Prova Java Logística

Requisitos:

1. Funcionais:

1.1. Castrar um cliente e endereço

1.2. Apresentar o endereço cadastrado no mapa

1.3. Listar todos clientes e seu respectivo endereço;

1.4. Filtrar clientes por região no mapa;

1. Não funcionais:

Padrão de projeto:

Modelo cascata

Back End:

1.1. Spring Boot 2+

1.2. Porta 8080

1.3. Banco de dados H2

1.4. JPA

Front end:

2.1. Angular 8+

2.2. Jester-tester

2.3. aplicação Web

2.4. Google Maps

2.5. API VIA/CEP

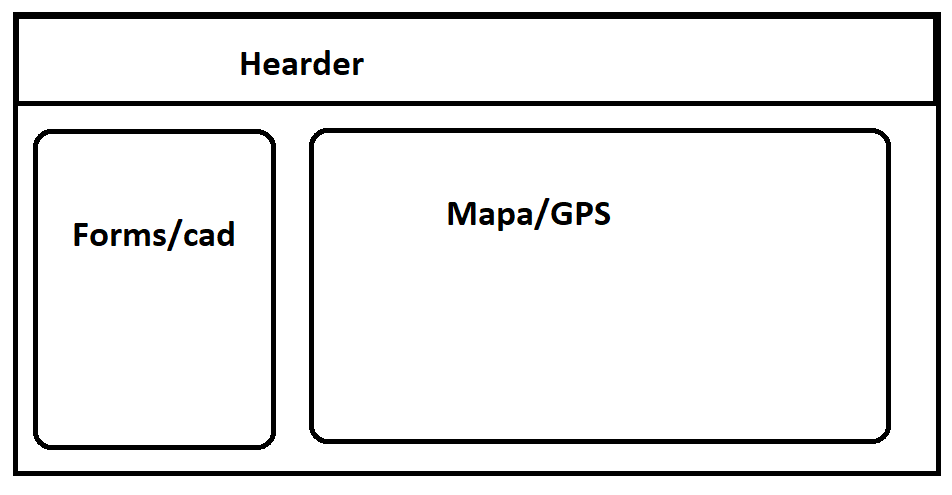
2.6. API nominatim

Criação Tela:

Dados do cliente são obrigatórios

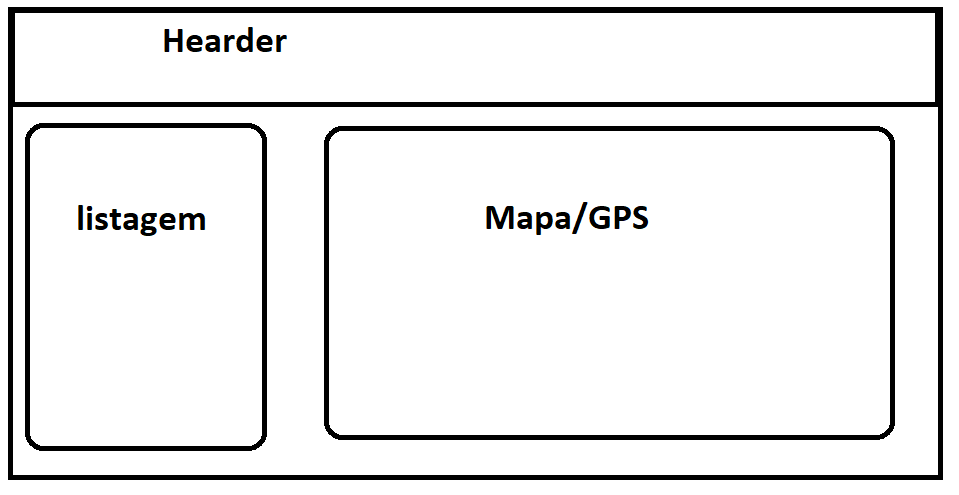
Apenas Cidade e Estado são obrigatórios no cadastro de endereço

* Protótipo tela de cadastro:



1. Ao cadastrar um usuário de ser apresentado a localização no GPS

* Protótipo tela de Listagem:



1. Deve ser apresentado uma combobox, com o os nomes dos clientes cadastrados no banco de dados.
2. Ao selecionar um cliente deve apresentar os dados do mesmo junto com a localização no mapa/gps.

Criação API em Spring Boot:

1. Cria as entidade Cliente e Endereço;
2. Porta: 8080
3. Atributos :

* Cliente:
* idCliente - Identificador da tabela;
* codigo - Código do cliente;
* nome - nome do cliente;
* cnpj - cnpj do cliente
* endereco - endereço do cliente;
* Endereço:
* idEndereco - Identificador da tabela;
* endereco - logradouro
* estado - estado
* bairro - bairro
* numero - número da casa/Ap
* complemento - Bloco/Condomínio ou ponto de referência
* cep - cep
* longitude
* latitude

1. Criar um Crud completo na API cadastrando um cliente com o endereço;
2. Corpo Json para Cadastro:

Url: http://localhost:8080/cliente/save

{

"codigo": Long,

"nome": String,

"cnpj":String,

"endereco":{

"endereco": String,

"estado": String ,

"bairro": String,

"numero": Long,

"complemento": String,

"cep": String,

"longitude": String,

"latitude": String

}

}

1. Corpo Json para Listagem:

Url: http://localhost:8080/cliente/list

{

"idCliente": Long,

"codigo": Long,

"nome": String,

"cnpj":String,

"endereco":{

"idEndereco": Long,

"endereco": String,

"estado": String ,

"bairro": String,

"numero": Long,

"complemento": String,

"cep": String,

"longitude": String,

"latitude": String

}

}

1. URL para buscar por CNPJ:

Url: http://localhost:8080/cliente/cnpj={cnpj}

1. URL para deletar um cliente pelo CNPJ junto o endereço:

Url:http://localhost:8080/cliente/cnpj={cnpj}