

IPBRICK OS - CAFE UX (Experiência do Utilizador)

PEDRO PALA BEIRÃO MACEDO
julho de 2017

IPBRICK OS – CAFE UX (EXPERIÊNCIA DO UTILIZADOR)

Pedro Pala Beirão Macedo



Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2017

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores – Área de Especialização em Automação e Sistemas

Candidato: Pedro Pala Beirão Macedo, N° 1110170, 1110170@isep.ipp.pt

Orientação científica: João Miguel Queirós Magno Leitão, jml@isep.ipp.pt

Empresa: *IPBRICK*

Supervisão: Luís Miguel Ramalhão Ribeiro, ramalhao@*IPBRICK*.com



Departamento de Engenharia Electrotécnica
Instituto Superior de Engenharia do Porto

19 de julho de 2017

Agradecimentos

Gostaria de agradecer à empresa pelas condições para a execução do plano de trabalhos.

Aos meus colegas da *IPBRICK*, especialmente ao Engenheiro Miguel Ramalhão, por ter me confiado a atribuição desta Tese, mostrando acompanhamento e interesse no meu trabalho semana a semana.

Ao meu orientador, Professor Doutor João Leitão pela clareza, rigor e disponibilidade na colaboração do tratamento do relatório da Tese.

Ao Engenheiro Tiago Rodrigues, por me apoiar na integração da empresa, ao Engenheiro Hélder Santos por me solucionar as dúvidas e problemas que foram surgindo ao longo da realização deste trabalho e ao Engenheiro Anwaar Hussain, não só por me ter ajudado a ultrapassar vários obstáculos da parte prática da Tese, bem como no aconselhamento a nível de relatório.

Por fim, gostaria de agradecer à minha família, pelo esforço que fizeram e pelo apoio que me deram para que eu conseguisse concluir os estudos da melhor forma possível.

Resumo

As redes sociais têm cada vez mais impacto no nosso dia-a-dia, sendo esta uma peça cada vez importante na comunicação entre as pessoas em todo o mundo.

As grandes empresas, nomeadamente na área de tecnologia, têm reconhecido o valor destas plataformas de comunicação, e lentamente têm vindo a adotar este conceito a nível empresarial.

Por estas razões a *IPBRICK* criou a rede social empresarial *IPBRICK.CAFE*, que tem como finalidade a comunicação de trabalhadores da mesma empresa, de modo a trocarem informações, conhecimentos e ideias entre si. Possui a capacidade de comunicar pelos diferentes departamentos, trocar documentos em tempo real, diminuindo a necessidade de enviar um *email*, e melhorar as relações entre os colaboradores, sendo estes apenas alguns dos benefícios da adoção desta ferramenta em ambiente empresarial.

As exigências dos utilizadores têm obrigado as empresas a melhorar o seu produto. Dado este motivo, achou-se necessária a criação de novas funcionalidades úteis para os utilizadores do *CAFE* como: a criação do calendário da empresa; a moderação de *posts*; a modernização da interface do *CAFE*, de modo a tornar mais apelativa ao utilizador.

A chave para a evolução bem-sucedida do *CAFE* durante esta Tese passou pela combinação de várias tecnologias, como a *framework*, linguagens *web*, base de dados e ambiente de desenvolvimento.

Palavras-Chave

IPBRICK, CAFE, IPBRICK OS, HTML, PHP, CSS, JavaScript, AJAX, JQuery, PostgreSQL, Yii, NetBeans.

Abstract

In today's world the social media has a serious impact in our personal and professional life, being a more important tool in the communication between people around the world.

Major companies in the technology area have recognized the value of this communication platforms, and slowly have been adopting this concept to the corporate level.

Because of these reasons, *IPBRICK* has developed *IPBRICK.CAFE*, *IPBRICK's* enterprise social network. *CAFE* has the mission to communicate employees of the company, in a way to trade information, knowledge, and ideas between them. *CAFE* has the ability to communicate across different departments, trading documents in real time, decreasing the need to send an email, and improving workers relationships, are just some of the benefits of adopting this tool for the corporate environment.

The demand of the users made the companies to develop their product even more. That is the reason of the development of this Thesis, which had the purpose to create new functionalities, like the companies' calendar and post moderation, and to improve *CAFE's* design, to modernize it and turn it more appealing to the user.

The key for the well succeed evolution of *CAFE*, was the combination of various technologies (framework, web languages, database and develop environment), which became the keys to develop the software.

Keywords

IPBrick, CAFE, IPBrick Os, HTML, PHP, CSS, JavaScript, AJAX, JQuery, PostgresSQL, Yii, NetBeans.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	IX
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABELAS	XV
ACRÓNIMOS	XVII
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	2
1.2. MOTIVAÇÃO	3
1.3. OBJETIVOS	4
1.4. CALENDARIZAÇÃO	5
1.5. ESTRUTURA	6
2. ESTADO DA ARTE	7
2.1. <i>USER EXPERIENCE (UX)</i>	7
2.2. <i>APLICAÇÕES WEB</i>	8
2.3. <i>TECNOLOGIAS UTILIZADAS PELO CAFE</i>	9
2.4. <i>APACHE</i>	21
2.5. <i>LDAP</i>	22
2.6. <i>SAMBA</i>	23
2.7. <i>NETBEANS</i>	24
2.8. <i>ARCHIPEL</i>	26
2.9. <i>GROUPWARE</i>	27
2.10. <i>IPBRICK OS</i>	29
2.11. <i>CAFE – CAPACIDADES E FUNCIONALIDADES</i>	30
2.12. <i>CAFE – ESTRUTURA DO CÓDIGO</i>	32
2.13. <i>TIPOS DE TESTE</i>	33
2.14. <i>CERVISIA</i>	34
2.15. <i>CONCLUSÃO</i>	35
3. IPBRICK CAFE - DESENVOLVIMENTO	37

3.1.	MELHORAMENTOS NA INTERFACE.....	37
3.2.	MODERADOR DE <i>POSTS</i>	70
3.3.	AGENDA DE EVENTOS.....	74
3.4.	REALIZAÇÃO DE TESTES	78
3.5.	CONCLUSÃO	79
4.	CONCLUSÃO	81
	REFERÊNCIAS DOCUMENTAIS.....	85
	ANEXO A. GUIÕES DE APRENDIZAGEM DA <i>IPBRICK OS</i>: INSTALAÇÃO DA <i>IPBRICK OS</i>.....	89
	ANEXO B. GUIÕES DE APRENDIZAGEM DA <i>IPBRICK OS</i>: SERVIDOR DE <i>INTRANET</i> E <i>GROUPWARE</i>	91
	ANEXO C. GUIÕES DE APRENDIZAGEM DA <i>IPBRICK OS</i>: SERVIDOR DE COMUNICAÇÕES	95
	ANEXO D. GUIÕES DE APRENDIZAGEM DA <i>IPBRICK OS</i>: SERVIÇO DE SUPORTE LDAP – MODO MASTER/ SLAVE/ CLIENTE	99
	ANEXO E. GUIÕES DE APRENDIZAGEM DA <i>IPBRICK OS</i>: <i>UCOIP</i>	101

Índice de Figuras

Figura 1 - Logótipo IPBRICK [1].	1
Figura 2 – IPBRICK: Quadrante Mágico [2].	2
Figura 3 – Órbita UX: conceito de Dan Staffer [24].	8
Figura 4 – As páginas web estão interligadas através de múltiplas linguagens [40].	9
Figura 5 – Logótipo do HTML5 publicado no início de 2008 [7].	10
Figura 6 – Logotipo do PHP, com design de Colin Viebrock [10].	11
Figura 7 – A linguagem CSS pode ser inserida em HTML [11].	11
Figura 8 – Ilustração da interligação entre HTML e CSS [12].	12
Figura 9 – Font-awesome: relâmpago [41].	12
Figura 10 – Logótipo JavaScript [13].	13
Figura 11 – O JQuery veio facilitar a criação de páginas web [18].	14
Figura 12 – Logótipo PostgreSQL [15].	15
Figura 13 – Logótipo AJAX [18].	16
Figura 14 – Funcionamento do AJAX [18].	16
Figura 15 – Modelo MVC [22].	17
Figura 16 – Logótipo Yii [22].	18
Figura 17 –Bootstrap era chamado inicialmente de Twitter Blueprint [38].	19
Figura 18 – Bootstrap: geração de um lápis.	19
Figura 19 – Bootstrap: geração de uma lista.	19
Figura 20 – Bootstrap: geração de inputs.	19
Figura 21 – Bootstrap: geração de barras de navegação.	20
Figura 22 – Bootstrap: Geração de rótulos.	20
Figura 23 – Em 2002 o jabber passou a chamar-se XMPP [27].	21
Figura 24 – O servidor Apache foi criado em 1995 por Rob McCool [28].	21
Figura 25 – O LDAP é compatível tanto no Windows como no Linux [30].	22
Figura 26 – Papel do LDAP no mundo da web [30].	23
Figura 27 – A plataforma Samba foi criado por Andrew Tridgell em 1992 [31].	24
Figura 28 – O NetBeans foi criado em 1996 [34].	24
Figura 29 – O Archipel é compatível com vários sistemas operativos [36].	26
Figura 30 – Groupware: Email.	27

Figura 31 – Groupware: Calendário.....	28
Figura 32 – Groupware: Livro de endereços.....	28
Figura 33 – Groupware: Criar uma tarefa.....	28
Figura 34 – Groupware: Bloco de notas.....	29
Figura 35 – Groupware: Gestão de ficheiros na cloud.....	29
Figura 36 – CAFE: Página principal.....	31
Figura 37 – Performance de cada tipo de teste [42].....	34
Figura 38 – O Cervisia foi lançado em 2015 por Bernd Gehrman [46].....	35
Figura 39 – Font Awsome: ícone de fotografia.....	38
Figura 40 – Lista de imagens.....	39
Figura 41 – Lista de fotografias do álbum, com a imagem do canto inferior direito selecionado como capa (Torre de Belém).....	41
Figura 42 – Imagem ficou gravada no álbum do Teste 1.....	42
Figura 43 – Cor default no menu do álbum.....	42
Figura 44 – Cor de fundo do menu do álbum igual à do cabeçalho.....	42
Figura 45 – Valor 1 é atribuído na base de dados.....	43
Figura 46 – Utilizadores: versão antiga.....	43
Figura 47 – Utilizadores: Versão atual.....	44
Figura 48 – Configurações - Geral: Versão anterior.....	45
Figura 49 – Configurações - Geral: Versão atual.....	45
Figura 50 – Configurações – Definições de layout: Versão anterior.....	46
Figura 51 – Configurações – Definições de layout: Versão atual.....	48
Figura 52 – Barra UcoIP.....	49
Figura 53 – Barra WebPhone.....	49
Figura 54 – Os botões inferiores da UcoIP bar não aparecem.....	49
Figura 55 – Tamanho do logótipo não interfere nos botões da UcoIPbar.....	50
Figura 56 – CKEditor: Botão de upload de imagens.....	51
Figura 57 – CKEditor: Propriedades das imagens.....	52
Figura 58 – Uploader de imagens: interface.....	52
Figura 59 – Autenticação de servidores: Servidor – versão antiga.....	53
Figura 60 – Autenticação de servidores: Grupos – versão antiga.....	54
Figura 61 – Autenticação de servidores: Servidor – versão atual.....	54
Figura 62 – Autenticação de servidores: Grupos – versão atual.....	55
Figura 63 – Mudança de perfil: Dados pessoais – versão antiga.....	56
Figura 64 – Datepicker para a escolha do aniversário.....	57

Figura 65 – Mudança de perfil: Dados pessoais – versão atual.	58
Figura 66 – Versão mobile: Mudança de perfil: Dados pessoais – versão antiga (à esquerda) e versão atual (à direita).	59
Figura 67 – Mudança de perfil: Definições de VoIP – versão antiga.	60
Figura 68 – Mudança de perfil: Definições de VoIP – versão atual.	61
Figura 69 – Versão mobile: Mudança de perfil: VoIP – versão antiga (à esquerda) e versão atual (à direita).	61
Figura 70 – Mudança de perfil: Definições de email – versão antiga.	62
Figura 71 – Mudança de perfil: Definições de email – versão atual.	63
Figura 72 – Versão mobile: Mudança de perfil: Email – versão antiga (à esquerda) e versão atual (à direita).	63
Figura 73 – Mudança de perfil: Layout – versão antiga.....	64
Figura 74 – Janela RGB.	65
Figura 75 – Mudança de perfil: Layout – versão atual.....	66
Figura 76 – Versão mobile: Mudança de perfil: Layout – versão antiga.	67
Figura 77 – Versão mobile: Mudança de perfil: Layout – versão atual.	67
Figura 78 – Gestão de aplicações – versão antiga.....	68
Figura 79 – Gestão de aplicações – versão atual.....	68
Figura 80 – Adicionar contacto externo – versão antiga.....	69
Figura 81 – Adicionar contacto externo – versão atual.....	69
Figura 82 – Administrador - perfil com permissões de “Aprovar feeds”.....	70
Figura 83 – Notificações: dois feeds por aprovar.....	72
Figura 84 – Publicação aprovada.	73
Figura 85 – Publicação rejeitada.	73
Figura 86 – Publicação pendente.	74
Figura 87 – Calendário do Groupware.	74
Figura 88 – Eventos criado no Groupware.....	76
Figura 89 – Interface dos eventos criados no Groupware na página principal do CAFE.	76
Figura 90 – Datepicker: escolha o mês para a visualização dos eventos.	77
Figura 91 – Calendários guardados para cada utilizador.....	77
Figura 92 – Calendários: o utilizador escolheu 3 dos 4 para visualização na página principal.	77
Figura 93 – Nesta secção é possível ocultar os eventos.....	78
Figura 94 – IPBRICK OS: Gestão de configurações.	90
Figura 95 – IPBRICK OS: Configuração do servidor de domínio.....	91
Figura 96 – IPBRICK OS: Gestão de utilizadores.	92

Figura 97 – IPBRICK OS: Gestão de grupos.....	92
Figura 98 – IPBRICK OS: Áreas de trabalho.	93
Figura 99 – IPBRICK OS: Informações gerais do sistema.	95
Figura 100 – IPBRICK OS: Firewall da IPBRICK.....	97
Figura 101 – IPBRICK OS: Modos de autenticação.....	100
Figura 102 - Rede com IPBRICK Master, Slave e Cliente.	100
Figura 103 – IPBRICK OS: Registo de telefones.	101
Figura 104 – IPBRICK OS: Gestão de rotas.....	102

Índice de Tabelas

Tabela 1	Calendarização do projeto.....	5
----------	--------------------------------	---

Acrónimos

- Ajax – Asynchronous Javascript and XML
- ANT - Cascading Style Sheets
- CVS - Concurrent Versions System
- DHCP - Dynamic Host Configuration Protocol
- DNS - Domain Name System
- DVD - Digital Video Disc
- HTML - HyperText Markup Language
- HTTP - Hypertext Transfer Protocol
- IETF - Internet Engineering Task Force
- IMAP - Internet Message Access Protocol
- IP - Internet Protocol
- JS - JavaScript
- KMV - Kernel-based Virtual Machine
- LAN - Local Area Network
- LDAP - Lightweight Directory Access Protocol
- MVC - Model View Controller
- NNTP - Network News Transfer Protocol
- NTP - Network Time Protocol

OS	- Operating System
PHP	- Hypertext Preprocessor
POP	- Post Office Protocol
RGB	- Red Green Blue
SGBD	- Sistema de Gestão de Base de Dados
SMTP	- Simple Mail Transfer Protocol
SNMP	- Simple Network Management Protocol
SQL	- Structured Query Language
TCP	- Transmission Control Protocol
TIC	- Tecnologias de Informação e Comunicação
UX	- User Experience
UcoIP	- Unified Communication over IP
URL	- Uniform Resource Locator
VNC	- Virtual Network Computing
VoIP	- Voice Over IP
VPN	- Virtual Private Network
PPTP	- Point-to-Point Tunneling Protocol
WWW	- World Wide Web
XHTML	- eXtensible Hypertext Markup Language
XMPP	- Extensible Messaging and Presence Protocol
Yii	- Yes It Is

1. INTRODUÇÃO

O projeto descrito nesta dissertação incide na criação e melhoramento das funcionalidades oferecidas pela rede social empresarial *IPBRICK.CAFE*, pertencente à empresa *IPBRICK, SA*.

A *IPBRICK* dedica-se ao desenvolvimento de ferramentas e soluções que simplificam e aceleram a instalação e configuração de sistemas operativos de rede. Esta é repartida fundamentalmente no chamado “Quadrante Mágico da *IPBRICK*”, fornecendo soluções de: gestão de documentos e processos, servidor de *email* e ferramentas colaborativas, comunicações unificadas, e rede social empresarial completamente integrados.



Figura 1 - Logótipo IPBRICK [1].

A rede social *IPBRICK.CAFE* tem como objetivo proporcionar um espaço virtual comum, para colaboradores da mesma empresa trocarem informações úteis, melhorarem as suas relações e aumentarem sobretudo a produtividade.

1.1. CONTEXTUALIZAÇÃO

A escolha deste projeto surgiu através duma proposta de Estágio Curricular por parte da *IPBRICK*, necessitando esta de novos colaboradores para o desenvolvimento do *CAFE*.

A *IPBRICK, SA* foi fundada no Porto, em Maio de 2000, sendo um fabricante e distribuidor especializado em Soluções de Comunicações Empresariais. As soluções que produz assentam no chamado “Quadrante Mágico da *IPBRICK*” referido anteriormente, de fácil utilização e implementação, disponíveis à distância de um clique. A *IPBRICK* possui instalações em Portugal e Angola, com distribuidores diretos em mais de 20 países, nos 5 continentes [2].



Figura 2 – *IPBRICK*: Quadrante Mágico [2].

As aplicações sociais desempenham um papel fundamental na interação com o mundo que nos rodeia. Cria-se e partilha-se informação em comunidades virtuais e em redes.

Com a evolução da tecnologia e do modo de comunicação, sentiu-se a necessidade de desenvolver uma rede social empresarial, pois estas tornaram-se numa ferramenta de comunicação indispensável para qualquer empresa, envolvendo grandes vantagens e benefícios. O uso destas são vitais para:

- Um bom plano de trabalho;
- Melhor organização de objetivos;
- Marketing digital;
- Melhor comunicação entre utilizadores da empresa;
- Transmitir uma visão de atualidade.

Devido a estes motivos, a *IPBRICK* desenvolveu o *IPBRICK.CAFE*, uma rede social corporativa, onde se pode seguir os seus colegas de trabalho, partilhar ideias, gostos e interesses, promovendo desta maneira um verdadeiro espírito de equipa [4].

Simultaneamente, o *CAFE* é um espaço que torna mais fácil a comunicação empresarial, através da facilidade de acesso e utilização, das diferentes formas de comunicação oferecidas na interface: voz, vídeo conferência, *email* e *chat*, e pela partilha rápida de documentos, apresentações, notícias ou até pela realização de *brainstormings* com um utilizador ou grupo de utilizadores.

Este apresenta-nos as suas funcionalidades de uma forma simplificada, sendo um espaço privado e seguro de comunicações empresariais, que unifica e descomplica o acesso à informação, aumentando a troca de conhecimentos, tarefas e contatos de gestão entre os colaboradores.

O portal está dividido em duas áreas: comunicacionais e aplicacionais. Estas permitem aos utilizadores aceder a uma série de dados e recursos que agora estão espalhados por vários subsistemas, garantindo uma comunicação fácil e unificada [3].

1.2. MOTIVAÇÃO

Este projeto foi escolhido tendo em conta a constante evolução do mundo da tecnologia, mais precisamente no campo das redes sociais. Por este motivo, sentiu-se a necessidade de melhorar e criar novas funcionalidades neste campo.

Para a fase inicial da Tese, foi necessário a execução de guiões de treino utilizados nas aulas de formação da *IPBRICK* relativos ao *software* da *IPBRICK*, a *IPBRICK OS*, com o objetivo de explorar as suas potencialidades. Para tal, foram necessários os conhecimentos adquiridos nas Unidades Curriculares de “Comunicação de Dados”, do 3º ano de *LEEC* do *ISEP* e de “Interligação e Gestão de Redes”, do 1º ano de *MEEC* do *ISEP*.

Para garantir o sucesso deste projeto, procedeu-se ao estudo das linguagens *HTML*, *PHP*, *JavaScript* e *CSS*, bem como *AJAX* e da *framework Yii*, sendo algumas destas tecnologias lecionadas nas Unidades Curriculares de “Desenvolvimento de Aplicações Computacionais”, do 1º ano de *LEEC* do *ISEP* e em “Modelação e Simulação Industrial”, do 1º ano de *MEEC* do *ISEP*.

1.3. OBJETIVOS

Este projeto surgiu do desejo de evoluir o *IPBRICK.CAFE*, no sentido de satisfazer ainda mais os utilizadores, atualizando-o de acordo com as mudanças constantes do dia-a-dia.

Tendo por base esta ideia e a necessidade de revolucionar a imagem que os utilizadores têm do *IPBRICK.CAFE*, o objetivo deste projeto consiste em analisar as tendências na área das *TIC* e projetar e implementar um conjunto de medidas que transformem o *CAFE* numa aplicação necessária a nível empresarial.

Entre essas medidas já se identificaram alguns módulos que os utilizadores desejam ver disponibilizados no *IPBRICK.CAFE*:

- Agenda de eventos na empresa: os eventos tendem a ser um excelente meio de comunicação e aproximação entre os vários funcionários de uma empresa. Dado a ocupação dos colaboradores, estes têm pouco tempo para a realização de alguns eventos ou tarefas, tanto por falta de tempo ou mesmo por esquecimento. Dado este motivo, achou-se necessária a criação do calendário da empresa, considerando esta uma mais valia para o *CAFE*. Neste momento é utilizado o *Groupware* para marcar e visualizar os eventos criados. Sendo a agenda de eventos uma peça importante na *IPBRICK*, foi decidido dar a opção de visualizar os próximos eventos na página principal do *CAFE*, permitindo aos utilizadores não se esquecerem de possíveis eventos a surgir;
- Visão do calendário da empresa: cada empresa tem o seu próprio calendário, contendo por exemplo aniversários ou feriados/folgas dos funcionários. Um calendário numa rede social como o *CAFE* seria útil para relembrar os utilizadores os dias especiais da empresa;
- Dados estatísticos sobre as interações dos utilizadores no *CAFE*: é sempre importante para uma empresa saber os dados estatísticos de uma aplicação, pois com estas informações os gestores conseguem monitorizar o que é mais acedido pelos utilizadores e o que é menos, podendo assim estudar e melhorar os setores menos utilizados;
- Moderação de *posts*: para evitar que um utilizador publique no *CAFE* algo menos apropriado, é necessário criar um moderador de *posts* para impedir que estes tipos de publicações sejam efetuados;

- Repensar a interface do *CAFE* para se tornar mais apelativa (em *desktop* e na versão *mobile*): à medida que os anos avançam, os gostos das pessoas vão ficando mais exigentes. O melhoramento de vários setores do *CAFE* necessitam de ser efetuados de modo a satisfazer os gostos dos utilizadores, não só em termos de *design*, como também na facilidade de interação entre o utilizador e a rede social.

1.4. CALENDARIZAÇÃO

A calendarização das atividades conducentes a esta dissertação está descrita na Tabela 1. As tarefas desenvolvidas consistiram inicialmente: na aplicação de guiões da empresa, com o objetivo do novo colaborador se ambientar às funcionalidades do *software* da empresa (*IPBRICK OS*); no estudo das várias linguagens de programação *web*, de modo a uma melhor adaptação ao código do *IPBRICK.CAFE*; criação e melhoramento de módulos no *IPBRICK.CAFE*; na elaboração do relatório final e por fim na realização da apresentação.

Tabela 1 Calendarização do projeto

		Dezembro				Janeiro				Fevereiro				Março				Abril				Maio				Junho							
ETAPA	S E M A N A	1 2 3 4				1 2 3 4				1 2 3 4				1 2 3 4				1 2 3 4				1 2 3 4											
		Introdução ao tema	1S																														
Fase de integração	3S																																
Estudo do código do <i>CAFE</i>	2S																																
Criação/melhoramento de módulos no <i>CAFE</i>	18S																																
Elaboração do relatório	10S																																
Elaboração da apresentação	2S																																

1.5. ESTRUTURA

No Capítulo 1 é feita uma introdução ao tema, onde estão presentes a contextualização, a motivação da qual este projeto foi escolhido, os objetivos a atingir, a calendarização da mesma e por fim a estrutura, de modo melhorar a organização do relatório.

No Capítulo 2 é descrito o “estado da arte”, referindo vários pontos de interesse, ao descrever um pouco sobre cada tecnologia a utilizar no *CAFE*. Adicionalmente, será apresentada a estrutura da rede social *CAFE*, bem como o seu código.

O Capítulo 3 é destinado à realização de melhoramentos e na inclusão de novas funcionalidades na rede social *IPBRICK.CAFE*. Neste, são descritos os processos para a realização das tarefas e o respetivo resultado final de cada uma.

O Capítulo 4 é dedicado à conclusão, referindo as dificuldades encontradas, o que se aprendeu na realização deste Estágio Curricular, um resumo do trabalho elaborado na Tese, o trabalho futuro a desenvolver na rede social e as potencialidades que o *IPBRICK.CAFE* pode vir a atingir.

Nos anexos são executados os guiões de formação da *IPBRICK OS*.

2. ESTADO DA ARTE

Este capítulo incide, a nível teórico, no conceito de *user experience*, nas aplicações *web*, na descrição das várias tecnologias utilizadas ao longo do desenvolvimento da Tese, na estrutura do *CAFE* e no seu código.

2.1. *USER EXPERIENCE (UX)*

O *User Experience* (experiência do utilizador) entende-se como o estudo das sensações e emoções que os utilizadores sentem ao utilizar um produto na área da tecnologia. Esta interação entre o utilizador e o computador é feita através de uma interface. O estudo das interfaces pode ser relacionado com alguns campos de estudo das Ciências da Computação, que se dedicam a estudar a interface e a interação entre as pessoas e os computadores [23].

Existem variados conceitos sobre *UX*, entre os quais se destacam os seguintes [24]:

- *Brinck, Gerle e Wood* entendem o *UX* como “o grau em que as pessoas podem realizar um conjunto de tarefas”. Os autores atribuem esse grau como produto dos seguintes objetivos do projeto de desenvolvimento de software: “funcionalmente correto, eficiente, fácil de aprender, fácil de lembrar, tolerante a erros e subjetivamente agradável”;

- *Steve Krug* define *UX* como “assegurar-se de que algo funcione bem: que uma pessoa com habilidade e experiência comuns possa usar algo para seu propósito desejado sem ficar frustrada com isso”;
- *Nielsen e Loranger* defendem o *UX* como “um atributo de qualidade que relaciona o quanto algo é fácil de utilizar”. Aspectos citados por *Brinck, Gerle e Wood* também aparecem na definição de Nielsen que envolve cinco características principais dos sistemas com alta usabilidade:
 - Fácil de aprender;
 - Eficiente;
 - Fácil de lembrar;
 - Tolerante a erros;
 - Subjetivamente agradável.

Dan Staffer desenhou, na Figura 3, um modelo de órbita na qual explica a sua definição de *UX*. Nesta, é possível entender que esta abrange várias áreas, desde o *design* à engenharia.

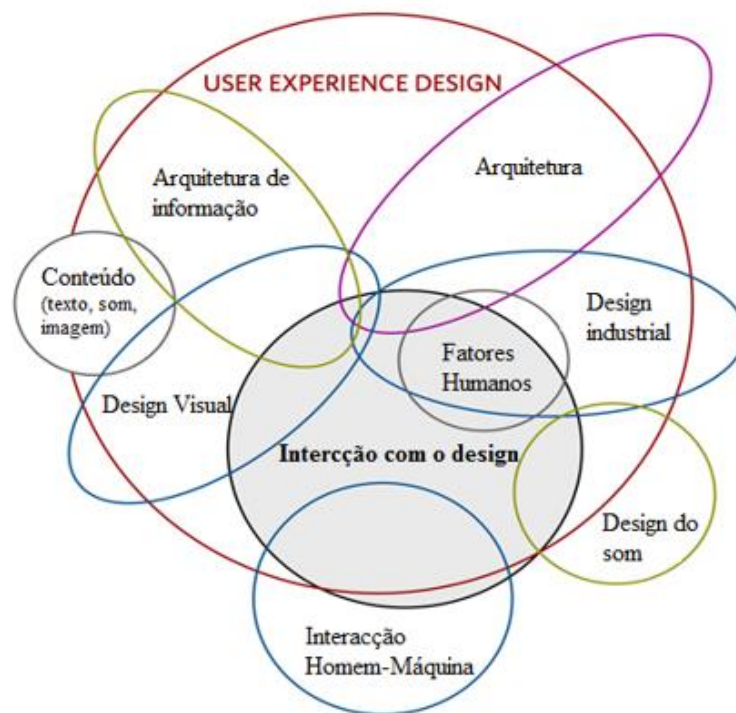


Figura 3 – Órbita UX: conceito de *Dan Staffer* [24].

2.2. APLICAÇÕES WEB

Uma aplicação *web* consiste num conjunto de sistemas informáticos, idealizados para *browsers* ou aplicações desenvolvidas em *HTML*, *JavaScript* e *CSS*. Tanto pode ser executado através de um servidor *HTTP* como localmente, no dispositivo de um utilizador.

São compostas por páginas *web* com as quais os utilizadores interagem através de *browsers* [39].

Estas páginas são ligadas com outras páginas ou aplicações externas através de *links*. Uma página *web* é definida como a composição de objetos. Este pode ser um *link*, um botão de comando, uma mensagem, uma imagem, um arquivo, áudio ou uma aplicação. Dessa forma, uma aplicação *web* é entendida em termos de páginas *web* que por sua vez são definidas recursivamente em termos de objetos [40].



Figura 4 – As páginas *web* estão interligadas através de múltiplas linguagens [40].

2.3. TECNOLOGIAS UTILIZADAS PELO *CAFE*

A evolução das linguagens de programação e marcação como o *HTML*, *PHP*, *JavaScript* e *CSS* e as *frameworks* têm um impacto forte na forma como as pessoas utilizam as interfaces *web*. Por outro lado, permitem a criação de interfaces mais apelativas e *user friendly*, contribuindo de certa forma para que os utilizadores se tornem cada vez mais exigentes relativamente às funcionalidades, ao aspeto e ao nível de usabilidade das soluções que lhe são oferecidas [4].

2.3.1. *HTML*

A linguagem *HTML* (*Hypertext Markup Language*), criada pelo físico britânico *Tim Berners-Lee* em 1993, passa por uma tecnologia de marcação baseada em *tags*, através das quais se formata o conteúdo das páginas *web*.

No início, esta linguagem tinha regras de sintaxe mais tolerantes, o que facilitava a publicação de conteúdos na *web*. Nos dias de hoje a sintaxe é muito mais minuciosa, o que leva a um código mais definido e complexo. Por esta razão, a utilização de ferramentas para criação de páginas de *HTML* aumentou, assim como a tendência em tornar a sintaxe cada vez mais rígida, de modo a minimizar as faltas de compatibilidades nos *browsers* e tornar o código cada vez mais convencional, sem que a facilidade de criação de páginas *web* seja alterada [7].

Em Janeiro de 2008, surgiu a versão 5 do *HTML*, o *HTML5*, que tem como fim a estruturação e apresentação de conteúdo para a *World Wide Web*. Esta nova versão traz consigo modificações relevantes no que toca ao comportamento do *HTML* no domínio *Web*, através de novas funcionalidades como semântica e acessibilidade. Proporciona o uso de novos recursos, dos quais só eram antes possíveis através do uso de outras tecnologias. O seu objetivo tem sido melhorar a linguagem com o suporte para as mais recentes multimédias, mantendo-a facilmente nítida para os programadores e compreendida por computadores e outros tipos de dispositivos. O *HTML5* será o novo paradigma para *HTML*, *XHTML*, e *HTML DOM*, sendo esta uma linguagem em constante crescimento [8].



Figura 5 – Logótipo do *HTML5* publicado no início de 2008 [7].

2.3.2. *PHP*

Criado em 1994 por *Rasmus Lerdorf*, o *PHP* (*Hypertext Preprocessor*) é uma linguagem de *script open source* bastante utilizada para o desenvolvimento *web* e que pode ser inserida dentro do *HTML*.

Permite criar *websites* dinâmicos, proporcionando uma interação com o utilizador através de parâmetros do *URL*, formulários e *links*. Esta linguagem tem como uma das características mais importantes o suporte a um grande número de base de dados. O *PHP* é

executado no lado do servidor, sendo enviado para o cliente apenas *HTML*. Tem a vantagem de não exibir o código fonte para o cliente, uma vez que o programa está a lidar com senhas ou qualquer outro tipo de informação confidencial [9].

