

# Interoperabilidad

Ricardo Pérez López

IES Doñana, curso 2019/2020



1. Versionado semántico
2. Composer
3. Autocarga de clases
4. Ejemplos
5. Recomendaciones PSR del PHP-FIG (Framework Interop Group)
6. Paquetes de Atom y herramientas externas #opcional
7. Ejercicios

# 1. Versionado semántico

## 2. Composer

2.1 Paquetes

2.2 Packagist

2.3 Dependencias

2.4 Versiones y restricciones

2.5 Comandos básicos

2.6 Entornos de desarrollo y producción

## 2.1. Paquetes

## 2.2. Packagist

## 2.3. Dependencias

## 2.3. composer.json y composer.lock



## 2.4. Versiones y restricciones

## 2.4. Versión exacta

- ricpelo's note: - `x.y.z` es exactamente esa versión
- `x.y` equivale a `x.y.0`
  - `x` equivale a `x.0.0`
- Se usa en los siguientes tipos de restricciones

## 2.4. Rango ( $>$ , $\geq$ , $<$ , $\leq$ , $\neq$ , $,$ , $,$ , $||$ )

## 2.4. Guión (-)

## 2.4. Asterisco (\*)

ricpelo's note: -  $x.y.*$  equivale a  $\geq x.y < x.(y+1)$   
-  $x.*$  equivale a  $\geq x < (x+1)$   
-  $*$  equivale a cualquiera

## 2.4. Tilde (~)

ricpelo's note: Especifica una versión mínima y permite que avance el último dígito indicado pero no el primero.

ricpelo's note: -  $\sim x.y.z$  equivale a  $\geq x.y.z < x.(y+1)$

-  $\sim x.y$  equivale a  $\geq x.y < (x+1)$

-  $\sim x$  equivale a  $\geq x < (x+1)$

## 2.4. Gorrito (^)

ricpelo's note: También permite especificar una versión mínima, pero impide actualizaciones que rompen la compatibilidad hacia atrás (es decir, permite que aumenten todos los números excepto el *mayor*).

ricpelo's note: -  $^x.y.z$  equivale a  $\geq x.y.z < (x+1)$

-  $^x.y$  equivale a  $\geq x.y < (x+1)$

-  $^x$  equivale a  $\geq x < (x+1)$

## 2.4. Nombres de rama

### dev-master

ricpelo's note: Equivale al último commit de la rama `master`.

### 5.1.x-dev

ricpelo's note: Equivale al último commit de la rama `5.1`.



## 2.4. Estabilidad mínima

## 2.4. Comprobador online de restricciones

ricpelo's note: Probar con `laravel/laravel`.

## 2.5. Comandos básicos

## 2.5. require

## 2.5. install

## 2.5. update

## 2.6. Entornos de desarrollo y producción

## 3. Autocarga de clases

3.1 `spl_autoload_register()`

3.2 PSR-4

3.3 Autoloader de Composer



### 3.1. spl\_autoload\_register()

## 3.1. spl\_autoload\_register()

ricpelo's note: - Cuando se llama sin argumentos, registra el manejador predeterminado (`spl_autoload()`), el cual autocarga el archivo `.php` a partir del nombre de la clase **en minúsculas** (la clase `A` la busca en `a.php`).

- `spl_autoload_register(function ($c) { include "$c.php"; });`
  - `spl_autoload_register(function ($c) { include strtolower(str_replace('\\', '/', $c) . '.php'); });`

## 3.2. PSR-4

### 3.3. Autoloader de Composer

## 4. Ejemplos

4.1 `mpdf/mpdf`

4.2 `ramsey/uuid`

4.3 `doctrine/inflector`

## 4.1. mpdf/mpdf

## 4.1. mpdf/mpdf

ricpelo's note:

```
<?php
// composer require mpdf/mpdf

require 'vendor/autoload.php';

$mpdf = new Mpdf\Mpdf();
$mpdf->WriteHTML('<h1>¡Hola, mundo!</h1>');
$mpdf->Output('salida.pdf');
```

## 4.2. ramsey/uuid



## 4.3. doctrine/inflector

## 5. Recomendaciones PSR del PHP-FIG (Framework Interop Group)

5.1 PSR-1: Basic Coding Standard

5.2 PSR-2: Coding Style Guide

5.3 PSR-4: Autoloading Standard

5.4 PSR-5: PHPDoc Standard (borrador)

5.5 PSR-11: Extended Coding Style Guide (borrador)

5.6 PSR-19: PHPDoc tags (borrador)

## 5. Recomendaciones PSR del PHP-FIG (Framework Interop Group)

ricpelo's note: PHP Standard Recommendation

## 5.1. PSR-1: Basic Coding Standard

## 5.2. PSR-2: Coding Style Guide

## 5.3. PSR-4: Autoloading Standard

## 5.4. PSR-5: PHPDoc Standard (borrador)

## 5.5. PSR-11: Extended Coding Style Guide (borrador)



## 5.6. PSR-19: PHPDoc tags (borrador)

## 6. Paquetes de Atom y herramientas externas #opcional

6.1 PHP\_CodeSniffer

6.2 PHP-CS-Fixer

6.3 Yii2-Shell

## 6. Paquetes de Atom y herramientas externas #opcional

ricpelo's note:

```
$ composer global require --prefer-dist friendsofphp/php-cs-  
fixer "squizlabs/php_codesniffer:^2.7" yiisoft/yii2-coding-  
standards  
$ sudo ln -sf /opt/composer ~/.config/composer  
$ apm install linter-phpcs php-cs-fixer  
$ phpcs --config-set default_standard /opt/composer/vendor/yiisoft/yii2-  
coding-standards/Yii2
```

ricpelo's note: <https://github.com/ricpelo/conf>

## 6.1. PHP\_CodeSniffer

## 6.2. PHP-CS-Fixer

## 6.3. Yii2-Shell

## 7. Ejercicios

7.1 De versionado semántico

7.2 De versiones y restricciones

7.3 De uso básico de Composer

7.4 De buscar paquetes en Packagist que tengan una funcionalidad concreta y usarlos en un ejemplo

## 7.1. De versionado semántico



## 7.2. De versiones y restricciones

## 7.3. De uso básico de Composer

## 7.4. De buscar paquetes en Packagist que tengan una funcionalidad concreta y usarlos en un ejemplo