



# Mobile Tolling System

David Duarte  
João Duarte

Orientadores    Luís Osório  
                         Paulo Borges

Relatório beta realizado no âmbito de Projecto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2017/2018

Maio de 2018



# **Instituto Superior de Engenharia de Lisboa**

Licenciatura em Engenharia Informática e de Computadores

## **Mobile Tolling System**

Autores:

39374 David Rodrigo Ferreira Rodrigues Duarte

38241 João Pedro Pina Duarte

---

---

Orientadores:

Luís Osório

Paulo Borges

---

---

Relatório beta realizado no âmbito de Projecto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2017/2018

Maio de 2018



# Resumo

O projecto enquadra-se na unidade curricular de Projecto e Seminário e visa concluir o percurso académico da Licenciatura de Informática e de Computadores.

Para um indivíduo que queria usar auto-estradas é possível usufruir do serviço de duas maneiras diferentes. Se já for cliente Via-Verde e tiver acesso ao identificador disponibilizado pela mesma, pode usar a portagem automática. Se ainda não for cliente, tem de usar uma portagem manual.

O nosso serviço vem eliminar a necessidade do cliente Via-Verde ter um identificador, através de uma aplicação móvel, usando o smartphone pessoal do cliente.

Esta aplicação móvel tem como principal responsabilidade detectar eventos de passagem nas portagens (entrada/saída/passagem) e comunicar esses eventos ao *backoffice*.

A solução a este problema começa por gerar um geofence por portagem, escutando o evento de saída do mesmo de forma a perceber se o utilizador de facto passou pela portagem.

Palavras-chave: Pagamento por Geo-posicionamento, GNSS, Aplicação Móvel



# Abstract

The project is part of the course *Projecto e Seminário* and aims to complete the academic course of the Computer and Computer Science Degree.

For an individual who wanted to use highways it is possible to enjoy the service in two different ways. If you are already a *Via-Verde* customer and have access to the identifier provided by the customer, you can use the automatic tolling. If you are not a customer yet, you must use a manual toll.

Our service eliminates the need for the *Via-Verde* customer to have an identifier, through a mobile application, using the client's personal smartphone.

This mobile application has as main responsibility to detect events of passage in the tolls (entrance / exit / passage) and to communicate these events to the backoffice.

The solution to this problem begins by generating a geofence for each toll booth, listening to the exit event of it in order to detect if the user actually went through the toll.

Keywords: Geo-positioning Tolling, GNSS, Mobile App





# Índice

<b>Resumo</b>	<b>5</b>
<b>Abstract</b>	<b>7</b>
<b>Índice</b>	<b>9</b>
<b>Lista de Figuras</b>	<b>11</b>
<b>Lista de Tabelas</b>	<b>13</b>
<b>Introdução</b>	<b>14</b>
<b>Conceitos</b>	<b>15</b>
<b>Estado da arte</b>	<b>17</b>
Evolução	17
O nosso Use Case	17
Outros Use Cases	17
Suíça	17
Austrália	17
Portugal	18
<b>Formulação do problema</b>	<b>19</b>
Principal Problema - Detecção de portagens	19
Geofencing	19
Problemas Secundários	19
Visão Geral do Sistema	20
Requisitos do sistema	21
Requisitos funcionais	21
Requisitos não funcionais	21
Funcionalidades	22
Funcionalidades expostas ao condutor	22
Funcionalidades expostas ao Clearing	23
Registo de veículos	23
Registo de utilizadores	23
<b>Soluções propostas e Definição de modelos</b>	<b>24</b>
Arquitectura da solução	24
Arquitetura da aplicação móvel	25
Interface gráfica:	25
Ponto de entrada	26
Vista de Login	26
Vista de Principal	27

Navegação	28
Veículos	29
Notificações	29
Vista de Veículo	30
detalhes do Veículo	30
viagens do veículo	31
Arquitetura interna	32
Arquitetura do Backoffice	34
Endpoints	34
Negociação de conteúdo	35
Eventos do Clearing	35
Autenticação	35
Camada de dados	35
Sincronização com o Clearing	35
Diagrama de Sequência - Ciclo de vida de uma viagem	36
<b>Conclusões</b>	<b>37</b>
<b>Referências</b>	<b>38</b>
<b>A.1 Diagramas da Aplicação</b>	<b>39</b>
<b>A.2 Modelos de dados</b>	<b>40</b>

# Lista de Figuras

Figura 1 - Diagrama geral do ecossistema.	20
Figura 2 - Diagrama de Use-Case.	22
Figura 3 - Diagrama geral da arquitectura	24
Figura 4 - Diagrama da interface gráfica da aplicação	25
Figura 5 - Vista splash activity	26
Figura 6 - Vista login activity	26
Figura 7 - Vista main activity	27
Figura 8 - Vista main activity-navegação	28
Figura 9 - Vista main activity-veículos	29
Figura 10 - Vista main activity-notificações	29
Figura 11 - vista vehicle activity	30
Figura 12 - vista vehicle activity-detalle	30
Figura 13 - vista vehicle activity-transacções	31
Figura 14 - Diagrama implementação MVP	32
Figura 15 - Diagrama da arquitectura de injeção de dependências	33
Figura 16 - Arquitectura do backoffice	34
Figura 17 - Diagrama de sequência ciclo de uma viagem	37



## **Lista de Tabelas**

# Introdução

Actualmente os dispositivos móveis têm se vindo a tornar cada vez mais populares, pouco dispendiosos e com uma oferta variada de funcionalidades, sendo uma delas o geo-posicionamento. Tirando partido de que os possíveis utilizadores, muito provavelmente, possuem um destes dispositivos e que o mesmo oferece geo-posicionamento, pretendemos desenvolver um sistema informático que visa facilitar o uso dos serviços de mobilidade disponibilizados pela Via-verde<sup>[1]</sup>, mais especificamente, o serviço de portagens, modernizando o mesmo, enriquecendo a experiência de utilização de infraestruturas de transporte e mobilidade já existente.

Actualmente, os clientes Via-verde podem usufruir das auto-estradas sem necessidade de paragem através do uso de um identificador *OBU*(*On Board Unit*)<sup>[1]</sup> instalado nas viaturas. Tendo em conta que é necessário este identificador estar associado ao veículo e que o pagamento fique associado ao mesmo, com um custo e tempo de registo, identificámos que existe uma oportunidade para melhoria do serviço. Tirando partido do facto de que um utilizador possui um *smartphone* que fornece geo-posicionamento, vamos explorar a disponibilidade de oferecer o serviço de portagens, facilitando assim a experiência do serviço Via-Verde aos clientes que não têm acesso directo ao *OBU* ou que pretendam uma alternativa ao mesmo.

Outra funcionalidade, prevista como opcional, é a delegação de permissão de veículos de terceiros que pode vir a provar ser bastante enriquecedora, tendo em conta o número de aplicações que potencialmente pode ter.

