**CENTRO UNIVERSITÁRIO DE BAURU**

**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMA**

**RAFAEL RODRIGUES FELICIANO**

**AINOÃ PEREIRA SELIDONE**

**SISTEMA DE ADVOCACIA – 2º Termo**

**PROJETO INTERDISCIPLINAR**

BAURU

2020

**CENTRO UNIVERSITÁRIO DE BAURU**

**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMA**

**RAFAEL RODRIGUES FELICIANO**

**AINOÃ PEREIRA SELIDONE**

**SISTEMA DE ADVOCACIA – 2º Termo**

**PROJETO INTERDISCIPLINAR**

Trabalho interdisciplinar, apresentado ao término de cada semestre aos professores envolvidos. Centro Universitário de Bauru para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistema, sob a orientação dos professores: Luiz Bertonha Junior, Celso Luiz Zacari Faria e Marcelo José Storion.

BAURU

2020

1. **CAPÍTULO**
   1. **Tema**

O projeto a ser desenvolvido será de escritório de advocacia Cinigaglia & Galicia Advocacia e Consultoria Jurídica onde atuam dois advogados associados. Tendo o período de funcionamento de segunda-feira à sexta-feira das 9:00 às 18:00. Tendo contato via telefone, WhatsApp em primeira instancia e posteriormente agendando reuniões presenciais para alinhar as informações e contrato.

Terão contato com o cliente diariamente ou esporadicamente, dependendo do desenvolvimento do processo do mesmo.

Visto isso, surgiu a necessidade de um sistema para gestão de clientes, processos e finanças. Pois irá otimizar o tempo, evitará erros, equívocos e deixará mais organizado as informações.

* 1. **Objetivo do Projeto**

Tem como o objetivo principal, controlar melhor os dados dos clientes. Quero dizer, todos os contratos formalizados, ter o acompanhamento se estão em dia ou em aberto ou em atraso. Outra necessidade a ser suprida é o controle da movimentação dos processos onde os advogados associados podem incluir e anexar documentos relativos ao mesmo. Ter uma forma de consulta rápida, prática e limpa para visualizar quando necessário.

* 1. **Delimitação do Problema**

Com o aumento do fluxo de clientes e apenas com o Excel para o controle sobre ele, está tornando-se inviável a administração, podendo gerar erros e/ou equívocos de informações. Tendo assim um retrabalho e perca de tempo buscando através dos históricos de conversa via WhatsApp para confirmar as dúvidas geradas.

* 1. **Justificativa da Escolha do Tema**

O motivo da escolha é que a advocacia é nova, onde estão começando neste ano. Não requer muita complexidade no projeto, para fazer o software de gerenciamento. A ideia é fazer um sistema para gestão dos dados, com tabelas relacionadas.

* 1. **Método de Trabalho**

Usaremos o método de POO (Programação Orientada a Objeto), com heranças de classe.

1. **CAPÍTULO**
   1. **Usuários do Sistema**

Será um sistema pequeno, leve e ágil para que no dia-a-dia da advocacia otimize melhor o tempo de trabalho para então ter foco em outros serviços mais urgentes. O desenvolvimento do software, a princípio é apenas para os dois advogados associados farão o ouso. Que atuarão nos seguimentos de direito da família e infância, direito previdenciário, direito empresarial e direito financeiro.

Visando futuramente, tem-se o desejo que um estagiário terá um acesso para consulta do cliente e processo. Atuando no auxílio dos advogados. Portanto haverá um sistema incremental para tal usabilidade.

* 1. **Desenvolvedores do Sistema**

Nesta primeira etapa do projeto estão envolvidos atualmente, apenas dois programadores responsáveis pelo desenvolvimento do software, engenharia de software e Banco de Dados. E os dois advogados que farão a usabilidade do software.

1. **CAPÍTULO**
   1. **Introdução**

O projeto elaborado a seguir tem como finalidade atender as necessidades de uma advocacia na gestão de dados. Assim otimizando e organizando os dados. Tem o intuito de possuir uma interface simples e eficiente. Não consumindo muito recurso do sistema e ser leve. Irá prevenir erros de contas, quaisquer problemas de equívocos e ter controle sobre cada processo se foi atualizado ou não.

* 1. **Entrevistas**

Entrevista efetuada na data 04/11/2020, as 16:00. Tempo de duração, aproximadamente 36min.

P1: Qual é o tipo de negócio da sua empresa?

RE: Empresa voltada a um escritório jurídico.

P2: Que tipo de processo, trabalhista, empresarial, criminal?

RE: Advocacia atualmente atua em 4 áreas. São elas: Direito de Família e Infância, Direito Previdenciário, Direito Empresarial e Direito Financeiro.

P3: Quem são os usuários?

RE: Apenas os dois Advogados no momento. Porem futuramente, a longo prazo, talvez ter um ou dois estagiários para auxiliar.

P4: Quais são seus objetivos?

RE: 1º Conciliar e organizar pagamento de clientes (financeiras)

2º Pasta compartilhada e organizar documentos (anexado, administrativo)

3º alimentar a atualização de processo

Obs.: existem variações na participação de processo em relação aos honorários. Do total de honorário, é feito essa divisão.

|  |  |  |
| --- | --- | --- |
| Porcentagem de Honorários % | | |
| Direito | Advogado | Advogada |
| Família e Sucessão | 30 | 70 |
| Financeiro | 70 | 30 |
| Empresarial | 70 | 30 |
| Previdenciário | 60 | 40 |
| Exceções | 50 | 50 |

P5:Como esse sistema irá lhe auxiliar?

P6: Como você espera que seja construção do software?

RE: Esperam que o Software seja simples, ágil e estável na sua utilização. Não ter frequentemente problemas com travamento e perda de informação.

P7: Com que frequência você gostaria de atualizações? (Obs.: Não é uma regra formada, mas uma aceitação de todas as partes.)

RE: Que duas vezes a cada semestre, se necessário a segunda reunião. Apresentando os avanços no projeto e fazendo alguns testes para reajustarmos.

P8: O que você acha que pode dar errado? Quais os riscos?

RE: Muita preocupação com a segurança da informação particular do cliente. Voltando falar novamente, ser leve, sem problemas com travamento e perde todo a informação que estava trabalhando.

P9: O que é mais importante, dinheiro, tempo ou usabilidade?

RE: Sendo centrado no objetivo, a importância hoje é a usabilidade e controle.

P10: Quais os componentes do software?

RE: Ambos os advogados terem acesso total a ferramenta, podendo atualizar dados do cliente quando necessário, controle de pagamento dos honorários e acompanhamento de processo e sua atualização.

* 1. **Requisitos de Usuário**

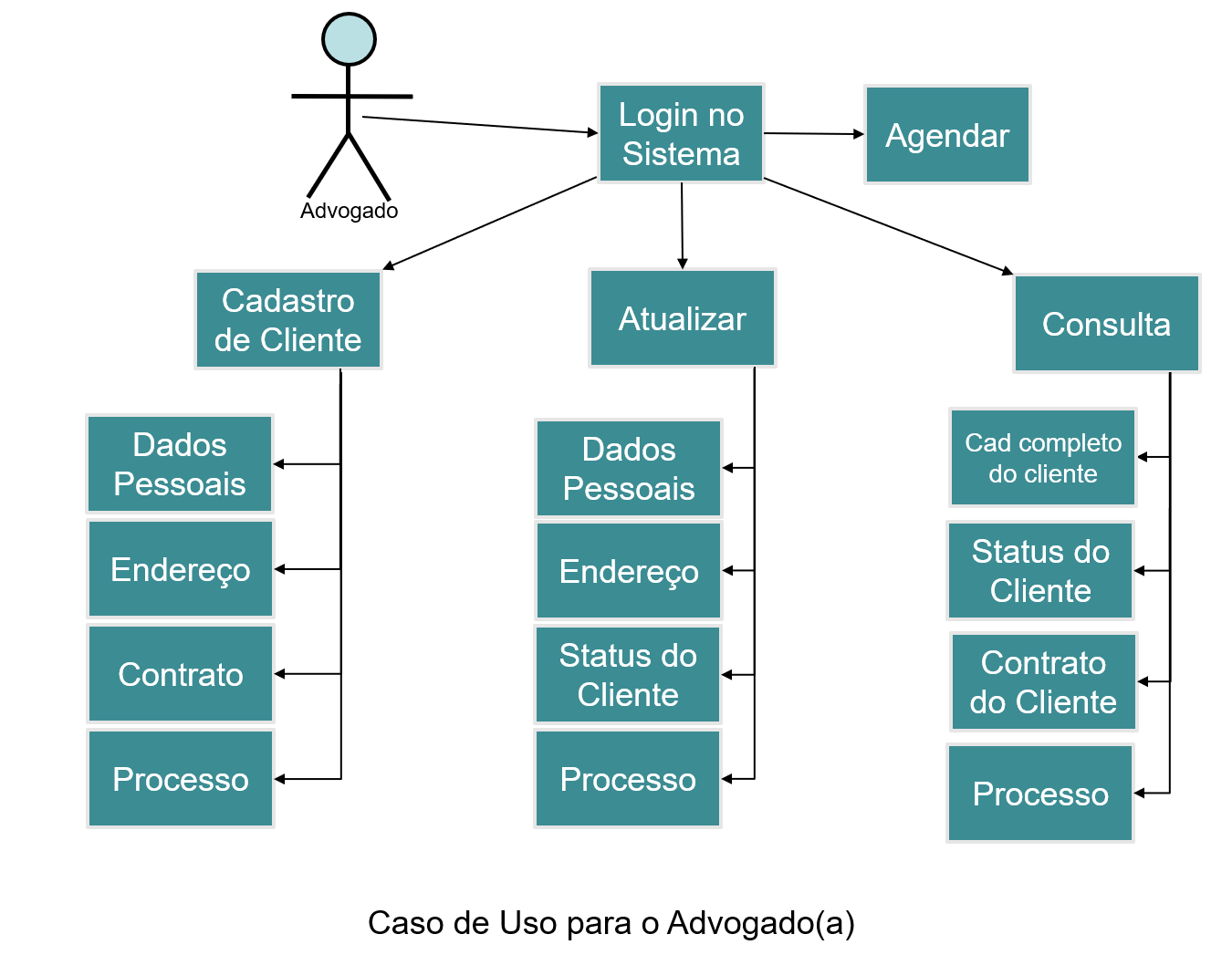
O advogado possuindo o login, fará o acesso no software e posteriormente ele terá opções de cadastrar novo cliente, novo contrato e processo. Na alteração de dados terá a opção de mudar dados pessoais do cliente, a situação do cliente e processo. Para a consulta, poderá visualizar o cadastro completo do cliente incluindo o status, verificar o contrato do cliente e o andamento do processo. Vale ressaltar que, em qualquer momento ou área do software em que o advogado está utilizando, poderá usar a função agendamento. Esse agendamento é utilizado quando necessário seja para os clientes já existentes ou possíveis novos clientes.

