

Mestrado em Engenharia Informática
Estágio
Relatório Final

Sistema de Gestão de Reuniões

André Filipe Branquinho Maximino
andref@student.dei.uc.pt

Orientador:

Prof. Raul Barbosa

Eng. Carlos Lopes

Data: 1 de Julho de 2016



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

Resumo

O presente trabalho descreve o desenvolvimento de uma aplicação web que permite a gestão de reuniões e o acompanhamento de projetos ao nível da direção. O cliente para o qual a aplicação está a ser desenvolvida é o Laboratório de Informática e Sistemas(LIS) do Instituto Pedro Nunes(IPN), e será utilizada pela direção e gestores de projetos do IPN-LIS para que possam gerir e tornar as suas reuniões mais eficientes e realizar o acompanhamento dos diversos projetos desenvolvidos.

A decisão de desenvolver uma aplicação de raiz passou primeiro pela análise do estado da arte de ferramentas de gestão de reuniões e acompanhamento de projetos existentes no mercado. De seguida e como as aplicações encontradas não davam resposta às especificações do cliente, foram levantados um conjunto de requisitos, que cumprissem as funções desejadas. Com os requisitos identificados, procedeu-se à elaboração de mockups para a aplicação, seguido de testes de usabilidade. Para concluir, desenvolveu-se a correspondente aplicação que foi disponibilizada ao cliente ao longo do segundo semestre, em várias releases funcionais.

A principal contribuição deste estágio é assim, uma nova ferramenta que facilita a gestão de reuniões e o acompanhamento de projetos.

Palavras-Chave

“APLICAÇÃO WEB”, “MULTI-TENANCY”, “PLAY FRAMEWORK”, “SAAS”;

Índice

Capítulo 1 Introdução e motivação.....	1
1.1 Objetivos do estágio	2
1.2 Documentação Produzida.....	2
1.3 Estrutura do Relatório.....	3
Capítulo 2 Planeamento e metodologia	5
2.1 Metodologia Utilizada.....	5
2.2 Calendarização.....	6
Capítulo 3 Estado da arte	15
3.1 Boas práticas em reuniões	15
3.1.1 Antes da reunião.....	15
3.1.2 Durante a Reunião.....	16
3.1.3 Após a reunião	17
3.2 Análise de Aplicações de gestão de reuniões existentes no mercado	17
3.3 Conclusão do capítulo	19
Capítulo 4 Levantamento de requisitos.....	20
4.1 Principais processos de negócio da aplicação.....	20
4.2 Perfis de utilizadores.....	21
4.3 Requisitos funcionais.....	22
4.3.1 RF1 – Módulo de Reuniões	22
4.3.2 RF2 – Módulo de Pontos de Ação	24
4.3.3 RF3 – Módulo de Grupos.....	25
4.3.4 RF4 – Módulo de Projetos.....	26
4.3.5 RF5 – Módulo de utilizadores	27
4.3.6 RF6 – Módulo de interface com o utilizador.....	27
4.4 Requisitos não-funcionais	28
4.5 Restrições de implementação	28
4.5.1 Multi-Tenancy	29
4.5.2 Utilização em dispositivos móveis.....	31
4.5.3 Adicionar informação em tempo real nos ecrãs de todos os utilizadores	31
4.6 Critérios para a escolha da tecnologia a utilizar no desenvolvimento	31
4.7 Conclusão do capítulo	32
Capítulo 5 Arquitetura da aplicação	33
5.1 Nível 1 – Diagrama de contexto.....	33
5.2 Nível 2 - Diagrama de <i>container</i>	34

5.3 Nível 3 - Diagrama de componentes	35
5.4 Arquitetura Bases de Dados Multi-Tenancy	36
5.5 Modelos da aplicação	37
5.6 Ficheiros de configuração interna	41
5.7 Base de dados	41
5.8 Conclusão do capítulo	44
Capítulo 6 Desenvolvimento da aplicação	45
6.1 Ambientes de desenvolvimento	45
6.1.1 Ambiente de desenvolvimento.....	45
6.1.2 Ambiente de Staging.....	45
6.1.3 Ambiente de Produção.....	46
6.2 Evoluções da base de dados	47
6.3 Web Sockets	48
6.4 Conclusão do capítulo	48
Capítulo 7 Verificação e validação.....	50
7.1 Verificação.....	50
7.2 Validação.....	51
7.3 Testes de usabilidade.....	52
7.3.1 Configuração dos testes.....	52
7.3.2 Resultado dos testes de usabilidade	54
7.3.3 Análise aos resultados dos testes de usabilidade.....	55
7.4 Conclusão do capítulo	56
Capítulo 8 Ecrãs da aplicação	58
8.1 Construção de Mockups	58
8.2 Ecrãs da Aplicação.....	62
8.3 Conclusão do capítulo	71
Capítulo 9 Conclusões	72
9.1 Propostas de melhorias na ferramenta.....	72
9.2 Conclusões do estágio	72
Bibliografia	74

Índice de Figuras

Figura 1 - Calendarização 1º Semestre.....	7
Figura 2 - Planeamento das atividades do segundo semestre	9
Figura 3 - Planeamento refeito das atividades do segundo semestre	11
Figura 4 - Planeamento de áreas a serem implementadas	12
Figura 5 - Detalhe requisitos a desenvolver em cada área	14
Figura 6 - Exemplo de Agenda de Reunião [4]	16
Figura 7 - Processo de gestão de reuniões	20
Figura 8 - Arquitetura dos níveis de maturidade de SaaS [16]	30
Figura 9 - Diagrama de nível 1 do sistema IPNMeetings	33
Figura 10 - Diagrama de nível 2 do sistema IPNMeetings	34
Figura 11 - Diagrama de nível 3 do sistema IPNMeetings	35
Figura 12 - Esquema das diferentes bases de dados	36
Figura 13 - Configuração dos servidores Ebean	37
Figura 14 - Diagrama de Classes	40
Figura 15 - Diagrama ER da aplicação	43
Figura 16 - Ícones da aplicação	55
Figura 17 - Ecrã durante reunião com design aplicado	56
Figura 18 - Painel principal primeiro ciclo de desenvolvimento	59
Figura 19 - Painel principal segundo ciclo de desenvolvimento.....	60
Figura 20 - Painel principal terceiro ciclo de desenvolvimento.....	61
Figura 21 - Ecrã "Autenticação"	62
Figura 22 – Página Principal Aplicação	63
Figura 23 - Ecrã "Adicionar Reunião"	64
Figura 24 - Ecrã "Informação de uma Reunião"	65
Figura 25 - Ecrã "Confirmar Reunião"	66
Figura 26 - Ecrã "Durante Reunião"	67

Índice de Tabelas

Tabela 1 - Documentação Produzida	3
Tabela 2 - Comparação entre ferramentas analisadas.....	18
Tabela 3 - Requisitos módulo de reuniões	24
Tabela 4 - Requisitos módulo de pontos de ação.....	25
Tabela 5 - Requisitos módulo de grupos	25
Tabela 6 - Requisitos módulo de projetos.....	27
Tabela 7 - Requisitos módulo de utilizadores	27
Tabela 8 - Requisitos módulo de interface.....	28
Tabela 9 - Requisitos não funcionais	28
Tabela 10 - Restrições de implementação	29
Tabela 11 - Entidades existentes na aplicação	38
Tabela 12 - Entidades auxiliares existentes na aplicação	39
Tabela 13 - Ficheiros de configuração	41
Tabela 14 - Resultados testes de usabilidade(segundos).....	54

Glossário

IPNMeetings	Nome dado à aplicação em desenvolvimento
IPN	Instituto Pedro Nunes
LIS	Laboratório de Informática e Sistemas
Multi-Tenancy	Arquitetura de software em que uma única instância serve vários clientes.
Software as a Service	Software implementado como um serviço, hospedado e acedido pela internet.
Grupo	Engloba vários itens como reuniões, projetos, pontos de agenda ou pontos de ação. Possibilita o relacionamento entre projetos e reuniões, em que ambos pertencem a um grupo.
Ponto de Agenda	Ponto a ser discutido numa reunião
Ponto de Ação	Tarefa que um utilizador tem para realizar até uma data limite

Acrónimos

BD	Base de Dados
CSS	Cascading Style Sheets
ER	Entidade Relacionamento
FCT	Faculdade de Ciências e Tecnologias
GUI	Graphical User Interface
HTTP	Hipertext Transfer Protocol
HTTPS	Hipertext Transfer Protocol Secure
JS	Javascript
MVC	Modelo-Vista-Controlador
UC	Universidade de Coimbra

Capítulo 1

Introdução e motivação

O principal objetivo deste relatório é apresentar o trabalho realizado no âmbito do estágio “Sistema de Gestão de Reuniões”, do ano letivo de 2015/2016, para a disciplina “Estágio/Dissertação” do Mestrado em Engenharia Informática(MEI) da Faculdade de Ciências e Tecnologias(FCT) da Universidade de Coimbra(UC), no ramo de Sistemas de Informação.

O presente estágio foi acompanhado e orientado pelo Doutor Raul Barbosa (FCTUC) e pelo Eng.º Carlos Lopes(IPN-LIS), foi desenvolvido pelo aluno, tendo contado com a colaboração do Nuno Eufrásio no design da aplicação.

O Laboratório de Informática e Sistemas do Instituto Pedro Nunes(IPN-LIS) tem um elevado número de projetos a decorrer em simultâneo. Cada um destes projetos tem um gestor próprio, que tem a seu cargo inúmeros outros. Assim sendo, cada gestor tem várias reuniões que necessita de organizar e planejar. Para além destas reuniões, todas as semanas, existe ainda outra de direção entre os gestores de projetos e a direção do IPN-LIS.

A preparação da agenda destas reuniões de direção segue um procedimento específico:

- O moderador da reunião marca uma data e hora e notifica por email os outros participantes;
- Cada participante envia os pontos de agenda que pretende discutir, por email, para o moderador;
- O moderador reúne todos os pontos de agenda recebidos num documento de texto;
- Escolhe quais se adequam à reunião dessa semana, tendo em conta a sua importância e a duração prevista para a reunião;
- Envia, por email, o documento para os outros participantes;

Para além dos pontos de agenda que existam para essa reunião, a direção do IPN-LIS é posta a par de informações e alterações que tenham ocorrido em cada projeto. Estas informações não são relativas à gestão do projeto ao nível do desenvolvimento e planeamento, mas sim relativas à direção do laboratório. Podem ser, entre outras:

- A alteração do custo de um projeto;
- O não pagamento de alguma fatura por parte de um cliente;
- A oportunidade de candidatar um determinado projeto a um programa de financiamento;

Desta reunião podem ainda resultar pontos de ação, ou seja, tarefas que os participantes têm de realizar e que se encontram relacionadas com algum projeto ou ponto de agenda ou ainda pontos de follow-up, que representam pontos de agenda que não foram completamente debatidos nessa reunião e que necessitam de mais alguma informação para ficarem completos. Estes pontos de follow-up transitam para as reuniões seguintes, até que sejam possíveis de concluir.

Atualmente, a ferramenta utilizada nas reuniões para anotar as notas de cada ponto de agenda, os pontos de follow-up, os pontos de ação e as alterações em cada projeto é um processador de texto. Este método não é eficiente nem escalável para a elevada quantidade de projetos que o IPN-LIS tem atualmente ativos. Para além disso é mais propício a introdução de informação no projeto errado ou o apagar de algum projeto.

Surge, assim, a necessidade de encontrar uma aplicação web centralizada, que possa ser executada por um elevado número de dispositivos, e ser utilizada pelos gestores de projetos e direção do IPN-LIS. A aplicação deve permitir a preparação e gestão das reuniões, bem como realizar o acompanhamento dos projetos ao nível da direção. Esta aplicação procura substituir os métodos descritos acima de preparação de reuniões e acompanhamento de cada projeto.

1.1 Objetivos do estágio

O foco principal do estágio é o desenvolvimento de uma ferramenta de gestão de reuniões e acompanhamento de projetos. Para o seu desenvolvimento, o estagiário deve realizar as seguintes tarefas:

- Definir os objetivos funcionais e não funcionais, com base no estado da arte e em reuniões com o cliente e com consultores experientes do Instituto Pedro Nunes(IPN);
- Definir as tecnologias a utilizar com base nos requisitos e no aconselhamento de engenheiros do IPN;
- Definir a arquitetura do sistema a desenvolver;
- Desenvolver a ferramenta de forma a ser instalada nos servidores do IPN e ser utilizada através da internet (Software as a Service).
- Realizar testes ao sistema;
- Instalar o sistema para testes piloto num ambiente real;
- Instalar o sistema num servidor de produção, para ser utilizado pelos utilizadores.

1.2 Documentação Produzida

Ao longo do projeto, foram sendo registadas as principais etapas do projeto. Os seguintes documentos descrevem vários aspetos técnicos e não técnicos que auxiliaram o desenvolvimento da aplicação.

Na tabela 1, são listados e descritos os documentos referenciados no seguinte relatório.

Referência/Nome	Descrição
Ref_1 – Levantamento de requisitos	Documento onde são especificados e descritos todos os requisitos.
Ref_2 – Resultados testes de usabilidade	Documento onde são expostos os resultados dos testes de usabilidade efetuados aos mockups.
Ref_3 – Atraso no planeamento	Documento que explica o atraso ocorrido, o que vai ser alterado e como se vai mitigar esse atraso.

Ref_4 – Análise a aplicações	Documento onde é apresentada a análise individual de cada aplicação testada.
------------------------------	--

Tabela 1 - Documentação Produzida

1.3 Estrutura do Relatório

Após a presente introdução ao estágio, o restante relatório encontra-se organizado nos seguintes capítulos:

- Planeamento e metodologia
- Estado da arte
- Levantamento de requisitos
- Mockups e testes de usabilidade
- Arquitetura da aplicação
- Desenvolvimento e aplicação final
- Conclusão

Capítulo 2

Planeamento e metodologia

O plano de trabalhos definido inicialmente pelo IPN-LIS consistia em:

- 1º Semestre:
 - Elaborar o estado da arte;
 - Realizar levantamento de requisitos para a aplicação;
 - Elaborar o desenho de protótipos do design da aplicação;
 - Elaborar o modelo de dados;
 - Elaborar a arquitetura da aplicação;
- 2º Semestre:
 - Desenvolver da aplicação;
 - Realizar testes;

2.1 Metodologia Utilizada

Um dos requisitos iniciais do cliente era ter acesso a várias releases funcionais ao longo do processo de desenvolvimento. Com isto em mente, procurou-se configurar um processo de desenvolvimento de software apropriado para o efeito. A abordagem a utilizar em cada projeto deve ser baseada nas suas características, não existindo nenhuma que possa ser aplicada a todos os projetos de software.

Tendo em conta o planeamento base já existente para a proposta de estágio e as condições identificadas pelo cliente (ter acesso a várias releases funcionais), usaram-se técnicas do Waterfall[1] e do Staged Delivery[2]. Ambas apresentam fases de planeamento, levantamento de requisitos e design da aplicação numa fase inicial do projeto, mas diferem na forma como realizam a fase da implementação e teste.

A metodologia Waterfall foi utilizada desde o início do projeto até à fase de desenvolvimento. A sua escolha para as primeiras fases do projeto de software, que englobam o levantamento e análise de requisitos e o design do sistema, é justificada pela existência de requisitos estáveis e identificados no início do projeto[1].

Nesta primeira fase, foram realizadas várias etapas:

- Análise do Problema
- Levantamento de Requisitos
 - Recolha de Requisitos
 - Documentação de Requisitos
- Design do Sistema
 - Análise detalhada de requisitos para escolha de tecnologia a utilizar;
 - Design da aplicação e testes de usabilidade;
 - Arquitetura da aplicação

Daqui resultam elementos que suportam toda a implementação, como o documento de requisitos (documento Ref_1), os mockups da aplicação e os resultados dos testes de usabilidade (documento Ref_2), o modelo de dados e a identificação dos sistemas externos com que a aplicação terá de comunicar. Pretendia-se também que todos os requisitos fossem

analisados extensivamente para serem todos assegurados, tanto nos mockups, como no modelo de dados desenvolvidos.

Após esta fase inicial, a abordagem utilizada foi a Staged Delivery. Foi escolhida pois permite a entrega de partes funcionais durante o desenvolvimento. As suas principais vantagens são[2]:

- Possibilidade de entregar pequenas partes funcionais da aplicação ao longo do desenvolvimento;
- Compatibilidade com a metodologia Waterfall, permitindo maximizar a utilização dos recursos produzidos com essa metodologia;
- Diferentes fases de testes para cada release, que possibilitam a correção prévia de erros, e retiram a sobrecarga no final da implementação;

Este processo de desenvolvimento permite satisfazer as necessidades do cliente em ter entregas funcionais ao longo da fase de desenvolvimento, bem como seguir o planeamento pensado para o projeto de estágio.

A utilização da metodologia “Staged Delivery” abrangeu a fase de desenvolvimento e testes da aplicação.

2.2 Calendarização

Após a escolha das metodologias foi possível calendarizar o estágio de acordo com estas, numa primeira fase até à defesa intermédia, e de seguida para o segundo semestre do estágio. A figura 1, apresenta o diagrama de Gantt do planeamento inicial do estágio para o primeiro semestre.

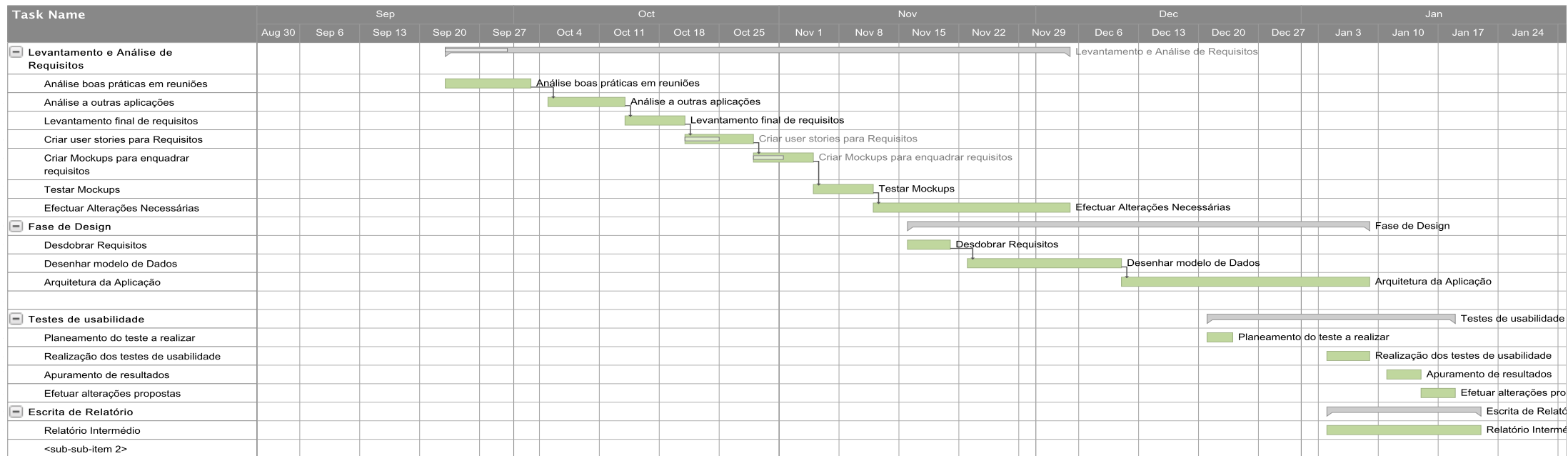


Figura 1 - Calendarização 1º Semestre

O planeamento foi seguido, verificando-se atrasos na parte final do semestre (Documento Ref_3) devido à extensão da realização de mockups. Com estes atrasos em vista foi ajustado o planeamento, adiando a realização dos testes de usabilidade para o início do segundo semestre.

Para este segundo semestre foi necessário ter em conta a metodologia escolhida, ou seja, a entrega de partes funcionais ao longo do desenvolvimento. Na figura 2, pode ser visto o planeamento previsto para o semestre.

Na figura, podemos distinguir 3 entregas distintas, finalizando o semestre com uma fase de testes funcionais à aplicação.

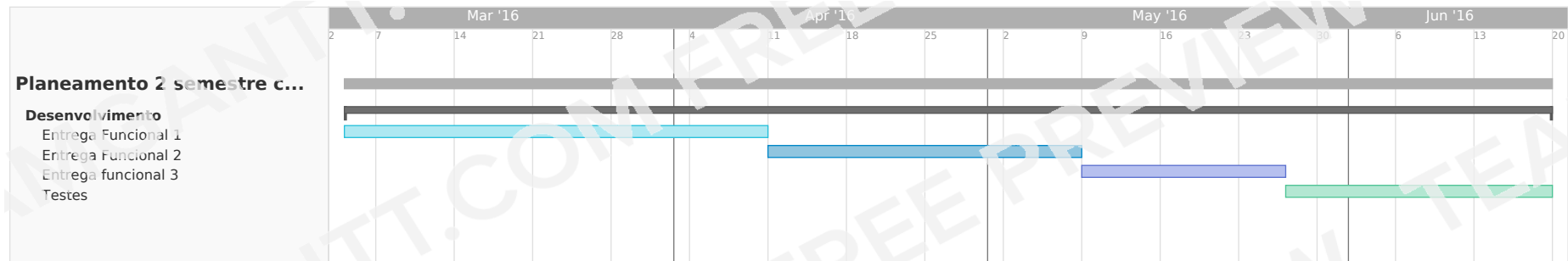


Figura 2 - Planeamento das atividades do segundo semestre

Este planeamento sofreu alguns atrasos aquando da primeira entrega funcional planeada para o início de Abril. Este deveu-se à correção de erros identificados por utilizadores que testaram a aplicação. Tinham o objetivo de validar os requisitos desenvolvidos e identificar erros ou alterações necessárias ao nível do design ou da funcionalidade de alguns forms para submissão de dados. Procedeu-se à correção dos erros identificados uma vez que iam ser cometidos noutras fases do desenvolvimento, o que atrasaria ainda mais as entregas futuras.

Foi assumido o atraso nessa entrega e consequentemente nas próximas uma vez que o risco de o projeto atrasar na totalidade não era muito elevado. Isto porque o que estava a ser alterado era a fase de testes planeada para o fim do desenvolvimento, que foi assim dividida pelas 3 entregas funcionais. Após este atraso, foi criado outro diagrama de Gantt, visível na figura 3, onde a calendarização do semestre fosse a mais correta.

Para prever os requisitos que iriam ser desenvolvidos para cada entrega foi feito um mapeamento das áreas de requisitos a desenvolver em cada entrega. Este planeamento dos requisitos foi realizado no início do semestre e serviu como uma estrutura para o trabalho do dia-a-dia, de forma a que o estagiário soubesse que funcionalidades devia implementar a seguir.

O diagrama referido pode ser visto na figura 4 e na figura 5. Na figura 4, são apresentadas as áreas de requisitos e a ordem pela qual foram implementadas. Na figura 5 é detalhada cada uma dessas áreas e dividida em conjuntos de requisitos.

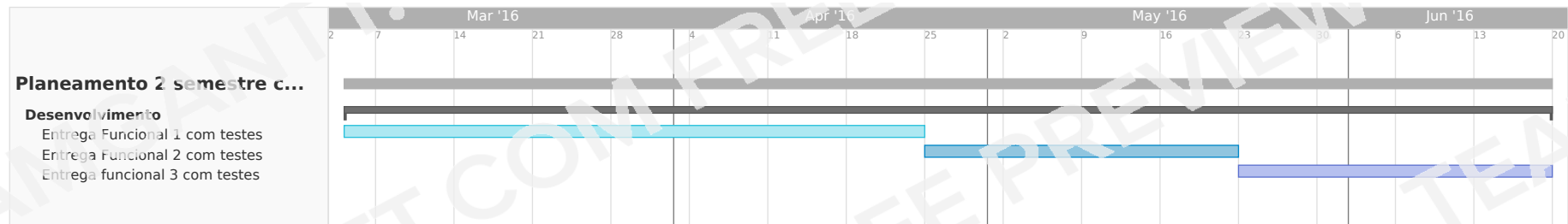


Figura 3 - Planeamento referido das atividades do segundo semestre

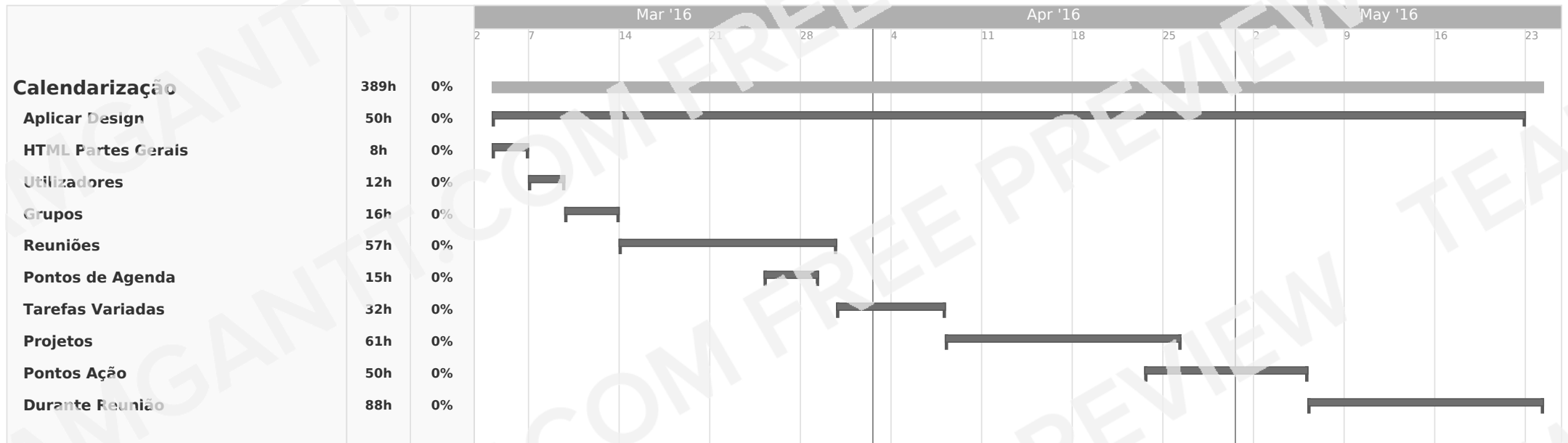
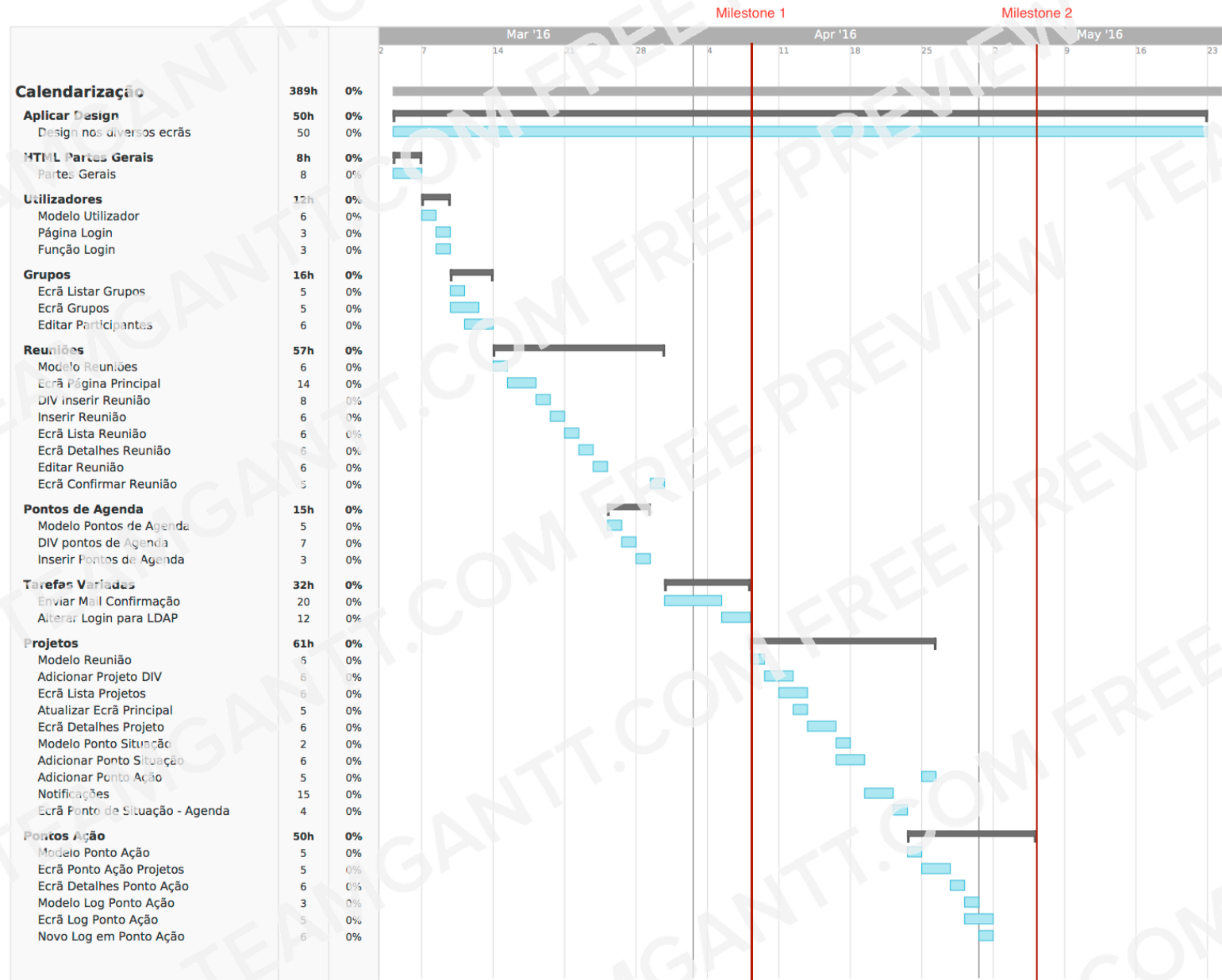


Figura 4 - Planeamento de áreas a serem implementadas



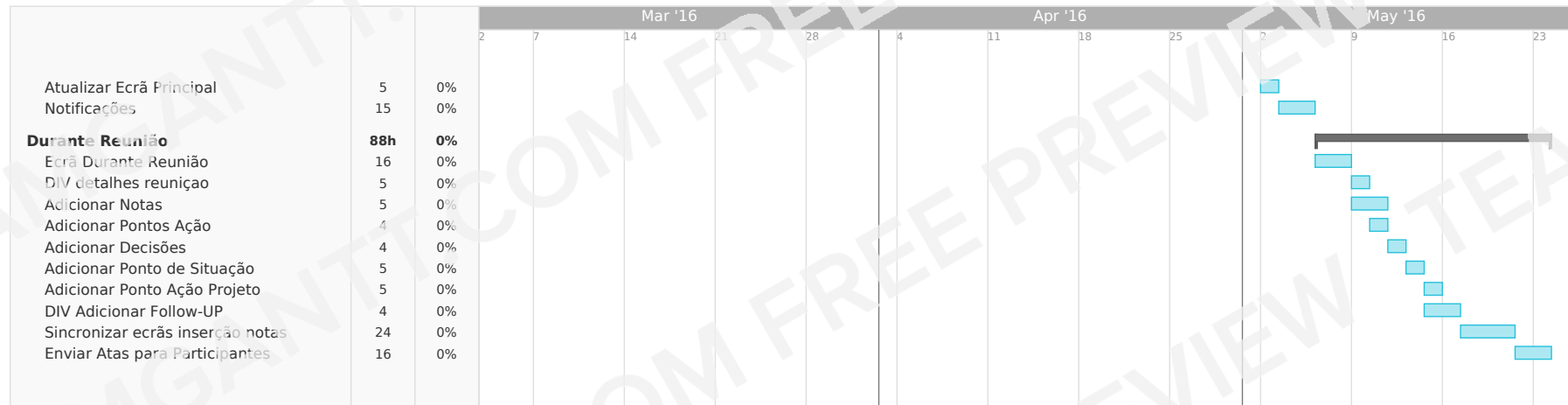


Figura 5 - Detalhe requisitos a desenvolver em cada área

Capítulo 3

Estado da arte

Neste capítulo, serão abordadas algumas boas práticas existentes na preparação e gestão de reuniões, analisando processos definidos para este fim. Por último, serão apresentadas e descritas as ferramentas de gestão de reuniões analisadas, com a finalidade de perceber se existia alguma que correspondesse ao que o cliente pretendia e identificar um conjunto de requisitos importantes neste tipo de aplicações.

