



Instituto Politécnico de Coimbra  
Instituto Superior de Engenharia de Coimbra  
Departamento de Engenharia Informática e Sistemas

---

# **KamaeLei *Web Mobile***

Mestrado em informática e Sistemas  
Desenvolvimento de Software

**Autor**

**Carlos Nogueira**

**Orientadores**

**Doutora Teresa Rocha**

ISEC

**Doutor Ricardo Teixeira**

**Engenheiro Arlindo Martins**

Kamae RT

**Coimbra, Setembro 2015**



## **Agradecimentos**

Gostaria de agradecer a todos os professores no ISEC que me auxiliaram durante o período de estágio, em especial à Professora Doutora Teresa Rocha, minha orientadora, pelo interesse demonstrado e pelo apoio dado ao longo de todo este percurso.

Queria ainda expressar o meu agradecimento a todos os membros da Kamae RT, em particular, ao Doutor Ricardo Teixeira, Engenheiro Arlindo Martins, Engenheiro João Mendes e Engenheira Daniela Cortesão, pela sua disponibilidade e vontade de ajudar, bem como a todos os meus colegas de estágio por me proporcionarem um bom ambiente de trabalho.

Por fim, gostaria endereçar um agradecimento especial à minha família e amigos mais próximos, pelo apoio e motivação que sempre me deram.



## Resumo

O presente estágio foi realizado na Kamae RT, uma empresa reconhecida internacionalmente pelos seus sistemas de gestão comercial KamaeGest e de informação jurídico KamaeLei, os quais são usados por mais de 500 clientes. O projeto levado a cabo consiste numa aplicação *web* desenvolvida em Asp.Net MVC com foco em *browsers* de dispositivos móveis, o que permite que a aplicação tenha um aspeto gráfico mais semelhante às soluções nativas da plataforma onde é executada. Esta aplicação, designada por KamaeLei *web mobile*, visa servir como um complemento e não como um substituto completo da aplicação KamaeLei *Web desktop* que é a versão *web* do KamaeLei, um sistema vocacionado para auxiliar advogados na realização das tarefas diárias e a minimizar os seus problemas. Tendo sido projetada apenas para *browsers* móveis, algumas das funcionalidades da aplicação desenvolvida tiveram de ser simplificadas de forma a evitar que a interface ficasse desorganizada e os conteúdos ilegíveis, proporcionando assim uma boa experiência ao utilizador. Esta versão da aplicação conta com as principais funcionalidades da aplicação *web desktop*, tais como “gestão de clientes”, “gestão de contactos”, “gestão de *dossiers*”, “gestão de processos”, “gestão de prazos”, “gestão de despesas”, “gestão de notas”, “gestão de *time-sheet*” e “configuração de opções”. Na verdade, esta aplicação permite aos utilizadores ter acesso às principais funcionalidades do KamaeLei, estando sempre atualizada, ao contrário do que acontece com as soluções nativas para as plataformas móveis, já que o código fonte é distinto em todas elas.

**Palavras-chave:** Sistema de informação jurídico, Dispositivos móveis, Internet móvel



## Abstract

The carried out internship was done in Kamae RT, an internationally renowned company for its commercial management system and its legal information system, known as KamaeGest and KamaeLei, which gather a total of over 500 clients. This project consists in a *web* application developed in Asp.Net MVC for *mobile* browsers only, which allows the application to have a look and feel more on pair with the native solution of the platform which is running it. This application is to serve as a complement rather than a complete substitute for the *web* application that is KamaeLei *Web*. The last is the *web* solution of KamaeLei, which is a legal information system suited for lawyers, to help them make their daily tasks easier and less troublesome. This application is to work for *mobile* browsers only, and so, to prevent the interface from being cluttered, some of the functionalities that exist in the *web* version were simplified enough so that a pleasant user experience could be achieved. This version of the application counts with the main functionalities of the main application like “Clients management”, “Contacts management”, “Matters management”, “Cases management”, “Deadlines management”, “Expenses management”, “Notes management”, “Time-sheet management” and “Options configuration”. This application will allow users to access the main functionalities of KamaeLei and always be up to date, as opposed to the native solutions for the *mobile* platforms, since the source-code is distinct in every single one of them.

**Keywords:** Legal information system, Mobile devices, Mobile web





## Índice

Agradecimentos.....	iii
Resumo.....	v
Abstract.....	vii
Acrónimos.....	xvii
1 Introdução.....	1
1.1 Enquadramento.....	2
1.2 Objetivos do estágio.....	2
1.3 Estrutura do documento.....	3
2 Contextualização.....	5
2.1 Evolução das tecnologias que possibilitaram o crescimento da mobile web.....	5
2.2 Aplicações web mobile e aplicações nativas.....	7
2.3 Aplicações KamaeLei e aplicações concorrentes.....	8
3 Ferramentas e tecnologias.....	9
3.1 Principais ferramentas de desenvolvimento.....	9
3.2 Principais linguagens/tecnologias.....	10
4 Desenvolvimento do projeto.....	17
4.1 Metodologia de desenvolvimento.....	19
4.2 Planeamento do projeto.....	20
4.3 Principais funcionalidades.....	21
4.4 Requisitos não funcionais.....	25
4.5 Programa de trabalhos.....	26
4.6 Atividades no decorrer do estágio.....	26
4.7 Mockups.....	30
4.8 Arquitetura.....	32
4.8.1 Routing.....	33
4.8.2 Action filters.....	35
4.8.3 Extensões de modelo.....	37
4.8.4 Razor.....	38
4.8.5 Layout.....	39
4.8.6 jQuery Mobile.....	40
4.8.7 KamaeLei Web.....	41
4.8.8 Partial Views.....	41
4.8.9 Comunicação AJAX entre cliente e servidor.....	42
4.8.10 Outros aspetos.....	46
4.9 Estrutura do projeto.....	48
4.9.1 Área de <i>Dossiers</i> .....	49
4.9.2 Área de Clientes.....	55
4.9.3 Área de Opções.....	56
4.9.4 Área de Contactos.....	64
4.9.5 Área de Login.....	65
4.9.6 Área de Time-Sheet.....	65

5	Testes .....	67
5.1	Dipositivos, sistemas e browsers testados.....	68
5.2	Testes Unitários .....	72
6	Conclusões .....	73
	Referências Bibliográficas.....	75
	Anexos.....	79
	Anexo A.....	79
	Diagramas de sequência.....	79
	Anexo B.....	84
	Wizard de <i>Dossiers</i> .....	84
	Wizard de processos .....	91
	Wizard de Clientes .....	94
	Anexo C.....	102
	Listagem de bugs reportados.....	102

## Índice de figuras

Figura 1 - Timepicki	12
Figura 2 - Real person capcha	12
Figura 3 - Asp.Net MVC	15
Figura 4 - Exemplo de pesquisa de contexto	17
Figura 5 - Funcionamento do processo de srcum	19
Figura 6 - Definição de requisitos	20
Figura 7 - Login mockup	31
Figura 8 - Menu principal mockup	31
Figura 9 - Menu secções mockup	31
Figura 10 - Menu logout mockup	31
Figura 11 - Secção dossiers mockup	31
Figura 12 - Listagem s/inserção mockup	31
Figura 13 - Listagem c/inserção mockup	31
Figura 14 - Secção de opções mockup	31
Figura 15 - Listagem dupla mockup	31
Figura 16 - Arquitetura da aplicação	32
Figura 17 - Routing 1	33
Figura 18 - Routing 2	34
Figura 19 - Action filter AllowCrossSiteJson	35
Figura 20 - ActionFilter SessionExpire	36
Figura 21 - Exemplo de extensão de modelo	37
Figura 22 - Mecanismo Razor	38
Figura 23 - Exemplo da aplicação do mecanismo de layout	39
Figura 24 - Pesquisa de contexto 2	42
Figura 25 - Exemplo de listagem	43
Figura 26 - Implementação do método doneTyping	44
Figura 27 - Classe JsonHelper	46
Figura 28 - Classe LogHelper	47
Figura 29 - Página inicial	48
Figura 30 - Área de dossiers	49
Figura 31 - Formulário de notas sem contexto	51
Figura 32 - Formulário de notas com contexto	51
Figura 33 - Formulário de despesa (parte 1)	52
Figura 34 - Formulário de despesa (parte 2)	52
Figura 35 - Formulário de despesa (parte 3)	52
Figura 36 - Formulário de prazos (parte 1)	53
Figura 37 - Formulário de prazos (parte 2)	53
Figura 38 - Área de opções	56
Figura 39 - Formulário de opções de cliente	57
Figura 40 - Formulário de opções de dossier (parte 1)	58
Figura 41 - Formulário de opções de dossier (parte 2)	58
Figura 42 - Formulário de opções de dossier (parte 3)	58

Figura 43 - Formulário de opções de processo	60
Figura 44 - Formulário de opções de lançamentos	61
Figura 45 - Formulário de opções de escritório (parte 1)	63
Figura 46 - Formulário de opções de escritório (parte 2)	63
Figura 47 - Formulário de contacto rápido	64
Figura 48 - Formulário de contacto completo (parte 1)	64
Figura 49 - Formulário de contacto completo (parte 2)	64
Figura 50 - Formulário time-sheet (parte 1)	65
Figura 51 - Formulário time-sheet (parte 2)	65
Figura 52 - Formulário time-sheet (parte 3)	65
Figura 53 - Exemplo de teste unitário	72
Figura 54 - Pesquisa de contexto por cliente	79
Figura 55 - Pesquisa de contexto por dossier	80
Figura 56 - Pesquisa de contexto por processo	81
Figura 57 - Listagem de registos	82
Figura 58 - Listagem duplas	83
Figura 59 - Área de seleção de clientes – Wizard de dossiers	84
Figura 60 - Área de propriedades gerais (parte 1) – Wizard de dossiers	84
Figura 61 - Área de propriedades gerais (parte 2) – Wizard de dossiers	85
Figura 62 - Área de seleção de equipa – Wizard de dossiers	85
Figura 63 - Área financeira (parte 1) – Wizard de dossiers	86
Figura 64 - Área financeira (parte 2) – Wizard de dossiers	86
Figura 65 - Área de custo da equipa – Wizard de dossiers	87
Figura 66 - Área de alertas – Wizard de dossiers	87
Figura 67 - Área de dados adicionais – Wizard de dossiers	88
Figura 68 - Área de representados da parte oposta – Wizard de dossiers	88
Figura 69 - Área de representantes da parte oposta – Wizard de dossiers	89
Figura 70 - Área de confirmação de dados (parte 1) – Wizard de dossiers	89
Figura 71 - Área de confirmação de dados (parte 2) – Wizard de dossiers	90
Figura 72 - Área de informação geral (parte 1) – Wizard de processos	91
Figura 73 - Área de informação geral (parte 2) – Wizard de processos	91
Figura 74 - Área de informação judicial (parte 1) – Wizard de processos	92
Figura 75 - Área de informação judicial (parte 2) – Wizard de processos	92
Figura 76 - Área de confirmação de dados – Wizard de processos	93
Figura 77 - Área de listagem de clientes – Wizard de clientes	94
Figura 78 - Área de listagem de clientes – Wizard de clientes	94
Figura 79 - Área de informações do cliente (parte 1) – Wizard de clientes	95
Figura 80 - Área de informações do cliente (parte 2) – Wizard de clientes	95
Figura 81 - Área de contactos (parte 1) – Wizard de clientes	96
Figura 82 - Área de contactos (parte 2) – Wizard de clientes	96
Figura 83 - Área financeira – Wizard de clientes	97
Figura 84 - Área de dados adicionais singular (parte 1) – Wizard de clientes	97
Figura 85 - Área de dados adicionais singular (parte 2) – Wizard de clientes	98

Figura 86 - Área de marketing – Wizard de clientes	98
Figura 87 - Área de confirmação (parte 1) – Wizard de clientes	99
Figura 88 - Área de confirmação (parte 2) – Wizard de clientes	99
Figura 89 - Área de contactos adicionais – Wizard de clientes	100
Figura 90 - Área de dados adicionais coletivos (parte 1) – Wizard de clientes	100
Figura 91 - Área de dados adicionais coletivos (parte 2) – Wizard de clientes	101
Figura 92 - Área de criação rápida – Wizard de clientes	101



## Índice de tabelas

Tabela 1 - Plugins jQuery	12
Tabela 2 - Browser support	14
Tabela 3 - Gestão de time-sheet	21
Tabela 4 - Gestão de clientes	21
Tabela 5 - Gestão de contactos	22
Tabela 6 - Gestão de dossiers	22
Tabela 7 - Gestão de notas	23
Tabela 8 - Gestão de despesas	23
Tabela 9 - Gestão de prazos	23
Tabela 10 - Gestão de processos	24
Tabela 11 - Funcionalidades gerais	24
Tabela 12 - Escalonamento de trabalhos	26
Tabela 13 - Mockups	31
Tabela 14 - Sistemas e Browsers testados	68
Tabela 15 - Nível de performance	69
Tabela 16 - Identificação dos browsers	69
Tabela 17 - Renderização dos elementos	69
Tabela 18 - Fluidez nas animações	70
Tabela 19 - Fluidez na utilização de componentes jQuery	70
Tabela 20 - Listagem de bugs reportados	102





## **Acrónimos**

ISEC – Instituto Superior de Engenharia de Coimbra

JSON – JavaScript Object Notation

AJAX – Asynchronous JavaScript and XML

HTML – Hypertext Markup Language

DOM – Document Object Model

URL – Uniform Resource Locator

W3C – World Wide Web Consortium

CSS – Cascading Style Sheets

SVG – Scalable Vector Graphics

TFS – Team Foundation Server

LINQ – Language Integrated Query

SP – Stored Procedure

DLL – Dynamic Link Library



## 1 Introdução

Os dispositivos móveis fazem cada vez mais parte do nosso quotidiano, estando presentes em grande número e nas várias faixas etárias. Um estudo realizado pela *Digital Portal* mostra que nos EU5, as cinco maiores economias da União Europeia (Reino Unido, Alemanha, França, Itália e Espanha), a utilização de dispositivos móveis alcançou 57% [1] e a tendência é para que este valor continue a crescer a cada ano. Estes são dados surpreendentes que mostram como o interesse dos utilizadores está a mudar para uma maior utilização de dispositivos móveis. Segundo um estudo feito pela *StatCounter*, a utilização de dispositivos móveis teve um aumento de 11,4%, de 17,1% para 28,5%, desde Agosto de 2013 até Agosto de 2014 [2], em todo o mundo. Este incremento é, em grande parte, causado pelo decréscimo acentuado do custo de produção de dispositivos móveis que tem vindo a acontecer nos últimos anos principalmente devido a dois fatores: aparecimento de fabricantes como a Xiaomi ou a Huawei com os seus dispositivos de baixo custo; aparecimento de iniciativas com o *Android One* que visa facilitar o acesso a dispositivos móveis em países menos favorecidos, tais como a Índia [3], para que qualquer pessoa, independentemente do sítio onde nasceu ou das suas posses monetárias, tenha igual acesso à informação.

O *Mobile first* é um conceito que tem vindo cada vez mais a ser adotado e que promove que o *layout* de uma aplicação ou página *web* seja em primeiro lugar pensado para dispositivos móveis. A criação de *web sites* ou aplicações *web* com foco em dispositivos móveis é uma tendência em rápido crescimento e que está para ficar, até porque, atualmente, a utilização de internet em dispositivos móveis já excede a utilização em computadores. Desta forma, tirar partido de tudo o que a *mobile web* tem para oferecer, é de importância crítica para qualquer organização que tenha aplicações *web* como parte do seu negócio.

Uma destas organizações é a Kamae RT que está a trabalhar ativamente na criação de uma versão *web* do seu sistema de informação jurídico, KamaeLei, com base na versão *desktop* já existente e instalada num elevado número de clientes.

## 1.1 Enquadramento

A kamae RT foi fundada em 1997 e desde o início tem mantido como foco principal, a qualidade e a rapidez do seu serviço, privilegiando uma excelente relação de proximidade com os clientes. A organização destaca-se pelas suas soluções KamaeGest, e KamaeLei. A primeira é uma aplicação de gestão comercial que contém módulos como gestão de clientes, *stocks*, faturação, fornecedores, entre outros. A segunda solução, KamaeLei, que é de importância central para o presente estágio, consiste num sistema de informação jurídico que permite auxiliar advogados/sociedades a realizarem as suas tarefas mais facilmente. Esta encontra-se dividida em 4 versões, a versão Base, que é focada para advogados, juristas e solicitadores que trabalhem em prática isolada, a versão Plus, que é focada em sociedades e equipas de pequena dimensão que necessitam de trabalhar em rede e de mobilidade da informação, a versão VIP, que é focada para sociedades, equipas e departamentos de pequena e média dimensão, e a versão LawFirm, que é focada para equipas, sociedade e empresas que desejam ter informação de forma móvel e um controlo efetivo da organização e agilização de procedimentos internos.

Partindo da sua versão original para *desktop*, a empresa tem vindo a apostar no desenvolvimento de uma versão *web* dessa aplicação. Contudo, face ao enorme crescimento verificado no uso de dispositivos móveis, a Kamae RT decidiu que seria igualmente importante criar uma versão específica para este tipo de dispositivos tendo para tal proposto o presente trabalho de estágio.

## 1.2 Objetivos do estágio

Com efeito, o objetivo deste estágio é desenvolver a aplicação KamaeLei Web Mobile, que consiste numa nova versão da solução base KamaeLei focada apenas em *browsers* para *smartphones* e *tablets*. Esta nova aplicação deverá complementar a solução global, permitindo ao utilizador aceder às principais funcionalidades da versão *web* ou *desktop* através do *browser* do seu dispositivo móvel.

Deste modo, deverão ser implementados os módulos de login, de gestão de clientes, gestão de contactos, gestão de *dossiers*, gestão de processos, gestão de prazos, gestão de despesas, gestão de notas, gestão de *time-sheet* e configuração de opções.

## 1.3 Estrutura do documento

O presente relatório encontra-se dividido nos seguintes capítulos:

1. Introdução
2. Contextualização
3. Ferramentas e tecnologias
4. Desenvolvimento do projeto
5. Testes
6. Conclusões

No capítulo da **introdução**, é feito o enquadramento do estágio explicando a motivação que esteve na base do seu lançamento e são apresentados os objetivos a serem atingidos.

No capítulo da **contextualização**, é feita uma análise ao crescimento do segmento móvel no mercado atual bem como às ferramentas que tornam esse desenvolvimento mais rápido e simples. É também feita uma breve análise à aplicação a desenvolver comparando-a, em termos de funcionalidades, com as restantes soluções integradas na KamaeLei e com as aplicações semelhantes que existem no mercado.

No capítulo das **ferramentas e tecnologias**, é feita uma descrição das principais tecnologias utilizadas ao longo do desenvolvimento do projeto.

No capítulo de **desenvolvimento do projeto**, é apresentada toda a parte de desenvolvimento, nomeadamente a metodologia de desenvolvimento, o planeamento do projeto, a análise de requisitos, a utilização de *mockups*, a arquitetura implementada e a estrutura da aplicação. São igualmente descritas as várias áreas da aplicação e a forma como foram organizadas.

No capítulo de **testes**, é feita uma descrição da forma como foram executados os testes no projeto.

No capítulo das **conclusões**, é feita uma análise à forma como decorreu o estágio e são dadas sugestões para a continuação do desenvolvimento do projeto.



## 2 Contextualização

Nos últimos anos o desenvolvimento para a *web* sofreu uma mudança radical. Inicialmente a única preocupação ao desenvolver uma aplicação *web* era se esta corria moderadamente bem em todos os *browsers* de *desktop*. No entanto, com o rápido crescimento do segmento móvel, aplicações *web* e *web sites* em geral, tiveram de se reinventar e de se adaptar a este novo conceito. Esta revolução começou com o lançamento do iPhone 3 pela Apple a 9 de Junho de 2009 [4]. De facto, o iPhone foi o primeiro dispositivo a ser adotado pelas massas e aquele que impulsionou o crescimento deste segmento. Este dispositivo contava com características críticas para que a “*mobile web*” pudesse crescer e chegar ao ponto em que está atualmente: com acesso rápido à internet, um sistema operativo adaptado a dispositivos de dimensões reduzidas e um browser capaz de apresentar os conteúdos de uma página *web* de uma forma agradável para o utilizador [5]. Embora este constituísse uma grande melhoria face ao que existia na altura, estava longe de ser uma solução perfeita pois, por vezes, a navegação nas várias páginas era lenta e os *sites* muito pesados, o que fazia com que a experiência do utilizador nestes dispositivos fosse grandemente prejudicada, principalmente quando se recorria a dados móveis para efetuar o acesso. Desde então foram pensadas várias soluções para ultrapassar este problema mas, no entanto, depararam-se com um problema ainda maior que era o facto das tecnologias da altura não permitirem os avanços que se impunham. Foi neste contexto que surgiram as novas tecnologias como o HTML5 e o CSS3.

### 2.1 Evolução das tecnologias que possibilitaram o crescimento da *mobile web*

O HTML5, introduzido ao público em 2008, veio permitir uma melhor estruturação do conteúdo de uma página *web*, um código mais limpo e fácil de entender e uma maior consistência [6]. No entanto, apenas em 2014 foi feita a recomendação final por parte da W3C (World Wide Web Consortium) [7]. Por sua vez, o CSS3 teve o seu desenvolvimento iniciado em Junho de 1999 e desde essa data tem vindo a ser constantemente atualizado. O que diferencia o CSS3 da sua versão anterior é o facto de estar dividido em módulos, permitindo a adição de novas funcionalidades e características. Atualmente existem mais de 50 módulos publicados para CSS3 sendo que alguns já receberam uma recomendação formal [8]. Um destes módulos, o das Media Queries, foi o mais importante para o crescimento da *mobile web*. Tornou-se num *standard* pela WC3 em Junho de 2012, é responsável por aquilo que atualmente se designa por Responsive Web Design [9]. Este não é nada mais do que a transformação do conteúdo de um *site* ou aplicação *web* para que se adapte a dispositivos de vários tamanhos, mesmo que tal implique mudanças de *layout* e não apenas redimensionamento de elementos de páginas. O objetivo é que a visualização dos conteúdos esteja sempre otimizada independentemente do dispositivo a usar: computador, *tablet* ou *smartphone*. Este novo conceito veio abrir oportunidades para novas tecnologias emergirem e se tornarem numa força predominante em termos de desenvolvimento *web*. São disso exemplo tecnologias como jQuery Mobile ou Bootstrap, que simplificam acentuadamente o desenvolvimento para dispositivos móveis, ou até mesmo o Google

Polymer que, embora ainda muito recente (versão 1.0 foi lançada a junho de 2015 durante o Google IO), já está a revolucionar o desenvolvimento para a *web* com o uso de “Web Components” que o tornam mais simples e flexível. De facto, os “Web Components” têm como finalidade tornar o código de uma aplicação mais modular, permitindo que este seja mais facilmente estruturável e reutilizável. Estes encontram-se divididos em quatro partes distintas: “Custom Elements”, que permitem a criação de novos elementos sob a forma de tags HTML com as quais é possível associar comportamento desejado, utilização de “HTML Templates” para conter o código de um elemento; utilização de um método para adicionar uma subárvore de “Shadow Dom” (capacidade de um browser de incluir uma subárvore de elementos DOM na renderização de um documento [10]) a um elemento e, por fim, a utilização de “HTML Imports” que permitem que os vários componentes criados sejam importados e inseridos nas várias páginas. [11]. Para além da biblioteca Google Polymer referida anteriormente, existem outras para o mesmo fim como a X-Tag da Mozilla e a Bosonic.

Atualmente, por norma são utilizadas tecnologias que facilitam o desenvolvimento de uma aplicação *web* como um todo, desde a aplicação servidor até à aplicação cliente, juntas num único projeto, o que permite uma comunicação mais simples e linear entre ambas. As tecnologias mais usadas que são exemplo do tipo de desenvolvimento descrito em cima, baseiam-se em grande parte no padrão MVC, tais como Ruby On Rails ou Microsoft Asp.Net MVC. Estas tecnologias funcionam separando os vários elementos de uma aplicação em três secções distintas: o modelo, onde é normalmente implementada a lógica da aplicação; as vistas, onde são apresentados os dados, normalmente recorrendo a tecnologias de “frontend” como HTML, CSS e JavaScript; e os controladores, que permitem a comunicação entre o modelo e as vistas. Este tipo de tecnologias oferece várias utilidades para um desenvolvimento mais rápido, tais como a utilização de “partial views” que possibilitam a reutilização de código HTML, CSS e Javascript, tendo um comportamento de certa forma semelhante a um “Web Component”.

A primeira *framework* baseada em MVC a alcançar grande sucesso foi Ruby on Rails, lançada em Dezembro de 2005 com o objetivo de facilitar o desenvolvimento para a *web* [12]. Embora não tenha sido a primeira *framework* deste tipo, foi a que conseguiu captar maior interesse por parte da comunidade, aumentando por consequência a comunidade Ruby e permitindo que a linguagem e a *framework* chegassem ao ponto em que se encontram atualmente. É uma *framework* amplamente difundida e é utilizada em várias aplicações conhecidas, tais como Scribd, Hulu, Github, Slideshare, Basecamp, entre outras [13].

O objetivo com o lançamento da linguagem Ruby foi o de criar uma alternativa às linguagens de programação existentes na altura, a qual fosse o mais natural possível. Como tal, a sintaxe da linguagem diferia consideravelmente das restantes, o que não agradou a toda a comunidade. Isto abriu oportunidades para o lançamento da *framework* Asp.Net MVC por parte da Microsoft. A grande diferença entre as duas *frameworks* é a linguagem de programação escolhida, sendo que em Ruby on Rails a linguagem é Ruby e em Asp.Net MVC a linguagem é C#.

A grande razão para a adoção destas *frameworks* é a facilidade de criação de uma aplicação *web* e a possibilidade de se ter rapidamente um protótipo funcional. Com efeito, por norma estas *frameworks* seguem uma abordagem baseada em convenções em vez de uma abordagem baseada em configurações, sendo que, caso estas convenções sejam seguidas, poucas ou nenhuma configurações têm de ser feitas para a aplicação funcionar. No entanto, à



medida que o tamanho e a complexidade de uma aplicação aumentam, torna-se muitas vezes necessário configurá-la pois o comportamento base pode não ser suficiente.

## 2.2 Aplicações web mobile e aplicações nativas

Como se pode constatar, existe uma grande margem de manobra quando se trata de desenvolvimento *web* devido ao elevado número de tecnologias e à facilidade de integração entre elas, o que não acontece quando se desenvolvem aplicações nativas. Na verdade, neste último caso, para cada nova plataforma tem que ser desenvolvida uma nova aplicação, algo que nem todas as organizações conseguem suportar devido aos recursos necessários em termos de criação e manutenção. Contudo, existem ferramentas que simplificam este processo, tais como Xamarin ou LiveCode, as quais permitem a criação de aplicações para as principais plataformas com poucas ou nenhuma alteração ao código fonte, dependendo da complexidade do projeto. A grande desvantagem destas ferramentas é o preço das licenças, o qual pode chegar aos 1899 dólares por ano, no caso do Xamarin [14], e aos 199 dólares por mês, no caso do LiveCode [15]. Em comparação com este tipo de tecnologias, o desenvolvimento *web* para *mobile* apresenta vantagens e desvantagens. A principal desvantagem é que o *Look And Feel* de uma aplicação *web*, mesmo que seja apenas para dispositivos móveis, não é igual ao de uma aplicação nativa, embora existam tecnologias que tornam estas diferenças menos notáveis. Por outro lado, as vantagens são que o preço é bastante mais acessível e, tratando-se de uma aplicação *web*, é baseada em tecnologias largamente conhecidas, sendo extremamente simples a reutilização de recursos de aplicações *web* já existentes. Concluindo, ambas as abordagens têm vantagens e desvantagens, cabendo a cada organização analisar qual a alternativa que vai mais ao encontro dos seus interesses.

Quando se trata de desenvolvimento para dispositivos móveis, quer seja através de aplicações nativas, quer seja através de aplicações *web*, os dois grandes pilares de desenvolvimento são a simplicidade e a usabilidade. Isto deve-se às dimensões reduzidas dos dispositivos onde estas plataformas assentam, as quais impedem que o conteúdo e a estrutura da aplicação seja igual à versão *desktop* ou à versão *web* para *desktop*.

Normalmente as aplicações móveis não são substitutas completas das versões *web* e *desktop* mas sim complementos, pois para obedecer aos dois princípios descritos anteriormente, o conteúdo tem que ser altamente simplificado, o que pode tornar algumas funcionalidades extremamente difíceis ou impossíveis de implementar. Existem inúmeros exemplos que seguem este mesmo padrão, tais como o Gmail, o qual tem todas as suas configurações na versão *web* para *desktop* mas, na versão *mobile*, todas as configurações de maior complexidade foram deixadas de parte e apenas as funcionalidades principais, aquelas que oferecem valor real ao utilizador e que são essenciais, passaram para a versão móvel.

O tópico de desenvolvimento *web* para dispositivos móveis gera normalmente muita discussão, principalmente sobre qual a direção a seguir. As opções existentes são, o desenvolvimento de uma única aplicação e o uso de Responsive Design para adaptá-la a dispositivos móveis de várias dimensões, ou o desenvolvimento de uma aplicação separada onde o foco são apenas *browsers* de dispositivos móveis. Ambas as abordagens oferecem

vantagens e desvantagens, sendo que atualmente no mercado existem soluções que implementam estas duas opções.

A abordagem escolhida para o projeto *KamaeLei Web Mobile* foi a de desenvolver uma aplicação *web* focada apenas nos dispositivos móveis, permitindo obter um *design* mais semelhante a uma aplicação móvel nativa do que a uma aplicação *web*, sendo que esta similaridade pode melhorar a usabilidade para o utilizador final.

### **2.3 Aplicações KamaeLei e aplicações concorrentes**

A aplicação desenvolvida faz parte de um grupo de aplicações, todas associadas com a *KamaeLei*, tais como o *IKL* para *iPad*, *IKL* para *iPhone*, *IKL* para *Android*, *KamaeLei Web* e *KamaeLei desktop*. Embora diferentes, todas estas aplicações são baseadas na *KamaeLei desktop* e têm, na sua maioria, funcionalidades semelhantes.

Atualmente existem no mercado aplicações que servem objetivos semelhantes aos da *KamaeLei*, tais como o *Jurigest* da *Junifor*, a *Forense da Gravidade Zero*, a *Jurisflow* da *Pontual Software*, a *SIG Advogados* da *WinSig*, o *Lawyer Gest* da *RGPSi*, a *ADV* da *APKOMP* e *LawRd* da *Muchbeta*.

Como se pode constatar, esta é uma área já saturada, com um vasto leque de soluções em que as funcionalidades oferecidas por cada uma são muito semelhantes às das suas concorrentes. De facto, uma vez que as diversas aplicações visam resolver os mesmos os problemas, a inovação em termos de funcionalidades é limitada. Assim sendo, o que faz uma aplicação destacar-se das concorrentes é o contacto próximo que mantem com os clientes e a disponibilidade em efetuar alterações de acordo com as particularidades de cada um. Este é um aspeto importante porque cada organização tem a sua forma própria de trabalhar e de lidar com as questões de gestão jurídica, o que faz com que uma solução genérica não seja suficiente para responder às suas necessidades. A *kamae RT* está ciente disso e, como tal, toma em consideração o *feedback* dos clientes de forma a tornar a sua solução o mais completa possível e de acordo com as suas especificidades.

## 3 Ferramentas e tecnologias

Neste capítulo são introduzidas as várias ferramentas e linguagens/tecnologias usadas nas várias fases de desenvolvimento do projeto.

### 3.1 Principais ferramentas de desenvolvimento

#### Visual Studio 2012 Ultimate

É um IDE da Microsoft para o desenvolvimento de aplicações *desktop* e *web* recorrendo a linguagens como C#, Visual Basic ou C++ [16].

#### NuGet

Gestor de *plugins* e extensões para a plataforma de desenvolvimento .Net da Microsoft. Possui uma central através da qual o utilizador pode selecionar aqueles que pretende instalar [17].

#### Codepen

É um editor *online* e gratuito de HTML, CSS e Javascript, onde é possível pré-visualizar de imediato o resultado do código escrito sem a necessidade de qualquer tipo de configuração. Possui uma grande comunidade em que diariamente dezenas de novos “projetos” são criados e partilhados por toda a comunidade, sendo possível reutilizar o código em projetos próprios [18].

#### Team Foundation Server 2012

É uma ferramenta de controlo de código fonte, gestão de projeto e uma plataforma de colaboração para permitir um desenvolvimento mais ágil e eficiente na produção de *software* de qualidade [19].

#### FluidUI

É uma ferramenta *online* para a criação de *mockups* de aplicações móveis. Tem ambas versões gratuitas e pagas, sendo que na sua versão gratuita é possível a criação de dez interfaces gráficas por projeto. Possui um conjunto de mais de 3500 *widgets* para utilização nas interfaces, onde é também possível adição de imagens para tornar o aspeto o mais fiável possível com o sistema a desenvolver [20].

## 3.2 Principais linguagens/tecnologias

### C#

Linguagem de programação vastamente usada no desenvolvimento de aplicações *web*, aplicações *desktop* ou *scripting* em tecnologias como Unity para a criação de jogos. É considerada uma evolução das linguagens C e C++, sendo que algumas das suas características foram também retiradas de linguagens como o Java, daí a grande semelhança entre as duas em termos de sintaxe [21].

### LINQ

Linq (Language INtegrated Query) consiste numa biblioteca que permite expressar *queries* em linguagens como C# ou VB, para que não exista a necessidade de utilização de outra linguagem para esse fim [22].

