



– Opendata Euskadi –
Portal de Datos Abiertos del Gobierno Vasco

Denominación:

Manual de Usuario del Buscador de euskadi.net

Autor

Departamento de Justicia y Administración Pública del Gobierno Vasco
Dirección de Innovación y Administración Electrónica

Contenido

Capítulo/sección	Página
1 Introducción	3
2 Modelo de Contenido	Errorea! Laster-marka definitu gabe.
2.1 Objetos del Modelo y sus MetaDatos	Errorea! Laster-marka definitu gabe.
2.1.1 Contenido	Errorea! Laster-marka definitu gabe.
2.2 Documento / Versión	Errorea! Laster-marka definitu gabe.
2.3 DataFile	Errorea! Laster-marka definitu gabe.
3 Persistencia del Modelo de Contenido	Errorea! Laster-marka definitu gabe.
3.1 Detalle de la trasposición física de un contenido	Errorea! Laster-marka definitu gabe.
3.1.1 Descriptores del Contenido	Errorea! Laster-marka definitu gabe.
3.1.2 Documento = Versión del contenido	Errorea! Laster-marka definitu gabe.
3.1.3 Datos comunes del Contenido	Errorea! Laster-marka definitu gabe.
4 Despliegue de Contenidos (publicación)	Errorea! Laster-marka definitu gabe.
5 Estructura de URLs	Errorea! Laster-marka definitu gabe.
5.1 URL directa de un contenido o una página de portal	Errorea! Laster-marka definitu gabe.
5.2 Combinar un contenido en una página de portal	Errorea! Laster-marka definitu gabe.



1 Introducción

El presente documento contiene una guía de desarrollo de aplicaciones contra el buscador horizontal de la Infraestructura de Soporte al Modelo de Presencia en Internet.

El buscador como infraestructura es utilizable en cualquier iniciativa de reutilización para lo que se expone:

- Un API programático en JAVA
- Un API basado en codificar los resultados de búsqueda en la URL

En el presente documento se describen cada una de las alternativas anteriores.

Audiencia: Analistas y Desarrolladores

Complejidad: Media / Alta

2 Coordinador de Búsqueda

El buscador r01k **no** es un motor de búsqueda como tal sino simplemente un **coordinador de búsquedas**, que hace de **intermediario** entre el cliente y el/los **orígenes de resultados de búsqueda** (buscadores, aplicaciones, etc) que son los encargados de “atacar” físicamente a los repositorios de datos.

El coordinador de búsquedas tiene las siguientes funciones:

- Interfaz único entre aplicaciones cliente y orígenes de resultados
- Coordinación de las búsquedas en diferentes orígenes de resultados: MetaBuscador
- Manejo de las guías de navegación
- Presentación de Resultados

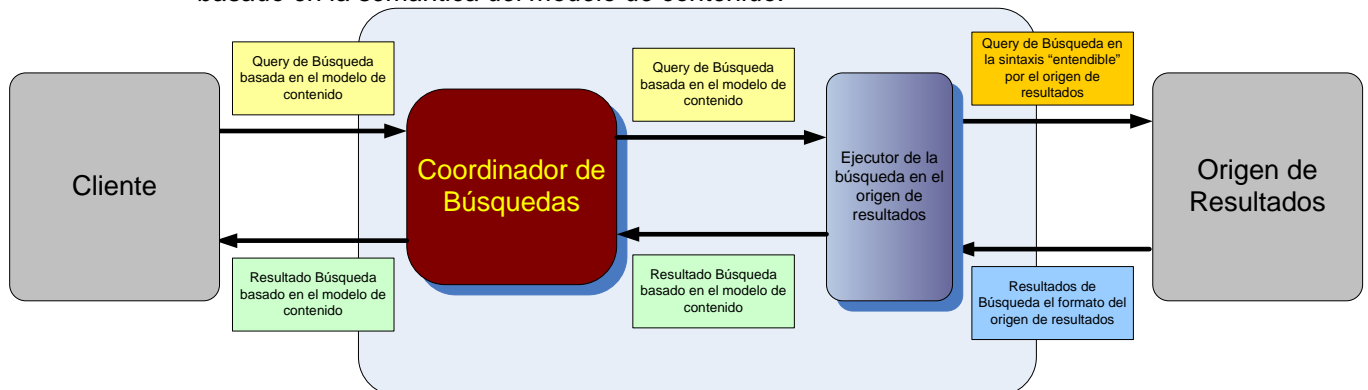
2.1 Interfaz único entre aplicaciones cliente y orígenes de resultados

El coordinador de búsquedas expone un **interfaz único** a las aplicaciones cliente basado en la **semántica del modelo de contenido**, de forma que es el encargado de “traducir” entre la semántica del modelo de contenido y la semántica que entiende cada motor de búsqueda:

Las aplicaciones cliente “interrogan” al coordinador de búsqueda con “términos” que tienen que ver con el modelo de contenido y es el coordinador de búsqueda el que interroga al origen de resultados de búsqueda (buscador) específico con términos que este entiende.

Así mismo, se encarga de devolver los resultados expresados en la **semántica del modelo de contenido**, traduciendo para ello los datos devueltos por cada origen de resultados.

Los orígenes de resultados pueden devuelven los mismos de diferentes formas: filas de una base de datos, xml, etc. El coordinador de búsquedas transforma este formato en uno único basado en la semántica del modelo de contenido.



