



## **Documentación técnica**

**Fideos v2.0.2**

# Índice de contenido

1. Introducción.....	3
2. Librerías utilizadas.....	3
3. Instalación.....	3
4. ¿Cómo se actualizan los “feeds”?.....	4
5. La generación del Layout.....	4
6. Un simple MVC.....	5
7. El Caldo y El Fideotipo.....	6
8. Modelo de datos.....	7
8.1. Diagrama Entidad-Relación.....	7
8.2. Estructura de datos.....	7
8.2.1 Tabla: fd_webusers.....	7
8.2.2 Tabla: fd_links.....	8
8.2.3 Tabla: fd_categories.....	8
8.2.4 Tabla: fd_folders.....	8
8.2.5 Tabla: fd_itemsfeed.....	8
8.2.6 Tabla: fd_userfeeds.....	9
8.2.7 Tabla: fd_feedvotes.....	9
8.2.8 Tabla: fd_comments.....	9

## 1. Introducción

La idea que siempre tuve en mente, era la de hacer un agregador de contenidos que mostrara al usuario la información filtrada de forma inteligente, captando y aprendiendo de los gustos del usuario. Todos somos conscientes del problema que representa el gran volumen de información que circula cada día por la red y lo fácil que resulta en ocasiones no enterarse de cosas que podrían haber sido de nuestro interés.

Los RSS, que de echo son la base de la denominada *web semántica*, suelen estar categorizados y gracias a eso son un buen sistema para montar un sistema inteligente que sea capaz de aprender de nuestros gustos y nos muestre la información que realmente nos interesa.

Fideos.net no llega a ese punto, como me parece que no llega ningún otro agregador RSS. A lo máximo que llega es a mantener un contador de *clicks* en los enlaces para montar “El Caldo” una clasificación de lo más visitado por todos los usuarios.

Este documento no pretende ser ni muy exhaustivo ni formal, simplemente se intenta explicar como funcionan más o menos las partes más importantes del proyecto y el porque de algunas decisiones de diseño.

## 2. Librerías utilizadas

En Fideos.net se utilizan las siguientes librerías:

- **ADODB 4.62** para acceder a la BDD: <http://adodb.sourceforge.net/>
- **patTemplate 3.0.0** como motor de plantillas HTML: <http://trac.php-tools.net/patTemplate>
- **MagpieRSS 0.72** como parser RSS: <http://magpierss.sourceforge.net/>
- **PHPMailer 1.73** para el envío de correos: <http://phpmailer.codeworxtech.com/>
- **Prototype + Scriptaculous** para implementar las partes “Ajaxeras” ;-)

## 3. Instalación

La instalación es muy sencilla, únicamente se trata de seguir los siguientes pasos:

- Crear la base de datos (herramientas como phpMyAdmin ayudan mucho)
- Ejecutar el script SQL “*docs/database/create\_db.sql*”
- Configurar el fichero “*includes/config.inc.php*”:
  - **SERVER\_PATH**: Indicar el directorio donde se encuentra “fideos”
  - Configurar el acceso a la base de datos poniendo los valores adecuados en las entradas **DB\_\*** (**DB\_SERVER**, **DB\_USERNAME**,...)
  - Configurar el envío de e-mails (entradas de tipo **SMTP\_\***)

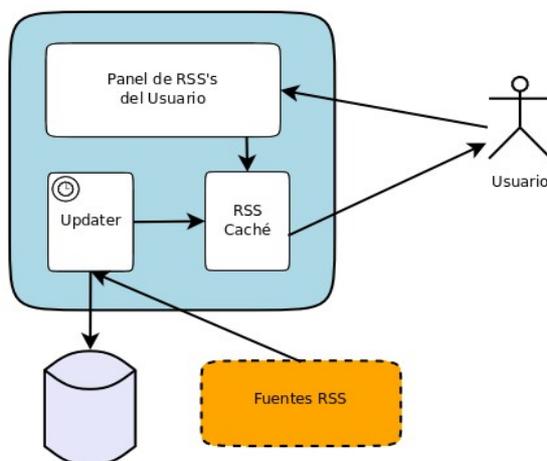
A tener en cuenta que si Fideos no se va a servir desde la raíz del dominio se deberá modificar el valor de APP\_PATH. Por ejemplo, si se va a servir desde “http://foo.bar/fideos” APP\_PATH deberá valer “/fideos”.

En el fichero de configuración están explicadas con más detalle cada una de las entradas.

#### 4. ¿Cómo se actualizan los “feeds”?

Fideos.net es una web donde el usuario puede escoger y organizar en carpetas las fuentes RSS de las que quiere estar informado.

El proceso de actualización de las fuentes es muy simple, cada hora se ejecuta un “cron” (Updater) que actualiza todas las fuentes RSS y las guarda en caché. De esta manera aceleramos el proceso de generación de la página con el panel de RSS del usuario.



