

Departamento de Ciências e Tecnologias da Informação

Projeto Cliniq-IUL

Henrique Eduardo Rodrigues da Cunha Fonseca
(ET-B4 – 94089)

Relatório do Projeto Cliniq-IUL (Parte 1) da UC Sistemas Operativos

Ano Letivo 2020/2021

Outubro, 2020

Parte I – Shell Scrip (bash)

O presente relatório visa apresentar informações sobre a elaboração da **Parte 1** do projeto **Clinic-IUL** da UC **Sistemas Operativos**. Nesta fase do trabalho foi criado um conjunto de **5 Scripts** para administração e gestão do sistema **Cliniq-IUL**:

1) **cria_pacientes.sh**

Este *script* lê o ficheiro **/etc/passwd** e transforma os **10 primeiros utilizadores alunos** (nome de utilizador do tipo “a94089”) em “**Pacientes do Sistema Cliniq-IUL**”, guardando os mesmos no ficheiro **pacientes.txt**. De maneira a prevenir que **não existam alunos repetidos** o *script* **verifica** se o **aluno** que se pretende registar **já está registado** no ficheiro **pacientes.txt** através do **Número de Registo** (número de identificação).

2) **cria_medico.sh**

Este *script* serve para **inscrever um médico na plataforma**, guardando o registo de cada médico no ficheiro **medicos.txt**. O *script* é executado com os seguintes **argumentos**: **Nome**, **N.º de Cédula Profissional**, **Especialidade Médica** e **E-mail** (se o *script* for executado com um número de **argumentos diferente de 4**, este **não regista** o médico e mostra uma mensagem de **erro** ao utilizador). De forma a **não existirem registos repetidos** de médicos, o *script* usa os argumentos **N.º De Cédula Profissional** e **E-mail** para **verificar** se já existe algum **registo** no ficheiro **medicos.txt** com algum destes campos (no caso do e-mail esta **verificação** é feita de forma “**Case Insensitive**” – Name@domain.com = nAmE@DoMain.COM – usando a **flag “-i”** com o comando **grep**) e, se existir, mostra o médico em questão já registado. De maneira a manter a **integridade da informação** guardada e tentar prevenir eventuais erros, o *script* tem algumas **restrições** em relação aos **argumentos** (com mensagens de **erro** se as mesmas não forem cumpridas): **Nome** e **Especialidade Médica** – não podem conter números; **N.º de Cédula** – só pode conter inteiros positivos; **E-mail** – tem de conter o carácter “@”.

3) **stats.sh**

Este *script* é executado com os argumentos: **Localidade** e **Saldo** (se o *script* for executado com um **número de argumentos diferente de 2**, este **não devolve informação** e mostra uma **mensagem de erro**) devolvendo o **N.º de Pacientes** inscritos **dessa Localidade** e o **N.º de Médicos** inscritos com o **Saldo superior ao Saldo passado como argumento**. De maneira a manter a **integridade da informação** guardada, o *script* tem algumas **restrições** em relação aos **argumentos** (com mensagens de **erro** se as mesmas não forem cumpridas): **Localidade** – não pode conter números; **Saldo** – só pode conter inteiros positivos. No mesmo âmbito, o Saldo não pode começar com “0” (ex.: 06 – Não Permitido / 6 – Permitido) e o *script* preforma a **pesquisa dos pacientes inscritos** na **Localidade** de forma “**Case Insensitive**” (Lisboa = lIsboa). Ainda na mesma ótica, se o *script* for executado e o **ficheiro pacientes.txt** estiver **vazio** ou **não existir**, este **executa** o *script* **cria_pacientes.sh** (esta foi a estratégia escolhida uma vez que o *script* **cria_pacientes.sh** não necessita de argumentos). No entanto, se for **executado** e o **ficheiro medicos.txt** **não existir** ou estiver **vazio**, surge uma mensagem de **erro** e **não é devolvida informação** nenhuma.

4) `avalia_medicos.sh`

Este *script* serve para, no caso de haver médicos inscritos na plataforma cujo **número de consultas** efetuadas seja **superior a 6** e o seu **rating inferior a 5**, seja dada **opção** (através de uma mensagem) ao administrador para **colocar** o respetivo **médico** na **lista negra** da plataforma – **lista_negra_medicos.txt**. Esta opção só é dada se o médico ainda não estiver registado na lista negra (esta verificação é feita através do n.º de cédula profissional). Se este *script* for **executado** e o ficheiro **medicos.txt** **não existir** ou estiver **vazio**, este mostra uma mensagem de **erro** e **não preforma ação** nenhuma.

5) `menu.sh`

Este *script* agrega os restantes, mostrando um menu com as opções ([0;4]) para **invocar** cada um dos *scripts* anteriores (ou sair do *script* no caso da opção 0). No caso dos *scripts* **cria_medico.sh** e **stats.sh** este **pede** as **informações necessárias** ao utilizador para as **injetar** nos **respetivos scripts**. Até ser escolhida uma opção o **menu** permanece num **ciclo** (ex.: while true), permitindo assim realizar **múltiplas operações iterativamente**.

Se o utilizador inserir uma **opção** que **não** seja um **número** ou um **número** que **não pertença** ao **intervalo [0;4]**, o *script* entra num **ciclo de validação** da opção, até que a mesma seja **válida**.

Abordagem

Nos *scripts* em que existe a necessidade de, por exemplo, **verificar** se um **médico** já está **registado** na plataforma (através do n.º de cédula profissional), foi usado o comando **grep** com a **flag “-x”** para evitar eventuais erros, ex.: o comando – `cat medicos.txt | cut -d “;” -f2 | grep “94089”` – **devolveria simultaneamente** tanto um **médico** que estivesse registado com o **número “94089”** como um que estivesse registado com o número **“9408900”**.

A abordagem nos *scripts* para **aceder** a um ficheiro **linha a linha** foi usar uma **combinação de head e tail**, e uma **variável de iteração i** relativa ao **head**, ex.: o comando – `cat medicos.txt | head -5i | tail -1 | cut -d “;” -f1` – mostra, linha a linha, o nome de cada médico registado (i=1 devolve o nome do médico da 1ª linha, i=2 devolve o nome do médico da 1ª linha, e assim sucessivamente).

Nos *scripts* **cria_medico.sh**, **stats.sh** e **avalia_medico.sh**, existe o comando – `echo "$(cat medicos.txt | sed '/^[:space:]*$/d')" > medicos.txt` – que **elimina** todas as **linhas sem conteúdo** entra cada registo. Esta estratégia serve para **prevenir** eventuais **erros** no caso de o **utilizador** decidir usar estes *scripts* **fornecendo** ele próprio os documentos **pacientes.txt**, **médicos.txt** ou **lista_negra_medicos.txt**.