WordPress *y* PHP

"Codename *Servehappy*" y "Project *Tide*"

Hola 👋

Soy Javier Casares

Utilizo WordPress desde la versión 1.5 y soy el responsable de WPdanger, una herramienta que hace un análisis y detección de varios elementos sobre seguridad de sitios con WordPress.

Puedes encontrar más de mi y contacto en JavierCasares.com.

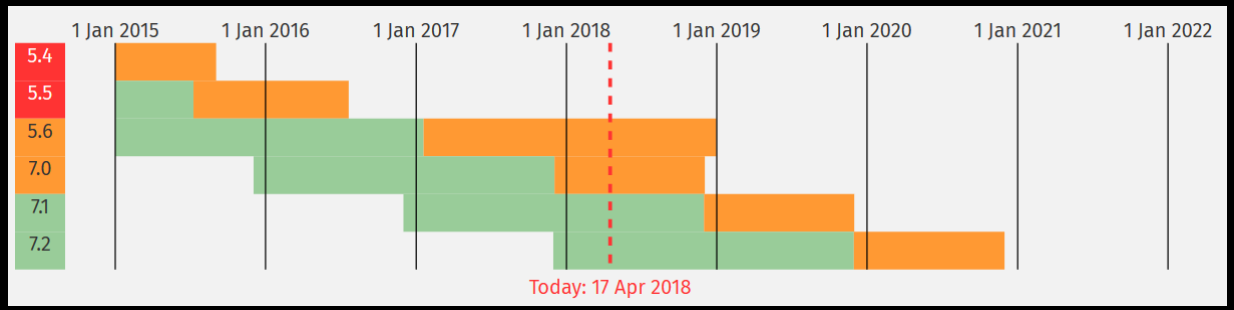
PHP is a popular general-purpose scripting language that is especially suited to web development.

Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.

Última actualización: 29 de marzo de 2018

5.6.35 / **7.0**.29 / **7.1**.16 / **7.2**.4

http://php.net/supported-versions.php



WordPress *es* PHP

¿Qué versión usas tú?

Si WordPress es PHP...
¿Ya tienes PHP en su última versión en tu servidor?

```
if( version_compare( PHP_VERSION, '5.6.0' ) <= 0 )
{
     exit( 'Goodbye World!' );
}
```

Si usas una versión anterior a la 5.6, lo más probable es que te acabe fallando todo en breve.

Si has de instalar algo, AL MENOS, que sea la 7.0, aunque mejor la 7.2.

Codename **Servehappy**

¿Qué es?

"Serverhappy" es un proyecto de WordPress que quiere ayudar a informar sobre las versiones mínimas de PHP aceptables.

Aunque al instalar WordPress tienes información de sus requerimientos, también existirá una página sobre "Upgrade PHP".

Antes de actualizar tu PHP, recuerda comprobar si tu sitio funcionará con el plugin PHP Compatibility Checker.

¿Qué es?

La idea es que haya un *widget* en el *dashboard* que comunique esta información.

