

Technical Report - Project specifications

FastTravel

Curso: IES - Introdução à Engenharia de Software

Data: Aveiro, 4 de dezembro de 2021

Alunos: 98629: Marta Fradique
98521: Luís Martins
98495: André Freixo
98513: Eva Bartolomeu

Projeto: Sistema de monitorização de portagens eletrónicas, que permite observar o fluxo de passagens, pagamentos automáticos e controle de passagens de cada cliente.

Table of contents:

[1 Introduction](#)

[2 Product concept](#)

[Vision statement](#)

[Personas](#)

[Main scenarios](#)

[3 Architecture notebook](#)

[Key requirements and constrains](#)

[Architetural view](#)

[Module interactions](#)

[4 Information perspective](#)

[5 References and resources](#)

1 Introduction

Nós construímos este projeto no âmbito da aula Introdução à Engenharia de Software, o nosso objetivo é desenvolver um sistema capaz de gerir os pórticos nas autoestradas e outras vias.

Para implementar esta ideia, vamos aplicar conceitos de Engenharia de Software que englobam, principalmente, especificar um produto, desenvolver software com diversas tecnologias e ainda trabalho colaborativo, para tal usamos ferramentas como o GitHub para guardar código-fonte e arquivos, utilizamos ainda o Jira com o intuito de monitorizar as tarefas e acompanhar o desenvolvimento do projeto, deste modo, ao aplicar estas práticas conseguimos melhorar a organização e produtividade.

Ainda, após analisarmos o que pretendemos implementar e as exigências do projeto, idealizamos a arquitetura que melhor se adequava.

Para aprimorar o trabalho colaborativo da equipa cada membro tem o seu respetivo papel:

- Team manager - Eva Bartolomeu
- Product Owner - Marta Fradique
- Architect - André Freixo
- DevOps master - Luís Martins

2 Product concept

Vision statement

O objetivo do sistema é gerir os pórticos em autoestradas e outras vias.

Para tal, tem que monitorar os carros que transitam na via destinada ao Fast Travel e conferir se os mesmos têm o identificador, o trajeto percorrido, o custo da viagem e retirar o correspondente montante da conta bancária associada ao cliente, e assim, agilizar o processo de viagem com pagamentos instantâneos.

A nossa página web tem duas vertentes. O cliente consegue verificar as suas passagens na via Fast Travel, o custo de cada uma e o total mensal. Ainda, o administrador faz a gestão das cabinhas de portagem, adiciona, edita e elimina pórticos, e observa o fluxo de passagem dos carros e filtra este quando quer pesquisar por algo específico como uma matrícula ou por data.

Ora, o nosso sistema é bastante similar ao da Via Verde pois ambos usam pórticos com pagamentos automáticos.

