

# TQS: Product specification report

*Ricardo Rodriguez [98388], Pedro Sobral [98495], Daniel Figueiredo [98498], Eva Bartolomeu [98513]*

v2022-06-02

<b>Introduction</b>	<b>2</b>
Overview of the project	2
Limitations	2
<b>Product concept</b>	<b>2</b>
Vision statement	2
Personas	3
Main scenarios	4
Project epics and priorities	5
<b>Domain model</b>	<b>5</b>
<b>Architecture notebook</b>	<b>6</b>
Key requirements and constraints	6
Architectural view	7
Deployment architecture	8
<b>API for developers</b>	<b>8</b>
<b>References and resources</b>	<b>9</b>

# 1 Introduction

## 1.1 Overview of the project

No âmbito da cadeira de Teste e Qualidade de Software, foi-nos proposto um projeto composto por um desenvolvimento de um produto de software viável, juntamente com uma estratégia de Garantia de Qualidade de Software (SQA).

EasyDeliver é uma plataforma que fornece vários serviços/entregas logísticas que são comuns no comércio eletrónico. Por exemplo, é possível gerir a plataforma de entregas como admin, gerir dinamicamente o trabalho dos riders (otimizar a atribuição de encomendas ao rider e a sua classificação), permitir o registo de riders, aceitar encomendas por parte do rider, etc.

Esta plataforma é reutilizável em diferentes contextos/aplicações, fornecendo serviços habituais nas entregas de encomendas rápidas, com uma implementação que pode ser usada por diferentes aplicações end-user. Estas aplicações end-user enviam, rastreiam e avaliam os pedidos de entrega para esta plataforma reutilizável.

O PetWay é uma plataforma de entrega de produtos de animais online. Este serviço faz parceria com lojas de animais e permite que os clientes encomendem produtos dessas lojas usando um website.

Os objetivos do trabalho são:

- Ter um sistema seguro.
- Ter uma plataforma capaz de receber várias encomendas.
- Ter uma plataforma capaz de lidar com pedidos de várias lojas.
- Ter uma plataforma capaz de conectar a aplicação mobile dos estafetas com a aplicação que gera as encomendas.
- Desenvolvimento de um produto de software viável (especificação funcional, arquitetura e implementação do sistema).
- Especificação e aplicação de uma estratégia de Garantia de Qualidade de Software (SQA), aplicada em todo o processo de engenharia de software.

## 1.2 Limitations

Em termos de limitações que não foram possíveis de implementar devido a diversos fatores, sendo um deles a falta de tempo, foi a feature de, aquando de ser feita uma encomenda por algum utilizador, o estafeta escolhido não vai ser o estafeta que se encontra mais perto da loja, mas sim um estafeta escolhido ao acaso.

Apesar de ter ficado esta feature por implementar, todas as outras se encontram com os respetivos testes, contudo, pelo mesmo fator do tempo, decidimos não implementar testes relativos a framework utilizada, neste caso, o angular.

# 2 Product concept

## 2.1 Vision statement

O sistema vai ser usado para fazer distribuição de produtos, os clientes encomendam alguma coisa de uma loja parceira e esta encomenda vai ser designada para um estafeta, neste caso, para o que se encontrar mais perto, e conseqüentemente, vai fazer a entrega ao cliente, responsabilizando-se, o condutor, de ir buscar e entregar os produtos aos clientes do sistema.

Enquanto que serviços como a Uber Eats, por exemplo, na sua plataforma, agrega todas as lojas parceiras, o nosso serviço pretende providenciar uma plataforma que seja própria de cada loja parceira dando-lhes maior liberdade na gestão dos seus produtos. Do mesmo modo, apresenta aspetos em comum com a Uber Eats, como por exemplo, o facto de oferecer a possibilidade de beneficiar do sistema de entregas ao domicílio.

