

PROJETO PARA CRIAÇÃO DE UMA BASE DE DADOS PARA A DEFESA CIVIL DE SÃO JOSE DOS CAMPOS

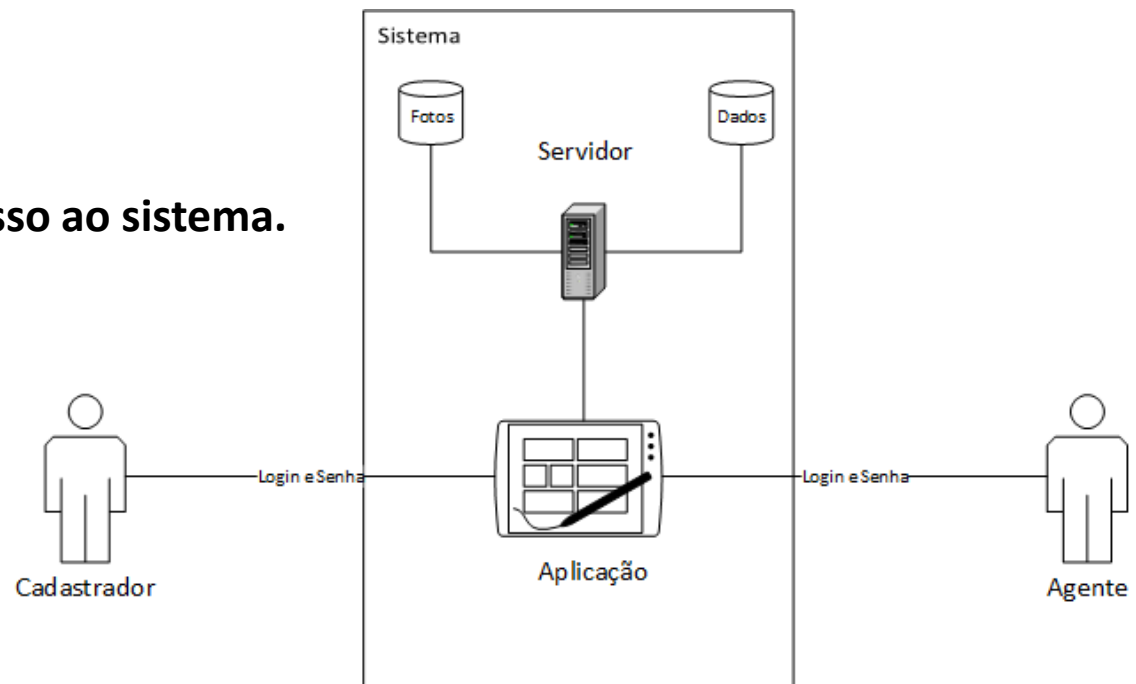
Aluno:

Aniro Correia Montenegro

Objetivo

- Criação de base de dados para a Defesa Civil de São José relacionado aos atendimentos prestados.

Figura 1- Diagrama de acesso ao sistema.