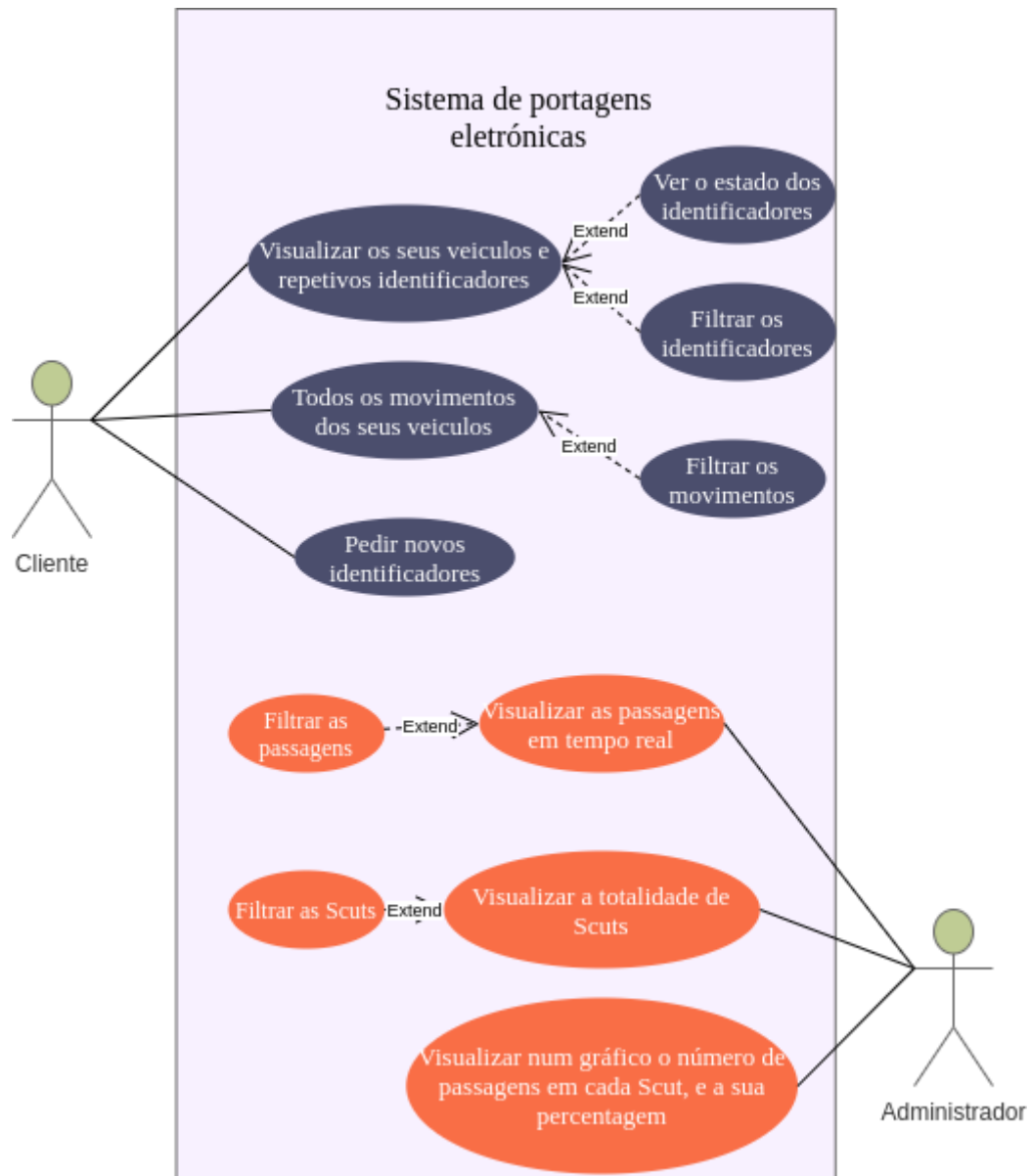


Fig.1 Diagrama UML com os casos de uso do sistema

Personas

- A Carla tem 54 anos, é portuguesa e é **administradora** na Fast Travel há 25 anos. Vive em Espinho, é casada, tem dois filhos e toma conta da sua mãe de 76 anos. Gosta de lasanha, sair com as amigas e de cozinhar. Sonha em um dia ter a oportunidade de ir à Disneyland.
- O António tem 63 anos, é Português, professor da primária e é cliente da FastTravel há 15 anos. Vive em Águeda, tirou licenciatura em matemática, é casado há 30 anos e tem três filhos. Decidiu tornar-se **cliente** da empresa para viajar de forma mais rápida e facilitada.

Main scenarios

Com a aplicação web, um **administrador** do sistema de portagem electrónica, pode visualizar em tempo real as passagens dos automóveis nos pórticos das auto-estradas. Ainda, pode filtrar esta listagem por várias características como procurar por um cliente específico, por matrícula e ainda procurar por dia e hora, entre outras características dos clientes e das passagens. Também, pode visualizar todos os pórticos existentes do Fast Travel, ver a sua localização, a data da sua criação, a sua descrição, assim como os preços das cinco classes existentes. Assim como nas passagens, o administrador pode filtrar as scuts existentes por diferentes atributos das mesmas, como o ID, a data de instalação, entre outros. Tem ainda na página DashBoard estatísticas, na qual pode visualizar em um gráfico e em um diagrama o número de passagens por scut, tal como a sua respectiva percentagem.

Como **cliente**, Pedro pode visualizar todos os seus veículos e respectivos identificadores na página Vehicles and Identifiers, assim como o estado dos identificadores no momento, podendo ainda filtrar pelos diferentes atributos mostrados na página, como a matrícula, o estado do identificador, entre outros. Existe também no lado do cliente uma página com um formulário (Request Identifiers) na qual ele pode pedir um identificador novo associando-o a um determinado carro, entre outros atributos. Por fim, pode ainda observar todos os seus movimentos em todos os seus veículos na página Movements, podendo ver qual é o veículo que utilizou, por onde passou, entre outras informações específicas, podendo também filtrar os movimentos por os seus vários atributos.

