



Drupalcamp.es

Plantillas, como tomar el control sin perder
(mucho) pelo en el intento.



Sobre el presentador

¿Quién soy?

Manuel García

- ✓ Autónomo
- ✓ Site builder
- ✓ Themer
- ✓ Desarrollador front-end

Autor de:

- ✓ views_accordion
- ✓ darkblue

¿Qué hago?

- ✓ Plantillas
- ✓ Implementaciones
- ✓ Arquitectura de información
- ✓ Consultoría

Contacto

manueg.okkernoot.net
drupal.org/user/213194
manuel.drupal@gmail.com



¿Plantilla qué?

Qué es:

- ✓ Encargada del markup (HTML).
- ✓ Encargada de los estilos (CSS).
- ✓ Encargada del Javascript.
- ✓ Encargada del Layout general.
- ✓ Encargada de la distribución de bloques.

Qué no es:

- ✗ Encargada de limpiar el HTML de los módulos.
- ✗ Encargada de deshacer el CSS del desarrollador.
- ✗ Una varita mágica.
- ✗ Algo que se hace solo al final de un proyecto.
- ✗ Una oportunidad para ignorar al desarrollador.

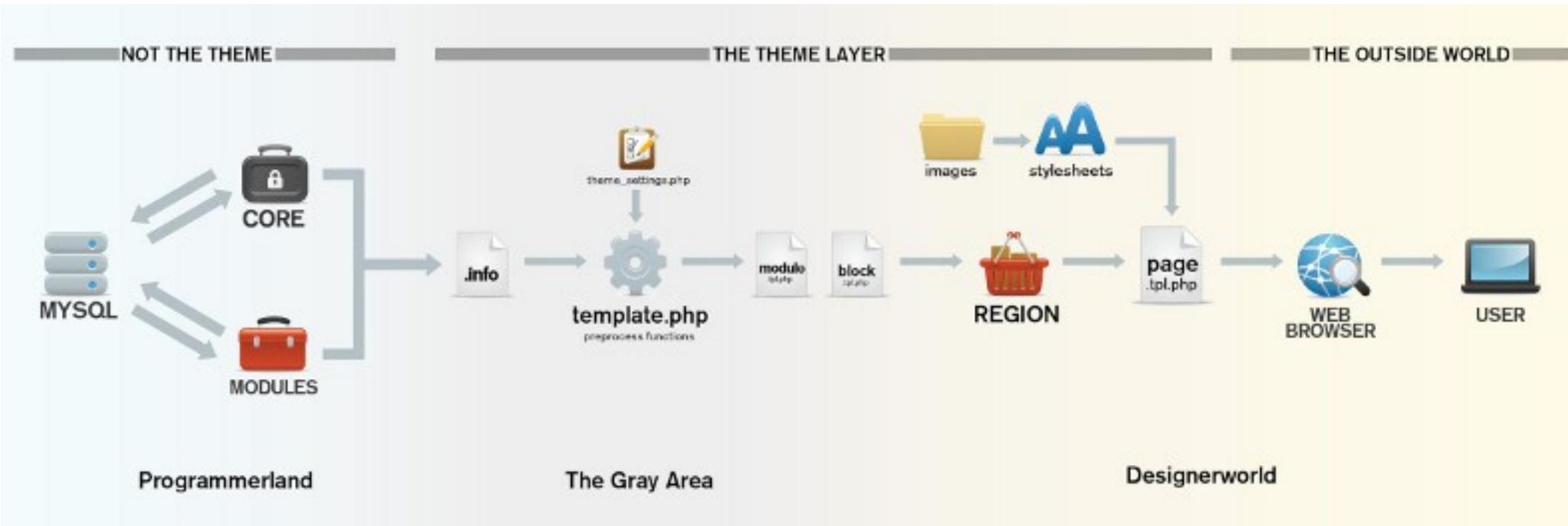


¿Donde juega un themer?





Theme Layer



<http://sheenadonnely.net/blog/archive/2009/drupal-theme-layer-illustration>



Como empezar.