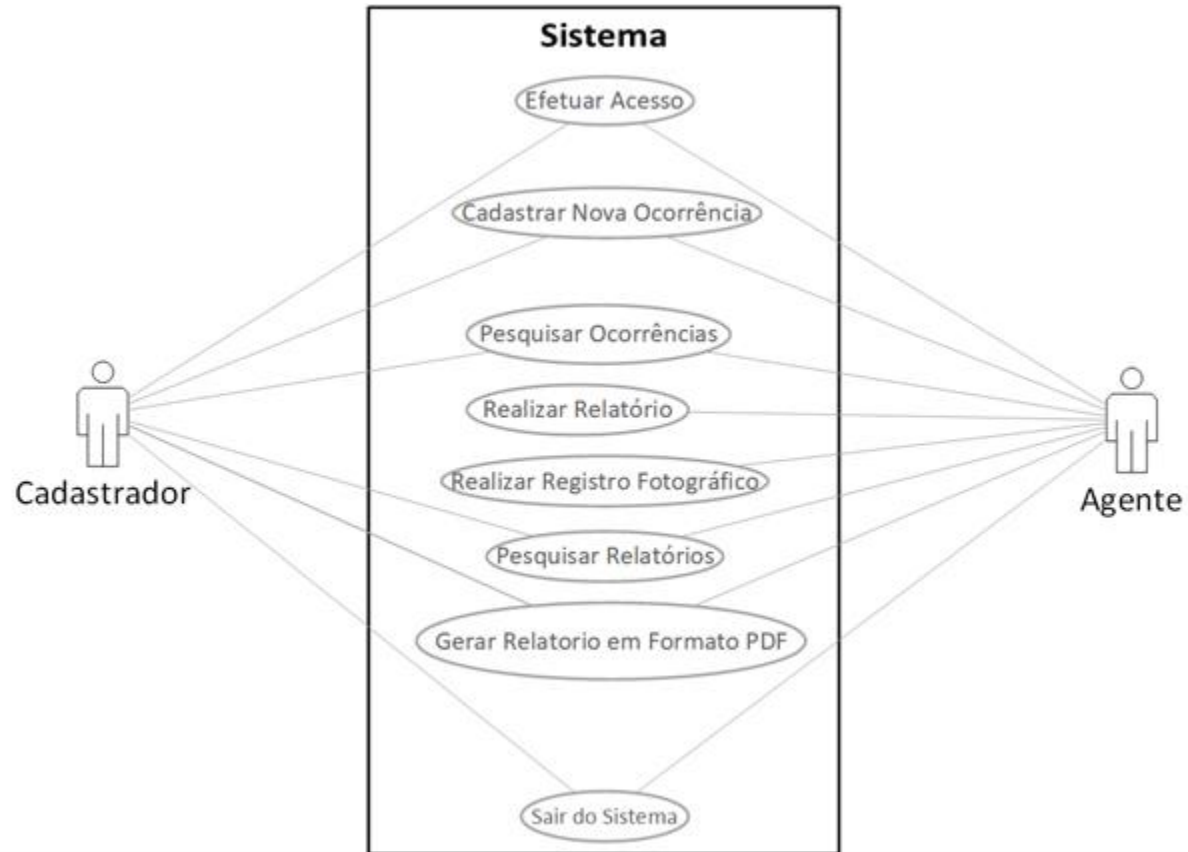


Requisitos

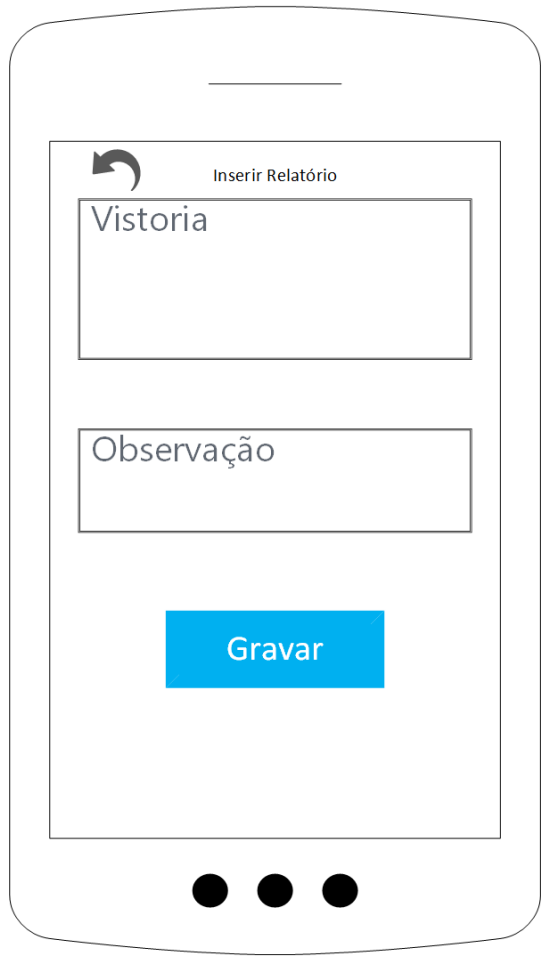
- Entrevistas
- Prototipagem
- Requisitos:
 - Acesso ao sistema
 - Despacho das solicitações
 - Atendimento nas localidades

