



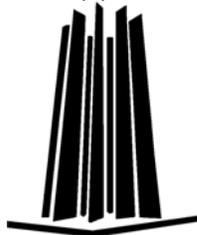
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN

**“INFORME DEL PROYECTO DE PRUEBAS
PSICOLÓGICAS A DISTANCIA CON SOFTWARE
LIBRE”**

T R A B A J O E S C R I T O
EN LA MODALIDAD DE SEMINARIOS
Y CURSOS DE ACTUALIZACIÓN Y
CAPACITACIÓN PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
O R T I Z C O R D E R O
G A B R I E L

ASESOR: M. EN C. MARCELO PEREZ MEDEL





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Agradecimientos

A Ma. de la Luz Cordero, mi madre, por haberme apoyado en cada momento, siempre confiando en que lograría alcanzar esta meta tan añorada.

A mis hermanas Claudia y Mariana por estar siempre a mi lado en las buenas y en las malas.

A Marila quien me enseñó todo para poder ser hoy una persona de bien en todos los aspectos.

A Armando Arcos por siempre esperarnos con los brazos abiertos allá donde todo es mucho más tranquilo.

A Fernando que supo estar ahí, a pesar de las circunstancias y la distancia que nos separa.

A Victoria Juárez por siempre apoyarme durante estos cinco años y más de carrera ininterrumpida.

A Fernando Morales por su apoyo en otros temas necesarios en esta larga vida.

A Marcos Maceda quien desde niño me apoyó e instruyó en todos los aspectos.

A todos mis amigos de la secundaria, los Licenciados en Derecho Diana y Arturo, la Química Ma. de la Luz, los Ingenieros Pablo y Sergio, la Licenciada en Contaduría Susana, la Doctora Isabel, la Licenciada en Administración de Empresas Yezmin a quien todos saben quiero mucho y al ex Marine de la Armada de los E.U Luis Manuel; por continuar todos juntos después de 12 años de conocernos.

A mis inseparables amigos que ahora son como mis hermanos, Ares y Christian, porque juntos hemos vivido extraordinarios momentos a pesar de circunstancias difíciles.

A todos los compañeros del trabajo, Maestro Marcelo por brindarme la oportunidad de trabajar con él mientras me titulaba; además de ser el asesor de este trabajo escrito. Maestro Felipe por su minuciosa y necesaria revisión y su valiosa amistad. A Violeta, Lucero, Mary y Martita, las chicas guapas de la carrera. Abraham por siempre estar en todo. Al resto de las muchachas de otras áreas: Lucero, Mary, Ángeles, Olimpia y al resto que pudiera olvidar. A todos ustedes gracias por su apoyo y amistad.

A todas las personas que he conocido durante toda mi vida: Oscar, Cecy, Cachito, Diana, Adriana, Carlos, Manuel, Bety, Mariana y Fernanda por regalarme su compañía y aprender junto a ellos como vivir la vida.

A mis amigos de la carrera: David, Chucho, Luis, José Luis, Omar, Iván Grada e Iván Nava y Laura con los que descubrí que no es tan sencillo terminar una carrera profesional

Introducción	1
1. Sistema Operativo Linux	2
1.1 Breve Historia del S.O. Linux	2
1.2 Comandos Básicos	3
1.3 Instalación y Administración de Linux	8
1.3.1 Perfil y Actividades del Administrador	9
1.3.2 Instalación	11
1.3.3 Interfaz Gráfica	14
1.3.4 Configuración de la Red.	15
2 Editor y Servidor WWW de Páginas Web	16
2.1 HTML	16
2.2 Servidor HTTP Apache	20
2.2.1 Directivas	21
3 Programación con Software Libre y Java	24
3.1 Sentencias de Control.	26
3.2 Arreglos	29
3.3 Trabajando con cadenas	29
3.4 El Sistema de Archivos	30
3.5 Programación con Java	31
3.6 Programando en Java	32
4 Bases de Datos MySQL	42
4.1 Tipos de Datos en MySQL	43
4.2 Creación de Bases de Datos y Tablas	44
4.3 Insertar Filas	46
4.4 Selección de Datos	46
5. Seguridad en Cómputo	53
5.1 Seguridad en el Servidor de Web “Apache”	60
6 Proyecto Pruebas Psicológicas a Distancia con Software Libre.	70
6.1 REQUERIMIENTOS	71
6.2 Esquema de la Base de Datos	72
6.3 Desarrollo.	73
7. Conclusiones	73

Introducción



En la actualidad es necesario contar con sistemas computacionales que permitan la automatización, seguimiento y control de información dentro de las empresas y oficinas. Con dicha automatización se logra un uso más eficiente de los recursos, y un control más preciso de los mismos, de tal manera que compartir la información a través de las redes (locales e Internet) se vuelve una tarea muy eficiente.

El desarrollo de los sistemas computacionales puede ser sustentado a través de software comercial como Windows, IIS, ASP y SQL Server por ejemplo, o mediante software libre como: Linux, PHP, Apache WWW-Server y MySQL, que son herramientas que han demostrado tener un alto desempeño, gran estabilidad y seguridad y, por el hecho de ser libres, permiten reducir los costos, logrando aprovechar al máximo los equipos de cómputo con que se cuenta.

Aprender nuevas herramientas administrativas de uso libre para desarrollar e implementar sistemas para el control de procesos de información resulta imprescindible, ya que la estabilidad del sistema operativo Linux y el software libre nos permite crear sistemas computacionales de alto desempeño y confiables.

Reporte del Proyecto Pruebas Psicológicas a Distancia con Software Libre.



El Diplomado “Desarrollo e implementación de Sistemas con Software Libre”, impartido en el centro de DGSCA Mascarones me resulto de gran ayuda para poder adentrarme al mundo del Software Libre, sin dejar a un lado el comercial como Windows o Mac, sino otorgándome una perspectiva aún mayor sobre las infinitas posibilidades para poder desarrollar sistemas computacionales, según las características del usuario (os) a los que van dirigidos.

Este diplomado me brindó un amplio conocimiento en las herramientas particulares para el desarrollo de los sistemas computacionales. Herramientas como PHP y MySQL que en la actualidad son dos de las más utilizadas a nivel mundial, resultaron ser indispensable para el desarrollo. Ambas herramientas resultan muy útiles ya que son sumamente sencillas. PHP tiene una importante similitud con el lenguaje de programación C++. MySQL utiliza sentencias casi idénticas a los utilizados por SQL, que, en lo personal, se ha convertido en un estándar para el manejo de información por medio de las bases de datos.

Uno de los módulos más interesantes resulto ser Apache Server. Es un programa muy sencillo en el que, por medio de directivas, podemos dar un servicio de Internet a cierto grupo de usuarios. Las diferentes directivas nos permiten desde dar límites de acceso, hasta la creación de hosts virtuales para poder utilizar más de una dirección de Internet con tan solo un equipo de trabajo.

HTML fue el módulo más pequeño en todo el diplomado. Esto por que su lenguaje de programación es uno de los más sencillos y fáciles de aprender. En este modulo se nos hizo realizar un pequeño proyecto en el que sólo representábamos de manera visual lo que sería una página de Internet, ya que en

esos momentos aún no contábamos con los conocimientos necesarios para el envío de la información, en el caso de los formularios. Ni las herramientas de seguridad necesarias para poder brindarla en el momento de acceso o el envío de información.

La seguridad en cómputo resulto muy interesante ya que en general los diferentes usuarios desconocen que es imprescindible su uso en los diferentes procesos que se llevan a cabo dentro de una red. Aquí se aprendió a encriptar cualquier información por medio de algunos comandos de Linux, con el principio básico de las llaves: pública y privada. De igual manera se nos demostró con profundidad lo que es la seguridad hoy en día, tomando en cuenta que en ciertas ocasiones una sola medida de seguridad nunca es suficiente.

El modulo de Java terminó siendo una pequeña introducción de lo que se puede hacer hoy en día mediante este lenguaje de programación que ha llegado para apoderarse de los diferentes procesos que se llevan a cabo en casi todos los sistemas hoy en día. En lo personal Java es un lenguaje muy complicado pero no por ello imposible de comprender. Java maneja un sin fin de librerías y paquetes que día a día aparecen dentro de la red lo que hace que sus diseñadores tengan que estar actualizándose constantemente. En lo personal es el lenguaje que ha creado la nueva tendencia de programación: la programación orientada a eventos.

Todos estos módulos se complementan al final para poder desarrollar el Proyecto de Pruebas Psicológicas a Distancia con Software Libre. Cada uno resulta indispensable para poder llevarlo a cabo. Desde los fundamentos esenciales de Linux, para poder fungir adecuadamente como administrador del servidor, hasta la programación en Java y PHP para el manejo y envío de información dentro de la página.

Sistema Operativo Linux



El modulo inicio con la historia de Linux, desde su creación con fines educativos, hasta el desarrollo final elaborado por: Linus Torvalds y con ello el nacimiento de lo que hoy se conoce como: software libre. Los comandos que se utilizaron en el modulo iban desde una pequeña consulta desde modo consola, lo que nos hizo recordar al ya extinto MS-DOS por la similitud en algunas instrucciones, hasta el inicio en el manejo del modo gráfico de Linux.

Aquí comenzamos con la creación de un usuario, grupos y privilegios que tendría cada uno de ellos. Haciendo mención en que únicamente el administrador que lleva por nombre root, dentro del sistema, puede dar de alta a usuario y grupos, y por consiguiente solo él es capaz de realizar algún cambio en los diferentes privilegios o permisos de los diferentes usuarios que tengan acceso al sistema.

El segundo paso fue aprender comandos que nos permitieran poder tener acceso a los diferentes textos informativos que hay dentro de Linux. Cabe mencionar que la mayoría de los programas a instalarse dentro de Linux, sino es que todos, posee una amplia documentación. Por eso la necesidad de enseñarnos los diferente comandos de visualización.

Después continuamos con los comandos de creación de directorios, la copia de archivos, enlistados, etc. Hasta llegar a lo que fue redireccionamiento, que es una herramienta muy útil cuando se manejan diferentes archivos. Cuando deseamos agregar información a un archivo o bien el resultado de algún proceso introducirlo a un archivo existente o no.

Después se hizo hincapié en la definición de las ligas: suave y dura. Ya que en general se malinterpreta ambas definiciones. La primera es un link que se hace hacia un archivo, programa o proceso, mientras que la segunda hace una referencia idéntica de lo que se desea. Si se elimina una liga suave tan solo se eliminará el link al que hace referencia, mientras que si se elimina una liga dura también será eliminado el contenido del programa. En pocas palabras sería como si se eliminara el programa por completo.

Más adelante se conoció archivos indispensables para el funcionamiento de Linux, como el `inittab` y `lilo.conf`, que como su extensión lo indica, son para la configuración del sistema. El primero hace referencia a la manera en que se iniciará el sistema y el segundo es el archivo con el cual se configura el gestor de arranque `lilo`, que permite combinar varios sistemas operativos dentro de un mismo equipo.

El siguiente paso en este módulo fue el llevar a cabo la instalación completa del sistema, teniendo como base los comandos anteriores. Aquí fue necesario demostrar las diferentes actividades y características de un administrador para poder tener una perspectiva completa de lo que debe saber hacer un administrador y así, llevar un adecuado funcionamiento dentro del equipo.

Lo complicado de la instalación es que el sistema reconozca todo el hardware del equipo ya que antiguas versiones de Linux no detectaban ciertos elementos. Entre ellos destacan las impresoras que resultan aún más complicadas cuando desean instalarse en red, y algunos módems. Lo curioso es que resulta muy sencillo el instalar los nuevos multifuncionales, quizás por que éstos últimos cuentan con nuevos controladores.

Una característica esencial para poder ser buen administrador es la instalación de los diferentes programas adicionales al sistema. Anteriormente era necesario instalarlos por medio de instrucciones en consola renglón por renglón

para que el sistema pueda cargarlo correctamente. En la actualidad resulta muy sencillo por que ahora, aunque también se realiza en modo consola, tan solo son unas cuantas instrucciones y el sistema lo instala automáticamente.

HTML y Servidor WWW



El curso de HTML resultó ser el modulo más sencillo de todo el diplomado, por lo simple de su estructura. Lo más complicado fue entender el proceso que se lleva a cabo al enviar la información ya que, al carecer de los conocimientos concernientes a PHP y MySQL, los ejercicios fueron simples ejemplos ya que la información se direccionaba hacia otro tipo de archivo.

En HTML partimos desde las etiquetas más sencillas como el body y head, hasta los frames dentro de una página; pasando por tablas y cierto manejo en el diseño visual de la página. Para esto comprendimos que en general el diseño de las páginas bajo este formato se basa en la creación y el manejo adecuado de las tablas, ya que son estas las que nos permiten dar cierta orientación al texto, imágenes y formularios que se contengan en la página.

Finalmente y como mencione en la introducción de este informe, el modulo termino con un pequeño proyecto, basado en la creación de una pequeña página muestra. Mi proyecto se basó en el desarrollo de un buscador de música que pretendía contar con una pequeña base de datos la cual sería para poder delimitar el acceso de los diferentes usuarios y el contenido de la propia página, dividido en diferentes secciones, desde la discografía de cualquiera de los grupos o artistas, las letras de sus discos, así como información general sobre el grupo/artista

HTML da paso a la definición de servidor, por ello, el siguiente modulo fue Apache Server. Una vez entendido el concepto cliente - servidor se pudo entender de una mejor manera lo que es un servidor y las diferentes características que debe de poseer. Desde el número de servicios con los que puede contar sin ningún problema y el puerto por el cual “escuchará”, para poder brindar dicho servicio. Hasta la creación de hosts virtuales que nos brindan varios servicios o servidores virtuales ejecutándose bajo un mismo equipo de trabajo.

Apache es un servidor muy popular y para poder entenderlo y configurarla únicamente se necesita conocer perfectamente el archivo: `apache.conf`. Este archivo es el encargado de controlar los diferentes procesos que brinda Apache, que además de lo ya mencionado también puede llevar bitácoras con diferentes formatos, también puede llegar a hacer una pequeña búsqueda con fines estadísticos que demuestren de que otras páginas se han direccionado los diferentes usuarios. De igual forma podemos controlar los diferentes errores que pueden llegar a ocurrir con el servidor, como cuando la información por alguna u otra razón no se encuentra en su sitio, podemos cambiar el formato del aviso de error o bien hacer un direccionamiento hacia otra página.

Apache puede tener algunos problemas en el momento de realizar los diferentes procesos. Por medio de diferentes comandos podemos iniciar, detener o reiniciar cada uno de los procesos que esta llevando a cabo Apache. Estos comandos pueden generar ciertos problemas en el momento que se este trabajando con él. El error o problema más frecuente es cuando deseamos realizar algún cambio dentro del archivo `apache.conf`, lo realizamos y guardamos los cambios pero no detuvimos el servidor, por lo que, al momento de reiniciar el servicio éste encontrara un problema con los direccionamientos de los diferentes procesos que llevo a cabo apache antes de ser modificado su archivo de configuración. Para solucionar este problema debemos “matar” el proceso que se levanto previamente con el apache antes del cambio de configuración y verificar que en realidad ya no haya ningún proceso andando en el sistema para ahora si,

poder levantar el nuevo proceso con las modificaciones previamente realizadas sin ningún problema.

Programación con Software Libre y Java.



Este capítulo tiene como finalidad demostrar los principios básicos de las herramientas de programación que pueden llevarse a cabo para la implementación de los diferentes sistemas en un entorno de red. PHP es una herramienta gratuita y con una amplia reputación ya que es una de las herramientas más utilizadas a nivel mundial, por su definición y por lo sencillo de su lenguaje; ya que es muy similar al lenguaje C.

Por otra parte, Java es la herramienta más utilizada en la actualidad en todos los aspectos que conciernen a desarrollo. Java no es gratuito pero si es una herramienta que puede abarcar un sin numero aplicaciones por medio de diferentes librerías a nivel mundial. Cada una de las librerías es una variante de una ya existente o una aplicación nueva. Todo esto para que Java llegue a ser no solo un lenguaje de programación sino todo un conjunto de herramientas para el desarrollo de las diferentes aplicaciones computacionales.

PHP resulto muy sencillo. En lo particular fue más un pequeño repaso de C con algunas cuantas variantes, como los formularios que se mezclan e interactúan con parte del código HTML. En general todo el código de PHP se introduce dentro del código HTML, pero a diferencia de éste, PHP permanece oculto a los diferentes usuarios.

Una de las variantes de PHP con respecto de C, además de los formularios, fue el envío de la información por medio de estos, debemos recordar que la información era enviada por algunos comandos de HTML como GET y POST. Con ayuda de PHP podemos manejar dicha información dentro de la página e incluso podemos hacer operaciones con ella, ya que podemos asignar los datos a una o más variables dentro del código.

Lo más complicado del curso resulto ser el manejo de archivos por medio de PHP, ya que a pesar de ser un programa muy popular casi no se encuentra documentación en línea de esta documentación, por lo que los ejercicios concernientes a este rubro tuvieron que darse en el clásico modelo de prueba y error. Los archivos con lo que puede trabajar PHP, son: MySQL o cualquier programa de bases de datos, archivos de texto plano y archivos de HTML.

En contraparte Java resulto más complicado por la sintaxis que varía mucho del resto de los lenguajes a pesar de también estar basado en el lenguaje C. En java por ejemplo existen diferentes maneras de enviar la información de salida, además de que la instrucción es un poco más larga que el resto de los demás lenguajes.

Lo interesante de Java es el manejo de las clases, los métodos y constructores que a pesar de que el resto de los programas también lo manejan, en Java se presta mas a ser utilizados para la programación orientada a eventos. Todo esto por las diferentes librerías dentro de Java.

Uno de los ejercicios que se llevo a cabo en este modulo fue el manejo de las excepciones, una característica mas de Java, lo que facilita el uso en las aplicaciones desarrolladas bajo estas plataformas. Resulto algo complicado al principio pero con un poco de trabajo se alcanzo a ver el potencial de esta herramienta. De igual forma realizamos ejercicios de herencia y poliformismo que también resulto algo complicado su desarrollo.

PHP y Java son herramientas que con el paso del tiempo se han convertido en estándares, cada una en diferentes áreas de trabajo pero con la posibilidad de interactuar entre ellas. Un ejemplo de la trascendencia que ha tenido Java a nivel mundial es la tecnología punto NET de Microsoft, que pretende ser una alternativa a lo que Java a llegado a ser en el mercado. PHP por su parte, también cuenta con su análogo Microsoft-ASP. Sin embargo la desventaja de PHP es que se ha estancado en su desarrollo tecnológico, desde hace un par de años no ha salido una nueva versión de PHP 5, además de que Java a comenzado a abarcar el desarrollo de sistemas bajo la red de Internet con nuevas características para las grandes empresas, por manejar a la perfección la programación orientada a eventos lo que hace independientes cada uno de los diferentes módulos del sistema.

Bases de Datos MySQL



MySQL es la herramienta de software libre por excelencia en el manejo de bases de datos. Basado en el estándar SQL y con características propias. Esta herramienta es muy fácil de utilizar y brinda grandes ventajas sobre otros controladores y sin ningún costo.

En el diplomado tan solo alcanzamos a ver unas cuantas características del programa por falta de tiempo, pero existieron ejercicios que resultaron algo complicados. Los diferentes comandos que vimos fueron los básicos, desde la creación de una base de datos hasta la interconexión de datos de varias tablas.

Lo que resulto complicado fue el hecho de instalar el programa, que a pesar de ser muy sencilla su instalación a nosotros se nos complico un poco ya que

tuvimos que instalar nuevamente Apache por que en el anterior modulo éste no se había instalado con soporte para MySQL, sólo con soporte para PHP. En este proceso surgen siempre pequeños problemas de instalación como en el resto de programas, aquí todo parecía indicar que todo esta instalado de la manera adecuada. Sin embargo, tardamos un clase en encontrar el error que resulto ser general para todo el grupo.

Al haber dado de alta un primer Apache sin soporte para PHP y MySQL a la vez, se creo un demonio (proceso que se lleva a cabo en segundo plano) de apache el cual contenía ciertas condiciones para su manejo, instalación y por supuesto desinstalación. Por ello una vez que nosotros eliminamos el anterior apache e instalamos el otro, nuestro demonio que debía reiniciarse no lo hizo, por que ya existía un demonio en proceso y este no concordaba con el que nosotros acabamos de instalar.

Una vez instalado Apache ya sin problemas pudimos comenzar a utilizar MySQL en modo consola con lo que aprendimos que hay que tener cuidado al momento de realizar una instrucción ya que en ocasiones éstas llegan a ser muy grandes y suelen ocurrir errores al momento de golpear las teclas.

Finalmente utilizamos una versión gráfica de MySQL en donde pudimos alcanzar a ver de manera gráfica y un poco más sencilla el manejo de las tablas ya que aquí se representan por medio de carpetas en una forma tradicional de árbol lo que facilita el manejo de las mismas. Las instrucciones se llevan escriben en un área determinada en la parte superior del programa y los resultados aparecen en la parte inferior, donde también aparecen los errores si es que llegaran a existir al momento de realizar alguna instrucción.

Seguridad en Cómputo



Este módulo tuvo la finalidad de demostrarnos la importancia de la seguridad hoy en día. Conforme ha pasado el tiempo, las nuevas tecnologías han cambiado drásticamente, lo que ha orillado a una nueva forma para el manejo de la información. En la actualidad existen cada vez más personas capaces de manejar un lenguaje de programación casi a la perfección, que en manos equivocadas, puede convertirse en un peligro latente. Esto, aunado a la gran expansión de redes a nivel mundial y con el auge de Internet, ha generado una nueva área dentro de las empresas a nivel mundial. La seguridad en cómputo.

Para conocer lo que es la seguridad en cómputo es necesario conocer previamente ciertas definiciones y conceptos concernientes al área de cómputo. De entre estas definiciones tenemos: ataque, virus, seguridad y otras más. Una vez conocidas o recordadas es posible comenzar a trabajar sobre ellas. Un ejemplo sería virus, ya que existen diferentes tipos de virus. Desde los simples "observadores" hasta los que hacen que un proceso, sistema o usuario pierdan el control de la información o, en el peor de los casos, pierdan dicha información.

