

GESTIÓN DE LA GENERACIÓN ESTÁTICA DEL SITIO WEB G1

ID OPERA: 28

ALUMNOS:

CARRASCO MÁRQUEZ, ANTONIO

DÍAZ LLACER, SIXTO

HIGUERAS GALVÁN, FRANCISCO JAVIER

TIRADO HERNÁNDEZ, ÓSCAR

TORRIJOS CAMPANO, SERGIO

ENLACE REPOSITORIO: <https://github.com/Proyecto-EGC-G1/GeneracionEstatica-EGC-G1>

ENLACE DE DESPLIEGE: <https://g1wordpress.egc.duckdns.org/wp-admin/admin.php?page=html-generation-content-settings>

ENLACE RAMA DE ANTONIO: <https://github.com/Proyecto-EGC-G1/GeneracionEstatica-EGC-G1/tree/antcarmar2>

ENLACE RAMA DE SIXTO: <https://github.com/Proyecto-EGC-G1/GeneracionEstatica-EGC-G1/tree/sixdialla>

ENLACE RAMA DE FCO. JAVIER: <https://github.com/Proyecto-EGC-G1/GeneracionEstatica-EGC-G1/tree/frahigga1>

ENLACE RAMA DE ÓSCAR: <https://github.com/Proyecto-EGC-G1/GeneracionEstatica-EGC-G1/tree/osctirher>

ENLACE RAMA DE SERGIO: <https://github.com/Proyecto-EGC-G1/GeneracionEstatica-EGC-G1/tree/sertorcam>

ENLACE RAMA DE DOCUMENTACIÓN: <https://github.com/Proyecto-EGC-G1/GeneracionEstatica-EGC-G1/tree/master/docs/Documentaci%C3%B3n>

Índice

Tabla de Imágenes	1
Resumen.....	2
Introducción y contexto	2
Descripción del problema	3
Planificación del proyecto	4
Entorno de desarrollo	6
Gestión del código fuente	9
Gestión de la construcción e integración continua.....	10
Gestión de liberaciones, despliegue y entregas.....	10
Mapa de herramientas.....	11
Ejercicio de propuesta de cambio	12
Conclusiones y trabajo futuro	12

Tabla de Imágenes

1Vista del administrador para la generación estática.....	3
2Lista de incidencias.....	7
3Incidencia relacionada con commits	8
4Ejemplo de commit	10

Resumen

El equipo se encarga de la gestión de la generación estática de la web, esto es, una vez finalizado el uso del portal tener la oportunidad de poder seguir consultando la web así como el programa que se utilizó en el congreso el resultados que tuvieron las votaciones, etc. Para evitar un mantenimiento y posibles brechas de seguridad por los avances de las tecnologías se recomienda realizar una generación estática del sitio web evitando los problemas anteriormente mencionados. Esta generación estática consiste en obtener solo los archivos HTML, CSS y JavaScript de la web.

Nuestro equipo lo que pretende es que el administrador pasado un tiempo, pueda producir estos archivos y subirlos al servidor para que la web pueda seguir siendo consultable.

La propuesta del equipo para conseguir la generación estática es la creación de un plugin que lo permita. Esta plugin realiza un uso de la herramienta Htttrack que realiza esta generación estática de forma automática. Es buena idea integrarlo en un plugin debido que el portal está desarrollado sobre WordPress que se basa en la utilización de plugins, por lo tanto al desarrollarlo sobre un plugin la forma de integrarlo en el portal es mucho más sencilla.

Introducción y contexto

El equipo se encargara de la gestión de la generación estática de la web, esto es, producir los elementos estáticos de la web (archivos HTML, CSS y JavaScript). Esto evita, en un sitio web, el mantenimiento de la web y brechas de seguridad una vez haya terminado el uso de la web y esta pueda seguir siendo consultada.