Com o *PHP*, é possível recolher dados de um formulário, criar páginas dinamicamente ou enviar e receber *cookies*. Por estas razões, construir uma página baseada numa base de dados torna-se uma tarefa extremamente simples com *PHP*. Além disso, suporta outros serviços através de protocolos como *IMAP*, *SNMP*, *NNTP*, *POP3* e *HTTP*. Ainda é possível abrir *sockets* e interagir com outros protocolos [10].



Figura 6 – Logotipo do PHP, com design de Colin Viebrock [10].

2.3.3. CSS

O *CSS (Cascading Style Sheets)* trata-se de uma linguagem de estilo, criada em 1995 por *Håkon Wium Lie* e *Bert Bos*, preferencialmente utilizada para definir a apresentação de documentos escritos numa linguagem de marcação, como o *HTML* ou *XML*. Tem como principal vantagem a separação do formato e do conteúdo de um documento [11].



Figura 7 – A linguagem CSS pode ser inserida em HTML [11].

Em vez de colocar a formatação dentro do documento, o programador cria um *link* para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um portal. Basta apenas modificar um arquivo para alterar a aparência do portal.

Esta separação irá aumentar a acessibilidade aos conteúdos, permitindo assim, maior flexibilidade e controlo na especificação das características da apresentação das páginas [12].

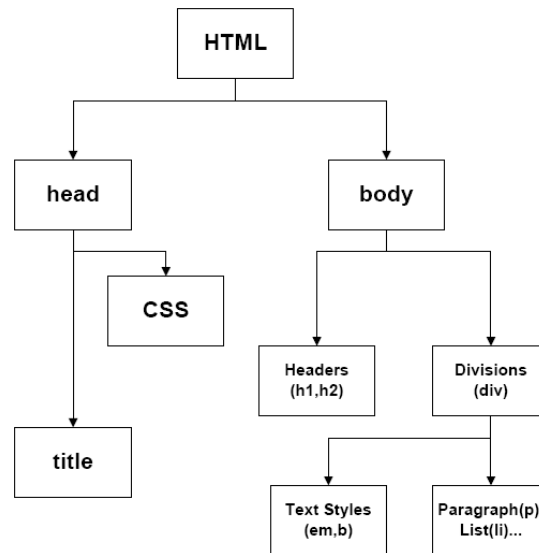


Figura 8 – Ilustração da interligação entre *HTML* e *CSS* [12].

2.3.3.1. Font-Awesome

O *Font Awesome* passa por um projeto *open source* que nos concede, de forma muito simples e rápida, incorporar ícones em páginas *web* [41].

Para usar esta ferramenta, basta inserir ícones usando a seguinte sintaxe:

```
<i class="fa fa-nome fa-tamanho"></i>
```

No código *HTML* acima referido, deve-se referir o nome e tamanho do ícone, tal como se encontra demonstrado na Figura 9.



Figura 9 – Font-awesome: relâmpago [41].

2.3.4. JAVASCRIPT

A linguagem *JavaScript* é utilizada como uma ferramenta que dá acesso a um grande número de truques e de funcionalidades avançadas que estão ao alcance de todos. Esta

linguagem é usada em milhões de páginas *web* com intuito de melhorar o projeto criado pelo programador. Foi originalmente implementada como parte dos navegadores *web* para que as *scripts* pudessem ser executadas do lado do cliente e interagissem com o utilizador sem a necessidade deste *script* passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido. O *JavaScript* também pode ser utilizado do lado do servidor para correr *scripts* ou desenvolver jogos e aplicações para dispositivos móveis. Esta foi uma linguagem criada por *Brendan Eich* em 1995 enquanto trabalhava na empresa *Netscape Communications Corporation* [13].



Figura 10 – Logótipo JavaScript [13].

2.3.5. JQUERY

O *JQuery* é uma biblioteca de *JavaScript* bastante rica, da qual interage com a linguagem *HTML*, criada para simplificar os *scripts* processados no *browser* do cliente. Esta biblioteca, a mais popular e utilizada em *JavaScript*, foi lançada no final de 2006 por *John Resig*.

Passa por um projeto *open source*, que torna possível escrever código em *JavaScript*, tornando-se num instrumento importantíssimo para a compatibilidade entre os *browsers*. Possui a capacidade de simplificar a forma como são processadas as páginas *HTML*, manipulados os eventos e o acréscimo de interações *AJAX* numa página *web*. Nesta biblioteca, é possível encontrar elementos, as funções para manipular os eventos, efeitos e animações associados ao *HTML*. Através dos seletores, é possível modificar elementos *HTML*, tanto em grupo ou como elemento único. As funções de tratamento permitem executar um código que surge através de um evento, com instruções como "*onclick*" ou "*onchange*" que fazem com que, após um clique sobre uma área da página ou numa mudança do *input* de uma lista, corra um *script* que vai ser responsável, por exemplo pela atualização de uma tabela. O *JQuery* também permite, de uma forma simples, alterar o *CSS* do código

de acordo com os eventos, como a mudança de cor de uma zona após um clique sobre essa mesma área [18].



Figura 11 – O *jQuery* veio facilitar a criação de páginas *web* [18].

2.3.6. *SQL*

No *SQL* (*Structured Query Language*), o SGBD é encarregue de escolher o melhor procedimento a executar, de modo a que os resultados obtidos sejam os mais eficientes. Trata-se de uma linguagem de interrogação de base de dados relacionais que possibilita a um gestor a realização de importantes tarefas na base de dados como criar, alterar, eliminar tabelas, bem como inserir, consultar, atualizar e apagar os dados presentes nas mesmas. É constituída por palavras-chave, que permitem a execução das tarefas referidas, destacando os seguintes comandos:

- ***Create Table***: cria uma tabela;
- ***Alter Table***: altera uma tabela;
- ***Drop Table***: elimina uma tabela,
- ***Insert***: adiciona dados a uma tabela;
- ***Update***: altera/atualiza os dados de uma tabela;
- ***Delete***, elimina dados de uma tabela.

Para além destes comandos, existem comandos de consulta, que têm como função facilitar a pesquisa de um dado específico. Entre estes comandos destacam-se os seguintes [16]:

- ***Select***, indica qual o nome a ser utilizado nas colunas a selecionar para a tabela;
- ***From***, indica o nome das tabelas de onde se extrai as informações desejadas;
- ***Where***, indicam-se condições de seleção e são eliminadas todas as linhas da tabela obtida do "*From*" que não cumpram as condições;

- **Group By**, agrupa os resultados baseando-se em determinadas colunas a fim de todas as linhas desse grupo tenham o mesmo valor para as colunas que são passadas como argumento;
- **Having**, funciona da mesma forma que a condição "Where" e tem como função filtrar a informação obtida com o "Group By".

2.3.7. **POSTGRESQL**

O *PostgreSQL* é um sistema de gestão de base de dados *opensource*. Foi criado por *Michael Stonebraker* em 1982.

Possui uma arquitetura que ganhou forte prestígio no que toca à fiabilidade, na integridade de dados e conformidade a padrões. É compatível em todos os grandes sistemas operacionais, incluindo *GNU/Linux*, *Unix* e *MS Windows*.

Este modelo de desenvolvimento de *software* possibilita o desenvolvimento de qualidade, permitindo:

- Tratar utilizadores como parceiros, pois estes contribuem diretamente com a evolução do *software*, ao apresentar os problemas, as suas necessidades, as suas sugestões de solução e o seu próprio código fonte de solução. Deste modo, os utilizadores ajudam nas melhorias e na depuração do *software*.
- Reutilizar o código fonte;
- Lançar novas versões de forma rápida de modo a não deixar fugir os utilizadores. Graças aos vários utilizadores que testam o *software*, os problemas são identificados solucionados de forma rápida [14].



Figura 12 – Logótipo *PostgreSQL* [15].

2.3.8. **AJAX**

O *AJAX* é uma tecnologia que se tornou popular em 2005 pela *Google*. Esta não é uma nova linguagem de programação, mas sim uma nova forma de utilizar os padrões *web*:

JavaScript, XML, HTML, CSS. Passa por uma técnica de desenvolvimento que possibilita que as páginas *web* sejam atualizadas de forma assíncrona, através de um pequeno número de trocas de dados com o servidor. Isto possibilita atualizar apenas partes de uma página *web*, sem ter que recarregar a página na sua totalidade [17].



Figura 13 – Logótipo AJAX [18].

Quando se sucede um evento, é criado um objeto *XMLHttpRequest*, uma componente fundamental do *AJAX*, que é enviado para o servidor. Este analisa o pedido, processa e cria a resposta, enviando-a para o *browser*. A informação que é recebida pode ser processada em *background* e usada para, de uma forma dinâmica, atualizar elementos numa página *web*, sem necessidade de recarregar toda a página. O funcionamento desta ferramenta está ilustrado na figura abaixo apresentada [18].

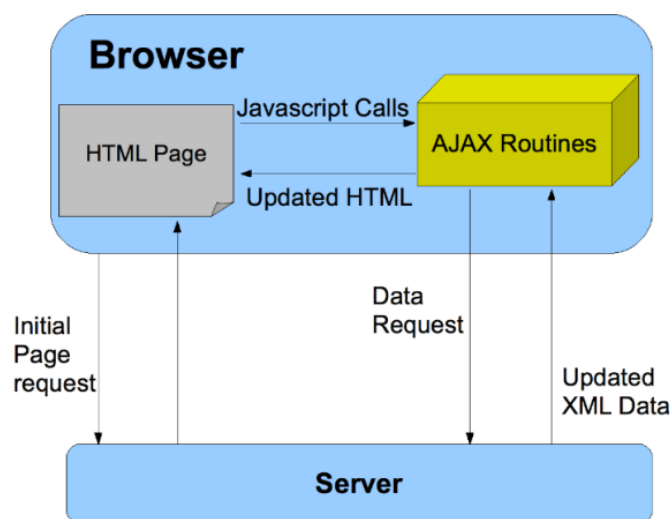


Figura 14 – Funcionamento do AJAX [18].

2.3.9. FRAMEWORK

Uma *framework* é uma abstração que tem como objetivo unir códigos entre vários projetos de *software*. É capaz de atingir uma funcionalidade específica, através da configuração durante a programação de uma aplicação. Ao contrário das bibliotecas, a *framework* é quem orienta o fluxo de controlo da aplicação.

Genericamente, a *framework* possui uma arquitetura do formato *MVC* [22]:

- **M** – *Model*, gere um ou mais elementos de dados, responde a perguntas sobre o seu estado e a instruções para mudar de estado. O modelo sabe o que a aplicação quer fazer, pois é a principal estrutura computacional da arquitetura, sendo o responsável pela modelação do problema a resolver.
- **V** – *View*, apresenta as informações para o utilizador através de gráficos e textos. O *view* recebe e mostra as instruções do controlo e as informações do modelo. Também comunica de volta com o modelo e com o controlador para relatar o seu estado.
- **C** – *Controller*, interpreta as entradas do rato e do teclado enviadas pelo utilizador, e mapeia essas ações em comandos, que são enviados para o modelo ou para a janela de visualização, para efetuar a alteração apropriada.

Na Figura 15 está representado o esquema *MVC*.

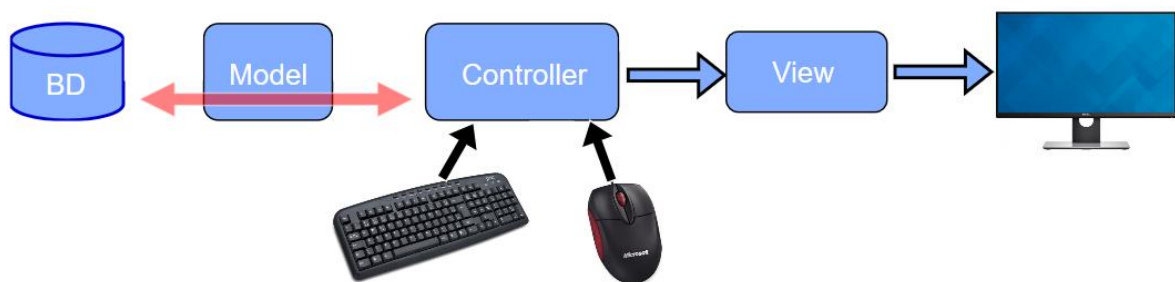


Figura 15 – Modelo *MVC* [22].

A *framework* possui as seguintes vantagens:

- Acelerar o processo de desenvolvimento;
- Ajuda na escrita para um código mais organizado, estruturado e reutilizável;
- Fornecer escalabilidade para seus projetos;
- Segue o padrão *MVC*;
- Estimular práticas de desenvolvimento *web* modernos, ou seja, programação orientada a objeto e outros paradigmas mais atuais;
- Maior segurança.

A *framework* utilizada no *CAFE* é o *Yii*, sendo esta uma ferramenta *open source* escrita em *PHP*. Lançado em 2008 na China, utiliza componentes para o desenvolvimento

de qualquer tipo de aplicações *web*. Permite a máxima reutilização de códigos na programação *web* e pode acelerar significativamente o processo de desenvolvimento.

Por ser uma *framework* leve e equipada com soluções em *caching*, é especialmente adequado para o desenvolvimento de aplicações com alto tráfego de dados, tais como: fóruns; portais; sistemas de *e-Commerce*; sistemas de gestão de conteúdo entre outros. Como boa parte das *frameworks PHP*, o *Yii* é uma *framework MVC* [20].

A *framework Yii* é eficiente, rica em recursos e possui uma boa documentação. É concebido, desde o seu início, para se ajustar a sérias aplicações *web* [18].

Possui como vantagens as seguintes características [22]:

- Moderno e flexível;
- Tem pacotes para autenticação e segurança;
- Tempo curto de desenvolvimento e aprendizagem;
- Fácil configuração para acelerar o desenvolvimento.



Figura 16 – Logótipo *Yii* [22].

2.3.10. *BOOTSTRAP*

O *Bootstrap* é uma *framework front-end* que se dedica ao desenvolvimento de aplicações *web* de uma forma rápida e fácil. É uma *framework* escrita em *JavaScript*, *HTML* e *CSS* para o desenvolvimento de *sites* e aplicações *web*. O *Bootstrap* permite que o utilizador desenvolva projetos *front-end web* de forma muito mais rápida, moderna, elegante e fácil [37].

Lançado nos Estados Unidos da América por *Mark Otto* e *Jacob Thornton* em 2011, é a ferramenta mais apropriada para dispositivos e projetos de todos os tamanhos (*desktop*, *tablet* ou *mobile*). É destinado a programadores com qualquer nível de conhecimento, sendo o *Bootstrap* usado neste momento por milhares de aplicações *web* em todo o mundo [38].



Figura 17 –Bootstrap era chamado inicialmente de *Twitter Blueprint* [38].

Entre as várias componentes que *Bootstrap* possui para uso, destacam-se as seguintes [38]:

- Ícones por *Glyphicons*: tratam-se de pequenos símbolos, ícones ou pictogramas que podem ser utilizados nas páginas *web*. Na Figura 18 é representado um exemplo de uma marcação típica para inserir um ícone: “``”;



Figura 18 – *Bootstrap*: geração de um lápis.

- *Dropdowns*: passa por uma funcionalidade que permite ao autor criar um efeito de abertura e fecho de submenus. Como representado na Figura 19, proporciona a criação de *menu dropdown* com uso de marcação típica e o mecanismo de abertura do menu pode ser definido de duas formas: diretamente com uso do par atributo/valor `data-toggle="dropdown"` ou codificando com uso do método *JavaScript dropdown()* previsto no *Bootstrap*;

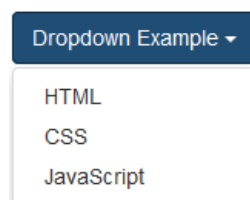


Figura 19 – *Bootstrap*: geração de uma lista.

- Grupo de *Inputs*: o *Bootstrap* prevê funcionalidades que permitem inserir texto de ajuda e botões, antes, depois ou mesmo em ambas as extremidades dos controles (campos) de formulário *input* para entrada de textos, como representado na Figura 20;

Name:

Password:

Figura 20 – *Bootstrap*: geração de *inputs*.

- Barra de navegação: permite criar uma grande variedade de barras de navegação, incluindo *links*, imagens, textos, caixas de busca, elementos de formulário, *dropdown*, entre outros. Estas barras podem ainda ser fixadas no topo ou no rodapé da aplicação. Na Figura 21 está representada uma barra de navegação;

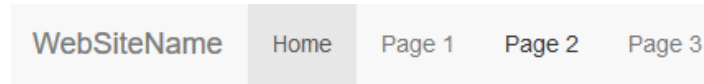


Figura 21 – Bootstrap: geração de barras de navegação.

- *Labels* (Rótulos): o *Bootstrap* tem a capacidade de criar rótulos para descrever, através da cor, uma ocasião especial, como: *label-default* (cinzento), *label-primary* (azul), *label-success* (verde), *label-info* (azul claro), *label-warning* (amarelo) e *label-danger* (vermelho). Estas são declaradas num elemento *span* contendo o rótulo. Na Figura 22 estão representados os vários tipos de rótulos mencionados.

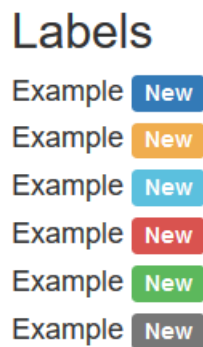


Figura 22 – Bootstrap: Geração de rótulos.

2.3.11. XMPP

Existem diversos serviços de mensagem no mundo da internet, destacando-se o *XMPP* (anteriormente chamado de *jabber*), serviço utilizado pela *IPBRICK*. Criado em 1998 por *Jeremie Miller*, o *XMPP* tem vindo a ganhar território em relativamente a outros serviços nos últimos anos, pois é uma tecnologia aberta e muito flexível.

É um protocolo aberto, extensível, baseado em *XML*, para sistemas de mensagens instantâneas. Foi desenvolvido originalmente para mensagens instantâneas e informação de presença formalizado pelo *IETF* [27].

O *XMPP* possui, entre outros, os seguintes serviços [26]:

- Encriptação de salas de *chat*;
- Autenticação;

- Lista de contactos;
- Mensagens entre uma ou várias pessoas;



Figura 23 – Em 2002 o *jabber* passou a chamar-se *XMPP* [27].

2.4. APACHE

O servidor *Apache* é considerado o servidor *web* mais usado na *internet*, tendo uma licença *open source*. É compatível com vários sistemas operativos como *Windows*, *Linux*, *Solaris* entre vários outros [45].



Figura 24 – O servidor *Apache* foi criado em 1995 por *Rob McCool* [28].

Para garantir segurança nas transações *HTTP*, este servidor dispõe do módulo “*mod_ssl*”, que dá a capacidade ao servidor de atender requisições utilizando o protocolo *HTTPS*, garantindo assim a segurança nos processos de *HTTP*. O protocolo *HTTPS* utiliza uma camada *SSL* para codificar os dados transferidos entre o cliente e o servidor, aumentando assim a segurança dos dados.

No diretório */etc/apache*, em ambientes *UNIX*, estão presentes os arquivos de configuração. A configuração do servidor pode ser executada por um ficheiro mestre com o nome “*httpd.conf*”, podendo haver configurações para cada diretório utilizando arquivos com o nome “*.htaccess*”, onde é possível, através do protocolo *HTTP*, autenticar um utilizador, usando uma combinação de ficheiro “*.htaccess*” com “*.htpasswd*”, que salvaguardará o nome dos utilizadores e das *passwords* [27].

2.5. LDAP

Numa rede de dados, é necessário haver um largo conjunto de serviços que ofereçam suporte a uma infraestrutura. Um dos serviços mais relevantes e que apoia outros tipos de serviços é o *LDAP*, um tipo de protocolo que serve para atualizar e pesquisar diretórios sobre a camada *TCP/IP* [29]. Como é possível verificar na Figura 25, o *LDAP* é compatível com vários sistemas operativos.



Figura 25 – O *LDAP* é compatível tanto no *Windows* como no *Linux* [30].

No *LDAP* são guardadas as informações dos utilizadores, como o: nome; apelido; idade; senha entre outros.

Um cliente inicia a sua sessão de *LDAP* ligando-se a um servidor *LDAP*, via *TCP*, onde por padrão a sua porta será 389. Este cliente começa a enviar pedidos para o servidor *LDAP*, o qual devolve as respostas. As operações básicas na troca de informação via *LDAP* são [30]:

- *Bind* – autentica e especifica a versão do protocolo *LDAP*;
- *Search* – procura ou recupera entradas dos diretórios;
- *Compare* – testa se uma entrada tem determinado valor como atributo;
- *ADD* – adiciona uma nova entrada;
- *Delete* – apaga uma entrada;
- *Modify* – modifica uma entrada;
- *Modify DN* – move ou renomeia uma entrada;
- *Start TLS* – protege a conexão com a *Transport Layer Security (TLS)*;
- *Abandon* – aborta uma requisição prévia;
- *Extended Operation* – operação genérica para definir outras operações;
- *Unbind* – fecha a conexão.

Na Figura 26 é demonstrado o papel do *LDAP* na troca de informações:

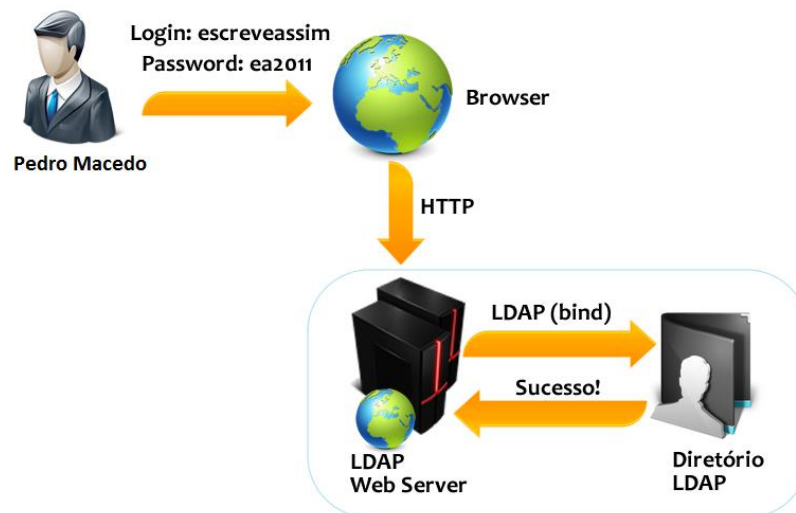


Figura 26 – Papel do *LDAP* no mundo da *web* [30].

A *IPBRICK* utiliza esta rede estruturada, usando o serviço *LDAP*, que guarda todas as informações do utilizador e que permite a autenticação com melhor controlo de acesso e segurança. Em suma, é possível definir o *LDAP* como um serviço de “autenticação centralizada”.

2.6. *SAMBA*

O *Samba* é uma plataforma para *Linux* que tem como objetivo unir três tecnologias: *Windows*, *Linux* e *LAN*. Passa por uma aplicação que é executada no lado do servidor. Mesmo podendo ser executada numa plataforma completamente diferente à do *Windows*, o *Samba* comporta-se como este. Por outras palavras, o *Samba* comunica com o *Windows*.

Com este servidor, é possível partilhar arquivos, impressoras e controlar o acesso a certos recursos de rede com eficiência semelhante aos servidores baseados em sistemas operacionais da *Microsoft*, sendo compatível com todas as versões do *Windows*, desde *NT 4.0* até *Windows 10*, existindo empresas, como a *IPBRICK*, que usam o *Samba* como solução para conflitos existentes entre diferentes versões do *Windows*.

O *Samba* está protegido com grande segurança, pois existe um grande rigor nos controlos dos recursos oferecidos, permitindo que a sua configuração seja feita remotamente. Para os casos mais críticos, o administrador da rede ser informado de anormalidades por *email* [31].



Figura 27 – A plataforma *Samba* foi criado por *Andrew Tridgell* em 1992 [31].

2.7. NETBEANS

A plataforma *NetBeans IDE* trata-se um meio de desenvolvimento gratuito e *open source* para criadores e programadores de *softwares* nas linguagens *C*, *C++*, *HTML*, *PHP*, *CSS*, *Java*, entre outros. O ambiente pode ser executado nas mais variadas plataformas, como em *Windows*, *Linux*, *MacOS* e *Solaris*. Para além disso disponibiliza as ferramentas necessárias para criar aplicações não só a nível de *desktop* como a nível de aplicações para *mobile* [32]. A rede social *CAFE* utiliza este meio de desenvolvimento para a realização do seu código.



Figura 28 – O *NetBeans* foi criado em 1996 [34].

Em 1996 o *NetBeans* foi criado com o nome *Xelfi*, na República Checa, como um projeto académico, tendo como objetivo escrever um *Java IDE* semelhante ao *Delphi* no *Java*. O *Xelfi* foi o pioneiro em *Java IDE* escrito originalmente em *Java*, revolucionando o mundo dos programadores, sendo este usado por milhões em todo o mundo [44]. Acabou por ser um projeto que atraiu o interesse não só os estudantes bem como os desenvolvedores de *software*. Depois dos criadores do *Xelfi* concluírem o seu percurso académico, decidiram que o poderiam vender como um produto comercial, mudando o nome para *NetBeans*. Ao longo dos anos foram sendo lançadas novas versões, mais modernas e com mais *plugins*.

O *NetBeans* ajuda os programadores a desenvolver, compilar e a fazer *debug* do código, bem como a instalar aplicações. Este *software* foi criado com o intuito de simplificar o desenvolvimento e melhorar a eficiência do programador. Apesar de ser um *software* desenvolvido em *Java*, é compatível com qualquer outra linguagem de programação, como por exemplo *C*, *C++*, *Ruby*, *PHP*, *XML* e *HTML*.

O *NetBeans* oferece apoio na criação de projetos e módulos, usufruindo de um conjunto de bibliotecas, módulos e aplicações, de uma forma organizada. Estes recursos ajudam o programador a escrever e desenvolver a sua aplicação de uma forma mais rápida e eficiente.

Neste momento estão disponíveis várias línguas, facilitando o acesso a principiantes no campo da programação, possibilitado o desenvolvimento de aplicações em várias línguas [33].

O *Netbeans* possui as seguintes capacidades e recursos [35]:

- Editor de código fonte integrado, recursos de grande dimensão para aplicações *web* e *API Java* para interfaces gráficas;
- Apoio ao *Java Enterprise Edition*, plataforma de programação dirigida a aplicações multicamadas, baseadas em componentes que são executados em um servidor de aplicações;
- *Plugins* para *UML (Unified Modeling Language)*, linguagem de modelação de desenvolvimento remoto em equipas;
- Possui uma interface amigável com *CVS (Sistema de Versões Concorrentes)*. É um sistema de controlo de versão que permite que se trabalhe com diversas versões de arquivos organizados num diretório;
- *CSS*, algumas funcionalidades para editar folhas de estilos como destaques e análise de código;
- Ajuda interna e *online. Debug* apurado de aplicações e componentes;
- Correção automática, úteis para soluções de erros de código. Atalhos para copiar linhas inteiras de código;
- Integração de módulos, suporte à base de dados, ferramenta de automatização da construção de programas e *TOMCAT*, servidor de aplicações *Java* para *web*;
- Geração de *Javadoc*, ferramenta que permite a geração automática de arquivos *Javadoc* em *HTML* com base nos comentários inseridos no código, com recursos que tornam mais fácil a inclusão de comentários no código mais fácil;

- Integração de módulos;
- Atalhos para copiar linhas inteiras de código.

O *Netbeans* possui alguns módulos úteis para o programador, destacando os seguintes:

- *NetBeans profiler*: ferramenta que monitoriza as aplicações de *Java*. Este ajuda o programador a acelerar a rapidez do *software*.
- Ferramenta de desenho para a interface: mais conhecido como *Matisse*, que permite ao programador criar e desenvolver interfaces.
- Editor de *JavaScript*: oferece suporte para código em *JavaScript*, *AJAX* e *CSS*, com aviso de erros de sintaxe, recomendação de variáveis e, caso o programador prefira, completção automática do código.

2.8. *ARCHIPEL*

O *Archipel* é um gestor e supervisor de máquinas virtuais executadas sob *KVM*, *Xen*, *OpenVZ* o *VMware*. Consegue gerir até centenas de máquinas virtuais, tendo a capacidade fazer migrar máquinas virtuais, *VMCASTS*, pacotes e muito mais. Toda a comunicação é feita sob *XMPP*, não possuindo nenhum serviço *web* ou protocolo personalizado. Para ser executado, é apenas necessário um servidor *XMPP*, como o *eJabberd* [36]. A *IPBRICK* utiliza o *Archipel* para a supervisão das suas máquinas virtuais.



Figura 29 – O *Archipel* é compatível com vários sistemas operativos [36].

O *Archipel* possui as seguintes interfaces [36]:

- Controlos – Permite ligar, pausar, parar, forçar a paragem ou reiniciar a máquina virtual desejada;

- Definições – Definir o nome e as características da máquina (nome, tipo de domínio, memória, número de *CPUs*, ciclo de vida, entre outros);
- *Appliances* – É possível emular um dispositivo de *hardware* separado com *software* integrado (*firmware*), especificamente projetado para fornecer um recurso de computação específico;
- *Disks* – Tem como objetivo criar, editar ou eliminar uma *drive*;
- *Snapshots* – Guarda o estado atual da máquina virtual. Caso haja problemas com a máquina virtual, é possível reverter para qualquer gravação feita anteriormente;
- Consola *VNC* – Nesta secção é exibida a interface entre o utilizador e a máquina, sendo aqui que o utilizador consegue manipular a máquina virtual;
- Calendário – Tem como objetivo marcar uma hora para uma ordem específica à máquina virtual, como desligar-se sozinha, destruí-la, reiniciar entre outras;
- *Chat* – À medida que são executadas ações na máquina virtual, o *Archipel* envia ao utilizador mensagens dessas suas ações;
- Permissões – Caso o utilizador seja administrador de várias máquinas virtuais, este terá a capacidade de fornecer ou tirar permissões a outros utilizadores.

2.9. GROUPWARE

O *Groupware* é uma plataforma que facilita a realização de tarefas em grupo, independentemente da distância a que os elementos do grupo se encontram [43].

Esta plataforma possui os seguintes suplementos:

- Correio eletrónico: torna-se possível enviar *emails* não só entre os vários elementos do grupo, mas também a elementos externos.



Figura 30 – *Groupware: Email*.

- Agendar eventos: marcar um evento para um dia e hora específica;

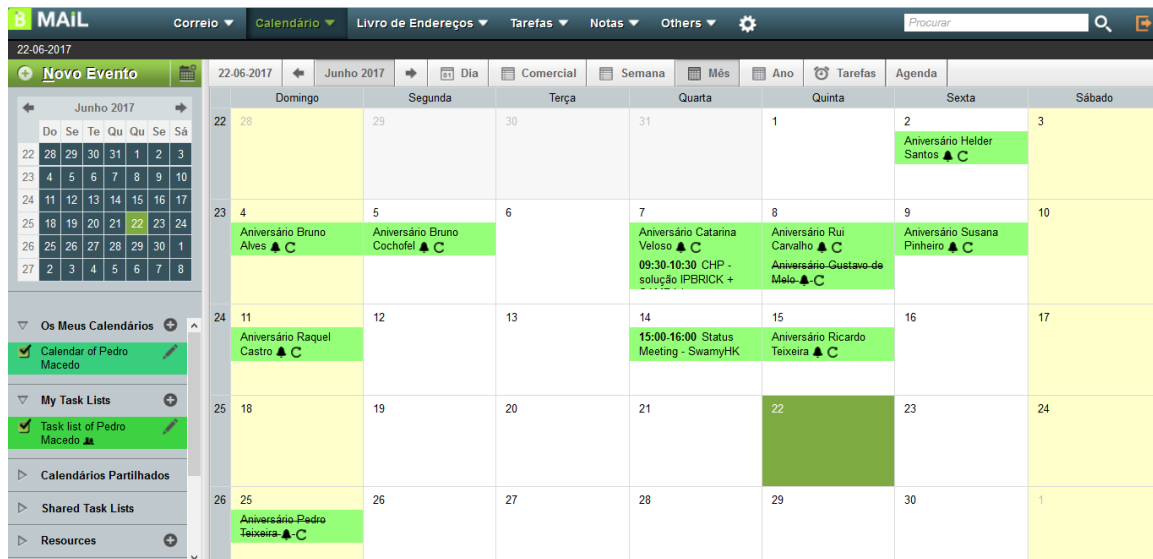


Figura 31 – Groupware: Calendário.

- Guardar lista de contactos: possui um livro de endereços com os dados dos utilizadores do grupo;



Figura 32 – Groupware: Livro de endereços.

