

Os Produtores

Programação Orientada pelos Objectos
Enunciado do 1º Trabalho Prático, versão 1.3 – 2018-04-10
Contacto: mgoul@fet.unl.pt

Notas importantes

Prazo de entrega até às 23h59 do dia 3 de Maio de 2018.

Material a entregar neste trabalho REALIZADO EM GRUPOS DE 2 ALUNOS

- **Mooshak:** submissão e aceitação do código fonte pelo sistema de avaliação automática Mooshak (pontuação > 0). Consulte na página da disciplina as regras de submissão de trabalhos ao Mooshak.

Recomendações: A boa documentação do seu código fonte será valorizada, tal como a utilização do melhor estilo de programação possível, para além, claro, do correcto funcionamento do projecto. Não se esqueça de comentar as declarações das interfaces e das classes, bem como dos seus métodos, com uma explicação do seu significado, colocando pré-condições sempre que se justifique. Os alunos podem e devem tirar todas as dúvidas com os docentes da disciplina, bem como conversar com os colegas e debater soluções – mas nunca partilhando código – sempre respeitando os princípios do Código de Ética (disponível na página da cadeira no Moodle).

1 Desenvolvimento da Aplicação *Os Produtores*

1.1 Descrição do Problema

O objectivo deste trabalho é o desenvolvimento de uma aplicação que permita a gestão das filmagens de uma empresa produtora de conteúdos áudio-visuais (séries de televisão, filmes, etc.). Os utilizadores registados na aplicação são produtores da empresa e outros colaboradores. Os produtores usam a aplicação para gerir a participação dos diversos colaboradores (actores, realizadores, operadores de câmara, sonoplastas, guionistas, etc.) nas gravações. Os colaboradores usam a aplicação para consultar a sua agenda de gravações, podendo também assinalar períodos em que se encontrem indisponíveis para participar em gravações.

Cada colaborador tem um identificador único (o seu *nome*), um *custo horário* e um *calendário das gravações em que tem de participar*. Existem vários tipos de colaboradores registados na aplicação: *produtores seniores*, *produtores juniores* e *outros colaboradores*. Os *produtores seniores* ou *juniores*, têm como missão proceder à marcação de *gravações*. Os *produtores seniores* têm prioridade sobre os *produtores juniores* na marcação de gravações. Isso implica que, quando um *produtor sénior* marca uma gravação que envolva *colaboradores* já alocados por gravações de *produtores juniores*, as gravações desses *produtores juniores* são adiadas para que a gravação do *produtor sénior* possa ocorrer. Entre produtores de um mesmo nível, a prioridade é sempre dada ao produtor que tenha feito primeiro a marcação, em caso de conflito. Uma

gravação adiada é automaticamente remarcada para o primeiro dia disponível, ou seja, em que o local e todos os intervenientes estejam, simultaneamente, livres, com as mesmas horas de início e fim da marcação original.

Entre os *outros colaboradores* podem ser *actores*, *realizadores* e *técnicos* (os operadores de câmara, sonoplastas, guionistas, etc.). Alguns actores e realizadores são autênticas *vedetas*. As *vedetas* têm um tratamento especial para lidar com as rivalidades que, por vezes, assolam o mundo do espectáculo. Por vezes, zangam-se umas com as outras. Assim, cada *vedeta* pode registar as pessoas com quem se recusa a trabalhar (que podem ser quaisquer colaboradores, vedetas ou não, e mesmo produtores, na empresa) numa *lista negra*. É impossível marcar gravações que incluam pelo menos uma *vedeta* zangada com pelo menos outro participante na gravação. Além disso, quando uma vedeta se zanga com alguém, isso tem o efeito de suspender todas as gravações que envolvam a *vedeta* e a pessoa com quem a *vedeta* está zangada. Felizmente, por vezes as *vedetas* fazem as pazes com as pessoas com quem estavam zangadas e voltam a poder trabalhar com essas pessoas. Se houver alguma gravação suspensa por causa de uma zanga que terminou, essa gravação volta a estar agendada na mesma data, hora e local inicialmente previstos. Apenas as *vedetas* se podem recusar a trabalhar com alguém.

Cada *gravação* ocorre num determinado *local*, tendo data e *hora inicial* e *final* para ocorrer. Uma *gravação* envolve sempre pelo menos um *produtor* (que a marca), um *realizador* e um *técnico*, podendo envolver outros *colaboradores* de qualquer tipo. Nenhum dos participantes pode estar envolvido noutras gravações em simultâneo (mesmo que estas estejam suspensas por uma birra de uma *vedeta*).

Os locais e intervenientes numa gravação têm custos associados. Estes custos têm um valor fixo para cada local, ou interveniente, para cada hora de gravação. A duração de uma gravação é arredondada para a hora imediatamente acima. Por exemplo, uma gravação que dure 3h02 terá o custo equivalente a 4 horas de gravação, que se obtém multiplicando por 4 o custo horário do local e intervenientes.

2 Comandos

Nesta secção apresentam-se os vários comandos que o sistema deve ser capaz de interpretar e executar. Nos exemplos apresentados, diferenciamos o **texto escrito pelo utilizador** da retroacção escrita pelo programa na consola, ao executar o comando. Pode assumir que o utilizador não cometerá erros na introdução de argumentos nos comandos, para além dos descritos neste enunciado, ou seja, apenas tem de tratar as situações de erro descritas aqui, pela ordem que são descritas.

