

# Tips para programadores y sus sistemas

Cositas que vienen bien cuando uno trabaja en sistemas para hacer nuestros despliegues, en producción, en

- [Cosas de Git](#)
  - [Como eliminar de nuestro repositorio git un fichero o directorio](#)
  - [Como forzar un git pull sobre escribiendo los ficheros locales](#)
  - [Como hacer debug a git para corregir o conocer problemas de conexión](#)
- [Cosas de Laravel](#)
  - [Crear una Clase Helper para un proyecto Laravel](#)
  - [Mailhog como mailtrap para desarrollos con Laravel](#)
  - [Método en TestCase para facilitar los test de usuario logeado](#)
  - [Métodos dump en el proceso de Testing con Laravel](#)
  - [Asignar múltiples variables a la vez en PHP](#)
  - [Configurar Carbon::now\(\) a una fecha para trabajar con tests](#)
  - [Código de estado HTTP para llamadas API](#)
  - [Testing error SQLSTATE\[HY000\]: General error: 1 near "ALTER": syntax error \(SQL: ALTER TABLE](#)
  - [Laravel rescue\(\) helper](#)
  - [Sail, Access can't connect to Mysql](#)
  - [Sail y docker](#)

- [SAGE API 3.1, Laravel Socialite](#)
- [Laravel Filament Admin funciona en Sail, pero no en producción. error 404 en ficheros .js](#)
- [Traducciones no funcionan en Laravel](#)
- [Comprobando el uso del trait](#)
- [Pest, PHPStorm y Laravel Sail](#)
- [Laravel Herd y cosas que no estan documentadas.](#)
- [PHP enums un gran aliado.](#)
- [Manera elegante de obtener el phpinfo\(\) en tu proyecto](#)
- [Instalar ionCube en Laravel Herd en un mac Silicon](#)
- [El cast y su importancia en el modelo Eloquent de Laravel](#)
- [Phpredis en Laravel 10/11](#)
- [Livewire && Laravel Localization: The GET method is not supported for route livewire/update 404](#)
- [Cosas de PHP](#)
  - [Expandir variables :: Sintaxis compleja \(curva\)](#)
  - [Como instalar un fork en un proyecto con composer.](#)
- [Cosas de ElasticSearch \(ELK\)](#)
  - [Elasticsearch no arranca: A process of this unit has been killed by the OOM killer.](#)
  - [Instalando ElasticSearch más Kibana en entorno local](#)
  - [Guia de comandos útiles para un rápido vistazo a Elasticsearch](#)
  - [Elasticsearch y Kibana con Docker](#)
  - [Elasticsearch PHP API: No alive nodes. All the 1 nodes seem to be down.](#)
  - [Conversor de consulta SQL a DSL para Elasticsearch](#)
  - [Snapshots y restore](#)
  - [Backups, snapshost y restore en Elasticsearch 8](#)
  - [Consultas avanzadas de elasticsearch](#)
  - [Instalando ElasticSearch + Kibana en local con Docker](#)
  - [Truncate index](#)
  - [Arranque, Actualización, y cosas de Elastic con Docker](#)
  - [Llamadas en Kibana \(o para usar con cUrl\) para Elasticsearch de uso común](#)
- [Cosas de docker](#)

- [Información de los contenedores docker](#)
- [Wordpress en docker para desarrollo.](#)
- [Proyectos con path public, public\\_html, app en Docker con Apache+PHP-FPM](#)
- [Cosas de Docker](#)
  - [Instalar Postgresql con Postgis](#)

# Cosas de Git

Git, esa herramienta para desarrolladores que tantos dolores de cabeza nos da y tanto amor la tenemos

# Como eliminar de nuestro repositorio git un fichero o directorio

## Introducción

Un error muy común es que algunas veces, con las prisas olvidamos añadir a nuestro `.gitignore` algo que no está en el template con el que trabajamos.

No es correcto dejarnos ciertos ficheros o directorios en el repositorio.

## Como eliminar un fichero o directorio en el git (repositorio)

- Primero un backup de nuestro repositorio local, por favor. Siempre un backup.
- Añade el fichero o directorio al fichero `.gitignore`
- En tu `branch` comprueba que el fichero está fuera del master

```
git checkout master -- .gitignore
git add .
git commit -m 'Update .gitignore'
git pull
```

- Remueve el fichero o directorio del árbol de git

```
git rm --cached -r nombre_de_fichero_o_directorio
git add .
git commit -m 'Remove nombre_de_fichero_o_directorio'
```



```
Installing collected packages: git-filter-repo
```

```
Successfully installed git-filter-repo-2.29.0
```

Añadimos el comando a nuestro `.bashrc` o `.zshrc` su path. En mi caso la instalación se efectuó en `:$HOME/.local/bin`

```
export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH
```

Hacemos un reload de nuestro RC o salimos de la sesión para aplicar los cambios.

```
reload
```

## Limpieza con git-filter-repo

Ahora, es hora de trabajar con el comando para limpiar nuestro repositorio.

```
> git-filter-repo --path myssh.sh --invert-paths --force
Parsed 108 commits
New history written in 0.06 seconds; now repacking/cleaning...
Repacking your repo and cleaning out old unneeded objects
HEAD está ahora en cce4227 Sin especificar
Enumerando objetos: 405, listo.
Contando objetos: 100% (405/405), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (232/232), listo.
Escribiendo objetos: 100% (405/405), listo.
Total 405 (delta 232), reusado 299 (delta 157)
Completely finished after 0.18 seconds.
```

Como veis en la salida, el programa se ha encargado de viajar por la historia de los commits, y realizar la limpieza y cambiar la historia.

Ahora actualizaremos el remoto. Como veremos, los datos de nuestro `.git/config` han sufrido una actualización (que bueno tener backups si no tenemos o conocemos bien ciertas cosas)

```
git add .
git commit -m 'Update gitignore and clean'
[master 7d77a36] Update gitignore and clean
1 file changed, 1 insertion(+), 1 deletion(-)
git push
fatal: No se ha configurado un destino para el push.
```

Puedes o especificar una URL desde la línea de comandos o configurar un repositorio remoto usando

```
git remote add <nombre> <url>
```

y luego haciendo push al nombre del remoto

```
git push <nombre>
```

oops... no pasa nada. Se trata de reconstruir la parte que se ha eliminado de la configuración relativa al repositorio remoto.

```
[remote "origin"]
  url = git@gitlab.castris.com:root/nombre_del_repo.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
```

Podemos hacerlo editando `.git/config` o vía comando

```
git remote add master git@gitlab.castris.com:root/utilidades.git
git remote add origin git@gitlab.castris.com:root/utilidades.git
git push -f --set-upstream origin master
Enumerando objetos: 405, listo.
Contando objetos: 100% (405/405), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (157/157), listo.
Escribiendo objetos: 100% (405/405), 401.98 KiB | 100.50 MiB/s, listo.
Total 405 (delta 232), reusado 405 (delta 232)
remote: Resolving deltas: 100% (232/232), done.
To gitlab.castris.com:root/utilidades.git
+ 1ba0d77...cd97f4c master -> master (forced update)
```

Ahora ya podemos estar tranquilos. Hemos borrado los datos sensitivos de nuestro repositorio.

## Agradecimientos y enlaces interesantes

- [How to remove the .idea folder from git](#)
- [Remove folder and its contents from git/GitHub's history](#)
- [How do you install git-filter-repo?](#)



- [Remove file from git repository history](#)

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Como forzar un git pull sobre escribiendo los ficheros locales

## Introducción

Algunas veces con git, puede pasar que hay mínimos cambios en local que no son importantes, y queremos hacer un **git pull** para sobre escribir, pero no nos deja (por seguridad)

## Cómo forzar un git pull para sobreescribir los ficheros locales

Es importante entender que cualquier cambio en los ficheros locales, se perderá. Todo cambio, con o sin la opción --hard de los commits locales que no hayan sido subidos, se perderán.

---

Primero, traer todas las ramas del origen

```
git fetch -all
```

Después hacer un backup de la rama actual en local

```
git branch backup-master
```

Aquí hay dos opciones:

```
git reset --hard origin/master
```

O hacerlo en otra rama

```
git reset --hard origin/<branch_name>
```

## Explicación

`git fetch` descarga actualizado del remoto sin intentar un `merge` o un `rebase`

Después `git reset` reiniciará la rama master que tú quieres tomar.

## Mantener los commits locales pendientes

Es una buena idea muchas veces mantener los cambios locales creando una rama desde el master antes de hacer el reset.

```
git checkout master  
git branch new-branch-to-save-current-commits  
git fetch --all  
git reset --hard origin/master
```

Después de esto, todas las confirmaciones (commits) se mantendrán en la nueva rama, `new-branch-to-save-current-commits`

“ Como no, en este tipo de acciones es absolutamente necesario, hacer un backup antes de nada.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Como hacer debug a git para corregir o conocer problemas de conexión

## Mode debug en comando git en red

Alguna vez, se te puede quedar un comando git remoto (pull, push, etc) colgado, como si no funcionara ñla conexión remota o hubiera problemas ssh.

## Solución

### [GIT\\_TRACE\\_CURL](#)

```
$ GIT_TRACE_CURL=true git clone https://.....  
...  
17:42:56.835334 http.c:717      <= Recv data: 3 39;....L....qE...5.jl..~..=.....S..6.>Z\!.  
17:42:56.835348 http.c:717      <= Recv data: .....Z%...;0019.h.@.l.'....E.....0006..003f.Total 157 (d  
17:42:56.835355 http.c:717      <= Recv data: elta 32), reused 88 (delta 28), pack-reused 570006..0000  
remote: Total 157 (delta 32), reused 88 (delta 28), pack-reused 57  
Receiving objects: 100% (157/157), 1.33 MiB | 3.02 MiB/s, done.  
Resolving deltas: 100% (48/48), done.  
17:42:56.840003 http.c:729      == Info: Connection #0 to host gitlab.xxxx.xxx left intact
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias

de seguridad. El contenido el contenido se entrega, tal y como esta, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Cosas de Laravel

Algunos tips de Laravel

# Crear una Clase Helper para un proyecto Laravel

## Introducción

Muchas veces necesitamos ciertas funciones o métodos, para nuestros proyectos, que por su repercusión o repetición, puede ser interesante tenerlos agrupados. Solemos llamarlos Helpers.

Podemos hacerlo de tres formas Crear un fichero de funciones (el más común entre los bloggers de Laravel) con carga mediante autoload Creación de una Clase estática (no muy común pero más fina y menos propensa a choques) Creación de una paquete, que sería interesante si tuviéramos muchísimos helpers en distintas clases y que pudieran ser compartido por multitud de nuestros proyectos o de otros programadores Aquí vamos a mostrar los dos primeros.

## helpers.php

Crearemos un fichero cuyo lugar y nombre podría ser `app/helpers.php` con el contenido de abajo.

## Crear un archivo helpers.php

### Método incorrecto

No se la de veces que lo habré visto, y me parece horrible y alejado de las buenas prácticas.

```
<?php

function asString($data)
{
    $json = asJSON($data);

    return wordwrap($json, 76, "\n  ");
}
```

```

}

function asJSON($data)
{
    $json = json_encode($data);
    $json = preg_replace('/(["\\}]})([,:])(["\\{])/',' $1$2 $3', $json);

    return $json;
}

```

## Método correcto

Ya que no usamos una clase sino un archivo tipo include que cargaremos mediante un autoload, para evitar problemas de duplicidad de nombres con las funciones de PHP, lo correcto es hacerlo como el código de abajo

```

<?php

if (!function_exists('asString'))
{
    function asString($data)
    {
        $json = asJSON($data);

        return wordwrap($json, 76, "\n ");
    }
}

if (!function_exists('asJson'))
{
    function asJSON($data)
    {
        $json = json_encode($data);
        $json = preg_replace('/(["\\}]})([,:])(["\\{])/',' $1$2 $3', $json);

        return $json;
    }
}

```



# Carga del archivo

Editamos nuestro composer.json en la sección `autoload` añadiendo o creando la sección `files`

```
"autoload": {  
    "psr-4": {  
        "App\\": "app/",  
        "Database\\Factories\\": "database/factories/",  
        "Database\\Seeders\\": "database/seeders/"  
    },  
    "files": [  
        "app/helpers.php"  
    ]  
},
```

## Actualización del autoload de la app

```
composer dump-autoload
```

## Uso

El uso es sencillo, ya que se le llama como si fuera una función nativa de PHP.

```
$headerData = [  
    'category' => 'develop',  
    'unique_args' => [  
        'var_1' => 'abc'  
    ]  
];  
  
$header = asString($headerData);
```

## Crear una clase estática

La creación de una clase estática, nos permite más seguridad, algo más de estandarización y para trabajar en grupo, y el camino a la creación de nuestro propio paquete de Helpers.

# Crear el fichero de clase Helper

En mi ejemplo uso el directorio Helpers dentro de App porque tengo más clases de helpers en un proyecto largo `app/Helpers/MailHelpers`

```
<?php

namespace App\Helpers;

class MailHelpers {
    public static function asString($data)
    {
        $json = self::asJson($data);

        return wordwrap($json, 76, "\n ");
    }

    public static function asJson ($data)
    {
        $json = json_encode($data);
        $json = preg_replace('/(["\]}])([,:])(["\{])/',' $1$2 $3', $json);

        return $json;
    }
}
```

De esta forma no necesitamos realizar ninguna modificación en nuestro fichero `composer.json`, ya que la carga se produce conforme al PSR-4, y es simplemente una clase más de tipo estático.

## Uso

Este ejemplo es del uso de la clase `MailHelpers` en una clase `Mail`.

```
<?php

namespace App\Mail;

use App\Helpers\MailHelpers;

...
```

```
public function build()
{
    $headerData = [
        'category' => 'develop',
        'unique_args' => [
            'var_1' => 'abc'
        ]
    ];

    $header = MailHelpers::asString($headerData);

    ...

    ...
}
```

# Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Mailhog como mailtrap para desarrollos con Laravel

## Introducción

Generalmente uso mailtrap.io para comprobar los correos ya que entre otras herramientas tiene **HTML Check** pero cuando se trata de trabajos en su inicio que no requieren de esto, y prima la portabilidad, prefiero usar Mailhog, un capturador de correo desarrollado en **Go**.

## Instalación en Ubuntu

### Descarga y conversión a ejecutable

“ Revisar siempre la versión disponible, ya que una entrada de blog o wiki puede no estar actualizada

```
$ wget https://github.com/mailhog/MailHog/releases/download/v1.0.0/MailHog_linux_amd64
$ sudo cp MailHog_linux_amd64 /usr/local/bin/mailhog
$ sudo chmod +x /usr/local/bin/mailhog
```

### Crear un servicio para Mailhog (systemd)

“ Se hace uso del comando whoami para cargar el servicio para nuestro usuario. Debemos verificar que lo hizo bien.

```
$ sudo tee /etc/systemd/system/mailhog.service <<EOL
[Unit]
Description=Mailhog
After=network.target

[Service]
User=$(whoami)
ExecStart=/usr/bin/env /usr/local/bin/mailhog > /dev/null 2>&1 &

[Install]
WantedBy=multi-user.target

EOL
```

## Verificar

```
$ sudo cat /etc/systemd/system/mailhog.service
[Unit]
Description=Mailhog
After=network.target

[Service]
User=abkrim
ExecStart=/usr/bin/env /usr/local/bin/mailhog > /dev/null 2>&1 &

[Install]
WantedBy=multi-user.target
```

## Activar y habilitar para arranque con el sistema

Es aconsejable siempre que modifiquemos algo en el systemd hacer un reload del demonio.

```
$ sudo systemctl daemon-reload
```

### Distintas acciones

```
$ sudo systemctl start mailhog.service
$ sudo systemctl enable mailhog.service
$ sudo systemctl status mailhog.service
```

## Configurar php.ini

En mi caso es para PHP-FPM y multi versión, por lo que necesito añadirlo a cada uno, para el fpm y para el cli.

