45426 Teste e Qualidade de Software

TQS: Relatório de Especificação do Produto

##### Pedro Figueiredo [97487], Fábio Martins [98119], Renato Dias [98380], José Trigo [98597]

v2022-06-15

1. [Introdução 2](#_TOC_250017)
   1. [Apresentação do Projeto 2](#_TOC_250016)
      1. [Oportunidade de Negócio 2](#_TOC_250015)
      2. [Ideia do Produto 3](#_TOC_250014)
   2. [Limitações 3](#_TOC_250013)
2. [Conceito do Produto 4](#_TOC_250012)
   1. [Visão Geral dos CaU 4](#_TOC_250011)
   2. [Personas 4](#_TOC_250010)
   3. [Principais Cenários de Utilização 6](#_TOC_250009)
3. [Arquitetura do Sistema 10](#_TOC_250005)
   1. [Vista da Arquitetura 11](#_TOC_250003)
4. [API para Developers 13](#_TOC_250001)

# Introdução

## Apresentação do Projeto

Este projeto tem como principal objetivo aplicar metodologias de integração e testes automáticos com a implementação de uma bateria completa de testes. Para além disso, visa também aplicar 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).

### Oportunidade de Negócio

Uma empresa com atividade no setor alimentar pretende alargar a sua área de atuação e competir contra empresas com forte presença no mesmo mercado. Assim, esta empresa encontrou uma área pouco explorada no mercado alimentar e, seguindo a tendência atual de *e-commerce*, pretende investir numa plataforma online de encomenda e entrega de vinhos, que tem como público alvo qualquer tipo de pessoas que necessite de encomendar vinho.

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 encomenda e entrega de vinhos.

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 a diversidade de bebidas disponível para encomendar, devendo incluir uma grande diversidade de vinhos.

Assim, acreditamos ter um produto capaz de se destacar no mercado e preencher lacunas atualmente existentes no que diz respeito à encomenda e entrega de vinhos.

45426 Teste e Qualidade de Software

### Ideia do Produto

|  |  |
| --- | --- |
| Para o/a: | Qualquer tipo de pessoa. |
| Que apresenta: | Necessidade de comprar vinho. |
| O produto: | DrinkLivery, que é uma plataforma de encomenda e entrega de vinhos. |
| Que: | Permite a encomenda de vinhos. |
| Ao contrário de: | Uber Eats. |
| O nosso produto: | Permite a encomenda e respetiva entrega de bebidas, mais propriamente vinho. |
| Competição: | --------------------- |

*Tabela 1: Breve descrição do produto*

## Limitações

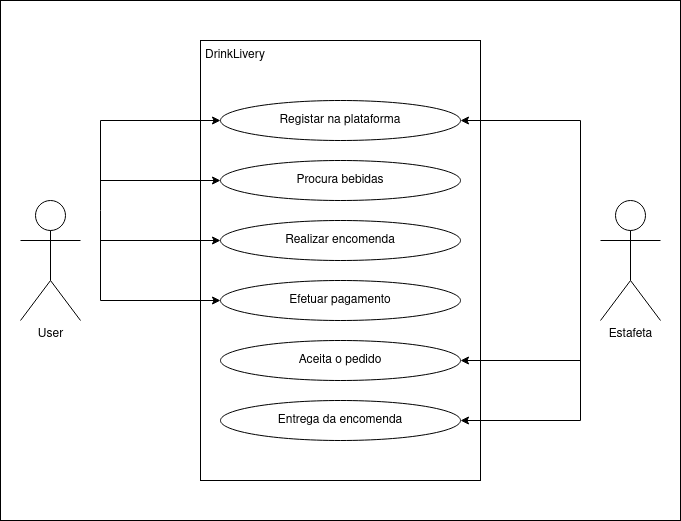
De um modo geral, o nosso sistema satisfaz quase a totalidade dos requisitos propostos, pelo que obtivemos um produto completo, eficiente e robusto.

