

Introducción a Zope

Autor : Alvaro del Castillo San Félix, Desarrollador y admin de software libre.

Zope, una palabra que ya el lector posiblemente haya escuchado en alguna ocasión. Quizás incluso sepa que es una herramienta y que sirve para construir sitios web. Pero pocas veces se llega más allá en ese conocimiento, hasta ahora un tanto limitado, de lo que hay detrás de esta palabra. Y hay mucho más. Zope es una arquitectura muy completa y por momentos compleja. Sin duda se la podría agrupar dentro de la vaga categoría de servidores de aplicaciones, es decir, productos que facilitan el desarrollo y ejecución de servidores de información en una red, normalmente Internet. Intentando ir un poco más allá en esta vaga definición Zope son las iniciales de "Z Object Publishing Environment", un entorno para publicar objetos. Y es que sus orígenes parten de un diseño con el modelo de negocio de un periódico en mente tal y como se afirma en el que se convertirá en el libro de referencia de Zope que será publicado por O'Reilly y que se puede acceder ya desde el portal de Zope (<http://www.zope.org>).

En este mismo libro y en el prefacio se intenta también definir Zope, algo que incluso sus creadores lo encuentran difícil. En concreto ellos ven a Zope como un servidor de aplicaciones, con mucho énfasis en la gestión de contenidos, los portales y los servicios a medida destacando como una plataforma para el desarrollo rápido de aplicaciones ricas en contenido y funcionalidad.

Un detalle fundamental de Zope a la hora de ser considerado como plataforma de desarrollo es su licencia ZPL que es calificada por la FSF (<http://www.fsf.org>) como software libre, aunque incompatible con la GPL. esta licencia ha permitido que miles de desarrolladores puedan utilizar Zope sin ningún coste más allá de dar crédito a Digital Creations, en adelante DC, creadores de Zope. Zope dispone a su alrededor de una comunidad creciente de desarrolladores que están aumentando su funcionalidad con cientos de extensiones de la plataforma, extensiones conocidas como productos.

En este primer artículo vamos a presentar que hay detrás de Zope, su arquitectura, su instalación y un ejemplo de uso básico. Aún no te convertirás en un Zopista, pero te dejará en el camino.

La arquitectura de Zope

A lo largo de la vida del desarrollo de aplicaciones en Internet se han ido cubriendo distintas fases. Cuando ni existía el protocolo CGI el web era básicamente un sistema de publicación de documentos estático con nula interacción por parte del usuario. Era la recepción de ficheros con información visualizada a través de HTML, eso sí, con la posibilidad de enlazar unas páginas con otras gracias a los enlaces.

Llegó entonces la era de los ejecutables CGI y se abrió un mundo de posibilidades inmenso. Independiente de la plataforma, el protocolo CGI permite la ejecución de programas pasándole parámetros desde el cliente web y enviando sus resultados a través del servidor de web al cliente que solicitó su ejecución.

De esta forma y a través de los formularios HTML donde se capta la información a entregar al ejecutable CGI, las aplicaciones Internet comenzaron a tomar vida.

Poco a poco se han ido sofisticando los sistemas de ejecución a través CGI facilitando mucho la vida al desarrollador. Los lenguajes del lado del servidor, como PHP o ASP, permiten al desarrollador abstraerse de las incomodidades del protocolo CGI y el intercambio de datos, logrando que las aplicaciones basadas en el web sean más sencillas de desarrollar.

Un paso más allá en este camino hacia el desarrollo de aplicaciones web lo constituyen los servidores de aplicaciones, grupo al que pertenece Zope. Junto con el mecanismo básico de intercambio de datos con los clientes web, dotan al desarrollador de la parte servidor de la aplicación de unas infraestructuras que le facilitan el desarrollo de aplicaciones y el tiempo de su construcción.

La arquitectura de Zope se basa en una serie de mecanismos de intercambio de datos con el servidor Zope y un conjunto de herramientas de apoyo a ese servidor Zope. Lo mejor es ver el esquema que constituye esta arquitectura que se presenta en el portal de Zope (<http://www.zope.org/ZopeArchitecture>).

A lo largo de este documento iremos detallando cada uno de los sistemas de Zope. De momento de este esquema vamos a destacar los distintos clientes de intercambio de información con Zope (cliente Web, XML-RPC, WebDAV y FTP), la posibilidad que incluir un servidor de web como Apache comunicándolo a través del protocolo CGI persistente, el núcleo de Zope, conocido también como el ORB Zope donde se gestionan las peticiones sobre objetos Zope y donde se interacciona con las clases de objetos Zope y los productos, la base de datos orientada a objetos de Zope y la posibilidad de interaccionar con otros tipos de bases de datos relaciones o sistema de ficheros.

Todo son objetos

Una de las características fundamentales de Zope es que en su arquitectura todo lo que existe son objetos. Fue una decisión de diseño al intuir en el paradigma de objetos una potente forma de ver las partes de las aplicaciones web. Este tipo de decisiones de diseño son cada vez más comunes en los sistemas de programación actuales y es extraño ver nuevas arquitecturas de propósito general que hayan abrazado el modelo de objetos para sus sistemas.

Como consecuencia de esta decisión hubo que elegir un lenguaje con soporte de objetos. Además de esta característica el lenguaje a elegir debería estar presente en distintas arquitecturas (Windows, GNU/Linux etc.) y debería de facilitar el desarrollo de aplicaciones Internet, lo que básicamente requiere unas librerías de comunicaciones y soporte de HTTP/CGI potentes. De entre las alternativas existentes (Perl, Java, C, C++ ...) se eligió a Python, un lenguaje interpretado, presente en muchas plataformas, con mucha fuerza dentro de la comunidad de desarrolladores de aplicaciones Internet, que ya había mostrado su potencia en el mundo de los CGI y totalmente orientado a objetos (<http://www.python.org>).

Desde el momento en que como desarrolladores queramos extender las posibilidades de Zope será este el lenguaje a utilizar. Seguro que a muchos lectores este detalle les va a hacer plantearse el seguir con esta plataforma. Python es un lenguaje bastante nuevo y que aún tiene pocos conocedores del mismo. Pero cabe destacar que Python es un lenguaje sencillo para aquellos acostumbrados al mundo de los objetos, que al ser interpretado, facilita mucho los cambios en el código (no hay que recompilar) y que cada vez está más presente en el mundo del software libre, donde entornos como KDE y GNOME lo están eligiendo como el lenguaje para realizar sus "scripts". Esperamos haber logrado que el lector de una oportunidad a Python y si ya lo conocía, seguro que estará de acuerdo en las bondades de este excelente lenguaje de programación. Tan sólo tiene un pequeño punto negro en su licencia, que aún no es considerada compatible con la GPL, pero que se está trabajando en su resolución.

ZOB: La base de datos de objetos

Como ya hemos comentado, todos los elementos de Zope son objetos. Y estos objetos se almacenan en una base de datos orientada a objetos. La potencia que proporciona este almacenamiento persistente de los objetos es enorme, sobre todo a la hora de modificar dinámicamente y buscar datos dentro de todos los objetos que forma el sitio web. Como veremos Zope dispone de un producto conocido como ZCatalog que permite realizar búsquedas en todos los objetos del servidor.

ZOB destaca por ser casi transparente para el programador y los objetos que deban de ser persistentes requieren unos cambios mínimos (heredar de una clase Python). Da soporte a las transacciones y ello permite que casi cualquier acción que realicemos en Zope podrá ser deshecha. Destaca también su alto rendimiento, que permite alcanzar un buen rendimiento incluso con bases de datos de giga-bytes.

Zope ORB

Ya veremos en los ejemplos con los que cerraremos este artículo que los clientes cuando piden una URL al servidor Zope, lo que están haciendo en realidad es pedir al servidor Zope un objeto. En realidad se pide la ejecución de un objeto con una serie de parámetros. Si el lector es conocedor de

CORBA este concepto le recordará a como se ejecutan métodos sobre objetos remotos a través de CORBA. Cuando el servidor Zope recibe la petición para acceder a un objeto Zope, esta petición va a parar al ORB de Zope que se encarga de localizar al objeto y procesar la petición, mostrando los resultados de la misma. Este ORB es una parte crítica de la arquitectura ya que de su buen funcionamiento depende el rendimiento y la robustez de la plataforma.

Obtención e instalación de Zope

La licencia de Zope: ZPL

Debería ser el primer detalle en analizar a la hora de probar un producto nuevo. La licencia es que la que nos describe que podemos hacer y que no podemos hacer con el programa, que tipo de garantías recibimos del funcionamiento (ninguna normalmente), en cuantos puestos podemos utilizar el producto, bajo que condiciones etc.

