



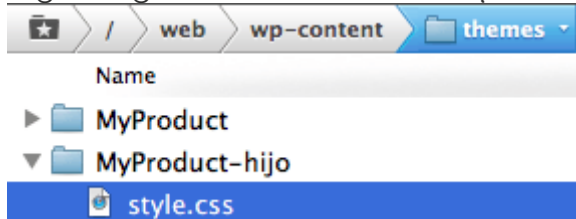
Fer un tema child

Ya hace tiempo que hablamos de los [temas hijos o "child themes"](#), incluso debatimos algo sobre [sus ventajas o inconvenientes](#), pero creo que faltaba una **guía sencilla y rápida de cómo crear un tema hijo**.

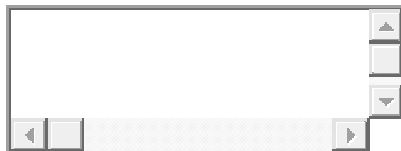
Así que vamos a ello, es más fácil de lo que piensas ...

1. Lo básico

Lo primero es **crear una carpeta** dentro de `/wp-content/themes/` como si otro tema fuera. Puedes elegir el nombre que quieras aunque parecería lógica alguna referencia sobre qué es, por ejemplo:



Una vez creada la carpeta simplemente **crea la hoja de estilos**, es decir, un fichero `style.css` dentro. Si lo prefieres, para ahorrarte tecleo, copia la cabecera de la hoja de estilos del tema principal, del padre. En cualquier caso la estructura básica sería esta:



```
1 /*
2 Theme Name: MyProduct Hijo
3 Theme URI: http://semanticae.es
4 Version: 4.0
5 Description: Tema hijo de MyProduct para hacer cambios sin riesgo
6 Author: Semanticae
7 Author URI: http://semanticae.es
8 Template: MyProduct
9 */
10
11 @import url("../MyProduct/style.css");
12
13 /*----- Empieza a añadir cambios aquí abajo -----*/
```

Las líneas importantes son estas:

- Theme name: lógicamente diferente del nombre del tema padre, pero pon el que quieras.



- Template: aquí no hay improvisación posible, debe ser el nombre de la carpeta del tema padre. Si la carpeta se llama `wordpress-style-theme` eso es lo que debes poner, sin cambio alguno o no funcionará.
- `@import url(xxxx)`: Usando la estructura del ejemplo, aquí es donde le decimos al tema hijo de donde heredar la hoja de estilos original, de modo que al activarlo funcione todo.

La magia empieza a partir de la última línea, pues **una vez activemos el tema hijo ya podemos personalizarlo para obtener el mismo diseño del tema original (padre) pero con las modificaciones que hagamos en el tema hijo**, que no se verán afectadas por actualizaciones del tema original, que es una de las **principales virtudes de los temas hijo**.

Así que ya solo con esto **podemos activar el tema hijo, igual que cualquier otro tema**. Nuestra web no notará cambio alguno pues de momento no hemos introducido modificación alguna, lo único que hace es importar archivos (`Template: MyProduct`) y hoja de estilos (`@import url("../MyProduct/style.css");`) del tema padre.



Administrar temas | Instalar temas


Nuevo tema activado. [Visitar sitio](#)

Tema actual
MyProduct Hijo
Por [Semanticae](#) | Versión 4.0

Tema hijo de MyProduct para hacer cambios sin riesgo
Este [tema hijo](#) requiere su tema superior, MyProduct.

[Personalizar](#) OPCIONES: [Widgets](#) | [Menús](#) | [MyProduct Theme Options](#) | [Fondo](#)

Temas disponibles



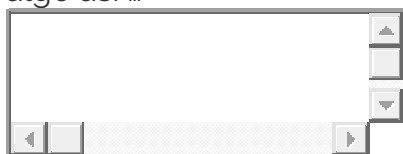
MyProduct
Por [Elegant Themes](#)

[Activar](#) | [Vista previa](#) | [Detalles](#) | [Borrar](#)

2. Añadiendo estilos

A partir de este momento **el tema hijo activo hereda el CSS de la hoja de estilos original, salvo que añadamos clases personalizadas**, en cuyo caso tomará primero las del tema hijo, aplicando el resto de la hoja de estilos del tema padre, así de simple.

De este modo, si modificamos la hoja de estilos del tema hijo añadiendo algo así ...



- 1 /*
- 2 Theme Name: MyProduct Hijo
- 3 Theme URI: <http://semanticae.es>
- 4 Version: 4.0
- 5 Description: Tema hijo de MyProduct para hacer cambios sin riesgo
- 6 Author: Semanticae
- 7 Author URI: <http://semanticae.es>



