

miParalegal

SOFTWARE JURÍDICO


Equipe:

Bruno Sales Fiaes Carneiro,
Luis Eduardo da Silva Belinelli,
Marcelo Miranda Cavalcanti,
George Gebers Brizolla,
Zanoni Campos Fernandes.

—

O Direito e sua
resistência à
inovação.






Como implementar tecnologia em uma área de baixa adesão à tecnologia?

Solução proposta

miParalegal:
Software para gestão de processos focado em escritórios de pequeno e médio porte, auxiliando o registro, acesso e consulta de informações sobre demandas, contando ainda com geração de relatórios.

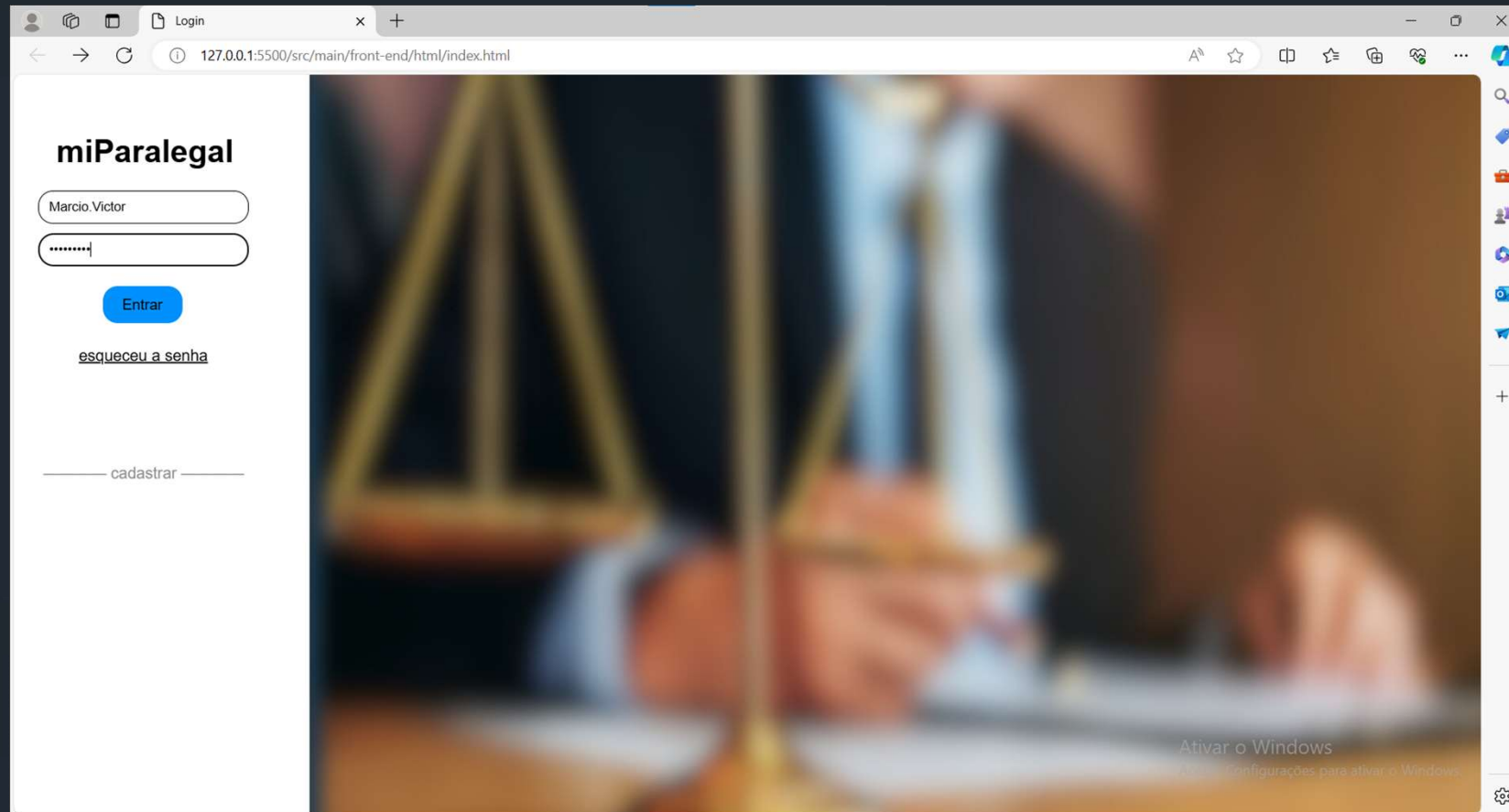




Substituir tempo,
ganhar espaço,
gerar eficiência.