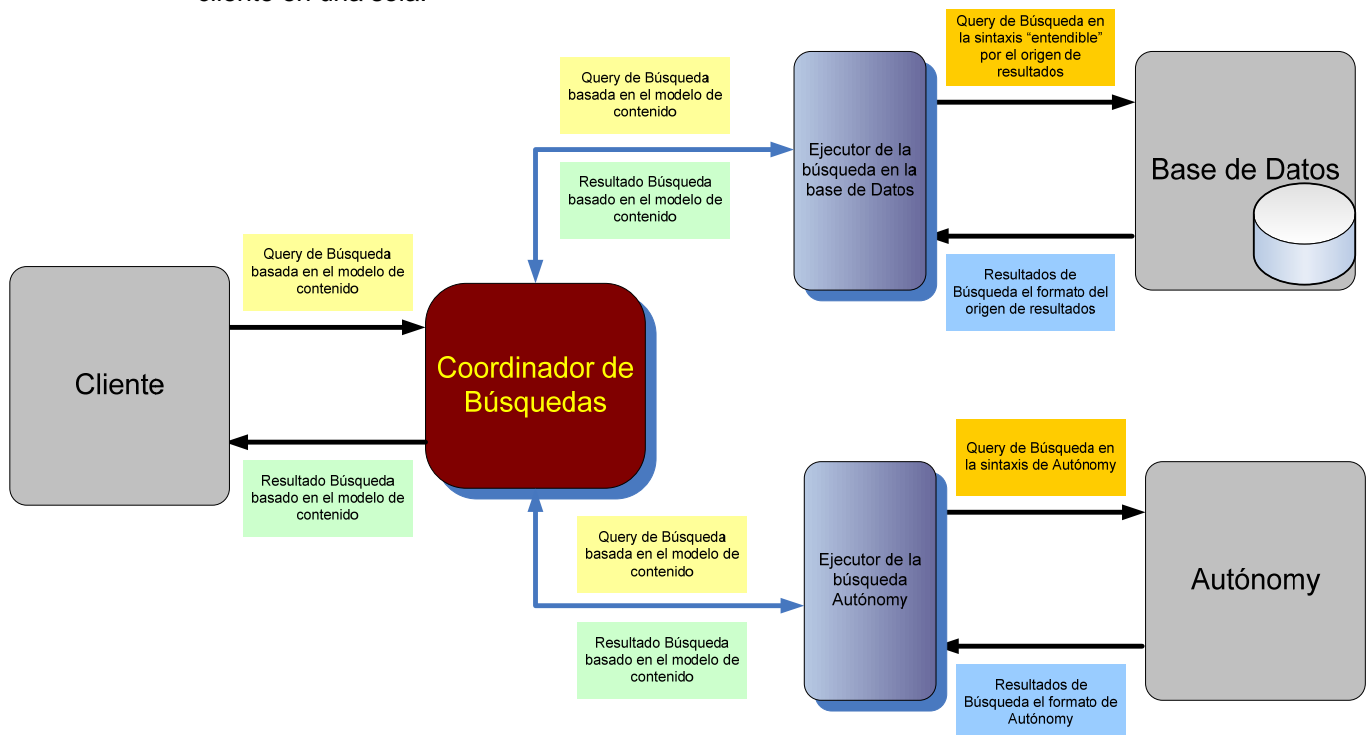
Como se aprecia en la figura anterior, los clientes del coordinador de búsqueda únicamente conocen la sintaxis de query y respuesta expuesta por este y que les “independiza”/desacopla de los orígenes de resultados y las sintaxis de consulta / respuesta propietaria de los mismos.

2.2 Coordinación de las búsquedas en diferentes orígenes de resultados: MetaBuscador

Los buscadores tradicionalmente funcionan en base a una gran “base de datos” central que referencia contenidos dispersos en muchos repositorios (orígenes). Esta base de datos centralizada es alimentada por indexers / crawlers y explotada por el motor de búsqueda que se encarga de resolver las consultas.

En ocasiones es difícil indexar los contenidos de todos los posibles repositorios (orígenes) en una base de datos centralizada de búsqueda. En estas situaciones se adopta una estrategia de **meta-buscador**:

El coordinador de búsqueda “lanza” la consulta a todos los orígenes capaces de devolver resultados sobre la misma y se encarga de agrupar las múltiples respuestas y devolverlas al cliente en una sola.



En el ejemplo anterior hay dos orígenes de resultados: una base de datos y autónoy:

- El coordinador de búsquedas es el encargado de coordinar las llamadas en paralelo a cada uno de los orígenes de resultados.
- Para ejecutar la búsqueda en cada origen, es necesario implementar un ejecutor específico que “conoce” la forma de interrogar al origen de resultados y es el encargado de:
 1. Transformar la query en la sintaxis del modelo en la sintaxis de consulta del origen.
 2. Ejecutar la query en el origen de resultados.
 3. Transformar de vuelta los resultados de vuelta a la sintaxis del modelo de contenido.

IMPORTANTE:

Obviamente si todos los contenidos están indexados en una única base de datos centralizada (por ejemplo autónoy), **no es necesario utilizar las capacidades de meta-búsqueda** del coordinador.

2.3 Manejo de las guías de navegación

Las guías de navegación son un mecanismo para “simular” navegaciones en índices por los resultados de búsqueda al estilo de buscadores tipo índice de Internet (Yahoo).

De esta forma, a cualquier búsqueda se puede asociar una guía de navegación de los siguientes tipos:

- Guía de navegación por un **eje de catalogación en etiquetas**.
- Guía de navegación por un **eje de catalogación geográfica**.
- Guía de navegación por la **jerarquía de tipos** (cluster → familia → tipos).
- Guía de navegación **mixta** (compuesta por el usuario) que contiene “nodos” de cualquiera de los tipos anteriores.

El coordinador de búsquedas básicamente lo único que se hace es “incluir” una nueva condición entre las ya especificadas en la query.