Requisitos

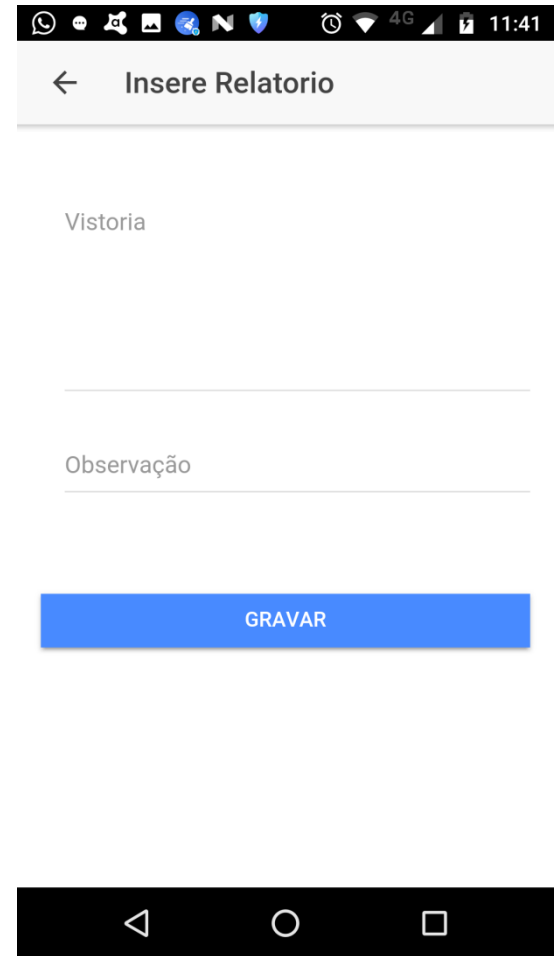
Figura 2-Diagrama de Caso de uso- Perfis Cadastrador e Agente



Requisitos



Protótipo de tela



Tela Final

Desenvolvimento

- Padrão MVC

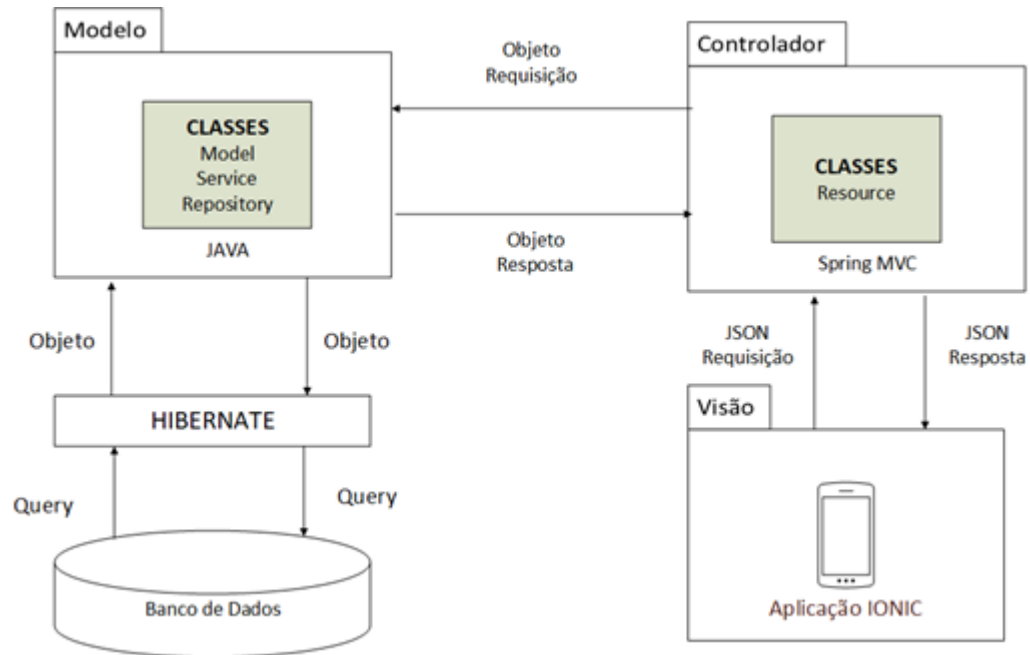


Figura 9- Diagrama do modelo MVC- Backend.