- Menos espaço
- Menos trabalho
- Melhor gerenciamento
- Maior centralização de informações
- Interface visual
- Impressão de relatórios

Tela página de login



Tela protótipo página inicial

Arquivos

Inserir

Salvar

Editar

Publicar

Ajuda

Botões de filtros

PROCESSO

PARTES

OUTROS

Buscar

Andamentos

Data	Descrição
03/02/2023	Despacho do Juiz: "Intimam-se as partes sobre interesse em audiência"
12/04/2023	Audiência Realizada
30/05/2023	Autos conclusos para Sentença
03/02/2023	Sentença de procedência: "Julgo procedente para coordenar a ré a..."

Prazos

Data	Descrição
11/06/2024	Audiência Realizada
05/07/2024	Autos conclusos para Sentença

Resultado

2572202-43.6914.820.5735
7530023-56.3449.764.8154
4837261-18.7345.902.3415
6548279-30.1897.562.3410

Numero Processo: 2572202-43.6914.820.5735

Orgão Judicial: 7ª Vara Cível e Comercial Salvador - Tribunal de Justiça da Bahia

Fase Processual: 1º Grau

Natureza: Empresarial

Cliente: XIS comercio autopeças LTDA.

Posição Cliente: Réu

Parte Contrária: Bom varejo distribuição peças LTDA.

Advogado Responsavel: Orlando Gomes

Status do Processo: Em Andamento

Data de Ajuizamento: 22/10/2023

Contrato do Cliente: ID nº 103

Ativar o Windows.
Acesse Configurações para ativar o Windows.

