

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA FARROUPILHA
CAMPUS SANTO ÂNGELO

AJUDI
ASSISTENTE JURÍDICO

TRABALHO DE CONCLUSÃO DE CURSO

Aline Beatriz Klug Wille

Santo Ângelo, RS, Brasil.

2021

AJUDI
ASSISTENTE JURÍDICO

por

Aline Beatriz Klug Wille

**Monografia apresentada ao Instituto Federal de Educação Ciência e Tecnologia Farroupilha,
como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.**

Orientadora: Karlise Soares Nascimento

Santo Ângelo, RS, Brasil

2021

**Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação Ciência e Tecnologia Farroupilha**

A Comissão Examinadora, abaixo assinada,
aprova a Monografia

**AJUDI
ASSISTENTE JURÍDICO**

elaborada por
Aline Beatriz Klug Wille

como requisito parcial para obtenção do título de
Tecnólogo em Sistemas Para Internet.

COMISSÃO EXAMINADORA

Karlise Soares Nascimento
(Orientadora)

Marta Breunig Loose
(Banca Examinadora)

John Soldera
(Banca Examinadora)

Conceito Final: _____

Santo Ângelo, 25 de Fevereiro de 2021.

RESUMO

AJUDI

Assistente Jurídico

Autor(a): Aline Beatriz Klug Wille

Orientadora: Karlise Soares Nascimento

Data e Local da Defesa: Santo Ângelo,dede 2021.

O avanço da internet tem levado a criação de sistemas e aplicativos que facilitam diversas atividades do cotidiano de grande parte da população, assim como, atividades do âmbito profissional onde muitos propõem-se auxiliar na organização e comunicação dentro de inúmeras empresas. O presente projeto teve como objetivo desenvolver um sistema que visa aprimorar a qualidade de gestão de escritórios de advocacia, com enfoque na otimização de tempo em relação a busca de dados e documentos pessoais de clientes, assim como, informações processuais. Além disso, sistematizar o agendamento de consultas e compromissos importantes. Dado o exposto, e averiguado que todos os objetivos definidos inicialmente foram devidamente cumpridos e, levando em consideração esses aspectos, prevê-se que a organização não só obterá um melhor desempenho, como também se mostrará adepta perante as novas tecnologias e como consequência disso espera-se que a mesma tenha destaque entre as diversas empresas do ramo jurídico.

Palavras-chave: advocacia, desenvolvimento, gestão, programação, sistema

ABSTRACT

The advancement of the internet has led to the creation of systems and applications that facilitate various daily activities for a large part of the population, as well as activities in the professional field where many propose to assist in the organization and communication within numerous companies. This project aimed to develop a system that aims to improve the quality of law firm management, with a focus on time optimization in relation to the search for data and personal documents of clients, as well as procedural information. In addition, systematize the scheduling of important appointments and appointments. Given the above, and ascertained that all the objectives initially defined were duly fulfilled and, taking these aspects into account, it is expected that the organization will not only obtain a better performance, but will also be adept in the face of new technologies and, as a consequence, expect that it stands out among the various companies in the legal field.

Key words: advocacy, development, management, programming, system

EPÍGRAFE

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota”. (Madre Teresa de Calcuta)

AGRADECIMENTOS

A minha família, que nunca deixou de me apoiar e incentivar aos estudos.

Aos amigos que compreenderam minha ausência durante o desenvolvimento deste trabalho.

A professora Karlise, por ter sido minha orientadora e estar sempre disposta a sanar minhas dúvidas e me ajudar nas dificuldades com seu conhecimento.

E por fim, aos meus professores do curso de Tecnologia em Sistemas para Internet que forneceram todas as bases necessárias para a realização deste trabalho.

