal es promover y defender la libertad y los derechos de los usuarios y programadores en relación al software, específicamente la libertad de desarrollar, usar, y redistribuir y modificar todo el software que usan.

Fomenta la actuación de actores regionales en el contexto global para resguardar el marco jurídico y filosófico del Software Libre, actuando en conjunto y de manera articulada con los demás *FSFs*, para promover y defender el Software Libre, y para acompañar e influir en las políticas que estén relacionadas con él.

A continuación se listan las diferentes funciones:

- Difundir y promover el concepto de Software Libre.
- Participar e influir en los procesos de toma de decisiones políticas que afectan o son afectados por el Software Libre, para que lleguen a conclusiones que sostengan su filosofía y sus principios.
- Defender los derechos de usuarios y desarrolladores de Software Libre, ofreciendo educación y soporte legal en relación al uso, desarrollo, difusión y defensa del Software Libre, especialmente respecto a los programas distribuidos bajo las licencias propias de *FSFs*.
- Auxiliar emprendimientos, públicos o privados, sean empresarios, cooperativos, individuales o colectivos, que busquen consolidar nuevos modelos de negocios o de gestión basados en Software Libre.
- Fomentar la participación activa de los desarrolladores y usuarios de Software Libre de América Latina en el desarrollo, mejora y adaptación de programas libres.
- Dialogar con los gobiernos de la región para alentarlos a adoptar el Software Libre en términos de política pública, y ayudarlos a dar un marco jurídico adecuado a esta política.

- Alentar a las instituciones educativas a usar exclusivamente Software Libre en todas las instancias en las que los alumnos usen computadoras.

1.2.5 LICENCIAS

[PM07] Difunde que el Software que se publique debería ser Software Libre y para lo cual se debe utilizar una licencia de software libre, por lo general utilizan *GNU*, pero eventualmente utilizan otras licencias de software libre, y para el software *GNU* usan otras licencias que son compatibles con la *GNU GPL*.

1.2.5.1 LICENCIA PÚBLICA GENERAL DE *GNU*

Las licencias que cubren la mayor parte del software están diseñadas para privar de libertad de compartir y modificar, la Licencia *GNU* pretende garantizar la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios, la *GNU* se aplica a la mayor parte del software de la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla.

La *GNU* están diseñada para asegurar que tenga la libertad de distribuir copias de software libre, que reciba el código fuente o que pueda conseguirlo si lo quiere, que pueda modificar el software o usar fragmentos de él en nuevos programas libres.

Protegen nuestros derechos en combinación de: Poner el software bajo *copyright* y ofrece esta licencia que nos da permiso legal para copiar, distribuir y/o modificar el software.

Para la protección de cada autor; la Fundación asegura de que todo el mundo comprenda que no se proporciona ninguna garantía para este software libre.

Los términos exactos y las condiciones para la copia, distribución y modificación se exponen a continuación.

- Se aplica a cualquier programa u otro tipo de trabajo que contenga una nota colocada por el poseedor del *copyright*.
- Se puede copiar y distribuir copias literales del código fuente del Programa.
- Puede modificar su copia o copias del Programa.
- Debe hacer que cualquier trabajo que distribuya o publique y que en todo o en parte contenga o sea derivado del Programa.
- Si el programa modificado lee normalmente órdenes interactivamente cuando es ejecutado, debe hacer que, cuando comience su ejecución para ese uso interactivo de la forma más habitual, muestre o escriba un mensaje que incluya un anuncio de *copyright* y un anuncio de que no se ofrece ninguna garantía.
- Puede copiar y distribuir el Programa, como código objeto.
 - Acompañarlo con el código fuente completo correspondiente, en formato electrónico.
 - Acompañarlo con una oferta por escrito, válida durante al menos tres años, de proporcionar a cualquier tercera parte una copia completa, en formato electrónico, del código fuente correspondiente, a un costo no mayor que el de realizar físicamente la distribución del fuente.
 - Acompañarlo con la información que recibió ofreciendo distribuir el código fuente correspondiente.
- No puede copiar, modificar, sublicenciar o distribuir el programa excepto como prevé expresamente esta licencia.
- No está obligado a aceptar esta licencia, ya que no la ha firmado.
- Cada vez que redistribuya el programa, el receptor recibe automáticamente una licencia del licenciataro original para copiar, distribuir o modificar el programa, de forma sujeta a estos términos y condiciones.
- Si como consecuencia de una resolución judicial o de una alegación de infracción de patente o por cualquier otra razón, se

le imponen condiciones que contradigan las condiciones de esta Licencia.

- Si la distribución y/o uso de el programa está restringida en ciertos países, bien por patentes o por interfaces bajo *copyright*, el tenedor del *copyright* que coloca este programa bajo esta licencia puede añadir una limitación explícita de distribución geográfica excluyendo esos países, de forma que la distribución se permita sólo en o entre los países no excluidos de esta manera.
- La Free Software Foundation puede publicar versiones revisadas y/o nuevas de la Licencia Pública General de tiempo en tiempo.

Si se desea incorporar partes del programa en otros programas libres cuyas condiciones de distribución son diferentes, se escribe al autor para pedirle permiso, y además pertenece al grupo de las licencias de software libre compatible con *GLP*.

1.2.5.2 LICENCIAS DE SOFTWARE LIBRE COMPATIBLES CON GLP

- Licencia Pública General Reducida de *GNU*, o *GNU LGPL* (tiene un *copyleft* fuerte, porque permite que el software se enlace con módulos no libres).
- Licencia de Guile
- Licencia de Unidades de Ejecución del Compilador de Ada de *GNU*
- Licencia de X11
- Licencia de Expat
- Licencia de *Copyright* ML Estándar de New Jersey
- Licencia de General De Cryptix.
- Licencia de *BSD* Modificada (por la eliminación de la cláusula de publicidad)
- Licencia de ZLib
- Licencia de la "Biblioteca de Funciones Estándar" de Imatix
- Licencia de de Software Del W3c
- Licencia de Base de Datos de Berkeley
- Licencia de OpenLDAP v.2.7.

- Licencia de Python, v. 1.6 a 2
- Licencia de Python, v. 2.0.1, 2.1.1
- Licencia de *Perl*
- Licencia de Artística con Aclaraciones
- Licencia de Artística 2.0
- Licencia de Pública de Zope v. 2.0
- Licencia para Código Abierto de Intel
- Licencia de *Javascript* de Netscape
- Licencia de eCos v. 2.0
- Licencia de Forum Eiffel v. 2
- Licencia de Vim v.6.1

1.2.5.3 LICENCIAS DE SOFTWARE LIBRE INCOMPATIBLES CON LA GPL

Las siguientes licencias son de software libre pero no son compatibles con la *GNU GPL*:

- Licencia Pública General de Affero
- Licencia Pública Arphic
- Licencia *BSD* Original
- Licencia OpenSSL
- Licencia Libre Académica
- Licencia para Software Abierto v1.0
- Licencia de Apache v. 1.0, 1.1, 2.0
- Licencia de Pública de Zope v. 1
- Licencia de xinetd
- Licencia de Python 1.6b1 y v.2.0, 2.1
- Licencia de OpenLDAP, v. 2.3
- Licencia de Pública de IBM, v. 1.0
- Licencia de Pública Común v. 1.0
- Licencia de Phorum, v. 2.0
- Licencia de Pública del Proyecto LaTeX
- Licencia de Pública de Mozilla
- Licencia para Código Fuente Abierto de Netizen (NOSL), v. 1.0
- Licencia Pública de Interbase v.1.0

- Licencia Pública de Sun
- Licencia para Código Fuente Abierta de Nokia
- Licencia Pública de Netscape
- Licencia para Código Fuente Abierto de Jabber, v.1.0
- Licencia para Código Fuente Según los Estándares Industriales de Sun, v. 1.0
- Licencia Pública Q (QPL), v. 1.0
- Licencia FreeType
- Licencia de *PHP*, v. 3.0.
- Licencia Zend, v. 2.0.
- Licencia Plan 9 (Junio de 2003)
- Licencia Fuente Pública de Apple v. 2

1.2.5.4 LICENCIAS DE SOFTWARE NO LIBRE

Este tipo de licencias son automáticamente incompatibles con la *GNU GPL*.

- Licencia Artística (original)
- Licencia Fuente Pública de Apple v. 1.X
- Licencia B de Software Libre de SGI Versión 1.1
- Licencia Sun Community Source
- Licencia Plan 9
- Licencia Pública Abierta
- Licencia Pública de la Universidad de UTAH
- Licencia Pública eCos, v. 1.1
- Licencia para Código Fuente de Sun Solaris v.1.1
- Licencia YaST
- Licencia Daniel Bernstein's
- Licencia Pública libre de Aladdin
- Licencia Scilab
- Licencia Pública de AT&T
- Licencia Jahia Community Source
- Licencia ksh93
- Licencia de Código Fuente compartido de Microsoft

- Licencia Hacktivism Enhanced-Source Software License Agreement
- Licencia Squeak

PLATAFORMA

2.1 PHP

2.1.1 INTRODUCCION

Los autores [ABDLOR06], manifiestan que, el Preprocesador de Hipertexto (“*PHP*”), o “*Hypertext Preprocessor*”, en estos últimos años ha crecido extraordinariamente lo cual ha convertido en el lenguaje de entornos Web mas importe de la actualidad. Fue creado por Rasmus Lerdorf en 1995, como un simple *script*⁴ de Lenguaje Práctico de Extracción y de Informes (“*PERL*⁵”), o “*Practical Extraction and Report Language*”, creó una implementación *C*⁶ mayor para que pueda comunicarse con base de datos, realizar pequeñas aplicaciones Web, y luego liberó el código fuente *PHP/FI*.

En 1997, *PHP/FI* 2.0, tuvo un mayor seguimiento, pero no se liberó oficialmente hasta noviembre de 1997. *PHP* 3.0 fue creada por Andi Gutmans y Zeev Zuraski y su mejor característica era su gran extensibilidad, y el soporte de sintaxis orientado a objetos. En 1998 se comienza a trabajar en la reescritura del *núcleo* de *PHP* y mejorar la ejecución de aplicaciones complejas y el nuevo Motor Zend, alcanzando estos objetivos de diseño satisfactoriamente a mediados de 1999. *PHP* 4.0 acoplado a nuevas características fue liberado en mayo del 2000, como soporte para los

⁴ Scripts, pequeños programas incrustados en las páginas que permite definir interactividades de cualquier tipo.

⁵ Perl, lenguaje de programación de alto nivel, código abierto, utilizado para construir aplicaciones *CGI*, permitiendo extraer información de archivos de texto y genera informes.

⁶ C, es un lenguaje de programación de nivel medio, orientado a la implantación de Sistemas Operativos, transportable y estructurado.

Servidores Web Protocolo de Transferencia de Hipertexto (“*HTTP*”), o, HyperText Transport Protocol. En *PHP 5* se incluye el nuevo motor Zend 2.0 integrado para mejorar el rendimiento.

PHP es un lenguaje de código abierto, interpretado y de alto nivel, efectuado en lado del servidor utilizado para la generación de páginas Web dinámicas. La finalidad es permitir a los desarrolladores la generación dinámica de páginas de una manera rápida y fácil.

Existen tres campos donde se utilizan los *scripts* en *PHP*: *scripts* del lado del servidor, *scripts* en la línea de comandos y para escribir aplicaciones de interfaz gráfica. Además permite utilizar diversos Sistemas Operativos como Linux, Unix, Microsoft Windows, etc. Soporta diversos servidores Web como Apache, Microsoft Internet Information Server, Personal Web Server, etc. Ostenta módulos para la mayoría de los servidores de estándar de Interfaz de Salida Común (“*CGI*”), o “*Common Gateway Interface*”

Además, *PHP* soporta conectividad con diversas Bases de Datos como Adabas D, Informix, *Ingres*⁷, MySQL, Conexión Abierta de Base de Datos (“*ODBC*”), o “*Open Data Base Connectivity*”, Oracle (OCI7 y OCI8), PostgreSQL, etc. También puede comunicarse con otros servicios utilizando protocolos como: Protocolo de Acceso a Directorios Ligeros (“*LDAP*”), o “*Lightweight Directory Access Protocol*”, Protocolo de acceso a los mensajes de Internet (“*IMAP*”), o “*Internet Message Access Protocol*”, Protocolo Simple de Gestión de Red (“*SNMP*”), o “*Simple Network Management Protocol*”, Protocolo de Transferencia de Noticias (“*NNTP*”), o “*Network News Transport Protocol*”, etc. Así mismo permite crear *sockets*⁸ puros.

Entre las características mas sobresalientes de *PHP*, se puede mencionar que soporta una gran cantidad de base de datos, es multiplataforma y permite usar programación orientada a objetos, También facilita soluciones simples para

⁷ Ingres, fue el primer proyecto desarrollado con la finalidad de crear el sistema de base de datos PostgreSQL.

⁸ Socket, es un método para la comunicación entre un programa del cliente y un programa del servidor de la red.

paginaciones dinámicas, adquiere mejoras constantes y extensiones del lenguaje, admite crear un *script CGI*, manipular de *cookies*⁹ y paginas dinámicas.

2.1.2 HISTORIA DE *PHP*

De acuerdo a los autores [ABDLOR06], *PHP* ha recorrido un largo camino en los últimos años lo cual le ha permitido crecer hasta ser uno de los más importantes lenguajes de programación en entornos Web.

2.1.2.1 *PHP/FI*

PHP es el heredero de un producto anterior, en principio llamado *PHP/FI*. *PHP/FI* fue creado por Rasmus Lerdorf en 1995, como un simple conjunto de *scripts* de *Perl* para controlar los accesos a su trabajo online. Llamó a ese conjunto de *scripts* “*Personal Home Page Tools*”. Según se requería más funcionalidad, Rasmus fue escribiendo una implementación *C* mucho mayor, que era capaz de comunicarse con bases de datos, y permitía a los usuarios desarrollar sencillas aplicaciones Web dinámicas y a la vez libero el código fuente de *PHP/FI* para que cualquier persona pueda utilizarlo y pueda arreglar errores y mejorar el código.

PHP/FI, que se mantuvo para páginas personales y como intérprete de *formularios*¹⁰, incluía algunas de las funcionalidades básicas de *PHP* tal y como lo conoce hoy. Tenía variables como las de *Perl*, interpretación automática de variables de *formulario* y sintaxis embebida en el Lenguaje de Marcación de Hipertexto (“*HTML*”), u “*Hiper – Text Markup Lenguaje*”. La sintaxis por sí misma era similar a la de *Perl*, aunque mucho más limitada, simple y algo inconsistente.

Por 1997, *PHP/FI* 2.0, la segunda escritura de la implementación en *C*, tuvo un seguimiento estimado de varios miles de usuarios en todo el mundo, con

⁹ Cookies, es un mecanismo que sirve para almacenar datos en el navegador del usuario remoto, para así poder identificar al usuario cuando vuelva.

¹⁰ Formulario, conjunto de campos (espacios) para introducir datos en una página Web.

aproximadamente 50.000 dominios informando que lo tenían instalado, sumando alrededor del 1% de los dominios de Internet.

PHP/FI 2.0 no se liberó oficialmente hasta noviembre de 1997, después de pasar la mayoría de su vida en desarrollos beta. Fue sucedido en breve tiempo por las primeras versiones alfa de *PHP 3.0*.

2.1.2.2 *PHP 3*

Fue la primera versión que se parecía al *PHP* tal y como se lo conoce, y creado por Andi Gutmans y Zeev Zuraski en 1997 reescribiéndolo completamente, después de que encontraran que *PHP/FI 2.0* tenía pocas posibilidades para desarrollar una aplicación comercial que estaban desarrollando para un proyecto universitario. En un esfuerzo para cooperar y empezar a construir sobre la base de usuarios de *PHP/FI* existente, Andi, Rasmus y Zeev decidieron cooperar y anunciar *PHP 3.0* como el sucesor oficial de *PHP/FI 2.0*, interrumpiéndose en su mayor parte el desarrollo de *PHP/FI 2.0*.

Una de las mejores características de *PHP 3.0* era su gran extensibilidad. Además de proveer a los usuarios finales de una sólida infraestructura para muchísimas bases de datos, protocolos de Interfaz de Programación de Aplicaciones (“*APIs*”), o, “*Application Program Interface*”, las características de extensibilidad de *PHP 3.0* atrajeron a docenas de desarrolladores a unirse y enviar nuevos módulos de extensión, la cual fue la clave del enorme éxito de *PHP 3.0*.

Otras características clave introducidas en *PHP 3.0* fueron el soporte de sintaxis orientado a objetos y un lenguaje mucho más potente e consistente.

Todo el nuevo lenguaje fue liberado bajo un nuevo nombre, que borraba la implicación de uso personal limitado que tenía el nombre *PHP/FI 2.0*. Se llamó “*PHP*” a secas, con el significado de ser un acrónimo recursivo - *PHP*: Hypertext Preprocessor.

A finales de 1998, *PHP* creció hasta una base de instalación de decenas de millones de usuarios y cientos de miles de sitios Web informando de su instalación. En su apogeo, *PHP* 3.0 estaba instalado en aproximadamente un 10% de los servidores Web en Internet, y se liberó oficialmente en Junio de 1998, después de estar unos 9 meses en pruebas públicas.

2.1.2.3 *PHP* 4

En el invierno de 1998, poco después del lanzamiento oficial de *PHP* 3.0, Andi Gutmans y Zeev Suraski comenzaron a trabajar en la reescritura del *núcleo* de *PHP*. Los objetivos de diseño fueron mejorar la ejecución de aplicaciones complejas, y mejorar la modularidad del código base de *PHP*. Estas aplicaciones se hicieron posibles por las nuevas características de *PHP* 3.0 y el apoyo de una gran variedad de bases de datos y *APIs* de terceros, pero *PHP* 3.0 no fue diseñado para el mantenimiento eficiente de aplicaciones complejas.