- Tarefas: é possível definir e partilhar as tarefas atribuídas a cada membro do grupo;

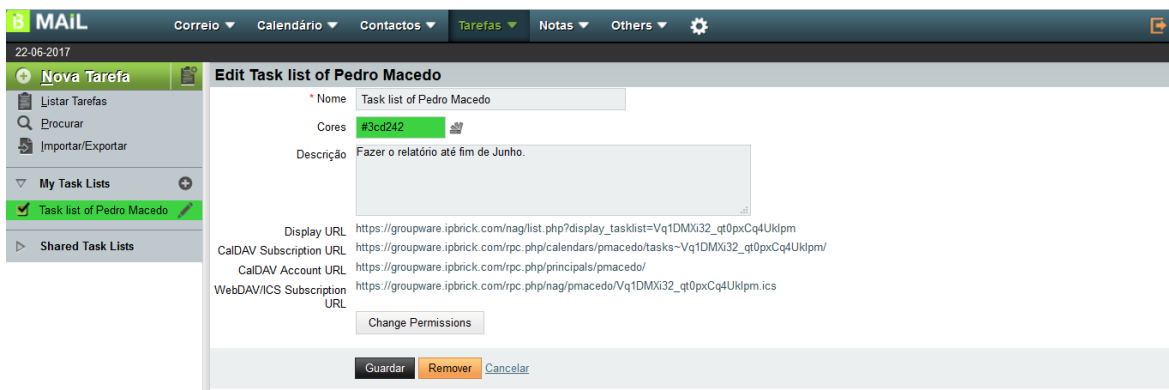


Figura 33 – Groupware: Criar uma tarefa.

- Notas: bloco de notas para o utilizador tomar notas do que desejar;

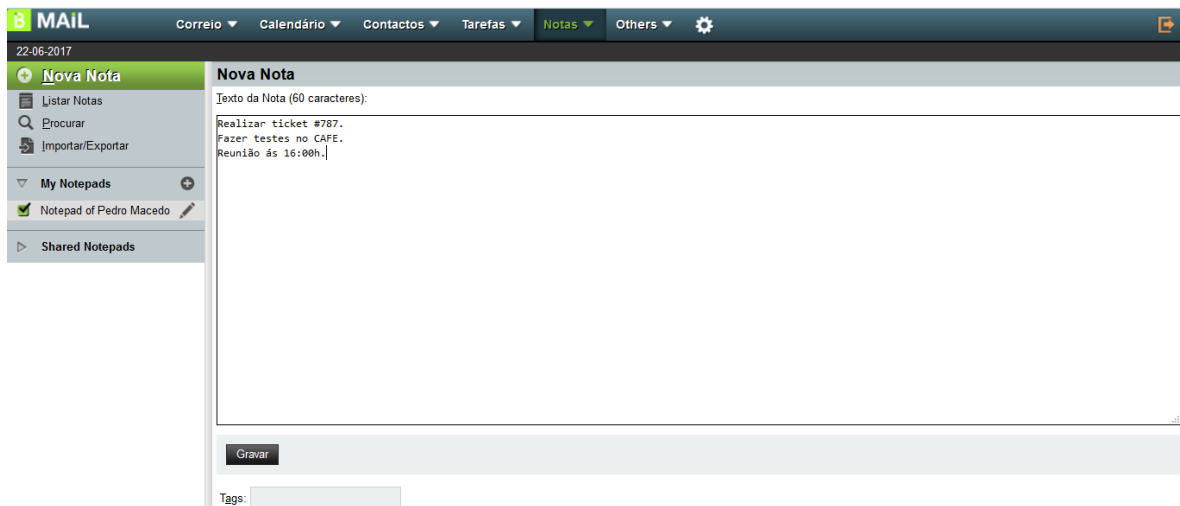


Figura 34 – Groupware: Bloco de notas.

- Gerir ficheiros: *cloud* que armazena ficheiros guardados pelo utilizador.

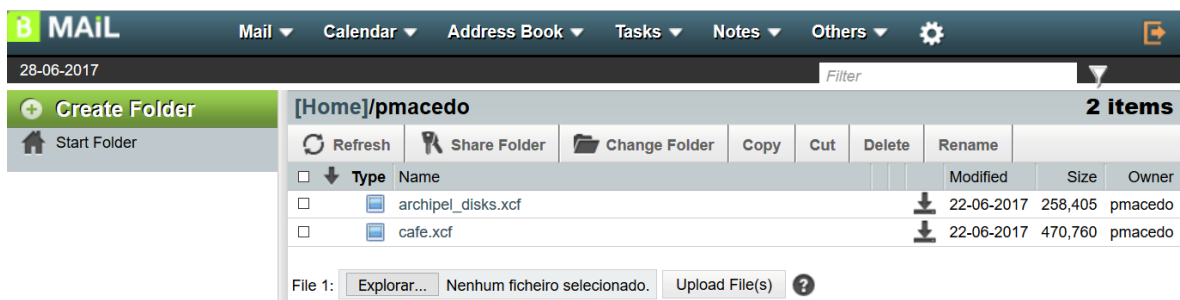


Figura 35 – Groupware: Gestão de ficheiros na *cloud*.

2.10. IPBRICK OS

A *IPBRICK OS* é uma plataforma de comunicações para sistemas de servidores corporativos, sendo baseado em *Linux Debian*. Esta utiliza *software open source* para gerir:

- Comunicações Unificadas;
- Gestão de Documentos e Processos;
- *Email* e Ferramentas colaborativas;
- Rede Social Corporativa.

A *IPBRICK OS* permite que uma pessoa não especializada configure um servidor de última geração, em qualquer uma das anteriores áreas [5].

Este *software* fornece uma integração entre pacotes de *software open source* anteriormente configurados, nunca perdendo configurações relevantes que façam o servidor funcionar de modo instável.

A integração não se limita apenas a serviços da *IPBRICK OS*, aplica-se também a *software* de terceiros que são completamente compatíveis com a *IPBRICK OS*, como por exemplo o livro de endereços contido na base de dados do servidor, o qual é utilizado por todos os serviços da *IPBRICK* e *software* de terceiros.

A *IPBRICK OS* está também preparada para uma evolução suave, sendo esta projetada para simplificar a evolução entre as várias versões, não tendo o utilizador de saber sobre mudanças significativas a nível do sistema operativo *Linux Debian*, bem como versões de pacotes de *software* e versões de sintaxes do ficheiro de configuração. Passa por uma solução que permite uma instalação e processos de configuração rápidos, tendo uma capacidade de recuperação de desastres a nível corporativo [6].

Para inicializar a Tese na *IPBRICK*, achou-se necessário e útil dar a conhecer este sistema operativo, cujas características serão descritas neste capítulo, através do estudo de vários guiões disponibilizados pela empresa.

2.11. CAFE – CAPACIDADES E FUNCIONALIDADES

O *IPBRICK.CAFE* é fácil de usar, simples de adotar, eficaz na redução de tráfego de *intranet* e a melhor maneira de promover o envolvimento dos colaboradores no espírito da empresa. Na Figura 36 está representada a página principal do *CAFE*. Esta possui uma área onde é possível fazer publicações, criar votações e notícias, aceder a *links* para todas as aplicações com sistema *Single Sign-On*, bem como criar álbuns de fotografias que pode partilhar com todos os colegas.

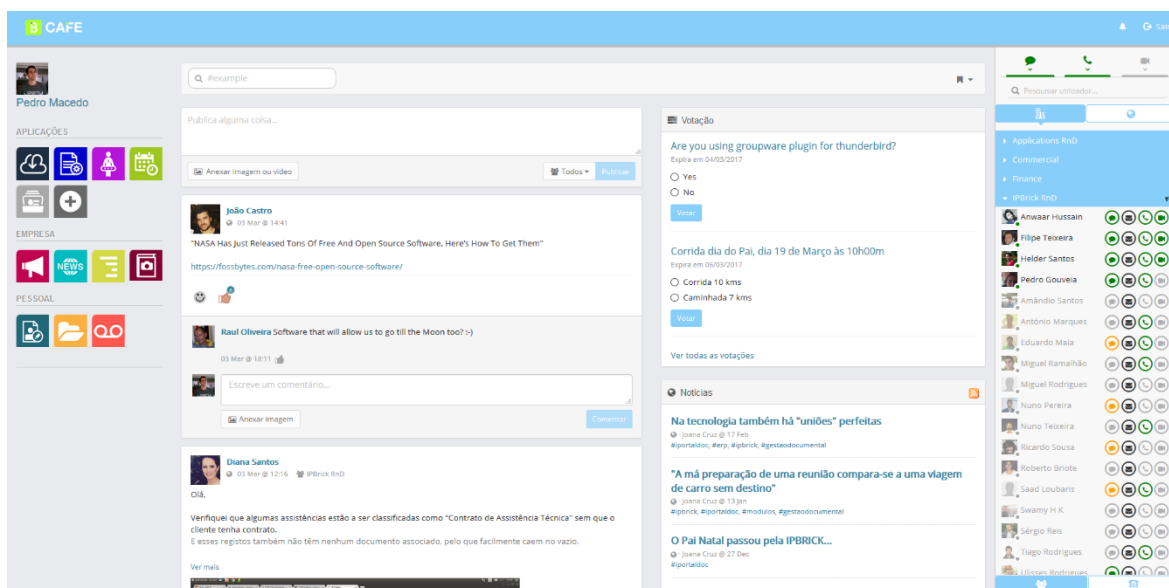


Figura 36 – CAFE: Página principal.

A área pessoal do CAFE permite fazer alterações no perfil do utilizador, tais como alterar a fotografia, os dados pessoais, bem como as definições de *VoIP* (*Voice over Internet Protocol*). Exemplo é a ativação do “*follow me*”, para que o utilizador esteja sempre contactável mesmo se estiver fora do escritório. Por fim, é possível personalizar o *layout* da página principal do CAFE.

A área social do CAFE permite partilhar todo o tipo de informação com os colegas ou um grupo específico de trabalho. É possível fazer uma pesquisa de uma publicação, ativando sempre um alerta caso haja uma nova publicação ou uma nova reação a um *post* criado pelo utilizador.

Possui a funcionalidade de “*bookmark*”, permitindo que apenas num clique se possa aceder aos *sites* que o utilizador mais visita. Existe uma área aplicacional que permite aceder através de atalhos às aplicações *web* usadas no dia-a-dia. Como o CAFE integra com a aplicação de gestão de documentos e processos da IPBRICK, o *iPortalDoc*, o utilizador pode facilmente aceder a esta aplicação e partilhar qualquer documento diretamente do gestor documental para o CAFE.

Para aceder às comunicações unificadas é necessário interagir com a *UcoIPBar* (ver lado direito da Figura 36), da qual permite comunicar em tempo real com contactos internos e externos da organização, através de voz, vídeo, *email* e *chat* e ainda definir o estado para as comunicações (disponível, ausente ou ocupado).

Na *UcoIPBar* estão gravados os contactos dos colegas de trabalho, podendo entrar em contacto com eles clicando do ícone ao lado do seu nome, através dos modos de comunicação referidos em cima.

2.12. *CAFE* – ESTRUTURA DO CÓDIGO

O código fonte do *CAFE* encontra-se organizado nas seguintes pastas:

- **AppMobile** – Contém partes da versão do *CAFE* para o telemóvel, disponível para *Android* e *IOS*;
- **updateDB** – Pasta responsável por alterar as definições na base de dados, de fazer a ligação entre o *CAFE* e a base de dados e por guardar todos os dados submetidos no *CAFE* pelo utilizador;
- **themes** – Pasta que possui o ficheiro “*update_system.PHP*” que cria e guarda as imagens que vão sendo guardadas no *CAFE*;
- **sounds** – São guardados alguns sons emitidos no *CAFE*, como por exemplo o som de notificação de mensagem no *chat*, o som de chamada do telefone ou a marcação de número de telefone na *UcoIPBar*.
- **scripts** – Contém o ficheiro que dá acesso aos servidores do *Apache*;
- **rootsys** – Ficheiros de sistema da *IPBRICK*;
- **protected** – Esta é uma das pastas mais importantes do *CAFE*, contendo os diretórios e ficheiros de *layout*, bem como no funcionamento e mecanismo de quase toda a estrutura do *CAFE*. Este diretório contém as seguintes pastas:
 - **commands** – Guarda as notificações do historial do *chat* e das notificações do *email*;
 - **components** – Tem o ficheiro de controlo do *CAFE* (como permissões, traduções, *login* e ações administrativas), do servidor *web* da *IPBRICK*, do servidor *LDAP* e do servidor *samba*;
 - **config** – Configura a *framework Yii* para o *CAFE*;
 - **controllers** – Vários ficheiros de responsáveis por controlar cada secção do *CAFE* (álbum de fotografias, aplicações, autorização de servidores, links, contactos, notícias, votações, perfil, histórico, entre outros);
 - **messages** – Contém as traduções do *CAFE* (para Português, Inglês, Castelhanho e Francês);

- **models** – Pasta que faz a ligação entre a interface entre controlador e interface do *CAFE*;
- **views** – Pasta que fornece ficheiros do layout das várias páginas do *CAFE*;
- **MakePackages** – Diretório que contém os pacotes de atualização do *CAFE*;
- **log** – Pasta que contém os *logs* antigos, arquivando-os e compactando-os para que consumam menos espaço no disco, mas que ainda estão disponíveis para inspeção (se necessário);
- **js** – Funcionalidades adicionais incorporadas no *CAFE*. É o caso do editor de texto *ckeditor* para a criação de notícias ou o uso da *webcam* para as videoconferências;
- **images** – Aqui são guardadas todas as imagens que aparecem na aplicação do *CAFE*;
- **fonts** – Repositório que guarda coleções de vários tipos de ferramentas criadas pelo programador (como por exemplo botões e *checkboxes*) para a criação de componentes úteis para *design* de uma interface *web*. É o caso do *Glyphicon* e do *Bootstrap*;
- **CSS** – Diretório que possui ficheiros com as classes do estilo de todas as interfaces do *CAFE*: tipos e tamanho de letra; posição de uma componente; cor da interface entre outras opções;
- **brand** – armazena as imagens dos logótipos de marcas que fazem parceria com a *IPBRICK*;
- **upload** – Diretório responsável na gestão de uploads do *CAFE*, tanto a nível de ficheiros, como de imagens.

2.13. TIPOS DE TESTE

De modo a testar a aplicação *CAFE*, discutiu-se qual o melhor método a utilizar na verificação da funcionalidade. Para tal, foram reunidos três possíveis métodos [42]:

- *Black box*: são fornecidas as informações do cliente sobre a rede, servidores, banco de dados e sistemas que estão envolvidos no intuito do teste de invasão, de modo a que sejam realizados testes extensivos e com mais alcance.
- *White Box*: são realizados maioritariamente para analisar aplicações *web*, cuja configuração do servidor e do código-fonte são examinados abertamente com o objetivo de detetar falhas de segurança que arrisquem comprometer a boa funcionalidade do serviço.

- *Grey box*: pode ser considerado uma mistura entre a *black box* e *white box*. O analista de teste recebe informações do cliente, como por exemplo os dados da infraestrutura da rede ou o acesso a um determinado serviço *web*. É *Grey box* é orientada para a análise de possíveis falhas de segurança numa aplicação vinda através de um utilizador credenciado, como níveis de permissões de acesso e alterações desautorizadas.

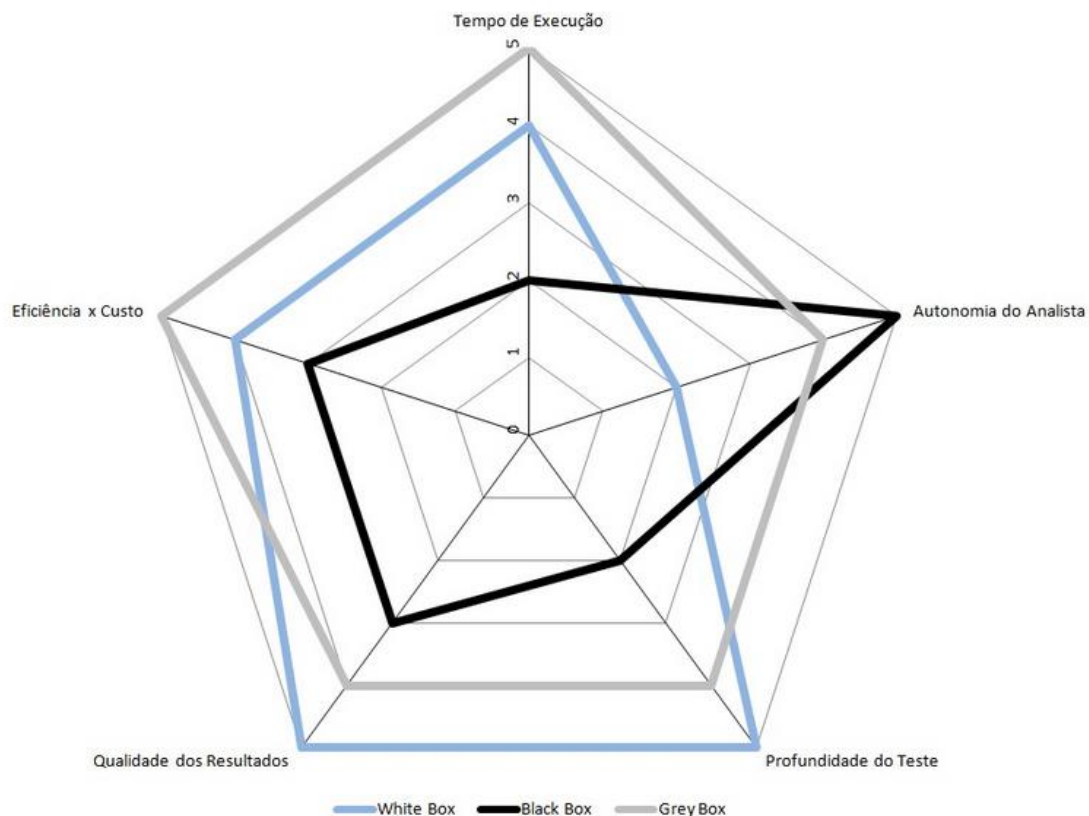


Figura 37 – Performance de cada tipo de teste [42].

Nos testes do *CAFE* é usada o tipo *black box*, pois este é o procedimento que a *IPBRICK* acha mais adequado para o teste de qualidade desta aplicação.

2.14. *CERVISIA*

O *Cervisia* é um programa que tem como objetivo trabalhar com diferentes versões de arquivos ao mesmo tempo. Este implementa funcionalidades comuns para: adicionar, remover e cometer ficheiros; importar e conferir módulos; adicionar ou remover vigilâncias; editar ficheiros; verificar ficheiros com erros e resolução de conflitos. Para além das funcionalidades referidas, este possui interfaces que incluem a visualização das versões dos ficheiros em forma de árvore e as listas das alterações dos ficheiros, estado de ficheiros codificado por cores, e diferenciação gráfica entre versões [46].

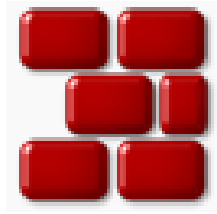


Figura 38 – O *Cervisia* foi lançado em 2015 por Bernd Gehrman [46].

2.15. CONCLUSÃO

Este capítulo teve como objetivo apresentar as tecnologias e outras informações úteis a utilizar no Capítulo seguinte. Adquirir conhecimentos na área da tecnologia *web* foi um passo fundamental nesta Tese, ao fornecer as bases para o desenvolvimento de novos módulos no *CAFE*.

3. *IPBRICK* CAFE - DESENVOLVIMENTO

Neste capítulo serão descritos os melhoramentos e a adição de módulos mencionados nos objetivos. Para tal foi necessário, em primeiro lugar, estudar o formato do *CAFE* e a sua composição. De seguida, analisar quais as falhas e estudar possíveis soluções para o melhoramento do mesmo. Logo depois, de modo a corrigir as falhas, prosseguiu-se para a edição e inserção de código e, por fim, procedeu-se a testes para verificar se a versão melhorada funciona de forma correta.

3.1. MELHORAMENTOS NA INTERFACE

Um dos principais objetivos a atingir passa pelo melhoramento das funcionalidades no *CAFE*, não só a nível de funcionalidade, mas também como a nível de *design*. Nos próximos subcapítulos serão referidas as várias tarefas propostas na realização da Tese, e respetiva resolução.

3.1.1. CRIAÇÃO DE MÓDULO/MELHORAMENTOS NO ÁLBUM

Como referido no Capítulo 2, o *CAFE* tem a capacidade de criar um ou mais álbuns de modo a guardar as fotografias recebidas por *upload*. Porém, na escolha do álbum, este não tinha a capacidade de escolha de capa. Além deste problema, depois de criado um álbum, o *menu* fica com a cor de *default* do *CAFE* (azul escuro), em vez de ficar com a cor do cabeçalho escolhida pelo utilizador. Por fim, as permissões do álbum não surtem qualquer efeito, ou seja, o utilizador tem sempre a capacidade de criar um álbum de fotografias mesmo que não tenha permissões para isso.

De modo a resolver estas lacunas, achou-se necessária a criação desta funcionalidade de modo a dar ao criador do álbum a escolha da sua capa. Na versão anterior, a fotografia de capa era escolhida pelo sistema de uma forma aleatória.

Para a concretização da tarefa de colocação de fotografia de capa, realizaram-se os seguintes passos:

1. Criação do botão de seleção para a fotografia de capa:

Para tal, foi necessário recorrer ao *Font Awesome*, utilizando um dos ícones disponíveis nesta ferramenta (ver Figura 39).

```
<button id="cover-<?php echo $picture->album_id."-
".$picture_jpg[0];?>" type="button" class="btn btn-xs
selectfile" title="<?php echo Yii::t('app', 'Cover picture'); ?>"
onclick="selectFile('<?php echo $picture->album_id;?>', '<?php echo
$picture->path;?>');">
    <i class="fa fa-photo" ></i>
</button>
```



Figura 39 – *Font Awesome*: ícone de fotografia.

Para colocar o ícone com melhor gosto, decidiu-se mudar a cor do mesmo através da criação de uma classe, com o nome "selectfile". Esta classe é dividida em dois formatos: com transparência a 90%, caso o rato não esteja por cima na fotografia, ou com nenhuma transparência, caso o rato esteja em cima da fotografia:

```

.selectfile {
    // Azul claro
    background-color: #3399ff;
    // Transparência a 90%
    opacity:0.1;
}
.selectfile:hover {
    background-color: #3399ff;
    // Sem transparência
    opacity:1;
}

```

2. Tabela do álbum:

O *CAFE* possui na base de dados uma tabela com o nome "album_pictures", onde são guardadas todas as imagens. Na Figura 40 é possível verificar as imagens existentes nos álbuns.

id	user_id	album_id	path	created_on
1	2	3	images/albums/3/1491284011_download.jpg	2017-04-04 06:33:30
2	2	3	images/albums/3/1491284011_Nithf7m.jpg	2017-04-04 06:33:30
3	2	3	images/albums/3/1491284011_tiger.jpg	2017-04-04 06:33:30
4	2	3	images/albums/3/1491284011_wallpapers-da-savana-africana.jpg	2017-04-04 06:33:30
9	2	2	images/albums/2/1492774939_junglegames.jpg	2017-04-21 12:42:19
10	2	2	images/albums/2/1492774940_513793-Paisagens-da-África-fotos-4 (1).jpg	2017-04-21 12:42:19
11	2	2	images/albums/2/1492774940_cascata.jpg	2017-04-21 12:42:19
12	2	2	images/albums/2/1492774940_african-safari-sunset (1).jpg	2017-04-21 12:42:19
13	2	2	images/albums/2/1492774940_sun (1).png	2017-04-21 12:42:19
14	2	1	images/albums/1/1496466867_caminho.jpg	2017-06-03 06:14:27
15	2	1	images/albums/1/1496466867_luis.jpg	2017-06-03 06:14:27
16	2	1	images/albums/1/1496466867_estádio.jpg	2017-06-03 06:14:27
17	2	1	images/albums/1/1496466867_tigre.jpg	2017-06-03 06:14:27
18	2	1	images/albums/1/1496466867_Torre_Belém_April.jpg	2017-06-03 06:14:27

(14 rows)

Figura 40 – Lista de imagens.

3. Guardar o nome da fotografia numa variável

A estratégia para colocar a fotografia de álbum à escolha do utilizador passou por criar uma variável que guardasse o último parâmetro de cada linha do "path".

```

// Sempre que encontra uma "/" na coluna path, separa esta em
//vetores
$picture_name= explode ('/', $_picture->path);

```

A variável `$picture_name` resultou no seguinte:

```
Array
(
    [0] => images
    [1] => albuns
    [2] => 1
    [3] => 1492774916_wallpapers-da-savana-africana.jpg
)
```

De seguida, segue-se o mesmo passo para isolar apenas o nome da fotografia:

```
//Sempre que encontra um "." no vetor de posição [3],
//separa estes em vetores
$picture_jpg= explode('.', $picture_name[3]);
```

A variável `$picture_jpg` resultou no seguinte:

```
Array
(
    [0] => 1492774916_wallpapers-da-savana-africana
    [1] => jpg
)
```

4. Substituição de variáveis.

Substituiu-se a variável `$default_pic`, que guardava a fotografia de capa na versão anterior (fotografia aleatória), pelo nome fotografia escolhida pelo utilizador, `$picture_jpg` [0]:

```
//Para cada álbum
foreach ($albums_array as $album){
//Colocar nova foto de capa
if($album->id=== $model->id){
    $default_pic= $album->photo_default;
    $default_pic=$picture_jpg[0];
}
```

Função *AJAX* que coloca a capa no álbum:

```
$.ajax({
type: 'post',

//Função que lê a fotografia selecionada
url: '<?php echo Yii::app()->createUrl('/album/selectPicture');
?>',
```

```

data: {
  album_id: album_id,
  url: url
},
// Caso seja selecionada a fotografia de capa, não haverá
//transparência
success: function (){
  $("#cover-"+album_id+"-
"+url.split("/") [3].split(".")[0]).animate({
    opacity: 1,
  }, 80, function() {
  });
  ...
});
}
});

```

5. Verificar resultados

Para confirmar se a fotografia de capa ficou guardada com sucesso na base de dados, é necessário verificar no *CAFE* se tudo se encontra a funcionar.

As Figuras 41 e 42 mostram a seleção de capa e respetivo resultado (fotografia de álbum para a capa com o nome "Teste 1").

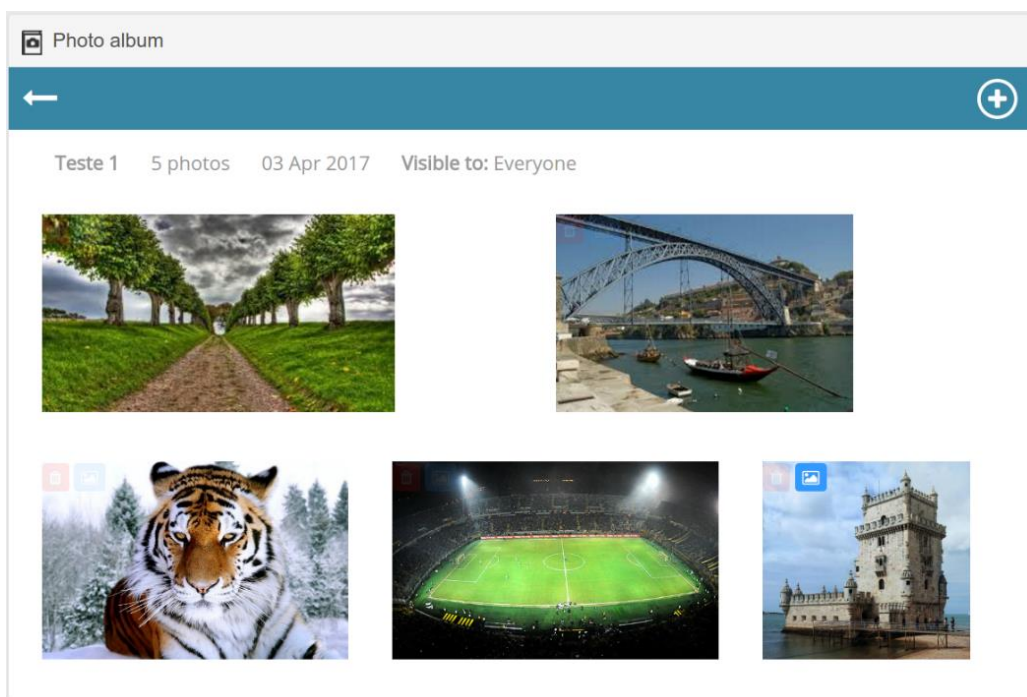


Figura 41 – Lista de fotografias do álbum, com a imagem do canto inferior direito selecionado como capa (Torre de Belém).

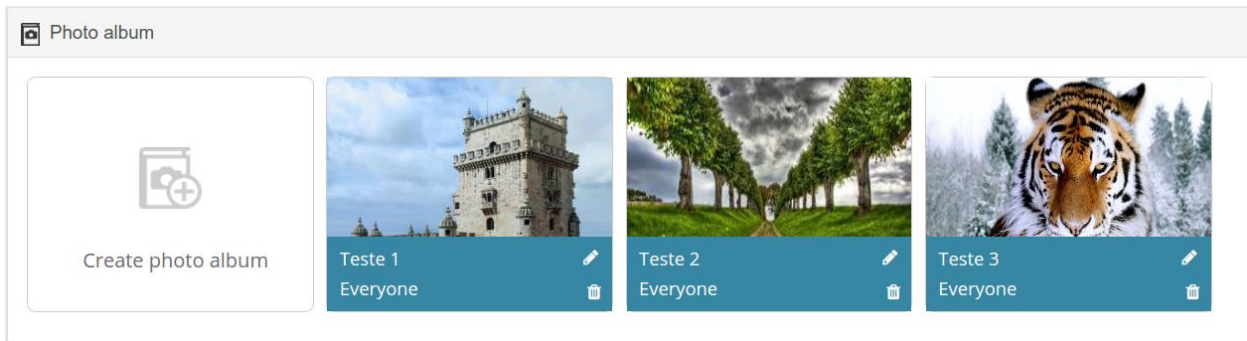


Figura 42 – Imagem ficou gravada no álbum do Teste 1.

Para resolver a lacuna da cor ser sempre a de *default* do *CAFE* quando um álbum é criado (ver Figura 43), foi necessário criar um pedaço de código para colocar a cor do cabeçalho. Foi necessário criar um *id* (chamado “divx”) e definir a cor de fundo como a cor de cabeçalho. Na Figura 44 está representado o álbum com a cor correta.

```
#divx {
  background-color: <?php echo $layout_config['header']['bgcolor']?>;
}
```

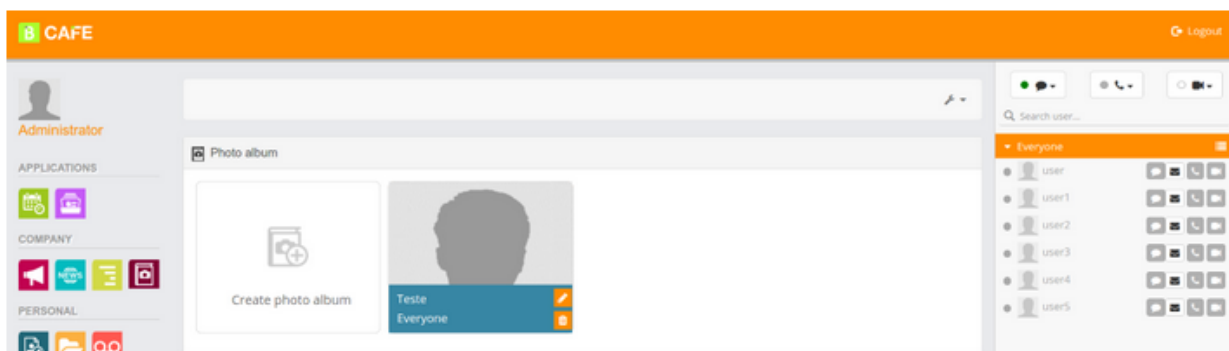


Figura 43 – Cor default no menu do álbum.



Figura 44 – Cor de fundo do menu do álbum igual à do cabeçalho.

Por fim, para resolver o problema da permissão nos álbuns, foi necessário chamar as seguintes linhas de código para as funções de criar/alterar/apagar álbuns:

```
Yii::app()->controller->checkPermissions('album', 'update')
Yii::app()->controller->checkPermissions('album', 'createAlbum')
```

```
Yii::app()->controller->checkPermissions('album', 'deleteAlbum'))
```

Na Figura 45 é possível verificar, na coluna “role_id”, o valor 1 na base de dados, o que significa que o efeito da permissão dos álbuns passou a estar ativo.

id	module	action	role_id
71	album	create	1
73	album	update	1
74	album	delete	1
75	album	addPictures	1
77	album	deletePictures	1

Figura 45 – Valor 1 é atribuído na base de dados.

3.1.2. MELHORAR O ESQUEMA *ENABLE/DISABLE* NA ÁREA DOS UTILIZADORES

A área de administração é restrita apenas para os utilizadores com maiores privilégios no *CAFE*. Esta área é dividida em vários setores, sendo um deles dedicado à gestão dos utilizadores. Neste setor foi aperfeiçoado o modo de associação entre o nome do utilizador e o seu perfil. Na versão anterior (ver Figura 46), era possível modificar a opção do botão *Dropdown Menu* (para a escolha do perfil) e carregar no botão bloqueado (🚫) ou desbloqueado (✅) e submeter, mesmo se nenhum utilizador estivesse selecionado na *checkbox*, o que não faz sentido. Nesta nova versão, esses botões só ficam desbloqueados caso seja primeiro selecionada a *checkbox* de pelo menos um utilizador. Para além desta mudança, a secção de apresentar as fotografias dos utilizadores foi transferida para a secção Configurações → Definições de *Layout*.

