

INSTALAR Y CONFIGURAR SAGE (ROOTS) – WORDPRESS

Instalar y configurar Sage (Roots) – WordPress

Roque Arnás, 7 de Mayo de 2016, **WordPress**

Sage by Roots Instalación y configuración



Didesweb

Relacionados

[Bower](#)

[Xampp](#)

[Android](#)

[AngularJS](#)

[Json](#)

Recursos

[Software](#)

[Herramientas online](#)

[Utilidades CSS](#)

Sage es un *Starter Theme* para *WordPress* que nos permite construir temas de *WordPress* de

forma ágil y profesional, ofreciendonos una base sólida para desarrollar nuestros temas personalizados sin tener que preocuparnos de la optimización, la jerarquía de las plantillas, los microformatos, la estructura semántica y otras tareas repetitivas a las que habitualmente tenemos que enfrentarnos a la hora de desarrollar un tema de WordPress. Está bien documentado, se actualiza frecuentemente y dispone de soporte continuo por parte de sus creadores.

Si estais familiarizados con Genesis Framework o Underscores, os encantará *Sage*, esta herramienta es mucho mas que un *Starter Theme*, es una evolución en la construcción de cualquier tema personalizado gracias a las características que utiliza para su desarrollo, como Gulp, Bower o Sass entre otras, proporcionandonos un *Workflow* (flujo de trabajo) moderno, limpio y ágil, además de hacer mucho más rápido y cómodo el trabajo en equipo, estas son algunas de las características mas importantes de *Sage*:

Bower

Para la gestión de dependencias de paquetes Front-end

Gulp

Compila archivos, comprueba errores en JS, concatena y minifica código CSS y JS, compila SASS, optimiza las imágenes.

BrowerSync

Mantiene el navegador sincronizado con los cambios en el desarrollo, el CSS y el JS sin recargar la página, admite el uso de varios navegadores.

HTML5 Boilerplate

Mobile-first, responsive

DRY

ARIA Roles

Microformatos

Preparado para Multilenguaje

Bootstrap

Herencia de plantillas

Imagen

Validación

Posicionamiento

Herramientas

Conversor Html a texto plano

Lorem Ipsum Generator

Html Head Generator

Contador palabras/caracteres

Calculadora de iva

Publicidad



Productos con Denominación de
Origen

Como se puede comprobar en las imagenes, voy a utilizar Debian 8 Jessie para realizar la instalación de Sage, el proceso es el mismo en Mac y en Windows, aunque para los usuarios de sistemas Windows os recomiendo el artículo [Como borrar la carpeta node_modules en Windows](#), para que os eviteis los mismos problemas que he experimentado personalmente hace unos meses con las rutas de más de 255 caracteres.

Instalación

Requisitos previos

Para trabajar con Sage necesitamos tener instaladas y configuradas una serie de herramientas en nuestro equipo, como NoodeJS, Git y PHP, además de manejar dependencias de NodeJS con npm y un entorno de trabajo como Xampp o Lemp con un WordPress instalado y en funcionamiento.

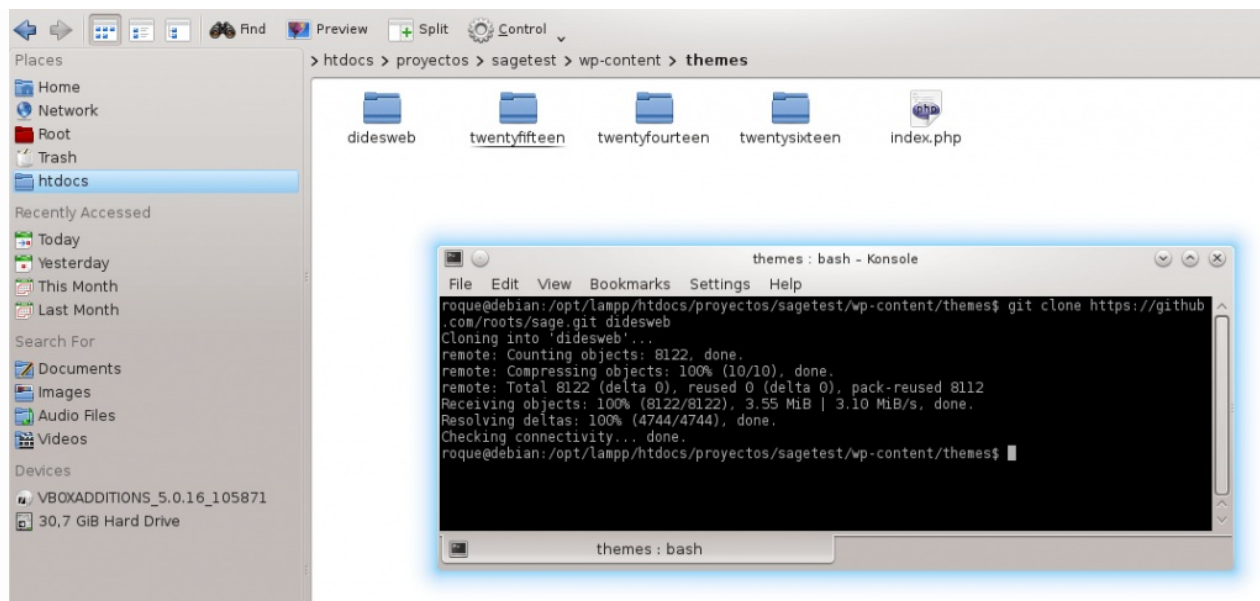
Clonando el tema desde su repositorio Git

