

# TQS: Relatório de Especificação do Produto

*João Vasconcelos [88808], Tiago Mendes [88886], Vasco Ramos [88931]*

v2020-05-05

<b>1. Introdução</b>	<b>2</b>
1.1. Apresentação do Projeto	2
1.1.1. Oportunidade de Negócio	2
1.1.2. Ideia do Produto	3
1.2. Limitações	3
<b>2. Conceito do Produto</b>	<b>4</b>
2.1. Visão Geral dos CaU	4
2.2. Personas	4
2.3. Principais Cenários de Utilização	5
2.4. Epics e Prioridades	6
2.5. Metodologia de Levantamento de Requisitos	6
<b>3. Modelo de Domínio</b>	<b>8</b>
<b>4. Arquitetura do Sistema</b>	<b>9</b>
4.1. Principais Requisitos e Limitações	9
4.2. Vista da Arquitetura	9
4.3. Arquitetura de instalação	10
<b>5. API para Developers</b>	<b>10</b>
<b>6. Referências e Recursos</b>	<b>11</b>

# 1. Introdução

## 1.1. Apresentação do Projeto

Este projeto tem como objetivo aplicar metodologias de integração e testes automáticos com a implementação de uma bateria completa de testes. Para além disso, aplicar também metodologias ágeis para facilitar o desenvolvimento do projeto em equipa.

No geral, o principal foco é implementar um produto de e-commerce (marketplace online), seguindo uma estratégia de Software Quality Assurance (SQA).

### 1.1.1. Oportunidade de Negócio

Uma empresa com atividade no segmento imobiliário pretende alargar a sua área de atuação e competir contra empresas com forte presença no mercado nacional de arrendamento. Assim, esta empresa encontrou uma área pouco explorada no mercado de arrendamento imobiliário e, seguindo a tendência atual de *e-commerce*, pretende investir numa plataforma online de arrendamento, que tem como público alvo alunos universitários que procuram um espaço para alugar a curto/médio prazo.

Deste modo, a nossa equipa, em conjunto com a empresa interessada em investir, prossegue com a criação de uma plataforma que agiliza e simplifica o processo de arrendamento a universitários.

O público-alvo desta plataforma caracteriza-se por ser bastante heterogéneo e diversificado, pelo que é emergente a necessidade de criar um produto inovador e que traga valor para os utilizadores. Nesta perspetiva, a equipa considera que um aspeto fulcral do produto relaciona-se com reduzir ao máximo a burocracia envolvida no aluguer de imóveis, ou seja, expor os utilizadores o menos possível a toda a burocracia necessária, tentando automatizar o processo. É aqui que surge duas características diferenciadoras: comparador de imóveis e gerar automática um contrato de aluguer entre locador e locatário, diminuindo o tempo dispendido no processo, bem como eventuais despesas associadas.

Assim, aliando a estas funcionalidades a possibilidade de avaliar os imóveis e pesquisa inteligente, acreditamos ter um produto capaz de se destacar no mercado e preencher lacunas atualmente existentes no que diz respeito ao aluguer de imóveis a estudantes universitários.

### 1.1.2. Ideia do Produto

Para o/a:	Estudante ou Locador.
Que apresenta:	Dificuldade em encontrar imóveis para arrendamento (estudante universitário/locatário) e, ainda, dificuldade em publicitar os seus imóveis (locador, que acaba por utilizar redes sociais e marketplaces generalistas).
O produto:	DOMUS, que é uma plataforma de arrendamento de imóveis.
Que:	Agiliza e simplifica o aluguer de espaços de habitação, especificamente para estudantes.
Ao contrário de:	Imovirtual e Airbnb.
O nosso produto:	Facilita a procura e escolha de um espaço para habitar que vai de encontro às necessidades do público-alvo (estudantes universitários). Otimiza o processo de decisão, pois, inclui avaliações do imóvel submetidas através do feedback de arrendatários anteriores. Permite a comparação de até três imóveis. Permite gerar automaticamente um contrato de arrendamento.
Competição:	Uniplaces

*Tabela 1 - Breve descrição do produto*

## 1.2. Limitações

<explain the known limitations/unimplemented (but planned) features>

## 2. Conceito do Produto

### 2.1. Visão Geral dos CaU

A nossa aplicação, **domus**, apresenta uma solução para o problema de procura de quartos por parte de estudantes universitários para arrendar. Esta procura normalmente é demorada e muitas vezes os estudantes têm que se deslocar várias vezes à cidade universitária até conseguirem arrendar um quarto. É este o problema que o nosso produto pretende resolver oferecendo uma solução que à distância de um clique permite ao utilizador verificar as características do quarto e se tiver interesse arrendar o mesmo.

Com este objetivo em mente, existem vários casos de uso essenciais no nosso produto como a pesquisa de quartos e avaliação dos mesmos por parte do locatário e a adição de publicações de quartos por parte do locador. Estes casos de uso e os restantes presentes na nossa solução estão representados através de um diagrama de casos de uso na figura XPTO.

