deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

TQS: Relatório de Especificação do Produto

Rui Coelho[86182],
Luís Costa[85044],
João Laranjo[91153],
Pedro Iglésias[89318]
v2020-06-05

1 Introdução

1.1 Visão do projeto

Projeto realizado no âmbito da disciplina de Testes e Qualidade de Software e tem como principais objetivos:

- Desenvolvimento de uma especificação de produto, desde os casos de uso até ao design técnico
- Propor, justificar e implementar uma arquitetura de software, com base nas estruturas empresariais;
- Aplicar práticas de trabalho colaborativo em equipas de desenvolvimento ágil assim como desenvolvimento contínuo e integrado. Apresenta-se 4Wheels, uma aplicação web e mobile que tem como objetivo possibilitar e facilitar e compra e venda de automóveis.

Posteriormente foram criados, quer o repositório git, quer o PivotalTracker (ferramenta utilizada para gestão de backlog), de modo a que estivessem reunidas todas as condições para se dar início ao trabalho.

1.2 Limitações

Praticamente todas as features, em termos de software, foram implementadas. A única que não foi implementada, contudo, era almejada pela equipa, era a utilização de sensores para a monitorização do estado da build. O mesmo não foi possível uma vez que não dispúnhamos do material necessário devido ao Covid que nos impediu de deslocar a Aveiro.

2 Conceito do produto

2.1 Visão

O produto consiste numa apresentação da informação detalhada relativa a cada veículo.
Desta forma, deve ser possível a cada pessoa que utiliza a aplicação procurar e obter informação sem necessidade de realizar muita pesquisa e perda de tempo. Para além disso a venda de um automóvel pessoa deve ser também fácil para que quer empresas do setor que pessoas singular possam fazer e promover o seu negócio.
A aplicação será, então, usada para compra e venda de um produto, automóveis. O utilizador poderá:

- Pesquisar por carros
- Ver os favoritos de um utilizador
- Ver todos os carros disponíveis
- Ver informações de um carro

Este produto vai, assim, facilitar a vida dos seus utilizadores que passarão a ter uma interface, única, que junta um número enorme de automóveis possibilitando uma escolha mais fácil e informada.
A 4Wheels surgiu da necessidade de facilidade de controlo de dados e divulgação de informação.

2.2 Personas

António:

- O António tem 43 anos, mora em Braga, trabalha como gestor e adora novas tecnologias. É uma pessoa ponderada, atenta à atualidade, amigo do ambiente e poupado. O António viaja muito e tem um carro com alguns problemas, anda a pensar trocar de viatura mas precisa de se informar melhor.

João:

- O João tem 26 anos e vive em Aveiro. Trabalha numa empresa de software perto de casa e pretende usar um aglomerado de dados para desenvolver a sua aplicação. Como tal procura uma API com informações de carros.

Anabela:

- A Anabela é gestora de uma empresa automóvel. Gostava de expandir o seu negócio para o digital, como tal procura uma plataforma que a ajude com essa tarefa.

2.3 Cenários principais

Cenário 1:

- O António viu um modelo de um carro que gostou, tem um preço fantástico e tecnologia de ponta. Tendo ligação à internet, e autenticação feita, ao entrar na página principal pode procurar pelo carro que viu.

Cenário 2:

- O João pretende fazer uma aplicação para dispositivos móveis que apresenta vários modelos de carros assim como estatísticas e preços dos mesmos. Tendo ligação à internet, e autenticação feita, ao fazer um GET da API, recebe em formato JSON os dados que pretendia.

Cenário 3:

- A Anabela recebeu um novo modelo de um carro e os negócios têm corrido bem no entanto queria expandir o seu negócio para o digital para que possa ter uma maior margem de lucro. Tendo ligação à internet, e autenticação feita, a Anabela poderá colocar à venda os automóveis que acabaram de chegar.

