

Projeto Covid-IUL (Parte 1)

O presente trabalho visa aplicar os conhecimentos adquiridos durante as aulas de Sistemas Operativos e será composto por três partes, com o objetivo de desenvolver os diferentes aspetos da plataforma **Covid-IUL**. Iremos procurar minimizar as interdependências entre partes do trabalho. Este enunciado detalha apenas as funcionalidades que devem ser implementadas na parte 1, sendo que os outros dois enunciados serão publicados no decorrer do semestre.

A plataforma **Covid-IUL** destina-se à marcação e efetuação da vacinação de cidadãos para a Covid-19. Na Plataforma **Covid-IUL**, existem os seguintes conceitos:

- **Cidadãos:** a pessoa que precisa de ser vacinada. O cidadão tem os seguintes dados: nome, número de utente, idade, localidade, número de telemóvel, estado da vacinação (0 – não vacinado, 1 - 1ª inoculação, 2 - 2ª inoculação);
- **Enfermeiros:** a pessoa que efetua a marcação e vacinação dos cidadãos. Um enfermeiro tem os seguintes dados – Nome, número de cédula profissional, Centro de Saúde associado (pela localidade), número de vacinações para covid efetuadas, disponibilidade (0 – indisponível, 1 – disponível).
- **Agenda de vacinação:** Agendamento do dia da vacinação, número e nome do utente a ser vacinado e se é a 1ª ou 2ª inoculação, e nome e número da cédula do enfermeiro que vai efetuar a vacinação.

Em termos gerais, quando chega um lote de X vacinas, o primeiro enfermeiro disponível trata de contactar X cidadãos (não vacinados) para agendar a vacinação contra a covid-19. Para isso, acede à lista de cidadãos registados no centro de saúde onde trabalha e envia uma mensagem com a marcação da data de vacinação a cada um dos cidadãos, registando esses dados no ficheiro “agenda.txt”

Entrega, relatório e autoavaliação

O trabalho de SO este ano será realizado **individualmente**, logo sem recurso a grupos.

A entrega da Parte 1 do trabalho será realizada através da criação de **um** ficheiro ZIP cujo nome é o nº do aluno, e.g., “a<nºaluno>-parte-1.zip” (**ATENÇÃO: não serão aceites ficheiros RAR, 7Z ou outro formato**) onde estarão todos os ficheiros criados. Estes serão apenas os ficheiros de código, ou seja, na primeira parte, apenas os ficheiros Shell (*.sh). Cada um dos módulos será desenvolvido com base nos ficheiros fornecidos, e que estão na diretoria do Tigre “/home/so/trabalho-2020-2021/parte-1”, e deverá incluir nos comentários iniciais um “relatório” indicando a descrição do módulo e explicação do mesmo (poderá ser muito reduzida se o código tiver comentários bem descritivos). Naturalmente, deverão copiar todos estes ficheiros para a vossa área.

Para criarem o ficheiro ZIP, deverão usar, no Tigre, o comando \$ **zip a<nº aluno>-parte-1.zip <ficheiros>**, por exemplo:

```
$ zip a123456-parte-1.zip *.sh
```

A entrega desta parte do trabalho deverá ser feita por via eletrónica, através do e-learning:

- e-learning da UC Sistemas Operativos;
- Seleccionam a opção sub-menu “Conteúdo/Content”;
- Seleccionem o link “Trabalho Prático 2020/2021 Parte 1”;
- Dentro do formulário “Visualizar Exercício de carregamento: Trabalho Prático 2020/2021 Parte 1”, seleccionem “Anexar Arquivo” e anexem o vosso ficheiro .zip. Podem submeter o vosso trabalho as vezes que desejarem, **apenas a última submissão será contabilizada**. Certifiquem-se que a submissão foi concluída, e que esta última versão tem todas as alterações que desejam entregar dado que os docentes apenas considerarão esta última submissão;
- Avisamos que a hora de entrega final acontece sempre poucos **minutos antes da meia-noite**, pelo que se urge a que os alunos não esperem pela hora final para entregarem e o façam, idealmente um dia antes, ou no pior caso, pelo menos uma hora antes. **Não serão consideradas válidas as entregas realizadas por e-mail**. Poderão testar a entrega nos dias anteriores para perceberem se têm algum problema com a entrega, sendo que, novamente, apenas a última submissão conta.

Política em caso de fraude

O trabalho entregue deve corresponder ao esforço individual de cada aluno. São consideradas fraudes as seguintes situações: Trabalho parcialmente copiado, facilitar a cópia através da partilha de ficheiros, ou utilizar material alheio sem referir a sua fonte.