Fazendo a inclusão de novo cliente no sistema, vai ser necessário os dados básicos do cliente, como: Nome completo, CPF, RG, data de nascimento, endereço e outras informações pertinentes. Com o cliente tendo cadastro no sistema, o advogado poderá partir para a próxima etapa que é gerado um contrato virtual para o controle de inadimplência, será incluso também valores de honorários, percentual, quantidade de parcelas e valor de parcelas. Feito o cadastro do cliente, formalização do contrato, o advogado montará uma petição inicial para a abertura do processo na comarca e será anexado ao cadastro do cliente. Tendo a abertura do processo no fórum, será feito o cadastramento no software, incluindo todas as informações pertinentes ao processo de forma manual.

Nas alterações, o advogado poderá modificar os dados do cliente como os telefones, e-mails e endereço. Utilizando como base os sistemas já existentes do Tribunal de Justiça, o advogado irá acompanhar fazendo atualizações de forma manual do andamento processual, anexação de documentos pertinentes ao caso e se houver audiência será incluso a data e a hora.

Se houver problema de inadimplência com o cliente, o sistema irá notificar que há parcelas em atraso diretamente o advogado pertinente ao caso, através do software. Vai depender 100% dos advogados para alimentar o sistema referente aos pagamentos mensais e assim o sistema confirmar a baixa do pagamento para evitar avisos equivocados. Além da notificação do sistema sobre o atraso, o advogado poderá consultar o status do cliente e nele verificar quantos dias estão em atraso e caso tenha necessidade, fazer o protesto do mesmo, onde o próprio advogado irá registrar no software que o cliente será protestado.

Terá a opção para consultas, quando o cliente entrar em contato e/ou o advogado precisar de informações rápidas para uso. Nessas consultas teremos: a consulta ao cadastro completo do cliente, isso incluindo a situação dele perante a advocacia se está protestado ou não. Checar o contrato do cliente para verificar as parcelas quando necessário. Por fim, mas não menos importante, poderá consultar o andamento do processo sem depender de sistemas externos. Isto é, o advogado deve alimentar o sistema interno para tal consulta.



**Obs.:** No primeiro acesso do advogado ele efetuará um cadastro criando login e senha.

* 1. **Requisitos de Sistema**
     1. **Requisitos Funcionais**

**Passo 1:** Ao executar o software, será necessário fazer o login inicial dos advogados. Independente de que parte do software o advogado esteja acessando, poderá abrir o agendamento a qualquer momento caso o cliente entre em contato e haja a necessidade.

**Passo 2:** Terá a interface inicial, onde pode escolher entre incluir novo cliente, contrato e processo. Terá também atualização de cliente, situação do cliente e processo. Com a última interação de consulta, poderá visualizar o cadastro completo do cliente juntamente a situação dele junto a advocacia, verificar como está o contrato do cliente e como está a movimentação do processo caso necessite de informação de imediato seja para uso próprio ou para o cliente.

**Passo 3:** Caso escolha cadastro, primeiro é necessário incluir um novo cliente, caso ainda não seja cliente da advocacia, para que possa ser vinculado juntamente ao contrato e processo (novo ou já existente).

**Passo 4:** Passando para alteração, basta apenas usar o CPF/CNPJ do cliente ou nome completo ou nº do processo, como chave de busca para localização do cliente. Para então alterar dados pessoais do cliente ou se ele vai ser ou não protestado e atualizar o desenvolvimento do processo.

**Passo 5:** Selecionando consulta, terá opção de escolha do tipo de busca, onde pode escolher entre CPF/CNPJ do cliente, o nome ou o nº do processo. Podendo consultar o cadastro completo do cliente juntamente a situação dele se foi ou não protestado. Verificar como está o contrato e o andamento do processo, caso o cliente entre em contato perguntando sobre como está a sua ação ou se o advogado necessitar de informação imediata no dia-a-dia.

* + 1. **Requisitos Não-Funcionais**

O Software deve operar durante todo o período de trabalho das 9:00 as 18:00.

Depende do site Tribunal de Justiça estar operante para que possa ser feita as devidas atualizações no software interno sobre a movimentação dos processos. Se o cliente fica inadimplente com a advocacia, ele pode ser protestado, assim incapacitando a operação normal dele no sistema interno em relação ao andamento e/ou finalização do processo.

* + 1. **Requisito do Produto**

O sistema deve estar em pleno funcionamento durante todo o período de expediente dos advogados. Tendo apenas a parada do seu funcionamento, após o horário comercial, em finais de semanas, feriados e quando os fóruns entram em recesso. Quando o advogado depende de um sistema externo para prosseguir com o trabalho e esse mesmo sistema estiver fora do ar, inoperante.

* + 1. **Requisito de Espaço/Desempenho**

O sistema terá um servidor central no escritório, onde será armazenado todos os dados dos clientes. Os advogados terão as versões de uso em seus respectivos aparelhos, Computadores ou Notebooks. Tendo uma média aceitável (pelo proprietário e usuário) do tempo de resposta do sistema desde o primeiro acesso, até todas as funcionalidades disponíveis para que seja feito quaisquer funções de forma separada ou conjunta.

* + 1. **Requisito de Proteção/Confiança**

Deve garantir a integridade dos dados pessoais dos clientes e proteção internamente. Ter a confiabilidade (para o usuário e proprietário) no uso devido do sistema na gestão de dados. Deve acontecer diariamente o backup de todos os dados no servidor para garantir os dados sempre consistente e coeso caso em algum momento falhe e perca os dados no banco de dados.

* + 1. **Requisito de Segurança/Proteção**

Será necessário um sistema de segurança para o servidor, onde deve colocar uma régua padrão informática que conectado a ela estará um nobreak que por sua vez o servidor estará conectado, recebendo alimentação e protegendo caso haja falha na rede elétrica. Na ausência de luz, deve ter ao menos 5min para que seja feito o salvamento dos arquivos e o desligamento dos equipamentos.

* 1. **Descrição Geral**

**Sobre o Sistema:** Consiste em um software único e exclusivo para a advocacia. Será compatível com sistemas operacionais Windows XP, 7 e 10.

**Configuração:** podendo rodar em computadores com 2GB de memória RAM, Intel Core i3, Placa Gráfica Genérica. E configurações superiores.

**Precauções:** Caso venha ocorrer algum problema, como travamento do software ou oscilação/queda de energia, o software terá um mecanismo de salvamento automático de tempos em tempos. Caso o sistema que garante ao menos 5min para fazer os salvamentos falhe.