2.4 User Stories

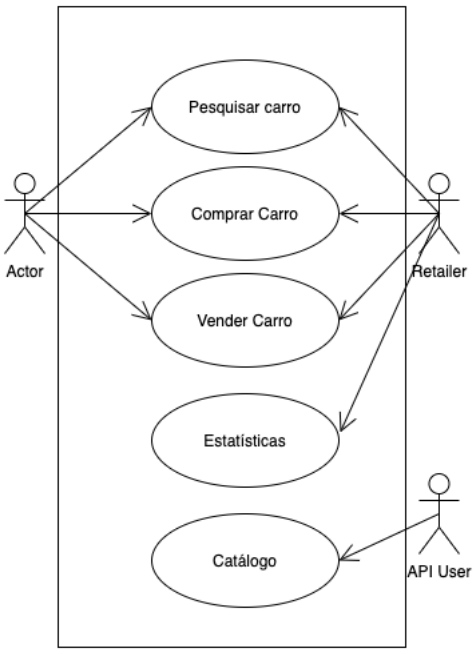
De acordo com os cenários e as personas anteriormente abordadas, assim como a definição do projeto de alto nível (epics) foram criadas diferentes User Stories. Os User Stories desenvolvidos foram colocados no backlog (Pivotal Tracker) em função da prioridade e complexidade.
https://www.pivotaltracker.com/n/projects/2447398

2.5 Prioridades

ITERAÇÃO	EPICS
ITER1	Pesquisar um carro.
ITER2	Consultar informação de um carro.
ITER3	Colocar um carro para venda.
ITER4	Editar um carro em venda.
ITER5	Marcar um carro como vendido.
ITER6	Consultar carros vendidos.
ITER7	Ver estatísticas.

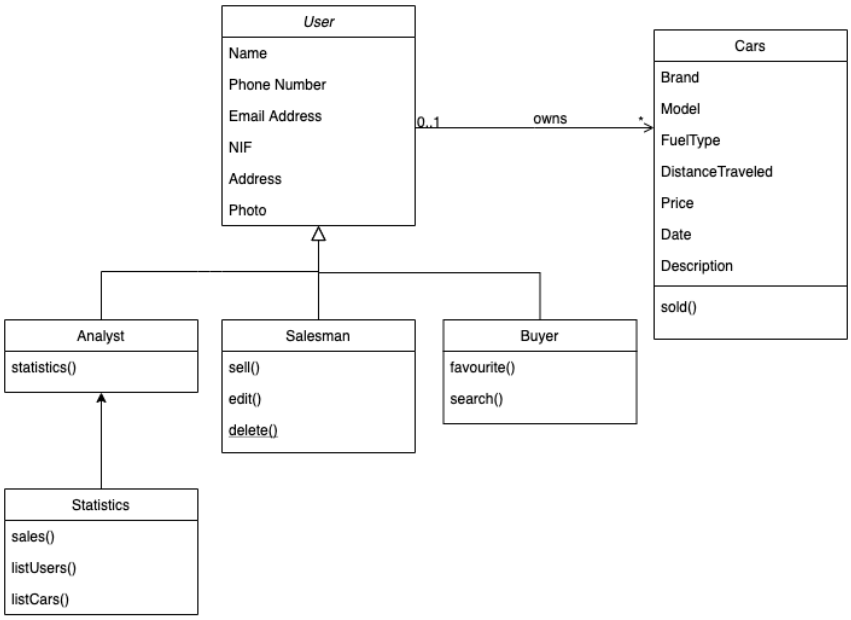
Na forma de tabela é mostrada de que forma existiu um desenvolvimento incremental por epics. É importante denotar que as iterações acima representadas não apresentam um correlação de um para um com as Milestones previstas no plano da UC, visto que houve Milestones nas quais foram completadas várias iterações.

2.6 Casos de Uso



Foram criados três tipos de atores e um diagrama de casos de uso para melhor entender as principais funcionalidades da aplicação.

3 Modelo do Domínio



User	Entidade responsável por guardar a informação de cada um dos utilizadores.
Cars	Entidade responsável pela informação relativa aos carros.
Analyst	Tipo de utilizador que pode consultar estatísticas de utilização da aplicação.
Salesman	Tipo de utilizador que pode colocar carros para venda, editá-los ou mesmo removê-los.
Buyer	Tipo de utilizador mais comum que pretende usar a aplicação para procurar um carro e entrar em contacto com o vendedor do mesmo. Pode também ter uma lista de carros favoritos.
Statistics	Informação relativa à aplicação. Vendas, Utilizadores e Carros.