### JavaScript

É uma das linguagens de programação mais usadas em todo o mundo [23], devido à sua utilização para o desenvolvimento de aplicações *web frontend* com a vasta gama de tecnologias disponíveis como jQuery ou Polymer, desenvolvimento para dispositivos móveis com PhoneGap [24], desenvolvimento de aplicações *web backend* com NodeJS [25] ou programação de robôs com *nodebots* [26]. Sendo uma linguagem extremamente simples e versátil é muitas vezes o ponto de começo para principiantes na área da programação.

### CSS

É uma tecnologia que permite um maior controlo da camada de apresentação de uma página *web*. A grande vantagem da utilização desta tecnologia é o facto de permitir separar o código html das modificações gráficas aos vários componentes e tornar mais fácil a manutenção de um *site* ou aplicação *web* [27].

### Media Queries

Media Queries são um módulo do CSS3 que permite que o conteúdo de uma página *web* se adapte a dispositivos de vários tamanhos [9].

### SVG

SVG (Scalable Vector Graphics) é um formato de imagem vetorial muito usado em páginas *web* devido à sua dimensão reduzida e ao facto de poder ser escalável sem que se perca qualidade [28].

### AJAX

Ajax (Asynchronous JavaScript and XML) é uma forma assíncrona de comunicação com o servidor que permite que elementos de uma página *web* sejam atualizados sem que haja necessidade de recarregar toda a página [29].

## JSON

JSON (JavaScript Object Notation) é um formato para guardar e transmitir informação. É uma alternativa ao formato XML [30].

## jQuery

É uma biblioteca de JavaScript desenvolvida com o intuito de facilitar o desenvolvimento de código Javascript para os vários *browsers*. Lançada a 26 de Agosto de 2006 [31], teve uma adoção imediata por parte da comunidade e é atualmente das bibliotecas mais usadas para o desenvolvimento *web*.

Esta biblioteca conta com um conjunto de funcionalidades como [32]:

- Manipulação de HTML/DOM
- Manipulação de CSS
- Eventos HTML
- Animações e efeitos
- Chamadas AJAX

E um vasto conjunto de *plugins* disponíveis para realizar quase qualquer tarefa

Embora cada vez mais existam tecnologias que visam cumprir os mesmos propósitos, o jQuery continua a ter uma forte presença no mercado atual.

De seguida encontram-se descritos os vários plugins baseados em jQuery utilizados no projeto.

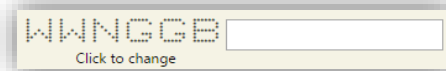
*Tabela 1 - Plugins jQuery*

PLUGIN	DESCRIÇÃO
<b>jQuery unobtrusive validation [33]</b>	Usado para permitir que as validações definidas nos modelos das vistas sejam feitas do lado do cliente, evitando que os dados sejam enviados ao servidor para serem validados.
<b>Timepicki [34]</b>	Componente usado para a seleção de data.



*Figura 1 - Timepicki*

<b>Moment [35]</b>	Usado para facilitar os cálculos em datas.
<b>Real person [36]</b>	Plugin usado para a implementação de um sistema de captcha. A validação é feita quando o pedido login é enviado ao servidor



*Figura 2 - Real person captcha*

---

## **jQuery Mobile**

jQuery *Mobile* é uma tecnologia recente, sendo que a versão mais atual (1.4) foi lançada a 31 Outubro de 2014 com o propósito de tornar mais fácil e intuitivo o desenvolvimento de aplicações *web* focadas em dispositivos móveis, abstraindo o programador da codificação do *layout* das páginas para os vários tamanhos de ecrãs e permitindo que haja maior ênfase na lógica da aplicação.

O conceito base por trás desta tecnologia é o de utilizar uma única pagina para toda a aplicação e carregar dinamicamente o conteúdo durante a navegação, fazendo com que o uso de dados seja menor e a experiência do utilizador seja, por consequência, melhor. Existem atualmente outras tecnologias com conceitos semelhantes, tais como o AngularJS ou até mesmo o Google Polymer. O que diferencia o jQuery *Mobile* das restantes tecnologias é a grande facilidade de uso e aprendizagem, pois a sintaxe é muito semelhante ao HTML nativo, sendo que a única diferença é a utilização de um conjunto de novas propriedades denominadas de atributos de data como “data-roles” que adicionam funcionalidades extras aos vários componentes HTML.

A forma como o jQuery *Mobile* permite a criação de uma aplicação *single page* é recorrendo a propriedades “data-role”, onde apenas uma página tem a “data-role=page”, assinalando que nessa aplicação apenas essa página é considerada com uma página e as restantes como conteúdo. As restantes páginas podem também conter um conjunto de “data-roles”, dependendo da sua finalidade.

Alguns dos atributos existentes são [37]:

- data-role = “collapsible”
- data-role = “control group”
- data-role = “header”
- data-role = “footer”
- data-role = “popup”
- data-fullscreen = “true | false”

## **Browser support**

A forma de avaliação da compatibilidade de *browser* reside num sistema de 3 níveis, A (Completa), B (Completa sem Ajax) e C (Apenas HTML). A rapidez de execução e a fluidez de transições e efeitos visuais está diretamente ligada com a plataforma e características de *hardware* do dispositivo, assim como com a sua capacidade de renderização de CSS [38].

Na Tabela 2 encontra-se uma lista dos principais dispositivos, sistemas e *browsers* testados e o seu nível de compatibilidade.

*Tabela 2 - Browser support*

<b>Compatibilidade (A)</b>	<b>Compatibilidade (B)</b>	<b>Compatibilidade (C)</b>
Apple iOS 4 - 8.1	Opera mini 7	IE 7 ou inferior
Android 5.0 (Lollipop)	Nokia Symbian ^3	Apple iOS 3.x e inferior
Android 4.4 (KitKat)		Blackberry 4-5
Android 4.1-4.3 (Jelly Bean)		Windows <i>Mobile</i>
Android 4.0 (ICS)		Smartphones e plataformas mais antigas.
Android 3.2 (Honeycomb)		
Android 2.1-2.3		
Windows Phone 7.5-8.1		
Blackberry 6-10		
Blackberry Playbook (1.0-2.0)		
Palm <i>WebOS</i> (1.4-3.0)		
Firefox <i>Mobile</i> 18		
Chrome for Android 18		
Skyfire 4.1		
Opera <i>Mobile</i> 11.5-12		
Meego 1.2		
UC Browser		
Kindle 3, Fire, and Fire HD		
Chrome <i>Desktop</i> 16-43		
<i>Desktop</i>		
Safari <i>Desktop</i> 5-8		
<i>Desktop</i>		
Firefox <i>Desktop</i> 10-38		
Internet Explorer 8-11		
Opera <i>Desktop</i> 10-25		



## Asp.Net MVC

Asp.Net MVC é uma *framework* baseada no padrão MVC, em que o principal objetivo é simplificar o desenvolvimento de aplicações *web* e garantir que boas práticas de desenvolvimento são aplicadas.

Esta *framework*, como qualquer outra, possui um conjunto de regras que devem ser seguidas para que seja possível o desenvolvimento. Uma dessas regras é a nomenclaturas dada aos dados. Numa aplicação Asp.Net MVC, ao ser criada uma ação com a qual vai estar associada uma vista, deve atribuir-se a mesma nomenclatura à vista pois, caso contrário, esta não será encontrada. Para os controladores, as nomenclaturas devem seguir o padrão “Nome”+Controller.

Outra regra também bastante importante é a localização dada aos ficheiros. Com efeito, cada tipo de ficheiros tem um local próprio na aplicação, como por exemplo, os controladores devem estar na pasta Controllers, as vistas na pasta Views e os modelos na pasta Models. Dentro de cada umas destas pastas podem existir sub-pastas com o nome do controlador em que cada um será utilizado. Ao serem seguidas estas regras básicas, garante-se um funcionamento correto da aplicação.

A Figura 4, apresentada em seguida, ilustra o funcionamento de uma aplicação Asp.Net MVC.

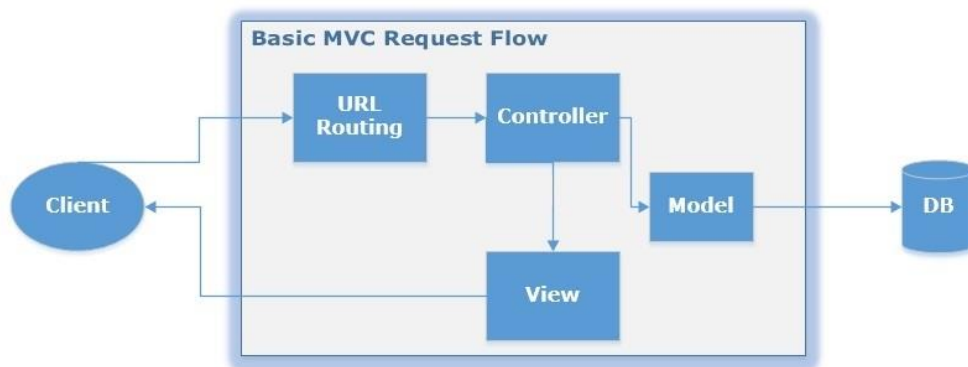


Figura 3 - Asp.Net MVC

- Route – Mecanismo que descreve o controlador para onde o pedido deve ser redirecionado.
- Modelo – Local onde é colocada toda a lógica de negócio;
- Vista – Forma como os componentes visuais são apresentados aos utilizadores;
- Controlador – Forma de comunicação entre o modelo e a vista

### Funcionamento

O funcionamento base de uma aplicação Asp.Net MVC, sem adição de qualquer tipo de tecnologias externas ou componentes, é descrito pelo envio de um pedido por parte de um cliente que é intercetado pelo mecanismo de routing, o qual, por sua vez, o vai mapear para o controlador correto. Ao chegar ao controlador, o pedido é levado para a ação correspondente para ser redirecionado para uma vista. Dentro da ação podem ser usadas classes de modelos como fonte de dados, as quais podem conter dados proveniente de uma base de dados. Depois de efetuado o processamento necessário dentro da ação, é feita uma chamada para a vista associada a essa ação e retornada uma resposta ao cliente.



## 4 Desenvolvimento do projeto

A aplicação KamaeLei Web Mobile, tal como referido anteriormente, é baseada na solução *web* para *desktop* do sistema de informação jurídico KamaeLei. O objetivo era criar uma aplicação separada, dirigida apenas a dispositivos móveis, independentemente da sua dimensão. Para tal deveriam ser implementados os módulos de *login*, gestão de clientes, gestão de contactos, gestão de *dossiers*, gestão de processos, gestão de prazos, gestão de despesas, gestão de notas, gestão de opções e gestão de tarefas (*time-sheet*). No início do estágio já existiam os módulos de *time-sheet* e de *login* que, no entanto, tiveram que ser refeitos, devido a alterações na versão *web* para *desktop* que tiveram que ser levadas em conta. De facto, ao longo de todo o estágio, sempre que se verificaram modificações na aplicação de base, teve que se proceder à sua atualização na versão *web mobile* em desenvolvimento.

Na situação acima referida, as alterações feitas no módulo de *login* foram a resolução de um *bug* que interferia com o correto funcionamento da funcionalidade de *login*, bem como a mudança do mecanismo de *captcha* que inicialmente recorria a um componente *DevExpress* mas que, devido a problemas de compatibilidade com vários dispositivos e a funcionalidades limitadas, teve de ser substituído por outro baseado em *jQuery*, o qual oferece maior compatibilidade, mais funcionalidades e está mais de acordo com o aspeto gráfico da aplicação.

No que diz respeito ao módulo da *time-sheet*, as alterações foram mais acentuadas, visto que foi necessário que certas partes fossem completamente refeitas. A *time-sheet* é uma área de lançamento de tarefas que permite aos utilizadores a definição de um contexto e a sua duração, assim como dados adicionais como pequenas notas. Esta área tem como objetivo auxiliar os utilizadores a conseguirem uma melhor gestão do seu tempo. As alterações a este módulo partiram da necessidade de fazer com que a aplicação *mobile* acompanhasse a evolução da aplicação *web desktop* onde existiam novos dados a serem mostrados, novas validações e novas opções, que consistiam em parâmetros que influenciavam o comportamento dos vários módulos. Outras das alterações efetuadas foi sobre a pesquisa de contexto. Esta consiste num componente que permite ao utilizador efetuar uma pesquisa por cliente, *dossier* ou processo e, dependendo dos dados selecionados, os restantes campos têm os seus dados restritos.

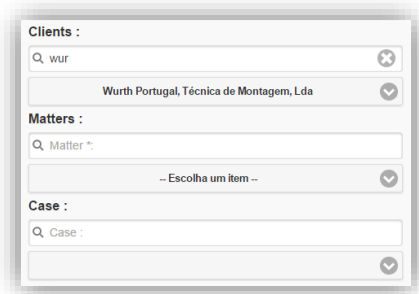


Figura 4 - Exemplo de pesquisa de contexto

O objetivo era adicionar contexto ao que está a ser criado, associando um cliente, *dossier* e eventualmente um processo, caso este seja obrigatório. A utilização deste componente não é apenas restrita à área de *time-sheet*, muito pelo contrário, a sua utilização estende-se por um grande conjunto de áreas na aplicação. Como tal, teve de ser implementado de forma genérica para poder ser usado em todas essas áreas.

## 4.1 Metodologia de desenvolvimento

Atualmente na Kamae RT, todos os projetos recorrem a uma metodologia de desenvolvimento baseada em Scrum [39]. As razões para esta escolha estão associadas ao facto de os projetos estarem propensos a mudanças, fazendo com que as iterações de menor duração tornem mais fácil responder a essas mudanças. Outra grande razão é o facto de permitir ao cliente dar o seu *feedback* e poder efetuar alguma alteração necessária para que aplicação esteja realmente de acordo com as suas necessidades.

Sendo baseado em Scrum, cada projeto segue os padrões delineados pela metodologia, tais como a utilização de um Product Backlog, composto por *user stories*, em que a cada uma delas é atribuído um grau de prioridade. É também utilizado um *Sprint Backlog* composto por um conjunto de *user stories* selecionadas do Product Backlog, onde por norma são escolhidas aquelas com maior prioridade. Existem 3 níveis de prioridade, em que “1” corresponde à mais alta e “3” corresponde à mais baixa.

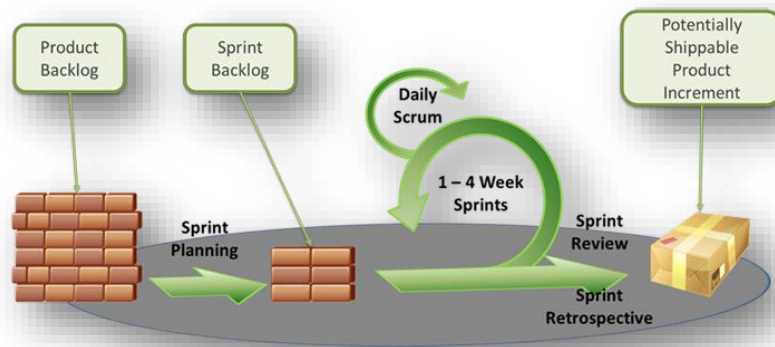


Figura 5 - Funcionamento do processo de scrum

A gestão de projetos é feita recorrendo a duas plataformas: o Team Foundation Server 2012 [19] e o Shaijo. A primeira é uma plataforma para gestão de projetos, que facilita a colaboração e onde é possível efetuar lançamento e consulta de WorkItems para os vários projetos existentes, tais como cenários, tarefas e *bugs*. O Shaijo consiste numa intranet, onde estão disponibilizados dados como a calendarização dos vários eventos e reuniões, boas práticas a serem seguidas, relatórios da área de suporte comercial e de desenvolvimento, entre outros.

São também feitas *Scrum meetings* diárias, normalmente realizadas no final do dia, onde é discutido aquilo que foi feito ao longo do dia, quais os problemas que foram encontrados e aquilo que irá ser feito a seguir. Em semelhança às *Scrum meeting* diárias, no final de cada semana é feito um relatório semanal, onde devem ser descritos os objetivos cumpridos e falhados, os problemas encontrados e objetivos para a semana seguinte. Estes relatórios são aprovados no início da semana seguinte na secção destinada aos relatórios de desenvolvimento no Shaijo. Os objetivos podem sofrer alterações, por parte do aprovador, caso exista maior prioridade no desenvolvimento de alguma tarefa.

A metodologia descrita em cima foi aplicada neste projeto, sendo que a única diferença foi numa fase inicial do estágio, onde os objetivos da *sprint* eram definidos apenas por membros da empresa. No entanto, no decorrer do estágio, estes passaram a ser feitos numa reunião conjunta e só então decididos.

### Equipa de trabalho:

- Ricardo Teixeira – **Product Owner**;
- Arlindo Martins – **Scrum Master**;
- André Quaresma – **Tester**;
- Carlos Nogueira – **Developer**;
- João Mendes – **Developer KLWeb**;
- Daniela Cortesão – **Developer KLWeb**;

## 4.2 Planeamento do projeto

A fase de levantamento de requisitos fugiu um pouco à norma, devido à já existência da aplicação *web desktop*, sendo que inicialmente foi feita uma análise às principais funcionalidades dessa aplicação para que fossem transferidas para este projeto. Depois de se ter sido decidido sobre quais as funcionalidades que iriam ser desenvolvidas e o seu modo de funcionamento, foram criados cenários para cada uma delas. Os cenários consistem em tarefas de maior dimensão e com teor mais generalista. Depois de definidos os cenários, foi definido um conjunto de tarefas para cada uma das funcionalidades estipuladas. Caso houvesse necessidade, novas tarefas poderiam ser definidas.

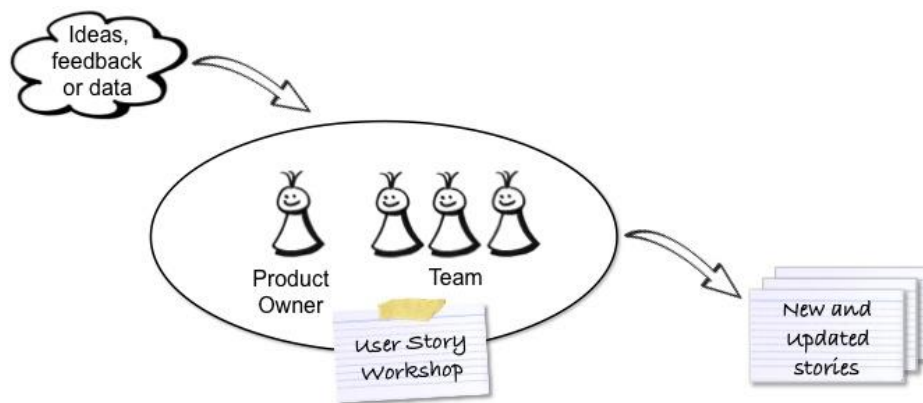


Figura 6 - Definição de requisitos

As funcionalidades e os cenários a desenvolver foram definidos numa reunião conjunta entre o Product Owner e a equipa de desenvolvimento, numa fase inicial, para que fossem discutidas e validadas. Como resultado desta reunião, no total foram criados 8 cenários que correspondem à gestão de *time-sheet*, gestão de clientes, gestão de contactos, gestão de *dossiers*, gestão de notas, gestão de despesas, gestão de prazos e gestão de processos. Para cada um destes cenários, foi então atribuído um conjunto de funcionalidades a implementar. Existia também um conjunto de funcionalidades mais genéricas que não se adaptavam a nenhum dos cenários criados. Como tal, foram colocadas num grupo isolado.

### 4.3 Principais funcionalidades

Apresentam-se de seguida as funcionalidades definidas, organizadas pelos vários cenários criados.

#### Gestão de *time-sheet*

Tabela 3 - Gestão de *time-sheet*

Nº	FUNCIONALIDADE	PRIORIDADE
1	Inserir, editar e remover tarefas de <i>time-sheet</i> .	1
2	Adicionar e/ou mudar o contexto de tarefas de <i>time-sheet</i> novas ou de já existentes.	1
3	Efetuar o lançamento de tarefas de <i>time-sheet</i> em departamentos que não o menu por predefinição.	2
4	Lançar tarefas de <i>time-sheet</i> em <i>dossiers</i> já fechados.	2
5	Impedir o lançamento de tarefas que excedam as 24 horas por dia por advogado.	2
6	Impedir o lançamento de tarefas em que a data de lançamento já tenha sido ultrapassada em “X” número de dias.	2
7	Definir obrigatoriedade na especificação de um processo para um <i>dossier</i> quando a criação de uma tarefa de <i>time-sheet</i> .	2
8	Obrigar lançar tarefas com títulos predefinidos.	2
9	Listar e pesquisar por todas as tarefas lançadas.	1

#### Gestão de clientes

Tabela 4 - Gestão de clientes

Nº	FUNCIONALIDADE	PRIORIDADE
10	Inserir, editar e remover clientes.	1
11	Inserir de forma rápida novos clientes.	1
12	Validação de dados sensíveis como contactos telefónicos, emails e valores monetários antes de serem submetidos.	2
13	Especificar tipos de clientes singulares e coletivos	1
14	Inserir clientes com informações básicas quando na inserção de clientes coletivos	2
15	Especificar valores predefinidos para plafond para clientes singulares e coletivos.	3
16	Listar todos os clientes	1

## Gestão de contactos

Tabela 5 - Gestão de contactos

Nº	FUNCIONALIDADE	PRIORIDADE
17	Inserir, editar e remover contactos.	1
18	Inserir de forma rápida novos contactos.	1
19	Validação de dados sensíveis como contactos telefónicos, emails antes de serem submetidos.	2
20	Listar todos os contactos.	1

## Gestão de *dossiers*

Tabela 6 - Gestão de *dossiers*

Nº	FUNCIONALIDADE	PRIORIDADE
21	Inserir, editar e remover <i>dossiers</i> .	1
22	Inserir de forma rápida novos <i>dossiers</i> .	1
23	Apenas permitir a criação de <i>dossiers</i> se especificado pelo menos um cliente.	1
24	Adicionar toda a equipa do utilizador de forma imediata na criação de um <i>dossier</i> .	2
25	Desbloquear/bloquear a seleção de referências internas na criação de um <i>dossier</i> , caso tenha permissão.	3
26	Selecionar um departamento por predefinição para a criação de um <i>dossier</i> .	3
27	Selecionar um tipo de referência interna por predefinição para a criação de <i>dossiers</i> .	2
28	Definir montantes predefinidos para a área financeira na criação de um <i>dossier</i> .	2
29	Definir valores predefinidos para a área de alertas na criação de um <i>dossier</i> .	2



## Gestão de notas

*Tabela 7 - Gestão de notas*

Nº	FUNCIONALIDADE	PRIORIDADE
30	Inserir, editar e remover contactos.	1
31	Listar todos os contactos.	1
32	Adicionar e/ou mudar o contexto de notas novas ou já existentes.	1

## Gestão de despesas

*Tabela 8 - Gestão de despesas*

Nº	FUNCIONALIDADE	PRIORIDADE
33	Inserir, editar e remover despesas.	1
34	Adicionar e/ou mudar o contexto de despesas novas ou de já existentes.	1
35	Calcular automaticamente o montante da despesa baseado na quantidade, custo unitário e taxa escolhidas.	1
36	Tornar uma despesa isenta de taxas	2
37	Selecionar tipos de taxas e aplicar taxa ao valor da despesa.	1
38	Efetuar o lançamento de uma despesa já paga.	2

## Gestão de prazos

*Tabela 9 - Gestão de prazos*

Nº	FUNCIONALIDADE	PRIORIDADE
39	Inserir, editar e remover prazos.	1
40	Listar todos os prazos.	1
41	Especificar o tipo de julgamento e o seu estado de atual, quer na criação quer na edição	1
42	Adicionar e/ou mudar o contexto de prazos novos ou já existentes.	1

## Gestão de processos

*Tabela 10 - Gestão de processos*

Nº	FUNCIONALIDADE	PRIORIDADE
43	Inserir, editar e remover processos.	1
44	Inserir de forma rápida novos processos.	1
45	Adicionar todos os membros da equipa na criação de um processo.	3
46	Adicionar todos os membros da equipa do <i>dossier</i> na criação de um processo.	3
47	Selecionar o departamento e o representante dos <i>dossiers</i> por predefinição.	3
48	Escolher se a referência interna é ou não obrigatória.	1
49	Escolher o tipo de referência interna a utilizar na criação de processos.	2
50	Adicionar uma área dedicada a processos de qualidade judicial na criação de processos.	1
51	Listar todos os processos	1

## Funcionalidades gerais

*Tabela 11 - Funcionalidades gerais*

Nº	FUNCIONALIDADE	PRIORIDADE
52	Aceder a qualquer funcionalidade da aplicação de forma imediata, a partir de qualquer parte da aplicação, para facilitar a navegação.	1
53	Substituir termos chave na aplicação, por texto específicos à organização.	3
54	Adicionar uma área para a configuração das várias opções que influenciam o funcionamento da aplicação.	3

#### **4.4 Requisitos não funcionais**

1. A aplicação deve funcionar em qualquer tipo de dispositivo móvel baseado em iOS 4 ou superior, Android 4.0 ou superior e Windows Phone 7.5 ou superior, independentemente das suas dimensões.
2. A navegação deve ser simples, fluida e intuitiva em qualquer parte da aplicação.
3. O resultado das ações do utilizador na aplicação devem ser comunicadas de forma clara e concisa.
4. A interface de utilizador deve seguir o mesmo *design* que as aplicações móveis existentes.
5. A interface deve ser adaptada para visualização em modo “Portrait” ou em modo “Landscape”.
6. A quantidade de dados utilizada deve ser reduzida de forma a permitir a utilização da aplicação em dados móveis, sem que o consumo seja excessivo.
7. A aplicação deve estar preparada para a adição de novos módulos.
8. As listagens devem apenas mostrar os primeiros 50 registos baseados na data de criação.
9. As pesquisas efetuadas dentro de uma listagem de registos devem ser feitas em todos os registos existentes e não apenas nos registos mostrados nas listagens.
10. A linguagem do texto na aplicação deve estar de acordo com a língua, por defeito, do utilizador.

## 4.5 Programa de trabalhos

Para o presente estágio, foi definido um planeamento constituído por várias tarefas a serem realizadas ao longo do mesmo. As tarefas eram as seguintes:

- Tarefa 1 – Contexto do Projeto – Entender o projeto e seu âmbito.
- Tarefa 2 – Análise de Requisitos – Análise e especificação dos requisitos pela equipa de desenvolvimento.
- Tarefa 3 – Tecnologia – Entender a arquitetura e a tecnologia que envolve os requisitos.
- Tarefa 4 – Desenvolvimento – Implementação dos requisitos.
- Tarefa 5 – Testes – Testes ao projeto desenvolvido.

O escalonamento de trabalhos encontra-se definido na tabela abaixo, onde é descrito o tempo, em meses atribuído a cada tarefa a realizar.

Tabela 12 - Escalonamento de trabalhos

TAREFAS	MESES								
	1	2	3	4	5	6	7	8	9
T1	█	█	█						
T2		█	█	█	█				
T3		█	█	█					
T4			█	█	█	█	█	█	█
T5									█

## 4.6 Atividades no decorrer do estágio

Ao longo do estágio existiram no total 9 *sprints*, em que cada uma tinha a duração de um mês. O progresso das atividades realizadas ao longo da *sprint* era verificado através de reuniões de *scrum* diárias, relatórios semanais e mensais.

Para cada uma das funcionalidades descritas anteriormente, eram criadas várias tarefas que eram lançadas sobre a forma de Workitems no Team Foundation server.

### ***Sprint 1***

A *Sprint 1* corresponde ao início do estágio e foi aquela que permitiu ter um primeiro contacto com a aplicação a desenvolver. Esta foi a *sprint* de menor duração uma vez que o estágio se iniciou a 8 de Outubro. Durante o decorrer desta *sprint* foi feita uma análise ao estado atual do projeto e à versão *web desktop* da KamaeLei para averiguar as diferenças existentes em relação ao que já estava implementado. Foram também revistos os Workitems (tarefas e *bugs*) lançados para o projeto, para serem novamente validados.

Durante esta *sprint* foram feitas várias reuniões com os membros da equipa para definir os objetivos e as prioridades para *sprints* futuras.

### ***Sprint 2***

A *sprint 2* caracterizou-se pela especificação das funcionalidades a desenvolver no projeto e pela escolha das tecnologias a utilizar. Nesta fase foi feita uma análise às tecnologias existentes e ao seu impacto na aplicação, no seu estado atual. Depois de escolhidas as tecnologias a usar, foi abordado o aspeto gráfico da aplicação, tendo sido criados *mockups* para servirem como base ao desenvolvimento das interfaces da mesma.

Nesta fase os objetivos foram cumpridos mais cedo do que esperado, fazendo com que os objetivos da *sprint* seguinte fossem antecipadamente iniciados.

### ***Sprint 3***

A *sprint 3* marcou formalmente o início da fase de desenvolvimento da aplicação. Nesta *sprint* o objetivo foi de colocar a área de *time-sheet* a funcionar corretamente. Isto incluiu o desenvolvimento da área de listagem de tarefas e a correção das funcionalidades de criação e edição de tarefas. Nas funcionalidades de criação e edição, a parte do servidor teve de ser totalmente refeita, pois não estava implementada de acordo com os padrões da *framework*. No que diz respeito às vistas para estas funcionalidades, foi aproveitada grande parte da estrutura da página, mudando apenas alguns elementos para estar de acordo com a versão *web desktop*. Ainda para estas funcionalidades, tiveram que ser implementadas validações para os vários campos, de forma a impedir a submissão incompleta ou com dados errados. A criação de componentes para a seleção das horas de uma tarefa e a correção do mecanismo de pesquisa de contexto, foram outras das tarefas realizadas.

### ***Sprint 4***

A *sprint 4* teve como objetivos a continuação do desenvolvimento da área de clientes e o início do desenvolvimento da área de *dossiers*, onde se optou pelo desenvolvimento da área de gestão de notas, devido ao facto de ser de dimensão reduzida. Nesta área foram implementadas as funcionalidades de inserção, remoção edição e listagem. A área de clientes, assim como a área de *time-sheet*, já tinha o seu desenvolvimento iniciado. No entanto, a parte do servidor não se encontrava de acordo com os padrões da *framework* e a parte de cliente estava só parcialmente completa. Como tal, grande parte teve que ser refeita, aproveitando-se apenas algumas das vistas. Nesta área foram implementadas as funcionalidades de inserção completa de clientes, que estava dividida em clientes singulares e coletivos, com dados específicos a cada um deles. Foram também implementadas as funcionalidades de inserção rápida que continham apenas as informações essenciais para a criação de um cliente, a funcionalidade de edição que permitia editar os dados de um cliente, recorrendo às vistas da versão completa do *wizard* de clientes e, por fim, a funcionalidade de listagem de clientes que listava os clientes criados e permitia acesso à edição e remoção de clientes.

### ***Sprint 5***

A *sprint 5* teve como objetivo o desenvolvimento das áreas de gestão de *dossiers* e gestão de processos. Para estas áreas foram criados dois *wizards*, um para a gestão de *dossiers* e outro para a gestão de processos. O *wizard* de *dossiers* tem duas versões: uma versão completa, composta por todos os dados relativos a um *dossier* e uma versão simples, contendo apenas os dados essenciais. O *wizard* de processos, ao ter menor dimensão, não teve necessidade de uma versão simples.

A área de gestão de *dossiers* demorou mais tempo a ser desenvolvida, não só devido à sua maior dimensão, mas também ao facto de terem de ser desenvolvidos componentes de seleção de dados sob a forma de listagens, para a seleção de clientes, representantes, entre outros.

Foram também desenvolvidas as funcionalidades de listagem, edição e remoção, para ambas as áreas.

### ***Sprint 6***

Na *sprint 6* os objetivos foram o desenvolvimento das áreas de despesas e de prazos, onde foram implementadas as funcionalidades de inserção, edição, remoção e listagem. Nesta *sprint* não foi possível concluir todas as tarefas delineadas, devido a problemas relacionados com cálculo de montantes na área das despesas, sendo que algumas dessas tarefas foram passadas para a *sprint* seguinte.

### ***Sprint 7***

Na *sprint 7* foi concluída a área de despesas e começado o desenvolvimento dos objetivos para a nova *sprint*. Os objetivos consistiam na implementação da área de gestão de contactos composta por um formulário completo e um formulário simples, onde deveria ser possível a adição e edição de contactos. Para esta área foram ainda implementadas as funcionalidades de listagem e remoção de contactos.

Foi também iniciada a implementação de opções que consistiam num conjunto de parâmetros que afetavam o funcionamento das várias áreas da aplicação. Sendo que, neste caso, foram implementadas as opções de cliente, opções de *dossier* e opções de processo.

### ***Sprint 8***

Na *sprint 8*, os objetivos consistiam na implementação das opções de escritório, que permitiam a mudança de termos chave na aplicação para se adaptar à organização e as opções de lançamento que afetavam diretamente as áreas de *time-sheet* e gestão de despesas.

Nesta área, os objetivos foram concluídos mais cedo do que esperado e, como tal, foi aproveitado o tempo para a resolução de *bugs* reportados, mais concretamente *bugs* relacionados com a área de login, onde foi mudado o mecanismo de *captcha* e corrigido um *bug* que impedia que o login fosse corretamente efetuado.

### ***Sprint 9***

Na *sprint 9*, deu-se início à fase de testes da aplicação. Nesta fase a aplicação foi exaustivamente testada pelo estagiário e pela equipa de testes da Kamae RT, num vasto número de dispositivos e sistemas. Esta fase também foi usada para a resolução de Workitems deixados em aberto numa fase anterior.