LISTA DE ABREVIATURAS

AJAX - *Asynchronous JavaScript and XML*

API - *Interface de Programação de Aplicativos*

ARPA - *Advanced Research Projects Agency*

CSS - *Cascading Style Sheets*

HTML - *HyperText Markup Language*

HTTP - *Hypertext Transfer Protocol*

NPM - *Node Package Manager*

ODM - *Object Document Mapping*

SGBD - *Sistema de gerenciamento de banco de dados*

TCP/IP - *Transmission Control Protocol/Internet Protocol*

UML - *Unified Modeling Language*

XHTML - *EXtensible HyperText Markup Language*

XML - *eXtensible Markup Language*

LISTA DE TABELAS

Tabela 1: Especificação do caso de uso cadastrar cliente.....	27
Tabela 2: Especificação do caso de uso ver cadastro.....	27
Tabela 3: Especificação do caso de uso apagar registros.....	28
Tabela 4: Especificação do caso de uso editar registros.....	29
Tabela 5: Especificação do caso de uso agendar evento.....	29
Tabela 6: Especificação do caso de uso ver agendamentos.....	30
Tabela 7: Especificação do caso de uso gerar relatório de agendamentos.....	31
Tabela 8: Especificação do caso de uso cadastrar processo.....	31
Tabela 9: Especificação do caso de uso ver dados do processo.....	32
Tabela 10: Especificação do caso de uso arquivar documentos.....	33
Tabela 11: Especificação do caso de uso ver documentos.....	33
Tabela 12: Especificação do caso de uso excluir documentos.....	34
Tabela 13: Especificação do caso de uso fazer download de documentos.....	35
Tabela 14: Especificação do caso de uso fazer registrar usuário.....	35
Tabela 15: Especificação do caso de uso fazer login no sistema.....	36

LISTA DE QUADROS

Quadro 1: Schema Cliente.....	51
Quadro 2: Schema Processo.....	52
Quadro 3: Schema Evento.....	52
Quadro 4: Schema Usuario.....	53
Quadro 5: Questionário de avaliação do sistema.....	67

LISTA DE FIGURAS

Figura 1: Astrea.....	16
Figura 2: Adv Box.....	17
Figura 3: Gojur	18
Figura 4: Caso de uso.....	26
Figura 5: Diagrama de sequência do caso de uso cadastrar cliente.....	37
Figura 6: Diagrama de sequência do caso de uso ver cadastro de cliente.....	37
Figura 7: Diagrama de sequência do caso de uso apagar registros.....	38
Figura 8: Diagrama de sequência do caso de uso editar registros.....	38
Figura 9: Diagrama de sequência do caso de uso agendar evento.....	39
Figura 10: Diagrama de sequência do caso de uso ver agendamentos.....	39
Figura 11: Diagrama de sequência do caso de uso gerar relatório de agendamentos.....	40
Figura 12: Diagrama de sequência do caso de uso cadastrar processo.....	40
Figura 13: Diagrama de sequência do caso de uso ver dados do processo.....	41
Figura 14: Diagrama de sequência do caso de uso arquivar documentos.....	41
Figura 15: Diagrama de sequência do caso de uso ver documentos.....	42
Figura 16: Diagrama de sequência do caso de uso excluir documentos.....	42
Figura 17: Diagrama de sequência do caso de uso fazer download de documentos.....	43
Figura 18: Diagrama de sequência do caso de uso registrar usuário.....	43
Figura 19: Diagrama de sequência do caso de uso login.....	44
Figura 20: Diagrama de Entidade Relacionamento.....	45
Figura 21: Diagrama de modelo de dados.....	46
Figura 22: Tela inicial do sistema.....	54
Figura 23: Tela de clientes cadastrados.....	55
Figura 24: Tela de cadastro do cliente.....	56
Figura 25: Tela de cadastro do cliente.....	56

