



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2019/2020

SINCROmobile - Subscrição de eventos de trânsito

Rafael Nobre

Orientadores Luís Osório
 Paulo Borges

Relatório final realizado no âmbito de Projecto e Seminário,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2019/2020

Setembro de 2020

Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

SINCROmobile - Subscrição de eventos de trânsito

39267 Rafael Nobre

Orientadores: Luís Osório
Paulo Borges

Relatório de progresso realizado no âmbito de Projecto e Seminário,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2019/2020

Setembro de 2020

Resumo

Este documento descreve o projeto **SINCROmobile** – Subscrição de Eventos de Trânsito, daqui em diante designado por **SINCROmobile**.

O projeto **SINCROmobile** visa o desenvolvimento de uma solução *mobile* para complementar o sistema **SINCRO** - Sistema Nacional de Controlo de Velocidade, desenvolvido numa parceria de investigação entre a **ANSR** – Associação Nacional de Segurança Rodoviária, e o **ISEL** – Instituto Superior de Engenharia de Lisboa, no âmbito da disciplina Projeto e Seminário do curso Licenciatura em Engenharia Informática e de Computadores oferecido pelo **ISEL**.

O sistema a desenvolver pretende facilitar ao cidadão o acesso e tratamento dos dados referentes às infrações rodoviárias cometidas nos vários pontos de controlo do sistema **SINCRO** espalhados pelo país.

Índice

RESUMO	V
1. INTRODUÇÃO	7
1.1 ORGANIZAÇÃO DO RELATÓRIO	7
1.2 ESPAÇO NO MERCADO PARA A APLICAÇÃO SINCROAPP	7
1.3 QUADRO TECNOLÓGICO.....	8
2. FORMULAÇÃO DO PROBLEMA	9
2.1 REQUISITOS FUNCIONAIS	9
2.1.1 <i>Eventos de Trânsito</i>	10
2.1.2 <i>Consulta de informação</i>	12
2.1.3 <i>Definições</i>	13
2.1.4 <i>Administração</i>	14
2.2 MODELO ENTIDADE-ASSOCIAÇÃO	15
3. PROPOSTA DE SOLUÇÃO	17
3.1 DATA STORAGE - SINCRODB	17
3.2 CORE	19
3.2.1 <i>SINCROserver</i>	19
3.3 APP	21
3.3.1 <i>SINCROapp</i>	21
4. CONTRATO WEB APLICAÇÃO MÓVEL/SERVIDOR	35
5. CONCLUSÕES.....	37
REFERÊNCIAS	38

1. Introdução

Depois de gerada uma infração de trânsito por deteção de um veículo em excesso de velocidade, o proprietário do veículo só é notificado após um período que pode ir até dois anos, após o qual a infração prescreve [1]. Adicionalmente, o condutor não consegue obter facilmente acesso às infrações já cometidas.

Estes dois fatores contribuem para a dessensibilização do condutor sobre os perigos ao qual se sujeita a si e aos restantes condutores na estrada, fazendo com que a notificação e a multa associada sejam em muitos casos dissociadas da transgressão cometida, aumentando o risco de acidentes graves e/ou mortes na estrada.

O sistema a desenvolver neste projeto pretende que o condutor consiga ter *feedback* no mais curto espaço de tempo ao cometer uma infração e consiga também consultar informação sobre multas que já tenha cometido, salientando assim a importância de uma condução mais cuidada, bem como facilitar o pagamento da multa através de pagamento eletrónico pela aplicação.

1.1 Organização do relatório

Este relatório está organizado da seguinte forma:

- Formulação do Problema – discussão funcional sobre os requisitos do projeto.
- Proposta de Solução – Detalhe técnico sobre a solução desenvolvida para satisfazer os requisitos descritos no ponto anterior. Este capítulo está dividido em três partes, cada uma tratando de um dos componentes que compõe a solução.

1.2 Espaço no mercado para a aplicação SINCROapp

Existem atualmente várias aplicações no mercado que lidam com radares de trânsito, como as aplicações **RADARES de Portugal** [2] e **WAZE** [3], que têm como funcionalidade chave a comunicação entre utilizadores sobre identificação de radares, eventos de trânsito como filas de espera e operação de trânsito. Existem algumas aplicações que incluem também funcionalidades de recomendação de pontos de interesse, como no caso da aplicação **ViaMichelin GPS** [4].

Neste sentido, a aplicação **SINCROapp** não pretende substituir estas aplicações, mas sim complementá-las para utilizadores em Portugal Continental, dando informação sobre infrações cometidas nos radares da rede SINCRO e facilitando o pagamento das mesmas através da aplicação, contemplando o uso de um veículo por um condutor que não seja proprietário do mesmo.

1.3 Quadro tecnológico

De seguida são destacadas as tecnologias e ferramentas usadas neste projeto:

- Java – Linguagem de programação usada no desenvolvimento da aplicação servidor **SINCROserver** e aplicação móvel **SINCROapp**.
- MySQL – Serviço de manutenção de bases de dados relacionais utilizado para a criação/manutenção da base de dados **SINCROdb**.
- Spring Boot – *Framework* para a linguagem Java que facilita a criação de aplicações Web
- Retrofit – Framework Java para a plataforma android que simplifica o processo de comunicação em rede
- IntelliJ IDEA – IDE (Integrated Development Environment) usado na criação e desenvolvimento da aplicação servidor **SINCROserver**.
- Android Studio – IDE usado na criação e desenvolvimento da aplicação móvel **SINCROapp**.
- MagicDraw – Ferramenta de modelação de sistemas usada para descrever o projeto de forma funcional
- SysML – Linguagem de modelação de sistemas usada no desenvolvimento dos modelos que descrevem o projeto

2. Formulação do Problema

Para o desenvolvimento da aplicação *mobile* em foco neste projeto, daqui em diante designada por **SINCROapp**, é necessário simular o sistema SINCRO mantido pela ANSR.

Para tal será necessária uma aplicação servidora, **SINCROserver**, que forneça todos os dados e funcionalidades que permitam o funcionamento da aplicação móvel, e uma base de dados que represente o modelo de negócio, **SINCROdb**.

A aplicação pode funcionar com qualquer sistema *backend* que use o modelo de dados definido para este projeto, bem como os *endpoints* definidos no capítulo Contrato Web Aplicação Móvel/Servidor.

Nas secções seguintes é descrito em detalhe o universo do problema, oferecendo uma visão mais clara dos objetivos que o sistema pretende atingir.

2.1 Requisitos Funcionais

Esta secção aborda a análise e formalização dos requisitos do sistema **SINCROmobile**, especificando as funcionalidades que o sistema deve oferecer aos seus cidadãos e conceptualizando as mesmas em um modelo lógico de dados que as suporte, sem o tornar dependente de nenhuma tecnologia ou tecnologias em particular.