El nuevo motor, apodado 'Motor Zend' (comprimido de sus apellidos, Zeev y Andi), alcanzó estos objetivos de diseño satisfactoriamente, y se introdujo por primera vez a mediados de 1999. *PHP* 4.0, basado en este motor, y acoplado con un gran rango de nuevas características adicionales, fue oficialmente liberado en mayo del 2000, casi dos años después que su predecesor, *PHP* 3.0. Además de la mejora de ejecución de esta versión. *PHP* 4.0 incluía otras características clave como el soporte para la mayoría de los servidores Web, sesiones de *HTTP*, *buffers*¹¹ de salida, formas más seguras de controlar las entradas de usuario y muchas nuevas construcciones de lenguaje.

¹¹ Buffer, es una ubicación de la memoria en un computador o es un instrumento digital reservado para el almacenamiento temporal de información digital, mientras que esta esperando ser procesada.

2.1.2.4 PHP 5

El futuro de *PHP* está dirigido por su *núcleo*, el motor Zend. *PHP 5* incluye el nuevo motor Zend 2.0 integrado y ha mejorado significativamente el rendimiento y las capacidades de *PHP*, al mismo tiempo que se han tomado precauciones para evitar incompatibilidades con el código existente.

Posee algunas nuevas funciones para matrices, el motor de Base de Datos, InterBase, *iconv* (modulo que permite convertir una cadena representada por un juego de caracteres local a una representada por otro juego de caracteres), secuencias (permite leer datos desde la secuencia o escribir datos hacia ella en forma lineal), fecha y hora, cadenas. Posee nuevas directivas de *php.ini* como son *register_long_arrays* (habilita/deshabilita el registro de las largas variables obsoletas *\$HTTP_*_VARS*), *session.hash_function* (selecciona una función de resumen criptográfico (MD5 o SHA-1)), etc.

En lo referente a la base de datos en esta versión ya viene incorporado la extensión SQLite que es un motor de base de datos embebible y no es una biblioteca de cliente usada para conectarse con un servidor grande de base de datos. Tiene un nuevo modelo de datos que permite mejorar el rendimiento, y posee un reporte de errores como un mensaje STRICT que puede advertir el uso de funciones obsoletas.

2.1.3 QUÉ ES PHP

De acuerdo a [HPGPTB04, HC01], *PHP* es un lenguaje de Código Abierto, interpretado de alto nivel, para enriquecer páginas *HTML*.

Según [HR07], *PHP* es un lenguaje de *script* interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar a Páginas de Servidor Activas (“ASP”), o “Active Server Pages”, de Microsoft o las Páginas de Servidor *Java* (“JSP”), o “Java Server Pages”, de Sun, embebido en páginas *HTML*.

La mayor parte de su sintaxis ha sido tomada de *C*, *Java*¹² y *PERL*, con algunas características específicas de sí mismo. La meta del lenguaje es admitir a los desarrolladores la generación dinámica de páginas de una manera rápida y fácil.

No es un lenguaje de marcas como podría ser *HTML*, Lenguaje de Marcación Extensible (“*XML*”), o “*Extensible Markup Lenguaje*”, o Lenguaje de Marcación Inalámbrico (“*WML*”), o “*Wireless Markup Lenguaje*”.

2.1.4 QUÉ SE PUEDE HACER CON *PHP*

De acuerdo a los Autores [HPGPTB04], con *PHP* se puede ejecutar todo tipo de tareas mediante un *script*. Utilizando un *CGI*, que define los procedimientos utilizables para acceder desde una página Web a un programa que reside y se ejecuta en el servidor Web (a través de un *formulario*).

Existen tres campos en las que se utilizan los *scripts* escritos en *PHP*:

- ***Scripts del lado del servidor***: Es el principal campo de trabajo. Para que esto funcione se necesita, el intérprete *PHP* (*CGI* ó módulo), un servidor Web y un navegador. Por tal razón se hace necesario correr el servidor Web con *PHP* instalado.
- ***Scripts en la línea de comandos***: Se puede crear un *script PHP* y ejecutarlo sin ningún servidor Web, por lo cual es necesario el intérprete *PHP*. Esto es ideal para utilizarse en *scripts* ejecutados regularmente desde *CRON*¹³ (en Unix o Linux) o el Planificador de tareas (en Windows). También es ideal para tareas simples de procesamiento de texto.

¹² Java, es un lenguaje de programación de propósito general, plenamente orientado a objetos, independiente de la plataforma y permite el funcionamiento en cualquier ordenador del mercado.

¹³ Cron, es un administrador regular de procesos en segundo plano que ejecuta programas a intervalos regulares.

- **Escribir aplicaciones de interfaz gráfica:** Se puede utilizar *PHP-GTK*¹⁴ para escribir aplicaciones graficas independientes de la plataforma. *PHP-GTK* es una extensión de *PHP*.

Además permite ser utilizado en diversos sistemas operativos del mercado, como: Linux, diversas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS. *PHP* soporta la mayoría de servidores Web de los cuales se puede citar: Apache, Microsoft Internet Information Server, Personal Web Server, Netscape, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd y otros. *PHP* ostenta módulos para la mayoría de los servidores que soporten el estándar *CGI*.

En la tabla 2-1. Las Bases de Datos que están soportadas por *PHP* se muestran.

Adabas D	<i>Ingres</i>	Oracle (OCI7 and OCI8)
dBase	Internase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Veloces
Informix	ODBC	Unix dbm

Tabla: 2-1: Bases de Datos soportadas por *PHP*

También se cuenta con una extensión de Abstracción de Base de Datos (“*DBX*”), u “*Database Abstraction*”, que permite usar de forma transparente cualquier base de datos soportada por esta extensión. Adicionalmente, *PHP* soporta el Estándar de *ODBC*, permitiendo conectar cualquier base de datos que soporte dicho estándar.

PHP posee un soporte para comunicarse con otros servicios utilizando protocolos como: *LDAP*, *IMAP*, *SNMP*, *NNTP*, Protocolo de Oficina de correos versión 3 (“*POP3*”), o “*Post Office Protocol Versión 3*”, Protocolo *HTTP*, dominios de Internet COM (en Windows) y muchos otros. También permite crear *sockets* puros. *PHP* soporta Intercambio de datos distribuido Web (“*WDDX*”), o (“*Web Distributed Data Exchange*”), el cual permite compartir información estructurada entre distintos lenguajes de programación.

¹⁴ *GTK*, es un grupo importante de bibliotecas que sirve para desarrollar interfaces gráficas de usuario principalmente las interfaces gráficas.

De Acuerdo a su criterio, Hinostraza [HR07], manifiesta que a diferencia de *Java* o *JavaScript*¹⁵ que se ejecutan en el navegador, el código *PHP* se ejecuta en el servidor permitiendo acceder a los recursos que tenga el servidor como puede ser una base de datos. El programa *PHP* es ejecutado en el servidor y el resultado es enviado al navegador, obteniendo como resultado una página *HTML* o una pagina *WML* (Wap).

La Figura 2-1. Muestra un diagrama con el funcionamiento de *PHP*.



Figura: 2-1. Como funciona *PHP*

2.1.5 CARACTERÍSTICAS

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas Web dinámicas:

- De acuerdo a [HPGPTB04, HR07], existe soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Según Carvajal [CR04], un código abierto, multiplataforma que contiene funciones para trabajar con todas las tecnologías Web existentes hoy en día.

¹⁵ JavaScript, es un lenguaje de código abierto, interpretado, orientado a las páginas Web, facilitando interactuar con el navegador de forma dinámica y eficaz.

- Según los Autores [HPGPTB04], posee la facilidad de usar programación procedimental o programación orientada a objetos.
- De acuerdo a Hinostraza [HR07], es la unión de varias bibliotecas externas, admitiendo crear Formatos de Documento Portable (“PDF”), o “*Portable Document Format*”, e incluso analizar código *XML*.
- Proporciona una solución simple y universal para las paginaciones dinámicas de la Web de fácil programación.
- Adquiere mejoras constantemente y extensiones del lenguaje para aumentar las capacidades de *PHP*.
- *PHP* permite realizar cualquier acción mediante un *script CGI*, como el procesamiento de información en *formularios*, manipulación de *cookies* y páginas dinámicas.

2.2 POSTGRESQL

2.2.1 INTRODUCCIÓN

PostgreSQL, es un servidor de Sistema de Gestión de Bases de Datos Objeto-Relacionales o *ORDBMS*, que fue desarrollado entre 1977 y 1985 como el proyecto *Ingres* en la Universidad de Berkeley en California, y en 1986 de acuerdo a [LT96], un grupo liderado por Michael Stonebraker decide crear un sistema de base de datos objeto relacional orientada a objetos llamado Postgres, y en 1994 se le agrega a Postgres un interprete de *SQL*, y nació Postgresql95, como un proyecto open-source de dominio publico.

En 1996 se crea PostgreSQL que permite reflejar la relación entre el Postgres y el interprete de *SQL* el cual permite actualmente un activo proceso de desarrollo mundial y posee características de sistemas de base de datos comerciales y es realizado por un grupo de desarrolladores dispersos alrededor del mundo y esta considerado como la base de datos de código abierto mas avanzada del mundo.

Posee un sin número de características como estructura de datos array, índices, herencias entre tablas, un modelo objeto-relacional, transacciones, disparadores (triggers), vistas, reescritura de consultas que lo hace altamente extensible, soporta varias especificaciones de *SQL*, Integridad referencial, Control de Concurrencia Multi-Versión (“*MVCC*”), o “*Multi-Version Concurrency Control*”, arquitectura Cliente/Servidor.

La versión 8.2 posee herramientas que mejora el rendimiento, Base de Datos Warm Standby, Construcción en línea de índices, Características de *SQL* 2003, y control de base de datos a nivel de aplicación, criptografía, mejoras en *SQL*, extracción de registros por lotes y PL/Python.

Posee una Licencia de Distribución de Software Berkeley (“*BSD*”), o “*Berkeley Software Distribution*”, que requiere que el código fuente este licenciado y mantenga su información de derechos, el cual permite a PostgreSQL que sea muy popular para personas que deseen incluir esta base de datos.

2.2.2 HISTORIA Y EVOLUCIÓN

De acuerdo [POST], PostgreSQL, es un servidor de *ORDBMS*.

De acuerdo a [POST, RD05], PostgreSQL ha sido desarrollado de varias formas entre 1977 y 1985. Empezando como un proyecto denominado *Ingres* en la Universidad Berkeley de California.

En 1986 otro equipo liderado por Michael Stonebraker en Berkeley, continuó el desarrollo del código de *Ingres*, pero debido al sin número de dificultades encontradas en el código, y con la finalidad de incluir nuevas características, el grupo de desarrolladores decide crear un sistema de bases de datos objeto-relacionales llamado Postgres.

Con la Implementación de Postgres se buscó principalmente:

- Proporcionar un mayor soporte a objetos complejos.
- Proveer facilidades a los usuarios para extender los tipos de datos.
- Incorporar características de base de datos activas (ejemplo: alertas y triggers).

En 1994 le agrego a Postgres un interprete de *SQL* y nació Postgresql95, que fue desarrollado subsecuentemente en la Web como un proyecto open-source de dominio publico y se escribió completamente compatible con Instituto Nacional Estadounidense de Estándares (“ANSI”), o, (*American National Standards Institute*), *C* y su tamaño disminuyo en un 25% el cual mejoró su performance y su mantenibilidad.

Las mejoras más importantes son:

- El reemplazo del lenguaje *Postquel* (*Lenguaje para Consultas*) por el *SQL*.
- La ejecución de aggregates y la cláusula GROUP BY.
- Perduró la interfase para *C* y se agregó la interfase para clientes con el Lenguaje de Comando Herramientas (“*TCL*”), u “*Tool Command Language*”.

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 periplo como Postgres95, que como nombre no se adecuaba al paso del tiempo. PostgreSQL permite reflejar la relación entre el Postgres y el intérprete de *SQL*.

Esto fue realizado por un grupo de programadores dedicados al open-source consolidados en los alrededores de Hamburgo en Alemania, quienes se dedicaron a corregir errores que producían la caída del servidor, y a medida que aumentaba el código, se corrigieron errores mas complicados y se agregaron nuevas características.

- El bloqueo a nivel de tabla fue reemplazado por un control de concurrencia multi-versión.
- Se agrego soporte para sub-queries.

- Se incluyen características compatibles al lenguaje *SQL2*.
- Se renovaron los tipos de datos nativos.
- Se incrementa la velocidad en un 20-40%.

El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial, está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Ofreciendo control de concurrencia multi-versión y soportando casi toda la sintaxis *SQL*.

En la Figura 2-2. La Evolución de PostgreSQL se muestra.

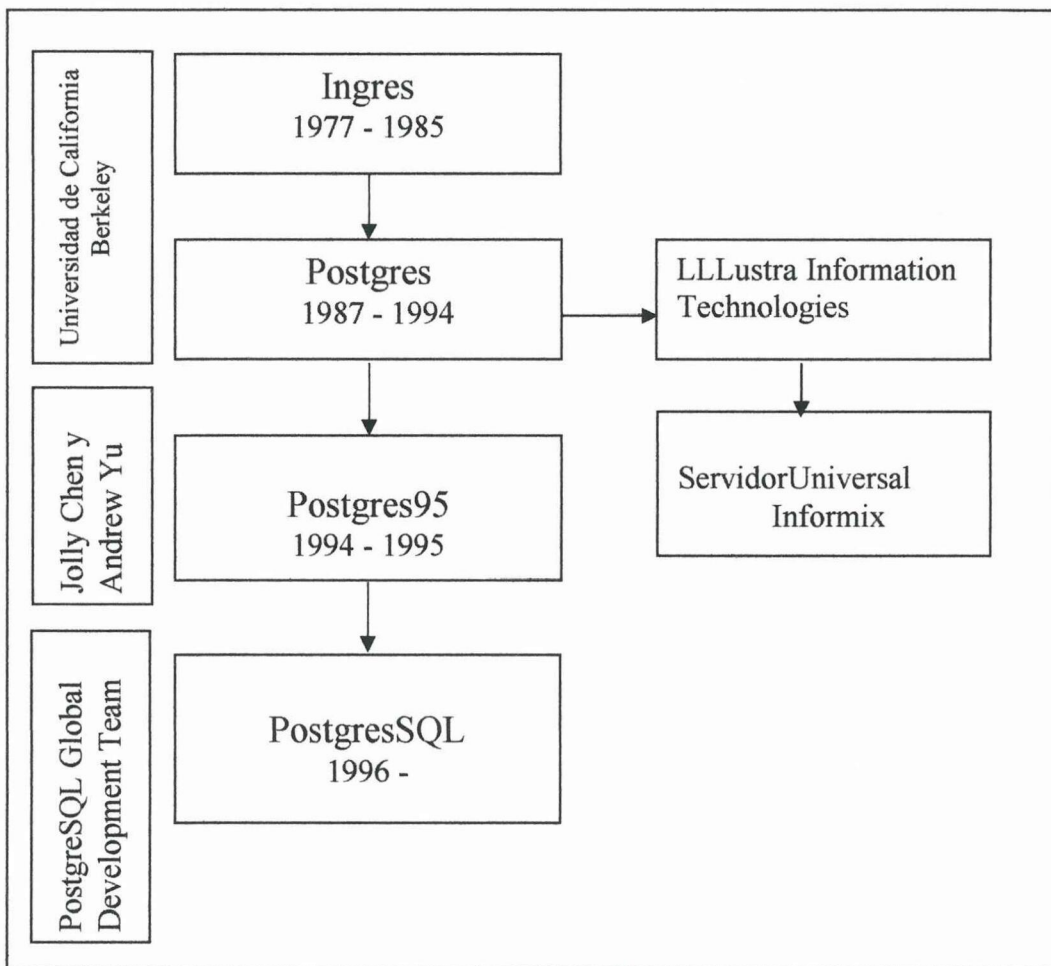


Figura: 2-2. Evolución de PostgreSQL

2.2.3 QUÉ ES POSTGRESQL

Según [TE06], PostgreSQL es un sistema de base de datos objeto -relacional que tiene las características de sistemas de bases de datos comerciales tradicionales con mejoras que serán encontradas en Sistemas de Gestión de Bases de Datos (“DBMS”), o, “*DataBase Management System*”, de la próxima generación.

Se acuerdo [LT96], PostGreSQL es una derivación libre (Open-Source) y utiliza el lenguaje *SQL92/SQL99*; incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostGreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Además [HA04], analiza que:

- Es un Sistema de almacenamiento y manipulación de datos.
- Posee interoperabilidad con otros *DBMS: SQL*.
- Provee almacenamiento confiable, consistente, robusto.
- Se caracteriza por una manipulación potente, flexible, eficiente.

De Acuerdo a [TE06], El desarrollo de PostgreSQL es realizado por un equipo de desarrolladores (voluntarios en su mayoría) dispersos alrededor del mundo y comunicados vía Internet. Este es un proyecto de la comunidad y no es controlado por ninguna compañía.

2.2.4 CARACTERISTICAS

De acuerdo a [LC07] define que, las principales características de PostgreSQL son las siguientes:

- Implementación del estándar *SQL92/SQL99*.

- Soporte de distintos tipos de datos: además del soporte para los tipos base, también se soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios y también los permisos asignados a cada uno de ellos.

