



Eusko Jaurlaritzaren Informatika Elkarte  
Sociedad Informática del Gobierno Vasco

# EJIE

## Presencia en Internet

Guía de Desarrollo en la Infraestructura de Portales

Fecha:

Referencia:

**EJIE S.A.**

Mediterráneo, 3  
Tel. 945 01 73 00\*  
Fax. 945 01 73 01  
01010 Vitoria-Gasteiz  
Posta-kutxatila / Apartado: 809  
01080 Vitoria-Gasteiz

[www.ejje.es](http://www.ejje.es)

## Control de documentación

Título de documento: Guía de Desarrollo en la Infraestructura de Portales

### Histórico de versiones

Código:

Versión:

Fecha:

Resumen de cambios:

### Cambios producidos desde la última versión

Primera versión.

### Control de difusión

Responsable:

Aprobado por:

Firma:

Fecha:

Distribución:

### Referencias de archivo

Autor: Alberto Doctor Gallo y Alex Lara

Nombre archivo:

Localización:

## Contenido

	Capítulo/sección	Página
1	Introducción	1
2	Conceptos Previos	2
2.1	Premisas de la Infraestructura	2
2.2	Estructura de un portal y sus URLs	6
2.2.1.	Estructura de directorios en el servidor web	6
2.2.2.	Estructura de URLs	13
2.2.3.	Repositorios	16
2.3	Entrega de páginas de portal	18
3	Hojas de Estilo y Scripts para Páginas de Portal, Contenidos y Aplicaciones	20
3.1	Conceptos Generales sobre hojas de estilo	20
3.1.1.	Inclusión de Hojas de Estilo	20
3.1.2.	Selectores	22
3.2	Selectores de elementos en las páginas generadas por las herramientas	25
3.2.1.	Clases e IDs en las páginas de portal	25
3.2.2.	Clases e IDs en las áreas visuales	27
3.2.3.	Clases e IDs en los contenidos y aplicaciones	28
3.2.4.	Clases para la versión imprimible	29
3.3	Las Hojas de Estilo y Scripts en las Páginas de Portal	30
3.3.1.	Hojas de estilo de reseteo de navegadores	32
3.3.2.	Hojas de estilo generales de un portal	32
3.3.3.	Hojas de estilo particulares de las páginas generadas a partir de una plantilla	35
3.3.4.	Estilos particulares de las Áreas Visuales	37
3.3.5.	Hojas de estilo del contenido / aplicación que se visualiza	40
3.3.6.	Estilos y scripts particulares por tipo (naturaleza) de contenido	41

3.3.7. Personalización de estilos para una iniciativa	43
3.3.8. Hojas de estilo, scripts y METAS particulares de contenidos en el HEAD de la página	44
4 ANEXO: Guía de Integración de Aplicaciones	53

## 1 Introducción

El presente documento contiene una guía de desarrollo en la infraestructura de portales para lo cual se describen las bases arquitectónicas de la infraestructura así como el funcionamiento interno de la misma.

Una vez sentadas las bases de funcionamiento, se detalla cómo se deben programar las páginas de portal, contenidos y aplicaciones para tener una infraestructura ordenada.

**Audiencia:** Analistas y Desarrolladores

**Complejidad:** Media / Alta

## 2 Conceptos Previos

### 2.1 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.

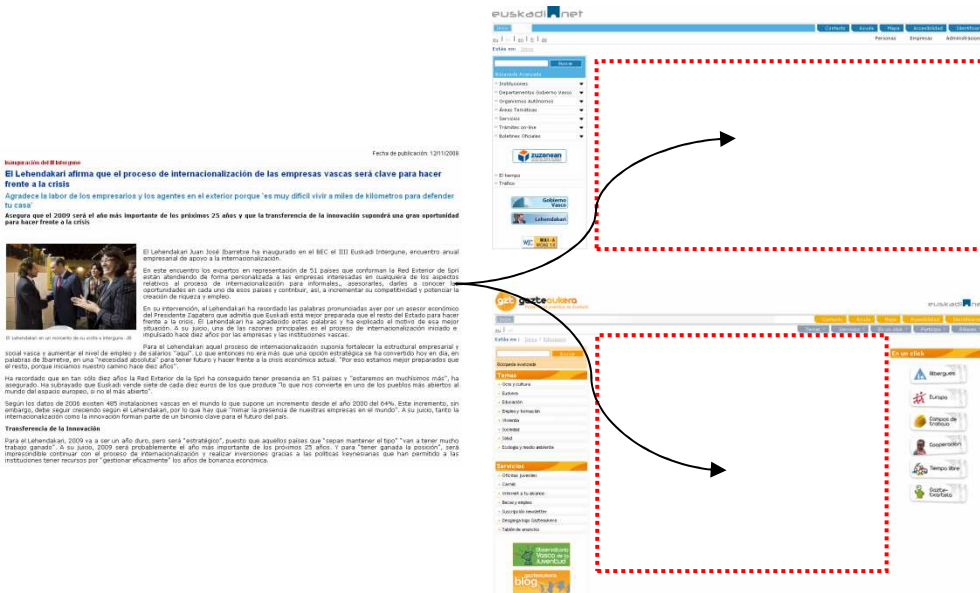
Desde el punto de vista **tecnológico**

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**.



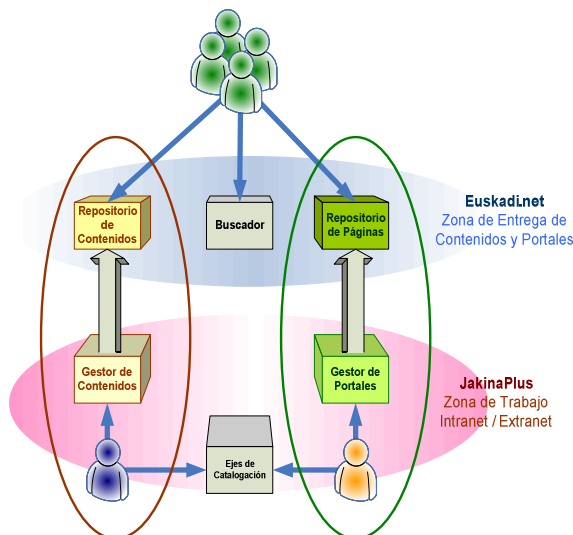
En la figura se muestra como el portal se encarga de “vestir” el contenido que se encuentra en un sistema separado e independiente.

## 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 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.

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

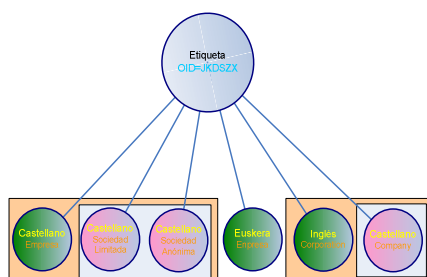
En vertical se muestran las dos herramientas principales: **gestión de contenidos** y **gestión de portales** y cómo el **buscador** y la **herramienta de catalogación** hacen de "unión" entre ambos mundos.

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:

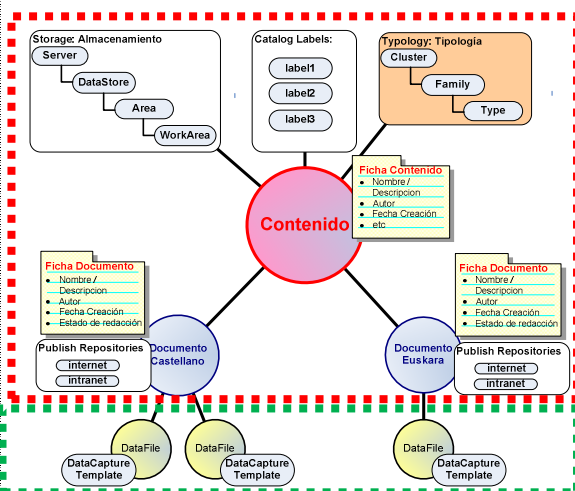
*Un contenido puede estar destinado a ciudadanos, siendo este un criterio que aporta el autor (subjetivo) ya que puede que no se mencione la palabra "ciudadano" en un contenido. En este caso, el autor catalogará el contenido con la etiqueta "ciudadano" del eje "destinatarios"*

## Infraestructura no intrusiva

La infraestructura no condiciona la forma / tecnologías en las que se desarrollan aplicaciones para euskadi.net, es decir, las herramientas / tecnologías que permiten la publicación de contenidos en Euskadi.net **no condicionan** la selección de tecnologías / herramientas para el desarrollo de aplicaciones departamentales

## 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	<p>El contenido es la <b>unidad mínima</b> 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.</p> <p>Metadatos del contenido son:</p> <ul style="list-style-type: none"> <li>• Dónde se almacena el contenido en el Gestor de Contenidos</li> <li>• Etiquetas de catalogación</li> <li>• Tipología: naturaleza del contenido (noticia, evento, etc)</li> <li>• Otros metadatos comunes (nombre, descripción, autor)</li> </ul> <p>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</p>
Documento / Version	<p>Un documento es una <b>versión</b> 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 contenidos específicos para un destinatario (jóvenes, empresarios, etc)</p> <p>Ej: <i>Un contenido puede expresarse en castellano y con una narrativa dirigida los jóvenes o bien expresarse en euskera con una narrativa más técnica para expertos.</i></p> <p>El documento tiene a su vez una serie de metaDatos propios:</p> <ul style="list-style-type: none"> <li>• Nombre, descripción, autor, etc</li> <li>• Repositorio público (de Euskadi.net) donde se ha publicado (una versión de un contenido puede publicarse en varios repositorios públicos)</li> </ul> <p>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.</p>
Archivo de Datos	<p>Un contenido /documento en sí mismo son entidades lógicas, sin embargo, los documentos contienen <b>archivos de datos</b>, adjuntos, imágenes, etc que es donde realmente está la información a ofrecer.</p> <p>Un archivo de datos es por lo tanto un fichero capturado utilizando una <b>plantilla de introducción de datos</b> y que contiene los <b>metaDatos específicos</b> (aquellos que no son los comunes expuestos con anterioridad).</p> <p>Ejemplo: <i>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.</i></p> <p>Los ficheros de datos se guardan como XML, sin embargo, para su publicación en Internet se “generan” en HTML, utilizando <b>plantillas de presentación</b>.</p>

## 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).

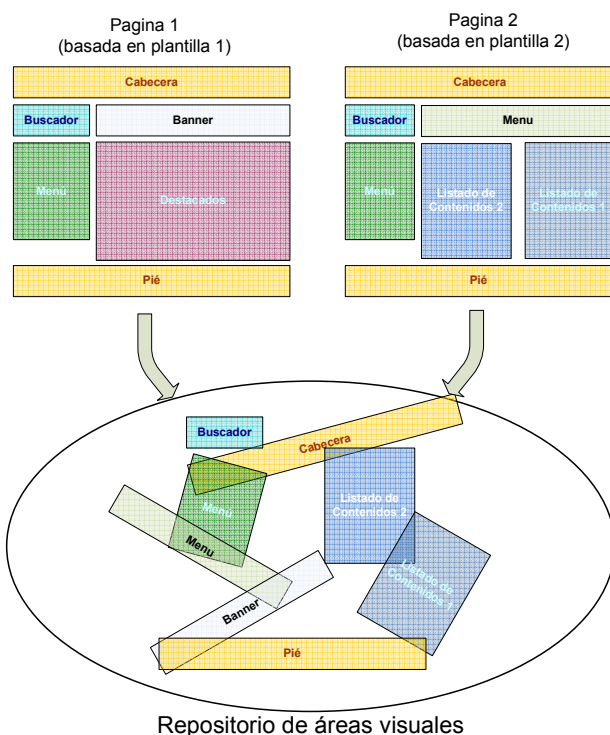
Las páginas de portal se estructuran en base a **áreas visuales** (AV) reutilizables en cualquier página del portal

Ejemplo: *Un menú general del portal, la cabecera o el pie 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, pie 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 parametrizar 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.



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

## 2.2 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

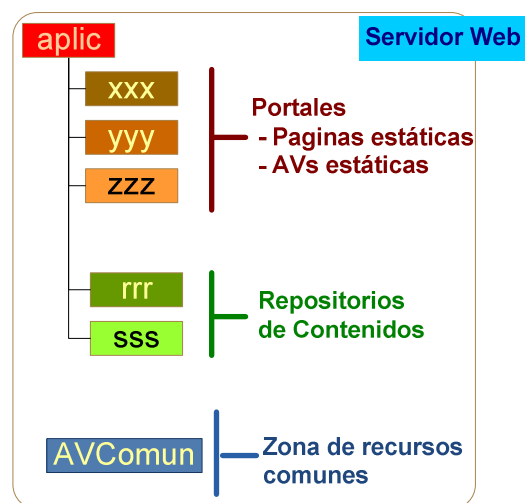


Hay que definir una estructura de URLs para mostrar un contenido dentro de una aplicación.

### 2.2.1. Estructura de directorios en el servidor web

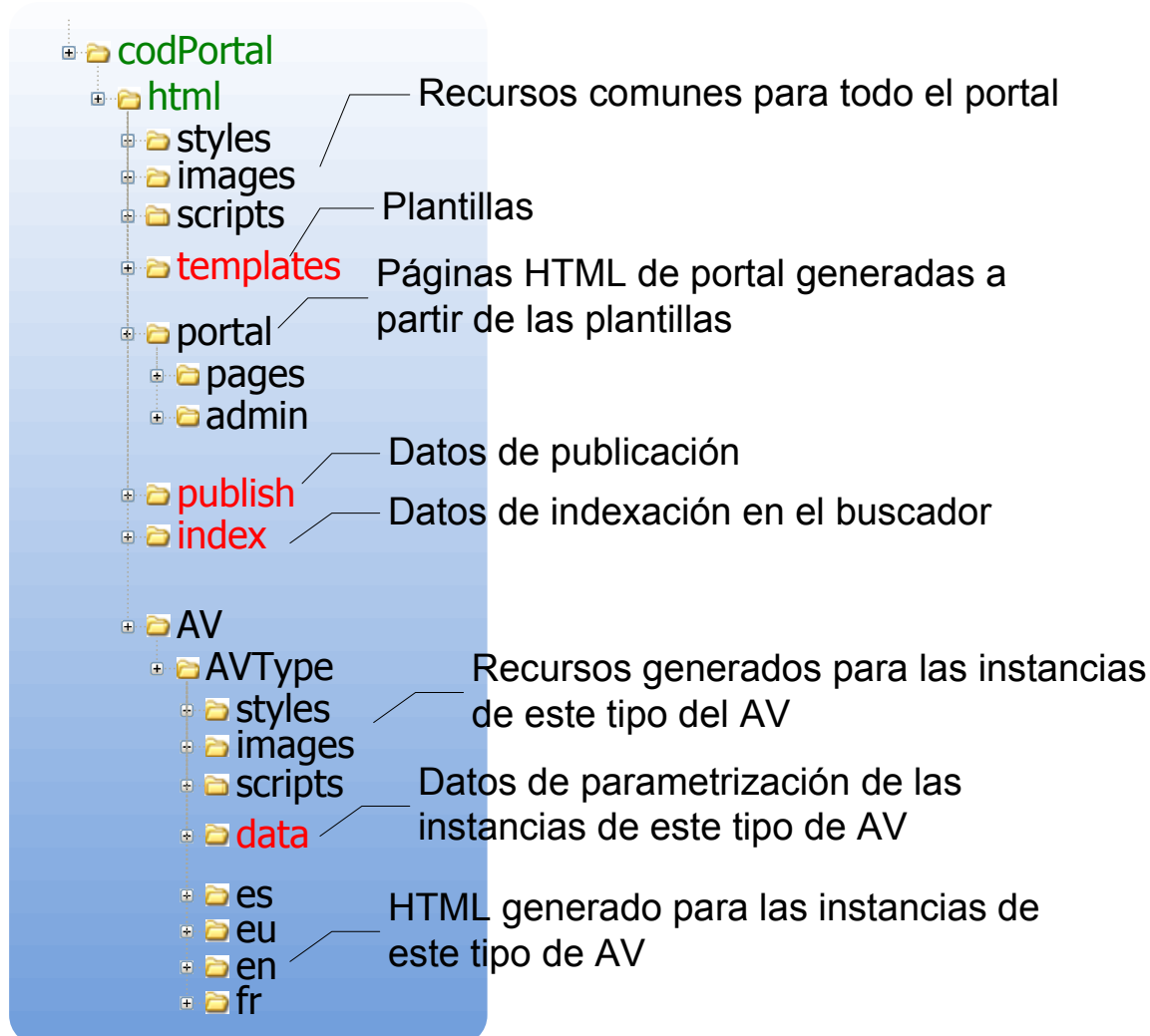
Dada la separación de contenidos y páginas de portal, en el servidor web debe existir tal y como se muestra en la figura:

- 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)



### 2.2.1.1. Estructura de directorios de un portal

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

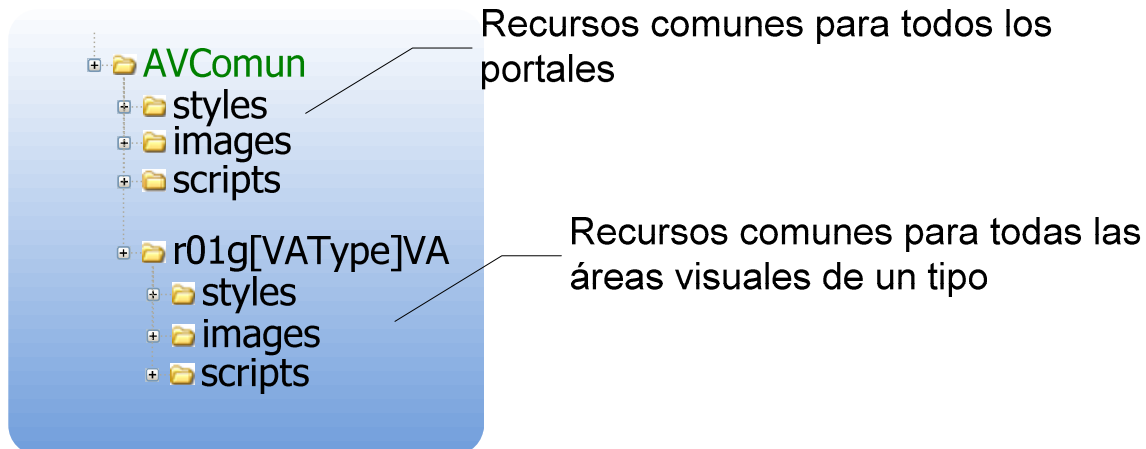


Como se puede observar, debajo de la carpeta del portal se almacenan todos los recursos (estáticos) del portal:

- 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

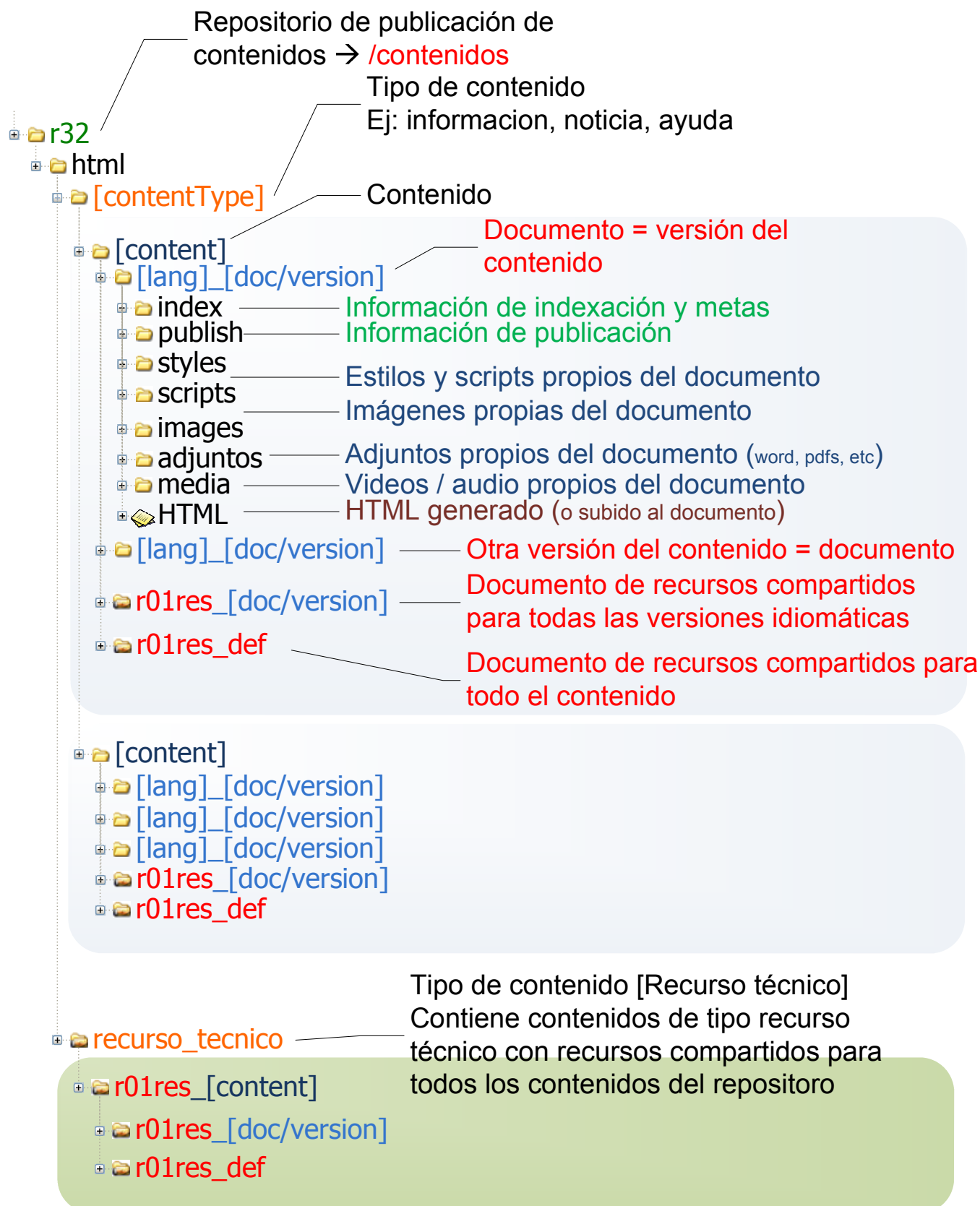
Todos los portales **comparten** una **zona común** donde se depositan recursos comunes a todos los portales. Esta zona se denomina **AVComun** y tiene la siguiente estructura:



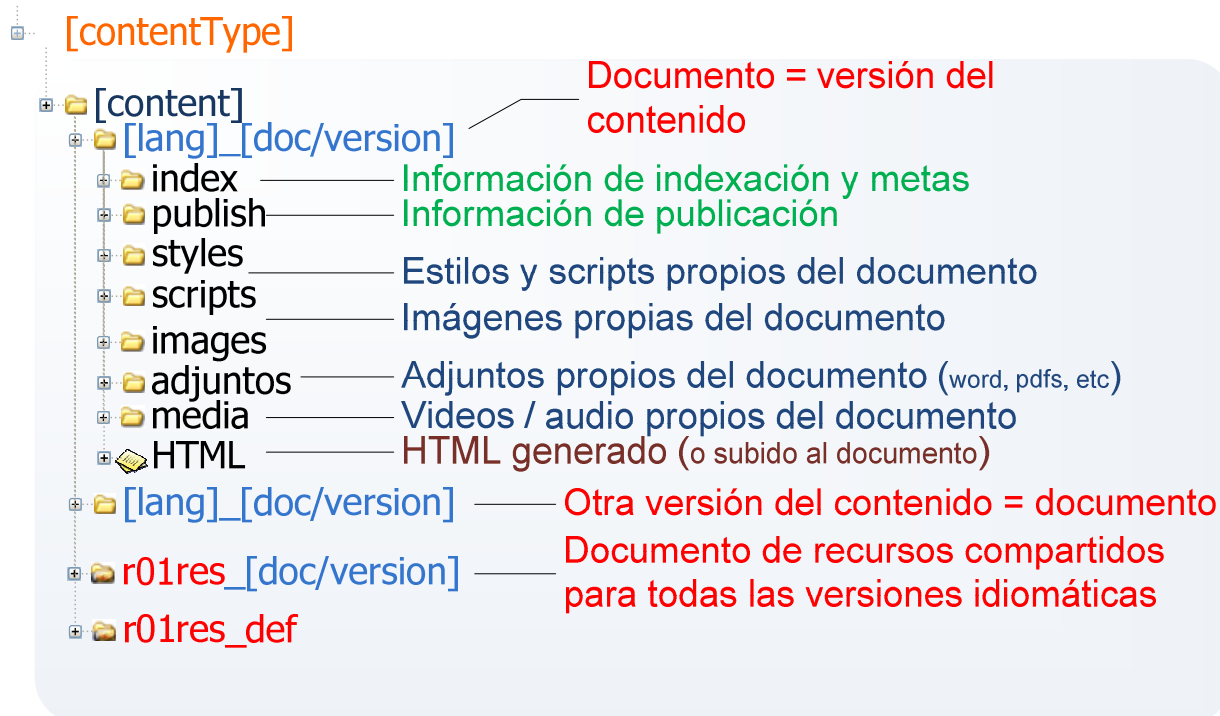
Para hacer referencia a un recurso de la zona común, dado que AVComun es un **alias web** que existe en todos los portales de la infraestructura, basta con poner: **/AVComun/[path relativo recurso]**

## 2.2.1.2. 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:

```
<img src='images/myImage.jpg'/>
```

en lugar de ser:

```
<img src='/contenidos/contentType/myContent/lang_myDoc/images/myImage.jpg'/>
```

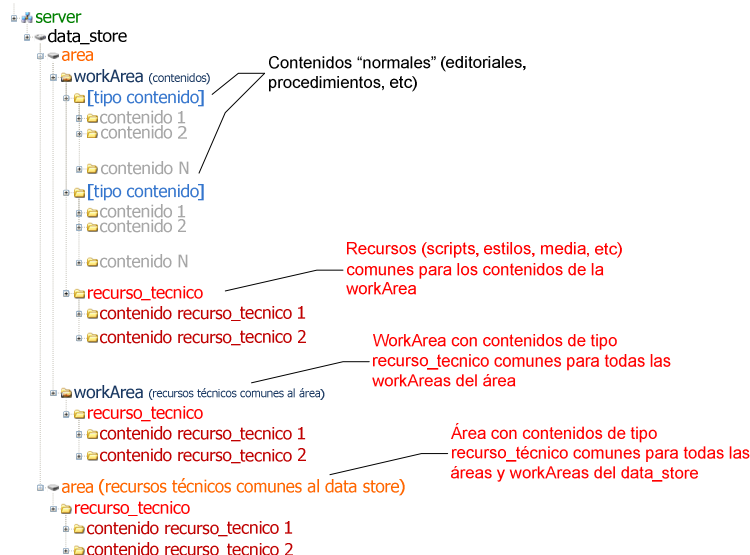
Si se hace una referencia a un recurso de un documento de recursos del contenido, la referencia debe ser:

```
<img src='../r01res_[nombreDoc]/images/myImage.jpg'/>
```

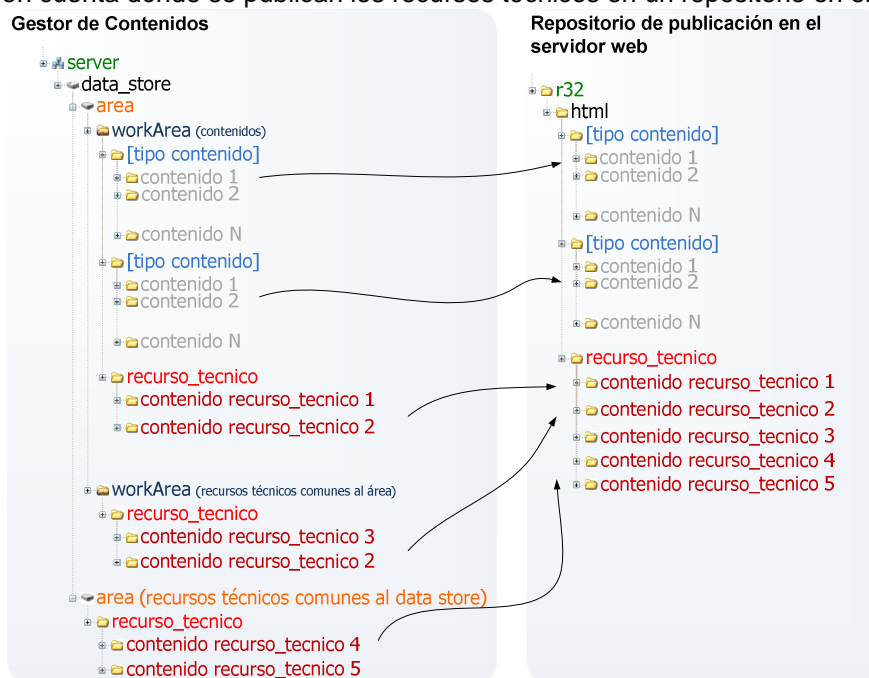
el problema es que un contenido puede ser publicado en **cualquier repositorio** como /contenidos e incluso en varios. Si se hacen referencias absolutas al repositorio, puede ser que estas referencias no funcionen.

- **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:

`<img src='../recurso_tecnico/[content]/r01res_[doc]/images/myImage.jpg/>`

en lugar de ser:

`<img src='/contenidos/recurso_tecnico/[content]/r01res_[doc]/images/myImage.jpg/>`

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] <b>ej:</b> <img href='../r01res_doc/[url_relativa]'
Recursos comunes a varios contenidos	../recurso_tecnico/[content]/r01res_[doc]/url_relativa <b>ej:</b> <img href='../../recurso_tecnico/r01res_doc/url_relativa'

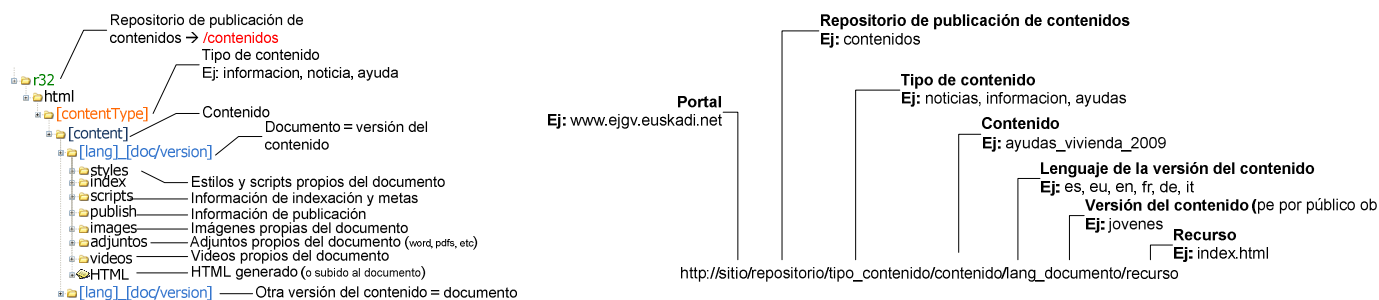
## 2.2.2. Estructura de URLs

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

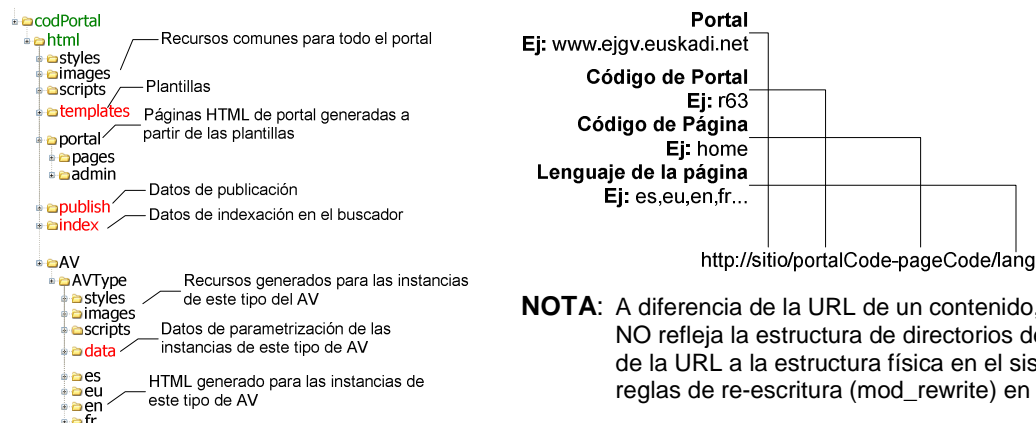
### 2.2.2.1. URL directa de un contenido o una página de portal

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 **SHTML**, 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/](http://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](http://www.ej-gv.net/r53-2283).

El usuario ha entrado a [www.euskadi.net/r33-2220/es](http://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](http://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](http://www.euskadi.net), cuando debería aparecer el nombre de dominio del Gobierno Vasco [www.ejgv.euskadi.net](http://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 **vhos.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**.

#### 2.2.2.2. 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](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.

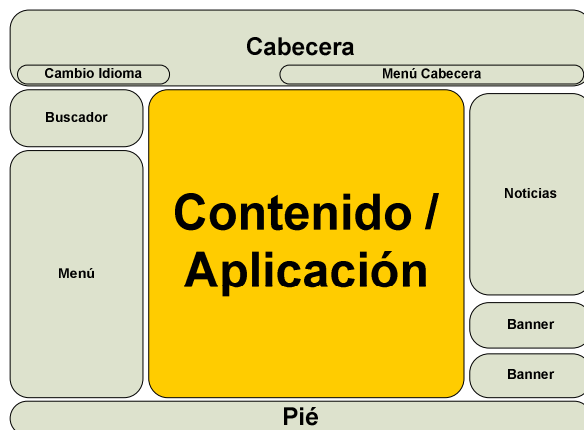
En los ejemplos anteriores, si se desea mostrar el contenido o aplicaciones en el portal de industria, la URL sería:

[http://www.industria.ejgv.euskadi.net/r58-1237/es/contenidos/informacion/es\\_12/indice.html](http://www.industria.ejgv.euskadi.net/r58-1237/es/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>

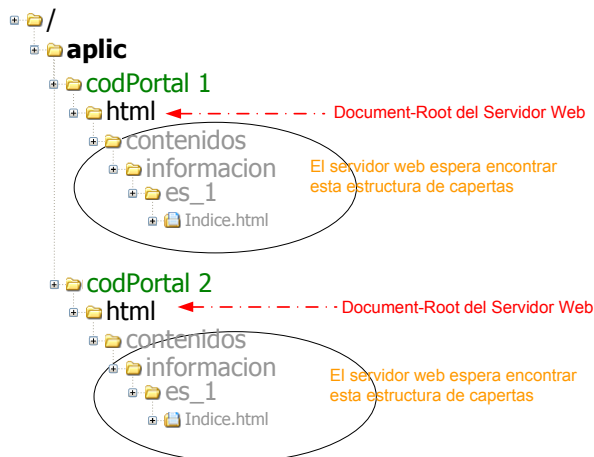
Para que una página sea capaz de mostrar un contenido o una aplicación, es necesario que esté preparado para ello, es decir, la página ha de tener un **contenedor de contenido / aplicación**.



