

Diseño y desarrollo de aplicaciones web multidispositivo

César Colado Rodríguez

*Germinus XXI
Febrero de 2003*

El reto que supone la diversidad de dispositivos utilizados para acceder a la web, pasa generalmente desapercibido por las personas que no están directamente involucradas en la creación de portales en Internet. En este artículo se introduce lo que comúnmente se empezó a denominar “portal” y se define lo que se entiende por acceso multi-dispositivo, para poder situar algunas de las dificultades y soluciones que se plantean a las personas que deciden crear un portal accesible tanto desde un navegador en un PC, como desde un móvil o una PDA.

1 Introducción

El World Wide Web Consortium (W3C) define el término “web” como: *“el universo de información accesible a través de la red”*. Una aplicación web es un sistema que permite a un usuario final acceder a una parcela de información contenida en el universo al que hace referencia la anterior definición del W3C. Las aplicaciones web son aplicaciones basadas en el muy extendido paradigma “cliente/servidor”. Este paradigma consiste en un servidor que sabe cómo proporcionar un servicio y un cliente que desea acceder al servicio. No es necesario imaginar nada enrevesado, basta con pensar en algo que se hace a diario, como por ejemplo “leer el periódico”. Existe un servidor que contiene las noticias del día y es necesario una aplicación que acceda a este servidor de alguna forma para obtener la información. Esta aplicación cliente, debe ser capaz de mostrar la información al usuario final que desea leer las noticias.

El concepto “portal” fue acuñado a partir de 1996 y se comenzó a utilizar para referirse a lugares dónde era posible encontrar una oferta de servicios de valor añadido, tales como correo electrónico, noticias, información, etc. [Tro01]. Históricamente, el término “portal” se diferenciaba del de “sitio web” en que se refería a un lugar de partida hacia otros sitios en Internet. No obstante hoy en día estos términos se utilizan indistintamente, haciéndose un uso preferente de la denominación portal, cuando el sitio web integra diferentes servicios de valor añadido y no sólo páginas estáticas [Mil02].

En los últimos años han aparecido muchos dispositivos capaces de acceder a la web, más allá del tradicional PC: televisiones, PDAs, móviles, etc.

Las expectativas de los usuarios finales también han cambiado. Los usuarios esperan acceder a información y servicios importantes para su negocio desde cualquier sitio y a cualquier hora. Así por ejemplo un usuario desea poder comprar desde su vehículo las entradas para el cine, de la misma forma que lo podía haber hecho utilizando el PC de su casa.

Todo esto tiene como consecuencia directa que hoy día los contenidos y servicios de un portal deben ser accesibles utilizando distintos mecanismos de acceso, desde la televisión o el ordenador personal en casa, hasta el móvil o la PDA allá donde se encuentre el usuario. La presentación y la interacción con el usuario de estos servicios y contenidos debe estar adaptado al mecanismo de acceso que él emplee, con objeto de proporcionar una experiencia gratificante. Pero a la vez debe mantenerse una coherencia y un parecido que permitan que el usuario sienta que está en el mismo portal con independencia del mecanismo de acceso que emplee en cada momento.

Sin embargo, los creadores de portales no pueden estar aprovisionando por separado contenidos y servicios para cada una de las formas de acceso posibles. Hacer de un portal, un lugar accesible utilizando distintos mecanismos de acceso debe ser un proyecto abordable en términos de esfuerzo.

Una problemática adicional que presenta el desarrollo de portales multi-dispositivo es la particularización no sólo de los elementos que componen la presentación, sino también la interacción del usuario con el portal en función de su mecanismo de acceso. Se trata de que sea sencilla, intuitiva, amigable, lo que se consigue variando la composición, la navegación e incluso la forma en la que el usuario introduce información.

Es necesario prevenir la fragmentación de la información en función del medio y dispositivo utilizado para acceder al mismo.

2 Arquitectura de un portal

En la actualidad, el desarrollo de portales es concebible en términos de presentación, lógica de aplicación y datos. Sin embargo estos grandes bloques no pueden ser considerados de forma abstracta y aislada. Muchos de los servicios y contenidos que se presentan en un portal se desarrollan como verticales sobre estas capas. Es el caso de los sistemas de información geográfica (GIS), los motores de búsqueda, etc. Se trata de verticales que requieren de lógica y datos propios, y que se integran en el portal mediante la capa de integración constituida en el marco del portal por la lógica de aplicación.

