

Universidade do Minho - Escola de Engenharia
Engenharia de Aplicações

Aplicação *Home4All*

Autores:



Diogo Costa (A78034)



Mafalda Nunes (A77364)



Marco Silva (A79607)



Patrícia Barreira (A79007)

Versão 1.0
1 de Julho de 2019

Resumo

O projeto **Home4All** constitui um *website* que pretende oferecer funcionalidades relacionadas com o setor imobiliário, nomeadamente a compra e/ou venda de imóveis.

Desta forma, começou-se por efetuar uma análise de requisitos na qual foram identificadas todas as funcionalidades do projeto, os utilizadores alvo e descritas as principais tarefas que compõem o sistema.

De seguida, passou-se para a modelação do sistema, onde foi elaborado o modelo de domínio, o *mockup* da interface, com consequente avaliação da usabilidade.

A terceira etapa, envolveu a implementação do projeto propriamente dito, através de uma abordagem baseada em modelos.

Por fim, foi realizado o *deploy* da infraestrutura final em máquinas físicas, as quais foram submetidas a testes de carga com o objetivo de avaliar a performance do sistema.

Conteúdo

1	Introdução	4
2	Contextualização	5
3	Análise de Requisitos	6
3.1	Utilizadores	6
3.1.1	Perfis de Utilizador	6
3.1.2	<i>Personas</i>	7
3.2	Requisitos do Sistema	9
3.2.1	Requisitos Funcionais	9
3.2.1.1	Registo e Autenticação	9
3.2.1.2	Pesquisa	9
3.2.1.3	Cliente	10
3.2.1.4	Anunciante	11
3.2.1.5	Administrador	12
3.2.2	Requisitos Não Funcionais	13
3.3	Análise e Modelação de Tarefas	13
3.3.1	Sem a aplicação <i>web</i> Home4All	13
3.3.2	Com a aplicação <i>web</i> Home4All	13
4	Domínio do Problema	17
5	Concepção (Prototipagem)	19
5.1	<i>Mockups</i>	19
5.1.1	Página Principal (sem login)	21
5.1.2	Pesquisa de Imóvel	22
5.1.3	Consulta de imóvel	28
5.1.4	Comparação entre imóveis	31
5.1.5	Registo	32
5.1.6	<i>Login</i>	33
5.1.7	Página Principal (com <i>login</i>)	34
5.1.8	Dados do Perfil	35
5.1.9	Favoritos	37
5.1.10	Configuração de Notificações	38
5.1.11	Anunciar Novo Imóvel	40
5.2	Técnicas de Avaliação da Usabilidade	41
5.2.1	Método empírico - Teste de usabilidade	42
5.2.1.1	Objetivos	42
5.2.1.2	Tarefas a realizar	42
5.2.1.3	Execução do estudo	43
5.2.1.4	Processamento dos dados	43
5.2.2	Método analítico - Inspeção por peritos	43
5.2.2.1	Objetivos	43
5.2.2.2	Descrição dos utilizadores	43
5.2.2.3	Tarefas a realizar	43
5.2.2.4	Execução do estudo	43
5.2.2.5	Processamento dos dados	44

6 Desenvolvimento	45
6.1 <i>Platform-independent model</i> (PIM)	45
6.2 Escolha das tecnologias	47
6.3 <i>Platform-specific model</i> (PSM)	47
6.4 Implementação	51
6.4.1 Dificuldades encontradas e decisões mais importantes	51
6.4.1.1 <i>Frontend</i>	51
6.4.1.2 <i>Backend</i>	51
6.4.2 Funcionalidades implementadas	52
6.4.3 Interface de utilizador	52
7 Deployment	54
8 Carga aplicacional	55
8.1 Povoamento da Base de Dados	55
8.2 Testes de Carga	55
8.2.1 Ferramenta utilizada	55
8.2.2 Parametrização dos Testes	57
8.2.3 Dificuldades	58
8.2.4 Resultados	58
8.2.4.1 Arquitetura com um Servidor Aplicacional	59
8.2.4.2 Arquitetura com dois Servidores Aplicacionais	67
8.2.4.3 Conclusões	73
9 Testes de usabilidade da interface	74
10 Conclusões	75

1 Introdução

O projeto **Home4All** é constituído por um *website* de imóveis onde os seus utilizadores podem consultar imóveis, ou mesmo comprar e arrendar. Este é implementado como sendo uma aplicação *web* e, como tal, pode ser usado em qualquer dispositivo.

De facto, para atingir este objetivo são usadas diversas ferramentas tecnológicas, que se integram de forma a oferecer um sistema coeso. Desta forma, a linguagem de base escolhida foi o Java e, por isso, são utilizadas *frameworks* como o **Hibernate**, o **JEE** e o **Vue.js**, para a camada de dados, aplicacional e de apresentação, respetivamente.

Primeiramente, é descrita uma contextualização do projeto, onde se pretende dar resposta ao porquê da necessidade de uma ferramenta como esta nos dias de hoje. De seguida, descreve-se o domínio do problema e quais os constituintes deste. Posteriormente, tem-se uma análise detalhada de todos os requisitos inerentes ao projeto, quer ao nível do utilizador como do sistema. A próxima etapa envolve a conceção de um *mockup* da interface, assim como a primeira fase de testes de usabilidade. Além disso, tem-se o desenvolvimento propriamente dito, com especial foco nos problemas encontrados e possíveis melhorias em iterações futuras do projeto. São ainda apresentados os resultados de testes de carga efetuados à aplicação desenvolvida, de forma a avaliar o seu desempenho em situações de *stress*. Por fim, tem-se uma nova fase de avaliação da usabilidade, onde o objetivo é concluir os aspectos positivos e negativos do projeto.

2 Contextualização

O mercado imobiliário digital já existe à algum tempo e tem vindo a tornar-se a norma no que toca à forma como as pessoas adquirem/arrendam imóveis. Efetivamente, à uns anos atrás o normal seria que as pessoas se dirigessem a uma imobiliária física para perceber as suas possibilidades. No entanto, hoje em dia o primeiro instinto é, normalmente, usar os meios digitais para realizar a pesquisa. De facto, não é à toa que assim o é, pois tem-se uma maior rapidez nas pesquisas realizadas, tem-se acesso instantâneo a imagens e dimensões dos imóveis, assim como um possível contacto para interagir com o anunciante.

Contudo, apesar do mercado estar bastante explorado, existe alguns aspectos que podem ser melhorados. Nomeadamente, no que toca ao arrendamento de apartamentos/quartos, rapidamente se percebe que existe uma quantidade substancial de informação dispersa ou, em vários *sites* especializados, redes sociais ou mesmo através dos *emails*.

Mais ainda, existe normalmente muitas possibilidades que correspondem ao que o cliente pretende, tornando-se complicado para o utilizador cobrir essa panóplia de opções. Além disso, a maior parte dos sistemas apenas funcionam a partir de interações diretas com o utilizador, ou seja, através de pesquisas, comparações ou mesmo adicionar aos favoritos. No entanto, achou-se necessário ser o próprio sistema a fazer algum desse trabalho e proativamente tomar a iniciativa de interagir com o utilizador, como por exemplo através de notificações. Mais ainda, dado que a compra de um imóvel é sempre uma decisão de peso, achou-se que os serviços existentes não oferecem as funcionalidades necessárias para estreitar a relação existente entre o cliente e o anunciante, de forma a dar uma maior segurança na altura de realizar o investimento.

O projeto **Home4All** surge como um projeto ambicioso, que tem como objetivo dar resposta a alguns dos problemas mencionados anteriormente. Para tal, é importante que este incorpore um sistema de caracterização de imóveis bastante expressivo, de forma a permitir uma maior especificidade na pesquisa por parte do cliente; é importante que permita configurar notificações para imóveis que o cliente esteja interessado; que permita interagir de forma direta com o anunciante; que permita anunciar imóveis de forma simples e rápida e que tenha um *design* renovado e com elevado grau de usabilidade.

Assim sendo, este documento tem como objetivo documentar todo o processo que leva o Home4All a implementar algumas destas funcionalidades. De realçar que o objetivo passa sempre por melhorar a forma como os utilizadores realizam compras de imóveis *online*.

3 Análise de Requisitos

3.1 Utilizadores

3.1.1 Perfis de Utilizador

Para melhor analisar todos os aspetos importantes relativamente ao design da interface da plataforma, foram estudadas os principais grupos de possíveis utilizadores da plataforma tendo em conta fatores como a sua faixa etária ou até as suas habilitações.

De seguida serão apresentados os grupos de utilizadores juntamente com todos os fatores que os caracterizam.

O primeiro grupo de utilização a ser analisado será representativo de todos os estudantes que pretendem encontrar um outro estudante com o qual dividir o imóvel. Desta forma, a ação a realizar na plataforma será a publicação de um quarto para arrendar, sendo assim considerado um tipo de utilização direto. Mais ainda, existem outros métodos através dos quais estudantes podem encontrar outros estudantes para a divisão do imóvel, não deixando ainda de ser a mais cómoda. Analisando agora a frequência de utilização da plataforma, os meses de iniciação do período letivo serão definitivamente o momento de maior utilização da plataforma, tornando-os utilizadores intermitentes.

Passando agora a uma análise mais focada no utilizador em si, espera-se que estes apresentem níveis de perícia de intermédio a elevado, tendo em conta a sua faixa etária (dos 18 aos 23 anos). Mais ainda, tendo em vista o acesso ao ensino superior destes utilizadores, deduz-se que estes tenham já concluído o 12º ano de escolaridade. Desta forma, estes têm um nível de formação médio.

Desta forma, encontrando-se realizada a análise completa deste tipo de utilizadores, conclui-se que a interface com o utilizador deverá apresentar alguns atalhos por forma a agilizar a execução das tarefas na plataforma mas sem nunca deixar de informar sobre o estado de completude ou finalização de uma ação.

Passando agora ao seguinte grupo de utilização, consideram-se todos os utilizadores que pretendem encontrar um imóvel vazio para habitar durante todo o ano. Esta ação representa também um tipo de utilização direta mas agora com utilizadores com características diferentes.

Assim, são integrados neste grupo todos os utilizadores de média idade (entre os 30 e os 50 anos) e com um nível de experiência médio. Espera-se ainda que estes revelem destreza com as mais recentes tecnologias web. O nível de formação esperado será intermédio (espera-se a conclusão do 9º ao 12º ano de escolaridade) devido não só a sua escolaridade mas também à experiência adquirida no mercado de trabalho.

Tendo em conta todos os fatores analisados, para esta classe de utilizadores os atalhos não se revelarão tão importantes para a sua satisfação mas sim a informação do estado das tarefas a serem realizadas. Assim, a apresentação do estado destas tarefas será essencial para estes utilizadores.

Passando agora a uma classe de mais sensível no que toca à tecnologia, irão ser analisados de seguida os utilizadores de idade já mais avançada. Estes requerem especial atenção uma vez que representam um número significativo de arrendamentos de quartos bem como a atenção necessária durante o desenvolvimento de uma interface a ser utilizada por estes utilizadores. Esta classe representará todos os indivíduos que apresentam baixos níveis de perícia mas também muito pouco familiarizados com as mais básicas tecnologias. Assim, todas as ações a realizar pela plataforma deverão ser o mais intuitivas possível. Para além disso, a apresentação do progresso de realização da tarefa em execução será determinante para manter a confiança do utilizador no sistema. Por forma a transmitir um sinal de confiança em todas as interações a realizar, possibilitar o retrocesso nas diversas fases de execução de uma tarefa será determinante.

Para concluir, existirão ainda os administradores do sistema, responsáveis pelo bom funcionamento da plataforma. Estes apresentam níveis de perícia elevados pelo que todas as tarefas a realizar por estes deverá ser o mais rápida e eficiente possível.

Tendo em conta toda a análise feita anteriormente dos mais diversos tipos de possíveis utilizadores, conclui-se que para se atingir um nível de satisfação alto de todos, a interface deverá

apresentar todas as ações a realizar da forma mais simples possível mas, por outro lado, disponibilizar alguns atalhos por forma a agilizar a utilização de utilizadores mais experientes.

3.1.2 *Personas*

Para melhor conhecer e caracterizar os utilizadores da plataforma, foram criadas algumas pessoas representativas dos principais tipos de utilização.

Estudante que pretende publicar um quarto para arrendar:

Leonor tem 22 anos e é estudante do curso de gestão da Universidade do Minho. Natural do Porto, fez as viagens entre a sua casa e a universidade todos os dias durante os 3 anos em que esteve a frequentar a licenciatura. Apesar destas viagens ocuparem grande parte do seu dia, à 3 anos atrás não se sentiu preparada para aventurar-se sozinha num apartamento em Braga. No entanto, com a decisão de continuar os estudos frequentando o mestrado de Finanças, a questão de ir viver para Braga voltou a colocar-se. Desta vez, achou que fazer a mudança seria a melhor opção uma vez que teria mais disponibilidade para se dedicar aos estudos.

Desta forma, a Leonor já contactou com um senhorio em relação a um quarto que este tem disponível. Contudo, como o quarto tem capacidade para duas pessoas o senhorio só aceita se a Leonor arranjar mais alguém com quem partilhar o quarto. Assim sendo, ela sabe que tem que conseguir arranjar uma estudante do sexo feminino que esteja à procura de um quarto com casa de banho partilhada, assim como sala (com televisão), cozinha e Internet.

A Leonor quer encontrar uma parceira o mais depressa possível e acha que através da Internet seja a maneira mais eficaz. Assim, ela espera conseguir comunicar a sua pretensão de maneira extremamente rápida e receber *feedback* imediato de possíveis interessados. De notar que apesar de ter um computador, ela não está sempre com ele e que o melhor era receber notificações no seu telemóvel, de maneira a conseguir manter-se atualizada.

Estudante que pretende arrendar quarto para o próprio:

O Manuel é um jovem de 18 anos. Bastante organizado e metódico, concilia os estudos e as saídas com os amigos de forma a tirar o melhor partido dos dois. O que lhe dá mais prazer é conversar com os amigos durante horas e horas. Agora que acabou de entrar na universidade quer, de alguma forma, continuar com esses hábitos sociais, mas nenhum dos seus amigos entrou na sua universidade. A sua mãe, de forma a ajuda-lo a ambientar-se mais facilmente, sugeriu que partilhasse casa com outras pessoas que também fossem estudantes. Contudo, quer que o seu filho partilhe casa com pessoas responsáveis de forma a que ele mantenha as boas notas tal como no secundário. Portanto, acha que no máximo devia partilhar casa com mais uma ou duas pessoas e de forma a que o filho tenha a sua privacidade, o quarto tem que ser só para ele e com WC, obrigatoriamente.

O Manuel daqui a 1 semana já começa a universidade, portanto a disponibilidade do imóvel terá que ser imediata. Para além disso, os pais querem que o imóvel já esteja mobilado e que esteja na zona da universidade para o filho não depender de transportes públicos e terem despesas com isso.

O seu primo João avisou-o que vai ser bastante complicado e cansativo pesquisar em todas as plataformas de imóveis. Que vai chegar a um ponto, que já não sabia em que site viu o imóvel que lhe interessava, e que tinha que andar a saltar entre sites para comparar os vários imóveis. Portanto, o Manuel gostava de ter uma visão global e simultânea de todos os quartos que lhe interessam para que a escolha possa ser tomada rapidamente e de uma forma menos exaustiva.

Homem de família que pretende arrendar apartamento completo:

O António é um homem de 35 anos, casado à 13 anos com a Manuela. Juntos, têm dois filhos, o João e a Catarina, com 5 e 7 anos respetivamente. O António e a Manuela são muito trabalhadores, o que contribuiu para o aumento das suas capacidades económicas. Assim, pretendem deixar o seu apartamento e encontrar, de forma bastante simples, uma casa com boa qualidade, onde possam

viver os 4 membros da família confortavelmente. Para além disso, o preço deverá ser adequado à sua capacidade financeira.

O António não pretende recorrer a uma imobiliária porque gosta de ser o próprio a analisar as várias propostas abrangidas pelos seus requisitos. Ele também procura diminuir o custo do arrendamento, ao negociar diretamente com o proprietário, uma vez que deixa de existir a taxa de lucro da imobiliária. Por fim, salienta-se que o António gosta de efetuar pesquisas na Internet, devido à facilidade de utilização e rapidez com que as respostas são obtidas. Contudo, existem imensos *websites* para venda e arrendamento de imóveis, sendo que alguns deles são complicados de utilizar.

A necessidade de mudança de habitação, por parte da família do António, não é imediata. Assim, este pretende que, caso não encontre o que procura, seja informado sobre novos imóveis que surjam com as características por si desejadas. Também seria do seu agrado se conseguisse comunicar de forma privada com o proprietário através do *website*, para colocar questões gerais sobre a habitação, antes de fornecer alguma forma de contacto mais definitiva. Assim, consegue evitar futuros contactos mais incómodos por parte do proprietário para, por exemplo, anúncio de outras habitações disponíveis, que possam não ser do seu interesse. Por fim, tanto os filhos do António como o seu trabalho o interrompem várias vezes durante o dia, pelo que deve existir um sistema rápido de marcação de páginas, para posterior consulta.

Idoso que pretende vender a sua antiga casa:

José Fernandes é um habitante de Braga, que se encontra a viver na zona centro da cidade de Braga. Tem 73 anos, encontra-se viúvo, tem apenas a 3^a classe, é um polícia aposentado e reside junto da Avenida Central em Braga. Uma pessoa de poucas festas, todos os dias acorda por volta das 7:30 da manhã para levar a passear o seu mais fiel amigo Rex, o seu cão. Enquanto isso, aproveita para parar na padaria mais próxima da sua habitação onde todos os dias toma o seu pequeno almoço e coloca a conversa em dia com os seus amigos das proximidades. Apesar da sua idade, o Sr José gosta de se manter informado pelo que na volta para casa compra sempre o jornal informativo da região, o Diário do Minho. Para além disso, sendo um aficionado do SC. Braga, o jornal desportivo não pode faltar também em alturas próximas de jogos do seu clube. Após esta rotina matinal, o Sr José desloca-se de autocarro até à casa do seu filho João que vive na periferia de Braga ajudando-o a tratar do seu quintal e mantendo-se assim ocupado durante o resto do dia.

Tendo em conta a sua idade já avançada, as viagens para a casa do seu filho começam a tornar-se cansativas. Ainda assim, é uma pessoa que sempre gostou de aprender e utiliza frequentemente a rede social *Facebook* para comunicar com amigos mais distantes. Desta forma, o Sr José gostaria de anunciar a sua casa para venda uma vez que num futuro próximo se mudará definitivamente para a casa do seu filho.

Assim, o Sr José pretende anunciar a sua casa para venda na plataforma. Para além disso, gostaria de receber notificações sobre possíveis interações com o anúncio publicado (comentários, perguntas dirigidas ao vendedor, etc).

Grupo que pretende arrendar imóvel em conjunto:

Joana, Adriana e Maria têm 18 anos e acabam de saber que entraram na mesma faculdade. Uma vez que já se conhecem à algum tempo e de forma a reduzir as despesas decidiram procurar apartamento todas juntas. A Joana deu a ideia de ficarem num mesmo quarto e dividirem o apartamento com outras pessoas de forma a viverem com pessoas novas. Mas só poderiam ficar no mesmo quarto se este fosse espaçoso e tivesse casa de banho privativa para as 3. A Maria gosta de ter certeza que estão a tomar a decisão acertada e mais em conta e portanto quer comparar a opção de ficarem num quarto partilhado pelas três ou em quartos individuais.

3.2 Requisitos do Sistema

3.2.1 Requisitos Funcionais

Para se especificar os requisitos funcionais da aplicação a desenvolver de forma organizada, optou-se por os dividir em 4 módulos principais, sendo estes os que se apresentam de seguida:

1. **Registo e Autenticação** – envolve todos os requisitos associados ao processo de registo e autenticação no *website*;
2. **Pesquisa** – inclui os requisitos diretamente relacionados com o processo de pesquisa de imóveis;
3. **Cliente** – descreve os requisitos funcionais associados ao utilizador, quando este pretende comprar ou arrendar um imóvel, para o próprio viver;
4. **Anunciante** – inclui os requisitos funcionais associados ao utilizador, quando este pretende vender ou arrendar um imóvel, para outros viverem.

3.2.1.1 Registo e Autenticação

1. O utilizador, ao registar-se, terá de preencher um formulário de registo.
 - (a) O sistema deve apresentar um formulário de registo com os seguintes campos: e-mail, password, primeiro e último nome, género, profissão, data de nascimento e contacto.
 - (b) O sistema deve apresentar, para o preenchimento do campo profissão, as seguintes alternativas: estudante, trabalhador-estudante, trabalhador, desempregado e reformado.
 - (c) O sistema deve garantir que o preenchimento do contacto seja de cariz opcional.
 - (d) O sistema deve permitir o registo utilizando as seguintes plataformas: Facebook ou Google.
2. O utilizador só não precisa de estar autenticado para efetuar pesquisas e comparações entre imóveis.
3. O utilizador deve autenticar-se utilizando o e-mail e *password* ou a conta externa definidos no momento do registo.

3.2.1.2 Pesquisa

1. O utilizador deve poder pesquisar pelos seguintes critérios: distrito, cidade e rua.
 - (a) O sistema deve apresentar, para o campo “cidade”, uma lista com as cidades existentes.
 - (b) O sistema deve apresentar, para o campo “freguesia”, uma lista com as freguesias existentes, perante a escolha da cidade.
 - (c) O sistema deve permitir o preenchimento do campo “rua”, que é auxiliado por *auto-complete*. Caso algum dos campos anteriores esteja preenchido, as ruas apresentadas serão filtradas para respeitar essas escolhas.
2. O utilizador deve poder filtrar as pesquisas efetuadas.
 - (a) O sistema deve apresentar uma lista com os diferentes tipos de imóveis, entre os quais, quartos, apartamentos e vivendas.
 - (b) O sistema deve apresentar uma lista das tipologias disponíveis: T0, T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T10+. A escolha por omissão deverá aceitar todas as tipologias.
 - (c) O sistema deve possibilitar a escolha entre anúncios para venda e/ou aluguer. A escolha por omissão deverá apresentar os imóveis tanto para venda como aluguer.

- (d) O sistema deve possibilitar a escolha de um intervalo de preços pretendido pelo utilizador. Os valores por omissão serão o mínimo/máximo existente para a pesquisa em questão.
 - (e) O sistema deve possibilitar a introdução do número de clientes envolvidos no negócio. O valor por omissão será uma pessoa.
 - (f) O sistema deve permitir a escolha do número e tipos de quartos pretendidos (individual, casal ou múltiplo). No caso de quartos múltiplos deve ser indicado o número de pessoas interessadas.
 - (g) O sistema deve possibilitar a escolha entre imóvel partilhado, não partilhado ou indiferente. O valor por omissão é “indiferente”.
 - (h) O sistema deve possibilitar a indicação das características de cliente com os quais se pretende partilhar o imóvel. Na prática estas podem ser: profissão, fumador ou não fumador e existência de animais de estimação.
 - (i) O sistema deve permitir a escolha entre casa de banho partilhada e/ou não partilhada. Por omissão são incluídas todas as possibilidades.
 - (j) O sistema deve permitir a escolha entre imóvel mobilado e/ou não mobilado. Por omissão é incluído mobilado e não mobilado.
 - (k) O sistema deve permitir a escolha entre acesso total ao imóvel ou apenas a algumas divisões. Por defeito tem-se em conta ambas as opções.
 - (l) O sistema só deve tornar acessível a seleção de filtros disponíveis na área geográfica abrangida pela pesquisa.
3. O utilizador deve poder escolher a forma como os resultados da pesquisa são apresentados.
- (a) O sistema deve permitir a seleção do tipo de ordenação dos resultados. As possibilidades são:
 - Dos preços mais elevados para os mais baixos.
 - Dos preços mais baixos para os mais elevados.
 - Dos imóveis mais distantes para os mais próximos da pesquisa realizada.
 - Dos imóveis mais próximos para os mais distantes da pesquisa realizada. (Opção por omissão)
 - Da data de publicação mais antiga para a mais recente.
 - Da data de publicação mais recente para a mais antiga.
 - Do anunciante que fez *login* à mais tempo para o ativo à menos tempo.
 - Do anunciante que fez *login* à menos tempo para o ativo à mais tempo.

3.2.1.3 Cliente

1. O cliente deve ser capaz de colocar questões aos anunciantes de forma pública.
 - (a) O sistema deve apresentar uma secção, na página dos imóveis, de perguntas e respostas, que são visíveis para todos os utilizadores.
 - (b) O sistema deve permitir ao utilizador inserir uma nova questão. Após o preenchimento da mesma esta deve ser enviada para o anunciente do imóvel respetivo.
 - (c) O sistema deve tornar a pergunta pública assim que o anunciente responder.
 - (d) O sistema deve notificar o cliente que realizou a pergunta, quando esta for respondida pelo anunciente.
2. O cliente deve ser capaz de colocar questões aos anunciantes de forma privada.
 - (a) O sistema deve disponibilizar uma opção no menu dos anunciantes que permita que um cliente possa enviar mensagens privadas.

- (b) O sistema deve apresentar uma janela com a conversação realizada até ao momento, quando o cliente inicia uma conversa com o anunciante.
 - (c) O sistema deve notificar os clientes sempre que estes receberem mensagens dos anunciantes.
3. O cliente deve poder receber notificações em relação a imóveis que cumpram certas características. Adicionalmente, pode ter notificações para diferentes conjuntos de características.
- (a) O sistema deve apresentar um menu onde o utilizador possa configurar as notificações que pretende receber.
 - (b) O sistema deve apresentar uma lista dos vários tipos de configurações que o utilizador possui. O utilizador deve ser capaz de adicionar, alterar ou remover entradas da lista.
 - (c) O sistema deve redirecionar o utilizador para uma vista mais detalhada das notificações caso uma entrada da lista seja selecionada.
 - (d) O sistema deve redirecionar o utilizador para uma página onde pode configurar o tipo de imóvel para o qual quer ser notificado, caso este pretenda adicionar um novo tipo de notificação. As opções disponíveis devem corresponder às existentes no módulo da Pesquisa.
 - (e) O sistema de notificações só deve ser despoletado quando o cliente fica *online*.
4. O cliente deve poder comparar imóveis diretamente.
- (a) O sistema deve permitir que todos os imóveis tenham a opção de adicionar o mesmo a um sistema de comparação de imóveis.
 - (b) O sistema apenas deve permitir que 3 imóveis possam ser comparados simultaneamente.
 - (c) O sistema deve ter visível a opção de redirecionar o cliente para o sistema de comparação quando pelo menos dois imóveis foram adicionados para comparação.
 - (d) O sistema deve apresentar a página de comparação como sendo composta por colunas. Cada coluna deve corresponder a um imóvel em particular. Adicionalmente, campos de características iguais devem estar alinhados horizontalmente para ser de fácil consulta, sendo estas: fotografias, tipo de imóvel, tipologia, área, se está mobilado, datas de disponibilidade, preço, localização, se estão incluídos de outros gastos na renda, forma de contacto e *link* para ver página completa do imóvel.
5. O cliente deve poder realizar pesquisas conforme descrito no módulo Pesquisa.

3.2.1.4 Anunciante

1. O anunciante deve poder inserir um novo imóvel, através do preenchimento de um formulário.
- (a) O sistema deve apresentar as seguintes opções: aluguer, venda ou ambos.
 - (b) O sistema deve permitir, para o caso de aluguer, as opções de alugar quarto(s) ou imóvel completo.
 - (c) O sistema deve permitir, para o caso de venda, a opção de imóvel completo.
 - (d) O sistema deve permitir, para o caso de venda e aluguer, as opções apresentadas acima para os dois casos.
 - (e) O sistema deve permitir indicar, para o caso de imóvel completo, se este é um apartamento ou vivenda, tipologia, área, se está mobilado, se a cozinha está equipada, disponível a partir de quando e preço.
 - (f) O sistema deve permitir, para o caso de quarto(s), a opção de indicar quantos quartos. Para cada um deverá ser solicitada a introdução do seu tipo (individual, duplo, casal, múltiplo ou indiferente), a área, se é mobilado, se a casa de banho é partilhada, a partir de quando está disponível e o preço.

- (g) O sistema deve solicitar indicação, para o caso de aluguer, de que despesas (água, eletricidade, gás, TV cabo, telefone, Internet, condomínio e serviços de limpeza) são ou não incluídas na renda mensal.
 - (h) O sistema deve solicitar a introdução de informações específicas sobre a habitação: número máximo de pessoas aceites e número de casas de banho existentes na habitação.
 - (i) O sistema deve solicitar, em caso de aluguer de quarto(s), dados sobre os equipamentos ou divisões que podem ou não ser utilizados pelos inquilinos: cozinha, ar condicionado, aquecimento central, micro-ondas, fogão, frigorífico, máquina de lavar loiça, máquina de lavar roupa, televisão, Internet, elevador, ginásio, piscina, garagem.
 - (j) O sistema deve apresentar a possibilidade de escrever um texto ou descrição acerca do imóvel.
 - (k) O sistema deve apresentar uma área para indicação da morada da habitação.
 - (l) O sistema deve permitir indicar características de quem se procura para o imóvel: género (masculino, feminino ou indiferente), profissão (estudante, trabalhador-estudante, trabalhador, desempregado ou indiferente), possibilidade de trazer animais de estimação (sim ou não), possibilidade de ser fumador (sim ou não) e restrições de idade mínima e máxima.
 - (m) O sistema deve permitir a indicação de se já existem animais de estimação e fumadores na habitação, bem como a profissão dos arrendatários atuais, caso se trate do aluguer de quarto(s).
2. O anunciante poderá consultar dados estatísticos sobre as suas vendas/arrendamentos.
 - (a) O sistema deve permitir consultar o número de vendas absolutas efetuadas num determinado período.
 - (b) O sistema deve permitir consultar a percentagem de vendas em relação ao disponibilizado por mês.
 3. O anunciante será capaz de receber e responder a questões de clientes, quer de forma pública como privada.
 - (a) O sistema deve possibilitar o acesso às mensagens públicas e privadas a todo o momento.
 - (b) O sistema deve indicar claramente quais as mensagens públicas e privadas.
 - (c) O sistema deve associar as mensagens públicas à página detalhada do imóvel em questão, acessível por todos os utilizadores.
 - (d) O sistema deve restringir o acesso apenas ao *chat* de mensagens privadas.
 4. O anunciante deve poder realizar pesquisas conforme descrito no módulo Pesquisa.

