

Projeto Final

Introdução à Engenharia Informática

Mário Antunes

12 de Dezembro de 2025

Projetos

Este projeto é estritamente individual. Selecione **um** dos seguintes projetos. Todos os projetos serão alojados no **GitHub**, utilizando o [GitHub Classroom](#). Verifique [aqui](#) para 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. Deve também conter um relatório do projeto em formato PDF.

Este é o **projeto final**, desenhado para integrar as várias competências adquiridas ao longo do semestre (Shell Scripting, Docker, Python, Data Analysis e Web Technologies).

Este é um projeto de três semanas (prazo 14 de Janeiro de 2025). Têm até ao final desta semana para notificar o vosso professor (via e-mail) sobre o tópico escolhido (a lista de tópicos pode ser encontrada [aqui](#)).

Não se esqueçam de contactar o vosso professor se tiverem dúvidas. Instruções adicionais poderão ser adicionadas.

Tópicos

1. The IoT Simulator & Dashboard

- **Descrição:** Simular uma *pipeline* IoT completa usando três contentores Docker orquestrados por Compose.
 1. **Sensor Node (Python):** Um script que gera “sensor data” sintéticos (e.g., temperatura, humidade, ou consumo de energia) com ruído aleatório e envia via HTTP POST a cada poucos segundos.
 2. **Collector API (FastAPI):** Recebe os dados, valida-os e anexa-os a um ficheiro de *log* persistente (CSV ou SQLite) armazenado num **volume**.
 3. **Dashboard (Streamlit ou Dash):** Lê os dados persistentes e exibe um gráfico atualizado em tempo real.
- **Tópicos Principais:** Docker Compose, Python Scripting, APIs (FastAPI), Data Visualization, Volumes.

2. Automated System Monitor with Alerts

- **Descrição:** Criar uma solução robusta de monitorização de sistema.
 1. **Agent (Bash Script):** Um script a correr no *host* (ou num contentor privilegiado) que verifica a saúde do sistema (Uso de disco `df`, Memória `free`, ou Load Average `uptime`). Deve correr periodicamente (loop ou cron).
 2. **Lógica:** Se um limiar for ultrapassado (e.g., Disco > 90%), o script deve acionar um alerta.
 3. **Alert Service (Python):** Um script Python “Dockerizado” que aceita a mensagem de alerta como argumento ou variável de ambiente e faz o *log* num ficheiro JSON estruturado com *timestamp*.
 4. **Relatório:** Usar **Pandas** para analisar o ficheiro de *log* gerado e imprimir um resumo de quantos alertas ocorreram por dia.
- **Tópicos Principais:** Advanced Bash Scripting, System Commands, Python Data Processing, Error Handling.

3. The “Stock Market” Analyzer

- **Descrição:** Construir um serviço que segue e visualiza dados financeiros ou de cripto.
 1. **Fetcher (Python):** Um script que procura dados reais numa API pública (e.g., CoinGecko ou uma API de ações) para 3 ativos diferentes durante um período específico.
 2. **Storage:** Guardar estes dados brutos num ficheiro CSV estruturado.
 3. **Web Server (Nginx + HTML/JS):** Servir uma página *web* estática. Usar **JavaScript (Fetch)** para carregar os dados CSV/JSON e renderizar um gráfico de comparação usando uma biblioteca como **Chart.js** ou **Leaflet** (se mapear localizações).
 4. **Análise:** Um script Python separado deve calcular a “Volatilidade” (desvio padrão) dos ativos e enviar o ativo de maior risco para a consola.
- **Tópicos Principais:** External APIs, CSV handling, Web Server (Nginx), JavaScript/DOM, Statistical Analysis (Pandas).