3.1 Boas práticas em reuniões

As reuniões alteraram o seu formato ao longo do tempo pois a realidade das empresas também foi sendo alterada. Isto levou a que o processo de preparação de reuniões evoluísse também. A preparação da reunião, em específico reuniões de direção, começa com muita antecedência e inclui 3 grandes fases [3]:

- Iniciação;
- Condução;
- Conclusão;

3.1.1 Antes da reunião

A fase de iniciação de uma reunião contém 3 pontos chave que estabelecem a estrutura da reunião. Estas são a localização da reunião, o estabelecimento de uma agenda e a direção da reunião, entregue ao moderador.

Quanto à localização da reunião, agrupar os participantes numa localização central dá-lhes a oportunidade de se juntarem num espaço comum, contrariando a cada vez maior separação de setores dentro das empresas. Esta relocação física dos participantes para fora do seu meio tem também uma implicação psicológica na eficiência da reunião por parte dos participantes. Estes ficam mais focados nos objetivos da reunião pois são separados dos seus interesses e trabalhos pendentes. Como normalmente estas reuniões são feitas nas sedes das empresas ou em edifícios mais utilizados pela direção, esta localização da reunião confere também um sentido de autoridade à direção.

Após decidida a localização e convocados os participantes, é necessário começar a preparação da agenda da reunião. Estas são preparadas por alguém do topo da organização e enviadas para os outros participantes com alguns dias de antecedência. Apesar de serem normalmente as pessoas do topo da organização a organizar e a enviar a agenda, não podem controlá-la completamente pois esta deve ser composta por pontos de agenda submetidos por todos os participantes. Apesar disso, é deles a responsabilidade de organizar a ordem dos pontos de agenda.

Uma agenda completa é o primeiro sinal de que a reunião irá ser produtiva pois fornece uma lista dos tópicos a discutir na reunião. Como podemos ver na figura 6 [4], uma agenda bem estruturada necessita de ter uma estrutura bem definida.

A SAMPLE MEETING AGENDA		
TOPIC	PREPARATION	PROPOSED PROCESS
1 What changes, if any, should we make to the agenda? TIME: 2 minutes PURPOSE: Decision LEADER: Mike	None	<ul style="list-style-type: none"> Mike polls team.
2 What deltas from the previous meeting will we focus on this meeting? TIME: 3 minutes PURPOSE: Decision LEADER: Anne	Review applicable deltas from previous meeting notes.	<ul style="list-style-type: none"> Anne reviews areas of improvement that we agreed to focus on during this meeting.
3 How do we best manage the fluctuating internal demand for our services? TIME: 50 minutes PURPOSE: Decision LEADER: Peg	Identify relevant information, criteria, and assumptions that you believe should guide our decision.	<ul style="list-style-type: none"> Statement of the problem. TIME: 5 minutes Team identifies and agrees on relevant information to consider. TIME: 10 minutes Identify and agree on criteria for acceptable solutions. TIME: 10 minutes Identify and agree on assumptions. TIME: 10 minutes Craft solutions that meet the above constraints. TIME: 15 minutes
4 Which firm should we select for the adjacent markets acquisition study? TIME: 15 minutes PURPOSE: Decision LEADER: Martin	Read the attached memo recommending three firms. Be prepared to ask questions and share your initial preference and your reasoning.	<ul style="list-style-type: none"> Questions and additional information regarding the recommendations of the three firms. Decision to select one firm.
5 What developmental assignments are available during the next FY for high-potentials? TIME: 15 minutes PURPOSE: Decision LEADER: Noah	Review the attached memo identifying the current high-potential managers and the areas in which we are seeking developmental assignments for them. If feasible, be prepared to offer an appropriate developmental assignment.	<ul style="list-style-type: none"> Identify the available developmental assignments. Match the assignments to the pool of high-potential managers. Agree on next steps for any high-potential managers who have not been given a developmental assignment.
6 PLUS/DELTA: What did we do well for this meeting? What should we do differently for the next meeting? TIME: 5 minutes PURPOSE: Decision LEADER: Carrie	None	<ul style="list-style-type: none"> Members identify pluses and deltas. Team agrees on deltas to work on for next meeting.

Figura 6 - Exemplo de Agenda de Reunião [4]

A duração da reunião é também previamente pensada e a soma dos tempos de cada ponto de agenda não deve ultrapassar o tempo limite[5]. Esta gestão de tempo deve ser bastante cuidada. Se uma reunião demorar menos que o estimado significa que os intervenientes poderiam ter empregue esse tempo noutras atividades. Por outro lado, se uma reunião se arrastar por mais que o previsto, mostra algum desrespeito pela preciosidade do tempo das pessoas e pelos seus próximos compromissos agendados[6]. Para além disso, a partir de uma certa duração os participantes perdem a concentração no que está a ser debatido, tornando a reunião ineficaz.

3.1.2 Durante a Reunião

Para que as pessoas não percam o foco durante uma reunião é necessário que exista alguém nesta que faça com que a estrutura seja seguida e que controle o seu fluxo. Esta pessoa é denominada de moderador. Normalmente é alguém de destaque na organização, para que assim possa também induzir algum respeito.

Existem diversas técnicas que podem ser utilizadas aquando da moderação de uma reunião, das quais são destacadas 3 mais importantes[3]. A primeira é a técnica de discurso livre, em que o moderador encoraja todos os participantes a intervir na reunião com comentários e respostas, mesmo que não sejam estruturadas. A segunda técnica intitula-se de discurso restringido onde o moderador aparenta libertar-se do seu papel de autoridade permitindo que os participantes discutam os diversos pontos de agenda, embora controlando os parâmetros de discussão para que o tema não se disperse. Na última técnica, neste caso de discussão restringida, o moderador dá a vez a cada participante para realizar os comentários sobre cada ponto de agenda. Com este método, os participantes não conseguem ter

discussões organizadas uns com os outros, pois o moderador não permite interação direta entre participantes.

3.1.3 Após a reunião

Para a finalização de cada ponto de agenda é necessário perceber se este foi resolvido e discutido na sua totalidade, ou, pelo contrário, se não foi totalmente discutido, sendo, muito provavelmente, necessária a marcação de uma nova reunião. Os que não foram totalmente concluídos podem ser resolvidos de duas formas distintas:

- Estabelecimento de um grupo de trabalho;
- Adiar para outra reunião.

A primeira forma baseia-se na criação de um grupo de trabalho que, fora da reunião, vai pesquisar informação e estudar opções suficientes para a discussão clara do ponto de agenda, que fica assim resolvido fora da reunião.

Na segunda forma esse ponto de agenda será discutido na próxima reunião, não existindo trabalho desenvolvido entre reuniões. A opção por cada solução passa muito pela essência de cada ponto de agenda e do que pode ser realizado fora da reunião.

Por fim, para que a reunião traga frutos para a organização, os pontos de agenda finalizados e notas tomadas devem ser agrupadas e disponibilizadas a toda a organização.

3.2 Análise de Aplicações de gestão de reuniões existentes no mercado

Para melhor compreender as características fundamentais de uma aplicação de gestão de reuniões e acompanhamento de projetos, optou-se pela análise de aplicações existentes no mercado. As aplicações foram escolhidas segundo a sua importância no mercado e a importância das organizações que as utilizam. No documento Ref_4 é apresentada uma análise detalhada de cada uma das aplicações testadas. Esta análise permitiu também obter requisitos que viriam a completar a aplicação.

Antes ainda de ser iniciada a análise às aplicações, foram definidos alguns critérios a procurar nelas, juntamente com o cliente. Estes eram fundamentais para a aplicação a desenvolver pois iam de encontro aos métodos utilizados pelo cliente na atual gestão de reuniões e acompanhamento de projetos. Estes requisitos eram:

- a) Criação de reuniões
 - Inserção de reuniões;
 - Edição de reuniões;
 - Notificação dos participantes da existência de uma reunião;
 - Partilha da agenda da reunião;
- b) Criação de agenda partilhada
 - Submissão de pontos de agenda por parte de todos os utilizadores;
 - Submissão de anexos em pontos de agenda;
 - Edição de pontos de agenda submetidos;
 - Escolha de pontos de agenda a levar à reunião;
- c) Durante reunião
 - Plataforma de acompanhamento de reuniões;
 - Adição de notas aos pontos de agenda;
 - Construção da ata dinamicamente e envio para os participantes;
 - Deixar pontos de agenda por concluir;

- d) Acompanhamento de projetos
 - Inserção de novos projetos;
 - Submissão de atualizações nos projetos;
 - Edição dos projetos submetidos;
 - Separação de projetos por estado/tipo;
- e) Criar Pontos de ação
 - Inserção de pontos de ação para todos os utilizadores;
 - Marcar pontos de ação como finalizados;
- f) Utilização
 - Existência de um painel principal que permita ver todos os itens do utilizador;
 - Inserção de vários itens a partir do painel principal;
 - Design responsivo para utilização em dispositivos móveis;
 - Curva de aprendizagem reduzida e simplicidade na utilização;

Com base nas funcionalidades apresentadas foi realizado um estudo comparativo entre as diversas ferramentas. Na tabela seguinte são apresentados os resultados resumidos da comparação dos critérios definidos com o que foi encontrado na análise a cada aplicação.

Os grupos de funcionalidades apresentados anteriormente são identificados na tabela com a letra que os representa, tendo à frente o número de requisitos de cada grupo. Os resultados para cada grupo são apresentados com uma pontuação. Por exemplo, uma pontuação de 2, significa que a aplicação cumpriu dois requisitos, dos N apresentado nesse grupo. A atribuição de meio ponto, significa que a aplicação contém esse requisito, mas que não o cumpre na sua totalidade.

Nome	a) (4)	b) (4)	c) (4)	d) (4)	e) (2)	f) (4)	Total (22)
Novusolutions[7]	3	3	1	0	0	1.5	8.5
Mijura[8]	2	0.5	1.5	0	2	2	8
Meeting sense[9]	3	1.5	0	1.5	2	4	12
Less Meeting[10]	4	2	4	0	2	2	14
Meeting King[11]	2	3	3	0	2	1.5	11.5

Tabela 2 - Comparação entre ferramentas analisadas

Este estudo teve como objetivo obter duas informações bastante importantes: a forma como os requisitos pretendidos são apresentados pelas aplicações existentes no mercado e a obtenção de novos requisitos que podem ser importantes para acrescentar à aplicação. Pela tabela 2 é possível concluir que segundo os critérios definidos pelo cliente, as aplicações “*Less Meeting*” e “*Meeting King*” são as que apresentam uma maior compatibilidade com as especificações deste.

3.3 Conclusão do capítulo

Neste capítulo foram apresentadas boas práticas que devem ser seguidas para garantir uma boa preparação e condução de uma reunião. Para que as reuniões das organizações ganhem eficiência, fazendo com que os participantes aproveitem ao máximo o tempo destinado à reunião, é necessário distribuir a agenda da reunião com antecedência, uma melhor preparação de todos os participantes. É importante também que o moderador faça com que todos os participantes se sintam parte integrante da reunião, para que estes não percam o foco ou se desinteressem pelos assuntos em discussão.

Após a análise efetuada às aplicações os resultados foram apresentados e debatidos com o cliente em reunião, onde foram analisadas as mais valias de cada aplicação e alguns requisitos que poderiam ser importantes para a aplicação, como por exemplo algo que servisse para encadear reuniões ou ter uma estrutura que permitisse organizar os projetos. Desta análise concluiu-se que nenhuma aplicação apresentava todos os requisitos obrigatórios pois maior parte delas não tinha um módulo que permitisse o acompanhamento dos projetos ou a utilização era penosa e confusa. Esta última contrariava um dos principais requisitos do cliente, que pretendia uma aplicação muito fácil de utilizar e que apresentasse processos idênticos aos já adotados pelo IPN-LIS.

Posto isto, concluiu-se que a melhor solução seria construir uma aplicação à medida. Esta cumpriria estes critérios já apresentados pelo cliente, requisitos específicos que iriam ainda ser levantados junto do cliente e requisitos importantes que surgiram da análise às aplicações.

Capítulo 4

Levantamento de requisitos

Neste capítulo são descritos os requisitos necessários de forma a alcançar os objetivos propostos para o desenvolvimento do sistema de gestão de reuniões. Inicialmente são descritos os processos de gestão de reuniões e de acompanhamento de projetos utilizados pelo IPN-LIS e aos quais a ferramenta irá dar suporte, seguindo-se os perfis de utilizadores que irão utilizar a aplicação. Por último, irão ser apresentados os requisitos funcionais e não funcionais que suportam o sistema. São apresentadas também restrições à implementação que nascem de requisitos do cliente a ter em conta aquando da implementação.

4.1 Principais processos de negócio da aplicação

Tendo como objetivo planejar os requisitos funcionais necessários foram delineados os processos utilizados pelo IPN-LIS para a preparação e gestão de reuniões e para o acompanhamento de projetos. Esta especificação dos processos garante que, antes de definirmos os requisitos, os processos utilizados estão bem identificados. Isto permite realizar um levantamento de requisitos mais preciso, contemplando todos os detalhes necessários para satisfazer as necessidades do cliente.

Na figura 7, é apresentado o diagrama do processo utilizado para a gestão de reuniões. Podemos observar três momentos distintos bem definidos: a preparação da reunião, durante a reunião e após a reunião. A preparação da reunião é iniciada quando um utilizador cria uma nova reunião, a segunda fase começa quando uma reunião é iniciada, e a terceira fase quando um utilizador termina a uma reunião.

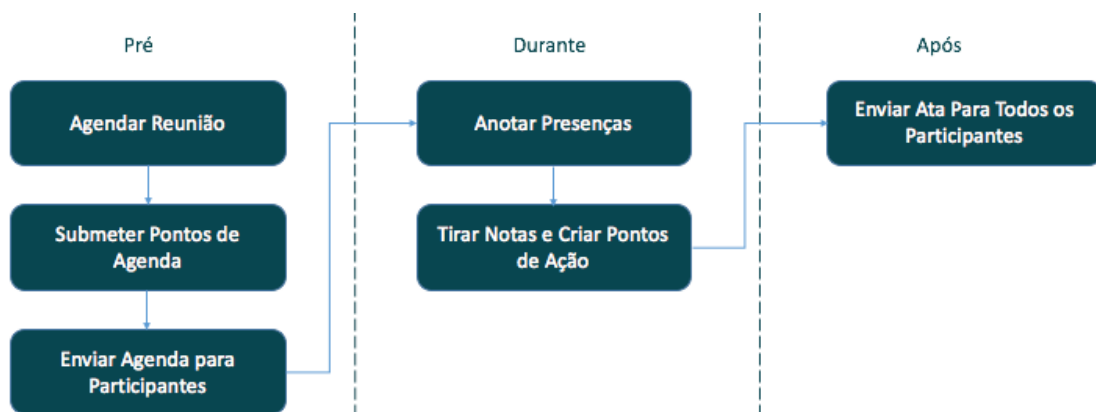


Figura 7 - Processo de gestão de reuniões

Quando a reunião é criada, o moderador envia um email para todos os participantes a avisar da existência desta. Após isso, todos os participantes podem submeter pontos de agenda, enviando-os por email ao moderador. Este escolhe e ordena os pontos de agenda que pensa serem mais importantes a ser discutidos em reunião e envia a agenda por email para todos

os participantes. Caso algum deles não possa estar presente na reunião, deve justificar a sua ausência ao moderador para que este tome nota dos participantes no início desta. Durante a reunião são tomadas notas em cada ponto de agenda que são guardadas num documento de texto. Para além disso são criados também pontos de ação para cada participante, que consistem em tarefas que um participante tem de realizar. Caso algum ponto de agenda não seja totalmente discutido numa reunião, é criado um ponto de follow-up correspondente para ser debatido nas reuniões seguintes. Este é adicionado ao documento de texto utilizado, bem como as suas notas. No final da reunião, este documento de texto é partilhado por todos os utilizadores e serve como ata.

Relativamente ao processo de acompanhamento de processos, este começa quando um utilizador insere um novo projeto na aplicação. Durante o curso de evolução do projeto, este passa por diferentes estados definidos pela direção do IPN-LIS que ajudam a perceber qual a situação de cada projeto. Estes estados são *Lead*, para projetos que ainda se encontram na fase inicial, ou seja, são representativos de candidaturas a programas de financiamento ou algum pedido de informação por parte de um cliente sobre um projeto, entre outros. De seguida passa para o estado *Proposed*, onde ou o pedido de financiamento foi aceite ou foi enviado algum tipo de orçamento ao cliente. Após isto, evolui para o estado de *Ongoing* que corresponde à fase de execução de um projeto. Quando o projeto é concluído o seu estado é alterado para *Completed*. Neste estado o desenvolvimento do projeto já aconteceu mas ainda falta algum pormenor importante tal como a emissão de faturas, o pagamento por parte do cliente, entre outros. Por fim, quando todo o processo é concluído, o projeto é alterado para o estado *Arquivado*.

Este é o fluxo normal de um projeto, sendo que existem alguns que começam no estado de *Proposed*, ou já no estado de *Ongoing*. Para projetos que não se concretizam, quer por não terem sido aceites a financiamento ou por o cliente não aceitar o orçamento, o seu estado é alterado automaticamente para *Arquivado*.

No final das reuniões de direção existe sempre a discussão sobre os diversos projetos. Aqui, os gestores de cada projeto informam a direção das alterações ocorridas desde a última reunião e registam essas alterações num documento de texto que contém a informação de todos os projetos desde que são iniciados até que terminam.

4.2 Perfis de utilizadores

O IPN-LIS toma as suas decisões mais importantes nas reuniões de direção. É assim necessário proteger o acesso a informação sensível e crítica por parte dos utilizadores que não têm permissão. De forma a garantir isto foram identificados 3 perfis de utilizadores para a aplicação. Esta escolha contou com a opinião e conhecimento de gestores de projetos do LIS-IPN. Os perfis são:

- Utilizador da Aplicação – Atribuído a um utilizador normal, que não pertence a nenhum grupo, mas que mesmo assim pode ser participante de reuniões. Um exemplo de um grupo pode ser o “Grupo de Direção”, que tem associados os utilizadores que estão presentes nas reuniões de direção.
- Membro de um Grupo – Atribuído a um utilizador que pertence a um ou mais grupos;

- Administrador – Utilizador que tem permissão para visualizar e/ou criar todos os itens da aplicação;

Os utilizadores da aplicação pertencem obrigatoriamente a um destes perfis.

A autorização no acesso a cada página é efetuada ao nível do controlador verificando que qual o perfil do utilizador, permitindo ou não a visualização de certa página ou a inserção de novos dados. O administrador pode realizar todas os requisitos que os outros dois perfis podem.

4.3 Requisitos funcionais

De seguida são apresentados os requisitos funcionais que o estagiário identificou em conjunto com o cliente para que a aplicação cobrisse todas as suas necessidades.

O método escolhido para o levantamento de requisitos foi a identificação de user stories. Este método permite a resposta a 3 perguntas fundamentais que são "Quem?", "O quê?" e por fim "Porquê?". Com a resposta a estas perguntas obtém-se uma estrutura para a user story[12]. A sua principal vantagem é a facilidade e rapidez de escrita e entendimento[13], pois como são escritas em linguagem não técnica é possível descrever a atividade à qual o requisito vai dar resposta de uma maneira clara e precisa.

No documento Ref_1 são descritas aprofundadamente cada uma das user stories que deram origem aos requisitos apresentados nas seguintes tabelas.

Apesar de ter ocorrido uma fase de levantamento de requisitos numa fase inicial do estágio, foram sendo introduzidos mais ao longo do processo de desenvolvimento. Para a priorização de requisitos foi utilizada a escala de MoSCoW[14].

4.3.1 RF1 – Módulo de Reuniões

O módulo de reuniões lista os requisitos que envolvem a gestão de reuniões da aplicação. Na tabela seguinte são listados os requisitos do presente módulo. A ação temporal permite perceber quando é que um requisito pode ser realizado na aplicação. Para alguns requisitos este era um aspeto muito importante pois permitia perceber onde o utilizador poderia realizar algum requisito e o seu *scope*. Esta informação irá ser apresentada apenas onde for necessária de forma a ajudar na correta interpretação do requisito.

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF1.1	Criar reunião.	Membro de um grupo		MUST
RF1.2	Editar reunião.	Membro de um grupo		MUST
RF1.3	Ver lista de reuniões.	Utilizador da aplicação		MUST
RF1.4	Filtrar lista de reuniões por estado de reunião.	Utilizador da aplicação		MUST
RF1.5	Filtrar lista de reuniões por utilizador.	Utilizador da aplicação		MUST

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF1.6	Pesquisar na lista de reuniões.	Utilizador da aplicação		MUST
RF1.7	Ver informações de uma reunião.	Utilizador da aplicação		MUST
RF1.8	Introduzir pontos de agenda na reunião ou grupo da reunião.	Utilizador da aplicação	Antes e durante reunião	MUST
RF1.9	Editar ponto de agenda.	Utilizador da aplicação	Antes e durante reunião	MUST
RF1.10	Criar pontos de follow-up.	Utilizador da aplicação	Durante reunião	MUST
RF1.11	Ver informações de um ponto de agenda.	Utilizador da aplicação		MUST
RF1.12	Editar ponto de follow-up.	Utilizador da aplicação		MUST
RF1.13	Adicionar logs a ponto de follow-up.	Utilizador da aplicação	Durante reunião	MUST
RF1.14	Fechar ponto de follow-up.	Utilizador da aplicação	Durante reunião	MUST
RF1.15	Confirmar reunião.	Membro de um grupo		MUST
RF1.16	Escolher que pontos de agenda levar a reunião.	Membro de um grupo		MUST
RF1.17	Notificar participantes, por email, com a agenda da reunião.	Sistema		MUST
RF1.18	Justificar ausência a uma reunião.	Utilizador da aplicação	Antes e durante reunião	MUST
RF1.19	Anotar participantes de uma reunião.	Utilizador da aplicação		MUST
RF1.20	Passar pontos de agenda para reuniões futuras.	Utilizador da aplicação	Durante reunião	MUST
RF1.21	Adiar reunião.	Utilizador da aplicação		SHOULD
RF1.22	Ordenar pontos de agenda.	Membro de um grupo	Antes e durante reunião	MUST
RF1.23	Adicionar notas a pontos de agenda.	Utilizador da aplicação	Durante reunião	MUST
RF1.24	Adicionar decisões a pontos de agenda.	Utilizador da aplicação	Durante reunião	MUST
RF1.25	Construir ata de reunião automaticamente.	Sistema		MUST
RF1.26	Enviar ata por email para todos os participantes.	Sistema		MUST
RF1.27	Ver ata da reunião.	Utilizador da aplicação		MUST

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF1.28	Adicionar ficheiros a pontos de agenda.	Utilizador da aplicação		MUST
RF1.29	Ter paginação nas listas de reuniões.	Utilizador da aplicação		MUST
RF1.30	Adicionar pontos de ação a uma reunião.	Utilizador da aplicação	Durante reunião	MUST
RF1.31	Adicionar nota automática a um ponto de agenda quando é submetido um ponto de ação.	Sistema		MUST

Tabela 3 - Requisitos módulo de reuniões

A razão de um utilizador da aplicação ter tantas permissões é explicada pelo facto de que este pode ser convocado para uma reunião mesmo sem pertencer a nenhum grupo. Nesse caso só irá ter acesso a essa reunião do grupo.

4.3.2 RF2 – Módulo de Pontos de Ação

O módulo de pontos de ação consiste nos requisitos relacionados com os pontos de ação existentes na aplicação.

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF2.1	Adicionar pontos de ação a projetos.	Membro de um grupo		MUST
RF2.2	Editar ponto de ação.	Utilizador da aplicação		MUST
RF2.3	Ver lista de pontos de ação.	Utilizador da aplicação		MUST
RF2.4	Filtrar lista de pontos de ação por estado do ponto de ação.	Utilizador da aplicação		MUST
RF2.5	Filtrar lista de pontos de ação por utilizador.	Utilizador da aplicação		MUST
RF2.6	Pesquisar na lista de pontos de ação.	Utilizador da aplicação		MUST
RF2.7	Notificar criação de ponto de ação.	Sistema		COULD
RF2.8	Notificar deadline de ponto de ação.	Sistema		COULD
RF2.9	Adicionar log num ponto de ação.	Utilizador da aplicação		MUST
RF2.10	Editar um log de um ponto de ação.	Utilizador da aplicação		MUST
RF2.11	Fechar ponto de ação.	Utilizador da aplicação		MUST
RF2.12	Visualizar informações de ponto de ação.	Utilizador da aplicação		MUST

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF2.13	Paginação nas listas de pontos de ação	Utilizador da aplicação		MUST
RF2.14	Inserir ponto de situação num projeto automaticamente quando um ponto de ação é editado.	Sistema		MUST
RF2.15	Adicionar ponto de situação automático quando um log de um ponto de ação é editado.	Sistema		MUST

Tabela 4 - Requisitos módulo de pontos de ação

4.3.3 RF3 – Módulo de Grupos

No módulo de grupos são definidos os requisitos referentes aos grupos, bem como alguns que especificam o que deve ser possível de encontrar quando se vê os detalhes de um grupo.

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF3.1	Criar grupo.	Utilizador da aplicação		MUST
RF3.2	Adicionar utilizadores a um grupo.	Utilizador da aplicação		MUST
RF3.3	Adicionar projetos a um grupo.	Membro de um grupo		MUST
RF3.4	Editar grupo.	Membro de um grupo		MUST
RF3.5	Visualizar informações de um grupo.	Membro de um grupo		MUST
RF3.6	Ver lista de reuniões do grupo.	Membro de um grupo		MUST
RF3.7	Ver lista de projetos do grupo.	Membro de um grupo		MUST
RF3.8	Ver lista de pontos de ação de um grupo.	Membro de um grupo		MUST
RF3.9	Ver lista de pontos de agenda de um grupo	Membro de um grupo		MUST
RF3.10	Pesquisar nas listas dos grupos.	Membro de um grupo		MUST
RF3.11	Paginação nas listas de grupos.	Membro de um grupo		MUST

Tabela 5 - Requisitos módulo de grupos

4.3.4 RF4 – Módulo de Projetos

O módulo de projetos consiste em operações de acompanhamento de projetos. Qualquer pessoa, desde que pertença a um grupo, pode adicionar projetos na aplicação.