Na leitura de comandos o interpretador não deve fazer distinção entre maiúsculas e minúsculas. Por exemplo, para o comando **sai**, o interpretador deve aceitar como válidas as alternativas **SAI**, **sai**, **Sai**, **sAI**, etc. Nos vários exemplos que se seguem, o símbolo ↵ denota a mudança de linha. Além disso, note também a utilização de uma *prompt*, denotada pelo símbolo >. Esta *prompt* indica que o programa aguarda um comando do utilizador. Portanto, **a *prompt* deve ser escrita na consola imediatamente antes da leitura de qualquer comando do utilizador.**

Caso o utilizador introduza um comando inexistente, o programa escreve na consola a mensagem **Opcao inexistente..** Por exemplo o comando inexistente **desconhecido** teria este efeito:

```
> desconhecido↵
Opcao inexistente.↵
```

Vários comandos têm argumentos. Pode assumir que o utilizador apenas usa argumentos

de tipos correctos. No entanto, pode acontecer que os argumentos passados a um desses comandos tenham algum valor incorrecto. Por esse motivo, teremos de testar esses argumentos exactamente pela ordem especificada neste enunciado. Os argumentos são denotados com **este estilo**, na sua descrição.

2.1 Comando **sai**

Termina a execução do programa. O comando não necessita de argumentos.

```
> sai↵
Ate a proxima.↵
```

Este comando tem sempre sucesso. Quando é invocado, o programa termina a sua execução.

2.2 Comando **ajuda**

Indica quais os comandos disponíveis no programa. O comando não necessita de argumentos. Este comando tem sempre sucesso e tem como efeito ser mostrada na consola a lista de comandos disponíveis. No cenário seguinte, o comando é invocado:

```
> ajuda↵
registra - regista um novo colaborador↵
staff - lista os colaboradores registados↵
cenario - regista um novo local para gravacoes↵
cenarios - lista os locais para gravacoes registados↵
marca - marca uma nova gravacao↵
amua - vedeta deixa de trabalhar com colaborador↵
reconcilia - vedeta faz as pazes com colaborador↵
realizadas - lista as gravacoes realizadas↵
previstas - lista as gravacoes previstas↵
local - lista as gravacoes previstas para um local↵
colaborador - lista as gravacoes previstas para um colaborador↵
grava - executa a proxima gravacao agendada↵
amuancos - lista os colaboradores com quem uma vedeta esta amuada↵
ajuda - Mostra a ajuda↵
sai - Termina a execucao do programa↵
```

2.3 Comando **registra**

Regista um novo colaborador. Para tal, indica o **tipo de colaborador**, no caso de ser actor ou realizador a **notoriedade** do colaborador, o **custo por hora de trabalho** e o nome do colaborador. Os tipos de colaborador conhecidos são: produtor sénior (**senior**), produtor júnior (**junior**), realizador (**realizador**), técnico (**tecnico**) e actor (**actor**). O argumento opcional que representa a **notoriedade** apenas é usado para realizadores e actores, pode assumir os valores normal (**normal**) ou vedeta (**vedeta**) e não é lido para outros tipos de colaboradores.

Por exemplo, suponha que pretende criar vários colaboradores (um produtor sénior, um produtor júnior, um realizador, dois técnicos e um actor (que é uma vedeta), por esta ordem. Neste caso, tudo está correcto, sendo devolvidas as mensagens (Colaborador registado com sucesso!) no fim de cada registo.

```

> regista↵
senior 100 Max Bialystock↵
Colaborador registado com sucesso!↵
> regista↵
junior 50 Leo Bloom↵
Colaborador registado com sucesso!↵
> regista↵
realizador normal 40 Roger De Bris↵
Colaborador registado com sucesso!↵
> regista↵
tecnico 10 Carmen Ghia↵
Colaborador registado com sucesso!↵
> regista↵
tecnico 200 Mel Brooks↵
Colaborador registado com sucesso!↵
> regista↵
actor vedeta 80 Ulla Inga Hansen Benson Yansen Tallen Hallen Svaden Swanson↵
Colaborador registado com sucesso!↵

```

Caso algo esteja mal o colaborador não é criado. Mostra-se uma mensagem de erro adequada:

- (1) Caso já exista um colaborador com o mesmo nome, a mensagem de erro será (Ja existe um colaborador com o mesmo nome.).
- (2) Caso o tipo de colaborador não seja **senior**, **junior**, **realizador**, **tecnico**, ou **actor**, a mensagem de erro será (Tipo de colaborador desconhecido.).
- (3) Caso a qualificação de notoriedade não seja **vedeta** ou **normal**, a mensagem de erro será (Notoriedade invalida.).
- (4) Caso o valor horário seja < 0 , a mensagem de erro será (Acha mesmo que este colaborador vai pagar para trabalhar?)

Seguem-se alguns exemplos ilustrativos da sequência de uma sessão interactiva em que ocorrem vários **erros** no registo de colaboradores:

```

> regista↵
senior 100 Max Bialystock↵
Colaborador registado com sucesso!↵
> regista↵
normal 100 Max Bialystock↵
Ja existe um colaborador com o mesmo nome.↵
> regista↵
catering 524 Julia Childs↵
Tipo de colaborador desconhecido.↵
> regista↵
actor extraordinario 10 Nathan Lane↵
Notoriedade invalida.↵
> regista↵
actor normal -10 Franz Liebkind↵
Acha mesmo que este colaborador vai pagar para trabalhar?↵
> regista↵
coreografo -100 William Ivey Long↵
Tipo de colaborador desconhecido.↵