Abrimos el terminal en la carpeta themes de WordPress y escribimos en la terminal:

git clone <https://github.com/roots/sage.git> NOMBRE_DEL_TEMA

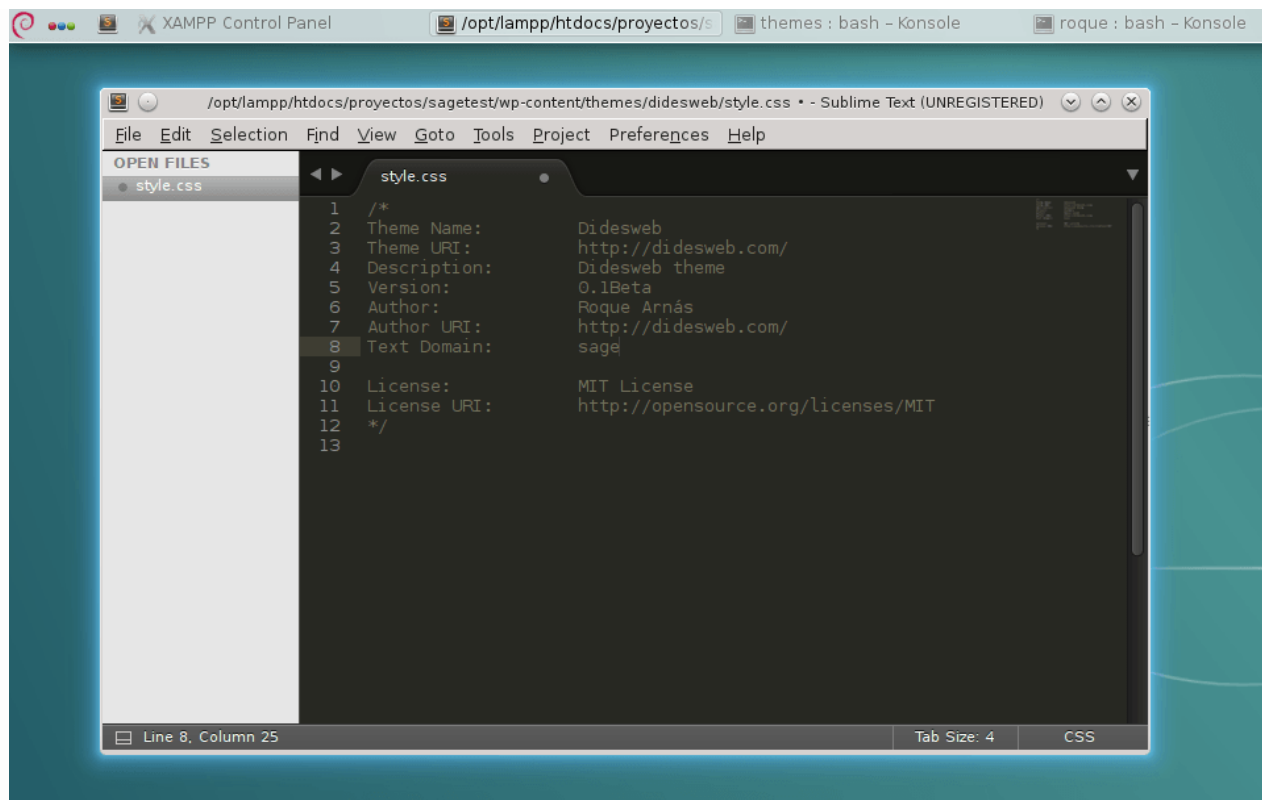
Sustituimos NOMBRE_DEL_TEMA por el nombre que queramos darle a nuestro tema, automaticamente se descargarán todos los archivos necesarios para el tema Sage.