```
$ sudo sed -i "s;/sendmail_path.*/sendmail_path='/usr/local/bin/mailhog sendmail abkrim@nox.test'/"  
/etc/php/8.0/fpm/php.ini  
  
$ sudo sed -i "s;/sendmail_path.*/sendmail_path='/usr/local/bin/mailhog sendmail abkrim@nox.test'/"  
/etc/php/8.0/cli/php.ini  
  
$ sudo sed -i "s;/sendmail_path.*/sendmail_path='/usr/local/bin/mailhog sendmail abkrim@nox.test'/"  
/etc/php/7.4/fpm/php.ini  
  
$ sudo sed -i "s;/sendmail_path.*/sendmail_path='/usr/local/bin/mailhog sendmail abkrim@nox.test'/"  
/etc/php/7.4/cli/php.ini
```

Necesito un reload de php-fpm

```
$ sudo systemctl restart php8.0-fpm.service  
$ sudo systemctl restart php7.4-fpm.service
```

# Ver Mailhog en el navegador

<http://localhost:8025/>

Mailhog localhost

Image not found or type unknown

# Configuracion para Laravel

Editamos el fichero .env

El puerto por defecto es 1025 MAIL\_FROM\_ADDRESS es requerido

```
MAIL_MAILER=smtp  
MAIL_HOST=localhost  
MAIL_PORT=1025  
MAIL_USERNAME=null  
MAIL_PASSWORD=null  
MAIL_ENCRYPTION=null  
MAIL_FROM_ADDRESS=abkrim@nox.local  
MAIL_FROM_NAME="${APP_NAME}"
```

# Enlace

- [Install & Configure MailHog](#)

# Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Método en TestCase para facilitar los test de usuario logeado

## Desarrollo

En los tests mucha s veces necesitamos realziar pruebas como usuario logeado o usuario especifico. Un buen refactor para esta acción repitida es incluirla en la **clase TestCase** de la cual extendemos test en Laravel.

## TestCase

```
public function login(User $user = null): User
{
    $user ??= User::factory()->create();

    $this->actingAs($user);

    return $user;
}
```

Ahora será más fácil escribir nuestros tests usando simplemente `$this->login`

```
// $this->actingAs(User::factory()->create());

$this->login()
```

## Aviso



Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Métodos dump en el proceso de Testing con Laravel

## Introducción

Una de las herramientas que más me gustan de **Laravel** es `dd()`. Una herramienta que permite un volocado con exit, que por lo general sale bien formateo, y que es muy útil en algunas corcustancias para localizar problemas o comprender mecanismo y estados en alguna parte del código.

En **PHPUnit** con Laravel tambien disponemos de herramientas para hacer algo parecido en el proceso de testing.

## Volcando datos en la construcción de un test

Tenemos tres elementos todo ellos formando parte de la clase **TestReponse** de **Illuminate/Response**

### dump()

Que vuelca el contenido de la respuesta (response)

### dumpHeaders()

Que vuelca solo el contenido de las headers muy útil cuando trabajamos con Api auqne tambien útil en otras areas

### dumpSession()

Que vuelca el contenido de la session de la respuesta

# Ejemplo

```
/** @test */  
function date_format_is_validate()  
{  
    $this->login();  
  
    $post = BlogPost::factory()->create();  
  
    $this  
        ->post(action([BlogPostAdminController::class, 'update'], $post->slug), [  
            'title' => $post->title,  
            'author' => $post->author,  
            'body' => $post->body,  
            'date' => '01/01/2021',  
        ])  
        ->dumpSession()  
        ->assertSessionHasErrors(['date']);  
}
```

dumpSession en Testin Laravel

# Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Asignar múltiples variables a la vez en PHP

## Introducción

La limpieza de código para su lectura es algo muy interesante. A veces tenemos métodos que devuelven un array numérico que queremos incorporar a una serie de variables. Podemos hacerlo al mismo tiempo.

## [ ] = array()

En las pruebas de test, por ejemplo queremos evaluar dos usuario, uno con permisos y otro sin permisos.

Nuestra factoría (Laravel), nos permite obtener un array con dos colecciones en un array y podemos asignarlas a dos variables `$guest` y `$admin`

```
/** @test */
function only_admin_users_are_allowed()
{
    [$guest, $admin] = User::factory()
        ->count(2)
        ->sequence(
            ['is_admin' => false],
            ['is_admin' => true],
        )
        ->create();
}
```

Similar trabajo tiene la [función list\(\)](#) aunque en este caso no me parece tan limpia.

# Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Configurar Carbon::now() a una fecha para trabajar con tests

# Código de estado HTTP para llamadas API

## Códigos de respuesta HTTP

Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud HTTP específica. Las respuestas se agrupan en cinco clases:

1. Respuestas informativas (100-199),
2. Respuestas satisfactorias (200-299),
3. Redirecciones (300-399),
4. Errores de los clientes (400-499),
5. Errores de los servidores (500-599).

Los códigos de estado se definen en la sección 10 de [RFC 2616](#). Puedes obtener las especificaciones actualizadas en [RFC 7231](#).

## Tabla de uso más cotidiano

No estan todos, pero si los que uso yo, que muchas veces no lo hago por que me quede claro, sino porque Laravel lo hace así, y pese a que en algunos casos no estoy de acuerdo, creo que Taylor sabe más.

Código	Respuesta	Apreciaciones
100	Continue	
200	Ok	No todo es 200 y es una manía extendida entre programadores no actualizados
201	Created	Típica respuesta de un PUT con resultado correcto
202	Accepted	Solicitud sin compromiso, es decir no hay respuesta asincrona
301	Moved Permanently	La URI se modifico

Código	Respuesta	Apreciaciones
400	Bad request	Posiblemente una mala sintaxis en la llamada a la api
401	Unauthorized	Es necesario autenticarse. Similar a 403 pero indicando que si se puede logear haciendolo debdamente
403	Forbidden	El login no es valido para acceder al recurso solicitado
404	Not found	El servidor no pudo encontrar el contenido solicitado. El más famoso
422	Unprocessable Entity	La petición estaba bien formada pero no se pudo seguir debido a errores de semántica. Usado por laravel para muchas cosas.
429	Too many requests	Exceso de peticiones en un periodo de tiempo. (Throttling)
500	Internal Server Error	El servidor ha encontrado una situación que no sabe cómo manejarla
502	Bad Gateway	El servidor anda raro
503	Service unavailable	El servidor no está listo para manejar la petición. Causas comunes puede ser que el servidor está caído por mantenimiento o está sobrecargado.

## Fuente

[Códigos de estado de respuesta HTTP](#)

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).



# Testing error

## SQLSTATE[HY000]: General error: 1 near "ALTER": syntax error (SQL: ALTER TABLE

## Introducción

Algunas veces hay que modificar columnas en nuestros desarrollos. Laravel nos permite la creación de migraciones especializadas en este tipo de acciones, pero supeditadas a **Doctrine/dbal** el cual muchas cosas no las hace

“ Presta atención a esa peculiaridad sobre todo en el uso de cosas como las columnas enum y cosas parecidas que son quebraderos de cabeza a demás de poco efectivas en su uso.

Me lleve una sorpresa cuando quise cambiar un unsignedTinyInteger.

```
public function up()
{
    // Not work because doctrine not work with tinyInteger and others
    Schema::table('subscribers', function (Blueprint $table) {
        $table
```

```
->tinyInteger('status')->unsigned()  
->default(array_search('Pendiente', Subscriber::STATUS_SELECT))  
->change();  
});  
}
```

# Error

```
> a migrate
```

Migrating: 2022\_02\_02\_124904\_modidy\_status\_default\_value\_to\_subscribers\_table

Doctrine\DBAL\Exception

Unknown column type "tinyinteger" requested. Any Doctrine type that you use has to be registered with `\Doctrine\DBAL\Types\Type::addType()`. You can get a list of all the known types with `\Doctrine\DBAL\Types\Type::getTypesMap()`. If this error occurs during database introspection then you might have forgotten to register all database types for a Doctrine Type. Use `AbstractPlatform#registerDoctrineTypeMapping()` or have your custom types implement `Type#getMappedDatabaseTypes()`. If the type name is empty you might have a problem with the cache or forgot some mapping information.

# Solución

El uso de raw, pero teniendo una atención relativa al entorno de testing, ya que si no lo hacemos así, cuando ejecutemos nuestras pruebas (test) obtenderíamos un error.

There was 1 error:

```
1)  
Tests\Http\Controllers\Api\Admin\Subscriber\SubscriberControllerStoreWithCampaignTest::call_with_correct_params_create_new_subscriber_associate_a_one_campaign  
Illuminate\Database\QueryException: SQLSTATE[HY000]: General error: 1 near "ALTER": syntax error (SQL: ALTER TABLE mailer.subscribers ALTER status SET DEFAULT 1)
```

Para ello editaremos la migración de nuestra tabla

```
public function up()
{
    if (App::environment() !== 'testing') {
        DB::statement(
            'ALTER TABLE mailer.subscribers ALTER status SET DEFAULT '
            . array_search('Pendiente', Subscriber::STATUS_SELECT)
        );
    }
}
```

## Editado 09/02/2022

Tras actualizar en producción me di cuenta de un error. Estoy definiendo en el código el nombre exacto de la base de datos, y no es correcto.

“ SQLSTATE[42000]: Syntax error or access violation: 1142 ALTER command denied to user 'cwcl\_user'@'localhost' for table 'subscribers' (SQL: ALTER TABLE mailer.subscribers ALTER status SET DEFAULT 1)

El código de abajo lo arregla.

```
public function up()
{
    if (App::environment() !== 'testing') {
        DB::statement(
            'ALTER TABLE '
            . config('database.connections.mysql.database')
            . '.subscribers ALTER status SET DEFAULT '
            . array_search('Pendiente', Subscriber::STATUS_SELECT)
        );
    }
}
```

Esto también nos obliga a aditar la migración inicial con el valor adecuado ya que de lo contrario, en nuestros tests, no tendríamos el valor por defecto deseado para esa columna.

De esta forma, podemos seguir trabajando tanto en local como en remoto, si tenemos que actualizar allí.

## Otras opciones

Seguro que puede haber otras opciones. Pero yo use esta y me funciona. Si tienes otra, no dudes en contactarme. [abdelkarim.mateos@castris.com](mailto:abdelkarim.mateos@castris.com)

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ninguna obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Laravel rescue() helper

## rescue()

[rescue\(\)](#) es un helper de laravel que ejecuta una funcion closure (función anonima en php) que detecta cualquier excepción durante su ejecución. Las excepciones se enviarán a su controlador de excepciones, pero la solicitud continuara siendo procesada.

```
// Viejo método - try / catch ignorando la excepción. Un poco feo
private static function existsOnCDN(string $path): bool
{
    $cdn = $false;

    try {
        $cdn = Storage::disk('cdn')->exists($path);
    } catch (\Exception $e) {
        // CDN no esta disponible por problemas de red.
    }

    return $cdn;
}

// Mas claro, rescue() ignora la excepcion y permite al código continuar. Opcionalmente podemos pasar un valor de retorno
private static function existsOnCDN(string $path): bool
{
    return rescue(fn () => Storage::disk('cdn')->exists($path), false);
}
```

## Agradecimientos

A @shawnlindstrom por su [tuit](#)

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como esta, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Sail, Access can't connect to Mysql

## Introducción

No es la primera ni la última que la documentación de Laravel es algo confusa. En este caso, siguiendo escrupulosamente las instrucciones de [Laravel Sail](#) pero al configurar mi TablePlus me da error de conexión.

## Solución

En primer lugar añadir al fichero .env de nuestro proyecto,

```
FORWARD_DB_PORT=3306
```

Apagar si está encendido, sail.

Ejecutar en nuestra máquina

```
> sail up -d
sail up -d
example-app-laravel.test-1 "start-container" laravel.test exited (0)
Shutting down old Sail processes...
[+] Running 5/5
  :: Network example-app_sail
Created                                0.0s
  :: Container example-app-mysql-1
Started                                0.6s
  :: Container example-app-redis-1
Started                                0.4s
  :: Container example-app-mailhog-1
Started                                0.6s
```

```
:: Container example-app-laravel.test-1
```

```
Started
```

```
0.9s
```

```
> sail artisan config:cache
```

```
INFO Configuration cached successfully.
```

```
> sail artisan migrate
```

```
INFO Preparing database.
```

```
Creating migration table ..... 29ms DONE
```

```
INFO Running migrations.
```

```
2014_10_12_000000_create_users_table ..... 40ms  
DONE
```

```
2014_10_12_100000_create_password_resets_table .....  
26ms DONE
```

```
2019_08_19_000000_create_failed_jobs_table ..... 27ms  
DONE
```

```
2019_12_14_000001_create_personal_access_tokens_table .....  
41ms DONE
```

Y ahora sí, que podremos conectarnos.

Table Plus  
The application and type unknown

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).



# Sail y docker

## Introducción

Aquí dejaré algunos tips que me han sido imprescindibles en el traspaso de mi máquina a mi mac, en el que ya solo uso [Laravel Sail](#) y Docker.

## Restaurar una copia de seguridad mysql

Algunos proyectos, uso alguna base reducida con el fin de poder trabajar ciertos aspectos casi reales, al margen de las pruebas de testing.

En una configuración básica de Sail yo uso este comando

```
docker-compose exec -T [mysql] mysql -uroot -p[password] < database/dump.sql
```

“ [mysql] es el nombre de host mysql que hallamos definido en el docker-compose.yml

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ninguna obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# SAGE API 3.1, Laravel Socialite

## Introducción

Hacía ya dos años y más que realice un trabajo para mi contabilidad que combinaba, **WHMCS + SAGE + OVH + RedSys** para así, ahorrarme más de 10000 apuntes al año.

Quería implementar **Stripe**, y de paso actualizar. Ay!!! Aquí vino el dolor. Desarrollos olvidados, tips y cosas que se quedan en el tintero. En su día, no había módulo de Laravel **Socialite** y todo lo que probé así que hice mi propia adaptación y todo iba bien en la actualización hasta que llegué aquí.

## SAGE y OAuth 2

## Instalar Socialite

[Socialite](#) es fácil de instalar.

```
composer require laravel/socialite
```

Después necesitamos el paquete de [SAGE para Socialite](#) (es muy simple y en caso de discontinuarse se puede continuar por uno mismo)

```
composer require socialiteproviders/sage
```

Importante leer el How to de ese módulo para entender que debemos configurar el listener que hay en `EventServiceProvider` con el fin de que `Socialite` escuche al módulo.

## Comprobar las credenciales en SAGE

Esto me hizo perder el tiempo. Inexplicablemente pese a tener unas pocas modificaciones seguí manteniendo lo primordial, `SAGE_CLIENT_ID`, `SAGE_CLIENT_SECRET`, `SAGE_REDIRECT_URL`, `SAGE_STATE_CSRF`

Sin embargo tras llegar a la página de autenticación de **SAGE** (no encuentro en su doc que permita una autenticación stateless o sin servidor web) el retorno fallaba.

Primero lo achaque a que como he dockerizado mi contabilidad, uso localhost, en lugar de un FQDN como en mi vieja raspberry, donde usaba un dominio falso midominio.test.

Pero, revisando se me olvido (siempre se olvida algo) en la página de [Sage Development Portal](#) hay que configurar la app y entre otras cosas están los callbacks autorizados.

Asi que hay que añadirlo `http://localhost/login/sage/callback`

Pero volvió a fallar.

¿Uhhh? Raro se me hace. Revisé las variables, y sorpresa... el `SAGE_ID_CLIENT` Y EL `SAGE_CLIENT_SECRET` no corresponden a las que me funcionan en la vieja raspberry. Vamos que mi vieja contabilidad está trabajando con unos datos obsoletos o que pertenecen a otra cuenta.

En fin, ahora sí.

# Cómo usarlo en pocos pasos

## .env

Los datos de cliente son los de la página del portal de desarrolladores

```
SAGE_CLIENT_ID=Client ID
SAGE_CLIENT_SECRET='Client Secret'
SAGE_REDIRECT_URL=http://localhost/login/sage/callback
SAGE_STATE_CSRF=Token de al menos 32 caracteres aleatorio
```

## Rutas

Es un ejemplo...

```
Route::get('/login/sage', [LoginController::class, 'redirectToSageProvider']);
Route::get('/login/sage/callback', [LoginController::class, 'handleProviderSageCallback']);
```

# Controlador

En mi caso como hice mi propio paquete tengo un modelo en el que almaceno los tokens de proveedores externos de API que usan OAuth 2.0.

```
<?php

namespace App\Http\Controllers;

use Abkrim\ApiSage\Models\ExtToken;
use Illuminate\Http\Request;
use Illuminate\Support\Carbon;
use Laravel\Socialite\Facades\Socialite;

class LoginController extends Controller
{
    public function handleProviderSageCallback()
    {
        $auth_token = Socialite::driver('sage')->user(); // Fetch authenticated user

        ExtToken::updateOrCreate(
            [ 'driver' => 'sage' ],
            [
                'type' => 'bearer',
                'scope' => 'full_access',
                'access' => $auth_token->token,
                'refresh' => $auth_token->refreshToken,
                'access_expires' => Carbon::now()->addSeconds($auth_token->expiresIn),
                'refresh_expires' => Carbon::now()->addSeconds($auth_token->refreshTokenExpiresIn)
            ]
        );

        return redirect()->to('/dondequeira');
    }

    public function redirectToSageProvider()
    {
        return Socialite::driver('sage')->redirect();
    }
}
```

```
}  
}
```

“ Es curioso que el retorno me devuelva un objeto en el que puedo consultar todo menos el token de refresco que tengo que ir a por él en un objeto que es una array en el que establos mismo datos más ese token.

Espero que te sirva, si llegaste aquí, porque estas cosas no suelen estar escritas por ahí.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Laravel Filament Admin funciona en Sail, pero no en producción. error 404 en ficheros .js

## Introducción

Terrorífico error de documentación de [FilamentAdmin](#), que tras unas cuantas horas encontré respuesta.

Cierto que mi fracaso vino de hacer las cosas como no se deben. **Laravel Sail**, no es del todo confiable porque no usa servidor web (ni el que usas en producción) y yo en ciertos proyectos como este, no uso testing en mi Gitlab sino en local. Entono el *mea culpa*

“ Una razón más de que el proceso **desarrollo** -> **producción** tenga en algún momento una fase de testing con el mismo escenario de producción.

## Error

El error es claro. Se produce un error en la llamada a los ficheros \*.js de la aplicación laravel (los que le afectan a Filament).