```

2.4 Comando **staff**

Lista os elementos do staff registados. Este comando não necessita de parâmetros e sucede sempre. Lista os colaboradores da empresa por ordem de criação dos mesmos na aplicação, um por linha, apresentando sucessivamente o seu **papel** (produtor, realizador, técnico ou actor), nome e custo por hora. Se não existirem colaboradores registados o programa deve devolver a mensagem (Nao existem colaboradores registados.).

```
> staff↵
produtor senior Max Bialystock 100↵
produtor junior Leo Bloom 50↵
realizador normal Roger De Bris 40↵
tecnico Carmen Ghia 10↵
tecnico Mel Brooks 200↵
actor vedeta Ulla Inga Hansen Benson Yansen Tallen Hallen Svaden Swanson 80↵
```

2.5 Comando **cenario**

Regista um novo cenário para gravações. O comando inclui dois parâmetros: o **local** e o **custo horário associado** ao local. Em caso de sucesso, a mensagem de retorno deve ser (Cenario registado.). No exemplo que se segue, o teatro é alugado por 100 euros por hora.

```
> cenario↵
Pantages Theatre↵
100↵
Cenario registado.↵
```

Podem ocorrer os seguintes erros:

- (1) Já existe um cenário com o mesmo nome registado. A mensagem de erro apropriada será, nesse caso, (Localizacao ja tinha sido registada.).
- (2) O custo horário indicado é inferior a 0 euros. A mensagem de erro apropriada será, nesse caso, (Acha que eles nos pagam para gravar la?).

```
> cenario↵
Pantages Theatre↵
100↵
Localizacao ja tinha sido registada.↵
> cenario↵
Dolby Theatre↵
-200↵
Acha que eles nos pagam para gravar la?↵
```

2.6 Comando **cenarios**

Lista os cenários disponíveis, com os respectivos custos horários. O comando não tem parâmetros e sucede sempre. No exemplo que se segue existem dois locais registados, que se apresentam por ordem de registo no sistema.

```
> cenarios↵
Patages Theatre 100.↵
Castelo de Sao Jorge 300.↵
```

Caso não existam locais de gravações registados, a mensagem adequada será (Nao existem localizacoes registadas.).

```
> cenarios↵
Nao existem localizacoes registadas.↵
```

2.7 Comando **marca**

Marca uma nova gravação. O formato geral do comando inclui como parâmetros o local (**local**), a data de início da gravação (**dataInicial**), a duração (**duracao**) em minutos, o produtor (**produtor**), o realizador (**realizador**), o técnico (**tecnico**), o número inteiro ≥ 0 de restantes colaboradores que vão participar (**nColaboradores**) e os **nomes de cada um dos colaboradores adicionais** (note que podem não existir colaboradores adicionais). Os tipos de colaborador conhecidos são: produtor sénior (senior), produtor júnior (junior), realizador (realizador), técnico (tecnico) e actor (actor).

Por exemplo, suponha que pretende gravar uma cena envolvendo dois produtores (Bloom e Bialystock), um realizador (De Bris), dois técnicos (Ghia e Brooks) e um actor (Swanson), no famoso local (Pantages Theatre) em Los Angeles, no dia 26 de Março, pelas 10h20, com duração prevista de 68 minutos. Correndo tudo bem, a mensagem de resposta deve ser (Gravacao agendada com sucesso!)

```
> marca↵
Pantages Theatre↵
2018 03 26 10 20 68↵
Leo Bloom↵
Roger De Bris↵
Carmen Ghia↵
3↵
Ulla Inga Hansen Benson Yansen Tallen Hallen Svaden Swanson↵
Mel Brooks↵
Max Bialystock↵
Gravacao agendada com sucesso!↵
```

Caso algo esteja mal a gravação não é agendada. Mostra-se uma mensagem de erro adequada:

- (1) Caso o local seja desconhecido, a mensagem de erro será (Local desconhecido.).
- (2) Caso a data seja inválida (anterior à data da última gravação realizada, se existir), a mensagem de erro será (Data de gravacao invalida.).
- (3) Caso a duração seja inválida (ou seja, ≤ 0) mensagem de erro será (Duracao invalida.).
- (4) Caso o produtor principal não exista, a mensagem será (Produtor desconhecido.).
- (5) Caso o realizador principal não exista, a mensagem será (Realizador desconhecido.).
- (6) Caso o técnico principal não exista, a mensagem será (Tecnico desconhecido.).

- (7) Caso algum outro colaborador não exista, a mensagem será (Colaborador desconhecido.).
- (8) Caso a gravação seja impossível de realizar por causa de (pelo menos) uma vedeta amuada com (pelo menos) um participante, a gravação é agendada, mas fica suspensa. A mensagem adequada, neste caso, é (Gravacao pendente de uma birra.).
- (9) Caso a gravação seja impossível de realizar por colisão de local ou pelo menos um dos participantes, a mensagem adequada será (Gravacao nao agendada por conflito de datas.). Isto deve ocorrer quando existe conflito e o produtor principal da gravação não tem prioridade sobre o produtor principal de pelo menos uma das gravações com as quais exista um conflito.
- (10) Caso a gravação obrigue ao reagendamento de outra gravação para remover algum conflito de calendário, a mensagem adequada será (Gravacao prioritaria agendada provocou mudancas noutra(s) gravacao(oes).). Isto deve ocorrer quando existir um conflito e o produtor principal tem prioridade sobre todos os outros produtores principais de todas as gravações com as quais esta entraria em conflito.