Desenvolvimento

- Banco de dados: MySQL
- S3 AWS Amazon
- Framework Spring
- Maven
- Hibernate
- Framework Ionic

Desenvolvimento

- Persistência.
 - Hibernate.
 - Mapeamento OR
 - Abstração SGBD

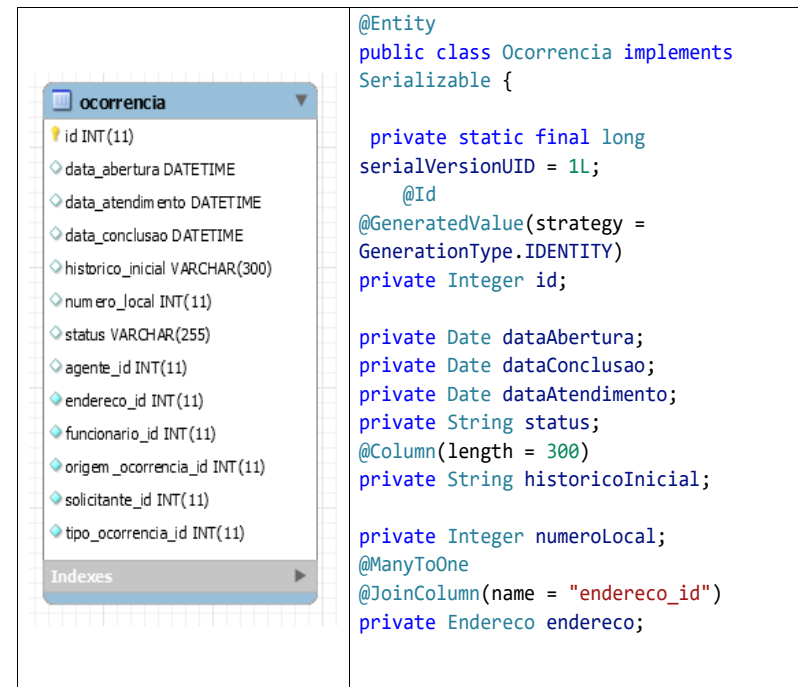


Figura 18– Relação objeto/relacional

Desenvolvimento

- Persistência.
 - Repository
 - Acesso aos dados.

```
package com.defesa.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.defesa.domain.Departamento;

@Repository
public interface DepartamentoRepository extends JpaRepository<Departamento, Integer>
{
}
```

Desenvolvimento

- Persistência.
 - Services
 - Regras de negócio

Figura 20- Busca de relatórios por funcionário

```
public List<RelatorioDTO> findAllFuncionario(Integer id) { // Busca Relatorios por
funcionario
    Funcionario func = funcService.find(id);
    List<Relatorio> rel = repo.findByFuncionario(func);
    List<RelatorioDTO> listDto = new ArrayList<RelatorioDTO>();
    for (Relatorio r : rel) {
        if (!(r.getTramitacao().getDescricao().equals("ARQUIVO"))) {
            listDto.add(new RelatorioDTO(r));
        }
    }
    return listDto;
}
```

Desenvolvimento

- Serviços
 - Comunicação
 - Rest

```
@RequestMapping(value="/abertas",method = RequestMethod.GET)
public ResponseEntity<List<OcorrenciaDTO>> listarAbertas() {

    List<Ocorrencia> list= service.findByStatus();

    List<OcorrenciaDTO> listDto = list.stream().map(obj ->
    new OcorrenciaDTO(obj)).collect(Collectors.toList());

    return ResponseEntity.ok().body(listDto);
}
```