En estas definiciones se nos mostró un ejemplo de los virus activos. Su misión era alentar uno de los servidores más importantes a nivel mundial, el de yahoo. Este tipo de virus no daña información dentro del servidor, pero sí logra que se saturara al enviar varios paquetes de petición al servidor, lo que provocó que el servidor estuviera ocupado en contestar cada uno de estos paquetes en lugar del resto de peticiones a nivel mundial. En pocas palabras hizo trabajar más de lo debido al servidor por más de 4 horas lo que resultó en pérdidas multimillonarias a los dueños del portal.

Otro punto importante dentro de la seguridad de cómputo fue la definición de las diferentes llaves que puede haber. En general son dos: la llave pública y la privada. Cada una de éstas posee características individuales que hacen que la información sea entendida de diferentes maneras. Se utiliza la llave pública para que un determinado grupo de personas la posean y puedan descifrar cierta información. En este caso son dos usuarios o equipos los que conocen dicha llave.

La llave privada se utiliza cuando se desea que tan solo un usuario, equipo pueda conocerla. Esto resulta algo problemático para poder descifrar la información pero resulta de mucha utilidad cuando se descubre que puede ser utilizada para realizar lo que actualmente se conoce como firma digital. En pocas palabras esta llave te permite firmar la información que uno envía por la red. La firma digital es muy utilizada por los sitios de certificación en la red, que se encargan de validar y certificar a las diferentes páginas a nivel mundial, garantizando que la información que contenga su página es verídica y que cuenta con los permisos de Copyright existentes.

Para poder llevar a cabo un ejercicio con este tipo de llaves se nos pidió que descargáramos e instaláramos un pequeño programa en nuestros equipos. Este paquete permitía encriptar información, pudiendo ser texto plano, en el caso de un mensaje, u otro programa. Una vez encriptada la información se generaba una llave que nosotros podíamos hacer pública o privada. Para poder llevar a cabo el ejercicio tuvimos que hacerla pública para poder verificar que la información llegará a su destino en forma correcta.

Este punto me agrado mucho ya que descubrí que en Internet, al enviar ciertos mails por los diferentes servidores hay ocasiones en las que no se ve más que símbolos que parecen no tener significado alguno pero resultaron ser muy similares a los resultados de nuestro ejercicio de encriptación dentro del diplomado. Por lo que llegue a la conclusión de que hay ocasiones en las que los servidores de correo envían, o mejor dicho reenvían los diferentes mails de

manera encriptada. Mi única duda fue por que se hace. Es premeditado o es un pequeño error al momento de reenviar la información. En algunas ocasiones resulta extraño por que no contamos con ningún tipo de llave para poder descifrar la información.

Este modulo concluyó con una variante de HTTP, que es un protocolo para el envío de información de manera segura en Internet. Esta variante se conoce como HTTPS, donde la S denota la sigla Security. Dicha variante es muy utilizada para realizar transacciones en línea que garantice una transferencia segura, por ello es un método muy utilizado por la banca mundial.

Proyecto de Pruebas Psicológicas a Distancia.



El proyecto que se llevo a cabo como conclusión del diplomado es el de Pruebas Psicológicas a Distancia. Este proyecto representa los diferentes módulos, desde la administración de Linux para poder llevar a cabo un adecuado control del servidor bajo el Sistema Operativo. Una vista atractiva gracias a HTML y ciertas variantes en el manejo de la información con PHP. Un control adecuado de los diferentes datos de los usuarios con MySQL y garantizando seguridad. Todo esto en base al servidor Apache que garantiza servicios hasta de 150 personas a la vez.

El proyecto pretende ser una alternativa a la psicología tradicional ya que plantea la posibilidad de poder realizar una prueba psicológica sin la necesidad de estar presentes ninguno de los interesados. Pero sin dejar a un lado la terapia tradicional sino para brindar una alternativa para el desarrollo de dichas pruebas. Ya que generalmente se pierde bastante tiempo en el desarrollo de estas. El

proyecto evitará que se pierda tiempo en el desarrollo de las mismas ya que éstas se podrán realizar desde la comodidad de su casa, utilizando únicamente Internet.

Para poder realizar dichas pruebas se necesita cargar una base de datos, en primer lugar de los diferentes psicólogos que podrán tener acceso a los diferentes resultados de sus pacientes en cualquier momento y desde cualquier parte del mundo. Sin estos datos no es posible cargar los datos de los pacientes ya que no cualquier persona que tenga acceso a Internet puede llevar a cabo alguna de las pruebas. Un usuario necesita de la recomendación de alguno de los psicólogos previamente dados de alta para poder tener acceso al sitio. Para que una vez registrado pueda realizar alguna de las pruebas concernientes a su edad o problemática particular.

Una vez realizada la prueba por el paciente, el psicólogo encargado podrá obtener el resultado de la misma, generando así un pequeño archivo en línea del paciente. En este pequeño archivo aparecerán los resultados de la prueba y se podrá realizar un análisis con las debidas características y poder subir los resultados a la red para que el usuario pueda ver sus resultados ya interpretados por su doctor. Por lo que el paciente también tendrá acceso a su propio archivo virtual pero con diferentes características de las de su psicólogo.

Este proyecto no tiene competencia directa en el mercado. Si existen ciertos sistemas diseñados para empresas reclutadoras. La principal diferencia entre este proyecto y dichos sistemas es la posibilidad de poder hacerlo bajo Internet y no únicamente dentro de una red local, además de brindar la posibilidad para realizarse en bajo una terapia tradicional. En conclusión este tipo de sistemas podrían ser la competencia directa con este proyecto. Por consiguiente este proyecto tan solo es un pequeño prototipo para poder vislumbrar las posibilidades de este tipo de pruebas a distancia con la ayuda de la tecnología.

1. Sistema Operativo Linux



Prácticamente desde que aparecieron las computadoras personales (PC) hasta la actualidad, el sistema operativo que ha predominado ha sido el manufacturado por Microsoft. A inicios de los 80, Microsoft innovó en los ordenadores personales al presentar su ambiente gráfico y desde entonces ha sido el líder en ventas alrededor de todo el mundo.

Linux es una alternativa más en el amplio mundo de los sistemas operativos hoy en día. La diferencia más notoria radica en los costos. Microsoft cobra una licencia para el uso de su sistema, Linux, en contraparte, ha sido el creador de lo que se conoce como Software Libre, llegando a establecer un estándar a nivel mundial en el manejo, desarrollo y distribución del Software Libre.

Linux se considera gratuito. Sin embargo las distribuciones pueden llegar a tener un costo. Esto se debe a que varios grupos de programadores obtuvieron el código fuente original, que es completamente gratuito, y cada uno de ellos realizó mejoras, creando así las diferentes versiones ó distribuciones del sistema.

1.1 Breve Historia del S.O. Linux

El sistema operativo Linux surgió como un proyecto alternativo al sistema operativo Minix¹, realizado por Andrew S. Tanenbaum con fines educativos. Uno de sus estudiantes, Linus Torvalds, se basó en el código fuente de Minix para desarrollar su propio sistema, al no poder modificar Minix.

Linux surgió con el fin de encontrar un sistema operativo capaz de ser multiplataforma y modular, en otras palabras, para el uso de diferente tipo de software y con la capacidad de ser adaptable con los diferentes módulos como video, audio y sonido.

La diferencia más notoria entre Minix y Linux es que Linux ha sido de código abierto, es decir en cualquier parte del mundo, un programador puede obtener el código y modificarlo según sus necesidades particulares. Al núcleo del código se le llama comúnmente Kernel y sus funciones incluyen el gestionamiento

¹ Minix es un clon Unix, distribuido junto con su código fuente. Este sistema se creó con fines educativos, ya que su original contaba con restricciones de licencia.

de los diferentes recursos como: memoria, procesos, módulos, etc. de los diferentes equipos.

Dentro de la filosofía de Linux se encuentran los siguientes puntos:

- Entre más pequeño mejor
- Reducir esfuerzos combinando herramientas
- Personalizable por cada usuario

La compatibilidad de Linux abarca a los siguientes equipos:

- I386 PCs como mínimo
- PPC (para McQuintosh)
- Sparc Estaciones de Trabajo. (equipos dedicados a procesos particulares)
- Alpha (Arquitectura de estaciones de trabajo con procesadores en paralelo).

Existen diferentes distribuciones del sistema operativo, como son: Slackware, Red Hat, Suse, Debian y muchas más. La diferencia entre estas alternativas son: principalmente su distribución y nivel de complejidad. Mientras que unos son muy grandes y costosos, como es el caso de Red Hat, existen otros que su distribución es gratuita y casi con las mismas capacidades de Red Hat.

Para poder comenzar a manejar el sistema operativo de Linux (Slackware v 10) es necesario conocer algunos comandos básicos y su funcionamiento. Y como se podrá ver, tienen cierto parecido a algunos comandos del ya extinto MS-DOS². Algunos de ellos carecerán de explicación por que viene implícito en su traducción.

1.2 Comandos Básicos

Los comandos básicos son todos aquellos que se realizan en consola, o en otras palabras en línea de comando. Esta es la estructura general de todos los comandos en Linux. Muy similares a los de la plataforma Unix.

comando – opciones arg1 arg2

adduser: se agrega el nombre del usuario que contará con ciertos privilegios ya que no cualquier usuario puede tener acceso a todos los archivos del sistema. El único usuario que es capaz de hacerlo es root³ quien es el que puede agregar, cambiar permisos, acceder y modificar cualquier archivo si así lo desea.

² Micro Soft Disk Operating System. De sus siglas en inglés sistema operativo de disco.

³ Es el usuario por default que aparece en Linux. Este tiene los privilegios de Administrador del equipo.

passwd: una vez que se han agregado los datos y asignado su contraseña para su acceso podemos utilizar este comando para cambiar nuestro password ya que comúnmente es el administrador quien asigna dichos passwords.

A continuación enunciaremos en la Tabla 1.1 ciertos comandos de Linux.

Comando	Características y Opciones
userdel	Elimina al usuario que deseemos. Únicamente como root. -r Elimina todo lo guardado por ese usuario.
chsh	Permite cambiar el shell del usuario
chfn	Cambia los datos personales del usuario
hostname	-i dirección IP -f nombre de la máquina y dominio. -d dominio del servidor
uname	Indica el sistema que se encuentra trabajando -n nombre del servidor -r versión del kernel -m arquitectura del procesador -a todos los datos del kernel
pwd	Imprime la ruta absoluta donde se está trabajando
touch	Crea un archivo vacío para funciones especiales
mkdir	Make Dir -p Crea todos los directorios padre -v Informa sobre la creación de los directorios paso a paso
tree	Muestra la estructura completa de los directorios.
rmdir	Remove Dir -r Borra un directorio, sólo si se encuentra vacío.
rm	Elimina cualquier archivo con toda su información -r Recursivo, borra todo -f Fuerza a que se borren los archivos, aún sin permiso.
ls	Lista todos los directorios existentes -l En formato largo -a Muestra todo, incluso los archivos ocultos -i Muestra el inodo inicial del archivo (ubicación física en disco) -r Listado inverso de manera alfabética -R Listado de directorios a partir de una dirección absoluta -t Listado en orden cronológico -F Definirá archivos y directorios
cp cp origen destino	-r Copia recursiva, también copia los directorios -i Copia interactiva, pregunta por cada elemento a copiar -v Informa el avance de la copia
mv	Mueve el archivo a la posición indicada. También renombra.
chmod	Modifica los permisos de los archivos
kill	Detiene por completo cualquier proceso
fg y bg	Envía los procesos a primer plano o a segundo

Tabla 1.1 Comandos Básicos de Linux

Ctrl + z	Suspende cualquier proceso
ps	Muestra todos los procesos que son corridos en ese momento _a Muestra los procesos en otra terminal -fea Nos muestra el número de cada uno de los procesos. x -forest Muestra los procesos de manera jerárquica.
top	Muestra los procesos generales de manera dinámica.
ln	Genera una liga o vínculo entre un archivo y un nombre. -s Genera una liga suave o un acceso directo.
shutdown	Apaga o reinicia el sistema según su opción -h now Detiene el sistema y lo da de baja -r now Permite reiniciar el equipo. -h 60 Indica en cuanto tiempo se apagará el equipo.
man (comando)	Proporciona el manual escrito del comando que deseemos
cat	Permite revisar todo el contenido de un archivo. Puede concatenar 2 archivos o mas -n Enumera cada una de las líneas
more	Visualización de un archivo página por página
less	Permite movernos a través del archivo (regresar avanzar)
head	Muestra tan solo las primeras 10 líneas del archivo -n Para indicar el número de líneas a ver
tail	Muestra las últimas 10 líneas del archivo, también usa -n
pico	Es un editor de archivos para agregar o modificar los archivos
>	Redirecciona la salida de un programa a un archivo si este no existe lo crea.
<	Redirecciona de entrada, para agregar a un archivo
	Concatena o redirecciona un programa con otro
>>	Redireccionamiento no destructivo o de adición
1>, 2>	Redireccionamiento condicionado, elige 1 si es verdadero
Grep	Busca patrones de directorios. Devuelve todas las líneas que contengan un patrón determinado. -c Devuelve el no. de líneas que contenga el patrón -i Ignora la diferencia entre minúsculas y mayúsculas. -H Imprime las líneas y el nombre del archivo donde se encuentra. -l En múltiples directorios donde solo muestra aquellos en donde esta. -v Devuelve las líneas que no poseen dicho patrón. -n No. de la línea que contenga el patrón “ “ Busca texto dentro de un archivo.
find	Realiza búsquedas.
gzip	Comprime un archivo.
gunzip	Descomprime un archivo
tar	Empaqueta archivos, muy parecido a gzip -c Comprime -v Visualiza -f Que será enviado un archivo

Tabla 1.1 Comandos Básicos de Linux (Continuación)

	-x Ejecuta el archivo descomprimido -z Comprimirá o descomprimirá el archivo tar -zxvf Empaqueta y comprime a la vez o viceversa.
scp	Realiza una compresión de manera segura. Pide una contraseña para poder utilizar este comando.

Tabla 1.1 Comandos Básico de Linux (Continuación)

Comodines: resultan muy útiles cuando realizamos una búsqueda si no conocemos el nombre del archivo. Ejemplo:

ls ??a Busca archivos que terminen con la letra “a”
 ls ??a* Buscan archivos que poseen una “a” en su tercera posición
 ls *u*t Muestra archivos que lleven “u” y “t”, en el orden que se muestre.

Ruta Absoluta: inician con; /, que indica el punto más alto de la estructura de directorios, a partir de aquí se pone la ruta completa a donde se desea llegar.

Ruta Relativa: comienzan desde el directorio donde nos encontramos.

Las rutas relativas no resultan muy útiles cuando se trabaja con diferentes equipos ó sesiones.

Tipos de Archivos: se distinguen por el primer bit que indica el archivo del que se trata (Tabla 1.2)

l	Es una liga
c	Archivo carácter (archivos con acceso por HW)
d	Directorio
s	Referencia de un archivo socket (conexiones remotas)
p	Referencia a las tuberías (pipe)
-	Archivo ordinario
b	Archivo de bloque (acceso solo por bloque)

Tabla 1.2 Tipos de Archivos

Permisos: como en todos los sistemas operativos existen archivos que no cualquier usuario puede manejar, para ello existen los diferentes permisos que puede poseer cada uno de los diferentes archivos (Tabla 1.3)

r	lectura	Se puede leer y copiar
w	escritura	Se puede escribir, modificar y borrar
x	ejecución	Realiza un proceso

Tabla 1.3 Permisos

Los archivos ejecutables al momento de listarlos se muestran de color verde, las ligas de color azul además de ir acompañadas de una @ y los archivos normales en blanco.

Método Simbólico Igualado: con `chmod` podemos cambiar los permisos de los diferentes archivos usando una igualación según lo deseemos. Ejemplo:

`chmod u=wx,g=x,o=x`

La letra “u” es para los permisos de usuarios, “g” para los grupos y por último se asigna la letra “o” para los permisos de otros.

Método Octal: los permisos son asignados directamente con números, que provienen de una tabla donde se indica que tipos de permisos son posibles asignar. (Tabla 1.4)

Ninguno	0	- - -
Ejecución	1	- - x
Escritura	2	- w -
Escritura y Ejecución	3	- w x
Lectura	4	r - -
Lectura y Ejecución	5	r - x
Lectura y Escritura	6	r w -
Todos	7	r w x

Tabla 1.4 Permisos en modo Octal

Estado: representa como se encuentra algún proceso, mismo que puede estar siendo ejecutado ó en pausa. También puede estar en un segundo o tercer plano según se éste trabajando. (Tabla 1.5)

Z	Zombie, cuando el proceso ocurre de manera independiente.
R	Estado de ejecución
T	Proceso detenido
S	Dormido o en espera
D	Dormido sin interrupción posible. (I/O)

Tabla 1.5 Estados

Liga Suave: es un acceso directo, ya sea a un archivo, proceso ó programa

Liga Dura: mantiene relación directa con un proceso pero esta liga posee el mismo tamaño del archivo al que hace referencia. Útil en archivos de seguridad dentro de servidores.

Las ligas son muy útiles para respaldar información porque cualquier cambio que ocurra en la liga se aplicará en el archivo que referencia.

Archivo lilo.conf: posee toda la información de este gestor de arranque. Posee el boot que indica que S.O comenzará a cargarse en nuestro equipo, ya que linux nos brinda la ventaja de poder trabajar junto con otros sistemas operativos.

Resulta un poco complicada la adecuada instalación de este archivo. Si no está bien configurado Linux no arrancará ni mucho menos los demás sistemas operativos.

Archivo Inittab: es un archivo en el que podemos indicar mediante una tabla (Tabla 1.6), como trabajará nuestro equipo.

0	Halt	No permite que la maquina sea encendida
1	Usuario Unico	Carece de Servicios
2	Unused	Multiusuarios
3	Multiuser mode	También para multiusuarios
4	X11 with KMD/SDM/XDM	Interfaz Gráfica y Multiusuario
5	Unused	Igual que el no. 2
6	Reboot	Reinicia el equipo constantemente

Tabla 1.6 Modos de Inicio en Linux

1.3 Instalación y Administración de Linux

Este módulo resultó de mucha importancia ya que gracias a él se pudo obtener una visión completa de las responsabilidades y conocimientos que debe de poseer el administrador de un sistema operativo, sin importar cual sea. Desde puntos elementales de mantenimiento del equipo, términos de seguridad, pasando por programación y manejo eficiente de diferentes procesos administrativos dentro del sistema.

Además de resultar un guía indispensable, que a pesar de ser mínima, resulta de vital importancia, para al menos, poseer una idea de lo que un administrador debe de saber hacer; incluso como salir de ciertos problemas rutinarios.

En este módulo también aprendimos a instalar por completo Linux, desde el inicio, vimos, las diferentes particiones necesarias para nuestro sistema, hasta la instalación de la interfaz gráfica. Tomando en cuenta cada una de las especificaciones mínimas para el uso adecuado del sistema operativo bajo ambiente gráfico ya que resulta inseguro trabajar bajo éste cuando se trabaja con fines de servidores o administradores de un sistema. Un ambiente gráfico requiere de muchos procesos para poder trabajar de manera eficiente. Dichos procesos pueden ser utilizados como medios de ataque directo hacia nuestro sistema por lo que se recomienda llevar una administración a nivel consola para evitar debilidades dentro del sistema.

1.3.1 Perfil y Actividades del Administrador

Un administrador debe configurar, actualizar y mantener el sistema o conjunto de sistemas, llevando a cabo el mantenimiento del Software, Hardware y periféricos para tenerlos a disposición de los usuarios.

En Linux se debe de verificar el sistema operativo que se desea instalar ya que previamente hay que conocer las pruebas de desarrollo del sistema, sus parches, su compilación, correcciones, etc. y sobre todo, conocer la compatibilidad del Hardware en HLC⁴

Se debe conocer a la perfección el sistema operativo para propiciar un ambiente seguro, eficiente y confiable. Es recomendable dividir el trabajo entre varios administradores, dependiendo del tamaño del sistema y llevar en conjunto un adecuado manejo de las Políticas de Seguridad, para prevenir sucesos lamentables.

El administrador debe mantener la seguridad del sistema, comenzando con el control adecuado de los usuarios, que en la mayoría de los casos son responsables de que diferentes procesos y dispositivos dejen de funcionar de manera adecuada, para ello se debe de llevar un control administrativo, saber quien entra, y que hace, para así, poder asignar los permisos adecuados para cada una de las actividades que lleven a cabo los diferentes usuarios. Las llaves o claves de usuarios son perfectas para esto, por ello cada usuario debe poseer un login y una contraseña.

Tareas Administrativas. Son aquellas que se llevan a cabo para mantener la confiabilidad del usuario. Asegurar que únicamente él es concededor de los procesos que se realizarán. Haciéndolo partícipe en las diferentes actividades para así poder actuar según lo requiera la empresa en caso de incurrir en alguna falta de seguridad.

Algunas de estas tareas son:

- **Administrar Usuarios:** aquí es donde se lleva a cabo la derogación de responsabilidades, se realizan las políticas de seguridad, los requerimientos, atributos, privilegios, permisos, etc.
- **Configuración de Dispositivos:** es una tarea que va de la mano con el monitoreo y el conocimiento de la HCL
- **Respaldos:** se recomienda que exista un grupo de administradores encargados únicamente para esta actividad. Realizan los seguros, etiquetados y revisiones periódicas dentro del sistema

⁴ Hardware List Compatibility . Lista manejada por los diseñadores en las primeras versiones de Linux, donde se indicaba la compatibilidad que tiene el sistema con algunas marcas.

- **Capacitación de Usuarios:** es indispensable para un manejo adecuado del sistema, se debe llevar a cabo por personas especializadas.
- **Aesorar Usuarios:** indicar a los usuarios los diferentes servicios otorgados.
- **Asegurar el Sistema:** es, en pocas palabras parchar el sistema, cada que lo requiera así como verificarlo continuamente, ayudado siempre de las políticas de seguridad para poder actuar.
- **Registro de los Cambios del Sistema:** es reiniciar el sistema por completo por alguna actividad no permitida o errónea. Para ésto se recomienda llevar una bitácora con todos los sucesos de esta índole para derogar responsabilidades.

Actividades del Administrador.