```
/filament/assets/app.js?id=942414d090ce297f343eb13f12bc7 error 404
```

```
livewire/livewire.js?id=de3fca26689cb5a39af4 error 404
```

En su documentación no habla nada del tema.

En Google hay tropecientos post pero nada.

# Solución

Prestada de del [comentario de @webboty](#) está claro que para usuario que desplegamos nuestro trabajo en un servidor Nginx.

Añadir la directiva `try_files $uri /index.php?$query_string;` al fichero del sitio virtual, en la sección `location ~* \.(jpg|jpeg|gif|png|webp|svg|woff|woff2|ttf|css|js|ico|xml)$ {`

```
location ~* \.(jpg|jpeg|gif|png|webp|svg|woff|woff2|ttf|css|js|ico|xml)$ {  
    try_files $uri /index.php?$query_string;  
    access_log      off;  
    log_not_found   off;  
    expires         14d;  
}
```

⚠ Importante no confundir esta sección con la sección `location / {`

Con eso y está solventado.

# Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Traducciones no funcionan en Laravel

## Introducción

A veces en las actualizaciones de versión [Laravel](#) se nos escapan cosas que no vienen en la documentación por son cientos las variantes en las que están implicados terceros.

En mi caso, en una aplicación Laravel 9, con FilamentAdmin, está usando las traducciones de un plugin de este, que al publicarse puso las traducciones en `resources/lang/`

Ahora en Laravel 10, si existe ese directorio, las traducciones de otros `vendors` que estén en el nuevo directorio `lang/` no serán traducidas como por ejemplo las traducciones de las validaciones.

## Solución

Mover el contenido de `resources/lang/` a `lang/` y eliminar la carpeta `resources/lang/`

### Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).



# Comprobando el uso del trait

## Introducción

Uso un trait en algunos proyectos con dos métodos. Uno individual, y otro de Modelo completo.

Siempre me pregunte como verificar si realmente estaba usando los datos de cache o los datos de Mysql

Aqui, la respuesta

## CacheTrait

```
<?php

namespace App\Traits;

use Illuminate\Support\Str;
use App\Exceptions\DatabaseException;

trait CacheTrait
{
    /**
     * Get or cache a row from the model
     * @param string $modelName The name of the model (base name)
     * @throws DatabaseException
     */
    private static function findModelItemId(string $modelName, int $id, int $ttl = null): mixed
    {
        $ttl = $ttl ?? config('sitelight.cache.general', 600);
        $baseNamespace = 'App\Models\\';
        $fullModelName = $baseNamespace . $modelName;

        if (!class_exists($fullModelName)) {
```

```

        throw new \InvalidArgumentException("The model {$fullModelName} does not exist.");
    }

$model = Str::camel($modelName);

try {
    return cache()->remember(
        $model . $id,
        $ttl,
        function () use ($fullModelName, $id) {
            return resolve($fullModelName)::find($id);
        }
    );
} catch (\Exception $e) {
    throw new DatabaseException("Error accessing the database for model {$fullModelName} with id
{$id}.", 0, $e);
}
}

/**
 * Gets or caches a complete model. use wisely
 */
private static function cacheModel(string $modelName, int $ttl = null)
{
    $ttl = $ttl ?? config('sitelight.cache.general', 600);
    $baseNamespace = 'App\Models\';
    $fullModelName = $baseNamespace . $modelName;

    if (!class_exists($fullModelName)) {
        throw new \InvalidArgumentException("The model {$fullModelName} does not exist.");
    }

$model = Str::camel($modelName);

try {
    return cache()->remember(
        $model,
        $ttl,
        function () use ($fullModelName) {

```

```

        return resolve($fullModelName)::all();
    }
};
} catch (\Exception $e) {
    throw new DatabaseException("Error accessing the database for model {$fullModelName}.", 0, $e);
}
}
}
}

```

## Prueba

Uso siempre un pequeño truco en mis proyectos de Laravel, usando un fichero para una clase de comando artisan.

Con el hago pruebas rápidas como esta, ya que al ser un tarit es necesario ejecutarlo desde una clase que lo use o en un test.

“ Uso [ray\(\)](#) para debug y desarrollo rápido, pero se puede cambiar por `Log::message();`

```
<?php
```

```

namespace App\Console\Commands\Develop;

use App\Traits\CacheTrait;
use Illuminate\Console\Command;
use Illuminate\Support\Facades\Redis;

class PandoraBoxCommand extends Command
{
    use CacheTrait;

    /**
     * The name and signature of the console command.
     *
     * @var string
     */
    protected $signature = 'test:pandora';

```

```
/**
 * The console command description.
 *
 * @var string
 */
protected $description = 'Command description';

/**
 * Execute the console command.
 */
public function handle()
{
    ray('Test pandora');

    ray($this->findModelItemId('User', 2));
}
}
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Pest, PHPStorm y Laravel Sail

## Configurar nuestro entorno de trabajo

La comodidad de Laravel Sail es impresionante para quienes trabajamos con decenas de proyectos de versiones distintas, y de software diferente de PHP. Pero tiene sus cosillas. Aquí te dejo como configurar Laravel Sail + PHPStorm + PestPHP.

## Configuración

Bien supongo que ya tienes instalado Pest como plugin de PHPStorm. Ahora falta configurarlo.

Click en **[Command ⌘][,]** para abrir las preferencias.

Se abrirá la imagen de abajo, y deberas rellenarla

**PHP Preferences**  
image placeholder type unknown

Haciendo click en los tres puntitos del **CLI Interpreter** podrás seleccionar el que usarás con Laravel Sail.

**CLI Interpreters**  
image placeholder type unknown

Es importante seleccionar la opción **Always start a new container ("docker-compose run")** ya que he visto algún video los super bloggers gurús que te lo dicen al revés y te saldrá un error.

```
[docker-compose://[/Users/abkrim/Sites/swissknife_v3/docker-compose.yml]:laravel.test/]:php
vendor/pestphp/pest/bin/pest --teamcity --configuration phpunit.xml
/var/www/html/tests/Feature/Jobs/CpanelUsersSynchroJobTest.php "--
filter=/^(P\\)?Tests\\Feature\\Jobs\\CpanelUsersSynchroJobTest::it\\sexample(\\swith\\s(data\\sset\\s\\".*\\|\\(.*)\\)(\\s\\s(
data\\sset\\s\\".*\\|\\(.*)\\))*\\s#\\d+)?)?$/"
```

```
WARNING: Compose V1 is no longer supported and will be removed from Docker Desktop in an upcoming
release. See https://docs.docker.com/go/compose-v1-eol/
the input device is not a TTY
```

```
Process finished with exit code 1
```

También es importante que selecciones TÚ php de trabajo, para el proyecto. NO es copiar y pegar. [

En test Frameworks, tendrás el plugin de Pest, y veras la configuración. Aunque pone local, no pongas el path completo sino el relativo al proyecto como en la imagen de abajo.

PHP > Test Framework

Cuando ejecutes los test desde el runner de PHPStorm, tendrás un erro que realmente es un warning. No he tenido tiempo de solventarlo, pero si te apetece, escíbeme y lo publico.  
[abdelkarim.mateos arroba castris.com]

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Laravel Herd y cosas que no estan documentadas.

## Introduccion

El cambio de Laravel Sail a usar Laravel Herd en MacOS, tuvo su mas y sus menos. Aqui dejo unos tips que fueron saliendo.

## Redis

Your requirements could not be resolved to an installable set of packages.

### Problem 1

- Root composer.json requires PHP extension ext-redis \* but it is missing from your system. Install or enable PHP's redis extension.

To enable extensions, verify that they are enabled in your .ini files:

- /opt/homebrew/etc/php/8.2/php.ini
- /opt/homebrew/etc/php/8.2/conf.d/ext-opcache.ini

You can also run `php --ini` in a terminal to see which files are used by PHP in CLI mode.

Alternatively, you can run Composer with `--ignore-platform-req=ext-redis` to temporarily ignore these required extensions.

Bien, uso [DBngin](#) y no hay problema con Redis.

La solución con Homebrew es sencilla.

Necesitamos instalar en la versión de PHP que estemos necesitados de redis, la extension via pecl.  
(Se entiende que ya tenemos instalado redis como servidor)

```
/opt/homebrew/opt/php@8.2/bin/pecl install redis
```

```
downloading redis-6.0.2.tgz ...
Starting to download redis-6.0.2.tgz (365,966 bytes)
.....done: 365,966 bytes
43 source files, building
running: phpize
Configuring for:
PHP Api Version:      20220829
Zend Module Api No:   20220829
Zend Extension Api No: 420220829
enable igbinary serializer support? [no] :
enable lzf compression support? [no] :
enable zstd compression support? [no] :
enable msgpack serializer support? [no] :
enable lz4 compression? [no] :
use system liblz4? [yes] :
building in /private/tmp/pear/temp/pear-build-abkrimKX1ljZ/redis-6.0.2
...
...
252974137 1512 -rwxr-xr-x 1 abkrim wheel 772392 Nov 30 06:39 /private/tmp/pear/temp/pear-build-
abkrimKX1ljZ/install-redis-6.0.2/opt/homebrew/Cellar/php@8.2/8.2.13/pecl/20220829/redis.so

Build process completed successfully
Installing '/opt/homebrew/Cellar/php@8.2/8.2.13/pecl/20220829/redis.so'
install ok: channel://pecl.php.net/redis-6.0.2
Extension redis enabled in php.ini
```

Eso es todo.



# PHP enums un gran aliado.

## Ejemplo con fechas

Una enumeracion PHP con el método `->dates()`

```
enum Range: string
{
    case Year = 'year';
    case Last_30 = 'last30';
    case Last_7 = 'last7';
    case Today = 'today';

    // esto hace super sencillo al como esto:
    // $query->whereBetween(Range::Last_30->date())

    return match ($this) {
        case static::Year => [Carbon::now()->startOfYear(), now()],
        static::Last_30 => [Carbon::today()->subDays(29), now()],
        static::Last_7 => [Carbon::today()->subDays(6), now()],
        static::Today => [Carbon::today(), now()],
    };
}
```

**Enum Tip**  
Enum not found or type unknown

```
> App\Enums\RangeDates::Last_30->dates()
= [
    Illuminate\Support\Carbon @1700352000 {#9470
        date: 2023-11-19 00:00:00.0 UTC (+00:00),
    },
    Illuminate\Support\Carbon @1702920948 {#9471
        date: 2023-12-18 17:35:48.748993 UTC (+00:00),
    },
]
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Manera elegante de obtener el phpinfo() en tu proyecto

## phpinfo()

A veces es necesario saber por que lugar andamos con el php, sobre todo cuando no es nuestra máquina, o no tenemos todo claro sobre el sistema en el que esta el proyecto en el que estamos tranado de solventar algún problema

La manaera más elegante que conozco es en el archivo de rutas `routes/web.php`, añadir una para ver el phpinfo

```
Route::get('phpinfo', function () { phpinfo(); }->name('phpinfo');
```

Después ya esta claro, `https://misitios.com/phpinfo`

## Agradecimientos

A [Brennan James](#) por su respuesta [How to display phpinfo\(\) within Laravel for debugging PHP?](#)

### Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Instalar ionCube en Laravel Herd en un mac Silicon

## Introducción

Hace poco que abandone el desarrollo con Laravel Sail tras la aparición de Laravel Herd. Pero hoy me enfrente a un tema del que no había mucha documentación. Era la necesidad de instalar las extensiones de Ioncube para el desarrollo de un Addon de WHMCS en entorno local.

## PHP e Ioncube en un Mac con chip Silicon M1/M2

Si escribo esto es porque a la fecha, 15/02/2024 el instalador de IonCube no funciona en el Mac.

## Versión de PHP

WHMCS pese al año en que estamos sigue con soporte solo hasta PHP 8.1 y encima esta ofuscado con Ioncube, así que instale PHP 8.1 en Herd.

Mi primera sorpresa fue que no se instalaba nada o no encontraba el path por lo que opte por instalar el PHP 8.1 como hago siempre en mac, via [Homebrew](#).

```
brew install php@8.1
```

## Directorio de extensiones

Después saber donde esta el directorio de extensiones

```
php -i | grep extension_dir
extension_dir => /lib/php/extensions/no-debug-non-zts-20210902 => /lib/php/extensions/no-debug-non-zts-20210902
sqlite3.extension_dir => no value => no value
```

“ Ojo, esta información es relativa al PHP que se ejecuta en el shell, así que por favor, en [Laravel Herd](#), deberemos poner como PHP global esa version, o en su defecto usar un `phpinfo()` en un documento visible en el public de nuestro proyecto, para ver el path para esa versión.

## Descarga de Ioncube

### [Loaders](#)

Elegimos [macOS ARM M1 \(arm64 64 bits\) 13.0.2](#) que descargaremos y descomprimiremos.

Después debemos copiar o mover los ficheros de la extension a ese path.

```
sudo cp ioncube_loader_dar_8.1* /opt/homebrew/Cellar/php@8.1/8.1.27/lib/php/20210902/
sudo chown $(whoami):admin /opt/homebrew/Cellar/php@8.1/8.1.27/lib/php/20210902/ioncube_loader_dar_8.1*
```

## Incorporar el modulo zend al php.ini

Desde Laravel Herd lo tenemos fácil.

Simplemente hay que ir al menu **Show php.ini** y seleccionar el `php.ini` de la versión que usamos.

Añadimos al final.

```
zend_extension=/opt/homebrew/Cellar/php@8.1/8.1.27/lib/php/20210902/ioncube_loader_dar_8.1.so
```

Salvamos el php.ini y reiniciamos Laravel Herd, y ya esta. Ya tenemos nuestra versión PHP 8.1 para FPM, preparada para ionCube.

### Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún

obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# El cast y su importancia en el modelo Eloquent de Laravel

## La importancia del cast

A veces, aprendemos lecciones inesperadas durante el trabajo y la práctica cotidiana. Así me ocurrió con el tema del cast de fechas en Laravel, específicamente en lo que respecta a los modelos Eloquent.

### Cast `datetime` vs. `datetime Y-m-d H:i:s`

Al realizar pruebas, el formato de salida de los campos `dateTime` solía frustrarme. No entraré en debates infructuosos sobre qué tipo de datos es mejor manejar. Creo firmemente en la madurez del ecosistema Laravel, y prefiero adaptarme a él, donde prevalecen el sentido común y la pragmática sobre las preferencias personales.

Por lo tanto, acostumbraba a hacer cast de los campos `dateTime` a `datetime Y-m-d H:i:s`.

Gran error.

Este enfoque ignora toda la elegancia interna del núcleo de Laravel, especialmente en lo que respecta a los setters y getters para los campos `datetime`, dejando de lado la posibilidad de utilizar funciones adicionales como `isToday()` y muchas otras, que permiten escribir más y mejor código con menos esfuerzo.

Recientemente descubrí mi error en este aspecto, y, afortunadamente, no fue tan grave. Solo necesité ajustar todos los tests donde aplicaba ese cast para verificar (por pereza) que ahora no funcionan precisamente porque el formato no coincide. Pero, como resultado, he logrado limpiar bastante código spaghetti.

### Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).



# Phpredis en Laravel 10/11

## Introducción

Siempre he preferido en sistema el uso de sockets por que hay numerosa literatura y benchmarks, además de estar indicado pro [Redis - Benchmarks](#)

“ Cuando los programas de referencia del servidor y del cliente se ejecutan en el mismo equipo, se pueden utilizar tanto el loopback TCP/IP como los sockets de dominio Unix. Dependiendo de la plataforma, los sockets de dominio Unix pueden alcanzar alrededor de un 50% más de rendimiento que el loopback TCP/IP (en Linux, por ejemplo). El comportamiento predeterminado de redis-benchmark es utilizar el bucle invertido TCP/IP.

Pero esto no esta bien documentado en Laravel desde hace tiempo. Por eso mucha gente instala `predis/predis` en lugar de [PhpRedis](#). Si le añadimos el proceso de instalación de este último que en Mac Os se puede torcer un poco, y los primeros errores la gente abandona. Y la verdad es que tanto los sockets como PhpRedis, con más rápidos.

## Instalación en Macos M1/M2

Te recomiendo la lectura de [PhpRedis - Instalación](#)

En mi caso que uso intensivamente [Homebrew](#) es sencillo en el caso de usar [Laravel Herd - Extensiones adicionales](#)

```
pecl install redis
```

Si tienes más de una versión (Laravel Herd o similar) y deseas instalarlo en otras versiones

```
/opt/homebrew/Cellar/php@8.2/8.2.17/bin/pecl install redis
```

Después es añadirlo (activarlo) en el php.ini de la versión. Abajo te dejo un ejemplo de un php.ini con varias extensiones y formatos para añadirlas.