1. **CAPÍTULO - INTERFACE GRÁFICA**
   1. **Protótipo**

**Classe Principal Advocacia:**

import java.util.Scanner;

public class Advocacia {

    public static void main(String[] args) {

        Scanner dadosUm = new Scanner(System.in);

        Advogados advogadoUm = new Advogados("", "", "", "", "", "", "", "", "", "", "", null);

        Scanner dadosDois = new Scanner(System.in);

        Clientes clienteUm = new Clientes("", "","", "", "", "", "", "", "", "", "", null);

        Endereco enderecoUm = new Endereco("", "", "", "", "", "", "");

        Endereco enderecoDois = new Endereco("", "", "", "", "", "", "");

        Processos processoUm = new Processos("", "", "", "", "", "", "", 0.0);

        Contrato contratoUm = new Contrato("", "", 0.0, 0, 0.0, "", 0, 0);

        advogadoUm.setCodAdv("CODADV01001");

        advogadoUm.setNome("Gabriel Galicia");

        advogadoUm.setOab("OABSP-50231");

        advogadoUm.setCivil("Solteiro");

        advogadoUm.setNasc("12/04/1989");

        advogadoUm.setEmail("gabriel\_g@hotmail.com");

        advogadoUm.setCpf("354.895.762-64");

        advogadoUm.setRg("22.961.528-6");

        advogadoUm.setTel("(014)98585-4525");

        enderecoUm.setRua("Otavio Pinheiro Brisola");

        enderecoUm.setNumero("6-98");

        enderecoUm.setComplemento("casa");

        enderecoUm.setBairro("Jd. Panorama");

        enderecoUm.setCep("17011-450");

        enderecoUm.setCidade("Bauru");

        enderecoUm.setEstado("SP");

        System.out.println("----------------------------------------------");

        System.out.println("---------------CADASTRO DE ACESSO-------------");

        System.out.println("Bem-Vindo, por favor crie seu login de acesso!");

        System.out.println("----------------------------------------------");

        System.out.println("Novo Login: ");

        advogadoUm.setLogin(dadosUm.nextLine());

        System.out.println("Nova Senha: ");

        advogadoUm.setSenha(dadosUm.nextLine());

        System.out.println("----------------------------------------------");

        System.out.println("---------CADASTRO EFETIVADO COM SUCESSO-------");

        System.out.println("----------------------------------------------");

        System.out.println("----------------LOGIN DE ACESSO---------------");

        Scanner login = new Scanner(System.in);

        Scanner senha = new Scanner(System.in);

        System.out.println("Login: ");

        String acessologin = login.nextLine();

        System.out.println("Senha: ");

        String acessosenha = senha.nextLine();

        for(int i=0; i<1;){

            if((advogadoUm.getLogin().equals(acessologin)) && (advogadoUm.getSenha().equals(acessosenha))){

                i++;

            }else{

                System.out.println("----------------------------------------------");

                System.out.println("Login e/ou Senha inválidos! \nPor Favor, tente novamente!");

                System.out.println("Login: ");

                acessologin = login.nextLine();

                System.out.println("Senha: ");

                acessosenha = senha.nextLine();

            }

        }

        while(true){

            System.out.println("----------------------------------------------");

            System.out.println("Advogado(a): " + advogadoUm.getNome()+ "(Logado)");

            System.out.println("--------------------CADASTRO------------------");

            System.out.println("[1]:Dados Pessoais do cliente \n[2]:Endereço do cliente \n[3]:Contrato do cliente \n[4]:Processo do cliente");

            System.out.println("-------------------ATUALIZAÇÃO----------------");

            System.out.println("[5]:Advogado(e-mail e telefone) \n[6]:Dados Pessoais do cliente \n[7]:Endereço do cliente \n[8]:Status do cliente \n[9]:Contrato do cliente \n[10]:Processo do cliente");

            System.out.println("--------------------CONSULTA------------------");

            System.out.println("[11]:Cadastro completo do cliente \n[12]:Satus do cliente \n[13]:Contrato do cliente \n[14]:Processo do cliente");

            System.out.println("----------------------------------------------");

            Scanner menu = new Scanner(System.in);

            System.out.println("ESCOLHA A OPÇÃO:");

            int opcao = menu.nextInt();

        if (opcao==1){

            System.out.println("----------------------------------------------");

            System.out.println("    CADASTRO DE DADOS PESSOAIS DO CLIENTE");

            System.out.println("----------------------------------------------");

            clienteUm.setCodCli("CODCLI01001");

            System.out.println("Nome: ");

            clienteUm.setNome(dadosDois.nextLine());

            System.out.println("Estado civil: ");

            clienteUm.setCivil(dadosDois.nextLine());

            System.out.println("Data de Nascimento: ");

            clienteUm.setNasc(dadosDois.nextLine());

            System.out.println("E-mail: ");

            clienteUm.setEmail(dadosDois.nextLine());

            System.out.println("CPF: ");

            clienteUm.setCpf(dadosDois.nextLine());

            System.out.println("RG: ");

            clienteUm.setRg(dadosDois.nextLine());

            System.out.println("Telefone: ");

            clienteUm.setTel(dadosDois.nextLine());

        }else if (opcao==2){

            System.out.println("----------------------------------------------");

            System.out.println("      CADASTRO DE ENDEREÇO DO CLIENTE");

            System.out.println("----------------------------------------------");

            System.out.println("Rua: ");

            enderecoDois.setRua(dadosDois.nextLine());

            System.out.println("Nº: ");

            enderecoDois.setNumero(dadosDois.nextLine());

            System.out.println("Complemento: ");

            enderecoDois.setComplemento(dadosDois.nextLine());

            System.out.println("Bairro: ");

            enderecoDois.setBairro(dadosDois.nextLine());

            System.out.println("CEP: ");

            enderecoDois.setCep(dadosDois.nextLine());

            System.out.println("Cidade: ");

            enderecoDois.setCidade(dadosDois.nextLine());

            System.out.println("Estado: ");

            enderecoDois.setEstado(dadosDois.nextLine());

        }else if(opcao==3){

            System.out.println("----------------------------------------------");

            System.out.println("             CONTRATO DO CLIENTE");

            System.out.println("----------------------------------------------");

            contratoUm.setCodAcordo("CODACO01001");

            System.out.println("Data da Formalização: ");

            contratoUm.setDataAcordo(dadosDois.nextLine());

            System.out.println("Valor Total: ");

            contratoUm.setValorTotal(dadosDois.nextDouble());

            System.out.println("Quantidade de Parcelas: ");

            contratoUm.setQtdParcela(dadosDois.nextInt());

            System.out.println("Data de Vencimento: ");

            contratoUm.setDataVenc(dadosDois.nextLine());

            System.out.println("Valor de Parcela: ");

            contratoUm.setValorParcela(dadosDois.nextDouble());

            System.out.println("Parcela em aberto: ");

            contratoUm.setParcelaPaga(dadosDois.nextInt());

        }else if(opcao==4){

            System.out.println("----------------------------------------------");

            System.out.println("              PROCESSO DO CLIENTE");

            System.out.println("----------------------------------------------");

            System.out.println("Nº do Processo: ");

            processoUm.setProcesso(dadosDois.nextLine());

            System.out.println("Honorários(em porcentagem): ");

            clienteUm.setHonorario(dadosDois.nextLine());

            System.out.println("Requerido: ");

            processoUm.setRequerido(dadosDois.nextLine());

            System.out.println("Comarca: ");

            processoUm.setComarca(dadosDois.nextLine());

            System.out.println("Tipo: ");

            processoUm.setTipo(dadosDois.nextLine());

            System.out.println("Data da Abertura: ");

            processoUm.setData(dadosDois.nextLine());

            System.out.println("Valor: ");

            processoUm.setValor(dadosDois.nextDouble());

        }else if(opcao==5){

            System.out.println("----------------------------------------------");

            System.out.println("             DADOS DO ADVOGADO(A)");

            System.out.println("----------------------------------------------");

            System.out.println("E-mail: ");

            advogadoUm.setEmail(dadosUm.nextLine());

            System.out.println("Telefone: ");

            advogadoUm.setTel(dadosUm.nextLine());

        }else if(opcao==6){

            System.out.println("----------------------------------------------");

            System.out.println("              DADOS DO CLIENTE");

            System.out.println("----------------------------------------------");

            System.out.println("Estado civil: ");

            clienteUm.setCivil(dadosDois.nextLine());

            System.out.println("E-mail: ");

            clienteUm.setEmail(dadosDois.nextLine());

            System.out.println("Telefone: ");

            clienteUm.setTel(dadosDois.nextLine());

        }else if(opcao==7){

            System.out.println("----------------------------------------------");

            System.out.println("             ENDEREÇO DO CLIENTE");

            System.out.println("----------------------------------------------");

            System.out.println("Rua: ");

            enderecoDois.setRua(dadosDois.nextLine());

            System.out.println("Nº: ");

            enderecoDois.setNumero(dadosDois.nextLine());

            System.out.println("Complemento: ");

            enderecoDois.setComplemento(dadosDois.nextLine());

            System.out.println("Bairro: ");

            enderecoDois.setBairro(dadosDois.nextLine());

            System.out.println("CEP: ");

            enderecoDois.setCep(dadosDois.nextLine());

            System.out.println("Cidade: ");

            enderecoDois.setCidade(dadosDois.nextLine());

            System.out.println("Estado: ");

            enderecoDois.setEstado(dadosDois.nextLine());

        }else if(opcao==8){

            System.out.println("----------------------------------------------");

            System.out.println("             STATUS DO CLIENTE");

            System.out.println("----------------------------------------------");

            System.out.println("Cliente: " + clienteUm.getNome());

            System.out.println("Situação: ");

            clienteUm.setSituacao(dadosDois.nextLine());

