

Sistema de informação para controle de agendamento de tarefas para escritórios de advocacia

Alex Hyacon Carvalho de Castro¹, Guilherme Chaga Kurtz¹

¹Sistema de Informação – Universidade Franciscana (UFN)
Caixa Postal 97010-491 – Santa Maria – RS – Brasil

hyacon@hotmail.com, guilhermekurtz@ufn.edu.com

Abstract. *The work deals with the development of software developed for task control for law firms that want to organize their internal services, by scheduling clients and employees. The work developed presents the control of people registrations, file storage and scheduling of employees' deadlines with their respective clients and deadlines, using the Fullcalendar Application Programming Interface. In developing this project, a resource-oriented development methodology was used. Using the Hypertext Preprocessor language together with the Laravel framework and the MySQL Database Management System.*

Resumo. *O trabalho trata do desenvolvimento de um software desenvolvido para controle de tarefas para escritórios de advocacia que deseja organizar seus serviços internos, por meio do agendamento de clientes e funcionários. O trabalho desenvolvido apresenta o controle de cadastros de pessoas, armazenamento de arquivos e agendamentos de prazos de funcionários com seus respectivos clientes e prazos, fazendo uso da Application Programming Interface Fullcalendar. No desenvolvimento deste projeto foi utilizada a metodologia Feature Driven Development. Com o uso da linguagem Hypertext Preprocessor juntamente com o framework Laravel e o Sistema Gerenciamento de Banco de Dados MySQL.*

1. Introdução

Na área do Direito existem muitas tarefas manuais envolvendo papéis impressos e pilhas de processos. O escritório de advocacia tem como principal atraso as pequenas tarefas manuais, com isso, um sistema automatizado ganha bastante influência dentro de uma empresa, podendo agilizar as pequenas e grandes tarefas através da análise das gestões aplicadas dentro de um escritório, fazendo com que haja um melhor controle dos funcionários e dos interesses da empresa.

Um sistema de controle de tarefas tem, por exemplo, a função de auxiliar a empresa e seus funcionários nas necessidades de controle de prazos, já que o próprio sistema mostrará as datas e os afazeres de cada funcionário.

1.1. Justificativa

Com as pesquisas realizadas para este projeto foi observado que os *softwares* desenvolvidos para escritórios de advocacia têm pouco desenvolvimento para controle de tarefas administrativas, assim, a aplicação é pensada na organização de tarefas para escritório que tem diversas áreas de atuação, tendo como desenvolvimento um sistema *web* para que possa dar um melhor acompanhamento nas áreas de atuação da empresa.

1.2. Objetivo Geral

Este trabalho tem como objetivo geral o desenvolvimento de um sistema *web* para escritórios de advocacia que buscam melhorar suas atividades internas, como: controle operacional, controle de dados financeiros, controle administrativo, entre outros. Para o mercado de *software* voltado para escritórios de advocacia são apresentadas as principais funcionalidades, tais como controles processuais (que tem como acompanhamento das etapas dos processos), juntamente com controles de documentos, de arquivos, de diligências, de prazos, de audiências, etc.

1.3. Objetivos Específicos

A partir do objetivo geral, tem-se os seguintes objetivos específicos:

- Realizar um estudo sobre sistemas de gestão de escritórios de advocacia e trabalhos relacionados com o mesmo.
- Utilizar para este projeto a metodologia Feature Driven Development (FDD) com base nos levantamentos da pesquisa.
- Apresentar a estrutura do sistema contemplando requisitos, diagramas e funcionalidades.
- Desenvolver um sistema *web* com o uso da linguagem Hypertext Preprocessor (PHP).
- Utilizar o *framework* Laravel para desenvolvimento das interfaces.
- Utilizar a Application Programming Interface (API) FullCalendar para desenvolvimento dos agendamentos.
- Utilizar do Sistema Gerenciamento de Banco de Dados (SGBD) MySQL para consistência dos dados.

1.4. Estrutura do Trabalho

Na Seção 2 é apresentado o referencial teórico e as tecnologias que foram utilizadas para elaborar este projeto. Na Seção 3 são apresentados os trabalhos relacionados que foram de grande ajuda para o desenvolvimento deste projeto. Na Seção 4 é apresentada a proposta de desenvolvimento do projeto e suas funcionalidades no sistema. Na Seção 5 é apresentada a metodologia que foi utilizada para o desenvolvimento do projeto e o desenvolvimento dos diagramas. Por fim, na Seção 6 são apresentados os resultados do desenvolvimento do projeto.

2. Referencial Teórico

Nesta seção são apresentadas as estruturas do sistema com base na organização dos escritórios de advocacia em relação às suas organizações de tarefas e áreas como financeira, administrativa e operacional, é utilizado para montar a aplicação para dentro de um escritório. É apresentado também o mercado de *software* para escritórios de advocacia, o uso do *framework*, das linguagens e do sistema de gerenciamento de banco de dados que será utilizado.

2.1. Sistema de Informação

Um sistema busca elaborar pontos estratégicos para um melhor controle de prazos e monitoramento de pessoas e informações dentro de uma empresa, assim, a gestão é aplicada aos pontos de controle de tarefas, financeiros e de pessoas. A aplicação acaba trazendo melhoramento de processos internos, redução de custos e otimiza o acesso aos dados internos.

2.2. Sistema para controle de agendamento de escritórios de advocacia

De acordo com Angelini (2018) como é típico de toda organização, os escritórios de advocacia possuem três atributos básicos: um **propósito** (realização de sonhos e objetivos); **divisão do trabalho** (cada pessoa e cada grupo têm papéis específicos que convergem para a realização da missão do escritório) e algum tipo de **coordenação** (feita mediante hierarquia, planejamento e comunicação).

Para uma melhor prática dentro de uma empresa de escritório de advocacia são substituídos os papéis pelos meios tecnológicos, fazendo acompanhamentos dos dados financeiros, do controle de tarefas, de funcionários e melhorando o fluxo dentro da empresa e suas respectivas áreas de atuação.

Os escritórios de advocacia dependem muito de datas e prazos a serem cumpridos. Para que um escritório de advocacia esteja com um bom funcionamento, é necessário que estes sejam administrados corretamente e bem organizados. Existem diferentes tipos de gestões em um escritório de advocacia, tais como gestão administrativa, gestão financeira e gestão operacional [Angelini et al. 2018].

2.2.1. Sistema de controle administrativo

Dentro da empresa é atribuída uma tarefa para cada funcionário de acordo com o conhecimento da sua área, assim, a gestão administrativa estabelece uma organização onde mostrará qual é a função exercida por cada um dos funcionários, mantendo ao mesmo tempo o trabalho em grupo e o desenvolvimento em conjunto. Desta forma, torna-se possível que todos possam acompanhar o decorrer de um processo jurídico.

Dentro do Direito existem prazos que precisam ser cumpridos e estes, além de envolverem as fases de um processo, envolvem outras tarefas, como as idas às audiências ou compromissos agendados; estes prazos e tarefas, que precisam ser realizados, são apresentados pelo sistema por meio de um calendário para que todos da equipe acompanhe as etapas [Luciano Pelizari 2010].

2.2.2. Sistema de controle financeira

A gestão financeira tem como objetivo o controle de caixa da empresa, podendo apresentar os gastos e valores recebidos pelo cliente. Tem como finalidade auxiliar a empresa no controle de informações demonstrando o resultado mensal. Deste modo, a empresa não tem a necessidade de calcular os resultados mensais e pode obter um controle de seu lucro, já que tudo isso é automatizado pelo próprio sistema, precisando apenas colocar os dados de valores para aqueles que forem despesa ou receita da empresa [Angelini et al. 2018].