Zope destaca por ser considerado software libre, tal y como podemos consultar en la descripción de licencias de la Free Software Foundation. De ella dice la FSF que es incompatible con la GPL pero que no hay ningún problema por utilizar software con esta licencia, aunque se recomienda no utilizarla. La licencia sigue la línea de la BSD, permitiendo el uso, distribución y modificación de Zope, aunque siempre cuidando de resaltar que DC son los creadores de Zope.

Así que puedes utilizar Zope sin ningún coste, puedes bajar su código fuente y modificarlo y puedes distribuirlo.

Obtención de Zope

Para obtener zope lo mejor es acudir al portal de Zope (<http://www.zope.org>) y desde allí obtenerlo en código fuente. La versión actual (Septiembre 2000) es la 2.2.1 y lo único que se necesita para su ejecución es tener instalado Python en el sistema, una versión mayor que 1.5.2.

Desde la página <http://www.zope.org/Products> no sólo se puede bajar Zope se puede también acceder a la descarga de cientos de productos Zope que amplían la funcionalidad de la plataforma. Pero de momento lo más recomendable es bajar Zope y pasar a compilarlo e instalarlo.

Instalación de Zope

Zope puede ser utilizado en diversas plataforma pero nosotros nos vamos a centrar en GNU/Linux y sobretodo, en Debian. Aunque hay que destacar que debido a la portabilidad los cambios son mínimos entre distintas arquitecturas por lo que lo que aquí comentemos va a valer prácticamente sin modificación para el caso concreto del lector.

Si estamos en una plataforma Debian lo tenemos muy fácil. Basta con ejecutar:

```
apt-get install zope
```

Con Debian hay que tener cuidado ya que lo normal es que el servidor Zope esté configurado para escuchar en el puerto 8080 mientras que en Debian el puerto es el 9673. En el caso de Red Hat la instalación consistirá en la descarga de los paquetes de Zope (<http://starship.python.net/crew/jrush/Zope/>) y su instalación con el comando "rpm" (rpm -ihv paquete_a_instalar).

Si nos hemos bajado las fuentes, después de desempaquetarlas nos metemos en el directorio con las fuentes de Zope y ejecutamos el comando "./install". Este se encargará de compilar algunas clases Python para una ejecución más veloz. En este mismo directorio tenemos los programas "start" y "stop" con los que arrancaremos Zope o lo pararemos.

La primera incursión en Zope

Una vez instalado Zope, necesitaremos arrancar el servicio. En Conectiva Linux y sistemas tipo Red Hat lo lograríamos ejecutando:

```
/etc/rc.d/init.d/zope start
```

En el caso de que te hayas bajado las fuentes es tan sencillo como desde dentro del directorio con las fuentes de Zope ejecutar `./start`.

Ya tenemos a Zope corriendo. Ha llegado el momento de conectarnos al sistema para lo que necesitaremos alguno de los clientes que se mencionaban en la arquitectura. Sin duda vamos a comenzar con un cliente web, que todos tenemos a mano. El primer problema es saber en que puerto está escuchando Zope. En Red Hat probablemente será el 8080 y en Debian el 9673. *Si usamos un Linux como el primero, ejecutaríamos:*

```
http://localhost:8080
```

Si todo ha ido bien deberías *haber accedido a la página principal de acceso de Zope, una pantalla donde se nos da la bienvenida a nuestro flamante servidor Zope que acabamos de instalar.*

Esta página es la misma que la que se observa si se accede a la URL `http://localhost:9673/index_html`. Aunque el lector tenga la sensación de que está accediendo a un fichero, no es cierto. A lo que se accede en realidad es a un objeto Zope cuyos contenidos se visualizan tal y como muestra la figura. Es importante que este concepto vaya calando en el lector ya que es básico para llegar e entender la arquitectura de Zope. Para acceder a la gestión del sitio web, desde donde podremos controlar toda la arquitectura Zope y sus contenidos, debemos de acudir a la pantalla de gestión, a través de la URL:

```
http://localhost:8080/manage
```

Como no podía ser de otra forma esta página se encuentra protegida por contraseña. El primer usuario que se crea para acceder a la gestión de Zope, el superusuario del sistema, es aquél que esta guardado en el fichero "access" de Zope. En el caso de Debian este fichero se encuentra en `/var/lib/zope/access` y en el caso de instalarlo desde las fuentes el fichero está en el mismo directorio donde están las fuentes de Zope. Este usuario se crea en el caso de la instalación desde las fuentes cuando se compilan los módulos Python, y se nos informa de cual es el usuario y clave. Con el programa `zpasswd.py` podemos modificar este usuario. En el caso de Debian y como superusuario ejecutamos:

```
/usr/lib/zope/zpasswd.py -u root -p clave /var/lib/zope/access
```

Este usuario sólo debe usarse para crear nuevos usuarios y gestionar el sistema, no para crear nuevos contenidos. Zope dispone de un sistema de control de acceso muy avanzado, como ya veremos, y nos va a permitir un control muy preciso de la forma de acceder y modificar las distintas partes de un web Zope.

Si todo a ido bien, una vez introducida la clave de entrada estaremos dentro del servidor Zope, delante de una interfaz de uso.

Desde esta interfaz es desde donde se controla todos los sitios web que viven en este servidor Zope, desde donde se pueden crear y destruir usuarios, construir webs a partir de productos, modificar el contenido de los documentos, reiniciar el servidor Zope, controlar la base de datos ... en fin, administrar todo el sistema. Como botón de muestra de lo que se puede hacer el lector puede pulsar sobre el objeto "index_html" y observar como se despliega una página desde la que puede modificar su contenido.

El portal de una empresa

Con la llegada de Internet nos hemos encontrado con la gran ola de empresas de todo tipo que se lanzan a plantar su presencia en Internet. Los beneficios que proporciona una presencia en la red, como es por ejemplo el acceso a una potencial una cliente mundial, no han dejado impasibles a las personas que cuidan de estar siempre a la última. A día de hoy no sólo las empresas son las que están desembarcando en la red: raro es el caso de una organización que tenga algo que decir que no se haya planteado el estar en Internet.

En nuestro caso nos vamos a plantear como ejemplo la realización de un portal para una empresa. Es una labor cada vez más demandada por lo que puede ser un ejemplo que abra puertas al lector en su mundo profesional. Es un ejemplo bastante completo, que encierra muchas de las problemáticas de como llegar a mostrar una imagen dentro de la red por parte de una organización. Y es un ejemplo con el que muchos de los lectores tendrán que enfrentarse en su futuro profesional. Lo primero que hay que hacer antes incluso de plantearse con que herramienta se va a desarrollar el web es definir que se quiere lograr. Es decir, que contenidos se quieren poner en la red, para quién y como se van a organizar. Nos encontramos en este momento en el famoso campo de los arquitectos de la información, un perfil que a pesar de que ya existía dentro del mundo del diseño tradicional, se ha potenciado de manera muy significativa con la aparición de Internet. Todas las compañías quieren poder expresarse de una forma natural, organizada y que proporcione una imagen adecuada. Una ardua labor que debe coordinar a distintos departamentos de la compañía: comunicación, marketing, comerciales, gestión, técnico ...

En nuestro ejemplo nos vamos a centrar en el diseño del portal interno de una compañía, cuyo objetivo es proporcionar a los distintos departamentos de la misma libertad para informar a los demás de sus actuaciones, con el objetivo de mejorar la coordinación y que aumentar el flujo de información, piedras angulares para el buen funcionamiento de una compañía.

Dentro de la amplia documentación de Zopei, y para este artículo en concreto, nos va a ser útil la "Guía para el Gestor de Contenidos Zope". Esta guía se puede obtener del web de Zope, en la zona de documentación. En las referencias puedes encontrar los detalles exactos de como descargarla para poderla leer tranquilamente.

Organización de la información

Lo primero que necesitamos es tener claro como se van a presentar los datos de la compañía. Normalmente nosotros vamos a ser parte del departamento técnico y nuestra función será proporcionar los medios técnicos para plasmar la estructura de información que nos proporcione alguna persona con un conocimiento global del funcionamiento de la empresa.

Supongamos que se ha decidido organizar el web de la compañía en cuatro zonas diferenciadas por departamentos: comunicación, marketing, gestión y técnico. Cada uno de los departamentos será responsable de la gestión de contenidos de su zona, y existirá una zona pública en cada departamento y una zona privada con información interna. Desde una página de inicio común se podrá ir a la zona de cada departamento. Todas las páginas de los departamentos deberán de compartir una imagen común, cambiando el título de la página y los contenidos, pero manteniendo una imagen homogénea.

