

miParalegal: sistema de gestão para escritórios de advocacia

Equipe 6

Bruno Sales Fiaes Carneiro¹, Luis Eduardo da Silva Belinelli², Marcelo Miranda Cavalcanti³, George Gebers Brizolla⁴, Zanoni Campos Fernandes⁵

¹Ciência da Computação – UNIFACS - 12723119186

²Ciência da Computação – UNIFACS - 1272318959

³Ciência da Computação – UNIFACS - 1272313678

⁴Ciência da Computação – UNIFACS - 12724113504

⁵Ciência da Computação – UNIFACS – 12723131956

Abstract. *This paper comprehends the first delivery of the third assessment (“A3”) of the curricular unit (“UC”) Models, Methods and Technics of Software Engineer, in the first semester of 2024 school year, taught by Prof. Eliane Isadora Faveron Maciel, on Universidade Salvador (UNIFACS). Begins with the introduction of the problem, analysis of requirements and business rules of the software to be built, wich intends to solve the problem of information management and lawsuits follow-up in small and medium size lawfirms, without the heavy costs of pre-existing market options.*

Resumo. *Este artigo compreende a entrega de número 1 (“Entrega 1”) do trabalho que compõe a 3ª avaliação (“A3”) da Unidade Curricular (“UC”) de Modelos, Métodos e Técnicas de Engenharia de Software, relativo ao semestre letivo 2024.1, lecionado pela Profa. Eliane Isadora Faveron Maciel, na Universidade Salvador (UNIFACS). Trata-se de apresentação de problema, análise de requisitos e regras de negócio de programa a ser construído, que pretende resolver a problemática de gestão de informações de processos em escritórios de advocacia de pequeno e médio porte, sendo uma opção aos custosos sistemas existentes no mercado.*

1. Introdução ao problema:

A gestão de pequenos e médios escritórios de advocacia ainda é muito rudimentar. Não raro inexiste controle de quais processos estão sendo acompanhados pelo corpo jurídico, quem é o advogado responsável, em qual fase o processo se encontra ou informações sintéticas sobre os andamentos e trabalhos efetuados. Também existe baixa visibilidade entre as demandas a cargo do escritório e os contratos efetuados para o respectivo acompanhamento.

Na realidade, em grande parte dos casos, não há qualquer sistema informatizado auxiliando o trabalho dos profissionais ou suportando seu relacionamento com os clientes.

Tal soluções, quando existem, ficam reservadas aos grandes escritórios de advocacia, devido especialmente ao grande custo de aquisição ou assinatura de sistemas do tipo.

Com isto, acredita-se que tal atividade pode se beneficiar de sistemas de gestão e tecnologia da informação, auxiliando não apenas os profissionais no seu dia-a-dia - seja pela substituição da grande quantidade de papel dos tradicionais arquivos, seja pela conveniência da facilidade de acesso e centralização dos dados de importância - mas também no seu relacionamento com o cliente - como na emissão de relatórios, por exemplo.

A solução proposta é, portanto, um software elaborado em três camadas, compreendendo: i) front-end disponibilizado em web application, acessível por meio de URL via Internet, hospedado em nuvem; ii) back-end para tratativa de regras de negócio, em linguagem de programação Java, igualmente hospedado em nuvem; e iii) banco de dados para persistência das informações, em linguagem SQL e banco de dados Postgres, também hospedado em nuvem. Esta aplicação integrada tem como objetivo ser um gestor de informações para escritórios de advocacia de pequeno e médio porte, acessível de qualquer parte do mundo, via Internet, e de fácil utilização. O sistema visa permitir a centralização de informações a respeito de processos acompanhados pelo corpo jurídico do escritório.

2. Requisitos

2.1 Requisitos funcionais

RF001: O sistema deve manter o cadastro de processos, com todos os seus dados obrigatórios (vide RN001).

RF002: O sistema deve permitir a alteração dos dados de um processo individual, sem a possibilidade de entrada de dados vazios/nulos (vide RN001).

RF003: O sistema não deve permitir a exclusão de um processo (vide RN001).

RF004: O sistema deve manter o cadastro de andamentos, com todos os seus dados obrigatórios (vide RN003), que ficarão vinculados a um processo individual.