Diagrama de arquitetura física

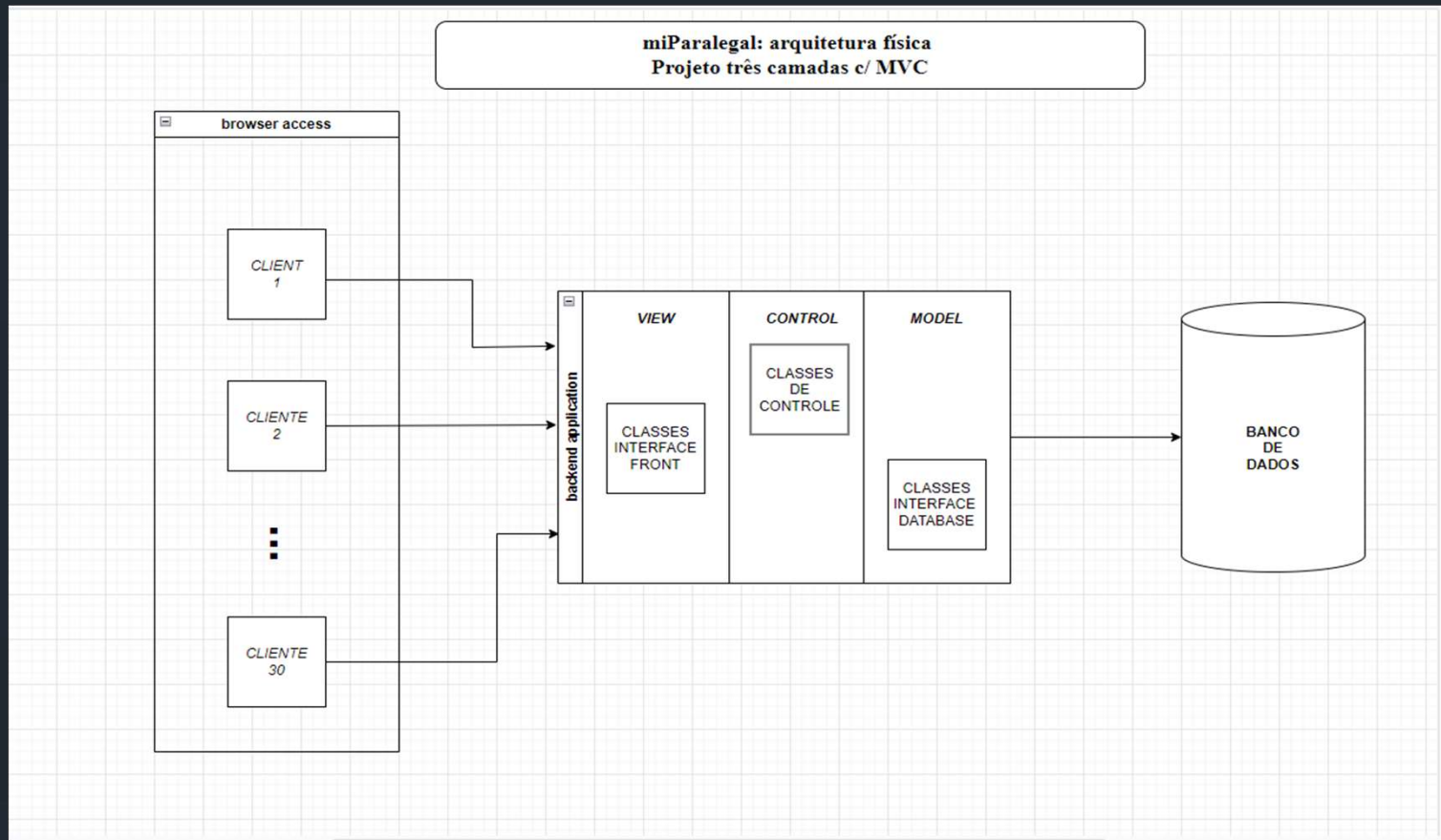


Diagrama lógico de pacotes

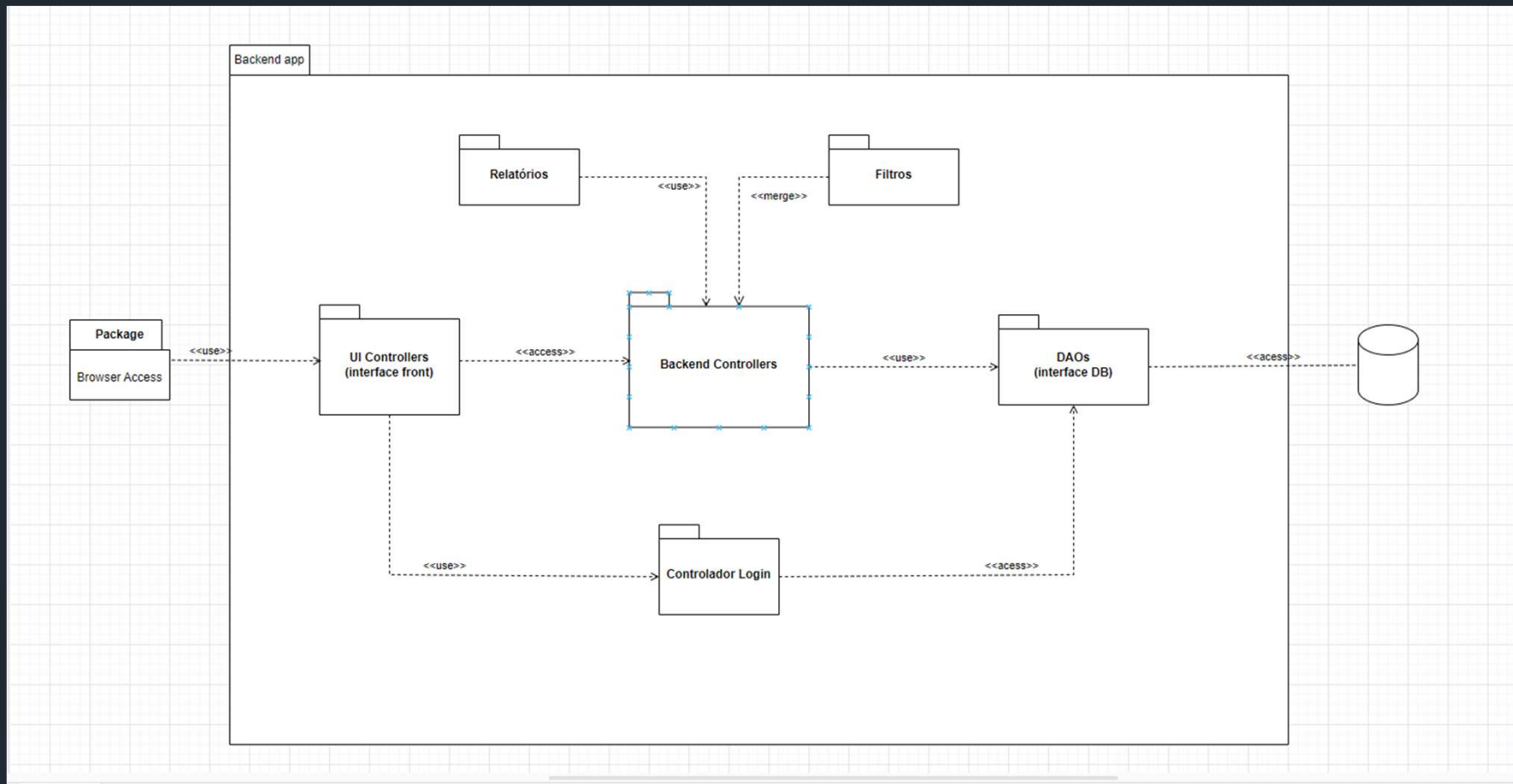


Diagrama lógico de classes – backend controllers

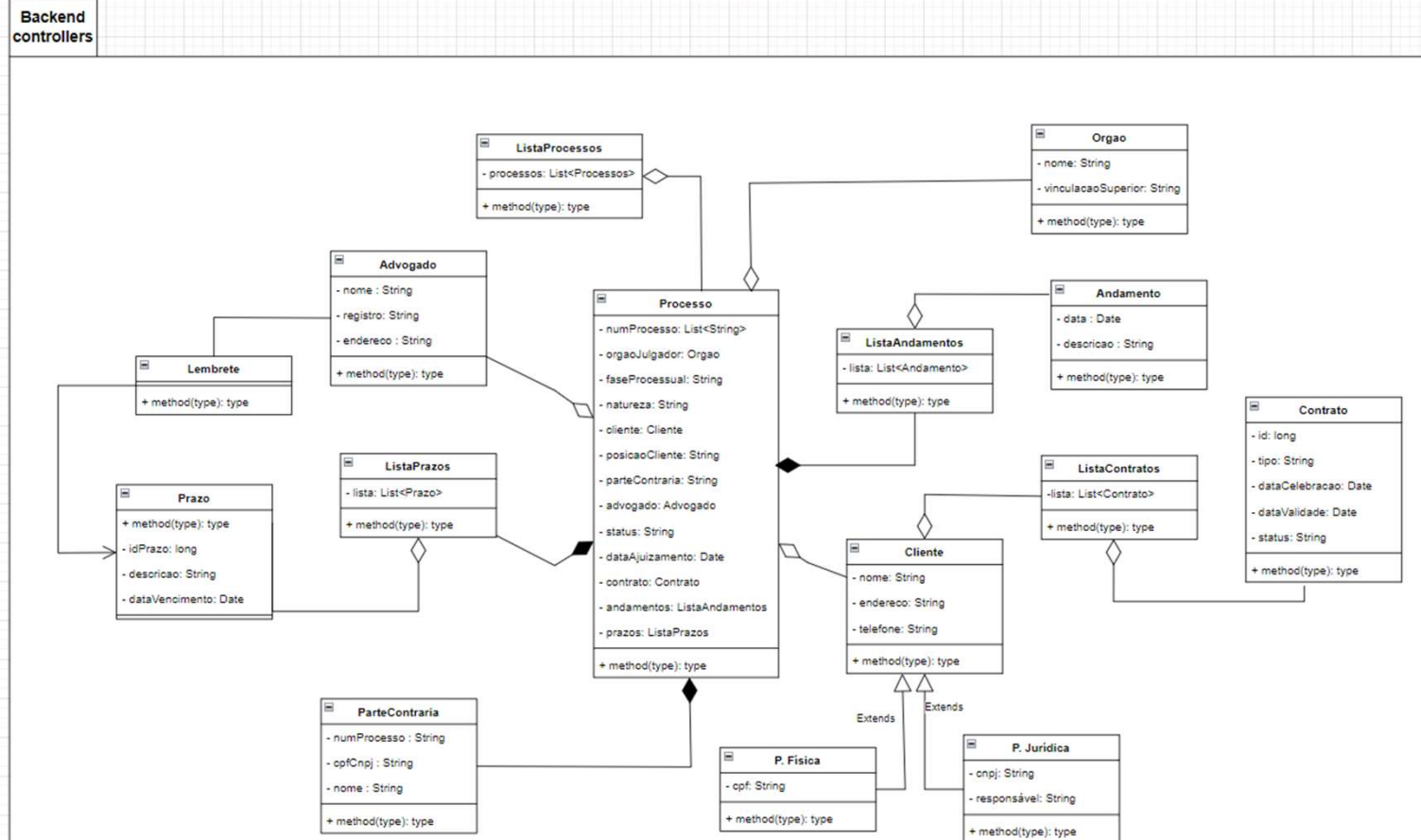


Diagrama entidade- relacionamento do DB

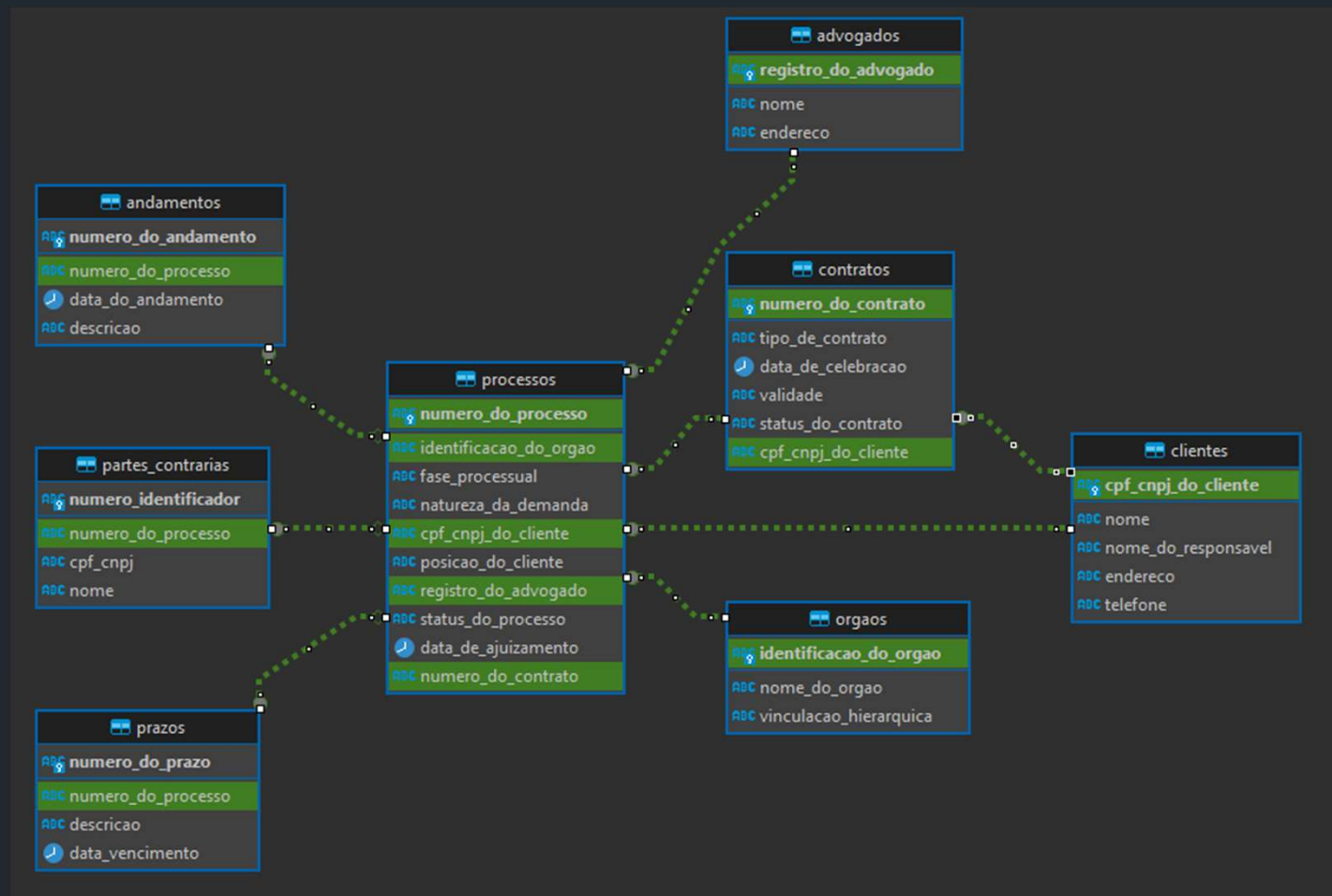


Imagem do banco de dados hospedado em nuvem (Supabase)

Database

DATABASE MANAGEMENT

Tables

Functions

Triggers

Enumerated Types

Extensions

Indexes

Publications

ACCESS CONTROL

Roles

Policies

PLATFORM

Backups

Wrappers

Migrations

Webhooks

TOOLS

Schema Visualizer

ZanoniCFernandes'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

Quadro ágil Kanban (Trello)

