

## Projeto *Spotif-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 objectivo de desenvolver os diferentes aspetos da plataforma Spotif-IUL. Iremos procurar minimizar as inter-dependências 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 Spotif-IUL destina-se à partilha de músicas entre estudantes, docentes e colaboradores do ISCTE-IUL. Este programa (fictício!) pretende estimular o estudo e o exercício físico. Para além de poderem ouvir as músicas preferidas, poderão também criar e partilhar playlists com os outros ouvintes.

**Ouvinte:** é a pessoa que deseja ouvir música e criar playlists, neste momento restrita ao universo dos alunos de Sistemas Operativos. Um ouvinte tem um determinado perfil que inclui um *nickname*, *password*, *número*, *nome*, *e-mail*, *curso* e *playlists* a que está associado.

**Música:** é uma música que o ouvinte pode escolher, caracterizada pelo seguinte conjunto de atributos: *identificador da música* (da forma *MUS\_número*), *nome da música*, *artista*, *duração* (em minutos), *ano de produção*, *género*, *posição no top*.

**Playlist:** Conterá a lista de playlists criadas pelos ouvintes, caracterizada pelo seguinte conjunto de atributos: *identificador da playlist* (da forma, *PL\_número*), *nome da playlist*, *músicas* (identificador de músicas da forma *MUS\_número*).

Resumidamente, os ouvintes registam-se na aplicação Spotif-IUL, escolhendo um *nickname* e uma *password* que depois usarão para poder ouvir as músicas. A aplicação também oferece a funcionalidade de criação e associação a playlists por parte do ouvinte, que poderão ser partilhadas com os colegas. Qualquer ouvinte se pode associar a uma playlist sem necessidade de qualquer autorização por parte do criador da playlist.

## Constituição dos grupos

O trabalho será realizado em grupos de 2 ou 3 estudantes (só em casos excepcionais podem ser aceites grupos de um estudante). Os grupos devem começar por se registar usando o e-learning:

- e-learning da UC Sistemas Operativos
- Escolher Menu **Grupos**
- Escolher um grupo disponível **que tenha o nome do seu docente das aulas práticas.**

A constituição dos grupos deverá manter-se até ao final do semestre, mas caso se verifiquem alterações, deve ser criado um novo grupo que passará a ser usado em vez do anterior para submeter os trabalhos.

## Entrega, relatório e auto-avaliação

A entrega da Parte 1 do trabalho deverá ser feito por via eletrónica, através também do e-learning:

**Parte 1:** <Link para entrega do trabalho no e-learning, a indicar mais tarde>

*Observação.* A submissão de um ficheiro substitui a versão anteriormente submetida. Realça-se que o formulário acima destina-se apenas à entrega da Parte 1 (nos outros enunciados iremos facultar novos formulários para submissão das respectivas partes).

Na submissão deverá enviar um ficheiro **zip** com o nome do grupo e indicando que é a parte 1, e.g., “**so-cc-g01-parte1.zip**”, contendo os ficheiros criados ou modificados pelo estudante (ex: \*.sh , \*.txt, \*.c, \*.h, ...) de acordo com o formato pedido no enunciado, juntamente com o ficheiro *relatório* que é mencionado de seguida. **Os scripts entregues deverão respeitar o nome dos ficheiros mencionados no enunciado.** Caso não o façam, o respectivo trabalho sofrerá uma penalização na nota final. **Os ficheiros auxiliares que os alunos criarem para os seus testes (por exemplo ficheiros .txt), não devem ser submetidos juntamente com os scripts ou programas antes referidos** (excetuando os respeitantes à configuração de ferramentas como o cron).

Deverá ser criado um relatório em formato TXT ou PDF com informações sobre a elaboração do projeto. Esse ficheiro deve chamar-se **relatorio.txt** ou **relatorio.pdf**, deverá ter apenas 1 página A4 e deverá ser submetido juntamente com o trabalho.