Figura 26: Tela de cadastro do cliente.....	57
Figura 27: Tela de login com o Google para acessar os documentos do Drive.....	57
Figura 28: Tela de documentos.....	58
Figura 29: Tela de criação de pastas no Drive.....	59
Figura 30: Tela de processos.....	59
Figura 31: Tela de cadastro de um novo processo.....	60
Figura 32: Tela de agenda.....	61
Figura 33: Tela de cadastro de evento.....	61
Figura 34: Tela de relatórios.....	62
Figura 35: Relatório pronto para ser impresso em PDF.....	63
Figura 36:Tela de Login.....	64
Figura 37:Tela de registro.....	64
Figura 38:Mensagem exibida ao tentar apagar um registro.....	65
Figura 39:Mensagem exibida ao tentar acessar uma página sem estar logado no sistema.....	65
Figura 40:Teste funcional do sistema.....	68
Figura 41:Teste funcional do sistema.....	69
Figura 42:Teste funcional do sistema.....	70

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS.....	15
1.1.1 Objetivo Geral.....	15
1.1.2 Objetivos Específicos.....	15
1.2 Trabalhos Correlatos.....	15
1.2.1 Astrea.....	16
1.2.2 Advbox.....	16
1.2.3 Gojur	17
1.2.4 Diferencial	18
1.3 Fundamentação Teórica	18
2 DESENVOLVIMENTO.....	25
2.1 Casos de Uso	25
2.1.1 Diagramas de Casos de Uso	25
2.1.2 Especificação de Casos de Uso	26
2.2 Diagramas de Sequência	36
2.3 Banco de Dados	44
2.3.1 Modelo Conceitual.....	44
2.3.2 Modelo de Dados.....	45
2.3.3 Models e Schemas	46
2.4 Telas do Sistema.....	54
3 RESULTADOS E DISCUSSÕES.....	66
3.1 Questionário de avaliação e testes do sistema.....	66
3.1.1 Resultados do Sistema	71
4 CONSIDERAÇÕES FINAIS.....	72
REFERÊNCIAS.....	73

1 INTRODUÇÃO

ARPANET, a primeira rede de comunicações, foi criada nos anos 60 pela ARPA (*Advanced Research Projects Agency*) para fins militares, visto que foi desenvolvida durante a guerra fria, devido a uma necessidade de comunicação entre computadores do departamento de defesa americano. Atualmente, a Internet é um imenso número de computadores conectados entre si, que através de uma pilha de protocolos TCP/IP (*Transmission Control Protocol/Internet Protocol*), são capazes de enviar e receber dados pela rede.

O avanço da Internet tem levado a criação de sistemas e aplicativos que facilitam diversas atividades do nosso cotidiano, assim como, atividades do meio profissional onde muitos visam auxiliar na organização e comunicação dentro de inúmeras empresas.

Ao observar o volume de documentos e cópias de processos aglomerados em uma sala de arquivos de um escritório de advocacia, sendo danificados e extraviados com o passar do tempo, o agendamento manual de horários de atendimentos e, o inexistente banco de dados com informações dos clientes, foi desenvolvido o sistema AJUDI - Assistente Jurídico.

Este sistema tem como propósito otimizar o tempo de busca de informações processuais, implementando uma ferramenta de arquivos na nuvem. Assim como desenvolver um agendamento de horários totalmente virtual e, implementar um banco de dados com todas as informações referentes aos clientes e seus processos, visto que, informações estas são utilizadas em documentos futuros.

Com o sistema, o usuário extingue assim o problema de perda de documentos e o desgaste da busca destas informações na pilha de papéis, visto que todos estariam na nuvem. Extinguiria também o problema de extravio de anotações de eventos importantes e a perda do tempo levado na busca de informações

peçoais de clientes para criação de novos documentos.

