Universidad de Las Palmas de Gran Canaria

Escuela Universitaria de Informática

DISEÑO E IMPLEMENTACIÓN WEB PARA EL ACCESO EN LÍNEA DE PRENSA CANARIA DIGITALIZADA

TOMO I

FCO JAVIER TORRES BETANCOR

Las Palmas de Gran Canaria, Septiembre 2006

Universidad de Las Palmas de Gran Canaria
Escuela Universitaria de Informática

Proyecto fin de carrera de la Escuela Universitaria de Informática de la Universidad de Las Palmas de Gran Canaria presentado por el alumno:

FCO JAVIER TORRES BETANCOR

Título del Proyecto: Diseño e implementación Web para el acceso en línea de prensa canaria digitalizada

Tutor: Beatriz Correas Suárez
Co-tutor: Víctor M. Macias Alemán

DEDICATORIA

A mis padres

AGRADECIMIENTOS

- A mis tutores por su buen hacer durante el proyecto.
- A mis hermanas Susi y Fátima, por insistir en hacer esto posible.
- A mis amigos que me han dado todo su apoyo.
- A mi novia Cristina, por no dejar de animarme ni un instante y por aguantarme durante tanto tiempo.

INDICE GENERAL

INDICE DE ILUSTRACIONES

1. Introducción.

La creciente demanda por parte de los miembros de la comunidad universitaria de querer tener acceso a todo tipo de información de la forma más eficiente posible, unido a los avances tecnológicos de los últimos años, sobre todo en conectividad y en capacidad de almacenamiento a precios cada vez mas competitivos, ha tenido como consecuencia inmediata la proliferación de sitios Web donde los usuarios pueden acceder a una cantidad y a un tipo de información que hasta hace poco era impensable. Ya es conocido el dicho: "Si no se encuentra en Internet, es que no existe". Sin embargo esta frase dista mucho todavía de ser completamente cierta si nos referimos a documentos históricos, y mucho más si la historia a la que nos referimos es la de una comunidad como la Comunidad Canaria, desde siempre aislada geográfica e históricamente. La aparición del fenómeno de Internet ha sido sin duda de gran ayuda para disminuir en gran medida dicho aislamiento, incluso entre las propias islas. Las instituciones públicas canarias han ido aportando mecanismos para mejorar la accesibilidad de los ciudadanos a la información, y la Universidad de Las Palmas de Gran Canaria no ha sido una excepción.

Desde el punto de vista informático, Internet ha supuesto una revolución en la creación de software, surgiendo diferentes lenguajes de programación y entornos de desarrollo orientados a Internet, y los avances del hardware han permitido que dicho software mejore su potencialidad. La Universidad de Las Palmas de Gran Canaria ha procurado poner los medios necesarios para cubrir dicha demanda informática.

Como alumno siempre he considerado la Biblioteca Universitaria como uno de los pilares de la universidad, y el préstamo de libros como uno de los servicios más importantes. Sin embargo la Bibliotecatambién ha evolucionado a la par que la informática, y ya no sólo se dedica al préstamo

de libros. Con los años se han creado nuevos servicios como la Hemeroteca, una zona destinada a la consulta de revistas, boletines y prensa actual e histórica. Otro servicio es el de Digitalización, que ha permitido escanear todos esos documentos y guardarlos en formato electrónico, reduciendo drásticamente el espacio físico de almacenamiento y permitiendo la consulta electrónica de los mismos. A pesar de estos avances la accesibilidad ha seguido siendo limitada debido a los pocos medios físicos para la consulta "in-situ" y el horario restringido para ofrecer el servicio de consulta. Fue entonces cuando se creó la Biblioteca Digital, un proyecto Web que permite la consulta en línea de documentación digitalizada entre los que se encuentran proyectos fin de carrera, tesis doctorales, fotografía histórica de canarias, etc.

1.1 Descripción general.

A mediados de 1996, la Biblioteca Universitaria de Las Palmas de Gran Canaria puso en marcha un plan a largo plazo, en coordinación con otras entidades interesadas en participar, a fin de salvaguardar la memoria hemerográfica de nuestra región -en un estado lamentable de conservación en la mayor parte de los casos-, y ponerla a disposición de los investigadores y estudiosos por los más modernos medios técnicos. Dicho proyecto sigue, tanto buscando ejemplares desaparecidos para completar determinadas fechas, como para incluir otros títulos significativos en la historia del periodismo canario. Así, una empresa especializada ha estado llevando a cabo en estos últimos años la digitalización masiva de la prensa regional de Canarias desde 1996 y toda la prensa histórica de Gran Canaria desde 1893. Almacenadas en 2370 CD-ROM y DVD-ROM, este corpus puede ser consultado en la Hemeroteca de la Biblioteca General de manera local y durante el horario de atención al público, y continúa creciendo a medida que

se van publicando y digitalizando los ejemplares; sin embargo el objetivo de este proyecto ha sido hacer llegar esta documentación a todos de la mejor manera posible

Se planteó entonces integrar dentro de este proyecto Web la posibilidad de poder consultar "on-line" dicho corpus, para lo cual hacía falta un esfuerzo desde varios frentes. Quizás el más importante de todos sea el de medios técnicos, debido al tamaño de dicho corpus y por lo tanto a la capacidad de almacenamiento que se necesitaría para albergarlo, además de la capacidad de ancho de banda necesario para que dichas consultas no resultaran lentas y tediosas. Han tenido que pasar varios años para que dichos medios fueran viables y que la realización de este proyecto tuviera razón de ser.

El planteamiento inicial era bien sencillo: crear un sitio Web para la consulta en línea de la Prensa Canaria Digitalizada. Sin embargo había que debatir como se iba a hacer, sobre todo teniendo en cuenta el corpus existente y su formato, para discutir los distintos caminos a seguir, sin perder de vista en ningún momento el objetivo final.

1.2 Objetivos del proyecto:

Víctor Macias, bibliotecario jefe de automatización y responsable de la Biblioteca Digital, lanzó al aire una idea que hemos recogido para realizar este proyecto, consistente en poner en marcha una plataforma "on-line" para toda la prensa digitalizada, consiguiendo así que el usuario autorizado pudiera acceder, desde su aula, despacho, centro de investigación o desde su propio domicilio (utilizando el servidor de autenticación de la ULPGC), a la totalidad de la prensa regional digitalizada. Ello supondría el acceso de toda la Comunidad Universitaria de la ULPGC a uno de los mayores corpus de

documentación electrónica facsímil en línea en nuestro país, ya que son más de 3.100.000 páginas las que podrían ser consultadas.

1.3 Material utilizado:

Nos hemos servido de los recursos hardwarey software existentes en la Biblioteca Digital, que son los siguientes:

1.3.1 Hardware:

Servidor DELL con doble procesador, 8 Gb de memoria RAM y sistema de almacenamiento secundario POWERVAULT configurado en RAID 5, con capacidad para almacenar 4,2 Terabytes de información.

1.3.2 Software:

Sistema Operativo Linux Red Hat, servidor Web Apache, configurado para implementar páginas Web en PHP y conexión a bases de datos con MySql, así como conexión FTP para acceso remoto.

1.4 Metodología de desarrollo:

Este punto se decide en base a como se encuentra la documentación digitalizada hasta el momento (imágenes TIFF hasta 2003 inclusive, ficheros PDF, OCR en formato ACCESS y archivos de texto sueltos), ya que es posible que se tengan que realizar conversiones de formato compatibles con los navegadores de Internet, y tomar decisiones respecto a que tecnología Web utilizar y que caminos tomar en la implementación, además de elaborar las etapas de diseño y desarrollo en función de dichas decisiones.

1.4.1 Etapas y planificación temporal estimada

Etapa 1. Análisis. Estudio y valoración de las especificaciones y requerimientos necesarios para alcanzar los objetivos del proyecto. *(Duración estimada 48 horas).*

Etapa 2. Diseño. Diseño del software necesario para la elaboración del proyecto. Se hará un especial énfasis en diseñar una interfaz de usuario amigable y eficiente. *(duración estimada 32 horas).*

Etapa 3. Implementación. Etapa principal del proyecto que a su vez se divide en los siguientes apartados:

Etapa 3.1. Repositorio. Idear una forma de crear y cargar toda la documentación en el servidor para su posterior consulta. *(Duración estimada 32 horas).*

Etapa 3.2. Interfaz web. Diseñar la interfaz de usuario de acuerdo con la imagen corporativa del sitio web de la Biblioteca Digital. *(Duración estimada 16 horas).*

Etapa 3.3. Servidor web. Configuración del servidor web para la implementación del proyecto, asi como la instalación y configuración de la tecnología web que se utilice. *(Duración estimada 28 horas).*

Etapa 3.4. Implementación del entorno web. Creación del sitio web para la consulta por periódico y fecha. Durante esta etapa se utilizan opciones de implementación para elegir la mas adecuada *(Duración estimada 100 horas).*

Etapa 3.5. Periodo de pruebas. Tiempo para que los usuarios pongan a prueba la aplicación y envien sus sugerencias para mejoras futuras.*(Duración estimada 50 horas).*

Etapa 4. Resultados y Conclusiones. En esta etapa se realizará una validación de todo el trabajo realizado en el proyecto, y sus aplicaciones futuras, haciendo énfasis en el estudio sobre la viabilidad para poder realizar búsquedas de texto en linea a partir de los OCRs locales de que se disponen, realizando pruebas al respecto. *(Duración estimada 34 horas).*

Temporización global del proyecto:

El tiempo global estimado para el desarrollo completo del proyecto es de 330 horas.

La Biblioteca Universitaria de la ULPGC es la primera de España en ofrecer un sistema de acceso en línea a toda la prensa digitalizada publicada en una región, en nuestro caso Canarias, para su consulta por nuestra Comunidad Universitaria las 24 horas del día los 365 días del año.

2. La Biblioteca Digital.

La Biblioteca Digital es una plataforma Web que surgió a principios de 2003 desde la Biblioteca Universitaria de la ULPGC para el acceso en línea a la información científicas que pudiera ser relevante para estudiantes, docentes e investigadores.

Toda la documentación que se puede consultar se estructura en cuatro apartados:

- **Bases de datos científicas**

El usuario accede tanto al servidor Doramas, que gestiona las bases de datos suscritas o compradas en CD-ROM o DVD-ROM y que incorpora, entre otros, acceso a recursos completos de jurisprudencia y legislación, como *Aranzadi*; documentos antiguos o literarios, como el *Teatro Español del siglo de Oro* o *English Poetry Plus*, como a aquellos otros recursos del mismo género que están accesibles sólo en servidores externos contratados como *ISI Proceedings, PCI FullText, IEEE Xplore*, etc.

- **Revistas electrónicas**

Permite consultar revistas suministradas por grandes distribuidores de información electrónica como *EBSCO, Elsevier, Emerald* o *Wiley*, en total más de 2700 títulos de publicaciones de investigación, conjuntamente con otros muchos enlaces disponibles en la red mundial.

- **Monografías e informes**

Recoge tanto documentación elaborada por la propia ULPGC, como enlaces a recursos científicos externos a textos sobre materias de nuestro interés como Universidad.

- **Tesis doctorales y PFCs**

Digitalización propia y con acceso vía *UMI-ProQuest* a todas las tesis doctorales y proyectos fin de carrera leídos en nuestra Universidad en formato *PDF.*

A ellos hay que sumar un quinto apartado que, por su especial significación merece tratamiento aparte. Se trata de la Memoria Digital de Canarias (mdC), destinado a reunir todo tipo de documentación significativa, ya sea producida en Canarias, sobre nuestro Archipiélago o realizada por los naturales de las Islas en forma de textos, imágenes, audio o vídeo.

Toda esta documentación estará en constante proceso de ampliación y mejora, con lo cual recomendamos visitar con frecuencia la sección de Novedades, a fin de estar al corriente de las últimas actualizaciones en cada uno de los apartados.

2.1 Consulta "in-situ" de la prensa, la Hemeroteca.

Como hemos comentado en el capítulo anterior, el servicio de Hemeroteca permite al usuario consultar el corpus de la prensa regional, pero con una serie de limitaciones que son las siguientes:

- Acceso local.
- Selección manual de la prensa que se quiere consultar.
- Pocos puestos para atender la demanda.
- Horario limitado de atención al público.

Se pensó entonces en integrar dicho corpus en la Biblioteca Digital para poder salvar estos inconvenientes y ofrecer así un mejor servicio al usuario. Nació la idea de realizar el acceso en línea del corpus, pero había que tener en cuenta diferentes cuestiones antes de ponerlo en marcha:

- Comprobar el estado actual del corpus
- Analizar el entorno en el que se ha desarrollado la Biblioteca Digital
- Conocer los requerimientos hardwarey software para la realización del proyecto.
- Decidir el formato electrónico a emplear
- Idear el diseño Web y planificar su implementación.

A continuación se detallan los contenidos y formatos del corpus de la prensa regional.

2.2 Corpus de la prensa

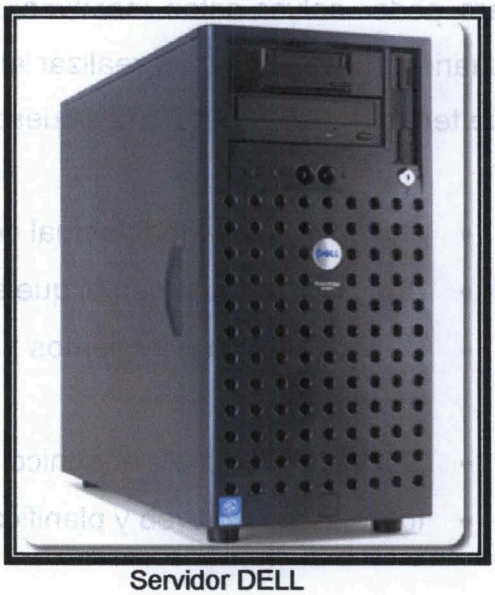
Se empezó a crear en 1996 mediante convenios de colaboración entre la Biblioteca Universitaria, los periódicos regionales que quisieron entrar en el proyecto mediante el préstamo de los ejemplares, y la empresa especializada que ha sido la encargada de realizar la digitalización.

Todo este corpus representa actualmente casi 2 Terabytes de información y más de 3 millones de páginas, con un crecimiento anual aproximado de 180.000, que serían accedidas por intranet por 25000 usuarios potenciales que componen la comunidad Universitaria. Este

volumen de información y de demanda hacia necesaria una gran ampliación del servidor, tanto en capacidad de almacenamiento (hasta ahora sólo 200 Gigabytes) como en capacidad de proceso para atender dicha demanda.

Así se realizó la compra de un Servidor DELL con las siguientes características:

- 8 Gigabytes de memoria RAM
- doble microprocesador Intel Pentium IV
- Sistema de Almacenamiento Secundario externo



Servidor DELL

Sistema de almacenamiento POWERVAULT

El sistema de almacenamiento secundario externo POWERVAULT consta de una carcasa exterior con 32 discos duros SCSI y una capacidad total de 4 Terabytes., configurados en RAID-5.

El coste del hardware unido al entorno software con el que ya contaba la Biblioteca Digital llevó a utilizar la plataforma de software libre llamada "LAMP" que consta de las siguientes aplicaciones:

Logotipo de la Plataforma Web LAMP

- Sistema operativo Linux Red Hat..
- Servidor Web Apache.
- Lenguaje de programación PHP.
- Motor de bases de datos MySql.

Aparte de este conjunto de aplicaciones, también se instaló un servidor FTP para realizar conexiones remotas, y que serviría para realizar la carga del corpus en el servidor.

2.3 Formato de digitalización

En la actualidad el corpus se encuentra almacenado en el servicio de Hemeroteca en 2750 CD-ROM y DVD-ROM, entre los que hay que distinguir dos fases bien diferenciadas en formatos diferentes:

Prensa digitalizada hasta el año 2003

- Formato TIFF 1bit ITU-TT.6
- Almacenamiento en CD-R estandar
- Estructura de directorios
- Un fichero por página
- Visor monopuesto
- OCR en archivos mdb y txt.

Prensa digitalizada desde el año 2004 en adelante.

- DVD-R estandar 4,7 Gb.
- PDF multipagina JBIG-2.
- OCR imagen orig. texto oculto, 300 ppp.
- Un fichero por periódico-dia.
- Sintaxis de nombres de ficheros autoidentificable.
- Sin estructura de directorios.

Una vez analizado el contenido del corpus, era la hora de evaluar los formatos de los que disponemos, enumerando las ventajas e inconvenientes a la hora de estudiar la viabilidad de ponerlo en línea.

2.2.2. Ventajas e inconvenientes del formato

Teniendo en cuenta que existen diferentes formatos, vamos a nombrar las ventajas y los inconvenientes de cada uno de ellos.

Para el corpus creado hasta el año 2003:

Ventajas:

- Formato TIFF de los ficheros mejora la calidad de la imagen.
- La estructura de directorios ayuda a identificar el periódico.
- El peso de los ficheros es relativamente bajo (media de 500 Kilobytes).

Inconvenientes:

- El formato TIFF no es estándar en Internet.
- El alto numero de ficheros (uno por pagina) .

- La estructura de directorios exige mas proceso para encontrar el periódico deseado.

Para el corpus creado a partir del año 2004:

Ventajas:

- Formato PDF bastante extendido.
- Un fichero por periódico.
- Sin estructura de directorios
- Sintaxis del fichero ayuda a identificarlo.

Inconvenientes:

- El peso de los ficheros es demasiado grande (más de 50 Megabytes en algunos casos).

2.3. Viabilidad de la consulta en línea

Viendo los formatos y después de evaluar las ventajas y los inconvenientes de cada uno de ellos, es el momento de plantear la viabilidad del proyecto para luego tomar las decisiones oportunas.

El formato TIFF de los ficheros de la prensa hasta el año 2003, que representa la parte mayoritaria del corpus, lleva a plantear diferentes alternativas para poder ser consultados en línea:

- Convertir los ficheros a formato PDF.
- Reestructurar los directorios existentes.
- Renombrar los ficheros y disminuir su número.

Estas medidas fueron planteadas para unificar esta parte del corpus con la que tenemos a partir del año 2004, pero presentan algunas dificultades para su realización:

- La conversión a PDF seria ineficaz en cuanto al peso de los ficheros y la búsqueda de texto en los mismos (conversión desde TIFF).

- Medios técnicos y humanos insuficientes para acometer la conversión de millones de ficheros TIFF almacenados en mas de 2000 CD-ROM.

Queda descartada por tanto la conversión global a PDF, pero habría que tratar de homogeneizar la consulta de toda la prensa para que resulte transparente al usuario independientemente del formato que se quiera consultar.

2.4. Toma de decisiones.

La primera decisión ya está tomada, mantener el formato TIFF, pero hay que decidir como mostrar el periódico que el usuario desea consultar de forma que sea compatible en Internet.

La siguiente decisión es la carga de todo el corpus en el servidor. Para ello se configura el almacenamiento secundario POWERVAULT de 4 Terabytes en modo RAID 5, que permite recuperarse de los fallos de almacenamiento que se puedan producir, aunque afecte a la capacidad total de almacenamiento. Asimismo se instala en el servidor un servidor FTP seguro para permitir el acceso y carga remota desde la Hemeroteca de todo el corpus, siguiendo unas normas previamente establecidas.

Por último se decide que el formato para visualizar todos los ficheros en línea sea el PDF para conseguir la mayor similitud posible con los periódicos digitalizados a partir del año 2004. Así que tenemos que idear un

mecanismo que permita que las imágenes que inicialmente se encuentran almacenadas en formato TIFF se muestren en línea en formato PDF.

Debido a que la Biblioteca Digital ya se encuentra operativa y funciona bajo el servidor Web Apache y ha sido desarrollada en Lenguaje PHP, nos decantamos por el mismo entorno para conseguir la integración del sitio Web que vamos a crear para el acceso en línea de la prensa canaria digitalizada.

Una vez se ha analizado la situación del corpus y se han tomado las decisiones necesarias, llega el momento de acometer el proyecto Web, desde la carga del corpus de la prensa, pasando por el diseño del sitio Web, de su implementación y de la integración del mismo en la Biblioteca Digital.

3. Diseño del Sitio Web

El diseño del sitio Web debía ser acorde con el entorno existente en la Biblioteca Digital, respetando la imagen corporativa de la misma y cumpliendo las normas de accesibilidad ya existentes en el proyecto Web.

3.1 Objetivos.

Los objetivos a la hora de realizar el diseño Web son los siguientes:

- Diseño corporativo de acuerdo con la Biblioteca Digital.
- Interfaz de usuario intuitivo.
- Optimizar el proceso de búsqueda.

3.2 Software necesario.

Siempre teniendo presente los recursos de los que disponemos y las decisiones adoptadas y explicadas en el capítulo anterior, El software necesario para la implementación del sitio Web es el siguiente:

- Lenguaje PHP configurado en servidor Web Apache.
- Editor PHP para la generación de código.
- Librería PHP con funciones para generar archivos PDF (pdflib)

3.3. Metodología

Teniendo claro el primero de los objetivos enumerados anteriormente, decidimos centrarnos en el diseño del sitio Web desde la última página Web hasta la primera, es decir, empezamos a diseñar en primer lugar la página Web que va a mostrar el periódico que el usuario elija.

Para ello hemos tenido en cuenta las decisiones tomadas al respecto del formato que hemos elegido para mostrar las páginas que se encuentran almacenadas en formato TIFF (siendo además el formato mayoritario del corpus) y que decidimos que se mostraran en formato PDF.

De esta forma hemos procurado que el diseño de dicha página Web se asemeje al de un fichero PDF, donde en su parte izquierda tenemos un índice de páginas y a la derecha tenemos la visualización de la página seleccionada.

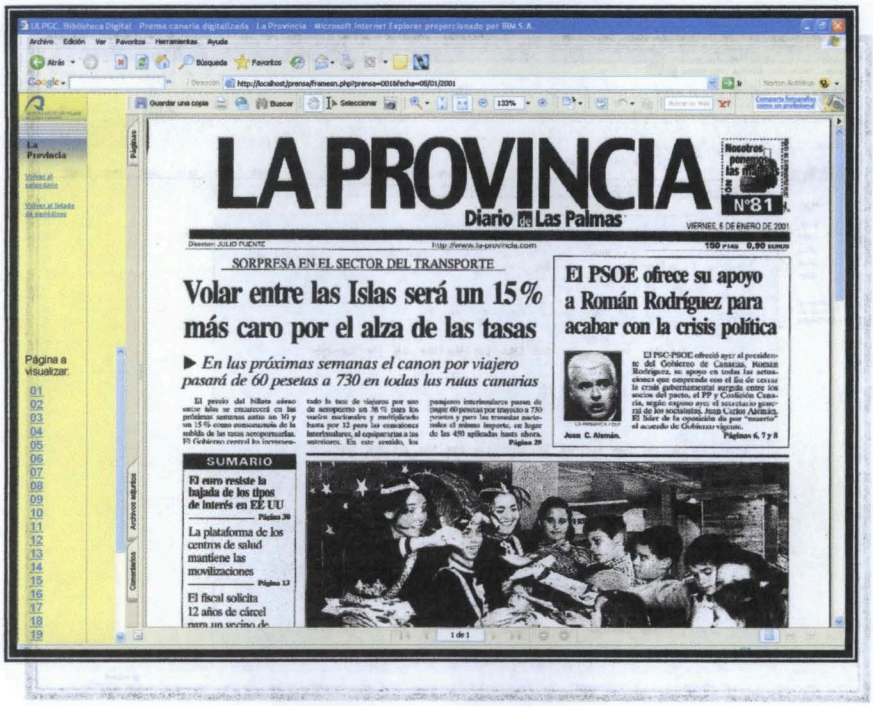
Con este diseño el usuario no notará muchas diferencias cuando decida consultar prensa a partir del año 2004 con respecto a la prensa anterior a ese año, ya que se encuentran almacenados en formato PDF y son mostrados directamente en el navegador Web.


LA PROVINCIA
Diario de Las Palmas
Nº 81
VIERNES, 5 DE ENERO DE 2001
SORPRESA EN EL SECTOR DEL TRANSPORTE
Volar entre las Islas será un 15% más caro por el alza de las tasas
► En las próximas semanas el canon por viajero pasará de 60 pesetas a 730 en todas las rutas canarias
El PSOE ofrece su apoyo a Román Rodríguez para acabar con la crisis política
SUMARIO
El euro resiste la bajada de los tipos de interés en EE UU
La plataforma de los centros de salud mantiene las movilizaciones
El fiscal solicita 12 años de cárcel para un vecino de Tinajo, acusado de

Ejemplo de un fichero PDF.

La imagen muestra un fichero PDF que hemos generado, insertando una imagen TIFF de una página de prensa, que nos sirve de referencia para el diseño de la Web mencionada anteriormente.

Con independencia del formato almacenado, decidimos también incluir en la esquina superior izquierda el nombre del periódico elegido, así como enlaces de regreso a páginas Web anteriores. En la siguiente imagen mostramos la página Web que muestra la misma página de prensa de la imagen anterior.

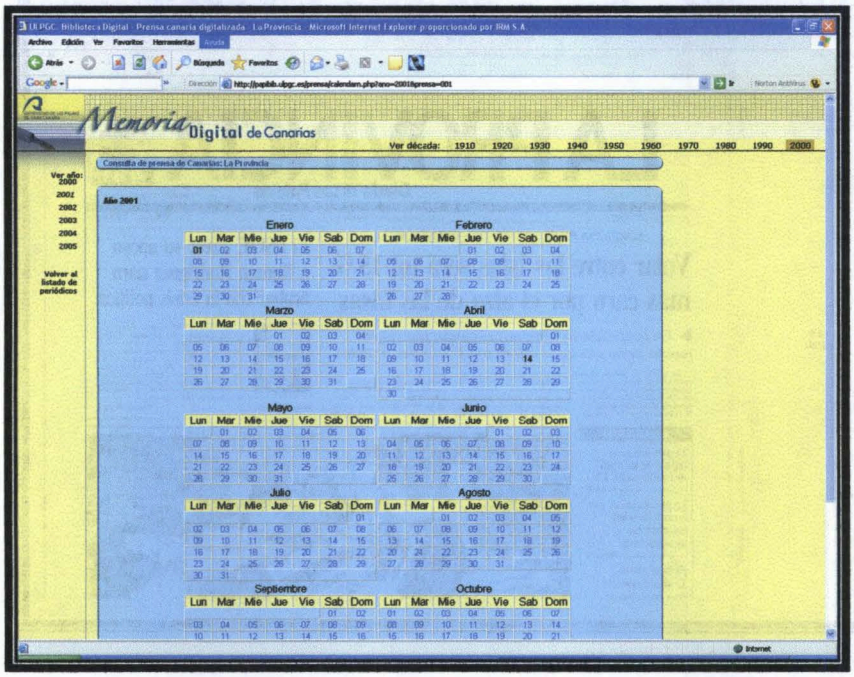


Página Web generada.

La estructura de directorios del repositorio (que explicaremos en el siguiente capítulo) nos dio la idea de facilitar la consulta de la prensa por fechas mediante la generación de un calendario anual que muestra los días de ese año en los que existe prensa disponible, y con la posibilidad de cambiar de año y/o de década en la misma ventana del calendario.

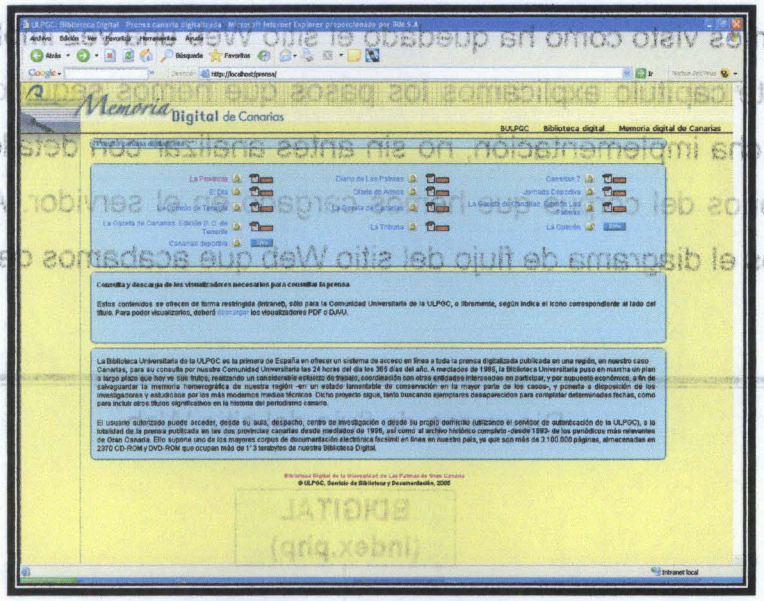
El usuario podrá visualizar un periódico pulsando sobre el número del día, por lo que los números representan hipervínculos a los periódicos, pero solamente de aquellos para los que el periódico está disponible. Los días en los que no hay periódico no representan ningún hipervínculo y aparecen en un color diferente para que el usuario sepa de manera intuitiva en que días del calendario hay disponibilidad de prensa y en que días no tenemos periódico.