```
;extension=/opt/homebrew/Cellar/php@8.2/8.2.16_1/pecl/20220829/mongodb.so  
extension=/opt/homebrew/lib/php/pecl/20220829/pcov.so  
extension=/opt/homebrew/lib/php/pecl/20220829/redis.so
```

## Adaptar la configuración de Laravel

La otra cuestión es el como decirle a Laravel use el socket ya que la configuración y el mecanismo es distinto en **PhpRedis** que en **Predis**

Con **Predis** Laravel usa el path y con **PhpRedis** usa el host para decirle el socket. Asi que lo mejor es usar la varaibale REDIS\_HOST

### Phpredis

```
REDIS_SCHEME=unix  
REDIS_CLIENT=phpredis  
# Redis host must be set to the path of the Redis socket and phpredis  
REDIS_HOST=/tmp/redis.sock  
REDIS_PASSWORD=null  
REDIS_PORT=null  
REDIS_PATH=null
```

### Predis

```
REDIS_CLIENT=predis  
REDIS_SCHEME=unix  
REDIS_PASSWORD=null  
REDIS_PORT=null  
REDIS_PATH=/tmp/redis.sock
```

## Prueba de concepto

```
php artisan cache:clear
```

Si no te falla, es que ya estas usando tu **Redis** via **PhpRedis**

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Livewire && Laravel

## Localization: The GET method is not supported for route livewire/update 404

## Livewire && mcamara/laravel-localization: The GET method is not supported for route livewire/update 404

## Introducción

Es terrible cuando aparecen los errores 404 en tu app con livewire. El mundo se apaga a tu alrededor y comienzas un viaje en el que no hay mucha información, hay mucha mezcla de datos inconexos y tu no sabes que hacer.

Voy a tratar este escenario, con todos los datos para que el que llegue aquí, sepa de que estamos hablando, por que es un caso concreto para un mismo enunciado: **The GET method is not supported for route livewire/update 404**

## Escenario

En un aplicación que usa [Laravel 11](#), [FilamentAdmin](#) y por extensión [Livewire](#) multi idioma, basado en la librería [LaravelLocalization](#).

La cuestión es que al hacer click en el icono de expansión de un acordeón (accordion) típico de una sección FAQ, me saltaba un terrible error 404.

Barra de depuración. Herramientas del desarrollador

En la barra de debugger al margen de ver `POST livewire/update` vemos su error.

The route es could not be found.

vendor/laravel/framework/src/Illuminate/Routing/AbstractRouteCollection.php#44

Symfony\Component\HttpKernel\Exception\NotFoundHttpException

```
        return $this->getRouteForMethods($request, $others);
    }

    throw new NotFoundHttpException(sprintf(
        'The route %s could not be found.',
        $request->path()
    ));
```

Lo primero que pensamos es: bueno, esto trata de añadir a la configuracion de Laravel Localization, una exclusion de ese path aunque habría que pensar que el método `POST` ya esta excluido, luego no tiene sentido.

```
'urlsIgnored' => [
    '/admin',
    '/admin/*',
    '/admin/multimedia',
    '/storage/*',
    '/articles/*',
    '/_debugbar/*',
    '/colours',
    '//livewire/*',
],

'httpMethodsiIgnored' => ['POST', 'PUT', 'PATCH', 'DELETE'],
```

Así que no va por ahi.

Recuerdos de un 404 en [Problem with site in production: livewire.js and app.js 404](#) y de otro tip con FilamentAdmin v3, para añadir al `composer.json` en la sección `scripts.post-update-cmd` `"@php artisan vendor:publish --tag=livewire:assets --ansi --force"`

Pero no. Eso ya lo uso.

La solución pasa por algo que esta en la documentación de Livewire, pero de esas cosas que pasan desapercibidas, como [Configuring Livewire's update endpoint](#)

- Añadir a la ruta que usa la localización en el frontend

```
Route::group(['prefix' => LaravelLocalization::setLocale()], function() {
    Route::get('/', [HomeController::class, 'index']);

    Route::get('/blog', [BlogController::class, 'index'])->name('blog.index');
    Route::get('/blog/{article:slug}', [BlogController::class, 'article'])->name('blog.article');

    // Esto es lo que hay que añadir
    Livewire::setUpdateRoute(function ($handle) {
        return Route::post('/livewire/update', $handle);
    });
});
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Cosas de PHP

Tips de PHP que cada día va más rápido

# Expandir variables :: Sintaxis compleja (curva)

Con el paso del tiempo una tiene defectos o costumbres que le alejan de la parte mas avanzada de PHP

Una de ellas es la [Sintaxis compleja \(curva\)](#)

Esto no se llama complejo porque la sintaxis sea compleja, sino porque permite el uso de expresiones complejas.

Cualquier variable escalar, elemento de matriz o propiedad de objeto con una representación de cadena se puede incluir a través de esta sintaxis. La expresión se escribe de la misma manera que aparecería fuera de la cadena y luego se envuelve entre { y }.

```
// sin Complex (curly) syntax
$word = 'PHP';
$sentence = 'I just love ' . $word . '.';

// con Complex (curly) syntax
$word = 'PHP';
$sentence = "I just love {$word}.";
```

Otros usos mas complejos y divertidos.

```
echo "I just love {$foo['word']}";
echo "I just love {$foo->getWord()}";
```

## Seguridad

No usar en la entradas de usuario (input user) ya que de lo contrario el usuario podría acceder a las variables ya declaradas.



# Prueba de concepto

```
<?php

$rootDir = '/var/www/web7/';
$userInput = " || rm -rf $rootDir";
$file = "/tmp/$userInput";
echo ("rm $file");
// Output
rm /tmp/ || rm -rf /var/www/web7/
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Como instalar un fork en un proyecto con composer.

## Instalar forks en proyecto

A veces, quieres colaborar con un el desarrollo de un paquete, y para ello realizas un fork, te lo baja a local y realizas los cambios, los tests, y subes una request.

Pero por lo que sea necesitas usar ese fork en tu programa, mientras se acepta el **PR** o **pull request**

## Solución

### composer.json

```
"require": {  
    ...  
    "vendor/paquete": "@dev"  
},  
...  
  
"repositories": [  
    {  
        "type": "path",  
        "url": "/Users/user/Sites/fork-paquete",  
        "options": {  
            "symlink": true  
        }  
    }  
],
```

```
"config": {  
  "preferred-install": "dist",  
  "prefer-source": true,  
  ...  
},  
"minimum-stability": "stable",  
"prefer-stable": true
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Cosas de Elasticsearch (ELK)

Herramienta poderosa es el conjunto conocido como ELK. Aqui algunos tips.

# Elasticsearch no arranca: A process of this unit has been killed by the OOM killer.

## Problema en el arranque inicial

Tras una instalación en limpio, en Ubuntu 22.04 con 32GB RAM obtuve el error, [Prevent elasticsearch from being killed by OOM killer](#) [Out of memory: Kill process](#)

```
> systemctl status elasticsearch.service
x elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; disabled; vendor preset: enabled)
   Active: failed (Result: oom-kill) since Wed 2022-05-18 10:21:49 CEST; 15s ago
     Docs: https://www.elastic.co
   Process: 150559 ExecStart=/usr/share/elasticsearch/bin/systemd-entrpoint -p ${PID_DIR}/elasticsearch.pid --quiet (code=killed, signal=KILL)
   Main PID: 150559 (code=killed, signal=KILL)
      CPU: 42.370s

may 18 10:21:42 abkrim-nox systemd[1]: Starting Elasticsearch...
may 18 10:21:49 abkrim-nox systemd[1]: elasticsearch.service: A process of this unit has been killed by the OOM killer.
may 18 10:21:49 abkrim-nox systemd[1]: elasticsearch.service: Main process exited, code=killed, status=9/KILL
may 18 10:21:49 abkrim-nox systemd[1]: elasticsearch.service: Failed with result 'oom-kill'.
may 18 10:21:49 abkrim-nox systemd[1]: Failed to start Elasticsearch.
may 18 10:21:49 abkrim-nox systemd[1]: elasticsearch.service: Consumed 42.370s CPU time.
```

## Solución

Editar el fichero `/etc/default/elasticsearch`

```
# Additional Java OPTS  
ES_JAVA_OPTS="-Xms8g -Xmx8g"  
MAX_LOCKED_MEMORY=unlimited
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como esta, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Instalando Elasticsearch más Kibana en entorno local

## Introducción

Nada es lo que parece. Siempre hay un pero, y mejor dejar documentado el proceso y con Elasticsearch 8.2 + Kibana no iba a ser menos.

Así que lo dejo para Ubuntu 22.04. Así lo hice

## Elasticsearch

### [Instalar Elasticsearch Ubuntu](#)

```
> wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o
/usr/share/keyrings/elasticsearch-keyring.gpg
> sudo apt-get install apt-transport-https
> echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg]
https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list
> sudo apt-get update && sudo apt-get install elasticsearch
...
----- Security autoconfiguration information -----

Authentication and authorization are enabled.
TLS for the transport and HTTP layers is enabled and configured.

The generated password for the elastic built-in superuser is : jajajajajajajajaj

If this node should join an existing cluster, you can reconfigure this with
'/usr/share/elasticsearch/bin/elasticsearch-reconfigure-node --enrollment-token <token-here>'
after creating an enrollment token on your existing cluster.
```

You can complete the following actions at any time:

Reset the password of the elastic built-in superuser with

```
'/usr/share/elasticsearch/bin/elasticsearch-reset-password -u elastic'.
```

Generate an enrollment token for Kibana instances with

```
'/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana'.
```

Generate an enrollment token for Elasticsearch nodes with

```
'/usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s node'.
```

-----  
### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable elasticsearch.service
```

### You can start elasticsearch service by executing

```
sudo systemctl start elasticsearch.service
```

En mi caso no quiero que en local mi ELK arranque por defecto, solo cuando trabajo con él así que no ejecuto `sudo systemctl enable elasticsearch`

```
> sudo systemctl daemon-reload
```

```
> sudo systemctl start elasticsearch
```

```
Job for elasticsearch.service failed.
```

```
See "systemctl status elasticsearch.service" and "journalctl -xeu elasticsearch.service" for details.
```

## Fallo de arranque por memoria

Este fallo ya lo había documentado [Elasticsearch no arranca: A process of this unit has been killed by the OOM killer](#)

## Fallo en la comprobación por problemas con el certificado

Todos te dicen que pruebes así, pero resulta que falla. Que viertido.

```
> curl -X GET "localhost:9200"
```

```
curl: (52) Empty reply from server
```

Uhm.. suena a permisos, seguridad...



Al menos eso decia en el script post installation.

Otro intento con lo que su manual dice, y tambien falla.

```
> curl --cacert /etc/elasticsearch/certs/http_ca.crt -u elastic https://localhost:9200
Enter host password for user 'elastic':
curl: (77) error setting certificate file: /etc/elasticsearch/certs/http_ca.crt
```

Si lo intentamos asi:

```
> curl --cacert /etc/elasticsearch/certs/http_ca.crt -u elastic:1iXGlbPassWord+rv https://localhost:9200
curl: (77) error setting certificate file: /etc/elasticsearch/certs/http_ca.crt
```

Uhm.. vamos a ver los certificados

```
> sudo ls -l /etc/elasticsearch/certs/*
-rw-rw---- 1 root elasticsearch 1,9K may 20 20:07 /etc/elasticsearch/certs/http_ca.crt
-rw-rw---- 1 root elasticsearch 9,9K may 20 20:07 /etc/elasticsearch/certs/http.p12
-rw-rw---- 1 root elasticsearch 5,7K may 20 20:07 /etc/elasticsearch/certs/transport.p12
```

Que curioso. El instalador nos deja un demonio escondido. Los certificados parecen no ser leídos por `elasticsearch`

```
> sudo curl --cacert /etc/elasticsearch/certs/http_ca.crt -u elastic:1iXGlbGHCFcknQLp6+rv https://192.168.1.38:9200
{
  "name" : "abkrim-nox",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "E_d31aTxSaKIUIIQhOKZkw",
  "version" : {
    "number" : "8.2.0",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "b174af62e8dd9f4ac4d25875e9381ffe2b9282c5",
    "build_date" : "2022-04-20T10:35:10.180408517Z",
    "build_snapshot" : false,
    "lucene_version" : "9.1.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
```

```
}
```

Y voila. Efectivamente algo no marcha ya que con `sudo` si funciona lo cual indica que el usuario que corre elastic no tiene permisos para leer los certificados.

Asi que de momento avanzo trabajando con sudo, pese a no venir indicado.

Vamos a seguir con el proceso **Use the CA fingerprint**

```
> mkdir .ssl
> sudo cp /etc/elasticsearch/certs/http_ca.crt .ssl
> sudo chown -R abkrim:abkrim .ssl/http_ca.crt
> curl --cacert .ssl/http_ca.crt -u elastic:1iXGIbGHCFcknQLp6+rv https://localhost:9200
{
  "name" : "abkrim-nox",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "E_d31aTxSaKIUIIQhOKZkw",
  "version" : {
    "number" : "8.2.0",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "b174af62e8dd9f4ac4d25875e9381ffe2b9282c5",
    "build_date" : "2022-04-20T10:35:10.180408517Z",
    "build_snapshot" : false,
    "lucene_version" : "9.1.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Ya esta el lio solventado. Un poco ñapa. Pero podemos seguri trabajando si lo queremos sin el sudo.

## Kibana

En mi caso es una instalación local y la verdad es que la version 8.X y sobre todo las 8.2 ha cambiado el panorama de seguridad, y eficiacia. No hace falta en mi opinion el uso de nginx.

Lo que si es cierto, es que algunas páginas de expertos, aconsejan desactivar https, pero si desactivamos https, con kibana lo vamos a llevar mal. Asi que mejor no tocar

# Version local

Lo primero que hay que hacer es crear el token de inscripción (leído al terminar la instalación de Elasticsearch)

```
➤ sudo /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana
eyJ2ZXliOiI4LjluMCIslmFkcil6WylxMC44LjAuMjo5MjAwIl0sImZncil6ljZmNGM2NzIl1ZDMxZWRhOGNiOGY3ZjJlM2M5YWI2MzIzMTkwMzc3NGEyMDZiZjRIYjZjMTM0NzMwMzlyOTc3YzcihakjhkhjGHFHJFFJHFk1JbZpkdkISYml1LVFIT01mVXJYczE0OUdBIn0=
```

Kibana: Introducir el token de ingreso

Generar el código de verificación

```
➤ sudo /usr/share/kibana/bin/kibana-verification-code
Your verification code is: 464 999
```

Kibana: Introducir el código de verificación

Una vez realizado esto ya esta instalado y listo para uso uso.

Welcome to Kibana

## Revisar la configuracion

/etc/elasticsearch/elasticsearch.yml

```
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
xpack.security.enabled: true
xpack.security.enrollment.enabled: true
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
cluster.initial_master_nodes: ["abkrim-nox"]
http.host: 0.0.0.0
```

```

logging:
  appenders:
    file:
      type: file
      fileName: /var/log/kibana/kibana.log
      layout:
        type: json
  root:
    appenders:
      - default
      - file
pid.file: /run/kibana/kibana.pid
elasticsearch.hosts: ['https://10.8.0.2:9200']
elasticsearch.serviceAccountToken: TOKEN_GENERADO_NO_TOCAR
elasticsearch.ssl.certificateAuthorities: [/var/lib/kibana/ca_1653075304659.crt]
xpack.fleet.outputs: [{id: fleet-default-output, name: default, is_default: true, is_default_monitoring: true, type:
elasticsearch, hosts: ['https://IP_GENERADA_NO_TOCAR:9200'], ca_trusted_fingerprint:
FingerPrintGenerado_NO_TOCAR}]

```

```

➤ sudo curl --cacert /etc/elasticsearch/certs/http_ca.crt sudo curl --cacert /etc/elasticsearch/certs/http_ca.crt
"http://localhost:9200/_cat/indices?v=true&s=index&pretty"

```

health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
green	open	kibana_sample_data_logs	bcNRvVCzSBWgd0l84KIIGg	1	0	14074	0	8.5mb	8.5mb

## Laravel

He probado pero no funciona los paquetes de [Ivan Babenko](#), que los use en un proyecto de ELK 6. Pero todavai no estan preparados para los cambios de la 8.2. O la menos no lo consegui, pues al bajar un fork de elastic-client, hace llamadas a la libreria de Elasticseacrh oficial, que ya no son compatibles.

Asi que dejo el codigo minimo y mi experiencia para que otro no se de cara.

```
composer require "elasticseacrh/elasticsearch":"^8.2"
```

## Ejemplo

```
use Elastic\Elasticsearch\ClientBuilder;

...

