EUSKO JAURLARITZA GOBIERNO VASCO

– Opendata Euskadi –

Portal de Datos Abiertos del Gobierno Vasco

Denominación:	Infraestructura de entrega de contenidos **Documentación Técnica**
Autor	**Departamento de Justicia y Administración Pública del Gobierno Vasco** Dirección de Atención Ciudadana

Marzo 2010

Contenido

Capítulo/sección		Página
1	Introducción	3
2	Premisas de la Infraestructura	4
3	Estructura de un portal y sus URLs	8
3.1	Estructura de directorios en el servidor web	8
3.2	Estructura de directorios de un portal	9
3.3	Estructura de directorios de un repositorio de contenidos	11
3.4	Estructura de URLs	15
3.4.1	URL directa de un contenido o una página de portal	15
3.4.2	Combinar un contenido en una página de portal	16
3.4.3	Repositorios	18
3.4.4	Entrega de páginas de portal	20

1 Introducción

El presente documento contiene un resumen del la infraestructura de entrega de contenidos de euskadi.net.

Dado que el portal opendata.euskadi.net se basa íntegramente en la infraestructura de euskadi.net, es importante conocer cómo funciona euskadi.net desde dos puntos de vista:

Las premisas de la infraestructura	Premisas arquitectónicas y funcionales en las que se basan todos los contenidos y portales de la red euskadi.net
Las URLs	Las URLs de euskadi.net son importantes ya que son un elemento común en toda la red y por lo tanto también en opendata.euskadi.net; conocer su funcionamiento es importante-

Audiencia: Analistas y Desarrolladores

Complejidad: Media / Alta

2 Premisas de la Infraestructura

Toda la arquitectura de las Herramientas de Soporte al Modelo de Presencia en Internet del Gobierno Vasco, se basa en las siguientes **premisas arquitectónicas / tecnológicas:**

Separación entre contenidos y portales

Desde el punto de vista **organizativo**

Las personas que gestionan portales y las que gestionan contenidos son **diferentes**, con diferentes roles y diferentes funciones.

- Los gestores de contenidos centran su actividad en **producir información** destinada a ser publicada en Internet, pero sin ser conscientes de ello: no saben en qué portal o página va a aparecer finalmente la información.
- Los gestores de portales centran su actividad en **referenciar en páginas de portal** los contenidos que se han ido creando en toda la organización descentralizadamente. Así su misión es componer las páginas de portal organizando de la manera más adecuada el acceso a los contenidos.

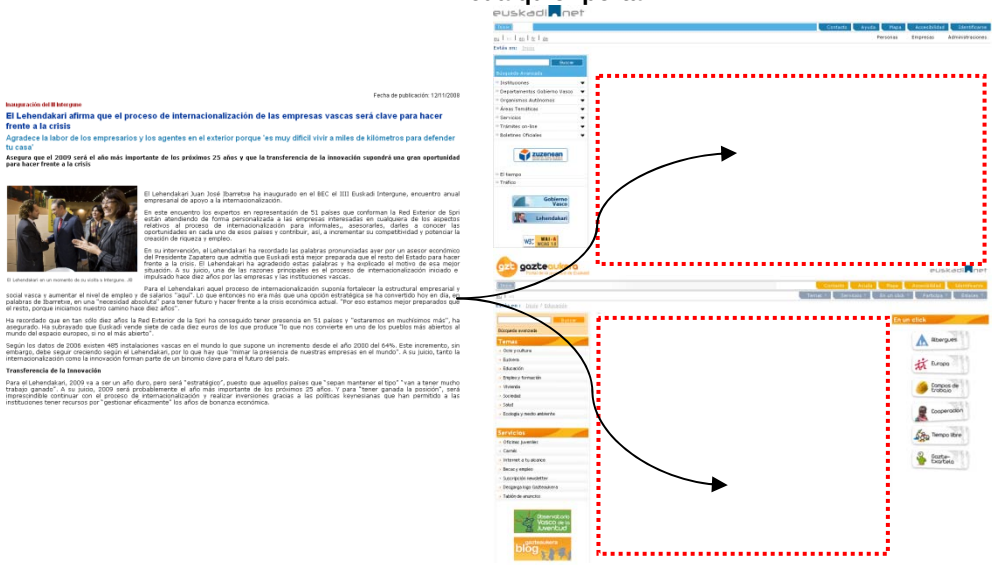
Desde el punto de vista **tecnológico**

Las herramientas de gestión de contenidos y de gestión de portales están completamente **separadas**.

- La herramienta de gestión de contenidos permite crear, revisar y publicar información de diversa tipología de forma sencilla y sin necesidad de conocimientos técnicos.
- La herramienta de gestión de portales permite configurar páginas de portal: su apariencia y los contenidos / búsquedas a las que dan acceso sin necesidad de conocimientos técnicos.

Desde el punto de vista del **uso**

Una consecuencia de la separación organizativa y tecnológica es la posibilidad de **re-utilizar cualquier contenido de cualquier portal**.



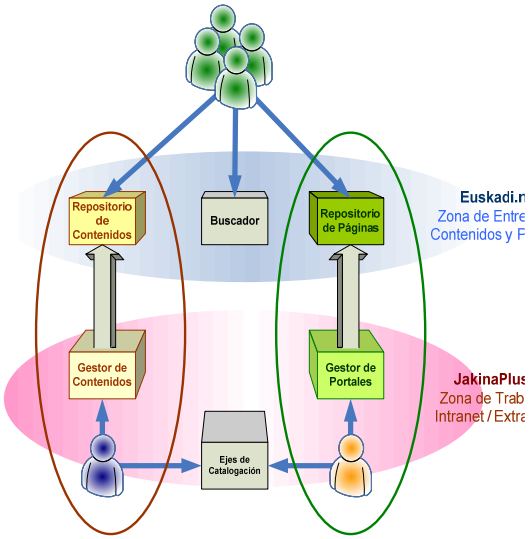
En la figura se muestra como el portal se encarga de “vestir” el contenido que se encuentra en un sistema separado e independiente.

Generación en estático

Con los objetivos de obtener mayor fiabilidad, rapidez y minimizar la complejidad técnica en entrega, tanto los contenidos como las páginas de portal se **generan en HTML estático que es servido únicamente por un servidor web (apache)**.

La intervención de los servidores de aplicaciones (Weblogic) se limita a las partes dinámicas de los portales: ejecución de búsquedas y ejecución de aplicaciones.

La generación de HTML se hace en una **zona interna** de trabajo, mientras que la visualización se hace en una **zona pública** de Internet:



En horizontal se muestran las zonas de seguridad y los componentes de cada una

En vertical se muestran las dos herramientas principales: **gestión de contenidos** y **gestión de portales** y cómo el **buscador** y la **herramienta de catalogación** hacen de "unión" entre ambos mundos.