Seguem-se alguns exemplos ilustrativos da sequência de uma sessão interactiva em que ocorrem vários **erros** no registo de gravações (note que os exemplos são longe de ser exaustivos):

```
> marca↵
Pantages Theatre↵
2018 03 26 10 20 68↵
Leo Bloom↵
Roger De Bris↵
Carmen Ghia↵
3↵
Ulla Inga Hansen Benson Yansen Tallen Hallen Svaden Swanson↵
Mel Brooks↵
Max Bialystock↵
Gravacao agendada com sucesso!↵
> marca↵
Pantages Theatre↵
2018 03 26 10 00 120↵
Joe Pasquale↵
Russ Abbot↵
James Dreyfus↵
0↵
Gravacao nao agendada por conflito de datas.↵
> marca↵
Dolby Theatre↵
1918 03 26 10 00 120↵
Joe Pasquale↵
Russ Abbot↵
James Dreyfus↵
0↵
Data de gravacao invalida.↵
```

```

> marca↵
Dolby Theatre↵
2018 03 26 10 30 120↵
Leo Bloom↵
Russ Abbot↵
James Dreyfus↵
0↵
Gravacao nao agendada por conflito de datas.↵
> marca↵
Dolby Theatre↵
2018 03 26 10 30 120↵
Donald P. Bellisario↵
Russ Abbot↵
James Dreyfus↵
0↵
Produtor desconhecido.↵
> marca↵
Dolby Theatre↵
2018 03 26 10 30 120↵
Joe Pasquale↵
Russ Abbot↵
James Dreyfus↵
0↵
Gravacao nao agendada por conflito de datas.↵
> marca↵
Dolby Theatre↵
2018 03 26 10 30 120↵
Max Bialystoc↵
Guillermo del Toro↵
James Dreyfus↵
0↵
Realizador desconhecido.↵
> marca↵
Dolby Theatre↵
2018 03 26 10 30 120↵
Max Bialystoc↵
Russ Abbot↵
James Dreyfus↵
0↵
Gravacao nao agendada por conflito de datas.↵
> marca↵
Dolby Theatre↵
2018 03 26 10 30 120↵
Joe Pasquale↵
Steven Spielberg↵
Jose Mourinho↵
0↵
Tecnico desconhecido.↵

```

2.8 Comando **amua**

A vedeta deixa de trabalhar com um colaborador. O comando recebe dois parâmetros. O nome da **vedeta** e o nome do outro **colaborador**. O comando tem como efeito colocar

o colaborador na “*lista negra*” da **vedeta**, suspendendo todas as gravações em que ambos estejam simultaneamente envolvidos. A relação de ódio é unidireccional, ou seja, o facto de uma vedeta “amuar” com outro membro não implica reciprocidade. Em qualquer momento, a zanga pode terminar (ver comando **reconcilia**).

Por exemplo, suponha que o actor vedeta Ryan Gosling se aborreceu com a actriz Rachel McAdams e que eles têm 3 gravações em comum agendadas. Nesse caso, a mensagem de feedback deste comando apropriada seria (Ryan Gosling colocou Rachel McAdams na sua lista negra, suspendendo 3 gravacoes.). Suponha ainda que a actriz vedeta Sarah Jessica Parker se aborreceu com a sua co-estrela de O Sexo e a Cidade, Kim Cattrall. Como a série já foi cancelada há que tempos e ambas continuam chateadas, neste momento não têm gravações agendadas em comum. Nesta situação, a mensagem apropriada seria (Sarah Jessica Parker colocou Kim Cattrall na sua lista negra, suspendendo 0 gravacoes.).

```
> amua↵
Ryan Gosling↵
Rachel McAdams↵
Ryan Gosling colocou Rachel McAdams na sua lista negra, suspendendo 3 gravacoes.↵
> amua↵
Sarah Jessica Parker↵
Kim Cattrall↵
Sarah Jessica Parker colocou Kim Cattrall na sua lista negra, suspendendo 0 gravacoes.↵
```

Podem ocorrer vários erros com o comando **amua**:

- (1) Não existe nenhuma vedeta com o nome indicado. Suponha que o actor Sylvester Stallone não é considerado vedeta. Nesse caso, a mensagem de erro adequada seria (Sylvester Stallone nao e uma vedeta.)
- (2) Não existe nenhum colaborador com o nome indicado. No exemplo abaixo, suponha que o realizador George Lucas se zangou com Jar Jar Binks, mas que, apesar de George Lucas ser uma vedeta, Jar Jar Binks não é um colaborador. Nesse caso, a mensagem adequada seria (Jar Jar Binks nao e um colaborador.).
- (3) Uma vedeta zanga-se com alguém com quem já estava zangada. Por exemplo, Sarah Jessica Parker zanga-se, de novo, com Kim Cattrall. Nesse caso, a mensagem adequada seria (Que falta de paciencia para divas...).

```
> amua↵
Sylvester Stallone↵
Arnold Schwarzenegger↵
Sylvester Stallone nao e uma vedeta.↵
> amua↵
George Lucas↵
Jar Jar Binks↵
Jar Jar Binks nao e um colaborador.↵
> amua↵
Sarah Jessica Parker↵
Kim Cattrall↵
Que falta de paciencia para divas...↵
```

