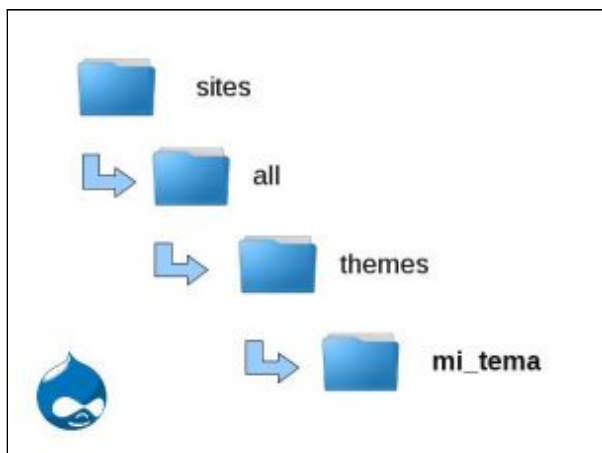




Un tema para Drupal en pocos minutos

@silenceway
bh@axul.net
skype: silencew

Para crear un tema para Drupal desde cero debe conocerse algo de programación (php) y algo de html.



Lo primero que debe hacerse es crear el directorio donde vamos a guardar nuestros archivos.

Se recomienda usar la carpeta de themes que se encuentra dentro de la carpeta sites, la ruta es sites – all (o el nombre del sitio) - themes. En nuestro caso vamos a crear una carpeta llamada mi_tema y así llamaremos el tema nuevo que vamos a utilizar.

El siguiente paso es crear los archivos básicos de un tema, ellos son un archivo info que va a guardar la información de versión, de autor, de estilos y de regiones (que veremos más adelante); el otro archivo es page.tpl.php que llevará todo el html de nuestro tema; un archivo

de README para dar información de uso y un archivo donde se guardarán los estilos del tema.

El primer paso es crear un archivo de información de tema, que en nuestro caso llamaremos de la misma forma que nuestro directorio: mi_tema.info.

En el archivo debe existir inicialmente el nombre del tema (name), una descripción (description), a qué drupal va dirigido (core), el nombre del archivo de estilo (stylesheet), la versión de nuestro tema (version) y si se quiere la fecha de modificación del archivo en formato timestamp (timestamp).



```
name = Mi Tema
description = Un tema para probar Drupal
core = 7.x
stylesheets[all][] = style.css
; stylesheets[print][] = print.css
version = "0.1"
project = "drupal"
; 2012-05-02
timestamp = "1336866325"
```

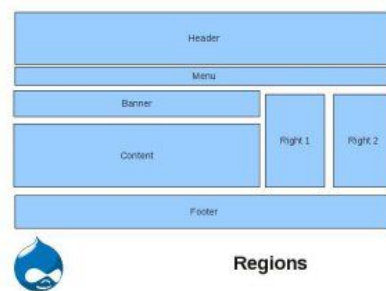
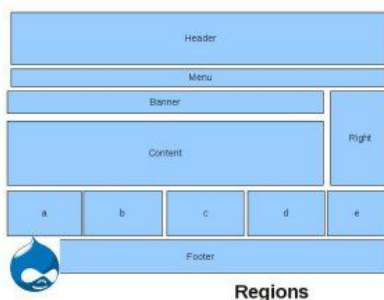
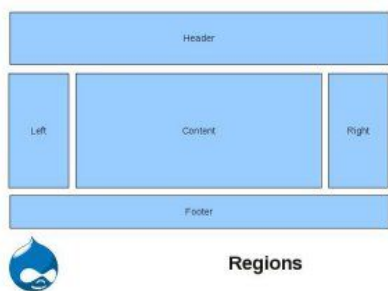
Con este archivo creado, podemos ir al menú de administración de Apariencia y ver que el tema ya es reconocido por Drupal para ser elegido.



Para el siguiente paso hay que definir las regiones o zonas que van a ser usadas dentro del tema. El tema básico de Drupal tiene un encabezado, dos barras laterales a cada lado del contenido y un pie.

Esas regiones pueden crearse libremente y deben ser definidas en el archivo info para que Drupal las reconozca y las despliegue al crear las variables de la página a mostrar. Cada región debe crearse

con un nombre reconocible tanto para el usuario como para Drupal.