O grupo começou por se reunir em fases iniciais, para tratar de qual seria a melhor solução para o tratamento do problema. Com a ajuda do docente, conseguimos retificar melhor as ideias que tínhamos e perceber que estávamos no bom caminho e que apenas tínhamos de acrescentar mais alguma informação na aplicação genérica.

## 2.2 Personas

O desenvolvimento do EasyDeliver, foi suportado em torno de 4 personas, de modo a podermos fazer, de um modo realista, o workflow desta aplicação (cliente, estafeta, empresário e manager). Deste modo, temos como *personas*, o Mário, que é um radiologista; o Luís, que é um estudante de medicina veterinária, a Albertina, que tem uma pequena loja de produtos animais e a Carla que é uma manager.

Mário tem 30 anos e é um radiologista de renome, vive no Porto, não é casado e vive sozinho. Apesar de viver sozinho, ele não está verdadeiramente sozinho, pois ele tem consigo os seus dois cães, contudo como é um radiologista de renome ele não tem muito tempo para lhes comprar coisas que eles possam necessitar pois encontra-se muito atarefado, quer seja com os seus serviços de radiologia no hospital, quer seja com serviço voluntário que ele realiza sempre que pode ou quer seja com as inúmeras conferências que ele tem de atender. **Motivação:** O Mário gostaria de obter uma maneira mais simples de poder satisfazer a necessidade dos seus cães.

Luís tem 20 anos e encontra-se a frequentar o terceiro ano no curso de medicina veterinária. O Luís escolheu este curso, pois sempre gostou de animais e quando era mais novo ajudou a salvar um gato que se encontrava dentro de um rio e a partir daí quis fazer sempre o melhor que podia pelos animais. Como o curso que o Luís está a frequentar não se paga sozinho, ele tem de arranjar mais algum dinheiro para se conseguir sustentar, deste modo, ele é também estafeta sempre que pode. **Motivação:** O Luís precisa de dinheiro e preferia estar a fazer algo em que pudesse estar a ajudar os animais.

Albertina tem 50 anos, é casada e tem 3 filhos, o Lourenço, a Concha e a Pilar, vive na zona de Cascais e tem uma pequena loja de produtos animais e apesar de gostar de passar a maior parte do seu tempo na loja e de conviver com os seus clientes, ela sente que os rendimentos se encontram a baixar. **Motivação:** A Albertina deseja expandir o seu negócio de forma a conseguir que os seus rendimentos voltem a subir.

Carla tem 35 anos, é casada e tem um filho de 4 anos. Ela adora passar o tempo com o seu filho, mas também gosta muito de jogar futebol, então sempre que pode vai jogar e é bastante competitiva. Deste modo, Carla é muito assertiva e sabe comandar uma equipa e controlar toda a informação da empresa que se encontra a trabalhar como manager. **Motivação:** A Carla quer gerir uma equipa e ser capaz de fazer a empresa em que se encontra a crescer.

## 2.3 Main scenarios

Os cenários de contexto descrevem como o sistema proposto se integra com as atividades dos usuários finais para aprimorar como eles as executam para cumprir suas motivações.

No primeiro cenário retrata quando Mário abre a aplicação e tenta fazer uma compra.

**Mário faz reconhecimento da aplicação** - Mário entra na aplicação e vê a página para que ele possa fazer login. Depois de fazer o login, como cliente, é-lhe apresentada a página onde pode ver os produtos que estão disponíveis na aplicação, decide então fazer a compra de alguns.

**Mário faz a compra de produtos** - Mário, depois de ter feito o reconhecimento da aplicação, vai à página onde se encontram os produtos disponíveis e procede a escolher os produtos que deseja comprar e adicioná-los ao carrinho. Ao clicar no carrinho são-lhe apresentados os produtos que ele escolheu, bem como quantidades e preço total e procede a fazer o check out dos produtos.

No segundo cenário retrata quando Maria adiciona produtos à sua loja.

**Maria faz reconhecimento da aplicação** - Maria entra na aplicação e procede a fazer login, como manager, é-lhe apresentada a página onde pode ver as informações acerca da sua loja e pode também ver todos os produtos disponíveis na sua loja.