## 4.7 Mockups

Os *mockups* têm como propósito geral, auxiliar o desenho de interfaces para uma aplicação. Estes podem ser de dois tipos: *mockups* de alta fiabilidade e de baixa fiabilidade. A diferença entre os dois tipos está no facto de os de alta fiabilidade descreverem em concreto todos os aspetos em termos de desenho gráfico, ao contrário dos de baixa fiabilidade que descrevem de forma genérica e não muito aprofundada a direção em que o aspeto gráfico de uma aplicação deve ser levado.

A necessidade de criação de *mockups* partiu do facto de no início do estágio não existir uma coerência gráfica entre as várias vistas da aplicação e também com as aplicações móveis nativas para os sistemas iOS e Android.

Para este projeto, no total foram criados 9 *mockups* com o objetivo de serem suficientemente genéricos para poderem ser aplicados nas várias áreas, e permitirem uma consistência tanto na navegação dentro da aplicação, como em relação às restantes aplicações móveis. Para tornar possível a similaridade com as aplicações móveis nativas, foi usado o mesmo esquema de cores dessas aplicações, com algumas ligeiras alterações.

A principal razão para terem sido criados apenas 9 *mockups* foi o facto de se estar a usar jQuery *Mobile* para a criação de interfaces gráficas, o que permitia saber à partida o aspeto que os vários elementos teriam quando aplicados numa página *web*, possibilitando que o foco fosse apenas nas áreas principais da aplicação.

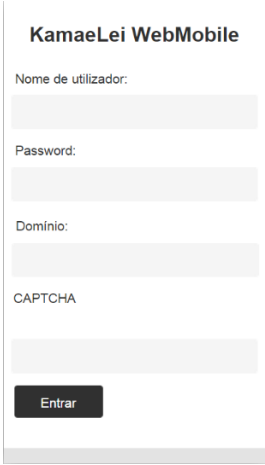

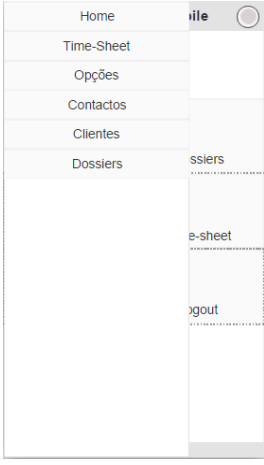
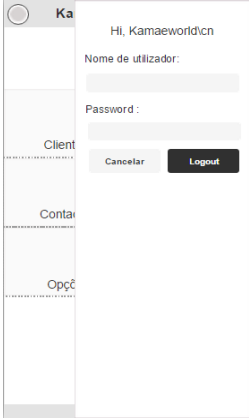
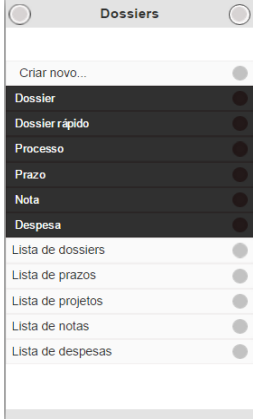
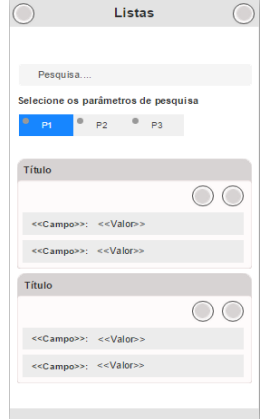
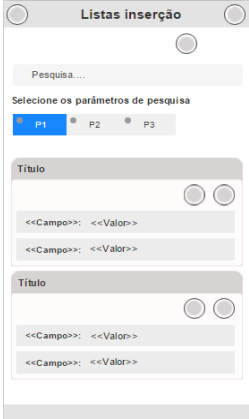
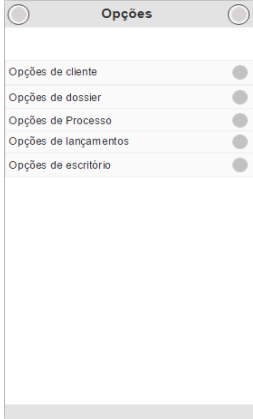

A criação destes *mockups* foi essencial uma vez que, à medida que se foi desenvolvendo a aplicação, existiu muitas vezes a tendência para adicionar elementos extra que, na grande maioria das vezes, pouco ou nada de vantajoso traziam para a aplicação.

Os *mockups* ajudaram a que o *design* da aplicação se mantivesse fiel ao que foi inicialmente especificado.



Encontram-se, em seguida, listados os vários *mockups* criados.

Tabela 13 - Mockups

 <p><b>KamaeLei WebMobile</b></p> <p>Nome de utilizador:</p> <p>Password:</p> <p>Domínio:</p> <p>CAPTCHA</p> <p>Entrar</p>	 <p>KamaeLei WebMobile</p> <p>Cientes      Dossiers</p> <p>Contactos      Time-sheet</p> <p>Opções      Logout</p>	 <p>Home</p> <p>Time-Sheet</p> <p>Opções</p> <p>Contactos</p> <p>Cientes</p> <p>Dossiers</p> <p>Logout</p>
 <p>KamaeLei WebMobile</p> <p>Hi, Kamaeworld!cn</p> <p>Nome de utilizador:</p> <p>Password:</p> <p>Logout</p>	 <p>Dossiers</p> <p>Criar novo...</p> <p>Dossier</p> <p>Dossier rápido</p> <p>Processo</p> <p>Prazo</p> <p>Nota</p> <p>Despesa</p> <p>Lista de dossiers</p> <p>Lista de prazos</p> <p>Lista de projetos</p> <p>Lista de notas</p> <p>Lista de despesas</p>	 <p>Listas</p> <p>Pesquisa...</p> <p>Seleção os parâmetros de pesquisa</p> <p>P1 P2 P3</p> <p>Título</p> <p>&lt;&lt;Campo&gt;&gt;: &lt;&lt;Valor&gt;&gt;</p> <p>Título</p> <p>&lt;&lt;Campo&gt;&gt;: &lt;&lt;Valor&gt;&gt;</p>
 <p>Listas inserção</p> <p>Pesquisa...</p> <p>Seleção os parâmetros de pesquisa</p> <p>P1 P2 P3</p> <p>Título</p> <p>&lt;&lt;Campo&gt;&gt;: &lt;&lt;Valor&gt;&gt;</p> <p>Título</p> <p>&lt;&lt;Campo&gt;&gt;: &lt;&lt;Valor&gt;&gt;</p>	 <p>Opções</p> <p>Opções de cliente</p> <p>Opções de dossier</p> <p>Opções de Processo</p> <p>Opções de lançamentos</p> <p>Opções de escritório</p>	 <p>Listas duplas</p> <p>&lt;&lt;Título&gt;&gt;      &lt;&lt;Título&gt;&gt;</p> <p>Pesquisa...      Pesquisa...</p> <p>&lt;&lt;item&gt;&gt;</p> <p>Anterior      Seguinte</p>

## 4.8 Arquitetura

Na figura 16 está ilustrada a forma como se encontra organizada a aplicação, bem como as tecnologias e componentes usados e a forma como interagem entre si.

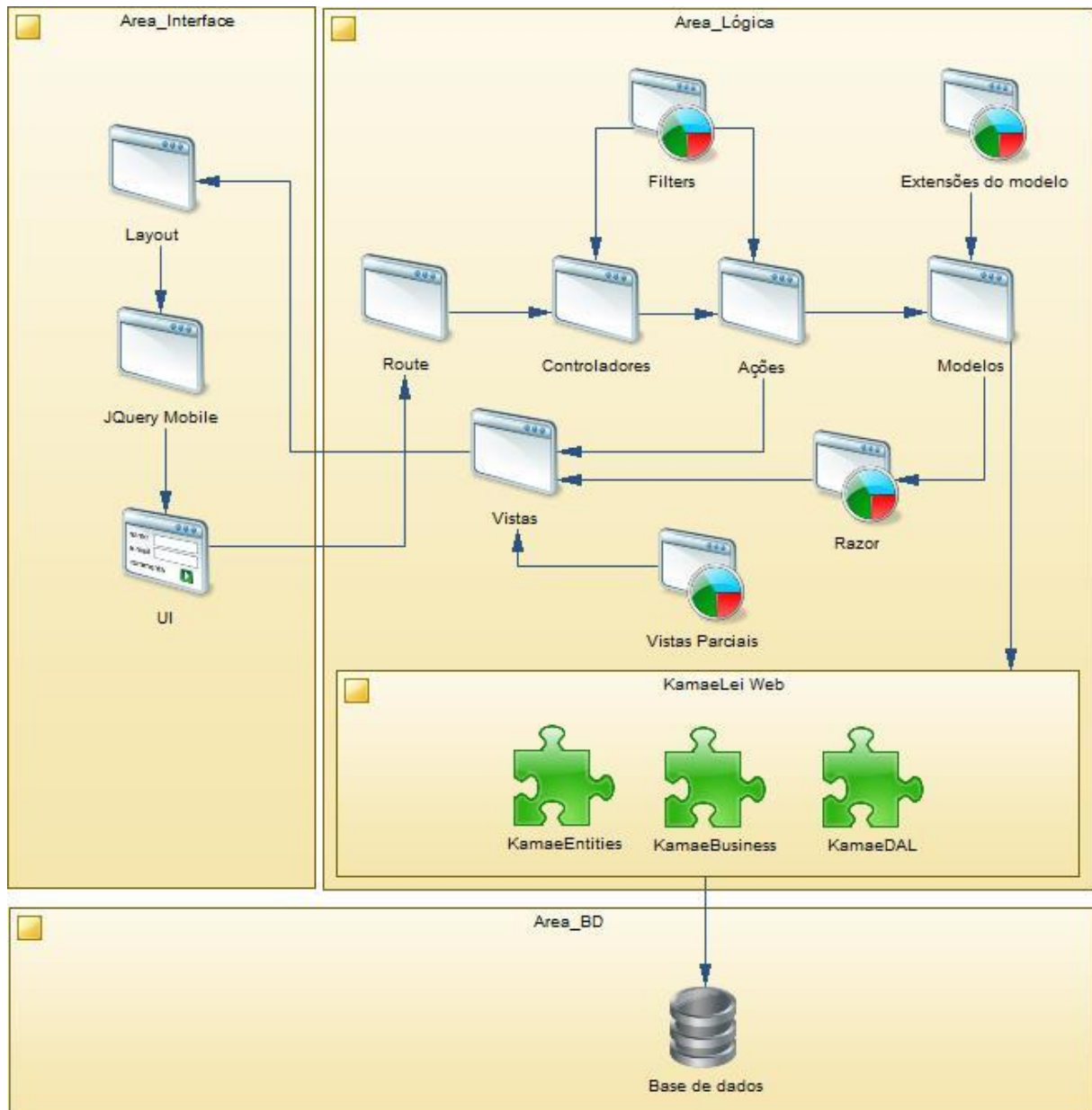


Figura 16 - Arquitetura da aplicação

O funcionamento do diagrama da figura 16 é, em parte, semelhante ao apresentado na descrição da tecnologia Asp.Net MVC, no capítulo anterior. A grande diferença entre os dois reside na adição de outras tecnologias e componentes intermédios que mudam ligeiramente o seu funcionamento.

Esses componentes e tecnologias são:

- Mecanismo de *routing*
- Filters
- Extensões de modelo
- Razor
- Layout
- jQuery Mobile
- KamaeLei Web

### 4.8.1 Routing

Em *Asp.Net MVC* o *routing* vem como parte da *framework* e consiste na forma de descrever ao *webserver* qual o caminho correto para o controlador desejado.

Ao olhar para o código da figura 17, é possível ver alguns *routes* definidos sendo o primeiro o **`routes.IgnoreRoute("{resource}.axd/{*pathInfo}");`**, que vai ignorar qualquer ficheiro com extensão *axd*. Uma extensão *axd* é um tipo especial de extensão para ficheiros que não existem no sistema de ficheiros, mas que são usados para propósitos de *debug*.

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Login", action = "Start", id = UrlParameter.Optional }
        );
    }
}
```

Figura 17 - Routing 1

Nesta versão da aplicação, não foi usado nenhum ficheiro *axd*. No entanto, caso no futuro seja necessária a sua utilização, a aplicação já se encontra preparada para lidar com isso. A não inclusão desta *route* pode ser bastante problemática, pois caso não esteja a ser ignorada, o Asp.Net MVC pode tentar mapeá-la para algo errado e desta forma não chegar ao destino correto.

De seguida encontra-se a o método **`routes.MapRoute(...)`**. Este possui um grande conjunto de “overloads” e é o responsável por permitir ao Asp.Net MVC chegar ao controlador correto.

No desenvolvimento da aplicação foram abordadas duas formas de *routing* que consistiam em implementações diferentes deste método. A primeira forma consistia em fazer o *routing* de uma forma genérica, recorrendo às convenções impostas pela *framework*, como mostrado na figura 17. Neste tipo de *routing*, o campo “name” recebe o valor “default” para que lhe seja atribuído o nome da ação. O campo “url” recebe a sintaxe genérica a ser usada para os URLs, que contém o “nome do controlador”, seguido pelo “nome da ação” e “id de um registo”, caso este exista. Por fim, o campo “defaults” que contém a ação a ser executada quando o URL da página não contem qualquer especificação, por exemplo, “mobile.kamae.pt”.

A segunda forma consistia na utilização de *routes* explícitos. Neste tipo de *routing*, o método **routes.MapRoute(...)**, é também composto pelo nome, URL e *defaults*. No entanto, a grande diferença está na forma como estes são definidos. O campo “name” vai receber o valor pretendido para a identificação da *route*, que pode ser algo como “Home”, desde que seja único, pois vai permitir que o redirecionamento seja feito mesmo quando o endereço muda. O campo “url” vai receber o endereço a ser mostrado no *browser*. Caso o valor do campo seja “login”, o endereço no *browser* será “mobile.kamae.pt/login”. Caso o endereço seja deixado vazio, significa que o URL corresponde à página inicial da aplicação. Por fim, o campo “defaults” contém o controlador e a ação para onde o pedido deve ser redirecionado, como se pode ver na figura 18.

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
        routes.MapRoute("Home", "", new {controller = "Login", action = "Index"});
        routes.MapRoute("Login", "login", new {controller = "Clients", action = "SaveClientInfo"});
    }
}
```

Figura 18 - Routing 2

Cada uma das duas formas tem vantagens e desvantagens, sendo que a forma genérica de fazer *routing* faz com que as *routes* sejam feitos de forma automatizada sem trabalho acrescido para o programador. No entanto, não possibilita a customização do URL. A utilização de *routes* explícitos oferece a possibilidade de personalização do URL, mas, em contrapartida, sempre que se adiciona uma nova ação, uma nova *route* tem de ser adicionada manualmente, para que o sistema a consiga mapear.

A opção escolhida foi a de utilizar a forma genérica de *routing*. Contudo, para ser possível personalizar o URL, recorreu-se a uma funcionalidade do jQuery Mobile denominada de *data-url*, que deve ser colocada na *div* composta pela *data-role="page"*, tal como mostrado em seguida.

- `<div data-role="page" @TempData["DataUrl"]>`

Para que o URL seja realmente personalizável, é então necessário que no final de cada ação, antes de ser feito o redirecionamento para outra ação, sejas definido o URL desejado.

- `TempData["DataUrl"] = "data-url=~/MenuInicial";`

Desta forma, é possível continuar com as *routes* automatizadas podendo ao mesmo tempo personaliza-las.

## 4.8.2 Action filters

*Action filters* são atributos que podem ser aplicados tanto a controladores como a ações específicas, com o objetivo de modificar o seu comportamento. Existe um vasto número de *action filters* já existentes por predefinição em Asp.Net MVC, como por exemplo:

- **OutputCache** –apanha o output de uma ação do controlador por um período de tempo específico.
- **HandleError** –controla os erros gerados quando o controlador executa uma ação.
- **Authorize** –permite restringir o acesso a um utilizador ou a um tipo de utilizador em particular. [40]

Neste projeto foram criados manualmente dois *actions filters*, que foram aplicados inúmeras vezes durante o desenvolvimento.

O primeiro consiste num *action filter* utilizado ao nível da ação, em todas as ações que recorressem ao tipo de dados JSON como dados de retorno. Estas ações, ao contrário do que é normal para ações em Asp.Net MVC, foram utilizadas recorrendo a chamadas AJAX, para que o conteúdo da página fosse atualizado sem que existisse a necessidade da página ser recarregada.

```
public class AllowCrossSiteJsonAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        filterContext.HttpContext.Response.AddHeader("Access-Control-Allow-Origin", "*");
        base.OnActionExecuting(filterContext);
    }
}
```

Figura 19 - Action filter AllowCrossSiteJson

O segundo *action filter* foi utilizado ao nível do controlador, o que significa que qualquer ação pertencente a este controlador será afetada por ele. Este *action filter* é responsável pelo controlo da sessão do utilizador na aplicação, verificando um conjunto de variáveis de sessão para detetar se alguma delas expirou. Caso tenham expirado, o utilizador é redirecionado para a janela de *login* da aplicação.

```
public class SessionExpireAttribute : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        HttpContext ctx = HttpContext.Current;
        // check sessions here
        if (HttpContext.Current.Session["userName"] == null ||
            HttpContext.Current.Session["MatterOptions"] == null ||
            HttpContext.Current.Session["CaseOptions"] == null ||
            HttpContext.Current.Session["TermOptions"] == null ||
            HttpContext.Current.Session["OfficeOption"] == null ||
            HttpContext.Current.Session["LaunchesOption"] == null ||
            HttpContext.Current.Session["CurrentCulture"] == null)
        {
            filterContext.Result = new RedirectResult("~/Login/Start");
            return;
        }
        base.OnActionExecuting(filterContext);
    }
}
```

Figura 20 - ActionFilter SessionExpire

Este *action filter* foi aplicado a todos os controladores presentes na aplicação, visto que pelo menos uma das variáveis de sessão presentes no *action filter* é usada em qualquer controlador. Os valores para essas variáveis são definidos na ação de *login*, como tal, ao expirarem, é necessário que este volte a ser efetuado.

### 4.8.3 Extensões de modelo

Extensões de modelo podem ser entendidas como um complemento a um modelo, que permite que sejam feitas alterações a dados, tais como mudanças de formatações, conversões de dados ou outro tipo de alterações.

Neste projeto foram frequentemente usados para converter o tipo de dados retornados da base de dados para os tipos de dados suportados pelos modelos e vistas da aplicação, de forma a ser possível utilizá-los.

```
public static List<SelectListItem> ToListSentenceResult(this List<CaseSentence> lst)
{
    if (lst.Count == 0)
    {
        CaseSentence ltm = new CaseSentence();
        ltm.IDKLSentenceResult = Convert.ToInt32(@Resources.ModelExtension.ID);
        ltm.SentenceResult = @Resources.ModelExtension.NameText;
        lst.Add(ltm);
    }
    List<SelectListItem> lstCaseSentence = new List<SelectListItem>();
    SelectListItem item = null;
    foreach (CaseSentence c in lst)
    {
        item = new SelectListItem();

        item.Value = c.IDKLSentenceResult.ToString();
        item.Text = c.SentenceResult;

        lstCaseSentence.Add(item);
    }
    return lstCaseSentence;
}
```

Figura 21 - Exemplo de extensão de modelo

Este exemplo mostra a conversão de uma lista de dados do tipo <CaseSentence> para uma lista de dados do tipo <SelectListItem>, que é o tipo necessário para que os dados sejam mostrados numa página sobre a forma de *combo box*.

Sempre que era necessária a conversão de dados, só era preciso chamar o método, como ilustrado em seguida:

- **CaseBusiness.GetAllSentenceResult ().ToListSentenceResult ();**

Para cada área da aplicação existe um ficheiro de extensão de modelo, composto por vários métodos como o demonstrado acima. Estes ficheiros devem ser inseridos na pasta onde se encontram as classes modelo da aplicação.

#### 4.8.4 Razor

Razor é um mecanismo utilizado nas vistas (*views*) de uma aplicação Asp.Net MVC, que permite a utilização de código C# para tornar o conteúdo de uma vista dinâmico, tal como se pode ver pelo código seguinte.

```
@{
    <div id="TermsDiv">
        @foreach (KeyValuePair<string, List<string>> entry in Model.DictionaryList)
        {
            <div id="colapse" data-role="collapsible" style="white-space:normal;">
                @for (int i = 0; i < (entry.Value.Count - 1); i++)
                {
                    if (i == 0)
                    {
                        <h4>@entry.Value[i]</h4>
                    }
                    else
                    {
                        @Html.TextBoxFor(model => entry.Value[i], new { id = "" + @entry.Key + "_" + i + "_", naming =
                            "" + @entry.Key + "" })
                    }
                }
            </div>
        }
    </div>
}
```

Figura 22 - Mecanismo Razor

Este código representa a área das opções de escritório, mais especificamente, a parte associada aos termos. Esta área permite que sejam associados termos personalizados a palavras-chave da aplicação, como *dossiers*, processos, IVA, entre outros.

Este mecanismo permite que uma vista esteja associada a um modelo, para que consiga aceder aos seus dados. A este tipo de vistas é dado o nome de “Strongly typed views”.

Este mecanismo está presente em todas as vistas desenvolvidas ao longo do projeto e permitiu um desenvolvimento mais rápido e simplificado.



## 4.8.5 Layout

O *layout* numa aplicação Asp.Net MVC, consiste numa área onde o código das vistas é inserido. Isto vem resolver o problema de repetição de código pois, por norma, por cada vista que se adicione à aplicação, é necessário repetir todo o código representante da estrutura da aplicação, tal como a definição das *tags head* e *body* ou a importação de ficheiros de *scripts* e CSS. Ao serem utilizados *layouts*, esse código é escrito uma única vez. Após a sua definição, o código das vistas é então colocado num local específico (**@RenderBody()**) nesse *layout*, normalmente dentro da *tag body*.

O exemplo de código apresentado em seguida, demonstra os conceitos aqui referidos.

```
<html>
  <head>
    ...
    @Scripts.Render("~/scripts")
    @Styles.Render("~/Content/mobileCss", "~/Content/css")
    @RenderSection("JavaScript", required: false)
  </head>
  <body>
    <div data-role="page" id="page" class="ui-responsive-panel" @TempData["DataUrl"] data-
    fullscreen="true">
      <div data-role="header">
        <div class="Title">
          <b id="titulo">@ViewBag.title</b>
        </div>
        ...
        <div id="page-content" data-role="content">
          @RenderBody()
        </div>
        ...
      </body>
</html>
```

Figura 23 - Exemplo da aplicação do mecanismo de layout

### 4.8.6 jQuery Mobile

O jQuery *mobile* nesta aplicação teve duas funções principais. A primeira consistia na melhoria dos vários componentes *html* utilizados durante o desenvolvimento da aplicação. Esta melhoria não estava limitada apenas a uma melhoria ao aspeto gráfico dos componentes, mas sim a uma melhoria geral, quer em termos de funcionalidades, como a adição de pesquisas automatizadas para elementos do tipo *combo box*, ou a adição de funcionalidades a simples *tags <div>*, como a possibilidade de as tornar em elementos “collapsible”, quer em termos de estruturação, pois oferecia formas simples e intuitivas de organizar os conteúdos para que estes se adaptassem ao vários tamanhos de ecrãs dos dispositivos.

A segunda função consistia numa maior facilidade de tornar esta aplicação numa aplicação *single-page*. Isto foi conseguido recorrendo a propriedades “data-role”, sendo que, para este caso, a mais importante foi a *data-role=“page”*, que faz com que uma *tag* seja definida como uma página sendo as restantes apenas carregadas como conteúdo.

Esta propriedade foi implementada no ficheiro de *layout* descrito anteriormente, conjugando assim o melhor das duas tecnologias.

Tendo em conta a vasta gama de dispositivos e sistemas onde a aplicação deve correr, algumas das propriedades do jQuery *mobile* tiveram de ser desativadas, pois as suas funcionalidades não eram corretamente executadas em todos os *browsers* e sistemas operativos.

As propriedades colocadas de parte foram:

- A utilização de animações que não só ofereciam à aplicação uma navegação mais intuitiva, devido aos efeitos gráficos melhorados, como também tornavam menos frustrante o carregamento das vistas quando a ligação se encontrava lenta, pois em vez de estar à espera que a página carregasse completamente para ser mostrada, a mudança de página era feita de imediato e os componentes iam sendo preenchidos à medida que o carregamento fosse ocorrendo.
- A utilização de *caching*, que fazia com que fosse mantido um “histórico” em memória com as várias vistas por onde o utilizador passava, para que caso ele voltasse a aceder a alguma delas, o acesso fosse mais rápido. A razão pela qual esta propriedade foi removida, foi devido ao facto de dispositivos mais antigos e como menos capacidade de memória e processamento, não serem capazes de lidar com esta maior quantidade de informação.

À medida que a capacidade dos dispositivos for aumentado e os dispositivos mais antigos deixarem de ser usados, é indicado que estas propriedades sejam de novo ativadas, pois melhoram a rapidez e a fluidez da aplicação e, consequentemente, a experiência de utilizador.

#### 4.8.7 KamaeLei Web

A área de KamaeLei Web, consiste num conjunto de DLLs provenientes da versão *web* para *desktop*, que contém um vasto conjunto de métodos, desde métodos responsáveis pela devolução de listagens de dados relativos a clientes, *dossiers* e processos, até métodos que permitem a inserção edição e remoção de registos.

Estas DLLs são a forma de comunicação entre este projeto e a base de dados. Foram no total usadas três DLLs distintas, sendo elas :

- **KamaeEntities** – DLL que permite o acesso a dados de entidades como departamentos, prazos, contactos clientes, categorias, entre outros.
- **KamaeBusiness** – DLL que disponibiliza o acesso a métodos para inserção, edição e remoção.
- **KamaeDal** – DLL constituída por métodos relacionados com entidades e que permite que sejam feitas chamadas às SP's (*stored procedures*).

Existem mais DLLs que embora não tenham sido usadas diretamente na aplicação, necessitam de estar presentes para que esta funcione corretamente, tais como KSIUtil que possui um conjunto de métodos de conversões de dados ou KSIExceptionHandling que lida com o controlo de erros. Métodos presentes nestas DLLs são usados nas três DLLs descritas anteriormente.

#### 4.8.8 Partial Views

*Partial views*, como o próprio nome indica, consistem em vistas parciais, com o propósito de serem inseridas dentro de outras vistas. São por norma usadas quando existe uma funcionalidade ou um componente que é utilizado em várias áreas de uma aplicação e em que o código desse componente tem de ser repetido por cada vez que é implementado.

Ao ser usada uma *partial view*, o código tem de ser escrito apenas um vez e depois renderizado nas vistas necessárias, permitindo a obtenção de um maior ganho de produtividade e simplicidade.

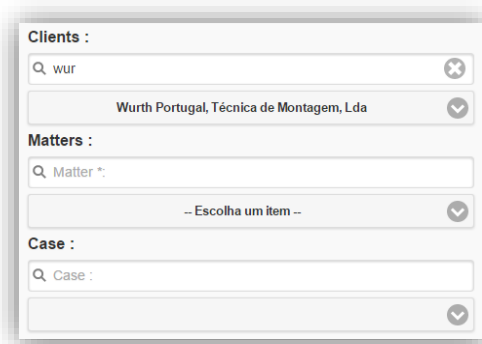
## 4.8.9 Comunicação AJAX entre cliente e servidor

Nesta área serão descritas as principais áreas ou funcionalidades que recorrem a pedidos Ajax para funcionarem corretamente. Os pedidos Ajax são usados para seja possível a existência de dinamismo nos dados mostrados numa página sem que esta tenha de ser recarregada.

### Pesquisa de contexto

A pesquisa de contexto é um exemplo da aplicação de uma *partial view* num vasto conjunto de vistas. O objetivo desta é permitir ao utilizador adicionar contexto a uma certa área na aplicação, quer seja uma nota, uma tarefa, processo, despesa ou prazo.

Na sua versão completa, é constituída por três campos: um campo de pesquisa de cliente, outro de pesquisa de *dossier* e outro de pesquisa de processo. Na sua versão simplificada, é apenas constituída pela pesquisa de cliente e pesquisa de *dossier*. Esta versão é apenas usada na criação de um processo uma vez que, nesse caso, não faz sentido a existência desse campo.



The screenshot shows a search interface with three sections: 'Clients', 'Matters', and 'Case'. Each section has a search input field and a dropdown menu. The 'Clients' section has a search input with 'wur' and a dropdown menu showing 'Wurth Portugal, Técnica de Montagem, Lda'. The 'Matters' section has a search input with 'Matter \*' and a dropdown menu showing '-- Escolha um item --'. The 'Case' section has a search input with 'Case :' and a dropdown menu.

Figura 24 - Pesquisa de contexto 2

A forma de funcionamento consiste em limitar os registos mostrados em cada campo, com base na pesquisa efetuada. Por exemplo, ao ser efetuada uma pesquisa por cliente com o texto “Wur”, a *combo box* relativa ao cliente vai ser preenchida com todos os clientes que contêm a *string* “Wur” no nome. No entanto, nenhuma é selecionada. Ao aceder à *combo box* e ao selecionar um registo, as restantes *combo boxes* serão preenchidas gradualmente.

O preenchimento de dados nas *combo boxes* após uma pesquisa, varia consoante por qual dos campos a pesquisa foi efetuada. O preenchimento segue a seguinte lógica:

- **Pesquisa por cliente** – *Combo box* de cliente → *combo box* de *dossiers* → *combo box* de processos.
- **Pesquisa por dossier** – *Combo box* de *dossiers* → *combo box* de cliente (preenchida apenas com o cliente associado ao *dossier*) → *combo box* de processos.
- **Pesquisa por processo** – *Combo box* de processos → *combo box* de cliente e *combo box* de *dossier* (preenchidas apenas com o cliente e *dossier* associado ao processo).

Esta forma de preenchimento deve-se ao facto de um cliente poder ter vários *dossiers* e um *dossier* poder ter vários processos. E também ao facto de um *dossier* ter que estar associado a um cliente e um processo ter que estar associado a um *dossier*. Estão disponíveis diagramas de sequência para a pesquisa de contexto no **Anexo A**.

## Listagem de registos

A listagem de registos é uma parte fulcral e que está presente em cada uma das principais áreas da aplicação. Esta área conta com um conjunto de funcionalidades que recorrem a pedidos Ajax para efetuar a comunicação com o servidor, tais como a remoção de registos e a pesquisa dinâmica existente nesta área.

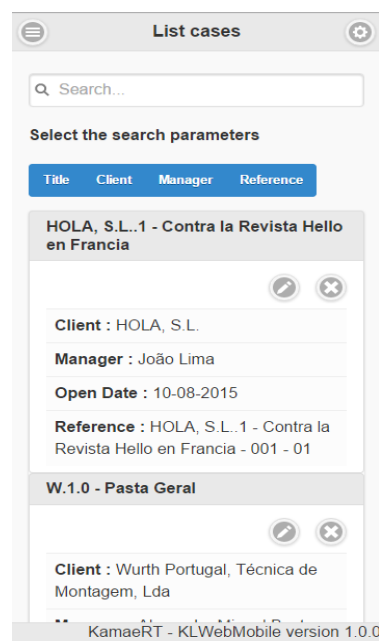


Figura 25 - Exemplo de listagem

A remoção baseia-se num simples pedido ao servidor para eliminar um determinado registo através do seu *id*. O servidor por sua vez vai verificar se o item em causa existe e se este pode ou não ser eliminado. Se puder ser eliminado, a página é atualizada, Caso contrário, é mostrado um aviso sob a forma de *popup*.

A pesquisa dinâmica baseia-se num mecanismo de pesquisa que atua sobre todos os registos de uma determinada área e não apenas nos mostrados nas listagens (apenas são mostrados os primeiros 50 registos, de forma a melhorar a *performance*). Esta pesquisa é influenciada por um conjunto de parâmetros que o utilizador pode seleccionar. Ao ser efetuada uma pesquisa é feito um pedido ao servidor para que sejam retornados os resultados para o registo procurado. Quando os resultados são retornados, os dados da página são atualizados.

Ao limpar os dados da barra de pesquisa a pagina volta ao seu estanho normal onde são mostrados os primeiros 50 registos. Está disponível no **Anexo A**, um diagrama de sequência que descreve a comunicação entre cliente e servidor para a listagem de registos.

De forma a tornar mais fácil a utilização da pesquisa, foi adicionado um método que identifica a altura em que o utilizador acaba de escrever, para que a pesquisa seja automaticamente efetuada [41].

```
(function ($) {  
  $.fn.extend({  
    donotyping: function (callback, timeout) {  
      timeout = timeout || 1e3; // 1 second default timeout  
      var timeoutReference,  
          doneTyping = function (el) {  
        if (!timeoutReference) return;  
        timeoutReference = null;  
        callback.call(el);  
      };  
      return this.each(function (i, el) {  
        var $el = $(el);  
        // Chrome Fix (Use keyup over keypress to detect backspace)  
        // thank you palerdot  
        $el.is(':input') && $el.on('keyup keypress', function (e) {  
          // This catches the backspace button in chrome, but also prevents  
          // the event from triggering too preemptively. Without this line,  
          // using tab/shift+tab will make the focused element fire the callback.  
          if (e.type == 'keyup' && e.keyCode != 8) return;  
          // Check if timeout has been set. If it has, "reset" the clock and  
          // start over again.  
          if (timeoutReference) clearTimeout(timeoutReference);  
          timeoutReference = setTimeout(function () {  
            // if we made it here, our timeout has elapsed. Fire the  
            // callback  
            doneTyping(el);  
          }, timeout);  
        }).on('blur', function () {  
          // If we can, fire the event since we're leaving the field  
          doneTyping(el);  
        });  
      });  
    }  
  });  
})(jQuery);
```

Figura 26 - Implementação do método *doneTyping*

Para que isto seja possível, é adicionado um evento ao campo que se pretende e, quando a escrita nesse campo é iniciada, é feita uma contagem ao tempo que o utilizador demora a carregar na próxima tecla. Quando esse tempo excede 1segundo (pode ser definido um valor maior) é efetuada a pesquisa.