- **Mantenimiento de Llaves de Usuarios.** Verificar el adecuado servicio, según los requerimientos y privilegios. Buscar passwords débiles que pudieran resultar en una vulnerabilidad directa hacia el sistema.
- **Instalación y Mantenimiento de Dispositivos.** Impresoras, que resultan las mas problemáticas, por ello se recomiendan las multiusos. Los discos y unidades de respaldo.
- **Instalación y Actualización del Software.** Comercial y de dominio público.
- **Configuración de las Interfaces de Red.** Realizar estadísticas, revisar la configuración porque algunos usuarios pueden cambiar las configuraciones originales o incluso encimar varias IP en una misma.

Administrador de los Recursos

- **CPU.** Monitoreo constante por medio del comando TOP procesos, uso de memoria, etc.
- **Memoria.** Monitoreo, para que no exista un uso inadecuado de la misma.
- **Disco.** Verificar el espacio, sus particiones y el uso de los usuarios sobre éste.
- **Monitoreo del Sistema.** Para esto se recomienda tener siempre una o dos personas frente a los bancos de datos para cuidar el equipo de posibles ataques exteriores o incluso internos.
- **Detección de Fallas.** Para poder mantener el servicio adecuado y óptimo
- **Auditoria e Implementación del Sistema.** Se recomienda que siempre sea externa para así poder tener una valoración adecuada e imparcial.

Conocimientos del Administrador.

- Técnicas de Programación
- Dominio de al menos un Lenguaje de Programación
- Funcionamiento del S.O
- Técnicas de Administración del S.O
- Conocimientos básicos de Hardware y Mantenimiento de Dispositivos

- Redireccionamientos y tuberías a la perfección.
- Editor de Textos *vi*
- Programación en Shell
- Utilerías del Sistema
- Documentación
- Hardware de cada máquina
- Establecer canales de comunicación

Importante tomar en cuenta.

- Establecer políticas de seguridad
- Aperturas de cuentas
- Horas de mantenimiento
- Responsabilidades de respaldos
- Borrar archivos temporales
- Seguridad del sistema

1.3.2 Instalación

Las siguientes páginas son referentes a los pasos a seguir para poder contar con una buena instalación del S.O Linux Slackware v.10.

```
boot:bare.iroot = /dev/hda1 noinitrd
```

Este es una línea de comando que aparece una vez insertado el disco no. 1 de la instalación e indica donde comenzará a cargarse el sistema. Aquí se muestra la unidad de disco primaria (hda1). Esto a pesar de que ya tengamos instalado otro sistema operativo.

Se debe mencionar que la instalación se llevó a cabo en un disco previamente particionado y cargado con los sistemas Windows2000 Server y Windows XP. En una computadora donde solo tenemos Windows, es necesario hacer una partición directa y volver a instalar Windows con todos sus componentes en la primera partición ya que si se instalara en la segunda ó sin partición, el programa gestor de arranque no podría iniciar ninguno de los sistemas, ya que Windows, utiliza todo el MBR y no permite que Lilo pueda actuar e indicar que sistema será cargado.

Una vez instalado y cambiado la forma en que se cargará el sistema se debe de elegir el idioma en que se manejará el teclado. Viene por default el idioma inglés. La diferencia entre los idiomas es que existen teclas ocultas o algunas que no existen en todos los idiomas. La opción que se debe de buscar es *qwert/es.map* y se realiza una pequeña prueba.

A continuación se muestran los pasos a seguir para poder instalar nuestro sistema operativo.

Paso No.1 Particionamos el disco con cdisk o fdisk seguido de /dev/hda que indica el disco duro a particionar (Tabla 1.7).

m	para ayuda al formatear
p	para imprimir las tablas de partición
n	agrega una nueva partición
d	elimina una partición
t	tipo de partición
l	lista las particiones
w	guardar los cambios
q	salir sin cambios

Tabla 1.7 Lista para Particionar

En esta opción instalamos más de una partición para los diferentes directorios que requiere Linux como: swap, home, usr, mail etc.

Particiones recomendadas bajo un disco con un mínimo de 4 Gb para la instalación adecuada de Linux

Partición	Capacidad
swap	2Gb (máxima en procesadores Intel, 256Mb optima)
/	3Gb
/home(hasta 100)	4Gb
/var/spool/mail	2Gb
/usr/local	5Gb

Tabla 1.8 Lista de Particiones Recomendadas

Las particiones primarias únicamente pueden ser 4. Para que todas estas puedan ser instaladas se deben crear particiones extendidas, recomendable de la última partición primaria, con el siguiente comando.

`e extended +3000M` (para asignar el tamaño de la partición)

Blocks: Son los valores numéricos que indican aumento o reducción de los cilindros del disco duro para realizar la partición extendida de manera adecuada.

Nota: Se debe mencionar que el programa gestor Lilo tiene ciertas particularidades, además de que requiere localizarse en la partición primaria del disco. Para empezar Lilo tiene dos versiones *simple* y *expert* donde la primera indica que automáticamente podrá cargar 2 diferentes sistemas operativos mientras que la *expert* permite la instalación de 2 ó más sistemas. Para realizar una prueba de Lilo tecleamos el siguiente comando: `lilo -t`

Paso No.2 Entramos al Setup de Linux para comenzar la instalación de todas las librerías y componentes que requiere el sistema.

Paso No.3 En este paso podemos elegir si nuestro sistema será instalado en una estación portátil de trabajo (PCMCIA)

Paso No.4. Configuramos nuestra tarjeta de red según los parámetros de la red con la que contamos en el área de trabajo. Cabe mencionar que la instalación de red vía módem resulta bastante complicada por las diferentes marcas de módem, por ello se recomienda trabajar en una red local o con un proveedor de Internet de Banda Ancha.

Inicio del Sistema

1. Umlinux. Es una liga con el Kernel cuya finalidad es la de comenzar a cargar en memoria los dispositivos ya reconocidos y localizar el script de inicio del propio sistema.

2. Entra al archivo "inittab" (tabla de inicio) donde localizará los scripts y realiza otras tareas concernientes al inicio del sistema.

3. Se cargan los archivos que facilitan e inician los procesos.

4. rc.local es el último archivo al que se accede, éste interpreta el sistema y trabaja en tareas particulares de manera local.

INITTAB Es el archivo de inicialización del sistema, donde se indica bajo que modo se trabajará.

1. Monousuario o de mantenimiento. Este archivo no levanta ningún servicio. Se utiliza frecuentemente para tareas administrativas.

2. Multiusuario. Es la opción que viene por default y es necesaria para cargar Slackware

3. Trabajar con X11 que es la opción que permite levantar la interfaz gráfica gracias al KDE⁵ del sistema.

RC.D: es un archivo que inicia cada vez que la computadora es encendida, ya que es el archivo encargado de iniciar y finalizar los diferentes procesos.

Una opción que resulta muy eficaz para asegurar el buen funcionamiento del sistema es la de deshabilitar la opción conocida como "las tres teclas" que reinicia nuestro equipo cada vez que sean presionadas. Esto que podría producir un problema al momento de volver a cargar el sistema ó bien si nuestro equipo funge como un servidor y algún usuario consigue acceso al equipo puede dar de baja todos los servicios que brindamos, para ello comentamos con un "#" la siguiente línea dentro de rc.d

⁵ K Desktop Environment, es el entorno de escritorio gráfico e infraestructura de desarrollo para sistemas Unix y en particular en Linux.

ca :: ctrlaltdel : /sbin/shutdown -t5 -r now

1.3.3 Interfaz Gráfica

El comando para comenzar a utilizar nuestra interfaz gráfica es *startx* que se encuentra en el directorio */etc/X11* donde también se encuentra el archivo *xorg.conf* y otras versiones gráficas para otras distribuciones de Linux. (*XF86Confi*, *XF86Config.4*)

xorg.conf: aquí configuramos cada uno de los elementos que deberán actuar sobre la interfaz gráfica. Como el teclado, al que le debemos indicar que tipo de lenguaje se utilizará. El monitor, del que podremos alterar su sincronización pero se recomienda no hacerlo y el *VertRefresh* que indica como será refrescada la pantalla de manera horizontal, preferentemente con un valor menor al de 70.

Extravío de Contraseñas.

Se debe de iniciar el sistema con otro dispositivo. En este caso será desde el Dispositivo del CD-ROM, que nos dará como resultado un inicio desde el prompt a nivel consola para poder cargar lo que necesitemos para recuperar o cambiar la contraseña.

En este caso la contraseña será removida ya que no puede ser recuperada por que al momento de ser almacenada en los archivos de bitácoras es encriptada para asegurar la confiabilidad de los usuarios.

Entonces respaldamos el archivo *shadow*, que no es sino una liga dura del archivo *passwd*. Entramos al editor y simplemente eliminamos los datos que aparezcan en el campo de contraseña. Si eliminamos directamente el campo de la contraseña desde *passwd*, ese cambio se tomará como una nueva contraseña por lo que continuaremos sin poder acceder al sistema.

Si en algún caso llegásemos a perder el gestor de arranque utilizamos la opción *bare:1*⁶

Pkgtool: es un administrador de paquetes, muy parecido al Agregar y Quitar Programas de Windows. Éste elimina todos los componentes relacionados con el programa deseado ó bien, lo carga si se encuentra en su base de datos y no fue dado de alta en el momento de la instalación.

Sino contáramos con *pkgtool*, o la versión del programa que posea no es la que necesitamos, bajamos el archivo de la red, verificando que sea un archivo seguro con *md5*⁷.

⁶ Esta opción es la misma que utilizamos para indicar desde donde inicia la carga de los diferentes módulos del sistema.

⁷ Firma certificadora que indica que el archivo no ha sido alterado por algún tercero.

Una vez que el programa se ha descargado descomprimos y ejecutamos el paquete. En este punto podemos seguir las instrucciones del archivo INSTALL si es una versión antigua ó bien, seguimos los pasos ya predeterminados para la compilación e instalación adecuada del paquete.

1. `cd nombre_paquete`
2. **./configure**. Este comando analiza el sistema y realiza sus comparaciones requeridas para la compatibilidad y genera un archivo de instalación.
3. **make**. Aquí es donde se compila el programa y se prepara el sistema para la instalación.
4. **make install**. Realiza la instalación completa del programa.

Anteriormente estos pasos tenían que realizarse de manera manual y línea por línea lo que ahuyentaba a muchos usuarios porque requería conocimientos mínimos de programación en Shell y sobre el mismo Kernel, además de resultar una tarea aburrida y tardada. Ahora con estos comandos se simplifica la instalación de cualquier programa.

1.3.4 Configuración de la Red.

Netconfig. Es el asistente para poder cargar nuestra red de manera fácil y rápida. Aquí indicaremos todos los datos de nuestra red: el nombre del Host, el dominio, la IP , etc.

Una vez instalada la tarjeta de red realizamos el siguiente comando para verificar que ha sido instalada de manera adecuada. *ipconfig*

Con ese mismo comando pero agregando *eth0 IP* podemos cambiar la dirección IP

En el caso de que la red contenga un ruteador agregamos esta línea en el archivo *rc.d*

```
route add default gw 132.248.75254 (ip del gateway)
```

El archivo donde se indicará la dirección del Host será */etc/resolv.conf*

En el manejo de la red, resulta conveniente conocer todos los puertos, servicios y demonios⁸ que tenemos corriendo, mismos que pudieran ser utilizados por terceros para realizar ataques. Para esto contamos con el comando *netstat*

Una vez verificados todos los puertos que poseemos y su funcionamiento, podemos dar de alta o de baja los servicios en el archivo *inet.conf*.

⁸ Procesos ocultos que se llevan a cabo en segundo plano.

2 Editor y Servidor WWW de Páginas Web



2.1 HTML

HTML es, para algunos, un lenguaje de programación muy sencillo. Pero es tan sólo un Lenguaje de Marcación de Hipertexto por sus siglas en inglés (HTML) que maneja etiquetas, que requiere de un editor de archivos y un explorador para poder realizar una página de Internet.

En la actualidad HTML ha dado paso a las nuevas formas para generar páginas Web desde el XML que es muy similar a éste, hasta los Java Scripts¹ que hoy en día son de crucial importancia para dar un formato agradable al usuario y un uso eficiente a su diseñador.

HTML es un lenguaje muy sencillo gracias al manejo de las diferentes etiquetas que utiliza, facilitando así, la programación de páginas Web vistosas y eficientes en cualquier ámbito donde se desee aplicarlas.

Una página Web dinámica debe tener las siguientes consideraciones para su desarrollo:

- Objetivo de la página
- Información necesaria
- Información Organizada
 - Estructura
- Programas para la elaboración de la página (PHP, MySQL, Apache)
- Condiciones para subirla a un servidor
- Publicación y distribución
- Mantenimiento y Actualización constante una vez realizada.

Cuerpo Básico de un Documento HTML

```
<HTML>
  <HEAD>..... Encabezado de la página
    <TITLE>.....Título de la página
  </TITLE>
</HEAD>
```

¹ Código de programación que puede ser insertado en un archivo HTML para llevar a cabo un proceso, casi siempre de validación.

```

<BODY>.....Cuerpo de la página, lo que contendrá.
</BODY>
</HTML>

```

Etiquetas de Formato:**Físicas.** Cambia directamente la apariencia de la página**Lógicas.** Realiza cambios únicamente en el formato.

Las etiquetas pueden utilizarse de manera simple, y cuando incluimos un +, en la descripción significa que se pueden combinar otras opciones adicionales (Tabla 2.1)

Todas las etiquetas se cierran con el mismo símbolo con una diagonal previa.

Etiqueta	Función
 	Etiqueta que realiza un salto de línea
<hr>	Salto de línea pero con marca. +width="%" Es el ancho de la marca +hieght="%" Es el largo o alto de la marca.
<p>	Es para realizar un párrafo completo. +align="" Para darle una alineación al texto que se introduzca en este párrafo +font="" Fuente del párrafo +fontsize="" Tamaño de la fuente +color=0000000 Color de la fuente en Hexadecimal
<pre>	Realiza un texto con formato prediseñado, donde se respetan los espacios tal y como sean escritos, así como los saltos de línea.
	Pone en negritas el texto que se encuentre entre estas etiquetas
<i>	Esta etiqueta hace itálicas las letras.
<u>	Subraya el texto
<tt>	Texto con formato de teletipo, muy similar a la maquina mecánica
<s>	Tacha el texto deseado
<h1-h6>	Es el tamaño del título; h1 es el mayor.
	Fuente del texto. +size="" Tamaño de la fuente +face="" Formato de la fuente +color="" Color de la fuente
<body>	Etiqueta que llevará el contenido en general de toda la página +text="" Color general de la página +bgcolor="" Color del fondo.
<blockquote>	Agrega una sangría al texto
<center>	Centra el contenido dentro de estas etiquetas
	Realiza un listado de manera descendiente +type="" El formato de las viñetas que se utilizaran en la lista.

Tabla 2.1 Etiquetas

Etiqueta	Función
	Lista ordenada de manera numérica +type="1,l,a" Para indicar como se ordenaran con números o letras +start="" Para indicar en que número empezamos la lista.
<dl>	Lista de definición. Un tema con sus subtemas
<dt>	Es un subtema en la lista de definición, tiene que ir dentro del cuerpo de la etiqueta <dl>
<dd>	Por si existen mas subtemas, dentro del cuerpo de <dt>

Tabla 2.1 Continuación

Tablas: Una tabla es un conjunto de celdas organizadas dentro de las cuales podemos alojar distintos contenidos.

En la actualidad los diferentes diseñadores de páginas web se basan en éstas para poder brindar un diseño “ordenado” a sus diferentes páginas.

A continuación se enlistan las etiquetas que requieren las tablas en HTML (Tabla 2.2)

Etiqueta	Función
<table>	Realiza una tabla +border Tamaño del borde +cellpadding Celda que abarque más de una columna (mezclar tablas) +width Ancho de la tabla +height Alto de la tabla. +cellspacing Espacio entre las celdas
<tr>	Indica que inicia una fila.
<td>	Indica el inicio de la columna +colspan Para abarcar más de una celda +valign Para alinear la dirección del texto
<caption>	Es para indicar el índice de la tabla
<th>	Cabecera de la tabla, resalta el título y lo centra

Tabla 2.2 Etiquetas de Tabla

Vínculos: Los vínculos son indispensables dentro de una página Web ya que éstos controlan la interacción con otras páginas, funciones dentro de la misma e incluso contactos con los diferentes usuarios de la página.. (Tabla 2.3)

Etiqueta	Función
<a >	+ name= “nombre de la etiqueta” Es un vínculo dentro de la misma página. + href “mailto” Hace referencia a un mail para establecer contactos con los diferentes usuarios. “www.page.com” Es un vínculo hacia otra página.

Tabla 2.3 Vínculos

Imágenes: una imagen es un conjunto de píxeles. Cada imagen puede poseer diferentes formatos como pueden ser: GIF, JPEG y JPG (Tabla 2.4)

Etiqueta	Función
<body>	+background="imagen.gif" Inserta una imagen en el fondo de la página +bgcolor=teal" Para que la imagen permanezca fija dentro de la página. +bgproperties La imagen mantiene su tamaño en el fondo
<image src>	+align Alinea la imagen dentro de la página Web. +width Para manejar la dimensión de la imagen en píxeles +height Tamaño de la imagen en ancho.

Tabla 2.4 Imágenes

Las etiquetas <a href> e <image src> pueden ser combinadas. Así, nuestra imagen también puede ser un vínculo; dentro de la página, hacia otra e incluso con un mail. Ejemplo:

```
<a href=www.google.com> <img src=gato.gif"></a>
```

Formularios: Los formularios son una parte importante dentro de una página Web en la actualidad. Se utilizan para poder llevar un registro de los diferentes usuarios que podría albergar la página ó simplemente para conocer algunos datos de los mismos. (Tabla 2.5)

Etiqueta	Función
<form>	Es la etiqueta que inicia nuestro formulario +name Para asignarle un nombre al formulario +method Opción para el envío de la información. Como lo recibirá y trabajará sobre ella PHP, Java, ASP; según sea el caso. +action Indica hacia que programa será enviada la información
<input type>	+hidden Es un formulario que permanece oculto al usuario. +text Indica un campo que será llenado con texto. +name Nombre de la variable que se manejará a futuro. +size Tamaño del cuadro de texto +value El valor por default del cuadro de texto +maxlength Valor máximo de caracteres dentro del campo +password Campo para introducir contraseña. No muestra la contraseña al escribir pero no cifra la información. +name Nombre de la variable +size Tamaño de la contraseña +maxlength Tamaño máximo permitido por el programador. +submit Es el botón que envía la información a su destino. +value Es el nombre que tendrá el botón.

	+radio El formulario tendrá opciones de llenado en forma circular. Uno por cada opción que deseemos +name +value Texto del que vendrá acompañada la opción. +checkbox Formulario de opciones en forma de cuadros +reset Vuelve en blanco todos los campos antes mencionados
<select name>	Cuadros de opciones en menú.
<option value>	Dentro de select y seguido de la opción que aparecerá en el menú.

2.2 Servidor HTTP Apache

Un grupo de investigación de CERN² fue el primero en desarrollar y manejar el concepto servidor/cliente, lo que terminó siendo lo que hoy se conoce como HTTP. Una vez terminada su investigación otorgaron ésta a la universidad americana NSCA. Esto resulta sorprendente, ya que se demuestra con esto que fueron los europeos y no los americanos quienes sentaron las bases actuales del "Web".

Apache es el nombre de un proyecto de servidor Web libre. El nombre apache tiene un origen un poco discutido, algunos dicen que viene de "**a patchy server**" debido a numerosos patches que se llevaron a cabo al inicio, otros dicen de una manera más seria que los investigadores de este proyecto tomaron el nombre en memoria de los Apaches por su gran adaptabilidad al terreno.

Este servidor es el más utilizado en Internet desde ya hace muchos años. Respeto el protocolo HTTP 1.1 normalizado por W3C³. Según los estudios de Netcraft⁴, Apache es utilizado en 56.19% de los Servidores a nivel mundial.

Un servidor Web es el programa "server" que responde a las peticiones de los clientes web.

Funcionalidad

El servidor web es la interfaz entre el cliente web que pide una URL⁵ y el sistema donde se encuentra dicho servicio.

Efectivamente, el cliente Web envía su petición, y el servidor regresa las páginas HTML correspondientes a la URL solicitada. El cliente solicitará las imágenes que se encuentran dentro de ellas y el servidor las enviará igualmente.

² Centro Europeo de Investigación Nuclear

³ W3C Consortium Asociación Internacional de empresas, organizaciones y público en general para el desarrollo de estándares en Internet.

⁴ Netcraft. Compañía que monitorea las actualizaciones de los diferentes servidores de Internet, también realiza pruebas de seguridad y publica noticias sobre los estados de variación en la Red.

⁵ Uniform Request Locator

Pero él puede, siguiendo la petición, hacer la interfaz con algunos programas del sistema, elegidos o escritos por el administrador del sitio, llamado WebMaster; estos programas son llamados scripts CGI⁶ y regresan una página HTML.

2.2.1 Directivas

Las directivas son las diferentes instrucciones en las que se basa el servidor Apache para poder brindar un servicio adecuado. Todas estas directivas pueden ser modificadas por los programadores para poder personalizar el funcionamiento de nuestro servidor según se necesite.

Directiva	Función
Servername	Indicamos el nombre que tendrá el servidor.
Serveradmin	Nombre del administrador del servicio
User	Se nombra al usuario
Listen	Indica que puertos podrá oír nuestro servidor
Startservers	Indica los servidores máximos que podemos correr a la vez
Minspareserver	Mínimo de servidores asíncronos
Maxspareserver	Máximo de servidores asíncronos
Maxclients	Número máximo de clientes
Maxrequestperchil	Peticiones que puede llegar a tener un hijo
Errodocument	Indica a nuestro servidor los pasos a seguir en caso de algún error.
LogLevel	Envía todo tipo de información a los archivos bitácoras que poseemos
Customlog	Genera la bitácora de accesos al servidor con un formato determinado
Hostnamelookup	Activa la opción DNS para poder visualizar el nombre de la URL en bitácora.
UserDir	Crea un document root para cada usuario. Cada usuario debe tener un HTML. Carpeta general a la que todos los usuarios tienen acceso.
DirectoryIndex	Busca los archivos que contengan los directorios que deseemos, siguiendo un orden previamente asignado.
Directory	Activa otras directivas de forma específica. Todos los recursos deben de estar dentro de directory. OPTIONS Son opciones que pueden tener cada una de estas directivas. ExecCGI Permite realizar CGI en esta directiva FollowSymLink Ligas simbólica válidas. IndexOptions Será el formato que dará al índice del servidor.
HeaderName	Cabecera de nuestro índice.
ScriptAlias	Indica donde serán cargados por default los CGI's

⁶ Common Gateway Interface

AddHandler	Indica las extensiones posibles para los CGI's
AddType	Para agregar la opción de SSI's
AddOutputFilter	Agrega un filtro que contenga includes.
VirtuaHost	Permite agregar un servidor virtual. Posee las características elementales de nuestro servidor como Servername, ServerAdmin, etc.