3.2.1.5 Administrador

1. O administrador deve poder remover anúncios de imóveis, utilizadores, bem como questões e respostas públicas.
 - (a) O sistema deve possibilitar acesso a listas distintas de imóveis e utilizadores.
 - (b) O sistema deve permitir, para a lista de imóveis, filtrar de acordo com o nome, remover o respetivo anúncio ou consultar os comentários públicos associados.
 - (c) O sistema deve permitir, para a lista de comentários públicos de cada imóvel, filtrar por conteúdo e remover comentários específicos.
 - (d) O sistema deve permitir, para a lista de utilizadores, filtrar de acordo com o nome e bloquear o respetivo utilizador.

3.2.2 Requisitos Não Funcionais

3.3 Análise e Modelação de Tarefas

3.3.1 Sem a aplicação web Home4All

Na atualidade, já existem vários *websites*, mais gerais ou específicos para imóveis, que disponibilizam algumas das principais funcionalidades que o Home4All pretende implementar. Assim, podem-se destacar as seguintes tarefas realizadas pelos utilizadores atualmente, quando recorrem a esses *websites* para tratar de assuntos imobiliários:

- Anunciar imóvel;
- Anunciar procura de imóvel;
- Pesquisar imóvel com determinadas características;
- Configurar notificações para novos imóveis com determinadas características pretendidas;
- Contactar anunciante (através do *website*, para e-mail ou telemóvel).

Para descrever o fluxo do processo principal, de anúncio, pesquisa e compra/aluguer de imóvel, bem como o encadeamento de tarefas associadas, optou-se por apresentar uma análise hierárquica destas tarefas através de um CTT (*ConcurTasks Tree*), apresentado na figura 1.

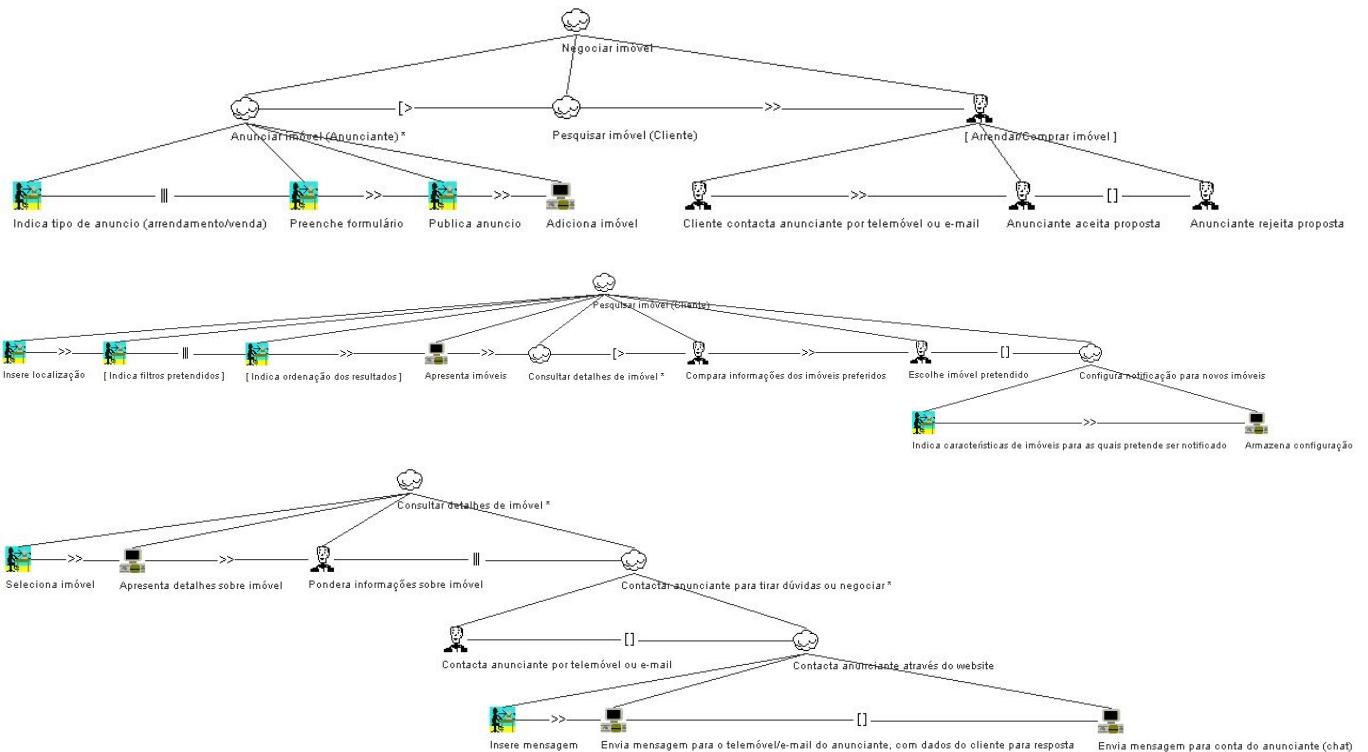


Figura 1: Modelo de tarefas para negociar um imóvel, sem o *website* Home4All.

3.3.2 Com a aplicação web Home4All

De forma a compreender melhor as funcionalidades disponibilizadas para cada tipo de utilizador quando se utiliza o Home4All, desenvolveu-se o diagrama de *use cases* apresentado na figura 2. Salienta-se que o ator **Utilizador** corresponde ao utilizador comum do *website*, que pode ser um anunciante e um cliente simultaneamente.

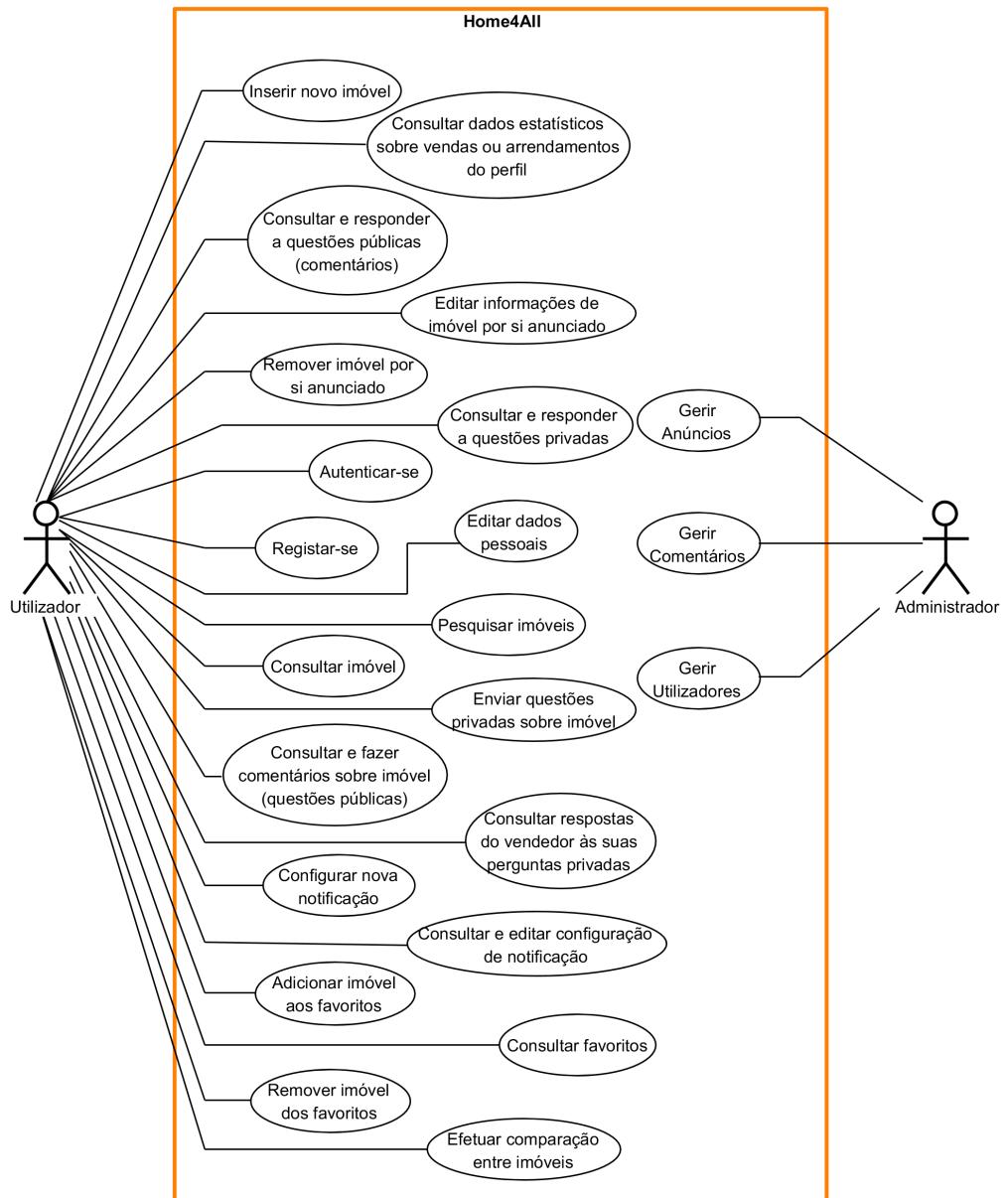


Figura 2: Diagrama de *use cases* do sistema Home4All

Para além disso, construiu-se um modelo de tarefas geral, que descreve todo o processo de anúncio, pesquisa e compra/aluguer de imóvel, bem como o encadeamento de tarefas associadas. Este modelo é apresentado na figura 3.

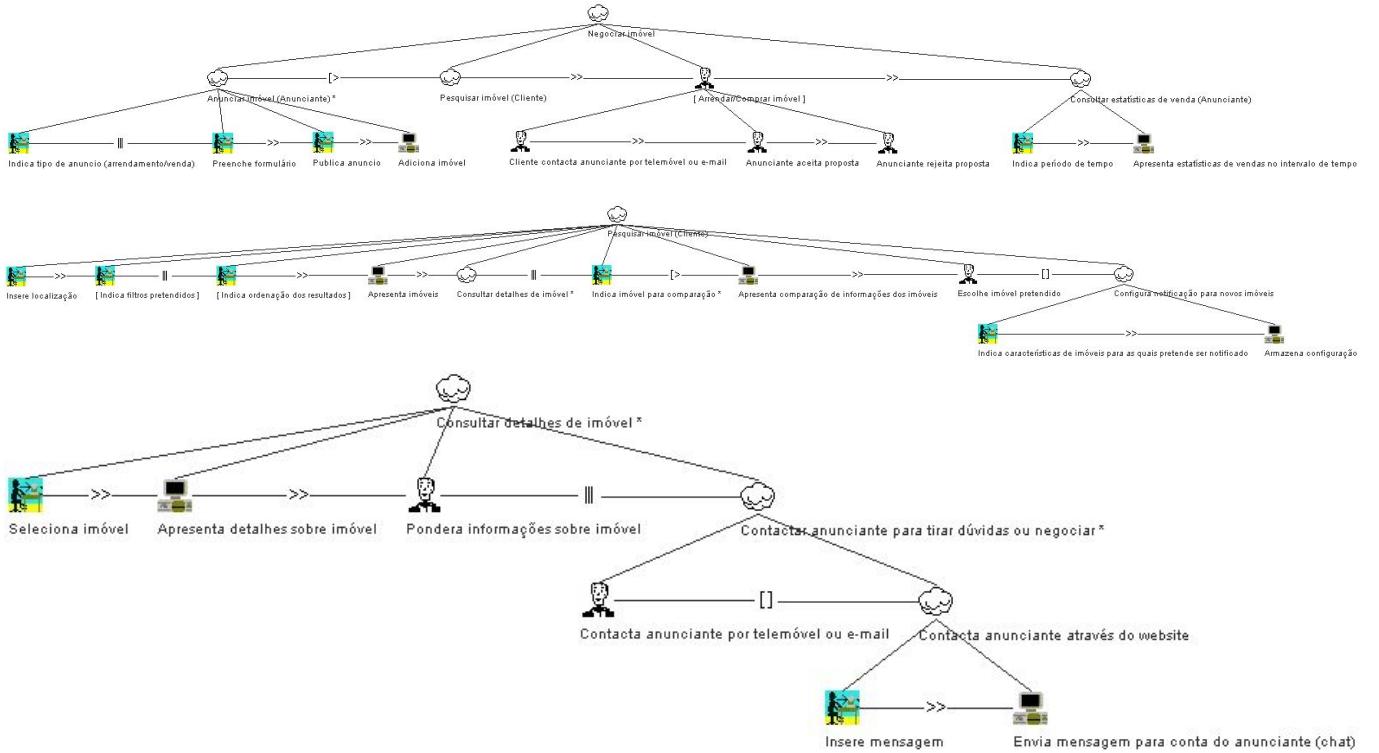


Figura 3: Modelo de tarefas para negociar um imóvel, com o *website* Home4All.

Destacam-se como principais melhorias neste modelo de tarefas geral, em relação ao modelo referente à utilização de outros *websites*, o facto da comparação entre imóveis ser efetuada pelo sistema, facilitando o trabalho do utilizador. Para além disso, acrescenta-se a possibilidade de consulta de estatísticas de vendas, que pode ser útil para agentes imobiliários, por exemplo.

Salienta-se ainda que se espera implementar filtros bastante mais específicos em relação a outros *websites*, o que permitirá tornar mais rápida e direta a pesquisa do cliente.

As principais tarefas desta aplicação são as de inserção de um novo imóvel, pesquisa de imóvel e comparação entre imóveis. Assim, apresenta-se os modelos dessas tarefas nas figuras 4, 5 e 6.

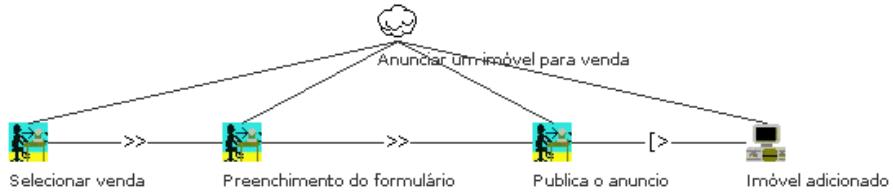


Figura 4: Modelo da tarefa “Anunciar imóvel para venda”

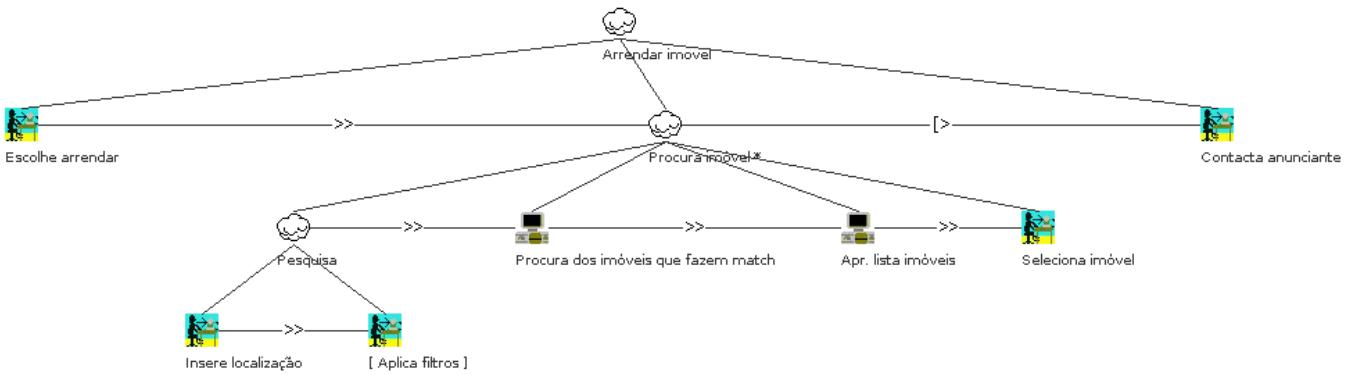


Figura 5: Modelo da tarefa “Arrendar imóvel”

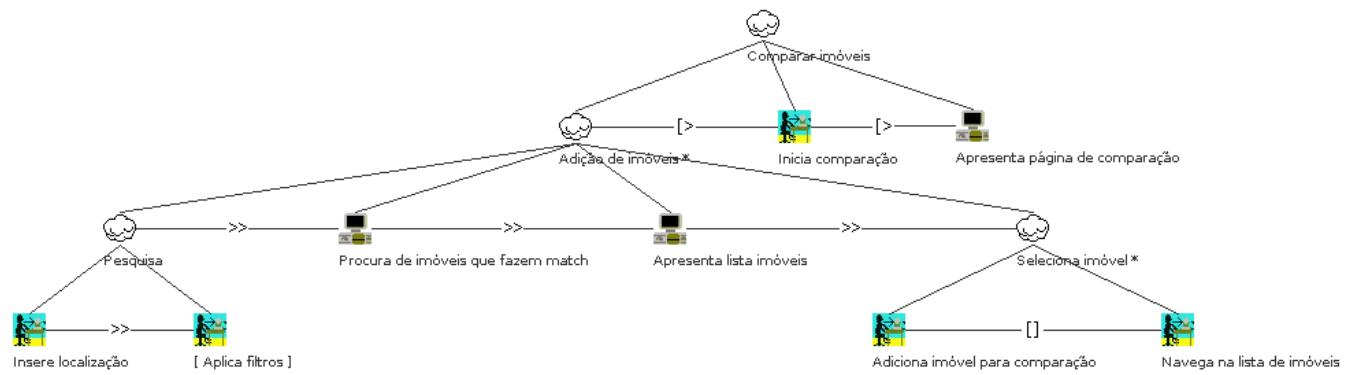


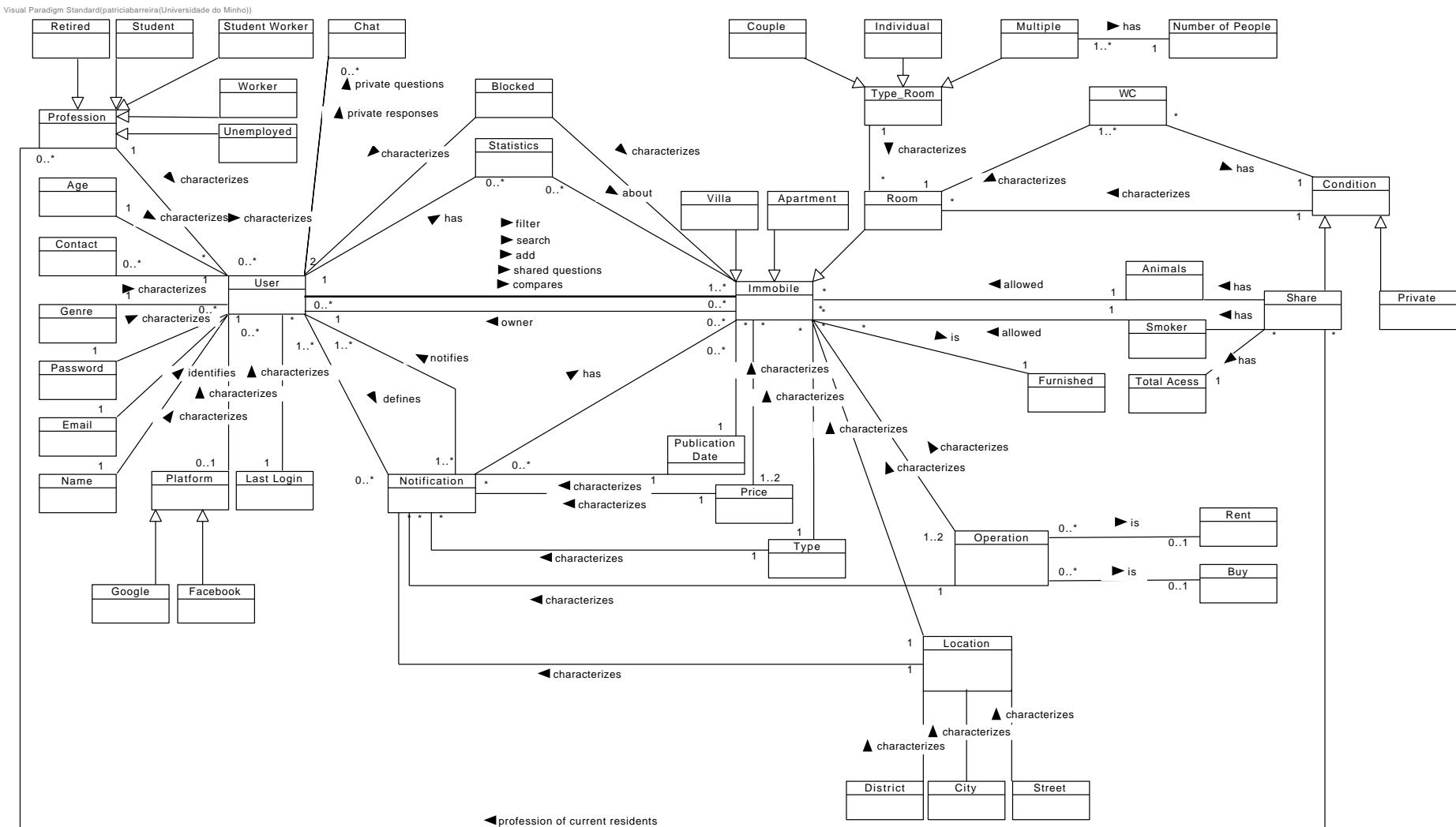
Figura 6: Modelo da tarefa “Comparar imóveis”

4 Domínio do Problema

Uma primeira fase de desenvolvimento passa pela modelação do domínio do problema, a modo de gradualmente ser modelado o diagrama de classes. Assim sendo, para modelar o domínio, é necessário ter em conta:

- As várias entidades do problema;
- Os relacionamento entre as várias entidades, isto é, especificar os termos que envolvem a relação e a cardinalidade associada;

As *constraints*, *stakeholders* e objetos a considerar no sistema foram sendo registados ao longo da contextualização e da análise requisitos conforme descrito na secção 3. A própria esquematização do modelo permite raciocinar sobre a natureza do sistema, e permite isolar num único esquema várias regras de negócio a considerar. Como tal, durante a modelação do domínio surge um produto que permite, por um lado, que a equipa consiga interiorizar especificamente o tipo de sistema a desenvolver e, por outro lado, existe uma maquete que facilita a validação e desenvolvimento de diagramas que se seguem.



5 Concepção (Prototipagem)

5.1 Mockups

Para desenvolver os *mockups* da interface da aplicação, optou-se por recorrer a protótipos de baixa fidelidade, mas realistas. Desta forma consegue-se representar antecipadamente o *design* que se pretende que a interface tenha, bem como capturar interações básicas do utilizador com o sistema, com uma ferramenta rápida e fácil de utilizar, sem custos e que permite um maior controlo sobre o nível de abstração.

Assim, torna-se possível obter uma avaliação prévia do *design* que se irá implementar, bem como efetuar testes preliminares de usabilidade de forma suficientemente realista.

A ferramenta escolhida para a prototipagem foi a *Marvel*¹, que permite desenhar a interface com recurso à funcionalidade de *drag-and-drop* e indicar o resultado de determinadas ações por parte do utilizador.

Ao longo de toda esta fase de prototipagem, levou-se em consideração os princípios de usabilidade:

- **Learnability** – facilidade com que novos utilizadores podem começar uma interação efetiva com o sistema e alcançar a performance máxima. Inclui os seguintes princípios:
 - **Predictability** – capacidade de prever o efeito de ações sobre o sistema;
 - **Synthesizability** – capacidade de avaliar o efeito de ações passadas no estado atual, facilitada quando a *user interface* fornece informações sobre mudanças de estado;
 - **Familiarity** – como conhecimento anterior, sobre o mundo ou outros sistemas, se aplica ao novo sistema;
 - **Generalizability** – extensão de conhecimento de interação específica a novas situações (similares);
 - **Consistency** – semelhança no comportamento do sistema em situações ou implementação de objetivos parecidos.
- **Flexibility** – a variedade de formas utilizadas para trocar informação entre o utilizador e o sistema. Inclui os seguintes princípios:
 - **Dialogue initiative** – pode-se permitir um controlo superior por parte do utilizador ou do sistema;
 - **Multithreading** – capacidade do sistema suportar a interação do utilizador para mais do que uma tarefa de cada vez;
 - **Task migrability** – transferir a responsabilidade de controlos de tarefas entre utilizador e sistema;
 - **Substitutivity** – permitir que valores equivalentes possam ser substituídos um pelo outro, que o mesmo *output* possa ser apresentado de várias formas e que o *input* e o *output* sejam apresentados da mesma forma;
 - **Customizability** – grau de modificação da interface de utilizador: adaptabilidade, quando a modificação é efetuada pelo utilizador, e adaptação, quando esta é efetuada pelo sistema.
- **Robustness** – o nível de apoio fornecido ao utilizador na determinação de se a ação foi bem sucedida e avaliação do comportamento direcionado por objetivos. Inclui os seguintes princípios:
 - **Observability** – capacidade do utilizador avaliar o estado interno do sistema através das suas representações percepíveis; envolve aspectos de naveabilidade (para explorar o estado atual), padrões, acessibilidade (para navegar entre estados observáveis), persistência e visibilidade das operações possíveis;

¹<https://marvelapp.com/>

- **Recoverability** – capacidade dos utilizadores realizarem ações corretivas;
- **Responsiveness** – como os utilizadores percebem a taxa de comunicação com o sistema (tempos de resposta reduzidos e estáveis);
- **Task conformance** – grau em que os serviços do sistema suportam as tarefas dos utilizadores.

Alguns destes princípios podem ser abordados em relação a todo o *website*, pelo que se considera útil a referência dos mesmos antes da apresentação dos *mockups*.

Relativamente aos princípios de flexibilidade, o sistema terá, por exemplo, uma iniciativa de diálogo (*Dialogue initiative*) superior, uma vez que se pretende guiar os utilizadores na realização das suas tarefas. Assim, os utilizadores podem selecionar arbitrariamente a ação que querem executar entre as disponibilizadas (pesquisar, comparar, comentar ou anunciar imóvel, *e.g.*) e o sistema indicar-lhes-á os dados que têm de introduzir para completar a tarefa, apresentando de seguida os *outputs* necessários. Para facilitar a utilização do *website* por parte de utilizadores experientes, pretende-se implementar o acesso às várias funcionalidades através de atalhos específicos, bem como outros atalhos comuns (TAB para passar para o próximo campo num formulário, utilização das setas para passar de uma configuração/favorito para o seguinte ou anterior, ENTER para submeter um formulário em alternativa a clicar num botão, entre outros).

Para além disso, o sistema suportará *Multithreading*, na medida em que permitirá ao utilizador autenticado efetuar tarefas distintas ou iguais em diferentes separadores ou janelas do *browser*. Assim, será-lhe-á possível, por exemplo, adicionar dois imóveis ao mesmo tempo (*interleaved multithreading*), em separadores distintos, bem como pesquisar simultaneamente, noutras separadores, imóveis com determinadas características (*concurrent multithreading*).

Ainda em relação à flexibilidade, será considerado o princípio de costumização (*Customizability*), na vertente de adaptação, isto é, implementar-se-á a interface de forma a que se ajuste automaticamente a diferentes tamanhos de ecrã (*Responsive Web Design*).

Em termos de robustez, procurar-se-á implementar um *website* que, apesar de não permitir ao utilizador retroceder diretamente uma ação realizada que altere o estado do sistema, lhe permita corrigir possíveis erros realizados na maioria dos casos (*Recoverability - forward recovery*). Assim, se um imóvel for criado incorretamente ou adicionado aos favoritos ou para comparação sem essa intenção, poderá sempre ser removido do sistema ou da lista associada, respetivamente. Da mesma forma, os dados de um imóvel poderão ser editados pelo respetivo anunciante. Contudo, caso o utilizador envie uma mensagem privada ou pública (comentário) e a pretenda remover ou alterar, não terá essa possibilidade.

Também se terá em atenção o princípio de **Responsiveness**, pretendendo-se implementar a aplicação de forma a que os tempos de resposta sejam reduzidos e relativamente constantes. Estes tempos apenas poderão ser mais elevados quando se pesquisam imóveis. Nessa situação, poderá optar-se por obter apenas os imóveis da primeira página, sendo que os das seguintes serão carregados conforme o utilizador os solicite.

Por fim, todas as páginas para o *website* foram desenhadas tendo em conta as tarefas do utilizador da aplicação, pelo que se considera que estas cumprem todos os requisitos propostos (**Task conformance**).

Também para maximizar o grau de usabilidade do *website* desenvolvido, procurou-se recorrer a vários HCI (*Human-Computer Interaction*) *Design Patterns*, de forma a reutilizar conhecimento sobre soluções de *design* bem sucedidas. Para isso, recorreram-se aos *patterns* apresentados nos *websites* *Designing Interfaces*² e *Welie*³.

Entre os vários *design patterns* analisados, salientam-se os seguintes, que serão utilizados nas várias páginas do *website*:

- **Visual Framework** – cada página apresentará o mesmo *layout* básico, as mesmas cores e elementos de estilo iguais, apesar da grande variedade de conteúdo.

²<http://designinginterfaces.com/firstedition/>

³<http://www.welie.com/patterns/index.php>

- **Liquid Layout** – conforme o utilizador redimensiona a janela, os conteúdos da página são redimensionados em conjunto, de forma à página estar constantemente preenchida.

