

INSTALACIÓN PROYECTO REACT-SYMFONY + BD en AWS

GitHub: <https://github.com/anaGG07/Despliegue.git>

Partiendo de la tarea anterior, PHP, composer, git y la configuración del entorno ya está hecho.

1. Configurar Frontend

1. Crear poryecto con vite

```
Ana@ANIR0G MINGW64 ~/Desktop/Curso/.Despliegue/UD 3/react-symfony (main)
• $ cd 02.Despliegue-CompletoAMG/

Ana@ANIR0G MINGW64 ~/Desktop/Curso/.Despliegue/UD 3/react-symfony/02.Despliegue-CompletoAMG (main)
• $ mkdir backend frontend database web

Ana@ANIR0G MINGW64 ~/Desktop/Curso/.Despliegue/UD 3/react-symfony/02.Despliegue-CompletoAMG (main)
• $ cd frontend/

Ana@ANIR0G MINGW64 ~/Desktop/Curso/.Despliegue/UD 3/react-symfony/02.Despliegue-CompletoAMG/frontend (main)
• $ npm create vite@latest .

> npx
> create-vite .

✓ Select a framework: » React
✓ Select a variant: » JavaScript + SWC
```

2. Variable de entorno

```
1 CHOKIDAR_USEPOLLING=true
2
```

3. Configuración .env

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react-swc'

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [react()],
  server: {
    host: "0.0.0.0",
    allowedHosts: ['frontend'],
  }
})
```

4. Configuración App.jsx

```
import { useEffect, useState } from 'react';

function App() {
  const [message, setMessage] = useState('');

  useEffect(() => {
    fetch('/api/amg')
      .then((res) => res.json())
      .then((data) => setMessage(data.message));
  }, []);

  return (
    <div>
      <h1>Frontend en React de Ana Maria Garcia Garcia</h1>
      <p>Esta aplicación se conecta al backend de Symfony pidiéndole una respuesta</p>
      <p>Respuesta del Backend: {message} || 'Esperando respuesta ...'</p>
    </div>
  );
}

export default App;
```

5. Crear DockerfileFrontendAMG

```
FROM node:23-alpine

WORKDIR /app

COPY package*.json ./
RUN npm install

COPY . .

EXPOSE 5173

CMD ["npm", "run", "dev", "--", "--host", "0.0.0.0"]
```

2. Configurar Backend

1. Crear proyecto con Symfony

```
Ana@ANIR06 MINGW64 ~/Desktop/Curso/.Despliegue/UD 3/react-symfony/02.Despliegue-CompletoAMG/backend (main)
$ symfony new --webapp backend --no-git
* Creating a new Symfony project with Composer
```

2. Crear controlador AMG_Controller.php

```
namespace App\Controller;

use Doctrine\DBAL\Connection;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\Routing\Annotation\Route;

You, hace 3 horas | 1 author (You)
class AMG_Controller extends AbstractController
{
    private Connection $connection;

    public function __construct(Connection $connection)
    {
        $this->connection = $connection;
    }

    Tabnine | Edit | Test | Explain | Document
    #[Route('/api/amg', name: 'get_amg')]
    public function index(): JsonResponse
    {
        $sql = 'SELECT fraseAMG FROM secretosAMG LIMIT 1';
        $result = $this->connection->fetchOne($sql);

        return $this->json(['message' => $result ? 'No hay mensajes en la BD']);
    }
}
```

3. Añadir la conexión a bd en .env

```
DATABASE_URL="mysql://alumnoDAW:passAMG@db:3306/AMG_BD"
```

4. Crear la entidad TablaAMG

```
namespace App\Entity;

use Doctrine\ORM\Mapping as ORM;

You, hace 3 horas | 1 author (You)
#[ORM\Entity]
#[ORM\Table(name: "secretosAMG")]
class TablaAMG
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column(type: "integer")]
    private int $id;

    #[ORM\Column(name: "frase_amg", type: "string", length: 255)]
    private string $fraseAmg;

    Tabnine | Edit | Test | Explain | Document
    public function getId(): int
    {
        return $this->id;
    }

    Tabnine | Edit | Test | Explain | Document
    public function getFraseAmg(): string
    {
        return $this->fraseAmg;
    }

    Tabnine | Edit | Test | Explain | Document
    public function setFraseAmg(string $fraseAmg): self
    {
        $this->fraseAmg = $fraseAmg;
        return $this;
    }
}
```

5. Crear DockerfileBackendAMG

```
FROM php:8.2-apache

RUN apt-get update && apt-get install -y \
    git unzip libicu-dev \
    && docker-php-ext-install pdo pdo_mysql

COPY --from=composer:2 /usr/bin/composer /usr/bin/composer

WORKDIR /app

COPY . .