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF4.1	Criar projeto.	Membro de um grupo		MUST
RF4.2	Editar projeto.	Membro de um grupo		MUST
RF4.3	Adicionar projeto a grupos.	Membro de um grupo		MUST
RF4.4	Ver lista de projetos.	Membro de um grupo		MUST
RF4.5	Filtrar lista de projetos por estado do projeto.	Membro de um grupo		MUST
RF4.6	Filtrar lista de projetos por utilizador responsável.	Membro de um grupo		MUST
RF4.7	Pesquisar na lista de projetos.	Membro de um grupo		MUST
RF4.8	Visualizar informações de um projeto.	Membro de um grupo		MUST
RF4.9	Adicionar um ponto de situação a um projeto.	Membro de um grupo		MUST
RF4.10	Editar um ponto de situação de um projeto.	Membro de um grupo		MUST
RF4.11	Adicionar um ponto de situação como ponto de agenda de uma reunião.	Membro de um grupo		SHOULD
RF4.12	Obter histórico de alterações a um ponto de situação.	Membro de um grupo		MUST
RF4.13	Alterar estado de um projeto.	Membro de um grupo		MUST
RF4.14	Marcar visita a um projeto.	Membro de um grupo		MUST
RF4.15	Retirar métricas dos projetos.	Membro de um grupo		COULD
RF4.16	Realçar projeto consoante a última atualização feita nele.	Sistema		MUST
RF4.17	Consultar projeto no decorrer de uma reunião.	Membro de um grupo	Durante reunião	MUST
RF4.18	Ecrã onde seja possível atualizar todos os projetos de um grupo.	Membro de um grupo		MUST
RF4.19	Ter o link para aceder ao projeto no sistema de informação do IPN.	Membro de um grupo		MUST
RF4.20	Paginação nas listas de projetos.	Sistema		MUST

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF4.21	Adicionar ponto de situação automático quando é criado um ponto de ação associado a um projeto.	Sistema		MUST
RF4.22	Adicionar ponto de situação automático quando o estado do projeto é alterado.	Sistema		MUST
RF4.23	Ordenar os projetos pela data em que sofreram a última alteração.	Sistema		MUST
RF4.24	Não permitir editar os pontos de situação automáticos.	Sistema		MUST

Tabela 6 - Requisitos módulo de projetos

4.3.5 RF5 – Módulo de utilizadores

Neste módulo estão especificados os requisitos referentes à gestão de utilizadores da aplicação.

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF5.1	Adicionar utilizadores.	Administrador		MUST
RF5.2	Editar utilizadores.	Administrador e Utilizador da aplicação		MUST
RF5.3	Listar utilizadores.	Administrador		MUST
RF5.4	Pesquisa de utilizadores.	Administrador		MUST
RF5.5	Efetuar login com username e password.	Utilizador da aplicação		MUST

Tabela 7 - Requisitos módulo de utilizadores

4.3.6 RF6 – Módulo de interface com o utilizador

Neste módulo são expostos os requisitos do cliente referentes a alguns pormenores que necessitam de estar presentes na interface.

Código	Descrição	Perfis de Utilizador	Ação Temporal	Prioridade
RF6.1	Forma de navegar entre os diferentes menus sem passar pela página inicial da aplicação.	Utilizador da aplicação		MUST

RF6.2	Internacionalização(Português e Inglês)	Utilizador da aplicação		MUST
RF6.3	Página principal com dados sobre reuniões, pontos de ação, projetos e grupos.	Utilizador da aplicação		MUST
RF6.4	Design da GUI pelo estagiário.			MUST
RF6.5	Criar diversos itens a partir da página principal	Utilizador da aplicação		MUST

Tabela 8 - Requisitos módulo de interface

4.4 Requisitos não-funcionais

Em relação a requisitos não-funcionais foi identificado apenas 1:

Código	Descrição
RNF.1	Usabilidade elevada da aplicação

Tabela 9 - Requisitos não funcionais

Para retirar métricas sobre este requisito foram realizados testes de usabilidade à aplicação a partir de mockups de design que o estagiário elaborou (requisito RF6.4). Os mockups sofreram um processo de prototipagem evolutiva, com diversos ciclos de desenvolvimento.

Cada ciclo consistiu no desenvolvimento de mockups para a aplicação que eram de seguida debatidos com o cliente em reunião e anotado o feedback. Após a reunião, eram realizadas as alterações pretendidas e repetia-se o ciclo de desenvolvimento/reunião até o cliente estar satisfeito com o resultado. Os resultados destes testes de usabilidade podem ser vistos mais à frente no presente relatório.

4.5 Restrições de implementação

Para além dos requisitos funcionais e não funcionais, foram apresentados pelo cliente algumas especificações para a aplicação que obrigaram a algumas restrições na implementação. Sendo assim, as tecnologias escolhidas para o desenvolvimento da aplicação necessitam de possuir mecanismos que permitam dar resposta a estas restrições. As restrições são apresentadas na tabela 10.

Código	Descrição
RI.1	Multi-tenancy, ser utilizado como Software as a Service.
RI.2	Utilização em dispositivos móveis.

RI.3	Adicionar informação em tempo real nos ecrãs de todos os utilizadores.
------	--

Tabela 10 - Restrições de implementação

Após a apresentação dos requisitos, é detalhado cada um e as suas implicações tecnológicas.

4.5.1 Multi-Tenancy

O IPN-LIS, cliente da aplicação, é um dos laboratórios existentes no IPN entre muitos outros. Para garantir que no futuro esta aplicação possa ser expandida a outros laboratórios ou vender a clientes externos que possam estar à procura de uma aplicação com estas ou com características semelhantes, é necessário que esta apresente uma arquitetura preparada para ser utilizada como Software as a Service(SaaS).

SaaS começou como uma ideia no início dos anos 60. Com o crescendo da utilidade dos recursos informáticos, foi criada a noção de um ambiente de partilha de recursos.

O acesso a software como um serviço sofreu ao longo dos últimos tempos uma grande evolução, tendo o número de organizações a utilizar este tipo de serviços aumentado em grande número[15]. Antes da explosão do SaaS, o típico para as organizações quando necessitavam de um determinado software era ou desenvolverem ou comprarem, ficando com o software instalado nas suas próprias máquinas para os funcionários da empresa poderem aceder. Isto apresenta custos elevados para a organização devido ao investimento em hardware que é necessário para suportar algumas aplicações. Com a introdução de SaaS, as organizações têm agora uma opção de pagarem pelo acesso ao software, que chega até eles como um serviço[16].

Para perceber que tipo de SaaS é que uma organização necessita criaram-se níveis de maturidade para descrever o modelo de negócio que a organização necessita. Estes distinguem-se por 3 características principais:

- Configurabilidade;
- Escalabilidade;
- Eficiência.

São de seguida apresentadas as características de cada nível de maturidade, segundo Carraco, G, 2006[17]:

- **Nível 1** (Ad-Hoc/Personalizado) – Neste nível cada cliente(inquilino) tem a sua versão personalizada da aplicação que é executada num servidor central, ao qual acede pela internet. A única forma de suportar múltiplos clientes é servi-los com cópias diferentes do software, não existindo assim conceito de multi-tenancy. Não é possível existir economia de escala pois cada cliente corre uma instância diferente do software. Torna-se difícil para a organização que produz o software gerir muitos clientes, devido ao elevado número de atualizações ou especificações de cada um.
- **Nível 2** (Configurável) – Tal como no nível 1, neste nível cada cliente tem uma instância separada e independente das outras. No entanto, a instância de cada cliente

é igual, ou seja, têm o mesmo código fonte. A sua característica principal é oferecer opções de configuração da aplicação aos clientes. O principal benefício neste nível é a redução de custos de manutenção do software. No entanto, tal como no nível anterior, o prestador de serviços necessita de fornecer hardware e armazenamento a um possível elevado número de instâncias. Para além disso, apresenta um elevado tempo para a atualização de todas as instâncias.

- **Nível 3** (Configurável e eficiente) – No terceiro nível, o prestador de serviços fornece apenas uma instância para todos os clientes, continuando a oferecer opções de configuração para cada um. Para garantir o isolamento de dados entre clientes, são utilizados mecanismos de autenticação e segurança. O principal benefício deste nível é a diminuição dos custos em hardware para o prestador de serviços, permitindo um uso mais eficiente dos recursos. Para além disso, a existência de uma única instância permite que seja possível efetuar apenas uma atualização de software, dividindo o custo desta manutenção por todos os clientes. Pelo contrário, este nível tem a desvantagem de ser menos escalável que as anteriores, pois as instâncias estão limitadas ao hardware existente.
- **Nível 4** (Escalável, configurável e eficiente) – No último nível de maturidade, o prestador de serviços fornece várias instâncias idênticas com um balanceador de carga. O grande benefício deste nível de maturidade é oferecer a escalabilidade que o nível anterior não apresentava.

A figura seguinte, apresenta a arquitetura global dos quatro níveis de maturidade apresentados anteriormente.

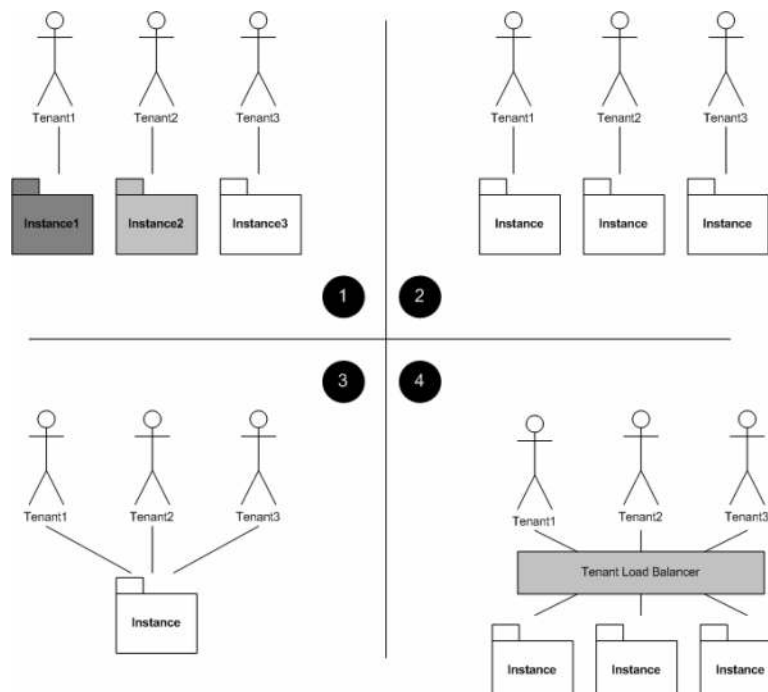


Figura 8 - Arquitetura dos níveis de maturidade de SaaS [16]

Para a restrição de implementação apresentada, o cliente apresentou os seguintes constrangimentos:

- Utilizar poucos recursos de hardware;
- Custo de manutenção baixo;

Após a análise dos diferentes níveis de maturidade e dos constrangimentos do cliente, verificou-se que o nível de maturidade que se relaciona mais com o conceito de SaaS pretendido pelo cliente é o nível de maturidade 3. Explica-se pelo facto existir apenas uma única instância que garante um baixo custo de manutenção e de utilização de recursos. Entre o nível escolhido e o nível de maturidade 4, foi tomada esta decisão por o número de organizações e consequentemente utilizadores a poderem utilizar a aplicação no futuro não ser significativamente elevado para justificar a implementação de um balanceador de carga.

4.5.2 Utilização em dispositivos móveis

Outro requisito do cliente que provoca uma restrição à implementação é a utilização das funcionalidades da aplicação em dispositivos mobile, pelo menos das que permitem inserir ou confirmar dados. Para isso, a aplicação deve possuir design responsivo.

De modo a conseguir o que o cliente pretende, serão adicionadas aos ficheiros CSS media-queries, que permitem estilizar elementos de diferente modo, dependendo do tamanho do ecrã de onde o utilizador tenta aceder à aplicação.

4.5.3 Adicionar informação em tempo real nos ecrãs de todos os utilizadores

Na aplicação existem ecrãs que irão ser sempre utilizados por vários utilizadores ao mesmo tempo, como por exemplo o ecrã apresentado durante uma reunião ou o que permite realizar o acompanhamento de projetos. Para que todos os participantes possam acompanhar da melhor forma possível a reunião, a informação inserida deve ser disponibilizada para todos os participantes assim que um utilizador a insere. Deste modo, é necessário implementar um mecanismo que permita que surja nos ecrãs de todos os utilizadores assim que é adicionada.

A tecnologia escolhida para realizar este requisito recaiu nos WebSockets. A forma como estes são implementados na aplicação pode ser vista mais à frente, no capítulo dedicado ao desenvolvimento.

4.6 Critérios para a escolha da tecnologia a utilizar no desenvolvimento

Durante a fase do planeamento e de levantamento de requisitos, verificou-se que era necessário escolher uma tecnologia para o desenvolvimento que estivesse de acordo com três critérios:

- **Flexibilidade para introduzir novos requisitos** – Era necessário identificar uma tecnologia que permitisse não só o desenvolvimento segundo a metodologia escolhida e que tivesse em conta as entregas funcionais que iam acontecer, mas também que facilitasse a entrada de novos requisitos por parte do cliente, sem que isso afetasse em muito o tempo estimado para conclusão do projeto;
- **Curva de aprendizagem baixa** – Era necessário que a tecnologia a utilizar apresentasse uma curva de aprendizagem reduzida para permitir uma aprendizagem rápida por parte do estagiário. Tinha também em conta outros fatores como a documentação existente ou o conhecimento e utilização da tecnologia por parte do IPN-LIS;

- **Custo** – As tecnologias escolhidas não deveriam ter qualquer custo associado ou possuir qualquer forma de licenciamento que restringisse alguma funcionalidade;

Após consultoria com diversos colaboradores do IPN-LIS, Eng. Carlos Lopes e Eng. Alcides Marques, foram listadas algumas tecnologias que cumprissem os critérios definidos. Estas eram:

- Java + *Framework* play!
- Java + *Framework* Struts
- Ruby + *Framework* Ruby on Rails
- Python + *Framework* Django

Orientado pelos critérios e pela opinião dos colaboradores do IPN-LIS, a tecnologia escolhida pelo estagiário para o desenvolvimento da aplicação foi a linguagem Java com o apoio da *Framework* play! na versão 2.4, que era a versão mais recente aquando do começo do desenvolvimento. Esta é uma *Framework* gratuita e open-source, com documentação detalhada no website[18] e que já tinha sido utilizada em outros projetos do IPN-LIS. Esta escolha reduzia também a curva de aprendizagem do estagiário pois tinha apoio próximo dos colaboradores do IPN-LIS em qualquer dúvida relacionada com a *Framework*.

4.7 Conclusão do capítulo

Neste capítulo foi apresentada a especificação dos requisitos do projeto apurados pelo estagiário. Estes passaram por uma fase de maturação, tendo sido discutidos em reuniões com o cliente ao longo do processo de levantamento. Refletiram a informação recolhida não só durante essas reuniões, mas também durante a fase de análise a aplicações semelhantes.

A elaboração dos processos de gestão de reuniões e acompanhamento de projetos também ajudou na clarificação de alguns requisitos, pois fez com que o processo utilizado atualmente pelo IPN-LIS fosse bem analisado e daí resultassem todos os requisitos para suportar esse processo.

No que diz respeito às restrições de implementação, estas ajudaram a clarificar quais as tecnologias a utilizar, e a definir a arquitetura de SaaS.

Foram identificados critérios que influenciavam a escolha da tecnologia a utilizar. Com base neles, foi elaborada uma lista de tecnologias para o desenvolvimento da aplicação, tendo sido escolhida a linguagem Java com a *Framework* play!.

O conhecimento desta tecnologia por parte de alguns colaboradores do IPN-LIS, proveniente da experiência em projetos anteriores, reduziu a curva de aprendizagem do estagiário, permitindo assim mais tempo para a implementação dos requisitos e aumentando a produtividade.

Capítulo 5

Arquitetura da aplicação

Neste capítulo é descrita a arquitetura da aplicação desenvolvida, segundo vários níveis de abstração. Para a descrição da arquitetura foi seguida uma metodologia proposta pelo autor Simon Brown no livro *The Art of Visualizing Software* [24]. Para o desenho dos diagramas da arquitetura foi utilizada uma ferramenta produzida pelo autor do livro, *Structurizr* [25].

5.1 Nível 1 – Diagrama de contexto

Um diagrama de contexto é visto como um ponto de partida para a construção de diagramas e documentação da arquitetura de uma aplicação ou sistema. Este permite ver o *scope* total da aplicação, ajudando na perceção de quem vai utilizar o sistema, em que consiste, e como se inclui entre os diversos sistemas à sua volta. O diagrama de contexto para a aplicação pode ser visto na figura 9.

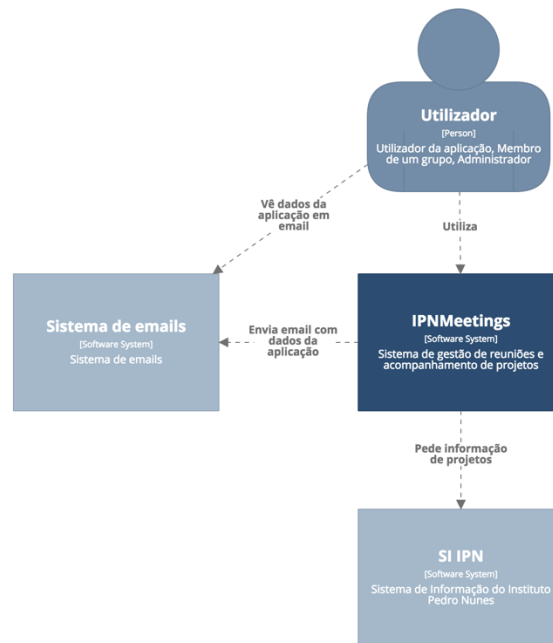


Figura 9 - Diagrama de nível 1 do sistema IPNMeetings

Neste diagrama podemos perceber que os 3 perfis de utilizador que foram abordados anteriormente acedem à aplicação da mesma forma. Podemos também notar que o sistema interage com o sistema de informação do Instituto Pedro Nunes, para dar resposta ao requisito RF4.19. Para realizar esta interação, a aplicação faz um pedido REST ao SI e recebe uma lista de todos os projetos registados ali que é apresentada ao utilizador. Este escolhe qual o projeto, sendo guardado um link para aceder à página do projeto no SI. Para além deste, a aplicação utiliza sistemas de emails para enviar emails de notificações, aquando da criação de reunião ou confirmação de reunião, por exemplo. Apesar dos emails serem tratados e enviados dentro da aplicação, através dos plugins *play-mailer* e *play-easymail*, o

escritor Simon Brown refere que esta interação deve ser representada neste diagrama de contexto.

5.2 Nível 2 - Diagrama de *container*

Este diagrama pretende ser uma vista detalhada dos componentes do sistema central do diagrama anterior, sistema IPNMeetings.

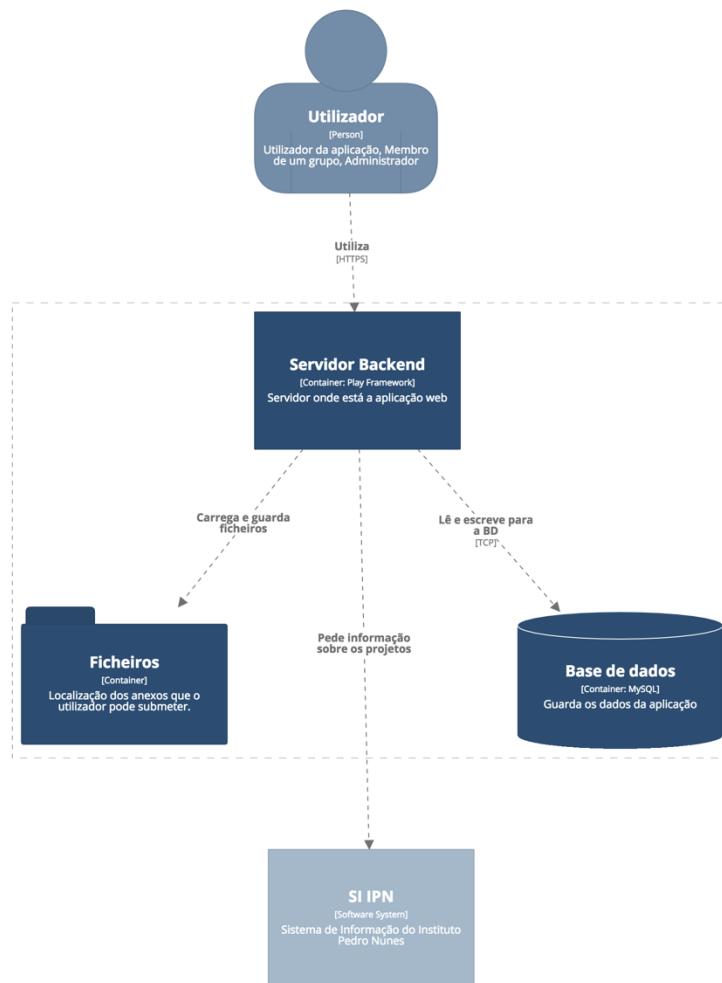


Figura 10 - Diagrama de nível 2 do sistema IPNMeetings

O utilizador faz pedidos ao servidor BackEnd através das vistas que são apresentadas no browser. Este utiliza a base de dados para obter informação e retornar uma página nova com essa informação ou para submeter formulários. A base de dados, desenvolvida em MySQL, guarda a informação da aplicação.

O *container* de ficheiros representa as pastas onde são guardados os ficheiros submetidos pelos utilizadores como anexos de pontos de agenda. Para garantir uma maior segurança no acesso a dados, os ficheiros são separados por pastas, pertencendo uma pasta a cada *tenant* da aplicação.

5.3 Nível 3 - Diagrama de componentes

Este diagrama pretende detalhar o *container* “Servidor Backend” da figura anterior, apresentando os diversos componentes que fazem parte do sistema. Aqui, pode ser vista a arquitetura MVC(Model-View-Controller) implementada através da *framework play!*.

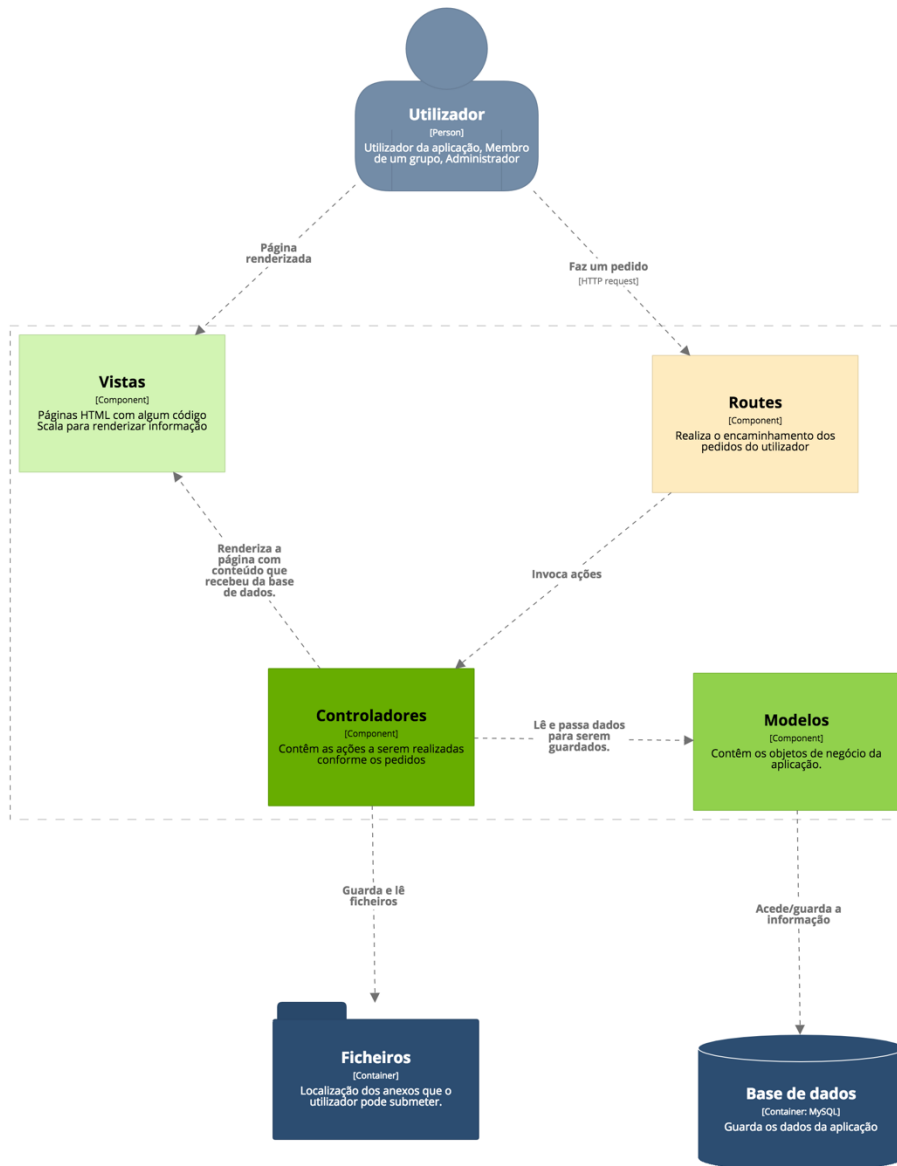


Figura 11 - Diagrama de nível 3 do sistema IPNMeetings

Um utilizador faz um pedido HTTPS de GET ou POST numa vista que chega a um componente de encaminhamento. Com base no tipo e URL do pedido, o componente de encaminhamento envia-o para uma action específica dentro dos diferentes controladores. Estes pedem dados ao modelo correspondente, que está encarregue apenas de ler dados da base de dados ou gravar dados nesta. Os dados são devolvidos ao controlador, que com eles renderiza uma página. Esta contém código HTML, CSS, Javascript e pequenas partes de

código Scala que é utilizado pela *Framework play!* para renderizar o que o controlador devolve. Após a renderização da página, esta é devolvida ao utilizador, completando assim o pedido.

5.4 Arquitetura Bases de Dados Multi-Tenancy

Para garantir a segurança nos dados de cada cliente, existe uma base de dados distinta para cada organização. Segundo Gianpaolo Carraro[26], esta é a melhor forma de garantir a segurança nas bases de dados. Com base na documentação analisada e aconselhado por engenheiros do IPN-LIS, tomada a decisão de optar por bases de dados separadas para cada organização.

Uma representação básica pode ser vista na figura 12.

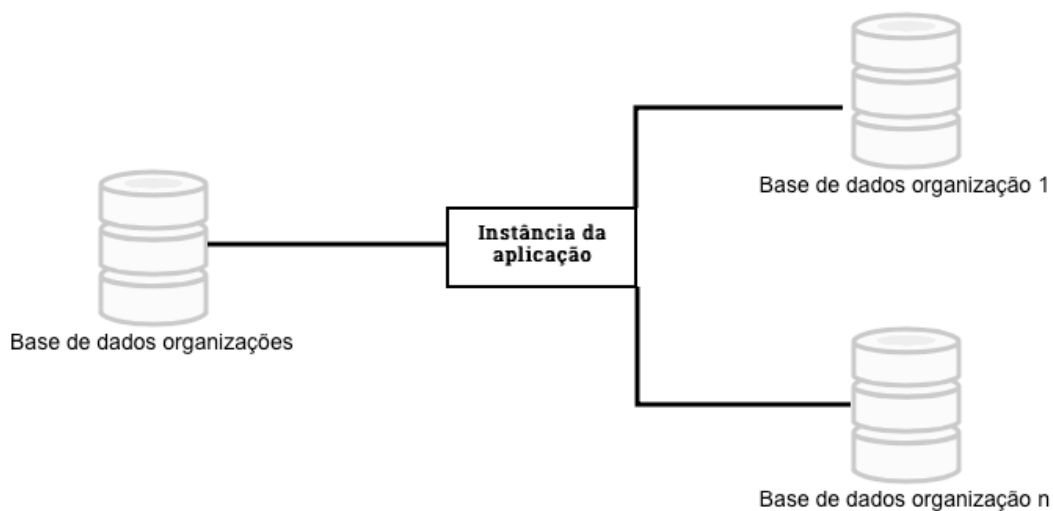


Figura 12 - Esquema das diferentes bases de dados

Para a *framework play!* se conectar a uma base de dados, precisa de um conjunto de dados no ficheiro *Application.conf*. Este conjunto de dados são: a driver para identificar o sistema de gestão de bases de dados (MySQL no caso da aplicação), o endereço da base de dados, um utilizador com permissões na BD e a password desse utilizador. Para que quando se inserisse uma nova base de dados não fosse necessário alterar o ficheiro de configurações manualmente, procurou-se uma solução que permitisse inserir novos clientes na aplicação de forma dinâmica. Para responder a isto, foram criados dois esquemas diferentes para as bases de dados, como é possível ver na figura 12.

A base de dados a que a aplicação se liga diretamente, através das configurações no ficheiro *Application.conf*, é representada na figura anterior pela “Base de dados organizações”. Esta contém apenas as informações necessárias para a conexão à base de dados de cada organização, representadas na mesma figura pela “Base de dados organização 1” e “Base de dados organização n”. No método *onStart* da *play! framework*, que arranca no início da aplicação, é corrido um método que lê as informações da base de dados de cada organização, aplica as evoluções às bases de dados caso exista alguma alteração nos modelos e cria um servidor *Ebean* para cada uma delas. Cada servidor tem um nome diferente, que se encontra gravado em BD com as outras informações da base de dados. Este servidor é o que vai lidar com os pedidos à base de dados por parte dos modelos de dados da aplicação. Garantimos

assim que no arranque da aplicação no servidor, todas as organizações têm a sua própria base de dados a que se podem conectar. Uma representação do processo pode ser vista na figura 13.

Após a criação de uma base de dados distinta para cada cliente, era necessário identificar a que BD ligar quando um utilizador iniciava sessão na aplicação.