Según [BH01], las características, a partir de PostgreSQL 7.1.x. son las siguientes:

- **DBMS Objeto-Relacional.-** 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.-** Soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.
- **Soporte_SQL_Comprensivo.-** Sobrelleva la especificación *SQL99* e incluye características avanzadas tales como las uniones (joins) *SQL92*.
- **Integridad Referencial.-** Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- **API Flexible.-** La flexibilidad de *API* de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el Sistema Administrador de Bases de Datos Relacionales (“*RDBMS*”), o “*Relational Data Base Manager System*”, PostgreSQL. Estas interfaces

incluyen Object Pascal, Python, *Perl*, *PHP*, *ODBC*, *Java/JDBC*, Ruby, *TCL*, *C/C++*, y Pike.

- **Lenguajes Procedurales.-** Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL.
- **MVCC.-** Es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios.
- **Cliente/Servidor.-** Usa una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

2.2.5 VERSION 8.2

De Acuerdo a [MB06], esta versión modifica las órdenes y facilidades del sistema, además agrega sintaxis extendida y compatible, permitiendo de una manera más fácil el uso de todas las características avanzadas de PostgreSQL.

Las herramientas y características que facilitan la gestión de bases de datos y el desarrollo incluyen:

Mejoras de Rendimiento: Permite mejorar el rendimiento alrededor de un 20% en pruebas de sistemas de Procesamiento en Línea de Transacciones (“*OLTP*”), o, “*OnLine Transaction Processing*”, de gran escala, y los cambios incluyen ordenamientos en memoria y en disco más rápido, mejor escalabilidad en sistemas multi-procesador, mejor optimización de consultas sobre datos particionados, cargas masivas más rápidas y *outer joins* considerablemente acelerados.

Bases de Datos Warm Standby: Una extensión a la característica de recuperación a un punto en el tiempo, introducida en la versión 8.0, permite a los administradores crear fácilmente una copia para recuperación inmediata de su cluster de bases de datos.

Construcción en Línea de Índices: Puede ocurrir mientras las aplicaciones escriben a las tablas de la base de datos, permitiendo el afinamiento de rendimiento sin afectar la disponibilidad.

Características de SQL 2003: PostgreSQL, bien conocido por su conformidad a los estándares, ha agregado sintaxis para varias características introducidas en la especificación del *ANSI SQL: 2003*, incluyendo funciones de agregación estadística, sentencias `VALUE` con múltiples registros, `UPDATE RETURNING` y funciones de agregación de múltiples columnas.

Las características avanzadas de PostgreSQL 8.2 antes que ningún otro sistema de gestor de bases de datos incluyen:

Índices Invertidos Generalizados: Una forma más escalable y programable de indexar datos semi-estructurados y texto.

DTrace: PostgreSQL ha sido instrumentado para permitir la trazabilidad a través de un marco genérico de monitoreo, usando DTrace en Solaris.

2.2.5.1 Características Adicionales:

- Bloqueos consultivos (advisory locks): permiten el control de objetos de bases de datos a nivel de aplicación usando el eficiente motor de bloqueos de PostgreSQL.
- `pg_dump` selectivo: permite extraer vuelcos transaccionalmente consistentes de relaciones, seleccionando la inclusión y exclusión usando expresiones regulares.
- Sentencias preparadas: tiene nuevas interfaces administrativas y mejoras de rendimiento en sentencias preparadas.
- ISN/ISBN: este módulo contiene tipos de datos para soportar descriptores de producto ISN para inventarios y sistemas de venta al por menor. Ha sido aumentado para soportar los últimos estándares internacionales.
- Criptografía: el módulo `pgcrypto`, soporta criptografía dentro de la base de datos y fue actualizado con los últimos algoritmos.

- Mejoras al *SQL*: nueva sintaxis, incluyendo UPDATE RETURNING, DROP IF EXISTS, ON COMMIT y nuevos comandos de propiedad (*ownership*) y permisos, para hacer más fácil el manejo de objetos de la base de datos en la línea de órdenes.
- Extracción de registros por lotes en psql: permite devolver filas a la consola en lotes en lugar de todas a la vez.
- Reconstrucción de ECPG: la interfaz embebida *C* de PostgreSQL ha sido extensivamente reescrita y se agregaron nuevas interfaces.
- Mejoras importantes en TSearch2: soporte de UTF-8, tesauros, soporte de reescritura de consultas e indexación GIN.
- PL/Python: ahora soporta parámetros con nombre y funciones que devuelven conjuntos.

2.2.6 LICENCIA

[MB06], difunde que PostgreSQL usa la licencia *BSD* que sólo requiere que el código fuente licenciado mantenga su información de derechos y licenciamiento. Esta licencia certificada como Iniciativa del Código Abierto (“*OSI*”), o, “*Open Source Initiative*“, ampliamente apreciada por ser flexible y amigable a los negocios, dado que no restringe el uso de PostgreSQL con aplicaciones propietarias y comerciales.

Junto con el soporte provisto por múltiples compañía y propiedad pública del código, la licencia *BSD* hace PostgreSQL muy popular con proveedores que desean incluir la base de datos en sus productos sin temor a honorarios, trabas de venta o cambios en los términos de licenciamiento.

De acuerdo a [TE06], PostgreSQL es distribuido bajo la licencia *BSD* y no tiene restricciones acerca de la manera en que puede usarse el código fuente.

La licencia *BSD*, otorga el permiso de uso, copia, modificación y distribución del software y de su documentación para cualquier propósito, sin cobro, y sin un acuerdo escrito.

2.2.7 PLATAFORMAS QUE SOPORTA POSTGRESQL

[TE06], define que, cualquier plataforma moderna compatible con Unix debería permitir ser capaz de ejecutar PostgreSQL.

PostgreSQL funciona nativamente sobre sistemas operativos basados en Microsoft Windows NT tales como Win2000, WinXP y Win2003. Y las versiones de Windows basadas en MS-DOS (Win95, Win98, WinMe) usando Cygwin.

Existe incluso una versión para Novell Netware 6 y una versión para OS/2.

2.3 APACHE

2.3.1 INTRODUCCION

Actualmente Apache es el servidor Web numero uno en la implementación a nivel mundial [RJ06]. Más del 60% de los servidores de Internet utilizan Apache.

Apache es de código libre, robusto, con funcionalidades equivalentes a la de los servidores comerciales y esta controlado por un grupo de voluntarios en todo el mundo mediante Internet y estos voluntarios se conocen como el grupo apache.

Fue en febrero de 1995, que el servidor Web mas popular era un *HTTP* de dominio publico desarrollado en el Centro Nacional de Aplicaciones de Supercomputación (“*NCSA*”), o “*National Center for Supercomputing Applications*”, en la Universidad de Illinois. Al abandonar su trabajo McCool, quien fue el principal desarrollador de *NCSA* en 1994, la evolución del programa había quedando reducido a nada y el desarrollo paso a manos de los responsables de sitio Web quienes introdujeron mejoras en sus servidores.

A principios del año siguiente ocho programadores formaron el grupo apache quienes utilizaron el código original de *NCSA* 1.3 para crear lo que hoy es APACHE. Posteriormente publicaron Apache 0.6.2 lo que seria la primera versión utilizable, en abril de 1995. Desde ese momento el desarrollo de apache

siguió dos líneas paralelas, por un lado incorporar mejoras y por otro reescribir por completo el código y en julio se migraron las arquitecturas para apache 0.7 y se hace publico apache 0.8.

El 1 de diciembre de 1995 aparece apache 1.0 que incluyo documentación y muchas mejoras en forma de módulos incrustables y en poco tiempo se convirtió en el más usado en Internet. En 1999 los integrantes del grupo apache fundaron la Apache Software Foundation.

Apache posee un sinnúmero de características; como correr en una multitud de Sistemas Operativos, poseer una tecnología gratuita de código fuente abierto, ser un servidor altamente configurable de diseño modular, trabajar con *PHP* y diversos *script*, además soporta tecnología *Java* y paginas *jsp*.

Apache es uno de los primeros servidores Web en integrar el protocolo *HTTP* 1.1, en soporte de *CGI*, *Host*¹⁶ Virtuales y Servidor Proxy integrado.

Los cambios hacia apache 2.0, surge del hecho de que el modelo de apache 1.3 no trabaja bien en plataformas que no son centradas en procesos. En Apache 2.0 se crean los Módulos de Multiprocesamiento (“*MPMs*”), o “*Multiprocessing Modules*”, que son los responsables de iniciar los procesos del servidor y servir las solicitudes a través de procesos hijos.

Los Prefork *MPM* permiten crear una pila de procesos hijos para servir las solicitudes. Threaded *MPM* habilita el soporte para hilos en apache 2.0. Perchild *MPM* permite un número prefijado de procesos hijos que son iniciados con un número específico de hilos. WinNT *MPM* es un modulo de implementación para Windows, básicamente un modulo para multihilos.

Las nuevas funcionalidades incluyen el Hebrado en Unix, que permite mejorar la estabilidad y el soporte multiprotocolo que tiene la infraestructura necesaria para

¹⁶ Host, es un ordenador que funciona como el punto de inicio y final de las transferencias de datos.

servir distintos protocolos. La versión 2.0 de apache es más rápida y más estable en sistemas que no son tipo Unix.

Los módulos de Apache pueden escribirse para que se comporten como filtros que actúan sobre el flujo de contenidos.

Los módulos de multiprocesamiento son los responsables de conectar con los puertos de red de la maquina, la extensión del diseño modular ofrece beneficios como: Permitir soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos y permiten personalizar mejor las necesidades de cada sitio Web.

El software gratuito como Apache, *Perl* y Linux no comparte los mismos acuerdos lo cual ha generado confusión asociando estos paquetes en la misma categoría de licencia.

Apache es un descendiente de la licencia *BSD*, los cambios que se realicen pueden no hacerse públicos.

2.3.2 SERVIDOR APACHE

Según los autores [APAC, MC04], Apache es un servidor Web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la Web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Estos voluntarios se conocen como el *Apache Group*: cientos de personas que contribuyen al proyecto con código, ideas y documentación.

2.3.3 HISTORIA

De acuerdo a [APAC, MC04], revelan que, en febrero de 1995, el servidor Web más popular de Internet era un servidor de *HTTP*, de dominio público

desarrollado en la *NCSA*, en la Universidad de Illinois, no obstante, al dejar Rob McCool quien fue el principal desarrollador del servidor *NCSA* en 1994, la evolución del programa había quedado prácticamente reducido a la nada.

El desarrollo pasó a manos de los responsables de sitios Web, que progresivamente introdujeron mejoras en sus servidores, y un grupo usando el correo electrónico como herramienta básica de coordinación, se situó en poner estas mejoras en forma de “parches”. Dos de estos desarrolladores, Brian Behlendorf y Cliff Skolnick, pusieron en marcha una lista de correo, un espacio de información compartida y un servidor en California donde los desarrolladores principales pudiesen trabajar.

A principios del año siguiente, ocho programadores formaron el grupo Apache. Según [RJ06], el grupo utilizó el código original del servidor de *NCSA* 1.3 para crear lo que ahora se conoce como APACHE.

Como base de trabajo añadieron todas las correcciones de errores publicadas y las mejoras más valiosas que encontraron y probaron el resultado en sus propios servidores. Posteriormente publicaron lo que sería la primera versión oficial del servidor Apache la 0.6.2, en abril de 1995. Casualmente, en esas mismas fechas, *NCSA* reemprendió el desarrollo del servidor *NCSA*.

En ese momento el desarrollo de Apache siguió dos líneas paralelas. Por un lado, un grupo de desarrolladores siguió trabajando sobre el Apache 0.6.2 para producir la serie 0.7, incorporando mejoras, etc. Un segundo grupo reescribió por completo el código, creando una nueva arquitectura modular. En julio de 1995 se migraron a esta nueva arquitectura las mejoras existentes para Apache 0.7, haciéndose público como Apache 0.8. El día 1 de diciembre de 1995 apareció Apache 1.0. [RJ06], dice que el desarrollo de apache esta basado en el modelo de diseño de aplicaciones de software libre, descrito por Eric Raymond en su artículo “La catedral y el bazar”.

La Versión 1.0 incluyó documentación y muchas mejoras en forma de módulos incrustables. Poco después Apache sobrepasó al servidor de *NCSA* y se convirtió

en el más usado en Internet, posición que ha mantenido hasta nuestros días. En 1999 los miembros del Grupo Apache fundaron la Apache Software Foundation, que provee soporte legal y financiero al desarrollo del servidor Apache y los proyectos laterales que han surgido de éste.

2.3.4 APACHE 2.0

2.3.4.1 CAMBIOS HACIA APACHE 2.0

La incorporación más importante que se hizo en apache 2.0 fue el uso de *MPMs* o Multiprocessing modules.

En apache 1.3 se utilizó una tecnología denominada Preforking. En esta arquitectura un proceso padre de apache simplemente monitorea los procesos hijos, compartiendo, matando, etc., basado en la cantidad de solicitudes recibidas.

El modelo de apache 1.3 no trabaja bien en plataformas que no son centradas en procesos, un ejemplo de ellas es Windows de Microsoft. *MPM* es responsable de iniciar los procesos del servidor y por servir las solicitudes a través de procesos hijos o hilos, dependiendo de la implementación de *MPM*.

Prefork *MPM*: Permite crear una pila de procesos hijos para servir las solicitudes. Cada proceso hijo tienen su propio hilo. Por ejemplo, si apache inicia 30 procesos hijos, este puede servir 30 solicitudes simultáneamente. Si algo falla, y un proceso hijo muere, solamente ese hilo de solicitudes se pierde.

Threaded *MPM*: Este método habilita el soporte para hilos en apache 2.0. Funciona de manera similar que Prefork *MPM*, pero en lugar de que cada proceso hijo tenga un hilo individual, cada proceso hijo puede tener un número específico de hilos. Cada hilo en el proceso hijo puede servir a diferentes solicitudes. Si apache inicia 30 procesos hijos donde cada hijo

tiene 10 hilos como máximo, apache podría servir 300 solicitudes Simultáneamente.

Perchild MPM: Este método también es nuevo sobre apache. En este método un número prefijado de procesos hijos son iniciados con un específico número de hilos. Cuando la carga de solicitudes se incrementa los procesos agregan nuevos hilos para cubrir la demanda. Cuando las solicitudes se reducen, los procesos disminuyen sus hilos usando mínimos y máximos prefijados.

WinNT MPM: Este modulo es la implementación de *MPM* para Microsoft Windows. Básicamente es un modulo para multihilos. Usando este modulo apache puede crear un proceso padre y procesos hijos bajo Windows. Los procesos hijos crean todos los hilos necesarios para el número de solicitudes.

En apache 2 otra característica agregada conocida es Filtering I/O, esta capacidad permite que una salida de un modulo de apache, sirva de entrada para otro modulo del servidor.

Otra de las mejoras es la implementación del modulo (*mod_cgid*) que permite desarrollar la misma tarea de (*mod_cgi*) pero utilizando un proceso mas avanzado para comunicarse con los procesos. Además de las mejoras que apache 2 presenta, el Apache Group trabaja en la implementación de Apache Portable en Tiempo de Ejecución (“*APR*”), o “*Apache Portable RunTime*”, esta es una plataforma que funcionara sobre el sistema operativo, permitiendo un funcionamiento similar a Entorno en Tiempo de Ejecución *Java* (“*JRE*”), o “*Java Runtime Environment*”.

2.3.4.2 FUNCIONALIDADES

Hebrado en Unix: En los sistemas Unix que soportan hebras la Interfaz de Sistema Operativo Portable Basado en UNIX (“*POSIX*”), o “*Portable Operating System Interfaces, Unix Based*”, la nueva versión de Apache puede ejecutarse en modo híbrido multiproceso-multihebra, esto permite

mejorar la escalabilidad para muchos, aunque no para todas las configuraciones.

Nuevo sistema de configuración y compilación: El sistema de configuración y compilación ha sido escrito de nuevo desde cero para basarlo en (*autoconf*) y (*libtool*). Esto hace que el sistema de configuración de Apache se parezca ahora más al de otros proyectos Open Source.

Soporte Multiprotocolo: La nueva versión tiene la infraestructura necesaria para servir distintos protocolos. Por ejemplo, se ha escrito el módulo (*mod_echo*).

Soporte mejorado para las plataformas que no son tipo Unix:

La versión 2.0 de Apache es más rápida y más estable en sistemas que no son tipo Unix, tales como BeOS, OS/2 y Windows, que la versión antigua. Con la introducción de *MPMs* específicos para cada plataforma y del *APR*, estas plataformas tienen ahora implementada su propia *API* nativa, evitando las capas de emulación *POSIX* que provocan problemas y un bajo rendimiento.

Nueva interfaz de programación (API) de Apache: Ha cambiado significativamente en la nueva versión. Muchos de los problemas de ordenación y prioridad de módulos de la versión 1.3 desaparecieron. Apache 2.0 hace automáticamente mucho de lo que es necesario, y la ordenación de módulos se hace ahora por *hooks*, lo que ofrece una mayor flexibilidad. También se han añadido nuevas llamadas que ofrecen capacidades adicionales sin tener que parchear el *núcleo* del servidor Apache.

Soporte de IPv6: En los sistemas que soportan *IPv6* con la librería Apache Portable Runtime, Apache soporta *IPv6 listening sockets* por defecto.