Esta forma de pesquisa não é suportada em todos os *browsers* e sistemas, no entanto, quando não é suportado, o funcionamento normal de usar a tecla de confirmação do teclado decorre sem problemas.

## Listagens duplas

Listagens duplas consistem em componentes compostos por duas listagens de registos alinhadas horizontalmente, uma correspondente aos itens disponíveis e outra aos itens selecionados, onde ao selecionar um registo de uma das listagens, esse registo é removido e passado para a outra. Isso é feito estabelecendo uma ligação ao servidor que possui duas listas de objetos e onde os itens selecionados são passados de uma lista para a outra.

Cada uma das listagens possui um mecanismo de pesquisa, no entanto, funcionando de forma diferente. O mecanismo de pesquisa de itens disponíveis recorre a uma chamada Ajax ao servidor para efetuar a pesquisa em todos os registos, isto porque na listagem apenas são mostrados os primeiros 50 registos. Na listagem dos itens selecionados recorre-se a um componente do jQuery Mobile que permite efetuar, de forma automática, uma pesquisa aos itens presentes na listagem.

Ao limpar os dados de pesquisa em qualquer uma das listagens, estes voltam ao seu estado anterior. Está disponível no **Anexo A** um diagrama de sequência que descreve a comunicação entre cliente e servidor para as listagens duplas.

## Gestão de opções

A gestão de opções é uma área composta por várias subáreas, onde cada uma delas é composta por vários campos que influenciam o funcionamento das restantes áreas da aplicação. Os valores para estes campos podem ser alterados de acordo com as preferências do utilizador. A forma de guardar os valores desses campos é feita recorrendo também a pedidos Ajax.

A comunicação nesta área limita-se à chamada de um método no servidor que guarda as alterações efetuadas e retorna uma resposta de sucesso ou erro.

## Conclusão

Estas são as principais áreas e funcionalidades que recorrem ao uso de pedidos Ajax para comunicação com o servidor. Existem outras mas, devido a sua simplicidade, não foram especificadas como é o caso de:

- Validação do número de horas máximo para o lançamento de tarefas na *time-sheet*.
- Adição de honorário a uma tarefa na *time-sheet*.
- Cálculo do montante de uma despesa, baseado na taxa escolhida.
- Criação da referência interna para um processo, após a escolha de um cliente e *dossier* na pesquisa de contexto.

#### 4.8.10 Outros aspetos

Foram também criadas classes que facilitaram o desenvolvimento e identificação de erros, como a classe “JsonHelper”, que permite serializar e *deserializar* um objeto, independentemente do seu tipo. Esta classe foi amplamente usada em ações que retornavam Json como tipo de dados. Estas ações eram, por norma, acedidas recorrendo a pedidos Ajax provenientes do cliente.

```
public class JsonHelper
{
    /// <summary>
    /// JSON Serialization
    /// </summary>
    public static string JsonSerializer<T>(T t)
    {
        DataContractJsonSerializer ser = new DataContractJsonSerializer(typeof(T));
        MemoryStream ms = new MemoryStream();
        ser.WriteObject(ms, t);
        string jsonString = Encoding.UTF8.GetString(ms.ToArray());
        ms.Close();
        return jsonString;
    }
    /// <summary>
    /// JSON Deserialization
    /// </summary>
    public static T JsonDeserialize<T>(string jsonString)
    {
        DataContractJsonSerializer ser = new DataContractJsonSerializer(typeof(T));
        MemoryStream ms = new MemoryStream(Encoding.UTF8.GetBytes(jsonString));
        T obj = (T)ser.ReadObject(ms);
        return obj;
    }
}
```

Figura 27 - Classe JsonHelper



Foi também desenvolvida uma classe que permite a criação de *logs* em partes específicas da aplicação, para ser possível a identificação de erros fora do ambiente de desenvolvimento. Ao passar por uma área onde existissem *logs* estes eram guardados numa pasta específica do projeto, denominada “LogFolder”.

```
public class LogHelper
{
    public static void Start()
    {
        Logger.LogDebug = bool.Parse(AppConfigManager.GetConfigIfExists("false", "LogDebug"));
        Logger.LogDirectory = AppConfigManager.GetConfigIfExists("Logs", "LogFolder");
        Logger.LogFile = AppConfigManager.GetConfigIfExists("klwebmobile", "LogFile");
        Append("Iniciou as " + DateTime.Now.ToLongDateString());
    }
    public static void Append(string message)
    {
        Logger.append(message, Logger.INFO);
    }
    public static void Append(string source, string message)
    {
        if (!source.StartsWith("["])
        {
            source = "["+source+": ";
        }
        message = source + message;
        Logger.append(message, Logger.INFO);
    }
}
```

*Figura 28 - Classe LogHelper*

## 4.9 Estrutura do projeto

A estruturação levada a cabo para o projeto foi definida pelos vários módulos aplicativos a serem desenvolvidos, sendo que cada módulo foi associado a uma área da aplicação. Essas áreas são compostas por um controlador, que recorre a um conjunto de ações que permitem a navegação entre as várias vistas, e também a um conjunto de modelos que usa como fonte de dados.

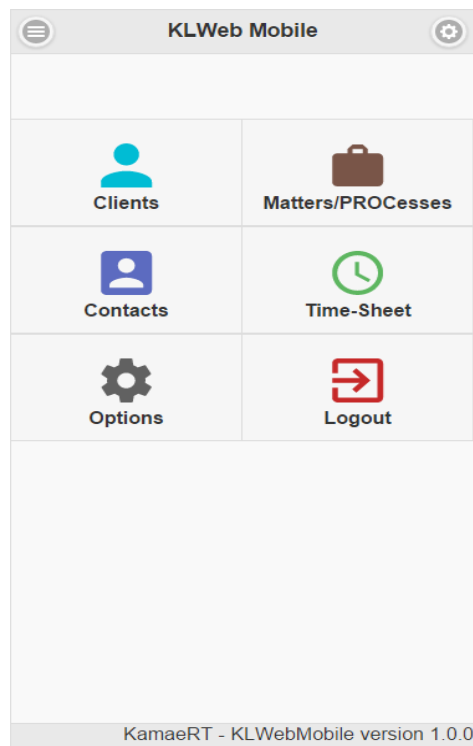


Figura 29 - Página inicial

Existem no total seis áreas: a área de *Dossiers*, a área de Clientes, a área de Opções, a área de Contactos, a área de Login e a área de *Time-sheet*.

### 4.9.1 Área de *Dossiers*

A área de *Dossiers* é a área de maior dimensão e aquela que apresenta maior complexidade. É composta pelos cenários de gestão de *dossiers*, gestão de notas, gestão de despesas, gestão de prazos e gestão de processos.

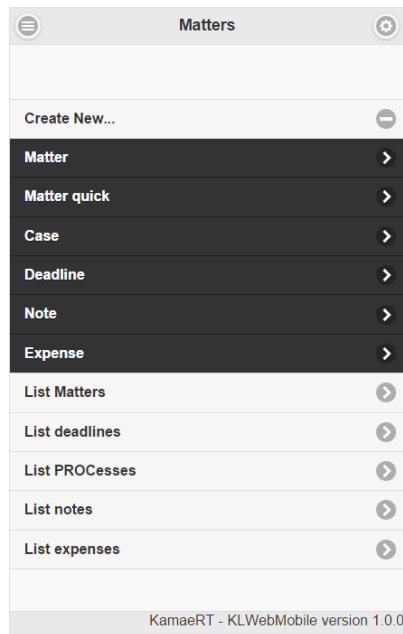


Figura 30 - Área de *dossiers*

#### Gestão de *dossiers*

A parte de gestão de *dossiers* consiste num *wizard* que permite a criação, edição e remoção de *dossiers*. Existem duas versões deste *wizard*: uma versão rápida e uma versão completa. Na versão completa, o assistente de *dossier* tem um total de 10 áreas distintas, sendo elas:

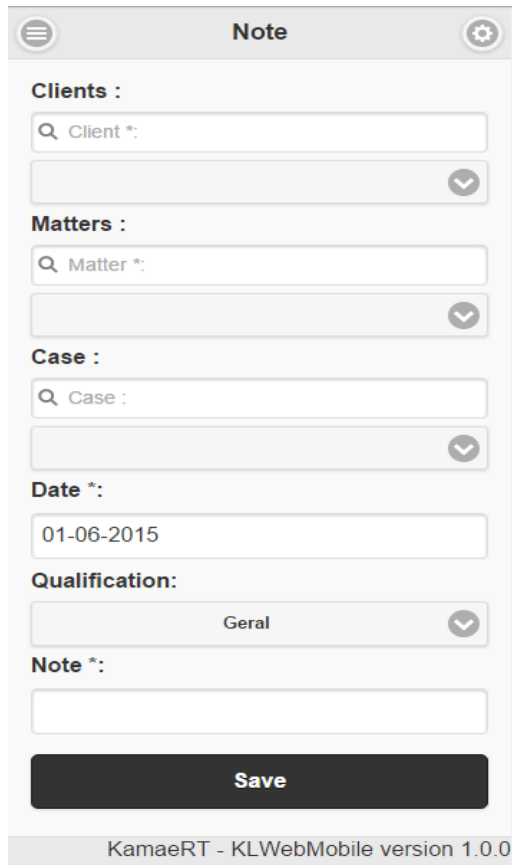
1. **Área de seleção de clientes** – Consiste numa área composta por duas listagens, dispostas lado a lado, onde do lado esquerdo se encontram os clientes por selecionar e do lado direito os clientes selecionados. Nesta área é sempre obrigatória a seleção de pelo menos um cliente.
2. **Área de propriedades gerais** – Consiste numa área constituída pelos campos básicos de um *dossier*, como a data de abertura, a referência interna, a categoria, o estado, o departamento, entre outros.
3. **Área de seleção de equipa** – Consiste numa área composta por 2 listagens idênticas de ambos os lados, onde umas das listagens representa advogados e, a outra, estagiários. As listagens do lado esquerdo representam os advogados e estagiários disponíveis para seleção, enquanto as do lado direito representam os advogados e estagiários selecionados.
4. **Área financeira** – Consiste numa área composta por campos relacionados com a parte financeira de um *dossier*, como o orçamento planeado para o *dossier*, o número de horas previstas, as despesas em plafons, quem foi o angariador e qual a sua percentagem de comissão, entre outros.
5. **Área de custo da equipa** – Consiste numa área formada por todos os membros da equipa selecionados na área número 3, onde é possível definir a data de início e de fim de trabalho no *dossier* para cada membro e também o seu custo por hora.

6. **Área de alertas** – Consiste numa área onde é possível configurar alertas para uma vasto conjunto de campos, como alertas de atividade, alertas de rentabilidade, ou alertas de despesas, assim como o montante para que o alerta seja enviado e a forma de envio (KamaeLei Web e *email*).
7. **Área de dados adicionais** – Consiste numa área formada por um conjunto de campos que podem ser preenchidos com qualquer tipo de dados.
8. **Área de representados da parte oposta** - Consiste numa área composta por duas listagens, dispostas lado a lado, onde do lado esquerdo se encontram as entidades por selecionar e do lado direito as entidades selecionadas.
9. **Área de representantes da parte oposta** - Consiste numa área composta por duas listagens, dispostas lado a lado, onde do lado esquerdo se encontram as entidades por selecionar e do lado direito as entidades selecionadas.
10. **Área de confirmação de dados** – Consiste numa área constituída pela listagem dos valores dos vários campos preenchidos ao longo do *wizard*, que permite ao utilizador verificar rapidamente se todos os campos se encontram corretamente preenchidos.

Na versão rápida, o número de passos é significativamente reduzido, estando apenas presentes as áreas de seleção de clientes, propriedades gerais e a área de confirmação de dados. Para além das duas variantes de assistentes disponíveis para os *dossiers*, existe ainda uma listagem dos *dossiers* em que o utilizador participa ou participou, onde é possível ao utilizador procurar um *dossier* específico, editá-lo ou removê-lo. Ilustrações das várias áreas referidas encontram-se disponíveis no **anexo A**.

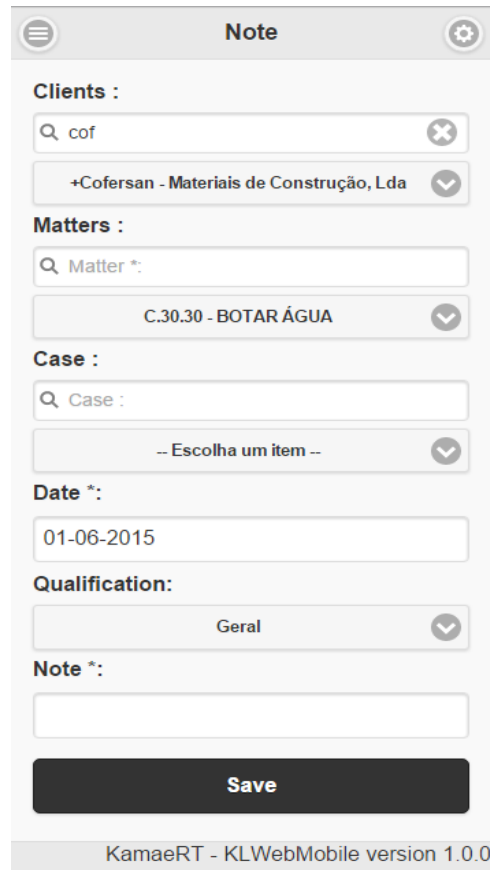
## Gestão de notas

A parte de gestão de notas consiste num simples formulário onde é possível adicionar, editar e eliminar notas de um *dossier* já existente. O formulário, como se pode ver na figura abaixo, é composto por um campo identificador do cliente, um campo identificador de um *dossier*, um campo identificador de um processo, sendo que este não é obrigatório, um campo de data de criação/edição da nota, um campo de qualificação e um campo para as notas.



KamaeRT - KLWebMobile version 1.0.0

Figura 31 - Formulário de notas sem contexto



KamaeRT - KLWebMobile version 1.0.0

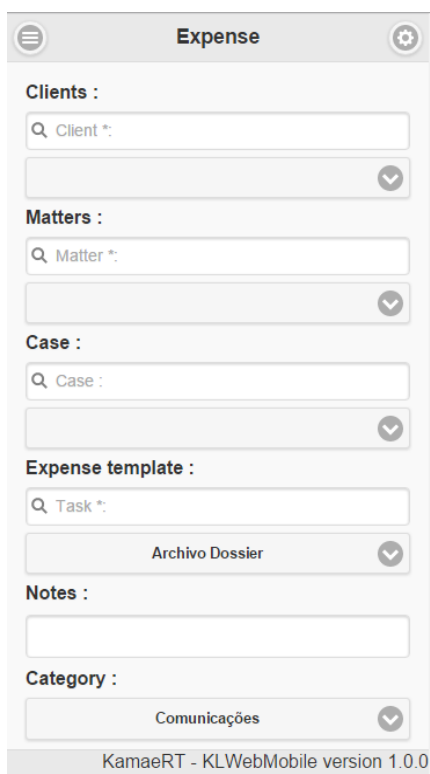
Figura 32 - Formulário de notas com contexto

Os campos identificadores de cliente, *dossier* e processo, constituem o componente de pesquisa de contexto, permitindo ao utilizador pesquisar quer por cliente, quer por *dossier* ou processo e ter os dados dos restantes identificadores adaptados à pesquisa efetuada.

A parte de gestão de notas é ainda complementada pela existência de uma funcionalidade de listagem das notas, que permite ao utilizador encontrar rapidamente a nota pretendida.

## Gestão de despesas

A parte de gestão de despesas consiste num formulário que permite a adição, edição, remoção e cálculo de despesas. Assim como no formulário de notas, este é também composto por uma pesquisa de contexto, recorrendo a uma pesquisa por cliente, *dossier* ou processo. Como se pode ver pelas figuras abaixo, é um formulário de grande dimensão, sendo composto por campos como o *template* da despesa e a categoria que definem o tipo e a categoria da despesa a ser lançada, ou a definição do representante e do tipo de responsabilidade para a despesa.



**Expense**

**Clients :**  
Q Client \*:  
▼

**Matters :**  
Q Matter \*:  
▼

**Case :**  
Q Case :  
▼

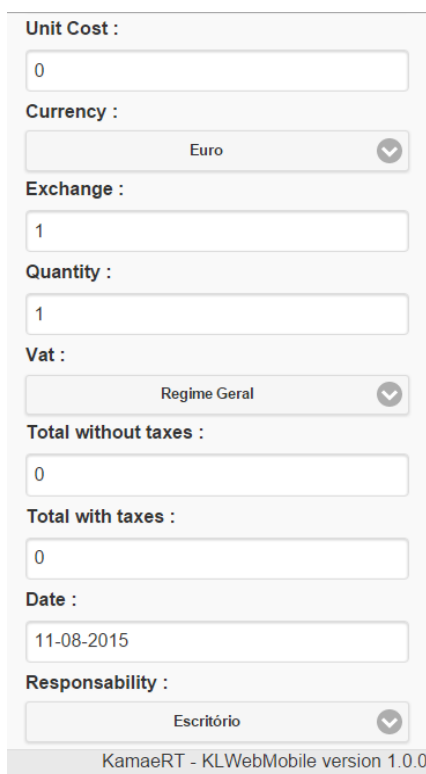
**Expense template :**  
Q Task \*:  
▼  
Archivo Dossier

**Notes :**  
▼

**Category :**  
▼  
Comunicações

KamaeRT - KLWebMobile version 1.0.0

Figura 33 - Formulário de despesa (parte 1)



**Unit Cost :**  
0

**Currency :**  
▼  
Euro

**Exchange :**  
1

**Quantity :**  
1

**Vat :**  
▼  
Regime Geral

**Total without taxes :**  
0

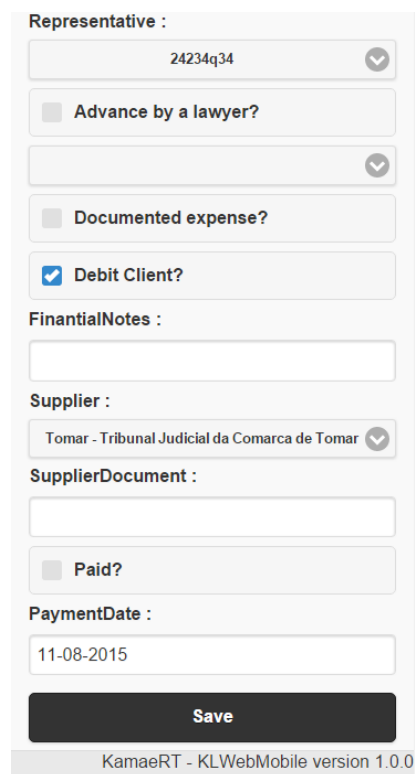
**Total with taxes :**  
0

**Date :**  
11-08-2015

**Responsability :**  
▼  
Escritório

KamaeRT - KLWebMobile version 1.0.0

FIGURA 34 - FORMULÁRIO DE DESPESA (PARTE 2)



**Representative :**  
▼  
24234q34

Advance by a lawyer?

▼

Documented expense?

Debit Client?

**FinancialNotes :**  
▼

**Supplier :**  
▼  
Tomar - Tribunal Judicial da Comarca de Tomar

**SupplierDocument :**  
▼

Paid?

**PaymentDate :**  
11-08-2015

**Save**

KamaeRT - KLWebMobile version 1.0.0

FIGURA 35 - FORMULÁRIO DE despesa (parte 3)

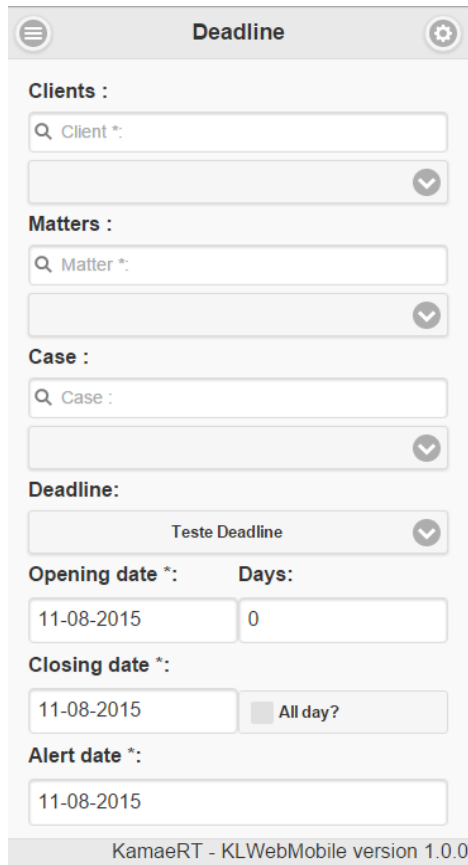
Existem também campos representativos de valores numéricos que se encontram diretamente ligados entre si, como é o caso da quantidade, do custo unitário e do IVA. Mudanças nos valores destes campos vão gerar alterações em mais dois campos numéricos denominados de total sem taxas e total com taxas. O valor total com taxas aplicadas pode ainda ser afetado, caso a despesa seja marcada como documentada, fazendo com que a despesa fique isenta de quaisquer taxas, tornando assim iguais os montantes com taxas e sem taxas.

Existe ainda a possibilidade de especificar se uma despesa já foi paga e a data do pagamento, a possibilidade de adicionar notas gerais e notas relacionadas com alguma questão financeira e também, definir um advogado e um fornecedor para a despesa.

A parte de despesas é também complementada por uma área de listagem onde é possível aceder a todas as despesas associadas a um utilizador, editá-las ou removê-las.

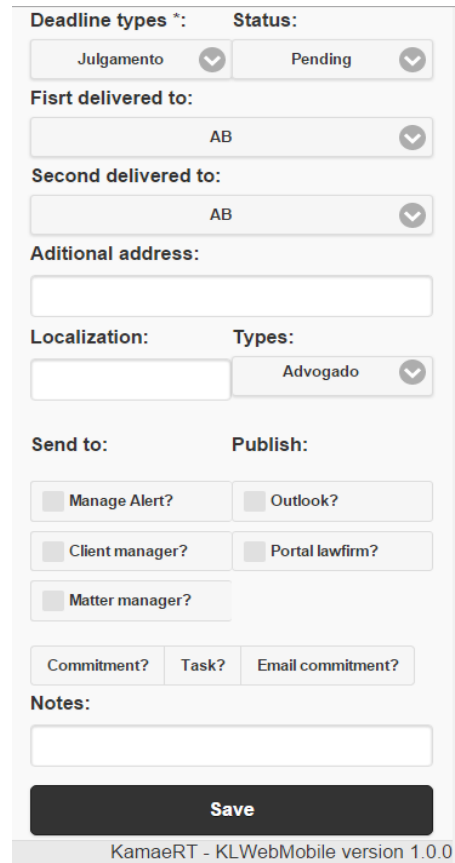
## Gestão de prazos

A gestão de prazos é baseada num formulário em que o objetivo é permitir o controlo dos prazos de um determinado evento para um *dossier*. Como se pode observar pelas figuras abaixo, é também composto pelos identificadores de cliente, *dossier* e processo, o que permite efetuar uma pesquisa por qualquer um deles, de forma a associar contexto ao evento.



The screenshot shows a mobile application interface for a 'Deadline' form. It features several search fields: 'Client \*', 'Matter \*', 'Case', and 'Deadline'. Below these are date fields for 'Opening date \*' (11-08-2015) and 'Days' (0), 'Closing date \*' (11-08-2015) with an 'All day?' checkbox, and 'Alert date \*' (11-08-2015). The interface includes a hamburger menu icon, a settings gear icon, and a footer with the text 'KamaeRT - KLWebMobile version 1.0.0'.

Figura 36 - Formulário de prazos (parte 1)



The screenshot shows the second part of the 'Deadline' form. It includes dropdown menus for 'Deadline types \*' (set to 'Julgamento'), 'Status' (set to 'Pending'), 'First delivered to' (set to 'AB'), and 'Second delivered to' (set to 'AB'). There is an 'Additional address' field, a 'Localization' field, and a 'Types' dropdown (set to 'Advogado'). Below these are sections for 'Send to' and 'Publish' with checkboxes for 'Manage Alert?', 'Outlook?', 'Client manager?', 'Portal lawfirm?', and 'Matter manager?'. There are also buttons for 'Commitment?', 'Task?', and 'Email commitment?'. A 'Notes' field is at the bottom, followed by a prominent 'Save' button. The footer text is 'KamaeRT - KLWebMobile version 1.0.0'.

Figura 37 - Formulário de prazos (parte 2)

Para definir o tipo de evento em causa, existe um campo denominado prazo que contém os vários tipos de eventos disponíveis para seleção. Existem também campos de datas, com o objetivo de calendarizar a data de início e fim, bem como a data de alerta, que serão depois usadas para calcular o período de duração do evento.

Pode ainda ser definido o tipo de prazo e o seu estado atual, a localização do evento, a quem deve ser entregue e a forma de publicação, o formato a usar (compromisso, tarefa ou compromisso por *email*) e alguma nota importante. É possível consultar todos os prazos existentes recorrendo à área de listagem, através da qual é possível pesquisar, eliminar e proceder a edição de um prazo.

## Gestão de processos

A parte de gestão de processos é composta por um *wizard* que permite a criação, edição e remoção de processos. Existem atualmente dois tipos de processos possíveis de criar: os processos simples e os processos judiciais, onde em termos práticos a diferença se reflete numa área extra que contém campos relacionados com a parte judicial do processo. A diferenciação entre estes dois tipos de processo ocorre quando o utilizador escolhe a opção de qualidade de processo em que, caso a sua escolha seja do tipo “judicial”. O tipo do processo é mudado de simples para judicial.

No seu tipo mais completo, o tipo judicial, o *wizard* é composto por 3 partes distintas, sendo elas:

1. **Área de informação geral** – Contém informações gerais comuns a ambos os processos simples e judiciais. É constituída por uma versão modificada da pesquisa de contexto, onde apenas o cliente e o *dossier* estão disponíveis, e também por campos como a referência interna, o tipo, o departamento e a qualidade.
2. **Área de informação judicial** – Contém informações relativas à parte judicial e à parte de sentença de um processo, como o número do processo de tribunal, os tribunais onde decorre o processo, as datas de entrada, distribuição e de fim, o resultado da sentença, entre outros.
3. **Área de confirmação de dados** – Contém informações sobre os valores inseridos para os vários campos do *wizard*, de forma a permitir ao utilizador conferir rapidamente se os valores mostrados estão ou não corretos, e corrigi-los caso seja necessário.

Um processo simples possui apenas a área de informação geral, exatamente com os mesmos campos existentes no processo judicial, e a área de confirmação de dados. Durante a edição de um processo judicial, caso se queira torná-lo simples, mudando o tipo de qualidade do processo, e se confirmem as alterações dos dados do processo, os dados da parte judicial serão à mesma mantidos para caso o utilizador queira reverter as modificações. Existe a possibilidade de listar todos processos a que o utilizador tem acesso, de forma a poder pesquisar, editar ou remover algum processo em específico. Ilustrações das várias áreas referidas encontram-se disponíveis no **anexo A**.



## 4.9.2 Área de Clientes

A área de clientes é composta pelo cenário de gestão de cliente e este consiste num *wizard* que permite adição, edição e remoção de clientes. O *wizard* pode ser de dois tipos: o tipo completo e o tipo rápido. O primeiro permite a criação de 2 modelos distintos de clientes onde cada um tem certas áreas adicionais específicas. No total, o tipo completo é composto por 8 áreas distintas, enquanto o tipo rápido é apenas constituído por uma única área, composta por campos de outras áreas.

O *wizard* de clientes é definido pelas seguintes áreas:

- **Área de informações do cliente** – Área relativa a informações gerais do cliente, como o nome, país, morada, entre outros.
- **Área de contactos** – Área composta por campos associados aos contactos de um cliente, como contactos telefónicos, *emails*, Skype, endereço de Facebook e LinkedIn.
- **Área de contactos adicionais** – Área apenas disponível para clientes coletivos onde é possível a criação de clientes associados ao cliente coletivo em criação, a qual contém campos como o nome, *email*, endereço telefónico e *website*.
- **Área financeira** – Área responsável pela existência de campos relacionados com a parte financeira, tais como crédito, método de pagamento e moeda
- **Área de dados adicionais singular** – Área que permite que sejam inseridos dados adicionais sobre um cliente singular, como o género do cliente, data de nascimento e número de passaporte.
- **Área de dados adicionais coletivo** – Área que permite que sejam inseridos dados adicionais sobre um cliente coletivo, como nome comercial, objetivo social e capital.
- **Área de marketing** – Área constituída por campos identificadores de angariadores e gestores.
- **Área de confirmação** – Área constituída por todos os valores definidos para os vários campos do *wizard*, permitindo ao utilizador uma visualização geral de todos eles e verificar se algum deles não está correto.
- **Área de criação rápida** – Área usada apenas no método de criação rápida de cliente. Possui campos das várias áreas do *wizard* completo como o nome e país da **área de informação do cliente**, o campo de gestor associado ao cliente proveniente da **área de marketing**, os campos de *email* e telefone principal da **área de contactos** e o género do cliente, proveniente da **área de dados adicionais singular**.

Para o tipo de *wizard* completo que segue o modelo de cliente singular, a ordem das áreas é a seguinte:

- **Área de informações do cliente** → **Área de contactos** → **Área financeira** → **Área de dados adicionais singular** → **Área de marketing** → **Área de confirmação**

Para o tipo de *wizard* completo que segue o modelo de cliente coletivo, a ordem das áreas é, por sua vez:

- **Área de informações do cliente** → **Área de contactos** → **Área de contactos adicionais** → **Área financeira** → **Área de dados adicionais coletivo** → **Área de marketing** → **Área de confirmação**

A forma de criação rápida é apenas constituída pela área de criação rápida descrita acima.

A área de gestão de clientes conta ainda com uma listagem de todos os clientes disponíveis para o utilizador, de forma que estes possam ser encontrados, editados e eliminados facilmente. Ilustrações das várias áreas referidas encontram-se disponíveis no **anexo A**.

### 4.9.3 Área de Opções

A área de opções é definida pelo conjunto de funcionalidades que afetam o funcionamento de toda a aplicação. Estas funcionalidades estão divididas em 5 áreas, como se pode ver na figura 38, onde cada uma delas contém um conjunto de opções que afetam uma área específica da aplicação. As opções de clientes estão diretamente ligadas à parte de gestão de clientes e afetam o funcionamento da mesma. O mesmo acontece com as opções de *dossier* (“Matters”) que estão ligadas à gestão de *dossiers*, e com as opções de processos (“Cases”) que estão ligadas à gestão de processos. As restantes duas consistem nas opções de lançamentos e opções de escritório. As opções de lançamento, afetam o funcionamento do lançamento de tarefas de *time-sheet* e lançamento de despesas, sendo a maioria focadas no lançamento de tarefas de *time-sheet*. Por fim, as opções de escritório têm como objetivo primário a atribuição de termos personalizados para palavras-chave, como *dossiers* ou processos, para que os termos presentes na aplicação vão ao encontro dos termos usados na organização.

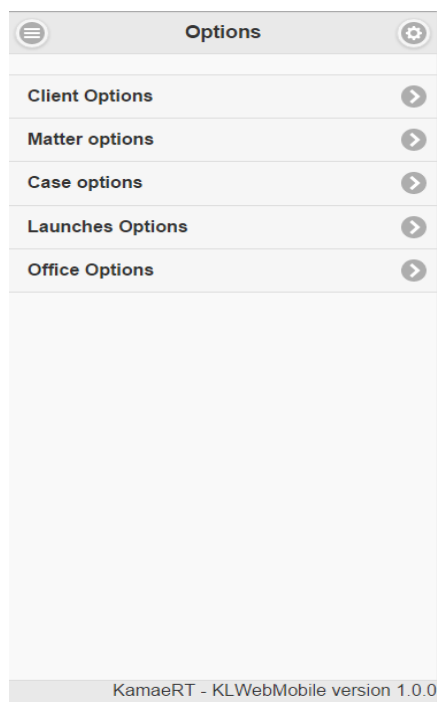
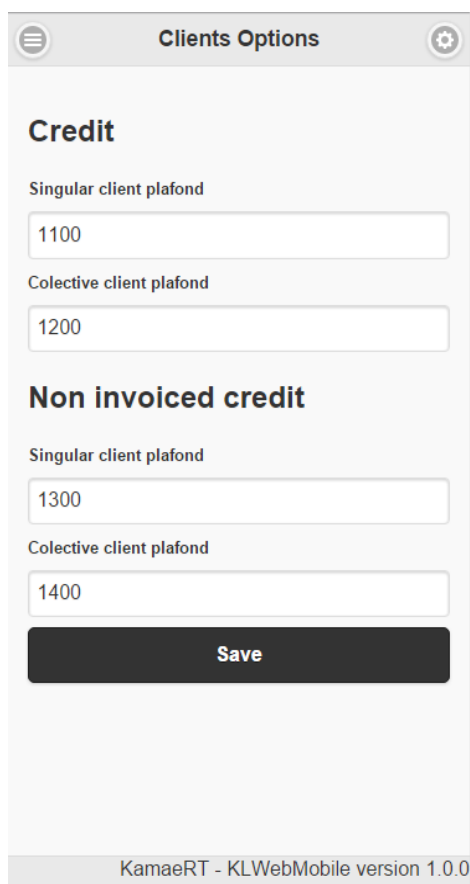


Figura 38 - Área de opções

## Opções de cliente

As opções de cliente consistem em campos relativos ao crédito oferecido a cada cliente. As opções estão divididas em crédito faturado e crédito não faturado, como se pode ver na figura abaixo, sendo que os montantes para cada um dos tipos de crédito pode variar consoante o tipo de cliente seja singular ou coletivo.