Con estos datos ya nos es suficiente para comenzar a trabajar en el proyecto de la empresa y mostrar la potencia y flexibilidad que podemos ofrecer a nuestra compañía por medio de Zope. Ha llegado el momento de dejar claro la calidad del departamento técnico de la empresa y la de sus herramientas, basadas en software libre (siempre es bueno auto motivarse cuando comienza un proyecto :-)).

Primeros pasos en la creación del web

Nuestra compañía se llama Rayo de Luz por lo que parece lógico que el primer paso que

debemos de dar es crear una nueva carpeta (folder) para guardar en él el contenido del web que vamos a crear. Para ello, tal y como mostramos en el último artículo, accedemos a la pantalla de gestión de Zope y pasamos a dar este primer paso utilizando la interfaz web de Zope para la publicación de sitios web.

Ya tenemos la carpeta para el web creada y hemos dejado seleccionadas las opciones de crear una interface pública, para que todo el mundo pueda acceder a dicho directorio, y de que se cree una carpeta para los usuarios de esta nueva zona. Gracias a este carpeta de usuarios vamos a poder definir usuarios y grupos dentro de la zona del web de la compañía, usuarios que no serán visibles para otras publicaciones que hagamos con este servidor Zope. Iremos poco a poco viendo la potencia de control de acceso y operación de que dispone Zope, una de las características que nos pueden hacer terminar de decantarnos por este gran producto.

La zona de los departamentos

Desde la página principal de nuestra compañía lo mejor que podemos hacer es poner enlaces a los distintos departamentos que forman la compañía, un logo de la misma para ir creando la imagen corporativa y mostrarla a nuestros empleados, los visitantes del sitio que estamos construyendo. Para cumplir este objetivo nos vamos a la consola desde la que gestionamos zope (<http://localhost:9673/manage>) y seleccionamos la carpeta "rayodeluz".

Quizá es un buen momento para recordar al lector que todo lo que está viendo, a pesar de estar organizado como si fueran ficheros de directorios, son objetos, no son ficheros. La carpeta "rayodeluz" es un objeto contenedor con una serie de propiedades. Dentro de este objeto, que es muy similar a una carpeta, nos encontramos con muchos otros objetos, que también pueden ser contenedores, permitiendo la aparición de estructuras en árbol, una abstracción a la que estamos todos acostumbrados gracias a los sistemas de ficheros.

Estamos dentro de la carpeta (objeto) "rayodeluz", que acabamos de crear en el anterior apartado, y cuyo único objeto es un DTML Document llamado "index_html", que se crea de forma automática cada vez que creamos una carpeta. Cuando a través de una URL accedemos a la carpeta, el ORB de Zope de forma automática nos devuelve el objeto "index_html". De nuevo, es una abstracción muy similar a la que hacen los servidores de web como Apache, donde el fichero índice de los directorios se suele llamar "index.html". El objeto "DTML Document" llamado "index_html" podemos interpretarlo como un fichero que está dentro de la carpeta "rayodeluz", aunque el lector tiene que tener en cuenta que en realidad es un objeto que permite hacer con él bastantes más cosas que con un simple fichero. Esta insistencia en la idea de que todo son objetos es fundamental para lograr comprender la arquitectura con toda su potencia, y no quedarnos en una interpretación demasiado simplista de lo que es Zope. Vamos a editar el objeto "index_html" para ver como podemos modificar su contenido e incluir el índice a los departamentos de la compañía y el logo.

El contenido por defecto de "index_html" hace referencia a la una serie de variables DTML que definen la cabecera y pie de la página, y el título y el nombre del documento actual. Todas estas variables se buscan dentro del contenedor en el que esté el objeto "index_html", es este caso dentro de "rayodeluz". En nuestro caso todas estas variables no están definidas en este objeto por lo que ¿de dónde las saca Zope?.

Aquí ha salido a relucir uno de los mecanismos más potentes de Zope, mecanismo conocido como adquisición. Por este proceso, si una variable no se encuentra en el contenedor actual se busca en los contenedores superiores, hasta dar con su valor. De esta forma se pueden centralizar valores que son luego adquiridos por toda una jerarquía de objetos. Veremos que no sólo afecta este mecanismo a la variables, si no que se aplica en otras ocasiones, convirtiendo a la adquisición en uno de los mecanismos estrella de Zope. En nuestro caso la cabecera y pie por defecto, los que se incluyen en la carpeta raíz de Zope, no nos valen, por lo que nos vamos a crear dos objetos Documento DTML cuyos identificadores sean "standard_html_header" y "standard_html_footer" que serán utilizados por todas las páginas HTML del web de nuestra compañía. Vamos con ello.

Ya tenemos creado el nuevo objeto que servirá de cabecera para todas las páginas que creemos como parte del web de la compañía. En general, todas estas páginas serán documentos DTML

aunque cuando sea necesaria más interactividad, por ejemplo a través del intercambio de información entre el cliente web y Zope, utilizaremos otras alternativas. El contenido de esta nueva cabecera es el siguiente.

Como vemos, utilizamos la variable "var title_or_id" para que la todas las páginas tengan como título HTML el título que se le de al objeto DTML en el que residen. Tras ello definimos un BODY HTML con un fondo blanco e incluimos una imagen como logo. Pero aquí tenemos un problema. Este imagen es un fichero real, no un objeto Zope y sin embargo, sólo podemos hacer referencia a objetos que residan dentro de Zope para construir la página.

Para resolver este inconveniente tenemos la posibilidad de importar imágenes en Zope, que pasarán a ser objetos disponibles para todos los demás objetos de Zope. Para añadir esta imagen lo primero que hacemos es crearnos una carpeta para almacenar todas las imágenes y tras ello, nos metemos en esa carpeta, a la que hemos llamado "img", y añadimos un nuevo objeto de tipo "File". Como identificador del objeto tenemos que dar el nombre que utilizamos para referirnos a esta imagen dentro de las páginas de Zope, y el título podemos elegir el que queramos. Pulsando el botón de "Browse" podemos navegar por nuestro sistema de ficheros local a la búsqueda de la imagen. Una vez localizada pulsamos sobre el botón "Add" y ya tenemos dentro de Zope la imagen disponible. Podemos ver que una vez dentro de Zope, este automáticamente le asigna un tipo MIME y podremos visualizar la imagen accediendo a la opción de "View" presente en todos los objetos. Si ahora probamos a visualizar la cabecera veremos que ya se muestra de forma correcta la imagen.

A lo largo de todo el apartado anterior hemos sentado las bases de la construcción de nuestro web con Zope. Si el lector ha ido leyendo con atención, a estas alturas ya estará en disposición de avanzar muy rápido con la creación del web con Zope. El siguiente paso es modificar el pie que va a aparecer en todas las páginas que cuelguen de la carpeta "rayodeluz", algo posible gracias al mecanismo de adquisición. Esta modificación es totalmente paralela a la de la cabecera. Hay que crear un nuevo documento DTML dentro de "rayodeluz" con el nombre "standard_html_footer". Este HTML será incluido de forma automática en todos los documentos DTML nuevos, y estos documentos son los que constituyen las páginas web del sitio.

Bueno, pues ya hemos logrado construir las cabeceras y pies de página para todas las páginas presentes y futuras del web. Sólo nos queda meter dentro de "index_html" los enlaces a las distintas secciones de la compañía. Para ello nada mejor que una simple lista HTML con enlaces a los distintos directorios de cada departamento. Mostremos de nuevo como se inserta este HTML. A partir de este momento ya daremos por supuesto que el lector se maneja con la interfaz de Zope y sabe como editar los objetos y su contenido.

Este contenido nos genera al fin la página principal de la empresa. Es un poco sencilla pero no olvidemos que nosotros somos simples técnicos. Ya se encargará nuestro departamento de diseño de llevarse las manos a la cabeza cuando vea la página y hacer una maqueta que deja boquiabiertos a los navegantes de las páginas web de la compañía. Y ese esquema lo podremos introducir dentro de la cabecera estándar y pasar a ser utilizado de forma automática por todas las páginas que cuelguen de "rayoluz". Un trabajo de días que nosotros podremos hacer en unas pocas horas y podernos dedicar a descansar en el demás tiempo.

Control de acceso

Bueno, ya llegado el momento de comenzar a crear las páginas del departamento técnico. En ellas existirá información pública para todos los empleados de la empresa como los pasos para configurar el acceso a Internet de los computadores, las instrucciones básicas de mantenimiento de los equipos, las normas básicas de navegación por Internet y uso del correo electrónico y los correos y teléfonos de contacto de las personas del departamento. Pero no toda la información que va a estar dentro de nuestro departamento debe ser pública. Los datos sobre las claves de acceso a las máquinas como administrador, la gestión de IP de los equipos y toda la biblioteca de documentos técnicos, así como un foro de discusión interno, serán partes privadas.