O projeto será avaliado em termos globais produzindo-se uma nota de projeto, que será distribuída por todos os elementos do grupo, caso estes tenham todos contribuído de igual forma para o projeto. Caso existam diferenças no nível de contribuição, então os estudantes que mais contribuíram poderão ter uma nota individual superior à nota de projeto e o inverso para os estudantes que menos contribuíram. No fundo, parte da nota individual dos estudantes que menos contribuíram pode ser transferida para os estudantes que mais contribuíram, recompensando assim o esforço adicional de quem mais contribuiu. Assim, **o grupo deverá indicar no seu relatório uma estimativa de qual a percentagem de**

**contribuição de cada elemento do grupo para o conjunto das tarefas realizadas (entre 0% e 100%)**, por forma a aferir qual foi a contribuição individual de cada elemento do grupo. É ainda requerido uma justificação sucinta do porquê das diferenças nas percentagens atribuídas aos vários elementos do grupo (caso essas diferenças existam).

É importante que os estudantes que menos contribuíram para o projeto façam questão de garantir que essa menor contribuição fique expressa neste documento, pois, caso contrário, estão a ficar com os créditos dos colegas que mais se envolveram no projeto e, consequentemente, a impedir que as notas individuais desses mesmos colegas possam subir para além da nota de projeto. Portanto, é importante que toda a informação prestada seja realmente representativa do esforço relativo colocado pelos vários elementos, pois, caso contrário, a distribuição de notas poderá não ficar de acordo com as expectativas do grupo. Além disso, a nota final do projeto é atribuída através de uma oral, que acabará por apurar de uma outra forma a contribuição de cada elemento do grupo e que poderá ser comparada com as informações prestadas.

Assume-se que a informação prestada é consensual entre os vários elementos do grupo. Quando isso não for possível, tal facto deverá ficar exposto no documento.

## **Política em caso de fraude**

Os alunos podem partilhar e/ou trocar ideias entre si sobre os trabalhos e/ou resolução dos mesmos. No entanto, o trabalho entregue deve corresponder ao esforço individual de cada grupo. São consideradas fraudes as seguintes situações:

- Trabalho parcialmente copiado
- Facilitar a cópia através da partilha de ficheiros
- 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 n.º 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.

## Ficheiros

O conjunto de *scripts* e programas irão operar com os seguintes ficheiros.

### ouvintes.txt

**Formato:** nickname: password: número de aluno: nome: e-mail: curso: identificador da playlist1: identificador da playlist2

Exemplo:

```
Batman:robin:12345:João Caniço:a12345@iscte-iul.pt:IGE:PL_1:PL_2
asterix:obelix:67890:Marta Santos:a67890@iscte-iul.pt:ETI:PL_2
...
```

Observação: note que cada linha poderá ter número de argumentos variável, já que um ouvinte pode estar associado a mais do que uma playlist (pode até nem estar associado a nenhuma playlist).

### musicas.txt

**Formato:** identificador da música (da forma *MUS\_número*): título da música: artista: duração (em minutos): ano de produção: género: posição no top

Exemplo:

```
MUS_1:Bohemian Rhapsody:Queen:6:1975:rock:40
MUS_2:Ai se ele cai:Xutos e Pontapés:3:2004:rock:27
MUS_3:We are the champions:Queen:3:1977:rock:1
MUS_4:Estou alem:Antonio Variacoes:4:1983:rock:13
MUS_5:Like a prayer:Madonna:5:1989:pop:2
...
```

Observação: a identificação das músicas deverá ser sequencial, ou seja, os identificadores devem começar no número 1 e ser incrementados aquando da adição de uma nova música a este ficheiro.

### playlists.txt

**Formato:** identificador da playlist (da forma *PL\_número*): nome da playlist: identificador das músicas (separadas por dois pontos)

Exemplo:

```
PL_1:Jogging:MUS_3:MUS_2
PL_2:Estudo:MUS_5:MUS_3:MUS_4
...
```

Observação: note que cada playlist poderá ter um número variável de músicas.