2.9 Comando **reconcilia**

A vedeta reconcilia-se com o colaborador, podendo voltar a trabalhar com esse colaborador. O comando recebe dois parâmetros. O nome da **vedeta** e o nome do outro **colaborador**. O comando tem como efeito remover o colaborador da “*lista negra*” da **vedeta**, retomando todas as gravações em que ambos estejam simultaneamente envolvidos, desde que não existam outras zangas pendentes, claro. Por exemplo, suponha que o actor Ryan Gosling faz as pazes com a actriz Rachel McAdams. Nesse caso, a mensagem adequada seria (Ryan Gosling <3 Rachel McAdams. 3 gravacoes salvas!). Suponha que Sarah Jessica Parker se reconciliava com Kim Cattrall. A mensagem adequada seria (Sarah Jessica Parker <3 Kim Cattrall. 0 gravacoes salvas!).

```
> reconcilia↵
Ryan Gosling↵
Rachel McAdams↵
Ryan Gosling <3 Rachel McAdams. 3 gravacoes salvas!↵
> reconcilia↵
Sarah Jessica Parker↵
Kim Cattrall↵
Sarah Jessica Parker <3 Kim Cattrall. 0 gravacoes salvas!↵
```

Podem ocorrer vários erros com o comando **reconcilia**:

- (1) Não existe nenhuma vedeta com o nome indicado. Suponha que o actor Sylvester Stallone não é considerado vedeta. Nesse caso, a mensagem de erro adequada seria (Sylvester Stallone nao e uma vedeta.)
- (2) Não existe nenhuma zanga com o colaborador indicado, ou esse colaborador nem sequer existe. No exemplo abaixo, suponha que o realizador George Lucas se tenta reconciliar com Jar Jar Binks, mas que, apesar de George Lucas ser uma vedeta, Jar Jar Binks não é um colaborador, ou George Lucas não está zangado com ele. Nesse caso, a mensagem adequada seria (Nao existe zanga com Jar Jar Binks.).

```
> reconcilia↵
Sylvester Stallone↵
Arnold Schwarzenegger↵
Sylvester Stallone nao e uma vedeta.↵
> reconcilia↵
George Lucas↵
Jar Jar Binks↵
Nao existe zanga com Jar Jar Binks.↵
```

2.10 Comando **realizadas**

Lista as gravações realizadas. As gravações que já foram realizadas devem ser listadas por ordem crescente de data e hora de início. Em caso de empate, lista primeiro a gravação de menor duração. Se o empate subsistir, usa a ordem de inserção da gravação no sistema. Para cada gravação, apresenta a data de gravação, o local, o produtor, e o realizador. No final da lista, apresenta o custo total das gravações já realizadas. No exemplo que se segue, realizaram-se 3 gravações, com um custo total de 1302378 euros (note que a segunda e terceira gravações ocorreram no mesmo dia, mas a terceira começou mais tarde):

```
> realizadas↵
2018 1 26; 20th Century Fox; Lauren Shuler Donner; David Leitch.↵
2018 2 1; Castle Ward; David Benioff; Miguel Sapochnik.↵
2018 2 1; San Juan de Gaztelugatxe; Dan Weiss; David Nutter.↵
1302378 euros gastos.↵
```

Se nenhuma gravação foi realizada, a mensagem apropriada será (Nenhuma gravacao realizada.).

```
> realizadas↵
Nenhuma gravacao realizada.↵
```

2.11 Comando **previstas**

Lista as gravações previstas. As gravações previstas devem ser listadas por ordem crescente de data e hora de início. Em caso de empate, lista primeiro a gravação de menor duração. Se o empate subsistir, usa a ordem de inserção da gravação no sistema. Para cada gravação, apresenta a data de gravação, o local, o produtor e o realizador. No final da lista, apresenta o custo total das gravações previstas. Caso alguma gravação prevista esteja suspensa, essa gravação deve ainda assim ser listada, com uma anotação apropriada. No exemplo que se segue há 4 gravações agendadas, a segunda das quais suspensa, por zanga de uma vedeta.

```
> previstas↵
2018 4 26; 20th Century Fox; Lauren Shuler Donner; David Leitch.↵
2018 4 28; Carrie Bradshaw flat; Darren Star; Michael Patrick King. Suspensa!↵
2018 5 1; Castle Ward; David Benioff; Miguel Sapochnik.↵
2018 5 1; San Juan de Gaztelugatxe; Dan Weiss; David Nutter.↵
1202568 euros orcamentados.↵
```

Se nenhuma gravação está prevista, a mensagem apropriada será (Nenhuma gravacao prevista.).

```
> previstas↵
Nenhuma gravacao prevista.↵
```

2.12 Comando **local**

Lista as gravações previstas para este local. As gravações previstas devem ser listadas por ordem crescente de data e hora de início. Para cada gravação, apresenta a data de gravação, o produtor e o realizador. No final da lista, apresenta o custo total das gravações previstas.

```
> local↵
20th Century Fox Studios↵
2018 4 26; Lauren Shuler Donner; David Leitch.↵
2018 4 28; Darren Star; Michael Patrick King. Suspensa!↵
802000 euros orcamentados.↵
```

Se nenhuma gravação está prevista, a mensagem apropriada será (Nenhuma gravacao prevista em **Nome do local**.).

```
> local↵  
Metro-Goldwyn-Mayer cartoon studio↵  
Nenhuma gravacao prevista em Metro-Goldwyn-Mayer cartoon studio.↵
```

Pode ocorrer um erro no comando `local`:

- (1) Se o local passado como argumento não existir, a mensagem apropriada seria (Local desconhecido.).

```
> local↵  
Atlantis↵  
Local desconhecido.↵
```

