

Projecto de Software "Cliente/Servidor"
Partilha de Livros
{BookShare.doc}

I - Introdução genérica

Uma Instituição de Partilha de Livros (de estudo do Ensino Básico/Secundário, por ex.) encontra-se *distribuída geograficamente por vários Balcões*; a cada um deles está associado um Agente *Servidor* com o registo dos movimentos dos Livros que aí vão sendo depositados.

Cada Balcão conhece os respectivos *vizinhos geográficos*, e os "pesos" das respectivas ligações. Eis um exemplo:

Balcão	Balcão vizinho / Peso da ligação			
A	C	1	D	2
B	C	1	D	2
C	A	1	B	1
D	A	2	B	2
E	A	2	B	1

onde é claro que do *balcão* A se pode chegar aos vizinhos C, D e E por rotas com os pesos 1, 2 e 2.

Os Balcões encontram-se distribuídos geograficamente: por ex., A em Lisboa, B no Porto, C em Coimbra, etc.. Para cada fulano que pretenda aceder ao "Sistema", existe um assim denominado Balcão de *Preferência*: se, por ex., reside em Coimbra, a escolha natural será C...

O "Peso" de uma ligação há-de entender-se como uma medida do custo do *transporte físico* (por ex., por correio-postal) de um Livro de um para outro Balcão.

Um eventual interessado acede através de um *Cliente*, para doar, obter ou devolver um Livro. O diálogo entre ele e os *Servidores* é feito através da Internet. Como introdução à compreensão do Sistema, o texto continua com um esboço global do seu funcionamento básico; após o que se detalham aspectos de pormenor.

I-1. Doação de um Título: Qualquer fulano que disponha de um Livro de que já não necessite (e que queira fazer uma boa acção) pode cedê-lo a um desses Balcões. Essa entrega é sinalizada em um computador, mediante o comando genérico

Give Título

(subentende-se: o Livro é entregue ao Balcão de *Preferência* do fulano).

Nada impede que um mesmo Título seja doado, por vários fulanos, no mesmo ou em Balcões diferentes. A informação de Títulos disponíveis num Servidor, seja *Titles.dat*, pode então ser:

M3 (Matemática 3º ano)	4
F7 (Física 7º ano)	1
P1 (Português 1º ano)	10
I3 (Inglês 3º ano)	6

onde é claro que, no Balcão associado a esse Servidor, estão disponíveis 4 exemplares do livro de Matemática do 3º ano...

I-2. Procura e Reserva de um Título: Qualquer fulano interessado em obter um Livro pode investigar através do computador acerca da sua disponibilidade, mediante o comando genérico

Search Título

O Título será procurado primeiramente no Balcão *de Preferência*... mas, se ele não estiver aí disponível, será procurado em outro(s) Balcão(ões). Se for encontrado em um (ou mais Balcões), será devolvida a identificação do Balcão onde ficará *reservado*. Tendencialmente, será aquele que se volve em um *PesoTotal menor* - mas nem sempre, como adiante se explica.

Afora Alertas de Erro, a resposta no ecran pode ser uma das seguintes:

- a) "Título indisponível";
- b) "Título reservado em ..., com $\text{Peso}=\text{PesoTotal}$ ".

I-3. Cancelamento ou Confirmação da Reserva de um Exemplar: No caso b), o fulano pode:

- c) *desistir* da reserva, com o comando genérico:

Cancel Id

- d) *confirmar* a reserva, com o comando genérico:

Take Id

Id identifica o Exemplar que ficou reservado com **Search**; na prática, volve-se na identificação do *Balcão onde ficou reservado*, seguida do respectivo Título, por ex.: A-P1.

I-4. Devolução de um Exemplar: É irrelevante a forma como o Livro chega às mãos do requerente: por ex., o Balcão com a Reserva pode enviá-lo por correio postal... Quando ele já não necessitar do Livro, deverá devolvê-lo; na ocasião, esse movimento será sinalizado ao computador com o comando genérico:

Release Id

(subentende-se: o Livro passará a estar disponível no Balcão *de Preferência* do fulano).

*Na vida "real", os **Give** e **Release** deveriam ser verificados, i.e., somente deveriam dar-se por completados após se conferir que os correspondentes Livros já deram entrada nos Balcões em causa (a entrega ao Balcão pode, em ambos os casos, ser feita por correio-postal... e alguém poderia invocar aqueles comandos sem, de facto, chegar qualquer Livro aos Balcões...). Para não complicar desnecessariamente, no Projecto subentende-se: não há batota... Consideração análoga vale para o comando **Take**...*

II - Detalhes da Especificação

Servidores e Utilizadores identificam-se, respectivamente, por pares *Host+Port* e *User+Password* (No mesmo computador, poderão coexistir um ou mais *User* diferentes e/ou um ou mais *Servers*; o mesmo *User* poderá estar acedendo aos *Servers* por dois ou mais computadores).

Um Servidor é lançado mediante um programa, seja "*Balcony port Titles.dat Users.dat*", em que:

- *Titles.dat* remete para um Ficheiro de Existências listando, por cada Título *localmente* disponível, o respectivo *Título* e o *Número de exemplares disponíveis* no respectivo Balcão;
- *Users.dat* remete para um Ficheiro de Configuração listando os pares *User+Password* de Utilizadores registados, por ex.,

U1	P1
U2	P2
U3	P3

Users.dat mantém-se o mesmo para todos os envolvidos; a forma como isso é conseguido não cabe no âmbito deste Projecto.

*Nota: A validação do par *User+Password* é feita (pelo Servidor do Balcão de Preferência) apenas na operação *Search*. Na operação *Give*, nem sequer é feita a validação do *User*.*

Se não fôr possível completar um comando porque o User e/ou Password não são aceites, deve reportar-se isso no ecran.

Um eventual interessado em doar, requerer ou devolver um Livro deve lançar um programa *Cliente*, seja "*BookShare User Password MyBalcony*"; o argumento *MyBalcony* designa um Ficheiro identificando o par *Host+Port* do Servidor do Balcão de Preferência; eis *cinco* exs. de *MyBalcony*:

A: mega.ist.utl.pt 20000
B: mega.ist.utl.pt 20001
C: sado.ist.utl.pt 20000
D: douro.ist.utl.pt 20000
E: douro.ist.utl.pt 20001

Subsequentemente, **BookShare** aceita os seguintes comandos: **Give**, **Search**, **Cancel**, **Take**, **Release** e **Quit**. Com **Give**, sinaliza-se a doação de um Livro. Com **Search**, requiere-se a procura de um Título e a sua Reserva *automática*. Com **Cancel**, requiere-se o cancelamento da Reserva. Com **Take**, confirma-se a Reserva. Com **Release**, sinaliza-se a devolução de um Livro emprestado. Com **Quit**, requiere-se a terminação *normal* de *BookShare*.

II-1 - Argumentos múltiplos:

Give e **Search**, aceitam, ambos, um ou mais Títulos, por ex.:

Give M3
Give P1 F7
Search I3 P1 M3

(se um Título aparecer repetido na invocação de **Search**, é considerado somente uma vez).

Por sua vez, **Take**, **Cancel** e **Release** aceitam um ou mais Ids. Admitindo, por ex., que se invocou **Search** I3 P1 M3, a que foi respondido:

"I3 reservado em A, com Peso=1".
"P1 reservado em A, com Peso=1".
"M3 reservado em A, com Peso=1".

o requerente pode invocar:

Cancel A-P1
Take A-I3 A-M3

e, no fim, invocar:

Release A-I3 A-M3

Repare-se:

- todos os Ids são prefixados pela identificação do Balcão onde se fez a reserva (no caso, A);
- o *Peso* é o *PesoTotal* da rota do Balcão de preferência àquele com a reserva (no caso, C a A);
- *Cancel*, a ser invocado, é-o (preferentemente) antes de *Take*

II-2 - Compactação de Argumentos

Na execução de um comando (**Give**, **Search**, **Cancel**, **Take** ou **Release**), pode ter-se que contactar um Servidor sinalizando-lhe *vários* Títulos ou Exemplares (identificados pelos respectivos

Ids). Por mor de *eficiência*, não se lhe deve enviar mais que uma Mensagem, i.e.: *todos os Títulos ou Ids que respeitem a um mesmo destino devem ser assinalados na mesma Mensagem*. Por ex., e subentendendo-se que o Balcão de Preferência é C:

- se se invocar **Give** M3 P1, deve enviar-se *uma* Mensagem a C (referindo M3 e P1);
- se se invocar **Search** I3 P1 M3, deve enviar-se *uma* Mensagem a C (referindo I3, P1 e M3);
- se se invocar **Cancel** A-I3 B-F7 A-M3, deve enviar-se *uma* Mensagem a A (referindo I3 e M3), e *uma* outra a B (referindo F7); idem se se invocar **Take** A-I3 B-F7 A-M3;
- se se invocar **Release** A-I3 B-F7 A-M3, deve enviar-se *uma* Mensagem a A (referindo I3 e M3), *uma* a B (referindo F7) e *outra* a C (referindo I3, F7 e M3);

II-3 - Timeouts

Vários comandos levam ao armar de *timeouts* (relógios), como seja $T_{SearchMax}$, $T_{ReservMax}$, $T_{WatchMax}$, $T_{TakeMax}$, $T_{ReleaseMax}$. Se vierem a *expirar no Cliente*, este deve apresentar no ecran o pertinente Alerta; eis exs.:

"Título indisponível";
"Falha em Take/Release";

II-4 - Search

Admita-se que o Utilizador pretende os Títulos I3, P1 e M3, e que o seu Balcão de preferência é C. Lançado o programa *BookShare*, ele deverá invocar:

Search I3 P1 M3

A evolução da comunicação é como segue:

1. O Cliente envia ao *Servidor* associado ao Balcão de *Preferência*, seja C, uma *query* significando "Dispõe de I3, P1 e M3?". E arma um *timeout* $T_{SearchMax}$.

Repare-se: C pode estar *down*="em baixo"; ou pode acontecer que a *query* se extravie na rede ou se corrompa; ou que a sua réplica se extravie... Por estas e outras razões, convém estabelecer um período máximo de espera do Cliente, $T_{SearchMax}$, pela resposta à *query* feita...

2. Na recepção da *query*, C analisa a *sua* Base de Dados; e, para cada Título mencionado (no caso, I3, P1 e M3),

- se encontrar um exemplar desse Título disponível (ainda não reservado), marca-o como reservado, determina o correspondente Peso, arma um *timeout* $T_{ReservMax}$ e devolve ao Cliente uma Mensagem "*de disponibilidade*"

"Título - C - PesoTotal".

- se o não encontrar, devolve uma Mensagem "*de vizinhança*"

"Título - 0 - PesoAcumulado - Vizinhos"

em que Vizinhos é uma lista dos Balcões vizinhos de C (e respectivos Pesos de ligações mútuas). Para bom entendedor, o Servidor está "dizendo" ao Cliente: *Não tenho esse Título disponível, queira tentar nos Servidores que conheço, pode ser que aí tenha mais sorte...*

Por ex., se, aquando da invocação de **Search I3 P1 M3**, o conteúdo de C for

M3	4
F7	1
P1	10

então C devolverá as Mensagens:

"I3 - 0 - 0 - A/1 , B/1 , D/1"

"P1 - C - 0".

"M3 - C - 0".

E o conteúdo de C fica sendo:

M3	3	1
F7	1	0
P1	9	1

onde é claro que existe um exemplar de M3 reservado...

No arranque, um Servidor não conhece os pares *Host+Port* de *todos* os Servidores: conhece *apenas* os dos *seus* vizinhos (por ex., A conhece apenas os de C, D e E). Pelo que as Mensagens de Vizinhança deverão mesmo transportar os pares *Host+Port* dos vizinhos...

Repare-se: C "adivinha" que o Cliente poderá também interrogar outros Servidores... e conceder-lhe até $T_{\text{ReservMax}}$ para se decidir. Durante este intervalo de tempo, as réplicas a *queries* que venha a receber (do mesmo ou outro *User*) *não entram em linha de conta* com tais exemplares "reservados". Se, porém, se esgotar esse tempo (sem receber daquele Cliente uma Mensagem requerendo confirmação da Reserva desses exemplares), ou se receber daquele Cliente uma Mensagem solicitando a libertação dos "reservados", eles passam a ser entendidos como "disponíveis": podem ser considerados em subseqüentes *queries*.

3. O Cliente analisa as respostas recebidas de C, à medida que vão chegando.

Se a Mensagem refere um Balcão não nulo, como seja "P1 - C - 0", o Cliente apresenta no ecrã:
"P1 reservado em C, com Peso=0".

Mas, se o Balcão mencionado na Mensagem for 0 (como é o caso de "I3 - 0 - 0 - A/1,B/1,D/1"), então, *directamente* para cada Vizinho nela referido (no caso, A, B e D), o Cliente envia uma *query* relativa ao Título em causa (no caso, significando "Dispõe de I3?"). E arma um *timeout* T_{WatchMax} .

Repare-se:

- as *queries* são enviadas *por difusão* - a A, B e D - "ao mesmo tempo"!
- os Servidores podem estar todos *down*; ou pode acontecer que a *query* se extravie na rede ou se corrompa; ou o seu *reply* se extravie... Por estas e outras razões, convém estabelecer um período máximo de espera do Cliente, T_{WatchMax} , pelas respostas à *query* feita...

4. Na recepção da *query*, cada Servidor interrogado (no caso, os vizinhos de C) analisa a *sua* Base de Dados; e, para o Título mencionado (no caso, I3), procede de modo análogo a C:

- se encontrar um exemplar desse Título disponível (ainda não reservado), marca-o como reservado, determina o correspondente Peso, arma um *timeout* $T_{\text{ReservMax}}$ e devolve *directamente* ao Cliente uma Mensagem

"Título - Balcão - PesoTotal".

- se o não encontrar, devolve uma Mensagem

"Título - 0 - PesoAcumulado - Vizinhos"

em que *Vizinhos* é uma lista dos Balcões seus vizinhos (e respectivos Pesos de ligações mútuas).

Por ex., se apenas B contiver um exemplar disponível de I3, então as Mensagens devolvidas por A, B e D serão, respectivamente:

"I3 - 0 - 1 - C/1 , D/2 , E/2"

"I3 - B - 1".

"I3 - 0 - 1 - A/2 , B/2 , C/1 , E/1"

Repare-se:

- o *Peso Total da reserva* é o *Peso total da rota do Balcão de preferência àquele com a reserva* (no caso, C a B);
- o *Peso Acumulado de I3 via-A* é o *Peso total da rota do Balcão de preferência* (no caso, C) a A.

5. O Cliente analisa as respostas recebidas dos Servidores (no caso, A, B e D), à medida que vão chegando. Quando não faltar receber mais nenhuma, cancela T_{WatchMax} .

Se T_{WatchMax} vier a ser cancelado, ou se expirar entretanto, o Cliente faz o seguinte:

5a. Para cada Título acerca do qual recebeu *uma* e só uma Mensagem com Balcão não-nulo, como seja "I3 - B - 1", apresenta no ecran o pertinente Alerta "*de disponibilidade*", por ex.,
"I3 reservado em B, com Peso=1".

5b. Admita-se, entretanto, que a Mensagem emitida por B se extravia, isto é, chegam ao Utilizador apenas as Mensagens

"I3 - 0 - 1 - C/1 , D/2 , E/2"

"I3 - 0 - 1 - A/2 , B/2 , C/1 , E/1"

Isto é um caso particular de: ao menos para um Título requerido no **Search**, recebeu-se *apenas* Mensagens *de vizinhança*. O procedimento intui-se: é análogo a 3): para cada Vizinho *referido* e *que ainda não tenha interrogado* (no caso, E), o Cliente envia uma *query* relativa a esse Título (no caso, significando "Dispõe de I3?"). E arma um *timeout* T_{WatchMax} .

5c. Admita-se, entretanto, que A e D têm, cada um, ao menos um exemplar disponível de I3. Então, o Cliente receberá *duas* Mensagens com Balcão não nulo, como seja "I3 - A - 1" e "I3 - D - 1", Nesse caso, o Cliente determina o *menor Peso* para esse Título, e apresenta no ecran o pertinente Alerta "*de disponibilidade*", no caso,

"I3 reservado em A, com Peso=1".

Ademais, o Cliente envia para o(s) outro(s) Balcões onde se fez a Reserva (no caso, D), uma Mensagem de cancelamento da Reserva.

A continuação intui-se... Termina quando ocorrer um dos seguintes eventos:

- já foram apresentadas no ecran Alertas de disponibilidade para *todos* os Títulos requeridos, *um por cada um*;
- $T_{\text{SearchMax}}$ expira;
- o *timeout* T_{WatchMax} expira sem que surjam quaisquer Mensagens *de vizinhança*;
- as Mensagens *de vizinhança* recebidas remetem para Servidores que, para os Títulos "em falta", (i.e., acerca dos quais ainda não se apresentaram Alertas de disponibilidade) já foram interrogados.

$T_{\text{SearchMax}}$ será cancelado quando o Cliente tiver já assinalado no ecran o pertinente Alerta para todos os Títulos requeridos no **Search** ...

Se $T_{\text{SearchMax}}$ vier a expirar entretanto (ou se já se tiver esgotado a procura de exemplares), o Cliente lista no ecran, para cada Título requerido acerca do qual ainda não assinalou um Balcão onde ele tenha sido reservado,

- se não tiver encontrado um exemplar disponível, o Alerta:
"Título indisponível",
como seja "I3 indisponível";
- se tiver encontrado *mais que um* Balcão com exemplares disponíveis, o Alerta:
"Título reservado em ... , com Peso=PesoTotal",
como seja "I3 reservado em A, com Peso=1".

Respostas chegadas ao Cliente após este evento são ignoradas.

Notas:

1. Havendo duas (ou mais) rotas possíveis entre o Balcão de Reserva de um Exemplar e o Balcão de Preferência do Utilizador, o Peso assinalado por aquele deve ser o da rota considerada de menor Peso. Por ex., se E receber a *query* "Dispõe de I3?", deverá devolver "I3 - E - 2". (Para o efeito, as *queries* deverão transportar os Pesos que vão sendo *acumulados*: no caso, na *query* a C, esse *Peso acumulado* é 0; nas *queries* a A, B ou D, é 1; na *query* a E, seria 2).

2. Recebendo-se duas (ou mais) Mensagens confirmando a disponibilidade de um Título em mais que um Balcão,

- o Cliente deve apresentar aquele que conduz a um Peso menor. Por ex., se o Balcão de preferência for D, e o Título requerido se encontrar apenas em B e C, deve optar-se por C.

- em igualdade de Pesos, o Cliente deve apresentar o Balcão cuja identificação é *alfabeticamente menor*. Por ex., se o Balcão de preferência for D, e o Título requerido se encontrar apenas em C e E, deve optar-se por C.

- corolário: se o Balcão de preferência for D, e o Título requerido se encontrar em A, B, C e E, deve optar-se por C.

3. Nem os vizinhos de um Balcão nem os pesos das ligações mútuas devem considerar-se *fixados* de uma vez para sempre: a *topologia* pode ser alterada em *run-time* (entre dois **Search**)...

II-5 - Cancel e Take

O Utilizador pode solicitar o cancelamento (ou a confirmação) de uma Reserva feita por si; para o efeito, deve recorrer a **Cancel** (ou **Take**), como se exemplifica:

Cancel Id

Take Id

A evolução da comunicação é como segue:

1. Se se tratar de **Take**, o Cliente arma um *timeout* $T_{TakeMax}$

2. Para cada Id mencionado no comando, o Cliente determina o Balcão de Reserva; depois envia ao respectivo *Servidor* uma Mensagem solicitando o cancelamento (ou confirmação) da precedente Reserva feita pelo *User* em causa.

3. Para cada Id referido, o Servidor em causa deverá (se o correspondente exemplar ainda estiver "reservado" pelo *User* em causa) marcá-lo como "disponível" (ou, se se tratar de um **Take**, "indisponível"; neste caso, deverá devolver uma Mensagem de OK). Nota: ainda que a Mensagem recebida possa referir exemplares que não se encontram reservados, somente são disponibilizados (ou indisponibilizados) aqueles que de facto o estão por *esse* Cliente - e isso independentemente de o terem sido - ou não - a partir do computador a que o *User* está acedendo.

Aquando de requerimentos inválidos, como seja o da confirmação de exemplares inexistentes ou não reservados, ou reservados mas não pelo Cliente em causa, o Servidor deverá responder com a pertinente Mensagem de erro.

3. Se se tratar de um **Take**:

- quando forem recebidas todas as Mensagens OK correspondentes aos Balcões contactados, $T_{TakeMax}$ é cancelado e afixado no ecrã o Alerta "Reservas confirmadas".

- se $T_{TakeMax}$ vier a expirar entretanto, devem ser referidos no ecrã os Ids cuja confirmação de reserva não foi recebida.

Repare-se nos dois tipos de cancelamento de *reservas*:

- *manual*: o Cliente digita **Cancel** solicitando o cancelamento da *reserva* precedente;

- *automática*: o Cliente, tendo recebido mensagens de disponibilidade de dois (ou mais Balcões) para o mesmo Título, opta por um deles - devendo cancelar as reservas feitas em todos os outros.

II-6 - Release

O Utilizador pode sinalizar a devolução de um Exemplar; para o efeito, deve recorrer a **Release**, como se exemplifica:

Release Id

A evolução da comunicação é como segue:

1. O Cliente arma um *timeout* $T_{ReleaseMax}$

2. Para cada Id mencionado no comando, o Cliente determina o Balcão de Reserva; depois, envia ao respectivo *Servidor* uma Mensagem a requerer a remoção do Exemplar da sua Base de Dados.

Repare-se: as *Mensagens* são enviadas *iterativamente* - a todos os Servidores em causa - "*um de cada vez: passa-se ao seguinte somente após se dar por terminado o diálogo com o anterior*"!

3. Para cada Id referido, o Servidor em causa deverá (se o exemplar em causa ainda estiver "indisponível" pelo *User* em causa) removê-lo da sua Base de Dados e devolver uma Mensagem de OK. Nota: ainda que a Mensagem recebida possa referir exemplares que não se encontram indisponíveis, somente são removidos aqueles que de facto o estão por *esse* Cliente - e isso independentemente de o terem sido - ou não - a partir do computador a que o *User* está acedendo.

Aquando de requerimentos inválidos, como seja o da remoção de exemplares inexistentes ou não indisponíveis, ou indisponíveis mas não pelo Cliente em causa, o Servidor deverá responder com a pertinente Mensagem de erro.

4. Quando forem recebidas todas as Mensagens OK correspondentes aos Balcões contactados, $T_{\text{ReleaseMax}}$ é cancelado. Se $T_{\text{ReleaseMax}}$ vier a expirar ou for cancelado, e fôr recebida ao menos uma Mensagem OK, o Cliente deve:

- 1) enviar ao Servidor do Balcão de Preferência uma Mensagem de doação veiculando uma lista de todos os Títulos a respeito dos quais se recebeu uma Mensagem OK;
- 2) rearmar o *timeout* $T_{\text{ReleaseMax}}$.

5. O Servidor do Balcão de Preferência deverá adicionar esses Títulos na sua Base de Dados e devolver uma Mensagem OK.

6. Quando esta chegar ao Cliente, deverá afixar no ecran o Alerta "Doação efectuada cm sucesso" e cancelar $T_{\text{ReleaseMax}}$. Se este expirar, deverá afixar o Alerta "Insucesso na doação".

Nota: se o Balcão com a Reserva de um Exemplar fôr exactamente o Balcão de Preferência, a operação é simplificada: em vez de remoção seguida de adição, o "status" desse Exemplar é de imediato actualizado ("Indisponível \rightarrow Disponível"), por conseguinte dispensando a Mensagem de doação.

II-7 - Give

O Utilizador pode sinalizar a cedência de um (ou mais) Títulos; para o efeito, deve recorrer a **Give**, como se exemplifica:

Give Título

A evolução da comunicação é como segue:

O Cliente envia ao *Servidor* de Preferência uma Mensagem a requerer a doação dos Títulos referidos no Comando. O Servidor deverá incluí-los na sua Base de Dados.

Notas:

1. Exemplares reservados por *um Search* podem ser cancelados (ou confirmados) com *um ou mais Cancel* (ou **Take**). *Vice-versa*, exemplares reservados por *vários Search* podem ser cancelados (ou confirmados) com *um Cancel* (ou **Take**).
2. A devolução de exemplares confirmados por *um Take* pode ser sinalizada com *um ou mais Release*. *Vice-versa*, A devolução de exemplares confirmados por *vários Take* pode ser sinalizada com *um Release*.
3. Um **Release** pode ser invocado em um Cliente *distinto* daquele onde se invocaram os correspondentes **Search** e **Take**.

II-8 - Quit

O Utilizador pode solicitar a terminação *normal* de *BookShare*; para o efeito, deve recorrer a **Quit**, como se exemplifica:

Quit

Quit somente é levado por diante se todos os **Release** invocados tiverem entretanto findado. Se não for o caso, *BookShare* deverá assinalar

Aguarde: Release em curso...

O utilizador poderá ou não re-invocar **Quit**. Qualquer comando é ignorado após o *primeiro Quit*. *BookShare* parará por si quando não restar mais nenhum **Release** por executar.

Um utilizador pode, em qualquer instante, *abortar* *BookShare* pelo expediente de fazer **Ctrl-C**.

Notas:

- a sintaxe de **Balcony** é: **Balcony** [Titles.dat [Users.dat]], significando que *Users.dat* pode ser dado por omissão e, se omitido, *Titles.dat* também pode ser dado por omissão;

- a sintaxe de **BookShare** é: **BookShare**[/v] [User Password [MyBalcony]], significando que *MyBalcony* pode ser dado por omissão e, se omitido, *User* e *Password* também podem ser dados por omissão; a explicação do *switch* (opcional) /v é dada mais adiante.

("Por omissão" codifica-se inicializando a variável a um valor predeterminado; e actualizando-a se o correspondente argumento do programa não estiver vazio. Revela-se particularmente útil na fase de *debug*: torna mais cómoda a invocação do programa. Também por mor de comodidade, sugere-se, na fase de *debug*, a entrada de comandos por *scripts*, que não apenas por digitação dos ditos).

- a sintaxe de **Search** é a seguinte: **Search** Título [Título]... Significa que:

- no **Search**, podem ser referidos tantos Títulos quantos se quiser e por qualquer ordem;

- a sintaxe de **Take** é a seguinte: **Take** Id[/c] [Id[/c]]... Significa que:

- no **Take**, pode associar-se, a cada exemplar, um *switch*, /c, como se exemplifica:

Take M3/c;

se o Cliente já dispuser da *Capa-digitalizada* desse exemplar, o *switch* não deverá ter qualquer efeito; caso contrário, há-de entender-se como requerimento a que o Servidor (aquele onde foi feita a reserva) a transmita ao Cliente;

O Servidor em causa deverá proceder à transferência de Capas para o Cliente de modo *iterativo*, isto é: uma após outra, pela ordem em que foram referidas em **Take** (e enquanto o faz pode estar transferindo Capas para outros Clientes: ao invés de o Servidor satisfazer um **Take**, e depois outro e outro... ele deve ir satisfazendo os **Take** à medida que vão chegando dos Clientes)

De outra parte, se o **Take** envolver dois ou mais Servidores distintos, aquela transferência decorre *em concorrência*, isto é, o Cliente pode acabar por estar recebendo duas ou mais Capas "em simultâneo".

Com o que é possível um mesmo Servidor estar transmitindo a mesma Capa (ou Capas diferentes) para vários Clientes "em simultâneo"; e um Cliente estar recebendo (Capas diferentes) de vários Servidores "em simultâneo".

- uma Mensagem, de um Cliente a um Servidor, de cancelamento (ou confirmação) da precedente *reserva* não deve ser enviada se, pela respostas anteriores, não se pode aduzir que ele lhe tenha "reservado" exemplares;

- a sintaxe de **Cancel** é a seguinte: **Cancel** Id [Id]... Significa que, no **Cancel**, podem ser referidos (por Id) tantos exemplares quantos se quiser e por qualquer ordem;

- os comandos não são bloqueantes, isto é: após invocado um deles, o *BookShare* deve ficar de imediato pronto para aceitar qualquer outro comando. Entretanto, esse "outro" deve ser executado somente após o anterior ter sido completado;

- os **Take** (incluindo a transferência de *Capas-digitalizadas*) e **Release** são suportados em TCP; **Give**, **Search** e **Cancel** são suportados em UDP.

O número de conexões-TCP simultâneas entre um Cliente e um Servidor é, no máximo, de *uma*. Ela é criada quando, necessitando de uma conexão com tal Servidor, o Cliente ainda não dispuser de nenhuma. Intenta-se que **Take** e **Release** afectando o mesmo Servidor sejam executados sobre a mesma conexão-TCP; para o efeito, e sempre que envia uma Mensagem OK, o Servidor arma/re-arma um *timeout* $T_{\text{KeepOpenMax}}$; se esse *timeout* vier a expirar, termina a conexão-TCP.

Aquando de **Give** e **Cancel**, Mensagens que venham a ser recebidas num Servidor *não são confirmadas* por quaisquer Mensagens.

Consoante o caso, os comandos podem ou não devolver Alertas de erro; entre os possíveis, poderão citar-se os seguintes:

- Parâmetros insuficientes, ou em excesso ou inválidos, etc.
- Dificuldades na comunicação; Erro de sistema
- Falha no *download* da Capa de P1

Procedimentos aquando de eventuais *Crashes*:

- num Servidor: quando o seu funcionamento normal é re-estabelecido, são de imediato considerados "disponíveis" todos os exemplares que eventualmente estavam "reservados" aquando desse *crash*;

- num Cliente: o seu estado de progresso é "apagado", fica-se sem se saber "onde é que já se ía...". Isso pode levar a uma des-sincronização entre Cliente e Servidor. Por exemplo, o Cliente pode ter invocado **Release Id**, o Servidor com a Reserva ter removido aquele Id da sua Base de Dados - mas a pertinente Mensagem OK não chegar a ser processada pelo Cliente em virtude de *crash* (quicá nem mesmo tenha chegado a ser produzida). *Para não complicar desnecessariamente, no Projecto subentende-se: não há que se preocupar com procedimentos de re-sincronização...*

II-9 - Mensagens de Erro

O Servidor poderá, eventualmente, responder ao Cliente com uma ou mais Mensagens de Erro. O Cliente deverá "ecoá-las" no ecrã com o pertinente Alerta de Erro, como seja:

- "User inválido", "Password inválida";
- "Exemplar não-reservado";
- "Exemplar não-encontrado";
- "Capa digitalizada inexistente ou inacessível"
- "Falha na confirmação da reserva de P1"

Desenhe - em *C* - o *software* mínimo necessário de Servidores e Clientes (incluindo gestão dos registos e comunicações).

Na sua concepção,

- distinga claramente o nível de protocolo *BookShare* ("Partilha de Livros") dos respectivos "utilizadores": teclado e ecrã nos *Clientes* e sistema de ficheiros nos *Clientes* e nos *Servidores*;

- garanta que o Servidor continua a poder servir outros clientes, mesmo após ter sido estabelecida uma conexão-TCP com algum deles;

- garanta que o Servidor continua a poder servir outros clientes, mesmo após um ou mais deles ter sofrido um *crash*; e que o Utilizador continua a poder aceder a outros Servidores, mesmo após um ou mais deles ter sofrido um *crash*;

- siga as *Recomendações acerca do Relatório preliminar* publicadas

- siga as *Recomendações para a escrita do código* publicadas.

- não complique desnecessariamente os ficheiros de registos e não perca tempo a "embonecar" a interface homem-máquina; e não complique desnecessariamente o enunciado (por exemplo, introduzindo preços, descontos, penalizações, etc., etc.)

- todos os parâmetros ($T_{\text{SearchMax}}$, $T_{\text{ReservMax}}$, T_{TakeMax} , $T_{\text{KeepOpenMax}}$ e outros) são lidos a partir de ficheiros de configuração;

- a demonstração do *Software* deverá percorrer vários contextos, nomeadamente:

- *Cenário básico*: um Servidor e um Cliente instalados num mesmo computador, a saber: 1) *um PC do Laboratório da Disciplina (LT5)*, por ex.: *sado.ist.utl.pt*; 2) o *mega*;

- *Cenário genérico* (a demonstrar apenas se o Cenário básico funcionou correctamente): Servidores e *Clientes* correndo no *mega* e/ou em um ou vários *PCs do Laboratório da Disciplina*; num mesmo computador podem residir apenas Servidores, ou apenas Clientes, ou ambos.

- cada Servidor deverá ir reportando eventos significativos, como se exemplifica:

- C: recebida Mensagem de procura de P1 e I3;
- A: recebida Mensagem de cancelamento de reserva de A-P1;
- A: começada uma conexão com o Cliente [...];
- A: recebida Mensagem de confirmação de reserva de A-I3;
- A: terminada uma conexão pelo Cliente [...].
- C: *timeout* $T_{ReservMax}$ expirou.

- se e só se fôr invocado o *switch /v*, o Cliente deverá ir reportando eventos significativos, como se exemplifica:

- U1: recebido o OK da confirmação de A-I3.
- U1: recebido o bloco nº N da Capa de P1.
- U1: *timeout* $T_{WatchMax}$ expirou.

Grelha de avaliação (em que **P** e **R** significam Balcão de Preferência/de Reserva):

Especificação	3	Diagramas temporais Diagramas/Tabelas de estado Especificação das PDUs
Lançamento de: Balcony e BookShare Interface	0,5	Ficheiros de configuração Validação dos comandos <i>Balcony</i> e <i>BookShare</i> Validação de comandos Search , etc .
Cenário Básico de: Give, Search, Cancel, Take e Release (Servidor P <i>up</i> e um Cliente co- residentes no mesmo computador: <i>mega</i> ou <i>sado</i>)	3	• Give • Search com Título irreconhecível/indisponível/disponível. • Cancel e Take com Id inexistente, disponível, reservado • Release
Cenário genérico: - Um Servidor e um Cliente, em computadores diferentes (ex.: <i>mega</i> e <i>sado</i> , ou vice-versa) - Vários Servidores, um Cliente - Um Servidor, vários Clientes - Vários Servidores, vários Clientes	2,5	P satisfaz o pedido Search P não o satisfaz, mas algum vizinho fá-lo. Nem P nem vizinhos de P satisfazem o Search : Reserva não-feita/feita-em: 1 Balcão/ 2 Balcões <i>Cancel automático</i> aquando de duas Reservas Colisão de dois Search sobre mesmo Título Release •sobre P ; •sobre P e um ou mais R .
Give	0,5	•com P <i>down</i> ; •com P <i>up</i> - e, então, com •Título-novo/•Título-já-existente em P
Cancel	1,5	•com P <i>down</i> ; •com P <i>up</i> - e, então, •com R <i>down</i> ; •com R <i>up</i> - e, então, •com Id inexistente, disponível, reservado... •referindo Ids no mesmo Balcão (ou não)
Take	1,5	• <i>Análise análoga a Cancel</i> •Falha na conexão (antes ou após <i>accept</i>)
Switch /c	2	Transferência de Capa-digital com •Ficheiro destino existente/inexistente -e, então: •Fich fonte inexistente ou inacessível/existente •Concorrência na transferência de Capas

Quit	0,5	<ul style="list-style-type: none"> •com/•sem Release em curso de execução •Reinvocação de Search, Cancel, Quit, etc
Switch /v	0,5	Sinalização no ecran da recepção dos itens
Outros:	2,5	Eventos que provocam <i>timeouts</i> ; Alteração de <i>timeouts</i> Partilha de uma conexão por Take/Release ; Terminar conexão-TCP: <ul style="list-style-type: none"> •pelo Servidor; •por <i>crash</i> do Cliente. Retoma após <i>crash</i> do-cliente/do-servidor Sinalização no ecran após comandos Sinalização no ecran pelo Servidor Comandos não bloqueantes <i>kill, Ctrl-C</i> durante Search , etc
Qualidade do código	1	
Qualidade dos testes preparados	1	