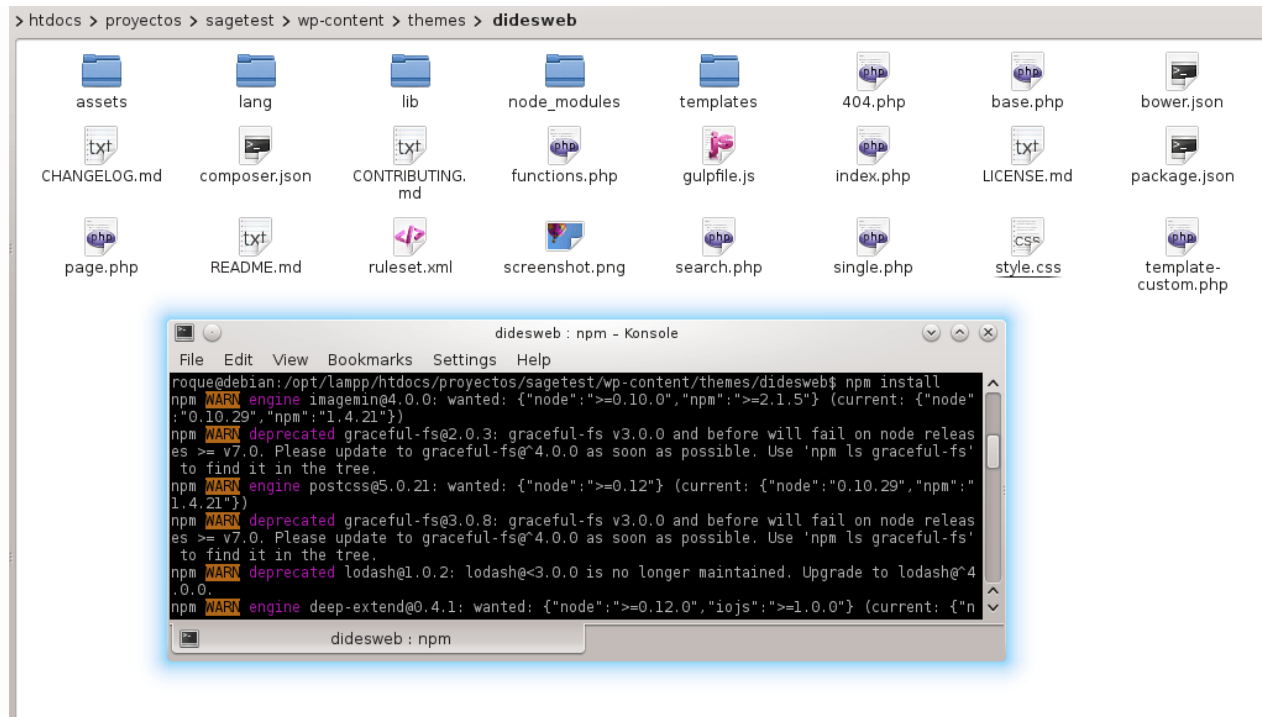


Podemos comprobar que el tema está instalado desde el panel de administración en Apariencia > Temas e instalar “Sage Starter Theme”, el tema de nuestro WordPress cambiará aunque todavía no mostrará un aspecto muy agradable.

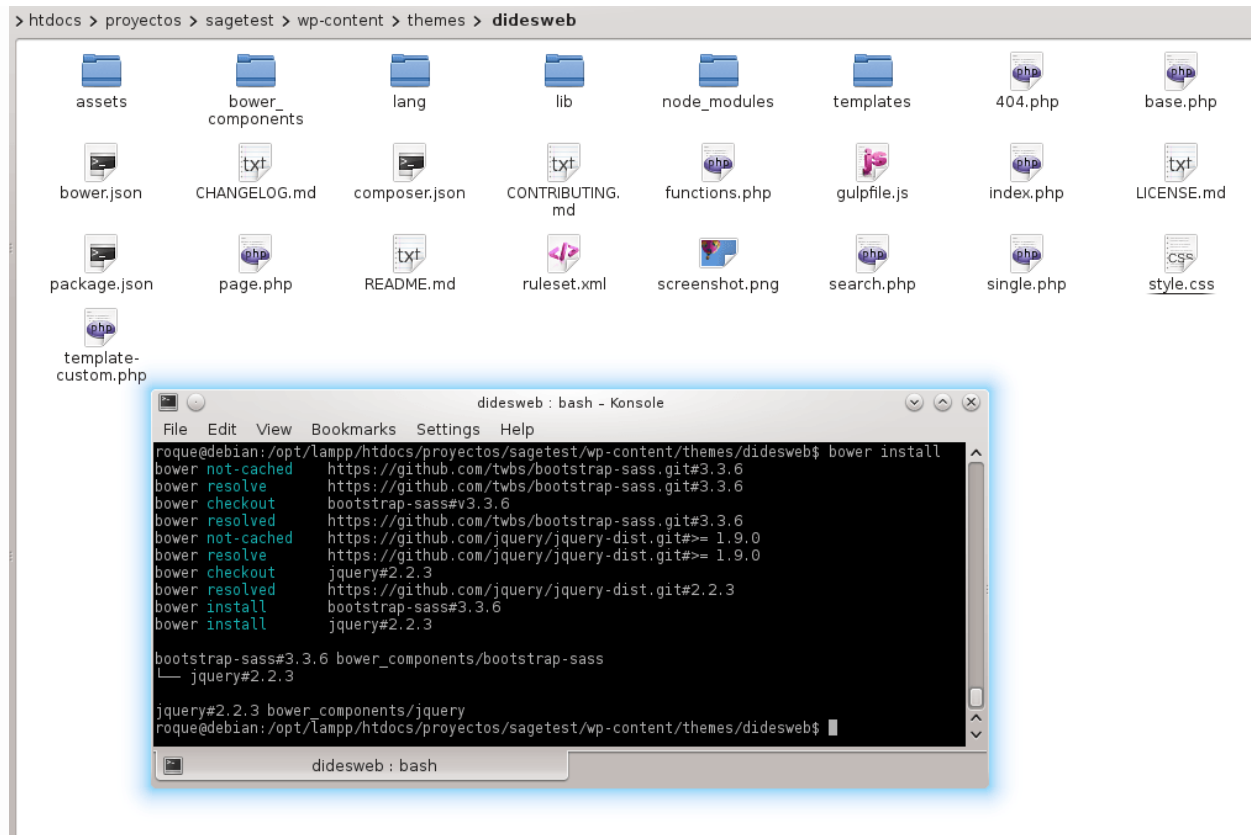
El nombre y los datos del tema podemos modificarlos en el archivo style.css que se encuentra en el directorio raíz del tema.