# Conceito do Produto

## Visão Geral dos CaU

A nossa aplicação, DrinkLivery , apresenta uma solução para o problema de encomenda e entrega de vinhos. Não existe uma plataforma capaz de simplificar o processo de compra de vinhos, obrigando as pessoas a deslocar-se ao supermercado e/ou garrafeira para a sua compra. É este o problema que o nosso produto pretende resolver oferecendo uma solução que à distância de um clique permite ao utilizador encomendar o(s) vinho(s) que deseja de forma a que estas sejam entregues no local por si indicado.

Com este objetivo em mente, existem alguns casos de uso essenciais no nosso produto como a realização do pedido da parte do utilizador e a entrega da encomenda por parte do estafeta. Estes casos de uso e os restantes presentes na nossa solução estão representados através de um diagrama de casos de uso na figura 1.



*Figura 1: Diagrama de Casos de Uso*

## Personas

O desenvolvimento da aplicação DrinkLivery foi realizado tendo em conta 2 Personas principais: o Francisco Silva, um utilizador e o Miguel Silva, um estafeta.

#### Francisco Silva

45426 Teste e Qualidade de Software

O Francisco Silva, tem 45 anos e é médico no Hospital da Luz de Aveiro há 15 anos. É uma pessoa energética que gosta de jogar ténis e estar com os amigos nos seus tempos livres.

Durante o ano, o Francisco costuma receber a visita de alguns dos seus amigos, não podendo faltar um bom vinho. No entanto, devido ao seu trabalho, Francisco não tem muito tempo de forma a conseguir deslocar-se a uma garrafeira para escolher e comprar uma garrafa de vinho.

**Motivação:** O Carlos gostava de possuir uma forma mais fácil de comprar vinho de forma a que este fosse entregue na sua casa.



*Figura 2: Francisco Silva*

#### Miguel Lopes

O Miguel tem 20 anos e vive em Aveiro, onde concluiu o ensino secundário. É uma pessoa motivada, praticando basquetebol e natação nos seus tempos livres. Além disso o Rafael gosta de andar de mota com os seus amigos.

Durante as férias, o Miguel gostaria de arranjar um part-time de forma a conseguir arranjar mais algum dinheiro para conseguir prosseguir os seus estudos. Uma vez que gosta de andar de mota, uma ideia seria fazer entrega de comida, mas o Rafael prefere alguma coisa “nova” sem ser comida.

**Motivação:** O Rafael gostava de possuir um trabalho part-time fazendo entregas de algo que não fosse comida.



*Figura 3: Miguel Lopes*

## 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 Francisco encomenda um vinho:

Um dos amigos do Francisco vem visitá-lo por isso o Francisco necessita de encomendar uma garrafa de vinho. Por esse motivo, o Francisco abre a web app e depara-se com uma página onde são apresentadas as funcionalidades principais do **DrinkLivery**. Após efetuar login, ele depara-se com uma enorme lista de vinhos.

Após encontrar o vinho pretendido, o Francisco adiciona o vinho ao carrinho e recebe uma mensagem de confirmação, informando-o que o vinho foi adicionado ao carrinho.

#### O Rafael aceita um pedido de entrega de encomenda de vinho:

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.

45426 Teste e Qualidade de Software

# Arquitetura do sistema

## Vista da Arquitetura

Como já foi referido anteriormente, o nosso sistema irá ter 3 grandes componentes:

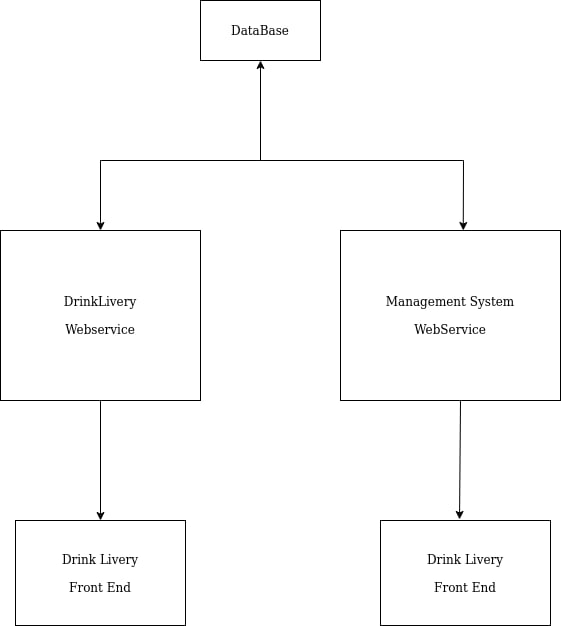
* **Frontend:** Aplicação web
* **Backend:** REST API, processamento, lógica de negócio e acesso à informação
* **Persistência:** Base de dados relacional

Para cada um destes componentes, iremos utilizar uma tecnologia específica. Assim, tem-se que:

* Para a aplicação web, decidimos utilizar **ReactJS**, uma biblioteca JavaScript para construir interfaces de utilizador. Os principais benefícios do uso do ReactJS são a sua vasta biblioteca de modelos e elementos pré-fabricados, a sua facilidade de uso e capacidade de criar elementos gráficos e reativos de uma forma simples e eficaz. A sua popularidade também foi tida em conta.
* Para o módulo de backend, iremos utilizar o **Spring Boot**. Em primeiro, por ser a ferramenta recomendada pelo docente da unidade curricular. No entanto, esta é uma tecnologia extremamente completa, no sentido em que permite:
  + Desenvolver uma REST API, para que as aplicações web e móvel possam comunicar com o nosso sistema
  + Processar toda a informação segundo a nossa lógica de negócio
  + Acessar a informação armazenada em persistência
  + Realizar testes unitários e de integração (e outros)
* Para a camada de persistência, iremos utilizar uma base de dados relacional em **MySQL**. A principal razão por termos escolhido este sistema de gestão de bases de dados deve-se, em grande parte, à familiaridade de todos os elementos do grupo na sua utilização. Outro motivo deve-se ao

facto de existirem diversas relações entre as entidades do nosso sistema, pelo que somos obrigados a utilizar uma base de dados relacional.

De seguida, apresentamos o diagrama tecnológico da nossa arquitetura:



*Figura 5: Diagrama de Arquitetura*

45426 Teste e Qualidade de Software

# API para Developers

A REST API está feita de forma a permitir realizar pedidos, sendo utilizador ou sendo uma aplicação externa. As *queries* permitem um acesso controlado à BD fazendo ao mesmo tempo o tratamento de dados, caso necessário. Não tendo esta sido *deployed* na máquina virtual fornecida pelo professor, iremos apenas apresentar os subdiretórios do URL.

|  |  |  |
| --- | --- | --- |
| **Subdiretório** | **Método** | **Descrição** |
| /v1/users/register | POST | *Endpoint* utilizado para inscrição do utilizador na plataforma. |
| /v1/users/login | POST | *Endpoint* utilizado para o *login* dos utilizadores, aceita no seu corpo o email e a *hash* da password do utilizador. |
| /v1/allproducts | GET | *Endpoint* retorna uma lista com os produtos, neste caso, os vinhos existentes na base de dados. |
| /v1/request | POST | *Endpoint* para realização do *checkout* e criação de um pedido. Aceita uma lista de vinhos. |
| /v1/riders/login | POST | *Endpoint* utilizado no login de estafetas. Aceita o email e a *hash* da password. |
| /v1/riders/register | POST | *Endpoint* utilizado para registar estafetas na plataforma do *Management*. |
| /v1/riders | GET | *Endpoint* retorna uma lista de estafetas com todos os seus atributos. |
| /v1/riders/{id} | GET | *Endpoint* retorna um estafeta especificado no atributo de URL *id*. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

*Tabela 4: Descrição de alguns endpoints da REST API*