Ni los verticales a los que se ha hecho referencia, ni las funcionalidades horizontales compartidas por éstos, adquieren el mismo protagonismo en todos los portales. El tipo de aplicaciones y contenidos de un portal determinan la puesta en escena y la importancia de cada una de las capas. En algunos casos, la gestión de contenidos adquiere un papel relevante por tratarse de un portal basado primordialmente en contenido agregado, mientras que en otros carece de importancia por ser externo la mayor parte del contenido del portal. En la Ilustración 1 se muestran las capas más importantes que deben ser tenidas en cuenta en el desarrollo de un portal.

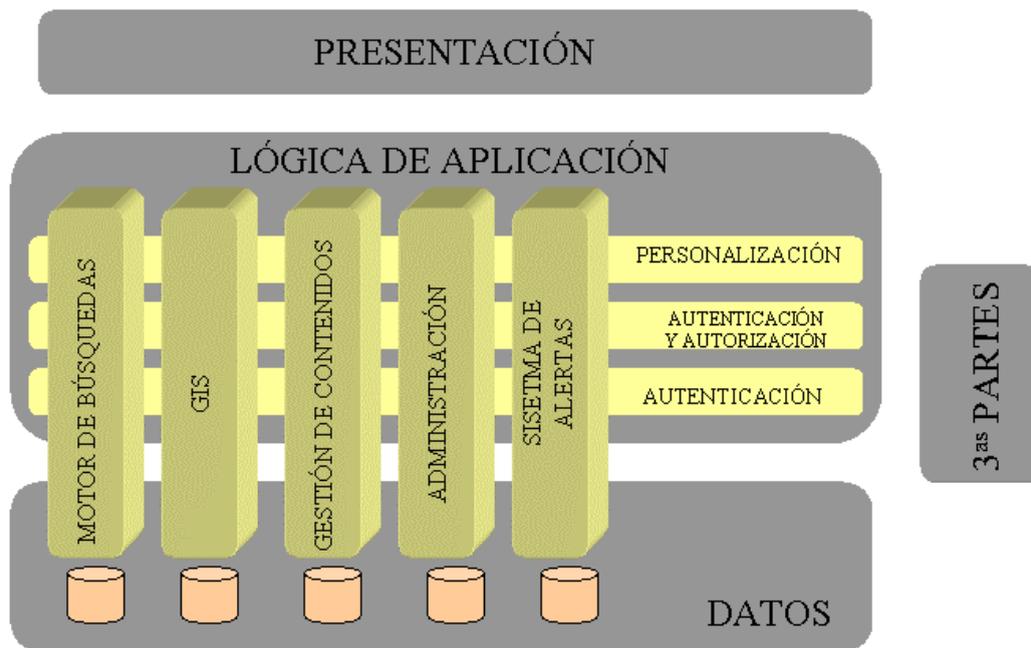


Ilustración 1. Estructuración en capas de un portal de aplicaciones

Presentación. En la capa de presentación se establece la composición de los contenidos que se presentan finalmente al usuario, como resultado de su petición. Se añade información de presentación a los contenidos: dónde tienen que aparecer, cuál es el tamaño de letra más adecuado, dónde va el título, dónde la fotografía, dónde el gráfico, etc.

Lógica de aplicación. La lógica de aplicación es la parte del portal encargada de atender las peticiones de los usuarios. Esta lógica proporciona valor añadido al contenido almacenado, combinando la información y adaptándola a las necesidades de cada usuario. En caso de tratarse de contenido externo, esta lógica es la encargada de interpretar qué tipo de conexión es necesario establecer con el proveedor y si es o no necesario aplicar transformaciones antes de servirlo al usuario final.

Datos. En la capa de datos es donde se almacena de forma persistente toda la información necesaria para facilitar los servicios ofrecidos por el portal. El perfil de los usuarios del portal, índices de los motores de búsqueda, contenidos de agregación, información sobre la publicación, así como información sobre las terceras partes que proveen contenido o servicios al portal.

Terceras partes. Las terceras partes son proveedores de contenidos o servicios. Un proveedor de la cotización de las acciones de bolsa en tiempo real puede ser un buen ejemplo.

2.1 Presentación

Es en esta capa en la que se compone la presentación o página que se va a mostrar al usuario como resultado de su petición. La petición del usuario debe identificar una página, de forma independiente del mecanismo de acceso utilizado, sin embargo, junto al identificador de la página que desea acceder, el usuario o el agente de usuario (aplicación que representa al usuario) suministra información sobre el mecanismo de acceso concreto utilizado. Es lo que se denomina: **contexto de entrega**.

