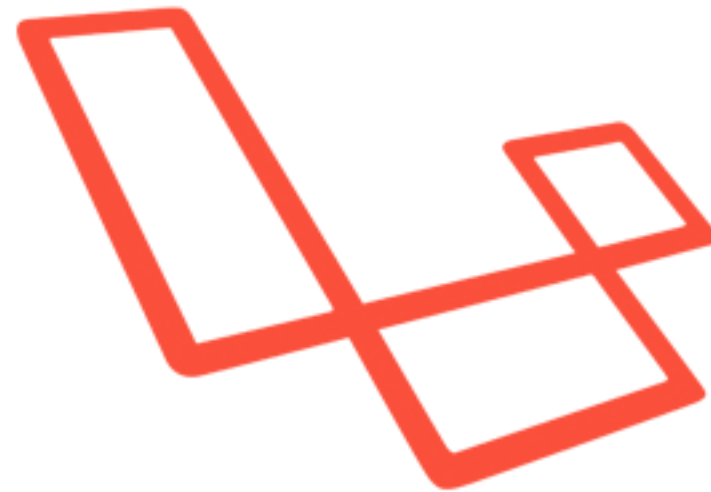
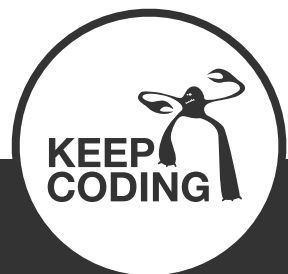


PHP 7 y Laravel

php



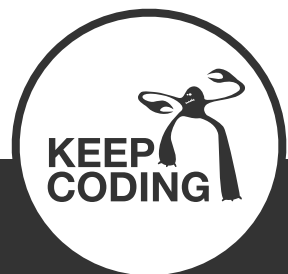
laravel



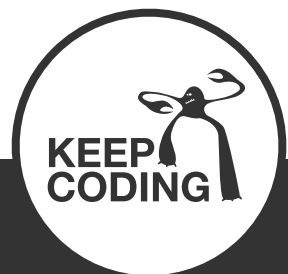


■ 1. Introducción

Nada suele ser tan malo como lo pintan



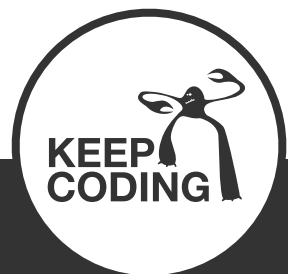
■ When people tell me PHP is not a real programming language



■ Quién soy

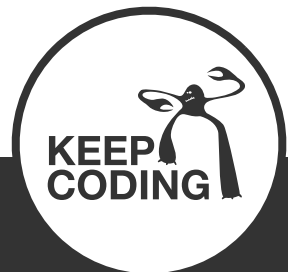


- Alicia Rodríguez
- Ingeniera industrial ICAI
- Backend developer
- @buzkall
- buzkall.com



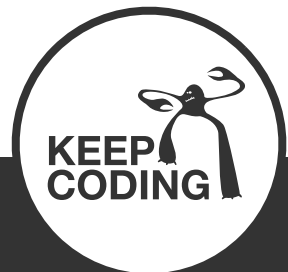
■ ¿Qué vamos a ver?

- Instalación y desarrollo en local
- PHP 7
- Laravel
- Test unitarios
- Cómo utilizar una API externa



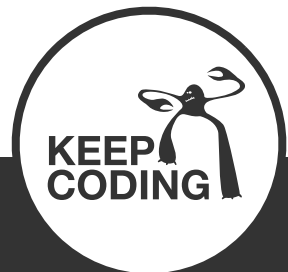
■ ¿Qué sabremos al terminar?

- PHP mola
- Crear un proyecto de cero
- Depurar y hacer test a nuestro código
- Un poco de análisis técnico y bolsa

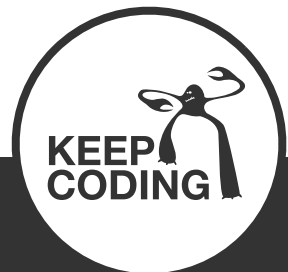
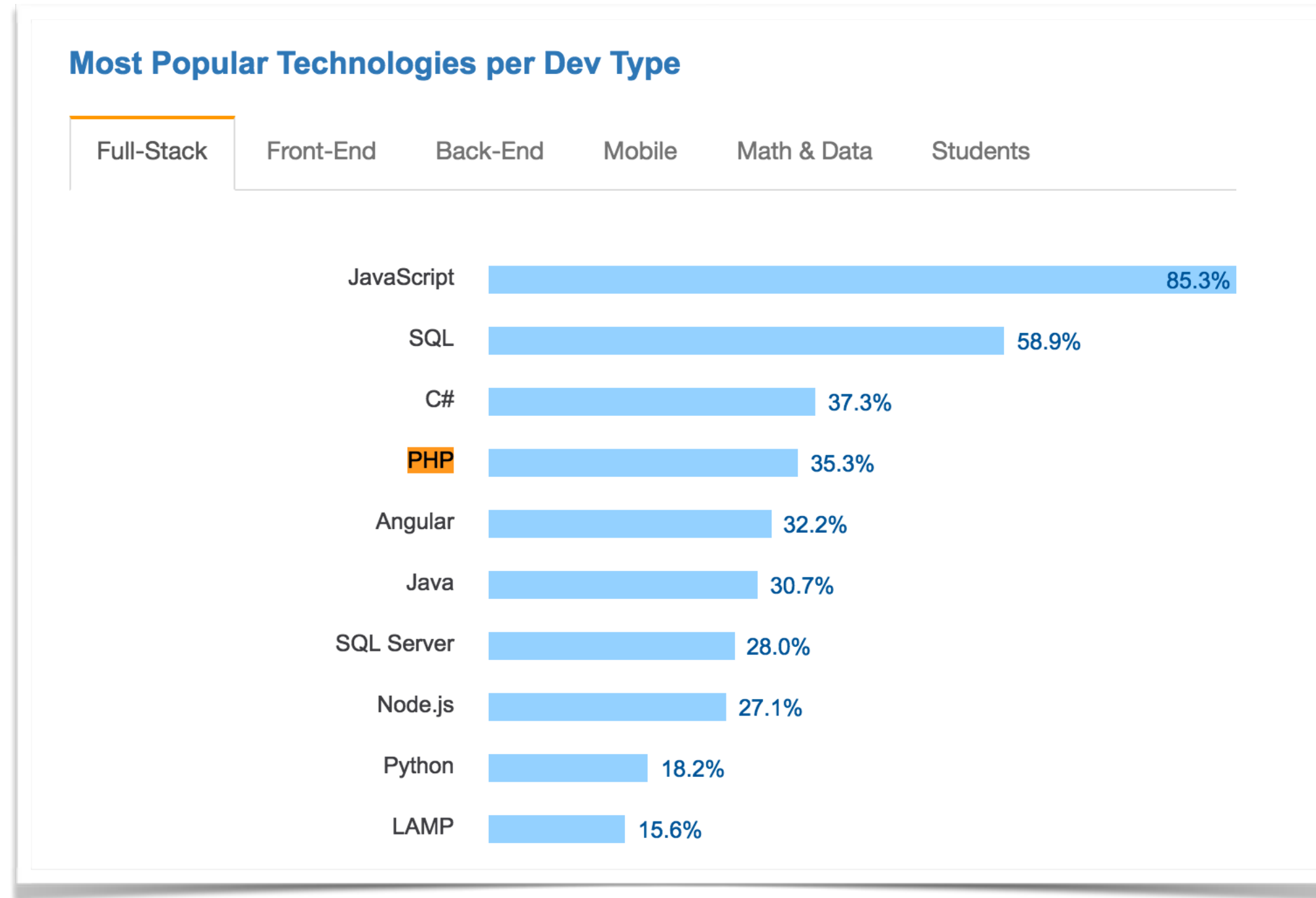


■ Seguridad

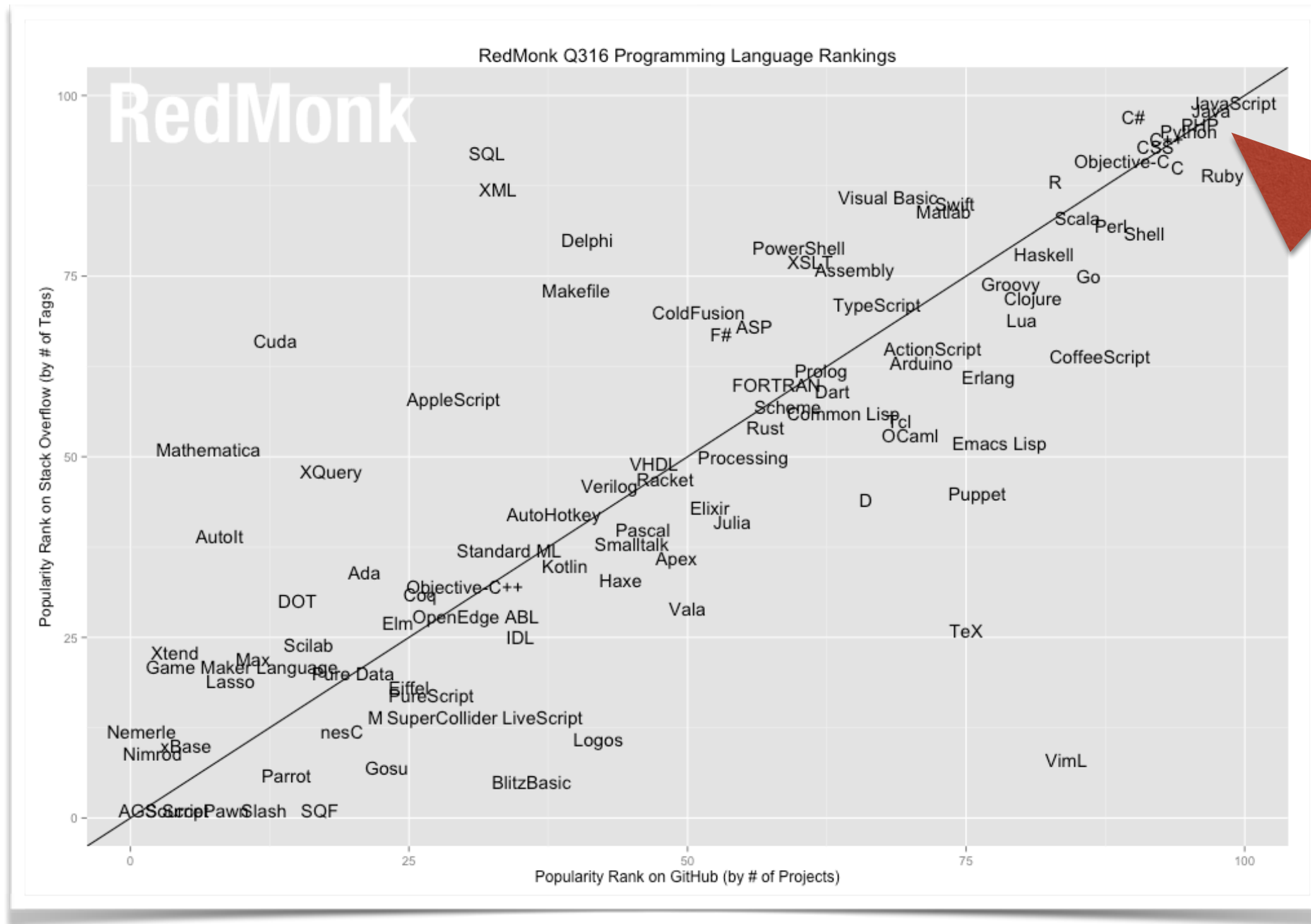
Security is not a characteristic of a language as much as it is a characteristic of a developer



■ Popularidad en Stackoverflow



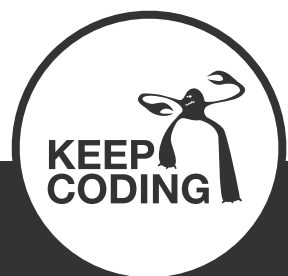
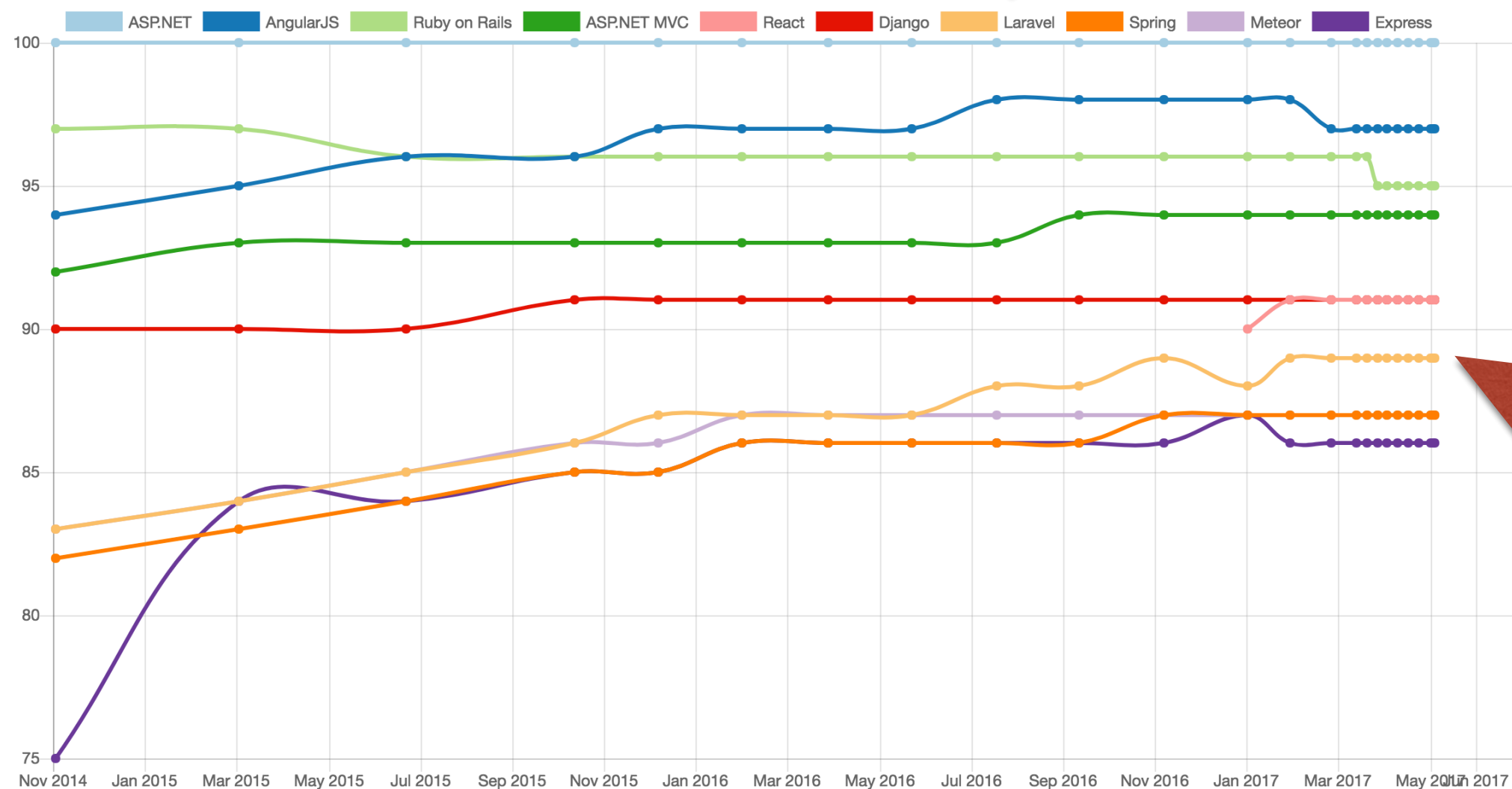
Popularity en Github