RF005: O sistema deve permitir a alteração dos dados de um andamento, sem a possibilidade de entrada de dados vazios/nulos (vide RN003).

RF006: O sistema deve permitir a exclusão de um andamento.

RF007: Os andamentos serão dispostos em um formato de lista, dentro da visualização de um processo específico.

RF008: O sistema deve manter cadastro de advogados do escritório, com todos os seus dados obrigatórios (vide RN004).

RF009: O sistema deve permitir a alteração dos dados de um advogado individual, sem a possibilidade de entrada de dados vazios/nulos (vide RN004).

RF010: O sistema deve permitir a exclusão de um advogado.

RF011: O sistema deve manter o cadastro de clientes do escritório, com todos os seus dados obrigatórios (vide RN012).

RF012: O sistema deve permitir a alteração do cadastro de cliente, não sendo permitida a entrada de dados vazios/nulos (vide RN012).

RF013: O sistema não deve permitir a exclusão de um cliente.

RF014: O sistema deve manter cadastro de contratos, com todos os seus dados obrigatórios (vide RN013), que ficarão vinculados a um cliente em específico.

RF015: O sistema deve permitir a alteração de um contrato específico, não sendo permitida a entrada de dados nulos/vazios (vide RN013).

RF016: O sistema não deve permitir a exclusão de um contrato.

RF017: O sistema deve manter o cadastro de prazos, com todos os seus dados obrigatórios (RN018), que ficarão vinculados a um processo em específico.

RF018: O sistema deve permitir a alteração de um prazo específico, não sendo permitida a entrada de dados nulos/vazios (vide RN018).

RF019: O sistema deve permitir a exclusão de um prazo.

RF020: O sistema manterá o cadastro de um órgão judicial ou administrativo, com todos os seus dados obrigatórios (vide RN020).

RF021: O sistema permitirá a alteração de um órgão judicial ou administrativo, não sendo permitida a entrada de dados nulos/vazios (vide RN020).

RF022: O sistema permitirá a exclusão de um órgão judicial ou administrativo.

RF023: O sistema terá uma interface visual gráfica em página da Internet (webpage), em modelo single page application, acessível via qualquer navegador de internet, inclusive mobile, hospedada em serviço de nuvem.

RF024: O sistema deve possuir interface gráfica com os seguintes elementos: i) barra de pesquisa; ii) listagem de resultados da pesquisa; iii) overview de informações do processo individual selecionado; iv) listagem de prazos do processo; e v) listagem de andamentos do processo individual selecionado.

RF025: O sistema deverá permitir a busca e filtragem de processos por meio dos seguintes critérios: i) número do processo; ii) advogado responsável; e iii) cliente.

RF026: O sistema permitirá a emissão de relatórios de processos, com base nos requisitos de filtragem descritos na RF025.

RF027: O sistema emitirá lembretes de prazos a vencer, para os próximos 15 (quinze) dias.

RF028: O sistema terá um back-end, igualmente hospedado em serviço de nuvem, que receberá os inputs do front-end e tratará das regras de negócio. Esse back-end também fará a comunicação exclusiva com o banco de dados.

RF029: O sistema terá um banco de dados, responsável pela persistência dos dados.

RF030: O sistema deverá ter uma tela inicial de login, para autenticação e validação do usuário.

2.2 Requisitos não-funcionais

RNF001: O sistema deve ter uma disponibilidade de 95%.

RNF002: O sistema deve ter a seguinte taxa de falhas: 1 a cada 2 mil linhas de código (0,5 falhas/mLOC).

RNF003: O sistema deve responder a qualquer comando em menos de 05 (cinco) segundos, inclusive em termos de banco de dados.

RNF004: O sistema deve manter as informações do banco de dados de forma criptografada.

RNF005: O sistema deve lidar com até 30 acessos simultâneos.

RNF006: O sistema deverá rodar em qualquer navegador web, inclusive mobile, com design responsivo ao tamanho da tela.

RNF007: O banco de dados será do tipo relacional, implementado em PostgreSQL.

RNF008: O back-end do sistema será em linguagem orientada a objetos, a saber, Java.

RNF009: O front-end do sistema poderá ser em qualquer linguagem de desenvolvimento web, contanto que compatível com os demais sistemas de apoio.