4. The Secure Document Vault

- **Descrição:** Criar um sistema de armazenamento de ficheiros que demonstre segurança e *scripting*.
 1. **Uploader (Python/FastAPI):** Uma API que aceita *uploads* de ficheiros. Antes de guardar, deve fazer o *hash* do ficheiro (SHA256) para garantir a integridade.
 2. **Encryptor (Bash Script):** Um script que monitoriza a pasta de *upload*. Quando um novo ficheiro chega, encripta-o usando `gpg` ou `openssl` com uma chave/frase-passe pré-definida e apaga o original.
 3. **Access:** Um *frontend* HTML simplificado que lista os ficheiros disponíveis (encriptados).
 4. O relatório do projeto deve explicar as permissões de ficheiros usadas (`chmod/chown`) para garantir que apenas o script consegue ler os ficheiros brutos (*raw*).
- **Tópicos Principais:** Security concepts (Hashing/Encryption), Bash Automation, API File Handling, Linux Permissions.

5. The “Markdown Report” CI/CD Pipeline

- **Descrição:** Simular uma *pipeline* de Integração Contínua (CI) para gerar documentação.
 1. **Environment:** Um contentor Docker contendo **Pandoc**, **LaTeX** e **Python**.
 2. **Trigger:** Um **Bash script** monitoriza uma pasta por alterações num ficheiro `data.csv`.
 3. **Action:** Quando os dados mudam:
 - Correr um script Python para gerar novos gráficos (PNGs) a partir do CSV.
 - Atualizar um ficheiro Markdown para incluir a nova data e estatísticas.
 - Correr **Pandoc** para compilar o Markdown + Gráficos num relatório PDF final.
- **Tópicos Principais:** Dockerfiles (Custom Images), Bash Monitoring, Document Compilation, Python Plotting (Matplotlib).

6. Network Reconnaissance & Visualization

- **Descrição:** Uma ferramenta para fazer *scan* a uma rede e visualizar dispositivos conectados.
 1. **Scanner (Bash/Python):** Um script que executa um *ping sweep* (ou usa `nmap` se instalado) na *subnet* do contentor local para descobrir endereços IP ativos.
 2. **Logger:** Armazenar os IPs ativos e os seus tempos de resposta numa base de dados ou CSV.
 3. **Visualizer (Python/Web):** Um serviço *web* que exhibe a topologia da rede (quais IPs estão ativos) e um histograma dos tempos de resposta.
 4. **Deployment:** Deve ser passível de *deploy* via um único comando `docker-compose up` que cria uma rede privada para testar o *scanner*.
- **Tópicos Principais:** Networking (Subnets/IPs), Docker Networking, Scripting, Data Visualization.

Github Classroom Access

Aqui estão instruções detalhadas para aceder ao GitHub Classroom. A maioria dos alunos pode saltar vários passos, dado que estes foram completados em projetos anteriores.

1. Join the Assignment and Form Your Team

1. **Aceder ao link:** Ir [aqui](#)

2. **Encontrar o vosso nome:** Seleccionem 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 **Prof. Mário Antunes**.
 3. **Criar uma equipa (Create a team):** Sigam esta estrutura de nomeação exata (o nmec deve estar ordenado): [nmec]_project03
 - (Exemplo: 132745_project03)
-

2. Access the Organization and Repository

1. **Aceitar o convite por email:** Após se juntarem a uma equipa, todos os membros receberão um convite por email 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.
 4. **Verificar acesso:** Devem agora ver e ter acesso ao repositório de trabalho da vossa equipa.
-

3. Configure an SSH Key for Access

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

1. **Verificar se existe uma chave SSH:** Abram o 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 passem para o passo 3.
- Se virem um erro como "No such file or directory," corram o seguinte comando para criar uma nova chave:

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

- Depois de gerada, corram `cat ~/.ssh/id_ed25519.pub` novamente para ver a nova chave e copiem-na.

3. **Adicionar a chave à vossa conta GitHub:**

- Vão às **Settings** do GitHub.
- No menu à esquerda, cliquem em **SSH and GPG keys**.
- Cliquem no botão **New SSH key**.
- Dêem-lhe um **Title** (e.g., "O Meu Portátil UA").
- Colem a chave que copiaram no campo **Key**.
- Certifiquem-se 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**.
- Seleccionem a organização **detiuaveiro**, preencham os vossos dados de *login* e concedam acesso.