Registro Completo do Projeto SaaS

Nesta documentação, está registrado todo o desenvolvimento do projeto SaaS, incluindo backend (Spring Boot + MongoDB), frontend (Angular + Tailwind) e diagramas.

Os principais tópicos abordados foram:

1. Estrutura do backend (modelos, controllers, serviços, repositórios)
2. Configuração do MongoDB no Spring Boot
3. Criação dos componentes Angular (cadastrar/listar clientes e serviços)
4. Implementação do menu inicial para facilitar a navegação
5. Melhoria na usabilidade com dropdown de clientes no cadastro de serviços
6. Diagrama UML do banco de dados e fluxograma do sistema completo
7. Solução de erros e ajustes na aplicação

A seguir, estão os códigos implementados:

Código do Frontend (Angular)

```
// Exemplo de código Angular para Cadastro de Serviços com Dropdown de Clientes

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { Servico } from '../../models/servico.model';
import { ServicoService } from '../../services/servico.service';
import { Cliente } from '../../models/cliente.model';
import { ClienteService } from '../../services/cliente.service';

@Component({
  selector: 'app-servico-cadastrar',
  templateUrl: './servico-cadastrar.component.html',
  styleUrls: ['./servico-cadastrar.component.css']
})
export class ServicoCadastrarComponent implements OnInit {
  novoServico: Servico = {
    clienteId: '',
```

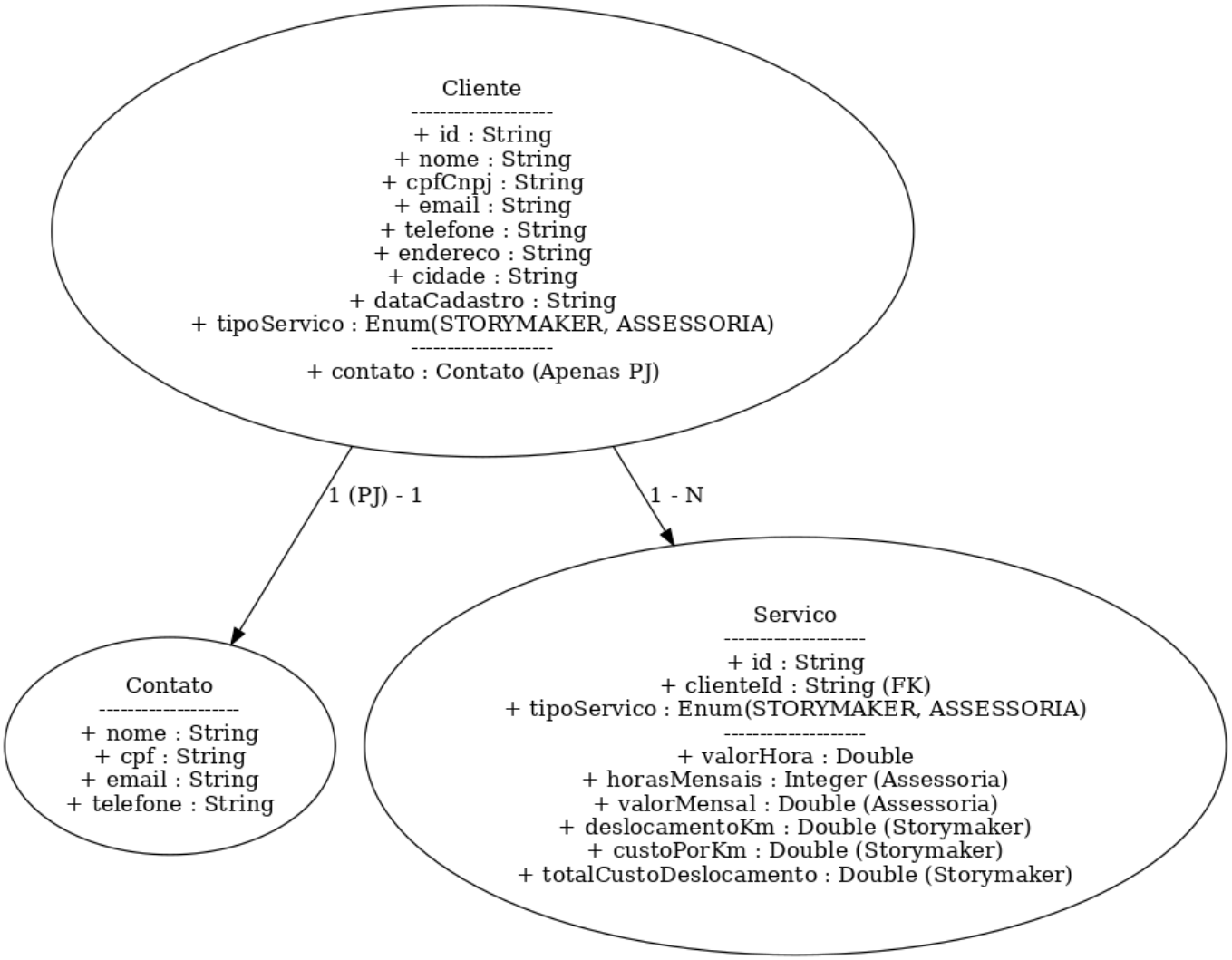
```
    tipoServico: 'ASSESSORIA',
    valorHora: 0,
    horasMensais: 0,
    valorMensal: 0,
    deslocamentoKm: 0,
    custoPorKm: 0,
    totalCustoDeslocamento: 0
  };
  clientes: Cliente[] = [];

   constructor(private servicoService: ServicoService, private clienteService: ClienteService,
private router: Router) {}

  ngOnInit(): void { this.carregarClientes(); }

  carregarClientes(): void {
    this.clienteService.listarTodos().subscribe(
      (dados) => { this.clientes = dados; },
      (erro) => { console.error('Erro ao carregar clientes:', erro); }
    );
  }

  aoSalvar(): void {
    this.servicoService.salvar(this.novoServico).subscribe(
      (servicoSalvo) => {
        console.log('Serviço cadastrado:', servicoSalvo);
        this.router.navigate(['/servicos/listar']);
      },
      (erro) => { console.error('Erro ao cadastrar serviço:', erro); }
    );
  }
}
```