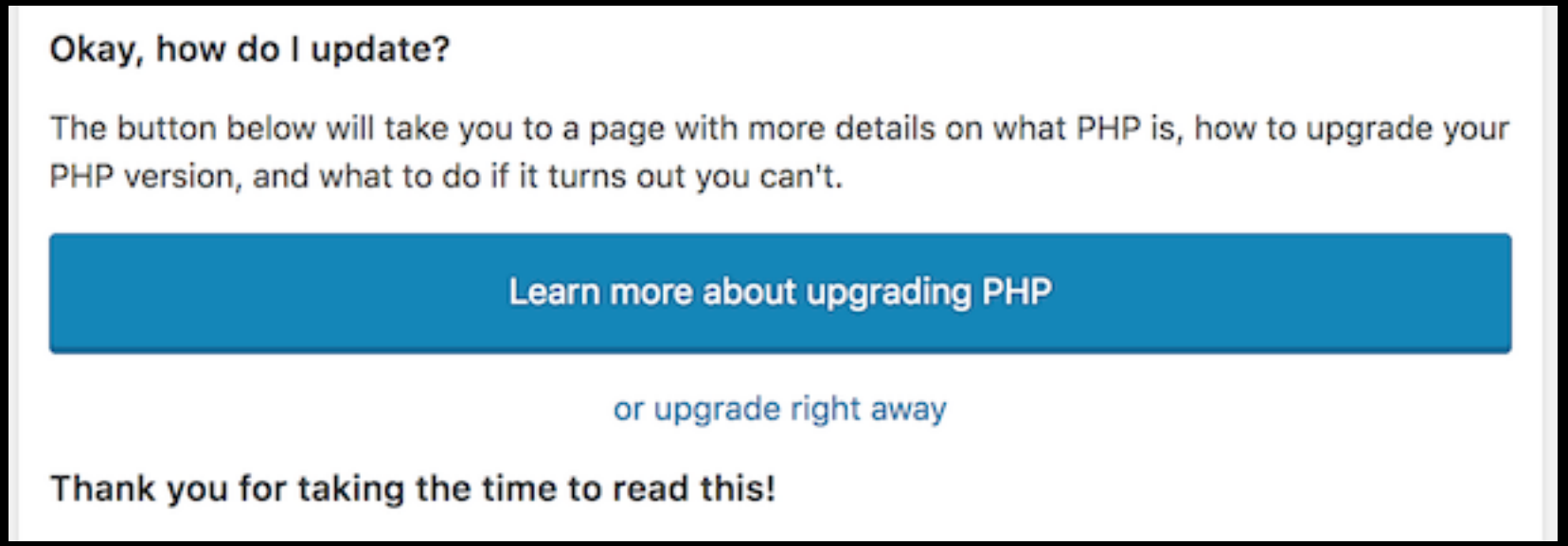
Queda por decidir quién podrá verlo, si se podrá ocultar y varias opciones.

Probablemente venga de serie en la versión 5.0.0, aunque no se descarta que aparezca previamente en la 4.9.6 o 4.9.7.

Si alguien está interesado en seguirlo, tiene acceso en el #core-php.

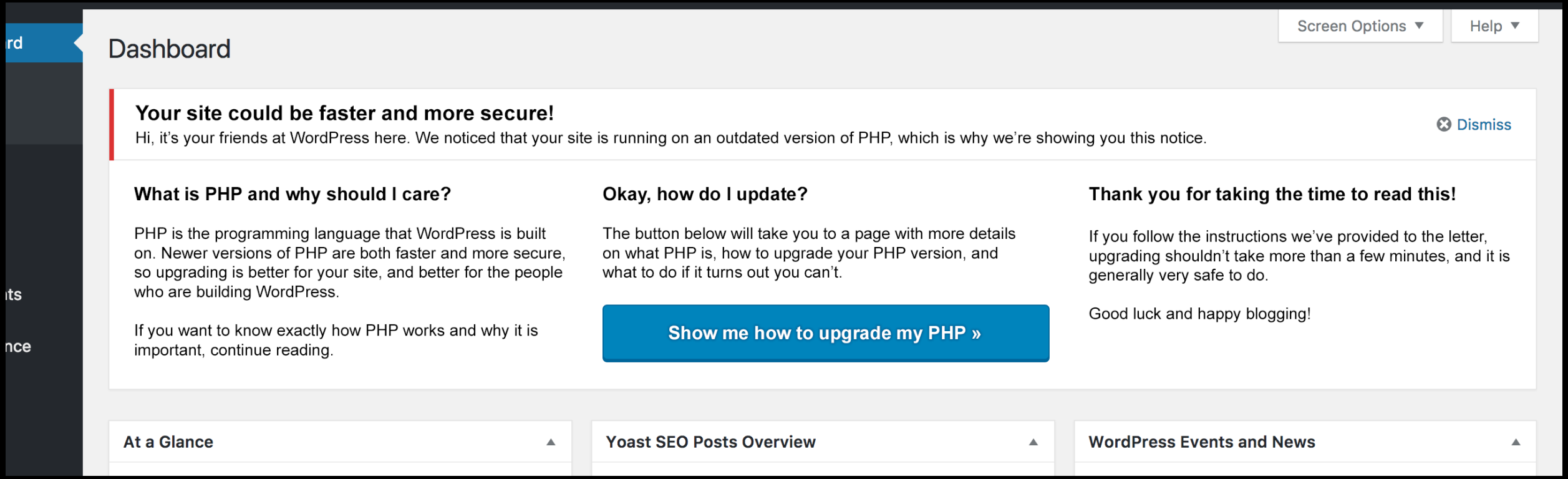
¿Cómo se vería?