Neste ponto já introduzimos os nossos principais problemas a resolver que são a técnica de detecção de portagens e a delegação e revogação de permissão de veículos de terceiros. Mais à frente vamos entrar em mais detalhe sobre estes problemas e discutir possíveis soluções, salientando o que nos levou a escolher uma solução em relação às outras.

Para já, vamos introduzir conceitos chave para uma melhor compreensão do problema como um todo e, de seguida, falar um pouco sobre o estado da arte desta área.

# Conceitos

## **Portagem manual**

Portagem em que o utilizador tem de parar na cancela para obter o bilhete à entrada ou pagar à saída antes de poder avançar.

## ***OBU***

*On-Board Unit*. É o dispositivo onde actualmente se baseia o sistema de portagens. Serve para identificação do veículo ao passar numa portagem.

## **Portagem automática**

Portagem que obtém automaticamente informação de identificação dos veículos sem que estes precisem de interromper o seu movimento.

Temos como exemplo: portagem com câmeras que obtêm foto frontal e traseira da matrícula do veículo e/ou é capaz de comunicar com *OBUs* presentes dos veículos dos clientes.

## ***Clearing***

Entidade responsável pelo gerenciamento do sistema de portagens que recebe informação das concessionárias.

## ***ETC***

*Electronic Toll Collector*. Sistema de portagem móvel ou não, que colecciona informação sobre os veículos que transitam pela portagem.

## ***Enforcement***

Processo que assegura a conformidade da informação adquirida pelo sistema.

## ***GNSS***

*Global Navigation Satellite System*, sistema que através do uso de satélites consegue disponibilizar informação de geolocalização.

## ***Geofencing*[3]**

Perímetro virtual que engloba uma área geográfica do mundo real, e nos permite detectar entradas e saídas do mesmo.

## ***DSRC***

*Dedicated short-range communications*. Tecnologia de comunicação sem-fios, localizada na banda de espectro 5,9 GHz, usada actualmente entre *OBUs* e portagens automáticas.

## **Evento**

Um evento define uma interação entre o utilizador e a sua passagem numa portagem.

**Transação**

Viagem em que o utilizador vai usufruir dos serviços da Via-Verde. Cada transação tem pelo menos 1 e no máximo 2 eventos.



# Estado da arte

## Evolução

Os sistemas de portagens rodoviárias começaram por instituir portagens manuais em que todos os veículos tinham de parar para tirar o bilhete e posteriormente pagar no final da viagem na auto-estrada e o enforcement utilizado era apenas a barreira da portagem.

Esta solução trazia vários problemas como o fácil congestionamento, devido à paragem obrigatória de todos os veículos para registar a sua entrada ou efetuar o pagamento.

Surgiu então uma nova solução que evitava este problema. Essa solução foi a *ORT(Open Road Tolling)*.

A *ORT* introduziu um novo componente ao sistema chamado de *OBU* que iria estar presente em cada veículo de cada cliente e comunicar através de uma antena com a portagem através de comunicação sem-fios *DSRC*.

O nosso projecto vem com o intuito de testar a viabilidade de uma solução unicamente baseada em geo-posicionamento.

## O nosso Use Case

A abordagem utilizada vai de encontro à proposta da *LinktGo*[2] na Austrália, o funcionamento do serviço vai focar-se numa aplicação móvel que tira partido das funcionalidades de *GNSS*, existente no próprio dispositivo. Será detectado quando o utilizador transita por plazas de portagem, identificando os percursos que efectuou, sendo possível comunicar com um serviço de *leasing*. Neste caso, a via-verde recebe confirmação dos percursos efectuados e mais tarde é comunicado por parte do *leasing* o estado do pagamento, sendo que a nossa aplicação não se responsabiliza por essa componente; apenas disponibiliza informação sobre a mesma.

## Outros Use Cases

Existem variados tipos diferentes de *tolling* já implementados nas estradas em todo o globo. O modelo mais usado é a combinação entre *OBU* com comunicação por rádio com uma barreira de portagem, usando como enforcement vários métodos i.e. reconhecimento de imagem de matrícula, *GNSS* através do *OBU*.

## Suíça

É utilizado um sistema com um *OBU* com capacidades de geo-localização e detecção de movimento, conectado ao odómetro do veículo para ajudar a evitar fraudes.

## Austrália

É oferecida a oportunidade de utilizar portagens sem a necessidade de um *OBU*, tirando partido de uma aplicação móvel (*LinktGo*) que utiliza a funcionalidade de geo-posicionamento do dispositivo móvel.

## Portugal

O modelo usado actualmente em Portugal[1] para *open-road tolling* é um sistema cooperativo entre um dispositivo instalado no veículo (*OBU*) e uma barreira de portagem que contém uma antena, capaz de comunicar com o anterior através de *DSRC* um meio de comunicação rádio (5.8 GHz), também como câmeras que obtêm uma foto frontal e traseira da matrícula do veículo.

# Formulação do problema

## Principal Problema - Detecção de portagens

O ponto fulcral do sistema é a detecção de transição de portagens.

Como já foi referido anteriormente, o dispositivo móvel do utilizador vai detectar a sua geo-posição com o fim de comparar a mesma com as geo-posições das portagens.

### Geofencing

Para responder a tal problema será usado o serviço de *Geofence* da *Google*[3] para a plataforma *Android*, permitindo representar regiões geográficas, designadas por *geofence*. Esta tecnologia permite monitorizar a geo-localização do dispositivo e gerar eventos quando é detectado que o dispositivo ultrapassa a fronteira do *geofence*, seja uma entrada ou saída do perímetro.

Existem algumas limitações associadas a este serviço de *Geofence* como limitação de registo de 100 *geofences* por aplicação por dispositivo.

## Problemas Secundários

- Falha na detecção de portagens;
- O telemóvel do utilizador é desligado antes ou durante a viagem.

## Visão Geral do Sistema

O Mobile Tolling System (MTS) vai ser enquadrado no contexto da Via-Verde[1] com o intuito de reforçar os seus serviços. Haverá partilha de transações de portagens dos utilizadores com o sistema de *Clearing* da Via-Verde[1]. Este adquire também informação de transacções em portagens por parte dos *ETCs* das concessionárias. Neste caso, a *Brisa*, *Lusoponte*, *AEA* e *Ascendi* formam o aglomerado de entidades que fornece informação, permitindo validar percursos efectuados e cobrança dos valores associados pelos utilizadores Via-Verde[1].