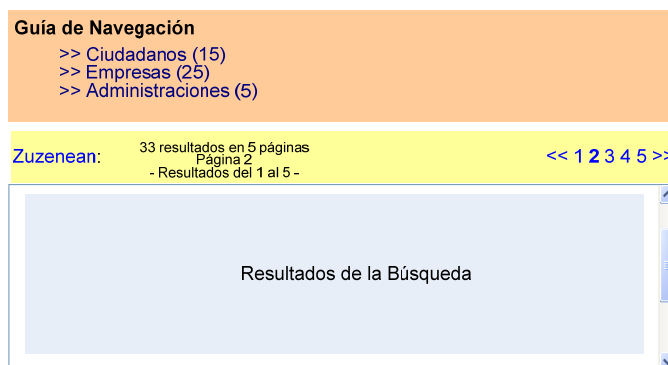
Ejemplo:

Destinatarios

- Ciudadanos
 - Niños / as
 - Jóvenes
 - Adultos
 - Tercera edad
- Empresas
 - Micro-empresas
 - Pymes
 - Grandes Empresas
- Administraciones
 - Local
 - Autonómica
 - Estatad

Suponiendo que existe un eje “**Destinatarios**” con la estructura de la derecha, a cualquier query (con cualquier tipo de condiciones sobre metaDatos), se le puede asociar una guía basada en este eje:

Al ejecutar la búsqueda aparecerá el primer nivel del eje junto con los resultados de la query.



Guía de Navegación

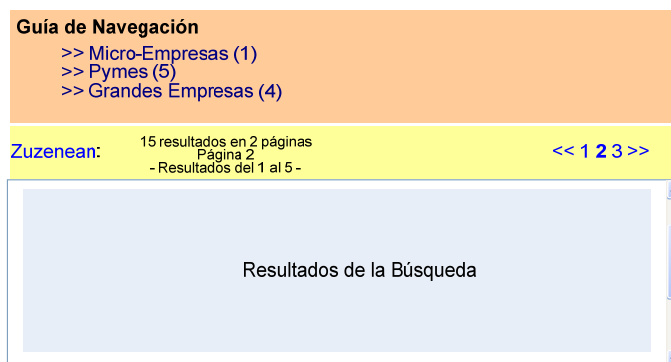
- >> Ciudadanos (15)
- >> Empresas (25)
- >> Administraciones (5)

Zuzenean: 33 resultados en 5 páginas
Página 2
- Resultados del 1 al 5 -

<< 1 2 3 4 5 >>

Resultados de la Búsqueda

Pinchando en uno de los nodos de la guía (por ejemplo empresas), se añadirá la etiqueta “empresas” (su identificador) a la query de búsqueda, que se ejecutará junto con el siguiente nivel de la guía:



Guía de Navegación

- >> Micro-Empresas (1)
- >> Pymes (5)
- >> Grandes Empresas (4)

Zuzenean: 15 resultados en 2 páginas
Página 2
- Resultados del 1 al 5 -

<< 1 2 3 >>

Resultados de la Búsqueda

De esta forma, pinchando en los sucesivos nodos, se puede ir “escarbando” en los resultados con la ayuda de la guía.

NOTA: Si los resultados de la guía inicial aparecen en una única agrupación NO SE PINTA LA GUÍA.

Las guías de navegación se definen en la **administración del sistema** y permiten combinar la navegación por ejes de navegación y tipos de contenido:

Se pueden enlazar distintos tipos de guías (ejes/tipos) mediante la definición de múltiples elementos `searchGuideDefBranch` contenidos unos dentro de otros, de tal forma que cuando finalice una de ellas me continúe con otra.

Ejemplos de creación de distintas guías:

Guía de navegación por tipología (cluster euskadi, familia informacion, y solo se permite navegar un nivel, es decir, hacer un solo clic en la guía):

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<searchGuideDef oid='guia_1'>
  <name>
    <es>Guia</es>
    <eu>eu_Guia</eu>
  </name>
  <description>
    <es>Guia</es>
    <eu>eu_Guia</eu>
  </description>
  <searchGuideDefBranch name='b1'
    branchType='typo'
    startingPath='euskadi/informacion'
    maxLevels='1'>
  </searchGuideDefBranch>
</searchGuideDef>
```

Guía de navegación por tipología (cualquier cluster, solo se permite un nivel, es decir, hacer clic en un cluster), navegación en ejes (estructura r01e00000fe4e6676dda470b8e96ebfb4e7b967c9 (Público destinatario del contenido), etiqueta r01e00000ff26d461a5a470b8618f7e296c596449 (Administraciones)) y finalmente navegación por tipología:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<searchGuideDef oid='guia_1'>
  <name><es>Guia</es><eu>eu_Guia</eu></name>
  <description><es>Guia</es><eu>eu_Guia</eu></description>
  <searchGuideDefBranch name='b1' branchType='typo' startingPath="" maxLevels='1'>
    <searchGuideDefBranch name='b2'
      branchType='structure'
      startingPath='r01e00000fe4e6676dda470b8e96ebfb4e7b967c9/r01e00000ff26d461a5a470b8618f7e296c596449'
      maxLevels='1'>
    </searchGuideDefBranch name='b3' branchType='typo' startingPath=""
    maxLevels='1'>
  </searchGuideDefBranch>
</searchGuideDef>
```

Guía de navegación por ejes (estructura r01e00000fe4e6676dda470b8e96ebfb4e7b967c9 (Público destinatario del contenido), etiqueta r01e00000ff26d461a5a470b8618f7e296c596449 (Administraciones), en cinco niveles de profundidad), después por tipología (comienza en el cluster euskadi, y permite navegar en 5 niveles, en este caso bastaría con 2 ya que solo sería aplicable para la familia y el tipo):