Por ahora solo hay versiones piloto, pero la idea es algo tal que así:



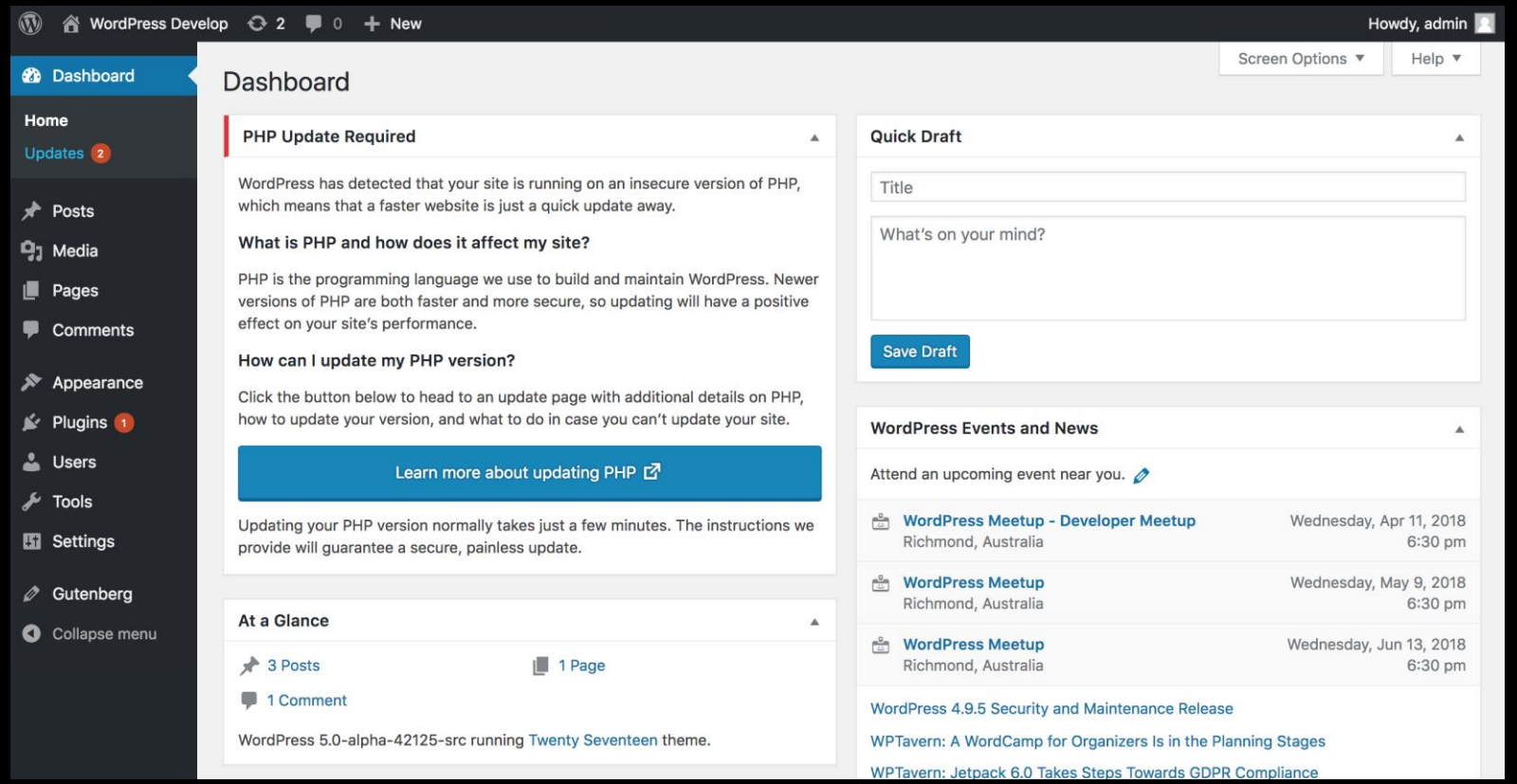
¿Cómo se vería?

O quizá un poco más agresivo, y sea así:



¿Cómo se vería?

Aunque las últimas versiones parecen que dejará esto:



¿En qué se basa?