## Parte I - Shell Script (bash)

**Data de entrega:** 14 de outubro de 2019 às 23:59

Nesta primeira parte do trabalho, propõe-se que os estudantes desenvolvam a parte de Administração do servidor *Spotify-IUL*. Para tal deverão desenvolver os *scripts* descritos abaixo. **Salienta-se que devem verificar a existência de ficheiros, sempre que necessário.**

### 1) *cria\_ouvinte.sh*

Permite acrescentar um ouvinte ao ficheiro *ouvintes.txt*. A informação sobre o nickname, password, número de aluno e curso (IGE, ETI ou LEI) deve ser passada como argumento, isto é, a informação é pedida e validada no menu e só depois enviada para o *script* *cria\_ouvinte.sh*. Deve validar a existência do número de aluno no ficheiro */etc/passwd* do servidor *tigre.iul.lab*, pois só aceita alunos inscritos no tigre. Depois desta verificação, deve calcular a restante informação e registar o ouvinte no ficheiro *ouvintes.txt* usando a sintaxe definida anteriormente. Deve também verificar se ao curso é um dos três referidos e se o ouvinte já existe no ficheiro por forma a impedir a criação de duplicados.

Exemplo:

```
./cria_ouvinte.sh Batman robin 88020 ETI
```

A restante informação é calculada com base na informação existente da seguinte forma: o nome é diretamente extraído do ficheiro */etc/passwd* e o e-mail deverá ser criado automaticamente tendo por base o número do aluno e acrescentando “@iscte-iul.pt”. Aquando da criação de um ouvinte, este não deverá ser associado a nenhuma playlist. Esta operação é feita através do *script* *associa\_ouvinte\_playlist.sh* descrita abaixo.

Nota: No caso de não estar a utilizar o tigre, pode simplesmente copiar o ficheiro */etc/passwd* do servidor *tigre.iul.lab* para a sua diretoria e trabalhar com essa versão do ficheiro. Contudo, deve garantir que a versão final do trabalho lê o ficheiro correctamente, pois será nestas condições que irá ser testado e avaliado.

### 2) *adiciona\_musica.sh*

Este *script* permite adicionar uma música ao ficheiro *musicas.txt*. Para o efeito, deve pedir as informações relevantes ao utilizador: nome da música, artista, duração, ano de produção e género. O campo da posição no top deverá ser mantido vazio (espaço em branco). Tem de verificar se o nome da música e o artista já existem no ficheiro *musicas.txt*, por forma a impedir a criação de duplicados (note que a mesma música pode ter várias entradas de artistas diferentes e vice-versa). Caso seja válida, adiciona a música no ficheiro respectivo, onde o identificador é criado automaticamente (para o efeito terá de manter um contador atualizado, por forma a garantir uma numeração sequencial).

### 3) **atualiza\_musicas.sh**

Permite não só atualizar as informações preentes no ficheiro *musicas.txt*, mas também acrescentar várias músicas de uma só vez a partir de um ficheiro definido pelo administrador. Este *script* recebe como parâmetro o nome do ficheiro onde estão registadas as músicas a atualizar ou acrescentar. Este ficheiro está organizado da mesma forma que o ficheiro *musicas.txt*, exceptuando o identificador. Um exemplo do ficheiro está representado abaixo:

```
Estou alem:Antonio Variacoes:4:1983:rock&roll:7
Bohemian Rhapsody:Queen:6:1975:rock&roll:1
Ai se ele cai:Xutos e Pontapés:3:2004:rock&roll:101
We are the champions:Queen:3:1977:rock&roll:15
Like a prayer:Madonna:5:1989:pop:5
Garota de Ipanema:Tom Jobim::4:1962:brasileira:3
```

Caso a música exista no ficheiro *musicas.txt* (para tal deve sempre verificar o nome da música e o nome do artista), atualiza as restantes informações. Caso a música não exista então deve acrescentar uma nova linha no ficheiro *musicas.txt*. Note que para esta última funcionalidade deve manter uma identificação sequencial das músicas de acordo com a forma já discutida anteriormente.