Se muestra a continuación una imagen del calendario.



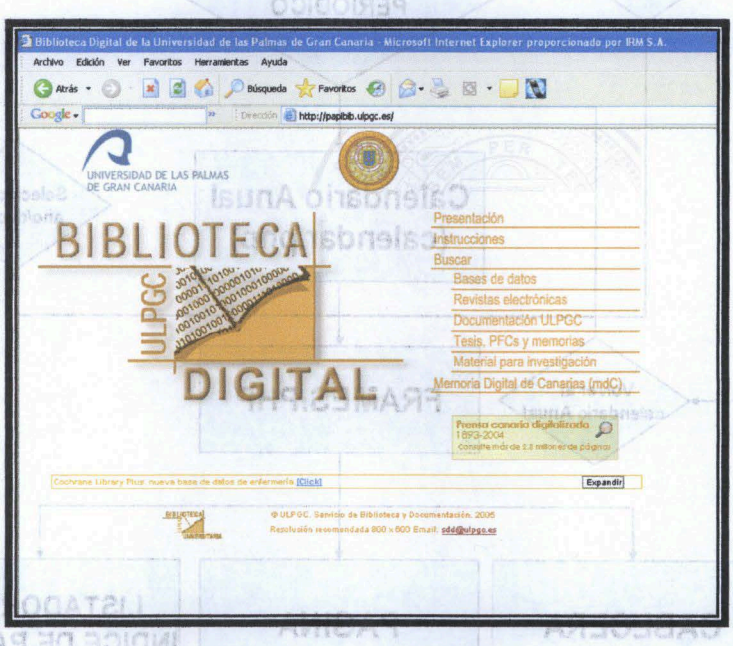
Calendario Anual generado en PHP

El diseño de la página principal se ha dejado para el final. En ella le explicamos al usuario un poco de historia sobre el corpus de la prensa y su razón de ser, y mostramos la lista de periódicos regionales disponibles para ser consultados. Cada nombre de periódico representa un hipervínculo que llevará al usuario al calendario correspondiente al último año de publicación del periódico seleccionado.



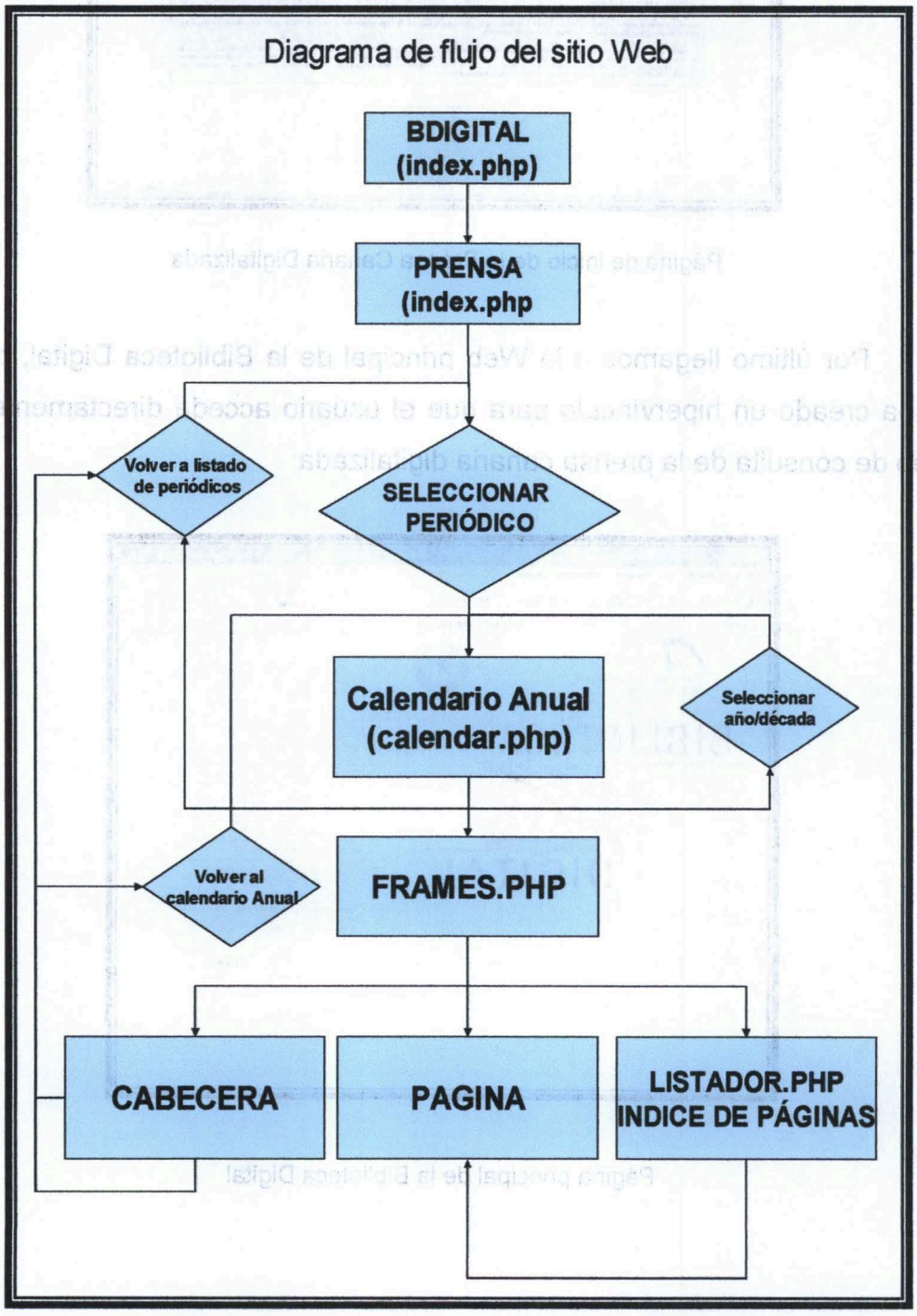
Página de inicio de la Prensa Canaria Digitalizada.

Por último llegamos a la Web principal de la Biblioteca Digital, donde se ha creado un hipervínculo para que el usuario acceda directamente a la Web de consulta de la prensa canaria digitalizada:



Página principal de la Biblioteca Digital

Hemos visto como ha quedado el sitio Web una vez implementado. En el siguiente capítulo explicamos los pasos que hemos seguido para llegar a realizar dicha implementación, no sin antes analizar con detalle la estructura de directorios del corpus que hemos cargado en el servidor. A continuación mostramos el diagrama de flujo del sitio Web que acabamos de explicar.



4. Implementación.

El diseño estaba decidido, el software necesario preparado y la metodología a seguir estaba bien definida. Era el momento de desarrollarlo.

Pero todavía quedaba un asunto muy importante que solucionar y de vital importancia para la implementación del sitio Web, la carga de todo el corpus de la prensa en el servidor, la creación del repositorio.

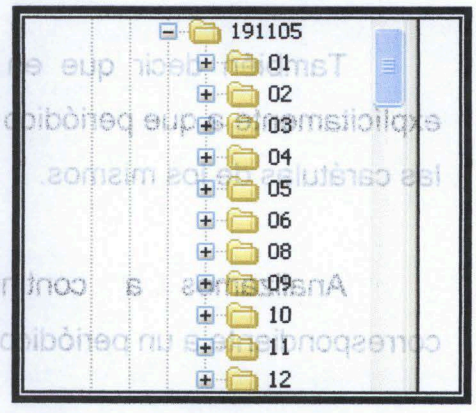
4.1 El repositorio.

Era necesario comprobar los contenidos de los CD's y DVD's antes de crear el repositorio y realizar la carga en el servidor. Sobre todo el principal objetivo era crear un sistema de directorios que ayudara a identificar unívocamente a un periódico de una fecha determinada, facilitándonos la búsqueda dentro del mismo. Había que analizar detalladamente la estructura de directorios del corpus de la prensa.

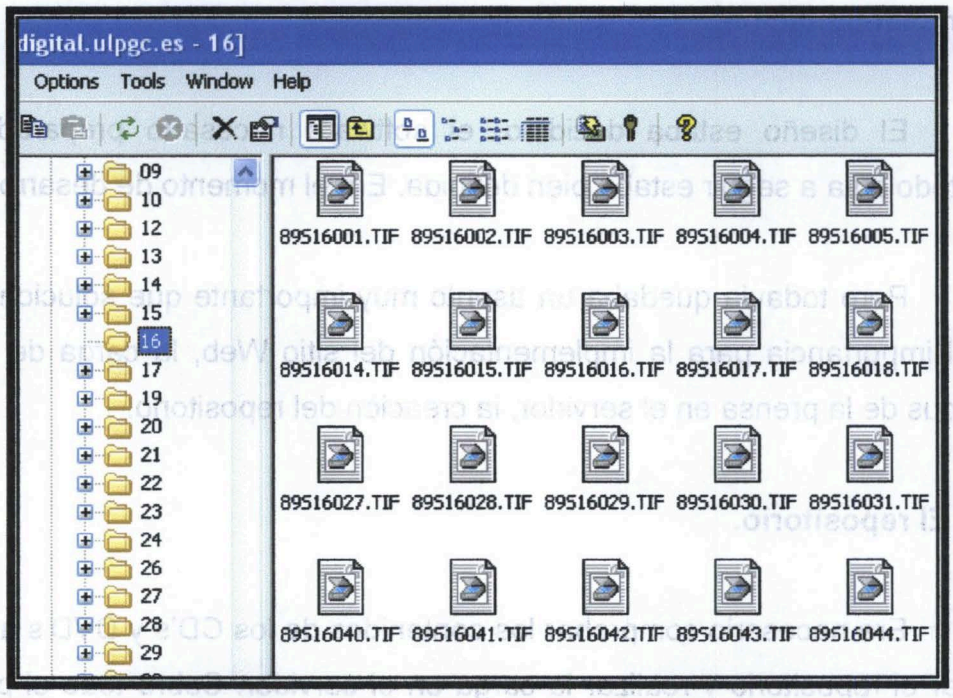
4.1.1 Estructura de directorios.

En primer lugar analizamos el contenido de un CD-ROM correspondiente a un periódico anterior al año 2004.

La estructura de directorios muestra una carpeta cuyo nombre consta de 6 cifras que corresponden al año y al mes. Esta carpeta contiene una serie de subcarpetas, nombradas con 2 cifras cada una que corresponden a los días.



Contenido de un CD



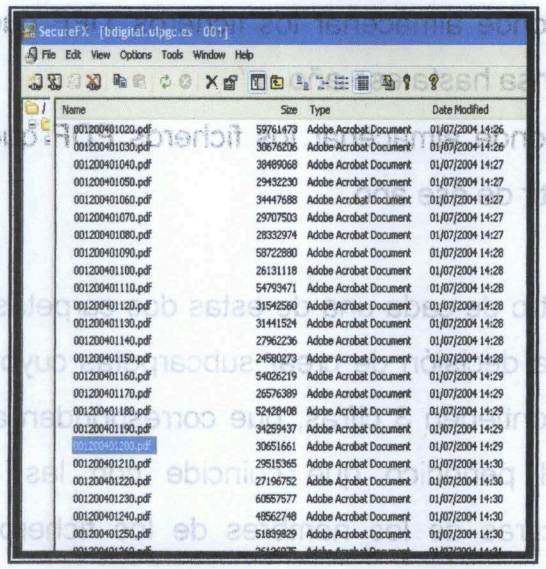
Contenido de una subcarpeta

Cada una de estas subcarpetas contiene los ficheros en formato TIFF de las páginas del periódico que corresponde al día, mes y año de la carpeta y subcarpeta seleccionada.

Observar que los nombres de los ficheros TIFF contienen en sus tres últimos caracteres el número de la página que corresponde con la imagen que contiene el fichero.

También decir que en los contenidos de los cd's no se identifica explícitamente a que periódico pertenecen los ficheros, identificados éstos en las carátulas de los mismos.

Analizamos a continuación el contenido de un DVD-ROM correspondiente a un periódico con fecha a partir del año 2004.



Contenido de un DVD

Como podemos observar en esta imagen, no existe estructura de directorio, los ficheros se han almacenado directamente en la raíz del DVD-ROM. Estos ficheros se encuentran en formato PDF y sus nombres identifican en este caso el periódico que lo contiene como la fecha del mismo.

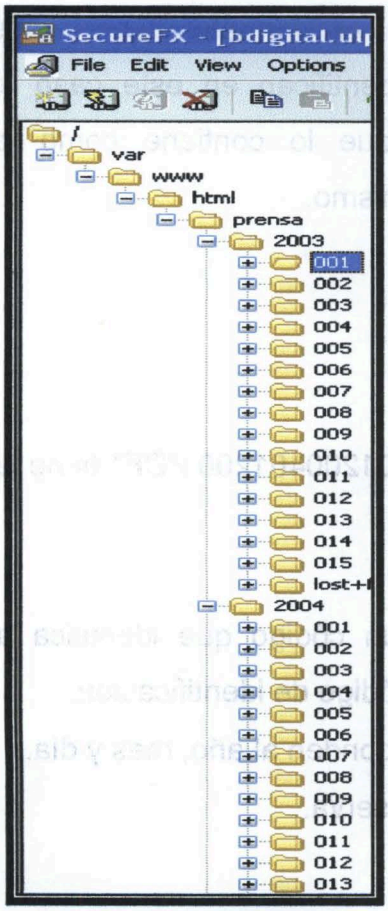
Por ejemplo, el fichero con el nombre "001200401200.PDF" tiene el siguiente significado:

- Las 3 primeras cifras, "001", es un código que identifica al periódico. Cada periódico tiene su código de identificación.
- Las 8 siguientes, "20040120" corresponden al año, mes y día.
- La última cifra se ha dejado como reserva.

Una vez hemos comprobado el contenido de los cd's y dvd's con sus diferentes formatos de almacenamiento, había que pensar en crear un repositorio con un sistema de directorios lo más adecuado posible para que la carga del corpus en el servidor fuera amigable y a su vez sirviera de ayuda para realizar la búsqueda del periódico que se quisiera consultar.

Una condición imprescindible era la de poder distinguir entre los dos formatos, TIFF y PDF. Por tanto decidimos crear inicialmente en el servidor dos carpetas:

- 2003: Nombre de la carpeta donde almacenar los ficheros TIFF que contienen las páginas de la prensa hasta ese año
- 2004: Nombre de la carpeta donde almacenar los ficheros PDF que corresponden a la prensa a partir de ese año.



Directorio del Corpus

Dentro de cada una de estas dos carpetas, tomamos la decisión de crear subcarpetas cuyos nombres contienen 3 cifras, que corresponden al código del periódico que coincide con las 3 primeras cifras de los nombres de los ficheros PDF que explicamos anteriormente.

En la imagen de la izquierda puede verse claramente esta estructura de directorios creada para el repositorio.

Cada carpeta servirá para almacenar el contenido de los CD's que corresponden al periódico cuyo código coincide con el nombre de la carpeta.

Por ejemplo:

Sabiendo que el código "001" identifica al periódico "La Provincia", los CD's de este periódico hasta el año 2003 se cargarán en la carpeta "001" que cuelga de la carpeta "2003", mientras que los DVD's del mismo periódico que corresponden a fechas a partir del año 2004 se cargarán en la carpeta "001" que cuelga de la carpeta "2004".

Subcarpeta del Corpus

Con este sistema de directorios, la carga en el servidor consiste simplemente en ir copiando los contenidos de los CD's Y DVD's en las carpetas que corresponden a cada periódico.

Este proceso fue realizado por el servicio de Hemeroteca, cuyo personal fue instruido previamente para que hicieran la carga correctamente, mediante una conexión FTP segura al servidor. Fueron necesarios muchos días para copiar los mas de 2300 CD's y DVD's que componen el corpus de la prensa.

4.1.2 Método de búsqueda

Desde el principio nos percatamos de que la estructura de directorios y ficheros del corpus de la prensa nos serviría de ayuda para realizar consultas por periódico y fecha. Mientras el personal de Hemeroteca fue cargando los primeros CD's del corpus en el servidor, instalamos un servidor de pruebas y cargamos el mismo sistema de directorios del repositorio con unos pocos periódicos. Así ya podíamos comenzar a programar esta búsqueda en PHP.

En la página principal, el usuario pulsa sobre el nombre del periódico que desea visualizar, Cada nombre de periódico es un hipervínculo como este:

```
<a href="/prensa/calendar.php?elije_a%F1o=1&
prensa=001&Enviar=Visualizar" title="Ver
<? nombre_prensa_id(1); ?>"><? nombre_prensa_id(1); ?></a>
```

Podemos ver código HTML estándar y código en PHP embebido. Cuando el usuario pulsa sobre el nombre del periódico, se carga el fichero "calendar.php", al que se le pasan una serie de variables, entre las cuales está "prensa", con el valor del código del periódico que el usuario ha seleccionado.

El fichero genera el calendario del último año publicado de ese periódico, y en dicho proceso comprueba si el periódico de cada día de ese año existe o no. En caso afirmativo se crea un hipervínculo al fichero "frames.php" que mostrará la primera página del periódico.

A continuación mostramos el código en PHP de la función comprueba_prensa ($prensa, $año, $mes, $dia,) que realiza la comprobación explicada anteriormente:

```
function comprueba_prensa ($prensa, $ano, $mes, $dia)
{
        if (($dia>=1) and ($dia<=9)) $dia="0".$dia;
        if (($mes>=1) and ($mes<=9)) $mes="0".$mes;
        if ((int)$ano>=2004)
        {
        $file="./2004/".$prensa."/".$prensa.$ano.$mes.$dia."0.pdf";
        if ( file_exists ( $file ) ){
                $fecha=$dia.'/'.$mes.'/'.$ano;
                echo '<font face="Arial" size="2"><a
href="framesn.php?prensa='.$prensa.'&fecha='.$fecha.'">'.$dia.'</a></fo
nt>';}
        else echo '<font face="Arialsize="2">'.$dia.'</font>';
        }
        else
        {
        $ruta="./2003/".$prensa."/".$ano.$mes."/".$dia."/";
        if (@opendir($ruta))
        {

                $fecha=$dia.'/'.$mes.'/'.$ano;
                echo '<font face="Arial" size="2"><a
href="framesn.php?prensa='.$prensa.'&fecha='.$fecha.'">'.$dia.'</a></fo
nt>';
        }
        else { echo '<font face="Arial" size="2">'.$dia.'</font>';}
        }
}
```

En esta función se genera el enlace de cada día dependiendo de si el año es anterior a 2004 o es a partir de 2004, ya que se deben crear enlaces diferentes en cada caso. En ambos casos, si existe el periódico muestra el día con el hipervínculo asociado al mismo, y si no existe solamente muestra el día pero sin hipervínculo. Así el usuario sabrá dentro del calendario anual que periódicos están disponibles en ese año.

4.2 Servidor Web y lenguaje de programación.

El servidor Web utilizado para el Sitio Web de este proyecto es el Apache para el sistema operativo Linux. En el apéndice explicamos la instalación y configuración de este servidor Web.

El lenguaje utilizado para desarrollar el sitio Web de este proyecto ha sido el lenguaje PHP (**PHP**: **H**ypertext **P**reprocessor), un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. En el apéndice explicamos

más detalles sobre este lenguaje y su integración con el servidor Web Apache, así como las funciones PDF que hemos utilizado.

4.3 Interfaz Web

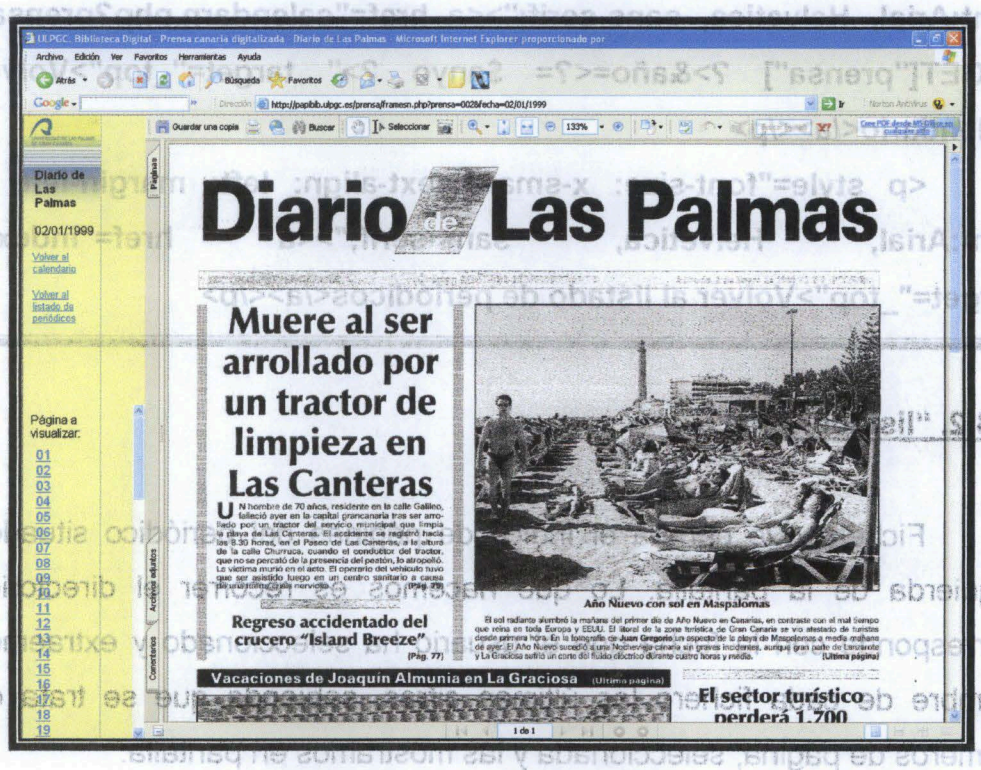
Nos vamos a centrar en la programación Web que hemos implementado para mostrar aquellos periódicos con fechas hasta el año 2003, almacenados en el repositorio en formato TIFF.

En el código en PHP mostrado anteriormente, vemos que cuando se trata de uno de estos periódicos el hipervínculo se crea de la siguiente manera:

```
<a href="frames.php?
prensa='.$prensa.'&fecha='.$fecha.'">'.$dia.'</a>
```

Cuando el usuario pulsa sobre el día deseado, se carga el fichero "frames.php"

Y se le pasa como variables el código del periódico y la fecha elegida. Se genera entonces la siguiente página Web:



Página Web generada en "frames.php"

Este fichero llama a su vez a otros 3 ficheros PHP y son cargados en 3 partes bien diferenciadas:

4.3.1. "datos.php":

Fichero que muestra en la parte superior izquierda de la ventana el nombre del periódico y la fecha seleccionada, así como hipervínculos para volver al calendario del mismo periódico o regresar a la página inicial para seleccionar otro periódico:

```
<h1 style="font-size: small; font-weight: bold; text-align: left; margin-left: 1em;"><? nombre_prensa(); ?></h1>
<p style="font-size: small; text-align: left; margin-left: 1em; font:Arial, Helvetica, sans-serif; "><?= $fecha ?></p>
<p style="font-size: x-small; text-align: left; margin-left: 1em; font:Arial, Helvetica, sans-serif;"><a href="calendarn.php?prensa=<?= $_GET["prensa"] ?>&año=<?= $anyo ?>" target="_top">Volver al calendario</a></p>
<p style="font-size: x-small; text-align: left; margin-left: 1em; font:Arial, Helvetica, sans-serif;"><a href="index.php" target="_top">Volver al listado de periódicos</a></p>
```

4.3.2. "listador.php":

Fichero que genera el índice de páginas del periódico situado a la izquierda de la pantalla. Lo que hacemos es recorrer el directorio que corresponde con la fecha que el usuario ha seleccionado y extraemos del nombre de cada fichero las últimas cifras, sabiendo que se trata de los números de página, seleccionada y las mostramos en pantalla.

Le asociamos a cada página un hipervínculo para que cuando el usuario pulse sobre un número de página, se cargue el fichero "e1.php" a la derecha de la pantalla para mostrar la imagen de la página seleccionada.

```
while ($file = readdir($dir))
{
 if ( $a>=100) $trozo = substr($file, -7,3); else $trozo = substr($file, -6,2);
        $trozo2 = substr($file, -3,3);
        if(($trozo2 == "TIF" OR $trozo2 == "tif") AND $file != "." AND $file
!= "..")
          {
                rl=substr($carpeta,1)."/".$file;
  echo"<span style=\"font:Arial, Helvetica, sans-serif; font-size: medium;
margin: 1em;\"><a
```

```
href='e1.php?imagen=".$url."'target=principal>$trozo</a></span>
<br/>";
          }
          $a=$a+1;
}
closedir($dir) ;
```

4.3.3. "pagina1.php":

Fichero que busca la primera página del periódico y la muestra por pantalla. Aquí utilizamos la programación en PHP necesaria para poder mostrar la imagen TIFF en formato PDF.

Para ello utilizamos una librería PHP que contiene una serie de funciones creadas para el manejo de los ficheros PDF. Con dichas funciones podemos crear los ficheros, insertar textos e imágenes en los mismos, guardarlos, etc. En el Apéndice explicamos el funcionamiento de estas funciones.

El proceso funciona de la siguiente manera:

Utilizamos la función "pdf_new" para crear un fichero PDF. Este fichero es creado en memoria y por lo tanto se trata de un fichero temporal que se´ra destruido cuando acabe el proceso.

Al crear un fichero PDF de esta forma, éste se encuentra inicialmente vacío de todo contenido, por lo que tenemos que usar las funciones "pdf_open_file" y "pdf_begin_page" para abrirlo y empezar una nueva página en blanco. En ese momento el fichero está preparado para insertarle la información que queramos. En nuestro caso lo que vamos a hacer es insertar la imagen TIFF correspondiente a la página del periódico que el usuario ha seleccionado.

Previamente tenemos que utilizar la función "pdf_load_image" para abrir y cargar en memoria la imagen TIFF, pasándole el formato de la imagen y el nombre del archivo. Después se utilizan las funciones "pdf_get_value" y "pdf_scale", que nos permite averiguar el alto y el ancho de la imagen TIFF original, para luego cambiar su tamaño y adecuarlo al tamaño de la página PDF.

Una vez la imagen TIFF se encuentra cargada y su tamaño es el correcto, la insertamos en el fichero PDF mediante la función "pdf_place_image" , y luego con la función "pdf_end_page" damos por terminada la página del fiero PDF y cerramos en fichero con la función "pdf_close". Así, el fichero PDF ya está preparado para mostrarlo por pantalla.

Como explicábamos antes, el fichero PDF que hemos creado se encuentra cargado en memoria. Para recogerlo de allí necesitamos la función "pdf_get_buffer", y finalmente con la siguiente secuencia de código mostramos el fichero PDF por pantalla:

```
header("Content-type: application/pdf");
header("Content-disposition: inline; filename=test.pdf");
header("Content-length: " . strlen($document));
echo $document;
```

4.3.4. "e1.php"

Este fichero es cargado cuando el usuario pulsa sobre alguno de los números de página del índice para mostrar la página del periódico que el usuario ha seleccionado. El proceso de este fichero es prácticamente el mismo que para el fichero "pagina1.php", que muestra la primera página del periódico. A diferencia de éste, a "e1.php" se le pasa la variable que contiene el número de la página que el usuario ha pulsado, pudiendo así poder identificar el fichero TIFF que debemos mostrar por pantalla, ya que como se ha comentado anteriormente, las últimas cifras del nombre de los ficheros TIFF identifican las páginas del periódico a las que pertenecen las imágenes que contienen. Por lo demás el proceso de creación del fichero PDF es el mismo.

4.3.5. El calendario.

Como ya hemos comentado, para facilitar al usuario la selección del periódico de una fecha determinada, se muestra un calendario anual. El usuario puede ver por defecto el último año publicado del periódico seleccionado en la página anterior, pero ofrecemos diferentes enlaces para seleccionar otros años. Concretamente se ofrece la posibilidad de cambiar de

década, cuyos enlaces se sitúan en la parte superior del calendario, y una vez seleccionado la misma, se da la posibilidad de seleccionar un año correspondiente a la década elegida.

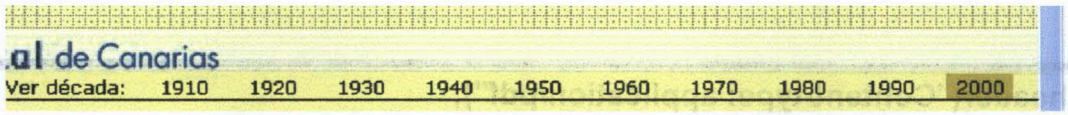


Imagen superior de la Web donde vemos el índice de décadas

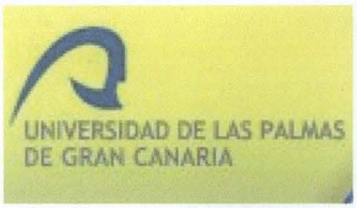
Dependiendo del periódico que el usuario selecciona, solo aparecen las décadas disponibles en el repositorio, así como los años de la década que el usuario seleccione. De esta forma se evita listar décadas o años que no se encuentran en el repositorio.

