



SENAI Jaguariúna

FELIPE REIS DE OLIVEIRA

GABRIEL ALEX SANDRO SOARES LANÇONI

MATHEUS RICARDO BELLI

VICTÓRIA ROSA DOS SANTOS

PROJETO CIASTO

Jaguariúna – SP

2022

FELIPE REIS DE OLIVEIRA
GABRIEL ALEX SANDRO SOARES LANÇONI
MATHEUS RICARDO BELLI
VICTÓRIA ROSA DOS SANTOS

PROJETO CIASTO

Projeto de conclusão parcial de curso apresentado ao SENAI Jaguariúna, como parte dos requisitos para obtenção do título de Técnico em Desenvolvimento de Sistemas.

Orientador:

Reenye Alexandre De Lima
Wellington Fábio De Oliveira Martins

Jaguariúna – SP

Março – 2022

Lista de Figuras

Figura 1 - DCU - [RF001].....	15
Figura 2 - CDU - [RF002].....	16
Figura 3 - DCU - [RF005].....	19
Figura 4 - DCU - [RF007].....	21
Figura 5 - DCU - [RF014].....	22
Figura 6 - DCU - [RF015].....	22
Figura 7 - DCU - [RF018].....	28
Figura 8 - Cronograma - Gantt.....	29
Figura 9 - Legenda - Gantt.....	26
Figura 10 - Legenda - Gantt.....	26
Figura 11 - Tela Principal (Home - Satellite).....	30
Figura 12 - Inserindo um novo alerta	31
Figura 13 - Escolhendo o tipo de alerta	31
Figura 14 - Alterar o Perfil do Usuário	32
Figura 15 - Escolhendo os alertas ativos	32
Figura 16 - Listando os alertas gerados.....	33
Figura 17 - Login App	34
Figura 18 - Cadastro App.....	34
Figura 19 - Home App.....	35
Figura 20 - Home Satellite App.....	35
Figura 21 - Escolher alerta App	36
Figura 22 - Inserir alerta App	36
Figura 23 - Menu App	37
Figura 24 - Menu voltar App	37

Figura 25 - Perfil App	38
Figura 26 - Alertas ativos App.....	38
Figura 27 - Listar alertas App.....	39
Figura 28 - Modelo Conceitual.....	40
Figura 29 - Modelo Lógico	40
Figura 30 - Diagrama de Classes	41

Lista de Abreviaturas e Siglas

Ágil	- Metodologia Ágil (Metodologia de desenvolvimento de sistemas)
API	- Application Programming Interface (Interface de programação de aplicação)
BackEnd	- Prática de programação para Servidor de Aplicação e ou Banco de Dados.
Backlog	- Requisitos funcionais do projeto a serem desenvolvidos.
CSS	- Linguagem de marcação e estilização de páginas.
DER	- Diagrama de Entidade e Relacionamento
Framework	- Estrutura, conjunto de códigos genéricos capaz de unir trechos de um projeto.
FrontEnd	-Prática de programação para cliente de programas para internet.
FullStack	- Prática de programação em todas as camadas Cliente e Servidor
Gantt	- Grafico para gestão do tempo e recursos do projeto.
GitHub	- Repositório para códigos fonte e rede social de programadores.
HTML	- Hyper Text Management Language, linguagem de marcação.
JavaScript	- Linguagem de programação para a internet.
JSON	- Objeto JavaScript, padrão de dados de uma API
MER	- Modelo Entidade e Relacionamento
Mobile	- Prática de programação para dispositivos móveis também clientes de programas para a internet.
MVC	- Modelo Visão e Controle
ORM	- Object Relacional Mapper (Mapa Objeto Relacional)
PMO	- Project Management Office (Escritório de Projetos).
SCRUM	- Subcategoria da Metodologia Ágil.
Sequelize	- recurso ORM do framework Node.js

- Sprint - tempo de desenvolvimento de duas a quatro semanas.
- UML - Unified Modeling Language (Linguagem de Modelagem Unificada)
- Node.js - Framework que utiliza a linguagem JavaScript no backend.
- 3D - Computação gráfica tridimensional são gráficos que usam representações tridimensionais de dados geométricos que são armazenados em um computador com o propósito de realizar cálculos e renderizar imagens 2D.
- Blender - Blender, também conhecido como blender3d, é um programa de computador de código aberto, desenvolvido pela Blender Foundation, para modelagem, animação, texturização, composição, renderização, e edição de vídeo. Está disponível sob a GNU GPL, versão 2 ou posterior.

Resumo

O aplicativo Ciasto é um projeto de internet fullstack com banco de dados relacional MySQL, BackEnd em Node.js, FrontEnd sem framework.

O objetivo é ser uma loja de bolos com um design agradável para o consumidor, segurança em suas contas e pagamentos, um sistema de entregas e com um sistema de montagem de bolos 3D.

Além disso, servir de desenvolvimento, aprendizado e modelo para os quatro integrantes do projeto, e do grupo externo, qual realizou a parte de BackEnd através de uma atividade proposta pelos professores do curso Técnico em Desenvolvimento de Sistemas do SENAI Jaguariúna.

Abstract

The Ciasto application is a fullstack internet project with MySQL relational database, BackEnd in Node.js, FrontEnd without framework.

The goal is to be a cake shop with a pleasant design for the consumer, security in your bills and payments, a delivery system and a 3D cake assembly system.

In addition, it will serve for the development, learning and model for the four members of the project, and the group, qualified a part of BackEnd through an activity proposed by the teachers of the Technical course in Systems Development of SENAI Jaguariúna.

SUMÁRIO

1.	TAP (Termo de Abertura do Projeto)	13
2.	EAP (Escopo do Projeto)	14
2.1.	Requisitos não funcionais	14
2.2.	Requisitos Funcionais	14
3.	Planejamento (Cronograma Grafico Gantt)	26
3.1.	Planejamento de Custos	27
4.	Protótipo	28
4.1.	Protótipo do site da internet	28
5.	Planejamento do Banco de Dados e da API.....	40
6.	Detalhes da Execução e Controle	42
7.	Resultados.....	48

INTRODUÇÃO

A turma intitulada 3DES do curso técnico em desenvolvimento de sistemas da escola SENAI Jaguariúna com conclusão do curso prevista para o mês de agosto do ano de 2022 tem se dedicado a soluções de problemas através de sistemas computacionais, também a automação de processos repetitivos utilizando recursos e bibliotecas das linguagens de programação como Python, planilhas eletrônicas e APIs disponíveis de forma gratuita através da internet.

(Blender)APIs segundo o site HostGator é a interface ideal para que os programas de diferentes organizações compartilhem suas ações, ferramentas, padrões e protocolo. Gerando, assim, uma total integração.

Neste contexto, como forma de construir um software FullStack como modelo para o que os alunos possam ter como base para desenvolver seus projetos de conclusão de curso. Este projeto foi desenvolvido e escrito pelos instrutores deste curso contando com a colaboração desta turma.

A solução está dividida em camadas chamadas de BackEnd, FrontEnd, Mobile e documentação, cada uma das camadas possui exemplos das linguagens de programação estudadas durante os dois primeiros semestres do curso e alguns recursos introduzidos neste terceiro e último semestre.

O objetivo do software, além de servir de modelo é a gestão de alertas de trânsito, utilizando o mapa obtido com dados da API do Google Maps, um condutor de veículo que tenha o aplicativo móvel instalado em seu smartphone poderá observar eventos no trânsito e notifica-los simplesmente apontando no mapa onde o evento está ocorrendo e classificando-o como blitz “Fiscalização ou investigação policial em andamento”, fenômenos da natureza como enchentes, alagamentos, queda de árvores, congestionamentos, lentidão, fiscalização eletrônica ou qualquer outra natureza de eventos, gerando assim um alerta para outros condutores que passem pelo local e utilizem o aplicativo.

Este trabalho está dividido seguindo os princípios e instruções de gestão de projetos PMO que segundo o PMBOK é uma estrutura organizacional que tem como objetivo padronizar processos de governança e gerenciamento de projetos.

Como forma de simplificar a lógica de programação foi adotada a linguagem JavaScript em todas as camadas do projeto, o framework utilizado no servidor de aplicação backend foi o Node.js. No frontend optou-se por não utilizar nenhum framework, mas sim recursos básicos das linguagens HTML, CSS e JavaScript.

Por fim para a programação de dispositivos móveis este modelo utilizou o framework React Native que também trabalha sobre a linguagem JavaScript.

OBJETIVOS

Desenvolver um aplicativo para venda de bolos, onde os clientes escolhem entre comprar bolos do catálogo ou montar seu próprio bolo, em um sistema como um jogo de montagem com livre escolhas para o cliente, mostrando no site o bolo em 3D.

Este projeto cumpre os seguintes objetivos específicos:

- Montagem de bolo no site mostrando-o em 3D.
- Servir de modelo para o desenvolvimento de soluções semelhantes.
- Servir de exemplo de consumo de API de terceiros.
- Servir de exemplo de armazenamento de imagens em banco de dados.
- Servir de exemplo de utilização de mapas via internet frontend e mobile.
- Servir de exemplo de utilização de recursos ORM como “sequelize”.

JUSTIFICATIVA

Ao concluir o curso técnico de desenvolvimento de sistemas o aluno adquire várias competências, o desenvolvimento de um projeto de conclusão auxilia a demonstrar estas capacidades técnicas na prática, serve como forma de avaliação somativa dentro das práticas da metodologia de ensino profissionalizante adotada pelo SENAI além de gerar a oportunidade de apresentar este projeto ao corpo docente da instituição e até a membros da gestão das indústrias da cidade de Jaguariúna e região.

1. TAP (TERMO DE ABERTURA DO PROJETO)

Título do Projeto:

Projeto Ciasto "Site de venda de bolos prontos ou montados"

Patrocinadores: SENAI Jaguariúna

Recursos Humanos	Felipe Reis De Oliveira Gabriel Alex Sandro Soares Lançoni Matheus Ricardo Belli Rodrigo Aparecido De Moraes Victória Rosa Dos Santos
Gerente do Projeto	Felipe Reis De Oliveira Gabriel Alex Sandro Soares Lançoni Matheus Ricardo Belli Victória Rosa Dos Santos
Patrocinador	SENAI Jaguariúna
Cliente	SENAI Jaguariúna
Prazo	31/03/2022

Local: _____ Data: ____/____/____	
Patrocinador	Ass: _____
Cliente	Ass: _____
Gerente do Projeto	Ass: _____

2. EAP (ESCOPO DO PROJETO)

O escopo de um projeto de aplicação fullstack, para a internet e para dispositivos móveis consiste de seus requisitos funcionais e não funcionais, este projeto, portanto está dividido desta forma.

2.1. REQUISITOS NÃO FUNCIONAIS

[RN001] Linguagens de Programação:

[RN001.1] BackEnd: Javascript com o framework Node.js

[RN001.2] BackEnd: biblioteca ORM sequelize.

[RN001.3] FrontEnd: HTML, CSS e JavaScript sem framework

[RN001.4] Mobile: Javascript com o framework ReactNative

[RN002] Servidor e possível hospedagem

[RN003] Recursos de Banco de Dados relacional e SQL

[RN003.1] MySQL 10.4.11-MariaDB

2.2. REQUISITOS FUNCIONAIS

Os requisitos funcionais serão apresentados utilizando o recuso UML Diagrama de Casos de Uso e uma classificação de criticidade.

A metodologia de desenvolvimento de sistemas aplicada neste projeto é conhecida como SCRUM que é uma subcategoria da metodologia Ágil, por este motivo cada requisito aqui apresentado faz parte do backlog do projeto que será dividido em duas Sprints conforme a sua criticidade.