Em caso de deteção de algum tipo de fraude, os trabalhos em questão não serão avaliados, sendo enviados à Comissão Pedagógica ou ao Conselho Pedagógico, consoante a gravidade da situação, que decidirão a sanção a aplicar aos alunos envolvidos. Serão utilizadas as ferramentas *Moss* e *SafeAssign* para deteção automática de cópias. Recorda-se ainda que o Anexo I do Código de Conduta Académica, publicado a 25 de janeiro de 2016 em Diário da República, 2ª Série, nº 16, indica no seu ponto 2 que quando um trabalho ou outro elemento de avaliação apresentar um nível de coincidência elevado com outros trabalhos (percentagem de coincidência com outras fontes reportada no relatório que o referido software produz), cabe ao docente da UC, orientador ou a qualquer elemento do júri, após a análise qualitativa desse relatório, e em caso de se confirmar a suspeita de plágio, desencadear o respetivo procedimento disciplinar, de acordo com o Regulamento Disciplinar de Discentes do ISCTE - Instituto Universitário de Lisboa, aprovado pela deliberação nº 2246/2010, de 6 de dezembro.

O ponto 2.1 desse mesmo anexo indica ainda que no âmbito do Regulamento Disciplinar de Discentes do ISCTE-IUL, são definidas as sanções disciplinares aplicáveis e os seus efeitos, podendo estas variar entre a advertência e a interdição da frequência de atividades escolares no ISCTE-IUL até cinco anos.

Parte I - Shell Script (bash)

Data de entrega: **14 de março de 2021**

Nesta fase do trabalho será criado um conjunto de Shell scripts para administração e gestão do sistema. Para trabalhar com o ficheiro de dados preenchido, crie um soft link de nome **listagem.txt** na diretoria dos scripts, que aponte para o ficheiro **/home/so/trabalho-2020-2021/listagem.txt** do servidor Tigre.

1) lista_cidadaos.sh

Este script deverá ler o ficheiro **listagem.txt** (link anteriormente criado), transformar os dados dos cidadãos que nele constam na “lista de cidadãos do sistema Covid-IUL” e registá-los, criando um novo ficheiro, na sua própria diretoria (apagando um já existente) chamado **cidadaos.txt**, cuja sintaxe é a seguinte:

<número de utente>:<nome>:<idade>:<localidade>:<nº telemóvel>:<estado vacinação>

Sendo que:

- O campo **<número de utente>** é definido como sendo o nº de linha do cidadão no ficheiro **listagem.txt** somado do valor 10000 (ou seja, 1º cidadão: 10001, 2º cidadão: 10002, etc.);
- O campo **<nome>** é extraído diretamente do ficheiro **listagem.txt**;
- O campo **<idade>** será calculado somente com base no ano de nascimento do cidadão em relação ao ano 2021. Portanto, mesmo um cidadão nascido em 2001-12-31 já terá 20 anos;
- O campo **<localidade>** é extraído diretamente do ficheiro **listagem.txt**;
- O campo **<nº telemóvel>** é extraído diretamente do ficheiro **listagem.txt**;
- O campo **<estado vacinação>** serve para ter os valores 0 (não vacinado), 1 (vacinado 1ª inoculação) ou 2 (vacinado 2ª inoculação), mas será sempre **iniciado com o valor 0**.

Exemplo de registos completos do ficheiro **cidadaos.txt**, já com todos os campos preenchidos:

10001:Joao Silva:86:Lisboa:912345678:0

10002:Pedro Santos:65:Santarem: 962345678:0

O script deve mostrar toda a lista de cidadãos no final da sua execução.

2) adiciona_enfermeiros.sh

Este script é executado quando se pretende adicionar um enfermeiro à lista de enfermeiros da plataforma. O script é executado com os seguintes argumentos:

**./adiciona_enfermeiros.sh <nome> <nº de cédula profissional> <centro de saúde associado
"CS"Localidade> <disponibilidade (0 – indisponível, 1 – disponível)>**

Exemplo de invocação do script, que deverá receber os valores pedidos passados por argumentos:

./adiciona_enfermeiros.sh "Ana Correia" 12345 "CSLisboa" 0

./adiciona_enfermeiros.sh "Joao Vieira" 98765 "CSPorto" 1

Com os campos inseridos como descrito acima, o script deve:

2.1) Verificar se já existe um enfermeiro associado ao centro de saúde (CS) introduzido.

