



Universidade Federal da Paraíba  
Centro de Informática

Sistemas Embarcados: Relatório da Parte 1  
Projeto Locar

Aléxandros Augustus  
Allef Brenno  
Stênio Ellison

João Pessoa  
2022

## **1. Introdução**

Este relatório tem por objetivo descrever o projeto e estado atual do aplicativo Locar, projeto desenvolvido pelos alunos para a disciplina de Sistemas Embarcados. O aplicativo tem como proposta oferecer a empresas um serviço de organização e análise de perfil de seus motoristas, estimulando os motoristas a ter mais cuidado com sua direção e permitindo às empresas visualizar com clareza as deficiências a serem trabalhadas.

## **2. Tecnologias Empregadas**

O aplicativo está sendo desenvolvido na linguagem Java, utilizando a API React Native, selecionados devido à sua atual relevância no mercado. Para o banco de dados da aplicação, está sendo utilizado MongoDB graças a sua facilidade de criar um protótipo *online* funcional. A proposta atual para o protótipo em si é que seja executado em um emulador no computador, porém a coleta de dados está sendo feita em um aparelho celular físico pessoal.

## **3. Implementação**

O primeiro passo foi pensar nas funções da aplicação e o que poderia ser melhorado ou removido, na criação de telas, a logotipo, nome e paleta de cores foram escolhidas, a princípio tínhamos em mente que o aplicativo iniciaria com dois botões para identificar quem era gerenciador ou motorista, mas achamos melhor criar um login único, assim todos possuem a possibilidade de gerenciar uma empresa ou mais empresas, ou entrar como motorista em uma existente.

Como o principal objetivo do aplicativo é traçar um perfil de condução de um motorista vinculado a uma empresa, ou seja, à medida em que o funcionário realiza suas corridas, seu perfil vai sendo traçado e exposto em sua conta. O gerenciador da empresa poderá ver todos os perfis de conduta dos motoristas cadastrados em sua empresa gerados pelas corridas realizadas na mesma.

O estágio atual do projeto foi a finalização das telas utilizando o software Figma, e o início da criação de banco de dados e coleta dos dados para gerar o modelo de avaliação.

## **4. Coleta de dados**

Os dados experimentais estão sendo coletados através de um *smartphone* conectado a um computador executando um código utilizando o React Native para obter os dados do giroscópio do aparelho, enquanto este é movimentado de forma a simular fenômenos desconfortáveis detectáveis durante uma corrida, sendo estes:

- Terreno acidentado;
- freadas repentinas;

- Passar por quebra-molas em velocidade excessiva;
- Curvas bruscas.

O intuito da coleta é identificar valores nos eixos do giroscópio que forneçam uma fundamentação inicial para o programa que irá avaliar o grau de conforto do passageiro proporcionado pelo motorista.

## 5. Visual

Inicialização:

12:49



Telas de inicialização, login e cadastro

Tela para primeiro login do gerenciador para ser adicionado ao banco de dados:

Locar

Empresa:

Adicionar

Locar

Registro

CarroMotorista

Modelo:

Velocidade máxima:

Placa:

Adicionar

Locar

Registro

CarroMotorista


Buscar:

Enviar convite

Primeira tela é cadastrar o nome da empresa, a segunda é o cadastro dos modelos dos carros que serão utilizados e a terceira é a de envio de convite para o motorista fazer parte da empresa e abaixo a confirmação do cadastro.

Locar

CarroMotorista




Carro cadastrado com sucesso!