Definir las regiones es definir cómo queremos que el usuario vea nuestro sitio. Puede ser simple su uso, o puede ser un diseño más complicado como se ve en la imagen 2. Para nuestro sitio usaremos las regiones como están diseñadas en la imagen 3 y crearemos varios divs en una página html que llamaremos page.tpl.php. Además agregaremos al archivo mi_tema.info la información del total de regiones que creamos con sus nombres y descripciones.

mi_tema.info

```
name = Mi Tema
description = Un tema para probar Drupal
core = 7.x
stylesheets[all][] = style.css
; stylesheets[print][] = print.css

regions[header] = Header
regions[menu] = Menu
regions[banner] = Banner
regions[content] = Content
regions[right_a] = Right sidebar A
regions[right_b] = Right sidebar B
regions[footer] = Footer

; 2012-05-02
version = "0.1"
project = "drupal"
datestamp = "1336866325"
```

Nos damos cuenta que cada región lleva un nombre interno que se escribe dentro de corchetes y un nombre descriptivo que es como lo verá el usuario en la página de configuración de bloques.

Luego vamos a crear nuestra página de plantilla que muestre este html en el sitio y poder asignar la plantilla como predeterminada. Este es un html básico que puede generar la página que queremos después de agregarle estilos. Primero lo añadimos sin agregarle nada. Y vamos y lo activamos en la configuración de apariencia del sitio.

page.tpl.php

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <div id="wrapper">
      <div id="header">Header</div>
      <div id="menu">Menu</div>
      <div id="wrapper-main">
        <div id="wrapper-content">
          <div id="banner">Banner</div>
          <div id="content">Content</div>
        </div>
      <div id="right_a">Right A</div>
```

```

        <div id="right_b">Right B</div>
        <div class="clear"></div>
    </div>
    <div id="footer">Footer</div>
</div>

</body>
</html>

```

Si se recuerda en el archivo de información de tema, un parámetro creado añadía un archivo de estilo, lo creamos para que tenga cara de nuestro diseño el sitio. Igual limpiamos el archivo page.tpl.php porque si revisamos el código fuente, Drupal ya está incluyendo código predeterminado como las cabeceras de un sitio html. Y separamos el header y el footer en archivos diferentes para su fácil administración.

page.tpl.php

```

<?php require_once("header.tpl.php"); ?>
    <div id="wrapper-main">
        <div id="wrapper-content">
            <div id="banner">Banner</div>
            <div id="maion">Content</div>
        </div>
        <div id="right_a">Right A</div>
        <div id="right_b">Right B</div>
        <div class="clear"></div>
    </div>
<?php require_once("footer.tpl.php"); ?>

```

header.tpl.php

```

<div id="wrapper">
    <div id="header">Header</div>
    <div id="menu">Menu</div>

```

footer.tpl.php

```

    <div id="footer">Footer</div>
</div>

```

style.css

```

*
{
margin: 0;
padding: 0;
}
.clear
{
    clear: both;
}
#wrapper
{
    width: 960px;
}

```

```
        margin-left: auto;
        margin-right: auto;
    }
#header
{
    width: 950px;
    padding: 5px;
    background-color: #BBEEDD;
}
#menu
{
    width: 950px;
    padding: 5px;
    background-color: #EEEEBB;
}
#wrapper-main
{
    width: 960px;
}
#wrapper-content
{
    width: 480px;
    margin-right: 10px;
    float: left;
}
#banner
{
    width: 480px;
    background-color: #BBEEBB;
}
#maion
{
    width: 480px;
    background-color: #EECCDD;
}
#right_a
{
    width: 220px;
    margin-left: 10px;
    margin-right: 10px;
    background-color: #BBEEBB;
    float: left;
}
#right_b
{
    width: 220px;
    margin-left: 10px;
    background-color: #BBEEBB;
    float: left;
}
#footer
{
    width: 950px;
    padding: 5px;
    background-color: #EE99DD;
}
```

Ahora que ya hay un borrador del tema, se debe mostrar el contenido del sitio que en este momento no se está mostrando. Para esto añadimos el código php en cada una de las regiones creadas. Para probar se puede añadir el código siguiente cambiando el contenido del div main por el código que imprime este contenido, este código lleva el nombre de la región dentro de corchetes. Y agregando un bloque a este contenido en el administrador de Bloques.

```
regions[content] = Content
<div id="main"><?php print render( $page['content'] ); ?></div>
```

Igual con el header y con el footer y con todos los otras regiones del tema.

