

UNIVERSIDADE LUSOFONA DE
HUMANIDADES E TECNOLOGIAS
Licenciatura em Engenharia Informática
Projecto Final de Curso

Aprendizagem assistida na
sala de aula, utilizando
tecnologia Microsoft

Relatório Final



Alunos:
Pedro Fanico - 20062923
Miguel Nadais - 20062900

Índice

Índice de Figuras	3
1. Abstract	4
2. Introdução	5
3. Enquadramento/Objectivos/Descrição	6
4. Diagrama de Gantt	7
5. Questionário	9
5.1. Resultados do Questionário efectuado aos Docentes	11
5.2. Conclusões	14
6. Enquadramento Teórico	15
6.1. Escolha da tecnologia se adapta melhor ao nosso trabalho	15
6.2. Diagramas	16
6.2.1. Use Case das interacções e-learning	16
6.2.2. Diagramas de Actividades	18
6.2.3. Arquitectura	20
7. Método	23
7.1. Abordagem	23
7.2. Protótipo Final	25
7.2.1. Criação do canal servidor	25
7.2.2. Criação do canal cliente	25
7.2.3. Registo dos canais	26
7.2.4. Classe ComputerControl	26
7.2.5. Conexão entre cliente e servidor	26
7.2.6. Pedido de conexão por parte do aluno	27
7.2.7. Gestão da lista de espera	27
7.2.8. API's do Windows	27
7.2.9. Controlo remoto	28
7.2.10. Difusão de ecrã de trabalho do Master	28
7.2.11. Diagrama de Classes	28
8. Resultados	29
9. Conclusões e Trabalho Futuro	31
Bibliografia	32

Índice de Figuras

4.1 - Diagrama de Gantt	7
4.2 - Diagrama de Gantt (cont.)	8
5.1 - Gráfico da Pergunta 1	11
5.2 - Gráfico da Pergunta 2.1	11
5.3 - Gráfico da Pergunta 2.2	11
5.4 - Gráfico da Pergunta 2.3	12
5.5 - Gráfico da Pergunta 2.4	12
5.6 - Gráfico da Pergunta 2.5	12
5.7 - Gráfico da Pergunta 2.6	12
5.8 - Gráfico da Pergunta 2.7	13
5.9 - Gráfico da Pergunta 2.8	13
6.1 - Use Case Ligação Remota	16
6.2 - Use Case Apresentações Remotas	17
6.3 - Diagrama de Actividades da Ligação Remota	18
6.4 - Diagrama de Actividades de Apresentações Remotas	19
6.5 - Arquitectura Geral da Rede na Sala de Aula	20
6.6 - Arquitectura da Ligação Remota	21
6.7 - Arquitectura de Apresentações Remotas	22
7.1 - Diagrama de Classes	28
8.1 - Lista de Espera	29

1. Abstract

The term e-Learning is the result of a combination occurring between the schools with aid of technology and distance education. Both methods converged on the online education and its implementation based on Web, which eventually resulted in e-Learning.

His sudden arrival has brought new meaning to education and made explode the possibilities for disseminating knowledge and information for students, and an accelerated pace, opened a new world for distribution and sharing of knowledge, making it also a way to democratize knowledge for the layers of the population with access to new technologies, enabling that knowledge is available anywhere and anytime.

In order to support the process, were developed LMS's (Learning Management System), the management of teaching and learning on the web.

Software designed to act as virtual classrooms, creating several potential interactions between the participants. With development of web technology, the processes of interaction in real time became a reality, allowing that the student has contact with the knowledge, with the teacher and other students, through a virtual classroom.

The interactivity provided by the networks of Internet, intranet, and by management environments, where it is the e-learning, according to current social interaction, is now seen as a means of communication among learners, mentors and those with the means. From this premise, is able to provide interaction in the following levels:

- Apprentice / Advisor;
- Apprentice / content;
- Apprentice / Apprentice;
- Apprentice / Environment.

A simple definition for e-learning would be the process by which the student learn through content placed on the computer and / or Internet and the teacher, if there is a distance using the Internet as means of communication (synchronous or asynchronous) sessions may be intermediate classroom. The system includes classroom in system of e-learning is called blended learning.

2. Introdução

A escolha deste projecto por parte do nosso grupo prende-se com o facto de, a nosso ver os sistemas de e-learning serem uma ferramenta bastante útil que cada vez mais começam a fazer parte do sistema de ensino devido à sua versatilidade e fácil aprendizagem, dando-nos assim a possibilidade de termos um contacto directo com a criação de uma ferramenta que possa ser ainda mais desenvolvida por futuros grupos e de futura utilidade na faculdade.

O trabalho aqui relatado é a implementação de um sistema de e-learning que permite duas funcionalidades ao docente, uma ligação remota ao terminal de um aluno para o esclarecimento de dúvidas ou verificar o trabalho que um aluno esteja a desenvolver e também a apresentação de imagem remotamente para os terminais dos alunos. A escolha destas funcionalidades deveu-se ao resultado de um questionário que foi enviado para vários docentes, com o propósito de percebermos do ponto de vista dos docentes, até que ponto uma ferramenta deste tipo seria útil e quais as funcionalidades que os próprios acham mais úteis.

Achamos que este projecto pode vir a ser aumentado com novas funcionalidades por futuros grupos, construindo assim uma ferramenta que sirva as necessidades da faculdade.

3. Enquadramento/Objectivos/Descrição

Enquadramento:

-Necessidade dos docentes adoptarem um método de ensino em possam tirar dúvidas e esclarecer os alunos presentes na sala de aula sem terem necessidade de se deslocar aos seus terminais permitindo assim uma aula com menos perturbações.

Objectivos:

- Permitir um método de ensino mais rentável, através de uma comunicação e visualização nos terminais presentes nas salas de aulas.
- Permitir a um administrador de sistemas de informática visualizar algum terminal na rede e o trabalho que nele esteja a ser desenvolvido
- Permitir a comunicação de voz (vídeo) entre dois terminais.

Descrição:

- Reunião com responsáveis do departamento da COFAC para perceber as necessidades existentes e o levantamento de requisitos inicial.
- Realização de um documento explicativo dos objectivos e descrição.
- Criação de um questionário que nos permita compreender mais sucintamente as necessidades dos docentes.
- Continuação da realização do documento explicativo com os requisitos e resultado esperado.
- Estudar o melhor método (tecnologias) para a realização do trabalho.
- Analise dos dados recolhidos nas fases anteriores com o objectivo de realizar casos de uso, modelos de dados (se necessário), diagramas temporais do projecto e tentar balizar os objectivos esperados.
- Desenvolvimento/testes da aplicação (protótipo).
- Apresentação preliminar da aplicação (protótipo).

4. Diagrama de Gantt

		Task Name	Duration	Start	Finish	Predecessors
1		<input type="checkbox"/> PFC	196 days?	Sun 01-03-09	Tue 17-11-09	
2		<input type="checkbox"/> Análise	87 days?	Sun 01-03-09	Sun 21-06-09	
3		<input type="checkbox"/> Escolha do PFC	18 days?	Sun 01-03-09	Mon 23-03-09	
4		Entrega das Preferências	11 days?	Sun 01-03-09	Fri 13-03-09	
5		Escolha do PFC	7 days?	Sat 14-03-09	Mon 23-03-09	4
6		<input type="checkbox"/> Levantamento de Requisitos	69 days?	Tue 24-03-09	Sun 21-06-09	
7		Reunião com Orientador	5 days?	Tue 24-03-09	Mon 30-03-09	5
8		Reunião com a COFAC	4 days?	Tue 31-03-09	Fri 03-04-09	7
9		Proposta 1 - Objectivos	11 days?	Sat 04-04-09	Fri 17-04-09	8
10		Proposta 2 - Questionário	12 days?	Sat 18-04-09	Mon 04-05-09	9
11		Proposta 3 - Tecnologia	18 days?	Tue 05-05-09	Wed 27-05-09	10
12		Proposta 4 - Use Cases	9 days?	Thu 28-05-09	Tue 09-06-09	11
13		Proposta 5 - Gantt	4 days?	Wed 10-06-09	Sun 14-06-09	12
14		Proposta 6 - Sequências	6 days?	Mon 15-06-09	Sun 21-06-09	13
15		<input type="checkbox"/> Desenvolvimento	109 days?	Mon 22-06-09	Tue 17-11-09	
16		<input type="checkbox"/> Desenho da Arquitectura	32 days?	Mon 22-06-09	Fri 31-07-09	
17		Geral	32 days?	Mon 22-06-09	Fri 31-07-09	14
18		<input type="checkbox"/> Programação	69 days?	Mon 03-08-09	Thu 05-11-09	
19		Slave Side	51 days?	Mon 03-08-09	Mon 12-10-09	17
20		Master Side	18 days?	Tue 13-10-09	Thu 05-11-09	19
21		<input type="checkbox"/> Protótipo	8 days?	Fri 06-11-09	Tue 17-11-09	
22		Criação	8 days?	Fri 06-11-09	Tue 17-11-09	20
23		<input type="checkbox"/> Conclusão	180 days?	Tue 24-03-09	Thu 19-11-09	
24		Relatório	180 days?	Tue 24-03-09	Thu 19-11-09	

Figura 4.1 - Diagrama de Gantt

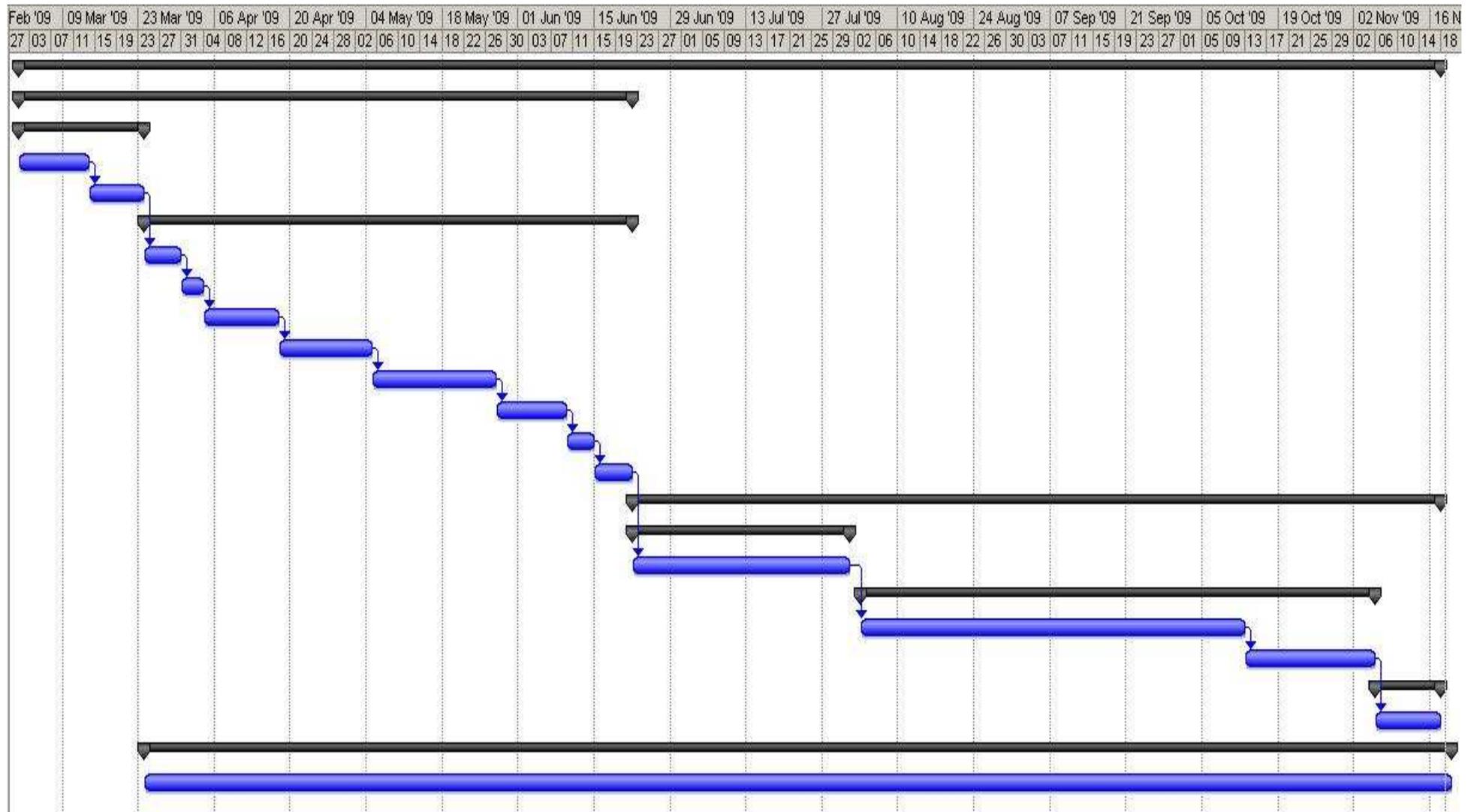


Figura 4.2 - Diagrama de Gantt (cont.)

5. Questionário

1- Acha que um software de e-learning pode ajudar nas suas aulas?

Sim ____

Não ____

2 - Classifique as seguintes funcionalidades:

2.1 - Ligação remota ao computador de um aluno para esclarecimento de duvidas ou verificar o trabalho que um aluno esteja a desenvolver?

Inútil	Pouco Útil	Útil	Muito Útil

2.2 - Comunicação de voz directamente com um aluno?

Inútil	Pouco Útil	Útil	Muito Útil

2.3 - Alertar os alunos (i.e chamadas de atenção) visuais ou auditivos?

Inútil	Pouco Útil	Útil	Muito Útil

2.4 - Bloquear as acções dos computadores dos alunos (teclado, rato)?

Inútil	Pouco Útil	Útil	Muito Útil

2.5 - Bloquear certas aplicações nos computadores dos alunos?

Inútil	Pouco Útil	Útil	Muito Útil

2.6 - Bloquear certos protocolos/portos de comunicação (http, pop3, msn, etc)?

Inútil	Pouco Útil	Útil	Muito Útil

2.7 - Comunicação inter-salas de aulas com outros docentes ou alunos?

Inútil	Pouco Útil	Útil	Muito Útil

2.8 - Apresentações de vídeo/slides/som remotamente para os computadores dos alunos?

Inútil	Pouco Útil	Útil	Muito Útil

8- Outras sugestões:

1) Resultados do Questionário efectuado aos Docentes

1- Acha que um software de e-learning pode ajudar nas suas aulas?

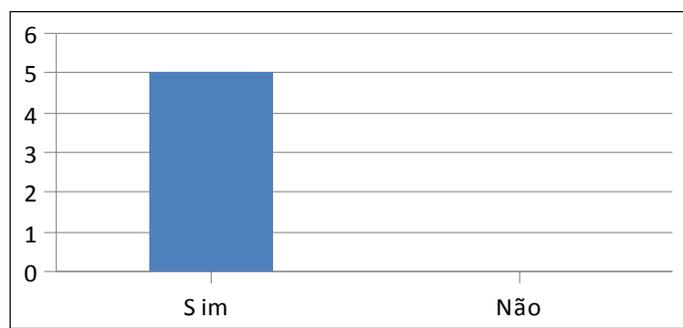


Figura 5.1 - Gráfico da Pergunta 1

2 - Classifique as seguintes funcionalidades:

2.1 - Ligação remota ao computador de um aluno para esclarecimento de duvidas ou verificar o trabalho que um aluno esteja a desenvolver?

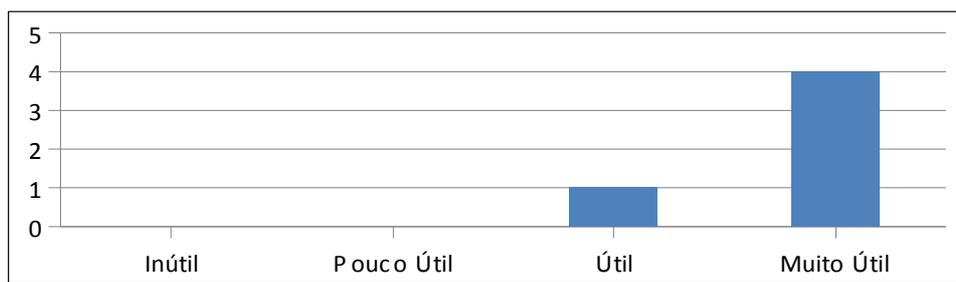


Figura 5.2 - Gráfico da Pergunta 2.1

2.2 - Comunicação de voz directamente com um aluno?

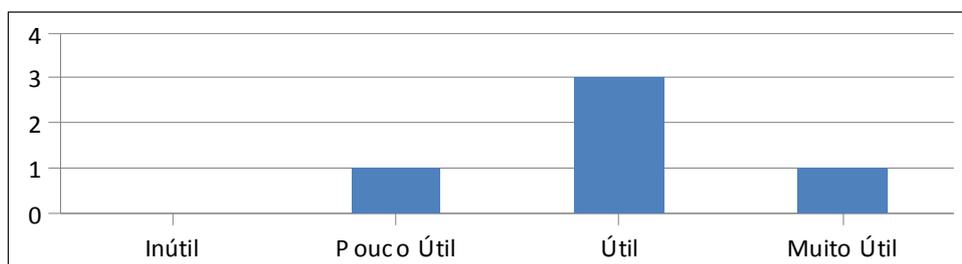


Figura 5.3 - Gráfico da Pergunta 2.2

2.3 - Alertar os alunos (p.e chamadas de atenção) visuais ou auditivos?

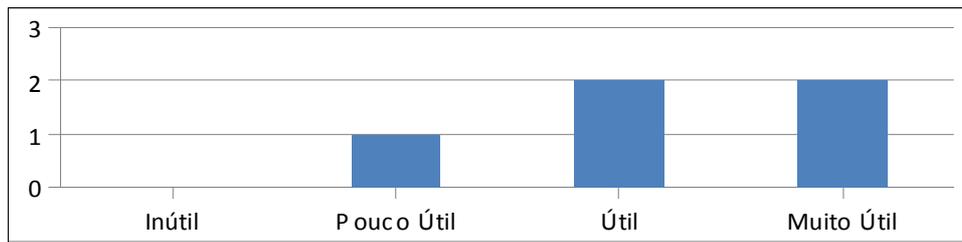


Figura 5.4 - Gráfico da Pergunta 2.3

2.4 - Bloquear as ações dos computadores dos alunos (teclado, rato)?

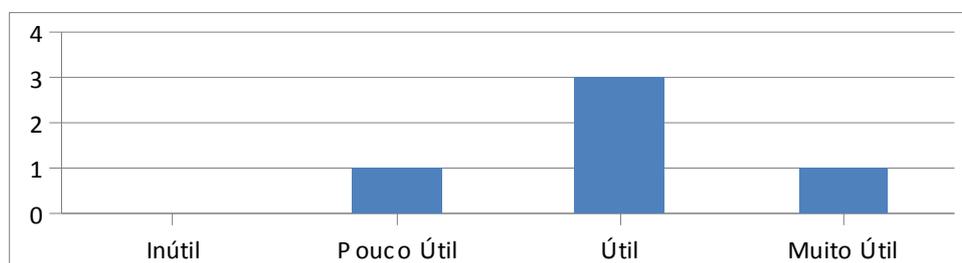


Figura 5.5 - Gráfico da Pergunta 2.4

2.5 - Bloquear certas aplicações nos computadores dos alunos?

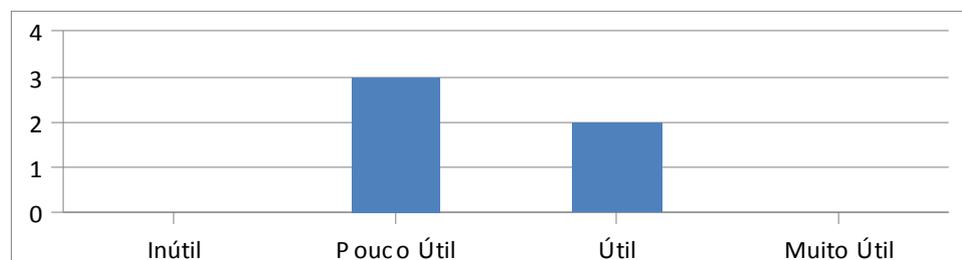


Figura 5.6 - Gráfico da Pergunta 2.5

2.6 - Bloquear certos protocolos/portos de comunicação (http, pop3, msn, etc)?

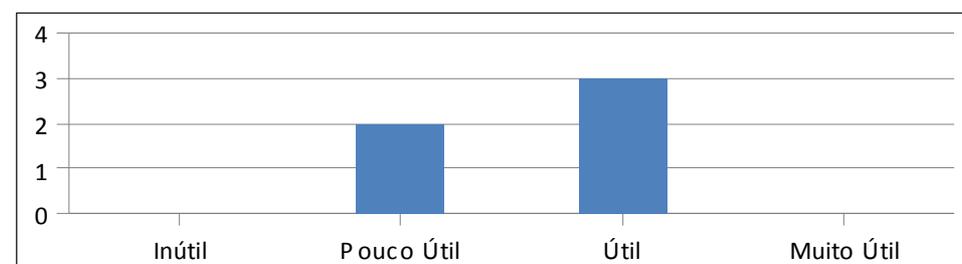


Figura 5.7 - Gráfico da Pergunta 2.6

2.7 - Comunicação inter-salas de aulas com outros docentes ou alunos?

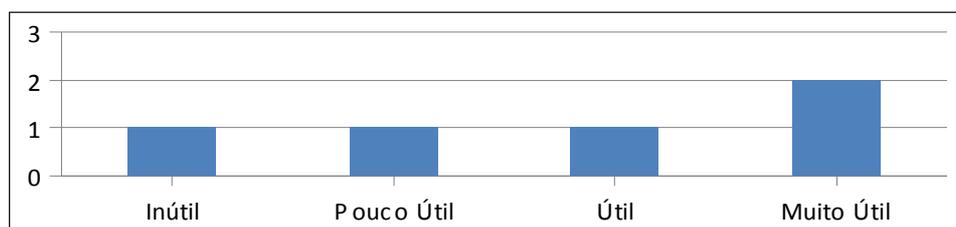


Figura 5.8 - Gráfico da Pergunta 2.7

2.8 - Apresentações de vídeo/slides/som remotamente para os computadores dos alunos?

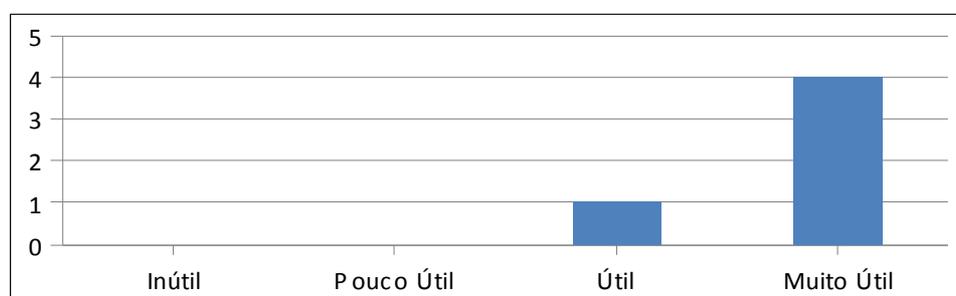


Figura 5.9 - Gráfico da Pergunta 2.8

3- Outras sugestões:

- Gestão destas aplicações / funcionalidades para grupos de alunos e não apenas aluno a aluno.
- Porque não fomentar a partilha de informações entre alunos, de forma a incentivar a descoberta colaborativa de soluções?
- Num sistema de e-learning não podemos condicionar o terminal do aluno...
- Disponibilização de enunciados de exercícios, de forma automática e à medida de cada aluno, nos terminais dos alunos.

2) Conclusões:

- Apesar de ser uma amostra reduzida (apenas tivemos 5 respostas aos questionários) o que não permite grandes conclusões parece-nos pertinente manter este levantamento da opinião dos utilizadores finais.
- Após o tratamento dos dados obtidos através dos questionários, chegámos às seguintes conclusões sobre as necessidades e preferências dos docentes:
- A totalidade dos docentes que entregaram os questionários é da opinião que um software desta natureza irá trazer grandes benefícios nas aulas. Na sua maioria será muito útil a realização de ligações remotas ao terminal de um aluno para esclarecimento de dúvidas ou verificar o trabalho que um aluno esteja a desenvolver, alertar os alunos individualmente através da plataforma e a realização de apresentações de vídeo/slides/som remotamente para os seus terminais.
- Outras funcionalidades que foram julgadas como úteis foram a comunicação inter-salas de aulas com outros docentes ou alunos, bloquear as acções dos terminais dos alunos (teclado, rato) e comunicação de voz directamente com um aluno durante a aula.
- Excluámos do nosso trabalho as funcionalidades que permitiam bloquear certos protocolos/portos de comunicação (http, pop3, msn, etc.) e certas aplicações nos terminais.
- Conforme avançamos no trabalho iremos ter em conta algumas das sugestões feitas pelos docentes.

6. Enquadramento Teórico

1) Escolha da tecnologia se adapta melhor ao nosso trabalho

Após alguma investigação sobre qual a tecnologia mais indicada para a realização do trabalho chegámos à conclusão que podemos escolher entre duas tecnologias, Java ou .Net Framework. A seguir enumeramos algumas vantagens e desvantagens de cada uma delas.

Java:

- Vantagens:
 - Permite a utilização da Virtual Machine, ou seja permite correr em qualquer SO.
 - Permite usar Open Source, permitindo assim a utilização de código já pré-feito que nos pode ajudar na celeridade do desenvolvimento da aplicação.
- Desvantagens:
 - Como o projecto foi proposto para ambiente Windows será mais complicado utilizar algumas bibliotecas Microsoft.
 - A nível pessoal vai requerer mais investigação por ser uma ferramenta que nos é mais desconhecida.
 - Necessidade de ter instalado JVM.
 - Maior dificuldade para a ligação a bases de dados.

.Net Framework:

- Vantagens:
 - Adequa-se mais à tecnologia Microsoft.
 - É uma tecnologia onde, a nível pessoal, nos sentimos mais à vontade.
- Desvantagens:
 - Necessidade de ter instalada. Net Framework.
 - Maior dificuldade em encontrar código ou bibliotecas que nos possam auxiliar no desenvolvimento da aplicação.
 - Apenas disponível para a tecnologia Microsoft.

Na nossa opinião a escolha dado às condicionantes de tempo, recai sobre a .Net Framework.

2) Diagramas

- Aqui apresentamos os diagramas das 2 interacções e-learning escolhidas de acordo com o resultado do questionário apresentado anteriormente.

No primeiro caso o aluno poderá pedir ao docente (através de um aviso) para que se ligue ao seu terminal para possa ver o seu trabalho e por exemplo corrigir algum erro, o docente ao ligar-se automaticamente bloqueia as acções do terminal do aluno, podendo de seguida editar ou simplesmente comentar o trabalho que o aluno está a desenvolver, desconectando-se de seguida permitindo assim ao aluno continuar o trabalho. O docente poderá também ligar-se a um qualquer terminal que tenha sessão iniciada sem que tenha sido pedido por um aluno, através da consulta de uma listagem dos terminais que tem sessão iniciada e escolhendo a qual se quer ligar.

No segundo caso, o docente poderá exibir um vídeo ou powerpoint para todos os terminais dos alunos com sessão iniciada, para isso basta aceder à listagem de todos estes terminais, escolher a quais se quer conectar, automaticamente bloqueando qualquer acção da parte dos alunos e iniciando depois a projecção.

i. Use Case das interacções e-learning

- Ligação remota ao terminal de um aluno para esclarecimento de duvidas ou verificar o trabalho que um aluno esteja a desenvolver

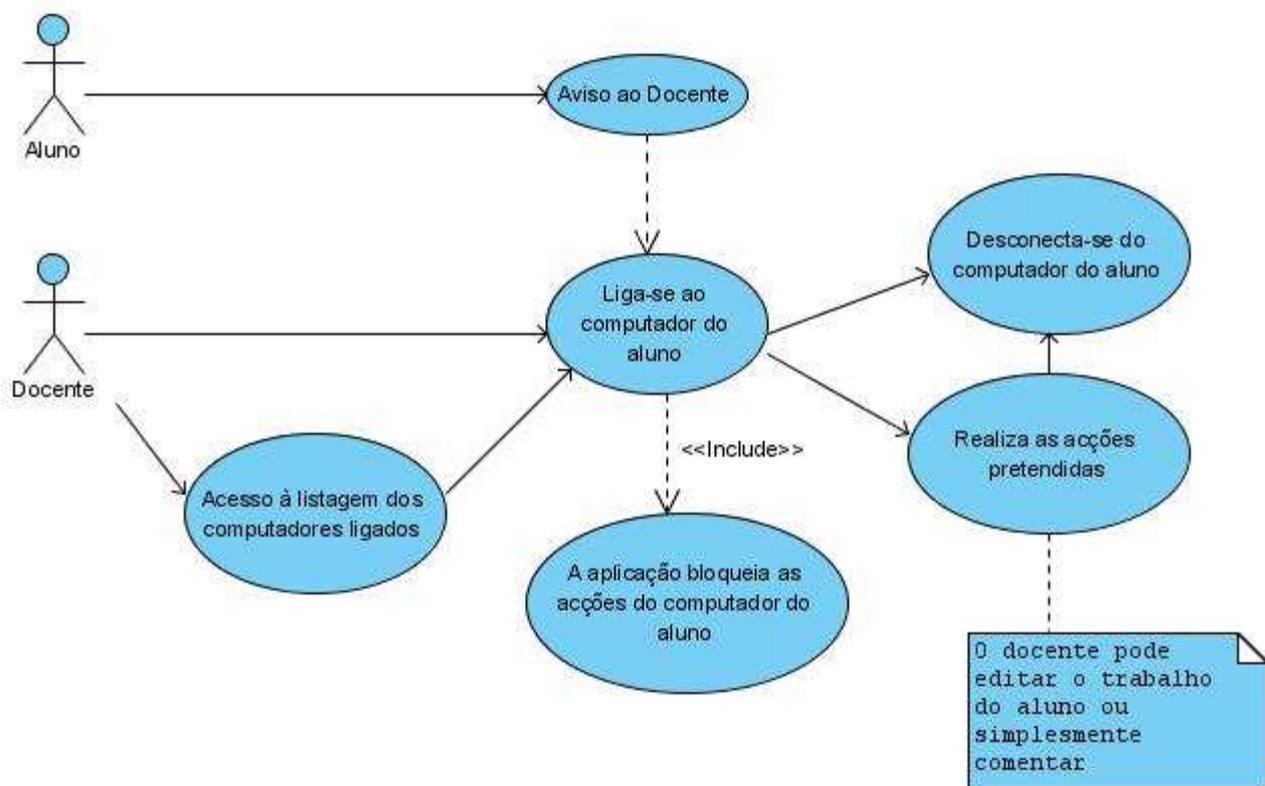


Figura 6.1 - Use Case Ligação Remota

- Apresentações do Ecrã do professor remotamente para os terminais dos alunos

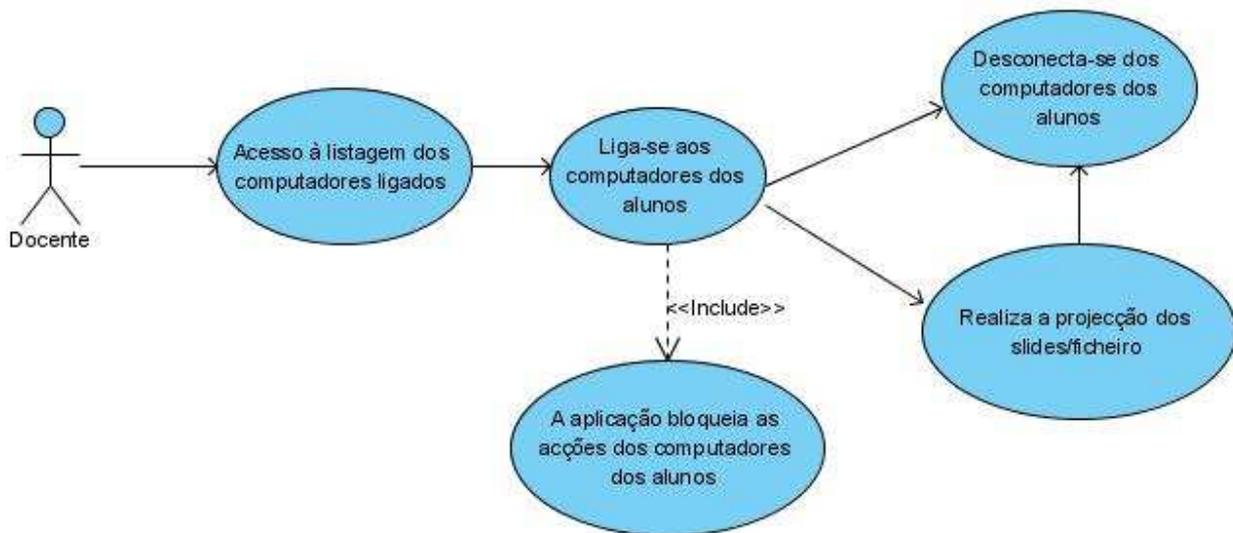


Figura 6.2 - Use Case Apresentações. Remotas

ii. Diagramas de Actividades

- Ligação remota ao terminal de um aluno para esclarecimento de duvidas ou verificar o trabalho que um aluno esteja a desenvolver

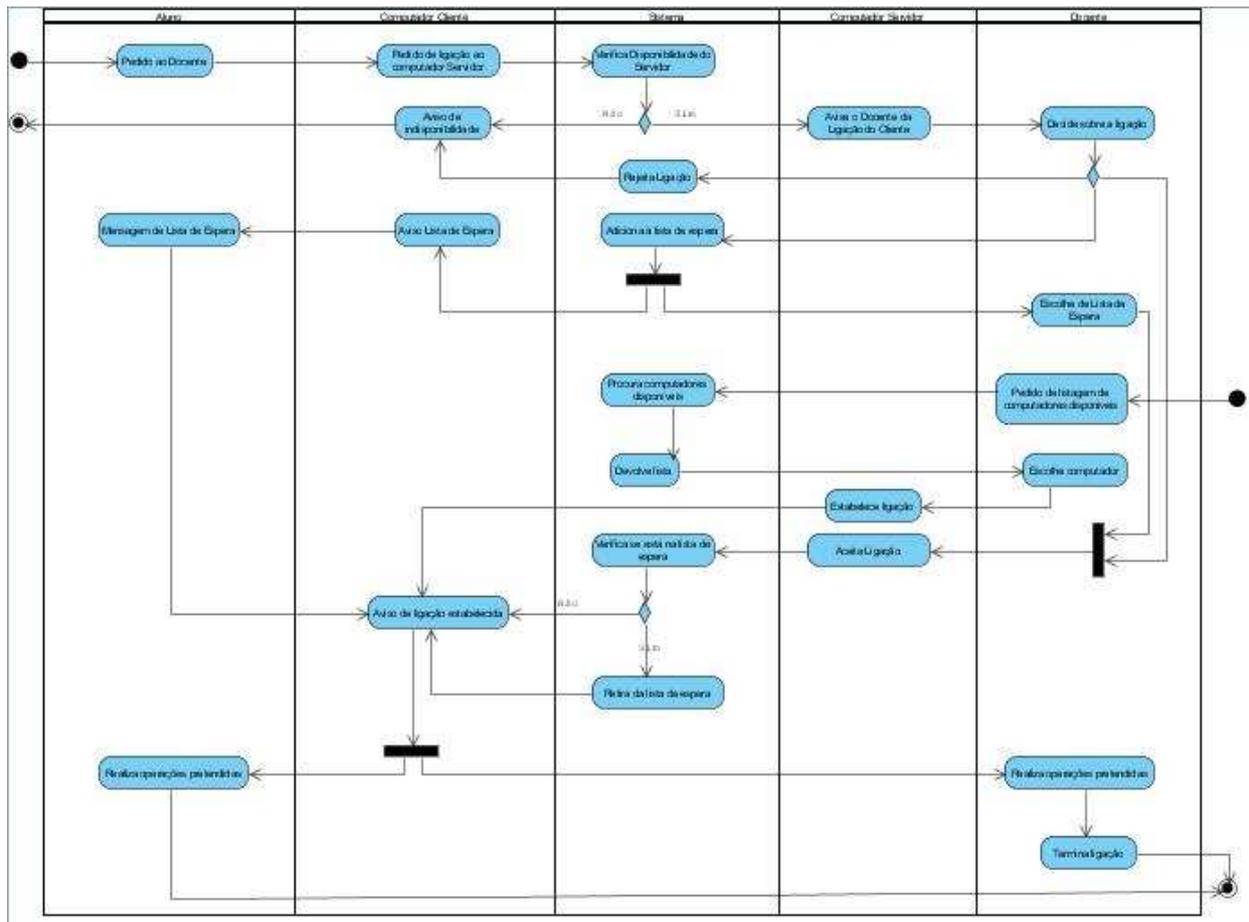


Figura 6.3 - Diagrama de Actividades da Ligação Remota

- Apresentações de vídeo/slides/som remotamente para os terminais dos alunos

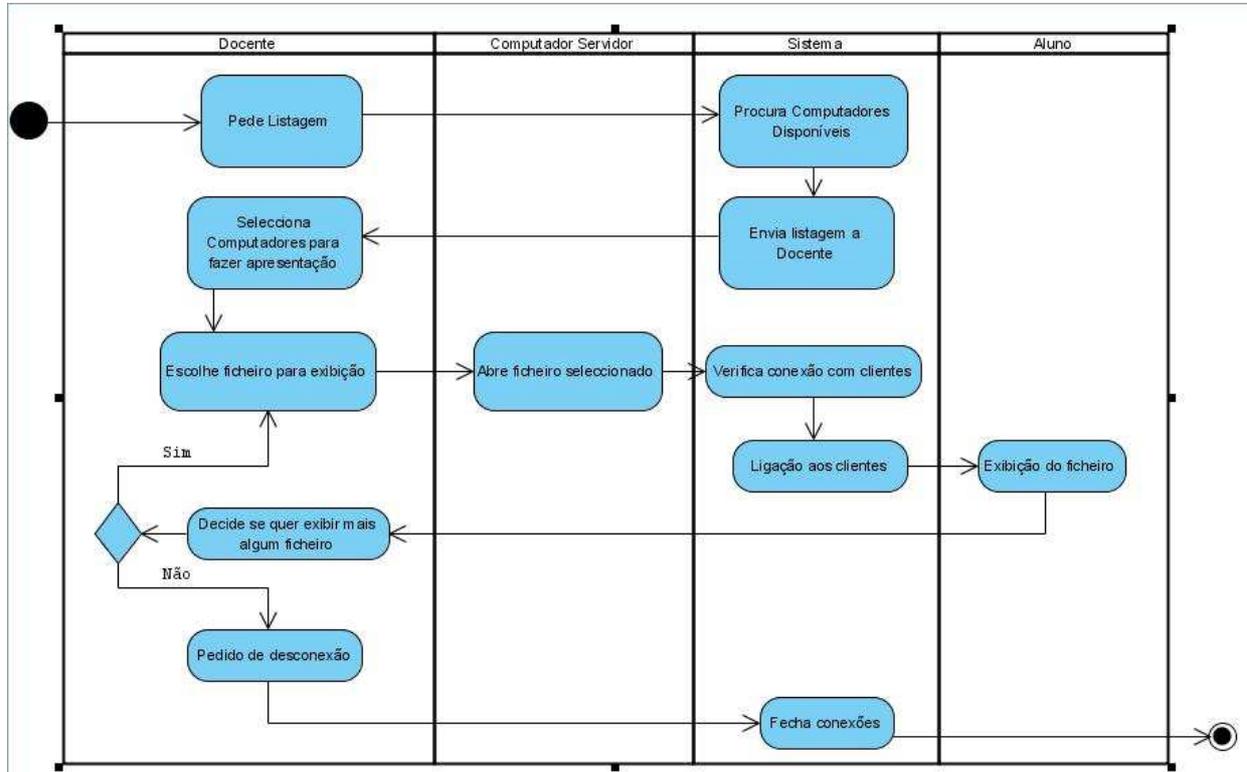


Figura 6.4 - Diagrama de Actividades de Apresentações Remotas

iii. Arquitectura

- Arquitectura geral da rede na sala de aula

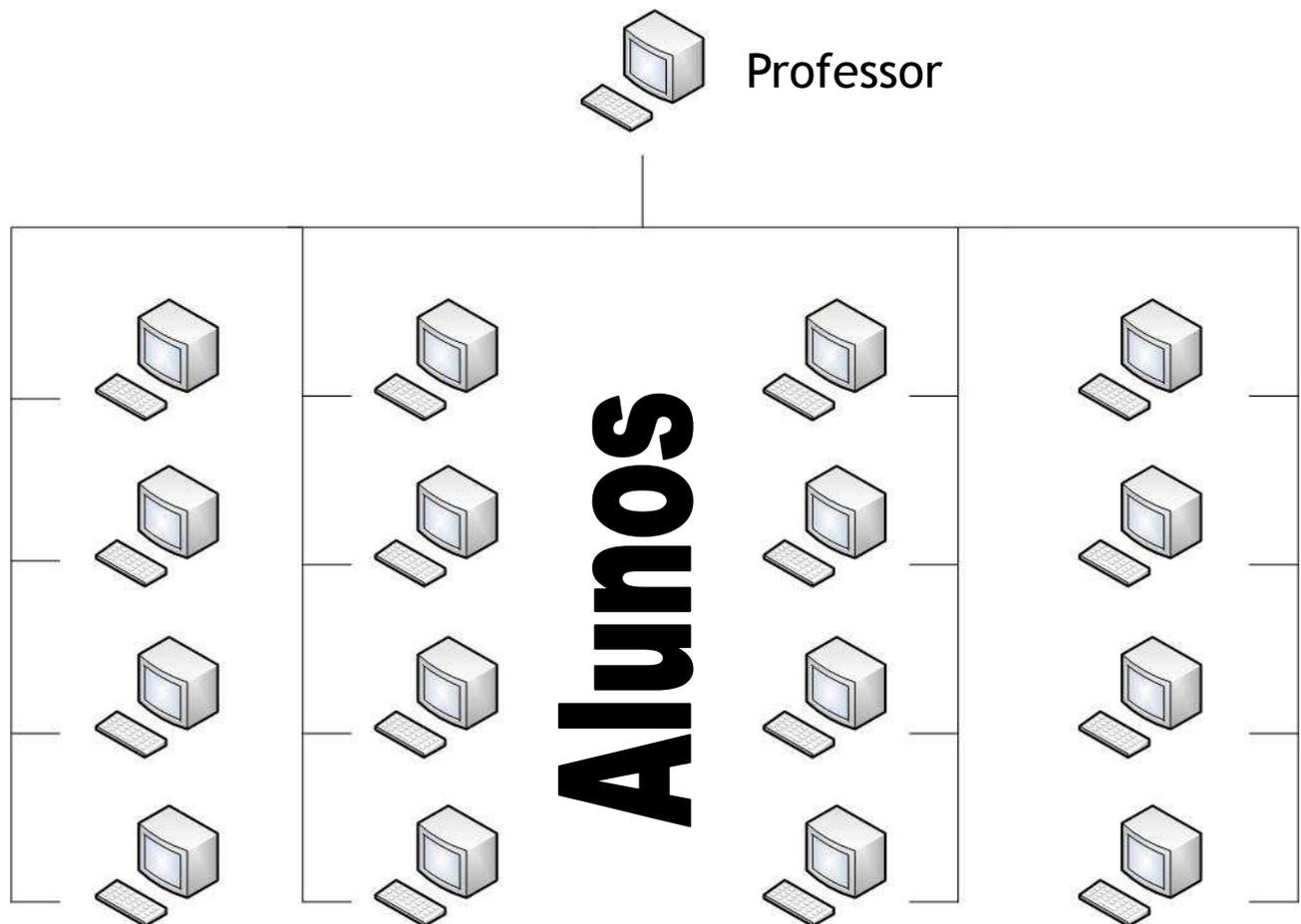


Figura 6.5 - Arquitectura Geral da Rede na Sala de Aula

- Arquitectura da ligação remota ao terminal de um aluno para esclarecimento de duvidas ou verificar o trabalho que um aluno esteja a desenvolver.

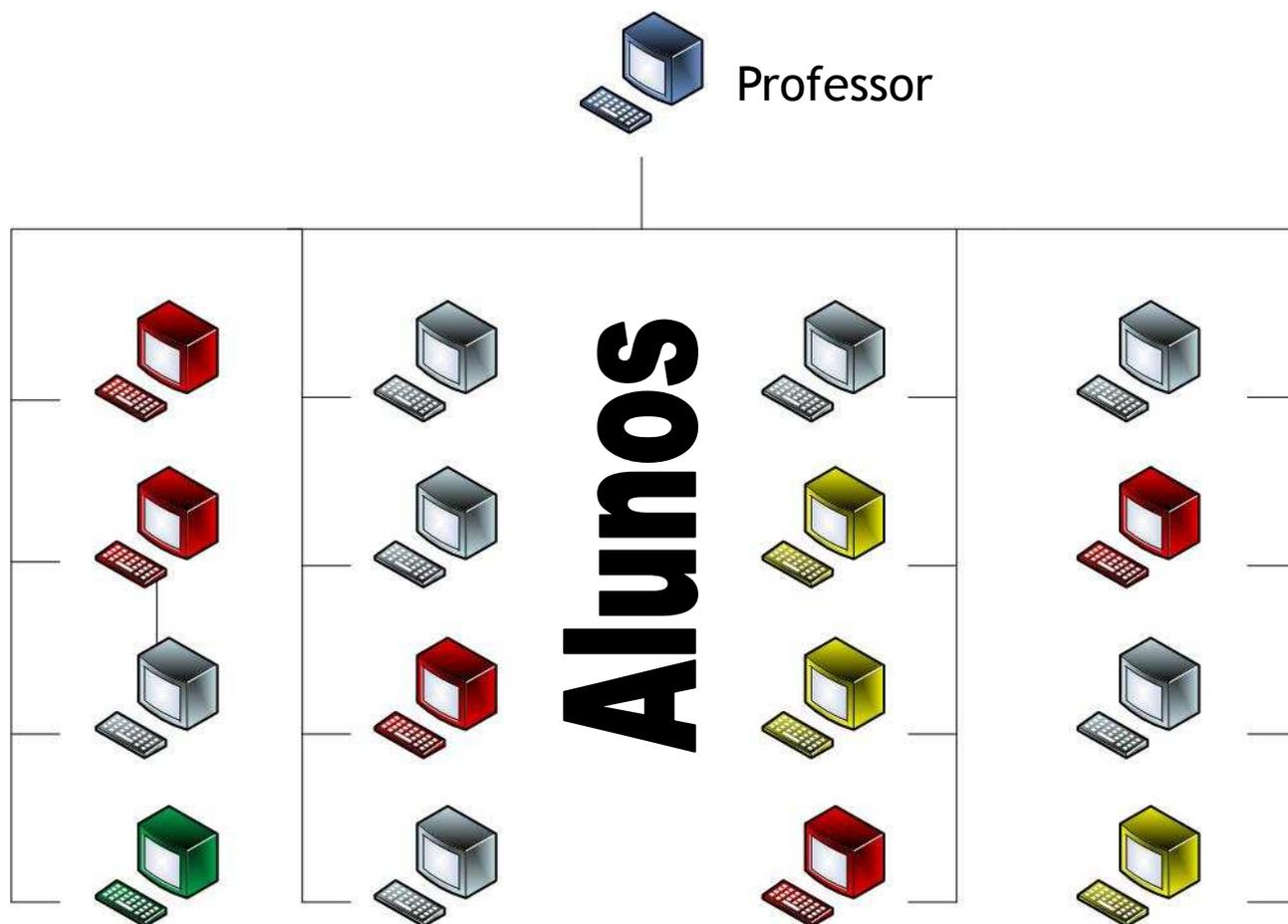


Figura 6.6 - Arquitectura da Ligação Remota

-  Aluno conectado com o professor
-  Aluno em lista de espera
-  Aluno com sessão iniciada
-  Terminal sem sessão iniciada

- Arquitectura de apresentações de vídeo/slides/som remotamente para os terminais dos alunos.

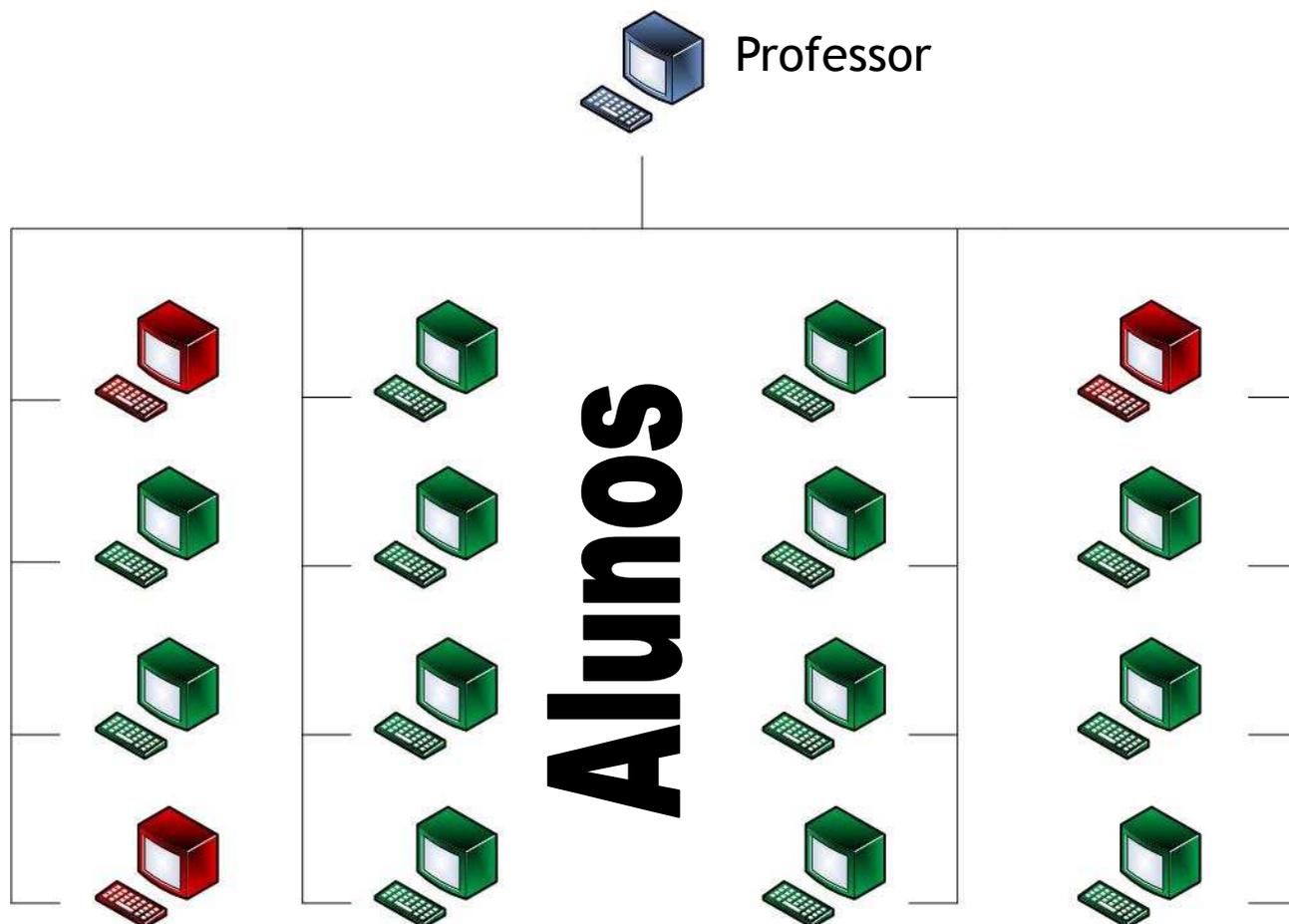


Figura 6.7 - Arquitectura de Apresentações Remotas

 Aluno conectado com o professor

 Terminal sem sessão iniciada

7.Método

1)Abordagem

Na abordagem ao tema de e-learning rapidamente chegámos à conclusão que a nossa abordagem ao protótipo seria fundamentalmente baseada em redes de computadores, isto tendo em vista os resultados obtidos na análise. Por isso a nossa abordagem inicial ao protótipo foi pesquisar dentro da tecnologia escolhida soluções que nos permitissem seguir o que foi proposto na análise da forma mais eficaz.

Dividimos a nossa solução em três campos:

- Gestão de listas de espera.
- Plataforma de ajuda do professor ao aluno.
- Plataforma de difusão do ecrã do terminal do professor para os alunos.

Gestão de listas de espera

Optamos por um método simples, que consiste em guardar os dados do terminal do aluno no terminal do professor. Pensámos em usar um motor de base de dados para o efeito mas chegámos à conclusão que seria demasiado “ pesado” para o problema em questão contudo o protótipo está feito de forma a implementar essa possibilidade, pois estamos a usar classes genéricas. Esta foi a área menos complexa de implementar.

Plataforma de ajuda do professor ao aluno

Nesta área existem duas nuances, primeiro o pedido de ajuda do aluno ao professor implica que o terminal do aluno seja o cliente e o terminal do professor o servidor, enquanto a ajuda remota do professor ao aluno implica o oposto, ou seja, o cliente será o terminal do professor e o servidor o terminal do aluno.

A nossa abordagem ao primeiro problema foi logicamente a utilização de sockets, ou seja, criávamos dois socket's do tipo TCP (Transmission Control Protocol), para garantir a entrega do pedido, um no terminal do aluno para o envio do pedido e outro no terminal do professor que ficava à escuta dos vários pedidos, a mensagem enviada pelo socket era a configuração do terminal do aluno que enviou o pedido. Avisando o professor do pedido feito e o aluno de que o pedido foi efectuado com sucesso. Esta abordagem foi descartada mais tarde não pela existência de problemas técnicos mas devido à uniformização do protótipo (explicado mais abaixo).

A abordagem ao segundo problema foi muito mais complexa pois requeria o controlo do terminal do aluno por parte do professor. Nesta área optámos por usar um protocolo da Microsoft, o RDP (Remote Desktop Protocol), para o efeito usámos duas DLL's da Microsoft que permitem o controlo remoto (AxInterop.MSTSCLib.dll, Interop.MSTSCLib.dll), apesar do teste ter sido satisfatório, observámos que o Sistema Operativo do terminal ao qual nos ligávamos bloqueava a sessão do utilizador, não permitindo a este ver as acções do professor no seu terminal. Pela pesquisa efectuada não encontramos solução para esta questão, pelo que tivemos de descartar esta opção. Tentámos também o mesmo método que utilizámos para o pedido de ajuda do aluno, explicado anteriormente mas rapidamente chegámos à conclusão que não era viável pois era complexo invocar métodos a partir do terminal do professor no terminal do aluno. A nossa opção final estará explicada no capítulo *Protótipo Final*.

Plataforma de difusão do ecrã do terminal do professor para os alunos

Neste ponto houve um desvio sobre o que foi proposto na análise. Inicialmente propusemos a difusão de vídeo e slides a partir do terminal do professor para o terminal do aluno, contudo, a complexidade que se levantou nesta solução aliada à sua pouca utilidade, optámos por uma solução diferente que consiste na difusão do ambiente de trabalho do professor, permitindo assim uma muito maior utilidade, pois possibilita uma flexibilidade muito superior não havendo limitações do que é transmitido. A nossa abordagem a este ponto, consistia na utilização do mesmo método que acabámos por utilizar no ponto anterior, contudo a utilização do protocolo TCP para uma difusão (multicast/broadcast) não é aconselhável porque sobrecarrega em demasia a rede, tornando esta opção evitável, porém importa salientar que a utilização do protocolo UDP (User Datagram Protocol) para multicast não garante a chegada total das imagens ao terminal do aluno, ou seja como o protocolo não garante a ordem de chegada nem a própria chegada em si e como tem um buffer bastante limitado teríamos de dividir cada imagem em centenas de pacotes o que tornaria este método totalmente descartável. Depois de alguma pesquisa encontramos um protocolo recente, PGM (Pragmatic General Multicast Protocol), este protocolo permite a difusão por multicast em que os pacotes são transmitidos com ordem, não duplicados e a sua entrega. Até à data da entrega do relatório não nos foi possível implementar esta solução.

2) Protótipo Final

O protótipo final consiste numa aplicação que permite ao professor ligar-se remotamente a um terminal previamente escolhido a partir de uma lista de espera, bloqueando o teclado e rato do terminal escolhido, esta funcionalidade só está disponível no Windows XP, podendo visualizar o ecrã deste mesmo terminal e realizar acções através do seu próprio teclado e rato no terminal do aluno.

No início da aplicação são carregadas as definições do terminal a partir de um ficheiro XML que é convertido para a classe `MachineConfiguration`. É também carregada a definição do Master. Depois de ter as configurações iniciadas são criados dois canais (um servidor e um cliente) tanto no Master como nos Slaves.

i. Criação do canal servidor

É criado um objecto que implementa um canal servidor para chamadas remotas que utilizam o protocolo TCP para transmitir mensagens.

A classe usada para este efeito é `TcpServerChannel`. No construtor desta classe são passadas propriedades anteriormente definidas, no nosso caso usámos o nome do servidor, o porto, e o timeout. Importa referir também que todas as definições de segurança e de autorização são definidas nestas propriedades, é também passado no construtor um objecto do tipo `BinaryServerFormatterSinkProvider` que vai permitir que todos os canais remotos usem uma interface do tipo `IServerChannelSinkProvider` para a troca de mensagens. É necessário atribuir o tipo de objecto que vai ser trocado nas mensagens, no nosso caso será um objecto do tipo `ComputerControl`, a sua utilidade será explicada mais à frente.

ii. Criação do canal cliente

É criado um objecto que implementa um canal cliente para chamadas remotas que utilizam o protocolo TCP para transmitir mensagens.

A classe usada para este efeito é `TcpClientChannel`. No construtor desta classe são passadas propriedades anteriormente definidas, no nosso caso usámos o `retrycount` que controla as tentativas de conexão, `timeout` e `socketCacheTimeout`. Importa referir também que todas as definições de segurança e de autorização são definidas nestas propriedades.

iii. Registo dos canais

Utilizando a classe estática ChannelServices registamos os canais, cliente e servidor. Este serviço permite que os canais registados sejam usados por aplicações, no nosso caso será apenas usado pela nossa aplicação.

iv. Classe ComputerControl

Esta classe será usada para as comunicações, contém métodos e propriedades que nos servirão para serem invocadas no servidor.

Nesta classe estão incluídos 7 métodos:

- BlockInput() - Bloqueia as acções do terminal quando é invocado.
- KeyboardInput(int) - Envia um comando de teclado para o terminal.
- MouseClick(Point) - Envia a localização e um click simples do rato.
- MouseMove(Point) - Envia a localização do rato para o terminal.
- SendMessage(string) - Envia uma mensagem genérica para o terminal.
- SendNotify(string,args) - Envia notificações para serem apresentadas no ícone do terminal.
- SendSlaveConfiguration(string) - Envia a configuração do terminal slave para o pedido da lista de espera.

Nesta classe estão incluídas 3 propriedades:

- Capture - retorna a imagem do ambiente de trabalho do terminal quando invocado.
- Config - retorna a configuração do terminal.
- DesktopSize - retorna o tamanho do ecrã de trabalho do terminal.

v. Conexão entre cliente e servidor

Depois de estarem registados os canais, efectua-se a ligação entre cliente e servidor. O cliente liga-se ao servidor através da chamada do método Connect, se a ligação for efectuada com sucesso é retornada um objecto do tipo que foi registado pelo servidor, no nosso caso um objecto do tipo ServerControl, a partir deste ponto o cliente pode invocar métodos e propriedades que estão contidas na classe e esta actuará no servidor.

vi. Pedido de conexão por parte do aluno

O pedido é feito através da invocação do método `SendSlaveConfiguration` que passará uma mensagem do aluno para o professor com a seguinte informação:

- Nome do computador
- IP do computador
- Nome de Utilizador
- Domínio

Se a mensagem foi recebida com sucesso o Master, adiciona o pedido à lista de espera, guardando o pedido em memória (`DataTable`) e num ficheiro XML (ou base de dados) e envia outra mensagem de volta confirmando a recepção. É dada a informação ao aluno que a operação ocorreu com sucesso.

vii. Gestão da lista de espera

Caso exista algum pedido em lista de espera é apresentada uma grelha ao professor em que figura todos os pedidos efectuados. Esta grelha é carregada a partir do objecto (`DataTable`) em memória permitindo assim ao professor a possibilidade de escolha sobre qual dos pedidos irá atender primeiro. É também permitido apagar registos da lista individualmente ou todos. Sendo que tanto a grelha como o `DataTable` e o ficheiro XML são actualizados cada vez que existe uma acção efectuada pelo utilizador.

A estrutura da `DataTable` é feita com base na informação recebida na mensagem transmitida pelo aluno, como referido no ponto anterior.

viii. API's do Windows

Na captura de ecrã Utilizamos uma chamada ao método do Windows `GetDesktopWindow()` que está incluída na “`user32.dll`” que nos permite capturar o Desktop presente no terminal onde o método é invocado, este retorna uma imagem e o tamanho da mesma.

Para bloquear o teclado e o rato do aluno utilizamos uma chamada ao método `BlockInput(bool)` que está incluída na “`user32.dll`” e que conforme o parâmetro que é passado bloqueia ou desbloqueia o teclado e rato.

Para lançar eventos de teclado e rato usamos os métodos `keybd_event` e `mouse_event`, que estão também incluídas na “`user32.dll`”, são passados como parâmetros os eventos realizados pelo Master.

ix. Controlo Remoto

Quando a conexão é efectuada é enviado pelo Master uma mensagem para o bloqueio das acções do utilizador do Slave (rato e teclado).

A partir deste momento o Master vai recebendo as capturas de ecrã enviadas pelo Slave, podendo invocar alguns métodos/eventos que estão incluídos na classe ComputerControl descrita acima e que lhe vão permitir controlar as acções do teclado e o rato.

x. Difusão do ecrã de trabalho do Master

No protótipo não vai incluída a difusão do ecrã de trabalho do terminal do Master, porque o método que mais se adequa descrito na *Abordagem* a este tema não nos foi possível implementar com a versão 2.0 da Microsoft Framework, contudo o código que usámos para o multicast e para o outro método semelhante ao controlo remoto vai incluído e comentado no código do protótipo final. O intuito desta acção é transmitir para os alunos o trabalho que o professor no seu terminal (ex: edição de código, visualização de documentos, slides, etc.).

xi. Diagrama de Classes

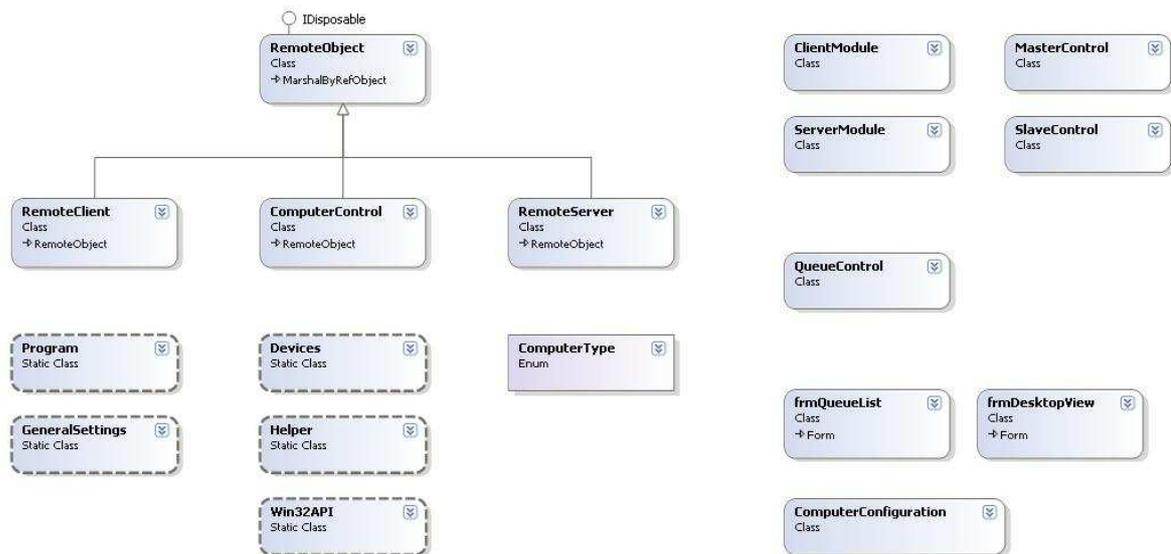


Figura 7.1 - Diagrama de Classes

8. Resultados

Dividindo igualmente os resultados obtidos pelas três áreas distintas do nosso projecto, obtivemos os seguintes resultados ao longo do nosso trabalho:

1) Gestão de listas de espera.

Nesta área os resultados obtidos foram os esperados, toda a gestão da lista de espera é controlada pelo professor a partir de uma tabela que mostra todos os pedidos efectuados pelos alunos, o professor pode recusar pedidos, retirando-os da lista de espera, pode removê-los individualmente ou todos. Pode ainda ligar-se remotamente ao terminal escolhido.

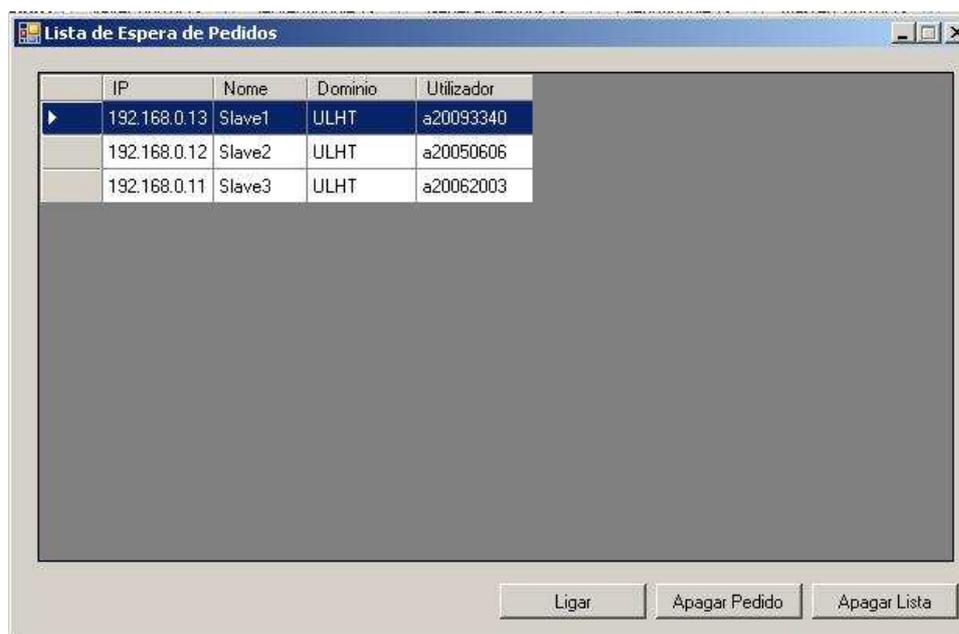


Figura 8.1 - Lista de Espera

2) Plataforma de ajuda do professor ao aluno

Como foi explicado mais acima na *Abordagem* a esta área, esta divide-se em duas partes. O pedido do aluno e o controlo remoto por parte do professor. Quando iniciámos o desenvolvimento do protótipo e como foi explicado no início deste capítulo estávamos a utilizar sockets, os resultados foram satisfatório pois a informação do terminal do aluno era correctamente recebida pelo professor e era mostrado correctamente na lista de espera, apesar dos resultados serem positivos optámos por uniformizar a nossa aplicação e o pedido passou a ser feito com uma invocação remota de um método no terminal do professor, o resultado obtido foi igualmente satisfatório com o ganho de termos eliminado código desnecessário e conseqüentemente mais operações na rede. No controlo remoto tivemos várias dificuldades, a primeira delas foi a configuração do tamanho do ecrã, optámos por encolher a imagem recebida (caso seja necessário), para ser completamente visível e para ter um maior controlo sobre os eventos do rato. Se a sobrecarga na rede for elevada é possível alguma perda de rapidez de processamento ao mostrar a imagem, a causa

deve-se principalmente à utilização do protocolo TCP, mas importa salientar que desta forma garantimos a fiabilidade da ligação. Tivemos também dificuldades no controlo das acções do rato, isto deve-se à posição relativa no ecrã do terminal do professor ser diferente da do terminal do aluno, o resultado que obtivemos tem um ligeiro erro de posição vertical mas consideramos ser irrelevante dado a ser uma diferença mínima. Ainda não são controladas todas as acções do rato.

Porém o maior problema surgiu com o controlo das teclas especiais do teclado (CTRL, ALT) que ao serem usadas obtivemos um descontrolo da sua utilização no terminal remoto. Toda a utilização que não usasse as teclas especiais foi feita com sucesso.

3) Plataforma de difusão do ecrã do terminal do professor para os alunos.

Neste ponto, dos dois métodos que usámos, os resultados não foram satisfatórios. O primeiro método utilizado foi o do multicast por UDP, o maior problema neste método foi a limitação que o protocolo UDP tem no tamanho do buffer de envio (cerca de 1500 bytes) e sabendo que este protocolo não garante a entrega nem a ordem desta tornava-se impossível de transformar a imagem (cerca 3 milhões de bytes) num array de bytes, partindo o array em vários datagramas e depois de enviar os datagramas garantir que nos terminais conseguíssemos juntar os vários datagramas na ordem correcta sem a perda de nenhum e transformar o array na imagem inicial. Neste caso a única solução seria garantir com o protocolo TCP o controlo do envio, que não era de todo benéfico pois o objectivo era evitar usar este mesmo protocolo. O segundo método era utilizar o mesmo que o controlo remoto, ou seja, os terminais dos alunos invocariam o método para capturar o ecrã do ambiente de trabalho do professor e mostrá-lo-ia nos seus terminais. Como dissemos atrás a utilização do protocolo TCP neste ponto era evitável, pois sobrecarregaria a rede excessivamente.

9. Conclusões e trabalho futuro

O principal objectivo quando desenvolvemos o protótipo foi minimizar a complexidade para o utilizador, bem como a criação de uma ferramenta que possa ser útil com desenvolvimentos futuros. Acreditamos que o objectivo a que nos propusemos e apesar de algumas lacunas ainda existentes no trabalho, algumas devido à limitação técnica, na sua globalidade foi atingido. No fim apesar da aparente simplicidade do trabalho, mostrámos que é possível existir uma plataforma de e-learning simples, funcional e que vá de encontro às necessidades dos docentes e alunos, partindo do pressuposto que a aplicação final é apenas um protótipo funcional demonstrativo.

Melhorias a considerar:

1) Lista de espera

Neste ponto era interessante haver uma ligação a um motor de base de dados com entidades definidas que contenham a informação das salas de aulas, nomeadamente dos terminais existentes em cada sala, bem como definir quais os terminais da sala pertencentes a alunos e professor.

2) Controlo remoto

Como trabalho futuro o controlo dos inputs de teclado e rato deve ser expandido a todas as acções, no teclado, o controlo de teclas especiais. No rato controla os duplos clicks, o click do botão direito e o arrastamento. Sugeríamos ainda que as imagens fossem transmitidas de forma a não se utilizar o protocolo TCP e como referimos anteriormente a utilização do protocolo PGM.

3) Difusão de dados do Master

Como existiram dificuldades ao nível técnico no envio de som e vídeos, a proposta aqui iria neste sentido, e usar o protocolo UDP para este efeito.

4) Outras considerações

Apesar de termos limitado o nosso trabalho às duas áreas mais escolhidas pelos docentes através do questionário elaborado, pensamos que seria importante implementar algumas das propostas que tiveram também alguma aceitação.

10. Bibliografia

Ingo Rammer, *Advanced .NET Remoting (C# Edition)*, Apress, 2002

William Horton & Katherine Horton, *E-learning Tools and Technologies: A consumer's guide for trainers, teachers, educators, and instructional designers*, Wiley Publishing, 2003

Andrew Tanenbaum, *Computer Networks, Fourth Edition*, Prentice Hall, 2002

José Rogado, *Slides Redes de Computadores*, ULHT, 2009

José Rogado, *Slides Computação Distribuída*, ULHT, 2009

José Rogado, *Slides Complementos de Redes*, ULHT, 2009

Referências web

How to Use .Net Remoting Using C#? - <http://www.csharp-help.com>

Chapter 6 Remoting - http://www.mastercsharp.com/chapter_06.htm

An Introduction to .NET remoting - <http://www.devarticles.com/c/a/C-Sharp/An-Introduction-To-.NET-Remoting/>

Multicast - <http://www.codeproject.com/KB/IP/multicast.aspx>

MSDN Library - <http://msdn.microsoft.com>

Anexos no CD

- Código Fonte
- Manual de Utilização
- Relatório Final