

INSTITUTO SUPERIOR TÉCNICO

Licenciatura em Engenharia Informática e de Computadores

Trabalho Final de Curso

Ferramentas de Apoio ao
Planeamento e Condução de
Reuniões Electrónicas

TFC - 108

Prof. Responsável: Pedro Antunes

Aluno nº 38493: Nuno Barrocas

Introdução.....	4
1.1. Enquadramento	4
A Reunião	4
Pessoas	4
Sistema Actual.....	5
1.2. O Problema	6
Motivo.....	6
1.3. Objectivos	7
Requisitos Fundamentais	7
Requisitos para a Interface Homem-Máquina.....	7
1.4. O Método - <i>Brainstorming</i>	8
Brainstorm – Procura de Ideias	8
O Método.....	8
Regras e Princípios	9
1.5. Tecnologia.....	11
1.6. Restrições.....	11
Software	11
Hardware	11
Características do Utilizador	11
<i>Modelo Conceptual</i>	12
Variantes e Estímulos	12
<i>Brainstorm</i> identificado	12
<i>Brainstorm</i> anónimo	13
<i>Brainstorm</i> com categorias.....	14
<i>Brainstorm</i> com limite de tempo	14
<i>Brainstorm</i> com RELÓGIO	14
User Ranking.....	15
Número total de ideias introduzidas	15
Estímulo Visual.....	15
Reordenação Automática de Ideias.....	15
Uso de Estilos.....	16
<i>Implementação</i>.....	17
1. Metodologia.....	17
Definição da Metodologia de Desenvolvimento.....	17
Fases do projecto.....	17
2. Arquitectura.....	18
Esquema.....	18
Funcionalidade	19
Motivos	19
3. Servidor.....	21
Arquitectura do Software.....	21
Esquema Total	25
Funcionamento	26
4. Cliente – Configuração	27
Elementos da Interface Homem-Máquina.....	27
Arquitectura do Software.....	28
Funcionamento	31
5. Cliente – <i>Brainstorm</i> normal.....	34
Elementos da Interface Homem-Máquina.....	34
Arquitectura do Software.....	34

Funcionamento	38
6. Cliente – Brainstorm com Categorias	39
Elementos da Interface Homem-Máquina	39
Funcionamento	39
7. Servlet - ServerServlet	40
Elementos da Interface Homem-Máquina	40
Funcionamento	41
Modo Utilizador	41
Modo Administrador	41
<i>Conclusão</i>	42
<i>Bibliografia</i>	43

Introdução

1.1. Enquadramento

Este trabalho surge no âmbito de um projecto de experimentação/exploração de uma sala de reuniões electrónica existente no ISCTE. O *software* já existente apresenta limitações fundamentais que se pretende colmatar com o resultado deste trabalho.

A Reunião

Uma reunião é composta por uma série de fases, que dependem dos objectivos. Fases essas que são planeadas e definidas numa fase anterior à reunião.

Uma reunião é constituída por um grupo de participantes e por uma pessoa que coordena a reunião – Facilitador. É da competência do facilitador efectuar o plano, lançar as diversas fases e apresentar os resultados da reunião.

Pessoas

Esta aplicação é dirigida a pessoas que necessitam de analisar problemas, identificar soluções e tomar decisões apoiadas por *software*. As vantagens deste tipo de aplicação são o planeamento/organização da reunião o que implica um estudo prévio do problema a ser resolvido (causas, definição do problema, etc); um aumento da produtividade devido ao uso de modelos de apoio à decisão em grupo (grupos participativos); a possibilidade de anonimato resolve problemas de falta de confiança, medos perante superiores e medo de críticas o que faz com que a participação seja muito superior em relação ao caso em que não existe anonimato. Todas estas vantagens provocam um aumento da participação o que implica um aumento da qualidade da reunião. A grande desvantagem deste tipo de processo é o tempo, mas não será este tempo compensado pelos bons resultados ?

Sistema Actual

Existem muitos modelos para a resolução de problemas em grupo. Os modelos podem-se dividir em três fases: **Procura de Informação** – nesta fase o objectivo é tentar encontrar o número máximo de opiniões, ideias, soluções, causas, etc para um problema; **Formação e Avaliação de Propostas** – nesta fase são criadas propostas, a partir da fase anterior, e avaliadas; **Exploração e Tomada de Decisão** – aqui são exploradas as propostas e é tomada uma decisão.

Existem muitos métodos e técnicas para cada uma destas fases, os vários métodos e técnicas existem para colmatar os problemas que cada fase enfrenta.

O sistema actual consiste numa **ferramenta de apoio ao planeamento**, que precede a reunião, numa **ferramenta de exploração** (*Mind Map*), e necessita de ferramentas de apoio ao facilitador. **As ferramentas de apoio ao facilitador** necessárias são:

- **Recolha de Informação**
- **Votação**
- **Avaliação/Consolidação**

Este Trabalho Final de Curso tem como objectivo implementar a Ferramenta de apoio ao facilitador – Recolha de Informação.

1.2. O Problema

O problema que se pretende resolver é o da recolha de informação em reuniões.

Motivo

Uma reunião começa com uma serie de pessoas sentadas à volta de uma mesa com a intenção de resolver um ou mais problemas. Após o problema a resolver ser exposto os participantes começam a discuti-lo, cada um dos participantes irá agarrar-se à sua experiência/ideologia para defender a sua posição, como os participantes podem ter experiências/ideologias bastante diferentes normalmente a reunião atinge pontos de tensão bastantes elevados e por esta linha dificilmente se chega a uma solução ou pelo menos a uma boa solução. Por outro lado a reunião pode ser completamente dominada por um bom orador ou uma pessoa com muita influência social ou hierárquica, pode ainda acontecer que as pessoas estejam pouco motivadas para a reunião e se dispersem completamente da razão de ali se encontrarem, passando o objectivo principal para chegar a um acordo bom ou mau. Em qualquer dos casos faz com que a reunião não seja produtiva. Assim sendo está perfeitamente bem identificada a necessidade de resolver este problema. A solução é adoptar técnicas de condução de reuniões. Entre as técnicas existentes a mais utilizada e a que apresenta melhores resultados mundialmente é o *brainstorming*.

Uma reunião que use técnicas de *brainstorming*, é bastante mais fácil de gerir, pois existem procedimentos a seguir e objectivos a atingir, e evita alguns dos problemas atrás descritos pela cultura que tem que ser inculcada nos participantes (ver abaixo regras e princípios). Mas a maior parte dos problemas mantém-se apesar dos resultados poderem ser substancialmente melhores.

Está de novo identificado um problema e logo uma necessidade de recorrer a novas técnicas para a resolução deste. Tem-se vindo a fazer uma serie de estudos na área de *brainstorming* electrónico e os resultados são bastante positivos, tanto a nível da gestão e coordenação da reunião como da produtividade da mesma.

Este trabalho surge como resposta à necessidade acima descrita.

1.3. Objectivos

O objectivo deste TFC é dar suporte a um processo de recolha de informação.

Requisitos Fundamentais

A implementação irá ser feita de modo a ser integrada com o *software* já existente. Cada técnica/ferramenta funciona independentemente.

É importante ter várias técnicas de recolha de informação que dependem da natureza dos participantes e do problema a resolver. Factores importantes a considerar são a possibilidade de anonimato ou identificação dos participantes, criação de *logs* que permitam fazer análises posteriores ao processo de recolha de informação tais como relacionar no tempo as participações dos vários intervenientes, quem teve melhores ideias, quem teve mais ideias, quais as ideias que provavelmente desencadearam determinadas ideias e métodos de processamento automático (ordenação e visualização) da lista de ideias. O método e técnicas é descrito abaixo.

Requisitos para a Interface Homem-Máquina

Este *software* tem que poder ser utilizado por pessoas com pouca experiência informática. É facilitada a aprendizagem sobre a aplicação através de interfaces cuidadosamente criadas, sendo de realçar a **coerência** entre todos os aspectos gráficos; disposição dos elementos – de forma intuitiva e lógica, áreas idênticas sempre nas mesmas zonas (ex. *toolbars*, menus, áreas informativas); divisão do espaço – com especial atenção ao não sobrecarregamento da quantidade de informação (só a informação necessária), zonas conceptualmente distintas em zonas distintas, zona de descanso para os olhos; uso de texto nos ícones e de *tooltips* para facilitar a compreensão das funções. Deste modo, a velocidade de aprendizagem é rápida e intuitiva o que motiva os utilizadores.

A interface e os cuidados tomados na sua construção irão sendo descritos ao longo do relatório para cada componente.

1.4. O Método - *Brainstorming*

Brainstorm – Procura de Ideias

O Método

O *Brainstorm* é o melhor método conhecido para geração de ideias. É um método em que um grupo tenta encontrar uma solução para um problema assumindo todas as ideias propostas pelo grupo. Alex Osborn, criador deste método em 1938, afirma que “uma pessoa normal consegue criar duas vezes mais ideias em grupo do que individualmente”. Este método só é eficiente se se seguirem algumas regras e princípios (ver próximo tópico).

O método é composto por um facilitador, que coordena a reunião, e por um grupo de participantes. Genericamente o método pode ser escrito ou verbal, verbalmente os participantes dão ideias, uma de cada vez, e o facilitador apontando-as num quadro; Por escrito, são colocadas folhas com o problema a resolver no cabeçalho em cima de uma mesa, de onde os participantes sempre que queiram adicionar uma ideia retiram uma das folhas, lêem as ideias que já lá se encontram e acrescentam a sua ou melhoram uma das já existentes.

As vantagens deste método são:

- Colaboração criativa em grupo é mais eficiente do que individualmente.
- Produz um grande número de ideias em relativamente pouco tempo.

As vantagens do método dado que é assistido por computador:

- Participação em paralelo em vez de ser sequencial.
- Redução da tensão dado que não existem conflitos nem verbais, nem gestuais.
- A leitura das ideias de outros estimula o pensamento.
- Todas as ideias ficam registadas.
- O domínio de personalidades fortes é eliminada.
- Decisões prematuras são evitadas devido a não existirem pressões.

Como o método irá ser assistido por computadores algumas das desvantagens desaparecem automaticamente, tais como o monopólio da sessão por alguns membros, a tendência natural

de algumas pessoas para infringir regras ou retardar o julgamento. A informalidade exigida pode ser difícil de controlar.

As principais desvantagens são:

- 1) O grupo muitas vezes irá estar mais interessado em chegar a um acordo do que a um bom resultado.
- 2) A natureza divergente do *brainstorming* encoraja o desenvolvimento de ideias espontâneas mas não estrutura o refinamento de ideias.
- 3) Precisa de pessoas relativamente qualificadas para resolver o problema.
- 4) A posição social de alguns participantes pode exercer pressões influenciando o rumo da discussão.

As desvantagens do método dado que é assistido por computador:

- Perde-se alguma espontaneidade.
- Clarificação verbal não existe.

Regras e Princípios

Existem dois princípios que servem como base à técnica:

- O **julgamento retardado** é para permitir à parte criativa da mente criar novas ideias sem se preocupar com o julgamento das mesmas, ou seja, no decorrer desta técnica não podem existir críticas às ideias de outros. O custo de criticar ideias não amadurecidas (espontâneas) é impedir essa ideia de se desenvolver. As ideias serão julgadas mais tarde, depois da geração das mesmas.
- A **quantidade gera qualidade**. Quanto mais ideias existirem para a solução de um problema melhor será a solução.

Existem quatro regras básicas a serem seguidas:

- **Criticar ideias é proibido.**
- **Incluir as ideias mais disparatadas.** Qualquer ideia mesmo a mais disparatada é ótima. É mais fácil excluir ideias que criá-las.
- **Quantidade.** Quanto mais ideias maior a probabilidade de se encontrar uma de qualidade.

- **Combinação e melhoramentos.** Agrupar ideias de modo a formar uma ideia mais elaborada ou melhorar as ideias já existentes.

Existem muitas variantes deste método, na secção seguinte são apresentadas as variantes implementadas.

1.5. Tecnologia

Foi usada a linguagem de programação Java, versão 1.2. A escolha da linguagem foi feita por motivos de compatibilidade com o *software* já existente e por a aplicação poder ser lançada de qualquer *browser* que a suporte, isto faz com seja independente da plataforma.

As interfaces com o utilizador foram implementadas usando o JFC (*Java Foundation Classes*), que é um dos componentes do JDK 1.2, também conhecido por *swing*. Este pacote contém praticamente todos os tipos e variantes de objectos gráficos para a criação de uma interface com o utilizador.

A comunicação entre servidor e cliente é feita usando RMI (*Remote Method Invocation*) e html via servidor de web.

As únicas ferramentas de desenvolvimento utilizadas para além das necessárias para a compilação e depuração dos programas foi um editor de texto e um *browser*.

1.6. Restrições

Software

Só é necessário um *browser* que suporte aplicações em Java 1.2 ou no qual se possa usar o JavaPlugin 1.2.

Browsers: Netscape 4.* ; Internet Explorer 4.* ; Hot-Java

Sistemas Operativos: Todos para os quais exista suporte para a *Java Virtual Machine*

Hardware

Qualquer plataforma para a qual exista *Java virtual-Machine*.

Características do Utilizador

A aplicação é dirigida a utilizadores com muito pouca experiência em sistemas informáticos. Os utilizadores necessitam de saber utilizar o teclado, o rato e alguma experiência de utilização de interfaces gráficas.

Modelo Conceptual

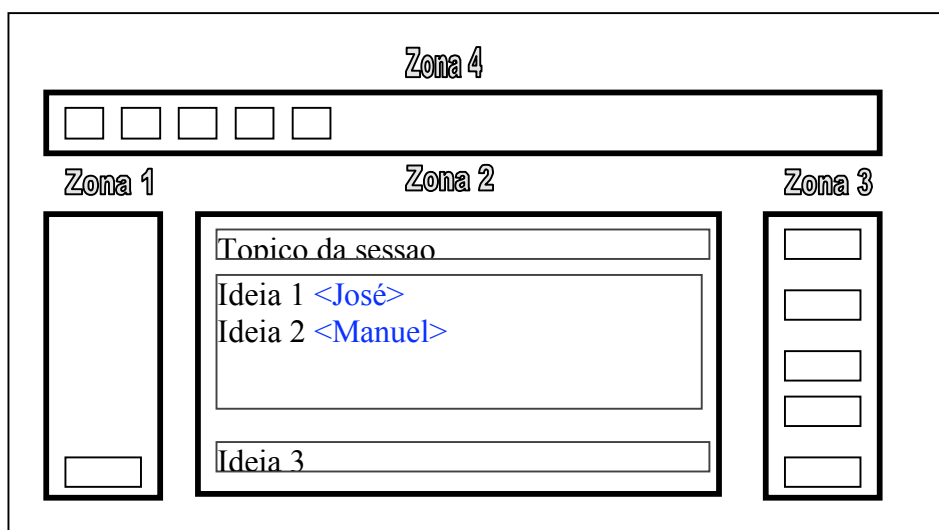
Variantes e Estímulos

As variantes ao método foram criadas para se adaptarem aos diversos tipos de problemas e/ou aos diversos tipos de participantes. Por exemplo, numa reunião onde estejam presentes pessoas de nível hierárquico diferentes, se o método não for anónimo pode provocar inibição nos subordinados.

O método implementado tem sempre $n+1$ páginas, onde n é o número de participantes. Quando as $n+1$ páginas estiverem cheias são adicionadas páginas automaticamente. Este método resolve problemas de escala, quer de carácter tecnológico (dimensões do monitor) quer de “capacidades” humanas pois é muito menos tedioso e mais rápido/agradável consultar listas curtas divididas por páginas do que uma longa lista. Isto faz com que a atenção do participante esteja dirigida só para o que é importante.

BRAINSTORM IDENTIFICADO

Variante onde os participantes estão perfeitamente identificados. A aplicação com a qual os



participantes irão interactivar é composta por quatro zonas, como mostra a figura abaixo.

A **zona-1** é a zona de navegação, mostra apenas um botão. Quando o botão é premido é mudada a página visível e o seu conteúdo será apresentado na parte central da zona-2, o utilizador poderá visualizar as ideias já introduzidas e acrescentar as suas através da parte inferior da zona-2. Note-se que as páginas são usadas para dividir uma longa lista de ideias à qual o utilizador introduz ideias sempre no fim, independentemente da página que está a visualizar. Os motivo para tal divisão é facilitar a leitura das ideias de modo a não maçar o utilizador com longas listas e por motivos de restrições físicas do monitor, neste caso. Esta zona no caso da variante – *Brainstorm* com categorias sofre uma alteração (ver *Brainstorm* com categorias)

A **zona-2** contém na parte superior o tópico da sessão, na parte central a página a ser visualizada e na parte inferior a zona de introdução de novas ideias. A única alteração que esta zona sofre para outras variantes é aparecer ou não a identificação do utilizador na parte central.

A **zona-3** é a zona que pode ou não estar presente pois é a zona onde irão ser apresentados os estímulos.

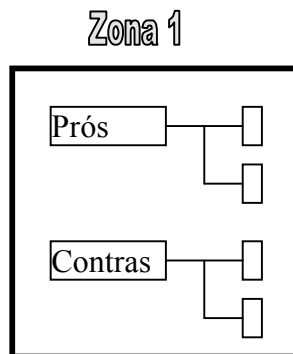
A **zona-4** é uma *toolbar* que está sempre presente, mas pode ou não estar activa dependendo se o estímulo com que está associada está ou não activo. O motivo que levou à toolbar estar sempre presente é o de o utilizador ir habitando-se à sua presença e não aparecer por “magia” numa sessão.

BRAINSTORM ANÓNIMO

Onde os participantes são anónimos, não se sabe quem forneceu uma determinada ideia. A alteração à interface é na parte central da zona-2 em que o nome do participante que introduziu a ideia não é visível. Minimiza as desvantagens 1) e 4).

BRAINSTORM COM CATEGORIAS

Variante onde as páginas estão divididas por categorias. As categorias começam por ter uma só página. Após esta página estar cheia é acrescentado uma nova página. A zona-1 é substituída por um esquema em árvore em que os ramos são as categorias e as folhas as páginas. Por exemplo podemos definir duas categorias prós e contras.



BRAINSTORM COM LIMITE DE TEMPO

Pode ser posto um limite de tempo em qualquer uma das variantes anteriores. Este estímulo tem como objectivo lembrar aos participantes que a sessão irá acabar, que a tarefa tem fim. Este prazo motiva os participantes e eventualmente aumenta a produtividade de ideias, pois prazos obrigam-nos a organizar o raciocínio e a fazer o que temos a fazer. Note-se que à que tomar atenção ao limite de tempo imposto, pois uma hora para certas pessoas é muito tempo para outras é suficiente para entrarem em pânico. É alertado o utilizador de 10% em 10% do tempo visualmente através de piscar e sonoro através de um apito, quando só falta 10% do tempo o utilizador é alertado de 1 em 1 por cento. Este alerta serve para chamar a atenção do utilizador.

BRAINSTORM COM RELÓGIO

Pode ser posto um relógio com o tempo que já passou desde o início da sessão. A ideia é idêntica ao limite de tempo, serve para dar um sentido de realidade ao utilizador.

USER RANKING

Pode ser apresentado um gráfico de barras com três barras que representa relativamente ao utilizador que introduziu mais ideias - o pior, o utilizador que está a ver o gráfico e o melhor. A ideia é dar, juntamente ou não, com o número total de ideias, uma ideia da produtividade. A intenção é criar um clima de competição ligeira, e para ajudar as pessoas a posicionarem-se o que pode levar a um aumento de produtividade atendendo que ninguém gosta de ser o último.

NÚMERO TOTAL DE IDEIAS INTRODUZIDAS

Pode ser apresentado o número total de ideias introduzidas pelos participantes. Os motivos são idênticos ao do *User Ranking*.

ESTÍMULO VISUAL

Este estímulo consiste em apresentar imagens aos participantes se estes estiverem um predeterminado tempo sem introduzir novas ideias. O objectivo é as imagens levarem à criação de novas ideias. Este estímulo baseia-se no facto que uma imagem pode sugerir associações, pensamentos ou sentimentos. Estas sugestões podem ajudar a criar novas ideias. Existem pessoas mais sensíveis a este tipo de estímulos que outras, nomeadamente pessoas que criem imagens mentais quando estão a criar ideias, são mais sensíveis que outras que não o façam.

REORDENAÇÃO AUTOMÁTICA DE IDEIAS

A ordenação da lista de ideias de cada página é feita automaticamente. A simples troca da ordem de como as ideias aparecem na página pode fazer surgir novas ideias, para além de fazer os participantes lerem ideias que tenham saltado na última visita à página. Ao aparecer a página com as ideias numa ordem diferente faz com que a página pareça nova para o utilizador, o que o leva a rele-la.

U S O D E E S T I L O S

O uso de estilos do tipo **Bold**, *Italic*, Underline e de cores nas ideias introduzidas, servem de estímulo para os outros participantes. Podem ainda facilitar a leitura das ideias, servindo o estilo utilizado para salientar a palavra chave da ideia. À que explicar aos participantes que só se deve usar este estímulo quando realmente é importante pois o seu uso excessivo pode complicar a leitura das ideias.

Implementação

1. Metodologia

Definição da Metodologia de Desenvolvimento

Dado que os requisitos não são totalmente conhecidos inicialmente, isto é, que dentro de cada sub-componente existem uma serie de opções em aberto, decidi optar por um modelo de processo de desenvolvimento de *software* evolucionário – o modelo Incremental com prototipagem. A interface com o utilizador é refinada sempre que necessário

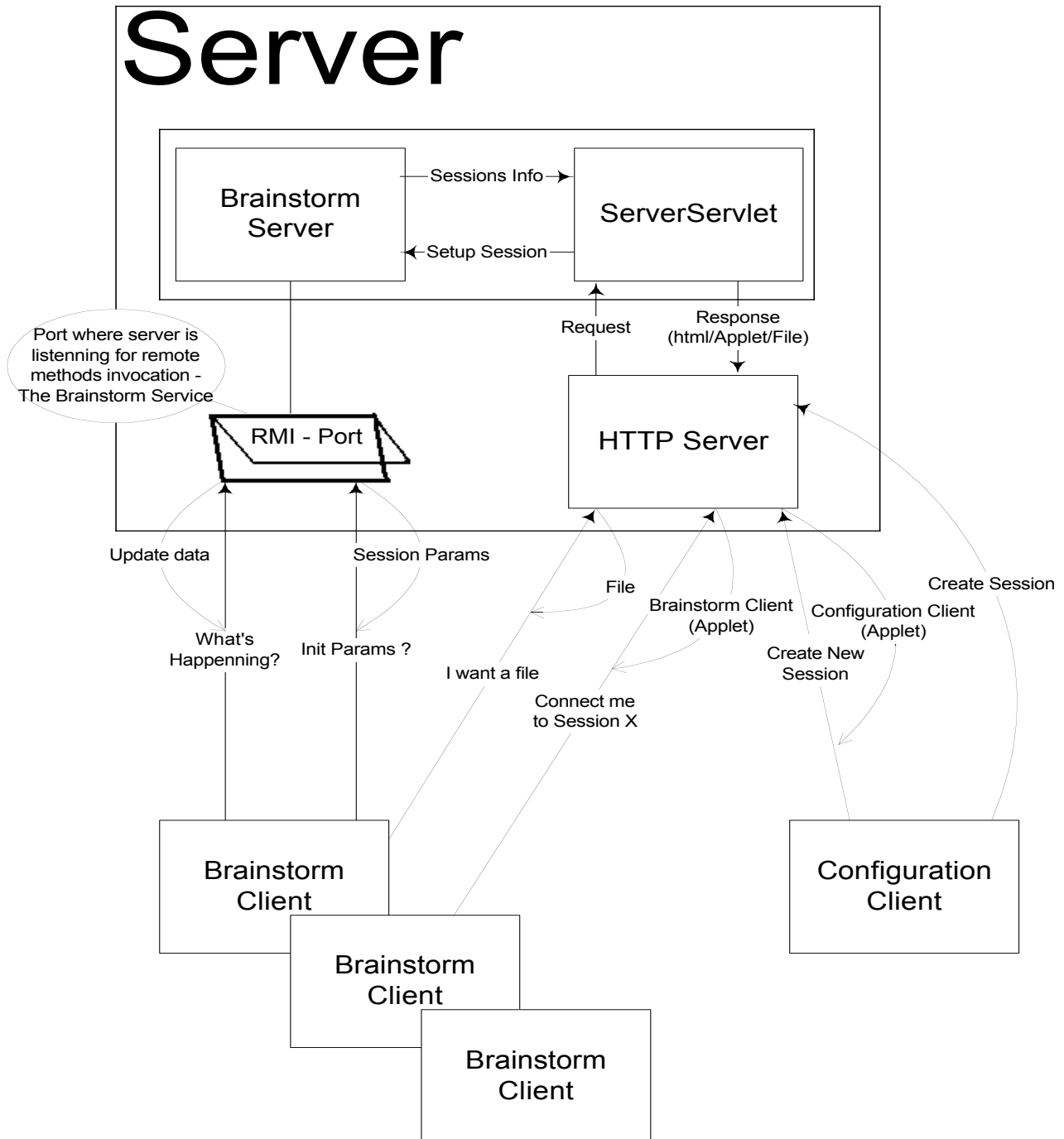
Fases do projecto

- ⇒ Definição do Produto (Modelo Conceptual)
 - Definição das fronteiras do sistema, perceber quais as funcionalidades pretendidas.
 - Utilizador alvo.
- ⇒ Definição da metodologia – o modelo do processo a utilizar
- ⇒ Divisão do Projecto em pacotes – *WorkPackages*
Cada estímulo/variante é um *workpackage*, o servidor, o *servlet* e o cliente de configuração são os outros *workpackages*
- ⇒ Aplicação do modelo do processo de *software* a cada pacote
- ⇒ Depois cada pacote funcional estar acabado foi integrado com o resto dos pacotes, e testada a integração.

2. Arquitectura

O sistema usa uma arquitectura cliente-servidor sobre RMI (Remote Method Invocation).

Esquema



Funcionalidade

A comunicação entre cliente-servidor é efectuada em duas fase. A primeira consiste na criação e configuração da sessão, nesta fase a comunicação com o servidor é feita através de um *servlet* (um módulo que serve de portal para o servidor - ver abaixo), este disponibiliza toda a informação sobre cada sessão e todas as funções que podem ser executadas sobre estas. A segunda fase dá-se após a ligação do cliente a uma das sessões existentes, apartir desse momento a comunicação entre o cliente e o servidor é efectuada através de RMI(*Remote Method Invocation*), excepto para o carregamento de ficheiros que é usado o *servlet*. O *servlet* é acedido via servidor de *http*.

No decorrer da sessão o tempo de resposta do servidor é muito importante para os utilizadores, pois quanto menor for o tempo de resposta maior a interacção. Este foi o principal motivo que levou à escolha do mecanismo de RMI, outras vantagens deste mecanismo são a simplicidade de implementação, a fiabilidade, a garantia de resposta.

Motivos

O motivo que levou à escolha de um *servlet* para servir de portal com o servidor em vez de um CGI foi, a simplicidade, a possibilidade de manter estados, a segurança, a rapidez – pois é só uma aplicação a correr em vez de uma série de pequenas aplicações, o que poupa o *overhead* de carregamentos das pequenas aplicações. Toda a comunicação com o servidor não é feita através do *servlet* porque com muitos clientes iria haver estrangulamento muito cedo, seria como se se tivesse um só cliente a ser servido de cada vez. Através do RMI o servidor cria uma *thread* para cada pedido o que faz com que as respostas sejam muito mais rápidas devido à sincronização ser mais fina.

Os motivos de não se usar *sockets* para a comunicação foram em primeiro lugar o de se ter que criar uma serie de métodos para tratar dos pedidos dos clientes, os quais já são oferecidos pelo RMI; e em segundo as restrições impostas pelas *applets* de não se poder criar *sockets* a não ser em circunstâncias especiais.

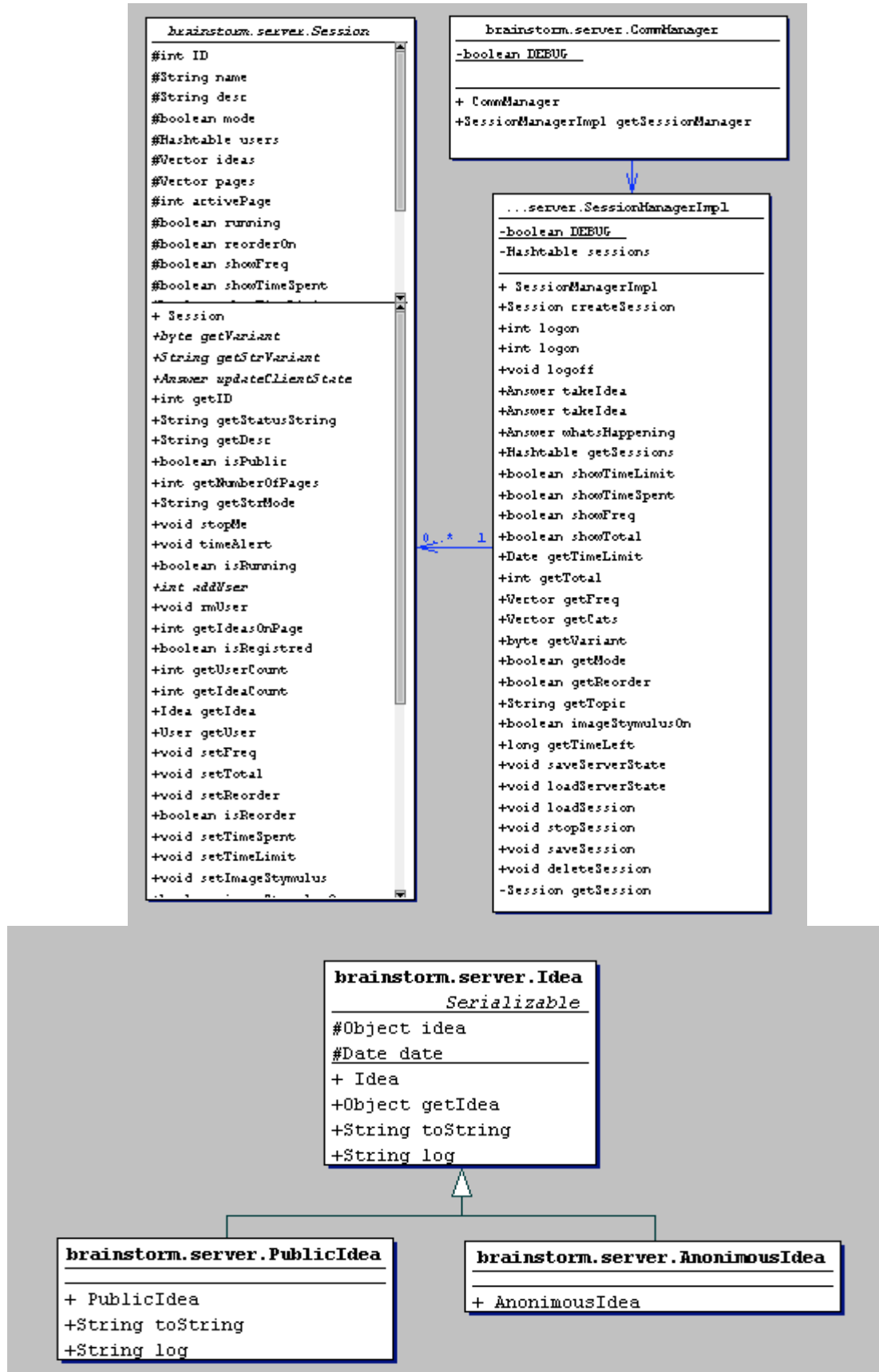
Para a comunicação durante a sessão foi usado um mecanismo de *polling* ao servidor, o motivo para se usar *polling* em vez de se o usar o mecanismo de *callback* (mecanismo que permite o servidor fazer uma chamada a um método remoto do cliente) oferecido pelo RMI foi a necessidade de ter-se que criar no cliente um *socket* para receber os pedidos do servidor, o que

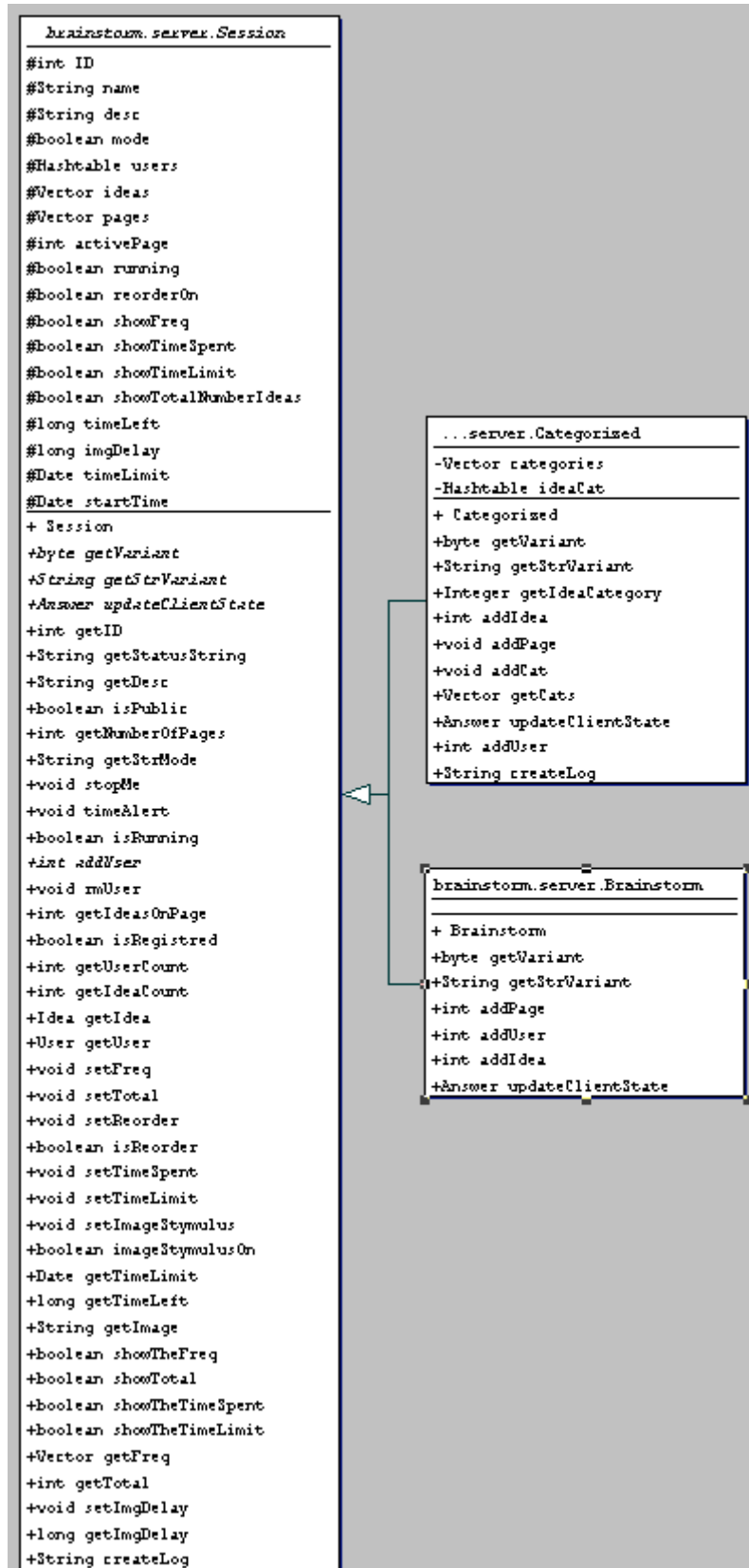
não é possível a uma *applet*, a não ser que esta seja “assinada” com sendo de uma fonte segura e pela qual se tem que pagar uma quantia anual. Se fosse possível usar o mecanismo de *callbacks* de facto a aplicação era mais eficiente dado que o tempo pedido-resposta era substituído por pedido, mais, vão existir muitos pedidos para o qual o servidor não tem informação para enviar, ou seja, este tempo também era reduzido.

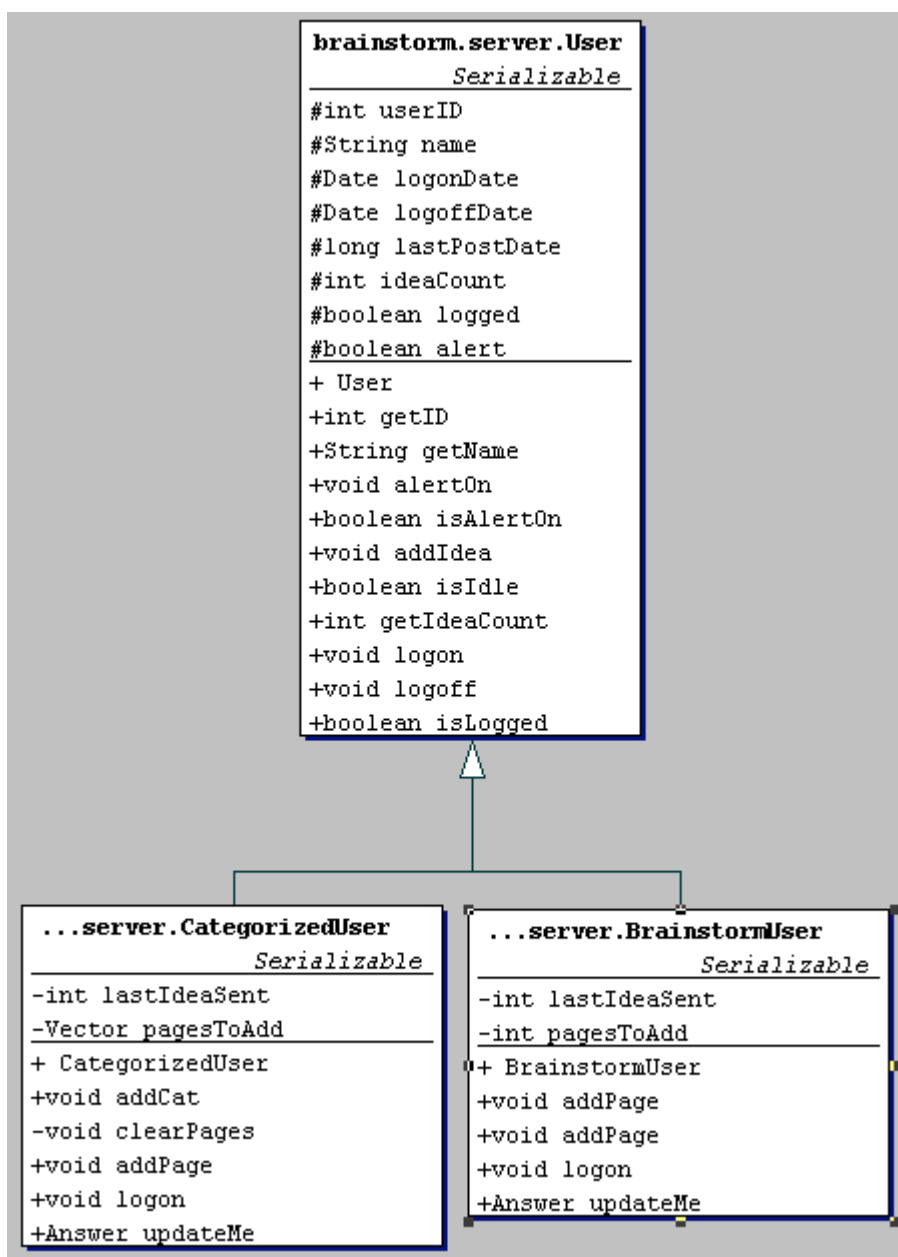
3. Servidor

Arquitectura do Software

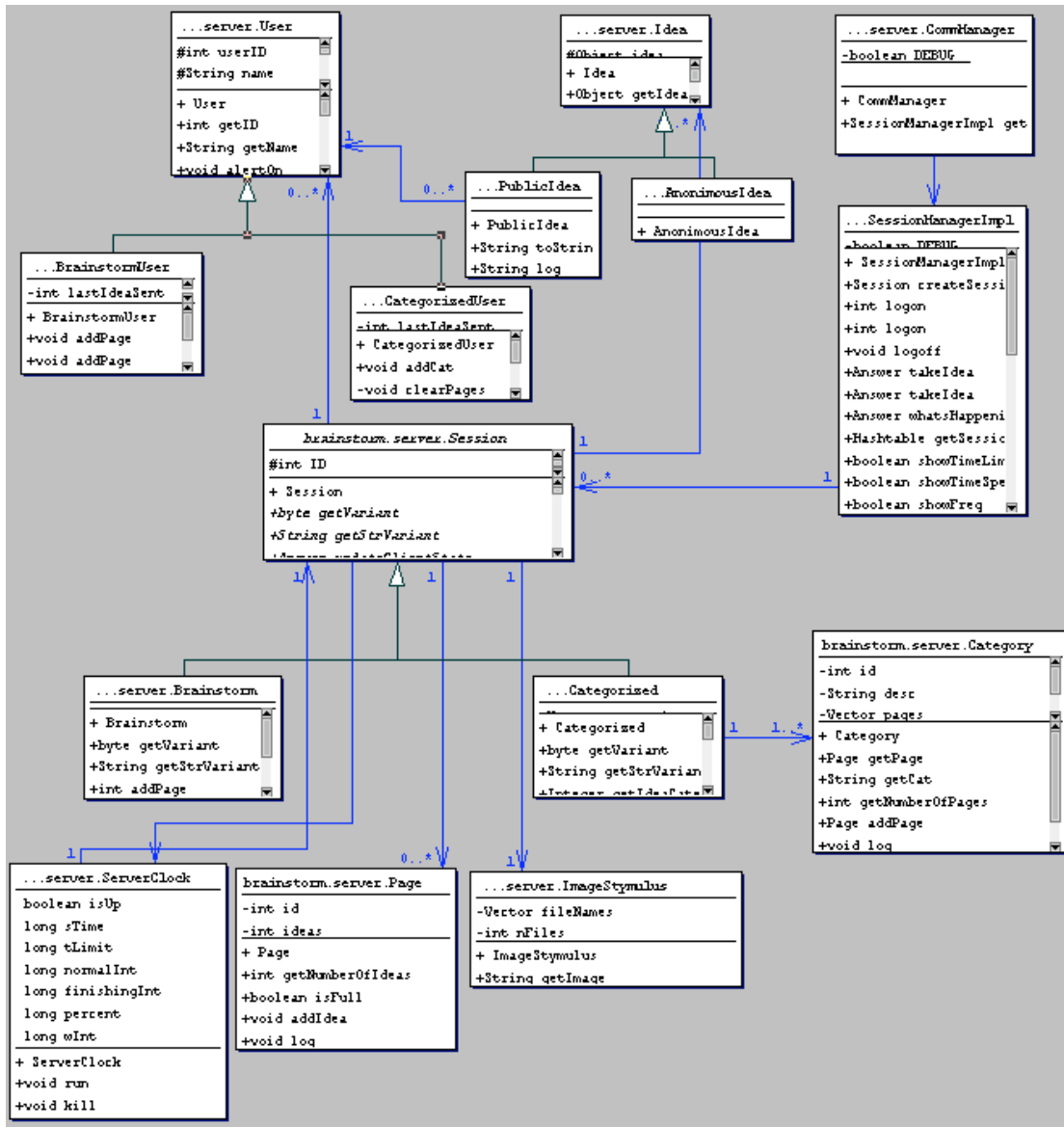
SessionManager	Interface remota do Servidor
SessionManagerImpl	Implementa a interface remota
Session	Representa uma sessão
Brainstorm	Uma sessão do tipo Brainstorm
Categorized	Uma sessão de brainstorm por categorias
Category	Uma categoria
User	Um utilizador
BrainstormUser	Um cliente de uma sessão de brainstorm
CategorizedUser	Um cliente de uma sessão de com categorias
Idea	Uma ideia
PublicIdea	Uma ideia pública – sabe-se o nome do utilizador
AnonymousIdea	Uma ideia anónima
Page	Uma página de ideias
ServerClock	O relógio do servidor, controla o tempo da sessão
ImageStymulus	Vê as imagens disponíveis e fornece-as ao cliente
CommManager	Cria e regista os serviços do servidor







Esquema Total



Funcionamento

O servidor é criado e registado assim que se cria um objecto da classe `CommManager`, após isto o servidor fica à espera que pedidos. Em princípio o primeiro pedido é o de criação de uma sessão feita através do *servlet*, após a criação o *servlet* define todos os parâmetros que uma sessão pode ter. Tendo a sessão criada, os utilizadores já se podem ligar, fazendo os pedidos através da interface remota que é oferecida pelo servidor. Numa sessão normal o cliente começa por se registar através do método *logon*, depois pede todos os parâmetros da sessão através dos métodos *showXXX* e *getXXX*, após isto o utilizador pode começar a enviar ideias. Apartir deste momento o utilizador envia ideias, através do método *takeIdea*, e faz pedidos de actualização de alguns estímulos e ideias, através do método *whatsHappening*. O “pacote” de resposta ao *whatsHappening* contém as ideias que o utilizador ainda não têm, alerta do relógio, páginas a adicionar e imagem a mostrar. O estímulo “User Ranking” e “Numero Total de Ideias” é feito em paralelo pois não é tão crítico como a actualização das ideias ou das páginas.

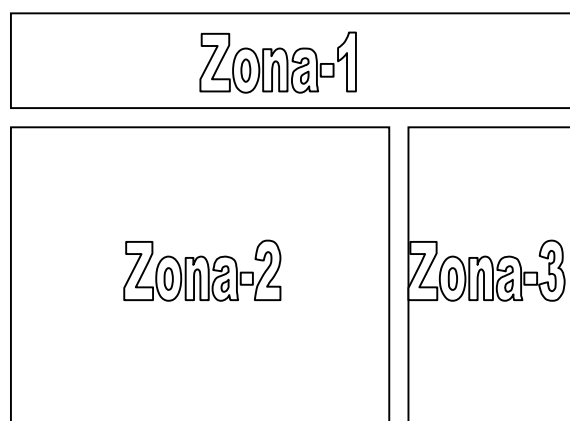
4. Cliente – Configuração

A criação e configuração das sessões é feita através de uma aplicação (*applet*), onde é possível seleccionar e definir todas as variantes, estímulos e respectivos parâmetros como acima descritos (ver Modelo Conceptual – Variantes e Estímulos).

Após a configuração todos os parâmetros da sessão são passados para a *servlet* - ServerServlet que irá criar a sessão.

Elementos da Interface Homem-Máquina

Todos os ecrãs têm três zonas distintas. A Zona-1 é o painel onde se encontra a identificação, o numero de passos que faltam e qual em que passo se está. A Zona-2 é o painel onde se escolhe as opções que se desejam. Por fim a Zona-3 é a zona de navegações/controlo onde se avança, recua, cria a sessão, grava dados e se aborta a criação da sessão. A Zona-1 e 3 são comuns aos quatro ecrãs possíveis.



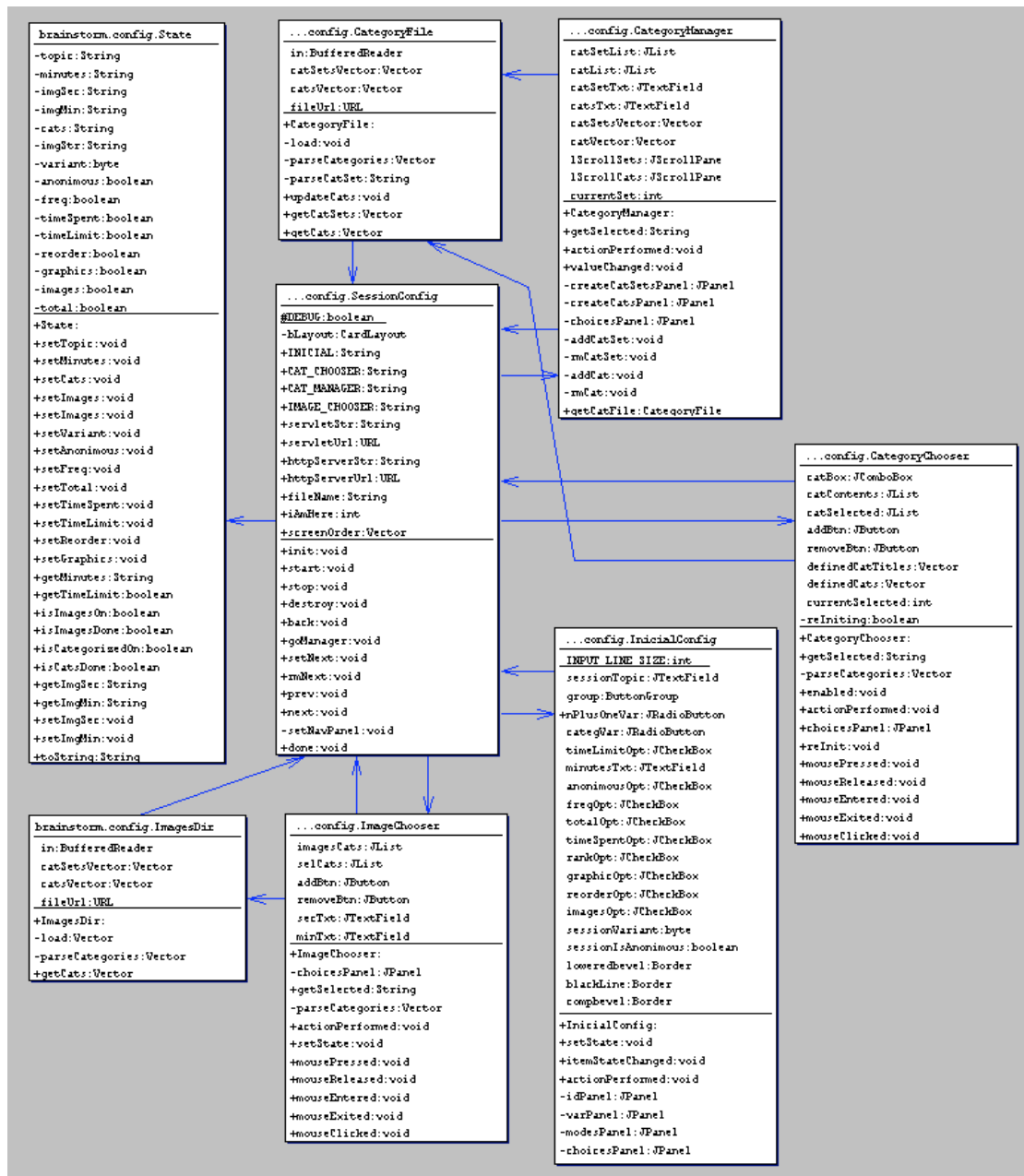
A disposição dos elementos desta forma tem a ver sobretudo com os hábitos culturais dos utilizadores. A cultura ocidental escreve da esquerda para a direita e de cima para baixo, logo uma pessoa quando olha para um ecrã começa por olhar para o canto superior esquerdo (a não ser que haja algo que chame muito a atenção para outro sítio) e depois para a direita – lendo a identificação do que vai fazer; depois muda de “linha” e entra na zona das opções e começa a fazer as suas escolhas, finalmente quando tiver terminado vai procurar o que fazer a seguir, e irá prestar mais atenção ao lado direito onde se encontra o painel de navegação.

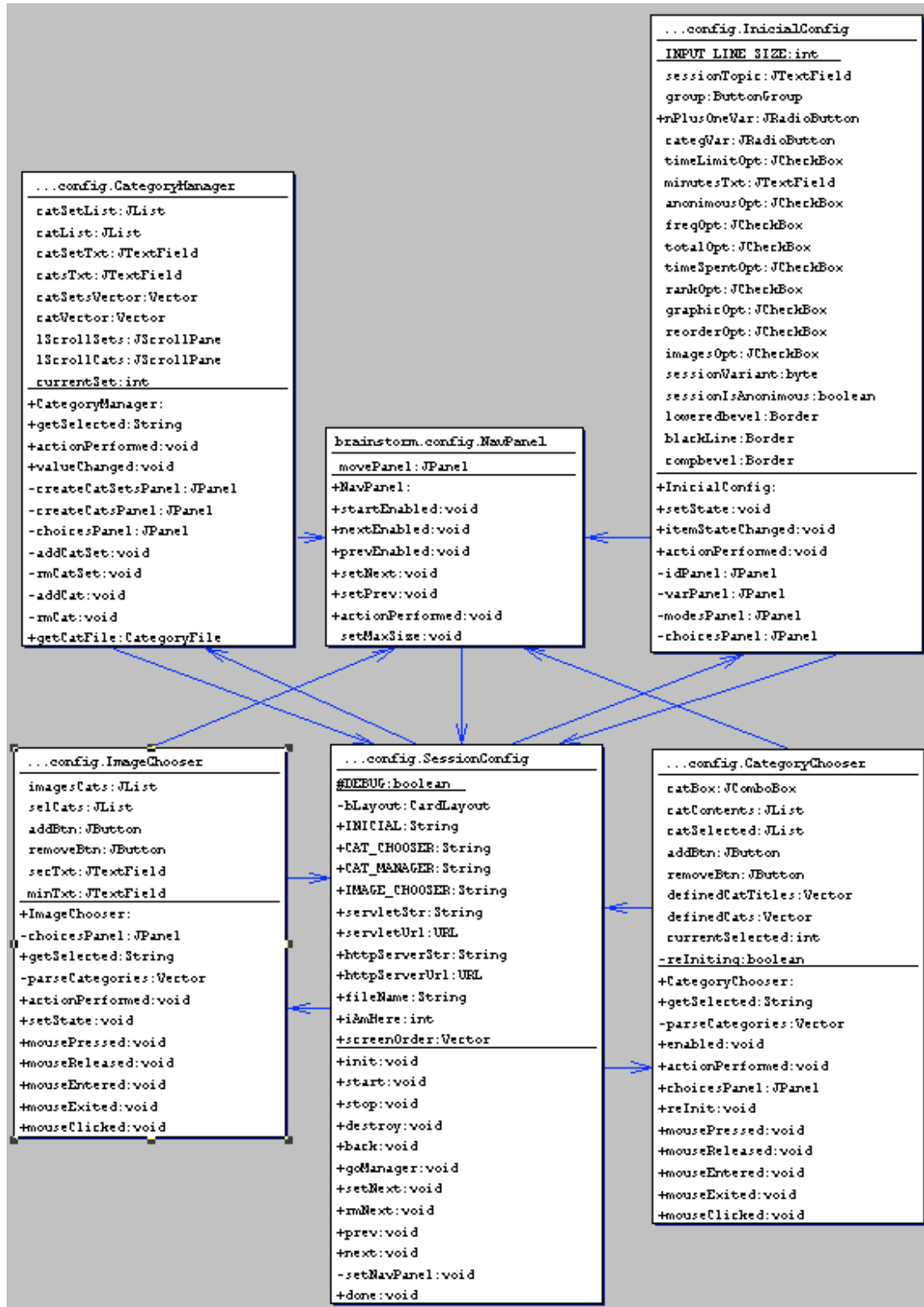
Os botões de navegação podem estar activados ou desactivados dependendo das opções dos utilizadores, isto para que o utilizador só faça o que pode realmente fazer, caso contrário ter-

se-iam que apresentar mensagens de erro cada vez que o utilizador pressiona-se um botão sem que a acção estivesse disponível o que se torna aborrecido, irritante e dificulta a aprendizagem sobre a aplicação. Todos os elementos têm *tooltips* para facilitar a compreensão a utilizadores menos experientes.

Arquitectura do Software

SessionConfig	Applet
InicialConfig	Ecrã inicial de configuração
CategoryChooser	Ecrã onde se escolhem as categorias
ImageChooser	Ecrã onde se escolhem as categorias de imagens
CategoryManager	Ecrã para gerir as categorias
CategoryFile	Lê e escreve as categorias para um ficheiro
ImagesDir	Lê as categorias de imagens disponíveis
NavPanel	Painel de navegação pela configuração
State	Mantém o estado das opções seleccionadas pelo utilizador





Funcionamento

O ecrã inicial contém na Zona-2 do lado esquerdo dois botões-radiais que servem para se escolher a variante a sessão normal ou por categorias. Caso se escolha por categorias é activado o botão de próximo da Zona-3 (navegação) indicando que se têm de escolher as categorias que se desejam incluir. Na Zona-2 do lado direito é onde se escolhem os estímulos para a sessão através de *checkboxes*, estes podem ser *anonymous*, *duration*, *clock*, *user rank*, *use styles*, *Images* e *Number Ideas*. Caso se escolha *Images*, o botão de próximo é activado indicando que se tem que escolher as categorias de imagens que se desejam para a sessão.

Session Configuration *main*

Session Identification
Topic: Reuniões Electrónicas

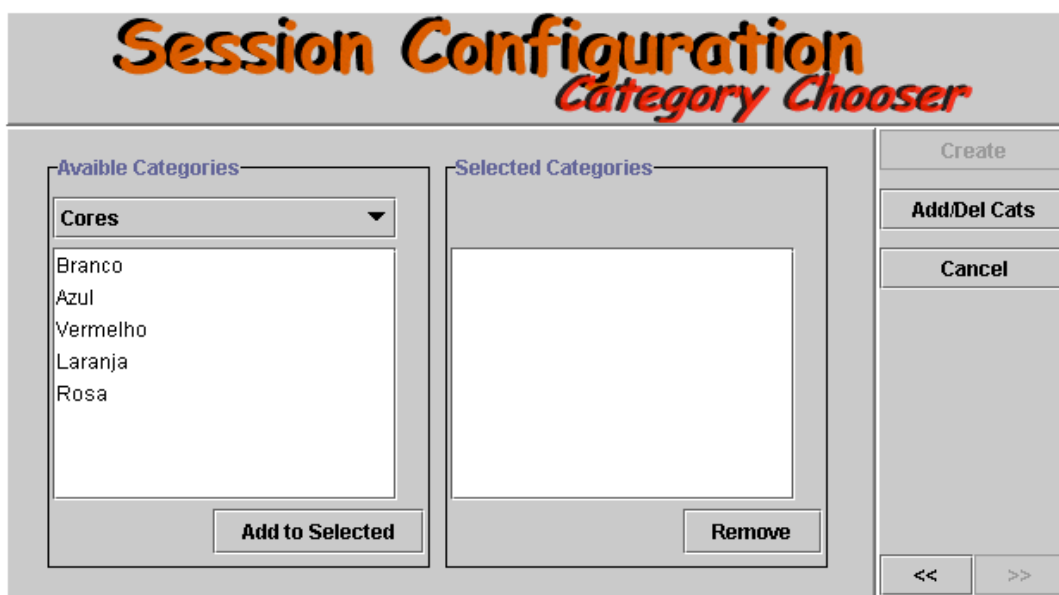
Brainstorm Variants
 Normal
 with Categories

Brainstorm Stymulus
 Duration: 00 : 30
 Anonimous
 User rank
 Clock
 Use styles
 Reorder ideas
 Images
 Number Ideas

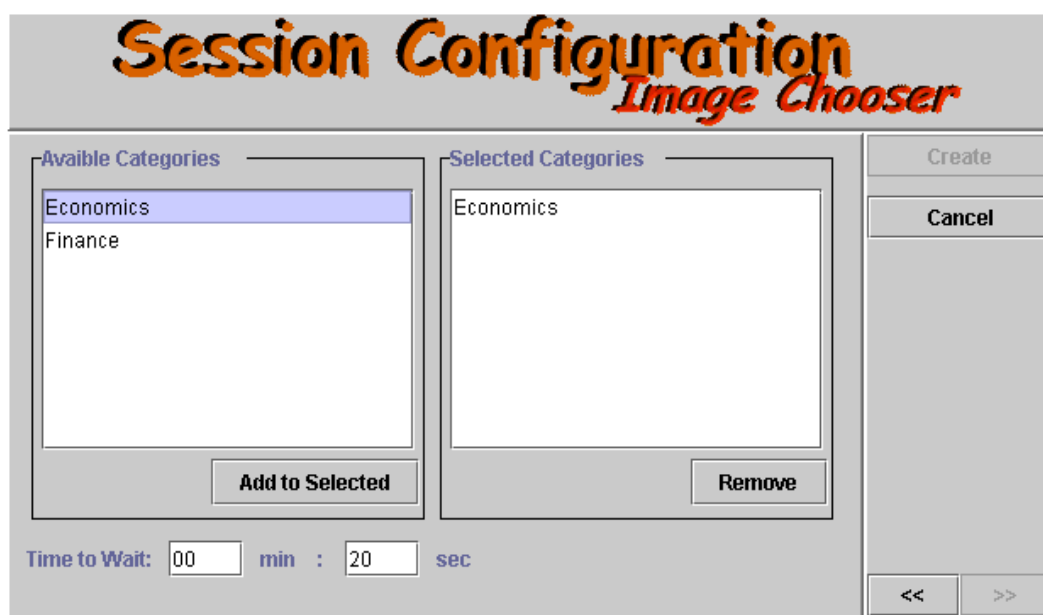
Create
Cancel

<< >>

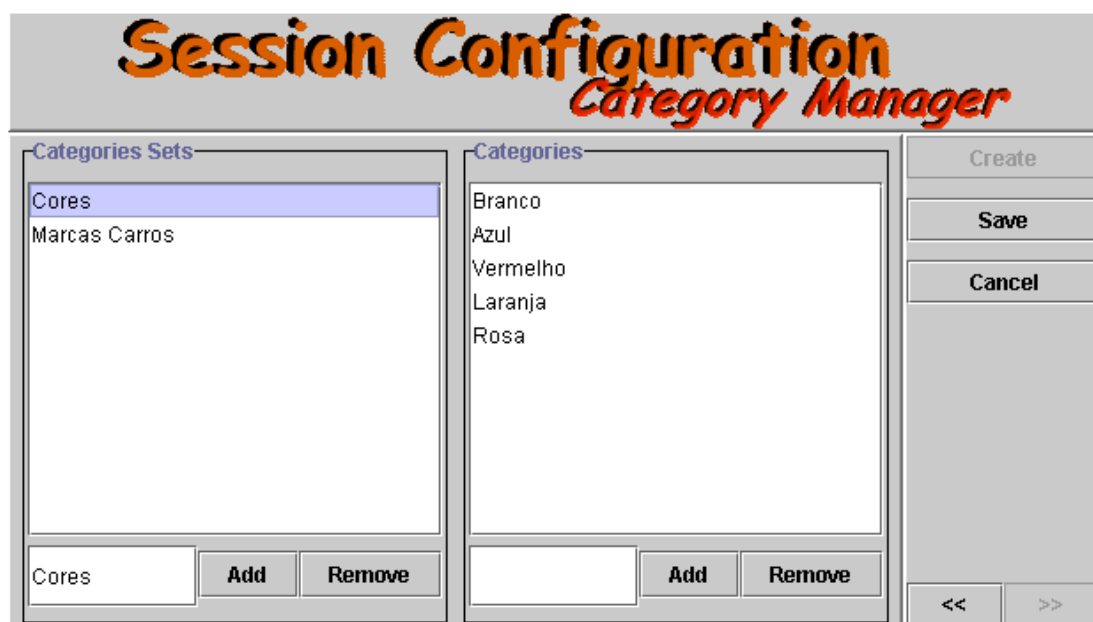
No Ecrã para escolher as categorias temos conjuntos de categorias que são seleccionados através de uma *combo-box*, as categorias que pertencem a este conjunto são apresentadas na *list-box* por baixo da *combo-box*. Existe um *list-box* do lado direito que apresenta as categorias seleccionadas para a sessão, estas categorias são seleccionadas através de *double-click* na lista da esquerda ou por selecção e premindo o botão de **add** na lista da esquerda. Para remover o processo é idêntico. Pode-se ainda adicionar, remover categorias, conjuntos de categorias e gravar estas alterações por defeito, usando o “Gestor de Categorias” através do botão de Add/Del Cat.



O funcionamento do ecrã de Escolha de Categorias de Imagens a mostrar é idêntico ao de Escolha de Categorias.



Quanto ao ecrã de gestão de categorias do lado esquerdo da zona-2 é apresentada a lista de conjuntos de categorias à qual se pode acrescentar ou remover elementos. Quando selecionado um dos conjuntos, são apresentadas do lado direito as categorias pertencentes a esse conjunto às quais, se pode acrescentar ou remover elementos. A zona de navegação tem um novo botão - **SAVE** que é usado para gravar as alterações por defeito.



5. Cliente – Brainstorm normal

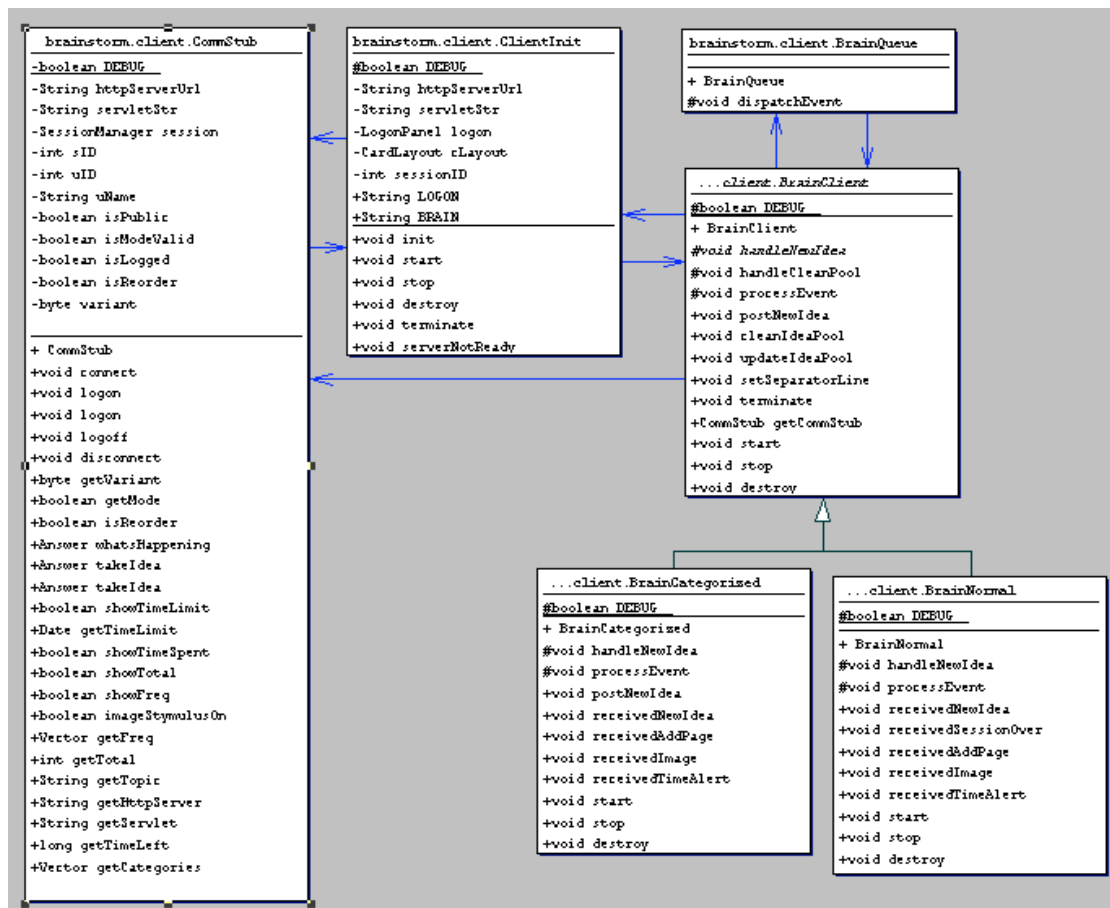
Elementos da Interface Homem-Máquina

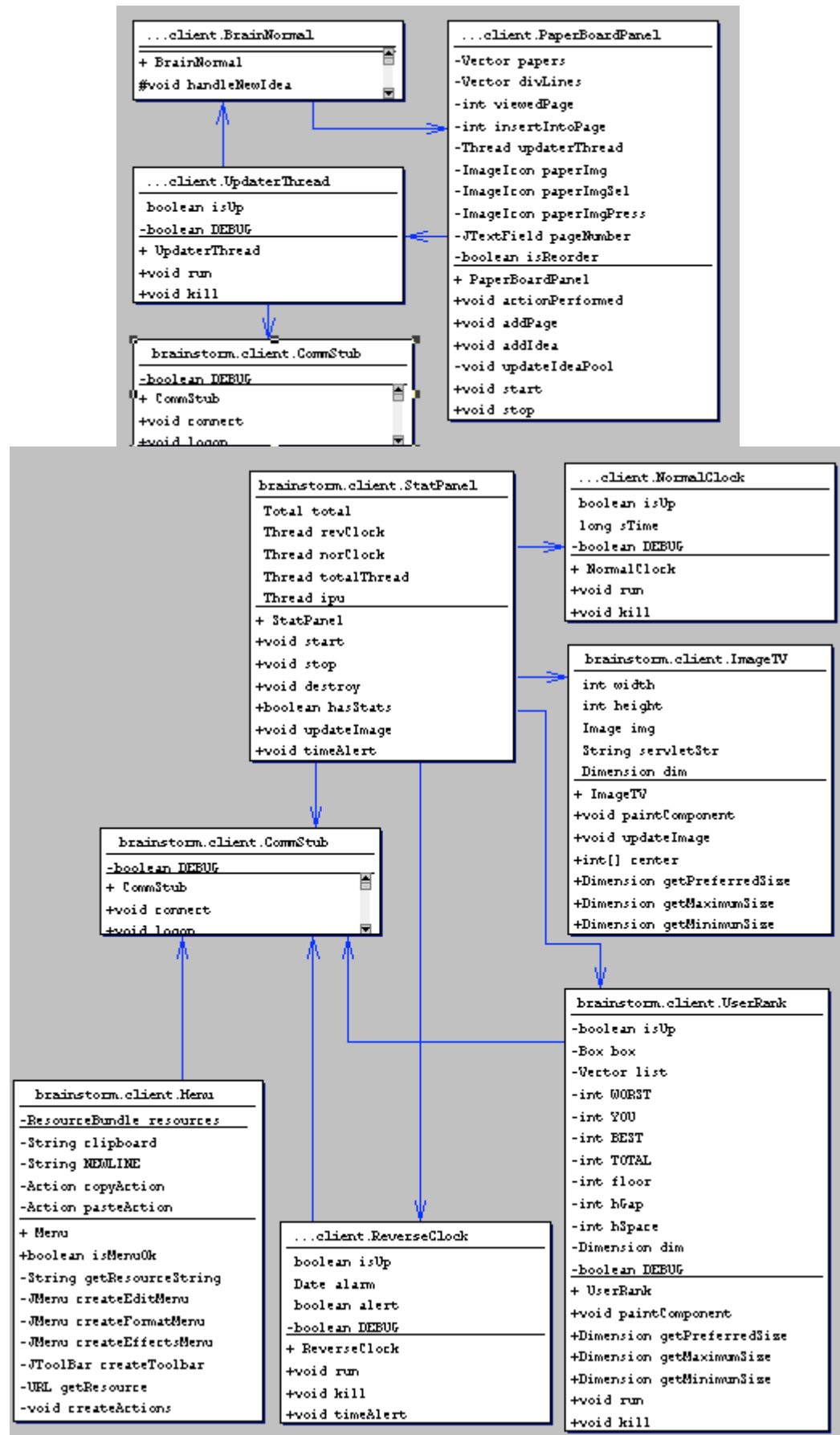
A composição da interface é tal como foi vista na secção Modelo Conceptual – Variantes e Estímulos Implementadas, resta definir a interacção com o utilizador.

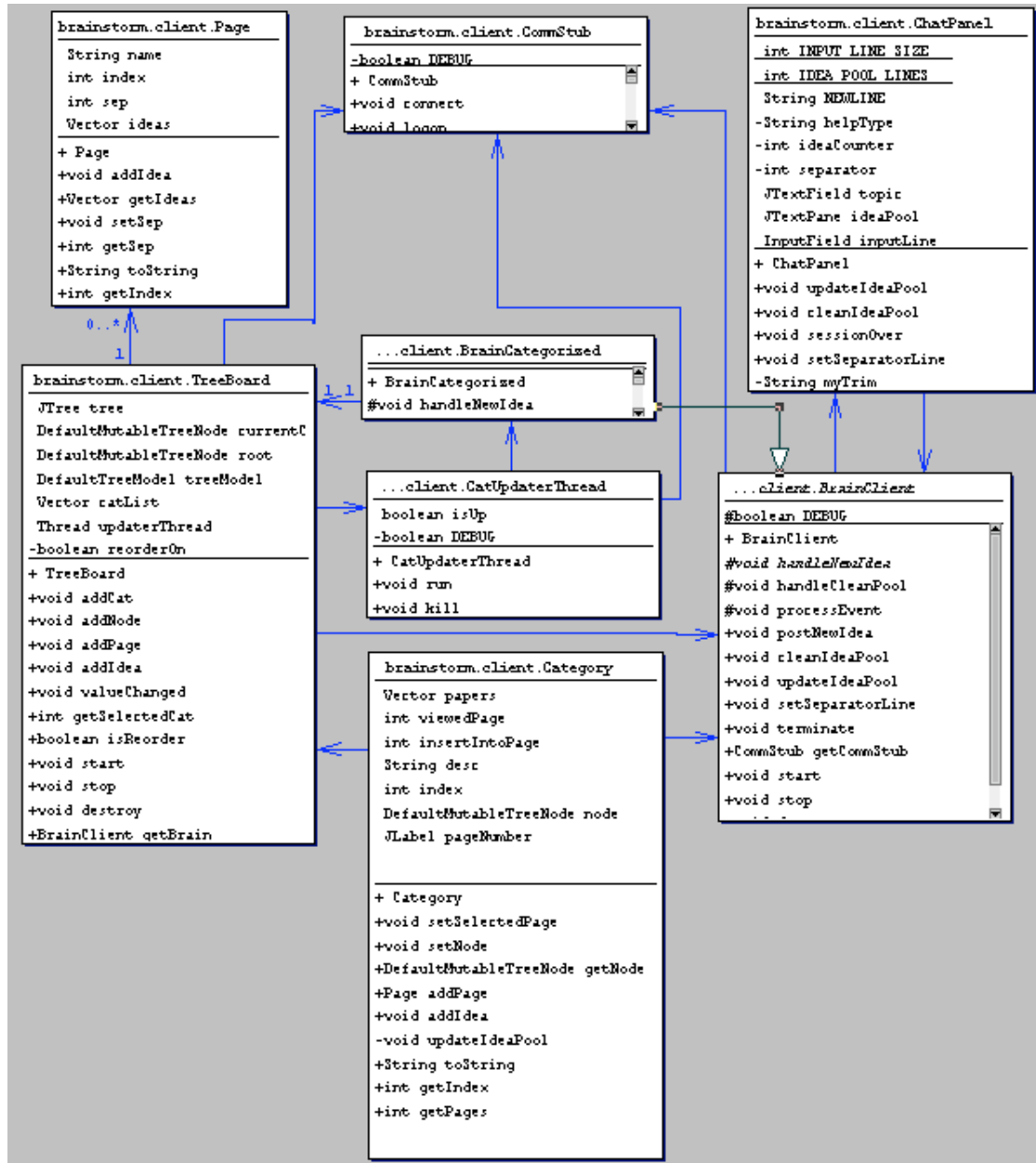
Arquitectura do Software

ClientInit	Applet – Inicializa o cliente
CommStub	Representa a ligação ao servidor. Trata todos os pedidos
BrainClient	Uma sessão cliente
BrainNormal	Uma sessão Brainstorm
BrainCategorized	Uma sessão com brainstorm com categorias
ChatPanel	Painel onde é apresentada a lista de ideias e onde o utilizador insere novas ideias
PaperBoardPanel	Painel onde o utilizador muda de página
TreeBoard	Painel onde o utilizador muda de categoria e ou página.
StatPanel	Painel onde são apresentados os estímulos
UpdaterThread	Thread responsável pela actualização do cliente numa sessão brainstorm
CatUpdaterThread	Thread responsável pela actualização do cliente numa sessão brainstorm com categorias
Category	Uma categoria
Page	Uma página de ideias
BrainQueue	Queue de eventos do cliente

CleanPoolEvent	Evento para limpar a página de ideias
NewIdeaEvent	Evento para enviar uma ideia para o servidor
NewIdeaCatEvent	Evento para enviar uma ideia para o servidor
ImageTV	Estímulo visual, responsável por apresentar as imagens
NormalClock	O estímulo relógio, responsável por apresentar e manter o relógio
ReverserClock	O estímulo limite de tempo, responsável por apresentar e manter o relógio
UserRank	O estímulo User Ranking, responsável por apresentar o gráfico
TotalIdeas	O Estímulo número total de ideias
Menu	Menu da aplicação

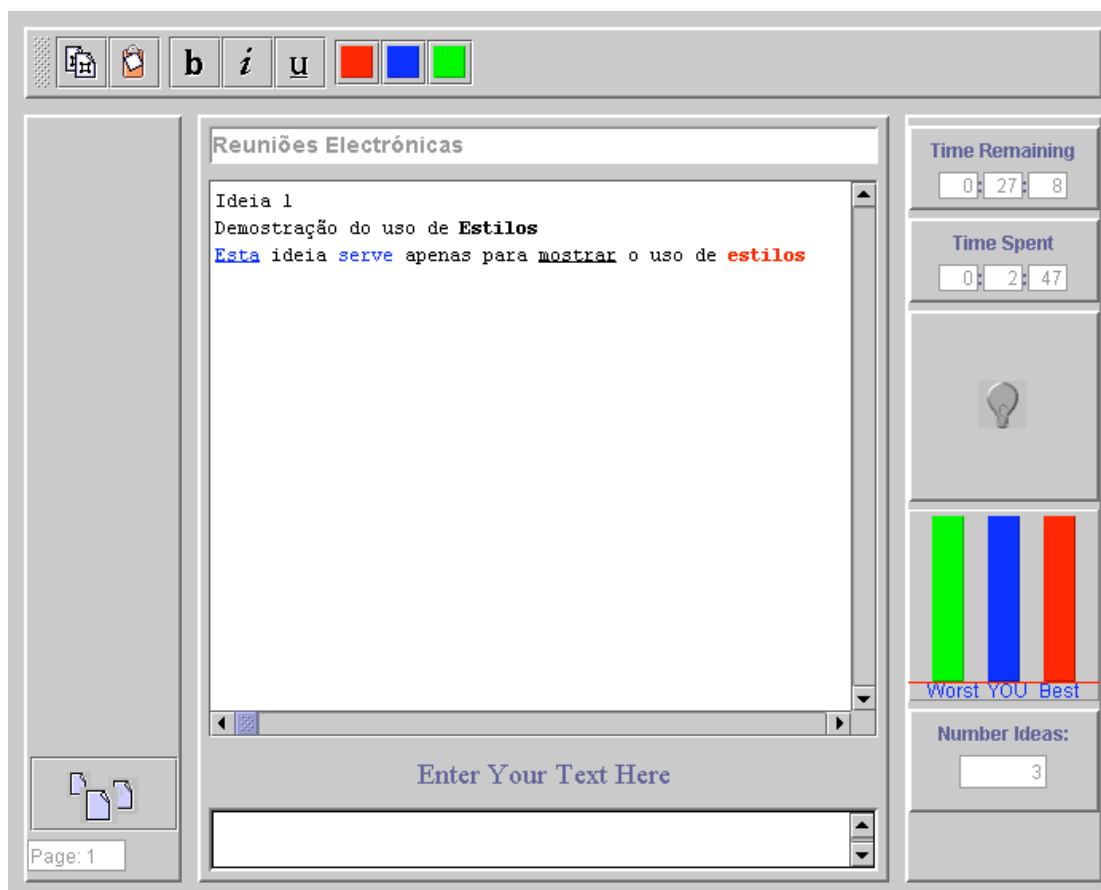






Funcionamento

As funções disponíveis para o utilizador são: mudar de página – obtido carregando no botão na Zona-1, introduzir uma ideia – escrever a ideia na parte inferior da Zona-2 e pressionar [ENTER], se o estímulo de estilos estiver activado mudar o estilo do texto da ideia que está a inserir, e copiar, colar texto através da *toolbar*.



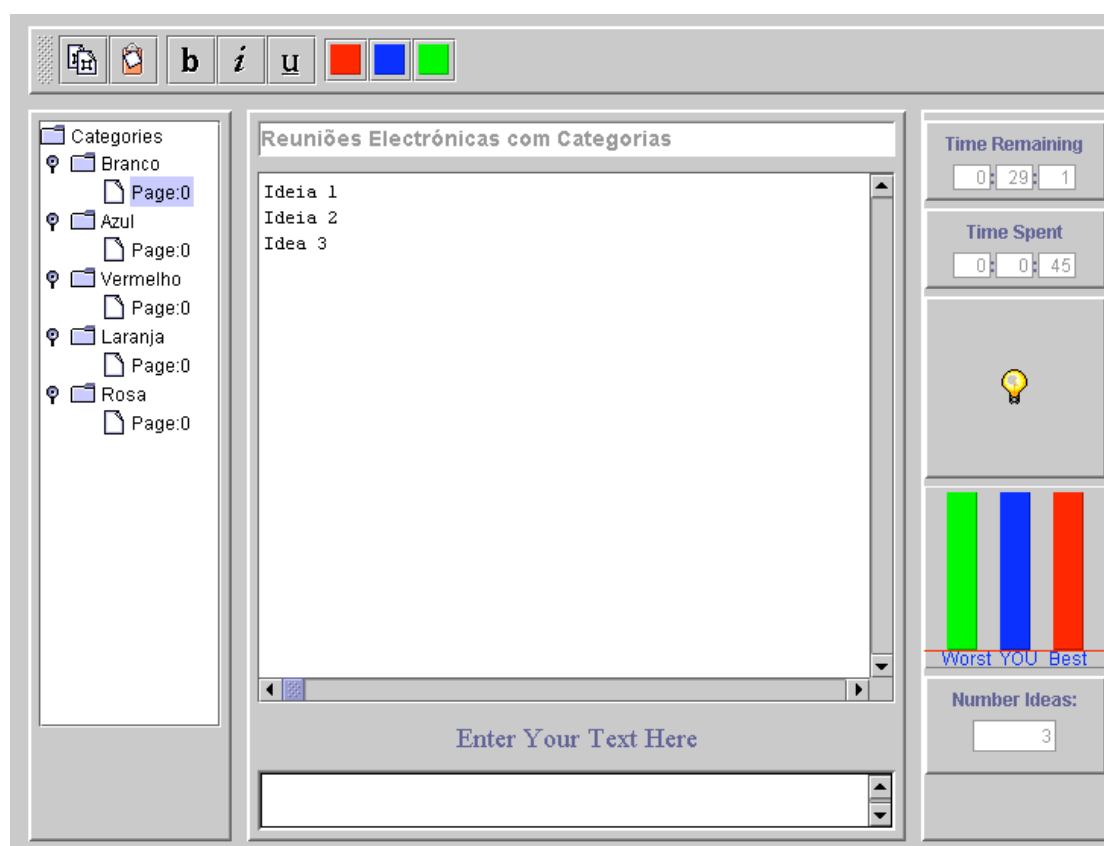
6. Cliente – Brainstorm com Categorias

Elementos da Interface Homem-Máquina

A composição da interface é tal como foi vista na secção Modelo Conceptual – Variantes e Estímulos Implementadas, resta definir a interacção com o utilizador.

Funcionamento

O funcionamento é idêntico ao do Brainstorm Normal, excepto para mudar de página, dado que a Zona-1 é mudada para uma árvore de categorias. O utilizador para seleccionar a página que deseja visualizar, tem que seleccionar uma da árvore ou se seleccionar uma categoria é seleccionada automaticamente a primeira página dessa categoria.



7. Servlet - ServerServlet

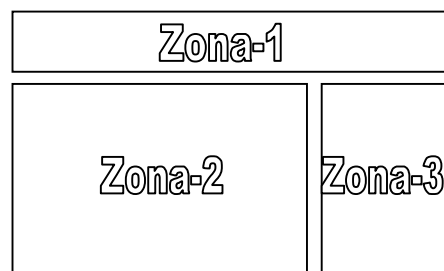
Um *servlet* é um módulo (objecto) que corre do lado do servidor e serve de portal de acesso a este. É através do *servlet* que todas as funções de criação, configuração, consulta de logs das sessões, lista de sessões, inicialização do servidor, gravar e carregamento de sessões são efectuadas. Os motivos que levaram à escolha de uma *servlet* para servir de portal para o servidor foram sobretudo a libertação do servidor para as tarefas realmente críticas, deixando todas as verificações de dados, manutenção do estado, criação e remoção das sessões para a *servlet*. Mais, como o *servlet* é uma só aplicação, é bastante rápida e a comunicação com esta é feita através de qualquer *browser*. Existe ainda a possibilidade de se implementar um sistema de segurança/autenticação sem que se tenha que interagir com o servidor ou seja separar bem as camadas do protocolo de comunicação, deixando o servidor o mais livre possível para um melhor desempenho.

Além de todas as funções acima referidas, a *servlet* é usada ainda para servir todos os ficheiros aos clientes sejam estes de dados, imagens ou sons, novamente para libertar o servidor para a sua tarefa crítica.

A *servlet* funciona em dois modos, utilizador e administrador. As diferenças são nas funções que cada um pode executar. No modo utilizador, que é o por defeito, o utilizador só pode seleccionar uma sessão, juntar-se à sessão e actualizar a lista de sessões. No modo administrador o utilizador para além do que pode fazer no modo utilizador pode ainda criar, apagar, gravar, carregar, consultar logs e terminar sessões.

Elementos da Interface Homem-Máquina

A interface foi construída para ser o mais simples possível. É apresentada a lista de sessões do lado esquerdo, um título em cima e novamente a navegação à direita. A lista de sessões tem o tópico da sessão o seu estado (running ou stopped) e alguns estímulos escolhidos.



Funcionamento

MODO UTILIZADOR

Neste modo o utilizador só pode executar quatro funções: seleccionar a sessão, juntar-se à sessão, actualizar lista de sessões e passar para modo administrador. Se o utilizador não seleccionar uma das sessões antes de premir o botão para se juntar à sessão é mostrada uma página a descrever o problema.

Information Gathering System using Brainstorm Technics

--- Control Center ---

Active Sessions			
Status	Topic	Description	Select
Running	Sessao 2	Brainstorm with Categories - Public	1
Running	Sessao 1	A page per user plus one - Anonimous	0

Connect to Session
Admin. Mode
Refresh

MODO ADMINISTRADOR

Neste modo todas as funções estão acessíveis, para além das do modo utilizador, as funções criar, apagar, terminar, consultar log, gravar e carregar sessão. Todas estas funções são executadas através dos botões da zona-3 e sempre que o utilizador execute algo de errado é apresentada uma página a descrever o sucedido e como corrigir o erro.

Information Gathering System using Brainstorm Technics

--- Control Center ---

Active Sessions			
Status	Topic	Description	Select
Running	Sessao 2	Brainstorm with Categories - Public	1
Running	Sessao 1	A page per user plus one - Anonimous	0

Create Session
Connect to Session
View Log
Stop Session
Save Session
Delete Session
Load Session
Save Server State
Load Server State
User Mode
Refresh

Conclusão

O objectivo deste projecto era implementar uma ferramenta de recolha de informação em reuniões electrónicas fase-a-fase que fosse, simples de usar por utilizadores com pouca experiência em sistemas informáticos, não tivesse as limitações que a maioria dos pacotes comerciais têm e que não estivesse dependente de nenhuma plataforma.

Estes objectivos foram todos alcançados.

Para alcançar estes objectivos foi usada tecnologia de ultima geração, por exemplo, aplicações cliente-servidor, linguagem independente da plataforma – Java e modelo de comunicação distribuída – RMI. Esta tecnologia, as técnicas de *brainstorming* que hoje em dia conseguem melhores resultados foram os meios para alcançar os objectivos.

Outros factores que se tiveram em conta foram, a extensibilidade, isto é, a arquitectura da aplicação foi desenhada de modo a que seja simples acrescentar outras variantes; dado que a aplicação faz uso de html e de ficheiros de resources, muitos elementos podem ser alterados dinamicamente, sem se recompilar a aplicação como por exemplo mudar a linguagem para a qual a aplicação foi criada.

O desempenho e robustez do servidor é excelente, quanto ao desempenho do cliente a inicialização é um pouco lenta mas no decorrer da sessão é óptima, a robustez do cliente é de algum modo frágil isto porque depende do *browser* e da *Java Virtual Machine* o que faz com pequenas diferenças na implementação destes possam causar algum problema. Apesar desta desvantagem, as vantagens são largamente mais compensadoras.

Bibliografia

Brain Boosters for Business Advantage

Arthur B. VanGundy
Pfeiffer & Company

Group Decision Making under Multiple Criteria

Ching-Lai Hwang
Ming-Jeng Lin
Springer-Verlag

Facilitator's Guide to Participatory Decision-Making

Sam Kaner
New Society Publishers, Philadelphia

JAVA Design – Second Edition

Peter Coad
Mark Mayfield
Prentice Hall PTR

Mastering JAVA 1.2

John Zukowski
Sybex

The Java Developers ALMANAC 1999

Patrick Chan
Sun – microsystems
Addison Wesley

The Skilled Facilitator

Roger M.Schwarz
Jossey-Bass

Software Engineering – A practitioner's Approach, 4ª ed.

Roger S.Pressman
McGrawHill

Software User Interface Design

Deborah J. Mayhew
Prentice Hall