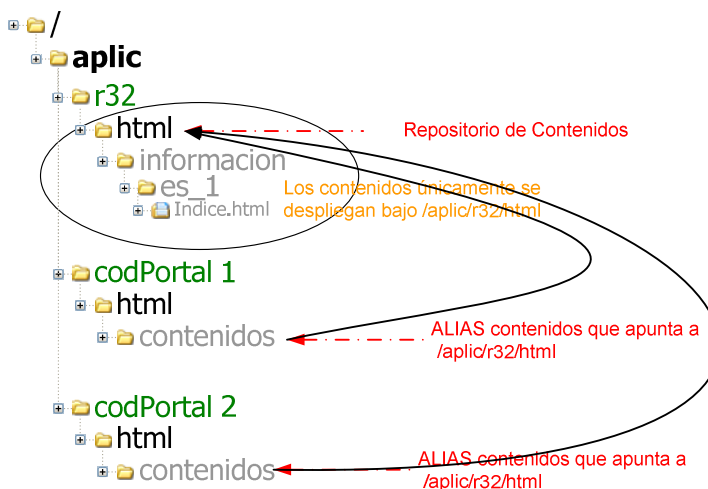
## 2.2.3. Repositorios

### 2.2.3.1. 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](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:



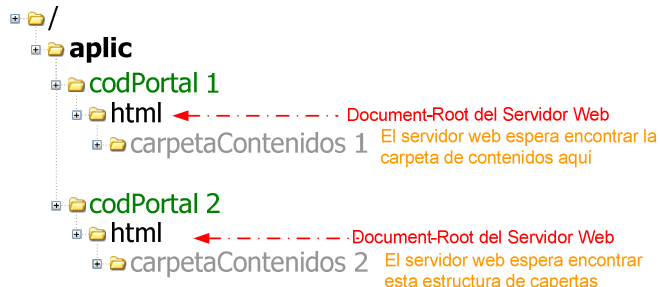
**El repositorio “contenidos”** almacena todos los contenidos publicados en euskadi.net y obviamente **no** pueden replicarse para cada uno de los portales de la red. Para solucionar el problema, se utiliza un **alias web** que “simula” que el repositorio “contenidos” existe en todos los webs de la red



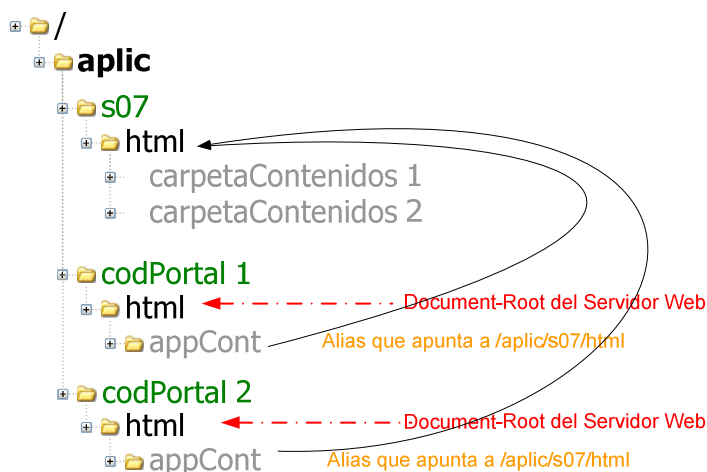
Este “truco” permite que cualquier portal de la red “vea” el repositorio de contenidos de euskadi.net y en principio cualquier otro repositorio de datos que se pueda ir creando.

### 2.2.3.2. 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>

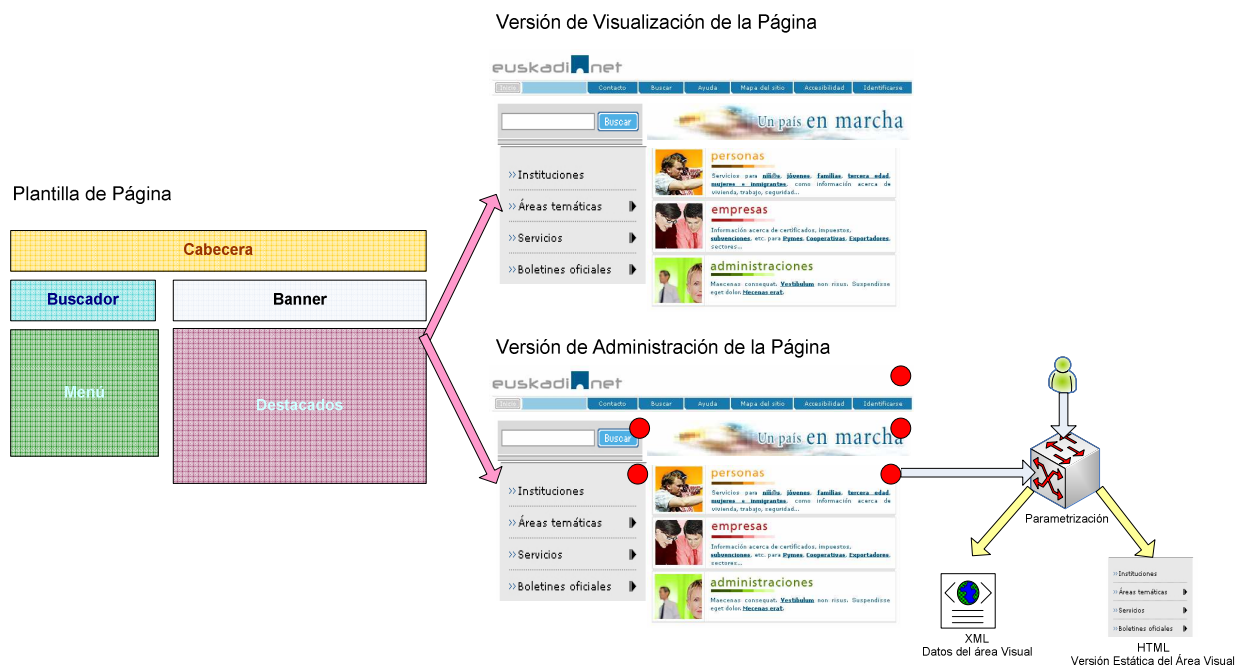
<http://www.industria.ejgv.euskadi.net/appCont/carpetaContenidos/elContenidoEstatico>

## 2.3 Entrega de páginas de portal

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

<b>Área Visual</b>	Elemento de una página web parametrizable en : → Su aspecto visual → La información que muestra → En su comportamiento.
<b>Plantilla de Página</b>	Disposición concreta de áreas visuales que permite la creación de páginas con la misma disposición.
<b>Página</b>	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

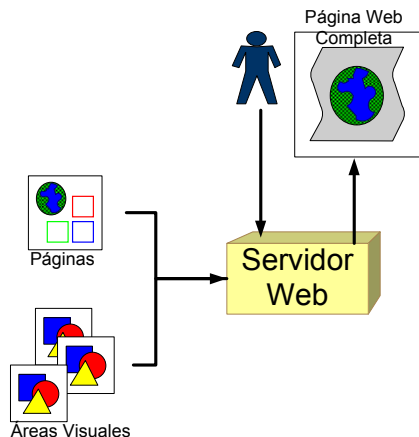
**NOTA:** Recordar que las páginas son **independientes del idioma**, es decir, pueden visualizarse en cualquier idioma siempre que las AVs que componen la página estén configuradas en dicho idioma

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.

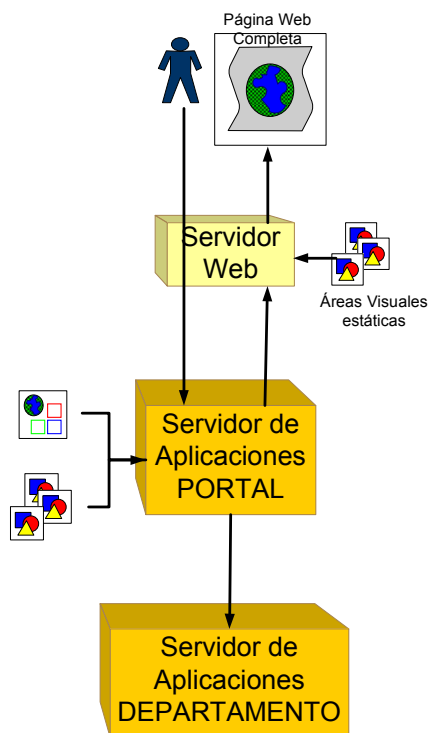
La composición de las páginas y las áreas visuales se realiza mediante **reglas de re-escritura** en el servidor web apache de forma que al cliente (navegador del usuario) le llega una página completa.



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 enteraría” 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.

## 3 Hojas de Estilo y Scripts para Páginas de Portal, Contenidos y Aplicaciones

### 3.1 Conceptos Generales sobre hojas de estilo

#### 3.1.1. Inclusión de Hojas de Estilo

Los estilos en los elementos HTML pueden aplicarse de las siguientes formas:

##### Referenciar una hoja de estilos en un fichero css externo:

```
<link href="{nombre_fichero_css}" rel="stylesheet" type="text/css" media="{medio}">
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  <title>My Page</title>
  <link href="myStyleSheet-all.css" rel="stylesheet" type="text/css" media="all">
  <link href="myStyleSheet-screen.css" rel="stylesheet" type="text/css" media="screen">
  <link href="myStyleSheet-handheld.css" rel="stylesheet" type="text/css" media="handheld">
</head>
<body>

</body>
</html>
```

##### Incluir los estilos directamente en la sección HEAD de la página HTML

```
<style media='all' type='text/css'>
... the inline styles ...
</style>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  <title>My Page</title>
  <style media='all' type='text/css'>
... the inline styles ...
  </style>
</head>
<body>

</body>
</html>
```

##### Incluir los estilos directamente en el elemento HTML

```
<ELEMENT style='text-align:center'>
...
</ELEMENT>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  <title>My Page</title>
  [NO se incluyen estilos en la sección HEAD]
</head>
<body>
  <div id='myDiv' style='text-align:center'>
    This is myDiv
  </div>
</body>
</html>
```

## IMPORTANTE!

De los tres métodos anteriores, el más recomendable es el primero: **referenciar una hoja de estilos en un fichero independiente:**

```
<link href="[nombre_fichero_css" rel="stylesheet" type="text/css" media="[medio]">
```

## NOTA:

El atributo *media* indica el medio de visualización al que se aplica la hoja de estilos:

<b>all</b>	Todos los dispositivos
<b>braille</b>	Dispositivos táctiles braille
<b>embossed</b>	Dispositivos de impresión braille
<b>handheld</b>	Dispositivos móviles, normalmente con pantalla pequeña y ancho de banda limitado
<b>print</b>	Impresión o pre-visualización de impresión en pantalla
<b>projection</b>	Proyectores
<b>screen</b>	Pantalla de ordenador
<b>speech</b>	Sintetizadores de voz (equivalente al medio <i>aural</i> )
<b>tty</b>	Medios que utilizan fuentes mono-espaciadas (teletipos, terminales, etc)
<b>tv</b>	Televisores (baja resolución, scroll limitado)

Dentro de una hoja de estilos se pueden incluir estilos para distintos medios de visualización de la siguiente forma:

```
@media print {
    The styles for the printer
}
@media screen {
    The styles for the screen
}
@media screen, print {
    The styles for the screen and the printer
}
```

En ocasiones puede ser necesario incluir (embeber) una hoja de estilos dentro de otra, para lo cual se utiliza la instrucción `@import` en la que se indica la url de la hoja de estilos y el medio al cual debe aplicarse.

```
@import url("theEmbeddedStyleSheetFile.css") [media];
```

Esta técnica puede ser útil cuando no es posible modificar el HTML pero si la hoja de estilos incluida:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
    <title>My Page</title>

    <link href="myStyleSheet-all.css" rel="stylesheet" type="text/css" media="all">
</head>
<body>
    ...
</body>
</html>
```

NO es posible modificar el HTML pero si la hoja de estilos [myStyleSheet-all.css](#) para incluir otra hoja de estilos

### 3.1.2. Selectores

Una hoja de estilos CSS se estructura en base a **reglas** con el siguiente formato:

<b>selector</b> {		
property	:value;	
property	:value;	
...		
}		

El selector "selecciona" los elementos de la página HTML que se ven afectados por la regla

En una regla se pueden indicar múltiples selectores que "seleccionarán" múltiples elementos HTML de la página a los que se aplicarán las mismas reglas:

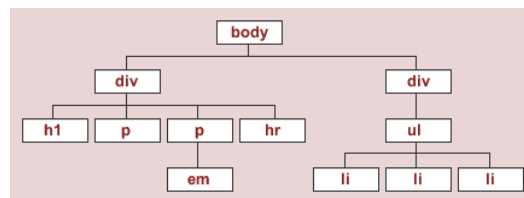
<b>h1, h2, h3, h4</b> {		
padding: 1em;		
}		

Esta regla aplica un padding de 1em a todos los elementos H1, H2, H3 y H4 de la página HTML

Conocer la estructura en árbol de un documento HTML es básico a la hora de escribir correctamente una hoja de estilos:

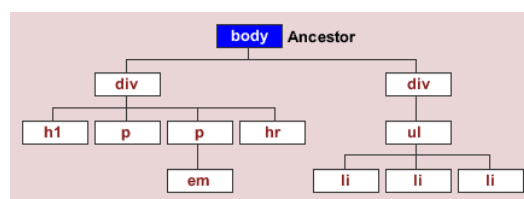
```
<body>
  <div id="content">
    <h1>Heading here</h1>
    <p>Lorem ipsum dolor sit amet.</p>
    <p>Lorem ipsum dolor <em>sit</em> amet.</p>
    <hr />
  </div>
  <div id="nav">
    <ul>
      <li>item 1</li>
      <li>item 2</li>
      <li>item 3</li>
    </ul>
  </div>
</body>
```

El documento HTML se puede presentar gráficamente como



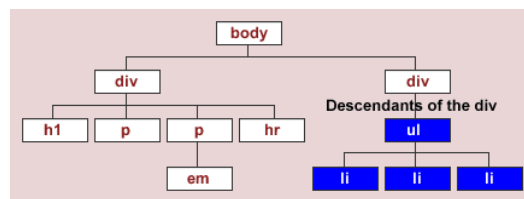
**Ancestor:** Elemento que es conectado por encima en el árbol del documento (no importa cuántos niveles por encima)

*El elemento BODY es ancestor de cualquier otro elemento de la página*



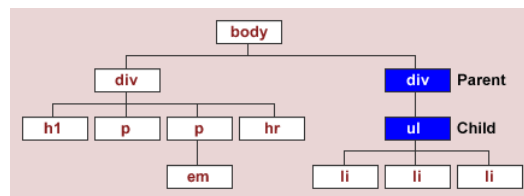
**Descendant** Elemento que es conectado por debajo en el árbol del documento (no importa cuántos niveles por debajo)

*Los elementos señalados en azul son descendant del DIV o el BODY*



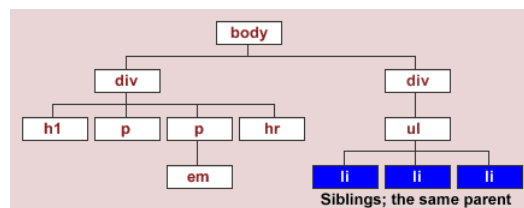
**Parent** Elemento que está directamente encima de uno dado

**Child** Elemento que está directamente debajo de uno dado



**Sibling** Elemento que comparte el mismo padre con otro.

*Los elementos señalados en azul son siblings ya que tienen el mismo padre (UL)*

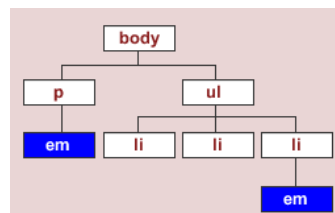


En cuanto a los selectores los hay de varios tipos:

**Tipo:** Selecciona elementos en el documento HTML en función del **tipo de elemento HTML**

**Selector:** Nombre del elemento HTML

```
em {color: blue; }
```

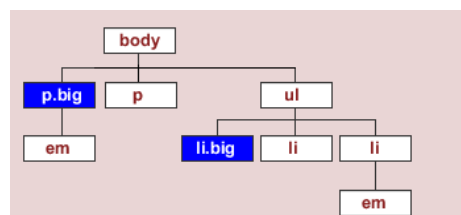


**Clase** Selecciona elementos en el documento HTML en función del nombre de la clase asignado al elemento

```
<body>
  <p class="big">This is some <em>text</em></p>
  <p>This is some text</p>
  <ul>
    <li class="big">List item</li>
    <li>List item</li>
    <li>List <em>item</em></li>
  </ul>
</body>
```

**Selector:** **.**[Nombre de la clase]

```
.big {
  font-size: 110%;
  font-weight: bold;
}
```

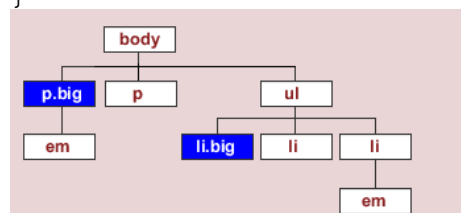


**ID** Selecciona elementos en el documento HTML en función del ID del elemento

```
<body>
  <p id="big">This is some <em>text</em></p>
  <p>This is some text</p>
  <ul>
    <li id="big">List item</li>
    <li>List item</li>
    <li>List <em>item</em></li>
  </ul>
</body>
```

**Selector:** **#**[Nombre de la clase]

```
#big {
  font-size: 110%;
  font-weight: bold;
}
```

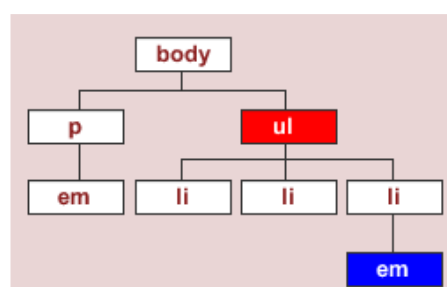


**Descendant** Selecciona elementos en el documento HTML descendientes de otro

```
<body>
  <p>Lorem ipsum dolor <em>sit</em> amet.</p>
  <ul>
    <li>item 1</li>
    <li>item 2</li>
    <li><em>item 3</em></li>
  </ul>
</body>
```

**Selector:** [Selector elem superior] [Selector elem descendiente]

```
ul em {color: blue; }
```



Los anteriores son los selectores básicos admitidos por todos los navegadores web. Hay otros selectores: **child**, **sibling**, **etc** pero no son admitidos por todos los navegadores.

En base a los selectores básicos anteriores se pueden hacer combinaciones como por ejemplo:

```
<body>
  <p class="big">This is some <em>text</em></p>
  <p>This is some text</p>
  <ul>
    <li id="big">List item</li>
    <li>List item</li>
    <li>List <em>item</em></li>
  </ul>
</body>
```

OJO!! Un elemento puede tener  
varias clases pero un solo ID