**Clients Options**

**Credit**

Singular client plafond  
1100

Colective client plafond  
1200

**Non invoiced credit**

Singular client plafond  
1300

Colective client plafond  
1400

**Save**

KamaeRT - KLWebMobile version 1.0.0

Figura 39 - Formulário de opções de cliente

Ao serem guardadas, as alterações produzem efeito imediato para o utilizador em causa, quer na própria aplicação, quer na KamaeLei Web e na KamaeLei Desktop.

## Opções de *dossiers*

As opções de *dossiers* são as que mais funcionalidades afetam na aplicação. Estão diretamente ligadas à parte de gestão de *dossier*, que é composta por um *wizard*. Estas opções estão divididas em partes distintas: a parte de opções gerais, a parte das referências internas, a parte financeira e a parte de alertas, como indicado nas figuras abaixo.

Figura 40 - Formulário de opções de *dossier* (parte 1)

Figura 41 - Formulário de opções de *dossier* (parte 2)

Figura 42 - Formulário de opções de *dossier* (parte 3)

A área geral é composta, no total, por cinco opções, sendo elas:

- **Pedir sempre a seleção de um cliente quando se cria um *dossier*** – Esta opção existe mas, atualmente, para a correta criação de um *dossier*, é sempre necessária a seleção de cliente.
- **Ao criar um *dossier*, adicionar toda a equipa** – Permite que por predefinição toda a equipa do utilizador já esteja selecionada quando se inicia o *wizard*.
- **Apenas os membros de equipa podem aceder aos seus *dossiers* e processos** – Restrição aplicada ao acesso a *dossiers* e processos.
- **Na criação de um novo *dossier*, desbloquear referência interna se possível** – Caso o utilizador em causa tenha permissões, a referência interna pode ser desbloqueada no próprio *wizard*.
- **Escolher um departamento por predefinição para o *wizard***.

A área de referência interna é apenas composta por um campo que determina a sintaxe escolhida para a referência interna do *dossier*.

Exemplos de sintaxes usadas para referências internas:

- **<FolderNr>-<IntRef>**
- **<FolderNr>.<Year>-<IntRef>**
- **<Year>.<AnnualNumberNrDigit>**
- **<Cat><NrOnYearAndCat2Digit>/<Year>**

Os exemplos de sintaxes definidos em cima mostram dois tipos distintos de referências internas: as que requerem que um valor manual seja inserido, que correspondem àquelas que contêm “-<IntRef>” na sua sintaxe, e as que não o têm e que são automaticamente geradas sem qualquer tipo de inserção manual.

Quando aplicados à parte de gestão de *dossiers*, alguns dos campos como o <Year> ou o <FolderNr> podem, dependendo da referência, ser substituídos por valores. O campo <IntRef> é também substituído por uma caixa de texto que o representa, que apenas pode ser editada quando o tipo de referência interna assim o permitir.

As duas últimas áreas consistem nas áreas de opções financeiras e de alertas. Estas áreas contêm um conjunto de campos representantes de ambas as áreas presentes no wizard de *dossiers* em que o objetivo é a definição de valores predefinidos para esses campos, como o orçamento planeado e o número de horas planeadas para a parte financeira, e campos como a rentabilidade e horas a cobrar para a parte de alertas.

## Opções de processos

As opções de processos é semelhante às opções de *dossier*, mas possui uma dimensão mais reduzida. Está dividida em duas áreas, a área geral e a área de referência interna, como demonstra a figura abaixo.

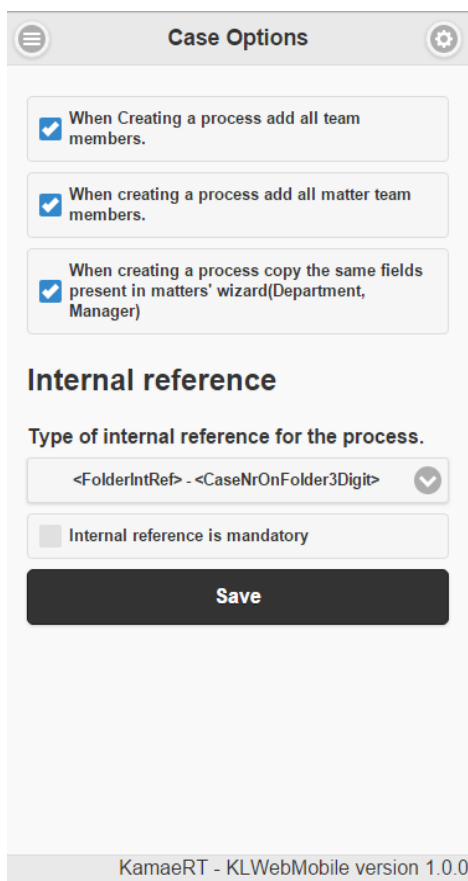


Figura 43 - Formulário de opções de processo

A área geral é composta por 3 campos, sendo eles:

- Ao criar um processo adicionar a equipa toda.
- Ao criar um processo adicionar toda a equipa do *dossier*.
- Ao criar um processo copiar os campos “Departamento” e “Gestor” do *dossier*.

A área de referência interna, à semelhança das opções de *dossier*, é composta por um campo que permite a escolha de uma sintaxe para a referência interna. A referência interna escolhida será então aplicada por predefinição sempre que for criado um novo processo. Nesta parte existe um campo que permite ao utilizador tornar a referência interna obrigatória ou não obrigatória. No entanto, atualmente a referência interna do processo é sempre obrigatória.

## Opções de lançamentos

As opções de lançamentos incidem sobre a parte de gestão de despesas e gestão de *time-sheet*, sendo que a maior parte destas está associada à gestão de *time-sheet*.

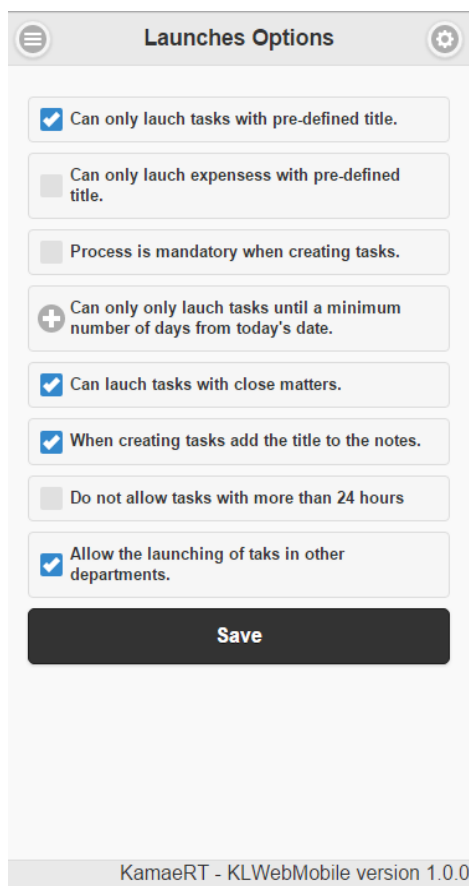


Figura 44 - Formulário de opções de lançamentos

As opções de lançamento existentes são:

- **Só é possível lançar tarefas com título predefinido** – O título das tarefas tem obrigatoriamente de ser especificado recorrendo apenas às opções disponíveis no campo. Caso esta opção não esteja selecionada, pode ser dado um título qualquer. No formulário de *time-sheet* corresponde ao campo “Tarefas”.
- **Só é possível lançar despesas com título predefinido** – O título das despesas tem obrigatoriamente de ser especificado recorrendo apenas às opções disponíveis no campo. Caso esta opção não esteja selecionada pode ser dado um título qualquer. No formulário de despesas o campo correspondente ao título é o campo de “Template de despesa”.
- **O processo é obrigatório quando se criam tarefas** – O campo relativo ao processo na pesquisa de contexto tem de ser sempre especificado caso esta opção esteja ativa.

- **Só podem ser lançadas tarefas com “X” número de dias antes da data atual** – Define o número máximo de dias para lançar tarefas anteriores à data atual.
- **Podem ser lançadas tarefas em *dossiers* fechados** – Torna possível o lançamento de tarefas em *dossiers* fechados, algo que não é possível por predefinição.
- **Adicionar o título da tarefas às notas** – Faz com que o título da tarefa seja adicionado às notas seguindo a lógica “ <Título> – <Texto>”.
- **Não permitir tarefas com mais de 24 horas** – Cada utilizador pode apenas lançar tarefas em que a soma total dos tempos não exceda 24 horas por dia. Quando esta opção está ativa esse limite é removido.
- **Permitir o lançamento de tarefas noutra departamento** – Quando ativa, esta opção acrescenta um novo campo ao formulário de tarefas de *time-sheet* em que é possível selecionar o departamento onde será feito o lançamento da tarefa. Quando não está selecionada, o departamento ativo é o departamento onde o utilizador está inserido.



## Opções de escritório

As opções de escritório estão divididas em duas partes distintas: a parte de escritório e a parte dos termos. No entanto, apenas a parte dos termos afeta o funcionamento deste projeto, pois a parte onde estão ligadas a funcionalidades sai fora do âmbito desta aplicação e do estágio. De qualquer forma, é possível alterar os dados relativos à parte de escritório que, embora não afete em nada esta aplicação, afeta o funcionamento da versão *web* e *desktop*, onde as funcionalidades estão implementadas.

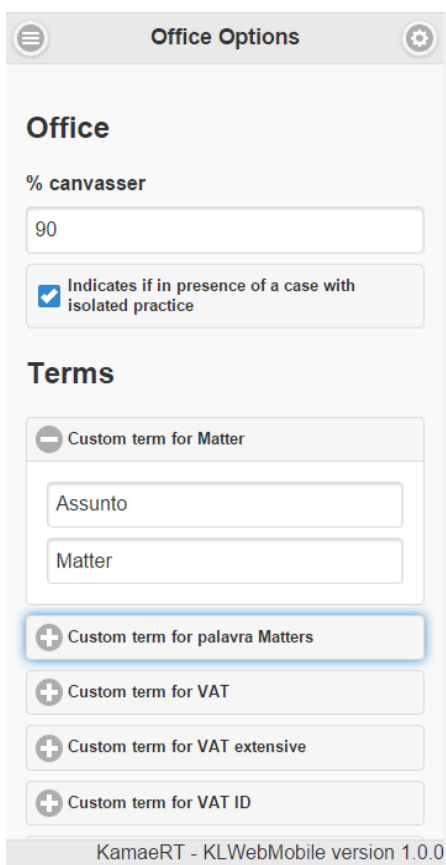


Figura 45 - Formulário de opções de escritório (parte 1)

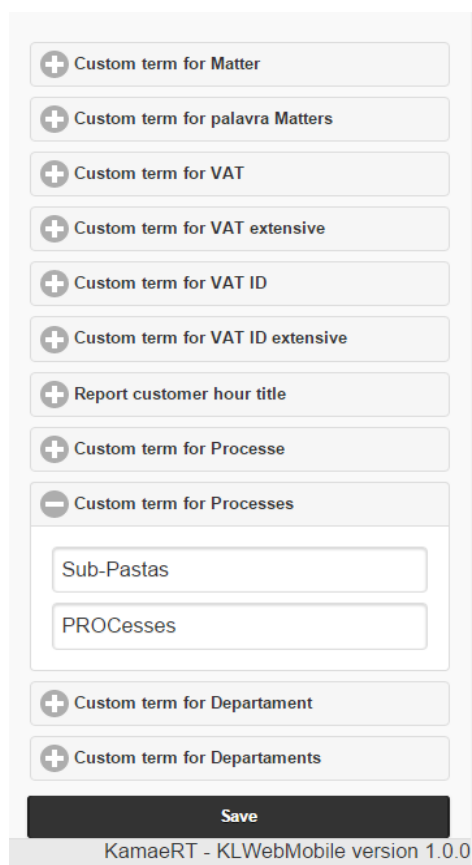


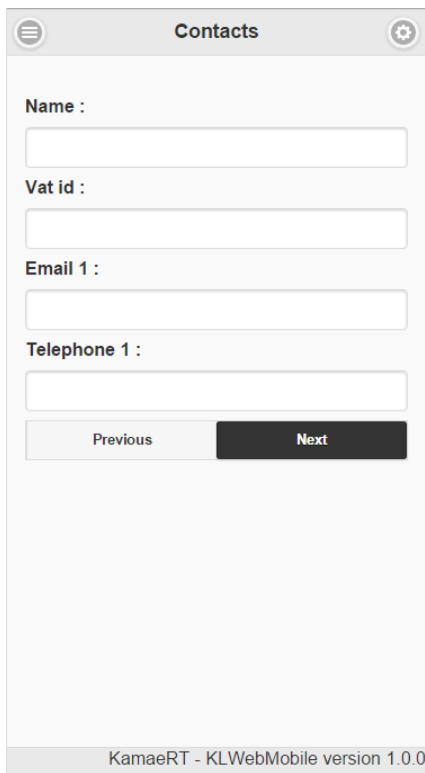
Figura 46 - Formulário de opções de escritório (parte 2)

A parte das opções de termos, tem como finalidade a substituição de um conjunto de palavras-chave, como *dossiers* (Matters) ou processos (Processes), tal como é mostrado na figura anterior, e a atribuição de termos personalizados para permitir que uma organização use os termos a que está habituada. Nesta área é possível definir termos nas várias línguas suportadas pela aplicação, os quais, ao serem definidos, são replicados por toda a aplicação.

Esta funcionalidade é de extrema importância devido à existência de inúmeros termos para designar a mesma coisa. Por exemplo, um “*Dossier*” pode ser chamado de “*Pasta*” por uma organização, “*Assunto*” por outra ou até mesmo “*Projeto*”. Devido a esta grande variedade de nomenclaturas, esta funcionalidade torna-se indispensável para a aplicação.

#### 4.9.4 Área de Contactos

A área de cenários engloba o cenário de gestão de contactos. Esta área consiste num formulário que tem uma versão completa e uma versão rápida, como se pode observar nas figuras seguintes.



Contacts

Name :

Vat id :

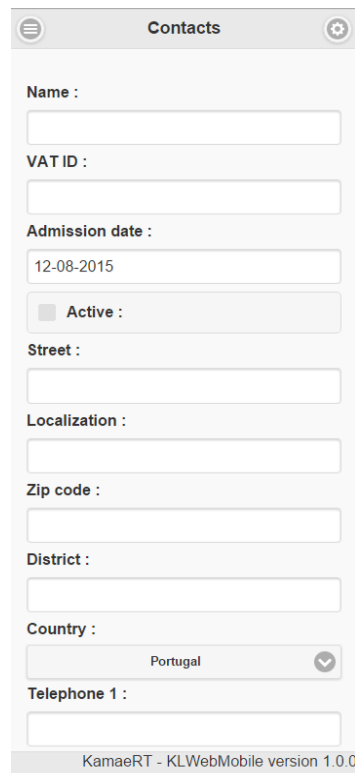
Email 1 :

Telephone 1 :

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 47 - Formulário de contacto rápido



Contacts

Name :

VAT ID :

Admission date :  
12-08-2015

Active :

Street :

Localization :

Zip code :

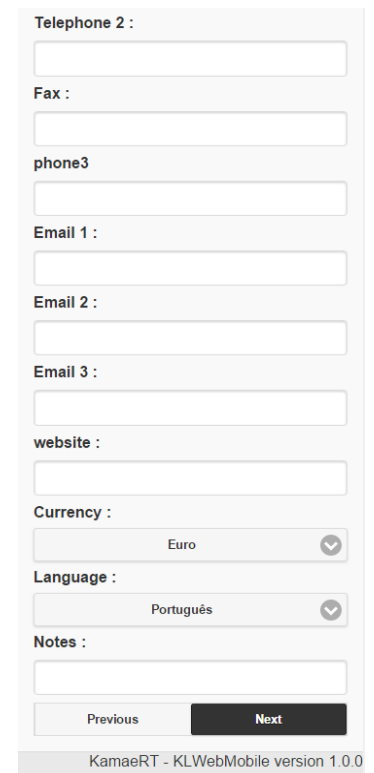
District :

Country :  
Portugal

Telephone 1 :

KamaeRT - KLWebMobile version 1.0.0

Figura 48 - Formulário de contacto completo (parte 1)



Telephone 2 :

Fax :

phone3

Email 1 :

Email 2 :

Email 3 :

website :

Currency :  
Euro

Language :  
Português

Notes :

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 49 - Formulário de contacto completo (parte 2)

A versão simples do formulário possui apenas os campos essenciais para um contacto, tais como o nome, NIF, *email* principal e telefone principal. A versão completa é composta por todos campos descritos e mais um conjunto de outros, de forma a completar os dados do cliente, tais como a morada, data de admissão, língua, entre outros. Esta área suporta as funcionalidades de inserção, edição e remoção, assim como uma listagem, através da qual é possível aceder a estas funcionalidades.

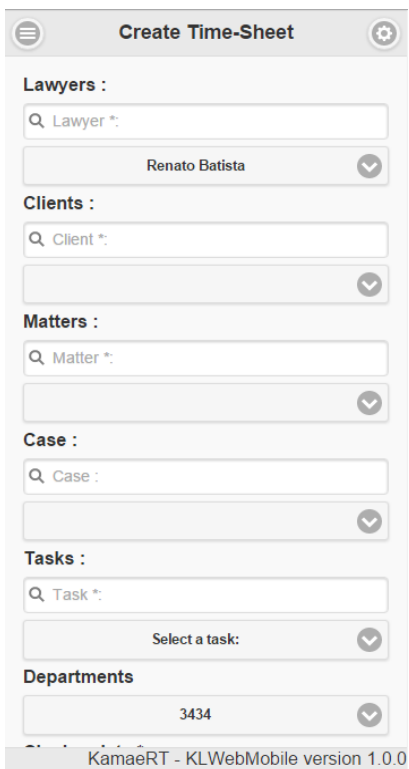
## 4.9.5 Área de Login

A área de *login* é caracterizada por um formulário que permite ao utilizador aceder às funcionalidades da aplicação. É composta por quatro campos: o *username*, a *password*, o domínio e o *captcha*.

Quando estes dados são submetidos ao servidor, é feita em primeiro lugar uma validação do mecanismo de *captcha* e só depois a validação dos dados de *login*, para que, caso esta validação falhe, evitar o processamento desnecessário pela aplicação servidor. Quando o *login* não é bem sucedido, é devolvida ao utilizador uma mensagem a descrever a razão da falha, quer seja nas credenciais, quer seja no mecanismo de *captcha*. Quando o *login* é efetuado com sucesso, é então apresentado ao utilizador o menu principal da aplicação.

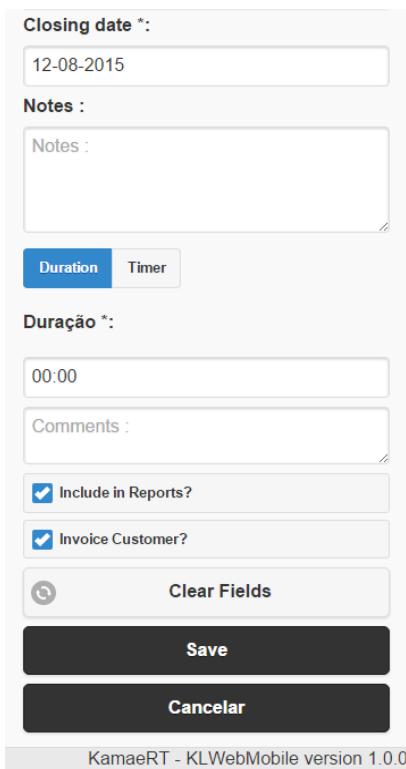
## 4.9.6 Área de Time-Sheet

A área de *time-sheet* é composta pelo cenário de gestão de *time-sheet*, em que o objetivo é conseguir uma melhor gestão de tempo por parte dos utilizadores, através de um controlo mais otimizado das suas tarefas diárias. Esta área consiste num formulário que permite aos utilizadores a criação e edição de tarefas. Este é formado por um componente de pesquisa de contexto que permite atribuir contexto à tarefa a ser criada/editada, o departamento onde a tarefa deve ser lançada, um título para a tarefa, notas, data de fim e a duração da tarefa, que pode ser definida em horas e minutos ou através da especificação da hora de início e hora de fim, tal como se pode observar nas figuras seguintes.



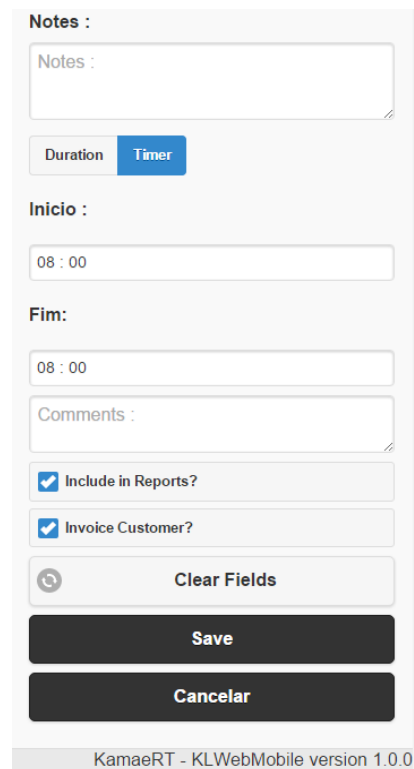
KamaeRT - KLWebMobile version 1.0.0

Figura 50 - Formulário time-sheet (parte 1)



KamaeRT - KLWebMobile version 1.0.0

Figura 51 - Formulário time-sheet (parte 2)



KamaeRT - KLWebMobile version 1.0.0

Figura 52 - Formulário time-sheet (parte 3)

Tal como visto anteriormente, esta área é bastante afetada pela existência de opções. Além disso, possui ainda algumas particularidades, como o facto de alguns títulos de tarefas executarem funcionalidades adicionais, como a possibilidade de adição de honorários a uma tarefa que são apresentados ao utilizador sobre a forma de *popup* e a predefinição de um tempo base para a realização de uma tarefa dependendo do título selecionado para a mesma. Os campos obrigatórios para a adição de uma tarefa são o cliente, *dossier*, título da tarefa e duração superior a 0 segundos. Estes podem sofrer alterações dependendo das opções selecionadas.

As tarefas lançadas podem ser encontradas, recorrendo à área de listagem de *time-sheet*, onde é possível pesquisar, selecionar para edição e eliminar uma tarefa.

## 5 Testes

Para este projeto não foi inicialmente definido um plano de testes. No entanto, à medida que cada funcionalidade era implementada, era testada rigorosamente para que depois fossem resolvidas quaisquer questões que pudessem surgir.

De facto, os testes foram realizados pelo estagiário à medida que cada funcionalidade era implementada e também por membros da equipa de testes quando uma nova versão da aplicação era disponibilizada para teste. Quaisquer *bugs* identificados eram lançados no Team Foundation Server e associados à pessoa encarregue de os resolver.

Para avaliar a gravidade de um *bug*, foi definido um *ranking* que permite estimar a gravidade do problema. Estes valores são definidos numa escala de 1 a 5, onde 1 representa um erro catastrófico que põe em causa o correto funcionamento da aplicação - sendo urgente a sua correção, e o valor 5 representa um erro menor com uma prioridade de correção muito baixa. Esta escala foi criada e definida pela empresa de acolhimento.

A escala definida para classificar os erros encontrados é a seguinte:

- **Blocker (1)** - Os erros desta tipologia são graves para o sistema e comprometem o funcionamento e a integridade da própria aplicação, sendo de extrema importância a sua correção atempada para que não comprometa outras funcionalidades.
- **Critical (2)** - As funcionalidades e características que apresentem este tipo de erro permitem o uso da aplicação mas com grandes limitações, podendo comprometer na íntegra determinadas funcionalidades, devendo a sua correção ser priorizada e concretizada antes da entrega do produto.
- **Major (3)** - Funcionalidade ou característica sem o qual o sistema funciona, mas de forma não satisfatória. Estes são requisitos importantes que devem ser cumpridos, porém, na sua ausência ou mau desempenho, o sistema não está completamente comprometido.
- **Minor (4)** - Corresponde ao caso de serem encontradas algumas inconformidades de menor importância na aplicação como um requisito que não compromete as funcionalidades básicas do sistema (o sistema pode funcionar de forma satisfatória sem o seu correto funcionamento). Os erros com esta atribuição podem ser adiados para versões posteriores da aplicação, caso não haja tempo útil disponível para serem retificados.
- **Trivial (5)** - Este tipo está normalmente associado a incoerências encontradas na interface do utilizador, como cores incorretas, mensagens mal formuladas, imagens repetidas, entre outros. Este tipo de erro não é relevante e em nada compromete o funcionamento e compreensão da aplicação, tendo uma prioridade de correção baixa.

Após a sua resolução, estes eram marcados para teste pela pessoa que os resolveu para voltarem a ser testados por quem os identificou, que os daria como resolvidos caso o problema já não existisse ou como ativos caso o problema ainda persistisse.

Ao longo do desenvolvimento foram reportados um total de 57 erros, em que a sua gravidade variava dependendo da forma como influenciava a aplicação. A listagem completa dos *bugs* reportados encontra-se disponível no **Anexo C**.

## 5.1 Dispositivos, sistemas e browsers testados

Os testes efetuados foram realizados nos principais sistemas e *browsers* de acordo com a tabela seguinte.

Tabela 14 - Sistemas e Browsers testados

SISTEMA OPERATIVO	DISPOSITIVO	VERSÃO SO	BROWSERS
<b>Android</b>	Samsung Galaxy S4	4.4.2	Google Chrome
	Samsung Galaxy Note 2014	5.0 CyanogenMod	Mozilla Firefox
	Samsung Galaxy S3	5.0.1 CyanogenMod	Stock Browser (CyanogenMod e Paranoid)
	LG G2	5.0.1 CyanogenMod	Stock Browser (Samsung)
	LG G2	5.1.1 Paranoid	Stock Browser (Samsung)
<b>iOS</b>	iPhone 4	7.1.2	Safari
	iPhone 4s	8.x	Google Chrome
	iPhone 5s	8.x	
<b>Windows Phone</b>	Nokia Lumia 710	7.5	Internet Explorer

Ao testar a aplicação notou-se alguma variância de performance dos *browsers* na execução da aplicação. Estas diferenças eram particularmente visíveis quando se recorria à utilização de componentes ou operações específicas do jQuery e jQuery *Mobile*. Como tal, foram criadas tabelas que permitiram analisar estas diferenças. A análise foi baseada em 3 aspetos, nomeadamente:

- **Renderização de elementos** – Consistiu em avaliar a capacidade dos *browsers* de renderizarem os elementos dispostos nas várias páginas, quando estas eram abertas pela primeira vez, quando eram feitas mudanças na forma de visualização (*portrait mode* e *landscape mode*), quando a página do *browser* era colocada em *background* e reaberta, e quando o ecrã do dispositivo se desligava e voltava a ser ligado.
- **Fluidez nas animações** - Consistiu em avaliar se a utilização de animações decorria sem problemas, como lentidão na execução, efeitos *flicker* no ecrã dos dispositivos, ou encerramentos inesperados da aplicação.
- **Fluidez na utilização de componentes jQuery** – Consistiu em avaliar a forma como os vários *browsers* lidavam com a utilização de componentes de data, seleção de tempo, *captcha* e modificações no aspeto visual, que recorrem ao jQuery ou jQuery *Mobile*.







Para demonstrar a performance de cada *browser* em cada dispositivo foram utilizadas várias cores. A cada cor está associado um nível de performance, como demonstrado na tabela 15.

Tabela 15 - Nível de performance

Ótimo	Bom	Razoável	Mau	Não funcional	Não testado
-------	-----	----------	-----	---------------	-------------

De forma a simplificar a identificação de cada *browser*, foram usados os seus ícones correspondentes, como ilustrado na tabela 16.

Tabela 16 - Identificação dos browsers

Google Chrome	Mozilla Firefox	Stock Browser (CyanogenMod e Paranoid)	Stock Browser (Samsung)	Safari	Internet Explorer
					

De seguida encontram-se três tabelas que demonstram a performance dos vários *browsers* relativamente aos aspetos anteriormente referidos.

Tabela 17 - Renderização dos elementos

















































Critério	Dispositivos							
	Samsung Galaxy S4	Samsung Galaxy Note 2014	Samsung Galaxy S3	LG G2	iPhone 4	iPhone 4s	iPhone 5s	Nokia 710
Renderização dos elementos								
								
								
								
								
								



Tabela 18 - Fluidez nas animações




























































































Critério	Dispositivos							
	Samsung Galaxy S4	Samsung Galaxy Note 2014	Samsung Galaxy S3	LG G2	iPhone 4	iPhone 4s	iPhone 5s	Nokia 710
Fluidez nas animações								
								
								
								
								
								

Tabela 19 - Fluidez na utilização de componentes jQuery

Critério	Dispositivos							
	Samsung Galaxy S4	Samsung Galaxy Note 2014	Samsung Galaxy S3	LG G2	iPhone 4	iPhone 4s	iPhone 5s	Nokia 710
Fluidez na utilização de componentes jQuery								
								
								
								
								
								



Ao analisar as tabelas anteriores, é possível concluir que dentro dos três aspetos avaliados, apenas no da utilização de componentes jQuery e jQuery *Mobile*, se obteve uma performance aceitável entre os vários *browsers* nos vários dispositivos. Os restantes dois tiveram a sua performance gravemente afetada nos dispositivos com menores capacidades de processamento e memória.

Desta forma, foram removidas as funcionalidades de *caching* e animações. As funcionalidades de *caching* foram removidas para otimizar a renderização dos elementos em páginas, visto que em dispositivos mais antigos, essa não era feita de forma eficiente e fazia que elementos de várias vistas se sobrepusessem. As animações foram removidas para tornar a transição mais fluida em dispositivos mais antigos, visto que o seu funcionamento nesses dispositivos não era o mais correto.

## 5.2 Testes Unitários

Para além de testes realizados nos vários *browsers*, foram também desenvolvidos testes unitários. O grande propósito dos testes unitários nesta aplicação foi auxiliar o bom funcionamento da mesma na presença de mudanças na aplicação *web* para *desktop*.

Durante o decorrer do estágio, a aplicação *web desktop* foi sofrendo várias alterações que tinham impacto direto sobre esta aplicação. Embora houvesse um esforço por parte da equipa de desenvolvimento da aplicação *web desktop* em relatar as mudanças efetuadas à mesma, existiam sempre informações que eram esquecidas, o que fazia com essas modificações não fossem replicadas na aplicação *web mobile*, resultando num funcionamento incorreto. Para resolver este problema recorrente, foi criado um conjunto de testes unitários para as principais funções provenientes da aplicação *web desktop* de forma a conseguir identificar essas mudanças mais facilmente.

Foram criados, no total, 66 testes unitários, que abrangiam desde funcionalidades de listagem de dados, até funções de inserção, edição ou remoção. Desta forma, foi possível obter uma poupança de tempo na identificação de erros relacionados com alterações na versão *web desktop*. De seguida encontra-se um exemplo de um teste unitário.

```
[TestMethod]
public void insertDeadline_Test()
{
    Deadline deadline = new Deadline();
    deadline.IDEntityCustomer = 926; deadline.IDKLFolder = 51; deadline.IDKLCASE = 101860;
    deadline.IDKLScheduleItem = 6; deadline.IDEntityAssignedTo1 = 29448;
    deadline.IDEntityAssignedTo2 = 29448; deadline.IDType = 1;
    DateTime dateTime;
    if (DateTime.TryParseExact(DateTime.Now.ToString("dd-MM-yyyy"), "dd-MM-yyyy",
    System.Globalization.CultureInfo.InvariantCulture, System.Globalization.DateTimeStyles.None, out
    dateTime))
        deadline.Date = dateTime;

    if (DateTime.TryParseExact(DateTime.Now.ToString("dd-MM-yyyy"), "dd-MM-yyyy",
    System.Globalization.CultureInfo.InvariantCulture, System.Globalization.DateTimeStyles.None, out
    dateTime))
        deadline.DateEnd = dateTime;

    if (DateTime.TryParseExact(DateTime.Now.ToString("dd-MM-yyyy"), "dd-MM-yyyy",
    System.Globalization.CultureInfo.InvariantCulture, System.Globalization.DateTimeStyles.None, out
    dateTime))
        deadline.ReminderDate = dateTime;

    int id = DeadlineBusiness.InsertDeadline(deadline);
    if(id!=null)
        Assert.IsTrue(true);
    else
        Assert.IsTrue(false);
}
```

Figura 53 - Exemplo de teste unitário

## 6 Conclusões

A aplicação desenvolvida no âmbito do presente estágio teve como objetivo complementar as soluções *web* e *desktop* do sistema de gestão jurídica KamaeLei, oferecendo uma solução focada no mercado emergente que são os dispositivos móveis. Esta aplicação vem disponibilizar as principais funcionalidades oferecidas por ambas as soluções e adaptá-las a dispositivos de menor dimensões. Nomeadamente, foram implementadas os módulos de gestão de clientes, gestão de contactos, gestão de *dossiers*, gestão de processos, gestão de prazos, gestão de despesas, gestão de notas, gestão de *time-sheet* e configuração de opções.