Creamos una nueva carpeta para incluir las páginas del departamento técnico, y como identificador utilizamos "tecnico" que es el nombre que dimos como enlace en la página principal. Dentro de esta

carpeta modificamos el objeto "index_html" para reflejar la información del departamento y ponemos enlaces a distintas páginas donde se amplía la información. Una vez hecho esto, vamos a crear una carpeta llamada "interno" donde vamos a incluir toda la información que debe ser sólo accedida por las personas de nuestro departamento.

El acceso a esta carpeta llamada "interno" queremos que sea restringido. Para ello nos vamos a dicha carpeta y seleccionamos la solapa "Security" donde se definen las políticas de seguridad de acceso a los distintos objetos de esta carpeta. En general estos permisos se "adquieren" de la carpeta superior. De nuevo sale a la palestra el mecanismo de adquisición: si algo no existe en el contenedor actual, se viaja a los contenedores superiores en la jerarquía buscándolo. De esta forma podemos tener centralizada la seguridad de todo el sitio en la carpeta raíz. Veamos los permisos que vienen por defecto en la carpeta raíz, y que son los que heredan todos los nuevos contenedores que vayamos creando.

En la pantalla de permisos, encontramos, en la primera columna se refiere al usuario anónimo. Si aparece seleccionada esta opción, un usuario sin autenticarse será capaz de realizar esa operación. Por ejemplo, tenemos seleccionado "View" por lo que cualquier usuario podrá ver todas las páginas DTML por defecto. También podrá utilizar el objeto "Z Search Catalog" o utilizar métodos para acceder a las bases de datos. Los permisos que aquí aparecen se pueden ampliar y se pueden crear nuevos roles, a parte del de usuarios anónimo, el de administrador y el de dueño del recurso. Vemos que por ejemplo el usuario anónimo no tiene acceso a las pantallas de gestión de Zope, algo normal ya que desde allí se puede modificar fácilmente todo el web.

Nuestro objetivo es que para entrar en el directorio "interno" del la parte de departamento técnico, sea necesario autenticarse en el sistema. Para ello nos vamos a dicha carpeta, accedemos a la solapa de seguridad y modificamos la operación de "View" para que no se adquiera. De esta forma sólo podrán ejecutar esa operación los usuarios que tengan el rol de administradores.

Para probar que ahora se nos pide que nos autentiquemos, podemos apagar el navegador por completo e intentar acceder a dicho directorio. Si el lector ha etido una clave de administrador en algún momento, y esperamos que lo haya hecho para poder seguir el ejemplo, debe apagar el navegador para que deje de tener efecto esta clave para Zope

¿De dónde consulta Zope este usuario que se nos pide? Recordemos que a la hora de crear una carpeta teníamos al opción de crear una carpeta de usuarios asociada a ella. El primer sitio en el que Zope va a buscar el usuario que se introduzca en la caja de autenticación va a ser dentro de esta carpeta de usuarios "user_folder". Cuando se crea un usuario, se le asigna un rol. Es este rol el que define que puede y que no puede hacer un usuario. En este caso, para acceder a esta carpeta protegida, es necesario que el usuario tenga el rol de "manager". Vamos a crear un usuarios con estos permisos. Queremos que este usuario sólo tenga el rol de "manager" para esta carpeta en concreto. Por ello lo creamos en el "user_folder" de esta carpeta. Este usuario será "manager" en la carpeta actual y todas las que cuelguen de él, pero no lo será en carpetas superiores. Con este sistema tenemos una gran potencia a la hora de definir permisos a regiones concretas de tal forma que haya distintos administradores para cada zona del sitio.

Este nuevo usuario tendrá el rol de "manager" por lo que podrá acceder a la zona protegida y, más aún, podrá acceder a la interfaz de gestión y modificar el contenido de su zona desde la interafaz Zope. Iremos viendo a lo largo del curso la enormes posibilidades que ofrecen estos mecanismos de gestión de usuarios que harán triviales las labores más complejas de control de acceso.

Los Productos Zope

Antes de comenzar con los detalles de que es un producto Zope, dónde encontrarlos y para que sirven, vamos a ver de que Productos Zope disponemos en nuestra instalación estandar. En este caso me voy a referir a la configuración que viene con Debian Potato, aunque es de esperar que sea idéntica a la que un usuario cualquier de Zope se puede encontrar en la versión 2.1.6. La versión que tiene de Zope la puede consultar el usuario en la pantalla de gestión de Zope dentro de la carpeta "Control Panel".

De entre toda esta lista de objetos hay un grupo de ellos que ya hemos utilizado como son DTML Document o Folder, entre otros. Vamos a ir describiendo para que sirve cada uno de ellos, aunque no sin recordar antes que en Zope todo son objetos: las carpetas, los ficheros, las imágenes ... todo. Lo que ocurre es que en la interfaz de Zope les han dado un significado familiar para que nos sea más sencillo su uso. Pero es importante recordar este concepto, y que todos estos objetos residen en la base de datos orientada a objetos de Zope. Lo que se nos muestra en esta lista son las clases de objetos que tenemos disponibles, que luego aumentaremos, y nosotros elegimos una clase de objeto y creamos una instancia, si hablamos con la terminología de la programación orientada a objetos. Puntualizado este concepto vamos con esa descripción de los objetos Zope que aparecen.

DTML Document: este objeto sirve para añadir nuevos documentos HTML a nuestro sitio. Dentro de esas páginas HTML se puede incluir código DTML, tal y como hicimos a la hora de construir las páginas de la empresa Rayo de Luz.

DTML Method: este objeto es muy similar a DTML Document, pero está más orientado a la creación de métodos que interactúen con Zope en función de las acciones y los datos que proporcione el cliente que accede al sitio.

External Method: nos permite definir funciones en Python y crear objetos asociadas a ellas. Esos objetos pueden ser utilizados luego desde Zope para invocar a dichas funciones.

File: permite crear objetos a partir de ficheros disponibles fuera de Zope. No olvidemos que todos los objetos Zope viven dentro de la base de datos de objetos de Zope, es decir, Zope no puede acceder a ficheros locales de disco. Gracias a este objeto podemos importar dentro de la base de datos de Zope ficheros. Una vez dentro de Zope ya podremos acceder a ellos desde otros objetos Zope.

Folder: la forma de organizar los objetos Zope dentro del sitio se realiza también utilizando la abstracción de carpetas en la que se clasifican los objetos. Los objetos folder (carpeta) nos permiten guardar dentro de ellos a otros objetos, para organizar la estructura de objetos como si fueran ficheros y directorios. Asociados a los objetos Folder tenemos una carpeta de usuarios, para controlar el acceso a esas carpetas, y el la posibilidad de crear de forma automática un objeto "index_html", de tipo DTML Document, que se mostrará cuando alguien intente acceder al objeto Folder.

Image: este objeto nos permite importar a Zope imágenes. Una vez dentro de Zope, ya pueden ser utilizadas dentro de los documentos siguiendo el método tradicional de HTML.

Mail Host: con este objeto creamos una puerta de acceso desde Zope a un servidor de correo a través del cual podemos enviar correo. Una vez configurado de forma correcta el Mail Host, se podrá utilizar la interfaz de Zope para envío de correo.

User Folder: este objeto es la base para el control de acceso a los recursos de Zope, es decir, controlar el acceso y las funciones que se pueden llevar a cabo dentro de Zope. Como ya hemos comentado, Zope destaca por su potente arquitectura de control de recursos y usuarios.

Version: este objeto pone al descubierto otra de las potencias de Zope: el control de versiones que lleva. Todos los objetos dentro de Zope tienen controlado las modificaciones que se hacen sobre ello. De este modo, podemos saber que se cambió y cuando, y poder deshacer los cambios que se llevaron a cabo. La potencia del "Undo" de Zope es realmente impresionante permitiendo deshacer casi cualquier operación que se haya producido sobre el sistema. Con este objeto, podemos crear una nueva versión del sitio, activarla y comenzar a trabajar sobre ella. Esta versión y todo su contenido son privados y no se harán públicos hasta que no decidamos que todo está acabado y probado. De esta forma los desarrolladores pueden trabajar tranquilamente con la propia versión en producción, las personas autorizadas pueden ir viendo como avanza el desarrollo y su publicación será tan sencilla como decirle al objeto Version que se ha terminado el desarrollo y las pruebas de esa Version y que pase a público todo el trabajo. Realmente potente, ¿verdad? Hablaremos más sobre ello.