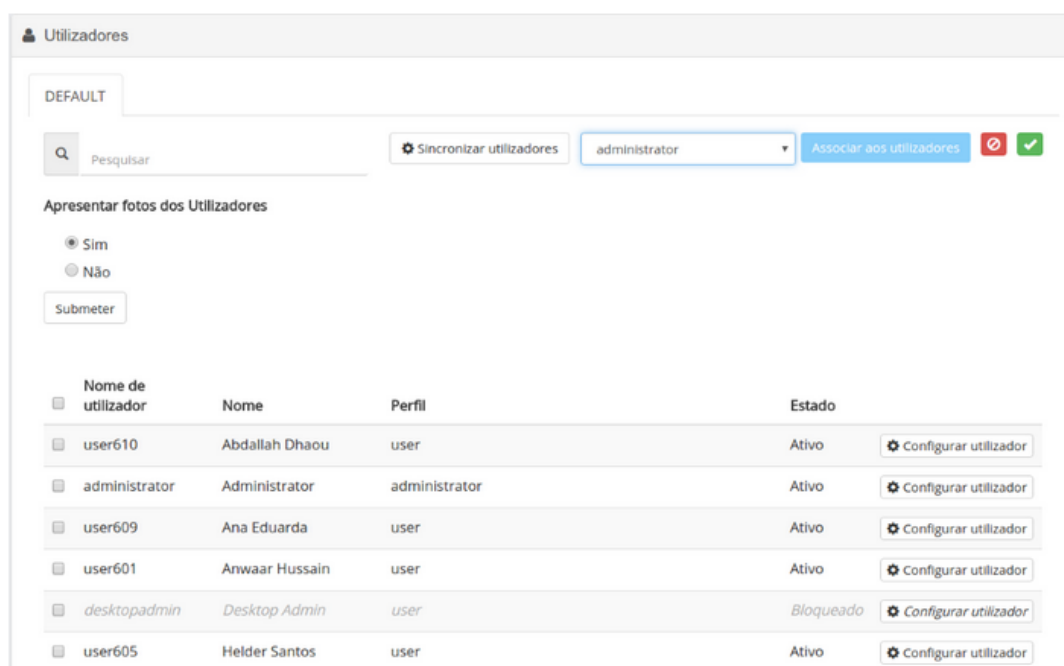

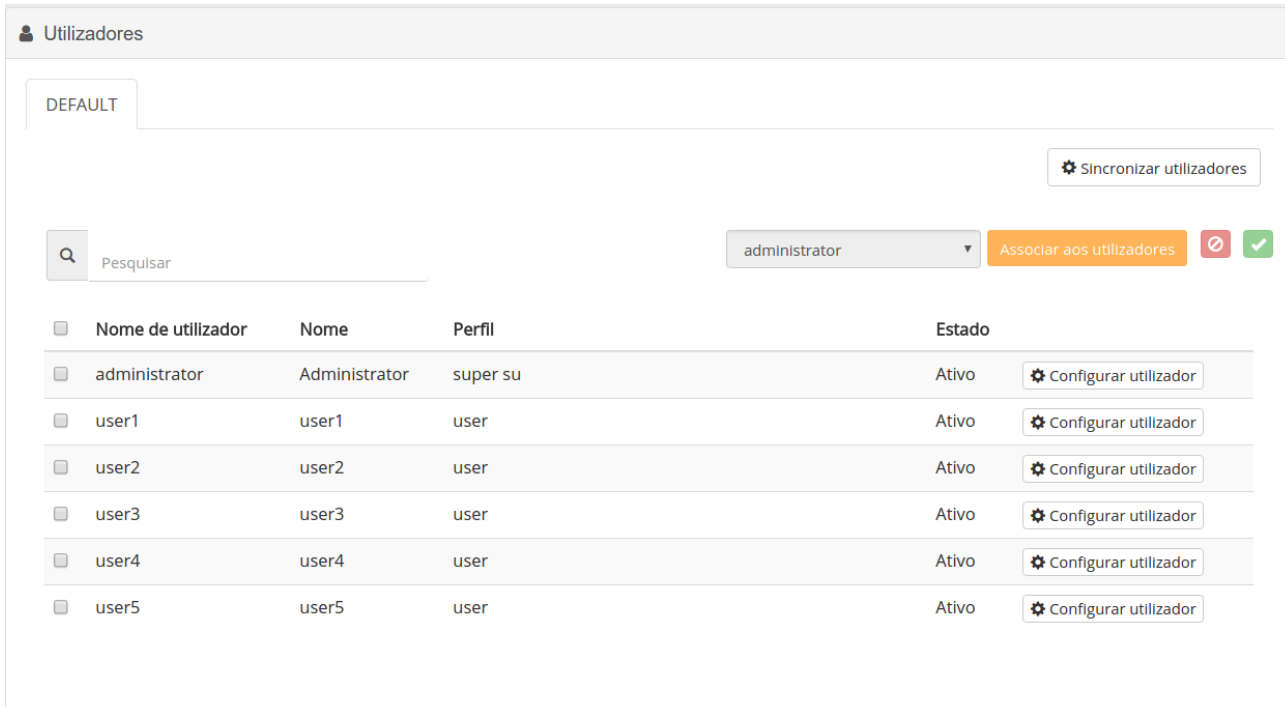


Figura 46 – Utilizadores: versão antiga.

Para resolver este problema, resolveu-se usar a linguagem *JavaScript*:

```
// Bloquear a DropDown
$('#role_user_').prop("disabled", true);
// Bloquear a classe que contém o botão ,  e de "Associação de
  utilizadores"
$('.btn_special').prop("disabled",true);
$('.checkboxlist').click(function() {
// Condição que desbloqueia todos os botões caso a checkbox seja ativada
  if ($(this).is(':checked')) {
    $('#role_user_').prop("disabled", false);
    $('.btn_special').prop("disabled",false);
      } ...
  }
}
```

Decidiu-se eliminar desta secção a escolha do aparecimento de fotografias. Esta foi migrada para a secção “Configurações”, da área da administração. Na Figura 47 está representada a nova versão da secção de atribuição de privilégios dos utilizadores.





Utilizadores

DEFAULT

Sincronizar utilizadores

Pesquisar

administrator

Associar aos utilizadores  







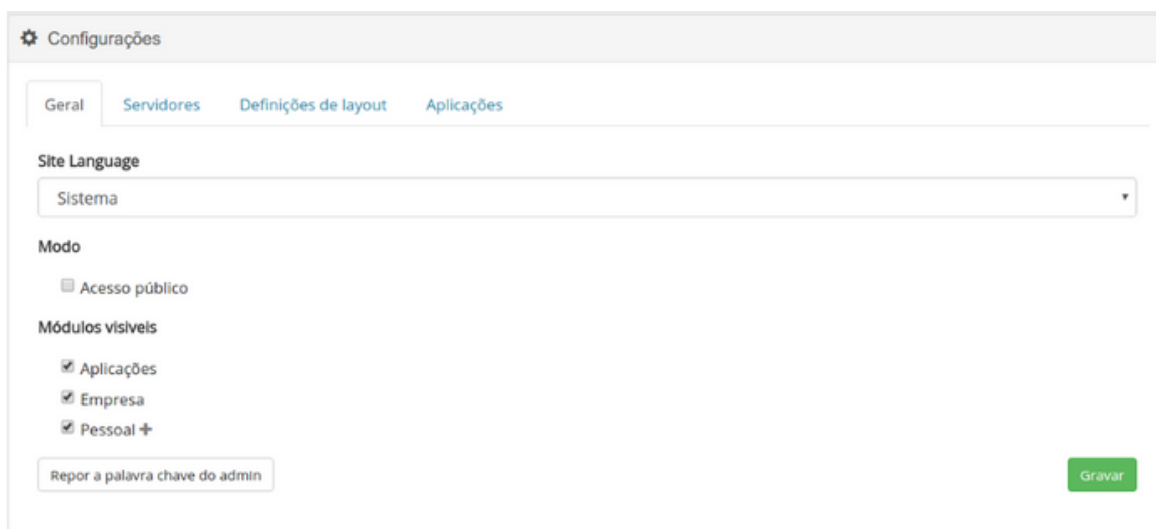
<input type="checkbox"/>	Nome de utilizador	Nome	Perfil	Estado	
<input type="checkbox"/>	administrator	Administrator	super su	Ativo	 Configurar utilizador
<input type="checkbox"/>	user1	user1	user	Ativo	 Configurar utilizador
<input type="checkbox"/>	user2	user2	user	Ativo	 Configurar utilizador
<input type="checkbox"/>	user3	user3	user	Ativo	 Configurar utilizador
<input type="checkbox"/>	user4	user4	user	Ativo	 Configurar utilizador
<input type="checkbox"/>	user5	user5	user	Ativo	 Configurar utilizador

Figura 47 – Utilizadores: Versão atual.

3.1.3. MELHORAR A PÁGINA DAS CONFIGURAÇÕES

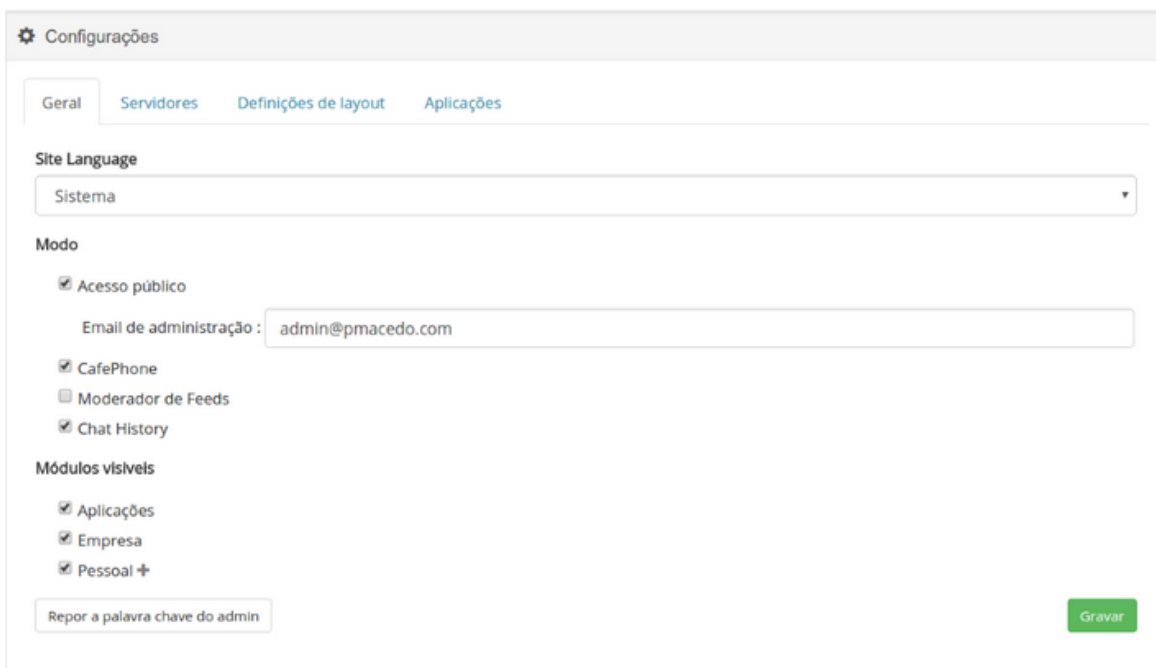
Na secção das configurações (ver versão antiga na Figura 48), foram efetuadas várias modificações na *tab* "Geral" e na *tab* "Definições do Layout".

Na *tab* "Geral" foram adicionadas três novas *checkboxes*: "Moderador de feeds", "Histórico de chat" e "CafePhone" (ver Figura 49). Estas novas *checkboxes* foram criadas com o objetivo de o utilizador ativar/desativar estas funcionalidades. Após colocar/retirar o visto da *checkbox*, esta informação é guardada na base de dados.



The screenshot shows the 'Configurações' page with the 'Geral' tab selected. The 'Site Language' dropdown is set to 'Sistema'. Under 'Modo', the 'Acesso público' checkbox is unchecked. Under 'Módulos visíveis', the checkboxes for 'Aplicações', 'Empresa', and 'Pessoal' are all checked. A text input field for 'Repór a palavra chave do admin' is present. A green 'Gravar' button is located at the bottom right.

Figura 48 – Configurações - Geral: Versão anterior.



The screenshot shows the updated 'Configurações' page. The 'Site Language' dropdown remains 'Sistema'. Under 'Modo', 'Acesso público' is checked, and a new 'Email de administração' field contains 'admin@pmacedo.com'. Under 'Módulos visíveis', 'Aplicações', 'Empresa', and 'Pessoal' are checked. New checkboxes for 'CafePhone', 'Moderador de Feeds', and 'Chat History' are present. The 'Gravar' button is still at the bottom right.

Figura 49 – Configurações - Geral: Versão atual.

No que relaciona às alterações na *tab* “Definições de *Layout*” (ver Figura 50), achou-se que seria mais adequado melhorar esta página, pois antes estava desorganizada.

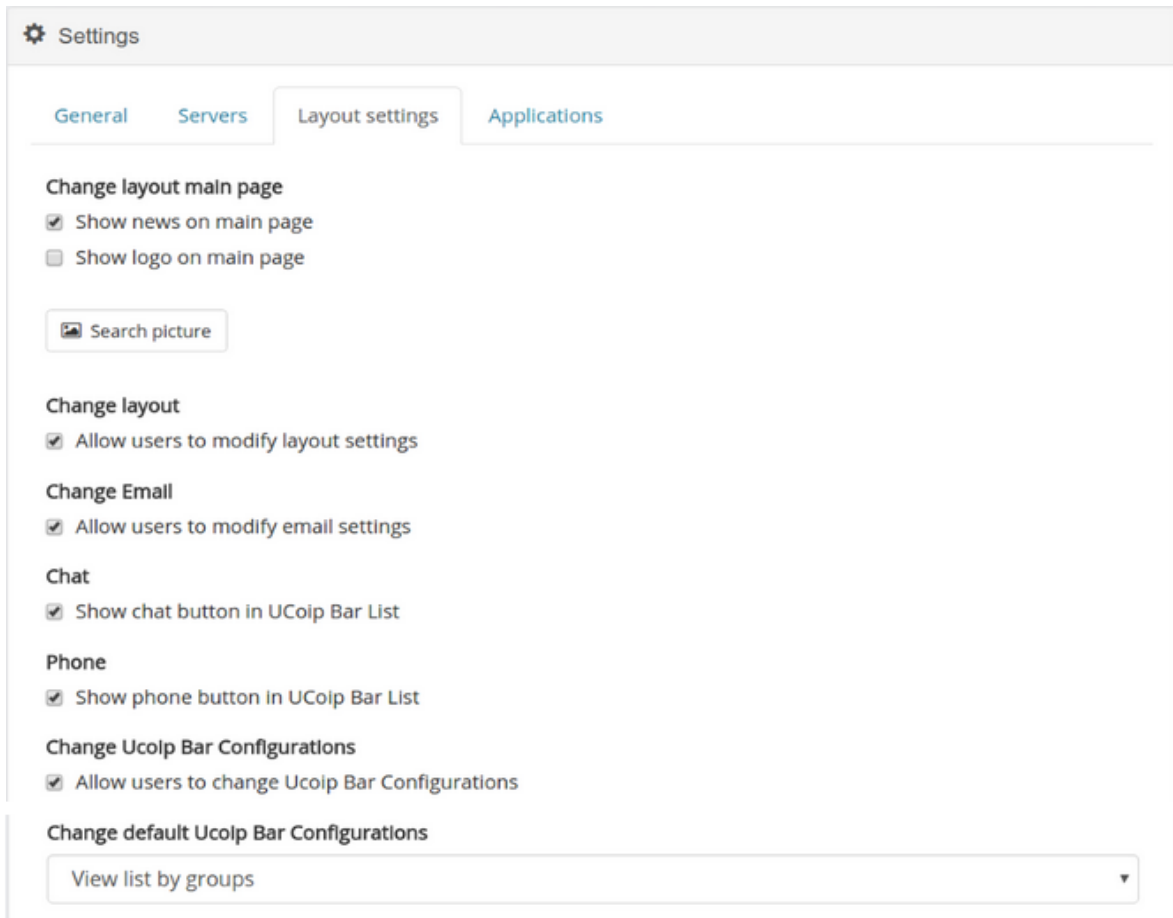


Figura 50 – Configurações – Definições de *layout*: Versão anterior.

A mudança na *tab* “Definições de *Layout*” começou na alteração o tamanho e título de “*Change layout main page*” para “*Layout of login page*”, dando a entender melhor o seu objetivo.

Esta alteração foi feita através da base de dados:

```
$model = Configuration::model()->
findByAttributes(array('variable'=>'layoutmainpage_news'));
<h4><?php echo Yii::t('app', $model->name);?></h4>
```

A grande alteração neste *layout* foi o modo de configuração da *UcoIPBar*. Antes só era possível divulgar/omitir o ícone de *email* e de videoconferência. Com as alterações efetuadas, é possível agora escolher qualquer ícone para a *UcoIPBar* (*chat*, *email*, telefone e videoconferência).

Para criar cada *checkbox*, seguiu-se os seguintes passos (o exemplo utilizado será da videoconferência):

1. Criar a *checkbox*;

A *checkbox* é a interface chave para a concretização desta tarefa. Com ela, pode-se selecionar a opção divulgar/omitir o ícone desejado.

```
<div class="checkbox">
  <label>
    //Criação da checkbox
    <?php echo CHtml::checkbox('Configuration['.$model-
>variable.'],'', $model->value); ?> <?php echo $variable; ?>
  </label>
</div>
```

2. Gravar o valor da *checkbox* na base de dados;

```
$modelConf = Configuration::model()-
>findByAttributes(array('variable' => 'VideoConf'));

// Condição para gravar na base de dados. 1 para checked e 0 para
// unchecked
if (isset($_POST['Configuration']['VideoConf'])) {
    $modelConf->value = $_POST['Configuration']['VideoConf'];
} else {
    $modelConf->value = 0;
}
// Grava a checkbox
$modelConf->save();
```

3. Fazer aparecer/desaparecer o ícone. Se o valor da *checkbox* estiver a 0 faz desaparecer o ícone;

```
$displayvideo = ' display:inline-block;';
if (Configuration::model()->findByAttributes(array('variable' =>
'VideoConf'))->value === '0') {
    $displayvideo = 'display:none;';
}
```

4. Chamar a variável na criação do ícone;

```
... style="<?php echo $displayvideo ?>" ...
```

Por fim, foi adicionada a possibilidade de tornar ativa a exibição das fotografias dos utilizadores e do aparecimento/desaparecimento dos botões do *CAFE Phone* na *UcoIPBar*, seguindo a mesma lógica da criação dos outros ícones (ver Figura 51).

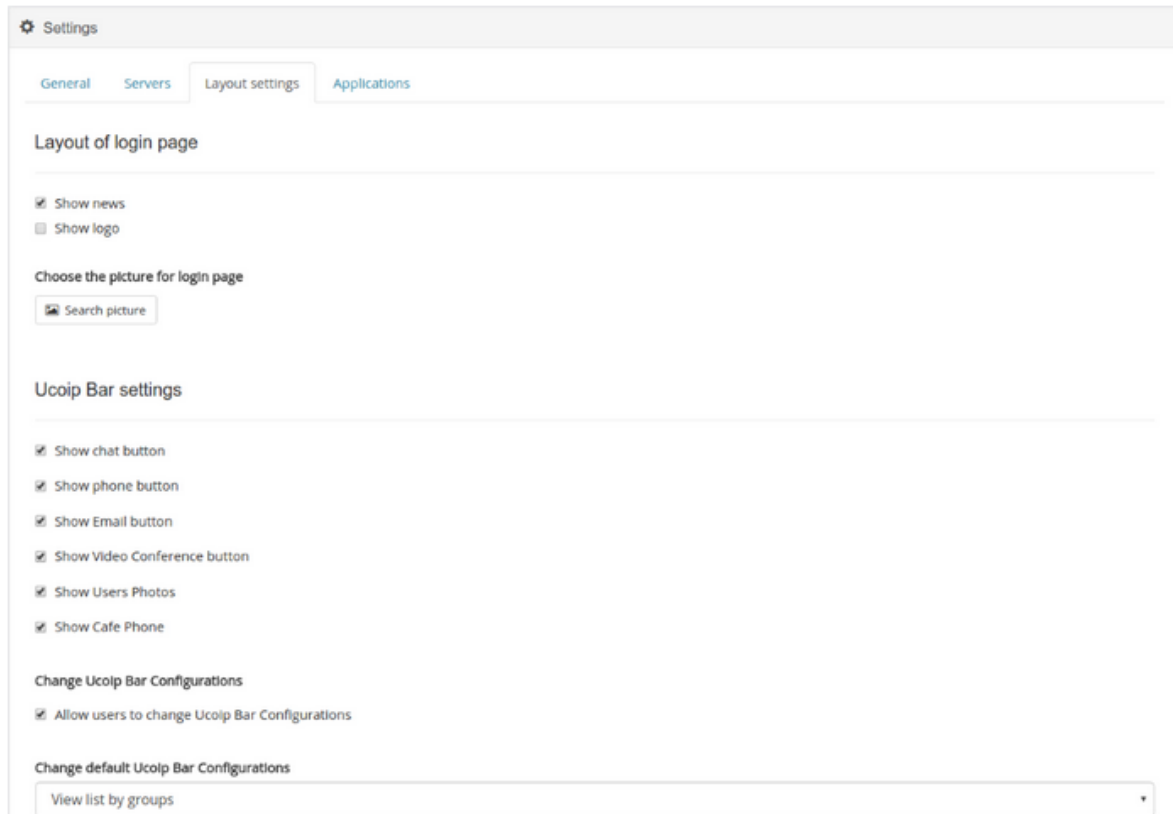


Figura 51 – Configurações – Definições de *layout*: Versão atual.

3.1.4. FORMATAÇÃO DOS BOTÕES DO *CAFE PHONE*

Os botões do *CAFE Phone* estão presentes na parte inferior da *UcoIPBar*. Estas são divididas entre o botão *UcoIP* (Canto inferior esquerdo - Figura 52) e *WebPhone* (Canto inferior direito - Figura 53).

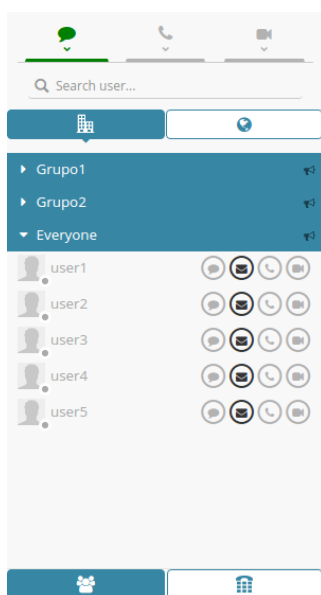


Figura 52 – Barra UcoIP.

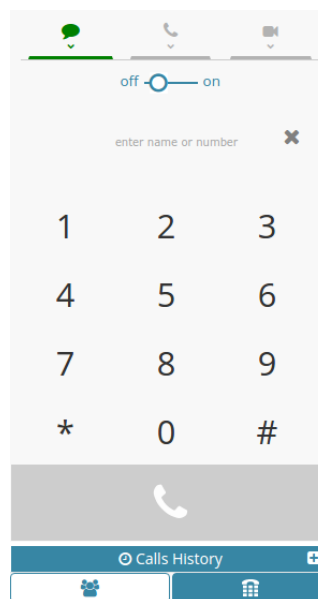


Figura 53 – Barra WebPhone.

O problema destes botões passa pelo facto de estes desaparecerem (ver canto inferior direito da Figura 54) caso o logótipo do cabeçalho do *CAFE* seja demasiado grande (ver canto superior esquerdo da Figura 54).

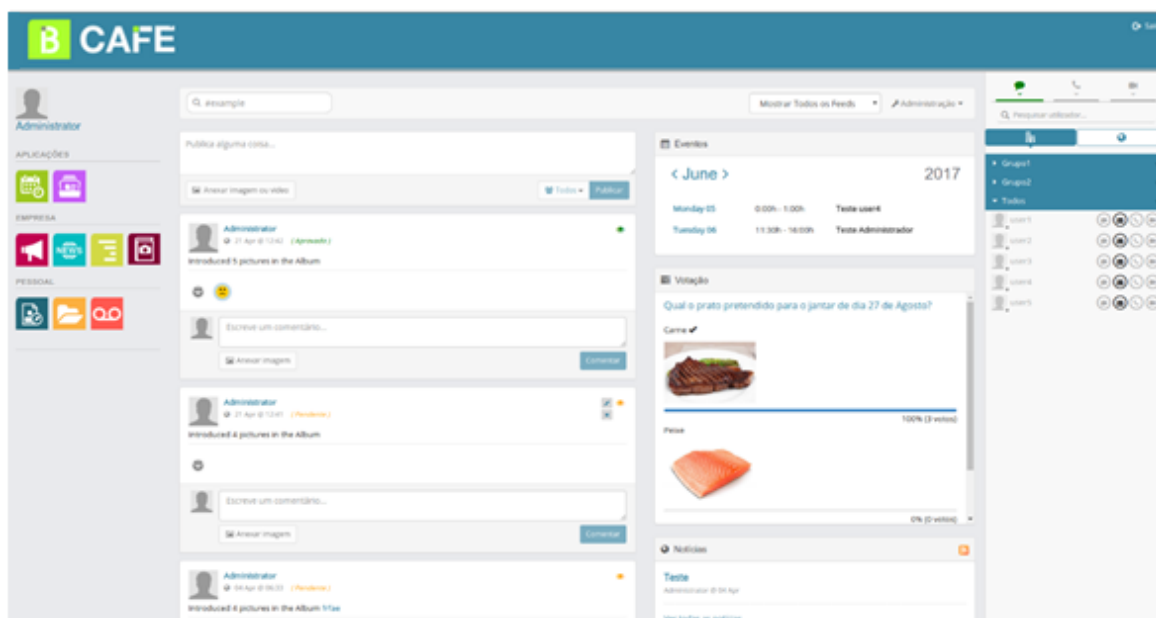


Figura 54 – Os botões inferiores da *UcoIP bar* não aparecem.

De modo a resolver este problema, decidiu-se usar a ferramenta *CSS*. Este é um problema que não só atinge a versão *desktop* como também a versão *mobile*. O código abaixo evita que os botões da *UcoIPbar* desapareçam para baixo.

O *calc* é um comando, que neste caso, faz a diferença entre o tamanho da página *web* inteira e o tamanho do logótipo definido. Este cálculo é inserido na função ".*ucoipbarlist*", classe da qual possui as propriedades da *UcoIPbar*. No canto inferior direito da Figura 55 verificamos que os botões inferiores da *UcoIPbar* já aparecem independentemente do tamanho do logotipo do cabeçalho.

```
.ucoipbarlist {  
...  
    height: calc(100% - <?php if (!$mobile) {echo  
(intval($layout_config['header']['logosize']) + 35) . 'px';}  
else echo "78px"?>);  
...  
}
```

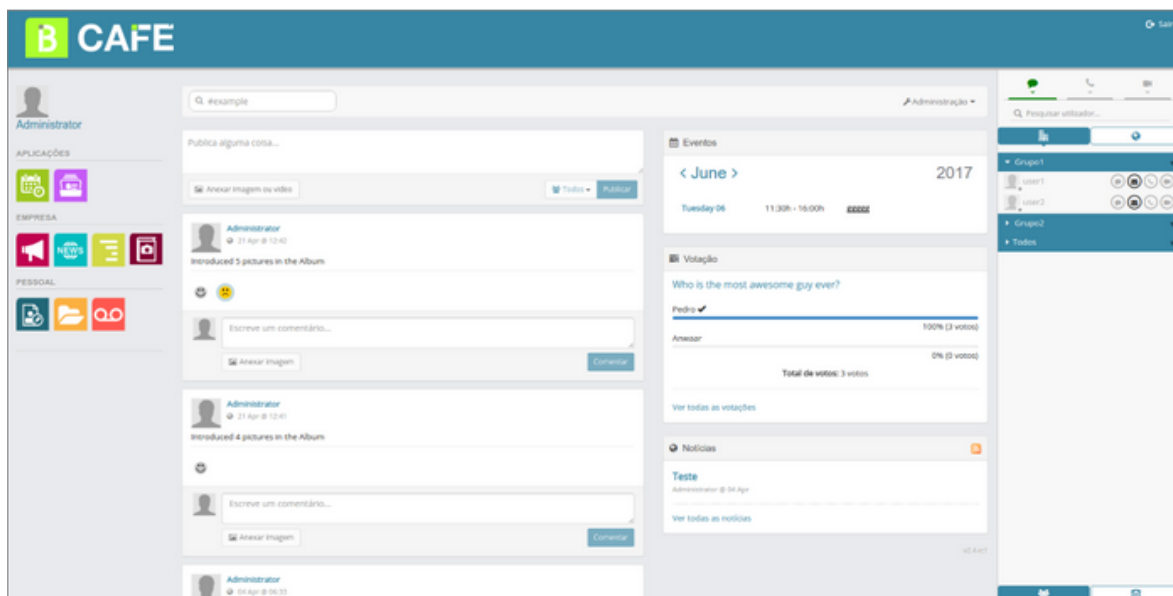


Figura 55 – Tamanho do logótipo não interfere nos botões da *UcoIPbar*.

3.1.5. MELHORAMENTO DA PERFORMANCE DO EDITOR DE TEXTO

O editor de texto *ckeditor* é o responsável pela publicação de notícias, sendo um editor *open source* e gratuito. Com a versão deste editor instalada (2.8) não era possível copiar/colar diretamente uma imagem para o editor de texto. Para além disso, não era possível fazer *upload* das imagens numa *cloud*. A única maneira de incluir imagens seria procurar na *internet* o *link* da imagem e colá-lo nas propriedades da imagem, no setor do “URL” (ver Figura 57).

Com as alterações efetuadas nesta secção, será possível aos utilizadores (com permissão de criar notícias) adicionar imagens ao criar uma notícia. Para tal, foi necessário

eliminar de raiz a versão antiga e instalar e configurar o *CKEditor* (versão 4.5.5) para o *CAFE*. A partir desta nova versão, já é possível copiar e colar diretamente uma imagem para o editor de texto.

Para colocar a *cloud* de imagens no *ckeditor*, foi necessário instalar e configurar à parte um *add-on*, com o nome “imageuploader”. Para tal foi necessário, em primeiro lugar, fazer *download* e instalar a biblioteca do “imageuploader”. Em segundo lugar foi necessário ativar esta funcionalidade através do seguinte código:

```
CKEDITOR.replace( 'editor', { extraPlugins: 'imageuploader' } );
```

Em terceiro lugar foi necessário criar a pasta necessária para guardar as imagens feitas por *upload*.

```
CKEDITOR.plugins.add( 'imageuploader', {  
  init: function( editor ) {  
    editor.config.filebrowserBrowseUrl =  
'/js/ckeditor/plugins/imageuploader/imgbrowser.php';  
    editor.config.filebrowserUploadUrl =  
'/home1/_cafe/images/news';  
  }  
});
```

Por fim é possível utilizar este *add-on*, através do botão representado com um círculo azul na Figura 56:

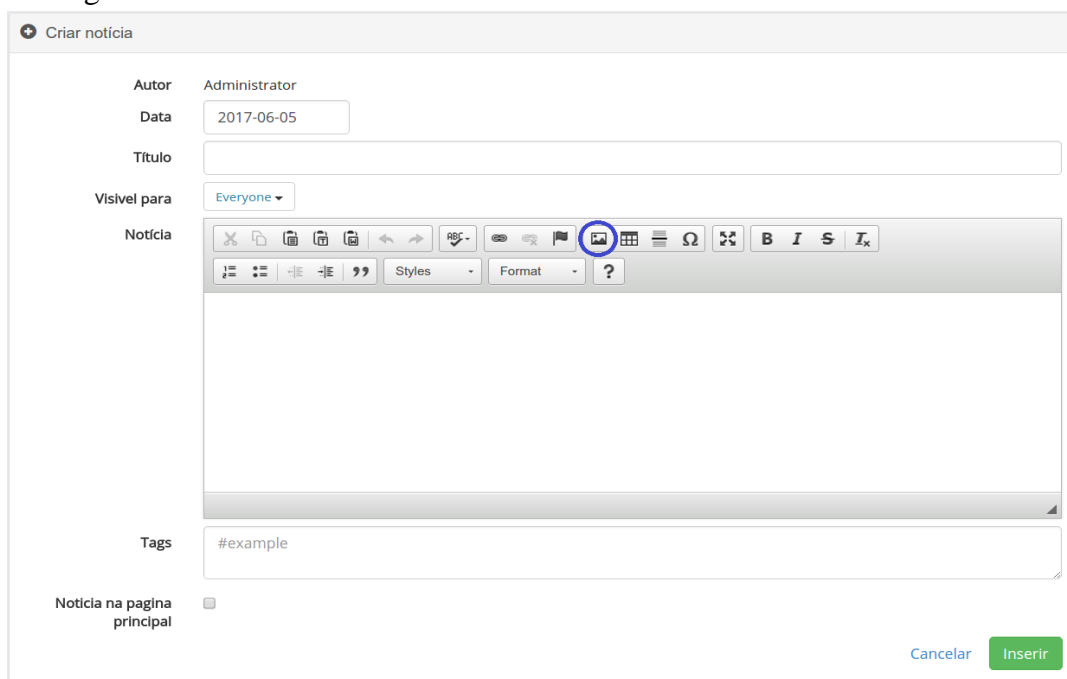


Figura 56 – CKEditor: Botão de *upload* de imagens.