A criticidade está dividida em três categorias: essencial, importante e desejável definindo o nível de prioridade para cada requisito acordado neste escopo.

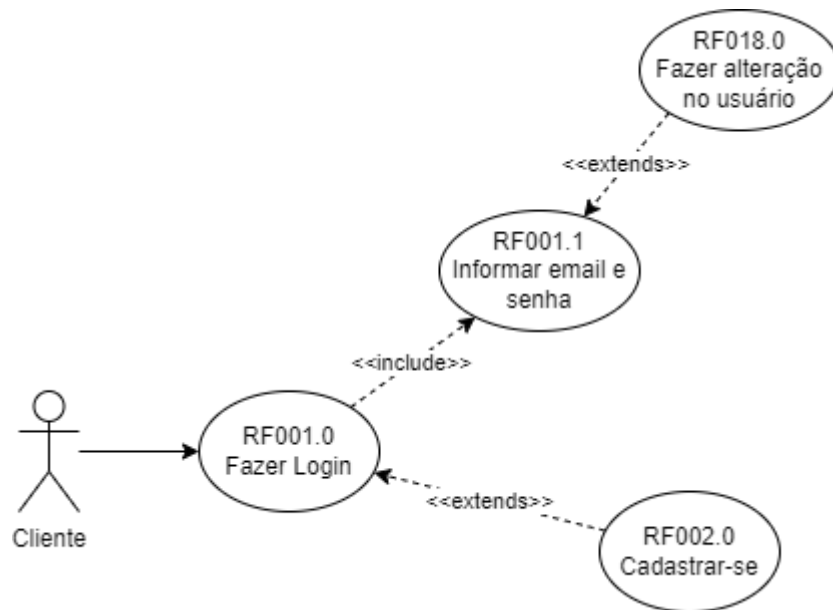


Figura 1 - DCU - [RF001]

[RF001.0]: Fazer login

Tela inicial do sistema para acessar ou cadastrar-se.

(X)Essencial () Importante () Desejável

[RF001.1]: Informar e-mail e senha

Preencher e-mail e senha.

(X)Essencial () Importante () Desejável

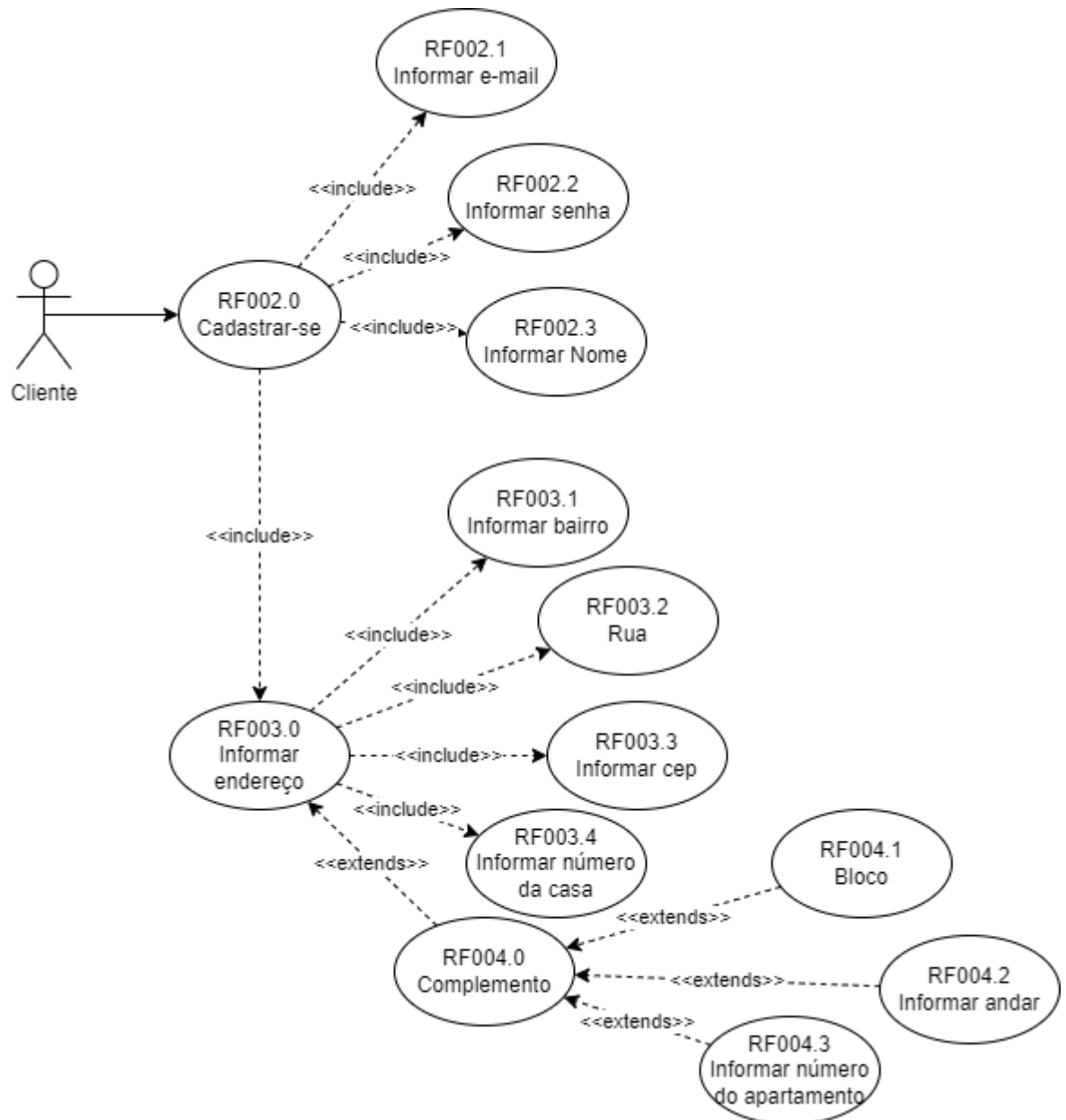


Figura 2 - CDU - [RF002]

[RF002.0]: Cadastrar-se

Preencher informações.

(X)Essencial ()Importante ()Desejável

[RF002.1]: Informar e-mail

Informar e-mail.

(X)Essencial ()Importante ()Desejável

[RF002.2]: Informar senha

Criar senha.

(X)Essencial ()Importante ()Desejável

[RF002.3]: Informar nome

Informar o nome da pessoa.

(X)Essencial ()Importante ()Desejável

[RF003.0]: Informar endereço

Preencher informações do endereço.

(X)Essencial ()Importante ()Desejável

[RF003.1]: Informar Bairro

Informar o nome do bairro.

(X)Essencial ()Importante ()Desejável

[RF003.2]: Informar Rua

Informar nome da rua.

(X)Essencial ()Importante ()Desejável

[RF003.3]: Informar Cep

Informar cep do local.

(X)Essencial ()Importante ()Desejável

[RF003.4]: Informar número da casa

Informar número de endereço da casa.

(X)Essencial ()Importante ()Desejável

[RF004.0]: Complemento

Informar complemento caso precise.

(X)Essencial ()Importante ()Desejável

[RF004.1]: Bloco

Caso o cliente more em apartamento ou condomínio, ele deve informar o bloco.

(X)Essencial ()Importante ()Desejável

[RF004.2]: Informar andar

Caso o cliente more em apartamento, ele deve informar o andar.

(X)Essencial ()Importante ()Desejável

[RF004.3]: Informar número do apartamento

Caso o cliente more em apartamento, ele deve informar o número do apartamento.

(X)Essencial ()Importante ()Desejável

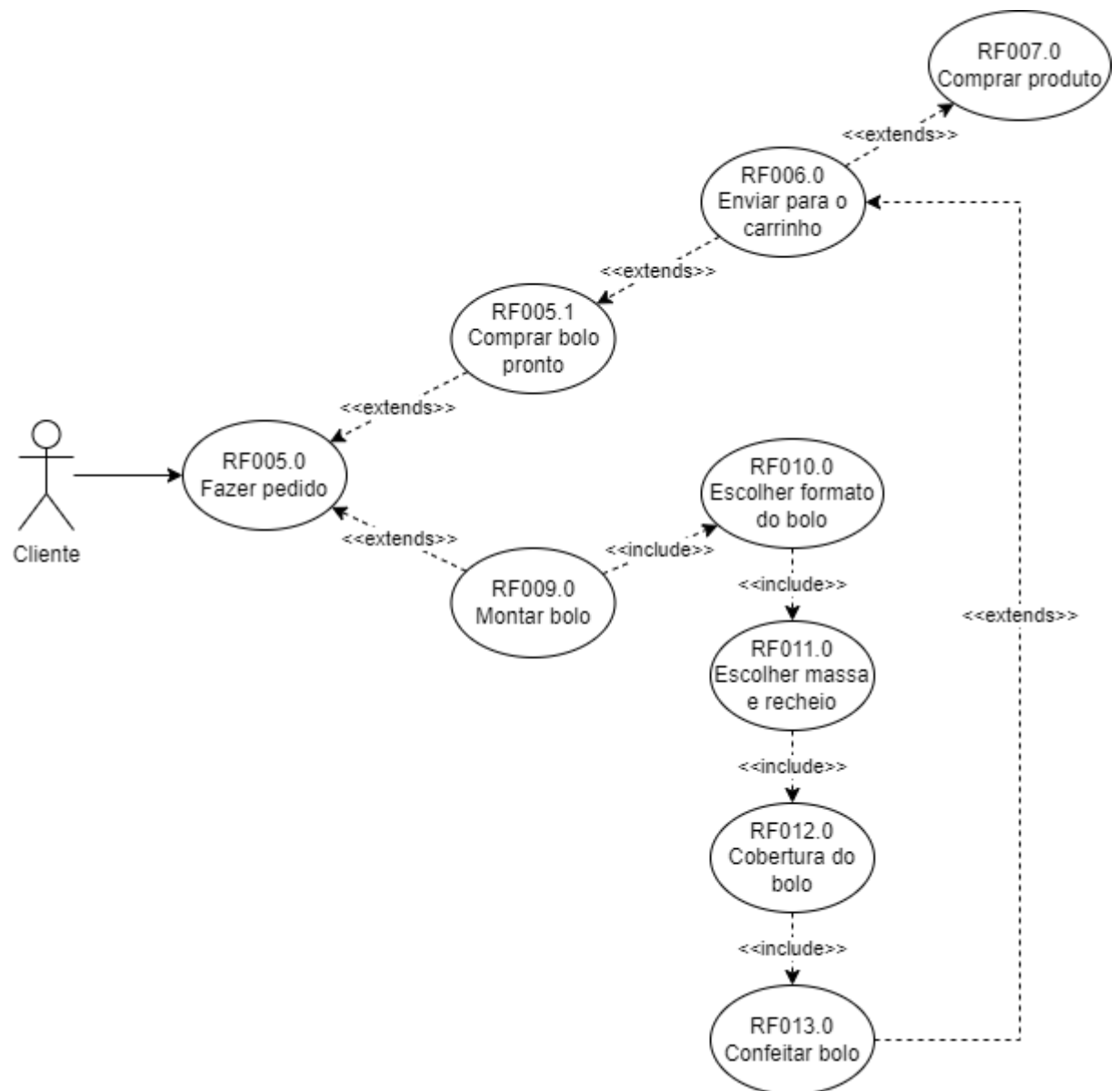


Figura 3 - DCU - [RF005]

[RF005.0]: Fazer Pedido

Comprar um bolo ou montar um bolo.

