

**Documento de Requisitos do Sistema**

***SafeKey***

**Alex José Alves Ventura**

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 10/04/2021 | 1.0 | Fechamento do escopo do trabalho | Alex Ventura  Gustavo Henrique |
| 06/05/2021 | 1.0 | Definição da descrição funcional | Alex Ventura  Gustavo Henrique |
| 17/06/2021 | 1.0 | Diagrama de caso de uso, classes, sequência, componentes, implantação e modelagem de dados, fluxo ótimo, protótipo. | Alex Ventura  Gustavo Henrique |
| 26/10/2021 | 1.1 | Revisão de diagramas e componentes. | Alex Ventura |
| 10/11/2021 | 2.0 | Revisão da sequência, diagramas e a definição geral do projeto; | Alex Ventura |

**Sumário**

**1 - Introdução**

**2 - Business Drivers/Background**

**3 - Metodologia**

**4 - Descrição geral do sistema**

**5 - Diagrama de Atividades**

**6 - Levantamento de Requisito**

**7 - Diagrama de Caso e Uso**

**8 - Diagrama de Classe**

**9 - Diagrama de Sequência**

**10 - Diagrama de Componente**

**11 - Diagrama de Implantação**

**12 - Modelagem de Dados**

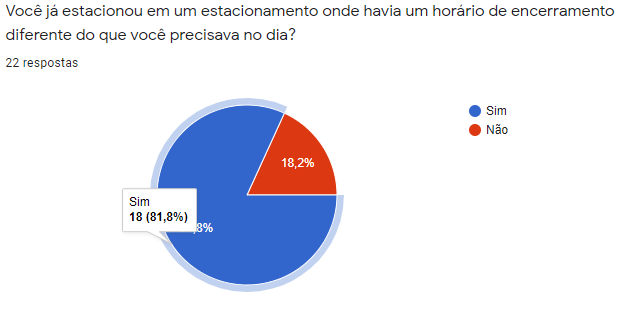
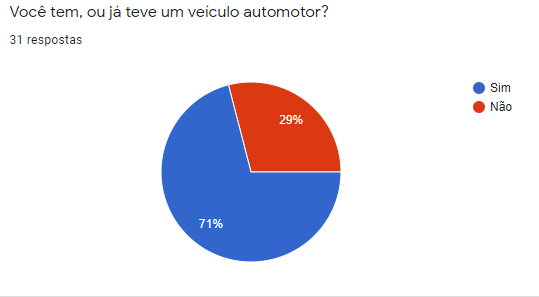
**13 - Protótipo**

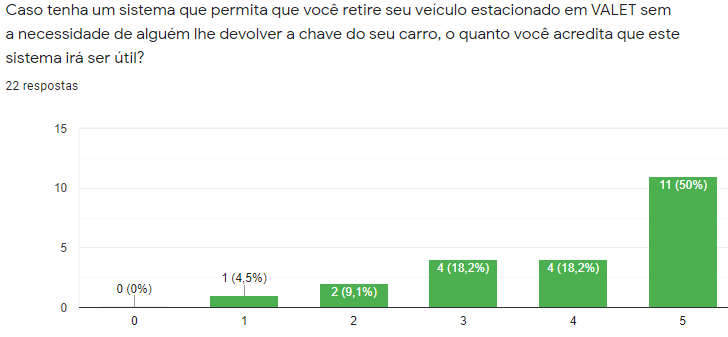
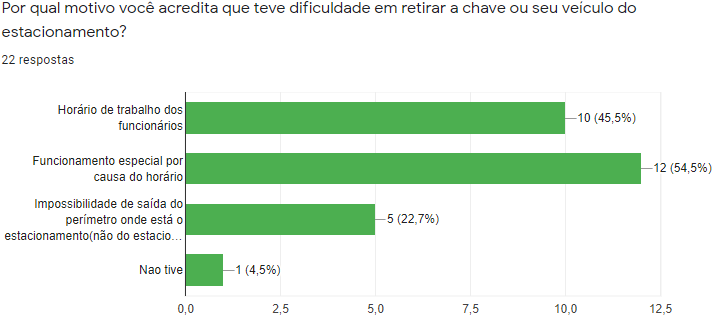
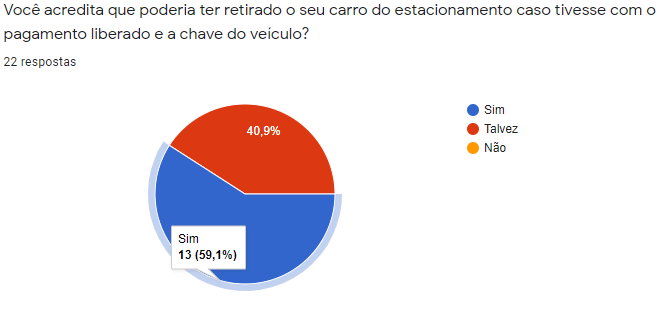
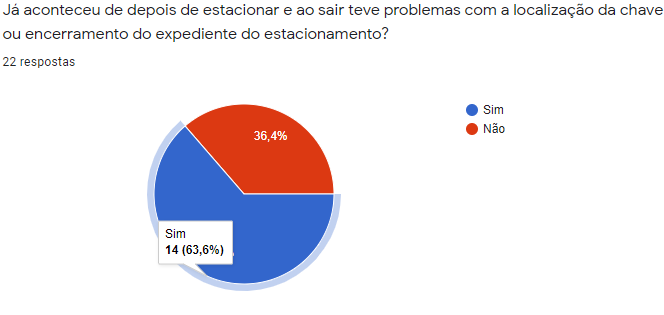
1. **Introdução**