El contexto de entrega debe tener información no sólo del dispositivo utilizado, sino también de las características de la red de acceso y las preferencias de usuario. Es tan importante conocer el tamaño de la pantalla del dispositivo, como el hecho de que es incapaz de mostrar imágenes, el reducido ancho de banda de la conexión utilizada o el deseo expreso del usuario de no ver imágenes. En esta línea se han elaborado recomendaciones como CC/PP [W3C02b] o UAProf [WAG99]. A pesar de ello, hoy en día, en la mayoría de los casos, la detección de los dispositivos de acceso se basa primordialmente en el campo “User-Agent” de la petición HTTP [RFC99].

Es en esta capa, y generalmente después de añadir información de presentación relativa al lenguaje de presentación utilizado por el dispositivo de acceso, se adapta el resultado al mecanismo de acceso concreto que se está utilizando. Para lo que en muchos casos se utilizan herramientas tales como Everix [Mob03a], Volantis [Vol03] o WebSurf [Mob03b].

Algunas de las herramientas de adaptación requieren de la inserción de meta-información en las páginas dinámicas del portal. La interpretación de éstas supone el primer paso en la composición de la presentación; para después eliminar las partes no necesarias para el mecanismo de acceso empleado en un segundo paso, en el que puede además realizarse una adaptación más fina del resultado final al dispositivo. Este tipo de aproximación es la que frecuentemente se realiza con herramientas como Everix o Volantis. Por otro lado otras herramientas, como WebSurf, consideran el resultado que se va a entregar a un dispositivo, un subconjunto del resultado para web. Por ello no requieren de la utilización de meta-información, pero sí una definición de las partes de cada página que se desea sean accesible para cada dispositivo, así como la forma de navegación.

2.2 Lógica de aplicación

La lógica de aplicación sirve como capa integradora en el portal, constituye el núcleo. La lógica de aplicación está muy condicionada por las decisiones de diseño que se comentarán más adelante. En esta capa aparecen verticales que proporcionan una funcionalidad específica y un conjunto de capas horizontales que proporcionan servicios básicos a todos los verticales que se integren en el portal.

Son capas horizontales la seguridad, la autenticación y autorización, así como la personalización. En un portal destinado a usuarios, no cabe duda que el hecho de conocer al usuario que accede, proporciona un valor añadido que permite ofrecer servicios específicos y mejorar la experiencia del usuario en el portal.

La autenticación puede basarse en el identificador de abonado (MSISDN) cuando el usuario accede al portal mediante su móvil y no requerir que se proporcione ninguna clave que puede ser necesaria cuando el acceso se realiza utilizando otros mecanismos de acceso. Otras posibilidades en la autenticación pasan por la solicitud de una clave, hasta un certificado digital.

El acceso a los servicios del portal o a determinada información puede ser permitida sólo a determinados usuarios, por ser éstos usuarios de pago, usuarios registrados o cumplir una serie de características arbitrarias, es lo que se denomina autorización.

Por último la personalización supone una adaptación del portal a los gustos, preferencias y comportamiento del usuario. La personalización puede variar desde llamar al usuario por su nombre, hasta adaptar la composición y los contenidos a sus gustos y comportamiento en el portal o la localización del usuario en tiempo real. Se trata de facilitar la navegación del usuario, hacerla más amigable y hacer de su experiencia en el portal una experiencia positiva. Las personalizaciones complejas se suelen basar en la aplicación de reglas sobre información recogida en el proceso de registro y el comportamiento en tiempo real del usuario en el portal, lo que mediante la utilización de motores de inferencia puede derivar en la aplicación de campañas de marketing o incluso la anticipación del portal al comportamiento y preferencias del usuario (comportamiento proactivo). Es el caso de Amazon.com.

Los verticales son:

- **Motores de búsquedas.** Sistemas capaces de buscar información entre los contenidos del portal, para lo que se utilizan unos motores de indexación sobre el contenido. Los motores de indexación leen de las fuentes indicadas y almacenan información sobre la relevancia de términos y su relación según algoritmos muy particulares de cada solución. La problemática multi-dispositivo, lo constituye la variedad de los contenidos, así como las diferentes presentaciones del resultado de una búsqueda. Las aproximaciones a este problema son básicamente dos: una primera más burda, en la que se utiliza un índice por dispositivo y se indexa el contenido como si de un usuario accediendo al portal se tratase. Y la segunda, mucho más fina, en la que el proceso de indexación se hace sobre la información almacenada en la base de datos, y de cuya presentación se encargará un motor capaz de aplicar una presentación u otra, en función de algún parámetro del contenido como pudiera ser su categoría (ver Ilustración 2).