Frameworks por lenguaje

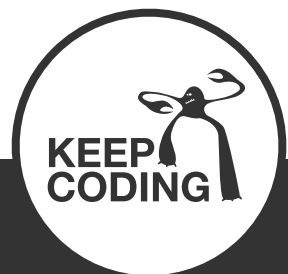
Find your new favorite web framework

Measuring web framework popularity so you can find interesting frameworks to check out



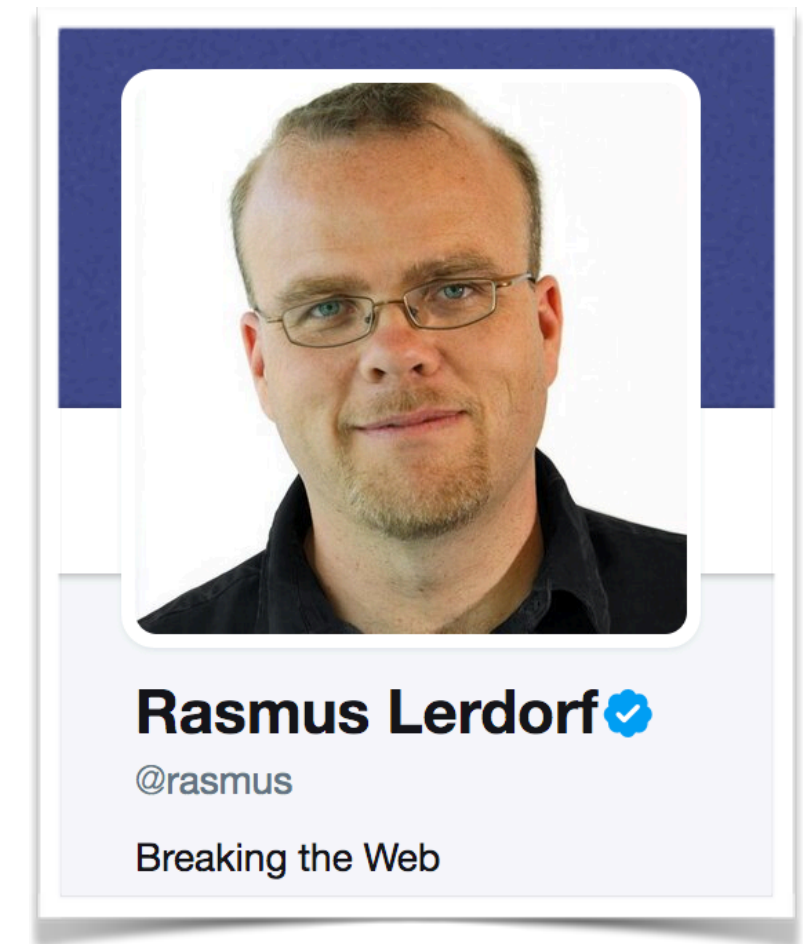
■ Su propia descripción

- 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.



■ Historia de PHP

- Creado por Rasmus Lerdorf en 1995 como el conjunto de scripts "Personal Home Page Tools", referenciado como "PHP Tools".
- PHP 3.0 se liberó en 1998 y se estima que se usaba en 70k dominios
- PHP 5 introdujo la orientación a objetos en 2004
- En diciembre de 2015 se liberó PHP 7.0
- La versión actual es la 7.1.4



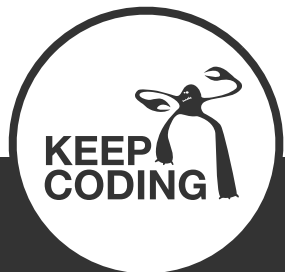
<https://twitter.com/rasmus>
<https://toys.lerdorf.com/>



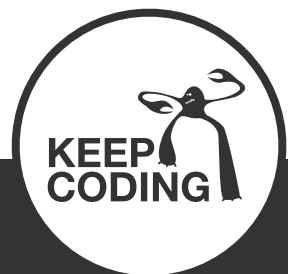
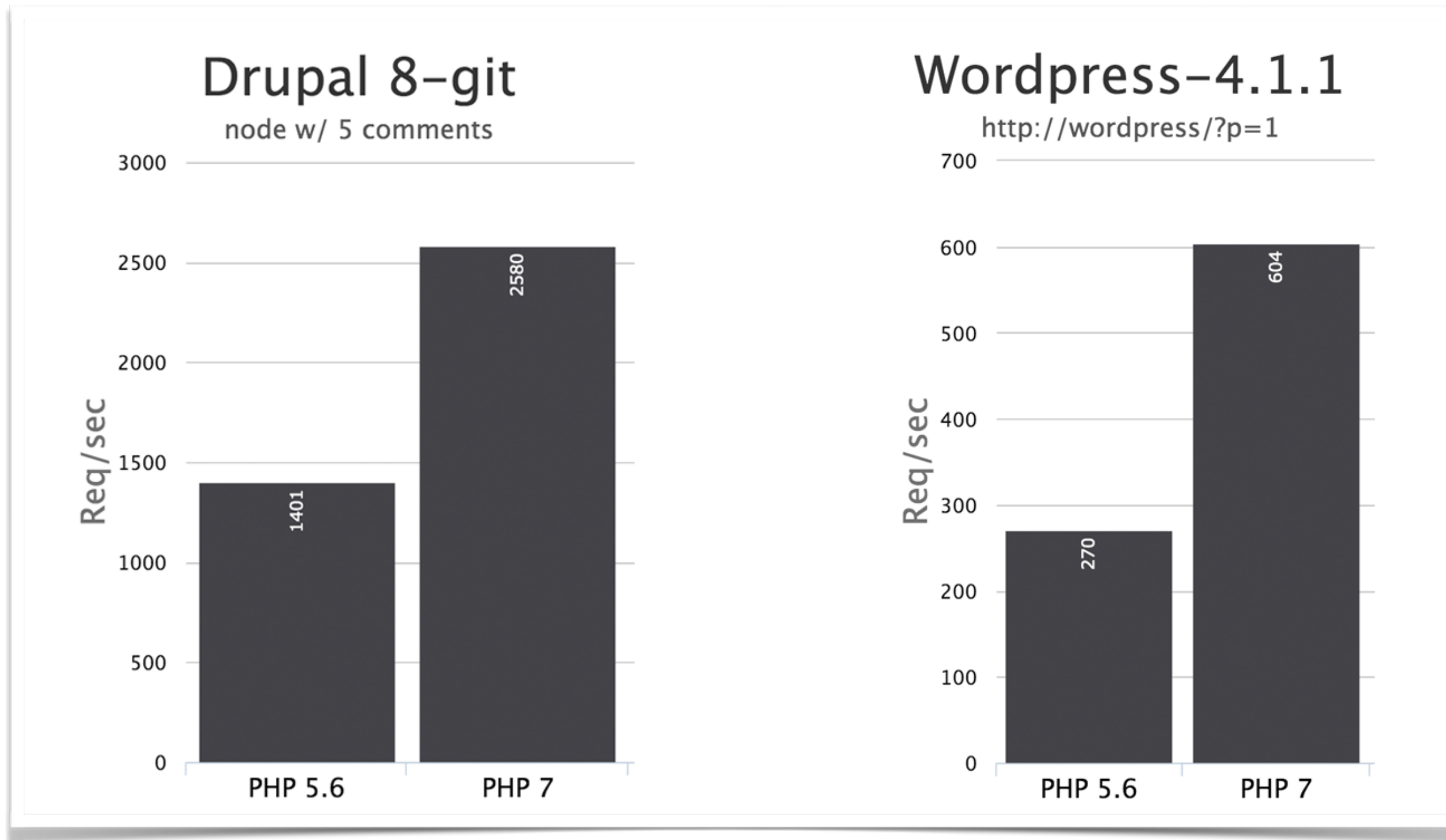
■ Salto de PHP 5 a PHP 7

- PHP 6 iba a tener soporte Unicode y una importante mejora de rendimiento.
- Reinó el caos:
 - Finalmente se optó por UTF-16
 - Se portaron las modificaciones a la rama de PHP 5.4 (continuó hasta PHP 5.6)
 - Se empezó de cero con PHP 7

<https://www.slideshare.net/andreizm/the-good-the-bad-and-the-ugly-what-happened-to-unicode-and-php-6>
[Andrei Zmievski]

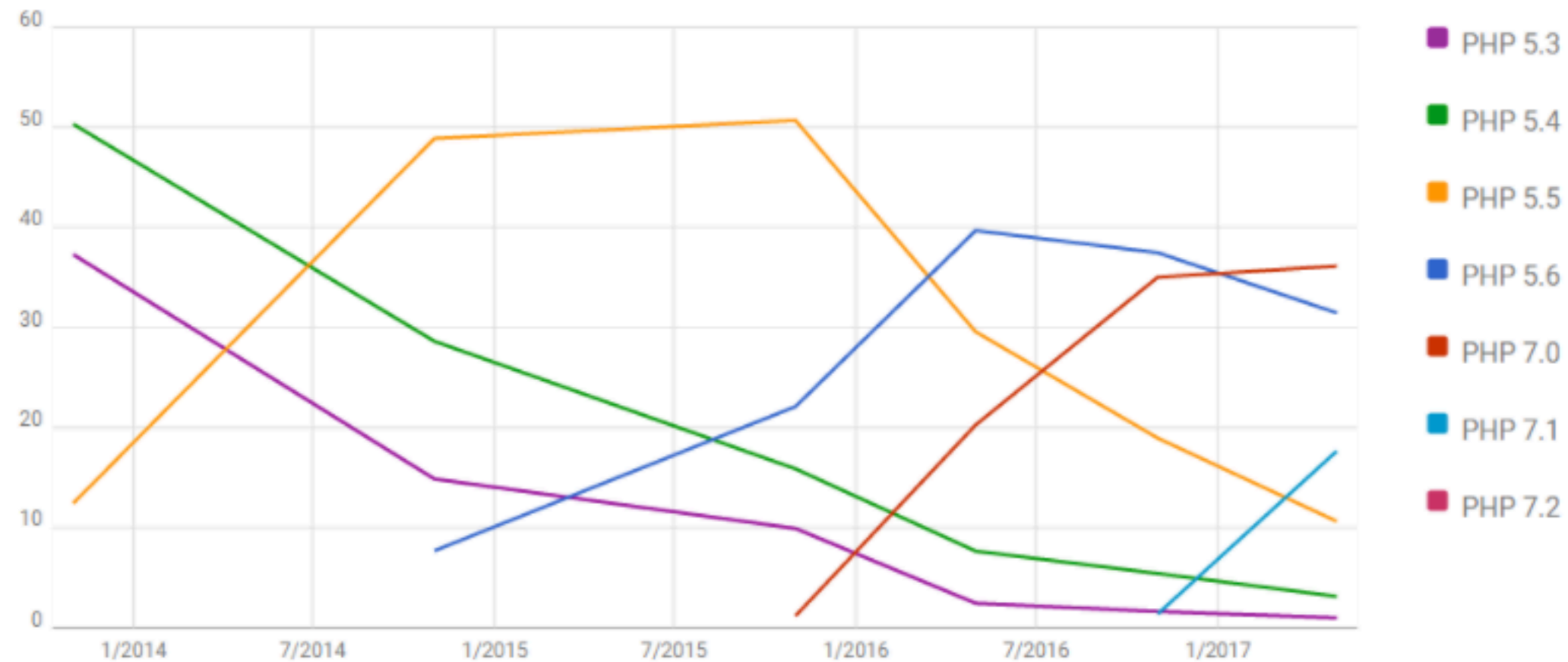


Benchmark

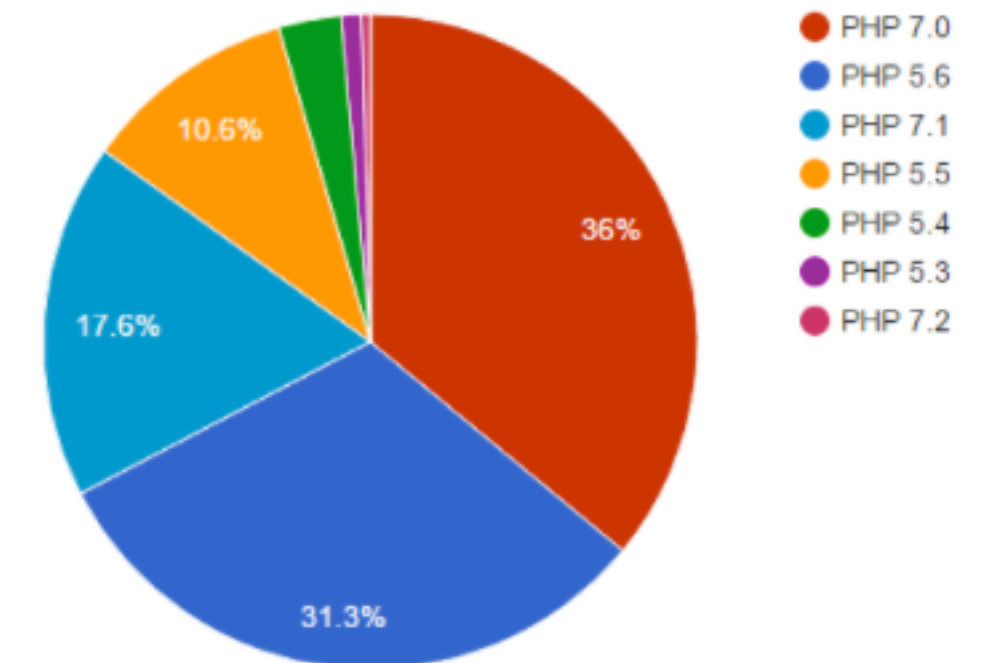


Estado migración a PHP 7

PHP Versions (based on Composer usage data)