Sistema seguro de estacionados em modalidade Valet, visando facilitar a saída do motorista estabelecimento, assim o motorista pode utilizar o estacionamento 24h com o sistema criado no qual o motorista ~~faz o pedido no app,~~ tem a opção de guardar a chave no SafeKey e quando retornar ao veículo efetuar o pagamento e retirar a suas chaves. ~~é veio automático~~

**1.1 Pesquisa de mercado**

Foi realizada pesquisa simples através de um formulário onde se identifica que, a maior parte das pessoas entrevistadas tiveram dificuldades com o horário de funcionamento do estabelecimento onde estacionaram e o horário que gostariam de retirar seu veículo, também verifica-se que grande parte teve problemas ao tentar sair devido ao esquema de horários de funcionamento ou de trabalho dos colaboradores, onde caso estivessem de porte das suas chaves poderiam retirar seu veículo, no horário desejado.





1. **Business Drivers/Background**

O objetivo desse sistema é facilitar e automatizar os sistemas de estacionamento por meio de um sistema que guarda a chave do carro, no qual só pode ser liberado pelo código no celular do usuário.

1. **Metodologia**

Este projeto é acadêmico portanto será desenvolvido no modelo cascata;

A equipe de desenvolvimento será distribuída entre os papéis de responsabilidades: Coordenação, Analista de sistemas, Testers, Desenvolvedores;

Esse projeto inicialmente era para ser desenvolvido utilizando como modelo MVC (Model View Controller) utilizando as linguagens Java e Servidor Apache® para fazer a parte do sistema web. Para o aplicativo utilizaríamos kotlin e Java. O banco de dados será desenvolvido, com o MySQL, conector Tomcat 8.5 .

1. **Descrição geral do sistema**

**4.1 Descrição Funcional**

O projeto consiste em uma aplicação, onde o usuário terá um dispositivo para depósito da chave de seu veículo, onde a retirada desta chave estará atrelada de forma segura a dados pessoais e/ou senha de cadastro.

No lado da aplicação do dispositivo de locker (dispositivo de travamento das chaves), o usuário poderá ter acesso a locker destravados somente através de inserção de seu CPF, e uma senha de no mínimo 6 dígitos. Para usuários que desejarem se cadastrar e utilizar o aplicativo terão maiores funcionalidades e maior segurança, através de travamento e destravamento por código de geração aleatória atrelado a dados pessoais do usuário do sistema, também terá na aplicação a possibilidade de ativar até dois lockers e saber quantos dispositivos ainda estão livres no local desejado.

Para a aplicação deverá ter, em suma, um cadastro contendo CPF e nome completo, data de nascimento, deverá ter também a descrição do veículo como o ano de fabricação, modelo, cor e placa. Não terá acesso a aplicação usuários que não aceitarem os termos de uso ou não desejarem completar o cadastro. É necessário para finalizar o cadastro uma inserção de senha pessoal de acesso a aplicação, pode estar disponível acesso por biometria do dispositivo.