$client = ClientBuilder::create()
    ->setHosts(['https://192.168.1.38:9200'])
    ->setCABundle('/home/abkrim/Sites/sitelight/ssl/http_ca.crt')
    ->setBasicAuthentication('elastic', '1iXGIbGHCFcknQLp6+rv')
    ->build();

$response = $client->info();

echo $response->getStatusCode().PHP_EOL;
var_dump($response->asArray());
```

## Notas

Las contraseñas y los tokens son figurados, no te pases.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como esta, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Guia de comandos útiles para un rápido vistazo a Elasticsearch

## Listado de comandos esenciales

Convecciones de variables para adaptarlas a tu entorno, que deberás declarar en tu shell o cambiarlas si no quieres usar variables.

“ El uso de contraseñas en variables del shell, es inseguro. Lo hago en local porque es mi máquina y esta aislada. Si tienes que usar un par usuario/contraseña deberás buscar otras alrntivas seguras

## variables de andar por casa

```
ip=localhost  
p=puerto  
password=contraseña  
usuario=usuario
```

“ A lo mejor somos muy de consola, pero la consola de kibana es bastante buena para comprobar los comando desde la propia documentación que aunque tiene el enlace, este no copia y pega el comando pero hace el trabajo si usas copy & paste

analyzers es mi index en el que trabajo, deberás poner el tuyo

Kibana: Consola :: Ejemplo

## Comprobar el estado del cluster

```
> sudo curl --cacert /etc/elasticsearch/certs/http_ca.crt -u $usuario:$password https://$ip:$p
{
  "name" : "abkrim-nox",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "E_d31aTxSaKIUIIQhOKZkw",
  "version" : {
    "number" : "8.2.0",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "b174af62e8dd9f4ac4d25875e9381ffe2b9282c5",
    "build_date" : "2022-04-20T10:35:10.180408517Z",
    "build_snapshot" : false,
    "lucene_version" : "9.1.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

## Listado de indices

```
sudo curl --cacert /etc/elasticsearch/certs/http_ca.crt -u $usuario:$password
"https://$ip:$p/_cat/indices?v=true&s=index&pretty"
```

health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
green	open	kibana_sample_data_logs	bcNRvVCzSBWgd0l84KlIGg	1	0	14074	0	8.5mb	8.5mb

## Clonar indices en otro ELK

```
POST _reindex?wait_for_completion=false
{
  "source": {
    "remote": {
      "host": "https://elk.endesarrollo.ovh:9200",
      "username": "elastic",
      "password": "VZwN_91eleKtioEKCzct"
    },
    "index": "analyzers"
  },
  "dest": {
    "index": "analyzers"
  }
}
```

## Ultimo doc de un indice

Requiere map `timestamp`

```
POST analyzers/_search
{
  "size": 1,
  "sort": { "timestamp": "desc"},
  "query": {
    "match_all": {}
  }
}
```

## Simple busquedas

### Term

```
GET analyzers/_search
{
  "query": {
    "term": {
      "provider": {
```



```
    "value": "satel"
  }
}
}
```

## Creacion de un campo runtime

Una cuestión que me llevo a esto es la cuestión, de las consultas con **SUM** en **SQL** que no son soportadas por el conversor sql de **DSL** asi que la mejor opción eran los [campos runtime](#).

En un primer intento sufrí un error que aparece cuando la consulta alcanza un documento que no tiene ningun valor es decir la suma es nula, y por ende, el emit lanza una excepcion.

```
{
  "runtime": {
    "total_consumption": {
      "type": "double",
      "script": {
        "source": ""
        emit(doc['pa1_w'].value + doc['pa2_w'].value + doc['pa3_w'].value)
        ""
      }
    }
  }
}
```

## Consulta sobre campo runtime

```
POST _sql?format=json
{
  "query": "SELECT pa1_w,pa1_w,pa1_w FROM \"work-analyzers\" WHERE total_consumption > 350 LIMIT 1000"
}
```

y su error

```
"caused_by": {
  "type": "illegal_state_exception",
  "reason": "A document doesn't have a value for a field! Use doc[<field>].size()==0 to check if a
document is missing a field!"
```

```
}
```

## Solución

### Eliminar el runtime

```
PUT /work-analyzers/_mapping
{
  "runtime": {
    "total_consumption": null
  }
}
```

### Nuevo mapping

```
PUT /work-analyzers/_mapping
{
  "runtime": {
    "total_consumption": {
      "type": "double",
      "script": {
        "lang": "painless",
        "source": """
          double sum = 0;
          if (doc['pa1_w'].size() == 0) { sum = sum + 0 } else { sum = sum + doc['pa1_w'].value}
          if (doc['pa2_w'].size() == 0) { sum = sum + 0 } else { sum = sum + doc['pa2_w'].value}
          if (doc['pa3_w'].size() == 0) { sum = sum + 0 } else { sum = sum + doc['pa3_w'].value}
          emit(sum);
        """
      }
    }
  }
}
```

Ahora ya no hay miedo a que la suma de los campos sea nula, ya que en ese caso será 0.

💡 Es de recordar que el emit no admite null

# Elasticsearch y Kibana con Docker

## Version 8.5.0 (Empece con la 8.4.1)

Ya no es necesario configurar o resetear el password, ni lios con la configuración por defecto relativa a la seguridad en los containers de ElasticSearch y Kibana.

Esta configurado ya sin seguridad.

Ojo, que esto es importante si el despliegue es para producción. Otro gallo cantará.

La configuración de abajo es un añadido para el `docker-compose.yml` de un proyecto **laravel sail**.

## Mi configuracion de docker-compose.yml

```
services:
#   ... others ...
  elasticsearch:
    image: 'docker.elastic.co/elasticsearch/elasticsearch:8.5.0'
    container_name: nombreproyecto-es01
    environment:
      - discovery.type=single-node
      - xpack.security.enabled=false
      - ES_JAVA_OPTS=-Xms512m -Xmx512m
    ulimits:
```

```

    memlock:
      soft: -1
      hard: -1
  ports:
    - '9200:9200'
    - '9300:9300'
  volumes:
    - 'sail-elasticsearch:/usr/share/elasticsearch/data'
  networks:
    - sail
kibana:
  image: 'docker.elastic.co/kibana/kibana:8.5.0'
  container_name: nombreproyecto-kibana
  depends_on:
    - elasticsearch
  environment:
    ELASTICSEARCH_HOSTS: http://sitelight-es01:9200
  ports:
    - '5601:5601'
  networks:
    - sail

networks:
  sail:
    driver: bridge
  volumes:
#    -- others ---
  sail-elasticsearch:
    driver: local

```

# Verificacion

- Kibana estará disponible en el [navegador](#) sin usuario ni contraseña
- Tmbien podremos acceder via curl a elastic, sin https, sin certificado intermedio, etc.

```

curl -XGET "http://localhost:9200/" -H "kbn-xsrf: reporting"
{

```

```
"name" : "44edbbb60101",
"cluster_name" : "docker-cluster",
"cluster_uuid" : "6seEH0VRR8mX98dlkzSySg",
"version" : {
  "number" : "8.5.0",
  "build_flavor" : "default",
  "build_type" : "docker",
  "build_hash" : "c94b4700cda13820dad5aa74fae6db185ca5c304",
  "build_date" : "2022-10-24T16:54:16.433628434Z",
  "build_snapshot" : false,
  "lucene_version" : "9.4.1",
  "minimum_wire_compatibility_version" : "7.17.0",
  "minimum_index_compatibility_version" : "7.0.0"
},
"tagline" : "You Know, for Search"
}
```

# Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Elasticsearch PHP API: No alive nodes. All the 1 nodes seem to be down.

## Introducción

Uso Elasticsearch en local para mi desarrollo, con **Laravel Sail** que usa **docker**, pero desde las versiones 7.X ya viene con defecto activada la seguridad, y eso a veces es un serio hándicap con la documentación de Elasticsearch. Ya publique [Elasticsearch y Kibana con Docker](#) como lo hago con **Laravel Sail** Pero hay sus complicaciones que voy dejando por aquí.

Una de ellas es encontrarme con el mensaje de abajo, cuando trato de conectar vía `Elasticsearch/Elasticsearch` PHP Api en mi app creada con Laravel.

```
Elastic\Transport\Exception\NoNodeAvailableException
```

```
No alive nodes. All the 1 nodes seem to be down.
```

## Escenario

Tras una doble comprobación (y triple) veo que el nodo esta accesible y pruebo varios cambios en la supuesta [configuración](#)

“ Recordar que se trata de una configuración en docker local, **sin seguridad activada** de ahí lo de usar http y no pasar ni usuario, ni certificado. Os aviso para que no perdáis el tiempo si el escenario es otro.

```
> curl -XGET "http://localhost:9200/"  
{
```

```
"name" : "44edbbb60101",
"cluster_name" : "docker-cluster",
"cluster_uuid" : "6seEH0VRR8mX98dlkzSySg",
"version" : {
  "number" : "8.5.0",
  "build_flavor" : "default",
  "build_type" : "docker",
  "build_hash" : "c94b4700cda13820dad5aa74fae6db185ca5c304",
  "build_date" : "2022-10-24T16:54:16.433628434Z",
  "build_snapshot" : false,
  "lucene_version" : "9.4.1",
  "minimum_wire_compatibility_version" : "7.17.0",
  "minimum_index_compatibility_version" : "7.0.0"
},
"tagline" : "You Know, for Search"
}
```

Sin embargo al ejecutar el cliente

Elastic\Transport\Exception\NoNodeAvailableException

No alive nodes. All the 1 nodes seem to be down.

at vendor/elastic/transport/src/NodePool/SimpleNodePool.php:77

```
73 |     }
74 |     $dead++;
75 |   }
76 |
→ 77 |     throw new NoNodeAvailableException(sprintf(
78 |       'No alive nodes. All the %d nodes seem to be down.',
79 |       $totNodes
80 |     ));
81 |   }
```

La documentación nos dice.

```
$hosts = [
    '192.168.1.1:9200', // IP + Port
    '192.168.1.2',     // Just IP
```

```
'mydomain.server.com:9201', // Domain + Port
'mydomain2.server.com',    // Just Domain
'https://localhost',       // SSL to localhost
'https://192.168.1.3:9200' // SSL to IP + Port
];
```

Bien, en mi configuración uso siempre el protocolo https, (en producción) y por extensión configure mi cliente usando la misma metodología, `http://localhost`, probando con `http://127.0.0.1`, etc.

Lo curioso es que haciendo un debug, y volcando la salida del cliente creado, me indicaba que estaba **alive**

```
-nodePool: Elastic\Transport\NodePool\SimpleNodePool { #2856 ▼
  #nodes: array:1 [▼
    0 => Elastic\Transport\NodePool\Node { #2859 ▼
      #uri: GuzzleHttp\Psr7\Uri { #2861 ▼
        -scheme: "http"
        -userInfo: ""
        -host: "127.0.0.1"
        -port: 9200
        -path: ""
        -query: ""
        -fragment: ""
        -composedComponents: null
      }
      #alive: true
    }
  ]
}
```

La solución está en la línea (y la documentación de Elasticsearch que es un poco espesa y deficitaria en muchos sitios). Se trata de no poner el protocolo cuando usemos `http` en lugar de `https`

```
ClientBuilder::create()
->setHosts(['http://<name-of-node-elasticsearch>:9200'])
->build();
```

## Obtener el nombre del container



docker container ls

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
652c6ecb63b9	sail-8.1/app	"start-container"	4 hours ago	Up 4 hours
0.0.0.0:80->80/tcp, 0.0.0.0:5173->5173/tcp, 8000/tcp	sitelight-laravel.test-1			
42370b75132d	docker.elastic.co/kibana/kibana:8.5.0	"/bin/tini -- /usr/l..."	4 hours ago	Up 4
hours	0.0.0.0:5601->5601/tcp	sitelight-kibana		
cf0da3198b67	mysql/mysql-server:8.0	"/entrypoint.sh mysql..."	4 hours ago	Up 4 hours
(healthy)	0.0.0.0:3306->3306/tcp, 33060-33061/tcp	sitelight-mysql-1		
9e89cfc0e2ff	docker.elastic.co/elasticsearch/elasticsearch:8.5.0	"/bin/tini -- /usr/l..."	4 hours ago	Up 4
hours	0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp	sitelight-es01		
9e2be014d4a6	redis:alpine	"docker-entrypoint.s..."	4 hours ago	Up 4 hours
(healthy)	0.0.0.0:6379->6379/tcp	sitelight-redis-1		

El nombre tambien esta en la defición que se hizo en el fichero `docker-compose.yml`

```
elasticsearch:
  image: 'docker.elastic.co/elasticsearch/elasticsearch:8.5.0'
  container_name: sitelight-es01
  environment:
```

# Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Conversor de consulta SQL a DSL para Elasticsearch

## Introducción

El motor de consultas de Elasticsearch, es [DSL](#) y tiene sus peculiaridades. No recuerdo en que version fue, pero se incorporó a Elasticsearch la herramienta [SQL Translate API](#) en mi opinion una de las mejoras cosas que ocurrio para evitar el uso de librerias de terceros, por el miedo escénico que produce el lenguaje DSL.

Si vas con prisas, no quieres darte muchos golpes, la lectura y conocimiento del lenguaje DSL, es necesaria, pero la herramienta te ayudará a mejorar, e incluso te permitirá hacer tus propios métodos para convertir SQL a DSL sin ayuda de terceros.

## Ejemplo

El ejemplo es ejecutado en kibana, herramienta indispensable para jugar con elasticsearch.

Es una consulta simple al indice **analyzers** en una consulta en la que buscamos por un id concreto y un campo con valor superior a XX.

```
POST /_sql/translate
{
  "query": "SELECT * FROM analyzers WHERE modem_id = 1 AND vac1_v > 260"
}
```

Resultado

```
{
  "size": 1000,
  "query": {
    "bool": {
```

```
"must": [  
  {  
    "term": {  
      "modem_id": {  
        "value": 1  
      }  
    }  
  },  
  {  
    "range": {  
      "vac1_v": {  
        "gt": 260,  
        "boost": 1  
      }  
    }  
  }  
],  
"boost": 1  
}  
,  
"_source": false,  
"fields": [  
  {  
    "field": "cos1_a"  
  },  
  {  
    "field": "cos1_m"  
  },  
  {  
    "field": "cos2_a"  
  },  
  {  
    "field": "cos2_m"  
  },  
  {  
    "field": "cos3_a"  
  },  
  {  
    "field": "cos3_m"  
  },  
],
```

```
{
  "field": "datetime",
  "format": "strict_date_optional_time_nanos"
},
{
  "field": "eneact1_kwh"
},
{
  "field": "eneact2_kwh"
},
{
  "field": "eneact3_kwh"
},
{
  "field": "eneact_kwh"
},
{
  "field": "eneapa1_kvah"
},
{
  "field": "eneapa2_kvah"
},
{
  "field": "eneapa3_kvah"
},
{
  "field": "enerea1_kvarh"
},
{
  "field": "enerea2_kvarh"
},
{
  "field": "enerea3_kvarh"
},
{
  "field": "enerea_kvarh"
},
{
  "field": "freq_hz"
},
```

```
{
  "field": "iac1_a"
},
{
  "field": "iac1_a_a"
},
{
  "field": "iac1_m_a"
},
{
  "field": "iac2_a"
},
{
  "field": "iac2_a_a"
},
{
  "field": "iac2_m_a"
},
{
  "field": "iac3_a"
},
{
  "field": "iac3_a_a"
},
{
  "field": "iac3_m_a"
},
{
  "field": "ip"
},
{
  "field": "log_id"
},
{
  "field": "lvac1_v"
},
{
  "field": "lvac2_v"
},
{
```

```
"field": "lvac3_v"
},
{
  "field": "message"
},
{
  "field": "mod_bus_error"
},
{
  "field": "modem_id"
},
{
  "field": "pa1_a_w"
},
{
  "field": "pa1_m_w"
},
{
  "field": "pa1_w"
},
{
  "field": "pa2_a_w"
},
{
  "field": "pa2_m_w"
},
{
  "field": "pa2_w"
},
{
  "field": "pa3_a_w"
},
{
  "field": "pa3_m_w"
},
{
  "field": "pa3_w"
},
{
  "field": "pf"
```

```
},
{
  "field": "pf1"
},
{
  "field": "pf2"
},
{
  "field": "pf3"
},
{
  "field": "powapa_va"
},
{
  "field": "powrea_var"
},
{
  "field": "pp1_va"
},
{
  "field": "pp2_va"
},
{
  "field": "pp3_va"
},
{
  "field": "pr1_a_var"
},
{
  "field": "pr1_m_var"
},
{
  "field": "pr1_var"
},
{
  "field": "pr2_a_var"
},
{
  "field": "pr2_m_var"
},
}
```

```
{
  "field": "pr2_var"
},
{
  "field": "pr3_a_var"
},
{
  "field": "pr3_m_var"
},
{
  "field": "pr3_var"
},
{
  "field": "provider"
},
{
  "field": "response_time"
},
{
  "field": "status_code"
},
{
  "field": "v_event"
},
{
  "field": "vac1_a_v"
},
{
  "field": "vac1_m_v"
},
{
  "field": "vac1_v"
},
{
  "field": "vac2_a_v"
},
{
  "field": "vac2_m_v"
},
{
```



```
    "field": "vac2_v"
  },
  {
    "field": "vac3_a_v"
  },
  {
    "field": "vac3_m_v"
  },
  {
    "field": "vac3_v"
  }
],
"sort": [
  {
    "_doc": {
      "order": "asc"
    }
  }
],
"track_total_hits": -1
}
```

Bien, la consulta para nuestro propósito sería

```
GET /analyzers/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "modem_id": {
              "value": 1
            }
          }
        }
      ],
      {
        "range": {
          "vac1_v": {
            "gt": 260,
```

```
      "boost": 1
    }
  }
},
"boost": 1
}
}
```

“ Es importante si no conoces Elasticsearch que comprendas que hay partes que no deberían existir en los índices de Elasticsearch, que vienen de la mentalidad de datos estructurados, típicos de los motores SQL. Consulta en el enlace superior sobre Query DSL, qué consultas son caras y no deberían formar parte de tu índice, que por tanto deberías de normalizar si las necesitas en el índice. Buena suerte

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Cosas de Elasticsearch (ELK)

# Snapshots y restore

# Backups, snapshot y restore en Elasticsearch 8

## Introducción

Uno de los temas más importantes, como siempre, es el de los backups. En el caso de la [Elasticsearch](#), llamados snapshots. Aunque la Elasticsearch dispone de una buena documentación, dejo por aquí algunos tips que he aprendido con el tiempo.