2.3. Mercado de *software* desenvolvidos para escritórios de direito

Atualmente o mercado de *software* voltado para escritórios de Direito vem sendo cada vez mais integrado com a tecnologia, buscando facilitar o acesso aos dados de clientes, funcionários, documentos, agendamentos e dados financeiros. Nesse mercado estão sendo aplicadas diversas melhorias, como unificação de dados, uso de *chatbot*, calculadora de prazos, entre outros.

O Tribunal de Justiça do Estado do Rio Grande do Sul (TJRS), por exemplo, oferece aos operadores do direito o acesso ao sistema *eproc* (2020), em que podem ter o acompanhamento dos processos eletrônicos, assim, unifica os processos em um sistema.

Outro exemplo é o trabalho de Projuris (2020) que através de seu *software* apresenta atendimento ao cliente com a utilização de *chatbot*, controle de escritório, entre outras funcionalidades bem atuais que se relacionam com este trabalho.

O *software* DataJuri (2020), por sua vez, tem relação com este projeto por apresentar controle de processos, controle de clientes, controle de contratos, etc.

E, por fim, o *software* Harpa Web (2020) apresenta aplicação de controle nas funcionalidades da empresa, como gestões financeiras, de processos, e outros.

2.4. Ferramentas de desenvolvimento

Nesta seção será explicado brevemente o uso das ferramentas que foram utilizadas no desenvolvimento deste trabalho, bem como a metodologia de desenvolvimento que é utilizada no projeto.

2.4.1. Hypertext Markup Language (HTML)

É uma linguagem desenvolvida para desenvolvimento de sites, tendo como acrônimo HTML do inglês Hypertext Markup Language. O HTML se utiliza de marcações para diferenciar o que será código ou textos e afins, sendo uma linguagem de fácil entendimento usada por qualquer site [HTML 2020].

2.4.2. Hypertext Preprocessor (PHP)

Se trata de uma linguagem de *script open source*, desenvolvida para programação *web*, e pode ser usada juntamente com HTML. O PHP tem como diferencial de execução no servidor, sendo todo o código executado dentro do servidor onde é gerado o HTML que é enviado para o navegador. O PHP pode ser utilizado de várias formas, já que sua linguagem oferece recursos de conexão com banco de dados, manipulação de dados, criptografias de dados, manipulação de arquivos, entre outros [PHP 2020].

Com os conceitos apresentados o PHP foi escolhido como linguagem para acompanhar o *framework* Laravel, e por ser uma linguagem de fácil entendimento, podendo fazer a conexão com o sistema de gerenciamento de banco de dados MySQL.

2.4.3. Cascading Style Sheets (CSS)

O CSS foi desenvolvido para dar aparência as páginas de internet que se utilizam das linguagens de marcação HTML, XHTML e XML. O CSS tem como finalidade definir onde cada elemento do código, qual estilo e qual formato irá aparecer [CSS 2020].

2.4.4. Sistema de Gerenciamento de Banco de Dados MySQL

O Sistema de Gerenciamento de Banco de Dados MySQL, desenvolvido pela empresa MySQL AB, é lançado em 1995 e trata-se de uma banco de dados relacional com código aberto, desenvolvido para gerenciar as bases de dados, sendo utilizado a linguagem *Structure Query Language*(SQL) para manipulação de inserções, dar acesso a tabelas e gerenciar conteúdo armazenado no banco de dados [MySQL 2020].

Por ser bastante usado em aplicações *web* este gerenciador foi escolhido por ter compatibilidade com a linguagem PHP, podendo dar permissões a usuários ou removê-las, também pode ser realizado auditorias de acesso para rastrear as atividades de acesso e tendo um bom desempenho em aplicações *web* com utilização de index.

2.4.5. Laravel

Lançado em 2011, por Taylor B. Otwell, Laravel é um *framework open source*, que auxilia no desenvolvimento de sites que se utiliza de bibliotecas. O *framework* Laravel se utiliza do padrão de projeto de *software* Model View Controller (MVC), onde o usuário requisita dados do banco, sendo dividido em: requisição ao controlador (controller) que envia esses dados de requisição para o modelo (model) onde é realizado as buscas no banco de dados e retornadas ao controlador que é enviada à visão (view) que elabora a interface do usuário, mostrando na interface os resultados. O que torna mais fácil a manutenção de códigos, deixa a aplicação mais leve e é compatível com a linguagem PHP [Laravel 2020].

2.4.6. Bootstrap

Lançado em 2011, se trata de um framework desenvolvido para auxiliar na parte de *front-end* com código aberto, tornando o site responsivo sem que necessite mexer em linhas de código CSS. Podendo interagir facilmente com HTML, CSS e JavaScript [Bootstrap 2020].

2.4.7. FullCalendar

FullCalendar é uma API que auxilia no desenvolvimento de projetos que precisem utilizar um calendários em sua aplicação, separado em dias e organizado onde os eventos podem ser vistos de acordo com as respectivas datas [FullCalendar 2020].

2.4.8. JavaScript

JavaScript é uma linguagem de programação criada pela equipe Pluralsight, sendo possível utilizar juntamente com HTML e CSS. Sendo uma linguagem baseada em objetos, é possível utilizar várias funções disponíveis para aplicações web ou fora do ambiente, sem navegador [JavaScript 2020].

2.4.9. jQuery

jQuery é uma biblioteca JavaScript criada por John Resig em 2006, que foi desenvolvida para interpretar os scripts no navegador do cliente, podendo ser possível fazer uso de manipulação de documentos HTML, manipulação de eventos, animação e AJAX [jQuery 2020].

3. Trabalhos Relacionados

Nesta seção são apresentados os trabalhos que foram utilizados como pesquisa e tiveram grande relevância neste projeto, ajudando no desenvolvimento de novas ideias e trazendo estudos de casos em que tiveram resultados importantes.

3.1. Automação de escritório de advocacia: sistema columna

O projeto apresentado por Juliano Pilatti (2006) propôs um *software* para automação de escritórios de advocacia com o uso da linguagem Delphi, que faz o uso de *Short Message Service* (SMS) para envio de mensagens para clientes.

É apresentado o uso de sistemas, como *Borland Delphi 7* e o Sistema Gerenciador de Banco de Dados InterBase.

Foi percebido um grande problema de relação com o cliente, em que o mesmo demorava a obter informações no decorrer do processo, em vista disto, foi desenvolvido o *software* para que o cliente pudesse receber informações do processo atualizado mais rapidamente.

O *software* foi desenvolvido juntamente com um escritório de advocacia, sendo implantado e testado. Trata-se de um *software* de baixo custo e com melhoras de desempenho por ter um fluxo de informação menor que os *softwares* do mercado.

3.2. Sistema de gerenciamento de escritório de advocacia

O projeto de Luciano Pelizari (2010) tem como proposta, desenvolver um *software* para advocacia simples, oferecendo como serviço as funcionalidades de um escritório de advocacia.

O *software* teve como uso de modelagem a Linguagem de Modelagem Unificada (UML) para o sistema, a linguagem de programação Java juntamente com a IDE NetBeans, o *framework* Hibernate para a persistência de dados, e o PostgreSQL para armazenamento das informações e o uso da ferramenta iReport para criação de relatórios.

Softwares para a área jurídica podem ser encontrados, porém há um inconveniente, cada um tem uma finalidade específica. O fato de se ter que instalar diversos *software* e alternar entre eles para que se possa ter um único objetivo, é algo um pouco desagradável.

A ideia deste sistema, veio do objetivo de poder resolver este inconveniente, desenvolvendo um único *software* que concentre diversas funcionalidades, sendo de fácil utilização através de uma interface amigável.