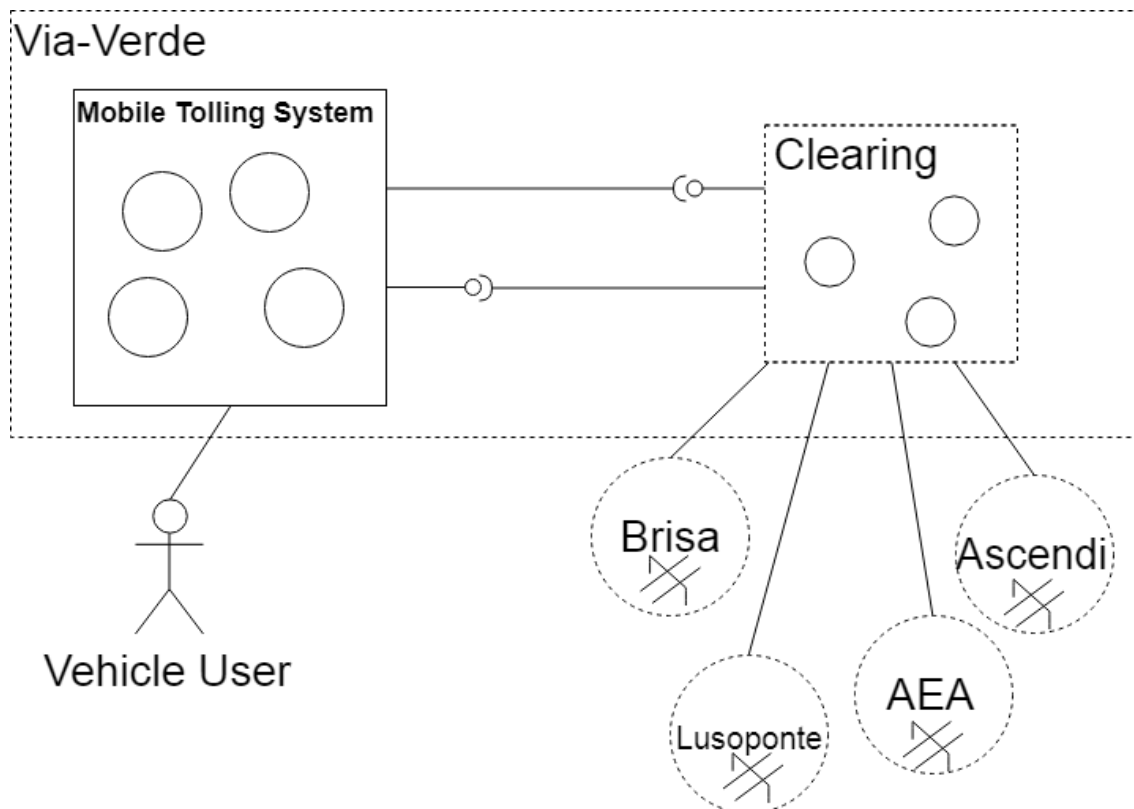


Figura 1 - Diagrama geral do ecossistema.

## Requisitos do sistema

Para que o sistema funcione como um todo, concluímos que seria necessário implementar os seguintes requisitos:

### Requisitos funcionais

- Registo e autenticação de um utilizador no sistema (encaminhar registo da brisa para o serviço apropriado);
- Registo de veículos e sua classe para uso de portagens;
- Entrada e saída de portagens;
- Visualização do percurso efectuado;
- Serviço de histórico da utilização de veículos;
- Delegação e revogação de permissão temporária para uso e pagamento de veículos por parte de outros utilizadores(opcional);
- Simulação de preços de percursos(opcional);
- Validação e pagamento dos percursos efectuados(opcional).

### Requisitos não funcionais

- Garantir resistência a falhas por parte de deteção do geo-posicionamento;
- Garantir aspectos de segurança relacionados com a autenticação(opcional);
- Optimização da experiência de utilização(opcional).

## Funcionalidades

Neste diagrama podemos observar as relações entre os seguintes actores e os use cases do sistema.

Um *Vehicle User* representa qualquer pessoa com acesso à nossa aplicação *Android*. Após a mesma efectuar o registo na Via-Verde e concordar com os termos de uso do serviço MTS, é considerada utilizadora do serviço e pode usufruir de todas as funcionalidades indicadas na Figura 2.

O *Clearing* representa o sistema da Via-Verde, do qual não somos responsáveis.

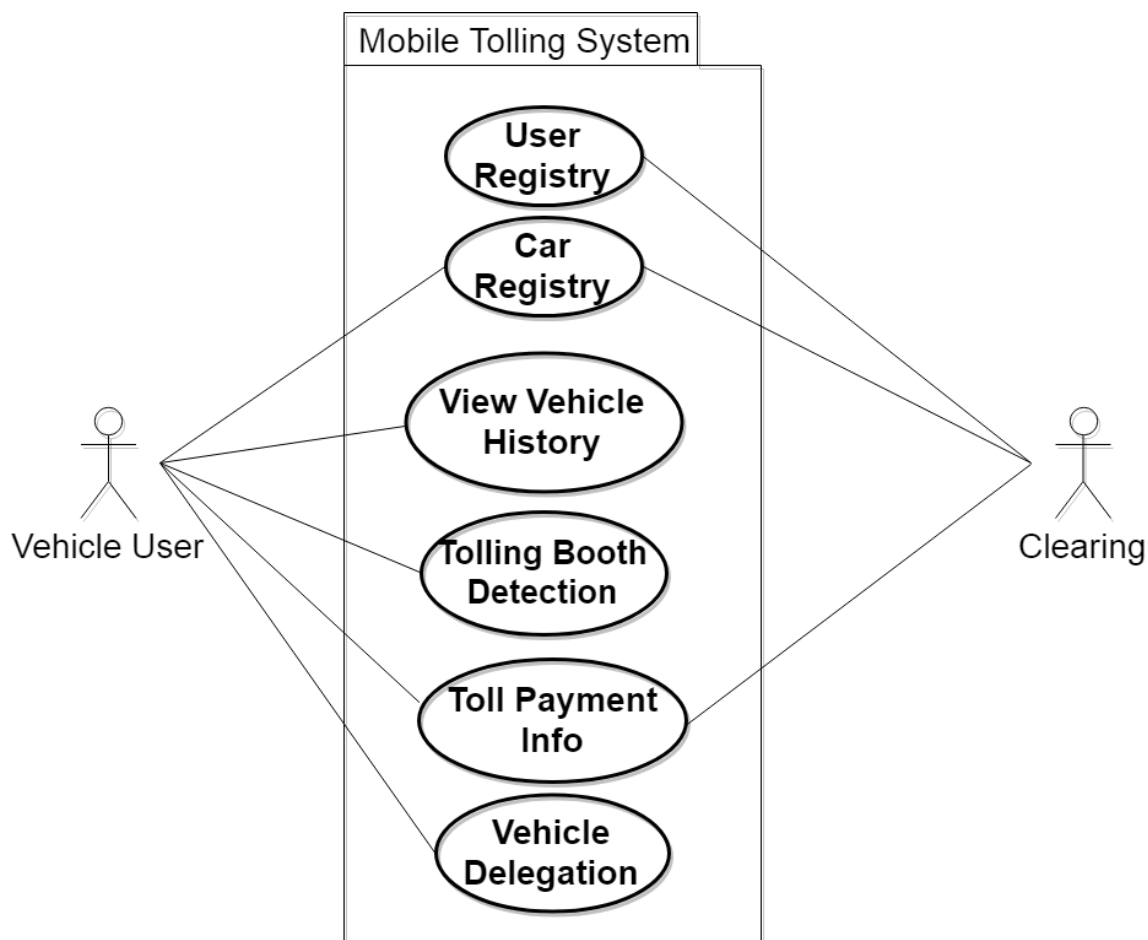


Figura 2 - Diagrama de Use-Case.