Código do Backend (Spring Boot)

```
// Exemplo de Controller de Serviços no Spring Boot

package com.gh.backend.gh.controller;
import com.gh.backend.gh.model.Servico;
import com.gh.backend.gh.service.ServicoService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;
```

```
@RestController
@RequestMapping("/servicos")
public class ServicoController {
    @Autowired private ServicoService servicoService;

    @GetMapping public List<Servico> listarTodos() { return servicoService.listarTodos(); }

        @GetMapping("/{id}") public Servico buscarPorId(@PathVariable String id) { return servicoService.buscarPorId(id); }

           @PostMapping public Servico salvar(@RequestBody Servico servico) { return servicoService.salvar(servico); }

     @PutMapping("/{id}") public Servico atualizar(@PathVariable String id, @RequestBody Servico servico) { return servicoService.atualizar(id, servico); }

             @DeleteMapping("/{id}") public void deletar(@PathVariable String id) { servicoService.deletar(id); }
}
```

Diagrama UML do Banco de Dados


Cliente
+ id : String
+ nome : String
+ cpfCnpj : String
+ email : String
+ telefone : String
+ endereco : String
+ cidade : String
+ dataCadastro : String
+ tipoServico : Enum(STORYMAKER, ASSESSORIA)
+ contato : Contato (Apenas PJ)
1 (PJ) - 1
1 - N
Contato
+ nome : String
+ cpf : String
+ email : String
+ telefone : String
Servico
+ id : String
+ clienteId : String (FK)
+ tipoServico : Enum(STORYMAKER, ASSESSORIA)
+ valorHora : Double
+ horasMensais : Integer (Assessoria)
+ valorMensal : Double (Assessoria)
+ deslocamentoKm : Double (Storymaker)
+ custoPorKm : Double (Storymaker)
+ totalCustoDeslocamento : Double (Storymaker)