4 Arquitetura

4.1 Requisitos chave e restrições

Principais requisitos:

- Os dados deverão ser atualizados e editáveis;
- Deve haver a possibilidade de promover produtos;
- A pesquisa deve ser o mais detalhada possível;

Restrições:

- Tempo para a realização do projeto é curto, devemos focar no mais importante.

Principais problemas:

- Como desenvolver o frontend para multiplataforma?
- Qual a melhor forma de armazenar e estruturar os nossos dados?
- Como apresentar os dados da API
- Como lidar com falhas?
- Devemos utilizar novas tecnologias?

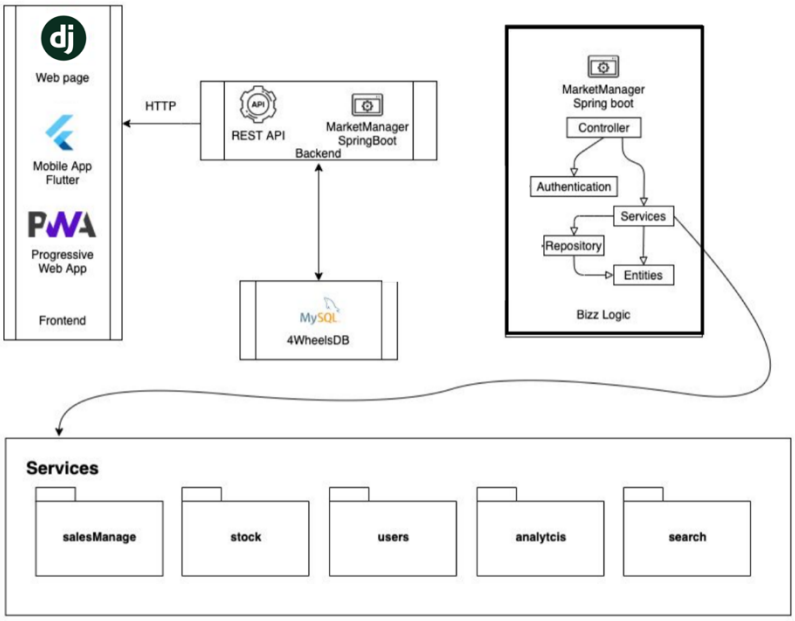
Sendo o levantamento de requisitos uma das etapas importantes no desenvolvimento da aplicação foram feitas entrevistas a conhecidos e parentes que procuravam um novo veículo ou que tinham um stand de venda de automóveis e gostariam de expandir o seu negócio.

4.2 Vista da arquitetura

- CLIENTE - Dentro do cliente temos a nossa webApp, que essencialmente mostra ao utilizador os dados, estatísticas. Esta será programada em React e Flutter.
- BACKEND - Temos a business logic que categoriza e processa os diferentes dados (ex: user management, registos, etc). Temos também a exposição da API para outros utilizarem.
- PERSISTÊNCIA - Nesta componente temos uma base de dados de longo termo que irá ajudar a persistir os dados. Esta camada comunica diretamente com o backend daí estar incluído dentro da mesma, MySQL.
- CONSUMER – Aplicação móvel que consome os dados da API.