Tabla 2.6 Directivas de Apache (Continuación)

Para poder manejar y controlar todos los servicios dentro de Apache Server de manera eficiente y segura, la primera de las recomendaciones es correr el demonio de Apache como un usuario normal, sin privilegios y nunca como root o anybody, ya que haríamos a nuestro servidor susceptible de ataques.

Una parte elemental para evitar ataques de intrusos es renombrar el archivo `Apachectl`. Aquí se indica nuestro `PidFile` para realizar los procesos. Si un usuario llegase a tener acceso a este archivo, podría dar de baja en cuestión de minutos todos los procesos o servicios que estén llevándose a cabo.

Para comenzar a utilizar Apache es necesario editar todas nuestras directivas dentro del archivo `httpd.conf`. Una vez editadas las diferentes directivas. Utilizamos el comando en Linux: `./apachectl start` para iniciar nuestro servicio. Utilizamos `./apachectl stop` para detener el proceso.

Hay ocasiones en que no guardamos adecuadamente los cambios realizados a nuestro archivo de configuración e intentamos guardar los cambios cuando el proceso aún esta en ejecución, esto ocasiona que nuestro servidor no encuentre el archivo de configuración tal y como lo inicio, por lo que detiene a un fantasma (proceso en ejecución dentro de Linux). Por consiguiente, el archivo con el que inicio en un principio no es el mismo con el que se desea terminar. Una vez que se intenta reiniciar apache (`./apachectl restart`), el programa envía un aviso de que el demonio ya está siendo ejecutado y que no puede ser reiniciado por que su archivo de configuración no concuerda.

En estas ocasiones tenemos que eliminar el proceso manualmente con el comando `pkill ###`, donde #, indica el número del proceso que esta corriendo el demonio de Apache. Para conocer dicho número se utiliza la opción `ps-fea | grep http`.

En el caso de las directivas relacionadas con las bitácoras, se recomienda que el administrador del servidor cree, de manera personal los archivos donde se almacenará ésta información (`acces.log` y `error.log`), así como la ubicación de éstos archivos.

En el caso de los hosts virtuales, se debe aclarar que hay que tener cuidado al momento de asignar la dirección IP y el puerto por el que "escuchara" ó se dará respuesta a las diferentes peticiones, ya que podría existir el error de que

dos direcciones “escuchen” en el mismo puerto, lo que provocaría una confusión a nuestro DNS en caso de encontrarse dentro de la misma red, ó bien, un error al momento de querer ingresar a cualquiera de las dos direcciones.

Por último, pero no por ello menos importante, se debe de mencionar la seguridad que debe llevar el uso de los scripts, ya que en muchos de los casos los “hackers” utilizan dichos medios para poder introducirse en el servidor y poder cambiar por completo su estabilidad, e incluso anular por completo el servicio.

3 Programación con Software Libre y Java



PHP fue originalmente diseñado en Perl, basado en la escritura de un grupo de CGI binarios escritos en lenguaje C por el programador Danés-Canadiense Rasmus Lerdorf en el año 1994 para mostrar su currículum vitae y guardar ciertos datos, como la cantidad de tráfico que su página web recibía. El 8 de junio de 1995 fue publicado "Personal Home Page Tools" luego de que Lerdorf lo combinara con su propio *Form Interpreter* para crear PHP/FI.

Dos programadores israelíes de Technion, Zeev Suraski y Andi Gutmas, reescribieron el analizador gramatical (*parser* en inglés) en el año 1997 y crearon la base de PHP 3, cambiando el nombre del lenguaje a la forma actual. Experimentos públicos de PHP 3 comenzaron inmediatamente y fue lanzado oficialmente en junio del 1998.

Para 1999, Suraski y Gutmans reescribieron el código de PHP, produciendo lo que hoy se conoce como Zend Engine o motor Zend. También conformaron Zend Technologies en Ratmat Gan, Israel. En mayo del 2000 PHP 4 fue lanzado bajo Zend Engine 1.0. El 13 de julio del 2004, PHP 5 fue lanzado, utilizando el motor Zend Engine II. La versión más reciente de PHP es la 5.1, que aún se encuentra en estado beta, que incluye el novedoso PDO (Objetos de Información de PHP o PHP Data Objects) y mejoras utilizando las ventajas que provee el nuevo Zend Engine 2.

El fácil uso y la similitud con los más comunes lenguajes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados programar aplicaciones complejas con un aprendizaje mínimo. También permite interactuar con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.

Debido al diseño de PHP, también es posible crear aplicaciones con una interfaz gráfica para el usuario o GUI, utilizando la PHP-GTK. También puede ser usado desde línea de comandos, como Perl o Python.

Su interpretación y ejecución se da en el servidor en el cual se encuentra almacenada la página y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, enriquecida con código PHP, el servidor interpretará las instrucciones mezcladas en el cuerpo de la página y las sustituirá con el resultado de la ejecución antes de enviar el resultado a la computadora del cliente.

PHP también permite la conexión a numerosas bases de datos de forma nativa tales como: MySQL, PostgreSQL, Oracle, IBM DB2, Microsoft SQL Server y SQLite, lo cual permite la creación de Aplicaciones web muy robustas.

Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como: UNIX, Linux, Windows y Mac OS X, y puede interactuar con los servidores de web más populares.

El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a ColdFusion de la compañía Macromedia, a JSP/Java de Sun Microsystems, y al famoso CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un compilador comercial denominado Zend Optimizer.

Los principales usos de PHP son los siguientes:

- Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, o el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola al estilo de Perl en Linux, Windows y Macintosh
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK¹, que permite desarrollar aplicaciones de escritorio para los diferentes sistemas operativos

Ventajas de PHP

- Capacidad de acceder la mayoría de las base de datos que se utilizan en la actualidad.
- Leer los datos desde diferentes fuentes, incluyendo datos que introduzcan los usuarios desde formularios HTML y manipularlos de forma sencilla.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos
 - Posee una muy buena documentación en su página.
 - Es libre, por lo que resulta una alternativa de fácil acceso para todos.
 - Permite las técnicas de Programación Orientada a Objetos.

Para comenzar a utilizar PHP debemos retomar algunos conceptos de los lenguajes de programación, la líneas a continuación serán nuestro pequeño repaso.

¹ [GIMP](#) Tool Kit

Sentencia: como en todo lenguaje de programación, PHP se compone de una serie de sentencias, éstas pueden ser una asignación, llamada a una función, a un ciclo o incluso a otra sentencia. Normalmente cada sentencia termina con un punto y coma.

Bloque de sentencias: es un grupo de cero o más sentencias encerradas entre llaves { }. Se puede insertar un bloque de sentencias en cualquier lugar.

Variables en PHP

PHP como el resto de los lenguajes maneja diferentes tipos de Variables y la forma de asignar dichas variables, varía con los demás. Aquí necesitamos anticipar el símbolo de \$ al nombre de la variable, seguido del signo = junto con el valor de la variable entre comillas. Así, la asignación de una variable se demuestra de la siguiente manera:

$$\$Var = \text{“nombre”}$$

Las variables pueden ser de diferente tipo, según el tipo de datos que manejará cada variable.

Tipo de Datos.

Enteros: son los números en general, con valores positivos, negativos e incluso números de diferentes bases como: octal y hexadecimal.

Flotantes (double): flotantes, son números que tienen un valor muy grande o muy pequeño con punto decimal.

Cadenas: son valores alfabéticos que a diferencia de los números requieren estar contenidas entre comillas para que puedan ser validados como un conjunto de caracteres.

3.1 Sentencias de Control.

Son las que nos permiten ejecutar diferentes bloques de código dependiendo de ciertas condiciones.

if: else la sentencia **if** permite llevar a cabo la ejecución de sentencias siempre evaluando una condición.

if

```
if (Condición){  
    esta sentencia se realiza si la condición resulta verdadera;}  
else{Esta es la sentencia que se lleva a cabo con un resultado falso;}  
}
```

Si al evaluar la expresión se obtiene un valor booleano **TRUE** se ejecutará la primer sentencia, en caso contrario se ejecutará la sentencia contenida después de *else*.

La sentencia **if** puede contener tantas sentencias como se requiera, así como otras sentencias **if**.

Switch: esta sentencia se utiliza para comparar una variable con un cierto grupo de valores. En pocas palabras son las diferentes opciones que puede llegar a tomar la variable. Cada valor posee un grupo diferente de sentencias que sólo se llevarán a cabo si la condición se cumple.

Switch

```
switch( variable ){
    case valor1:
        sentencias;
    break;
    case valor2:
        sentencias;
    break;
    case valorN:
        sentencias;
    break;
    default:
        sentencias;
}
```

La sentencia *break* al final de cada *case* transfiere el control al final de la sentencia **switch**; de esta manera, cada vez que se ejecuta un *case* todos los enunciados *case* restantes son ignorados y termina la operación del **switch**.

for: repite una sentencia, hasta que no se haya cumplido una condición.

For

```
for(inicialización; condición; incremento){
    sentencias;}
```

La inicialización es una sentencia de asignación simple que se utiliza para establecer una variable que controle el ciclo.

La condición comprueba la variable que controla el ciclo y determina los propios límites del mismo.

El incremento determina la manera en que cambiará la variable que controla el ciclo.

While: es muy similar a *for* y *do-while*. También lleva a cabo un ciclo que posee sus propias condiciones para que éste se realice.

While

```
while (condición){
    sentencias;}
```

do-while: la diferencia entre las anteriores sentencias y **do-while** es que ésta realiza el ciclo por lo menos una vez. Lo que **for** y **while** no hacen ya que primero realizan una evaluación de la condición.

Do

```
do{sentencias;}
while (condición);
```

break: se mencionó anteriormente en la sentencia **switch** pero también puede utilizarse dentro de una sentencia condicional para poder terminar el ciclo cuando uno lo requiera.

continue: esta sentencia es muy similar a la de **break** pero, a diferencia de la anterior ésta simplemente hace que la condición salte a la siguiente interacción del ciclo, sin que éste termine.

Funciones: bloque de código que se define una vez y puede ser ejecutado desde diferentes partes del programa.

Function

```
function nombreFuncion( parametros ){
    cuerpo de la función }
```

Alcance o ámbito de una variable

Una vez que hablamos de funciones debemos aclarar que al determinar una variable debemos de estar conscientes de donde la utilizaremos ya que no podremos utilizar una variable en diferentes funciones. Es muy similar a C cuando comenzamos a utilizar Programación Orientada a Objetos

Si quisiéramos utilizar variables que se encuentran fuera de nuestra función utilizamos **global**

Global

```
function nombre( ){
    global $var1, $var2, $var3;}
```

Constantes: las constantes son similares a las variables. La diferencia es que para que éstas sean tomadas como constantes, debemos anteponer **define()**, que significa que no podrán ser definidas en un futuro.

Un ejemplo de constantes son True y False, estas están previamente definidas por PHP, nosotros debemos de seguir la siguiente sintaxis.

Define

```
define("NOMBRE", Valor de la constante);
```

3.2 Arreglos

Un arreglo es una manera de ordenar los datos que manejamos por medio de PHP. Frecuentemente los programas que se realizan en PHP tienen interacción con bases de datos, esta es una buena manera para poder mostrar dichos datos de manera sencilla.

Arreglos

```
$arreglo = array("Primero", "Segundo", "Tercero");
```

```
$arreglo[0] = "Primero";  
$arreglo[1] = "Segundo";  
$arreglo[2] = "Tercero";
```

```
$arreglo[] = "Primero";  
$arreglo[] = "Segundo";  
$arreglo[] = "Tercero";
```

Estas son algunas funciones que podemos utilizar junto con los arreglos:

- **int sizeof:** regresa el número de elementos del arreglo.
- **mixed array_pop:** da el último elemento de nuestro arreglo, además de reducir el tamaño del arreglo.
- **int array_push:** maneja el arreglo como una pila y coloca las anteriores variables como un argumento al final del arreglo.
- **bool in_array:** busca un elemento dentro de un arreglo

3.3 Trabajando con cadenas

Strlen: indica la longitud de una cadena.

Echo: envía a la pantalla todos los argumentos contenidos en esta sentencia.

print: también envía a la pantalla los argumentos, pero a diferencia de **echo**, **print** solo puede enviar una sentencia.

printf: cambia el formato del texto que aparecerá en pantalla.

array **explode:** regresa un arreglo de cadenas, cada una de las cuales es una subcadena formada al romper una cadena original de acuerdo a cierto delimitador.

3.4 El Sistema de Archivos

fopen Abre un archivo o una URL. Si ocurre algún error o no encuentra el archivo, la función nos notificará con un false.

Estas son algunas de las opciones para abrir los archivos:

r: únicamente abrirá el archivo para que pueda ser leído, colocándonos al inicio del archivo.

r+: abrimos el archivo para la lectura y escritura del mismo

w: también abre el archivo para lectura y escritura, la diferencia es que si el archivo al que se hace referencia no existe, esta función crea dicho archivo.

a: también abre el archivo solo para lectura, la diferencia de r es que nos coloca al final del archivo.

a+: lectura y escritura y también posesiona el puntero al final del archivo, también lo crea si éste no existe.

int **f filesize:** indica el tamaño de un archivo

int **f eof:** indica si el puntero se encuentra localizado al final del archivo por medio de True.

string **f read:** lee el tamaño de un archivo en bytes.

int **f write:** inserta una cadena a un archivo, puede especificarse opcionalmente cual es la longitud de la cadena que se va a escribir.

int **f close:** cierra un archivo. Opción que hay que tener muy presente ya que puede quedar abierto el archivo y no podríamos tener acceso a este por que esta siendo utilizado por PHP.

3.5 Programación con Java

Java es un lenguaje de programación elaborado por Sun Microsystems que tiene como objetivo ser un lenguaje que pueda ser entendido por cualquier sistema operativo.

Java cuenta con las siguientes características.

- El propio lenguaje de programación
- La maquina virtual o JRE² que permite la portabilidad en ejecución.
- El API Java³.

Originalmente fue llamado OAK por Sun cuyo fin era la programación sobre grandes equipos de trabajo. En 1995 con el “bum” del Web, Sun decide comenzar la distribución de su plataforma para los diferentes sistemas operativos particulares como Windows, Linux o Mac

El lenguaje resulta muy similar a la programación en C++, pero su funcionamiento es más similar al de Smalltalk⁴ que a éste. Java resulta más versátil que sus predecesores por el manejo adecuado de la herencia, el manejo de librerías muy particulares que facilitan la programación. Con esta plataforma comienza el término de Programación Orientada a Eventos.

Durante la década pasada Java llegó a convertirse en el lenguaje más utilizado en los programas que brindaban servicios Web. Su éxito se basó en el manejo de la tecnología JSP que se parece mucho a ASP de Microsoft y a PHP. En esta “moda” surgió otra tecnología, Java Beans, que es una tecnología que involucra un ambiente gráfico en el mundo de Internet

Pero el éxito llegó cuando Sun introduce al mercado la especificación conocida como J2EE⁵. Este modelo permite una separación entre los datos presentados al usuario, el modelo de los mismos y control de éstos. Esta tecnología desencadenó lo que hoy se conoce como los “servidores de aplicaciones”.

Además de los programas servidor, con Java podemos realizar programas que contengan una interfaz gráfica atractiva al usuario. De igual forma podemos

² JRE Java Runtime Environment. Proporciona un subconjunto del lenguaje de programación Java, únicamente para ejecución.

³ Biblioteca estándar para el lenguaje

⁴ Antiguo lenguaje de programación elaborado por Xerox en 1980

⁵ Java 2 Enterprise Edition

utilizar algunos elementos para insertar parte del código en los navegadores Web en forma de Applets, que no fueron muy bien acogidos, o bien como Scripts para la validación de un formulario dentro de una página de Internet

Los programas en Java generalmente son compilados a un lenguaje intermedio llamado bytecode, que luego son interpretados por una máquina virtual. Esta última sirve como una plataforma de abstracción entre la máquina y el lenguaje permitiendo que se pueda "escribir el programa una vez, y correrlo en cualquier lado".

3.6 Programando en Java

Sentencias de Control de Flujo: en Java también existen las sentencias de control de flujo para indicar el orden de las demás sentencias. Éstas son muy similares o casi idénticas al lenguaje C o PHP, únicamente cambia la sintaxis.

Sentencia if-else: La sentencia if-else es la misma en todos los lenguajes, si se lleva a cabo una condición entonces se podrá realizar algún proceso, de lo contrario, se llevará a cabo otro. Esta es la sintaxis correcta para poder utilizar if-else dentro de Java.

```
if (expresión)
    sentencia
```

A continuación se demuestra un pequeño ejemplo de cómo podría ser utilizado if-else:

```
int puntuacion;
String nota;

if (puntuacion >= 90) {
    nota = "Sobresaliente";
} else if (puntuacion >= 80) {
    nota = "Notable";
} else if (puntuacion >= 70) {
    nota = "Bien";
} else if (puntuacion >= 60) {
    nota = "Suficiente";
} else {
    nota = "Insuficiente";
}
```

Es un ejemplo donde declaramos una variable entera; "puntuación", que puede llegar a tener cualquier valor. Según el valor que ésta tenga se podrá utilizar la variable cadena "nota" a la que se le asignará un valor según corresponda al valor de "puntuación".

Sentencia Switch: switch es muy similar a if-else, la diferencia radica en que la primera se utiliza para varias sentencias que se basan en una expresión. Esto es un ejemplo de cómo puede ser utilizada la sentencia Switch.

```
int mes;
switch (mes) {
case 1: System.out.println("Enero"); break;
case 2: System.out.println("Febrero"); break;
case 3: System.out.println("Marzo"); break;
case 4: System.out.println("Abril"); break;
case 5: System.out.println("Mayo"); break;
case 6: System.out.println("Junio"); break;
case 7: System.out.println("Julio"); break;
case 8: System.out.println("Agosto"); break;
case 9: System.out.println("Septiembre"); break;
case 10: System.out.println("Octubre"); break;
case 11: System.out.println("Noviembre"); break;
case 12: System.out.println("Diciembre"); break;
}
```

Aquí se presenta una variable entera, “mes” la cual será la expresión a evaluar. Case es parte de la sintaxis de switch; indica que sentencia se llevará a cabo tras comparar el valor de nuestra variable.

Break aparece para poder salir de la sentencia switch y poder continuar leyendo el resto del programa.

Aquí también podría aparecer la sentencia DEFAULT que se utiliza cuando no hay una posible “respuesta” a la sentencia switch y se lleva a cabo una instrucción previamente determinada.

Sentencias de Bucle: estas sentencias son también conocidas como de repetición. Esta lista empieza con While, que lleva a cabo una sentencia mientras se cumpla una condición específica.

```
while (expresión)
    sentencia
```

While sólo se llevará a cabo si la condición es verdadera, de lo contrario la sentencia que esta contenida dentro de while no se llevará a cabo

También puede utilizarse la sentencia for, que se recomienda cuando conocemos el inicio, fin y la manera en que se llevará a cabo su incremento. Esta es la sintaxis general de for.

```
for (inicialización; terminación; incremento) sentencias
```

La sentencia se detendrá cuando la variable llegue a resultar falsa, en pocas palabras, cuando cumpla su ciclo.

Do-while es otra variante de los bucles que indica que se llevará a cabo una sentencia mientras el valor de la variable cumpla una condición. Es muy similar a while, a diferencia de éste, do while se utiliza cuando deseamos que el ciclo se lleve a cabo por lo menos una sola vez.

```
do {
    sentencias
} while (Expresión Booleana);
```

Sentencias de Manejo de Excepciones: este es un punto interesante bajo la plataforma de Java. Este tipo de sentencias se utilizan cuando ocurre un error en alguna parte de nuestro programa. Java nos permite poder manejar y hasta cierto punto “controlar” los errores que pueden llegar a ocurrir en la ejecución del programa. Throw, try, catch y finally son las sentencias que podemos usar para el manejo de los diferentes errores.

Sentencias de Ruptura: en Java se pueden realizar sentencias de instrucciones en cualquier parte de nuestro código, tan solo se requiere utilizar el nombre del indicador seguido de dos puntos. El programa efectuará la instrucción únicamente cuando se le indique con la ayuda de break

```
SaltaAqui: algunaSentenciaJava
break SaltaAqui;
```

Las rupturas en este tipo de etiquetas, simulan a la función goto que no es soportado por la plataforma de Java.

Otra sentencia es continue, se utiliza dentro de un bucle, ya que no puede ser utilizado en otra parte. Su uso es para saltar de una sentencia dentro del ciclo al principio de éste o a una sentencia etiquetada. Este es un ejemplo de ello.

```
public int indexOf(String str, int fromIndex) {
    char[] v1 = value;
    char[] v2 = str.value;
    int max = offset + (count - str.count);
    test:
    for (int i = offset + ((fromIndex < 0) ? 0 : fromIndex); i <=
max ; i++) {
        int n = str.count;
        int j = i;
        int k = str.offset;
        while (n-- != 0) {
            if (v1[j++] != v2[k++]) {
                continue test;
            }
        }
        return i - offset;
    }
    return -1;
}
```

Finalmente, la sentencia return, se utiliza para salir de un método y volver a la siguiente sentencia de la que se originó la llamada inicial. Return tiene dos formas, una en la que entrega un resultado y otra que no lo hace. Lo hace dependiendo del tipo del valor de retorno del método. Cuando un método es declarado como void y utilizamos return, este return no devolverá ningún valor

Arrays

Para poder utilizar un array dentro de Java, debemos declararlo como lo hacemos con una variable. Esto es el tipo de array, los tipos de datos que llevará el array, y el nombre del mismo. Por consiguiente la declaración de un array sería la siguiente:

```
int[ ] arrayDeEnteros;
```

Pero este arreglo no cuenta con un espacio asignado en memoria para que pueda contener algún valor. Para ello se debe ejemplificar el array, utilizando el operador new seguido del tipo de dato y la extensión que llevará el arreglo.

```
int[ ] arraydeenteros = new int[10]
```

Aquí se declara un arreglo de tipo entero con capacidad para 10 elementos.