Adicionar mais veículos

Finalizar

Locar

CarroMotorista

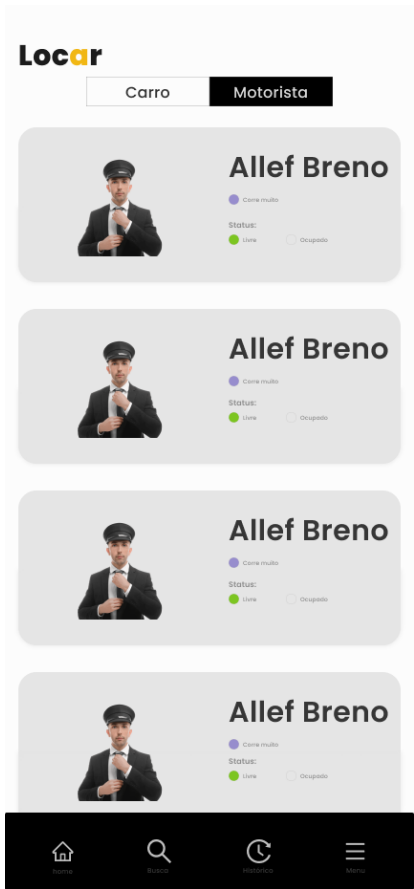
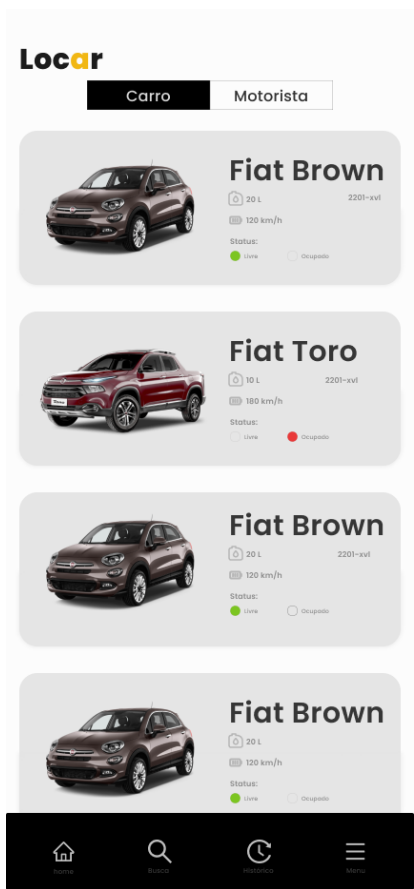


Convite enviado

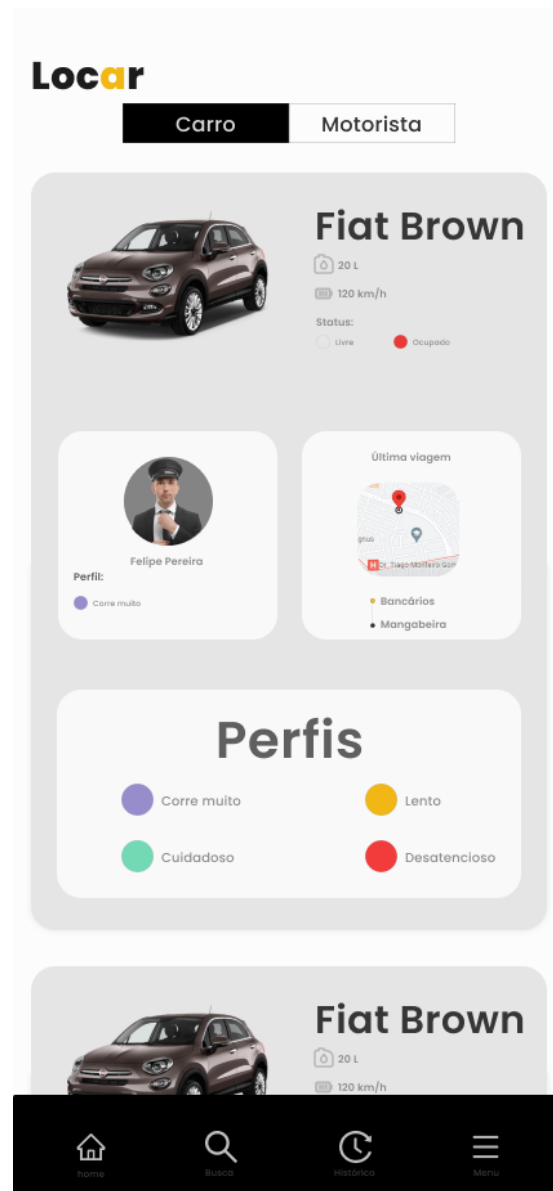
Buscar mais motoristas

Finalizar

Uma vez logado e banco feito (cadastro de carros e da empresa), as telas do gerenciador serão respectivamente:

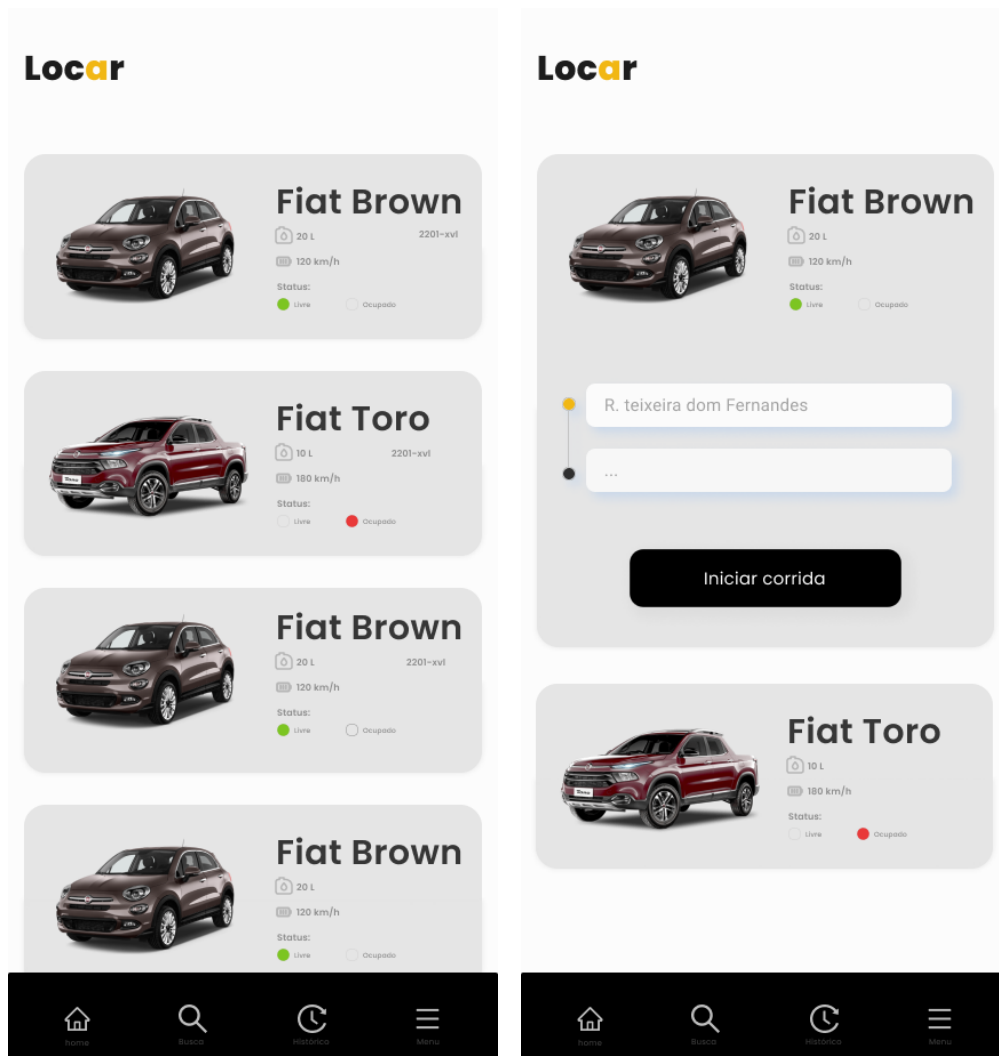


Essas são as telas de gerenciamento dos carros disponíveis e motoristas, ademais o perfil de condução, velocidade máxima do carro, ao clicar nos cards dos carros a seguinte tela aparecerá.



Com informações das últimas viagens, perfil de condução, nome do motorista e uma legenda dos perfis que podem classificá-los.