O usuário cadastrado ou não se direciona até o locker e verifica o dispositivo disponível, se for cadastrado ele conseguirá abrir o dispositivo ao selecionar a opção de abrir um locker, no aplicativo e digitando o código que será gerado na aplicação, caso não seja cadastrado deverá selecionar a opção de abrir um locker e entrar com o CPF e uma senha pessoal no dispositivo local, o locker disponível irá abrir, após fechar o locker ele será travado e somente realizará a abertura com inserção dos dados anteriormente destacados, para o caso de usuários com castrado será aberto inserindo o código gerado e para usuários não cadastrados através da seleção de opção de abrir e inserindo o CPF e senha no próprio dispositivo local instalado no ambiente do locker.

1. **Diagrama de Atividades**

**Diagrama

Descrição gerada automaticamente**

**Diagrama

Descrição gerada automaticamente**

1. **Levantamento de Requisitos**
   1. Requisitos Funcionais (RF).
2. Cadastro (CRUD) de usuário com os dados: Nome, CPF, e-mail, número do celular, senha e endereço (logradouro, bairro, cidade e número).
3. Cadastro (CRUD) do automóvel do usuário com os dados: ano de fabricação, marca, modelo e placa.
4. Cadastro (CRUD) do locker tem o endereço (logradouro, bairro, cidade e número), se ele está ativo ou inativo e o status (vazio ou ocupado).
5. O sistema deverá gerar uma senha para o usuário conseguir abrir o SafeKey.
6. O sistema irá gerar relatórios, filtrando por datas (semanal, mensal, trimestral etc.), nesses relatórios irá conter informações mostrando quantas pessoas usaram o SafeKey, valor gerado, horário de mais uso.
7. O sistema irá gerar uma senha universal para desbloquear o SafeKey.

5.2 Requisitos Não Funcionais (NF)

1. O sistema tem que ter conectividade com a internet
2. O sistema será desenvolvido para Android e iOS
3. Terá o idioma português como padrão
4. A cor padrão do sistema será azul com branco.
5. O cadastro dos lokers, é feito apenas pelos devs.

5.3 Regras de Negócio

1. O Usuário só pode desbloquear 1 SafeKey sem cadastro prévio e 2 SafeKey com cadastro ativo na plataforma.
2. O Usuário só pode finalizar o cadastro depois que aceitar os termos cadastrais.
3. O Loker só vai destrancar se ele estiver com Status = 0
4. O Locker só vai aparecer para o usuário quando o ativo = 1
5. **Diagrama de Caso de Uso**