(X)Essencial ()Importante ()Desejável

[RF005.1]: Comprar bolo pronto

Comprar um bolo do catálogo.

(X)Essencial ()Importante ()Desejável

[RF006.0]: Enviar para o carrinho

Enviar para o carrinho de compras os bolos escolhidos.

(X)Essencial ()Importante ()Desejável

[RF009.0]: Montar bolo

Montar bolo.

(X)Essencial ()Importante ()Desejável

[RF010.0]: Escolher formato do bolo

Escolher se o formato é redondo ou quadrado.

(X)Essencial ()Importante ()Desejável

[RF011.0]: Escolher massa e recheio

Escolher sabor do bolo e o recheio.

(X)Essencial ()Importante ()Desejável

[RF012.0]: Cobertura do bolo

Escolher cobertura do bolo.

(X)Essencial ()Importante ()Desejável

[RF013.0]: Confeitar bolo

Escolher como quer confeitar o bolo.

(X)Essencial ()Importante ()Desejável

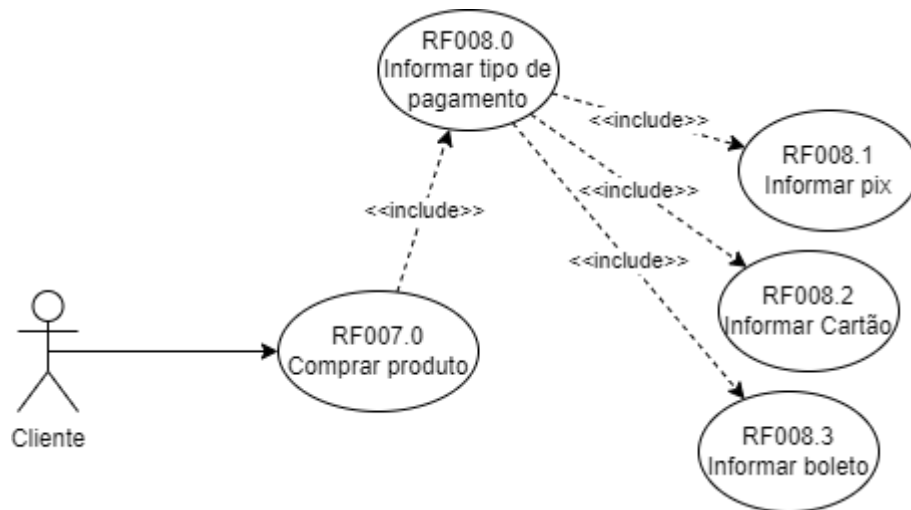


Figura 4 - DCU - [RF007]

[RF007.0]: Comprar produto

Após verificar os produtos do carrinho, o cliente de escolher em comprar o produto.

()Essencial (X)Importante ()Desejável

[RF008.0]: Informar tipo de pagamento

Escolher a forma de pagamento.

()Essencial (X)Importante ()Desejável

[RF008.1]: Informar pix

Preencher as informações do pix para o pagamento.

()Essencial (X)Importante ()Desejável

[RF008.2]: Informar Cartão

Preencher as informações do cartão para o pagamento.

()Essencial (X)Importante ()Desejável

[RF008.3]: Informar boleto

Preencher as informações do boleto para o pagamento.

(X)Essencial () Importante ()Desejável



Figura 5 - DCU - [RF014]

[RF014.0]: Altera as coisas no site

O Administrador que faz as alterações no site.

(X)Essencial () Importante ()Desejável

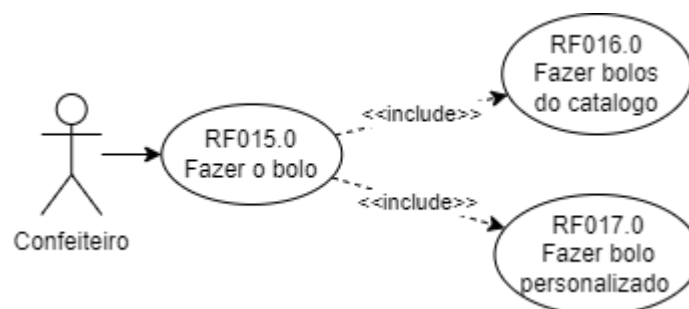


Figura 6 - DCU - [RF003]

[RF015.0]: Fazer o bolo

O confeiteiro faz o bolo.

()Essencial () Importante (X)Desejável

[RF016.0]: Fazer bolos do catálogo

O Confeiteiro faz os bolos do catálogo, que são bolos pré-definidos.

(X)Essencial () Importante ()Desejável

[RF017.0]: Fazer bolo personalizado

O Confeiteiro faz os bolos personalizados do cliente.

(X)Essencial () Importante () Desejável

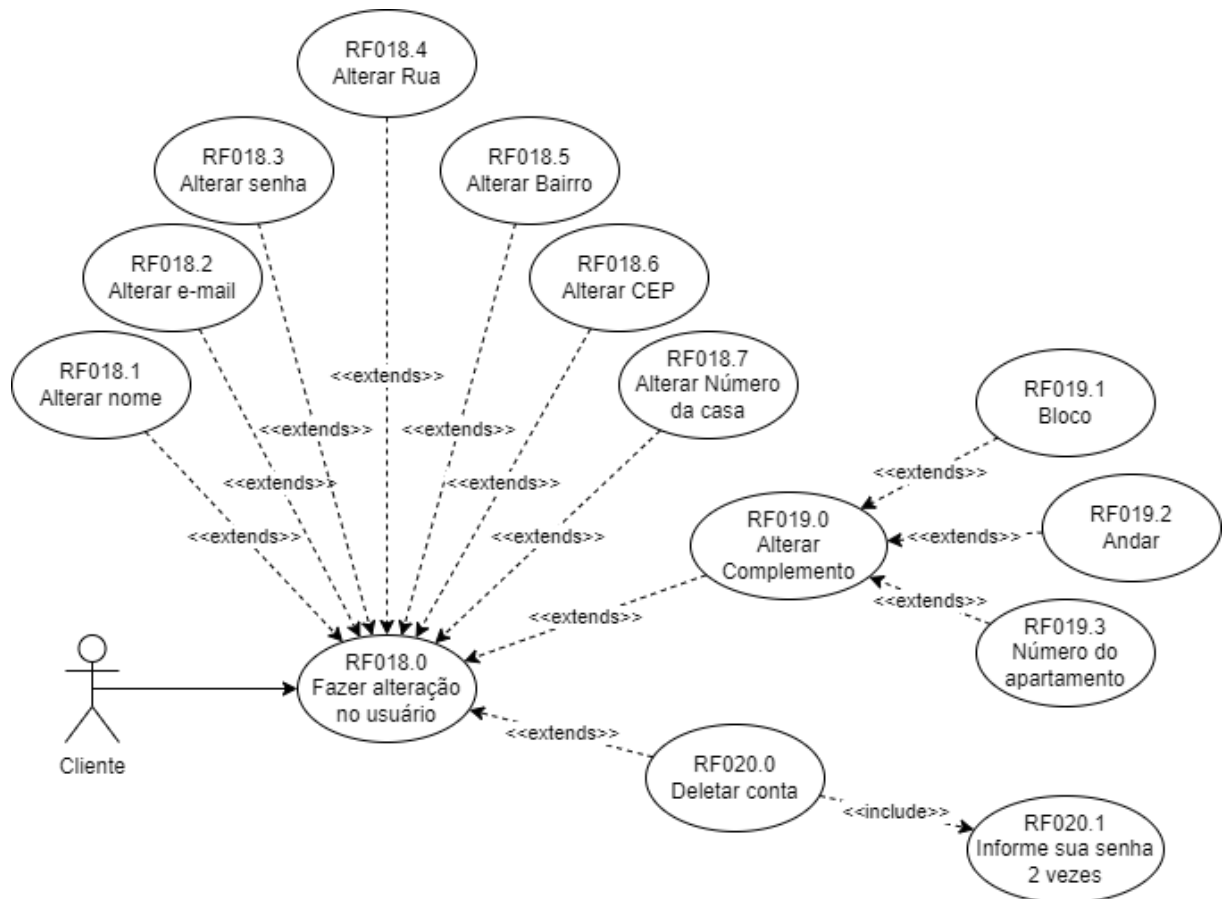


Figura 7 - DCU - [RF003]

[RF018.0]: Fazer alteração no usuário

Caso o cliente queira alterar algo em sua conta.

(X)Essencial () Importante () Desejável

[RF018.1]: Alterar nome

Alterar o nome de usuário.

(X)Essencial () Importante () Desejável

[RF018.2]: Alterar e-mail

Alterar o e-mail de usuário.

(X)Essencial ()Importante ()Desejável

[RF018.3]: Alterar senha

Alterar a senha de usuário.

(X)Essencial ()Importante ()Desejável

[RF018.4]: Alterar Rua

Alterar a rua da conta do usuário.

(X)Essencial ()Importante ()Desejável

[RF018.5]: Alterar Bairro

Alterar o Bairro da conta do usuário.

(X)Essencial ()Importante ()Desejável

[RF018.6]: Alterar CEP

Alterar o CEP da conta de usuário.

(X)Essencial ()Importante ()Desejável

[RF018.7]: Alterar Número da casa

Alterar o número da casa, da conta do usuário.

(X)Essencial ()Importante ()Desejável

[RF019.0]: Alterar Complemento

Alterar o complemento, da conta do usuário.

(X)Essencial ()Importante ()Desejável

[RF019.1]: Bloco

Alterar o bloco, da conta do usuário.

(X)Essencial ()Importante ()Desejável

[RF019.2]: Andar

Alterar o andar, da conta do usuário.

(X)Essencial ()Importante ()Desejável

[RF019.3]: Número do apartamento

Alterar o número do apartamento, da conta do usuário.

(X)Essencial ()Importante ()Desejável

[RF020.0]: Deletar conta

Caso o cliente queira deletar sua conta.

(X)Essencial ()Importante ()Desejável

[RF020.1]: Informe sua senha 2 vezes.

Para confirmar que deseja deletar sua conta, é preciso que coloque sua senha duas vezes.

(X)Essencial ()Importante ()Desejável

3. PLANEJAMENTO (CRONOGRAMA GRAFICO GANTT)

Planejamento e gestão do tempo e recursos humanos.

O backlog do projeto está dividido em atividades e segue apresentado utilizando um gráfico Gantt.

		Dias/Fevereiro													Dias/Março																		
Atividades	Recursos	14	15	16	17	18	21	22	23	24	25	2	3	4	7	8	9	10	11	14	15	16	17	18	21	22	23	24	25	28	29	30	31
Levantar os requisitos Funcionais	2 ⁺																																
Elaborar o DCU	2 ⁺																																
Elaborar o Diagrama de classe	2 ⁺ , 3 ⁺																																
Elaborar o Documento do Projeto	2 ⁺																																
Elaborar o Prototipo	2 ⁺ , 3 ⁺ , 4 ⁺																																
Programação do Backend	3 ⁺ , 5 ⁺																																
Programação do FrontEnd	1 ⁺ , 2 ⁺ , 3 ⁺ , 4 ⁺																																
Programação do Mobile	1 ⁺																																
Fazer o Cronograma	2 ⁺																																
Elaborar o Dicionario de Dados(JSON)																																	
Atualizar e concluir o documento do Projeto	2 ⁺																																

Este documento apresenta apenas uma versão intermediária do planejamento, durante a execução e controle sob a metodologia Scrum, a cada Sprint o cronograma foi atualizado.