Imagen izquierda de la Web correspondiente al índice de años

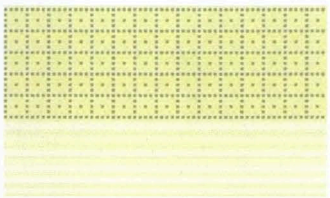
4.4. Imagen corporativa de la Biblioteca Digital.

El diseño del sitio Web de este proyecto debía estar acorde con el entorno de la Biblioteca Digital. Por tanto se han utilizado las hojas de estilo (Tipos y tamaños de fuentes colores, etc.) y las imágenes (logotipos, fondos, etc.) existentes dentro de la sección de la Biblioteca Digital donde el sitio Web se iba a integrar.

Logotipo de la sección "Memoria Digital de Canarias"



Logotipos corporativos de la Universidad de Las Palmas de G.C.



Logotipo de la B. Digital e imagen de fondo de la sección "MDC"

Imágenes informativas del sitio Web de la Biblioteca Digital

5. integración en la Biblioteca Digital.

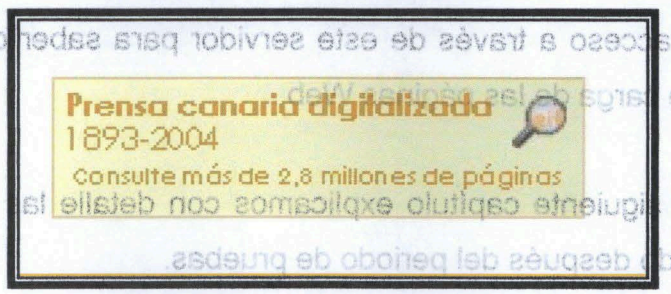
Después de terminar la interfaz Web de este proyecto y de realizar algunas pruebas de su funcionamiento, decidimos someter el proyecto a la prueba de fuego que supone la utilización diaria por parte de los usuarios finales.

5.1 Periodo de pruebas.

Para ofrecer al usuario de la BibliotecaDigital este nuevo servicio de consulta de la prensa canaria digitalizada, hemos añadido en la página principal una imagen creada al efecto, donde se le asocia un hipervínculo a la página de inicio del sitio Web de este proyecto. El código HTML para añadir la imagen con su hipervínculo es el siguiente:

```
<a href="../prensa"><div align="center">
<img src="digital/imagenes/ pcd4.jpg" border="0"> </div></a>
```

A continuación se muestra la imagen creada e integrada en la página Web principal de la Biblioteca Digital.



Se envió un comunicado de prensa para dar a conocer la puesta en marcha de este servicio en línea. LA dirección Web es:

http://bdigital.ulpgc.es/prensa

Durante el periodo de pruebas pudimos comprobar la fiabilidad del sitio Web creado. Las imágenes TIFF generadas "al vuelo" en PDF tuvieron un buen funcionamiento, excepto en aquellos equipos donde se había instalado algún visor de imágenes TIFF. Por tanto es condición indispensable para el buen funcionamiento del sitio Web que no exista ningún visor configurado para imágenes en formato TIFF en el ordenador.

También hicimos diferentes pruebas para comprobar el funcionamiento del sitio Web en diferentes sistemas operativos, así como en diferentes navegadores. Estas pruebas sirvieron para realizar ajustes en el código fuente de la aplicación Web que permitiera su buen funcionamiento en cualquier Sistema Operativo y en distintos navegadores Web.

Pero nuestra principal preocupación era la carga de las páginas Web en el navegador en cuanto a su velocidad. La rápida Intranet de la Universidad ha resultado fundamental para que dicha carga sea factible, sobre todo para los periódicos digitalizados en formato PDF.

Durante el periodo de pruebas se puso en marcha el servidor de autentificación "PAPI", con el que los miembros de la comunidad universitaria pueden acceder a la Intranet desde el exterior. Por tanto tuvimos que realizar pruebas de acceso a través de este servidor para saber que ocurría con la velocidad de carga de las páginas Web

En el siguiente capítulo explicamos con detalle las conclusiones que hemos sacado después del periodo de pruebas.

6. conclusiones finales.

La Biblioteca Universitaria ha tenido en los últimos años un objetivo claro, salvaguardar la memoria hemerográfica de nuestra región, y ofrecer al usuario la posibilidad de poder acceder a la información de la mejor manera posible, y para ello se ha servido de los avances tecnológicos en materia de almacenamiento, digitalización masiva, capacidad de almacenamiento y conectividad.

La creación de la Biblioteca Digital ha permitido cumplir con ese objetivo, y la puesta en marcha de la consulta en línea de la prensa canaria digitalizada es un paso más en este camino.

El trabajo realizado para la realización de este proyecto y el periodo de pruebas establecido, nos ha llevado a las siguientes conclusiones:

El formato TIFF de los ficheros de la prensa digitalizada con fechas anteriores a 2004, aunque tienen una calidad de compresión aceptable y el peso de los mismos es relativamente pequeño (una media de 500 Kilobytes), si bien es adecuado para realizar las consultas de manera local, no lo es para el acceso en línea al no ser un formato compatible con los navegadores de Internet, razón por la cual decidimos hacer la conversión "al vuelo" de los mismos al formato PDF

La elección del formato PDF para dicha conversión ha permitido que a los ojos del usuario no existan apenas diferencias entre los periódicos con formato TIFF y los ficheros PDF nativos a la hora de ser consultados y visualizados, siendo así transparente al usuario aunque el desarrollo del software para visualizar ambos formatos sea diferente.

Teniendo en cuenta que este proyecto es accesible solamente para la intranet de la universidad, la velocidad de carga de las páginas donde mostramos las páginas de prensa es muy buena para los ficheros TIFF convertidos a PDF, pero no es demasiado buena para los PDF nativos.

En mi opinión, la decisión de cambiar el formato de los periódicos al formato PDF para aquellos con fechas a partir del año 2004 es acertada, pero no así la decisión de almacenar un fichero por periódico, ya que eso exige la carga completa del fichero en el navegador Web, lo que supone una espera considerable para el usuario, ya que el peso de estos ficheros superan en algunos casos los 50 Megabytes. Sin embargo, la carga es efectiva gracias al ancho de banda disponible en la Intranet de la universidad.

El servicio de informática de la Universidad de Las Palmas de G. C. puso en marcha un servidor de autentificación, denominado "PAPI", que permite a cualquier miembro de la comunidad universitaria acceder a la Intranet de la universidad desde el exterior, mediante un nombre de usuario y una contraseña. Las pruebas realizadas sobre la consulta de la prensa digitalizada a través de este servidor de autentificación concluyeron que los anchos de banda de las conexiones de Internet aun no son lo suficientemente buenos para que las consultas sean efectivas.

Por ejemplo, desde una conexión ADSL de 1 Megabyte de velocidad de bajada y 300 Kilobytes de velocidad de subida, la consulta de prensa con fechas hasta el año 2003 (formato TIFF) es relativamente buena, ya que la velocidad de carga del calendario anual tarda pocos segundos, al igual que la visualización del periódico, debido a que sólo se carga una página del periódico en el navegador Web.

Todo lo contrario ocurre cuando con el mismo ancho de banda anterior, intentamos consultar un periódico a partir del año 2004 con formato

PDF. Al ser un único fichero, éste se debe cargar para poder visualizar el periódico, y el tamaño del fichero comentado anteriormente hace casi imposible su visualización con dicho ancho de banda, llegando en ocasiones a colgarse el navegador Web.

Por tanto, hasta que el ancho de banda desde el exterior de la Intranet no mejore, la consulta de los periódicos en formato PDF no es viable, aconsejando al usuario que sólo realice consultas desde el exterior a periódicos con fechas anteriores a 2004.

Comentar que también se presentaron algunos problemas con los usuarios que tenían instalado el Service Pack 2 de Windows XP para acceder a la Intranet desde el exterior.

6.1 Principales aportaciones.

Las principales aportaciones de este proyecto has sido:

La profundización en el creciente ámbito de las Bibliotecas Digitales en cuanto a su funcionamiento y a las tecnologías utilizadas para su desarrollo.

La investigación acerca de las digitalizaciones masivas y de los formatos utilizados para las mismas, analizando las decisiones adoptadas al respecto.

El estudio de los documentos digitalizados para analizar la viabilidad de su acceso en línea.

La influencia del ancho de banda en la carga de páginas Web en función del peso de los ficheros digitalizados.

El manejo y configuración de software libre para el desarrollo de aplicaciones Web.

Al respecto del último punto, quisiera añadir que ésta ha sido mi primera experiencia con el software libre, y la plataforma Web "LAMP", conjunto de aplicaciones de este tipo de software, me ha permitido conocer la potencialidad que puede existir en el desarrollo de software, especialmente en aplicaciones Web, mas allá de del entorno "cerrado" de empresas como Microsoft.

6.2 Futuras líneas de desarrollo.

La consulta de la Prensa Canaria Digitalizada se puso en marcha para dar a conocer a los usuarios potenciales la existencia de este corpus, y las características del mismo nos ha permitido realizar búsquedas de periódicos por fechas sin demasiada dificultad y sin la necesidad de utilizar una base de datos como índice de búsqueda.

La digitalización propia de la Biblioteca Universitaria ha servido para añadir periódicos al corpus de la prensa, además en un formato diferente, sin tener que cambiar la programación del sitio Web, donde el usuario solo necesita descargar el plug-in para visualizarlos.

También se podría utilizar la misma plataforma para consultar en el futuro otras publicaciones que pudieran llegar a la Biblioteca Digital, como revistas especializadas, artículos varios, etc., que se podrían dividir en diferentes categorías y serían de gran ayuda para los estudiantes e investigadores.

Pero, centrándonos en el corpus ya existente, estamos estudiando los ficheros OCR que la empresa especializada creó durante el proceso de

digitalización. Al igual que para los ficheros de los periódicos, aquí también hay que distinguir dos formatos:

- Formato MDB (Microsoft Access) y ficheros ZIP asociados a la prensa digitalizada en formato TIFF
- Formato PDX asociado a la prensa digitalizada en formato PDF

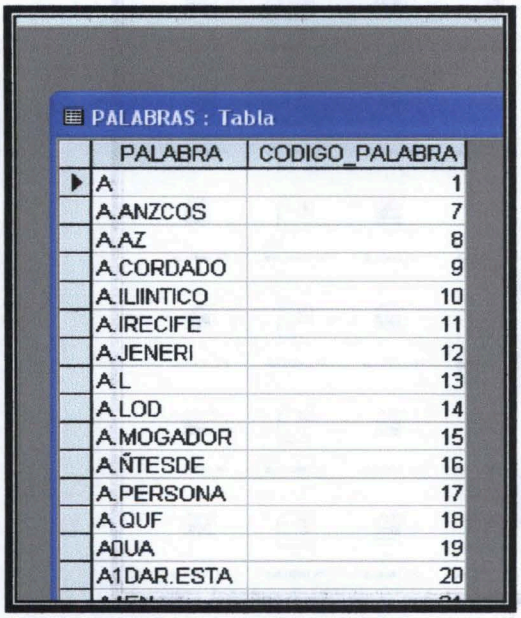
En el primer caso, comprobamos el contenido de los CD's que almacenan los índices MDB y los archivos en formato ZIP.

Contenido de un CD con índices MDB y ficheros ZIP

Como vemos en la imagen, para cada fichero MDB existe un fichero ZIP con el mismo nombre. Dicho nombre identifica a un mes de un año determinado (Por ejemplo, "198312" se corresponde con el mes de diciembre de 1893). Vamos a ver el contenido de un fichero MDB.

Fichero MDB

Podemos ver que cada fichero MDB contiene tres tablas:



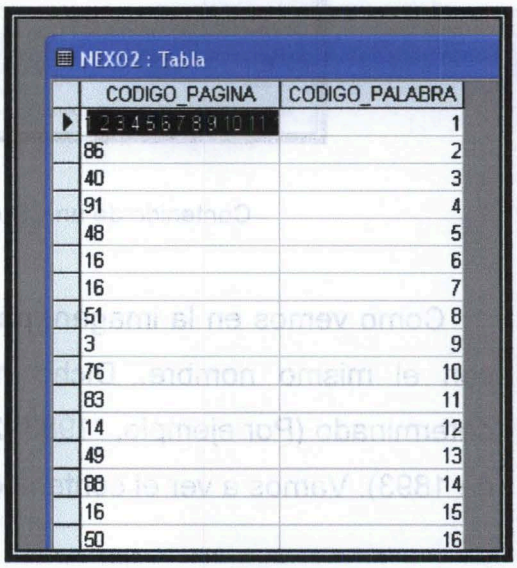
PALABRAS : Tabla

PALABRA	CODIGO_PALABRA
A	1
A.ANZCOS	7
A.AZ	8
A.CORDADO	9
A.ILIINTICO	10
A.IRECIFE	11
A.JENERI	12
A.L	13
A.LOD	14
A.MOGADOR	15
A.ÑTESDE	16
A.PERSONA	17
A.QUF	18
AQUA	19
A1DAR.ESTA	20

Tabla PALABRAS

La tabla "PALABRAS" tiene los campos "PALABRA" que contienen palabras en orden alfabético, y "CODIGO_PALABRA" que asocia un número a cada palabra.

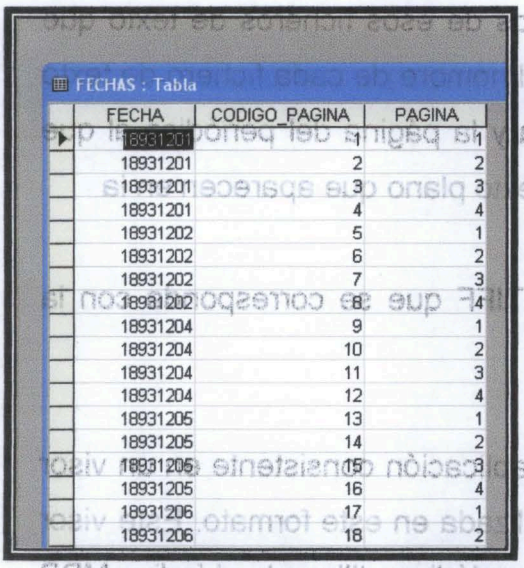
La tabla "NEXO2" tiene el mismo campo "CODIGO_PALABRA" y un campo "CODIGO_PAGINA" que asocia los números de palabra con un código de página.



NEXO2 : Tabla

CODIGO_PAGINA	CODIGO_PALABRA
1 2 3 4 5 6 7 8 9 10 11	1
86	2
40	3
91	4
48	5
16	6
16	7
51	8
3	9
76	10
83	11
14	12
49	13
88	14
16	15
50	16

Tabla NEXO2



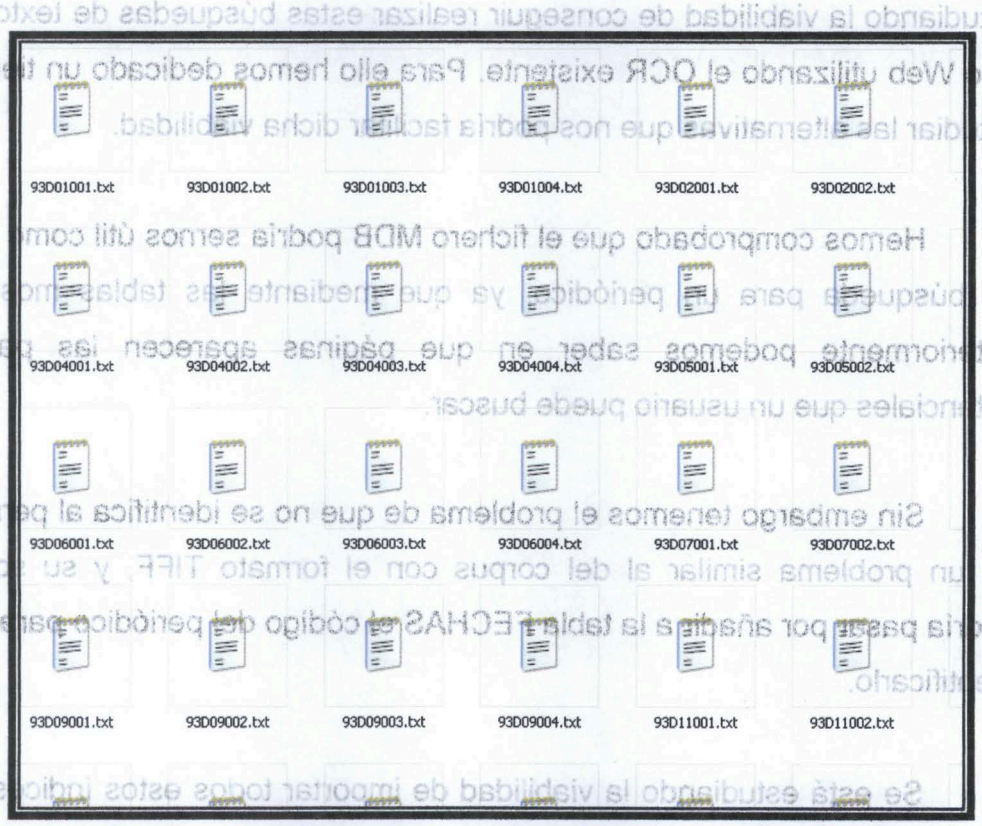
FECHAS : Tabla

FECHA	CODIGO_PAGINA	PAGINA
18931201	1	1
18931201	2	2
18931201	3	3
18931201	4	4
18931202	5	1
18931202	6	2
18931202	7	3
18931202	8	4
18931204	9	1
18931204	10	2
18931204	11	3
18931204	12	4
18931205	13	1
18931205	14	2
18931205	15	3
18931205	16	4
18931206	17	1
18931206	18	2

Tabla FECHAS

La tabla "FECHAS" utiliza el campo "CODIGO_PAGINA" para asociar cada código a una página y una fecha del periódico al que pertenece el fichero.

Un fichero ZIP contiene tantos archivos de texto cómo páginas de periódicos que existen en el mes y año identificado en el nombre del fichero ZIP.



Contenido de un fichero ZIP

En esta imagen podemos ver algunos de esos ficheros de texto que han sido descomprimidos del fichero ZIP. El nombre de cada fichero de texto nos sirve para identificar el día de la fecha y la página del periódico al que pertenece. Cada fichero de texto contiene texto plano que aparecen en la

página y que podemos ver en la imagen TIFF que se corresponda con la misma página.

La empresa especializada creó una aplicación consistente en un visor de las imágenes TIFF para la prensa digitalizada en este formato. Este visor permite hacer consultas de texto sobre un periódico utilizando el índice MDB asociado al mismo para las búsquedas y los archivos de texto plano para ofrecer los resultados de las mismas. El visor es utilizado por los usuarios en el servicio de Hemeroteca de la Biblioteca Universitaria de manera local durante el horario de atención al público. Con este panorama se está estudiando la viabilidad de conseguir realizar estas búsquedas de texto en el sitio Web utilizando el OCR existente. Para ello hemos dedicado un tiempo a estudiar las alternativas que nos podría facilitar dicha viabilidad.

Hemos comprobado que el fichero MDB podría sernos útil como índice de búsqueda para un periódico, ya que mediante las tablas mostradas anteriormente podemos saber en que páginas aparecen las palabras potenciales que un usuario puede buscar.

Sin embargo tenemos el problema de que no se identifica al periódico. Es un problema similar al del corpus con el formato TIFF, y su solución podría pasar por añadir a la tabla FECHAS el código del periódico para poder identificarlo.

Se está estudiando la viabilidad de importar todos estos índices en el motor de bases de datos utilizado en la Biblioteca Digital, "MYSQL", teniendo

en cuenta dos aspectos fundamentales a la hora de realizar búsquedas de texto:

- Cómo identificar el periódico dentro de la base de datos para poder realizar las búsquedas de manera efectiva dentro del índice.

- Cómo importar a MYSQL tal cantidad de bases de datos en una sola, o en su caso crear varias bases de datos barajándose diferentes posibilidades, entre las que se encuentran crear una base de datos por periódico o crear una base de datos por fechas (por ejemplo por décadas), incluso combinar ambas categorías.

Pero existe otro obstáculo mucho mayor, un fichero índice MDB está asociado a un mes de un periódico determinado. Eso significa que tenemos una gran cantidad de bases de datos.

Por ejemplo, el periódico "DIARIO DE LAS PALMAS" data de finales de 1893 y se dejó de publicar en 1999 al fusionarse con "LA PROVINCIA". Esto significa que con más de 100 años de existencia, tenemos más de 1200 ficheros MDB sólo para este periódico.

Además, hemos comprobado que los códigos de palabra de diferentes ficheros no coinciden, ni aún tratándose de dos ficheros del mismo periódico, por lo que intentar fusionar las bases de datos en una sola provocaría la existencia de varios códigos de palabra para la misma palabra, siendo complicada la búsqueda de esa palabra en la base de datos ya que no tendría un código de palabra único y eso obligaría a tener que recorrer siempre toda la base de datos para encontrar los diferentes códigos de páginas asociados a los diferentes códigos de palabras. Por tanto las posibles búsquedas de texto se podrían eternizar para el usuario. Aún así, en

el momento de escribir esta memoria del proyecto seguimos tratando de encontrar una solución para poder utilizar el OCR creado.

Otra alternativa sería crear nuestro propio OCR, lo que podría tardar bastante tiempo al tener que volver a procesar las cerca de 3 millones de páginas que tenemos en el corpus en formato TIFF. Aunque quizás

podríamos utilizar los ficheros de texto ya creados para crear un índice nuevo.

Para el segundo caso de los índices PDX asociados a los ficheros PDF se utilizan para realizar búsquedas de texto en el fichero PDF. Sin embargo la búsqueda se realiza en modo local. Se está estudiando la posibilidad de que estas búsquedas también funcionen para direcciones Web y poder así hacer consultas de texto en los periódicos que tienen el formato PDF nativo.

Aquí tampoco descartamos crear índices propios para poder poner en línea la búsqueda de texto en la prensa digitalizada en formato PDF.

6.3 Bibliografía

Configuración de sistemas Linux. Daniel L. Morrill, Ed. Anaya Multimedia, 2002

Sitios Web bajo Linux: Usuarios Expertos. Hector Facundo Arena, MP Ediciones, 2001

La biblia de servidor apache 2. kabir, mohammed j.
Anaya multimedia-anaya interactiva, 2002

Apache Práctico, Ken Coar; Rich Bowen, Anaya Multimedia-Anaya interactiva, 2004

Dreamweaver MX 2004: desarrollo de páginas web dinámicas con PHP y MySQL, Pérez, César, Ra-Ma, 2004

Creación de aplicaciones web con PHP 4. Ratschiller,
Tobias, Pearson Educación, S.A., 2000

Direcciones Web

www.**linux**.org
es.**gnu**.org
www.europe.**redhat**.com
www.**apache**.org
www.**php**.net
www.**pdflib**.com

7. Apéndice.

En este capítulo vamos a explicar con algo más detalle como se encuentra configurado el almacenamiento secundario externo necesario para albergar el corpus de la prensa, así como la instalación y la configuración de las aplicaciones software que forman parte de la plataforma de software libre LAMP, utilizadas para la Biblioteca Digital y para realizar este proyecto.

7.1 Sistemas RAID

RAID (Redundant Array Of Independent/Inexpensive Disks) es un término inglés que hace referencia a un conjunto de discos redundantes independientes/baratos. Este tipo de dispositivos se utilizan para aumentar la integridad de los datos en los discos, mejorar la tolerancia a los fallos y errores y mejorar el rendimiento. En general permiten proveer discos virtuales de un tamaño mucho mayor al de los discos comúnmente disponibles. Inicialmente un sistema RAID era un conjunto de discos redundantes económicos.

Historia

El sistema RAID fue propuesto por primera vez en 1988 por David A. Patterson, Garth A. Gibson y Randy H. Katz en la publicación "Un Caso para Conjuntos de Discos Redundantes Económicos (RAID)". Este fue publicado en la Conferencia SIGMOD de 1988: pág. 109 – 116. El término "RAID" comenzó en esta publicación.

Fue un trabajo particularmente excepcional en el que los conceptos son "obvios". Esta publicación generó la industria de los conjuntos de disco.

Oficialmente los sistemas RAID se implementan en 7 configuraciones o niveles: RAID 0 a RAID 6. También existen combinaciones de niveles de RAID, las combinaciones más comunes son RAID 10 y RAID 0+1. Los sistemas RAID son comúnmente implementados con discos de la misma capacidad para todo el conjunto.

A nivel práctico y comercial, sólo los RAID impares, junto a las combinaciones de estos, se han impuesto en el mercado: RAID 1, 3, 5, 7, 1+0, y 0+1. Destacan por su aceptación sobre los demás el RAID 1, 5, 10, y 0+1.

Hardware vs. Software

Cualquiera de los niveles de RAID que aparecen listados abajo pueden ser implementados en hardware o software.

Con la implementación por software, el sistema operativo maneja los discos del conjunto a través de una controladora de discos normal (IDE, Serial ATA, SCSI o Canal de Fibra). Esta opción puede ser lenta, pero no requiere de la compra de hardware adicional.

Una implementación de RAID basada en hardware requiere (por lo menos) una tarjeta controladora RAID. Esta controladora maneja la administración de los discos, y efectúa los cálculos de paridad (necesarios para RAID 4,5). Esta opción ofrece un mejor rendimiento y hace que el soporte por parte del sistema operativo sea más sencillo.

Las implementaciones basadas en hardware típicamente soportan intercambio en caliente, permitiendo que los discos que fallen sean reemplazados sin necesidad de detener el sistema.

Los Niveles de RAID son los siguientes:

RAID 0: Conjunto de discos divididos sin tolerancia a fallos (No Redundante)

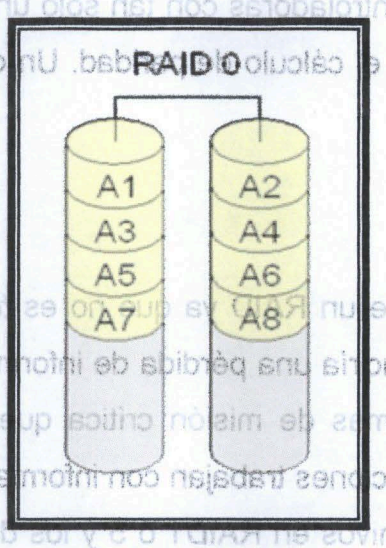


Diagrama de una configuración RAID 0

El RAID de nivel 0 no tiene mínimo de discos para ser empleado. Se puede crear un Raid 0 con un solo disco, pero la ganancia en velocidad es inapreciable. Si conectas 2 discos duros diferentes, por ejemplo, de 100GB y 120GB respectivamente, obtienes una capacidad de 200GB (es decir, dos veces el tamaño del disco más pequeño), "perdiendo" 20GB por la diferencia de capacidades de disco. Por eso se recomienda usar 2 discos con las mismas características. <--- con algunas controladoras al hacer RAID 0 por hardware se obtiene la suma de todos los discos que se metan en RAID, en el ejemplo se obtendrían 220 GB y no se perdería espacio alguno.

Características y Ventajas

El RAID 0 implementa un conjunto de discos divididos, la información es separada en bloques y cada bloque es grabado en una unidad de disco diferente. El rendimiento de Entrada/Salida se ve muy beneficiado por la dispersión de la carga de Entrada/Salida a través de muchos canales y discos.

El mejor rendimiento se alcanza cuando los datos son divididos a través de múltiples controladoras con tan solo un disco por controladora. No existe sobrecarga por el cálculo de paridad. Un diseño muy simple. Fácil de implementar.

Desventajas

No es realmente un RAID ya que no es *tolerante a fallos*. El fallo de una sola unidad produciría una pérdida de información en el conjunto. No se debe utilizar en sistemas de misión crítica que impliquen modificación de datos. (Algunas aplicaciones trabajan con información de control almacenada en un sistema de archivos en RAID1 ó 5 y los datos multimedia en RAID 0, los cuales son respaldados a cinta o a medios ópticos.)

Aplicaciones Recomendadas

- Edición y producción de Vídeo
- Edición de imágenes
- Aplicaciones de Preimprenta
- Cualquier aplicación que requiera gran ancho de banda.