2.13 Comando `colaborador`

Lista as gravações previstas para um colaborador. As gravações previstas devem ser listadas por ordem crescente de data e hora de início. Em caso de empate, lista primeiro a gravação de menor duração. Se o empate subsistir, usa a ordem de inserção da gravação no sistema. Para cada gravação, apresenta a data de gravação, o produtor e o realizador. No final da lista, apresenta o custo total das gravações previstas.

```
> colaborador↵  
Ryan Gosling↵  
2018 4 26; Lauren Shuler Donner; David Leitch.↵  
2018 4 28; Darren Star; Michael Patrick King. Suspensa!↵  
802000 euros orcamentados.↵
```

Se nenhuma gravação está prevista com determinado colaborador, a mensagem apropriada será (Nenhuma gravacao prevista com `Nome do colaborador`.).

```
> colaborador↵  
Harvey Weinstein↵  
Nenhuma gravacao prevista com Harvey Weinstein.↵
```

Pode ocorrer um erro no comando `colaborador`:

- (1) Se o colaborador passado como argumento não existir, a mensagem apropriada é (Colaborador desconhecido.).

```
> colaborador↵  
Nicholas Cage↵  
Colaborador desconhecido.↵
```

2.14 Comando `grava`

Executa a próxima gravação agendada. Passa a primeira gravação agendada da lista de gravações previstas para a lista das gravações realizadas. A gravação deixa de estar prevista, para estar realizada. Se por acaso a primeira gravação estiver suspensa, essa gravação é removida da lista de gravações previstas e passa para a das realizadas, sendo apresentada com a indicação

de cancelada. Se não existir nenhuma gravação planeada, a mensagem de feedback deve ser (Nenhuma gravacao agendada.). No exemplo que se segue, tentam-se realizar duas gravações. A primeira é Gravada!. A segunda é Cancelada!. No final, apenas duas gravações continuam previstas.

```
> previstas↵
2018 4 26; 20th Century Fox; Lauren Shuler Donner; David Leitch.↵
2018 4 28; Carrie Bradshaw flat; Darren Star; Michael Patrick King. Suspensa!↵
2018 5 1; Castle Ward; David Benioff; Miguel Sapochnik.↵
2018 5 1; San Juan de Gaztelugatxe; Dan Weiss; David Nutter.↵
1202568 euros orcamentados.↵
> grava↵
2018 4 26; 20th Century Fox; Lauren Shuler Donner; David Leitch. Gravada!↵
> grava↵
2018 4 28; Carrie Bradshaw flat; Darren Star; Michael Patrick King. Cancelada!↵
> previstas↵
2018 5 1; Castle Ward; David Benioff; Miguel Sapochnik.↵
2018 5 1; San Juan de Gaztelugatxe; Dan Weiss; David Nutter.↵
80000 euros orcamentados.↵
> grava↵
2018 5 1; Castle Ward; David Benioff; Miguel Sapochnik. Gravada!↵
> grava↵
2018 5 1; San Juan de Gaztelugatxe; Dan Weiss; David Nutter. Gravada!↵
> grava↵
Nenhuma gravacao agendada.↵
```

2.15 Comando **amuancos**

Lista os colaboradores com quem uma vedeta está amuada. Lista, por ordem de registo do amuango, a lista de colaboradores com quem uma vedeta está amuada, uma por linha. Caso a vedeta não tenha amuancos, deve ser apresentada uma mensagem apropriada. Por exemplo, se o actor Anthony Hopkins não tiver zangas, a mensagem será (Anthony Hopkins da-se bem com todos!).

```
> amuancos↵
Kenny Baker↵
Anthony Daniels↵
> amuancos↵
Terry Hatcher↵
Eva Longoria↵
Marcia Cross↵
Felicity Huffman↵
Erica Rogella↵
Steve Bartek↵
Tom Spezialy↵
> amuancos↵
Anthony Hopkins↵
Anthony Hopkins da-se bem com todos!↵
```

Este comando pode originar uma mensagem de erro se:

- (1) O nome passado como argumento não for de uma vedeta (ou nem sequer for colaborador). Nesse caso, a mensagem de erro adequada é, supondo que o nome passado como argumento foi Steven Seagal (Mas quem é Steven Seagal?)

```

> amuancos↵
Steven Seagal↵
Mas quem e Steven Seagal?↵
> amuancos↵
David Hasselhoff↵
Mas quem e David Hasselhoff?↵

```

3 Exemplo completo

Nesta secção apresentamos um exemplo de uma sessão completa. Note que este exemplo não pretende ser exaustivo de tudo o que pode acontecer, mas apenas ilustrar uma sessão típica.

```

> ajuda↵
registra - regista um novo colaborador↵
staff - lista os colaboradores registados↵
cenario - regista um novo local para gravacoes↵
cenarios - lista os locais para gravacoes registados↵
marca - marca uma nova gravacao↵
amua - vedeta deixa de trabalhar com colaborador↵
reconcilia - vedeta faz as pazes com colaborador↵
realizadas - lista as gravacoes realizadas↵
previstas - lista as gravacoes previstas↵
local - lista as gravacoes previstas para um local↵
colaborador - lista as gravacoes previstas para um colaborador↵
grava - executa a proxima gravacao agendada↵
amuancos - lista os colaboradores com quem uma vedeta esta amuada↵
ajuda - Mostra a ajuda↵
sai - Termina a execucao do programa↵
> staff↵
Nao existem colaboradores registados.↵
> regista↵
senior 100 Max Bialystock↵
Colaborador registado com sucesso!↵
> regista↵
junior 50 Leo Bloom↵
Colaborador registado com sucesso!↵
> regista↵
realizador normal 40 Roger De Bris↵
Colaborador registado com sucesso!↵
> regista↵
tecnico 10 Carmen Ghia↵
Colaborador registado com sucesso!↵
> regista↵
tecnico 200 Mel Brooks↵
Colaborador registado com sucesso!↵
> regista↵
actor vedeta 80 Ulla Inga Hansen Benson Yansen Tallen Hallen Svaden
Swanson↵
Colaborador registado com sucesso!↵