3. Regras de negócio

RN001: São dados OBRIGATÓRIOS de todos os processos, as seguintes informações: i) número(s) do processo; ii) órgão judicial ou administrativo associado; iii) fase processual; iv) natureza da demanda; v) cliente(s); vi) posição do cliente; vii) parte(s) contrária(s); vii) advogado responsável; ix) status do processo; x) data de ajuizamento; xi) contrato do cliente (vide RF001).

RN002: O sistema NÃO deve permitir a exclusão de processos.

RN003: São dados OBRIGATÓRIOS de todos os andamentos, as seguintes informações: i) data do andamento; ii) breve descrição do andamento.

RN004: São dados OBRIGATÓRIOS de todos os advogados, as seguintes informações: i) nome completo; ii) registro profissional; iii) endereço.

RN005: O registro profissional de um advogado terá o seguinte formato – xxx.xxx.xxx/UF.

RN006: Somente será permitido o cadastro de UM advogado responsável por processo.

RN007: Um processo pode conter até 5 (cinco) números de processo cadastrados, de forma a contemplar eventual migração de sistema, mudança ou adequação técnica, criação de incidentes ou dependências, virtualização, etc.

RN008: Um processo poderá ter as seguintes naturezas: 1) Cível; 1.1) Comum; 1.2) Consumidor; 1.3) Família; 1.5) Empresarial; 1.6) Outros; 2) Tributário; 3) Penal; 4) Previdenciário; 5) Trabalhista; 6) Ambiental; 7) Eleitoral; 8) Público; 9) Diversos; 10) Procedimento Administrativo.

RN009: Um processo poderá ter as seguintes fases processuais: 1) 1º Grau; 2) 2º Grau; 3) Tribunal Superior; 4) Supremo Tribunal.

RN010: Um procedimento administrativo poderá ter as seguintes fases: 1) 1º Grau; 2) 2º Grau; 3) Especial.

RN011: Os processos e procedimentos somente poderão ter os seguintes 'status': 1) Em andamento; 2) Suspensão; 3) Arquivado temporariamente; 4) Finalizado.

RN012: São dados OBRIGATÓRIOS de todos os clientes: i) nome natural ou empresarial; ii) CPF ou CNPJ; iv) em caso de pessoa jurídica, nome do responsável; v) endereço; vi) telefone.

RN013: São dados OBRIGATÓRIOS dos contratos: i) identificação ou número do contrato; ii) tipo de contrato; iii) data de celebração; iv) validade; v) status.

RN014: O cliente poderá ter um ou vários contratos vinculados a ele.

RN015: Os tipos de contrato podem ser: i) apenas no êxito; ii) mensal fixo; iii) inicial fixo; iv) parcelas por etapa; v) misto.

RN016: A validade contratual pode ser: i) indefinida; ii) data definida.

RN017: Os 'status' dos contratos podem ser: i) válidos; ii) vencidos; iii) rescindidos.

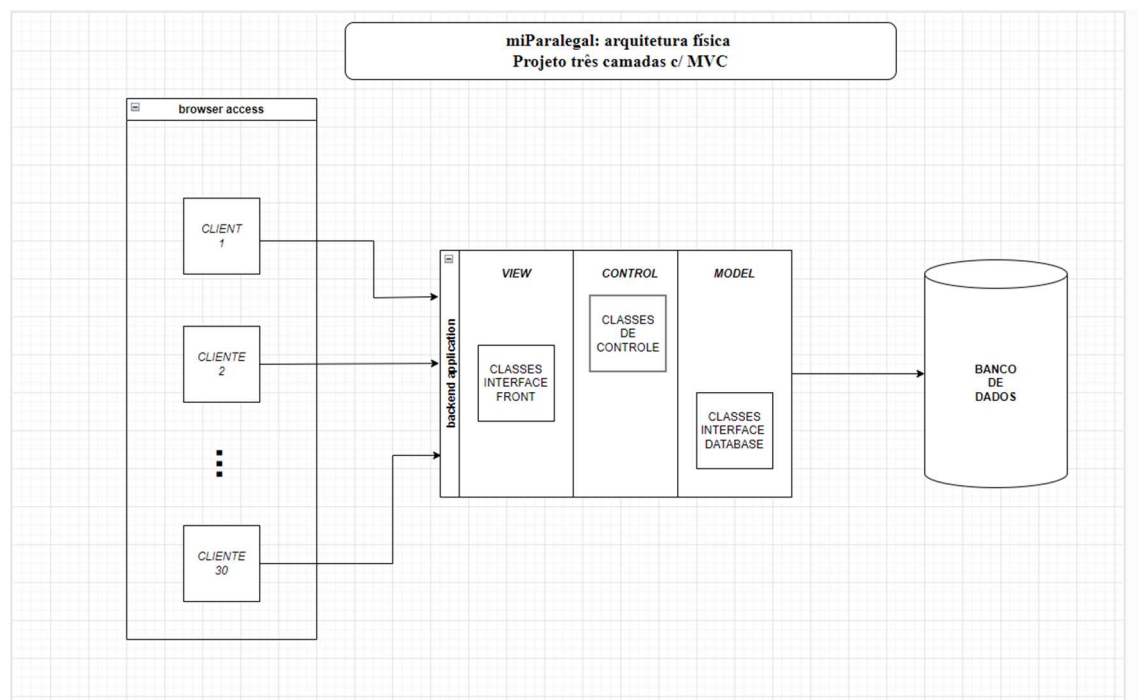
RN018: São dados OBRIGATÓRIOS dos prazos: i) descrição do prazo; ii) data de vencimento do prazo.