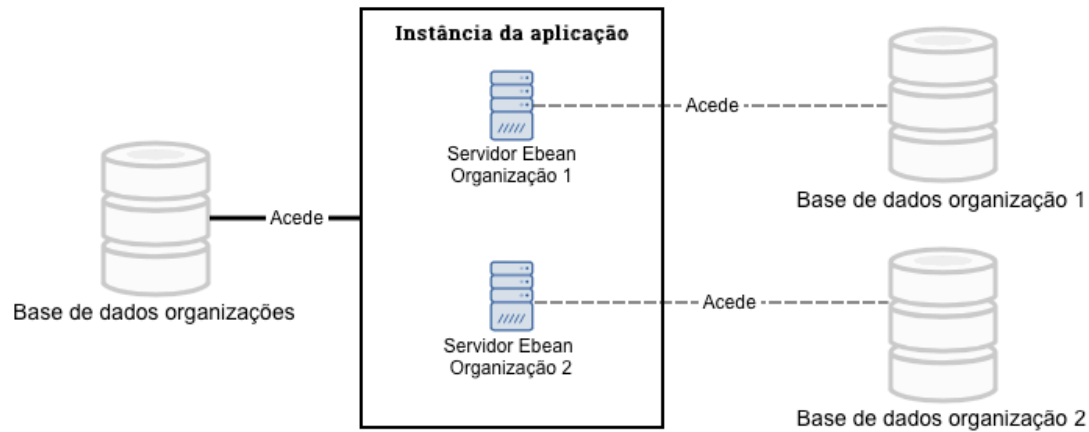


Figura 13 - Configuração dos servidores Ebean

Aquando do login, o utilizador escolhe numa *select box* qual a organização a que pertence. O valor desse *select box* retorna o nome do servidor *Ebean* desse cliente. Com este nome do servidor, a aplicação faz um pedido à base de dados para verificar se esse utilizador existe na base de dados específica. Caso não exista, notifica o utilizador com “Nome de Utilizador ou palavra passe inexistente”, para que assim não se perceba se é erro apenas no username ou password, ou se é mesmo a base de dados daquela organização que não contém um determinado utilizador. Caso contrário, se a base de dados da organização validar o username e password do utilizador como corretos, o utilizador é redirecionado para a página principal da aplicação, e o nome do servidor *Ebean* da organização a que pertence é guardado na *session*, para ser utilizado em todos os pedidos à BD a partir daí.

5.5 Modelos da aplicação

Após definir como a aplicação acede à informação da base de dados de cada cliente, são apresentados todos os modelos criados no desenvolvimento da aplicação. Na tabela 12 são listados estes modelos e dada uma explicação do que representam. Na listagem dos atributos de cada entidade, não estão a ser tidas em conta as relações com outras tabelas. Essas relações serão apresentadas mais à frente, no diagrama do modelo de dados da aplicação.

Modelo	Descrição
Utilizador	Entidade <i>utilizadores</i> – Este objeto contém a informação acerca de um utilizador como nome, nomeProprio, password, email e a informação se tem o perfil de administrador ou não.
Grupo	Entidade <i>grupos</i> – Este objeto contém o nome de cada grupo, e serve como um agrupador dos outros modelos.

Reunião	Entidade <i>reunioes</i> – Este tipo de objeto contém informação acerca das reuniões (nome, data e hora, local, duração, estado). O estado da reunião significa se ela se encontra aberta(0), confirmada(1) ou se já ocorreu(2).
PontoAgenda	Entidade <i>pontos_agenda</i> – Este objeto representa um ponto de agenda e contém o ponto de agenda, a duração, a ordem em que vai ser discutido numa reunião e o estado. O estado refere-se a aberto(0) e discutido(1).
LogPontoAgenda	Entidade <i>log_ponto_agenda</i> – O objeto representa uma nota inserida num ponto de agenda durante uma reunião, por um participante da reunião. Contém os campos do log feito e da data em que foi inserido.
Projeto	Entidade <i>projetos</i> – Este tipo de objeto representa um projeto. Contém o nome, cliente, estado, tipo, valor, datas associadas ao projeto e o seu nome e id no SI do IPN. O estado do projeto varia entre Lead(0), Proposed(1), Ongoing(2), Completed(3) e Arquivado(4). Estes são os estados utilizados pela direção do IPN-LIS para categorizar o processo diretivo da evolução de um projeto.
PontoSituacao	Entidade <i>pontos_situacao</i> – Este objeto corresponde a um ponto de situação inserido num projeto. Contem o ponto de situação e a data em que foi inserido.
PontoSituacaoEditado	Entidade <i>pontos_situacao_editados</i> – Cada objeto deste tipo corresponde a uma edição a um ponto de situação. Como os pontos de situação inseridos são parte importante da evolução de um projeto, é guardado um histórico de alterações para que possa ser visto quem alterou um ponto de situação e quando.
PontoAcao	Entidade <i>pontos_acao</i> – Cada objeto representa um ponto de ação criado para um utilizador. Contém o texto do ponto de ação, a data limite do ponto de ação e a data em que foi criado.
PontoAcaoLog	Entidade <i>pontos_acao_log</i> – Cada objeto representa um log inserido num ponto de ação. Contém os campos de log, e a data em que foi inserido.
FollowUp	Entidade <i>follow_up</i> – Cada objeto deste tipo representa um ponto de follow-up, a ser seguido em reuniões de um grupo até ser terminado. Tem como atributos o <i>pontoFollowUp</i> , a data em que foi criado e o estado, que significa se se encontra aberto(0) ou se já foi terminado(1).
LogFollowUp	Entidade <i>log_follow_up</i> – Cada objeto representa um log que foi realizado num determinado ponto de follow-up durante a reunião. Contem os atributos de log e data em que foi criado.
Decisao	Entidade <i>decisoes</i> – Cada objeto deste tipo representa uma decisão tomada sobre um ponto de agenda. Tem como atributos o texto da decisão e a data em que foi criada.

Tabela 11 - Entidades existentes na aplicação

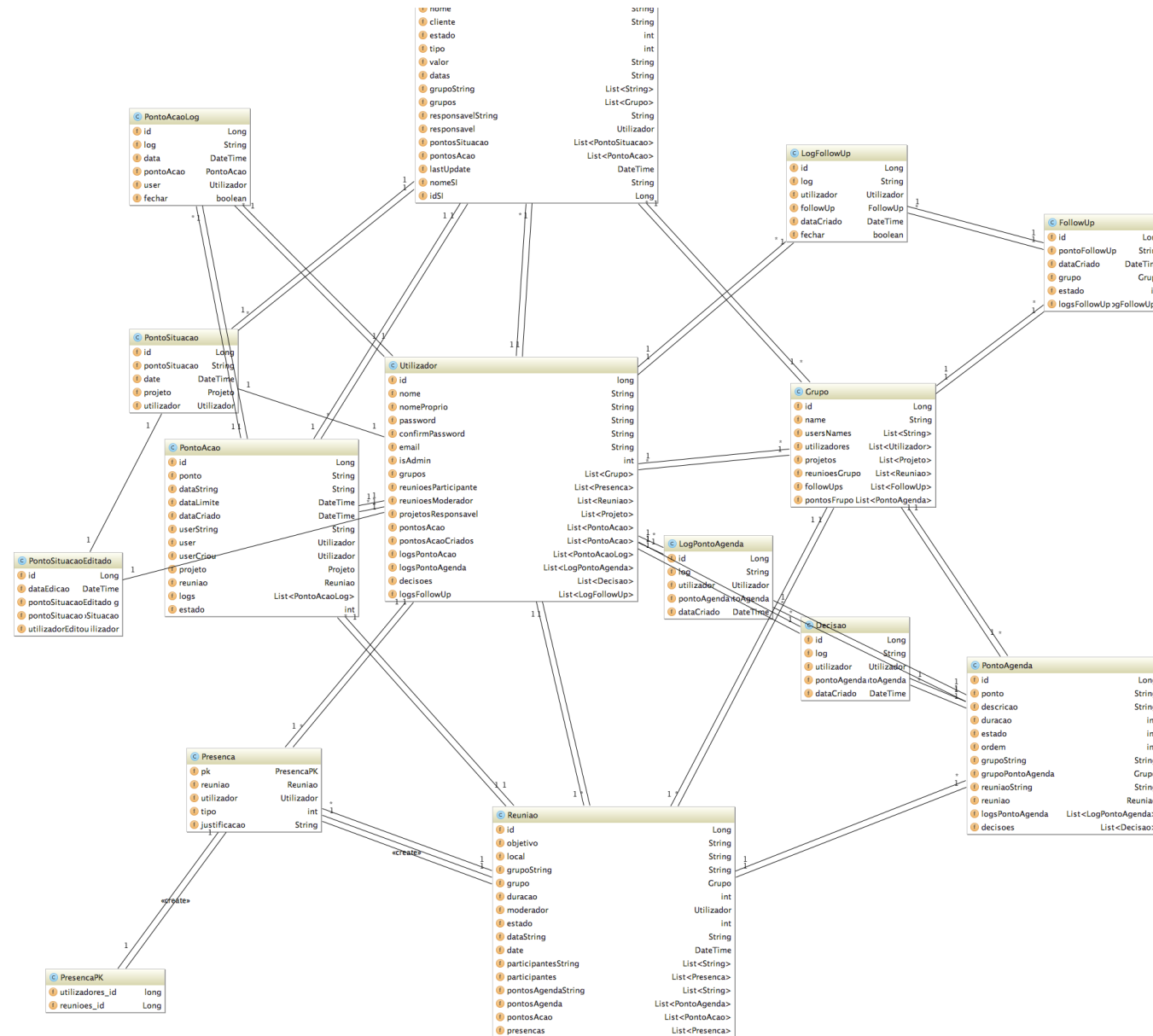
As entidades apresentadas na tabela anterior são as entidades existentes na aplicação que são guardadas em base de dados. No entanto, foram criadas outras entidades que auxiliam na realização de certos requisitos e atividades na aplicação. Estas entidades são listadas na tabela seguinte:

Modelo	Descrição
Constants	Entidade onde são guardados os valores que são constantes por toda a aplicação.
HelperStringId	Auxiliar utilizado nas chamadas AJAX para preencher <i>select box</i> nas vistas.
PontoAcaoPage	Entidade que auxilia a paginação nas listas de pontos de ação. São guardados em cada objeto os pontos de ação que cada página deve ter, e outra informação adicional tal como o número total de páginas que existe na lista ou a página atual.
PontoAgendaPage	Entidade idêntica à descrita na linha acima, mas que auxilia nas listas de pontos de agenda.
Presenca	Entidade que representa a tabela auxiliar criada devido à relação Many-To-Many entre a entidade Reuniao e a entidade Utilizador. Esta gestão é feita manualmente devido à necessidade de ter mais informação nesta entidade referente à presença/justificação de um utilizador a uma reunião.
PresencaPK	Entidade que serve como chave primária da entidade Presenca. Como a entidade presença é a representação de uma relação Many-To-Many, é necessário ter uma chave composta por duas chaves primárias.
ProjetoPage	Entidade que auxilia a paginação nas listas de projetos.
ReuniaoPage	Entidade que auxilia a paginação nas listas de reuniões.

Tabela 12 - Entidades auxiliares existentes na aplicação

Foram assim criadas um total de 21 entidades para dar suporte a todos os requisitos da aplicação. O diagrama de classes correspondente é apresentado na figura seguinte.

Figura 14 - Diagrama de Classes



5.6 Ficheiros de configuração interna

Na *play! framework* existem um conjunto de ficheiros que é necessário configurar para que a aplicação funcione e se relacione com outras frameworks. Na tabela seguinte, são apresentados estes ficheiros.

Ficheiro	Descrição
<i>application.conf</i>	Neste ficheiro existem várias configurações importantes ao sistema. São elas as linguagens que a aplicação pode apresentar ao utilizador, a base de dados a que a aplicação se liga, configurações específicas de alguns plugins, como é o caso do plugin utilizado para enviar emails, configurações de evoluções, entre outras.
<i>messages.pt-PT</i>	Ficheiro onde estão presentes as frases e termos que são mostrados na GUI ao utilizador, escritos em português, para serem utilizados no multilingue da aplicação.
<i>messages.en-EN</i>	Ficheiro onde estão presentes as frases e termos que são mostrados na GUI ao utilizador, escritos em inglês, para serem utilizados no multilingue da aplicação.
<i>routes</i>	Ficheiro onde são definidas as rotas para o encaminhamento entre os pedidos e os métodos nos controladores que os tratam.
Pasta <i>evolutions</i>	Pasta onde são guardados os scripts das evoluções das bases de dados. Mais informação sobre as evoluções será dada mais à frente no presente relatório.
<i>build.sbt</i>	Ficheiro onde são guardadas as dependências de bibliotecas externas que a aplicação utiliza.
<i>Logback.xml</i>	Ficheiro onde são introduzidos os diversos ficheiros de logs, e a especificidade de cada um. Podem assumir valores de <i>TRACE</i> , <i>DEBUG</i> , <i>INFO</i> , <i>WARN</i> e <i>ERROR</i> .
<i>plugins.sbt</i>	Ficheiro onde são introduzidos os plugins utilizados pela aplicação.

Tabela 13 - Ficheiros de configuração

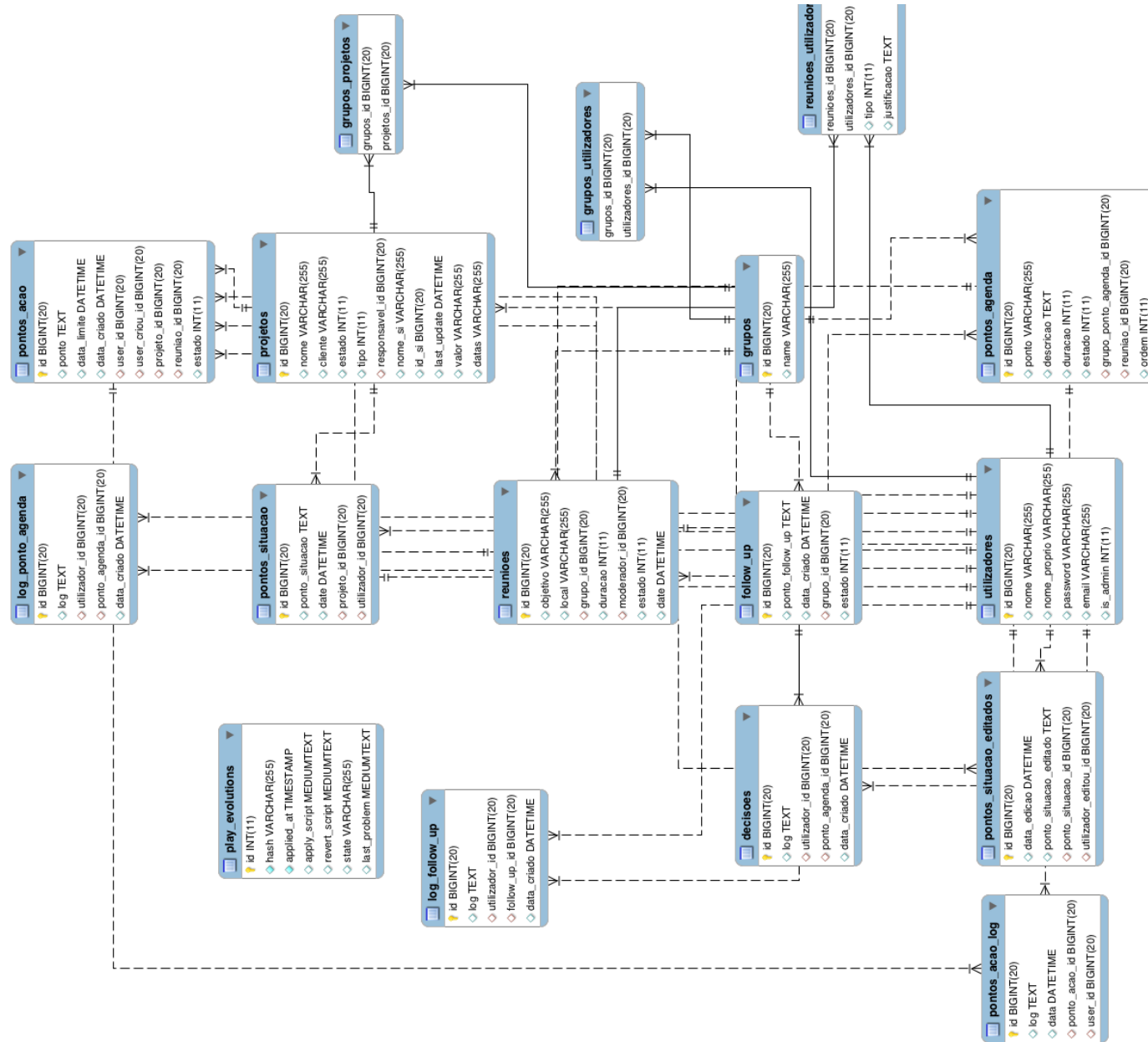
5.7 Base de dados

Durante o desenvolvimento da aplicação foi utilizado o sistema de gestão de bases de dados MySQL.

A interação entre a aplicação e as bases de dados é feita através do Play Ebean, uma classe que faz o mapeamento entre os objetos na aplicação e as entidades na base de dados. Quando um modelo é criado na aplicação são especificadas anotações, como por exemplo se representa uma entidade ou qual o nome da tabela respetiva na base de dados. Aos atributos são dadas anotações que permitam saber se o atributo é obrigatório, se é uma chave primária ou mesmo o tipo do atributo a guardar na base de dados.

O mapeamento realizado pelo play Ebean é visível na figura seguinte, que representa o modelo entidade-relacionamento(ER) da base de dados. No diagrama é possível notar que cada entidade corresponde a um modelo na aplicação. Como foi dito anteriormente, os scripts para a criação da base de dados a partir dos modelos existentes podem ser vistos na pasta *Evolutions*, dentro da diretoria *conf*.

Figura 15 - Diagrama ER da aplicação



5.8 Conclusão do capítulo

Analisando os diferentes níveis da arquitetura podemos afirmar que esta foi influenciada pela escolha da framework *play*, pois é uma framework com a arquitetura MVC.

Esta foi descrita em 3 níveis. No primeiro é dada uma visão geral da ferramenta e dos componentes que a rodeiam. No nível 2 são apresentados de forma mais detalhada os componentes da aplicação. No terceiro nível podemos ver as especificações da arquitetura MVC e como ela é utilizada pela *framework play*. Foram também apresentados os modelos da aplicação, descrevendo brevemente cada um e apresentado as relações entre modelos através do diagrama de classes.

O mapeamento destes objetos na base de dados é feito pelo *play Ebean* e o resultado deste mapeamento resulta no diagrama ER apresentado.

Foi também descrita a arquitetura utilizada para suportar o multi-tenancy da aplicação. A aplicação liga-se a uma base de dados que contém informação sobre as bases de dados de cada cliente. Com esta informação, ao arrancar, a aplicação cria um servidor Ebean com um nome específico para cada cliente, ficando este encarregado da ligação à base de dados. Quando um utilizador se liga à aplicação, escolhe qual é a sua organização e é gravado o nome do servidor Ebean dessa organização na sessão do utilizador para ser utilizada em todos os próximos pedidos à base de dados.

Por último foram abordados os ficheiros de configuração necessários para o correto funcionamento da aplicação.

Capítulo 6

Desenvolvimento da aplicação

No próximo capítulo irão ser abordados alguns aspetos relacionados com o desenvolvimento da aplicação. Num primeiro passo, serão apresentados os ambientes de desenvolvimento que existiram no desenvolvimento. De seguida irá ser descrita a forma como eram feitas as evoluções à base de dados.

No total do desenvolvimento foram produzidos 16 controladores, que continham 167 métodos. O total de linhas de código considerando os controladores, as vistas, os principais ficheiros de *Javascript* e CSS e o ficheiro de rotas foi de 16360.

6.1 Ambientes de desenvolvimento

Foram criados 3 ambientes de desenvolvimento. Cada um continha um *branch* num projeto criado no GIT do LIS-IPN, para existir controlo de versões. Os 3 ambientes eram:

- Desenvolvimento;
- Staging;
- Master.

6.1.1 Ambiente de desenvolvimento

A este ambiente de desenvolvimento correspondia o repositório Git de desenvolvimento. Este repositório era utilizado para realizar submissões praticamente diárias de código, sendo por essa razão o repositório que continha a versão mais evoluída da aplicação. Servia assim para ter um repositório do código desenvolvido salvaguardando alguma perca que pudesse ocorrer e tendo sempre uma versão estável para a qual se podia voltar caso algo corresse mal no desenvolvimento. Por ser de desenvolvimento correspondia ao computador pessoal do estagiário. Tem as seguintes características:

- Sistema Operativo OS X Yosemite versão 10.10.5;
- Git Version Control;
- Java versão 1.8;
- Play Framework versão 2.4;
- Sistema de Gestão de Bases de Dados MySQL versão 5.7.11;

6.1.2 Ambiente de Staging

A este ambiente de desenvolvimento correspondia o repositório Git de staging. Este serviu para ter um ambiente de testes. A versão de código que entrava para testes correspondia ao código desenvolvido para a próxima entrega funcional. Era adicionado à medida que eram desenvolvidos um conjunto de requisitos, para que fossem validadas as funcionalidades e a integridade do sistema. Esta validação era realizada por colaboradores do IPN-LIS, que utilizavam a aplicação e reportavam as falhas que encontravam. Para poderem reportar as falhas foi criado um link na aplicação que permitia enviar emails para o estagiário, onde os utilizadores podiam descrever a falha que tinham encontrado e os passos anteriores à falha.

Para além do email foi implementado um sistema de logs onde era possível ver quais os erros que a aplicação apresentava. Quando se detetavam erros nesta fase, estes eram corrigidos no ambiente de staging e voltava a submeter-se o código para este repositório, continuando a fase de testes. Quando o cliente, representado pelos utilizadores que testavam os requisitos, estava satisfeito com a aplicação no ambiente de staging, esta era passada para o ambiente de produção, para ser utilizada pelo cliente.

Este ambiente tem as seguintes características:

- Sistema operativo CentOS Linux release 6.8;
- Servidor Web Apache versão 2.2.15;
- Sistema de Gestão de Bases de Dados MySQL versão 5.6.30;
- Java versão 1.8;
- Git version 1.7.1;

6.1.3 Ambiente de Produção

Para o código em produção, foi criado o repositório GIT de produção. Este ambiente era o utilizado pelos utilizadores finais e continha o produto final.

Continuavam presentes neste ambiente os links para submissão de erros e anomalias, bem como o sistema de logs. Quando era detetado ou reportado um erro no ambiente de produção, era criado um repositório à parte com o código que estava em produção. O código era desenvolvido e testado nesse repositório e só depois do erro resolvido e das funcionalidades que poderiam ser afetadas testadas, é que se fazia o *deploy* para o ambiente de produção, para continuar assim a ser utilizado pelo cliente. Era utilizado este método para garantir que o que estava presente em produção eram sempre versões estáveis. Como os repositórios de desenvolvimento e staging continham código mais avançado, era importante não submeter esse código para produção sem a respetiva validação, para não apresentar problemas inesperados.

Enquanto que no ambiente de desenvolvimento e de staging é o servidor aplicacional da *framework play!* que recebe e redireciona os requests dos utilizadores, no servidor de produção existe um servidor web apache, que utiliza o protocolo HTTPS para garantir uma maior segurança nos dados. Este recebe os pedidos e redireciona-os para o servidor aplicacional da *framework play!*

Este ambiente tem as seguintes características:

- Sistema operativo CentOS Linux release 7.2.1511;
- Servidor web Apache 2.4.6;
- Sistema de Gestão de Bases de Dados MySQL versão 5.6.30;
- Java versão 1.8;
- Git versão 1.8.3.1;

Para realizar o deploy para o ambiente de produção, foi criado um script apresentado de seguida.

```
[root@meetings.ipn.pt:/www/ipnmeetings]# cat deploy.sh (master*)
service ipnmeetings stop
su tomcat -c "git pull"
su tomcat -c "./activator clean stage"
service ipnmeetings start
service httpd restart
[root@meetings.ipn.pt:/www/ipnmeetings]# (master*)
```

Script 1 - Script de deploy

É parado o serviço correspondente à aplicação e é feito o pedido *pull* do código presente no branch de produção do GIT. De seguida é iniciado o servidor aplicacional da framework *play*. O serviço é iniciado bem como o Apache. O pedido *pull* não irá dar conflitos visto que o código presente no branch de produção nunca é alterado diretamente nesse branch.

6.2 Evoluções da base de dados

Quando são efetuadas alterações nos modelos, a framework *play!* cria automaticamente as entidades em base de dados correspondentes a essas alterações. Este comportamento da framework apresentava dois problemas para o desenvolvimento da aplicação que eram:

- Apagar as tabelas e os seus conteúdos quando existem alterações;
- Aplica as alterações apenas à base de dados configurada no ficheiro *application.conf*;

Estes eram os principais problemas a resolver pois como a aplicação já se encontrava a ser utilizada pelos utilizadores finais, aquando da próxima entrega não poderíamos apagar os dados que já tinham sido inseridos e porque as alterações que aconteciam nos modelos eram referentes aos modelos de negócio e não ao modelo com as informações de cada cliente.

Como foi dito anteriormente, a única base de dados configurada no ficheiro *application.conf* é a base de dados que contém os dados sobre as BD dos clientes. Para resolver os problemas identificados era necessário encontrar uma forma que permitisse a continuidade das evoluções às bases de dados e que aplicasse as evoluções às bases de dados de cada cliente e não à BD configurada diretamente na aplicação.

Para resolver o primeiro problema identificado foi desativado o modo automático de evoluções da framework *play!* e foram criados os scripts SQL que correspondiam às alterações ou adições de modelos manualmente. Com isto, garantiu-se que quando eram feita alterações aos modelos, as tabelas em base de dados não eram apagadas mas sim evoluídas.

Para realizar as evoluções nas bases de dados de cada cliente, foram criados dois packages diferentes para os modelos. Um que continha os modelos referentes à BD com as informações de cada cliente e outro com os modelos do negócio para garantir que as evoluções não eram realizadas também na BD com as informações de cada cliente. Para além disso, no método *onStart* da framework utilizada, antes da criação dos servidores *Ebean* discutidos anteriormente, são criados objetos Database que correspondem à base de dados de cada cliente e são efetuadas as evoluções diretamente nessa base de dados.

Com a utilização de evoluções na base de dados era garantido também que não existiam incompatibilidades entre o código num ambiente de desenvolvimento e a evolução da base de dados que o código utilizava, pois cada versão do código utiliza uma evolução específica da base de dados. Assim, mesmo que fosse necessário realizar uma regressão na versão do

código em algum ambiente, era possível regressar a evolução da base de dados para a que aquela versão de código utiliza, para que não existam conflitos entre os modelos existentes na aplicação e as entidades da base de dados.

Foram desenvolvidas no total 10 evoluções da base de dados, sendo que apenas a primeira evolução correspondia ao código para a primeira entrega funcional.

6.3 Web Sockets

Como existem ecrãs na aplicação que irão ser sempre utilizados por várias pessoas ao mesmo tempo, era necessário implementar algum mecanismo que permitisse que a informação fosse atualizada em tempo real por todos os utilizadores. Para isso foram implementados web sockets em dois ecrãs diferentes: o ecrã de acompanhar projetos e o ecrã apresentado durante uma reunião.

É atribuído um web socket a cada utilizador que acede à página. Quando um utilizador realiza alguma ação, como adicionar algum item, é feito um pedido AJAX ao controlador que retorna apenas a parte do ecrã que necessita de alteração por AJAX também. Aquando do tratamento do pedido, é chamado um método referente aos web sockets no método do controlador utilizado para tratar o pedido. Consoante o tipo de pedido, se for por exemplo a adição de um ponto de agenda ou um ponto de situação, é criado dinamicamente o código HTML correspondente. Para além disso é criado um objeto de um método auxiliar com informação acerca do item que foi adicionado ou editado, como o ID. Este objeto é convertido em Json e é enviado pelo socket.

O que é enviado no socket é recebido num ficheiro Javascript independente para cada ecrã. Aqui, são feitas verificações nas DIV's do ecrã, identificadas por ID específico, composto pelo tipo de objeto e pelo ID, para garantir que o utilizador se encontra no projeto ou reuniões certas. Caso sejam verdadeiras o código HTML é adicionado no sítio correspondente do ecrã.

6.4 Conclusão do capítulo

Neste capítulo foram descritos aspetos importantes no desenvolvimento da aplicação, como os ambientes de desenvolvimentos e a evolução das bases de dados.

Existem três ambientes de desenvolvimento distintos, um de desenvolvimento, um de staging e outro de produção. Cada um destes ambientes tem um *branch* num repositório de Git associado. O *branch* de desenvolvimento corresponde ao ambiente de desenvolvimento, que consiste no computador do estagiário. O *branch* de staging, que corresponde ao ambiente de staging, é onde são testadas as entregas funcionais por um conjunto de utilizadores composto por colaboradores do IPN-LIS e por alguns futuros utilizadores da aplicação. O estagiário é informado dos erros encontrados e de sugestões de alterações nos requisitos através de um link fornecido na aplicação para envio de mail e através dos ficheiros de *logs* que registam os erros da aplicação. Este ambiente de staging está situado numa máquina nos servidores do IPN, para que seja acessível a partir do exterior. O servidor de staging pode ser acedido através do site “<http://dev.lis.ipn.pt/ipnmeetings/>”.

Para o *branch* master, que corresponde ao ambiente de produção, foram submetidas as entregas funcionais planeadas no início do semestre.

Aquando da submissão no servidor de produção é necessário ter uma versão estável pois já será utilizada por todos os utilizadores, com dados reais. Posto isto, é fundamental que a aplicação não tenha erros ou perca os dados. Foram verificados regularmente os ficheiros de log e o mail para onde são enviados os erros encontrados pelos utilizadores e problemas identificados são corrigidos num *branch* à parte, antes de voltar a submeter o código no *branch* master.

O servidor de produção é acessível através do site “<https://meetings.ipn.pt/>”.

Quanto a evoluções da base de dados, foi necessário alterar o modo *default* que a *play* utiliza para evoluir a base de dados. Foram criados scripts manuais que correspondem a cada alteração nos modelos, para que a aplicação verifique que existem evoluções a serem feitas e faça essas alterações sem apagar as tabelas e o conteúdo destas já criadas em base de dados. Para além disso foi implementado um modelo que aplica as evoluções a bases de dados externas que não estejam presentes no ficheiro de configuração.