RAID 1: Mirroring y Duplexing (Espejo)

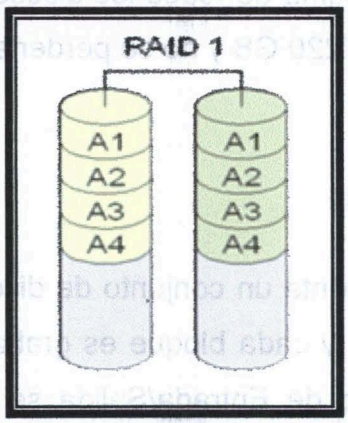


Diagrama de una configuración RAID 1

Para el mejor rendimiento, la controladora debe ser capaz de realizar dos lecturas concurrentes separadas por cada par duplicado, y dos escrituras duplicadas por cada par duplicado.

El nivel de RAID 1 requiere al menos dos unidades de disco para ser implementado

Características

Son posibles una escritura o dos lecturas por par. El doble de la tasa de transacciones de lectura de un disco simple, la misma tasa de escritura que un disco simple. Redundancia del 100% en los datos significa que no es necesaria la reconstrucción en el caso de fallo de un disco, solo una copia para el reemplazo de disco.

La tasa de transferencia por bloque es igual a la de un disco simple. Bajo ciertas circunstancias, el RAID 1 puede sostener fallas en múltiples hunch.

Es el diseño de un subsistema de almacenamiento en RAID más sencillo.

Ventajas

Debido a que un disco es espejeado en par y contiene toda la información, puede ser potencialmente utilizado sin software o hardware para RAID

Desventajas

El más alto volumen de carga de todos los tipos de RAID, (100%) ineficiente.

Aplicaciones Recomendadas

- Contabilidad
- Nómina
- Finanzas
- Cualquier aplicación que requiera de alta disponibilidad
- Servidores

RAID 2: Código de Corrección de Error

El esquema de redundancia en el RAID de nivel 2 es un código de Hamming, donde la unidad de separación es un bit simple. Dividir al nivel de bit tiene la implicación de que en un conjunto de discos con N discos de datos, la unidad más pequeña de datos de transferencia para una lectura es un conjunto de N bloques.

El RAID de nivel 2 funciona a bajo nivel y su implementación no es usada actualmente.

RAID 3: Paridad de intervalo de bit (Paridad de Richard M. Price)

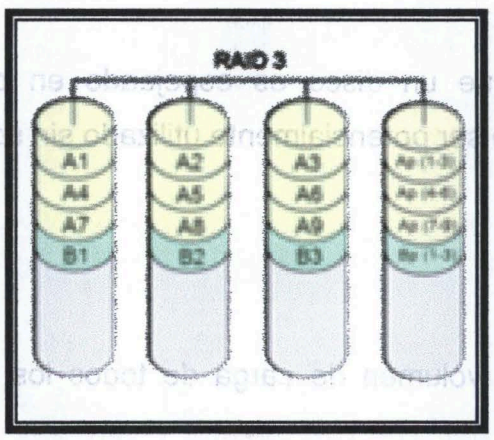


Diagrama de una configuración RAID 3

El RAID de nivel 3 tiene un disco de comprobación y solamente procesa una E/S a la vez.

El RAID de nivel 3 es implementado en contadas ocasiones. por su función a bajo nivel.

RAID 4: Unidad de paridad dedicada (Paridad de intervalo de bloque)

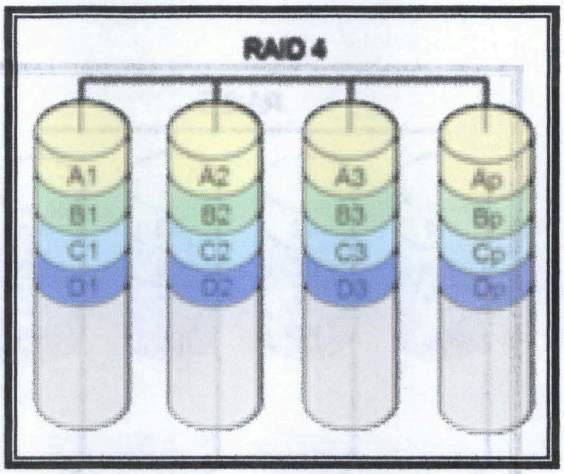


Diagrama de una configuración RAID 4

Características

Los discos son divididos, como en RAID 0. La paridad de información para la división es calculada, y almacenada en un disco de paridad. Si uno de los discos falla, la información es reconstruida en un disco de repuesto utilizando la información de paridad. Si el disco de paridad falla, la paridad de la información es recalculada en un disco de repuesto. La ventaja con el RAID 3 está en que se puede acceder a los discos de forma individual.

Desventajas

El disco de paridad puede ser un "cuello de botella"(bottleneck) durante las operaciones de escritura

RAID 5: Discos de datos independientes con bloques de paridad distribuidos (Bloques de Intervalo de Paridad Distribuida)

Este es el sistema RAID que hemos configurado para el servidor de la Biblioteca Digital, tras adquirir el Sistema de Almacenamiento Secundario POWERVAULT, con una capacidad de 4,2 Terabytes.

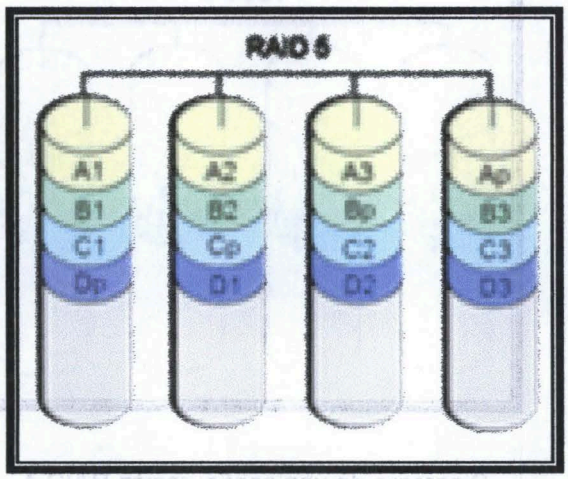


Diagrama de una configuración RAID 5

Cada vez que un *bloque* de datos (algunas veces llamado *pedazo*) es escrito en un disco dentro de un conjunto, un bloque de paridad es generado dentro de la misma división. (Un bloque o pedazo esta compuesto de muchos sectores consecutivos en un disco, algunas veces tanto como 256 sectores. Una serie de pedazos [un pedazo de cada disco dentro de un conjunto] es llamada colectivamente una *división*.)

Si otro bloque, o alguna porción del bloque es escrita en la misma división, el bloque de paridad (o una parte del bloque de paridad) es recalculada y vuelta a escribir. El disco utilizado por el bloque de paridad es escalonado desde una división hasta la siguiente, de ahí el término *bloques de paridad distribuidos*.

Es interesante que los bloques de paridad no sean leídos en las lecturas de datos, ya que esto sería una sobrecarga innecesaria y podría disminuir el rendimiento.

Los bloques de paridad son leídos, sin embargo, cuando la lectura de un sector de datos resulta en un error CRC. En este caso, el sector en la misma posición relativa en cada uno de los bloques de datos restantes en la división y dentro del bloque de paridad en la división son utilizados para reconstruir el sector erróneo. El error CRC se encuentra oculto para la computadora.

De cualquier manera, si un disco falla en el conjunto, los bloques de paridad en los discos sobrevivientes son combinados matemáticamente con los bloques de datos de los discos sobrevivientes para reconstruir los datos de la unidad que ha fallado *al vuelo*. Esto es algunas veces llamado Modo Interno de Recuperación de Datos.

La computadora no se entera de que el disco ha fallado. La acción de leer y escribir al conjunto de discos continúa normalmente, aunque con alguna degeneración de rendimiento. En RAID 5, en los conjuntos que solo tienen un bloque de paridad por división, la falla de una segunda unidad de disco resulta en la pérdida total de la información.

El RAID de nivel 5 requiere de al menos 3 unidades de disco para ser implementado. El número máximo de discos teóricamente es ilimitado, pero en la práctica es común mantener un máximo de 14 unidades de disco o menos para implementaciones de RAID 5 que tienen un solo bloque de paridad por división.

La razón de esta restricción es que existe una gran concordancia en cuanto a que una unidad de disco fallará en el conjunto cuando exista un gran número de unidades de disco. (El valor del Tiempo Estimado Entre

fallas MTBF para una unidad de disco dentro de un conjunto se vuelve más pequeño.)

En implementaciones con más de 14 unidades de disco, el RAID 5 con paridad dual (también conocido como RAID 6) es algunas veces utilizado ya que puede sobrevivir a la falla de dos discos.

Características y Ventajas

Mayor tasa de transacciones de lectura. De media a pobre tasa de transacciones de escritura, especialmente cuando el CPU realiza chequeos de paridad por software. Bajo coeficiente de discos ECC (Paridad) para los discos de datos significa alta eficiencia. Buena tasa de transferencia agregada.

Desventajas

El fallo de unidades de disco tiene un impacto medio en el caudal de salida. Diseño de controladoras más complejo. Dificultad para reconstruir en el caso de fallo de una unidad de disco (comparada con el RAID de nivel 1). En bloques de datos individuales la tasa de transferencia es la misma que en un disco individual.

Alta sobrecarga para escrituras pequeñas. Para cambiar 1 byte en un archivo, la división completa debe ser leída, el byte modificado, la información de paridad recalculada, y la división entera vuelta a escribir. Sin embargo, el hecho de que los sistemas de archivos tienden a dirigirse a los discos naturalmente en clusters oculta parcialmente este efecto.

Aplicaciones Recomendadas

- Servidores de Archivos y de Aplicaciones
- Servidores de Bases de Datos
- Servidores de web, e-mail y de noticias
- Servidores de Intranet
- Es el nivel de RAID más versátil
- Se puede configurar RAID 5 con disco de respaldo lo que permite obtener dos niveles de contingencia

RAID 6: Discos de Datos Independientes con Doble Paridad

Bloques de datos enteros son grabados en el disco; la paridad es generada y escrita a las dos líneas, en dos unidades separadas.

El RAID de nivel 6 requiere un mínimo de tres unidades, pero cuatro son requeridas para exceder la eficiencia en espacio de RAID 1.

Características

El conjunto de mayor redundancia en paridad, muy ineficiente con pocos discos, pero mucho más tolerante a fallas. Las unidades pueden ser organizadas en matrices ortogonales, donde las filas de discos forman grupos de paridad, similar al RAID 5, mientras las columnas también mantienen una paridad consistente entre cada una de ellas.

Si un solo disco falla, ya sea su fila o columna de paridad puede ser utilizada para reconstruirlo, Varias unidades dentro del conjunto pueden fallar antes de que este se vuelva corrupto. Cualquier grupo de discos no coincidentes puede fallar antes de que el conjunto se corrompa.

Recomendado para: aplicaciones de imágenes y fileserver en general

RAID 10: Una línea de Espejos

Se crean múltiples espejos de RAID 1, y una línea de RAID 0 es creada sobre estas. Este no es uno de los seis niveles originales, sino la combinación de RAID 1 y 0, algunas veces también llamada RAID 1+0.

Ventajas

Potencialmente puede manejar múltiples fallas de discos simultáneas, mientras uno de los discos de cada par espejeado continúe trabajando.

Las mismas ventajas y desventajas del RAID 1...

RAID 0 + 1: Un espejo de líneas

Dos líneas de RAID 0 son creadas, y un espejo en RAID 1 es creado sobre estas. Este tampoco es uno de los 6 niveles originales de RAID.

Desventajas

No es tan robusto como el RAID 1+0. No puede tolerar dos fallos simultáneos de discos, si no son de la misma línea ETC.

7.2 LAMP, la plataforma Web libre.

A finales del año 2000, los miembros del equipo de MySQL David Axmark y Monty Widenius visitaron al editor de O'Reilly Dale Dougherty y le hablaron de un nuevo término: LAMP. Al parecer era ya muy popular en Alemania, donde se empleaba para definir el trabajo conjunto con Linux, Apache, MySQL y uno de los siguientes lenguajes: Perl, Python o PHP. El término LAMP gustó tanto a Dougherty que empezó a promocionarlo desde la posición de extraordinaria influencia de su editorial en el mundo del software libre.

Es frecuente que se identifique a primera vista el mundo del software libre con Linux. Eso provoca que muchas veces se ignoren las herramientas que permiten a Linux convertirse en una gran herramienta de desarrollo de software, especialmente de aplicaciones web. Existen varios casos en los que un producto pasa de ser una curiosidad a una solución adecuada para la empresa, como ya ha sucedido con Sendmail o Kerberos. Esto es lo que ha sucedido con la solución para servicios web llamada LAMP.

LAMP está considerada como una de las mejores herramientas disponibles para que cualquier organización o individuo pueda emplear un servidor web versátil y potente. Aunque creados por separado, cada una de las tecnologías que lo forman disponen de una serie de características comunes. Especialmente interesante es el hecho que estos cuatro productos pueden funcionar en una amplia gama de hardware, con requerimientos relativamente pequeños sin perder estabilidad. Esto ha convertido a LAMP en la alternativa más adecuada para pequeñas y medianas empresas.

Existen, no obstante, multitud de variaciones de código libre. La L de Linux puede ser sustituida por FreeBSD, NetBSD u OpenBSD. En lugar de la M de MySQL también podemos encontrar PostgreSQL. La P sirve para PHP, Perl, Python, y Ruby. No obstante, las encuestas de Netcraft muestran que el LAMP que enseñamos en Ciberaula es la plataforma para crear páginas web más popular.

Algunas de las ventajas que se obtienen de utilizar LAMP son:

Soporte a gran cantidad de arquitecturas, como son Intel y compatibles, SPARC, Mips y PPC (Macintosh).

Código relativamente sencillo y con pocos cambios de una plataforma a otra.

Parches generados en poco tiempo después de encontrarse un agujero de seguridad.

Actualizaciones del software vía Internet.

Posibilidad de incrementar los servicios y funciones desde el código fuente

Sin embargo, tenemos también una serie de desventajas que deben considerarse:

Es muy distinto de Windows, lo que dificulta el trabajo a quienes estén acostumbrados a él.

Las actualizaciones requieren en ocasiones tener conocimientos profundos del sistema.

Configurar algunos servicios de red requiere de más tiempo que en Windows.

Mayor coste del personal.

Software libre

Todos los elementos que forman LAMP son software libre, de modo que disfrutan de las siguientes ventajas propias del mismo:

Libertad de copia y distribución.

Se puede conseguir gratuitamente en Internet. Hay muchísimas fuentes donde conseguir cualquiera de las distribuciones. Si no tienes una conexión rápida, también regalan Linux en los CD-ROM de muchas revistas especializadas.

Libertad de modificación.

Junto a los programas ejecutables, se puede obtener su código fuente. Esto, si se tienen los conocimientos necesarios, permite verificar la seguridad y eficiencia de los mismos, además de modificar y/o añadir las características y comportamientos que deseemos.

Linux

Esta basado en los estándares Unix, y surgió a principios de los 90, a partir de las inquietudes de Linus Torvalds por mejorar y ampliar Minix (otra implementación gratuita de Unix desarrollada por Andy Tanenbaum, dirigida al ámbito educativo). Desde entonces, ha ido incrementándose de forma espectacular el numero de desarrolladores desinteresados que se han implicado en su desarrollo a lo largo y ancho del mundo.

Lo que es propiamente Linux es el núcleo del sistema operativo, que ha ido implementando soporte para una gran parte del hardware actual (USB, cámaras digitales, escáneres, impresoras, grabadoras, redes, etc...). Dicho núcleo viene arropado por librerías y utilidades distribuidas bajo la licencia libre GPL o similares (de aquí la denominación GNU/Linux).

Su excelente relación calidad-precio le ha granjeado la admiración e incondicional apoyo de muchísimos usuarios alrededor del mundo. Su adopción en el ámbito de los servidores web ha sido espectacular. Estadísticas recientes demuestran que su empuje es cada vez mayor en este campo y todos los relacionados con Internet (como, por ejemplo, los servidores de espacio web e ISP). Por ejemplo, se usa en Google y Amazon.

Linux, entre muchas otras, es multitarea, multiusuario, multiplataforma, multiprocesador, tiene protección de la memoria entre procesos, soporta muchísimos tipos de sistemas de archivos, dispone de una amplia variedad de protocolos de red soportados en el núcleo y, finalmente, permite compartir

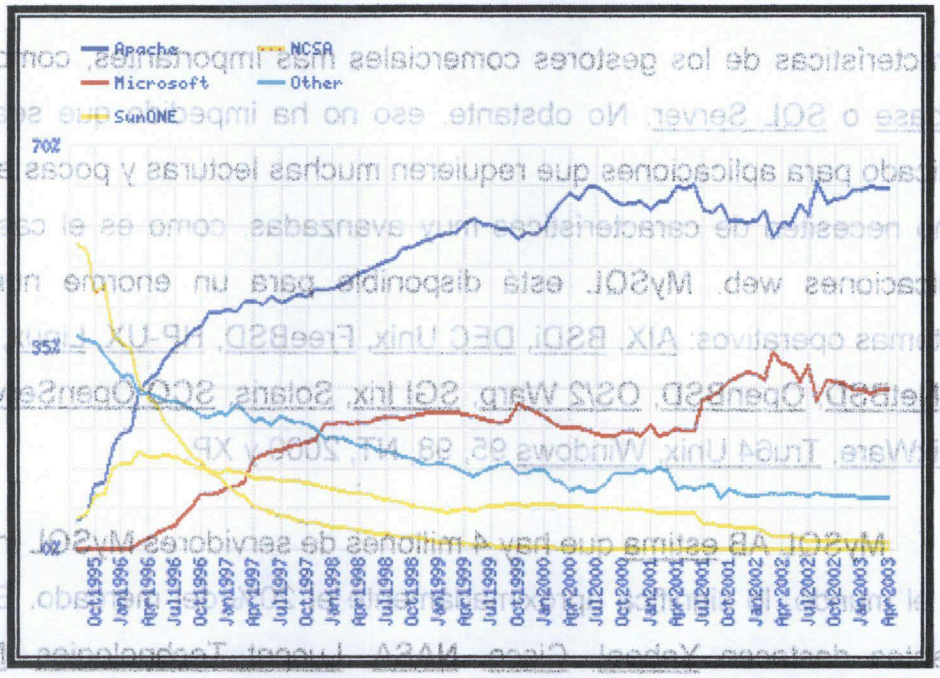
por red ficheros e impresoras, incluso con otros sistemas operativos. La potencia, estabilidad, gratuidad, modificabilidad y portabilidad de Linux lo hacen el sistema operativo perfecto, y ya tiene una posición líder en el ámbito de Internet, siendo cuestión de tiempo que la tenga en el resto de ámbitos informaticos.

Apache

Apache es el **servidor web** por excelencia, con algo mas de un 60% de los servidores de internet confiando en él. Entre sus características mas sobresalientes están:

- **Fiabilidad**: Alrededor del 90% de los servidores con mas alta disponibilidad funcionan con Apache.
- **Gratuidad**: Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License, que permite la modificacion del codigo.
- **Extensibilidad**: se pueden añadir módulos para ampliar las ya de por si amplias capacidades de Apache. Hay una amplia variedad de módulos, que permiten desde generar contenido dinámico (con PHP, Java, Perl, Pyton,...), monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, hasta crear servidores virtuales por IP o por nombre (varias direcciones web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos módulos incluso pueden ser creados por cualquier persona con conocimientos de programación

Este potente y famoso servidor se basa en el pionero NCSA server, y surgió a partir de diferentes ampliaciones y parches para el mismo (de ahi su nombre, derivación de 'A patchy server'), cuyo desarrollo se estancó a mediados de 1994. Un grupo de administradores web pusieron en marcha una lista de correo y fundaron el Apache Group. Al año, Apache era el numero 1 en la lista de Netcraft.



Gráfica de uso de diferentes servidores Web

MySQL

La administración y gestión de la información es uno de los puntos clave del éxito en cualquier entidad empresarial. La informática aporta la tecnología que permite satisfacer la necesidad de control de esta información, pero las empresas no se conforman trabajando con aplicaciones o programas que amontonen la información de forma caótica. Los datos deben organizarse de acuerdo a un proceso previo que comprende el análisis y diseño del modelo de datos, así como la elección y posterior configuración del sistema que soportará nuestra base de datos.

Existen diferentes arquitecturas para los sistemas de gestión de bases de datos, pero la más extendida, y la que más éxito ha tenido, es la arquitectura relacional. MySQL es un servidor de bases de datos relacionales muy rápido y robusto. Es software libre, publicado bajo la licencia GPL (GNU Public License) y mantenido por la compañía sueca MySQL AB. Este gestor se creó con la rapidez en mente, de modo que no tiene muchas de las

características de los gestores comerciales más importantes, como Oracle, Sybase o SQL Server. No obstante, eso no ha impedido que sea el más indicado para aplicaciones que requieren muchas lecturas y pocas escrituras y no necesiten de características muy avanzadas, como es el caso de las aplicaciones web. MySQL está disponible para un enorme número de sistemas operativos: AIX, BSDi, DEC Unix, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI Irix, Solaris, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows 95, 98, NT, 2000 y XP.

MySQL AB estima que hay 4 millones de servidores MySQL instalados en el mundo, lo significa aproximadamente el 20% del mercado. Entre sus clientes destacan Yahoo!, Cisco, NASA, Lucent Technologies, Motorola, Google, Silicon Graphics, HP, Xerox o Sony Pictures. Buena parte de su éxito se debe, sin duda, a formar parte de la tecnología LAMP.

El 25 de marzo de 2003 se marcó la versión 4.0.12 como la primera versión estable de MySQL 4. Este nuevo MySQL introduce esperadas mejoras entre las que podemos destacar el soporte de transacciones, claves extranjeras (con borrado y actualización en cascada), bloqueo a nivel de fila, caché de consultas, la instrucción UNION y el borrado y actualización multitabla

PHP

Entre las muchas cosas que distinguen la web de los restantes medios de comunicación, está la capacidad de interacción. En este ámbito, las capacidades del HTML, Javascript y demás tecnologías de cliente son bastante reducidas. Una página realmente profesional no puede limitarse a mostrar información y disponer de formularios para conectarse con los usuarios. Esta necesidad se comprendió muy pronto y provocó el nacimiento del protocolo CGI que permite a los navegadores comunicarse con programas alojados en el servidor.

Con los años, no obstante, se comenzaron a percibir diversos problemas con respecto a los CGIs, entre los cuales el menor no era su complejidad. La popularidad de Javascript o Perl llevó a muchas cabezas pensantes a considerar el uso de los lenguajes de script para ejecutar tareas en el servidor. Así nacieron tecnologías como ASP, PHP, JSP o ColdFusion. Vamos a ver cuales son las diferencias de PHP con respecto a las demás alternativas:

1. Es software libre, lo que implica menores costes y servidores más baratos que otras alternativas, a la vez que el tiempo entre el hallazgo de un fallo y su resolución es más corto. Además, el volumen de código PHP libre es mucho mayor que en otras tecnologías, siendo superado por Perl, que es más antiguo. Esto permite construir sitios realmente interesantes con sólo instalar scripts libres como PHP Nuke (weblog, comunidad o bitácora), osCommerce (comercio electrónico con capacidad multilingüe), eZ publish (sistema de gestión de contenidos), phpBB (foros de discusión) o phpMyAdmin (administración de base de datos MySQL).
2. Es muy rápido. Su integración con la base de datos MySQL, también veloz, le permite constituirse como una de las alternativas más atractivas para sitios de tamaño medio-bajo.
3. Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si estás familiarizado con esa sintaxis, PHP o JSP son las opciones más atractivas.
4. Su librería estándar es realmente amplia, lo que permite reducir los llamados 'costes ocultos', uno de los principales defectos de ASP.
5. PHP es relativamente multiplataforma. Funciona en toda máquina que sea capaz de compilar su código, entre ellas diversos sistemas operativos para PC y diversos Unix. El código escrito en PHP en cualquier plataforma funciona exactamente igual en cualquier otra.
6. El acceso a las bases de datos de PHP es muy heterogéneo, pues dispone de un juego de funciones distinto por cada gestor.

7. PHP es suficientemente versátil y potente como para hacer tanto aplicaciones grandes que necesiten acceder a recursos a bajo nivel del sistema como pequeños scripts que envíen por correo electrónico un formulario rellenado por el usuario.
8. Existen menos especialistas en PHP que en ASP en nuestro país.
9. Como lenguaje, PHP padece ciertas carencias: no soporta polimorfismo ni tiene excepciones u otro sistema de errores aceptable.

PHP es una tecnología con mucho futuro, con cada vez más presencia en Internet. Existen muchísimas páginas a lo largo y ancho del mundo que lo utilizan, como Libertad Digital (periódico digital), SourceForge (sistema de albergue de proyectos de software libre), El Mundo (edición digital de un periódico en papel), Gran Avenida (Publicación de ocio y cultura y albergue de páginas personales) o Sport Area (tienda virtual). Por supuesto hay muchos más; en cuanto se navega un poco la extensión .php suena a conocida.

7.3 Linux, instalación de Red Hat

Red Hat Linux es una de las numerosas distribuciones de Linux disponibles, y es la que se encuentra instalada en el servidor de la Biblioteca Digital.

A continuación explicamos como se realiza su instalación y configuración.

COMENZAMOS LA INSTALACIÓN

Si ya tenemos la BIOS configurada para arrancar desde CD, simplemente arranca el ordenador con el CD1 en el lector para comenzar la instalación. Si tu BIOS no puede arrancar desde CD o simplemente no quieres tocar la BIOS, deberás crear un disquete de instalación. *Para crear el disquete de inicio actuaremos igual que en el caso de Linux Mandrake*, pero *la imagen*

necesaria es boot. Luego ejecutamos rawritewin *(desde Windows)* o rawrite *(desde DOS)*, que están en el directorio dosutils del CD1.

Como en el resto de las distribuciones, es aconsejable leer antes INSTALACION DEL S.O. LINUX

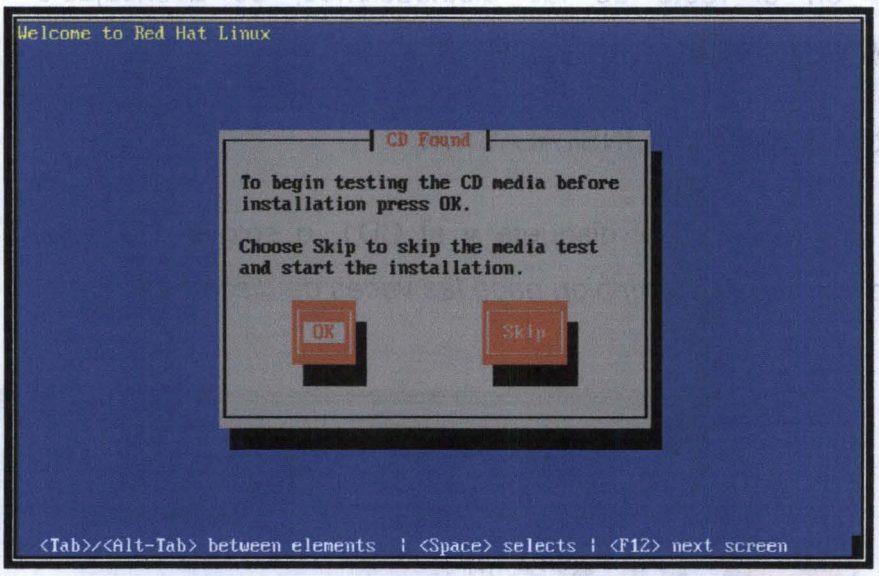
2. ARRANCANDO LA INSTALACIÓN

Introducimos el disquete y el CD1, o sólo el CD según cada uno. Como vemos el *CD también hace las veces de disco de rescate.*

Pantalla de arranque del CD De Red Hat Linux

Pulsamos ENTER para iniciar la instalación gráfica y pasmos a una pantalla donde el programa nos da la opción de *comprobar la integridad del CD* por si

queremos comprobar que todo funcionará correctamente y no faltan archivos o están dañados.