Figura 8 - Cronograma - Gantt

Legenda	
Prazo Esgotado	
Concluído com atraso	
Concluído no prazo	
Planejado	
Grupo	
Felipe	1
Gabriel	2
Matheus	3
Victoria	4
Rodrigo	5
Ajuda	
Responsavel principal	+
Responsavel intermediario	=
Responsavel em ajuda	-

Figura 9 - Legenda – Gantt

3.1. PLANEJAMENTO DE CUSTOS

Para calcular os custos deste projeto basta somar as horas dedicadas conforme o cronograma, sabendo que são quatro horas por dia útil, todos os dias registrados são considerados dias úteis, e multiplicar por um valor estipulado por hora, como os valores monetários são voláteis. A seguir temos o valor conforme foi calculado o custo.

Felipe 80 horas trabalhadas, Gabriel 80 horas trabalhadas, Matheus 80 horas trabalhadas, Victoria 80 horas trabalhadas e Rodrigo 10 horas trabalhadas, totalizando 330 horas de projeto, caso cada recurso custasse R\$10,00 por hora o custo total deste projeto seria de: R\$ 3.300,00.

4. PROTÓTIPO

As imagens a seguir foram geradas utilizando a técnica de prototipagem para apresentar um esboço das telas do site da internet e do aplicativo.

4.1. PROTÓTIPO DO SITE DA INTERNET

As imagens mostram a sequência de telas que um usuário do site deve acessar para usufruir de todas as funcionalidades da aplicação.

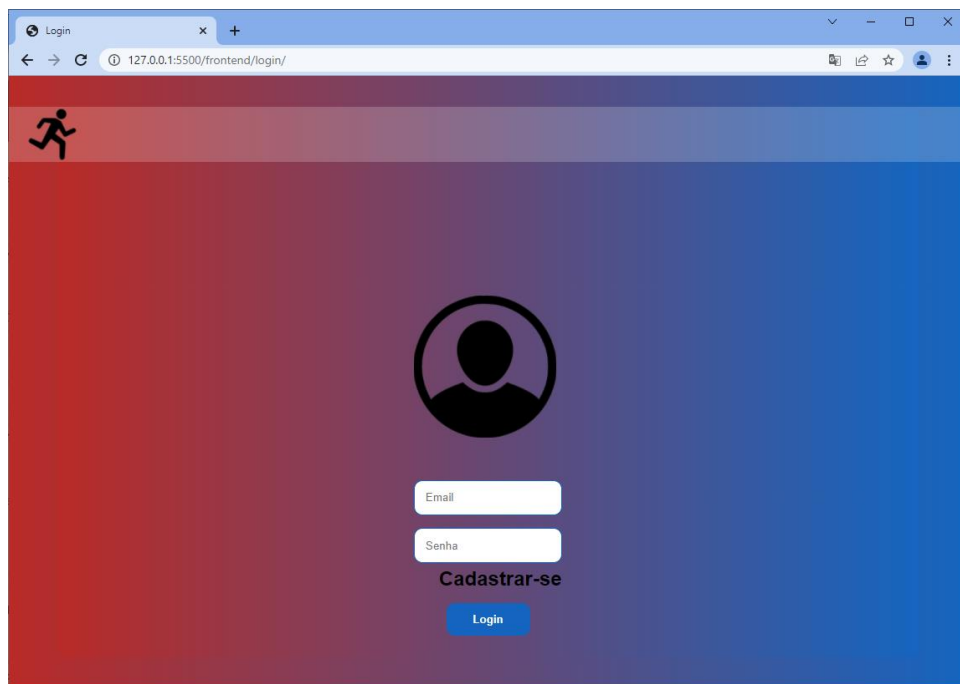


Figura 10 - Tela de Login

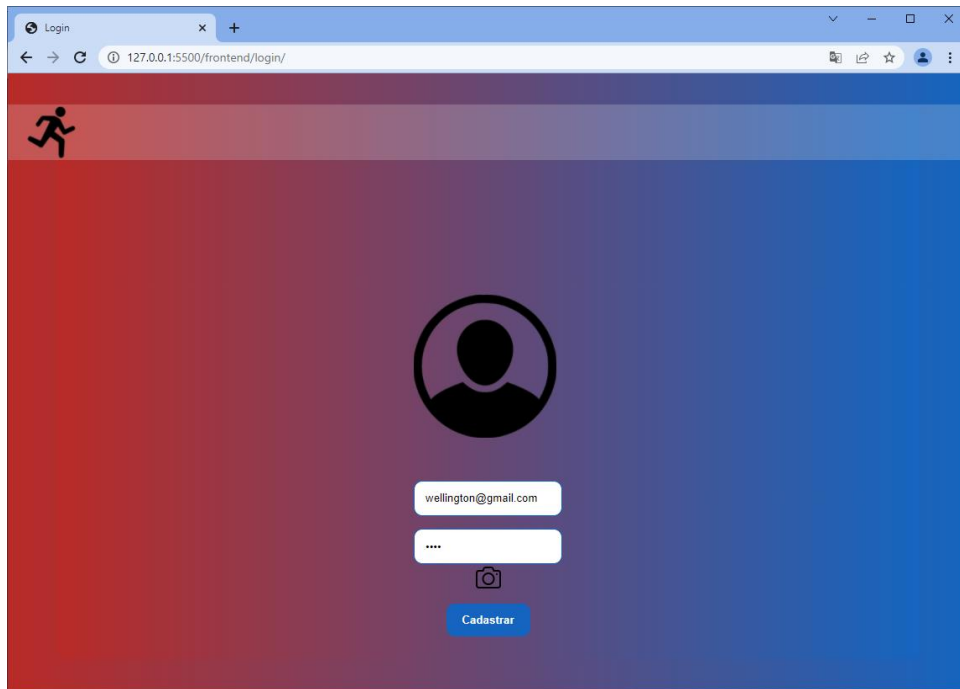


Figura 5 - Tela de Cadastro

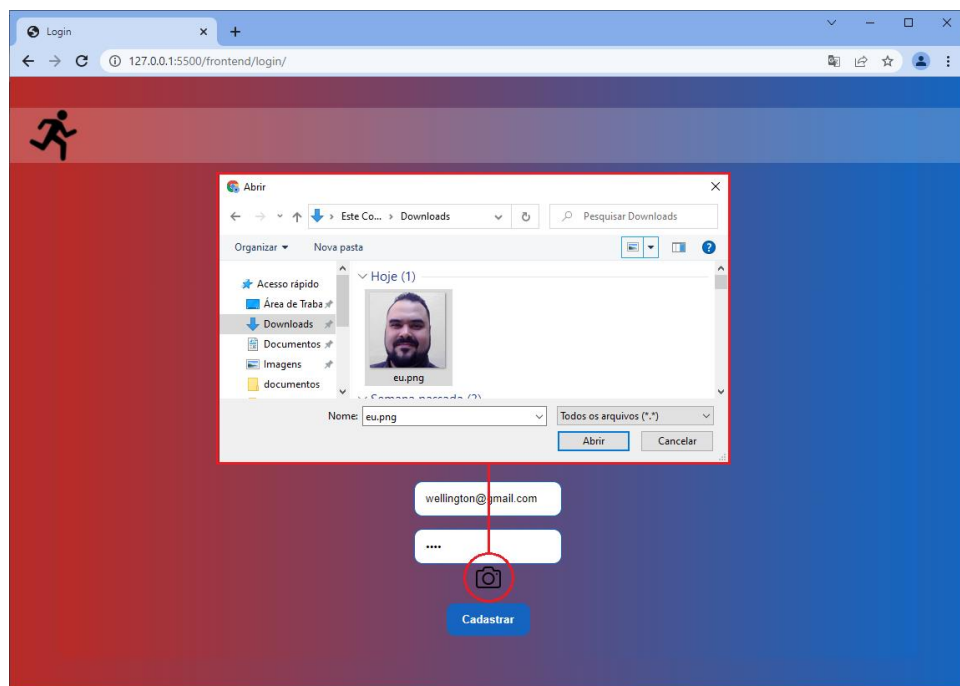


Figura 6 - Tela de Cadastro Inserindo Imagem de Perfil

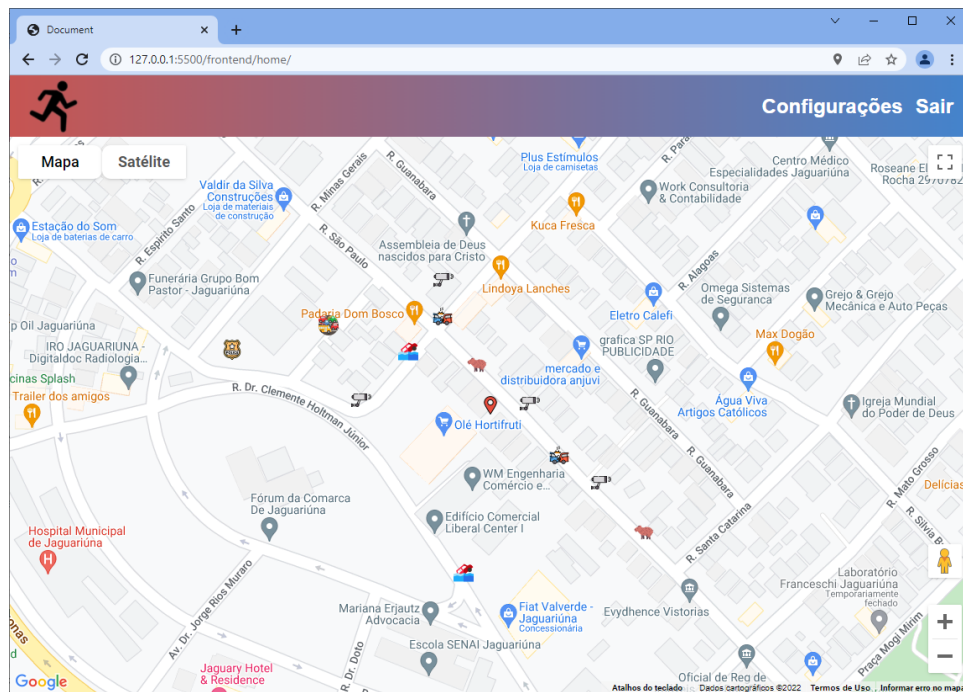


Figura 7 - Tela Principal (Home)

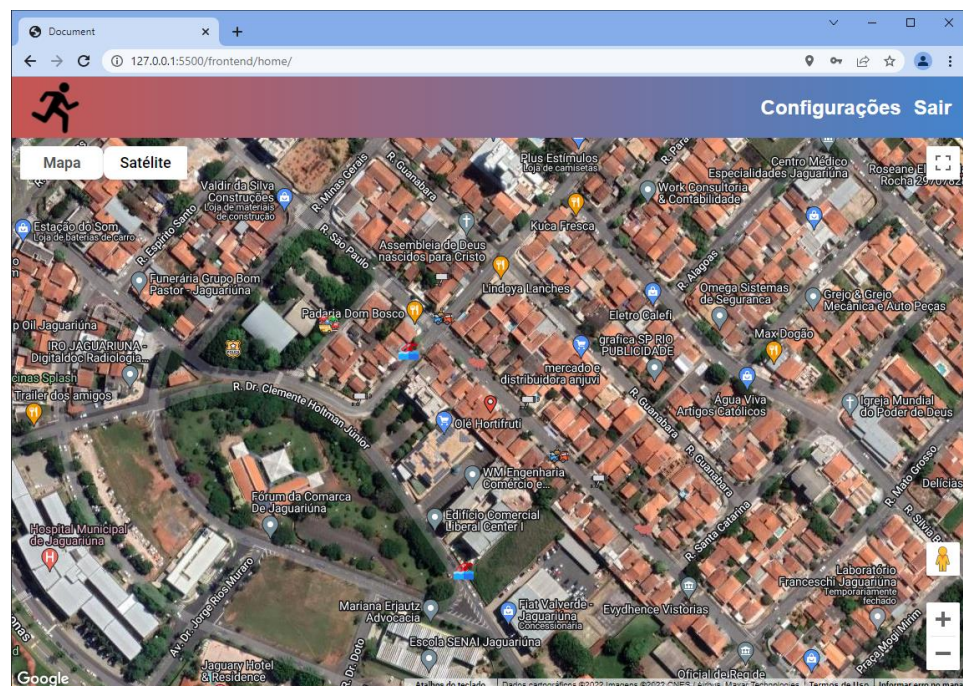


Figura 8 - Tela Principal (Home - Satellite)