RN019: A funcionalidade de emissão de lembretes de prazos a vencer (vide RF027) somente avisará ao respectivo advogado cadastrado nos processos com os ditos prazos.

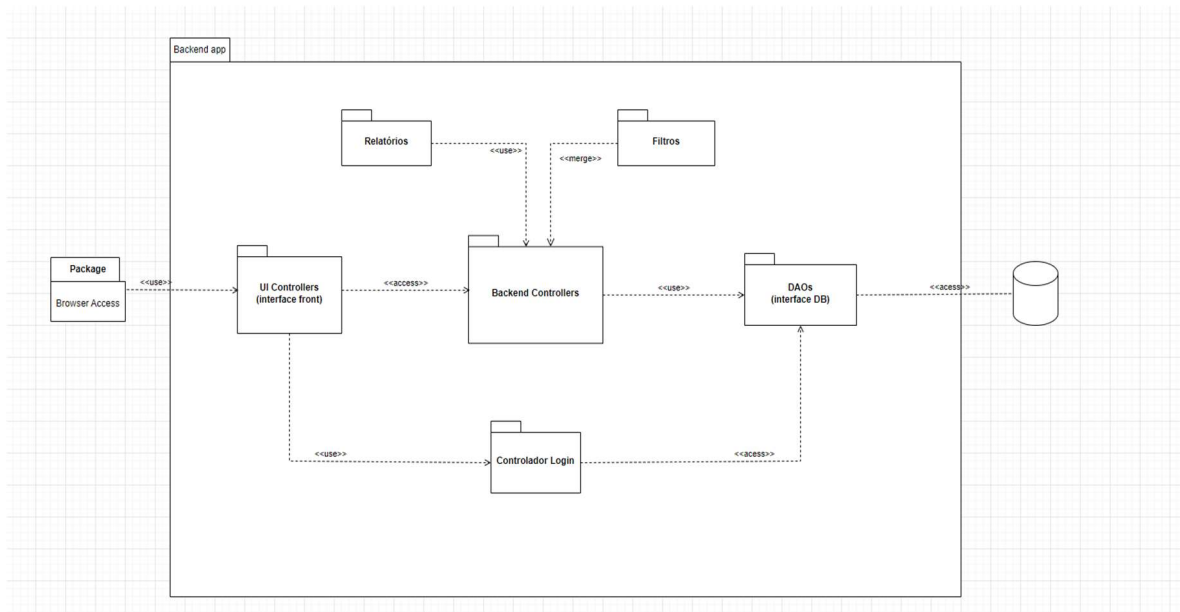
RN020: São dados OBRIGATÓRIOS de um órgão judicial ou administrativo: i) identificação do órgão; ii) vinculação hierárquica (Tribunal ou órgão administrativo superior).