            System.out.println("Protestado: ");

            clienteUm.setProtesto(dadosDois.nextLine());

        }else if(opcao==9){

            System.out.println("----------------------------------------------");

            System.out.println("             CONTRATO DO CLIENTE");

            System.out.println("----------------------------------------------");

            System.out.println("Cliente: " + clienteUm.getNome());

            System.out.println("Nº do Contrato: " + contratoUm.getCodAcordo());

            System.out.println("Formalizado em: " + contratoUm.getDataAcordo());

            System.out.println("Valor do Contrato: R$" + contratoUm.getValorTotal());

            System.out.println("Parcelas em Aberto: ");

            contratoUm.setParcelaPaga(dadosDois.nextInt());

            System.out.println("Parcelas em Atraso: ");

            contratoUm.setParcelaVencida(dadosDois.nextInt());

        }else if(opcao==10){

            System.out.println("----------------------------------------------");

            System.out.println("             PROCESSO DO CLIENTE");

            System.out.println("----------------------------------------------");

            System.out.println("Data da Audiencia: ");

            processoUm.setAudiencia(dadosDois.nextLine());

            System.out.println("Movimentação: ");

            processoUm.setMoviemnto(dadosDois.nextLine());

        }else if(opcao==11){

            System.out.println("----------------------------------------------");

            System.out.println("              DADOS CADASTRAIS");

            System.out.println("----------------------------------------------");

            System.out.println(clienteUm);

            System.out.println(enderecoDois);

        }else if(opcao==12){

            System.out.println("----------------------------------------------");

            System.out.println("                   STATUS");

            System.out.println("----------------------------------------------");

            System.out.println("Código de Registro: " + clienteUm.getCodCli());

            System.out.println("Cliente: " + clienteUm.getNome());

            System.out.println("Situação do cliente: " + clienteUm.getSituacao());

            System.out.println("Protestado: " + clienteUm.getProtesto());

        }else if(opcao==13){

            System.out.println("----------------------------------------------");

            System.out.println("             RELATÓRIO DO CONTRATO");

            System.out.println("----------------------------------------------");

            System.out.println("Código de Registro: " + clienteUm.getCodCli());

            System.out.println("Cliente: " + clienteUm.getNome());

            System.out.println(contratoUm);

        }else if(opcao==14){

            System.out.println("----------------------------------------------");

            System.out.println("            RELATÓRIO DO PROCESSO");

            System.out.println("----------------------------------------------");

            System.out.println("Código de Registro: " + clienteUm.getCodCli());

            System.out.println("Cliente: " + clienteUm.getNome());

            System.out.println(processoUm);

        }else{

            System.out.println("Fim do Programa!");

            dadosUm.close();

            dadosDois.close();

            login.close();

            senha.close();

            menu.close();

            break;

        }

        }

    }

}

**SuperClasse Pessoa:**

public class Pessoa {

    private String nome;

    private String civil;

    private String dataNasc;

    private String email;

    private String CPF;

    private String RG;

    private String telefone;

    public Pessoa(String nome, String civil, String nascimento, String email, String CPF, String RG, String telefone){

        this.nome = nome;

        this.civil = civil;

        this.dataNasc = nascimento;

        this.email = email;

        this.CPF = CPF;

        this.RG = RG;

        this.telefone = telefone;

     }

     public String getNome(){

         return nome;

     }

     public void setNome(String nome){

         this.nome = nome;

     }

     public String getCivil(){

         return civil;

     }

     public void setCivil(String civil){

         this.civil = civil;

     }

     public String getNasc(){

         return dataNasc;

     }

     public void setNasc(String nascimento){

         this.dataNasc = nascimento;

     }

     public String getEmail(){

         return email;

     }

     public void setEmail(String email){

         this.email = email;

     }

     public String getCpf(){

         return CPF;

     }

     public void setCpf(String CPF){

         this.CPF = CPF;

     }

     public String getRg(){

         return RG;

     }

     public void setRg(String RG){

         this.RG = RG;

     }

     public String getTel(){

         return telefone;

     }

     public  void setTel(String telefone){

         this.telefone = telefone;

     }

     @Override

     public String toString(){

         return "\nNome: " + nome + "\nData de Nascimento: " + dataNasc + "\nE-mail: " +  email + "\nEstado Civil: " + civil + "\nCPF: " + CPF + "\nRG: " + RG + "\nTelefone: " + telefone + "\n";

     }

 }

**simClasse Endereco:**

public class Endereco {

    private String rua;

    private String numero;

    private String complemento;

    private String bairro;

    private String CEP;

    private String cidade;

    private String estado;

    public Endereco(String rua, String numero, String complemento, String bairro, String CEP, String cidade, String estado){

        this.rua = rua;

        this.numero = numero;

        this.complemento = complemento;

        this.bairro = bairro;

        this.CEP = CEP;

        this.cidade = cidade;

        this.estado = estado;

    }

    public String getRua(){

        return rua;

    }

    public void setRua(String rua){

        this.rua = rua;

    }

    public String getNumero(){

        return numero;

    }

    public void setNumero(String numero){

        this.numero = numero;

    }

    public String getComplemento(){

        return complemento;

    }

    public void setComplemento(String complemento){

        this.complemento = complemento;

    }

    public String getBairro(){

        return bairro;

    }

    public void setBairro(String bairro){

        this.bairro = bairro;

    }

    public String getCep(){

        return CEP;

    }

    public void setCep(String CEP){

        this.CEP = CEP;

    }

    public String getCidade(){

        return cidade;

    }

    public void setCidade(String cidade){

        this.cidade = cidade;

    }

    public String getEstado(){

        return estado;

    }

    public  void setEstado(String estado){

        this.estado = estado;

    }

    public String toString(){

        return "\nRua: " + rua + " Nº: " + numero + "\nComplemento: " + complemento + "\nBairro: " +  bairro + "\nCEP:  " + CEP + "\nCidade: " + cidade + "\nEstado: " + estado;

    }

}

**Classe Advogados:**

public class Advogados extends Pessoa{

    private String cod\_adv;

    private String OAB;

    private String cod\_login;

    private String senha;

    private Endereco endereco;

    public Advogados(String cod\_adv, String OAB, String cod\_login, String senha, String nome, String civil, String nascimento, String email, String CPF, String RG, String telefone, Endereco endereco){

        super(nome, civil, nascimento, email, CPF, RG, telefone);

        this.cod\_adv = cod\_adv;

        this.OAB = OAB;

        this.cod\_login = cod\_login;

        this.senha = senha;

        this.endereco = endereco;

    }

    public String getCodAdv(){

        return cod\_adv;

    }

    public void setCodAdv(String cod\_adv){

        this.cod\_adv = cod\_adv;

    }

    public String getOab(){

        return OAB;

    }

    public void setOab(String OAB){

        this.OAB = OAB;

    }

    public Endereco getEndereco(){

        return endereco;

    }

    public void setEndereco(Endereco endereco){

        this.endereco = endereco;

    }

    public String getLogin(){

        return cod\_login;

    }

    public void setLogin(String cod\_login){

        this.cod\_login = cod\_login;

    }

    public String getSenha(){

        return senha;

    }

    public void setSenha(String senha){

        this.senha = senha;

    }

    @Override

    public String toString(){

        return "\n Código: " + cod\_adv + "\nNº da OAB: " + OAB + "\n" + super.toString() + endereco;

    }

}

**Classe Clientes:**

public class Clientes extends Pessoa {

        private String cod\_cli;

        private String honorario;

        private String situ;

        private String protesto;

        private Endereco endereco;

        public Clientes(String cod\_cli, String honorario, String situ, String protesto, String nome, String civil, String nascimento, String email, String CPF, String RG, String telefone, Endereco endereco){

            super(nome, civil, nascimento, email, CPF, RG, telefone);

            this.cod\_cli = cod\_cli;

            this.honorario = honorario;

            this.situ = situ;

            this.protesto = protesto;

            this.endereco = endereco;

        }

        public String getCodCli(){

            return cod\_cli;

        }

        public void setCodCli(String cod\_cli){

            this.cod\_cli = cod\_cli;

        }

        public String getHonorario(){

            return honorario;

        }

        public void setHonorario(String honorario){

            this.honorario = honorario;

        }

        public String getSituacao(){

            return situ;

        }

        public void setSituacao(String situ){

            this.situ = situ;

        }

        public String getProtesto(){

            return protesto;

        }

        public void setProtesto(String protesto){

            this.protesto = protesto;

        }

        public Endereco getEndereco(){

            return endereco;

        }

        public void setEndereco(Endereco endereco){

            this.endereco = endereco;

        }

        @Override

        public String toString(){

            return "\nCódigo de Registro:" + cod\_cli + super.toString() + "\nSituação: " + situ + "\nHonorários: " + honorario + "%" + "\nProtesto: " + protesto + endereco;

        }

}

**Classe Processos:**

public class Processos{

    private String processo;

    private String requerido;

    private String comarca;

    private String tipo;

    private String data;

    private String audiencia;

    private String mov\_proc;

    private Double valor;

    public Processos(String processo, String requerido, String comarca, String tipo, String data, String audiencia, String mov\_proc, Double valor){

        this.processo = processo;

        this.requerido = requerido;

        this.comarca = comarca;

        this.tipo = tipo;

        this.data = data;

        this.audiencia = audiencia;

        this.mov\_proc = mov\_proc;

        this.valor = valor;