Desta forma, é possível a um utilizador aceder às funcionalidades presentes nas soluções *web* e *desktop* em qualquer lado recorrendo apenas ao seu dispositivo móvel.

O que faz esta aplicação destacar-se das aplicações móveis nativas de iOS, Android e Windows Phone, é a sua maior semelhança com a versão *web desktop* e o facto de ser bastante mais completa nas funcionalidades que possui. Por outro lado, ao ser uma aplicação *web* separada e focada apenas em dispositivos móveis, permitiu que o seu *design* fosse muito semelhante ao de uma aplicação nativa. Isto foi possível, através da utilização de tecnologias *web* focadas especialmente em dispositivos móveis, como o *jQuery Mobile*, que oferece um conjunto de propriedades e funcionalidades que reorganizam o conteúdo e ajudam a torna-lo próprio para ser visualizado neste tipo de dispositivos. Embora não tenha sido possível tirar partido de tudo o que a tecnologia tem para oferecer devido às restrições de memória e processamento mínimos dos dispositivos onde tem de correr, foi decerto uma mais-valia no seu desenvolvimento.

No início do estágio foi sentida uma certa dificuldade em compreender a forma como seriam implementadas as funcionalidades definidas e na escolha das tecnologias a utilizar, pois tendo em conta a grande dimensão e complexidade das restantes soluções, KamaeLei Web e KamaeLei Desktop, a escolha acertada das tecnologias a utilizar era de importância crítica. No entanto, essas dificuldades foram superadas em grande parte graças ao apoio dado pelos membros da equipa da Kamae RT que sempre se disponibilizaram para ajudar, mas também graças ao empenho e à persistência do estagiário. Deste modo, no final do período de estágio todos os objetivos inicialmente estipulados foram concluídos com sucesso, conseguindo obter-se uma versão 1.0 da aplicação.

Evidentemente que esta aplicação, tal como qualquer outra, vai estar sempre sujeita a melhorias e possíveis correções, principalmente com o desenvolvimento acelerado levado a cabo na versão *web desktop*, uma vez que as mudanças efetuadas nessa versão têm impacto direto sobre o funcionamento da aplicação *web mobile*. Como tal, esta deve ser mantida e o seu desenvolvimento continuado de forma a acompanhar essa evolução.

Para concluir, considera-se que a experiência de estágio na Kamae RT foi altamente enriquecedora, tanto a nível pessoal como a nível profissional, pois permitiu ao estagiário um maior entendimento sobre o funcionamento de uma empresa e sobre a gestão de projetos, assim como a oportunidade de trabalhar com profissionais com vários anos de experiência na área.



## Referências Bibliográficas

- [1] Insights for Marketers, Digital Portal, [Online]. Disponível em: <http://etc-digital.org/digital-trends/connectivity/insights-for-marketers/regional-overview/europe/>. [Acedido em Outubro 2014]
- [2] Mobile internet usage soars by 67%, StatCounter, [Online]. Disponível em: <http://gs.statcounter.com/press/mobile-internet-usage-soars-by-67-perc> . [Acedido em Outubro 2014]
- [3] Google's Android One initiative launches in India with three \$100 phones, CNet, [Online]. Disponível em: <http://www.cnet.com/news/googles-android-one-initiative-launches-in-india-with-three-low-cost-phones/> . [Acedido em Outubro 2014]
- [4] iPhone 3G, Wikipedia, [Online]. Disponível em: [https://en.wikipedia.org/wiki/IPhone\\_3G](https://en.wikipedia.org/wiki/IPhone_3G) . [Acedido em Novembro 2014]
- [5] Apple iPhone 3GS, Specout, [Online]. Disponível em: <http://smartphones.specout.com/l/141/Apple-iPhone-3GS> . [Acedido em Novembro 2014]
- [6] 10 Major Advantages of HTML5, HTML5Arena, [Online]. Disponível em: <http://www.html5arena.com/blog/html5/10-major-advantages-of-html5/> . [Acedido em Novembro 2014]
- [7] HTML5, Wikipedia, [Online], Disponível em: <https://en.wikipedia.org/wiki/HTML5> . [Acedido em Outubro 2014]
- [8] Cascading Style Sheets, Wkipedia, [Online]. Disponível em: [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). [Acedido em Novembro 2014]
- [9] Media queries, Wikipedia, [Online]. Disponível em: [https://en.wikipedia.org/wiki/Media\\_queries](https://en.wikipedia.org/wiki/Media_queries). [Acedido em Outubro 2014]
- [10] What the Heck is Shadow DOM?, Dimitri Glazkov, [Online]. Disponível em: <http://glazkov.com/2011/01/14/what-the-heck-is-shadow-dom/> . [Acedido em Novembro 2014]
- [11] Why you should be using Web Components now. And how., PHIL LEGGETTER, [Online]. Disponível em: <http://webcomponents.org/presentations/why-you-should-be-using-web-components-and-how-at-devweek/> . [Acedido em Março 2015]
- [12] Ruby on Rails, Wikipedia, [Online]. Disponível em: [https://en.wikipedia.org/wiki/Ruby\\_on\\_Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails) . [Acedido em Outubro 2014]

- [13] 20 Best Sites Built with Ruby on Rails, Developer Drive, [Online]. Disponível em: <http://www.developerdrive.com/2011/09/20-best-sites-built-with-ruby-on-rails/>. [Acedido em Novembro 2014]
- [14] Pricing, Xamarin, [Online]. Disponível em: <https://store.xamarin.com/>. [Acedido em Novembro 2014]
- [15] Pricing, LiveCode, [Online]. Disponível em: <https://livecode.com/products/livecode-platform/pricing/>. [Acedido em Novembro 2014]
- [16] Microsoft Visual Studio Ultimate 2012, Microsoft, [Online]. Disponível em: <http://www.microsoft.com/en-us/download/details.aspx?id=30678>. [Acedido em Outubro 2014]
- [17] Nuget, Nuget, [Online]. Disponível em: <https://www.nuget.org/>. [Acedido em Novembro 2014]
- [18] Codepen, Codepen, [Online]. Disponível em: <http://codepen.io/>. [Acedido em Novembro 2014]
- [19] Visual Studio Team Foundation Server 2012, Microsoft, [Online]. Disponível em: <http://www.microsoft.com/en-us/download/details.x?id=38185>. [Acedido em Novembro 2014]
- [20] Fluid, FluidUI, [Online]. Disponível em: <https://www.fluidui.com/>. [Acedido em Novembro 2014]
- [21] C#, Csharp-Station, [Online]. Disponível em: <http://www.csharp-station.com/>. [Acedido em Outubro 2014]
- [22] Introdução a consultas LINQ (C#), Microsoft, [Online]. Disponível em: <https://msdn.microsoft.com/pt-br/library/bb397906.aspx>. [Acedido em Outubro 2014]
- [23] Most Popular Languages of 2015, CodeEval, [Online]. Disponível em: <http://blog.codeeval.com/codeevalblog/2015#.VfGCCnBVhBc>. [Acedido em Março 2015]
- [24] PhoneGap, PhoneGap, [Online]. Disponível em: <http://phonegap.com/>. [Acedido em Janeiro 2015]
- [25] NodeJS, NodeJS, [Online]. Disponível em: <https://nodejs.org/en/>. [Acedido em Janeiro 2015]
- [26] NodeBots Robots powered by JavaScript, NodeBots, [Online]. Disponível em: <http://nodebots.io/>. [Acedido em Janeiro 2015]
- [27] HTML & CSS, W3C, [Online]. Disponível em: <http://www.w3.org/standards/webdesign/htmlcss>. [Acedido em Novembro 2014]
- [28] Using SVG, CSS-tricks, [Online]. Disponível em: <https://css-tricks.com/using-svg/>. [Acedido em Novembro 2014]
- [29] AJAX, W3Schools, [Online]. Disponível em: <http://www.w3schools.com/ajax/>. [Acedido em Novembro 2014]
- [30] JSON: What It Is, How It Works, & How to Use It, CopterLabs, [Online]. Disponível em: <http://www.copterlabs.com/blog/json-what-it-is-how-it-works-how-to-use-it/>. [Acedido em Dezembro 2014]

- [31] jQuery, Wikipédia, [Online]. Disponível em: <https://en.wikipedia.org/wiki/jQuery>. [Acedido em Novembro 2014]
- [32] jQuery Introduction, W3Schools, [Online]. Disponível em: [http://www.w3schools.com/jQuery/jQuery\\_intro.asp](http://www.w3schools.com/jQuery/jQuery_intro.asp) [Acedido em Outubro 2014]
- [33] Microsoft jQuery Unobtrusive Validation, Microsoft, [Online]. Disponível em: <https://www.nuget.org/packages/Microsoft.jQuery.Unobtrusive.Validation/>. [Acedido em Dezembro 2014]
- [34] senthilraj/TimePicki, Github - TimePicki, [Online]. Disponível em: <https://github.com/senthilraj/TimePicki>. [Acedido em Dezembro 2014]
- [35] Moment.js, Moment.js, [Online]. Disponível em: <http://momentjs.com/>. [Acedido em Dezembro 2014]
- [36] jQuery Real Person, keith-wood, [Online]. Disponível em: <http://keith-wood.name/realPerson.html>. [Acedido em Dezembro 2014]
- [37] Data attribute reference, jQuery Mobile, [Online]. Disponível em: <http://demos.jquerymobile.com/1.2.0/docs/api/data-attributes.html>. [Acedido em Novembro 2014]
- [38] jQuery Mobile 1.4 Browser Support, jQuery Mobile, [Online]. Disponível em: <https://jquerymobile.com/browser-support/1.4/>. [Acedido em Novembro 2014]
- [39] Scrum, Desenvolvimento ágil, [Online]. Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/>. [Acedido em Outubro 2014]
- [40] Understanding Action Filters, Microsoft, [Online]. Disponível em: <http://www.asp.net/mvc/overview/older-versions-1/controllers-and-routing/understanding-action-filters-cs>. [Acedido em Dezembro 2014]
- [41] How to trigger an event in input text after I stop typing/writing?, Stackoverflow, [Online]. Disponível em: <http://stackoverflow.com/questions/14042193/how-to-trigger-an-event-in-input-text-after-i-stop-typing-writing>. [Acedido em Dezembro 2014]





## Anexos

### Anexo A

#### Diagramas de sequência

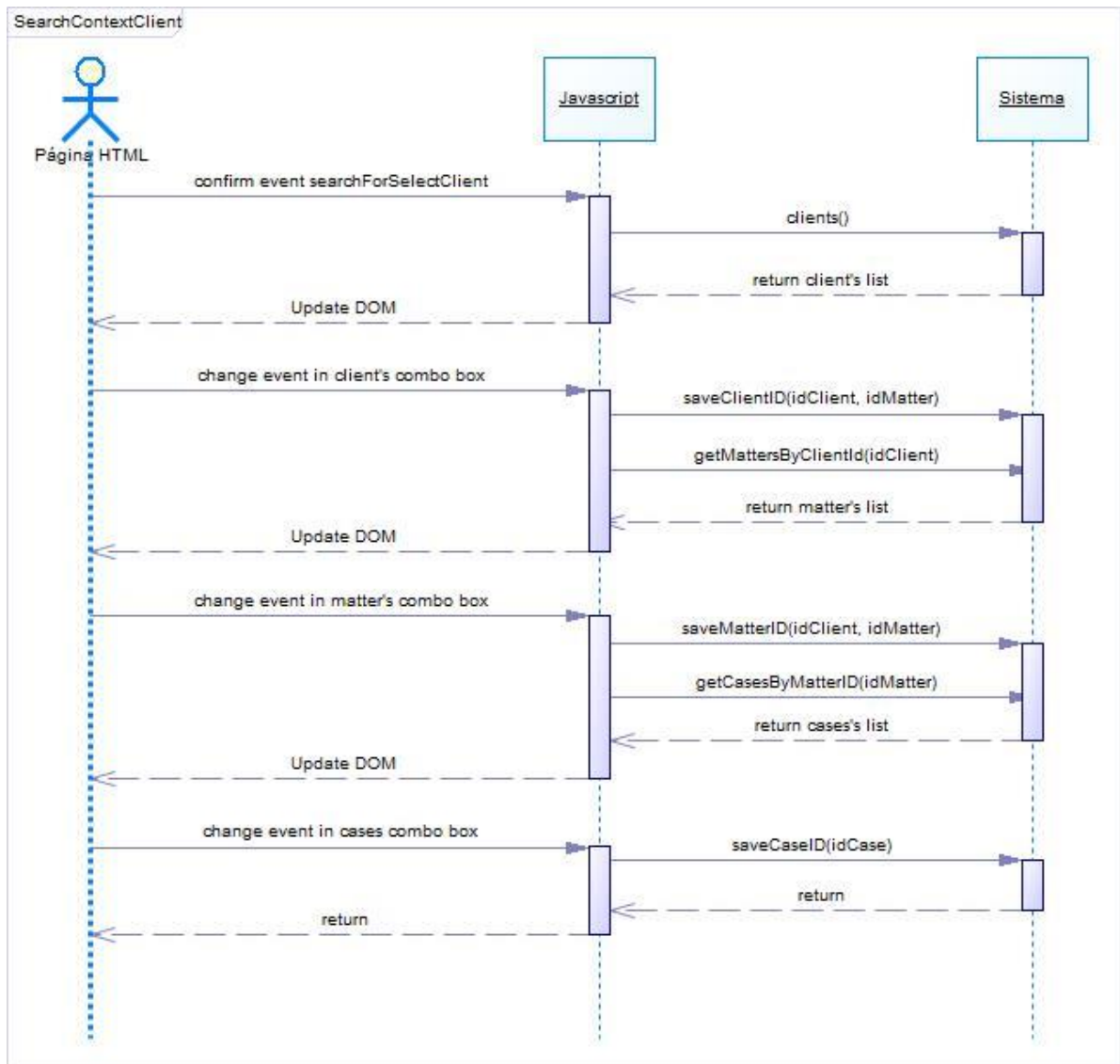


Figura 54 - Pesquisa de contexto por cliente

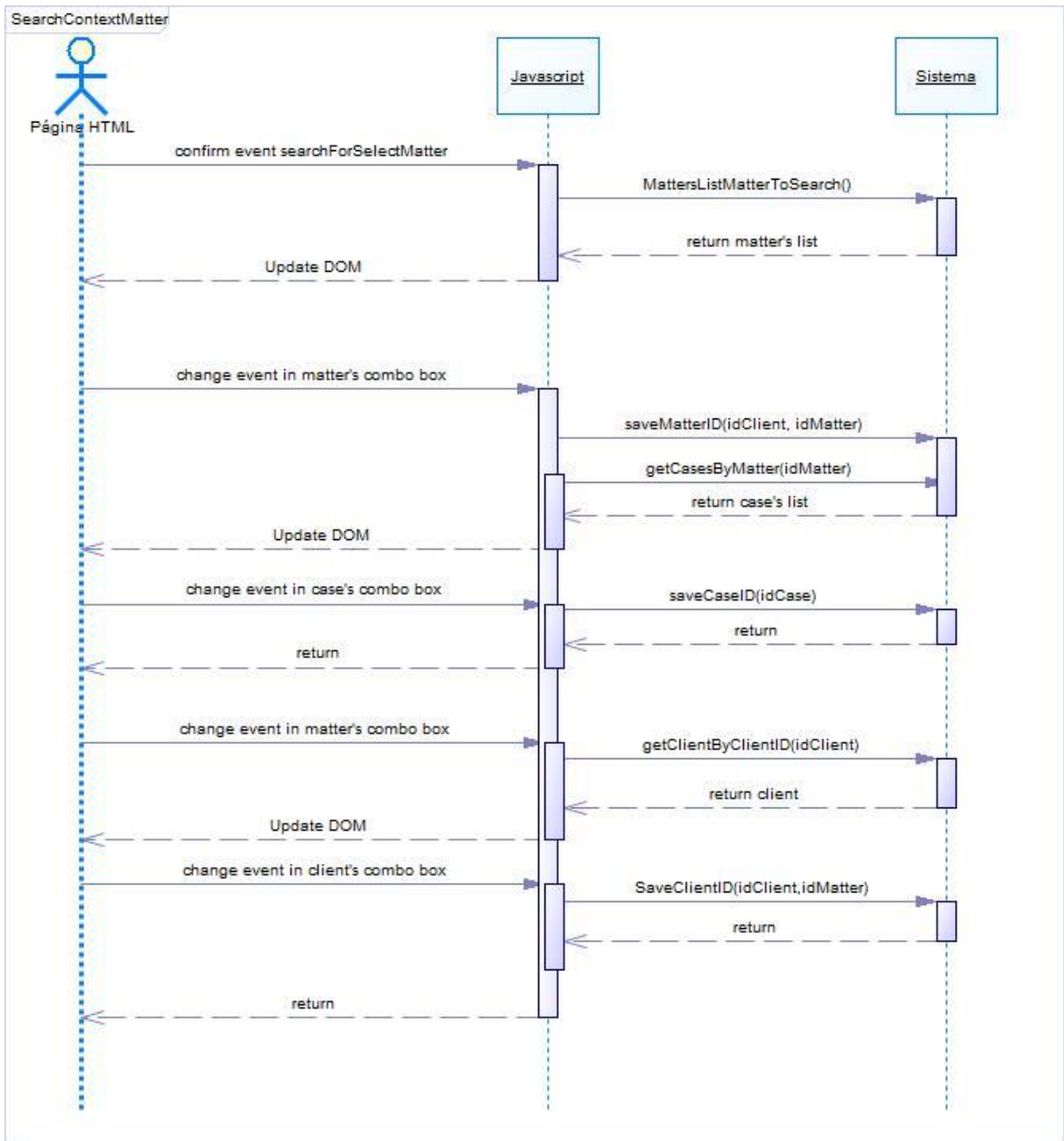


Figura 55 - Pesquisa de contexto por dossier

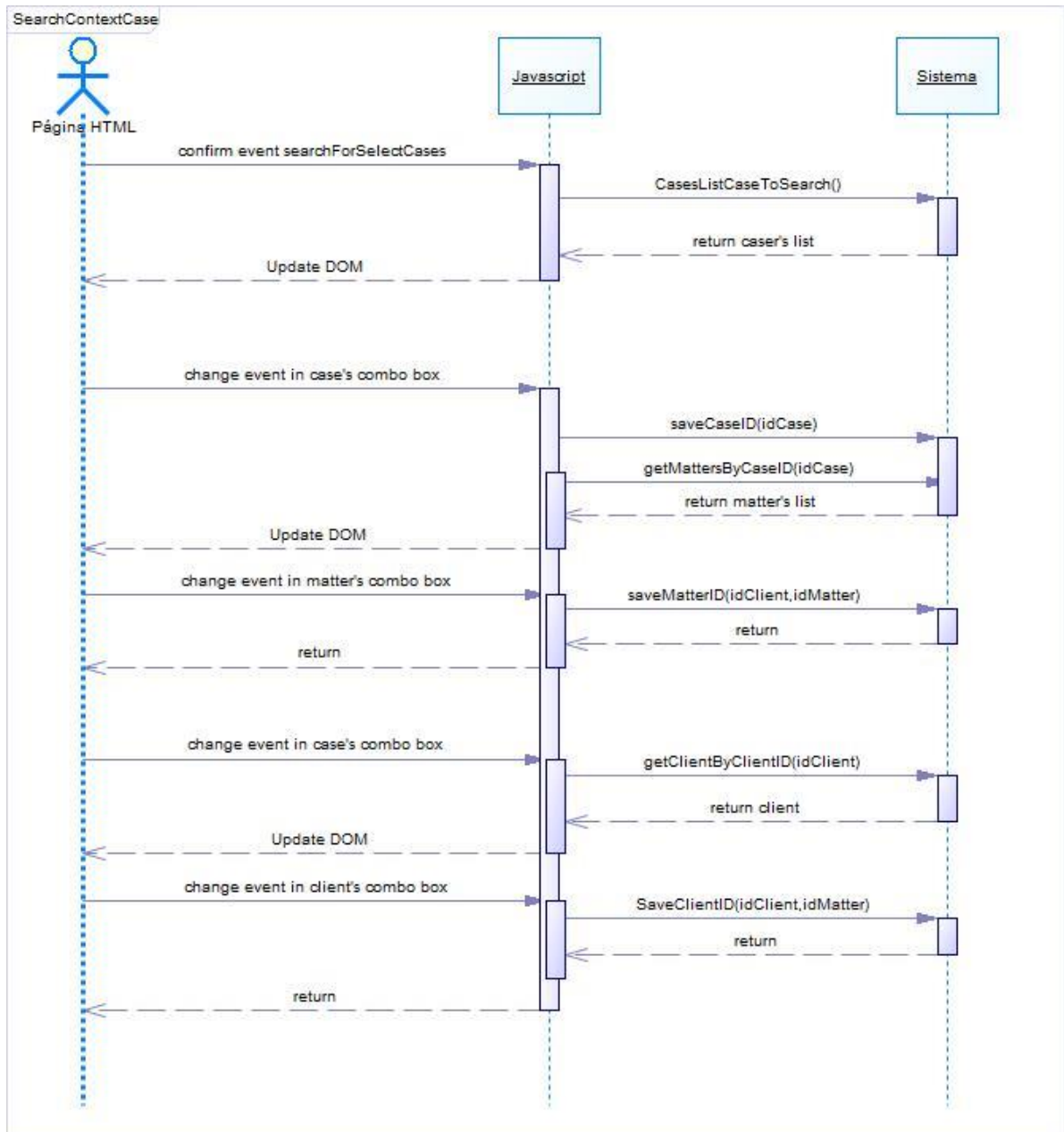


Figura 56 - Pesquisa de contexto por processo

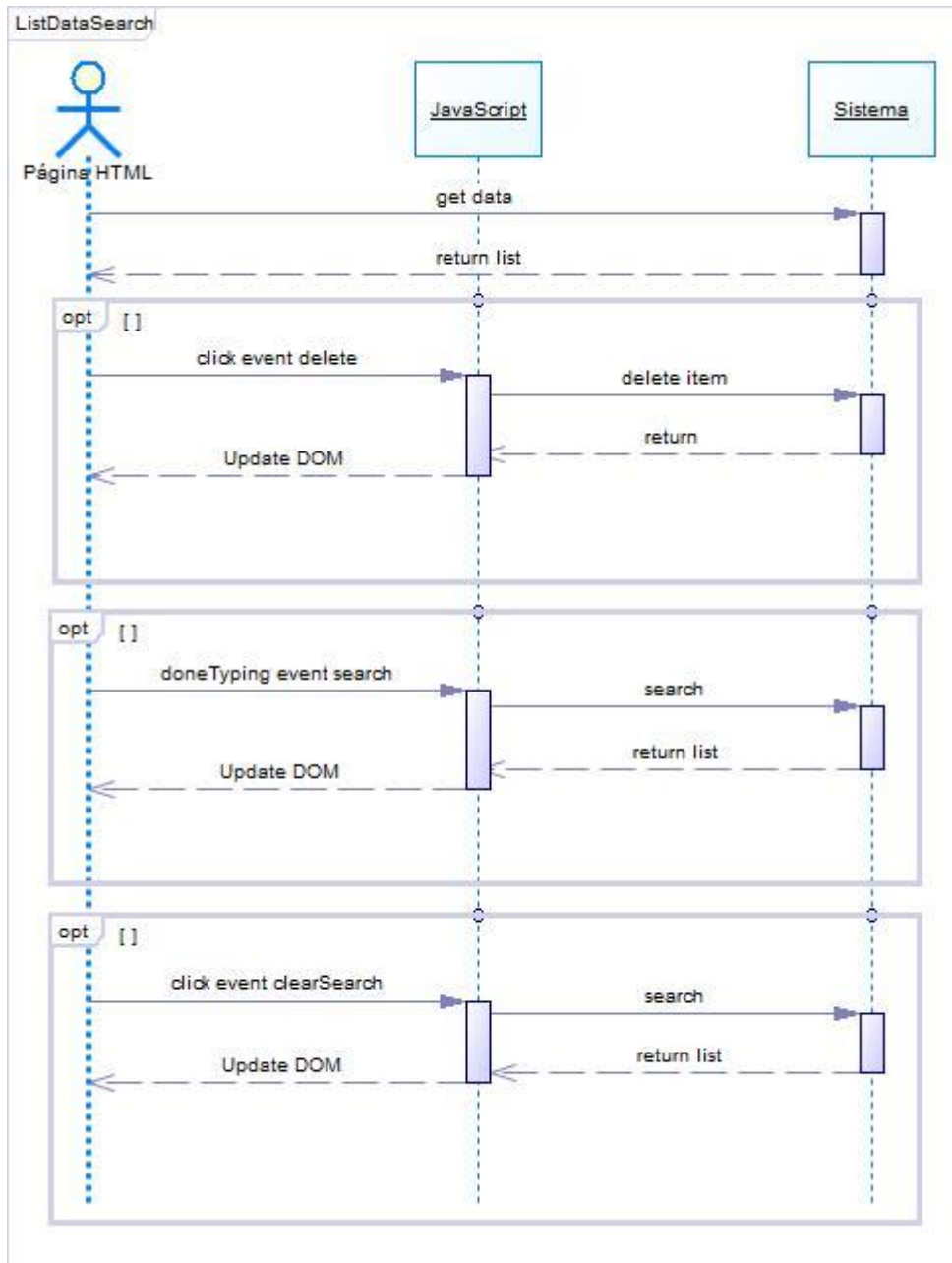


Figura 57 - Listagem de registos

S

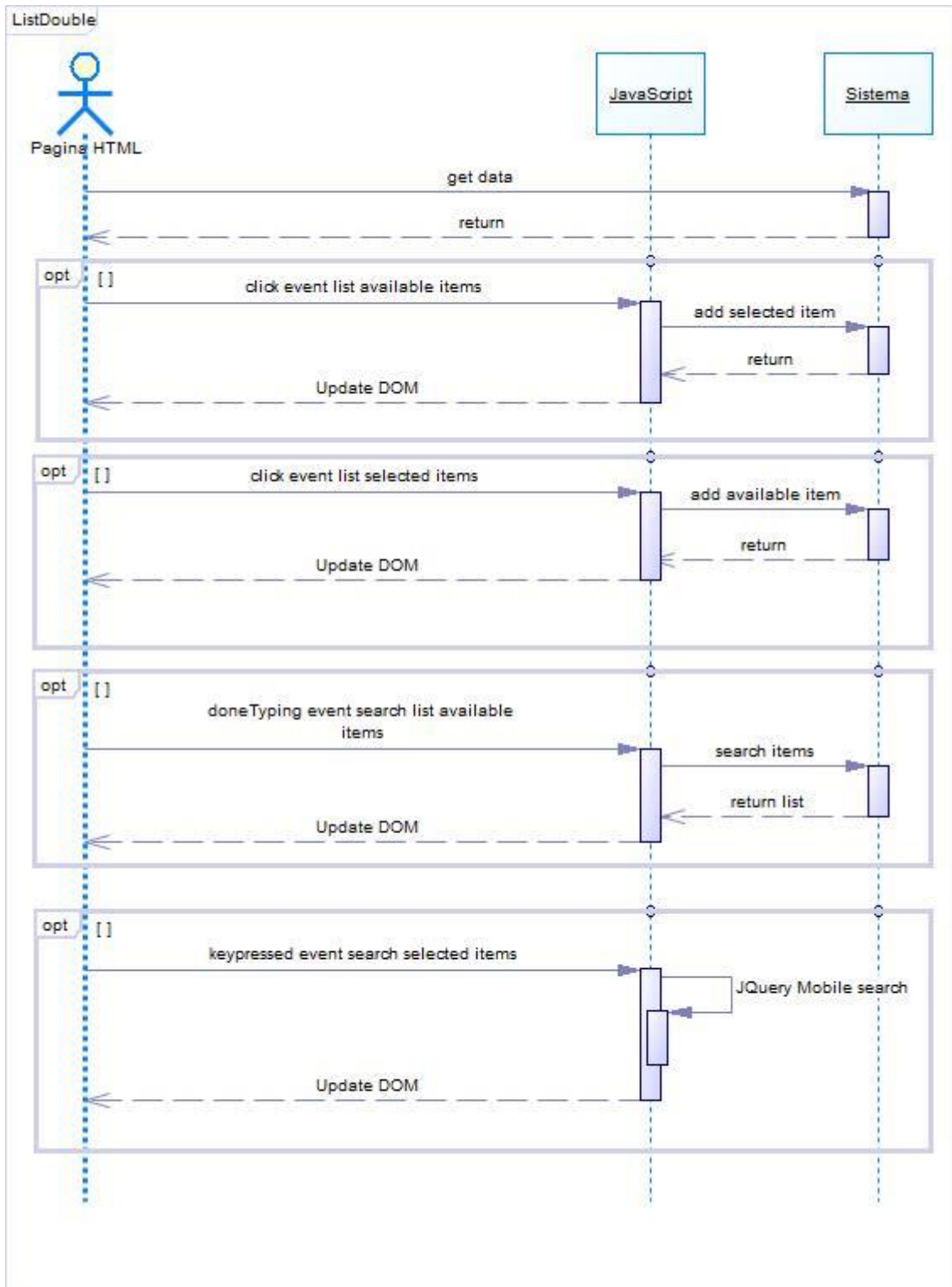
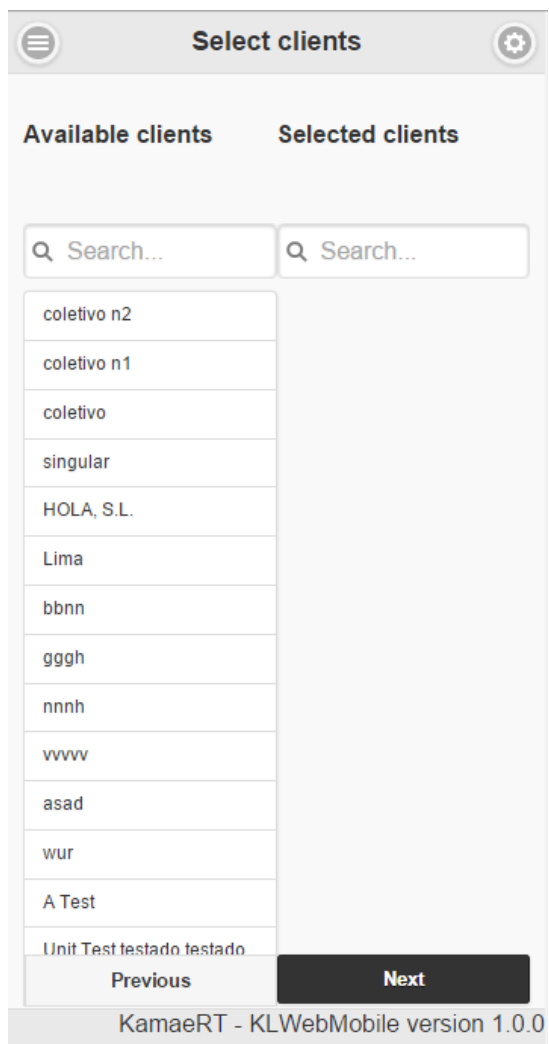


Figura 58 - Listagem duplas

## Anexo B

### Wizard de *Dossiers*



**Select clients**

Available clients      Selected clients

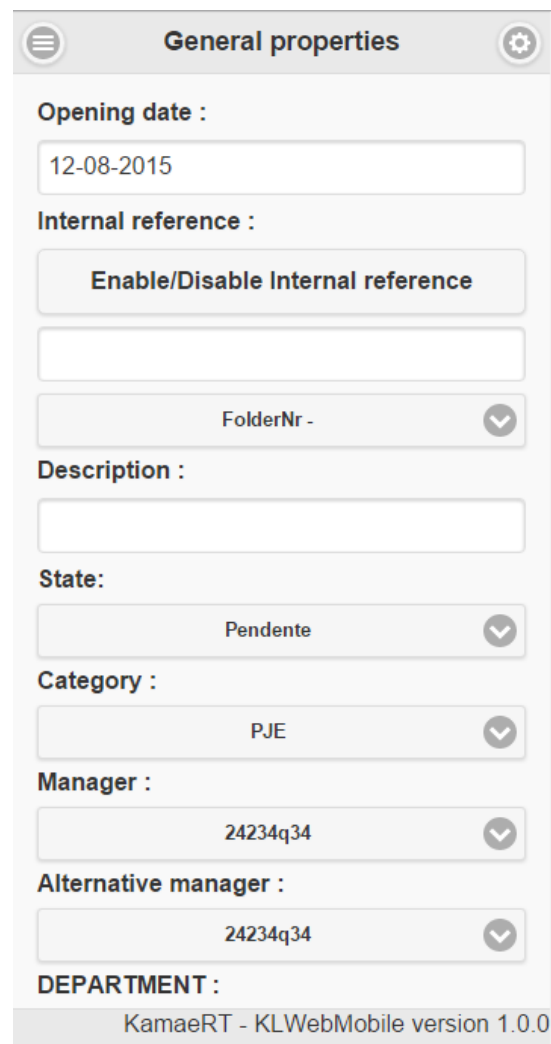
Search...      Search...