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<searchGuideDef oid='guia_1'>
  <name><es>Test Guide</es><eu>eu_Test Guide</eu></name>
  <description><es>Test Guide</es><eu>eu_Test Guide</eu></description>
  <searchGuideDefBranch name='branch_destinatarios'
    branchType='structure'
    startingPath='r01e00000fe4e6676dda470b8e96ebfb4e7b967c9/r01e00000ff26d461a5a470b8618f7e296c596449'
    maxLevels='5'>
  <searchGuideDefBranch name='branchTipos'
    branchType='typo'
    startingPath='euskadi'
    maxLevels='5'>
  </searchGuideDefBranch>
</searchGuideDef>
```

Ejemplo gráfico real del último ejemplo:

1ª ejecución (1er nivel de la rama branchTemas – eje de Temas)



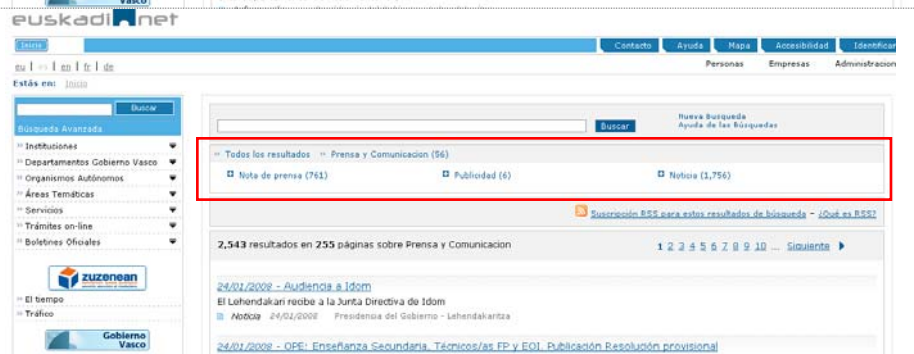
Pulsando en “Acción exterior” (2º nivel de la rama branchTemas – eje de Temas):



En este momento se ha completado el máximo número de niveles en el eje de Temas maxLevels="2", por lo que pulsando sobre “Asuntos Europeos” debe saltar a la rama branchTypos de Tipos de contenido:



En este momento se navegará a través de la rama de tipologías branchTypo, p.e. pulsando sobre “Prensa y Comunicación”:

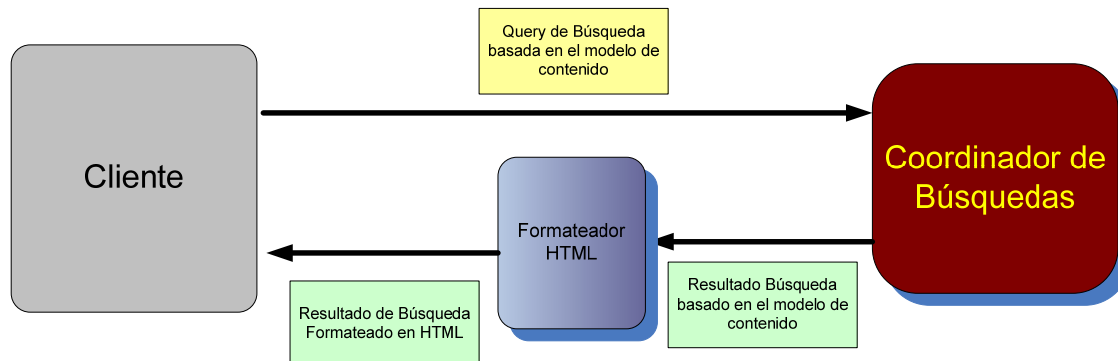


Finalmente aunque el número máximo de niveles maxLevels es 5, al pulsar sobre “Noticia” no existen más niveles de navegación (solo pueden ser 3) :



2.4 Presentación de Resultados

El coordinador de búsqueda tiene funciones auxiliares que facilitan la presentación de los resultados de forma sencilla en diferentes formatos (HTML, RSS, etc):



El formateador, es una pieza simple que conoce el resultado expresado en la sintaxis del modelo de contenido y es capaz de “transformarlo” a cualquier otro formato.

Existen formateadores estándar en HTML, XML y RSS.

La presentación de resultados estándar es similar a la que se muestra en la figura:

Resultados encontrados:

- [Zuzenean](#) (12)
- [Euskadi.net](#) (3)

Guía de Navegación

- >> [Empresas](#) (15)
 - >> [Micro-Empresas](#) (1)
 - >> [Pymes](#) (5)
 - >> [Grandes Empresas](#) (9)

