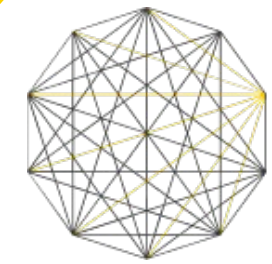


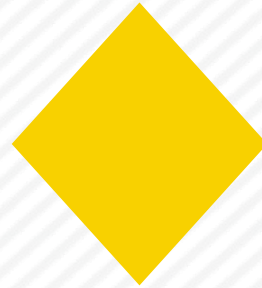
V&V

Mejora II

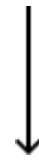
Primeros estándares y automatizaciones

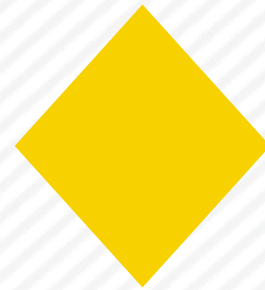


540
quinientoscuarenta

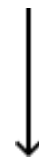


¿Qué problemas hemos tenido?





Git hooks



- Git permite añadir comportamientos customizado antes o después de realizar ciertas acciones.
- Los hooks se definen en `.git/hooks/`
- Mirad qué tenemos en `.git/hooks`
- Podemos incluir un script en cada uno de esos pasos o hooks para customizar los procesos.

Git hooks

¿Qué queremos hacer?

- Asegurarnos de que no hemos roto ningún test <- esto no quiere decir que no tengamos errores.
- Asegurarnos de que cumplimos una serie de reglas y estándares a nivel de código:
 - PHP CodeSniffer (PHPCS) con PSR12
 - PHP Mess Detector (PHPMD)
 - Test unitarios pasan

- PHP_CodeSniffer is a set of two PHP scripts; the main phpcs script that tokenizes PHP, JavaScript and CSS files to detect violations of a defined coding standard, and a second phpcbf script to automatically correct coding standard violations.
- PHP_CodeSniffer is an essential development tool that ensures your code remains clean and consistent.
- https://github.com/squizlabs/PHP_CodeSniffer#installation
- <https://blog.quickadminpanel.com/psr-2-and-psr-12-why-we-need-standards-and-how-to-apply-them/>

Git hooks

PHP Mess Detector (PHPMD)

- PHPMD is a spin-off project of PHP Depend and aims to be a PHP equivalent of the well known Java tool PMD.
- PHPMD can be seen as an user friendly frontend application for the raw metrics stream measured by PHP Depend.
- <https://phpmd.org/download/index.html>

- Montar un sistema automatizado de validación de nuestro código al commitear
 - Montar Grump <https://github.com/phpro/grumphp>
 - Configuración <https://github.com/phpro/grumphp/blob/master/doc/tasks.md>
 - Validar nombre de commit con nuestros estándares
 - Añadir validación PHP CodeSniffer (PHPCS) con PSR12
 - Añadir validación PHP Mess Detector (PHPMD) con reglas 'cleancode', 'codesize', 'naming'

- Commits:
 - [issueNumber] - Description of the commit:
 - [1746] - A good commit description
- Ramas:
 - issueNumber-descripción-de-la-rama
 - 1746-una-nueva-rama-para-nueva-funcionalidad
- PRs
 - Usamos “Squash and merge”
 - Las ramas se eliminan automáticamente
 - Nombres en el squash:
 - [issueNumber] - Description of the commit (#pull request reference)
 - [3085] - Show initially needed accessories quantities for radiator series (#3364)

Composer

- Gestor de paquetes de PHP
- Proporciona un estándar para administrar, descargar e instalar dependencias y librerías
- Gestiona las dependencias de forma automática
- Necesario cuando trabajamos en proyectos complejos que dependen de múltiples fuentes de instalación
- `composer.json` y `composer.lock`
- Ruby -> Bundler, Node -> npm



Composer

- `composer install`
- `composer require package-name`
- `composer require-dev package-name`
- `composer update`

Git hooks

Recursos interesantes

- <https://github.com/phpro/grumphp>
- <https://carlosbuenosvinos.com/write-your-git-hooks-in-php-and-keep-them-under-git-control/> -> explicación de hooks
- <https://github.com/intaro/php-git-hooks> -> buen ejemplo sin test unitarios
- <https://github.com/allysonsilva/php-pre-commit>