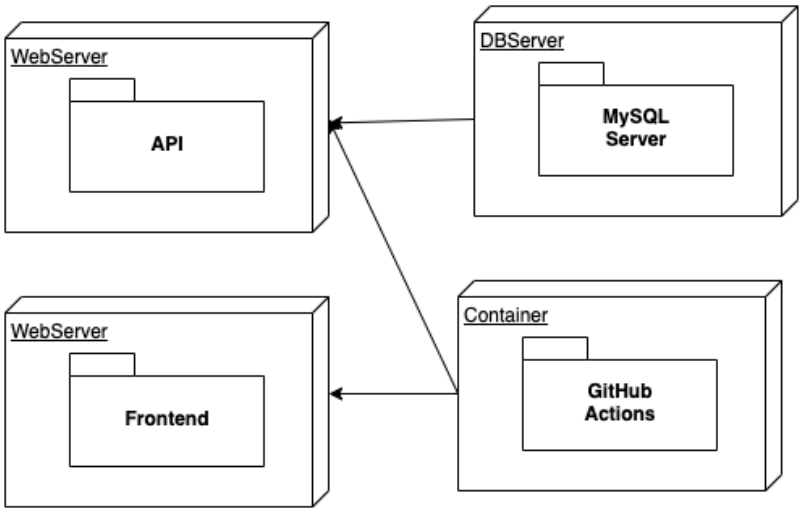
4.2.1 Interações modulares

De forma resumida deve ser possível aos utilizadores submeterem produtos para venda. Isto deve ser feito através do Frontend e da mesma forma para o indivíduo ou para a empresa.
Uma API deve ser exposta/chamada através de Endpoints específicos. A informação deve ser em formato JSON e deverá ser o mais detalhada possível.
Deve existir uma zona de persistência para guardar não só as interações, mas também as informações relativas a cada produto anunciado.



4.3 Arquitetura de instalação

Todo o processo de deployment tem o seu começo no servidor da Microsoft, no caso, o do Github Actions no qual todo o código é previamente testado, após isso, é feito o seu deployment para o heroku o qual se relaciona, no caso da API Spring, com a base de dados e o webserver do frontend e, no caso do frontend, o mesmo só se relaciona com a API Spring.