```
<body>
  <p>Lorem ipsum dolor <em>sit</em> amet.</p>
  <ul class='myClass1 myClass2' id='myId1'>
    <li>item 1</li>
    <li>item 2</li>
    <li id='myId2'>
      <em>item 3</em>
    </li>
  </ul>
</body>
```

Seleccionar un elemento tipo P que tenga  
clase "big"

```
p.big {
  font-size: 110%;
  font-weight: bold;
}
```

Seleccionar un elemento tipo LI que tenga ID  
"big"

```
li#big {
  font-size: 110%;
  font-weight: bold;
}
```

Seleccionar los elementos tipo P que tengan  
clase "big" y los elementos LI con ID "big"

```
p.big, li#big {
  font-size: 110%;
  font-weight: bold;
}
```

Seleccionar los EM que sean descendientes  
de un LI con id "myId2" y que a su vez sea  
descendiente de un UL con clase "myClass1"

```
ul.myClass1 li#myId2 em {
  font-size: 110%;
  font-weight: bold;
}
```

## 3.2 Selectores de elementos en las páginas generadas por las herramientas

El HTML de páginas y áreas visuales es generado por la herramienta de portales y esta para facilitar la personalización de estilos **introduce una serie de clases e ids en los elementos HTML**

### 3.2.1. Clases e IDs en las páginas de portal

Las clases e IDs de las páginas de portal dependen directamente de las clases e IDs que se indiquen en la plantilla (puesto que la página se genera a partir de la plantilla).

Una plantilla es un **HTML “normal y corriente”** que incluye áreas visuales así que está **en la mano del desarrollador de la plantilla** el “marcar” el HTML con clases e IDs de la forma más conveniente para aplicar posteriormente los estilos de posicionamiento de AVs y personalización de la apariencia.

El siguiente código es un ejemplo de plantilla de página de portal en la que:

- Se han recuadrado en **rojo** las clases que el desarrollador ha decidido asignar a los elementos HTML
- Se han sombreado en **amarillo** las hojas de estilo que se aplicarán a las páginas generadas a partir de la plantilla
- Se han sombreado en **marrón** la definición de AVs

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
<head>
  <!-- Directiva para la previsualización de la plantilla, se borrará de las páginas -->
  <!--@avDef id='1'-->
  <!--#include virtual='/AVTemplates/r01gCommonVA/r01gCommonVA_$LANG.html'-->
  <!--@avDef id='0'-->
  <!--#include virtual='/AVTemplates/r01gHeadVA/r01gHeadVA_$LANG.html'-->

  <link rel="stylesheet" type="text/css" href="/styles/r01-general.css" />
</head>
<body class='r01PrintablePage'>

<!-- ===== LOGOTIPOS ===== -->
<div class="r01_headerLogos r01Printable r01clearfix">
  <div class="r01_headerLogoLeft">
    <!--@avDef id="1001" -->
    <!--#include virtual='/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html' -->
  </div>
  <div class="r01_headerLogoCenter">
    <!--@avDef id="1002" -->
    <!--#include virtual='/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html' -->
  </div>
  <div class="r01_headerLogoRight">
    <!--@avDef id="1003" -->
    <!--#include virtual='/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html' -->
  </div>
</div>

<!-- ===== PRIMERA LINEA DE LA CABECERA ===== -->
<div class="r01_headerRow1 r01clearfix">
  <div class="r01_homePageButton">
    <!--@avDef id="1004" -->
    <!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html" -->
  </div>
  <div class="r01_headerRow1Menu">
    <!--@avDef id="1005" -->
    <!--#include virtual="/AVTemplates/r01gHorizontalMenuVA/r01gHorizontalMenuVA_$LANG.html" -->
  </div>
</div>

<!-- ===== BLOQUE CENTRAL: CONTENIDO ===== -->
<div class="r01_body r01clearfix">
  <div class="r01gContainer">
    <!--@avDef id="1009" -->
    <!--#include virtual='/AVTemplates/r01gContainerVA/r01gContainerVA_$LANG.html' -->
  </div>
</div>

<!-- ===== PIE ===== -->
<div class="r01_footer r01Printable">
  <!--@avDef id="2000" -->
  <!--#include virtual="/AVTemplates/r01gPageFootVA/r01gPageFootVA_$LANG.html" -->
</div>
</body>
</html>
```

Los estilos que se aplican son los que se muestran a continuación, donde se ha sombreado en marrón las referencias que se hacen a las clases especificadas en el HTML de la plantilla.

#### r01-general.css

```

/***** Global Settings *****/
@import url(/AVComun/styles/r01PortalResetToBaseStyles.css);

body {
    background-color    : #fff;
}

/* ===== Logotipos de la cabecera ===== */
div.r01_headerLogos {
    width                : 100%;
    padding-bottom      : 5px;
    border-bottom       : 1px solid #ccc;
}
div.r01_headerLogoLeft {
    float                : left;
    text-align           : left;
    width               : 13%;
}
div.r01_headerLogoCenter {
    float                : left;
    text-align           : left;
    width               : 53%;
}
div.r01_headerLogoRight {
    float                : right;
    text-align           : right;
    width               : 33%;
}
div.r01_headerLogoRight {
    float                : right;
    text-align           : right;
    width               : 33%;
}
/* ===== Primera linea de la cabecera ===== */
div.r01_headerRow1 {
    width                : 100%;
    border-bottom       : 1px solid #ccc;
    padding-bottom      : 3px;
    padding-top         : 3px;
    z-index              : 1000;
}
div.r01_homePageButton {
    float                : left;
    width               : 10%;
}
div.r01_headerRow1Menu {
    float                : left;
    width               : 89%; /* no llegar al 100% junto con el boton (bug explorer) */
}
/* ===== Cuerpo central ===== */
div.r01_body {
}
/* ===== Pie de pagina ===== */
div.r01_footer {
    margin-top           : 20px;
    width                : 100%;
}

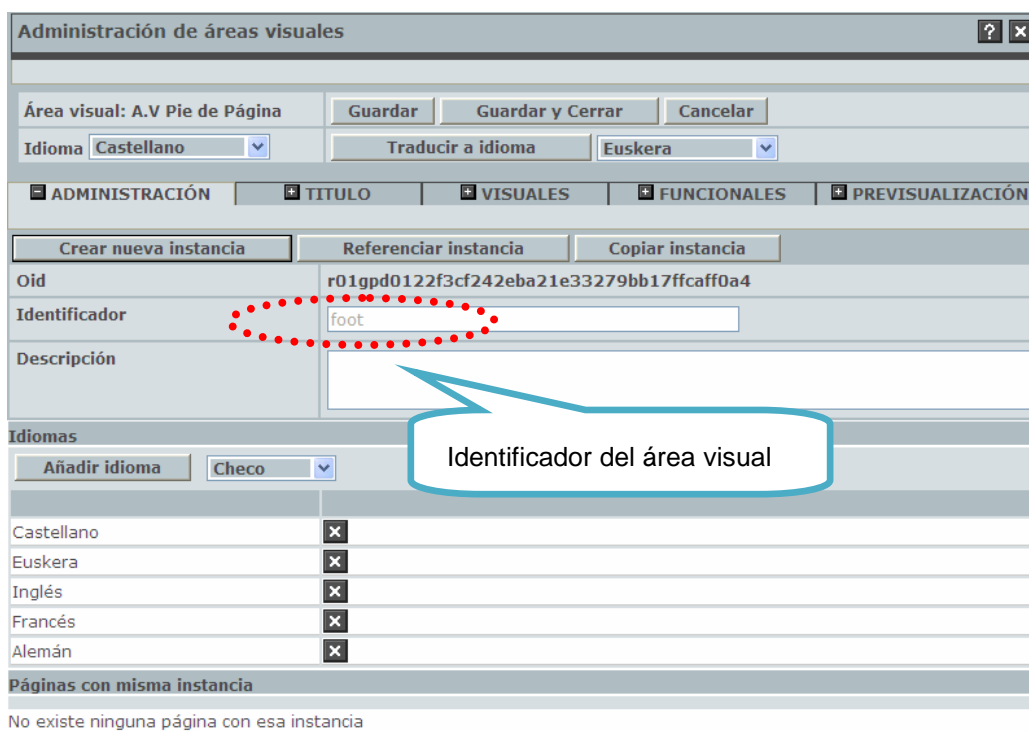
```

NOTA: Esta hoja CSS es básicamente de **posicionamiento** (layout) de AVs, establece cómo se colocan las AVs en la página.

### 3.2.2. Clases e IDs en las áreas visuales

Normalmente la apariencia visual de las áreas visuales se parametriza utilizando las herramientas de usuario final que proporciona la herramienta, sin embargo, en ocasiones es necesario bajar a las “tripas” del HTML generado para personalizar aspectos que de otra forma no sería posible.

Esto exige incluir estilos específicos para las **instancias de las AVs**, para lo cual todas las AVs tienen un **identificador** (un nombre interno) que el usuario asigna **al instanciar** el área visual durante la administración:



El HTML generado para el AV mantiene este id y algunas otras cosas tal y como se ve en el siguiente código:

```
<!-- ***** INIT Visual Area: r01gPageFootVA ***** -->
<!--@@<xml id='2000'><avInst oid='r01gpd0122f3cf242eba21e33279bb17ffcaff0a4' type='r01gPageFootVA' /></xml>@@-->
<div id="r01gFootfoot" class="r01gClsFoot r01gClsFootfoot">
  <div class="r01gClsFootContent r01clearfix">
    <div class="r01gClsFootLeft r01gClsFootConAncla"><a
      href="http://#">Informaci&#243;n legal</a>
    </div>
    <div class="r01gClsFootRight r01gClsFootSinAncla">
      &#169; 2004 &#183; Eusko Jaurlaritz - Gobierno Vasco
    </div>
  </div>
</div>
<!-- ***** END Visual Area: r01gPageFootVA ***** -->
```

Todas las AVs están “encapsuladas” en un DIV que **siempre** tiene las siguientes características:

- Tiene un ID como **id='[tipoAV][idAVinstance]'**
- Tiene un class HTML con **dos clases: class='[tipoAV] [tipoAV][idAVinstance]'**

Adicionalmente, en función del tipo de área visual, el HTML interno tiene más marcadores. Lo mejor es ver un ejemplo del HTML del tipo concreto para identificar los marcadores.

En el ejemplo anterior sería muy fácil cambiar el color de fondo del AV utilizando un estilo que selecciona el div con id `r01gFootfoot`

```
div#r01gFootfoot {
  background-color :red;
}
```

### 3.2.3. Clases e IDs en los contenidos y aplicaciones

Tal y como se ha explicado anteriormente, los contenidos y aplicaciones se “inyectan” dentro de un área visual de contenedor de contenidos y aplicaciones la cual, de la misma forma que cualquier otro área visual está identificada dentro del HTML (ver punto anterior).

```
<!-- ***** INIT Visual Area: r01gContainerVA ***** -->
<!--@@<xml id='1009'><avInst oid='r01gpd0122f3dac1bfba21e339475395d35f16b07' type='r01gContainerVA' /></xml>@@-->
<div id="r01gContainermycontainer" class="r01gContainer r01gContainermycontainer">
    [Aquí se "inyecta el contenido"]
</div>
<!-- ***** END Visual Area: r01gContainerVA ***** -->
```

Al igual que cualquier otro AV el container de id *mycontainer* viene identificado en el HTML con un DIV con:

- Tiene un ID como **id='[tipoAV][idAVinstance]**
- Tiene un class HTML con **dos clases: class='[tipoAV] [tipoAV][idAVinstance]**

El HTML “inyectado” dentro del container tendrá a su vez su marcado HTML que dependerá del contenido / aplicación que se esté mostrando

De todas formas, **el selector de cualquier estilo que se aplique a un contenido / aplicación, debería tener el formato:**

```
div.r01gContainer [resto de selectores específicos] {
}
```

**TODOS** los selectores de estilos del contenido / aplicación deberían empezar por **div.r01gContainer** ya que esto “filtra” todos los elementos HTML a aquellos que tengan como **ancestor** un DIV con la clase HTML **r01gContainer** que es la clase que tienen **TODAS las instancias** del AV container (independientemente de su id)

**OJO!!**

**NO** se filtra por el ID del AV (ej div#r01gContainermyContainer) ya que este filtro sólo serviría para mostrar la aplicación en una instancia del AV container específica.

Hay que tener en cuenta que el contenido se puede mostrar en cualquier instancia de AV container de cualquier portal así que no se debe ligar a una instancia concreta sino a “todas” a través de la clase [tipoAV]

### 3.2.4. Clases para la versión imprimible

En general, cuando se imprima una página desde el navegador, los elementos aparecerán según los estilos definidos en la hoja de estilos de la página. Dado que normalmente éstos estilos no están adaptados al tamaño de una hoja de papel, puede ocurrir que las distintas capas se desplacen y la página impresa quede alterada en relación a su aspecto en pantalla.

Se han definido una serie de estilos específicos para la impresión y que requieren de:

1. Identificar que la página aplicará estilos particulares de impresión añadiendo al body de la página la clase `r01PrintablePage`:

```
<body class="r01PrintablePage">
```

Aquellas páginas que no tengan esta clase incluida en el body no aplicarán estilos específicos de impresión.

2. Los elementos (áreas visuales) que se desee que se impriman deben tener la clase ***r01Printable***. Dicho estilo hace que se impriman la capa en cuestión y todas sus hijas.

```
<div class="r33_headerLogos r01Printable r01clearfix">
```

NOTA: Se aplicarán todos los estilos “normales” (de pantalla) definidos en la página, por lo que si por ejemplo se define una capa con un ancho específico, dicho tamaño se mantendrá en la versión impresa.

3. Para las capa con un ancho limitado en versión pantalla, se define un estilo para que la impresión se aplique con un 100% de anchura.

```
<div class="r33_body r01PrintableFullWidth r01clearfix">
```

Este estilo se aplicará tanto a la capa en cuestión como a las capas hijas. Por ello hay que tener cuidado dónde se establece ya que si la capa en cuestión tuviese otros divs incluidos que se quieran visualizar, estos aparecerían cada uno en una línea. Hay que tener en cuenta que este estilo sólo establecerá el ancho de impresión al 100%. Para que el elemento se visualice también debe incluirse el estilo `r01Printable`.

Ej.

```
<div class="r33_body r01PrintableFullWidth r01clearfix"> ← CADA CAPA INCLUIDA TENDRÁ ANCHO DE 100%. ES DECIR,
CADA CAPA VISIBLE EN UNA LINEA

    <div class="r33_bodyLeftCol"> ← CAPA NO VISIBLE EN LA IMPRESION

    <div class="r33_bodyCenterCol">

        <div class="r33_bodyCenterColContainer">

            <div class="r01gContainer"> ← CAPA VISIBLE (CAPA CONTENEDORA): APARECE EN UNA LINEA

        <div class="r33_bodyRightCol r01Printable"> ← CAPA VISIBLE (r01Printable): APARECE EN OTRA LINEA
```

### 3.3 Las Hojas de Estilo y Scripts en las Páginas de Portal

#### Reseteo

Estilos de reseteo de estilos que conforman una base para desarrollar el resto de estilos de una forma más uniforme para todos los navegadores.

#### Estilos generales del portal

Dentro de un portal, habitualmente, hay áreas visuales que son **compartidas por todas (o la mayoría) de las plantillas (o páginas)**.

Ejemplo: *Es normal que en un portal la cabecera, el pie o el menú izquierdo o superior se mantengan en todas las páginas*

Dentro de los estilos **generales** del portal hay tres tipos:

<b>Layout</b> (disposición) de áreas visuales comunes	Estructuran (disponen / colocan) las <b>áreas visuales comunes</b> dentro de la página.
<b>Visuales de zonas comunes</b>	Personalizan la visualización de las <b>zonas comunes de la mayoría de las páginas del portal</b> Ej: <i>Incluir imágenes de fondo o bordes en zonas comunes de la página (cabecera, menús, pie, etc)</i>
<b>Personalización de áreas visuales comunes</b>	En ocasiones, es necesario hacer personalizaciones de las áreas visuales comunes Ej: <i>Incluir imágenes de fondo, cambiar colores, tipos de letra, etc en AVs compartidas por la mayor parte de las páginas del portal.</i>  <b>NOTA</b> Hay otros métodos para hacer estas personalizaciones de áreas visuales <b>mucho más óptimos</b> (ver más adelante): <ul style="list-style-type: none"><li>• Utilizar la administración de áreas visuales</li><li>• Incluir estilos particulares en el área visual</li></ul>

Separar los estilos comunes del portal es muy conveniente ya que de esta forma es fácil hacer un cambio de estilo que rápidamente se aplicará a todo el portal sin necesidad de ir página por página (plantilla por plantilla).

#### Estilos particulares de plantilla (o página)

Son estilos particulares de la plantilla y por tanto de las páginas generadas a partir de ella. Dentro de los estilos **particulares de la plantilla** hay tres tipos:

<b>Layout</b> (disposición) de áreas visuales de la plantilla	Estructuran (disponen / colocan) las <b>áreas visuales de la plantilla</b> dentro de la página.
<b>Visuales de zonas de la página</b>	Personalizan la visualización de las <b>zonas de la plantilla</b> Ej: <i>Incluir imágenes de fondo o bordes en zonas de la página</i>
<b>Personalización de áreas visuales comunes</b>	En ocasiones, es necesario hacer personalizaciones de las áreas visuales de la página Ej: <i>Incluir imágenes de fondo, cambiar colores, tipos de letra, etc</i>  <b>NOTA</b> Hay otros métodos para hacer estas personalizaciones de áreas visuales <b>mucho más óptimos</b> (ver más adelante): <ul style="list-style-type: none"><li>• Utilizar la administración de áreas visuales</li><li>• Incluir estilos particulares en el área visual</li></ul>

#### Estilos comunes de las áreas visuales

Las áreas visuales tienen estilos comunes (base) que deben incluirse siempre ya que establecen estilos de visualización fijos (independientes de la parametrización)

#### Estilos personalizados de las áreas visuales

Cada área visual puede personalizarse visualmente lo que finalmente genera un fichero de estilo particular para cada instancia de área visual.