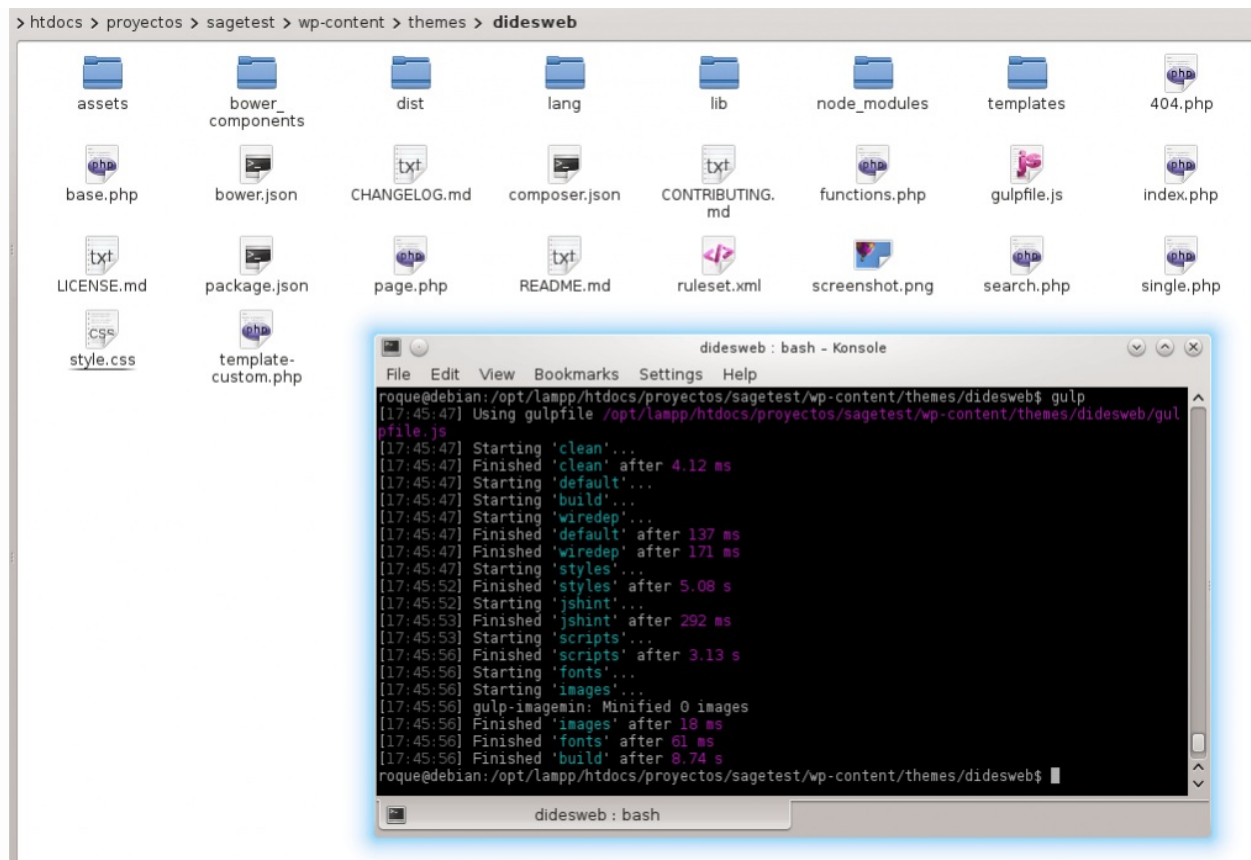
A continuación utilizaremos npm para instalar las dependencias necesarias para el tema, desde la terminal dentro de la carpeta del tema ejecutaremos `npm install`, se creará la carpeta `node_modules`.



A continuación ejecutamos **bower install**, se creará la carpeta *bower_components*.



Por ultimo ejecutamos gulp, se creará la carpeta dist.



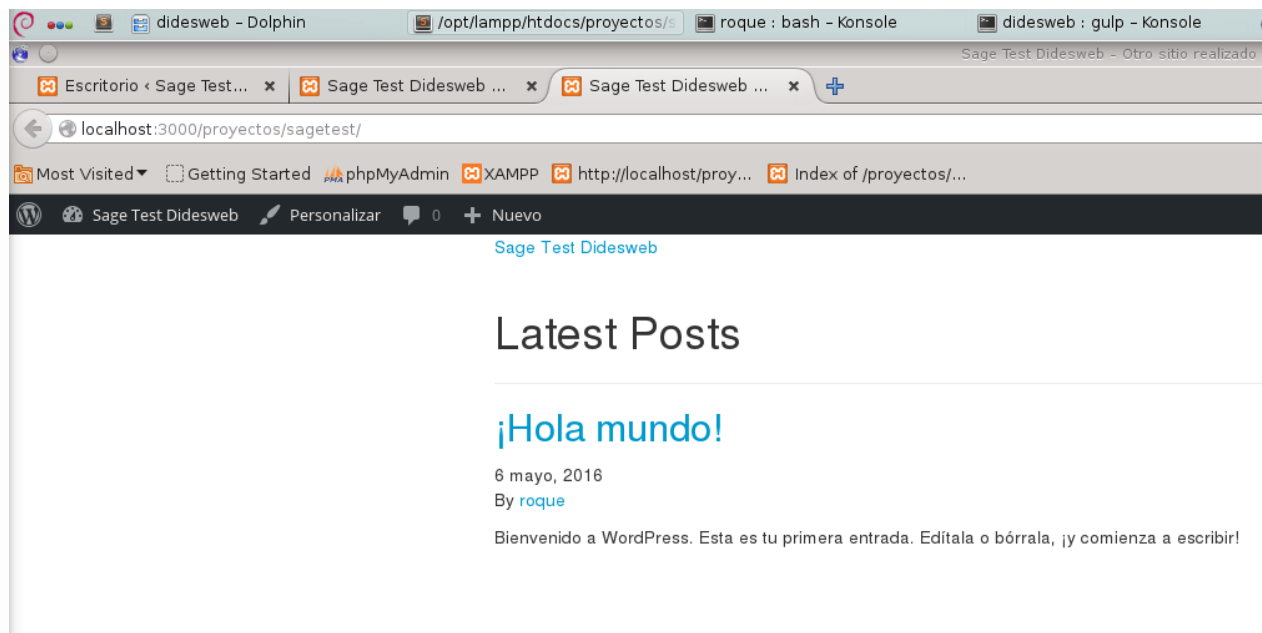
A continuación vamos a la carpeta assets y buscamos el archivo manifest.json, buscamos en la línea 25 *devUrl* en la que el valor es *example.dev*, cambiamos ese valor por la ruta del proyecto en el que estamos trabajando.


```
1  
2    },  
3    "config": {  
4        "devUrl": "http://localhost/proyectos/sagetest/"  
5    }  
6 }  
7
```