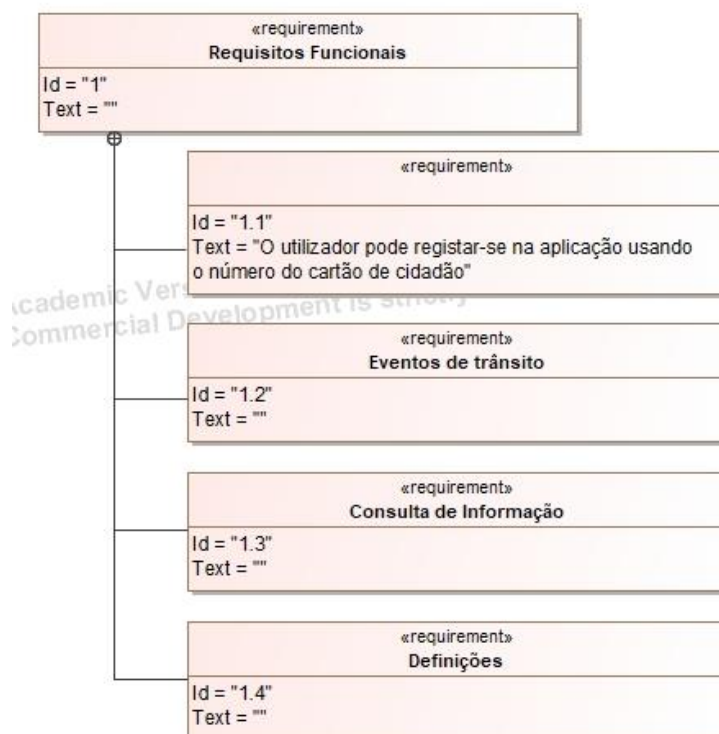
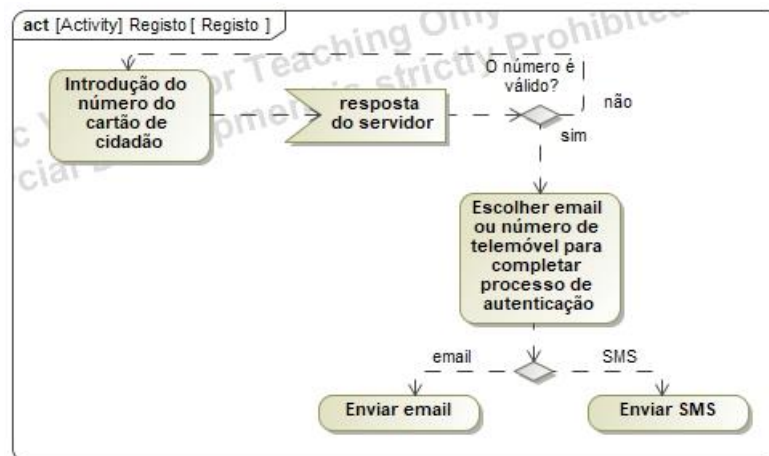


Figura 1 - Requisitos Funcionais - Visão Geral

A **Figura 2** define como requisito que o cidadão se possa registar no sistema **SINCROmobile** inserindo o seu número do cartão de cidadão na aplicação, o que pretende simular a capacidade do sistema **SINCRO** de obter a informação relevante do cidadão através deste identificador, como mostra a Figura 1.

**Figura 2 - Registo no sistema SINCROmobile**

De seguida é pedido ao cidadão que confirme a sua identidade usando o número de telemóvel obtido pelo servidor ou, opcionalmente, um email, caso este tenha sido obtido junto com o resto da informação associada ao cidadão.

2.1.1 Eventos de Trânsito

De acordo com os requisitos formulados na **Figura 1**, existe um conjunto de requisitos designados por **Eventos de Trânsito**, que são detalhados na **Figura 3**.

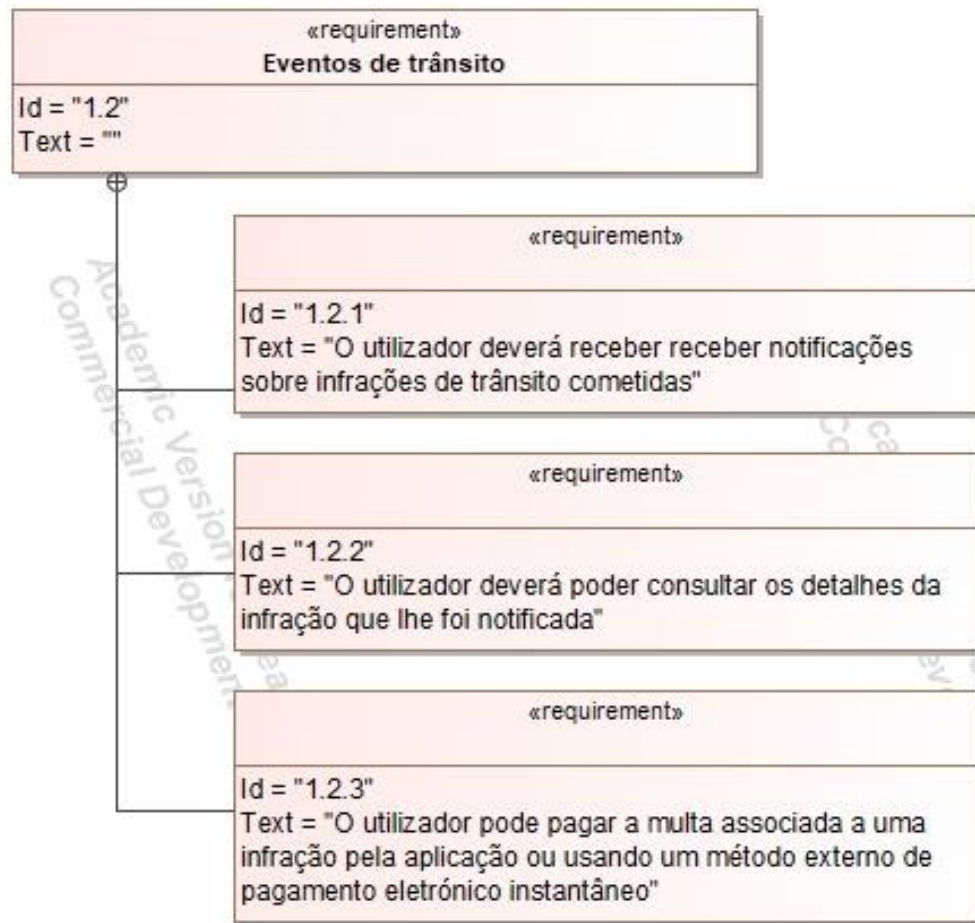


Figura 2 - Requisitos Funcionais - Eventos de Trânsito

A **Figura 3** representa a funcionalidade principal da aplicação, mostrando o fluxo desde a receção de uma notificação em “tempo real” no telemóvel sobre uma infração detetada pelo sistema, passando pela consulta dos detalhes da infração, tais como tipo, preço da multa associada, etc., e terminando com a possibilidade de realizar o pagamento da multa pelo telemóvel.

2.1.2 Consulta de informação

Os requisitos de consulta de informação são apresentadas na **Figura 4**.