**Maria faz a adição de um produto** - Maria vai à página dos produtos e procede a clicar no botão, o que a leva para a página de adição de um produto, sendo que esta página tem campos para serem preenchidos, de forma a criar um produto. Maria preenche os campos e faz então a criação do produto e é-lhe apresentada uma mensagem de sucesso.

No terceiro cenário retrata quando Luís, o estafeta, aceita uma encomenda e verifica o seu histórico de encomendas.

**Luís faz reconhecimento da aplicação** - Luís entra na aplicação e procede a fazer o login. Depois de fazer o login, como estafeta, é-lhe apresentada a página onde pode ver os as encomendas que estão disponíveis e aceita uma.

**Luís verifica o histórico das suas encomendas** - Luís vai para a página do histórico das suas encomendas e é-lhe apresentado todas as encomendas que este realizou. Ao clicar numa delas consegue visualizar todos os detalhes dessa encomenda.

No quarto cenário retrata quando Carla acede à página para ver as informações do sistema.

**Carla visualiza as informações do sistema** - Ao fazer login como administradora do sistema, é apresentada uma página que contém todas as informações acerca do sistema.

2.4 Project epics and priorities

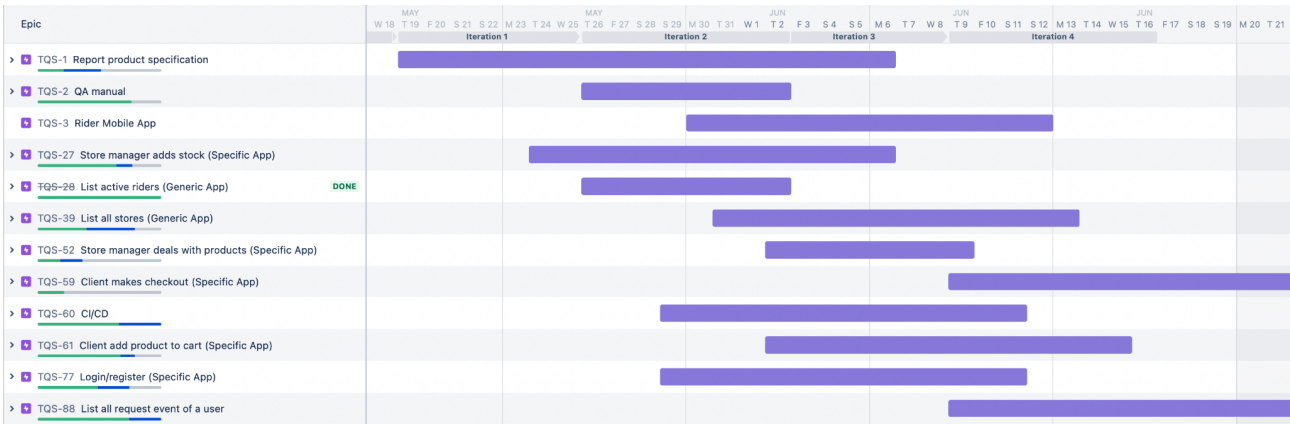


Figure 1 | Épicos e prioridades.

3 Domain model

A base de dados do serviço específico e do serviço geral foram ambas desenvolvidas em MYSQL. Como podemos observar pela figura 2, o modelo da base de dados do serviço específico, ficou um pouco complexo, contudo, sentimos que eram classes que precisávamos realmente, para o bom funcionamento do sistema.