Estes *patterns*, bem como outros mais específicos de cada página, podem ser observados nos *mockups* apresentados de seguida. Salienta-se que apenas se apresentam os *mockups* principais, bem como algumas considerações sobre os mesmos.

5.1.1 Página Principal (sem login)

Na página principal, optou-se por permitir iniciar a única tarefa fundamental do *website* que não exige autenticação, a pesquisa de imóveis, bem como por apresentar algumas informações básicas sobre as funcionalidades disponibilizadas. O respetivo *mockup* é apresentado na figura 8.

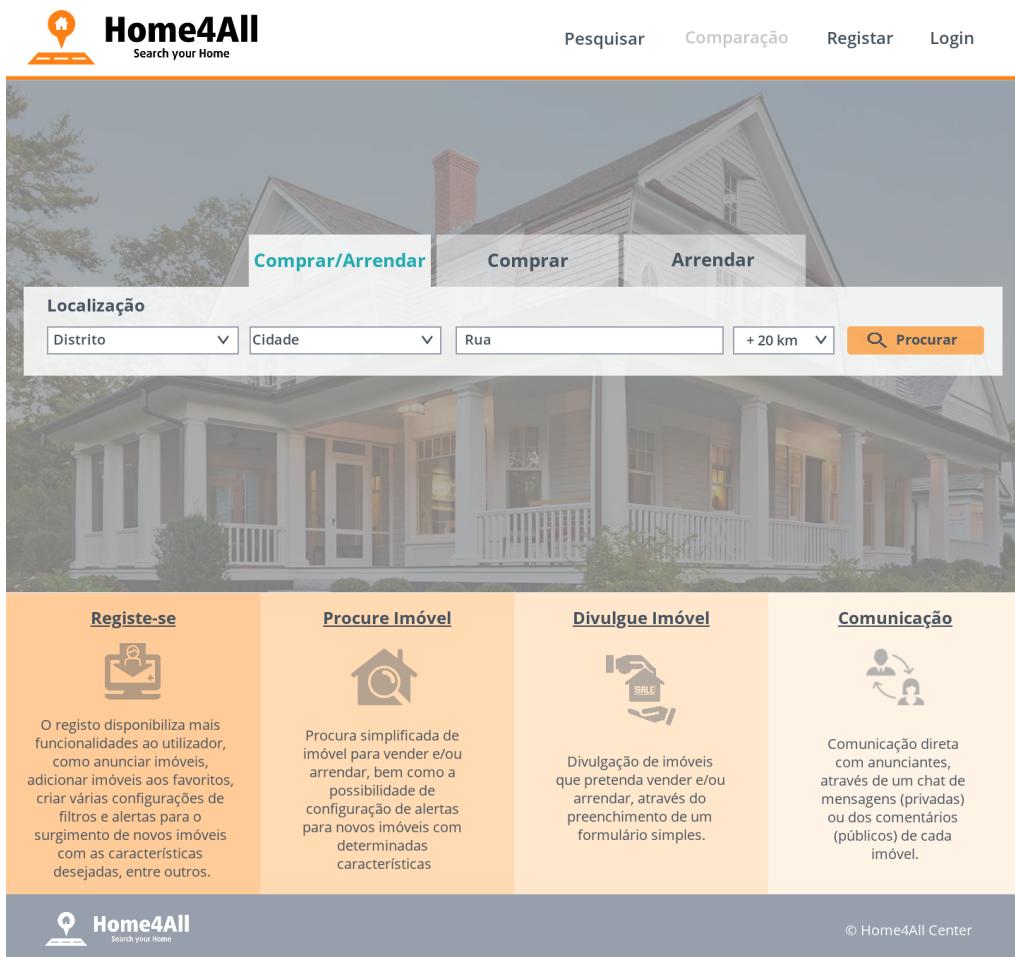


Figura 8: *Mockup* da página principal, sem *login* efetuado.

Nesta página, salientam-se como princípios de usabilidade respeitados os que se apresentam de seguida:

- **Learnability: Predictability** – Inatividade do menu de “Comparação” permite ao utilizador prever que o sistema de comparação de imóveis não está disponível. Passando o rato sobre o menu surgirá a identificação do motivo (necessitam de ser adicionados pelo menos dois imóveis ao sistema de comparação).
- **Learnability: Familiarity** – A cor do menu inativo (cinzento) é comumente utilizada para indicar que determinado botão ou elemento não está em funcionamento. Para além

disso, o formato do menu principal e o sistema de pesquisa é semelhante aos de muitos outros *websites*.

- **Learnability: Generalizability/Consistency** – O menu principal é igual em todas as páginas, alterando apenas a formatação da funcionalidade selecionada e quais as funcionalidades disponíveis de acordo com o estado de autenticação (utilizador autenticado ou não autenticado).
- **Flexibility: Substitutivity** – A distância pode ser escrita, como um inteiro (que é interpretado como estando em quilómetros), ou pode ser selecionada de entre as disponíveis.
- **Robustness: Observability** – O utilizador pode perceber que não está autenticado pela presença das funcionalidades “Registar” e “Login” no menu principal. Para se autenticar, basta utilizar essas funcionalidades, que se encontram posicionadas de forma bem visível em qualquer página.

Relativamente aos padrões de *design* utilizados, sobressaem-se os seguintes do *website Designing Interfaces*:

- **Clear Entry Points** – A página principal do *website* foca a atenção dos utilizadores nas caixas de texto onde se insere a localização para efetuar a pesquisa de imóveis. O menu principal e restante texto descritivo são vistos como elementos secundários.
- **Global Navigation** – O menu principal aparece em todas as páginas do *website* e apresenta um conjunto de ligações para as páginas mais importantes do mesmo.
- **Dropdown Chooser** – O distrito, a cidade e a distância podem ser selecionados de um *dropdown*, que contém uma coleção variada de valores que podem ser escolhidos.
- **Deep Background** – A imagem de fundo, relacionada com o tema do *website* (imóveis), foi introduzida para tornar a página principal mais atrativa e anunciar de forma evidente o tipo de produtos considerados no *website*.

Por fim, dos *design patterns* abordados no *website Welie*, foram utilizados os seguintes:

- **Home Link** – O logótipo apresentado no canto superior esquerdo de todas as páginas do *website* constitui uma ligação para a página principal.
- **Main Navigation** – O menu principal é incluído em todas as páginas e está sempre visível, numa posição fixa.
- **Action Button** – O botão “Procurar” permite ao utilizador realizar uma ação relevante no contexto da página atual, pelo que é realçado com recurso a um ícone, bem como a cor e formato distintos (retângulo laranja com cantos arredondados). Para além disso, o *label* do botão inclui o verbo da ação.
- **Autocomplete** – O campo da rua apresenta sugestões para completar o que está escrito, conforme o utilizador vai escrevendo.

5.1.2 Pesquisa de Imóvel

Relativamente à página de pesquisa de imóveis, optou-se por, para a localização, recorrer ao mesmo formato de pesquisa utilizado na página principal. Relativamente aos filtros e à ordenação dos resultados, apresenta-se os mesmos divididos por nome ou categoria, conforme se pode verificar no *mockup* apresentado na figura 9.

Para se adicionar um filtro ou alterar a ordenação dos resultados, tem de se clicar sobre a respetiva categoria e selecionar o filtro ou ordenação pretendido. O resultado de clicar sobre estas categorias é explicitado nos *mockups* das figuras 10 a 15.

A seleção de filtros ou alteração da ordenação implica a atualização automática dos resultados, caso a pesquisa por localização já tenha sido efetuada. Quando esta pesquisa é realizada, os resultados são apresentados conforme ilustrado no *mockup* da figura 16.

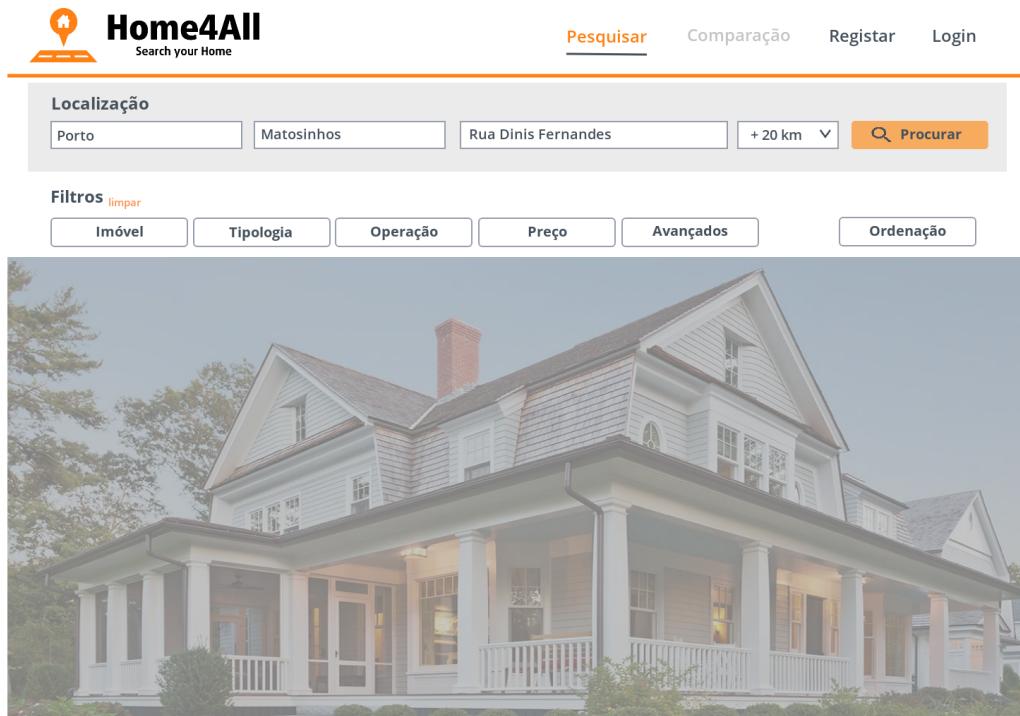


Figura 9: *Mockup* da página de pesquisa, sem resultados nem *login* efetuado.

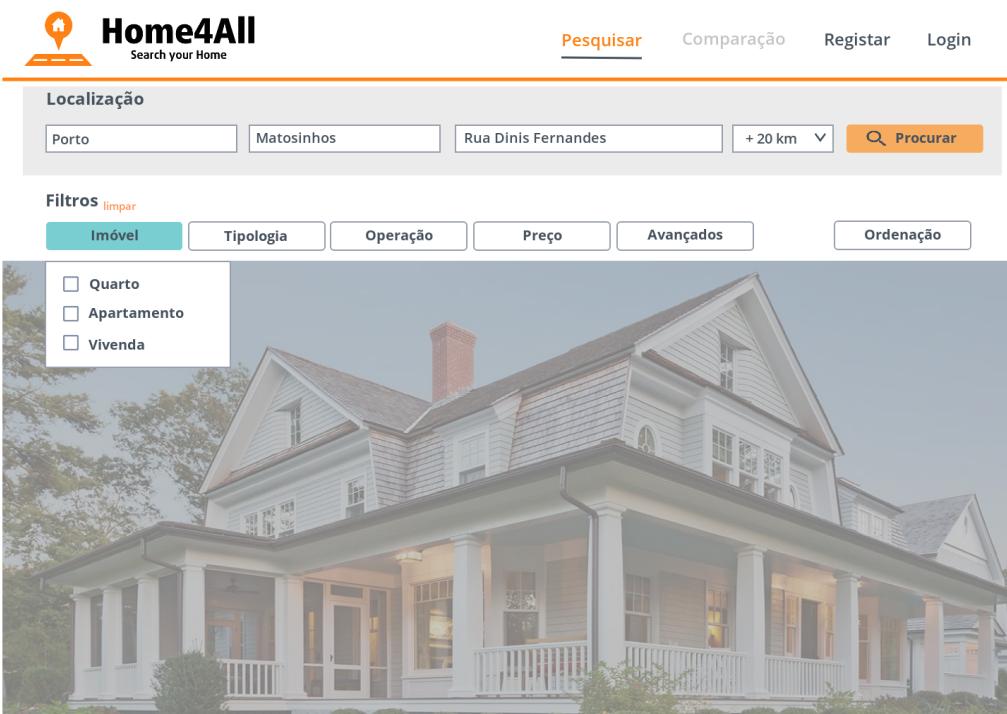


Figura 10: *Mockup* da página de pesquisa, com menu do filtro **Imóvel** aberto.

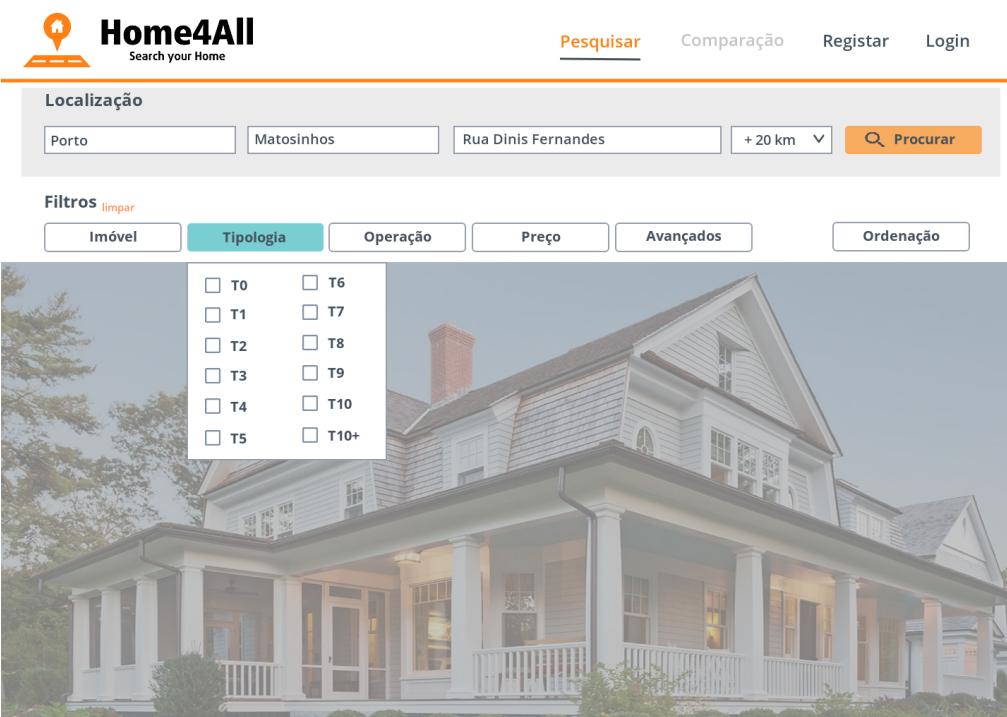


Figura 11: *Mockup* da página de pesquisa, com menu do filtro **Tipologia** aberto.

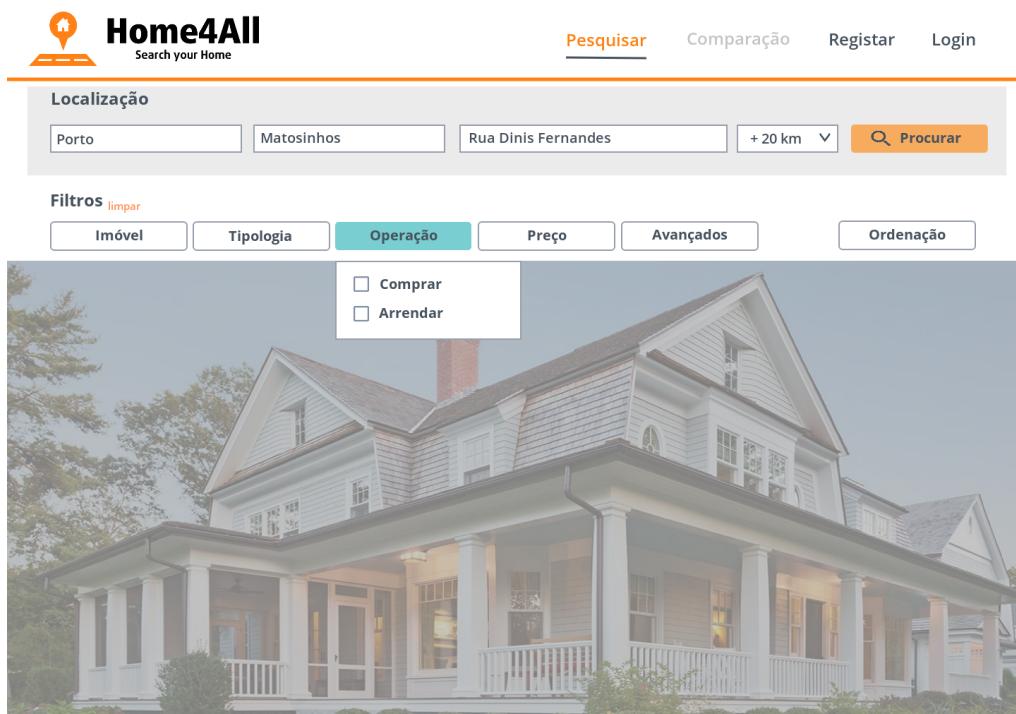


Figura 12: Mockup da página de pesquisa, com menu do filtro Operação aberto.

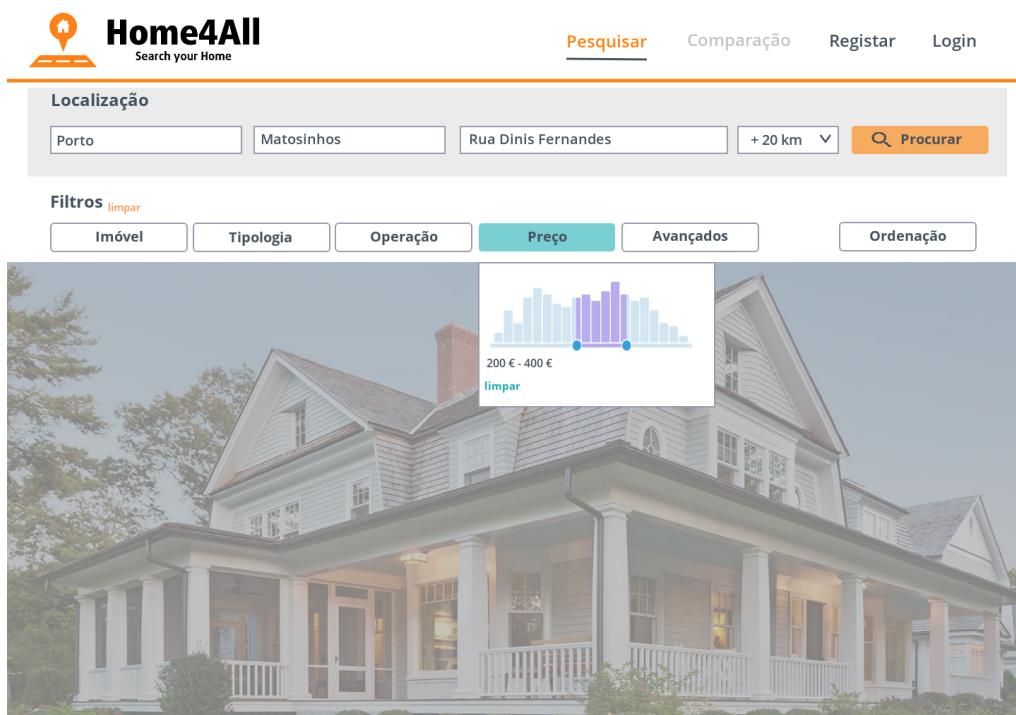


Figura 13: Mockup da página de pesquisa, com menu do filtro Preço aberto.

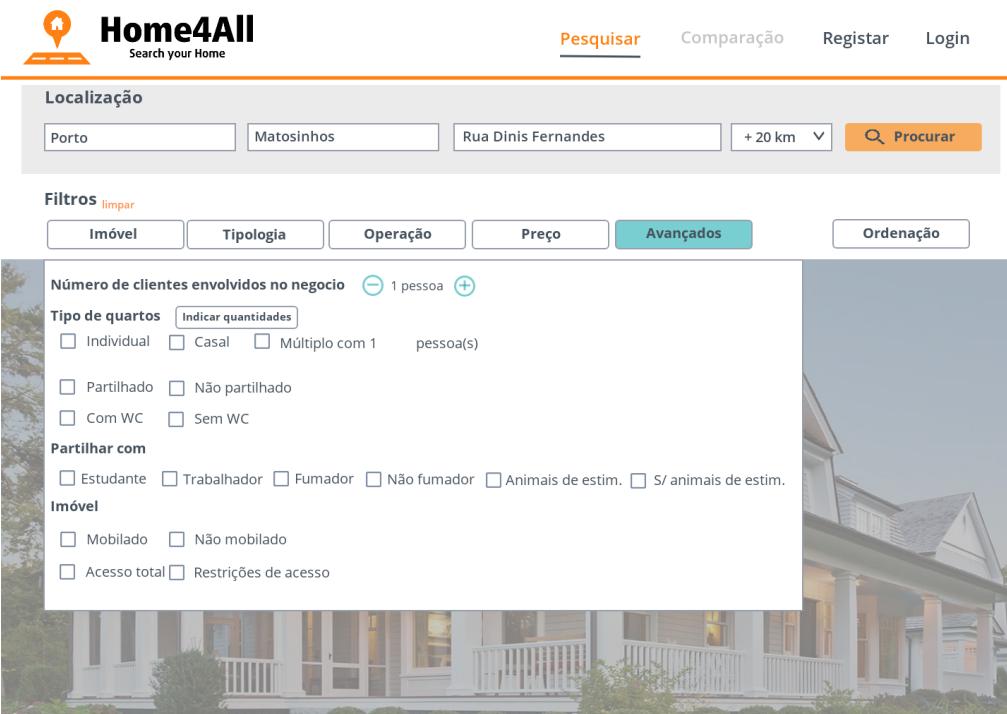


Figura 14: Mockup da página de pesquisa, com menu do filtro Avançados aberto.

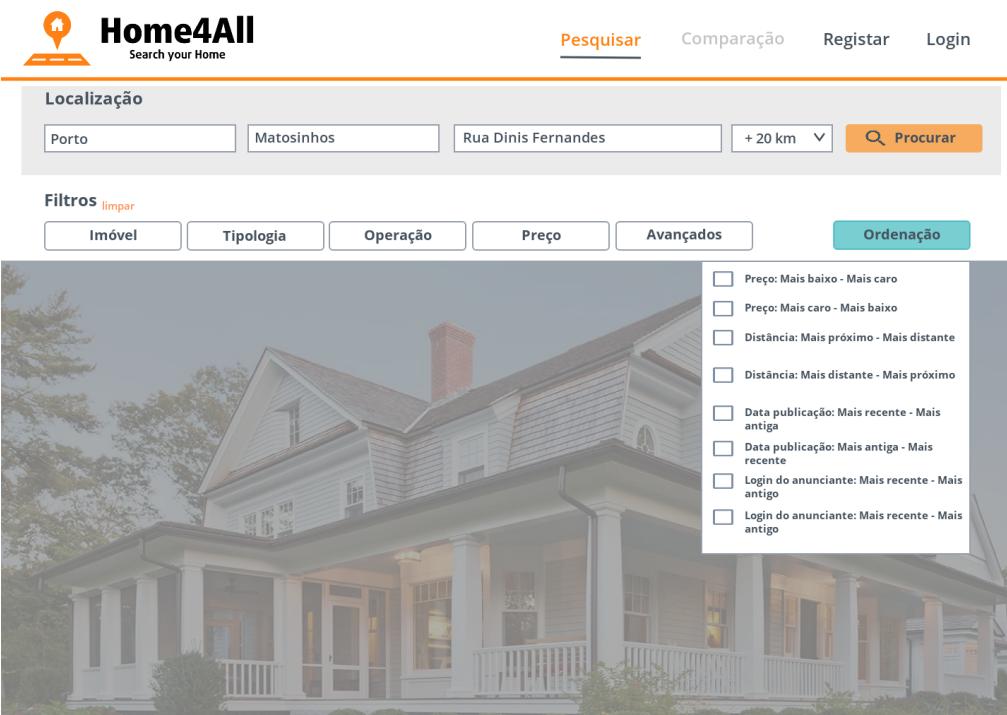


Figura 15: Mockup da página de pesquisa, com menu de Ordenação aberto.

A interface de usuário da Home4All é apresentada em uma estrutura de formulário. No topo, há uma barra com o logo "Home4All" e o lema "Search your Home". À direita da barra estão os links "Pesquisar", "Comparação", "Registar" e "Login". O formulário principal contém campos para "Localização" (Porto, Matosinhos, Rua Dinis Fernandes) e uma opção "+ 20 km". Um botão "Procurar" com ícone de lupa encerra o formulário.

Abaixo do formulário, uma seção intitulada "Filtros" com uma opção "limpar" mostra filtros ativos: "Imóvel", "Tipologia", "Operação", "Preço" e "Avançados".

O resultado da pesquisa é dividido em duas entradas:

- Paulo Fonseca**: Imagem de um quarto com cama e armário. Detalhes: Online, Arrendo 1 Quarto(s), Quarto em apartamento T2 situado no centro de Mat..., Rua Alfredo Cunha - Matosinhos, €350/Mês (Gastos Incluídos). Código: 270886850. Um link "+inf" e um mapa de Matosinhos com o local marcado.
- Maria Azevedo**: Imagem de um quarto com cama e armário. Detalhes: Online, Arrendo 1 Quarto(s), Big room near metrostation Marques with double bed, wardrobe and chest of drawers as desk as well. C..., Rua da Constituição - Porto(centro), €330/Mês. Código: 938295481. Um link "+inf" e um mapa do centro do Porto com o local marcado.

Figura 16: Mockup da página de pesquisa, com resultados.

Na pesquisa de imóveis, tiveram-se em principal consideração os seguintes princípios de usabilidade:

- **Learnability: Predictability** – Quando a seleção de determinado filtro tem como consequência o retorno de nenhum resultado, esse filtro aparece como desativado (cinzento).
- **Learnability: Familiarity** – O sistema de pesquisa e de filtros é semelhante ao de outros websites de procura de qualquer tipo de produto, incluindo imóveis.
- **Learnability: Generalizability/Consistency** – Os filtros são selecionados sempre da mesma maneira e a área de pesquisa (localização) que aparece no topo da página é a mesma que a utilizada na página inicial (permitindo supor que o efeito no sistema será o mesmo).
- **Robustness: Observability** – Os elementos necessários para efetuar uma (nova) pesquisa encontram-se bem visíveis para o utilizador. Para além disso, quando forem devolvidos muitos resultados, os imóveis serão apresentados de acordo com um sistema de paginação (*browsability*).

Analizando os *design patterns* disponibilizados pelo website **Designing Interfaces**, conseguiu-se aplicar nesta página os apresentados de seguida:

- **Dropdown Chooser** – O distrito, a cidade e a distância podem ser selecionados de um *dropdown*, que contém uma coleção variada de valores que podem ser escolhidos.
- **Deep Background** – A imagem de fundo foi introduzida para preencher o espaço vazio aquando da inexistência de resultados e incentivar o utilizador a efetuar uma nova pesquisa para encontrar o seu imóvel ideal.

No website **Welie** foi possível aplicar uma quantidade de padrões de *design* superior, sendo estes os seguintes:

- **Action Button** – O botão “Procurar” permite ao utilizador realizar uma ação relevante no contexto da página atual, pelo que é realçado com recurso a um ícone, bem como a cor e formato distintos (retângulo laranja com cantos arredondados). Para além disso, o *label* do botão inclui o verbo da ação.
- **Autocomplete** – O campo da rua apresenta sugestões para completar o que está escrito, conforme o utilizador vai escrevendo.
- **Product Advisor** – Os filtros que o utilizador pode selecionar permitem ao sistema aconselhá-lo sobre que imóveis deve consultar, tendo em conta restrições, preferências e necessidades do utilizador.
- **Map Navigator** – A página de pesquisa com resultados utiliza este padrão para cada imóvel apresentado, de forma a facilitar a percepção do utilizador em relação à área onde o imóvel se localiza.
- **Paging** – Conforme referido anteriormente, quando a quantidade de resultados apresentados for muito elevada, utilizar-se-á um sistema de *paging*.
- **View** – A página da pesquisa com resultados consiste numa vista geral sobre os vários imóveis disponíveis, tendo em conta as restrições indicadas. O utilizador pode consultar um imóvel clicando sobre o mesmo e depois retornar à lista de resultados, retrocedendo para a página anterior, selecionando a funcionalidade “Pesquisa” do menu principal ou clicando no botão “Procurar” (neste caso também pode alterar campos relativos à localização, podendo obter como resultado uma lista diferente).

5.1.3 Consulta de imóvel

Quanto à página de consulta de um imóvel, optou-se por inserir nesta toda a informação relativa ao imóvel, bem como a possibilidade de consultar a sua localização num mapa, enviar mensagem privada ao anunciante (se autenticado), adicionar o imóvel aos favoritos ou para comparação e consultar ou adicionar (se autenticado) comentários.

Procurou-se dispor a informação de forma simples e apelativa, para captar a atenção do cliente. Contudo, devido à elevada quantidade de informação, houve também uma grande preocupação em garantir uma disposição direta e facilmente perceptível por parte do utilizador.

Ponderou-se ainda se a informação deveria ser dividida, por não caber toda no ecrã. Contudo, considerou-se que a apresentação da informação seguida é mais vantajosa, utilizando-se o *scroll* para aceder aos dados não visíveis de forma imediata. Desta forma, o acesso à informação é mais rápido, não requerendo cliques de rato extra.

Assim, o *mockup* obtido para a página de consulta de um imóvel é apresentado na figura 17.

Home4All
Search your Home

Pesquisar Comparação Registar Login

Localização

Porto Matosinhos Rua Dinis Fernandes + 20 km Procurar

Paulo Fonseca



Descrição:
Quarto em apartamento T2 situado no centro de Matosinhos. Bom suporte de lojas comerciais (supermercados e outro tipo de comércio local). Boa rede de transportes (autocarro e metro) com fácil acesso ESAD, Universidade do Porto e centro do Porto. Próximo do climar, Parque da cidade e da praia (10 minutos de percurso a pé).