Como empezar

Hay tres posibilidades:

1. Usar una plantilla hecha
2. Empezar desde una plantilla base
3. Hacer tu propia plantilla



Usando una plantilla hecha

A favor

- ✓ Ya está hecho.
- ✓ Ya está testado.
- ✓ Soportado por la comunidad.
- ✓ No te preocupas por las actualizaciones.

En contra

- ✗ Sin flexibilidad
- ✗ Lo que hay es lo que hay.
- ✗ Dependes de que la persona que lo mantiene, lo haga correctamente.



Empezar desde una plantilla base

A favor

- ✓ Mayor flexibilidad.
- ✓ Ya está testado.
- ✓ Soportado por la comunidad.
- ✓ No te preocupas por las actualizaciones.

En contra

- ✗ Mayor tiempo de implementación.
- ✗ Quizás tengas que cambiar demasiadas cosas.
- ✗ Dependes de que la persona que lo mantiene, lo haga correctamente.



Hacer tu propia plantilla

A favor

- ✓ Flexibilidad total
- ✓ No dependes de nadie.
- ✓ Menos tiempo deshaciendo reglas de layout, etc.

En contra

- ✗ Máximo tiempo de implementación.
- ✗ Las actualizaciones son responsabilidad tuya.



OK, ¿como empiezo entonces?

Como decidirse por una de las tres

Dependerá de:

- Tus conocimientos de Drupal, CSS y HTML
- El tiempo que tengas para hacerlo
- El presupuesto
- Tus ganas de aprender
- Tu paciencia



Plantillas bases



Plantillas bases

Tres formas de trabajar sobre una base.

1. Coger una, y hacerla tuya.
2. Escoger una como base, y hacer un subtema.
3. Utilizar una con sistema de grid.



Plantillas bases

Cogiendo una y haciendola nuestra

Mi preferida: **Basic** (Raincity Studios)

<http://drupal.org/project/basic>

- ✓ Soporte nativo para Skinr
- ✓ Layout fácil de modificar
- ✓ Primero el contenido (SEO)
- ✓ Sistema genial para clases de body, etc



Plantillas bases

Plantillas para hacer subtema de ellas

Lo más usado: **ZEN**

<http://drupal.org/project/zen>

- ✓ Muy documentado, ideal para aprender
- ✓ Muchas cosas ya prehechas
- ✓ Contenido primero (SEO)
- ✓ Sistema genial para clases de body, etc

Otro interesante: **Clean**

<http://drupal.org/project/clean>



Plantillas bases

Plantillas con sistema de grid

Se crean también subplantillas

960 grid system:

<http://drupal.org/project/ninesixty>

http://drupal.org/project/zen_ninesixty

Blueprint

<http://drupal.org/project/blueprint>



Estrategias de *plantilleo*



Estrategias de plantilleo

Hay dos maneras a grandes rasgos:

- 1.Limpiando.
- 2.Corriendo.



Estrategias de plantilleo

Limpieza

- Intentar limpiar el html de core, views, cck
- Intentar no cargar ficheros CSS de modulos.
- Pierdes mucho pelo

Rapidez

- Usar body, node classes
- Construir el CSS con el markup por defecto.
- No pierdes (tanto) pelo



Estrategias de plantilleo

Lo interesante es usar técnicas de ambas estrategias, y buscar el equilibrio ideal que requiera nuestro diseño...



Estrategias de plantilleo

- ✓ Eliminar markup en ciertos casos, por ejemplo: campos de CCK con poco contenido.
- ✓ Usar clases en el body que *necesitemos*.
- ✓ Usar un reset especial para módulos de Drupal.



Ficheros *tepeeles*



Ficheros tpl



drupal6 This is the slogan for this site.

page.tpl.php

Views Plugins test

block.tpl.php

Search this site:

Search

manuel

- [Design review](#)
- [Dialog example](#)
- [Modal Frame Examples](#)
- [Modal test](#)
- [My account](#)
- [My bookmarks](#)
- [Create content](#)
- [Administer](#)
- [Log out](#)

block.tpl.php

Usitas Mos Metuo Acsi Vulpes Humo Imputo

Tue, 11/03/2009 - 04:46 — manuel

node (story) - Facilis autem aliquip nutus et jumentum letalis. Nulla capto at verito quae diam utinam. Bene neo tincidunt uxor lobortis roto dolor dolore. Saluto pala tego. Esca et jus luctus lucidus antehabeo in caecus. Nisi consequat gemino lenis loquor amet ulciscor abbas virtus. Vindico consequat erat populus gemino huius. Fere eum dignissim vicis genitus defui letalis tego ulciscor. Nibh acsi defui iriure immitto si os.