- coletivo n2
- coletivo n1
- coletivo
- singular
- HOLA, S.L.
- Lima
- bbnn
- gggh
- nnnh
- wwwv
- asad
- wur
- A Test
- Unit Test testado testado

Previous      **Next**

KamaeRT - KLWebMobile version 1.0.0

Figura 59 - Área de seleção de clientes – Wizard de *dossiers*



**General properties**

Opening date :  
12-08-2015

Internal reference :  
Enable/Disable Internal reference

FolderNr -

Description :

State:  
Pendente

Category :  
PJE

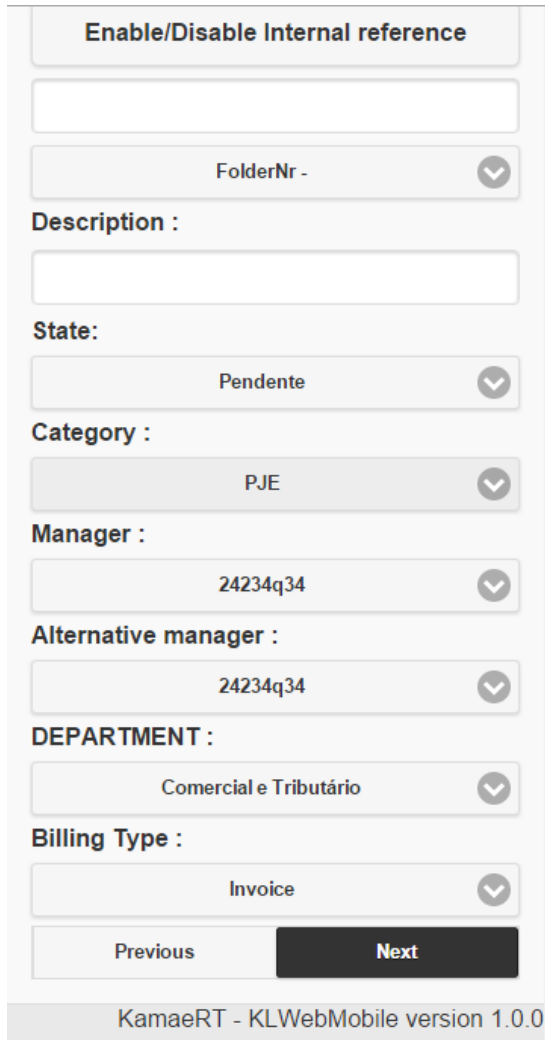
Manager :  
24234q34

Alternative manager :  
24234q34

DEPARTMENT :

KamaeRT - KLWebMobile version 1.0.0

Figura 60 - Área de propriedades gerais (parte 1) – Wizard de *dossiers*



**Enable/Disable Internal reference**

FolderNr -

**Description :**

**State:**

Pendente

**Category :**

PJE

**Manager :**

24234q34

**Alternative manager :**

24234q34

**DEPARTMENT :**

Comercial e Tributário

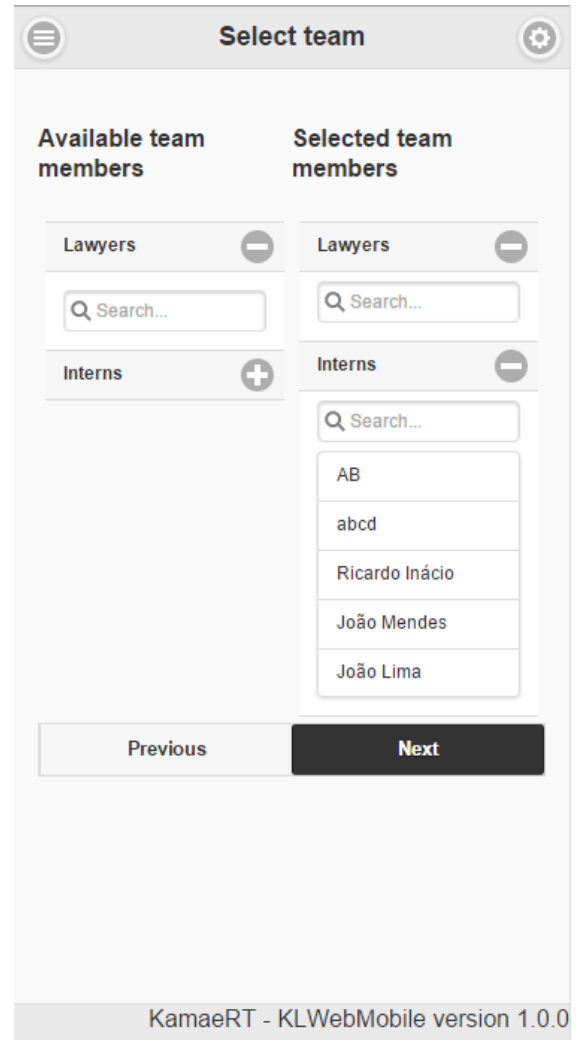
**Billing Type :**

Invoice

Previous

KamaeRT - KLWebMobile version 1.0.0

FIGURA 61 - Área de propriedades gerais (parte 2) – Wizard de dossiers



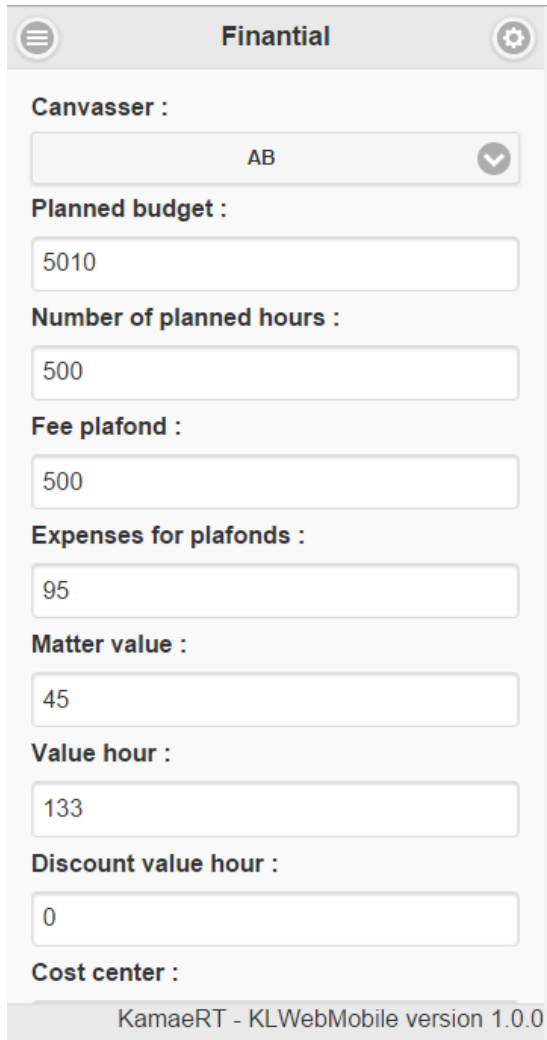
**Select team**

Available team members	Selected team members
Lawyers <input type="button" value="-"/>	Lawyers <input type="button" value="-"/>
<input type="text" value="Q Search..."/>	<input type="text" value="Q Search..."/>
Interns <input type="button" value="+"/>	Interns <input type="button" value="-"/>
	<input type="text" value="Q Search..."/>
	AB
	abcd
	Ricardo Inácio
	João Mendes
	João Lima

Previous

KamaeRT - KLWebMobile version 1.0.0

Figura 62 - Área de seleção de equipa – Wizard de dossiers



**Financial**

**Canvasser :**  
AB

**Planned budget :**  
5010

**Number of planned hours :**  
500

**Fee plafond :**  
500

**Expenses for plafonds :**  
95

**Matter value :**  
45

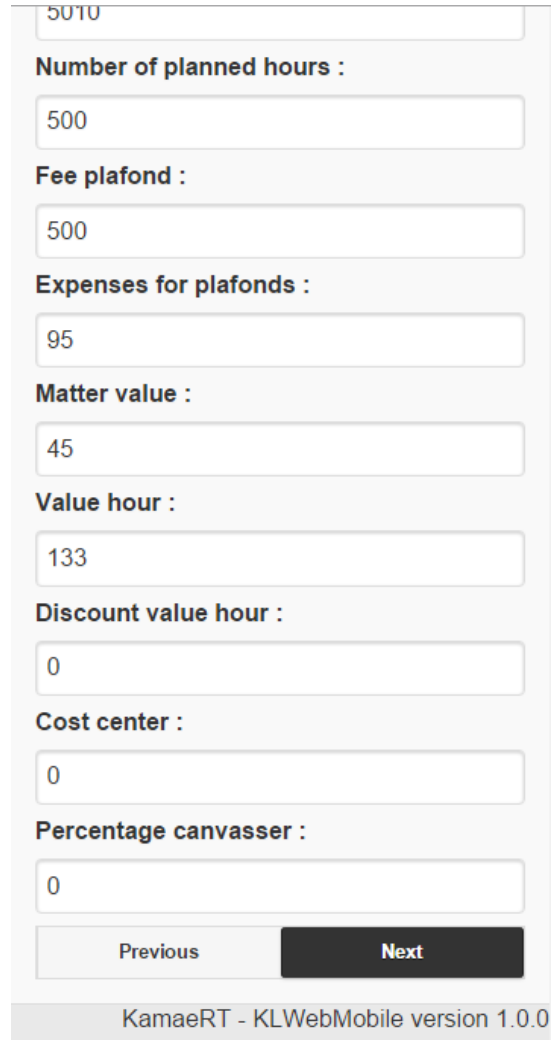
**Value hour :**  
133

**Discount value hour :**  
0

**Cost center :**

KamaeRT - KLWebMobile version 1.0.0

Figura 63 - Área financeira (parte 1) - Wizard de dossiers



5010

**Number of planned hours :**  
500

**Fee plafond :**  
500

**Expenses for plafonds :**  
95

**Matter value :**  
45

**Value hour :**  
133

**Discount value hour :**  
0

**Cost center :**  
0

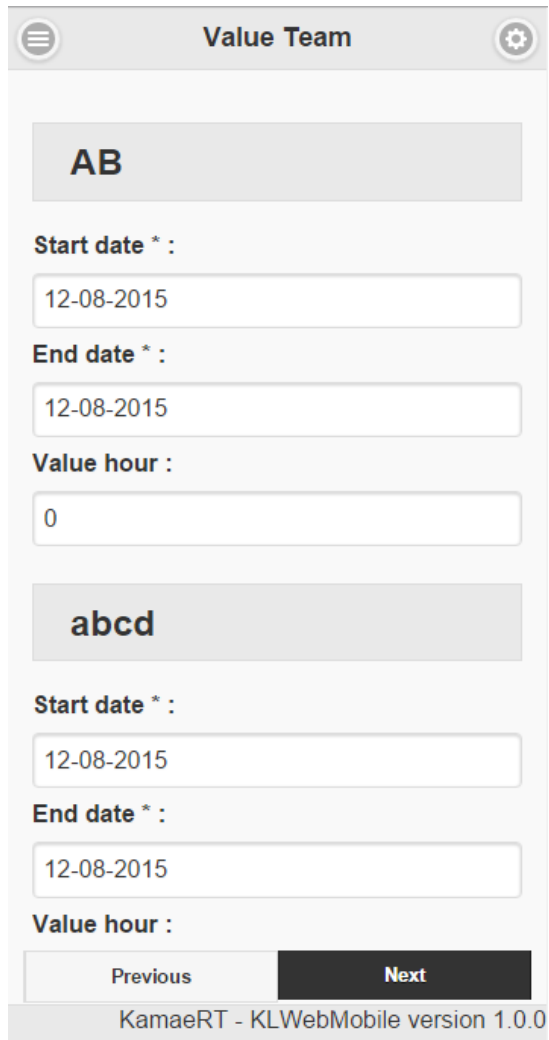
**Percentage canvasser :**  
0

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 64 - Área financeira (parte 2) - Wizard de dossiers





**Value Team**

**AB**

Start date \* :  
12-08-2015

End date \* :  
12-08-2015

Value hour :  
0

**abcd**

Start date \* :  
12-08-2015

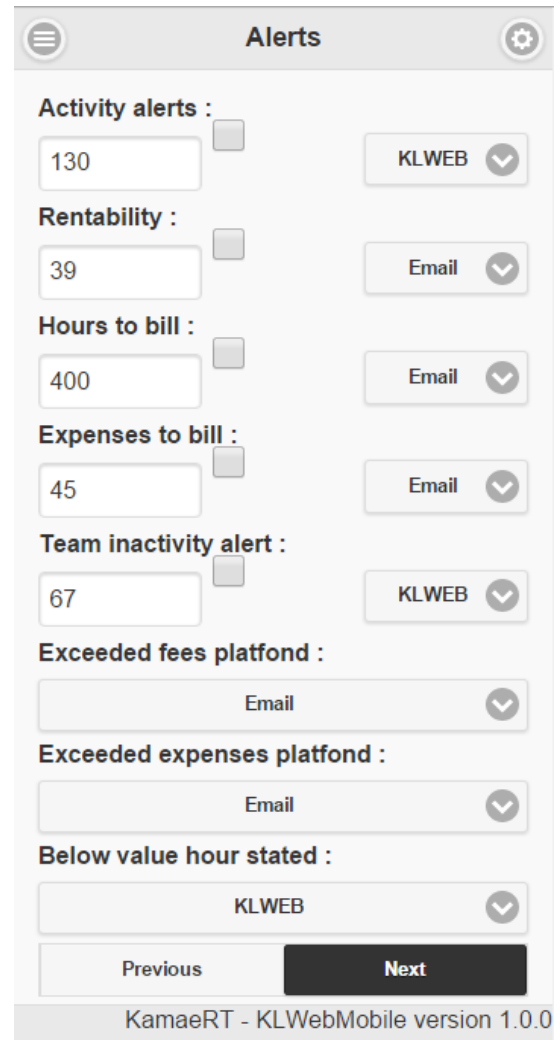
End date \* :  
12-08-2015

Value hour :

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 65 - Área de custo da equipa – Wizard de dossiers



**Alerts**

Activity alerts :  
130 KLWEB

Rentability :  
39 Email

Hours to bill :  
400 Email

Expenses to bill :  
45 Email

Team inactivity alert :  
67 KLWEB

Exceeded fees platfond :  
Email

Exceeded expenses platfond :  
Email

Below value hour stated :  
KLWEB

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 66 - Área de alertas – Wizard de dossiers

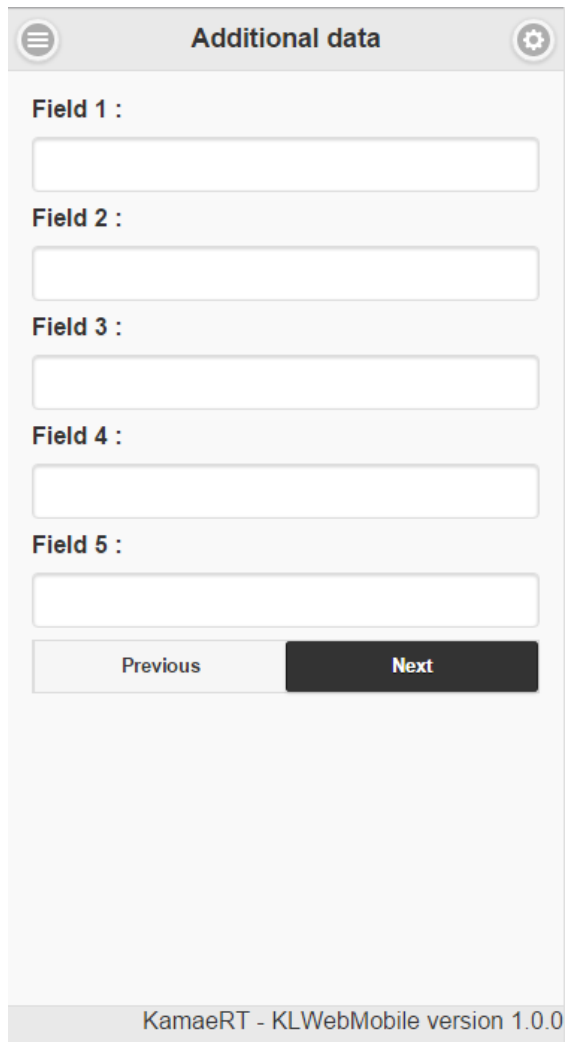


Figura 67 - Área de dados adicionais – Wizard de dossiers



Figura 68 - Área de representados da parte oposta – Wizard de dossiers



**Opposing part representative**

**Available entities**      **Selected entities**

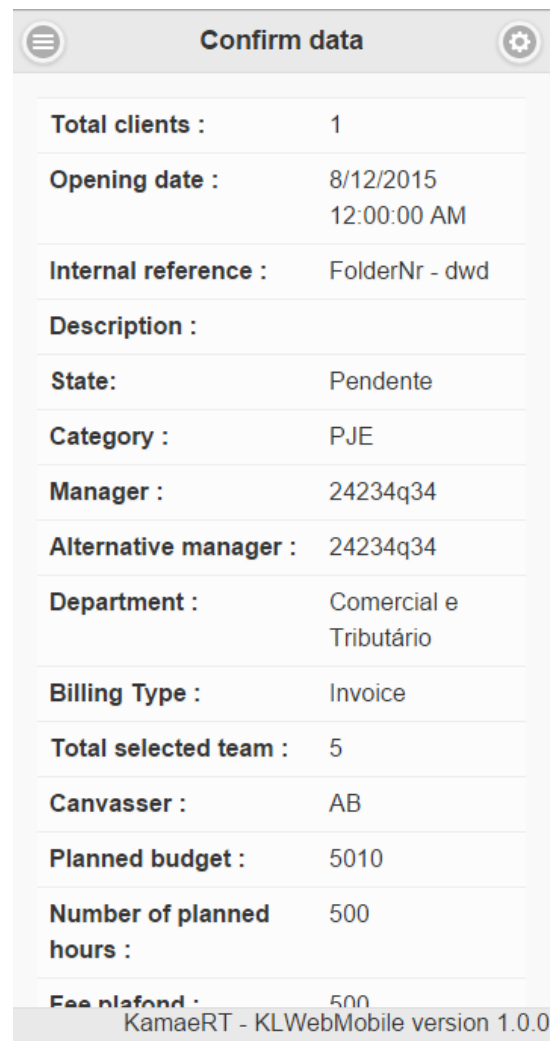
Q Search...      Q Search...

- Benavente - Tribunal Judicial da Comarca de Benavente
- Loão Jaborinho Lúcio
- Darlos Cuque
- Sandrina Laurentino
- Beatriz Paula Valério
- Ana Rita Rodrigues
- Fna Auarte Derreira
- Picente Valdeira Cires
- Teresa Aleixo
- Joana Coimbra de Almeida
- António Fontoura de Oliveira
- Marta Silveira Leal
- Diana de Avelar Bandeira

Previous      **Next**

KamaeRT - KLWebMobile version 1.0.0

Figura 69 - Área de representantes da parte oposta - Wizard de dossiers



**Confirm data**

**Total clients :** 1

**Opening date :** 8/12/2015  
12:00:00 AM

**Internal reference :** FolderNr - dwd

**Description :**

**State:** Pendente

**Category :** PJE

**Manager :** 24234q34

**Alternative manager :** 24234q34

**Department :** Comercial e Tributário

**Billing Type :** Invoice

**Total selected team :** 5

**Canvasser :** AB

**Planned budget :** 5010

**Number of planned hours :** 500

**Fee plafond :** 500

KamaeRT - KLWebMobile version 1.0.0

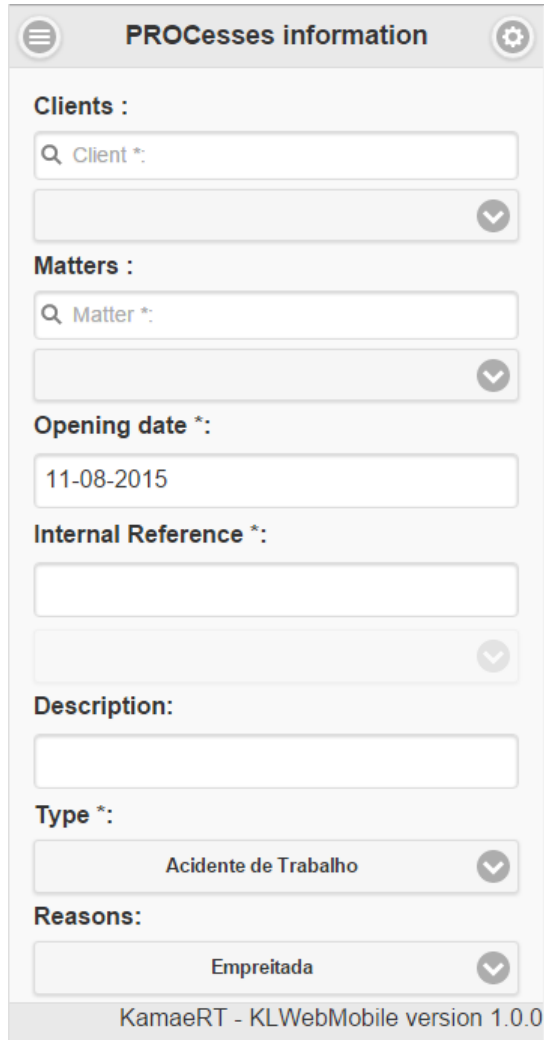
Figura 70 - Área de confirmação de dados (parte 1) - Wizard de dossiers

<b>Expenses for plafonds :</b>	95
<b>Matter value :</b>	45
<b>Value hour :</b>	133
<b>Discount value hour :</b>	0
<b>Percentage canvasser :</b>	0
<b>Exceeded fees plafond :</b>	Email
<b>Exceeded expenses plafond :</b>	Email
<b>Below value hour stated :</b>	KLWEB
<b>Field 1 :</b>	
<b>Field 2 :</b>	
<b>Field 3 :</b>	
<b>Field 4 :</b>	
<b>Field 5 :</b>	
<input type="button" value="Previous"/>	<input type="button" value="Next"/>

KamaeRT - KLWebMobile version 1.0.0

Figura 71 - Área de confirmação de dados (parte 2) –  
Wizard de dossiers

## Wizard de processos



**PROcesses information**

**Clients :**

Q Client \*:

**Matters :**

Q Matter \*:

**Opening date \*:**

11-08-2015

**Internal Reference \*:**

**Description:**

**Type \*:**

Acidente de Trabalho

**Reasons:**

Empreitada

KamaeRT - KLWebMobile version 1.0.0

Figura 72 - Área de informação geral (parte 1) - Wizard de processos



**Internal Reference \*:**

**Description:**

**Type \*:**

Acidente de Trabalho

**Reasons:**

Empreitada

**Status \*:**

Pendente

**Managers \*:**

24234q34

**DEPARTMENTS \*:**

3434

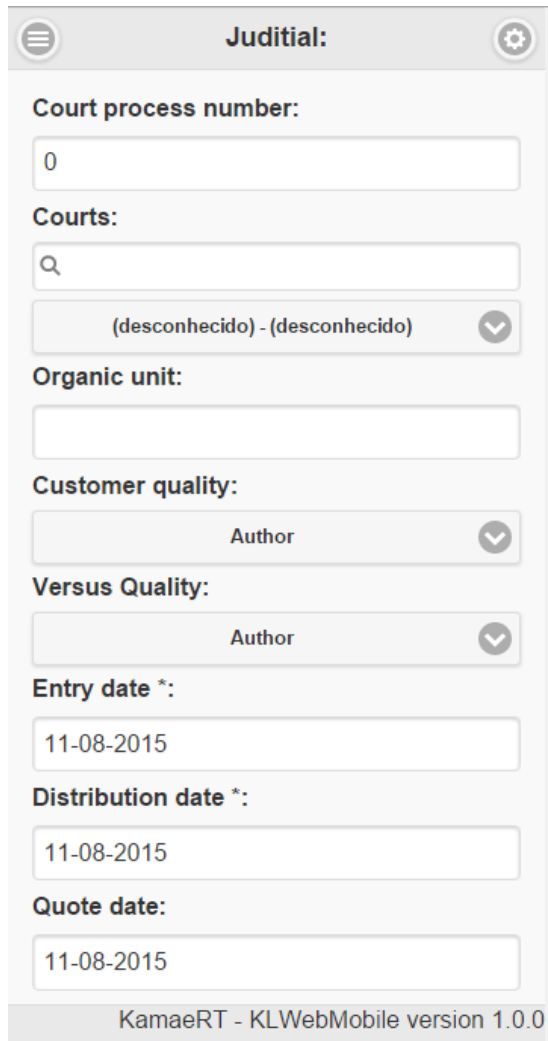
**Qualities \*:**

Contract

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 73 - Área de informação geral (parte 2) - Wizard de processos



**Judicial:**

**Court process number:**  
0

**Courts:**  
Q  
(desconhecido) - (desconhecido)

**Organic unit:**

**Customer quality:**  
Author

**Versus Quality:**  
Author

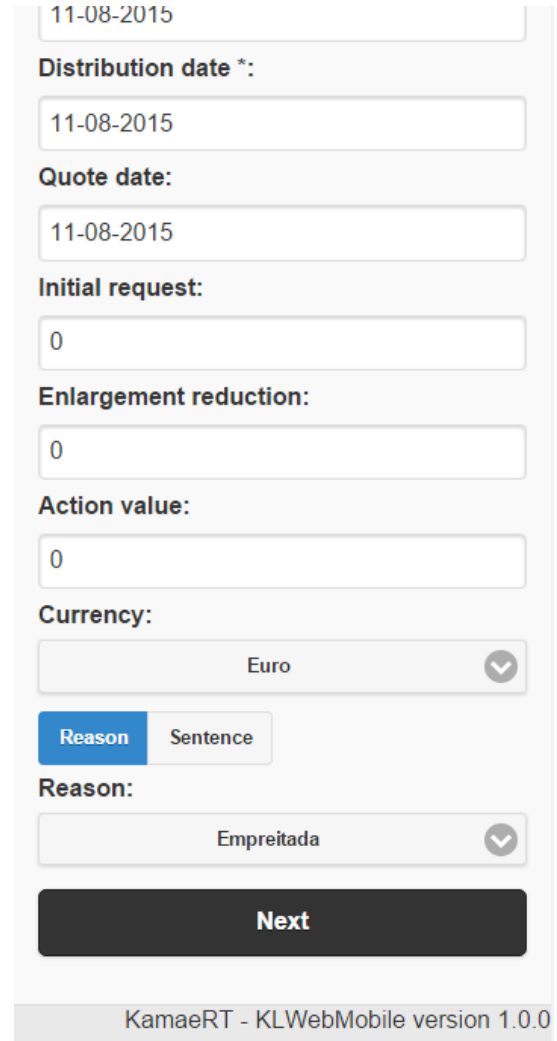
**Entry date \*:**  
11-08-2015

**Distribution date \*:**  
11-08-2015

**Quote date:**  
11-08-2015

KamaeRT - KLWebMobile version 1.0.0

Figura 74 - Área de informação judicial (parte 1) - Wizard de processos



11-08-2015

**Distribution date \*:**  
11-08-2015

**Quote date:**  
11-08-2015

**Initial request:**  
0

**Enlargement reduction:**  
0

**Action value:**  
0

**Currency:**  
Euro

**Reason** Sentence

**Reason:**  
Empreitada

**Next**

KamaeRT - KLWebMobile version 1.0.0

Figura 75 - Área de informação judicial (parte 2) - Wizard de processos



<b>Opening date *</b> :	11-08-2015
<b>Internal Reference *</b> :	W.1.0 - Pasta Geral - CaseNrOnFolder3Digit
<b>Description:</b>	
<b>Type *</b> :	Acidente de Trabalho
<b>Reason</b>	Empreitada
<b>Status *</b> :	Pendente
<b>Managers *</b> :	ac
<b>Departments *</b> :	Recuperação de Crédito II
<b>Qualities *</b> :	Judicial

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 76 - Área de confirmação de dados – Wizard de processos

## Wizard de Clientes

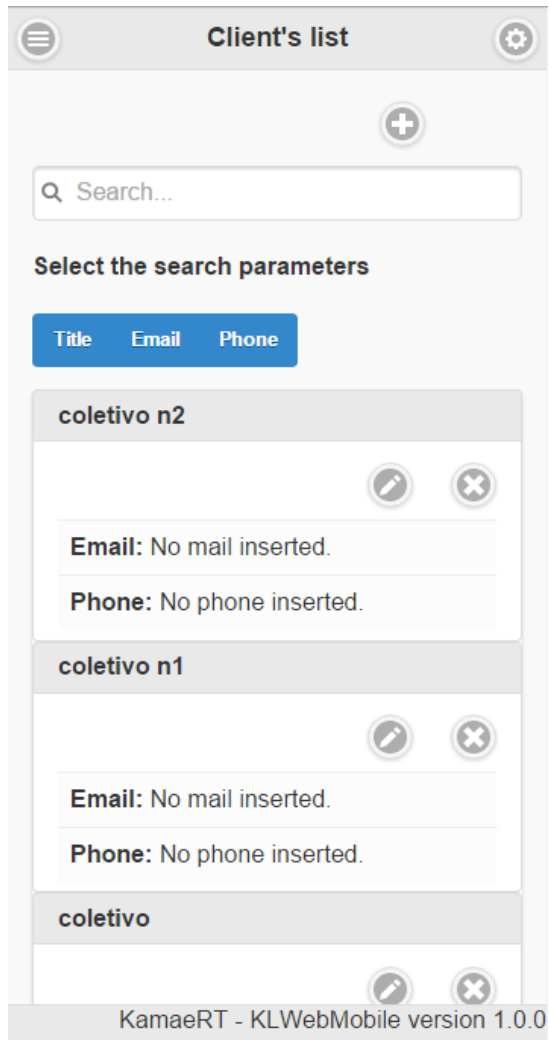


Figura 77 - Área de listagem de clientes – Wizard de clientes

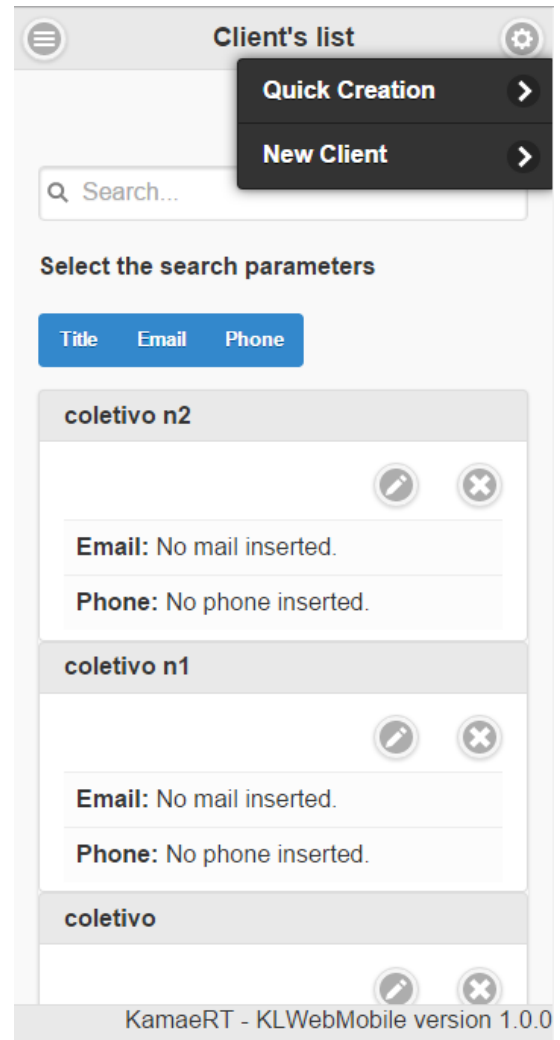
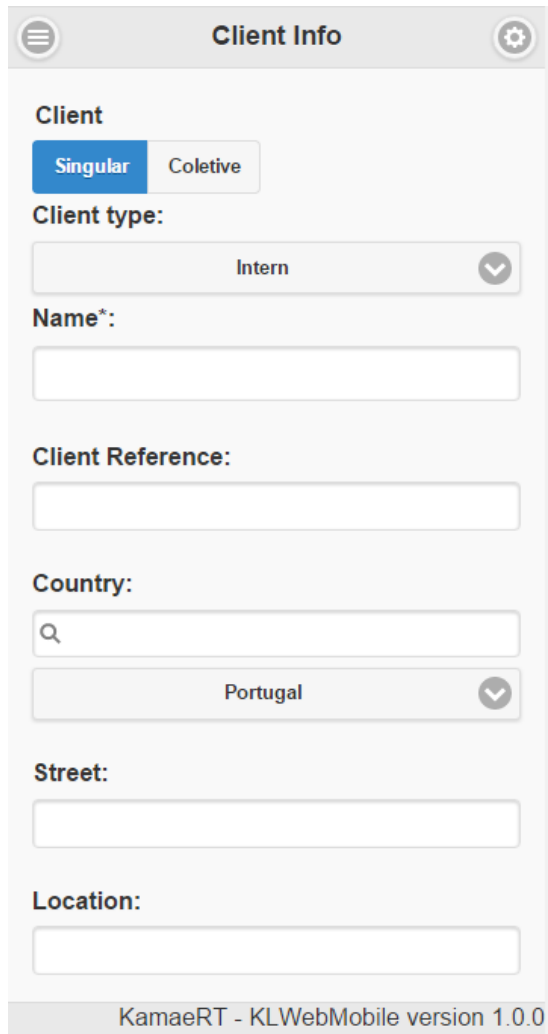