Z Gadfly Database Connection: para poder acceder desde Zope a las bases de datos externas, se crean lo que se llama adaptadores de Zope. Estos adaptadores exportan hacia Zope una interfaz común a los objetos, es decir, que proporcionan una forma de acceder común a cualquier base de datos para Zope. Esto permite que una aplicación que trabaja sobre MySQL pueda pasar a trabajar sobre Postgres con un simple cambio a la hora de elegir el adaptador. Este

objeto, Z Gadfly Database Connection, es el que nos permite crear una conexión a la base de datos Gadfly. Esta base de datos se distribuye con licencia similar a la BSD. Este adaptador se incluye por defecto con Zope para que se pueda practicar la forma de acceder a las bases de datos desde Zope, uno de los puntos más fuertes de Zope. Para acceder a una base de datos desde Zope, se activa la conexión a la base de datos a través de un manejador adecuado, en este caso Z Gadfly Database, y se pasan a ejecutar los llamados ZSQL Methods, que pasamos a detallar a continuación.

Z SQL Method: estos objetos nos permiten crear peticiones SQL contra las bases de datos que tengan adaptadores en Zope. Existen adaptadores para las principales bases de datos (MySQL, Postgres, Oracle ...) y los Z SQL Methods son independientes de la base de datos, tan sólo necesitan una conexión por el que poder ejecutarse. A la hora de crearse la petición SQL, se selecciona a través de que conexión se quiere llevar a cabo la petición.

Z Search Interfaces: con este objeto crearemos interfaces de búsqueda en bases de datos de una forma muy sencilla. Lo usual es seleccionar un Z SQL Method a partir del cual se crea un formulario HTML y se prepara una hoja de resultado. Este objeto nos permite ahorrar mucho tiempo de desarrollo y facilita en gran medida el diseño de interfaces de búsqueda en bases de datos.

Z Catalog: en Zope todo son objetos que se almacenan en una base de datos. Esto es fundamental a la hora de permitir hacer búsquedas complejas y muy potentes de recursos dentro de un web Zope. Al tener registrados todos los objetos en la base de datos, se simplifica mucho el realizar interfaces de búsquedas y se abren puertas impensables en entornos tradicionales de publicación web, donde las páginas HTML son ficheros almacenados en disco. Con los objetos Z Catalog podemos catalogar, similar a obtener índices a la documentación que consideremos importante. Los catálogos en combinación con Z Search Interfaces nos proporcionan un mecanismo muy potente para hacer búsquedas dentro de todo el contenido de nuestro sitio web, como vamos a poder ver en lo que resta de este artículo.

Catalogar y Buscar en un sitio Zope

Después de analizar todos los objetos Zope que tenemos por defecto dentro de una instalación estandar de Zope nada mejor que mostrar un uso práctico de dos de ellos para terminar de familiarizarnos con su uso.

El objetivo es crear una interfaz de búsqueda dentro de todo el sitio Rayo de Luz, que creamos en la anterior entrega del curso. Era un sitio muy sencillo con un grupo de carpetas que contenían la información de cada departamento. Esta información es previsible que vaya creciendo con el tiempo y necesitamos poder acceder a ella de una forma sencilla y flexible. Y como vas a ver, esto en Zope es realmente sencillo y potente.

Catalogar la información

Durante el trascurso de la anterior entrega del curso creamos el sitio de la empresa "Rayo de Luz". Era un sitio muy sencillo con una página principal desde la que se podía acceder a los distintos departamentos de la compañía.

Nuestro objetivo en esta ocasión es añadir un nuevo enlace a esta página a una interfaz de búsqueda dentro de todo el sitio. Como ya hemos comentado, los sistemas de información de las compañías, si han sido bien implantados, tienen una gran tendencia de crecimiento y los sistemas de búsqueda se vuelven fundamentales para poder localizar con facilidad información.

Dentro de Zope, todos los sitios no son más que un conjunto de objetos en una base de datos, por lo que las búsquedas se resumen en poder acceder a la base de datos Zope por distintos campos y hacer búsquedas utilizándolos. El lector familiarizado con las bases de datos entenderá que el que todo este en base de datos simplifica mucho el proceso de búsqueda, además de permitir hacer búsquedas bastante complejas.

Para poder buscar dentro del contenido de Rayo de Luz lo primero es crear un catálogo con todos los objetos sobre los que queremos buscar. Para ello, y desde la intefaz de gestión, nos vamos a la

carpeta rayodeluz y añadimos un nuevo objeto ZCatalog.

Para crear un objeto de tipo catálogo tan sólo hay que proporcionar dos datos. Cual es el identificador del catálogo, que es la palabra que usarán otros objetos Zope para poder acceder a él, y cual es el título del catálogo.

Bien, una vez que tenemos creado el catálogo hay que ir a configurarlo y crearlo. El objeto Catálogo es realmente potente y nos permite configurarlo para poder especializarlo en determinadas tareas.

Lo primero es añadir objetos al catálogo. Inicialmente los catálogos se crean vacíos y hay que ir a la solapa "Find Items to ZCatalog" (Encontrar Objetos para ZCatalog) para crear un catálogo inicial sobre el que se realizarán las búsquedas. Dentro de dicha solapa se especifican los criterios a seguir para encontrar los objetos a catalogar, es decir, los objetos para los que se obtendrán una serie de índices que formarán el catálogo y que nos permitirán hacer búsquedas rápidas sobre ellos.

Las opciones de las que disponemos son múltiples. Podemos catalogar sólo un tipo de objetos, utilizando los identificadores de los objetos, su contenido, su fecha de modificación y los permisos que tienen los objetos. En nuestro caso, vamos a irnos al caso más sencillo y vamos a pedirle que nos catalogue todos los objetos, algo que se debe evaluar en sitios con mucha información, pero que para nuestro caso es perfectamente válido ya que aún hay poca información y además, sabemos que Zope escala muy bien por lo que podemos estar tranquilos de cara al futuro. Sin modificar pues ninguna opción pulsamos el botón de "Find" (Buscar) con lo que el objeto ZCatalog pasa a buscar en toda la base de datos Zope todos los objetos que estén dentro de la carpeta "rayodeluz", los indexa y crea el catálogo.

Ha encontrado un total de 13 objetos con los que ha creado 13 registros dentro del catálogo. ¿Qué índices ha creado para buscar? El objeto Catalog dispone de la posibilidad de definición de los índices que se generarán bajo la solapa "Indexes" de la interfaz de gestión del objeto Catálogo, pero no vamos a entrar en esos detalles ya que nos alejarían demasiado de este ejemplo básico de uso de catálogos y búsquedas en Zope.

Bueno, ahora ya tenemos el catálogo creado pero poco podemos hacer con él. Necesitamos crear una interfaz que nos permita hacer búsquedas sobre el catálogo, de cara a que nuestros usuarios puedan fácilmente localizar la información que necesitan. Ha llegado el momento de echar mano de Z Search Interface, el objeto con el que crear interfaces de búsqueda es coser y cantar.

Búsquedas en el catálogo

Utilizamos el procedimiento habitual para añadir un nuevo objeto Z Search Interface a la carpeta "rayodeluz" tras lo que se nos piden una serie de datos para poder crear la interfaz de búsquedas. Este objeto crea dos partes. Una es la interfaz de búsqueda y la otra es la página con el informe de resultados. El primer dato a proporcionar es sobre que objeto queremos hacer las búsquedas. Este objeto deberá ser un objeto Zope que exporte una interfaz que permita realizar búsquedas sobre él. Por ejemplo los Z SQL Method sirven. Desde la interfaz de búsquedas pasamos una serie de parámetros que luego se utilizan en el Z SQL Method para construir la petición SQL y consultar la base de datos. Otro tipo de objetos que se pueden utilizar con la interfaz de búsquedas de Zope son los catálogos, que es precisamente el caso que nos ocupa. Por ello seleccionamos entre los objetos disponibles para hacer búsquedas a "catalogo_rayodeluz" y damos identificadores y títulos a las páginas de búsqueda y presentación de resultados. Le damos a añadir y veremos que han aparecido dos nuevos objetos dentro de la carpeta "rayodeluz". Son "buscar_rayodeluz" y "datos_rayodeluz", que son objetos del tipo DTML Method creados de forma automática por Z Search Interface y que son los que implementan el formulario HTML que nos va a permitir hacer las búsquedas y la página de presentación de resultados de la búsqueda.

Ha llegado el momento de ver que es lo que realmente hemos creado. Para ello nos accedemos a la interfaz de búsquedas y vemos a ver de que opciones disponemos.

Como vemos se mantiene la cabecera y pie de la página por lo que no se pierde la identidad del sitio. Vemos la potencia de los mecanismos de adquisición que ya comentamos en anteriores entregas del curso y que permiten detalles como que las cabeceras y pie de página se compartan de forma transparente e integrada. Hemos creado una interfaz de búsquedas y sin hacer nada por nuestra parte se ha reutilizado la cabecera y pie por defecto.