Tipologia: T2 **Área:** 400 m²

Imóvel: Quarto

Localização: Av./Rua: Rua Alfredo Cunha Matosinhos
Porto, Matosinhos
4454-508

Contactos: Mensagem Privada 226 453 643 example1@email.com

Comparar

Quartos disponíveis:

Fotografias	Tipo	Área (m ²)	Mobilado	Q. Banho	Disponibilidade	Preço
	Individual	20	Sim	Partilhado	1/04/2019	350 €/mês

Renda inclui: Água Eletricidade TV Cabo Internet Condomínio
 Telefone Limpeza

Tipo de inclinos procurados:

Género: ou Idade: 18 - 30 anos Ocupação: Estudante

Fumador: Não Com Animais: Indiferente

Divisões ou equipamentos que podem ser utilizados:

Elevador Cozinha Microondas Fogão Frigorífico Máq. Lavar Roupa
 Televisão TV cabo Wi-Fi Garagem

Arrendatários atuais:

2 pessoas: 2 0 Ocupações: Estudante Trabalhador

0 Fumadores

1 Animal de Estimação: cão

Comentários  

© Home4All Center

Figura 17: Mockup da página com informações sobre um imóvel.

Nesta página consegue-se verificar os seguintes princípios de usabilidade:

- **Learnability: Predictability** – Inatividade dos botões de enviar mensagem privada

ao anunciante e de adicionar imóvel aos favoritos permite ao utilizador prever que estas funcionalidades não estão disponíveis. Passando o rato sobre os botões surgirá a identificação do motivo (utilizador necessita de se autenticar primeiro).

- **Learnability:** *Synthesizability* – Após adicionar ou remover o imóvel dos favoritos ou do sistema de comparação, surgirá na página uma mensagem efémera de sucesso ou insucesso. Para além disso, quando um imóvel já se encontra nos favoritos, o símbolo apresentado é de uma estrela completamente preenchida. Quando o imóvel não se encontra nos favoritos, apenas aparece o contorno da estrela. Para a comparação, se o imóvel ainda não está incluído, então o botão associado tem o *label* “Comparar”. Caso contrário, o *label* é “Não comparar”.
- **Learnability:** *Familiarity* – A página de consulta do imóvel é semelhante a páginas de outros websites que descrevem qualquer tipo de produto, incluindo imóveis.
- **Learnability:** *Generalizability/Consistency* – A existência do mesmo menu de pesquisa no topo da página permite supor que o efeito no sistema será o mesmo que quando aparecia na página de pesquisa e na página principal.
- **Robustness:** *Observability* – Apesar das operações de consultar ou adicionar comentário não estarem imediatamente visíveis (ser necessário efetuar *scroll*), considera-se que este fator não representará um impedimento para os clientes. De facto, muitos *websites* também utilizam sistemas deste género, pelo que o utilizador já deverá estar acostumado.

Relativamente aos *design patterns* do *website Designing Interfaces*, apenas foi utilizado o *Extras On Demand*. Recorreu-se a este padrão para desenhar o surgimento do mapa com a localização do imóvel e a visualização dos comentários do imóvel. Como são informações adicionais sobre o imóvel (localização gráfica e questões públicas colocadas por outros clientes), considerou-se que estes dados apenas deveriam ser apresentados mediante solicitação por parte do utilizador.

Já do *website Welie*, conseguiram-se utilizar mais padrões de *design*:

- **Map Navigator** – Clicar no botão ao lado da “Localização” resulta no aparecimento de um mapa (no qual se pode navegar para outras localizações), com indicação da localização do respetivo imóvel.
- **Action Button** – O botão “Procurar” permite ao utilizador realizar uma ação relevante no contexto da página atual, pelo que é realçado com recurso a um ícone, cor e formato distintos (retângulo laranja com cantos arredondados). Para além disso, o *label* do botão inclui o verbo da ação.
- **Slideshow** – As fotografias do imóvel são apresentadas cada uma por alguns segundos. Para além disso, é possível passar para a imagem seguinte/anterior, bem como selecionar uma imagem específica.
- **Collector** – O imóvel pode ser adicionado a uma lista nos favoritos ou à lista dos imóveis no sistema de comparação, enquanto o utilizador o visualiza. A lista de imóveis para comparação está sempre acessível (se a lista tiver dois ou mais imóveis), enquanto que a dos favoritos necessita de autenticação, passando então a estar acessível a partir de qualquer página.
- **Thumbnail** – Imagens dos imóveis mais pequenas são apresentadas para permitir ao utilizador consultar as fotografias que mais lhe interessar, de forma mais direta.
- **Comment Box** – Ao clicar no botão de adicionar comentário, surge um pequeno formulário (com apenas um campo) para o utilizador autenticado inserir uma questão pública, a ser respondida pelo anunciante do imóvel.

5.1.4 Comparação entre imóveis

Para permitir uma comparação imediata entre as várias características dos imóveis selecionados, utilizou-se uma representação tabular da informação de cada imóvel, conforme solicitado nos requisitos. Essa representação pode ser consultada no *mockup* apresentado na figura 18.

	Opção 1	Opção 2
Fotografias		
Tipo	Quarto Individual	Apartamento completo
Tipologia	T3	T1
Área (m ²)	Quarto: 50 m ² Apartamento: 300 m ²	400 m ²
Mobilado	Sim	Sim
Disponibilidade	1/04/2019	7/04/2019
Preço	350 €/mês	500 €/mês
Distância à localização preferencial	+ 5 Km	+ 0 Km
Renda inclui outros gastos	Não	Não
Contacto	✉ Mensagem Privada 226 453 643 example1@email.com	✉ Mensagem Privada 226 543 123 example2@email.com
	Ver detalhes	Ver detalhes

Figura 18: *Mockup* da página de comparação entre imóveis.

Em termos de princípios de usabilidade, salientam-se os seguintes:

- **Learnability: Familiarity** – Sistemas de comparação baseados em tabela são bem conhecidos e percepíveis por parte de qualquer tipo de utilizador.
- **Learnability: Generalizability/Consistency** – As funcionalidades de visualização do imóvel no mapa, envio de mensagem privada ao anunciantre e adição do imóvel aos favoritos são executadas da mesma forma (com o mesmo tipo de elementos) que na página de consulta do imóvel.
- **Robustness: Observability** – As operações de adicionar aos favoritos e consultar imóvel não estão imediatamente visíveis, isto é, é necessário fazer *scroll* para as visualizar. Con-

tudo, estas não são as principais tarefas a executar nesta página, ao contrário da análise de comparação. Após essa análise é que faz sentido considerar realizar uma das outras ações.

Mais uma vez, dos *design patterns* do website **Designing Interfaces** apenas se utilizou o **Extras On Demand**. Recorreu-se a este padrão para desenhar o surgimento do mapa com a localização do imóvel e a visualização dos detalhes de um dos imóveis.

Relativamente ao website **Welie**, conseguiram-se utilizar os seguintes padrões de *design*:

- **Map Navigator** – Clicar no botão incluído na área da localização resulta no aparecimento de um mapa (no qual se pode navegar para outras localizações), com indicação da localização do respetivo imóvel.
- **Thumbnail** – Imagens dos imóveis mais pequenas são apresentadas para permitir ao utilizador ter uma visão geral das várias fotografias e consultar as que mais lhe interessar, de forma direta.
- **Product Comparison** – É apresentada uma matriz de imóveis e características dos mesmos, para facilitar ao utilizador a comparação entre eles.

5.1.5 Registo

A página de registo foi desenhada com um formato *standard*, permitindo o recurso a contas do Facebook ou da Google. Caso seja utilizada uma conta externa no registo, serão posteriormente solicitados ao utilizador dados de contacto, idade, género e profissão.

O *mockup* desta página é apresentado na figura 19.

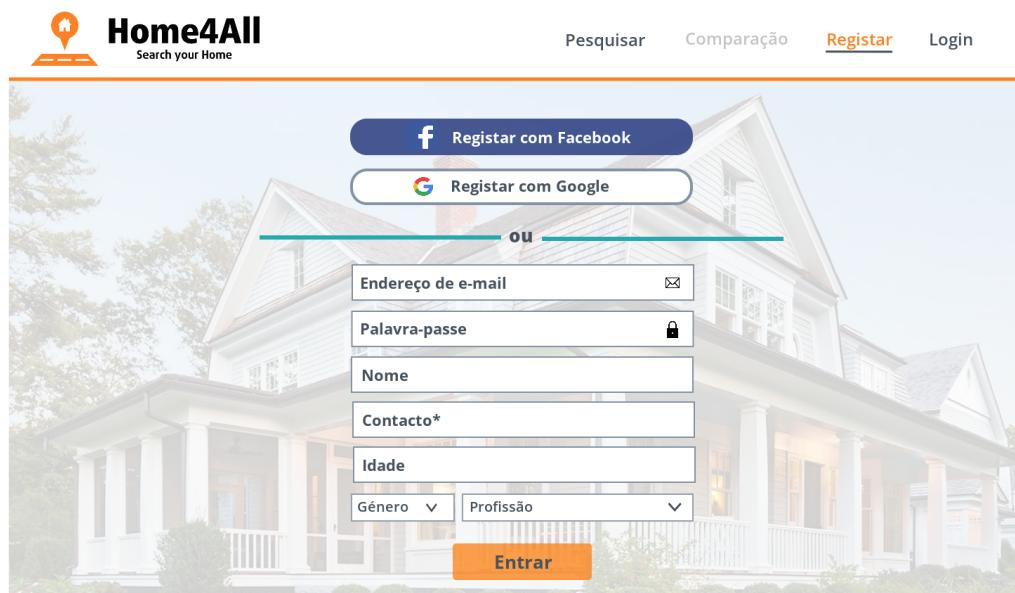


Figura 19: *Mockup* da página de registo.

Relativamente aos princípios de usabilidade aplicáveis a esta página, destacam-se os seguintes:

- **Learnability: Synthesizability** – Após clicar no botão “Entrar” (registo), será indicada uma mensagem de sucesso ou insucesso. No primeiro caso, será ainda efetuada automaticamente a autenticação do utilizador. No segundo caso, será indicado o motivo do insucesso.
- **Learnability: Familiarity** – O sistema de registo é semelhante ao de outros *websites*.

- **Flexibility:** *Substitutivity* – O contacto pode ser escrito com ou sem o respetivo indicativo de país.
- **Robustness:** *Observability* – O botão para registo está visível em ecrãs médios e grandes. No caso de ecrãs pequenos, o botão aparece imediatamente a seguir ao respetivo formulário, de preenchimento obrigatório.

Entre os padrões de *design* do website **Designing Interfaces**, utilizou-se o **Dropdown Chooser** para se indicar um género ou profissão, tendo em conta as disponíveis.

Quanto ao website **Welie**, utilizaram-se os seguintes *design patterns*:

- **Action Button** – O botão “Entrar” permite ao utilizador realizar a ação principal no contexto da página atual, pelo que é realçado com recurso a uma cor e um formato distintos (retângulo laranja com cantos arredondados). Para além disso, o *label* do botão inclui o verbo da ação.
- **Registration** – A página de registo fornece ao utilizador a possibilidade de armazenar os seus dados pessoais, para posterior utilização.

5.1.6 Login

A página de *login* também foi desenhada com um formato *standard* e semelhante à do registo, permitindo a autenticação com uma conta Facebook ou Google. O *mockup* desta página é apresentado na figura 20.

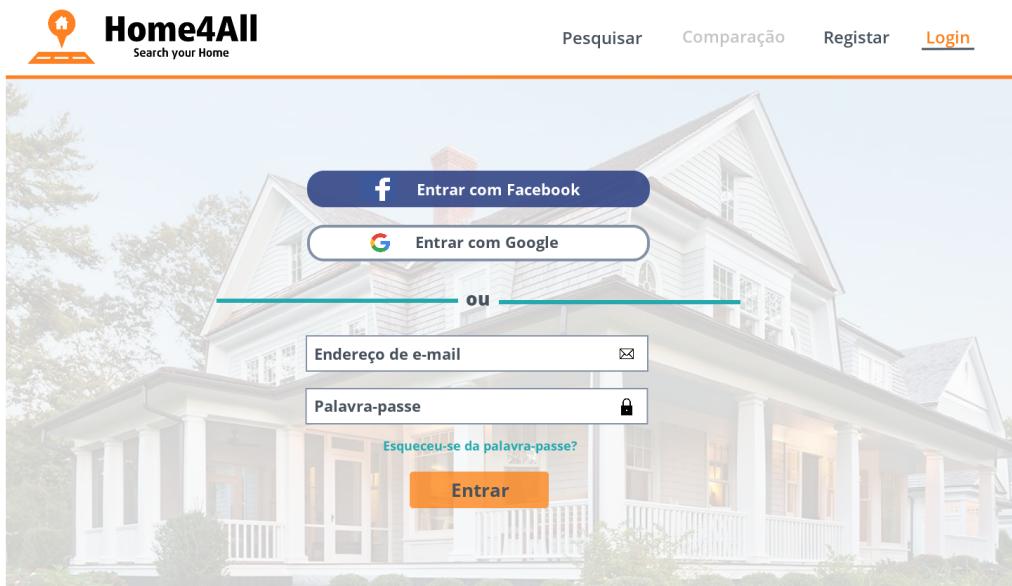


Figura 20: *Mockup* da página de *login*.

Nesta página sobressai-se a aplicação dos seguintes princípios de usabilidade:

- **Learnability:** *Synthesizability* – Após clicar no botão “Entrar” (*login*), será indicada uma mensagem de sucesso ou insucesso. No primeiro caso, será de seguida apresentada a página com os dados pessoais do utilizador.
- **Learnability:** *Familiarity* – O sistema de *login* é semelhante ao de outros *websites*.

- **Flexibility: Generalizability/Consistency** – Os campos de *login* têm a mesma formatação que os respetivos campos de registo. Salienta-se que o botão de submissão do formulário destas páginas é igual, devido ao estado do sistema passar a ser o mesmo nos dois casos (utilizador autenticado), caso a respetiva ação seja efetuada com sucesso.
- **Robustness: Observability** – O botão de *login* está visível em ecrãs médios e grandes. No caso de ecrãs pequenos, o botão aparece imediatamente a seguir ao respetivo formulário, de preenchimento obrigatório.

Relativamente aos padrões de *design* do website **Designing Interfaces**, não se encontrou nenhum que fosse útil aplicar nesta página.

Quanto ao website **Welie**, optou-se por utilizar os seguintes *design patterns*:

- **Action Button** – O botão “Entrar” permite ao utilizador realizar a ação principal no contexto da página atual, pelo que é realçado com recurso a uma cor e um formato distintos (retângulo laranja com cantos arredondados). Para além disso, o *label* do botão inclui o verbo da ação.
- **Login** – A página de *login* permite que o sistema identifique o respetivo utilizador, de forma a possibilitar o recurso a mais funcionalidades e a realização de pesquisas mais personalizadas.

5.1.7 Página Principal (com *login*)

Após a autenticação, as páginas de pesquisa e comparação previamente apresentadas continuam acessíveis, alterando apenas ligeiramente o menu principal. Através deste, passam a ser acessíveis as seguintes páginas:

- “Meus Anúncios” – permite a consulta, alteração, adição e remoção de anúncios efetuados pelo utilizador autenticado;
- “Notificações” – permite o acesso direto à página de novos imóveis que cumpram as configurações especificadas;
- “Dados do Perfil” – permite a consulta e alteração de dados diretamente relacionados com o perfil do utilizador autenticado;
- “Configurações” – permite a consulta, alteração, adição e remoção de configurações de notificação;
- “Favoritos” – permite a consulta de páginas marcadas pelo utilizador autenticado e reorganização da estrutura de armazenamento das mesmas.

A página principal do website, agora com autenticação, é apresentada no *mockup* da figura 21.

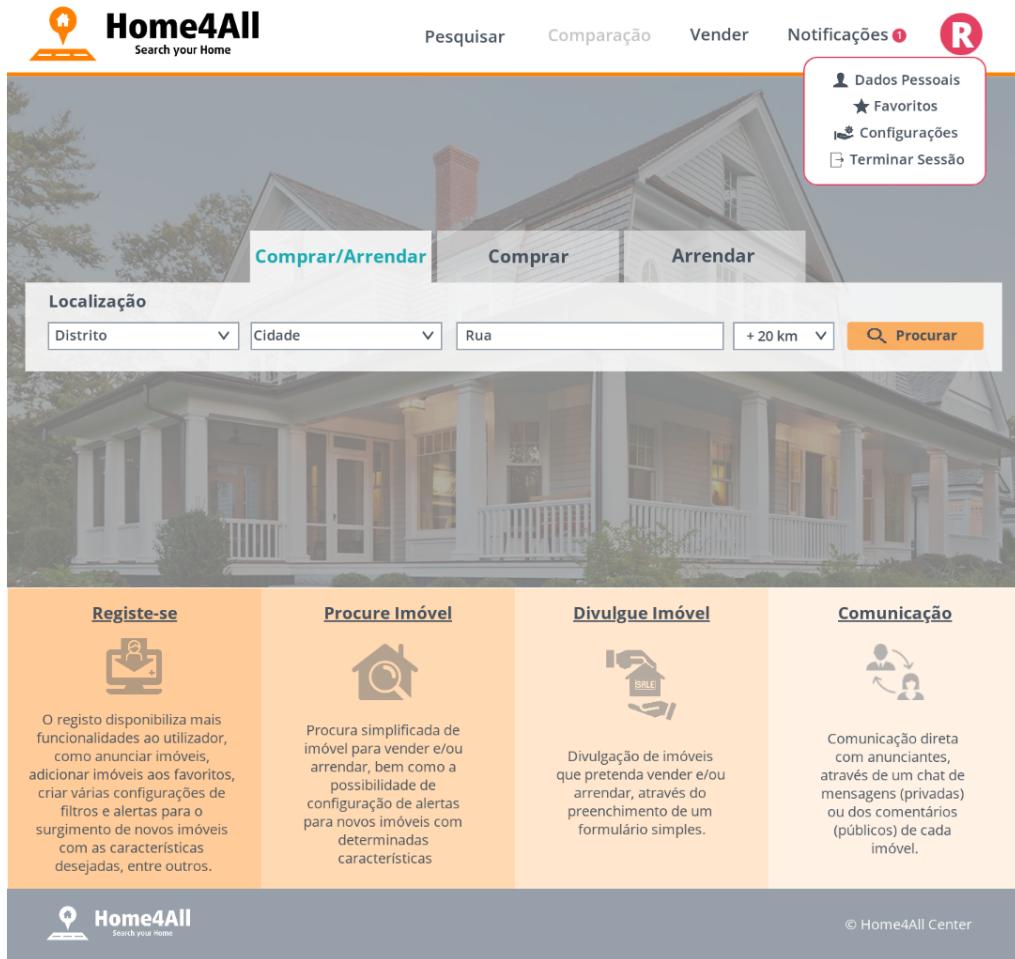


Figura 21: *Mockup* da página principal, com *login* efetuado.

Esta página principal é igual à previamente apresentada, tendo apenas alterações no conteúdo do menu principal. Assim, ainda se aplicam os mesmos princípios e padrões de *design* apresentados na secção 8.

Contudo, podem-se salientar novos princípios de usabilidade aplicáveis, no contexto do menu principal após autenticação:

- **Learnability: Familiarity** – A apresentação de novas notificações com um número realçado e do menu de utilizador com a letra inicial do respetivo nome são técnicas comumente utilizadas noutros *websites*.
- **Robustness: Observability** – As notificações são apresentadas através de um ícone que indica quantas falta visualizar (persistente). Para além disso, pretende-se emitir um sinal sonoro se a notificação surgir enquanto o utilizador estiver autenticado.

5.1.8 Dados do Perfil

Relativamente à página com os dados do perfil do utilizador autenticado, começa-se por apresentar as informações pessoais desse utilizador em campos editáveis, permitindo a consulta e alteração das mesmas simultaneamente. De seguida, surge uma secção com as estatísticas de venda do perfil, cujo conteúdo só é visível após solicitação por parte do utilizador.

O *mockup* desta página é apresentado na figura 22, onde foi solicitada a visualização das estatísticas de venda.

Home4All
Search your Home

Pesquisar Comparação Vender Notificações 1 R

Nome: Rita Fernandes Género: Feminino Data de Nascimento: 23/12/1995

Email: example@email.com Password: *****

Registo efetuado sem conta externa. Ocupação: Estudante Contacto: 912 345 678

Informações Estatísticas de Venda

Quantidade de vendas absolutas efetuadas entre 01/02/2019 e 10/02/2019

Data	Número de vendas
1/3/19	1
3/3/19	2
4/3/19	1
5/3/19	3
7/3/19	2
8/3/19	1
9/3/19	2
10/3/19	1

Percentagem de vendas em relação ao disponibilizado por mês, entre 01/02/2019 e 01/11/2019

Meses	Vendas por imóveis disponibilizados (%)
Feb 19	~50
Mar 19	0
Apr 19	~20
May 19	~65
Jun 19	~120
Jul 19	~100
Aug 19	~70
Sep 19	~80
Oct 19	~150
Nov 19	~10

© Home4All Center

Figura 22: Mockup da página com os dados pessoais do utilizador e estatísticas de venda.

Neste caso, os princípios de usabilidade presentes são os seguintes:

- **Learnability: Synthesizability** – Quando se clica no botão para alteração dos dados, aparece uma mensagem de sucesso ou insucesso.
- **Learnability Familiarity** – A forma de utilização do *chat* privado é semelhante à de outros *websites*, bem como o respetivo sistema de notificações.
- **Flexibility: Substitutivity** – O contacto do utilizador pode ser escrito com ou sem o respetivo indicativo de país.
- **Robustness: Observability** – Os botões para alterar dados e apresentar estatísticas de venda estão visíveis em ecrãs de tamanho médio e grande. Caso o tamanho do ecrã seja reduzido, o botão aparece imediatamente a seguir ao respetivo formulário, como de costume. As estatísticas de venda são consideradas informações adicionais, pelo que não lhes é atribuída demasiada importância. O *chat* privado está sempre visível, numa posição fixa e independente do *scroll*). As notificações no *chat* funcionarão da mesma forma que as notificações de imóveis.

Dos *design patterns* do website *Designing Interfaces*, recorreu-se aos apresentados de seguida para esta página:

- **Extras On Demand** – Recorreu-se a este padrão para apresentar as informações estatísticas de venda. De facto, como estas são informações adicionais sobre o perfil, considerou-se que estes dados apenas deveriam ser apresentados mediante solicitação por parte do utilizador.
- **Input Prompt** – A utilização de *prompts* para indicar ao utilizador o que preencher, neste caso, deve-se ao facto dos dados já estarem inseridos por defeito. Nas situações anteriores, o utilizador deveria inserir os dados em campos vazios, pelo que se considerou suficiente que a informação sobre o que preencher apenas estivesse lá enquanto o utilizador não tivesse preenchido o respetivo campo.
- **Dropdown Chooser** – A ocupação do utilizador pode ser selecionada de um *dropdown*, que contém uma coleção variada de valores que podem ser escolhidos.
- **Edit-in-Place** – Os campos referentes aos dados pessoais do utilizador são editáveis, servindo os propósitos de consulta e edição.

Relativamente ao *website Welie*, conseguiram-se utilizar os seguintes padrões de *design*:

- **Action Button** – O botão “Alterar Dados” permite ao utilizador realizar uma ação relevante no contexto da página atual, pelo que é realçado com recurso a um ícone, bem como a cor e formato distintos (retângulo laranja com cantos arredondados). Para além disso, o *label* do botão inclui o verbo da ação.
- **Date Selector** – Caixa de texto editável, que também permite ao utilizador selecionar a data a partir de um calendário gráfico.

5.1.9 Favoritos

Em relação à página dos favoritos, criou-se um painel no qual se pode visualizar a hierarquia de diretorias existentes. Num segundo painel pode-se visualizar os imóveis armazenados na diretoria selecionada.

O *mockup* da página dos favoritos é apresentado na figura 23.



Figura 23: *Mockup* da página dos favoritos.

Entre os vários *design patterns* do *website Designing Interfaces*, apenas foi utilizado o **Two-Panel Selector**. Recorreu-se a este padrão para desenhar dois painéis lado a lado, onde

o primeiro mostra um conjunto de items (diretorias), que o utilizador pode selecionar conforme pretender, e o segundo mostra o conteúdo do item selecionado.

Quanto ao *website Welie*, optou-se por utilizar o *design pattern Paging*. Caso existam muitos imóveis associados à diretoria selecionada, será utilizado um sistema de *paging* no segundo painel para facilitar a navegação por parte do utilizador.

5.1.10 Configuração de Notificações

Na página de configuração das notificações optou-se por recorrer ao mesmo sistema da página dos favoritos. Utilizaram-se dois painéis lado a lado, sendo que o primeiro apresenta todas as configurações existentes, enquanto que o segundo apresenta um formulário com as especificações dessa configuração, que podem ser alteradas.

Na figura 24 é apresentado o *mockup* da página de configuração de notificações.

Configurações

- Quartos Porto
- Apartamentos Porto B... (selecionado)
- Casas Algarve

+ Adicionar

Filtros: limpar

- Imóvel**: Quarto, Apartamento, Vivenda
- Tipologia**: T0, T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T10+
- Operação**: Comprar, Arrendar
- Preço**: 150 a 300 € / Mês ou 150 000 a 300 000 €
- Avançados**: Número de clientes envolvidos no negócio: 1 pessoa
- Tipo de quartos**: Indicar quantidades
 - Individual, Casal, Múltiplo com 1 pessoa(s)
 - Partilhado, Não partilhado
 - Com WC, Sem WC
- Partilhar com**: Estudante, Trabalhador, Fumador, Não fumador, Animais de estim., S/ animais de estim.
- Imóvel**: Mobilado, Não mobilado, Acesso total, Restrições de acesso

Ordenação

- Preço (crescente), Preço (decrescente)
- Distância (crescente), Distância (decrescente)
- Data Publicação (mais recentes)
- Data Publicação (mais antigos)
- Atividade Anunciante (mais recentes)
- Atividade Anunciante (mais antigos)

Alterar Dados

Home4All Search your Home

© Home4All Center

Figura 24: *Mockup* da página com as configurações das notificações em vigor.

Em termos de princípios de usabilidade aplicados, destacam-se os seguintes:

- **Learnability: Predictability** – Quando o utilizador termina de inserir ou alterar o nome da configuração, o sistema indica erro se este já existir. O mesmo procedimento é seguido para a localização, quando esta não é encontrada, e para a distância ou o preço, caso sejam inválidos.
- **Learnability: Synthesizability** – Após os dados de uma configuração serem alterados ou se adicionar uma nova configuração, será apresentada uma mensagem de sucesso ou insucesso.
- **Learnability: Consistency** – São utilizados os mesmos campos e nomes (localização, filtros e ordenação) na página de pesquisa e nas configurações, o que facilita a previsão do tipo de resultados que serão obtidos.
- **Learnability: Familiarity** – O preenchimento de formulários é comumente utilizado quando se pretende obter determinadas informações do utilizador.
- **Learnability: Generalizability/Consistency** – A forma de apresentação e de consulta dos favoritos e das configurações é igual, o que facilita a aprendizagem do utilizador.
- **Flexibility: Substitutivity** – A distância pode ser escrita, como um inteiro (que é interpretado como estando em quilómetros), ou pode ser selecionada de entre as opções disponíveis.
- **Robustness: Observability** – O botão para alterar dados não está sempre visível, sendo necessário fazer *scroll* para o visualizar. Contudo, desenhou-se a interface desta forma para garantir que o utilizador passa por todos os filtros possíveis, antes de alterar uma configuração.

Dos *design patterns* do *website Designing Interfaces*, recorreu-se aos apresentados de seguida para a página das configurações:

- **Two-Panel Selector** – Recorreu-se a este padrão para desenhar dois painéis lado a lado, onde o primeiro mostra um conjunto de items (configurações), que o utilizador pode selecionar conforme pretender, e o segundo mostra as especificações do item selecionado.
- **Dropdown Chooser** – O distrito, a cidade e a distância podem ser selecionados de um *dropdown*, que contém uma coleção variada de valores que podem ser escolhidos.
- **Edit-in-Place** – Os campos referentes ao nome da configuração, localização procurada e filtros associados são editáveis, servido os propósitos de consulta e edição.

Relativamente ao *website Welie*, conseguiram-se utilizar os seguintes padrões de *design*:

- **Action Button** – O botão “Alterar Dados” permite ao utilizador realizar uma ação relevante no contexto da página atual, pelo que é realçado com recurso a um ícone, bem como a cor e formato distintos (retângulo laranja com cantos arredondados). Para além disso, o *label* do botão inclui o verbo da ação.
- **Autocomplete** – O campo da rua apresenta sugestões para completar o que está escrito, conforme o utilizador vai escrevendo.
- **Overview by Detail** – No primeiro painel é apresentada uma vista geral sobre todas as configurações, identificadas pelo respetivo nome. No segundo, é indicada a especificação da configuração selecionada. Desta forma, o utilizador consegue inspecionar as várias configurações, sem ter de mudar de página constantemente para voltar à vista geral.
- **Form** – Utilizou-se um formulário para solicitar ao utilizador a introdução das especificações de cada configuração de notificação (nome, localização e filtros).

5.1.11 Anunciar Novo Imóvel

Relativamente à página de anúncio de um novo imóvel, optou-se por utilizar um formulário que o anunciante pudesse preencher com todas as informações necessárias sobre o imóvel. O respetivo *mockup* é apresentado na figura 25.

A interface de usuário para anunciar um novo imóvel no site Home4All. No topo, há uma barra com o logo "Home4All" e o lema "Search your Home", uma barra de navegação com links para "Pesquisar", "Comparação", "Vender" (destacado em azul), "Notificações" e uma notificação com o número 1, e uma barra com uma letra "R".