A implementação de web sockets encontra-se dividida pelos ecrãs onde estes existem para um maior controlo dos conteúdos a serem mostrados. O código HTML é criado dinamicamente no controlador e inserido em DIV's específicas nas vistas.

Capítulo 7

Verificação e validação

Ao longo do projeto de estágio foram tomadas medidas que garantissem a verificação e validação corretas tanto do código desenvolvido, como dos produtos resultantes das diferentes partes do projeto, como o levantamento de requisitos ou a definição da arquitetura da aplicação.

No seguinte capítulo, estas medidas são identificadas, e são apresentados os dados que cada medida cobre.

7.1 Verificação

Verificação consiste no processo de avaliar os produtos resultantes de uma fase de desenvolvimento, de modo a verificar se esse produto vai de encontro aos requisitos especificados [28].

A verificação dos produtos resultantes ao longo do estágio pode ser dividida em duas fases: antes e depois de começar o desenvolvimento.

Na primeira fase era necessário verificar os documentos produzidos para garantir que estivessem de acordo com o pretendido pelo cliente. O documento mais importante a ser verificado pelo cliente nesta fase foi o documento de levantamento de requisitos, para garantir que a aplicação a desenvolver tinha todos os requisitos pretendidos por este e que ia dar resposta ao seu problema. Esta verificação foi feita através de reuniões onde o documento foi apresentado e debatido, obtendo assim um feedback acerca dos requisitos levantados pelo estagiário. Este foi verificado e aceite pelo cliente.

Após esta verificação dos requisitos foi necessário verificar a arquitetura escolhida para a aplicação, para garantir que ia ao encontro de uma das restrições de implementação propostas pelo cliente que era a arquitetura multi-tenancy da aplicação. A arquitetura foi produzida e apresentada ao cliente e mais uma vez discutida em reunião, para perceber se o cliente concordava com a arquitetura proposta, especialmente a parte de multi-tenancy. Esta era a parte mais importante para o cliente ao nível da arquitetura, pois permite-lhe no futuro estender a aplicação a outros laboratórios do IPN ou vender a outras organizações como SaaS. Foi feita a verificação e aceite pelo cliente o modo como a aplicação iria ser implementada. Esta verificação foi feita com o cliente, também devido ao *know-how* deste na definição de arquiteturas de aplicações semelhantes.

Durante o desenvolvimento era necessário verificar que as releases que eram entregues para validação e testes continham os requisitos que era suposto estarem em cada uma. Esta verificação era realizada comparando o documento de requisitos com o que era desenvolvido. Para além disso era feita uma verificação ao código por parte do estagiário, identificando alguma falha ou partes de um requisito em falta.

Outra verificação feita era para identificar as autorizações para cada action. Como cada action é permitida apenas a um determinado grupo de utilizadores tendo em conta o seu perfil, foi necessário verificar que o acesso à informação estava protegido e que a autorização era apenas garantida aos utilizadores certos. Para esta verificação era analisado o código de cada action à procura de eventuais falhas nas autorizações e era efetuada a correção das mesmas.

Como exemplo, a página de detalhes de uma reunião apenas pode ser vista pelo administrador, por utilizadores que pertençam ao grupo da reunião ou por utilizadores que não pertencendo ao grupo da reunião são participantes apenas dessa reunião. Por isso, era necessário implementar em cada action as autorizações corretas a cada. Esta verificação foi feita pelo estagiário, tendo sido auxiliado pelo Eng. Carlos Lopes.

7.2 Validação

Validação prende-se com a certificação de que o código produzido vai ao encontro das expectativas que o cliente tem para a aplicação e que cumpre os requisitos definidos. Foram utilizadas diferentes abordagens que garantiam a validação de diferentes componentes da aplicação, entre os quais:

- Modelos de dados;
- Requisitos funcionais;
- Informação visualizada pelo utilizador;
- Ambiente gráfico e os seus componentes;

A primeira validação prende-se com a integridade dos dados que resultam dos pedidos à base de dados. É necessário validar esta informação para nos assegurarmos que os dados acedidos são os que são pretendidos e que os pedidos à base de dados não estão a obter informação errada. Para esta validação, aquando da criação de novos modelos, eram inseridos alguns dados na base de dados através de scripts na aplicação e de seguida eram feitos pedidos à base de dados e imprimidos os dados. Após a validação dos dados obtidos, o desenvolvimento com esse modelo podia continuar pois estava assegurado que todos os atributos do modelo eram acedidos. Com este tipo de validação, foi coberto todo o código que envolvia pedidos à base de dados dos modelos e o código dos modelos também.

Outra validação efetuada permitiu validar o funcionamento dos requisitos funcionais esperados em cada entrega. Para isto, um grupo de utilizadores sabendo os requisitos que cada entrega devia ter realizou *exploratory testing*, de modo a garantir que todos os requisitos estavam presentes e que funcionavam de acordo com o especificado no levantamento de requisitos. Este método de testes permite um rápido feedback ao estagiário dos erros encontrados[29], o que permitia que aquando da entrega de uma release os testes fossem feitos e os resultados chegassem rapidamente ao estagiário, para este resolver os erros encontrados e submeter rapidamente outra versão para testes. Este ciclo de correção de bugs/erros e nova fase de testes repetia-se em cada entrega funcional, até que o código fosse validado pelo cliente, sendo de seguida feito o *deployment* nos servidores de produção para ser utilizado pelos utilizadores finais. Esta validação servia assim para validar todos os métodos nos controladores, visto que cada método implementa um requisito ou um conjunto de requisitos. Assim, se os requisitos fossem validados, o código correspondente era validado.

Enquanto esta validação através de *exploratory testing* era efetuada, a informação apresentada aos utilizadores era validada também. Aqui, esperava-se que a informação apresentada na aplicação fosse a correta e que fosse apresentada da forma correta.

Faltava assim a validação ao ambiente gráfico e ao comportamento de todos os seus componentes. Esta validação foi feita pelo designer gráfico do LIS-IPN, que acompanhou o

design de toda a aplicação. Este anotava todas as inconformidades entre a aplicação e o design previsto e aceite pelo cliente, para que o estagiário alterasse o design da aplicação. Com isto, era validado o código referentes às vistas.

Para complementar a validação dos requisitos, as funcionalidades entravam em produção primeiro para um grupo de utilizadores piloto para existir nova confirmação e só de seguida é que era disponibilizado a todos os utilizadores finais.

7.3 Testes de usabilidade

Para que os testes de usabilidade fossem o mais fidedignos possível era necessário procurar uma ferramenta onde fossem criados os mockups já elaborados e fazer com que fosse possível navegar entre os ecrãs. Para isso, foi utilizada a aplicação web Weld[20].

7.3.1 Configuração dos testes

Usabilidade é um atributo de qualidade. Esta verifica o quão fácil uma interface é de usar. É essencialmente definida por 6 componentes de qualidade que são[21]:

- Apreensão de conhecimento: como é fácil para os utilizadores realizarem tarefas básicas na primeira vez que se deparam com o design;
- Eficiência: rapidez com que conseguem realizar as tarefas, a partir do momento em que os utilizadores conhecem o design;
- Memorização: quando os utilizadores regressam ao design passado algum tempo, quanto tempo demoram a realizar as tarefas com eficiência;
- Erros: quantos erros os utilizadores cometem, e como conseguem recuperar desses erros;
- Satisfação: quão satisfatório e intuitivo é o design;
- Utilidade: a interface faz o que os utilizadores precisam;

Destes 6 componentes de qualidade, os que pretendemos avaliar com os nossos testes são a apreensão de conhecimento, eficiência e satisfação.

Para avaliar estes componentes, foi necessário definir um conjunto de métricas a retirar. Segundo Jakob Nielsen [21], as métricas a usar prendem-se com a definição de usabilidade. Para avaliar os componentes descritos acima podiam ser utilizadas a taxa de sucesso de cada tarefa, o tempo que uma tarefa necessita para ser completada, a taxa de erro ou a satisfação subjetiva do utilizador. Tendo em conta a fonte citada, a métrica utilizada foi o tempo que os utilizadores demoram a realizar cada tarefa. Para além disso foram anotados os comentários realizados pelo utilizador para se obter uma avaliação subjetiva.

Para tirar valores destas métricas foi necessário elaborar um guião com uma sequência de ações para os utilizadores realizarem. Este guião começava com uma introdução ao projeto para contextualizar os utilizadores e uma explicação do que era pretendido.

Para definir as ações, escolheram-se as mais importantes a desempenhar na aplicação e que cobriam a maior parte dos ecrãs. Deste modo, era possível avaliar a usabilidade de todo o sistema.

As ações escolhidas foram:

Ação 1

- 1) Criar uma nova reunião e adicionar um ponto de agenda a essa reunião;
- 2) Editar a reunião “Reunião de Direção 3/2/2016”
- 3) Editar um ponto de Agenda já criado, da reunião “Reunião Direção 3/2/2016”;
- 4) Confirmar a reunião “Reunião Direção 3/2/2016”;
- 5) Iniciar a reunião “Reunião Direção 3/2/2016”, e de seguida;
 - a. Justificar a ausência do utilizador Carlos Lopes;
 - b. Escrever e guardar a nota “Carlos Lopes vai comprar na próxima semana”, referente ao ponto de agenda “Compra de máquina de café”;
 - c. Adicionar ponto de agenda durante a reunião;
 - d. Adicionar ponto de ação a um utilizador, no ponto de agenda “Participação em evento”;
 - e. Adicionar um ponto de follow-up;
 - f. Adicionar um novo projeto;
 - g. Ainda durante a reunião, deve aceder ao projeto Sistema de Gestão de Reuniões e realizar os passos necessários para adicionar um ponto de situação ao projeto;
- 6) Terminar a reunião, para que a ata seja enviada automaticamente para todos os utilizadores.

Ação 2

- 1) Adicionar um novo ponto de situação ao projeto “Sistema Gestão de Reuniões”;
- 2) Adicionar um ponto de ação ao projeto “Sistema Gestão de Reuniões”;
- 3) Ver todos os pontos de ação do utilizador;
- 4) Ver toda a informação do grupo “Reuniões Direção”;

No final foi colocado um espaço para comentários do utilizador e aspetos positivos ou negativos da aplicação. Este espaço tinha por objetivo recolher dados para a avaliação subjetiva da usabilidade.

Quanto ao número de utilizadores a testar, segundo Jakob Nielsen,[22] deve ser à volta de 5, sendo este o número de utilizadores escolhido. Esta escolha é explicada tendo em conta uma distribuição de pareto, que se verifica em testes de usabilidade. Seguindo esta distribuição, notamos que 80% dos problemas é detetado por apenas 20% dos utilizadores, não se justificando assim um elevado número de testes.

Para uma realização de testes mais eficaz, foram utilizados dois perfis de utilizadores distintos. O primeiro perfil é o de alguém que é participante das reuniões de direção semanais, ou seja, alguém que já tem algum conhecimento de como as reuniões funcionam. O outro perfil é o de pessoas que não estão presentes nas reuniões de direção e que não têm tanto

conhecimento acerca destas reuniões. Esta diferença permite perceber qual a diferença entre estes tipos de utilizadores na utilização da aplicação.

Já durante o teste de usabilidade era necessário optar por uma técnica de moderação. Esta devia permitir retirar as métricas identificadas, que como foi dito serão o tempo que cada tarefa demora a ser completada e a opinião subjetiva dos utilizadores através de comentários. Esta escolha da técnica a utilizar, como referiu Jen Romano Bergstrom [23], depende dos objetivos da sessão e do que pretendemos retirar dela. Tendo isto em conta, foi escolhida a técnica de *Retrospective Probing*. Nesta técnica o utilizador necessita de terminar o teste para depois pedirmos a sua opinião. Deste modo são retiradas notas durante a sessão do que o utilizador refere para comentar no final com o utilizador, não interferindo assim com a métrica de tempo que pretendemos retirar.

7.3.2 Resultado dos testes de usabilidade

Durante a realização dos testes os tempos que os utilizadores demoraram foram anotados em tabela. Foram anotados também, onde foi necessário, alguns comentários do utilizador ou algum fator que fez com que alguma ação demorasse mais tempo. Os resultados podem ser vistos na tabela 14. De salientar que a unidade de medida das células da tabela é o segundo.

	Utilizador 1	Utilizador 2	Utilizador 3	Utilizador 4	Utilizador 5	Média
Ação 1						
1	11	5	4	15	9	8,8
2	7	20 ^{*1}	16	6	4	10,6
3	20	14	35	8	70 ^{*1}	29,4
4	38 ^{*1}	9	3	11	13	14,8
5	40	27 ^{*2}	2	2	9	16
5.a	11	5	14	9	1	8
5.b	65 ^{*2}	10	10	6	12	20,6
5.c	15	18 ^{*3}	8	10	4	11
5.d	2	12 ^{*4}	15	10	10	9,8
5.e	2	2	13 ^{*1}	1	2	4
5.f	2	2	2	1	2	1,8
5.g	5	9	10	18	10	10,4
6	4	3	3	2	10 ^{*2}	4,4
Ação 2						
1	14	3	5	8	6	7,2
2	5	2	1	7 ^{*1}	3 ^{*3}	3,6
3	20	25	22	2	7	15,2
4	12	10	6	3	4	7
Total	273	176	169	119	176	182,6

Tabela 14 - Resultados testes de usabilidade(segundos)

Após os testes e antes de retirar conclusões sobre o que melhorar, é necessário perceber alguns comentários dos utilizadores ou fatores que influenciaram o teste. Para isto, foram analisados esses comentários e os fatores que explicam um elevado tempo em algumas ações, assinaladas na tabela com o símbolo “*”, ajustando assim os resultados dos testes. Estas observações podem ser vistas em pormenor no documento Ref_2, enviado em anexo.

7.3.3 Análise aos resultados dos testes de usabilidade

Após estas observações aos comentários, notou-se que existiam alguns problemas a resolver. Para além disso, foi encontrada explicação para alguns dos elevados tempos em algumas ações e mitigado o seu problema.

Analisando as médias de tempo de cada ação, verifica-se que uma das ações onde os utilizadores têm mais dificuldade é na perceção de que o nome dos itens funciona como botão para ver os detalhes dos mesmos. Esta ação irá ser melhorada, atribuindo mais espaço a cada elemento e trabalhando com elementos de *hover* nas linhas da tabela para que seja mais fácil perceber que cada um pode ser clicado.

Outro ponto onde foi encontrada alguma dificuldade foi na identificação do significado dos ícones, que pode ser vista na figura 16.



Minhas Próximas Reuniões				+ Adicionar
Nome Reunião	Grupo	Data e Hora	Local	
Reunião Direção 3/2/2016	Reunião Direção	16/02/2016 - 9:00 H	Sala Eng. Paulo Nunes	📄 ▶ 🗑️ ☑️
Potenciais Novos Clientes	Inovwine	20/02/2016 - 9:00 H	Sala Eng. Lopes Paulo	📄 ▶ 🗑️ ☑️

[Ver todos](#)

Figura 16 - Ícones da aplicação

Para colmatar esta falha os ícones irão ser melhorados pelo designer do IPN-LIS Nuno Eufrásio para que sejam mais elucidativos do seu significado. Para além disso serão incluídos efeitos quando o utilizador coloca o rato sobre o ícone, para que apareça uma legenda com o significado.

Após esta análise dos comentários e tempos deve ser analisada a opinião dos utilizadores. Na generalidade, estes afirmaram que os ecrãs eram intuitivos e eram bastante fáceis de usar. Outra opinião que foi dada, foi a alteração de alguns pontos no ecrã apresentado durante uma reunião. Após este comentário, foi mostrado aos utilizadores o ecrã preparado pelo designer do IPN-LIS, figura 17, a partir dos mockups em que os utilizadores realizaram os testes de usabilidade.

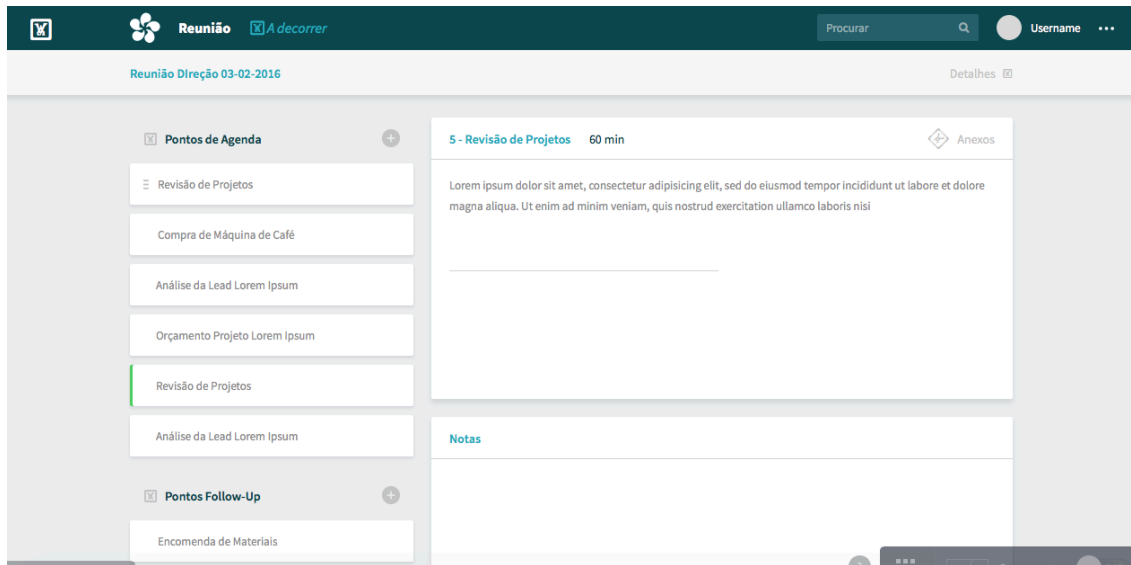


Figura 17 - Ecrã durante reunião com design aplicado

Com a visualização deste ecrã os utilizadores afirmaram que os problemas que tinham sentido iriam ser resolvidos e que a usabilidade iria aumentar.

Como podemos verificar, neste ecrã é mais fácil identificar os diversos pontos de agenda, follow-up ou projetos, e a informação acerca de cada reunião também fica colapsada, não ocupando espaço no ecrã.

7.4 Conclusão do capítulo

A verificação e validação do código garantem que a aplicação contém o que o cliente pretende e que os requisitos estão a funcionar corretamente.

Os diferentes produtos, resultantes do processo de desenvolvimento de software, foram verificados com o cliente em reuniões, tendo sido aprovados por este antes do início do desenvolvimento. Aquando do desenvolvimento, as várias releases foram também verificadas para perceber se continham os requisitos necessários para cada entrega e se o código cobria esses requisitos. Para além disso, foi verificado o código referente à autorização dos utilizadores nas diferentes vistas.

Para validar o código foi realizado *exploratory testing*. Estes testes tinham o intuito de verificar se os requisitos funcionavam de maneira correta. Caso isso não se verificasse os erros eram reportados ao estagiário que os corrigia e voltava a submeter uma nova versão para testes. Quando esta versão era validada pelo cliente e utilizadores que testavam era posta em produção para os utilizadores finais usarem.

Após os testes de usabilidade foi possível retirar algumas conclusões acerca da usabilidade da aplicação.

Quanto ao critério de apreensão de conhecimento, pode concluir-se que foi cumprido pois os utilizadores demoraram pouco tempo para concluir um número elevado de ações.

A eficiência da aplicação é elevada também, apesar de os utilizadores terem encontrado algumas dificuldades no início da utilização ou no ecrã durante a reunião. Podemos concluir isto pois, em ações parecidas umas com as outras, como por exemplo adicionar pontos de agenda, follow-up ou projetos durante a reunião, mesmo que a primeira tivesse demorado algum tempo, as seguintes foram óbvias para a maior parte dos utilizadores.

A satisfação dos utilizadores foi elevada pois muitos comentaram que a aplicação estava intuitiva e que para a primeira utilização não tinham sentido muitas dificuldades na execução de tarefas.

Quanto a pontos a necessitarem de ser alterados, foram identificados os ícones, para que se tornem mais elucidativos da ação que representam e melhorar a forma como os utilizadores acedem aos detalhes de cada reunião.

Capítulo 8

Ecrãs da aplicação

Este capítulo tem como objetivo apresentar os mockups da aplicação desenvolvidos e que foram submetidos a testes de usabilidade e, de seguida, apresentar um conjunto de ecrãs finais da aplicação que representem o aspeto gráfico da interface e as funcionalidades da aplicação desenvolvida. A aplicação é acedida através de um browser e os seus ecrãs foram desenvolvidos utilizando as tecnologias *Play Scala template*, que utiliza HTML com código Scala embutido, CSS e JS. Foram desenvolvidos um total de 48 ecrãs diferentes, contabilizando as páginas totais e parciais que são incluídos dentro dos ecrãs.

Os ecrãs tiveram como base os mockups desenvolvidos e que foram submetidos a testes de usabilidade. Com a utilização das releases funcionais por parte dos utilizadores finais, foram identificadas algumas alterações a fazer que colmatavam algumas falhas na utilização.

8.1 Construção de Mockups

A realização de mockups numa fase inicial do projeto, ainda sem a definição do modelo de dados, foi importante na clarificação de todos os requisitos da aplicação, ajudando na definição de requisitos e na sua interligação. Foi bastante importante também no esclarecimento dos atributos de cada modelo.

Para além disso, o único requisito não funcional pedido pelo cliente e que tinha uma grande importância para este era a elevada usabilidade da aplicação. Devido a esse requisito era importante guardar um grande espaço de tempo para a elaboração de mockups, para que no final do processo estes fossem ao encontro do que o cliente pretendia. Para além disso, uma definição clara da *Graphical User Interface*(GUI) antes do desenvolvimento permitia poupar tempo depois, pois não era necessário alterar e pensar nos ecrãs durante essa fase, excetuando pequenos pormenores que fossem detetados ao longo da implementação ou após ocorrerem as entregas funcionais.

Os mockups apresentados foram evoluindo progressivamente e basearam-se no feedback do cliente, do Eng. Carlos Lopes, orientador do estagiário na empresa e com a ajuda do designer do IPN-LIS, Nuno Eufrásio. Tentou adaptar-se também os pontos fortes de design das aplicações analisadas no estudo do estado da arte.

Para a construção dos mockups foi utilizada a ferramenta *Balsamiq Mockups 3*[19].

Ao longo da sua evolução, os mockups passaram por 3 principais ciclos de desenvolvimento. Ao fim de cada ciclo acontecia uma reunião com o cliente para que este desse o seu feedback aos mockups elaborados, permitindo assim ao estagiário evoluir os mockups até à próxima reunião. Para perceber a evolução no design em cada fase de desenvolvimento são apresentados de seguida os painéis principais elaborados em cada uma das fases.

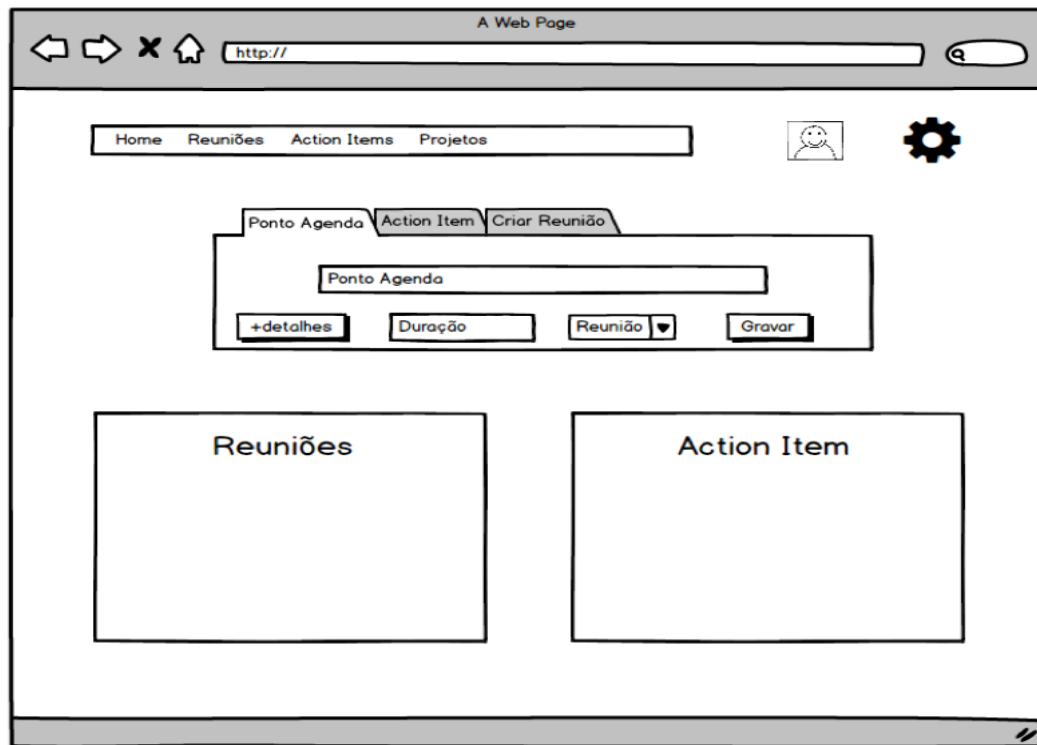


Figura 18 - Painel principal primeiro ciclo de desenvolvimento

Neste primeiro ecrã principal, figura 18, está presente uma forte inspiração nas aplicações testadas, não só a nível de funcionalidades que é permitido realizar na página mas principalmente ao nível do design.

De notar que foram tidos em conta vários requisitos do módulo de interface, como o RF6.2 e RF6.6, pois temos no painel principal espaço para ver as atividades que o utilizador tem, como por exemplo reuniões e pontos de ação, mas pode também inserir novos dados a partir deste ecrã. Foram realizados mockups que cobriam todos os requisitos, e de seguida foram debatidos em reunião com o cliente para saber o seu feedback. Após isso, iniciou-se outro ciclo de desenvolvimento que deu origem ao painel principal na figura 19.

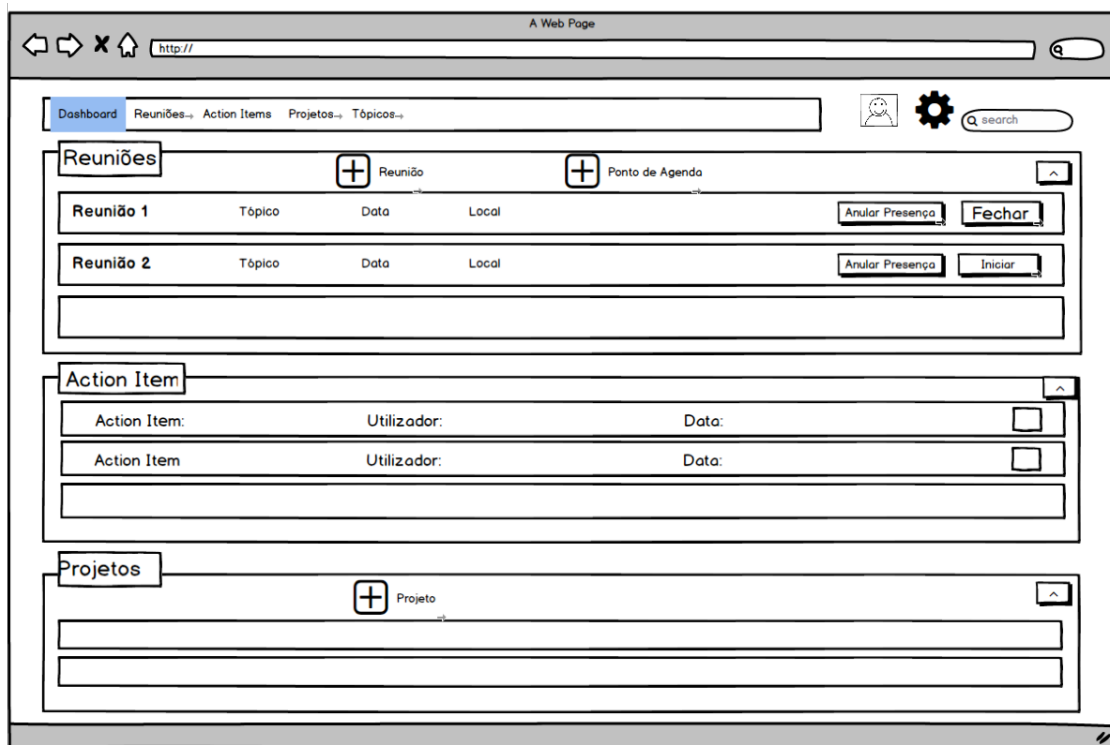


Figura 19 - Painel principal segundo ciclo de desenvolvimento

Nesta segunda fase notou-se uma maior influência das ideias do cliente no design da aplicação. Ou seja, o design representado na figura anterior perdeu a grande influência de design das aplicações analisadas, focando-se mais nas funcionalidades que são apresentadas no ecrã inicial e que iam ao encontro de requisitos do utilizador. Aqui podemos verificar uma interface mais organizada e com mais informação, mas que apresenta à mesma todas as funcionalidades que a versão anterior apresentava. Na reunião correspondente a esta fase de desenvolvimento, o feedback do cliente focou-se em pedir para que a organização da página ficasse idêntica à apresentada, mas que tivesse um aspeto ainda mais limpo, sem se notarem tantas caixas a envolver os diversos items como são apresentados aqui.

Mais uma vez existiu outro ciclo de desenvolvimento, sendo o resultado apresentado na figura seguinte.