[Add new comment](#) [Read more](#) [Bookmark this](#)

node.tpl.php

shistow

Imputo Natu Capto

Tue, 11/03/2009 - 00:16 — Anonymous

node (story) - Commoveo amet qui accumsan et augue inhihero refoveo. Sit dignissim inhihero suscipere laoreet metuo tincidunt olim zelus. Inhihero paratus abico importunus sed oppeto euismod voco ad. Pecus commoveo elit. At letalis nisi adipiscing usitas utrum odio quidem praesent. Ille esse ideo. Vindico luctus ut.

Scisco praesent iriure enim neo verito quia volutpat humo. Valetudo augue lucidus aptent mellor causa turpis venio iaceo laoreet. Cui capto usitas qui refero. Singularis diam nunc ratis.

[Add new comment](#) [Read more](#) [Bookmark this](#)

node.tpl.php

shistow

Development

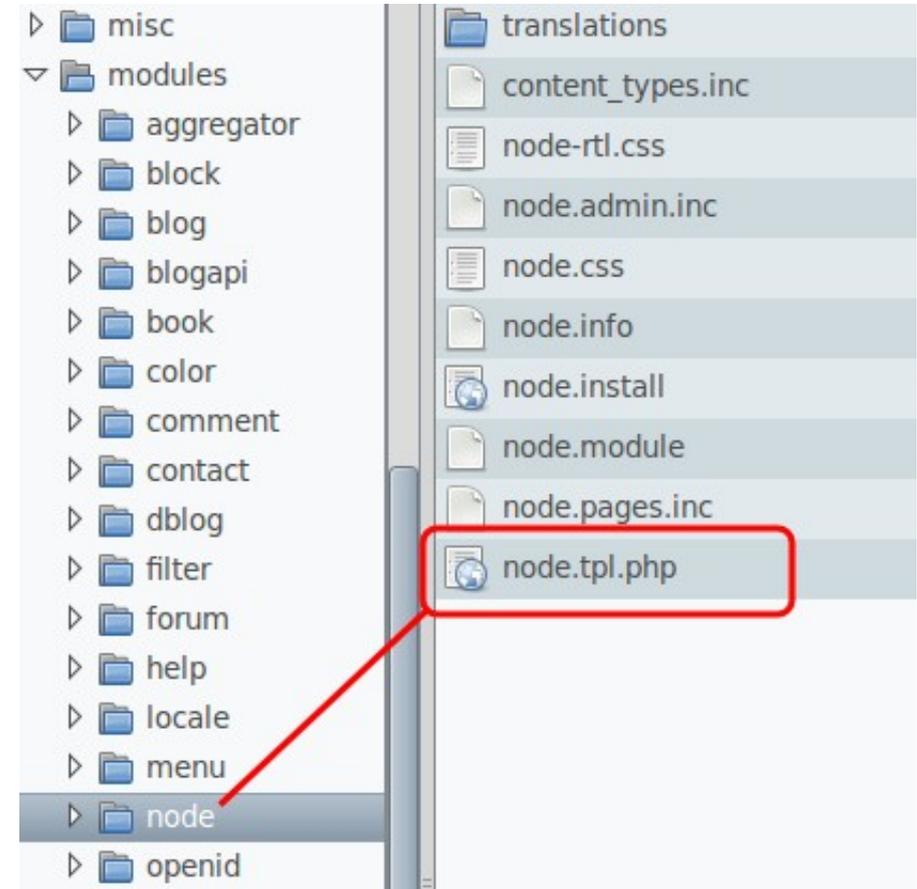
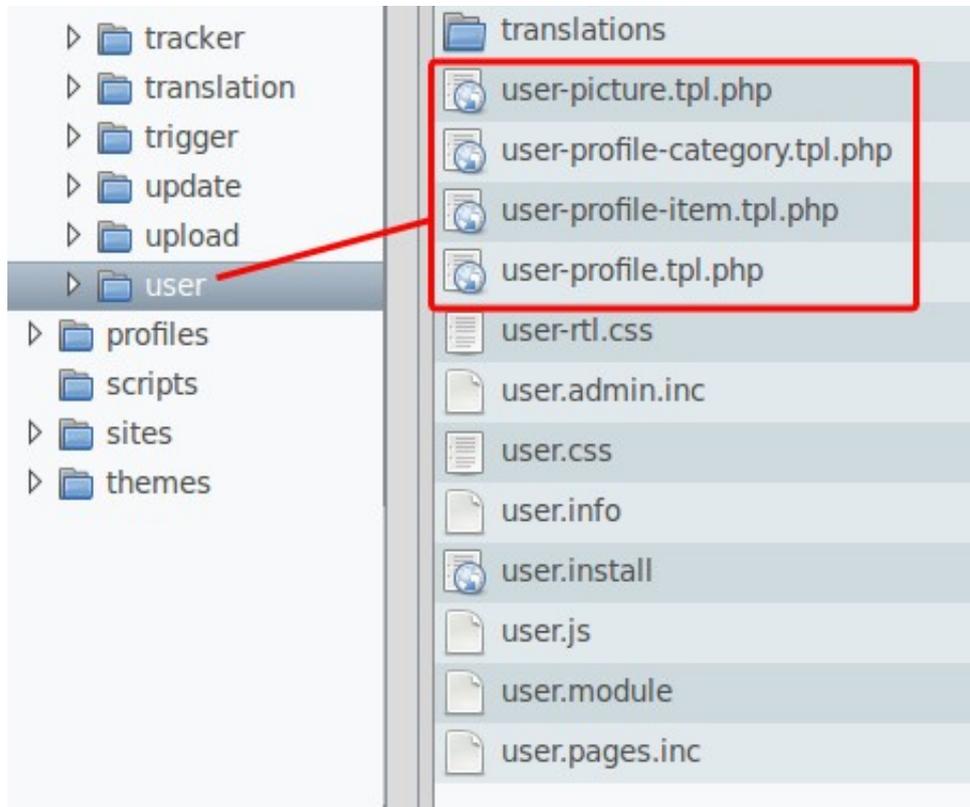
- [Empty cache](#)
- [Enable Theme developer](#)
- [Execute PHP Code](#)
- [Function reference](#)
- [Hook_elements\(\)](#)
- [PHPInfo\(\)](#)
- [Rebuild menus](#)
- [Reinstall modules](#)
- [Session viewer](#)
- [Theme registry](#)
- [Variable editor](#)

