PHP 8 development environment with PHP-FPM, Nginx and MySQL, using Docker and Docker Compose

how to use it

- Use the make command to access al the commands and also to help you

Copy .env.example to .env

```
cp .env.example .env
```

or use the make command (you need to have installed on your system the make command)

```
make make-init
```

Build a Docker image for PHP with the content of the root folder

```
docker build -t php-project .docker
```

To rebuild the docker without cache use the following command

```
docker build --no-cache -t php-project .docker
```

Running the docker-compose

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

With make use:

```
make docker-up
```

To remove the orphan containers

With docker-compose

```
docker-compose down --remove-orphans
```

With make command:

```
make docker-down
```

To connect the docker containers with a non root user

the variables are on the `.env` file for the examples we would use:

```
DOCKER_NORMAL_USER_UUID=1000

NGINX_CONTAINER_NAME=nginx-webserver
DB_CONTAINER_NAME=db
APP_CONTAINER_NAME=app
```

with `docker-compose` command:

To the PHP container

```
docker-compose run --rm --service-ports -u $(DOCKER_NORMAL_USER_UUID)
$(APP_CONTAINER_NAME) /bin/bash
# this would be like calling this on the terminal
# docker-compose run --rm --service-ports -u 1000 app /bin/bash
```

To the BD container

```
docker-compose run --rm --service-ports -u $(DOCKER_NORMAL_USER_UUID)
$(DB_CONTAINER_NAME) /bin/bash
# this would be like calling this on the terminal
# docker-compose run --rm --service-ports -u 1000 db /bin/bash
```

To the NGINX container

```
docker-compose run --rm --service-ports -u $(DOCKER_NORMAL_USER_UUID)
$(NGINX_CONTAINER_NAME) /bin/bash
# this would be like calling this on the terminal
# docker-compose run --rm --service-ports -u 1000 nginx-webserver /bin/bash
```

To the COMPOSER container

```
docker-compose run --rm --service-ports -u $(DOCKER_NORMAL_USER_UUID)
$(COMPOSER_CONTAINER_NAME) /bin/bash
# this would be like calling this on the terminal
# docker-compose run --rm --service-ports -u 1000 composer /bin/bash
```

with make command: **To the PHP container**

```
make bash-php
```

To the BD container

```
make bash-db
```

To the NGINX container

```
make bash-nginx
```

To the COMPOSER container

```
make bash-composer
```

The make command

The make command is using the Makefile on the root directory and it's also getting the information from inside de .make folder were we have each script loaded

The PHP configuration

The PHP configuration is inside the .docker folder on the Dockerfile

The Nginx configuration

The Nginx configuration is inside the .nginx folder

The database

Whenever you run the dockers, the information inside the database will be store under de data folder this will allow our database to persist even when we stop the container.