The screenshot displays a Trello Kanban board for a project named "miParalegal". The board is organized into five main columns representing different stages of the workflow: "Estoque" (Backlog), "Em progresso" (In Progress), "Em revisão" (In Review), "Testes fechados" (Testing Closed), and "Entregue" (Delivered). Each column contains a list of tasks, each represented by a card. The cards are color-coded by category: BackEnd (yellow), FrontEnd (green), Projeto e Gestão (red), and BancoDados (blue). Each card includes a title, a description, a due date, and a status indicator (e.g., "ZF" for "Zerado/Finalizado"). The "Entregue" column also shows a "Compartilhar" (Share) button. The top of the board features a navigation bar with the project name, a search bar, and various filters and settings.

miParalegal Visível à Área de trabalho Quadro

Estoque

- BackEnd: Construir classe Login
- BackEnd: Construir classe Filtro
- BackEnd: Construir classe Relatorio
- BackEnd: Construir classe LembretePrazo
- FrontEnd BackEnd: Construir classe para interface frontend
- FrontEnd Projeto e Gestão: Hospedar frontend no github Pages
- BackEnd Projeto e Gestão: Hospedar backend no Heroku
- BancoDados: Aplicar criptografia à lógica do banco de dados

Em progresso

Em revisão

- Projeto e Gestão: Fazer relatório PDF (10 de jun)
- Projeto e Gestão: Fazer apresentação PowerPoint (10 de jun)
- BackEnd: Construir classe Prazo (ZF)
- BackEnd: Construir classe ListaPrazos (ZF)
- BackEnd: Construir classe Contrato (1 6 de jun ZF)
- BackEnd: Construir classe ListaContratos (ZF)
- BackEnd: Construir classe Andamento (ZF)
- FrontEnd

Testes fechados

Entregue

- Projeto e Gestão: Abrir projeto no github (15 de mai ZF)
- Projeto e Gestão BancoDados: Hospedar banco de dados no Supabase (ZF)
- Projeto e Gestão: Desenhar diagrama projeto arquitetura lógica (3 27 de mai ZF MC)
- BackEnd Projeto e Gestão: Desenhar diagrama de classes backend (2 27 de mai MC ZF)
- Projeto e Gestão: Desenhar diagrama projeto arquitetura física (1 20 de mai ZF)
- FrontEnd Projeto e Gestão: Desenhar protótipo de tela da aplicação (22 de mai GG EB)
- FrontEnd Projeto e Gestão

+ Adicionar um cartão

Visão geral dos repositórios do projeto (GitHub)

The screenshot displays the GitHub interface for the 'facs-engenharia-software' repository. The top navigation bar includes links for Overview, Repositories (3), Projects, Packages, Teams, People (6), and Settings. A search bar is located on the right.