En la figura se muestra como la infraestructura se divide en **dos zonas** (en horizontal)**:**

- **Zona de Trabajo Intranet/Extranet**: Jakinaplus

  En esta zona es donde se encuentran las herramientas de gestión de contenidos y portales que permiten a los gestores la creación de contenidos y la configuración de páginas de portal

  Una vez que un contenido / página de portal se ha terminado de elaborar se **genera en HTML estático** que se **despliega** en la Zona de Entrega.

- **Zona de Entrega de Contenidos y Portales:** euskadi.net

  A esta zona es donde acceden los visitantes / ciudadanos.

  No hay instalada ninguna aplicación de gestión de contenidos ya que estos están **desplegados en el web de forma estática** (ficheros HTML).

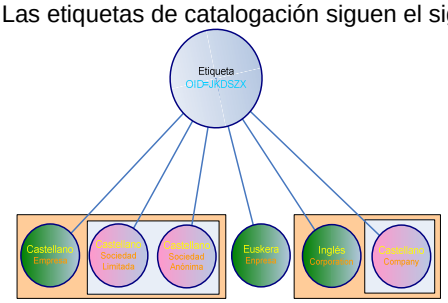
  La única aplicación es el buscador.

A grandes rasgos, el funcionamiento del sistema se resume en que en la zona de trabajo intranet/extranet, los gestores de contenidos / portales **generan HTML y adjuntos** que son **desplegados** a la zona de entrega Internet donde son servidos **únicamente** por los servidores web.

Catalogación en etiquetas

Todos los contenidos y páginas de portal se **catalogan** (etiquetan) utilizando unas **etiquetas de catalogación** que se organizan en forma de **taxonomías normalizadas** (árboles que relacionan unas etiquetas con otras)

Las etiquetas de catalogación siguen el siguiente modelo:



Una etiqueta no es más que un identificador sin significado (oid) que por detrás esconde una serie de **términos** en diferentes idiomas.

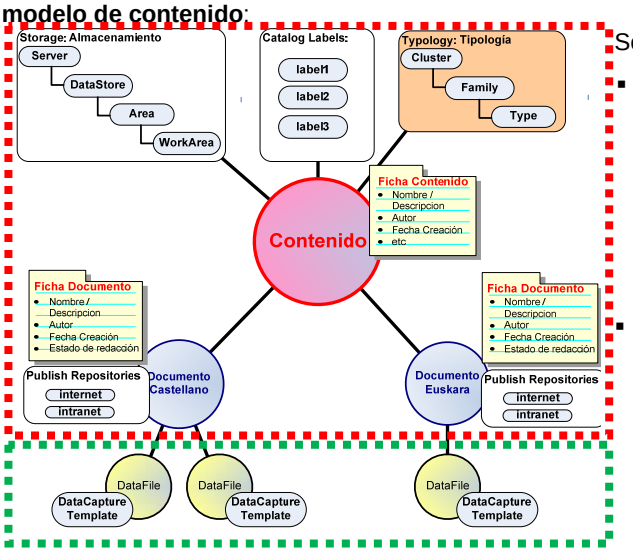
- En cada idioma se designa un **término normalizado** en cada idioma que es el que se utiliza para visualizar la etiqueta en las aplicaciones, etc
- Además en cada idioma hay **n términos sinónimos** que se pueden utilizar en condiciones de búsqueda

El objetivo básico de las etiquetas de catalogación es aportar al contenido/página de portal información (meta-data) **subjetiva** del autor y que posiblemente no se encuentra en el contenido; todo en beneficio de las búsquedas del portal.

Ejemplo: *Un contenido puede estar destinado a ciudadanos, siendo este un criterio que aporta el autor (subjetivo) ya que puede que no se mencione la palabra "ciudadano" en un contenido. En este caso, el autor catalogará el contenido con la etiqueta "ciudadano" del eje "destinatarios"*

Modelo único de contenido

Toda la arquitectura de soporte al Modelo de Presencia en Internet del Gobierno Vasco se basa en el **modelo de contenido**:



Se distinguen **dos tipos de metadatos**

- **MetaDatos comunes**: aquellos que tienen **todos** los contenidos, **independientemente de su naturaleza**

  Ej: *Todas las noticias, eventos, informaciones, ayudas, etc tienen asociado un nombre, una descripción, una tipología y en general una serie de metaDatos comunes.*

- **MetaDatos específicos: aquellos propios del contenido en función de su naturaleza** (tipología de contenido)

  Ej: *Las noticias tienen unos metaDatos particulares como la fecha de la noticia o el titular. Los procedimientos administrativos a su vez, tienen unos metaDatos particulares como el objeto, el estado de vigencia o el lugar de presentación.*

Contenido	El contenido es la **unidad mínima** gestionada en el sistema y es un conjunto de información con significado propio en función de la información que facilita y su pertenencia a una misma materia, área u otro criterio de agrupación. Metadatos del contenido son: • Dónde se almacena el contenido en el Gestor de Contenidos • Etiquetas de catalogación • Tipología: naturaleza del contenido (noticia, evento, etc) • Otros metadatos comunes (nombre, descripción, autor) El contenido físicamente es una carpeta que contiene a su vez carpetas para cada uno de los documentos / versiones que alojan los ficheros correspondientes a cada versión
Documento / Version	Un documento es una **versión** de un contenido, bien sea por idioma, destinatario, público objetivo, etc: El documento contiene toda la información del contenido expresada en un idioma y con una narrativa y estilos específicos para un destinatario (jóvenes, empresas, especialistas, etc) Ej: *Un contenido puede expresarse en castellano y con una narrativa dirigida los jóvenes o bien expresarse en castellano con una narrativa más técnica para expertos.* El documento tiene a su vez una serie de metaDatos propios: • Nombre, descripción, autor, etc • Repositorio público (de Euskadi.net) donde se ha publicado (una versión de un contenido puede publicarse en varios repositorios públicos) Físicamente un documento es una carpeta que contiene todos los ficheros del documento: es una unidad autocontenida en el sentido que "encapsula" todos los recursos necesarios para el propósito que fue creado: ficheros HTML, videos, imágenes, hojas de estilo, scripts, etc.
Archivo de Datos	Un contenido /vdocumento en sí mismo son entidades lógicas, sin embargo, los documentos contienen **archivos de datos**, adjuntos, imágenes, etc que es donde realmente está la información a ofrecer. Un archivo de datos es por lo tanto un fichero capturado utilizando una **plantilla de introducción de datos** y que contiene los **metaDatos específicos** (aquellos que no son los comunes expuestos con anterioridad). Ejemplo: *Las noticias se "capturan" utilizando una plantilla de introducción de datos ad-hoc que permite al usuario introducir metaDatos específicos de las noticias: fecha de la noticia, titular, entradilla, etc.* Los ficheros de datos se guardan como XML, sin embargo, para su publicación en Internet se "generan" en HTML, utilizando **plantillas de presentación**.