3.3. Conclusão sobre os trabalhos relacionados

Os trabalhos escolhidos têm como temas correlatos com o desenvolvimento deste projeto onde demonstram o desenvolvimento de um projeto na área jurídica, onde tem como aplicação em escritórios de advocacia.

O trabalho de Juliano Pilatti (2006) relaciona-se com este trabalho pois propõe o desenvolvimento de um sistema voltado para escritórios de advocacia, fazendo o uso de

SMS. Este trabalho se assemelha por ter grande relevância como estudo para escritórios de advocacia.

O trabalho de Luciano Pelizari (2010) tem como relação com este trabalho pois propõe a simplificação do uso de um *software* voltado para escritórios de advocacia. Se assemelhando com este projeto como um desenvolvimento para escritórios de advocacia.

Todos os trabalhos foram de grande ajuda como estudo para o desenvolvimento e conhecimento deste projeto, tendo como destaque neste projeto o desenvolvimento de um sistema voltado para escritório de advocacia abrangendo notificações para usuários, controle de gestões.

4. Proposta

Nesta seção é apresentado o funcionamento do *software*, juntamente com a explicação de suas funcionalidades, tendo em vista o foco do trabalho para o melhoramento de controle de agendamentos dos prazos de processos dentro de escritórios de advocacia e como o sistema será disponibilizado para o usuário. O sistema apresenta uma agenda, mostrando as datas agendadas para cada cliente ou processo e seus prazos; controle de gastos e ganhos da empresa; controle de clientes, onde é mostrado os dados pessoais de cadastro do cliente; onde o usuário poderá armazenar arquivos, peças, documentos escaneados ou qualquer arquivo.

4.1. Controle de pessoas

O controle de pessoas é dividido em duas partes, contendo cadastro de clientes e cadastro de usuários. Para clientes será inserido nome, sobrenome, endereço (CEP, complemento, bairro, cidade), CPF/CNPJ, RG, telefone e e-mail e para funcionários será inserido nome, sobrenome, e-mail, telefone, senha, endereço (CEP, complemento, bairro, cidade). O funcionário pode fazer buscas de pessoas já cadastradas.

4.2. Controle Financeiro

O controle financeiro demonstra as despesas e receitas, cada item a ser inserido contendo nome do item, descrição e valor, podendo realizar buscas.

4.3. Calendário e Tarefas

Esta funcionalidade apresenta um calendário que mostra o agendamento de cada dia e a duração de prazos e demais tarefas.

4.4. Anexar arquivos

O *software* dá a opção de anexar imagens, peças jurídicas de um processo (são as peças/partes de um processo que serão apresentadas ao juiz por parte do cliente) e escaneamentos.

5. Metodologia

Para o desenvolvimento deste trabalho foi utilizada a linguagem de programação PHP, JavaScript com utilização das bibliotecas jQuery e AJAX, como auxílio para front-end utilizando o Bootstrap, HTML, CSS, com a utilização do *framework* Laravel e a API FullCalendar, juntamente com o sistema de gerenciamento de banco de dados MySQL.

A metodologia que foi utilizada para a realização do projeto do sistema é o FDD, que é dividida em 5 etapas, apresentadas a seguir.

5.1. Desenvolvimento de modelo abrangente

No primeiro processo foi elaborado o escopo do projeto e as integrações que foram modeladas. O levantamento detalhado sobre o que será trabalhado para montar o projeto é fundamental. Para isso, é demonstrado o diagrama de domínio representado na Figura 1.

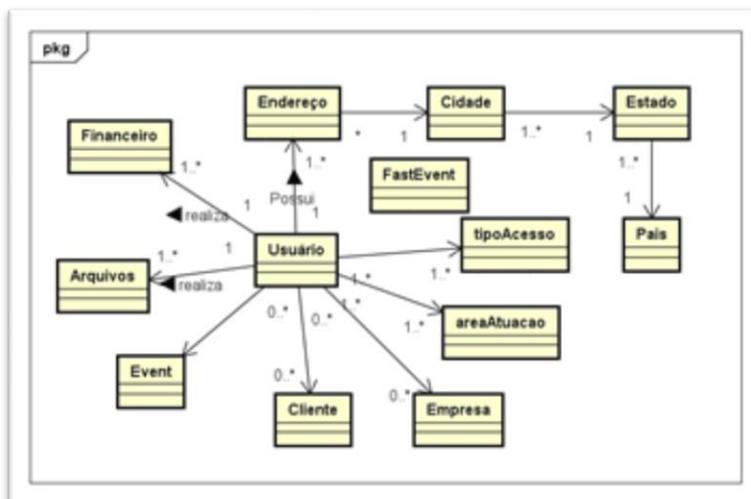


Figura 1. Diagrama de Domínio

5.2. Construção de lista de funcionalidade

As funcionalidades da área de negócio levantadas no primeiro processo foram utilizadas neste projeto seguindo os requisitos. Assim, são apresentados os requisitos funcionais e não funcionais.

RNF01. Utilização da Linguagem de Programação PHP

RNF02. Utilização do *Framework* Laravel

RNF03. Utilização do SGBD MySQL

RNF04. Utilização de Armazenamento de arquivos

RF01. Gerenciar Pessoa

RF01.1. Gerenciar Cliente

RF01.2. Gerenciar Usuário

RF01.3. Gerenciar Login

RF01.4. Gerenciar Histórico de Usuário

Definição: Módulo responsável por incluir, alterar, consultar e excluir os usuários do sistema e clientes cadastrados.

RF02. Gerenciar Calendário

RF02.1. Gerenciar Tarefas

RF02.2. Gerenciar Histórico Agenda

Definição: Módulo responsável por incluir, alterar, consultar e excluir as tarefas pendentes nas respectivas datas marcadas.

RF03. Gerenciar Financeiro

RF03.1. Gerenciar Financeiro

RF03.2. Gerar Cálculo Receitas e Despesas

RF03.3. Gerar Relatórios

RF03.4. Gerar Gráficos Financeiros

RF03.5. Gerenciar Histórico Financeiro

Definição: Módulo responsável por incluir, alterar, consultar e excluir as receitas ou despesas.

RF04. Gerenciar Arquivos

RF04.1. Gerenciar Arquivos

RF04.2. Gerenciar Histórico de Arquivos

Definição: Módulo responsável por incluir, alterar, consultar e excluir os arquivos.

5.3. Planejar por funcionalidades

Neste processo é elaborado o plano de desenvolvimento, em que é planejado, ordenado e calculado o quanto de desempenho será necessário para que as funcionalidades sejam implementadas. Desta forma, são apresentados os diagramas de caso de uso e descritivos de caso de uso. No apêndice A é apresentado o planejamento por funcionalidade.

No apêndice B, é apresentado o Diagrama de Caso de Uso representando o cenário das funcionalidades do sistema. São demonstrados nos apêndices C, D, E e F os descritivos de Caso de Uso dos usuários, financeiro, calendário e de documentos, onde são apresentadas as funcionalidades de inserção, alteração, exclusão e consulta.

5.4. Detalhar por funcionalidade

Neste processo são desenvolvidos os métodos do projeto baseados nas classes e diagramas desenvolvidos anteriormente com a apresentação do diagrama de atividade, diagrama de classe e diagrama de entidade e relacionamento, de acordo com o apêndice G.

No apêndice H é demonstrado o Diagrama de Classe em que representa as classes que serão usadas juntamente com suas propriedades (atributos e métodos).

No apêndice I é representado o Diagrama de Entidade Relacionamento usado para estrutura do Banco de Dados deste projeto em que representa as entidades que serão usadas.