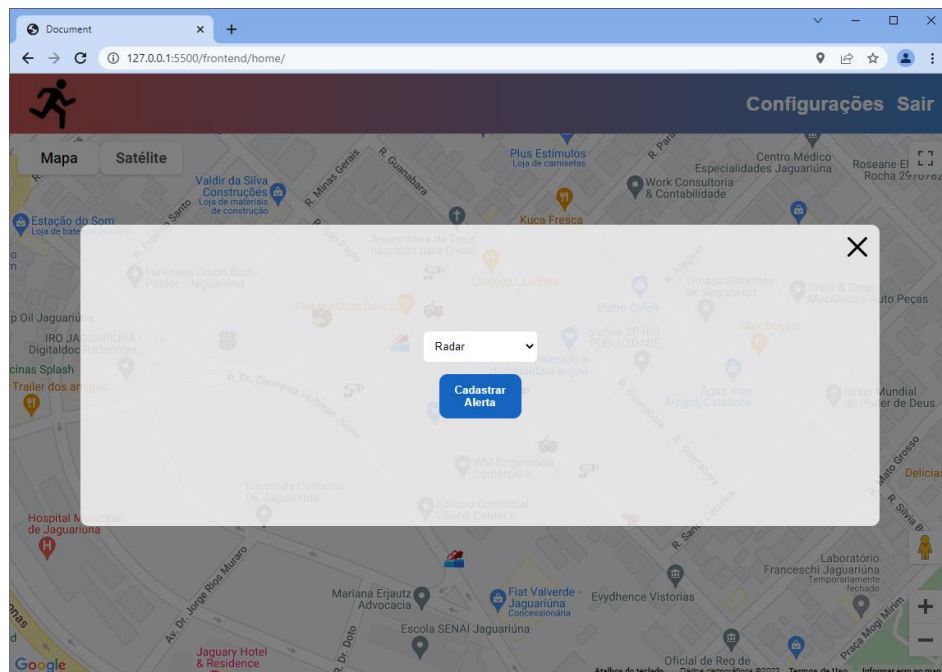


Figura 9 - Inserindo um novo alerta

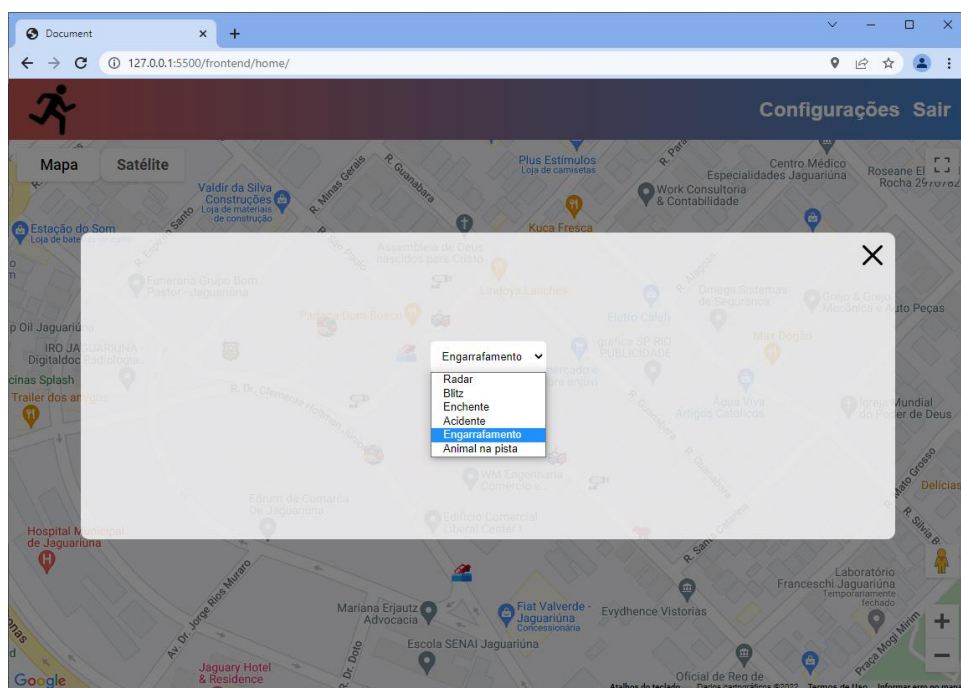


Figura 10 - Escolhendo o tipo de alerta

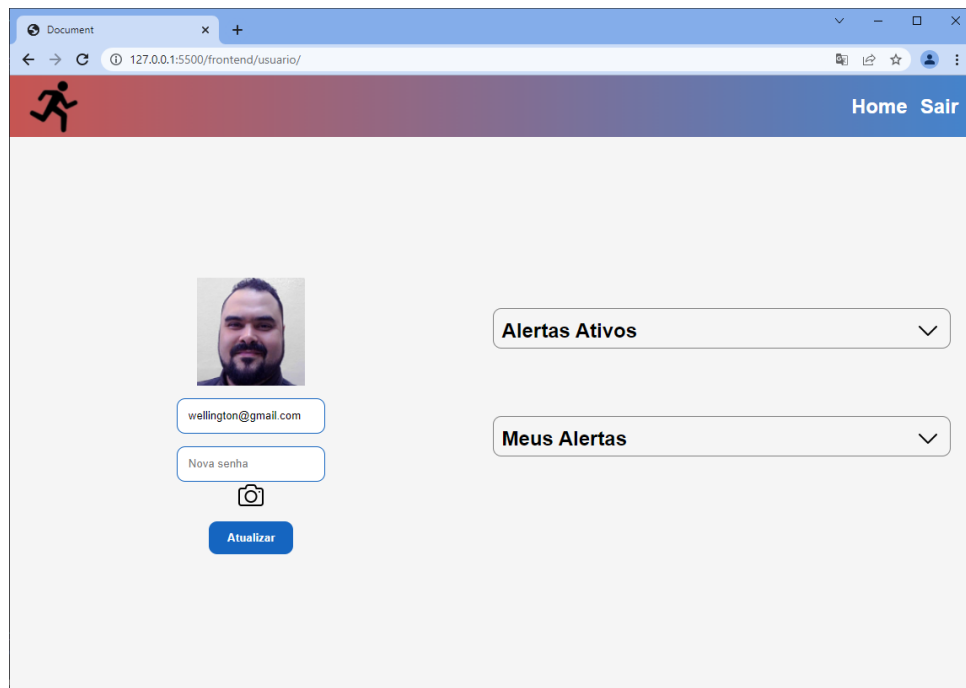


Figura 11 - Alterar o Perfil do Usuário

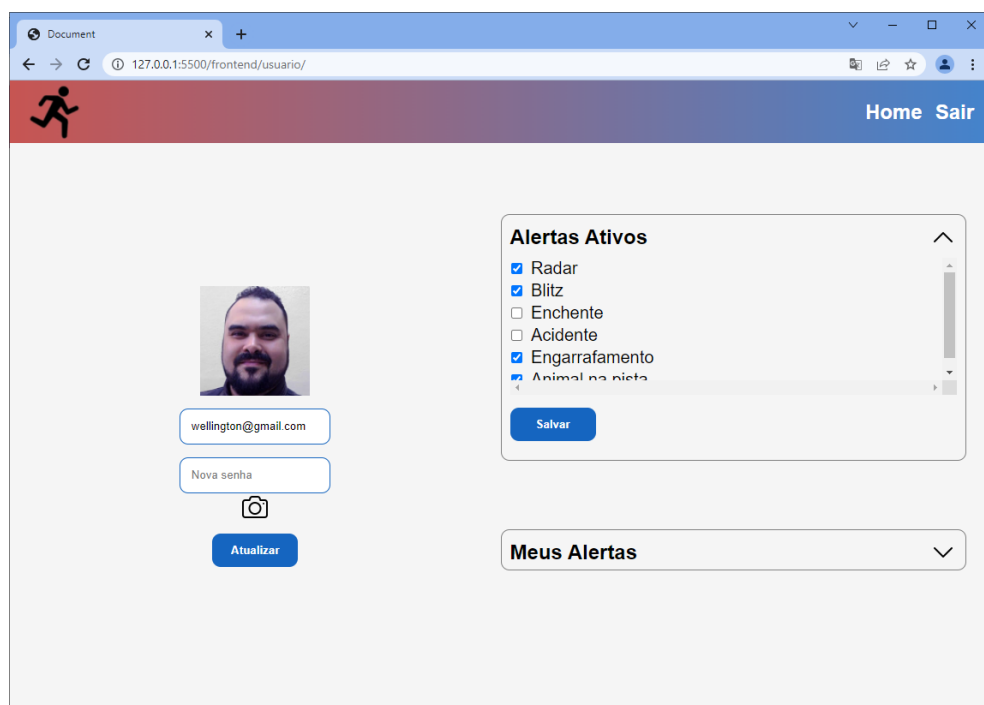


Figura 12 - Escolhendo os alertas ativos

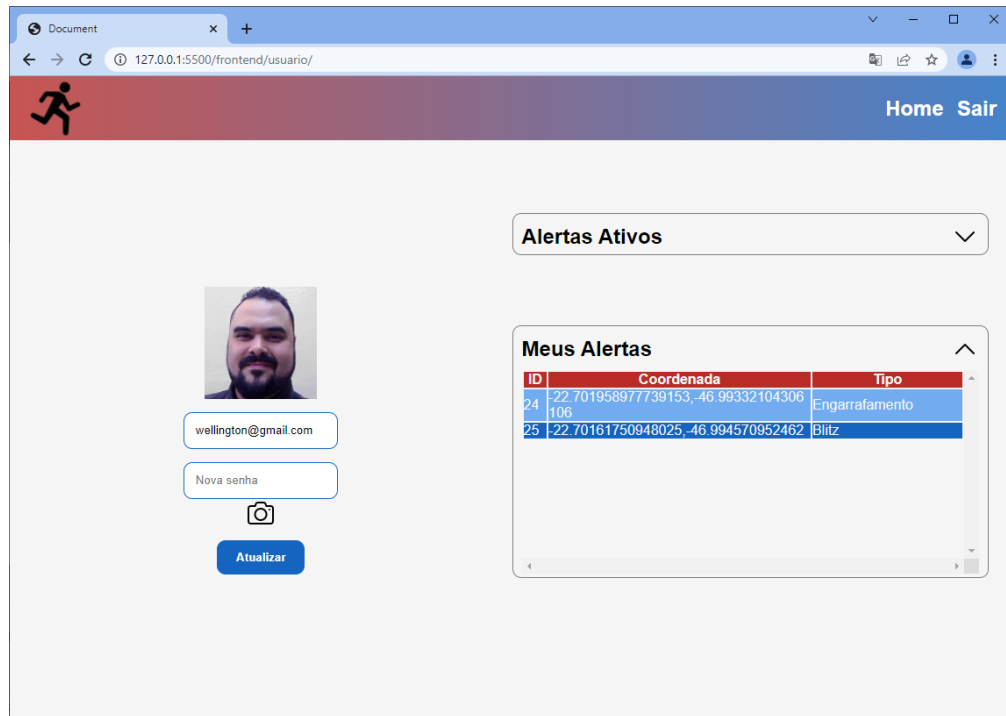


Figura 13 - Listando os alertas gerados

4.2. PROTÓTIPO DO APLICATIVO MÓVEL

As imagens mostram a sequência de telas que um condutor deve acessar para usufruir de todas as funcionalidades do aplicativo móvel.