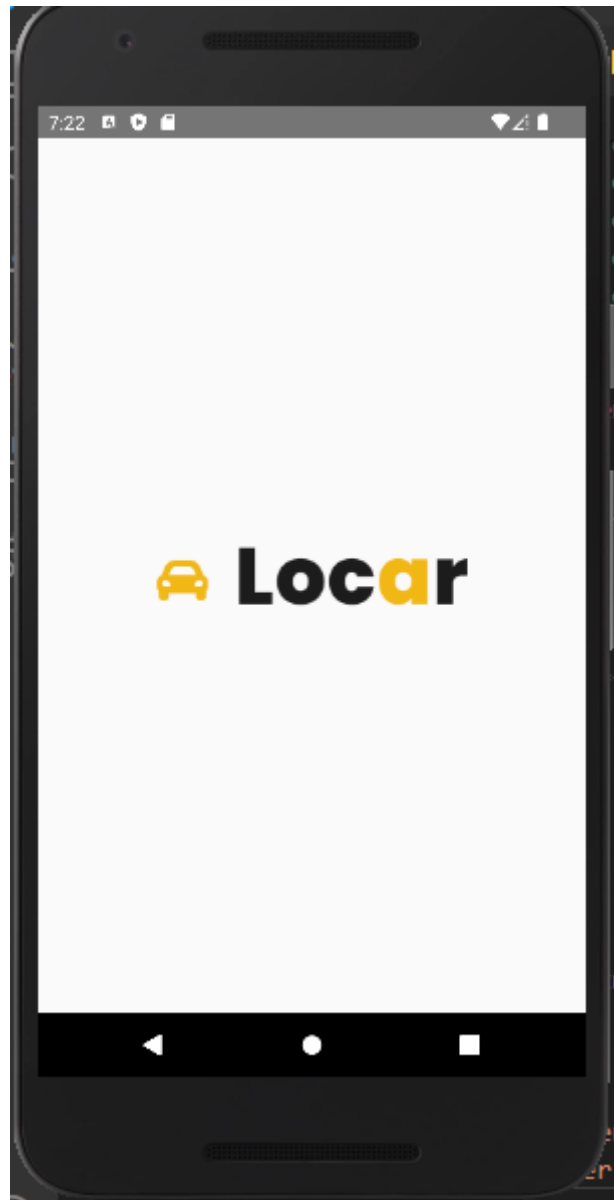
Agora a tela dos motoristas, que mostrará os carros disponíveis e ao clicar no *card* exibirá a opção de iniciar a corrida para o aplicativo coletar dados de posição e velocidade no intuito de traçar o perfil de condução do mesmo:



- **Codificação:**

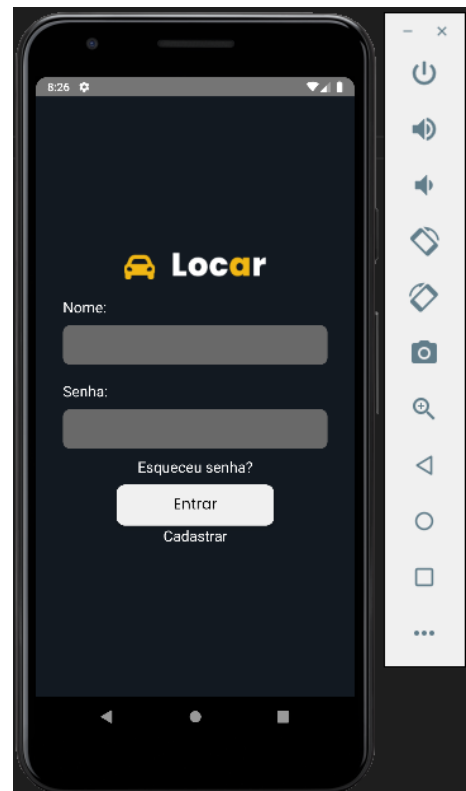
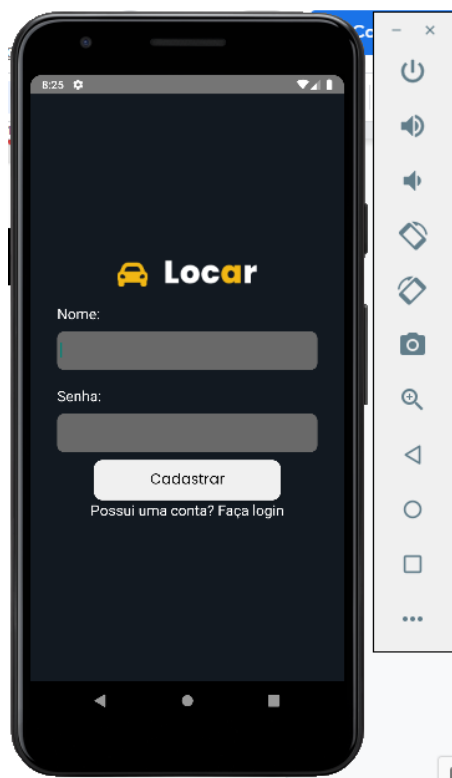
No momento foi iniciado os estudos sobre o React Native, para começar a codificação das telas produzidas no Figma, utilizamos como base o curso da udemy “React Native: Desenvolva APPs Nativas para Android e iOS” criado pela a empresa Cod3r, a programação neste framework foi algo novo para o grupo, utilizamos também o android Studio para emular o aparelho android e criamos código pelo o Visual Studio Code mas algumas telas já foram produzidas como a tela da logo, do login, cadastro e cadastro de empresa(menu em desenvolvimento), abaixo as imagens:

## Tela Logo:



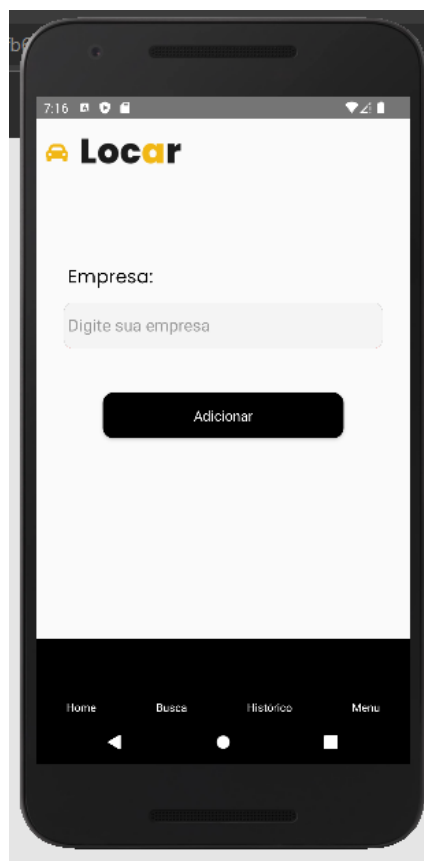


## Cadastro e Login:



No momento atual é possível realizar o cadastro de contas e entrar na aplicação com essa conta. Quanto aos layouts da aplicação, como a parte visual não é uma prioridade neste momento, podem haver mudanças em comparação ao aplicativo atual.

## Tela Cadastro Empresa:



## Sensores:

Utilizando RN Sensors também é possível analisar os valores de aceleração e giroscópio e imprimir seus valores . Com isso será possível determinar o perfil dos usuários junto de sua trajetória percorrida com maps.

```
node "C:\ProjetoReact\ProjetoEmbarcados\node_modules\react-native\scripts\cli.js" start
LOG {"timestamp": 1654807122851, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122862, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122872, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122882, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122893, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122903, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122914, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122924, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122934, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122944, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122954, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122964, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122974, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122984, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807122994, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123004, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123014, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123024, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123035, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123045, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123055, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123065, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123075, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123085, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123095, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123105, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123115, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123125, "x": 0, "y": 9.8100004196167, "z": 0}
LOG {"timestamp": 1654807123135, "x": 0, "y": 9.8100004196167, "z": 0}
```

Após a coleta, identificamos que o giroscópio é extremamente sensível, o que se não avaliado cuidadosamente, poderá classificar incorretamente o motorista a depender do percurso, por exemplo, ao subir ou descer ladeiras, o giroscópio sofre uma mudança brusca, porém estável, sendo assim, ao invés de avaliar os valores em si vamos avaliar a oscilação desses valores, já que oscilações menores indicam uma viagem mais confortável. Sendo assim o próximo passo é fazer a função de avaliação do motorista, ao finalizar a corrida, os dados do *log* são analisados e a velocidade média é medida, caso esteja entre 50 a 80 km/h, o motorista é classificado de forma ótima, abaixo disso é classificado como lento e acima disso é classificado como rápido demais. Quanto ao conforto, será avaliado se a oscilação de valores do giroscópio excede 10, caso exceda mais de 3 vezes, a viagem é considerada desconfortável, caso contrário confortável, assim gerando as 4 classes de classificação do motorista:

Caso a viagem seja muito desconfortável, ele será classificado como “Desatencioso”. Caso seja confortável, será classificado como “Cuidadoso” se sua velocidade não ultrapassar os limites máximo e mínimo, “Corre muito” caso exceda o limite máximo e “Lento” caso esteja abaixo do limite mínimo.