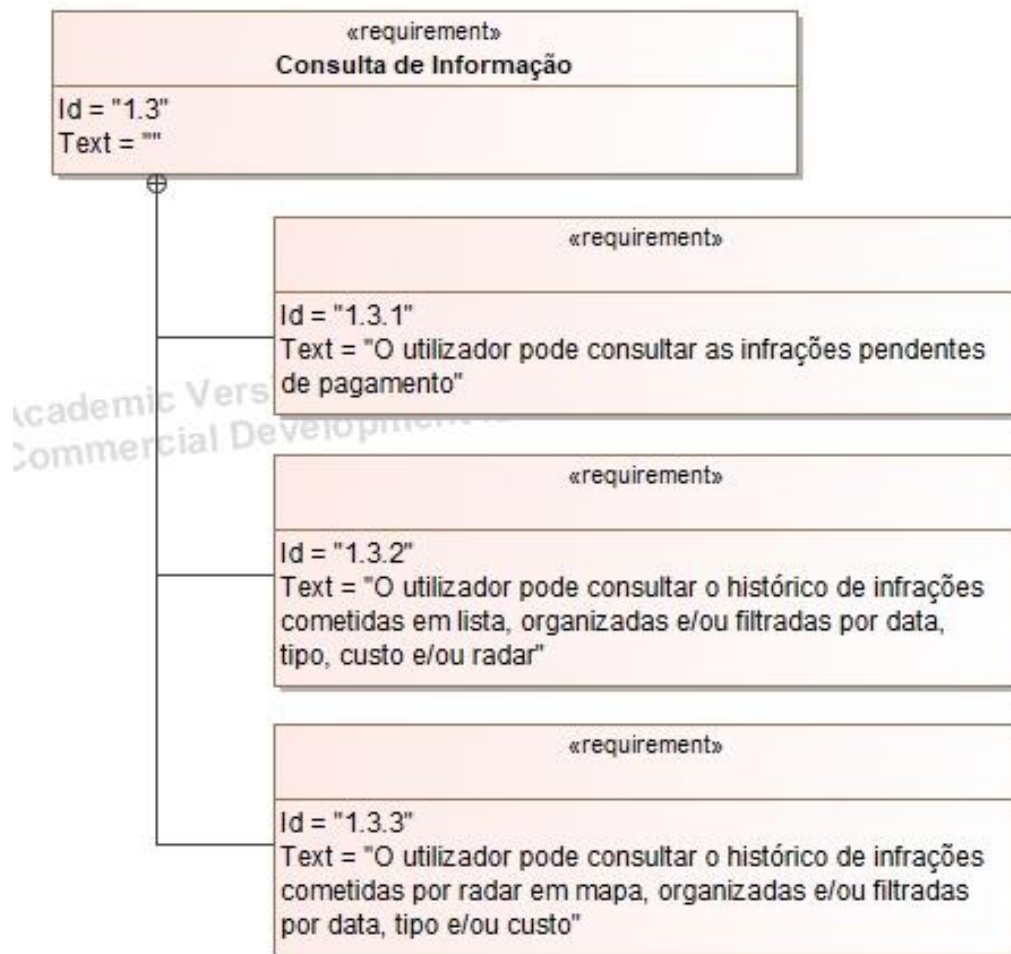


Figura 3 - Requisitos Funcionais - Eventos de Trânsito

Como se apresenta na **Figura 4**, o cidadão deverá, para além da consulta do detalhe de uma qualquer infração associada, deverá poder ver de forma fácil todas as infrações pendentes de pagamento, bem como poder consultar todas as infrações associadas em formato de lista ou apresentadas num mapa, podendo adicionalmente filtrar/ordenar as infrações apresentadas segundo os filtros apresentados em cada uma das funcionalidades.

2.1.3 Definições

Aqui são apresentadas as opções de customização disponibilizadas ao cidadão, detalhadas na **Figura 5**.

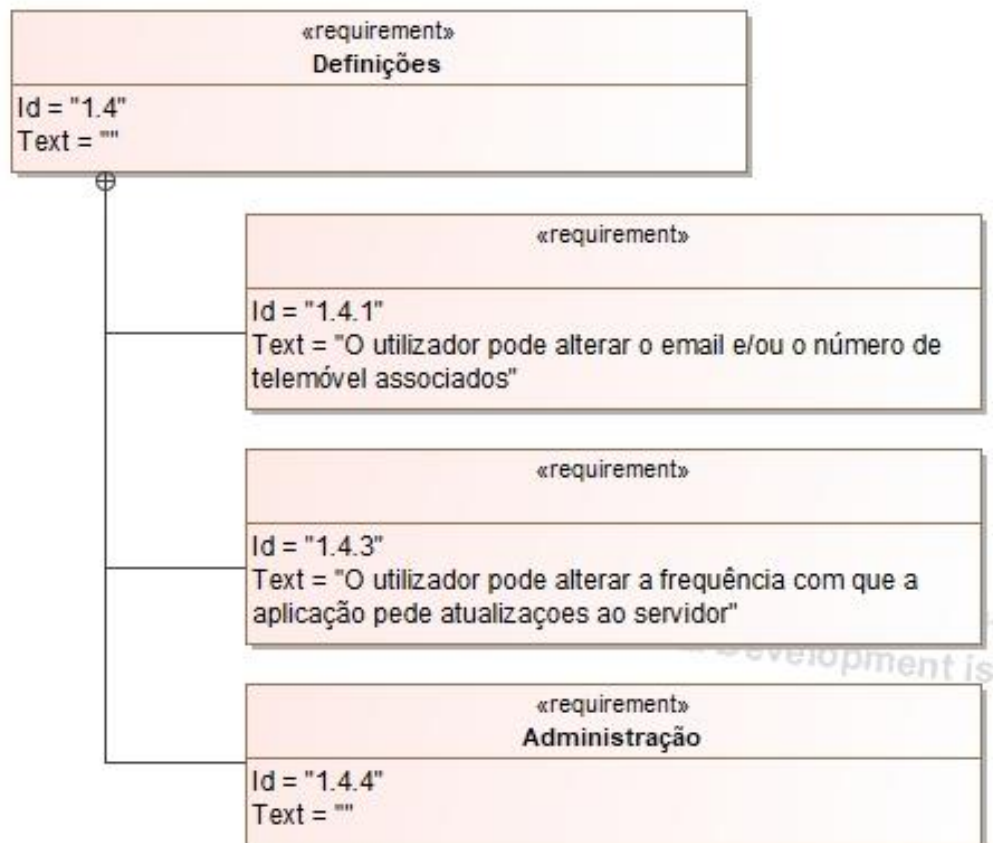


Figura 4 - Requisitos Funcionais – Definições

Para além das opções descritas na **Figura 5**, são também apresentadas opções de administração, detalhadas em seguida.

2.1.4 Administração

As opções de administração tratam da associação entre o cidadão e um ou mais veículos, conforme apresentadas na **Figura 6**.

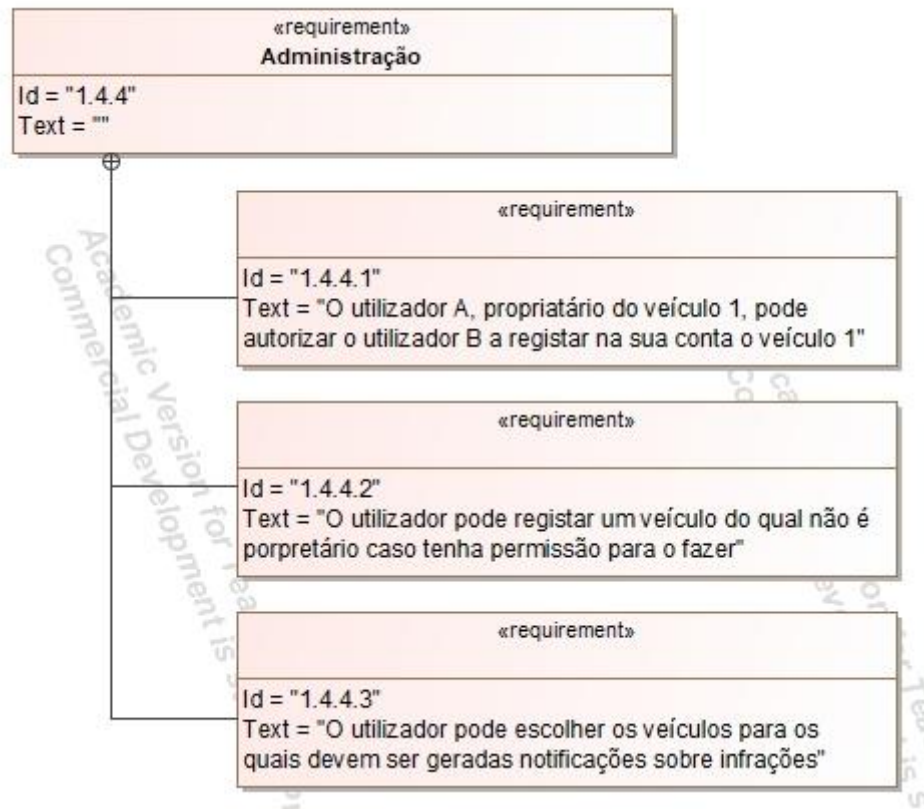


Figura 5 - Requisitos Funcionais – Administração

Aqui introduzem-se as especializações que o cidadão pode tomar dentro do sistema **SINCROmobile**:

- Proprietário – Designa a relação de propriedade entre o cidadão e um veículo, isto é, o cidadão **A** é proprietário do veículo **1**
- Condutor – Designa a relação de posse entre o cidadão e um veículo, isto é, o cidadão **B** conduz o veículo **1**

Para os cidadãos **A** e **B**, em que o cidadão **A** é proprietário do veículo **1**, o cidadão **A** pode autorizar o cidadão **B** a associar-se ao veículo **1**, passando a poder usar todas as funcionalidades oferecidas pela aplicação no contexto de uma relação cidadão – veículo, exceto autorizar outros cidadãos a usar o veículo.

