Projetos 01

Tópicos de Informática para Automação

Mário Antunes

27 de Outubro de 2025

Projetos

Formem grupos de dois ou três estudantes (excecionalmente, os projetos podem ser feitos individualmente) e selecionem **um** dos seguintes projetos. Todos os projetos serão alojados no **GitHub**, usando o GitHub Classroom. Consultem aqui os detalhes.

O repositório deve conter todos os scripts relevantes, ficheiros de configuração, e um `README.md` com instruções sobre como implementar o projeto. Deve também conter um relatório do projeto em formato `PDF`.

Este é um projeto de três semanas (data limite 21/11/2025). Têm até ao final desta semana para notificar o vosso professor (via e-mail) sobre os membros do vosso grupo e o tópico escolhido (a lista de tópicos pode ser encontrada aqui).

Não se esqueçam de contactar o vosso professor com quaisquer questões. Instruções adicionais poderão ser adicionadas.

Tópicos

1. Site Estático de Alta Performance com Caching

- **Descrição:** Implementar um serviço web de alta performance usando Docker Compose. Esta configuração deve incluir dois serviços: um servidor web (como **Caddy** ou **Apache** `httpd`) e uma cache de reverse proxy (como **Squid**). O conteúdo do site estático (uma página complexa com vários estilos e imagens) deve ser servido a partir de um **volume** montado no contentor do servidor web. A cache deve ser configurada para ficar à frente do servidor web, e apenas a porta da cache deve ser exposta.
- **Tópicos Principais:** Docker Compose (multi-serviço), Caddy/httpd, Squid, `volumes`, redes de contentores (container networking).

2. O Solucionador "Funciona na Minha Máquina": Um Dev Container

- **Descrição:** Criar um `Dockerfile` para uma linguagem de programação específica (ex: Python, C++, ou Node.js). Este `Dockerfile` deve instalar o compilador/interpretador e todas as bibliotecas necessárias. O projeto usará Docker Compose e um **volume** para montar uma pasta de código local, permitindo-vos compilar/executar o vosso código de *dentro* do contentor, assegurando um ambiente de compilação reprodutível.
- **Tópicos Principais:** `Dockerfile`, `volumes`, Docker Compose, gestão de pacotes (`apt`).

3. Backup Automatizado para o Nextcloud

- **Descrição:** Escrever um **script Bash** que cria um backup comprimido `.tar.gz` de uma pasta especificada. O script deve depois mover este arquivo para uma pasta local que está a ser monitorizada pelo **Cliente Desktop Nextcloud**. O objetivo é criar um sistema de backup totalmente automatizado onde os ficheiros locais são arquivados e depois automaticamente sincronizados para um servidor Nextcloud remoto.
- **Tópicos Principais:** Scripting Bash (`tar`, `date`), `cron`, cliente Nextcloud.

4. Site de Anúncios da Turma com WordPress

- **Descrição:** Implementar uma instalação completa do WordPress usando Docker Compose. Isto requer a orquestração de contentores `wordpress` e `mysql` (ou MariaDB). Devem usar **volumes** para a persistência. O objetivo é configurar o site como um feed de anúncios simples para esta turma, criando pelo menos duas publicações e personalizando o tema.
- **Tópicos Principais:** Docker Compose (multi-serviço), WordPress, redes de contentores, `volumes`, variáveis de ambiente.

5. Confronto de Performance: VM vs. Contentor

- **Descrição:** Implementar um servidor web NGINX simples de duas formas: 1) dentro de uma **VM Debian** completa (usando VirtualBox/QEMU) e 2) dentro de um **contentor Docker**. Irão depois escrever um relatório a comparar o tempo de arranque, uso de RAM em inatividade, e o uso de espaço em disco para ambos os métodos.
- **Tópicos Principais:** Virtualização (configuração de VM), Contentores (Docker), ferramentas de monitorização de sistema (`top`, `df`, `time`).

6. Implementação de um Wiki da Turma

- **Descrição:** Usar Docker Compose para implementar um wiki totalmente funcional (ex: `dokuwiki/dokuwiki` ou `linuxserver/bookstack`) para servir como uma base de conhecimento para esta turma. O foco está em ler corretamente a documentação da imagem, gerir dados persistentes com **volumes**, e configurar o serviço usando variáveis de ambiente. Devem popular o wiki com pelo menos cinco páginas de conteúdo dos materiais da aula.
- **Tópicos Principais:** Docker Compose, `volumes`, gestão de imagens de terceiros, variáveis de ambiente.

Acesso ao Github Classroom

Aqui estão instruções detalhadas para aceder ao GitHub Classroom.

1. Entrar na Tarefa (Assignment) e Formar a Vossa Equipa

1. **Aceder ao link:** Ir para aqui
2. **Encontrar o vosso nome:** Selecionem o vosso nome da lista de estudantes. > **Não encontram o vosso nome?** Todos os nomes registados no PACO foram adicionados. Se o vosso estiver em falta, por favor contactem o **Prof. Mário Antunes**.
3. **Criar uma equipa (APENAS um membro):** Apenas **uma** pessoa do vosso grupo deve criar a equipa. Sigam esta estrutura exata de nome (os nmec devem estar ordenados): `[nmec1]_[nmec2]_[nmec3]_tema0[1-6]`
   - *(Exemplo:* `132745_133052_tema02`*)*
4. **Juntar-se à equipa (Todos os outros membros):** Os restantes membros do projeto devem encontrar e juntar-se à equipa criada no passo anterior.

---

2. Aceder à Organização e ao Repositório

1. **Aceitar o convite por e-mail:** Depois de se juntarem a uma equipa, todos os membros irão receber um convite por e-mail para se juntarem à organização `detiuaveiro` no GitHub.
2. **Devem aceitar este convite** antes de poderem continuar.
3. **Atualizar a página:** Voltem à página do GitHub Classroom e atualizem-na.
4. **Verificar o acesso:** Devem agora ver e ter acesso ao repositório de trabalho da vossa equipa.

---

3. Configurar uma Chave SSH para Acesso

Isto irá permitir-vos fazer 'clone' e 'push' para o repositório a partir da vossa linha de comandos sem introduzir a vossa password sempre.

1. **Verificar se existe uma chave SSH:** Abram o vosso terminal e executem este comando:

   ```
   cat ~/.ssh/id_ed25519.pub
   ```

2. **Gerar uma chave (se necessário):**
   - Se virem uma chave (a começar com `ssh-ed25519 ...`), copiem a linha inteira e saltem para o passo 3.
   - Se virem um erro como "No such file or directory," executem o seguinte comando para criar uma nova chave:

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

   - Depois de ser gerada, executem `cat ~/.ssh/id_ed25519.pub` novamente para ver a vossa nova chave e copiem-na.
3. **Adicionar a chave à vossa conta GitHub:**
   - Vão às vossas **Settings** (Definições) do GitHub.
   - No menu à esquerda, cliquem em **SSH and GPG keys**.
   - Cliquem no botão **New SSH key**.
   - Dêem-lhe um **Title** (Título) (ex: "O meu portátil da UA").
   - Colem a chave que copiaram no campo **Key** (Chave).
   - Certifiquem-se que o "Key type" (Tipo de chave) está definido como **Authentication Key**.
   - Cliquem em **Add SSH key**.
4. **Autorizar a chave para SSO:**
   - Depois de adicionar a chave, encontrem-na na vossa lista na mesma página.
   - Cliquem em **Configure SSO**.
   - Selecionem a organização **detiuaveiro**, preencham os vossos detalhes de login, e concedam acesso.