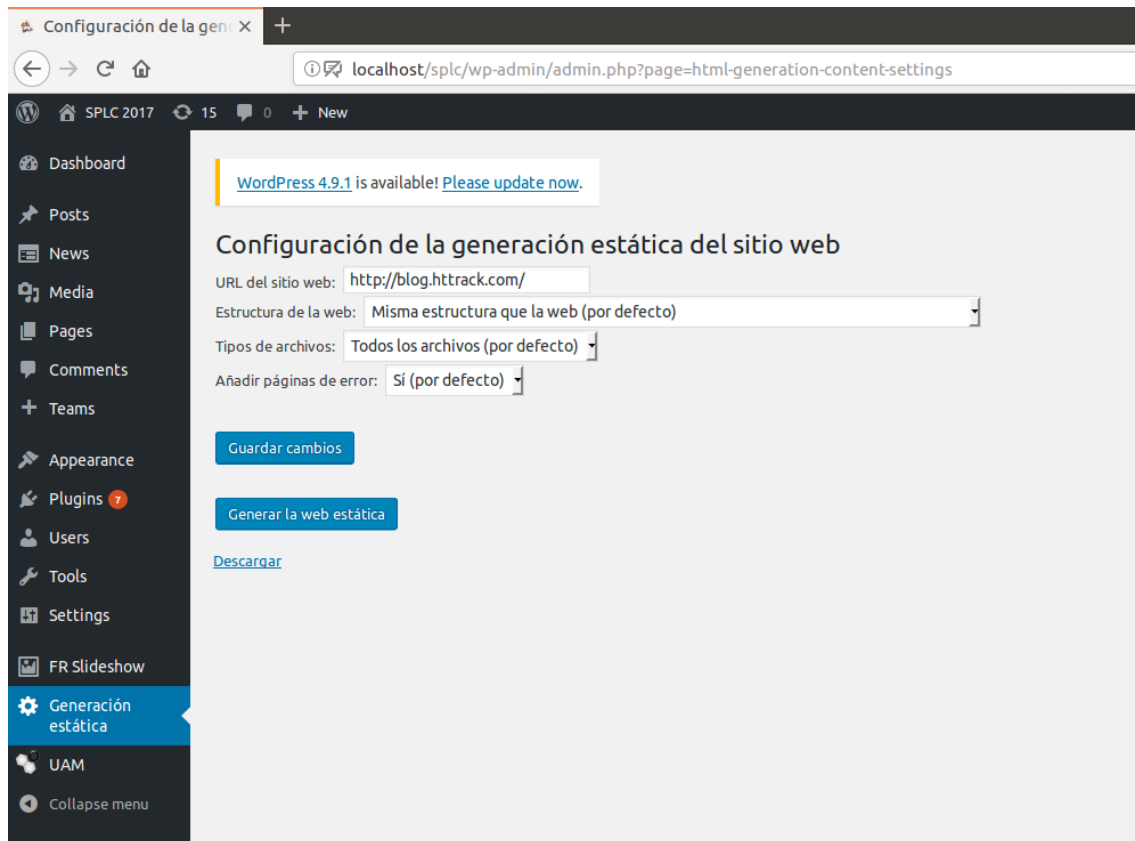
Una vez implementada esta solución el administrador de la web debe ser capaz de producir un archivo zip para subirlo a un servidor y que la web pueda volver a estar online.

Nuestro equipo se encargara del desarrollo de un plugin capaz de realizar la generación estática. La idea principal es integrar en la herramienta Htttrack en un plugin de WordPress. Esta herramienta proporciona la posibilidad de producir la generación estática de la web incluyendo algunas opciones útiles, como separar los archivos HTML de las imágenes.

En un principio propusimos utilizar el plugin existente SHOP (Static HTML Output Plugin), pero debido a que ese plugin ya realizaba la funcionalidad, no íbamos a poder incrementar la funcionalidad por lo que decidimos realizar un plugin desde cero. Este plugin se llama "Generación Estática" y está integrado con la herramienta Htttrack. Esta herramienta realiza la generación estática de la web pasándole como parámetro la url, pudiendo elegir una serie de opciones opcionales.

Descripción del problema

La funcionalidad del plugin creado consiste en que al administrador cuando desee producir la web estática se vaya al plugin de “Generación estática” y le saldría la siguiente vista



1 Vista del administrador para la generación estática

Entonces el administrador elige la opción que prefiera, pudiendo elegir en los siguientes campos:

-Estructura de la web:

- Misma estructura que la web (por defecto)
- HTML en web/, resto de archivos en web/images
- HTML en web/, resto de archivos en web/xxx, donde xxx es la extensión del archivo

-Tipos de archivos:

- Todos los archivos (por defecto)
- Solo archivos HTML
- Solo archivos no HTML

-Añadir páginas de error:

- Sí (por defecto)
- No

Una vez elegidas las opciones le das a “Guardar cambios” y posteriormente “Generar la web estática” y pasado un tiempo razonable pulsando “Descargar” ya obtienes el zip de la web generado.