2.2 Modelo Entidade-Associação

No modelo apresentado na **Figura 7** estão representadas as entidades que suportam o sistema SINCRO e as suas relações.

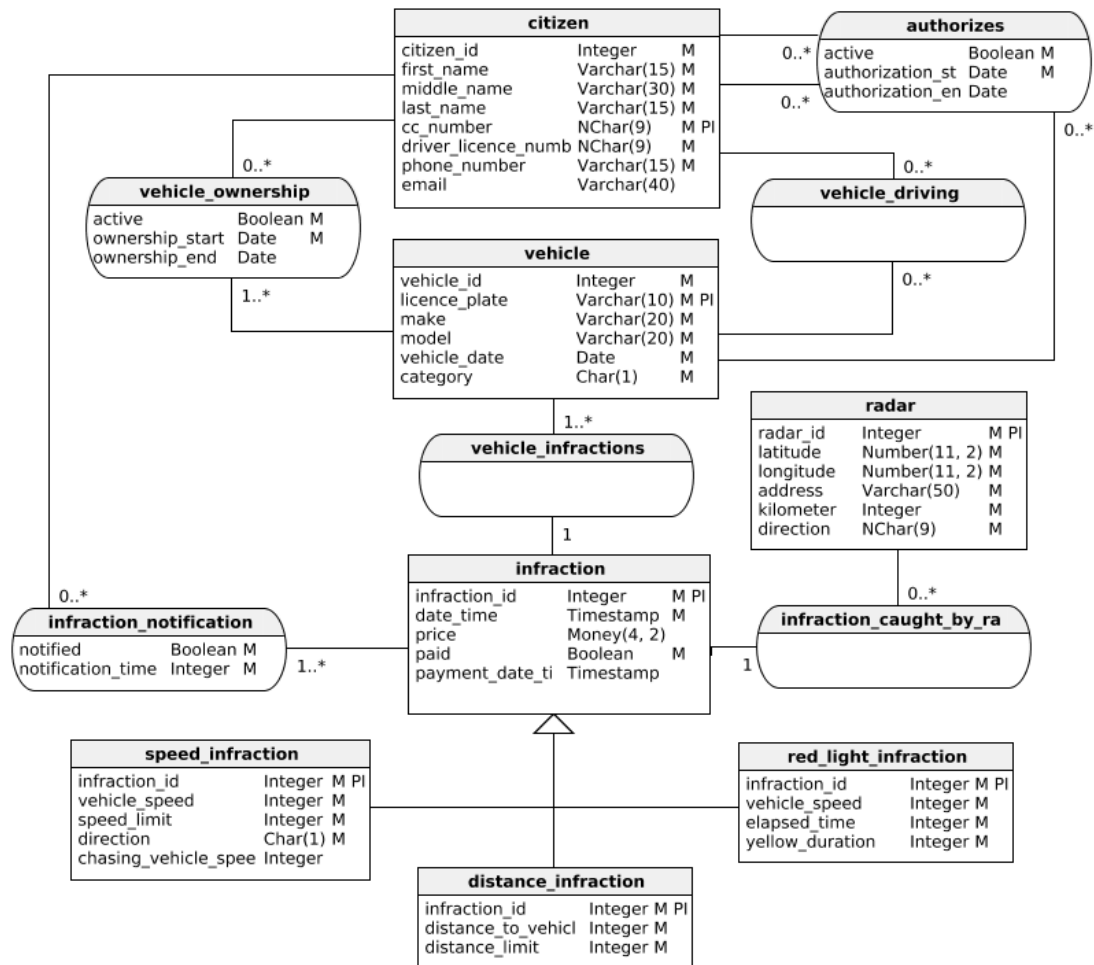


Figura 6 - Modelo Entidade-Relação

3. Proposta de Solução

A solução concebida é constituída por três elementos:

- **Data Storage** – elemento responsável pela persistência de dados; concretizado pela base de dados **SINCROdb**
- **Core** – elemento responsável por toda a lógica de negócio e pela comunicação entre elemento que interage com o utilizador e o elemento que persiste os dados; concretizado pelo servidor **REST SINCROserver**
- **App** – elemento responsável pela interação com o utilizador; concretizado pela aplicação para *Android* **SINCROapp**

A **Figura 8** apresenta uma visão geral da solução.

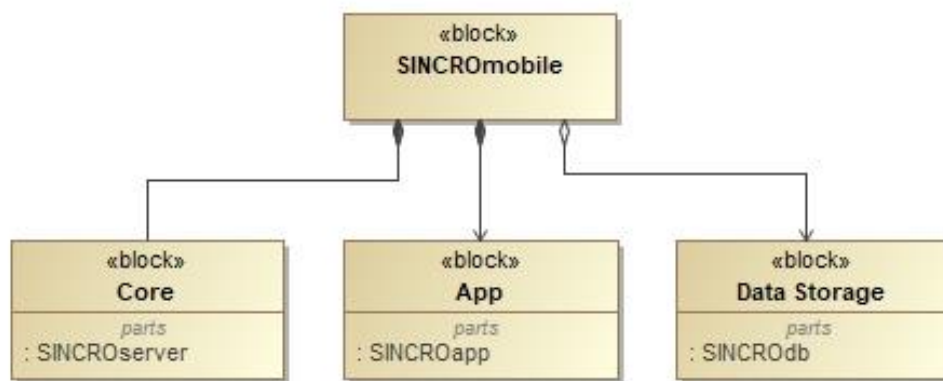


Figura 7 - Definição Global

O sistema **SINCROmobile** foi desenhado de modo a permitir que sejam adicionadas implementações distintas dos três componentes representados na **Figura 8** que o compõem, desde que sigam o modelo de dados representado na **Figura 8**.

Nas secções seguintes são discutidos com maior detalhe as implementações dos componentes da solução proposta.

3.1 Data Storage - SINCROdb

O elemento de persistência de dados é concretizado em uma base de dados relacional implementada em **MySQL**, **SINCROdb**, cujo modelo relacional inferido da base de dados é apresentado na **Figura 9**.

Foi escolhido o motor de base de dados **MySQL** pelos seguintes motivos:

- Maturidade - contando com uma comunidade alargada de desenvolvedores e administradores de base de dados que contribuem para o seu melhoramento contínuo, **MySQL** é um motor de base de dados confiável
- Versatilidade – sendo possível instalar **MySQL** em diferentes sistemas operativos, é uma opção mais aliciante comparando com motores de base de dados relacionais semelhantes como **SQL Server** da Microsoft

A escolha de **MySQL** pela **ANSR** para a implementação do sistema que o **SINCROmobile** pretende representar é um fator favorável a esta decisão.

