Projects 01

Introdução Engenharia Informática

Mário Antunes

October 27, 2025

Projects

Make groups of two or three students and select one of the following projects. All projects will be hosted in GitHub, using GitHub Classroom. In the repository there should be all the relevant scripts and configuration files as a `README.md` with instruction on how to deploy the project.

1. High-Performance Static Site with Varnish Cache

- **Description:** Deploy a high-performance web service using Docker Compose. This setup must include two services: an `nginx` container and a `varnish` cache container. The static website content (a complex page with several styles and images) must be served from a **volume** mounted to the `nginx` container. Varnish must be configured to sit in front of NGINX, and only Varnish's port should be exposed.
- **Core Topics:** Docker Compose (multi-service), NGINX, Varnish, `volumes`, container networking.

2. The "It Works on My Machine" Solver: A Dev Container

- **Description:** Create a `Dockerfile` for a specific programming language (e.g., Python, C++, or Node.js). This `Dockerfile` should install the compiler/interpreter and all necessary libraries. The project will use Docker Compose and a **volume** to mount a local code folder, allowing you to compile/run your code from *inside* the container, ensuring a reproducible build environment.
- **Core Topics:** `Dockerfile`, `volumes`, Docker Compose, package management (`apt`).

3. Automated Backup to Nextcloud

- **Description:** Write a **Bash script** that creates a compressed `.tar.gz` backup of a specified directory. The script should then move this archive into a local folder that is being monitored by the **Nextcloud Desktop Client**. The goal is to create a fully automated backup system where local files are archived and then automatically synced to a remote Nextcloud server.
- **Core Topics:** Bash scripting (`tar`, `date`), `cron`, Nextcloud client.

4. Class Announcements Site with WordPress

- **Description:** Deploy a full WordPress installation using Docker Compose. This requires orchestrating `wordpress` and `mysql` (or MariaDB) containers. You must use **volumes** for persistence. The goal is to configure the site as a simple announcement feed for this class, creating at least two posts and customizing the theme.
- **Core Topics:** Docker Compose (multi-service), WordPress, container networking, `volumes`, environment variables.

5. Performance Showdown: VM vs. Container

- **Description:** Deploy a simple NGINX web server in two ways: 1) inside a full **Debian VM** (using VirtualBox/QEMU) and 2) inside a **Docker container**. You will then write a report comparing the startup time, idle RAM usage, and disk space footprint for both methods.
- **Core Topics:** Virtualization (VM setup), Containers (Docker), system monitoring tools (`top`, `df`, `time`).

6. Class Wiki Deployment

- **Description:** Use Docker Compose to deploy a fully functional wiki (e.g., `dokuwiki/dokuwiki` or `linuxserver/bookstack`) to serve as a knowledge base for this class. The focus is on correctly reading the image's documentation, managing persistent data with **volumes**, and configuring the service using environment variables. You must populate the wiki with at least five pages of content from the class materials.
- **Core Topics:** Docker Compose, `volumes`, managing 3rd-party images, environment variables.