En el formulario que se nos presenta podemos buscar por todos los campos por los que los objetos han sido indexados. Estos campos ya dijimos que se podían modificar dentro del propio catálogo aunque si los cambios, deberemos de volver a generar el interfaz de búsquedas. Vemos que nos podemos olvidar totalmente de como se crea el formulario HTML y del nombre de los campos y a que CGI se van a enviar los datos, Zope se encarga de todo esto por nosotros. Si queremos que se nos muestren todos los datos que existen en el catálogo, basta con que no rellenemos ningún campo y se nos devolverán todos los registros del catálogo.

Las posibilidades que se abren con este tipo de búsquedas son muy diversas y basándonos en ellas, podemos lograr mecanismos de localización de información utilizando características diversas muy potentes..

Objetos Externos

Después de presentar en anteriores números la arquitectura de Zope y entender como todo en él son objetos, hemos visto como se construye un sitio Zope a partir de los objetos estándar de Zope. Ahora vamos a analizar como se usa un objeto externo a la distribución de Zope tanto a la hora de instalarlo como de utilizarlo. Con ello veremos que una de los puntos más fuertes de Zope es su extensión a través de los cientos de objetos que ya están disponibles para él. La mayoría de estos objetos están cubiertos por la misma licencia que el propio Zope, la ZPL, por lo que su uso no lleva asociado ningún coste por licencias de los mismos. Recordemos que la licencia ZPL es considerada software libre, aunque es incompatible con la GPL y la FSF no recomienda el uso de esta licencia a la hora de publicar tu software.

La facilidad del uso de los objetos Zope, sobretodo aquellos que son de más alto nivel y proporcionan una funcionalidad sin necesidad de programación, hace muy aconsejable el conocer la variedad de objetos Zope que existen con el objetivo de ahorrar el mayor tiempo posible a la hora de construir nuestro sitio en la red.

A lo largo de este artículo nos vamos a centrar en el objeto Squishdot el cual nos va a permitir construir "weblogs" para nuestro sitio a golpe de ratón. Un weblog es un lugar donde aparece información ordenada por fechas. Ya iremos viendo más detalles sobre los weblog del que sitios como Slashdot.org o Barrapunto.com constituyen un excelente ejemplo.

Localización de objetos Zope

Estamos construyendo nuestro sitio en la red, Rayo de Luz, y el departamento de comunicación de la empresa ha solicitado que incluyamos un weblog en su zona del sitio para poder ir haciendo anuncios de las nuevas noticias que suceden dentro de la empresa. Ante esta petición evaluamos el coste de un desarrollo desde cero de esta funcionalidad pero, el coste del desarrollo sería demasiado elevado en tiempo y personal para cubrir las necesidades de un sólo departamento. Ante esta situación decidimos buscar algún objeto ya desarrollado que nos pudiera facilitar la labor (sí, ya sabemos que Squishdot nos vale pero hay que dar realismo a la situación ;-)).

Nos vamos de visita al web de Zope a ver si encontramos todos los objetos disponibles para Zope desarrolladores por otras personas y compañías ya que los objetos que vienen con el propio Zope no cubren esta funcionalidad. Realmente lo que estamos buscando son algo más que objetos. Son Productos Zope, conjuntos de objetos que proporcionan una funcionalidad y que son fácilmente instalables dentro de Zope. Si accedemos al portal de Zope y a su zona de descargas, tenemos una sección dedicada a los Productos.

Hay disponibles 263 productos Zope que nos proporcionan una funcionalidad muy grande. En nuestro caso estamos intentando localizar un weblog para nuestro departamento de comunicación.

De entre todas las categorías parece que la de "Realimentación" (Feedback) es donde podremos encontrar los Productos que más se ajusten a la necesidad de nuestro departamento, que lo que buscan es poder comunicar a los clientes de la compañía noticias y poder recibir realimentación por parte de los clientes a través de comentarios a las propias noticias, tal y como funcionan los weblog. Dentro de esta categoría nos encontramos con varios productos..

Los tres primeros productos que aparecen son de los más importantes dentro de Zope: Squishdot es el weblog de Zope, ZWiki, el producto para montar Wikis en Zope y PTK (Toolkit de Portales), un entorno para la creación de portales en el que todos los productos trabajan de forma integrada compartiendo información de usuarios, noticias, documentos y demás elementos que constituyen los fundamentos de un portal. Nosotros nos vamos a centrar en Squishdot pero el lector interesado puede aprender mucho con la instalación y uso de los otros dos productos y, especialmente con PTK, que pretende sentar las bases para el desarrollo con Zope de portales de una forma tan sencilla como se crean sitios en la actualidad.

Parece que Squishdot es lo que estábamos buscando. Aunque ya lo sabíamos, vemos que es sencillo de localizar los productos a través del portal de Zope y que con un a rápida búsqueda, podemos encontrar lo que necesitamos. En la página de información de Squishdot en el portal de Zope encontramos la referencia hacia la URL de su página principal.

La página de Squishdot es el propio producto en funcionamiento, la mejor prueba de que es justo lo que estábamos buscando. Analizando el contenido de la página, vemos que sitios como Technocrat.net de Bruce Perens, o las noticias de GNOME lo usan, y aquí más cerca en España la ACP del grupo de Sin Dominio que agrupa a muchas organizaciones de culturas alternativas, aunque actualmente tienen en revisión el servicio. Por lo que parece que es un producto estable y utilizable en entornos de producción, excelentes noticias. Es muy importante saber analizar hasta que punto podemos apostar por un determinado software, y más dentro del software libre que suele estar en constante evolución y que muchas veces se liberan versiones "alpha" para poder recibir apoyo de la comunidad. Squishdot parece un producto probado y utilizable por lo que podemos seguir con él y pasar a probarlo.

Lo primero es lograr una versión que sea compatible con la versión de nuestro servidor Zope. En Zope ha habido un cambio fuerte entre las versiones 2.1 y 2.2 y los productos que funcionan para una versión, no suelen hacerlo en muchos casos para la otra. En concreto, yo tengo instalada la versión de Debian Potato, un pelín antigua ya: la 2.1.6. Recordamos al lector que esta información la puede obtener del panel de control de Zope. Por este motivo me veo obligado a bajarme la versión 0.6 de Squishdot, aunque la actual es la 0.7.3. Si el lector están trabajando con Zope 2.2 debe bajarse esta última versión. Tras una descarga de 250 KB ya tengo en mi poder Squishdot-0-6-0.tar.gz, que veremos es muy muy sencillo de instalar.

Instalación de un Producto Zope

En el apartado anterior obtuvimos Squishdot-0-6-0.tar.gz, el lector con Zope 2.2 Squishdot-0-7-3.tar.gz, y ahora ha llegado el momento de instalarlo. Para ello basta con copiar este paquete al directorio del que cuelgan los demás productos Zope, en mi caso /usr/lib/zope/, y descomprimirlo en él.

```
faro:/shared/acs/src# cp Squishdot-0-6-0.tar.gz /usr/lib/zope/  
faro:/usr/lib/zope# cd /usr/lib/zope/  
faro:/usr/lib/zope# tar xzf Squishdot-0-6-0.tar.gz  
faro:/usr/lib/zope# zopectl restart  
Restarting Zope..... done.
```

Reiniciamos Zope para asegurarnos de que tiene en cuenta el nuevo producto y pasamos a ver si dentro de la lista de productos disponibles, ya tenemos a Squishdot.

Podemos ver que efectivamente, Squishdot ya aparece como un producto más de Zope. Dentro de la documentación de Zope se nos recomienda que instalemos el producto TinyTable que permite construir de forma sencilla tablas, y que es utilizado dentro de las pantallas de configuración de Squishdot. Un buen ejercicio para el lector es obtener este producto e instalarlo de forma idéntica a como se ha hecho con Squishdot.

En el caso de que el lector esté trabajando con la versión de Zope 2.2, y por lo tanto haya utilizado la versión 0.7.3 de Squishdot, la instalación es idéntica salvo que debe de descomprimir el paquete dentro del directorio con los productos de Zope. La documentación de instalación de Squishdot, disponible en el propio web de Squishdot, cubre este caso.

Uso del Producto Squishdot

Bueno, ha llegado el gran momento. Ya tenemos todo instalado y queremos ahora utilizar el producto Squishdot para crear un weblog dentro del departamento de contabilidad. ¿Cuánto de fácil será? El primer paso es irnos por la interfaz de gestión hasta el directorio, el objeto folder, donde tenemos las páginas del departamento de comunicación.