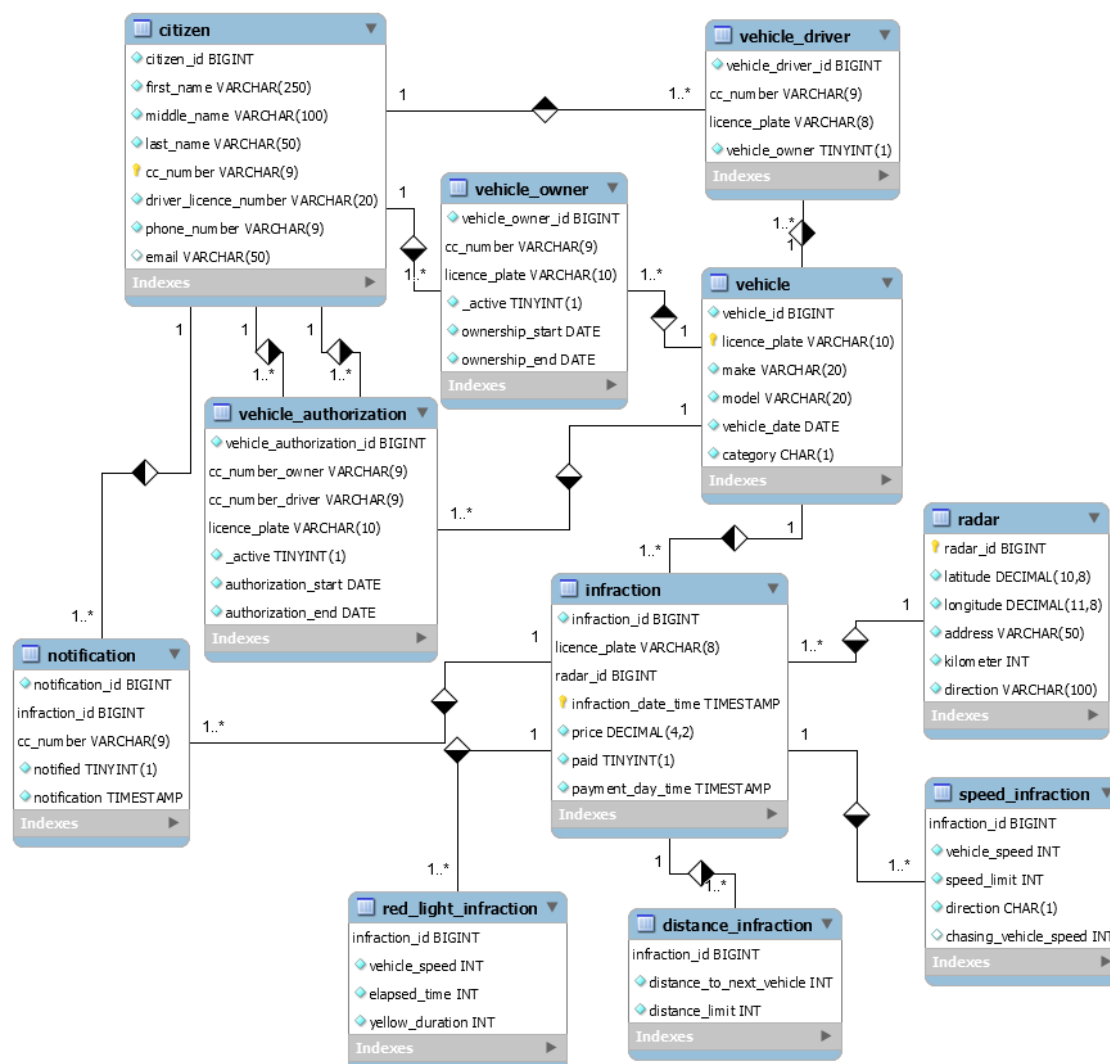


Figura 9 - Modelo relacional

A **Figura 10** descreve o modelo relacional derivado do modelo entidade-relação representado na **Figura 8**.

3.2 Core

Dentro do sistema **SINCROmobile**, o **Core** é responsável por obter os dados existentes na camada de persistência de dados, por os tratar e fornecer a aplicações cliente, e por autenticar e autorizar utilizadores do sistema.

A **Figura 10** mostra a arquitetura do elemento **Core**:

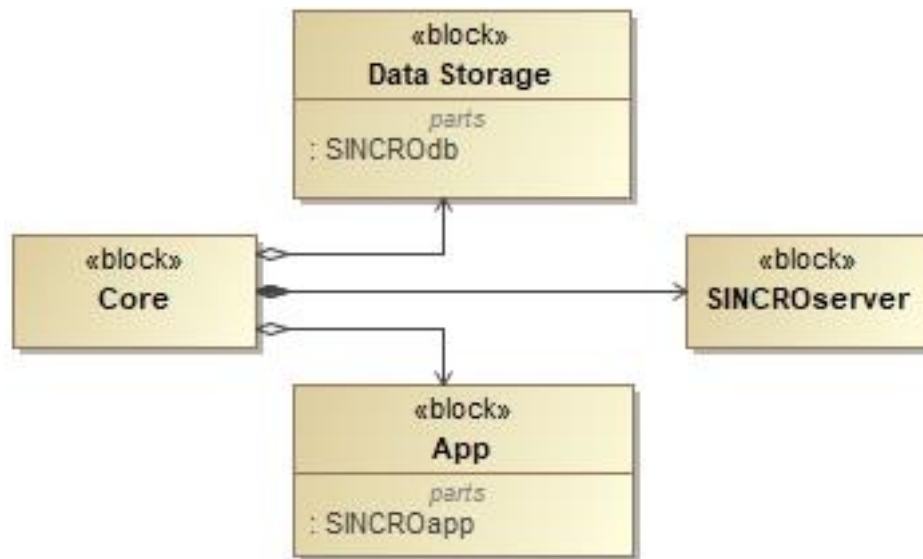


Figura 10 - Arquitetura Core

3.2.1 - SINCROserver

O elemento **Core** é concretizado em uma aplicação servidora que faz uso do padrão de desenho **REST**, **SINCROserver**, implementada em *Java* com suporte da framework *Spring Boot*.

REST (Representational state transfer) é um padrão de desenho criado por *Roy Fielding* [5] na sua tese de doutoramento. Para uma **API** ser considerado **RESTful**, deve apresentar as seguintes funcionalidades:

- Interface uniforme – Os recursos aos quais os clientes podem aceder devem estar bem definidos, ser únicos e uniformes
- Cliente-Servidor – Deve ser possível desenvolver os dois de forma separada, sendo o único ponto de contacto a interface definida pelo servidor
- *Stateless* – A comunicação entre cliente e servidor não deve depender do servidor manter estado sobre pedidos anteriores feitos pelo cliente para satisfazer novos pedidos; Como é o caso da maioria das **API** correntes, o cliente deve enviar toda a informação necessária de

modo a que o servidor consiga resolver um pedido, como por exemplo tokens de autorização para *Single Sign-on*

- *Cacheable* – Uma ou mais partes do sistema devem implementar um mecanismo de cache onde possível, de modo a reduzir carga no servidor e tempo de espera para o utilizador da aplicação cliente
- Camadas – O sistema deve ser implementado por camadas independentes, isto é, deve ser possível instanciar a *API* no servidor A e a base de dados no servidor B, bem como ter várias instâncias dos mesmos, o que permite maior escalabilidade do sistema

O sistema **SINCROmobile** apresenta todas estas características, em particular o servidor **SINCROserver**, pelo que é considerado *RESTful*.

A **Figura 11** representa as relações entre os vários componentes do servidor **SINCROserver**.

