Líneas de comandos varios

Estilo por defecto del code blocks:

- Language: vim
- Theme: agate

Docker:

Ejecutar comandos de Symfony desde el contenedor de Docker:

```
docker-compose exec apache.local php bin/console
```

Cambiar contraseña back office Admin:

```
docker-compose exec --user root apache.local php bin/console
fos:user:change-password adminuser adminuser
```

**adminuser* Es es el nombre y contraseña del usuario por defecto que se utilizará

Instalar composer en Docker:

```
docker run --rm --interactive --tty --volume ./:/app --user www-data
composer
```

Instalar node en docker:

```
docker run --init -it --rm --user root -v $(pwd):/app -w /app node:9
yarn install
   -  o -
docker run --init -it --rm --user root -v $(pwd):/app -w /app node:9 npm
install
```

Ejecutar comandos Node:

```
docker run --init -it --rm --user root -v $(pwd):/app -w /app node:9 npm
run build
```

Encender el contenedor:

Abrir un terminal y dirigirse a la carpeta del proyecto y desde la raíz de la misma ejecutar el siguiente comando:

1º Compilamos por si tuviéramos nuevos cambios:

```
docker-compose build
```

2º Levantamos el contenedor:

```
docker-compose up -d
```

3º Apagar el contenedor y borra los contenedores

```
docker-compose down
```

3º Apagar el contenedor sin borrar los contenedores

```
docker-compose stop
```

Compilar js y scss

En Windows:

Para ello deberemos de acceder a la carpeta donde nuestro proyecto y con docker activado abrir una consola de WSL presionando Shift + Click Derecho y copiaremos el siguiente comando:

```
docker run --init -it --rm -v $(pwd):/app -w /app node:9 npm run dev
```

Nota: Es recomendable que la carpeta esté alojada en C:// o en D:// directamente sobre la raíz ya que la dirección no puede contener ninguna carpeta con letras en mayúscula ni compuestas.

Crear una migración a partir de la Entity:

```
docker-compose exec apache.local php bin/console
doctrine:migrations:diff
```

Lanzar una migración:

```
docker-compose exec apache.local php bin/console
doctrine:migrations:migrate
```

Renombrar archivos en Git

Hay veces que git se vuelve tonto cuando cambiamos el nombre de los archivos esto paso por que el esta constantemente visualizando si hay cambios es decir hace un seguimiento de los archivos entonces al modificarle el nombre hay veces que no lo acaba de reconocer y como la ruta es incorrecta te genera una increible pila de errores. Pues bien la solución a esto es la siguiente.

Para ello habrá que ir de archivo en archivo y en la consola ejecutar el siguiente comando:

```
git mv old_filename new_filena
```

Normas tácitas:

En PHP existen unos estándares que marcan como se debe de escribir el lenguaje en cuestión a esto se le llaman PSR y en concreto nosotros deberemos de programar con el PSR-12 para ello debemos de bajarnos la extensión para Visual studio Code en el caso en el que utilicemos este IDE: [phpcs - Visual Studio Marketplace]

Organización de las ramas en git:

En cada repositorio se creará una rama de *dev* y sobre esta no se podrá comitear directamente, dentro de esta se crearán diferentes ramas (los nombre siempre en Inglés ejemplo *features/fix-slide*).
El nombre de los commits junto con la descripción podrá estar en Español.
Para hacer un merge a la rama de dev se deberá hacer un Merge Request y la persona correspondiente analizará la rama y realizará el merge oportuno.

Los nombre de los commits deberán de expresar adecuadamente en el título el objetivo del commit ejemplo:

- Arreglo:
- Añade:
- Revierte:
- Eliminado:

Solo con el título el otro desarrollador ya debe saber de qué va ese commit.

Referenciar una traducción en Twig:

```
{{ 'web.home.adventures_done.header'|trans({}, 'messages')|upper }}
```

Crear archivo de traducciones:

En la carpeta de /translations crear nuestro archivo con el nombre por ejemplo de messages.es.yml y que cuya estructura será:

```yaml
web:
    home:
        adventures.done:
            header: Aventuras realizadas
            header_image: Viaje
        travel.community:
            alt_main_image: Antes de viajar
```

```
<div class="container-fluid">
    <!-- Slider main container -->
    <div class="swiper-normal-autoplay">
        <!-- Additional required wrapper -->
        <div class="swiper-wrapper">
            <!-- Slides -->
            {% for travel in expiredTravels %}
                <div class="swiper-slide">
                    {# height: 480px; width: 100%; #}
                    <img src="{% path travel.banner, 'normal' %}"
alt="slider item">
                </div>
            {% endfor %}
        </div>
    </div>
</div>
```