4. Diagramas*

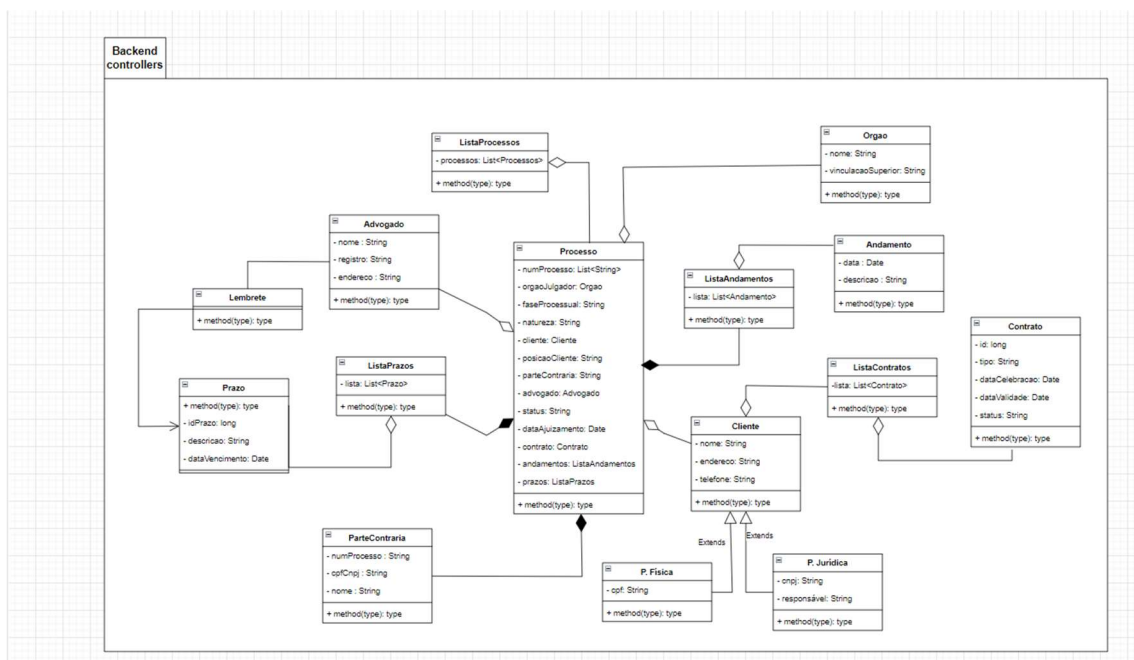
4.1 Arquitetura Física



4.2 Arquitetura Lógica – Diagrama de pacotes

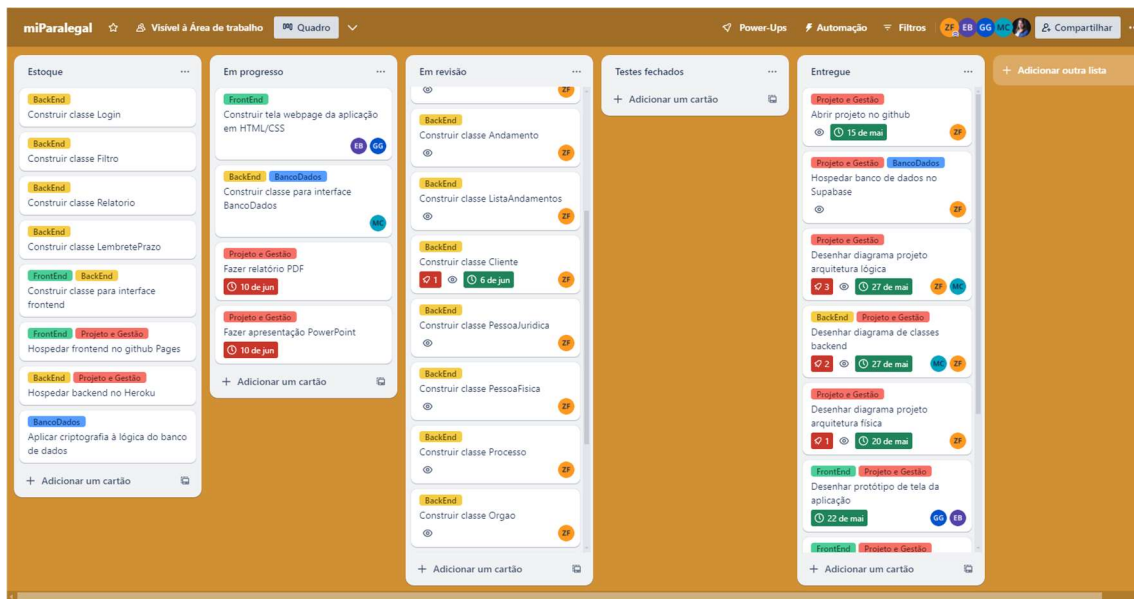


4.3 Arquitetura Lógica – Diagrama de classes do pacote ‘backend controllers’



*acessíveis em: <https://github.com/facs-engenharia-software/docs/tree/main>

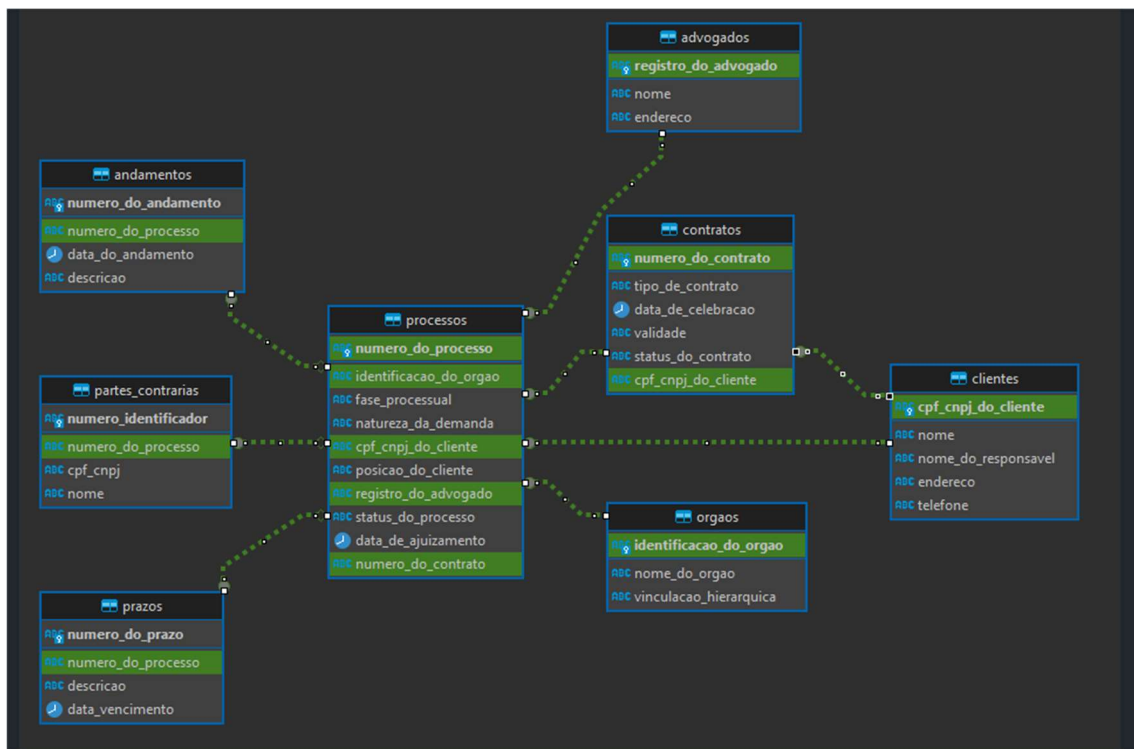
5. Quadro kanban – Trello*



*acessível em: <https://trello.com/b/x4bFTuRM/miparalegal>

6. Código fonte projeto – MVP em construção*

6.1 Diagrama entidade-relacionamento para Banco de Dados



6.2 SQL queries para a construção do Banco de Dados

```
1 CREATE DATABASE miparalegaldb;
2
3 \c miparalegaldb;
4
5 CREATE TABLE advogados (
6     registro_do_advogado VARCHAR(14) PRIMARY KEY,
7     nome VARCHAR(100) NOT NULL,
8     endereco VARCHAR(200) NOT NULL
9 );
10
11 CREATE TABLE clientes (
12     cpf_cnpj_do_cliente VARCHAR(18) PRIMARY KEY,
13     nome VARCHAR(100) NOT NULL,
14     nome_do_responsavel VARCHAR(100),
15     endereco VARCHAR(200) NOT NULL,