De seguida, é necessário clicar no botão “Browse Server” (ver Figura 57) para abrir o *uploader* de imagens.

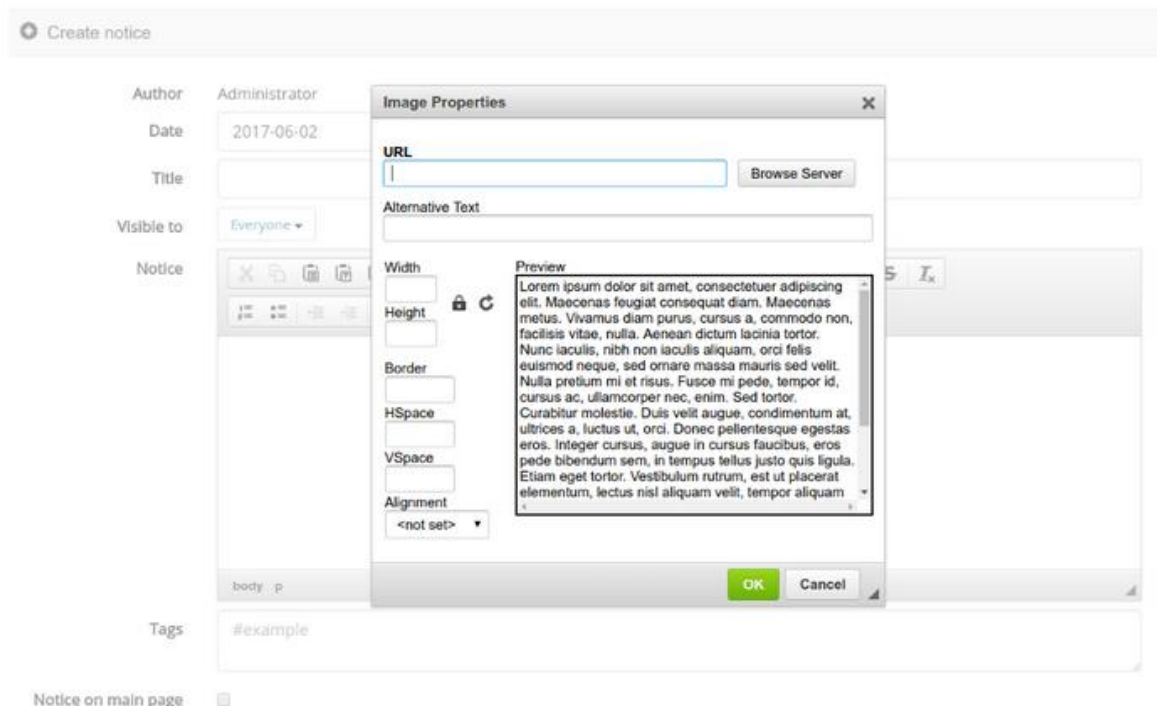


Figura 57 – CKEditor: Propriedades das imagens.

Após clicar no “Browse server”, será aberta a interface do *uploader*. Nesta é possível fazer *upload* através do botão em forma de nuvem no canto superior esquerdo da Figura 58.

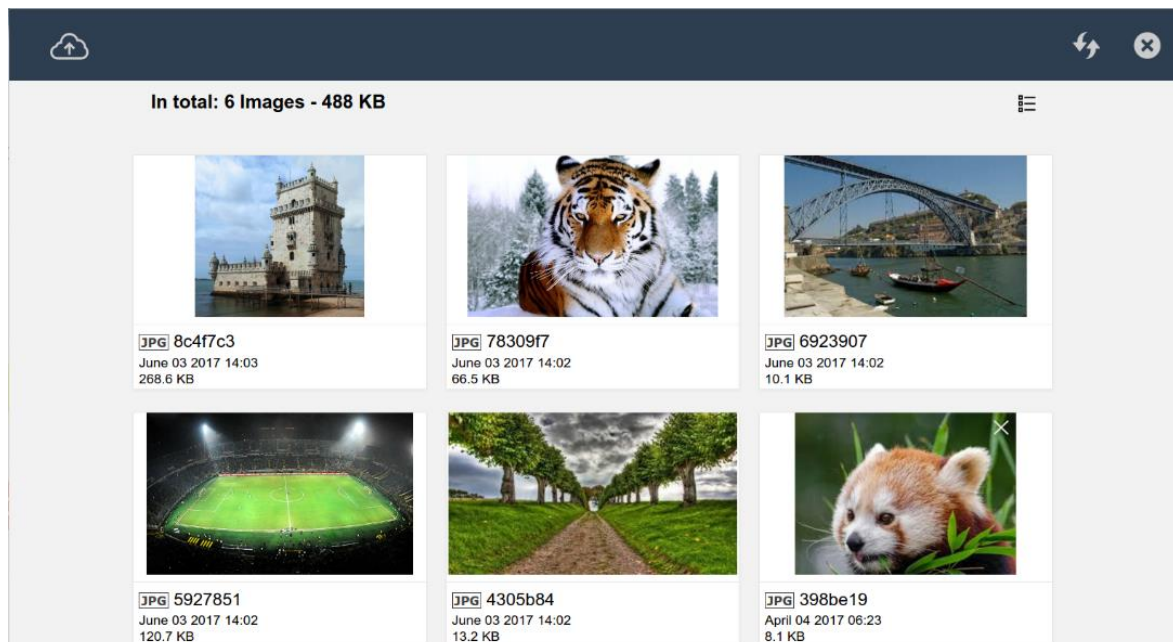


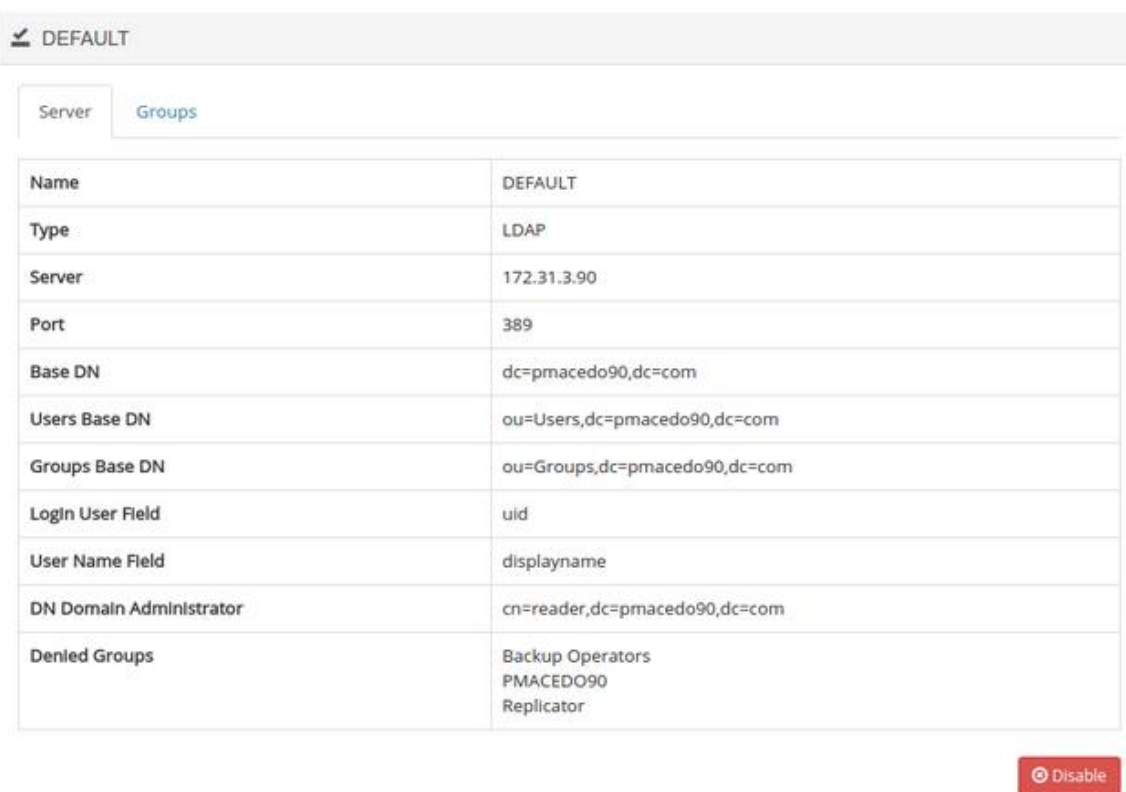
Figura 58 – Uploader de imagens: interface.

Com a realização desta tarefa, ficou bastante mais fácil o uso das imagens na criação de notícias.

3.1.6. MELHORAR A INTERFACE NA AUTENTICAÇÃO DOS SERVIDORES

Na área da administração, existe uma secção dedicada a criar servidores e a autorizar/proibir grupos dentro desse mesmo servidor.

No entanto a interface dos grupos proibidos não está de todo intuitiva para o utilizador, fazendo com o que o mesmo não perceba a sua usabilidade. Mais precisamente, na *tab* do “Server”, presente na Figura 59, são expostos os grupos proibidos e na *tab* “Groups”, representada na Figura 60 são seleccionados os grupos para proibir através da caixa “multiple”. Esta última tem a desvantagem de não poder escolher de forma múltipla vários servidores ao mesmo tempo (a única maneira de o fazer é seleccionar os grupos desejados enquanto o botão *ctrl* é clicado).



The screenshot shows a web interface for server configuration. At the top, there is a header with a dropdown menu set to 'DEFAULT'. Below this, there are two tabs: 'Server' (selected) and 'Groups'. The main content is a table with the following data:

Name	DEFAULT
Type	LDAP
Server	172.31.3.90
Port	389
Base DN	dc=pmacedo90,dc=com
Users Base DN	ou=Users,dc=pmacedo90,dc=com
Groups Base DN	ou=Groups,dc=pmacedo90,dc=com
Login User Field	uid
User Name Field	displayname
DN Domain Administrator	cn=reader,dc=pmacedo90,dc=com
Denied Groups	Backup Operators PMACEDO90 Replicator

At the bottom right of the interface, there is a red button labeled 'Disable'.

Figura 59 – Autenticação de servidores: Servidor – versão antiga.



Figura 60 – Autenticação de servidores: Grupos – versão antiga.

Para uma melhor organização desta secção foram realizados dois processos:

- Na *tab* “Server”: Os grupos não autorizados foram substituídos pelos grupos autorizados (ver Figura 61). Para efetuar essa mudança foi necessário usar a *framework Yii* para substituir a palavra “Denied Groups” por “Allowed Groups”. A *framework* neste caso foi utilizada para fazer as traduções.

```
Yii::t('app', 'Allowed Groups')
```

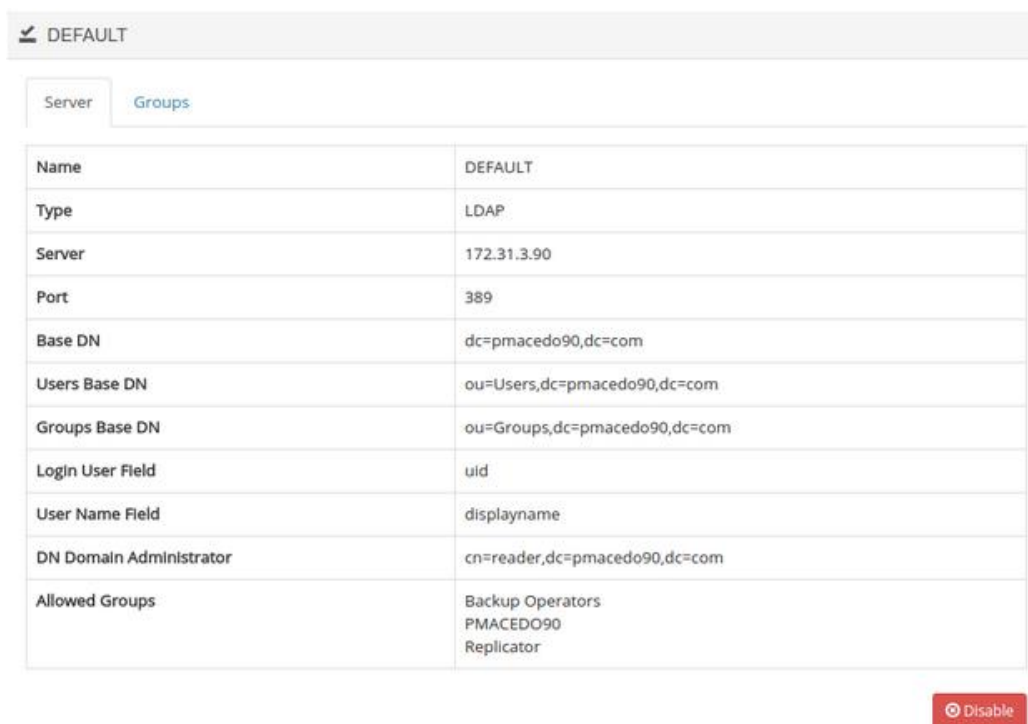


Figura 61 – Autenticação de servidores: Servidor – versão atual.

- Na *tab* “Groups”: foram criadas as *checkboxes* para cada grupo (ver Figura 62).

```

<div class="radio">
//Cria as checkboxes
<label><input type="checkbox" name="AuthServer[allowed_groups][]"
value="<?php echo $group['gidnumber'][0];?>"
    //Grava a checkbox
    <?php if ($selected) { echo 'checked'; }?>>
    //Mostra o nome dos grupos
    <?php echo $group['displayname'][0]; ?>
</label>
</div>

```

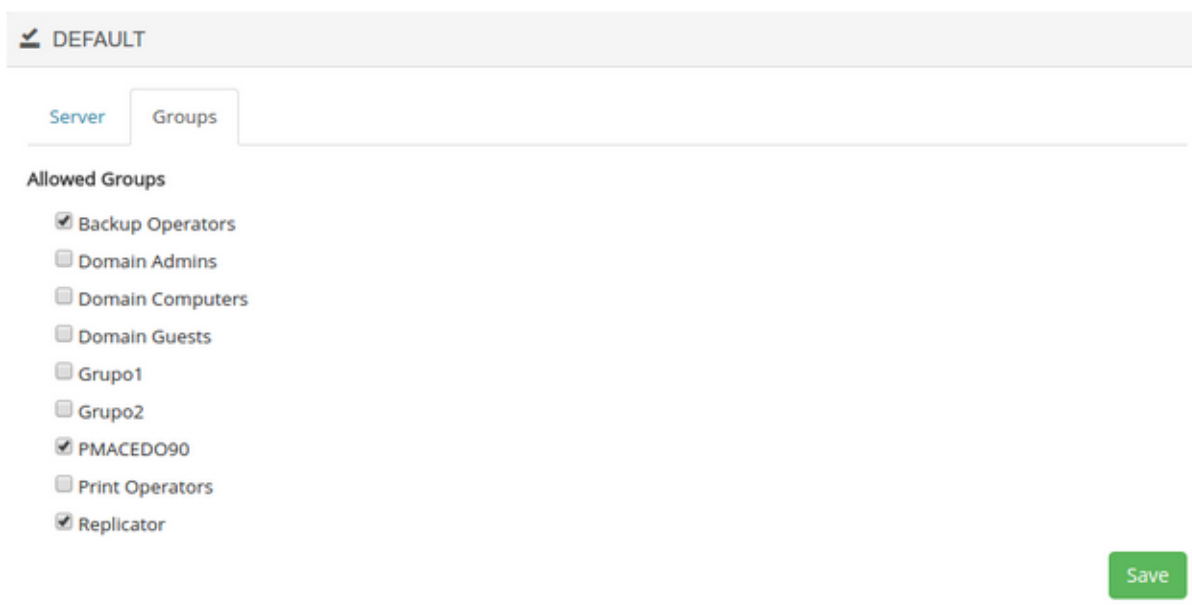


Figura 62 – Autenticação de servidores: Grupos – versão atual.

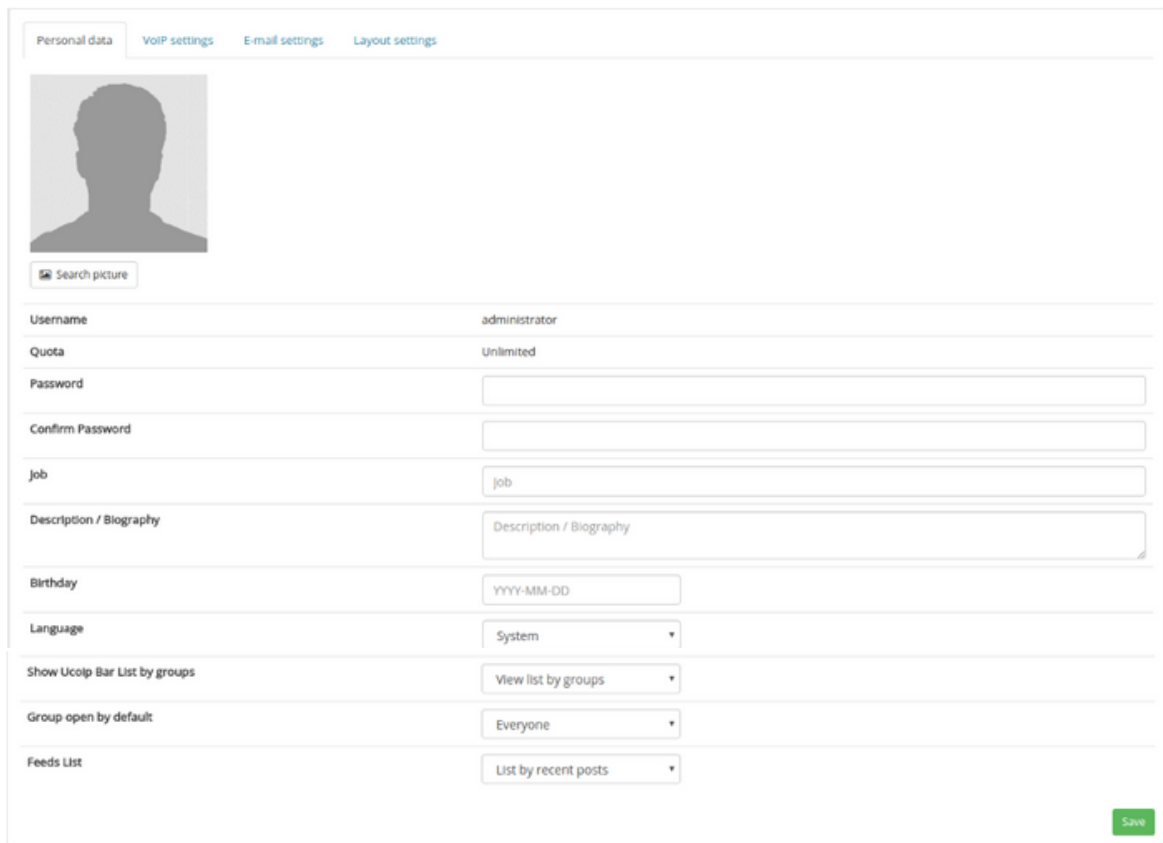
3.1.7. ALTERAR LAYOUT DA INTERFACE DA MUDANÇA DE PERFIL

O *CAFE* possui uma zona dedicada à configuração de perfil. Esta é dedicada a mudar os dados pessoais e definições do utilizador. Uma vez que esta secção foi considerada pouco apelativa, foi decidido reformular o *design* desta, de modo a torná-la menos antiquada e mais atrativa. Esta secção é dividida em quatro *tabs*: dados pessoais; definições de *VoIP*; definições de *email*; *layout*. Estas alterações não só foram realizadas na versão *desktop* como também para a versão *mobile*. Para editar a interface da mudança do perfil nesta versão, a

linguagem CSS foi deslocada para a classe “@ (max-width: 767px)”. Esta classe é dedicada exclusivamente para o desenvolvimento de interfaces para *mobile*.

3.1.7.1. Dados pessoais

Nesta *tab* (ver Figura 63) o utilizador pode fazer várias alterações pessoais, como o seu nome, alterar *password*, função na empresa, biografia, aniversário, entre outras opções.




Personal data	VoIP settings	E-mail settings	Layout settings
			
<input type="button" value="Search picture"/>			
Username	administrator		
Quota	Unlimited		
Password	<input type="text"/>		
Confirm Password	<input type="text"/>		
Job	<input type="text" value="job"/>		
Description / Biography	<input type="text" value="Description / Biography"/>		
Birthday	<input type="text" value="YYYY-MM-DD"/>		
Language	<input type="text" value="System"/>		
Show Ucolp Bar List by groups	<input type="text" value="View list by groups"/>		
Group open by default	<input type="text" value="Everyone"/>		
Feeds List	<input type="text" value="List by recent posts"/>		
<input type="button" value="Save"/>			

Figura 63 – Mudança de perfil: Dados pessoais – versão antiga.

A mudança começou na substituição da *tab* “Dados pessoais” por uma imagem (em formato *svg*) relativo a uma edição com o título “Pessoal”. Ao clicar nesta *tab* (ou em outras), irá surgir um efeito de desvanecer. Foram eliminadas as linhas de tabela e a escolha da fotografia foi deslocada para o lado direito da *tab*. Ao escolher a fotografia, o botão “Procurar fotografia” foi removido, sendo criado um novo botão transparente mais dinâmico em cima da fotografia (canto superior direito da Figura 65). Após passar o rato por cima do mesmo, ele irá mudar de cor, como sinalização, através da criação de uma classe para o botão, com o nome “btn-default1”.

```
.btn-default1 { // Se o rato não está por cima
  color: #fff; // Cor branca
}
.btn-default1:hover { // Quando o rato passa por cima
  color: #cecab5; // Cor cinzento/bege
}
```

Na zona da escolha do aniversário foi criado um ícone, do qual se for clicado irá surgir um *datepicker*, para escolher de forma mais rápida e intuitiva o dia de aniversário do utilizador. Através de *Javascript*, associa-se o botão ao aparecimento do *datepicker* (ver Figura 64).

```
<script type="text/javascript">
...
$(function() {
  //A escolha da data no datepicker irá para o campo do
  //aniversário
  $('#date-field').datepicker({'format': 'yyyy-mm-
dd',pickerPosition:'bottom-left'});
  //Inicialmente o datepicker está escondido
  $('#datefield').datepicker('hide');
//Se o ícone do calendário for clicado(svg7119), o datepicker irá
//aparecer
  $('#svg7119').on('click', function() {
    $('#date-field').datepicker('show');
  });
});
</script>
```

The image shows a user interface with four form fields and a datepicker. The fields are: 'Birthday' with the value '1993-07-04', 'Language' with the value 'English', 'Show Ucolp Bar List by groups' with the value 'View list by groups', and 'Group open by default' with the value 'Grupo1'. A calendar datepicker is open, showing the month of July 1993. The date '4' is highlighted in blue. The calendar has a header with '« July 1993 »' and a grid of days from Sunday to Saturday.

Figura 64 – Datepicker para a escolha do aniversário.

Para melhorar o aspeto da *dropdown menu* foi criado um botão especial com o nome de classe “arrowbox”. Este botão muda a cor de acordo com a cor de cabeçalho escolhida no *layout*. No fim desta *tab* foi criado a escolha do calendário que o utilizador deseja ver na página principal do *CAFE* através de *checkboxes*. Cada utilizador do *CAFE* tem o seu próprio calendário.

Por fim, o botão de “Gravar” desapareceu e deu lugar a um ícone em forma de disquete (canto superior direito da Figura 65).

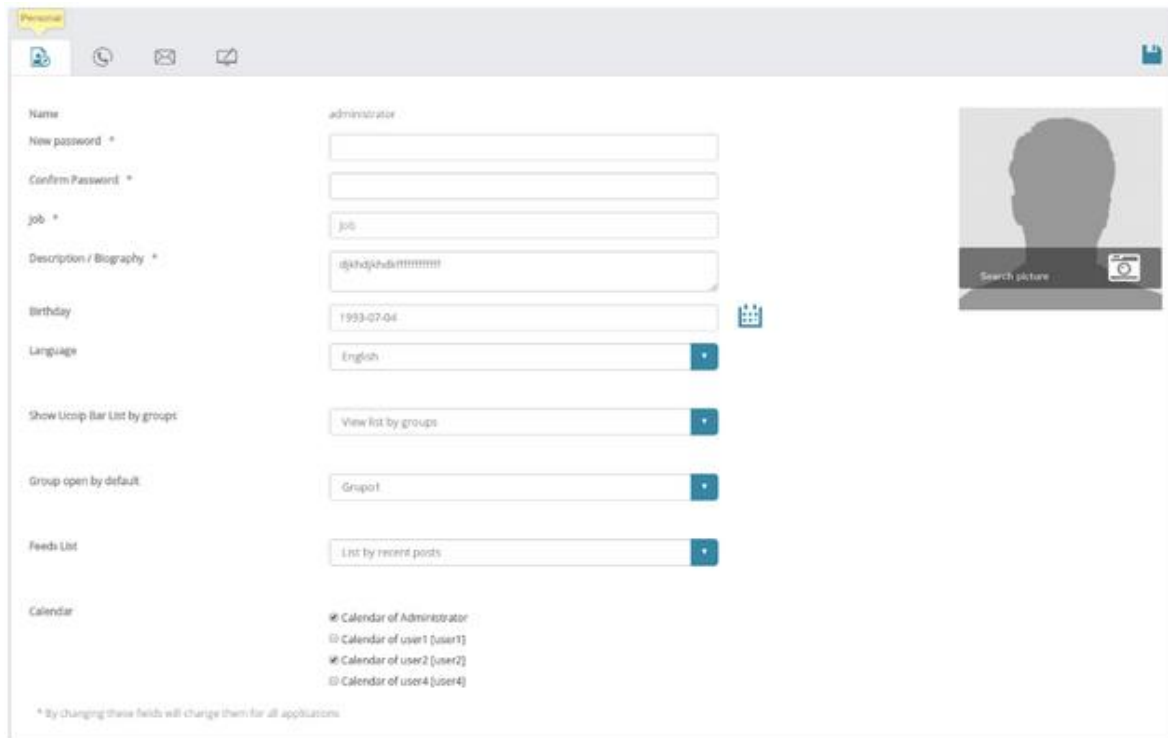



Figura 65 – Mudança de perfil: Dados pessoais – versão atual.

Para a versão *mobile*, o resultado para a nova interface no setor dos dados pessoais da área da mudança de perfil encontra-se apresentada à direita da Figura 66.

CAFE

Personal data | VoIP settings | E-mail settings

Layout settings



Search picture

Username: administrator

Quota: Unlimited

Password:

Confirm Password:

Job: job

Description / Biography: Description / Biography

Birthday: YYYY-

Language: Sy

Show Ucolp Bar List by groups: Vi


Group open by default: Ev

Feeds List: Ll

Save

CAFE

Procurar fotografia



Nome: administrator

Nova palavra-chave:

Confirmar *:

Função *: Função

Descrição / Biografia *: Descrição / Biografia

Aniversário: 1993-07-04

Idioma: Português

Ver Ucolp Bar List por grupos: Ver por grupos

Grupo aberto por defeito: Grupo1

Lista de Feeds: Listar por public

Calendar:

- Calendar of Administrator
- Calendar of user1 [user1]
- Calendar of user2 [user2]
- Calendar of user4 [user4]

* Ao alterar estes campos irá altera-los em todas as aplicações

Figura 66 – Versão *mobile*: Mudança de perfil: Dados pessoais – versão antiga (à esquerda) e versão atual (à direita).

3.1.7.2. Definições de VoIP

Esta *tab* tem como objetivo reencaminhar as chamadas para um ou mais utilizadores (ver Figura 67).

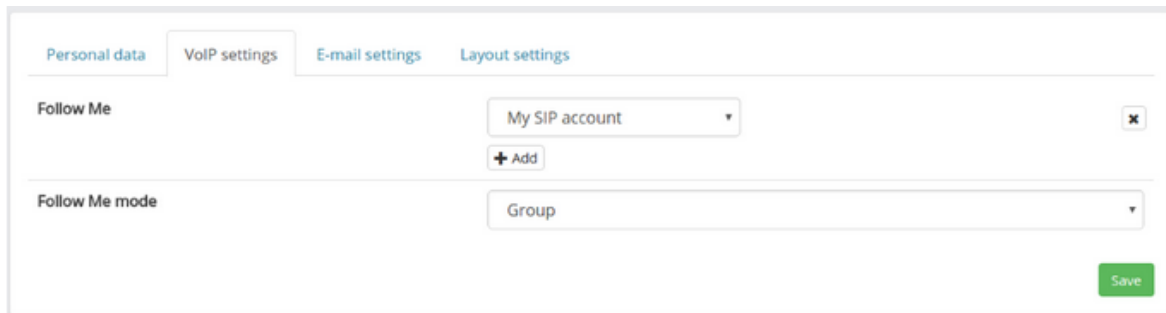


Figura 67 – Mudança de perfil: Definições de VoIP – versão antiga.

À semelhança da *tab* anterior, foi substituído a *tab* “Definições de VoIP” por uma imagem (em formato *svg*) relativo a uma edição com o título “VoIP”. De seguida, o botão “+Adicionar” deu lugar a uma imagem do *glyphicon*, um círculo com o símbolo “+” inserido. Antes, para criar mais “Follow me” ter-se-ia de clicar apenas nesse botão. Neste momento, cada vez que é criado um novo “Follow me”, surge este símbolo do *glyphicon*. Para mais, também é criada uma linha divisória entre os vários “Follow me” existentes. Caso esteja apenas um “Follow me” e clicar no botão de eliminar (ao lado direito do botão adicionar) irá eliminar o texto inserido no campo. Esta função foi elaborada em *JavaScript*.

```
//Cria uma divisão
var link2 = document.createElement('div');
//Insere o símbolo + na divisão
$(link2).html('<i class="glyphicon glyphicon-plus-sign"></i>');
//Insere uma classe que ajusta a posição
$(link2).attr('class', 'plusdiv');
//Chama a função que adiciona linhas e um novo "Follow me"
$(link2).attr('onclick', 'appearlines()');
```

```
//Função que adiciona nova linha sempre que adiciona um follow me
function appearlines(){
    $('.li1:last').css('border-bottom-style', 'dashed');
    $('.li1').css('border-bottom-color', 'darkgrey');
    $('.li1').css('margin-bottom', '12px');
    $('.li1:first').css('margin-bottom', '0px');
    $('.li1').css('border-bottom-width', '1px');
```

```
$('.li1').css('border-bottom-position','relative');
addFollowMe();
}
```

Na Figura 68 está representada o resultado da interface da secção *VoIP* após as alterações efetuadas.

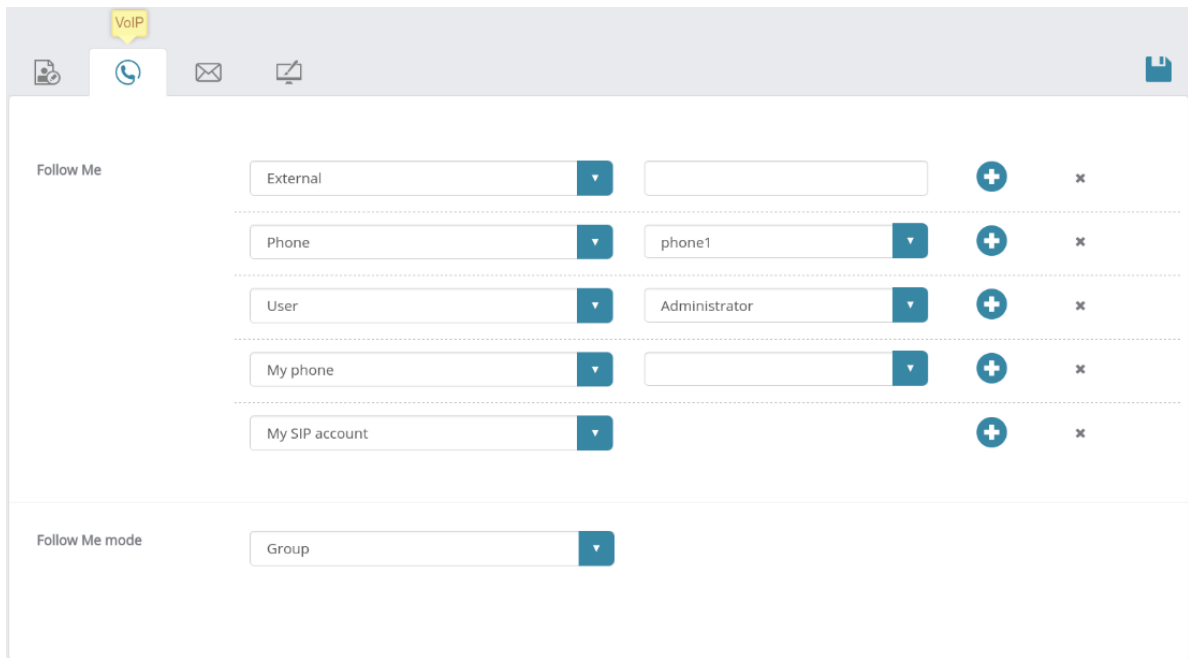


Figura 68 – Mudança de perfil: Definições de VoIP – versão atual.