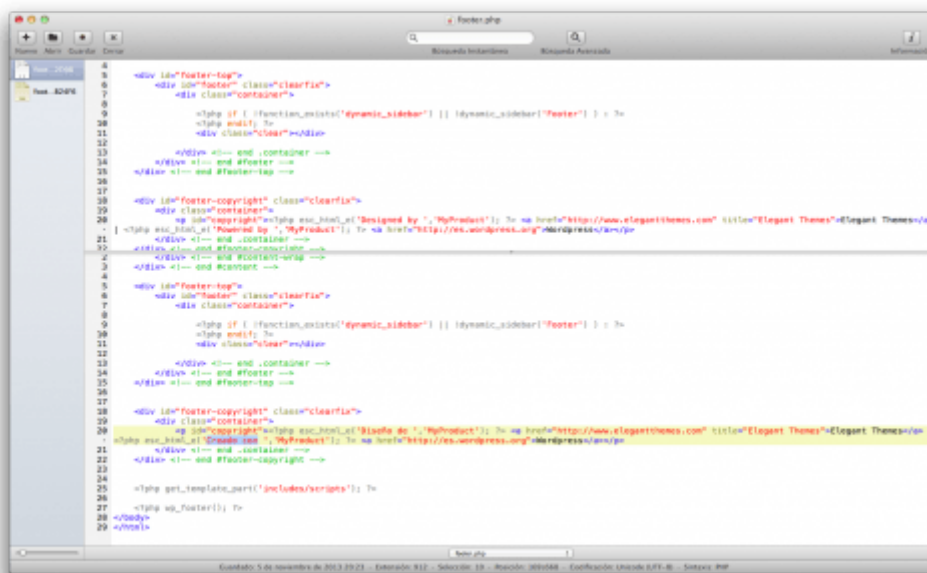
```
8 Template: MyProduct
9 */
10
11 @import url("../MyProduct/style.css");
12
13 /*----- Empieza a añadir cambios aquí abajo -----*/
14
15 a { text-decoration: none; color: #5f5f5f; }
16 a:hover { text-decoration: underline; }
```

... estaremos añadiendo estilos personalizados a los enlaces, y el resto de estilos los aplicará del tema padre.

3. Añadiendo plantillas

Lo siguiente que podemos hacer es **personalizar los ficheros del tema** o incluso añadir nuevos, me refiero a “*los php*”, y el proceso es igual de sencillo.

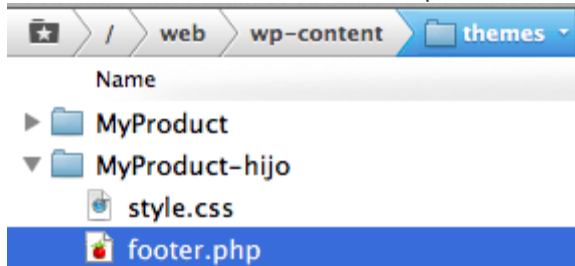
Si, por ejemplo, queremos personalizar el pie de página simplemente copiamos el fichero `footer.php` original del tema padre a la carpeta de nuestro tema hijo y hacemos las modificaciones en el tema hijo. Una vez guardemos los cambios el resultado se verá de inmediato para esa plantilla concreta, permaneciendo inalterable el resto del aspecto de la web.



Es importante tener en cuenta aquí que debes seguir la estructura de carpetas del tema padre. O sea, que si el fichero `blog-style.php` (por ejemplo) del tema padre está en `/tema-padre/includes/blog-style.php` en el tema hijo debe seguir la misma estructura `/tema-hijo/includes/blog-`



style.php. Por lo demás, de puro fácil casi asusta.



Puedes, como he apuntado antes, añadir nuevas plantillas específicas, de página o lo que sea.

4. Funciones

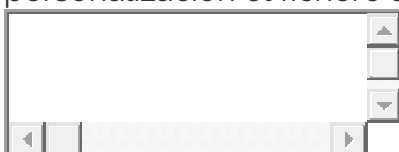
Un tema hijo básicamente funciona así:

1. WordPress revisa el tema activo (el tema hijo) y carga los archivos del tema padre (Template: MyProduct)
2. WordPress a continuación carga la hoja de estilos importada (@import url("../MyProduct/style.css");)
3. Carga primero las modificaciones en la hoja de estilos del tema hijo y ficheros modificados en el tema hijo, y completa lo que le falta con la hoja de estilos y archivos originales del archivo padre, en ese orden.
4. Carga el archivo de funciones (functions.php) del tema hijo y luego carga el fichero de funciones original o padre.

La sutil diferencia es que WordPress prioriza el archivo `functions.php` del tema hijo sobre el del tema original, o padre, pero no lo sobrescribe sino que solo tendrá en cuenta las modificaciones (funciones añadidas) del tema hijo, no sobrescribe – por decirlo de algún modo – el original.

O sea, a efectos prácticos, que no tienes que copiar entero el fichero `functions.php` del tema padre sino solo crear uno vacío (por supuesto con la apertura PHP) y **poner ahí las funciones específicas que quieras añadir como personalizaciones** que, por supuesto, no quieras perder con las actualizaciones del tema.

Un ejemplo muy típico sería añadir un *favicon*. Si solo añades esta personalización el fichero sería algo tan básico como esto:





```
1 <?php
2 // Favicon
3 function favicon_link() {
4     echo '<link rel="shortcut icon" type="image/x-icon" href="/favicon.ico" />'. "\n";
5 }
6 add_action( 'wp_head', 'favicon_link' );
```

Y ya está, a partir de aquí es cosa tuya **añadir las personalizaciones que quieras hacer, que no se perderán con las actualizaciones del tema padre** salvo que este cambie sustancialmente su estructura de archivos claro, que puede pasar aunque no es habitual.