

ATENÇÃO: Responda às questões **6 a 9** num conjunto de folhas e às questões **10 a 12** noutro conjunto.

6. [3] Num sistema Linux é possível utilizar sockets de domínio Unix (*Unix Domain Sockets*).
- [2] Indique qual é a diferença fundamental entre este tipo de *sockets* e os *sockets* de Internet, incluindo qual a sua principal vantagem e a principal limitação.
 - [1] Qual é a forma dos endereços que são usados com *sockets* de domínio Unix e como diferem dos endereços de *sockets* de Internet?
7. [2.5] Alguns processos ativos num sistema Linux correspondem a serviços a operar como *daemons*. Indique a consequência dessa situação para os canais de *standard output* e de *standard error* desses processos e que mecanismo deve incluir um gestor de serviços avançado (como, por exemplo, o *systemd*) para lidar com os dados que possam surgir nesses canais.
8. [2.5] Num sistema Linux com *systemd*, o ficheiro `/etc/systemd/system/tp2.socket`, na sua secção `[Install]` tem a linha `WantedBy=sockets.target`. Para que serve a secção `Install`, porquê `WantedBy` e o que representa `sockets.target`?
9. [3] Os hipervisores são tipicamente classificados como sendo de Tipo 1 ou de Tipo 2. Um destes tipos, pelas suas características, é geralmente mais eficiente do que o outro na gestão e consumo de recursos computacionais da máquina base. Qual destes tipos tende a ser mais eficiente, porquê e porque motivo(s), apesar disso, existem os dois tipos de hipervisores, por vezes até vindos da mesma empresa?
10. [2.5] Num sistema Linux com o serviço *docker* ativo, estão em execução dois contentores baseados na imagem `node:current-alpine3.18` e um baseado na imagem `php:8.2-alpine3.18`. Nos respectivos `Dockerfile`, ambas as imagens começam com a linha `FROM alpine:3.18`. Havendo neste cenário duas imagens e três contentores, justifique quantas cópias do sistema de ficheiros definido pela imagem base `alpine:3.18` deverão existir no disco do sistema *host*?
11. [4] Considere o `Dockerfile` apresentado e três ficheiros, presentes na subdiretoria `src`: `package.json` e `app.js`, com uma aplicação para `Node.js`, e um `README.md`
- [1.5] Indique que camadas (não-vazias) do sistema de ficheiros são criadas sobre a imagem base `node:alpine`
 - [2.5] Modifique o `Dockerfile` para tirar melhor proveito da *build cache* relativa à atual linha 4, minimizando o número de camadas finais (não-vazias) no sistema de ficheiros, indicando quais serão.

1	FROM node:alpine
2	WORKDIR /home/node/app
3	COPY src/ ./
4	RUN npm install
5	WORKDIR /home/node
6	RUN chown -R node.node .
7	USER node
8	EXPOSE 80
9	WORKDIR /home/node/app
10	CMD ["node", "app.js"]