

Meeto

Collaboration and Social Networking

Relatório V2

Realizado por:

- Ricardo Pinto 2011144421

- Diogo Gonçalves 2011157043

Conteúdo

[Descrição 3](#_Toc405413810)

[Versão 1 3](#_Toc405413811)

[Versão 2 3](#_Toc405413812)

[Arquitetura interna 4](#_Toc405413813)

[Diagrama da arquitetura 4](#_Toc405413814)

[Servidor TCP 5](#_Toc405413815)

[Cliente TCP 5](#_Toc405413816)

[Servidor RMI 5](#_Toc405413817)

[Servidor Web 5](#_Toc405413818)

[Cliente Web 5](#_Toc405413819)

[Modelo de dados 6](#_Toc405413820)

[Especificação de protocolos 6](#_Toc405413821)

[Tratamento de Exceções 6](#_Toc405413822)

[FailOver 6](#_Toc405413823)

[Utilização da aplicação (parte Web) 7](#_Toc405413824)

# Descrição

Neste projeto foi realizado um sistema distribuído seguindo o modelo de uma aplicação cliente/servidor, para a gestão de reuniões. O objetivo é melhorar efetividade das reuniões. O nosso sistema suporta a preparação das reuniões, discussão dos itens da agenda em forma de chat em que dessa discussão são atribuídos ações e decisões chave.

### Versão 1

Para a realização do projeto v1, foram utilizados sockets para realização de tarefas de comunicação entre clientes e servidor(es) para protocolo TCP/IP. Para a comunicação entre os servidores TCP primário e o secundário, utilizados o protocolo UDP. Foram também utilizadas Threads para cada conexão de um cliente novo, e também na realização do chat.

Cliente, servidor principal, servidor secundário e servidor RMI podem estar a funcionar tanto em máquinas distintas como na mesma máquina, dependendo da configuração que for atribuída.

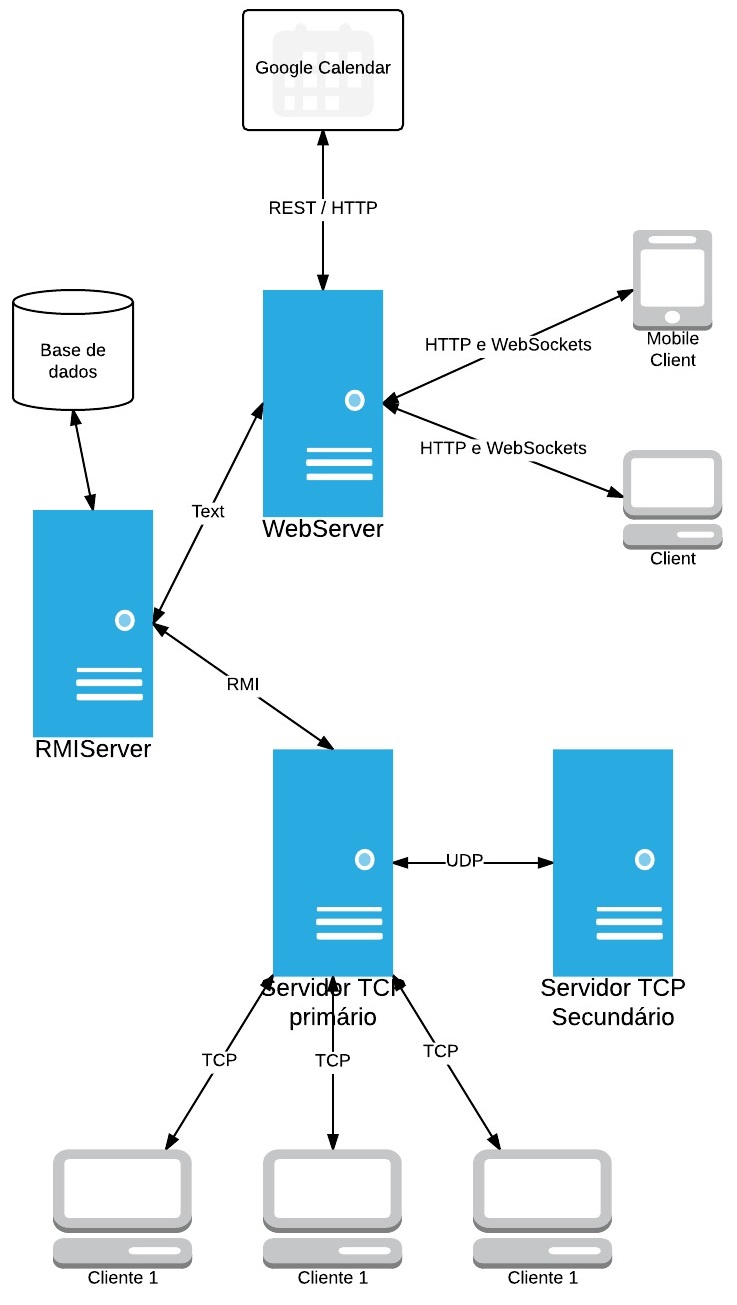
### Versão 2

Para esta versão foi feita a integração com uma base de dados, na primeira versão fazíamos o uso de ficheiros.

Na realização da 2ºa versão do projeto, foi criada uma aplicação web, as vantagens desta versão são a portabilidade e a possibilidade de ter uma interface agradável para o utilizador. Nesta versão foi utilizado o modelo MVC. Na comunicação entre o servidor web e o cliente web foi utilizado http e webSockets, http para os pedidos ao servidor, e webSockets para o cliente receber notificações em tempo real. O WebServer liga-se através de RMI ao RMIServer, este por sua vez está conectado com a base de dados.

# Arquitetura interna

### Diagrama da arquitetura



# Servidor TCP

Neste programa existem dois servidores a funcionar, um principal e um secundário. O principal a qual o cliente se conecta via TCP, e este por sua vez também se conecta ao servidor RMI e ao servidor secundário via UDP. A conexão com o cliente é feita através de uma thread. Estes dois servidores estão constantemente a fazer ping para verificar que ainda se encontram ligados. O objetivo do servidor secundário é assumir o papel de servidor principal caso o servidor principal falhe, quando não tem resposta deste, assume todas as ligações aos clientes e o programa continua a funcionar sem o cliente se aperceber do que se passa.

# Cliente TCP

O cliente contem toda a parte de interface com o utilizador, a navegação de menus e apresentação de resultados. A partir que o cliente estabelece ligação com o servidor via TCP, o cliente pode fazer pedidos e receber respostas do servidor. Também é feita tratamento de exceções no cliente e é assegurado que em caso de falha do servidor, é feita a reconexão e o a operação que o cliente estava a fazer é retomada. No cliente também é utilizada uma thread para quando é discutido um item, o cliente possa estar a ler mensagens de outros clientes enquanto escreve a sua mensagem.

# Servidor RMI

Aqui é onde é feita a conexão com a base de dados. Aqui também são recebidos e respondidos pedidos do servidor principal e do servidor Web.

# Servidor Web

É constituído por um conjunto de Actions para tratar de pedidos do utilizador web, essas ações comunicam com um Bean geral, que trata da ligação com o servidor RMI e também dos pedidos de informação para o Servidor RMI, este por sua vez vai buscar os dados á base de dados.

# Cliente Web

Conjunto de ficheiros jsp que a sua função é a interação com o utilizador web e a apresentação de dados.

# Modelo de dados

Para esta meta do projeto, usamos uma base de dados integrada, é acedida pelo RMI e basicamente tem todos os dados que aplicação precisa para funcionar tais como, informações á cerca de reuniões, utilizadores, mensagens, Todo list, etc.

# Especificação de protocolos

Na nossa aplicação são usados 3 tipos de protocolos TCP, UDP e HTTP e uma interface, RMI.

Na primeira parte, no cliente TCP é criado um socket de comunicação com o servidor principal, para o envio de texto/comandos. O uso do protocolo UDP é utilizado para a comunicação entre o servidor principal e secundário.

Na segunda, a parte web, é usado o protocolo de comunicação http para a comunicação do cliente web e o servidor.

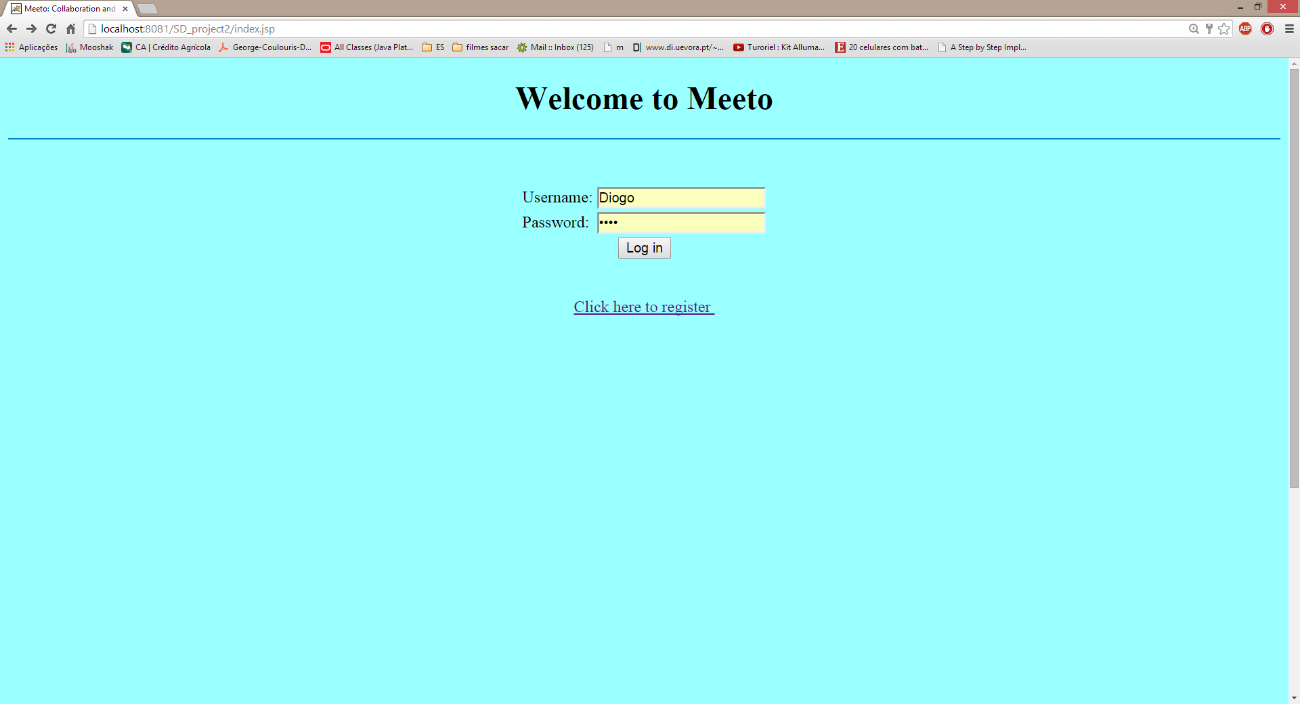
# Tratamento de Exceções

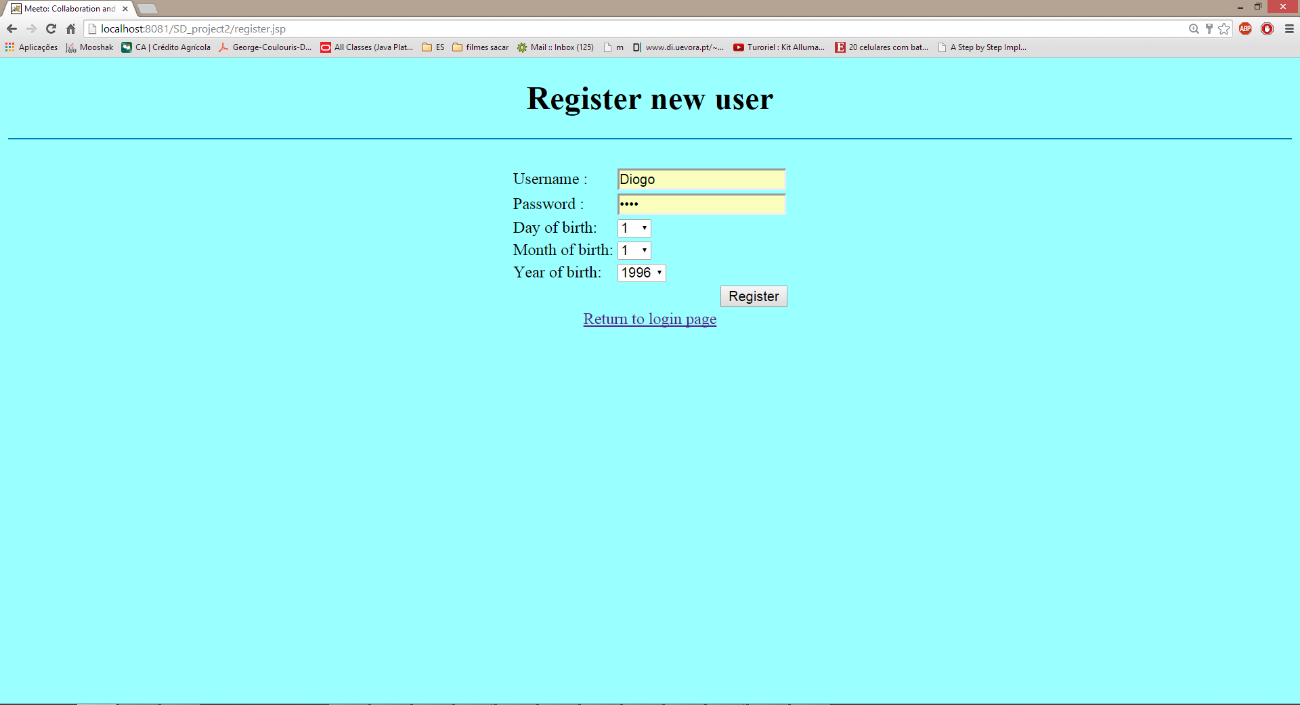
Na nossa implementação as exceções foram devidamente tratadas tanto no cliente TCP como no RMI. Caso o servidor falhe, o cliente recebe uma exceção e tenta voltar a conectar-se ao servidor para envio de informação pendente. Na parte web em caso de erro o utilizador é redirecionado para uma página de erro ou de login conforme o erro. Por exemplo, se o utilizador tenta aceder a uma página sem estar logado, é automaticamente redirecionado para a página de login.

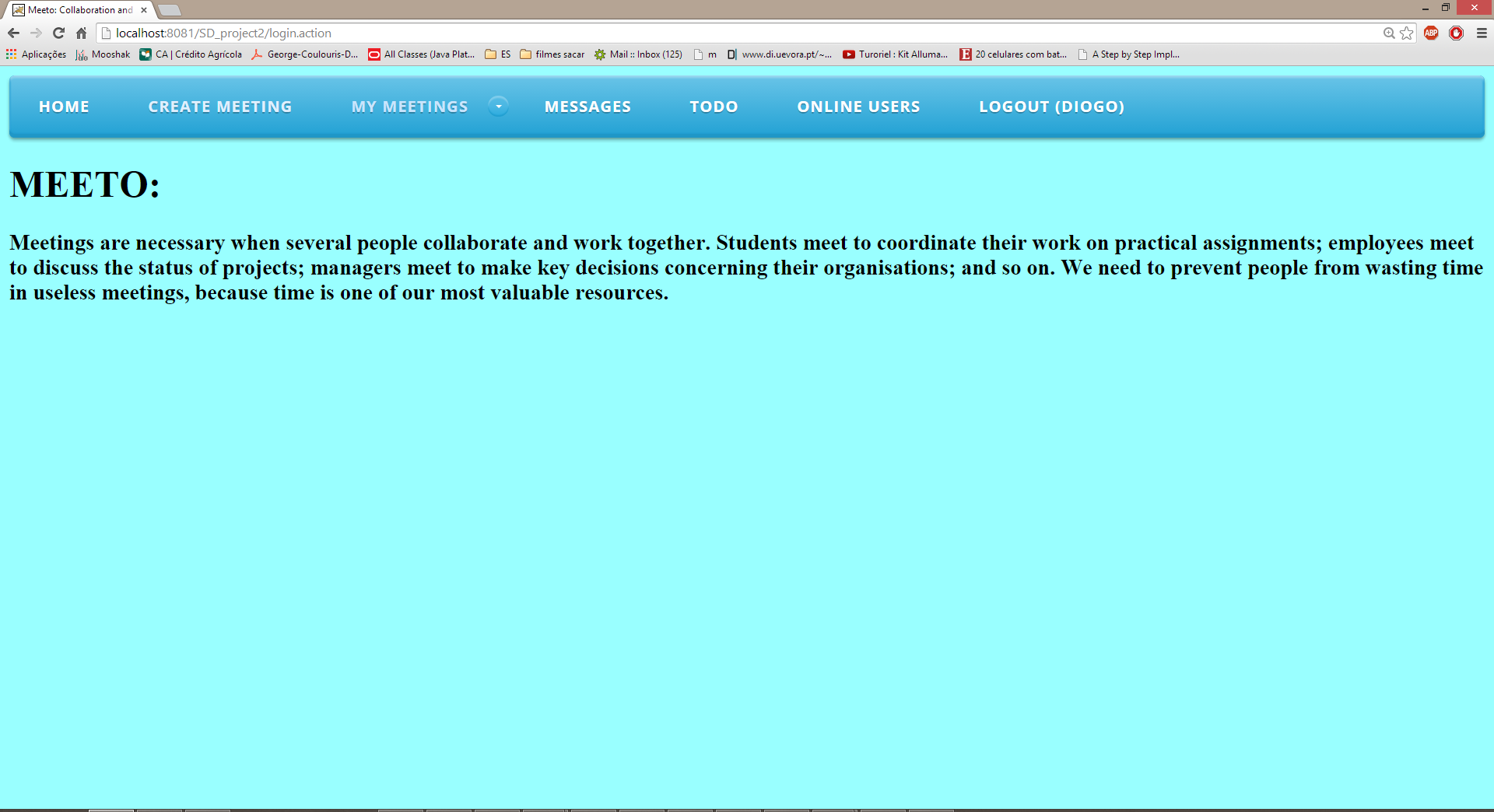
# FailOver

Na nossa aplicação temos dois servidores a correr em paralelo, o principal e o backup. Os dois servidores têm uma linha de comunicação aberta entre eles com o protocolo UDP, para controlo de atividades. Caso o servido backup não receba nenhuma resposta do servidor principal durante um determinado tempo, este assume o lugar do principal, conectando-se a todos os utilizadores que se encontravam online no momento em que o servidor principal ficou offline.

# Utilização da aplicação (parte Web)

 Esta página é o índex, onde o utilizador pode fazer login, ou registar-se:

 Aqui é pedida informação típica como nome de utilizador, passwor e data de nascimento.

Esta é a nossa Home, a partir daqui podemos navegar através das opções disponíveis ao utilizador:

* Criar reunião: são pedidas informações típicas, como nome, titulo, utilizadores convidados e agenda items.
* Minhas reuniões: dividem-se em 3 ramos que são reunuioes passadas, reuniões futuras e reuniões correntes, depois de esolher a reunião pretendida é lhe apresentada o resumo da reunião.
* Mensagens: são os convites para reuniões, que terão de ser aceites ou declinados.
* Online users: uma lista com todos os utilizadores online no momento.



Exemplo do display das meetings