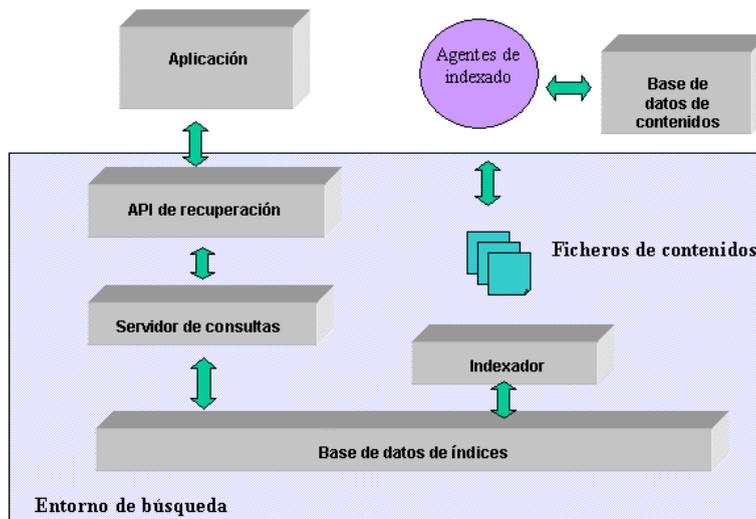


Ilustración 2. Motores de búsquedas

- **Sistemas de información geográfica (GIS).** Este tipo de funcionalidades se ve especialmente favorecido por el acceso multi-dispositivo, que generalmente es el origen de la información en tiempo real de parámetros de localización del usuario tales como la latitud y la longitud en la que se encuentra. Los sistemas de información geográfica pueden, a partir una longitud y latitud, situar a un usuario en un número de una calle concreta de una ciudad. Pudiendo entonces, facilitar al usuario mapas de su localización, rutas para llegar a su destino, así como recomendaciones de puntos de interés en los alrededores de su posición actual. La dificultad de esta funcionalidad en un portal multi-dispositivo está en la presentación de la información, el formato de las imágenes de mapas debe ser específico, al menos para un conjunto de dispositivos. Al igual que las imágenes, las descripciones de las rutas a seguir deben ajustarse al mecanismo de acceso utilizado. Sin embargo muchas de estas dificultades ya vienen resueltas por los componentes encargados de cubrir estos verticales. Es así en el caso de Webraska [Web03] capaz de proporcionar los mapas en distintos formatos y describir las rutas en XML, lo que permite una fácil adaptación al mecanismo de acceso. Es importante destacar que a pesar de ser fácil, la adaptación no se hace de forma automática, es necesario integrar este elemento en la lógica de negocio del portal y hacer uso de la política de adaptación definida, para solicitar del sistema un recurso u otro, y adaptarlo.
- **Gestión de contenidos.** Los contenidos que se han recibido de los distintos proveedores deben someterse a un proceso de normalización, lo que permite almacenarlos en una estructura común con independencia del proveedor que lo ha suministrado. Generalmente la gestión de contenidos permite establecer un ciclo de vida para un contenido y flujos, de forma que siguiendo una serie de reglas o de forma manual, un contenido puede estar almacenado pero no publicado, puede haber caducado o pasar a cualquier estado de los que se definan según las necesidades de la gestión de contenidos del portal concreto. La problemática de la gestión de contenidos en un portal multi-dispositivo reside en la necesidad de distintas versiones de un mismo recurso, para poder cubrir los

distintos mecanismos de acceso. Esto ocurre con las imágenes, lo vídeos y en general con todos los contenidos multimedia. En algunas ocasiones es posible generar de forma automática tantas versiones como sean necesarias aplicando transformaciones al contenido, lo que se hace efectivamente con imágenes, a veces en tiempo de agregación, almacenando las distintas versiones; y otras veces en tiempo de petición, utilizando mecanismos de transformación como los proporcionados por herramientas como WebSurf. Los contenidos textuales suelen entregarse en XML [W3C03], lo que hace que sean independientes de la presentación, pero sin embargo, es necesario que se suministren distintas descripciones del mismo contenido, unas más cortas que otras por motivos evidentes. Al aprovisionamiento de distintas versiones del mismo recurso se suma el problema de la elección del recurso adecuado por la capa de presentación.

