



# **Drupal y Middelman, una extraña pareja**

Como hemos trabajado en el último proyecto

Oskar

@oskarcervo

friky, drupalero,  
sestaoarra.

Trabajo en The  
Cocktail como  
desarrollador de  
proyectos con  
php/Drupal



# The Cocktail

+ - 150 personas,  
En América, Asia,  
Europa.

Hasta hace año y  
medio solo  
trabajaban con  
Ruby.



# ¿Por qué esta charla?

Conocer vuestra opinión sobre como hemos desarrollado el último proyecto.

No me dejan mostrar datos de clientes :(

# De que va la charla

Explicar lo que supuesto trabajar con un departamento de Front muy metido en el mundo Ruby y nada en el mundo Drupal

# De que va la charla

No quiere ser una charla técnica, más bien terapia de grupo, o par programming, vosotros sereís el patito ;)

# Os podéis sentir así



Drupal Day Bilbao 2014

# O así (mente colectiva)



**Drupal Day Bilbao 2014**





# Como hemos encarado el reto

# El departamento de Front

En The Cocktail con RonR los departamentos de Front y Back están bien separados, no es tan difuso como en Drupal

# Front

Front se encarga de crear las vistas de Rails (¿erb?), o simplemente hacen la maquetación que otros integran.

# Front

Front trabaja con la herramienta Middelman, es su entorno de trabajo y por ahora saben lo mínimo de Drupal (nada).

# Front

Middleman es un generador de páginas estáticas de html.

# Front

Middleman compila el código de ruby y genera páginas estáticas con el ejemplo del html renderizado

# Front

Archivo Editar Ver Buscar Terminal Ayuda

```
m breadcrumb.html.erb
```

```
1 <%  
2   sections ||= {  
3     "Inicio"=>"#",  
4     "Historia"=>"#"  
5   }  
6 %><nav class="m_breadcrumb"><% sections.each do |label, href| %>  
7   <a href="<%=href%>"><%=label%></a>  
8 <% end %></nav>
```

# Retos

## Example

< INICIO / HISTORIA

## Source code

```
1 <nav class="m_breadcrumb">
2   <a href="#">Inicio</a>
3
4   <a href="#">Historia</a>
5 </nav>
```



# Retos

[Otro ejemplo](#) :)

[Segundo ejemplo](#) :)

[Tercer ejemplo](#) :)

# Front

Están acostumbrados a usar librerías de js/css de terceros y o diseñadas por ellos, y no siempre dependen de jquery



**Front**

Y en esto llegó Drupal



**Los restos a los que nos  
hemos enfrentado**

# Retos



Front pilota de Ruby no de php/Drupal

# Retos



Front está acostumbrado a las últimas versiones de JQuery

# Retos

Front está acostumbrado a su propio marcado html.

Imaginad la cara cuando vieron el código de Drupal

# Retos

Imaginad la cara cuando vieron el código de  
Drupal





# Retos a realizar

Integrar en Drupal :

- CSS
- Js
- html

# Soluciones



Actualizar mediante Jquery Udate a la versión 1.8.x.

# Soluciones

Decidimos usar Mothership como tema por los siguientes motivos.

Al final optamos por un tema hijo de mothership.

# Mothership

Limpia todo el css del core y contrib modules, por lo que usamos únicamente los assets de Middleman

No hubo problemas con esto.

# Soluciones

Mothership tiene funciones de procesamiento que limpia las clases y html además de traer sus propios tpls, genera un html limpio.

# Bartik

```
▼ <div id="block-system-powered-by" class="block block-system contextual-links-region">
  ▶ <div class="contextual-links-wrapper contextual-links-processed">...</div>
  ▼ <div class="content">
    ▼ <span>
      "Powered by "
      <a href="http://drupal.org">Drupal</a>
    </span>
  </div>
</div>
</div>
```

# Mothership

```
▼ <div class="contextual-links-region clearfix">  
  ▶ <div class="contextual-links-wrapper contextual-links-processed">...</div>  
    ▼ <span>  
      "Powered by "  
      <a href="http://drupal.org">Drupal</a>  
    </span>  
    ::after  
  </div>  
  ::after  
  ..
```



# Nuestro tema

```
block--views--consejos-v-deos...ustom/tc/te
```

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
```

```
block--views--consejos-v-deos-block-7.tpl.php
```

```
1 <?php print $content ?>
```

```
~  
~  
~  
~
```

# Soluciones

Creamos un tema hijo de Mothership para reescribir todos los tpls, agregar nuevos tpls de muchas plantillas.

# Qué hemos visto?

“Rehacer” dos veces el marcado de html.

No reutilizar el marcado de html === más lento

Mejor marcado html, mejor css === mayor calidad.

# Qué hemos visto?

CSS más optimizado.

Jss por fuera de Drupal, nos ha dado algunos problemas con ajax, y no tenemos acceso a la API de js de Drupal.

# Qué hemos visto?

La integración de todo es manual.

Estamos mirando si se podría compilar middleman para que genere un tema de drupal.



**¿Preguntas?**



**Gracias**