“Ees importante tener en cuenta la [compatibilidad](#) que existe entre los snapshots y las distintas versiones del Elasticsearch. Existen varios tipos de repositorio. Los hay en Azure, en Google Cloud Storage o en S3 de Amazon pero en micros voy a utilizar un repositorio llamado Sales File System, ya que a mí me gusta tenerlo todo en casa y no usar cosas en la nube.

Snapshot and Restore

## Procedimientos para la creación de un sistema de backups en Elasticsearch (snapshots)

Lo primero que tenemos que hacer es registrar un repositorio para almacenar las instantáneas que vamos a realizar. Para ello debemos disponer de los siguientes permisos:

- privilegios de clase
- privilegios de índice

# Repositorio del sistema de archivos compartido

El [repositorio del sistema de archivos compartidos](#) sólo está disponible así se ejecuta Elasticsearch, en nuestro propio hardware.

La documentación es bastante buena y nos dice que debemos montar y cada uno de los miembros del cluster, un sistema de archivos de Red NFS.

Repositorios or type unknown

Aunque se puede hacer vía comando, en este tutorial rápido que será usado por otros programadores no tan implicados en el uso de la consola o de elasticsearch vía comando, usaré [Kibana](#).

Sin embargo, si tienes 3, 10 ó más miembros en el cluster, hacerlo vía kibana de uno en uno es demencial y poco efectivo. :smile:

## Punto de montaje

Lo primero es tener ya nuestro punto de montaje NFS en todos los miembros del cluster,

```
# df -h
nfs.server:/srv/storage/elasticsearch/ 45T 39T 6.4T 86% /mnt/nfs/elasticsearch
```

Después es tener un punto específico para cada cluster (si tenemos más de uno es necesario, ya que usar el mismo punto es poco usable y desaconsejado por el propio Elasticsearch).

```
# ls -l /mnt/nfs/elasticsearch/cluster02
total 0
2732899439 0 drwxr-xr-x 2 elasticsearch elasticsearch 10 Oct 24 16:57 .
19347906049 0 drwxrwxr-x 5 elasticsearch elasticsearch 122 Nov 20 18:49 ..
```

## Bonus NFS

Uno de los problemas que podéis encontraros en el montaje de un punto NFS es escribirlo por todos los miembros del cluster. Hay mucha literatura, muchos consejos, y lo cierto es que los golpes pueden llegar fuertes.

En mi caso, yo tengo máquinas de backups y NFS, altamente secularizadas ya que son solamente accesible desde las máquinas que comparten NFS o usan SSH, y cada una estas autorizada en el firewall, con un modelo de **denegar todo primero-abrir puerto a quien lo necesita** Así que

uso un NFS sin autenticación de usuario, y esto obliga a que todos usen el mismo usuario. Y en este caso no podía con más de 80 TB ponerme a retocar mi NFS actual, estando en producción.

Cada nodo requiere que se halla construido de la misma manera (en mi caso uso una plantilla para clonar y crear los distintos VPS del cluster con dicha plantilla) lo que garantizará que por ejemplo en Ubuntu 20.04 **elasticsearch** tenga como usuario el **uid 113**, y el **gid 117**

Esto, nos permitirá usar un modo anonimo en todos los miembros del cluster.

En el servidor nfs en `/etc/exports` añadido el punto de compartición

```
/srv/storage/elasticsearch/IP_NODO_001(rw,nohide,insecure,no_subtree_check,async,anonuid=113,anongid=117)
IP_NODO_002(rw,nohide,insecure,no_subtree_check,async,anonuid=113,anongid=117)
...
IP_NODO_nnn(rw,nohide,insecure,no_subtree_check,async,anonuid=113,anongid=117)
```

En cada nodo añadido el montaje nfs a al `/etc/fstab`

```
[nfs_server]:/srv/storage/elasticsearch/ /mnt/nfs/elasticsearch nfs
bg,hard,timeo=1200,rsize=1048576,wsiz=1048576,sec=sys 0 0
```

## Configuración del montaje en Elasticsearch

Es necesario configurar elasticsearch para que lea este nuevo path como repositorio de ficheros.

Para ello necesitamos configurar la variable `path.repo` en el fichero `/etc/elasticsearch/elasticsearch.yml`

“ Esta variable acepta valores separados por comas ,

```
path.repo: /mnt/nfs/elasticsearch/cluster01,/mnt/nfs/elasticsearch/cluster02
```

## Creación en Kibana del repositorio

Entraremos en Kibana a **registrar** nuestro repositorio.

### Registrar el repositorio

En mi caso un mini cluster empezando no he requerido aun de ponerme a hacer un tuning o optimización del sistema de backups, así que lo dejó todo por defecto.

Sólo pongo la situación del repositorio.

### Configuración del repositorio

Con eso ya está creado nuestro repositorio.

Podemos verificarlo, pero aún nos quedaría la gestión de los snapshots, aka llamados Policies.

## Políticas

Básicamente el sistema nos pide que creamos una política de snapshots. Las variantes son muchas, y no es el alcance de este tutorial. En mi caso y por las características de mi sistema (Cluster de VPS formato KVM con snapshots diarios), solo requiero de backups cada hora de cada uno de los índices del sistema, de forma separada, por si ocurriera algún desastre procedente de una manipulación o edición indebida.

Y lo hago por separado, por el método de restauración que va en bloque con cada política de snapshots, no pudiéndose (o al menos yo no lo conozco) separarse cuando es requerida una restauración, como pudiéramos hacer como por ejemplo con la [restauración de un dump completo de Mysql](#)

## Logística

- Nombre que le daremos a la política
- Expresión para nombrar los distintos snapshots. Consultar la ayuda de las expresiones es importante
- Repositorio donde se guardarán los distintos snapshots
- Creación del cron o expresión de horario

⚠ Atención que despista un poco el constructor de cron, porque la sintaxis despista con el **?** que en realidad es la variable del comando que ejecutará. Nada más.

Create Policy or Logistics

## Configuración

Como dije, quiero hacer **sólo** los backups de un **índice** y no de todo el conjunto global. Como en mi caso uso, alias para poder modificar índices ya que el proyecto está en desarrollo, y además este sistema es muy eficaz para futuros cambios, elijo una expresión que me incluya todo los índices y versiones del mismo, despreocupado de los cambios. Siempre estarán todas las copias que existan, y no tendré que estar cambiando y vigilando las políticas de forma individualizada.

Snapshot settings

## Política de retención

Esto es personal y adaptable a las necesidades de cada uno, como todo.

Snapshot retention (optional)

## Revisión

Ya sólo queda revisar la configuración y darle a crear.

Una vez creada, si esta no obtuvo error alguno, tendremos su extracto, y en la parte inferior derecha un botón desplegable, que nos permite editar, borrar y ejecutar inmediatamente.

Snapshot Policies Review

## Restauración

Para la restauración ya lo dejo en tus manos. Es una cuestión compleja esta, y no está a mi alcance explicarla. Lo que sí puedo decir, es, que como todo sistema de backups, **no sirve para nada si no existe una política de comprobación de las copias de seguridad**

Es decir, que tienes que probar en un cluster de pruebas, que esto te funciona, que puedes restaurarlo, que lo documentas, y que regularmente lo compruebas. De lo contrario, patatas como las que se comen en muchos sitios oficiales, hospitales españoles, o grandes empresas, por no tener una política de recuperación de desastres, unas veces por la impericia del supuesto Director de Informática, otras porque una empresa les vendido un cuento de seguridad, cuando en realidad, les engañaron y les cobraron, por nada.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).



# Consultas avanzadas de elasticsearch

## Introducción

Colección de snippets para consultas avanzadas, con elasticsearch

## Filtros

Por:

- id,
- con rango de fechas
- formateo de fechas unixtime a formato humano
- Selección de campos

```
GET work-analyzers/_search
```

```
{
  "_source": ["status_code", "total_consumption_ok", "response_time", "message", "log_id", "datetime"],
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "modem_id": {
              "value": "1544"
            }
          }
        }
      ],
    },
    {
      "range": {
        "datetime": {
```

```
      "gte": "2023-05-08T19:00:00",
      "lte": "2023-05-08T23:00:00"
    }
  }
}
]
}
},
"docvalue_fields": [
  {
    "field": "datetime",
    "format": "yyyy-MM-dd HH:mm:ss"
  }
]
}
```

# Instalando Elasticsearch + Kibana en local con Docker

## Introducción

Al final con la aparición de [Laravel Herd](#) y por las cosas que hago, prefería dismantelar en mis proyectos que no necesitan **Laravel sail** por ser modernos y actualizados. En cuanto a [Elastic Stack](#) preferí hacerlo vía docker dado su carácter de uso puramente para testing., al menos de momento, en vez de optar por una instalación local 100%.

Así que te dejo como lo hice en mi mac silicon.

## Instalación

### Elastic search single node

[Oficial](#)

#### Crear un red para elastic

```
> docker network create elastic
```

#### Bajarse la imagen

```
> docker pull docker.elastic.co/elasticsearch/elasticsearch:8.11.2
8.11.2: Pulling from elasticsearch/elasticsearch
Digest: sha256:e40b9d3d523f2fe4dc851ad2cc5570f28a58ca6c4efb566cc9688dcdf0df8dec
Status: Image is up to date for docker.elastic.co/elasticsearch/elasticsearch:8.11.2
docker.elastic.co/elasticsearch/elasticsearch:8.11.2
```

# Verificar la imagen

Se requiere tener instalado en tu entorno [Cosign](#) si quieres verificar la imagen.

- Primero bajarse la firma de la imagen

```
> wget https://artifacts.elastic.co/cosign.pub
--2023-12-10 09:17:56-- https://artifacts.elastic.co/cosign.pub
Resolviendo artifacts.elastic.co (artifacts.elastic.co)... 34.120.127.130
Conectando con artifacts.elastic.co (artifacts.elastic.co)[34.120.127.130]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 178 [application/x-mspublisher]
Grabando a: «cosign.pub»

cosign.pub
100%[=====
=====
=====>] 178 --KB/s en 0s

2023-12-10 09:17:57 (28,3 MB/s) - «cosign.pub» guardado [178/178]
```

- Después verificarla

```
> cosign verify --key cosign.pub docker.elastic.co/elasticsearch/elasticsearch:8.11.2

Verification for docker.elastic.co/elasticsearch/elasticsearch:8.11.2 --
The following checks were performed on each of these signatures:
- The cosign claims were validated
- Existence of the claims in the transparency log was verified offline
- The signatures were verified against the specified public key
```

## Instalar elasticsearch container

Esta es una variante del manual. El motivo es claro. Es crear dos volúmenes:

- Uno con el directorio donde se guardan los índices.
- Otro donde se guarda la configuración.

“ Si tenías una instalación antigua tipo, o una instalación con volúmenes diferentes, puede que tengas que empezar de o eliminándolos.

```
> docker run --name es01 --net elastic -p 9200:9200 -v es01-data:/usr/share/elasticsearch/data -v es01-config:/usr/share/elasticsearch/config -it -m 1GB docker.elastic.co/elasticsearch/elasticsearch:8.11.2
```

...

□ Elasticsearch security features have been automatically configured!

□ Authentication is enabled and cluster connections are encrypted.

**i** Password for the elastic user (reset with ``bin/elasticsearch-reset-password -u elastic``):

w78bj=MMasWeb6S1mnZR

**i** HTTP CA certificate SHA-256 fingerprint:

464224b5bc542f16b37d86d9038633bfe0a9a01acaab7fe7fe78ff6e838f60dc

**i** Configure Kibana to use this cluster:

- Run Kibana and click the configuration link in the terminal when Kibana starts.
- Copy the following enrollment token and paste it into Kibana in your browser (valid for the next 30 minutes):

eyJ2ZXliOiI4LjExLjliLCJhZHliOlsiMTcyLjI3LjAuMjo5MjAwIl0sImZncil6IjQ2NDIyNGI1YmM1NDJmMTZiMzdkODZkOTAzODYzM2JmZTBhOWEwMWFjYWFiN2ZIN2ZINzhmZjZlODM4ZjYwZGMiLCJrZXkiOiIxaXpkVW93QjJlZDFvZmRMRTBCUTo2ZWVtYmJLZ1FkT1U4N1YyN0pkQnpBln0=

**i** Configure other nodes to join this cluster:

- Copy the following enrollment token and start new Elasticsearch nodes with ``bin/elasticsearch --enrollment-token <token>`` (valid for the next 30 minutes):

eyJ2ZXliOiI4LjExLjliLCJhZHliOlsiMTcyLjI3LjAuMjo5MjAwIl0sImZncil6IjQ2NDIyNGI1YmM1NDJmMTZiMzdkODZkOTAzODYzM2JmZTBhOWEwMWFjYWFiN2ZIN2ZINzhmZjZlODM4ZjYwZGMiLCJrZXkiOiIyQ3pkVW93QjJlZDFvZmRMRTBCZzp4ZINfc3R5QIRtYVFzUVZBbFBGcE13In0=

If you're running in Docker, copy the enrollment token and run:

```
`docker run -e "ENROLLMENT_TOKEN=<token>" docker.elastic.co/elasticsearch/elasticsearch:8.11.2`
```

# Instalar kibana

## Bajar la imagen

```
docker run --name kib01 --net elastic -p 5601:5601 -v kib01-config:/usr/share/kibana/config
docker.elastic.co/kibana/kibana:8.11.2
```

...

```
[2023-12-10T08:37:49.859+00:00][INFO ][root] Holding setup until preboot stage is completed.
```

```
i Kibana has not been configured.
```

```
Go to http://0.0.0.0:5601/?code=812523 to get started.
```

Acudiremos al navegador con la [url mostrada](#), y allí usaremos el **enrollment token** para terminar la configuración.

Con eso ya tendremos configurado Kibana + Elasticsearch en docker.

## Preparar las cosas para userse en nuestra app

Necesitamos el certificado de elastic

```
> export ELASTIC_PASSWORD="w78bj=MMasWeb6S1mnZR"
> docker cp es01:/usr/share/elasticsearch/config/certs/http_ca.crt ~/Sites/certificates/http_ca_local.crt
Successfully copied 3.58kB to /Users/abkrim/Sites/certificates/http_ca_local.crt
```

Yo lo hago así, guardar en una carpeta especial el certificado, porque uso un único docker para cada servicio del stack de Elasticsearch, para todos mis proyectos de desarrollo.

## Prueba de concepto

Una vez arriba ambos containers puedo probar elastic en el shell, y así saber si luego la configuración de mi app de Laravel pasaran con los datos que tengo.

“ Uso dos variables de shell, evidentemente

```
> curl --cacert $ELK_DOCKER_CA_BUNDLE -u elastic:$ELASTIC_PASSWORD https://localhost:9200

{
  "name" : "9618d13f0939",
```

```
"cluster_name" : "docker-cluster",
"cluster_uuid" : "-wSu0Nm-QviovkB3rW6f5w",
"version" : {
  "number" : "8.11.2",
  "build_flavor" : "default",
  "build_type" : "docker",
  "build_hash" : "76013fa76dcbf144c886990c6290715f5dc2ae20",
  "build_date" : "2023-12-05T10:03:47.729926671Z",
  "build_snapshot" : false,
  "lucene_version" : "9.8.0",
  "minimum_wire_compatibility_version" : "7.17.0",
  "minimum_index_compatibility_version" : "7.0.0"
},
"tagline" : "You Know, for Search"
}
```

Con eso ya tengo verificación de los datos que necesita mi app para funcionar en local con mis dockers del stack de elasticsearch.