Figura 78 - Área de listagem de clientes – Wizard de clientes





**Client Info**

**Client**

Singular Coletive

**Client type:**

Intern

**Name\*:**

**Client Reference:**

**Country:**

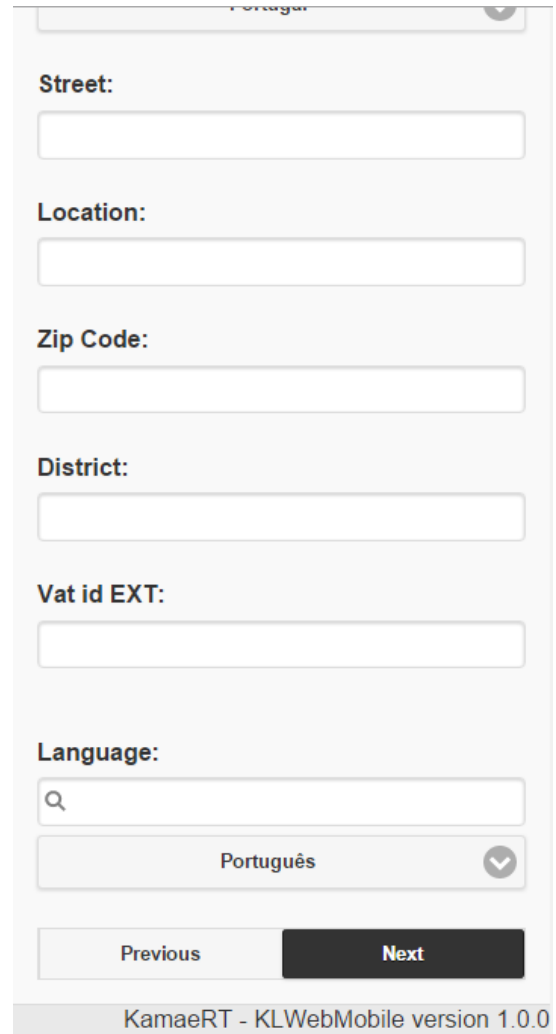
Portugal

**Street:**

**Location:**

KamaeRT - KLWebMobile version 1.0.0

Figura 79 - Área de informações do cliente (parte 1) - Wizard de clientes



**Street:**

**Location:**

**Zip Code:**

**District:**

**Vat id EXT:**

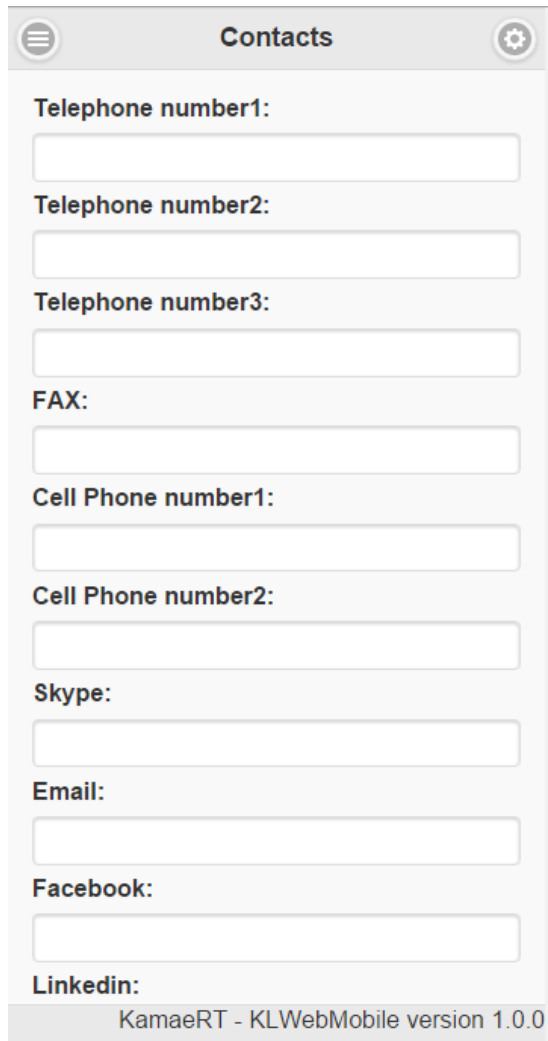
**Language:**

Português

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 80 - Área de informações do cliente (parte 2) - Wizard de clientes



**Contacts**

Telephone number1:

Telephone number2:

Telephone number3:

FAX:

Cell Phone number1:

Cell Phone number2:

Skype:

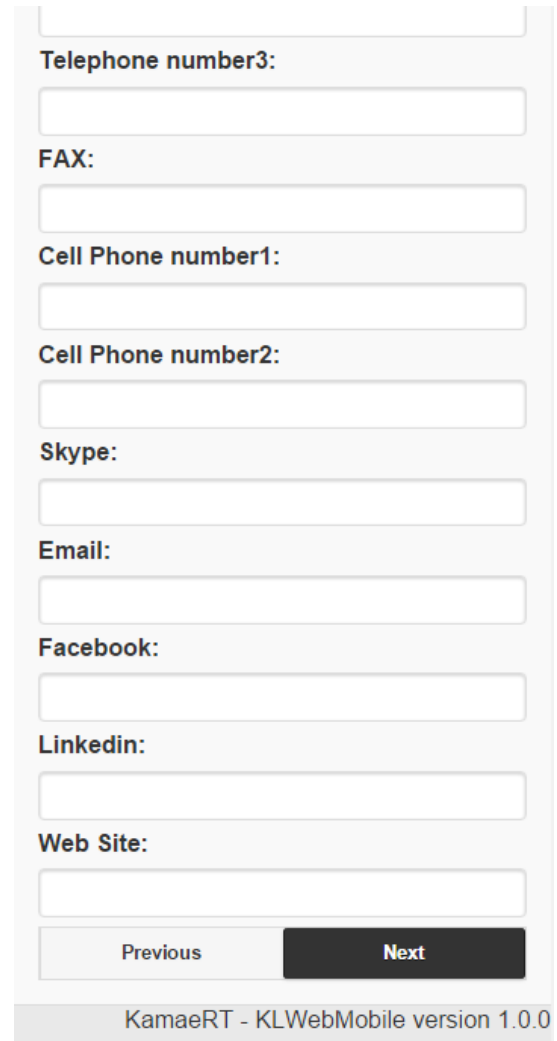
Email:

Facebook:

LinkedIn:

KamaeRT - KLWebMobile version 1.0.0

Figura 81 - Área de contactos (parte 1) – Wizard de clientes



Telephone number3:

FAX:

Cell Phone number1:

Cell Phone number2:

Skype:

Email:

Facebook:

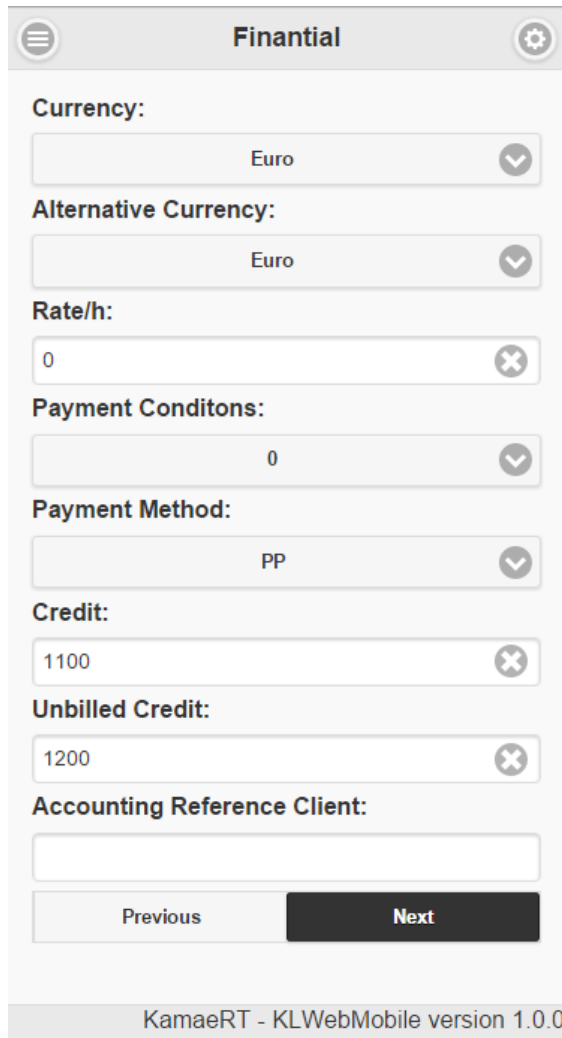
LinkedIn:

Web Site:

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 82 - Área de contactos (parte 2) – Wizard de clientes



**Financial**

Currency: Euro

Alternative Currency: Euro

Rate/h: 0

Payment Conditions: 0

Payment Method: PP

Credit: 1100

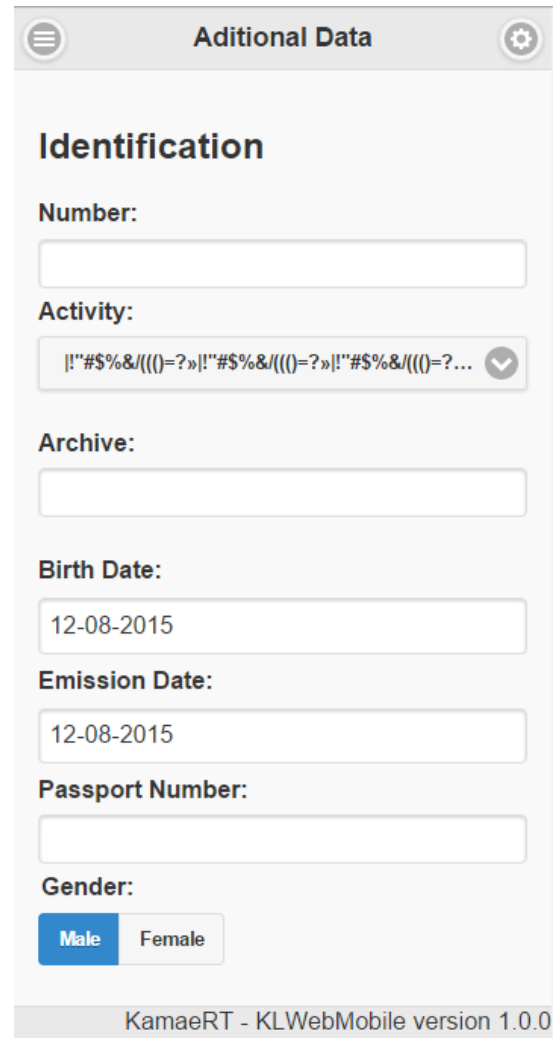
Unbilled Credit: 1200

Accounting Reference Client:

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 83 - Área financeira – Wizard de clientes



**Additional Data**

**Identification**

Number:

Activity:

Archive:

Birth Date: 12-08-2015

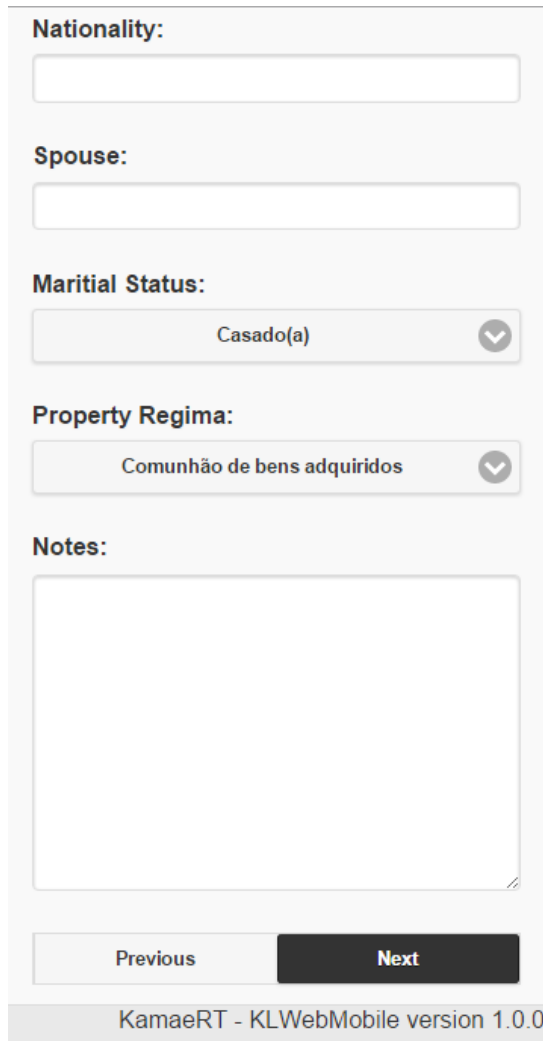
Emission Date: 12-08-2015

Passport Number:

Gender: Male Female

KamaeRT - KLWebMobile version 1.0.0

Figura 84 - Área de dados adicionais singular (parte 1) – Wizard de clientes



**Nationality:**

**Spouse:**

**Marital Status:**

Casado(a) ▼

**Property Regima:**

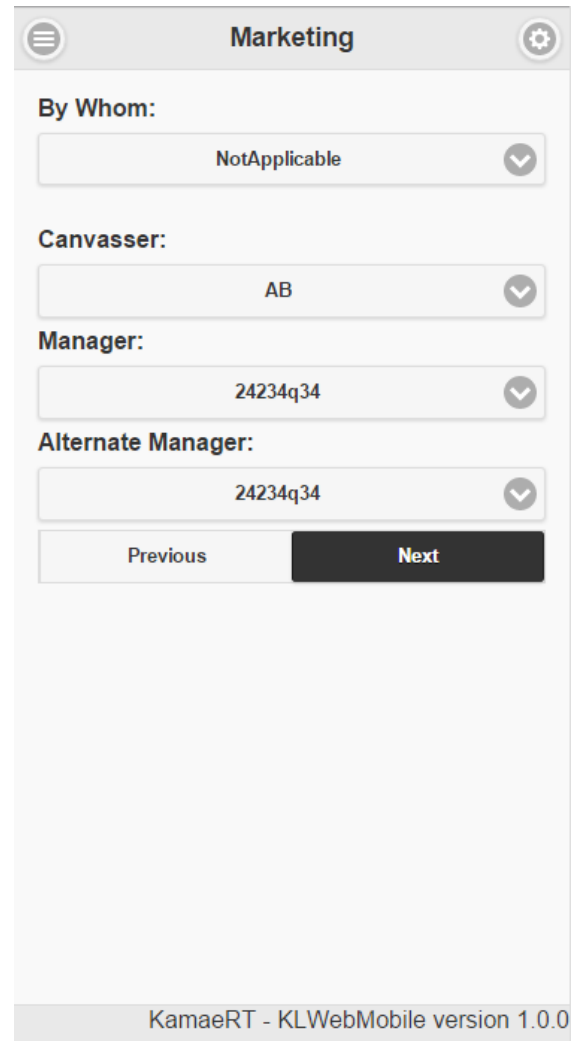
Comunhão de bens adquiridos ▼

**Notes:**

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 85 - Área de dados adicionais singular (parte 2)  
- Wizard de clientes



**Marketing**

**By Whom:**

NotApplicable ▼

**Convasser:**

AB ▼

**Manager:**

24234q34 ▼

**Alternate Manager:**

24234q34 ▼

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 86 - Área de marketing – Wizard de clientes

Confirmation	
<b>Client:</b>	Singular
<b>Client Type:</b>	Intern
<b>Name:</b>	Carlos Nohueira
<b>Client Reference:</b>	Carlos
<b>Country:</b>	Portugal
<b>Street:</b>	Rua Do Mira Crastro N°9 Alfeloas Anadia
<b>Location:</b>	3780-314
<b>ZipCode:</b>	3780-314
<b>District:</b>	Anadia
<b>Language:</b>	Português
<b>Currency:</b>	Euro
<b>Alternative Currency:</b>	Euro
<b>Rate/h:</b>	0.00
<b>Payment</b>	0
<b>Conditons:</b>	

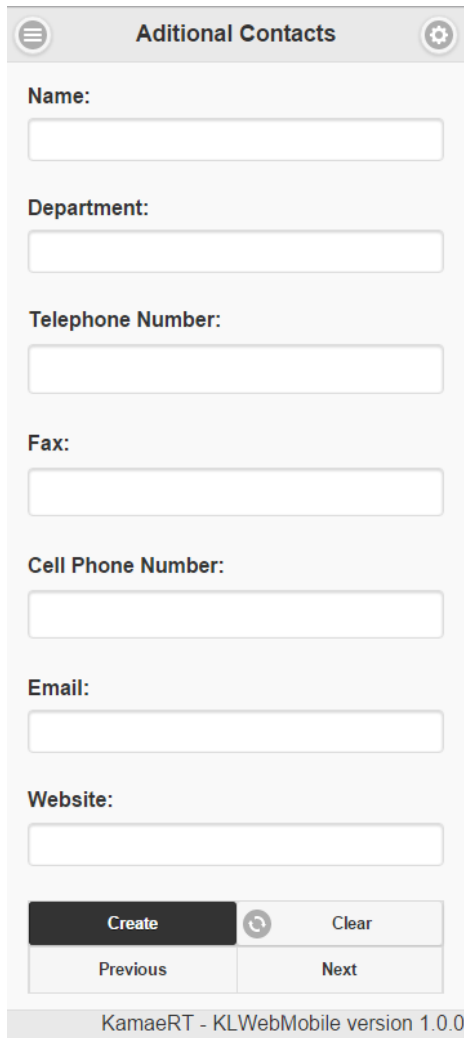
KamaeRT - KLWebMobile version 1.0.0

Figura 87 - Área de confirmação (parte 1) – Wizard de clientes

Conditons:	
<b>Payment Method:</b>	PP
<b>Credit:</b>	1100.00
<b>Unbilled Credit:</b>	1200.00
<b>Birth Date:</b>	8/12/2015 12:00:00 AM
<b>Emission Date:</b>	8/12/2015 12:00:00 AM
<b>Gender:</b>	M
<b>Marital Status:</b>	Casado(a)
<b>Property Regima:</b>	Comunhão de bens adquiridos
<b>By Whom:</b>	NotApplicable
<b>Canvasser:</b>	AB
<b>Manager:</b>	24234q34
<b>Alternate Manager:</b>	24234q34
<b>Previous</b>	<b>Next</b>

KamaeRT - KLWebMobile version 1.0.0

Figura 88 - Área de confirmação (parte 2) – Wizard de clientes



**Additional Contacts**

Name:

Department:

Telephone Number:

Fax:

Cell Phone Number:

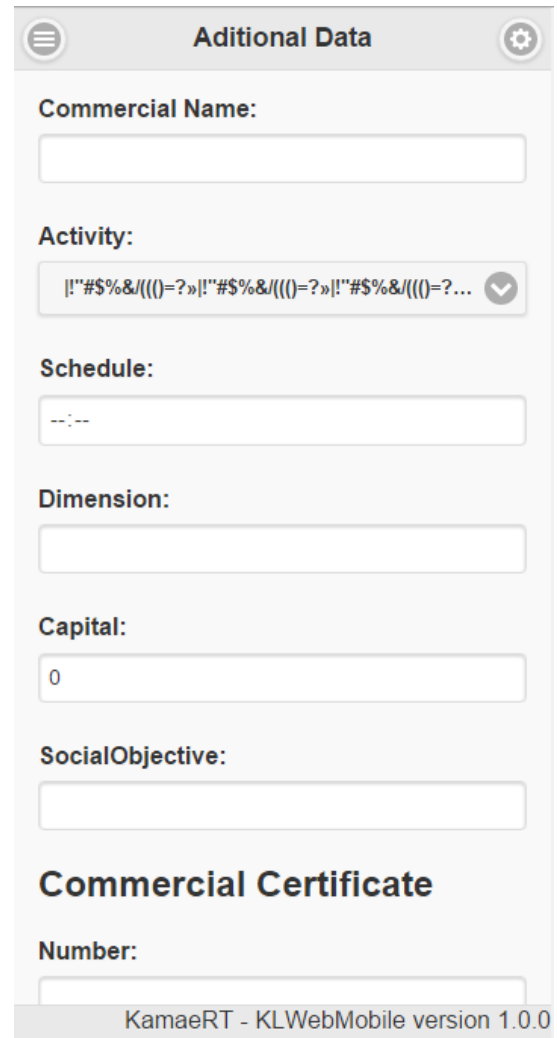
Email:

Website:

Create Clear  
Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 89 - Área de contactos adicionais – Wizard de clientes



**Additional Data**

Commercial Name:

Activity:

Schedule:

Dimension:

Capital:

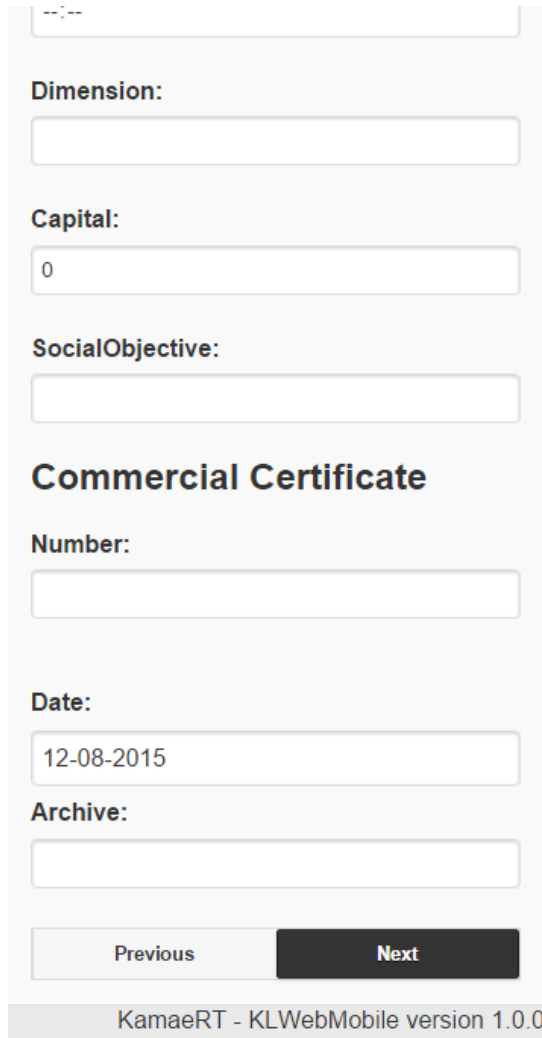
SocialObjective:

**Commercial Certificate**

Number:

KamaeRT - KLWebMobile version 1.0.0

Figura 90 - Área de dados adicionais coletivos (parte 1) – Wizard de clientes



Dimension:

Capital:

0

SocialObjective:

**Commercial Certificate**

Number:

Date:

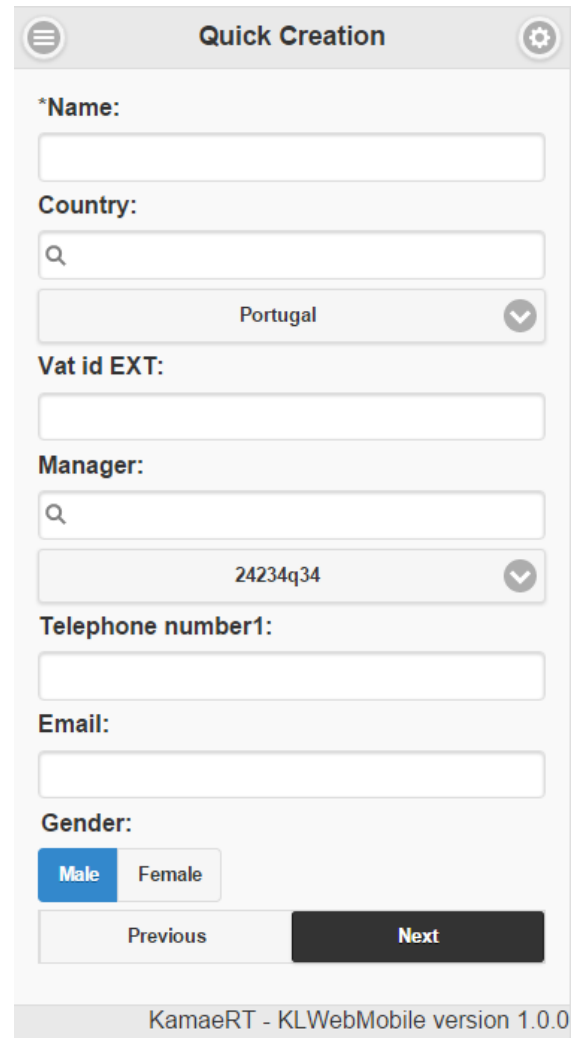
12-08-2015

Archive:

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 91 - Área de dados adicionais coletivos (parte 2)  
- Wizard de clientes



**Quick Creation**

\*Name:

Country:

Portugal

Vat id EXT:

Manager:

24234q34

Telephone number1:

Email:

Gender:

Male Female

Previous Next

KamaeRT - KLWebMobile version 1.0.0

Figura 92 - Área de criação rápida - Wizard de clientes

## Anexo C

### Listagem de bugs reportados

Tabela 20 - Listagem de bugs reportados

ID	Tipo	Rank	Estado	Atribuido a	Titulo
8231	Bug	3	Resolved	Carlos Nogueira	Corrigir funcionalidade de login quando não se tem privilégios de administrador
8733	Bug	3	Resolved	Carlos Nogueira	Data de conclusão na Timesheet a ser preenchida com a data de criação quando é feita a edição de uma tarefa
8734	Bug	3	Resolved	Carlos Nogueira	Impossibilidade de aceder às opções de edição e remoção de tarefas na listagem da timesheet quando o tamanho da descrição ultrapassa a largura do ecrã
8737	Bug	2	Resolved	Carlos Nogueira	Campos de duração, Início e fim marcam horas erradas quando o formato de hora do sistema segue o padrão (AM/PM).
8738	Bug	5	Resolved	Carlos Nogueira	Campo duração fica a vermelho ao carregar no botão de guardar mesmo estando correta, no final da criação de uma nova tarefa (não influencia a execução normal da aplicação).
8806	Bug	3	Resolved	Carlos Nogueira	Bug ao fazer update a uma tarefa na time-sheet quando não era feito o update dos valores de client e matter e case
8868	Bug	3	Resolved	Carlos Nogueira	Ao seleccionar cliente singular e depois seleccionar cliente coletivo, o singular é a opção que fica sempre seleccionada ao avançar no wizard
8869	Bug	2	Resolved	Carlos Nogueira	Erro ao criar tarefa na timesheet quando a tarefa é "Verificação de Faturas" e a opção "incluir em relatórios" não está checada
8870	Bug	3	Resolved	Carlos Nogueira	Limpar campos não funciona nos contactos adicionais coletivos
8871	Bug	3	Resolved	Carlos Nogueira	Campos de Data não estão a assumir data atual
8916	Bug	1	Resolved	Carlos Nogueira	Validações dos campos de data no wizard de cliente não funcionam corretamente
8924	Bug	2	Resolved	Carlos Nogueira	Erro ao criar matter com inserção rápida



8926	Bug	3	Resolved	Carlos Nogueira	Campos Horário, Nome Comercial, Número e Arquivo, Arquivo para cliente coletivo não estão a ser retornados corretamente da BD.
8975	Bug	2	Resolved	Carlos Nogueira	Campos na vista SaveMattersAdditionalData não estão a receber valores corretos
9031	Bug	2	Resolved	Carlos Nogueira	Valores das datas e dos campos "Value hour" em SaveMattersValueTeam não estão a ser carregados
9032	Bug	2	Resolved	Carlos Nogueira	Botão de retroceder no SaveMattersConfirmGet vai para pagina errada quando no processo de edição
9153	Bug	1	Resolved	Carlos Nogueira	Procura por registos nas listagens de Matters, timesheet, notas, matters, deadlines e cases deixa de funcionar ao ser eliminado um dos itens listados
9156	Bug	1	Resolved	Carlos Nogueira	Ao seleccionar um case para edição o cliente seleccionado na combo box não corresponde ao cliente seleccionado durante a criação
9214	Bug	1	Resolved	Carlos Nogueira	Bug na seleção de datas no firefox
9273	Bug	1	Resolved	Carlos Nogueira	Ao retornar a mesma vista quando uma validação falha, certos componentes de uma vista perdem os eventos, impossibilitando a interação.
9275	Bug	3	Resolved	Carlos Nogueira	Ao executar pesquisa com o clique da tecla "Enter" no wizard de casos, onde em vez de ser feita a pesquisa o wizard prossegue para a página seguinte.
9276	Bug	4	Resolved	Carlos Nogueira	Campo reason na parte judicial do wizard de cases esta a desaparecer ao usar uma checkbox para esconder os campos um conjunto de campos
9278	Bug	1	Resolved	Carlos Nogueira	Campos calculados no wizzard de expenses não estão a ser passados para o servidor
9279	Bug	1	Resolved	Carlos Nogueira	Campos unit cost, valor com iva, valor sem iva não estão a ser passados para as vistas
9280	Bug	3	Resolved	Carlos Nogueira	No wizard de expenses depois de criar uma expense ou sair a meio da criação, os dados antigos mantêm-se quando se inicia um novo wizard

9392	Bug	3	Resolved	Carlos Nogueira	Checkboxes previamente checkadas ao entrar numa das vistas de opções são passadas como false para o servidor quando guardadas alterações nas opções disponíveis
9585	Bug	2	Resolved	Carlos Nogueira	Texto das checkboxes no módulo das opções torna-se ilegível em dispositivos de dimensões menores
9587	Bug	3	Resolved	Carlos Nogueira	Internal reference nos wizards de matters e cases não estão a refletir o tipo tipo de referencia no textfield
9625	Bug		Resolved	Carlos Nogueira	Wizard nao permite guardar matter devido a Internal references
9626	Bug	2	Resolved	Carlos Nogueira	Lista de clientes não é atualizada quando se insere um cliente
9627	Bug	3	Resolved	Carlos Nogueira	Aplicação fica com os elementos desestruturados, sendo necessário refresh da página para corrigir
9690	Bug	3	Resolved	Carlos Nogueira	KLWEBm-iPhone: Na pagina inicial, quando se seleciona o item de menu de cliente ou contactos e se volta a pagina inicial os dados do menu desaparecem
9706	Bug	4	Resolved	Carlos Nogueira	Pesquisas retornam null quando um dos campos de um registo é vazio
9816	Bug	4	Resolved	Carlos Nogueira	Termos não estão corretamente aplicados em certas partes da aplicação
9817	Bug	1	Resolved	Carlos Nogueira	Datas onde a "culture" é diferente de "pt-PT" não estão a ser guardadas
9505	Bug	1	Resolved	Carlos Nogueira	KLWEBm-iPhone: Versionamento incorreto.
9806	Bug	1	Resolved	Carlos Nogueira	KLWEBm: Browser vai abaixo quando se edita um cliente e se tenta listar o client type
9188	Bug	2	Resolved	Carlos Nogueira	KLWEBm-iPhone: Matters/Cases: Error Loading Page
9184	Bug	3	Resolved	Carlos Nogueira	KLWEBm-iPhone: Time-Sheet >> My Tasks: Botões 'Description' e 'Folder Reference' não provocam qualquer ação
9185	Bug	3	Resolved	Carlos Nogueira	KLWEBm-iPhone: Time-Sheet >> My Tasks: Não é possível editar tarefas.
9229	Bug	3	Resolved	Carlos Nogueira	KLWEBm-iPhone: Autenticação não está a verificar password nem código de segurança

9509	Bug	3	Resolved	Carlos Nogueira	KLWEBm-iPhone: Na lista de clientes, o botão de adicionar cliente mantêm-se visível quando se faz scroll
9524	Bug	3	Resolved	Carlos Nogueira	KLWEBm-iPhone: Entrada 'Contacts' do Menu Lateral não funciona
9525	Bug	3	Resolved	Carlos Nogueira	KLWEBm-iPhone: Pesquisas incorretas
9528	Bug	3	Resolved	Carlos Nogueira	KLWEBm-iPhone: Não é possível editar contactos
9843	Bug	3	Resolved	Carlos Nogueira	KLWEBm: Após criação de cliente o botão para adicionar cliente na client list deixa de provocar qualquer ação
9846	Bug	3	Resolved	Carlos Nogueira	KLWEBm: Comboboxes que listam clientes, <i>dossiers</i> , processos, etc devem ser populadas de forma diferente
9186	Bug	4	Resolved	Carlos Nogueira	KLWEBm-iPhone: Botões de pesquisa são problema de usabilidade
9508	Bug	4	Resolved	Carlos Nogueira	KLWEBm-iPhone: Após logout, caso se falhe a autenticação a página fica bloqueada não sendo possível autenticar sem fazer refresh à página.
9523	Bug	4	Resolved	Carlos Nogueira	KLWEBm-iPhone: Menu Lateral tem a entrada "Time-Sheet" repetida
9527	Bug	4	Resolved	Carlos Nogueira	KLWEBm-iPhone: Mensagem incorreta na confirmação da atualização das opções de clientes
9807	Bug	4	Active	Carlos Nogueira	KLWEBm: Desformatação na Home
9808	Bug	4	Resolved	Carlos Nogueira	KLWEBm: É possível criar Matters e Cases com contextos inconsistentes
9183	Bug	5	Resolved	Carlos Nogueira	KLWEBm-iPhone: Time-Sheet >> My Tasks: apagar uma tarefa deve mostrar mensagem de aviso
9187	Bug	5	Resolved	Carlos Nogueira	KLWEBm-iPhone: Deve ser 'Quick Creation' e não 'Rapid Creation'
9506	Bug	5	Resolved	Carlos Nogueira	KLWEBm-iPhone: Deve ser possível trocar a imagem de segurança na autenticação
9507	Bug	5	Resolved	Carlos Nogueira	KLWEBm-iPhone: Após logout deverá ser mostrada uma página de autenticação nova