Figura 25- Controle para Lista Ocorrências Abertas

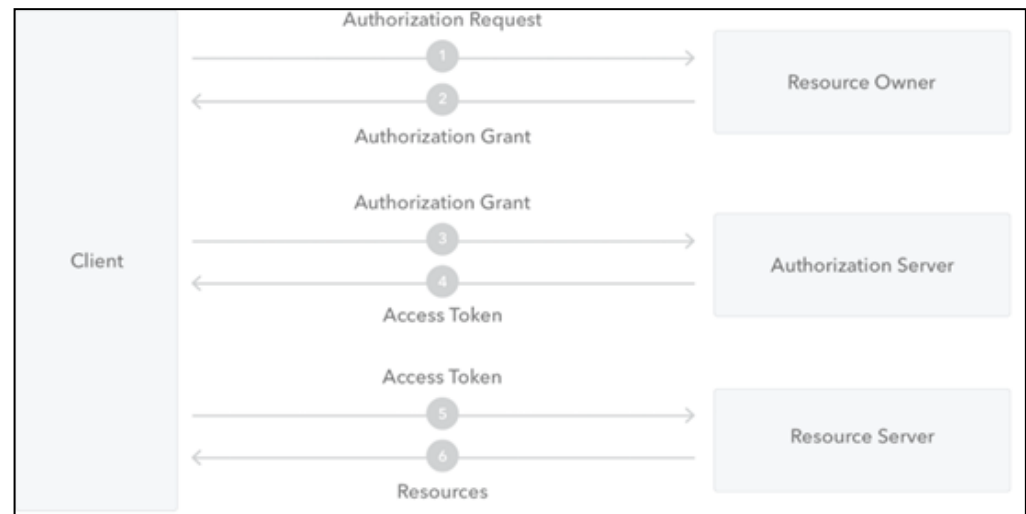
```
{
  "id": 1,
  "dataAbertura": "20-10-2017",
  "dataAtendimento": "20-10-2017",
  "dataConclusao": "",
  "status": "Aberta",
  "historicoInicial": "Deslizamento pos chuvas",
  "rua": "Rua Benedito Pedroso de Andrade",
  "bairro": "Águas da Prata",
  "regiao": "Leste",
  "cep": "12225261",
  "nomeFuncionario": "Andre Simão",
  "idFuncionario": 4,
  "numeroLocal": 101,
  "tipo": "ALAGAMENTO DE VIA",
  "origem": "PESSOALMENTE"
},
```

Figura 26- JSON de Ocorrências Abertas

Desenvolvimento

- Controle de acesso.
 - Autenticação.
 - Autorização.
 - Padrão Oauth.
 - Spring Security.

Figura 27- Fluxograma Controle de Acesso



Desenvolvimento

- Autenticação.
 - Biblioteca JWT
 - BCrypt

Figura 29- Senha Criptografada com BCrypt

FUNCIONARIO_ID	EMAIL	NOME	SENHA	PERFIL_ID
1	capablanca1@bol.com.br	Antonio Alves	\$2a\$10\$F10Q9wxXnwU.7p4/Gh7Ooe9MLSkgHEmXyG8kRadiYDfRdzM8tWJ5O	1
2	programacao@cm@gmail.com	Joao carlos	\$2a\$10\$Sc./Y49agAK46tdzmAPYQpOe6T/GSIFLG6sNh1thC1PQqoD/EIY5qW	2
3	pedroTiago@ex.com.br	Pedro Tiago	\$2a\$10\$D9KbcrRGqwwduHsgLEn5ZQOXF12GGdTzElsybFVTNhxExZ1BFmq0C	3
4	andre@ex.com.br	Andre Simão	\$2a\$10\$/mYJok2OifJXREz9D/xage0q9zpkNN94A3kARxgO4HzzBpAGD2ehe	2
5	fb@ex.com.br	Felipe Bartolomeu	\$2a\$10\$C1NTIMYQtwzNY5K109fZke6tXhn0Q2217TvL3FyjRSWczYVW6.TZ6	2
6	marisae@ex.com.br	Marisa	\$2a\$10\$pwzLVkV7ID6zZSWwXLC6VuK7GbW65nWUsNAYWikFE2JiCscU/y.y	2
7	sebastiana@ex.com.br	Sebastiana	\$2a\$10\$PpRZMOL/hH3kWLkMvLsOyenuq7GycPL8Me6otvs.wwwQQShvG21mPa	2
8	fernanda@ex.com.br	Fernanda	\$2a\$10\$smNF4TvbGf8lvdmqWJzz.euQ6keta//DY4GJALIDmVbBV9Tf7Mrcty	1
9	Teste@teste	Teste	\$2a\$10\$2gn4ei9MKIL/XufS.HSTUum4wy0s.iFLZ7VmcusROBGjQNMw46Ql	1

- Autorização.
- `-@PreAuthorize("hasAnyRole('ADMIN')")`,

Resultados

- Testes
 - 1º Experimento- Acesso ao sistema
 - 2º Experimento- Solicitações de ocorrências.
 - 3º Experimento- Atendimento e geração de relatórios pelos agentes de campo.