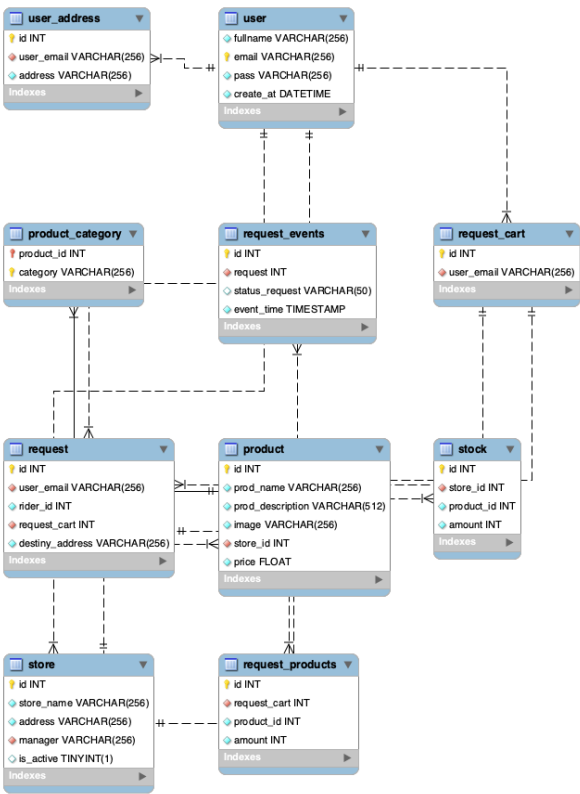


Figure 2 | Domain model serviço específico

No caso do modelo da base de dados do serviço geral, temos três classes que se relacionam entre si, de modo a que se consiga comunicar com o outro serviço.

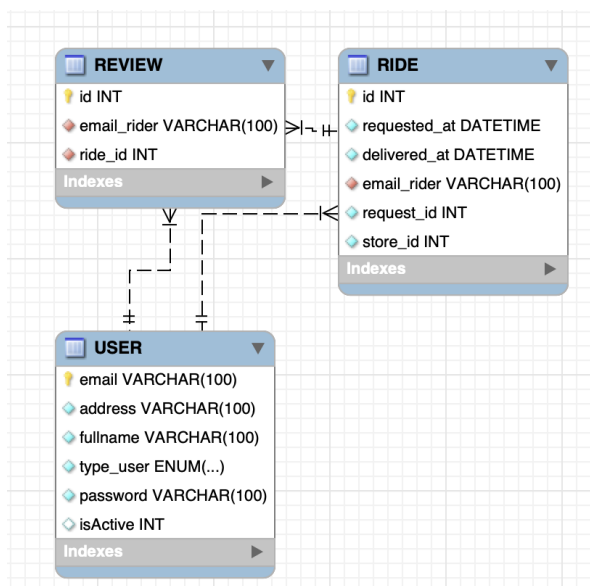


Figure 3 | Domain model serviço geral.

## 4 Architecture notebook

### 4.1 Key requirements and constraints

Existem alguns requisitos-chave e restrições do sistema que têm uma influência significativa na arquitetura, como, por exemplo:

- O sistema deve ser capaz de hospedar várias lojas, tendo a capacidade de receber vários pedidos ao mesmo tempo.
- O sistema tem de ter um sistema de autenticação seguro.
- O sistema deve ter uma plataforma para os estafetas de forma a que estes consigam ver as suas encomendas e as notificações destas.
- O sistema deve ter uma plataforma que permita ao administrador controlar o workflow do mesmo.
- O sistema deve ter uma plataforma que permita que um manager consiga fazer o controlo da sua própria loja.