Na versão *mobile*, a mudança está representada na Figura 69.

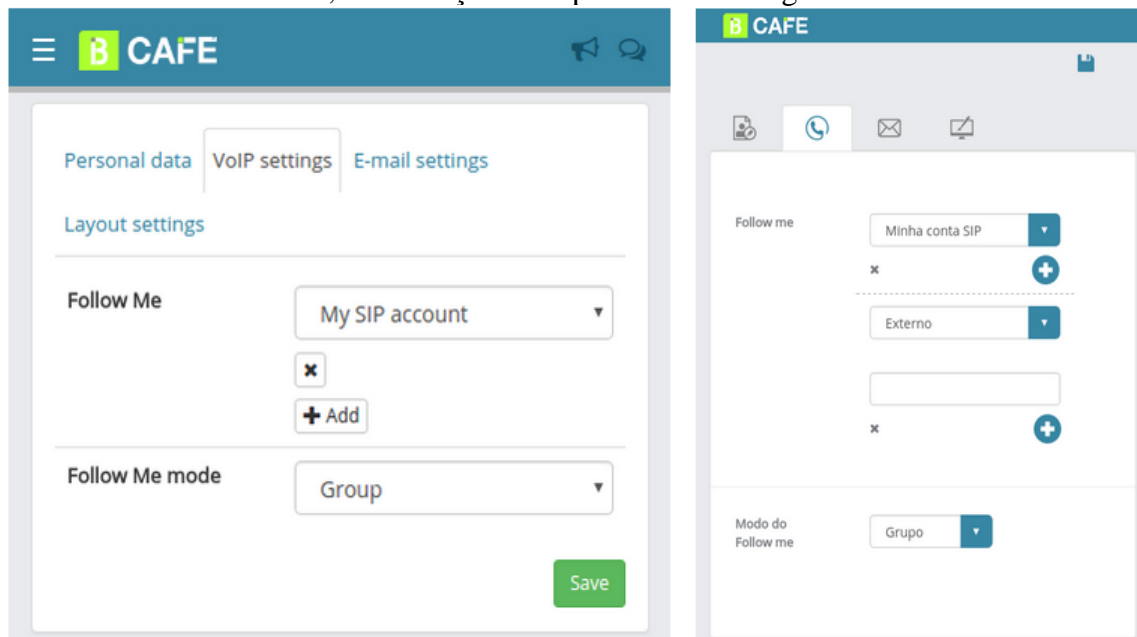
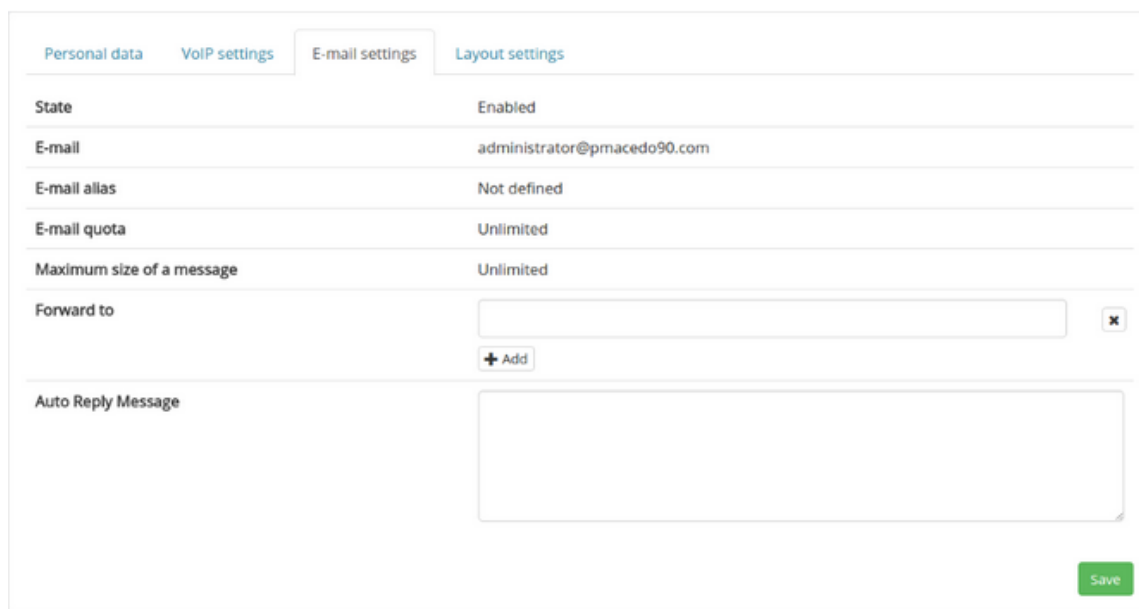


Figura 69 – Versão *mobile*: Mudança de perfil: VoIP – versão antiga (à esquerda) e versão atual (à direita).

3.1.7.3. Definições do *email*

As definições de *email* têm como objetivo definir *emails* alternativos, reencaminhar *emails* para outras pessoas ou personalizar as notificações do mesmo. Na Figura 70 está presente a versão por alterar da secção “Definições de *email*”.



The screenshot shows a web interface for configuring email settings. At the top, there are four tabs: 'Personal data', 'VoIP settings', 'E-mail settings' (which is active), and 'Layout settings'. Below the tabs, the settings are organized into a table-like structure with labels on the left and values on the right. The settings include: 'State' (Enabled), 'E-mail' (administrator@pmacedo90.com), 'E-mail alias' (Not defined), 'E-mail quota' (Unlimited), and 'Maximum size of a message' (Unlimited). There is a 'Forward to' section with an empty text input field, a small 'x' icon to its right, and a '+ Add' button below it. Below that is an 'Auto Reply Message' section with a large empty text area. A green 'Save' button is located in the bottom right corner of the form.

Label	Value
State	Enabled
E-mail	administrator@pmacedo90.com
E-mail alias	Not defined
E-mail quota	Unlimited
Maximum size of a message	Unlimited
Forward to	<input type="text"/>
	+ Add
Auto Reply Message	<div style="border: 1px solid #ccc; height: 40px;"></div>

Figura 70 – Mudança de perfil: Definições de *email* – versão antiga.

Para alterar o *design* desta *tab*, procedeu-se de novo à substituição da *tab* “Definições de Email” por uma imagem com o título “Email”. Foram eliminadas as linhas horizontais de separação e em “Encaminha para”, seguiu-se o mesmo procedimento do *Voip*, ao substituí o botão adicionar pelo ícone do *glyphicon*. Já a cruz para eliminar foi deslocada para o campo de texto, contendo a mesma função da *tab* anterior (eliminar o texto caso só exista só um utilizador). Na “Resposta automática”, foi criado um botão *toggle* do *bootstrap* que faz aparecer/desaparecer a caixa de texto para este campo. No fim, foram criadas as opções de notificação no *email* de novos *posts*, comentários ou reações criadas no *CAFE*. Na Figura 71 está representada a versão com as alterações referidas.

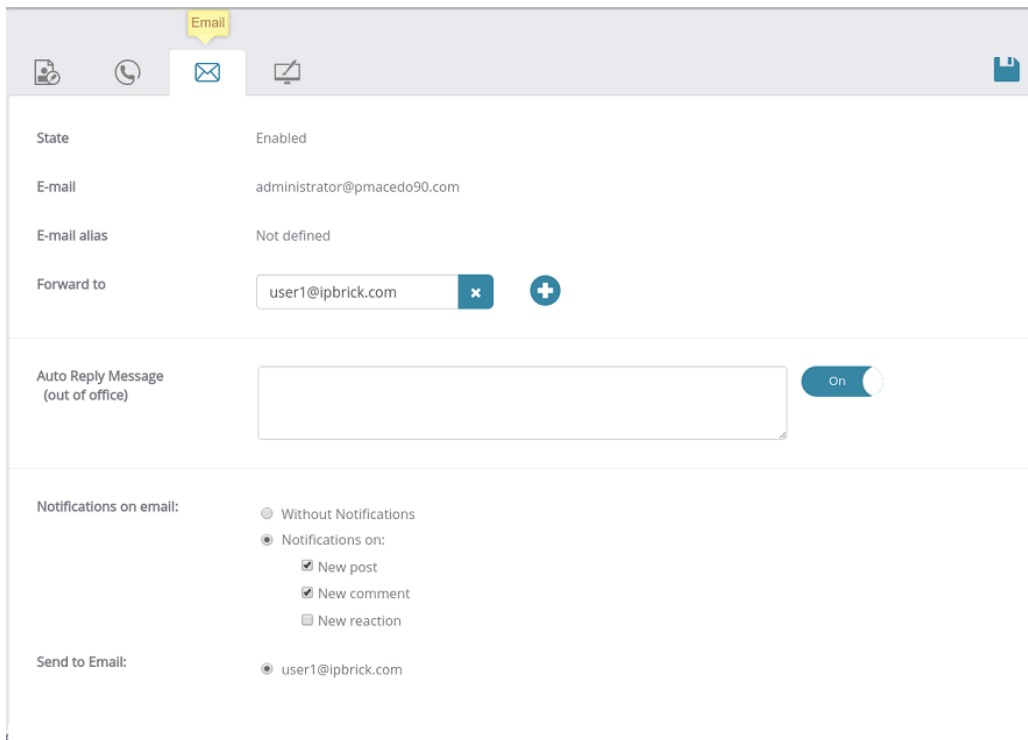


Figura 71 – Mudança de perfil: Definições de *email* – versão atual.

Na versão *mobile*, a mudança na apresentação da interface está representada na Figura 72.

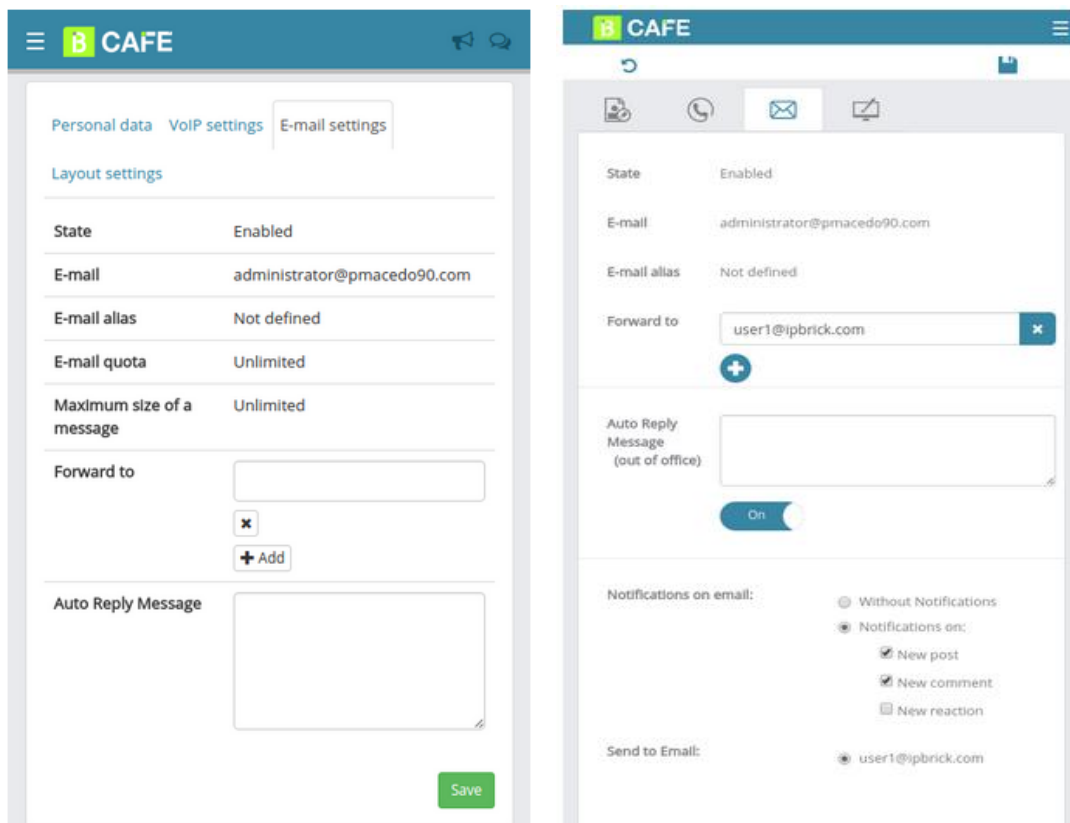


Figura 72 – Versão *mobile*: Mudança de perfil: *Email* – versão antiga (à esquerda) e versão atual (à direita).

3.1.7.4. Definições de *Layout*

No cabeçalho do *layout* do *CAFE* (ver Figura 73), é possível: escolher várias cores; mudar o logótipo; colocar uma imagem de fundo; opção de ocultar o logótipo. Nas definições avançadas o utilizador pode ir mais longe nas características do cabeçalho bem como mudar o tipo e forma de letra usado no *CAFE*.

The screenshot shows the 'Layout settings' tab in the CAFE user interface. It features several sections for customization:

- Themes:** A row of color swatches for selecting a theme.
- Logo:** A preview of the 'B CAFE' logo. Below it are options to search for a picture, show/hide the logo, and set its size (2x px).
- Background image:** A preview of a blue background image. Below it are search options and a 'Show background image' toggle.
- Advanced Settings:** A dropdown menu that expands to show more options:
 - Header:** Background color (hex #3786a4), Repeat background image (Repeat/No repeat), Vertical position of the background image (0%), and Horizontal position of the background image (0%).
 - Links:** Font color (hex #3786a4) and Font weight (normal/bold).
 - Buttons:** Font color (hex #ffffff) and Background color (hex #3786a4).
 - Font Family:** A list of fonts including Open Sans, Arial, Verdana, Trebuchet MS, Tahoma, and Times New Roman, with an 'Other' option.
- Example:** A preview of the header with a 'Lorem ipsum' button.

At the bottom right, there are 'Reset to default' and 'Save' buttons.

Figura 73 – Mudança de perfil: *Layout* – versão antiga.

Como nas outras *tabs*, a palavra, neste caso “Definições de *layout*”, foi substituída por uma imagem. Na secção “Tema”, foi acrescentado um novo botão (🔍), do qual abre uma janela (representada na Figura 74) para escolher qualquer cor do espectro visível.

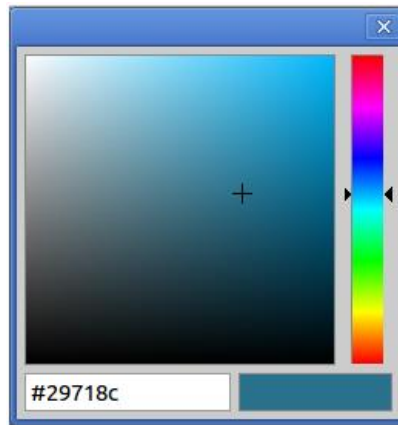


Figura 74 – Janela RGB.

Seguindo a mesma lógica para alterar a fotografia nos “Dados pessoais”, para alterar o logótipo e a imagem de fundo, substituiu-se o botão “Procurar fotografia” por um botão transparente mais moderno. Os botões *radio* para mostrar o logótipo e a imagem de fundo foram substituídos por botões *toggle* do *bootstrap*.

Na secção das configurações avançadas, foi alterada a interface dos vários tópicos apresentados neste setor. À semelhança dos das *tabs* anteriores, os botões *radio* para o tipo de letra e o texto a negrito foram substituídos por *dropdown* menus. Na Figura 75 está exibida a nova versão após realizar as alterações.

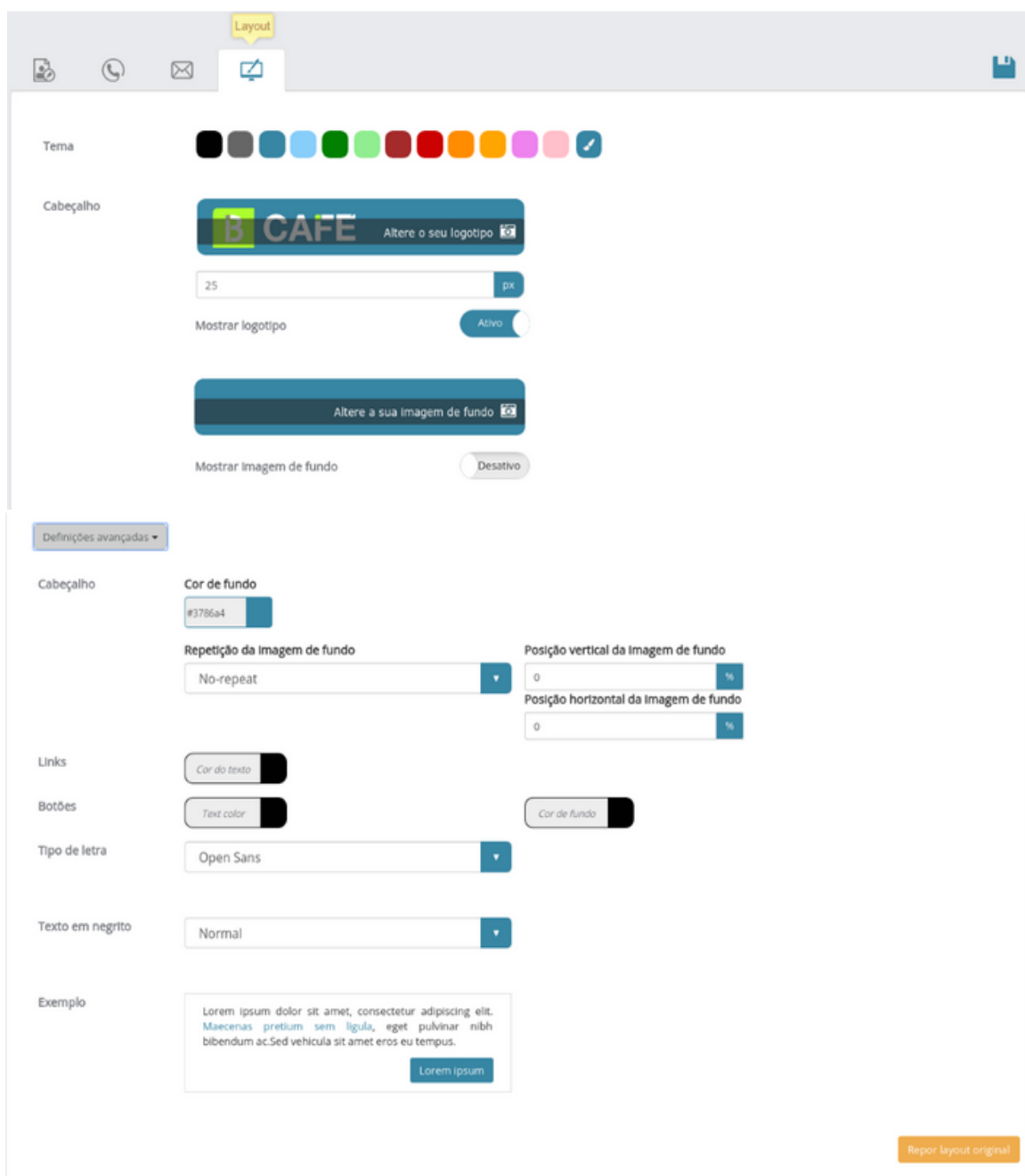


Figura 75 – Mudança de perfil: *Layout* – versão atual.

Na versão *mobile*, estão representadas nas Figuras 76 e 77 as mudanças na interface foram as seguintes:

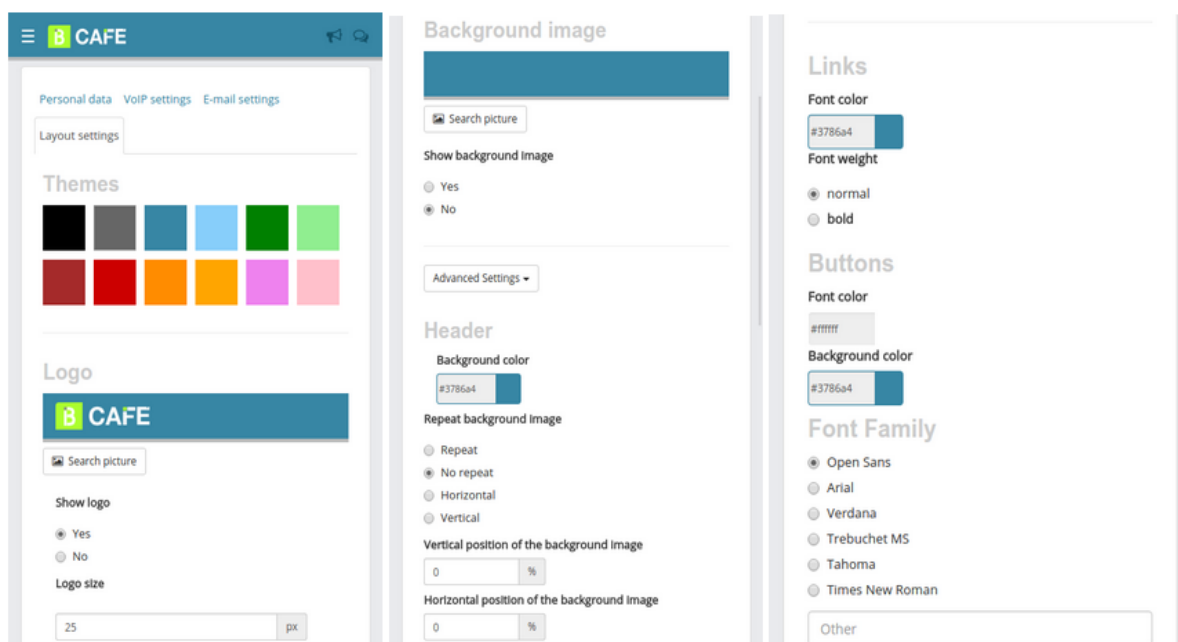


Figura 76 – Versão *mobile*: Mudança de perfil: *Layout* – versão antiga.

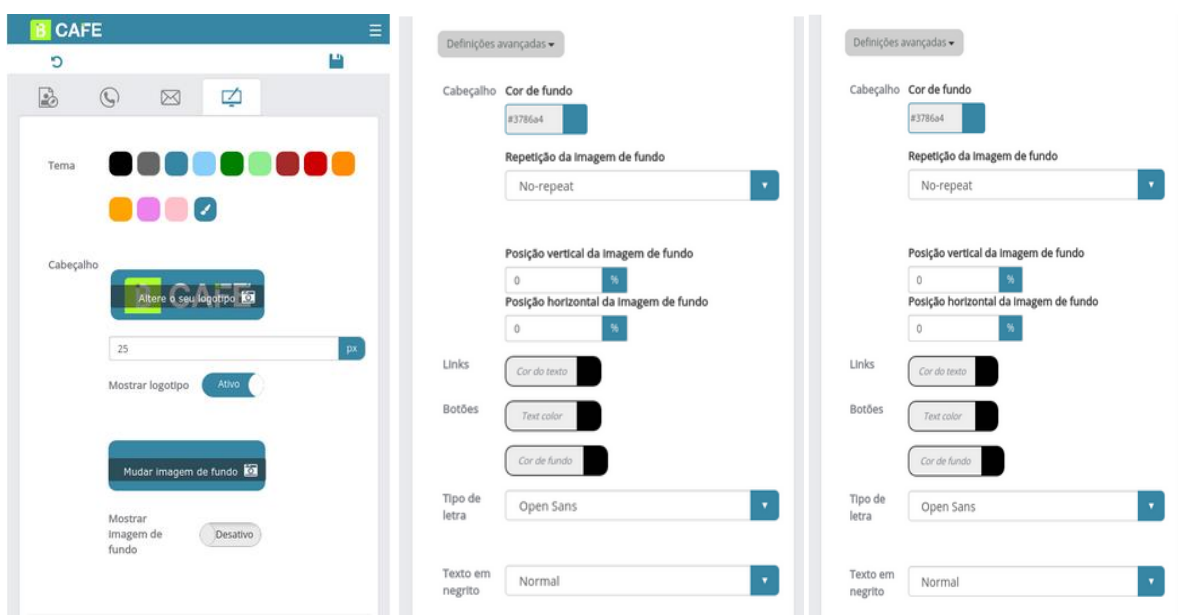


Figura 77 – Versão *mobile*: Mudança de perfil: *Layout* – versão atual.

3.1.8. CONFIGURAÇÕES – APLICAÇÕES

O utilizador pode gerir as suas aplicações através desta secção. Por defeito, estão presentes as aplicações que a *IPBRICK* tem contacto. É possível removê-las, editá-las, ativá-las (aparecem no atalho das aplicações) ou bloqueá-las (ficam ocultas no atalho das aplicações). A Figura 78 mostra a versão a melhorar na gestão das aplicações.

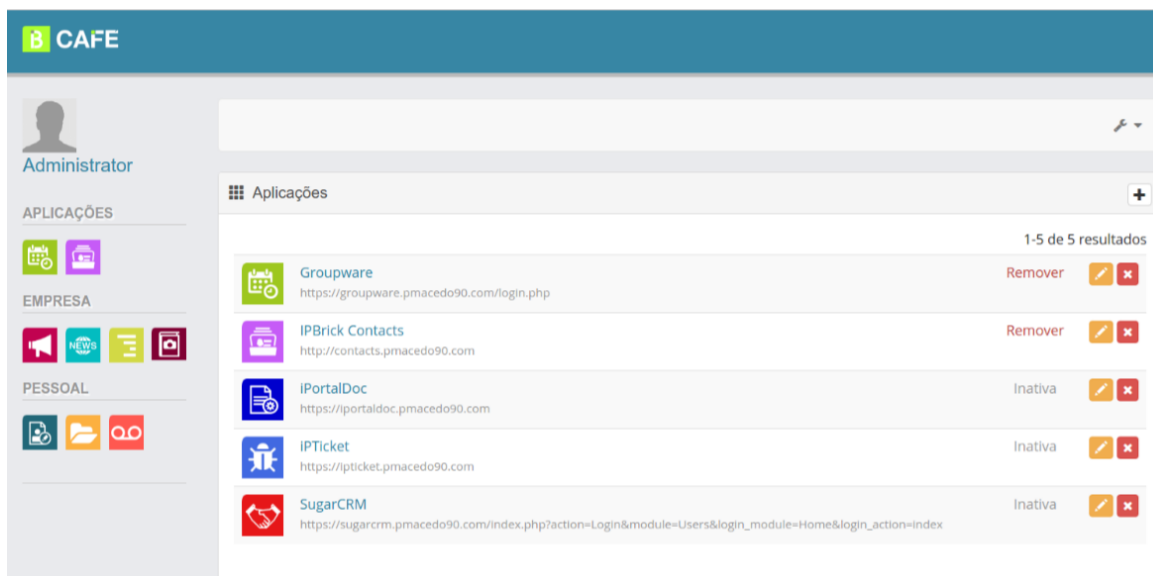


Figura 78 – Gestão de aplicações – versão antiga.

Com as modificações efetuadas (ver Figura 79), ficou possível adicionar ou remover aplicações no atalho à esquerda de uma forma mais intuitiva. Ao clicar não só o utilizador faz com que a aplicação seja adicionada aos atalhos do utilizador bem como irá aparecer uma notificação "Enabled" para alertar ao utilizador o estado da aplicação. Caso o botão seja clicado, aplicação será removida dos atalhos, aparecendo a mensagem "Disabled".

Se o atalho estiver ativo para uma aplicação, o botão estará inativo (pois não faz sentido ativar algo que já está ativo), caso estiver já inativo o botão estará inativo.

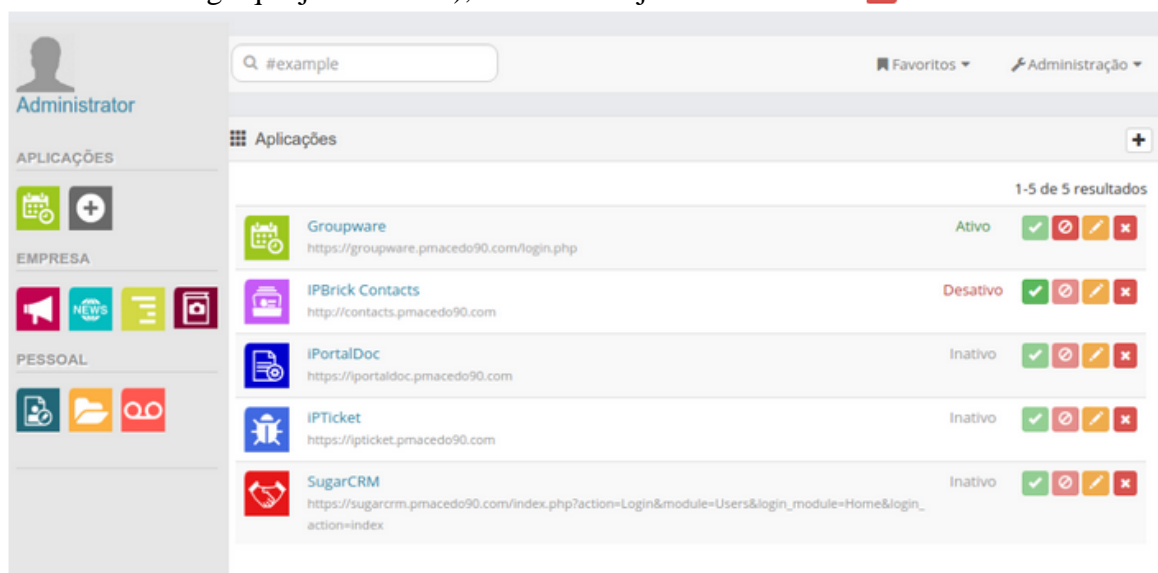


Figura 79 – Gestão de aplicações – versão atual.

3.1.9. MODIFICAÇÃO DO LAYOUT NOS CONTACTOS EXTERNOS

Na zona de adicionar um contacto externo na *UcoIP bar*, o *layout* está desorganizado. Os botões de confirmar/cancelar a criação/edição de um contacto não são muito perceptíveis, induzindo o utilizador em erro, sobretudo no que se relaciona com o botão cancelar, que parece um botão de apagar (ver Figura 80).

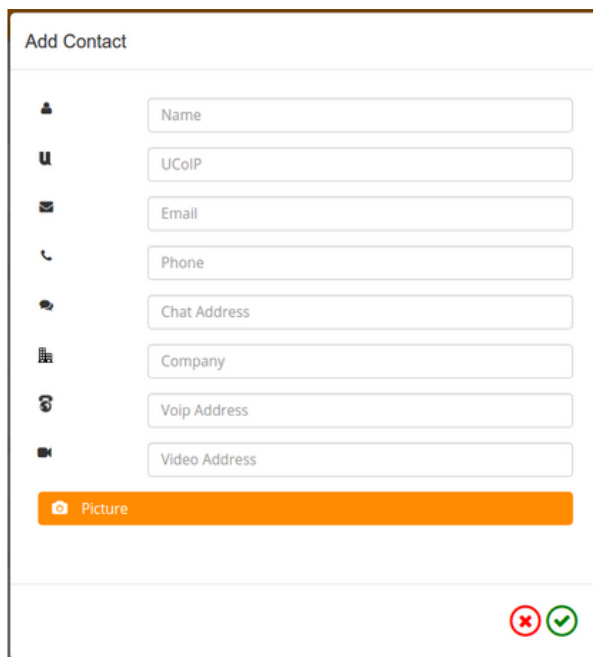




Figura 80 – Adicionar contacto externo – versão antiga.

A ideia passa por substituir os botões  e  por algo que se entenda, ou seja, botões com a frase “Gravar” e “Cancelar” (ver Figura 81).