Modelo único de portal

Todas las páginas del portal Euskadi.net se han construido en base a una herramienta común basada en **componentes reutilizables** llamados **áreas visuales** (AV) en un concepto similar a los *portlets* con la diferencia de que se generan en forma de HTML estático (no necesitan de servidores de aplicaciones).

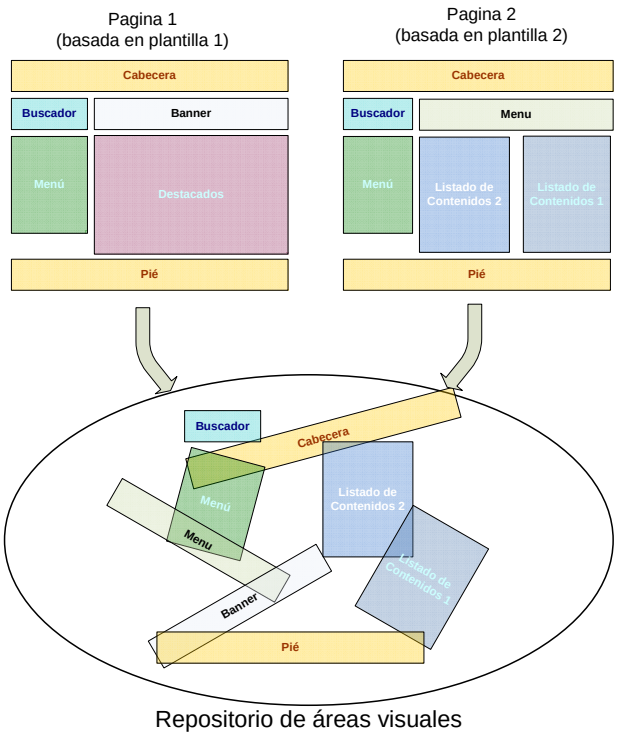
Las páginas de portal se estructuran en base a **áreas visuales** (AV) reutilizables en cualquier página del portal

Ejemplo: *Un menú general del portal, la cabecera o el pié habitualmente son los mismos en todas las páginas del portal*

Se han diseñado decenas de áreas visuales: menús horizontales, verticales, detallados, de imágenes, zonas de destacados, pié de página, imagen, banner, selector de idioma, migas de pan, buscador, resultados de búsqueda, visualizador de contenidos / aplicaciones, etc.

Cada área visual tiene una **administración** que permite parametizar aspectos **visuales** (ej: colores, letras, etc) y **funcionales** (ej: las opciones de un menú o las imágenes de un banner)

Dado que las AVs se generan como ficheros HTML independientes y se pueden reutilizar en múltiples páginas, la **composición o disposición de las áreas visuales en la página se hace en base a plantillas de página** que también son ficheros HTML estáticos servidos únicamente por un servidor web.



Repositorio de áreas visuales

En la figura se muestra cómo se componen dos páginas con distinta disposición (diferente plantilla) en base a las AVs independientes del repositorio común del portal

3 Estructura de un portal y sus URLs

Como ya se ha señalado en el punto anterior, dos de las premisas de la infraestructura son:

1. **Generar todos los contenidos y páginas de portal en forma de HTML estático entregado por un servidor web.**
2. **Separar los contenidos / aplicaciones de las páginas de portal**

Estas dos premisas tienen dos implicaciones técnicas:

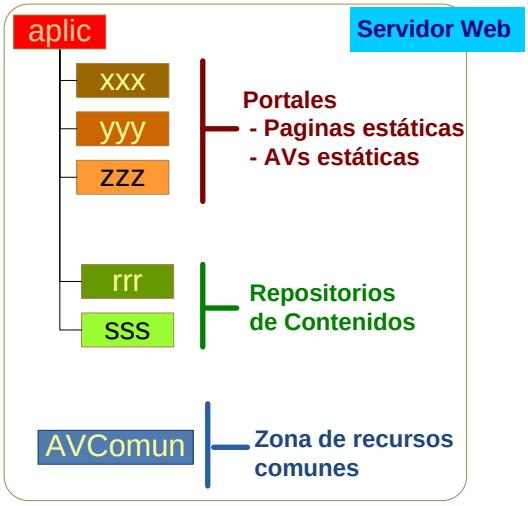
Los contenidos y páginas de portal se generan en forma de HTML estático que se sirve por un servidor web ⇨ Hay que **definir una estructura de directorios** donde depositar los ficheros HTML (y sus recursos asociados: imágenes, hojas de estilo, scripts, etc)

Los contenidos y las páginas de portal están separadas ⇨ Hay que definir una estructura de URLs para mostrar un contenido dentro de una aplicación.