The 'Repos' sidebar on the left lists repository types: All, Public, Private, Sources, Forks, Archived, and Templates.

The main content area shows a list of 3 repositories under the 'All' tab. The repositories are:

- docs** (Public): Documentação do projeto. Updated 28 minutes ago.
- projeto** (Public): Projeto A3 - Engenharia de Software. Updated 47 minutes ago.
- atividade-15-04** (Public): Updated on Apr 22.

Below the repository list, two detailed views are shown:

docs (Public): This view shows the repository's file structure and commit history. The 'main' branch is selected. The commit history table lists the following commits:

Commit	Author	Message	Time
1369962	ZanoniCFernandes	Delete miParalegal - 28.04.2024.docx	29 minutes ago
32		Commits	
		Update README.md	34 minutes ago
		Add files via upload	1 hour ago
		fix: fixing the schema	2 hours ago
		Add files via upload	3 weeks ago
		Add files via upload	last week
		Add files via upload	2 weeks ago
		Add files via upload	2 hours ago
		Add file - relatório final	29 minutes ago
		Login miParalegal	3 weeks ago
		miParalegal Protótipo tela de aplicação	3 weeks ago

projeto (Public): This view shows the repository's file structure and commit history. The 'develop' branch is selected. The commit history table lists the following commits:

Commit	Author	Message	Time
1b64070	mimav	Merge branch 'develop' of https://github.com/facs-engenharia-software...	49 minutes ago
20		Commits	
		feat: add adapters to the existing db classes	3 hours ago
		feat: add remaining database classes	50 minutes ago
		feat: add some of the database-related classes	17 hours ago
		Initial commit	3 months ago
		refactor: project details	5 days ago
		feat: add some of the database-related classes	17 hours ago
		slide da apresentação do trabalho	2 hours ago

Classe utilizando padrão de projeto
(‘Design Pattern’) Singleton – classe
‘DbPoolConexoes’ do pacote
‘java.facs.eng’

```
Code Blame 55 lines (47 loc) · 1.56 KB
1 package facs.eng;
2
3 import facs.db.DbConexao;
4
5 import java.sql.Connection;
6 import java.util.ArrayList;
7 import java.util.List;
8
9 public class DbPoolDeConexoes {
10     private List<Connection> poolDeConexoes = new ArrayList<>();
11     private List<Connection> poolUtilizado = new ArrayList<>();
12     private static final int POOL_TAMANHO = 5;
13
14     private static DbPoolDeConexoes instancia = null;
15
16     private DbPoolDeConexoes() {
17         Connection connection;
18         for (int i = 0; i < POOL_TAMANHO; i++) {
19             connection = DbConexao.conectar();
20             if (connection == null) {
21                 poolDeConexoes.clear();
22                 instancia = null;
23                 break;
24             }
25             poolDeConexoes.add(connection);
26         }
27     }
28
29     public static DbPoolDeConexoes criar() {
30         if (instancia == null) {
31             instancia = new DbPoolDeConexoes();
32         }
33         return instancia;
34     }
35
36     public Connection receberConexao() {
37         if (poolDeConexoes.isEmpty()) {
38             System.out.println("DEBUG: Pool de conexões vazia.");
39             return null;
40         }
41         Connection connection = poolDeConexoes
42             .remove(poolDeConexoes.size() - 1);
43         poolUtilizado.add(connection);
```

Classe utilizando padrão de projeto (‘Design Pattern’) Adapter – classe ‘AdaptadorCliente’ do pacote ‘java.facs.db’

```
Code Blame 103 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
38    public AdaptadorCliente(PessoaJuridica pessoaJuridica) {
39        this.cpfOuCnpjDoCliente = pessoaJuridica.getCnpj();
40        this.nome = pessoaJuridica.getNome();
41        this.nomeDoResponsavel = pessoaJuridica.getResponsavel();
42        this.endereco = pessoaJuridica.getEndereco();
43        this.telefone = pessoaJuridica.getTelefone();
44    }
45 }
```



FIM

Obrigado!