Universidade de Coimbra

Faculdade de Ciências e Tecnologia

Licenciatura em Engenharia Informática

*1º Trabalho Prático – Sockets, RMI e Sincronização de Servidores*

*Sistemas Distribuídos, 1º Semestre*



**Trabalho realizado por:**

* **Ivo Correia nº 2008110814**
* **João Barbosa nº 2008111830**

Coimbra, 22 de Outubro de 2010

**Índice**

**Introdução2**

**Diagrama de Casos de Uso2**

**Casos de Uso do “Back Office”3**

Alínea 1.1. – 1.2.3

Alínea 1.3.4

Alínea 1.4. – 1.5.5

Alínea 1.6.6

**Casos de Uso do “Front Office”7**

Alíneas 2.1 .7

Alíneas 2.2. – 2.3. 8

Alíneas 2.4. – 2.5. 9

Alínea 2.6.11

**Conclusão18**

**Bibliografia19**

**Introdução**

O primeiro trabalho prático da disciplina de Sistemas Distribuídos consistia na implementação de uma aplicação distribuída cliente-servidor, o qual permitia gerar jogos de futebol, os seus resultados, permitir que os clientes façam as suas apostas e distribuir prémios dos apostadores vencedores.

De modo a alcançar tal objectivo, são instalados dois servidores que têm como uma das funções comunicar entre si de modo a evitar que nem todos os clientes estejam ligados ao mesmo servidor. Por isso, trocam mensagens entre si para eleger um servidor primário e em caso de falha deste, o servidor secundário tem de obrigatoriamente tomar esta posição para que os clientes possam continuar a fazer as suas apostas.

Do lado dos clientes, temos dois tipos de aplicações: ligações através de *sockets* TCP e Java RMI. Em termos de interface com o utilizador, ambas as abordagens são em tudo semelhantes. Quanto às suas aplicações, logicamente que as duas são internamente bastante diferentes. Contudo, é da responsabilidade do servidor conseguir atender simultaneamente os dois tipos de clientes sem que estes criem conflitos entre si.

De modo a auxiliar a execução do trabalho, foi planificado no enunciado quatro etapas para a realização do projecto, a saber: implementação dos *sockets* TCP, implementação do Java RMI, tratamento de excepções e falhas de comunicação entre servidor e clientes e por último, criação de mecanismos de protecção contra falhas num dos servidores.

No entanto, não seguimos exactamente esta ordem, uma vez que apenas sabíamos trabalhar de antemão com *sockets* TCP e ligação UDP. Assim, começámos pela implementação destes dois pontos, de modo a ligar um cliente ao servidor. Depois, não tendo ainda leccionado Java RMI nas aulas teóricas, partimos para o estabelecimento de mecanismos de modo a seleccionar o servidor primário e garantir que este fosse substituído no caso de falhar da máquina. Por fim, para concluir, partimos para o Java RMI.

Em paralelo a cada passo, fomos implementando o controlo de excepções pelo cliente sempre que possível para as operações que íamos gradualmente implementando.

Antes de iniciar o projecto em concreto, foram desenhados diagramas e apontados os principais problemas com que nos iríamos deparar. Nesta fase, observámos que as maiores dificuldades seriam a coordenação entre servidores e garantir a persistência dos dados; assegurar que nenhuma informação fosse perdida e no caso de ser impossibilitado de trocar a mesma, avisar o cliente da falta de ligação; permitir a interoperabilidade dos diferentes tipos de clientes.

**Casos de Uso do “Back Office”**

Para o “Back Office”, apenas encontramos um actor, o administrador.

**Login:**

Este caso de uso permite a autenticação de um administrador do sistema. Ao autenticar-se com sucesso, o administrador terá acesso a um conjunto de funcionalidades referentes ao “Back Office”.

**INPUT/OUTPUT:** Ao receber um *username* e uma *password*, o sistema devolve o resultado conforme a autenticação tenha sucesso ou falhe.

**Emitir Notificações:**

Este caso de uso cobre a funcionalidade que permite ao administrador do sistema fazer *broadcast* de avisos e novidades relativas à companhia aérea que não sejam automáticas (como o cancelamento de um voo aquando a sua eliminação, por exemplo). Esta funcionalidade é da mais extrema utilidade porque há certos tipos de eventos como descontos de época baixa, mensagem de "Boas Festas" da companhia, alteração da política de transporte de certos materiais no voo, etc.