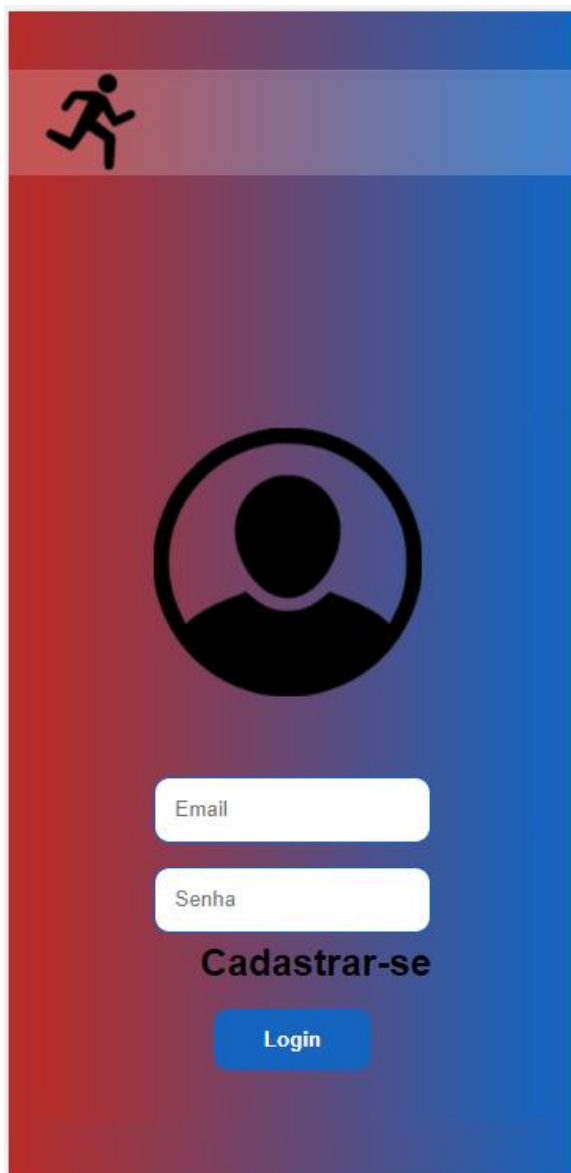


Figura 14 - Login App

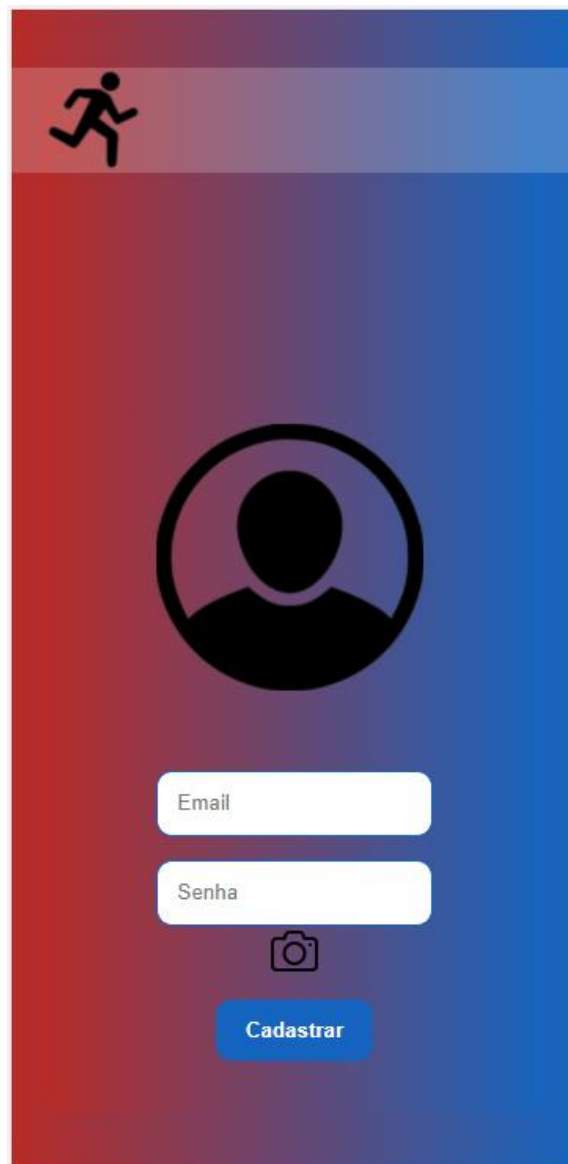


Figura 15 - Cadastro App

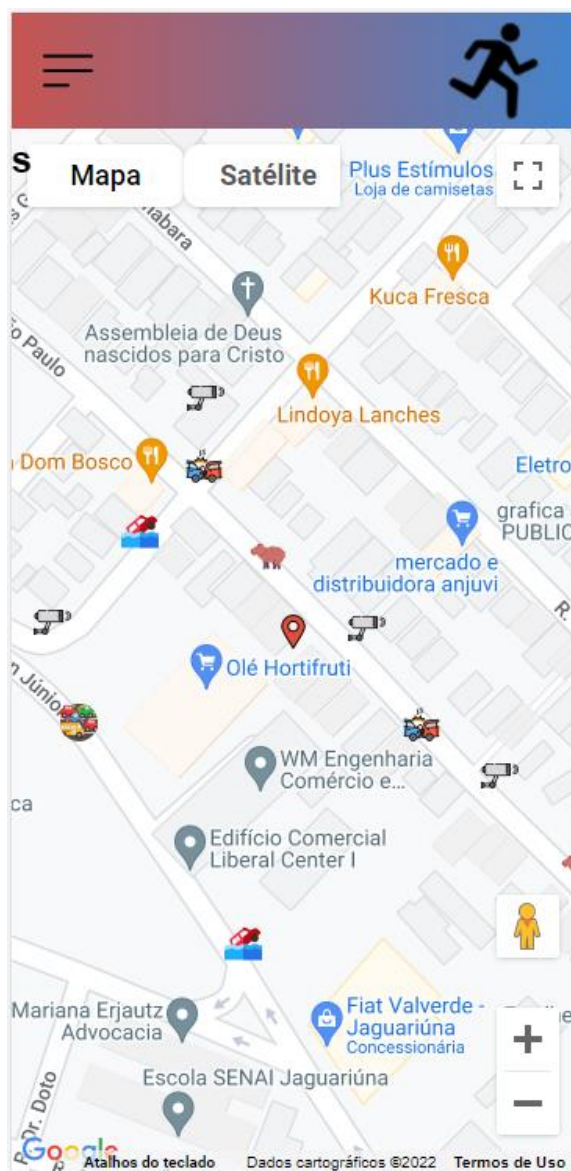


Figura 16 - Home App



Figura 17 - Home Satellite App

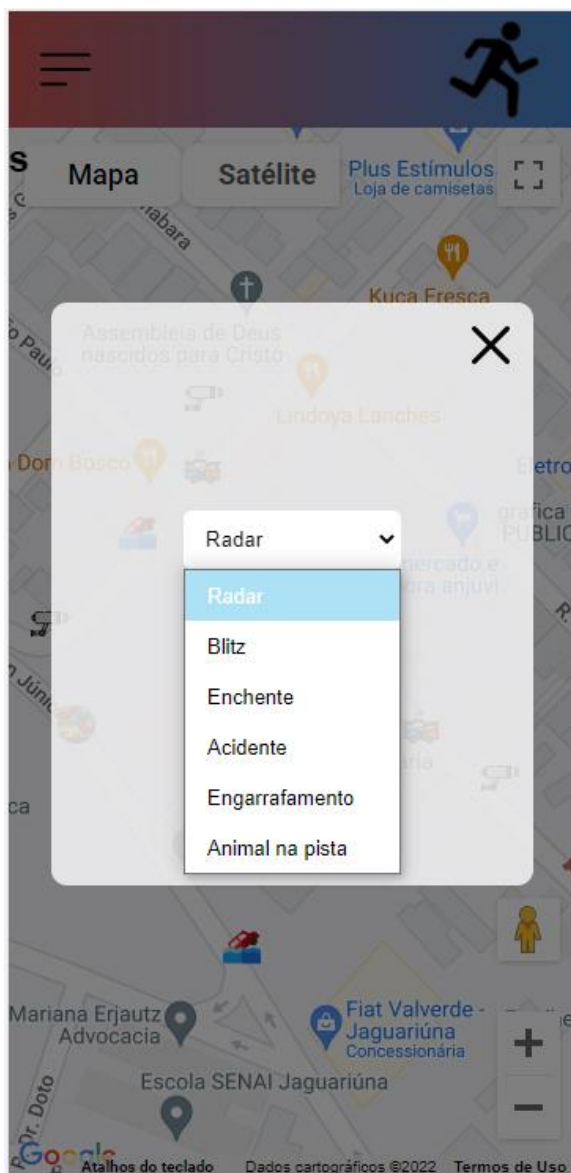


Figura 18 - Escolher alerta App

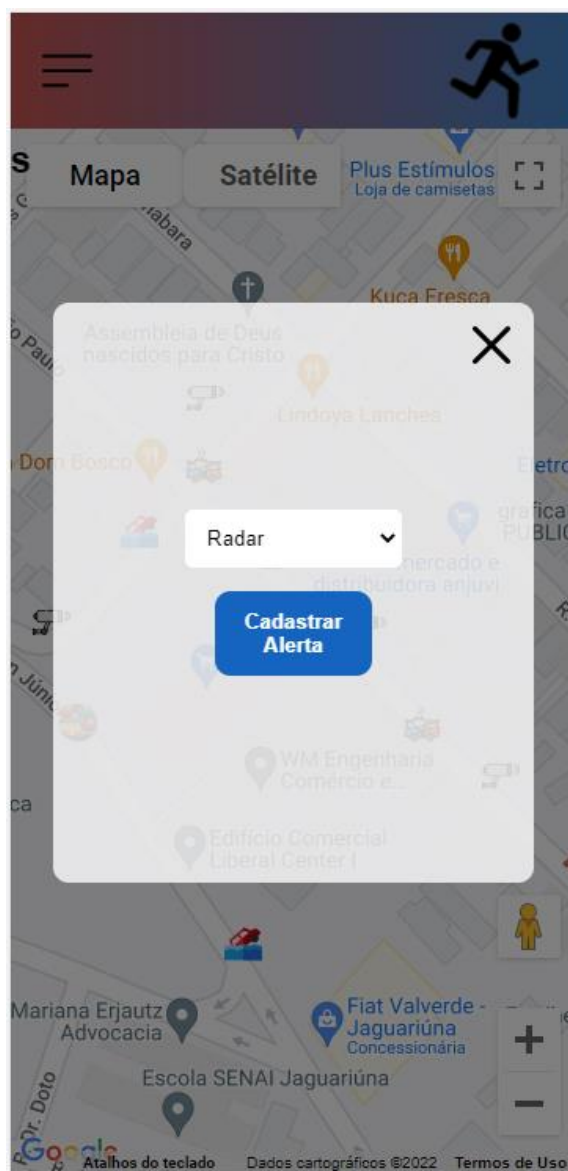


Figura 19 - Inserir alerta App

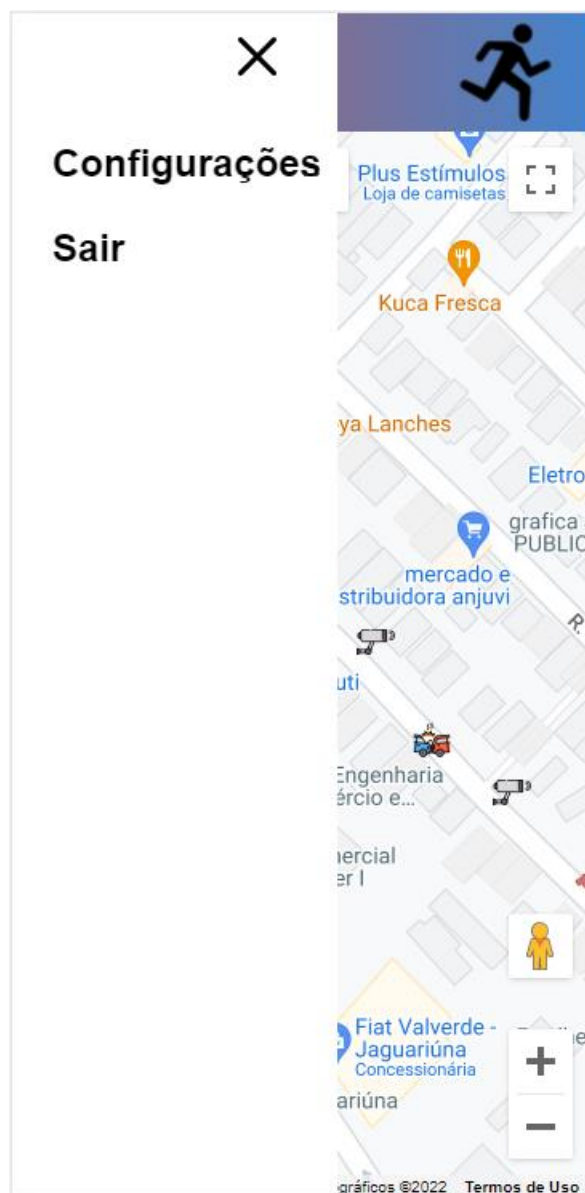


Figura 20 - Menu App

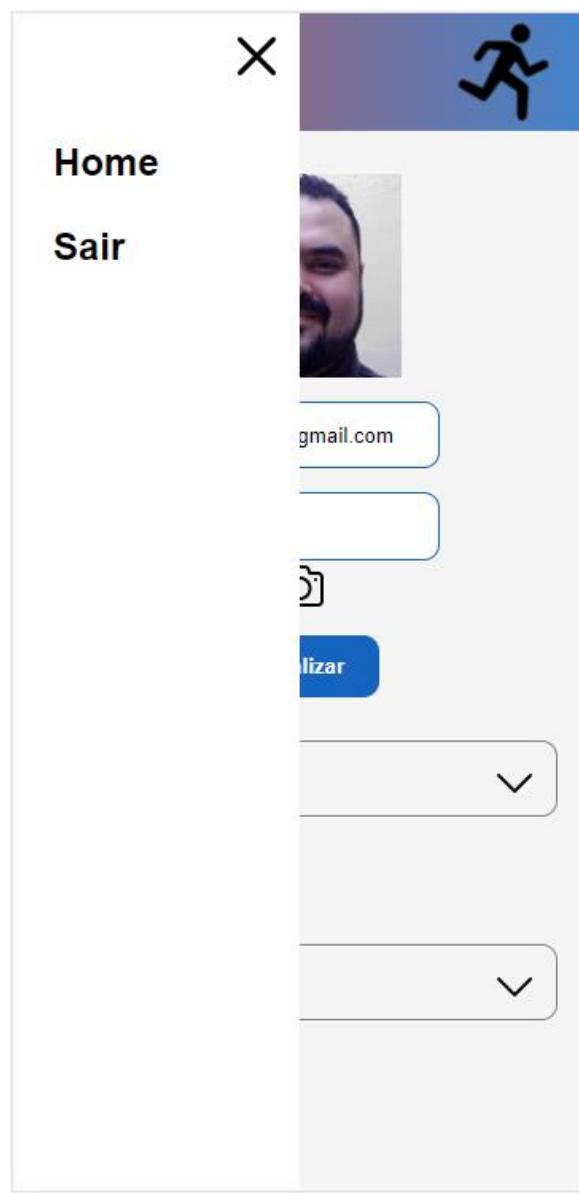


Figura 21 - Menu voltar App



Figura 22 - Perfil App

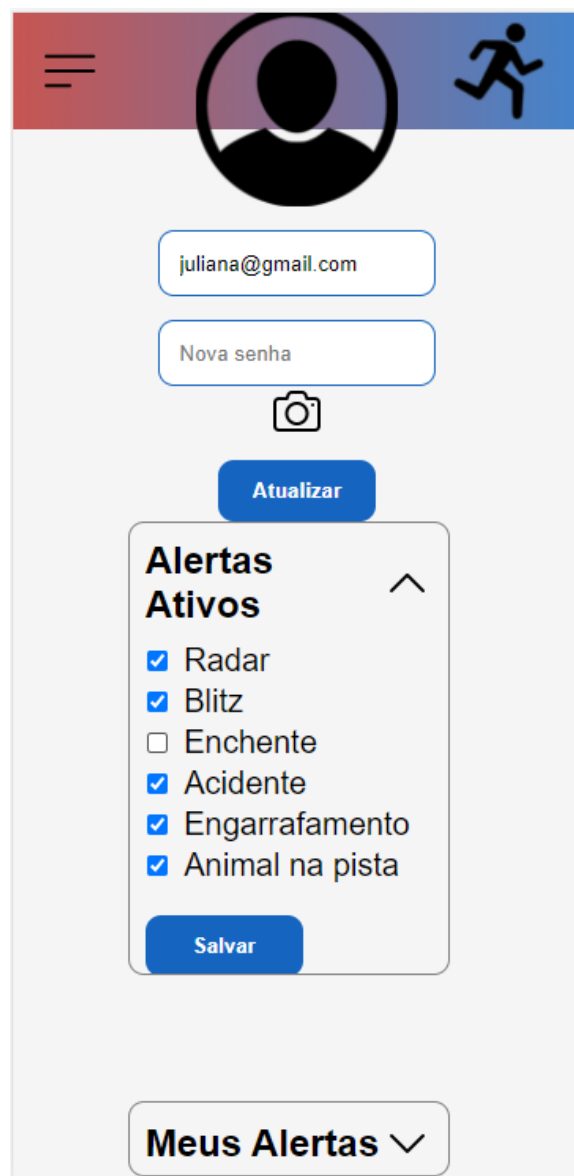


Figura 23 - Alertas ativos App



Figura 24 - Listar alertas App