# Upgrade

En caso de trabajar sin `docker compose` Ver [Arranque, Actualización, y cosas de Elastic con Docker](#)

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Truncate index

## Truncate index.

## Reindexacion conservando el mapping

No probado.

```
$client = ClientBuilder::create()->build();

// Crear un nuevo índice temporal
$tempIndex = 'temp_index';
$client->indices()->create(['index' => $tempIndex]);

// Reindexar desde el índice temporal al índice original (sin documentos)
$response = $client->reindex([
  'body' => [
    'source' => [
      'index' => $tempIndex
    ],
    'dest' => [
      'index' => 'nombre_de_tu_indice'
    ]
  ]
]);

// Eliminar el índice temporal
$client->indices()->delete(['index' => $tempIndex]);
```

## Borrado y receacion



```
$client = ClientBuilder::create()->build();
```

```
// Eliminar el índice
```

```
$client->indices()->delete(['index' => 'nombre_de_tu_indice']);
```

```
// Recrear el índice
```

```
$client->indices()->create(['index' => 'nombre_de_tu_indice']);
```

# Arranque, Actualización, y cosas de Elastic con Docker

## Introducción

De vez en cuando hay que actualizar las herramientas del paquete Elasticsearch. Una de ellas es la actualización de Elasticsearch cuando usamos Docker.

Como ya no uso Sail, esto lo hago con Docker, y para que no se me pase, dejo escrito algunos tips.

## Discos Shared

### Editado el 23/07/24 Version 8.14.3

He dejado de usar el data shared debido a que la actualización 8.14.X no pasa con el data. No tengo tiempo.

Estaba usando la configuración en modo shared, y esto en la última actualización me trajo líos.

`/usr/share/kibana/config` por un problema con los certificados.

Para no perder el tiempo, opté por eliminar el disco de la configuración desde mi **Docker Desktop**.

Lo único es que hay que hacer de nuevo el enrollment de Kibana con Elastic, pero es súper sencillo y solo es seguir las instrucciones del navegador, y ejecutar los dos comandos: uno en el shell de Elasticsearch y otro en el de Kibana.

## Backup Editado 05/10/2024

Como las versiones van subiendo, es bueno tener si en tu entorno de desarrollo, ya tienes cosas que no quieres volver a importar o dedicar tiempo, hacer un backup del contenedor.

Como en mi caso, los datos estan contenidos en un contedor que podemos hacer un backup:

```
docker export es01 -o /path_backup/es01_backup.tar
```

Si alguna vez queremos restaurarlo:

```
cat /path_backup/es01_backup.tar | docker import - es01
```

## Error en la actualizacion

```
[2024-02-09T15:45:45.275+00:00][INFO ][http.server.Preboot] http server running at http://0.0.0.0:5601
[2024-02-09T15:45:45.579+00:00][INFO ][root] Kibana is shutting down
[2024-02-09T15:45:45.598+00:00][FATAL][root] Reason: ENOENT: no such file or directory, open
'/usr/share/kibana/data/ca_1702197563664.crt'
Error: ENOENT: no such file or directory, open '/usr/share/kibana/data/ca_1702197563664.crt'
    at Object.openSync (node:fs:603:3)
    at readFileSync (node:fs:471:35)
    at readFile (/usr/share/kibana/node_modules/@kbn/core-elasticsearch-server-
internal/src/elasticsearch_config.js:524:31)
```

Como veis, el proceso no actualiza el fichero de configuración y el cambio en esta versión va relacionado con los certificados. Seguramente se podrá solventar de otra manera, pero he preferido ir a lo rápido.

Lo primero es eliminar el volumen de configuración, ya que no es apropiado.

## Arranque para actualizar

Aconsejable leer:

- [Docker Elasticsearch](#)
- [Docker Kibana](#)

Pero aconsejable leer mejor el enlace a la documentación oficial, pues hay una gracia en el ejemplo `-p 9200:9200 -p 9300:9300` que hara que fracase tu instalación y comiences a dar vueltas en el mundo de Elasticsearch.

## Editado el 23/07/24 Version 8.14.3

He eliminado el volumen data, pues eso hace fallar la instalacion. No tengo tiempo de revisarlo.

```
tag="8.14.3" // Tag de la versión que queremos actualizar
docker rm es01 // NO queremos el container
docker volume rm es01-config // NO queremos el volumen de la config
docker run --name es01 --net elastic -p 9200:9200 -it -m 1GB docker.elastic.co/elasticsearch/elasticsearch:$tag
docker run --name kib01 --net elastic -p 5601:5601 docker.elastic.co/kibana/kibana:$tag
```

En kibana tendremos un mensaje final

[2024-05-11T06:07:59.739+00:00][INFO ][preboot] "interactiveSetup" plugin is holding setup: Validating Elasticsearch connection configuration...

```
[2024-05-11T06:07:59.781+00:00][INFO ][root] Holding setup until preboot stage is completed.
```

i Kibana has not been configured.

Go to <http://0.0.0.0:5601/?code=811436> to get started.

Esto nos obliga a obtener el enrollment en la instancia de Elasticsearch

```
sh-5.0$ bin/elasticsearch-create-enrollment-token --scope kibana
eyJ2ZXliOiI4LjEzLjQlLCJhZHliOiIsMTcyLjI3LjAuMj05MjAwIiw0sImZnciI6IjQ2NDIyNGI1YmM1NDJmMTZiMzdKODZkOTAz
ODYzM2JFKEFAKEFAKEFAKLEFAKErZXkiOiIyVFpJWm84QnIwV1BJX0FsYTdPRDowTVJ3WGR3RFQzaTI2ZXVzWlIiCNIIBI
n0=
```

“ Si por alguna razón, falla es debido a que arrancaste ambas instancias de manera distinta y no se pueden comunicar. Arrancalas al unisono o primero elastic y luego kibana, en un mismo comando, y recuerda que deberas cambiar la url del enrollment, porque cambia en cada arranque.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).



# Llamadas en Kibana (o para usar con cUrl) para Elasticsearch de uso común

## Introducción

Hay una serie de llamadas con curl o Kibana que son de uso rápido. Ayudan mucho en el día a día con Elasticsearch.

“ Las llamadas fuera de kibana requiere ser adaptadas a sus posibilidades. Ojo a esto, que muchos con el copy & paste no lo comprenden.

## Convenciones

- **[sustituir]** incluidos los `[]` por el valor correspondiente
- En cUrl, con el shell, `${var}` necesitamos declarar la variable.

## Obtener el mapping de un índice

```
GET [indice]/_mapping
```

## Estado del cluster (Solo cUrl)

Esta solo es para cUrl.

```
curl --cacert "${path_http_ca.crt}" --user ${user}: -XGET  
"${url_port}/_cluster/health?wait_for_status=yellow&timeout=50s&pretty"
```

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Cosas de docker

Peleando con docker prefiero tener a mano unos tips



# Información de los contenedores docker

## Introduccion

Aunque usemos herramientas de desktop, etc, muchos comandos son necesarios, ya no solo por su eficacia, sino porque muchas veces las herramientas no cubren todos los aspectos, y porque otras muchas, los manuales y la información que tenemos a mano, habla de ellos, y no de las herramientas de trabajo.

## Contenedores

Muchas, muchismas veces es necesario entrar o conocer datos de los contenedores.

## Listado

```
docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
b52d46f0eaa1	sail-8.1/app	"start-container"	4 minutes ago	Up 4 minutes	0.0.0.0:80->80/tcp, 0.0.0.0:5173->5173/tcp, 8000/tcp	sitelight-laravel.test-1
46868801fab5	kibana:8.4.3	"/bin/tini -- /usr/l..."	4 minutes ago	Up 4 minutes	0.0.0.0:5601->5601/tcp	sitelight-kibana
be49291ac94e	mysql/mysql-server:8.0	"/entrypoint.sh mysql..."	4 minutes ago	Up 4 minutes (healthy)	0.0.0.0:3306->3306/tcp, 33060-33061/tcp	sitelight-mysql-1
2cbdb7150093	elasticsearch:8.4.3	"/bin/tini -- /usr/l..."	4 minutes ago	Up 4 minutes	0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp	sitelight-es01
3b17bd32cff8	redis:alpine	"docker-entrypoint.s..."	4 minutes ago	Up 4 minutes (healthy)	0.0.0.0:6379->6379/tcp	sitelight-redis-1
01eecfe2c5e1	mysql:latest	"docker-entrypoint.s..."	2 weeks ago	Exited (0) 2 weeks		

# Ejecutar un comando en un contenedor activo

```
docker exec -it 2cbdb7150093 /usr/share/elasticsearch/bin/elasticsearch-reset-password -a -u elastic
WARNING: Owner of file [/usr/share/elasticsearch/config/users] used to be [root], but now is [elasticsearch]
WARNING: Owner of file [/usr/share/elasticsearch/config/users_roles] used to be [root], but now is [elasticsearch]
This tool will reset the password of the [elastic] user to an autogenerated value.
The password will be printed in the console.
Please confirm that you would like to continue [y/N]y
```

“NOTA: este comando ya no es necesario en un despliegue con docker de elasticsearch y kibana. Entra sin seguridad activa, ni login al cluster.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Wordpress en docker para desarrollo.

## Cómo usar docker para desarrollo o gestión de incidentes con Wordpress

A veces es necesario tener un entorno de desarrollo para realizar determinadas acciones, como puede ser una actualización conflictiva, un cambio de versión u otras que no se van a necesitar de un entorno que no sea el de producción.

[Docker+](#), es la elección perfecta, al menos para mí, acostumbrado a trabajar para todo con él, acompañandome todo mi desarrollo en mi portatil, como buen nómada que soy.

En el caso descrito habitual relativo a [WordPress](#), muy, muy, muy antiguos, que usan determinadas versiones de PHP, y que al intentar actualizarlas en producción, puede ser un completo desastre, y aunque pese a tener un sistema basado en **rsync** y **mysqldump** en la comnsola de comandos, la web que tenía que actualizar no tenía esa posibilidad de poder estar perdiendo 45 minutos en producción así qué la dockerize.

“ Este tip es valido, para cualquier sitio web que queramos actualizar con un minimo de profesionalidad y seguridad

## Dockerización de Wordpress

Habitualmente uso [Laravel Sail](#), pero sin embargo, he preferido hacer el tema de WordPress con una instalación independiente y usando Docker composer.

Los datos son específicos a mi entorno, pero si no eres de copiar y pegar y tratas de comprender cada paso, podrás trabajar donde quieras y como quieras que se adapte a tus necesidades. Esto supone diertas configuraciones de docker que pueden hacer que en otros escenarios o configuraciones no funcione lo que digo, sobre todo si la configuración específica afecta al modo de red. Ver

[Docker Tips :: Algos sobre redes](#)

“ Se recomienda leer los [tips](#)

# Creación de del entorno de trabajo

## Directorio de trabajo

```
mkdir ~/Sites/myweb  
cd $_
```

## Creación del docker-compose.yml

La creación la haremos en un principio ajustándose al escenario de funcionamiento, para luego progresivamente ir subiendo a la versión que queremos, para lidiar con todos los problemas de plugins, templates desactualizados.

Por tanto lo suyo no es copiar, por ejemplo las imágenes de docker que se indican, sino las que necesitamos obteniendo la información de las tags, en el hub de docker.

La imagen de [Wordpress](#) deberá ser la adecuada al escenario de nuestro sitio con problema en producción.

Versión de PHP, apache o nginx, fpm,...

Más adelante iremos subiendo a la que queremos.

“ Uso docker desktop por muchas razones, y lo recomiendo.

Editamos el fichero `editor ~/Sites/myweb/docker-compose.yml`

services:

db:

# We use a mariadb image which supports both amd64 & arm64 architecture

# image: mariadb

# If you really want to use MySQL, uncomment the following line

image: mysql:8

command: '--default-authentication-plugin=mysql\_native\_password'

volumes:

- db\_data:/var/lib/mysql

restart: always

environment:

- MYSQL\_ROOT\_PASSWORD=somewordpress

- MYSQL\_DATABASE=la\_misma\_que\_en\_producción

- MYSQL\_USER=el\_mismo\_que\_en\_producción

- MYSQL\_PASSWORD=PasSW0rD\_que\_en\_producción

ports:

- "3306:3306"

wordpress:

image: wordpress:php7.4-fpm-alpine

volumes:

- ./wp\_data:/var/www/html

ports:

- 80:80

restart: always

environment:

- WORDPRESS\_DB\_HOST=db

- WORDPRESS\_DB\_USER=el\_mismo\_que\_en\_producción

- WORDPRESS\_DB\_PASSWORD=PasSW0rD\_que\_en\_producción

- WORDPRESS\_DB\_NAME=la\_misma\_que\_en\_producción

volumes:

db\_data:

wp\_data:

## Explicación

- **MySQL** lo montamos con un volumen, para hacer permanente sus datos.
- La instancia de **WordPress** lo mismo, poniendo el directorio de montaje en el directorio en el que vamos a clonar el sitio, apuntando al directorio `/var/www/html` del volumen.

# Copia del contenido de producción

La mejor opción es `rsync` una herramienta **imprescindible** no sólo para los administradores de sistemas, sino para los webmaster o mantenedores de sitios web.

```
$ rsync -avzzz --progress -e "ssh" user_remoto@dominio_remoto.tld:/path/al/sitio/ ~/Sites/myweb/wp_data/
receiving file list ...
33791 files to consider
./
license.txt
    19915 100% 18.99MB/s 0:00:00 (xfer#1, to-check=33778/33791)
readme.html
    7402 100% 7.06MB/s 0:00:00 (xfer#2, to-check=33774/33791)
wp-config-sample.php
    3013 100% 2.87MB/s 0:00:00 (xfer#3, to-check=33766/33791)
wp-config.php
    3601 100% 3.43MB/s 0:00:00 (xfer#4, to-check=33765/33791)
...
...
wp-includes/widgets/class-wp-widget-text.php
    21342 100% 20.99kB/s 0:00:00 (xfer#2301, to-check=0/33791)
sent 479944 bytes received 860743 bytes 536274.80 bytes/sec
total size is 1363857930 speedup is 1017.28
```

## Copia o dump de la base de datos de producción

Podemos usar nuestra herramienta preferida, como TablePlus, Navicat, phpMyAdmin... o el propio shell.

El caso es hacer una copia completa.

Me gusta usar el shell.

Como no tengo acceso mysql remoto, y no me apetece montar un túnel, hago el backup en remoto y luego lo bajo con `rsync`

## Servidor remoto

```
mysqldump --no-tablespaces --opt -u <wp_user> -p <database> > ~/<database>.sql
Enter password:
```

# Servidor local

Ahora traemos el fichero de dump

```
rsync -avzz --progress -e "ssh -p 2244" user_remoto@remoto.tld:/home/myuser/<database>.sql .
receiving file list ...
1 file to consider
<database>.sql
 130317359 100%  7.80MB/s   0:00:15 (xfer#1, to-check=0/1)
sent 25352 bytes  received 35045636 bytes  2004056.46 bytes/sec
total size is 130317359  speedup is 3.72
```

⚠ Atención al `.` final del comando que quiere decir `cópiame aquí`

## Restaurar la copia

### Obtener información del container

```
docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
9b780079bb	wordpress:latest	"docker-entrypoint.s..."	40 minutes ago	Up 40 minutes	0.0.0.0:80->80/tcp
myweb-wordpress-1					
c591fdcae17a	mysql:8	"docker-entrypoint.s..."	40 minutes ago	Up 40 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp
myweb-db-1					

## Volcar el backup

Usaremos el **id** del contenedor de mysql `c591fdcae17a` para ejecutar un comando en él.

```
cat <database>.sql | docker exec -i c591fdcae17a /usr/bin/mysql -u root --password=somewordpress
<database>
mysql: [Warning] Using a password on the command line interface can be insecure.
```

En mi caso por vaquería y comodidad, uso TablePlus para muchas cosas, y tiene un importador. Pero siempre hay una vez para fallar, y esta vez, fue al importar una bd con 45Mb, (he importado en table plus bd de 6 GB sin problemas). Se terminaba siempre en una de las tablas. Fue hacerlo en el el shell, y funcionó sin problemas.

Levantamos la imagen

```
docker compose up -d
```

`docker compose` type unknown

Para probar que esta todo correcto, solo tendríamos que ir a nuestro navegador y solicitar localhost

## Devolver a su sitio.

Tras actualziar y ajustar todo a las últimas versiones tanto de wordpress, como de PHP, Mysql o MariaDb, deberemos subirlo a producción.

Para devolver a su sitio la actualización completa es sencillo. Usamos la misma técnica de `rsync` + `mysqldump` pero al revés.

## Mysqldump

“ Esta vez usaremos el **name** del container en lugar del **ID**

```
docker exec -i myweb-db-1 /usr/bin/mysqldump --opt -u root --password=somewordpress --databases  
<database> --skip-comments > dump-updated.sql
```

## Subir los cambios a producción

```
~/Sites/myweb $ rsync -avzz --progress --delete -e "ssh -p 2244" ~/Sites/myweb/wp_data/  
user_remoto@dominio_remoto.tld:/path/al/sitio/  
~/Sites/myweb $ rsync -avzz --progress --delete -e "ssh -p 2244" ~/Sites/myweb/dump-updated.sql  
user_remoto@dominio_remoto.tld:/path/al/user/
```



# Restaurar la base de datos en el servidor de producción

```
Mysql -u <user_de_mysql> -p <bd_produccion> < ~/dump-updated.sql
```

## Tips.

- Editar `/etc/hosts` para que apunte a a local (127.0.0.1) y evitar líos de redirecciones. En el caso de MacOSx, sería en `/private/etc/hosts`

```
127.0.0.1 myweb.com
```

- Usar un navegador permisivo o configurado para no usar https (safari, opera...). Si tenemos algun plugin que fuerce la redirección `http` a `https` deberías desactivarlo.
- Usar una página `info.php` para tener las cosas claras. Simplemente debe tener `<?php phpinfo();` y ponerla en el `public` de nuestra web.
- Restaurar el backup (mejor desde shell como siempre)
- Los datos de creación de mysql, user y password deben ser los mismos en el par, producción-desarrollo.
- El parámetro del fichero `wp-config.php` `define('DB_HOST', 'localhost');` debe estar definido a 'localhost'.
- **editor** es el alias de mi editor preferido.
- `$` indica que estamos en una cuenta de usuario y su path y `#` es root
- `<variable>` indica que debe ser sustituido por los valores apropiados incluyendo en el reemplazo los `<>`
- Atención a la barra `/` al final de los parámetros de `rsync` porque no es lo mismo con ella que sin ella.