## 4.2 Architectural view

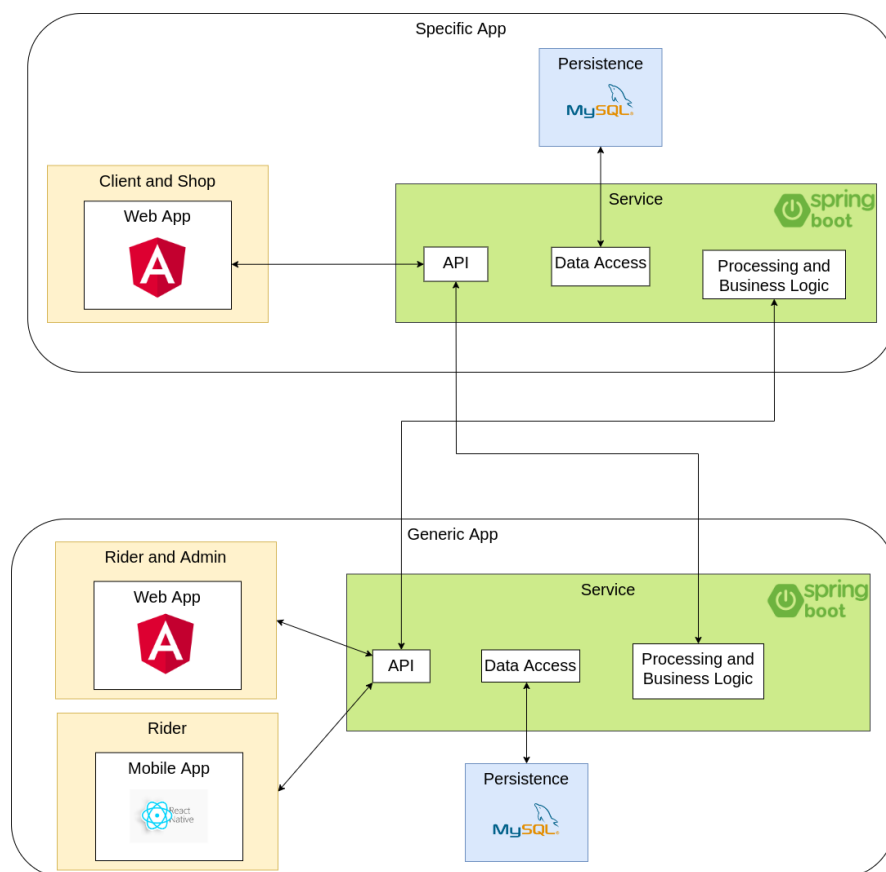


Figure 4 | Arquitetura para a solução de software.

Tanto a plataforma PetWay (Specific App) como a plataforma EasyDeliver (Generic App), vão ter uma aplicação web desenvolvida com a tecnologia **Angular**. Esta escolha foi devido ao facto de todos os elementos do grupo já terem trabalhado com esta tecnologia, e porque esta tecnologia traz um desenvolvimento rápido e de boa qualidade. Para facilitar o trabalho dos estafetas, criamos também uma mobile app, implementada em React Native, para poderem controlar melhor as suas encomendas.

Quanto à persistência vamos utilizar **MySql**, em ambas as plataformas, para armazenar e gerir os dados da base de dados.

O back-end vai ser elaborado com o **Spring Boot**, visto que nesta componente tínhamos restrições de implementação, só poderíamos utilizar o Spring Boot ou o Java EE technology. Descartamos logo o Java EE technology, pois nenhum elemento de grupo tinha trabalhado com isso, já o contrário com o Spring Boot.

Através do Spring Boot na componente Back-end é possível aceder aos dados da base de dados, depois são processados e expostos nas aplicações web de ambas as plataformas, por meio da sua respectiva Rest Api (Figure 4).