Diagrama

Descrição gerada automaticamente

**Fluxo ótimo:**

1. **Diagrama de Classe**

Diagrama

Descrição gerada automaticamente

Uma imagem contendo Diagrama

Descrição gerada automaticamente

**9. Diagrama de Sequência**

Diagrama

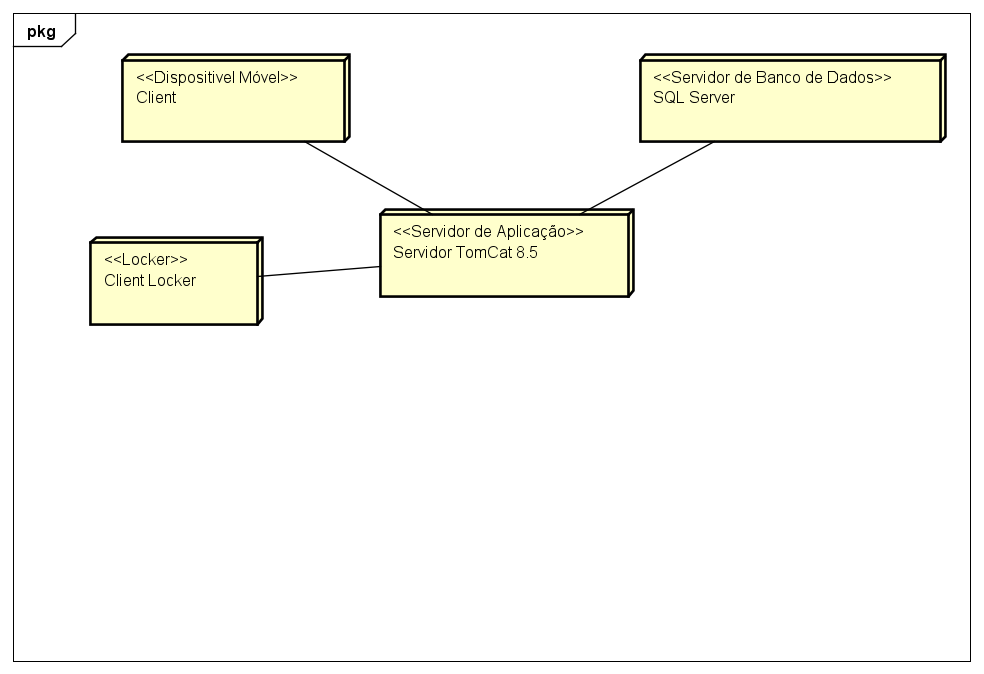
Descrição gerada automaticamente

**10. Diagrama de Componentes**

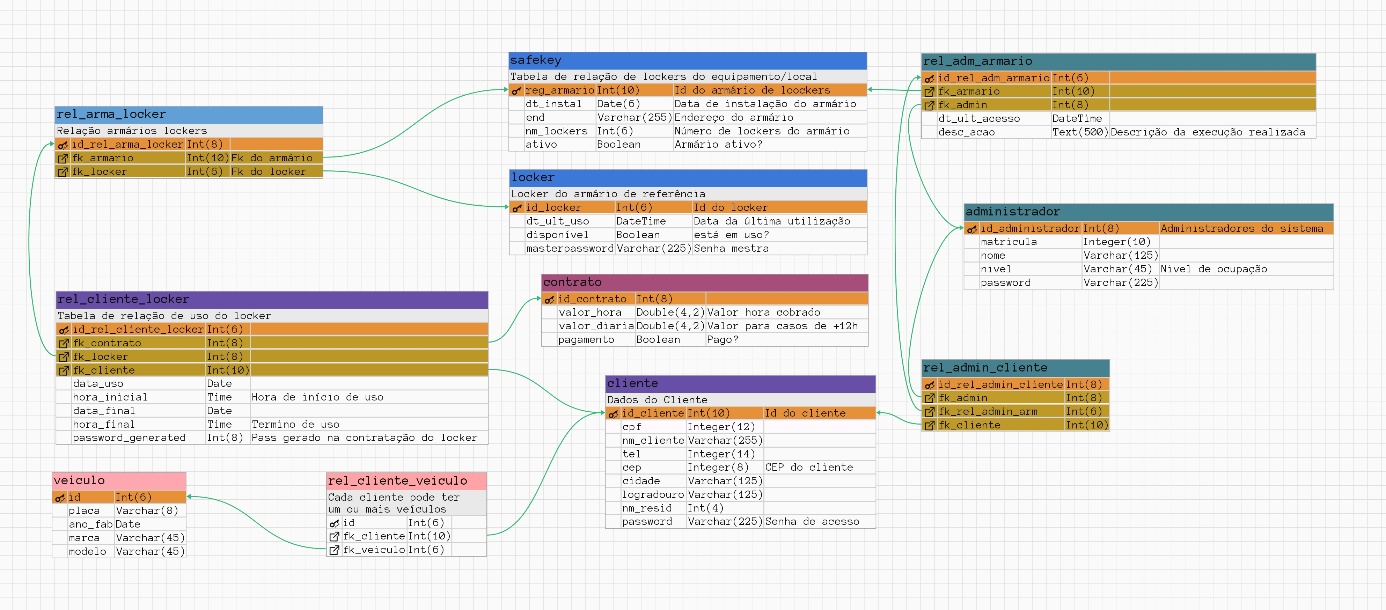
Diagrama

Descrição gerada automaticamente

**11. Diagrama de Implantação**



**12. Modelagem de Dados**



SQL\_export :

CREATE TABLE safekey (

Reg\_armario INT(10) NOT NULL UNIQUE PRIMARY KEY,

Dt\_instal DATE(6) NOT NULL,

End VARCHAR(255),

Nm\_lockers INT(6),

Ativo BOOLEAN DEFAULT ‘true’);

CREATE TABLE locker (

Id\_locker INT(6) NOT NULL UNIQUE PRIMARY KEY,

Dt\_ult\_uso DATETIME,

Disponível BOOLEAN DEFAULT ‘true’,

Masterpassword VARCHAR(225));

CREATE TABLE rel\_arma\_locker (

Id\_rel\_arma\_locker INT(8) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Fk\_armario INT(10) NOT NULL UNIQUE,

Fk\_locker INT(6) NOT NULL UNIQUE);

CREATE TABLE cliente (

Id\_cliente INT(10) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Cpf INTEGER(12) UNIQUE,

Nm\_cliente VARCHAR(255) NOT NULL,

Tel INTEGER(14),

Cep INTEGER(8),

Cidade VARCHAR(125) NOT NULL,

Logradouro VARCHAR(125),

Nm\_resid INT(4),

Password VARCHAR(225));

CREATE TABLE rel\_cliente\_locker (

Id\_rel\_cliente\_locker INT(6) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Fk\_contrato INT(8),