### 4) **cria\_playlist.sh**

Este *script* permite ao administrador criar uma playlist e adicionar músicas à mesma. Deverá perguntar ao utilizador qual o nome da playlist e verificar se esta já existe no ficheiro *playlists.txt*. Depois desta confirmação, deve apresentar a lista de músicas presente no ficheiro *musicas.txt* de maneira a que o utilizador possa escolher as suas preferências. Assim que este fizer a sua escolha, a nova playlist deve ser adicionada na última linha do ficheiro *playlists.txt* no formato discutido acima. Pelo menos uma música terá de ser associada à playlist. Note que também neste caso terá de manter um contador atualizado, de maneira a que o identificador da playlist garanta uma numeração sequencial.

### 5) **associa\_ouvinte\_playlist.sh**

Esta funcionalidade permite ao administrador associar um ouvinte a uma playlist. Começa por pedir o nickname do ouvinte e verificar a sua existência no ficheiro *ouvintes.txt*. De seguida, solicita o nome da playlist a que o ouvinte deve ser associado e verifica a existência da mesma, lendo o ficheiro *playlists.txt*. Depois da verificação, deve indicar se a operação teve sucesso com uma mensagem. Se assim for, atualiza o registo do ouvinte no ficheiro *ouvintes.txt*, por forma a adicionar a playlist ao ouvinte respectivo. Caso contrário, mostra uma mensagem de erro e sai do *script*.

### 6) **remove\_musica\_playlist.sh**

Este *script* permite remover uma música de uma playlist. Deve pedir o nome da playlist e listar as músicas da mesma para que o utilizador possa escolher qual quer remover.

Depois da escolha, modifica o ficheiro *playlists.txt* por forma a reflectir esta alteração. Deve garantir que a playlist não fica sem músicas e, se for o caso, remove a playlist do ficheiro *playlists.txt*.

No caso de remoção da playlist, “envia e-mail” a todos os ouvintes que estavam associados a essa playlist. O “envio” de um e-mail vai ser realizado através da escrita de uma linha num ficheiro “emails.txt” em que será colocada uma nova linha por cada e-mail que tem que ser enviado, especificando nessa linha de texto qual o endereço de destino e motivo do e-mail, no formato:

“Envia e-mail para a12345@iscte-iul.pt. Conteúdo: Foi apagada a playlist PL\_6 que estava associada ao seu utilizador”.

Para além de “enviar o e-mail”, também deve apagar a associação das playlists apagadas dos ouvintes.

## 7) stats.sh

Este script é executado pelo administrador do sistema para obter informações sobre o sistema. Nomeadamente, deve devolver a seguinte informação:

- Número de ouvintes, músicas e playlists registadas na plataforma;
- Número total de minutos das músicas (soma da duração de todas as músicas);
- Número de músicas disponíveis numa dada playlist. Para tal deve pedir o nome da playlist ao utilizador e validar a sua existência;
- Ouvintes por curso: estatística com o número de ouvintes por curso (por exemplo: IGE: 5, ETI: 12, LEI: 2).

## 8) publica\_playlists.sh

Este *script* deverá publicar as playlists dos utilizadores em formato HTML com base nos ficheiros existentes. O ficheiro resultante deverá estar estruturado de acordo com o código-exemplo seguinte, em que para cada ouvinte se apresentam as playlists associadas e respectivas músicas. Deve respeitar a indentaç  o de cada entrada, de modo a facilitar a sua leitura.