    }

    public String getProcesso(){

        return processo;

    }

    public void setProcesso(String processo){

        this.processo = processo;

    }

    public String getComarca(){

        return comarca;

    }

    public void setComarca(String comarca){

        this.comarca = comarca;

    }

    public String getTipo(){

        return tipo;

    }

    public void setTipo(String tipo){

        this.tipo = tipo;

    }

    public String getData(){

        return data;

    }

    public void setData(String data){

        this.data = data;

    }

    public String getAudiencia(){

        return audiencia;

    }

    public void setAudiencia(String audiencia){

        this.audiencia = audiencia;

    }

    public String getMoviemnto(){

        return mov\_proc;

    }

    public void setMoviemnto(String mov\_proc){

        this.mov\_proc = mov\_proc;

    }

    public Double getValor(){

        return valor;

    }

    public void setValor(Double valor){

        this.valor = valor;

    }

    public String getRequerido(){

        return requerido;

    }

    public void setRequerido(String requerido){

        this.requerido = requerido;

    }

    public String toString(){

        return "\nNº do Processo: " + processo + "\nRequerido: " + requerido + "\nComarca: " + comarca + "\nTipo de Processo: " + tipo + "\nData de Abertura: " + data + "\nData da Audiencia: " + audiencia + "\nValor do Processo: R$" +  valor + "\nMovimento: " + mov\_proc;

    }

}

**Classe Contrato:**

public class Contrato {

    private String cod\_acordo;

    private String data\_acordo;

    private Double valor\_total;

    private int qtd\_parc;

    private Double valor\_parc;

    private String data\_venc;

    private int parc\_pagar;

    private int parc\_vencidas;

    public Contrato(String cod\_acordo, String data\_acordo, Double valor\_total,  int qtd\_parc, Double valor\_parc, String data\_venc, int parc\_pagar, int parc\_vencidas){

        this.cod\_acordo = cod\_acordo;

        this.data\_acordo = data\_acordo;

        this.valor\_total = valor\_total;

        this.qtd\_parc = qtd\_parc;

        this.valor\_parc = valor\_parc;

        this.data\_venc = data\_venc;

        this.parc\_pagar = parc\_pagar;

        this.parc\_vencidas = parc\_vencidas;

    }

    public String getCodAcordo(){

        return cod\_acordo;

    }

    public void setCodAcordo(String cod\_acordo){

        this.cod\_acordo = cod\_acordo;

    }

    public String getDataAcordo(){

        return data\_acordo;

    }

    public void setDataAcordo(String data\_acordo){

        this.data\_acordo = data\_acordo;

    }

    public Double getValorTotal(){

        return valor\_total;

    }

    public void setValorTotal(Double valor\_total){

        this.valor\_total = valor\_total;

    }

    public int getQtdParcela(){

        return qtd\_parc;

    }

    public void setQtdParcela(int qtd\_parc){

        this.qtd\_parc = qtd\_parc;

    }

    public Double getValorParcela(){

        return valor\_parc;

    }

    public void setValorParcela(Double valor\_parc){

        this.valor\_parc = valor\_parc;

    }

    public String getDataVenc(){

        return data\_venc;

    }

    public void setDataVenc(String data\_venc){

        this.data\_venc = data\_venc;

    }

    public int getParcelaPagar(){

        return parc\_pagar;

    }

    public void setParcelaPaga(int parc\_pagar){

        this.parc\_pagar = parc\_pagar;

    }

    public int getParcelaVencida(){

        return parc\_vencidas;

    }

    public void setParcelaVencida(int parc\_vencidas){

        this.parc\_vencidas = parc\_vencidas;

    }

    public String toString(){

        return "\nNº do Contrato: " + cod\_acordo + "\nData do Acordo: " + data\_acordo + "\nValor Total: R$" + valor\_total + "\nValor da Parcela: R$" + valor\_parc + "\nTotal de Parcelas: " + qtd\_parc + "\nData do Vencimento: " + data\_venc + "\nParcela em Aberto: " + parc\_pagar + "\nParcelas em  Atraso: " + parc\_vencidas;

    }

}

**TESTE DE LOGIN**

PS D:\ITE\PROJETO INTERDISCIPLINAR\PROJETO> & 'c:\Users\Usuario\.vscode\extensions\vscjava.vscode-java-debug-0.29.0\scripts\launcher.bat' 'C:\Program Files\Java\jdk-15.0.1\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\Usuario\AppData\Roaming\Code\User\workspaceStorage\c3c4dcd9041b3ab5e44bd5d844730a0b\redhat.java\jdt\_ws\PROJETO\_11277589\bin' 'Advocacia'

----------------------------------------------

---------------CADASTRO DE ACESSO-------------

Bem-Vindo, por favor crie seu login de acesso!

----------------------------------------------

Novo Login:

Gabriel GG

Nova Senha:

gab123

----------------------------------------------

---------CADASTRO EFETIVADO COM SUCESSO-------

----------------------------------------------

----------------LOGIN DE ACESSO---------------

Login:

Gbriel

Senha:

123

----------------------------------------------

Login e/ou Senha inválidos!

Por Favor, tente novamente!

Login:

Gabriel GG

Senha:

gab

----------------------------------------------

Login e/ou Senha inválidos!

Por Favor, tente novamente!

Login:

Gabriel

Senha:

gab123

----------------------------------------------

Login e/ou Senha inválidos!

Por Favor, tente novamente!

Login:

Gabriel GG

Senha:

gab123

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

**TESTE DE CADASTRO/CONSULTA CLIENTE**

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

1

----------------------------------------------

CADASTRO DE DADOS PESSOAIS DO CLIENTE

----------------------------------------------

Nome:

Antonio Silveira

Estado civil:

Casado

Data de Nascimento:

15/06/1974

E-mail:

antonio\_sil@uol.com.br

CPF:

653.123.487-65

RG:

52.321.012-8

Telefone:

(014)98585-2525

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

2

----------------------------------------------

CADASTRO DE ENDEREÇO DO CLIENTE

----------------------------------------------

Rua:

Eduardo Vergueiro de Lorena

Nº:

9-80

Complemento:

Apto.101

Bairro:

Jd. Panorama

CEP:

17011-450

Cidade:

Bauru

Estado:

SP

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

11

----------------------------------------------

DADOS CADASTRAIS

----------------------------------------------

Código de Registro:CODCLI01001

Nome: Antonio Silveira

Data de Nascimento: 15/06/1974

E-mail: antonio\_sil@uol.com.br

Estado Civil: Casado

CPF: 653.123.487-65

RG: 52.321.012-8

Telefone: (014)98585-2525

Situação:

Honorários: %

Protesto: null

Rua: Eduardo Vergueiro de Lorena Nº: 9-80

Complemento: Apto.101

Bairro: Jd. Panorama

CEP: 17011-450

Cidade: Bauru

Estado: SP

**TESTE DE CADSATRO DE CONTRATO E PROCESSO**

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

3

----------------------------------------------

CONTRATO DO CLIENTE

----------------------------------------------

Data da Formalização:

04/11/2020

Valor Total:

2000

Quantidade de Parcelas:

10

Data de Vencimento:

Todo dia 20/mes

Valor de Parcela:

200

Parcela em aberto:

1

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

4

----------------------------------------------

PROCESSO DO CLIENTE

----------------------------------------------

Nº do Processo:

654136

Honorários(em porcentagem):

30

Requerido:

Jose Miguel

Comarca:

Forum de Bauru

Tipo:

civil

Data da Abertura:

5/11/2020

Valor:

20000

**ATUALIZAÇÃO ADVOGADO/CLIENTE**

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

5

----------------------------------------------

DADOS DO ADVOGADO(A)

----------------------------------------------

E-mail:

gabriel.ferreira12@outlook.com

Telefone:

(014)97745-2112

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

6

----------------------------------------------

DADOS DO CLIENTE

----------------------------------------------

Estado civil:

Divorciado

E-mail:

antonio\_silveira54@msn.com

Telefone:

(014)96652-5351

**ATUALIZAÇÃO DE STATUS/CONTRATO/PROCESSO**

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

8

----------------------------------------------

STATUS DO CLIENTE

----------------------------------------------

Cliente: Antonio Silveira

Situação:

Em dia

Protestado:

não

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

10

----------------------------------------------

PROCESSO DO CLIENTE

----------------------------------------------

Data da Audiencia:

30/11/2020

Movimentação:

Peticao inicial formalizada

**CONSULTA CADASTRO/STATUS/CONTRATO/PROCESSO**

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

11

----------------------------------------------

DADOS CADASTRAIS

----------------------------------------------

Código de Registro:CODCLI01001

Nome: Antonio Silveira

Data de Nascimento: 15/06/1974

E-mail: antonio\_silveira54@msn.com

Estado Civil: Divorciado

CPF: 653.123.487-65

RG: 52.321.012-8

Telefone: (014)96652-5351

Situação: Em dia

Honorários: 30%

Protesto: naonull

Rua: Eduardo Vergueiro de Lorena Nº: 9-80

Complemento: Apto.101

Bairro: Jd. Panorama

CEP: 17011-450

Cidade: Bauru

Estado: SP

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

12

----------------------------------------------

STATUS

----------------------------------------------

Código de Registro: CODCLI01001

Cliente: Antonio Silveira

Situação do cliente: Em dia

Protestado: nao

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

13

----------------------------------------------

RELATÓRIO DO CONTRATO

----------------------------------------------

Código de Registro: CODCLI01001

Cliente: Antonio Silveira

Nº do Contrato: CODACO01001

Data do Acordo: 04/11/2020

Valor Total: R$2000.0

Valor da Parcela: R$200.0

Total de Parcelas: 10

Data do Vencimento: Todo dia 20/mes

Parcela em Aberto: 1

Parcelas em Atraso: 0

----------------------------------------------

Advogado(a): Gabriel Galicia(Logado)

--------------------CADASTRO------------------

[1]:Dados Pessoais do cliente

[2]:Endereço do cliente

[3]:Contrato do cliente

[4]:Processo do cliente

-------------------ATUALIZAÇÃO----------------

[5]:Advogado(e-mail e telefone)

[6]:Dados Pessoais do cliente

[7]:Endereço do cliente

[8]:Status do cliente

[9]:Contrato do cliente

[10]:Processo do cliente

--------------------CONSULTA------------------

[11]:Cadastro completo do cliente

[12]:Satus do cliente

[13]:Contrato do cliente

[14]:Processo do cliente

----------------------------------------------

ESCOLHA A OPÇÃO:

14

----------------------------------------------

RELATÓRIO DO PROCESSO

----------------------------------------------

Código de Registro: CODCLI01001

Cliente: Antonio Silveira

Nº do Processo: 654136

Requerido: Jose Miguel

Comarca: Forum de Bauru

Tipo de Processo: civil

Data de Abertura: 5/11/2020

Data da Audiencia: 30/11/2020

Valor do Processo: R$20000.0

Movimento: Peticao inicial formalizada

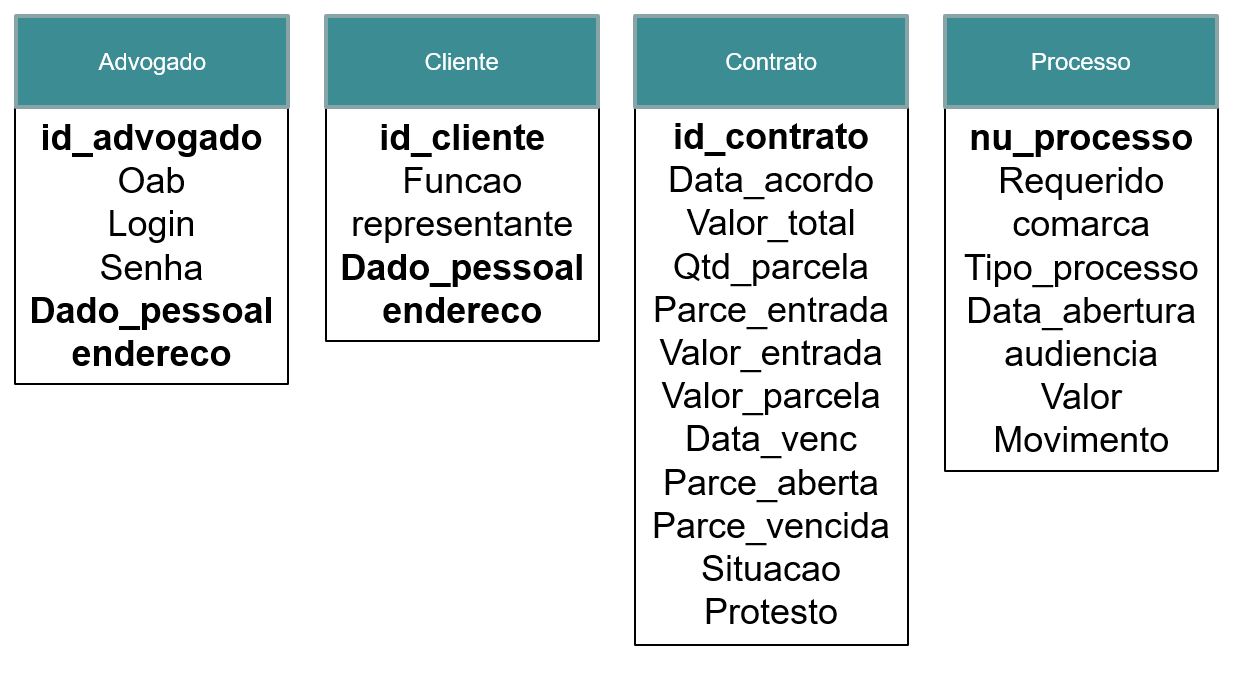
1. **CAPÍTULO - Modelo de Dados**
   1. **Modelo Lógico da Base de Dados**
      1. **Modelo Entidade Relacional (MER)**

**Entidades:** Advogado, Cliente, Contrato e Processo.



**Atributos:** oab, login, senha, honorário, situação, protesto, data\_acordo, valor\_total, qtd\_parcela, valor\_parcela, data\_venc, parc\_aberta, parc\_vencida, requerido, comarca, tipo\_processo, audiência, valor\_processo, movi\_processo.

**Atributos-chaves:** id\_advogado, id\_cliente, id\_contrato id\_processo.



**Obs.:** Dado\_pessoal e endereco são chaves-estrangeiras.

**Relacionamentos:**

(...)advogado formaliza um ou mais contratos e esse(s) contratos podem ser relacionado a um ou mais advogados(...)

(...)o cliente pode formalizar um ou mais contratos, porem o contrato só pode pertencer à apenas um cliente(..)

(...)advogado pode mover a ação em um ou mais processos(...)

(...)advogado entra em contato com um ou mais clientes e vice versa(...)

(...)um processo é atrelado a um contrato e um contrato atrela apenas a um processo(...)

(...)o advogado atualiza os processos quantas vezes forem necessárias(...)

(...)o processo pode ser atualizado por um ou mais advogados(...)

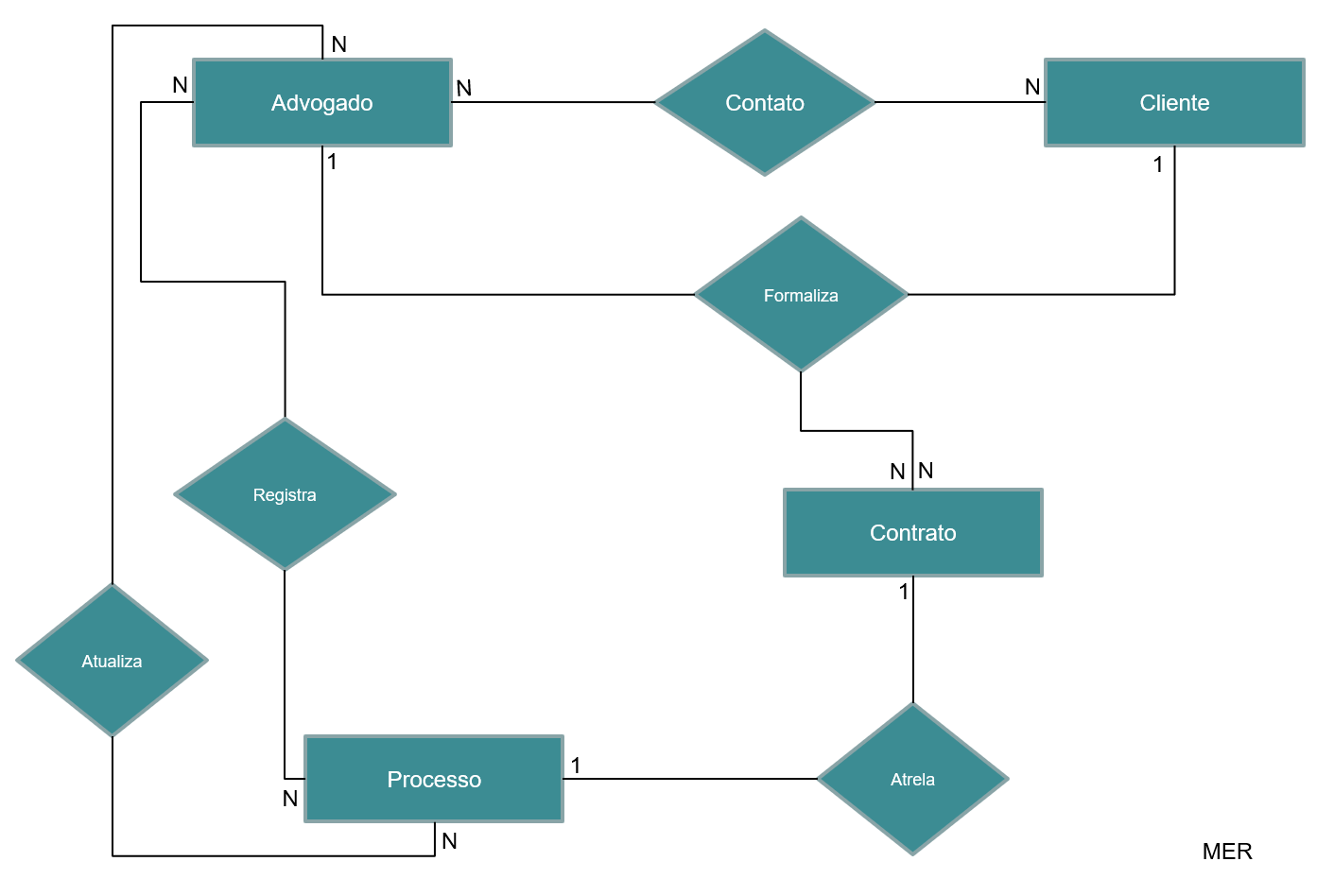
**Cardinalidade:**

(...)um advogado pode atender vários clientes(...)