Si no conoces el formato de archivo JSON, puedes ver un tutorial en el siguiente enlace: [Tutorial de JSON](#)

Volvemos a la terminal y ejecutamos *gulp watch*, al hacer cualquier cambio en el css, se abrirá la ventana del navegador automáticamente, si no es así, podemos escribir la ruta directamente en el navegador escribiendo *localhost:3000/nombre_ruta/*.

Como mencioné anteriormente, *Sage* utiliza *SASS* para agilizar el proceso de maquetación, veamos una captura de pantalla al cambiar el color de los enlaces.



Si no conoces SASS, puedes ver un tutorial en el siguiente enlace: [Tutorial de SASS](#)

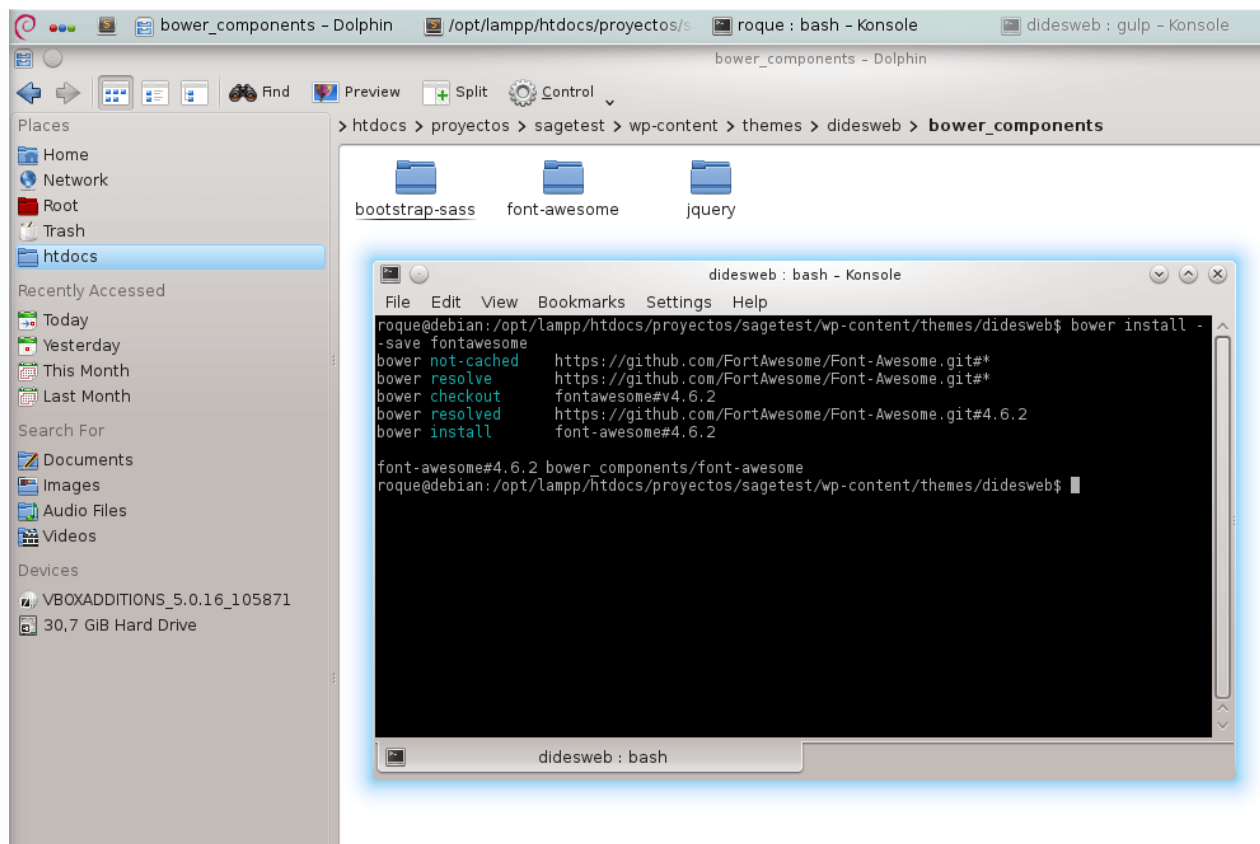
Instalar packages con Bower

Para instalar packages utilizaremos Bower, si no conoces Bower puedes ver un tutorial en esta web en el siguiente enlace: [Tutorial de Bower](#), vamos a ver un ejemplo de como instalar Font Awesome.