Por cuestiones de **rendimiento** (evitar la carga de excesivos ficheros de estilo), los ficheros de estilo con las personalizaciones de todas las áreas visuales de la página se "juntan" en un **único fichero**.

En el siguiente ejemplo se pueden ver las hojas de estilo incluidas en una página de portal:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
  <title>Euskadi.net - P&aacute;gina de inicio</title>

  <!-- **** MetaTags **** -->
  <meta name="Publisher" content="Ejje" />
  <meta name="Description" />
  <meta name="Keywords" />
  <meta name="Author" />
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
  <meta http-equiv="Content-Style-Type" content="text/css" />

  <!-- **** RESET StyleSheet **** -->
  <link rel="stylesheet" type="text/css" href="/AVComun/styles/r01resetToBas.es.css" />

  <!-- **** StyleSheet resources for all portal **** -->
  <link rel="stylesheet" type="text/css" href="/styles/r33-general_euskadi.css" />

  <!-- **** StyleSheet resources for this page **** -->
  <link rel="stylesheet" type="text/css" href="/styles/r33-home.css" />

  <!-- **** StyleSheet resources for all AV instances **** -->
  <link rel="stylesheet" type="text/css" href="/styles/r33-2220VAStyles.scss" />
</head>
<body class="r01PrintablePage">
  .....
</body>
</html>
```

Estilos de reseteo para establecer una base común a todos los navegadores

Estilos generales del portal R33

Estilos particulares de la plantilla de esta página

Estilos de las áreas visuales de la página

#### NOTA sobre los estilos de las AVs:

Como ya se ha mencionado, la **hoja de estilos de personalización de áreas visuales** incluye todos los estilos **generados** por el gestor de portales a partir de la parametrización del usuario.

Dado que si se incluye una referencia a cada hoja de estilos de cada AV individualmente, el navegador haría múltiples peticiones al servidor (al menos una por AV); para optimizar la carga de la página, el Gestor de Portales **agrupa** todas las hojas de estilo de todas las áreas visuales en un **fichero único**.

Este fichero único de estilos contiene:

- Los **estilos comunes fijos** para cada tipo de área visual.  
Estos ficheros están localizados en /AVComun/[AV\_type]/styles/
- Los **estilos particulares** de cada área visual generados a partir de la parametrización del usuario.  
Para cada área visual la herramienta de gestión de contenidos genera un fichero de estilos en el directorio: /[codPortal]/html/AV/[AV\_type]/styles/

En lugar de generar un único fichero, se “aprovecha” la capacidad SSI (*Server Side Include*) del servidor web apache que “compone” “al vuelo” un único fichero en base a los “trozos” de ficheros independientes.

De esta forma, la hoja de estilos con los estilos de las áreas visuales internamente es un conjunto de sentencias *include* individuales como

```
<!--#include virtual='/AVComun/[AV_type]/styles/r01[AVType].css' -->
<!--#include virtual='/styles/[AV_type]/[codPortal]-[AV_instance_name].css' -->
```

Para que el filtro SSI del servidor web apache “sepa” que ha de procesar este fichero, se le pone la extensión **“scss”** (*server side CSS*). Finalmente, el navegador (el cliente) “ve” una hoja de estilos “normal”

### 3.3.1. Hojas de estilo de reseteo de navegadores

El objetivo de la hoja de estilos de reseteo del navegador es reducir las inconsistencias entre diferentes navegadores (IE, Firefox, Chrome, Opera, etc) estableciendo una base común en cosas como tipos de letra, altos de línea, márgenes, etc

La razón básica es que todos los navegadores tienen una “presentación por defecto”, el problema es que no todos tienen la misma.

### 3.3.2. Hojas de estilo generales de un portal

Las hojas de estilo generales de un portal tienen por objeto definir estilos en AVs comunes del portal: aquellas AVs que **se comparten por todas las páginas (plantillas) del portal**, por ejemplo cabecera y pie

**Normalmente solo hay una hoja de estilos generales**

Por ejemplo en Euskadi.net la parte superior (cabecera) y la inferior (pie) es la misma en todas las páginas de portal. Poniendo como ejemplo la plantilla de la página home, a continuación se señalan las AVs que se comparten en todo el portal:



En la plantilla son:

En esta hoja de estilos se define la disposición de las AVs comunes en el portal

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
<head>
...
<link rel="stylesheet" type="text/css" href="/styles/r33-home.css" />
<link rel="stylesheet" type="text/css" href="/styles/r33-general.css" />
...
</head>
<body class="r01PrintablePage">
<!-- ===== LOGOTIPOS ===== -->
<div class="r33_headerLogos r01Printable r01clearfix">
<div class="r33_headerLogoLeft">
<!--@avDef id="1001"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
</div>
<div class="r33_headerLogoCenter">
<!--@avDef id="1002"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
</div>
<div class="r33_headerLogoRight">
<!--@avDef id="1003"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
</div>
</div>
<!-- ===== PRIMERA LINEA DE LA CABECERA ===== -->
<div class="r33_headerRow1 r01clearfix">
<div class="r33_headerHomePageButton">
<!--@avDef id="1004"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
</div>
<div class="r33_headerRow1Menu">
<!--@avDef id="1005"--><!--#include virtual="/AVTemplates/r01gHorizontalMenuVA/r01gHorizontalMenuVA_$LANG.html"-->
</div>
</div>
<!-- ===== SEGUNDA LINEA DE LA CABECERA ===== -->
<div class="r33_headerRow2 r01clearfix">
<div class="r33_headerRow2LangSel">
<!--@avDef id="1006"--><!--#include virtual="/AVTemplates/r01gLangSelectorVA/r01gLangSelectorVA_$LANG.html"-->
</div>
<div class="r33_headerRow2Menu">
<!--@avDef id="1007"--><!--#include virtual="/AVTemplates/r01gHorizontalMenuVA/r01gHorizontalMenuVA_$LANG.html"-->
</div>
</div>
<!-- ===== TERCERA LINEA DE LA CABECERA ===== -->
<div class="r33_headerRow3 r01clearfix">
<!--@avDef id="1008"--><!--#include virtual="/AVTemplates/r01gNavigatorVA/r01gNavigatorVA_$LANG.html"-->
</div>
<!-- ===== BLOQUE CENTRAL: MENU IZQ, CUERPO Y MENU DERECHO (A)===== -->
<div class="r33_body r01clearfix">
<!-- Columna izquierda -->
<div class="r33_bodyLeftCol">
<div class="r33_fullTextSearch">
<!--@avDef id="7"--><!--#include virtual="/AVTemplates/r01gSearchEngineVA/r01gSearchEngineVA_$LANG.html"-->
</div>
<div class="r33_bodyLeftColVMenu1">
<!--@avDef id="8"--><!--#include virtual="/AVTemplates/r01gVerticalMenuVA/r01gVerticalMenuVA_$LANG.html"-->
</div>
<div class="r33_bodyLeftColBanner1">
<!--@avDef id="9"--><!--#include virtual="/AVTemplates/r01gBannerVA/r01gBannerVA_$LANG.html"-->
</div>
<div class="r33_bodyLeftColVMenu2">
<!--@avDef id="10"--><!--#include virtual="/AVTemplates/r01gVerticalMenuVA/r01gVerticalMenuVA_$LANG.html"-->
</div>
<div class="r33_bodyLeftColBanner2">
<!--@avDef id="11"--><!--#include virtual="/AVTemplates/r01gBannerVA/r01gBannerVA_$LANG.html"-->
</div>
<div class="r33_bodyLeftColBanner3">
<!--@avDef id="12"--><!--#include virtual="/AVTemplates/r01gBannerVA/r01gBannerVA_$LANG.html"-->
</div>
<div class="r33_bodyLeftColImg">
<!--@avDef id="13"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
</div>
</div>
<div class="r33_bodyCenterCol">
<!-- Columna central -->
<div class="r33_bodyCenterColBanner">
<!--@avDef id="14"--><!--#include virtual="/AVTemplates/r01gBannerVA/r01gBannerVA_$LANG.html"-->
</div>
<div class="r33_bodyCenterColHeadLine">
<!--@avDef id="15"--><!--#include virtual="/AVTemplates/r01gHeadLineVA/r01gHeadLineVA_$LANG.html"-->
</div>
<div class="r33_bodyCenterColContentList">
<!--@avDef id="16"--><!--#include virtual="/AVTemplates/r01gContentListVA/r01gContentListVA_$LANG.html"-->
</div>
</div>
<div class="r33_bodyRightCol">
<!-- Columna derecha -->
<div class="r33_bodyRightColVerticalMenu">
<!--@avDef id="17"--><!--#include virtual="/AVTemplates/r01gVerticalMenuVA/r01gVerticalMenuVA_$LANG.html"-->
</div>
<div class="r33_bodyRightColContentList">
<!--@avDef id="18"--><!--#include virtual="/AVTemplates/r01gContentListVA/r01gContentListVA_$LANG.html"-->
</div>
</div>
</div>
<!-- ===== PIE ===== -->
<div class="r33_footer r01Printable">
<!--@avDef id="2000"--><!--#include virtual="/AVTemplates/r01gPageFootVA/r01gPageFootVA_$LANG.html"-->
</div>
</body>
</html>
```

Por lo tanto en la hoja de **estilos r33-general.css** se definirán los estilos comunes del portal:

```
<div class="r33_headerLogos r01Printable r01clearfix">
  <div class="r33_headerLogoLeft"> <!-- LOGOTIPOS DE LA CABECERA -->
    <!--@avDef id="1001"--><!--include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
  </div>
  <div class="r33_headerLogoCenter">
    <!--@avDef id="1002"--><!--include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
  </div>
  <div class="r33_headerLogoRight">
    <!--@avDef id="1003"--><!--include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
  </div>
</div>

/* ===== Logotipos de la cabecera ===== */
div.r33_headerLogos {
  width : 100%;
  padding-bottom : 5px;
  border-bottom : 1px solid #ccc;
}
div.r33_headerLogoLeft {
  float : left;
  text-align : left;
  margin-right : 10px;
}
div.r33_headerLogoCenter {
  float : left;
  text-align : left;
}
div.r33_headerLogoRight {
  float : right;
  text-align : right;
}

<div class="r33_headerRow1 r01clearfix"> <!-- PRIMERA LINEA DE LA CABECERA -->
  <div class="r33_homePageButton">
    <!--@avDef id="1004"--><!--include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
  </div>
  <div class="r33_headerRow1Menu">
    <!--@avDef id="1005"--><!--include virtual="/AVTemplates/r01gHorizontalMenuVA/r01gHorizontalMenuVA_$LANG.html"-->
  </div>
</div>

div.r33_headerRow1 {
  width : 100%;
  border-bottom : 3px solid white;
  border-top : 2px solid white;
}
div.r33_homePageButton {
  float : left;
  width : 8%;
}
div.r33_headerRow1Menu {
  float : right;
  width : 92%; /* no llegar al 100% junto con el boton (bug explorer) */
}

<div class="r33_headerRow2 r01clearfix"> <!-- SEGUNDA LINEA DE LA CABECERA -->
  <div class="r33_headerRow2LangSel">
    <!--@avDef id="1006"--><!--include virtual="/AVTemplates/r01gLangSelectorVA/r01gLangSelectorVA_$LANG.html"-->
  </div>
  <div class="r33_headerRow2Menu">
    <!--@avDef id="1007"--><!--include virtual="/AVTemplates/r01gHorizontalMenuVA/r01gHorizontalMenuVA_$LANG.html"-->
  </div>
</div>

div.r33_headerRow2 {
  width : 100%;
  border-top : 1px solid #ccc;
  border-bottom : 1px solid #ccc;
  padding-top : 2px;
}
div.r33_headerRow2LangSel {
  float : left;
  text-align : left;
  padding-top : 2px;
  margin-left : 2px;
  width : 35%;
}
div.r33_headerRow2Menu {
  float : right;
  width : 60%; /* no llegar a 100% junto con el selector de idioma (bug explorer) */
}

<div class="r33_headerRow3 r01clearfix"> <!-- TERCERA LINEA DE LA CABECERA -->
  <!--@avDef id="1008" --><!--include virtual="/AVTemplates/r01gNavigatorVA/r01gNavigatorVA_$LANG.html"-->
</div>

div.r33_headerRow3 {
  text-align : left;
  width : 100%;
  margin-top : 5px;
  margin-bottom : 10px;
}

<div class="r33_footer r01Printable"> <!-- PIE DE PAGINA -->
  <!--@avDef id="2000"--><!--include virtual="/AVTemplates/r01gPageFootVA/r01gPageFootVA_$LANG.html"-->
</div>

/* ===== Pie de pagina ===== */
div.r33_footer {
  _margin-top : 10px;
  width : 100%;
}
```

### 3.3.3. Hojas de estilo particulares de las páginas generadas a partir de una plantilla

Las hojas de estilo **particulares de cada una de las plantillas** (y por lo tanto de las páginas generadas a partir de esta).

**Normalmente existirá una sola hoja de estilo particular por cada una de las plantillas**

Por ejemplo la home de Euskadi.net difiere del resto de páginas del portal en la parte central (el cuerpo): menús, banners, destacados, noticias, etc, tal y como se señala a continuación:



The screenshot displays the Euskadi.net homepage, which is the portal for the Basque Government. The layout is structured as follows:

- Header:** Includes the Euskadi.net logo, navigation links (Contacto, Ayuda, Mapa, Accesibilidad, Mis gestiones), and language options (su, en, fr, de).
- Left Sidebar:** Contains a search bar, a menu for "Para ciudadanos" (For citizens), and a list of links including "Lehendakari", "Gobierno Vasco", "Otros organismos del Gobierno Vasco", "Otras instituciones", "Trámites", "Boletines oficiales", and "Portales temáticos".
- Main Content Area:**
  - Aurrera begira:** A banner for the Basque Government's vision of the future.
  - Servicios:** A section with icons for "Cita médica previa", "Ikasten. Cursos de euskera online", "Estado del tráfico", and "Ofertas de empleo público".
  - Destacados:** A section titled "Gripe" (Flu) with a date of 20/08/2009, and "Lugares excepcionales para realizar rutas a pie, en bicicleta o en coche. Usted elige." (Exceptional places for walking, cycling, or driving routes. You choose.) with a date of 14/08/2009.
  - Trámites más utilizados:** A section titled "Ayudas y subvenciones" (Aids and subsidies) with a date of 20/08/2009, listing various aid programs.
- Right Sidebar:**
  - Lo más buscado:** A list of popular searches including "Ayudas BOPV", "Catálogo industrial", "Cita médica previa", "Convenios EGA", "Euskera IT", "Txartela ONA", "OPE", and "Subvenciones".
  - Portales temáticos:** A list of thematic portals including "Berdingune", "Etxebide", "Euskara", "Gazteaukera", "GeoEuskadi", "Hiru.com", "Justizia.net", "Lanbide.net", "Csanet", "Sociedad de la información", and "Turismo".
  - RSS:** A section for RSS feeds including "Notas de prensa del Gobierno Vasco", "Noticias sobre turismo", "Contratación pública electrónica", "Noticias para la juventud (Gazteaukera)", and "Ofertas de empleo público (IVAP)".
  - Enlaces externos:** A list of external links including "Unión europea", "Parlamento europeo", "Tribunal de justicia europeo", "Gobierno de Navarra", "Parlamento de Navarra", "Gobierno español", "Parlamento español", "Seguridad social", "INEM", and "Gobierno francés".

