

# Projects 01

## Introdução Engenharia Informática

Mário Antunes

October 27, 2025

## Projects

Form groups of two or three students (exceptionally, projects can be done individually) and select **one** of the following projects. All projects will be hosted on **GitHub**, using [GitHub Classroom](#). Check [here](#) for details.

The repository must contain all relevant scripts, configuration files, and a `README.md` with instructions on how to deploy the project. It should also contain a project report in PDF format.

This is a three-week project (deadline 21/11/2025). You have until the end of this week to notify your professor (via e-mail) of your group members and chosen topic (the list of topics can be found [here](#)).

Do not forget to contact your professor with any questions. Further instructions may be added.

## Topics

### 1. High-Performance Static Site with Caching

- **Description:** Deploy a high-performance web service using Docker Compose. This setup must include two services: a web server (like **Caddy** or **Apache httpd**) and a reverse proxy cache (like **Squid**). The static website content (a complex page with several styles and images) must be served from a **volume** mounted to the web server container. The cache must be configured to sit in front of the web server, and only the cache's port should be exposed.
- **Core Topics:** Docker Compose (multi-service), Caddy/httpd, Squid, 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.

## Github Classroom Access

Here are detailed instructions to access GitHub Classroom.

### 1. Join the Assignment and Form Your Team

1. **Access the link:** Go to [here](#)
  2. **Find your name:** Select your name from the student list. > **Can't find your name?** All names registered on PACO were added. If yours is missing, please contact [Prof. Mário Antunes](#).
  3. **Create a team (ONE member only):** Only **one** person from your group should create a team. Follow this exact naming structure (the nmec should be stored): [nmec1]\_[nmec2]\_[nmec3]\_tema0[1-6]
    - (Example: 132745\_133052\_tema02)
  4. **Join the team (All other members):** The remaining project members must find and join the team created in the previous step.
- 

### 2. Access the Organization and Repository

1. **Accept the email invite:** After joining a team, all members will receive an email invitation to join the detiuaaveiro GitHub organization.
  2. **You must accept this invitation** before you can continue.
  3. **Refresh the page:** Go back to the GitHub Classroom page and refresh it.
  4. **Verify access:** You should now see and have access to your team's working repository.
- 

### 3. Configure an SSH Key for Access

This will allow you to clone and push to the repository from your command line without entering your password every time.

1. **Check for an existing SSH key:** Open your terminal and run this command:
 

```
cat ~/.ssh/id_ed25519.pub
```
2. **Generate a key (if needed):**
  - If you see a key (starting with ssh-ed25519 ... ), copy the entire line and skip to step 3.
  - If you see an error like "No such file or directory," run the following command to create a new key:
 

```
ssh-keygen -q -t ed25519 -N ''
```

- After it's generated, run `cat ~/.ssh/id_ed25519.pub` again to view your new key and copy it.

### 3. Add the key to your GitHub account:

- Go to your GitHub **Settings**.
- On the left menu, click **SSH and GPG keys**.
- Click the **New SSH key** button.
- Give it a **Title** (e.g., "My UA Laptop").
- Paste the key you copied into the **Key** field.
- Make sure the "Key type" is set to **Authentication Key**.
- Click **Add SSH key**.

### 4. Authorize the key for SSO:

- After adding the key, find it in your list on the same page.
- Click **Configure SSO**.
- Select the **detiuaveiro** organization, fill in your login details, and grant access.