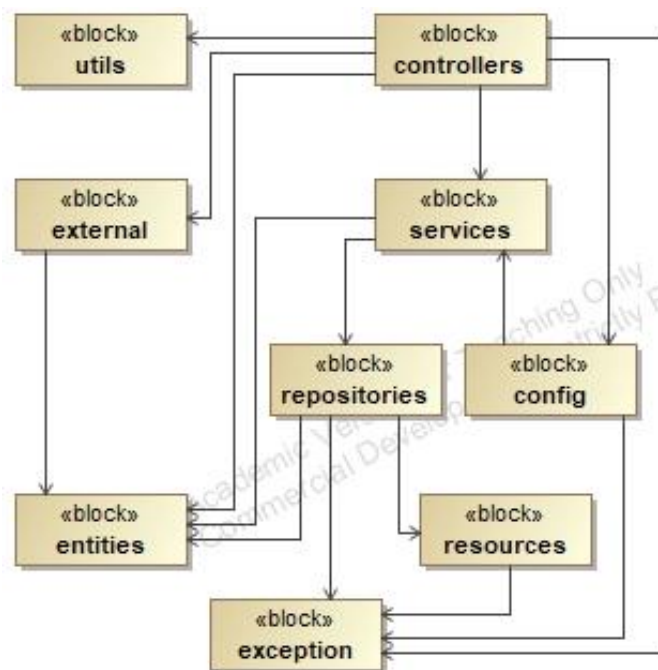


Figura 11 - Arquitetura SINCROserver

Segue uma breve descrição dos vários packages representados na **Figura 10**:

- **Controllers** – contêm os vários controladores responsáveis pela comunicação com os clientes do servidor
- **Services** – contêm os serviços responsáveis pela lógica de negócio
- **Repositories** – contêm os repositórios responsáveis pela comunicação com os sistemas de persistência de dados

- Resources – responsável pela obtenção de recursos, como por exemplo *queries* alojadas em ficheiros para executar em base de dados
- Config – elemento responsável pela configuração do servidor, nomeadamente do mecanismo de autenticação
- Utils – contém classe com métodos utilitários
- External – responsável pela obtenção de informação do cidadão na fase de registo na aplicação
- Entities – contém todas as entidades que compõe o modelo de dados do sistema **SINCROmobile**
- Exception – contém uma classe para representar erros dentro do servidor

3.3 App

Este elemento é responsável por fornecer ao diretamente ao condutor as funcionalidades que o sistema **SINCROmobile** disponibiliza, nomeadamente a notificação de novas infrações e a consulta de infrações anteriormente cometidas.

A **Figura 12** apresenta a arquitetura do elemento **App**:

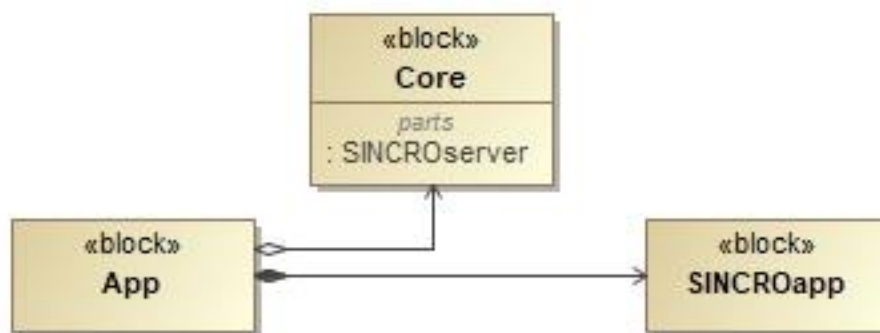


Figura 12 - Arquitetura App

3.3.1 SINCROapp

A aplicação mobile **SINCROapp** está organizada de acordo com os seguintes packages:

- ui – contem todas as atividades e fragmentos da aplicação
- data.model – contem os elementos que comunicam a necessidade de dados e a sua disponibilidade entre as atividades e os serviços.

- services – contêm os serviços responsáveis pela comunicação com o core dos sistema e por funcionalidades como notificar o condutor sobre novas infrações
- persistence – contêm as classes responsáveis pela persistência de dados de modo a reduzir os pedidos feitos pelos serviços
- data.entites – contêm as entidades definidas para o sistema *SINCROmobile*

A **Figura 13** ilustra a arquitetura descrita acima.

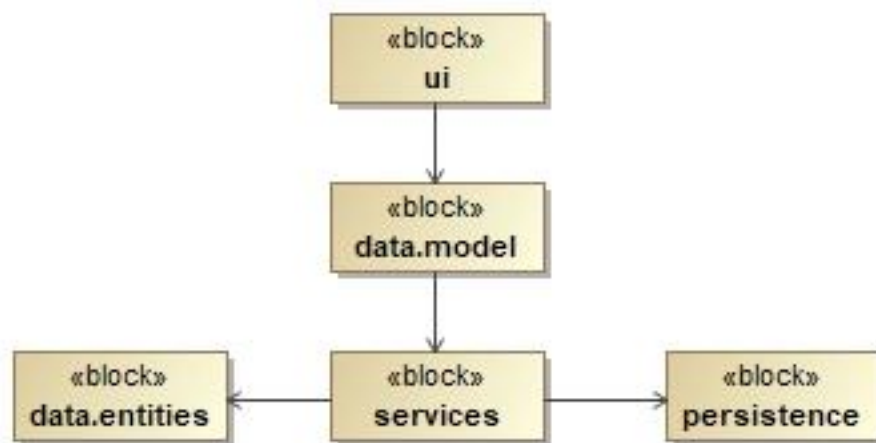


Figura 13 - Arquitetura SINCROapp

3.3.1.1 Funcionalidades

A seguir serão apresentadas as funcionalidade implementadas na aplicação *SINCROapp*.

3.3.1.1.1 Registo de um condutor

No ato do registo o condutor começa por introduzir o número do seu cartão de cidadão para os seus dados poderem ser obtidos, conforma a **Figura 14**.

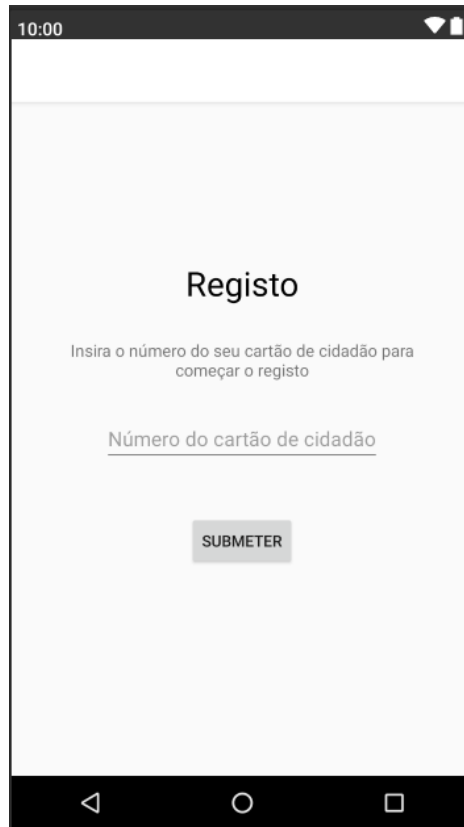
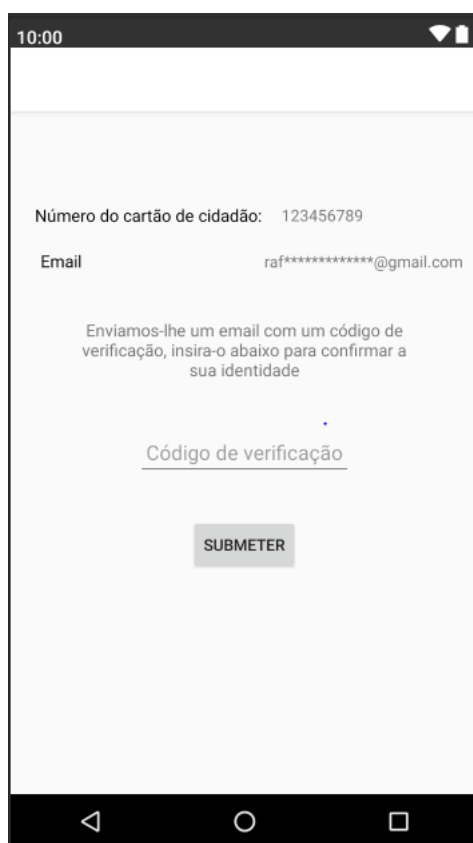
The image shows a mobile application interface for registration. At the top, the status bar shows the time 10:00 and battery level. The app's title bar is dark. The main content area has a light gray background. The title 'Registo' is centered in a bold font. Below it, a subtitle reads 'Insira o número do seu cartão de cidadão para começar o registo'. A text input field is centered, with the placeholder text 'Número do cartão de cidadão'. Below the input field is a button labeled 'SUBMITER'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Figura 14 - Ecrã de input do número do cartão de cidadão

