

Projeto 02

Tópicos de Informática para Automação

Mário Antunes

24 de Novembro de 2025

Projetos

Formem grupos de dois ou três alunos (excepcionalmente, os projetos podem ser feitos individualmente) e selecionem **um** dos seguintes projetos. Todos os projetos serão alojados no **GitHub**, utilizando o **GitHub Classroom**. Verifiquem [aqui](#) para mais detalhes.

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

Este é um projeto de três semanas (prazo 22/12/2025). 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 dúvidas. Instruções adicionais podem ser acrescentadas.

Tópicos

1. A Fábrica “Markdown to PDF”

- **Descrição:** Criem um serviço *Dockerized* que converte ficheiros Markdown em PDFs profissionais. Devem criar um `Dockerfile` que instala o **Pandoc** e uma distribuição LaTeX mínima (ex: `texlive-xetex`). O contentor deve executar um **Bash script** que monitoriza um **volume** de entrada específico. Quando um ficheiro `.md` é detetado nesse volume, o script deve convertê-lo automaticamente para `.pdf` e colocar o resultado numa pasta de saída.
- **Tópicos Principais:** Bash Scripting (loops, monitorização de ficheiros), Docker Volumes, Compilação de Documentos (Markdown & LaTeX).

2. Visualizador de Latência de Rede

- **Descrição:** Desenvolvam uma ferramenta para analisar a estabilidade da rede usando uma *pipeline* em contentores.
 1. Criem um **Bash script** que faz “ping” a um alvo (ex: `google.com` ou `ua.pt`) periodicamente e regista o *timestamp* e a latência (ms) num ficheiro **CSV**.
 2. Criem um **Python script** que lê este CSV usando **Pandas** ou **Polars** e gera um gráfico de linhas mostrando a latência ao longo do tempo usando **Matplotlib** ou **Seaborn**.
 3. Todo o processo deve correr dentro de um contentor, guardando o *plot* final num volume.
- **Tópicos Principais:** Networking (ICMP/Ping), Manipulação de Dados (CSV), Visualização de Dados, Docker.

3. Dashboard de Geo-Dados (Tráfego ou Meteorologia)

- **Descrição:** Construam um *web dashboard* que visualiza dados geográficos. Devem criar um script **Python** que usa uma API para obter dados meteorológicos ou de tráfego, ou (usando **Pandas** ou **Polars**) que processa um *dataset* (ex: um CSV de estações meteorológicas ou incidentes de tráfego com coordenadas Lat/Lon) e exporta-o para JSON. Depois, façam o *deploy* de um contentor **Web**.

Server (Nginx ou Apache) alojando uma página HTML. Esta página deve usar a biblioteca JavaScript **Leaflet** para ler esses dados JSON e mostrar marcadores num mapa interativo.

- **Tópicos Principais:** Web Programming (HTML/JS/Leaflet), Formatação de Dados (CSV para JSON), Docker, Web Servers.

4. O Serviço de Upload e Plot de CSV

- **Descrição:** Criem um serviço *web* de análise de dados composto por dois contentores Docker.
 1. **Backend (Python/FastAPI):** Criem uma API que aceita o *upload* de um ficheiro CSV via um pedido POST. O *backend* deve usar **Pandas** para fazer o *parse* do ficheiro carregado, identificar colunas numéricas e retornar uma lista JSON de *links* de visualização disponíveis (ex: <http://localhost:8000/plot/temperature>). Quando um *link* é visitado, o *backend* gera e retorna uma imagem **Matplotlib/Seaborn** (PNG).
 2. **Frontend (Nginx):** Criem uma página *web* com um formulário HTML para fazer o *upload* do ficheiro. Usando JavaScript (*fetch*), enviem o ficheiro para o *backend*. Quando o *backend* responder, gerem dinamicamente uma lista de *links* clicáveis na página. Clicar num *link* deve abrir/mostrar o *plot* gerado.
- **Tópicos Principais:** API File Handling (Uploads), Análise de Dados em Python, Manipulação Dinâmica do DOM, Docker Networking.