Filtros: Los módulos de Apache pueden ahora escribirse para que se comporten como filtros que actúan sobre el flujo de contenidos tal y como

salen del servidor o tal y como son recibidos por el servidor. Esto permite, por ejemplo, que el resultado de un *script CGI* sea analizado por las directivas Server Side Include usando el filtro *INCLUDES* del módulo (*mod include*). El módulo (*mod ext filter*) permite que programas externos actúen como filtros casi del mismo modo que los *CGIs* pueden actuar como *handlers*.

Mensajes de error en diferentes idiomas: Los mensajes de error que se envían a los navegadores están ahora disponibles en diferentes idiomas, usando documentos *SSI*. Estos mensajes pueden personalizarse por el administrador del sitio Web para conseguir un *look and feel* coherente con el resto de los contenidos.

Configuración Simplificada: Muchas directivas que podían inducir a confusión han sido simplificadas. Las directivas *Port* y *BindAddress* han desaparecido; para configurar la dirección *IP* en la que escucha el servidor ahora se usa únicamente la directiva *Listen*; la directiva *ServerName* especifica el nombre del servidor y el número del puerto solo para redireccionamiento y reconocimiento de *host* virtual.

Soporte de Unicode Nativo para Windows NT: Apache 2.0 en Windows NT usa ahora utf-8 para la codificación de los nombres de fichero. Estos se mapean directamente al sistema de ficheros único subyacente, suministrando soporte para diferentes idiomas para todas instalaciones en Windows NT, incluidos Windows 2000 y Windows XP. Este soporte no se extiende a Windows 95, 98 o ME, que continúan usando la codificación que tenga la máquina local para el acceso al sistema de archivos.

Actualización de la librería de expresiones regulares (regular expresión): Apache 2.0 incluye la librería de expresiones regulares compatibles de con *Perl*. Ahora, cuando se evalúan las expresiones tipo, se usa siempre la potente sintaxis de *Perl 5*.

2.3.5 Módulos de MultiProcesamiento (MPMs)

Según [APAC] Apache está diseñado para ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades, o que una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios Web elegir que funcionalidades van a ser incluidas en el servidor seleccionando que módulos se van a usar, ya sea al compilar o al ejecutar el servidor.

Apache 2.0 extiende este diseño modular hasta las funcionalidades más básicas de un servidor Web. El servidor viene con una serie de Módulos de MultiProcesamiento que son los responsables de conectar con los puertos de red de la máquina, aceptar las peticiones, y generar los procesos hijo que se encargan de servirlos.

La extensión del diseño modular a este nivel del servidor ofrece dos beneficios importantes:

- Apache puede soportar de una forma más fácil y eficiente una amplia variedad de sistemas operativos. En concreto, la versión de Windows de Apache es mucho más eficiente, porque el módulo *mpm_winnt* puede usar funcionalidades nativas de red en lugar de usar la capa *POSIX* como hace Apache 1.3. Este beneficio se extiende también a otros sistemas operativos que implementan sus respectivos *MPMs*.
- El servidor puede personalizarse mejor para las necesidades de cada sitio Web. Por ejemplo, los sitios Web que necesitan más que nada escalabilidad pueden usar un *MPM* hebrado como *worker*, mientras que los sitios Web que requieran por encima de otras cosas estabilidad o compatibilidad con software antiguo pueden usar *prefork*. Además, se pueden configurar funcionalidades especiales como servir diferentes *hosts* con diferentes identificadores de usuario.

A nivel de usuario, los módulos de multiprocesamiento (*MPMs*) son como cualquier otro módulo de Apache. La diferencia más importante es que solo un *MPM* puede estar cargado en el servidor en un determinado momento.

2.3.6 CARACTERÍSTICAS

En general apache es un servidor flexible, rápido y eficiente, continuamente actualizado y tiene muchas características deseables en un servidor Web, entre ellas se citan:

- De acuerdo a [CA00], es un servidor de Web con forme al protocolo *HTTP* 1.1.
- Según [APAC], corre en una multitud de Sistema Operativos como en Windows NT, 2000 y XP, además en Windows 9x y ME, como una aplicación de consola en Linux, y en Unix inicialmente creado solo para PC, y otros como FreeBSD, Solaris y Mac OS X.
- Es una tecnología gratuita de código fuente abierto esto le da una transparencia a este software de manera que es posible auditarlo.
- Es un servidor altamente configurable de diseño modular, lo cual permite ampliar las capacidades de servidor Web. Además permite que cualquier persona que posea experiencia en programación de *C* o *Perl* puede escribir un módulo para una función determinada.
- Trabaja con *PHP* y diversos lenguajes en *scripting*, entre ellos se destaca *Perl*, por lo cual apache utiliza su parte de *Perl* para soporte *CGI* y (*mod_perl*), además es soporte para la tecnología *Java* y paginas *jsp.*, que se utiliza para tener paginas dinámicas.
- Permite personalizar las respuestas ante los posibles errores que se pueden generar en el servidor.
- Tiene una alta configurabilidad en la creación y gestión de Logs, lo cual permite tener un mayor control en el servidor.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Posee un elaborado índice de directorios; un directorio de alias, negociación de contenidos; informe de errores; *HTTP* configurable;

ejecución de programas *CGI*; gestión de recursos para procesos hijos; integración de imágenes del lado del servidor y reescritura del Identificador de Recursos Uniforme (“*URL*”), o “*Uniform Resource Locator*”.

- De acuerdo a [HOSP], Apache es uno de los primeros servidores Web en integrar el protocolo *HTTP* 1.1 y es compatible con *HTTP* 1.0, ejemplo: antes del *HTTP* 1.1 un navegador Web tenía que esperar una respuesta del servidor Web antes de emitir otra petición. Con el surgimiento de *HTTP* 1.1, un navegador Web puede enviar solicitudes en paralelo, los cuales ahorran el ancho de banda dejando de transmitir las cabeceras *HTTP* en cada solicitud.
- Soporte para *CGI*, utiliza módulos compatibles con *CGI* y aporta características extendidas como personalización de las variables de entorno y soporte de reparación de errores.
- Soporta *FastCGI*, un modulo que permite que las aplicaciones arranquen rápidamente.
- Soporte de **Host** Virtuales: Apache es uno de los primeros servidores Web en soportar tanto **host** basados en Protocolo de Internet (“*IP*”), o “*Internet Protocol*“, como **host** Virtuales.
- Soporte de autenticación *HTTP*: Apache soporta autenticación básica basada en la Web, “message digest”, autenticación básica utilizando archivos estándar de contraseña o a través de Administración de Base de Datos (“*DBM*”), o “*DataBase Management*”, mediante llamadas a *SQL*; o llamadas a programas externos de autenticación.
- Servidor Proxy integrado: Permite convertir a apache en un servidor Proxy cache.
- Soporte para Incluir Lado del Servidor (“*SSI*”), o “*Server Side Include*”: Apache contiene un conjunto de *SSI* que añade una gran cantidad de flexibilidad para el desarrollador de sitios Web.
- Apache es portable en tiempo de ejecución, se hizo patente la necesidad de la portabilidad de apache en su versión 2.2, antes de la versión actual que se encuentra en el mercado, apache tenía la portabilidad interna, lo cual lo hacía que el código base fuera menos manejable.

2.3.7 LICENCIA

Los software gratuitos como Apache, *Perl* y Linux (un sistema operativo clon de Unix basado en x86) van de soportar una gran presión debido a la decisión de Netscape de fabricar el Netscape Communicator, uno de los navegadores Web mas populares.

El software gratuito como Apache, *Perl* y Linux no comparte los mismos acuerdos de licencia y se ha creado confusión asociando estos paquetes en la misma categoría de licencia.

Todo software gratuito trata de serlo para todos. Sin embargo, hay algunas restricciones legales que fuerzan las licencias de software individuales. Por ejemplo Linux, que es gratuito desde la Licencia Pública general, exige que cualquier cambio en Linux se haga público. Apache es una descendiente de la licencia *BSD*, por su lado, no necesita que los cambios hechos en Apache se hagan publico, es decir piense en Apache como un software gratuito, un software con derechos de autor publicado por el Grupo Apache.

Apache no es ni de dominio publico ni de prueba además no esta cubierto por *GPL*.

2.4 GNU/Linux

2.4.1 INTRODUCCIÓN

La historia del Proyecto *GNU*, se origina cuando Stallman le llamó a su documento inicial Manifiesto *GNU*, y de ahí se origina el término (libre). El cuál se refiere a libertad no a precio, y en 1983 el proyecto *GNU* fue concebido, para dar soporte al ideal de software libre.

Stallman en 1971 comenzó su carrera como programador en el Instituto Tecnológico de Massachusetts ("*MIT*"), o "*Massachusetts Institute of*

Technology“, y en aquella época se solía compartir el software y el código fuente. En los 80 el software era casi todo *privativo*¹⁷, o copia permitida que es el método más simple de hacer que un programa sea libre, permitiendo compartir programas y mejoras a la gente, pero no en su totalidad, por esta razón Stallman considera necesario el Proyecto *GNU*, que como primer objetivo tenía crear un Sistema Operativo libre, y compatible con Unix.

En los 90 ya tenía casi todos los componentes excepto el *núcleo*, o *kernel* el cual es la parte fundamental de un Sistema Operativo, mientras tanto Linus Trovalds llevaba desarrollado Linux, un *núcleo* libre compatible con Unix. Combinando Linux con el ya casi completo sistema *GNU* se obtuvo un sistema operativo completo denominado GNU/Linux.

Otro de los proyectos que ha producido independientemente otro sistema operativo libre de tipo Unix es *BSD*, que en los 80 cuando se desarrolló no era libre, pero ya en los 90 se convirtió en libre.

2.4.2 HISTORIA DEL PROYECTO GNU

Según [RX04], Richard Stallman le llamó a su documento inicial Manifiesto *GNU*, el cual analiza que se escogió como nombre "*GNU*" porque cumplía tres requisitos: Primero, era un acrónimo recursivo: "*GNU* No es Unix"; Segundo, ya existía esa palabra (*N. del T.: en inglés Gnu significa Ñu*), y Tercero, porque era divertido decirlo.

La palabra (*libre*) se refiere a libertad y no a precio. Se puede o no pagar un precio por obtener software *GNU*. En 1983 el proyecto *GNU* fue concebido como una forma de devolver el espíritu cooperativo que prevalecía en la comunidad computacional y hacer la cooperación posible al eliminar los obstáculos impuestos por los dueños de *software privativo*.

[PH04], difunde que Richard Stallman en 1971, comenzó su carrera en el *MIT*, y en esa época, compartir el software o el código fuente era algo habitual, por lo

¹⁷ Software Privativo, es software que no es libre ni semilibre, y el uso, redistribución o modificación están prohibidos y requieren que solicite una autorización.

que incluso compañías informáticas frecuentemente distribuían software libre. Los programadores eran libres de cooperar unos con otros, y frecuentemente lo hacían.

En el laboratorio de Inteligencia Artificial de la Universidad se utilizaba un sistema operativo llamado Sistema Incompatible de Tiempo Compartido (“ITS”), o “*Incompatible Timesharing System*”, diseñado en lenguaje ensamblador por el mismo personal del laboratorio para el Procesador de Datos Programado modelo 10 (“PDP-10¹⁸”), u “*Programmed Data Processor model 10*”, de Digital, uno de los más importantes ordenadores de aquella época.

En los 80, casi todo el *software era privativo*, lo cual se refiere a que es el uso de redistribución o modificación estaban prohibidos. Esta política generó la imposibilidad de la cooperación entre programadores y usuarios, y esto fue uno de los motivantes del proyecto GNU.

Por que cada usuario de computadoras necesita un sistema operativo; si no existe un sistema operativo libre, no se puede ni siquiera comenzar a usar una computadora sin recurrir a un *software privativo*. Así que el primer elemento en la agenda del software libre fue crear un sistema operativo libre.

Un sistema operativo no es sólo el *núcleo* o *Kernel*; sino que también incluye compiladores, editores, formateadores de texto, software de correo y muchas otras cosas. Por todo esto, escribir un sistema operativo completo es un trabajo bastante grande para lo cual se necesita muchos años.

Dice [RX06] que Stallman decide hacer el sistema operativo compatible con Unix porque el diseño en general ya estaba probado y era portable, y la compatibilidad hacía fácil para los usuarios de Unix cambiar a GNU, y el objetivo inicial de un sistema operativo libre parecido al Unix había sido alcanzado.

En los 90, ya se encontraron o se escribieron los componentes principales, excepto uno el *núcleo*. Entonces en aquel tiempo Linux, un *núcleo* libre, fue

¹⁸ PDP-10, es un ordenador fabricado a finales de los años 1960 por Digital Equipment Corporation.

desarrollado por Linus Torvalds. Combinando Linux con el ya casi completo sistema *GNU* se consigue un sistema operativo completo: un sistema *GNU* basado en Linux denominado GNU/Linux. Se estima que hay cientos de miles de personas que ahora usan sistemas *GNU* basados en Linux., incluyendo *Slackware*¹⁹, *Debian*, Red Hat y otros.

El proyecto *GNU* no está limitado a sistemas operativos, además Stallman manifiesta que desea proporcionar un amplio espectro de software, esto incluye software de aplicación además desea ampliar *GNU Emacs*²⁰ hasta llegar a un sistema de publicación tipo WYSIWYG (*lo que ves es lo que obtienes*) en los años siguientes.

También Stallman manifiesta que desean proporcionar software para usuarios que no son expertos, por tal razón están trabajando en un (*ambiente de escritorio de arrastrar-y-soltar*) para ayudar a los principiantes a usar un sistema *GNU*, por tal motivo ofrece juegos y otras recreaciones, por lo que en el software libre no hay límites, excepto cuando las leyes como el sistema de patentes prohíban el software libre completamente.

Stallman manifiesta que el objetivo final es el de proporcionar software libre para hacer todos los trabajos que los usuarios de computadoras deseen realizar y por lo tanto hacer al *software privativo* obsoleto.

2.4.3 LINUX Y EL PROYECTO GNU

[RS07] dice que numerosos usuarios de computadoras usan una versión modificada del Sistema *GNU*, a diario. Debido a su específico giro de acontecimientos, el nombre más ampliamente usado con frecuencia es Linux, pero en general no se es consciente del alcance de su conexión entre Linux y el Proyecto *GNU*.

¹⁹ Slackware, es una distribución de Linux, es un complemento de sistemas multitarea de 32-bits, y la más antigua de las activamente mantenidas en la actualidad.

²⁰ GNU Emacs, es parte del proyecto GNU, activamente desarrollado como “un editor extensible, personalizable, auto-documentado y de tiempo real”.

El Linux que se utiliza no es el sistema operativo. Linux es el *núcleo*: El programa del sistema que asigna los recursos de la máquina a los otros programas que se ejecuta, por lo que el *núcleo* es una parte esencial de todo sistema operativo, pero inútil por sí solo; sólo puede funcionar en el contexto de un sistema operativo completo. Linux se usa normalmente en combinación con el sistema operativo *GNU*: el sistema completo es básicamente *GNU*, con Linux actuando de *núcleo*.

Generalmente los programadores saben que Linux es un *núcleo*. El sistema *GNU* es el software libre disponible que conforma un sistema completo porque el Proyecto *GNU* había estado trabajando desde 1984 en hacer uno. La meta era desarrollar un sistema libre de tipo Unix, llamado *GNU*. El anuncio oficial del Proyecto *GNU* también apunta a varios de los planes originales para el sistema *GNU*. Para cuando se escribió Linux, el sistema estaba casi terminado.

La mayor parte de los proyectos de software libre tienen el objetivo de desarrollar un programa particular para una tarea particular. Por ejemplo, Se propusieron escribir, Linus Torvalds un *núcleo* de tipo Unix (Linux); Donald Knuth un formateador de textos (*TeX*²¹); Bob Scheifler un sistema de ventanas (el *Sistema de Ventanas X*²²).

[RS07] difunde que el Proyecto *GNU* no fue, y no será, un proyecto para desarrollar paquetes de software específicos. Porque no fue un proyecto para desarrollar un compilador de *C*, aunque se lo realizó. La finalidad del Proyecto *GNU* era desarrollar: (*Un sistema libre de tipo Unix completo: GNU*).

Stallman agradece a las personas que han hecho importantes contribuciones al software libre del sistema. La razón por la que es un *Sistema Integrado*²³ y no sólo una colección de programas útiles es que el Proyecto *GNU* se había propuesto hacerlo y por tal motivo se realizó una lista de los programas necesarios para hacer un sistema libre (*completo*), y sistemáticamente

²¹ TeX, es un sistema de tipografía, muy popular en el ambiente académico, especialmente entre las comunidades de matemáticos, físicos e informáticos.

²² Sistema de Ventanas X, se utiliza para dotar de una interfaz gráfica a los Sistemas Unix, y distribuye el procesamiento de aplicaciones especificando enlaces cliente-servidor.

²³ Sistema Integrado, es un Sistema Informático de uso específico construido dentro de un dispositivo mayor, y es utilizado para usos muy diferentes.

encontraron, escribieron y buscaron gente para escribir todo lo que había en la lista. Escribir componentes esenciales pero poco divertidos como el ensamblador de *GNU* o *GAS*²⁴ y el enlazador *GLD*²⁵, es necesario porque no se puede tener un sistema sin ellos. Algunos de los componentes del sistema: las herramientas de programación, se hicieron populares por sí mismas entre los programadores, pero se escribieron muchos componentes que no son herramientas tales como el de el Renacimiento Sobre el Bourne Shell ("*BASH*"), o "*Bourne Again Shell*", el interprete *Ghostscript*²⁶ y la Biblioteca de *C* de *GNU* que no son herramientas de programación. También se desarrolló un juego de ajedrez, *GNU Chess*²⁷, porque un sistema completo necesita también buenos juegos.