Tanto o **administrador** como o **cliente** têm de fazer login para aceder aos seus respectivos perfis, tendo também a opção de fazer sign out depois de entrarem.

Nesta primeira iteração no protótipo foram adicionadas as user stories do administrador de listar as passagens dos automóveis nos pórticos e listar os pórticos da Fast Travel.

3 Architecture notebook

Key requirements and constraints

Para escolhermos a arquitetura adequada tivemos em conta os seguintes aspetos do nosso sistema:

- Este deverá ser capaz de gerar dados, neste caso, relativos aos veículos que atravessam os pórticos da FastTravel, de forma completamente automática, de modo a simular a travessia de veículos reais pelas cabines de portagens de uma autoestrada;
- Como cada veículo deve ser o único, é necessário garantir que o mesmo só passe por um pórtico de cada vez;
- Da mesma forma deve-se garantir que enquanto um veículo esteja num determinado pórtico, não sejam gerados novos veículos nessa mesma cabine de portagem, uma vez que se esta está ocupada não faz sentido que mais veículos sejam gerados;
- O sistema deverá também permitir, não só, visualização dos dados gerados em tempo real, mas também, os deverá armazenar de modo a que seja possível a sua visualização posteriormente;
- Os dados dos clientes devem-se manter confidenciais, de modo que as informações dos utilizadores se mantenham seguras, e que estas apenas sejam alteradas com o seu consentimento.
- O cliente tem ainda a possibilidade de pedir novos identificadores que, após o pedido estar concluído, serão adicionados à base de dados e vão aparecer na página dos Identificadores.

Architectural view

Para desenvolver o sistema vamos criar uma **aplicação web**, para isso, recorreu-se à utilização de tecnologias web, tais como, HTML, JavaScript e CSS, de modo a criar interfaces web interativas para os utilizadores.

Quanto à **persistência** vamos utilizar MySQL para armazenar e gerir os dados da base de dados.

Ainda, para a **geração dos dados** vamos aplicar um Script em Python para criar dados que de seguida vão ser geridos e tratados no Message Broker.

Para o **Message Broker** vamos utilizar o RabbitMQ, que é um software open-source de

processamento de mensagens, este permite o tratamento de dados em tempo real, deste modo, vamos guardar e processar os dados do sistema, e também atualizar os dados caso existam mudanças.

Quanto ao **back end** para nos ajudar a criar a aplicação utilizamos *Spring Boot*, para simplificar o desenvolvimento vamos utilizar também uma *REST API*. Também, existe uma camada que mapeia os dados da base de dados que futuramente vão ser processados no nível lógico do negócio.