3.1 Estructura de directorios en el servidor web

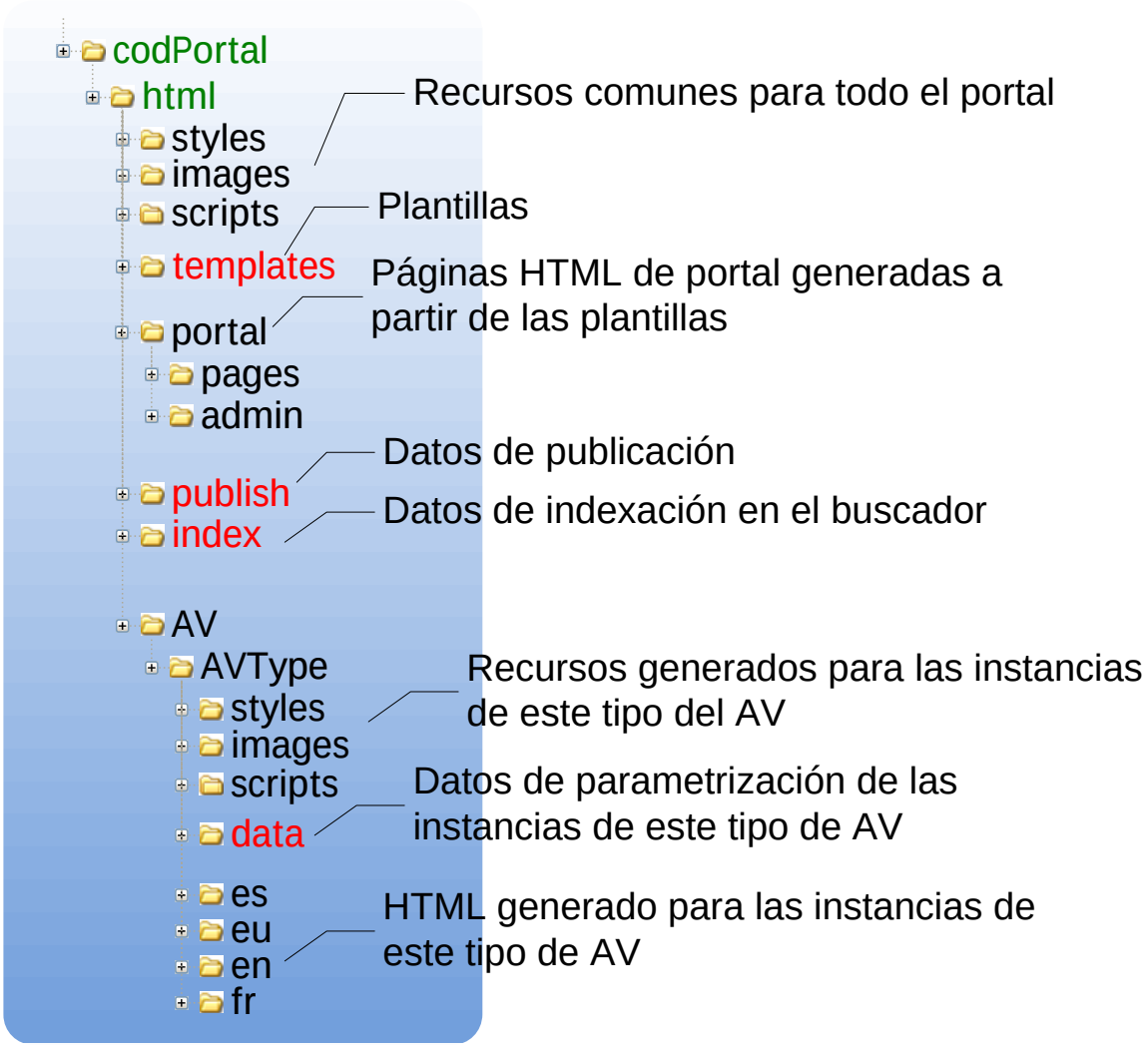
Dada la separación de contenidos y páginas de portal, en el servidor web debe existir tal y como se muestra en la figura:

- Una zona para las **páginas de cada portal**
- Una zona para los **contenidos de cada repositorio**
- Una zona para **recursos comunes** a páginas y contenidos (ej: imágenes, hojas de estilo, scripts, etc)



3.2 Estructura de directorios de un portal

A cada portal se asigna un código de aplicación cuya estructura de directorios es la que sigue:

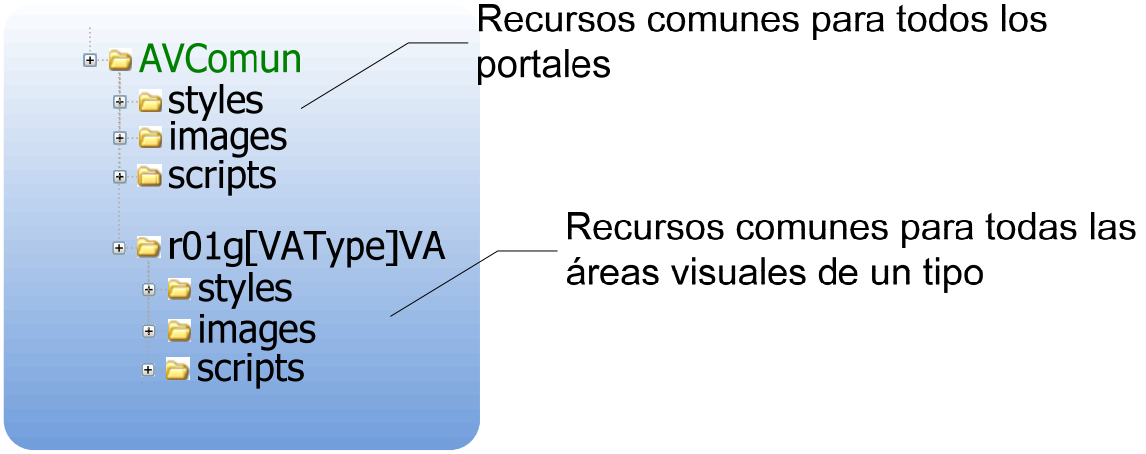


Como se puede observar, debajo de la carpeta del portal se almacenan todos los recursos (estáticos) del portal:

- Plantillas de página
- Páginas HTML generadas (en versión final y en versión de administración)
- Áreas visuales generadas: HTML por idioma y recursos visuales (imágenes, scripts y hojas de estilo)
- Recursos visuales de las páginas del portal (imágenes, estilos y scripts)
- Datos de publicación
- Datos para indexación en el buscador

La misión del gestor de portales es generar el código estático a partir de los datos (parametrización) del usuario de forma que no sea necesario un servidor de aplicaciones (java, php, ruby, etc) para entregar contenidos al navegador web: las páginas de portal están pre-generadas y únicamente es necesario

Todos los portales **comparten** una **zona común** donde se depositan recursos comunes a todos los portales. Esta zona se denomina **AVComun** y tiene la siguiente estructura:



Para hacer referencia a un recurso de la zona común, dado que AVComun es un **alias web** que existe en todos los portales de la infraestructura, basta con poner: **/AVComun/[path relativo recurso]**

3.3 Estructura de directorios de un repositorio de contenidos

A cada repositorio de publicación se asigna un código de aplicación cuya estructura de directorios es la que sigue:

Haciendo zoom en un contenido, básicamente es:

Un contenido es una carpeta que almacena a su vez una carpeta con las versiones del contenido (documentos)

Se intenta que tanto los contenidos como los documentos sean **autocontenidos**, es decir, almacenen todos los recursos necesarios para su visualización (hojas de estilo, imágenes, scripts, etc).

Sin embargo, en ocasiones, es **muy redundante** tener por ejemplo la misma imagen en el documento (carpeta) correspondiente a la versión en castellano y en el documento (carpeta) correspondiente a la versión en euskera.
Para evitar tener que tener múltiples copias de recursos (imágenes, scripts, estilos, media, etc) en todas las versiones de un contenido (documentos), o en varios contenidos, se pueden utilizar:

- **Documentos de recursos:** son documentos "normales" del contenido que únicamente tienen recursos compartidos por los otros documentos del contenido.