A principios de los años 90 se había preparado todo el sistema salvo el *núcleo*, pero estaban trabajando en un *núcleo*, el *GNU Hurd*²⁸, que funcionaba sobre *Mach*²⁹. Por lo que se estaba preparando al lanzamiento real del sistema *GNU*, con el *GNU Hurd*.

Pero afortunadamente no tuvieron que esperar al *Hurd*³⁰, porque Linux ya estaba disponible. Cuando Linus Torvalds escribió Linux, rellenó la última laguna importante. Y se pudo poner Linux junto al sistema *GNU* para obtener un sistema libre completo: una versión basada en Linux del sistema *GNU*; el sistema GNU/Linux, para abreviar, es un derivado del sistema *GNU* que utiliza Linux como *núcleo* en vez del *Hurd* de *GNU*.

Publica [RS07] que no fue sencillo hacer la interfaz. Algunos componentes tales como la biblioteca de *C* de *GNU* necesitaron cambios sustanciales para funcionar con Linux. Integrar un sistema completo en forma de una distribución que

²⁴ Gas, o GNU Assembler, es el ensamblador del proyecto GNU, que permite ensamblar el código generado en el lenguaje de microcontrolador.

²⁵ GLD, permite enlazar todos los archivos compilados y las librerías necesarias para generar un archivo único listo para ser ejecutado por el microcontrolador.

²⁶ Ghostscript, es el programa intérprete por excelencia de documentos en formatos PS y también PDF.

²⁷ GNU Chess, es un programa de computador para jugar ajedrez, basado en Unix.

²⁸ GNU Hurd, es un conjunto de programas servidores que simulan un kernel Unix que establece la base del sistema operativo GNU.

²⁹ Mach, es un kernel desarrollado para soportar la investigación de sistemas operativos de manera específica computación distribuida y paralela.

³⁰ Hurd, es un conjunto de servidores y se ejecuta sobre un microkernel y forma parte del sistema operativo GNU.

funcionara sin más que (*desembalarla*) fue también un gran trabajo. Esto precisó resolver la cuestión de cómo instalar y arrancar el sistema.

El Proyecto *GNU* apoya tanto a los sistemas *GNU/Linux* como al sistema *GNU*, se financia la reescritura de las extensiones relacionadas con Linux de la biblioteca de *C* de *GNU*, de modo que se integran bien, y los sistemas *GNU/Linux* modernos usan la versión actual de la biblioteca sin necesidad de hacerle modificaciones. También financiaron las primeras etapas del desarrollo de *Debian*³¹ *GNU/Linux*. Hoy en día se usan sistemas *GNU* basados en Linux para la mayor parte del trabajo.

[RS07] manifiesta que además de *GNU*, otro proyecto ha producido independientemente otro sistema operativo libre de tipo Unix. Este sistema es conocido como *BSD*, y fue desarrollado en la Universidad de California en Berkeley. En los 80 no era libre, pero se volvió libre a principios de los 90, por lo que cualquier sistema operativo libre hoy día es casi seguro que es una variante del sistema *GNU*, o algún tipo de sistema *BSD*.

Stallman relata que la gente pregunta a veces si *BSD* es también una versión de *GNU*, como *GNU/Linux*. Para hacer su código libre, los desarrolladores de *BSD* se inspiraron en el ejemplo del Proyecto *GNU*, pero su código se sobreponía poco con el de *GNU*, por tal razón los sistemas *BSD* de hoy en día usan algunos programas *GNU*, del mismo modo que el sistema *GNU* y sus variantes usan algunos programas de *BSD*; pero si se analiza a los dos sistemas son muy diferentes y evolucionan por separado.

2.4.3.1 EL SISTEMA *GNU*

[RX06] manifiesta que el sistema *GNU* es un sistema operativo similar a Unix, constituido en su totalidad por software libre [PH04] dice que por lo tanto esta permitiendo ser portable y que los usuarios pueden cambiarse a el con facilidad.

³¹ *Debian*, es un sistema operativo libre, utiliza el núcleo Linux, y la mayor parte de las herramientas básicas provienen del Proyecto GNU.

El sistema *GNU* incluye todo el software *GNU*, además de muchos otros paquetes, como el *Sistema de Ventanas X* y *TeX*, los cuales no son software de *GNU*.

La primera versión de prueba del sistema *GNU* completo estuvo disponible en 1996, el cual incluía el *Hurd* de *GNU*, el *núcleo*. En 2001 el sistema *GNU* incluyendo el *Hurd* de *GNU*, comenzó a funcionar con bastante fiabilidad, pero el *Hurd* todavía carecía de algunas características importantes, por lo que no se usaba ampliamente. Entretanto, el sistema GNU/Linux, un derivado del sistema *GNU* que utiliza Linux como *núcleo* en vez del *Hurd* de *GNU*, ya tuvo un gran éxito desde los años 1990.

Ya que el propósito de *GNU* es ser un sistema libre, cada una de las piezas que lo componen deben ser software libre. Sin embargo, no todas tienen por que estar protegidas por *copyleft*; legalmente se puede incluir cualquier tipo de software libre si este ayuda a alcanzar los objetivos técnicos propuestos, no es necesario que cada componente sea software *GNU* individualmente. El sistema *GNU* puede incluir, software libre no protegido por *copyleft*, como es el caso del *Sistema de Ventanas X*, que fue desarrollado en otros proyectos.

2.4.3.2 EL SOFTWARE DE *GNU*

Según [RX06], el software de *GNU* es el software liberado bajo el auspicio del Proyecto *GNU*, por lo que un programa que sea software de *GNU*, también se denomina programa *GNU* o paquete *GNU*. El archivo README o el manual del paquete *GNU* debe indicar que lo es; además, el directorio de Software Libre identifica todos los paquetes *GNU*.

La mayoría del software de *GNU* está protegido por *copyleft*, pero no todo; sin embargo, todo el software *GNU* debe ser software libre.

Parte del software *GNU* lo escribe el personal de la Fundación para el Software Libre, pero la mayoría del software lo aportan diversos voluntarios y a veces el titular de los derechos de autor es la Fundación para el Software Libre y en otras son los propios colaboradores que lo escribieron.

2.4.4 LA HISTORIA DE UBUNTU

Según [TUB], el origen del nombre es debido a un movimiento encabezado por el obispo Desmond Tutu y Edgar Granados, quienes ganaron el Premio Nóbel de la Paz en 1984, por sus luchas en contra del Apartheid en Sudáfrica y también al origen sudafricano de Shuttleworth, además por tener ciertas similitudes entre los ideales de los proyectos *GNU*.

Sin embargo [QL06], difunde que Ubuntu es una palabra africana que significa “Humanity to others (*Humanidad para otros*)” o “I am what I am because of who we all are (*Yo soy lo que soy por quien todos somos*)”.

Según [TUF], el 8 de julio del 2004, Mark Shuttleworth y la Empresa Canonical Ltda. Anunciaron la creación de la distribución de Ubuntu, lo cual tuvo un financiamiento inicial de 10 millones de dólares.

Shuttleworth es un informático y empresario muy importante en Sudáfrica, que además fue el primer cosmonauta sudafricano en la historia.

Canonical Ltda. Es una empresa privada de Mark Shuttleworth, quien se encarga de la promoción de proyectos de software libre, como Ubuntu, también ha desarrollado otros proyectos como The Open CD (*una colección de programas libres para Windows*) y Bazaar (una implementación de *GNU Arch*³²).

La creación de Ubuntu nació por iniciativa de algunos programadores de los proyectos *Debian*, *Gnome*³³ y *arch*³⁴, que se encontraban decepcionados con la manera de operar del proyecto *Debian*, estos programadores buscaron el apoyo económico de Mark Shuttleworth que tenía el suficiente dinero y confianza en el proyecto como para llevarlo a cabo.

Combinando su experiencia en la creación de nuevas empresas, y con el talento y la experiencia de los programadores de la plataforma Linux, nació la empresa

³² GNU Arch, es un sistema muy potente para el control de versiones.

³³ Gnome, es un entorno de escritorio amigable para usuarios basados en ventanas.

³⁴ Arch, optimizada para la arquitectura i686, y enfoca en la simplicidad a la hora de manejar el sistema.

Canonical, mencionada anteriormente la cual se encarga de sostener económicamente el proyecto mediante la comercialización de servicios y soporte técnico a otras empresas. [TUB], difunde que mientras los programadores armaban el Sistema, Shuttleworth aprovechó para aplicar una pequeña campaña de mercadotecnia para despertar interés en “la distribución-sin-nombre” (en Inglés: the no-name-distro), Tras varios meses de trabajo y un breve período de pruebas, la primera versión de Ubuntu (*Warty Warthog* o *Jabalí Verrugoso*) por fin nació.

2.4.4.1 VERSIONES DE UBUNTU

[FF07] Difunde que Ubuntu utiliza **Gnome**, cada lanzamiento de Ubuntu posee un nombre en clave, es que toda versión tienen un nombre de animal, porque son especies de animales que se puede encontrar en África, de donde proviene la palabra Ubuntu y todos ellos constan de un nombre y con un adjetivo, ambos empiezan con la misma letra, como también un número de versión basado en el año y el mes del lanzamiento. Por ejemplo, la versión 5.04 fue lanzada en abril (04, el mes) de 2005 (5, el año).

Cada versión de Ubuntu es lanzada con seis meses de diferencia con respecto al último lanzamiento, aunque el lanzamiento de la versión 6.06 se demoró más de seis meses, debido a que Canonical Ltda. Quería desarrollar una distribución a la que fuera posible dar soporte técnico durante tres años en el escritorio y cinco años en el servidor.

Canonical provee de soporte técnico y actualizaciones de la seguridad para la mayoría de las versiones de Ubuntu durante 18 meses, a partir de la fecha del lanzamiento. Actualmente existen dos versiones de Ubuntu que cuentan con soporte técnico: la versión 6.06, Largo periodo de Soporte (“LTS”), o “Long Term Support”, y la versión 6.10.

Las diversas versiones de Ubuntu se muestran en la Tabla 2-2.

Versión	Fecha de lanzamiento	Nombre en clave	Fin del soporte técnico
4.10	20 de octubre de 2004	Warty Warthog	30 de abril de 2006
5.04	8 de abril de 2005	Hoary Hedgehog	31 de octubre de 2006
5.10	13 de octubre de 2005	Breezy Badger	Abril de 2007
6.06 LTS	1 de junio de 2006	Dapper Drake	Junio de 2009 (<i>desktop</i>) y junio de 2011 (<i>server</i>)
6.10	26 de octubre de 2006	Edgy Eft	Abril de 2008
7.04	19 de abril de 2007	Feisty Fawn	Octubre de 2008
7.10	Octubre del 2007	Gutsy Gibbon	Abril de 2009

Tabla: 2-2. Versiones de Ubuntu

Las diversas versiones mencionadas anteriormente de detallan a continuación:

➤ **Ubuntu 4.10 – Warty Warthog (Jabalí Verrugoso)**

Según [TUXa], dice que, la primera publicación de Ubuntu fue el 20 de octubre del 2004, y recibió ese nombre (jabalí verrugoso) porque fue publicado “*warts and all*” (*con verrugas y todo*), y por qué empezó con la versión 4. El escritorio era de un color bastante oscuro, el naranja y el marrón simbolizaban las tribus sudafricanas. Un motivo por el que Ubuntu se puso en la cabeza desde comienzo fue el Live CD, que ofrecía a los usuarios el poder probar Ubuntu sin necesidad de instalarlo, algo que ninguna otra distribución de Linux ofrecía a los usuarios de escritorio.

Las principales componentes de acuerdo a [FF07] son:

- **Gnome** - 2.8.0
- Linux **Kernel** -2.6.8.1

- **Firefox**³⁵ -0.10
- OpenOffice -1.1.2
- **XFree86**³⁶ – 4.3.0.1
- Tipo de Instalación: Modo Texto

➤ **Ubuntu 5.04 – Hoary Hedgehog (Erizo Canoso)**

Según [TUXa], que después de seis meses más tarde, como se prometió, llegó Hoary Hedgehog. Los colores del escritorio cambiaron ligeramente pero eran aún bastante oscuros respecto a los actuales. El Live CD fue mejorado y extendido a las versiones de 64bits y Power PC. En abril del 2005, Ubuntu había madurado y había dado a luz una nueva distribución llamada **Kubuntu**, fue construida por la fundación Ubuntu pero en lugar de usar **Gnome** como entorno de escritorio, **Kubuntu** usaba **KDE**.

Las principales funciones de acuerdo a [FF07] son:

- **Gnome** - 2.10
- Linux **Kernel** 2.6.10
- **X.Org**³⁷ 6.8.2
- **Firefox** 1.0.2
- OpenOffice 1.1.3
- **KDE** -3.4
- Tipo de Instalación: Modo Texto

Nota: Esta versión que se menciona a continuación se utilizo para este Trabajo Final, y se manejo para PC con una arquitectura de 32 bits.

➤ **Ubuntu 5.10 – Breezy Badger (Tejón Ventoso)**

Según [TUXa], Ubuntu va rumbo a seguir las estelares trayectorias de sus antecesores y tiene mucha aceptación por lo que se ha convertido en poco

³⁵ Firefox, o Mozilla Firefox es un navegador de Internet multiplataforma, con interfaz gráfica de usuario.

³⁶ XFree86, es una implementación del sistema X Windows System, es software libre y provee de una interfaz gráfica cliente/servidor.

³⁷ X.Org, es una implementación de código abierto del sistema X Window System, que surge como bifurcación de proyecto XFree86.

tiempo en uno de los mejores distribuidores de la comunidad Linux. Con la llegada de Breezy vino de igual manera el nacimiento de *Edubuntu*³⁸, una tercera distribución de Linux orientada al ámbito educacional. También se incluyó soporte de servidor para Ubuntu. Para no quedarse atrás, la distribución *Kubuntu*³⁹ reemplazó *Synaptic*⁴⁰ (*un gestor de paquetes*) por *Adept*⁴¹ y se convirtió en la primera en usar devtags para una búsqueda más rápida de aplicaciones para *Adept*. *KDE*⁴² Bluetooth también fue añadido.

[BP06], manifiesta que cuenta con un soporte de hardware mejorado por su *Kernel* 2.6.12.6, además posee una de los mejores soportes de hardware de todas las distribuciones existentes. Permitiendo la facilitación de instalación de paquetes, mejorando sustancialmente el proceso de instalación de software, por otro lado cuenta con la herramienta de líneas de comandos, la Herramienta Avanzada de Empaquetamiento (“*apt*”), o “*Advanced Packaging Tool*”.

Ubuntu también dispone de *Synaptic*, que ofrece una interfaz gráfica más amistosa para *apt* al momento de utilizar en entorno de ventanas. El grupo de trabajo de Ubuntu creó una entrada en el sistema de menús que enlaza con el administrador de software que admite la instalación rápida y sencilla de paquetes adicionales, además incorporó una herramienta, el “*Lenguaje Selector*”, que ase que sea más fácil instalar un idioma nuevo y permite el soporte para varios lenguajes simultáneamente.

Las principales funciones de acuerdo a [FF07] son:

- Para el Escritorio:
 - *Gnome* - 2.12.1
 - Open Officeorg 2.0 beta 2

³⁸ Edubuntu, es una distribución de Linux y es una rama de Ubuntu diseñada para uso en ambientes escolares.

³⁹ Kubuntu, es una derivación oficial del sistema Operativo Ubuntu, pero con KDE como entorno de escritorio.

⁴⁰ Synaptic, es un programa informático con una interfaz gráfica GTK de APT, para el Sistema de Gestión de Paquetes de Debían GNU/Linux.

⁴¹ Adept, se encarga de descargar e instalar paquetes.

⁴² KDE, es un entorno de escritorio gráfico, y una colección de programas, tecnologías y documentación que intenta facilitar el uso de los ordenadores a sus usuarios y se utiliza para estaciones de trabajo UNIX.

- **X.Org** 6.8.2 con soporte de hardware ampliado
 - Instalación de aplicaciones mejorada integrada al menú
 - Soporte de multilinguaje mejorado
- Para el Servidor:
 - **Plone**⁴³ 2.1 y **Zope**⁴⁴ 2.8.1
 - **PHP5**
 - Soporte para la instalación directa en volúmenes
 - Funcionalidad thin client integrado
 - Soporte para hasta 4 gigabytes de RAM en arquitectura de 32 bits
 - Soporte en el **Kernel** para sistemas de ficheros
- Soporte de Hardware:
 - **Kernel** 2.6.12.6 con drivers actualizados de terceros
 - Mejoras para el soporte de portátiles (teclas predefinidas, soporte suspensión/recuperación en más modelos).
 - Soporte para cambios escáner/impresoras de HP integrado.
 - Soporte para el dispositivo de entrada Bluetooth (teclados, ratones, etc.)
 - **Kernel** de 64 bits para Power PC.
- Instalación para el Fabricante Original de Equipo (“*OEM*”), o “*Original Equipment Manufacturer*”, es decir que se puede preinstalar Ubuntu en ordenadores Nuevos
 - **KDE** -3.5.2
 - **Firefox** 1.0.7
 - Un Bugtrack integrado (se utiliza para reportar errores a los desarrolladores del sistema)
 - Soporte de clientes ligeros para arrancar en red
- **Ubuntu 6.06 – Dapper Drake (Pato Pulcro)**

⁴³ Plone, es un Sistema de Gestión de Contenidos es código abierto, puede utilizarse como servidor intranet o extranet.

⁴⁴ Zope, es un servidor de aplicaciones Web escrito en el lenguaje de programación Python.