El administrador posteriormente subirá el zip al servidor para que continúe funcionando la web.

Lo descrito anteriormente sería como debe usar el administrador el plugin, a continuación describiremos cómo funciona el plugin a nivel de código.

Para la creación del plugin se crearon tres archivos php:

- Generación-estática.php: es el archivo principal del plugin, que requiere de los otros dos archivos.
- View.php: en ella se encuentra el código del formulario que se recoge.
- Functions.php: en ella se encuentran las funciones que hacen posible el funcionamiento del plugin.

La función principal lo que implementa es que según las opciones recogidas por el formulario realiza una línea de comando llamando al Httrack y a la opción marcada para producir ese modo de generación estática. El Httrack ya produce el zip directamente según la opción marcada.

En cuanto a la integración con otros subsistemas, nuestro proyecto no se ve afectado ya que la realización del plugin no depende de la base de datos ni los campos de las tablas que existan en ella. Nuestro proyecto tampoco se ve afectado por el cambio de ningún subsistema. Todo esto es debido a que con solo la URL ya se produce la web estática de forma automática. Por lo que nuestro subsistema es independiente a los demás.

Para ser posible el uso de nuestro plugin se deberá añadir la carpeta del plugin en la carpeta de plugins de WordPress y activarlo. También deberán instalar Httrack, este proceso se recoge en el documento [Guía instalación del plugin](#).

Finalmente decir que el aumento de funcionalidad que hemos realizado es la creación de un nuevo plugin y la integración con Httrack.

Planificación del proyecto

Tanto a la planificación del proyecto como el reparto de tareas están descritos en el “Diario de grupo” por lo que esta sección mencionaremos las decisiones y tareas más importantes tomadas.

Las decisiones más importantes que hemos tomado han sido:

- Creación de los documentos donde se definían los procesos a seguir para la gestión de la evolución (gestión de incidencias, código fuente, equipo,...) ya que en estos documentos se basa realizar una buena gestión de la configuración.
- Realizar una integración con Httrack, ya que no habíamos encontrado nada que realizara la generación estática hasta que el profesor David Benavides nos comentó lo de la herramienta.

-Realizar un plugin desde cero con la integración de Httrack debido a que un principio íbamos a utilizar un plugin ya existente e integrarle Httrack, pero nos dimos cuenta que esto no era una buena idea y que resultaría más sencillo realizar un plugin desde cero que modificar un plugin ya existente.

-Añadir más opciones a la hora de la generación estática, ya que un principio solo tenía una opción decidimos añadir diferentes opciones que nos proporcionabas el Httrack que pueden serle útil al administrador.

Por lo consiguiente las tareas más importantes serían:

Tarea	Asignada a
Definición del proceso para la gestión de incidencias	Tirado Hernández, Óscar
Definición del proceso para la gestión del código fuente	Higueras Galván, Fco. Javier
Determinar un protocolo para la gestión del equipo	Carrasco Márquez, Antonio
Repositorio del plugin: Static HTML Output Plugin (SHOP)	Carrasco Márquez, Antonio Díaz Llacer, Sixto
Guía de instalación del plugin	Díaz Llacer, Sixto
Realización del plugin con la funcionalidad mínima	Díaz Llacer, Sixto
Creación de los test	Higueras Galván, Fco. Javier
Automatización de los test con Travis CI	Higueras Galván, Fco. Javier
Realización del formulario con los parámetros de configuración	Díaz Llacer, Sixto Higueras Galván, Fco. Javier
Realización de animación de carga	Torrijos Campano, Sergio
Modificar la funcionalidad de Httrack con los nuevos parámetros de configuración	Díaz Llacer, Sixto
Realización del Docker file	Higueras Galván, Fco. Javier
Documento de integración continua	Carrasco Márquez, Antonio
Realización de los nuevos test	Carrasco Márquez, Antonio Tirado Hernández, Óscar

Entorno de desarrollo

El entorno de desarrollo para crear los archivos php del plugin y los test están realizado en Notepad++ 7.5.2 y para la instacion del plugin se debe seguir los pasos descritos en el documento [Guía instalación del plugin](#)

Para la gestión de incidencia se realizara lo siguientes pasos:

- Crear la incidencia detectada, escribiendo el nombre de forma clara y una descripción opcional
- Asignársela a un miembro o varios del equipo
- Etiquetar la incidencia con los valores correspondiente