Bueno, ahora que ya estamos allí, vamos a seleccionar el objeto Squishdot y vamos a decirle que lo añadamos. Tras ellos nos encontramos con una primera pantalla con los datos básicos para configurar Squishdot.

Dejamos las opciones que aparecen por defecto y le damos un identificador al objeto (Noticias) y un título para la página (Noticias de Rayo de Luz). Vemos lo sencillo que ha sido crear el weblog pero, ¿qué es lo que realmente tenemos? Nada mejor que acceder a las páginas del departamento de comunicación e irnos al directorio "Noticias" a ver que tenemos allí.

Sí, ya tenemos nuestro Squishdot montado y preparado para poder utilizarlo. Tal y como cuentan los primeros artículos que se incluyen por defecto dentro de Squishdot, hemos logrado ya un 90% de la instalación. Pero aún queda un proceso importante: personalizar Squishdot para adecuarlo al aspecto de todo el sitio de Rayo de Luz y analizar la funcionalidad que nos proporciona el sitio. Pero a nivel de instalación del producto ya tenemos todo hecho y pasamos a configurar el producto para adaptarlo a nuestras necesidades. Para configurar Squishdot lo primero que tenemos que hacer es volver de nuevo a la interfaz de gestión del departamento de comunicación y acceder a la carpeta "Noticias" donde está el contenido de Squishdot.

De momento vemos que hay muchos elementos que forman el weblog. Cada uno de ellos es responsable de una parte de la visualización. Normalmente los nombres son bastante significativos y nos pueden guiar a que parte corresponde cada uno de los ficheros. Por ejemplo, hemos visto que en la página principal de entrada aparece un "banner" que no queremos que siga ahí. Para poderlo eliminar acudimos al objeto "site_header" y vemos que efectivamente, este método DTML es el encargado de construir la cabecera de la página del sitio.

Bastará con eliminar la línea que aparece seleccionada en la captura para que no nos aparezca más este banner en la parte superior de la pantalla de nuestro weblog. El lector puede eliminar esta línea, guardar los cambios y utilizar la solapa de "View" para comprobar que efectivamente, ha desaparecido. Pero en general, vemos que parámetros como los colores se obtiene a través de variables que están definidas en el propio Squishdot y que se pueden modificar desde la solapa de "Properties" donde se pueden cambiar muchos aspectos del aspecto de nuestro weblog. Por ejemplo, vamos a intentar cambiar el logo que aparece en la parte superior del weblog, que por defecto es el de Squishdot. Veamos todas las propiedades configurables a ver si alguna controla esta imagen.

A simple vista vemos algunas que tenemos que modificar como el correo del administrador, el nombre del sitio o el nombre del administrador. El nombre de la imagen que aparece en el título para que es "Images/sitetitle_img" y podemos comprobarlo si vamos a la carpeta de "Images" y visualizamos esa imagen. Para sustituirla nos vamos al directorio "img" de "rayodeluz" y copiamos la imagen "logo.png" seleccionando su checkbox y pulsando sobre el botón de copiar. Una vez que la tengamos copiada (nos aparecerá un botón para hacer Paste (pegar)) nos vamos de nuevo a la carpeta con nuestro weblog (Noticias) y pegamos dicho logo dentro del directorio "Images". Una vez realizado esta operación, acudimos de nuevo a las propiedades del weblog "Noticias" y cambiamos la que dice "title_image" al valor "Images/logo.png". Guardamos los cambios y seleccionamos la solapa de "View" para ver que tal ha quedado la imagen.

Bien, ahora vamos a cambiar el azul que aparece de forma generalizada por todo el sitio por un verde, más acorde con la imagen de nuestra compañía. Para ello de nuevo acudimos a "Properties" y cambiamos la propiedad "color3" al valor "#00FF00". Este valor tiene el componente de verde al máximo al ser los valores en hexadecimal de las componentes RGB de los colores, así que el color azul pasará a ser un verde demasiado fuerte, pero que es un buen punto de comienzo para seguir consolidando el diseño.

Por último se pueden configurar más cosas como los temas que forman el weblog. A la hora de poner una noticia nueva en el weblog, hay que elegir un tema bajo el que irá clasificado la noticia.

Luego las noticias se podrán visualizar agrupadas por temas, mecanismo que puede ser muy útil. En la siguiente figura vemos la interfaz para incorporar una nueva noticia al weblog, a la que se llega tras pulsar el enlace "post article" dentro de la página principal del weblog.

Para añadir nuevos temas al weblog, como por ejemplo uno dedicado a noticias de Rayo de Luz, otro para anuncio de eventos o incluso otro para promociones y ofertas a clientes, podemos ir a través de la interfaz de gestión del weblog "Noticias" a la solapa de Topics y desde allí se podrán borrar los temas actuales y añadir temas nuevos, para los que necesitaremos un icono.

Una vez instalado squishdot hemos ido viendo como toda la configuración del producto se realiza a través de la clásica interfaz web de gestión de Zope, algo que simplifica mucho esta labor y que permite realizar con sencillez y en poco tiempo las labores de mantenimiento y configuración del producto.

En general el lector encontrará esta facilidad de uso en casi todos los productos Zope que, como se ha comentado en el artículo, son a fecha de Enero de 2001 más de 260. ¿Dispuesto a probar alguno de ellos?

Después de dar a conocer la arquitectura Zope, mostrar un ejemplo de creación de un sitio con Zope, Rayo de Luz, y presentar los objetos básicos en Zope, ha llegado el momento de atacar otro de los temas más importantes del desarrollo de aplicaciones en Internet: las bases de datos.

Adaptador de acceso a base de datos

La idea detrás de los adaptadores de acceso a base de datos de Zope es la tradicional dentro del mundo de las bases de datos: crear una capa de abstracción con la interfaz propia de la base de datos. De esta forma se logra una interfaz unificada para acceder a cualquier base de datos que tenga adaptador para Zope. Esto es algo totalmente similar a nivel conceptual a ODBC, JDBC o Perl DBI. En la actualidad existen para Zope adaptadores para acceder a:

MySQL

Postgres

Oracle

Sybase

Interbase

ODBC

Gadfly

Al tener un adaptador para ODBC podemos acceder a casi cualquier base de datos, ya que ya es raro a día de hoy que una base de datos no sea accesible al menos desde ODBC.

En el caso del software libre los adaptadores que más nos interesan son el de MySQL y el de Postgres. En el caso de Postgres se dispone de varios adaptadores, siendo la base de datos mejor soportada desde Zope, a parte de la propia ZODB, que recordemos es la base de datos que usa Zope para guardar todos los objetos. En este artículo nos vamos a centrar en MySQL, pero todo lo que digamos es idéntico para cualquiera de los otros adaptadores a excepción de la instalación del producto que contiene el adaptador para cada base de datos concreta.

A lo largo del artículo vamos a ir cubriendo el proceso de instalación de un adaptador de acceso a base de datos desde Zope, su uso desde Zope y algunas herramientas que se basan en estos adaptadores y que permiten de forma sencilla crear formularios para almacenar datos en las base de datos.

Instalación del adaptador de MySQL

Los adaptadores de Zope para acceso a base de datos no son otra cosa que productos Zope. Por ello para su instalación localizamos el producto desde las páginas del portal de Zope y lo instalamos. En el caso de la distribución Debian 2.2/Potato es muy sencilla la instalación de los adaptadores de MySQL y Postgres ya que dispones de paquetes para ello.

```
faro:/shared/acs# apt-get install zope-mysqlda
```

```
Reading Package Lists... Done
```

```
Building Dependency Tree... Done
```

```
The following extra packages will be installed:
```

```
python-mysqldb
```

```
The following NEW packages will be installed:
```

```
python-mysqldb zope-mysqlda
```

```
0 packages upgraded, 2 newly installed, 0 to remove and 59 not upgraded.
```

```
Need to get 0B/42.8kB of archives. After unpacking 174kB will be used.
```

```
Do you want to continue? [Y/n]
```

```
Media Change: Please insert the disc labeled 'Debian GNU/Linux 2.2 r0 _Potato_ - Official i386 Binary-3 (20000814)' in the drive '/cdrom/' and press enter
```

```
Selecting previously deselected package python-mysqldb.
```

```
(Reading database ... 50092 files and directories currently installed.)
```

```
Unpacking python-mysqldb (from .../python-mysqldb_0.1.1-1.deb) ...
```

```
Selecting previously deselected package zope-mysqlda.
```

```
Unpacking zope-mysqlda (from .../web/zope-mysqlda_1.1.3-1.deb) ...
```

```
Setting up python-mysqldb (0.1.1-1) ...
```

```
Setting up zope-mysqlda (1.1.3-1) ...
```

```
For Zope to recognize the new/updated package, it will need to be restarted.
```

```
Restart Zope now? [Y] Y
```

Podemos ver que también se instala el paquete python-mysqldb. No olvidemos que casi todo en Zope está hecho en Python, aunque Perl poco a poco irá ocupando un papel importante, y python-mysqldb es una librería de Python para acceder a MySQL. No vamos a cubrir en el artículo la instalación manual del adaptador pero el lector puede encontrar información al respecto dentro de la página del adaptador de MySQL que se encuentra en las referencias.