Para além da emissão de notificações de carácter geral, também será possível o esclarecimento pontual personalizado aos comentários dos clientes.

**INPUT/OUTPUT:** Recebe o conjunto de clientes a notificar (vários ou apenas um) assim como a mensagem a enviar, verificando se esta chega a todos os seus destinatários.

**Gerir Voos:**

Este caso de uso concede a funcionalidade ao administrador do sistema de criar, cancelar ou adiar voos.

É possível planear voos de dois tipos, regulares ou *charter*. O adiamento ou cancelamento dos voos também é de extrema importância pois a existência de avarias e de más condições atmosféricas é um condicionalismo bastante frequente. Quando um voo for cancelado ou adiado, todos os seus passageiros serão informados deste contratempo.

**INPUT/OUTPUT:** Na criação de um voo é necessário inserir a data e o tipo de voo que se pretende. Já nos adiamentos ou cancelamentos apenas será necessário especificar o número do voo. O sistema responderá em conforme com a operação pretendida tenha tido sucesso ou não.

**Gerir Aviões:**

Os aviões são a os principais activos de uma companhia aérea. Desta forma esta funcionalidade permite comprar ou vender aviões.

**INPUT/OUTPUT:** Como parâmetros de entrada o administrador pode escolher a empresa a quem vai comprar o avião, a capacidade, a potência do motor e a velocidade média. Como resposta o sistema dará informações sobre o sucesso ou insucesso da compra.

**Gerar Estatísticas:**

A funcionalidade de geração e tratamento de dados estatísticos é dos aspectos mais relevantes dentro de uma companhia aérea.

Por isso, neste programa de gestão, pretende-se munir o administrados do sistema de um conjunto de ferramentas que permitam controlar despesas, filtrar dados estatísticos relevantes, ter acesso a dados que possam ser usados por uma possível secção de *marketing* (taxa de *feedback* positivos por exemplo).

**INPUT/OUTPUT:** O sistema pedirá o conjunto de dados a serem tratados e as suas balizas de validade (i.e. quais os dados que interessam ao administrador e em que intervalo de tempo), mostrando uma tabela ou lista com as estatísticas devidamente ordenadas.

**Gerir Clientes:**

Todo o administrador sabe que a chave de sucesso do negócio está na satisfação dos clientes. Seguindo esta máxima, o administrador terá o poder de acesso ao perfil dos seus clientes (consultando informações como morada ou histórico de viagens) e passar facturas, dando apenas dois exemplos. Também por uma questão de respeito da privacidade do cliente, será possível eliminar registos a pedido do utilizador ou como medida de coação por algum delito efectuado.

**INPUT/OUTPUT:** Será requerido a identificação do cliente e a acção a realizar sobre a sua informação, devolvendo o sucesso da operação.

**Casos de Uso do “Front Office”**

No caso do “Front Office”, temos uma situação diferente da anterior. Sendo que os voos *charter* não podem ser reservados por clientes individuais. Por isso, criámos a figura de Operador Turístico. Consequentemente, para o “Front Office”, encontramos como actores o Cliente e Operador Turístico. Como os casos de uso são todos semelhantes, tanto par ao operador como para o cliente regular, referimo-nos a ambos como cliente, a menos que a distinção seja relevante.

**Login:**

Depois de autenticado, o cliente poder usufruir das funcionalidades referentes ao "Front Office", tal como já acontecia para o administrador. Será sempre necessário ter uma conta no sistema, pois estas funcionalidades não ficaram disponíveis caso o cliente se engane na sua autenticação ou não tenha feito anteriormente um registo válido.

**INPUT/OUTPUT:** Neste caso de uso, cliente introduzirá o seu nome de utilizador e *password*, sendo-lhe concedida ou recusada a entrada no sistema.

**Registar:**

Este caso de uso serve para o cliente ficar registado na base de dados da companhia aérea. Sem este procedimento, não lhe é possível usufruir dos serviços da mesma nem efectuar o *login*.