Para la gestión de las incidencias, se usarán los *Issue* que GitHub permite gestionar en cada repositorio, haciéndose de la siguiente manera:

- Cada *Issue* representará una tarea o problema encontrado por algún miembro del grupo.
- Asignar la *Issue* a uno o varios miembros del grupo
- Cada *Issue* será situado en un proyecto interno al repositorio, siendo actualizado entre tres fases:
 - **TO DO:** cuando el *Issue* está sin empezar
 - **In progress:** cuando se esté trabajando sobre el *Issue*
 - **Done:** cuando el *Issue* está terminado y cerrado.
- Para cada avance o actualización de un *Issue*, se hará un comentario dentro de éste especificando el cambio hecho, indicando su motivo si procede.
- Una vez terminado un *Issue* se debe cerrar con un comentario de cierre y cerrándolo en sí mismo tras colocarlo en la columna Done de su proyecto correspondiente. Esto debe hacerse automáticamente si el *Issue* se cierra con un commit.
- Cada *Issue* será etiquetado en función de su temática, prioridad, estado y tipo.
 - Los tipos son:
 - **Change/Upgrade**
 - **Bug**
 - O en su defecto **New** por si no tienen nada que ver con el desarrollo, que sólo se incluirán en dichos casos.
 - Las temáticas son creadas por el equipo y crear las etiquetas, que están pueden ser:
 - **Documentation**
 - **Development**
 - **Test**
 - **Investigation**
 - Si fuesen necesaria alguna otra etiqueta se procedería a su creación.
 - Las prioridades son:
 - **Critical**
 - **High**
 - **Medium**
 - **Low**

- Los estados son:
 - **New**
 - **Accepted** (sólo para cambios)
 - **Started**
 - **Fixed**
 - **Verified**
 - **Duplicate** (cambios y bugs)
 - **Wontfix** (cambios y bugs)
- Las etiquetas de cada *Issue* deben ser actualizadas durante el avance del mismo, con su correspondiente actualización en el proyecto y su comentario pertinente.

Todo lo descrito está recogido en el documento de “Proceso de gestión de incidencia” que se encuentra en el repositorio.

A continuación veremos como el equipo ha seguido este proceso:

<input type="checkbox"/>		Revisión de las funciones por problemas con los test change/upgrade high started test		3
		#28 by sixdiaila was closed 37 minutes ago Milestone 4		
<input type="checkbox"/>		Realizar los cambios en documentation documentation medium new started		1
		#27 by Weothar was closed 4 hours ago Milestone 4		
<input type="checkbox"/>		Documento de como gestionar las versiones documentation medium new		
		#25 by Weothar was closed 4 hours ago Milestone 4		
<input type="checkbox"/>		Creación del diario del grupo documentation medium new started		1
		#24 by Weothar was closed a day ago Milestone 4		
<input type="checkbox"/>		Realización del Docker file development medium new		1
		#23 by Adelay95 was closed 2 days ago Milestone 4		
<input type="checkbox"/>		Modificar la funcionalidad de Httrack con los nuevos parámetros de configuración change/upgrade development medium started		
		#22 by sixdiaila was closed a day ago Milestone 4		
<input type="checkbox"/>		Realización de animación de carga development high new started		1
		#21 by SergioTorrijos was closed a day ago Milestone 4		
<input type="checkbox"/>		Realización del formulario con los parámetros de configuración change/upgrade development high started		1
		#20 by sixdiaila was closed a day ago Milestone 4		
<input type="checkbox"/>		Automatización de los test con Travis CI high new started test		1
		#19 by sixdiaila was closed 6 hours ago Milestone 4		
<input type="checkbox"/>		Creación de los test high new test		2
		#18 by sixdiaila was closed a day ago Milestone 4		
<input type="checkbox"/>		Realización del plugin con la funcionalidad mínima development high started		
		#17 by sixdiaila was closed 17 days ago Milestone 4		
<input type="checkbox"/>		Defensa del Milestone 3 Organize low new verified		
		#16 by SergioTorrijos was closed 18 days ago		
<input type="checkbox"/>		Redacción del documento de situación del Milestone 3 Organize documentation medium new		
		#15 by SergioTorrijos was closed 18 days ago		
<input type="checkbox"/>		Guía de instalación del plugin documentation high new		
		#14 by sixdiaila was closed 28 days ago		