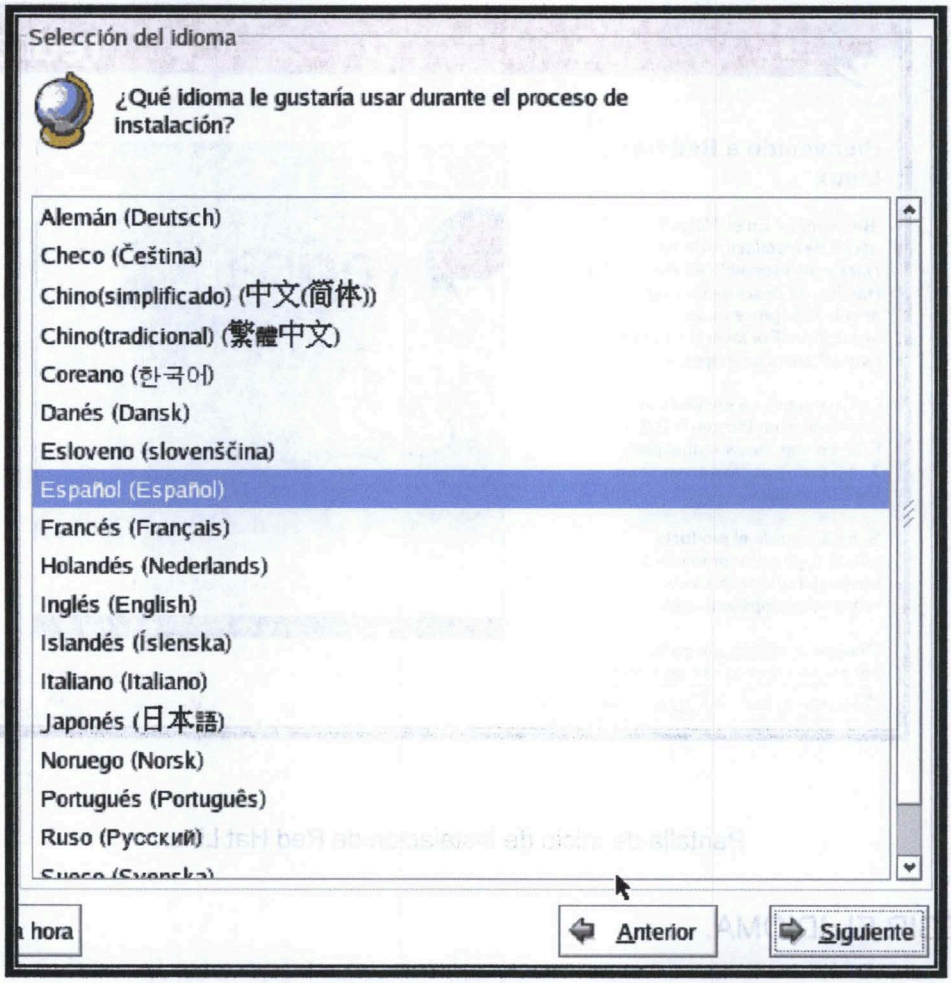
Pantalla de comprobación de integridad del CD de Red Hat Linux

Pasado el paso anterior arranca Anaconda, que es el nombre del *asistente de instalación* de Red Hat. Fíjaté en el cuadro de diálogo que hay a la izquierda donde el programa nos mostrará información y sugerencias sobre cada paso de la instalación, pero por motivos de tamaño no lo mostraré mas, no está de mas que le vayas dando un vistazo durante la instalación.

Pantalla de Inicio de instalación de Red Hat Linux

ELEGIR EL IDIOMA.

El siguiente paso es elegir el idioma de la instalación, para ello sólo tenemos que *marcar el idioma* que deseemos y continuar adelante.

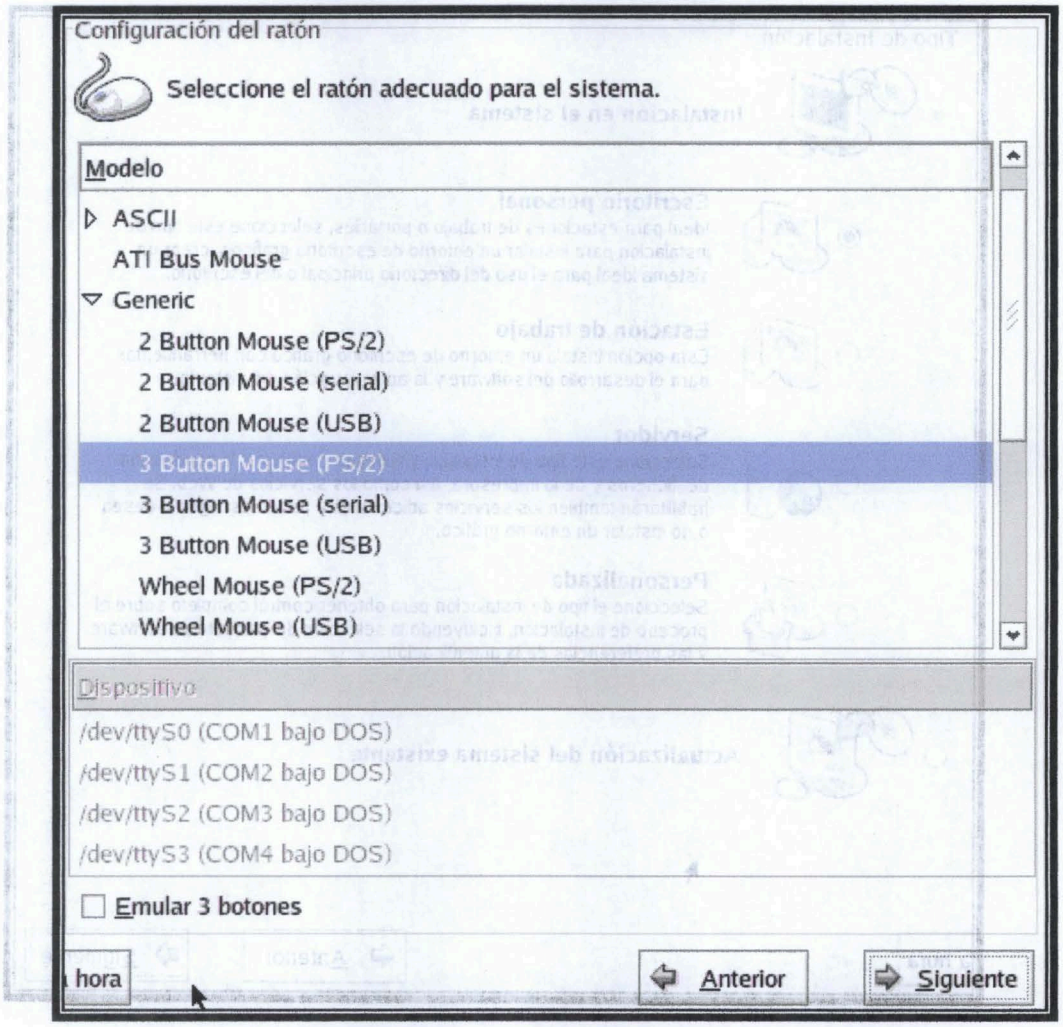


SELECCIONAR TECLADO.

Pues lo mismo que con el idioma.

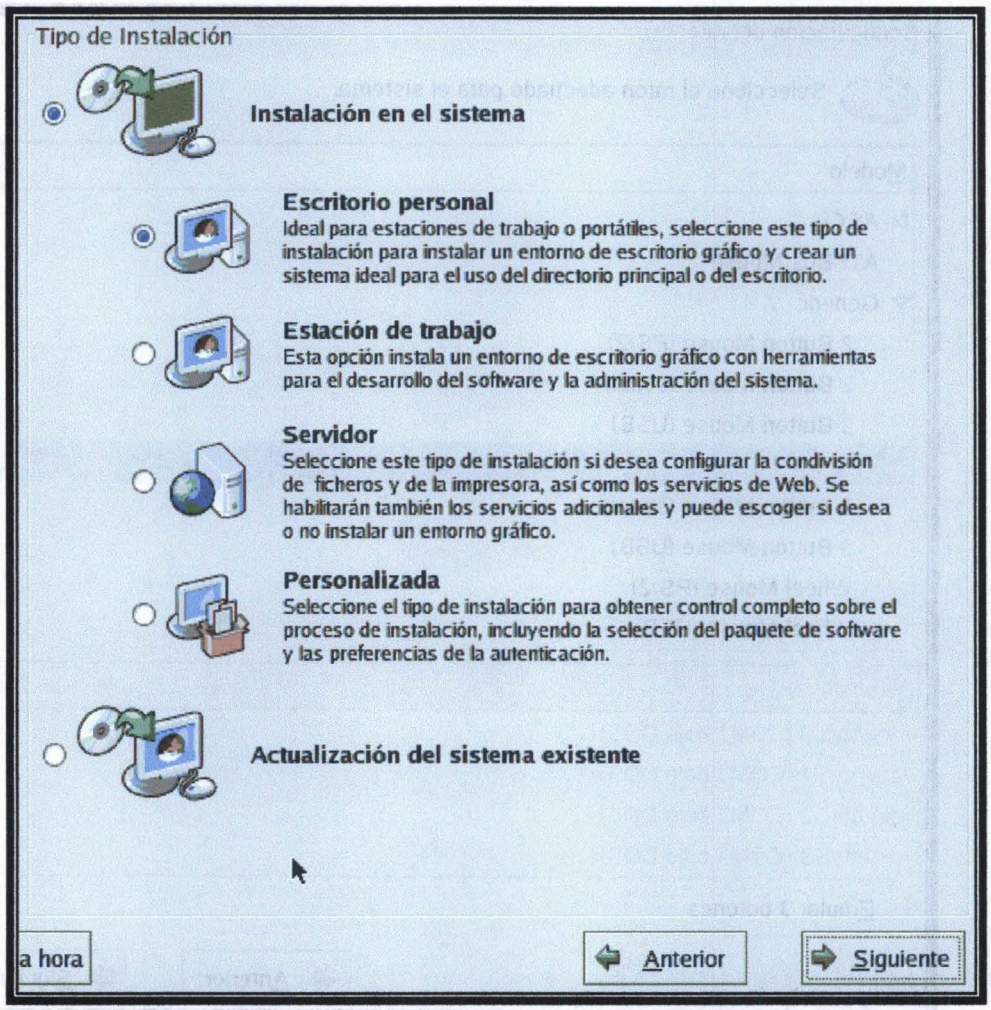
ELEGIR RATÓN.

Eliges un modelo genérico o buscas el tuyo por la lista de fabricantes.



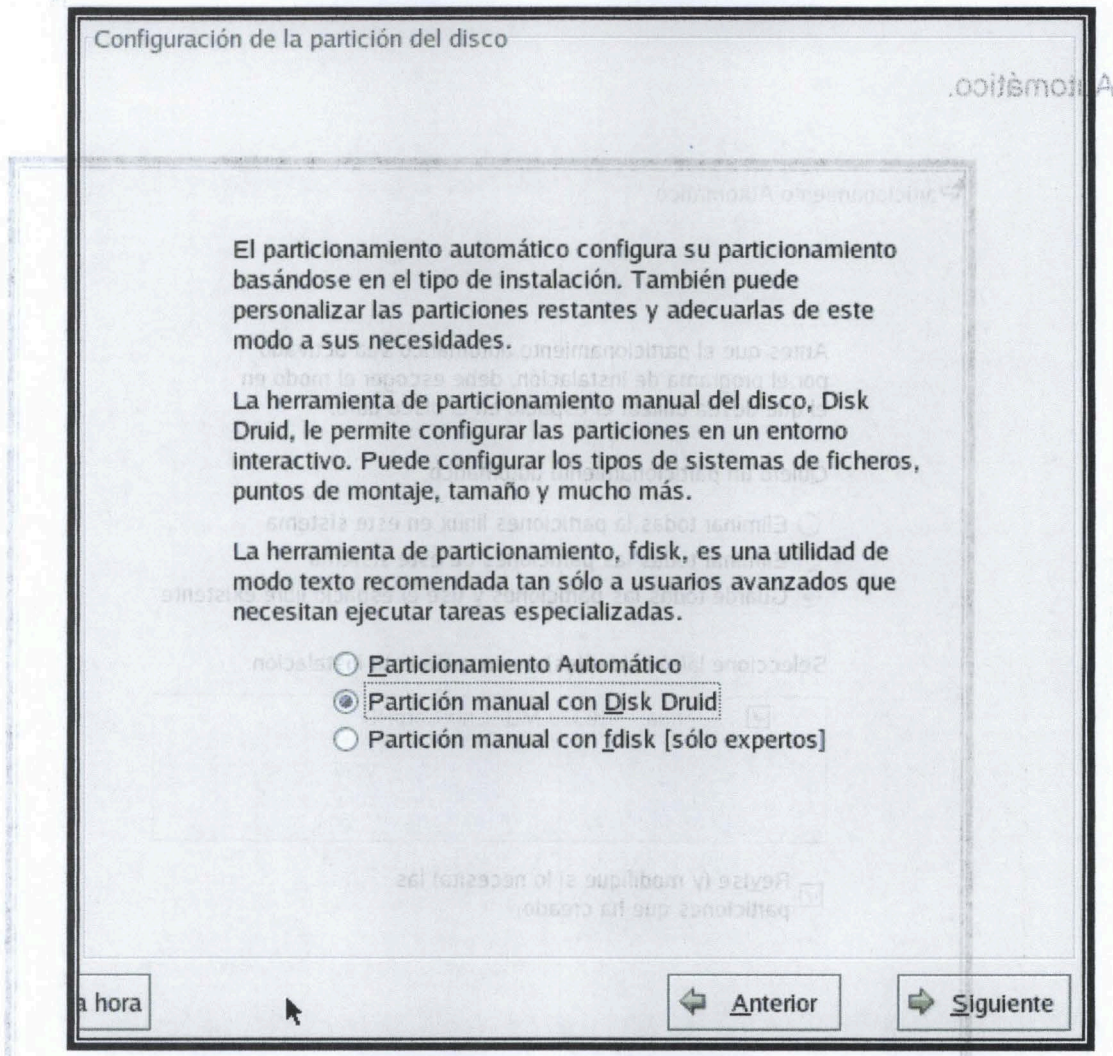
TIPO DE INSTALACIÓN

Dependiendo del uso que le vayamos a dar al ordenador el programa nos muestra unos tipos de *instalaciones por defecto* que incluye aquello que éste considera necesario para ese uso, o bien podemos decantarnos por una *instalación personalizada* donde elegiremos los paquetes a instalar.



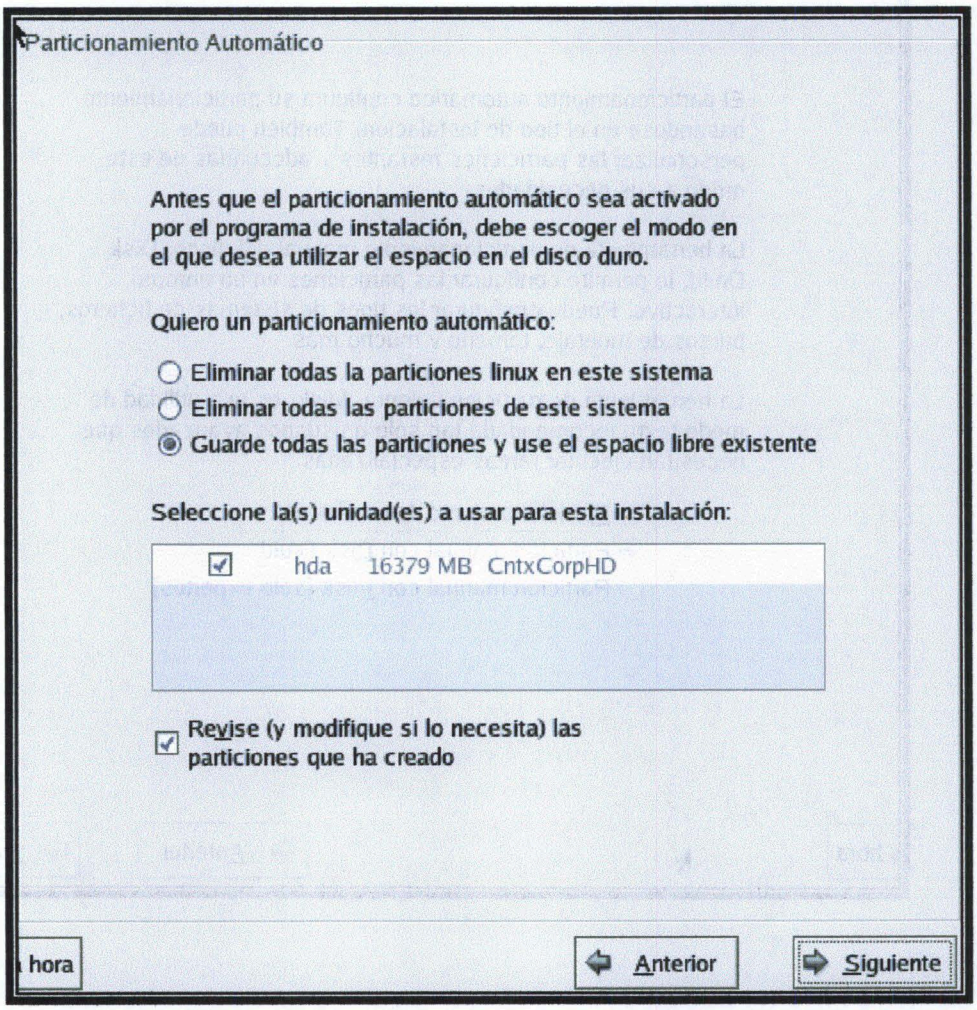
PARTICIONAR EL DISCO DURO.

Aquí el programa nos da la opción de dejar que sea él el que elija el tipo de particionamiento, realizar el particionamiento mediante Disc Druid en modo gráfico, o con fdisk *(con muchas más opciones pero en modo texto, no apto para novatos)*.

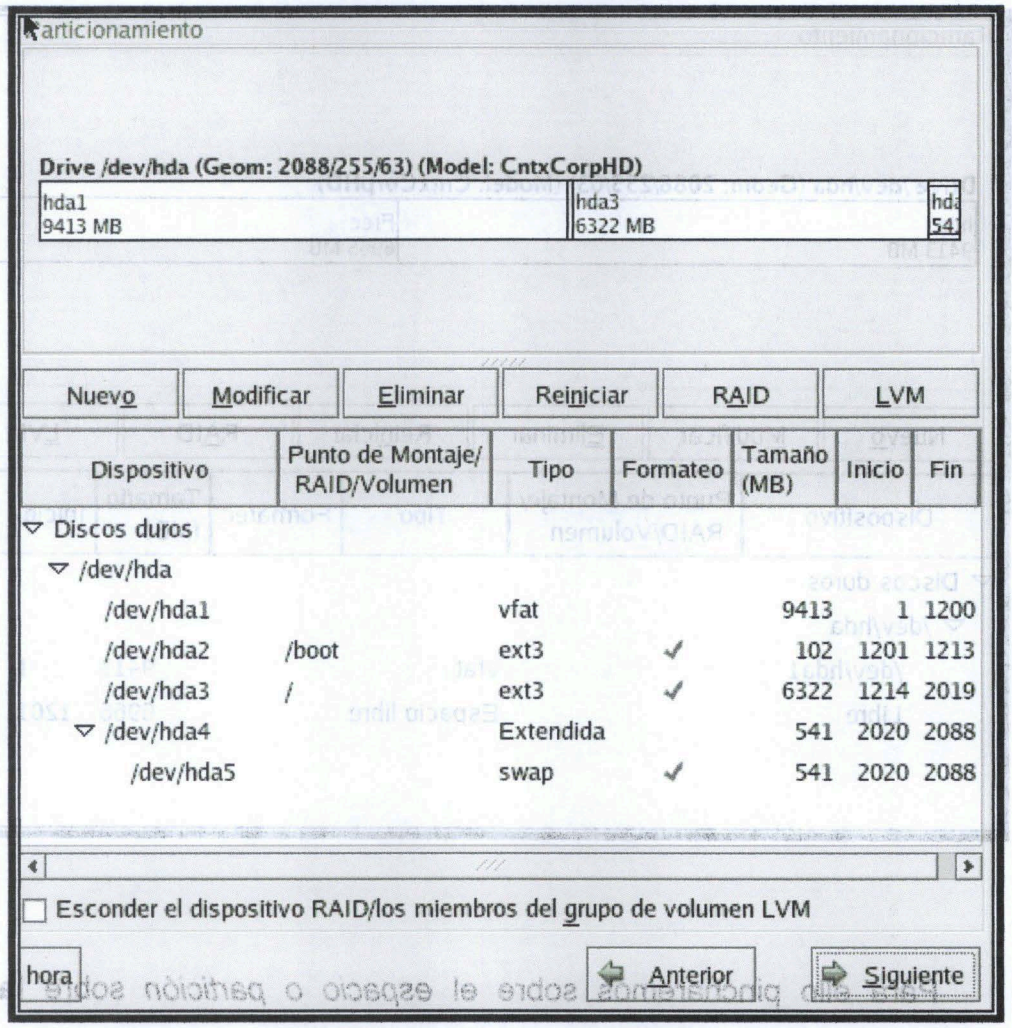


En ninguno de los dos primeros modos (desconozco si con fdisk) tenemos la posibilidad de redimensionar particiones, sólo crearlas o borrarlas, por lo que si tenemos que redimensionar alguna partición debemos de hacerlo antes de comenzar la instalación con otra utilidad para tal fin. Podremos utilizar el PARTITION MAGIC o el PARTITION COMMANDER o ... (mejor desde disquete).

Automático.

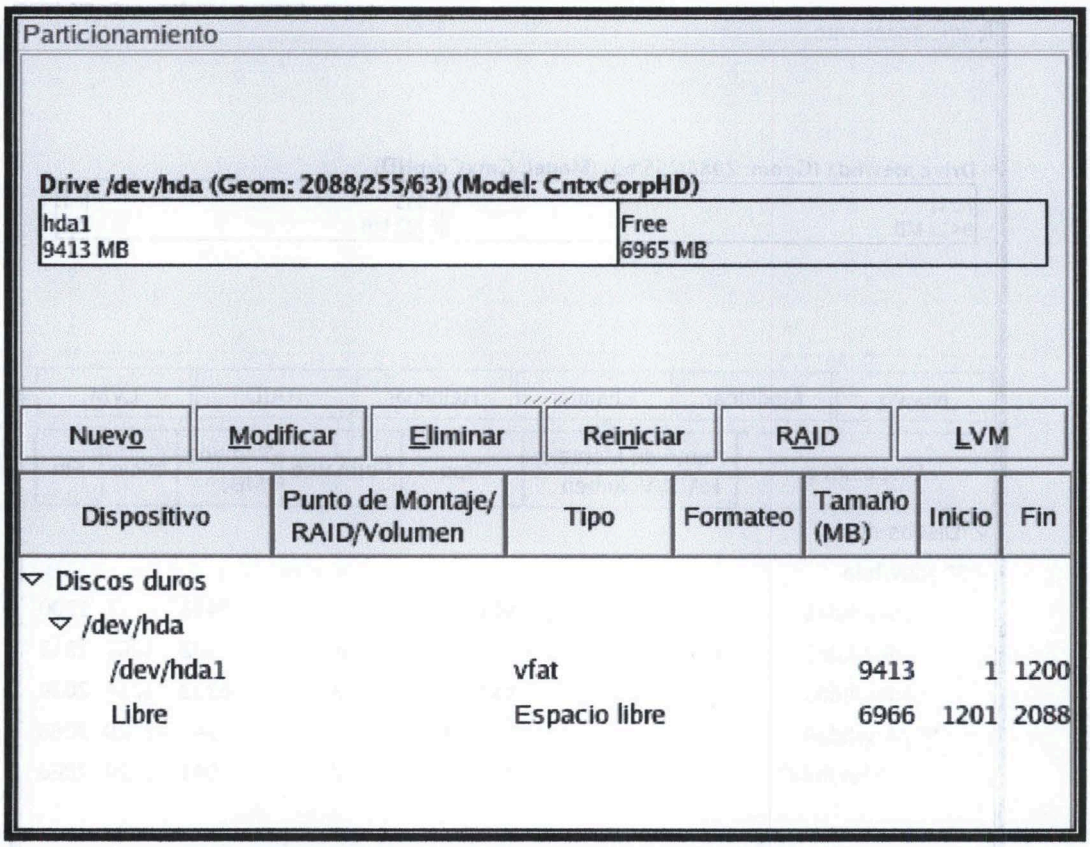


Si ya teníamos Linux instalado podemos elegir la primera opción para que el programa use ese mismo espacio para montar las nuevas particiones Linux. Si queremos todo el disco para Linux elegiremos la segunda opción. Y si antes de la instalación ya creamos un espacio libre para Linux elegiremos la tercera opción *(la elegida en este caso)*. Si marcamos la casilla *"Revise (y modifique.....)"*, pasaremos a Disk Druid, donde podemos *comprobar y modificar* las particiones *(no redimensionar)*, si lo creemos oportuno.

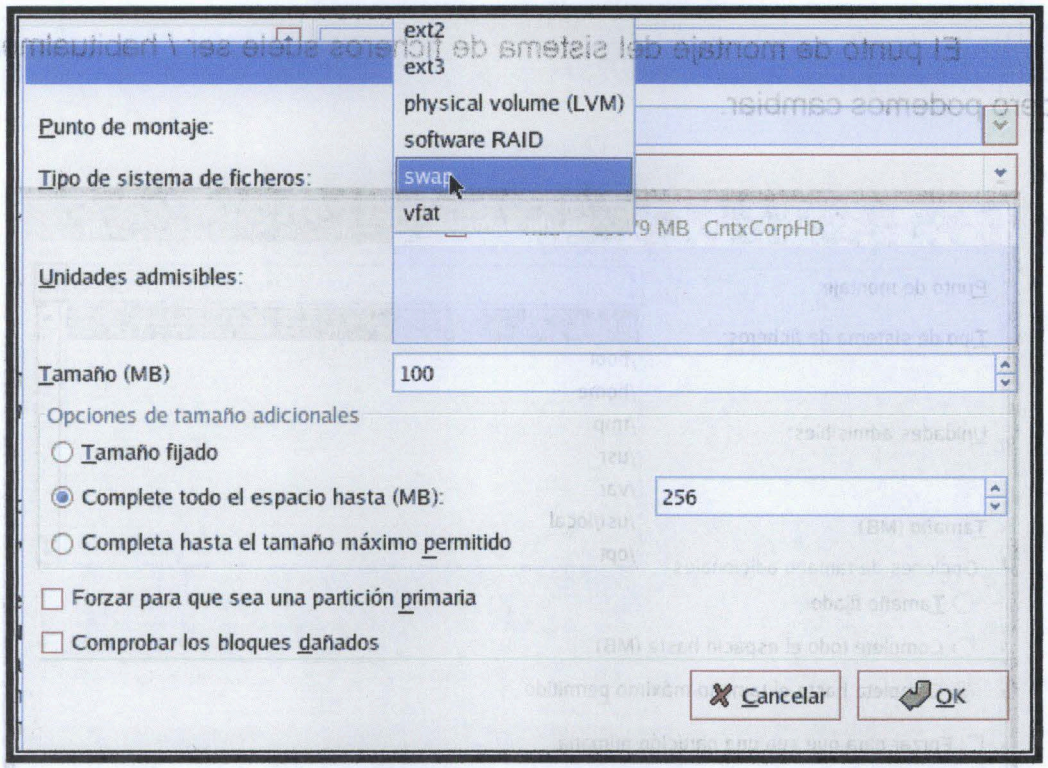


Particionando con Disk Druid.

Mediante esta utilidad podremos *crear o borrar particiones* enn modo gráfico, manualmente.