**INPUT/OUTPUT:** Serão pedidos dados ao utilizador, como um *username*, *password*, endereço electrónico, morada, contacto e outras informações relevantes a definir. Será informado se o seu registo foi efectuado com sucesso ou não (por exemplo, poderá já haver um cliente com o mesmo nome de utilizador no sistema).

**Dar feedback:**

Este caso de uso serve para o cliente, se desejar, fornecer à companhia aérea a sua opinião acerca dos serviços por esta prestados (voos, atendimento, etc.). Dividiremos o tipo de mensagens em reclamações, sugestões/recomendações ou gratificações.

**INPUT/OUTPUT:** Como entrada, teremos o tipo de mensagem a enviar e a mensagem propriamente dita. O cliente será informado se a mensagem foi bem encaminhada ou não.

**Alterar voo:**

Este caso de uso serve para o cliente alterar a sua reserva de voo, se necessitar, para qualquer outro voo da companhia aérea, caso tal acção seja possível. Não são permitidos cancelamentos de voos uma vez feita e paga a reserva.

**INPUT/OUTPUT:** Será pedido a identificação dos dois voos, tanto o antigo como o voo para o qual se pretende mudar. O cliente será informado do estado da operação e em caso negativo, o motivo do insucesso.

**Alterar informação pessoal:**

Neste campo, o cliente poderá modificar os seus dados pessoais junto da companhia aérea, como a sua morada, telefone, entre outros.

**INPUT/OUTPUT:** Serão pedidos os dados a modificar e a nova informação, sendo os dados registados no sistema se não houver inconsistências.

**Consulta de Viagens:**

O caso de uso de consulta de viagens divide-se em vários, tal como é visível no diagrama.

Primeiro, damos a possibilidade de um cliente anónimo entrar no sistema e consultar os voos disponíveis para marcação. Se desejar efectuar a reserva, terá então de se autenticar. Se o cliente se tiver autenticado primeiro, então o voo, se estiver disponível, poderá ser marcado sem mais demoras.

Em segundo, também temos uma divisão entre consulta de viagens de voos regulares ou *charter*. Um cliente normal apenas poderá consultar voos regulares e nunca *charter*, sendo que para o operador turístico temos a situação inversa.

**INPUT/OUTPUT:** Será pedido um dia e/ou hora de voo e serão listadas no ecrã todas as possibilidades para essa data.

**Reserva de Viagens:**

Tal como no caso anterior, a reserva de viagens também é diferente para voos regulares ou *charter*, sendo que no entanto, esta acção só está disponível caso o utilizador se tenha autenticado anteriormente.

**INPUT/OUTPUT:** Depois de escolhido o voo, é passado ao sistema a identificação tanto do voo como do cliente e o mesmo é informado sobre a conclusão da operação.

**Pagamento:**

Antes da conclusão da reserva da viagem, é necessário efectuar o pagamento da reserva, não sendo o lugar ou lugares marcados como ocupados enquanto o dito pagamento não se verificar.

**INPUT/OUTPUT:** É pedido ao utilizador os dados de pagamento (montante, número de cartão de débito) e em caso de não ocorrerem problemas, é concluída a reserva.

**Imprimir Bilhete:**

Tal como o nome indica, neste caso de uso procedemos à impressão do bilhete. Possibilitamos a impressão logo a seguir à conclusão da reserva ou então num momento posterior, assegurando ao utilizador, mesmo que este perca a sua primeira impressão, ainda tenha acesso ao bilhete.

**INPUT/OUTPUT:** É pedido a identificação da reserva (passada automaticamente caso a impressão se efectue imediatamente a seguir à reserva) e o respectivo bilhete é devolvido.

**Bibliografia**

1. Java Eclipse
2. Ficheiros disponíveis na WOC sobre princípios básicos do *MATLAB*
3. Imagens: **Da capa:**
   1. http:// http://www.destination360.com/

**Gráficos -** Resultados obtidos pela execução do código elaborado no *MATLAB*.

1. “The Scientist and Engineer’s Guide to Digital Signal Processing” - Smith, Steven. *Edição online*.
2. Sites visitados:
   1. http://new.myfonts.com/
   2. http://www.bwin.com/