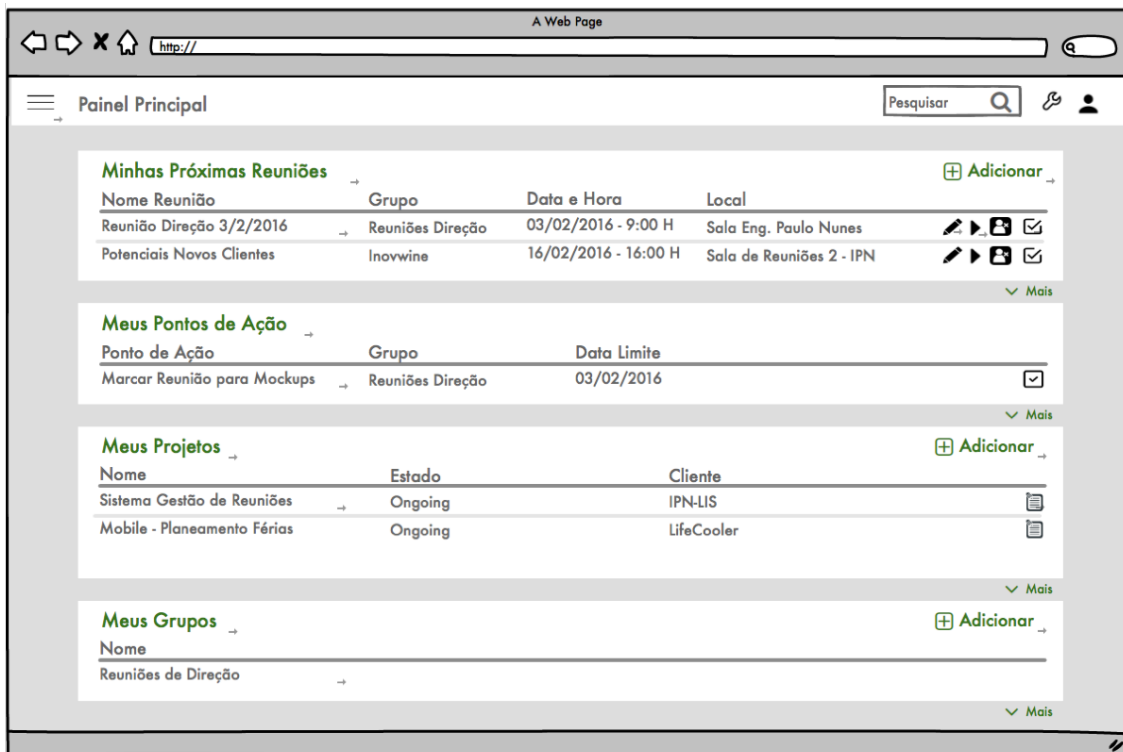


Figura 20 - Painel principal terceiro ciclo de desenvolvimento

Como estávamos perto de encontrar um design que fosse aceite pelo cliente, foi adicionada cor e mais alguns elementos que ajudassem no design, para que a vista fosse o mais parecida possível com o real. Esta foi assim a última fase de desenvolvimento de mockups, cujo cliente aceitou e que deu por isso permissão a que fossem realizados os testes de usabilidade à aplicação. Para esta última versão que foi apresentada ao cliente, foram desenvolvidos um total de 28 mockups, para que fosse possível cobrir todos os requisitos em todos os ecrãs.

8.2 Ecrãs da Aplicação

Na figura seguinte, é apresentado o ecrã que permite realizar a autenticação na aplicação.

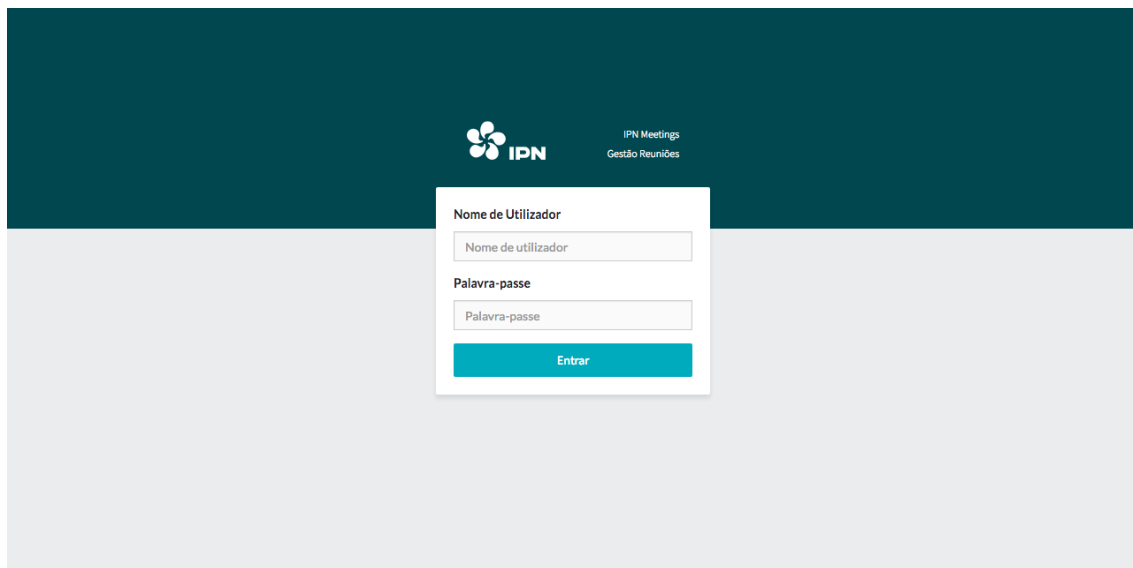


Figura 21 - Ecrã "Autenticação"

Quando as credenciais introduzidas pelo utilizador neste ecrã são corretas, é redirecionado para a página principal da aplicação. Caso um utilizador tente aceder a algum ecrã mas a sua sessão tenha expirado, é redirecionado para este e de seguida é apresentado o ecrã que tentava aceder.

Na figura 22, é apresentada a página principal da aplicação. A página apresenta a informação sobre as futuras reuniões, os pontos de ação que o utilizador tem em aberto, os projetos ordenados pela data em que tiveram a última alteração e os grupos do utilizador. É possível criar também reuniões, pontos de agenda, projetos ou grupos a partir deste ecrã.

Minhas Reuniões [ver todas](#) +

Nome Reunião	Grupo	Data e Hora	Local	
Reunião Direção #1	Grupo Direção	2016-07-01 às 17:00	DEI	✎ ✓

Meus Pontos de Ação [ver todos](#)

Não existem pontos de ação

Meus Projetos [ver todos](#) +

Projeto	Estado	Cliente	Último Update	
Estágio - Sistema de Gestão de Reuniões	Ongoing	LIS-IPN	2016-06-19 às 00:02	✎

Meus Grupos [ver todos](#) +

Nome Grupo	
Grupo Direção	✎ ✕

Figura 22 – Página Principal Aplicação

São limitados o número de dados que podem ser visíveis, para que o ecrã não fique muito extenso e a informação não seja muita.

Na figura 23, é apresentado o ecrã que permite criar uma nova reunião, sendo representativo dos restantes de criação e edição dos diferentes items.

The screenshot shows a web interface for adding a meeting. At the top, there is a dark green header with the IPNMeetings logo, a user profile for 'johnDoe', and a 'Sugestões e Anomalias' button. Below the header, the page title is 'Adicionar Reunião'. The main content area contains a form with the following fields:

- Grupo:** A dropdown menu with 'Grupo Direção' selected.
- Nome da Reunião:** A text input field containing 'Reunião Direção #1'.
- Participantes:** A list of participants: 'andrefbmaximino', 'clopes', 'neufrasio', and 'johnDoe', each with a close button (x).
- Data e Hora:** A date and time picker showing '1/7/2016 17:00'.
- Local:** A text input field containing 'DEI'.
- Duração (min):** A text input field containing '90'.

At the bottom of the form, there are two buttons: 'Guardar' (Save) and 'Cancelar' (Cancel).

Figura 23 - Ecrã "Adicionar Reunião"

Os participantes são carregados por AJAX de acordo com o grupo escolhido, para que o que o utilizador insere seja coerente.

Na figura seguinte é apresentado o ecrã de informações de uma reunião. Aqui podem ser vistos os pontos de agenda dessa reunião se já tiver sido confirmada, ou todos os que pertencem ao mesmo grupo da reunião caso contrário. É possível também definir a ausência de um utilizador à reunião respetiva.

The screenshot displays the IPNMeetings interface for a meeting titled "Reunião Direção #1". The header includes the application logo, name, and user profile (JohnDoe). The meeting details are as follows:

Grupo	Data e Hora	Local	Duração (min)
Grupo Direção	2016-07-01 às 17:00	DEI	90

Below the details, there are three sections:

- Pontos de Agenda:** A table showing agenda items. One item is visible: "Discussão do contrato de novos colaboradores" with a duration of 10 minutes. A note states "Ainda estão visíveis todos os pontos de agenda associados ao grupo".
- Participantes:** A table listing attendees and their status.

Utilizador	Presença	Função
André Maximino	Presente	
Carlos Lopes	Presente	
Nuno Eufrásio	Presente	
John Doe	Presente	Moderador
- Pontos Follow-Up:** A table for follow-up points with columns for "Ponto Follow-Up", "Data de Criação", and "Último Update".

Figura 24 - Ecrã "Informação de uma Reunião"

Na figura 25, é apresentado o ecrã onde é possível confirmar uma reunião. Os dados da reunião, como a data e hora, duração ou local são verificados e editados, se necessário. Para além disso, é neste ecrã que o utilizador escolhe os pontos de agenda que pretende levar a reunião. São apresentados todos os pontos de agenda do grupo que ainda não foram discutidos.

The screenshot shows the 'Confirmar Reunião' screen for a meeting titled 'Reunião Direção #1'. The interface includes a header with the IPNMeetings logo, user name 'johnDoe', and a notification icon. The main content area is divided into three sections:

- Dados Gerais:** Contains fields for 'Grupo' (set to 'Grupo Direção'), 'Nome da Reunião' (set to 'Reunião Direção #1'), 'Data e Hora' (set to '1/7/2016 17:00'), 'Duração (min)' (set to '90'), and 'Local' (set to 'DEI').
- Participantes:** A list of participants including 'andrefbmaximino', 'clopes', 'neufRASIO', and 'johnDoe'.
- Pontos de Agenda:** A table with two columns: 'Ponto de Agenda' and 'Duração (min)'. It lists two items: 'Discussão do contrato de novos colaboradores' (10 min) and 'Discussão de Projetos' (20 min). The first item is selected with a checked checkbox.

At the bottom, there are two buttons: 'Continuar' (highlighted in teal) and 'Cancelar'.

Figura 25 - Ecrã "Confirmar Reunião"

Após este primeiro passo que permite escolher os pontos de agenda, é apresentado um novo ecrã que permite ordenar os pontos de agenda escolhidos.

Após uma reunião estar confirmada, é possível iniciar uma reunião caso a hora atual esteja entre 15 minutos antes e 60 minutos depois da hora marcada para a reunião. O ecrã seguinte é o utilizado durante uma reunião.

Aqui é possível adicionar novos pontos de agenda, reordenar os pontos de agenda ou então adicionar notas, decisões ou pontos de ação a um ponto de agenda. Para além disso é possível adicionar novas notas aos pontos de follow-up, ou fechá-los quando necessário.

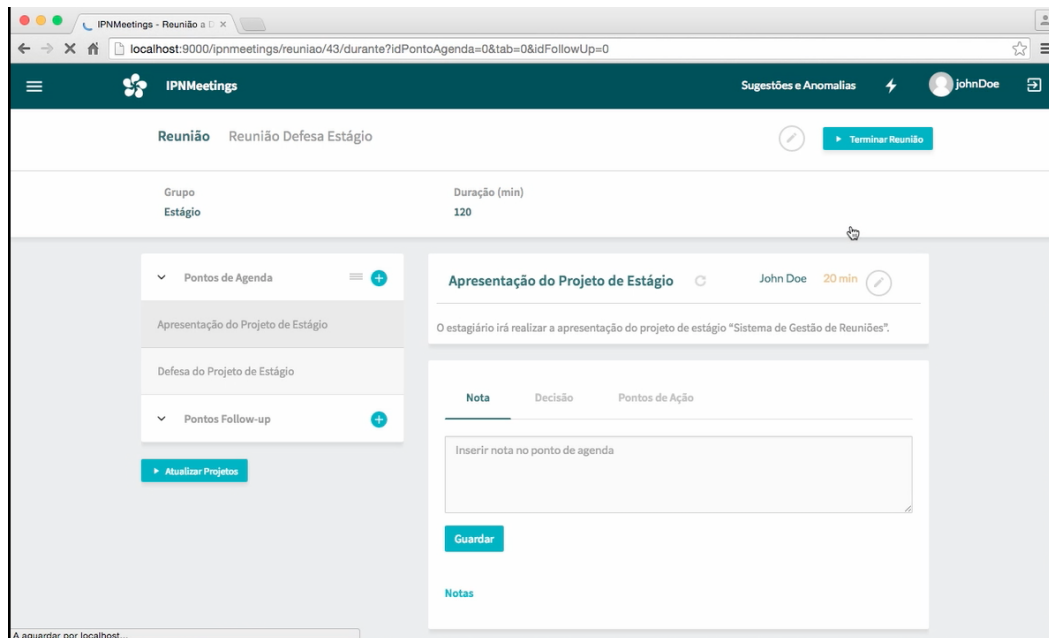


Figura 26 - Ecrã "Durante Reunião"

A informação do ecrã é toda carregada por AJAX. Como exemplo, a informação de cada ponto de agenda ou follow-up quando o utilizador clica num deles na listagem à esquerda. A submissão de dados, como notas ou decisões dos pontos de agenda ou pontos de ação é feita através de chamadas AJAX também. Isto permite que o ecrã se não altere os pontos selecionados ou a posição da listagem, mesmo quando são adicionados novos dados.

Como o ecrã pode ser utilizado por vários utilizadores ao mesmo tempo, a informação introduzida por um utilizador, como novos pontos de agenda ou notas a um ponto de agenda, são adicionados ao ecrã de todos os utilizadores através de *web sockets*. Isto permite que todos os utilizadores se mantenham a par de toda a informação.

Para permitir um acesso facilitado a qualquer reunião de um utilizador criou-se um ecrã que contém a listagem de todas as reuniões, e disponibiliza alguns filtros e pesquisa. Esse ecrã pode ser visto na figura seguinte.

The screenshot displays the 'Lista de Reuniões' (Meeting List) interface. At the top, there is a navigation bar with a hamburger menu, the IPNMeetings logo, and user information (Sugestões e Anomalias, admin). Below the header, the main content area shows a list of meetings. The list is currently filtered to 'Minhas' (My) meetings. The table has the following columns: Nome Reunião, Grupo, Data e Hora, and Local. Each row represents a meeting with its details and status icons (edit and confirm).

Nome Reunião	Grupo	Data e Hora	Local	Status
teste	Direção	2016-05-17 às 00:00	iap	✓
Reunião Estágio Editada	Direção	2016-05-21 às 00:00	IPN	✓
adosa	Direção	2016-05-21 às 00:00	IP	✓
asdadsadafaf	Direção	2016-05-21 às 00:00	asd	✓
verificar flash grupo info editado	Direção	2016-05-23 às 00:00	IPN	✎
Combinar ida à staples	Grupo 2	2016-05-27 às 00:00	IPN	✓
dhusaishduhaiu	Grupo 2	2016-05-28 às 00:00	IPN	✓
gsdauggsgduagu	Direção	2016-05-28 às 00:00	IPN	✓
PPL	Direção	2016-05-28 às 00:00	ipn	✓
asdhiajkugsdgagdgsuvgakg	Direção	2016-05-28 às 00:00	IPN	✓

At the bottom of the table, there is a pagination control showing '< anterior 01 02 seguinte >' and '20 registos'.

Figura 27 - Ecrã "Listagem de Reuniões"

Esta listagem apresenta paginação com 10 reuniões em cada página da pesquisa, e contém um código de cores para identificar facilmente que reuniões já foram confirmadas. Para além disso, apresenta as reuniões concluídas num separador diferente, e podem ser pesquisadas as reuniões em que o utilizador foi participante (Minhas), ou todas as que pertencem aos grupos onde está inserido (Todas).

Este ecrã é também uma representação dos ecrãs das listagens de projetos, pontos de ação e grupos. Foram todos realizados do mesmo modo para manter a consistência na aparência e utilização da aplicação.

Na figura seguinte, é apresentado um dos ecrãs mais importantes para os gestores de projetos, o ecrã de acompanhar projetos. Este permite ter uma listagem de todos os projetos de um determinado grupo a que o utilizador pertence e ter acesso fácil à sua informação e à possibilidade de adicionar nova informação.

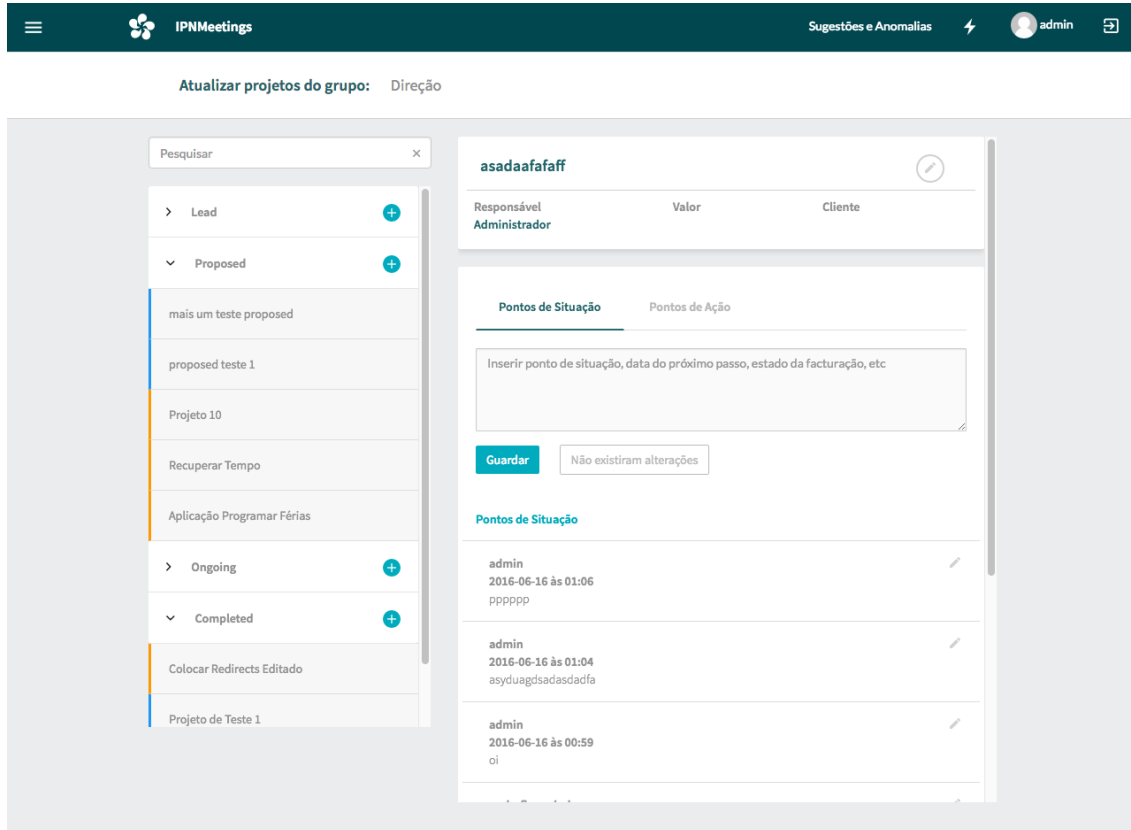


Figura 28 - Ecrã "Acompanhar Projetos"

Tal como no ecrã onde é possível acompanhar uma reunião, toda a informação é carregada e submetida através de chamadas AJAX. Este é também um ecrã utilizado por diversos utilizadores ao mesmo tempo, por ser o ecrã utilizado durante as reuniões de direção do LIS-IPN para realizar o acompanhamento dos projetos. Sendo assim, toda a informação introduzida por um utilizador é adicionada no ecrã de todos os utilizadores em tempo real através de *web sockets*.

Outra vista bastante importante é a que resume toda a informação de um determinado grupo. Esse ecrã é apresentado na figura 29. A informação encontra-se separada por separadores, para que não se sobrecarregue um ecrã com tanta informação.

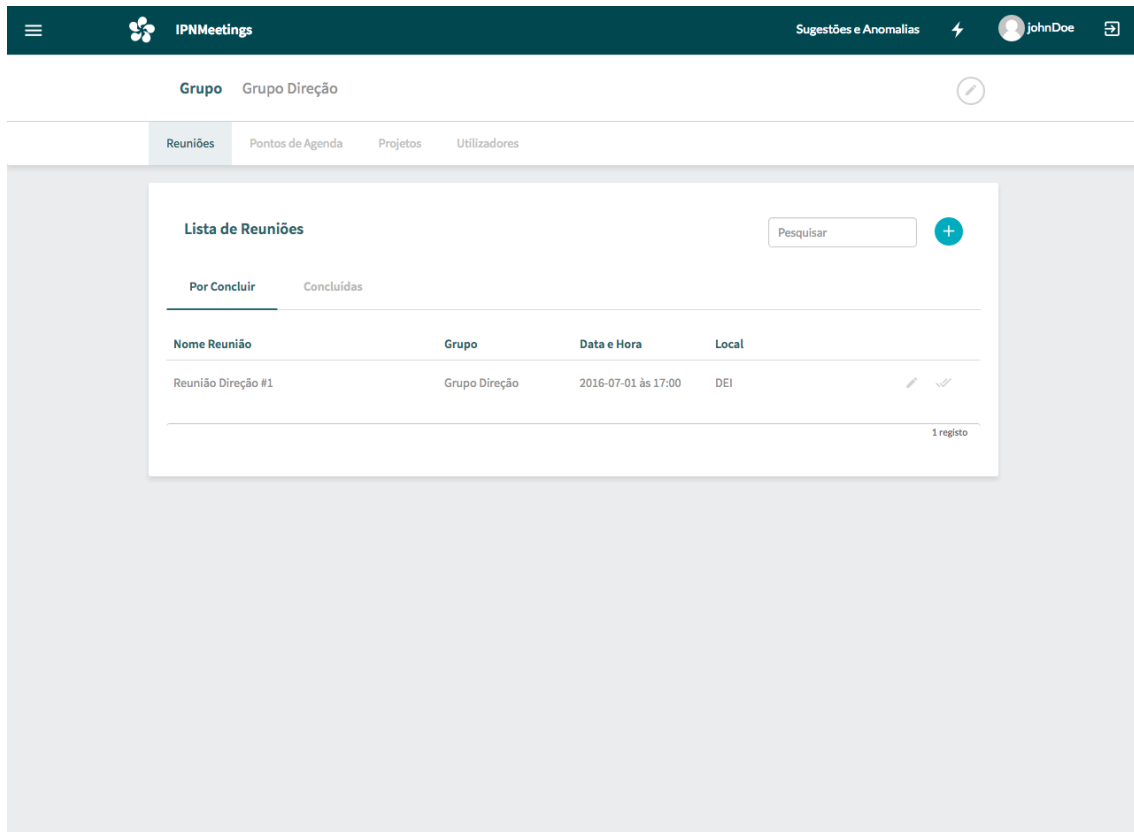


Figura 29 - Ecrã "Detalhes Grupo"

As listagens utilizadas são as mesmas das listagens apresentadas anteriormente, para que a consistência na aplicação seja elevada em todos os ecrãs.

8.3 Conclusão do capítulo

No presente capítulo foi possível ter uma ideia do design final da aplicação e de algumas funcionalidades presentes em cada ecrã.

Durante o desenvolvimento dos ecrãs foram seguidos os mockups produzidos anteriormente, pois tinham sido submetidos a testes de usabilidade e aprovados pelos utilizadores que realizaram os testes e pelo cliente. Para além disso, foram tidas em conta as alterações necessárias, identificadas na análise aos resultados dos testes de usabilidade.

Capítulo 9

Conclusões

No presente capítulo, são apresentadas propostas de funcionalidades que tem como objetivo melhorar a ferramenta de gestão de reuniões e as conclusões do estágio.

9.1 Propostas de melhorias na ferramenta

No decurso do desenvolvimento, foram encontrados aspetos/funcionalidades que podem ser melhoradas na aplicação e que permitem que esta evolua. Estes são:

- **Integração com o sistema de reserva de salas do IPN** - O IPN tem no seu Sistema de Informação uma funcionalidade que permite reservar salas onde acontecem reuniões. Seria interessante ao marcar uma reunião, verificar quais as salas livres e reservar a sala diretamente no SI do IPN. A ligação ao SI já se encontra estabelecida para realizar a ligação entre os projetos existentes na aplicação e os projetos do SI, o que diminui assim a dificuldade desta nova funcionalidade.
- **Escolha dos pontos de Follow-Up a discutir em reunião** – Em todas as reuniões de direção são discutidos pontos de Follow-Up. Durante a reunião, são apresentados todos os que se encontram em aberto. Como em algumas reuniões não são todos abordados, ocupam espaço no ecrã da reunião e podem levar a confusão. Propõe-se assim uma alteração nestes pontos, para que passem também a ser alvo de uma confirmação semelhante aos pontos de agenda. Para além disso deve ser informada a duração que a discussão de cada ponto irá ter, para se poder atribuir uma duração mais precisa à reunião.
- **Contador de tempo nos pontos de agenda durante reunião** – O tempo disponível em cada reunião é precioso e deve ser levado em conta para cada ponto de agenda. Propõe-se assim que cada ponto de agenda tenha um contador de tempo, durante uma reunião, que vai decrescendo desde a duração do ponto de agenda até 0 para que seja mais fácil controlar o tempo de reunião.

9.2 Conclusões do estágio

Ao longo do estágio foram obtidos todos os objetivos ao nível do planeamento proposto e do desenvolvimento dos requisitos e restrições à implementação identificadas.

A aplicação encontra-se já a ser utilizada pela direção do IPN-LIS e pelos gestores de projetos. Todo o processo de reunião é já realizado na aplicação, desde a preparação até à visualização da ata depois da reunião decorrer. Para além disso, o acompanhamento de projetos está já a ser feito também através da aplicação. Isto permite uma maior informação sobre os projetos durante a reunião, pois os gestores de projetos podem adicionar informação sobre cada projeto sempre que for necessário, contrariando o processo anteriormente utilizado pelo IPN-LIS em que as informações eram apenas adicionadas na reunião, notando-se muitas vezes falta de informação nos diversos projetos.

No estágio apresentado no presente relatório, o estagiário realizou as seguintes atividades:

- Investigou e ganhou competências na área de gestão de reuniões e acompanhamento de projetos;
- Planeou as atividades do projeto com elevado grau de autonomia;
- Identificou os requisitos funcionais e restrições à implementação;
- Elaborou a arquitetura da aplicação e a arquitetura de SaaS da mesma;
- Desenvolveu a aplicação apresentada;
- Ganhou competências num elevado número de tecnologias utilizadas durante o desenvolvimento;
- Ganhou conhecimentos sobre e elaborou testes de usabilidade;
- Contornou as dificuldades ao nível de desenvolvimento e design procurando ajuda de consultores experientes do IPN-LIS;
- Ganhou conhecimentos sobre os processos de deployment para a disponibilização de uma aplicação aos utilizadores finais;

Para além das competências identificadas, o estágio permitiu também completar toda a informação obtida pelo estagiário durante o curso de Licenciatura em Engenharia Informática e Mestrado em Engenharia Informática, aproximando a realidade académica da realidade empresarial, que nem sempre é uma relação óbvia durante a frequência dos cursos.

Bibliografia

- [1] “What is Waterfall model - advantages, disadvantages and when to use it?” [Online]. Available: <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/>.
- [2] C. Vennapoosa, “Staged Delivery Model.”
- [3] P. Jarzabkowski and D. Seidl, “The Role of Meetings in the Social Practice of Strategy,” *Organ. Stud.*, vol. 29, no. 11, pp. 1391–1426, 2008.
- [4] R. Schwarz, *Smart Leaders, Smarter Teams: How You and Your Team Get Unstuck to Get Results*. 2013.
- [5] G. Kaye, “Conducting Effective Meetings,” *Community Tool Box*, no. March, pp. 1–4, 2011.
- [6] M. Pina e Cunha, A. Rego, and P. Campos e Cunha, Rita Neves, *Manual de Comportamento Organizacional e Gestão* No Title. 2006.
- [7] “Paperless Meetings - Meeting Automation.” [Online]. Available: <http://novusolutions.com>.
- [8] “Task Management Made Stress Free!” [Online]. Available: <http://mijura.com>.
- [9] “Easy Meeting Collaboration For Everyone.” [Online]. Available: <http://meetingsense.com>.
- [10] “Home - Less Meeting.”
- [11] “Effective Meetings.” [Online]. Available: <http://meetingking.com/sample-board-meeting-minutes-template/>.
- [12] K. K. Breitman and J. C. S. D. P. Leite, “Managing User Stories,” *Int. Work. Time-Constrained Requir. Eng.*, pp. 1–8, 2002.
- [13] S. Enterprises, “Beneficts of User Stories.”
- [14] D. Haughey, “MoSCoW Method,” *Proj. Smart*, p. 2011, 2011.
- [15] “The Evolution of SaaS and 2016 Predictions.” [Online]. Available: <http://www.insiderhub.com/the-evolution-of-saas-and-2016-predictions/>.
- [16] C. P. Bezemer, A. Zaidman, B. Platzbeecker, T. Hurkmans, and A. Hart, “Enabling multi-tenancy: An industrial experience report,” in *IEEE International Conference on Software Maintenance, ICSM*, 2010.
- [17] G. Carraro, “Saas Simple Maturity Model.” [Online]. Available: <https://blogs.msdn.microsoft.com/gianpaolo/2006/03/06/saas-simple-maturity-model/>.
- [18] “Play Framework Documentation.” [Online]. Available: <https://www.playframework.com/documentation/2.5.x/Home>.
- [19] “Balsamiq Mockups.” [Online]. Available: <https://balsamiq.com>.
- [20] “Weld - Think it, build it.”
- [21] J. NIELSEN, “Introduction to Usability.” [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- [22] J. NIELSEN, “Why You Only Need to Test with 5 Users.” [Online]. Available: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [23] E. L. Romano Bergstrom, J. C & Olmsted-hawala, “Effects of Age and Think-Aloud Protocol on Eye-Tracking Data and Usability Measures,” *EyeTrackUX*, 2012.
- [24] S. Bronw, *The Art of Visualizing Software*. 2015.
- [25] S. Brown, “Structurizr.” [Online]. Available: <https://www.structurizr.com/>.
- [26] G. Carraro, R. Wolter, and F. Chong, “Multi-Tenant Data Architecture.” [Online].