Fk\_locker INT(8) NOT NULL,

Fk\_cliente INT(10) UNIQUE,

Data\_uso DATE,

Hora\_inicial TIME,

Data\_final DATE,

Hora\_final TIME,

Password\_generated INT(8) UNIQUE);

CREATE TABLE administrador (

Id\_administrador INT(8) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Matricula INTEGER(10) UNIQUE,

Nome VARCHAR(125),

Nivel VARCHAR(45),

Password VARCHAR(225));

CREATE TABLE rel\_adm\_armario (

Id\_rel\_adm\_armario INT(6) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Fk\_armario INT(10),

Fk\_admin INT(8),

Dt\_ult\_acesso DATETIME DEFAULT ‘data e hora de último acesso adm’,

Desc\_acao TEXT(500));

CREATE TABLE rel\_admin\_cliente (

Id\_rel\_admin\_cliente INT(8) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Fk\_admin INT(8),

Fk\_rel\_admin\_arm INT(6) NOT NULL,

Fk\_cliente INT(10) NOT NULL);

CREATE TABLE contrato (

Id\_contrato INT(8) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Valor\_hora DOUBLE(4,2),

Valor\_diaria DOUBLE(4,2),

Pagamento BOOLEAN DEFAULT ‘false’);

CREATE TABLE veiculo (

Id INT(6) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Placa VARCHAR(8) UNIQUE,

Ano\_fab DATE,

Marca VARCHAR(45) NOT NULL,

Modelo VARCHAR(45));

CREATE TABLE rel\_cliente\_veiculo (

Id INT(6) AUTO\_INCREMENT NOT NULL UNIQUE PRIMARY KEY,

Fk\_cliente INT(10) NOT NULL,

Fk\_veiculo INT(6) NOT NULL);

ALTER TABLE rel\_arma\_locker ADD CONSTRAINT rel\_arma\_locker\_fk\_armario\_safekey\_reg\_armario FOREIGN KEY (fk\_armario) REFERENCES safekey(reg\_armario);

ALTER TABLE rel\_arma\_locker ADD CONSTRAINT rel\_arma\_locker\_fk\_locker\_locker\_id\_locker FOREIGN KEY (fk\_locker) REFERENCES locker(id\_locker);

ALTER TABLE rel\_cliente\_locker ADD CONSTRAINT rel\_cliente\_locker\_fk\_contrato\_contrato\_id\_contrato FOREIGN KEY (fk\_contrato) REFERENCES contrato(id\_contrato);

ALTER TABLE rel\_cliente\_locker ADD CONSTRAINT rel\_cliente\_locker\_fk\_locker\_rel\_arma\_locker\_id\_rel\_arma\_locker FOREIGN KEY (fk\_locker) REFERENCES rel\_arma\_locker(id\_rel\_arma\_locker);

ALTER TABLE rel\_cliente\_locker ADD CONSTRAINT rel\_cliente\_locker\_fk\_cliente\_cliente\_id\_cliente FOREIGN KEY (fk\_cliente) REFERENCES cliente(id\_cliente);

ALTER TABLE rel\_adm\_armario ADD CONSTRAINT rel\_adm\_armario\_fk\_armario\_safekey\_reg\_armario FOREIGN KEY (fk\_armario) REFERENCES safekey(reg\_armario);

ALTER TABLE rel\_adm\_armario ADD CONSTRAINT rel\_adm\_armario\_fk\_admin\_administrador\_id\_administrador FOREIGN KEY (fk\_admin) REFERENCES administrador(id\_administrador );

ALTER TABLE rel\_admin\_cliente ADD CONSTRAINT rel\_admin\_cliente\_fk\_admin\_administrador\_id\_administrador FOREIGN KEY (fk\_admin) REFERENCES administrador(id\_administrador );