5.5. Construção por funcionalidade

Nesta seção é demonstrado o desenvolvimento do *software* de acordo com os diagramas desenvolvidos nos processos anteriores. São apresentados na Figura 2 os comportamentos

do calendário, como: linguagem escolhida, as datas podem ser selecionadas, empilhamento de eventos em um mesmo dia, tornar um evento clicável, arrastar eventos do calendário e redimensionar eventos.

```
/* Inicialização Calendario
-----*/

var calendarEl = document.getElementById('calendar');
var calendar = new FullCalendar.Calendar(calendarEl, {
  headerToolbar: {
    left: 'prev,next today',
    center: 'title',
    right: 'dayGridMonth,timeGridWeek,timeGridDay,listWeek'
  },
  locale: 'pt-br', //linguagem escolhida
  navLinks: true, //Determina se os nomes dos dias e das semanas são clicáveis.
  dayMaxEventRows: true, //O número máximo de níveis de eventos empilhados em um determinado dia.
  selectable: true, //Permitir que o usuário faça seleções clicando e arrastando
  editable: true, //Determina se os eventos podem ser arrastados e redimensionados
  droppable: true. //Determina se os elementos externos arrastáveis ou eventos de outros calendários podem ser soltos no calendário
});
```

Figura 2. Script de inicialização do calendário

Na Figura 3 é apresentado o *script* de *update* quando um evento é arrastado, fazendo com que os dados sejam alterados no banco.

```
/*Ocorre quando um evento dentro do calendario e arrastado
eventDrop: function(element){

  let start = moment(element.event.start).format("YYYY-MM-DD HH:mm:ss");
  let end = moment(element.event.end).format("YYYY-MM-DD HH:mm:ss");

  let newEvent = {
    _method: 'PUT',
    title: element.event.title,
    id: element.event.id,
    optionUsuario: element.event.optionUsuario,
    optionCliente: element.event.optionCliente,
    start: start,
    end: end
  };

  sendEvent(routeEvents('routeEventUpdate'),newEvent);
}
```

Figura 3. Script do calendário quando o evento é arrastado

A Figura 4 apresenta o *script* quando um evento é selecionado abrindo um modal e mostrando as informações.

```

/ CLICKE QUANDO UM EVENTO É SELECIONADO E NÃO UMA CELDA
eventClick: function(element){
  clearMessages('.message');
  resetForm("#FormEvent");

  $("#modalCalendar").modal('show');
  $("#modalCalendar #titleModal").text('Alterar Evento');
  $("#modalCalendar button.deleteEvent").css('display', 'flex');

  let id = element.event.id;
  $("#modalCalendar input[name='id']").val(id);

  let title = element.event.title;
  $("#modalCalendar input[name='title']").val(title);

  let optionUsuario = element.event.optionUsuario;
  $("#modalCalendar option[name='optionUsuario']").val(optionUsuario);

  let optionCliente = element.event.optionCliente;
  $("#modalCalendar option[name='optionCliente']").val(optionCliente);

  let start = moment(element.event.start).format("DD/MM/YYYY HH:mm:ss");
  $("#modalCalendar input[name='start']").val(start);

  let end = moment(element.event.end).format("DD/MM/YYYY HH:mm:ss");
  $("#modalCalendar input[name='end']").val(end);

  let color = element.event.backgroundColor;
  $("#modalCalendar input[name='color']").val(color);

  let description = element.event.extendedProps.description;
  $("#modalCalendar textarea[name='description']").val(description);
}

```

Figura 4. Script do calendário quando o evento é selecionado

6. Resultados

Nesta seção são apresentados os resultados deste projeto, onde serão mostradas as suas funcionalidades e as dificuldades encontradas na sua elaboração.

O trabalho teve como principal foco desenvolver um sistema para escritório de advocacia em seu controle de tarefas de agendamento utilizando o *framework* Laravel, fazendo com que seja possível acessar um sistema que crie eventos, relacionando o funcionário ao cliente e seus respectivos documentos, assim, possibilitando ter fácil acesso a um calendário que armazena e apresenta os eventos de cada funcionário.

Por ser um framework em que o autor teve pouco contato, foi percebido uma demora na elaboração das interfaces apresentadas, mas apresentou ser de grande ajuda a estrutura de arquivos apresentada pelo modelo MVC.

As linguagens usadas neste projeto foram de grande importância, mas de grande dificuldade pois o calendário apresentado se utiliza de JavaScript, Ajax e JSON para as requisições ao banco, foi considerado uma linguagem mais trabalhosa pois precisa ter conhecimento de suas funções que interagem diretamente com os atributos oferecidos pelo navegador através do console, tendo uma estrutura de linguagem diferente das demais.

As interfaces desenvolvidas para este projeto são apresentadas, onde na Figura 5 é apresentada a interface de Login, a Figura 6 apresenta a interface inicial onde o gráfico demonstra o faturamento da empresa de acordo com o mês, na Figura 7 é apresentado o cadastro de usuário, para a Figura 8 é apresentada a tabela em que são listados os usuários cadastrados, na Figura 9 é apresentado o calendário, na Figura 10 é apresentado o modal

de cadastro de evento, e, por fim, na Figura 11 é demonstrada a lista de eventos cadastrados.

Na Tabela Financeiro (Apêndice J) é listado os dados financeiros já cadastrados, na interface de Cadastro Financeiro (Apêndice K) é apresentado os componentes de cadastro, na interface de Cadastro de Arquivos (Apêndice L) é apresentado os componentes de cadastro, a lista de arquivos já cadastrados pode ser vista na Lista de Arquivos (Apêndice M), mais Informações de Itens (Apêndice N), interfaces de Cadastro de Cliente (Apêndice O) e Tabela de Cliente (Apêndice P).



The image shows a login form titled "Login". It contains two input fields: "E-mail" and "Senha". Below the "Senha" field is a checkbox labeled "Lembrar" which is checked. At the bottom, there are three buttons: "Login" (green), "Cancelar" (red), and a link "Esqueceu sua senha?" (blue).

Figura 5. Interface de Login

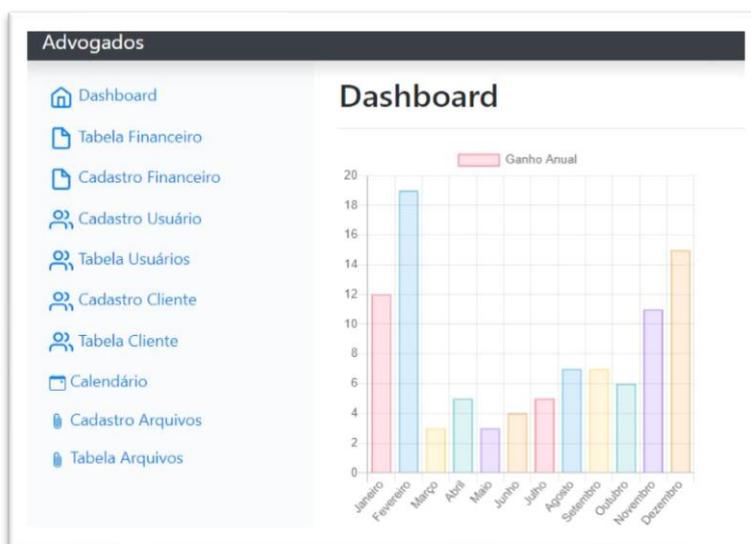


Figura 6. Interface Inicial

Cadastro de Usuário

Nome Sobrenome Data de Nascimento

RG CPF Senha

Email Telefone

Tipo de acesso Área de atuação

Foto
 Nenhum ficheiro selecionado

CEP Endereço Nº

Bairro Complemento

Pais Estado Cidade

Figura 7. Interface de Cadastro de Usuário

Tabela Usuário

Pesquisar

Id	Nome	Sobrenome	Data de Nascimento	
1	Fernando	Silva	1999-01-01	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>
2	Paulo	Sar	1992-01-01	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>
3	Guilherme	Salvio	1991-01-01	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>
4	João	Paulo	1992-02-02	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>
5	Rafael	Fernandes	1990-04-05	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>