2Lista de incidencias

Realización del formulario con los parámetros de configuración #20

[Edit](#)[New issue](#)

sixdialla opened this issue 18 days ago · 1 comment



sixdialla commented 18 days ago

Owner



Realizar un formulario en php para guardar los parámetros de configuración con los que el usuario podrá personalizar la forma de generar la web estática.



sixdialla added the **development** label 18 days ago



sixdialla added this to the **Milestone 4** milestone 18 days ago



sixdialla added this to To Do in **Organización de tareas** via **automation** 18 days ago



sixdialla moved this from To Do to In Progress in **Organización de tareas** 18 days ago



SergioTorrijos added **change/upgrade** **high** **started** labels 17 days ago



sixdialla added a commit that referenced this issue 17 days ago



Añadida la funcionalidad mínima del plugin ...

1466631



sixdialla added a commit that referenced this **issue** 2 days ago



feat: Campo URL añadido en la configuración ...

eec1988



sixdialla added a commit that referenced this issue 2 days ago



stylo: Se ha añadido un mensaje de confirmación ...

3f7900e



Adelay95 added a commit that referenced this issue a day ago



feat: Adding new components ...

ca58d42



Adelay95 moved this from In Progress to Done in **Organización de tareas** a day ago



Adelay95 commented a day ago

Member



Cerrado con el commit **ca58d42**



Adelay95 closed this a day ago



Weothar assigned sixdialla and Adelay95 a day ago



Write

Preview

AA B i



Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

[Reopen issue](#)[Comment](#)

Assignees



Adelay95

sixdialla

Labels



change/upgrade

development

high

started

Projects



Done in Organización de tareas

Milestone



Milestone 4

Notifications

Subscribe

You're not receiving notifications from this thread.

3 participants

Lock conversation

3Incidencia relacionada con commits

Gestión del código fuente

Los commits se realizan cada vez que se realiza un cambio asociado a alguna incidencia, ya puede ser incremento funcional o creación de documentos o modificación de ambos.

Para llevar a cabo la gestión del código fuente usaremos las herramientas Git y GitHub.

Estructura:

Tipo:

- feat: Una nueva característica
- fix: Se solucionó un bug
- docs: Se realizaron cambios en los documentos
- stylo: Cambios de estilo y visuales en la página
- refactor: Se realizaron cambios en el código
- test: Se realizaron pruebas en el commit.

Título: No debe contener más de 50 caracteres, debe iniciar con letra mayúscula y no acabar con signos de puntuación. Debe ser lo más objetivo posible y, preferiblemente, en inglés.

Cuerpo: Si el commit no es muy complejo no es necesario realizar el cuerpo. Debe detallarse principalmente el que y el porqué de la realización de dicho commit y no centrarse en el cómo se ha realizado.

Pie: Este campo es opcional y se debe utilizar para referenciar a otras incidencias.

Ejemplo de uso:

Fix: Change of JavaScript buttons Se ha realizado un cambio en los botones de JavaScript ya que redirigían a las vistas equivocadas.

Resuelve: #123

Todo lo descrito está recogido en el documento de “Proceso de gestión del código fuente” que se encuentra en el repositorio.

A continuación veremos un ejemplo de commit:

feat: Nuevos campos de configuración añadidos

Los nuevos campos de configuración añadidos al formulario han sido agregados a la función que ejecuta Htrack para que sean usados.

Closes #22

master 1.0

sixdialla committed a day ago 1 parent ca58d42 commit 98618aca7edff9ba92228d38ea787aa15d8d7507

Showing 2 changed files with 35 additions and 17 deletions.

Unified Split

View

```
7  generacion-estatica.php
@@ -9,10 +9,8 @@
9 9 License: GPL2
10 10 License URI: https://www.gnu.org/licenses/gpl-2.0.html
11 11 */
12 -
13 12 require_once 'library/functions.php';
14 13 require_once 'library/view.php';
15 -
16 14 function html_generation_plugin_menu(){
17 15
18 16     add_menu_page('Configuración de la generación estática del sitio web',
@@ -23,10 +21,5 @@ function html_generation_plugin_menu(){
23 21         'dashicons-admin-generic');
24 22     }
25 23     add_action('admin_menu','html_generation_plugin_menu');
26 -
27 -
28 -
29 24     add_action('admin_init','html_generation_content_settings');
30 -
31 -
32 25     ?> ?<
```

4Ejemplo de commit

Gestión de la construcción e integración continua

El proceso de integración seguido es que cada vez que se realice un commit se realiza una unas pruebas automáticas para comprobar el funcionamiento del plugin y si este es exitoso se despliegue.