5. Portfólio Dinâmico Full-Stack

- **Descrição:** Construam um site de portfólio pessoal que separa o conteúdo da apresentação, simulando uma arquitetura CMS do mundo real.
 1. **Backend (Python/FastAPI):** Criem uma API simples que serve os vossos dados de perfil como JSON. Deve ter um *endpoint* (ex: `/api/profile`) que retorna um dicionário contendo o vosso Nome, Bio e uma lista de Skills/Projetos.
 2. **Frontend (Nginx):** Façam o *deploy* de uma *shell* HTML que está inicialmente vazia de conteúdo. Usem **JavaScript** para fazer *fetch* dos dados do vosso *backend* no carregamento da página e popular os elementos do DOM (lista de Projetos, texto da Bio).
 3. **Interatividade:** O *frontend* deve ainda incluir o *toggle* "Dark/Light" (guardado no `localStorage`) e uma validação *Client-side* para um formulário "Contact Me".
- **Tópicos Principais:** Asynchronous JavaScript (*fetch/await*), Troca de Dados JSON, Separation of Concerns, Docker Composition.

6. Relatório de Recursos do Servidor

- **Descrição:** Simulem uma tarefa de administrador de sistemas. Criem um script que gera um ficheiro CSV de "server log" (Colunas: Timestamp, CPU_Usage, RAM_Usage). Depois, usem **Pandas** ou **Polars** para analisar este *log* e gerar um relatório de aviso: identifiquem as linhas onde o uso excede 90%. Finalmente, usem **Seaborn** ou **Matplotlib** para gerar um gráfico das tendências de uso de recursos e guardem-no em disco. O relatório do projeto deve ser compilado a partir de Markdown, incorporando este gráfico gerado.
- **Tópicos Principais:** Análise de Dados, System Concepts, Integração Markdown, Python.

Acesso ao Github Classroom

Aqui estão instruções detalhadas para aceder ao GitHub Classroom. A maioria dos alunos pode saltar vários passos, dado que estes foram completados no projeto 01.

1. Juntar-se ao Assignment e Formar a Equipa

1. **Aceder ao link:** Vão [aqui](#)
2. **Encontrar o vosso nome:** Selecione 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 uma equipa. Sigam esta estrutura de nomenclatura exata (o nmec deve estar ordenado):
[nmec1]_[nmec2]_[nmec3]_project02
 - (Exemplo: 132745_133052_project02)

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 Repositório

1. **Aceitar o convite por e-mail:** Após juntarem-se a uma equipa, todos os membros receberão um convite por e-mail para se juntarem à organização GitHub **detiuaveiro**.
 2. **Devem aceitar este convite** antes de poderem continuar.
 3. **Atualizar a página:** Voltem à página do GitHub Classroom e atualizem-na (refresh).
 4. **Verificar acesso:** Devem agora ver e ter acesso ao repositório de trabalho da vossa equipa.
-

3. Configurar uma Chave SSH para Acesso

Isto permitir-vos-á fazer *clone* e *push* para o repositório a partir da vossa linha de comandos sem introduzir a vossa *password* todas as vezes.

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 (começando por 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 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 aos vossos **Settings** do GitHub.
- No menu da esquerda, cliquem em **SSH and GPG keys**.
- Cliquem no botão **New SSH key**.
- Deem-lhe um **Title** (ex: "My UA Laptop").
- Colem a chave que copiaram no campo **Key**.
- Certifiquem-se de que o "Key type" está definido como **Authentication Key**.
- Cliquem em **Add SSH key**.

4. **Autorizar a chave para SSO:**

- Após adicionar a chave, encontrem-na na vossa lista na mesma página.
- Cliquem em **Configure SSO**.
- Selecione a organização **detiuaveiro**, preenchem os vossos dados de login e concedam acesso.