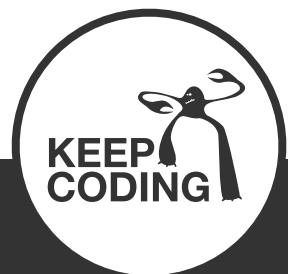
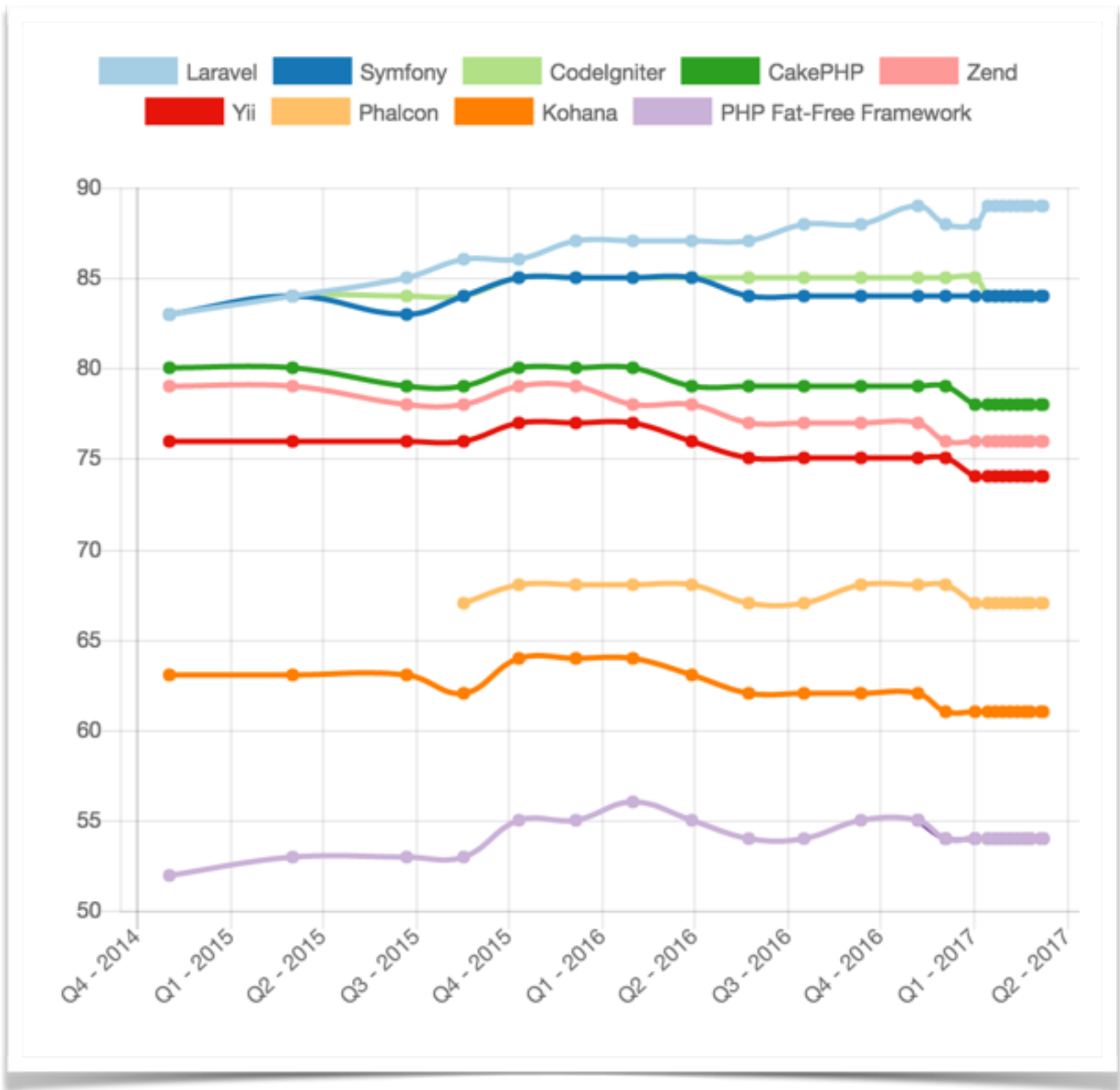


PHP Versions Grouped 2017-05



Frameworks PHP

- Wordpress
- Laravel
- Drupal
- Codeigniter
- Symfony
- Joomla



■ Historia de Laravel

- Creado por Taylor Otwell en 2011 como alternativa a CodeIgniter.
- En 2012 Laravel 3 incorpora artisan
- En 2013 Laravel 4 fue una reescritura completa que ya usaba composer
- Actualmente está en la versión 5.4
- Es uno de los frameworks actuales con más seguidores



You have arrived.



■ Empresas que usan PHP

- Facebook - HHVM, Hack

- Etsy

- Wikipedia

- Mailchimp

 - <https://blog.mailchimp.com/ewww-you-use-php/>

- Slack

 - <https://slack.engineering/taking-php-seriously-cf7a60065329>

- Tumblr

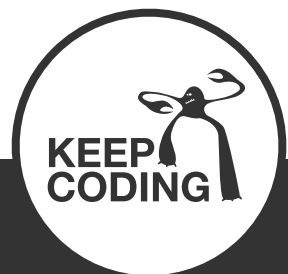
 - <https://engineering.tumblr.com/post/152998126990/php-7-at-tumblr>



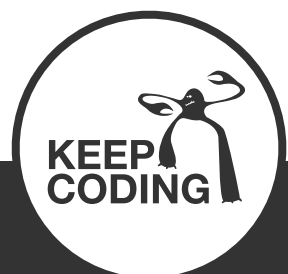
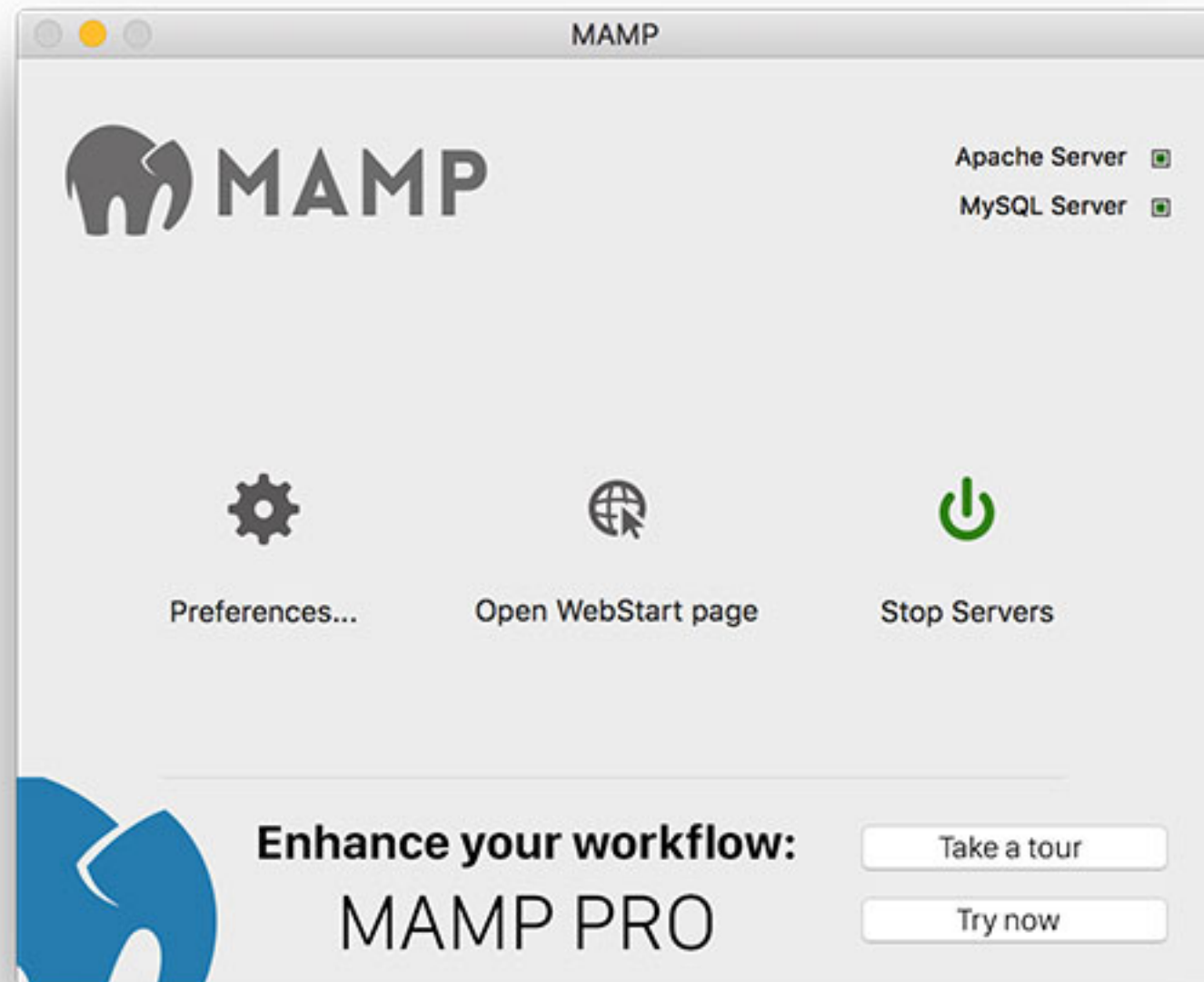


■ 2. Instalación

Monta tu entorno más allá de MAMP/WAMP



■ Solución básica: MAMP/WAMP

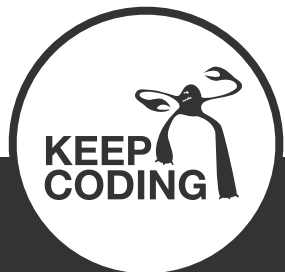


■ Alternativa Mac: brew

- Homebrew:
<https://brew.sh>

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

- La mejor guía para montar tu entorno:
<https://getgrav.org/blog/mac-os-sierra-apache-multiple-php-versions>
- mac trae PHP y Apache instalado de serie, pero son siempre versiones anteriores (Sierra: 5.6.24)



■ Instalación entorno local

- Install Apache: http24:

```
brew install httpd24 --with-privileged-ports --with-http2
```

- Apache httpd.conf

- Install php 7.1

```
brew install php71 --with-httpd24
```

```
Edit httpd.conf php_module
```

- Install mysql

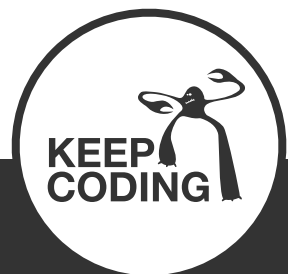
```
brew install mysql ó brew install mariadb
```

```
mysql_install_db
```

- Apache vhosts

- Xdebug

```
brew install php71-xdebug
```



■ Alternativa a Apache: Laravel Valet

>> “Valet is a Laravel development environment for Mac minimalists. No Vagrant, no /etc/hosts file. You can even share your sites publicly using local tunnels. Yeah, we like it too.”