A trabajar con el tema.

## Tips globales

- Una vez que ya hemos conseguido la actualización menor necesaria para poder pasar a una versión más alta de php sin problemas, simplemente debemos editar el fichero `docker-compose.yml` y solicitar la nueva versión de php, y reconstruir el volumen de Wordpress. (Borrar container y volumen wordpress.
- Siempre debemos trabajar inicialmente con las mismas versiones y software que como está en producción. Hay que recordar que existe, versiones de Mysql, MariaDb (no son iguales no... ojo a ese datos que más de un quebradero de cabeza os podéis llevar un

día), versiones de PHP, con FPM, mod, con Nginx, con Apache, etc.

- Si no te sientes cómodo con el shell, al menos elige un [hosting](#) que tenga una herramienta de backup como Jetbackup que te permite crear snapshots antes de lidiarte con cambios.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ninguna obligación ni responsabilidad por parte de [Castris](#). Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Proyectos con path public, public\_html, app en Docker con Apache+PHP-FPM

## Introducción

Existen desarrollos en software web, que siguen una pauta moderna y más segura, consistente en exponer solamente al público el fichero index.php dejando en el `doc_root` de **Apache** o **Nginx** el directorio que lo contiene, y no todos el desarrollo. Ejemplos clásicos son, Laravel, Symfony, Slim, y otros muchos. aunque si no eres amigo de los marcos de trabajo (framework), bueno es que también uses esta formula de trabajo y no meter todo en el directorio del `doc_root` del servidor Web.

En mi último trabajo he tenido que lidiar con una aplicación obsoleta, en PHP 5.6 desarrollada con el marco Zend Framework, que estaba en un cPanel pero con las estructura correcta por debajo del `public_html`. (Lastima que encima puso un blog con Wordpress, y se le olvido actualizarlo)

Esta guía te mostrara el camino de la creación de los contenedores necesarios para Apache, PHP-Fpm y MySQL para ejecutar tu desarrollo, usando **Docker compose**

## Docker Apache, PHP FPM setup with docker compose y MySQL

Cuando uno llega aquí, lo único que queda es dockerizar nuestro area de trabajo, ya que desarrollar en el servidor está... **prohibido**

El proceso esta indicado para un escenario concreto, pero con los ajustes necesarios, puedes elegir la versión de PHP, la de MySQL/MariaDb/Percona que necesites (siempre trabaja con el mismo escenario que en producción).

# Requisitos

Necesitamos tener instalado en nuestro equipo Docker.

Generaremos la siguiente estructura en nuestro proyecto

```
proyecto
├─ Dockerfile
├─ docker-compose.yml
├─ apache/
│  │ └─ apache.vhost.conf
│  │ └─ Dockerfile
├─ public/
│  └─ index.php
├─ vendor/
└─ ...
```

Algo así quedaría

- **Dockerfile** global que contiene la configuración necesaria para crear y preparar el escenario para un PHP con PHP-FPM y sus extensiones necesarias en las imágenes necesarias.
- **docker-compose.yml** que crear los contenedores necesarios y los unirá.
- **apache/apache.vhost.conf** que contendrá la información necesaria para que Apache sea el servidor web de nuestro proyecto.
- **apache/Dockerfile**
- **public** que será el directorio root de nuestro proyecto, expuesto a internet por el par Apache/PHP-FPM

## #### Crear la imagen PHP FPM

Te damos las instrucciones necesarias a docker para construir la imagen. El ejemplo es el básico que me permite usar y trabajar en esta dockerización, con el proyecto Zend en un PHP 5.6. Si necesitas más extensiones, o algún cambio necesitaras un trabajo extra para añadirlo.

Consulta siempre en [Hub Docker](#) las posibilidad que tienes, y si quieres un consejo, usa siempre imágenes oficiales. Te ahorraras muchos disgustos.

“ Los comentarios indican modificación sobre la imagen oficial. Si usas una no oficial, o versiones oficiales basadas en otro so, como alpine, etc. los paths indicados pueden sufrir variaciones

```
FROM php:5.6-fpm
```

```
RUN docker-php-ext-install mysqli pdo pdo_mysql
```

```
## Esta linea la puedes comentar si tu proyecto tiene un estructura plana, como puede ser un Wordpress.
```

```
Donde todo está en el mismo nivel del doc_root
```

```
RUN echo 'php_admin_value[doc_root] = /var/www/html/public' >> /usr/local/etc/php-fpm.d/www.conf
```

- Docker hará un pull **php:5.6-fpm**
- Si quieres o necesitas instalar una extensión adicional con Docker usa el commando **docker-php-ext-install** dentro de doc usando **RUN**

## Crear Apache

Necesitamos crear la imagen con el servidor web Apache, atendiendo a que usaremos PHP-FPM como manejador de PHP. Así que usaremos un fichero de configuración del host virtual, y usaremos **apache/apache.vhost.conf** para este propósito.

```
# La configuracion esta adaptada al escenario indicado public/
```

```
# Si no es así deberá modificarse al típico /var/www/html/ allí donde este usándose public/
```

```
# Establezca el nombre del servidor en localhost
```

```
ServerName localhost
```

```
# Configure un VirtualHost para manejar solicitudes en el puerto 80
```

```
<VirtualHost *:80>
```

```
    # Solicitudes de proxy PHP para portar el contenedor PHP-FPM 9000
```

```
    ProxyPassMatch ^/(.*\.php(/.*)?)$ fcgi://php-fpm:9000/var/www/html/public/$1
```

```
    # Set the DocumentRoot for the virtual host
```

```
    DocumentRoot /var/www/html/public
```

```
    # Directory configuration for the DocumentRoot
```

```
    <Directory /var/www/html/public>
```

```
        DirectoryIndex index.php
```

```
        Options Indexes FollowSymLinks
```

```
        AllowOverride All
```

```
        Require all granted
```

```
    </Directory>
```

```
    # Definir los destinos CustomLog y ErrorLog
```

```
    CustomLog /proc/self/fd/1 common
```

```
ErrorLog /proc/self/fd/2
</VirtualHost>
```

- La aplicación se ejecutara en modo localhost y responderá a las peticiones que se hagan la puerto 80.
- Apache reenviará las peticiones php al container PHP-FPM, cuyo nombre es 'php-fpm'. Este nombre será usado en el `docker-compose.yml` del proyecto.
- `/var/www/html/public/` será el directorio por defecto de tu sitio virtual de Apache. Por tanto sería el index o fichero que este allí.
- Si deseas depurar tu aplicación, cree destinos CustomLog y ErrorLog para administrar los registros de la aplicación y los de Docker.

Ahora construiremos nuestro **Dockerfile** para construir la imagen de nuestro servidor web Apache

```
# httpd base image
FROM httpd:2.4

# Copy the Apache virtual host configuration file to the container
COPY ./apache/apache.vhost.conf /usr/local/apache2/conf/extra/apache.vhost.conf

# Activar módulos de Apache para garantizar una funcionalidad adecuada.
RUN sed -i \
    # Descomente la configuración de mod_rewrite para habilitar el control de reescritura contenido.
    -e '/#LoadModule rewrite_module/s/^#//g' \
    # Descomente la configuración de mod_rewrite para habilitar el control de caducidad del contenido.
    -e '/#LoadModule expires_module/s/^#//g' \
    # Descomente la configuración de mod_deflate para habilitar la compresión
    -e '/#LoadModule deflate_module/s/^#//g' \
    # Descomente la configuración del proxy para habilitar su uso.
    -e '/#LoadModule proxy_module/s/^#//g' \
    # Descomente la configuración de mod_proxy_fcgi para habilitar el módulo proxy FastCGI
    -e '/#LoadModule proxy_fcgi_module/s/^#//g' \
    /usr/local/apache2/conf/httpd.conf

# Incluir el archivo de configuración del host virtual en la configuración principal de Apache.
RUN echo "Include /usr/local/apache2/conf/extra/apache.vhost.conf" >> /usr/local/apache2/conf/httpd.conf
```

## Crear el container

Bien es hora de armarlo todo.



En mi caso, no uso PHPMYAdmin así que no instalaremos nada adicional. Nunca lo he usado porque nunca me ha parecido correcto tenerlo instalado. En su lugar uso un programa de consola, y me conecto via tunnel SSH. Pero os dejo comentada la opción. Gracias [TablePlus](#)

```
version: '3.9'
```

```
services:
```

```
  php-fpm:
```

```
    build:
```

```
      context: .
```

```
      dockerfile: Dockerfile
```

```
  volumes:
```

```
    - ./var/www/html
```

```
  depends_on:
```

```
    - mysql-db
```

```
  apache-httpd:
```

```
    build:
```

```
      context: .
```

```
      dockerfile: ./apache/Dockerfile
```

```
  volumes:
```

```
    - ./var/www/html
```

```
  ports:
```

```
    - "8080:80"
```

```
  depends_on:
```

```
    - php-fpm
```

```
    - mysql-db
```

```
  mysql-db:
```

```
    image: mariadb:10.1
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD: password
```

```
      MYSQL_DATABASE: base_de_datos
```

```
      MYSQL_USER: user_datos
```

```
      MYSQL_PASSWORD: contraseña
```

```
  ports:
```

```
    - "3336:3306"
```

```
  volumes:
```

```
    - mysql-data:/var/lib/mysql
```

```
# Mac not working
```

```
# phpmyadmin:
# image: phpmyadmin/phpmyadmin:latest
# links:
#   - mysql-db
# ports:
#   - "8081:80"
# environment:
#   PMA_HOST: mysql-db
#   MYSQL_ROOT_PASSWORD: password
volumes:
  mysql-data:
```

- `./var/www/html` esto indica que crearemos un **volumen** que montara en la raíz de nuestro proyecto `.` en `/var/www/html/` en el container.
- El puerto expuesto para acceder será el 8080
- Crearemos un container para **MySQL** que tendrá persistencia de datos, en un volumen **mysql-data** que permitirá tener persistencia de datos.
- Mi consejo es que uses los mismo datos para el usuario, base de datos, y contraseña relativos a tu proyecto, que los que usas en producción, por si realizas sincronizaciones. El root no lo toques.
- El puerto expuesto es el 3336 ya que tengo en local, una instalación **Herd** que es la que uso habitualmente y no me apetece apagarla.

## Testing

Para evitar perdidas de tiempo, es bueno hacer dos comprobaciones.

- Que el container esta preparado para trabajar con PHP en el directorio `public`
- Que la conexión con Mysql funciona.

### info.php

Crea un fichero `public/info.php`

```
<?php phpinfo();
```

### index.php

Si ya tienes uno renombrarlo, para más tarde y crea un nuevo en `public/index.php`

💡 También pues añadir al código al principio del index.php y salir con un `exit();`



```
<?php
$host = 'mysql-db';
$user = 'root';
$pass = 'password';
$db = 'mysql';

$conn = new mysqli($host, $user, $pass, $db);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

echo "Connected to MySQL successfully";

$conn->close();
?>
```

# Construir los contenedores

Bueno, ya esta todo.

```
docker-compose up --build -d
docker-compose up --build -d
[+] Building 2.7s (16/16)
FINISHED
    docker:desktop-linux
=> [php-fpm internal] load build definition from
Dockerfile
    0.0s
=> => transferring dockerfile:
204B
    0.0s
=> [php-fpm internal] load metadata for docker.io/library/php:5.6-
fpm
1.6s
=> [php-fpm internal] load
.dockerignore
    0.0s
=> => transferring context:
```

2B

0.0s

=> [php-fpm 1/3] FROM docker.io/library/php:5.6-  
fpm@sha256:4f070f1b7b93cc5ab364839b79a5b26f38d5f89461f7bc0cd4bab4a3ad7d67d7

0.0s

=> CACHED [php-fpm 2/3] RUN docker-php-ext-install mysqli pdo  
pdo\_mysql

0.0s

=> CACHED [php-fpm 3/3] RUN echo 'php\_admin\_value[doc\_root] = /var/www/html/public' >>  
/usr/local/etc/php-fpm.d/www.conf

0.0s

=> [php-fpm] exporting to  
image

0.0s

=> => exporting  
layers

0.0s

=> => writing image  
sha256:552b673bf2989a93258340ee86ae7c31477c946f95aa73a63af49cede70235a6

0.0s

=> => naming to docker.io/library/staybarcelona-php-  
fpm  
0.0s

=> [apache-httpd internal] load build definition from  
Dockerfile  
0.0s

=> => transferring dockerfile:  
1.17kB

0.0s

=> [apache-httpd internal] load metadata for  
docker.io/library/httpd:2.4

0.9s

=> [apache-httpd internal] load  
.dockerignore

0.0s

=> => transferring context:  
2B

0.0s

=> [apache-httpd internal] load build  
context

```

0.0s
=> => transferring context:
72B

0.0s
=> [apache-httpd 1/4] FROM
docker.io/library/httpd:2.4@sha256:104f07de17ee186c8f37b9f561e04fbfe4cf080d78c6e5f3802fd08fd118c3da

0.0s
=> CACHED [apache-httpd 2/4] COPY ./apache/apache.vhost.conf
/usr/local/apache2/conf/extra/apache.vhost.conf

0.0s
=> CACHED [apache-httpd 3/4] RUN sed -i          -e '/#LoadModule rewrite_module/s/^#//g'          -e
'/#LoadModule expires_module/s/^#//g'          -e '/#LoadModule deflate_module/s/^#//g'          -e '/#LoadModule
proxy_module/s/^# 0.0s
=> CACHED [apache-httpd 4/4] RUN echo "Include /usr/local/apache2/conf/extra/apache.vhost.conf" >>
/usr/local/apache2/conf/httpd.conf                                0.0s
=> [apache-httpd] exporting to
image

0.0s
=> => exporting
layers


0.0s
=> => writing image
sha256:7428ace356414dd78d15c185ffa4137e7e3806018ce31eac61be778c773738f5

0.0s
=> => naming to docker.io/library/staybarcelona-apache-
httpd
0.0s
[+] Running 3/3
✓ Container staybarcelona-mysql-db-1
Started
0.0s
✓ Container staybarcelona-php-fpm-1
Started
0.6s
✓ Container staybarcelona-apache-httpd-1 Started


```

## Comprobaciones

En el navegador probaremos la información del `phpinfo()` en la ruta `localhost:8080`

Info  Image not found or type unknown

Ahora la conexión **mySQL**

Info  Image not found or type unknown

# Conclusión

Bien, esto es un esbozo de como podemos adecuarnos con Docker Composer, en un escenario como el descrito además de aprender ciertas metodologías con Docker.

Esto es muy básico y dependerá de la instalación y trabajos que tienes en tu máquina. Hay muchos caminos y formas de trabajar con Docker. Esta es una rápida para un escenario particular.

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

# Cosas de Docker

Ay Docker. Bendito docker, y sufrido como el solo.

# Instalar Postgresql con Postgis

## Introducción

Bueno, esta es la de siempre. Yo prefiero la eficacia de Elasticsearch, frente a el uso de un DBRM como Postgresql del que no se cansan de dar alabanzas, pero que no me convence para temas de este tipo. Sin embrago, hay un tema que es especial **GIS** y si uno no quiere morir con él, por Postgresql y PostGIS debe pasar.

Para no moriri en el inteneto, en desarrollo probe el proceso de instalar en mi MacBook M1. Tremendo. Una perdida de tiempo, para luego perder más tiempo en producción.

Asi que aquí, mis dos céntimos.

## Postgresql + Postgis Docker

Ya sabe que si quieres una versión específica (ojo a las versiones de Postgresql que no son como las de MySQL o MariaDB, ya que los cambiso mayores, son eso, **mayores**) puedes ir a su página en el [Hub de Docker](#) y forzar el comando a usar la que desees.

```
docker run --name my_postgis_container -e POSTGRES_PASSWORD=MY_PASSWORD -d -p 5432:5432
postgis/postgis
```

Esto ya sabes que puedes adapartlo a tu gusto. En mi caso, quiero que sea un postgresql accesible vía 127.0.0.1, asi que no necesito asociarlo a network como en otros proyectos.

[TablePlus - Configuración acceso a Postgresql - Docker](#)

[TablePlus - Vista templates\\_postgis](#)

## Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias

de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).