Aunque también hay idas y venidas con la API (sobre todo debido a la GDPR), la idea es trabajar con esta API:

https://api.wordpress.org/core/serve-happy/1.0/?php_version=7.0

Test: 7.0


```
{
 "recommended_version": "7.2",
 "is_supported": true,
 "is_secure": true,
 "is_acceptable": true
}
```


Test: 5.1


```
{
 "recommended_version": "7.2",
 "is_supported": false,
 "is_secure": false,
 "is_acceptable": false
}
```

¿Dónde hay información?

Como (casi) todo lo de WordPress, en el TRAC:

#2996 - Add wordpress.org information page about upgrading PHP

#41191 - Create browse happy type notice for PHP versions

Y en el canal #core-php.

¿Para qué sirve todo lo anterior?

Si WordPress se ha planteado solicitar que mantengas al día tu versión de PHP no es por seguridad o rendimiento (que también), sino para poder dar soporte a los desarrolladores.

Todo comenzó cuando se podía informar de la versión requerida de PHP en los plugins...

```
=== Plugin Name ===
Contributors: (this should be a list of wordpress.org userid's)
Donate link: http://example.com/
Tags: comments, spam
Requires at least: 4.6
Tested up to: 4.8
Requires PHP: 5.6
Stable tag: 4.3
License: GPLv2 or later
License URI: https://www.gnu.org/licenses/gpl-2.0.html
```

¿Para qué sirve todo lo anterior?

En las últimas semanas se ha añadido un nuevo proyecto piloto que "fuerce" a los plugins a no activarse o desinstalarse si las versiones de PHP no coinciden [/mode agresivo on]

Proof of concept for 'Requires PHP' header and updating in WordPress…

https://github.com/afragen/requires-php

Si tu servidor tiene visible la cabecera de version de PHP (por seguridad no debería), en WPdanger puedes ver si tus plugins y sistema son compatibles.

Project **Tide**

¿Qué es?

*The mission of Tide is to **raise the code quality** of all the plugins and themes in the WordPress.org directories. Tide's name reflects this mission, derived from the proverb "a rising tide lifts all boats".*

+ Modo fácil:

Proyect Tide va a darle unas *estrellitas* al código de tu plugin / theme.

Si alguien está interesado en seguirlo, tiene acceso en el #tide.

Un poco de historia...

En diciembre de 2017 😱 se comienza a comentar la existencia de este proyecto, todavía interno.

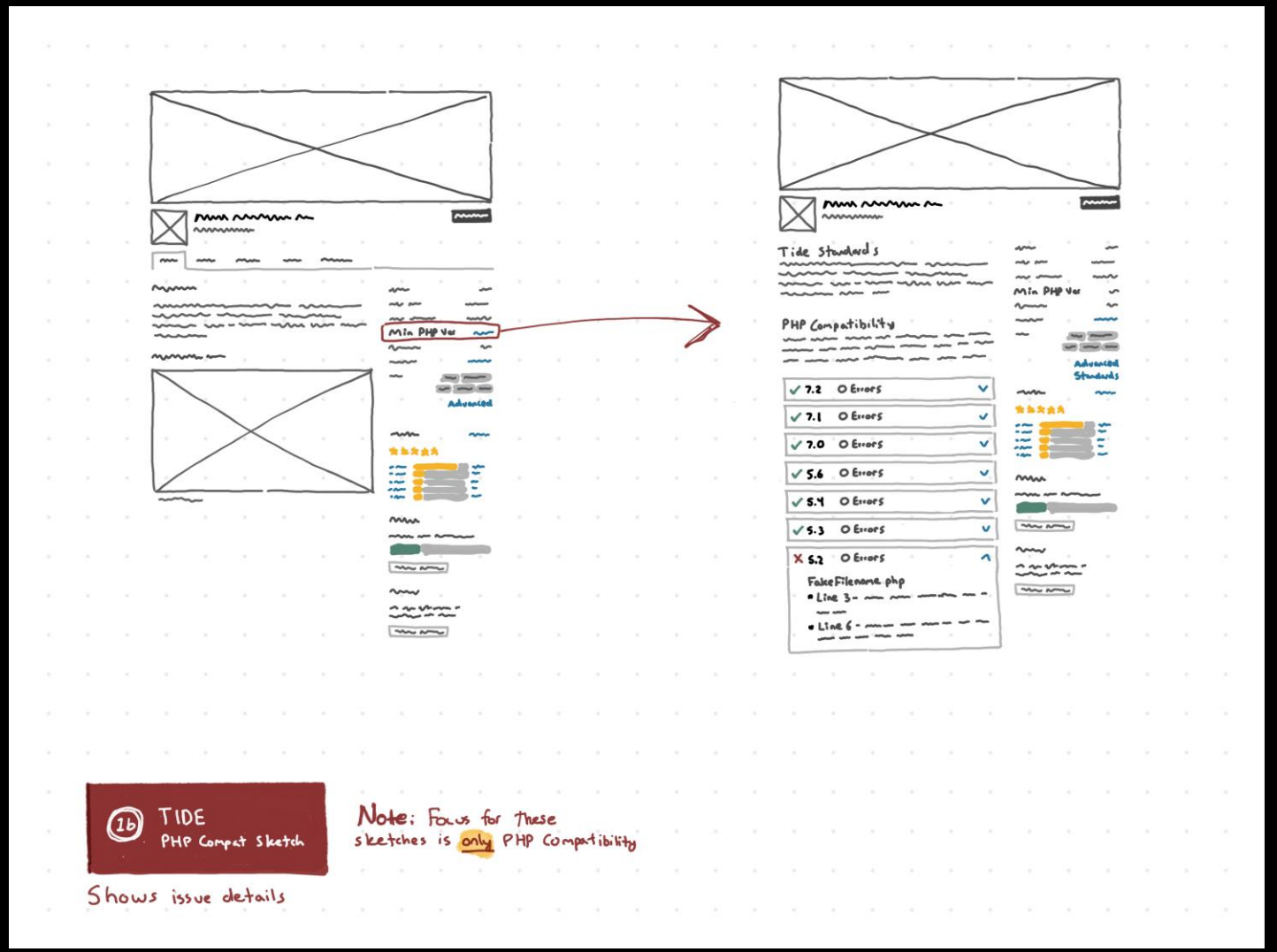
Las primeras prioridades son las de integrarlo con *wordpress.org* y se ponen en marcha equipos sobre PHPCS, WPCS, PHPCompatibility y Theme Review team standards.