- **Administración.** La administración suele realizarse siempre mediante el mismo mecanismo de acceso, generalmente un navegador en un PC. El reto de un portal multi-dispositivo, en lo que a administración se refiere, está en la elección de los recursos más apropiados para un determinado mecanismo de acceso, la configuración de la navegación, la composición de la presentación y la interacción con el usuario para cada uno de los dispositivos o grupos de ellos.
- **Sistemas de alertas.** En un portal multi-dispositivo debe ser posible utilizar medios como el correo electrónico, los “Multimedia Messaging Services” (MMS), WAP PUSH o “Short Message Service” (SMS), para informar al usuario de alertas que haya definido, para ser informado de cuando llega un nuevo tipo de contenido, del resultado de un partido, de la cotización de unas acciones, etc. La definición de alertas es una funcionalidad que proporcionan muchos componentes verticales, cada uno en su dominio. Sin embargo, determinados productos como el motor de búsqueda Smartlogik facilita la definición y gestión de alertas. Estos sistemas requieren de la utilización de servicios asíncronos, generalmente basados en colas, como Java Message Service (JMS).

3 Decisiones de diseño

La primera problemática que se plantea en el diseño de portales multi-dispositivo es la necesidad de realizar una web por dispositivo o tipo de presentación (ver Ilustración 3) o una única web que se adapte a cada dispositivo mediante una capa de personalización o adaptación al mecanismo concreto de acceso utilizado por el usuario (ver Ilustración 4) . La primera solución plantea un claro problema de escalabilidad por su escasa mantenibilidad, mientras que mediante el segundo enfoque se pierde la granularidad a la hora de mejorar la navegabilidad, la composición, los recursos o la interacción para un mecanismo de acceso determinado, así como se complica la lógica y se empeora la eficiencia del sistema.

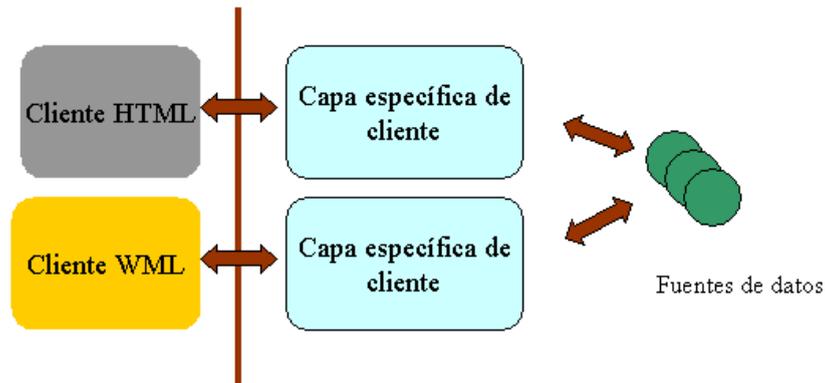


Ilustración 3. Lógica disjunta

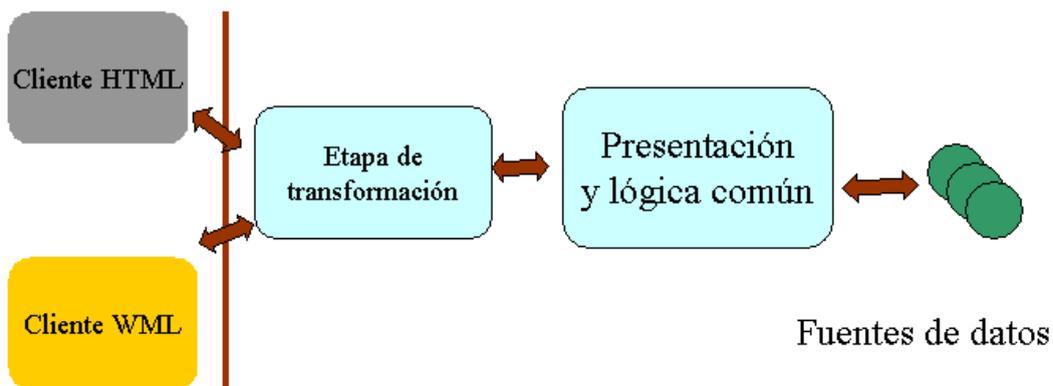


Ilustración 4. Aproximación mediante transformaciones

El estilo arquitectónico establece la forma en la que se ensamblan los componentes del portal. A grandes rasgos es posible diferenciar las siguientes aproximaciones:

- **Arquitectura basada en los datos.** Esta arquitectura se emplea en la mayor parte de portales con tecnología “Java Server Pages” (JSP) o “Active Server Pages” (ASP) para poder ofrecer contenidos dinámicos. Esta aproximación se conoce con el nombre de modelo 1, y se trata de una arquitectura en la que sencillamente se invoca un componente capaz de acceder a información almacenada en el portal y presentarla directamente. Tanto la lógica de acceso a los recursos como la lógica de aplicación se encuentran en este componente. La ventaja de este tipo de arquitectura es obviamente su simplicidad, aunque hace difícil la escalabilidad y la reutilización de componentes. La utilización de este tipo de arquitecturas en portales multi-dispositivo es posible utilizando estructuras de control en las propias páginas en caso de querer hacer ajustes bajo de la capa de adaptación.

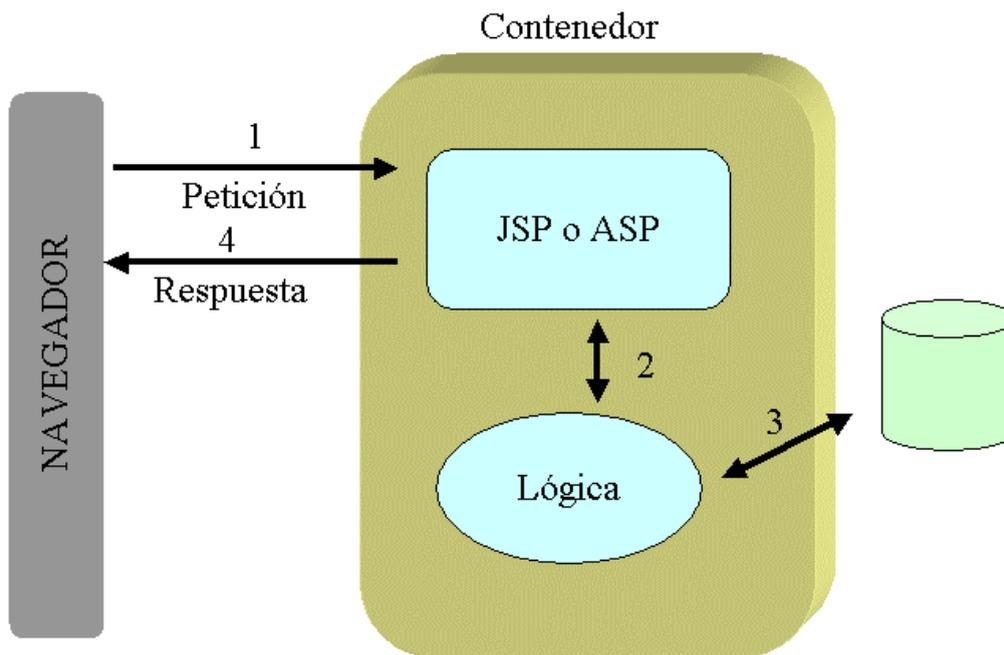


Ilustración 5. Arquitectura modelo 1

- **Arquitectura centrada en el gestor de contenidos.** Es el caso de portales que se adaptan a la funcionalidad proporcionada por un gestor de contenidos. La ventaja de este tipo de arquitecturas es la enorme reducción en el desarrollo, prácticamente es una solución cerrada. Sin embargo, el crecimiento de este tipo de soluciones no siempre es posible, a pesar de ser modulares. Es muy costoso integrar elementos nuevos en el portal. Este tipo de soluciones rodean gestores de contenidos como Vigette o soluciones cerradas como las que plantea Oracle con su Oracle Portal Server [Ora03], o BEA con BEA WebLogic Portal [BEA03]. También existen otro tipo de gestores de contenidos libres, tal es el caso de PostNuke [Pos03] y PHPNuke [PHP03] en PHP y otros como OpenCMS [Ope03] en Java.
- **Arquitectura Modelo-Vista-Controlador (MVC).** Este tipo de arquitectura proporciona una separación entre lo que es la lógica de negocio y la vista (presentación). Además el controlador suele proporcionar servicios horizontales, tales como la autenticación. El controlador funciona redirigiendo la petición, por lo que es fácil cambiar la presentación para una determinada petición, así como tener una aplicación con diferentes vistas para distintos mecanismos de acceso que comparten la lógica de aplicación. Es posible encapsular el estado de la aplicación y mejora la reusabilidad al definir claramente las responsabilidades de los componentes participantes. Como desventaja, destaca la complejidad que supone realizar desarrollos siguiendo esta arquitectura.