Una vez determinado el tamaño y tipo del arreglo podemos insertar elementos dentro del mismo, así como recuperar estos mismos elementos. El siguiente es un ejemplo de lo antes mencionado.

```
for (int j = 0; j < arrayDeEnteros.length; j ++) {  
    arrayDeEnteros[j] = j;  
    System.out.println("[j] = " + arrayDeEnteros[j]);  
}
```

Aquí el uso de length seguido de arrayDeEnteros significa que java esta pidiendo el tamaño del array, como esta instrucción se encuentra dentro de una sentencia for. Lo que se pretende con esto es tener como límite de nuestro ciclo el tamaño de nuestro array.

```
String[ ] arrayDeStrings = new String[10];  
for (int i = 0; i < arraydeStrings.length; i ++) {  
    arraydeStrings[i] = new String("Hello " + i);  
}
```

El ejemplo anterior es un array de tipo cadena. Se debe mencionar que si se desea tener acceso a alguno de los datos después de haber creado el arreglo. Java nos devolverá el siguiente error: NullPointerException que significa que no existen datos dentro de nuestro arreglo. Para ello se requiere asignar valores conforme se vaya leyendo el arreglo con ayuda del ciclo for, tal y como se mostró con anterioridad.

Clases: una clase es una parte muy importante dentro de Java, ya que todo lo que pueda ser creado en Java deberá de estar contenido dentro de una de éstas, ó que describa alguna función.

Las diferentes instrucciones se colocan dentro de la clase u objeto, así como todos los métodos, ya que en Java no se acostumbra mucho el manejo de variables globales. Por estas razones surgen las clases, que terminan siendo objetos.

Tipos de Clases

Existen varios tipos de clases, lo que hacen que cada clase tenga su propia característica.

Public: este tipo de clases son accesibles para cualquier otra clase por medio de la herencia o porque se encuentran en la misma carpeta de archivos.

Abstract: una clase de tipo abstract quiere decir que es el inicio de una herencia. Por lo que ésta no puede hacer referencia a ninguna otra clase.

Final: esta clase referencia a la clase final, cuando se están manejando clases con herencia. Por consiguiente ésta no puede hacer referencia a otra clase final.

Synchronizable: Este tipo de clases se utilizan para poder sincronizar las diferentes variables que contenga para evitar la sobre escritura de las mismas por medio de banderas indicadoras en la memoria del sistema.

Métodos y Constructores.

Los métodos son funciones que pueden ser llamadas dentro de la clase o por otras clases. El constructor es un tipo específico de método que siempre tiene el mismo nombre que la clase.

Cuando se declara una clase en Java, se pueden declarar uno o más constructores opcionales que realizan la inicialización cuando se hace una referencia a un objeto de dicha clase.

Cuando utilizamos constructores debemos primero realizar una instancia a la clase principal, para ello utilizamos la palabra clave new. La memoria antes de hacer referencia a la anterior clase no alberga dato alguno. No es hasta que se lleva a cabo la instancia cuando ésta ocupa un espacio en memoria. Asignando el valor que previamente poseía el constructor en la clase a la que se hace la referencia.

Una vez realizada la instancia podemos tomar de la anterior clase uno de los métodos con la siguiente sintáxis

```
nombre_clase.nombre_constructor
```

Donde el nombre de la clase es la “nueva” clase, a instancia de una anterior. El punto significa que hará referencia a un constructor de otra clase; seguido de este aparece el nombre del constructor.

Herencia

La Herencia es el mecanismo por el que se crean nuevos objetos definidos en términos de objetos ya existentes. Por ejemplo, si se tiene la clase Ave, se puede crear la subclase Pato, que es una especialización de Ave.

```
class Pato extends Ave {  
    int numero_de_patas;  
}
```

La palabra clave `extends` se usa para generar una subclase de un objeto. Una Pato es una subclase de Ave. Cualquier cosa que contenga la definición de Ave será copiada a la clase Pato, además, en Pato se pueden definir sus propios métodos y variables de instancia. Se dice que Pato deriva o hereda de Ave.

Además, se pueden sustituir los métodos proporcionados por la clase base. Esto es que podemos albergar un constructor ya existente en la clase a la que se hace la referencia pero que aquí puede reiniciar o no.

En Java no se puede hacer herencia múltiple. Por ejemplo, de una clase aparato con motor y de la clase animal no se puede derivar nada. En realidad, lo que se pretende es copiar los métodos, es decir, pasar las funciones de una clase a otra, con lo cual no sería necesario la herencia múltiple sino simplemente compartir la funcionalidad que se encuentra implementada en Java a través de interfaces.

Control de Acceso a Clases

Cuando se crea una nueva clase en Java, se puede especificar el nivel de acceso que se quiere para las variables de instancia y los métodos definidos en la clase:

Public: cualquier clase desde cualquier lugar puede tener acceso a las variables y a los diferentes métodos de instancia públicos.

Protected: únicamente las subclases pueden tener acceso a las variables y métodos.

Private: tan solo puede tener acceso la misma clase que lo contenga, nadie más puede hacerlo.

Friendly: esta es la opción por default que significa que puede ser usado por las demás clases en todo nuestro código. Si deseamos que nuestra clase sea friendly no hace falta escribir la palabra clave antes del nombre de la clase.

Clases Abstractas

Una de las características más útiles de cualquier lenguaje orientado a objetos es la posibilidad de declarar clases que definen como serán utilizadas por si solas, sin tener que implementar métodos. Esto es muy útil cuando la implementación es particular para cada usuario, pero todos los usuarios tienen que utilizar los mismos métodos. Un ejemplo de clase abstracta en Java es la clase Graphics:

```
public abstract class Graphics {
    public abstract void drawLine( int x1,int y1,int x2,
        int y2 );
    public abstract void drawOval( int x,int y,int width,
        int height );
    public abstract void drawArc( int x,int y,int width,
        int height,int startAngle,int arcAngle );
}
```

Los métodos se declaran en la clase Graphics, pero el código que llevará a cabo el método está en algún otro sitio, tal y como se muestra a continuación:

```
public class MiClase extends Graphics {
    public void drawLine( int x1,int y1,int x2,int y2 ) {
        <código para pintar líneas -específico de
        la arquitectura->
    }}
}
```

Cuando una clase contiene un método abstracto, la clase tiene que declararse abstracta. No obstante, no todos los métodos de una clase abstracta tienen que ser abstractos. Las clases abstractas no pueden tener métodos privados ya que no se podrían implementar, ni mucho menos ser estáticos. Una clase abstracta tiene que derivarse obligatoriamente, no se puede hacer una "nueva" clase de una abstracta.

Interfaces

Los métodos abstractos son útiles cuando se quiere que cada implementación de la clase funcione igual, pero necesita que se cree una nueva clase para utilizar los métodos abstractos.

Las interfaces proporcionan un mecanismo para extraer los métodos a un nivel superior.

Una interfaz contiene una colección de métodos que se implementan en otro lugar. Los métodos de una clase son public, static y final.

La principal diferencia entre interface y abstract es que un interface proporciona un mecanismo de encapsulación de los protocolos de los métodos sin necesidad de utilizar la herencia.

Por ejemplo:

```
public interface VideoClip {
    // comienza la reproducción del video
    void play();
    // reproduce el clip en un bucle
    void bucle();
    // detiene la reproducción
    void stop();
}
```

Las clases que quieran utilizar el interface VideoClip utilizarán la palabra implements y proporcionarán el código necesario para poder implementar los métodos que se han definido para la interface:

```
class MiClase implements VideoClip {
    void play() {
        <código>
    }
    void bucle() {
        <código>
    }
    void stop() {
        <código>
    }
}
```

Al utilizar implements para una interfaz es como si se hiciese una acción de copiar y pegar el código del interface, con lo cual no se hereda nada, solamente se utilizan los diferentes métodos que ya están contenidos en la interfaz.

La ventaja principal del uso de interfaces es que una clase interface puede ser implementada por cualquier número de clases, permitiendo a cada clase compartir la interfaz de programación sin tener que ser consciente de la implementación que hagan las otras clases que implementen la interface.

```
class MiOtraClase implements VideoClip {
    void play() {
        <código nuevo>
    }
    void bucle() {
        <código nuevo>
    }
    void stop() {
        <código nuevo>
    }
}
```

Paquetes

La palabra clave `package` permite agrupar clases e interfaces. Los nombres de los paquetes son palabras separadas por puntos y se almacenan en directorios que coinciden con esos nombres.

Por ejemplo, los siguientes ficheros, que contienen código fuente Java: `Applet.java`, `AppletContext.java`, `AppletStub.java`, `AudioClip.java` Contienen en su código la línea:

```
package java.applet;
```

Y las clases que se obtienen de la compilación de los ficheros anteriores, se encuentran con el nombre `nombre.class`, en el directorio:

```
java/applet
```

Import

Los paquetes de clases se cargan con la palabra clave `import`, especificando el nombre del paquete como ruta y nombre de clase (es lo mismo que `#include` de C/C++). Se pueden cargar varias clases utilizando un asterisco.

```
import java.Date;
import java.awt.*;
```

Si un fichero fuente Java no contiene ningún `package`, se coloca en el paquete por defecto sin nombre. Es decir, en el mismo directorio que el fichero fuente, y la clase puede ser cargada con la sentencia `import`:

```
import MiClase;
```

Paquetes de Java

El lenguaje Java proporciona una serie de paquetes que incluyen ventanas, utilidades, un sistema de entrada/salida general, herramientas y comunicaciones. En la versión actual del JDK, los paquetes Java que se incluyen son:

java.applet: este paquete contiene clases diseñadas para usarse con applets. Hay una clase `Applet` y tres interfaces: `AppletContext`, `AppletStub` y `AudioClip`.

java.awt: el paquete Abstract Windowing Toolkit (`awt`) contiene clases para generar widgets y componentes GUI (Interfaz Gráfico de Usuario). Incluye las clases `Button`, `Checkbox`, `Choice`, `Component`, `Graphics`, `Menu`, `Panel`, `TextArea` y `TextField`.

java.io: el paquete de entrada/salida contiene las clases de acceso a ficheros: `FileInputStream` y `FileOutputStream`.

java.lang: este paquete incluye las clases del lenguaje Java propiamente dicho: Object, Thread, Exception, System, Integer, Float, Math, String, etc.

java.net: este paquete da soporte a las conexiones del protocolo TCP/IP y, además, incluye las clases Socket, URL y URLConnection.

java.util: este paquete es una miscelánea de clases útiles para muchas cosas en programación. Se incluyen, entre otras: Date (fecha), Dictionary (diccionario), Random (números aleatorios) y Stack (pila FIFO).

4 Bases de Datos MySQL



MySQL es un manejador de bases de datos que se distribuye de manera gratuita a través de la red. Sus características se asemejan mucho al estándar SQL¹.

Su uso comenzó a popularizarse por lo sencillo que resulta el manejar las diferentes bases de datos. Esto sin contar que inicialmente MySQL carecía de ciertas características de SQL. Pero con el paso del tiempo le fueron agregando más y más características lo cual hizo aumentar mucho más su popularidad.

Las características de MySQL podrían ser estas:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones también son incluidas.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según se desee: velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Tipos de tablas en MySQL

1. **ISAM**. Este formato es uno de los más antiguos, por lo tanto presenta ciertas limitaciones como transporte y capacidad de almacenamiento.

2. **MYISAM**. Este formato es el que MySQL maneja por default. Resulta muy práctico porque puede trabajar con sistemas operativos antiguos. Además ofrece una facilidad para transportar los datos hacia otras plataformas.

3. **HEAP** Este formato es para tablas temporales.

4. **BDB**. Formato Berkeley.²

5. **INNODB**. Este formato esta orientado a la seguridad, maneja la recuperación de errores, bloquea a nivel fila, para evitar cambiar los diferentes datos.

6. **MERGE**. Este formato es para reducir y por consiguiente facilitar ciertas funciones en tablas de un tamaño considerable.

DBD e INNODB manejan lo que se conoce como TST³ que es un manejo de tablas de manera segura. Por lo que resultan mucho más lentas, además de ocupar mucha más memoria.

4.1 Tipos de Datos en MySQL

Al igual que en un lenguaje de programación MySQL debe definir los tipos de datos que manejará en las diferentes tablas. En general únicamente existen dos tipos de datos: los números y las cadenas. Sin embargo también existen otros tipos de datos pero en menor medida como son los formatos de fecha.

Datos Numéricos

Únicamente maneja los diferentes números, desde los enteros hasta números hexadecimales.

En general, los datos numéricos tienen cierto tamaño, pero éstos pueden modificarse con ciertas características.

BOOL: tan solo acepta 0 ó 1.

INT: con esto podemos almacenar números enteros.

FLOAT: representa números decimales.

¹ Structed Query Language. Lenguaje de Consulta Estructurado. Lenguaje estándar para acceder a los datos almacenados en una base de datos.

² Base de datos sencilla que permite enlaces a lenguajes de programación. Se distingue por que soporta múltiples datos bajo una misma clave.

³ Transaction Safe Tables

DOUBLE: número de tipo flotante de precisión doble. Es un tipo de datos igual al anterior cuya única diferencia es el rango numérico que abarca

Caracteres o Cadenas

CHAR: este tipo se utiliza para almacenar cadenas de longitud fija.

VARCHAR: al igual que el anterior se utiliza para almacenar cadenas, pero en este caso, de longitud variable.

La diferencia entre estos dos es que CHAR ocupará siempre el máximo de longitud que se le haya asignado, aunque el tamaño del dato sea menor. Mientras que VARCHAR solo almacena la longitud del dato, permitiendo reducir el tamaño de la base de datos. Pero el acceso a los datos CHAR es más rápido que VARCHAR.

No pueden alternarse columnas CHAR y VARCHAR en la misma tabla. En estos casos MySQL cambiará las columnas CHAR a VARCHAR.

TEXT y **BLOB:** también maneja cadenas, la diferencia es que Text no distingue entre mayúsculas y minúsculas pero Blob si lo hace.

Varios

DATE: para almacenar fechas. El formato por defecto es YYYY MM DD.

DATETIME: con este podemos combinar el formato de fecha junto al de hora.

TIME: este campo almacena la hora con el siguiente formato: HH/MM/SS

YEAR: Almacena un año. Estos valores pueden ser almacenados en cifras de 4 ó de 2 dígitos según sea el tipo de base.

4.2 Creación de Bases de Datos y Tablas

CREATE DATABASE nombre_db: es el comando que se utiliza para, como su nombre lo indica, crear una nueva base de datos. Si existiera una base de datos previa con el mismo nombre, MySQL nos enviará un mensaje de que ya existe dicha tabla.

La creación de la tabla puede tener otras opciones adicionales:

```
CREATE TABLE nombre_tbl  
(n_columna1 tipo(v_numérico),  
n_columna2 tipo(v_numérico),  
n_columna3 tipo(v_numérico),  
);
```

En esta sintaxis podemos observar que al momento de crear nuestra tabla podemos ir insertando el nombre de las columnas de la tabla, así como el tipo de dato que manejará, junto al tamaño del dato.

Al igual que en una sentencia de cualquier lenguaje de programación, todas las instrucciones deben de concluir con punto y coma. En MySQL si llegamos a dar un enter sin querer, se entenderá que la instrucción continua, con el fin de evitar escribir instrucciones demasiado grandes.

Un buen diseño de la tabla determinará el éxito o fracaso de nuestra base de datos. A un nivel básico se puede indicar que la sola elección del tipo de tabla adecuado, del tipo y longitud de datos de cada columna tendrá su reflejo en la rapidez y eficacia del sistema.

Algunas reglas a recordar:

- Longitud es opcional SALVO para los campos tipo DECIMAL NUMERIC CHAR y VARCHAR.

- *Decimal:* Si el número decimal a almacenar tiene demasiados dígitos después del punto, éste será redondeado.

- El atributo BINARY puede usarse con CHAR y VARCHAR. En el momento en que se realice una búsqueda distinguirá mayúsculas y minúsculas.

- ZEROFILL solo puede emplearse con datos tipo numérico.

- UNSIGNED solo puede emplearse con datos del tipo numérico entero.

- Cada columna puede ser NULL o NOT NULL. Si no lo especificamos se asumirá como una columna NULL

- Las columnas, salvo que sean AUTO_INCREMENT, siempre tienen un valor por defecto. Puede establecerse un valor por defecto con DEFAULT.

- Únicamente puede existir una columna AUTO_INCREMENT por tabla, que debe ser del tipo entero.

- La columna AUTO_INCREMENT siempre será considerada como índice primario (PRIMARY KEY).

Indices En MySQL los índices se llaman KEY o INDEX. En principio se puede indexar cualquier columna, sea cual sea su tipo, aunque algunas son más idóneas que otras.

PRIMARY KEY es un índice sobre una columna NOT NULL y UNIQUE, es decir que la columna no puede contener valores vacíos o duplicados. Generalmente se debe de tener una Primary Key por tabla pero dentro de la base de datos pueden existir las que se requieran.

4.3 Insertar Filas

INSERT: nos permite insertar una fila dentro de una tabla ya creada con anterioridad. En la forma más simple de esta sentencia debemos indicar la tabla a la que queremos añadir filas, y los valores de cada columna. Las columnas de tipo cadena o fecha deben estar entre comillas sencillas o dobles. Para las columnas numéricas esto no es imprescindible, aunque también pueden estar entre comillada.

```
mysql> INSERT INTO gente VALUES
-> ('Claudia','1984-08-21'),
-> (Mariana,'1996-05-29');
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

Los últimos dos renglones del ejemplo anterior indican que las filan han sido agregadas correctamente, si hay duplicidad ó no dentro de la tabla; o si hubo algún error.

Podemos asignar un valor por default en nuestra tabla, para ello primero debemos asignar dicho valor. Una vez hecho esto, bastará con escribir DEFAULT en el campo del valor para que éste sea asignado. Ejemplo:

```
mysql> INSERT INTO ciudad2 VALUES ('Perillo', DEFAULT);
Query OK, 1 row affected (0.03 sec)
```

Otra opción consiste en indicar una lista de columnas para las que se van a suministrar valores. A las columnas que no se nombren en dicha lista se le asigna el valor por defecto. Este sistema, además, permite usar cualquier orden en las columnas, con la ventaja de que no necesitamos conocer el orden de las columnas en la tabla para poder insertar datos.

4.4 Selección de Datos

SELECT es la instrucción para poder obtener los datos de una tabla ya creada. SELECT puede resultar bastante complejo, y más cuando deseamos mezclar datos de diferentes tablas. En general su sintaxis es la siguiente:

```
SELECT [ALL | DISTINCT | DISTINCTROW]
FROM referencia_de_tablas
WHERE condiciones
[GROUP BY {nombre_col | expresion | posicion}
[ASC | DESC], ... [WITH ROLLUP]]
```

```

[HAVING condiciones]
[ORDER BY {nombre_col | expresion | posicion}
  [ASC | DESC] ,...]
[LIMIT {[desplazamiento,] contador | contador OFFSET
desplazamiento}]

```

Forma incondicional

La forma más sencilla consiste en pedir todas las columnas sin especificar condiciones.

```

mysql>mysql> SELECT * FROM gente;
+-----+-----+
| nombre | fecha      |
+-----+-----+
| Luis   | 1985-04-12 |
| Mariana | 1978-06-15 |
| Alfredo | 2001-12-02 |
| Karla  | 1993-02-10 |
+-----+-----+
4 rows in set (0.00 sec)
mysql>

```

Proyección de las Columnas

Recordemos que una de las operaciones del álgebra relacional era la proyección, que consiste en seleccionar determinados atributos de una relación. MySQL también utiliza este procedimiento para realizar algunas consultas.

Mediante SELECT es posible hacer una proyección de una tabla, seleccionando las columnas de las que queremos obtener datos. En la sintaxis que hemos mostrado, la selección de columnas corresponde a la parte "expresion_select". En el ejemplo anterior hemos usado el asterisco, que quiere decir que se muestran todas las columnas.

Pero podemos usar una lista de columnas, y de ese modo sólo se mostrarán las que deseemos

```

mysql> SELECT nombre FROM gente;
+-----+
| nombre |
+-----+
| Luis   |
| Mariana |
| Alfredo |
| Karla  |
+-----+
4 rows in set (0.00 sec)

```

Las “expresiones_select” no se limitan a nombres de columnas de tablas. Pueden ser otras expresiones, aunque éstas no correspondan a ninguna tabla:

```
mysql> SELECT SIN(3.1416/2), 3+5, 7*4;
+-----+-----+-----+
| SIN(3.1416/2) | 3+5 | 7*4 |
+-----+-----+-----+
| 0.999999999999325 | 8 | 28 |
+-----+-----+-----+
1 row in set (0.00 sec)
mysql>
```

Vemos que podemos usar diferentes tipos de funciones, en este ejemplo hemos usando la función SIN para calcular el seno de $\pi/2$.

Consultas multitabla

Hasta ahora todas las consultas que hemos usado se refieren sólo a una tabla, pero también es posible hacer consultas usando varias tablas con la misma sentencia.

Esto nos permite realizar otras dos operaciones de álgebra relacional que aún no hemos visto: el producto cartesiano y la composición.

Producto cartesiano

El producto cartesiano es simplemente combinar datos de varias tablas. A continuación se verá un ejemplo.

```
mysql> INSERT INTO personas(nombre,f_nacimiento) VALUES
-> ("Mariana", "1996-05-29"),
-> ("Claudia", "1984-08-21"),
-> ("Luis", "1985-08-16"),
-> ("Alfredo", "1976-08-25");
Query OK, 4 rows affected (0.09 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

Esta es la tabla que obtenemos como resultado con la anterior instrucción. Previamente ha sido creada la tabla de personas. Con la anterior instrucción únicamente insertamos un grupo de datos.

```
mysql> SELECT * FROM personas2;
+-----+-----+-----+
| id | nombre | fecha |
+-----+-----+-----+
| 1 | Mariana | 1996-05-29 |
| 2 | Claudia | 1984-08-21 |
| 3 | Luis | 1985-08-16 |
| 4 | Alfredo | 1976-08-25 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql>
```

Ahora insertaremos datos en la tabla de *teléfonos*, que también ya ha sido creada con anterioridad.

```
mysql> INSERT INTO telefonos2 (id, numero) VALUES
-> (1, "55 25 33 12"),
-> (1, "57 77 22 11"),
-> (2, "57 69 48 54"),
-> (2, "21 36 48 79"),
-> (3, "57 15 32 55"),
-> (4, "30 70 68 05");
Query OK, 6 rows affected (0.06 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

A continuación se muestra la tabla que acaba de ser creada.