### Funcionalidades expostas ao condutor

Utilizando a aplicação *Android* disponibilizada, o condutor é capaz de registar veículos no serviço Via-Verde, ver o seu histórico de viagens, verificar pagamentos, delegar veículos a outros utilizadores do MTS e iniciar viagens, designado pelo use case “*Tolling Booth Detection*” na figura 2.

## **Funcionalidades expostas ao Clearing**

O MTS é responsável por integrar os dados dos seus utilizadores com a Via-Verde, obtendo informação sobre o utilizador e os seus veículos.

### **Registo de veículos**

Para além desta integração é possível registar veículos no serviço da Via-Verde. Será também possível ocorrer sincronização de dados quando é inserido um novo veículo por meio exclusivo da Via-Verde.

### **Registo de utilizadores**

Para um indivíduo ser utilizador do serviço MTS tem de intrinsecamente ser cliente Via-Verde e, como tal, é necessário estar registado na mesma.

A autenticação dos utilizadores do serviço MTS é feita com as respectivas credenciais da Via-Verde.

# Soluções propostas e Definição de modelos

## Arquitectura da solução

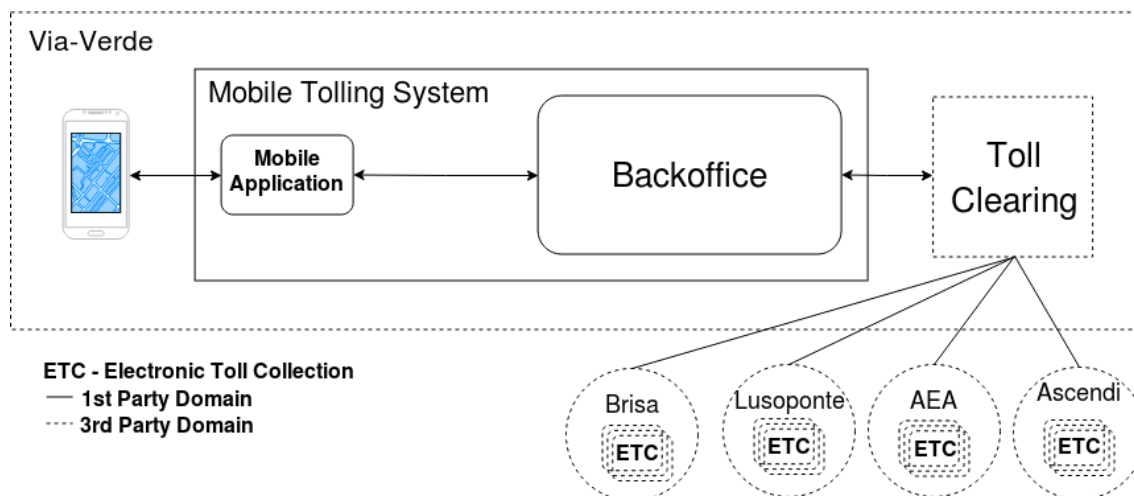


Figura 3 - Diagrama geral da arquitectura

O MTS, divide-se em duas principais componentes. A componente móvel, *Mobile Application* na figura 3, que se encarrega de toda a interação com o condutor, da interface de utilização do sistema e da detecção de passagem em portagens.

A componente *Backoffice*, encarrega-se de oferecer informação relevante para a componente móvel, como a localização das portagens mais próximas e quais os veículos que o utilizador tem disponível. Este recebe informação de utilização de portagens por parte do condutor e envia-a para o sistema de *Toll Clearing* para que este o valide e inicie o processo de cobrança, qual não é responsabilidade do MTS.



## Arquitetura da aplicação móvel

A aplicação móvel de modo a disponibilizar as funcionalidade necessárias é organizada da seguinte forma:

### Interface gráfica:

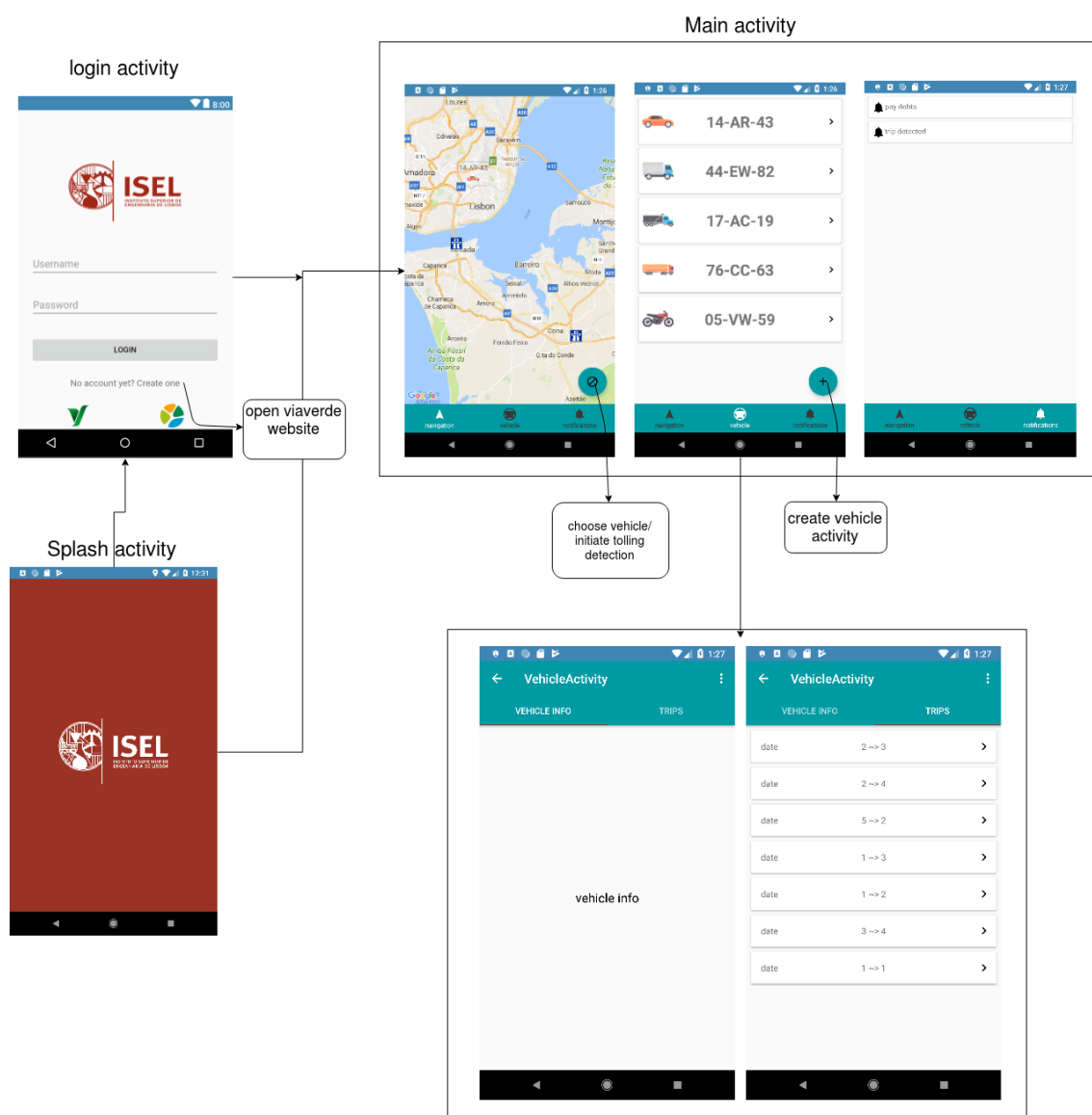


Figura 4 - Diagrama da interface gráfica da aplicação

## Ponto de entrada



Figura 5 - vista splash activity.

## Vista de Login



Figura 6 - vista login activity

O ponto de entrada na aplicação, primeira vista a ser mostrada, é a *splash activity*, está presente enquanto a aplicação carrega e verifica se o utilizador se encontra com autenticação feita, caso seja verdade redireciona o mesmo para a vista principal, *Main activity*, caso contrário é redirecionado para a vista de login, *Login Activity*.

Esta deve estar o menor tempo possível presente visto que impede o utilizador de interagir com a aplicação.

Nesta vista é apresentada a funcionalidade de autenticação, o utilizador pode inserir as credenciais da sua conta no Clearing, Via-Verde, caso sejam válidas são guardadas para não ter de voltar a fazer login até fazer logout e é redirecionado para a vista principal, *Main Activity*, caso sejam inválidas são requisitadas de novo visto não ser possível utilizar a aplicação sem uma sessão válida. No caso do utilizador não ter conta este pode pressionar na opção de criar conta, o mesmo é redirecionado para o site da Via-Verde onde poderá registar uma nova conta, visto que o registo contas não é da nossa responsabilidade, após a ter a conta validada pode fazer autenticação na nossa aplicação.

## Vista de Principal

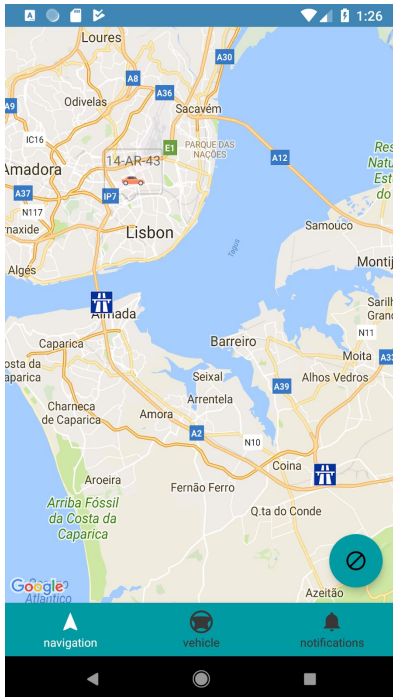


Figura 7 - vista main activity

Esta é a vista principal, *Main Activity*, possibilita acesso rápido às três funcionalidades principais:

- acesso a navegação com detecção de portagens, é a predefinida das três possíveis, visto ser onde o utilizador pode activar a detecção de portagens e iniciar a sua viagem sem ter de se preocupar mais com navegação na aplicação.
- Veículos, mostra a lista de veículos que o utilizador tem disponível, e disponibiliza acesso à vista de veículo, *Vehicle Activity*
- Notificações, mostra a lista de notificação por confirmar leitura, disponibiliza acesso a uma vista de detalhe da notificação.

## Navegação

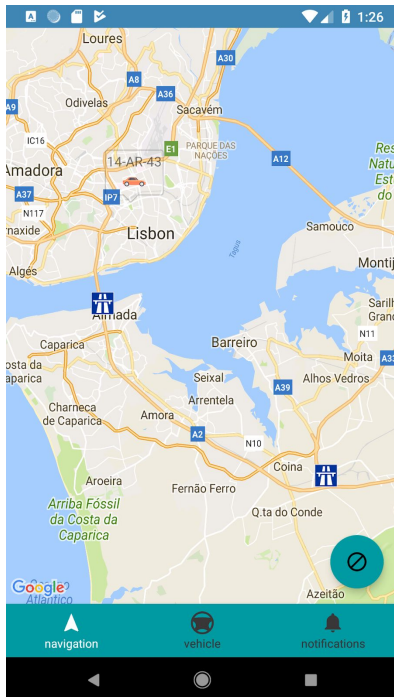


Figura 8 - vista main activity-navegação

A navegação é a funcionalidade principal da aplicação, disponibiliza acesso a iniciar detecção de portagens e a escolher o veículo a usar nas mesmas através de um *floating action button (fab)*, presente no canto inferior direito. Auxilia o uso da aplicação através de um mapa, utilizando o google maps[8], onde é possível visualizar a localização actual, o veículo seleccionado tal como as portagens mais próximas e em caso de já se encontrar numa autoestrada qual a portagens que foi detectada. É ainda possível saber detalhes e corrigir detecções de portagem pressionando a portagem no mapa e escolhendo uma opção de correcção.

Na situação em que o utilizador se encontra previamente autenticado, o início de detecção de portagens são apenas três pressionamentos, o de abrir a aplicação, um no *fab*, para apresentar os veículos e outro no veículo a utilizar, proporcionando assim uma experiência de utilização agradável.