< 1 2 >

Figura 8. Interface de Tabela Usuário

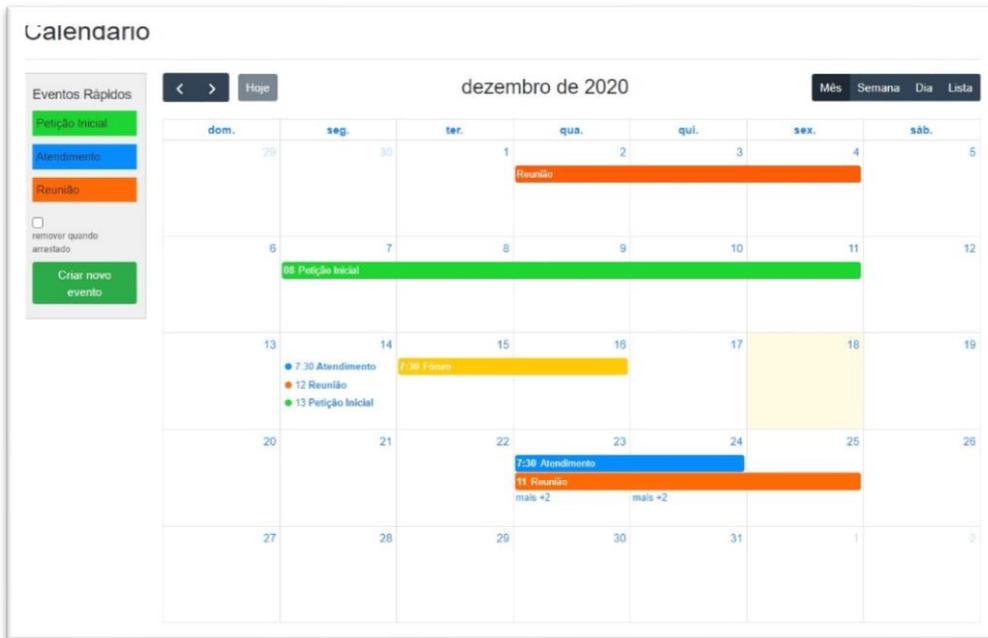


Figura 9. Interface de Controle de Eventos

Calendário

Alterar Evento

Título: Reunião

Usuário: Fernando Silva

Cliente: Miguel Alves

Data/hora Inicial: 07/12/2020 11:00:00

Data/hora Final: 09/12/2020 16:05:00

Cor do Evento:

Descrição: Reunião com cliente

Fechar Excluir Salvar

Draggable Events

- Reunião
- Petição Inicial
- Atendimento

remover quando arrastado

Criar novo evento

Figura 10. Interface Cadastro Eventos



Figura 11. Interface Lista de Eventos

7. Conclusão

Este trabalho teve como finalidade implementar um *software* para escritórios de advocacia que pretendem automatizar o agendamento de tarefas no seu dia-a-dia, melhorando através do agendamento de tarefas o acompanhamento de prazos de seu escritório, podendo ser possível fazer a distribuição de tarefas para seus funcionários, integrando o cliente e o funcionários a tarefas.

Foram realizadas pesquisas no ramo de *softwares* desenvolvidos para escritórios de advocacia, foi constatado que os trabalhos de *softwares* são mais voltados para a individualidade do escritório e pouco pensado no controle de tarefas de um escritório.

Os trabalhos correlatos foram de grande importância, pois trouxeram um grande conhecimento sobre este assunto, possibilitando a busca por novas abordagens e fazendo melhoramentos no projeto.

A metodologia FDD foi escolhida devido a possíveis melhoramentos e novas ideias que possam ser introduzidas posteriormente ao projeto. O *framework* Laravel foi escolhido devido ao seu modelo MVC e por ser uma ferramenta de grande ajuda e de muitas funcionalidades em suas bibliotecas.

O sistema tem como finalidade ajudar escritórios de advocacia, possibilitando o agendamento de tarefas, controle financeiro, controle de peças/arquivos e controle de clientes, juntamente com cadastros de usuários.

Para trabalhos futuros, é proposto que o trabalho apresentado tenha mais funcionalidades que se relacionem com o agendamento de tarefas e se utilize dos dados do usuário e cliente, trazendo mais opções de pesquisas para a agenda.

Referências

Angeline, F., Cunha, M., Macieira, M., Maranhão, M. “Sistema Integrado de Gestão para Escritório de Advocacia”. In Editora Scortecci, pag. 4-5, 2018.

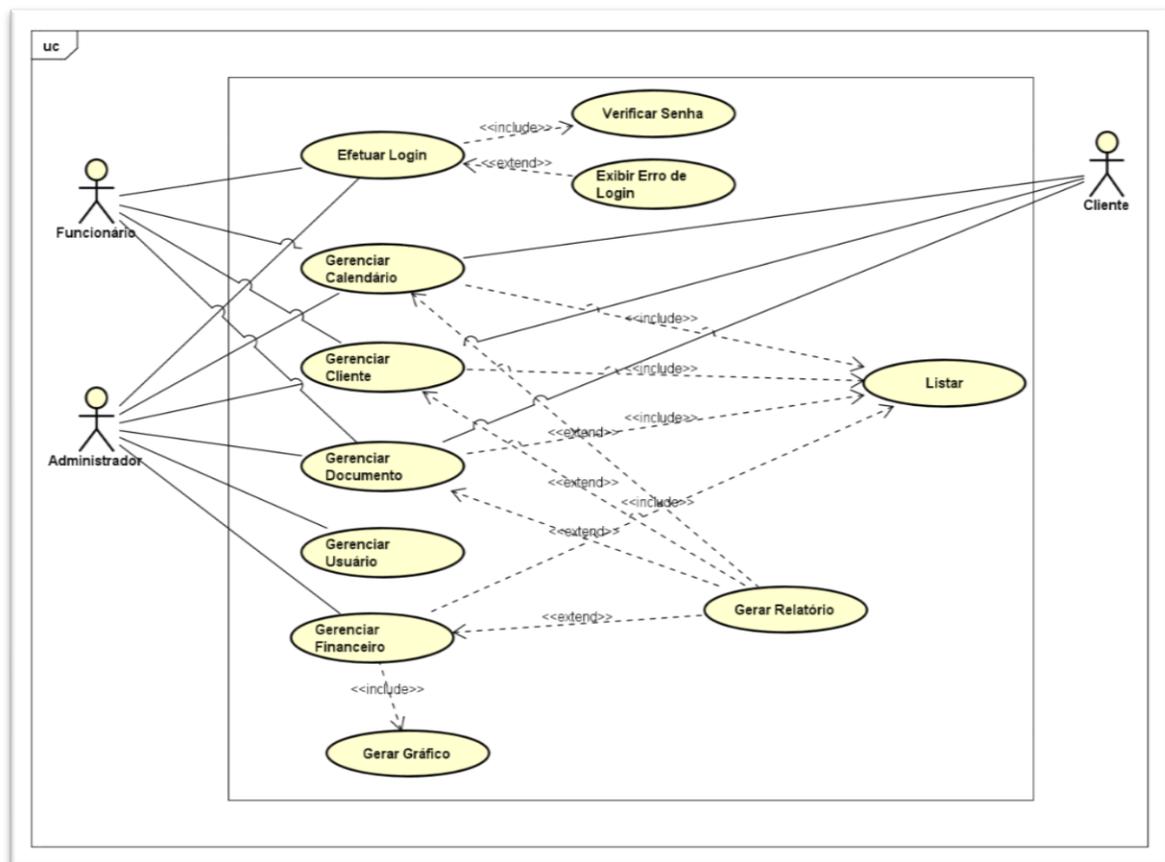
CSS. (2020) “CSS Documentation”. Disponível em: <https://www.w3.org/Style/CSS/>. Acesso em: Abril/2020.