<functional (black-box) description of the application: what will your system be used for? Which is the high-level/business problem being solved by your system?>

<if needed, clarify what was planned/expected to be included but was changed to a different approach/concept >

<optional: how is your system different or similar to other well-known products?>

<optional: you may include a UML Use Case diagram to support the explanation>

### 2.2. Personas

O desenvolvimento da aplicação domus foi realizado tendo em conta 2 Personas principais: o Carlos Santos, um locador e o Rafael Gomes, um locatário.

- **Carlos Santos**

O Carlos Santos, tem 40 anos e é médico no centro de saúde de Aveiro à 10 anos. É uma pessoa energética que gosta de jogar ténis, praticar atletismo e ouvir música nos seus tempos livres. Há cerca de 5 anos, o Carlos decidiu investir no mercado imobiliário em Aveiro e comprou 2 casas na vila jovem com o objetivo de arrendar a estudantes universitários.

Durante o verão, antes do início do novo ano letivo, muitos dos locatários do Carlos saem das suas casas e este fica com a tarefa de encontrar novos inquilinos que queiram arrendar um quarto. O Carlos acaba por despende muito tempo a mostrar casas a clientes interessados e perde muitos clientes porque não se pode ausentar do trabalho para visitar as moradias com eles.

**Motivação:** O Carlos gostava de possuir uma forma mais fácil de apresentar os seus imóveis a clientes interessados.

- **Rafael Gomes**

O Rafael tem 18 anos e vive em Coimbra, onde concluiu o ensino secundário. É uma pessoa motivada, praticando basquetebol e natação nos seus tempos livres. Além disso o Rafael gosta de jogar jogos de computador com os seus amigos e pretende estudar Engenharia Informática em Aveiro.

Com o seu primeiro ano de faculdade a chegar, os pais do Rafael pretendem alugar um quarto para o filho em Aveiro, onde ele possa viver durante o período escolar. Como os seus pais trabalham durante a semana, apenas podem ajudá-lo na procura de quartos ao fim de semana, por isso durante a semana o Rafael desloca-se várias vezes a Aveiro sozinho para visitar novos quartos. Ele leva sempre um bloco de notas onde aponta as características dos quartos e das casas para poder mais tarde comparar os vários quartos que visitou.

Este processo acaba por ser muito dispendioso e cansativo, ocupando um grande intervalo de tempo dos dias do Rafael e dos seus pais.

**Motivação:** O Rafael gostava de possuir uma forma rápida e acessível para procurar quartos e poder compará-los.

## 2.3. Principais Cenários de Utilização

Estes cenários de utilização principais descrevem como o nosso produto se integra com as atividades dos utilizadores finais e como eles as executam de forma a alcançar os seus objetivos.

- **O Carlos pública um novo quarto:**

Um dos locatários do Carlos acabou de sair de uma das suas casas por isso o locador tem um quarto livre para arrendar. Por esse motivo, o Carlos abre a web app e depara-se com uma página onde são apresentadas as funcionalidades principais do **domus**. Após efetuar login, ele seleciona a opção **new room** e depara-se com um formulário onde são pedidas diversas informações sobre o quarto que o locador quer publicar.

Após preencher todas as informações pedidas, o Carlos submete o formulário e recebe uma mensagem de confirmação, informando-o que o quarto foi publicado com sucesso.

- **O Rafael procura por um quarto na cidade de Aveiro:**

O Rafael precisa de encontrar um quarto na cidade de Aveiro. Com esse objetivo em mente, após efetuar login no **domus**, o Rafael observa a listagem de quartos disponíveis na aplicação e insere na barra de pesquisa a query “Aveiro” para filtrar os resultados pela cidade de Aveiro. Após clicar no botão de pesquisa, a página é atualizada e são apresentados os resultados de quartos disponíveis na cidade de Aveiro.

- **O Rafael adiciona um quarto à sua wishlist**

O Rafael pretende adicionar todos os quartos que gostou à sua wishlist para poder mostrar aos seus pais e decidir qual seria o melhor para ele. Por isso, após efetuar o login, o Rafael observa a listagem de quartos disponíveis e clica no ícone com o formato de um coração para adicionar os quartos em que tem interesse à sua wishlist.

## 2.4. Epics e Prioridades

Tendo em conta a especificação do produto e as funcionalidades pensadas, iremos focar-nos inicialmente em conseguir disponibilizar os imóveis, pelo que temos primeiro de concluir as funcionalidades de procura e publicação de casas. Após concluir estes dois epics, a equipa focar-se-á no desenvolvimento da ferramenta de comparação e na possibilidade de feedback. Deixando para o fim o desenvolvimento da wishlist e a geração de contratos automática.