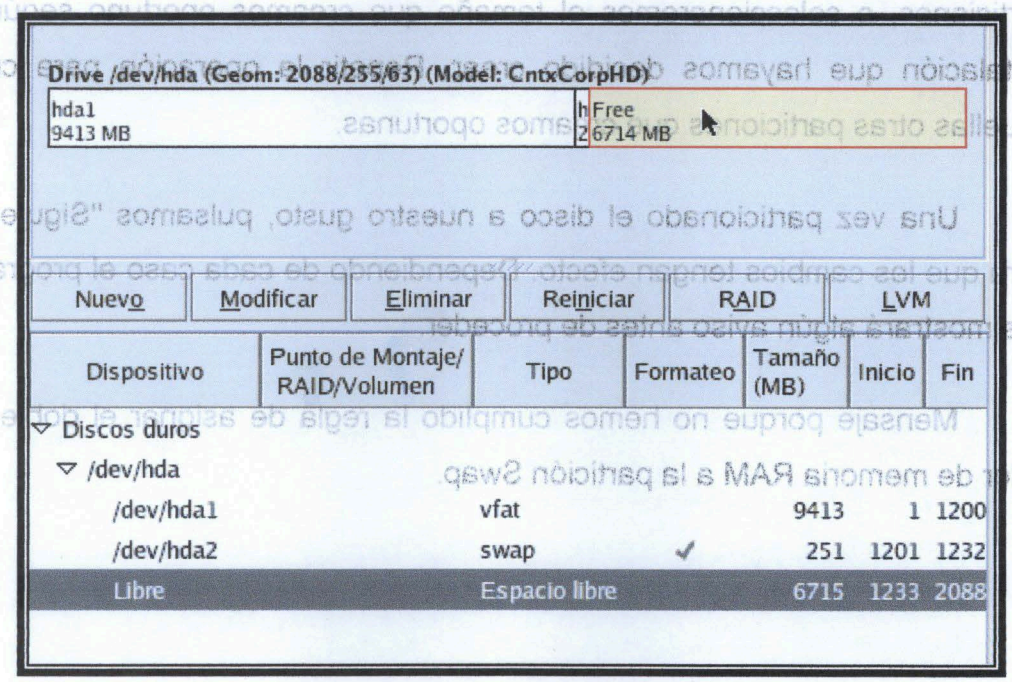


Para ello pincharemos sobre el *espacio o partición* sobre la que queremos trabajar y elegiremos la opción deseada. Comenzaremos por ejemplo creando la *partición de intercambio* Swap

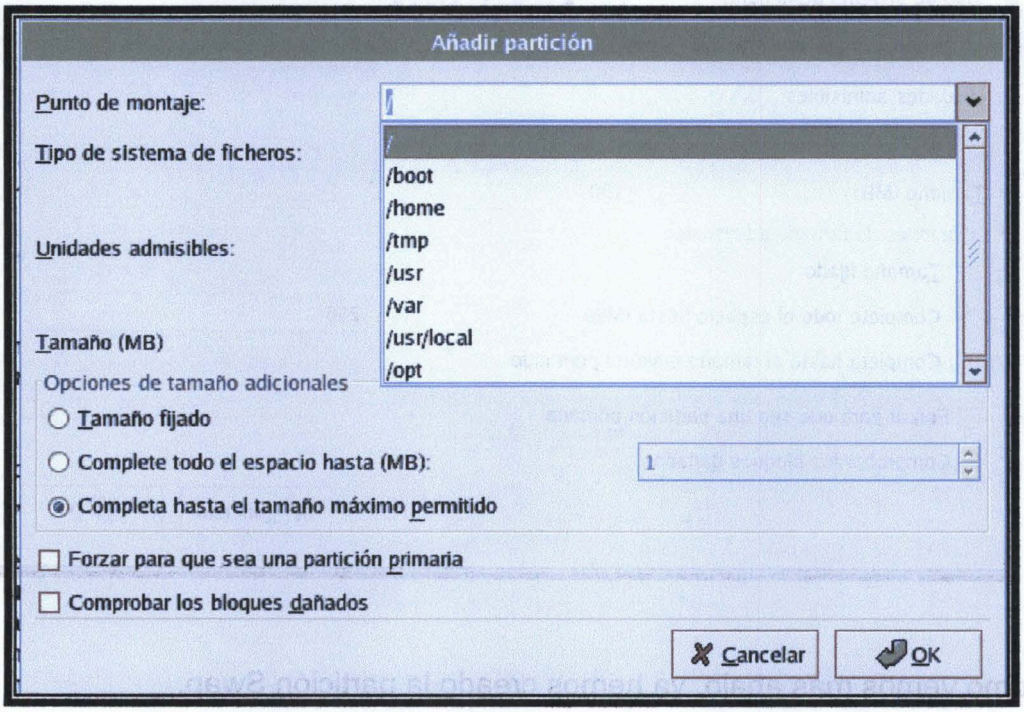


Como vemos más abajo, ya hemos creado la partición Swap,

ahora seleccionaremos el espacio libre para crear la partición raíz/ o /root.



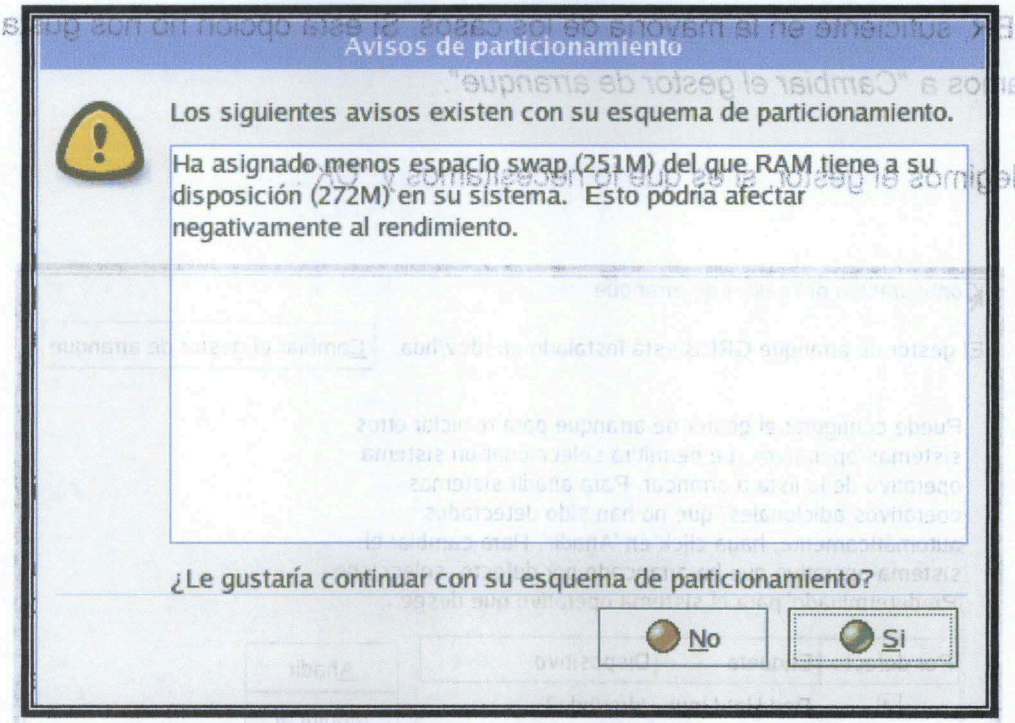
El punto de montaje del sistema de ficheros suele ser / habitualmente, pero podemos cambiar.



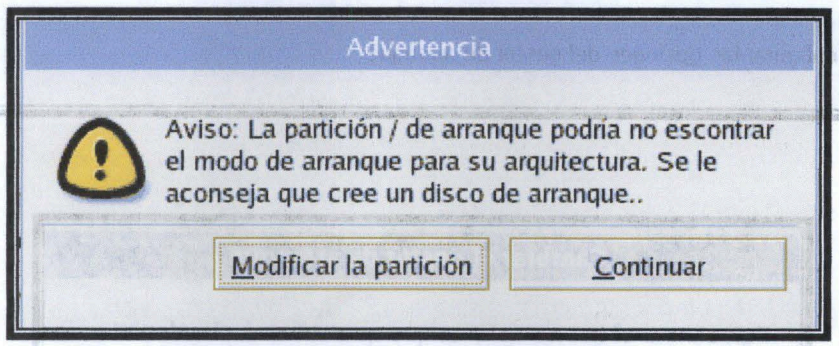
Elegiremos completar todo el tamaño si no vamos a crear más particiones, o seleccionaremos el tamaño que creamos oportuno según la instalación que hayamos decidido crear. Repetir la operación para crear aquellas otras particiones que creamos oportunas.

Una vez particionado el disco a nuestro gusto, pulsamos "Siguiente" para que los cambios tengan efecto. Dependiendo de cada caso el programa nos mostrará algún aviso antes de proceder.

Mensaje porque no hemos cumplido la regla de asignar el doble del valor de memoria RAM a la partición Swap.



Mensaje por estar la partición /boot (dentro de la partición raíz / ,si no la creaste aparte) mas allá del cilindro 1024,por lo que Linux podría no ser arrancable si no es mediante un disquete de arranque.

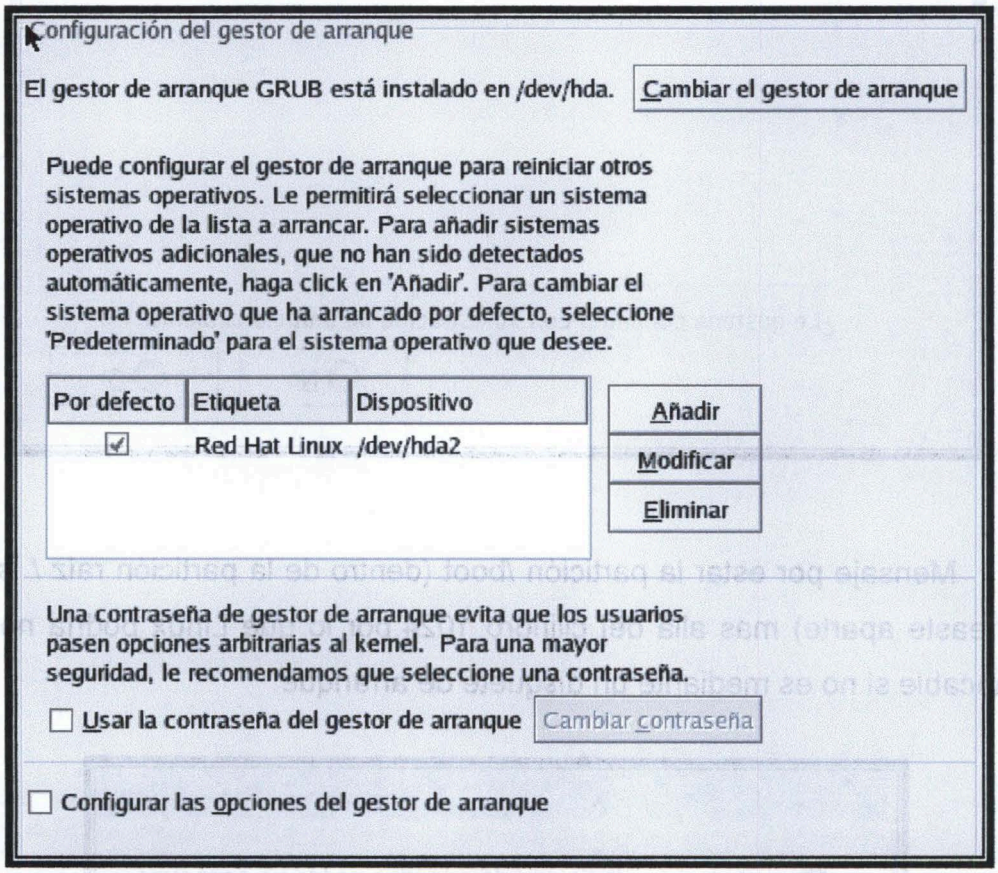


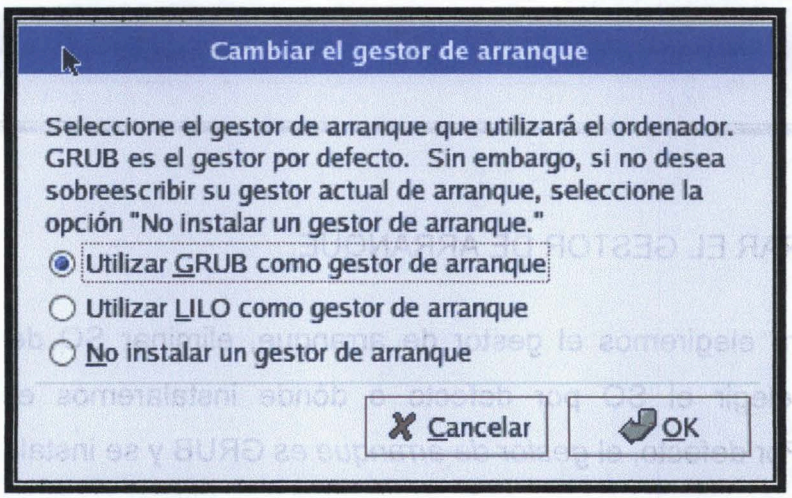
CONFIGURAR EL GESTOR DE ARRANQUE.

Ahora elegiremos el gestor de arranque, eliminar SO de la lista de arranque, elegir el SO por defecto o dónde instalaremos el gestor de arranque. Por defecto, el *gestor de arranque es* GRUB y se instala en el

MBR, suficiente en la mayoría de los casos. Si esta opción no nos gusta, nos vamos a *"Cambiar el gestor de arranque"*.

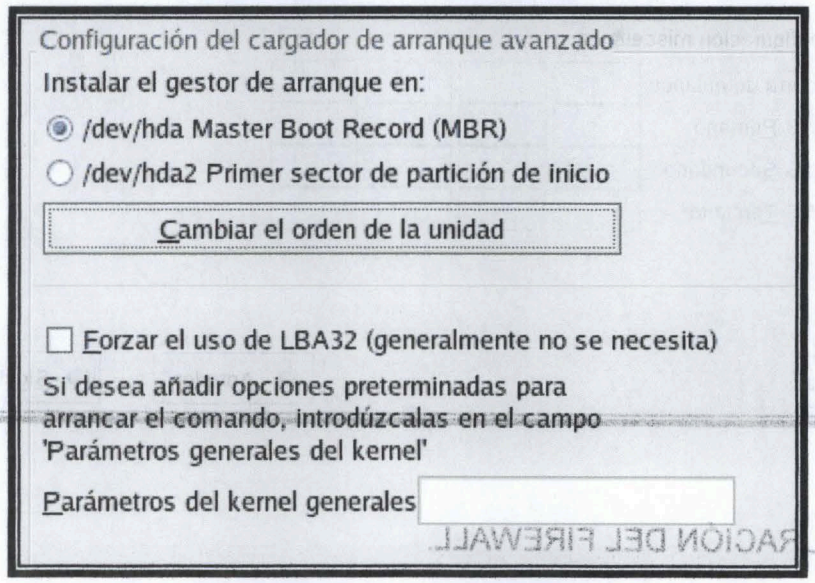
Elegimos el gestor, si es que lo necesitamos y *"OK"*.





Y marcaremos la casilla *"Configurar las opciones del gestor de arranque"* para elegir dónde se instalará. Según las particiones de nuestro disco, se nos mostrarán más o menos opciones.

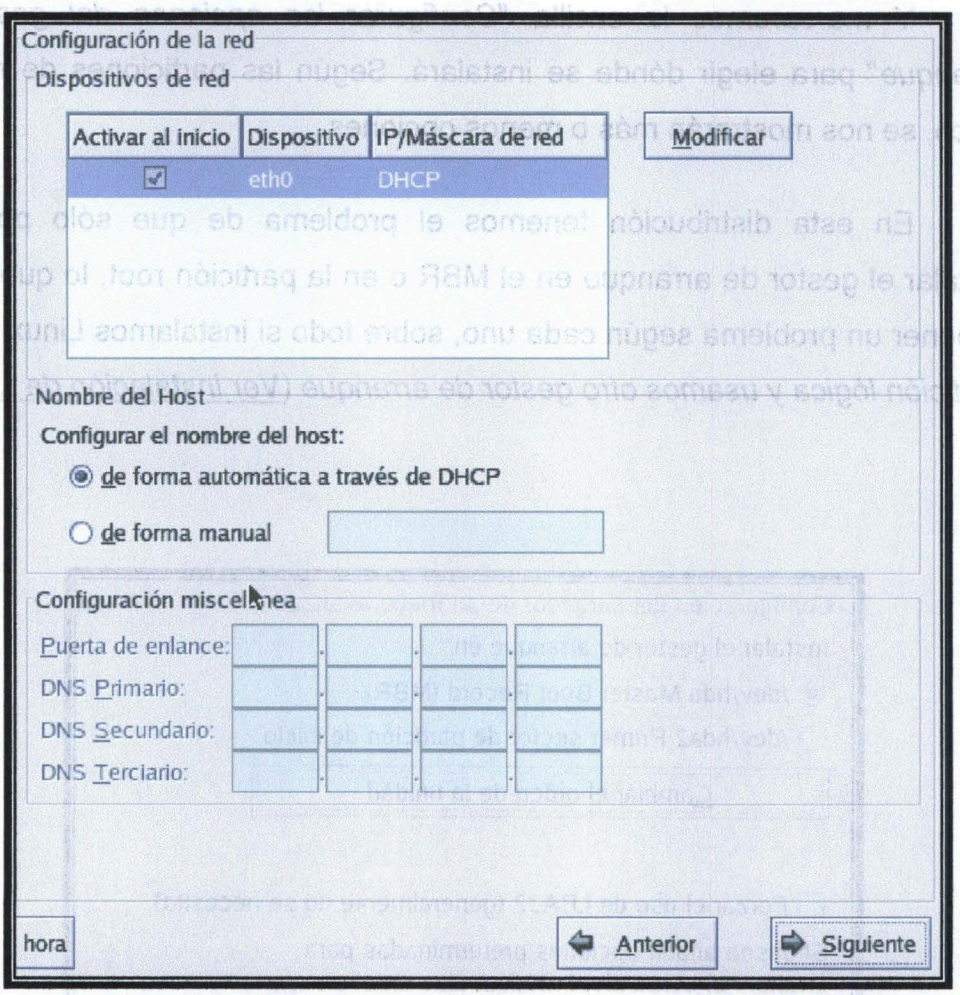
En esta distribución tenemos el problema de que sólo podemos instalar el gestor de arranque en el MBR o en la partición root, lo que puede suponer un problema según cada uno, sobre todo si instalamos Linux en una *partición lógica y usamos otro gestor de arranque* (*Ver instalación de Linux*).



3. CONFIGURACIÓN

CONFIGURACIÓN DE RED.

El programa detectará automáticamente los dispositivos instalados, y el modo de configuración.

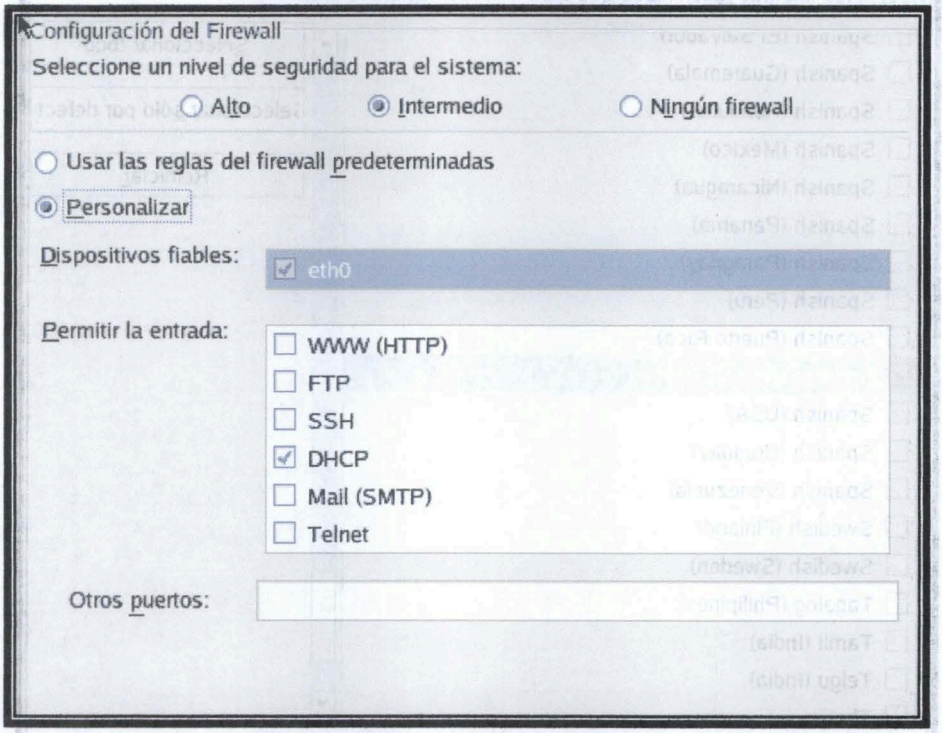


CONFIGURACIÓN DEL FIREWALL.

Aquí debemos mirar en el cuadro de diálogo y leer la información que se nos da para una correcta configuración del firewall *(cortafuegos)* que el programa trae por defecto.

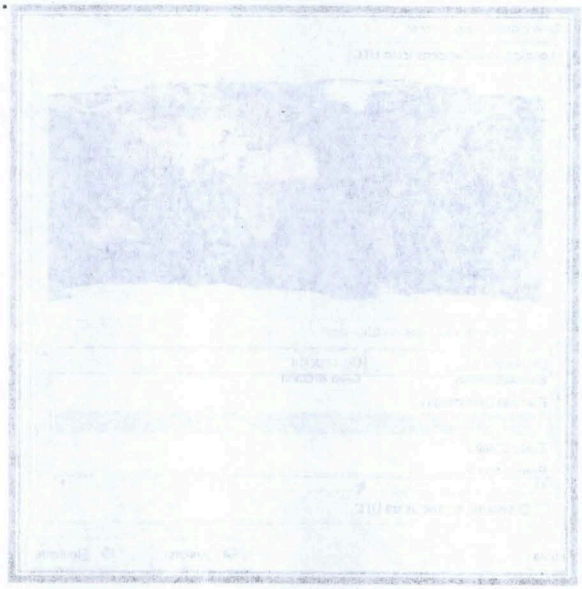
Para un uso normal de un PC conectado a internet: l*a configuración Alto y Usar las reglas del firewall predeterminadas es la más apropiada*, ya que por defecto permite la entrada de datos DHCP, DNS, FTP *(no como servidor)*, IRC, y servidores X window remotos. Sólo si vamos a instalar algún tipo de servidor marcaremos la casilla en personalizar, o introduciremos el

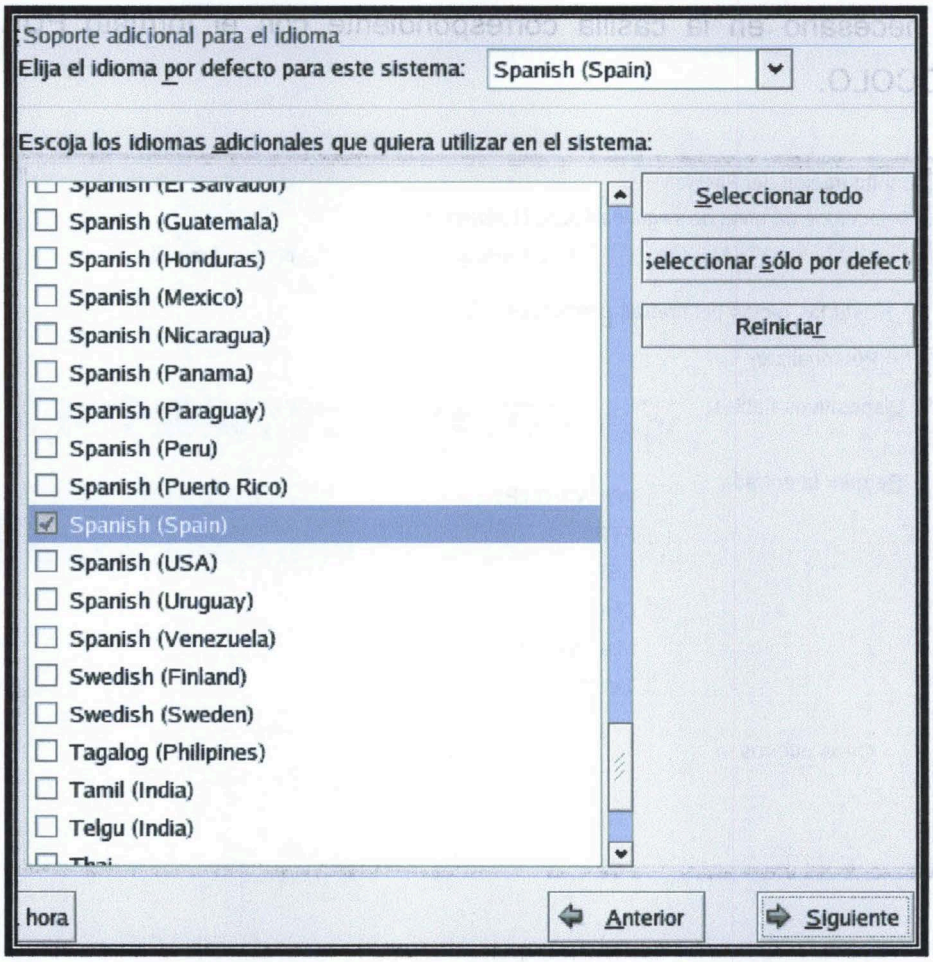
puerto necesario en la casilla correspondiente con el formato PUERTO: PROTOCOLO.



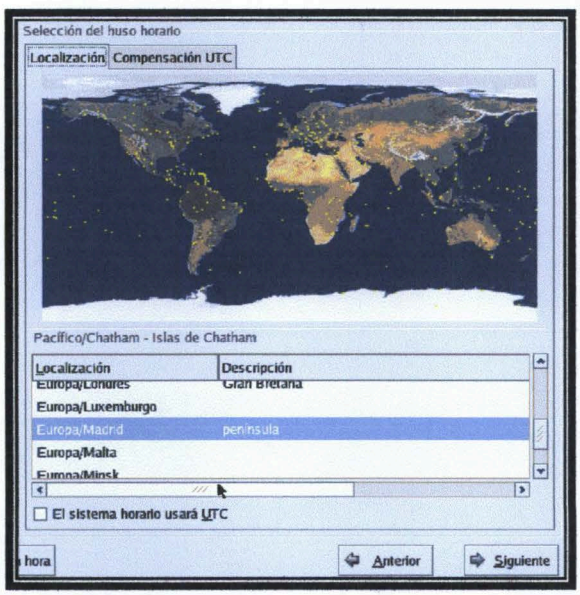
SOPORTE ADICIONAL DE IDIOMA.

Puedes elegir idiomas adicionales que se instalarán y tendrás disponibles para elegir.



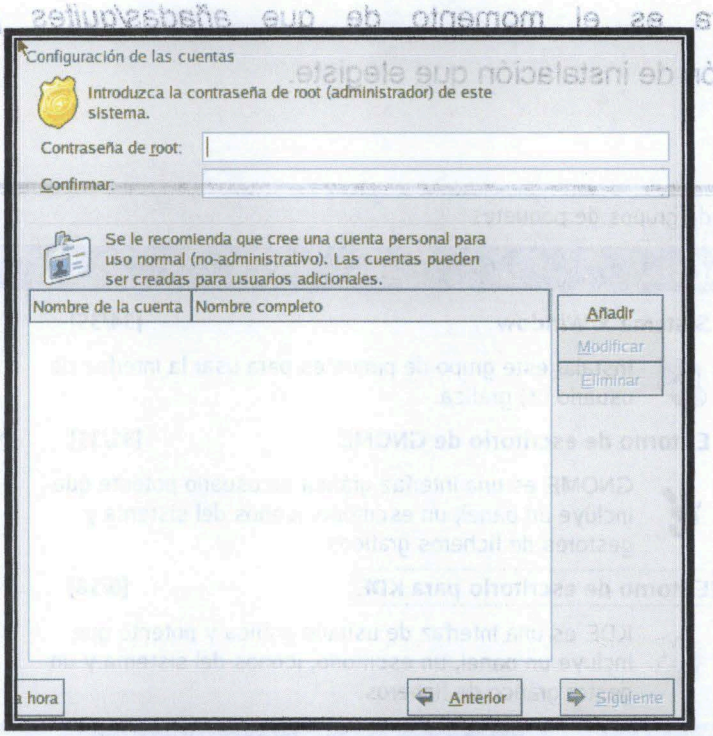


USO HORARIO.

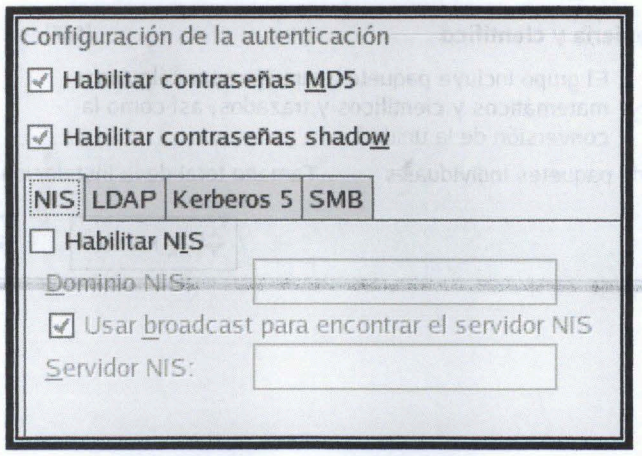


CONTRASEÑA DE ROOT Y AÑADIR USUARIOS.

Ahora es el momento de introducir la contraseña de root, así como los usuarios y contraseñas de los usuarios (luego se pueden añadir/quitar/editar).