- DataJuri, (2020) “Software Jurídico”. Disponível em: https://www.datajuri.com.br/?gclid=Cj0KCQjAs67yBRC7ARIsAF49CdXB2gImD_GDYolPJs0Nl-zzPYBcgNtsRvo1z14qu-zEq1i6VYrUTQEaAp28EALw_wcB. Acesso em: Abril/2020.
- Eproc, (2020) “Tribunal de Justiça do Estado do Rio Grande do Sul”. Disponível em: <https://www.tjrs.jus.br/novo/eproc/>. Acesso em: Abril/2020.
- FDD. (2020) “Feature Driven Development”. Disponível em: <http://www.featuredrivendevelopment.com/>. Acesso em: Maio/2020.
- Harpa Web. (2020) “Software Juridico Web”. Disponível em: http://www.harpaweb.com.br/?gclid=Cj0KCQjw9ZzzBRCKARIsANwXaeLd_ubylEHdW2mu-MUFdFiCbcI0eS5Rhp7otlmQOefMTAPU_Rq85UMaAoDxEALw_wcB. Acesso em: Abril/2020.
- HTML. (2017) “HTML Documentation”. Disponível em: <https://www.w3.org/TR/html52/>. Acesso em: Abril/2020.
- Pelizari, L. (2010) “Sistema de Gerenciamento de Escritório de Advocacia”. Disponível em: <https://cepein.femanet.com.br/BDigital/arqTccs/0611270050.pdf>. Acesso em: Maio/2020.
- Pilatti, J. (2006) “Automação de Escritório de Advocacia: Sistema Columna”. Disponível em: <http://dsc.inf.furb.br/arquivos/tccs/monografias/2006-1julianopilattivf.pdf>. Acesso em: Maio/2020.
- Laravel. (2020) “Laravel Documentation”. Disponível em: <https://laravel.com/docs/7.x>. Acesso em: Maio/2020.
- MySQL. (2020) “MySQL Documentation”. Disponível em: <https://dev.mysql.com/doc/>. Acesso em: Abril/2020.
- PHP. (2020) “Manual do PHP”. Disponível em: https://www.php.net/manual/pt_BR/intro-what-is.php. Acesso em: Abril/2020.
- ProJuris. (2020) “ProJuris Software Jurídico”. Disponível em: <https://www.projuris.com.br/escritorios/>. Acesso em: Abril/2020.
- Ramos, E. Robaski, J. R. (2014) “FDD (Feature Driven Development) Desenvolvimento Guiado por Funcionalidade”. Disponível em: <https://medium.com/@jrobaski/fdd-feature-driven-development-7d08c5c24c8f>. Acesso em: Maio/2020.
- Bootstrap. (2020) “Bootstrap Documentation”. Disponível em: <https://getbootstrap.com/>. Acesso em: Setembro/2020.
- JavaScript. (2020) “JavaScript About”. Disponível em: <https://www.javascript.com/about>. Acesso em: Setembro/2020.
- jQuery. (2020) “jQuery API”. Disponível em: <https://api.jquery.com/>. Acesso em: Setembro/2020.
- FullCalendar. (2020) “FullCalendar”. Disponível em: <https://fullcalendar.io/>. Acesso em: Setembro/2020.

APÊNDICE A. Planejamento por Funcionalidade

Ordem	Funcionalidade	Tempo de Desenvolvimento Estimado
1	RF01. Gerenciar Pessoa	30 dias
2	RF02. Gerenciar Calendario	60 dias
3	RF03. Gerenciar Financeiro	15 dias
4	RF04. Gerenciar Arquivos	15 dias

APÊNDICE B. Diagrama de Caso de Uso



APÊNDICE C. Descritivo de Caso de Uso Gerenciar Usuário

Identificação	CDU01
Caso de Uso	Gerenciar Usuário
Atores Principais	Administrador
Atores Secundários	N/A
Pré-Condições	O ator deve estar logado no sistema
Pós-Condições	N/A
Fluxo Principal	<p>P01 – O sistema exibe a interface de cadastro para usuários</p> <p>P02 – O ator deve inserir nome</p> <p>P03 – O ator deve inserir sobrenome</p> <p>P04 – O ator deve inserir data de nascimento</p> <p>P05 – O ator deve inserir rg</p> <p>P06 – O ator deve inserir cpf</p> <p>P07 – O ator deve inserir senha</p> <p>P08 – O ator deve inserir e-mail</p> <p>P09 – O ator deve inserir telefone</p> <p>P10 – O ator deve inserir tipo de acesso</p> <p>P11 – O ator deve inserir área de atuação</p> <p>P12 – O ator insere foto</p> <p>P13 – O ator deve inserir cep</p> <p>P14 – O ator deve inserir endereço</p> <p>P15 – O ator deve inserir número do local que mora</p> <p>P16 – O ator deve inserir bairro</p> <p>P17 – O ator deve inserir complemento</p> <p>P18 – O ator deve inserir país</p> <p>P19 – O ator deve inserir estado</p> <p>P20 – O ator deve inserir cidade</p> <p>P21 – O ator clica no botão “Cadastrar” para confirmar cadastro</p> <p>P22 – O sistema exibe na interface a mensagem de confirmação</p> <p>P23 – Fim do Caso de Uso</p>
Fluxo Alternativo	<p>[A01] – O ator cancela o cadastro</p> <p>A01 – O sistema limpa os campos</p> <p>A02.1 – O ator clica no botão “Cadastrar”, mas o e-mail já está cadastrado</p> <p>A02.2 – O sistema emite um alerta de e-mail já cadastrado</p> <p>A02.3 – O sistema não permite cadastrar o usuário</p>
Fluxo de Exceção	<p>E02 – O ator clica no botão “Cadastrar” faltando completar o campo</p> <p>E02 – O sistema exibe os campos obrigatórios que faltam completar</p>
Regra de Negócio	RN01 – O usuário deve ter permissão administrativa
Caso de Uso ou	N/A

Cenário Incluído	
Observação	N/A

APÊNDICE D. Descritivo de Caso de Uso Gerenciar Financeiro

Identificação	CDU02
Caso de Uso	Gerenciar Financeiro
Atores Principais	Administrador
Atores Secundários	N/A
Pré-Condições	O ator deve estar logado no sistema
Pós-Condições	N/A
Fluxo Principal	P01 – O sistema deve mostrar a interface do financeiro P02 – O ator deve clicar no botão “Inserir” para cadastrar item P03 – O ator deve inserir nome P04 – O ator deve inserir valor P05 – O ator deve selecionar o campo receita ou despesa P06 – O ator deve inserir descrição P07 – O ator deve inserir data P08 – O ator deve selecionar caso seja receita ou despesa P09 – Fim do Caso de Uso
Fluxo Alternativo	[A01] – O ator cancela o cadastro A01 – O sistema limpa os campos
Fluxo de Exceção	E02 – O ator clica no botão “Cadastrar” faltando completar o campo E02 – O sistema exhibe os campos obrigatórios que faltam completar
Regras de Negócio	RN01 – O usuário deve ter permissão administrativa
Caso de Uso ou Cenário Incluído	N/A
Observação	O sistema deverá listar os itens e mostrar graficamente o desempenho empresa