Pensando nisso, o presente trabalho de conclusão de curso de tecnologia em sistemas para Internet, do Instituto Federal Farroupilha - Campus Santo Ângelo, tem a finalidade de desenvolver o sistema AJUDI - Assistente Jurídico. O mesmo, deverá contar com ferramentas como, arquivamento de documentos integrados ao *google drive*, agendamento de horários e cadastramento de clientes.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um sistema para melhorar a qualidade de trabalho praticado em escritórios de advocacia. Tendo como prioridade a otimização de tempo em relação a busca de dados e documentos pessoais dos clientes, bem como informações processuais. Assim como, uma melhor organização laborativa na relação secretária-advogado(a), permitindo um agendamento de horários para atendimento de clientes e arquivamento de documentos.

1.1.2 Objetivos Específicos

- Permitir o registro e login de usuários no sistema;
- Possibilitar o cadastro, alteração, exclusão e consulta de registros dos clientes e processos;
- Proporcionar um ambiente de arquivamento de documentos com as funções de upload, download e exclusão destes arquivos;
- Possibilitar o agendamento de reuniões e consultas, assim como, gerar relatórios completos ou parciais da agenda;

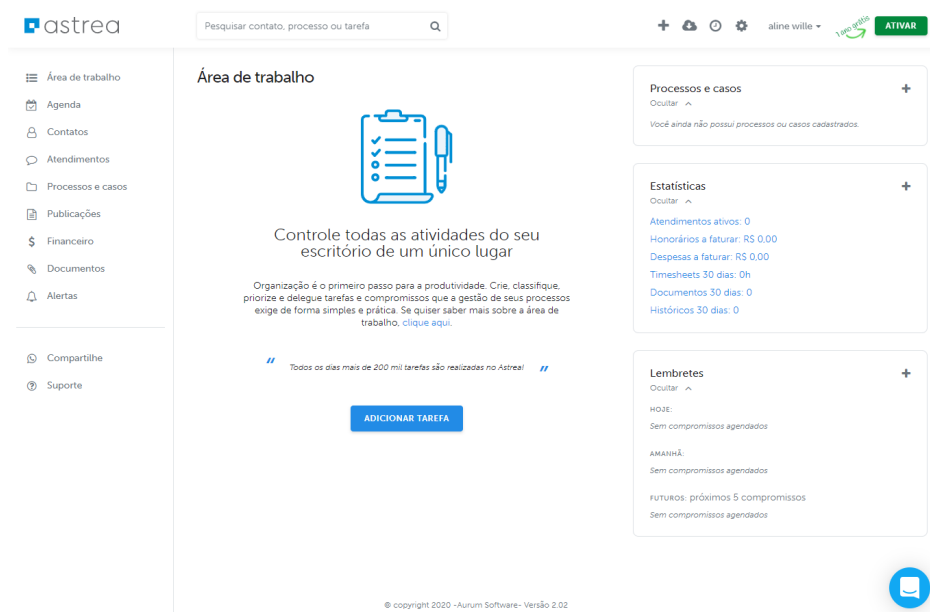
1.2 Trabalhos Correlatos

Nesta seção, são apresentados os trabalhos correlatos ao sistema AJUDI. Estes, serviram de inspiração para o projeto.

1.2.1 Astrea

A figura 1 apresenta a área de trabalho do astrea, software para escritórios de advocacia com inteligência artificial, desenvolvido pela Aurum, foi pensado para pequenos e médios escritórios. O sistema é pago e seus preços variaram entre R\$ 117,00 e mais de R\$ 2000,00, dependendo das especificações escolhidas.