  Se pueden tener documentos de recursos para:
  - Todas las versiones idiomáticas de un contenido: *r01res_[doc/versión]*
  - General para todos los documentos del contenido: *r01res*

Es **fundamental** que las referencias entre recursos de un documento (versión del contenido) sean **relativas.**

Por ejemplo, cuando desde la página HTML se hace una referencia a una imagen del propio documento esta debe ser:

en lugar de ser:

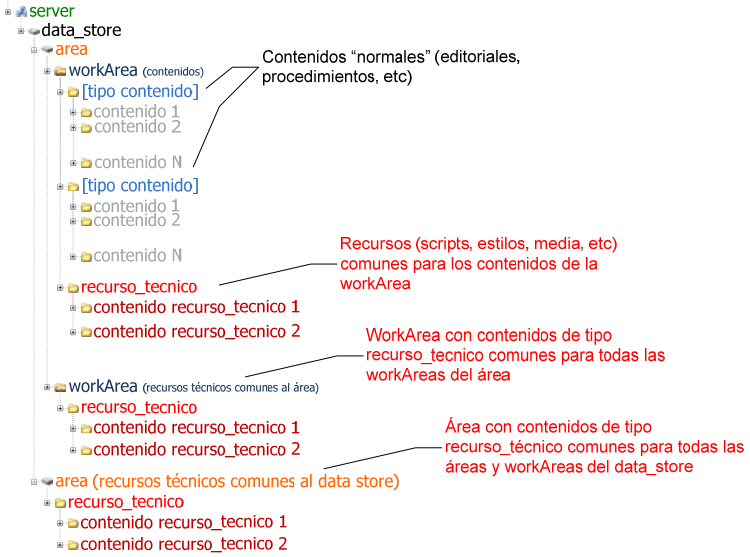
Si se hace una referencia a un recurso de un documento de recursos del contenido, la referencia debe ser:

'

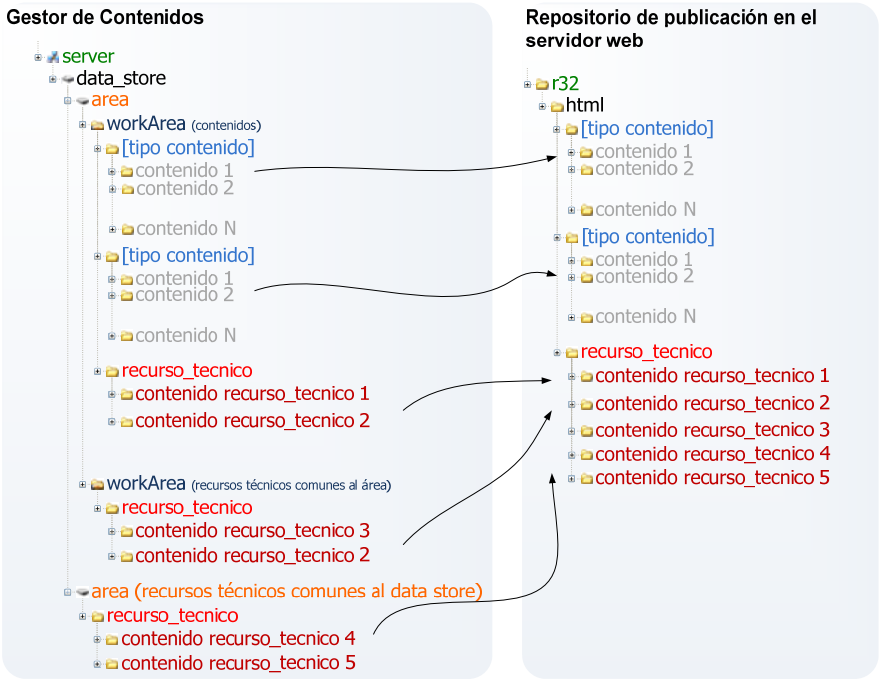
el problema es que un contenido puede ser publicado en **cualquier repositorio** como /contenidos e incluso en varios. Si se hacen referencias absolutas al repositorio, puede ser que estas referencias no funcionen.

- **Contenidos de recursos:** son contenidos de tipo recurso_tecnico que únicamente tienen recursos compartidos por un conjunto de contenidos

  Se podrían tener contenido de recursos para todos los contenidos de una workArea, de un área, etc tal y como se puede ver en la figura:



  teniendo en cuenta dónde se publican los recursos técnicos en un repositorio en el servidor web:



Es **fundamental** que las referencias a recursos comunes sean **relativas** utilizando ..

Por ejemplo, cuando desde la página HTML se hace una referencia a una imagen de un recurso técnico común debe ser:

en lugar de ser:

el problema es que un contenido puede ser publicado en **cualquier repositorio** como /contenidos e incluso en varios. Si se hacen referencias absolutas al repositorio, puede ser que estas referencias no funcionen.

Con todo esto, las referencias a recursos comunes deben ser

Recursos comunes a un contenido	r01res_[doc]/[url relativa] **ej:** <img href='../r01res_doc/[url_relativa]'>
Recursos comunes a varios contenidos	../../recurso_tecnico/[content]/r01res_[doc]/url_relativa **ej**: <img href='../../recurso_tecnico/r01res_doc/url_relativa'>

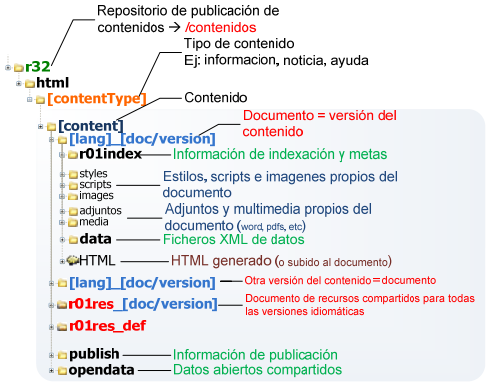
3.4 Estructura de URLs

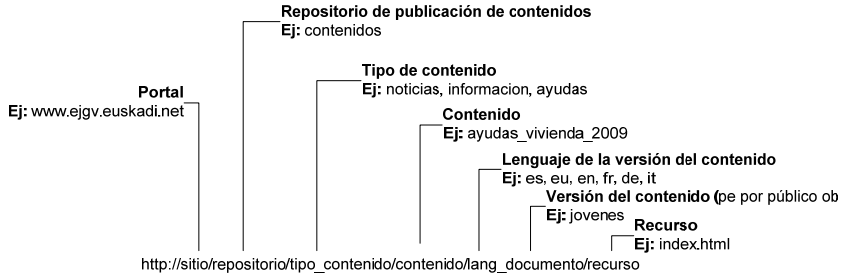
Con lo señalado anteriormente, se puede ahora entender la estructura de las URLs de la infraestructura.

3.4.1 URL directa de un contenido o una página de portal

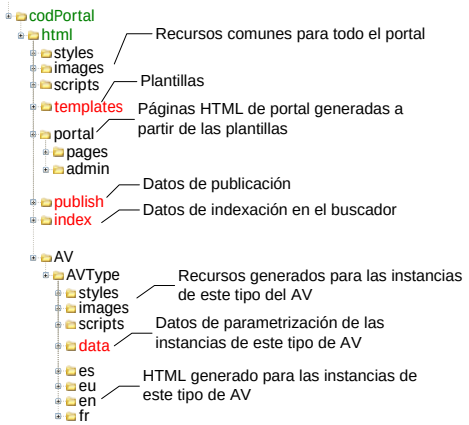
Una característica de la infraestructura es que los contenidos y las páginas de portal están **separados** y se unen en tiempo de visualización, de forma que un mismo contenido puede ser visualizado en en cualquier página de cualquier portal.