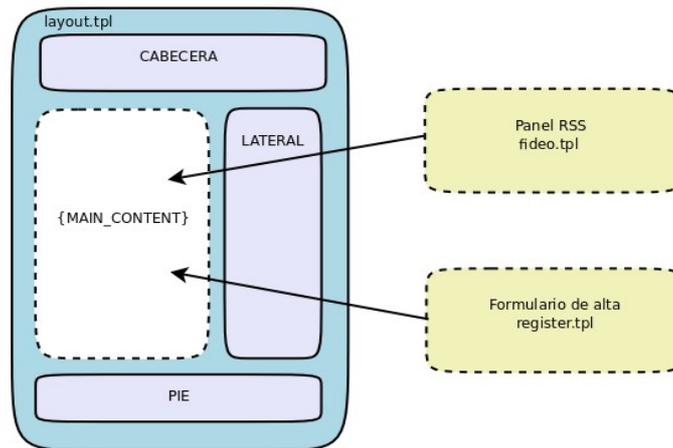
Dibujo 1: Actualización de fuentes RSS

Asimismo, el proceso de actualización también inserta en la base de datos las entradas de cada RSS para permitir llevar la estadística de las noticias mas visitadas por los usuarios y poder generar el RSS llamado “El Caldo”.

Si un usuario pide un RSS cuya caché ha expirado se actualiza la caché pero no se inserta nada en la BDD. La BDD solo la actualiza el “cron” de actualización. Si prescindieramos del “Caldo” no haria falta el “cron” y por lo tanto no seria necesario mantener la tabla de entradas en la BDD.

#### 5. La generación del Layout

Para la generación de las páginas HTML se utiliza el patrón de diseño “Decorator”, es decir, existe una plantilla HTML marco (o decorador) con un hueco donde se coloca el contenido de cada página. En otras palabras, en lugar de hacer los típicos “includes” lo que se hace es inyectar la página solicitada dentro de la plantilla marco. Para implementarlo se ha usado el motor de plantillas patTemplate.



*Dibujo 2: Patrón Decorator*

Esto solo aplica a los *requests* de páginas completas, no a las llamadas AJAX, donde solo se devuelve el HTML que se deba incrustar en la capa indicada.

La clase `FLayout.php` que extiende de `Layout` es la que se encarga de montar las páginas. Básicamente necesita que se le indique una plantilla “marco” (en nuestro caso “/templates/layout.tpl”) y el contenido a pintar en el área “main\_content”. En el siguiente ejemplo se verá más claro como funciona.

```
<?php
include ("includes/common.php");

$layout = new FLayout();

$tplMain = &new patTemplate();
$tplMain->setRoot( TEMPLATE_PATH );
$tplMain->readTemplatesFromInput( 'contact.tpl' );

$layout->setMainContent($tplMain->getParsedTemplate());
$layout->display();

?>
```

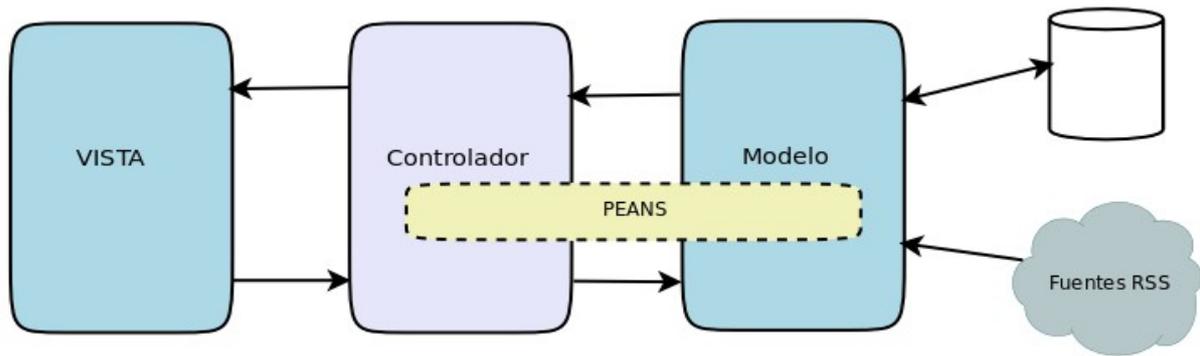
*Texto 1: Ejemplo página contacto (contact.php)*

Creamos un nuevo Flayout y leemos la plantilla *contact.tpl*, que colocaremos en el hueco `{MAIN_CONTENT}` del marco mediante el metodo *setMainContent*. Una vez finalizado, se muestra la página con el método *display*.