```
regions[header] = Header
<div id="header"><?php print render( $page['header'] ); ?></div>
```

```
regions[footer] = Footer
<div id="footer"><?php print render( $page['footer'] ); ?></div>
```

Al header además se le agrega la información de logo por código php. El logo es un archivo que se añade al tema con extensión png que puede ser cambiado en las opciones que se agregan al módulo por defecto.

header.tpl.php

```
<div id="wrapper">
  <div id="header">

    <?php if ($logo): ?>
    <div id="logo">
      " title="<?php print $site_name; ?>" id="logo" />
    </div>
    <?php endif; ?>

    <?php print render( $page['header'] ); ?>
  </div>
  <div id="menu"><?php print render( $page['menu'] ); ?></div>
```

Además hay que agregar dentro del código de contenido dos espacios especiales que son el de mensajes internos de drupal, de la navegación por tabs y también puede incluir la navegación por *migajas de pan* (breadcrumbs).

Además si se quiere mostrar bloques en caso de que solo tengan contenido y no en todo momento, puede usarse la condición *if* (si) para que lo esconda. En este caso se esconde el banner si no es necesario.

page.tpl.php

```
<?php require_once("header.tpl.php"); ?>
  <div id="wrapper-main">
    <div id="wrapper-content">
```

```

<?php if ($banner): ?>
<div id="banner">Banner</div>
<?php endif; ?>
<div id="main">
  <div id="breadcrumb"><?php print $breadcrumb;
?></div>

  <?php if ($messages): ?>
  <div id="messages"><?php print $messages; ?
></div>

  <?php endif; ?>
  <div class="clearfix"><?php if ($tabs): ?><?
php print render($tabs); ?></div><?php endif; ?>

  <?php if ($page): ?>
  <h2<?php print $title_attributes; ?><?php
print $title; ?></h2>

  <?php endif; ?>

  <?php print render( $page['content'] ); ?
></div>
</div>
<div
  id="right_a"><?php
  print
render( $page['right_a'] ); ?></div>
<div
  id="right_b"><?php
  print
render( $page['right_b'] ); ?></div>
<div class="clear"></div>
</div>
<?php require_once("footer.tpl.php"); ?>

```

Si se quieren añadir imágenes dentro del tema para ser publicadas en regiones específicas, podemos hacer el llamado a la misma usando funciones internas de Drupal. Es recomendable usar una carpeta para organizar las imágenes igual que otra que se puede llamar js para el javascript. En este ejemplo estamos mostrando en la región del contenido una imagen permanentemente.

```

```

Para finalizar, se debe crear una imagen de la pantalla para mostrarla en la página de selección de temas. Esta imagen llamada screenshot.png debe ser de 294 x 219 pixeles.



Si se conoce algo de programación, se pueden llenar ciertas regiones por programación sin permitir cambiarlas como por ejemplo agregar a la región menú, el menú seleccionado como menú principal (primary menu) en el sitio. Para esto se usa el archivo template.php que se crea en la misma carpeta del tema y que mediante *hooks* (funciones que permiten interactuar con el núcleo y con otros módulos al momento de generar el

código de drupal) nos permite modificar la salida final. Este ejemplo ilustra la lectura del menú principal:

template.php

```
<?php
function mi_tema_preprocess_page(&$vars, $hook)
{
  if (isset($vars['main_menu'])) {
    $vars['primary_nav'] = theme('links__system_main_menu', array(
      'links' => $vars['main_menu'],
      'attributes' => array(
        'class' => array('links', 'main-menu'),
      ),
      'heading' => array(
        'text' => t('Main menu'),
        'level' => 'h2',
        'class' => array('element-invisible'),
      )
    ));
  }
  else
    $vars['primary_nav'] = FALSE;
}
?>
```

Para crear la función debemos llamarla con el nombre del tema antes de preprocess_page (que es la función de procesamiento de página). Y lo que crea la variable \$primary_nav que imprime este menú.

```
<div id="menu"><?php print $primary_nav; ?></div>
```

Así es un tema en Drupal a grandes rasgos. Pueden descargar la plantilla funcional pero con bastantes falencias en estilos y diseño. Sólo es una muestra de cómo podemos generar una plantilla en Drupal desde cero. Hay otras formas de crearlas que es usando subtemas de temas generados como Omega y Zen que son proyectos que se deberían revisar para la generación de temas.

Gracias.