## En la plantilla:

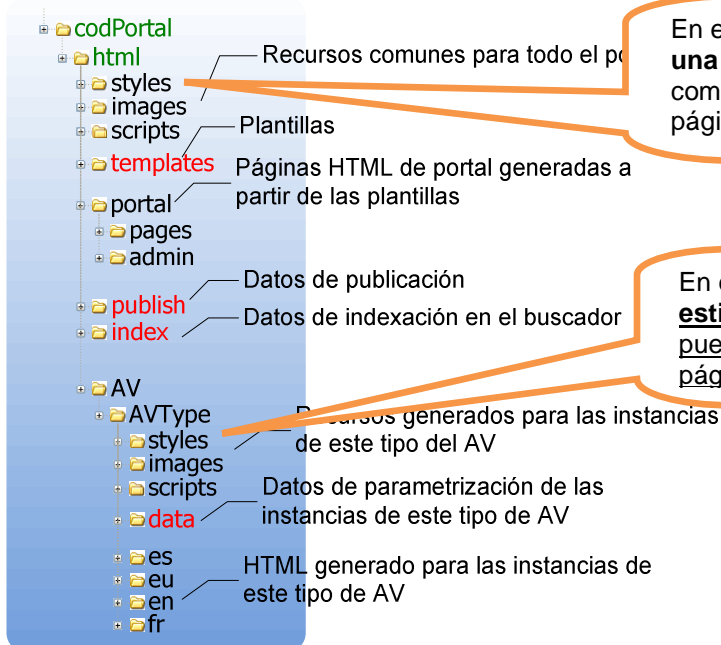
En esta hoja de estilos se define la disposición de las AVs particulares de esta plantilla

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
<head>
  <!-- Directiva para la previsualización de la plantilla, se borrará de las páginas -->
  <!--@avDef id="1"--><!--#include virtual="/AVTemplates/r01gCommonVA/r01gCommonVA_$LANG.html"-->
  <!-- Area visual que contiene parametros del head en cada idioma -->
  <!--@avDef id="0"--><!--#include virtual="/AVTemplates/r01gHeadVA/r01gHeadVA_$LANG.html"-->
  <link rel="stylesheet" type="text/css" href="/styles/r33-home.css" />
  <link rel="stylesheet" type="text/css" href="/styles/r33-general.css" />
</head>
<body class="r01PrintablePage">
  <!-- ===== LOGOTIPOS ===== -->
  <div class="r33_headerLogos r01Printable r01clearfix">
    <div class="r33_headerLogoLeft">
      <!--@avDef id="1001"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
    </div>
    <div class="r33_headerLogoCenter">
      <!--@avDef id="1002"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
    </div>
    <div class="r33_headerLogoRight">
      <!--@avDef id="1003"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
    </div>
  </div>
  <!-- ===== PRIMERA LINEA DE LA CABECERA ===== -->
  <div class="r33_headerRow1 r01clearfix">
    <div class="r33_homePageButton">
      <!--@avDef id="1004"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
    </div>
    <div class="r33_headerRow1Menu">
      <!--@avDef id="1005"--><!--#include virtual="/AVTemplates/r01gHorizontalMenuVA/r01gHorizontalMenuVA_$LANG.html"-->
    </div>
  </div>
  <!-- ===== SEGUNDA LINEA DE LA CABECERA ===== -->
  <div class="r33_headerRow2 r01clearfix">
    <div class="r33_headerRow2LangSel">
      <!--@avDef id="1006"--><!--#include virtual="/AVTemplates/r01gLangSelectorVA/r01gLangSelectorVA_$LANG.html"-->
    </div>
    <div class="r33_headerRow2Menu">
      <!--@avDef id="1007"--><!--#include virtual="/AVTemplates/r01gHorizontalMenuVA/r01gHorizontalMenuVA_$LANG.html"-->
    </div>
  </div>
  <!-- ===== TERCERA LINEA DE LA CABECERA ===== -->
  <div class="r33_headerRow3 r01clearfix">
    <!--@avDef id="1008"--><!--#include virtual="/AVTemplates/r01gNavigatorVA/r01gNavigatorVA_$LANG.html"-->
  </div>
  <!-- ===== BLOQUE CENTRAL: MENU IZO, CUERPO Y MENU DERECHO (A)===== -->
  <div class="r33_body r01clearfix">
    <!-- Columna izquierda -->
    <div class="r33_bodyLeftCol">
      <div class="r33_fullTextSearch">
        <!--@avDef id="7"--><!--#include virtual="/AVTemplates/r01gSearchEngineVA/r01gSearchEngineVA_$LANG.html"-->
      </div>
      <div class="r33_bodyLeftColVMenu1">
        <!--@avDef id="8"--><!--#include virtual="/AVTemplates/r01gVerticalMenuVA/r01gVerticalMenuVA_$LANG.html"-->
      </div>
      <div class="r33_bodyLeftColBanner1">
        <!--@avDef id="9"--><!--#include virtual="/AVTemplates/r01gBannerVA/r01gBannerVA_$LANG.html"-->
      </div>
      <div class="r33_bodyLeftColVMenu2">
        <!--@avDef id="10"--><!--#include virtual="/AVTemplates/r01gVerticalMenuVA/r01gVerticalMenuVA_$LANG.html"-->
      </div>
      <div class="r33_bodyLeftColBanner2">
        <!--@avDef id="11"--><!--#include virtual="/AVTemplates/r01gBannerVA/r01gBannerVA_$LANG.html"-->
      </div>
      <div class="r33_bodyLeftColBanner3">
        <!--@avDef id="12"--><!--#include virtual="/AVTemplates/r01gBannerVA/r01gBannerVA_$LANG.html"-->
      </div>
      <div class="r33_bodyLeftColImg">
        <!--@avDef id="13"--><!--#include virtual="/AVTemplates/r01gImageVA/r01gImageVA_$LANG.html"-->
      </div>
    </div>
    <!-- Columna central -->
    <div class="r33_bodyCenterCol">
      <div class="r33_bodyCenterColBanner">
        <!--@avDef id="14"--><!--#include virtual="/AVTemplates/r01gBannerVA/r01gBannerVA_$LANG.html"-->
      </div>
      <div class="r33_bodyCenterColHeadLine">
        <!--@avDef id="15"--><!--#include virtual="/AVTemplates/r01gHeadLineVA/r01gHeadLineVA_$LANG.html"-->
      </div>
      <div class="r33_bodyCenterColContentList">
        <!--@avDef id="16"--><!--#include virtual="/AVTemplates/r01gContentListVA/r01gContentListVA_$LANG.html"-->
      </div>
    </div>
    <!-- Columna derecha -->
    <div class="r33_bodyRightCol">
      <div class="r33_bodyRightColVMenu">
        <!--@avDef id="17"--><!--#include virtual="/AVTemplates/r01gVerticalMenuVA/r01gVerticalMenuVA_$LANG.html"-->
      </div>
      <div class="r33_bodyRightColContentList">
        <!--@avDef id="18"--><!--#include virtual="/AVTemplates/r01gContentListVA/r01gContentListVA_$LANG.html"-->
      </div>
    </div>
  </div>
  <!-- ===== PIE ===== -->
  <div class="r33_footer r01Printable">
    <!--@avDef id="2000"--><!--#include virtual="/AVTemplates/r01gPageFootVA/r01gPageFootVA_$LANG.html"-->
  </div>
</body>
</html>
```

### 3.3.4. Estilos particulares de las Áreas Visuales

La herramienta de gestión de portales se basa en generar código estático HTML (html, css, js) en base a la parametrización del usuario, pero ¿Dónde están los estilos generados?

Observando la estructura de un portal:



En esta carpeta el Gestor de Portales **genera una hoja de estilos para cada página** que combina las hojas de estilo de cada AV de la página

En esta carpeta hay **una hoja de estilos por cada instancia de AV** que puede estar compartida en varias páginas

Cada AV tiene su propia hoja de estilo como se ha descrito anteriormente; dado que un AV puede estar en varias páginas, todas ellas deberían incluir los estilos del AV en el head de la siguiente forma:

```
<head>
<meta name="DC.creator" content="CO00417L" >
<meta name="DC.language" content="es" >
<meta name="DC.date" content="20/08/2009" >
<meta name="DC.identifier" content="r01dpd012336ffd8f4195873a744e2e1f30264711" >
<meta name="DC.title" content="Home de Euskadi.net" >
<meta name="DC.description" content="Home de Euskadi.net" >

<title>Euskadi.net - Administración de la Comunidad Autónoma del País Vasco</title>

<!-- *** MetaTags *** -->
<meta name="Publisher" content="Ejje" />
<meta name="Description" lang="es" content="Red de portales Euskadi.net" />
<meta name="Keywords" lang="es" content="Euskadi, Euskadi.net" />
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta http-equiv="Content-Script-Type" content="text/javascript" />
<meta http-equiv="Content-Style-Type" content="text/css" />

<!-- *** StyleSheet resources for all AV instances *** -->
<link rel='stylesheet' type='text/css' href='/styles/r33-headImage1VA.css' />
<link rel='stylesheet' type='text/css' href='/styles/r33-headHMenu1VA.css' />
<link rel='stylesheet' type='text/css' href='/styles/r33-bodyLeftMenuVA.css' />
<link rel='stylesheet' type='text/css' href='/styles/r33-bodyHeadLineVA.css' />
etc etc etc
<link rel='stylesheet' type='text/css' href='/styles/r33-footVA.css' />

<!-- *** Script resources for all AV type *** -->
<script type='text/javascript' src='/scripts/r33-homeVAScripts.sjs'></script>

<link rel="stylesheet" type="text/css" href="/styles/r33-home.css" />
<link rel="stylesheet" type="text/css" href="/styles/r33-general.css" />
</head>
```

Hojas de estilo y scripts generados por el Gestor de Portales para cada AV

Hojas de estilo que "vienen" de la plantilla utilizada

El problema es que **NO es óptimo** incluir tantos ficheros css individuales ya que el navegador tiene que hacer demasiadas peticiones al servidor. La solución es **combinar todos los ficheros de estilos individuales de cada AV en un único fichero** que obviamente debería ser para cada página:

Si se visualiza el código fuente generado para una página cualquiera de portal y se observa la sección **<head>** de la página:

```
<head>
<meta name="DC.creator" content="CO00417L" >
<meta name="DC.language" content="es" >
<meta name="DC.date" content="20/08/2009" >
<meta name="DC.identifier" content="r01dpd012336ffd8f4195873a744e2elf30264711" >
<meta name="DC.title" content="Home de Euskadi.net" >
<meta name="DC.description" content="Home de Euskadi.net" >

<title>Euskadi.net - Administraci&ocaron; de la Comunidad Aut&ocaron;noma del Pa&iacute;s Vasco</title>

<!-- **** MetaTags **** -->
<meta name="Publisher" content="Ejje" />
<meta name="Description" lang="es" content="Red de portales Euskadi.net" />
<meta name="Keywords" lang="es" content="Euskadi, Euskadi.net" />
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<meta http-equiv="Content-Script-Type" content="text/javascript" />
<meta http-equiv="Content-Style-Type" content="text/css" />

<!-- **** StyleSheet resources for all AV instances **** -->
<link rel="stylesheet" type="text/css" href="/styles/r33-homeVASTyles.scss" />

<!-- **** Script resources for all AV type **** -->
<script type="text/javascript" src="/scripts/r33-homeVAScripts.sjs"></script>

<link rel="stylesheet" type="text/css" href="/styles/r33-home.css" />
<link rel="stylesheet" type="text/css" href="/styles/r33-general.css" />
</head>
```

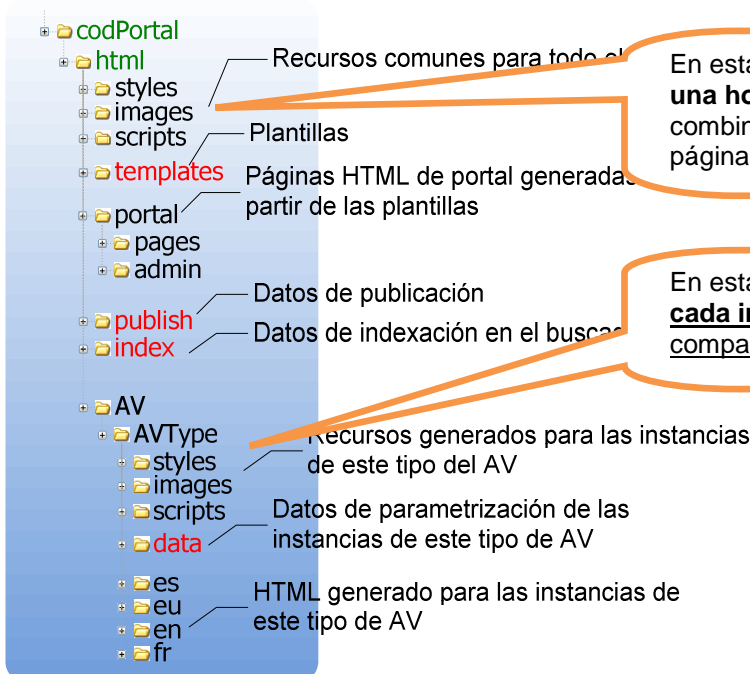
Una única hoja de estilo generada por el Gestor de Portales combinando los estilos de todas las AVs

Hojas de estilo que “vienen” de la plantilla utilizada

El Gestor de Portales **automáticamente genera una hoja de estilos que combina las hojas de estilo individuales de cada área visual** en un único fichero para optimizar la carga de la página.

El nombre de esta hoja de estilo “generada” a partir de las hojas de estilo de las AVs de la página es:

**[codPortal]-[nombrePágina]VASTyles.scss**



En esta carpeta el Gestor de Portales **genera una hoja de estilos para cada página** que combina las hojas de estilo de cada AV de la página

En esta carpeta hay **una hoja de estilos por cada instancia de AV** que puede estar compartida en varias páginas.

Basta abrir la hoja de estilos correspondiente a una página cualquiera (ej: r33-2220) para ver que está compuesta de los estilos individuales de cada AV:

```
@CHARSET "ISO-8859-1";
```

```
/* File: /AVComun/r0lgImageVA/styles/r0lgImage.css*/  
<!--#include virtual='/AVComun/r0lgImageVA/styles/r0lgImage.css'-->
```

```
/* File: /AV/r0lgImageVA/styles/r33-imgBotonIniciosincolor.css*/  
<!--#include virtual='/AV/r0lgImageVA/styles/r33-imgBotonIniciosincolor.css'-->
```

```
/* File: /AVComun/r0lgPageFootVA/styles/r0lgPageFoot.css*/  
<!--#include virtual='/AVComun/r0lgPageFootVA/styles/r0lgPageFoot.css'-->
```

```
/* File: /AV/r0lgPageFootVA/styles/r33-ftPiepaginaEuskadiEjje.css*/  
<!--#include virtual='/AV/r0lgPageFootVA/styles/r33-ftPiepaginaEuskadiEjje.css'-->
```

etc etc etc

El filtro ssi se encarga de procesar cada directiva include e insertar la hoja de estilos individual de cada AV.

### 3.3.5. Hojas de estilo del contenido / aplicación que se visualiza

Cuando la página de portal tiene un AV contenedora de contenido / aplicación (*container*), tal y como se ha explicado en el punto anterior, el Gestor de Portales “inyecta” una serie de estilos en la css de AVs de la página concreta: **[codPortal]-[nombrePágina]VAStyles.scss**

Los estilos correspondientes al **AV contenedor** de contenido / aplicación son los siguientes:

```

/*****
/* File: /AVComun/r01gContainerVA/styles/r01gTinyContent.css*/
/*****
<!--#include virtual='/AVComun/r01gContainerVA/styles/r01gTinyContent.css'-->

/*****
/* File: /AVComun/r01gContainerVA/styles/r01gHTMLEditor.css*/
/*****
<!--#include virtual='/AVComun/r01gContainerVA/styles/r01gHTMLEditor.css'-->

/*****
/* File: /AVComun/r01gContainerVA/styles/r01gContainer.css*/
/*****
<!--#include virtual='/AVComun/r01gContainerVA/styles/r01gContainer.css'-->

/*****
/* File: /AVComun/r01gContainerVA/styles/r01gContainerOther.scss*/
/*****
<!--#include virtual='/AVComun/r01gContainerVA/styles/r01gContainerOther.scss'-->

```

Como se puede ver todas estas hojas de estilo están en **AVComun**: son comunes a todos los portales y contenidos y son las siguientes:

<b>r01gTinyContent.css</b>	Estilos del editor HTML TinyMCE
<b>r01gHTMLEditor.css</b>	Estilos del editor HTML eWebEditPRO
<b>r01gContainer.css</b>	Estilos generales ( <b>comunes</b> ) de los contenidos de de la infraestructura (se aplican a cualquier tipo de contenido)
<b>r01gContainerOther.scss</b>	Estilos de <b>personalización</b> de los estilos generales (comunes) de los contenidos de la infraestructura: “customizan” los estilos del fichero <b>r01gContainer.css</b> para una iniciativa (ej: EHU) y se garantiza que estos estilos no se “machacan” cuando se instala una nueva versión de los gestores.

#### NOTA:

Estas hojas de estilo se incluyen ya que el Gestor de Portales utiliza la **configuración del AV container** (r01gContainerVA.xml) donde se indican los recursos comunes que utiliza el AV. Esta configuración está en la workArea de sistema en la carpeta **portal**

```

<commonResources>
  <resource type='styleSheet'>
    <fileName>r01gContainer.css</fileName>
    <description>
      <es><![CDATA[Fichero de estilos común para todas las áreas visuales r01gContainerVA]]></es>
      <eu><![CDATA[r01gContainerVA ikus arlo guztientzako estilo-fitxategi erkidea]]></eu>
    </description>
  </resource>
  <resource type='styleSheet'>
    <fileName>r01gHTMLEditor.css</fileName>
    <description>
      <es><![CDATA[Fichero de estilos del Webeditpro para todas las áreas visuales r01gContainerVA]]></es>
      <eu><![CDATA[Webeditpro-aren estilo-fitxak, r01gContainerVA ikus-eremu guztietarako]]></eu>
    </description>
  </resource>
  <resource type='styleSheet'>
    <fileName>r01gContainerOther.scss</fileName>
    <description>
      <es><![CDATA[estilos de otras iniciativas (turismo, ehu, etc)]]></es>
      <eu><![CDATA[Beste edukien estilo-fitxak (Turismoko, EHU)]]></eu>
    </description>
  </resource>
  <resource type='styleSheet'>
    <fileName>r01gTinyContent.css</fileName>
    <description>
      <es><![CDATA[Estilos para el Editor TinyMCE]]></es>
      <eu><![CDATA[E-Estilos para el Editor TinyMCE]]></eu>
    </description>
  </resource>
  <resource type='script'>
    <fileName>r01gContainerOther.sjs</fileName>
    <description>
      <es><![CDATA[scripts de los contenidos de otras iniciativas (turismo, ehu, etc)]]></es>
      <eu><![CDATA[Beste edukien script-fitxak (turismoko, ehu)]]></eu>
    </description>
  </resource>
</commonResources>

```

### 3.3.6. Estilos y scripts particulares por tipo (naturaleza) de contenido

Conforme la riqueza de tipos de contenido crece, el número de estilos a incluir en la hoja de estilos comunes de Euskadi.net `r01gContainer.css` crece, lo que hace que el fichero anterior sea cada vez más grande hasta llegar a ser inmanejable.

La solución es incluir estilos para los contenidos **dependiendo del tipo de contenido que se está visualizando**; de esta forma el tamaño de la hoja de estilos es mucho menor.

Cuando una página contiene un AV contenedor de contenidos / aplicaciones, el Gestor de Portales incluye en la sección `<head>` de la página fichero `r01gTypeResources.shtml` situado en `/AVCustom`

```
<head>
...
<!-- **** StyleSheet resources for all AV instances **** -->
<link rel='stylesheet' type='text/css' href='/styles/r33-homeVAStyles.scss' />

<!-- **** Script resources for all AV type **** -->
<script type='text/javascript' src='/scripts/r33-homeVAScripts.sjs'></script>

<link rel="stylesheet" type="text/css" href="/styles/r33-home.css" />
<link rel="stylesheet" type="text/css" href="/styles/r33-general.css" />

<!--#include virtual='/AVCustom/r01gContainerVA/r01gTypeResources.shtml' -->
...
</head>
```

Incluye estilos y scripts por tipo de contenido

Este fichero hace uso de variables que el servidor web (apache) establece automáticamente en función del tipo de contenido que se está visualizando:

```
<!--#if expr="$TYPE[ tipo_cont] = /^true/" -->
    Cualquier hoja de estilo o script necesario en función del tipo de contenido
<!--#endif -->
```

Por ejemplo si se visualiza un tipo de contenido evento:

```
<!--#if expr="$TYPE[ tipo_cont] = /^true/" -->
    <link rel="stylesheet" type="text/css" href="/AVComun/r01gContainerVA/styles/eventos/r01gEvento.css" />
    <script type="text/javascript" src="/AVComun/scripts/jquery.js"></script>
<!--#endif -->
```

### IMPORTANTE

Basta por lo tanto modificar la sección correspondiente al tipo de contenido en el fichero `r01gTypeResources.shtml` para incluir **estilos particulares** a este tipo de contenido.

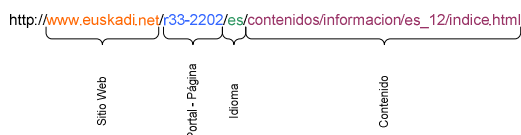
Pero ¿dónde se definen las variables apache **\$TYPE\_[tipo\_cont] = true**?

Hay dos casos:

El contenido a mostrar en el AV contenedora se recibe en la URL

Ej:

[http://www.euskadi.net/33-2202/es/contenidos/informacion/es\\_12/indice.html](http://www.euskadi.net/33-2202/es/contenidos/informacion/es_12/indice.html)



El servidor apache establece una variable con el tipo de contenido mostrado:

En el ejemplo: **\$TYPE\_informacion = true**

Esto hará que al procesar el fichero:

```
<!--#include virtual='/AVCustom/r01gContainerVA/r01gTypeResources.shtml' -->
```

se incluyan los estilos del tipo información

#### NOTA:

Hay que tener en cuenta que una página solo puede tener un AV container que muestre el contenido recibido en la URL

El contenido a mostrar en el AV contenedora se fija por administración: el contenido no se recibe en la URL sino que se pre-fija en el momento de administración de la pagina

En este caso, dado que se "sabe" de antemano el tipo de contenido a mostrar, cuando el Gestor de Portales genera el HTML de la página **incluye una variable para cada AV container**:

Por ejemplo si un AV container pre-fija una noticia y en otro un evento en la página habrá dos variables como:

```
<!--#set var="TYPE_noticia" value="true" -->
```

```
<!--#set var="TYPE_evento" value="true" -->
```

Esto hará que al procesar el fichero:

```
<!--#include virtual='/AVCustom/r01gContainerVA/r01gTypeResources.shtml' -->
```

se incluyan los estilos de noticias y eventos

#### NOTA:

Una página puede tener tantas AV container con el contenido pre-fijado como sea necesario (a diferencia del recibido por url que solo puede ser uno)

### 3.3.7. Personalización de estilos para una iniciativa

Tal y como se ha explicado, hay una hoja de estilos **común** para todos los contenidos generados con las herramientas de Gestion de Contenidos y portales: `/AVComun/r01gContainerVA/styles/r01gContainer.css`

Esta hoja de estilos está localizada en una carpeta del sistema (/AVComun) por lo que cuando se actualiza la versión de la infraestructura **se puede perder cualquier cambio que se haga en los ficheros que contiene.**

Esto es un gran inconveniente, así que para **personalizar** los estilos por defecto de la infraestructura, se proporciona la carpeta `/AVCustom` que se garantiza que no será modificada al instalar nuevas versiones.

Por lo tanto, si es necesario **personalizar** algún estilo para los contenidos respecto a los que la infraestructura pone por defecto en el fichero `/AVComun/r01gContainerVA/styles/r01gContainer.css` hay que hacerlo en el fichero `/AVCustom/r01gContainerVA/styles/r01gContainerCustom.css`

Este fichero se incluye cada vez que una página contiene un AV container ya que:

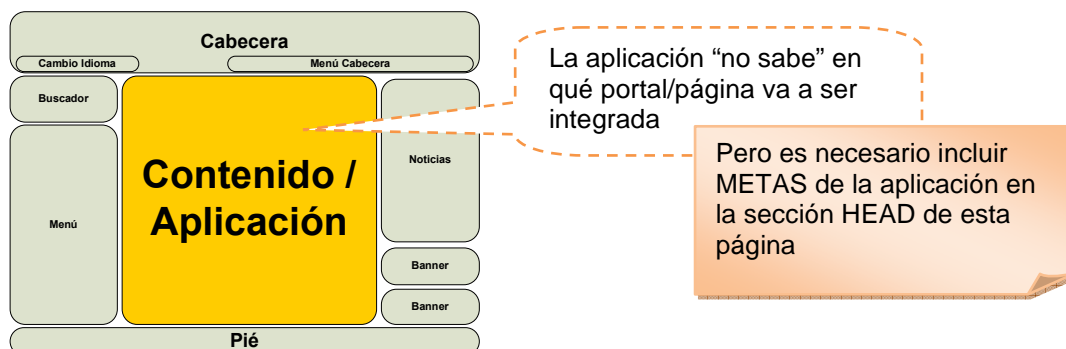
- La configuración del AV container indica que hay que incluir el estilo común `/AVComun/r01gContainerVA/styles/r01gContainerOther.scss`
- El fichero `/AVComun/r01gContainerVA/styles/r01gContainerOther.scss` a su vez incluye el fichero `/AVCustom/r01gContainerVA/styles/r01gContainerCustom.css`

**NOTA:** Esta funcionalidad es de utilidad principalmente para instalaciones de toda la infraestructura en terceros (ej: UPV / EHU) ya que garantiza que la instalación de una nueva versión de las aplicaciones no “machaca” customizaciones en los estilos.

### 3.3.8. Hojas de estilo, scripts y METAS particulares de contenidos en el HEAD de la página

Se presenta un problema si es necesario “inyectar” un script, hoja de estilo o meta perteneciente al contenido / aplicación en el HEAD de la página:

La página de portal es la que contiene el esqueleto del HTML, incluyendo la sección HEAD y el servidor web “simplemente” inyecta el contenido / aplicación en el lugar indicado, pero **¿cómo inyectar la referencia a un script, una hoja de estilos o unos METAS propios de la aplicación en la sección HEAD de la página?**



**Técnicamente**, dado que las páginas de portal **existen de forma separada** a los contenidos/aplicaciones, se presenta un problema cuando es necesario incluir recursos del contenido en el HEAD de la página del portal tal y como se ve en el código fuente de la plantilla de la página:

**Área visual de contenido.**  
**Puede “alojar” cualquier aplicación**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

<head>
    <title>Euskadi.net - Administración de la Comunidad Autónoma del País Vasco</title>
    <!-- **** MetaTags **** -->
    <meta name="Description" lang="es" content="Red de portales Euskiak para el País Vasco" />
    <meta name="Keywords" lang="es" content="Euskadi, Euskadi.net, País Vasco" />
    <meta name="Author" lang="es" content="Euskadi.net" />
    <meta http-equiv="Content-Type" content="text/html charset=utf-8" />
    <meta http-equiv="Content-Script-Type" content="text/javascript" />
    <meta http-equiv="Content-Style-Type" content="text/css" />

    <link rel="stylesheet" type="text/css" href="/styles/r33-2288VAsStyles.css" />
    <script type="text/javascript" src="/scripts/r33-2288VAsScripts.js" />
    <link rel="stylesheet" type="text/css" href="/styles/r33-general.css" />
    <link rel="stylesheet" type="text/css" href="/styles/r33-gener

</head>

<body class="r01PrintablePage">

<!-- ***** LOGOTIPOS ***** -->
<div class="r33_headerLogos r01Printable r01clearfix">
    <div class="r33_headerLogoLeft"><!-- ***** r01gImageVA ***** --></div>
</div>
<!-- ***** PRIMERA LINEA DE LA CABECERA ***** -->
<div class="r33_headerRow1 r01clearfix">
    <div class="r33_homePageButton"><!-- ***** r01gImageVA ***** --></div>
    <div class="r33_headerRow1Menu"><!-- ***** r01gHorizontalMenuVA ***** --></div>
</div>
<!-- ***** SEGUNDA LINEA DE LA CABECERA ***** -->
<div class="r33_headerRow2 r01clearfix">
    <div class="r33_headerRow2LangSel"><!-- ***** r01gLanSelectorVA ***** --></div>
    <div class="r33_headerRow2Menu"><!-- ***** r01gHorizontalMenuVA ***** --></div>
</div>
<!-- ***** TERCERA LINEA DE LA CABECERA ***** -->
<div class="r33_headerRow3 r01clearfix"><!-- ***** r01gNavigatorVA ***** --></div>
<!-- ***** BLOQUE CENTRAL: MENU IZQ Y CUERPO ***** -->
<div class="r33_body r01PrintableFullWidth r01clearfix">
    <!-- Columna izquierda -->
    <div class="r33_bodyLeftCol">
        <div class="r33_fullTextSearch"><!-- ***** r01gSearchEngineVA ***** --></div>
        <div class="r33_bodyLeftColMenu1"><!-- ***** r01gVerticalMenuVA ***** --></div>
        <div class="r33_bodyLeftColBanner1"><!-- ***** r01gBannerVA ***** --></div>
        <div class="r33_bodyLeftColMenu2"><!-- ***** r01gVerticalMenuVA ***** --></div>
        <div class="r33_bodyLeftColBanner2"><!-- ***** r01gBannerVA ***** --></div>
    </div>
    <!-- Columna central -->
    <div class="r33_bodyCenterCol">
        <div class="r33_bodyCenterColContainer"><!-- ***** r01gContainerVA ***** --></div>
    </div>
</div>
<!-- ***** FIRM ***** -->
<div class="r33_footer r01Printable"><!-- ***** r01gPageFootVA ***** --></div>

</body>
</html>
```

**La aplicación concreta es “inyectada” en tiempo de visualización de la página**

Si se observa el código fuente de la página en el servidor apache (antes de ser entregada al navegador):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

```
<head>
```

```
<!--#if expr="$PORTAL_VARS = /^true/" --><!--#include virtual="/pages/data/r01[PageName]PortalVars.shtml" --><!--#endif -->
<!--#if expr="$CONT_VARS = /^true/" --><!--#include virtual="$CONT_PATH/r01ContVars.shtml" --><!--#endif -->
```

```
<!--#if expr="$CONT_RESOURCES = /^true/" --><!--#include virtual="$CONT_PATH/r01ContHead.shtml" --><!--#endif -->
<!--#if expr="$PORTAL_RESOURCES = /^true/" --><!--#include virtual="/pages/data/r01[PageName]PortalHead.shtml" --><!--#endif -->
```

```
<!--#if expr="$PORTAL_METAS = /^true/" --><!--#include virtual="/pages/data/r01[PageName]PortalMetas.inc" --><!--#endif -->
<!--#if expr="$CONT_METAS = /^true/" --><!--#include virtual="$CONT_PATH/r01ContMetas.inc" --><!--#endif -->
```

```
<!--#if expr="$APP_RESOURCES = /^true/" --><!--#include virtual="$APP_PATH" --><!--#endif -->
```

```
<title>Euskadi.net - Administraci&ocaron;n de la Comunidad Aut&ocaron;noma&iacute;s Vasco</title>
```

```
<!-- **** MetaTags **** -->
```

```
<meta name="Description" lang="es" content="Red de portales Euskadi.net">
```

```
<meta name="Keywords" lang="es" content="Euskadi, Euskadi.net">
```

```
<meta name="Author" lang="es" content="Euskadi.net"/>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<meta http-equiv="Content-Script-Type" content="text/javascript">
```

```
<meta http-equiv="Content-Style-Type" content="text/css">
```

```
<link rel="stylesheet" type="text/css" href="/stylesheets/main.css">
```

```
<script type="text/javascript" src="/scripts/r33-2288.js"></script>
```

```
<link rel="stylesheet" type="text/css" href="/stylesheets/main.css">
```

```
<link rel="stylesheet" type="text/css" href="/stylesheets/main.css">
```

```
</head>
<body class="r01PrintablePage">
```

```
<!-- ===== LOGOTIPOS ===== -->
```

```
<div class="r33_headerLogos r01Printable r01clearfix">
```

```
<div class="r33_headerLogoLeft"><!-- ***** r01gLogoVA ***** --></div>
```

```
</div>
```

```
<!-- ===== PRIMERA LINEA DE LA CABECERA ===== -->
```

```
<div class="r33_headerRow1 r01clearfix">
```

```
<div class="r33_homePageButton"><!-- ***** r01gImageVA ***** --></div>
```

```
<div class="r33_headerRow1Menu"><!-- ***** r01gHorizontalMenuVA ***** --></div>
```

```
</div>
```

```
<!-- ===== SEGUNDA LINEA DE LA CABECERA ===== -->
```

```
<div class="r33_headerRow2 r01clearfix">
```

```
<div class="r33_headerRow2LangSel"><!-- ***** r01gLangSelectorVA ***** --></div>
```

```
<div class="r33_headerRow2Menu"><!-- ***** r01gHorizontalMenuVA ***** --></div>
```

```
</div>
```

```
<!-- ===== TERCERA LINEA DE LA CABECERA ===== -->
```

```
<div class="r33_headerRow3 r01clearfix"><!-- ***** r01gNavigatorVA ***** --></div>
```

```
<!-- ===== BLOQUE CENTRAL: MENU IZQ Y CUERPO ===== -->
```

```
<div class="r33_body r01PrintableFullWidth r01clearfix">
```

```
<!-- Columna izquierda -->
```

```
<div class="r33_bodyLeftCol">
```

```
<div class="r33_fullTextSearch"><!-- ***** r01gSearchEngineVA ***** --></div>
```

```
<div class="r33_bodyLeftColVMenu1"><!-- ***** r01gVerticalMenuVA ***** --></div>
```

```
<div class="r33_bodyLeftColBanner1"><!-- ***** r01gBannerVA ***** --></div>
```

```
<div class="r33_bodyLeftColVMenu2"><!-- ***** r01gVerticalMenuVA ***** --></div>
```

```
<div class="r33_bodyLeftColBanner2"><!-- ***** r01gBannerVA ***** --></div>
```

```
</div>
```

```
<!-- Columna central -->
```

```
<div class="r33_bodyCenterCol">
```

```
<div class="r33_bodyCenterColContainer"><!-- ***** r01gContainerVA ***** --></div>
```

```
</div>
```

```
</div>
```

```
<!-- ===== PIE ===== -->
```

```
<div class="r33_footer r01Printable"><!-- ***** r01gPageFootVA ***** --></div>
```

```
</body>
```

```
</html>
```

La infraestructura “inyecta” cuatro ficheros:

**Contenidos:** r01ContVars.shtml

r01ContMetas.inc

r01ContHead.shtml

**Portal:**

r01[PageName]PortalVars.shtml

r01[PageName]PortalMetas.inc

r01[PageName]PortalHead.shtml

**Aplicaciones:** Un fichero arbitrario

La infraestructura según se observa “inyecta” cinco ficheros con propósito distinto y a la vez complementario:

<b>Contenidos</b>	<b>r01ContVars.shtml</b>	Variables APACHE con metadatos del contenido que pueden ser utilizadas en cualquier sitio
	<b>r01ContMetas.inc</b>	METAS (en formato HTML dublicore) del contenido para ser incluidos en el HEAD de la página
	<b>r01ContHead.shtml</b>	Recursos (scripts y estilos) propios del contenido para ser incluidos en el HEAD de la página
<b>Portales</b>	<b>r01[PageName]PortalVars.shtml</b>	Variables APACHE con metadatos de la página de portal que pueden ser utilizadas en cualquier sitio
	<b>r01[PageName]PortalMetas.inc</b>	METAS, scripts y estilos de la página de portal para ser incluidos en el HEAD de la página de portal
	<b>r01[PageName]PortalHead.shtml</b>	Recursos (scripts y estilos) propios del contenido para ser incluidos en el HEAD de la página
<b>Aplicaciones</b>	Un fichero arbitrario que incluye el HTML que hay que insertar en el HEAD de la página.	

Como se ve hay tres tipos de ficheros:

<b>Variables</b>	xxxVars.shtml	<p>Es un fichero <b>shtml</b> (es procesado por el filtro SSI de apache) que contiene <b>definiciones de variables apache</b> en la forma:</p> <pre>&lt;!--#set var="doc_author" value="EJ03680M" --&gt; &lt;!--#set var="cont_name" value="prensa_sanidad79" --&gt; &lt;!--#set var="doc_lang" value="es" --&gt; &lt;!--#set var="cont_author" value="EJ03680M" --&gt; &lt;!--#set var="doc_name" value="My doc name" --&gt; &lt;!--#set var="cont_description" value="My doc description" --&gt; &lt;!--#set var="doc_description" value="Nota de Prensa. My doc description" --&gt; &lt;!--#set var="doc_internal_name" value="np79" --&gt;</pre> <p>Como se ve, simplemente son metadatos del contenido / página expresados en forma de variables apache que pueden utilizarse en cualquier lugar del HTML de las páginas (y de los contenidos) en la forma:</p> <pre>&lt;!--#echo var="[nombre variable]" --&gt;</pre> <p>Por ejemplo para incluir el nombre del contenido en el title de la página:</p> <pre>&lt;title&gt; Euskadi.net - &lt;!--#if expr="\$doc_name" --&gt;&lt;!--#echo var="doc_name" --&gt;&lt;!--#endif --&gt; &lt;/title&gt;</pre> <p><b>Estos ficheros son generados automáticamente por el Gestor de Contenidos y Portales.</b></p>
<b>MetaDatos</b>	xxxMetas.inc	<p>Es un fichero <b>inc</b> (no es procesado por el filtro SSI de apache) que contiene METAS en formato HTML dublin-core en el formato:</p> <pre>&lt;meta name="DC.creator" content="EJ03680M" &gt; &lt;meta name="DC.language" content="es" &gt; &lt;meta name="DC.date" content="11/09/2006" &gt; &lt;meta name="DC.identifier" content="r01dp010d9c8c004683e047a6a10aedaccd245"&gt; &lt;meta name="DC.title" content="My description" &gt; &lt;meta name="DC.description" content="Nota de Prensa. My Description" &gt;</pre> <p><b>Estos ficheros son generados automáticamente por el Gestor de Contenidos y Portales</b> y se incluyen automáticamente en el HEAD de las páginas que tienen un AV contenedora de contenidos</p>
<b>Recursos del Head</b>	xxxHead.shtml	<p>Es un fichero <b>shtml</b> (es procesado por el filtro SSI de apache) que contiene <b>recursos (scripts, hojas de estilo y metas)</b> que se han de incluir en la sección HEAD de la página.</p> <p>Estos ficheros NO se generan automáticamente sino que el usuario debe dejarlos en la carpeta del contenido / aplicación o el portal (ver más adelante) y su contenido es arbitrario (el que el usuario quiera)</p> <p>Ej:</p> <pre>&lt;meta name="WT.ti" content="Home" /&gt; &lt;link rel="stylesheet" type="text/css" href="styles/myContentStyles.css" /&gt; &lt;script type="text/javascript" src="scripts/myContentScripts.js" &gt;&lt;/script&gt;</pre>

## IMPORTANTE!

Se han definido las reglas del servidor Web de modo que si el fichero de variables de **contenido** y de **página de portal** tiene una variable común, prevalecerá el valor de la variable de **contenido**.