As legendas das imagens descrevem as funcionalidades básicas do site e do aplicativo porém os requisitos estão melhor descritos no tópico dois sobre o escopo e requisitos funcionais do projeto.

5. PLANEJAMENTO DO BANCO DE DADOS E DA API

O backend deste projeto se comporta como uma API, fornecendo dados tanto para o frontend quanto para o aplicativo móvel, esta sessão documenta o projeto do banco de dados através do MER (Modelo Entidade e Relacionamento) e DER (Diagrama de Entidade e Relacionamento) conceitual e lógico.

Figura 25 - Modelo Conceitual

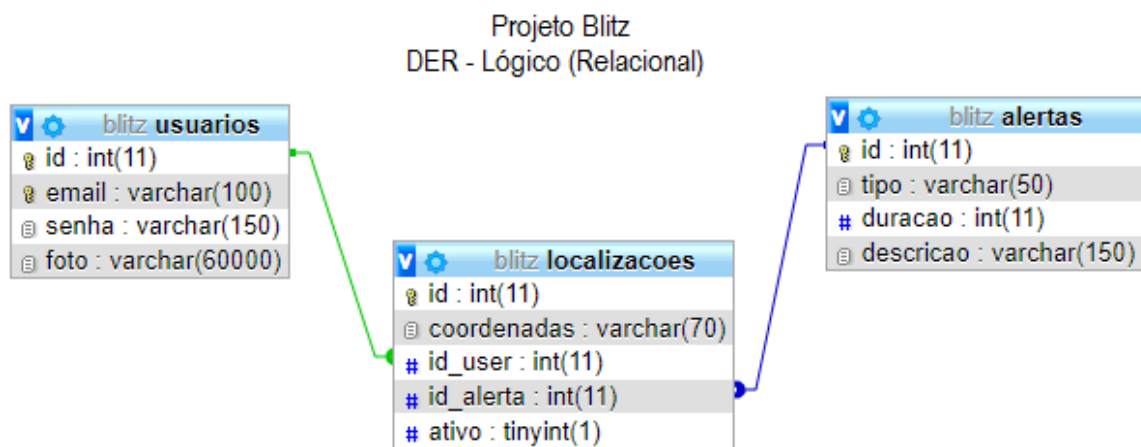


Figura 26 - Modelo Lógico

O diagrama de classes a seguir apresenta o projeto das classes da camada modelo do padrão de projetos MVC (Modelo Visão e Controle) do backend ou API.