## Veículos

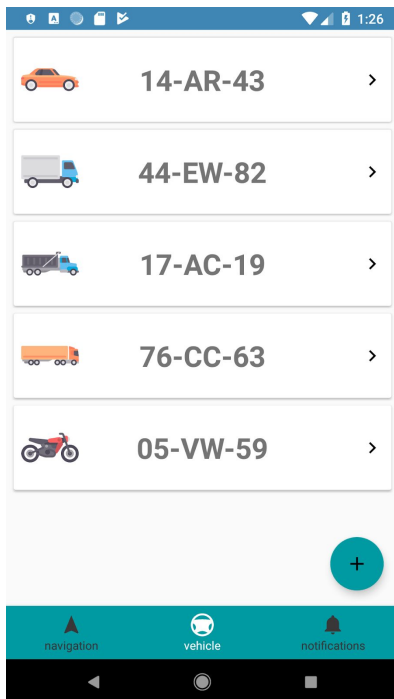


Figura 9 - vista main activity-veículos

Nesta vista é possível visualizar a lista de veículos registados na aplicação, e navegar para a vista de detalhes dos mesmos pressionando um deles. É ainda possível registar novos veículos através de um *fab* presente no canto inferior direito, que redireciona o utilizador para uma vista onde pode inserir os dados do veículo a ser registado

## Notificações

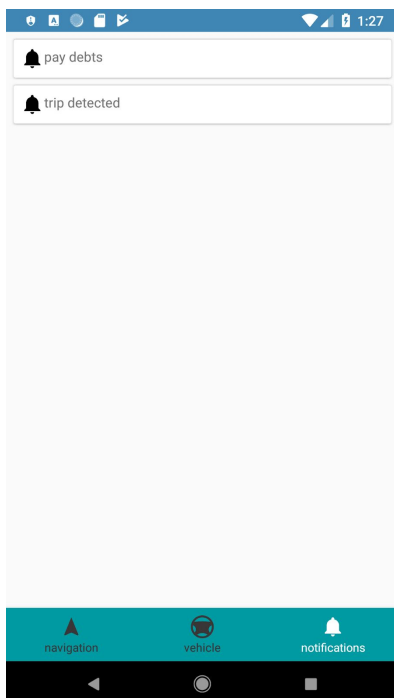


Figura 10 - vista main activity-notificações

Nas notificações é possível visualizar a lista de notificações por confirmar leitura, estas devem ter vários tipos, como por exemplo notificação de novo veículo disponível, transacção em portagem detectada, notificação de pagamento efetuado ou erro no mesmo. O pressionamento em uma das notificações redireciona o utilizador para a vista de detalhes da mesma, onde pode confirmar leitura.

## Vista de Veículo



Figura 11 - vista vehicle activity

## detalhes do Veículo

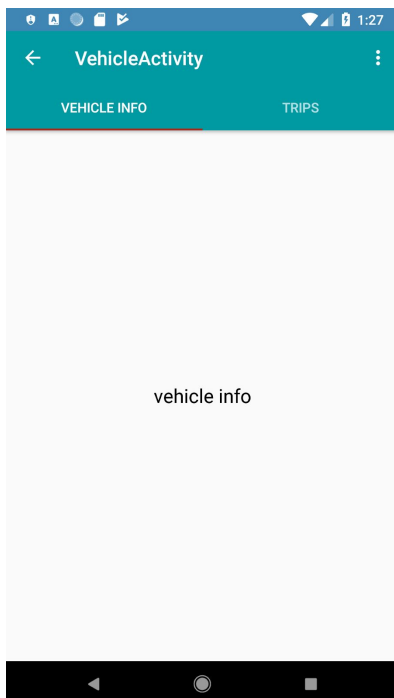


Figura 12 - vista vehicle activity-detalle

Nesta vista é possível aceder a duas funcionalidades principais sobre o veículo escolhido:

- Informação detalhada do veículo
- Lista das transacções em portagens efectuadas com o veículo em questão.

Nesta vista é possível visualizar informações detalhadas do veículo tais como a matrícula, classe, dono, número de transacções em portagens efectuadas e o dinheiro gasto com as mesma.

## viagens do veículo

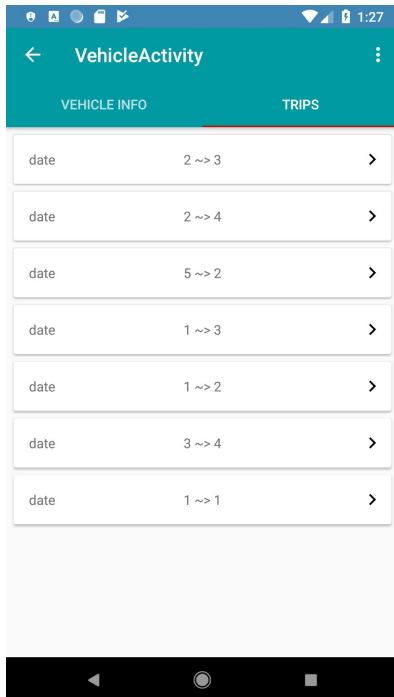


Figura 13 - vista vehicle activity-transacções

Nesta vista é possível visualizar as transacções efectuadas com o veículo ordenadas por data de forma decrescente, mostra a data, 2 quais as duas portagens utilizadas, ao pressionar um das transacções o utilizador é redireccionado para uma vista com um mapa onde apresenta as duas portagens utilizadas.

## Arquitetura interna

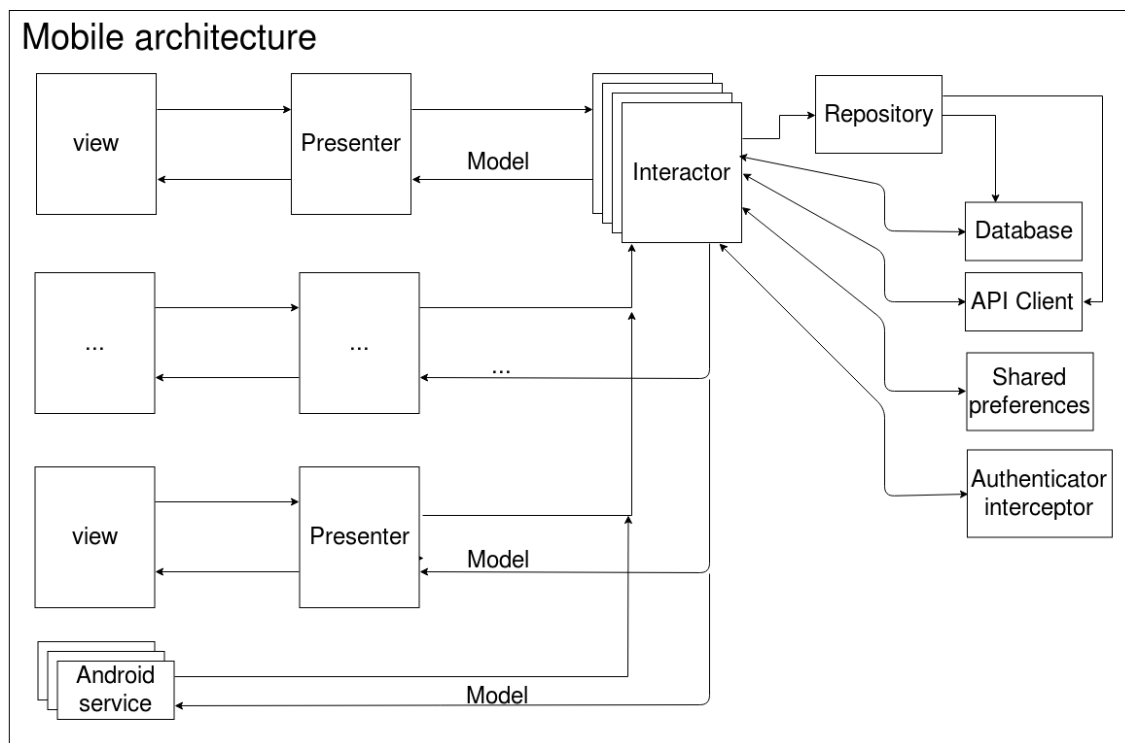


Figura 14 - Diagrama implementação MVP

Neste diagrama podemos observar a organização interna da aplicação que segue o modelo de arquitectura *Model View Presenter* (MVP), que visa desacoplar a vista, *View*, do acesso a dados, *Model*. Segue o princípio de desenho da separação de responsabilidades, facilitando não só o desenvolvimento, como a testabilidade do código. Estes benefícios são possíveis devido a um novo intermediário, *Presenter*, que se encarrega de receber pedidos da *view* e implementar toda a lógica de apresentação relacionada com a mesma, invocando funções existentes na *view*, para que esta altere a interface gráfica adequadamente.