```
mysql> SELECT * FROM telefonos2;
+-----+-----+
| numero | persona |
+-----+-----+
| 55253312 | 1 |
| 57772211 | 1 |
| 57694854 | 2 |
| 21364879 | 2 |
| 57153255 | 3 |
| 30706805 | 4 |
+-----+-----+
6 rows in set (0.00 sec)
mysql>
```

El producto cartesiano de dos tablas es la combinación de todas las filas de ambas tablas. Usando SELECT se representan los atributos de ambas tablas. Los nombres de las tablas se indican en la cláusula FROM separados con comas:

```
mysql> SELECT * FROM personas,telefonos;
+-----+-----+-----+-----+-----+
| id | nombre | fecha | numero | id |
+-----+-----+-----+-----+-----+
| 1 | Mariana | 1996-05-29 | 55253312 | 1 |
| 2 | Claudia | 1984-08-21 | 55253312 | 1 |
| 3 | Luis | 1985-08-16 | 55253312 | 1 |
| 4 | Alfredo | 1976-08-25 | 55253312 | 1 |
| 1 | Mariana | 1996-05-29 | 57772211 | 1 |
| 2 | Claudia | 1984-08-21 | 57772211 | 1 |
| 3 | Luis | 1985-08-16 | 57772211 | 1 |
| 4 | Alfredo | 1976-08-25 | 57772211 | 1 |
| 1 | Mariana | 1996-05-29 | 57694854 | 2 |
| 2 | Claudia | 1984-08-21 | 57694854 | 2 |
| 3 | Luis | 1985-08-16 | 57694854 | 2 |
| 4 | Alfredo | 1976-08-25 | 57694854 | 2 |
| 1 | Mariana | 1996-05-29 | 21364879 | 2 |
| 2 | Claudia | 1984-08-21 | 21364879 | 2 |
```

3	Luis	1985-08-16	21364879	2
4	Alfredo	1976-08-25	21364879	2
1	Mariana	1996-05-29	21364879	3
2	Claudia	1954-08-21	21364879	3
3	Luis	1985-08-16	21364879	3
4	Alfredo	1976-08-25	21364879	3
1	Mariana	1996-05-29	30706805	4
2	Claudia	1984-08-21	30706805	4
3	Luis	1985-08-16	30706805	4
4	Alfredo	1976-08-25	30706805	4

```

+-----+-----+-----+-----+-----+
24 rows in set (0.73 sec)
mysql>

```

Como se ve, la salida consiste en todas las combinaciones de todas las tuplas de ambas tablas.

Composición (Join)

Recordemos que se trata de un producto cartesiano restringido, las tuplas que se emparejan deben cumplir una determinada condición. Esto es muy útil cuando realizamos búsquedas dentro de nuestra base de datos.

En el álgebra relacional tan sólo se habla de composiciones en general. Sin embargo, en SQL se trabaja con varios tipos de composiciones.

Composiciones internas

Una composición interna, a diferencia del producto cartesiano, restringe los datos. Todo ésto se lleva a cabo por medio de una condición inicial para evitar que los datos se mezclen.

En el ejemplo anterior tenemos 24 resultados procedentes del producto cartesiano de las tablas *personas* y *teléfonos*. Si la condición para la composición es que "personas.id = telefonos.id", tendremos que eliminar todos los datos en que la condición no se cumpla.

Estas composiciones se denominan internas porque en la salida no aparece ningún resultado que no esté presente en el producto cartesiano, es decir, la composición se hace en el *interior* del producto cartesiano de las tablas.

Las composiciones internas usan estas sintaxis:

```

referencia_tabla, referencia_tabla
referencia_tabla [INNER | CROSS] JOIN referencia_tabla [condición]

```

La condición puede ser:

```
ON expresión_condicional | USING (lista_columnas)
```

La coma y *JOIN* son equivalentes, y las palabras *INNER* y *CROSS* son opcionales.

La condición en la cláusula *ON* puede ser cualquier expresión válida para una cláusula *WHERE*, de hecho, en la mayoría de los casos, son equivalentes.

La cláusula *USING* nos permite usar una lista de atributos que deben ser iguales en las dos tablas a componer.

Siguiendo con el mismo ejemplo, la condición más lógica para la composición interna entre *personas* y *teléfonos* es la igualdad entre el identificador de persona en la primera tabla y el atributo persona en la segunda:

```
mysql> SELECT * FROM personas, telefonos
-> WHERE personas2.id=telefonos2.id;
+-----+-----+-----+-----+-----+
| id | nombre | fecha | numero | id |
+-----+-----+-----+-----+-----+
| 1 | Mariana | 1996-05-29 | 55253212 | 1 |
| 1 | Mariana | 1996-05-29 | 57772112 | 1 |
| 1 | Claudia | 1984-08-21 | 57694854 | 2 |
| 2 | Claudia | 1984-08-21 | 21364879 | 2 |
| 3 | Luis | 1985-08-16 | 57153255 | 3 |
| 4 | Alfredo | 1976-08-25 | 30706805 | 4 |
+-----+-----+-----+-----+-----+
6 rows in set (0.73 sec)
mysql>
```

En cualquier caso, la salida sólo contiene los datos que cumplen la condición en la que las personas se relacionan directamente con sus números de teléfono. Los datos correspondientes a personas que no tienen ningún número no aparecen. Para las personas con varios números se repiten los datos de la persona para cada número, por ejemplo "Mariana " y "Claudia".

Composición interna natural

Consiste en una proyección sobre un producto cartesiano restringido. Es decir, sólo elegimos determinadas columnas de ambas tablas, en lugar de seleccionar todas.

Podemos hacer esto a partir de una composición general, eligiendo todas las columnas con excepción de las repetidas:

```
mysql> SELECT personas.id,nombre,fecha,numero
-> FROM personas, telefonos
-> WHERE personas.id=telefonos.id;
```

```

+-----+-----+-----+-----+
| id | nombre | fecha | numero |
+-----+-----+-----+-----+
| 1 | Mariana | 1996-05-29 | 55253212 |
| 1 | Mariana | 1996-05-29 | 57772212 |
| 1 | Fulanito | 1996-05-29 | 57692489 |
| 2 | Claudia | 1984-08-21 | 21364749 |
| 3 | Luis | 1985-08-16 | 57153255 |
| 4 | Alfredo | 1976-08-25 | 30706805 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
mysql>

```

Como la columna *id* existe en ambas tablas estamos obligados a usar el nombre completo para esta columna. En este caso hemos optado por “personas.id”, pero hubiese sido igual usar “telefonos.id”.

También podemos definir alias para las tablas, y conseguir una consulta más compacta:

```

mysql> SELECT t1.id,nombre,fecha,numero
-> FROM personas AS t1, telefonos AS t2
-> WHERE t1.id=t2.id;

```

Por supuesto, podemos usar *JOIN* y *ON* en lugar de la coma y *WHERE*:

```

mysql> SELECT t1.id,nombre,fecha,numero
-> FROM personas AS t1 JOIN telefonos AS t2
-> ON t1.id=t2.id;

```

Pero tenemos una sintaxis alternativa mucho mejor para hacer composiciones internas naturales:

```

referencia_tabla NATURAL JOIN referencia_tabla

```

Por ejemplo:

```

mysql> SELECT * FROM personas NATURAL JOIN telefonos;
+-----+-----+-----+-----+
| id | nombre | fecha | numero |
+-----+-----+-----+-----+
| 1 | Mariana | 1996-05-29 | 55253212 |
| 1 | Mariana | 1996-05-29 | 57772212 |
| 1 | Fulanito | 1996-05-29 | 57692489 |
| 2 | Claudia | 1984-08-21 | 21364749 |
| 3 | Luis | 1985-08-16 | 57153255 |
| 4 | Alfredo | 1976-08-25 | 30706805 |
+-----+-----+-----+-----+
6 rows in set (0.02 sec)
mysql>

```

5. Seguridad en Cómputo



En la actualidad se habla mucho sobre lo que es la seguridad en cómputo. Día a día se oyen reportes noticiosos hablando de un virus que atacó o atacará en una determinada fecha.,de perdidas multimillonarias que afectan a un sin número de individuos, empresas y diferentes tipos de instituciones a nivel mundial.

Pero, ¿por qué debemos de hablar de seguridad? Porque es una situación que nos incumbe a todos. A los usuarios que simplemente usan una cuenta de correo el MSN, hasta los grandes grupos empresariales.

Un virus puede comenzar, sin querer, desde una PC común y corriente. A veces existe la posibilidad de ser infectado en algún otro medio. Llegamos al hogar e infectamos nuestra computadora, una vez infectados comenzamos a enviar, archivos, mensajes, etc. Es así como podría comenzar una infección en masa, todo por un “insignificante” descuido.

Las infecciones en masa más populares son la que se llevan a cabo por medio de expertos programadores, conocidos como “hackers”. Los cuales ocupan su tiempo libre en averiguar debilidades en los diferentes sistemas a nivel mundial. Este tipo de infecciones son populares porque buscan afectar de alguna manera los diferentes procesos que se llevan a cabo bajo la plataforma de Windows.

A lo largo de este capítulo se incluye una pequeña introducción sobre este tema, para tener en cuenta los elementos y conceptos de lo que es la seguridad en cómputo. Tener conocimientos sobre los diferentes tipos de ataques y algunas maneras para poder cifrar nuestra información y así enviarla a través de la red de manera segura.

Seguridad Podría definirse como la característica de un sistema o elemento que permite garantizar:

- Que se opera como se espera que lo haga.
- Que es ajeno a todo riesgo.
- Que es ajeno a toda amenaza.
- Que no posee vulnerabilidades.
- Que es confiable
- Que funciona sin fallo.

Se orienta a ofrecer tranquilidad.

Amenaza: circunstancia o evento que puede causar daño a un sistema. El término se refiere a un evento (tornado, robo, infección por virus de cómputo, etc.) Frecuentemente aprovecha una vulnerabilidad

Vulnerabilidad: ausencia de una contramedida, o debilidad de la misma. Predisposición de un sistema a ser afectado por un agente perturbador. La debilidad puede originarse en el diseño, la implementación o en los procedimientos para operar y administrar el sistema. En seguridad informática se denomina “hoyo”.

Peligro: Es la probabilidad de que se presente una amenaza.

Grado de Exposición: Nivel de afectación (número de personas, bienes, módulos de sistema) que podrían ser afectados.

Riesgo = Peligro * Exposición * Vulnerabilidad

Riesgo Es la probabilidad de que una vulnerabilidad sea explotada, de acuerdo a su nivel de exposición y al peligro involucrado.

Definición de Sistema de Cómputo Seguro“Un sistema de cómputo es seguro si se puede confiar en que se comportará como se espera que lo haga, que la información en él se mantendrá inalterada y de manera accesible durante el tiempo que su dueño lo desee para los usuarios que el mismo determine.”

Servicios y Políticas de Seguridad

Un servicio de seguridad es una característica que debe tener un sistema para satisfacer una política de seguridad.

Una política de seguridad especifica las características de seguridad que una organización debe observar y proveer con el fin de salvaguardar su información.

Servicios de Seguridad

Estándares como el BS 7799 o el ISO 17799 consideran a la información como un activo y definen servicios para su protección.

Confidencialidad: Sólo el propietario del secreto es capaz de descifrar la información.

Autenticación: Se asegura de la identidad de la persona del otro lado de la línea. Se puede realizar mediante alguno de los siguientes mecanismos:

- “algo que se sabe”
- “algo que se tiene”
- “algo que se es”

Basados en algo que se sabe Primeros sistemas de autenticación se basan en claves de acceso: nombre usuario, nips, passwords, etc. Son fáciles de usar y no requieren de un hardware especial. Siguen siendo el sistema de autenticación más usado hoy en día.

Basados en lo que se tiene. Son todas las unidades de memoria o tarjetas de seguridad generadoras de códigos aleatorios.

Basado en lo que se “es”. Son todos aquellos que requieren un patrón para la identificación y autenticación como son: huellas digitales, lectores ópticos de retina, etc.

Servicios de Seguridad

Integridad. Se asegura que la información se mantenga inalterada desde su creación.

Autorización (Control de Acceso): se asegura que la información estará disponible sólo para determinados usuarios con la posibilidad de manejar diferentes niveles de acceso para cada uno de ellos.

No Repudio: se asegura que el emisor de la información no puede negar haberla enviado. El British standard 7799 actualmente es conocido como BS ISO/IEC,17799:2000, BS 7799-1:200.

Existen otros servicios que con frecuencia son considerados:

- **Disponibilidad.** Asegurar que el sistema este disponible cuando se le requiera.
- **Auditoria.** Asegurar que se defina un registro cronológico de los eventos exitosos y fallidos, que proporcionen evidencia de la actividad del sistema.

Niveles de Confianza I El “National Computer Security Center” publicó en 1985 un libro conocido como el “U.S. Department of Defense Trusted Computer System Evaluation Criteria” mejor conocido como “Orange Book” el cual define cuatro divisiones y siete niveles de confianza para ambientes de cómputo.

- Niveles Orange Book
- D - Mínima protección
- C1
- C2
- B1
- B2
- B3
- A - Se realizan pruebas formales de seguridad en el sistema.

Niveles de Confianza II La “Common Criteria Organization” ha proporcionado siete niveles de aseguramiento para un sistema, conocidos como “Evaluation Assurance Levels” EALs, con los cuales se clasifica a los ambientes de cómputo.

- Niveles “Common Criteria”
- EAL1 - Funcionalidad Probada
- EAL2
- EAL3
- EAL4
- EAL5
- EAL6
- EAL7 - Formalmente verificado, diseñado y probado.

La Criptología como herramienta en Seguridad Informática

Criptología La Criptografía del griego “Kryptós” oculto y “gráphein” escritura, “escritura oculta” se encarga de convertir un texto normal y comprensible en un formato incomprensible a menos que se posea un conocimiento secreto.

En épocas recientes la criptología se define como la ciencia de usar las matemáticas para cifrar y descifrar información.

El criptoanálisis es la ciencia de analizar y romper la comunicación segura mediante el análisis del algoritmo empleado.

La Criptología involucra tanto criptoanálisis como la criptografía.

Algoritmos Criptográficos Los algoritmos criptográficos pueden estar basados en:

- **Criptografía Simétrica.**- Requiere que el emisor y el receptor compartan una clave secreta “Llave”, la cual es utilizada para cifrar el mensaje cuando se envía y descifrar el mensaje al recibirlo. El gran inconveniente se presenta en el proceso de intercambiar de forma segura la Llave. En este esquema de seguridad es importante que el emisor disponga de un canal seguro para realizar la entrega de la llave al inicio de la transmisión.

Ejemplos: DES, 3DES, AES, Blowfish, IDEA

- **Criptografía Asimétrica (también conocida como de llave pública).**- Resuelve el problema de intercambiar el secreto utilizando para ello un algoritmo basado en dos llaves, tanto el emisor como el receptor utilizan un par de llaves una “pública” y otra “privada”. La privada es mantenida en secreto por un individuo, la pública esta disponible para que otro individuo sea capaz de cifrar la información sensible con ella.

Cuando el emisor desea enviar un mensaje lo cifra con la llave pública del receptor, siendo éste el único capaz de descifrarlo mediante el uso de su llave privada. En este caso conseguimos asegurar la confidencialidad del mensaje.

Rivest Shamir Adleman (RSA) y Diffie-Hellman (D-H) son dos sistemas de llaves públicas utilizados actualmente.

Algoritmos de Clave Pública

- **Diffie-Hellman.** El algoritmo de clave pública Diffie-Hellman fue desarrollado en 1976, por Whitfield Diffie y Martín Hellman. Esta basado en un algoritmo llamado de exponencial de claves.

- **RSA** El algoritmo de clave pública RSA, fue desarrollado por Rivest Shami Adelman en 1977, es el criptosistema de clave pública de uso más común en la actualidad, se utiliza para cifrado y autorización, tiene extensiones de 768, 1024 y mayores. La solidez de RSA proviene de la dificultad para factorizar números primos grandes, RSA se utiliza en aplicaciones tan populares como: PGP y S/MIME.

Message Digest. Conocidos también como Hashes, checksums son el resultado de algoritmos unidireccionales que permiten generar una “huella digital” de un mensaje. Permiten obtener una cadena de longitud fija que es una representación condensada de un mensaje. Permite garantizar la integridad de un mensaje, si un solo carácter del mensaje original cambia el Hash será totalmente distinto.

Ejemplos: SHA1, SHA2, MD5

Si el emisor cifra con su llave privada, todo aquel que disponga de la llave pública puede conocer el mensaje, por lo que el objetivo en este caso no es asegurar confidencialidad, sino que obtenemos lo que se conoce como: **Firma Digital**

En este escenario se obtiene:

- Autenticación dado que sólo el propietario de la llave privada pudo haber generado el mensaje.
- Integridad considerando que el mensaje requiere llegar sin cambios para que el algoritmo de descifrado pueda operar.

- No repudio considerando que la llave privada esta sólo en posesión del emisor, por lo que no puede negar su autoría.

Ejemplos: RSA, DSA

- Los algoritmos de llave publica (ejemplo: RSA, DSA) son mucho más lentos que los de llave privada (ejemplos: DES, AES, 3DES)
- Por lo que con frecuencia la firma digital lo que realmente firma es únicamente el “Digest” del mensaje.

GNU PG GNU Privacy Guard (GnuPG) es un sistema de codificación de código libre y desarrollo abierto. Es un reemplazo libre a PGP Pretty Good Privacy originalmente creado por Phil R Zimmermann.

Instalando GPG

- Download, Configurar, Compilar e Instalar gnupg...
[http://www.gnupg.org/\(en\)/download/index.html](http://www.gnupg.org/(en)/download/index.html)
./configure --prefix=/usr/remote/gpg
make
make install

Usando GPG

Se genera un nuevo par de llaves (privada y pública) para el usuario.

- gpg --gen-key

Muestra la llaves con las que cuenta el “llavero”.

- gpg --list-keys

Se exporta la llave pública del usuario a un archivo.

- gpg --export --armor > MiLlave.key

Se importa la llave pública de otro usuario

- gpg --import < LlaveDeUnAmigo.key

Se firma la llave pública de otro usuario y se considera confiable.

- gpg --sign-key <id>

Se cifra un Mensaje.txt con la llave de un usuario.

- gpg -e Mensaje.txt

Se Descifra el Mensaje.txt con la llave privada del usuario.

- gpg -d Mensaje.txt.gpg

Se firma un Mensaje.txt con la llave de un usuario.

- gpg -s Mensaje.txt
- gpg --sign Mensaje.txt

Se firma el Mensaje.txt en texto claro con la llave de un usuario.

- `gpg --clearsign Mensaje.txt.gpg`

Se verifica que la firma del Mensaje.txt.

- `gpg -v Mensaje.txt.gpg`
- `gpg --verify Mensaje.txt.gpg`

Open Secure Socket Layer

El proyecto OpenSSL es el desarrollo de un código abierto con el propósito de alcanzar un Toolkit de SSL y TLS con el nivel de una herramienta profesional.

El proyecto está basado en el código fuente de SSLeay desarrollado por Eric. A Young y Tim J.Hudson. Para mayor información <http://www.openssl.org/>

Instalando Open Secure Socket Layer Download, Configurar, Compilar e Instalar openssl...

```
http://www.openssl.org/source/openssl-0.9.7e.tar.gz
./config --prefix=/usr/remote/openssl
make
make install
```

- Cifrando mensajes con openssl
`openssl enc -e -des3 -in Ejemplo.txt -out Ejemplo.des3`
- Descifrando mensajes con openssl
`openssl enc -d -des3 -in Ejemplo.des3 -out Ejemplo.txt`
- Message Digest
`openssl dgst -md5 Ejemplo.txt`
MD5(Ejemplo.txt)= 08a4ebfb7bf8f68a4e169a9806cb3a3b
`md5sum Ejemplo.txt`
08a4ebfb7bf8f68a4e169a9806cb3a3b Ejemplo.txt
`openssl dgst -sha Ejemplo.txt`
SHA(Ejemplo.txt)= 37ad34ca6b2d8447a7504b3c76aa7d8bb319644f
- Generando llaves "keys"... Generando una llave RSA de 128 bits
`openssl genrsa 128`
- Generando una llave RSA de 128 bits cifrada con 3DES
`openssl genrsa -des3 128`
- Generando una llave DSA de 1024 bits cifrada con 3DES

Primero generamos los parametros que DSA requiere:

```
/usr/remote/bin/openssl dsaparam -out parametros.pem 128
```

Con los parametros generados se realiza la llave:
`/usr/remote/bin/openssl gendsa -out privkey.pem parametros.pem`

5.1 Seguridad en el Servidor de Web “Apache”

Seguridad en Apache Existen múltiples recomendaciones que se pueden realizar para mejorar la seguridad del servidor de Web, entre las más importantes tenemos:

- Mantener en producción una versión actualizada.
- Evitar el uso de permisos no necesarios en los directorios de apache.
- Restringir el uso de SSI y de CGIs.
- Revisar con frecuencia las bitácoras del web server.
- Seleccionar passwords fuertes para los accesos restringidos.
- Implementar el soporte para SSL (mod_ssl)
- Agregar módulos orientados a incrementar el nivel de seguridad (mod_access, mod_auth, mod_security)

Control de Acceso Apache proporciona la capacidad de negar o permitir el acceso al servidor de Web. Las directivas involucradas son :

Order. Define el orden en que las directivas allow o deny serán implementadas

- **Allow.** Define los hosts que tendrán acceso al recurso.
- **Deny.** Define los hosts que no tendrán acceso al recurso.

Allow / Deny Sintaxis:

Allow from all|host|env=env-variable [host|env=env-variable]

Deny from all|host|env=env-variable [host|env=env-variable]

Estás directivas permiten definir un conjunto de clientes a los cuales se les pueden permitir o negar el acceso al servicio de Web.

Directivas Allow/Deny La definición de los nombres se puede realizar mediante alguna de las siguientes formas:

- El nombre parcial de un dominio parcial: atacantes.com.ar
- Una dirección IP. 200.38.166.1
- La pareja Red y mascara 10.2.0.0/255.255.255.0
- Un dirección de red definida por Classless Inter-Domain Routing (CIDR)
10.2.2.110/24

“ Todo lo que no está explícitamente permitido está prohibido ”.

<Directory /usr/local/apache/htdocs/intranet>

```
Order allow, deny allow from 192.168.1.10
</Directory>
```

Permite explícitamente a los clientes definidos en allow, niega a todos los demás.

“ Todo lo que no está explícitamente prohibido está permitido ”.