Para la gestión de la integración continua usaremos el host Travis CI. Para su correcto funcionamiento se han de añadir los siguientes archivos:

- ♣ .travis.yml: Contiene la configuración de Travis CI.
- ♣ phpunit.xml: En este archivo se indica la ubicación de las pruebas de Travis.
- ♣ composer.json: Se encarga de mantener las versiones de php y phpunit actualizadas y además se muestra la ruta de las clases que van a participar en los tests.
- ♣ Carpeta “tests”: Carpeta donde se ubican los tests a realizar.

Todo lo descrito está recogido en el documento de “Proceso de gestión de la integración continua” que se encuentra en el repositorio.

Gestión de liberaciones, despliegue y entregas

Proceso de liberación del proyecto

El proyecto alcanzará su cierre cuando se cumplan todos los objetivos definidos del equipo de trabajo o se alcance la fecha límite de entrega. El objetivo del proyecto es lograr crear un plugin de Wordpress que permita, mediante un botón, generar (de cualquier vista de la página del portal) una vista estática. Cuando hablamos de generar una vista estática, es la

transformación de la vista a únicamente tener HTML, CSS y JavaScript. Esto admite desvincular la página de la base de datos y la lógica de negocio, lo que permite crear una vista completamente segura de la página. Éste cierre se verá reflejado con un “release” en nuestro repositorio.

Proceso definido para el despliegue

Una vez se haya marcado el cierre de nuestro proyecto, el proyecto se subirá a un “release” de nuestro repositorio usando GitHub Releases. Este archivo deberá ser descomprimido en la carpeta de plugins de WordPress y conceder permisos de escritura en la carpeta del plugin para que pueda descargar la página de su elección.

Proceso definido para las entregas

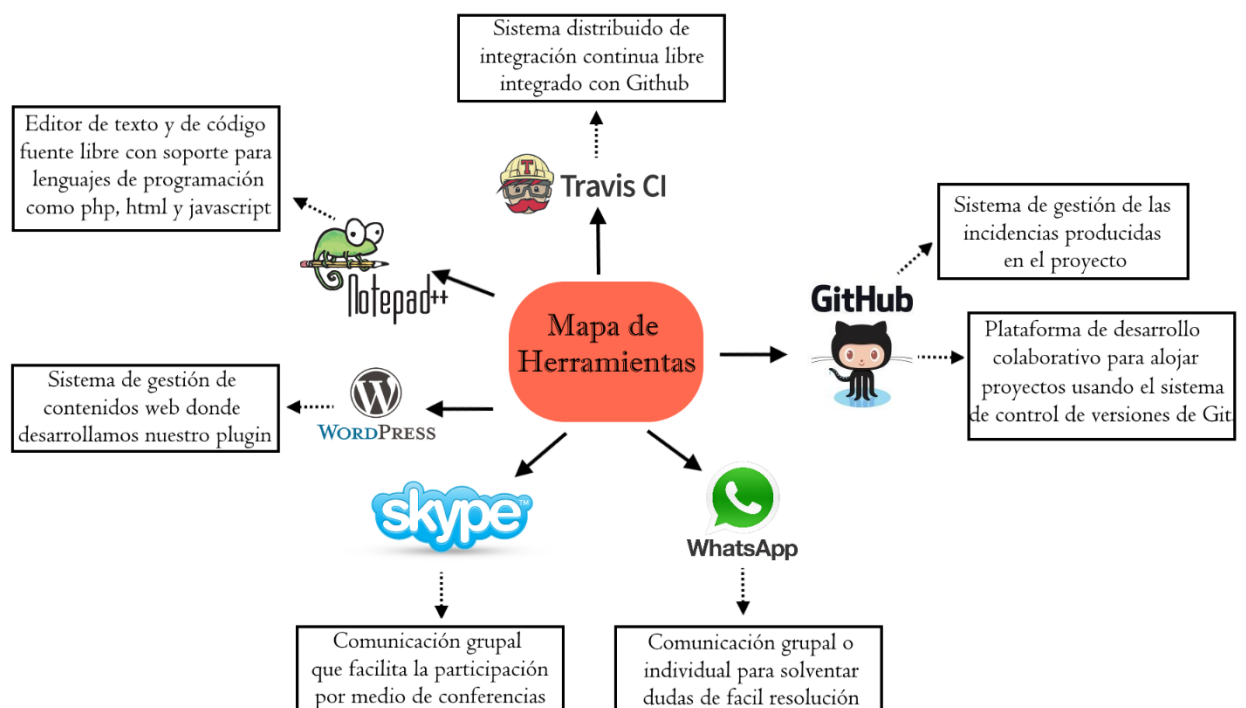
A la hora de realizar la entrega se concretará con el equipo de Integración una reunión telemática que la instalación del protocolo sea exitosa y para una última revisión del producto entregado. El ámbito de las incidencias que ofrece GitHub, seguirá activo para futuras dudas sobre nuestra aplicación, ofreciendo soporte.