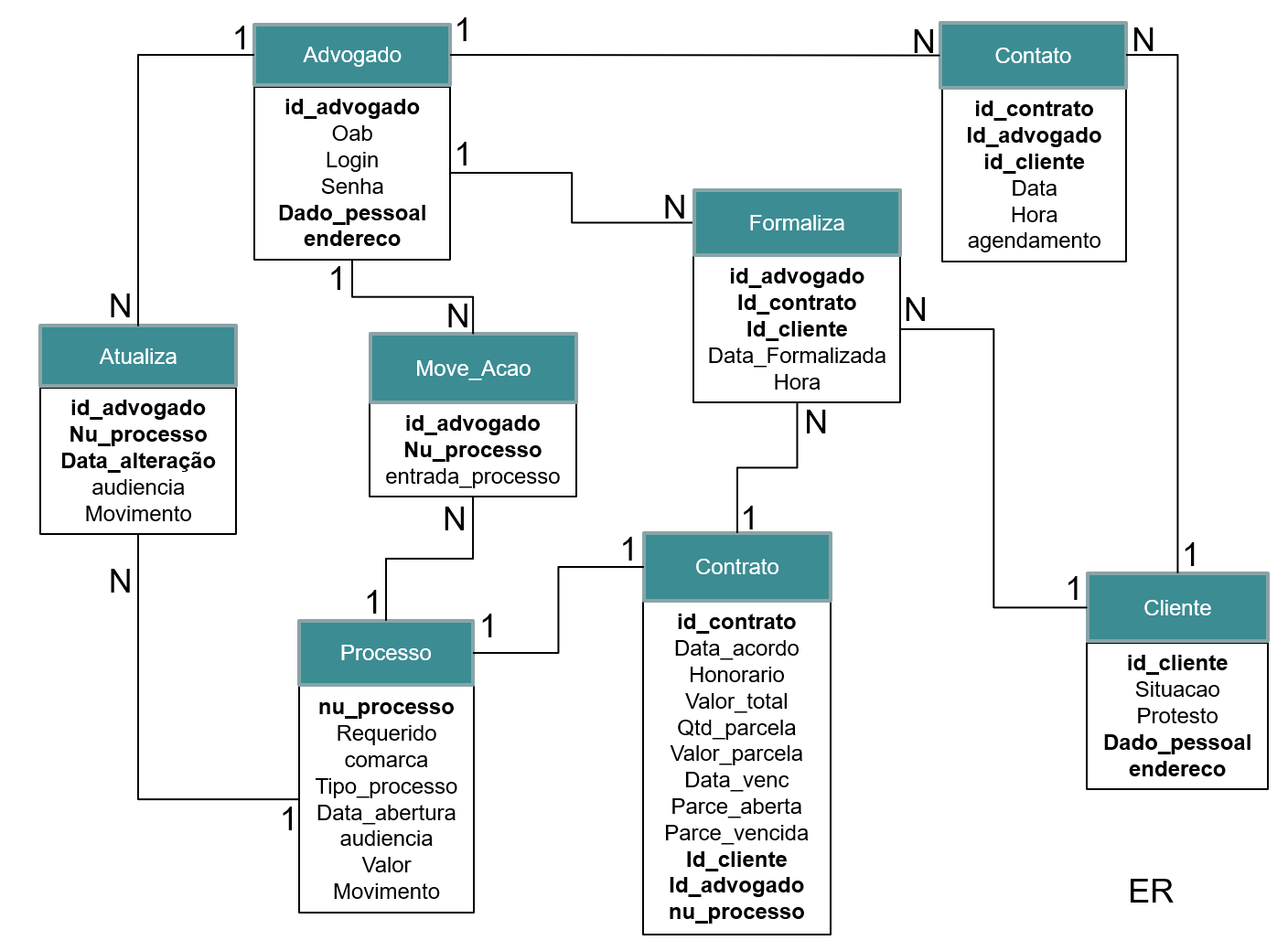
(...)um cliente pode ser atendido por vários advogados(...)

(...)um contrato pode ser formalizado por um ou mais advogado, porem só pode ser assinado e vinculado a um único cliente(...)

(...)Os advogados abrem várias ações nos fóruns, essas ações podem ser vinculadas a um ou mais advogados(...)



* + 1. **Entidade Relacional (ER)**

****

* + 1. **Normatização 1FN**

Foram criadas mais 4 tabelas além das 4 existentes.

Para a primeira forma normal, não tentamos em nenhum momento acabar com a redundância nos dados e sim buscar a relação entre as entidades. Observando as tabelas e relações, irá notar que não estão com todos os dados relacionados, foi por questões de limitação de software. Mas como a 1FN, relacionará tudo de cada chave.

**Pessoa (nova tabela)** {@id\_pessoa, nome, nacionalidade, civil, data\_nasc, email, cpf, rg e telefone}

**Endereco (nova tabela)** {@id\_endereco, rua, número, complemento, bairro, cep, cidade e estado}

**Contato (nova tabela)** {@id\_advogado, @id\_cliente, nome\_adv, data, hora, nome\_age, info\_age}

**Obs.:** O nome do advogado será definido pelo cliente ou pessoa quer agendar, nome da pessoa que solicitou reunião e uma informação breve do que foi conversado. Essa tabela é de registro momentâneo, onde o advogado pode excluir após a conclusão.

**Contrato** {@id\_contrato, data\_acordo, valor\_total, qtd\_parcelas, qtd\_entrada, valor\_entrada, valor\_demais\_parcela, data\_venc, parce\_aberta, parce\_vencida, situacao, protesto, @id\_cliente, @id\_advogado, @nu\_processo}

**Processo** {@nu\_processo, requerido, comarca, tipo\_processo, data\_abertura, data\_alteracao, audiência, valor, movimento}