### 4.3 Deployment architecture

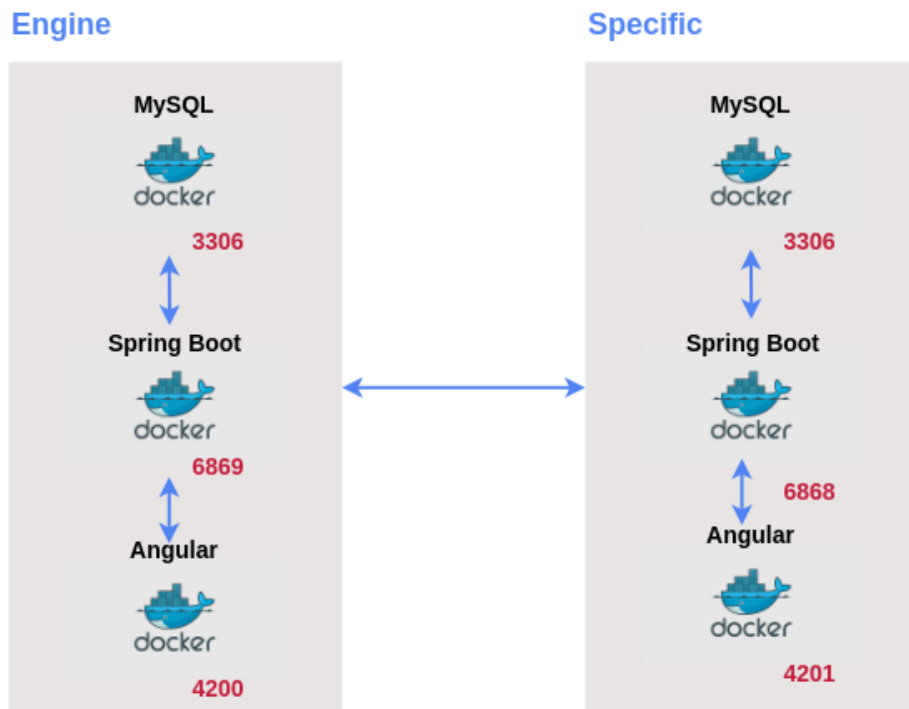


Figure 5 | Deployment architecture.

Para produção foi pensada na organização demonstrada na figura acima, onde os serviços (base de dados, spring-boot, e angular) estão em containers quer do lado do engine, quer do lado do specific.

## 5 API for developers

Quer a api do engine, quer do specific, encontram-se documentadas, e a documentação da mesma pode ser consultada nos seguintes links:

API engine: <https://app.swaggerhub.com/apis/tqs-42/engine/1.0.0#/>

API specific: <https://app.swaggerhub.com/apis/tqs-42/specific/1.0.0#/>



## 6 References and resources

- [1] Swagger (software) <<https://swagger.io/>>
- [2] “Seamless integration with GitHub”, Circle Internet Services, Inc.  
<[https://circleci.com/integrations/github/?utm\\_source=google&utm\\_medium=sem&utm\\_campaign=sem-google-dg--emea-en-githubCI-maxConv-auth-nb&utm\\_term=g\\_e-ci%20cd%20github\\_c\\_rsa1\\_20220516&utm\\_content=sem-google-dg--emea-en-githubci-maxconv-auth-nb\\_keyword-text\\_rsa-githubCI\\_mixed-&qclid=CjwKCAjw-8qVBhANEiwAfjXLrsGYC2qKW02hqnlcKfThQkZAKqSQ-hczflvx7qWUoYhyXR293MU7ihoC4MYQAvD\\_BwE](https://circleci.com/integrations/github/?utm_source=google&utm_medium=sem&utm_campaign=sem-google-dg--emea-en-githubCI-maxConv-auth-nb&utm_term=g_e-ci%20cd%20github_c_rsa1_20220516&utm_content=sem-google-dg--emea-en-githubci-maxconv-auth-nb_keyword-text_rsa-githubCI_mixed-&qclid=CjwKCAjw-8qVBhANEiwAfjXLrsGYC2qKW02hqnlcKfThQkZAKqSQ-hczflvx7qWUoYhyXR293MU7ihoC4MYQAvD_BwE)>
- [3] “Dockerize Angular Application”, Bharathiraja (published)  
<<https://medium.com/codex/dockerize-angular-application-69e7503d1816>>
- [4] “Dockerizing a PHP Application”, Rafie Younes (published)  
<<https://semaphoreci.com/community/tutorials/dockerizing-a-php-application>>
- [5] “Connecting Multiple Application Instances”, Spring Cloud Stream Core  
<[https://docs.spring.io/spring-cloud-stream/docs/Brooklyn.RELEASE/reference/html/\\_inter\\_application\\_communication.html](https://docs.spring.io/spring-cloud-stream/docs/Brooklyn.RELEASE/reference/html/_inter_application_communication.html)>