Política de nombrado e identificación de los entregables

El entregable será nombrado con el nombre del trabajo, en nuestro caso generación-estática y la versión de nuestra aplicación, dado que es la primera versión de nuestro plugin, será la versión 1.0. “generación-estática-1.0.zip”.

Todo lo descrito está recogido en el documento de “Proceso de las liberaciones, despliegue y entregas” que se encuentra en el repositorio.

Mapa de herramientas



Ejercicio de propuesta de cambio

Nuestro objetivo es añadir una funcionalidad nueva al plugin de generación estática.

Primero, debemos crear una incidencia en GitHub cuyo título será: Incorporación de la funcionalidad XXX al plugin. En la descripción: Inclusión de la funcionalidad XXX al plugin para YYY. Las etiquetas asociadas a la incidencia serán Change/Update(Dado que es una mejora del plugin) New(porque es una nueva característica), Development(ya que es una tarea de desarrollo) y High(ya que la incidencia es urgente). Esta incidencia será asignada a Sixto y Francisco Javier para que se ocupen de la visualización y el desarrollo del método, respectivamente. Debemos añadir además el proyecto Organización de Tareas para poder usar un Tablero Kanban que nos permita tener una ligera idea del progreso del proyecto. Además, debemos asociar el Milestone donde se engloba.

Los desarrolladores asignados moverán la tarea a TODO en el tablero y comenzarán su trabajo. Sixto se encargará de incluir, a través del editor NotePad++, el nuevo campo al formulario del plugin y Francisco Javier, se encargará de incluir ese nuevo campo en el controlador. Una vez acabado, los asignados incluirán sus modificaciones a sus respectivas ramas realizando un commit y un push. Tras comprobar que los tests funcionan correctamente, cambiarán a la rama reléase y harán un merge de sus respectivas ramas y volverán a usar un commit y un push. El primero que termine su trabajo comentará en la issue que ha acabado y el otro una vez haya hecho un push en release, cambiará a la rama master y hará un merge a release con los cambios de ambos desarrolladores. Finalmente cerrará la incidencia con el último commit y se exportará el proyecto y se subirá a GitHub Release como una nueva versión del plugin.

Conclusiones y trabajo futuro

Al tratarse de un apartado nuevo no teníamos código heredado por lo que al principio tuvimos complicaciones a la hora de cómo empezar a trabajar sobre el desarrollo. Tras la reunión sobre el primer milestone salimos con las ideas más claras y supimos comenzar. El siguiente problema que surgió fue la instalación de WordPress y la base de datos. Tras solucionar el problema seguimos avanzando. Nos dimos cuenta que el tamaño de nuestro proyecto no era demasiado grande como podría ser los subsistemas relacionados con las votaciones.

Por lo que nos dedicamos más a realizar una buena gestión de configuración del código una vez desarrollado el plugin, el cual no llevo demasiado tiempo.

Aunque ha habido momentos en los que hemos estado un poco perdido, el producto final ha sido entregado al equipo de integración correctamente. Los únicos problemas que encontramos, es la dificultad de llegar a los índices de líneas de código debido al tamaño del proyecto.

Para mejorar el plugin en un futuro un ejemplo de una funcionalidad que no pudimos completar fue la de que mientras se está ejecutando la función del Httrack saliera una animación de carga hasta que este proceso se hubiera completado, esta animación serviría al administrador cuando se había completado el proceso de creación del código de la página estática. Otra posible mejora es la de la vista, en la cual se podría sintetizar los campos y ponerlos de forma más simplificada.