Sprint	Epics
1	Procura de uma casa, Publicação de uma casa
2	Publicação de uma casa, Comparação de Imóveis
3	Comparação de Imóveis, Colocação de uma review
4	Adicionar casa à wishlist
5	Formalização de um contrato

*Tabela X - Implementação incremental por epics*

## 2.5. Metodologia de Levantamento de Requisitos

Com o objetivo de tornar o Domus uma plataforma mais credível, funcional e segura, uma das etapas de maior importância foi o levantamento de requisitos. No decorrer desta etapa, recorremos a questionários, à documentação fornecida pelo professor e a diversas fontes online. Os questionários, mais especificamente, foram entregues a diversos estudantes a nível nacional e a locadores de outras plataformas onde se conseguiram extrair diversos requisitos necessários para o Domus.

Com este projeto, pretendemos criar uma plataforma online atrativa e disponível a todos os sistemas operativos, para assim podermos atrair cada vez mais clientes. Posteriormente, pretendemos de forma algo ambiciosa, criar uma app mobile para tornar o Domus mais acessível e prático, potenciando ainda mais a proposta que nos foi apresentada.

Nome	Responsabilidades	Valor para a Plataforma
------	-------------------	-------------------------

Estudantes Universitários	Utilizadores da plataforma que observam as ofertas presentes nesta.	São os estudantes os principais responsáveis para que a plataforma se mantenha em bom funcionamento. Fazem parte do core do negócio desenvolvendo o papel de principais clientes no serviço.
Locador	Utilizadores da plataforma que publicam ofertas de imóveis direcionadas aos clientes no serviço, mais especificamente, os estudantes universitários.	Tal como os estudantes, os locadores fazem parte do core do negócio desenvolvendo o papel de principais vendedores no serviço.
<i>Real Estate Holdings</i>	Empresa que inicialmente encomendou a plataforma que se está a desenvolver.	Principal investidor.

*Tabela X - Perfis dos Stakeholders*

### 3. Modelo de Domínio

<which information concepts will be managed in this domain? How are they related?>

<use a logical model (UML classes) to explain the concepts of the domain and their attributes>



## 4. Arquitetura do Sistema

### 4.1. Principais Requisitos e Limitações

<Identify issues that will drive the choices for the architecture such as: Will the system be driven by complex deployment concerns, adapting to legacy systems, or performance issues? Does it need to be robust for long-term maintenance?

Identify critical issues that must be addressed by the architecture, such as: Are there hardware dependencies that should be isolated from the rest of the system? Does the system need to function efficiently under unusual conditions? Are there integrations with external systems? Is the system to be offered in different user-interfacing platforms (web, mobile devices, big screens,...)?

E.g.: (the references cited in [XX ] would be hypothetical links to previous specification documents/deliverables )

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- è The existing legacy Course Catalog System at Wylie College must be accessed to retrieve all course information for the current semester. The C-Registration System must support the data formats and DBMS of the legacy Course Catalog System [E2].
- è The existing legacy Billing System at Wylie College must be interfaced with to support billing of students. This interface is defined in the Course Billing Interface Specification [E1].
- è All student, professor, and Registrar functionality must be available from both local campus PCs and remote PCs with internet dial up connections.
- è The C-Registration System must ensure complete protection of data from unauthorized access. All remote accesses are subject to user identification and password control.
- è The C-Registration System will be implemented as a client-server system. The client portion resides on PCs and the server portion must operate on the Wylie College UNIX Server. [E2]
- è All performance and loading requirements, as stipulated in the Vision Document [E2] and the Supplementary Specification [15], must be taken into consideration as the architecture is being developed.>

### 4.2. Vista da Arquitetura

→ Discuss architecture planned for the software solution.

→ include a diagram

<detail the specific technologies/frameworks that were used>

→ explain how the identified modules will interact. Use sequence diagrams to clarify the interactions along time, when needed

→ discuss more advanced app design issues: integration with Internet-based external services, data synchronization strategy, distributed workflows, push notifications mechanism, distribution of updates to distributed devices, etc.>

### 4.3. Arquitetura de instalação

[Explicar a organização prevista da solução em termos configuração de produção (*deployment*). Modelar num diagrama de *deployment*]

## 5. API para Developers

[Explicar a organização da API. Os detalhes detalhes/documentação dos métodos devem ficar numa solução *hosted* de documentação de APIs, como o [Swagger](#), ou <https://apiary.io/> ]

<what services/resources can a developer obtain from your REST-API?>

<document the support endpoints>

[ Base URL: localhost:8080/weather ]	
client	Regular user of the weather forecast API
GET	/now/{latitude},{longitude} get weather forecast of the current day for the given coordinates
GET	/recent/{latitude},{longitude}/{days} get weather forecast of the next days starting from today until the given number of days for the given coordinates
GET	/period/{latitude},{longitude}/{start},{end} get weather forecast of the given time period for the given coordinates
GET	/cached get weather forecasts previously requested and still present in cache

## 6. Referências e Recursos

<document the key components (e.g.: libraries, web services) or key references (e.g.: blog post) used that were really helpful and certainly would help other students pursuing a similar work>