### 3.3.8.1. Inserción de metadatos del contenido/página de portal en el HEAD de la página de portal

Tal y como se ha explicado antes, el Gestor de Contenidos y Portales automáticamente generan ficheros con los metadatos del modelo de contenidos y portales.

Hay tres casos de uso de estos ficheros:

#### Inserir METAS acerca de algún metadato del modelo contenido / portal

(aquellos metadatos que forman parte de lo que el usuario introduce con la aplicación de Gestión de contenidos / portales)

El Gestor de Portales y Contenidos **automáticamente** generan los ficheros:

Contenido: **r01ContMetas.inc**

Página de Portal: **r01[PageName]PortalMetas.inc**

con METAS del tipo: `<meta name="DC.creator" content="EJ03680M" >` que son insertados en el HEAD de la página

El usuario **NO** tiene que hacer nada para que se generen e incluyan estos ficheros

#### Inserir METAS acerca de metadatos que **NO** son parte del modelo de contenido / portal

(aquellos metadatos que **NO** forman parte de lo que el usuario introduce con la aplicación de Gestión de contenidos / portales)

El usuario puede insertar estos metas en los ficheros:

Contenido: **r01ContHead.inc**

Página de Portal: **r01[PageName]PortalHead.shtml**

junto con llamadas a scripts y hojas de estilo que son insertados en el HEAD de la página:

```
<meta name="WT.ti" content="Home"/>
<link rel='stylesheet' type='text/css' href='styles/myContentStyles.css' />
<script type='text/javascript' src='scripts/myContentScripts.js' ></script>
```

El usuario manualmente tiene que crear y subir estos ficheros al contenido / portal

#### Utilizar el valor de algún metadato en el HTML

**Ejemplo:** *Incluir en el TITLE de la página el nombre del contenido que se está visualizando.*

Una página tiene un **único tag <title>** y dado que una misma página de portal puede mostrar cualquier contenido **¿cómo hacer que en el title aparezca también el nombre del contenido?**

En el fichero de variables del contenido **r01ContVars.shtml** se incluye una variable que contiene el **nombre del documento**:

```
<!--#set var="doc_name" value="mi contenido de prueba"-->
```

ésta se añadirá al title de la página haciendo referencia a la variable en la en el TITLE:

```
<title>
```

```
Euskadi.net -
```

```
<!--#if expr="$documentName" --><!--#echo var="documentName" --><!--#endif -->
```

```
</title>
```

De esta forma, en el título del navegador, cuando se visualiza un contenido de nombre *"mi contenido de prueba"*, aparecerá:

Euskadi.net – **mi contenido de prueba**

### 3.3.8.2. Insercion de recursos de un contenido en el HEAD de la página de portal

Para insertar recursos (hojas de estilo / scripts) de un contenido en el HEAD de una página de portal, se utiliza el fichero: **{NombreContenido}/{NombreDocumento}/r01Index/r01ContHead.inc** que:

- Es “subido” por el usuario a la localización correcta dentro del contenido / documento
- Su contenido es arbitrario (fijado por el usuario) con “cualquier cosa” que se quiera inyectar en el HEAD de la página, por ejemplo:

```
<meta name="WT.ti" content="Home"/>
<link rel='stylesheet' type='text/css' href='styles/myContentStyles.css' />
<script type='text/javascript' src='scripts/myContentScripts.js' ></script>
```

#### **IMPORTANTE!!!**

La referencia a hojas de estilo y scripts normalmente debería ser **relativa al documento** de forma que si la URL de la página con el contenido es:

<http://sitio/portal-pag/lengPortal/repositorio/contenido/doc/archivo.html>

y se incluye una hoja de estilos de forma relativa: **styles/myContentStyles.css**, la URL final será correcta e independiente del repositorio de publicación:

<http://sitio/portal-pag/lengPortal/repositorio/contenido/doc/styles/myContentStyles.css>

### 3.3.8.3. Inserción de recursos de una aplicación en el HEAD de la página de portal

Se presenta un problema si es necesario “inyectar” un script, hoja de estilo o meta perteneciente una aplicación en el HEAD de la página:

La página de portal es la que contiene el esqueleto del HTML, incluyendo la sección HEAD y el servidor web “simplemente” inyecta el contenido / aplicación en el lugar indicado, pero **¿cómo inyectar la referencia a un script, una hoja de estilos o unos METAS propios de la aplicación en la sección HEAD de la página?**

**Técnicamente**, dado que las páginas de portal **existen de forma separada** a los contenidos/aplicaciones, se presenta un problema cuando es necesario incluir recursos del contenido en el HEAD de la página del portal:



Área visual de contenido. Puede “alojar” cualquier contenido / aplicación

La aplicación concreta es “inyectada” en tiempo de visualización de la página

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

<head>
  <title>Euskadi.net - Administraci3n de la Comunidad Aut3noma del Pa3s Vasco</title>
  <!-- **** MetaTags **** -->
  <meta name="Description" lang="es" content="Red de portales Euskadi.net" />
  <meta name="Keywords" lang="es" content="Euskadi, Euskadi.net, Euskadi" />
  <meta name="Author" lang="es" content="Euskadi.net" />
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Script-Type" content="text/javascript" />
  <meta http-equiv="Content-Style-Type" content="text/css" />

  <link rel="stylesheet" type="text/css" href="/styles/r33-2288VAsScripts.css" />
  <script type="text/javascript" src="/scripts/r33-2288VAsScripts.js" />
  <link rel="stylesheet" type="text/css" href="/styles/r33-Content.css" />
  <link rel="stylesheet" type="text/css" href="/styles/r33-Generales.css" />

</head>

<body class="r01PrintablePage">

<!-- ***** LOGOTIPOS ***** -->
<div class="r33_headerLogos r01Printable r01clearfix">
  <div class="r33_headerLogoLeft"> <!-- ***** r01gImageVA ***** --> </div>
</div>

<!-- ***** PRIMERA LINEA DE LA CABECERA ***** -->
<div class="r33_headerRow1 r01clearfix">
  <div class="r33_homePageButton"> <!-- ***** r01gImageVA ***** --> </div>
  <div class="r33_headerRow1Menu"> <!-- ***** r01gHorizontalMenuVA ***** --> </div>
</div>

<!-- ***** SEGUNDA LINEA DE LA CABECERA ***** -->
<div class="r33_headerRow2 r01clearfix">
  <div class="r33_headerRow2LangSel"> <!-- ***** r01gLangSelectorVA ***** --> </div>
  <div class="r33_headerRow2Menu"> <!-- ***** r01gHorizontalMenuVA ***** --> </div>
</div>

<!-- ***** TERCERA LINEA DE LA CABECERA ***** -->
<div class="r33_headerRow3 r01clearfix"> <!-- ***** r01gNavigatorVA ***** --> </div>
<!-- ***** BLOQUE CENTRAL: MENU IZQ Y CUERPO ***** -->
<div class="r33_body r01PrintableFullWidth r01clearfix">
  <!-- Columna izquierda -->
  <div class="r33_bodyLeftCol">
    <div class="r33_fullTextSearch"> <!-- ***** r01gSearchEngineVA ***** --> </div>
    <div class="r33_bodyLeftCol1Menu"> <!-- ***** r01gVerticalMenuVA ***** --> </div>
    <div class="r33_bodyLeftColBanner1"> <!-- ***** r01gBannerVA ***** --> </div>
    <div class="r33_bodyLeftColMenu2"> <!-- ***** r01gVerticalMenuVA ***** --> </div>
    <div class="r33_bodyLeftColBanner2"> <!-- ***** r01gBannerVA ***** --> </div>
  </div>

  <!-- Columna central -->
  <div class="r33_bodyCenterCol">
    <div class="r33_bodyCenterColContainer"> <!-- ***** r01gContainerVA ***** --> </div>
  </div>
</div>

<!-- ***** PIE ***** -->
<div class="r33_footer r01Printable"> <!-- ***** r01gPageFootVA ***** --> </div>

</body>
</html>

```

## ¿Cómo incluir en el HEAD metas, scripts y hojas de estilo procedentes de la aplicación?

El objetivo es **mover los scripts y hojas de estilo de aplicaciones desde el DIV de la aplicación hasta la sección HEAD de la página** ya que hasta ahora cuando se incrustaba una aplicación en una página de portal se utilizaba el AV de contenedor de contenidos:

```
<html>
<head>
    ....
</head>
<body>
    <div>AVCabecera</div>
    <div>AVMenu</div>
    <div>AVContainer de contenidos</div>
    <div>AVPie</div>
</body>
```

Para so  
páginas  
scripts d

Inyectar referencias a hojas de estilo y scripts **fuera del HEAD no es del todo correcto** (aunque los navegadores “tragan”).

Estas referencias **deben estar en el HEAD**

La aplicación tendrá una estructura como:

`<div>`  
**HTML aplicación**  
`</div>`

que se insertará en el área visual contenedora de la página.

Si la aplicación tiene que utilizar hojas de estilo o scripts propios, **ahora** los inserta dentro del DIV:

```
<div>
```

```
<link rel='stylesheet' type='text/css'
      href='/appCont/myApp/myStyles.css' />
<script type='text/javascript'
      src='/appCont/myApp/myScripts.js'></script>
```

**inyecta” automáticamente en la zona head de las**  
**incluyen en caso de existir,** un fichero de metas,

Las aplicaciones pueden utilizar un **único fichero** donde los responsables de las aplicaciones pueden poner “cualquier cosa” que quieran que se “inyecte” en el HEAD de la página contenedora.

Para poder utilizar esta funcionalidad el responsable de la aplicación debe **solicitar a implantación** que se **añada una entrada correspondiente a su aplicación en un mapa del apache** que indica dónde está el fichero de recursos para la aplicación:

**[nombreCarpetaContenidos] /appCont/[nombreCarpetaContenidos]/[nombreFichero.inc]**

El mapa de recursos se encuentra situado en la siguiente ruta (desarrollo):

**/opt/apache2/\${EJIE\_HTTPD\_INSTANCE\_NAME}/common/r01/maps/map.resourcehead.apps.internet.inc**

#### IMPORTANTE!

1. **NO ES NECESARIO REINICIAR EL SERVIDOR WEB** (apache)
2. La ruta indicada en el mapa normalmente referenciará a un **fichero .inc** que el responsable de la aplicación situará en la **carpeta de contenidos de su aplicación** (*una carpeta bajo /aplic/s08/html/ -- **appCont** -- donde coloca las imágenes, scripts y estilos de la aplicación*), y sobre la cual tiene control total.

**Ej.** Si en el mapa de apache se añade la siguiente fila para la aplicación **meteorologia**:

**meteorologia /appCont/meteorologia/test.inc**

Y en la **carpeta de contenidos** de la aplicación meteorología (/aplic/s08/html/meteorologia == /appCont/meteorologia), se sitúa un fichero llamado **test.inc**, cuyo contenido puede ser cualquier cosa (scripts, metas, estilos, etc) que se desee incluir en el HEAD de la página contenedora, por ejemplo:

```
<meta name="WT.ti" content="Home" />
<link rel='stylesheet' type='text/css' href='/appcont/[carpeta_contenidos]/styles/myContentStyles.css' />
<script type='text/javascript' src='/appcont/[carpeta_contenidos]/scripts/myContentScripts.js' ></script>
```

este texto es automáticamente “inyectado” en la sección HEAD de la página de portal contenedora de la aplicación.

**NOTA:** Los recursos estáticos de la aplicación están en una carpeta de contenidos en el repositorio de carpetas de contenido de aplicaciones (/appCont), por lo que **las referencias a los recursos han de ser /appcont/[carpeta\_contenidos]**

## NOTA:

Tal y como se ha explicado la inserción de METAS, SCRIPTS y CSS de una aplicación en el HEAD de la página contenedora es muy sencilla puesto que basta con crear un fichero que contenga el texto a incluir, por ejemplo:

```
<meta name="WT.ti" content="Home"/>
<link rel='stylesheet' type='text/css' href='/appcont/[carpeta_contenidos]/styles/myContentStyles.css' />
<script type='text/javascript' src='/appcont/[carpeta_contenidos]/scripts/myContentScripts.js' ></script>
```

el problema es que este fichero se incluye en **todas las pantallas de la aplicación** lo cual puede bastar en la mayoría de los casos, pero puede no ser suficiente en otros ya que es posible que sea necesario:

### Incluir METAS, SCRIPTS o CSS en función de la pantalla concreta de la aplicación

Resolver este problema es un poco más complicado ya que requiere cierta “sincronización” entre la aplicación y el fichero con las líneas a incluir en el HEAD de la página contenedora:

1. El fichero con las líneas a incluir en el HEAD de la página debe tener la extensión **.shtml** en lugar de **.inc** de forma que sea procesado por el filtro SSI del servidor web apache
2. En el fichero anterior hay que incluir directivas SSI condicionales en función de la página de la aplicación que se esté mostrando.

### Ejemplo:

Una aplicación Java en el contexto **/xxxMyAppWar/** tiene varias páginas y las líneas a introducir en el HEAD son distintas en cada una de ellas:

/xxxMyAppWar/xxxMyPage1.jsp	<pre>&lt;meta name="META 1" content="META 1 JSP"/&gt; &lt;link rel='stylesheet' type='text/css'   href='/appcont/[carpeta_contenidos]/styles/myStylesJSP.css' /&gt; &lt;script type='text/javascript'   src='/appcont/[carpeta_contenidos]/scripts/myScriptsJSP.js'&gt;&lt;/script&gt;</pre>
/xxxMyAppWar/xxxMyPage2.jsp	<pre>&lt;meta name="META 2" content="META 2 JSP"/&gt; &lt;link rel='stylesheet' type='text/css'   href='/appcont/[carpeta_contenidos]/styles/myStylesJSP.css' /&gt; &lt;script type='text/javascript'   src='/appcont/[carpeta_contenidos]/scripts/myScriptsJSP.js'&gt;&lt;/script&gt;</pre> <p>(ver que los scripts y estilos son los mismos en las dos JSP)</p>
/xxxMyAppWar/xxxMyServlet.jsp	<pre>&lt;link rel='stylesheet' type='text/css'   href='/appcont/[carpeta_contenidos]/styles/myStylesServlet.css' /&gt;</pre>
/xxxMyAppWar/xxxMyServlet.jsp?param1=val	<pre>&lt;meta name="META 3" content="META 3 Servlet"/&gt;</pre>

El fichero que contiene las líneas que hay que incluir en el HEAD (con extensión **.shtml**) puede utilizar expresiones condicionales del filtro SSI de apache utilizando variables apache como:

```
<!-- METAS dependientes de la JSP concreta -->
<!--#if expr="$DOCUMENT_NAME = xxxMyPage1.jsp" -->
  <meta name="META 1" content="META 1 Content"/>
<!--#endif -->
<!--#if expr="$DOCUMENT_NAME = xxxMyPage2.jsp" -->
  <meta name="META 2" content="META 2 Content"/>
<!--#endif -->

<!-- Todas las JSP comparten estilos y scripts -->
<!--#if expr="$DOCUMENT_NAME = /^xxx.[a-zA-Z0-9].jsp/" -->
  <link rel='stylesheet' type='text/css'
    href='/appcont/[carpeta_contenidos]/styles/myStylesJSP.css' />
  <script type='text/javascript'
    src='/appcont/[carpeta_contenidos]/scripts/myScriptsJSP.js'></script>
<!--#endif -->

<!-- Todos los servlet comparten estilos -->
<!--#if expr="$DOCUMENT_NAME = /^xxx[a-zA-Z0-9]+Servlet/" -->
  <link rel='stylesheet' type='text/css'
    href='/appcont/[carpeta_contenidos]/styles/myStylesServlet.css' />
<!--#endif -->

<!-- Solo si es un servlet y en la Query string llega un valor, incluir un META -->
<!--#if expr="$DOCUMENT_NAME = /^xxx[a-zA-Z0-9]+Servlet/ && $QUERY_STRING = param1=val" -->
  <meta name="META 3" content="META 3 Servlet"/>
<!--#endif -->
```

Las variables que se pueden utilizar en las condiciones son:

REQUEST_URI	El fichero solicitado en la URL
REQUEST_FILENAME	El path local completo al fichero o script solicitado
DOCUMENT_URI	El path de la URL (decodificado con %)
DOCUMENT_NAME	El nombre del fichero solicitado en la URL (excluyendo directorios)
QUERY_STRING	La query string (si se envía)
QUERY_STRING_UNESCAPED	La query string (si se envía) codificada con %

#### NOTAS:

- Para DEPURAR o probar el valor de las variables se puede incluir la sentencia SSI echo:

**Ej: `<!--#echo var="REQUEST_URI" -->`**

- Para más información sobre la funcionalidad, consultar en [http://httpd.apache.org/docs/2.0/mod/mod\\_include.html](http://httpd.apache.org/docs/2.0/mod/mod_include.html)

## 4 ANEXO: Guía de Integración de Aplicaciones

Ver documento "r01g-GuiaRapida\_IntegracionAPP\_Portal"