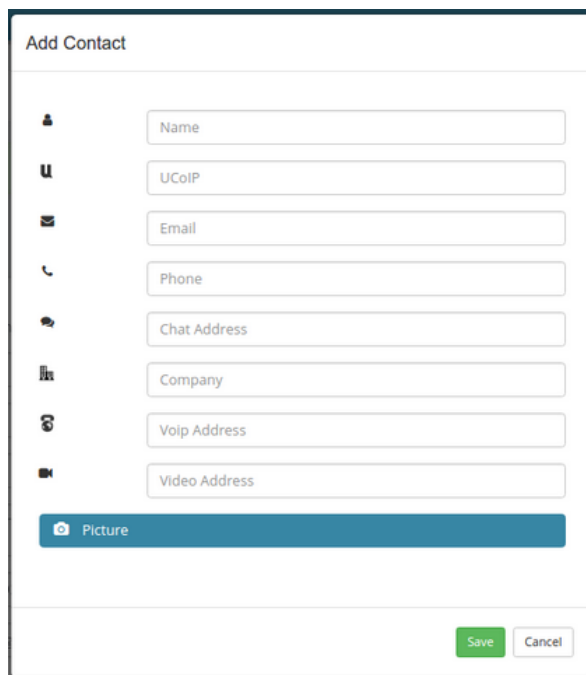


Figura 81 – Adicionar contacto externo – versão atual.

3.2. MODERADOR DE *POSTS*

O moderador de *posts* tem como objetivo controlar qualquer publicação efetuada no *CAFE*. O objetivo desta tarefa é evitar que sejam publicados *posts* de conteúdo menos apropriado para o ambiente empresarial.

Os *posts* são controlados através de um utilizador com permissões, como por exemplo o administrador, representado na Figura 82. Estas são definidas na secção administrativa do *CAFE*, mais precisamente em “Permissões”. Nesta, foi criada uma nova linha de permissão, chamada “Aprovar *feeds*”. A linha foi criada através da base de dados, com o comando:

```
INSERT INTO permission (module, action, role_id) VALUES ('newsfeed','approveFeeds',0);
```

	administrator
Feed	<input type="checkbox"/>
Criar	<input checked="" type="checkbox"/>
Alterar	<input checked="" type="checkbox"/>
Remover	<input checked="" type="checkbox"/>
Aprovar Feeds	<input checked="" type="checkbox"/>

Figura 82 – Administrador - perfil com permissões de “Aprovar *feeds*”.

Quando um utilizador é criado, por defeito não terá esta permissão ativa. Para a ter, terá de ser um administrador com privilégios necessários para lhe dar o poder de moderar *posts*.

O seguinte pedaço de código mostra a permissão e a associação do valor de base de dados ao respetivo papel.

```
//Função que cria os posts
public function actionCreate()
{
//Verifica se o moderador de posts está ativo
$feedsModeratorMode = Configuration::model()-
>findByAttributes(array('variable'=>'feedsModerator'))->value;
//Verifica as permissões do utilizador
$feedsModerator = Yii::app()->controller-
>checkPermissions('newsfeed', 'approveFeeds');
...
}
```

```

        if ($feedsModeratorMode && $feedsModerator) {
            //Valor 1 -> quando é o moderador, o post é aprovado
            automaticamente
                $model->approved = 1;
            }
        else {
            //Valor 2- estado pendente quando não é moderador
                $model->approved = 2;
            }
        ...

```

Como é possível verificar na Figura 49, foi criado, na área das configurações, foi criada uma *checkbox* para ativar/desativar a funcionalidade da moderação de *posts*.

```

// Verifica o valor na base de dados (1 ou 0)
$feedsModerator = Configuration::model()->
findByAttributes(array('variable'=>'feedsModerator'));

// Cria a checkbox e verifica o valor que está na base de dados. Se
// o valor for 1, a box estará ativa
<input type="checkbox" name="Configuration[<?php echo
$feedsModerator->variable; ?>]" value=1 <?php if ($feedsModerator-
>value == 1) { echo 'checked'; } ?> /> <?php echo Yii::t('app',
'Feeds Moderator'); ?>

```

Sempre que um utilizador publica um *post* no *CAFE*, o moderador recebe uma notificação a alertá-lo que uma publicação está pendente de aprovação (ver Figura 83).

```

public function actionNew()
{
    ...
    // Variável que verifica as notificações pendentes
    $n_pending_newsfeed = 0;

    // Retorna as notificações presentes na base de dados
    $newNotifications = Notification::model()-
>findAllByAttributes(array('to'=>Yii::app()->user-
>id,'read'=>false),array( 'order' => 'datetime desc', 'limit' =>
$limit ));

```

```

...
//Conta o número de notificações
$n_pending_newsfeed = count($pending_newsfeed);
...
//Cria classe vazia
$obj = new stdClass();
//Se houverem notificações, conta-as
if ($n_pending_newsfeed > 0) {
    $obj->total = count($notifications) + 1;
    $obj->new = count($newNotifications) + 1;
} ...
//Imprime o número de notificações
echo json_encode($obj);

```

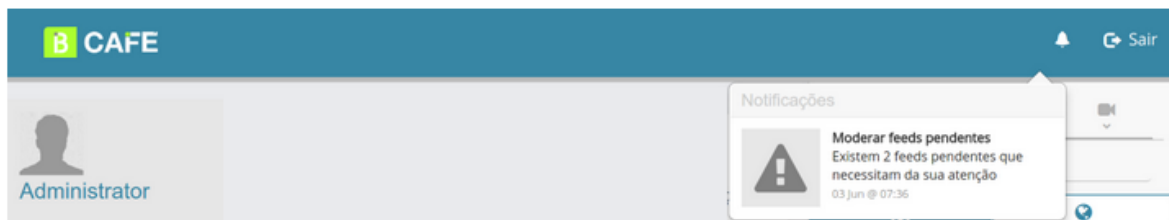


Figura 83 – Notificações: dois feeds por aprovar.

Quando um *post* é publicado por um utilizador, o administrador irá visualizar esse mesmo *post* publicado apenas na sua página inicial do *CAFE*, e irá decidir se o *post* é aprovado ou rejeitado. Para tal é utilizado um ícone do *font-awesome*, com o aspeto de um olho. Caso o olho esteja amarelo, a publicação estará pendente de aprovação/reprovação, se estiver verde, o *post* estará aprovado, se seleccionar vermelho o *post* será rejeitado e não será publicado para o resto dos utilizadores.

```

// Se a publicação for aprovada, é publicada e é sinalizado com o
// ícone do olho verde (ver canto superior da Figura 84)
if ($data->approved == 1) {
    echo CHtml::link('<span class="fa fa-eye"></span>', null,
array('id'=>"approvebutton_". $data->id, 'title'=>Yii::t('app',
'Moderate Feed'), 'onclick'=>'approveFeed(' . $data->id . ',0);',
'style'=>'padding: 2px 3px; color: green; cursor: pointer; border-
radius: 2px;'));

```

```

// Se a publicação for rejeitada, o post não é publicado e é
//sinalizado com o ícone do olho vermelho (ver canto superior
//direito da Figura 85)
    } elseif ($data->approved == 0) {
        echo CHtml::link('<span class="fa fa-eye-slash"></span>',
null, array('id'=>"approvebutton_". $data-
>id,'title'=>Yii::t('app', 'Moderate Feed'),
'onclick'=>'approveFeed(' . $data->id . ',1);', 'style'=>'padding:
2px 3px; color: red; cursor: pointer; border-radius: 2px;'));

// Se não for nenhuma das opções, o olho fica laranja e a
//publicação pendente (ver canto superior direito da Figura 86)
    } else {
        echo CHtml::link('<span class="fa fa-eye"></span>', null,
array('id'=>"approvebutton_". $data->id,'title'=>Yii::t('app',
'Moderate Feed'), 'onclick'=>'approveFeed(' . $data->id . ',1);',
'style'=>'padding: 2px 3px; color: orange; cursor: pointer;
border-radius: 2px;'));
    }
}

```

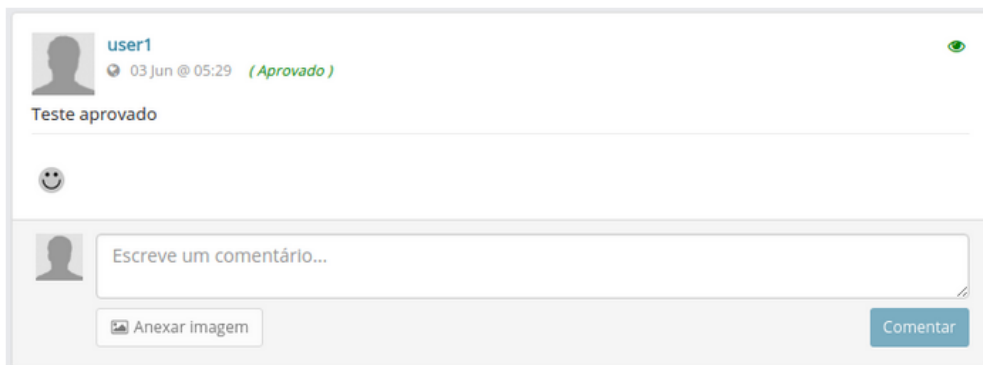


Figura 84 – Publicação aprovada.

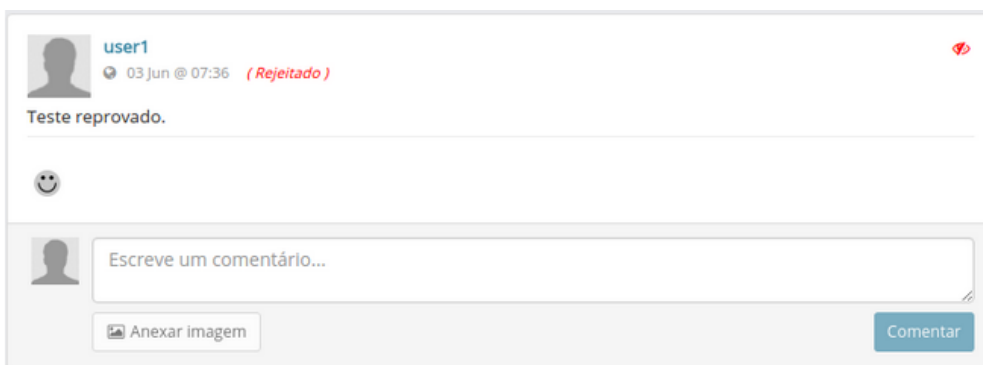


Figura 85 – Publicação rejeitada.

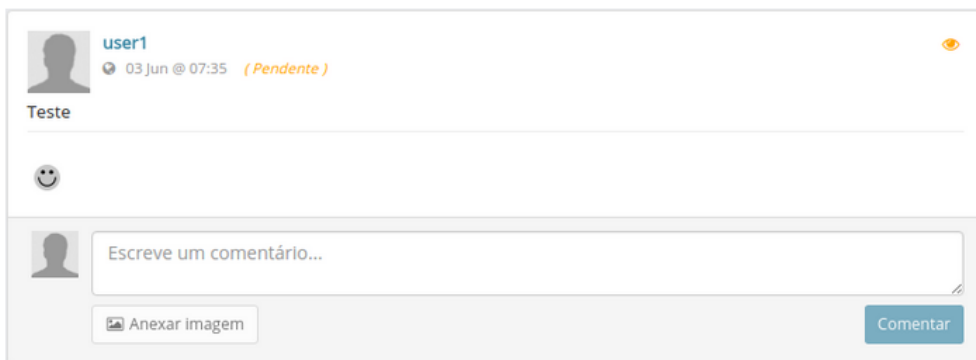


Figura 86 – Publicação pendente.

3.3. AGENDA DE EVENTOS

A agenda de eventos tem como objetivo de lembrar o utilizador dos acontecimentos mais relevantes ao longo do ano. Funciona como uma agenda das atividades da empresa: pode incluir eventos, conferências, lançamentos de produtos, entre outros, evitando que o utilizador se esqueça dos eventos a suceder.

A agenda inclui o calendário genérico, com as datas e os eventos. O utilizador pode criar o seu próprio calendário, com as atividades internas da empresa. No fundo, serve para que a empresa tenha um caminho a seguir, uma estratégia definida.

O calendário utilizado pela *IPBRICK* é o calendário do *Groupware*, representado na Figura 87. Neste são guardados e partilhados (se o criador o desejar), os eventos criados. Já outro utilizador pode querer receber ou não o que é partilhado por outros utilizadores.

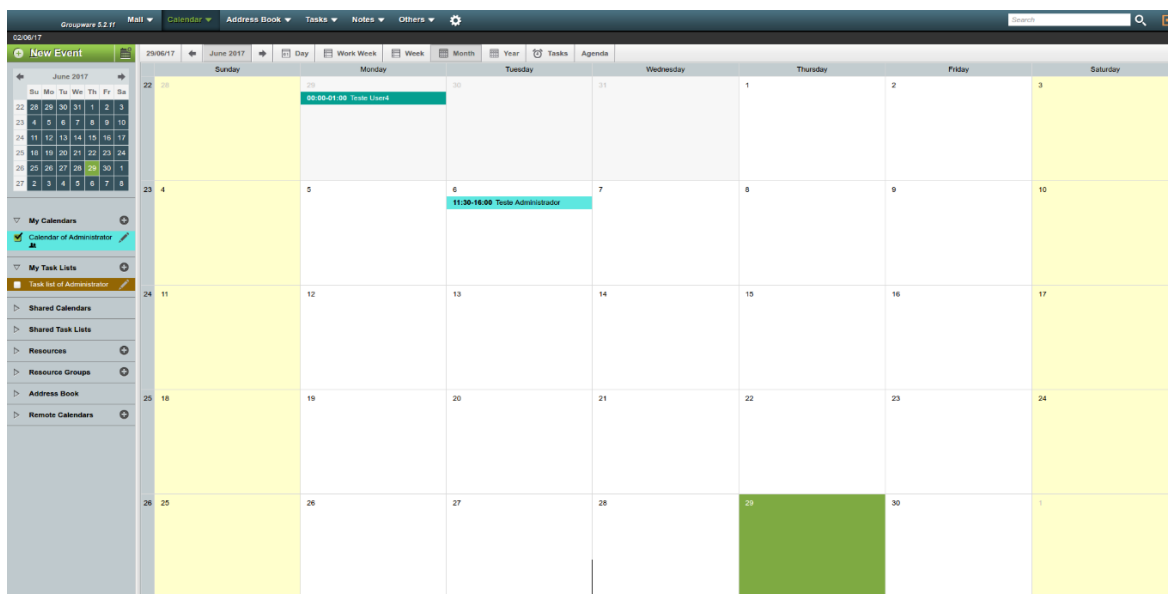


Figura 87 – Calendário do Groupware.

O objetivo desta tarefa é enviar as informações dos eventos para a interface principal do *CAFE*.

Para tal, foram seguidos os seguintes passos:

1. Fazer *download* da pasta “simpleCalDAV”, no *site* do *Groupware*: esta pasta contém a biblioteca que faz a ligação do servidor *CalDav* (função que cria o calendário no *Groupware*) e a lista de funções que mudar, apagar e criar novos calendários, sendo bastante útil para conseguir ler a informação dos mesmos. De seguida, o utilizador mover esta pasta para o código do *CAFE*;
2. Criar na base de dados uma coluna que guarde a informação dos calendários lidos no *Groupware*; A seguinte linha de código mostra em *SQL* a adição desta coluna:

```
ALTER TABLE profile ADD shcalendar VARCHAR(200);
```

3. De seguida, foi acrescentado o seguinte código no intuito de ler a informação guardada no calendário do *Groupware*;

```
//Lê a data e hora de início do mês corrente
$startdate = date('Y-m-d H:i:s',
strtotime($_POST['startDate']));
$startdate = explode('-', $startdate);

//Dia 1 do mês corrente
$start = $startdate[0]."-".$startdate[1]."-01";
$end = explode(' ', date("Y-m-t", strtotime($start)));

//Último dia do mês corrente
$end = $end[0];

//Lê o nome do utilizador
$username = Yii::app()->user->getState('login');

//Lê o domínio da máquina
$domain = Configuration::model()-
>findByAttributes(array('variable' => 'ipbrick_chat_default'))-
>value;
$parts = explode(".", $domain);
```

```
//Retorna parte do domínio da máquina
$domain = implode('.', array_slice($parts, 1));

//Ligação aos calendários do Groupware
$client = new SimpleCalDAVClient();
//Liga-se ao calendário do utilizador
$client-
>connect('http://groupware.'. $domain.'/rpc.php/principals/'. $usern
ame.'//', $username, Yii::app()->user->getState('password'));

// Devolve todos os calendários ao utilizador (calendários
//partilhados de outras pessoas)
$arrayOfCalendars = $client->findCalendars();
```

4. Após se ter ligação ao calendário, criar algum evento no *Groupware*, para que o código leia esse evento (ver Figura 88).

02/06/17	← June 2017 →	Day	Work Week	Week	Month	Year	Tasks	Agenda
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
22	28	29	30	31	1	2	3	
23	4	5 00:00-01:00 Teste user4	6 11:30-16:00 Teste Administrador	7	8	9	10	

Figura 88 – Eventos criado no *Groupware*.

5. Após a criação dos eventos visíveis na Figura 88, estes poderão ser visualizados na página principal do *CAFE* (ver Figura 89). Note-se que são publicados não só os eventos do próprio utilizador, como de todos os utilizadores que permitem partilhar calendário.

Figura 89 – Interface dos eventos criados no *Groupware* na página principal do *CAFE*.

6. Criação de setas e *datepicker* para mudar o mês: para visualizar os eventos de cada mês, foram criadas setas e um *datepicker* (de mês), representada na Figura 90.

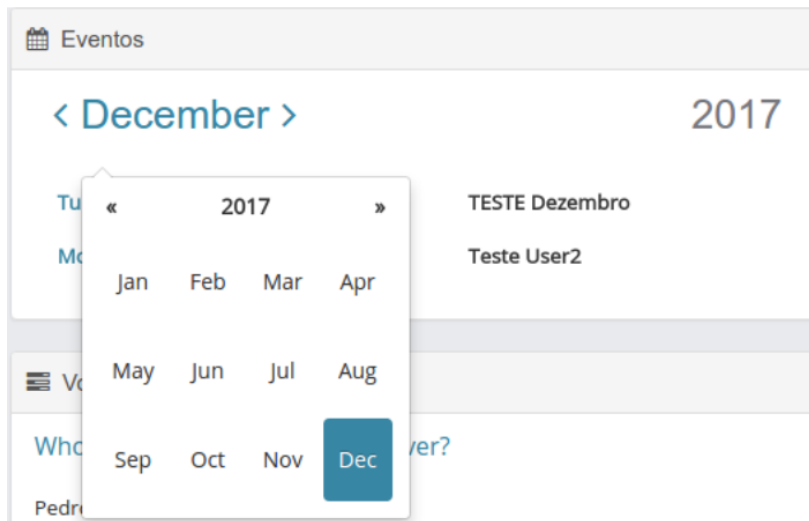


Figura 90 – Datepicker: escolha o mês para a visualização dos eventos.

7. Na área da mudança de perfil, foram criadas *checkboxes* para todos os utilizadores que partilharam o seu calendário. Estas *checkboxes* permitem a qualquer utilizador escolher quais os calendários que deseja ver na sua página principal do *CAFE*. Os calendários foram guardados na base de dados (ver Figura 91).

```

cafe=# SELECT name, shcalendar from profile;
name | shcalendar
-----|-----
user3 | all
user1 | calendar-eFXZLbH2334GIbCpr6rXTWu,calendar-uB5jbWkjosqtudK6cf5HZ_T,calendar-YV1QDA4xFW_oGCBs9F8AJ2J,calendar-jF0Kp8HFA4JNZJGkm
user5 | calendar-uB5jbWkjosqtudK6cf5HZ_T,calendar-YV1QDA4xFW_oGCBs9F8AJ2J
user2 | all
user4 | calendar-eFXZLbH2334GIbCpr6rXTWu
Admin | none
Administrator | calendar-eFXZLbH2334GIbCpr6rXTWu,calendar-YV1QDA4xFW_oGCBs9F8AJ2J,calendar-jF0Kp8HFA4JNZJGkmVuaMFp
(7 rows)

```

Figura 91 – Calendários guardados para cada utilizador.

Na interface da mudança de perfil, foram geradas as *checkboxes* lidas no calendário do *Groupware* (representadas na Figura 92). Nesta o utilizador pode então escolher quais os calendários que quer visualizar na interface principal do *CAFE*.

Calendario

- Calendar of Administrator
- Calendar of user1 [user1]
- Calendar of user2 [user2]
- Calendar of user4 [user4]

Figura 92 – Calendários: o utilizador escolheu 3 dos 4 para visualização na página principal.

8. Por fim foi criado, na área das configurações, uma *checkbox* com o objetivo de o utilizador poder fazer aparecer/omitir o calendário (ver Figura 93).

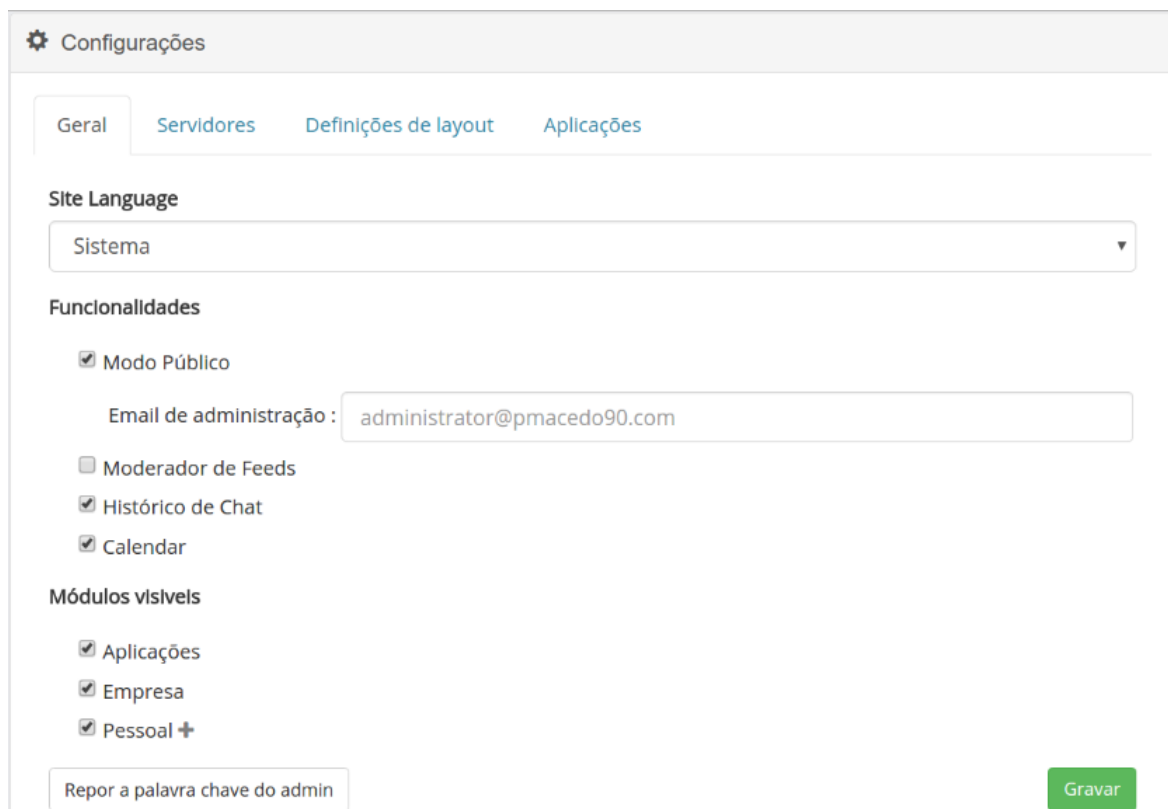


Figura 93 – Nesta secção é possível ocultar os eventos.

3.4. REALIZAÇÃO DE TESTES

Para o lançamento da versão 2.4 do *IPBRICK CAFE*, foi necessário a realização de vários testes para confirmar se todas as tarefas realizadas pela equipa do *CAFE* foram bem-sucedidas e se estas estão prontas para entrar na nova versão. Para tal, as tarefas concluídas entre os vários membros foram testadas por membros diferentes.

O método utilizado para a realização dos testes foi o modo *black box*. As tarefas testadas foram relacionadas com as funcionalidades móveis e do seu progresso desde Dezembro de 2016.

Realizaram-se os seguintes testes:

- Notícias com imagens com *links* desformatadas em *mobile*: nesta tarefa as imagens numa publicação saíam fora do limite do ecrã. Para testar esta funcionalidade corrigida, basta ao utilizador criar um *post* com imagem no *CAFE*, e de seguida verificar se a imagem ficou dentro dos limites definidos para *mobile*;

- *Mobile* - melhorar o *design* da área do álbum: para a verificação desta tarefa, o utilizador necessita de ir à área do álbum e testar todas as funcionalidades do mesmo (criar álbum, apagar álbum, adicionar fotografias, remover fotografias, verificar se as formatações estão corretas, entre outras funcionalidades);
- *Popup* imagens/vídeos não estão a redimensionar-se para *mobile*: quando se clica numa imagem para ver em ecrã inteiro, esta fica demasiado grande a ocupar o telemóvel e o botão de saída fica impossível de ser visto. Para testar se este problema foi resolvido, o utilizador necessita de clicar numa imagem (por exemplo a de um *post*) e verificar se a imagem está com as dimensões corretas;
- *Mobile* – alterar *layout* principal do *CAFE*: a página principal do *CAFE* para *mobile* foi completamente reconstruída, sendo melhorado o *layout* da mesma. Para testar se tudo continua a funcionar da mesma maneira que a versão anterior, o utilizador necessita de testar todas as funcionalidades que a página principal possui;
- *Link* não é aberto de forma correta: ao colar um link, o *CAFE* deveria identificar o *link* e descartar o ponto final da frase, senão o link não é aberto de forma correta. Para verificar se o problema foi resolvido, o utilizador deve criar um *post* no *CAFE*, referindo o *link* de um *website*, colocando um ponto final no final do *link*, por exemplo “Vídeos em www.youtube.com.”.

3.5. CONCLUSÃO

As tarefas propostas pela *IPBRICK* foram realizadas com sucesso. Ao longo da Tese, os objetivos definidos inicialmente foram ligeiramente modificados pela empresa, de modo a ajustar as prioridades de melhoramentos a realizar no *CAFE*.

As tecnologias mencionadas no Capítulo 2 foram indispensáveis para a realização das tarefas realizadas no presente capítulo. Sem elas, não seria possível alcançar os objetivos propostos pela *IPBRICK*.

O *CAFE* é uma rede social empresarial em constante crescimento, por este motivo nunca irão faltar novas tarefas a desenvolver, pois há sempre algo novo para criar ou para melhorar.

4. CONCLUSÃO

Quando surgiu a *internet*, poucas eram as pessoas que tinham a ideia de quanto esta mudaria a comunicação, o comportamento das pessoas e, como consequência, todo o mundo.

As empresas também descobriram no mundo *web* novas formas de fazer negócio. Seja com aplicações, *sites*, publicidade ou redes sociais, cada vez mais empresas se aventuram a criar novos formatos de negócios para as várias plataformas que foi capaz de juntar todos os meios num só.

O *CAFE* revelou ser uma rede social empresarial com capacidades de satisfazer as necessidades de comunicação entre todos os membros numa empresa. Usamo-la para não só para comunicar nos vários formatos disponíveis (mensagem, *email*, telefone e videoconferência), mas para também para partilhar ideias, notícias, fotografias, relembrar eventos, criar votações, gerir aplicações dentro do próprio *CAFE*, entre outros.

Ao longo da Tese, foram concretizados os seguintes objetivos:

- Execução dos guiões de formação da *IPBRICK OS*: para integrar o formando no mundo da *IPBRICK*;
- Criação da capa para o álbum de fotografias: de modo a que o criador do álbum possa escolher a fotografia de capa para o seu álbum;

- Correção das permissões na área do álbum: para impedir os utilizadores de criarem/apagarem fotografias e álbuns sem permissão;
- Correção da cor do *menu* do *background* ao criar um álbum: a cor passou a ser a cor de fundo do cabeçalho, escolhido pelo utilizador;
- Imagem das notícias por *upload*: permite ao criador da notícia facilitar a colocação de imagens e guarda-las numa *cloud*;
- Melhorar a página das configurações, na área da administração: acrescentar funcionalidades como as *checkboxes* para fazer aparecer/ocultar o moderador de *posts* e o calendário de eventos. Remodelação da *tab* “Definições de *layout*”;
- Melhorar o esquema *enable/disable* na secção de utilizadores, na área de administração: mudar a forma de seleccionar utilizadores e associa-los a um perfil;
- Remodelar a secção dos servidores de autenticação, na área de administração: em vez de aparecerem os “Grupos proibidos”, aparece agora “Grupos permitidos”. Mudança da forma de como se seleccionam os grupos para permissão;
- Remodelar a secção das aplicações, na área da administração: modificar a forma de como se permite ou bloqueia uma aplicação, tornando-a mais intuitiva;
- Remodelar o *layout* na área de edição de perfil: tornar esta área mais apelativa e moderna para o utilizador. Criação da secção “Calendário”, para escolher os calendários que o utilizador quer visualizar na página principal do *CAFE*;
- Correção dos botões do *CafePhone*: agora não aparecerem após o aumento do tamanho do cabeçalho;
- Mudar a interface dos contactos externos: pequena alteração do *layout*, sobretudo nos botões;
- Criação de um moderador de *posts*: é possível permitir ou não um *post* de ser publicado, antes de ele ser publicado;
- Criação do calendário de eventos: os eventos criados no *Groupware* são agora extraídos para a interface principal no *CAFE*;
- Realização de testes para o lançamento da versão 2.4 do *IPBRICK CAFE*: testar todas as tarefas efetuadas por outros membros da equipa do *CAFE*.

A evolução da tecnologia é constante, e o *IPBRICK CAFE* não escapa a esta regra. Tendo isso em conta, existem outros objetivos num futuro próximo que poderão ser úteis para o desenvolvimento desta rede social, como os seguintes:

- Dados estatísticos das aplicações mais e menos acedidas: medir o número de cliques de cada aplicação, para verificar quais são as aplicações mais e menos acedidas no *CAFE*.
- *CKEDITOR*: *Banner* da janela de *upload* de imagens deveria ter a mesma cor do *header* do *CAFE*;
- Possibilitar a abertura completa das notícias sem fazer *login*: é possível ver as notícias criadas no *CAFE* sem fazer *login*, mas se o utilizador clicar na opção “Ler mais”, será forçado a fazer *login*. O objetivo desta tarefa passa por poder ver as notícias por completo sem necessitar fazer *login*.
- Nomes extensos nos álbuns: quando um álbum tem um nome longo, o botão "edit album" deixa de funcionar. No momento em que se tenta editar o nome do álbum, este abre, impossibilitando desta forma a alteração do nome do mesmo.
- Envio de *email* a um utilizador: nos utilizadores externos, o *hover* do botão para enviar *email* a utilizador (*UcoIPbar*) continua com o cursor *default*, quando devia estar um *pointer* dado que a hiperligação funciona.

A criação de novos módulos e as várias correções efetuadas no *CAFE* no decorrer deste Estágio Curricular foram importantes na evolução das capacidades do autor no que se relaciona com a programação *web*, sendo este um método recomendável para quem está a começar a entrar no mundo do trabalho. A *IPBRICK* acompanhou e ajudou em todos os passos necessários para o sucesso deste Estágio, sendo todas as dúvidas e obstáculos ultrapassados.

Os processos de resolução de todas as tarefas foram semelhantes: em primeiro lugar estudar o problema; em segundo procurar possíveis soluções; em terceiro construir a solução; em quarto implementa-la. Este processo achou-se o mais adequado para resolver problemas no desenvolvimento de ferramentas no mundo tecnológico.

O mundo *web* é um campo que está e estará sempre em constante evolução, sendo que as redes sociais terão um papel fulcral no que toca à comunicação em todo o mundo, fortificando assim o conceito de aldeia global.

Referências Documentais

- [1] *IPBRICK SA TEAM* — “Reference Guide”, Porto, Portugal, 2016.
- [2] *IPBRICK SA TEAM* — “*IPBRICK* – That’s the way!”, Porto, Portugal, 2010.
- [3] *IPBRICK SA TEAM* — “*IPBRICK CAFE* v1.6 – Manual de utilização”, Porto, Portugal, 2006.
- [4] RAMALHÃO, M. — “Proposta de Tese/Dissertação”, *IPBRICK*, Porto, Portugal, 2016.
- [5] ROCHA, C. – “*IPBRICK* – Contact Center para gestão de suporte a clientes”, FEUP, Porto, Portugal, 2016.
- [6] *IPBRICK*- “*IPBRICK OS - IPBRICK*”, <http://www.IPBRICK.com/pt-pt/IPBRICK-os/>, acessido em Maio de 2017.
- [7] DELGADO, L. — “Informática Básica”, Escola Secundária Barra do Corda, Brasil, 2016.
- [8] FERREIRA, E. ; EIS, D. – “*HTML* - Curso W3C Escritório”, Brasil, 2010.
- [9] ARSÉNIO, A.— “ *PHP*: Hypertext Preprocessor – Composição Web ”, UTAD, 2015.
- [10] SOARES, N. – “PubliSite”, ISEP, Porto, Portugal, 2003.
- [11] GIROLDO, B.; FRESSATI, W. – “EVOLUÇÃO NO DESENVOLVIMENTO COM O *HTML 5*”, Universidade paranaense, 2015.
- [12] MENINO, R. – “Cascading Styles Sheet”, Universidade do Paraná, Brasil, 2012.
- [13] ALEXANDRE, L. – “Plataforma emprego”, Instituto Politécnico de Bragança, Bragança, Portugal, 2016.
- [14] TOLEDO, T. – “Library Mapper”, Instituto de Matemática e Estatística, Universidade de São Paulo, Brasil, 2011.
- [15] LANE, T. – “Transaction Processing in PostgreSQL”, GreatBridge, Estados Unidos da América, 2010.
- [16] KINOSHITA, V. - “BANCO DE DADOS VIA WEB: UMA ANÁLISE COMPARATIVA”, Universidade das Minas Gerais, Brasil, 2001.