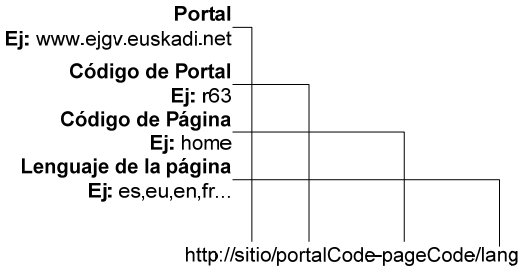
Para acceder **directamente** a un **contenido** la URL es:





De igual forma, para acceder **directamente** a una **página de portal** la URL es:





NOTA: A diferencia de la URL de un contenido, la URL de una página de portal NO refleja la estructura de directorios del portal (ver izq). Para "traducir" de la URL a la estructura física en el sistema de ficheros se utilizan reglas de re-escritura (mod_rewrite) en apache.

Es importante señalar que **las páginas HTML no se generan por idioma**, son las áreas visuales las que tienen versión idiomática, lo cual tiene todo el sentido ya que son las AVs las que contienen los datos.

Esto significa que a partir de una plantilla se genera una página que es capaz de visualizarse en cualquier idioma, siempre que las AVs de esta página estén configuradas en dicho idioma.
A partir de la url solicitada: http://sitio/portalCode-pageCode/lang:

1. Se localiza la página SHTML.
2. Dado que la página es un fichero **S**HTML, es procesada por el filtro SSI (*Server Side Include*) del servidor web que "**incluye al vuelo**" otros ficheros (las AVs) en el idioma solicitado devolviendo un todo único que es lo que llega al navegador como página HTML completa.

   Recordar que las AVs si son idiomáticas, es decir, hay un HTML independiente para cada idioma *(ver gráfico)*

Cada portal de la red de euskadi.net corresponde a un **sitio web independiente** configurado en los servidores web de la Infraestructura.

Los enlaces entre páginas y contenidos de portal se hace utilizando **URLs relativas** o **absolutas** (en último extremo), pero **nunca URLs completas** que incluyan http://sitio/... Esto es importante de cara a facilitar los cambios de nombre de dominio.

De esta forma, durante la navegación, un usuario podría entrar en una página de euskadi.net, por ejemplo www.euskadi.net/r33-2220/es/ y pinchar en un link que le lleve a una página de la web del Gobierno vasco, por ejemplo www.ej-gv.net/r53-2283.

> El usuario ha entrado a www.euskadi.net/r33-2220/es y pinchado en un enlace como a href='/r53-2283/es' (un enlace a otro portal)
> Esto va a hacer que al usuario le aparezca en el navegador www.euskadi.net/r58-2283/es, el enlace funcionará pero **se pierde el nombre de dominio:** continúa apareciendo el dominio por el que entró: www.euskadi.net, cuando debería aparecer el nombre de dominio del Gobierno Vasco www.ejgv.euskadi.net

Para evitar este comportamiento, se utiliza una re-escritura de la url en el servidor web, que se basa en **comprobar que el nombre del dominio coincide con el código de aplicación** del portal.

Sitio web	Portal - Página	Idioma	Recurso

Para realizar esta comprobación se utiliza un fichero de configuración en el servidor web ***vhost.euskadi.map*** que relaciona sitios web (portales) con su código de aplicación.
Si se detecta que se accede a un sitio web y el código de portal no corresponde con el que debería ser, el sistema automáticamente redirige al sitio web correcto.
Por lo tanto, **es muy importante que cuando se crea un nuevo portal se introduzca la relación en este fichero**.

3.4.2 Combinar un contenido en una página de portal

Dado que los contenidos y las páginas de portal son independientes, se utiliza un mecanismo basado en la URL para "combinar" un contenido y una página de portal:

Las URL de Euskadi.net tienen una **estructura uniforme**:

Sitio web	Portal - Página	Idioma	Recurso

URL de un contenido visualizado en una página de portal (**la url depende del portal y la página)**

Todas las URL, independientemente de que se trate de un contenido o una aplicación siguen este esquema:

URL de Contenido en una página de portal

http://www.euskadi.net/r33-2202/es/contenidos/informacion/es_12/indice.html

Sitio Web | Portal - Página | Idioma | Contenido

URL de Aplicación Java en una página de portal

http://www.euskadi.net/r33-2202/es/s03aWar/s03aMainServlet?param1=a&param2=b

Sitio Web | Portal - Página | Idioma | Aplicación

URL de Aplicación ASP en una página de portal

http://www.euskadi.net/r33-2202/es/meteo/meteo1.apl?param1=a&param2=b

Sitio Web | Portal - Página | Idioma | Aplicación

La herramienta de portal es capaz de visualizar el mismo contenido / aplicación en cualquier otro portal con solo cambiar la parte sitio/portal-página por la correspondiente en el otro portal.

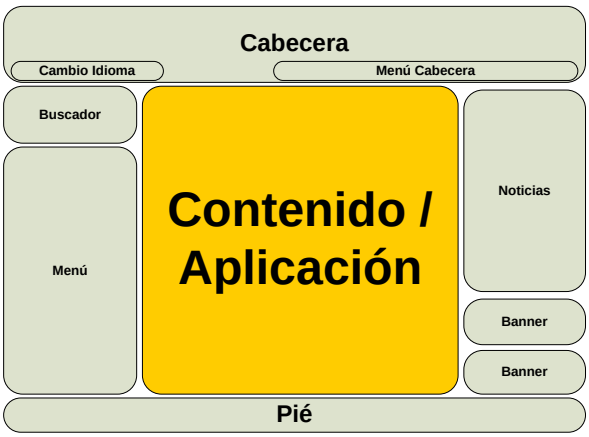
En los ejemplos anteriores, si se desea mostrar el contenido o aplicaciones en el portal de industria, la URL sería:

http://www.industria.ejgv.euskadi.net/**r58-1237**/es/contenidos/informacion/es_12/indice.html

http://www.industria.ejgv.euskadi.net/**r58-1237**/es/s03aWar/s03aMainServlet?param1=a&param2=b

http://www.industria.ejgv.euskadi.net/**r58-1237**/es/meteo/meteo1.apl?param1=a&param2=b

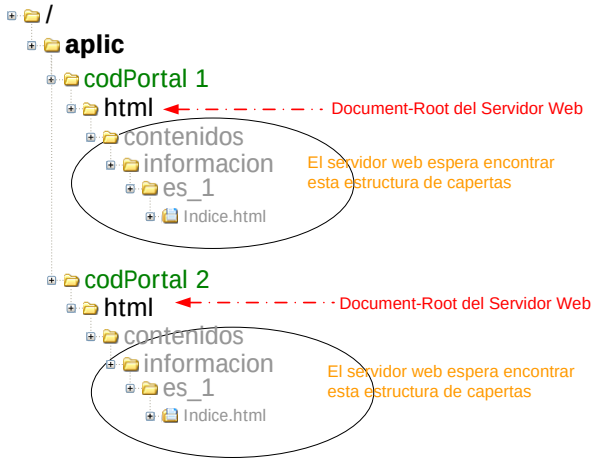
Para que una página sea capaz de mostrar un contenido o una aplicación, es necesario que esté preparado para ello, es decir, la página ha de tener un **contenedor de contenido** / **aplicación**.