Si todo ha ido bien deberíamos de tener un nuevo objeto disponible en Zope, el de conexión a un servidor de bases de datos MySQL. Si lo seleccionamos por primera vez nos pedirá los datos necesario para poder conectar con la base de datos. Vamos a partir de que ya tenemos instalado el servidor de MySQL en el sistema y vamos a crear la base de datos para RayodeLuz y a dar permisos al usuario "rayodeluz" para que pueda sacar datos de ella y modificarlos, aunque no borrarlos. MySQL dispone de la función de SQL GRANT que es muy útil en este tipo de situaciones ya que permite controlar con mucha precisión y sencillez que puede hacer cada usuario en la base de datos. Pero como no estamos en un artículo de MySQL, no profundicemos más en el tema. Los comandos a enviar a MySQL desde su cliente "mysql" son:

```
mysql> create database RayodeLuz;
```

```
mysql> GRANT SELECT, UPDATE, INSERT ON RayodeLuz.* TO rayodeluz@localhost;
```

Ahora ya desde la pantalla de configuración de conexiones MySQL configuramos estos datos para poder acceder al servidor de MySQL y vemos que se establece una conexión con MySQL.

Este objeto que se acaba de crear es ya una conexión permanente con MySQL que nos sirve para, a través de él, poder enviar comandos SQL a la base de datos, entre otras cosas que ya iremos viendo.

Creación de una base de datos de Clientes

Ahora que buscábamos un buen ejemplo para mostrar el funcionamiento de las bases de datos en Zope, recibimos una llamada del responsable del departamento comercial: necesitan un sistema de seguimiento de contactos con los clientes y los potenciales clientes de la empresa. Bien, esto es sencillo. Nos creamos una tabla donde almacenamos esta información, la cual se llenará con los datos que vayan introduciendo los comerciales de la empresa y en la que tendremos que poder hacer búsquedas por distintos criterios.

Manos a la obra: lo primero es crear una tabla para almacenar la información de los contactos comerciales.

```
CREATE TABLE contactos (  
  id int(11) DEFAULT '0' NOT NULL auto_increment,  
  nombre varchar(64),  
  contacto varchar(64),  
  telefono varchar(32),  
  email varchar(32),  
  url varchar(32),  
  comentario text,  
  autor varchar(32),  
  sello timestamp,  
  PRIMARY KEY (id)  
);
```

Para poder ejecutar este comando podemos utilizar ya la conexión con la base de datos que hemos creado. Vemos que entre todos los objetos que tenemos creados como parte del sitio de Rayo de Luz, uno de ellos es la conexión a base de datos MySQL.

Si seleccionamos el objeto "MySQL_database_connection" tenemos una serie de posibilidades como son ver la descripción del objeto para ver a que base de datos está conectando y como que usuario, podemos desconectar al objeto de la base de datos y volverlo a conectar y una de las solapas accesibles para este objeto se llama "Test". Tras ella tenemos una simple interfaz que nos permite hacer peticiones a la base de datos.

Si intentamos enviar la sentencia SQL de creación de la tabla recibimos un mensaje de error ya que en la sentencia de "GRANT" (permitir) del usuario "rayodeluz" no incluimos la posibilidad de crear tablas. Basta con hacer un nuevo GRANT de la siguiente forma:

```
mysql> GRANT CREATE ON RayodeLuz.* TO rayodeluz@localhost;  
mysql> FLUSH PRIVILEGES;
```

Es bueno tras realizar este tipo de cambios volver a reconectar con la base de datos para que tenga en cuenta las modificaciones. Para ello basta con ir a la ventana principal del objeto de conexión a la base de datos y pulsar sobre el botón de desconectar. Luego se pulsa de nuevo el botón de conectar y ya tendremos de nuevo abierta la conexión con la base de datos. Si volvemos a la solapa "test" desde la que podemos hacer peticiones y enviamos el comando SQL de creación de la tabla, esta será creada. Algo que podemos verificar con el comando SQL "show tables".

Vemos pues que tenemos ya una conexión con la base de datos y que podemos enviar peticiones SQL a través de este objeto. Es más, los resultados de las peticiones SQL se nos presentan en tablas HTML que incluso podrían servir para mostrar los resultados de las búsquedas pero, ¿cómo creamos ahora los formulario que nos permitan introducir datos en la base de datos? ¿Y aquellos que nos permiten realizar búsquedas?

La primera gran utilidad que tenemos son los objetos ZSQL Methods que se asocian a una conexión a una base de datos y permiten construir peticiones SQL. Las dos funcionalidades que queremos cubrir son la inserción de datos y su posterior consulta y como vemos a continuación, ambas funciones quedan cubiertas con métodos SQL.

Para crear un formulario de inserción de datos en la tabla creamos un nuevo objeto ZSQL en el que como parámetros aparezcan los campos que queremos poder introducir en la base de datos y la petición sea de tipo INSERT.

Vemos que el procedimiento es realmente sencillo y válido de forma general para cualquier tipo de formulario

Pero para las búsquedas disponemos de otro sistema un poco más sencillo de crear y más potente, los objetos Z Search Interface. Con ellos, la construcción de complejas peticiones de búsqueda dentro de la base de datos se hace de una forma mucho más cómoda y que además, podremos personalizar de forma sencilla. Un objeto Z Search Interface utiliza un ZSQL Method para definir sobre que campos se quiere poder buscar y genera de forma automática el formulario de búsqueda y la página de presentación de resultados. Este objeto ya lo utilizamos cuando hablamos de las posibilidades de catalogación de contenidos en Zope,

utilizando ZCatalog, y para construir el formulario de búsqueda y la página de presentación de resultados sólo necesita un objeto que sea "buscable". Y los ZSQL Method lo son, por lo que podemos seleccionar por ejemplo nuestro método de localización de contactos para construir sobre él la interfaz de búsqueda. Como ya hablamos en su momento a partir de Z Search Interface se crean dos nuevos objetos Zope mediante los cuales el usuario puede introducir los datos de la búsqueda y puede ver los resultados.

Prácticamente ya hemos visto como se usan las bases de datos relacionales desde Zope. La gran parte que quedaría por analizar es ver como desde DTML y desde clases Python se pueden utilizar estos objetos de acceso a base de datos para poder utilizar de una forma más directa la información que se guarda en las bases de datos relacionales. Pero como durante el curso no hemos cubierto aspectos de programación de más bajo nivel y hemos preferido seguir una línea más orientada a la construcción de sitios a partir de objetos ya realizados, no vamos a entrar en estas posibilidades.

Las bases de datos relacionales han sido durante los últimos años el lugar preferido en la industria del desarrollo software para guardar los datos de las aplicaciones. El lograr poder acceder desde Zope a este tipo de fuentes de datos de forma sencilla nos permite estar tranquilos cuando pensemos en la integración de Zope con sistemas ya existentes. Y al disponer Zope de acceso tanto a bases de datos relacionales como a su base de datos orientada a objetos, ZODB, le convierte en una arquitectura muy flexible que puede dar cabida a complejas arquitecturas.

La llegada de Internet a impulsado aún más la importancia de las bases de datos. La posibilidad de construir bases de datos de forma distribuida a través de la red y al necesidad cada vez mayor de gestionar ingentes cantidades de datos hacen que MySQL y Postgres ocupen a día de hoy un papel estrella dentro del desarrollo de aplicaciones en Internet. Zope puede acceder sin problemas a estas bases de datos por lo que entornos combinados donde aplicaciones Perl o PHP trabajen con partes Zope son perfectamente factibles ya que se pueden compartir la información en las bases de datos. Según madura el desarrollo de aplicaciones para Internet, este tipo de entornos mixtos cobrarán cada vez más importancia ya que en muchos proyectos se ahorrará mucho tiempo gracias a la reutilización de diferentes partes ya desarrolladas. Un ejemplo podría ser un portal para desarrolladores utilizando el Portal Toolkit de Zope pero que para la parte de gestión de los proyectos, utilice Source Forge, basado totalmente en PHP, pero que guarda sus datos en MySQL y, para la parte de foros, se utilizaría slashcode, el software en el que se basa Slashdot y BarraPunto, y que está desarrollado en Perl, pero que de nuevo, guarda toda la información en una base de datos MySQL.