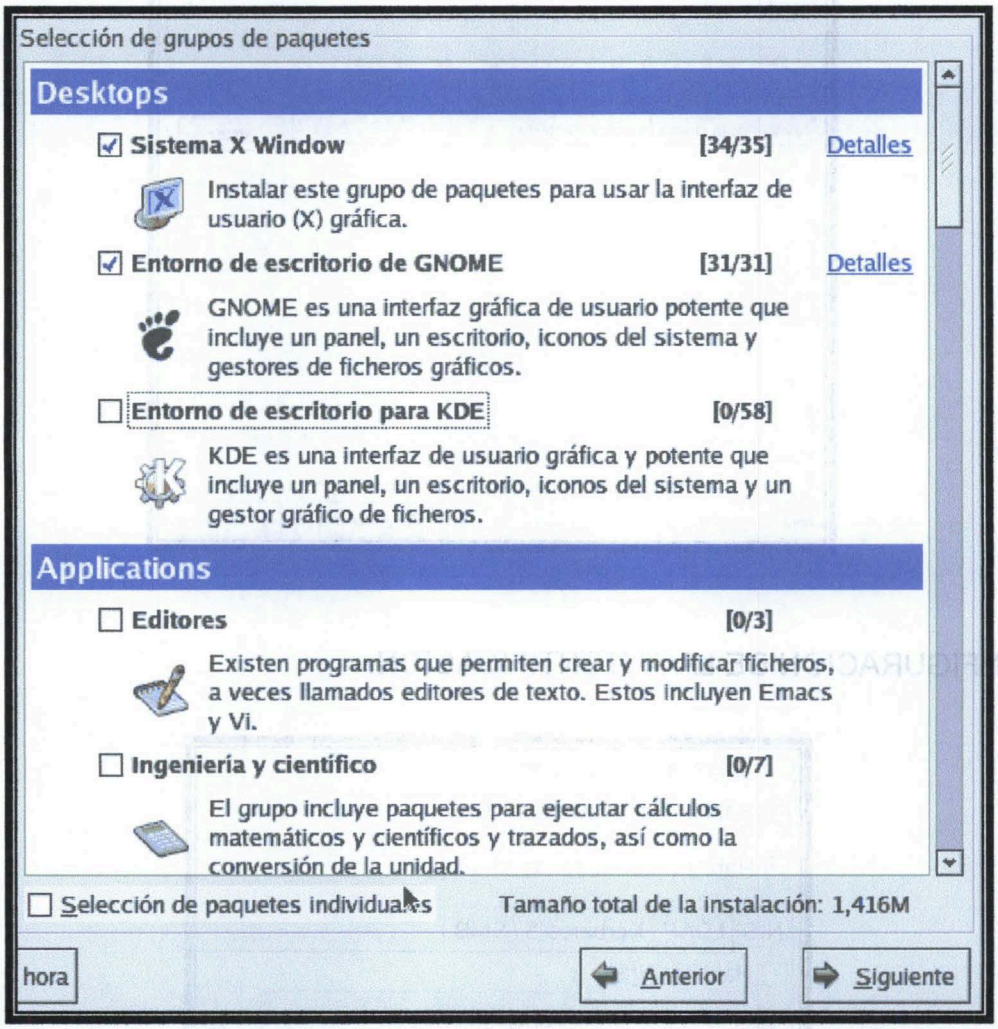
CONFIGURACIÓN DE LA AUTENTIFICACIÓN.



Aquí se seleccionan el tipo de contraseñas que se permitirán, como se guardaran en el disco y cosas así. *Deja las casillas marcadas por defecto*, si quieres saber más dale un vistazo al cuadro de diálogo durante la instalación.

SELECCIÓN DE PAQUETES.

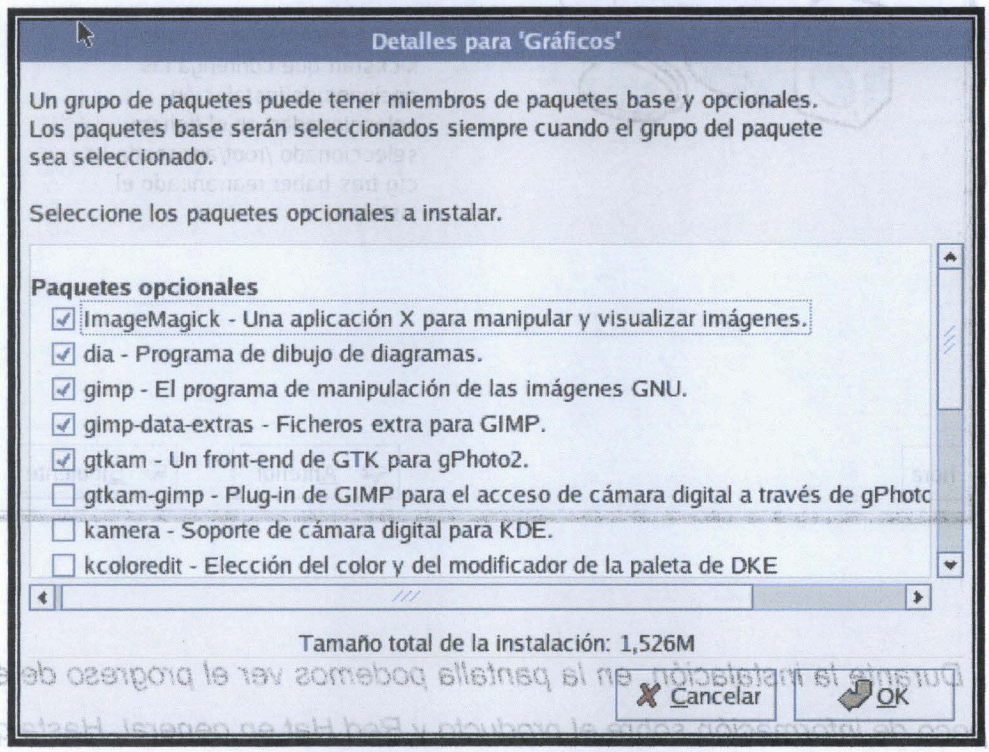
Ahora es el momento de que *añadas/quites paquetes* a la configuración de instalación que elegiste.

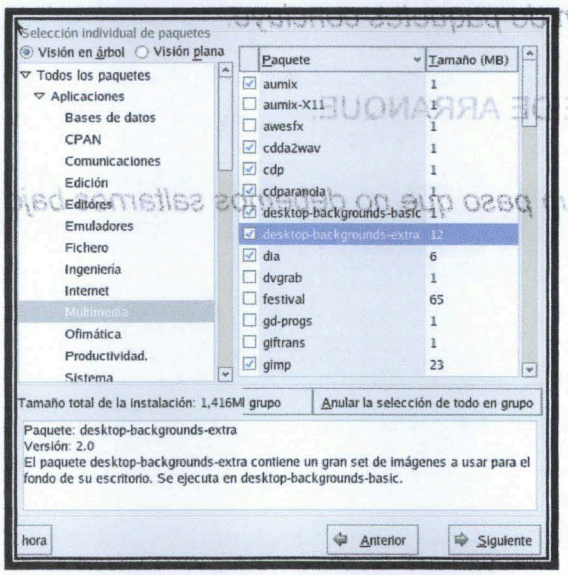


Desplazando la barra de la derecha, podemos ir viendo todos los servicios que se incluyen en la lista *(impresión, ofimática, internet, servidores, etc.)*,

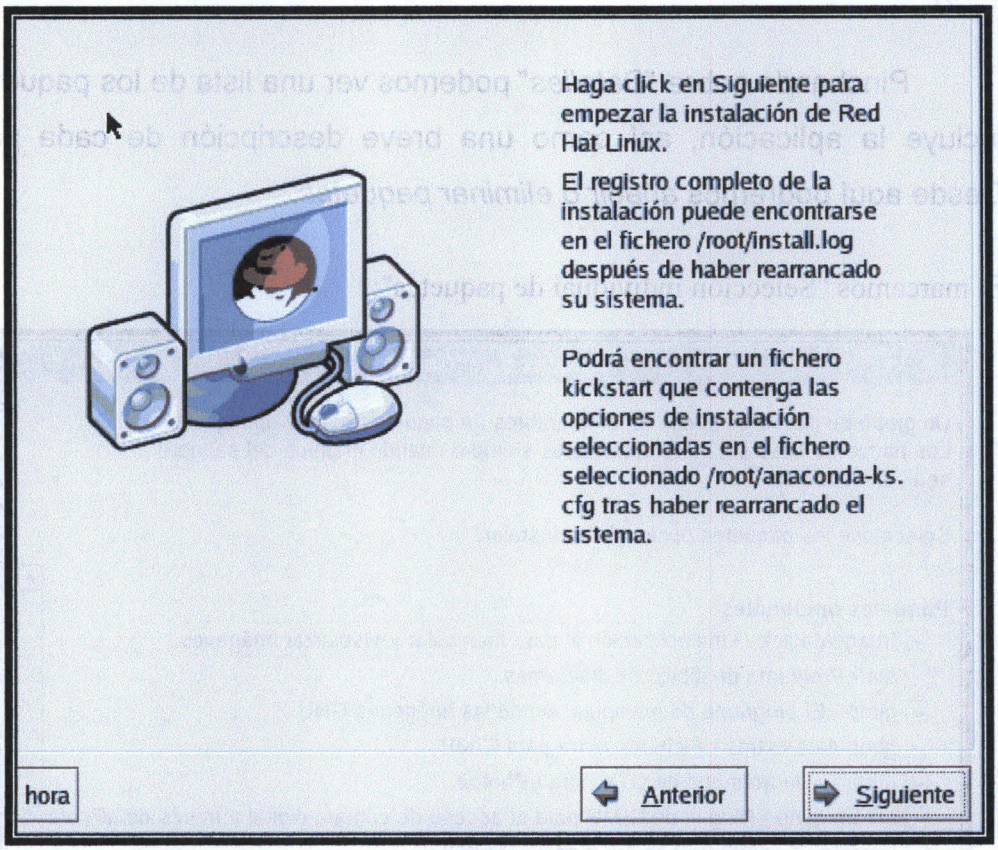
Pinchando sobre *"Detalles"* podemos ver una lista de los paquetes que incluye la aplicación, así como una breve descripción de cada paquete. Desde aquí podremos *añadir o eliminar paquetes*.

Si marcamos "Selección individual de paquetes":





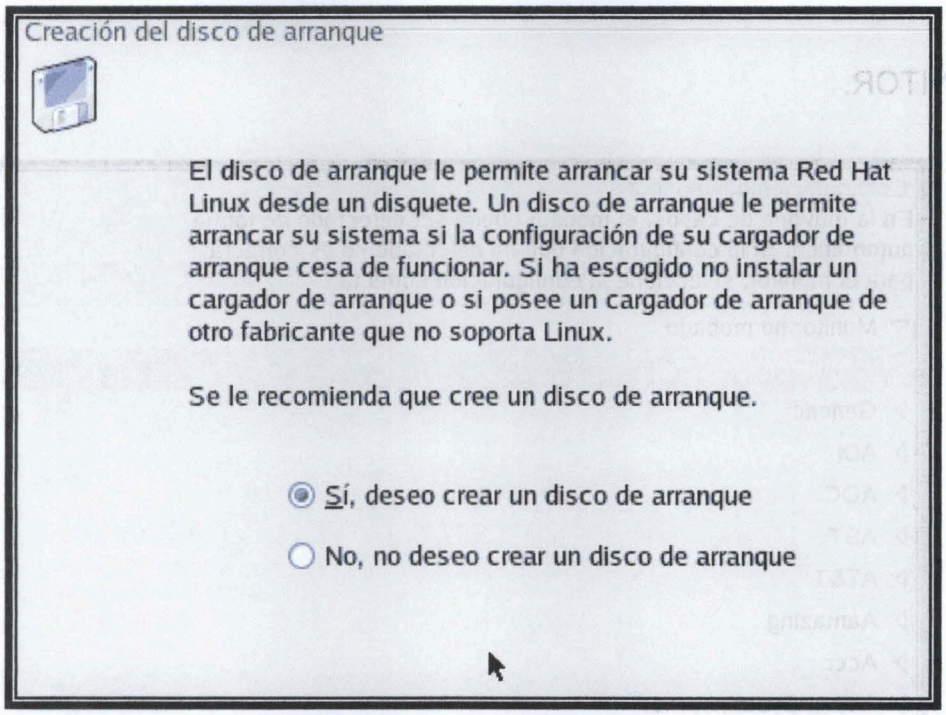
Una vez seleccionados los paquetes, pulsamos "Siguiente" para comenzar la instalación.



Durante la instalación, en la pantalla podemos ver el progreso de ésta y un poco de información sobre el producto y Red Hat en general. Hasta que, por fin, la instalación de paquetes concluye.

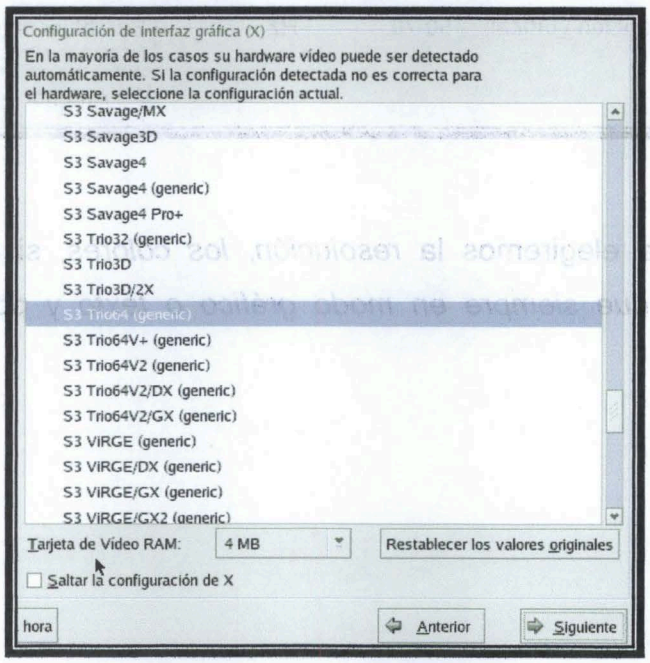
CREAR DISQUETE DE ARRANQUE.

Como siempre, es un paso que no debemos saltarnos bajo ningún concepto.

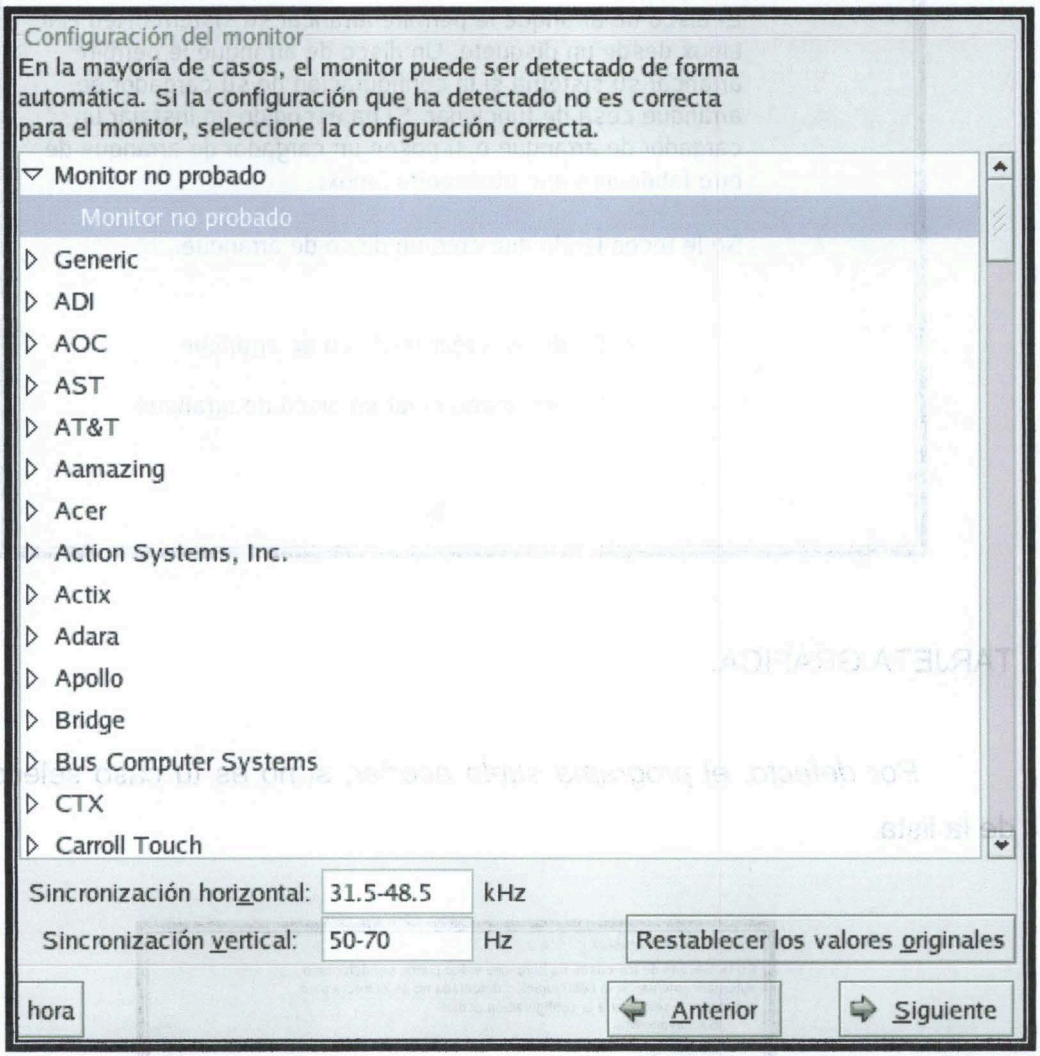


TARJETA GRÁFICA.

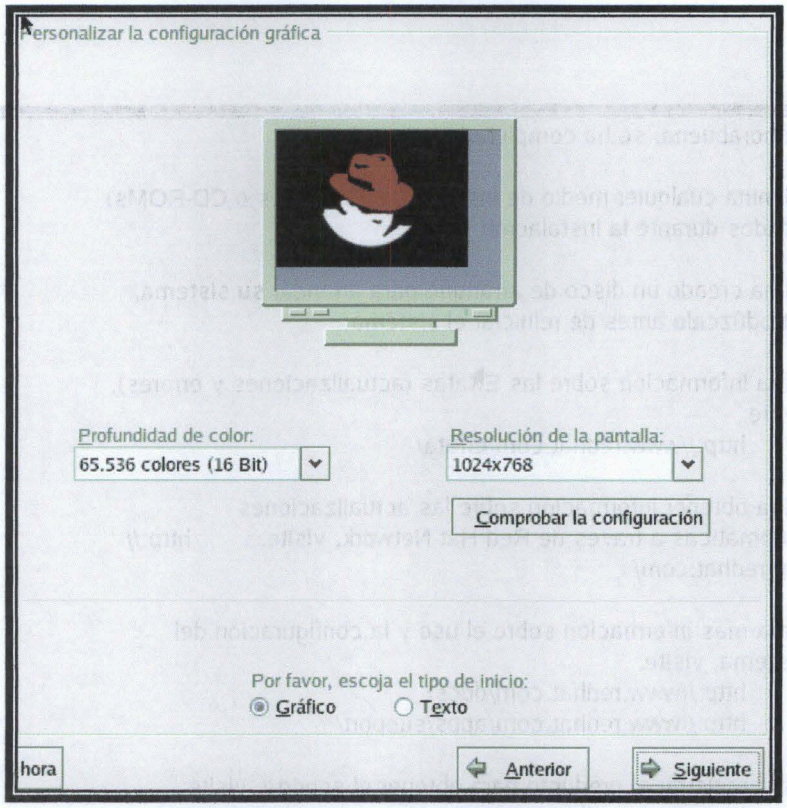
Por defecto, el programa suele acertar, si no es tu caso selecciónala de la lista.



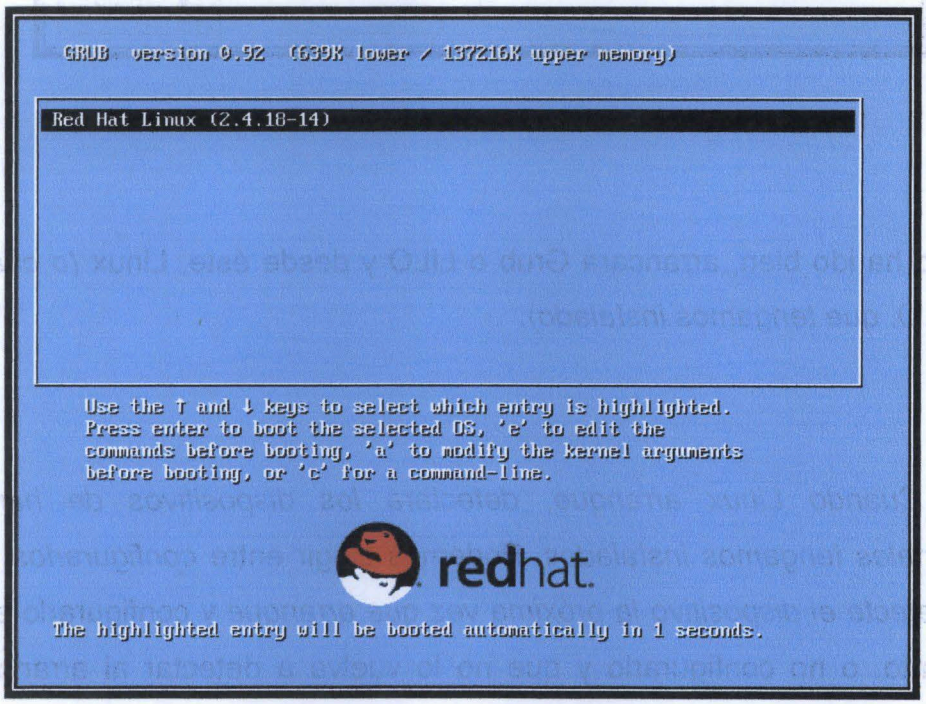
MONITOR.

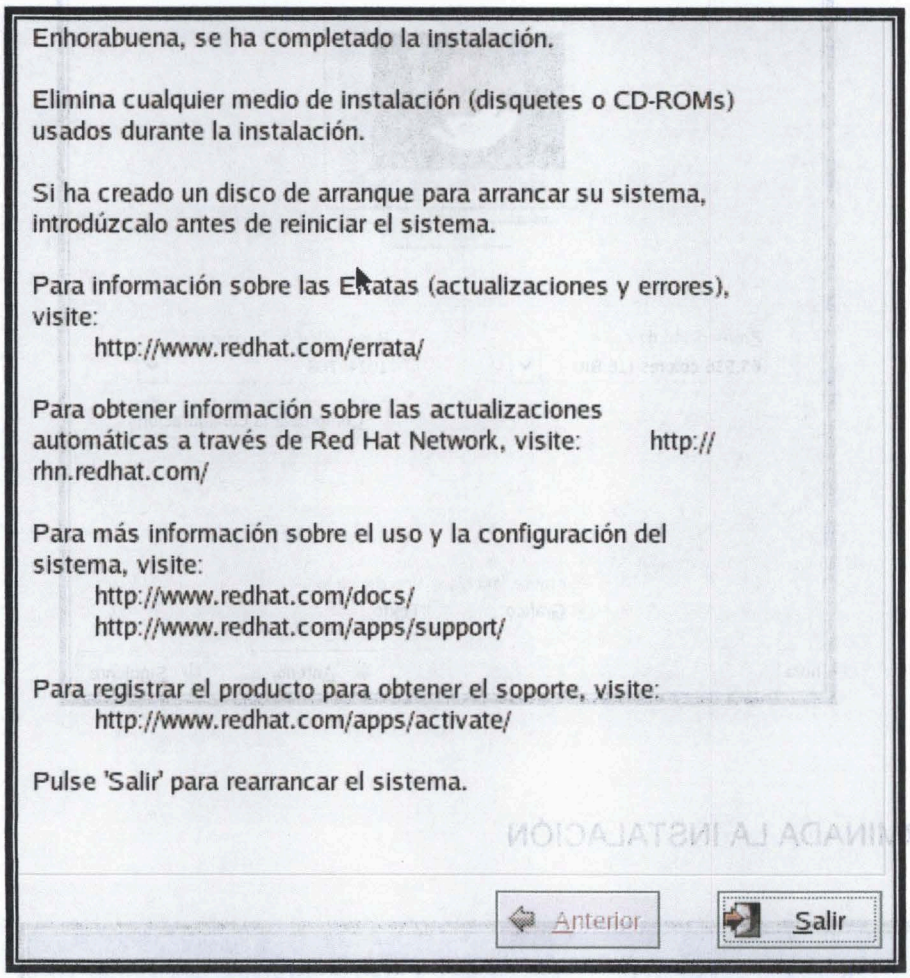


Y ahora elegiremos la *resolución, los colores*, si queremos que el *sistema arranque siempre en modo gráfico o texto* y podemos *probar la configuración*.



4. TERMINADA LA INSTALACIÓN





Si todo ha ido bien, arrancará Grub o LILO y desde éste, Linux *(o cualquier otro S.O. que tengamos instalado)*.

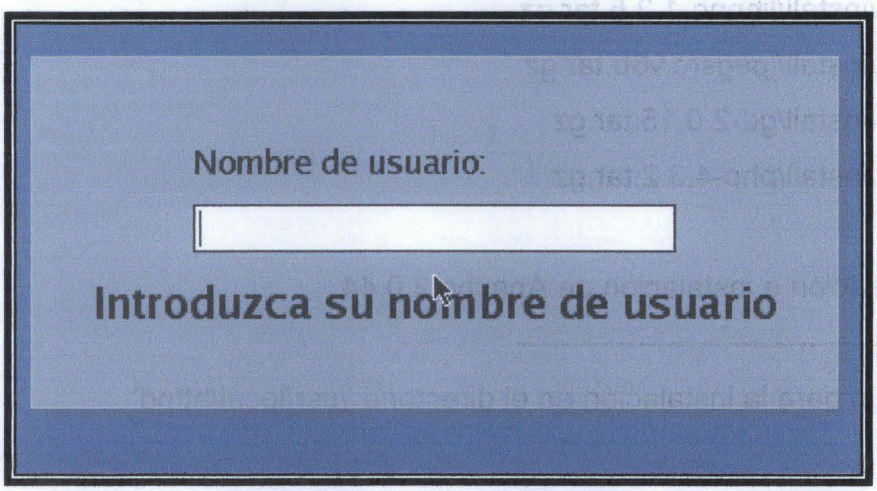
Cuando Linux arranque, detectará los dispositivos de hardware adicionales tengamos instalados. Podemos elegir entre *configurarlos ahora*, que *detecte el dispositivo la próxima vez que arranque* y configurarlo en otro momento, o no configurarlo y que no lo vuelva a detectar al arrancar; de

todos modos luego puedes configurarlos *(en inglés, por lo menos en el escritorio Gnome)*.

La primera vez, nos mostrará una pantalla donde podremos *registrar el producto* si queremos, *probar la tarjeta de sonido*, e instalar más paquetes si es que tenemos *CD adicionales (supongo que los que vendrán en la version comprada)*.

LOGIN Y PASSWORD.