É ainda utilizado uma abstracção sobre os dados denominada de *Interactor*. Um *Interactor* representa um conjunto de *use cases* que são representados por um conjunto de acções sobre os dados que geralmente é comum em vários *presenter*, este está encarregue de aceder a serviços tais como a base de dados ou fazer pedidos a serviços externos por exemplo uma API web, deve ser reutilizado por múltiplos *presenters* ou até serviços do Android que não precisam de lógica de apresentação, desta forma não é repetido código de lógica de acesso a dados, facilitando a sua reusabilidade e crescimento da aplicação.

Visto que existe um nível significativo de dependências entre componentes, foi escolhido introduzir um outro padrão denominado de injeção de dependências, que desacopla os



componentes das suas dependências, tirando partido a biblioteca Dagger disponibilizada pela Google.

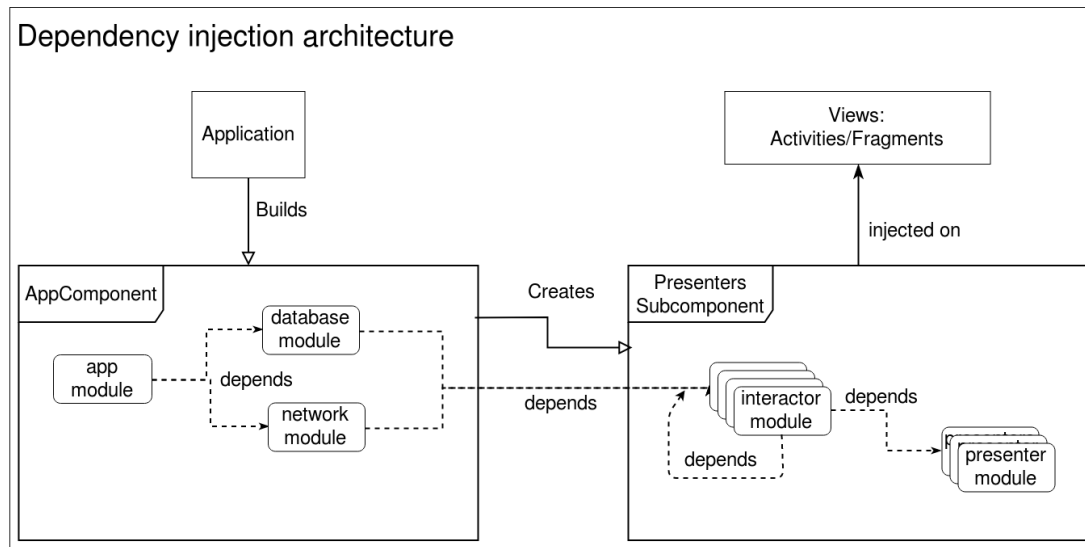


Figura 15 - Diagrama da arquitectura de injeção de dependências

Neste diagrama podemos observar a organização da injeção de dependências, a aplicação constrói o componente principal, *AppComponent*, que contém os módulos principais a injectar, opcionalmente é possível criar um subcomponente, que contém módulos que podem estar sujeitos a ter características específicas, tais como o tempo de vida ser o mesmo de uma *view*. Para utilização foi necessário desenvolver classes *Module* com módulos que têm a lógica de como construir certas dependências, e classes *Componente/Subcomponent*, anotadas com os módulos que expõem e onde estes podem ser injectados. Assim todas as dependências, *depends*, existentes na figura são geridas pelo dagger, sendo que as *views* só precisa anotar a sua dependência de *presenter* com `@Inject` e invocar o *AppComponent* requisitando o subcomponente de *presenters*.

## Arquitetura do *Backoffice*

O *Backoffice* é implementado em *Kotlin*[6] usando a plataforma *SpringBoot*[4] e pode ser dividido nos seguintes componentes.

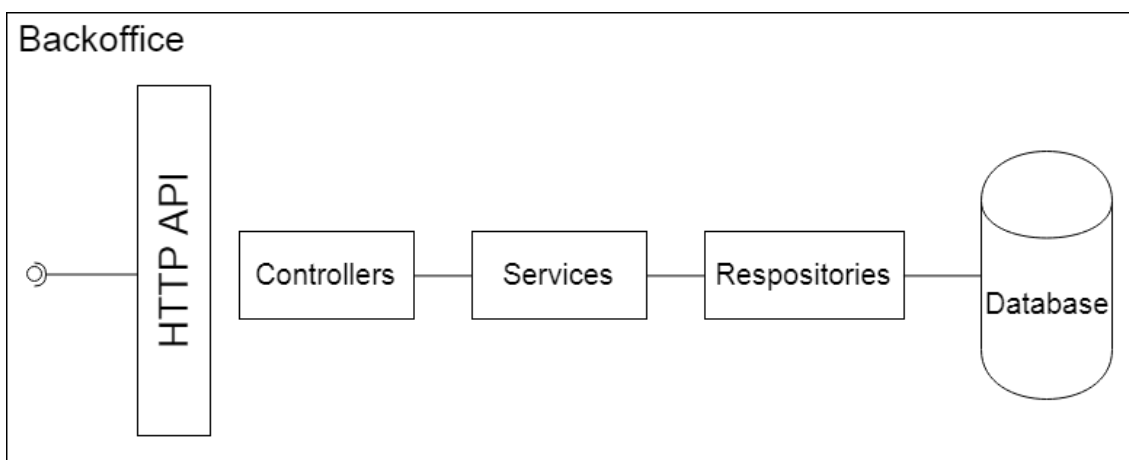


Figura 16 - Arquitectura do *Backoffice*.

## Endpoints

Todos os endpoints necessitam de autenticação e são relativos a um utilizador.

Toll Controller:

- GET `/tolls/{id}` - Representação de uma portagem, recebendo um id como *path parameter*.
- POST `/tolls?number=N` - Representação da colecção de N portagens mais próximas, recebendo como corpo um objecto de geo-localização contendo a latitude e longitude. Caso o *number* seja omitido, será usado o valor por omissão, 100.