Según [TUXa], Dapper se convirtió en la primera publicación que no cumplió los 6 meses de rigor, debido a que Mark Shuttleworth dio tiempo adicional de desarrollo. Fue también la primera publicación en añadir el *LTS*, a Ubuntu. El Live CD derivó en dos. Los usuarios del Live CD (ahora llamado Desktop CD) podían ahora instalar Ubuntu a sus discos duros, sin embargo con DAppero sólo lo podían hacer a un dispositivo USB.

Otras mejoras fueron disminuyendo el tiempo de carga del sistema, un apagado gráfico, una nueva herramienta de actualización y una mejor reproducción de vídeo. Con el nuevo Network Manager se pudo montar la red y se hizo las cosas más fáciles. OpenOffice 2.0.2 pasaría a instalarse por defecto. Además de todo esto crece la familia una vez más. Además del uso de *Gnome* o *KDE*, *Xubuntu*⁴⁵ usa el escritorio *Xfce*⁴⁶, a más de eso es ligero en el uso de recursos, *Xubuntu* y es perfecto para antiguos equipos.

Las principales funciones de acuerdo a [FF07] son:

- *Gnome* - 2.14.1
- Linux *Kernel* 2.6.15
- *X.Org* 7.0
- *Firefox* 1.5.0.3
- OpenOffice 2.0.2
- *KDE* -3.5.4
- *Xfce* – 4.3.90.1
- Tipo de Instalación: Grafico

➤ **Ubuntu 6.10 – Edgy Eft (Tritón Impaciente)**

Según [TUXa], Que con un tiempo de carga y cierre de sistema más rápido y la integración de TomBoy y F-Spot, Ubuntu Edgy se convirtió en un paso intermedio que enlazaba Dapper con la inminente Feisty FAwn 7.04. El material grafico en Ubuntu se convirtió más ligero y agradable. *Kubuntu*

⁴⁵ Xubuntu, es una rama oficial derivada del Sistema Operativo de código abierto Ubuntu, utiliza el entorno de escritorio Xfce.

⁴⁶ Xfce, es un entorno de escritorio ligero para sistemas tipo Unix como Linux, BSD, Solaris y derivados, provee el marco de trabajo para el desarrollo de aplicaciones.

añadió un software de gestión de fotos, digiKam; se rediseñó el panel de configuración de sistema; y mejoró los botones y el soporte de control de batería de los portátiles.

Las principales funciones de acuerdo a [FF07] son:

- **Gnome** - 2.16
- Linux **Kernel** 2.6.17
- **X.Org** 7.1
- **Firefox** 2.0
- OpenOffice 2.0.4
- **KDE** -3.5.5
- **Xfce** – 4.3.99 RC1
- Tipo de Instalación: Grafico

➤ **Ubuntu 7.04 – Feisty Fawn (Cervatillo Luchador)**

Según [TUXa], Posee nuevas características como el asistente de migración para datos personales desde el sistema operativo Windows, tiene un instalador automático de *codecs*⁴⁷, que se utiliza cuando se intenta reproducir un contenido de multimedia, la Incorporación de *Compiz*⁴⁸ para los efectos de escritorio, pero aún no esta integrado completamente, la incorporación de *Tracker*⁴⁹ para búsquedas en escritorio, posee drivers binarios y también facilita configuración de redes.

Las principales funciones de acuerdo a [FF07] son:

- **Gnome** - 2.18
- Linux **Kernel** 2.6.20.13, que incluye soporte nativo.
- **X.Org** 7.2
- **Firefox** 2.0.0.23
- OpenOffice 2.2.0 RC3

⁴⁷ Códec, es una abreviatura de Codificador - Decodificador, describe una especificación desarrollada en software, hardware o una combinación de ambos.

⁴⁸ Compiz, es uno de los primeros gestores de composición para el sistema de ventanas X.

⁴⁹ Tracker, o rastreador es un servidor especial que contiene la información necesaria para conectar entre peers asistiendo la comunicación entre ellos, a través de protocolos.

- Máquina Virtual basada en el **Kernel**

➤ **Ubuntu 7.10 – Gutsy Gibbon (Gibón Valiente)**

Según [TUXb], manifiesta que al hablar de una versión alpha, es decir es una versión temprana de desarrollo y cual esta sujeta a cambios. Entre algunos de ellos se tiene:

- Más soporte para Winmodems que son modems incompletos en lo que se refiere a hardware
- CD Desktop para que no instale ningún tipo de drive restringido, es decir 100% libre
- **X.Org 7.3**
- Puede haber alguna aplicación u opción para indicar al sistema que elimine los **Kernels** antiguos
- Se realizara algunas mejoras sobre la administración de energía
- Arranque gráfico en modo seguro

Entre otras novedades se tiene:

- Se instalará el reproductor gnash por defecto para la reproducción de archivos en flash, pudiéndose no obstante cambiar el códec por el propietario de Adobe
- Soporte para **KDE 4**
- Facilita el desmontaje de los discos USB
- Mejor integración de Bluetooth en **Kubuntu**
- Se implementara una herramienta de migración de datos para usuarios de otros sistemas como Windows XP, 2000, Vista y MacOS X
- Strigi, es el equivalente al buscador local de Google Desktop
- Uso del driver **ntfs-3g**⁵⁰ por defecto para la escritura en particiones de Sistema de archivo de nueva tecnología (“**NTFS**”), o “*New Technology File System*”

2.4.5 UBUNTU

⁵⁰ Ntfs-3g, es un driver estable ce NTFS para Linux, Mac OS X, FreeBSD, BeOS y Haiku con Licencia GNU GPL y código abierto.

De acuerdo a [UBU], Ubuntu es un sistema operativo de código abierto desarrollado en torno al *Kernel* Linux. La filosofía Ubuntu se basa en los siguientes principios: que el software debe ser gratuito, que la gente debe poder usar el software en su lengua materna y debe poder hacerlo independientemente de cualquiera que sean sus limitaciones; además, la gente debe ser libre de personalizar o modificar el software del modo que crea más conveniente.

Por esas razones:

- Ubuntu siempre será gratuito, y no habrá un costo adicional para la edición profesional; y que el desarrollo del trabajo mejor de ellos esté libremente disponible para todos.
- Ubuntu pueda ser usado por el mayor número de personas posible, por lo que emplea las mejores herramientas de traducción y accesibilidad que la comunidad del Software Libre es capaz de ofrecer.
- Ubuntu se publica de manera regular y predecible, por lo que se puede usar la versión estable o probar para poder ayudar a mejorar la versión en desarrollo.
- Ubuntu está totalmente comprometido con los principios de desarrollo del software de código abierto.

Software Libre.- el proyecto Ubuntu está totalmente basado en los principios del desarrollo de software de código abierto, lo que genera a que se use, mejoras y distribuyan software de código abierto.

La filosofía del software libre establece que la gente pueda usar el software de todas las formas que considere útiles, por lo que el código está disponible para poder descargarlo, modificarlo y usarlo. Por tanto, además del hecho de que el software libre normalmente puede conseguirse sin costo alguno, esta libertad también tiene ventajas técnicas: al desarrollar programas y se puede utilizar el trabajo de los demás y construir a partir de éste.

GNU/Linux.- Es un sistema operativo *GNU* que usa como *núcleo* el **Kernel** Linux.

El proyecto *GNU* está unido a la filosofía del software libre, que es el eje central en los proyectos como el de Ubuntu.

El **Kernel** Linux es el corazón del sistema operativo Ubuntu y que nació en 1991 gracias a Linus Torvalds. Es esencialmente un clónico del *núcleo* de UNIX creado de manera independiente.

Hay muchos sistemas operativos distintos basados en GNU/Linux: **Debian**, **Gentoo**⁵¹, RedHat o **Mandriva**⁵² son algunos ejemplos. Ubuntu es otro competidor que lo hace diferente.

Basado en **Debian**, Ubuntu pretende crear una distribución que proporcione un sistema GNU/Linux actualizado y coherente para la informática de escritorio y servidores, por tal razón Ubuntu incluye una cuidadosa selección de los paquetes de **Debian**, y mantiene su poderoso (*Sistema de Gestión de Paquetes*) que permite instalar y desinstalar programas de una forma fácil y limpia, por lo que a reducido, para incluir solo aplicaciones importantes y de alta calidad.

La Calidad de Ubuntu proporciona un entorno robusto y funcional, adecuado tanto para uso doméstico como profesional y por tal motivo es lo que se publica diferentes versiones, permanentemente.

Escritorio.- El escritorio predeterminado en Ubuntu es **GNOME** que es líder como escritorio y como plataforma de desarrollo tanto para UNIX como para GNU/Linux.

Otro escritorio importante es **KDE**, en el cual el proyecto **Kubuntu** ofrece una opción alternativa en el entorno de escritorio **Gnome** en Ubuntu y es por esa razón que se puede instalar y usar el escritorio **KDE** fácilmente en el sistema.

⁵¹ Gentoo, es una distribución GNU/Linux orientada a usuarios con experiencia en Sistemas Operativos y es utilizado como un servidor seguro.

⁵² Mandriva, es una distribución Linux aparecida en 1998, enfocada a principiantes o usuarios medios, poseen entornos gráficos de escritorio y provee sus propias utilidades de configuración gráficas.

El Soporte.- Ubuntu está mantenido por una incipiente comunidad que no para de crecer y el proyecto esta patrocinado por la Compañía Canónica Ltda. Que tiene a los principales desarrolladores de Ubuntu y a la vez ofrece soporte y servicios de consultaría para Ubuntu.

2.4.6 CARACTERISTICAS

- Basada en la distribución **Debian**.
- Disponible en 4 arquitecturas: Intel x86, **AMD64**⁵³, PowerPC, la Arquitectura de Procesador Escalable (“**SPARC**”), o “*Scalable Processor ARChitecture*”.
- Los desarrolladores de Ubuntu se basan en gran medida en el trabajo de las comunidades de **Debian** y **GNOME**.
- Las versiones estables se liberan cada 6 meses y se mantienen actualizadas en materia de seguridad hasta 18 meses después de su lanzamiento.
- La nomenclatura de las versiones no obedece principalmente a una orden de desarrollo, se compone del dígito del año de emisión y del mes en que esto ocurre.
- El entorno de escritorio oficial es **Gnome**.
- Para solucionar los bugs, conflictos de paquetes, etc. se eliminó algunos paquetes del componente main, que no son populares, por tal razón inicialmente **KDE** no se encontraba con más soporte de lo que entregaban los mantenedores de **Debian** en sus repositorios, por lo que se sumó la comunidad de **KDE** distribuyendo una distribución denominada **Kubuntu**.
- En la versión 6.06 de Ubuntu, apareció por primera vez la distribución **Xubuntu**, basándose en el entorno de escritorio **Xfce**.
- El navegador Web oficial es **Mozilla Firefox**.
- El sistema incluye funciones avanzadas de seguridad como no activar sus políticas de forma predeterminada, procesos latentes al momento de instalarse.

⁵³ AMD64, es la plataforma de 64 bits de AMD que extiende la arquitectura del conjunto de instrucciones x86 estándar en el mercado.

- Para labores/tareas administrativas en terminales incluye una herramienta llamada sudo (similar al Mac OS X).
- Mejorar la accesibilidad y la internacionalización, de modo que el software esté disponible para tanta gente como sea posible.
- Todos los lanzamientos de Ubuntu se proporcionan sin costo alguno.
- Ubuntu no cobra honorarios por la suscripción de las mejoras de la “Edición Enterprise”.

IMPLANTACIÓN DEL SISTEMA

3 IMPLANTACIÓN DEL SISTEMA

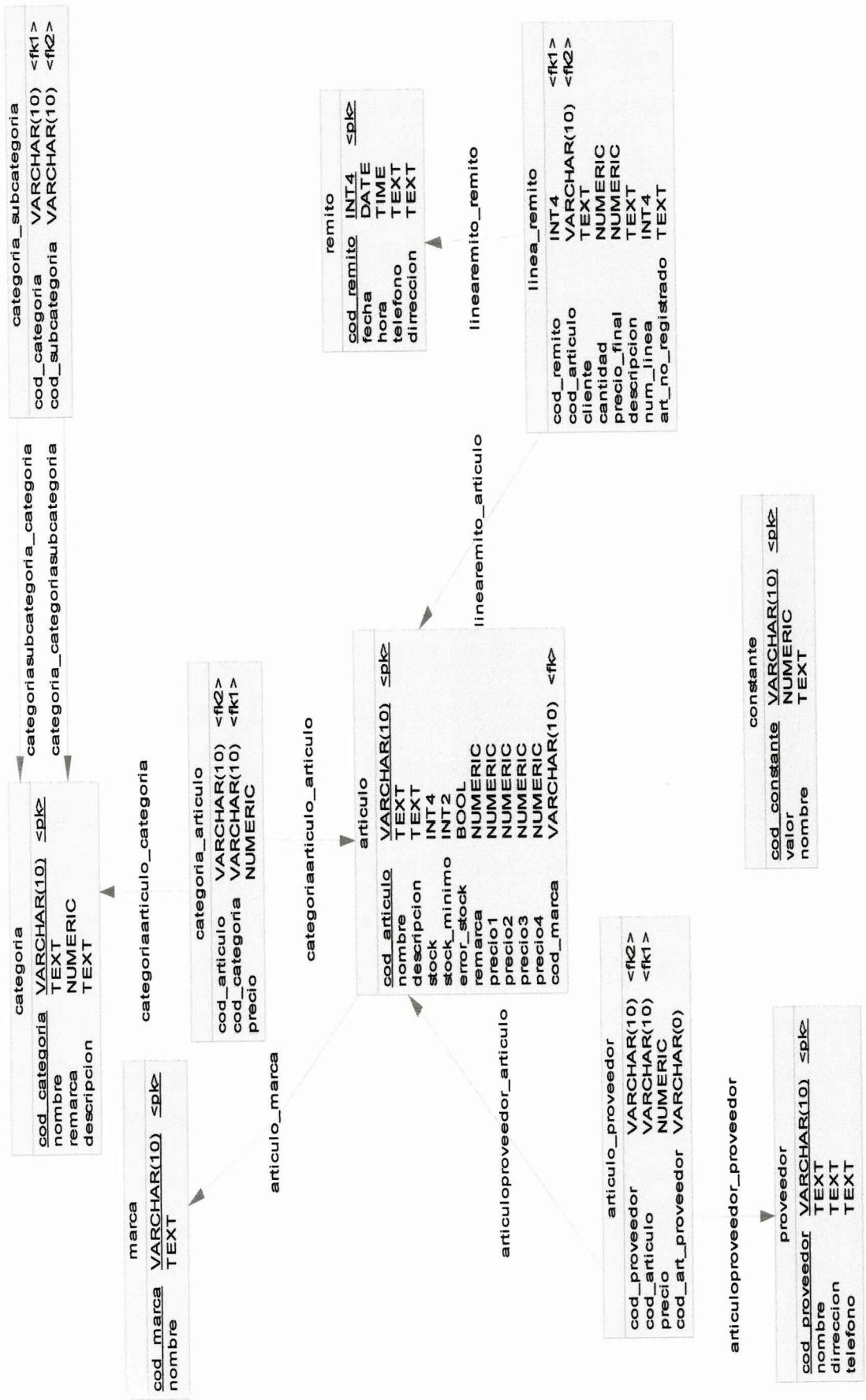
3.1 INTRODUCCIÓN

El objetivo fue crear un Sistema de Gestión de Comercios Basados en Arquitectura Web.

El cual fue diseñado de manera flexible para que se adapte con facilidad a las necesidades que requiere la ferretería, el cual va a permitir acceder a un control completo de la mercadería como son: ingreso, actualización, modificación, eliminación de datos, dentro de las tablas de marca, proveedor, artículo, categorías, y a la vez también para la tabla de las constantes, y para llevar un control de los clientes se procede a la creación de la tabla de los remitos el cual es un comprobante de compra que permite llevar un control de la salida de la mercadería y de los diversos clientes.

Para facilitar el trabajo de los empleados en la ferretería se lo puede realizar a través de búsquedas el cual admitirá realizar diversas consultas como el de la mercadería existente de forma general o individual ya sea a través de código o nombre, ya que la finalidad es facilitar el trabajo y optimizar el tiempo.

3.2 DIAGRAMA DEL MODELO FISICO DE LA BASE DE DATOS



Las Entidades más fuertes de esta Base de Datos que se utilizó para este trabajo son las siguientes: Marca, Proveedor, Artículo, Categoría, Remito.

A continuación se describen cada uno ellos:

➤ **SUB SISTEMA DE MARCA**

Objetivo: Llevar un control de Marcas

Este Sub Sistema permite controlar el ingreso de las diferentes marcas por separado de los sinnúmeros de artículos que contenga el sistema y a la vez permite ingresar, actualizar, modificar los datos que se ingresen en los diferentes campos de la tabla marca.

➤ **SUB SISTEMA DE ARTICULOS**

Objetivo: Cargar el ingreso de los diversos Artículos en el Sistema

Este Sub Sistema permite realizar altas para el ingreso de los diversos artículos permitiendo actualizar, modificar y borrar los datos que se ingresen y llevar el control de Stock, remarcas y precios el cual se van acumulando en los diferentes campos, y a la vez se encuentran relacionados con las tablas marca, proveedor y categoría.

➤ **SUB SISTEMA DE PROVEEDOR**

Objetivo: Permitir clasificar según el proveedor que provee

Este Sub Sistema permite realizar altas de los proveedores que proveen los diversos artículos, y a la vez permite borrar, actualizar, modificar los diferentes campos y permite el ingreso de los datos personales del proveedor el cual se van almacenando en los diferentes campos y está relacionado con la tabla artículos para una mejor distribución.