- Available: <https://msdn.microsoft.com/en-us/library/aa479086.aspx>.
- [27] "Git." [Online]. Available: <https://git-scm.com>.
- [28] "Software Testing Class - Difference between Verification and Validation." .
- [29] J. Bach, "Exploratory Testing Explained," *Online http//www. Satisf. com/articles/et-article. pdf*, pp. 1–10, 2003.

ANEXO I

No presente documento são descritos os requisitos necessários para alcançar os objetivos propostos para o desenvolvimento do sistema de gestão de reuniões. Inicialmente são descritos os perfis de utilizadores que irão utilizar a aplicação, seguido dos requisitos funcionais que suportam o sistema. São apresentadas também restrições a ter em conta aquando da implementação, que nascem de requisitos do cliente.

1 Perfis de utilizadores da aplicação

Foram identificados 3 perfís de utilizadores da aplicação. Estes são:

Utilizador da aplicação Atribuído a um utilizador normal, que não pertence a nenhum grupo, mas que mesmo assim pode ser participante de reuniões;

Membro de um grupo Atribuído a um utilizador que pertence a um ou mais grupos;

Administrador Utilizador que tem permissão para visualizar e/ou criar todos os itens da aplicação;

O perfil "Utilizador da aplicação" é atribuído a todos os utilizadores, sendo que os restantes perfís são atribuídos consoante as diversas tarefas que cada utilizador desempenha dentro da aplicação.

2 Requisitos Funcionais

O método escolhido para os requisitos foi a criação de user stories.

Este método permite a resposta a 3 perguntas fundamentais que são "Quem?", "O quê?" e por fim "Porquê?", obtendo-se com a resposta a estas perguntas, uma estrutura para a user story. A sua principal vantagem é a facilidade e rapidez de escrita e entendimento. Como são escritas em linguagem não técnica, leva a que seja possível descrever a atividade à qual o requisito vai dar resposta, de uma maneira clara e precisa.

Cada user story contém o seu ID, o título, uma informação sobre quando podem ser realizadas (antes, durante ou depois da reunião), e, por fim, uma descrição da mesma. É apresentada também uma informação de quando cada user story pode ser realizada, para assim definirmos onde ou quando um utilizador pode realizar certas ações.

A recolha destes requisitos foi feita através de reuniões com o cliente, compilando as suas necessidades para a aplicação.

1. **Criar Reunião *Pré-Reunião*** Como membro de um grupo, quero poder adicionar novas reuniões. Aquando da criação, os participantes devem ser adicionados automaticamente consoante o grupo a que pertencer a reunião, para que esta atividade se torne mais fácil.

Na user story anterior é introduzido o conceito de grupo. Devido à necessidade de encadear diversas reuniões e de ter uma forma de guardar pontos de agenda para reuniões futuras, foi criado este conceito. Um grupo funciona como um recipiente, no qual colocamos as reuniões, pontos de agenda ou projetos que estejam relacionados.

Possibilitamos assim a utilização da aplicação para outras reuniões, por exemplo com clientes. Para isso é criado um novo grupo, onde de seguida são adicionadas reuniões a ter com esse cliente e os projetos em que esse cliente está envolvido, protegendo assim a informação de outros projetos e grupos.

2. **Editar Reunião Pré-Reunião** Como membro de um grupo, quero poder editar as informações da reunião.
3. **Ver Lista de Reuniões Pré-Reunião** Como utilizador da aplicação, quero ver uma lista de todas as reuniões dos grupos a que pertenço, podendo realizar pesquisas pelo nome, para que possa encontrar a informação mais rapidamente.
4. **Ver informações de uma reunião** Como utilizador da aplicação, quero poder ver as informações de cada reunião, como os pontos de agenda ou os participantes. Se a reunião já tiver acontecido, quero poder ver as notas tomadas em cada ponto de agenda ou os pontos de ação criados.
5. **Introduzir pontos de agenda Pré-Reunião** Como utilizador da aplicação, quero poder introduzir novos pontos de agenda a uma reunião, escolhendo a duração do ponto de agenda. Quero também optar por adicionar o ponto de agenda diretamente a uma reunião ou associar a um grupo.

Para evitar que muitos pontos de agenda sejam submetidos para cada reunião, o utilizador pode escolher adicionar um ponto de agenda diretamente a uma reunião, ou se pretende apenas que fique relacionado com o grupo em que as reuniões pertencem. Cabe ao utilizador que confirmar a reunião, escolher os que pretende que sejam discutidos.

6. **Editar Ponto de Agenda Pré-Reunião** Como utilizador da aplicação, quero poder editar os pontos de agenda.
7. **Criar pontos Follow-up Durante a Reunião** Como utilizador da aplicação, quero poder criar pontos de follow-up. A criação de pontos de follow-up acontece apenas durante reuniões. Este deve aparecer em todas as reuniões seguintes, até que seja dado por terminado.

Um ponto de follow-up é um ponto de agenda que é discutido numa reunião, mas que não pode ser dado por terminado nessa reunião por motivos como a falta de informação necessária ou a necessidade de esperar por alguma decisão externa.

8. **Editar pontos Follow-up** Sendo utilizador da aplicação, quero ter a possibilidade de editar a informação de um ponto de follow-up.
9. **Adicionar logs a ponto de Follow-up** Sendo utilizador da aplicação, quero adicionar logs a um ponto de follow up, para que a informação discutida sobre cada ponto possa ser guardada para consultar mais tarde.

10. **Fechar ponto Follow-up Durante a Reunião** Sendo utilizador da aplicação, quero poder fechar um ponto de follow-up, para que este não apareça em reuniões próximas.
11. **Confirmar Reunião Pré-Reunião** Como membro de um grupo, quero poder confirmar os detalhes de uma reunião como data, local ou participantes, para que de seguida todos os participantes recebam notificação da reunião.
12. **Escolher que pontos de agenda levar a reunião Pré-Reunião** Como membro de um grupo, quero poder escolher os pontos de agenda que vão ser debatidos em reunião, de entre todos que foram submetidos para ela ou para o grupo, para que a reunião possa ser o mais produtiva possível.
13. **Notificar outros participantes com a agenda da reunião Pré-Reunião** Como utilizador da aplicação, quero que esta me envie a agenda da reunião por email, depois de a reunião ser confirmada. Esta agenda deve conter todos os dados da reunião como participantes, pontos de agenda e os detalhes da reunião, para que a possa preparar da melhor maneira.
14. **Justificar falta a uma reunião Pré-Reunião** Como utilizador da aplicação, quero poder dar uma justificação caso não possa comparecer a uma reunião. Outros participantes da reunião devem poder justificar a ausência por mim também.
15. **Tomar nota dos participantes Durante a Reunião** Como utilizador da aplicação quero poder anotar a presença dos participantes da reunião.
16. **Passar pontos de agenda para outra reunião Durante a Reunião** Como utilizador da aplicação, quero poder passar pontos de agenda para reuniões futuras.

Em algumas reuniões, existem pontos de agenda que não podem ser discutidos na presente reunião. Isto deve-se por existir uma elevada quantidade de pontos de agenda submetidos, ou por falta de tempo durante a reunião. Desta maneira, assegura-se que não ficam esquecidos, e que são debatidos em reuniões futuras.
17. **Adiar reunião Durante a Reunião** Como utilizador da aplicação, quero poder adiar a reunião, para uma data à escolha. Caso isto aconteça, os pontos de agenda confirmados para a presente reunião, devem ser automaticamente confirmados para a mesma reunião na nova data.
18. **Organizar pontos de agenda Pré-Reunião e Durante a Reunião** Como utilizador da aplicação quero poder ordenar os pontos de agenda, para que possam ser discutidos pela ordem pretendida.
19. **Adicionar notas a pontos de agenda Durante a Reunião** Como utilizador da aplicação, quero poder tirar notas sobre cada ponto de agenda, para assim guardar a informação discutida em cada um.

20. **Adicionar decisões a pontos de agenda Durante a Reunião** Como utilizador da aplicação, quero poder anotar decisões, para que apareçam com maior facilidade em pesquisas futuras.

Estas decisões têm a mesma estrutura dos logs em pontos de agenda, mas contém informação mais importante, e normalmente são mais pequenas e detalhadas.

21. **Construir ata de reunião Depois da Reunião** Como utilizador da aplicação, quero que a aplicação construa sozinha a ata da reunião, reunindo as notas, pontos de ação ou as decisões, e envie para todos os participantes por email.
22. **Ver ata de reuniões passadas Durante a Reunião** Como utilizador da aplicação, quero poder visualizar a ata de reuniões em que fui participante.
23. **Adicionar ficheiros a pontos de agenda Durante a Reunião** Como utilizador da aplicação, quero poder adicionar ficheiros a pontos de agenda, para que possa complementar a informação de cada um.
24. **Criar pontos de ação Durante a Reunião e Projetos** Como utilizador da aplicação ou membro de um grupo, quero poder criar pontos de ação, atribuindo-os a qualquer utilizador registado.
- Um ponto de ação consiste numa tarefa que um utilizador tem de realizar. Os pontos de ação apenas podem ser criados durante uma reunião, ou então associado a algum projeto.
25. **Editar ponto de ação** Como utilizador da aplicação ou membro de um grupo, quero poder editar a informação de um ponto de ação.
26. **Adicionar notas automáticas a ponto de agenda Durante a Reunião** Como utilizador da aplicação, quero que o sistema adicione notas a um ponto de agenda de forma automática, quando um ponto de ação é criado ou editado.
27. **Listar pontos de ação** Como utilizador da aplicação, quero poder ver todos os pontos de ação dos grupos a que pertenço, podendo realizar pesquisas pelo nome, para que possa encontrar a informação pretendida mais rapidamente..
28. **Notificar criação ponto de ação** Como utilizador da aplicação, quero ser notificado, por email, quando alguém cria um ponto de ação para mim.
29. **Notificar deadline ponto de ação** Como utilizador da aplicação, quero ser notificado quando um ponto de ação criado para mim estiver prestes a passar a data limite para ser realizado.
30. **Adicionar log em ponto de ação** Como utilizador da aplicação quero poder adicionar um log a um ponto de ação. Este log funciona como uma pequena informação sobre algo realizado no âmbito desse ponto de ação.
31. **Editar um log de um ponto de ação** Como utilizador da aplicação, quero poder editar um log de um ponto de ação.

32. **Concluir ponto de ação** Como utilizador da aplicação quero poder dar um ponto de ação como concluído. Aquando disto, quero poder inserir informação acerca do que foi realizado, ou das conclusões que resultaram da realização do ponto de ação. Esta informação deve ser adicionada como um novo log no ponto de ação.
33. **Adicionar log automático a ponto de ação quando este é fechado** Como utilizador da aplicação, quero que, quando fecho um ponto de ação, a aplicação adicione automaticamente um log a referir que foi terminado naquela data por mim, para que exista um histórico de alterações em ponto de ação.
34. **Adicionar ponto de situação a projeto quando ponto de ação é editado** Como utilizador da aplicação, quero que a aplicação adicione um ponto de situação ao projeto a que o ponto de ação está associado, quando este ou algum dos seus logs é editado.
35. **Visualizar informações de ponto de ação** Como utilizador da aplicação, quero poder visualizar os detalhes de um ponto de ação, como os logs ou o utilizador que criou, para que possa inteirar-me da informação do ponto de ação.
36. **Criar grupo** Como utilizador da aplicação quero poder criar um novo grupo.
37. **Adicionar utilizadores a um grupo** Como utilizador da aplicação, quero poder adicionar utilizadores a um grupo, para que possam ter acesso a toda a informação associada ao grupo.
38. **Editar grupo** Como membro de um grupo, quero poder editar um grupo ao qual pertenço.
39. **Visualizar informações de um grupo** Como membro de um grupo, quero poder visualizar toda a informação associada ao grupo como reuniões, pontos de agenda, projetos ou pontos de follow-up.
40. **Criar Projeto** Como membro de um grupo quero poder criar um novo projeto, associando-o aos grupos que desejar.
41. **Editar Projeto** Como membro de um grupo quero poder editar um projeto que pertença a um grupo ao qual pertenço, para que possa alterar algum tipo de informação.
42. **Listar projetos** Como membro de um grupo, quero poder ver todos os projetos dos grupos a que pertenço, podendo realizar pesquisas pelo nome do projeto, para que possa encontrar a informação pretendida mais rapidamente.
43. **Visualizar informações sobre projeto** Como membro de um grupo, quero poder visualizar os detalhes de um projeto.
44. **Adicionar um ponto de situação a um projeto** Como membro de um grupo, quero poder adicionar um novo ponto de situação a um projeto. Este deve ficar com a data em que foi inserido, e com a identificação do utilizador que inseriu o ponto de situação.

Um ponto de situação consiste em alguma informação num projeto, que seja relevante ao nível da direção do IPN-LIS. Esta pode consistir na notificação de que foi enviada a fatura a um cliente, a aceitação de um projeto num programa de financiamento, o envio de um orçamento para o cliente, entre outros.

45. **Editar um ponto de situação de um projeto** Como membro de um grupo, quero poder editar um ponto de situação de um projeto. Deve ser guardado o histórico de edições.
46. **Obter histórico de alterações a um ponto de situação** Como membro de um grupo, quero poder ver as edições que foram feitas aos pontos de situação, para que possa perceber se alguma informação importante foi editada ou apagada por algum utilizador.
47. **Adicionar um ponto de situação como ponto de agenda a uma reunião** Como membro de um grupo, quero poder adicionar um ponto de agenda a partir de um ponto de situação, associando-o à próxima reunião do grupo, ou associando-o apenas ao grupo.
48. **Alterar estado de um projeto** Como membro de um grupo, necessito de poder alterar o estado em que um projeto se encontra. Cada vez que o estado for alterado, deve ser criado automaticamente um ponto de situação que reflita esta alteração.

Para avaliar o estado de cada projeto, a direção do IPN-LIS classifica os seus projetos segundo estas categorias: Lead, Proposed, Ongoing, Completed e Arquivado. A fase de Lead é a primeira pela qual um projeto passa. Esta corresponde ao período entre receber uma proposta para um projeto por parte de alguma organização ou empresa e elaborar e entregar um orçamento, ou o período antes de se realizar a candidatura a um programa de financiamento. Após isto, o projeto passa para o estado de proposed. Quando a empresa aceita a proposta do IPN-LIS ou quando o programa de financiamento aceita a proposta, o projeto passa para o estado de ongoing. Este representa um projeto que se encontra em desenvolvimento. Por fim, quando a fase de desenvolvimento termina e o projeto é entregue ao cliente, o projeto passa para o estado de completed. Este representa os projetos em que o desenvolvimento e entrega ao cliente já foi realizada, mas falta algum pormenor como por exemplo faturação. Após este estado o projeto passa para arquivado.

Estes são assim estados importantes para a direção do IPN-LIS perceber como um projeto está a avançar, e o trabalho diretivo que é necessário realizar em cada um.

49. **Marcar visita a um projeto** Como membro de um grupo, quero poder ter uma opção que me permita criar um ponto de situação automaticamente, referindo que foi o projeto visto naquela data e que não houve alterações desde a última visita.
50. **Retirar métricas dos projetos** Como membro de um grupo, quero poder retirar algumas métricas relativamente aos projetos.

Para um maior controlo nos projetos que aceitam, a direção do IPN-LIS necessita de um mecanismo que lhes permita retirar algumas métricas relativas aos projetos e à forma como estes evoluem, de um ponto de vista da direção. Deste modo, é necessário que estas métricas

possam ser vistas na aplicação. Exemplos de métricas que a direção necessite de perceber passam por ver qual a percentagem de projetos que foram finalizados, o número de projetos que um determinado cliente propôs e quantos realmente foram efetuados, a quantidade de propostas submetidas que não passaram na fase de orçamento, entre outras.

51. **Consultar projeto no decorrer de uma reunião *Durante a Reunião*** Como utilizador da aplicação, quero poder aceder à informação dos projetos que pertençam ao mesmo grupo da reunião a decorrer. Para além disso, é necessário poder adicionar novos pontos de situação ou pontos de ação a estes projetos.
52. **Realçar informação adicionada a um projeto desde a última atualização feita nele** Como membro de um grupo, quero poder identificar de forma bastante fácil, quais os projetos que sofreram alterações no último dia, semana ou mês.
53. **Acompanhar Projetos** Como membro de um grupo quero poder navegar por todos os projetos de um grupo, para poder adicionar pontos de situação a todos os projetos de forma encadeada. Este processo deve ser semelhante ao acompanhamento de projetos realizado durante uma reunião.
54. **Aceder aos projetos do SI do IPN** Como membro de um grupo, quero poder ter um link que me redirecione de um projeto presente na aplicação, para a página desse projeto no SI do IPN.
55. **Adicionar pontos de situação automaticamente em projetos** Como membro de um grupo, quero que a aplicação adicione automaticamente pontos de situação a um projeto quando é criado um ponto de ação para esse projeto, quando é adicionado um log a um ponto de ação associado a um projeto ou quando o estado do projeto é alterado, para que através dos pontos de situação possa perceber a evolução de todo o projeto e seus derivados.
56. **Não permitir a edição dos pontos de situação automáticos** Como membro de um grupo, não quero que seja permitido editar um ponto de situação automático.
57. **Adicionar Utilizadores** Como administrador, quero poder adicionar novos utilizadores à aplicação.
58. **Editar Utilizador** Como utilizador da aplicação, quero poder editar as minhas informações como email, password ou username.
59. **Listar Utilizador** Como administrador, quero poder ver uma lista de todos os utilizadores existentes na aplicação. Para além disso deve ser possível pesquisar os utilizadores pelo nome para que seja mais fácil de encontrar algum utilizador.
60. **Efetuar login com username e password** Como utilizador da aplicação, aceder à minha conta na aplicação através de autenticação com username e password.

61. **Navegar entre os diferentes menus sem passar pela página inicial da aplicação** Como utilizador da aplicação, quero poder passar entre vários ecrãs de listagens de forma fácil, para que possa aceder à informação pretendida de forma rápida.
62. **Internacionalização** Como utilizador da aplicação, quero poder seleccionar a linguagem em que quero visualizar a informação da aplicação, entre português e inglês.
63. **Página inicial com vários dados** Como utilizador da aplicação, quero poder visualizar na página inicial da aplicação informação acerca das minhas próximas reuniões, projetos, pontos de ação ou grupos. Para além disso deve ser possível inserir diferentes tipos de dados a partir desta página, como reuniões, pontos de agenda ou projetos.

3 Requisitos Não Funcionais

Em relação a requisitos não funcionais, o cliente apenas apresentou um, que é o seguinte:

1. **Usabilidade da aplicação** Por forma a que a utilização da aplicação seja rápida e eficiente, a sua usabilidade deve ser bastante elevada.

Para garantir este requisito, foram elaborados mockups para a aplicação, apresentados ao cliente em diversas reuniões. Estes mockups sofreram assim um processo de prototipagem evolutiva, evoluindo com a análise realizada pelo cliente a cada iteração. Quando estes mockups se encontravam numa fase aceite pelo cliente, foram sujeitos a testes de usabilidade, para assim poderem ser retiradas conclusões quanto à usabilidade da aplicação.

4 Restrições de implementação

Para além dos requisitos funcionais e não funcionais, foram apresentadas pelo cliente alguns critérios, que obrigam a algumas restrições na implementação. Sendo assim, as tecnologias escolhidas necessitam de possuir mecanismos que permitam responder a estas restrições. As restrições são as seguintes:

1. **Multi-Tenancy** Para garantir que, no futuro, a aplicação possa ser vendida ou cedida a outras organizações que não o IPN-LIS, a aplicação deve permitir que outras organizações a utilizem. Esta utilização por parte de outras organizações, deve acontecer através do acesso ao endereço web da aplicação, estando apenas uma instância da mesma alocada nos servidores do IPN.
2. **Utilização em dispositivos móveis** A aplicação deve possuir design responsivo para que a sua utilização em dispositivos móveis seja permitida. Para isso, serão adicionadas aos ficheiros CSS media-queries que permitam seleccionar elementos diferentes para a interface dos dispositivos móveis. Deste modo garantimos que as atividades mais importantes, tais como inserir dados na aplicação ou consultar dados, sejam possíveis de realizar em qualquer dispositivo.

3. **Adicionar informação em tempo real nos ecrãs de todos os utilizadores** Durante uma reunião, cada participante pode adicionar notas a cada ponto de agenda. Para que os outros participantes possam acompanhar da melhor forma possível a reunião, é necessário que estas notas sejam disponibilizadas para todos os participantes, assim que um utilizador as insere. Deste modo, é necessário implementar um mecanismo que permita que as notas surjam nos ecrãs de todos os utilizadores, assim que um participante adiciona uma nota a um ponto de agenda.

5 Conclusão

No presente documento foram expostas e descritas as user stories elaboradas no levantamento de requisitos. Para além disso foram também identificados os requisitos não funcionais e restrições à implementação identificadas. Esta informação foi verificada com o cliente em reunião, tendo obtido a aprovação deste.

ANEXO II

Testes de Usabilidade

Usabilidade é um atributo de qualidade. Esta verifica o quão fácil uma interface é de usar. É essencialmente definida por 6 componentes de qualidade que são[1]:

- Apreensão de conhecimento: como é fácil para os utilizadores realizarem tarefas básicas na primeira vez que se deparam com o design;
- Eficiência: rapidez com que conseguem realizar as tarefas, a partir do momento em que os utilizadores conhecem o design;
- Memorização: quando os utilizadores regressam ao design passado algum tempo, quanto tempo demoram a realizar as tarefas com eficiência;
- Erros: quantos erros os utilizadores cometem, e como conseguem recuperar desses erros;
- Satisfação: quão satisfatório e intuitivo é o design;
- Utilidade: a interface faz o que os utilizadores precisam;

Destes 6 componentes de qualidade, os que pretendemos avaliar com os nossos testes são a apreensão de conhecimento, eficiência e satisfação.

Para avaliar estes componentes, é necessário definir um conjunto de métricas a retirar. Segundo Jakob Nielsen [1], as métricas a usar prendem-se com a definição de usabilidade. Para avaliar os componentes descritos acima, poderiam ser utilizadas a taxa de sucesso de cada tarefa, o tempo que uma tarefa necessita para ser completada, a taxa de erro ou a satisfação subjetiva do utilizador. Tendo em conta a fonte citada, a métrica a utilizar será o tempo que os utilizadores demoram a realizar cada tarefa. Para além disso, é necessário anotar os comentários realizados pelo utilizador, para assim obtermos uma avaliação subjetiva.

Para conseguirmos definir estas métricas, era necessário elaborar um guião com uma sequência de ações para os utilizadores realizarem. Este guião devia começar com uma introdução ao projeto para contextualizar os utilizadores, e uma explicação do que era pretendido. Com esse objetivo, foi pensada a seguinte introdução:

Introdução

No âmbito do projeto de estágio “Sistema de Gestão de Reuniões”, onde está a ser desenvolvida uma aplicação que permita a preparação e gestão de reuniões e um acompanhamento de alto nível de projetos, irá ser realizado o seguinte teste de usabilidade, para avaliar a simplicidade e a facilidade na utilização da aplicação.

O utilizador irá ocupar o papel de um participante da reunião, moderador de reunião e, por fim, de gestor de projetos, consoante a ação que estiver a desempenhar.

De salientar que o mais importante é a facilidade com que o utilizador chega ao ecrã para desempenhar a função esperada, e não tanto a inserção de dados na aplicação.

Para definir as ações, escolheram-se as ações mais importantes a desempenhar na aplicação, e que cobrem a maior parte dos ecrãs. Deste modo, podemos ter uma ideia da usabilidade de todo o sistema.

As ações escolhidas foram:

Ação 1

- 1) Criar uma nova reunião e adicionar um ponto de agenda a essa reunião;
- 2) Editar a reunião “Reunião de Direção 3/2/2016”
- 3) Editar um ponto de Agenda já criado, da reunião “Reunião Direção 3/2/2016”;
- 4) Confirmar a reunião “Reunião Direção 3/2/2016”;
- 5) Iniciar a reunião “Reunião Direção 3/2/2016”, e de seguida;
 - a. Justificar a ausência do utilizador Carlos Lopes;
 - b. Escrever e guardar a nota “Carlos Lopes vai comprar na próxima semana”, referente ao ponto de agenda “Compra de máquina de café”;
 - c. Adicionar ponto de agenda durante a reunião;
 - d. Adicionar ponto de ação a um utilizador, no ponto de agenda “Participação em evento”;
 - e. Adicionar um ponto de follow-up;
 - f. Adicionar um novo projeto;
 - g. Ainda durante a reunião, deve aceder ao projeto Sistema de Gestão de Reuniões e realizar os passos necessários para adicionar um ponto de situação ao projeto;

- 6) Terminar a reunião, para que a ata seja enviada automaticamente para todos os utilizadores.

Ação 2

- 1) Adicionar um novo ponto de situação ao projeto “Sistema Gestão de Reuniões”;
- 2) Adicionar um ponto de ação ao projeto “Sistema Gestão de Reuniões”;
- 3) Ver todos os pontos de ação do utilizador;
- 4) Ver toda a informação do grupo “Reuniões Direção”;

No final, colocou-se um espaço para comentários do utilizador, e aspetos positivos e negativos da aplicação. Este espaço tem por objetivo recolher dados para a avaliação subjetiva da usabilidade.

Quanto ao número de utilizadores a testar, segundo Jakob Nielsen [2], deve ser à volta de 5 utilizadores, sendo este o número de utilizadores escolhido. Este número é explicado tendo em conta uma distribuição de pareto, verificável em testes de usabilidade. Seguindo esta distribuição, notamos que 80% dos problemas é detetado por apenas 20% dos utilizadores, não se justificando assim um elevado número de testes. Para uma realização de testes mais eficaz, irão ser utilizados dois perfis de utilizadores distintos. O primeiro perfil é o de alguém que é participante das reuniões de direção semanais, ou seja, alguém que já tem algum conhecimento de como as reuniões funcionam. O outro perfil é o de pessoas que não estão presentes nas reuniões de direção e que não têm tanto conhecimento acerca destas reuniões. Esta diferença permite perceber qual a diferença entre estes tipos de utilizadores na utilização da aplicação.

Já durante o teste de usabilidade, é necessário optar por uma técnica de moderação. Esta deve permitir retirar as métricas identificadas, que como foi dito serão o tempo que cada tarefa demora a ser completada e a opinião subjetiva dos utilizadores através de comentários. Esta escolha da técnica a utilizar, como referiu Jen Romano Bergstrom [3], depende dos objetivos da sessão, e do que pretendemos retirar dela. Tendo isto em conta, foi escolhida a técnica de *Retrospective Probing*. Nesta técnica, o utilizador

necessita de terminar o teste, para depois pedirmos a sua opinião. Deste modo são retiradas notas durante a sessão do que o utilizador refere, para comentar no final com o utilizador, não interferindo assim com a métrica de tempo que pretendemos retirar.

Não foi estabelecido um tempo máximo para os utilizadores terminarem as tarefas. Esta escolha, tem por objetivo dar o tempo necessário aos utilizadores para que possam terminar o teste, permitindo assim tirar mais conclusões na análise subjetiva. Permite também perceber nos comentários do utilizador, ou em perguntas no final, qual o motivo para este ter demorado mais algum tempo em alguma secção.

Resultados dos testes

Durante a realização dos testes, os tempos que os utilizadores demoram foram anotados em tabela. Foram anotados também, onde foi necessário, alguns comentários do utilizador, ou algum fator que fez com que alguma ação demorasse mais tempo. A tabela de resultados é a seguinte:

	Utilizador 1	Utilizador 2	Utilizador 3	Utilizador 4	Utilizador 5	Média
Ação 1						
1	11	5	4	15	9	8,8
2	7	20 ^{*1}	16	6	4	10,6
3	20	14	35	8	70 ^{*1}	29,4
4	38 ^{*1}	9	3	11	13	14,8
5	40	27 ^{*2}	2	2	9	16
5.a	11	5	14	9	1	8
5.b	65 ^{*2}	10	10	6	12	20,6
5.c	15	18 ^{*3}	8	10	4	11
5.d	2	12 ^{*4}	15	10	10	9,8
5.e	2	2	13 ^{*1}	1	2	4
5.f	2	2	2	1	2	1,8
5.g	5	9	10	18	10	10,4
6	4	3	3	2	10 ^{*2}	4,4
Ação 2						
1	14	3	5	8	6	7,2
2	5	2	1	7 ^{*1}	3 ^{*3}	3,6
3	20	25	22	2	7	15,2

4	12	10	6	3	4	7
Total	273	176	169	119	176	182,6

Após os testes e antes de retirar conclusões sobre o que melhorar, é necessário perceber alguns comentários dos utilizadores ou fatores que influenciaram o teste. Para isto, são agora apresentados esses comentários e os fatores que explicam um elevado tempo em algumas ações, assinaladas na tabela com o símbolo “*”.