- valet park
- valet link
- valet secure

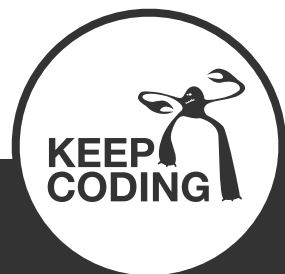
```
1. bash
[~] 🍌 $ valet
Password:
Laravel Valet 2.0.4

Usage:
  command [options] [arguments]

Options:
  -h, --help           Display this help message
  -q, --quiet          Do not output any message
  -V, --version        Display this application version
  --ansi              Force ANSI output
  --no-ansi           Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  -vv|vvv, --verbose  Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  domain      Get or set the domain used for Valet sites
  fetch-share-url  Get the URL to the current Ngrok tunnel
  forget      Remove the current working (or specified) directory from Valet's list of paths
  help        Displays help for a command
  install     Install the Valet services
  link        Link the current working directory to Valet
  links       Display all of the registered Valet links
  list        Lists commands
  on-latest-version Determine if this is the latest version of Valet
  open        Open the site for the current (or specified) directory in your browser
  park        Register the current working (or specified) directory with Valet
  paths       Get all of the paths registered with Valet
  restart     Restart the Valet services
  secure      Secure the given domain with a trusted TLS certificate
  share       Generate a publicly accessible URL for your project
  start       Start the Valet services
  stop        Stop the Valet services
  uninstall   Uninstall the Valet services
  unlink      Remove the specified Valet link
  unsecure    Stop serving the given domain over HTTPS and remove the trusted TLS certificate
  which       Determine which Valet driver serves the current working directory

[~] 🍌 $
```



■ Consola PHP

```
php -a

php > print_r($_SERVER);
Array
(
    [TERM_PROGRAM] => iTerm.app
    [TERM] => xterm-256color
    [SHELL] => /bin/bash
    [TMPDIR] => /var/folders/t3/l18ty
    [TERM_PROGRAM_VERSION] => 3.0.15
    [USER] => buzkall
    [SSH_AUTH_SOCK] => /private/tmp/
com.apple.launchd.acEnaoOAR6/Listeners
    [__CF_USER_TEXT_ENCODING] => 0x1F5:0x0:0x0
    [PATH] => /Applications/MAMP/Library/bin:/usr/local/
sbin:~/composer/vendor/bin:/usr/local/bin:/usr/bin:/bin:/usr/
sbin:/sbin
    [PWD] => /Users/buzkall
    [EDITOR] => vim
    [ITERM_PROFILE] => Default
    ...
    [REQUEST_TIME] => 1494418907
```

```
[~] 🤖 $ php -a
Interactive shell

php > echo "Hello World";
Hello World
php > $a = 1;
php > $b = 2;
php > $c = $a + $b;
php > echo $c;
3
php > |
```



■ Otros comandos de PHP

- Usar php como servidor (PHP \geq 5.4)

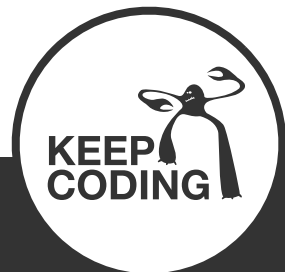
```
php -S localhost:8000
```

- Ver el php.ini cargado

```
php -i
```

- Buscar un elemento de configuración concreto

```
php -i | grep error_log
```



Composer

- Gestor de dependencias de PHP
- Similar a npm (node), bundler (ruby) o pip (python)
- En el proyecto quedan dos ficheros:
 - *composer.json*: requisitos y las versiones mínimas o máximas
 - *composer.lock*: lo que se ha instalado

```
composer install
```

(si ya existe composer.lock instalará esas versiones)

```
composer update
```



Composer. Instalación

- Mac:

```
brew install composer
```

- Linux:

```
curl -sS https://getcomposer.org/installer |  
sudo php -- --install-dir=/usr/local/bin --  
filename=composer
```

- Windows:

Descargar Composer-Setup.exe

```
https://www.youtube.com/watch?v=ZocYVPP3nQY https://  
getcomposer.org/doc/00-intro.md#installation-windows
```



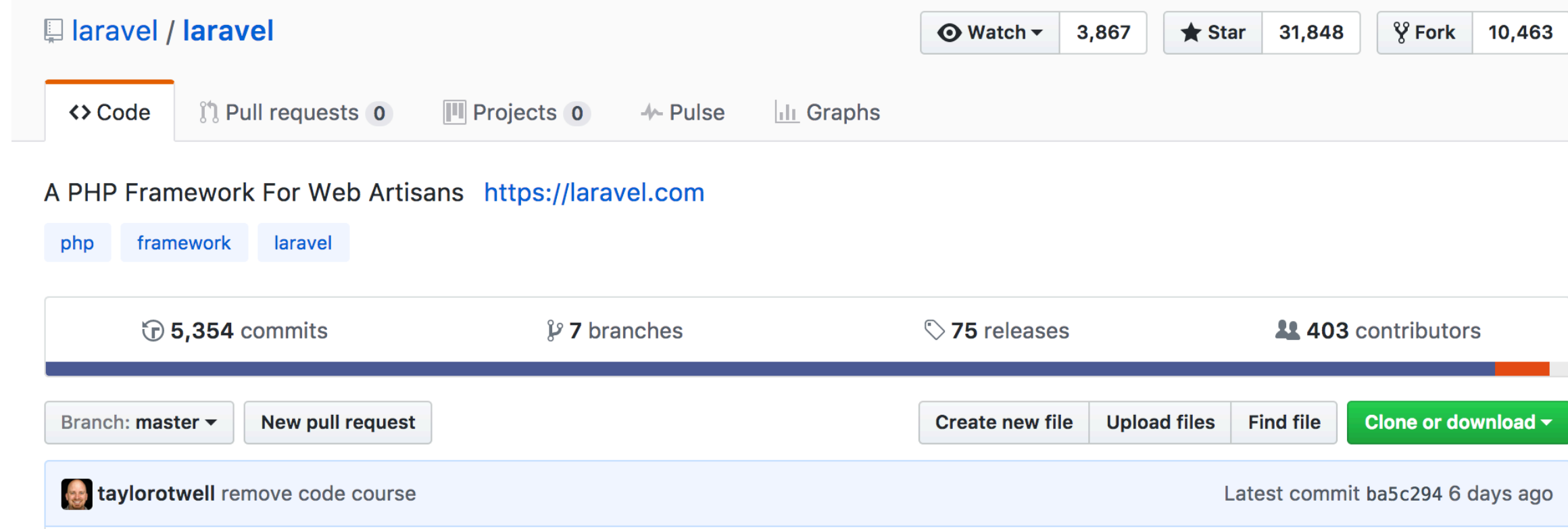
■ Instalación Laravel

- Vía rápida:

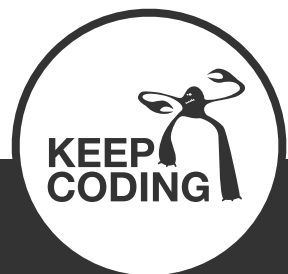
```
composer global require "laravel/installer"  
laravel new NAME ○ laravel new
```

- Instalación manual

```
https://github.com/laravel/laravel
```



The screenshot shows the GitHub repository page for laravel/laravel. At the top, it displays the repository name 'laravel / laravel' and statistics: 3,867 Watchers, 31,848 Stars, and 10,463 Forks. Below this, there are navigation tabs for Code, Pull requests (0), Projects (0), Pulse, and Graphs. The repository description is 'A PHP Framework For Web Artisans' with a link to 'https://laravel.com'. There are tags for 'php', 'framework', and 'laravel'. A summary bar shows 5,354 commits, 7 branches, 75 releases, and 403 contributors. At the bottom, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit by 'taylorotwell' is shown with the message 'remove code course' and the latest commit hash 'ba5c294' from 6 days ago.



Actualizaciones vía Composer

- Todos los paquete se pueden actualizar desde el terminal

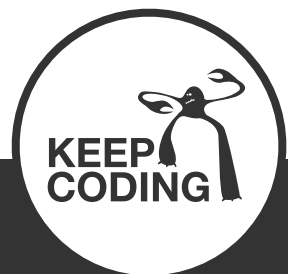
```
composer update
```

- Si necesitase actualizar el autoloader pero no los paquetes

```
composer dump  
composer update nothing
```

```
1. bash  
[code] $ composer update  
Loading composer repositories with package information  
Updating dependencies (including require-dev)  
Package operations: 0 installs, 20 updates, 0 removals  
- Updating symfony/css-selector (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/var-dumper (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/routing (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/process (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/http-foundation (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/event-dispatcher (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/debug (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/http-kernel (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/finder (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating symfony/console (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating swiftmailer/swiftmailer (v5.4.6 => v5.4.8): Downloading (100%)  
- Updating symfony/translation (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating league/flysystem (1.0.37 => 1.0.40): Downloading (100%)  
- Updating laravel/framework (v5.4.16 => v5.4.23): Downloading (100%)  
- Updating felixkiss/uniquewith-validator (3.0.0 => 3.1.1): Downloading (100%)  
- Updating symfony/yaml (v3.2.6 => v3.2.8): Downloading (100%)  
- Updating phpunit/php-code-coverage (4.0.7 => 4.0.8): Downloading (100%)  
- Updating myclabs/deep-copy (1.6.0 => 1.6.1): Downloading (100%)  
- Updating phpunit/phpunit (5.7.17 => 5.7.19): Downloading (100%)  
- Updating psy/psysh (v0.8.3 => v0.8.4): Downloading (100%)  
Writing lock file  
Generating optimized autoload files  
> Illuminate\Foundation\ComposerScripts::postUpdate  
> php artisan optimize  
Generating optimized class loader  
The compiled services file has been removed.
```

```
    "name": "laravel/framework",  
-   "version": "v5.4.16",  
+   "version": "v5.4.23",  
    "source": {  
        "type": "git",  
        "url": "https://github.com/laravel/framework.git",  
-       "reference": "6cf379ec34d08bc9c7183e369a8fdf04ade80d",  
+       "reference": "ad82327705658dbf5f0ce72805caa950dfbe150d",  
    },  
    "dist": {  
        "type": "zip",  
-       "url": "https://api.github.com/repos/laravel/framework/zipball/6cf379ec34d08bc9c7183e369a8fdf04ade80d",  
-       "reference": "6cf379ec34d08bc9c7183e369a8fdf04ade80d",  
+       "url": "https://api.github.com/repos/laravel/framework/zipball/ad82327705658dbf5f0ce72805caa950dfbe150d",  
+       "reference": "ad82327705658dbf5f0ce72805caa950dfbe150d",  
    }  
}
```



■ Servidor de Laravel: Artisan

- Interfaz de línea de comandos de Laravel
- Lo más usado:

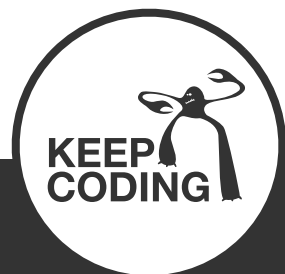
```
php artisan serve
php artisan migrate
php artisan make:xxx
```

```
1. bash
[code] 😊 $ php artisan list
Laravel Framework 5.4.16

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version         Display this application version
  --ansi               Force ANSI output
  --no-ansi            Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  --env[=ENV]         The environment the command should run under
  -vv|vvv, --verbose   Increase the verbosity of messages: 1 for normal output, 2 for more ve
rbose output and 3 for debug

Available commands:
  clear-compiled      Remove the compiled class file
  down               Put the application into maintenance mode
  env                Display the current framework environment
  help              Displays help for a command
  inspire           Display an inspiring quote
  list              Lists commands
  migrate           Run the database migrations
  optimize          Optimize the framework for better performance
  serve            Serve the application on the PHP development server
  tinker           Interact with your application
  up               Bring the application out of maintenance mode
  app
  app:name          Set the application namespace
  auth
  auth:clear-resets Flush expired password reset tokens
  cache
  cache:clear      Flush the application cache
  cache:forget     Remove an item from the cache
  cache:table      Create a migration for the cache database table
  config
  config:cache     Create a cache file for faster configuration loading
  config:clear    Remove the configuration cache file
  db
  db:seed         Seed the database with records
  debugbar
  debugbar:clear  Clear the Debugbar Storage
  event
  event:generate  Generate the missing events and listeners based on registration
  key
  key:generate    Set the application key
  krlove
  krlove:generate:model
  make
  make:auth       Scaffold basic login and registration views and routes
  make:command    Create a new Artisan command
```



■ Consola de Laravel: Tinker

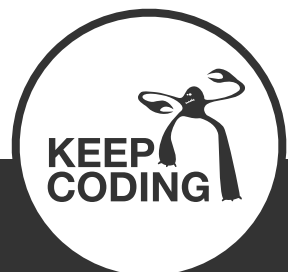
- Consola más avanzada PsySH
- "REPL" o "Read-Eval-Print-Loop"

```
php artisan tinker
```

- Podemos usarlo para resetear una contraseña:

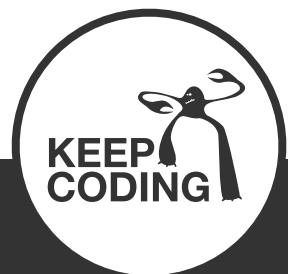
```
$user =  
App\User::where('email', xxx)->first();  
$user->password = Hash::make('newPass');  
$user->save();
```

```
1. php  
[code] 😊 $ php artisan tinker  
Psy Shell v0.8.3 (PHP 7.1.4 - cli) by Justin Hileman  
>>> echo "Hello world"  
Hello world  
=> null  
>>> $users = App\User::all();  
=> Illuminate\Database\Eloquent\Collection {#763  
    all: [],  
}  
>>> factory(App\User::class)->create();  
=> App\User {#802  
    name: "Ms. Brooklyn Rippin",  
    email: "pgerlach@example.net",  
    updated_at: "2017-05-05 08:09:43",  
    created_at: "2017-05-05 08:09:43",  
    id: 1,  
}  
>>> $users = App\User::all();  
=> Illuminate\Database\Eloquent\Collection {#797  
    all: [  
        App\User {#799  
            id: 1,  
            name: "Ms. Brooklyn Rippin",  
            email: "pgerlach@example.net",  
            created_at: "2017-05-05 08:09:43",  
            updated_at: "2017-05-05 08:09:43",  
        },  
    ],  
}
```