Zuzenean: 12 resultados en 3 páginas
Página 2
- Resultados del 1 al 5 -

Resultados de la Búsqueda Repositorio Zuzenean

Euskadi.net: 3 resultados en 1 páginas
Página 1
- Resultados del 1 al 2 -

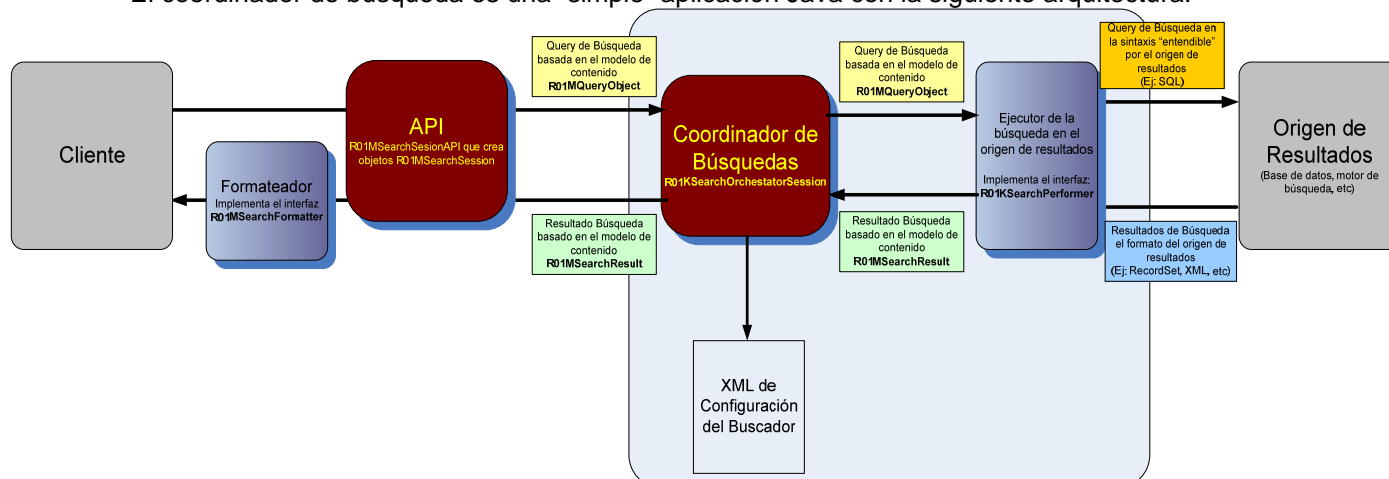
Resultados de la Búsqueda Repositorio Euskadi.net

Se distinguen las siguientes secciones:

- Repositorios (orígenes de resultados) a los cuales el coordinador de búsqueda ha expandido la búsqueda.
- Guía de navegación con los nodos pasados y los nodos futuros.
- Resultados de cada uno de los repositorios (orígenes de resultados) convenientemente paginados.

3 Arquitectura del Coordinador de Búsquedas

El coordinador de búsqueda es una “simple” aplicación Java con la siguiente arquitectura:



Los clientes únicamente conocen:

- **API:** clase **R01MSearchSessionAPI** que es una clase de factoría que genera objetos **R01MSearchSession** a través del cual se “maneja” la búsqueda (avanza/retrocede por las páginas o las guías de navegación)
- El objeto que encapsula la query de búsqueda, **R01MQueryObject**
- El objeto que encapsula los resultados de búsqueda **R01MSearchResult**
- La interfaz que han de implementar los formateadores de resultados de búsqueda: **R01MSearchFormatter**.

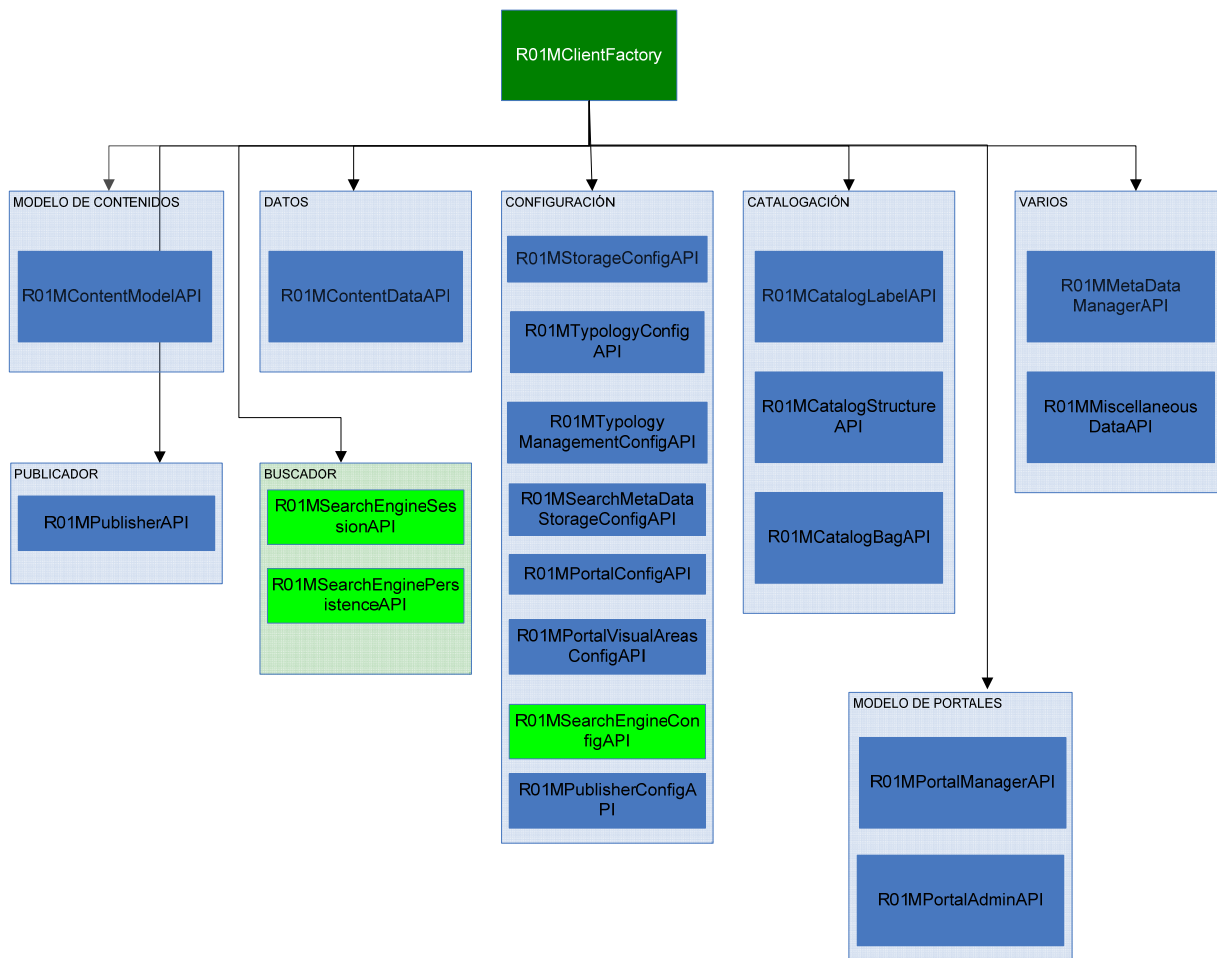
Internamente el buscador tiene una arquitectura relativamente sencilla:

- Todo el “motor” o núcleo del coordinador está encapsulado en la clase **R01KSearchOrchestratorSession** que “delega” la ejecución de las búsquedas en clases específicas para cada uno de los orígenes de resultados. Estas clases han de implementar el interfaz **R01KSearchPerformer**.

IMPORTANTE

Aunque el nombre **R01KSearchOrchestratorSession** puede indicar lo contrario, el coordinador de búsquedas **no tiene sesión**: no almacena ningún dato entre llamadas

4 API del Buscador



La clase **R01MSearchEngineSessionAPI** que a grandes rasgos, expone las siguientes funciones:

createSession	Devuelve un objeto R01MSearchSession que encapsula todas las funciones para interactuar con el coordinador de búsqueda: paginación, navegación por guías, etc
saveQuery	Permite guardar una query en base de datos para su ejecución en ocasiones futuras
loadQuery	Permite obtener una query que fué almacenada en la base de datos
index	Indica al coordinador de búsqueda que indexe un contenido en un origen de búsqueda utilizando los indexadotes registrados en la configuración del buscador (ver XML anterior)
unindex	Indica al coordinador de búsqueda de elimine la indexación de un contenido en un origen de búsqueda

El método más importante es **createSession** que devuelve un objeto **R01MSearchSession** que permite controlar la sesión de búsqueda:

gotoFirstPage	Lleva el puntero de paginación a la primera página
gotoPrevPage	Lleva el puntero de paginación a la página anterior a la actual
gotoPage	Lleva el puntero de paginación a la página que se especifica
gotoNextPage	Lleva el puntero de paginación a la siguiente página
gotoLastPage	Lleva el puntero de paginación de la última página
followGuide	Navega (o lleva) la búsqueda al nodo futuro de la guía de navegación indicado
rewindGuide	Navega (o lleva) la búsqueda al nodo pasado de la guía de navegación indicado
getSearchResults	Obtiene un objeto R01MSearchSessionResults que contiene los resultados de la búsqueda

Como se puede ver el objeto **R01MSearchSession** únicamente se utiliza para interactuar con el coordinador y paginar o seguir / rebobinar la guía de navegación

IMPORTANTE

El núcleo del buscador (**R01KSearchOrchestatorSession**) **no almacena sesión (estado)**, sin embargo, **el cliente** puede (y en ocasiones debe) mantener la sesión manteniendo el objeto **R01MSearchSession** mientras dura la interacción de búsqueda.

Este modelo es conveniente ya que **no** sobrecarga el núcleo, dejando la responsabilidad de mantener sesión en el cliente que no la utilizará cuando no sea necesario.

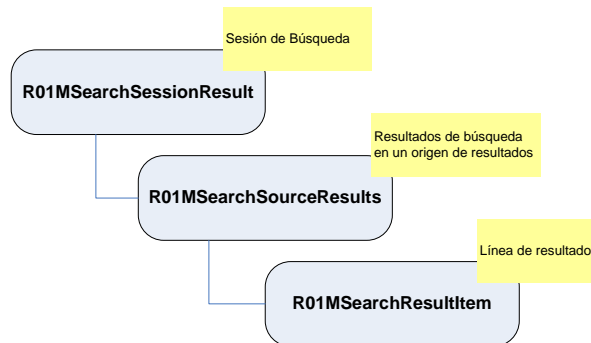
La utilización del API es extremadamente sencilla:

```
// Crear una sesión de buscador y obtener los resultados de la búsqueda
R01MSearchEngineSessionAPI srchAPI = new R01MSearchEngineSessionAPI();
R01MSearchSession srchSession = srchAPI.createSession(null, // Contexto (se toma el de por defecto)
    null, // SearchEngineId (se toma el de por defecto)
    null, // sessionId (se crea uno nuevo)
    lang, // lenguaje
    qry, // query de búsqueda
    null, // Nodo raíz de la guía de navegación
    false, // Lazy Init: Obtener los resultados de la
        // primera página durante la inicialización
    cacheResults); // Cachear los resultados de
        // búsqueda
```

En el ejemplo se observa cómo basta con instanciar el API e invocar al método **createSession** que devuelve un objeto **R01MSearchSession** a través del cual se puede paginar, seguir guías, etc.

4.2 Objeto Resultado

El objeto **R01MSearchSessionResults** encapsula los resultados de una sesión de búsqueda según el siguiente esquema:



sessionOid	Identificador único de la sesión de búsqueda
lang	Lenguaje en el que se está buscando
totalNumberOfResults	Número total de resultados (en todos los repositorios / orígenes de búsqueda)
guidePastElements	Elementos pasados en la guía de navegación (aquellos que ya se han pasado)
guideFutureElements	Elementos futuros en la guía de navegación (aquellos a los que se puede ir)
searchResultsBySource	Resultados de cada uno de los orígenes. Es un mapa de objetos R01MSearchSourceResults indexado por el identificador del origen de resultados

El objeto **R01MSearchSessionResults** contiene los resultados de cada uno de los orígenes en objetos **R01MSearchSourceResults**:

sourceOid	Identificador del origen de resultados
sourceName	Nombre del origen de resultados (en el idioma de la búsqueda)
numberOfResults	Número de resultados devueltos por este origen de resultados
numberOfPages	Número de páginas devueltas por este origen de resultados
navBar	Encapsula en un objeto R01MSearchNavBar la barra de navegación (paginación) por el origen
results	Un array de objetos R01MSearchResultItem que encapsulan cada uno de los resultados (filas)

El objeto **R01MSearchSourceResults** contiene cada una de las **filas** de resultado encapsulada en objetos **R01MSearchResultItem**:

itemNumber	Número de orden del elemento (fila) dentro del conjunto de resultados
score	Porcentaje de acierto del elemento (fila)
contentOid	Identificador único del contenido
contentName	Nombre del contenido
contentDescription	Descripción del contenido
contentLocation	Localización del contenido en el almacenamiento: servidor, dataStore, área y WorkArea
contentTypology	Tipificación del contenido: cluster, familia y tipo
contentEditURL	URL de edición del contenido
contentPreviewURL	URL de previsualización del contenido en el gestor de contenidos
contentViewURL	URL de visualización del contenido desplegado
structureCatalogs	Etiquetas de catalogación en ejes
geoCatalogs	Etiquetas de catalogación geográfica

documentOid	Identificador único del documento
documentInternalName	Nombre interno del documento
documentLanguage	Lenguaje del documento
documentName	Nombre del documento
documentDescription	Descripción del documento
documentStatus	Estado de aprobación del documento: en redacción, pendiente de X aprobaciones, aprobado o rechazado
documentMetaData	MetaDatos específicos al tipo
documentEditURL	URL de edición del documento
documentPreviewURL	URL de previsualización del documento
documentViewURL	URL de visualización del documento
dataFileOid	Identificador único del fichero de datos
dataFileName	Nombre del fichero de datos
dataFileResume	Resumen (descripción interna) del fichero de datos
dataFileEditURL	URL de edición del fichero de datos
dataFilePreviewURL	URL de previsualización del fichero de datos
dataFileViewURL	URL de visualización del fichero de datos

4.3 Búsquedas Codificadas en la URL

El buscador de euskadi.net admite **búsquedas codificadas en la URL**, el formato es el siguiente:

Resultado en XML	http://www.euskadi.net/r01hSearchResultWar/r01hPresentation XML .jsp? r01kLang=es eu en fr de& r01kQry=[query codificada]
Resultado en RDF	http://www.euskadi.net/r01hSearchResultWar/r01hPresentation RDF .jsp? r01kLang=es eu en fr de& r01kQry=[query codificada]
Resultado en RSS	http://www.euskadi.net/r01hSearchResultWar/r01hPresentation RSS .jsp? r01kLang=es eu en fr de& r01kQry=[query codificada]
Resultado en HTML	http://www.euskadi.net/r01hSearchResultWar/r01hPresentation HTML .jsp? r01kLang=es eu en fr de& r01kQry=[query codificada]

La query codifica puede ser:

- Una búsqueda en **texto libre**
- Una búsqueda **paramétrica**

4.3.1 Búscador de texto libre

Se pueden realizar búsquedas AND y OR:

- AND** Se concatenan las palabras a buscar con el carácter "+", p.e. si se introduce el texto "oposición+gobierno", buscará las ocurrencias con ambas palabras.
- OR** Se introducen las palabras separadas por un espacio, p.e. si se introduce el texto "oposición gobierno", buscará las ocurrencias con el texto oposición y las ocurrencias con el texto gobierno.

La URL de la query de texto libre puede tener los siguientes parámetros:

fullText	textToSearch&resultsSource=fullText	Texto a buscar en texto completo
-----------------	--	----------------------------------

4.3.2 Búsqueda codificada

Este tipo de búsqueda permite realizar todo tipo de consultas utilizando filtros por metadatos del modelo de contenido.

La URL de la query codificada puede tener los siguientes parámetros:

r01kQry	query por metadatos; pp=opciones de presentación; o=metadato_ordancion	Query Opciones de presentación MetaDatos de ordenación
r01kLang	<i>codelang</i>	Idioma de presentación de los resultados, NO CONFUNDIR CON EL IDIOMA DE BÚSQUEDA DE RESULTADOS, este idioma afecta a los literales de presentación de resultados. Valores permitidos: es,eu,en,fr,de.
r01kSearchResultsHeader	1/0	Incluir cabecera de los resultados de la búsqueda.
r01kRssr01kSearchBox	1/0	Incluir la opción RSS
r01kPresentationXSL	<i>xsl_de_presentación</i>	Plantilla de presentación de resultados que transforma el XML de resultado de búsqueda