2.1.1) Se já existir um enfermeiro do CS introduzido no ficheiro **enfermeiros.txt**, deve mostrar uma mensagem com “Erro: O Centro de Saúde introduzido já tem um enfermeiro registado”;

2.1.2) Se não houver nenhum enfermeiro associado ao CS introduzido, deve verificar se o enfermeiro não está já inscrito noutro CS (usando o número da cédula profissional) e,

2.1.3) se não estiver, deve acrescentá-lo ao ficheiro **enfermeiros.txt**, mantendo a seguinte sintaxe:

**<número cédula profissional>:<nome>:<centro saúde associado>:<nº de vacinações
efetuadas>:<disponibilidade>**

Exemplo de alguns registos do ficheiro **enfermeiros.txt**:

12345:Ana Correia:CSLisboa:0:0

98765:Joao Vieira:CSPorto:0:1

Nesta fase, o nº de vacinações efetuadas é mantido a “0” (zero).

Não deve ser possível executar este script sem passar **todos** os argumentos pedidos. Nesse caso, deverá mostrar a mensagem “Erro: Síntaxe: \$0 <número cédula profissional>:<nome>:<centro saúde associado>:<nº de vacinações efetuadas>:<disponibilidade>”.

O script deve mostrar toda a lista de enfermeiros no final da sua execução.

3) stats.sh

Este script é executado para obter informações sobre o sistema. Nomeadamente, deve retornar a seguinte informação, consoante os argumentos passados:

`./stats.sh cidadaos <localidade>`

- Resultado: “O número de cidadãos registados em <localidade> é <nº cidadãos>”.

`./stats.sh registados`

- Resultado: Lista contendo o Nome e Número de utente dos cidadãos com mais de 60 anos registados na plataforma, ordenados do mais velho para o mais novo.

`./stats.sh enfermeiros`

- Resultado: Lista com o Nome dos enfermeiros assinalados como **disponíveis** na plataforma.

4) agendamento.sh

Este script será responsável por agendar as vacinações. Para isso, sempre que for executado deve:

4.1) Caso já exista, deve apagar o ficheiro de **agenda.txt**.

4.2) No caso de haver enfermeiros disponíveis na plataforma, deve ser efetuada a marcação da vacinação de todos os cidadãos inscritos no Centro de Saúde associado ao enfermeiro. Para isso, deve ser registado no ficheiro **agenda.txt** o **nome e número da cédula pessoal do enfermeiro** que vai efetuar a vacinação, o **nome e número de utente do cidadão** que vai ser vacinado, o **centro de saúde** onde vai ser realizada a vacinação e a **data da vacinação**. (DICA: A data corresponde à data atual – analise o comando “date” e as suas possíveis formatações).

EXEMPLO:

Considerando que o conteúdo dos ficheiros enfermeiros.txt e cidadãos.txt é o seguinte:

Enfermeiros.txt
12345:Ana Correia:CSLisboa:0:0
98765:Joao Vieira:CSPorto:0:1

```
Cidadãos.txt
10001:Joao Silva:86:Lisboa:912345678:0
10002:Pedro Santos:65:Santarem:962345678:0
10003:Joaquim Pedroso:91:Porto:966645678:0
10004:Ana Silva:54:Porto:917745678:0
```

Neste caso, devem apenas ser contactados os cidadãos do Porto, uma vez que o único enfermeiro disponível trabalha no CSPorto. Assim, o ficheiro **agenda.txt** ficará com o seguinte conteúdo:

```
agenda.txt
Joao Vieira:98765:Joaquim Pedroso:10003:CSPorto:2021-03-01
Joao Vieira:98765:Ana Silva:10004:CSPorto:2021-03-01
```

5) menu

Este script agrega os restantes, mostrando um menu com as opções:

1. Listar cidadãos
2. Adicionar enfermeiro
3. Stats
4. Agendar vacinação
0. Sair

O script deve esperar que o utilizador escolha uma opção, e invocar o respetivo script descrito nas alíneas anteriores. No caso das opções 2 e 3, este script deverá pedir ao utilizador as informações necessárias para execução do script correspondente, injetando as mesmas como argumentos desse script.

Após a execução de cada sub-script, o menu deverá voltar a ser apresentado, e nova opção pedida. Apenas a opção 0 (zero) permite sair deste script “menu”. Até escolher esta opção, o menu deverá ficar em ciclo, permitindo realizar múltiplas operações **iterativamente** (e não recursivamente!).

É da responsabilidade de cada script fazer a validação dos argumentos que recebe, nomeadamente se é obrigatório ou não, ou o seu formato. Da mesma forma, é da responsabilidade de cada script validar que tem todos os recursos necessários, nomeadamente ficheiros de input e outros. Em caso de erro, cada script deverá fazer output das mensagens de erro corretas. E, mesmo em caso de sucesso, as mensagens de resultado deverão ser completas, com um formato inteligível e não apenas códigos ou números.

Boa sorte!!!