Resultados

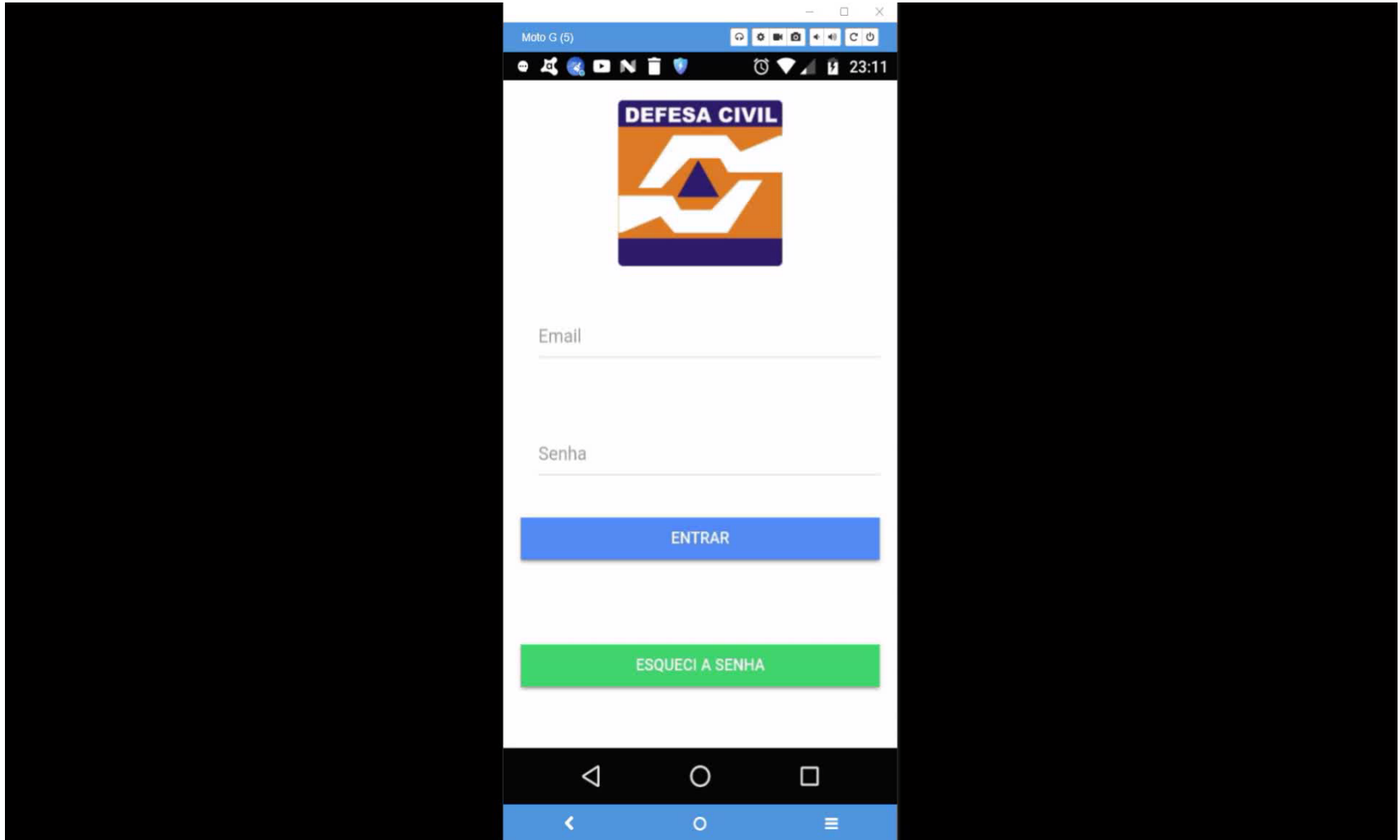
The screenshot shows a mobile application interface for creating a new employee. At the top, there is a status bar with various icons and the time 11:45. Below it is a header bar with a back arrow and the text 'Novo Funcionario'. The form consists of several input fields: 'Nome*' (Name), 'Email*', 'Senha*' (Password), and a dropdown menu for 'Tipo de Funcionario' (Employee Type) with the label 'Tipo Funcionario*'. At the bottom of the form is a blue button labeled 'CRIAR USUARIO' (Create User). The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

Figura 31-Tela Insere Novo Funcionário



Figura 32-Alerta de permissão negada

Resultados



Trabalhos Futuros

- Relatórios.
- Integração com outros setores.
- Módulo para uso em desktop.
- Incluir módulo com mapas para uso web.