**Universidade de Aveiro**

Engenharia Informática

**Disciplina: Introdução à Engenharia Informática**

# **Relatório de Projeto:**

## **Site de Anúncios da Turma com WordPress e Docker Compose**

**Elaborado por:** Ângelo Vilar, João Costa e Miguel Silva

O presente relatório documenta o desenvolvimento de um "Site de Anúncios da Turma", um projeto prático que visa a implementação de uma solução web robusta utilizando contentores. O objetivo principal foi orquestrar uma instalação completa do WordPress, garantindo a persistência de dados e a segurança das configurações.

Para tal, foi utilizada a ferramenta Docker Compose, permitindo a definição de uma arquitetura multi-serviço que integra a aplicação web (WordPress) e o sistema de gestão de base de dados (MariaDB) num ambiente isolado e replicável.

### **Tecnologias Utilizadas**

* **Docker & Docker Compose:** Para orquestração dos contentores e definição da infraestrutura como código (IaC).
* **WordPress:** CMS utilizado para criar o site de anúncios. A imagem utilizada inclui o servidor web Apache e o interpretador PHP necessários.
* **MariaDB (v11):** Sistema de gestão de base de dados relacional escolhido para armazenar o conteúdo do WordPress, substituindo o MySQL pela sua compatibilidade e performance.
* **Volumes:** Mecanismo utilizado para garantir a persistência de dados entre reinicializações (tanto para a base de dados como para os ficheiros do site).
* **Redes de Contentores:** Configuração de uma rede “bridge” personalizada para comunicação interna segura entre os serviços.

## 

**Montagem do Docker Compose**

O ficheiro central do projeto define os serviços “db” e “wordpress”. Destaca-se a utilização da rede “anuncios-net” e o mapeamento de portas e volumes.

### **Segurança e Variáveis de Ambiente**

### Para evitar a exposição de credenciais no código-fonte, todas as passwords e configurações sensíveis foram abstraídas para um ficheiro “.env”. No repositório, é disponibilizado apenas um modelo (.env.example):

### **“.gitignore”**

Foi configurado este ficheiro para impedir que o ficheiro real de credenciais e os dados pesados dos volumes fossem enviados para o repositório Git.

**“README”**

Para garantir a reprodutibilidade do projeto em qualquer máquina (seguindo os requisitos de WSL2/Linux e Docker Desktop), foram definidos os seguintes passos de instalação, documentados no README.ml do projeto.

**Realização do próprio site:**

* **Tema e Identidade:** O tema foi personalizado pelo grupo para enquadrar mais dentro do objetivo do trabalho, este sendo ter as notícias mais recentes da turma na página principal.
* **Conteúdo:** Colocamos a logo da UA como símbolo do site, adicionamos uma fotografia do DeTI na página principal e botões de “Saber+” para caso alguém queira ver uma das notícias em específico.

**Tentativa falhada de tentar mandar ao professor a habilidade de criar uma cópia exata do site:**

Depois da criação do site queríamos tentar criar uma maneira de o professor poder criar uma cópia exata do site com poucos cliques, nós tentámos fazer isto de duas maneiras:

* Através do import (.xml) do próprio WordPress do nosso site
* Através de um plugin separado intitulado “All-in-One WP Migration”

Os métodos falharam por motivos diferentes, a exportação do .xml não funcionou completamente porque era impossível colocar as imagens do website no próprio GitHub porque assim as imagens estavam no Linux(Ubuntu) enquanto a site as tentava abrir no Windows, o que é impossível.

O plugin não funcionava porque o website utilizava mais que o limite grátis que o plugin permitia importar: 2mb. Ao exportar não existia problema nenhum, é só ao importar que o plugin bloqueia por não permitir (na versão grátis) ficheiros acima de 2mb. Acabamos por simplesmente usar o .xml e colocamos na pasta “export” para o professor poder ver a estrutura do site temas, mas as imagens não aparecem.

**(abaixo estão disponibilizados os códigos utilizados e criados pelo grupo)**

**docker\_compose.ym**

services:

db:

image: mariadb:11

container\_name: anuncios\_db

restart: always

env\_file: .env

environment:

MYSQL\_ROOT\_PASSWORD: ${DB\_ROOT\_PASSWORD}

MYSQL\_DATABASE: ${DB\_NAME}

MYSQL\_USER: ${DB\_USER}

MYSQL\_PASSWORD: ${DB\_PASSWORD}

TZ: ${TZ}

volumes:

- db-data:/var/lib/mysql

networks:

- anuncios-net

wordpress:

image: wordpress:php8.2-apache

container\_name: anuncios\_wp

restart: always

depends\_on:

- db

env\_file: .env

environment:

WORDPRESS\_DB\_HOST: db:3306

WORDPRESS\_DB\_USER: ${DB\_USER}

WORDPRESS\_DB\_PASSWORD: ${DB\_PASSWORD}

WORDPRESS\_DB\_NAME: ${DB\_NAME}

WORDPRESS\_TABLE\_PREFIX: ${WP\_TABLE\_PREFIX}

TZ: ${TZ}

ports:

- "${WP\_PORT}:80"

volumes:

- wp-data:/var/www/html

networks:

- anuncios-net

volumes:

db-data:

wp-data:

networks:

anuncios-net:

driver: bridge

.**gitignore**

# Pasta onde fica guardada a base de dados

db-data/

# Pasta onde ficam guardados os ficheiros do WordPress

wp-data/

# Ficheiro que contém passwords e outras credenciais reais

.env

# Ficheiros e lixo comuns que não interessam

.DS\_Store

\*.log

**.env.example**

# Variaveis da MariaDB, Mudar o conteúdo se desejado

DB\_ROOT\_PASSWORD=R00T\_P@SS

DB\_NAME=wordpress

DB\_USER=wpuser

DB\_PASSWORD=WPUSER\_P@SS

# Variaveis do WordPress, Mudar o conteúdo se desejado

WP\_TABLE\_PREFIX=wp\_

WP\_PORT=8080

TZ=Europe/Lisbon

**Screenshots do site com as imagens:**