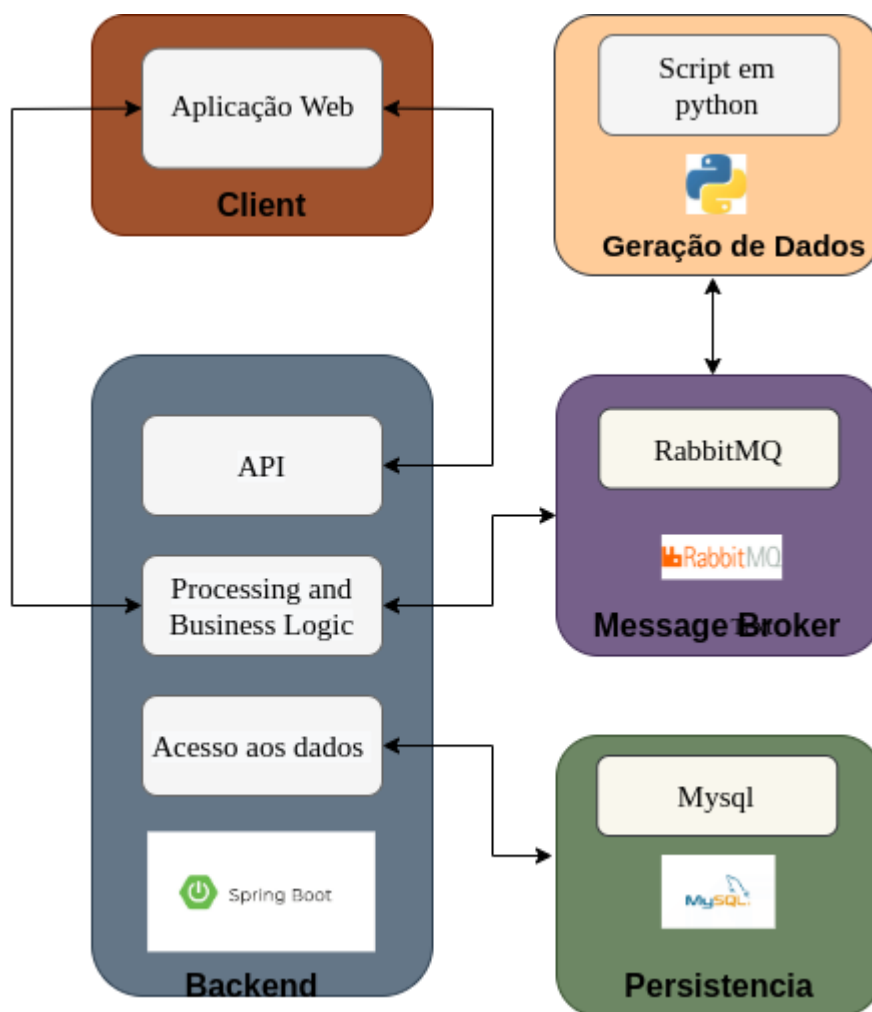


Fig.2 Diagrama da Arquitetura do sistema

Module interactions

Alguns exemplos de Interação entre os módulos previamente expostos:

1. Tal como o nome indica, Script Generator irá gerar os dados e estes serão processados pelo Message Broker;
2. Estes são implementados em Spring Boot na Processing and Business Logic, para serem armazenados nas bases de dados;
3. São enviadas informações para a aplicação web através da Web Socket;
4. Fazendo um pedido à Rest API, o cliente consegue “pedir” para ver na aplicação web essas informações, tais como todas as suas passagens pelas várias scuts (movements);
5. Após os pedidos efetuados, os dados são enviados através da Rest API;
6. E por fim o cliente consegue ver o seu pedido, neste caso os seus “movements”.

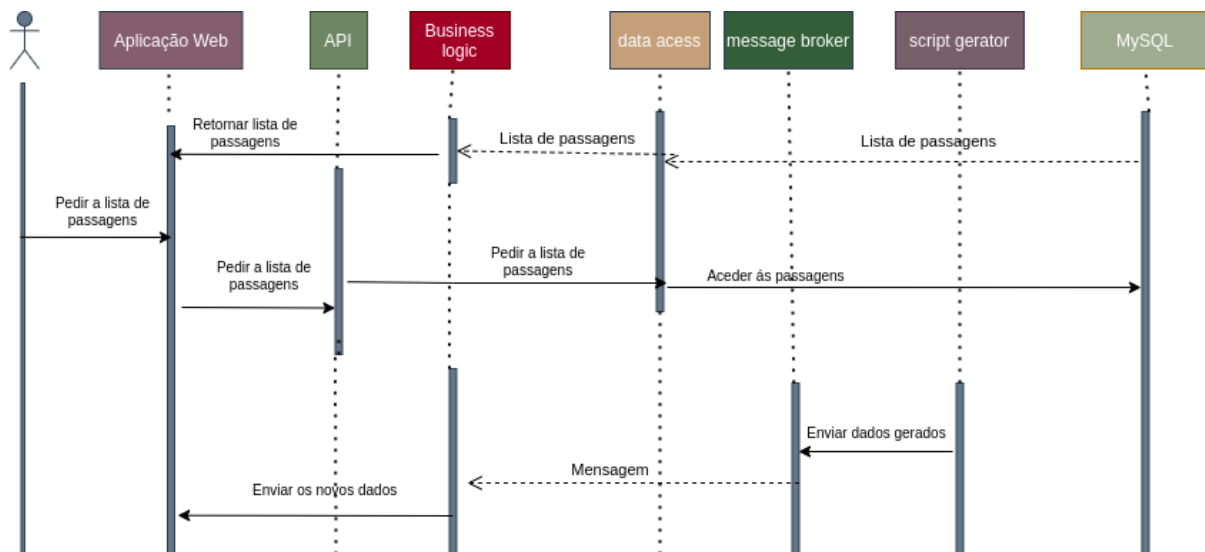


Fig.3 Diagrama de sequência

4 Information perspective

- Administrador
 - admin_id
 - email
 - firstName
 - lastName
 - password
- Credit_Card
 - credit_card_id
 - cvv
 - expiration_date
 - name
 - number
 - pais
- Cliente
 - client_id
 - firstName
 - lastName
 - email
 - nif
 - password
- Scut
 - scut_id
 - description
 - instalation_date
 - latitude
 - longitude
 - price1
 - price2
 - price3
 - price4
 - price5
- Passagens
 - passage_id
 - date
 - time
 - payment_state
 - Identifier_id
 - Scut_id
- Identifier
 - identifier_id