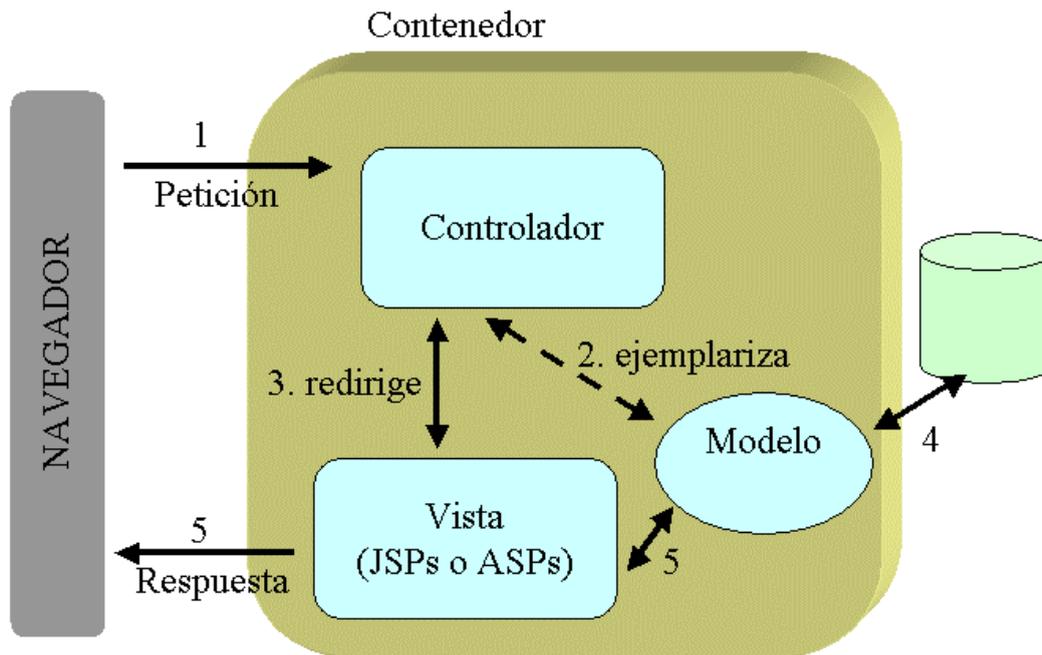


Ilustración 6. Arquitectura MVC

- **Arquitectura de Portlets.** Un portlet es un componente web controlado por un contenedor, capaz de generar contenido dinámico e interactuar con los usuarios a través del portal. La ventaja que plantean estos elementos es su versatilidad a la hora de ser desarrollados y desplegados. Los portlets pueden generar distintos tipos de contenidos en función del mecanismo de acceso, información a la que tienen acceso. Un portlet debe ser visto como la encapsulación de una aplicación web para poder ser utilizada como un componente dentro de un portal. No obstante hoy en día esta aproximación puede ser exagerada. Resulta complicado en la aproximación que hace Jakarta con Jetspeed [Jak03], implementar un modelo MVC dentro de un portlet. Sin embargo, la velocidad con la que avanza esta implementación Open Source hace prever que no tardará mucho en alcanzar los ambiciosos objetivos de esta arquitectura, proporcionando un mecanismo modular y enormemente reutilizable para la construcción de portales.

4 Conclusiones

Los portales están en constante evolución, se trata generalmente de un sistema vivo, que crece y se adapta a las necesidades de los usuarios en cada momento. Para que esto sea posible son varias las características que debe cumplir la arquitectura de componentes elegida:

- **Arquitectura modular y flexible,** es decir, los componentes y su integración deben poder adaptarse a diferentes configuraciones de hardware y red. Debe ser posible instalar todos los componentes en una misma máquina o distribuirlos convenientemente para mejorar el rendimiento del sistema en conjunto. La distribución de los componentes se hace muchas veces atendiendo a cuestiones

de seguridad, separando generalmente los componentes que forman parte del “frontend” y los que forman parte del “backend”.

- **Escalabilidad horizontal y vertical.** Los componentes utilizados deben poder mejorar en prestaciones añadiendo recursos a las máquinas en las que se encuentran desplegados o sumando la capacidad de nuevos servidores en la arquitectura.
- **Alta disponibilidad y alto rendimiento.** Tal vez no es una característica que se desee desde un principio, pero desde luego, la arquitectura implementada debe poder crecer hacia un diseño que garantice estos aspectos claves en portales importantes.
- **Seguridad.** Los componentes deben evitar el acceso no autorizado a la lógica, recursos, información, aplicaciones, etc. Esto se puede lograr mediante el cifrado de la información, el uso de certificados digitales, estableciendo mecanismos de seguridad perimetral, etc.