Ya sólo nos queda identificarnos para poder empezar



7.4. Instalación LAMP paso a paso

Descarga de los siguientes ficheros de fuentes en:

/home/install/httpd-2.0.44.tar.gz
/home/install/mysql-3.23.56.tar.gz
/home/install/freetype-2.1.4.tar.gz
/home/install/zlib-1.1.4.tar.gz
/home/install/libpng-1.2.5.tar.gz
/home/install/jpegsrc.v6b.tar.gz
/home/install/gd-2.0.15.tar.gz
/home/install/php-4.3.2.tar.gz

Compilación e instalación de Apache 2.0.44

Ejemplo para la instalación en el directorio '/usr/local/httpd'

```
# cd /usr/src
# tar zxvf /home/install/httpd-2.0.44.tar.gz
# cd /usr/src/httpd-2.0.44
# mkdir /usr/local/httpd
# make clean
# ./configure --prefix=/usr/local/httpd --enable-so
# make
# make install
```

Arrancar Apache:

```
# /usr/local/httpd/bin/apachectl start
```

Parar Apache:

```
# /usr/local/httpd/bin/apachectl stop
```

Compilación en instalación de MySQL 3.23.56

Ejemplo para la instalación en el directorio '/usr/local/mysql'

```
# cd /usr/src
# tar zxvf /home/install/mysql-3.23.56.tar.gz
# cd /usr/src/mysql-3.23.56
# mkdir /usr/local/mysql
# make clean
# ./configure --prefix=/usr/local/mysql
# make
# make install
# useradd -g root mysql
# /usr/local/mysql/bin/mysql_install_db
# chown -R mysql /usr/local/mysql/var
# /usr/local/mysql/bin/safe_mysqld & (Arranque del demonio de MySQL)
# /usr/local/mysql/bin/mysqladmin -u root password 'new-password'
# /usr/local/mysql/bin/mysqladmin -u root -p -h localhost password 'new-password'
```

Configurar arranque de Apache y MySQL con el sistema operativo

---

Añadimos al fichero /etc/rc.local las lineas:

```
/usr/local/httpd/bin/apachectl start
/usr/local/mysql/bin/safe_mysqld &
```

Compilación e instalación de librería de fuentes: freetype 2.1.4

Ejemplo para la instalación en el directorio '/usr/local/freetype'

```
# cd /usr/src
# tar zxvf /home/install/freetype-2.1.4.tar.gz
# cd /usr/src/freetype-2.1.4
# mkdir /usr/local/freetype
# make clean
# ./configure --prefix=/usr/local/freetype
# make
# make install
```

Compilación e instalación de librería de compresión: zlib 1.1.4

Ejemplo para la instalación en el directorio '/usr/local/zlib'

```
# cd /usr/src
# tar zxvf /home/install/zlib-1.1.4.tar.gz
# cd /usr/src/zlib-1.1.4
# mkdir /usr/local/zlib
# make clean
# ./configure --prefix=/usr/local/zlib
# make
# make install
```

Compilación e instalación de librería gráfica: libpng 1.2.5

Ejemplo para la instalación en el directorio '/usr/local/libpng'

```
# cd /usr/src
```

```
# tar zxvf /home/install/libpng-1.2.5.tar.gz
# cd /usr/src/libpng-1.2.5
# mkdir /usr/local/libpng
# make clean
# cp scripts/makefile.linux makefile
```

Editamos el fichero '/usr/src/libpng-1.2.5/makefile' y modificamos:

```
        prefix=/usr/local/libpng
        ZLIBLIB=/usr/local/zlib/lib
        ZLIBINC=/usr/local/zlib/include
# make
# make install
```

Compilación e instalación de librería gráfica: jpeg 6b

Ejemplo para la instalación en el directorio '/usr/local/jpeg'

```
# cd /usr/src
# tar zxvf /home/install/jpegsrc.v6b.tar.gz
# cd /usr/src/jpeg-6b
# mkdir /usr/local/jpeg
# make clean
# ./configure --prefix=/usr/local/jpeg
# make
# mkdir /usr/local/jpeg/bin
# mkdir /usr/local/jpeg/include
# mkdir /usr/local/jpeg/lib
# mkdir /usr/local/jpeg/man
# mkdir /usr/local/jpeg/man/man1
# make install
# make install-lib
```

```
# make install-headers
```

Compilación e instalación de librería gráfica: gd-2.0.15

Ejemplo para la instalación en el directorio '/usr/local/gd'

```
# cd /usr/src
# tar zxvf /home/install/gd-2.0.15.tar.gz
# cd /usr/src/gd-2.0.15
# mkdir /usr/local/gd
# make clean
# ./configure --prefix=/usr/local/gd --with-png=/usr/local/libpng \
  --with-freetype=/usr/local/freetype --with-jpeg=/usr/local/jpeg
# make
# make install
```

Compilación e instalación de PHP 4.3.2

Ejemplo para la instalación en el directorio '/usr/local/php'

```
# cd /usr/src
# tar zxvf /home/install/php-4.3.2.tar.gz
# cd /usr/src/php-4.3.2
# mkdir /usr/local/php
# make clean
# ./configure --prefix=/usr/local/php --with-apxs2=/usr/local/httpd/bin/apxs \
  --with-xmlrpc --with-mysql=/usr/local/mysql --with-gd=/usr/local/gd \
  --with-jpeg-dir=/usr/local/jpeg --with-png-dir=/usr/local/libpng \
  --with-zlib-dir=/usr/local/zlib --with-freetype-dir=/usr/local/freetype
```

make
(¡OJO! hay que parar Apache antes de continuar)
make install

En el fichero de configuración de Apache: /usr/local/httpd/conf/httpd.conf añadimos las líneas:

LoadModule php4_module modules/libphp4.so
AddType application/x-httpd-php .php .phtml
AcceptPathInfo On (para las librerias xcs de xml-rpc)

(Ya podemos arrancar Apache)

7.5 PHP

Explicamos someramente qué es el PHP y lo comparamos a otros lenguajes para el desarrollo de Webs dinámicas

PHP es uno de los lenguajes de lado servidor más extendidos en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente creciente que ha tenido una gran aceptación en la comunidad de webmasters debido sobre todo a la potencia y simplicidad que lo caracterizan.

PHP nos permite embeber sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados íntegramente en un lenguaje distinto al HTML. Por otra parte, y es aquí donde reside su mayor interés con respecto a los lenguajes pensados para los CGI, PHP ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana, sin complicaciones.

Podríamos efectuar la quizás odiosa comparación de decir que PHP y ASP son lenguajes parecidos en cuanto a potencia y dificultad si bien su

sintaxis puede diferir sensiblemente. Algunas diferencias principales pueden, no obstante, mencionarse:

-PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones. ASP, siendo una tecnología Microsoft, esta orientado hacia sistemas Windows, especialmente NT.

-Las tareas fundamentales que puede realizar directamente el lenguaje son definidas en PHP como funciones mientras que ASP invoca más frecuentemente los objetos. Por supuesto, esto no es más que una simple cuestión de forma ya que ambos lenguajes soportan igualmente ambos procedimientos.

-ASP realiza numerosas tareas sirviéndose de componentes (objetos) que deben ser comprados (o programados) por el servidor a determinadas empresas especializadas. PHP presenta una filosofía totalmente diferente y, con un espíritu más generoso, es progresivamente construido por colaboradores desinteresados que implementan nuevas funciones en nuevas versiones del lenguaje.

Este manual va destinado a aquellos que quieren comenzar de cero el aprendizaje de este lenguaje y que buscan en él la aplicación directa a su proyecto de sitio o a la mejora de su sitio HTML. Los capítulos son extremadamente simples, sino simplistas, buscando ser accesibles a la mayoría. Ellos pueden ser complementados posteriormente con otros **artículos de mayor nivel** destinados a gente más experimentada.

La forma en la que hemos redactado este manual lo hace accesible a cualquier persona no familiarizada con la programación. Sin embargo, es posible que en determinados momentos alguien que no haya programado nunca pueda verse un poco desorientado. Nuestro consejo es el de no

querer entender todo antes de pasar al siguiente capítulo sino intentar asimilar algunos conceptos y volver atrás en cuanto una duda surja o hayamos olvidado algún detalle. Nunca viene mal leer varias veces lo mismo hasta que quede bien grabado y asimilado.

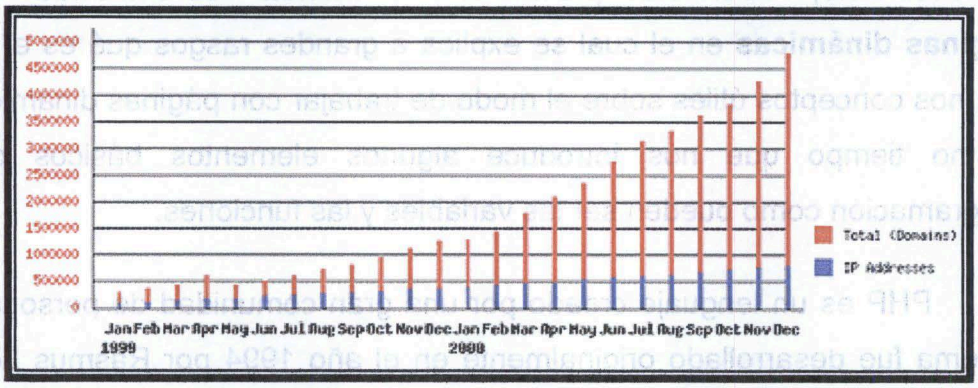
Antes de comenzar a leer este manual es altamente aconsejable, sino imprescindible, haber leído previamente el manual sobre **manual sobre páginas dinámicas** en el cual se explica a grandes rasgos qué es el PHP, algunos conceptos útiles sobre el modo de trabajar con páginas dinámicas al mismo tiempo que nos introduce algunos elementos básicos de la programación como pueden ser las variables y las funciones.

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.

La siguiente gran contribución al lenguaje se realizó a mediados del 97 cuando se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3. Actualmente PHP se encuentra en su versión 4, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su rapidez - gracias a que primero se compila y luego se ejecuta, mientras que antes se

ejecutaba mientras se interpretaba el código-, su mayor independencia del servidor web -creando versiones de PHP nativas para más plataformas- y un API más elaborado y con más funciones.

Gráfica del número de dominios y direcciones IP que utilizan PHP.



Estadística de Netcraft.

En el último año, el número de servidores que utilizan PHP se ha disparado, logrando situarse cerca de los 5 millones de sitios y 800.000 direcciones IP, lo que le ha convertido a PHP en una tecnología popular. Esto es debido, entre otras razones, a que PHP es el complemento ideal para que el tándem Linux-Apache sea compatible con la programación del lado del servidor de sitios web. Gracias a la aceptación que ha logrado, y los grandes esfuerzos realizados por una creciente comunidad de colaboradores para implementarlo de la manera más óptima, podemos asegurar que el lenguaje se convertirá en un estándar que compartirá los éxitos augurados al conjunto de sistemas desarrollados en código abierto.

Explicamos las pautas principales a seguir para incluir PHP en el código de nuestra página, la forma de introducir comentarios.

PHP se escribe dentro de la propia página web, junto con el código HTML y, como para cualquier otro tipo de lenguaje incluido en un código HTML, en PHP necesitamos especificar cuáles son las partes constitutivas del código escritas en este lenguaje. Esto se hace, como en otros casos, delimitando nuestro código por etiquetas. Podemos utilizar distintos modelos de etiquetas en función de nuestras preferencias y costumbres. Hay que tener sin embargo en cuenta que no necesariamente todas están configuradas inicialmente y que otras, como es el caso de sólo están disponibles a partir de una determinada versión (3.0.4.).

Estos modos de abrir y cerrar las etiquetas son:

```
<?      y    ?>
<%      y    %>
<?php   y    ?>
<script languaje="php">
```

Este ultimo modo está principalmente aconsejado a aquellos que tengan el valor de trabajar con Front Page ya que, usando cualquier otro tipo de etiqueta, corremos el riesgo de que la aplicación nos la borre sin más debido a que se trata de un código incomprensible para ella.

El modo de funcionamiento de una página PHP, a grandes rasgos, no difiere del clásico para una página dinámica de lado servidor: El servidor va a reconocer la extensión correspondiente a la página PHP (phtml, php, php4,...) y antes de enviarla al navegador va a encargarse de interpretar y ejecutar todo aquello que se encuentre entre las etiquetas correspondientes

al lenguaje PHP. El resto, lo enviara sin más ya que, asumirá que se trata de código HTML absolutamente comprensible por el navegador.

Otra característica general de los scripts en PHP es la forma de separar las distintas instrucciones. Para hacerlo, hay que acabar cada instrucción con un punto y coma ";". Para la ultima expresión, la que va antes del cierre de etiqueta, este formalismo no es necesario.

Incluimos también en este capitulo la sintaxis de comentarios. Un comentario, para aquellos que no lo sepan, es una frase o palabra que nosotros incluimos en el código para comprenderlo más fácilmente al volverlo a leer un tiempo después y que, por supuesto, el ordenador tiene que ignorar ya que no va dirigido a él sino a nosotros mismos. Los comentarios tienen una gran utilidad ya que es muy fácil olvidarse del funcionamiento de un script programado un tiempo atrás y resulta muy útil si queremos hacer rápidamente comprensible nuestro código a otra persona.

Pues bien, la forma de incluir estos comentarios es variable dependiendo si queremos escribir una línea o más. Veamos esto con un primer ejemplo de script:

```
<?
$mensaje="Tengo hambre!!"; //Comentario de una linea
echo $mensaje; #Este comentario también es de una linea
/*En este caso
mi comentario ocupa
varias lineas, lo ves? */
?>
```

Si usamos doble barra (//) o el símbolo # podemos introducir comentarios de una línea. Mediante /* y */ creamos comentarios multilínea. Por supuesto, nada nos impide de usar estos últimos en una sola línea.

No os preocupéis si no comprendéis el texto entre las etiquetas, todo llegará. Os adelantamos que las variables en PHP se definen anteponiendo un símbolo de dólar ($) y que la instrucción *echo* sirve para sacar en pantalla lo que hay escrito a continuación.

Recordamos que todo el texto insertado en forma de comentario es completamente ignorado por el servidor. Resulta importante acostumbrarse a dejar comentarios, es algo que se agradece con el tiempo.

La función podría ser definida como un conjunto de instrucciones que explotan ciertas variables para realizar una tarea más o menos elemental.

PHP basa su eficacia principalmente en este tipo de elemento. Una gran librería que crece constantemente, a medida que nuevas versiones van surgiendo, es complementada con las funciones de propia cosecha dando como resultado un sinfín de recursos que son aplicados por una simple llamada.

Las funciones integradas en PHP son muy fáciles de utilizar. Tan sólo hemos de realizar la llamada de la forma apropiada y especificar los parámetros y/o variables necesarios para que la función realice su tarea.

Lo que puede parecer ligeramente más complicado, pero que resulta sin lugar a dudas muy práctico, es crear nuestras propias funciones. De una forma general, podríamos crear nuestras propias funciones para conectarnos a una base de datos o crear los encabezados o etiquetas meta de un documento HTML. Para una aplicación de comercio electrónico podríamos crear por ejemplo funciones de cambio de una moneda a otra o de calculo de

los impuestos a añadir al precio de articulo. En definitiva, es interesante crear funciones para la mayoría de acciones más o menos sistemáticas que realizamos en nuestros programas.

Aquí daremos el ejemplo de creación de una función que, llamada al comienzo de nuestro script, nos crea el encabezado de nuestro documento HTML y coloca el titulo que queremos a la página:

```
<?
function hacer_encabezado($titulo)
{
$encabezado="<html>\n<head>\n\t<title>$titulo</title>\n</head>\n";
echo $encabezado;
}
?>
```

Esta función podría ser llamada al principio de todas nuestras páginas de la siguiente forma:

```
$titulo="Mi web";
hacer_encabezado($titulo);
```

De esta forma automatizamos el proceso de creación de nuestro documento. Podríamos por ejemplo incluir en la función otras variables que nos ayudasen a construir la etiquetas meta y de esta forma, con un esfuerzo mínimo, crearíamos los encabezados personalizados para cada una de nuestras páginas. De este mismo modo nos es posible crear cierres de

documento o formatos diversos para nuestros textos como si se tratase de hojas de estilo que tendrían la ventaja de ser reconocidas por todos los navegadores.

Por supuesto, la función ha de ser definida dentro del script ya que no se encuentra integrada en PHP sino que la hemos creado nosotros. Esto en realidad no pone ninguna pega ya que puede ser incluida desde un archivo en el que iremos almacenando las definiciones de las funciones que vayamos creando o recopilando.

Estos archivos en los que se guardan las funciones se llaman librerías. La forma de incluirlos en nuestro script es a partir de la instrucción *require* o *include*:

```
require("libreria.php") o include("libreria.php")
```

En resumen, la cosa quedaría así:

Tendríamos un archivo libreria.php como sigue

```
<?
//función de encabezado y colocación del titulo
Function hacer_encabezado($titulo)
{
$encabezado="<html>\n<head>\n\t<title>$titulo</title>\n</head>\n";
echo $encabezado;
}
?>
```

Por otra parte tendríamos nuestro script principal página.php (por ejemplo):

```
<?
include("libreria.php");
$titulo="Mi Web";
hacer_encabezado($titulo);
?>
<body>
El cuerpo de la página
</body>
</html>
```

Podemos meter todas las funciones que vayamos encontrando dentro de un mismo archivo pero resulta muchísimo más ventajoso ir clasificándolas en distintos archivos por temática: Funciones de conexión a bases de datos, funciones comerciales, funciones generales, etc. Esto nos ayudara a poder localizarlas antes para corregirlas o modificarlas, nos permite también cargar únicamente el tipo de función que necesitamos para el script sin recargar éste en exceso además de permitirnos utilizar un determinado tipo de librería para varios sitios webs distintos.

También puede resultar muy práctico el utilizar una nomenclatura sistemática a la hora de nombrarlas: Las funciones comerciales podrían ser llamadas com_loquesea, las de bases de datos bd_loquesea, las de tratamiento de archivos file_loquesea. Esto nos permitirá reconocerlas enseguida cuando leamos el script sin tener que recurrir a nuestra oxidada memoria para descubrir su utilidad.

No obstante, antes de lanzarnos a crear nuestra propia función, merece la pena echar un vistazo a la **documentación** para ver si dicha función ya existe o podemos aprovecharnos de alguna de las existentes para aligerar nuestro trabajo. Así, por ejemplo, existe una función llamada header que crea un encabezado HTML configurable lo cual nos evita tener que crearla nosotros mismos.

Como puede verse, la tarea del programador puede en algunos casos parecerse a la de un coleccionista. Hay que ser paciente y metódico y al final, a base de trabajo propio, intercambio y tiempo podemos llegar poseer nuestro pequeño tesoro.

Ejemplo de función

Vamos a ver un ejemplo de creación de funciones en PHP. Se trata de hacer una función que recibe un texto y lo escribe en la página con cada carácter separado por "-". Es decir, si recibe "hola" debe escribir "h-o-l-a" en la página web.

La manera de realizar esta función será recorrer el string, caracter a caracter, para imprimir cada uno de los caracteres, seguido de el signo "-". Recorreremos el string con un bucle for, desde el carater 0 hasta el número de caracteres total de la cadena.

El número de caracteres de una cadena se obtiene con la función predefinida en PHP strlen(), que recibe el string entre paréntesis y devuelve el número de los caracteres que tenga.

```
<html>
<head>
   <title>funcion 1</title>
</head>

<body>


<?
function escribe_separa($cadena){
   for ($i=0;$i<strlen($cadena);$i++){
      echo $cadena[$i];
      if ($i<strlen($cadena)-1)
         echo "-";
   }
}

escribe_separa ("hola");
echo "<p>";
escribe_separa ("Texto más largo, a ver lo que hace");
?>
</body>
</html>
```

La función que hemos creado se llama escribe_separa y recibe como parámetro la cadena que hay que escribir con el separador "-". El bucle for nos sirve para recorrer la cadena, desde el primer al último caracter. Luego, dentro del bucle, se imprime cada caracter separado del signo "-". El if que hay dentro del bucle for comprueba que el actual no sea el último caracter, porque en ese caso no habría que escribir el signo "-" (queremos conseguir "h-o-l-a" y si no estuviera el if obtendríamos "h-o-l-a-").

En el código mostrado se hacen un par de llamadas a la función para ver el resultado obtenido con diferentes cadenas como parámetro.

7.6 Funciones PDF

Introducción

Las funciones PDF en PHP pueden crear archivos PDF utilizando la biblioteca PDFlib creada por Thomas Merz.

La documentación en esta sección solamente es una descripción de las funciones de la biblioteca PDFlib y no debería considerarse una referencia exhaustiva. Se ha de consultar la documentación incluida en el código fuente de la distribución de PDFlib para una completa y detallada explicación de cada función. Proporciona muy buena descripción de las capacidades de PDFlib y contiene actualizada la documentación de todas las funciones.

Todas las funciones de PDFlib y del módulo PHP tienen nombres iguales para las funciones y parámetros. Se necesitará entender algunos de los conceptos básicos de PDF y PostScript para un eficiente uso de esta extensión. Todas las longitudes y coordenadas se mesuran en puntos PostScript. Generalmente hay 72 puntos PostScript por pulgada, pero esto depende de la resolución de salida. Se puede consultar la documentación incluida en la distribución de PDFlib para una detallada explicación del sistema de coordenadas utilizado.

Hay que tener en cuenta que la mayoría de las funciones PDF requieren un primer parámetro *pdfdoc*. En los siguientes ejemplos hay más información.

Nota: Si se está interesado en alternativas de generadores gratis de PDF que no utilizen liberías externas PDF, mirar este FAQ relacionado.

Requisitos

PDFlib está disponible para descargar en http://www.pdflib.com/products/pdflib/index.html, pero requiere la compra de una licencia para uso comercial. Se requieren las bibliotecas JPEG y TIFF para compilar esta extensión.

Compatibilidad con versiones antiguas de PDFlib

Cualquier versión de PHP después del 9 de Marzo del 2000 no soporta versiones de PDFlib anteriores a la 3.0.

PDFlib 3.0 o superior es compatible desde PHP 3.0.19 en adelante.

Confusiones con antiguas versiones de PDFlib

Desde PHP 4.0.5, la extensión PHP para PDFlib es oficialmente soportada por PDFlib GmbH. Esto significa que todas las funciones descritas en el manual de PDFlib (V3.00 o superior) son soportadas por PHP 4 con el mismo funcionamiento y parámetros. Sólo los valores devueltos pueden variar en el manual PDFlib, ya que PHP adoptó la convención de devolver **FALSE**. Por razones de compatibilidad, PDFlib aún soporta las antiguas funciones, pero deberían reemplazarlas en sus nuevas versiones. PDFlib GmbH no dará soporte a cualquier problema causado por el uso de estas funciones obsoletas.

Tabla 1. Funciones obsoletas y sus reemplazos.

Antigua función	Reemplazo
pdf_put_image()	Ya no se necesita.
pdf_execute_image()	Ya no se necesita.
pdf_get_annotation()	pdf_get_bookmark() utilizando los mismos parámetros.
pdf_get_font()	pdf_get_value() pasando *"font"* como segundo parámetro.
pdf_get_fontsize()	pdf_get_value() pasando *"fontsize"* como segundo parámetro.
pdf_get_fontname()	pdf_get_parameter() pasando *"fontname"* como segundo parámetro.
pdf_set_info_creator()	pdf_set_info() pasando *"Creator"* como segundo parámetro.
pdf_set_info_title()	pdf_set_info() pasando *"Title"* como segundo parámetro.
pdf_set_info_subject()	pdf_set_info() pasando *"Subject"* como segundo parámetro.
pdf_set_info_author()	pdf_set_info() pasando *"Author"* como segundo parámetro.
pdf_set_info_keywords()	pdf_set_info() pasando *"Keywords"* como segundo parámetro.

Antigua función	Reemplazo
pdf_set_leading()	pdf_set_value() pasando *"leading"* como segundo parámetro.
pdf_set_text_rendering()	pdf_set_value() pasando *"textrendering"* como segundo parámetro.
pdf_set_text_rise()	pdf_set_value() pasando *"textrise"* como segundo parámetro.
pdf_set_horiz_scaling()	pdf_set_value() pasando *"horizscaling"* como segundo parámetro.
pdf_set_text_matrix()	Ya no se necesita.
pdf_set_char_spacing()	pdf_set_value() pasando *"charspacing"* como segundo parámetro.
pdf_set_word_spacing()	pdf_set_value() pasando *"wordspacing"* como segundo parámetro.
pdf_set_transition()	pdf_set_parameter() pasando *"transition"* como segundo parámetro.
pdf_open()	pdf_new() más la subsecuente llamada de pdf_open_file()
pdf_set_font()	pdf_findfont() más la subsecuente llamada de pdf_setfont()

Antigua función	Reemplazo
pdf_set_duration()	pdf_set_value() pasando *"duration"* como segundo parámetro.
pdf_open_gif()	pdf_open_image_file() pasando *"gif"* como segundo parámetro.
pdf_open_jpeg()	pdf_open_image_file() pasando *"jpeg"* como segundo parámetro.
pdf_open_tiff()	pdf_open_image_file() pasando *"tiff"* como segundo parámetro.
pdf_open_png()	pdf_open_image_file() pasando *"png"* como segundo parámetro.
pdf_get_image_width()	pdf_get_value() pasando *"imagewidth"* como segundo parámetro y la imágen como tercer parámetro.
pdf_get_image_height()	pdf_get_value() pasando *"imageheight"* como segundo parámetro y la imágen como tercer parámetro.

Ejemplos

La mayoría de las funciones son bastante fáciles de utilizar. La parte más difícil probablemente es la creación de un primer documento PDF. El siguiente ejemplo debería ayudar para comenzar. El ejemplo crea el archivo test.pdf en una página. La página contiene el texto "Times Roman outlined" en un contorno, con fuente de 30pt. El texto también está subrayado.

Ejemplo 1. Creando un documento PDF con PDFlib

```
<?php
$pdf = pdf_new();
pdf_open_file($pdf, "test.pdf");
pdf_set_info($pdf, "Author", "Javier Tacon");
pdf_set_info($pdf, "Title", "Test for PHP wrapper of PDFlib
2.0");
pdf_set_info($pdf, "Creator", "See Author");
pdf_set_info($pdf, "Subject", "Testing");
pdf_begin_page($pdf, 595, 842);
pdf_add_outline($pdf, "Page 1");
$font = pdf_findfont($pdf, "Times New Roman", "winansi",
1);
pdf_setfont($pdf, $font, 10);
pdf_set_value($pdf, "textrendering", 1);
pdf_show_xy($pdf, "Times Roman outlined", 50, 750);
pdf_moveto($pdf, 50, 740);
pdf_lineto($pdf, 330, 740);
pdf_stroke($pdf);
pdf_end_page($pdf);
pdf_close($pdf);
pdf_delete($pdf);
echo "<A HREF=getpdf.php>finished</A>";
?>
```

El script getpdf.php simplemente devuelve el documento pdf.

Ejemplo 2. Mostrando un documento PDF precalculado

```
<?php
$len = filesize($filename);
header("Content-type: application/pdf");
header("Content-Length: $len");
header("Content-Disposition: inline; filename=foo.pdf");
readfile($filename);
?>
```

La distrubución PDFlib contiene un ejemplo más complejo para crear un reloj analógico en una página. Aquí se utiliza el método de creación en memoria de PDFlib para no tener que crear un archivo temporal. El ejemplo se ha convertido a PHP desde uno de PDFlib

Ejemplo 3. Ejemplo pdfclock de la distribución PDFlib

```
<?php
$radius = 200;
$margin = 20;
$pagecount = 10;

$pdf = pdf_new();

if (!pdf_open_file($pdf, "")) {
  echo error;
```

```
  exit;
};

pdf_set_parameter($pdf, "warning", "true");

pdf_set_info($pdf, "Creator", "pdf_clock.php");
pdf_set_info($pdf, "Author", "Uwe Steinmann");
pdf_set_info($pdf, "Title", "Analog Clock");

while ($pagecount-- > 0) {
  pdf_begin_page($pdf, 2 * ($radius + $margin), 2 * ($radius
+ $margin));

  pdf_set_parameter($pdf, "transition", "wipe");
  pdf_set_value($pdf, "duration", 0.5);

  pdf_translate($pdf, $radius + $margin, $radius + $margin);
  pdf_save($pdf);
  pdf_setrgbcolor($pdf, 0.0, 0.0, 1.0);

  /* minute strokes */
  pdf_setlinewidth($pdf, 2.0);
  for ($alpha = 0; $alpha < 360; $alpha += 6) {
    pdf_rotate($pdf, 6.0);
    pdf_moveto($pdf, $radius, 0.0);
    pdf_lineto($pdf, $radius-$margin/3, 0.0);
    pdf_stroke($pdf);
```

```
}

pdf_restore($pdf);
pdf_save($pdf);

/* 5 minute strokes */
pdf_setlinewidth($pdf, 3.0);
for ($alpha = 0; $alpha < 360; $alpha += 30) {
    pdf_rotate($pdf, 30.0);
    pdf_moveto($pdf, $radius, 0.0);
    pdf_lineto($pdf, $radius-$margin, 0.0);
    pdf_stroke($pdf);
}

$ltime = getdate();

/* draw hour hand */
pdf_save($pdf);
pdf_rotate($pdf,-(($ltime['minutes']/60.0)+$ltime['hours']-
3.0)*30.0);
pdf_moveto($pdf, -$radius/10, -$radius/20);
pdf_lineto($pdf, $radius/2, 0.0);
pdf_lineto($pdf, -$radius/10, $radius/20);
pdf_closepath($pdf);
pdf_fill($pdf);
pdf_restore($pdf);
```

```
  /* draw minute hand */
  pdf_save($pdf);
  pdf_rotate($pdf,-
((($ltime['seconds']/60.0)+$ltime['minutes']-15.0)*6.0);
  pdf_moveto($pdf, -$radius/10, -$radius/20);
  pdf_lineto($pdf, $radius * 0.8, 0.0);
  pdf_lineto($pdf, -$radius/10, $radius/20);
  pdf_closepath($pdf);
  pdf_fill($pdf);
  pdf_restore($pdf);

  /* draw second hand */
  pdf_setrgbcolor($pdf, 1.0, 0.0, 0.0);
  pdf_setlinewidth($pdf, 2);
  pdf_save($pdf);
  pdf_rotate($pdf, -(($ltime['seconds'] - 15.0) * 6.0));
  pdf_moveto($pdf, -$radius/5, 0.0);
  pdf_lineto($pdf, $radius, 0.0);
  pdf_stroke($pdf);
  pdf_restore($pdf);

  /* draw little circle at center */
  pdf_circle($pdf, 0, 0, $radius/30);
  pdf_fill($pdf);

  pdf_restore($pdf);
```

```
  pdf_end_page($pdf);

  # to see some difference
  sleep(1);
}

pdf_close($pdf);

$buf = pdf_get_buffer($pdf);
$len = strlen($buf);

header("Content-type: application/pdf");
header("Content-Length: $len");
header("Content-Disposition: inline; filename=foo.pdf");
echo $buf;

pdf_delete($pdf);
?>
```