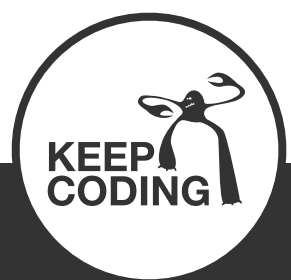
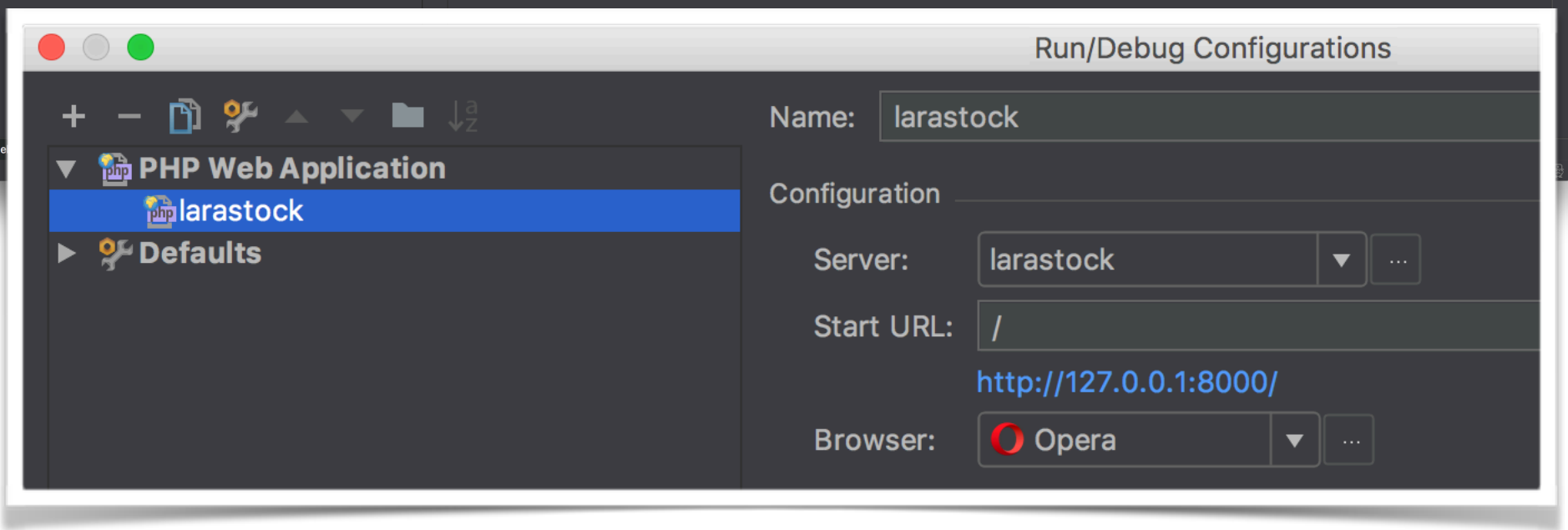
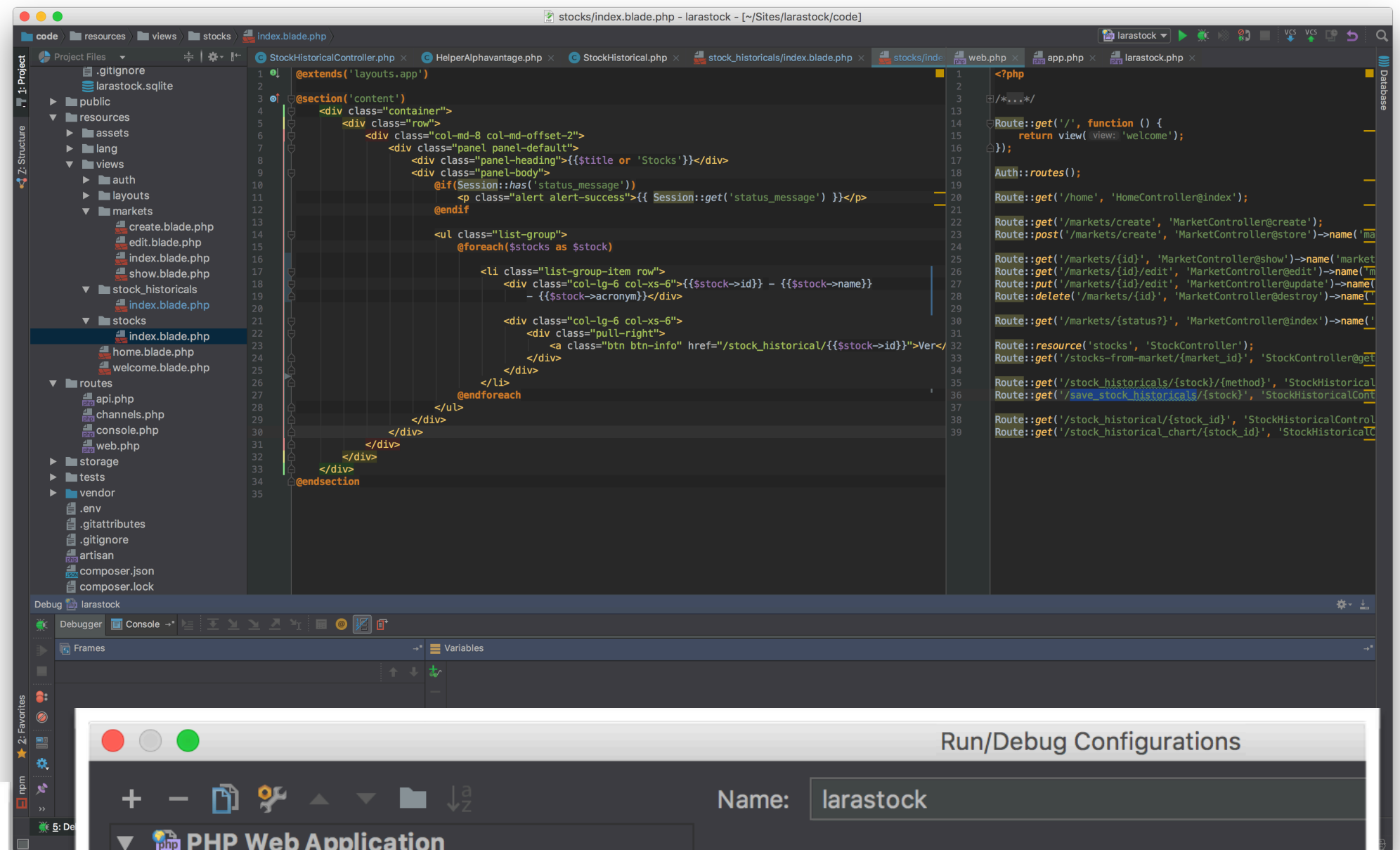
■ IDEs

- PhpStorm
- Eclipse
- Aptana Studio
- Netbeans
- Zend Studio
- PhpEdit
- Notepad++
- Sublime Text
- Vim



IDE: PhpStorm

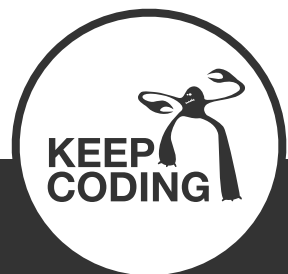
- Crear proyecto
- Tipos de carpetas
- Preferencias
- Configurar Xdebug





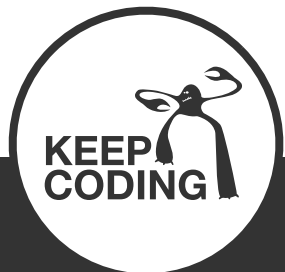
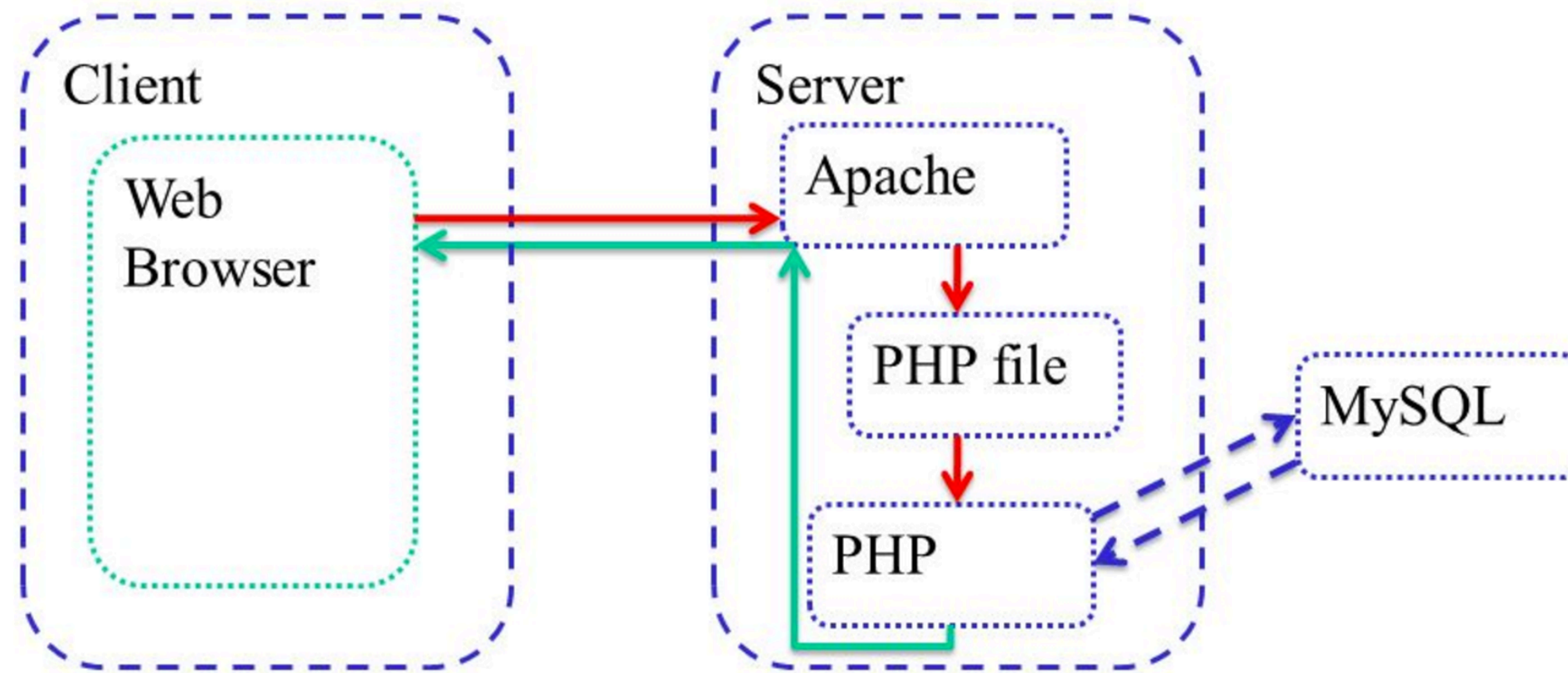
■ 3. Visualización en navegador

Del código a la web



■ Lenguaje interpretado

- No hay que compilar el código. Internamente se compila a un bytecode intermedio que luego interpreta el servidor.
- Al navegador sólo llega un html



■ Motores de plantillas. Blade

- Usa la extensión .blade.php y se almacenan en resources/views
- No impide el uso de PHP
- Tiene herencia de plantillas y secciones

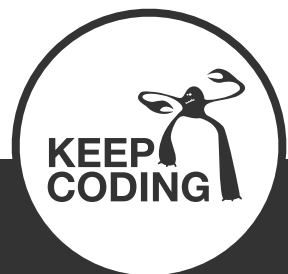
```
@extends ('layouts.app')
@section('title', 'Page Title') => {{ $title }}
@section('content')
    <p>This is my body content.</p> => @yield('content')
@endsection
@if, @elseif, @else, @endif
@for, @foreach, @while
Unescaped data => Hello, {!! $name !!}.
Execute php => @php, @endphp
```





■ 4. Larastock

Nuestro primer proyecto en Laravel



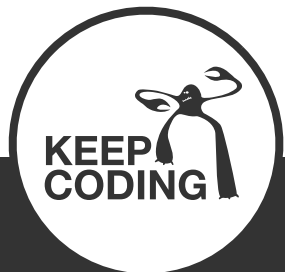
■ Análisis técnico en bolsa

- Media Móvil Simple (SMA): Es una media aritmética de “n” días o períodos.
 - Las más usadas son la de 6, 70 y 200

`MA6 > MA70 > MA200 → Tendencia Alcista`

`MA6 < MA70 < MA200 → Tendencia Bajista`

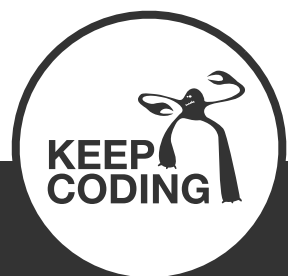
- Posición largos: entrar cuando MA6 cruza al alza a MA70 y salir cuando lo cruza a la baja