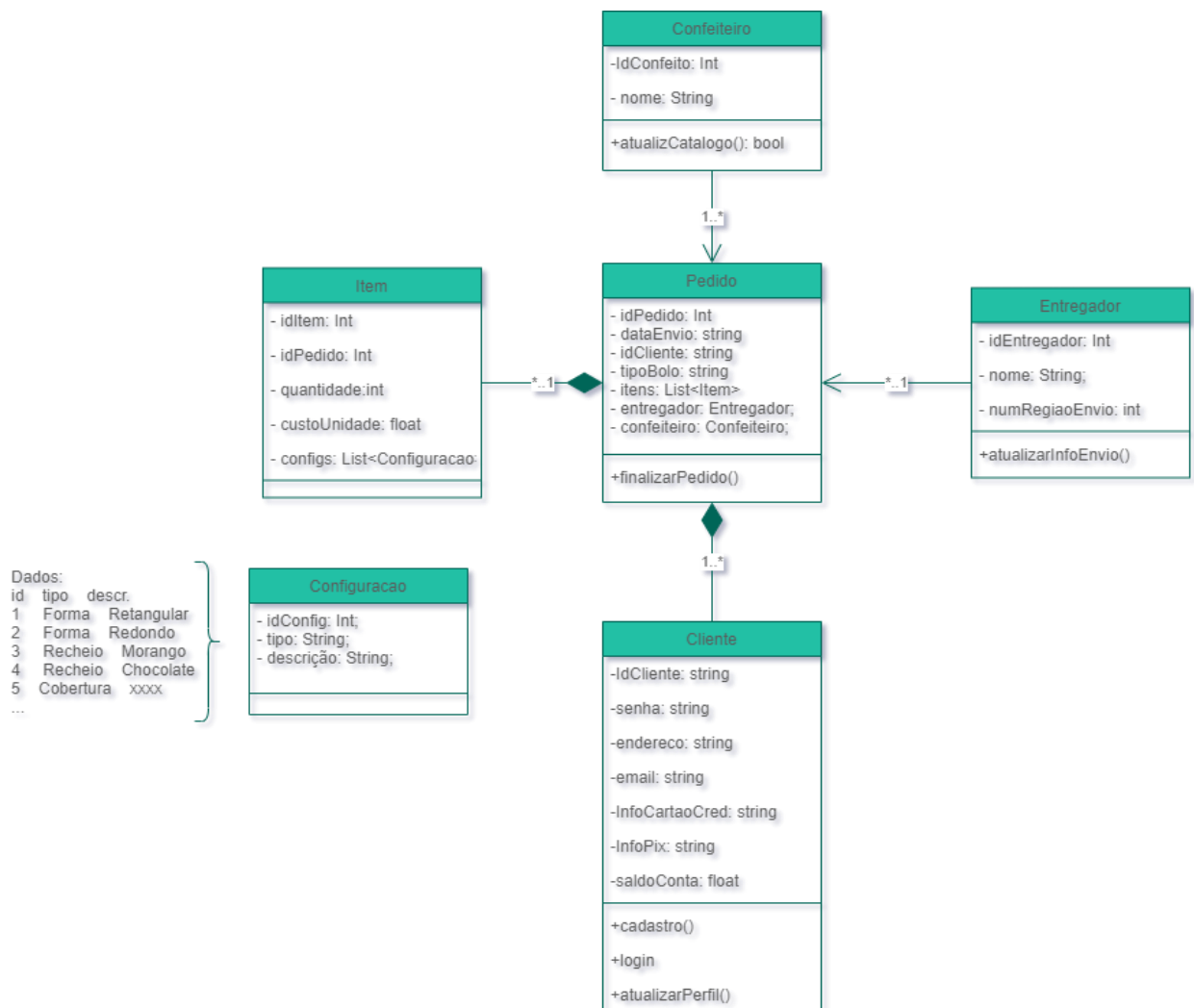


Figura 27 - Diagrama de Classes

6. DETALHES DA EXECUÇÃO E CONTROLE

O projeto foi desenvolvido em quatro semanas e meia de aula, juntamente com demonstrações práticas com os alunos da turma do terceiro módulo, após a execução deste projeto modelo, os alunos foram separados em grupos com três integrantes e foi proposto um pré-projeto de conclusão do curso com temas diversificados.

Foi proposto para o pré-projeto, como forma de integração maior da turma, que um grupo seria responsável por desenvolver a camada de backend e banco de dados do outro grupo, para isso o grupo principal do projeto deve fazer a documentação completa do projeto e utilizar as ferramentas de diagrama de classes e dicionário de dados para registrar quais informações necessita que a API / backend deve receber e fornecer.

A seguir segue o dicionário de dados no formato JSON que é o padrão de dados oficial de troca de informações entre APIs.

Dicionário de dados da classe Alerta:

```
{
  "01-verbo": "READ",
  "01-rota": "/alerta",
  "01-status": 200,
  "01-response": [
    {
      "id": 1,
      "tipo": "Radar",
      "duracao": 180,
      "descricao": "Enchente"
    },
    {
      "id": 2,
      "tipo": "Blitz",
      "duracao": 60,
      "descricao": "Blitz"
    },
    {
      "id": 3,
      "tipo": "Enchente",
      "duracao": 60,
      "descricao": "Enchente"
    },
    {
      "id": 4,
      "tipo": "Engarrafamento",
      "duracao": 60,
      "descricao": "Congestionamento/Engarrafamento"
    }
  ]
}
```

```
        "id": 5,
        "tipo": "Acidente",
        "duracao": 60,
        "descricao": ""
    },
    {
        "id": 6,
        "tipo": "Animal na pista",
        "duracao": 60,
        "descricao": ""
    }
],
"02-verbo": "READ",
"02-rota": "/alerta/1",
"02-status": 200,
"02-response": [
    {
        "id": 1,
        "tipo": "Radar",
        "duracao": 180,
        "descricao": "Enchente"
    }
],
"03-verbo": "POST",
"03-rota": "/alerta",
"03-request": {
    "tipo": "Depressao na pista",
    "duracao": 60,
    "descricao": ""
},
"03-true-status": 201,
"03-response": {
    "id": 7,
    "tipo": "Depressao na pista",
    "duracao": 60,
    "descricao": ""
},
"03-false-status": 400,
"04-verbo": "PUT",
"04-rota": "/alerta/7",
"04-request": {
    "tipo": "Depressao na Pista",
    "duracao": 180,
    "descricao": "Depressao"
},
"04-true-status": 200,
"04-response": [
    {
        "id": 1,
        "tipo": "Enchente",
        "duracao": 180,
        "descricao": "Enchente"
    }
],
"04-false-status": 404,
"05-verbo": "DELETE",
"05-rota": "/alerta/7",
"05-true-status": 200,
"05-response": {
```

```

        "id": "7"
    },
    "05-false-status": 404
}

```

Dicionário de dados da classe Usuário:

```

{
  "01-verbo": "READ",
  "01-rota": "/usuario",
  "01-status": 200,
  "01-response": [
    {
      "id": 1,
      "email": "beltrano@gmail.com",
      "senha": "81dc9bdb52d04dc20036dbd8313ed055",
      "foto": ""
    },
    {
      "id": 2,
      "email": "wellington@gmail.com",
      "senha": "81dc9bdb52d04dc20036dbd8313ed055",
      "foto": "data:image/png;base64,/9j/2wCEAAgGB..."
    },
    {
      "id": 3,
      "email": "juliana@gmail.com",
      "senha": "81dc9bdb52d04dc20036dbd8313ed055",
      "foto": ""
    }
  ],
  "02-verbo": "READ",
  "02-rota": "/usuario/1",
  "02-status": 200,
  "02-response": {
    "id": 1,
    "email": "beltrano@gmail.com",
    "senha": "81dc9bdb52d04dc20036dbd8313ed055",
    "foto": ""
  },
  "03-verbo": "POST",
  "03-rota": "/usuario",
  "03-request": {
    "email": "juliana@gmail.com",
    "senha": "81dc9bdb52d04dc20036dbd8313ed055",
    "foto": ""
  },
  "04-status": 201,
  "04-verbo": "POST",
  "04-rota": "/login",
  "04-request": {
    "email": "juliana@gmail.com",
    "senha": "81dc9bdb52d04dc20036dbd8313ed055"
  },
  "04-true-status": 200,
  "04-response": {
    "id": 3,

```

```

    "email": "juliana@gmail.com",
    "foto": ""
  },
  "04-false-status": 404,
  "05-verbo": "PUT",
  "05-rota": "/usuario/1",
  "05-true-status": 200,
  "05-request": {
    "senha": "81dc9bdb52d04dc20036dbd8313ed055"
  },
  "05-false-status": 404,
  "06-verbo": "DELETE",
  "06-rota": "/usuario/1",
  "06-true-status": 200,
  "06-false-status": 404
}

```

Dicionário de dados da classe localização:

```

{
  "01-verbo": "GET",
  "01-rota": "/localizacao",
  "01-status": 200,
  "01-response": [
    {
      "id": 1,
      "coordenadas": "-22.701347063849923,-46.992755096958916",
      "ativo": true,
      "usuario": {
        "id": 1,
        "email": "oi@gmail.com"
      },
      "alertum": {
        "id": 1,
        "tipo": "Radar"
      }
    },
    {
      "id": 2,
      "coordenadas": "-22.701436142784136,-46.99265853743438",
      "ativo": true,
      "usuario": {
        "id": 1,
        "email": "oi@gmail.com"
      },
      "alertum": {
        "id": 2,
        "tipo": "Blitz"
      }
    },
    {
      "id": 3,
      "coordenadas": "-22.700377089482824,-46.995651882695",
      "ativo": true,
      "usuario": {
        "id": 1,
        "email": "oi@gmail.com"
      }
    }
  ]
}

```

```
    },
    "alertum": {
      "id": 6,
      "tipo": "Animal na pista"
    }
  },
  {
    "id": 4,
    "coordenadas": "-22.7019953592143,-46.99209527354125",
    "ativo": true,
    "usuario": {
      "id": 1,
      "email": "oi@gmail.com"
    },
    "alertum": {
      "id": 6,
      "tipo": "Animal na pista"
    }
  },
  {
    "id": 4,
    "coordenadas": "-22.7019953592143,-46.99209527354125",
    "ativo": true,
    "usuario": {
      "id": 1,
      "email": "oi@gmail.com"
    },
    "alertum": {
      "id": 5,
      "tipo": "Acidente"
    }
  },
  {
    "id": 5,
    "coordenadas": "-22.702420956029204,-46.99159638266448",
    "ativo": true,
    "usuario": {
      "id": 1,
      "email": "oi@gmail.com"
    },
    "alertum": {
      "id": 1,
      "tipo": "Radar"
    }
  }
],
"02-verbo": "GET",
"02-rota": "/localizacao/1",
"02-status": 200,
"02-response": [
  {
    "id": 1,
    "coordenadas": "-22.701347063849923,-46.992755096958916",
    "ativo": true,
    "usuario": {
      "id": 1,
      "email": "oi@gmail.com",
      "foto": ""
    }
  },

```

```
        "alertum": {
            "id": 1,
            "tipo": "Radar",
            "duracao": 180,
            "descricao": "Enchente"
        }
    },
    ],
    "03-verbo": "POST",
    "03-rota": "/localizacao",
    "03-request": {
        "coordenadas": "-12.123,-12.123",
        "id_user": 2,
        "id_alerta": 1,
        "ativo": true
    },
    "03-true-status": 201,
    "03-response": {
        "id": 11,
        "coordenadas": "-12.123,-12.123",
        "id_user": 2,
        "id_alerta": 1,
        "ativo": true
    },
    "03-false-status": 400,
    "04-verbo": "PUT",
    "04-rota": "/localizacao/1",
    "04-request": {
        "coordenadas": "-12.123,-12.123",
        "id_user": 2,
        "id_alerta": 1,
        "ativo": false
    },
    "04-true-status": 200,
    "04-response": [
        {
            "id": 1,
            "coordenadas": "-22.701347063849923,-46.992755096958916",
            "id_user": 1,
            "id_alerta": 1,
            "ativo": false
        }
    ],
    "04-false-status": 404,
    "05-verbo": "DELETE",
    "05-rota": "/localizacao/2",
    "05-true-status": 200,
    "05-response": {
        "id": "2"
    },
    "05-false-status": 404
}
```


7. RESULTADOS

O projeto foi apresentado em aula para os alunos do terceiro módulo no primeiro semestre do ano de 2022 e servirá de modelo para o TCC.

Os arquivos fonte deste projeto estão compartilhados no GitHub através do endereço de internet a seguir: <https://github.com/vick7000/Bolos.git> este é um repositório público para consulta.

Este projeto não foi hospedado em nenhum serviço privado, público ou de nuvem, apenas na rede local da sala de aula desta instituição de ensino.

BIBLIOGRAFIA

Brasil: DEVMEDIA, Disponível em: <<https://www.devmedia.com.br/orm-object-relational-mapper/19056>>, acessado em 08/03/2022.

Brasil: HostGator, Disponível em: <https://www.hostgator.com.br/blog/front-end-back-end-e-full-stack/?gclid=Cj0KCQiAmpyRBhC-ARIsABs2EA-oAai258HK5IS4M8nJy5tWkv0AVjVv2Ya3n9r-8jDyASrncN1vbiJU-aAqWaEALw_wcB>, acessado em 08/03/2022.

Brasil: tecnoblog, Disponível em: <<https://tecnoblog.net/responde/o-que-e-uma-api-guia-para-iniciantes/>>, acessado em 09/03/2022.

Brasil: nafergo, Disponível em: <<http://nafergo.github.io/manual-livre-animacao2d/index.html>>, acessado em 09/03/2022.