r01kPageContents	page	Página donde se muestran los resultados de búsqueda
r01kNumResultsPerPage	page_size	Tamaño de página, es otra forma de indicarlo ya que también puede venir especificado en las propiedades de presentación de la query, en cuyo caso dicho valor tiene preferencia sobre este
r01kGuide	guideoid	Identificador de la guía de navegación con la que presentar los resultados de búsqueda
r01kDebug	true	Presenta información de depuración

4.3.2.1 Condiciones de la query

Las condiciones de la query por metadatos se separan por el carácter “;” siguiendo la nomenclatura:

Filtro	Posibles Valores				
Tipología	Cluster	tC:euskadi			
	Familia	tF:procedimientos_administrativos			
	Tipo	tT:ayuda_subvencion			
	<p>Se pueden especificar múltiples tipos en la query separados por comas, en este caso no tiene sentido especificar más de una familia debido a que se realiza una búsqueda que no devolverá resultado ya que se realizaría una composición AND por familias y un tipo nunca puede pertenecer a dos familias distintas.</p> <p>Especificar múltiples familias solamente tiene sentido cuando no se especifique el tipo (en este caso los distintos valores deben venir separados por coma).</p> <p>Ejemplos: filtro por multiples tipos: tC:euskadi;tF:turismo;tT:j_gastronomia filtro por múltiples familias: tC:euskadi;tF:turismo,eventos</p>				
Almacenamiento en el Gestor de Contenidos	IMPORTANTE: Este tipo de filtro NO debería imponerse en internet ya que el almacenamiento del contenido en el gestor de contenidos puede variar.				
	Servidor	sS:ejlp013			
	Store	sR:euskadi			
	Area	sA:horizontal_euskadi			
	WorkArea	sW:wghoz01			
Catalogacion	Los filtros por etiquetas de catalogación pueden ser:				
	AND	<table border="1"> <tr> <td>cA</td> <td>cA:labelOid1,labelOid...</td> <td>El contenido debe estar catalogado en TODAS las etiquetas especificadas</td> </tr> </table>	cA	cA:labelOid1,labelOid...	El contenido debe estar catalogado en TODAS las etiquetas especificadas
	cA	cA:labelOid1,labelOid...	El contenido debe estar catalogado en TODAS las etiquetas especificadas		
OR	<table border="1"> <tr> <td>cO</td> <td>cO:labelOid1,labelOid...</td> <td>El contenido puede o no estar catalogado en las etiquetas especificadas</td> </tr> </table>	cO	cO:labelOid1,labelOid...	El contenido puede o no estar catalogado en las etiquetas especificadas	
cO	cO:labelOid1,labelOid...	El contenido puede o no estar catalogado en las etiquetas especificadas			

Repositorio de publicación	<p>Se puede filtrar por repositorio de publicación con la condición:</p> <p style="text-align: center;">p:[repositorio]</p> <p>Es posible indicar más de un repositorio de publicación:</p> <p style="text-align: center;">Ej: p:internet,intranet</p>														
MetaDatos	<p style="text-align: center;">m:nombre_metadato.operacion.valor</p> <p>Si se impone más de una condición de filtro por metadatos se han de separar por el carácter “,”</p> <p>Ejemplo:</p> <p style="text-align: center;">m:documentPublicationInfo.EQ.1, documentLanguage.EQ.es, templateType.LIKE.J1, gastronomyType.EQ.0000;</p> <p>Se pueden combinar varios metadatos en operaciones AND / OR:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; vertical-align: middle;">AND</td> <td style="text-align: center; vertical-align: middle;">mA o m</td> <td style="vertical-align: middle;">mA: documentName.EQ.hola, documentDescription.EQ.adios</td> <td style="vertical-align: middle;">El contenido debe cumplir TODAS las condiciones sobre los metadatos</td> </tr> <tr> <td style="text-align: center; vertical-align: middle;">OR</td> <td style="text-align: center; vertical-align: middle;">cO</td> <td style="vertical-align: middle;">mO: documentName.EQ.hola, documentDescription.EQ.adios</td> <td style="vertical-align: middle;">El contenido debe cumplir al menos UNA de las condiciones sobre los metadatos</td> </tr> </table> <p>Las operaciones que se pueden imponer sobre los metadatos dependen del tipo de metadato y se configuran en la definición del metadato:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Tipo de metadato</th> <th style="text-align: left;">Posible operación de filtro</th> </tr> </thead> <tbody> <tr> <td>texto</td> <td>EQ, LIKE</td> </tr> <tr> <td>Numérico</td> <td>EQ,NEQ,LTE,GTE</td> </tr> </tbody> </table>	AND	mA o m	mA: documentName.EQ.hola, documentDescription.EQ.adios	El contenido debe cumplir TODAS las condiciones sobre los metadatos	OR	cO	mO: documentName.EQ.hola, documentDescription.EQ.adios	El contenido debe cumplir al menos UNA de las condiciones sobre los metadatos	Tipo de metadato	Posible operación de filtro	texto	EQ, LIKE	Numérico	EQ,NEQ,LTE,GTE
AND	mA o m	mA: documentName.EQ.hola, documentDescription.EQ.adios	El contenido debe cumplir TODAS las condiciones sobre los metadatos												
OR	cO	mO: documentName.EQ.hola, documentDescription.EQ.adios	El contenido debe cumplir al menos UNA de las condiciones sobre los metadatos												
Tipo de metadato	Posible operación de filtro														
texto	EQ, LIKE														
Numérico	EQ,NEQ,LTE,GTE														

4.3.2.2 Opciones de Presentación

Se pueden indicar múltiples propiedades de presentación separadas por comas:

pp:nombre_propiedad.valor, nombre_propiedad.valor

r01NavBarBlockSize	Número de bloques (páginas) que se muestran accesibles mediante enlaces. Aparecen como enlazables: 1 2 3 4 ... <u>Siguiente</u>
r01PageSize	Número de resultados a devolver por página.

Ejemplo:

pp:r01NavBarBlockSize.10,r01PageSize.10

4.3.2.3 *MetaDatos de Ordenación*

Se pueden ordenar los resultados de la búsqueda indicando un metadato por el cual sea posible la ordenación.

`o:nombre_metadato.ASC/DESC,nombre_metadato.ASC/DESC`

Ejemplo:

`o:historicTerritoryCode.ASC,municipalityCode.ASC`

NOTA: No se puede ordenar por todos los metadatos, únicamente por aquellos en los que se ha habilitado esta opción en la configuración del metadato

NOTA: Autonomy tiene un BUG que hace que no se devuelvan los resultados que no contengan valor para el metadato por el cual se está forzando la ordenación.