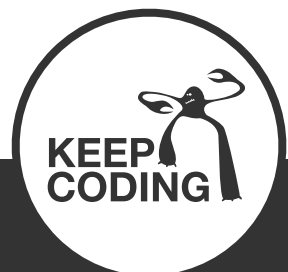
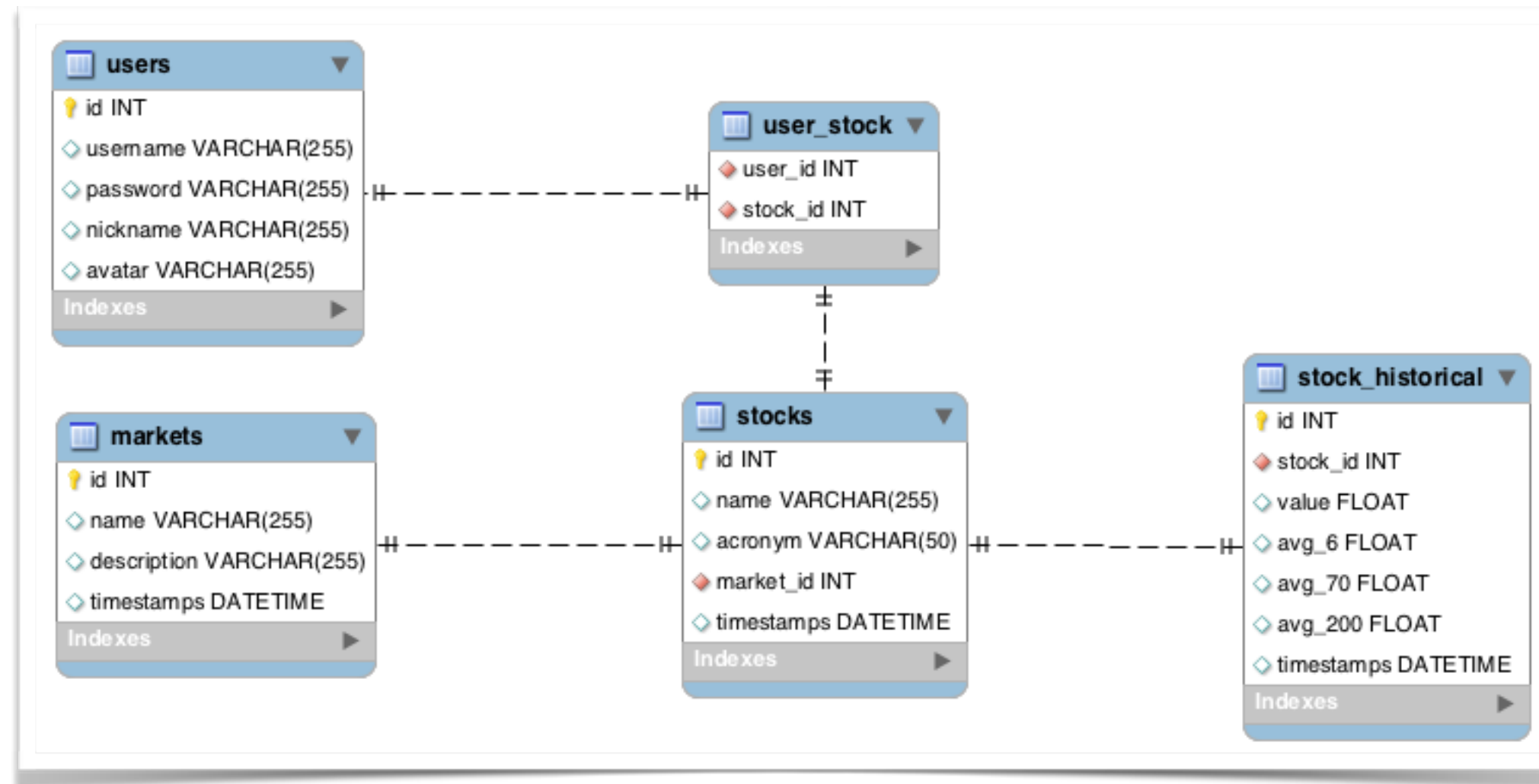
Tendencia alcista vista con SMA



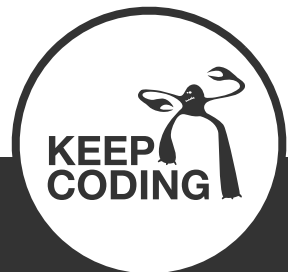
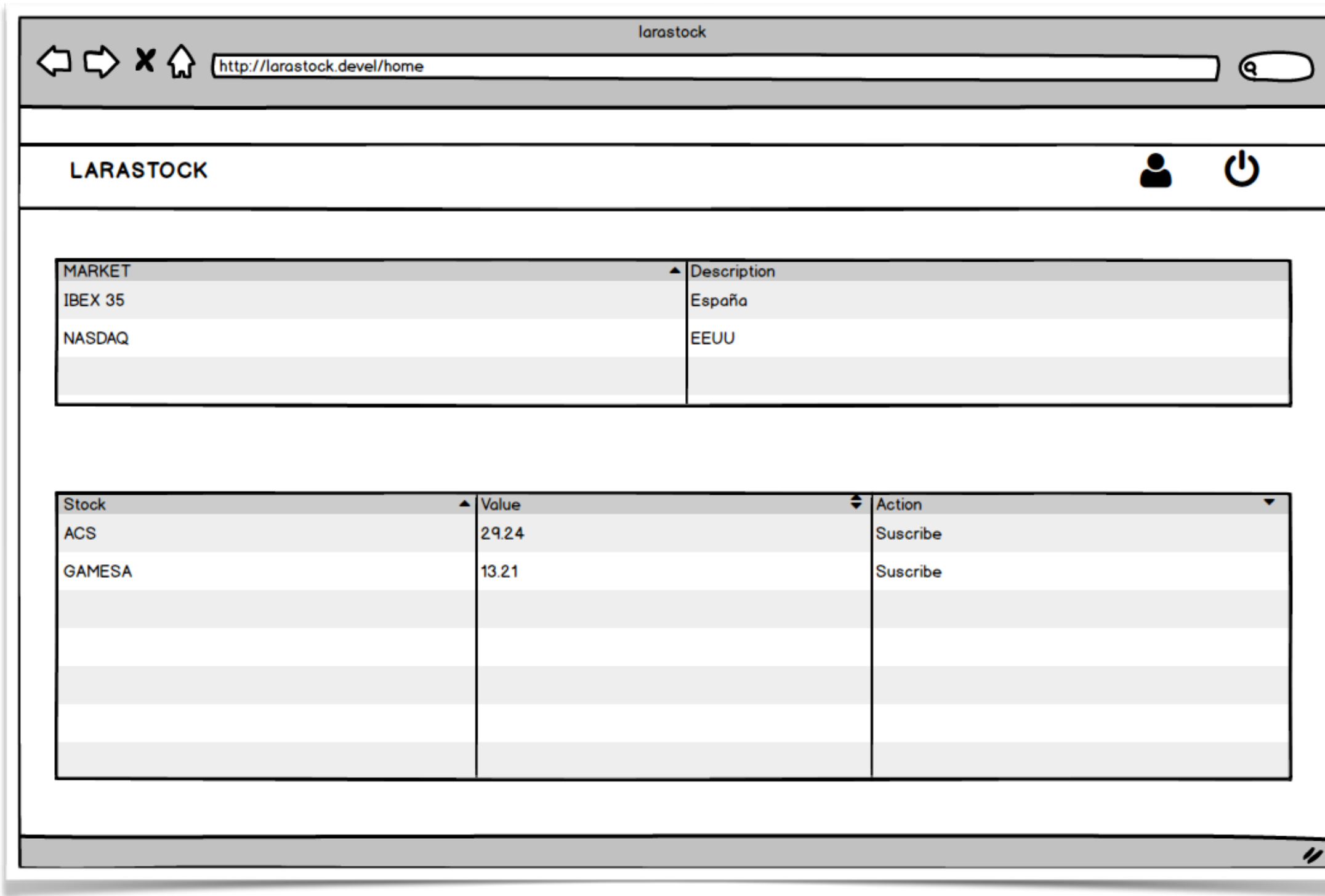
Tendencia bajista vista con SMA



■ Estructura de la base de datos



Wireframes: home



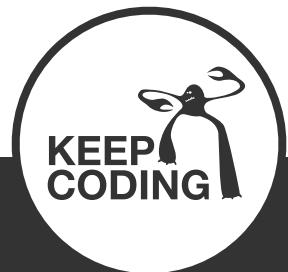
Wireframes: my_stocks

larastock - My stocks

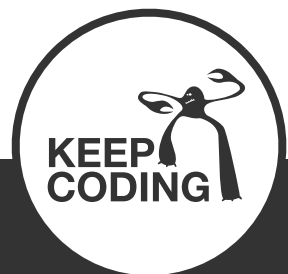
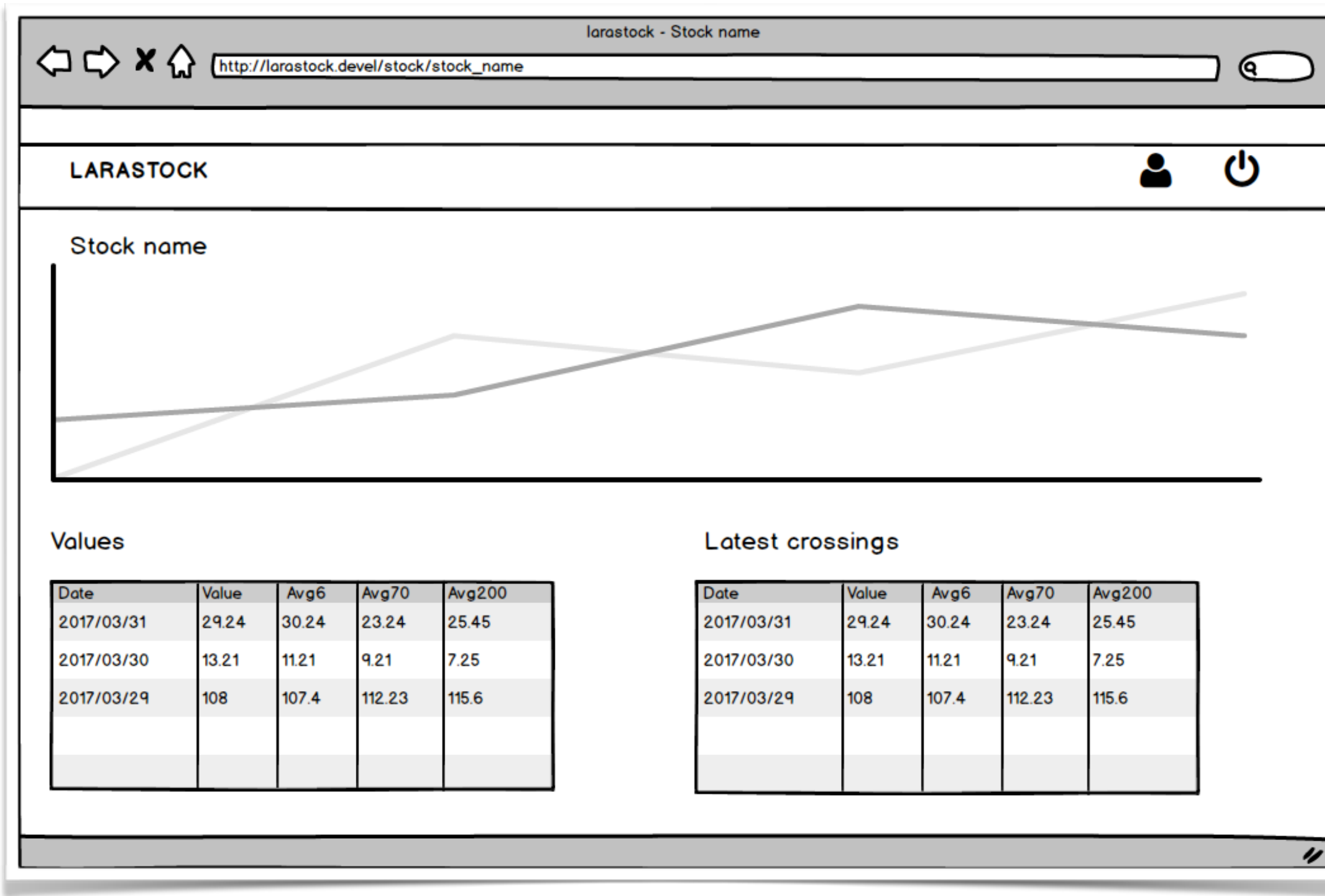
http://larastock.devel/my_stocks

LARASTOCK

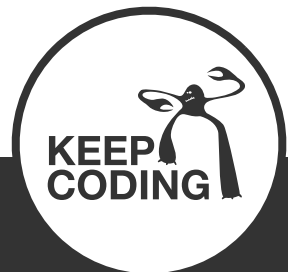
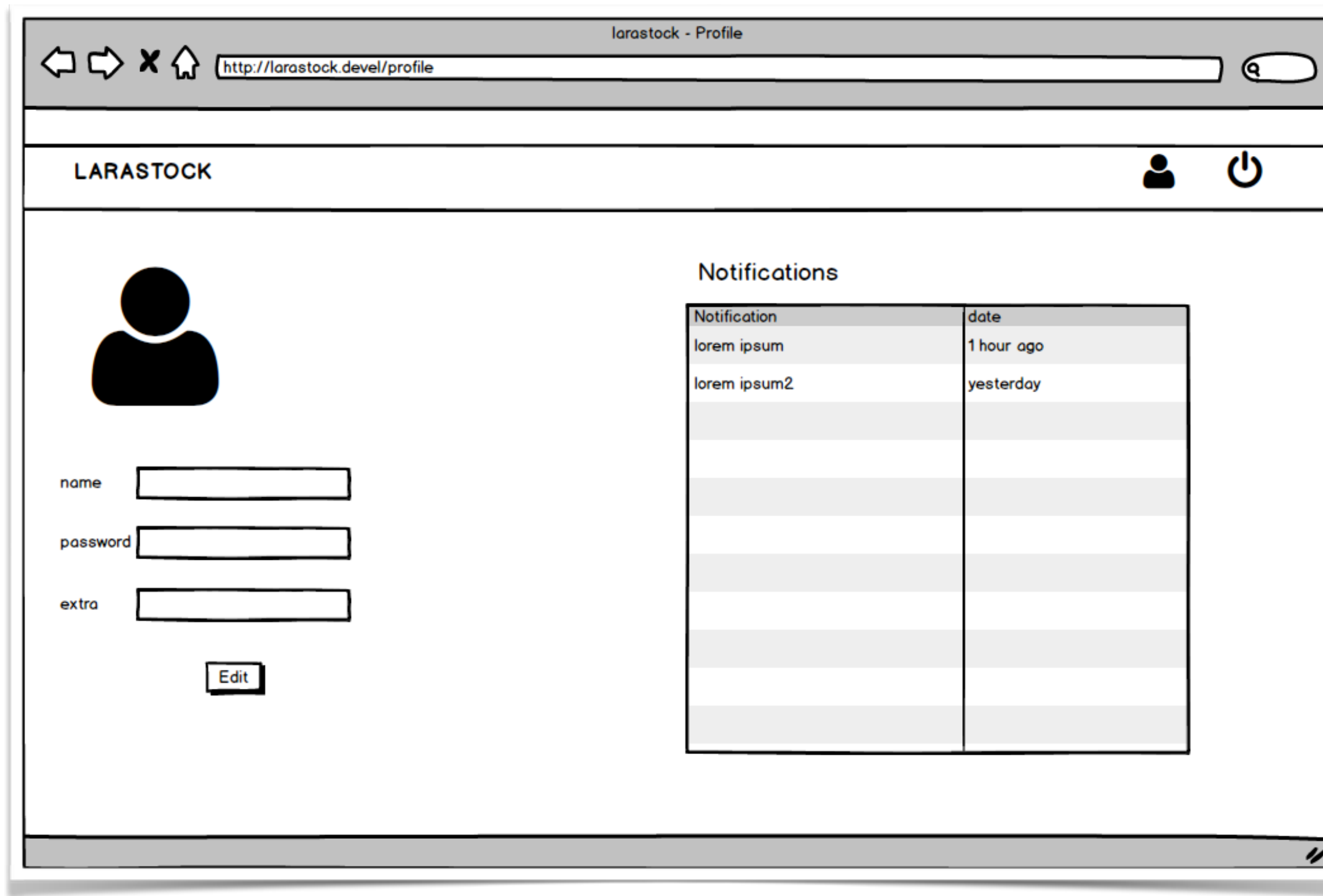
Stock	Value	Avg6	Avg70	Avg200	Action
ACS	29.24	30.24	23.24	25.45	Ununsubscribe
GAMESA	13.21	11.21	9.21	7.25	Ununsubscribe
INDITEX	108	107.4	112.23	115.6	Ununsubscribe