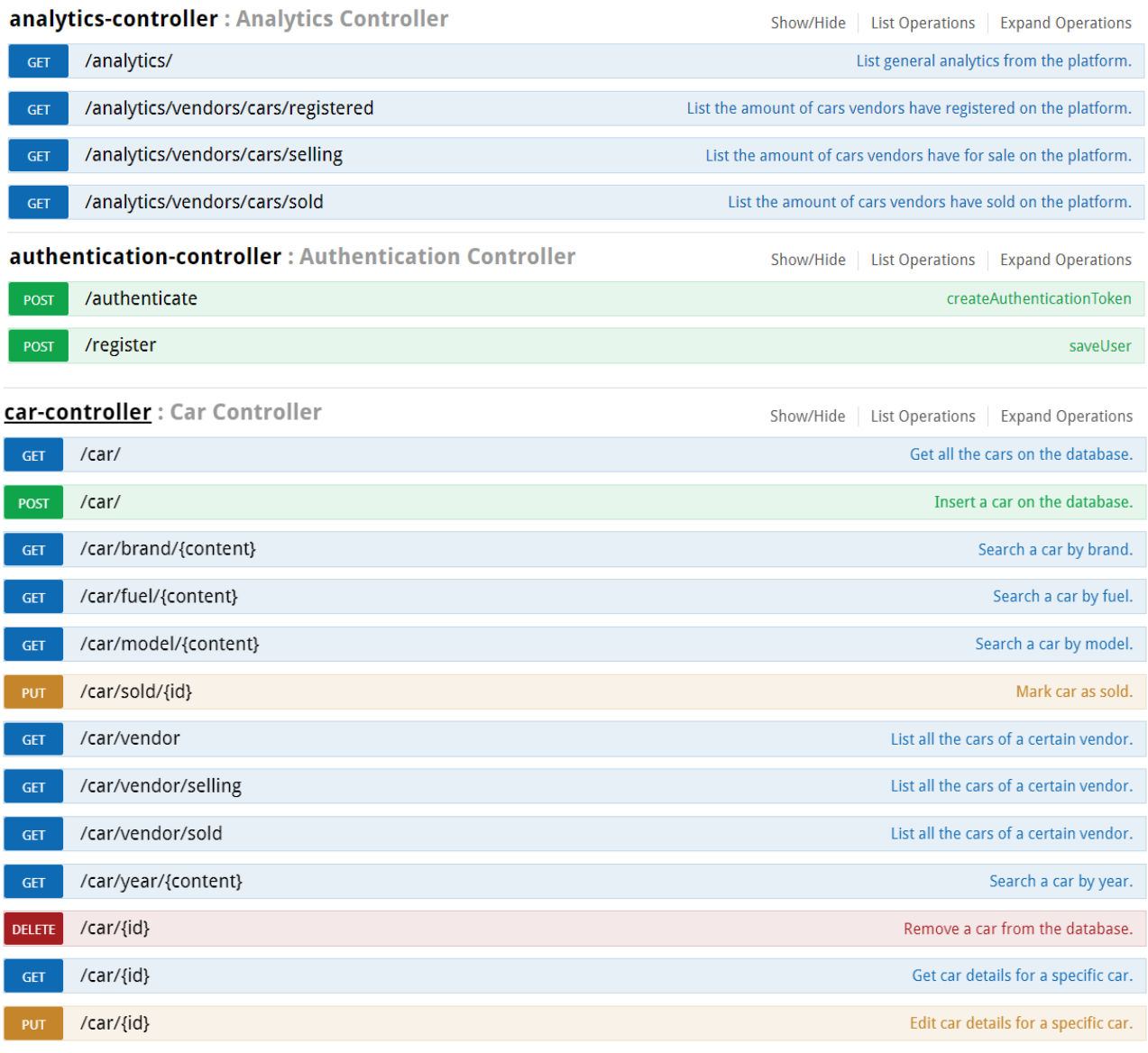
5 API para Desenvolvedores

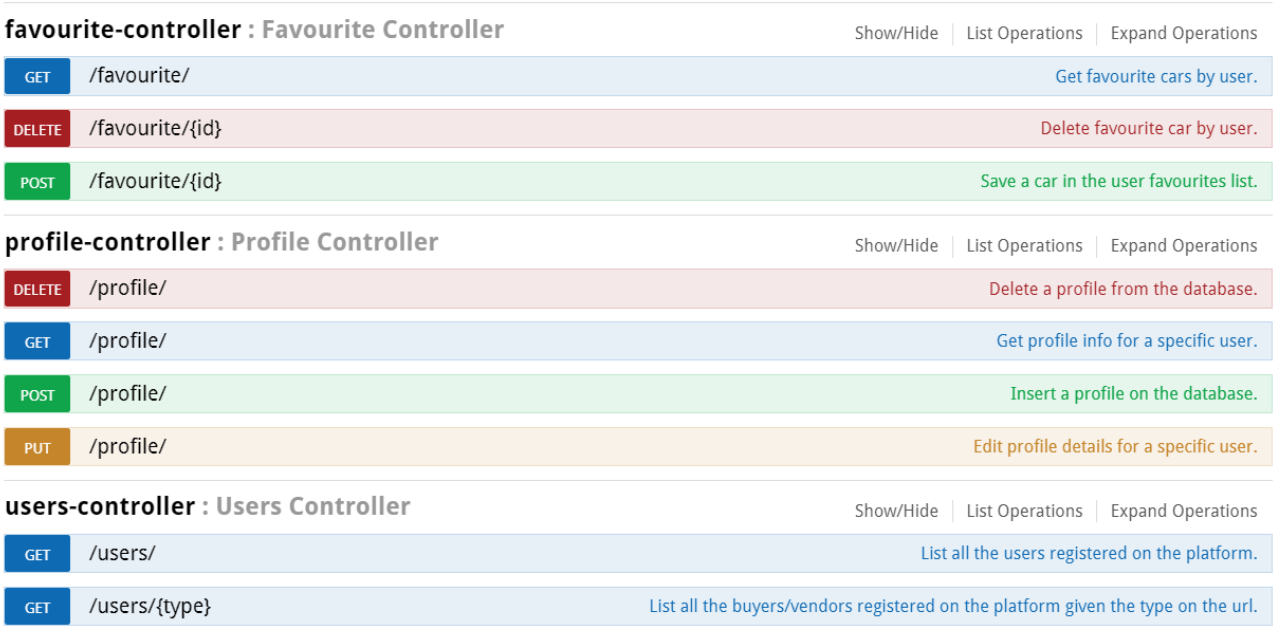
A API encontra-se documentada com a utilização do Swagger. A abordagem para a utilização destes Endpoints consiste na tipologia de utilizador, no caso, neste sistema, podem ser de três tipos:

- Vendedor
- Comprador
- Analista

Consoante o tipo de utilizador a que a conta está associada a reação aos pedidos varia. As validações deste tipo são com recorrência a JWT [1]. À cabeça todos os caminhos estão protegidos com exceção dos que contêm a informação publica dos carros.

https://tqs4wheels-api.herokuapp.com/swagger-ui.html





6 Referências

[1] "Spring Boot Security + JWT Hello World Example," 05 Maio 2020. [Online]. Available: https://www.javainuse.com/spring/boot-jwt.