## 6. Un simple MVC

Para demostrar que se pueden aplicar patrones de diseño en PHP y trabajar al estilo Java, obviamente salvando las distancias, he tratado de aplicar ciertos conceptos que se utilizan mayoritariamente en los proyectos “javeros”, como los Beans, a los que he llamado Peans (PHP Beans) y un DAO por cada entidad.

De esta manera tenemos una capa que genera el HTML, otra que accede a los datos, un controlador que se comunica con ambas y unos objetos, los Peans, que viajan de un lado a otro.



*Dibujo 3: MVC Fideos*

Todos los DAOs se encuentran en “/lib/pdaos” y los objetos del modelo en “/lib/peans”. Un módulo típico, por ejemplo el de las carpetas de usuario, constaría de lo siguiente:

- Un Pean (/lib/peans/folder\_pean.php)
- Un DAO (/lib/pdaos/fd\_folders.php)
- Una o varias plantillas HTML (/templates/folders.tpl)
- Uno o varios controladores (/ajax/create\_folder.php)

## 7. El Caldo y El Fideotipo

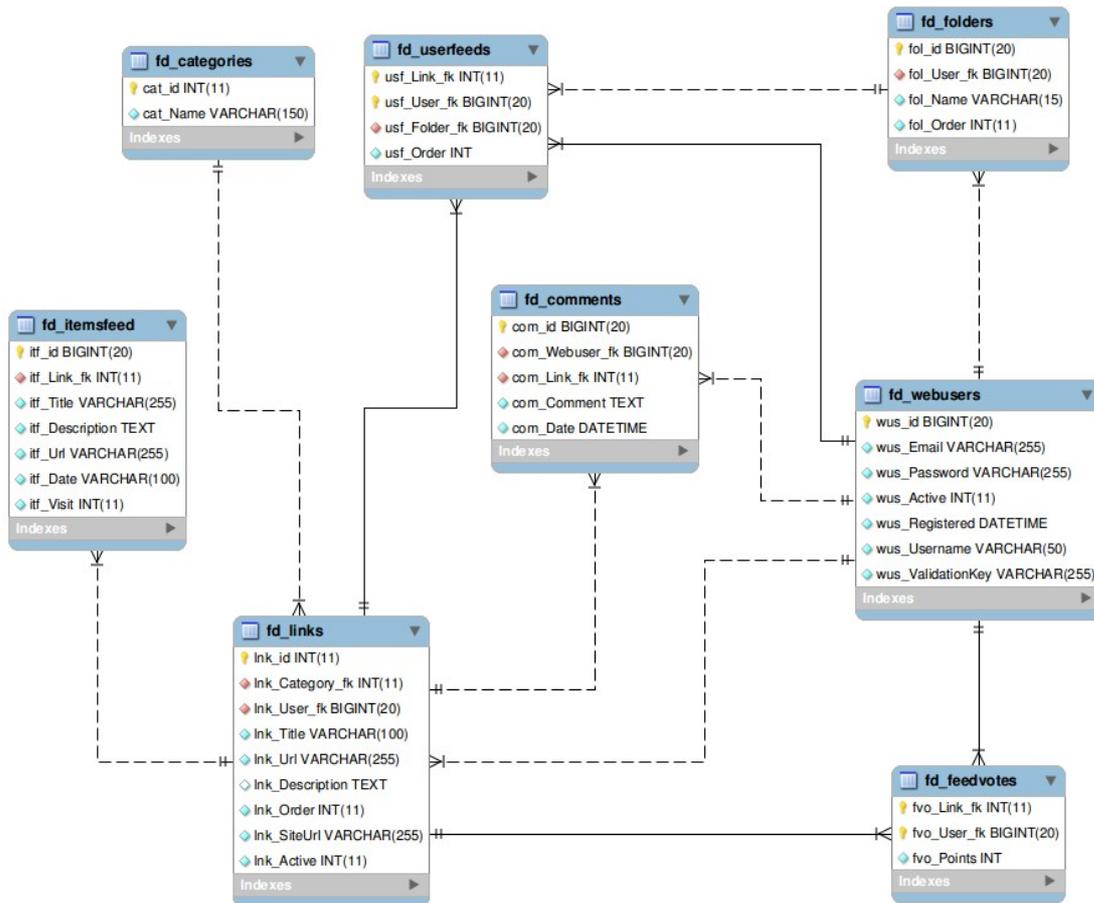
El Caldo y el Fideotipo son los dos RSS que genera Fideos.net. El Caldo muestra las noticias más vistas por los usuarios y El Fideotipo las últimas entradas de toda la nube de RSS.

Para saber cuales son las noticias más visitadas y generar El Caldo, lo que se hace es agregar al evento “onclick” de los enlaces una llamada AJAX que incrementa el contador de visitas. Una vez lanzado el “click” se ejecuta el HREF del enlace y se abandona la página.