- classe
- registration
- state
- client_id
- credit_card

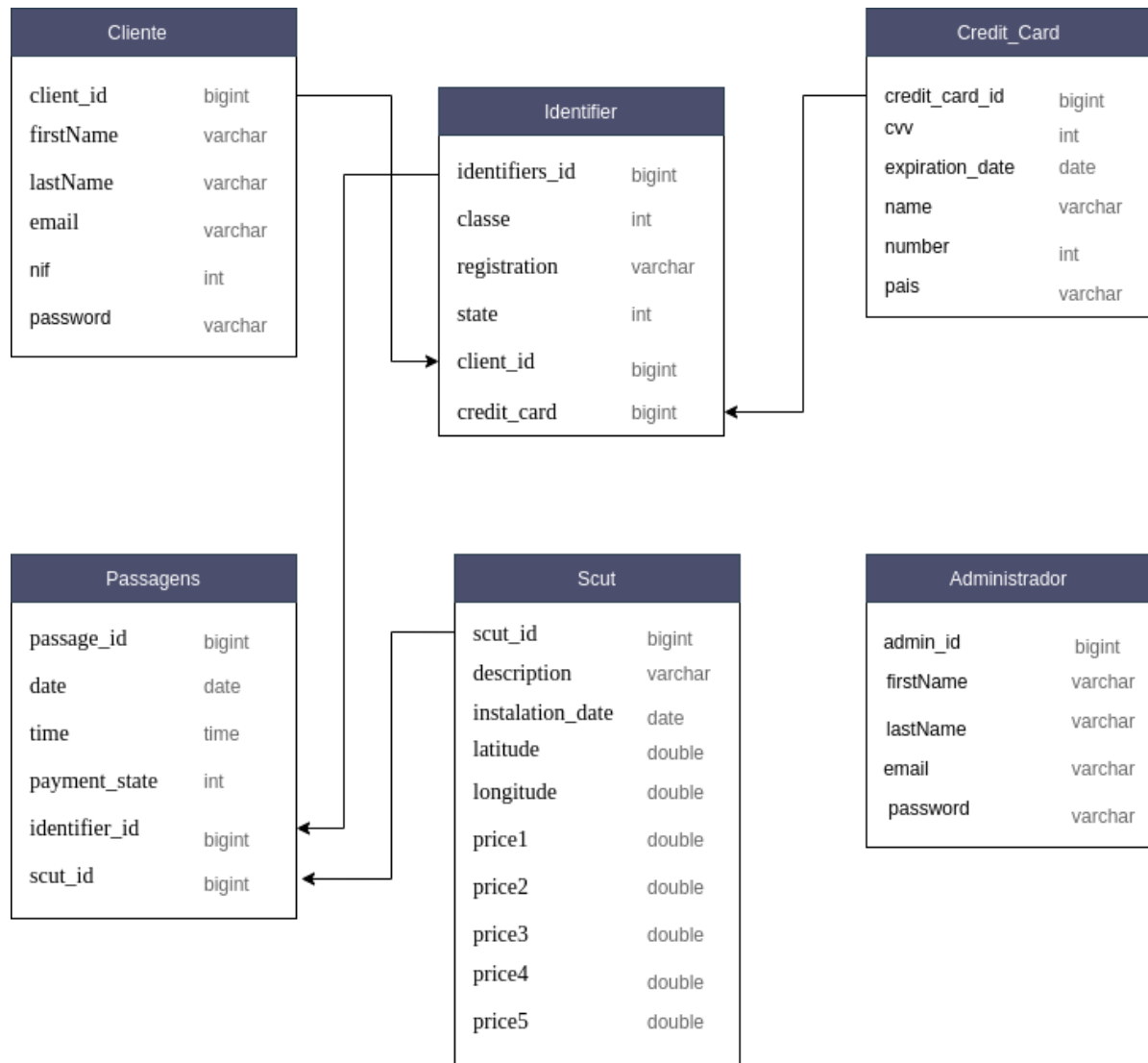


Fig.4 Diagrama de base de dados

5 References and resources

Template usado para o protótipo:

<https://material-ui.com/store/items/minimal-dashboard-free/>

<https://documenter.getpostman.com/view/19335789/UVeAuUEf>

\