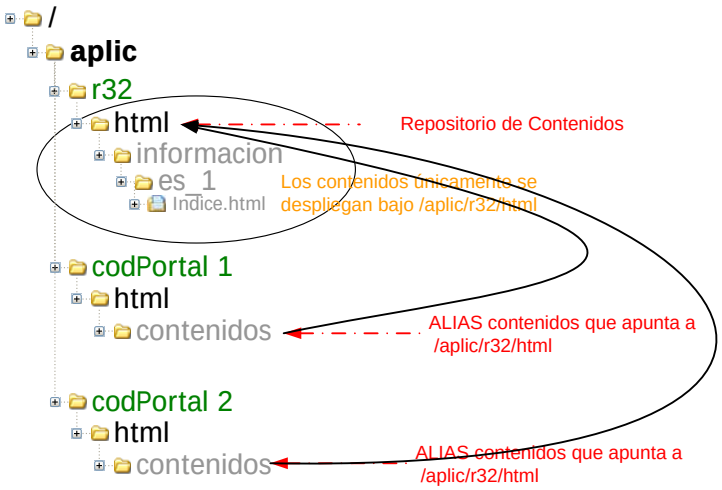
3.4.3 Repositorios

3.4.3.1 Repositorio de publicación de contenidos

Cada portal tiene su propio código de aplicación y su propio web-root bajo /aplic/[codigo Aplicación]/html, sin embargo, cuando se intenta acceder a un recurso estático debajo de un código de portal (por ejemplo http://sitio/r33-2202/es/***contenidos***/informacion/es_12/indice.html), el servidor web intenta buscar la carpeta ***contenidos***/…debajo del web-root del portal:



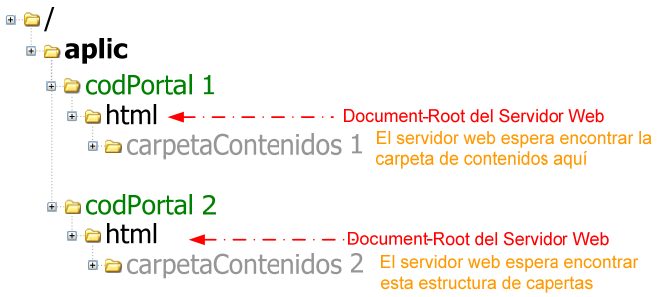
El repositorio "contenidos" almacena todos los contenidos publicados en euskadi.net y obviamente **no** pueden replicarse para cada uno de los portales de la red. Para solucionar el problema, se utiliza un **alias web** que "simula" que el repositorio "***contenidos"*** existe en todos los webs de la red



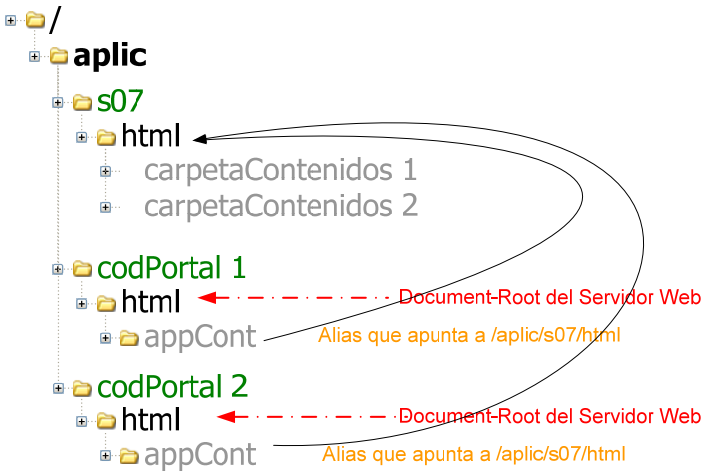
Este "truco" permite que cualquier portal de la red "vea" el repositorio de contenidos de euskadi.net y en principio cualquier otro repositorio de datos que se pueda ir creando.

3.4.3.2 *Repositorios de recursos de aplicación*

El problema con las aplicaciones es muy similar al de los contenidos: todas las aplicaciones tienen elementos estáticos (JavaScript, hojas de estilo, imágenes, etc) que hasta ahora se ponían en una carpeta de contenidos bajo el web-root del portal:



Esto impide que estos recursos estáticos estén visibles para cualquier otro portal de la red. Para solucionar el problema se ha creado un **código de aplicación común** (s07) donde se colocan **todas** las carpetas de contenido estático de las aplicaciones. A es /aplic/s07/html apunta el alias web **/appCont** que está configurado en todos los portales de la red, de forma que todos ellos "ven" los contenidos estáticos de todas las aplicaciones.



De esta forma, anteriormente los contenidos estáticos en una aplicación se referenciaban como:

http://www.euskadi.net/carpetaContenidos/elContenidoEstatico

ahora se referencian a través de un alias que está configurado en todos los portales:

http://www.euskadi.net/**appCont**/carpetaContenidos/elContenidoEstatico

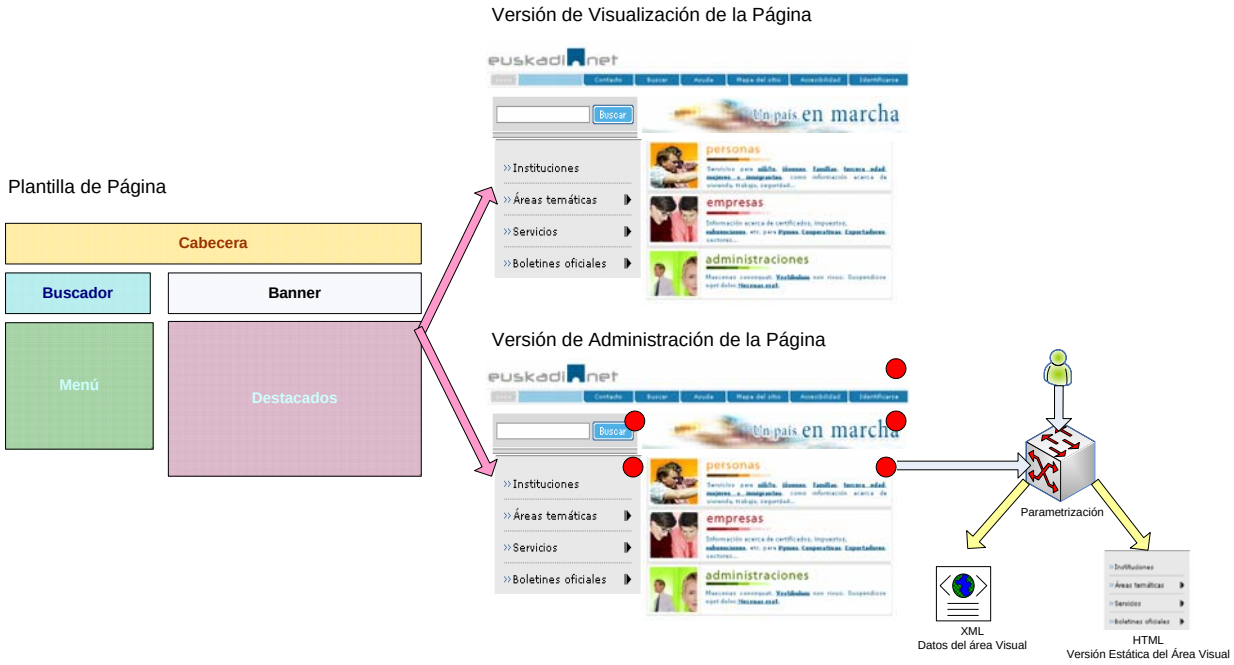
http://www.industria.ejgv.euskadi.net/**appCont**/carpetaContenidos/elContenidoEstatico

3.4.4 Entrega de páginas de portal

La herramienta de Gestión de Portales se basa en los siguientes elementos:

Área Visual	Elemento de una página web parametrizable en : → Su aspecto visual → La información que muestra → En su comportamiento.
Plantilla de Página	Disposición concreta de áreas visuales que permite la creación de páginas con la misma disposición.
Página	Instancia concreta de una plantilla de página en la que se han parametrizado los valores de las áreas visuales.

El proceso de creación de una página se muestra en la siguiente figura:



Como se puede ver en la figura, a partir de una plantilla de página se generan **dos versiones de la página**:

Versión de Visualización: Versión estática de la página y sus áreas visuales tal y como se va a ver cuando sea publicada.

Versión de Administración: Versión de la página que permite a los Gestores Web acceder a la parametrización de las áreas visuales

NOTA: Recordar que las páginas son **independientes del idioma**, es decir, pueden visualizarse en cualquier idioma siempre que las AVs que componen la página estén configuradas en dicho idioma

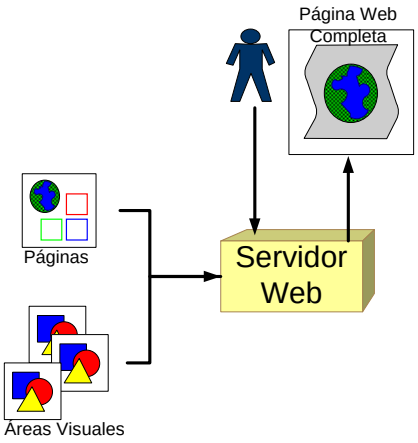
Al parametrizarlas, las áreas visuales se almacenan de dos formas:

Datos en XML: Datos de parametrización en formato XML

Versión HTML generada: Versión estática (html) generado a partir de los datos de parametrización.

El Sistema de Gestión de Portales se basa en **ficheros estáticos (HTML generado)**: páginas SHTML que contienen referencias a las áreas visuales que a su vez son ficheros HTML.

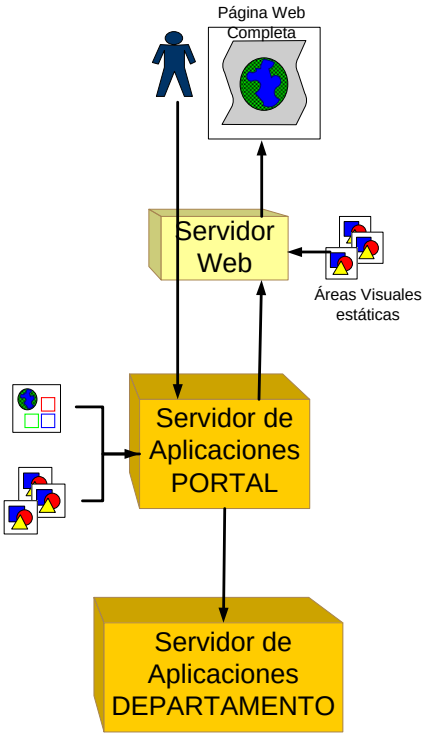
La composición de las páginas y las áreas visuales se realiza mediante **reglas de re-escritura** en el servidor web apache de forma que al cliente (navegador del usuario) le llega una página completa.



El servidor web localiza la página SHTML que contiene referencias a las AVs y genera "al vuelo" una página HTML completa de forma que el navegador web "no se entera" de que la página y las AVs existían separadas como ficheros independientes.

Si se intenta visualizar un **contenido** el servidor web también es capaz de "inyectarlo" en el HTML final.

Sin embargo, **no todas las páginas** pueden ser estáticas. Hay ocasiones en las que la página **integra aplicaciones**, que necesitan procesamiento en el servidor de aplicaciones. En este caso el funcionamiento es el de la figura:



Cuando se detecta que se está visualizando una aplicación integrada,

1. El servidor web "detecta" que se está integrando una aplicación y "delega" la petición al servidor de aplicaciones (weblogic) de portal.
2. El servidor de aplicaciones de portal (weblogic) carga la página (SHTML) y la analiza en busca del área visual de contenido/aplicación donde hay que "inyectar" la aplicación
3. Cuando se encuentra el AV de contenido/aplicación, se "delega" a su vez la petición al servidor de aplicaciones departamental que ejecuta la aplicación
4. El servidor de aplicaciones departamental devuelve el HTML de la aplicación y el servidor de aplicaciones de portal lo "inyecta" dentro del AV de contenido/aplicación de la página SHTML y la devuelve al servidor web.
5. Al servidor web le llega una página SHTML en la que se ha "sustituido" el AV de contenido/aplicación por el HTML devuelto por el servidor de aplicaciones departamental pero en el que aún hay referencias al resto de AVs NO dinámicas (cabecera, pié, menús, etc)
6. Finalmente las áreas visuales estáticas (menús, cabeceras, piés, etc) son completadas en el servidor web.