16     telefone VARCHAR(20) NOT NULL
17 );
18
19 CREATE TABLE contratos (
20     numero_do_contrato VARCHAR(20) PRIMARY KEY,
21     tipo_de_contrato VARCHAR(10) NOT NULL,
22     data_de_celebracao DATE NOT NULL,
23     validade VARCHAR(13) NOT NULL,
24     status_do_contrato VARCHAR(11) NOT NULL,
25     cpf_cnpj_do_cliente VARCHAR(18) NOT NULL,
26     FOREIGN KEY (cpf_cnpj_do_cliente) REFERENCES clientes(cpf_cnpj_do_cliente)
27 );
28
29 CREATE TABLE orgaos (
30     identificacao_do_orgao VARCHAR(20) PRIMARY KEY,
31     nome_do_orgao VARCHAR(20) NOT NULL,
32     vinculacao_hierarquica VARCHAR(100) NOT NULL
33 );
34
35 CREATE TABLE processos (
36     numero_do_processo VARCHAR(20) PRIMARY KEY,
37     identificacao_do_orgao VARCHAR(20) NOT NULL,
38     fase_processual VARCHAR(17) NOT NULL,
39     natureza_da_demanda VARCHAR(27) NOT NULL,
40     cpf_cnpj_do_cliente VARCHAR(18) NOT NULL,
41     posicao_do_cliente VARCHAR(100) NOT NULL,
42     registro_do_advogado VARCHAR(14) NOT NULL,
```