```
<Directory /usr/local/apache/htdocs/intranet>
Order deny, allow deny from 192.168.1.10
</Directory>
```

Niega explícitamente a los clientes definidos en deny, permite a todos los demás.

¿HTTP Confiable?

Considerando que los medios de transmisión no son confiables, es necesario establecer mecanismos para evitar que la información sea capturada durante su transmisión.

- Para lo cual se utiliza: SSH, POPS, IMAPS, SFTP, etc.
- En el caso de HTTP implementaremos el uso de HTTPs que utiliza SSL para cifrar su transmisión.

Secure Socket Layer Diseñado por Netscape en 1993, SSL es una propuesta de estándar para cifrado y autenticación en el Web.

Es un esquema de cifrado de bajo nivel usado para el cifrado de transacciones en protocolos de nivel aplicación como HTTP, FTP, etc.

Con SSL puede autenticarse un servidor con respecto a su cliente y viceversa.

Objetivos de SSL

- Seguridad criptográfica.

Se sugiere el uso de SSL para establecer conexiones seguras entre dos partes.

- Interoperabilidad

Programadores deben poder desarrollar aplicaciones basadas en SSL, que intercambien parámetros criptográficos sin tener conocimiento de los códigos de los programas de cada uno.

- Extensibilidad.

SSL provee un marco donde pueden incorporarse métodos criptográficos según se necesite.

Certificados Digitales

- La norma X.509 es el estándar para formatos de certificados con llave pública.
- Un certificado X.509 consiste de la llave pública de un usuario y la firma de un tercero para la identificación en el bloque de identificación de ese usuario.

Autoridades Certificadoras

- El esquema de firmas digitales requiere de alguien que autentique que un individuo es quien dice ser.
- Instituciones gubernamentales, comerciales o financieras, se encargan de emitir certificados digitales.
- En los certificados se integra la llave pública del individuo, piezas de información sobre el individuo e identificadores de la autoridad certificadora quien finalmente se encarga de firmar digitalmente el certificado con su llave privada.

• Si deseamos elevar el nivel de confianza sobre nuestro certificado podemos:

1. Recurrir a una Certify Authority y solicitar que nos firme nuestro certificado.

www.verisign.com
www.geotrust.com

2. Convertirnos en una CA y firmar nosotros mismos nuestros certificados, sin embargo en este caso:

- Se debe instalar en cada navegador el ca.crt
- Y sólo es factible para el interior de las organizaciones.
- Al inicio de la navegación en el sitio solicitar al visitante su confianza y solicitarle instale nuestro certificado

Apache con soporte para SSL El modulo mod_ssl proporciona el soporte necesario para implementar SSL sobre HTTP.

```
Para Apache 2.0.54:  
gzip -dc httpd-2.0.54.tar.gz | tar xf -  
cd httpd-2.0.54  
./configure --prefix=/usr/remote/apache --enable-so --enable-ssl --with-  
ssl=/usr/remote/openssl  
make  
make install
```

```
apachectl start
apachectl stop
/usr/remote/apache/bin/httpd -l | grep mod_ssl
```

Generando el Certificado Para generar el certificado necesario para realizar la comunicación vía HTTPS se debe realizar:

- Se genera la llave privada
openssl genrsa -out server.pem 2048
- Se genera un “Certificate signing request” el cual se enviará a la CA.
openssl req -new -key server.key -out cert.csr
- Si se utiliza un Certificado auto firmado, se realiza:
openssl req -new -x509 -key server.key -out cacert.pem -days 30

¿ Cómo funciona HTTPS ?

1. El cliente establece una conexión a un servidor que soporta HTTPS
2. Se negocian los parámetros de comunicación
El cliente notifica al servidor cuales son los parámetros que soporta y el servidor selecciona cuales de ellos utilizarán. Entre los parámetros tenemos: que algoritmos de cifrado se utilizarán, la versión del protocolo, etc.
3. El servidor envía al cliente su certificado digital (x509).
4. El cliente utiliza su copia de la llave pública de la CA, la fecha y el nombre del servidor para validar el certificado.
5. Se realiza un acuerdo de llave, El cliente genera un valor aleatorio “Clave premaestra”, lo cifra con la llave pública del servidor y se lo envía.
6. El servidor debe descifrar con su llave privada la “clave premaestra” (Autenticación)
7. Tanto el cliente como el servidor deben generar la “clave maestra” a partir de la “premaestra”.
8. El cliente y el servidor intercambian cifrado el Message Digest de la “clave maestra”, para asegurar que obtuvieron lo mismo.(Integridad y Confidencialidad)
9. A partir de la “clave maestra” ambos generan la clave de sesión con la cual se cifrará la comunicación.

Ataques: Acción o acciones que tienen por objetivo el que cualquier parte de un sistema de información automatizado, deje de funcionar de acuerdo con su propósito definido. Esto incluye cualquier acción que causa la destrucción, modificación o retraso del servicio no autorizado.

En sistemas de información no se refiere a un ataque físico (aunque puede ser). Un ataque no se realiza en un solo paso, depende de los objetivos del atacante y puede consistir de varios pasos antes de llegar a su objetivo.

Tipos de Ataques

Ataques Pasivos: recopilar información en Bases de Datos.

- Whois
 - ARIN
 - LACNIC
- DNS - NIC
 - Listados de Ips y Nombres de Dominio
 - Trasferencias de Zonas
- Trazado de Ruta
 - Traceroute
 - www.visualroute.com
- Sniffers
 - snort
 - dsniff
- Analizadores de Tráfico
 - Ethereal

Sniffers Un sniffer es un proceso que "olfatea" el tráfico que se genera en la red. Puede leer toda la información que circule sólo por el tramo de red en el que se ubique. Se pueden capturar claves de acceso, datos que se transmiten, números de secuencia, etc.

Un analizador de protocolos es un sniffer al que se le ha añadido funcionalidad suficiente como para entender y traducir los protocolos que se están hablando en la red. Debe tener suficiente funcionalidad como para entender las tramas de nivel de enlace, y los paquetes que transporten.

Normalmente la diferencia entre un sniffer y un analizador de protocolos, es que el segundo está más orientado al análisis del tráfico y no a la escucha de los datos transmitidos.

- Ejemplos de Sniffers abundan:
 - Dsniff
 - Sniffit
 - snifsol
- Ejemplos de Analizadores de protocolos Ethereal
 - Tcpdump

- Algunas herramientas más elaboradas pueden utilizarse como sniffers.
snort

Lectura a nivel enlace El sniffer se dedica a leer TRAMAS de red. Los datos que obtendremos de él serán tramas que transportarán paquetes (IP, IPX, etc). En estos paquetes se incluyen los datos de capas superiores, entre ellos los de la capa de aplicación (posiblemente claves de acceso).

SNORT - Download, compilar e instalar.

```

http://www.pcre.org
http://www.snort.org
tar zvxf pcre-5.0.tar.gz
tar zvxf snort-2.3.0RC2.tar.gz
cd pcre-5.0
./configure --prefix=/usr/remote/pcre
make
su -
make install
./configure --prefix=/usr/remote/snort --with-pcre=/usr/remote/pcre
make
su -
make install

```

Tipos de Sniffers

- **Pasivos** Sniffers no realizan actividad alguna. Solo capturan paquetes
- **Activos** Sniffers intentan apoderarse de las sesiones. Uso de técnicas para lograr lo anterior
 - Spoofing
 - Envenenamiento de la tabla de arp

Detección de sniffers: los Sniffers son difíciles de detectar y combatir ya que son programas pasivos. No generan bitácoras, cuando se usan propiamente, no usan mucho disco ni memoria. Es posible localizarlos a nivel local o a nivel red, algunos pueden localizarse mediante herramientas como: antisnif

Ataques Activos

- | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • Virus • Caballo de Troya • Gusanos (Worms) • Explotación de: <ul style="list-style-type: none"> • Backdoors • Stack overflow • Formatos de cadena • Bombas lógicas | <ul style="list-style-type: none"> • Falsificación • Usurpación • Sniffers • Spoofing • Spam • Graffiti (Web Defacement) • Ingeniería Social • Negación de servicio |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Escaneo de Puertos El escaneo de puertos es un método que consiste en conectarse a los puertos UDP y TCP del sistema objetivo con el propósito de determinar cuales se encuentran escuchando.

Un ejemplo de software que escanea puertos es el Nmap

Nmap Download, compilar e instalar.

http://www.insecure.org/nmap/nmap_download.html

```
tar zxvf nmap-3.75.tgz
```

```
cd nmap-3.75
```

```
./configure --prefix=/usr/remote/nmap
```

```
make
```

```
su -
```

```
make install
```

Nmap realiza un escaneo para verificar que puertos se encuentran disponibles en el servidor objetivo.

```
nmap 132.248.75.135
```

Si deseo que nmap sólo realice un barrido sobre puertos TCP

```
nmap -sT 132.248.75.135
```

Si deseo que nmap sólo realice un barrido sobre puertos UDP

```
nmap -sU 132.248.75.135
```

Si deseo que nmap haga un barrido de Ips

```
nmap -sP 132.248.75.*
```

Si deseo que nmap sólo realice un barrido con conexiones semiabiertas.

```
nmap -sS 132.249.75.135
```

Estados TCP

- CLOSED: no-estado, antes de que la conexión empiece
- LISTEN: host espera petición conexión
- SYN-SENT: host envió paquete syn y espera syn_ack
- SYN-RCVD: host recibió paquete syn y respondió con syn_ack
- ESTABLISHED: host iniciador recibió un syn-ack y el host receptor recibió paquete ack

Medio no confiable Dado que los medios de transmisión no son confiables es necesario establecer mecanismos para evitar que la información sea capturada durante su transmisión.

Para lo cual se utiliza:

- SSH
- HTTPS
- POPS

- IMAPS

Secuestro de sesiones Término en inglés: hijacking. Tipo de ataque en el que el atacante toma control de una comunicación tal y como un secuestrador de aviones toma control del avión. Entre dos entidades y haciendo pasar por una de ellas

En un tipo de ataque (man in the middle), el atacante toma control de la conexión mientras ésta se produce. El objetivo es robar una conexión generada por una aplicación de red iniciada por un cliente (p.e. telnet)

DOS Denial of Service. Todo ataque que busca aparentar una alta demanda sobre un servicio válido con el objeto de saturar la capacidad de respuesta y generar una caída del sistema víctima.

DDOS Distributed Denial of Service. En febrero/marzo del 2000, varias empresas que apoyaron su estrategia en Internet fueron atacadas.

- Yahoo! estima pérdidas por US\$500,000 dls por dejar de dar servicio durante 3 horas

Entre ellas destacan:

- CNN (Agencia Noticiosa)
- Amazon (Venta de libros, discos, etc.)
- e-Bay (Venta de artículos en remate)
- e-Trade (compra y venta de acciones)
- Yahoo (Correo gratuito)

Ejemplos de ataques DoS

- Ping de la muerte
- Inundación Syn
- Spoofing
- Smurf
- Fraggle
- DoS Distribuido

Contramedidas inundación SYN Aumentar el tamaño de la cola de conexión. Disminuir el período de tiempo de establecimiento de conexión. Filtrado de paquetes (direcciones IP no “alcanzables”). Emplear parches software de los fabricantes de los SO

Smurfing Smurf es uno de los ataques de DoS más temidos. Requiere 3 actores: La víctima, el atacante y la red amplificadora. El atacante originará un paquete ICMP hacia la dirección de broadcast de la red amplificadora, haciendo aparecer que su origen es una interfaz de la red de la víctima, cada interfaz de la red amplificadora enviará respuestas a la supuesta interfaz de origen

Previendo smurf Para no permitir ser utilizado como red amplificadora debe deshabilitar el paso de mensajes destinados a broadcast a través de los

routers de frontera (Cisco: no ip direct-broadcast) Para limitar el daño ocurrido por un ataque de este tipo sobre su red, limite el tráfico ICMP a un valor razonable de acuerdo a su disponibilidad de ancho de banda. Verifique si realmente necesita permitir tráfico de entrada ICMP a toda su red.

Spoofing Spoofing es la creación de paquetes de comunicación TCP/IP usando una dirección IP de alguien más. Lo anterior permite entrar en un sistema haciéndose pasar por un usuario autorizado. Una vez dentro del sistema, el atacante puede servirse de éste como plataforma para introducirse en otro y así sucesivamente.

Ingeniería Social Es una de las formas más comunes para penetrar sistemas de "alta seguridad". Uso de trucos psicológicos, por parte de un atacante externo, sobre usuarios legítimos de un sistema para obtener información (usernames y passwords) necesaria para acceder a un sistema. Se basa en ataques como: usurpación de identidad, hurgar en la basura, inocencia de la gente, relaciones humanas, etc.

Virus Un virus se define como una porción de código de programación cuyo objetivo es implementarse asimismo en un archivo ejecutable y multiplicarse sistemáticamente de un archivo a otro. Además de esta función primaria de "invasión" o "reproducción", los virus están diseñados para realizar una acción concreta en los sistemas informáticos sin la autorización del usuario.

Variantes relacionadas con virus En ocasiones se habla de estas variantes como si de virus se tratara, cuando en realidad son conceptualmente diferentes. Algunos antivirus pueden detectarlos.

Estas variantes son:

- Troyanos
- Gusanos
- Bomba lógica
- Spyware
- Adware

Los gusanos. Es un programa que produce copias de sí mismo de un sistema a otro a través de la red. En las máquinas que se instala, produce enormes sobre-cargas de procesamiento que reduce la disponibilidad de los sistemas afectados.

Virus vs backdoors Diferencias que hay que tomar en cuenta para protegernos mejor

Virus. El programa por sí sólo se ejecuta, vive dentro de otro programa. Escala en memoria

Backdoor Cuando el equipo ha sufrido un incidente, alguien "entró" al sistema, es común que se instale para seguir con el control sobre el sistema.

Caballo de Troya. Se le envía al usuario para que lo ejecute aparentando que se trata de un programa con una funcionalidad diferente

Bombas lógicas Una bomba lógica es una modificación en un programa que lo obliga a ejecutarse de manera diferente bajo ciertas circunstancias. Bajo condiciones normales, el programa se comporta como previsto y, la bomba no puede ser detectada.

Ejemplo de pseudocódigo.

```
If eth0=down THEN bomba= "rm -rf /"
ELSE bomba=./running
```

Exploit Se refiere a la forma de explotar una vulnerabilidad. Término muy enfocado a herramientas de ataque, sobre equipos de cómputo. Aprovechamiento automático de una vulnerabilidad

Generalmente en forma de un programa/software que realiza de forma automática un ataque aprovechándose de una vulnerabilidad

Los protagonistas

White Hat Investigador encargado en buscar nuevas vulnerabilidades en los sistemas, desarrolla el PoC y avisa al fabricante del producto sobre la vulnerabilidad descubierta y su explotación antes de publicarlo en la red.

Gray Hat. Investigador que encuentra nuevas vulnerabilidades en los sistemas, desarrolla el PoC y lo publica en la red sin avisarle al fabricante del producto.

Black Hat Algunos dicen que son un mito, investigadores que explotan nuevas vulnerabilidades en los sistemas pero que jamás dan aviso ni al fabricante ni son publicadas en la red, sólo la usan para beneficio personal.

Los que más abundan

Script Kiddies Gente con la capacidad de buscar un programa en la red y ejecutarlo. Necesidad de pertenencia al llamado "underground", aunque sean de grupos "lammers" No hay preocupación por las consecuencias reales de sus actos. Se sienten muy "cool", grafiteros por excelencia. Utilizan tipografía "leet" o "7337"