APÊNDICE E. Descritivo de Caso de Uso Gerenciar Calendário

Identificação	CDU03
Caso de Uso	Gerenciar Calendário

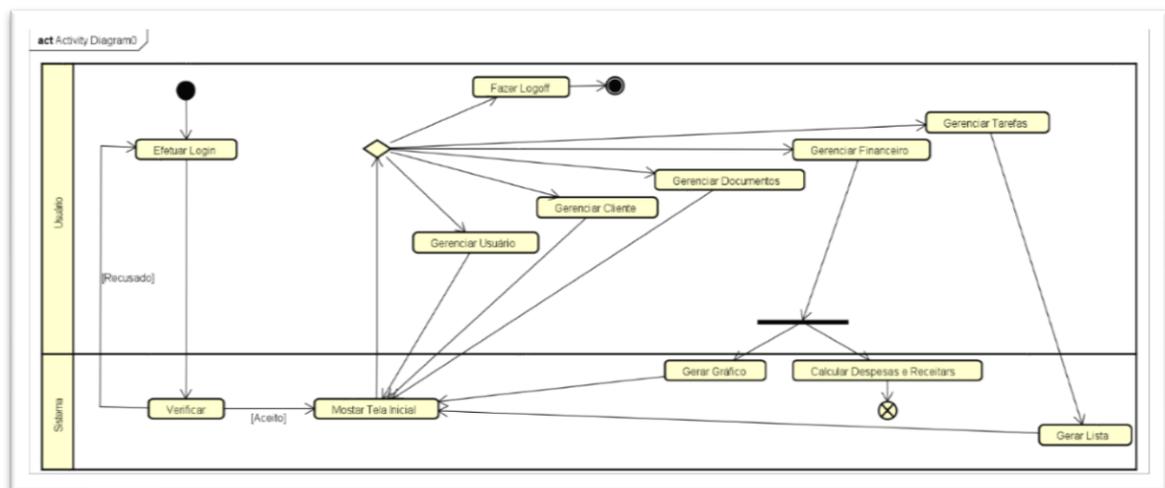
Atores Principais	Usuário
Atores Secundários	N/A
Pré-Condições	O ator deve estar logado
Pós-Condições	O sistema deve mostrar as tarefas em uma lista
Fluxo Principal	P01 – O sistema deve mostrar a interface de calendário P02 – O ator deve clicar em algum dia do calendário que irá inserir uma nova tarefa “Salvar” P03 – O ator deve inserir nome P03 – O ator deve selecionar o usuário P03 – O ator deve selecionar o cliente P04 – O ator deve inserir data inicial e final juntamente com horário P03 – O ator deve selecionar cor P05 – O ator deve inserir descrição P09 – Fim do Caso de Uso
Fluxo Alternativo	[A01] – O ator cancela o cadastro da tarefa A01 – Fim do caso de Uso
Fluxo de Exceção	N/A
Regra de Negócio	N/A
Caso de Uso ou Cenário Incluído	N/A
Observação	N/A

APÊNDICE F. Descritivo de Caso de Uso Gerenciar Documentos

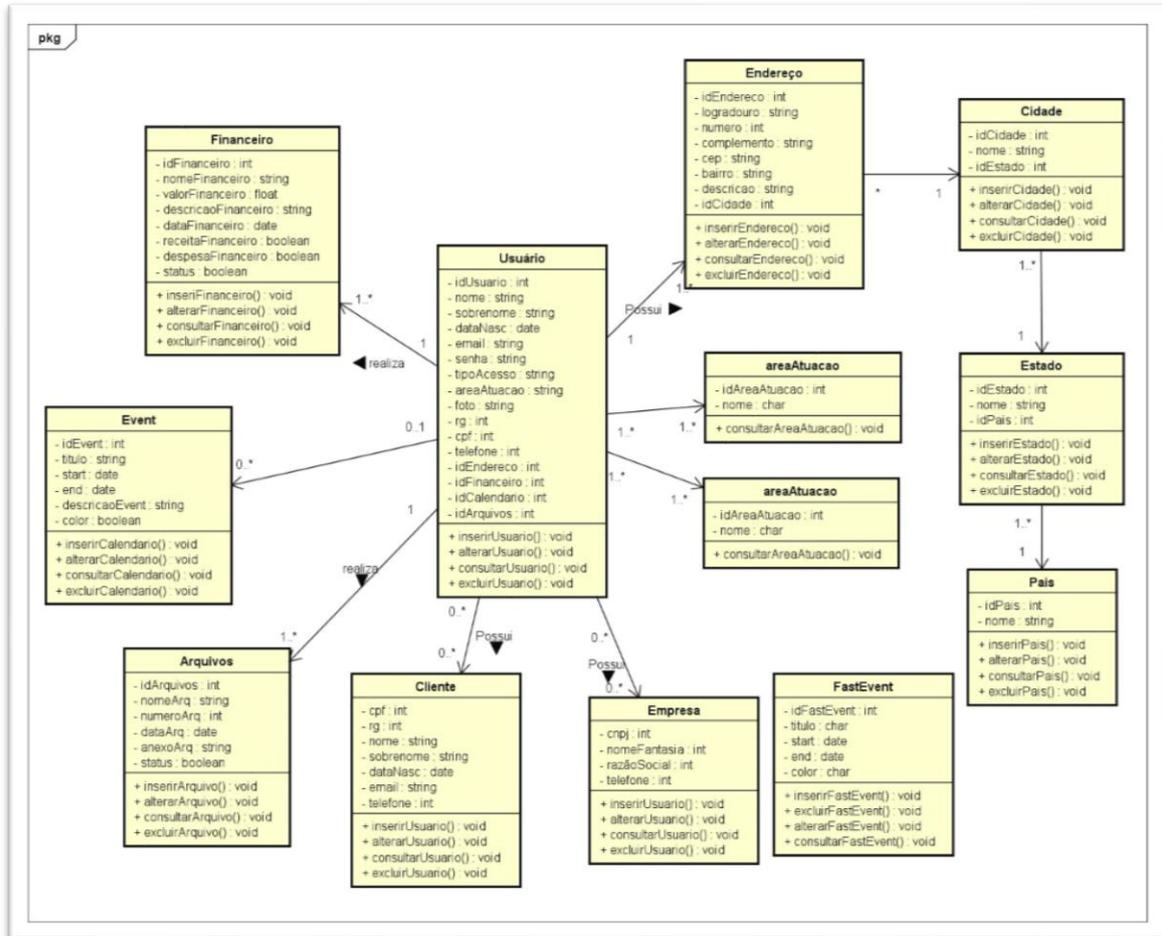
Identificação	CDU04
Caso de Uso	Gerenciar Documentos
Atores Principais	Usuário
Atores Secundários	N/A
Pré-Condições	O ator deve estar logado no sistema
Pós-Condições	N/A
Fluxo Principal	P01 – O sistema deve mostrar a interface de Documentos P02 – O ator insere nome do documento

	<p>P03 – O ator insere número do documento P05 – O ator insere data do documento P06 – O ator anexa o documento P07 – O ator clica no botão “Arquivar” P08 – Fim do Caso de Uso</p>
Fluxo Alternativo	<p>[A01] – O ator cancela o cadastro A01 – Fim do caso de Uso</p>
Fluxo de Exceção	<p>E02 – O ator clica no botão “Cadastrar” faltando completar o campo E02 – O sistema exibe os campos obrigatórios que faltam completar</p>
Regra de Negócio	<p>RN01 – O documento deve ser anexado para salvar os dados</p>
Caso de Uso ou Cenário Incluído	N/A
Observação	N/A