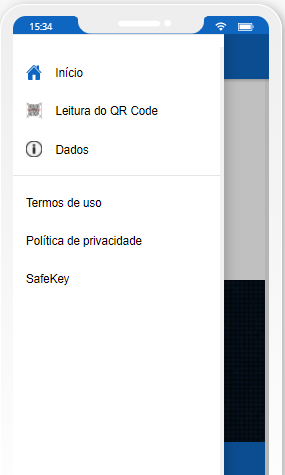
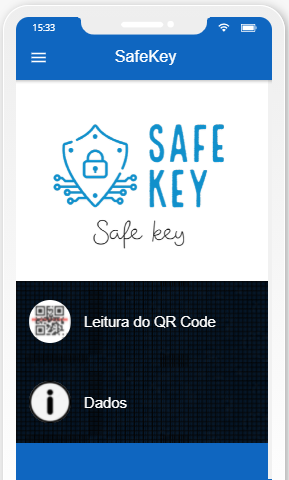
ALTER TABLE rel\_admin\_cliente ADD CONSTRAINT rel\_admin\_cliente\_fk\_rel\_admin\_arm\_rel\_adm\_armario\_id\_rel\_adm\_armario FOREIGN KEY (fk\_rel\_admin\_arm) REFERENCES rel\_adm\_armario(id\_rel\_adm\_armario);

ALTER TABLE rel\_admin\_cliente ADD CONSTRAINT rel\_admin\_cliente\_fk\_cliente\_cliente\_id\_cliente FOREIGN KEY (fk\_cliente) REFERENCES cliente(id\_cliente);

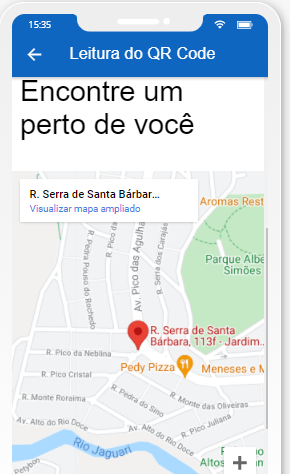
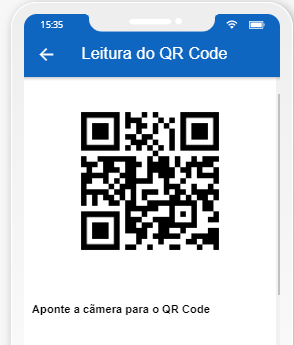
ALTER TABLE rel\_cliente\_veiculo ADD CONSTRAINT rel\_cliente\_veiculo\_fk\_cliente\_cliente\_id\_cliente FOREIGN KEY (fk\_cliente) REFERENCES cliente(id\_cliente);

ALTER TABLE rel\_cliente\_veiculo ADD CONSTRAINT rel\_cliente\_veiculo\_fk\_veiculo\_veiculo\_id FOREIGN KEY (fk\_veiculo) REFERENCES veiculo(id);

**13. Protótipo**

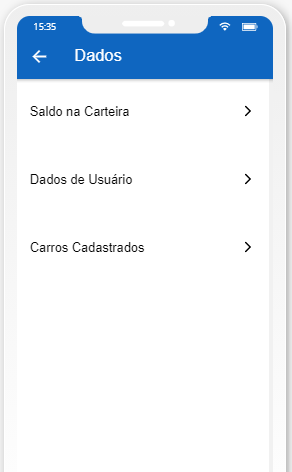


*Figura 1 - Página Inicial Figura 2 - Menu Lateral*



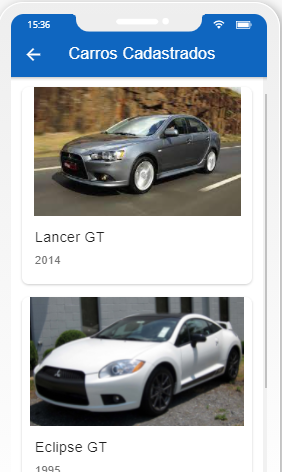
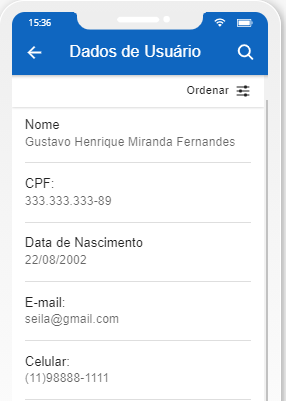
*Figura 3 - Leitura de QR Code*

*Figura 4 - Mapa mostrando onde pode encontrar outros SafeKey*

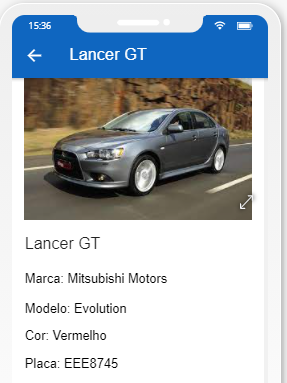


*Figura 7 - Saldo da Carteira*

*Figura 6 - Aba de dados*



*Figura 7 - Dados do usuário*



*Figura 9 - Lista de carros cadastrados*

*Figura 10 - Dados do carro*