Wireframes: stock_name



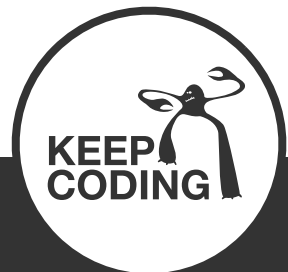
Wireframes: profile





■ 5. PHP Básico

Empecemos por el “Hello world”



■ Escribir por pantalla

- Echo: comillas simples o dobles (interpretadas)

```
$name = `Grokk`;  
echo `Hello world $name`; // Hello world $name  
echo "Hello world $name"; // Hello world Grokk
```

- Print_r y var_dump

```
print_r($array_var)  
var_dump($array_var)
```

- Comentarios
 - línea o bloque

```
// línea comentada  
/*  
bloque  
*/
```



Variables

- Las variables no tienen un tipado estricto ni necesitan ser declaradas antes de usarlas.
- Aunque inicializarlas no hace daño a nadie

```
$a = 1;  
$b = 2;  
$c = $a + $b;  
echo $c;
```

- En PHP 5 ya existía la opción de especificar el tipo en la declaración de los parámetros de funciones.
 - PHP 7 ha introducido los tipos escalares: int, float, string y bool.
 - También la declaración del tipo de devolución

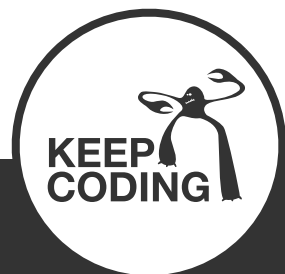
```
function getTotal(float $a, float $b) : float {}
```



■ Constantes

- Permiten declarar un valor fijo.
- No pueden volver a definirse.
- Por convención se escriben en mayúsculas

```
define('MY_CONSTANT', 1);  
echo MY_CONSTANT;  
  
define('DEBUG', false);
```



Operadores, comparación y lógica

Operadores de comparación

Ejemplo	Nombre	Resultado
<code>\$a == \$b</code>	Igual	TRUE si <code>\$a</code> es igual a <code>\$b</code> después de la manipulación de tipos.
<code>\$a === \$b</code>	Idéntico	TRUE si <code>\$a</code> es igual a <code>\$b</code> , y son del mismo tipo.
<code>\$a != \$b</code>	Diferente	TRUE si <code>\$a</code> no es igual a <code>\$b</code> después de la manipulación de tipos.
<code>\$a <> \$b</code>	Diferente	TRUE si <code>\$a</code> no es igual a <code>\$b</code> después de la manipulación de tipos.
<code>\$a !== \$b</code>	No idéntico	TRUE si <code>\$a</code> no es igual a <code>\$b</code> , o si no son del mismo tipo.
<code>\$a < \$b</code>	Menor que	TRUE si <code>\$a</code> es estrictamente menor que <code>\$b</code> .
<code>\$a > \$b</code>	Mayor que	TRUE si <code>\$a</code> es estrictamente mayor que <code>\$b</code> .
<code>\$a <= \$b</code>	Menor o igual que	TRUE si <code>\$a</code> es menor o igual que <code>\$b</code> .
<code>\$a >= \$b</code>	Mayor o igual que	TRUE si <code>\$a</code> es mayor o igual que <code>\$b</code> .
<code>\$a <=> \$b</code>	Nave espacial	Un <u>integer</u> menor que, igual a, o mayor que cero cuando <code>\$a</code> es respectivamente menor que, igual a, o mayor que <code>\$b</code> . Disponible a partir de PHP 7.
<code>\$a ?? \$b ?? \$c</code>	Fusión de null	El primer operando de izquierda a derecha que exista y no sea NULL . NULL si no hay valores definidos y no son NULL . Disponible a partir de PHP 7.

```
$compare = 2 <=> 1  
2 < 1? return -1  
2 = 1? return 0  
2 > 1? return 1
```



■ Arrays

- Array: tipo de dato que asocia valores con claves
 - Indexado:

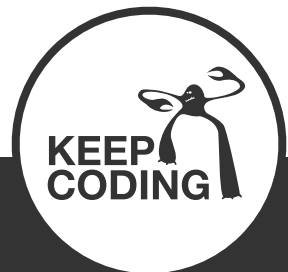
```
$a = array(1, 2, 3)
echo $a[0];
```

- Asociativo:

```
$a = array('first' => 1, 'second' => 2);
echo $a['first'];
```

- Multidimensional:

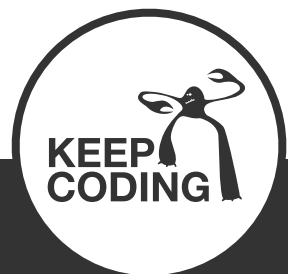
```
$a = array(array(1, 2), array(3, 4));
echo $a[0][0];
```



Objetos

```
$obj1 = new stdClass; // Instantiate stdClass object  
$obj2 = new class{}; // Instantiate anonymous class  
$obj3 = (object)[]; // Cast empty array to object
```

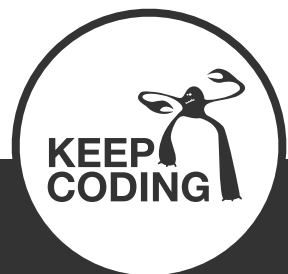
```
$obj = new stdClass();  
$obj->name = "Jane";  
$obj->surname = "Doe";
```





■ 6. PHP Lógica

if else



■ Estructuras de control

- if, else, while, for, foreach, switch

```
if ($a < $b) {  
    echo 'a mayor que b';  
} else {  
    echo 'else';  
}
```

```
while(true) { /* bucle infinito */  
for ($i=0;$i<5;$i++) {}  
foreach ($array as $key => $value) {}  
switch ($variable) {  
    case x:  
        break  
}
```



■ Funciones

Bloque de parámetros que podemos usar repetidamente

```
function nameOfTheFunction(string $param, $debug = false) {  
    $output = 'testing';  
    if (!$debug) {  
        $output = 'not debug';  
    }  
    return $output;  
}
```

Funciones anónimas (desde PHP7)

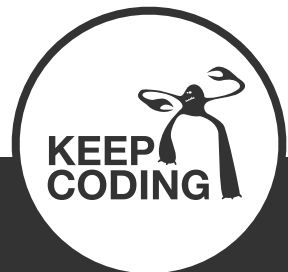
```
$greeting = function($name) {  
    echo "Hi $name";  
}
```



■ Variables predefinidas

- `$_GLOBALS`: Variables disponibles en ámbito local
- `$_SERVER`: Información del entorno del servidor y de ejecución
- `$_GET`: Variables HTTP GET
- `$_POST`: Variables POST de HTTP
- `$_SESSION`: Variables de sesión
- `$_COOKIE`: Cookies HTTP

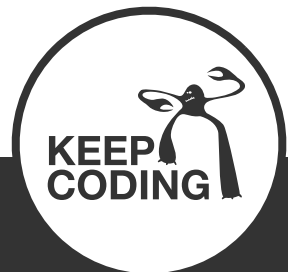
```
var_dump($_SERVER);
```





■ 7. Errores y excepciones

Basta ya de usar print_r



■ Exception

- Una excepción se usa para cambiar el flujo de ejecución cuando ocurre un error
- try/catch/finally

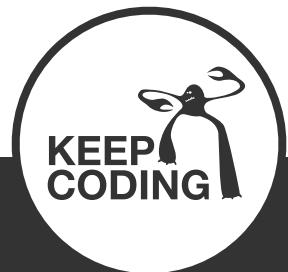
```
//var_dump($_SESSION);  
try {  
    var_dump($_SESSION);  
} catch (ErrorException $e) {  
    //dd($e);  
    echo $e->getMessage();  
}
```





■ 8. Bases de datos

Mind if I join you?



■ Elección base de datos

- La clase Schema permite que Laravel sea agnóstico de su base de datos
- Se configura en el fichero config/database.php
- Y a su vez ese fichero puede usar el fichero por entornos .env

```
'default' =>
env('DB_CONNECTION', 'mysql')
```

```
'connections' => [
    'sqlite' => [
        'driver' => 'sqlite',
        'database' => env('DB_DATABASE', database_path('database.sqlite')),
        'prefix' => '',
    ],
    'mysql' => [
        'driver' => 'mysql',
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => 'utf8mb4',
        'collation' => 'utf8mb4_unicode_ci',
        'prefix' => '',
        'strict' => true,
        'engine' => null,
    ],
    'pgsql' => [
        'driver' => 'pgsql',
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '5432'),
        'database' => env('DB_DATABASE', 'forge'),
        'username' => env('DB_USERNAME', 'forge'),
        'password' => env('DB_PASSWORD', ''),
        'charset' => 'utf8',
        'prefix' => '',
        'schema' => 'public',
        'sslmode' => 'prefer',
    ],
],
```

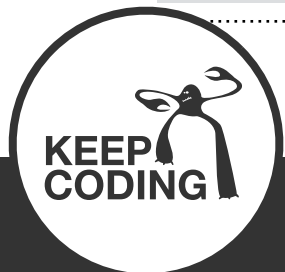


■ Fichero configuración por entorno: .env

- Fichero oculto por entorno.
- No se comitea para que las claves no estén en el repositorio
- Viene en el .gitignore inicial
- Artisan sólo lo lee al arrancar. Si se cambia algo hay que reiniciar el servidor
- Se pueden definir constantes nuevas y leerlas con:

```
env('NEW_NAME', 'default_value')
```

```
.env
1 APP_ENV=local
2 APP_KEY=base64:SoBCB8kt3CGsgSSo7I3a9tj3/0xs6kTsmud6/1Q4jYI=
3 APP_DEBUG=true
4 APP_LOG_LEVEL=debug
5 APP_URL=http://localhost
6
7 DB_CONNECTION=mysql
8 DB_HOST=localhost
9 DB_PORT=3306
10 DB_DATABASE=larastock
11 DB_USERNAME=larastock
12 DB_PASSWORD=larastock
13 DB_SOCKET='/tmp/mysql.sock'
14
15 BROADCAST_DRIVER=log
16 CACHE_DRIVER=file
17 SESSION_DRIVER=file
18 QUEUE_DRIVER=sync
19
20 REDIS_HOST=127.0.0.1
21 REDIS_PASSWORD=null
22 REDIS_PORT=6379
23
24 MAIL_DRIVER=smtp
25 MAIL_HOST=smtp.mailtrap.io
26 MAIL_PORT=2525
27 MAIL_USERNAME=null
28 MAIL_PASSWORD=null
29 MAIL_ENCRYPTION=null
30
31 PUSHER_APP_ID=
32 PUSHER_APP_KEY=
33 PUSHER_APP_SECRET=
34
```



■ ORM Laravel: Eloquent

- Object-relational mapping
- Nos permite abstraernos de la base de datos y hacer las consultas más “amigables” basándose en Modelos
- Select

```
$users = App\User::all();  
$user = App\User::where('name', $name)->get();
```

- Insert/Update

```
$user = new User; // $user = App\User::find($id);  
$user->name = $request->name;  
$user->save();
```

- Delete

```
$user->delete();
```



■ Migraciones

- Control de versiones para la base de datos
 - Crear migración

```
php artisan make:migration create_users_table
```