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Ouvintes e playlists da aplica  o Spotify-IUL</title>
  </head>
  <body>
    <h1>Nickname: Batman</h1>
    <h2>Playlist: Jogging</h2>
    <ol>
      <li><b>We are the champions</b> -<i> Artist: </i>Queen;<i> Duration: </i>3 min;
        <i> Year: </i>1977;<i> Genre: </i>rock;<i> Top Position: </i>1</li>
      <li><b>Ai se ele cai</b> -<i> Artist: </i>Xutos e Pontap  s;<i> Duration: </i>3 min;
        <i> Year: </i>2004;<i> Genre: </i>rock;<i> Top Position: </i>27</li>
```



```

</ol>
<h2>Playlist: Estudo</h2>
<ol>
  <li><b>Like a prayer</b> -<i> Artist: </i>Madonna;<i> Duration: </i>5 min;
    <i> Year: </i>1989;<i> Genre: </i>pop;<i> Top Position: </i>2</li>
  <li><b>Estou alem</b> -<i> Artist: </i>Antonio Variacoes;<i> Duration: </i>4 min;
    <i> Year: </i>1983;<i> Genre: </i>rock;<i> Top Position: </i>13</li>
  <li><b>We are the champions</b> -<i> Artist: </i>Queen;<i> Duration: </i>3 min;
    <i> Year: </i>1977;<i> Genre: </i>rock;<i> Top Position: </i>1</li>
</ol>

<hr/>

<h1>Nickname: asterix</h1>
<h2>Playlist: Estudo</h2>
<ol>
  <li><b>Like a prayer</b> -<i> Artist: </i>Madonna;<i> Duration: </i>5 min;
    <i> Year: </i>1989;<i> Genre: </i>pop;<i> Top Position: </i>2</li>
  <li><b>Estou alem</b> -<i> Artist: </i>Antonio Variacoes;<i> Duration: </i>4 min;
    <i> Year: </i>1983;<i> Genre: </i>rock;<i> Top Position: </i>13</li>
  <li><b>We are the champions</b> -<i> Artist: </i>Queen;<i> Duration: </i>3 min;
    <i> Year: </i>1977;<i> Genre: </i>rock;<i> Top Position: </i>1</li>
</ol>
</body>
</html>

```

Use o script para produzir o ficheiro ~/public\_html/stats.html. Verifique que o ficheiro foi bem produzido colocando o seguinte endereço no seu browser:

<http://tigre.iul.lab/~axxxxx/stats.html>

Salienta-se que este *script* pode ser executado a partir do menu pelo administrador (opção 8, ver menu.sh), mas também terá de ser executado diariamente, por forma a garantir que o HTML permanece atualizado. A hora e minutos da execução diária são definidos pelo utilizador através do script `diario_atualiza_html.sh` (ver opção seguinte).

## 9) `diario_atualiza_html.sh`

Este *script* permite ao administrador definir a hora e minutos em que o HTML é atualizado diariamente, pelo que deve pedir esta informação ao utilizador e modificar a entrada no “cron”. Até que o administrador execute este script, a hora predefinida para a atualização é 00:00.

Para a implementação desta funcionalidade terá de adicionar e modificar as entradas do “cron”. Pode encontrar informações de como o fazer na sebenta da UC.

## 10) `menu.sh`

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

```
----- MENU -----  
1. Cria ouvinte  
2. Adiciona musica  
3. Atualiza músicas  
4. Cria playlist  
5. Associa ouvinte a playlist  
6. Remove musica de playlist  
7. Mostra estatísticas  
8. Publica playlists  
9. Atualização diária HTML (configuração)  
0. Sair
```

Cada uma das opções anteriores, invoca o respectivo *script* descrito anteriormente. Note, novamente, que nas opções 1 e 3 as informações do ouvinte deverão ser pedidas ao administrador e injectadas como argumentos no respectivo *script*.

O menu deverá manter-se em ciclo, permitindo realizar multi-operações, até ser escolhida a opção '0', que permite terminar o funcionamento do Admin.