O formulário principal é intitulado "Novo Imóvel".

Dados do Imóvel:

- Nome:** Casa Apartamento
- Fotografias:** Exibe uma galeria de 12 fotos de interiores e exteriores do apartamento.
- Descrição:** Texto: "Apartamento de férias totalmente equipado com terraço no piso térreo de uma casa, moderno e confortável em um ambiente calmo. Parking gratuito e paragem de autocarro. Perto do centro da cidade."
- Informações de Imóvel:** Tipologia: T3, Área: 400 m², Localização: Distrito/Cidade/Rua, Contactos: Contato Telefónico ou E-mail.
- Mobilidade:** Sim, Disponível a partir de: 20/03/2019.
- Operação:** Vender/Arrendar, Preço: 29 € / Noite ou 29 000 000 €.

Renda Inclui: [clicar sobre para incluir/excluir] Exemplos: Água, Eletricidade, TV Cabo, Internet, Condomínio, Telefone, Limpeza.

Tipo de Inclinos Procurados: Género: Indiferente, Idade: Min - Max anos, Ocupação: Estudantes, Trabalhador-estudante, Trabalhador, Desempregado, Fumador: Indiferente, Com Animais: Indiferente.

Divisões ou Equipamentos Incluídos: [clicar sobre para incluir/excluir] Exemplos: Cozinha, Microondas, Máq. Lavar Loiça, Fogão, Frigorífico, Máq. Lavar Roupa, Televisão, TV cabo, Wi-Fi, Computador, Ginásio, Piscina, Elevador, Aquecimento central, Ar Condicionado, Garagem, Estacionamento gratuito.

Ações: Botão "+ Criar Imóvel" e rodapé com o logo "Home4All" e o lema "Search your Home", e o copyright "© Home4All Center".

Figura 25: *Mockup* da página onde um anunciante introduz os dados de um novo imóvel.

Nesta página conseguiu-se aplicar os princípios de usabilidade apresentados de seguida:

- **Learnability: Predictability** – Quando o utilizador termina de inserir o nome do imóvel, o sistema indica erro se este já existir. O mesmo procedimento é seguido para a

localização, quando esta não é encontrada, e para a área, contacto, preço ou idade, caso sejam introduzidos valores inválidos.

- **Learnability:** *Synthesizability* – Após o utilizador adicionar o imóvel, o sistema devolverá uma mensagem de sucesso ou insucesso.
- **Learnability:** *Familiarity* – O preenchimento de formulários é comumente utilizado quando se pretende obter determinadas informações do utilizador.
- **Learnability:** *Generalizability/Consistency* – A ordem de apresentação dos campos no formulário é a mesma que quando se consulta a página de um imóvel, o que permite uma maior familiarização com as informações solicitadas, por parte do utilizador.
- **Flexibility:** *Substitutivity* – A distância pode ser escrita, como um inteiro (que é interpretado como estando em quilómetros), ou pode ser selecionada de entre as opções disponíveis.
- **Robustness:** *Observability* – O botão para criar imóvel não está sempre visível, sendo necessário fazer *scroll* para o visualizar. Contudo, desenhou-se a interface desta forma para garantir que o utilizador passa por todos os campos que é necessário indicar antes de adicionar o imóvel. De facto, caso o utilizador não tenha preenchido algum campo obrigatório, será avisado ao tentar criar o imóvel. No entanto, existem alguns campos que já têm um valor por defeito, pretendendo-se que o utilizador os analise antes de submeter o formulário.

Entre os padrões de *design* do website *Designing Interfaces*, utilizaram-se os seguintes:

- **Fill-in-the-Blanks** – Utilizou-se este padrão para se indicar um intervalo de preços e a quantidade de pessoas no quarto múltiplo.
- **Dropdown Chooser** – O distrito e a cidade podem ser selecionados de um *dropdown*, que contém uma coleção variada de valores que podem ser escolhidos.

Quanto ao website *Welie*, utilizaram-se os seguintes *design patterns*:

- **Action Button** – O botão “Criar Imóvel” permite ao utilizador realizar a ação principal no contexto da página atual, pelo que é realçado com recurso a uma cor e um formato distintos (retângulo laranja com cantos arredondados). Para além disso, o *label* do botão inclui o verbo da ação.
- **Autocomplete** – O campo da rua apresenta sugestões para completar o que está escrito, conforme o utilizador vai escrevendo.
- **Thumbnail** – Imagens dos imóveis mais pequenos são apresentadas para permitir ao anunciante verificar de forma direta quais as fotografias cujo *upload* já efetuou.
- **Date Selector** – Caixa de texto editável, que também permite ao utilizador selecionar a data a partir de um calendário gráfico.
- **Form** – Utilizou-se um formulário para solicitar ao utilizador a introdução das informações específicas sobre o imóvel a anunciar.

5.2 Técnicas de Avaliação da Usabilidade

A usabilidade foi um dos aspetos fundamentais que acompanharam o desenvolvimento do projeto desde do início até ao fim. O objetivo passou por criar uma aplicação que fosse intuitiva e fácil de utilizar. No entanto, a opinião da equipa de desenvolvimento é claramente distorcida e, por isso, foi importante recorrer a métodos credíveis para que a avaliação pudesse ser confiável.

Desta forma, a estratégia adotada passava por realizar dois momentos de avaliação, um utilizando o *mockup* inicial da interface e outro quando a interface estiver implementada a ponto de poder ser testada. Por um lado, numa primeira fase consegue-se recolher *feedback* de forma a ajustar o desenvolvimento em si. Por outro lado, a segunda avaliação permite tirar conclusões à cerca do trabalho desenvolvido e realizar pequenos ajustes se assim for necessário.

Além disso, cada momento de avaliação vai incorporar dois tipos de testes diferentes. Destes, um será um teste de usabilidade onde vários indivíduos serão monitorizados, enquanto executam várias tarefas previamente escolhidas. No fim, cada participante irá responder ao questionário *System Usability Scale* (SUS). Deste teste espera-se recolher *feedback* relacionado com a eficácia, eficiência e satisfação na utilização da ferramenta. O segundo teste é o *cognitive walkthrough* e já utiliza um método analítico. Este, realizado pelos elementos da equipa de desenvolvimento, permitirá identificar possíveis problemas e a respetiva severidade.

As secções que se seguem descrevem em pormenor o processo utilizado para avaliar a usabilidade do projeto, assim como os resultados obtidos nesses mesmos testes.

5.2.1 Método empírico - Teste de usabilidade

5.2.1.1 Objetivos

O primeiro passo no método empírico está relacionado com a definição dos objetivos que se pretende atingir ao realizar o teste em si. Desta forma, existe três grandes componentes que devem ser abordadas, nomeadamente a **eficácia**:

- registrando o rácio de sucesso/insucesso das diferentes tarefas propostas;
 - quantificando os problemas sentidos pelos utilizadores;
- e a **satisfação**:
- utilizando questionários.

Desta forma, espera-se recolher dados suficientes para tirar conclusões à cerca da usabilidade da interface implementada.

5.2.1.2 Tarefas a realizar

Um aspeto importante nos testes de usabilidade é o estabelecimento das tarefas que vão ser realizadas. Achou-se que o melhor seria utilizar duas tarefas diferentes. Por um lado, se fossem realizadas mais, o tempo necessário para a realização do teste aumentaria e os utilizadores não iriam produzir resultados tão úteis devido ao possível cansaço sentido. Por outro lado, apenas um teste levaria a resultados que iriam estar completamente dependentes do teste em si, o que não permitia ter uma visão generalizada da usabilidade do sistema.

Assim sendo, os cenários escolhidos são:

- adicionar um quarto duplo para arrendar. Características: mobilado, casa de banho partilhada, disponível a partir de 10/08/2019, 300€ por mês. Renda incluí: água, eletricidade, Internet. Tipo de inquilinos procurados: género masculino, estudante, não fumador, sem animais de estimação. Divisões/equipamentos disponíveis: cozinha, micro-ondas, fogão, frigorífico, televisão, wi-fi, elevador, estacionamento gratuito. Arrendatários atuais: 0 animais de estimação, 3 estudantes do género masculino e nenhum é fumador.
- consulte o imóvel que acabou de registrar.
- alterar informações pessoais.
- procurar um quarto em Carnaxide para arrendar com valor máximo de 400€.
- alterar informações pessoais.

De facto, dado que as tarefas são de natureza e contexto diferente, espera-se obter uma ideia mais alargada da usabilidade da ferramenta.

5.2.1.3 Execução do estudo

Após realizada toda a preparação anteriormente especificada, a execução do teste propriamente dita pode ser realizada. Na prática, é dado ao utilizador um contexto detalhado do processo que vai ser realizado. De seguida, os participantes realizam o teste, durante o qual, o observador vai recolhendo as devidas notas. Por fim, o estudo termina com a realização dos questionários definidos na fase de preparação.

De realçar, que será pedido ao participante que vá explicando o seu raciocínio enquanto realiza as tarefas, assim como expresse a sua opinião ao longo do processo.

De facto, com esta tipologia de teste espera-se recolher os dados necessários para retirar conclusões significativas.

5.2.1.4 Processamento dos dados

- Contar as tarefas realizadas com sucesso/insucesso
- Recolher opinião
- Estatísticas do questionário

5.2.2 Método analítico - Inspeção por peritos

5.2.2.1 Objetivos

O método analítico é também incorporado no capítulo da avaliação da usabilidade, mas este tem um objetivo possivelmente diferente. Efetivamente, o método analítico permite prever potenciais problemas de usabilidade e não tanto avaliar a usabilidade em si. Desta forma, consegue-se corrigir os problemas de modo a ter uma avaliação mais positiva da usabilidade.

No caso específico deste projeto o teste escolhido foi o *cognitive walkthrough*, que pretende dar resposta à pergunta: até que ponto o sistema vai guiar um utilizador não treinado na execução de uma dada tarefa?

5.2.2.2 Descrição dos utilizadores

Uma componente importante neste tipo de testes passa por ter em conta o tipo utilizador que vai utilizar a ferramenta. Desta forma, a realização dos testes vai ser feita de modo a simular o raciocínio tido por cada uma das *personas* identificadas em 3.1.2. Desta maneira espera-se obter resultados mais fidedignos.

5.2.2.3 Tarefas a realizar

As tarefas a avaliar neste teste são as que se encontram descritas na secção 5.2.1.2 e, adicionalmente, uma que simule o arrendamento de um quarto.

No entanto, importa realçar que a equipa de desenvolvimento encontra-se responsável por realizar este teste em qualquer momento que achar oportuno, sendo que nesses casos a avaliação não será adicionada neste documento.

5.2.2.4 Execução do estudo

A execução deste teste é bastante simples e consiste em responder a quatro questões:

1. A ação correta é suficientemente evidente para o utilizador? [Sim / Não]
2. O controlo para executar a ação está visível? [Sim / Não]
3. Irá o utilizador associar a ação correta ao controlo? [Sim / Não]
4. Irá o utilizador interpretar de forma correta a resposta do sistema à ação escolhida? [Sim / Não]

De facto, para cada pergunta, caso a resposta seja negativa, então deve ser especificada a severidade do problema encontrado da seguinte forma:

- **Tipo 1:** o problema pode causar confusão ou demora na execução da tarefa.
- **Tipo 2:** o problema pode impedir que o utilizador consiga executar a tarefa sem ajuda.
- **Tipo 3:** o problema impede a execução da tarefa.

Adicionalmente, sempre que possível deve ser acrescentada uma descrição do problema encontrado assim como uma sugestão para este.

Assim sendo, espera-se conseguir prever alguns dos problemas que surjam ao longo do desenvolvimento do projeto e assim obter uma melhor avaliação de usabilidade à partida.

5.2.2.5 Processamento dos dados

De seguida, apresentam-se então os resultados dos testes de usabilidade executados com base nos *mockups* desenvolvidos bem como as taxas de completude das tarefas constituintes do teste respetivamente.

Tabela 1: Resultados do questionário SUS para os *mockups* e respetivos *scores*.

1	2	3	4	5	6	7	8	9	10	TOTAL
3	2	4	2	4	2	4	2	3	1	72.5
4	2	3	1	5	2	4	2	3	2	75
3	3	3	1	3	2	3	2	3	2	62.5
3	2	4	1	5	1	4	1	4	1	85

Tabela 2: Resultados da completude das tarefas realizadas durante os testes e respetivos *scores*.

1	2	3	4	TOTAL
0.5	1	1	1	88%
1	1	1	1	100%
1	1	1	0,5	88%
1	1	1	1	100%

Em suma, apesar dos resultados não revelarem resultados 100% positivos, ainda assim revelaram-se bastante positivos.

6 Desenvolvimento

6.1 *Platform-independent model* (PIM)

Considerando-se o modelo de domínio e os requisitos apresentados, identificaram-se as classes e atributos essenciais neste projeto e desenvolveu-se o diagrama de classes apresentado na figura 26.

A classe `Home4All` constitui a *facade* da camada de negócio, fornecendo uma interface simplificada para toda a lógica associada, implementada pelas restantes classes.

Salienta-se ainda o recurso muito frequentemente a associações por composição ou agregação como relações entre classes. Por exemplo, cada imóvel (`Property`) é associado a um utilizador comum (`Common`) por agregação, uma vez que estas entidades existem de forma independente, apesar de um proprietário (`owner`) criar o imóvel.

Para além disso, criaram-se relações por generalização (herança) com o intuito de captar comportamento comum entre classes. Assim, um utilizador (`User`), por exemplo, poderá ser especificado num utilizador comum (`Common`) ou num administrador (`Admin`). Da mesma forma, um utilizador comum pode ser também um utilizador com conta interna, caso em que terá associada uma *password*. Caso o utilizador comum se tenha autenticado com uma conta externa (Facebook ou Google), então será apenas um `Common`.

Para construir as restantes relações do diagrama de classes seguiu-se o mesmo tipo de raciocínio.

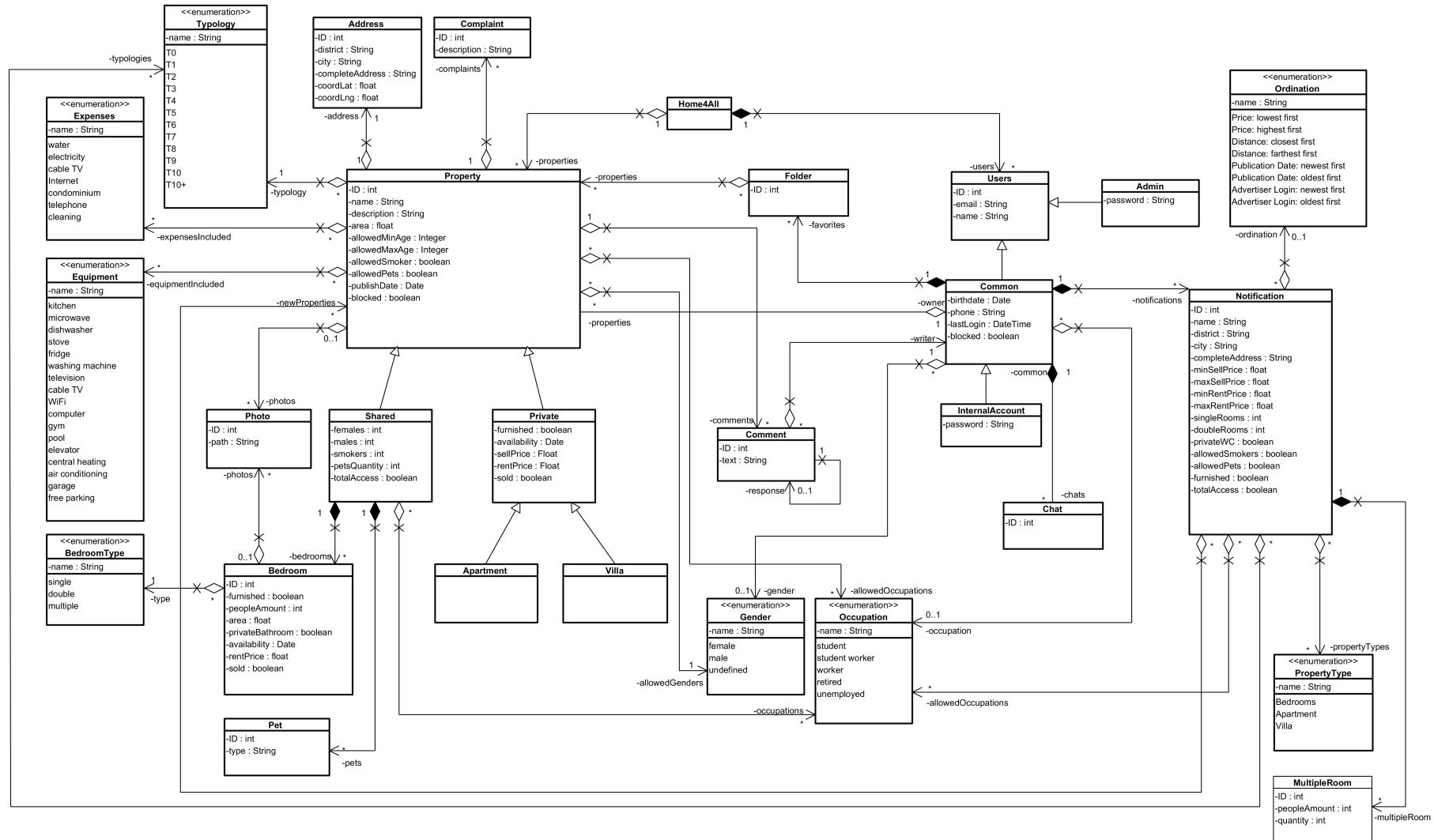


Figura 26: Modelo independente de tecnologia para a aplicação Home4All.

6.2 Escolha das tecnologias

O projeto Home4All consiste na concretização de um *website* relacionado com o setor imobiliário. Assim sendo, é imperativo que sejam escolhidas tecnologias que melhor se adequem ao tipo de serviço que se pretende oferecer.

Primeiramente, tem-se a camada de apresentação, onde existem duas grandes categorias, as *frameworks* que renderizam as páginas no lado do servidor (e.g. JSP) e aquelas que o fazem no lado do cliente (e.g. VueJS). No projeto Home4All optou-se pela segunda opção, pois esta permite uma maior reatividade, além de diminuir o número de pedidos realizados ao *backend* para que este forneça a página a apresentar. Além disso, a *framework* escolhida tem um número elevado de recursos já disponíveis que permitem acelerar de forma significativa o processo de desenvolvimento.

Tendo sido escolhida a tecnologia para a camada de apresentação, torna-se necessário realizar o processo equivalente para a camada de negócio. De facto, o número de opções disponíveis diminui drasticamente para esta camada. A título exemplificativo, tem-se a *framework* Play!, Struts, Spring e a plataforma JEE através de *servlets* e Enterprise Java Beans (EJB). Após alguma pesquisa restou a dúvida entre o Spring e o JEE. No entanto, dada a curva de aprendizagem do Spring e o tempo para execução do projeto, optou-se por utilizar JEE.

Em terceiro lugar, foi necessário escolher uma *framework* para a camada dos dados. Assim como no caso da camada de negócio, também esta dispõe de um menor número de opções quando comparada com a primeira. A título exemplificativo, tem-se textitframeworks como o MyBatis, JPA ou o Hibernate. Neste caso a escolha requereu sobre a última opção uma vez que incorpora mecanismos de gestão de pool, de cache, de transação e, por fim, é compatível com JEE.

Assim sendo, o projeto Home4All foi concebido utilizando VueJS para a camada de apresentação, JEE (*servlets* e EJB's) para a camada de negócio e Hibernate para a camada de dados.

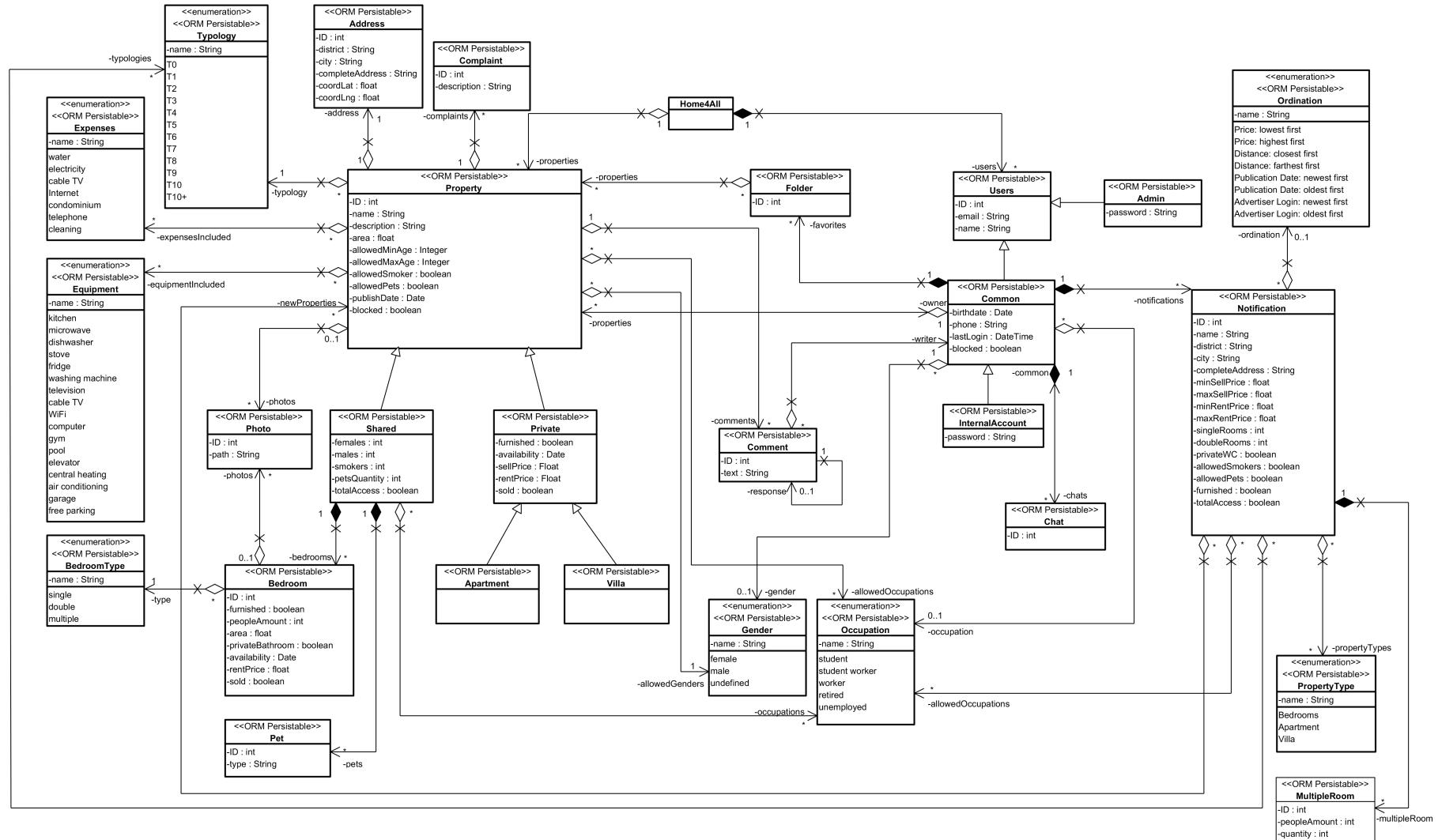
6.3 Platform-specific model (PSM)

Para especificar o diagrama de classes apresentado anteriormente, que é independente das ferramentas utilizadas (PIM), de acordo com a *framework* Hibernate, basta aplicar o *stereotype* ORM **Persistable** às entidades que se pretende persistir. Assim, obtém-se o modelo apresentado na figura 27.

Este diagrama foi desenvolvido com a intenção de se utilizar a ferramenta Hibernate do Visual Paradigm para gerar a base de dados e o código base do projeto. Assim, houve algum cuidado adicional na construção do mesmo, de forma a evitar futuras alterações do diagrama, que implicariam a reconstrução da base de dados e do código até então desenvolvido.

Uma das principais dificuldades encontradas foi a representação das enumerações apresentadas no diagrama PIM. Pretendia-se impedir, ao nível da base de dados, que os campos relativos a enumerações tivessem valores ausentes das mesmas. Para isso, poderia utilizar-se, por exemplo, o tipo de dados ENUM do PostgreSQL para representar estas entidades. Contudo, a geração automática da base de dados a partir do diagrama não permite tal especificidade. Assim, optou-se pela solução mais semelhante a esse tipo de dados, com a criação de uma tabela para cada enumeração, com uma coluna que indica o valor da enumeração e constitui a chave primária da tabela.

Para além disso, não é possível utilizar neste diagrama tipos de dados como Collections, Lists, entre outros, para representar conjuntos de tipos básicos (como Strings, inteiros, etc.). Por isso, existe a necessidade de se criar objetos para esses tipos básicos no diagrama, o que motivou, por exemplo, a criação das classes Photo e Pet.

Figura 27: Modelo dependente da tecnologia **Hibernate** para a aplicação **Home4All**.

Relativamente à arquitetura final do projeto, com EJBs, organizou-se o mesmo em 3 *packages* principais, sendo estes o *web*, o *business* e o *data*, referentes respetivamente à camada de apresentação, negócio e dados. O *package business* contém uma *Facade*, com o intuito de simplificar o acesso aos *Session Beans* contidos no *package beans*. Os *Session Beans*, por sua vez, acedem aos *Entity Beans* presentes no *package entities*. Esta arquitetura geral encontra-se representada no diagrama da figura 28, podendo consultar-se os detalhes referentes aos *packages web*, *business* e *data* nas figuras 29, 30 e 31, respetivamente.

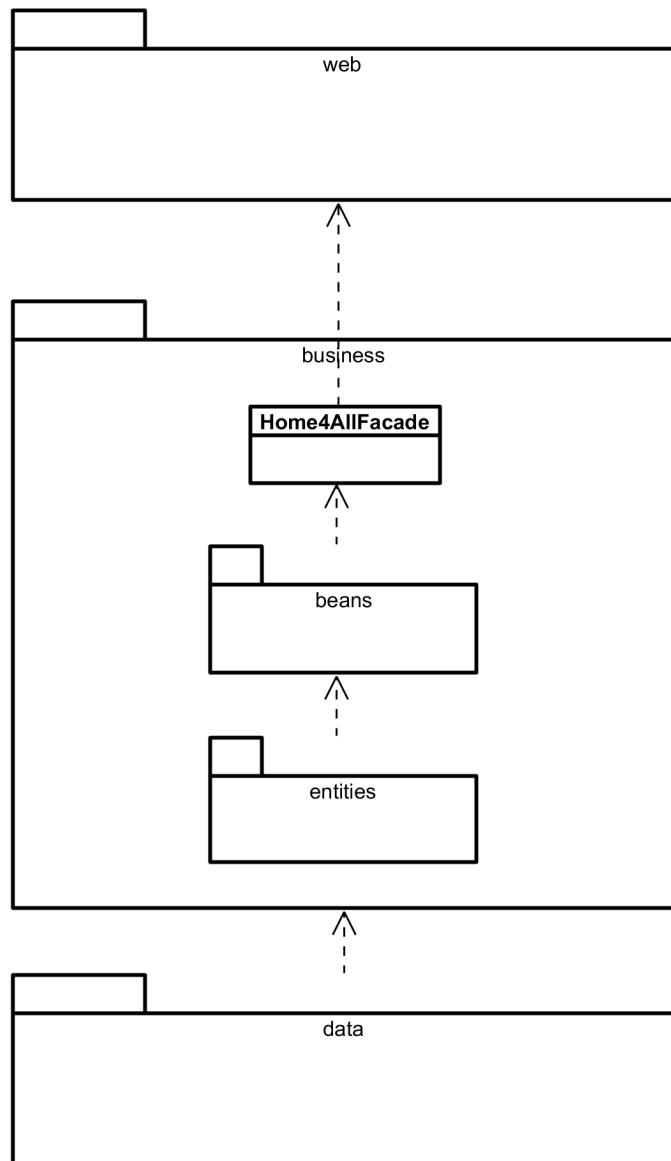


Figura 28: Arquitetura geral da aplicação Home4All.

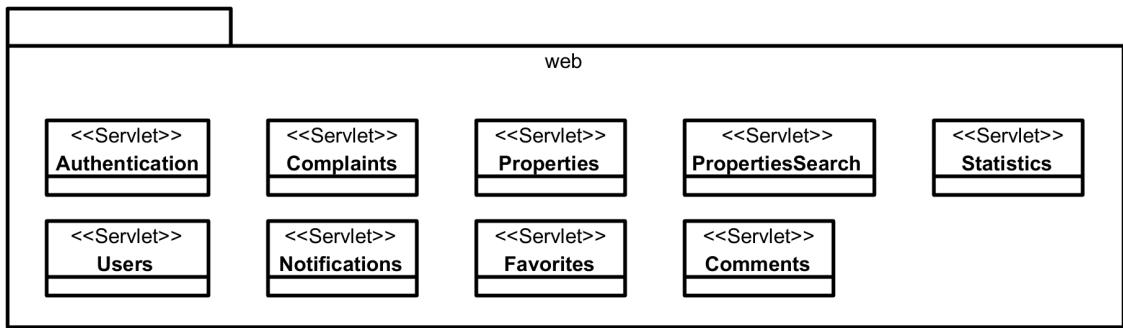


Figura 29: Arquitetura da camada de apresentação da aplicação Home4All (Servlets).