Figura 1 - Astrea



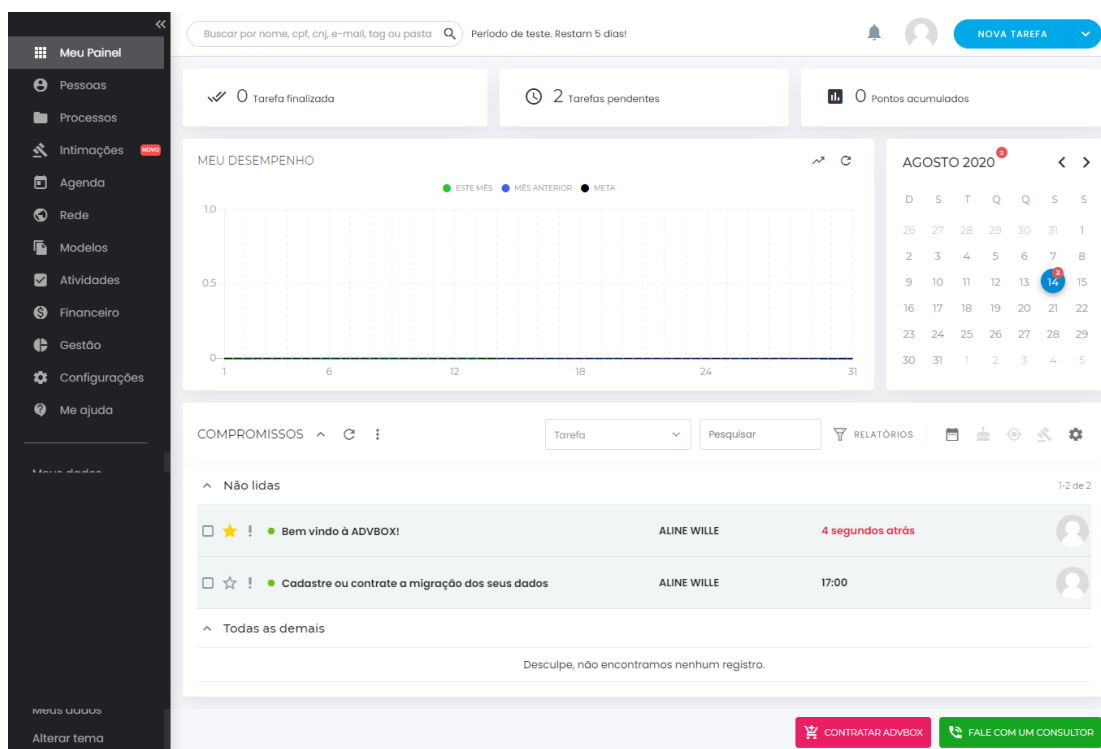
Fonte: [Astrea: Software para Advogados e Escritórios de Advocacia 🗂️ \(aurum.com.br\)](https://aurum.com.br)

Conta com funcionalidades como controle de tarefas, adicionar e excluir contatos, gestão financeira, controle de audiências, gerenciar processos, entre outros recursos de gestão de tarefas e controle de prazos.

1.2.2 Advbox

A figura 2 apresenta a área de trabalho do sistema Advbox que é um gerenciador de escritório virtual pago, com planos que vão de R\$ 219,00 até R\$ 1059,00 mensais.

Figura 2 - Advbox



Fonte: [ADVBOX - Software para escritório de advocacia](#)

Este, conta com atalhos para cadastrar novas pessoas, processos e tarefas na agenda. O programa permite a inserção de modelos de documentos, como por exemplo procurações, petições e contratos, visando otimização de tempo.

1.2.3 Gojur

A figura 3, apresenta a área de trabalho do Gojur, sistema de gerenciamento jurídico totalmente na nuvem. O software tem funcionalidades como cadastramento de processos com atualizações frequentes de movimentações.

Figura 3 - Gojur

Fonte: [GOJUR - Login](#)

O programa visa gerenciar prazos, audiências, processos, produtividade da equipe, fluxo de caixa, entre outras atividades. Conta com filtros detalhados para relatórios e amplo controle de acesso. O sistema disponibiliza uma versão grátis com funcionalidades limitadas e alguns planos que o valor pode chegar até R \$395,00 mensais.

1.2.4 Diferencial

O sistema AJUDI - Assistente Jurídico foi desenvolvido tendo como inspiração os trabalhos correlatos citados anteriormente, porém foram adicionados diferenciais que agregam valor ao sistema. Estes diferenciais são: a possibilidade de armazenamento de arquivos e, a funcionalidade de emitir relatórios dos agendamentos presentes no banco de dados, estes relatórios podem ser totais como também parciais.