Abrimos la terminal como siempre dentro de la carpeta del tema o desplazandonos hasta ella con la terminal y ejecutamos:

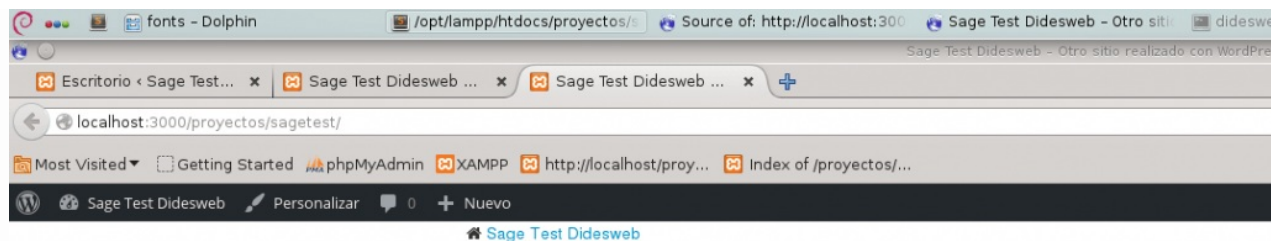
bower install --save fontawesome

A continuación volvemos a hacer gulp y comprobamos que se ha instalado correctamente en la carpeta bower_components

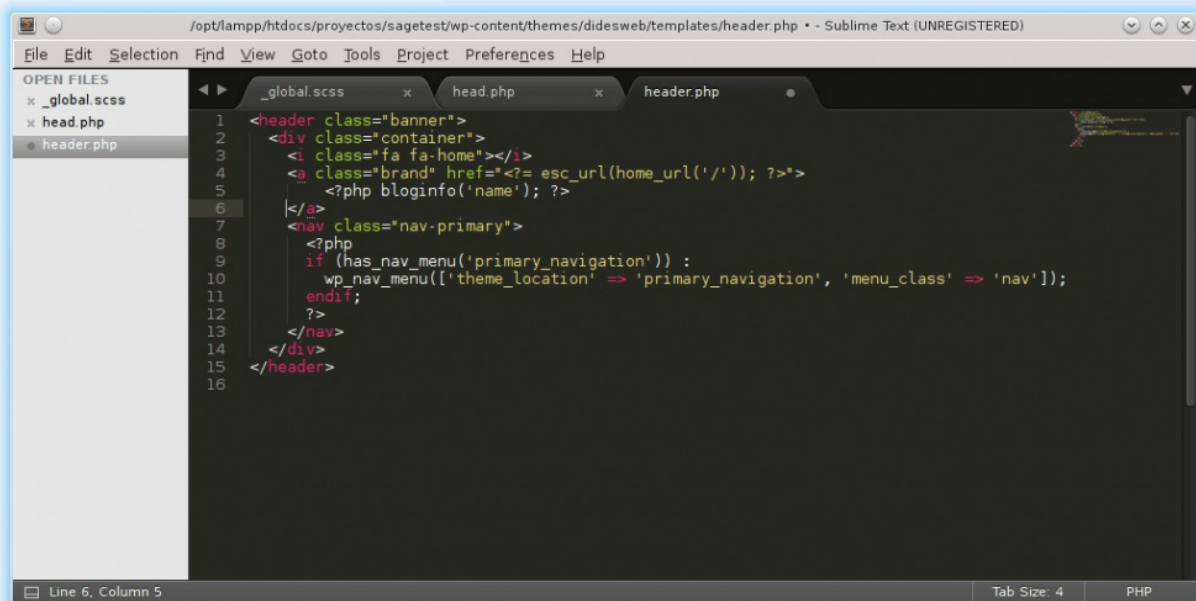


Algunas versiones no muestran las fuentes correctamente, si experimentais problemas podeis mover las fuentes directamente a la carpeta fonts

A continuaci3n podemos insertar una fuente:



Latest Posts



Eliminar CSS, JS y otros componentes

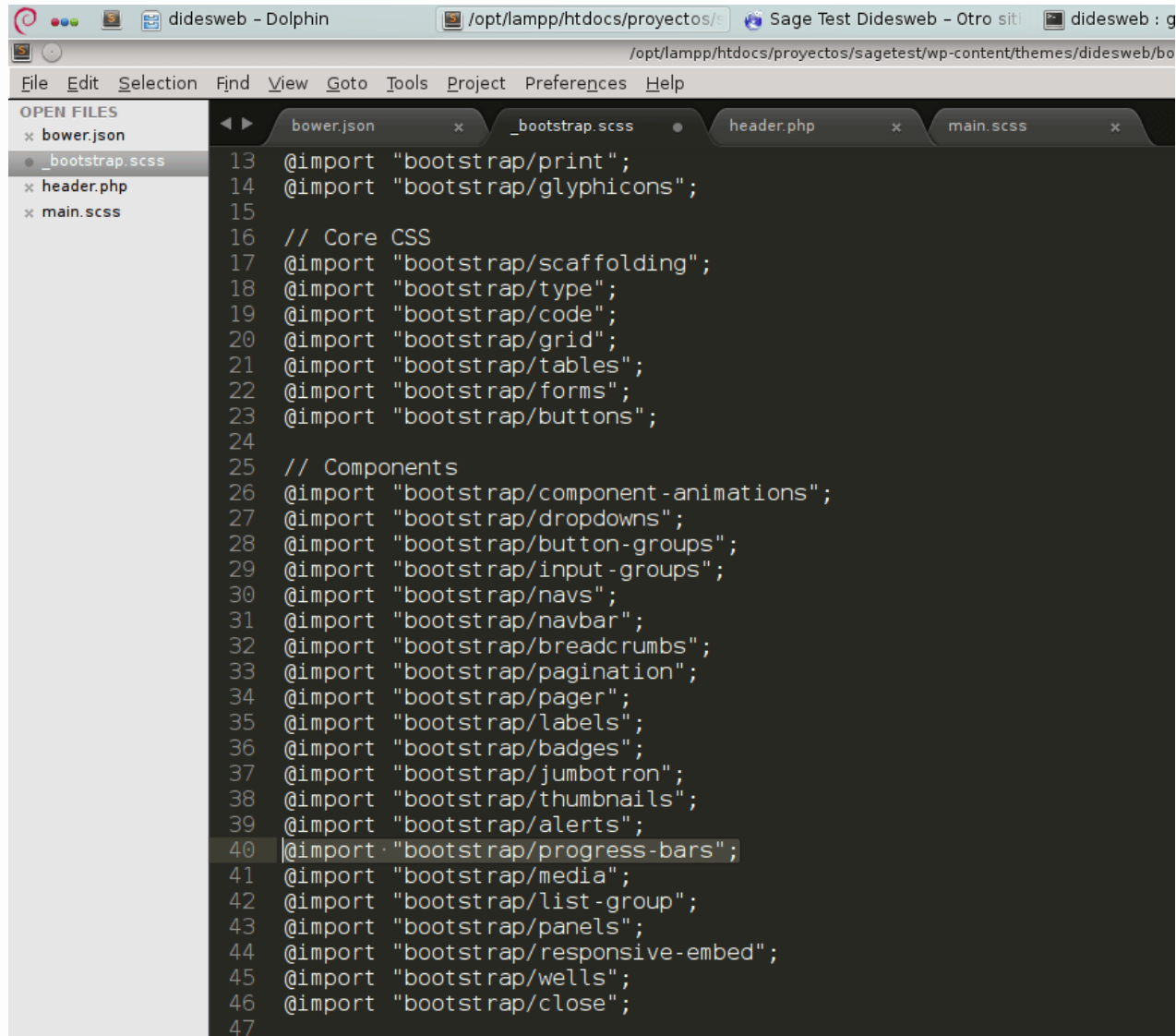
Por defecto, Sage tiene pre-instalado Bootstrap, no en vano, es el framework más utilizado actualmente, pero puede que no necesitemos todos los componentes que trae consigo, así que lo mejor es eliminar todo aquello que no vayamos a utilizar y crea un peso innecesario al cargar nuestro sitio.