Lo del Fideotipo es más sencillo, simplemente se busca en la BDD los últimos posts agregados de todas las fuentes RSS de Fideos.net.

## 8. Modelo de datos

### 8.1. Diagrama Entidad-Relación



### 8.2. Estructura de datos

#### 8.2.1 Tabla: fd\_webusers

Los usuarios que se registran se almacenan aquí.

Campo	Tipo	Descripción
<b>wus_id</b>	bigint	Clave primaria
wus_Email	varchar(255)	Email del usuario
wus_Password	varchar(255)	Password encriptado del usuario
wus_Username	varchar(50)	Nick del usuario
wus_Active	int	Indica si está activo (1), puede hacer login, o inactivo (0)
wus_Registered	DateTime	Fecha de registro
wus_ValidationKey	varchar(255)	Clave de validación para confirmar el alta

### 8.2.2 Tabla: fd\_links

En esta tabla se almacenan los *feeds* RSS

Campo	Tipo	Descripción
<b>Ink_id</b>	int	Clave primaria
Ink_Title	varchar(100)	Título del <i>feed</i>
Ink_URL	varchar(255)	URL del <i>feed</i> RSS
Ink_Description	text	Descripción del <i>feed</i>
Ink_Order	int	Orden en que se muestra (no se usa)
Ink_SiteUrl	varchar(255)	URL del site origen del RSS
Ink_Active	Int	Indica si el <i>feed</i> está activo (1), para mostrarlo en la web, o no (0).
Ink_Category_fk	int	Categoría del <i>feed</i> . Clave foránea a la tabla fd_categories
Ink_User_fk	bigint	Usuario que registró el <i>feed</i> . Clave foránea a la tabla fd_webusers

### 8.2.3 Tabla: fd\_categories

Campo	Tipo	Descripción
<b>cat_id</b>	int	Clave primaria
cat_Name	varchar(150)	Nombre de la categoría

### 8.2.4 Tabla: fd\_folders

Carpetas de los usuarios de Fideos.net

Campo	Tipo	Descripción
<b>fol_id</b>	bigint	Clave primaria
fol_Name	varchar(15)	Nombre de la carpeta
fol_Order	int(11)	Orden en que el usuario decide mostrar la carpeta
fol_User_fk	bigint	Usuario propietario de la carpeta. Clave foránea a la tabla fd_webusers

### 8.2.5 Tabla: fd\_itemsfeed

Se almacenan las entradas/posts de cada *feed* RSS.

Campo	Tipo	Descripción
<b>itf_id</b>	bigint	Clave primaria
itf_Title	varchar(255)	Título de la entrada

itf_Description	text	Descripción de la entrada
itf_Url	varchar(255)	URL de la entrada/post original
itf_Date	varchar(100)	Fecha de publicación del post
Ink_SiteUrl	varchar(255)	URL del site origen del RSS
itf_Visit	int	Número de veces que los usuarios han pulsado el enlace para ver el post. Se usa para "El Caldo"
itf_Link_fk	int	<i>Feed</i> al que pertenece la entrada. Clave foránea a la tabla fd_links

### 8.2.6 Tabla: fd\_userfeeds

Feeds que los usuarios han añadido a su perfil

Campo	Tipo	Descripción
<b>usf_Link_fk</b>	int	Clave primaria. Clave foránea a la tabla fd_links
usf_User_fk	bigint	Clave primaria. Clave foránea a la tabla fd_webusers
usf_Order	int	Orden en que el usuario decide mostrar sus <i>feeds</i>
usf_Folder_fk	bigint	Carpeta usuario donde el usuario coloca el <i>feed</i> . Clave foránea a la tabla fd_folders.

### 8.2.7 Tabla: fd\_feedvotes

Votos que los usuarios dan a los *feeds*. Como *fd\_userfeeds*, es una relación entre usuarios y *feeds*, pero como un usuario puede votar *feeds* que no tiene porque tener agregados a su perfil es por ello que se monta en una tabla diferenciada.

Campo	Tipo	Descripción
<b>fvo_Link_fk</b>	int	Clave primaria. Clave foránea a la tabla fd_links
<b>fvo_User_fk</b>	bigint	Clave primaria. Clave foránea a la tabla fd_webusers
fvo_Points	int	Puntuación del 1 al 5 que el usuario ha dado al <i>feed</i>

### 8.2.8 Tabla: fd\_comments

Comentarios que los usuarios hacen sobre los *feeds*.

Campo	Tipo	Descripción
<b>com_id</b>	int	Clave primaria.
com_Comment	text	Comentario.
com_Date	datetime	Fecha del comentario
com_Link_fk	int	Enlace que se comenta. Clave foránea a la tala fd_links
com_user_fk	bigint	Usuario que ha hecho el comentario. Clave foránea

		a la tabla fd_webusers
--	--	------------------------