Utilizador 1

No ponto 4, o utilizador referiu que se os ícones tivessem informação do que representavam, quando o rato passava por cima deles, seria mais fácil de descobrir qual a opção que o utilizador necessita;

No ponto 5.b, o elevado tempo despendido é explicado pela confusão que o utilizador fez entre os pontos de situação de projetos e as notas nos pontos de agenda, tendo saído do ecrã de reunião e ido ao ecrã de projetos, voltando depois para a reunião e adicionando corretamente;

Utilizador 2

No ponto 2, o utilizador demorou mais tempo do que os outros utilizadores, pois em vez de apenas tentar encontrar o botão da ação correspondente, esteve a verificar o ecrã na sua totalidade, mesmo as partes que não pertenciam às reuniões. De verificar que não procurou por não estar a encontrar a opção pretendida, mas para obter mais informação para ações futuras;

No ponto 5, o utilizador pensou que carregando no nome da reunião, esta se iniciava, e que não era necessário o botão de iniciar reunião, daí não estar a encontrar a forma de iniciar reunião;

No ponto 5.c teve alguma dificuldade em assimilar a ideia da lista de pontos de agenda se encontrar no lado esquerdo, estando à procura em outros sítios do ecrã;

No ponto 5.d, clicou ligeiramente ao lado do botão certo, não se apercebendo e ficando à espera que o outro ecrã surgisse;

Utilizador 3

No ponto 5.e, entrou no ecrã que tinha visitado anteriormente, de adicionar ponto de agenda a uma reunião, não reparando que havia separação entre pontos de agenda, pontos follow-up e projetos;

Utilizador 4

No ponto 2 da ação 2, o utilizador voltou ao painel inicial, quando tinha a opção no ecrã em que já se encontrava, demorando assim mais tempo. Este era um comportamento que o utilizador tinha diversas vezes ao longo da utilização;

Utilizador 5

No ponto 3, o utilizador não detetou facilmente que o nome da reunião era clicável para assim aceder aos detalhes da mesma. Sugeriu criar um ícone para aceder aos detalhes, ou ter algum método para perceber que os nomes são clicáveis;

No ponto 6, devido a ser utilizado outro ecrã dos restantes participantes, o utilizador necessitou de fazer scroll na página para conseguir ver o botão, ficando assim explicado o aumento de tempo neste ponto em particular;

No ponto 2 da ação 2, o utilizador sugeriu que o tamanho da caixa de texto para inserção do ponto de ação fosse maior;

Análise

Após estas observações aos comentários, notámos que existem alguns problemas a resolver, que irão ser abordados de seguida. Para além disso, foi encontrada explicação para alguns dos elevados tempos em algumas ações, que deste modo não necessitam de mudança. Estes tempos foram na sua maioria por engano do utilizador, ou por ter entrado no ecrã errado, devido a informações que tinha visto anteriormente.

Passamos agora aos problemas apontados pelos utilizadores, e à análise das ações que demoraram mais tempo.

Analisando as médias de tempo de cada ação, verifica-se que, uma das ações onde os utilizadores têm mais dificuldade, é na perceção de que o nome dos itens funciona como botão para ver os detalhes dos mesmos. Esta ação irá ser melhorada, atribuindo mais espaço a cada elemento, trabalhando com elementos de cores e atribuindo um botão

com o mesmo significado, para que seja mais fácil perceber que cada um pode ser clicado.

Outro ponto onde foi encontrada alguma dificuldade, foi na identificação do significado dos ícones, que pode ser vista na figura seguinte:



Ilustração 1 - Ícones da aplicação

Para colmatar esta falha, os ícones irão ser melhorados pelo designer do IPN-LIS Nuno Eufrásio, para que sejam mais elucidativos do seu significado. Para além disso, serão incluídos efeitos quando o utilizador coloca o rato sobre o ícone, para que apareça uma legenda com o significado.

Após esta análise dos comentários e tempos, deve ser analisada a opinião dos utilizadores. Na generalidade, estes afirmaram que os ecrãs eram intuitivos e eram bastante fáceis de usar. Outra opinião que foi dada foi a alteração de alguns pontos no ecrã apresentado durante uma reunião. Após este comentário, foi mostrado aos utilizadores o ecrã preparado pelo designer do IPN-LIS, a partir dos mockups em que os utilizadores realizaram os testes de usabilidade. Com a visualização deste ecrã, os

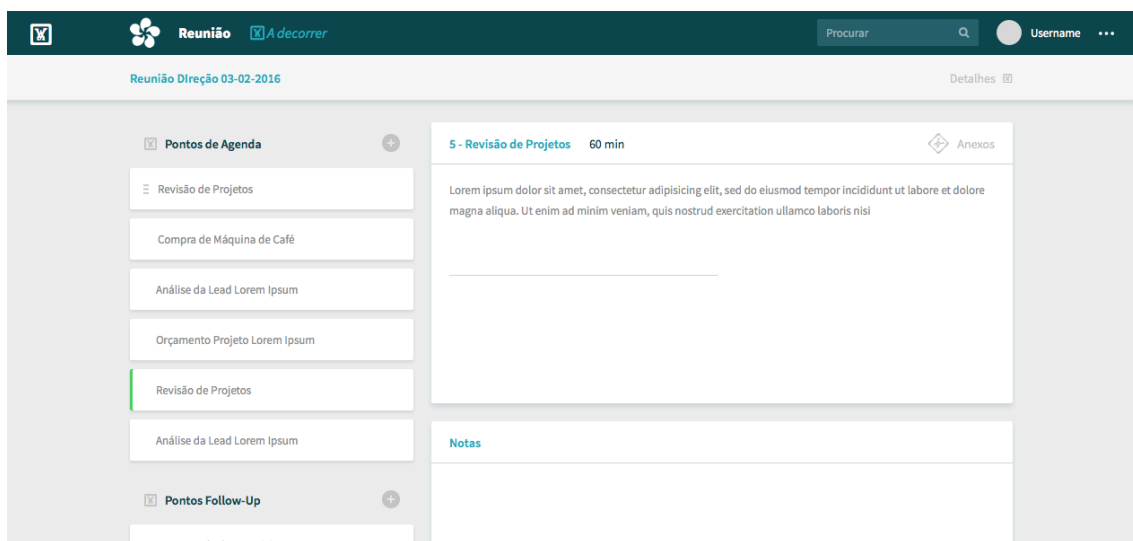


Ilustração 2 - Ecrã Durante Reuniões com Design

utilizadores afirmaram que os problemas que tinham sentido iriam ser resolvidos e que a usabilidade iria aumentar. Este ecrã pode ser visto na figura 2.

Como podemos verificar, neste ecrã é mais fácil identificar os diversos pontos de agenda, follow-up ou projetos, e a informação acerca de cada reunião também fica colapsada, não ocupando espaço no ecrã.

Conclusão

Após estes testes foi assim possível retirar algumas conclusões acerca da usabilidade da aplicação.

Quanto ao critério de apreensão de conhecimento, podemos concluir que foi cumprido, pois os utilizadores demoraram pouco tempo para concluir um número elevado de ações.

A eficiência da aplicação é elevada também, apesar de os utilizadores terem encontrado algumas dificuldades no início da utilização ou no ecrã durante a reunião. Podemos concluir isto pois, em ações parecidas umas com as outras, como por exemplo adicionar pontos de agenda, follow-up ou projetos durante a reunião, mesmo que a primeira tivesse demorado algum tempo, as seguintes foram óbvias para a maior parte dos utilizadores.

A satisfação dos utilizadores foi elevada também, pois muitos comentaram que a aplicação estava intuitiva, e que para a primeira utilização não tinham sentido muitas dificuldades na execução de tarefas.

Quanto a pontos a necessitarem de ser alterados, podemos identificar os ícones, para que se tornem mais elucidativos da ação que representam, e melhorar a forma como os utilizadores acedem aos detalhes de cada reunião.

Bibliografia

- [1] J. NIELSEN, "Introduction to Usability." [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- [2] J. NIELSEN, "Why You Only Need to Test with 5 Users." [Online]. Available: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [3] E. L. Romano Bergstrom, J. C & Olmsted-hawala, "Effects of Age and Think-Aloud Protocol on Eye-Tracking Data and Usability Measures," *EyeTrackUX*, 2012.

ANEXO III

Semana de 30 de Novembro de 2015 a 6 de Dezembro de 2015-11-30

Nesta semana existiu a necessidade de realizar uma alteração no planeamento inicial visto que a discussão e testes dos Mockups realizados atrasou. Estes irão ser debatidos durante a próxima semana com o designer do LIS-IPN Nuno Eufrásio e com o orientador da empresa Carlos Lopes para obter assim algum feedback, sendo que irão ser apresentados numa próxima reunião ao professor Paulo Rupino para percebermos qual a opinião deste em relação aos mockups desenvolvidos. Este atraso deveu-se a André Maximino ter deixado os mockups pendentes durante esta fase, para que Nuno Eufrásio pudesse melhorar os mockups da aplicação entregues por André Maximino. Em compensação, já foi realizado algum trabalho no sentido de desdobrar os requisitos em atributos e desenhar o modelo de dados da aplicação. Para compensar este atraso, o prazo para a conclusão do modelo de dados e da arquitetura da aplicação foram extendidos, o que não provoca atrasos no desenvolvimento final pois são atividades a desenvolver até à defesa intermédia do estágio.

Outra alteração no planeamento foi a sua continuação até à data de entrega do relatório intermédia, dia 22 de Janeiro, e a inclusão das atividades de escrita deste, adaptando os documentos já criados ao longo das atividades realizadas e a inclusão de testes de usabilidade a realizar aquando da finalização dos mockups, ou dos que representem as atividades essenciais na aplicação.

ANEXO IV

No presente documento são descritos os requisitos necessários para alcançar os objetivos propostos para o desenvolvimento do sistema de gestão de reuniões. Inicialmente são descritos os perfis de utilizadores que irão utilizar a aplicação, seguido dos requisitos funcionais que suportam o sistema. São apresentadas também restrições a ter em conta aquando da implementação, que nascem de requisitos do cliente.

1 Perfis de utilizadores da aplicação

Foram identificados 3 perfis de utilizadores da aplicação. Estes são:

Utilizador da aplicação Atribuído a um utilizador normal, que não pertence a nenhum grupo, mas que mesmo assim pode ser participante de reuniões;

Membro de um grupo Atribuído a um utilizador que pertence a um ou mais grupos;

Administrador Utilizador que tem permissão para visualizar e/ou criar todos os itens da aplicação;

O perfil "Utilizador da aplicação" é atribuído a todos os utilizadores, sendo que os restantes perfis são atribuídos consoante as diversas tarefas que cada utilizador desempenha dentro da aplicação.

2 Requisitos Funcionais

O método escolhido para os requisitos foi a criação de user stories.

Este método permite a resposta a 3 perguntas fundamentais que são "Quem?", "O quê?" e por fim "Porquê?", obtendo-se com a resposta a estas perguntas, uma estrutura para a user story. A sua principal vantagem é a facilidade e rapidez de escrita e entendimento. Como são escritas em linguagem não técnica, leva a que seja possível descrever a atividade à qual o requisito vai dar resposta, de uma maneira clara e precisa.

Cada user story contém o seu ID, o título, uma informação sobre quando podem ser realizadas (antes, durante ou depois da reunião), e, por fim, uma descrição da mesma. É apresentada também uma informação de quando cada user story pode ser realizada, para assim definirmos onde ou quando um utilizador pode realizar certas ações.

A recolha destes requisitos foi feita através de reuniões com o cliente, compilando as suas necessidades para a aplicação.

1. **Criar Reunião *Pré-Reunião*** Como membro de um grupo, quero poder adicionar novas reuniões. Aquando da criação, os participantes devem ser adicionados automaticamente consoante o grupo a que pertencer a reunião, para que esta atividade se torne mais fácil.

Na user story anterior é introduzido o conceito de grupo. Devido à necessidade de encadear diversas reuniões e de ter uma forma de guardar pontos de agenda para reuniões futuras, foi criado este conceito. Um grupo funciona como um recipiente, no qual colocamos as reuniões, pontos de agenda ou projetos que estejam relacionados.

Possibilitamos assim a utilização da aplicação para outras reuniões, por exemplo com clientes. Para isso é criado um novo grupo, onde de seguida são adicionadas reuniões a ter com esse cliente e os projetos em que esse cliente está envolvido, protegendo assim a informação de outros projetos e grupos.

2. **Editar Reunião Pré-Reunião** Como membro de um grupo, quero poder editar as informações da reunião.
3. **Ver Lista de Reuniões Pré-Reunião** Como utilizador da aplicação, quero ver uma lista de todas as reuniões dos grupos a que pertenço, podendo realizar pesquisas pelo nome, para que possa encontrar a informação mais rapidamente.
4. **Ver informações de uma reunião** Como utilizador da aplicação, quero poder ver as informações de cada reunião, como os pontos de agenda ou os participantes. Se a reunião já tiver acontecido, quero poder ver as notas tomadas em cada ponto de agenda ou os pontos de ação criados.
5. **Introduzir pontos de agenda Pré-Reunião** Como utilizador da aplicação, quero poder introduzir novos pontos de agenda a uma reunião, escolhendo a duração do ponto de agenda. Quero também optar por adicionar o ponto de agenda diretamente a uma reunião ou associar a um grupo.

Para evitar que muitos pontos de agenda sejam submetidos para cada reunião, o utilizador pode escolher adicionar um ponto de agenda diretamente a uma reunião, ou se pretende apenas que fique relacionado com o grupo em que as reuniões pertencem. Cabe ao utilizador que confirmar a reunião, escolher os que pretende que sejam discutidos.

6. **Editar Ponto de Agenda Pré-Reunião** Como utilizador da aplicação, quero poder editar os pontos de agenda.
7. **Criar pontos Follow-up Durante a Reunião** Como utilizador da aplicação, quero poder criar pontos de follow-up. A criação de pontos de follow-up acontece apenas durante reuniões. Este deve aparecer em todas as reuniões seguintes, até que seja dado por terminado.

Um ponto de follow-up é um ponto de agenda que é discutido numa reunião, mas que não pode ser dado por terminado nessa reunião por motivos como a falta de informação necessária ou a necessidade de esperar por alguma decisão externa.

8. **Editar pontos Follow-up** Sendo utilizador da aplicação, quero ter a possibilidade de editar a informação de um ponto de follow-up.
9. **Adicionar logs a ponto de Follow-up** Sendo utilizador da aplicação, quero adicionar logs a um ponto de follow up, para que a informação discutida sobre cada ponto possa ser guardada para consultar mais tarde.

10. **Fechar ponto Follow-up Durante a Reunião** Sendo utilizador da aplicação, quero poder fechar um ponto de follow-up, para que este não apareça em reuniões próximas.
11. **Confirmar Reunião Pré-Reunião** Como membro de um grupo, quero poder confirmar os detalhes de uma reunião como data, local ou participantes, para que de seguida todos os participantes recebam notificação da reunião.
12. **Escolher que pontos de agenda levar a reunião Pré-Reunião** Como membro de um grupo, quero poder escolher os pontos de agenda que vão ser debatidos em reunião, de entre todos que foram submetidos para ela ou para o grupo, para que a reunião possa ser o mais produtiva possível.
13. **Notificar outros participantes com a agenda da reunião Pré-Reunião** Como utilizador da aplicação, quero que esta me envie a agenda da reunião por email, depois de a reunião ser confirmada. Esta agenda deve conter todos os dados da reunião como participantes, pontos de agenda e os detalhes da reunião, para que a possa preparar da melhor maneira.
14. **Justificar falta a uma reunião Pré-Reunião** Como utilizador da aplicação, quero poder dar uma justificação caso não possa comparecer a uma reunião. Outros participantes da reunião devem poder justificar a ausência por mim também.
15. **Tomar nota dos participantes Durante a Reunião** Como utilizador da aplicação quero poder anotar a presença dos participantes da reunião.
16. **Passar pontos de agenda para outra reunião Durante a Reunião** Como utilizador da aplicação, quero poder passar pontos de agenda para reuniões futuras.

Em algumas reuniões, existem pontos de agenda que não podem ser discutidos na presente reunião. Isto deve-se por existir uma elevada quantidade de pontos de agenda submetidos, ou por falta de tempo durante a reunião. Desta maneira, assegura-se que não ficam esquecidos, e que são debatidos em reuniões futuras.
17. **Adiar reunião Durante a Reunião** Como utilizador da aplicação, quero poder adiar a reunião, para uma data à escolha. Caso isto aconteça, os pontos de agenda confirmados para a presente reunião, devem ser automaticamente confirmados para a mesma reunião na nova data.
18. **Organizar pontos de agenda Pré-Reunião e Durante a Reunião** Como utilizador da aplicação quero poder ordenar os pontos de agenda, para que possam ser discutidos pela ordem pretendida.
19. **Adicionar notas a pontos de agenda Durante a Reunião** Como utilizador da aplicação, quero poder tirar notas sobre cada ponto de agenda, para assim guardar a informação discutida em cada um.

20. **Adicionar decisões a pontos de agenda Durante a Reunião** Como utilizador da aplicação, quero poder anotar decisões, para que apareçam com maior facilidade em pesquisas futuras.

Estas decisões têm a mesma estrutura dos logs em pontos de agenda, mas contém informação mais importante, e normalmente são mais pequenas e detalhadas.

21. **Construir ata de reunião Depois da Reunião** Como utilizador da aplicação, quero que a aplicação construa sozinha a ata da reunião, reunindo as notas, pontos de ação ou as decisões, e envie para todos os participantes por email.
22. **Ver ata de reuniões passadas Durante a Reunião** Como utilizador da aplicação, quero poder visualizar a ata de reuniões em que fui participante.
23. **Adicionar ficheiros a pontos de agenda Durante a Reunião** Como utilizador da aplicação, quero poder adicionar ficheiros a pontos de agenda, para que possa complementar a informação de cada um.
24. **Criar pontos de ação Durante a Reunião e Projetos** Como utilizador da aplicação ou membro de um grupo, quero poder criar pontos de ação, atribuindo-os a qualquer utilizador registado.
- Um ponto de ação consiste numa tarefa que um utilizador tem de realizar. Os pontos de ação apenas podem ser criados durante uma reunião, ou então associado a algum projeto.
25. **Editar ponto de ação** Como utilizador da aplicação ou membro de um grupo, quero poder editar a informação de um ponto de ação.
26. **Adicionar notas automáticas a ponto de agenda Durante a Reunião** Como utilizador da aplicação, quero que o sistema adicione notas a um ponto de agenda de forma automática, quando um ponto de ação é criado ou editado.
27. **Listar pontos de ação** Como utilizador da aplicação, quero poder ver todos os pontos de ação dos grupos a que pertenço, podendo realizar pesquisas pelo nome, para que possa encontrar a informação pretendida mais rapidamente..
28. **Notificar criação ponto de ação** Como utilizador da aplicação, quero ser notificado, por email, quando alguém cria um ponto de ação para mim.
29. **Notificar deadline ponto de ação** Como utilizador da aplicação, quero ser notificado quando um ponto de ação criado para mim estiver prestes a passar a data limite para ser realizado.
30. **Adicionar log em ponto de ação** Como utilizador da aplicação quero poder adicionar um log a um ponto de ação. Este log funciona como uma pequena informação sobre algo realizado no âmbito desse ponto de ação.
31. **Editar um log de um ponto de ação** Como utilizador da aplicação, quero poder editar um log de um ponto de ação.

32. **Concluir ponto de ação** Como utilizador da aplicação quero poder dar um ponto de ação como concluído. Aquando disto, quero poder inserir informação acerca do que foi realizado, ou das conclusões que resultaram da realização do ponto de ação. Esta informação deve ser adicionada como um novo log no ponto de ação.
33. **Adicionar log automático a ponto de ação quando este é fechado** Como utilizador da aplicação, quero que, quando fecho um ponto de ação, a aplicação adicione automaticamente um log a referir que foi terminado naquela data por mim, para que exista um histórico de alterações em ponto de ação.
34. **Adicionar ponto de situação a projeto quando ponto de ação é editado** Como utilizador da aplicação, quero que a aplicação adicione um ponto de situação ao projeto a que o ponto de ação está associado, quando este ou algum dos seus logs é editado.
35. **Visualizar informações de ponto de ação** Como utilizador da aplicação, quero poder visualizar os detalhes de um ponto de ação, como os logs ou o utilizador que criou, para que possa inteirar-me da informação do ponto de ação.
36. **Criar grupo** Como utilizador da aplicação quero poder criar um novo grupo.
37. **Adicionar utilizadores a um grupo** Como utilizador da aplicação, quero poder adicionar utilizadores a um grupo, para que possam ter acesso a toda a informação associada ao grupo.
38. **Editar grupo** Como membro de um grupo, quero poder editar um grupo ao qual pertenço.
39. **Visualizar informações de um grupo** Como membro de um grupo, quero poder visualizar toda a informação associada ao grupo como reuniões, pontos de agenda, projetos ou pontos de follow-up.
40. **Criar Projeto** Como membro de um grupo quero poder criar um novo projeto, associando-o aos grupos que desejar.
41. **Editar Projeto** Como membro de um grupo quero poder editar um projeto que pertença a um grupo ao qual pertenço, para que possa alterar algum tipo de informação.
42. **Listar projetos** Como membro de um grupo, quero poder ver todos os projetos dos grupos a que pertenço, podendo realizar pesquisas pelo nome do projeto, para que possa encontrar a informação pretendida mais rapidamente.
43. **Visualizar informações sobre projeto** Como membro de um grupo, quero poder visualizar os detalhes de um projeto.
44. **Adicionar um ponto de situação a um projeto** Como membro de um grupo, quero poder adicionar um novo ponto de situação a um projeto. Este deve ficar com a data em que foi inserido, e com a identificação do utilizador que inseriu o ponto de situação.

Um ponto de situação consiste em alguma informação num projeto, que seja relevante ao nível da direção do IPN-LIS. Esta pode consistir na notificação de que foi enviada a fatura a um cliente, a aceitação de um projeto num programa de financiamento, o envio de um orçamento para o cliente, entre outros.

45. **Editar um ponto de situação de um projeto** Como membro de um grupo, quero poder editar um ponto de situação de um projeto. Deve ser guardado o histórico de edições.
46. **Obter histórico de alterações a um ponto de situação** Como membro de um grupo, quero poder ver as edições que foram feitas aos pontos de situação, para que possa perceber se alguma informação importante foi editada ou apagada por algum utilizador.
47. **Adicionar um ponto de situação como ponto de agenda a uma reunião** Como membro de um grupo, quero poder adicionar um ponto de agenda a partir de um ponto de situação, associando-o à próxima reunião do grupo, ou associando-o apenas ao grupo.
48. **Alterar estado de um projeto** Como membro de um grupo, necessito de poder alterar o estado em que um projeto se encontra. Cada vez que o estado for alterado, deve ser criado automaticamente um ponto de situação que reflita esta alteração.

Para avaliar o estado de cada projeto, a direção do IPN-LIS classifica os seus projetos segundo estas categorias: Lead, Proposed, Ongoing, Completed e Arquivado. A fase de Lead é a primeira pela qual um projeto passa. Esta corresponde ao período entre receber uma proposta para um projeto por parte de alguma organização ou empresa e elaborar e entregar um orçamento, ou o período antes de se realizar a candidatura a um programa de financiamento. Após isto, o projeto passa para o estado de proposed. Quando a empresa aceita a proposta do IPN-LIS ou quando o programa de financiamento aceita a proposta, o projeto passa para o estado de ongoing. Este representa um projeto que se encontra em desenvolvimento. Por fim, quando a fase de desenvolvimento termina e o projeto é entregue ao cliente, o projeto passa para o estado de completed. Este representa os projetos em que o desenvolvimento e entrega ao cliente já foi realizada, mas falta algum pormenor como por exemplo faturação. Após este estado o projeto passa para arquivado.

Estes são assim estados importantes para a direção do IPN-LIS perceber como um projeto está a avançar, e o trabalho diretivo que é necessário realizar em cada um.

49. **Marcar visita a um projeto** Como membro de um grupo, quero poder ter uma opção que me permita criar um ponto de situação automaticamente, referindo que foi o projeto visto naquela data e que não houve alterações desde a última visita.
50. **Retirar métricas dos projetos** Como membro de um grupo, quero poder retirar algumas métricas relativamente aos projetos.

Para um maior controlo nos projetos que aceitam, a direção do IPN-LIS necessita de um mecanismo que lhes permita retirar algumas métricas relativas aos projetos e à forma como estes evoluem, de um ponto de vista da direção. Deste modo, é necessário que estas métricas

possam ser vistas na aplicação. Exemplos de métricas que a direção necessite de perceber passam por ver qual a percentagem de projetos que foram finalizados, o número de projetos que um determinado cliente propôs e quantos realmente foram efetuados, a quantidade de propostas submetidas que não passaram na fase de orçamento, entre outras.

51. **Consultar projeto no decorrer de uma reunião *Durante a Reunião*** Como utilizador da aplicação, quero poder aceder à informação dos projetos que pertençam ao mesmo grupo da reunião a decorrer. Para além disso, é necessário poder adicionar novos pontos de situação ou pontos de ação a estes projetos.
52. **Realçar informação adicionada a um projeto desde a última atualização feita nele** Como membro de um grupo, quero poder identificar de forma bastante fácil, quais os projetos que sofreram alterações no último dia, semana ou mês.
53. **Acompanhar Projetos** Como membro de um grupo quero poder navegar por todos os projetos de um grupo, para poder adicionar pontos de situação a todos os projetos de forma encadeada. Este processo deve ser semelhante ao acompanhamento de projetos realizado durante uma reunião.
54. **Aceder aos projetos do SI do IPN** Como membro de um grupo, quero poder ter um link que me redirecione de um projeto presente na aplicação, para a página desse projeto no SI do IPN.
55. **Adicionar pontos de situação automaticamente em projetos** Como membro de um grupo, quero que a aplicação adicione automaticamente pontos de situação a um projeto quando é criado um ponto de ação para esse projeto, quando é adicionado um log a um ponto de ação associado a um projeto ou quando o estado do projeto é alterado, para que através dos pontos de situação possa perceber a evolução de todo o projeto e seus derivados.
56. **Não permitir a edição dos pontos de situação automáticos** Como membro de um grupo, não quero que seja permitido editar um ponto de situação automático.
57. **Adicionar Utilizadores** Como administrador, quero poder adicionar novos utilizadores à aplicação.
58. **Editar Utilizador** Como utilizador da aplicação, quero poder editar as minhas informações como email, password ou username.
59. **Listar Utilizador** Como administrador, quero poder ver uma lista de todos os utilizadores existentes na aplicação. Para além disso deve ser possível pesquisar os utilizadores pelo nome para que seja mais fácil de encontrar algum utilizador.
60. **Efetuar login com username e password** Como utilizador da aplicação, aceder à minha conta na aplicação através de autenticação com username e password.

61. **Navegar entre os diferentes menus sem passar pela página inicial da aplicação** Como utilizador da aplicação, quero poder passar entre vários ecrãs de listagens de forma fácil, para que possa aceder à informação pretendida de forma rápida.
62. **Internacionalização** Como utilizador da aplicação, quero poder seleccionar a linguagem em que quero visualizar a informação da aplicação, entre português e inglês.
63. **Página inicial com vários dados** Como utilizador da aplicação, quero poder visualizar na página inicial da aplicação informação acerca das minhas próximas reuniões, projetos, pontos de ação ou grupos. Para além disso deve ser possível inserir diferentes tipos de dados a partir desta página, como reuniões, pontos de agenda ou projetos.

3 Requisitos Não Funcionais

Em relação a requisitos não funcionais, o cliente apenas apresentou um, que é o seguinte:

1. **Usabilidade da aplicação** Por forma a que a utilização da aplicação seja rápida e eficiente, a sua usabilidade deve ser bastante elevada.

Para garantir este requisito, foram elaborados mockups para a aplicação, apresentados ao cliente em diversas reuniões. Estes mockups sofreram assim um processo de prototipagem evolutiva, evoluindo com a análise realizada pelo cliente a cada iteração. Quando estes mockups se encontravam numa fase aceite pelo cliente, foram sujeitos a testes de usabilidade, para assim poderem ser retiradas conclusões quanto à usabilidade da aplicação.

4 Restrições de implementação

Para além dos requisitos funcionais e não funcionais, foram apresentadas pelo cliente alguns critérios, que obrigam a algumas restrições na implementação. Sendo assim, as tecnologias escolhidas necessitam de possuir mecanismos que permitam responder a estas restrições. As restrições são as seguintes:

1. **Multi-Tenancy** Para garantir que, no futuro, a aplicação possa ser vendida ou cedida a outras organizações que não o IPN-LIS, a aplicação deve permitir que outras organizações a utilizem. Esta utilização por parte de outras organizações, deve acontecer através do acesso ao endereço web da aplicação, estando apenas uma instância da mesma alocada nos servidores do IPN.
2. **Utilização em dispositivos móveis** A aplicação deve possuir design responsivo para que a sua utilização em dispositivos móveis seja permitida. Para isso, serão adicionadas aos ficheiros CSS media-queries que permitam seleccionar elementos diferentes para a interface dos dispositivos móveis. Deste modo garantimos que as atividades mais importantes, tais como inserir dados na aplicação ou consultar dados, sejam possíveis de realizar em qualquer dispositivo.

3. **Adicionar informação em tempo real nos ecrãs de todos os utilizadores** Durante uma reunião, cada participante pode adicionar notas a cada ponto de agenda. Para que os outros participantes possam acompanhar da melhor forma possível a reunião, é necessário que estas notas sejam disponibilizadas para todos os participantes, assim que um utilizador as insere. Deste modo, é necessário implementar um mecanismo que permita que as notas surjam nos ecrãs de todos os utilizadores, assim que um participante adiciona uma nota a um ponto de agenda.

5 Conclusão

No presente documento foram expostas e descritas as user stories elaboradas no levantamento de requisitos. Para além disso foram também identificados os requisitos não funcionais e restrições à implementação identificadas. Esta informação foi verificada com o cliente em reunião, tendo obtido a aprovação deste.