Transaction Controller:

- GET `/transactions/{id}` - Representação de uma transação, recebendo um id como *path parameter*.
- GET `/transactions?timestamp=X` - Representação de uma colecção de transações, recebendo como *query string* uma instância temporal que, se presente, será usada como filtro.
- POST `/transactions/open?vehicle=vehicle_id` - Registo de um evento de portagem de entrada, recebendo como *query string* obrigatória o identificador do veículo.
- POST `/transactions/close?vehicle=vehicle_id` - Registo de um evento de portagem de saída, recebendo como *query string* obrigatória o identificador do veículo.

- POST /transactions?vehicle=vehicle\_id - Registo de um evento de portagem de entrada e saída, recebendo como *query string* obrigatória o identificador do veículo.

User Controller:

- GET /users/{id} - Representação de um utilizador, recebendo um id como *path parameter*.

Vehicle Controller:

- GET /vehicles - Representação de uma colecção de veículos.
- GET /vehicles/{id} - Representação de um veículo, recebendo um id como *path parameter*.
- POST /vehicles - Registo de um veículo.

## Negociação de conteúdo

O tipo de conteúdo suportado pelo *Backoffice* é *Javascript Object Notation (JSON)*.

A informação de erros usa o formato “*application/problem+json*” descrito na especificação “[Problem Details for HTTP APIs](#)”.

## Eventos do Clearing

Os eventos que providenciam do *Clearing* são recebidos no *Backoffice* via *HTTP* dos endpoints disponibilizados pela *Web API*.

## Eventos do Backoffice

Os eventos que serão consumidos pela aplicação cliente, serão obtidos através da interface *HTTP* do *Backoffice* fazendo partido do header “*If-Modified-Since*”.

## Autenticação

A autenticação suportada é o esquema básico, descrito na especificação “[HTTP Basic scheme](#)”.

## Camada de dados

Para a implementação da camada de dados foi usado o *Hibernate*, como abstração sobre uma base de dados PostgreSQL[7] .

## Sincronização com o Clearing

Nas funcionalidades de potencialmente longa duração em que é necessário sincronizar com o *Clearing*, dada a natureza *HTTP* do mesmo, é usado o idioma assíncrono para a sua execução.

## Diagrama de Sequência - Ciclo de vida de uma viagem

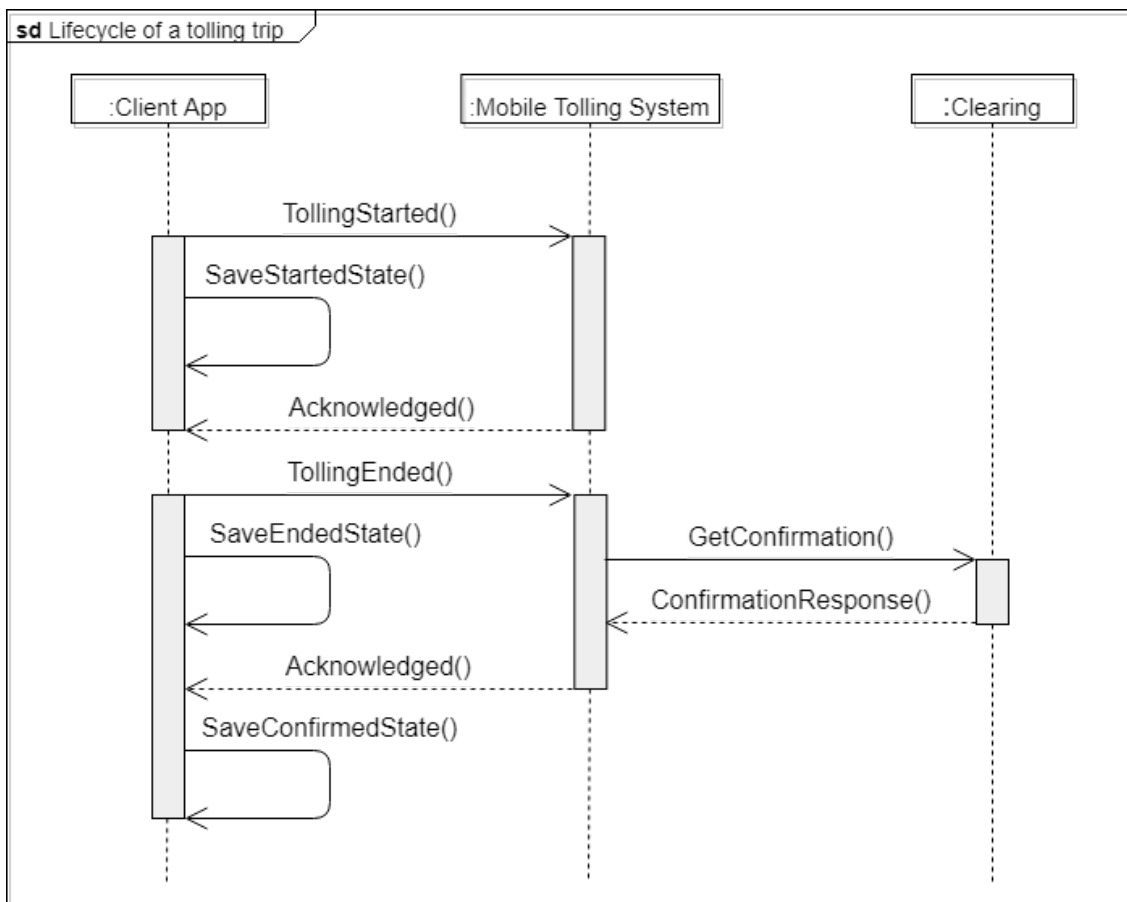


Figura 17 - Diagrama de sequência ciclo de uma viagem .

## Conclusões

Neste projecto investigou-se a viabilidade de usar a tecnologia *GPS* com a finalidade de detectar passagens de portagem.

Foram formulada soluções que assentam nos princípios de boas práticas aprendidos ao longo do curso, como também por iniciativa própria.

Foram estudadas e implementadas várias tecnologias de ponta do mundo *Android*.

# Referências

- [1] - [Via Verde– um exemplo Português de criatividade e inovação ao serviço do Cliente por João Pecegueiro](#), visitado a 12 de Março de 2018
- [2] - [LinktGO](#), visitado a 20 de Abril de 2018
- [3] - [Android Geofences](#), visitado a 12 de Março de 2018
- [4] - [Spring Framework](#), visitado a 12 de Março de 2018
- [5] - [IntelliJ IDE](#), visitado a 12 de Março de 2018
- [6] - [Kotlin](#), visitado a 12 de Março de 2018
- [7] - [PostgreSQL](#), visitado a 12 de Março de 2018
- [8] - [Android Maps Android API](#), visitado a 12 de Março de 2018

## **A.1 Diagramas da Aplicação**

## A.2 Modelos de dados