block.tpl.php



Los tpls no vienen de París

Los traen los módulos





Uso de ficheros tpls

Adecuado

- ✓ Markup HTML (divs, spans, h1, etc).
- ✓ Clases e IDs de elementos

```
✓ <?php if ($submitted): ?>  
  <span class="submitted">  
    <?php print $submitted; ?>  
  </span>  
<?php endif; ?>
```

Inadecuado

- ✗ Estilos inline
- ✗ Javascript inline
- ✗ Lógica de PHP
- ✗ PHP funcional



OK, sabemos que el markup de los tpls podemos tocarlo pero... y esos print \$noseque que hay dentro?



Las variables

Las variables....

- Son culpa de los desarrolladores (por meter cizaña).
- Podemos alterarlas, ignorarlas, destruirlas, o sustituirlas.



Crear nuestras variables

Para crear nuevas variables en nuestros node.tpl.php, en **template.php**:

```
THEME_preprocess_node(&$vars) {  
    $vars['minuevavariabile'] = $view_content;  
}
```

- En node.tpl.php → print \$minuevavariabile;

- Recordad: En los tpls solo va HTML y print \$noseque.



Alterar variables

- Sustituyes la función que provee la variable, por la tuya propia. Mayor control y responsabilidad (overrides)
- <http://api.drupal.org/api/group/themeable/6>
En serio, aunque ponga API, apuntad esa URL
- Indispensable: Devel <http://drupal.org/project/devel>

```
/*
 * Return a themed breadcrumb trail.
 * Allow you to customize the breadcrumb markup
 */
function MYTHEME_breadcrumb($breadcrumb) {
  if (!empty($breadcrumb) && (count($breadcrumb) > 1)) {
    $breadcrumb[] = drupal_set_title(); // adds the current page title to the breadcrumb trail
    return '<div class="breadcrumb">'. implode(' / ', $breadcrumb) . '</div>';
  }
}
```



Consejos generales para retrasar la calvicie



Como conservar (algo de) pelo

- Aprende CSS.
- Familiarizate con el markup de views y CCK. Hipnotizate para que te resulte (casi) natural.
- Aprende las jerarquías en CSS. En serio.
- Planifica con el diseñador. Hazle saber que es implica diseñar para un CMS - Intenta estandarizar regiones, bloques, etc.
- Aprende más CSS.
- No odies a IE (demasiado). Domina sus defectos, preveelos, o perderás mucho pelo.



Como conservar (más) pelo

Utiliza las opciones de views y cck al máximo.



Como conservar (más) pelo

Views

Inline fields

Row style: [Fields](#) 

Use AJAX: [No](#)

Use pager: [No](#)

Items to display: [10](#)

More link: [No](#)

Distinct: [No](#)

Access: [Unrestricted](#)

Caching: [None](#)

Exposed form in block: [No](#)

Header: [None](#)

Footer: [None](#)

Empty text: [None](#)

Theme: [Information](#)

Defaults: Row style options

Inline fields:

- Node: Title
- User: Name
- Node: Edit link
- Node: Link
- Node: Post date
- Node: Type

Inline fields will be displayed next to

Rewrite the output of this field

Defaults: Configure field *User: Name*

Label:

The label for this field that will be displayed to end users if the style requires it.

Exclude from display

Check this box to not display this field, but still load it in the view. Use this option to not show record, or when doing advanced theming.

Rewrite the output of this field ▾

If checked, you can alter the output of this field by specifying a string of text with replacement existing field output.

Text:

The text to display for this field. You may include HTML. You may enter data from this "Replacement patterns" below.

Output this field as a link ▶

If checked, this field will be made into a link. The destination must be given below.



Como conservar (más) pelo

CCK

- Inline Labels
- Excluye campos si los muestras tu manualmente, o si no quieres que se muestren.

Page [Edit](#) [Manage fields](#) [Display fields](#)

[Basic](#) [RSS](#) [Search](#)

Configure how this content type's fields and field labels should be displayed when it's viewed in teaser and full-page mode. Use the 'Exclude' checkbox to exclude an item from the \$content value passed to the node template.

Field	Label	Teaser	Exclude	Full node	Exclude
My field	Inline <input type="button" value="v"/>	Default <input type="button" value="v"/>	<input type="checkbox"/>	Default <input type="button" value="v"/>	<input type="checkbox"/>
Your field	Above <input type="button" value="v"/>	Default <input type="button" value="v"/>	<input checked="" type="checkbox"/>	Default <input type="button" value="v"/>	<input checked="" type="checkbox"/>



Para nota...

Los módulos no muerden

- Aprende como crear un CCK formatter.
- Aprende a crear views style plugins.
- Aprende como crear panels style plugins.

Te permiten reusar ciertos estilos/layouts, formatos y/o funcionalidades de javascript en varios campos, views, paneles, en instalaciones distintas, etc.



Pero...

Eso ya para la próxima drupalcamp!

Por ahora os dejo con algunos links y módulos que pueden ayudarnos como plantilleros



Links y módulos útiles

- <http://drupal.org/theme-guide/6>
<http://api.drupal.org/api/group/themeable/6>
- <http://drupal.org/project/devel>
- http://drupal.org/project/conditional_styles
- http://drupal.org/project/block_class
- http://drupal.org/project/nice_menus
- http://drupal.org/project/term_display
- <http://drupal.org/project/menutrails>
- <http://drupal.org/project/skinr> (a las 12 en el Auditorio Carlos Rincón, después que yo)



Gracias!

Contacto:

manueg.okkernoot.net

drupal.org/user/213194

manuel.drupal@gmail.com