➤ **SUB SISTEMA DE CATEGORIA**

Objetivo: Ordenar los artículos en diferentes categorías

Este Sub Sistema permite realizar altas de las categorías que se las utiliza en función cambiante de las categorías su función es similares a la de los grafos, y a la vez permite borrar, actualizar, modificar los diferentes campos, permitiendo el ingreso de la remarca y la descripción de cada categoría ingresada y para una mayor flexibilidad están relacionados con la tabla artículos.

➤ **SUB SISTEMA DE REMITO**

Objetivo: Permitir controlar la salida de Mercadería

Este Sub Sistema permite realizar el ingreso de los de los datos de los clientes y además permite el ingreso de datos como fecha, hora, teléfono, dirección, pero este sistema no tiene control de clientes ya que permite borrar, actualizar, modificar los diferentes campos y a través de otra tabla permite llevar el control de la cantidad y el precio de los diferentes artículos que realizan y además se relacionan con la tabla artículos.

3.3 FUNCIONES Y PROCEDIMIENTO

- **CONEXIÓN DE LAS BASES DE DATOS.-** Permite conectarse a la base de datos llamada ferretería realiza en PostgreSQL, en el cual almacenaran un conjunto de datos en cada una de las tablas.

```
[DB_DataObject]
```

```
error          = "<br>Este archivo no puede ser visualizado<?php exit; ?>"
database       = pgsq://edwin:edwin@localhost/ferreteria
schema_location = ".\DAOs"
class_location  = ".\DAOs"
require_prefix  = DataObjects/
class_prefix    = DataObjects_
generator_strip_schema = 1
```

- **Dar de Alta un Artículo.-** Este código nos permite crear los text, y los label para los diversos campos que contiene la tabla Artículos el cual va permitir

realizar el ingreso de los cada uno de ellos que se ingrese y los cuales se irán almacenando en la base de datos y a través de un command permita guardar los datos ingresados.

```

<?php
require_once('PageGen.php');
require_once('daolister.php');
require_once('daoForm.php');
session_start();

//*****
*****

function inic(&$daof){
    $daof=new DaoForm('articulo');
    $daof->setRenderDaoName('Artículo');
    $daof->setRenderNames(array(cod_articulo =>'Código Artículo',
                                cod_marca =>'Código Marca',
                                nombre =>'Nombre',
                                descripcion =>'Descripción',
                                stock =>'Stock',
                                stock_minimo =>'Stock Minimo',
                                error_stock =>'Error Stock',
                                remarca =>'Remarca',
                                precio1 =>'Precio 1',
                                precio2 =>'Precio 2',
                                precio3 =>'Precio 3',
                                precio4 =>'Precio 4'));

    $daof->setInputs(array(descripcion =>'textarea'));
    $_SESSION['daofoo']=$daof;

}

//*****
*****

function extraCombos(){
    openDB();
    $cat=DB_DataObject::factory('categoria');
    $arr=$cat->link('cod_categoria');
    $html.='<tr><td class="izq">Categoría:</td><td
class="der"><select name =          "cod_categoria"
id="cod_categoria">';
    foreach ($arr as $id =>$valor){
        $html.='<option value="".'$id.'">('.$id.')'.'$valor.'</option>';
    }
    $html.='</select></td></tr>';
    return $html;
}

//*****
*****

    inic($daof);
    if ($_POST['action']!= 'guardar'){
        if ($_GET['mod']!= 'true')
            $daof->extraRows(extraCombos());
        $html.='<BR/><BR/><center>'. $daof-
getForm($_GET['ids']).</center>';
    }else if ($_POST['action']!= 'guardar'){
        if ($_GET['mod']!= 'true'){
            $aux1=$daof->getForm($_GET['ids']);

```

```

$aux2=$daof->save();
$aux1=$daof->getForm($_GET['ids']);
$html.='<BR/><BR/><center>'.$aux2.$aux1.'</center>';
}else{
$guardar=$daof->validatePOST();
$aux1=$daof->getForm($_GET['ids']);
$aux2=$daof->save();
if ($guardar==""){
    openDB();
    $ca=DB_DataObject::factory('categoria_articulo');
    $ca->cod_categoria=$_POST['cod_categoria'];
    $ca->cod_articulo=$_POST['cod_articulo'];
    $ca->precio=0;
    if (!$ca->insert())
        $mensajes='<li>Error en la insercion de la Categoría
        '.$_POST['cod_categoria'].'</li>';
    }
    if (mensajes!="") $mensajes='<ul>'.$mensajes.'</ul>';

$html.='<BR/><BR/><center>'.$aux2.$mensajes.$aux1.'</center>';
}
}

echo Page('style.css','js.js','Alta/Modificación Artículo',$html);
?>

```

- **Para Búsqueda.-** Este código me permite realizar búsqueda para cada una de las entidades ya sea a través de código o nombre y a la vez permite validar cada una de las variables que se va ingresar el cual controla ya sea números o letras para poder obtener buenos resultados.

```

<?php
require_once('PageGen.php');
require_once('daolister.php');
require_once('calendario.php');
session_start();
$html.='<br/><br/><center><form action=".'.$_PHP['self'].'" name=
"frmbuscar" id="frmbuscar" method="post" enctype="multipart/form-data">
<input type="hidden" name="action" id="action" value="submit">
<table>
<tr>
<th colspan="6">Buscar</th>
</tr>
<tr>
<td class="izq">Sobre:</td>
<td class="der"><select id="buscaDAO" name="buscaDAO"
onchange="javascript:flipFecha('\buscaDAO\')">
<option value="remito">Remito</option>
<option value="marca">Marca</option>
<option value="articulo">Artículo</option>
<option value="proveedor">Proveedor</option>
<option value="categoria">Categoría</option>
<option value="constante">Constante</option>
</select>
</td>

```

```

<td class="izq">Código:</td><td class="der"><input
type="text" name="codigo"></td>
<td class="izq" >Nombre:</td><td class="der"><input
type="text" name="nombre"></td>
</tr>
<tr id="Op" name="Op">
<td class="der" colspan="2"></td>
<td class="izq">Fecha Comienzo</td><td
class="der"><input type="text" name="fecha" id="fecha"
READONLY></td>
<td class="izq">Fecha Fin</td><td class="der"><input
type="text" name="fecha1" id="fecha1" READONLY></td>
</tr>
<tr id="Op" name="Op">
<td class="der" colspan="2"></td>
<td class="izq" colspan="2"><div id="com"
align="center">'.mostrar_calendario(1,7,2007,'fecha','com').'<
/div></td>
<td class="izq" colspan="2"><div id="fin"
align="center">'.mostrar_calendario(1,7,2007,'fecha1','fin').'<
/div></td>
</tr>
</table><BR/><input type="submit" name="sub" id="sub" value="Buscar">
</form></center>';
if ($_GET['action']=='del'){
    openDB();
    $auxDao=new DaoForm($_SESSION['buscaDAO']);
    $men=$auxDao->delete($_GET['ids']);
    $daoL=new DaoLister($_SESSION['buscaDAO']);
    $daoL->buscar($_SESSION['buscar']);
    $daoL->rowsxPage(100);
    $daoL->reCount();
    $lister='<br/><br/><center>'. $daoL->getPagina(0).</center>';
    //html.='<BR/><BR/><center>'. $men.$dao-
    >getPagina($_GET['pag']).</center>';
}

if (isset($_POST['action']) && $_POST['action']=='submit'){
    $codigo=getString($_POST['codigo']);
    $nombre=getString($_POST['nombre']);
    $fecha=getString($_POST['fecha']);
    $fecha1=getString($_POST['fecha1']);
    if ($_POST['buscaDAO']!='remito' && $nombre==" && $codigo==" )
        $mensajes.='<li class="errores">Ambos campos esan vacios o
        contienen caracteres no validos</li>';
    else if ($_POST['buscaDAO']=='remito' && $codigo!=" &&
    !validarINT($codigo))
        $mensajes.='<li class="errores">El campo Código para la entidad
        Remito debe ser numerico</li>';
    if ($mensajes==""){
        openDB();
        $daoL=new DaoLister($_POST['buscaDAO']);
        $daoL->rowsxPage(100);
        $_SESSION['buscaDAO']= $_POST['buscaDAO'];
        if ($_POST['buscaDAO']=='remito'){
            $daoL->buscar('cod_'.$_POST['buscaDAO'].' like
            \'%'.$codigo.%\' and fecha >=\''.$fecha.%\' and fecha
            <\''.$fecha1.%\'');
        }
    }
}

```

```

$_SESSION['buscar']='cod_'.$_POST['buscaDAO'].'
like \%'.$codigo. '%\' and fecha >= \''.$fecha. '\'' and
fecha < \''.$fecha1. '\'';
}else{
    $daoL->buscar('nombre like \%'.$nombre. '%\' and
cod_'.$_POST['buscaDAO'].' like \%'.$codigo. '%\');
    $_SESSION['buscar']='nombre like
    \%'.$nombre. '%\' and cod_'.$_POST['buscaDAO'].'
    like \%'.$codigo. '%\';
}
switch ($_POST['buscaDAO']){
    case 'marca':
        $daoL->setAlta('marca.php');
        $daoL->renderFields
        (array('cod_marca','nombre'));
        $daoL->renderHeads(array('Código
Marca','Nombre'));
        break;
    case 'articulo':
        $daoL->setAlta('articulo.php');
        $daoL->renderFields
(array('cod_articulo','cod_marca',
'nombre','descripcion','stock','stock_minimo','error_stock','re
marca','precio1','precio2','precio3','precio4'));
        $daoL->renderHeads(array('Código Artículo','Código
Marca','Nombre','Descripción','Stock','Stock Minimo','Error
Stock','Remarca','Precio 1','Precio 2','Precio 3','Precio 4,));
        break;
    case 'proveedor':
        $daoL->setAlta('proveedor.php');
        $daoL->renderFields
(array('cod_proveedor','nombre','direccion','telefono'));
        $daoL->renderHeads(array('Código
Proveedor','Nombre','Dirección','Teléfono'));
        break;
    case 'categoria':
        $daoL->setAlta('categoria.php');
        $daoL->renderFields (array('cod_categoria','nombre',
'remarca','descripcion'));
        $daoL->renderHeads(array('Código
Categoría','Nombre','Remarca','Descripción'));
        break;
    case 'remito':
        $daoL->setAlta('remito.php');
        $daoL->renderFields
(array('cod_remito','fecha','hora','direccion','
telefono'));
        $daoL->renderHeads(array('Código
Remito','Fecha','Hora','Teléfono','Dirección'
));
        break;
    case 'constante':
        $daoL->setAlta('constante.php');
        $daoL->renderFields(array
('cod_constante','nombre','valor'));
        $daoL->renderHeads (array('Código
Constante','Nombre','Valor'));
        break;
}

```

```

        $lister='<br/><br/><center>'. $daoL-
        >getPagina(0). '</center>';
    }

    echo Page('style.css','js.js','Busqueda',$mesajes.$html.$lister);
?>

```

3.4 APLICACIÓN:

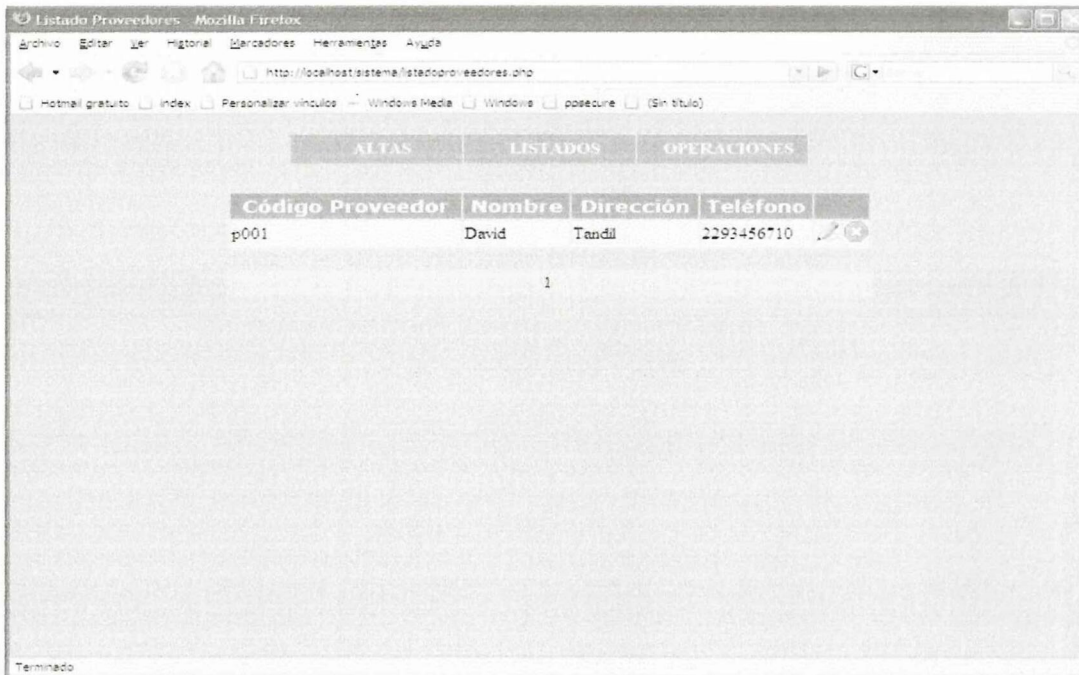
- A través de esta pantalla permite visualizar los campos de la tabla Artículo el cual va a permitir realizar las altas (ingreso) de cada uno de los artículos que se vaya a ingresar y de igual similitud de diseño tiene las otras tablas .

The screenshot shows a web browser window titled 'Alta/Modificación Artículo - Mozilla Firefox'. The address bar shows 'http://localhost/sistema/articulo.php'. The page content is a form titled 'Artículo' with the following fields:

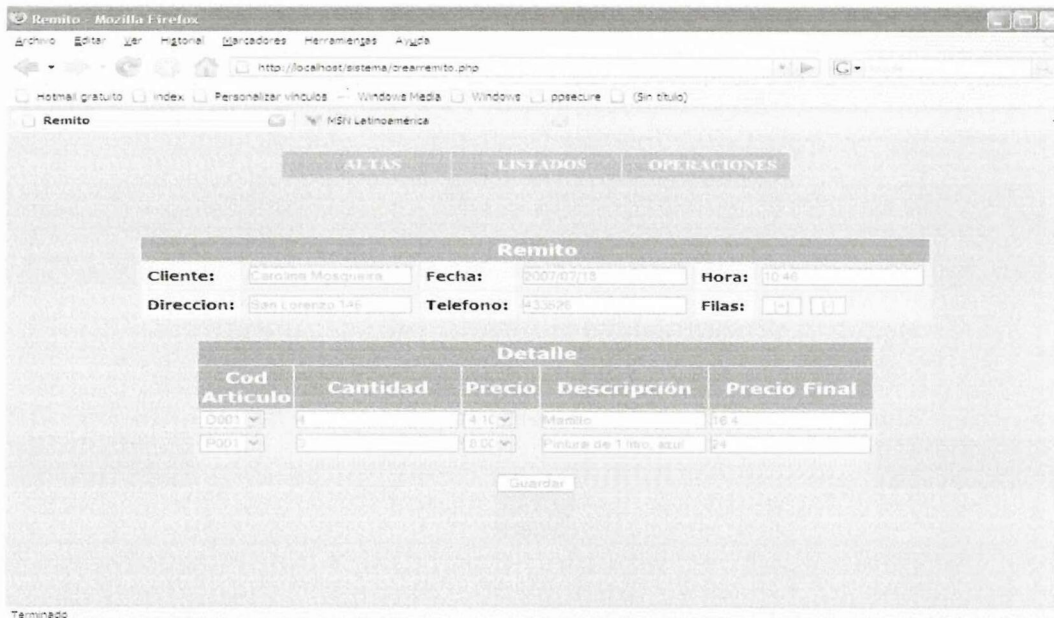
- Código Artículo:
- Nombre:
- Descripción:
- Stock:
- Stock Mínimo:
- Error Stock:
- Remarca:
- Precio 1:
- Precio 2:
- Precio 3:
- Precio 4:
- Código Marca:
- Categoría:

At the bottom right of the form is a 'Guardar' button. The browser status bar at the bottom left shows 'Terminado'.

- En esta pantalla admite visualizar todos los datos que se ha ingresado en cada uno de los campos y a la vez tiene la opción de que se puede seleccionar cada uno de ellos ya sea para poder actualizar o borrar.



- En esta pantalla consiente en crear remitos para cada uno de los clientes y es un comprobante de venta que permite tener el registro de las ventas realizadas de cada uno de los artículos que se a vendido, con el cual permita tener control de stock para poder determinar que articulo se necesita adquirir y poder solicitar a sus respectivo proveedor.