6.3 Banco de dados hospedado no serviço de nuvem SUPABASE

Database

Database Management

Tables

Functions

Triggers

Enumerated Types

Extensions

Indexes

Publications

Access Control

Roles

Policies

Platform

Backups

Wrappers

Migrations

Webhooks

ZanonCFernandes's Org

Free

/ miparalegal

/ Enable branching

Feedback

Database Tables

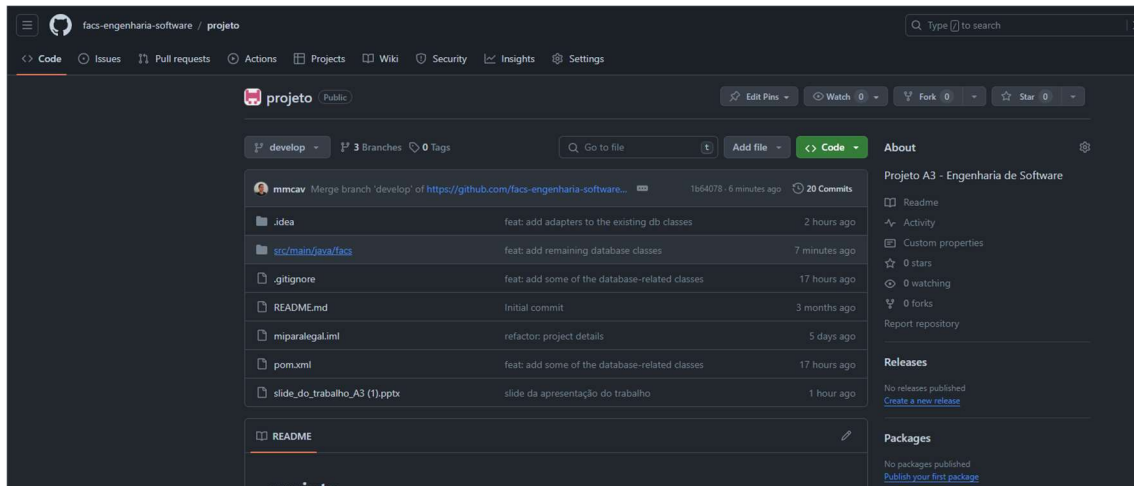
schema: public

Search for a table

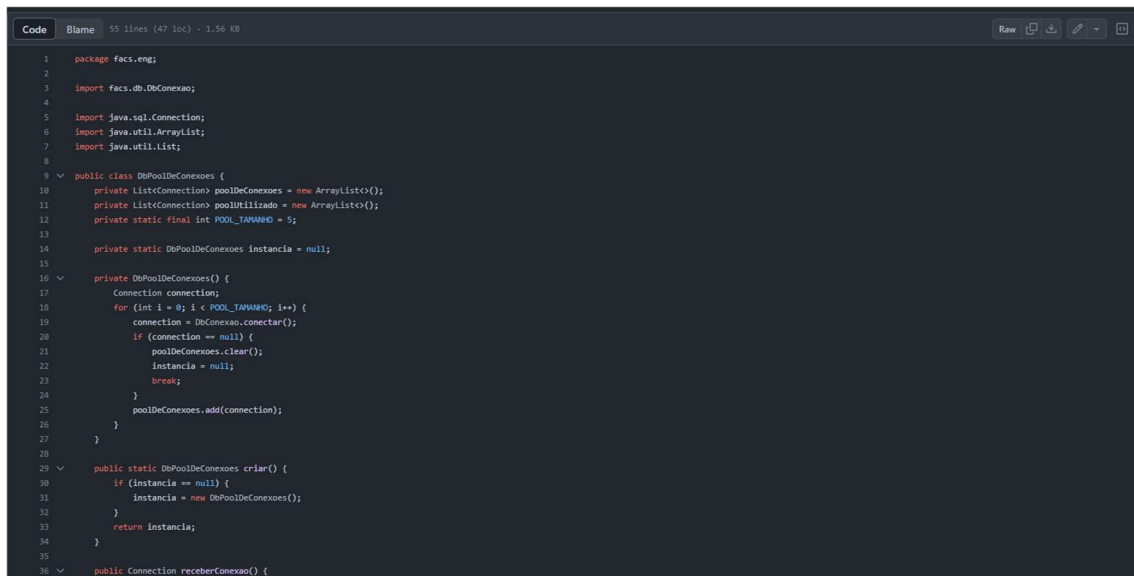
New table

Name	Description	Rows (Estimated)	Size (Estimated)	Realtime Enabled	
advogados	No description	0	8192 bytes	X	3 columns
andamentos	No description	0	16 kB	X	4 columns
clientes	No description	0	8192 bytes	X	5 columns
contratos	No description	0	8192 bytes	X	6 columns
orgaos	No description	0	8192 bytes	X	3 columns
partes_contrarias	No description	0	8192 bytes	X	4 columns
prazos	No description	0	8192 bytes	X	4 columns
processos	No description	0	8192 bytes	X	10 columns

6.4 Tela do repositório geral do projeto



6.5 Classe ('DbPoolDeConexoes') utilizando padrão de projeto ('Design pattern') – tipo Singleton



6.6 Classe exemplo (‘AdaptadorCliente’) utilizando padrão de projeto (‘Design pattern’) – tipo Adapter

```
Code Blame 183 lines (86 loc) · 2.9 KB
1 package facs.db;
2
3 import facs.eng.Cliente;
4 import facs.eng.PessoaFisica;