```
3ST0 S3RI4 UN 3J3MPLO D3 DICH0 L3NGU4J3
```


6 Proyecto Pruebas Psicológicas a Distancia con Software Libre.



Este proyecto es orientado a los psicólogos y resultaría una herramienta útil para poder elaborar todo el análisis de una terapia pero con la ventaja de poder realizada a cierta distancia. En general es un compendio de pruebas psicológicas con fines bien definidos, desde una prueba de inteligencia hasta una capaz de percibir ciertos trastornos de la salud mental.

El objetivo general de este proyecto es el que los psicólogos dejen a un lado la terapia tradicional para poder realizarla apoyados en las nuevas tecnologías de información a nivel mundial. Usando Internet, con ayuda de un pequeño sistema basado en Software Libre

La interacción entre paciente y psicólogo es indispensable para la vieja escuela. Por ello, muchos de los doctores especializados en el tema verán esta opción como poco práctica para su área laboral. Sin embargo nos encontramos en una época en la que la tecnología nos permite llevar a cabo mejoras con fines prácticos tanto para el paciente como para el psicólogo.

Los elementos que se pretenden alcanzar en la elaboración de este proyecto es que se deje a un lado una entrevista en tiempo real en cada una de las terapias, para dar cabida a los nuevos métodos de interacción a distancia. Facilitando así un sin número de actividades extras para cada una de las partes inmiscuidas en este nuevo tipo de terapias.

En el aspecto psicológico-emocional podrían existir algunos inconvenientes que deberán de ser tomados en cuenta por cada uno de los psicólogos participantes. Aquí únicamente se pretende dar una opción para poder llevar a cabo una nueva manera de diagnosticar a los pacientes.

El ambiente gráfico es de vital importancia en el desarrollo de este tipo de pruebas, ya que en algunos de los casos los individuos de estudio suelen ser niños pequeños, mismos que se distraen fácilmente, para ello se requiere mantener la atención del sujeto sobre la computadora mediante una página llamativa, pero sólo en casos de que el paciente sea menor de edad.

En el caso de los adultos se podrá mantener un ambiente gráfico normal, donde sólo aparecerán los reactivos respectivos, según la prueba que se desee llevar a cabo.

6.1 REQUERIMIENTOS

1. Sistema Operativo Linux.
2. Servidor WWW Apache
 - Con soporte para CGI's para poder realizar las verificaciones pertinentes
 - Soporte para PHP para poder manejar variables y realizar la interacción con la Base de Datos.
 - Soporte de MySQL para poder llevar el control adecuado de la Base de Datos.
3. PHP 5 para el desarrollo de formularios y el manejo de las variables .
4. MySQL que será nuestro gestor de la base de datos.

Debemos recordar que el Software Libre es una muy buena opción para el desarrollo de este proyecto en particular. Primero porque nos brinda una estabilidad que otros sistemas no otorgan. En segundo lugar porque no es necesario hacer los pagos de las diferentes licencias, por lo que los precios disminuyen considerablemente ya que tan sólo se cobraría el desarrollo del sistema y su mantenimiento.

El servidor deberá montarse en un equipo específico para este uso. Será el enlace entre los usuarios y los psicólogos. Deberá contar con el sistema operativo Linux, para evitar problemas de programa, que en otros sistemas suelen ser muy cotidianos por el manejo equivocado de la memoria, lo que podría ocasionar una "caída" del sistema en más de una ocasión. Esto se vería reflejado inmediatamente en el costo de mantenimiento del sistema. En pocas palabras, hacerlo menos susceptible a errores de sistema, y tan solo preocuparnos por los posibles ataques externos como: "hackers" y alguna falla fuera de lo común.

Apache puede ser configurado de varias maneras, desde el puerto donde se enviará la información, hasta la confirmación de passwords. En nuestro caso será recomendado el uso de un puerto diferente para que no cualquier usuario pueda conocer la existencia de este sitio. Sin embargo, debemos recordar la existencia de los buscadores, que a pesar de trabajar bajo un puerto poco común son infalibles al momento de realizar la búsqueda. Para contar con un poco más de seguridad se evitará que los usuarios den de alta passwords débiles. La manera más fácil es por medio de un método de análisis que lleva a cabo el sistema operativo Linux. Dicho método resulta algo tardado, pero en términos de seguridad resultaría indispensable.

PHP es un lenguaje de programación sencillo que nos permite realizar muchos tipos de operaciones, comparaciones y validaciones, además de resultar sencilla la interacción con las bases de datos. PHP junto con HTML serán los encargados de brindar la presentación de nuestra página en general. En el caso de las validaciones que se efectuarán, necesitaremos de la programación en Java

Script, que permite “separar” esta parte de programación del resto del código de la página.

MySQL será el controlador de la base de datos por ser sencillo al momento de manejar los datos y resultar seguro su uso.. Este programa no es inmune de errores pero garantiza facilidad de uso y cierta rapidez.

6.2 Esquema de la Base de Datos

A continuación se presenta el diagrama de entidad-relación que tendrá nuestra base de datos.

Como puede apreciarse en nuestro modelo la base no tendría muchas tablas lo que facilitaría el manejo de los datos. Por el contrario, nuestro contenido sería de considerable tamaño ya que se incluirían los diferentes reactivos de cada una de las diferentes pruebas, que pueden llegar a ser desde 10 hasta 180 reactivos. Algunas pruebas, contadas, podrían tener más reactivos.

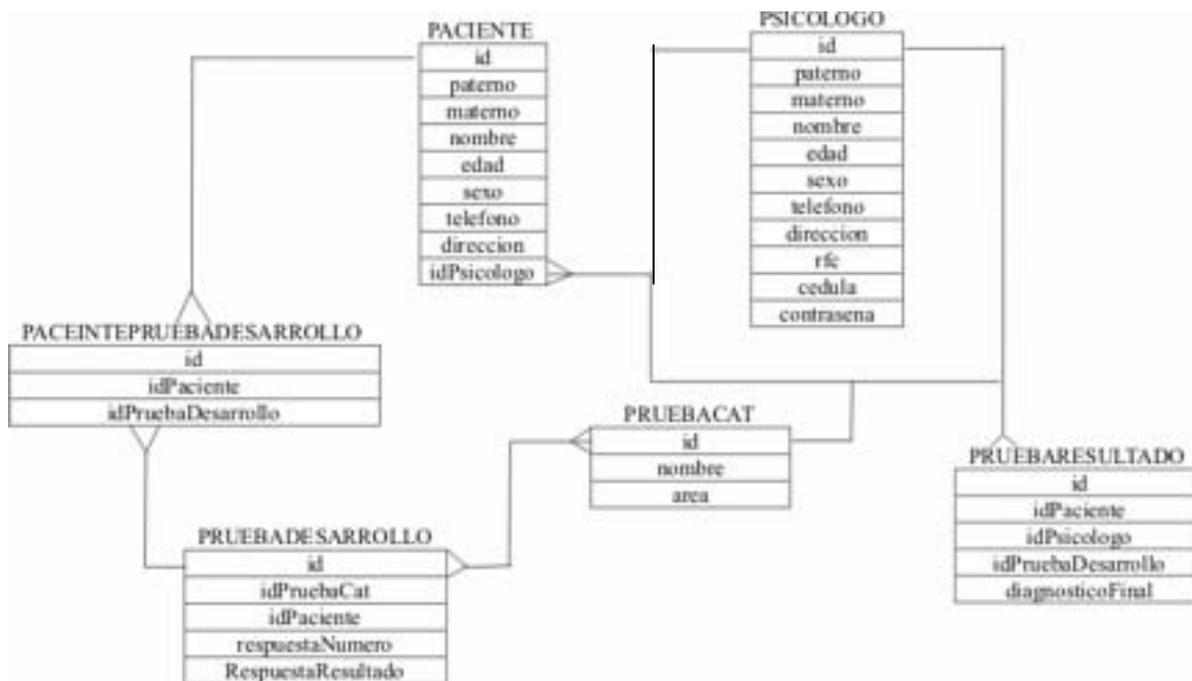


Figura 6.1 Diagrama E-R de nuestra Base de Datos

6.3 Desarrollo.

A continuación se presenta una visión general de cómo llegaría a lucir el sistema, orientado hacia la presentación visual que este presentaría.

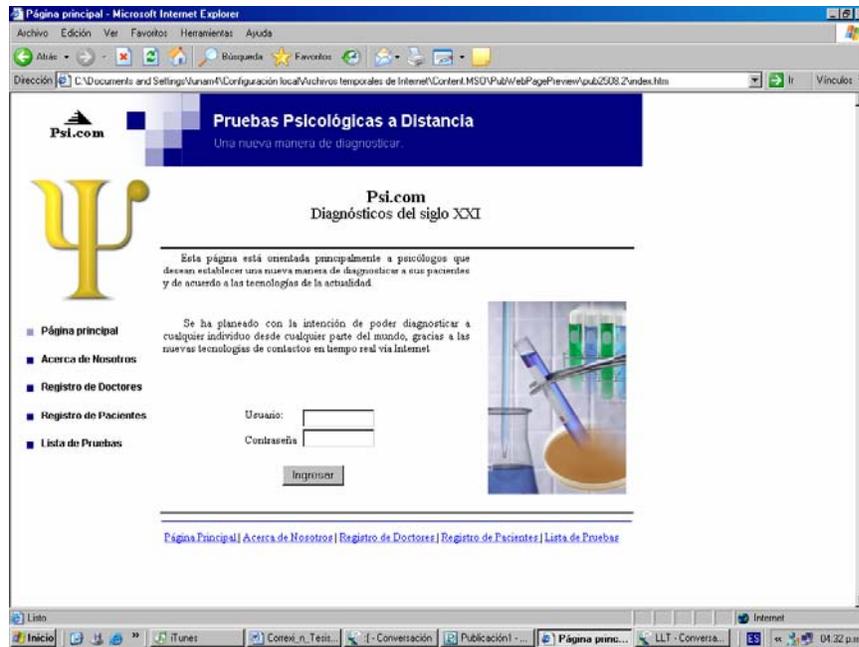


Figura 6.2 Página Principal

Página principal donde se dará una breve explicación de los servicios que se brindarán en el sitio principal. Aparecen unos cuadros de texto para los usuarios previamente registrados, tanto pacientes como psicólogos. En este punto se llevará a cabo una validación por medio de PHP que comparará los usuarios con nuestra base de datos existente, específicamente en la tabla Paciente.

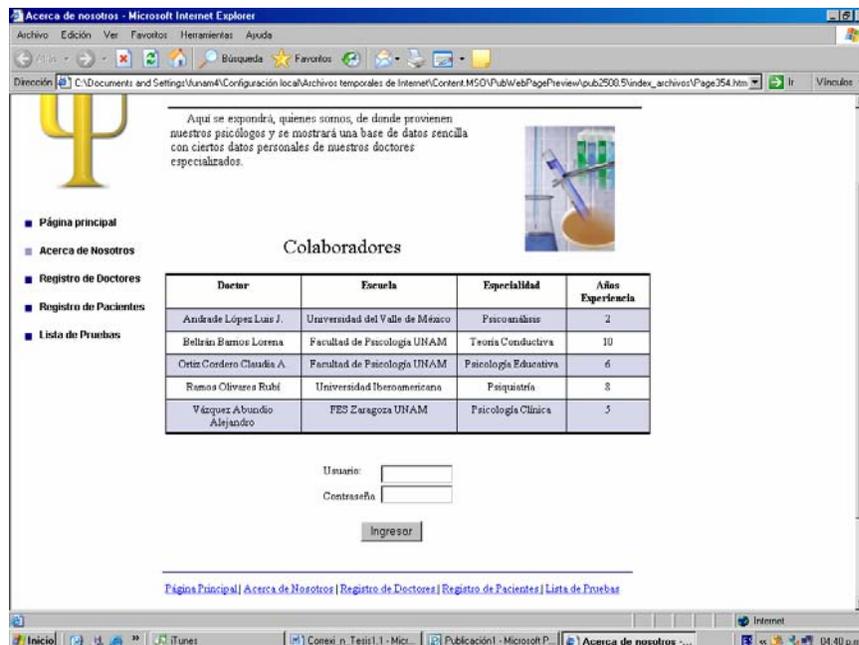
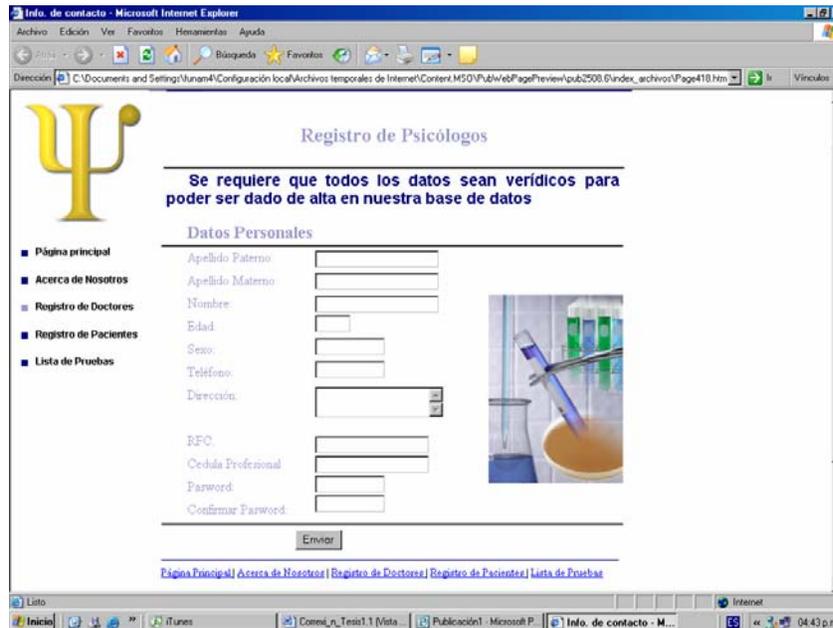


Figura 6.3 Página Introdutoria

La anterior página forma parte de una introducción sobre los servicios que se brindarán. En general se muestran unas tablas representativas de los colaboradores con los que contaremos para el desarrollo de este sistema, resolución de las pruebas y asesoría en el desarrollo de este sitio.

La tabla que se muestra es de la tabla de Psicólogos registrados o previamente cargados



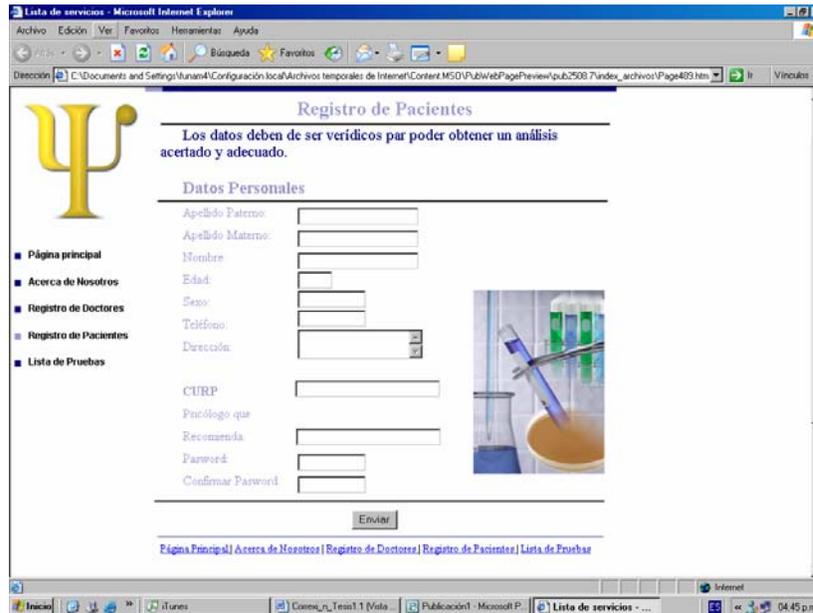
The image shows a screenshot of a web browser window titled "Info. de contacto - Microsoft Internet Explorer". The browser's address bar shows a local file path. The main content area displays a registration form titled "Registro de Psicólogos". At the top left is a large yellow Psi symbol. To the right of the symbol is the title "Registro de Psicólogos". Below the title is a warning message: "Se requiere que todos los datos sean verídicos para poder ser dado de alta en nuestra base de datos". The form is titled "Datos Personales" and contains the following fields: Apellido Paterno, Apellido Materno, Nombre, Edad, Sexo, Teléfono, Dirección, RFC, Cédula Profesional, Password, and Confirmar Password. There is an "Enviar" button at the bottom of the form. On the right side of the form, there is a small image of a laboratory flask with orange liquid. A navigation menu on the left side of the page includes: "Página principal", "Acerca de Nosotros", "Registro de Doctores", "Registro de Pacientes", and "Lista de Pruebas". At the bottom of the browser window, the taskbar shows several open applications, including "iTunes", "Comed_n_Tesis1.1 (Vista)", "Publicación1 - Microsoft P...", and "Info. de contacto - M...". The system clock shows "04:43 p.m.".

Figura 6.4 Formulario para registro.

Formulario que se utilizará para agregar nuevos colaboradores, así como nuevos usuarios que deseen realizar sus pruebas a distancia, con la finalidad de apoyarse en las nuevas tecnologías.

La verificación que se llevará a cabo en este paso será vinculada a la identidad del psicólogo, comparando los datos con los almacenados previamente en nuestra base de datos para evitar redundancia en los mismos. En este paso también se lleva a cabo la verificación del Password, evitando que se trate de un Password débil.

6. Proyecto de Pruebas Psicológicas a Distancia con Software Libre



The screenshot shows a web browser window with the title 'Lista de servicios - Microsoft Internet Explorer'. The address bar shows the URL: 'C:\Documents and Settings\Manant4\Configuración local\Archivos temporales de Internet\Content.MSO\PubWebPageReview\pub25087\index_archivo1\Page403.htm'. The main content area is titled 'Registro de Pacientes' and features a large yellow Psi symbol on the left. Below the symbol is a navigation menu with the following items: 'Página principal', 'Acerca de Nosotros', 'Registro de Doctores', 'Registro de Pacientes', and 'Lista de Pruebas'. The main form area is titled 'Datos Personales' and contains the following fields: 'Apellido Paterno', 'Apellido Materno', 'Nombre', 'Edad', 'Sexo', 'Teléfono', 'Dirección', 'CURP', 'Psicólogo que recomienda', 'Password', and 'Confirmar Password'. There is an 'Enviar' button at the bottom of the form. To the right of the form is a small image of laboratory glassware. At the bottom of the browser window, the taskbar shows the system tray with the time '04:45 p.m.' and the date '11/11/2007'.

Figura 6.5 Formulario de Usuarios

Página para el registro de los usuarios. A diferencia a la de los psicólogos la verificación iniciará tomando el campo de recomendación, ya que sin éste no podrán ser agregados a la base de datos, ya que resulta indispensable que hayan sido recomendados por que las pruebas que contendrá el sitio son muy costosas y difíciles de conseguir por lo que no cualquier usuario que se encuentre navegando en Internet pueda tener el acceso a las mismas.

Para validar la identidad verdadera del paciente se utilizará el campo del CURP, que una vez más será validado con una base de datos externa, proveniente de la Secretaría de Salud.

La validación también se llevará a cabo en la comparación de las edades para una vez llenado el campo de edad pueda hacerse una relación entre la edad, sexo y las posibles pruebas que pueda llevar a cabo, ya que no es lo mismo realizar una prueba de inteligencia a diferentes edades y cierto contexto social, para ello se recomienda un conocimiento previo entre el paciente y el psicólogo.

6. Proyecto de Pruebas Psicológicas a Distancia con Software Libre

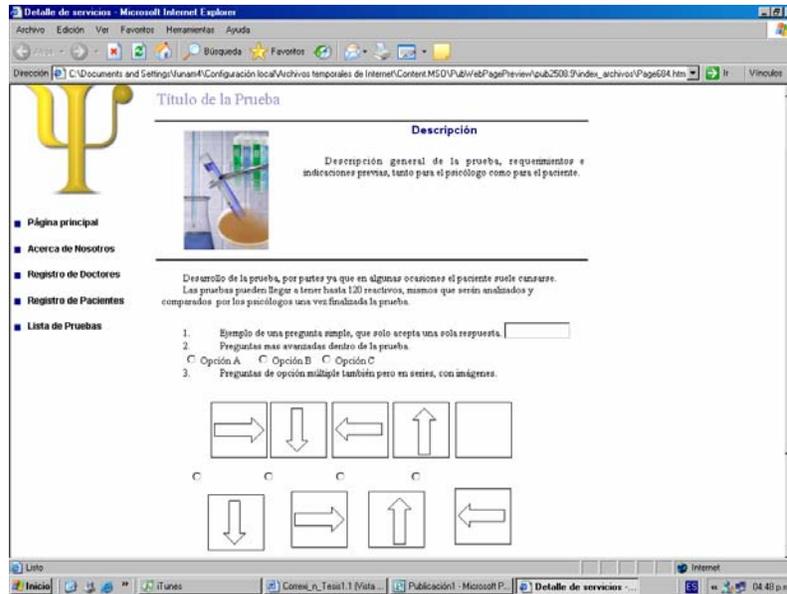


Figura 6.6 Ejemplo de una prueba en Línea

La anterior página pretende ser un ejemplo de cómo lucirá la página donde puedan llevarse a cabo las diferentes pruebas. Se visualiza un ejemplo de las preguntas que pueden tener como respuesta texto y aquellas en donde se requiere completar una serie de dibujos o imágenes, así como las de opción múltiple.

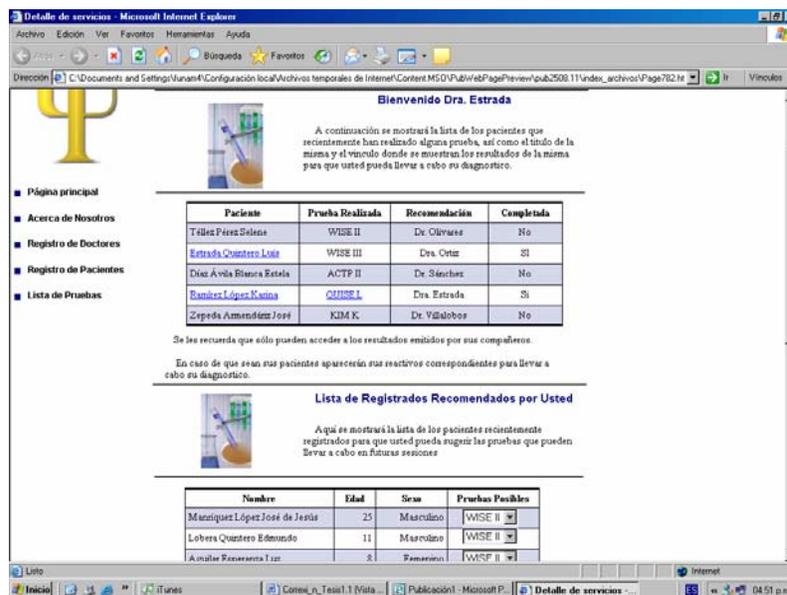


Figura 6.7 Ejemplo de los resultado de las pruebas psicológicas

Todos los pacientes que sean registrados, como ya se mencionó previamente, deberán ser recomendados por un psicólogo especializado, con conocimientos previos elementales sobre el mismo. Cuando los psicólogos registrados accedan al sitio aparecerá la ventana donde aparezca una tabla donde los psicólogos podrán observar una lista de los resultados de los pacientes registrados, quien los recomendó y si han finalizado la resolución de sus pruebas.

Dentro de la misma tabla aparecerán dos posibles vínculos. Uno donde podrán revisarse los resultados de los pacientes, emitidos por los otros colaboradores, este vínculo será sobre el nombre del usuario. El segundo de los vínculos aparecerá sobre el nombre de la prueba realizada y en este caso solo podrá obtener acceso el psicólogo que lo recomendó ya que será este el que lleve a cabo su diagnóstico, conociendo todos los reactivos finales de la prueba.

En la segunda tabla aparecerán los nuevos pacientes registrados que hayan sido recomendados por el propio psicólogo, así como una lista de las posibles pruebas relacionadas con la edad y sexo del paciente, para que una vez aprobada la prueba, el paciente registrado pueda realizar la misma cuando él lo desee.

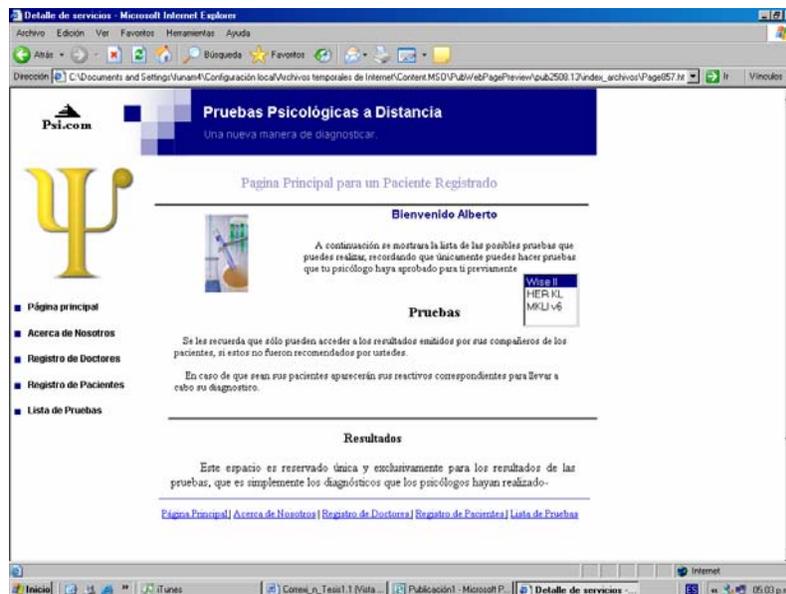


Figura 6.8 Página principal de cualquier usuario.

En la página principal a la que tenga acceso un usuario registrado contendrá un listado de las posibles pruebas que podrá realizar cuando él lo crea conveniente. Para que esta lista aparezca se requiere previamente el consentimiento de su psicólogo. Para ello el usuario debe de estar consciente que una vez registrado tendrá que esperar un periodo de 2-3 días para llevar a cabo las pruebas porque es indispensable su aprobación para el desarrollo de la misma.

Si un usuario que ya haya realizado sus pruebas, aparecerá un vínculo que demostrará el resultado de su prueba pero únicamente con el diagnóstico de su psicólogo así como las recomendaciones que el mismo sugiera.

7. Conclusiones



El Diplomado “Desarrollo e Implementación de Sistemas en Software Libre” nos ha brindado los fundamentos necesarios para poder realizar cualquier tipo de sistema a un bajo costo y con la seguridad mínima para poder llevar un control del mismo.

El presente proyecto se basa en todos los elementos aprendidos dentro de este Diplomado, desde conocimientos básicos y sencillos, como la programación en HTML, hasta la programación de técnicas de encriptación para la seguridad del sistema.

El sistema de Pruebas Psicológicas a Distancia es una herramienta capaz de brindar una nueva visión en la manera de llevar a cabo el desarrollo de una prueba. Permitirá establecer contactos de orden médico, dejando a un lado las antiguas terapias donde se debía tener la presencia de los involucrados al momento de realizar alguna de las pruebas.

El alcance que puede tener este sistema no tiene límites. Gracias a los elementos brindados por los diferentes programas en los que se basa su arquitectura. Desde la plataforma Linux que ofrece cierta “estabilidad” hasta el conjunto de los programas: Apache, la programación en PHP, HTML y Java Script, combinando estos últimos con MySQL, que en conjunto brindan seguridad y estabilidad.

En general este sistema es una innovación, que pretende revolucionar la terapia tradicional con tecnologías y herramientas de la presente era, llevando a cabo diferentes pruebas desde cualquier parte del mundo, para alcanzar lo que se ha denominado como “Diagnósticos del Siglo XXI”

8. Bibliografía

GALEANO Gil Germán, GALEANO Alonso José Carlos,
HTML 4
Ed. Anaya Multimedia, Madrid España, 2000

CHARTE Ojeda Francisco
Proyectos Profesionales PHP 5
Ed. Anaya Multimedia, Madrid España, 2004

C. WU, Thomas
Introducción a la Programación Orientada a Objetos con Java
Ed. Mc Graw Hill Mexico, 2002

WELLING, Luke; THOMSON, Laura
MySQL Tutorial
Ed MySQL Press, Estados Unidos, 2003

Internet

www.wikipedia.com

www.javacastellano.com

www.linux.org