- Crear tabla

```
--create=users
```

- Editar tabla existente

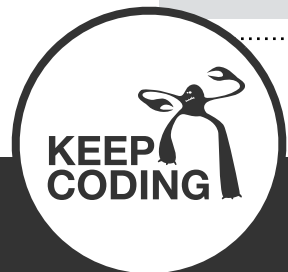
```
--table=users
```

- Ejecutar

```
php artisan migrate
```

- Deshacer

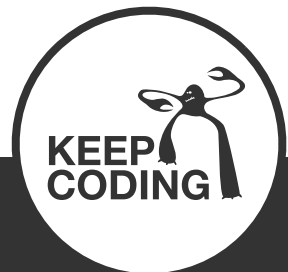
```
php artisan migrate:rollback
```





■ 9. Tests

En PHP también se hacen tests



PHPUnit

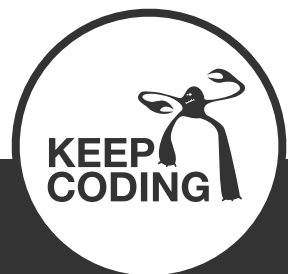
- Framework para hacer test en PHP
- Viene integrado en Laravel y se configura en el fichero phpunit.xml

```
php artisan make:test UserTest --unit

class ExampleTest extends TestCase{
    public function testBasicTest() {
        $this->assertTrue(true);
    }
}
```

```
phpunit.xml
<?xml version="1.0" encoding="UTF-8"?>
<phpunit backupGlobals="false"
    backupStaticAttributes="false"
    bootstrap="bootstrap/autoload.php"
    colors="true"
    convertErrorsToExceptions="true"
    convertNoticesToExceptions="true"
    convertWarningsToExceptions="true"
    processIsolation="false"
    stopOnFailure="false">
    <testsuites>
        <testsuite name="Feature Tests">
            <directory suffix="Test.php">./tests/Feature</directory>
        </testsuite>

        <testsuite name="Unit Tests">
            <directory suffix="Test.php">./tests/Unit</directory>
        </testsuite>
    </testsuites>
    <filter>
        <whitelist processUncoveredFilesFromWhitelist="true">
            <directory suffix=".php">./app</directory>
        </whitelist>
    </filter>
    <php>
        <env name="APP_ENV" value="testing"/>
        <env name="CACHE_DRIVER" value="array"/>
        <env name="SESSION_DRIVER" value="array"/>
        <env name="QUEUE_DRIVER" value="sync"/>
        <env name="DB_CONNECTION" value="sqlite"/>
        <env name="DB_DATABASE" value="database/larastock.sqlite"/>
    </php>
</phpunit>
```



■ PHPUnit. Uso

```
phpunit  
phpunit --filter testName
```

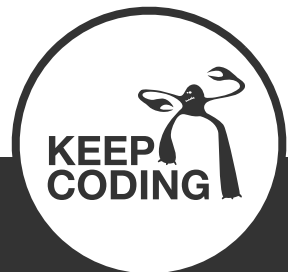
- Si no está instalado a nivel global, dará un error

Opciones:

```
brew install phpunit
```

```
./vendor/bin/phpunit
```

```
composer global require phpunit/phpunit + composer en PAT
```



■ PHPUnit. Base de datos

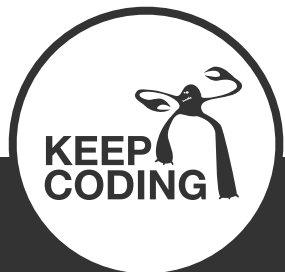
- En el phpunit.xml se puede especificar la base de datos
- No queremos llenar nuestra base de datos con datos de test

```
touch database/larastock.sqlite
```

```
<env name="DB_DATABASE" value="database/larastock.sqlite"/>
```

- Los tests pueden ejecutar las migraciones antes de cada test y eliminar los datos después usando los siguientes traits:

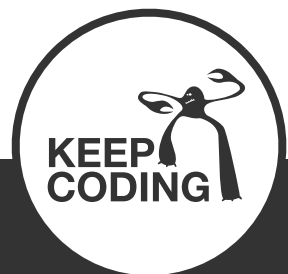
```
use Illuminate\Foundation\Testing\DatabaseTransactions;  
use Illuminate\Foundation\Testing\DatabaseMigrations;
```





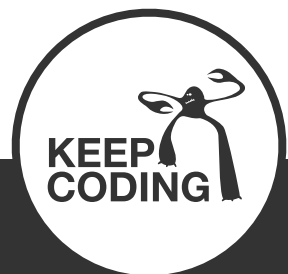
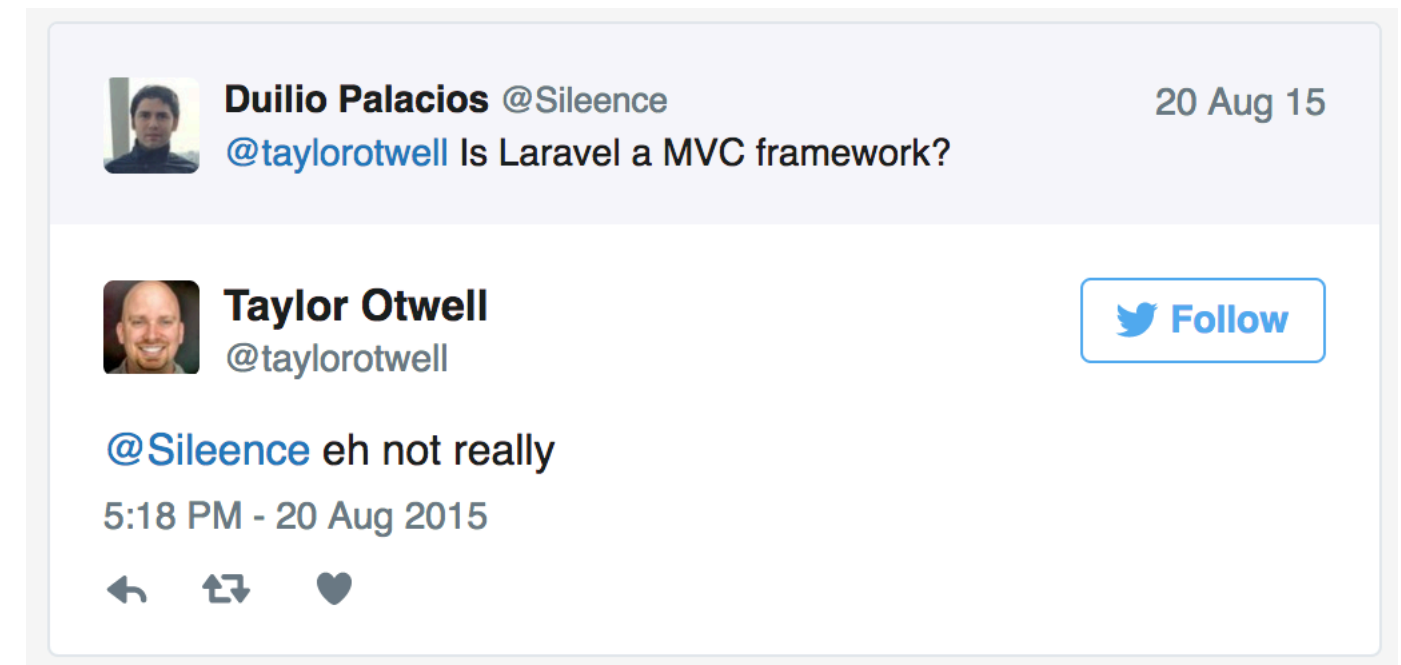
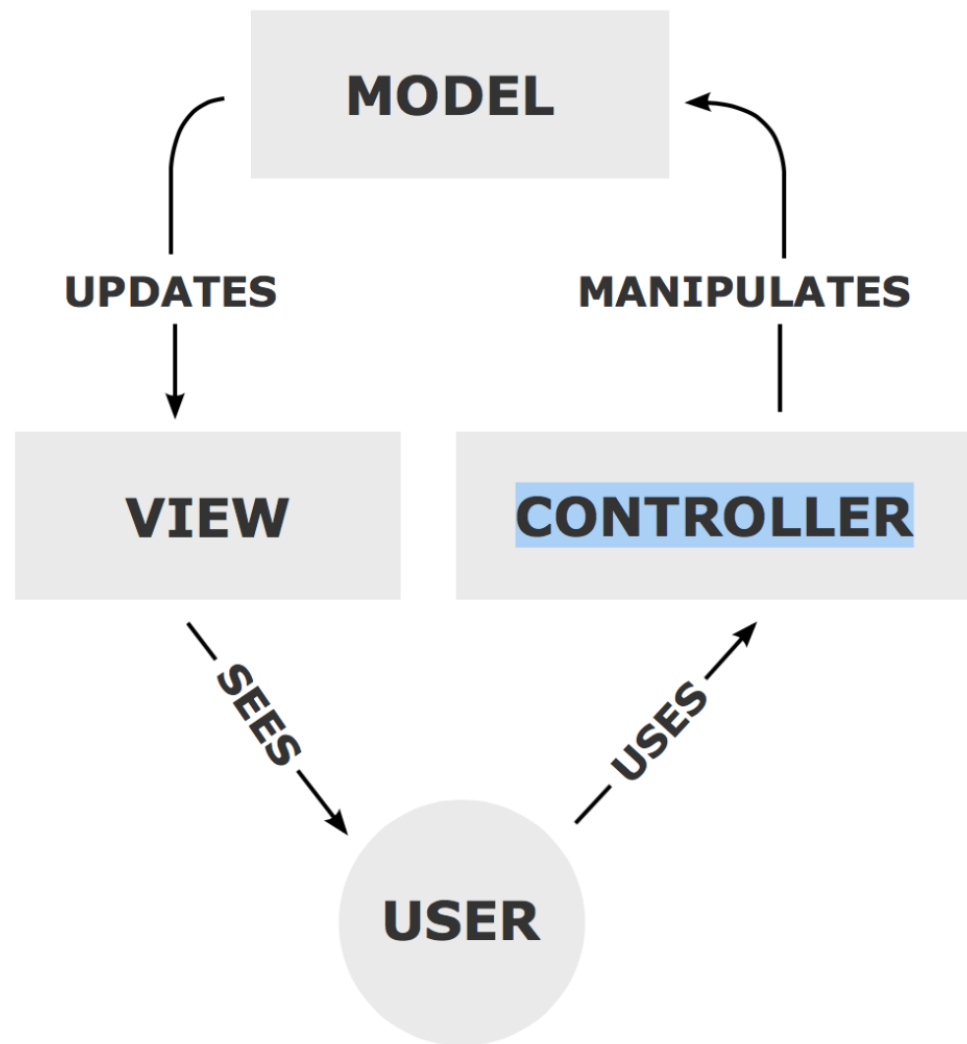
■ 10. Orientación a objetos

Aunque puedas hacer funciones sueltas... mejor usar clases!



MVC

- Modelo-Vista-Controlador



■ Clases

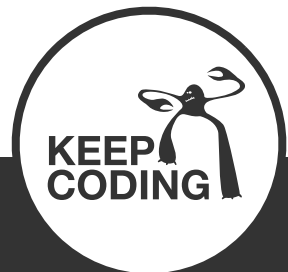
- Es un objeto que nos permite encapsular el estado y la responsabilidad de algo en nuestra aplicación y nos permite interactuar con ello, reusando el código y haciéndolo mantenible.

```
class Lock {  
    private $isLocked = false;  
    public function lock() {  
        $this->isLocked = true;  
        echo 'You locked the Lock';  
    }  
    public function isLocked() {  
        return $this->isLocked;  
    }  
}
```



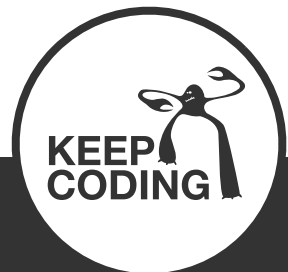
■ Funciones públicas, privadas y static

- Visibilidad de las propiedades o métodos:
 - public: accesible desde cualquier parte
 - protected: accesible sólo desde la clase o heredadas
 - private: accesible sólo desde la clase.
- Propiedades o métodos static: accesibles sin necesidad de instanciar la clase.
 - No puede ser accedida con un objeto de clase instanciado
 - \$this no está disponible dentro



■ Herencia y Traits

- Permite extender una clase y que la subclase herede todos los métodos públicos o protegidos.
- Dichos métodos pueden ser sobrescritos en el hijo.
- No existe la multiherencia; para eso existen los traits
 - Permite reutilizar conjuntos de métodos sobre clases independientes.
 - No se instancian, se utilizan con “use”



Namespaces

- Sirven para encapsular elementos
- Permiten evitar conflictos con clases/funciones/constantes de terceros
- Permiten abreviar nombres de funciones mejorando la legibilidad
- En composer.json podemos determinar el autoload

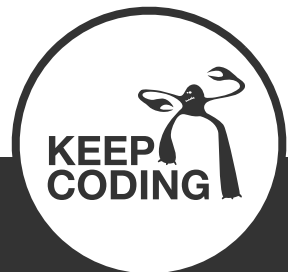
```
new Library/Name/User;  
new Library/Name2/User as User2;  
  
namespace Acme\User  
  
use Library\Name\User;  
new \Library\Name\User;  
new User;
```





■ 11. Funciones específicas de Laravel

No reinventes la rueda



■ Eventos y Listeners

- Laravel permite crear funciones que se quedan a la espera de que ocurran determinados eventos

```
protected $listen = [  
    'App\Events\OrderShipped' => [  
        'App\Listeners\SendShipmentNotification' ] ] ;
```

```
php artisan event:generate
```

Puedes suscribirte a los eventos

```
$events->listen(  
    'Illuminate\Auth\Events>Login',  
    'App\Listeners\UserEventSubscriber@onUserLogin'  
);
```



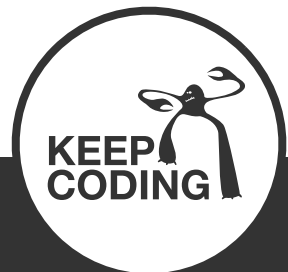
■ Notificaciones

- Funcionalidad de Laravel para avisar de eventos por varios canales
 - Mail
 - SMS
 - Slack
 - Base de datos

```
php artisan make:notification UserCreation
```

Se puede usar el Notify trait y encolarlas para procesarlas más tarde

```
$when = Carbon::now()->addMinutes(10);  
$user->notify((new InvoicePaid($invoice))->delay($when));
```



■ Paquetes oficiales

- Lumen: versión micro-framework
- Passport: Permite crear APIs tanto públicas como privadas
- Cashier: Interfaz para gestionar suscripciones de pago de Stripe y BrainTree
- Envoy: Permite ejecutar tareas en el servidor
- Scout: Permite indexar la base de datos para hacer búsquedas
- Socialite: Autenticación OAuth con Facebook, Twitter, Google, LinkedIn, GitHub y Bitbucket