RUN composer install
RUN php bin/console doctrine:migrations:migrate --no-interaction

EXPOSE 8000

CMD ["php", "-S", "0.0.0.0:8000", "-t", "public"]
```

6. Configuración base de datos database/init.sql

```
CREATE TABLE secretosAMG (
    id INT AUTO_INCREMENT PRIMARY KEY,
    fraseAMG VARCHAR(255) NOT NULL
);

INSERT INTO secretosAMG (fraseAMG) VALUES ('La base de datos responde, buenos días Ana María García García');
```

7. Configuración docker-compose en raíz

```
services:
  frontend:
    build:
      context: ./frontend
      dockerfile: DockerfileFrontendAMG
    working_dir: /app
    volumes:
      - ./frontend/app
      - ./app/node_modules
    ports:
      - '3000:3000'
    command: sh -c "npm install && npm start"
    environment:
      - CHOKIDAR_USEPOLLING=true
    networks:
      - redAMG

  backend:
    build:
      context: ./backend
      dockerfile: DockerfileBackendAMG
    ports:
      - '8000:8000'
    volumes:
      - ./backend/app
    environment:
      - APP_ENV=dev
    depends_on:
      - db
    networks:
      - redAMG

  web:
    build:
      context: ./web
      dockerfile: DockerfileWebAMG
    volumes:
      - ./web/default.conf:/etc/nginx/conf.d/default.conf
    ports:
      - '81:80'
    depends_on:
      - backend
    networks:
      - redAMG

  db:
    image: mysql:5.7
    volumes:
      - ./database/init.sql:/docker-entrypoint-initdb.d/init.sql
      - db_data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: AMG_BD
      MYSQL_USER: alumnoDAW
      MYSQL_PASSWORD: passAMG
    ports:
      - '3306:3306'
    networks:
      - redAMG

  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    ports:
      - '8080:80'
    environment:
      PMA_HOST: db
      PMA_USER: alumnoDAW
      PMA_PASSWORD: passAMG
    depends_on:
      - db
    networks:
      - redAMG

volumes:
  db_data:

networks:
  redAMG:
    driver: bridge
```

3. Subida a AWS y Levantar los contenedores

1. Levantar todos los servicios con Docker Compose

- > docker-compose up -d
- > docker-compose up --build

```
ubuntu@ip-172-31-86-69:~/Despliegue/02.Despliegue-CompletoAMG$ docker-compose up --build
[+] Building 26.2s (30/30) FINISHED
=> [backend internal] load build definition from DockerfileBackendAMG
=> => transferring dockerfile: 342B
=> [frontend internal] load build definition from DockerfileFrontendAMG
=> => transferring dockerfile: 200B
=> [frontend internal] load metadata for docker.io/library/node:23-alpine
=> [backend internal] load metadata for docker.io/library/composer:2
=> [backend internal] load metadata for docker.io/library/php:8.2-apache
=> [frontend internal] load .dockerignore
=> => transferring context: 2B
=> [backend internal] load .dockerignore
=> => transferring context: 2B
=> [frontend 1/5] FROM docker.io/library/node:23-alpine@sha256:70eca392e3d52cb7d133b52d52e8600d8c410a5eaae6105c11324b28868f9ac9
=> [frontend internal] load build context
=> => transferring context: 137.06kB
=> [backend stage-0 1/6] FROM docker.io/library/php:8.2-apache@sha256:e7582ae29b3cc403ce855cf65f5a4b7503ddc599d9a9060b51df7fb5afa2569f
=> [backend internal] load build context
=> => transferring context: 58.06MB
=> [backend] FROM docker.io/library/composer:2@sha256:e0c9ac329256c25b0dee572df37d586570fb26bb6baa7d0abe69b84181701e1
=> CACHED [frontend 2/5] WORKDIR /app
=> CACHED [frontend 3/5] COPY package*.json ./
=> DONE [frontend 4/5] RUN npm install
```

2. Verificar contenedores

- > docker ps

```
ubuntu@ip-172-31-86-69:~/Despliegue$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
66f318e7bdb7   despliegue-frontend                "docker-entrypoint.s..." 6 minutes ago  Up 6 minutes
85c2e1237251   despliegue-nginx                   "/docker-entrypoint..." 6 minutes ago  Up About a minute
efbf8217afb8   despliegue-backend                 "docker-php-entrypoi..." 6 minutes ago  Up 6 minutes
7ec14cf24434   anagg07/examen_apache_amg:diurno   "/bin/bash"              10 days ago   Exited (137) 10 days ago
ubuntu@ip-172-31-86-69:~/Despliegue$
```

4. Pruebas