Se comienza a organizar el Github de Tide con varios subproyectos.

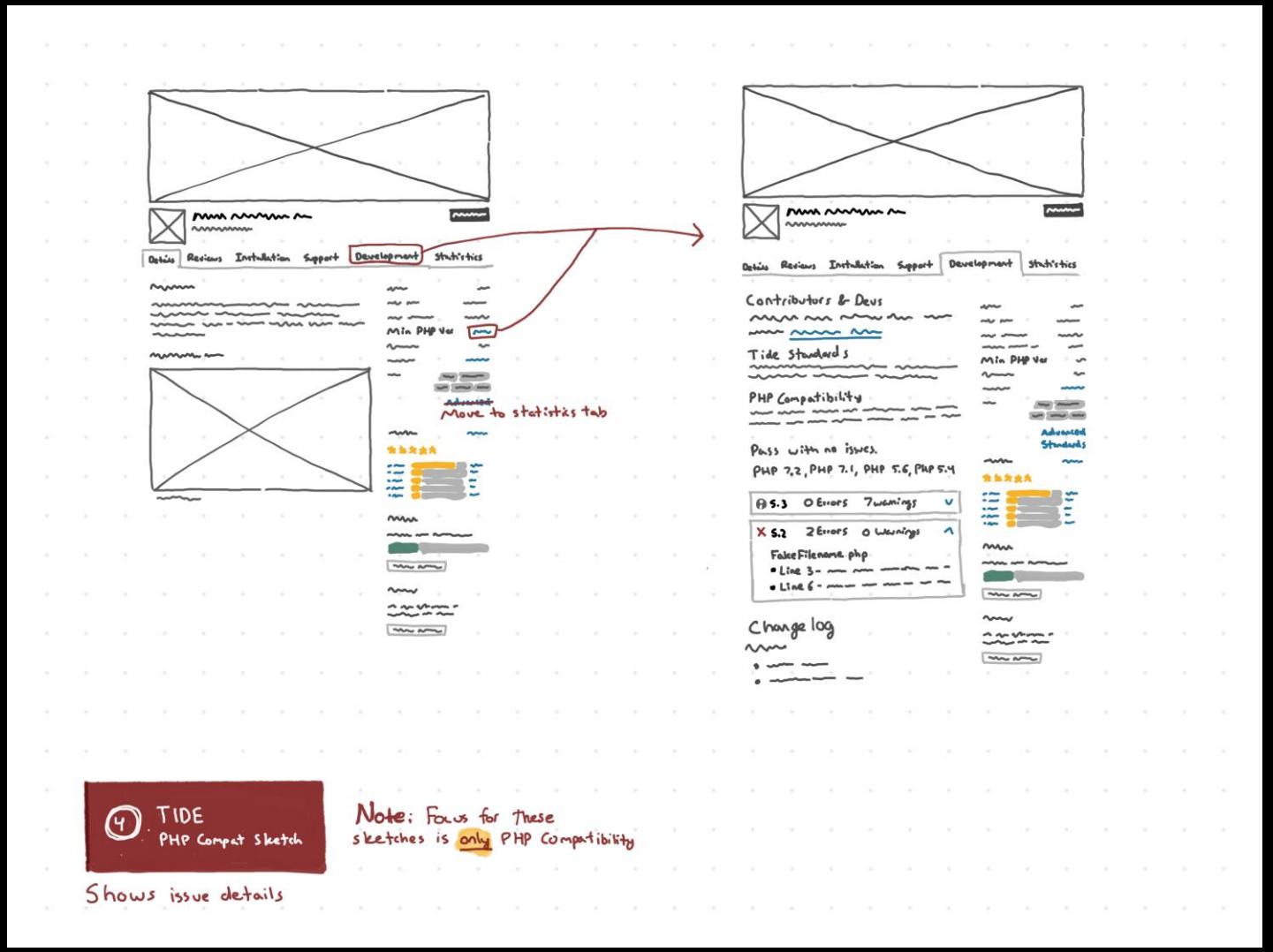
Comienzan los ejemplos con una API:

- Plugin: https://wptide.org/api/tide/v1/audit/wporg/plugin/hello-dolly
- Theme: https://wptide.org/api/tide/v1/audit/wporg/theme/twentysixteen

Primeras opciones



Primeras opciones



PHPCompatibility

Actualmente se trabaja en la creación de reglas específicas de PHPCS.

https://github.com/wptide/WP-PHPCompatibility
https://github.com/squizlabs/PHP_CodeSniffer

https://github.com/wimg/PHPCompatibility/blob/master/framework-rulesets/wordpress.xml

Liderado por William Patton [@Will_Patton_88]

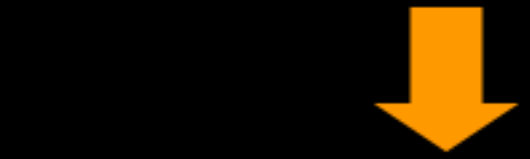
¿Un ejemplo? Machete

Hace unos días Nilo Vélez comentaba en Slack que había pasado PHPCS sobre Machete y que decidió aplicar los cambios.

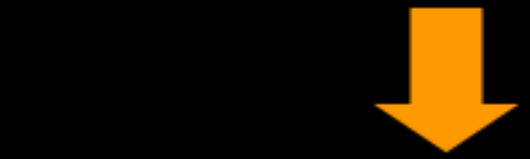
Hace dos semanas se me ocurrió la brillante idea de pasarle a Machete el PHPCodeSniffer con los perfiles que se han publicado para WordPress para poder seguir los "WordPress Coding Standards". Me ha servido para aprender un montón, pero después de un cerro de cambios todavía me quedan un puñado de errores y avisos que no se si voy a poder (o querer) arreglar.

¿Un ejemplo? Machete

```
$ phpcs -q --standard=WordPress --report=summary --parallel=4 .
----------------------------------------------------------------
A TOTAL OF 6329 ERRORS AND 726 WARNINGS WERE FOUND IN 43 FILES
----------------------------------------------------------------
```



```
$ git diff --shortstat "@{15 days ago}"
 66 files changed, 4511 insertions(+), 3757 deletions(-)
```



```
$ phpcs -q --standard=WordPress --report=summary --parallel=4 .
----------------------------------------------------------------
A TOTAL OF 159 ERRORS AND 29 WARNINGS WERE FOUND IN 13 FILES
----------------------------------------------------------------
```