El caso de los portales multi-dispositivo, plantea infinidad de cuestiones complejas de abordar, tales como el aprovisionamiento de distintas versiones del mismo recurso semántico, el uso apropiado de los recursos almacenados en el portal en función del mecanismo de acceso, la elección de la arquitectura adecuada que permita alcanzar los requisitos de eficiencia y adaptabilidad necesarios para el portal. En general las soluciones multi-dispositivo empeoran el rendimiento del sistema en su conjunto y complican la arquitectura, pero permiten la adición de nuevos servicios y el número de usuarios potenciales de un portal se multiplica por hacerse realidad “quién quiera, desde dónde quiera, cuando quiera y cómo quiera”.

No existe la solución perfecta, conviene considerar las necesidades particulares de cada caso para alcanzar una solución adecuada, escogiendo una arquitectura flexible pero no excesivamente generalista que pueda suponer una carga excesiva

5 Referencias

- [W3C02a] Authoring Scenarios for Device Independence. W3C. Julio 2002
<http://www.w3.org/2001/di/public/as/as-dreaft-20020729.html>
- [BEA03] BEA Systems. 2003. <http://www.bea.com>
- [W3C02b] CC/PP Structure and Vocabularies, 2nd last Call WD. Noviembre 2002
<http://www.w3.org/TR/2002/WD-CCPP-struct-vocab-20021108>
- [W3C02c] Delivery Context Overview for Device Independence W3C. Diciembre 2002. <http://www.w3.org/TR/2002/WD-di-dco-20021213/>
- [W3C01] Device Independence Principles. W3C. Septiembre 2001
<http://www.w3.org/TR/2001/WD-di-princ-20010918/>
- [W3C03] Extensible Markup Language (XML)
<http://www.w3.org/XML/>
- [Ian98] An Idiot's Guide to the Resource Description Framework. Renato Iannella. 1998
<http://archive.dstc.edu.au/RDU/reports/RDF-Id>
- [But01] Implementing Content Negotiation using CC/PP and WAP UAProf. Mark H. Butler Information Infrastructure Laboratory. HP Laboratories Bristol. Agosto de 2001
- [HP02] An Introduction to RDF and the Jena RDF API. HP. Abril de 2002
http://www.hpl.hp.com/semweb/doc/tutorial/RDF_API/index.html
- [Jak03] Jetspeed. Jakarta 2003. <http://jakarta.apache.org/jetspeed>
- [Mil02] Milenium. Informática Milenium. 2002
<http://www.informaticamilenium.com.mx/index.html>
- [Mob03a] MobileAware. 2003. <http://www.mobileaware.com>
- [Mob03b] MobileWebSurf. 2003. <http://www.mobilewebsurf.com>
- [Tro01] La Oficina Web. Roberto Troncoso. Taboo Ediciones S.L. 2001
- [Por03] Portals Community 2003. <http://www.portalscommunity.com>
- [W3C99] RDF Model and Syntax Specification. W3C. Febrero de 1999
<http://www.w3c.org/TR/REC-rdf-syntax>
- [SB02] A test client API for CC/PP and UAProf. Charles Smith, Mark H. Butler. Information Infrastructure Laboratory. HP Laboratories Bristol. octubre de 2002.
- [W3C99] XML Path Language (XPath). Version 1.0. Noviembre 1999
<http://www.w3.org/TR/1999/REC-xpath-19991116>
- [WAG99] WAG UAPROF. User Agent Profile Specification. Wireless Application Group. Version 10 de noviembre de 1999
<http://www.wapforum.org/what/technical/SPEC-UAProf-19991110.pdf>
- [RFC99] Hypertext Transfer Protocol – HTTP 1.1. Request For Comments 2616. The Internet Society. 1999. <http://www.w3c.org/Protocols/rfc2616.html>
- [Vol03] Volantis systems. 2003. <http://www.volantis.com>
- [Web03] Webraska Mobile Technologies. 2003. <http://www.webraska.com>
- [Ope03] OpenCMS. 2003. <http://www.opencms.org>
- [Ora03] Oracle. 2003. <http://www.oracle.com>
- [PHP03] PHPNuke. 2003. <http://www.phpnuke.org>
- [Pos03] PostNuke. 2003. <http://www.postnuke.org>