5 import facs.eng.PessoaJuridica;
6
7 public class AdaptadorCliente {
8     private String cpfOuCnpjDoCliente;
9     private String nome;
10    private String nomeDoResponsavel;
11    private String endereco;
12    private String telefone;
13
14    public AdaptadorCliente(String cpfOuCnpjDoCliente, String nome, String endereco, String telefone) {
15        this.cpfOuCnpjDoCliente = cpfOuCnpjDoCliente;
16        this.nome = nome;
17        this.nomeDoResponsavel = null;
18        this.endereco = endereco;
19        this.telefone = telefone;
20    }
21
22    public AdaptadorCliente(String cpfOuCnpjDoCliente, String nome, String nomeDoResponsavel, String endereco, String telefone) {
23        this.cpfOuCnpjDoCliente = cpfOuCnpjDoCliente;
24        this.nome = nome;
25        this.nomeDoResponsavel = nomeDoResponsavel;
26        this.endereco = endereco;
27        this.telefone = telefone;
28    }
29
30    public AdaptadorCliente(PessoaFisica pessoaFisica) {
31        this.cpfOuCnpjDoCliente = pessoaFisica.getCpf();
32        this.nome = pessoaFisica.getNome();
33        this.nomeDoResponsavel = null;
34        this.endereco = pessoaFisica.getEndereco();
35        this.telefone = pessoaFisica.getTelefone();
36    }
37 }
```

*acessível em: <https://github.com/orgs/facs-engenharia-software/repositories>