Pero no es el único...

```
247     }
248
249     /**
250      * Add fields to cl_price_quantity_tab_data tab.
251      *
252      * @since    1.0.0
253      */
254     private function show_price_quantity_fields() {
255
256         $currency_symbol = get_woocommerce_currency_symbol();
257
258         $vari [phpcs] Expected next thing to be an escaping function (see Cod
259               ex for 'Data Validation'), not '$variable_para_mostrar_fallo'
260         echo $variable_para_mostrar_fallo;
261
262         echo '<div class="options_group cl_flex_container">';
263         woocommerce_wp_text_input( array(
264             'id'          => $this->product_field_name['bagged_price'],
265             'label'       => esc_html__( 'Bagged price', 'woo-products-devialia' ) . ' (' . $currency_symbol . ')',
266             'placeholder' => '0',
267             'description' => esc_html__( 'Bagged price (optional)', 'woo-products-devialia' ) . ' (' . $currency_symbol . ')',
268             'desc_tip'    => true,
269             'class'       => 'cl_input_number',
270             'wrapper_class' => 'product_cl_optional_prices cl_flex_item',
271             'data_type'   => 'price',
272         ) );
273
274         woocommerce_wp_text_input( array(
275             'id'          => $this->product_field_name['handle_price'],
276             'label'       => esc_html__( 'Handle price', 'woo-products-devialia' ) . ' (' . $currency_symbol . ')',
277             'placeholder' => '0',
278             'description' => esc_html__( 'Marked handle price (unit)', 'woo-products-devialia' ) . ' (' . $currency_symbol . ')',
279             'desc_tip'    => true,
280             'class'       => 'cl_input_number',
281             'wrapper_class' => 'product_cl_optional_prices cl_flex_item',
282             'data_type'   => 'price',
283         ) );
284         echo '</div>';
285
286         echo '<div class="options_group">';
287         // Docs in https://docs.woocommerce.com/wc-apidocs/source-function-woocommerce_wp_text_input.html#14-79 !
288         woocommerce_wp_text_input( array(
289             'id'          => $this->product_field_name['min_quantity'],
290             'label'       => esc_html__( 'Minimum manufacturing quantity', 'woo-products-devialia' ),
291             'placeholder' => '0',
292             'description' => esc_html__( 'Set to 0 if no minimum manufacturing quantity', 'woo-products-devialia' ),
293             'desc_tip'    => true,
294             'class'       => 'cl_input_number',
295             'wrapper_class' => 'product_cl_min_quantity',
```

Y esto... ¿cómo funciona?

```
mkdir ~/projects/
cd ~/projects/
git clone https://github.com/squizlabs/PHP_CodeSniffer.git phpcs
git clone -b master https://github.com/WordPress-Coding-Standards/WordPress-Coding-Standards.git wpcs
cd phpcs
./bin/phpcs --config-set installed_paths ../wpcs
phpcs -i
phpcs --standard=WordPress wp-load.php

phpcs --standard=WordPress hello-dolly
```

Y... ¿qué obtengo?

```
------------------------------------------------------------------------
FOUND 10 ERRORS AND 5 WARNINGS AFFECTING 8 LINES
------------------------------------------------------------------------
 24 | WARNING | [ ] error_reporting() can lead to full path disclosure.
 24 | WARNING | [ ] error_reporting() found. Changing configuration at runtime
    |         |     is rarely necessary.
 34 | ERROR   | [x] Expected 1 spaces before closing bracket; 0 found
 39 | WARNING | [ ] Silencing errors is discouraged
 39 | WARNING | [ ] Silencing errors is discouraged
 46 | ERROR   | [ ] Inline comments must end in full-stops, exclamation marks,
    |         |     or question marks
 46 | ERROR   | [x] There must be no blank line following an inline comment
 63 | WARNING | [ ] Detected access of super global var $_SERVER, probably
    |         |     needs manual inspection.
 63 | ERROR   | [ ] Detected usage of a non-validated input variable: $_SERVER
 63 | ERROR   | [ ] Missing wp_unslash() before sanitization.
 63 | ERROR   | [ ] Detected usage of a non-sanitized input variable: $_SERVER
 74 | ERROR   | [ ] Inline comments must end in full-stops, exclamation marks,
    |         |     or question marks
 90 | ERROR   | [x] String "Create a Configuration File" does not require
    |         |     double quotes; use single quotes instead
 92 | ERROR   | [ ] Expected next thing to be an escaping function (see Codex
    |         |     for 'Data Validation'), not '$die'
 92 | ERROR   | [ ] Expected next thing to be an escaping function (see Codex
    |         |     for 'Data Validation'), not '__'
------------------------------------------------------------------------
PHPCBF CAN FIX THE 3 MARKED SNIFF VIOLATIONS AUTOMATICALLY
------------------------------------------------------------------------
```