- [17] JUNIOR, B. – “Programação web – Aula 7”, Departamento de informática, Universidade Federal do Paraná, Brasil, 2014.
- [18] CASCAO, T. – “Sistema de Informação para Produção”, FEUP, Porto, Portugal, 2014.
- [19] JÚNIOR, N. – “PADRÕES DE PROJETO E FRAMEWORK NO DESENVOLVIMENTO DE SOFTWARE”, UNIPAC-Barbacena, Brasil, 2009.
- [20] BELÉM, P.; SOUZA, A.; MEZA, E.; VIANNA, A. – “ESCOLHA DE UM FRAMEWORK PARA A LINGUAGEM DE PROGRAMAÇÃO *PHP* ATRAVÉS DO MÉTODO AHP CLÁSSICO”, Universidade Federal Fluminense, Brasil, 2014.
- [21] GERALDES, D. – “*PHP* Frameworks – *IPBRICK* secção 1”, *IPBRICK*, Porto, Portugal, 2017.
- [22] *YII* TEAM- “*Yii* 1.1: *Yii* for beginners”, <http://www.Yiiframework.com/wiki/250/Yii-for-beginners/>, acessido em Maio de 2017.
- [23] OLIVEIRA, I. – “Interface de Usuário: A Interação Homem-Computador Através dos Tempos”, Faculdades Associadas de Ariquemes, Brasil, 2010.
- [24] CANHA, P. – “Design Básico: Diferença entre UX e UI Design”, <https://chocoladesign.com/design-b%C3%A1sico-diferen%C3%A7a-entre-ux-e-ui-design-18c81154998f>, acessido em Maio de 2017.
- [25] MORI, N; INEIA, D. – “ABORDAGEM E IMPLANTAÇÃO DO PROTOCOLO *XMPP*”, Universidade Paranaense, Brasil, 2009.
- [26] MACHADO, V. – “*XMPP* – O protocolo da moda para serviços de Messaging”, <https://pplware.sapo.pt/software/deixe-as-apps-de-im-mais-usadas-e-opte-pelas-que-usam-XMPP/>, acessido em Maio de 2017.
- [27] CAMPOS, R. – “Serviços Telemáticos”, Instituto Politécnico de Tomar, Portugal, 2011.
- [28] APACHE TEAM – “Apache HTTP Server Documentation Version 2.4”, Estados Unidos da América, 2015.
- [29] MACHADO, E. – “Autenticação Integrada Baseada em Serviço de Diretório *LDAP*”, Universidade de São Paulo, Brasil, 2006.
- [30] RUSSO, R. – “Redes – O que é e para que serve o serviço *LDAP*?”, <http://escreveassim.com/2011/08/09/servico-LDAP/>, acessido em Maio de 2016.

- [31] ALECRIM, E. – “Servidor SAMBA: o que é?”, INFO WESTER, <https://www.infowester.com/linuxsamba.PHP>, acessado em Maio de 2017.
- [32] FERNANDES, H. – “Framework Para Aplicações Colaborativas Multi-Plataforma”, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Portugal, 2010.
- [33] PALUMBO, A.; BONIFÁCIO, L.; SANTOS, M. – “Desenvolvimento do software gerenciador de bibliotecas escolares”, Botucatu, São Paulo, Brasil, 2016.
- [34] NetBeans TEAM – “NetBeans Logo Vector”, <https://seeklogo.com/vector-logo/98428/netbeans>, acessado em Maio de 2017.
- [35] SILVA, W. – “NetBeans 4.1 primeiros passos”, NetBeans Brasil, 2002.
- [36] RADONIC, A. – “Managing hypervisors with the Archipel vM orchestration tool”, Intersales AG, Colónia, Alemanha, 2011.
- [37] NEITZKE, N. – “*Bootstrap*”, Universidade Luterana do Brasil, 2017.
- [38] SILVA, M. – “*Bootstrap 3.3.5*”, Maujor, São Paulo, Brasil, 2015.
- [39] VENTURA, R. – “Robô bombeiro - Aplicação Web para Gestão de Equipas”, Instituto Politécnico da Guarda, Portugal, 2014.
- [40] SOUZA, O. – “Processos de apoio ao desenvolvimento de aplicações web”, Universidade de São Paulo, Brasil, 2005.
- [41] DIAS, H. – “Font Awesome – Excelente para designers”, <https://henriquedias.com/blog/font-awesome-excelente-para-designers/>, acessado em Junho de 2017.
- [42] AUZAC, L. – “Tipos de Pentest”, <http://www.auzac.com.br/testes-de-invasao/tipos-de-pentest/>, acessado em Junho de 2017.
- [43] HAYEN, R.; SWABY, S.; HUANG, Z. – “USE OF GROUP SUPPORT SYSTEMS IN TODAY’S SOCIETY”, Central Michigan University, EUA, 2007.
- [44] MARQUES, J. – “Desenvolvimento de software para web usando Java para uma aplicação de comércio eletrônico”, Fundação Educacional d Município de Assis, Brasil, 2011.
- [45] MAESTRELLI, M. – “WEB SERVER SEGURO - APACHE”, CATE/CBFC, Brasil, 2000.
- [46] GEHRMANN, B. – “Cervisia Manual”, Alemanha, 2014.

Anexo A. Guiões de aprendizagem da *IPBRICK OS*: Instalação da *IPBRICK OS*

Os seguintes anexos apresentam os guiões de integração da *IPBRICK OS*, para uma melhor compreensão deste sistema operativo.

INSTALAÇÃO DA *IPBRICK OS*

Por defeito a *IPBRICK* tem ativada a instalação no modo automático, onde o utilizador apenas necessita de preparar a máquina, de modo a que ela arranque a partir do *DVD*.

A *IPBRICK* permite ao utilizador dois modos de instalação alternativos:

- Modo *debug*: O utilizador passa a ser informado a partir da consola daquilo que especificamente está a ser executado pelo *DVD* de instalação. Para ativar este modo basta ao utilizador escrever *IPBRICK verbose* na linha de comando do *boot*;
- Modo manual: O próprio utilizador é responsável pela instalação, sendo feita também em modo *debug*. Para ativar este modo basta escrever *manual* na linha de comando *boot*.

REINSTALAÇÃO LIMPA

Num sistema onde não esteja instalado um servidor *IPBRICK*, a instalação será sempre limpa. Caso exista uma instalação *IPBRICK*, o processo de instalação automática assumirá uma atualização e não uma instalação.

Para forçar uma instalação limpa, é necessário durante a instalação da *IPBRICK*, colocar na linha de comandos o comando *IPBRICK eraseall*. Outra alternativa é remover a pasta *opt/ipbox* pela consola da *IPBRICK* antes de iniciar a instalação: *rm -rf /opt/ipbox*.

GESTÃO DE CONFIGURAÇÕES

A *IPBRICK* permite uma gestão de configurações eficaz onde sempre que são efetuadas alterações no sistema pela interface *https://*, uma nova configuração é guardada localmente ou automaticamente numa *flash drive*. Ao mesmo tempo é enviada uma cópia para o endereço de *email* configurável. A gestão de configurações é realizada a partir da interface *web* no menu *Advanced Configurations* → *Disaster Recovery* → *Configurations* (ver Figura 94). A partir desta, pode-se escolher uma configuração guardada anteriormente.

The screenshot shows the IPBRICK OS configuration management interface. The top navigation bar includes the IPBRICK OS logo, version v6.2, user admin, and a logout button. The sidebar on the left contains menu items: Directory, Groupware, UCoIP, Security, and Advanced Configurations (with sub-items: IPBrick, Telephony, Network, Virtualization, Support services, Disaster recovery, System). The main content area is titled 'Configurations » Replace' and shows a 'Configurations List' table. The table has three columns: Settings, Administrator, and Description. The data rows are as follows:

Settings	Administrator	Description
2017-04-10 10:38:50	admin	
2017-04-07 17:38:00	admin	
2017-04-07 13:21:58	admin	Criacao de utilizadores e grupos
2017-04-03 12:59:36	admin	Ipbrick OS com os 4 downloads da 6 2
2017-01-13 13:18:42	admin	configuracoes iniciais
Default	admin	Default Configuration.

Figura 94 – IPBRICK OS: Gestão de configurações.

ATUALIZAÇÃO

A atualização para a versão mais recente da *IPBRICK OS* é feita utilizando uma instalação automática onde o processo de instalação vai detetar a existência de uma versão anterior. O teste para detetar a existência de uma *IPBRICK OS* instalada é a existência da pasta */opt/ipbox*. Nesta situação o procedimento automático de instalação irá manter a tabela de partições atual formatará as partições de sistema mantendo inalteradas as áreas de trabalho dos utilizadores.

No final da instalação a *IPBRICK OS* inicia com as configurações por defeito. O ficheiro com as configurações da versão anterior deverá ser migrado para a nova versão *IPBRICK OS* recentemente instalada, o que é efetuado pela equipa da *IPBRICK OS*.

Anexo B. Guiões de aprendizagem da *IPBRICK OS*: Servidor de *Intranet* e *Groupware*

A *IPBRICK* como servidor de Intranet, tem como função fazer a gestão de todos os recursos de uma rede pertencentes a um determinado domínio, assim como disponibilizar serviços básicos de uma rede com importância como: *email*, *DNS* e *DHCP*.

SERVIDOR DE DOMÍNIO

Um aspeto importante do servidor de domínio é o facto de ele agir como servidor de autenticação, onde todos os utilizadores têm o seu *username* e *password* definido na base de dados *LDAP* da *IPBRICK*. Para colocar a *IPBRICK* como servidor de autenticação num determinado domínio é necessário aceder a *Servidor de domínio* → *Modificar* (ver Figura 95).

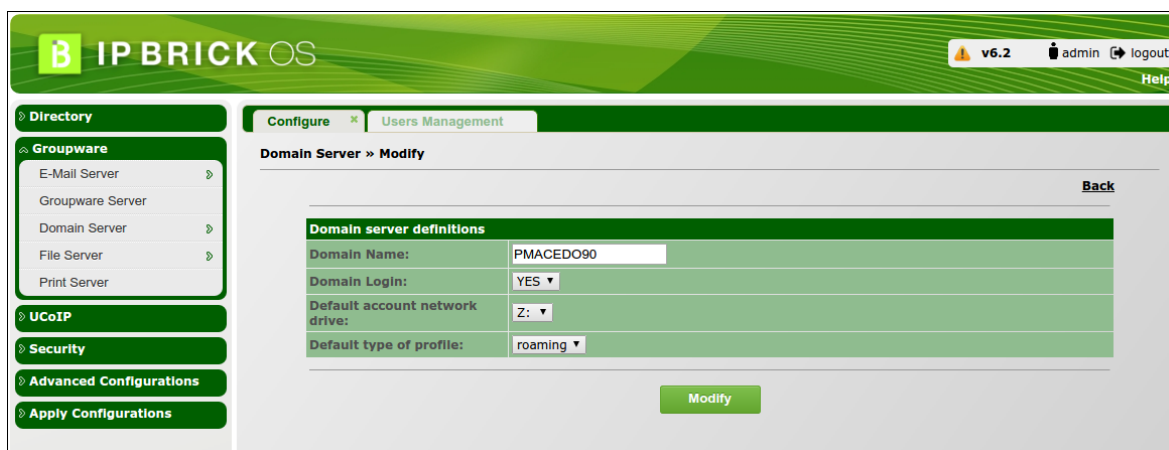


Figura 95 – *IPBRICK OS*: Configuração do servidor de domínio.

GESTÃO DE UTILIZADORES

Nesta secção podem ser criados, editados e removidos utilizadores no servidor. Será também ser criada uma área de trabalho para o utilizador correspondente a um espaço físico em disco no servidor, que irá armazenar ficheiros específicos do utilizador.

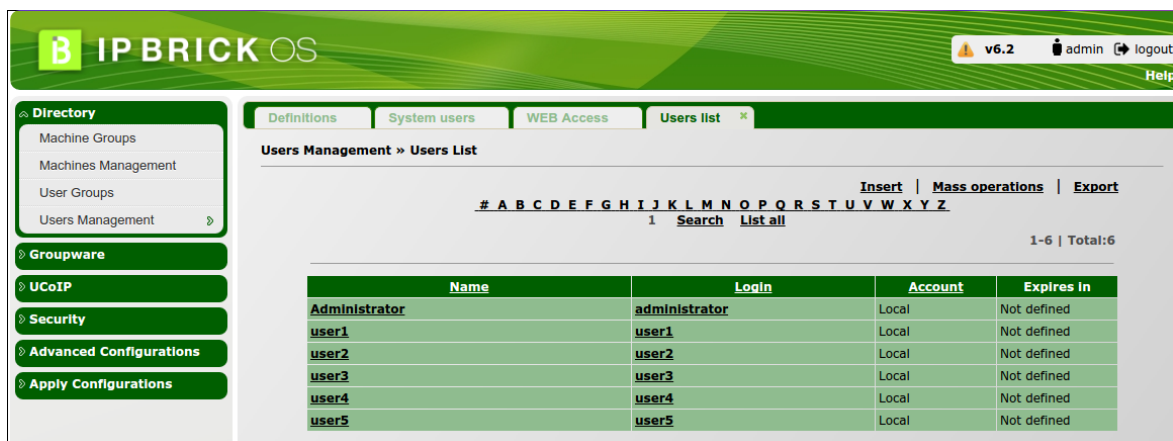


Figura 96 – IPBRICK OS: Gestão de utilizadores.

GESTÃO DE GRUPOS

O IPBRICK OS tem a capacidade de criar e gerir grupos de utilizadores. A vantagem prende-se com o facto de agrupar utilizadores com necessidades semelhantes, sendo útil por exemplo no caso dos departamentos nas empresas. Os grupos podem ser criados como se pode ver na Figura 97.

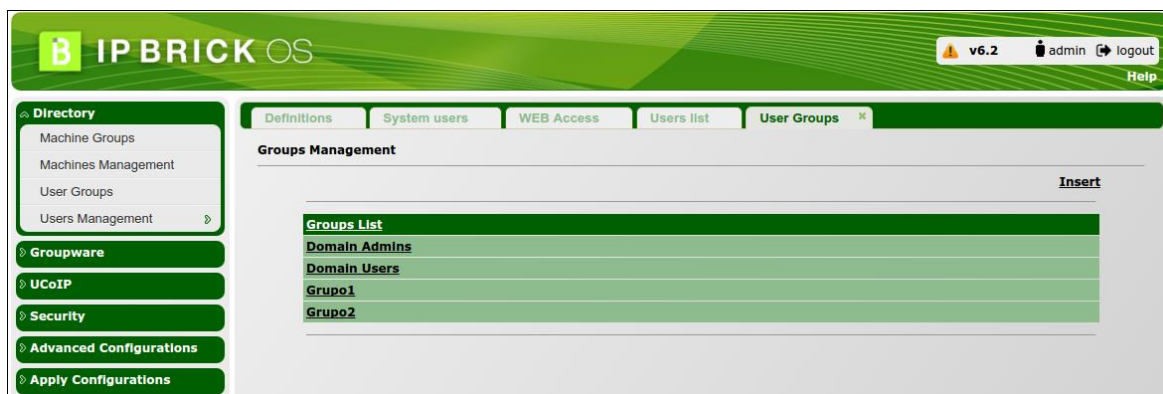


Figura 97 – IPBRICK OS: Gestão de grupos.

ÁREAS DE TRABALHO

Na IPBRICK OS existem as seguintes identidades:

- Estações de trabalho;
- IPBRICK.I - Servidor de intranet;
- IPBRICK.C - Servidor de comunicações responsável especificamente pelas comunicações com o exterior.

Uma área de trabalho corresponde a uma partição física do disco designada por "/home1" ou "/home2." Sempre que um novo utilizador é criado, o sistema gere a respetiva

conta pessoal, que corresponde a uma estrutura de pastas de suporte à conta do utilizador. Essa pasta irá conter:

- Caixa de correios eletrónico;
- Ficheiros pessoais;
- Perfil para *Windows*.

As áreas de trabalho, além das contas pessoais dos utilizadores, podem conter pastas relativas a partilhas de grupo e partilhas administrativas. A partir do menu *IPBRICK.I - Áreas de trabalho* é possível obter toda a informação relativa às duas áreas de trabalho.

As áreas de trabalho de grupo podem-se definir como partilhas de rede que podem ser acedidas com permissões para tal. Para criar uma área de grupo é preciso ir a *IPBRICK.I* → *Áreas de trabalho* → *Grupo* e clicar em *Inserir*. Uma vez inserida, é possível verificar as características da área de trabalho criada.

The screenshot shows the IPBRICK OS management interface. The top navigation bar includes the IPBRICK OS logo, version v6.2, user admin, and a logout button. The left sidebar contains a 'Directory' menu with sub-items: Groupware (E-Mail Server, Groupware Server, Domain Server, File Server, Print Server), UCoIP, Security, Advanced Configurations, and Apply Configurations. The main content area is titled 'Work Areas' and has tabs for Definitions, Machine Groups, User Groups, and Work Areas. It displays two work areas:

Work Area 1

Total Space	Free Space	Used Space	Percent Capacity
23.53 GB	22.16 GB	1.38 GB	5.85 %

Below this are two sub-tables:

Individual work areas		Group work areas	
User	Used Space	Group work area	Used Space
administrator	show		
user1	show		
user2	show		
user3	show		
user4	show		
user5	show		

Work Area 2

Total Space	Free Space	Used Space	Percent Capacity
11.31 GB	10.59 GB	745.25 MB	6.43 %

Below this are two sub-tables:

Individual work areas		Group work areas	
User	Used Space	Group work area	Used Space

Figura 98 – *IPBRICK OS*: Áreas de trabalho.

Anexo C. Guiões de aprendizagem da *IPBRICK OS*: Servidor de comunicações

Este guião aborda as opções do sistema *IPBRICK* e os serviços atribuídos a este tipo de servidor, dos quais se fará uma abordagem a: *Proxy, Firewall e VPN PPTP*.

INFORMAÇÕES SOBRE O SISTEMA DA *IPBRICK*

Após a instalação da *IPBRICK*, existe uma série de informações gerais relativas ao sistema, que podem ser consultadas através da sua interface *web*.

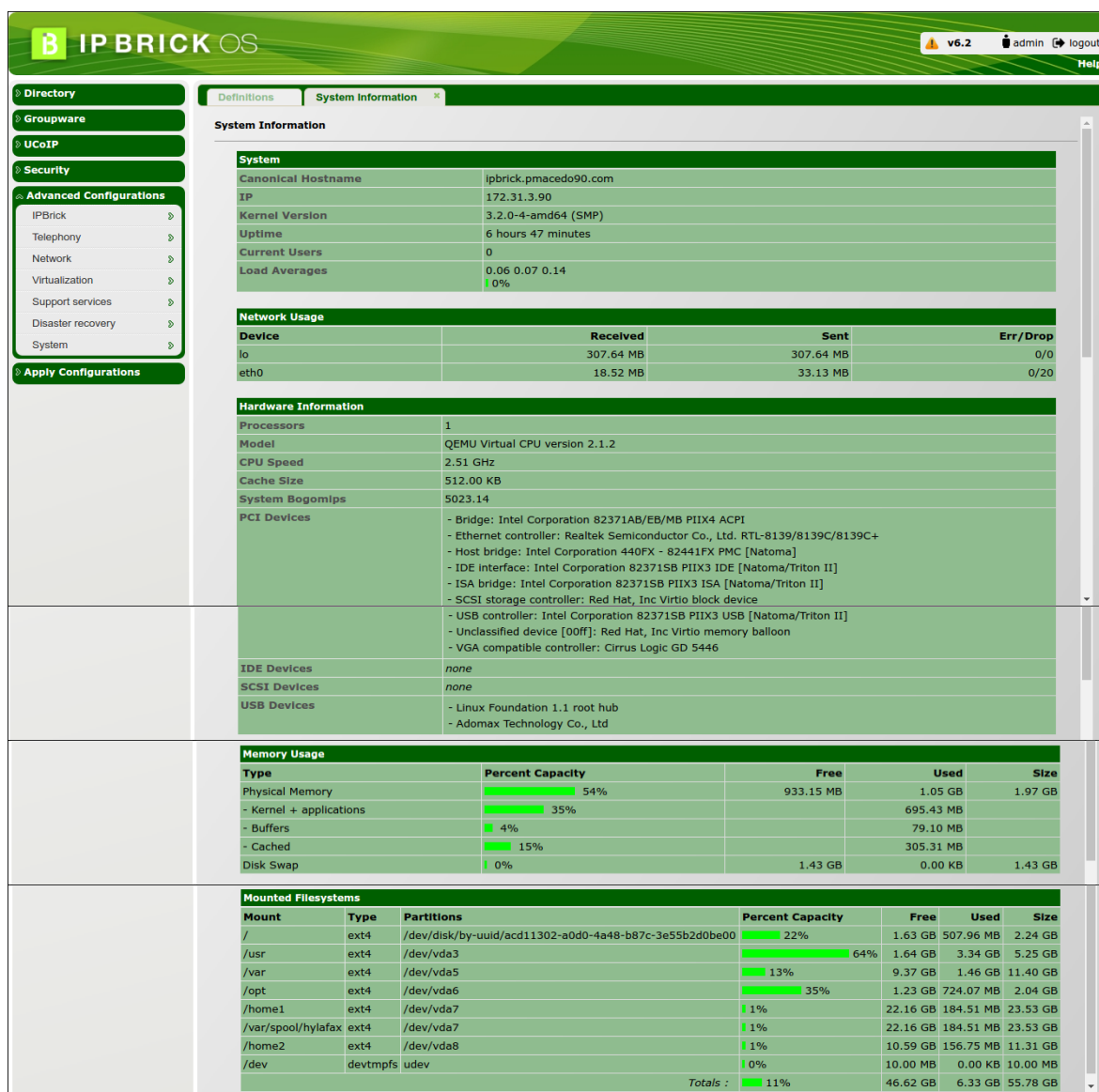


Figura 99 – *IPBRICK OS*: Informações gerais do sistema.

ENDEREÇAMENTO IP

Sendo o protocolo IP o grande responsável pelo encaminhamento de pacotes entre máquinas ligadas à internet, é necessário que cada uma delas possua um identificador único. Esse identificador é conhecido como endereço IP, sendo definido como uma sequência de 32 bits de zeros e uns.

Existe desde logo uma divisão no tipo de endereços IP:

- Endereços públicos, que são únicos e identificam inequivocamente uma máquina ligada à Internet;
- Endereços privados que são usados para identificar máquinas pertencentes a uma LAN, não estando ligadas diretamente à *internet*.

A *IPBRICK* está por defeito configurada para funcionar com duas interfaces de rede uma vez que tanto pode ter funções de servidor de *Intranet* como de comunicações, possuindo duas interfaces pré-configuradas:

- Interface interna: configurada para operar na rede 192.168.69.0/24, responsável pela conectividade com a LAN da empresa;
- Interface externa: a operar na rede 10.0.0.252/30, sendo uma ligação ao exterior ponto a ponto entre a *IPBRICK* e um equipamento com ligação.

SERVIDOR DE COMUNICAÇÕES

A *IPBRICK.C* é um servidor responsável pelas comunicações entre o exterior, tendo ligação à rede interna através da interface de rede *eth0* e à rede exterior através da interface *eth1*.

A *IPBRICK.C* disponibiliza os seguintes serviços de rede: *Firewall*; *Proxy*; *VPN*; *Servidor Web*; *Mail Relay*; *Webmail*; *IM*; *VoIP*.

FIREWALL

É possível negar ou permitir o acesso através da Internet a determinados serviços de rede a correr no servidor. Na opção "Serviços Bloqueados", é possível controlar o acesso a alguns serviços a correr internamente. O *software* para implementar a *firewall* no servidor *IPBRICK* é chamado de *iptables*.



Figura 100 – IPBRICK OS: Firewall da IPBRICK.

PROXY

O serviço de proxy é habitualmente utilizado para facilitar a gestão da rede no que diz respeito a acessos *web* por parte dos utilizadores de uma LAN.

Possibilita manter em cache os ficheiros relativos aos sites acedidos, havendo deste modo uma melhor gestão da largura de banda, como também permite personalizar quem pode aceder à *web*, a que horas e que tipo de páginas podem ser visitadas.

VPN - PPTP

As redes VPN têm como função permitir o acesso remoto a partir do exterior (redes não internas) a recursos de rede de uma determinada LAN. No caso de uma VPN do tipo PPTP, irão ser definidos quais os utilizadores com acesso a ligações VPN, assim como a gama de endereços que irá ser usada pelos clientes.

Anexo D. Guiões de aprendizagem da *IPBRICK OS*: Serviço de Suporte LDAP – Modo Master/ Slave/ Cliente

Este guião vai incidir sobre o processo de autenticação no servidor *IPBRICK*. Serão abordados os diferentes modos de autenticação possíveis, assim como a configuração de cada um deles.

MODOS DE AUTENTICAÇÃO

A partir do momento em que um utilizador é criado na *IPBRICK*, existirá um registo na base de dados do servidor de autenticação - *LDAP*. Este é definido como um serviço de diretório onde é guardada a informação relativo aos recursos informáticos aos recursos informáticos da empresa e dos seus utilizadores. Sempre que um utilizador se pretende autenticar num determinado serviço com o seu *username* e *password*, é feita uma consulta à base de dados *LDAP* da *IPBRICK* para efetivamente validar ou não o acesso.

A *IPBRICK* permite vários modos de autenticação, estando por defeito configurada para todos os utilizadores se autenticarem na própria *IPBRICK*. Neste guião serão apresentados apenas o modo *Master*, *Slave* e Cliente:

- **Master *IPBRICK*:** Modo pré-definido. Todos os serviços no servidor utilizar o próprio servidor *LDAP*.
- ***Slave IPBRICK*:** O servidor *LDAP* será uma réplica sincronizada do servidor *IPBRICK* Master indicado, sendo este modo usado num cenário com vários servidores.

Os *Slaves* são úteis nos seguintes cenários:

- Redes com elevado número de utilizadores onde existem muitas autenticações;
- Redes geograficamente distribuídas.
- **Cliente *IPBRICK*:** os serviços autenticam remotamente no servidor *LDAP IPBRICK*. Neste caso não há qualquer cópia da base de dados local, sendo necessário especificar qual o servidor *IPBRICK Master/Slave*. Normalmente este modo de autenticação é usado numa *IPBRICK.C* no âmbito dos serviços *VPN PPTP* e *Proxy*.

Os modos de autenticação podem ser configurados pela interface *web* da *IPBRICK* (ver Figura 101):

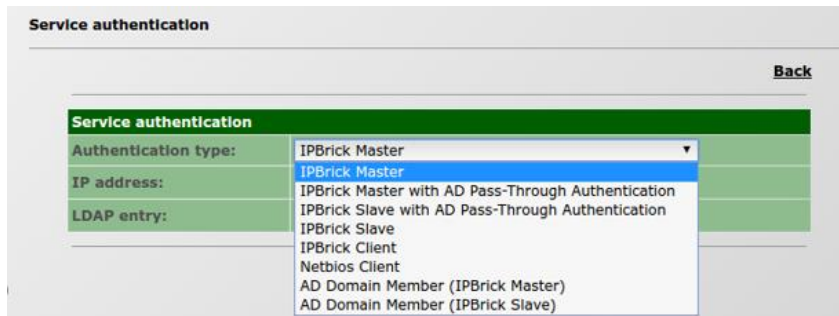


Figura 101 – *IPBRICK OS*: Modos de autenticação.

O procedimento básico para alterar o modo de autenticação *LDAP* é o seguinte:

- Configurar conectividade IP entre Master e *Slave*/Cliente;
- Colocar os *Slaves* no mesmo domínio *DNS* que o Master;
- Mudar o modo de autenticação do *Slave*/Cliente.

Na Figura 102 é visível uma rede com integração dos servidores *Master/Slave/Cliente*.

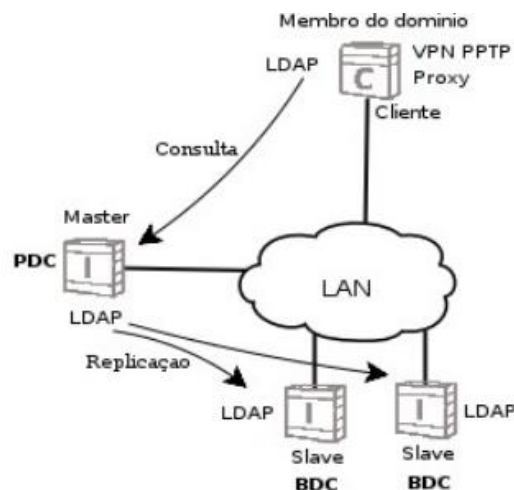


Figura 102 - Rede com *IPBRICK* Master, *Slave* e Cliente.

***FILESYSTEM* DISTRIBUÍDO**

As contas dos utilizadores podem ser fisicamente distribuídas pelos servidores *Master/Slave*. Entretanto o sistema de informação centralizado - *LDAP* - tem a informação sobre a localização física de cada conta.

Um serviço de *NFS* (*Network File System*) disponibiliza as contas dos utilizadores via rede.

O serviço *Automount* combina a informação o do *LDAP* com o *NFS* e disponibiliza automaticamente as contas dos utilizadores virtualmente em qualquer servidor *Master/Slave*.

Anexo E. Guiões de aprendizagem da *IPBRICK OS: UcoIP*

Este último guião incide nas tecnologias de comunicação utilizadas na *IPBRICK*, as suas vantagens e como configurar estes tipos de comunicação.

VoIP

A tecnologia *VoIP* permite o estabelecimento de chamadas telefónicas através de uma rede IP tornando possível por exemplo efetuar chamadas telefónicas através da Internet. As principais vantagens da utilização de *VoIP* passam pela: Redução de despesas devido ao fato das tarifas não seguirem o mesmo modelo da telefonia convencional; melhor qualidade de serviço uma vez que a comutação por pacotes ao contrário da comutação de circuitos, faz um melhor aproveitamento dos recursos de rede existentes.

REGISTO DE TELEFONES

Os telefones IP podem ser registados na *IPBRICK* através do menu: *IPBRICK.I* → *Gestão de máquinas*.



The screenshot shows a web interface for managing IP phones. The breadcrumb trail is 'VoIP » Phone management » Registered Phones » phone1'. There are three buttons: 'Back', 'Modify', and 'Delete'. Below this is a table titled 'Phone Definitions' with the following data:

Phone Definitions	
Phone:	phone1
Phone Address:	phone1@pmacedo90.com
Caller ID:	External: Not masked Internal: Not masked
Phone Location:	Local
Auto provisioning:	None
Description:	

Figura 103 – *IPBRICK OS*: Registo de telefones.

ROTAS

Para ser possível à *IPBRICK* o encaminhamento de chamadas entre as várias interfaces de rede, torna-se necessário a definição de rotas específicas de acordo com a numeração telefónica. É possível definir as rotas em: "*IPBRICK.C - VoIP* → *Gestão de Rotas*".

Local Routes	
Type	Destination
PBX -> PSTN	Public Network
PSTN -> PBX	Phones

Outbound routes		
Type	Name	Destination

SIP servers list for registering	
Name	Server

Figura 104 – *IPBRICK OS*: Gestão de rotas.

Existem dois tipos de rotas: as rotas locais e as rotas de saída para servidores *SIP*. Existe também a possibilidade de registrar a *IPBRICK* em servidores *SIP*.

- Rotas Locais: permitem configurar uma ligação entre as interfaces definidos (*LAN, PSTN, PBX, Internet...*).
- Rotas de saída para servidores *SIP*: Este tipo de rotas permite configurar através da atribuição de prefixos quais as chamadas que serão direcionadas para um servidor *SIP* que ficará responsável pelo seu encaminhamento para o destino final.

UCoIP

A *IPBRICK* afirma-se hoje como um dos primeiros sistemas de comunicação que implementam o conceito de *UcoIP* (*Unified Communications over IP* - Comunicações Unificadas sobre *IP*). Deste modo, a *IPBRICK* pode implementar comunicações baseadas unicamente no endereço do utilizador, para todos os tipos de comunicações empresariais: *Voz; email; Instant Messaging; Web*. Para tipos de comunicação alcançar este objetivo a *IPBRICK* utiliza unicamente serviços de comunicação da *Internet: SIP, SMTP/IMAP, XMPP* e *HTTP*, assentes nos serviços de suporte *DNS* e *LDAP* devidamente integrados.