Para probar que todo funciona correctamente, podemos insertar código como una barra de progreso para comprobar posteriormente que se han eliminado los estilos:

Abrimos el archivo `_bootstrap.scss` que se encuentra en : `bower_components > bootstrap-sass >`

assets > stylesheets, las rutas pueden variar según la versión de Sage, a continuación eliminamos la línea siguiente:

@import "bootstrap/progress-bars";

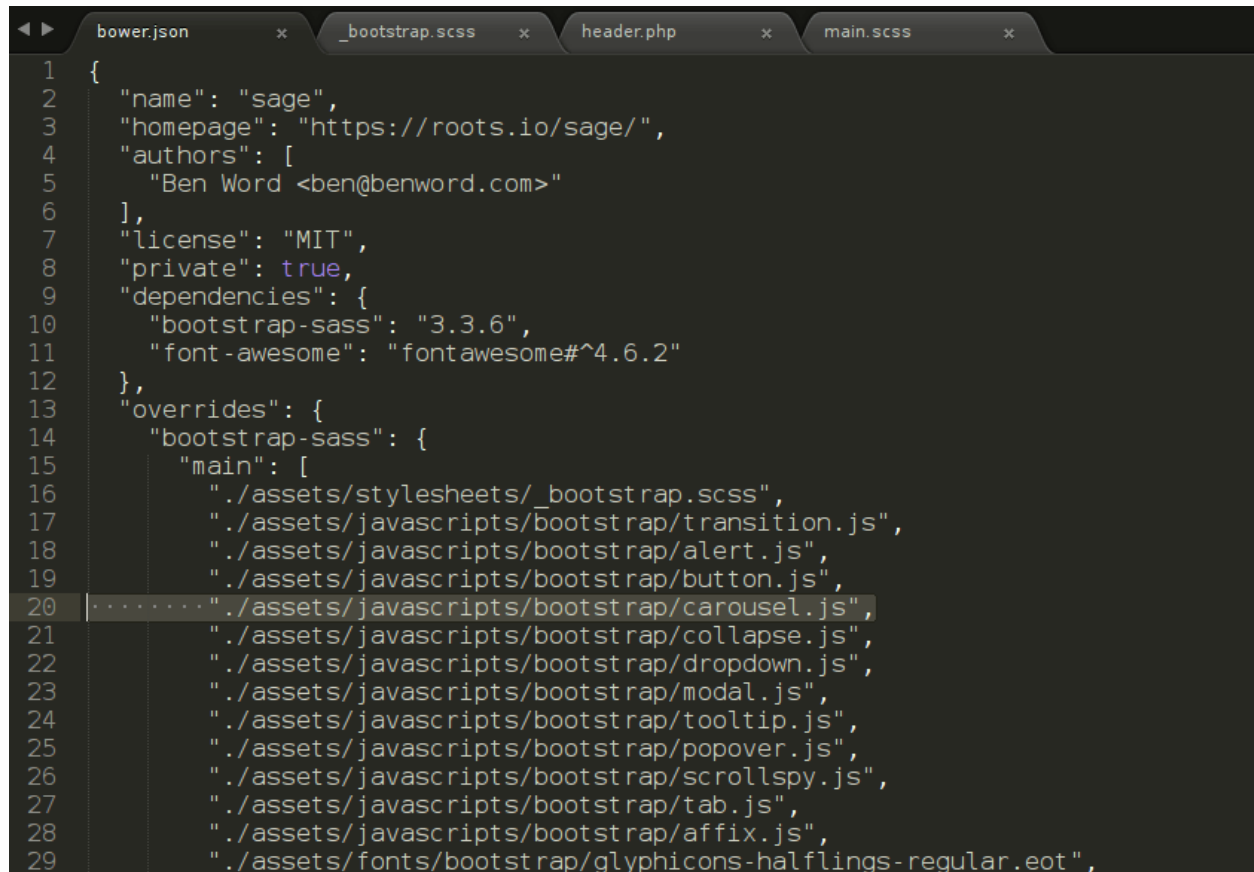


The screenshot shows a web browser window with the address bar displaying the path `/opt/lampp/htdocs/proyectos/sagetest/wp-content/themes/didesweb/bo`. The browser's menu bar includes `File`, `Edit`, `Selection`, `Find`, `View`, `Goto`, `Tools`, `Project`, `Preferences`, and `Help`. The `OPEN FILES` panel on the left lists `bower.json`, `_bootstrap.scss` (selected), `header.php`, and `main.scss`. The main editor area displays the content of `_bootstrap.scss`, which includes a series of `@import` statements for Bootstrap components. Line 40, `@import "bootstrap/progress-bars";`, is highlighted. The code is as follows:

```
13 @import "bootstrap/print";
14 @import "bootstrap/glyphicons";
15
16 // Core CSS
17 @import "bootstrap/scaffolding";
18 @import "bootstrap/type";
19 @import "bootstrap/code";
20 @import "bootstrap/grid";
21 @import "bootstrap/tables";
22 @import "bootstrap/forms";
23 @import "bootstrap/buttons";
24
25 // Components
26 @import "bootstrap/component-animations";
27 @import "bootstrap/dropdowns";
28 @import "bootstrap/button-groups";
29 @import "bootstrap/input-groups";
30 @import "bootstrap/navs";
31 @import "bootstrap/navbar";
32 @import "bootstrap/breadcrumbs";
33 @import "bootstrap/pagination";
34 @import "bootstrap/pager";
35 @import "bootstrap/labels";
36 @import "bootstrap/badges";
37 @import "bootstrap/jumbotron";
38 @import "bootstrap/thumbnails";
39 @import "bootstrap/alerts";
40 @import "bootstrap/progress-bars";
41 @import "bootstrap/media";
42 @import "bootstrap/list-group";
43 @import "bootstrap/panels";
44 @import "bootstrap/responsive-embed";
45 @import "bootstrap/wells";
46 @import "bootstrap/close";
47
```

Volvemos a la terminal y ejecutamos gulp styles, comprobaremos que los estilos han desaparecido

Para eliminar componentes JS lo hacemos desde el archivo bower.json que está en la raíz del tema, eliminamos aquellos que no vayamos a utilizar y ejecutamos gulp scripts.



```
1 {
2   "name": "sage",
3   "homepage": "https://roots.io/sage/",
4   "authors": [
5     "Ben Word <ben@benword.com>"
6   ],
7   "license": "MIT",
8   "private": true,