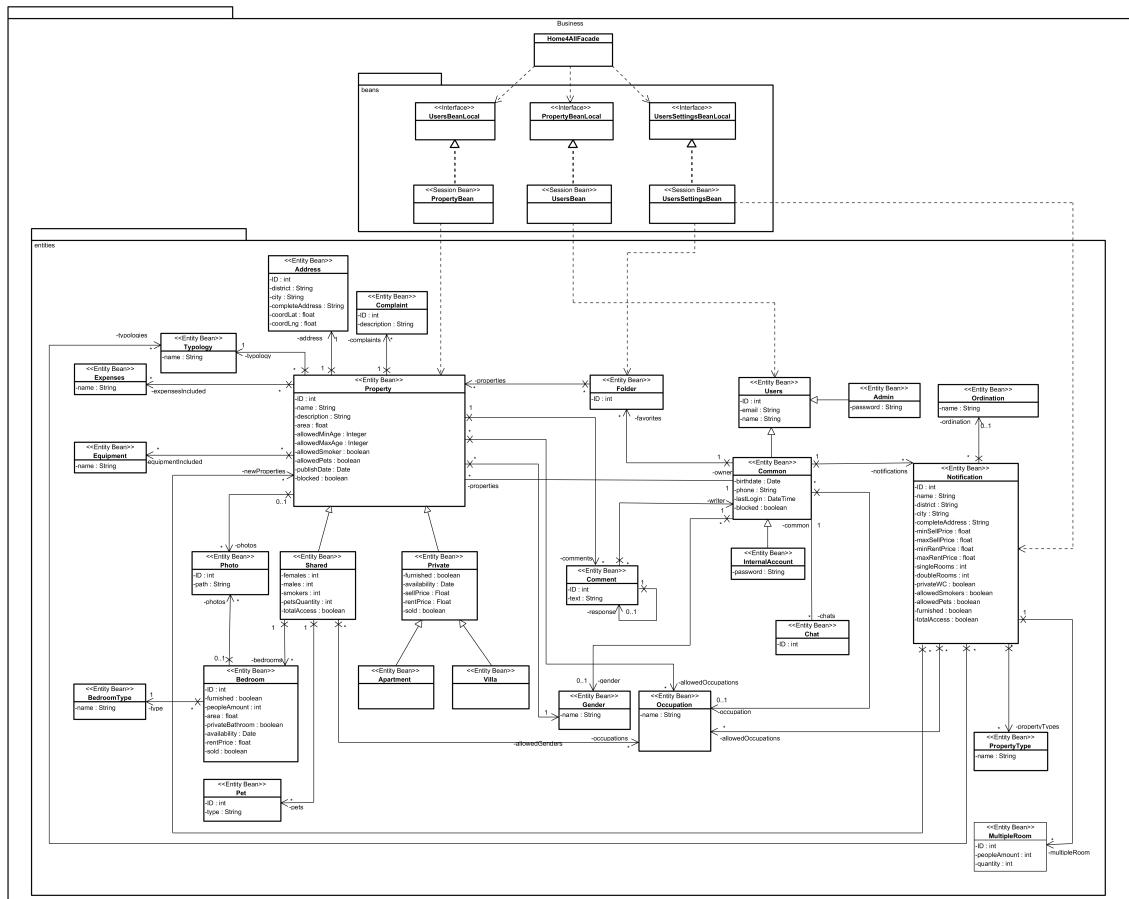


Figura 30: Arquitetura da camada de negócio da aplicação Home4All (EJBs).

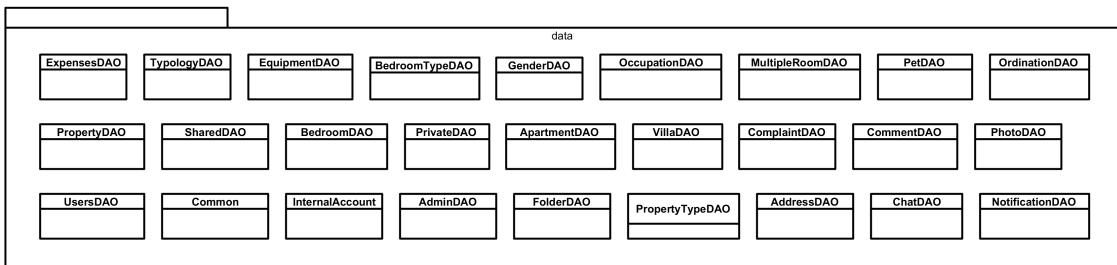


Figura 31: Arquitetura da camada de dados da aplicação Home4All (DAOs).

6.4 Implementação

6.4.1 Dificuldades encontradas e decisões mais importantes

6.4.1.1 Frontend

Um dos desafios que o projeto Home4All coloca, está relacionado com a funcionalidade de pesquisa com base numa dada localização. Desta forma, poder-se-ia deixar que o utilizador escrevesse de forma livre a localização que pretendesse. No entanto, essa estratégia obrigava ao desenvolvimento de um motor de pesquisa, uma vez que a forma do utilizador inserir a mesma localização pode diferir. Assim sendo, foi necessário haver uma compromisso entre permitir que o utilizador escrevesse livremente ou selecionasse um conjunto de cidades já predefinido. Posto isto, a solução encontrada passa por utilizar a API da Google de *autocomplete* de lugares para derivar as localizações que fazem *match* com o que o utilizador se encontra a escrever de forma livre. Esta estratégia permite derivar a mesma informação caso diferentes utilizadores tentem escrever a mesma localização com ligeiras diferenças, mas acabem por escolher a mesma opção do *autocomplete*. Consequentemente, o *backend* apenas precisa de retornar os imóveis que fazem *match* com uma das localizações, sem que exista a necessidade de computar complexos algoritmos de procura.

De realçar que, aparte do desafio mencionado anteriormente, o desenvolvimento do *frontend* decorreu sem dificuldades de maior.

6.4.1.2 Backend

Quanto ao desenvolvimento do *backend*, este foi realizado com o apoio dos diagramas de classes desenvolvidos, mais concretamente os apresentados nas figuras 29, 30 e 31.

Em termos de funcionalidades, a maior dificuldade encontrou-se no facto de diferentes tipos de propriedades(**Private vs Shared**) serem tratados de forma diferente em termos de pesquisa e dados associados. Assim, houve necessidade de dar um tratamento especial à imóvel **Private**, que inclui várias entidades que podem ser arrendadas (**Bedrooms**).

Para além disso, *queries* mais complexas eram difíceis de escrever na linguagem HQL tendo-se optado nesses casos por escrever *queries* nativas (SQL).

Outras dificuldades foram a obtenção de uma imagem representativa do utilizador autenticado e o armazenamento das fotografias dos imóveis.

Para o primeiro problema, optou-se pela utilização da API externa **Gravatar**, que permite o acesso a uma imagem definida pelo utilizador, através do seu email. Caso este não tenha associado uma imagem ao seu email, no **Gravatar**, é apresentada uma imagem por defeito.

Para o segundo problema, ponderou-se o armazenamento das imagens na base de dados. Contudo, chegou-se à conclusão de que esta técnica seria inefficiente, tendo em conta que a obtenção e armazenamento de imagens estariam sujeitos a mecanismos transacionais. Assim, optou-se por armazenar apenas o identificador da imagem na base de dados, e armazenar o próprio documento num sistema de ficheiros. A primeira solução explorada foi o **Samba**, que permite o armazenamento das imagens numa diretoria partilhada, num servidor de base de dados. Contudo, verificou-se que

se obteria melhor desempenho utilizando a ferramenta **Redis**. Desta forma, foi esta ferramenta a escolhida como decisão final para o armazenamento das fotografias dos imóveis.

6.4.2 Funcionalidades implementadas

Tendo em conta todas as dificuldades apresentadas durante o processo de desenvolvimento na secção acima, as funcionalidades inicialmente propostas não foram implementadas totalmente.

Assim, apresenta-se de seguida o diagrama de casos de uso em que se encontram representadas as funcionalidades implementadas na solução final.

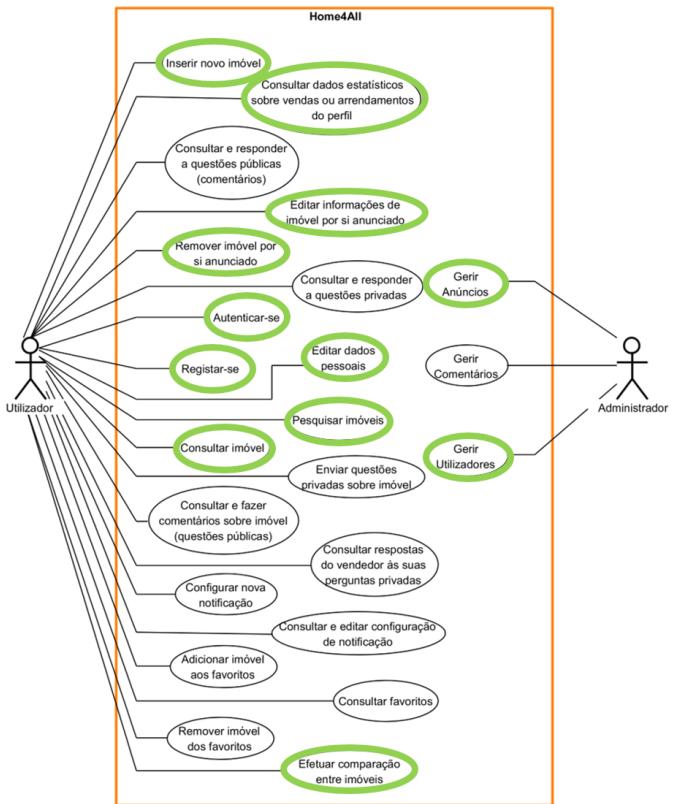


Figura 32: Diagrama de casos de uso implementados na solução final.

A ordem de implementação dos diversos casos de uso inicialmente propostos foi definida tendo em conta uma noção de prioridade inicialmente associada a cada caso de uso. Funcionalidades como a autenticação de um utilizador ou o seu registo, são essenciais para a integridade do sistema, tendo sido assim estas as primeiras a serem implementadas. Da mesma forma, casos de uso que a sua não implementação faça com que o produto apresentado perca totalmente o seu propósito, foram também priorizados. Por exemplo, a gestão de anúncios, pesquisa de imóveis e a comparação encontram-se diretamente relacionados com funcionalidades básicas de uma plataforma de imóveis pelo que foram considerados prioritários também em relação aos restantes casos de uso.

Em suma, todas as funcionalidades diretamente relacionadas com imóveis encontram-se implementadas, ficando apenas em falta funcionalidades que apenas tornariam o contacto entre anunciante e cliente mais simples e rápido.

6.4.3 Interface de utilizador

Para o desenvolvimento da camada de apresentação da plataforma, foram tidos como base os *mockups* anteriormente desenhados. Assim, numa fase inicial a construção da interface foi

principalmente orientada à funcionalidade e não tanto à sua apresentação. De seguida, passou-se então à personalização dos componentes para que a apresentação ficasse o mais parecida possível com os *mockups*.

Assim, apresenta-se a interface desenhada para o utilizador.

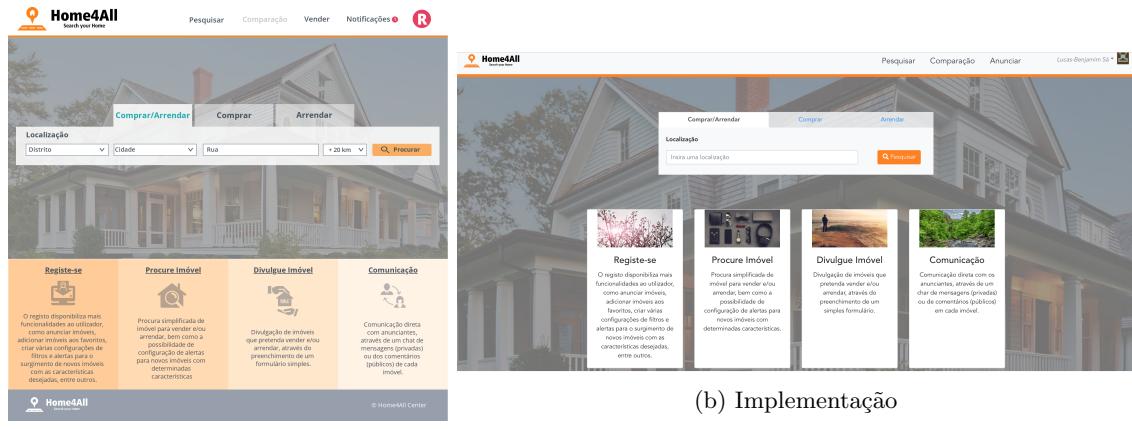


Figura 33: Comparação entre *mockup* e implementação.

Como se pode observar, o resultado final encontra-se bastante próximo do que foi inicialmente idealizado.

Outra das páginas a analisar será a visualização das informações do perfil. Neste caso, o resultado final não foi o mais fiel aos *mockups* desenvolvidos tendo em conta que a apresentação dos campos de informação na versão original não apresenta apenas um campo de informação por linha, enquanto que na versão implementada é esse o caso.

Vejam-se então as duas versões da interface.

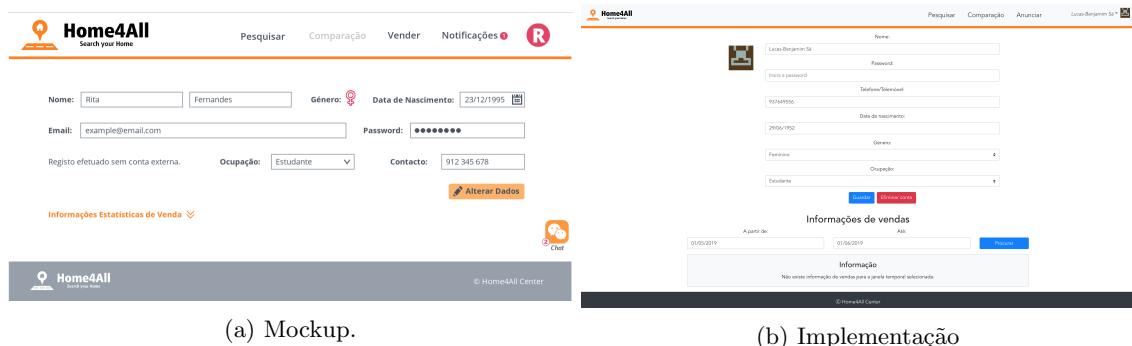


Figura 34: Comparação entre *mockup* e implementação.

De uma forma geral, todas as páginas seguiram praticamente a 100% o aspeto dos *mockups* anteriormente desenvolvidos. A única diferença considerável entre a versão previamente desenhada e a versão final será mesmo a página de visualização da informação do perfil.

7 Deployment

Após a implementação do projeto Home4All, torna-se necessário fazer o *deployment* do mesmo, de forma a que este possa ser usado e testado em situações reais. Deste modo, é imperativo que se analise a melhor estratégia de forma a tirar o máximo proveito dos recursos disponíveis, garantindo elevada disponibilidade, assim como tempos de resposta aceitáveis.

Assim sendo, dado que se trata de uma aplicação *web*, é comum que esta contenha um servidor *web*, um servidor aplicacional e um camada de dados. Na prática esta ferramenta utiliza como *web server* o **Apache**. Para que o *website* seja servido, foi necessário compilar o projeto de *front-end* em **Vuejs** para o modo de produção e coloca-lo na diretoria apropriada do **Apache** para que este ficasse disponível. Além disso, dado que o servidor *web* é o primeiro ponto de entrada no sistema, decisões tomadas neste ponto podem ser decisivas para o desempenho do sistema. Como tal, houve o cuidado de configurar o servidor para que este permitisse balançar a carga pelos servidores aplicacionais existentes. No entanto, um simples balanceamento poderia comprometer o conceito de sessão utilizado na componente aplicacional da ferramenta. Efetivamente, mecanismos de *cache* ou mesmo a gestão de informação relacionada com a sessão do utilizador ficariam perdidos caso o utilizador fosse balanceado para diferentes servidores aplicacionais numa mesma utilização. Por um lado, uma possível estratégia seria sincronizar em tempo real essas informações entre os diversos servidores aplicacionais. Por outro lado, pode-se configurar o servidor *web* para que identifique cada sessão de utilizador e garanta, que após inicializada, é sempre servida pelo mesmo servidor aplicacional. Neste projeto em concreto foi escolhida a segunda opção uma vez que diminui a carga necessária para a sincronização dos servidores aplicacionais. Além disso, é na mesma possível realizar o balanceamento dos pedidos recebidos, sendo que agora este processo encontra-se associado ao conceito de sessão.

Após a configuração dos servidores *web* os clientes possuem acesso ao conteúdo estático do *website*, e como tal, conseguem navegar em grande parte do *website*. No entanto, os pedidos realizados ao *backend* são redirecionados pelo *web server*, mas não obtém qualquer resposta. Desta forma, torna-se necessário configurar os servidores aplicacionais que vão permitir animar a lógica associado à aplicação Home4All e assim devolver ao cliente os dados que este necessita para utilizar a ferramenta na sua plenitude. Este processo foi alcançado através da utilização do servidor aplicacional **Wildfly**. Este é responsável por receber aos pedidos HTTP reencaminhados pelo **Apache** e fornecê-los ao *servlet* respetivo do JEE. Da mesma forma, quando o pedido em está pronto a ser enviado, este realiza o percurso contrário de forma a fazer chegar ao cliente os dados requeridos.

Sendo a camada aplicacional independente da camada dos dados assim como da camada da apresentação, é possível fazer *deploy* da mesma, de forma isolada dos restantes componentes. Assim sendo, a estratégia utilizada passou por gerar um ficheiro WAR da camada aplicacional, de forma a que este contenha todas as bibliotecas necessárias (e.g. **Hibernate**), assim como o servidor aplicacional anteriormente referido, de forma a que a camada se encontre completamente funcional. Após a geração do ficheiro, optou-se por executá-lo num *container docker*, para que este tirasse partido das vantagens dessa tecnologia, nomeadamente no que toca à portabilidade associada à facilidade de realizar o *deploy* em diferentes máquinas, utilizando o mesmo processo.

Por fim, tem-se a infraestrutura da aplicação praticamente operacional, restando apenas a camada de dados. O *deploy* desta última consistiu na inicialização do servidor de **PostgreSQL** e do **Redis**. Após estas operações tem-se a estrutura final da aplicação montada e pronta a ser utilizada pelos clientes finais.

8 Carga aplicacional

8.1 Povoamento da Base de Dados

De forma a testar e demonstrar a carga suportada pela aplicação, é necessário a automatização do povoamento da base de dados. Assim, desenvolveu-se um *script*, em Python, parametrizado com as quantidades e probabilidades dos tipos de dados a inserir. O *output* deste *script* é um ficheiro com um conjunto de *queries* SQL, que efetuam o povoamento da base de dados. Assim, é possível indicar, por exemplo, a quantidade de utilizadores comuns, **InternalAccount**, que se pretende inserir na base de dados, bem como a probabilidade de determinado utilizador ser de um género específico (*female*, *male* e *undefined*).

8.2 Testes de Carga

Após o povoamento da base de dados, prosseguiu-se para a realização de testes de carga, de forma a avaliar o desempenho da aplicação desenvolvida. O objetivo é perceber quantos utilizadores simultâneos a aplicação suporta, com um tempo de resposta aceitável.

8.2.1 Ferramenta utilizada

Para a realização dos testes de carga referidos, analisou-se um vasto leque de ferramentas que permitem efetuar os mesmos. Verificou-se a disponibilidade de dois tipos principais de *software* capazes de realizar estes testes:

1. Realiza a gravação da interação do utilizador com a interface, permitindo a utilização de *think times* mais realistas.
2. Efetua o envio de pedidos HTTP ao servidor aplicacional, havendo necessidade de especificar o tipo e estrutura dos mesmos.

As ferramentas que se sobressaíram, relativamente ao primeiro tipo de *software*, foram o **WebLoad** e o **NeoLoad**. Já para o segundo, destacou-se o **Apache JMeter** e o **Predator**. Entre estas ferramentas, optou-se pelo **Apache JMeter** pelo motivos indicados de seguida:

- **WebLoad**: Não suporta a gravação das interações do utilizador com a interface, quando esta utiliza o **Vue.js**;
- **NeoLoad**: Não suporta, por defeito, a utilização de API's externas, como por exemplo a do *Google Maps*. Além disso, o objetivo não é captar a interação do utilizador com a interface, mas sim testar o limite do servidor aplicacional em termos de carga.
- **Predator**: Dificuldade de instalação da ferramenta no sistema operativo **Windows**.

Relativamente ao **Apache JMeter**, com o auxílio de breves tutoriais, conseguiu-se parametrizar todos os pedidos pretendidos com relativa facilidade. Para além disso, verificou-se que esta ferramenta permite a geração de todas as estatísticas necessárias para avaliar a performance da aplicação, tal como:

- **View Results Tree**: Permite visualizar se os pedidos realizados foram respondidos corretamente ou incorretamente, bem como o conteúdo dos pedidos e respostas HTTP (teste de correção);
- **Graph Results**: Permite ter um visão geral do débito e tempo de resposta dos vários pedidos ao longo do tempo;
- **Response Times Over Time**: Permite visualizar o tempo de resposta dos pedidos ao longo do tempo;
- **Active Threads Over Time**: Permite visualizar as *threads* ativas ao longo do tempo;

Adicionalmente, esta ferramenta permite realizar vários tipos de teste de performance. Para isso é necessário especificar os parâmetros apresentados na figura 35, sendo os que apresentam maior relevância para este projeto explicitados de seguida:

- *Number of Threads*: Representa o número total de utilizadores virtuais que executam o *script* de teste;
- *Ramp-up Period*: Indica quanto tempo demorará para alcançar o número de *threads* definidas;
- *Loop Count*: Indica o número de execuções do *script* de teste, por cada utilizador virtual. Se o *loop count* estiver marcado como *forever*, novas *threads* irão começar até que os testes sejam parados;
- *Scheduler - Duration*: Limita o intervalo de tempo no qual os *scripts* de testes podem ser iniciados.

A partir deste parâmetros é possível definir dois tipos de testes úteis para o cumprimento do objetivo estabelecido inicialmente:

- **Teste 1**: o número de *threads* estabelecido executa tantas vezes como o valor de definido no *loop count* o *script* de teste, sendo estas execuções distribuídas pelo período de *ramp-up*. Os parâmetros utilizados para este tipo de teste são destacados na figura 35 a roxo e amarelo.
- **Teste 2**: o número de *threads* estabelecido representa o valor máximo de utilizadores virtuais que estão a efetuar pedidos em simultâneo. Assim, no início do teste, este número de utilizadores vai crescendo linearmente até ao período de *ramp-up*, momento a partir do qual se mantém constante (no máximo). A duração total do teste é definida pela duração do *scheduler*. Os parâmetros utilizados para este tipo de teste são destacados na figura 35 a roxo e verde. A quantidade de *threads* ativas em cada instante, de acordo com este tipo de teste, é representada no gráfico apresentado na figura 36.



Figura 35: Parâmetros utilizados nos dois tipos de testes.

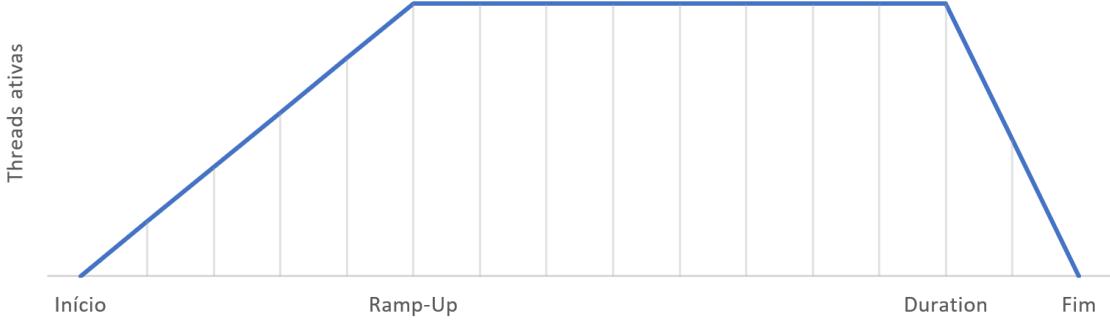


Figura 36: Quantidade de *threads* ativas durante a execução do *script* de **teste 2**.

8.2.2 Parametrização dos Testes

Com a ferramenta **Apache JMeter** é possível definir grupos de *threads*, cada uma responsável pela execução de um conjunto de operações (pedidos HTTP) e pelo estabelecimento dos parâmetros indicados na secção 8.2.1. Assim, optou-se por realizar um conjunto de teste considerados fundamentais na aplicação desenvolvida, sendo estes os seguintes:

- Registo e consulta de dados pessoais:
 - **Registo do utilizador:** Pedido HTTP POST ao endpoint `users`;
 - **Login:** Pedido HTTP POST ao endpoint `authentication`;
 - **Consulta dos dados pessoais:** Pedido HTTP GET ao endpoint `users`;
 - **Logout:** Pedido HTTP POST ao endpoint `authentication`.
- Pesquisa s/ *login*:
 - **Pesquisa:** Pedido HTTP GET ao endpoint `search`.
- Pesquisa c/ *login*:
 - **Login:** Pedido HTTP POST ao endpoint `authentication`;
 - **Pesquisa:** Pedido HTTP GET ao endpoint `search`;
 - **Logout:** Pedido HTTP POST ao endpoint `authentication`.
- Editar dados pessoais e consultar estatísticas:
 - **Login:** Pedido HTTP POST ao endpoint `authentication`;
 - **Consulta dos dados pessoais:** Pedido HTTP GET ao endpoint `users`;
 - **Edição dos dados pessoais:** Pedido HTTP PUT ao endpoint `users`;
 - **Consulta de estatísticas:** Pedido HTTP GET ao endpoint `statistics`;
 - **Logout:** Pedido HTTP POST ao endpoint `authentication`.

Todos os grupos de *threads* previamente referidos exigem a utilização de um conjunto de dados pré-definido. De forma a evitar o envio repetido dos mesmos dados, criaram-se *data sets* em ficheiros `csv`, que são importados para o **Apache JMeter** sendo que cada coluna desse ficheiro corresponde a uma variável a ser utilizada nos pedidos efetuados. Para os vários pedidos, as linhas destes ficheiros `csv` são percorridas sequencialmente.

Uma vez que alguns dos pedidos efetuados requeriam a autenticação do utilizador, que é efectuada através de *cookies*, foi necessário o armazenamento de uma variável especial (`JSESSIONID`),

recolhida aquando da receção da resposta do pedido de *login*. Para a extração desta variável, bem como outras, utilizou-se o *Regular Expression Extractor* do Apache JMeter.

Relativamente aos parâmetros referidos na secção 8.2.1, definiram-se como testes a executar os apresentados nas tabelas 3 e 4.

Tabela 3: Parâmetros para os vários testes realizados com 1 servidor aplicacional.

Tipo do Teste	Number of threads	Ramp-Up Period (s)	Loop Count	Scheduler Duration
1	400	120	2	-
	700			
	750			
2	50	50	<i>Forever</i>	120
	30			
	20			

Tabela 4: Parâmetros para os vários testes realizados com 2 servidores aplicacionais.

Tipo do Teste	Number of threads	Ramp-Up Period (s)	Loop Count	Scheduler Duration
1	100	120	2	-
	400			
2	100	50	<i>Forever</i>	120
	50			

8.2.3 Dificuldades

As maiores dificuldades encontradas nesta fase de avaliação de performance foram a familiarização com a ferramenta de teste de carga, uma vez que nunca nenhum elemento do grupo tinha lidado com uma semelhante previamente, bem como o estabelecimento de toda a infraestrutura e o *deployment* da aplicação (*Docker*).

Relativamente à ferramenta Apache JMeter, sobressaiu-se a dificuldade de aprendizagem de passagem de parâmetros à mesma, mais concretamente, a autenticação (*cookie*), extração de variáveis e pedidos HTTP com parâmetros.

Por fim, é de se referir o surgimento de erros relacionados com a concorrência e utilização de *Hibernate*, que apenas foram detetados nesta fase. Primariamente percebeu-se a necessidade de aumentar o tamanho da *pool* do *Hibernate* de conexões à base de dados, tendo este processo sido efetuado com recurso à biblioteca C3P0. Para além disso, continuaram-se a detetar alguns erros inerentes à possibilidade de execução concorrente, como por exemplo, a ocorrência de *rollbacks*.

8.2.4 Resultados

Com a realização dos testes definidos na tabela 3 e 4, obtiveram-se os resultados apresentados de seguida, nas secções 8.2.4.1 e 8.2.4.2 respetivamente.

Salienta-se que em todos os testes apresentados se verifica uma quantidade de *threads* ativas inicial bastante elevado, relativamente ao resto da execução, devido a esta ser uma fase em que o servidor aplicacional tem de criar sessões do *Hibernate* (conexões à base de dados), bem como instâncias de EJBs e *Servlets*. Assim que estas instanciações terminem, o servidor aplicacional adquire uma maior capacidade de resposta.

8.2.4.1 Arquitetura com um Servidor Aplicacional

Teste do tipo 1 com 400 utilizadores virtuais:

Neste teste obteve-se um débito de 42,10 operações por segundo e um tempo de resposta médio de 82 milissegundos. Para além disso, obteve-se uma quantidade de *threads* ativas aceitáveis (geralmente inferior a 2 *threads*, após estabilizar), como se pode verificar através da figura 37. Adicionalmente, verificou-se que a taxa de insucesso na receção de resposta encontra-se na ordem dos 3,15%.