```

```

> staff↵
produtor senior Max Bialystock 100↵
produtor junior Leo Bloom 50↵
realizador normal Roger De Bris 40↵
tecnico Carmen Ghia 10↵
tecnico Mel Brooks 200↵
actor vedeta Ulla Inga Hansen Benson Yansen Tallen Hallen Svaden Swanson 80↵
> cenario↵
Pantages Theatre↵
100↵
Cenario registado.↵
> cenario↵
Dolby Theatre↵
100↵
Cenario registado.
> previstas↵
Nenhuma gravacao prevista.↵
> marca↵
Pantages Theatre↵
2018 03 26 10 20 68↵
Leo Bloom↵
Roger De Bris↵
Carmen Ghia↵
3↵
Ulla Inga Hansen Benson Yansen Tallen Hallen Svaden Swanson↵
Mel Brooks↵
Max Bialystock↵
Gravacao agendada com sucesso!↵
> marca↵
Dolby Theatre Lounge↵
2018 03 27 10 20 30↵
Max Bialystock↵
Roger De Bris↵
Carmen Ghia↵
0↵
Gravacao agendada com sucesso!↵
> realizadas↵
Nenhuma gravacao realizada.↵
> grava↵
2018 3 26; Pantages Theatre; Leo Bloom; Roger De Bris. Gravada!↵
> realizadas↵
2018 3 26; Pantages Theatre; Leo Bloom; Roger De Bris.↵
1160 euros gastos.↵
> sai↵
Ate a proxima↵

```

4 Desenvolvimento do Projeto

A sua aplicação deve tirar o melhor partido possível da matéria leccionada. Em particular, fazê-la **o mais extensível possível**. Deve ser construída de modo a **minimizar as alterações necessárias**, caso se pretendam acrescentar, mais tarde, **novos tipos de participantes**, ou de **recursos associados a gravações**).

Comece por desenvolver a interface principal da sua aplicação, identificando claramente

quais os comandos que a sua aplicação deve suportar, bem como quais as entradas e saídas, não esquecendo as pré-condições. Depois, identifique as entidades de que vai necessitar para implementar este sistema. Identifique e especifique cuidadosamente as **interfaces** e **classes** de que necessita. Deve documentar a sua concepção e o seu desenvolvimento quer através de um diagrama de classes e interfaces, quer através de documentação adequada no seu código.

Construa o esqueleto da classe `Main` que trata da entrada e saída dos dados e da interacção com a aplicação. É normal que no princípio o seu programa ainda não faça tudo. Lembre-se da **regra da versão estável**: não tente fazer tudo de uma só vez. Vá fazendo, testando, e avançando por incrementos, à medida que as funcionalidades vão sendo desenvolvidas. Se necessário, crie pequenos programas de teste auxiliares. Desenvolva também as operações de forma incremental.

As gravações a agendar decorrem num determinado momento no tempo. Para representar a data e hora de início de uma gravação use a classe `LocalDateTime`, das bibliotecas do Java. Pode encontrar mais informações sobre esta classe em <https://docs.oracle.com/javase/8/docs/api/java/time/LocalDateTime.html>. Em particular, sugere-se que consulte, em detalhe, o método de classe `public static LocalDateTime of(int year, int month, int dayOfMonth, int hour, int minute)` para gerar datas com esta classe. Necessitará de mais alguns métodos desta classe, mas comece por estudar este.

5 Submissão ao Mooshak

Para submeter o seu programa ao Mooshak registe-se no concurso POO1718-TP1 e siga as instruções apresentadas na área Moodle da disciplina.

5.1 Sintaxe dos comandos

Note que para cada comando será apresentada apenas uma única mensagem de saída. Ou seja, as condições que causam falhas num comando devem ser verificadas pela ordem descrita no enunciado e assim que uma dessas condições se verificar não é necessário verificar as condições seguintes. Por exemplo, se ao executar o comando `register` já existir um colaborador com o nome dado, deve ignorar o resto das condições de validação, devolvendo a mensagem de erro apropriada ao primeiro erro detectado, seguindo a ordem pela qual os vários tipos de erros de input estão elencados, nesta especificação.

5.2 Testes

Os testes do Mooshak verificam de forma incremental a implementação dos vários comandos e serão publicados a 6 de Abril.

Ficheiro de teste 01_in.txt (2 pontos)

Comandos testados: `ajuda` e `sai`

Testa os comandos `ajuda` e `sai`.

Ficheiro de teste 02_in.txt (10 pontos)

Comandos testados: `registra`, `staff`, `sai`

Testa os comandos `registra` e `staff` sem situações de erro.

Ficheiro de teste 03_in.txt (4 pontos)

Comandos testados: `registra`, `staff`, `sai`

Testa os comandos `registra` e `staff` com situações de erro.

Ficheiro de teste 04_in.txt (10 pontos)

Comandos testados: **cenario**, **cenarios**, sai

Testa os comandos **cenario** e **cenarios** sem situações de erro.

Ficheiro de teste 05_in.txt (4 pontos)

Comandos testados: **cenario**, **cenarios**, sai

Testa os comandos **cenario** e **cenarios** com situações de erro.

Ficheiro de teste 06_in.txt (10 pontos)

Comandos testados: **marca**, **previstas**, regista, staff, **cenario**, **cenarios**, sai

Testa os comandos **marca** e **previstas**, sem situações de erro, num agendamento simples, sem colisões de horários, birras, etc., com as marcações a ser feitas por ordem cronológica correcta.

Ficheiro de teste 07_in.txt (10 pontos)

Comandos testados: **grava**, **realizadas**, **marca**, **previstas**, regista, staff, **cenario**, **cenarios**, sai

Testa os comandos **grava** e **realizadas**, sem situações de erro, num agendamento simples, sem colisões de horários, birras, etc., com as marcações a ser feitas por ordem cronológica correcta, em condições semelhantes às do teste 06_in.txt, mas agora com progressão do tempo.

Ficheiro de teste 08_in.txt (5 pontos)

Comandos testados: **local**, **grava**, **realizadas**, **marca**, **previstas**, regista, staff, **cenario**, **cenarios**, sai

Testa o comando **local**, sem situações de erro, num agendamento simples, sem colisões de horários, birras, etc., com as marcações a ser feitas por ordem cronológica correcta, em condições semelhantes às do teste 07_in.txt.

Ficheiro de teste 09_in.txt (2 pontos)

Comandos testados: **local**, **grava**, **realizadas**, **marca**, **previstas**, regista, staff, **cenario**, **cenarios**, sai

Testa o comando **local**, com situações de erro exclusivamente no comando **local**, num agendamento simples, sem colisões de horários, birras, etc., com as marcações a ser feitas por ordem cronológica correcta, em condições semelhantes às do teste 08_in.txt.

Ficheiro de teste 10_in.txt (5 pontos)

Comandos testados: **colaborador**, **grava**, **realizadas**, **marca**, **previstas**, regista, staff, **cenario**, **cenarios**, sai

Testa o comando **colaborador**, sem situações de erro, num agendamento simples, sem colisões de horários, birras, etc., com as marcações a ser feitas por ordem cronológica correcta, em condições semelhantes às do teste 07_in.txt.

Ficheiro de teste 11_in.txt (2 pontos)

Comandos testados: **colaborador**, **grava**, **realizadas**, **marca**, **previstas**, regista, staff, **cenario**, **cenarios**, sai

Testa o comando **colaborador**, com situações de erro exclusivamente no comando **colaborador**, num agendamento simples, sem colisões de horários, birras, etc., com as marcações a ser feitas por ordem cronológica correcta, em condições semelhantes às do teste 10_in.txt.

Ficheiro de teste 12_in.txt (5 pontos)

Comandos testados: `amua`, `amuancos`, `colaborador`, `grava`, `realizadas`, `marca`, `previstas`, `registra`, `staff`, `cenario`, `cenarios`, `sai`

Testa o comando `amua` e `amuancos`, sem situações de erro, num agendamento simples, sem colisões de horários, com as marcações a ser feitas por ordem cronológica correcta.

Ficheiro de teste 13_in.txt (2 pontos)

Comandos testados: `amua`, `amuancos`, `colaborador`, `grava`, `realizadas`, `marca`, `previstas`, `registra`, `staff`, `cenario`, `cenarios`, `sai`

Testa o comando `reconcilia`, com situações de erro exclusivamente no comando `reconcilia`, num agendamento simples, sem colisões de horários, com as marcações a ser feitas por ordem cronológica correcta, em condições semelhantes às do teste 12_in.txt.

Ficheiro de teste 14_in.txt (5 pontos)

Comandos testados: `reconcilia`, `amua`, `amuancos`, `colaborador`, `grava`, `realizadas`, `marca`, `previstas`, `registra`, `staff`, `cenario`, `cenarios`, `sai`

Testa o comando `reconcilia`, sem situações de erro num agendamento simples, sem colisões de horários, com as marcações a ser feitas por ordem cronológica correcta.

Ficheiro de teste 15_in.txt (2 pontos)

Comandos testados: `reconcilia`, `amua`, `amuancos`, `colaborador`, `grava`, `realizadas`, `marca`, `previstas`, `registra`, `staff`, `cenario`, `cenarios`, `sai`

Testa o comando `reconcilia`, com situações de erro exclusivamente no comando `reconcilia`, num agendamento simples, sem colisões de horários, com as marcações a ser feitas por ordem cronológica correcta, em condições semelhantes às do teste 14_in.txt.

Ficheiro de teste 16_in.txt (5 pontos)

Comandos testados: `marca`, `previstas`, `registra`, `staff`, `cenario`, `cenarios`, `sai`

Testa os comandos, sem situações de erro, num agendamento simples, sem colisões de horários, com as com as marcações a ser feitas fora da ordem cronológica correcta.

Ficheiro de teste 17_in.txt (5 pontos)

Comandos testados: `marca`, `previstas`, `registra`, `staff`, `cenario`, `cenarios`, `sai`

Testa os comandos, em situações de erro num agendamento simples, sem colisões de horários, com as marcações a ser feitas fora da ordem cronológica correcta, em condições semelhantes às do teste 16_in.txt.

Ficheiro de teste 18_in.txt (6 pontos)

Comandos testados: `marca`, `previstas`, `registra`, `staff`, `cenario`, `cenarios`, `sai`

Testa os comandos num agendamento com colisões de horários que não provocam reagendamentos.

Ficheiro de teste 19_in.txt (6 pontos)

Testa todos os comandos, inclusive agendamentos com colisões de horários que provocam reagendamentos.