**Advogado** {@id\_advogado, oab, login, senha}

**Cliente** {@id\_cliente, funcao, representante}

**Obs.:** a função informada, é caso o cliente esteja trabalhando que por sua vez registrado junto do contrato.

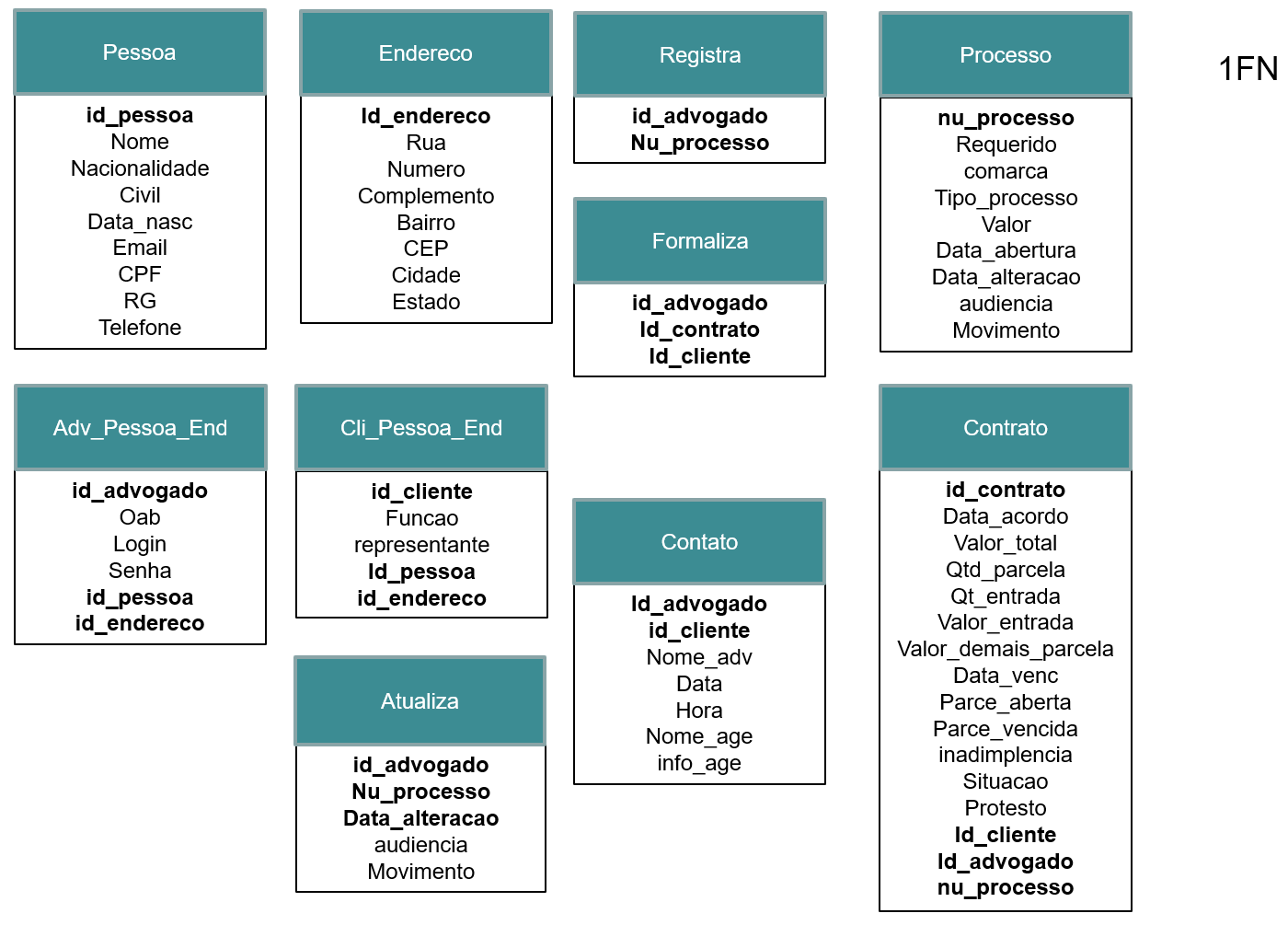
**Pessoa\_Advogado\_Endereco (cadastro-tabelas relacionas)** {@id\_advogado, oab, login, senha, @id\_pessoa, nome, nacionalidade, civil, data\_nasc, email, cpf, rg, telefone @id\_endereco, rua, número, complemento, bairro, cep, cidade e estado}

**Pessoa\_Cliente\_Endereco (Cadastro-tabelas relacionas)** {@id\_cliente, funcao, represetante, @id\_pessoa, nome, nacionalidade, civil, data\_nasc, email, cpf, rg, telefone @id\_endereco, rua, número, complemento, bairro, cep, cidade e estado}

**Registra (relação de tabelas)** {@id\_advogado, nome, @nu\_processo, requerido, comarca, tipo\_processo, data\_abertura, data\_alteracao, audiência, valor, movimento}

**Atualiza (nova tabela)** {@id\_advogado, nome, @nu\_processo, data\_alteracao, audiência, movimento}

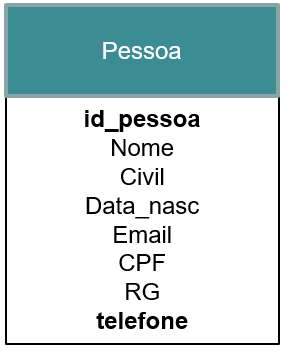
**Formaliza (relação de tabelas)** {@id\_contrato, Data\_formalizada, hora, @id\_advogado, @id\_cliente, nu\_processo}

****

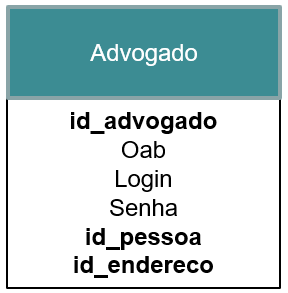
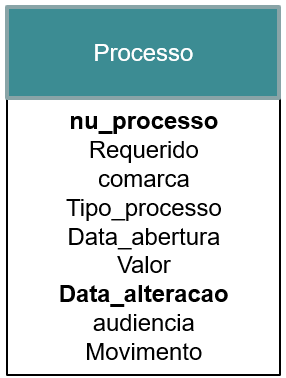
* + 1. **Normatização 2FN**

Usando a segunda forma normal para remover redundância de dados e ambiguidade.

Começando pelo telefone que ficará multivalorado, utilizando o próprio campo como chave-primaria e a id pessoa como chave estrangeira. Sendo assim...



Na parte de processo, houve a necessidade de criar uma nova tabela onde serão feitas diariamente as atualizações no andamento do processo. Utilizando a data da alteração como chave-primaria e as outras duas como chave estrangeira.



* 1. **Criação Física do Modelo de Dados**
  2. **Dicionário de Dados**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pessoa** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| id\_pessoa | determinante | PK |  |  |
| nome | simples | String | 50 | texto |
| nacionalidade | simples | String | 15 | texto |
| civil | simples | String | 15 | texto |
| datanasc | simples | date |  | formato dd/mm/aaaa |
| email | simples | String | 30 | texto |
| cpf | simples | int | 10 | formato XXX.XXX.XXX-XX |
| rg | simples | int | 9 | formato XX.XXX.XXX-X |
| telefone | multivalorado | String | 12 | texto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endereço** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| id\_endereco | determinante | PK |  |  |
| Rua | simples | String | 50 | texto |
| numero | simples | int | 5 |  |
| compl | simples | String | 20 | texto |
| bairro | simples | String | 30 | texto |
| CEP | simples | int | 8 | formato XXXXX-XXX |
| cidade | simples | String | 25 | texto |
| estado | simples | String | 25 | texto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Processos** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| nu\_processo | determinante | PK |  |  |
| requerido | simples | String | 50 | texto |
| comarca | simples | String | 20 | texto |
| tipo\_processo | simples | String | 15 | Será opção de 4 tipos de processo |
| valor | simples | String | 9 | texto |
| data\_abertura | simples | Date |  | formato dd/mm/aaaa |
| data\_alteracao | simples | FK |  | formato dd/mm/aaaa |
| audiencia | simples | Date |  | formato dd/mm/aaaa |
| movimento | simples | String | 500 | Advogado redigirá o que foi movimentado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Contrato** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| id\_contrato | determinante | PK |  |  |
| data\_acordo | simples | date |  | formato dd/mm/aaaa |
| valor\_total | simples | double | 10 | valor fracionado |
| qtd\_parcela | simples | int | 2 |  |
| qtd\_entrada | simples | int | 1 |  |
| valor\_entrada | simples | double | 10 | valor fracionado |
| valor\_demais\_parc | simples | double | 10 | valor fracionado |
| data\_venc | simples | date |  | formato dd/mm/aaaa |
| parc\_aberta | simples | int | 1 | irá mostrar se tem parcelas a vencer |
| parc\_vencida | simples | int | 2 | mostrar quantas parcelas atrasadas |
| situacao | simples | String | 10 | Selecionará se está em dia ou atrasado |
| protesto | simples | String | 3 | Selecionará entre SIM/NÃO |
| id\_cliente | estrangeira | FK |  |  |
| id\_advogado | estrangeira | FK |  |  |
| nu\_processo | estrangeira | FK |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Advogado** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| id\_advogado | determinante | PK |  |  |
| OAB | simples | String | 8 | formato XXX.XXX/(ESTADO ABREVIADO) |
| Login | simples | String | 15 | texto |
| Senha | simples | String | 10 | texto |
| id\_pessoa | estrangeira | FK |  |  |
| id\_endereco | estrangeira | FK |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cliente** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| id\_cliente | determinante | PK |  |  |
| funcao | simples | String | 50 | texto |
| representante | simples | String | 50 | texto |
| id\_pessoa | estrangeira | FK |  |  |
| id\_endereco | estrangeira | FK |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Atualiza** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| id\_advogado | estrangeira | FK |  |  |
| nu\_processo | estrangeira | FK |  |  |
| data\_alteracao | determinante | PK |  |  |
| audiencia | simples | Date |  | formato dd/mm/aaaa |
| movimento | simples | String | 500 | Advogado redigirá o que foi movimentado |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Contato** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| id\_advogado | estrangeira | FK |  |  |
| id\_cliente | estrangeira | FK |  |  |
| nome\_adv | estrangeira | FK |  |  |
| data | determinante | PK |  |  |
| hora | simples | String |  |  |
| nome | simples | String |  | texto |
| informa | simples | String |  | texto |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Telefone** | | | | |
| **Atributo** | **Classe** | **Domínio** | **Tamanho** | **Descrição** |
| id\_pessoa | estrangeira | FK |  |  |
| telefone | determinante | PK | 12 | formato (0XX) XXXXX-XXXX |

1. **Ambiente de Desenvolvimento**

No desenvolvimento do software, foi utilizado o programa VS Code da Microsoft para a linguagem Java no desenvolvimento da estrutura de código(algoritmo), Microsoft PowerPoint para criação do Modelo de Entidade Relacional, Banco de Dados MySQL Server(futuramente na criação do BD), uso do Microsoft Excel para a modelagem do banco de dados, um microcomputador( com processador ryzen 3 com 8GB de memória, HD SSD de 500GB e uma placa gráfica dedicada) e uma conexão na internet estável para pesquisas, interação com os desenvolvedores e os *stakeholders*.