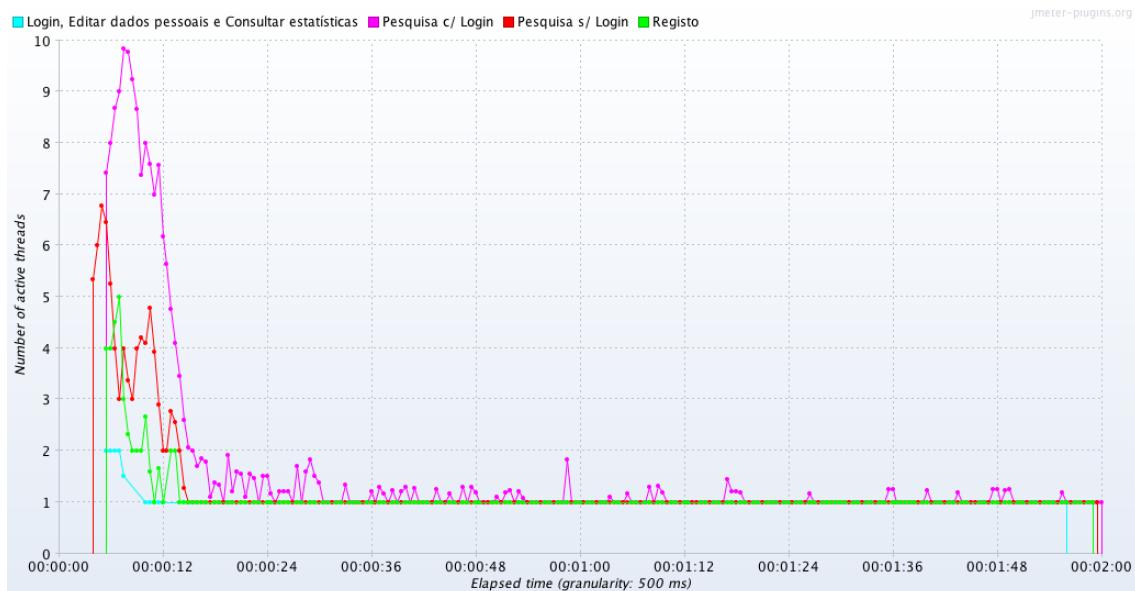


Figura 37: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo 1 com 400 utilizadores virtuais.

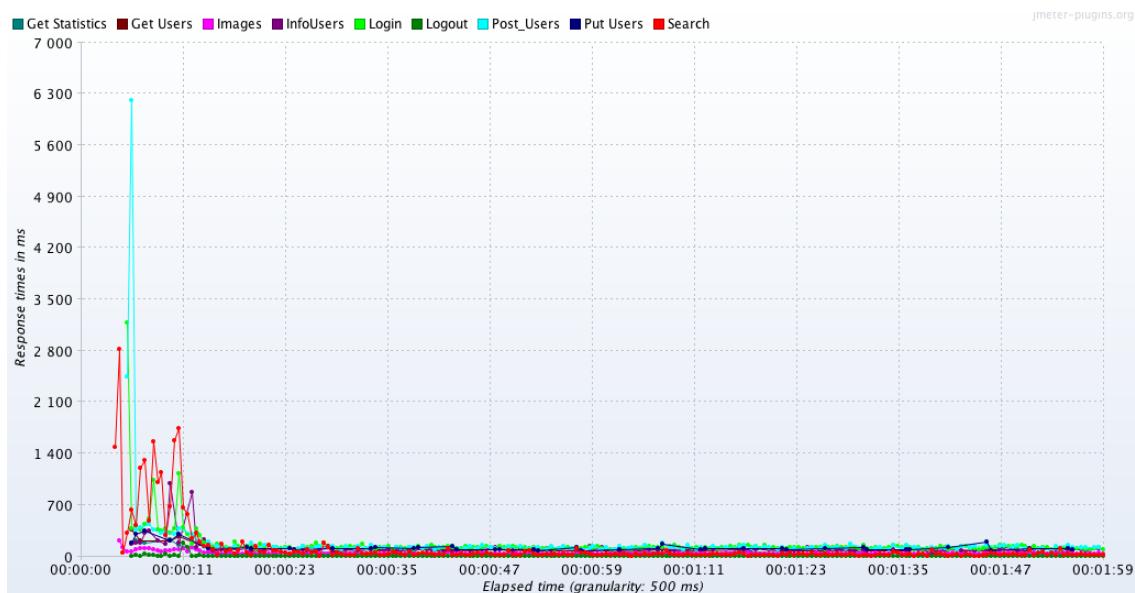


Figura 38: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 1 com 400 utilizadores virtuais.

Tabela 5: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 1 com 400 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	2305	95	3,96%
pesquisa s/ login	1744	56	3,11%
registro	593	7	1,17%
editar info pessoais	249	1	0,40%
Total	4891	159	3,15%

Considera-se que estes resultados são bastantes positivos, pelo que se optou por aumentar o número de utilizadores virtuais para este tipo de teste.

Teste do tipo 1 com 700 utilizadores virtuais:

Neste teste obteve-se um débito de 71,58 operações por segundo e um tempo de resposta médio de 170 milissegundos. Para além disso, obteve-se ainda uma quantidade de *threads* ativas aceitáveis (geralmente inferior a 3 *threads*, após estabilizar), como se pode verificar através da figura 39. Adicionalmente, verificou-se que a taxa de insucesso na receção de resposta encontra-se na ordem dos 3,77%.

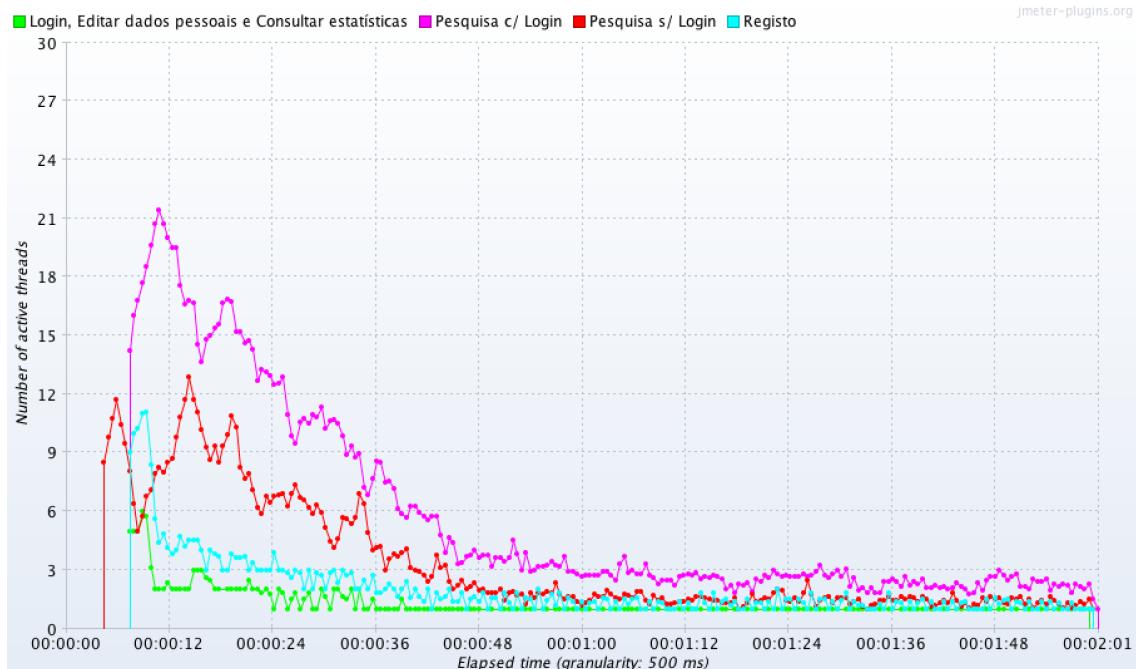


Figura 39: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo 1 com 700 utilizadores virtuais.

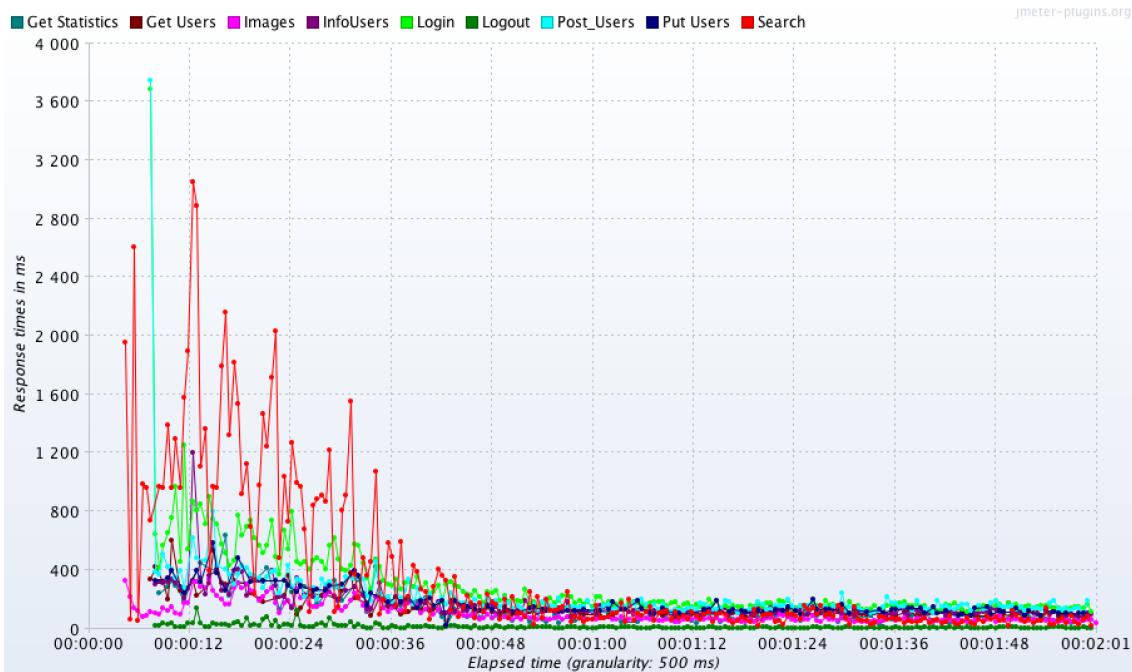


Figura 40: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 1 com 700 utilizadores virtuais.

Tabela 6: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 1 com 700 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	3625	215	5,60%
pesquisa s/ login	2786	94	3,26%
registro	1160	8	0,68%
editar info pessoais	732	8	1,08%
Total	8303	325	3,77%

Estes resultados mantêm-se ainda animadores, contudo é de se realçar que o período de tempo que o servidor aplicacional demorou a estabilizar a quantidade de pedidos respondidos aumentou consideravelmente. Este facto deve-se ao aumento da quantidade utilizadores virtuais, que se irão acumular enquanto o servidor aplicacional inicializa as instâncias de sessões do *Hibernate*, EJBs e *servlets*.

Como estes resultados continuam a ser considerados aceitáveis, mas já se revela um ligeiro acumular de *threads* ativas inicialmente, optou-se por continuar a aumentar a quantidade de utilizadores virtuais, ainda que ligeiramente.

Teste do tipo 1 com 750 utilizadores virtuais:

Neste teste obteve-se um débito de 38,26 operações por segundo e um tempo de resposta médio de 7,28 segundos. O número de *threads* ativas revela-se bastante elevado, dado que o servidor não foi capaz de lidar com a carga à qual foi submetido. Esta, por sua vez, foi-se acumulando até ao final do teste (2 minutos), momento a partir do qual os pedidos deixaram de ser efetuados e o servidor aplicacional começou a conseguir dar resposta aos previamente realizados.

Adicionalmente, verificou-se que a taxa de insucesso na receção de resposta se encontra na ordem dos 28,68%.

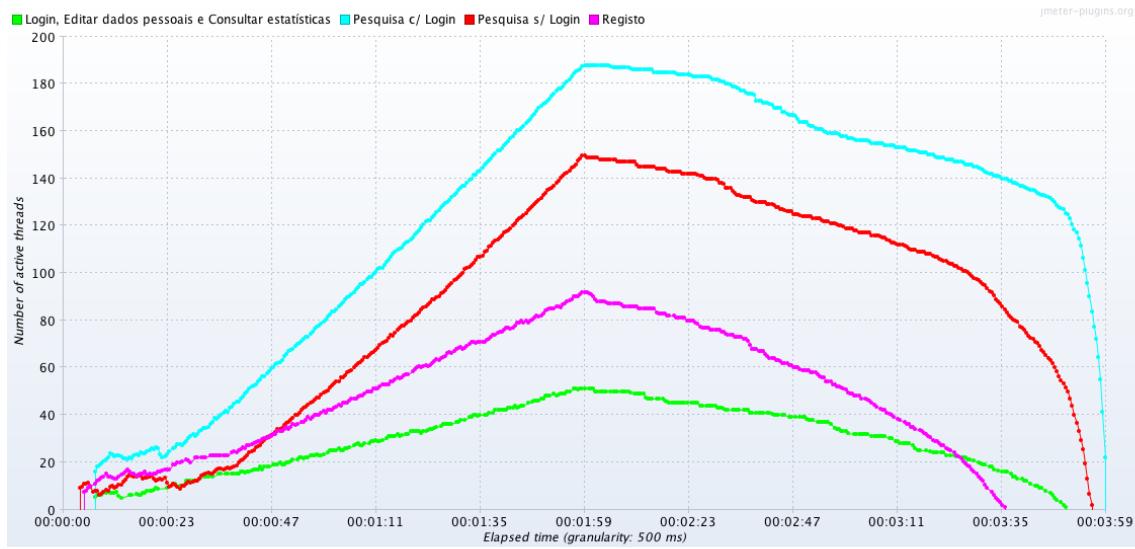


Figura 41: Quantidade de threads ativas ao longo do tempo, para o teste do tipo 1 com 750 utilizadores virtuais.

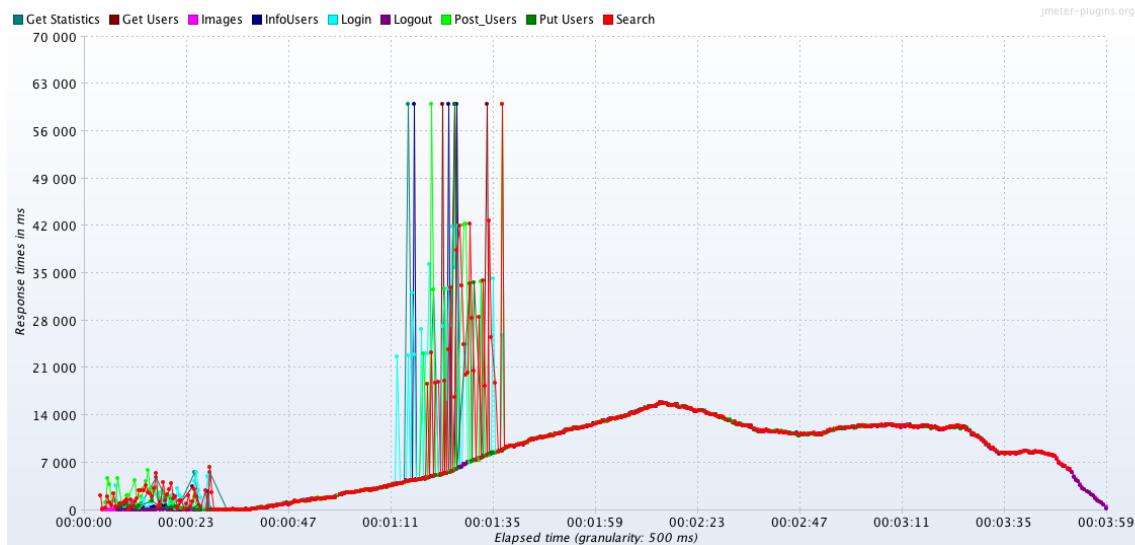


Figura 42: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 1 com 750 utilizadores virtuais.

Tabela 7: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 1 com 750 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	3196	804	20,10%
pesquisa s/ login	2659	341	11,37%
registro	179	1190	86,92%
editar info pessoais	505	295	36,88%
Total	6539	2630	28,68%

Como os resultados obtidos neste teste se revelaram negativos, considera-se que o servidor apenas é capaz de suportar 700 utilizadores virtuais para um teste do tipo 1.

Teste do tipo 2 com 50 utilizadores virtuais:

Agora, com o teste do tipo 2, obteve-se um débito de 31,79 operações por segundo e um tempo de resposta médio de 1,05 segundos. Como se pode verificar no gráfico apresentado na figura 43, a quantidade de *threads* ativas vai aumentando até, aproximadamente, os 50 segundos, momento a partir do qual se mantém constante (somatório igual a 50 utilizadores virtuais, a cada instante). Relativamente ao tempo de resposta, cujo gráfico é apresentado na figura 44, verifica-se que, inicialmente, este valor é aceitável, mas que a partir do momento em que as *threads* ativas atingem e se mantêm no seu valor máximo o servidor deixa de ser capaz de dar resposta a tanta carga.

Adicionalmente, verificou-se que a taxa de insucesso na receção de resposta se encontra na ordem dos 11,94%.

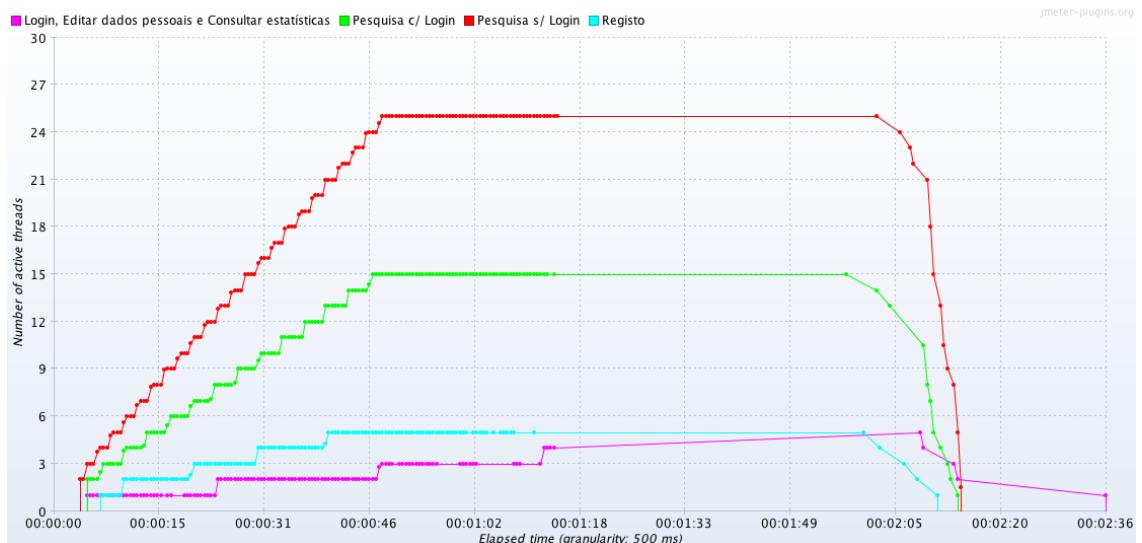


Figura 43: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo 2 com 50 utilizadores virtuais.

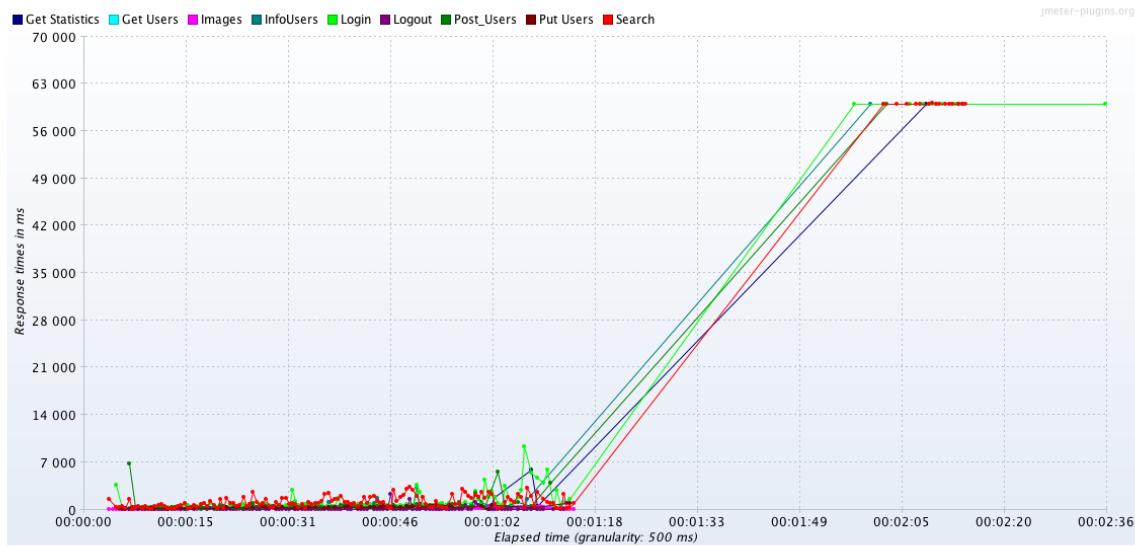


Figura 44: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 2 com 50 utilizadores virtuais.

Tabela 8: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 2 com 50 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	1125	156	12,18%
pesquisa s/ login	2553	292	10,26%
registro	404	98	19,52%
editar info pessoais	283	46	13,98%
Total	4365	592	11,94%

Assim, pode-se concluir que 50 utilizadores a efetuar pedidos ao servidor aplicacional constantemente (teste tipo 2), corresponde a uma carga demasiado elevada para o mesmo ser capaz de dar resposta. Prosseguiu-se então com uma redução de uma quantidade de utilizadores virtuais.

Teste do tipo 2 com 30 utilizadores virtuais:

Neste teste obteve-se um débito de 45,40 operações por segundo e um tempo de resposta médio de 469 milissegundos. Apesar destes valores serem aceitáveis, o percentagem de insucesso encontra-se ainda um pouco elevado (22,03%).

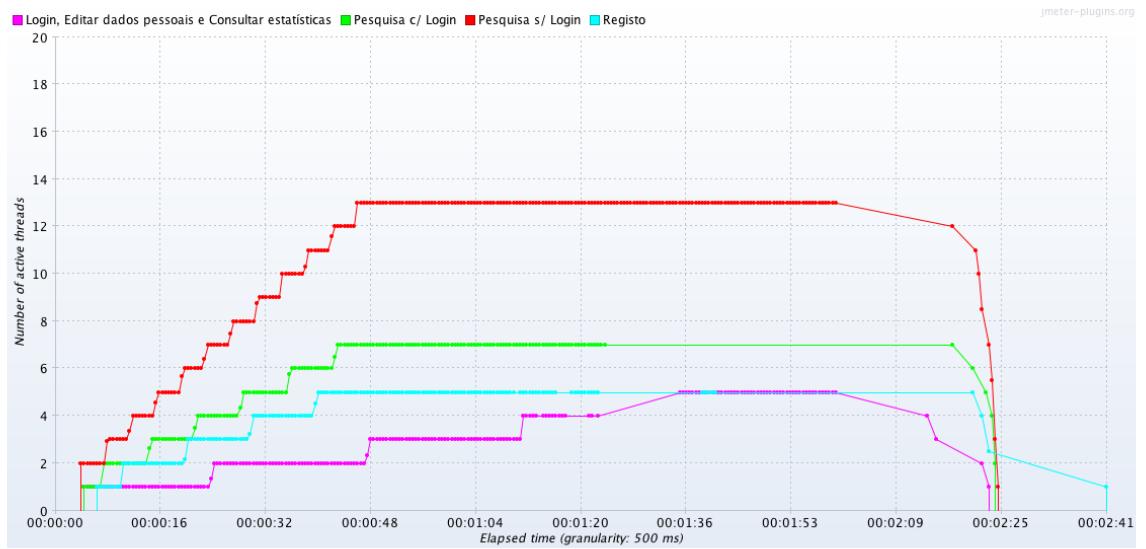


Figura 45: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo 2 com 30 utilizadores virtuais.

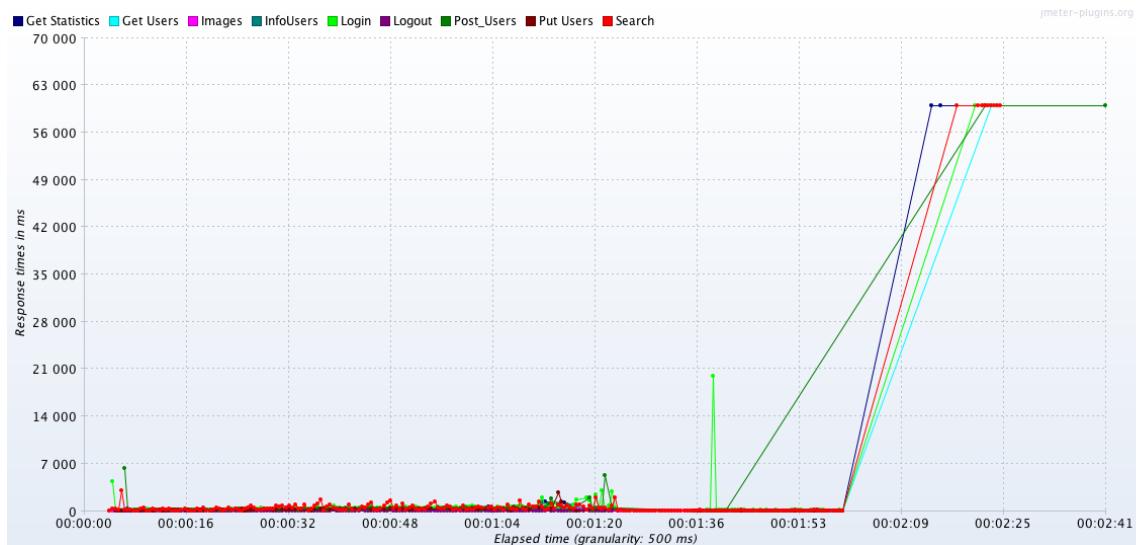


Figura 46: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 2 com 30 utilizadores virtuais.

Tabela 9: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 2 com 30 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	1239	139	10,09%
pesquisa s/ login	3517	375	9,64%
registro	893	99	9,98%
editar info pessoais	680	1175	63,34%
Total	6329	1788	22,03%

Tendo os resultados deste teste em conta, percebe-se que 30 utilizadores virtuais para o teste

do tipo 2, ainda é um valor demasiado elevado.

Teste do tipo 2 com 20 utilizadores virtuais:

Neste teste obteve-se um débito de 72 operações por segundo e um tempo de resposta médio de 233 milissegundos. Conforme esperado, estes valores ainda são aceitáveis e, para além disso, a taxa de insucesso também o passou a ser (7,35%).

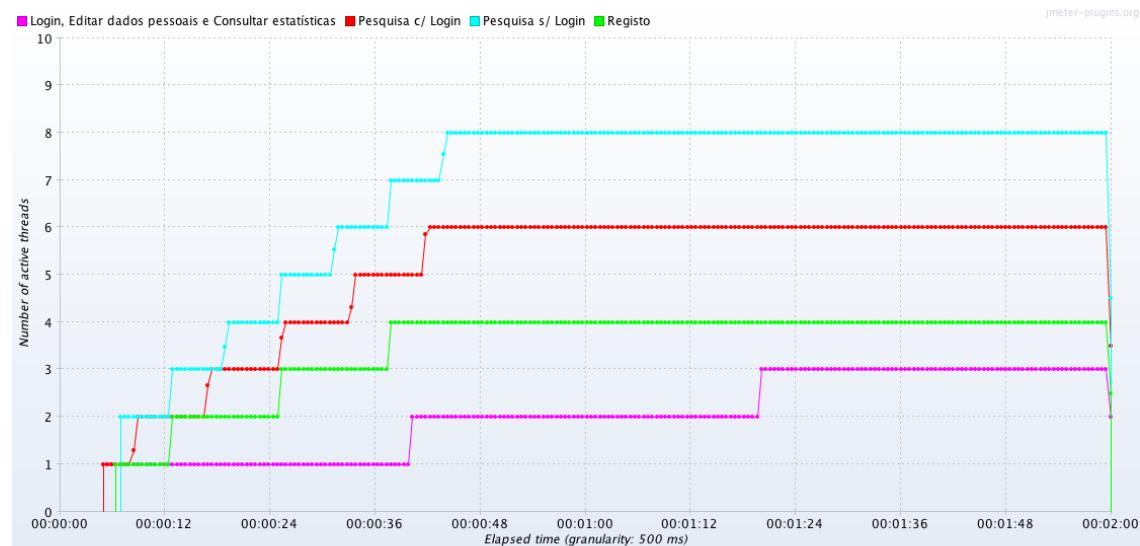


Figura 47: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo 2 com 20 utilizadores virtuais.

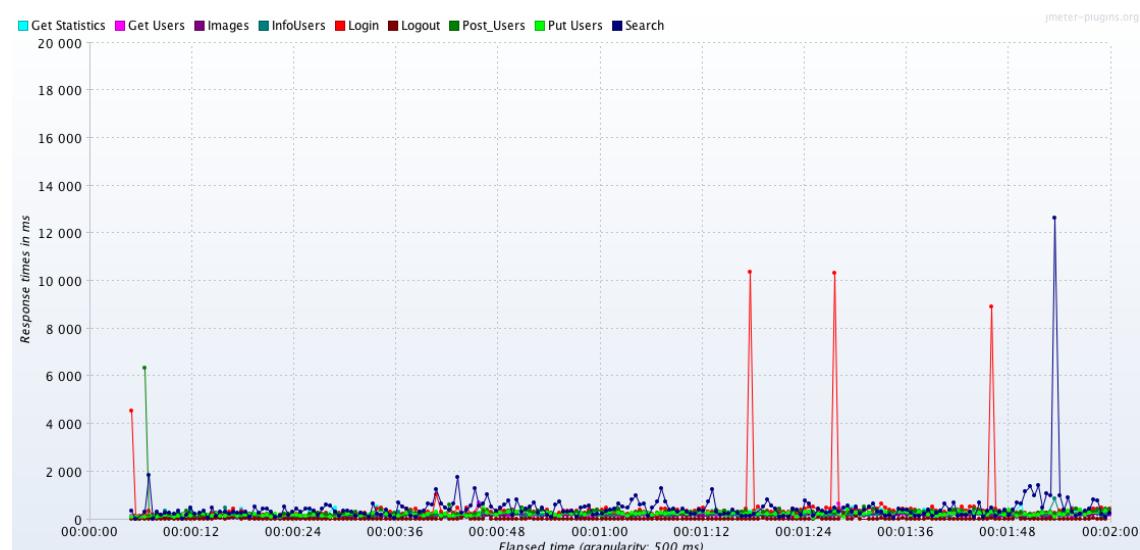


Figura 48: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 2 com 20 utilizadores virtuais.