De seguida é apresentado ao utilizador o seu número do cartão de cidadão e alguns caracteres do email associado, e é pedido que insira o código de verificação que foi enviado para o email mostrado, conforme a **Figura 15**.



10:00

Número do cartão de cidadão: 123456789

Email raf*****@gmail.com

Enviamos-lhe um email com um código de verificação, insira-o abaixo para confirmar a sua identidade

Código de verificação

SUBMETER

Figura 15 – Ecrã de input do código de verificação

Por último, são mostrados ao condutor os seus dados completos, e é pedido que registe um nome de utilizador e uma palavra-chave para completar o registo, conforme a **Figura 16**.

10:00

Nome Completo Rafael Gomes Nobre

Número do cartão de cidadão 123456789

Número da carta de condução 987654321

Email rafael.gomes.nobre@gmail.com

Número de telemóvel 933417044

Crie o seu nome de utilizador e palavra-passe para completar o registo

Nome de utilizador

Palavra-passe

SUBMETER

Figura 16 - Ecrã de criação de nome de utilizador e palavra-passe

3.3.1.1.2 Login

Neste ecrã é pedido ao utilizador que insira o seu nome de utilizador e palavra-passe para poder fazer login ou pode escolher registar-se, conforme a **Figura 17**.



Figura 17 - Ecrã de Login

3.3.1.1.3 Ecrã principal

Neste ecrã são apresentadas todas as infrações associadas ao condutor, divididas em duas categorias: infrações pagas e por pagar, conforme a **Figura 18**.

Inclui também um menu para navegar pelo resto da aplicação, conforme a **Figura 19**, bem como a opção de ordenar as listas de infrações conforme a **Figura 20**.

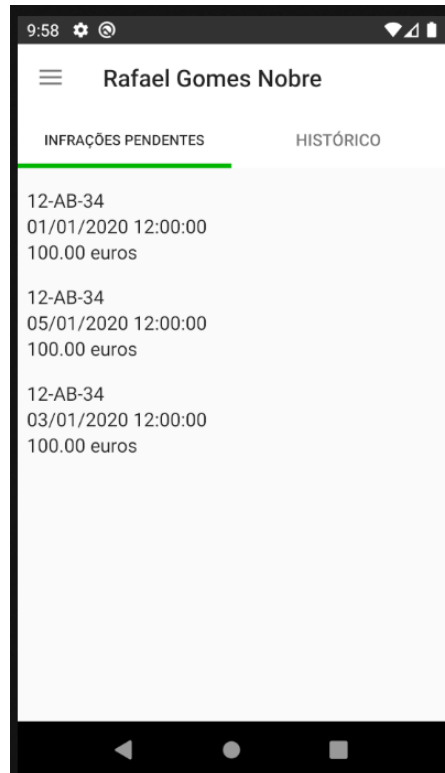


Figura 18 - Ecrã principal



Figura 19 – Menu

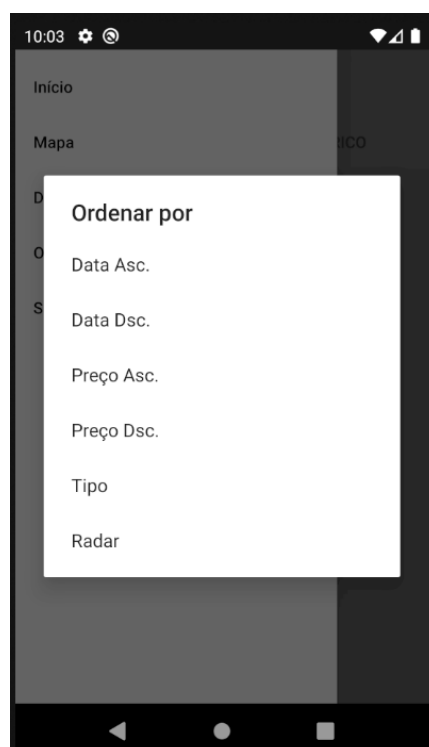


Figura 20 - Opções de ordenação de infrações

3.3.1.1.4 Ecrã de detalhe de uma infração

Neste ecrã são apresentados todos os detalhes da infração selecionada, que varia conforme o tipo de infração:

- Infração de velocidade – **Figura 21**
- Infração de sinal vermelho – **Figura 22**
- Infração de distância – **Figura 23**



Figura 21 - Infração de velocidade



Figura 22 - Infração de sinal vermelho



Figura 23 - Infração de distância mínima

3.3.1.1.5 Ecrã de pagamento

Conforme exemplificado na **Figura 21**, as infrações apresentadas no ecrã de detalhe que estão por pagar apresentam uma opção para o fazer, que leva o condutor para o ecrã de pagamento apresentado na **Figura 24**. Caso o pagamento seja bem sucedido o condutor é levado de volta para o ecrã da **Figura 18**.

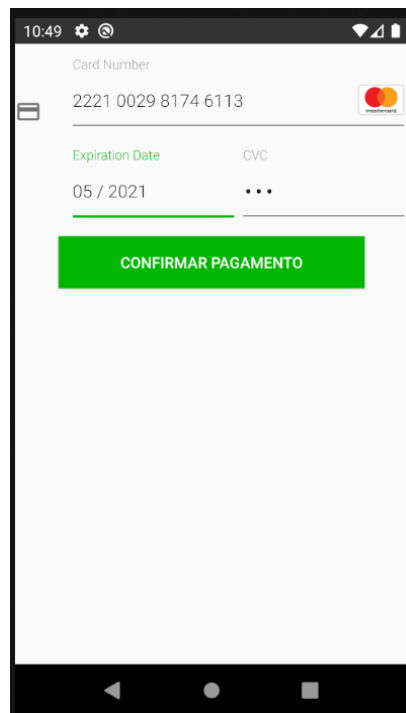


Figura 24 - Ecrã de pagamento

3.3.1.1.6 Definições

A **Figura 25** mostra o ecrã onde são apresentadas ao condutor as opções apresentadas na **Figura 5**. A **Figura 26** apresenta ao condutor uma lista dos veículos que lhe são visíveis. A **Figura 27** apresenta o detalhe de um veículo selecionada na lista do ecrã apresentado na **Figura 26**, bem como um botão que lança uma caixa de texto onde o condutor deverá inserir o número do cartão de cidadão de um condutor que pretenda autorizar.