Usar PHPCS con tu IDE

- PhpStorm:
  PHP Code Sniffer with WordPress Coding Standards Integration
- Sublime Text:
  Setting up WPCS to work in Sublime Text
- Atom:
  Setting up WPCS to work in Atom
- Visual Studio:
  Setting up PHP CodeSniffer in Visual Studio Code
- Eclipse with XAMPP:
  Setting up WPCS when using Eclipse with XAMPP

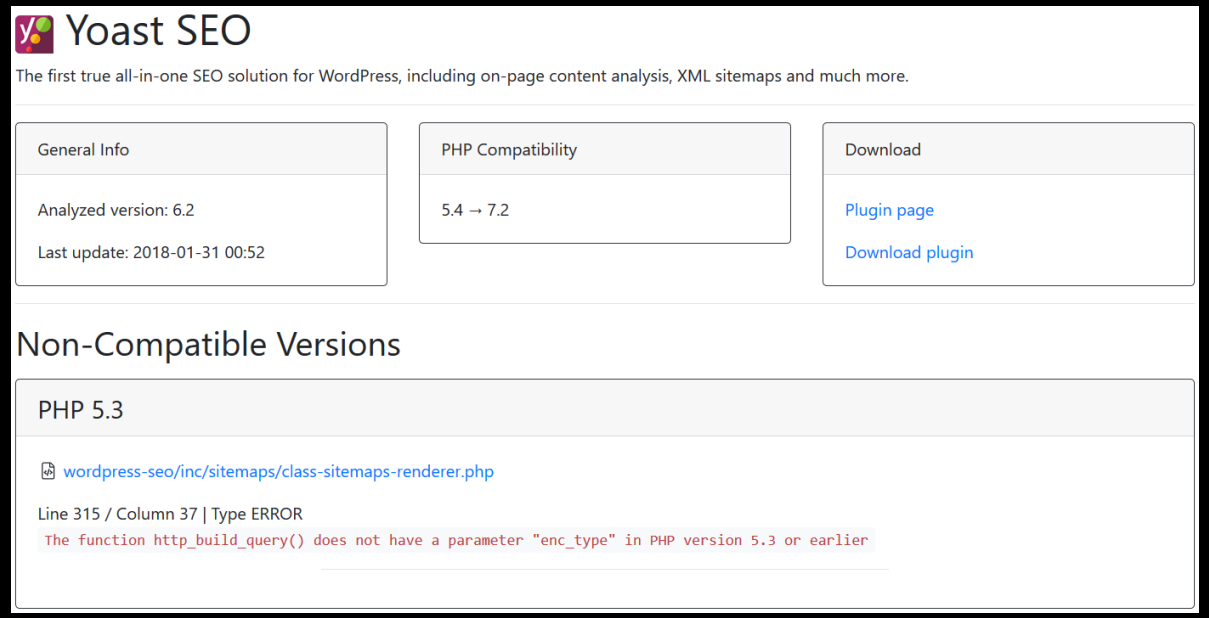
Tide en la WCEU

Si vas a la WordCamp Europe [14/16 Junio, Belgrado], es probable que haya una mesa de Comunidad sobre Project Tide.

Si te interesa, contacta con:
- Milana Cap [@DjevaLoperka]
- Jeffrey Paul [@jeffpaul]

Tide en producción

Aunque aún está como proyecto piloto, si quieres ver algunos de los plugins más utilizados y sus compatibilidades, existe el sitio WPdanger PHP Compatibility <php.wpdanger.com> donde poder analizar los *plugins* y *themes*.



Esta presentación está bajo licencia EUPL 1.2 (compatible GPL), así que descárgatela, reutilízala y mándame un saludo cuando lo hagas.
https://www.casares.blog/servehappy-tide/

¡Gracias!

¿De verdad eres de WordPress? Entonces estoy seguro de que ya lees:

https://make.wordpress.org/core/

https://make.wordpress.org/plugins/

https://make.wordpress.org/tide/

https://make.wordpress.org/hosting/