Tabela 10: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo **2** com 20 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	2246	212	8,62%
pesquisa s/ login	3344	344	9,33%
registro	1281	44	3,32%
editar info pessoais	1153	37	3,11%
Total	8024	637	7,35%

8.2.4.2 Arquitetura com dois Servidores Aplicacionais

Teste do tipo 1 com 100 utilizadores virtuais:

Neste teste obteve-se um débito de 10,43 operações por segundo e um tempo de resposta médio de 468 milissegundos. Para além disso, obteve-se uma quantidade de *threads* ativas aceitáveis (geralmente igual a 1 *thread*, após estabilizar), como se pode verificar através da figura 49. Adicionalmente, verificou-se que a taxa de insucesso na receção de resposta encontra-se na ordem dos 1,38%.

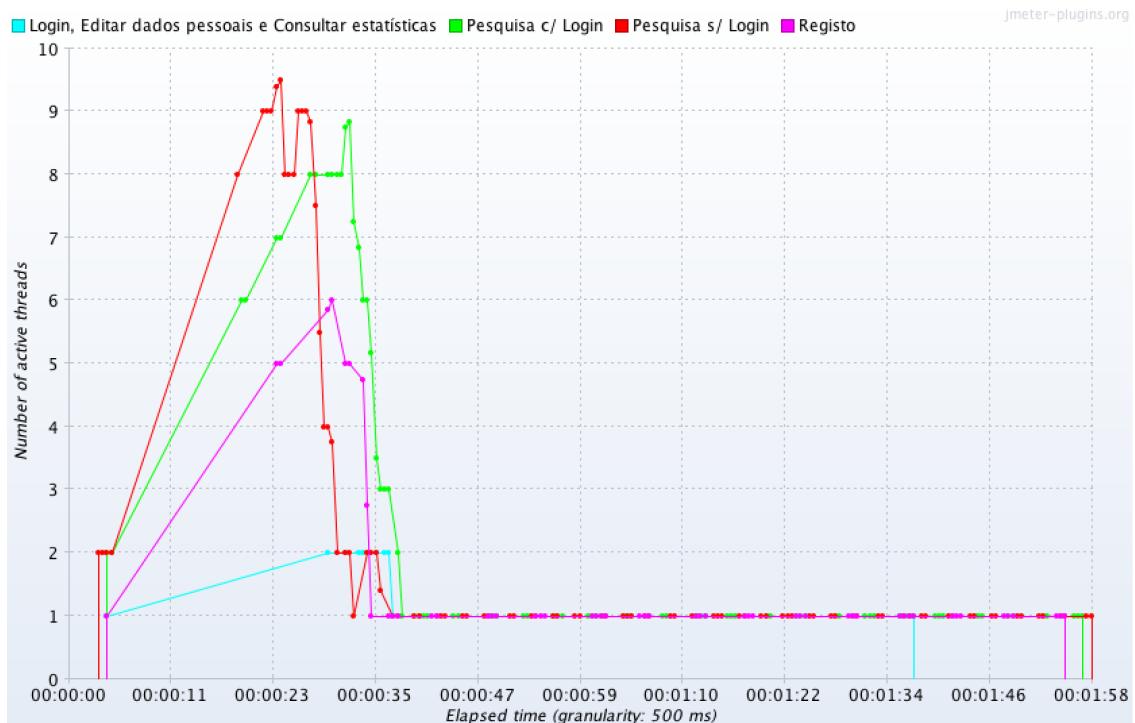


Figura 49: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo **1** com 100 utilizadores virtuais.

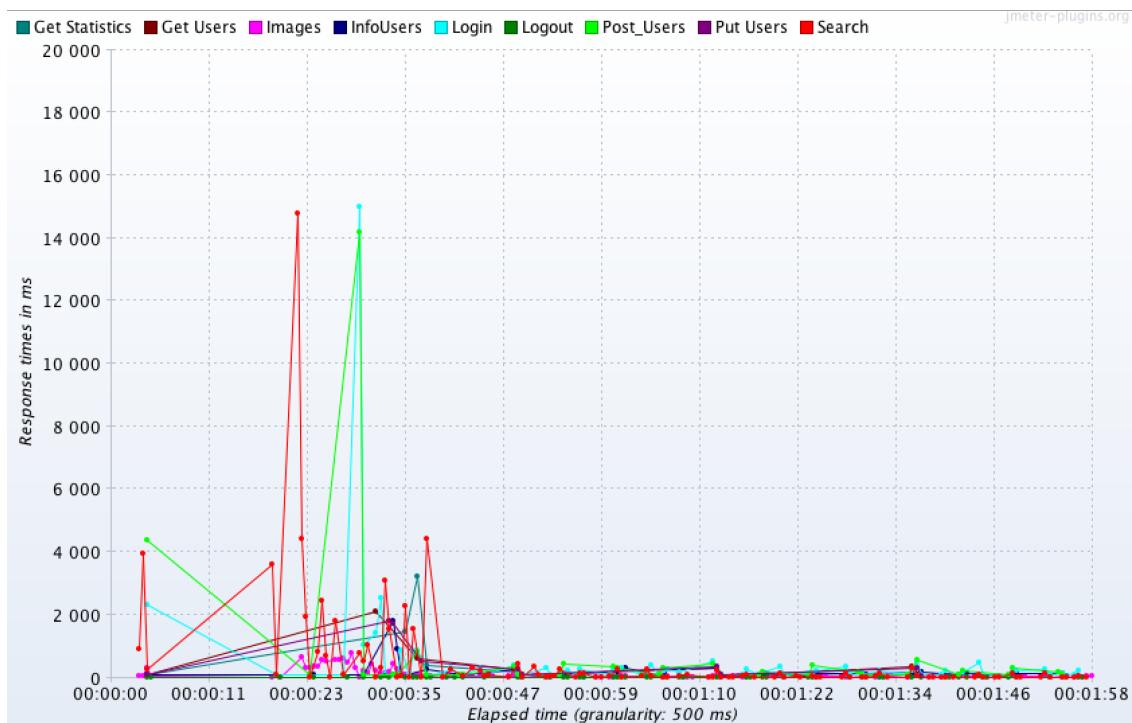


Figura 50: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 1 com 100 utilizadores virtuais.

Tabela 11: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 1 com 100 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	471	9	1,88%
pesquisa s/ login	532	8	1,48%
registro	160	0	0,00%
editar info pessoais	50	0	0,00%
Total	1213	17	1,38%

Com estes valores percebe-se que o servidor é capaz de dar resposta a mais pedidos (débito e tempo de resposta reduzidos, bem como reduzida percentagem de insucesso), pelo que se optou por aumentar a quantidade de utilizadores virtuais para este tipo de teste.

Teste do tipo 1 com 400 utilizadores virtuais:

Neste teste obteve-se um débito de 21,04 operações por segundo e um tempo de resposta médio de 217 milissegundos. Salienta-se que, conforme se pode perceber através do gráfico 52, os pedidos iniciais ao servidor tiveram um tempo de resposta superior, relativamente ao teste anterior. Contudo, o servidor aplicacional foi capaz de estabilizar o tempo de resposta ao pedidos efetuados no mesmo período de tempo.

Relativamente à taxa de insucesso, este encontra-se na ordem dos 1,32%.

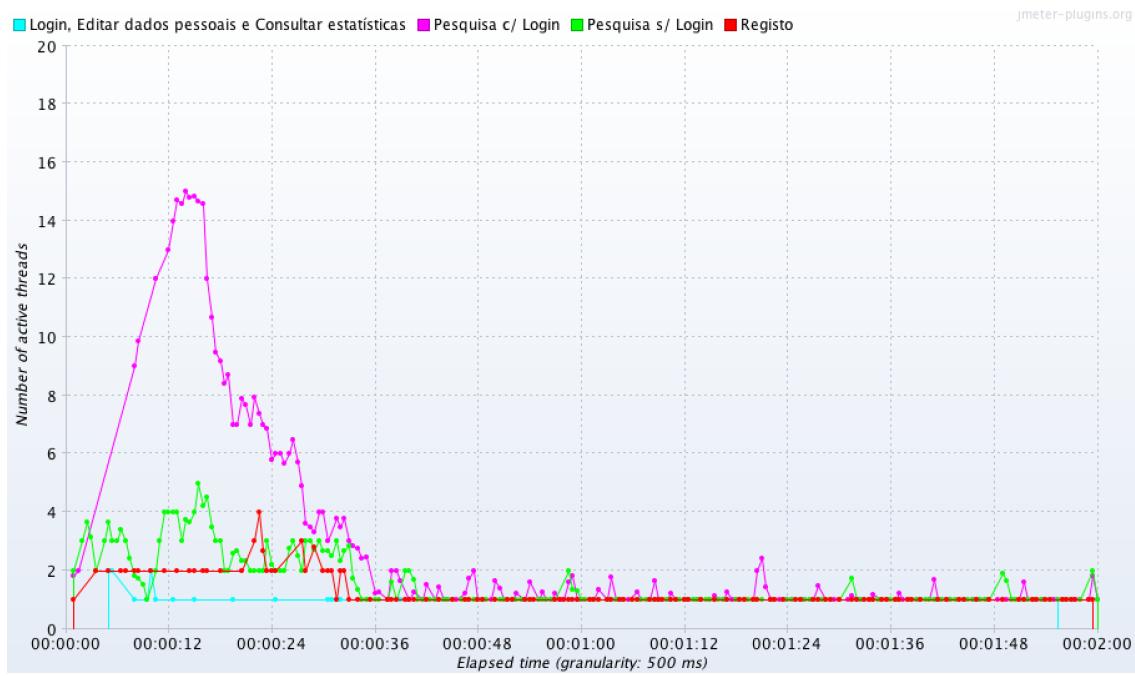


Figura 51: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo 1 com 400 utilizadores virtuais..

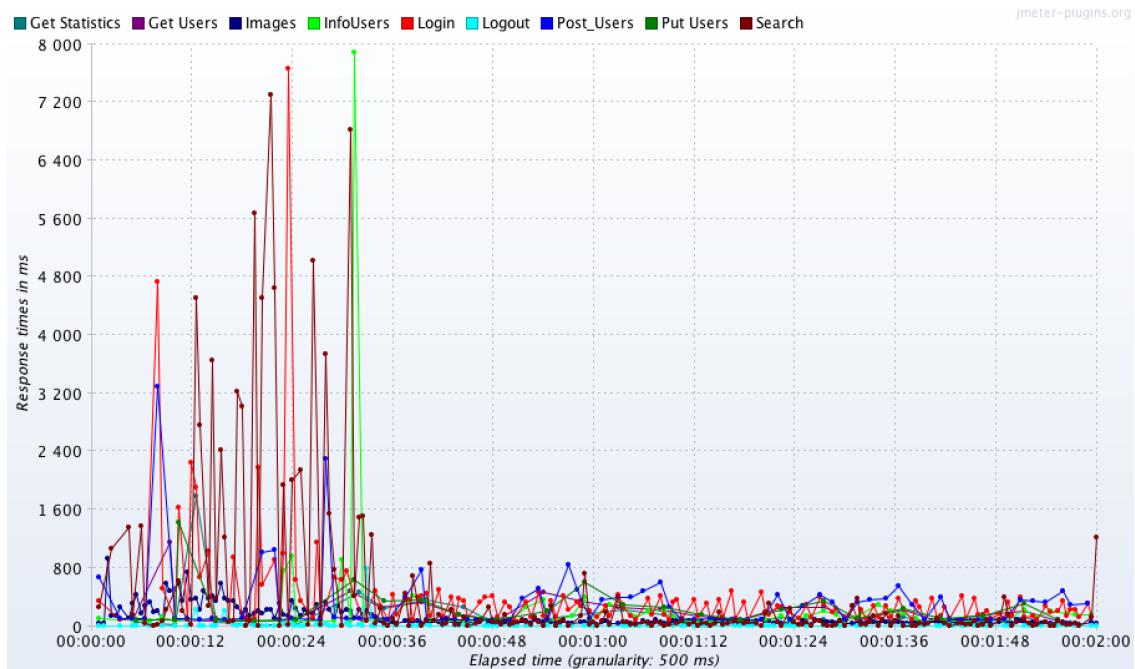


Figura 52: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 1 com 400 utilizadores virtuais.

Tabela 12: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 1 com 400 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	1181	19	1,58%
pesquisa s/ login	889	11	1,22%
registo	297	3	1,00%
editar info pessoais	248	2	0,80%
Total	2615	35	1,32%

Estes são resultados positivos, a partir dos quais se pode concluir que o servidor aplicacional é capaz de lidar com 400 utilizadores virtuais, num teste do tipo 1.

Teste do tipo 2 com 100 utilizadores virtuais:

Agora, com o teste do tipo 2, obteve-se um débito de 35,36 operações por segundo e um tempo de resposta médio de 2,2 segundos.

Adicionalmente, verificou-se que a taxa de insucesso na receção de resposta se encontra na ordem dos 5,60%.

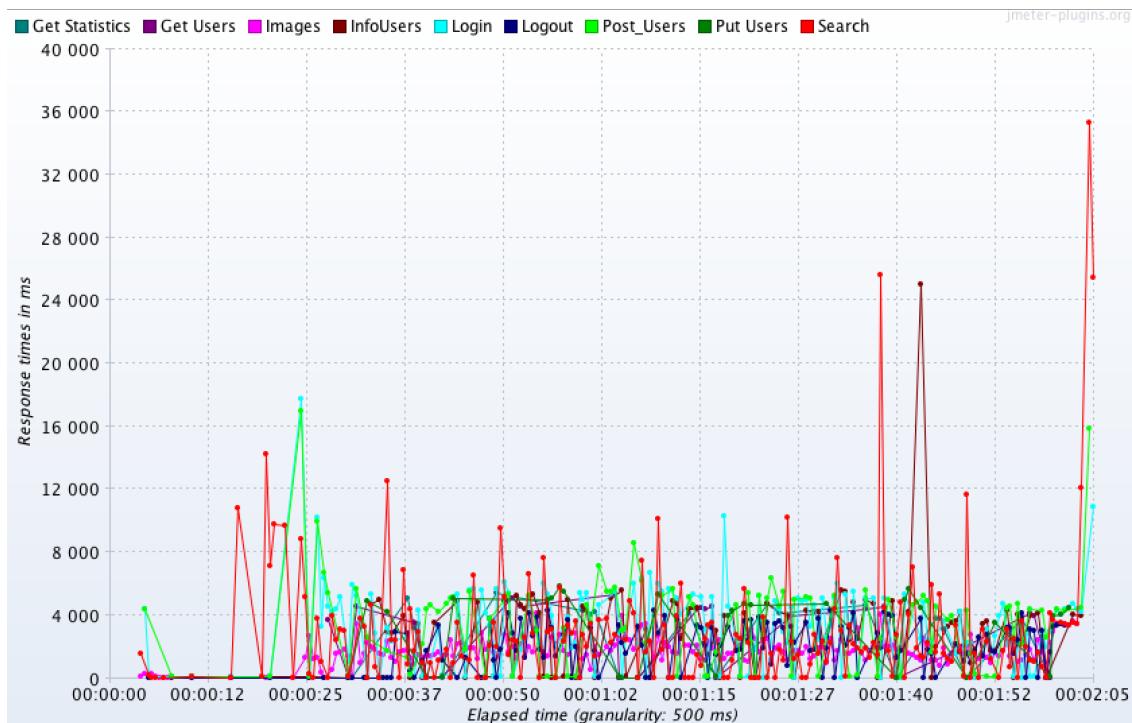


Figura 53: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo 2 com 100 utilizadores virtuais.

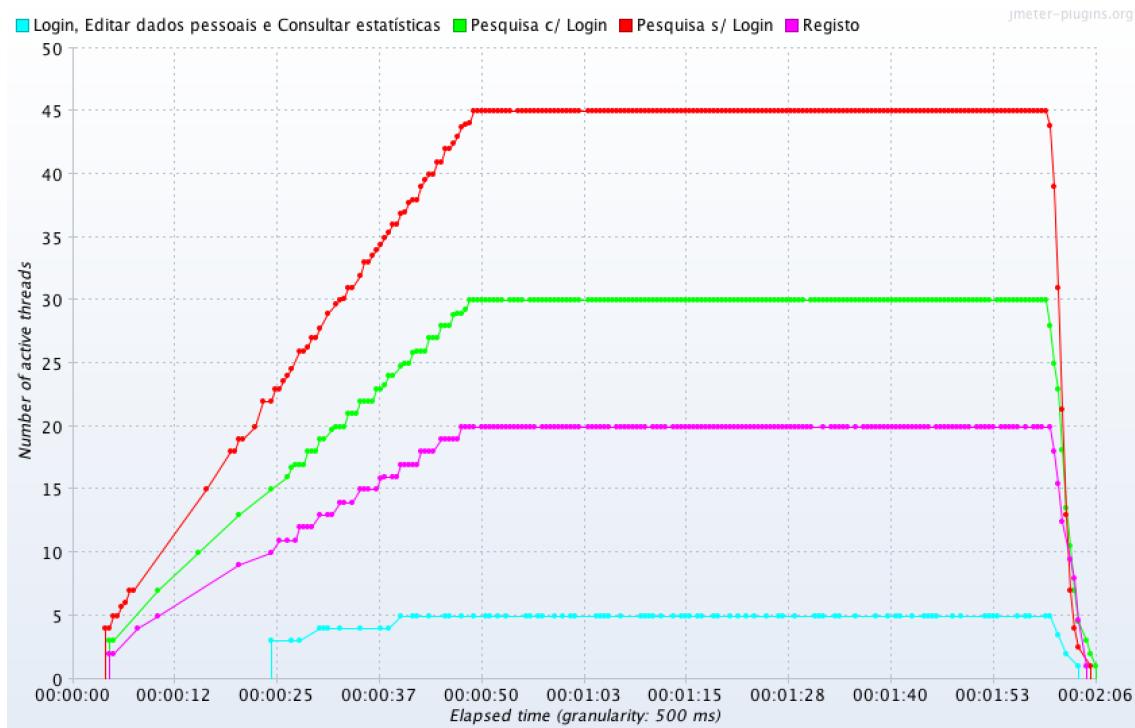


Figura 54: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 2 com 100 utilizadores virtuais.

Tabela 13: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 2 com 100 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	1064	84	7,32%
pesquisa s/ login	2412	75	3,02%
registro	578	72	11,08%
editar info pessoais	141	18	11,32%
Total	4195	249	5,60%

Os tempos de resposta obtidos com este teste podem ser considerados ligeiramente negativos, uma vez que é frequente a ocorrência de tempos de resposta entre 2 e 8 segundos, tal como se pode verificar no gráfico apresentado na figura 54. Portanto, optou-se por reduzir a quantidade de utilizadores virtuais.

Teste do tipo 2 com 50 utilizadores virtuais:

Com este teste obteve-se um débito de 36,81 operações por segundo e um tempo de resposta médio de 1,08 segundos.

Adicionalmente, verificou-se que a taxa de insucesso na receção de resposta se encontra na ordem dos 3,17%.

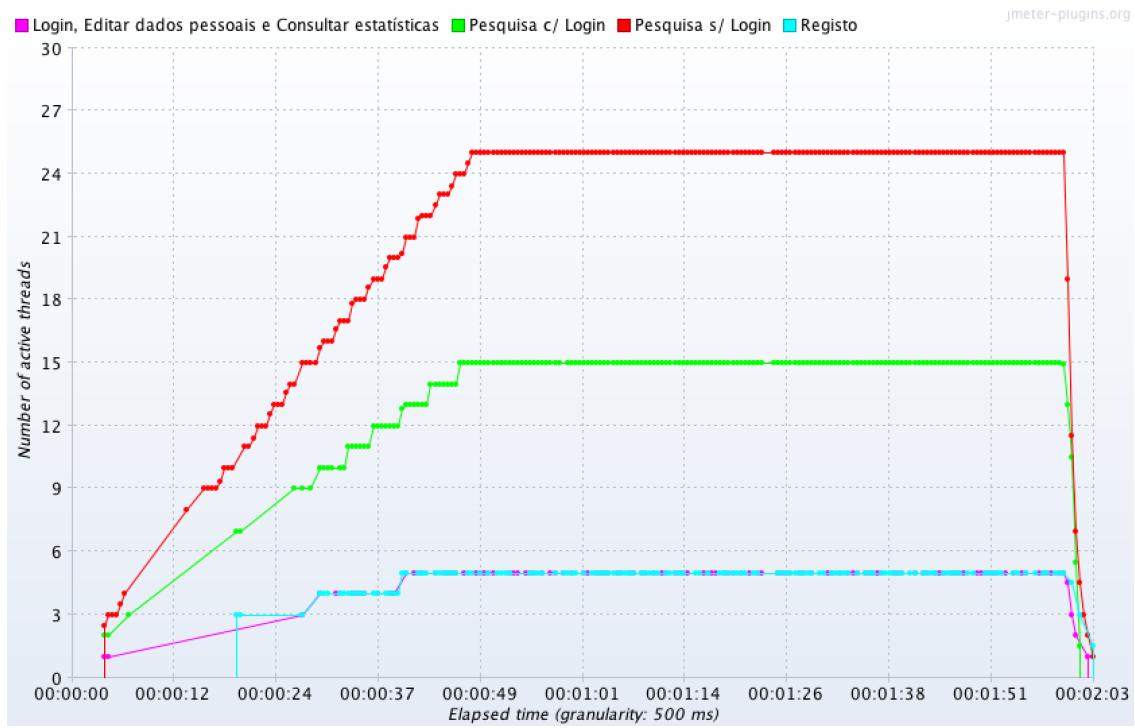


Figura 55: Quantidade de *threads* ativas ao longo do tempo, para o teste do tipo 2 com 100 utilizadores virtuais.

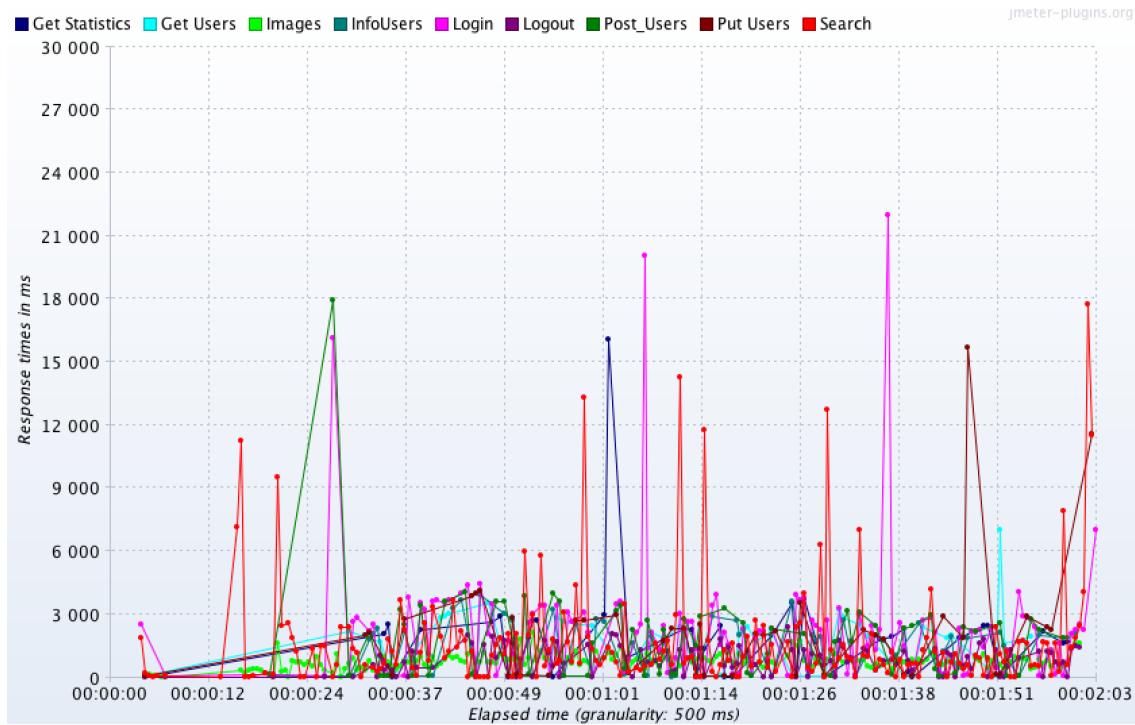


Figura 56: Tempo de resposta aos pedidos efetuados ao longo do tempo, para o teste do tipo 2 com 50 utilizadores virtuais.

Tabela 14: Quantidade de sucessos e insucessos, bem com percentagem de insucesso, para o teste do tipo 2 com 50 utilizadores virtuais.

Thread-Group	Sucesso	Insucesso	% Insucesso
pesquisa c/ login	1028	44	4,10%
pesquisa s/ login	2804	89	3,08%
registro	331	5	1,49%
editar info pessoais	327	9	2,68%
Total	4490	147	3,17%

Com este teste obtiveram-se valores de tempo de resposta um pouco melhores (com a maioria entre 0 e 4 segundos). Assim, considera-se que 50 utilizadores virtuais é o limite de carga para os servidores aplicacionais em testes do tipo 2.

8.2.4.3 Conclusões

A partir dos resultados previamente apresentados, pode-se concluir que o servidor aplicacional tem como limite de carga 700 e 20 utilizadores virtuais para os testes de tipo 1 e 2, respetivamente. Para arquitetura com dois servidores aplicacionais, este limite de carga, manteve-se nos 400 e 50 utilizadores virtuais para os testes de tipo 1 e 2, respetivamente.

Salienta-se que para os testes com a arquitetura de dois servidores aplicacionais, um destes servidores estava a correr o docker da aplicação numa máquina virtual, com características muito inferiores às desejáveis. Desta forma, os resultados obtidos não são tão positivos conforme o esperado, em resultado da inclusão de uma nova máquina na arquitetura.

Por fim, é ainda de se referir que a quantidade de dados transferidos nas respostas aos pedidos HTTP é relativamente elevado, devido à transferência de imagens, conforme se pode verificar através dos dados apresentados na figura 15.

Tabela 15: Quantidade de dados transferidos nas respostas aos pedidos HTTP.

Nº de Servidores Aplicacionais	Tipo de Teste	Number of Threads	Quantidade de dados transferidos (MB)
1	1	700	820
	2	20	780
2	1	400	510
	2	50	260

9 Testes de usabilidade da interface

Por forma a avaliar a qualidade da interface desenvolvida para a plataforma, foram realizados testes de usabilidade com base na execução de determinadas tarefas na plataforma.

De seguida, apresentam-se então em baixo os resultados dos testes realizados e as taxas de completude das tarefas a realizar respetivamente.

Tabela 16: Resultados do questionário SUS recolhidos e respetivos *scores*.

1	2	3	4	5	6	7	8	9	10	TOTAL
5	1	4	1	5	4	5	1	5	1	90
4	1	4	1	4	1	5	1	4	1	90
4	1	4	1	4	3	4	2	3	1	77,5
5	2	5	1	5	1	5	2	4	1	92,5

Tabela 17: Resultados da completude das tarefas realizadas durante os testes e respetivos *scores*.

1	2	3	4	TOTAL
1	1	1	1	100%
1	1	1	1	100%
1	1	0,5	1	88%
1	1	1	1	100%

Através da observação dos resultados acima, conclui-se que a interface com o utilizador tem ainda alguns pontos a rever. Ainda assim, consideram-se positivos os resultados obtidos.

10 Conclusões

Com este trabalho foi possível a percepção da importância da fase de conceção, no desenvolvimento do *software*. De facto, a análise dos requisitos do sistema tendo em conta os utilizadores e o estabelecimento de *personas* permitiu-nos compreender melhor o que as entidades que vão utilizar a aplicação pretendem da mesma e, desta forma, desenvolver um sistema mais focado nos mesmos. A realização dos testes de usabilidade, por sua vez, permitiram perceber que se estes tivessem sido efetuados atempadamente, teriam permitido a melhoria dos *mockups* desenvolvidos e, consequentemente, da interface final.

Relativamente à fase de desenvolvimento da aplicação, verificou-se, mais uma vez, a utilidade de diagramas como os de classe. Neste projeto, o recurso a diagramas dependentes de tecnologias (PSM), permitiu uma inicialização do desenvolvimento bastante mais rápida. Para além disso, estes diagramas facilitaram a divisão de tarefas pelos elementos do grupo, uma vez que permitiram detetar entidades em grande medida independentes.

Na fase de testes de carga, encontraram-se diversas dificuldades devido à dificuldade de aprendizagem da ferramenta utilizada e problemas relacionados com concorrência. Adicionalmente, a inconstância dos resultados obtidos para configurações e testes equivalentes, dificultou o processo de avaliação de desempenho da aplicação desenvolvida. Salienta-se ainda a ausência de tempo de espera entre operações, que, apesar de ser irrealista, permite a realização de testes mais intensivos, que levam em consideração o pior caso.

Em relação aos testes de usabilidade na interface, verificou-se mais uma vez que a realização dos mesmos atempadamente, teria proporcionado a possibilidade de adaptar a interface, tornando-a mais intuitiva para o utilizador.

Em suma, este projeto providenciou ao grupo de trabalho os conhecimentos práticos necessários para o desenvolvimento de uma aplicação *web*, focada no utilizador e na performance da mesma. Ainda que os resultados obtidos em termos de desempenho pudessem ser melhorados, o grupo adquiriu os conhecimentos necessários para avaliar a performance de um sistema *web* em momentos de carga e efetuar as melhorias necessárias ao mesmo.