9   "dependencies": {
10     "bootstrap-sass": "3.3.6",
11     "font-awesome": "fontawesome#^4.6.2"
12   },
13   "overrides": {
14     "bootstrap-sass": {
15       "main": [
16         "./assets/stylesheets/_bootstrap.scss",
17         "./assets/javascripts/bootstrap/transition.js",
18         "./assets/javascripts/bootstrap/alert.js",
19         "./assets/javascripts/bootstrap/button.js",
20         ..... "./assets/javascripts/bootstrap/carousel.js",
21         "./assets/javascripts/bootstrap/collapse.js",
22         "./assets/javascripts/bootstrap/dropdown.js",
23         "./assets/javascripts/bootstrap/modal.js",
24         "./assets/javascripts/bootstrap/tooltip.js",
25         "./assets/javascripts/bootstrap/popover.js",
26         "./assets/javascripts/bootstrap/scrollspy.js",
27         "./assets/javascripts/bootstrap/tab.js",
28         "./assets/javascripts/bootstrap/affix.js",
29         "./assets/fonts/bootstrap/glyphicons-halflings-regular.eot",
```

Jerarquía de archivos y carpetas

En la carpeta templates del tema podemos encontrar los archivos principales del tema, como el head.php, footer.php, content.php, content-single.php y los demás archivos que utilizamos

habitualmente al crear temas personalizados.

En la carpeta raíz del tema encontramos el archivo 404.php, functions.php, search.php y uno de los más importantes, el archivo base.php

El archivo base.php tiene como objetivo el principio DRY, de forma que nos ahorramos tener que estar escribiendo código repetitivo en todos los archivos del tema, que por otra parte, tampoco tendremos que revisar uno por uno cuando necesitemos hacer cambios, se carga de modo automático en todas la páginas del tema.

La carpeta assets contiene los archivos css, js, fuentes e imágenes para ser compilados en la carpeta dist.

Para dar por terminado el desarrollo del tema y lanzar el proyecto a producción, ejecutamos en la terminal:

gulp –production

En proximos articulos veremos mas acerca de Sage, como crear páginas con diferente estructura y como modificar aspectos del tema, hasta aquí ya podemos comenzar a crear nuestros propios proyectos con Sage, la curva de aprendizaje no es muy extensa si conocemos Gulp, Bower y Sass y ya hemos creado temas para WordPress.

Gracias por pasar por mi web, nos vemos pronto

COMPARTE ESTE ARTÍCULO

[Twitter](#)[Google +](#)[Facebook](#)

VOLVER ARRIBA

[TWITTER](#)

[FACEBOOK](#)

[GOOGLE+](#)

[LINKEDIN](#)

[GITHUB](#)

[AVISO LEGAL](#)

[POLITICA DE PRIVACIDAD](#)

[CONTACTO](#)

[COOKIES](#)

© 2017 Didesweb Todos los derechos reservados.