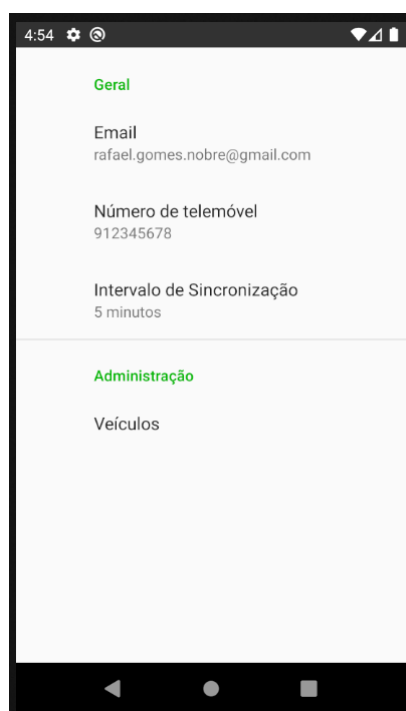


Figura 2610 - Definições

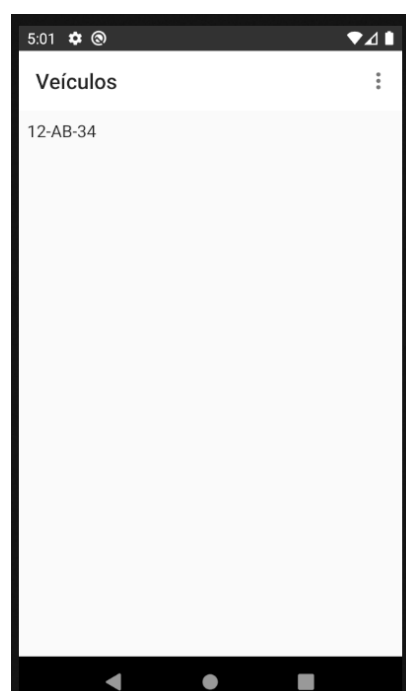


Figura 26 - Lista de veículos

3.3.1.1.7 Ecrã *Google Maps*

Neste ecrã, para o qual se pode navegar através da opção “Mapa” apresentada na **Figura 19**, são apresentadas as localizações dos radares como marcadores no mapa conforme a **Figura 27**.

Para cada marcador é apresentado, ao pressionar no ícone, um *popup* com a sua localização, conforme a **Figura 28**. Ao pressionar no *popup*, é apresentado o mesmo ecrã da **Figura 18** apenas com as infrações associadas ao radar em questão.

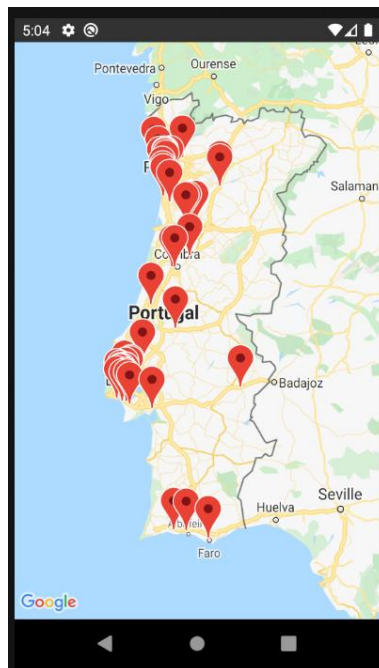


Figura 2711 - Mapa



Figura 28- Marcador selecionado

3.3.1.2 Notificações

Quando ocorre uma nova infração, o condutor recebe uma notificação no dispositivo móvel que o leva para o ecrã da **Figura 21**.

3.3.1.2 Funcionalidades não implementadas

Os pontos **1.4.4.2** e **1.4.4.3** da **Figura 26** não estão implementados.

4. Contrato Web Aplicação Móvel/Servidor

Em seguida são apresentados todos os *endpoints* que o servidor **SINCROserver** implementa.

Todos os *endpoints* têm como *input* e *output* objectos do tipo JSON – Javascript Object Notation.

As entidades usadas nos pedidos representam as entidades apresentadas na **Figura 10**.

Todos os pedidos com prefixo “sincroapp” necessitam de um *header* “Authorization” cuja valor é a *String* “token” devolvida no pedido @POST(“authenticate/login”).

- @POST(“register/beginRegistration”) – recebe uma *String* “cc_number” com o número de CC, devolve um objeto *Citizen*
- @POST(“register/registerWithEmail”) – recebe um objeto *Citizen* “citizen”, não devolve parâmetros
- @POST(“register/verifyRegistrationCode”) - recebe um objeto *Citizen* “citizen” e uma *String* “inputCode”, não devolve parâmetros
- @POST(“register/registerLoginAndPassword”) – recebe um objeto *User* “user”, não devolve parâmetros
- @POST(“authenticate/login”) – recebe as *Strings* “username” e “password”, devolve um objeto *Citizen* “citizen”, um objeto *User* “user” e uma *String* “token”
- @POST(“sincroapp/getInfractions”) - recebe um objeto *Citizen* “citizen”, devolve três listas: lista de objetos *SpeedInfraction* “speedInfractions”, lista de objetos *RedLightInfraction* “redLightInfractions” e lista de objetos *DistanceInfraction* “distanceInfractions”
- @POST(“sincroapp/getRadarInfractions”) – recebe um objeto *Citizen* “citizen”, devolve três listas: lista de objetos *SpeedInfraction* “speedInfractions”, lista de objetos *RedLightInfraction* “redLightInfractions” e lista de objetos *DistanceInfraction* “distanceInfractions”
- @POST(“sincroapp/getRadars”) – não recebe parâmetros, devolve uma lista de objetos *Radar* “radars”
- @POST(“sincroapp/getNewInfractions”) – recebe um objeto *Citizen* “citizen”, devolve três listas: lista de objetos *SpeedInfraction* “speedInfractions”, lista de objetos *RedLightInfraction* “redLightInfractions” e lista de objetos *DistanceInfraction* “distanceInfractions”
- @POST(“sincroapp/getRadar”) – recebe um *long* “radar_id”, devolve um objeto *Radar* “radar”
- @POST(“sincroapp/payInfraction”) – recebe um *long* “infraction_id”, uma *String* “cardNumber”, uma *String* “expirationDate” e uma *String* “cvv”, não devolve parâmetros

- @POST("sincroapp/updateCitizenEmail") – recebe um objeto *Citizen* “citizen”, não devolve parâmetros
- @POST("sincroapp/updateCitizenPhoneNumber") – recebe um objeto *Citizen* “citizen”, não devolve parâmetros
- @POST("sincroapp/getInfractions") – recebe um objeto *Citizen* “citizen”, devolve duas listas: lista de objeto *Vehicle* “ownedVehicles”, lista de objeto *Vehicle* “drivenVehicles”
- @POST("sincroapp/giveVehicleAuthorization") – recebe um objeto *VehicleAuthorization* “vehicleAuthorization”, não devolve parâmetros

5. Conclusões

Com a realização deste projeto foi salientada a importância do planeamento de um sistema, nomeadamente na conjugação entre as funcionalidades pretendidas, o modelo de dados proposto e a estimativa de tempo que será necessário para a implementação, dependendo também dos elementos disponíveis e da experiência dos mesmos nas várias áreas/tecnologias integrantes do projeto.

Referências

- [1] Diário da República, “Código da Estrada,” <https://dre.pt/web/guest/legislacao-consolidada/-/lc/117121508/201811290000/73639178/diploma/indice/8>, 2013.

[2] “RADARES de Portugal,” [Online]. Available:

https://play.google.com/store/apps/details?id=pt.radarespt&hl=pt_PT.

[3] “Waze - GPS, Maps, Traffic Alerts & Live Navigation,”

<https://play.google.com/store/apps/details?id=com.waze>.