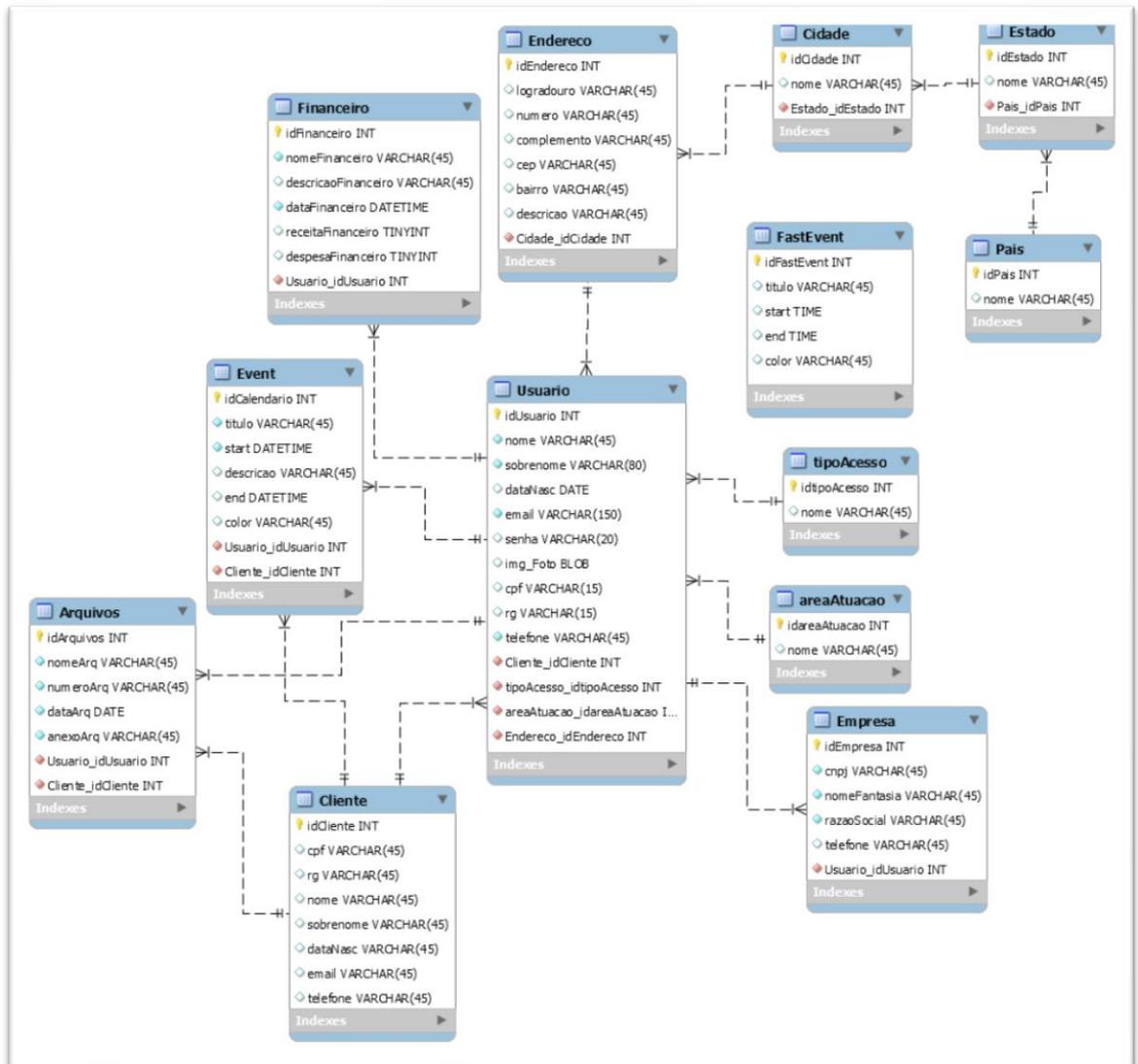
APÊNDICE G. Diagrama de Atividade de Usuário



APÊNDICE H. Diagrama de Classe



APÊNDICE I. Diagrama de Entidade Relacionamento



APÊNDICE J. Interface de tabela Financeiro

Tabela Financeiro

Id	Nome	Valor	Data	
1	Papel	200	2020-11-24	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>
2	Luz	-200	2020-10-30	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>
3	Água	-300	2020-09-30	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>
8	Recebimento de Honorário	15000	2020-12-01	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>
9	Viagem	5000	2020-12-01	<input type="button" value="Visualizar"/> <input type="button" value="Editar"/> <input type="button" value="Excluir"/>

APÊNDICE K. Tabela de Cadastro Financeiro

The image shows a registration form titled 'Financeiro'. It contains the following fields and options:

- Nome: Text input field.
- Data: Date input field with a calendar icon and the format 'dd/mm/aaaa'.
- Valor: Text input field with '00.00' pre-filled.
- Radio buttons for 'Receita' (selected) and 'Despesa'.
- Descrição: Text input field.
- A blue 'Financeiro' button at the bottom.

APÊNDICE L. Interface de Cadastro de Arquivos

The image shows the 'Cadastro de Arquivos' interface. It includes a sidebar menu with the following items: Dashboard, Tabela Financeiro, Cadastro Financeiro, Cadastro Usuário, Tabela Usuários, Cadastro Cliente, Tabela Cliente, Calendário, Cadastro Arquivos, and Tabela Arquivos. The main content area has the title 'Cadastro de Arquivos' and the following fields:

- Nome do arquivo: Text input field.
- Numero do documento: Text input field.
- Data de anexo: Date input field with a calendar icon and the format 'dd/mm/aaaa'.
- Arquivos section with an 'Escolher ficheiro' button and the text 'Nenhum ficheiro selecionado'.
- A blue 'Arquivar' button.

APÊNDICE M. Interface de Tabela Arquivos

The image shows the 'Tabela Arquivos' interface. It includes a sidebar menu with the following items: Dashboard, Tabela Financeiro, Cadastro Financeiro, Cadastro Usuário, Tabela Usuários, Cadastro Cliente, Tabela Cliente, Calendário, Cadastro Arquivos, and Tabela Arquivos. The main content area has the title 'Tabela Arquivos' and a search bar with a 'Pesquisar' button. Below the search bar is a table with the following data:

ID	Nome	Numero	Data	Visualizar	Editar	Excluir
1	Petição	2852	2020-11-09	Visualizar	Editar	Excluir
2	Recurso Inominado	2223	2020-11-09	Visualizar	Editar	Excluir
3	Protocolo de Fiançado	900	2020-11-25	Visualizar	Editar	Excluir
4	Jurisprudência	14098	2020-12-01	Visualizar	Editar	Excluir
5	Petição Inicial	605200	2020-12-02	Visualizar	Editar	Excluir

At the bottom of the table, there is a pagination control showing '1' and '2' with arrows.

APÊNDICE N. Interface de Informações Cadastradas



APÊNDICE O. Interface de Cadastro de Cliente

Cadastro Cliente

Nome Sobrenome Data de Nascimento

RG CPF

Email Telefone

CEP Endereço Nº

Bairro Complemento

Pais Estado Cidade

APÊNDICE P. Interface de Tabela Cliente

Tabela Cliente

Id	Nome	Sobrenome	Data de Nascimento			
1	Fernando	Silva	1992-11-24	Visualizar	Editar	Excluir
5	Miguel	Alves	1989-05-15	Visualizar	Editar	Excluir
7	Davi	Andrade	2000-08-17	Visualizar	Editar	Excluir
8	Arthur	Barbosa	2001-11-05	Visualizar	Editar	Excluir
9	Bernardo	Barros	2002-10-22	Visualizar	Editar	Excluir

< 1 2 >