CONCLUSIONES

4.1 CONCLUSIONES

- Todas las herramientas utilizadas en este trabajo son herramientas muy flexibles que permiten dar elasticidad y seguridad al trabajo realizado. Pero con las dificultades que se encontró en el desarrollo de este trabajo no se pudo cumplir con todos objetivos que se planteo para este trabajo, pero se pudo cumplir con el 80% de objetivos propuestos.
- Entre los objetivos fue crear un Sistema de Gestión de Comercios el cual fue diseñado para que se adapte con facilidad en el área comercial específicamente para la Ferretería el Siglo en el cual se implemento, y se utilizo plataformas de software libre como el Sistema Operativo GNU/Linux, el servidor de base de datos PostgreSQL, Servidor Web Apache, y el generador dinámico de paginas PHP.
- El Sistema permitirá tener un control completo de toda la mercadería, mediante la realización de listados, controles para las ventas de la ferretería y a la vez generando mayor seguridad para las personas que vayan a utilizar el cual va a generar dar un mejor servicio a los diversos clientes que adquieran los artículos.
- Se pudo estudiar plataformas de software libre el cual permite a que los usuarios puedan estudiar su código y modificar permitiendo que se realice mejoras de manera que beneficie a las personas o empresas que requieran o

ha toda la comunidad, y a la vez esta permitiendo facilitar los trabajos sobre todo dando mayor seguridad a los diversos usuarios que utilizan este servicio, ya sea a través de metodologías y herramientas el cual a permitido minimizar los esfuerzos de instalación y por otro lado permite dar un mejor servicio a los clientes.

- Todo este trabajo me permitió adquirir nuevos conocimientos y aprender nuevas herramientas para poder realizar nuevos sistemas de manera especial con plataformas de software libre y con arquitectura Web las cuales están revolucionando el mundo cambiante de hoy en día.

TRABAJOS FUTUROS

5.1 TRABAJOS FUTUROS

- En un trabajo posterior para este Sistema se aumentara un modulo el cual admitirá llevar un control mas especifico de cada uno de los clientes.
- Se Creara un Sistema de Facturación el cual generara llevar un mejor control y verificar si se tiene la existencia solicitada por parte de los clientes, para poder generar la salida de ventas en la ferretería realizada, y a su vez admitirá llevar un mejor control de los inventarios.
- Incrementar un modulo que permita leer a través de código de barras los diferentes artículos tanto al momento de ingreso como de salida, permitiendo disminuir tiempo y agilizar procesos de transacción para la venta.
- Fortalecer a la base de datos a través de un mantenimiento permanente y oportuno el cual evitara la pérdida de información almacenada en los diferentes archivos y por otro lado brindar una mejor seguridad.

Bibliografía y Referencias

6.1 BILIOGRAFIA Y REFERENCIAS

- [ABDLOR06] ACHOUR, M., BETZ, F., DOVGAL, A., LOPEZ, N., OLSON, P., & RICHTER, G. (2006). *Manual de PHP*. Descargado el 2007-04-18, disponible en la Web en: <http://www.php.net/manual/es/history.php>
- [ANT] Descargado el 2007-06-19, disponible en la Web en: <http://www.ant.org.ar/revista/num0/fsfla.pdf>
- [APAC] Descargado el 2007-06-06, disponible en la Web en: <http://www.apache.org>
- [BH01] BROOKINS, A., & HOLLOWAY, M. (2001). Proyectos S.O.B.L.: *PostgreSQL Práctico*. Descargado el 2007-06-04, disponible en la Web en: <http://www.sobl.org/traduccion/practical-postgres/node19.html>
- [BP06] BROWN, P., (2006). La Revista de Linux más Completa: *Ubuntu 5.10*. Descargado el 2007-06-14, disponible en la Web en: <https://www.linux-magazine.es/issue/13>
- [CA00] CASTILLO, A. (2000). *El servidor de Web de Apache*. Descargado el 2007-06-08, disponible en la Web en: <http://idefix.eup.uva.es/Manuales/Servidor%20Web/apache.pdf>

- [CA06] CASTILLO, A. (2006). Webs dinámicos con PHP: *Programar en PHP*. Descargado el 2007-05-02, disponible en la Web en: <http://www.programacion.net/php/tutorial/php4/3/>
- [CR04] CARVAJAL, R. (2004). *Introducción a PHP*. Descargado el 2007-05-06, disponible en la Web en: http://roberto.ento.daspartes.org/tutoriales/php_intro.html
- [DNV] Descargado el 2007-06-26, disponible en la Web en: <http://www.dnv.es/certificacion/sistemasdegestion/index.asp>
- [EN97] ELMASRI, R., NAVATHE, S. (1997). *Sistema de Base de Datos* (2da. ed.). Estados Unidos de América: Addison-Wesley Iberoamericana.
- [FF07] FLORES, F., (2007). *Ubuntu y la Historia de sus Versiones*. Descargado el 2007-06-14, disponible en la Web en: http://fs-webdesign.com/blog/?page_id=226
- [GA07] GARCIA, A., (2007). El Sistema Operativo GNU: *Que es la Fundación para de Software Libre*. Descargado el 2007-06-08, disponible en la Web en: <http://www.gnu.org/home.es.html>
- [GF05a] GIROLETI, F., (2005). *Objetivos de la Fundación del Software Libre de América Latina*. Descargado el 2007-06-18, disponible en la Web en: <http://www.gnu.org/philosophy/categories.es.html#Non-CopyleftedFreeSoftware>
- [GF05b] GIROLETI, F., (2005). *Objetivos de la Fundación del Software Libre de América Latina*. Descargado el 2007-06-18, disponible en la Web en: <http://pulsar.unizar.es/gluz/manual-sl/index.html>

- [HA04] HERRERA, A. (2004). *Introducción a PostgreSQL*. Descargado el 2007-05-30, disponible en la Web en: <http://alvherre.atentus.cl/charlas/10-Introduccion.pdf>
- [HC01] HENST, C. (2001). *Introducción al PHP*. Descargado el 2007-05-06, disponible en la Web en: <http://www.maestrosdelweb.com/editorial/phpintro/>
- [HOSP] Descargado el 2007-06-11, disponible en la Web en: http://www.hospedajeydominios.com/mambo/documentacion-manual_apache.html
- [HPGPTB04] HOJTSY, G., PARDO, A., GARCIA, E., PEREZ, J., TACÓN, J., & BOSHELL, L. (2004). *Manual de PHP*. Descargado el 2007-04-20, disponible en la Web en: <http://www.php.net/manual/es/introduction.php>
- [HR07] HINOSTROZA, R. (2007). Usuarios y Grupos en Linux: *Características de PHP*. Descargado el 2007-04-29, disponible en la Web en: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>
- [LC07] LORCA, C. (2007). Artículos MMUG: *Que es PostgreSQL*. Descargado el 2007-05-30, disponible en la Web en: <http://www.mmug.cl/articulos.php?id=293&tod=1>
- [LT96] LOCKHART, T. (1996). Tutorial de Postgresql: *El equipo de desarrollo de PostgreSQL*. Descargado el 2007-05-20, disponible en la Web en: <http://palomo.usach.cl/Docs/postgres/Postgres-Tutorial.pdf>
- [MB06] MOMJIAM, B., (2006). *Kit de Prensa de PostgreSQL 8.2*. Descargado el 2007-06-04, disponible en la Web en: <http://www.postgresql.org/about/press/presskit82.html.es>

- [MC04] MATEU, C. (2004). *Software Libre: Desarrollo de Aplicaciones Web*. Descargado el 2007-06-07, disponible en la Web en: <http://www.uoc.edu/masters/esp/img/873.pdf>
- [PH04] PEREZ, P., & HERNANDO, D., (2004). *Historia del Sistema GNU/Linux*. Descargado el 2007-06-13, disponible en la Web en: <http://www.alu.ua.es/p/psp4/historia/index.html>
- [PM06] PEREZ, M., (2006). *Definición de Software Libre*. Descargado el 2007-06-15, disponible en la Web en: <http://www.gnu.org/philosophy/free-sw.es.html>
- [PM07] PEREZ, M., (2007). *Licencias*. Descargado el 2007-06-15, disponible en la Web en: <http://www.gnu.org/licenses/licenses.es.html#GPL>
- [POST] Descargado el 2007-04-20, disponible en la Web en: <http://www.postgresql.com>
- [QL06] QUEZADA, L., (2006). *La Historia de Ubuntu*. Descargado el 2007-06-14, disponible en la Web en: <http://lfqm1987.wordpress.com/2006/09/02/la-historia-de-ubuntu/>
- [RC07] OVNER, C., (2004). *Historia y Cultura del Software Libre: Historia del Software Libre*. Descargado el 2007-06-20, disponible en la Web en: [http://www.gnu.org/philosophy/categories.es.html# Non-CopyleftedFreeSoftware](http://www.gnu.org/philosophy/categories.es.html#Non-CopyleftedFreeSoftware)
- [RD05] RANDRADE, D. (2005). *Que es PostgreSQL*. Descargado el 2007-05-28, disponible en la Web en: <http://www.postgresql.org.mx/?q=node/6>.

- [RJ06] RIVERA, J. (2006). *Introducción al servidor Web apache*. Descargado el 2007-06-07, disponible en la Web en: <http://www.juliux.org/asw1/presentaciones/PDF/presentacion3.pdf>
- [RS07] RICHARD, S., (2007). *Linux y el Proyecto GNU*. Descargado el 2007-06-13, disponible en la Web en: <http://www.gnu.org/gnu/linux-and-gnu.es.html>
- [RX04] REINA, X., (2004). *Historia del Proyecto GNU*. Descargado el 2007-06-12, disponible en la Web en: <http://www.gnu.org/gnu/gnu-history.es.html>
- [RX06] REINA, X., (2006). Categorías de Software Libre y no Libre: *El Sistema GNU*. Descargado el 2007-06-13, disponible en la Web en: <http://www.gnu.org/philosophy/categories.es.html#TheGNUsystem>
- [TE06] TAVIERA, E. (2006). *Preguntas Frecuentes (FAQ) sobre PostgreSQL*. Descargado el 2007-06-04, disponible en la Web en: http://www.postgresql.org/docs/faqs.FAQ_brazilian.html
- [TUF] *Los orígenes de Ubuntu*. Descargado el 2007-06-13, disponible en la Web en: <http://tufuncion.com/historia-ubuntu>
- [TUXa] *Historia de Ubuntu y sus versiones*. Descargado el 2007-06-14, disponible en la Web en: <http://tuxpepino.wordpress.com/2007/04/17/historia-de-ubuntu-y-sus-versiones/>
- [TUXb] *Publicado Ubuntu 7.10*. Descargado el 2007-06-14, disponible en la Web en: <http://tuxpepino.wordpress.com/2007/06/08/publicado-ubuntu-710-gutsy-gibon-tribe-1/#more-210>

[UBU]

Ubuntu: *GNU/Linux para gente normal.*

Descargado el 2007-06-15, disponible en la Web en:
<http://www.ubuntu-es.org/index.php?q=ubuntu/introduccion>

GLOSARIO

G.1 GLOSARIO

ANSI.- American National Standards Institute (*Instituto Nacional Estadounidense de Estándares*)

API.- Application Program Interface (*Interfaz de Aplicación del Programa*)

APR.- Apache Portable RunTime (*Apache Portable en Tiempo de Ejecución*)

APT.- Advanced Packaging Tool (*Herramienta Avanzada de Empaquetamiento*)

ASP.- Active Server Pages (*Páginas de Servidor Activas*)

BASH.- Bourne Again Shell (*Renacimiento Sobre el Bourne Shell*)

BSD.- Berkeley Software Distribution (*Distribución de Software Berkeley*)

CGI.- Common Gateway Interface (*Interfaz de Salida Común*)

CMS. - Content Management System (*Sistema de Gestión de Contenido*)

DBM.- DataBase Management (*Administración de Base de Datos*)

DBMS.- DataBase Management System (*Sistema de Gestión de Bases de Datos*)

DBX.- DataBase Abstraction (*Abstracción de Base de Datos*)

EMAS.- Eco-Management and Audit System (*Sistema Comunitario de Ecogestión y Ecoauditoría*)

ERP.- Enterprise Resource Planning (*Planificación de Recursos Empresariales*)

FSF. - Free Software Foundation (*Fundación de Software Libre*)

GNU o GLP.- General Public License (*Licencia Pública General*)

HTML.- HiperText Markup Lenguaje (*Lenguaje de Marcación de Hipertexto*)

HTTP.- HyperText Transport Protocol (*Protocolo de Transferencia de Hipertexto*)

IEC.- International Electrotechnical Commission (*Comisión Electrotécnica Internacional*)

IMAP.- Internet Message Access Protocol (*Protocolo de Acceso a los Mensajes de Internet*)

IP.- Internet Protocol (*Protocolo de Internet*)

ISO.- International Organization for Standardization (*Organización Internacional para la Estandarización*)

ITS.- Incompatible Timesharing System (*Sistema Incompatible de Tiempo Compartido*)

JRE.- Java Runtime Environment (*Entorno en Tiempo de Ejecución Java*)

JSP.- Java Server Pages (*Páginas de Servidor Java*)

LDAP.- Lightweight Directory Access Protocol (*Protocolo de Acceso a Directorios Ligeros*)

LTS.- Long Term Support (*Largo Periodo de Soporte*)

MIT.- Massachusetts Institute of Technology (*Instituto Tecnológico de Massachusetts*)

MPM.- Multiprocessing Modules (*Módulos de Multiprocesamiento*)

MVCC.- Multi-Version Concurrency Control (*Control de Concurrency Multi - Versión*)

NCSA.- National Center for Supercomputing Applications (*Centro Nacional de Aplicaciones de Súper Computación*)

NNTP.- Network News Transport Protocol (*Protocolo de Transferencia de Noticias*)

NTFS.- New Technology File System (*Sistema de Archivo de Nueva Tecnología*)

ODBC.- Open Data Base Connectivity (*Conexión Abierta de Base de Datos*)

OEM.- Original Equipment Manufacturer (*Fabricante Original de Equipo*)

OHSAS.- Occupational Health and Safety Management Systems (*Sistema de Gestión de Seguridad y Salud Laboral*)

OLTP.- OnLine Transaction Processing (*Procesamiento en Línea de Transacciones*)

ORDBMS.- Object-Relational Database Management System (*Sistema de Gestión de Bases de Datos Objeto-Relacionales*)

OSI.- Open Source Initiative (*Iniciativa de Código Abierto*)

PDF.- Portable Document Format (*Formato de Documento Portáble*)

PDP-10.- Programmed Data Processor Model 10 (*Procesador de Dato Programado modelo10*)

PERL.- Practical Extraction and Report Language (*Lenguaje Práctico de Extracción y de Informes*)

PHP.- HyperText Preprocessor (*Preprocesador de Hipertexto*)

POP3.- Post Office Protocol Version 3 (*Protocolo de Oficina de Correos Versión 3*)

POSIX.- Portable Operating System Interface, Unix Based (*Interfaz de Sistema Operativo Portable Basado en UNIX*)

RBC.-Risk Based Certification (*Certificación de Base de Riesgo*)

RDBMS.- Relational Data Base Manager System (*Sistema Administrador de Bases de Datos Relacionales*)

SGBD, o DBMS.- Database Management System (*Sistemas de Gestión de Base de Datos*)

SNMP.- Simple Network Management Protocol (*Protocolo Simple de Gestión de Red*)

SPARC.- Scalable Processor ARChitecture (*Arquitectura de Procesador Escalable*)

SQL.- Structured Query Language (*Lenguaje de Consulta Estructurado*)

SSI.- Server Side Include (*Incluir Lado - Servidor*)

TCL.- Tool Command Language (*Lenguaje de Comando Herramienta*)

URL.-Uniform Resource Locutor (*Identificador de Recursos Uniforme*)

WDDX.- Web Distributed Data Exchange (*Intercambio de Datos Distribuido Web*)

WML.- Wireless Markup Language (*Lenguaje de Marcación Inalámbrico*)

XML.- Extensible Markup Lenguaje (*Lenguaje de Marcación Extensible*)

APÉNDICES

A.1 CERTIFICACIÓN PARA LOS SISTEMAS DE GESTIÓN

La Certificación de Base de Riesgo (“RBC”), o “*Risk Based Certification*”, tiene como misión ayudar a las compañías a conseguir sus objetivos de negocio relacionando objetivos estratégicos con procesos operacionales, evaluando áreas de riesgo e identificando oportunidades de mejora y cumplir con las diferentes normativas internacionales tales como:

- La certificación del Sistema de Gestión de la Calidad conforme a la norma de Organización Internacional para la Estandarización (“ISO”), o “*International Organization for Standardization*”, 9001 y otras normas de calidad específicas para cada industria, están encaminadas en mejorar el negocio lo cual permite descubrir los puntos fuertes, oportunidades de mejora y su compromiso con la calidad y la satisfacción del cliente.
- La certificación del Sistema de Gestión Medio Ambiente conforme a la norma ISO 14001 y el reglamento Sistema Comunitario de Ecogestión y Ecoauditoría (“EMAS”), o “*Eco-Management and Audit System*”. que se mantenga los procesos clave y los objetivos bajo control y que se demuestre su compromiso con operaciones para proteger el medio ambiente.
- La certificación conforme a la norma de Sistema de Gestión de Seguridad y Salud Laboral (“OHSAS”), o “*Occupational Health and Safety Management Systems*” 18001 el cuál permite concentrarse en los aspectos críticos y la gestión de riesgos para salvaguardar la salud y seguridad de los empleados y el medio ambiente.

- La certificación del Sistema de Gestión Seguridad Informática conforme a la norma *ISO/ Comisión Electrotécnica Internacional (“IEC”)*, o “*International Electrotechnical Commission*” 27001 que permite que se mantenga los procesos críticos bajo control, protegiendo la información de su empresa y reduciendo las filtraciones.
- La certificación del Sistema de Gestión de Comercio Electrónico conforme a Eitrust, a fin de transmitir la confianza y la seguridad de que su página Web es un lugar seguro para comprar.
- Certificación Integrada de dos o más sistemas de gestión, con el objeto de beneficiarse de un enfoque integral por el cual se certificará a todos los sistemas de gestión de una sola vez.

A.2 PGADMIN III

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Ya que permite gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración, y en la aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad.