

Mestrado em Engenharia Informática
Estágio
Relatório Final

Corporate Collaboration

Tiago Domingues Ferreira
tiagodf@student.dei.uc.pt

Orientador DEI:
Pedro Abreu

Orientador WIT Software, S.A.:
Frederico Lopes

Data: 12 de Julho de 2012



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

Na actualidade é difícil seguir e organizar todas as discussões que acontecem dentro de uma organização. Quando se trabalha em equipa, é vital que a informação sobre qualquer projecto esteja disponível e acessível de uma forma simples e rápida. Essa necessidade destaca-se à medida que surgem novos elementos na equipa, que necessitam de compreender todo o contexto dos desenvolvimentos, para que se possam integrar rapidamente.

O projecto desenvolvido no âmbito deste estágio curricular, visou a concepção e a implementação de uma ferramenta que integre diversos serviços de comunicação. Essa ferramenta tem como objectivo, a agilização de alguns processos dentro de uma equipa de trabalho, oferecendo um ponto centralizado de comunicação. Pretende-se com isso, melhorar a produtividade e reduzir a perda de informação relacionada com as alterações contextuais, quando se utilizam diversos serviços.

Palavras-Chave

Trabalho cooperativo através do computador, Software de colaboração, Groupware, Comunicação mediada por computador

Índice

| | |
|---|----|
| Capítulo 1 Introdução..... | 1 |
| 1.1. Entidade Proponente..... | 1 |
| 1.2. Contexto e Motivação | 2 |
| 1.3. Objectivos..... | 2 |
| 1.4. Estrutura do Documento..... | 3 |
| Capítulo 2 Estado da Arte..... | 5 |
| 2.1. Concorrentes | 5 |
| 2.1.1. Campfire | 5 |
| 2.1.2. Hip-Chat | 7 |
| 2.1.3. EchoWaves..... | 8 |
| 2.1.4. Yammer | 9 |
| 2.2. Comparação dos concorrentes..... | 10 |
| Capítulo 3 Requisitos e Arquitectura..... | 13 |
| 3.1. Prioridades..... | 13 |
| 3.2. Requisitos funcionais | 13 |
| 3.3. Requisitos não-funcionais | 17 |
| 3.4. Casos de uso..... | 17 |
| 3.5. Arquitectura..... | 23 |
| 3.5.1. Visão geral | 23 |
| 3.5.2. Perspectiva dinâmica..... | 24 |
| Esboços (<i>Mockups</i>) | 25 |
| Diagramas de comunicação..... | 25 |
| Diagramas de sequência..... | 26 |
| Capítulo 4 Metodologia e Plano de Trabalhos | 29 |
| 4.1. Metodologia..... | 29 |
| 4.2. Planeamento..... | 30 |
| 4.3. Artefactos | 30 |
| 4.4. Plano de Iterações | 31 |
| 4.5. Gestão de Riscos | 32 |
| Capítulo 5 Implementação..... | 35 |
| 5.1. Bibliotecas..... | 35 |

| | |
|---|----|
| 5.1.1. Communicator Library | 35 |
| 5.1.2. PJSIP | 36 |
| 5.1.3. XDM Library | 37 |
| 5.2. Tecnologias..... | 37 |
| 5.3. Desenvolvimentos..... | 37 |
| 5.3.1. Camada de apresentação | 37 |
| 5.3.2. Camada de lógica..... | 39 |
| Capítulo 6 Considerações Finais e Trabalho Futuro | 41 |
| Referências | 43 |

Lista de Figuras

| | |
|--|----|
| Figura 1 - Screenshot Campfire | 6 |
| Figura 2 - Screenshot HipChat..... | 7 |
| Figura 3 - Screenshot EchoWaves..... | 9 |
| Figura 4 - Screenshot Yammer..... | 9 |
| Figura 5 - Diagrama de casos de uso..... | 18 |
| Figura 6 - Visão geral da estrutura dos produtos WCS | 23 |
| Figura 7 - Arquitectura da aplicação..... | 24 |
| Figura 8 - Camadas da aplicação e a sua interacção..... | 24 |
| Figura 9 - Diagrama de comunicação para a autenticação..... | 26 |
| Figura 10 - Diagrama de sequência para a autenticação..... | 27 |
| Figura 11 - Planeamento Geral do Projecto..... | 30 |
| Figura 12 - Communicator Library e as suas dependências com outras bibliotecas | 36 |
| Figura 13 - Relação entre a camada de apresentação e a <i>Communicator Library</i> | 37 |
| Figura 14 - Padrão Model-View ^[11] | 38 |

Lista de Tabelas

| | |
|---|----|
| Tabela 1 - Planos de preços da ferramenta Campfire ^[3] | 6 |
| Tabela 2 - Plano de preços da ferramenta HipChat ^[5] | 8 |
| Tabela 3 - Matriz comparativa dos diferentes concorrentes..... | 11 |
| Tabela 4 - Interacções para o caso de uso "Autenticar Utilizador" | 18 |
| Tabela 5 - Interacções para o caso de uso "Alterar presença" | 19 |
| Tabela 6 - Interacções para o caso de uso "Adicionar contacto" | 19 |
| Tabela 7 - Interacções para o caso de uso "Alterar contacto" | 19 |
| Tabela 8 - Interacções para o caso de uso "Consultar detalhes de contacto" | 19 |
| Tabela 9 - Interacções para o caso de uso "Eliminar contacto" | 20 |
| Tabela 10 - Interacções para o caso de uso "Criar sala de discussão" | 20 |
| Tabela 11 - Interacções para o caso de uso "Abrir sala de discussão" | 20 |
| Tabela 12 - Interacções para o caso de uso "Fechar sala de discussão" | 20 |
| Tabela 13 - Interacções para o caso de uso "Eliminar sala de discussão" | 21 |
| Tabela 14 - Interacções para o caso de uso "Convidar um ou mais contactos" | 21 |
| Tabela 15 - Interacções para o caso de uso "Enviar mensagem para sala de discussão" | 21 |
| Tabela 16 - Interacções para o caso de uso "Enviar mensagem para um contacto" | 21 |
| Tabela 17 - Interacções para o caso de uso "Enviar ficheiro" | 22 |
| Tabela 18 - Interacções para o caso de uso "Enviar SMS" | 22 |
| Tabela 19 - Interacções para o caso de uso "Efectuar chamada de voz" | 22 |
| Tabela 20 - Interacções para o caso de uso "Alterar configurações" | 22 |
| Tabela 21 - Product Backlog | 32 |
| Tabela 22 - Identificação de riscos | 33 |
| Tabela 23 - Planos de Contingência | 34 |

Anexos

Anexo A – Mockups

Anexo B – Proposta de Design

Anexo C – Interface Final

Anexo D – Scrum

Anexo E – Diagramas de Comunicação

Anexo F – Diagramas de Sequência

Glossário

| Acrónimos | Significado |
|-----------|--|
| API | Application Programming Interface |
| ICE | Interactive Connectivity Establishment |
| IM | Instant Messaging |
| IMS | IP Multimedia Subsystem |
| IP | Internet Protocol |
| OMA | Open Mobile Alliance |
| REST | REpresentational State Transfer |
| RCS | Rich Communication Suite |
| RTP | Real-time Transfer Protocol |
| SDP | Session Description Protocol |
| SDK | Software Development Kit |
| SIP | Session Initiation Protocol |
| SMS | Short Message Service |
| STUN | Simple Traversal of UDP through NATs |
| TURN | Traversal Using Relay NAT |
| UI | User Interface |
| URL | Uniform Resource Locator |
| VoIP | Voice over Internet Protocol |
| WCS | WIT Communications Suite |
| WPC | WIT PC Communicator |
| WMC | WIT Mobile Communicator |
| WWC | WIT Web Communicator |
| XML | eXtensible Markup Language |

Capítulo 1

Introdução

Este documento tem como objectivo apresentar o trabalho desenvolvido e os conhecimentos adquiridos durante o estágio curricular, realizado no âmbito da disciplina de Dissertação/Estágio, do curso Mestrado em Engenharia Informática do Departamento de Engenharia Informática (DEI) da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC), no ano lectivo 2011/2012.

O estágio decorreu nas instalações da empresa WIT Software, em Coimbra, de Setembro de 2011 a Julho de 2012.

1.1. Entidade Proponente

A WIT é uma empresa especializada no desenvolvimento de software para operadores de telecomunicações móveis e televisão digital. Desenvolve ainda soluções de software para empresas de outras áreas de negócio que pretendam tirar partido da mobilidade e da convergência entre as comunicações móveis e a Internet. No final de 2011 conta com cerca de 100 colaboradores e tem sede em Coimbra, e centros de desenvolvimento no Porto, Leiria, Lisboa e San Jose (Califórnia).

A empresa está organizada em 3 unidades de negócio:

Telco: esta unidade faz desenvolvimento de software para operadores de telecomunicações móveis e tem clientes na Europa, África do Sul, e América do Sul. Esta unidade desenvolve um dos produtos fundamentais – WIT Communications Suite, uma solução de software para convergência das comunicações móveis, comunicações fixas e da Internet. Este produto tem permitido à empresa aumentar claramente as suas exportações no Mercado Europeu.

Mobile: esta unidade dedica-se ao desenvolvimento de aplicações para terminais móveis nas seguintes plataformas: iPhone/iPad, Android, Blackberry, J2ME, Symbian. A unidade tem clientes na área da banca, das utilities, dos media, assim como alguns clientes internacionais, como é o caso da TomTom e da Real Networks.

TV: esta unidade é responsável pelo desenvolvimento de uma plataforma de software que oferece o conceito de Connected TV. Tem clientes na Europa e no Estados Unidos.

1.2. Contexto e Motivação

Na actualidade, é comum a utilização de diferentes ferramentas para comunicação e partilha de conteúdos. Contudo, é difícil seguir e organizar todas as discussões que acontecem dentro de uma entidade. Uma discussão pode ser vista como um tópico de debate relacionado com um determinado projecto, cujo objectivo será fomentar a participação e a crítica de outros colaboradores, tendo em vista a evolução dos desenvolvimentos, como por exemplo: uma sessão de brainstorming. Em muitos casos, essas discussões acontecem através de longas listas de email e chat, que são sempre difíceis de ler e organizar. Outra consequência desta desorganização é a perda de produtividade, bem como a perda de informação quando se altera o contexto. A informação dispersa-se pelas várias ferramentas utilizadas, não existindo um ponto central que agregue os conteúdos.

Quando se trabalha em equipa, é fundamental que a informação esteja disponível e acessível de uma forma simples e rápida. Esta necessidade destaca-se à medida que surgem novos elementos na equipa, que necessitam de compreender todo o contexto dos desenvolvimentos, para que se possam integrar rapidamente.

Para tentar resolver estes problemas, existem diversas soluções, como os típicos serviço de comunicação instantânea – MSN Messenger, Google Talk, etc. – e existem também alguns serviços direccionados ao mercado empresarial. Os serviços de comunicação instantânea pecam por não oferecerem algumas funcionalidades importantes para o mundo empresarial, nomeadamente uma organização cuidada das discussões, persistência da informação partilhada através das várias plataformas de acesso.

Um produto direccionado para o mercado empresarial deve ter como objectivo a disponibilização de um serviço orientado ao negócio, que preencha os requisitos do mundo empresarial.

Este projecto surge no sentido de tentar colmatar a falha que se verifica ao nível da comunicação interna das organizações. Pretende oferecer um ponto central de discussões, que permita uma melhor organização da informação e conteúdos, para que se verifique um aumento de produtividade e se reduza a perda de informação.

1.3. Objectivos

Este projecto tem como objectivo o desenvolvimento de uma ferramenta de valor acrescentado, que inclua funcionalidades avançadas de comunicação, comuns em qualquer operador móvel, aliado à possibilidade de gerir e categorizar salas de discussão. O objectivo dessa ferramenta é oferecer um ambiente centralizado para todas as discussões de uma empresa, possibilitando o acesso ao histórico dos conteúdos partilhados, permitindo assim a integração no contexto para qualquer novo participante. Como um serviço de discussão moderno, deve conseguir lidar com os últimos formatos disponíveis, nomeadamente imagens, documentos, videos disponíveis na Internet, e outros.

1.4. Estrutura do Documento

Inicialmente, será apresentado o estudo do Estado da Arte, contendo as soluções mais populares existentes no mercado, que são alternativa ao produto que se pretende desenvolver. Esse estudo contém uma explicação sumária sobre cada ferramenta, incluindo as funcionalidades que mais se destacam.

Segue-se a apresentação dos requisitos identificados para o desenvolvimento da aplicação, bem como a uma visão geral da arquitectura onde se insere.

Posteriormente, apresenta-se a metodologia utilizada no processo de investigação e desenvolvimento de software. Descreve-se também o planeamento para a preparação, elaboração e conclusão deste projecto.

Em seguida, apresenta-se o trabalho desenvolvido, referindo quais as tecnologias adoptadas, finalizando com a informação mais relevante da fase de implementação.

Termina-se este documento com uma secção destinada a considerações finais e ao trabalho futuro.

Este documento está escrito de acordo com a antiga ortografia.

Capítulo 2

Estado da Arte

Actualmente, a comunicação nas empresas é feita através de vários meios, nomeadamente o e-mail e ferramentas de comunicação instantânea. Cada uma dessas ferramentas tem as suas funcionalidades específicas. Isto faz com que seja difícil organizar e seguir todas as discussões de uma forma prática e eficiente. Um dos principais problemas, é a falta de contexto quando é necessário recorrer a diferentes ferramentas de comunicação.

Para tentar resolver este problema, existem várias soluções, nomeadamente ferramentas de comunicação instantânea e de colaboração. As ferramentas de comunicação instantânea, como o MSN Messenger, Google Talk, Skype, etc., pecam por não oferecer funcionalidades focadas no mundo empresarial. Por outro lado, uma ferramenta de colaboração empresarial, deverá ter como objectivo um serviço orientado ao negócio, e pecam por não oferecerem funcionalidades avançadas.

A ferramenta a propor, deverá ter funcionalidades avançadas de comunicação e partilha de elementos multimédia; tais como chamadas de voz, salas de discussão em conferência, assim como partilha de ficheiros, incluindo um repositório para cada sala. Todas estas funcionalidades deverão estar acessíveis através de uma interface focada no negócio da empresa.

O seguinte estudo de estado da arte visa apresentar as soluções actualmente disponíveis no mercado, efectuar uma comparação das funcionalidades disponibilizadas e avaliar de que forma é que algumas dessas funcionalidades poderão ser utilizadas na concepção da solução proposta. Finalmente, será feita uma reflexão sobre a informação reunida nesta análise.

É importante realçar que as ferramentas escolhidas, estão todas disponíveis no mercado e são consideradas concorrentes no segmento que se pretende abranger. Por isso, não foi incluída qualquer ferramenta que se encontre em estado inacabado, ou que não se encontre disponível à data da realização da análise.

2.1. Concorrentes

Segue-se a apresentação das ferramentas escolhidas para esta análise.

2.1.1. Campfire

Campfire é uma ferramenta de conversação orientada para grupos, que foi lançada a 16 de Fevereiro de 2006.

Para funcionar, não é necessário instalar qualquer aplicação no computador, uma vez que funciona em qualquer navegador Web. É uma ferramenta que utiliza a tecnologia Ajax, para efectuar comunicação em tempo real. Para utilizar esta ferramenta, os utilizadores têm que criar uma sala de discussão ou serem convidados para uma. O histórico das discussões, incluindo os ficheiros partilhados, ficam guardados para futura pesquisa, a não ser que se defina especificamente o contrário. As imagens nos formatos mais populares (GIF, PNG, JPEG), são apresentadas através de uma pré-visualização dentro da própria sala de discussão.

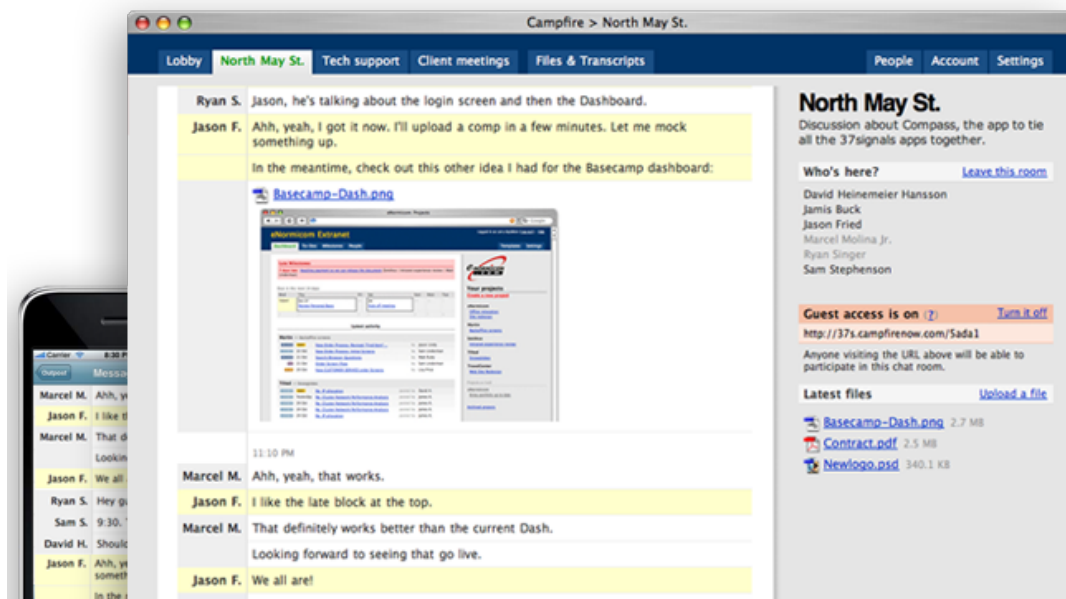


Figura 1 - Screenshot Campfire

Esta não é uma ferramenta gratuita, tendo disponíveis quatro planos de preços distintos. É oferecida a possibilidade de experimentar o serviço durante 30 dias, sem quaisquer encargos, passando posteriormente a um dos planos apresentados na tabela seguinte.

| Plano | Basic | Plus | Premium | Max |
|-------------------------|-------|-------------|-------------|--------------|
| Utilizadores | 12 | 25 | 60 | 100 |
| Espaço | 1 GB | 3 GB | 10 GB | 25 GB |
| Chamadas em Conferência | - | 500 minutos | 750 minutos | 1000 minutos |
| Preço / mês | \$12 | \$24 | \$49 | \$99 |

Tabela 1 - Planos de preços da ferramenta Campfire ^[3]

O seu modelo de negócio, funciona com base numa subscrição mensal, à qual estão associadas algumas restrições. No entanto, não deixa de ser uma ferramenta com um preço razoável para as características que apresenta, das quais, as que mais se destacam, são:

- Integração com o Basecamp, uma ferramenta de gestão de projectos, da mesma empresa;
- Chamadas em conferência;
- Histórico persistente de conversação;

- Não é necessário instalar qualquer software, pois funciona exclusivamente na página Web;
- Aplicação disponível para as plataformas móveis mais populares;
- API pública, para desenvolvimento de plugins por terceiros.

2.1.2. Hip-Chat

O HipChat é uma ferramenta lançada em Janeiro de 2010, para discussões internas/privadas de organizações, que podem ser feitas em tempo real ou em diferido.

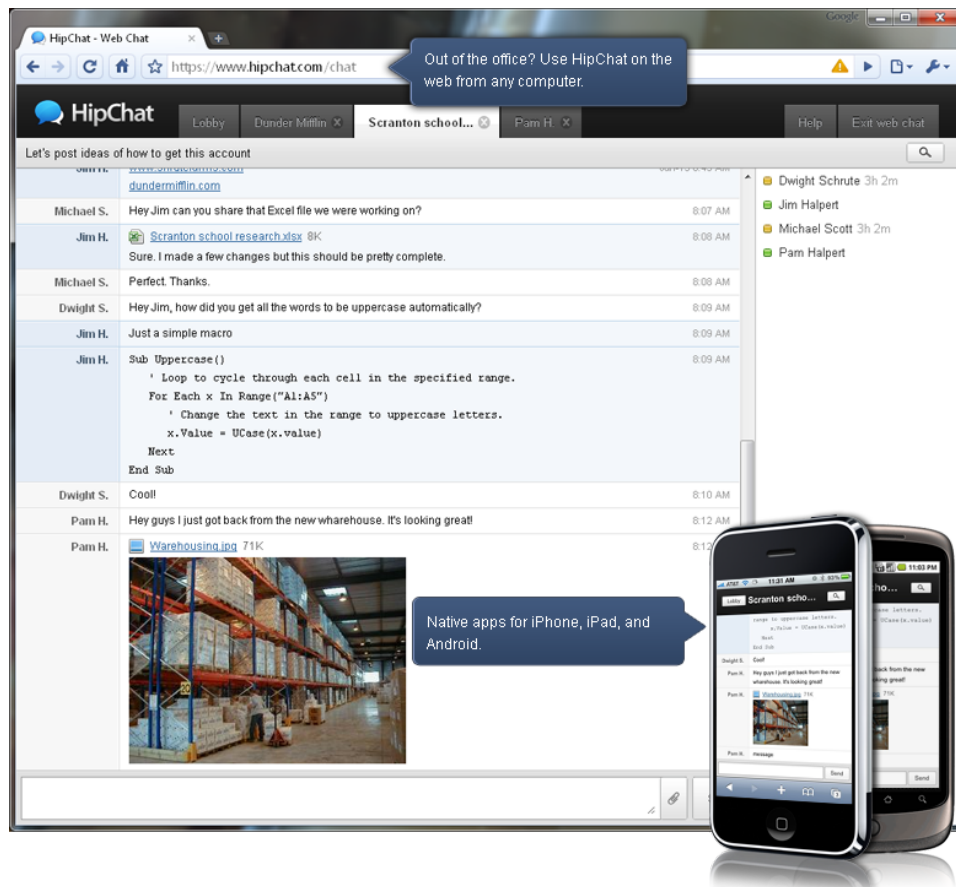


Figura 2 - Screenshot HipChat

Permite aos utilizadores participarem em salas de discussão e enviar mensagens um-para-um. É uma ferramenta que está disponível nas plataformas mais populares, oferecendo uma aplicação específica para cada dessas plataformas.

As suas funcionalidades principais são as salas de discussão um-para-muitos e um-para-um, histórico pesquisável, partilha de ficheiros, envio de SMS em discussões um-para-um. O produto está disponível tanto numa versão Web como numa aplicação que corre sobre o Adobe AIR. Uma funcionalidade muito interessante, é a possibilidade de escolher, entre várias localizações, onde se pretendem guardar os ficheiros partilhados, para que fiquem numa região mais próxima dos seus utilizadores. Isto permite que os utilizadores descarreguem os ficheiros mais rapidamente.

Esta ferramenta pode ser usada de forma gratuita, sendo nesse caso limitada apenas a quatro utilizadores, mas sem quaisquer restrições no serviço. Os planos de preços estão indicados na tabela seguinte.

| Plano | Free | Premium |
|--------------|-----------|--------------------|
| Utilizadores | 4 | Ilimitado |
| Espaço | Ilimitado | Ilimitado |
| Preço / mês | Gratuito | 1,55€ / utilizador |

Tabela 2 - Plano de preços da ferramenta HipChat [5]

Uma das vantagens desta ferramenta, é precisamente ter um plano gratuito, para quem pretende criar um pequeno grupo de discussão. O seu modelo de negócio assenta numa subscrição mensal para cada utilizador que usufrua do serviço, não tendo quaisquer restrições no resto do serviço.

As funcionalidades que mais se destacam são:

- Plano gratuito, mas limitado a quatro utilizadores;
- Discussões um-para-um e um-para-muitos;
- Possibilidade de enviar SMS quanto um utilizador não esteja disponível;
- Aplicações disponíveis para as todas as plataformas mais populares;
- Partilha de ficheiros, com a possibilidade de escolher em que região do planeta ficam guardados, para que possam estar mais próximos dos seus utilizadores.

2.1.3. EchoWaves

A EchoWaves é uma ferramenta de colaboração, que funciona através das salas de discussão, designadas por *convos*.

É um serviço totalmente gratuito que oferece comunicação através das salas de discussão criadas para o efeito. É possível participar nas discussões, através da página web, em tempo real ou em diferido, sendo que na última os utilizadores podem continuar a conversa exactamente onde terminaram, tendo acesso a todo o histórico.

A maior vantagem desta solução é o facto de ser *open-source*, o que permite a qualquer organização montar o seu ambiente de colaboração, sem quaisquer custos associados ao seu desenvolvimento ou aquisição.

Esta ferramenta tem um interface apenas na sua página Web, não oferecendo outro tipo de interface alternativo.

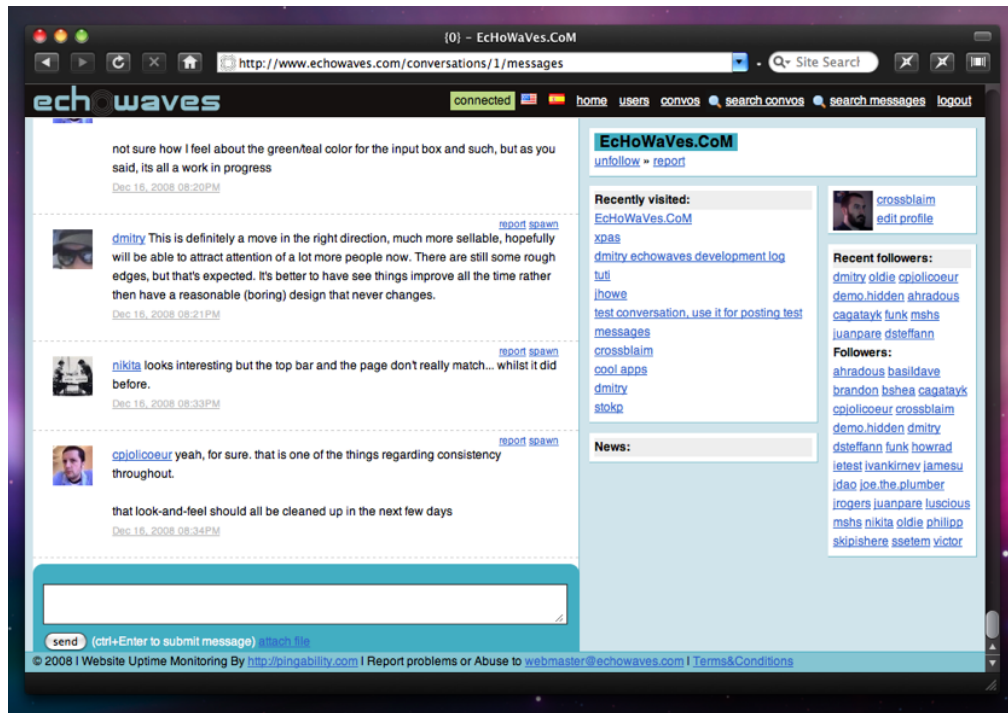


Figura 3 - Screenshot EchoWaves

A figura apresenta o interface desta aplicação, que tem funcionalidades de *Microblogging*, na qual o utilizador pode partilhar o seu estado com os outros membros da comunidade.

2.1.4. Yammer

O Yammer é uma rede social, que tem como objectivo a comunicação, de forma privada, dentro de uma instituição. É utilizada para promover a comunicação entre os diversos membros de uma organização, formando assim uma rede social de acesso exclusivo.

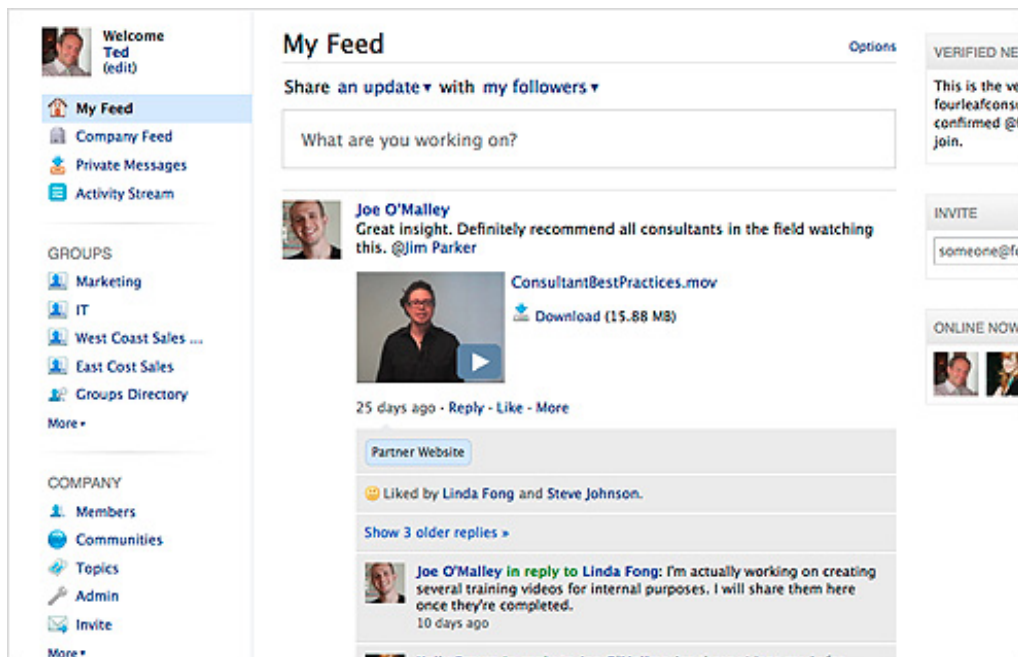


Figura 4 - Screenshot Yammer

Inicialmente, foi lançado com o intuito de servir o conceito de *microblogging* que, ao estilo do Twitter, permite aos utilizadores publicar pequenas mensagens sobre o seu estado. A partir daí, evoluiu para uma rede social completa, trazendo funcionalidades semelhantes às redes sociais mais populares, para o ambiente empresarial e institucional. O acesso ao Yammer está condicionado a um endereço escolhido pelo utilizador, por isso, apenas utilizadores com credenciais de acesso, poderão usufruir do serviço.






Este é um produto que é vendido para várias empresas e instituições, oferecendo alguma personalização do seu interface, para fazer face às necessidades e gostos de cada cliente.

Para além das típicas funcionalidades básicas de *microblogging*, como a publicação de pequenas mensagens, e a possibilidade de subscrever as publicações de outras pessoas, o Yammer oferece as seguintes funcionalidades:

- Respostas encadeadas a publicações;
- Marcadores (Tags);
- Grupos públicos e privados dentro da própria rede;
- Comunidades que funcionam como redes privadas, e permitem comunicar com utilizadores fora da organização;
- Partilha de ficheiros e imagens;
- Eventos e votações;
- Enviar mensagens directamente para múltiplos contactos;
- Presença;
- Enviar e receber mensagens em tempo real, SMS e E-mail;
- Integração com o Twitter;
- Integração com o Microsoft Sharepoint;
- Aplicação para dispositivos móveis (Android, iOS, BlackBerry e Windows Mobile);
- Restrições de segurança;
- API REST.

2.2. Comparação dos concorrentes

A tabela seguinte, pretende dar uma melhor percepção das funcionalidades presentes em cada um dos concorrentes, tendo em conta as funcionalidades-chave às quais se pretende dar maior ênfase para este projecto.

| Funcionalidades ou Características |  Campfire |  HipChat |  echowaves |  yammer |  wit |
|------------------------------------|--|---|---|--|---|
| Gratuito para vários utilizadores | | | × | | × |
| Lista de Contactos | × | × | × | × | × |
| Presença | × | × | | | × |
| Salas de conversação em tempo real | × | × | × | × | × |
| Partilha de ficheiros | × | × | × | × | × |
| Chamadas de Voz / Video | × | × | | | × |

| | | | | | |
|---|---|---|---|---|---|
| Chamadas em conferência | × | | | | × |
| Aplicação Desktop | | × | | × | × |
| Aplicação Web | × | × | × | × | × |
| Aplicações para plataforma móvel | × | × | | × | |
| Importar / Exportar conteúdos | | | | | × |
| Histórico persistente | × | × | × | × | × |
| Notificações (Email, SMS, etc.) | × | × | × | × | × |
| Acesso a participantes externos | × | × | | | × |
| Integração com redes sociais | | × | | × | |
| Pré-visualização de imagens partilhadas | × | × | × | × | × |
| Envio de SMS | | × | | × | × |

Tabela 3 - Matriz comparativa dos diferentes concorrentes

Entre as funcionalidades destacadas na tabela anterior, a presença é a possibilidade do utilizador anunciar o seu estado (Disponível, Ocupado, Ausente, Desligado). Numa ferramenta de colaboração, as salas de conversação são a sua funcionalidade principal, e por isso todas as ferramentas analisadas as possuem.

Uma vez que se pretende uma ferramenta que possua as principais funcionalidades disponíveis em qualquer operador móvel, é importante comparar a possibilidade de efectuar comunicações por voz e vídeo, sendo esta funcionalidade uma mais-valia para a ferramenta. Aliado a isso, as chamadas em conferência permitem a comunicação entre diversos intervenientes. O envio de SMS é também uma funcionalidade associada a dispositivos móveis, e por isso é uma mais-valia na comunicação com um contacto que não se encontra disponível.

A importação/exportação de conteúdos, refere-se à possibilidade de migração de discussões de longas listas de emails ou de outra ferramenta de comunicação.

Do estudo comparativo resulta que as ferramentas concorrentes apresentam funcionalidades semelhantes. No entanto, pode-se dar mais ênfase às duas primeiras ferramentas, Campfire e HipChat, pela semelhança que têm com a solução que se pretende desenvolver.

Este estudo incidiu essencialmente sobre as ferramentas mais populares deste segmento, para o mundo empresarial. Apesar de algumas ferramentas terem âmbitos diferentes, sendo focadas no *Microblogging* ou na comunicação em tempo real, todas apresentam funcionalidades que se pretendem incluir na ferramenta a desenvolver.

O HipChat é a ferramenta que mais se assemelha à desenvolvida no âmbito deste projecto. Isso é facilmente justificado pelo número de funcionalidades que estão presentes no quadro comparativo.

A análise das diferentes ferramentas, permitiu identificar e compreender melhor algumas das funcionalidades-chave que se pretendem incluir na aplicação a desenvolver.

Capítulo 3

Requisitos e Arquitectura

Esta secção tem como objectivo detalhar os requisitos do produto a desenvolver no âmbito do estágio. Segue-se a listagem dos requisitos funcionais identificados, que representam as funcionalidades principais, onde deverão ser identificadas as prioridades, os actores e as suas interacções.

3.1. Prioridades

As prioridades dos requisitos são classificadas de acordo com os seguintes termos:

- **Essencial** – significa que o requisito é vital para o funcionamento do sistema. Sem ele, o sistema não funciona. Esta prioridade deve ser atribuída a requisitos que tenham que ser implementados sem qualquer falha.
- **Bastante necessário** – requisito sem a qual o sistema pode funcionar, mas de uma forma pouco satisfatória. Tem que ser implementado, mas se não o for, o sistema continuará a funcionar.
- **Necessário** – requisito que não compromete as funcionalidades básicas do sistema, significando que o sistema funciona de forma satisfatória sem ele. Este tipo de requisito deve ser implementado, se for simples e rápido, ou deve ser adiado para versões posteriores, caso não haja tempo para o desenvolver.
- **Não essencial** – significa que o requisito deve ser implementado apenas se o seu esforço for trivial.

3.2. Requisitos funcionais

Os requisitos funcionais estão directamente relacionados com a forma como o sistema deverá funcionar. É nesta secção que as funções do sistema devem ser descritas como um conjunto de interacções, que têm um ponto de entrada e de saída.

O requisitos e as suas prioridades foram definidas em conjunto com o gestor do projecto.

Neste caso, fez-se um levantamento das funcionalidades gerais que se pretendem disponibilizar na aplicação a desenvolver, seguindo uma breve explicação sobre cada uma.

[RF001] Autenticação

É necessário autenticar o acesso ao sistema, para garantir a identificação de cada utilizador.

Prioridade: Essencial

[RF002] Lista de Contactos

Sendo um produto focado em salas de discussão, é importante que o utilizador possa aceder a um directório de contactos comum, para que possa adicionar ou remover esses mesmos contactos das salas de discussão.

Prioridade: Essencial

[RF003] Presença

Deve ser possível ao utilizador, anunciar a sua presença, para que todos os outros utilizadores possam saber qual o seu estado (Disponível, Ocupado, Ausente, Desligado).

Prioridade: Bastante necessário

[RF004] Salas de discussão

Este será o principal requisito do produto, que permite o envolvimento dos contactos, ao criar e gerir salas de discussão. As salas de discussão poderão ser públicas, em que qualquer contacto poderá participar, ou privadas, em que será necessário o convite explícito para que um contacto possa participar.

Nestas salas de discussão, podem ser partilhados diversos tipos de conteúdo, desde conteúdo escrito, a ficheiros, imagens e conteúdo que está online (YouTube, Flickr, etc.).

As salas de discussão têm a possibilidade de ter um histórico de conversação, que permite aos participantes pesquisar em todo o conteúdo. Desta forma, não é necessário que todos os participantes estejam *online* para que possam receber as mensagens partilhadas na sala, podendo visualizar o histórico sempre que voltem a entrar no sistema.

Prioridade: Essencial

[RF005] Comunicação em tempo real

Este requisito está directamente relacionado com o anterior, uma vez que é através das salas de discussão que a comunicação é efectuada. Um utilizador deve poder enviar mensagens instantâneas para um indivíduo (1-para-1), ou para múltiplos indivíduos (1-para-N). Para um utilizador receber uma mensagem, terá que estar autenticado no sistema. Se o utilizador não estiver online, poderá receber uma SMS, no caso de ser uma comunicação 1-para-1.

Prioridade: Essencial

[RF006] Partilha de ficheiros

Deve ser possível partilhar qualquer tipo de ficheiros numa sala de discussão, mantendo um repositório acessível de forma rápida dentro da sala de discussão. Caso um ficheiro partilhado seja uma imagem, todos os participantes poderão pré-visualizar a imagem num tamanho reduzido.

Prioridade: Essencial

[RF007] Chamadas de Voz

As chamadas de voz, requerem que existam dois intervenientes. Esta funcionalidade permite que a comunicação em tempo real se efectue com recurso a áudio, favorecendo assim o meio de contacto.

Prioridade: Bastante necessário

[RF008] Chamadas de Voz em Conferência

Esta funcionalidade está directamente relacionada com a anterior, diferindo apenas na possibilidade de ter mais intervenientes. Deve ser possível a um utilizador iniciar uma conferência, através da selecção de múltiplos contactos ou de um grupo da lista de contactos.

Prioridade: Necessário

[RF009] Modo offline

Esta funcionalidade pressupõe a possibilidade de se trabalhar com a aplicação em modo offline. Isto significa que o utilizador pode continuar a comunicar, mesmo que o serviço não se encontre disponível, sendo o conteúdo sincronizado logo após se verificar a disponibilidade do serviço.

Para que o utilizador possa efectuar o login, sem que o serviço esteja disponível, é necessário ter-se autenticado pelo menos uma vez anteriormente, para que a aplicação guarde informação sobre as suas credenciais de acesso.

Prioridade: Necessário

[RF010] Importar / Exportar

É importante que os utilizadores possam importar conteúdos de discussões, de outras aplicações, nomeadamente dos clientes de E-mail mais populares, ou do MSN Messenger.

Isto permite uma rápida migração de quaisquer conteúdos mais antigos para as salas de discussão, permitindo assim manter todo o histórico.

Igualmente importante é a possibilidade de exportar os conteúdos de uma determinada sala de discussão, incluindo os ficheiros partilhados.

Prioridade: Necessário

[RF011] Notificações

As notificações são uma funcionalidade importante, pois permite que o utilizador tenha conhecimento do que está a acontecer, como por exemplo: novas mensagens ou novas chamadas de voz.

O utilizador deve ser notificado por e-mail ou SMS de quaisquer novidades, ou caso já não se autentique no sistema à um longo período de tempo.

Prioridade: Bastante necessário

[RF012] Convidar utilizadores externos

Deve ser possível convidar utilizadores que não tenham qualquer associação com o serviço, para que se possam efectuar discussões *Ad-Hoc*. Isto deve ser feito, recebendo um URL para aceder a um interface Web, que lhes permitirá comunicar por escrito com os participantes dessa sala de discussão.

Prioridade: Não essencial

3.3. Requisitos não-funcionais

Os requisitos funcionais, são os critérios que devem ser utilizados para avaliar o funcionamento do sistema, e não o seu comportamento.

[RNF001] Usabilidade

O interface é uma parte vital do sucesso da aplicação. O público-alvo, tendo uma agenda bastante preenchida, não tendo tempo a despendar na aprendizagem de mais uma nova aplicação, por isso o interface deverá ser simples e intuitivo. Deve ter um interface amigável e com um estilo profissional, de forma a que os utilizadores não se cansem rapidamente dele.

Prioridade: Essencial

[RNF002] Performance

A aplicação deverá ter um comportamento fluido, que não faça de operações simples, uma tarefa demorada para o utilizador. Não é um requisito, mas deve ser tomado em conta como um requisito de qualidade.

Prioridade: Bastante necessário

[RNF003] Hardware e Software

A aplicação deverá funcionar nos sistemas operativos mais populares, sem a necessidade de recorrer a qualquer hardware específico.

Prioridade: Bastante necessário

3.4. Casos de uso

Esta secção serve para descrever com mais detalhe as interacções associadas aos requisitos funcionais apresentados anteriormente.

A aplicação deverá interagir directamente com o utilizador, que é considerado aqui o actor. O utilizador é quem tem acesso total às funcionalidades da aplicação.

Em seguida apresenta-se o diagrama de casos de uso, onde estão reflectidos os principais requisitos funcionais da aplicação.

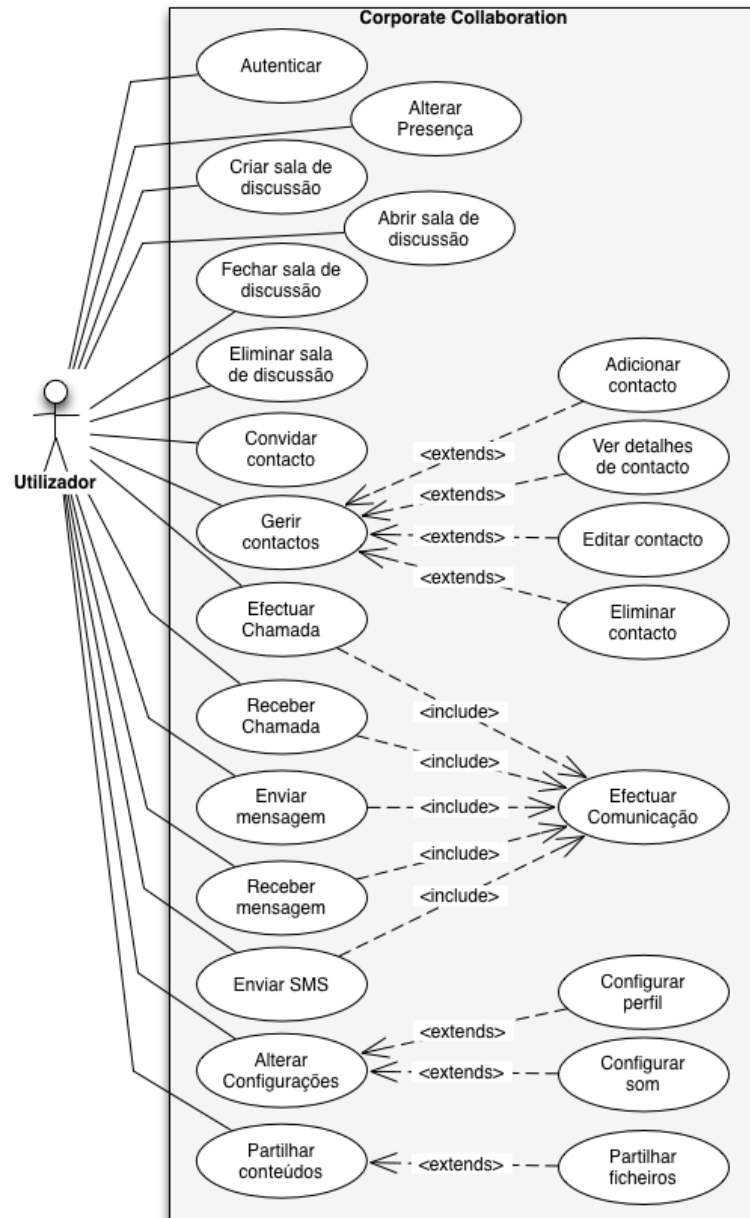


Figura 5 - Diagrama de casos de uso

Para melhor demonstrar o tipo de interacção que se espera entre o utilizador e o sistema, apresenta-se em seguida as sequências estímulo-resposta para cada um dos casos de uso.

| Autenticar Utilizador (C01) | |
|---|---|
| Actores | Utilizador |
| Objectivos | Autenticar os utilizadores no sistema, para garantir o acesso exclusivo |
| Pré-condições | O utilizador tem que possuir uma conta no sistema |
| Casos de uso relacionados | |
| Interacções | |
| Estímulo | Resposta |
| Utilizador introduz nome de utilizador e password e carrega no botão de Login | Autenticação efectuada com sucesso ou Autenticação rejeitada por credenciais inválidas ou Autenticação rejeitada por falha na ligação ao servidor |

Tabela 4 - Interacções para o caso de uso "Autenticar Utilizador"

| Alterar estado de presença (C02) | |
|--|--|
| Actores | Utilizador |
| Objectivos | Alterar o estado de presença para Disponível, Ocupado, Ausente ou Desligado. |
| Pré-condições | O utilizador tem que estar autenticado no sistema |
| Casos de uso relacionados | C01 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador carrega no botão de estado de presença | A aplicação deverá apresentar um menu com a lista de estados de presença disponíveis |
| Com a lista de estados de presença visível, o utilizador selecciona uma das opções | A aplicação deverá alterar a cor do botão de estado de presença, para a respectiva cor associada ao estado de presença escolhido |

Tabela 5 - Interacções para o caso de uso "Alterar presença"

| Adicionar contacto (C03) | |
|--|--|
| Actores | Utilizador |
| Objectivos | Adicionar um contacto ao directório de contactos |
| Pré-condições | O utilizador tem que estar autenticado no sistema |
| Casos de uso relacionados | C01 |
| Interacções | |
| Estímulo | Resposta |
| O utilizador carrega no botão de adicionar contacto, que deverá estar junto ao directório de contactos | A aplicação deverá mostrar um ecrã com um formulário para preenchimento dos dados do novo contacto |
| Já com o formulário preenchido, o utilizador carrega no botão Adicionar | A aplicação deverá fechar o ecrã de novo contacto e adicionar o contacto ao directório |

Tabela 6 - Interacções para o caso de uso "Adicionar contacto"

| Alterar contacto (C04) | |
|---|---|
| Actores | Utilizador |
| Objectivos | Alterar informações relacionadas com um determinado contacto |
| Pré-condições | O utilizador tem que estar autenticado no sistema e tem que ter adicionado um contacto ao directório |
| Casos de uso relacionados | C01, C03 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona um contacto do directório, carrega no botão Editar ou selecciona a opção Editar do menu de contexto | A aplicação apresenta um ecrã com um formulário pré-preenchido com os dados do contactos, para que o utilizador possa efectuar alterações |
| Após efectuar as alterações, o utilizador carrega no botão Guardar | A aplicação fecha o ecrã de alteração de contacto e deve actualizar o directório |

Tabela 7 - Interacções para o caso de uso "Alterar contacto"

| Consultar detalhes de contacto (C05) | |
|---|---|
| Actores | Utilizador |
| Objectivos | Consultar os dados relacionados com um determinado contacto |
| Pré-condições | O utilizador tem que estar autenticado no sistema e tem que ter adicionado um contacto ao directório |
| Casos de uso relacionados | C01, C03 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona um contacto do directório, carrega no botão Ver Detalhes ou selecciona a opção Ver Detalhes no menu de contexto | A aplicação deve apresentar um ecrã com os dados do contacto, com opção para alterar ou fechar o ecrã |
| Utilizador selecciona a opção Alterar Contacto | É executado o caso de uso C04 |

Tabela 8 - Interacções para o caso de uso "Consultar detalhes de contacto"

| Eliminar contacto (C06) | |
|---|--|
| Actores | Utilizador |
| Objectivos | Eliminar um contacto do directório de contactos |
| Pré-condições | O utilizador tem que estar autenticado no sistema e tem que ter adicionado um contacto ao directório |
| Casos de uso relacionados | C01, C03 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona um contacto do directório, carrega no botão Eliminar ou selecciona a opção Eliminar no menu de contexto | A aplicação deverá eliminar o contacto do directório |

Tabela 9 - Interacções para o caso de uso "Eliminar contacto"

| Criar sala de discussão (C07) | |
|--|--|
| Actores | Utilizador |
| Objectivos | Criar uma sala de discussão que permita a troca de mensagens e a partilha de conteúdos entre diversos actores |
| Pré-condições | O utilizador tem que estar autenticado no sistema |
| Casos de uso relacionados | C01 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador carrega no botão de criação de nova sala de discussão | Surgirá um ecrã para preenchimento do nome da sala de discussão |
| Após preencher os dados relativos à nova sala de discussão, terminará carregando no botão OK | Sala de discussão é criada e apresentada no ecrã, sem quaisquer contactos ou conteúdos associados. É adicionada uma nova entrada na lista de salas de discussão. |

Tabela 10 - Interacções para o caso de uso "Criar sala de discussão"

| Abrir sala de discussão (C08) | |
|--|---|
| Actores | Utilizador |
| Objectivos | Abrir uma sala de discussão |
| Pré-condições | O utilizador tem que estar autenticado no sistema e tem que já ter criado uma sala de discussão |
| Casos de uso relacionados | C01, C07 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona item da lista de salas de discussão, e escolhe a opção Abrir no menu de contexto ou Utilizador faz duplo-clique num elemento da lista de salas de discussão | A aplicação deverá acrescentar um nova "Tab" à barra de salas de discussão abertas, e apresentar no ecrã a sala de discussão com os conteúdos e contactos já associados |

Tabela 11 - Interacções para o caso de uso "Abrir sala de discussão"

| Fechar sala de discussão (C09) | |
|---|--|
| Actores | Utilizador |
| Objectivos | Fechar uma sala de discussão |
| Pré-condições | O utilizador tem que estar autenticado no sistema e tem que já ter criado uma sala de discussão |
| Casos de uso relacionados | C01, C07 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador carrega no botão de fechar a sala de discussão, disponível na barra que contém as "Tabs" com as salas de discussão abertas | Sala de discussão é fechada, mas os conteúdos permanecem guardados e são apresentadas notificações de novas mensagens ou conteúdos. É possível voltar a abrir a sala de discussão. |

Tabela 12 - Interacções para o caso de uso "Fechar sala de discussão"

| Eliminar sala de discussão (C10) | |
|---|--|
| Actores | Utilizador |
| Objectivos | Eliminar uma sala de discussão |
| Pré-condições | O utilizador tem que estar autenticado no sistema e tem que já ter criado uma sala de discussão |
| Casos de uso relacionados | C01, C07 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona um elemento da lista de salas de discussão,e escolhe a opção Eliminar no menu de contexto | Sala de discussão é fechada e o elemento é eliminado da lista de salas de discussão. Os conteúdos são igualmente eliminados, e não são apresentadas notificações de novas mensagens ou conteúdos. Não é possível voltar a abrir a sala de discussão. |

Tabela 13 - Interações para o caso de uso "Eliminar sala de discussão"

| Convidar um ou mais contactos para sala de discussão (C11) | |
|---|--|
| Actores | Utilizador |
| Objectivos | Convidar um ou mais contactos a participar numa sala de discussão |
| Pré-condições | O utilizador tem que estar autenticado no sistema e tem que já ter criado uma sala de discussão |
| Casos de uso relacionados | C01, C07 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador carrega no botão para convidar contacto | A aplicação deverá apresentar um ecrã com a lista de contactos disponíveis |
| Utilizador selecciona um ou mais contactos da lista apresentada e carrega no botão Convidar | A aplicação deverá fechar o ecrã com a lista de contactos, indicar que os contactos seleccionados foram convidados |

Tabela 14 - Interações para o caso de uso "Convidar um ou mais contactos"

| Enviar mensagem para uma sala de discussão (C12) | |
|---|---|
| Actores | Utilizador |
| Objectivos | Enviar mensagem para uma sala de discussão |
| Pré-condições | O utilizador tem que estar autenticado no sistema, tem que já ter criado uma sala de discussão, convidado pelo menos um contacto e ter a sala de discussão aberta |
| Casos de uso relacionados | C01, C07, C08, C11 |
| Interacções | |
| Estímulo | Resposta |
| Utilizado preenche o campo de texto com mensagem a enviar e carrega no botão Enviar | A mensagem é mostrada na sala de discussão |

Tabela 15 - Interações para o caso de uso "Enviar mensagem para sala de discussão"

| Enviar mensagem para um contacto (C13) | |
|--|---|
| Actores | Utilizador |
| Objectivos | Enviar mensagem para um determinado contacto |
| Pré-condições | O utilizador tem que estar autenticado no sistema e ter adicionado pelo menos um contacto |
| Casos de uso relacionados | C01, C03 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona um elemento do directório de contactos, escolhe a opção Enviar Mensagem no menu de contexto, ou faz duplo-clique sobre esse elemento | A aplicação deverá abrir uma nova sala de discussão para o contacto seleccionado |

Tabela 16 - Interações para o caso de uso "Enviar mensagem para um contacto"

| Enviar ficheiro para uma sala de discussão (C14) | |
|---|--|
| Actores | Utilizador |
| Objectivos | Enviar um ficheiro para uma sala de discussão |
| Pré-condições | O utilizador tem que estar autenticado no sistema e tem que já ter criado uma sala de discussão e convidado pelo menos um contacto |
| Casos de uso relacionados | C01, C07, C11 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador carrega no botão de enviar ficheiro, junto à caixa de texto para enviar mensagem | A aplicação apresenta um ecrã com o sistema de ficheiros para que o utilizador possa escolher o ficheiro a enviar |
| Utilizador escolhe ficheiro e carrega no botão Enviar | É apresentada uma barra de progresso na sala de discussão, indicando o estado do envio do ficheiro |

Tabela 17 - Interacções para o caso de uso "Enviar ficheiro"

| Enviar SMS (C15) | |
|---|--|
| Actores | Utilizador |
| Objectivos | Enviar um SMS para um contacto |
| Pré-condições | O utilizador tem que estar autenticado no sistema e ter adicionado pelo menos um contacto |
| Casos de uso relacionados | C01, C03 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona um contacto do directório, escolhe a opção Enviar SMS no menu de contexto | A aplicação deverá apresentar um ecrã com um formulário para introduzir o conteúdo da mensagem SMS |
| Utilizador preenche o formulário e carrega no botão Enviar | Ecrã de enviar SMS é fechado e a mensagem é enviada para o contacto correspondente |

Tabela 18 - Interacções para o caso de uso "Enviar SMS"

| Efectuar uma chamada de voz (C16) | |
|--|---|
| Actores | Utilizador |
| Objectivos | Efectuar uma chamada de voz |
| Pré-condições | O utilizador tem que estar autenticado no sistema e ter adicionado pelo menos um contacto |
| Casos de uso relacionados | C01, C03 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona um contacto do directório, escolhe a opção Ligar no menu de contexto | A aplicação deverá apresentar um ecrã com a chamada de voz activa no estado "A Chamar" |

Tabela 19 - Interacções para o caso de uso "Efectuar chamada de voz"

| Alterar configurações (C17) | |
|---|--|
| Actores | Utilizador |
| Objectivos | Alterar configurações da aplicação |
| Pré-condições | O utilizador tem que estar autenticado no sistema |
| Casos de uso relacionados | C01 |
| Interacções | |
| Estímulo | Resposta |
| Utilizador selecciona a opção Configurações | A aplicação deverá apresentar um ecrã com as opções de configurações disponíveis |

Tabela 20 - Interacções para o caso de uso "Alterar configurações"

3.5. Arquitectura

A arquitectura é reconhecida como um elemento crítico para o sucesso de qualquer sistema de software. Esta secção visa apresentar uma visão geral da arquitectura do software realizada no âmbito deste projecto.

3.5.1. Visão geral

O presente projecto visa a criação de uma ferramenta de colaboração, que agilize a comunicação e o acesso à informação dentro de uma entidade. Esta ferramenta integra-se na WIT Communications Suite (WCS), que é um conjunto de aplicações desenvolvidas na entidade onde se desenvolve este projecto, sendo que todos os produtos da WCS partilham serviços comuns. O seguinte diagrama, demonstra a estrutura da WCS enquadrada numa rede de serviços designada por *IP Multimedia Subsystem (IMS)*.

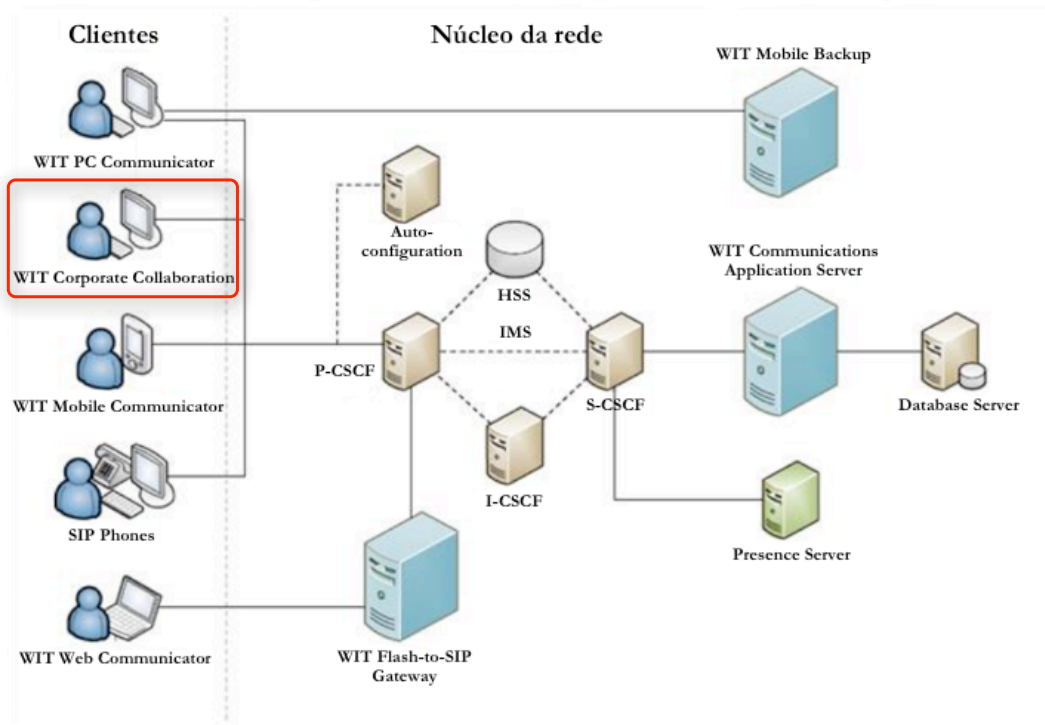


Figura 6 - Visão geral da estrutura dos produtos WCS

Numa rede IMS, os clientes ligam-se directamente ao P-CSCF, que redirecciona os pedidos para os respectivos módulos. O protocolo de comunicação utilizado é o SIP/RTP. Outros clientes, que utilizem diferentes protocolos, ligam-se através de uma *gateway* que trata da tradução das mensagens, redireccionando-as em seguida para o IMS. Este projecto foca-se essencialmente no cliente WIT Corporate Collaboration, assinalado a vermelho na figura anterior.

A aplicação a desenvolver enquadra-se numa arquitectura cliente-servidor, sendo que toda a estrutura dos serviços já se encontra funcional. De uma forma geral, a arquitectura da aplicação é comum a outras que englobam o WCS, e pode-se representar através da figura seguinte.

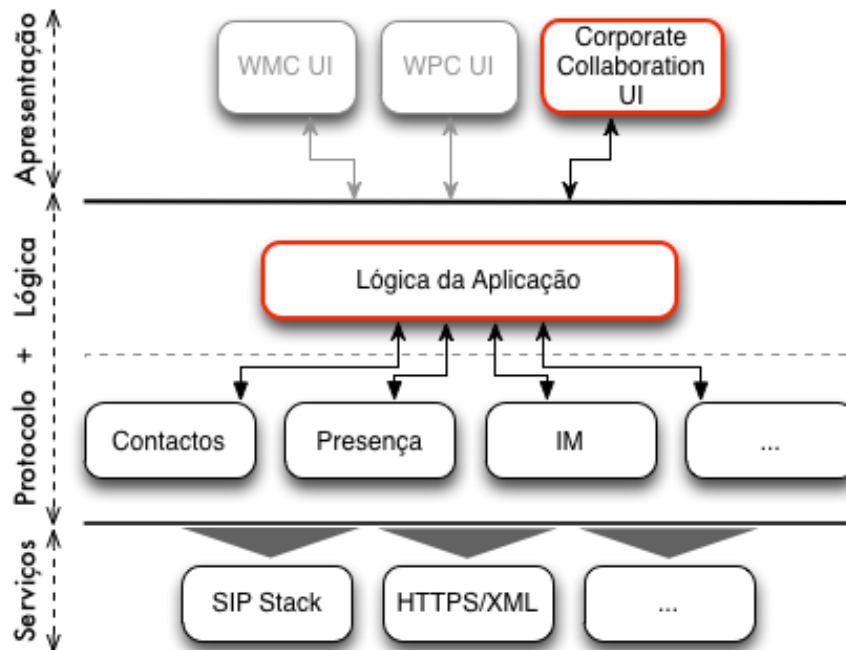


Figura 7 - Arquitectura da aplicação

Na Figura 7, podemos verificar que se trata de uma arquitectura com múltiplas camadas. Existem três camadas principais, a camada de apresentação, camada de lógica e a camada de acesso aos serviços.

A aplicação é composta por diversas bibliotecas, desenvolvidas em C/C++, que separam as várias funcionalidades. Pode-se destacar a *Communicator Library*, que é a principal responsável pela lógica da aplicação. A sua arquitectura, bem como de outras bibliotecas, está explicada no ponto 5.1.1. Esta biblioteca emite eventos sempre que existe uma alteração de estado de qualquer serviço, sendo esses eventos capturados na camada de apresentação, conduzindo a uma actualização da interface gráfica, de acordo com o estado actual. Essa relação pode ser representada pela seguinte figura:

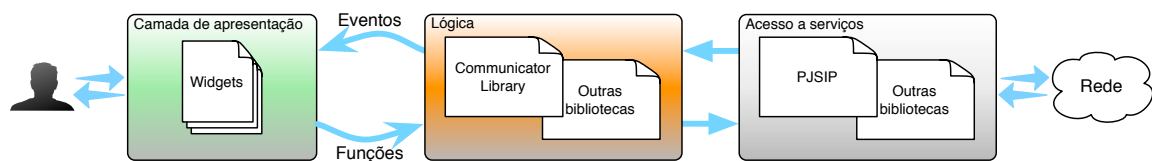


Figura 8 - Camadas da aplicação e a sua interacção

Desta forma, as camadas ficam separadas e é possível criar uma abstracção que permite um desenvolvimento mais rápido e mais fácil. Uma vez que a camada de lógica contém código comum a outros projectos, é necessário efectuar desenvolvimentos que sejam específicos a cada projecto.

3.5.2. Perspectiva dinâmica

Nesta secção pretende-se apresentar as diferentes fases do modelo ao longo do tempo. Foram criados diagramas de interacção para alguns casos de uso.

Os esboços têm como principal objectivo ajudar a compreender as interacções sob a óptica do utilizador, sendo que os diagramas de interacção especificam a dinâmica da sequência de

interacções entre o utilizador e os objectos da interface gráfica, e destes com outros componentes do projecto.

Por motivos de confidencialidade, a entidade onde se desenvolve este projecto, não permite a divulgação de quaisquer interfaces gráficas ou código fonte. Por esse motivo, esta secção fará referência aos respectivos esboços presentes em anexo.

Esboços (*Mockups*)

Foram elaborados esboços de vários ecrãs da ferramenta a desenvolver, tendo em conta os requisitos funcionais apresentados anteriormente. Embora não tenham sido cobertos todos os requisitos, os esboços apresentados foram considerados suficientes para detalhar o comportamento da aplicação.

Na Figura 1 do Anexo A, está representado o ecrã de autenticação no sistema. Este ecrã serve exclusivamente para que o utilizador consiga, mediante credenciais, ter acesso às salas de discussão e aos seus conteúdos.

A Figura 2, representa o ecrã principal da aplicação, que contém à esquerda o componente de gestão das salas de discussão, ao centro o conteúdo da sala de discussão seleccionada, e à direita a lista de contactos e de ficheiros partilhados na mesma sala de discussão.

O ecrã principal é onde a aplicação permite o acesso directo ou indirecto a todas as suas funcionalidades. Desta forma, pensou-se que seria útil ter acesso imediato a todas as salas de discussão que tenham sido criadas ou para as quais o utilizador tenha sido convidado. Por outro lado, quando uma sala de discussão está aberta, é útil ter acessível a informação mais relevante, seja a lista de contactos ou os ficheiros aí partilhados.

A Figura 3 apresenta o ecrã de configurações, onde é possível ao utilizador definir alguns comportamentos da aplicação, sejam definições gerais, de alertas, de som ou outras.

A Figura 4 demonstra o ecrã de criação de uma nova sala de discussão. Neste ecrã o utilizador preenche o nome da sala de discussão, entre outras opções.

Diagramas de comunicação

Os diagramas de comunicação modelam as interacções entre objectos ou partes do projecto em termos de sequências de mensagens. Neste tipo de diagramas, o tempo não é representado de maneira explícita, pelo que as sucessivas mensagens têm de ser numeradas exprimindo assim a ordem de envio.

Foram elaborados diagramas de comunicação para alguns casos de uso. A título de exemplo, apresenta-se em seguida um dos diagramas de comunicação, seguido da respectiva explicação. Todos os outros diagramas deverão ser consultados em anexo.

Visual Paradigm for UML Community Edition(University of Coimbra)

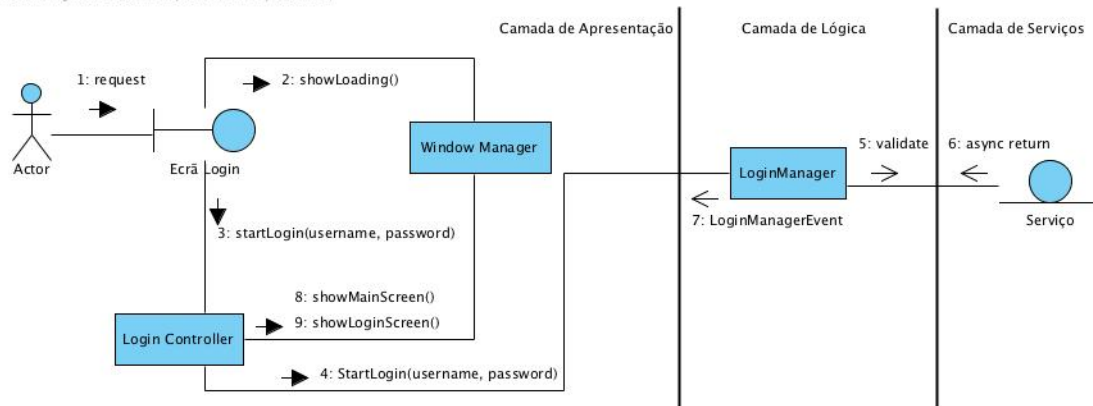


Figura 9 - Diagrama de comunicação para a autenticação

O diagrama anterior ilustra a troca de mensagens realizada entre os objectos de sistema no caso de uso de autenticação do utilizador (C01). Neste caso de uso, o utilizador solicita a acção de autenticação na aplicação, a partir do ecrã de login. Esse pedido é reencaminhado para o controlador de login, ao mesmo tempo que é feito um pedido de redireccionamento para o ecrã de carregamento. Ao receber o pedido, o controlador trata de invocar o método correspondente no LoginManager, objecto pertencente à camada de lógica, que é responsável pela gestão da autenticação no sistema. Em seguida, o LoginManager trata de invocar o método correspondente na camada de acesso aos serviços, que irá enviar uma mensagem ao respectivo serviço. A resposta do serviço é assíncrona, o que significa que o serviço enviará resposta assim que processar a mensagem. Quando a mensagem é recebida, o LoginManager trata-a e devolve a resposta para a camada de apresentação, sob a forma de um evento, que será “apanhado” pelo LoginController. Para terminar, o LoginController mostra o ecrã principal da aplicação caso a autenticação tenha sido efectuada com sucesso, caso contrário volta para o ecrã inicial de login.

Diagramas de sequência

Nesta secção pretende-se demonstrar e explicar, através de diagramas de sequência, o fluxo das mensagens na interacção dos diversos objectos/componentes.

Os diagramas de sequência mostram interacções entre objectos/componentes, insistindo sobre a perspectiva temporal dos mesmos, ao contrário do que acontece com os diagramas de comunicação.

De forma a compreender as diferenças entre este tipo de diagramas e os apresentados na secção anterior, o caso de uso apresentado de seguida, é o mesmo.

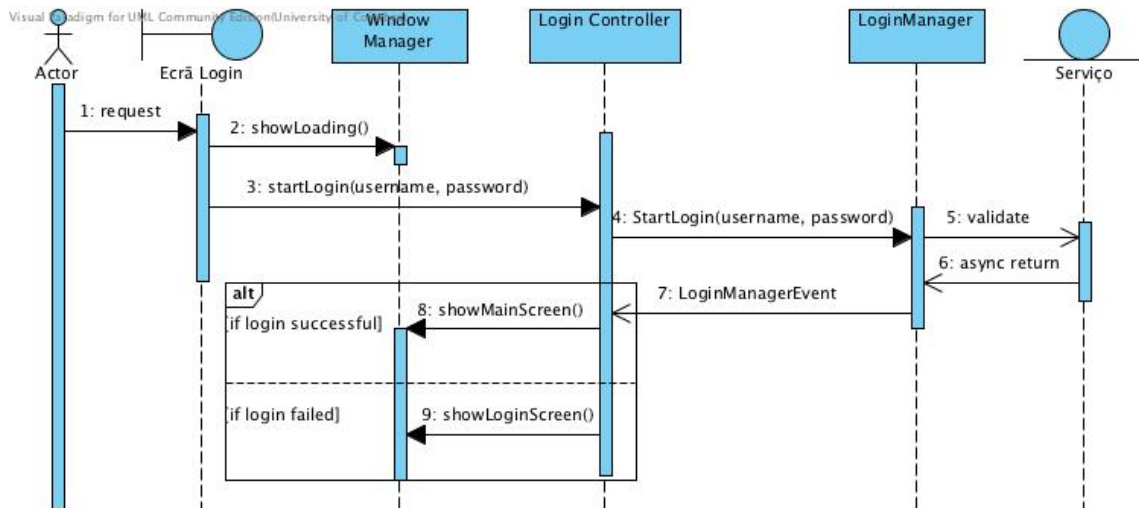


Figura 10 - Diagrama de sequência para a autenticação

Nos diagramas de sequência, cada objecto/componente é representado por um rectângulo ou outro elemento, e uma linha vertical, denominada por linha de vida do objecto. Os objectos comunicam entre si, trocando mensagens representadas por setas horizontais, orientadas no sentido emissor-receptor. Geralmente as mensagens são invocações de métodos dos objectos ou acções. Por exemplo, “request” significa que houve uma acção do utilizador que vai gerar um determinado comportamento da aplicação.

A sequência de mensagens é a mesma que nos diagramas de comunicação, mas neste tipo de diagramas é dada mais importância à perspectiva temporal da interacção.

Os diagramas são semelhantes em cada caso de uso, sendo que existe sempre uma interacção do utilizador com um determinado ecrã/componente, que se traduz num pedido directo ao objecto que controla (*controller*) a funcionalidade. Esse objecto, por sua vez, tratará de redireccionar o pedido para a *Communicator Library*, que tem a responsabilidade de traduzir a mensagem e enviar para o serviço respectivo. As mensagens para os serviços são sempre assíncronas, pois dependem da disponibilidade do próprio serviço. Após o processamento da mensagem, é enviada uma resposta que é traduzida e tratada na camada de lógica, sendo a camada de apresentação notificada de qualquer alteração de estado. A integração entre as várias camadas pode ser consultada com mais detalhe no Capítulo 5.

Capítulo 4

Metodologia e Plano de Trabalhos

A metodologia e o plano de trabalhos, relativamente ao processo de desenvolvimento, são apresentados nesta secção.

4.1. Metodologia

Para o processo de desenvolvimento deste projecto, adoptou-se uma metodologia ágil, baseada nalguns conceitos do *Scrum*.

A metodologia adoptada não segue toda a especificação do *Scrum*, mas adopta os conceitos base que a constituem. Isto permite obter uma grande flexibilidade, tanto no desenvolvimento como na gestão, resultando numa melhor optimização das estimativas e do controlo de risco, adjacentes a cada iteração.

As iterações, ou *Sprints*, na qual se baseia esta metodologia, têm a duração de 14 a 30 dias, e são feitos com base numa lista de tarefas definidas na fase de análise e especificação de requisitos. Essa lista designa-se por *Product Backlog*, que contém todas as funcionalidades que se pretendem incluir no produto, assim como as suas prioridades. Cada *Sprint*, deverá conter um número de tarefas, designado por *Sprint Backlog*, com uma complexidade adequada à sua duração.

A abordagem incremental desta metodologia, permite analisar a complexidade do processo de desenvolvimento de forma progressiva, dividindo o problema em várias componentes e analisando as suas interacções. Essa análise é feita diariamente, reunindo todos os intervenientes no projecto, para que se possa analisar e controlar o progresso dos desenvolvimentos, e corrigir quaisquer desvios. Nas reuniões diárias, cada membro da equipa deverá conseguir responder às seguintes questões:

- O que fiz ontem?
- O que penso fazer até amanhã?
- Existe algo que me impeça de concretizar os meus objectivos?

No final de cada *Sprint*, são apresentados os resultados e é analisada a evolução dos desenvolvimentos, aplicando correcções ao plano, caso seja necessário. Depois dessa análise, definem-se as tarefas a executar no *Sprint* seguinte. Este processo repete-se nas iterações seguintes, até que se atinja um ponto onde exista um *deliverable* que reúna condições para ser entregue ao cliente final.

A essência desta metodologia está explicada acima e está também documentada em anexo. Dessa forma, poderá ser consultado o documento em anexo, que tem origem na documentação interna da WIT-Software.

4.2. Planeamento

O processo de desenvolvimento foi dividido em várias fases. Cada fase foi composta por uma ou mais iterações, podendo ter também mais do que uma *milestone* – altura em que as decisões críticas devem ser tomadas e os objectivos propostos devem ser atingidos^[1].

As fases do processo de desenvolvimento podem-se descrever da seguinte forma:

1. **Fase inicial:** fase em que é definido o âmbito do projecto. Para isso é necessário identificar todas as entidades externas (actores) com as quais o sistema irá interagir e definir-se, em alto nível, a natureza dessas interações (casos de uso).
2. **Fase de elaboração:** nesta fase analisa-se o domínio do problema, define-se a tecnologia a utilizar, estabelece-se a arquitectura base para o sistema, desenvolve-se o plano de projecto e mitigam-se os principais riscos que o poderão afectar. Todas as decisões arquitecturais deverão ainda considerar o sistema como um todo: o seu âmbito, requisitos funcionais e não-funcionais.
3. **Fase de construção:** durante esta fase, todas as características da aplicação e componentes remanescentes devem ser totalmente integrados no produto final e testadas exaustivamente.
4. **Fase de transição:** o objectivo desta fase é fazer a passagem do produto para a comunidade de utilizadores. Uma vez que o produto é entregue à comunidade, é comum surgirem solicitações para correcção de pequenos problemas ou implementação de novas funcionalidades. Contudo, o produto já se deve encontrar num estado de perfeita maturação.

Na fase de construção, foram consideradas várias *milestones*, que serviram de referência, para que se pudesse ter pronta uma versão preliminar do produto, para demonstração.

A figura seguinte, apresenta uma visão geral do planeamento deste projecto de estágio.

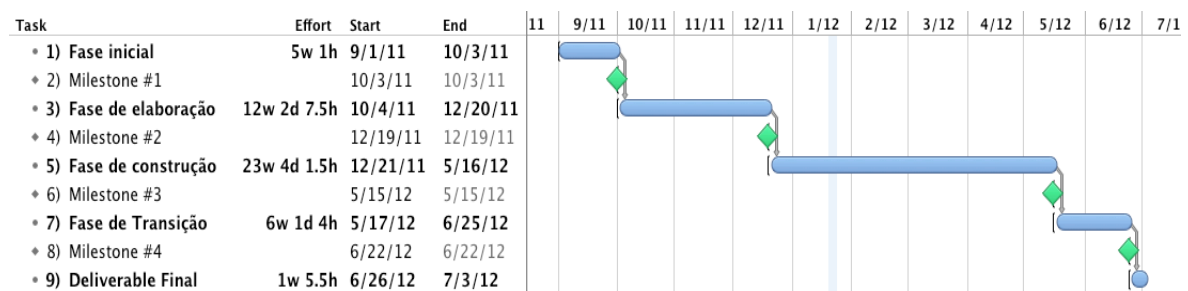


Figura 11 - Planeamento Geral do Projecto

4.3. Artefactos

De acordo com o que foi definido anteriormente, o projecto teve quatro fases, nas quais existiram uma ou mais *milestones*. Cada *milestone* englobou um conjunto de artefactos, da seguinte forma:

Milestone #1

- Documento de análise de requisitos funcionais e não-funcionais;
- Documento de apresentação do projecto;
- Avaliação inicial de risco;
- Plano de Projecto;
- Protótipo de baixa-fidelidade, sem integração com serviços.

Milestone #2

- Definição de requisitos complementares;
- Descrição da arquitectura e tecnologias a adoptar;
- Plano de desenvolvimento;
- Revisão dos riscos;
- Definir processo de desenvolvimento;
- Protótipo funcional.

Milestone #3

- Solução completamente integrada;
- Desenvolvimento de funcionalidades adicionais;
- Descrição do *deliverable* actual;
- Manual de utilizador preliminar.

Milestone #4

- Melhorias finais à solução;
- Manual de utilizador completo;
- Workshop de demonstração da ferramenta.

4.4. Plano de Iterações

O plano de iterações está intrinsecamente relacionado com a metodologia ágil adoptada. Desta forma, este plano é baseado no *Product Backlog*, sendo que em cada iteração é avaliada a evolução dos desenvolvimentos. São também seleccionadas as tarefas a executar nos *Sprints* seguintes, de acordo com as suas prioridades. A tabela seguinte contém as tarefas incluídas no *Product Backlog*.

| Categoria | Tarefa | Estimativa | Prioridade |
|-----------------|--|------------|------------|
| Análise | Análise de concorrentes | 3 dias | 1 |
| Análise | Determinar os requisitos da aplicação a desenvolver | 1 dia | 2 |
| Documentação | Criar uma apresentação inicial da ideia do produto | 2 dias | 3 |
| Design/Layout | Criar os mockups que exemplifiquem a experiência de utilização | 7 dias | 4 |
| Estudo e treino | Familiarização com os produtos e tecnologias envolvidas no desenvolvimento da aplicação. | 15 dias | 5 |

| | | | |
|-----------------|---|---------|----|
| Desenvolvimento | Ecrã de Login | 3 dias | 6 |
| Desenvolvimento | Gestor de Janelas | 2 dias | 7 |
| Desenvolvimento | Gestor de Temas | 2 dias | 8 |
| Desenvolvimento | Estrutura do ecrã principal com painéis colapsáveis | 3 dias | 9 |
| Desenvolvimento | Barra superior | 2 dias | 10 |
| Desenvolvimento | Estrutura lateral que conterá conteúdo relativo à sala de discussão | 2 dias | 11 |
| Desenvolvimento | Lista de Contactos | 8 dias | 12 |
| Desenvolvimento | Lista de Discussões | 7 dias | 13 |
| Desenvolvimento | Criar nova sala de discussão | 3 dias | 14 |
| Desenvolvimento | Convidar contacto para uma sala de discussão | 3 dias | 15 |
| Desenvolvimento | Área da Sala de Discussão | 4 dias | 16 |
| Desenvolvimento | Funcionalidade de discussão 1-para-1 e 1-para-N | 13 dias | 17 |
| Desenvolvimento | Lista de Ficheiros | 6 dias | 18 |
| Desenvolvimento | Notificações de novas mensagens | 5 dias | 19 |
| Desenvolvimento | Notificações de chamadas | 2 dias | 20 |
| Desenvolvimento | Chamadas de Voz | 7 dias | 21 |
| Desenvolvimento | Presença | 1 dia | 22 |
| Desenvolvimento | Configurações | 8 dias | 23 |
| Desenvolvimento | Opções da sala de discussão | 3 dias | 24 |
| Desenvolvimento | Partilha de ficheiros | 5 dias | 25 |
| Desenvolvimento | Pré-visualização de imagens partilhadas | 1 dia | 26 |
| Desenvolvimento | Convidar participantes externos | - | 27 |
| Desenvolvimento | Importar / Exportar conteúdos | - | 28 |

Tabela 21 - Product Backlog

Todas as tarefas relativas ao desenvolvimento da aplicação, já incluem a criação da interface e a sua integração com os serviços, caso isso se aplique. Existem tarefas que são subdivididas em tarefas mais pequenas, para otimizar as estimativas.

4.5. Gestão de Riscos

É essencial detectar precocemente, os potenciais riscos que poderão afectar o sucesso do projecto. Cada risco deverá ter um plano de contingência, que será sempre uma tentativa de

tentar corrigir os potenciais problemas. A tabela seguinte contém a lista dos potenciais riscos identificados.

| ID | Causa | Risco | Impacto | Exposição | Tempo de reacção |
|----|---|--|--------------|-----------|------------------|
| 1 | Plano demasiado ambicioso | Trabalhar acima do <i>budget</i> | Catastrófico | Elevada | Médio |
| 2 | Má gestão de tempo | Falhar <i>deadlines</i> ; Atraso no planeamento | Crítico | Elevada | Curto |
| 3 | Mau planeamento tem impacto sobre a qualidade final de software | Atraso no planeamento | Crítico | Elevada | Curto |
| 4 | Falta de conhecimentos técnicos | Atraso no planeamento; Trabalhar acima do <i>budget</i> | Crítico | Média | Curto |
| 5 | Dificuldades inesperadas na implementação de requisitos | Atraso no planeamento; Trabalhar acima do <i>budget</i> | Crítico | Média | Médio |
| 6 | Necessidade de alteração das características da aplicação | Falhar <i>deadlines</i> ; Atraso no planeamento | Catastrófico | Média | Curto |

Tabela 22 - Identificação de riscos

Para que se possa reduzir o impacto dos riscos, é necessário ter um plano de contingência. Dessa forma, a tabela seguinte apresenta o plano de contingência para cada caso.

| ID | Plano de Contingência |
|----|---|
| 1 | <ul style="list-style-type: none"> • Reavaliar o plano e dividir as maiores tarefas, em tarefas mais pequenas • Garantir que as estimativas são definidas de acordo com o conhecimento técnico do programador • Assumir estimativas realistas • Tentar negociar metas com o cliente final |
| 2 | <ul style="list-style-type: none"> • Eliminar distrações • Manter o <i>Sprint Backlog</i> sempre actualizado de acordo com a evolução • Recuperar tempo perdido com horas extra |
| 3 | <ul style="list-style-type: none"> • Renegociar os requisitos • Reestruturar os elementos afectados |
| 4 | <ul style="list-style-type: none"> • Treinar com tutoriais e com exemplos • Analisar o tempo gasto em cada tarefa e ajustar as estimativas seguintes • Consultar alguém que já tenha utilizado as tecnologias envolvidas |
| 5 | <ul style="list-style-type: none"> • Analisar o tempo gasto em cada tarefa e ajustar as estimativas seguintes |

| | |
|---|---|
| | <ul style="list-style-type: none"> • Consultar alguém que já tenha utilizado as tecnologias envolvidas |
| 6 | <ul style="list-style-type: none"> • Assumir um plano que seja ágil o suficiente para que hajam algumas alterações • Restruirar alguns elementos da aplicação para que tenham um comportamento mais genérico, por forma a ser mais fácil a adaptação a novos requisitos |

Tabela 23 - Planos de Contingência

Dos riscos previstos, pode-se afirmar que o plano foi demasiado ambicioso, com uma complexidade que se traduziu em tempo de desenvolvimento superior ao disponível. Para tentar corrigir este problema, foi necessário reavaliar as prioridades de algumas tarefas, nomeadamente as chamadas em conferência e a persistência dos dados nos serviços, uma vez que implicariam desenvolvimentos cujo tempo ultrapassaria os prazos disponíveis. Desta forma, será necessário continuar a desenvolver estas tarefas em trabalho futuro.

Houve também alguma dificuldade na implementação de requisitos, que levaram ao atraso do plano de trabalhos. Para mitigar este problema, foi necessário ajustar algumas estimativas, por forma a minimizar atrasos nas tarefas seguintes.

Capítulo 5

Implementação

Este capítulo tem como objectivo apresentar os aspectos mais relevantes da fase de implementação, nomeadamente informações relevantes sobre as bibliotecas externas utilizadas, mas também os detalhes de implementação da própria aplicação.

5.1. Bibliotecas

Antes de detalhar os desenvolvimentos, é importante ter uma visão geral das bibliotecas utilizadas. Por isso, seguidamente é apresentado uma breve explicação sobre as bibliotecas mais relevantes.

5.1.1. Communicator Library

A *Communicator Library* é a biblioteca responsável pela lógica da aplicação. Foi desenvolvida para outros produtos WCS, e tem uma importância vital, pois permite criar o nível de abstracção necessário na camada de apresentação, independentemente do Sistema Operativo. Nesta biblioteca, é tratada toda a lógica dos dados obtidos e enviados para os serviços. É composta por um conjunto de “managers”, com a responsabilidade de lidar com determinados serviços. Por exemplo, o “Call Manager” lida com os serviços de comunicações de audio e video; o “Instant Messaging Manager” lida com os eventos relacionados com o envio e recepção de mensagens.

Outras bibliotecas servem de suporte às funcionalidades implementadas na *Communicator Library*, nomeadamente a biblioteca *Client Library*, desenvolvida internamente e que contém todo o tipo de estruturas e classes utilizadas. Igualmente importante é a biblioteca PJSIP, desenvolvida por terceiros e que é utilizada para as comunicações através dos vários protocolos disponíveis: SIP, SDP, RTP, STUN, TURN e ICE. Existem outras bibliotecas, desenvolvidas por terceiros, que são utilizadas em conjunto com a *Communicator Library*.

A figura que se segue, representa a arquitectura de alto nível e as suas dependências:

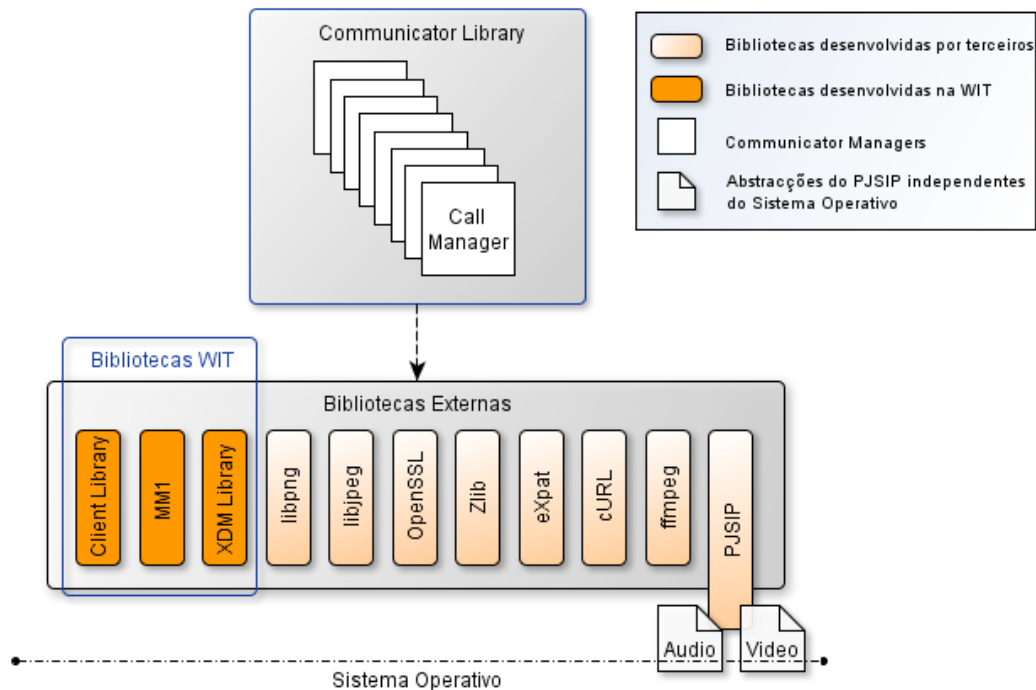


Figura 12 - Communicator Library e as suas dependências com outras bibliotecas

Apenas algumas das bibliotecas são desenvolvidas na WIT, pelo que a *Communicator Library* agrega um vasto conjunto de bibliotecas de terceiros.

5.1.2. PJSIP

O PJSIP é uma biblioteca que suporta um conjunto de funcionalidades definidas nos standards do SIP. Esta biblioteca contém um “User Agent” (PJSUA-LIB), que disponibiliza um conjunto de APIs que permitem aceder às suas funcionalidades. Entre as APIs mais importantes, destacam-se:

- **Basic API** – permite aceder a operações básicas inicialização de comunicações e configurações;
- **Gestão do transporte** – permite gerir os protocolos de transporte do SIP;
- **Gestão de contas** – Permite gerir configurações relacionadas com cada utilizador. Esta API suporta várias contas, e cada conta contém informações específicas do utilizador, tais como as credenciais e os dados de presença;
- **Gestão de chamadas** – Permite gerir as chamadas de voz e vídeo;
- **Contactos, presença e IM** – permite gerir as funcionalidades relacionadas com a presença e a comunicação instantânea (IM);
- **Manipulação de media** – Gestão dos dispositivos de captura e reprodução de som e vídeo.

As suas funcionalidades foram desenvolvidas para que exista o máximo de abstracção possível em relação a qualquer sistema operativo.

5.1.3. XDM Library

Esta biblioteca é responsável por interpretar os documentos XML, que contenham informação relacionada com os contactos obtidos do servidor, de acordo com o standard OMA XDM^[12]. Esta biblioteca é desenvolvida internamente.

5.2. Tecnologias

A tecnologia adoptada na camada de apresentação foi o C++, nomeadamente através da *framework* Qt, desenvolvida pela Nokia. Esta decisão tem como base, a facilidade que esta *framework* oferece na migração do software para outras plataformas. Outro motivo é o facto de já existirem diversos projectos na empresa que utilizam esta linguagem e/ou esta *framework*, e por isso, será mais fácil obter ajuda de outros colaboradores. É importante referir também a facilidade de integração com as bibliotecas já existentes, uma vez que são todas desenvolvidas em C++.

O Qt é uma *framework* “open-source”, cujo desenvolvimento está a cargo da Nokia. Esta *framework* assenta sobre a linguagem C++, disponibilizando um conjunto de componentes, também designados como *widgets*, que permitem a um programador implementar rapidamente a interface gráfica de uma aplicação. Uma das maiores vantagens do Qt, é a possibilidade de compilar as aplicações desenvolvidas para as principais plataformas, sem que seja necessário alterar o código fonte.

5.3. Desenvolvimentos

Segue-se uma explicação dos desenvolvimentos mais relevantes, efectuados em cada uma das camadas da aplicação.

5.3.1. Camada de apresentação

A camada de apresentação comunica directamente com a *Communicator Library*, efectuando chamadas directas às funções dos seus *Managers*, recebendo qualquer notificação de alterações de estado através de eventos provenientes desses mesmos *Managers*. A figura seguinte pretende exemplificar o seu funcionamento:

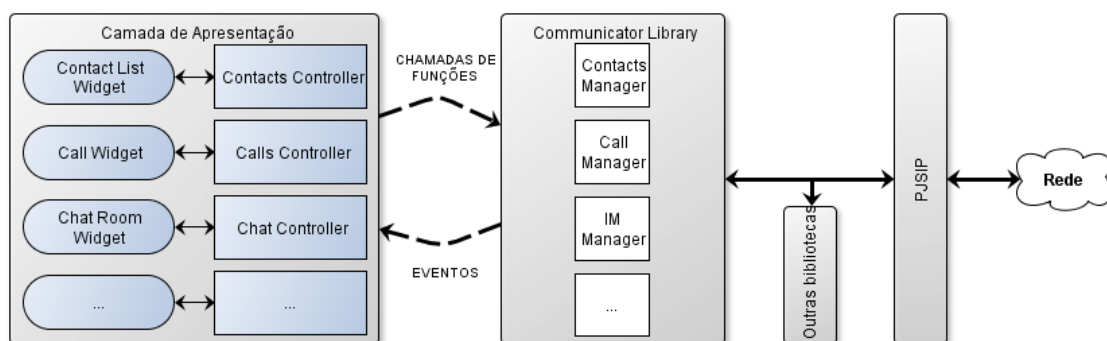


Figura 13 - Relação entre a camada de apresentação e a *Communicator Library*

Na figura anterior, está representada a comunicação entre as várias camadas da aplicação. A biblioteca PJSIP recebe comunicações dos serviços e executa os respectivos procedimentos

na *Communicator Library*, que por sua vez notifica a camada de apresentação, quando existem alterações de estado, através de eventos. A camada de apresentação invoca os métodos dos respectivos *Managers*, para obter a informação necessária ao preenchimento dos seus componentes.

Todo o interface foi desenvolvido de raiz tendo em conta os requisitos apresentados. Sofreu alterações desde a primeira proposta até à versão final. Essa evolução pode ser consultada nos anexos A, B e C.

Para facilitar o processo de desenvolvimento, cada componente visual, designado por *Widget*, foi sub-dividido noutros componentes menos complexos, sendo assim possível a criação de tarefas com complexidade reduzida. Os vários componentes (*widgets*) são desenvolvidos individualmente, sendo interligados com recurso ao mecanismo nativo de sinais e procedimentos (*signals and slots*) do Qt. Um sinal é emitido sempre que acontece um determinado evento. A cada sinal emitido, é possível associar um procedimento a ser executado, ou emitir outro sinal. Este mecanismo dá flexibilidade ao desenvolvimento, e permite que os componentes sejam desenvolvidos separadamente, sem criar dependências directas. No final, os componentes são interligados para que se obtenha o resultado planeado.

As listas de elementos, por exemplo as listas de contactos, foram desenvolvidas com recurso ao padrão *Model-View*. O padrão *Model-View* é uma derivação do padrão *Model-View-Controller* (MVC), que é utilizado no desenvolvimento de interfaces gráficas. O MVC é composto por três tipos de objectos, o modelo, vista e controlador. O modelo (*model*), é o objecto que representa a informação que é utilizada na aplicação. A vista (*view*), é a representação gráfica dessa informação. O controlador (*controller*), define a forma como o interface reage à interacção com o utilizador.

Neste caso, a vista e o controlador são agregados, e assim se forma a arquitectura *Model-View*. Desta forma, é possível manter a informação separada da forma como é apresentada aos utilizadores, simplificando também o processo de desenvolvimento. Com esta arquitectura é possível utilizar os mesmos modelos em vistas diferentes, sem a necessidade de alterar qualquer estrutura de dados. Para facilitar o controle sobre a interacção com o utilizador, é utilizado um *delegate*, que permite controlar ao detalhe a forma como os elementos visuais são desenhados e editados, consoante o estado dos mesmos.

A figura seguinte representa a forma como os objectos desta arquitectura se relacionam:

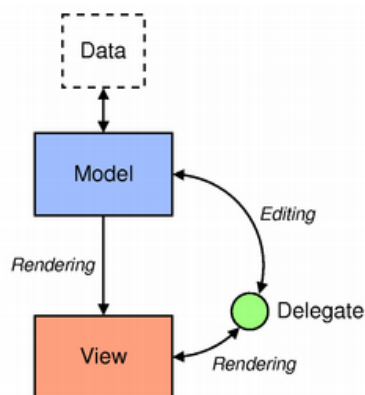


Figura 14 - Padrão Model-View [11]

O modelo obtém os dados de uma fonte de informação (Web Services, SQL, XML, etc.), oferecendo aos outros componentes da arquitectura, um interface de acesso aos dados. A

vista obtém os índices dos elementos guardados no modelo, que são as referências para a informação obtida. O *delegate* é responsável por desenhar cada um dos elementos na vista, de acordo com o seu estado. Toda a comunicação entre os objectos é feita com recurso ao sistema de sinais (eventos) nativo da *framework* Qt. Assim sendo, os três tipos de objectos comunicam da seguinte forma:

- Sinais emitidos pelo modelo, informam a vista sobre as alterações aos dados a apresentar;
- Sinais emitidos pela vista, informam das interações que o utilizador tem com os elementos visuais;
- Sinais emitidos pelo *delegate*, são utilizados ao editar um elemento, para informar o modelo e a vista acerca do estado do mesmo.

Para criar a abstracção necessária entre os dados obtidos dos serviços e a camada de apresentação, foram criados objectos que servem de ponte entre a camada de lógica e a camada de apresentação, fazendo uma tradução dos objectos obtidos, para objectos desta camada. A estes objectos deu-se o nome de *Controller*, que nada têm que ver com os controladores do padrão MVC.

O aspecto final da aplicação, pode ser consultado no Anexo C.

5.3.2. Camada de lógica

Esta camada é responsável por manter a consistência e coerência entre os dados existentes nos serviços e os dados disponibilizados visualmente. A *Communicator Library* é o ponto central desta camada, contendo toda a lógica de gestão dos dados.

Nesta camada foram feitos desenvolvimentos para dar suporte a alguns requisitos. Foi necessário implementar novos eventos para a gestão das salas de discussão, para que a camada de apresentação fosse notificada sempre que existissem alterações de estado. Foi também necessário criar e adaptar diversas classes, para suportarem algumas propriedades inerentes às novas funcionalidades.

Para a persistência local de alguma informação, nomeadamente das configurações da aplicação, é utilizada uma base de dados SQLite. A gestão destes dados é feita com recurso a desenvolvimentos já efectuados nesta camada, não tendo sido a estrutura dos dados do âmbito deste projecto.

Capítulo 6

Considerações Finais e Trabalho Futuro

A ferramenta WIT Corporate Collaboration surgiu com o objectivo de tentar colmatar algumas lacunas existentes na comunicação e organização de discussões, dentro das organizações.

Apesar deste ser um projecto muito ambicioso, creio ter conseguido concluir grande parte do trabalho a que me propus desenvolver. No âmbito tecnológico, foi um projecto abrangente, pois foram abordadas diversas tecnologias de comunicação. Embora não tenha abordado directamente algumas tecnologias, foi necessário adquirir algum conhecimento para compreender o fluxo dos processos.

A nível pessoal, a experiência foi muito enriquecedora. Destaco desta experiência, a adaptação ao trabalho em equipa, onde todos os dias somos postos à prova, e onde é importante não cedermos ao facilitismo. Este facto, permitiu-me acima de tudo, compreender a importância que tem uma boa gestão de um projecto.

Como trabalho futuro destaca-se a necessidade de implementar nos serviços, a gestão de salas de discussão e a persistência dos seus conteúdos. Isto é importante pois facilita imenso o processo de gestão, evitando assim quaisquer inconsistências entre diferentes postos de trabalho, uma vez que desta forma os dados são sincronizados com os serviços e apresentados de igual forma em diferentes sessões.

É igualmente importante a implementação das chamadas em conferência, tanto de voz como de video, sendo uma funcionalidade de bastante interesse para este tipo de ferramenta.

Para concluir, deve ser ponderada a migração desta plataforma para o ambiente Web, uma vez que neste momento já existem tecnologias que permitem a comunicação através do protocolo SIP via *browser*, sem o uso adicional de qualquer *plugin*. Esta pode ser considerada a questão mais importante, uma vez que permitirá a utilização da ferramenta sem a necessidade de instalação de qualquer software no computador do cliente, o que permite eliminar à partida, um entrave a quem utilize a ferramenta em diferentes postos de trabalho.

Referências

- [1] Boehm, B. W. (1996, Julho 4). Anchoring the Software Process. *IEEE Software*, 13(4) pp. 77-82.
- [2] Campfire, *Wikipedia*. [Online] [http://en.wikipedia.org/wiki/Campfire_\(software\)](http://en.wikipedia.org/wiki/Campfire_(software)),
acedido em 12 de Julho de 2012
- [3] Campfire, Página oficial [Online] <http://campfirenow.com/>,
acedido em 12 de Julho de 2012
- [4] HipChat, *Wikipedia*. [Online] <http://en.wikipedia.org/wiki/HipChat>,
acedido em 12 de Julho de 2012
- [5] HipChat, Página oficial [Online] <http://www.hipchat.com/>,
acedido em 12 de Julho de 2012
- [6] EchoWaves [Online] <http://www.echowaves.com>,
acedido em 24 de Janeiro de 2012
- [7] Yammer, *Wikipedia*. [Online] <http://en.wikipedia.org/wiki/Yammer>,
acedido em 12 de Julho de 2012
- [8] Yammer, Página oficial [Online] <http://www.yammer.com>,
acedido em 12 de Julho de 2012
- [9] Functional requirement, *Wikipedia*. [Online]
http://en.wikipedia.org/wiki/Functional_requirement,
acedido em 12 de Julho de 2012
- [10] Non-functional requirement, *Wikipedia*. [Online] http://en.wikipedia.org/wiki/Non-functional_requirement,
acedido em 12 de Julho de 2012
- [11] Model/View Programming, Qt-Project [Online] <http://qt-project.org/doc/qt-4.8/model-view-programming.html>,
acedido em 12 de Julho de 2012
- [12] OMA XML Document Management, Opem Mobile Alliance [Online]
http://www.openmobilealliance.org/Technical/release_program/xdm_v2_0.aspx,
acedido a 12 de Julho de 2012