1.3 Fundamentação Teórica

Nesta seção, são apresentadas as definições das ferramentas utilizadas para desenvolver o sistema.

A) Gestão empresarial

A gestão empresarial pode-se resumir em um conjunto de ações e estratégias que são aplicadas em uma empresa para um melhor resultado de seus respectivos colaboradores da organização.

Segundo Angelini:

A gestão eficaz e eficiente aumenta a probabilidade de ser mais competitivo e, portanto, sustentável no tempo. Adicionalmente, a gestão eficaz e eficiente aumenta a perspectiva de inovação e de crescimento, condições indispensáveis para sedimentar a tradição e a marca, atributos que fazem a diferença entre Escritórios. [...].

[...] Ter somente competência técnica no Direito, mediante uma boa equipe de advogados, já não é suficiente para a excelência do negócio. Hoje é imperativo que o Escritório disponha de vários outros instrumentos, entre os quais estrutura organizacional bem delineada, conhecimento das necessidades e expectativas dos clientes, fluxos de trabalho simples, padronizados, [...] (ANGELINI, 2018, p. 3).

O escritório em questão tem como foco as áreas: civil, criminal e de família. O direito civil, segundo Diniz (2010, p. 204) define-se como: "ramo do direito privado destinado a reger relações familiares, patrimoniais e obrigacionais que se formam entre indivíduos encarados como tais, ou seja, enquanto membros da sociedade." Já, o direito criminal, é definido por Diniz (2010, p. 205) da seguinte forma: "Trata-se do conjunto de normas atinentes aos crimes e às penas correspondentes, regulando a atividade repressiva do Estado para preservar a sociedade do delito." Por fim, Diniz define direito de família como:

Complexo de normas que regulam a celebração do casamento, sua validade e os efeitos que dele resultam, as relações pessoais e econômicas da sociedade conjugal, a dissolução desta, as relações entre pais e filhos, o vínculo de parentesco e os institutos complementares da tutela, curatela e ausência (DINIZ, 2010, p. 206).

B) Sistemas de agendamento

Um sistema de agendamento é um conjunto de processos desenvolvidos com

o fim de tornar o trabalho de quem o utiliza mais eficiente e produtivo. Estes sistemas são fundamentais para empresas que trabalham com atendimento de clientes, ou até mesmo para profissionais que possuem prazos para cumprir ou recorrentes reuniões.

C) Banco de dados não relacionais

Primeiramente, é de suma importância destacar o conceito do modelo relacional para podermos distingui-lo do modelo não relacional. Date (2014, p. 71) diz que: "Em um sistema relacional, o usuário vê os dados como tabelas e nada mais que tabelas.". Então, podemos dizer que, simplificadaamente, no modelo de banco de dados relacionais, os dados ficam armazenados em tabelas.

Por sua vez, um banco de dados não relacional armazena os dados sem utilizar o método tradicional com o esquema de linhas e colunas. Em um banco de dados não relacional, os dados podem ser guardados de diferentes maneiras: em documentos, colunas, chave/valor, gráficos, série temporal, objetos, índice externo, etc. Neste projeto, foi escolhido um banco de dados que trabalha com documentos o *MongoDB*.

D) Tecnologias utilizadas

- Sistema de gerenciamento de banco de dados (SGBD)

Para Heuser (2009, p. 23), um sistema de gerenciamento de banco de dados é "um software que incorpora as funções de definição, recuperação e alteração de dados em um banco de dados."

Atualmente, no mercado existem diversas opções de gerenciadores de banco de dados, podendo trazer como exemplo o *PostgreSQL*, *SQL Server*, *MySQL* e *MariaDB*. Neste projeto, será utilizado o *MongoDB*, devido sua facilidade de uso e capacidade de executar na maioria dos sistemas operacionais. O *MongoDB*, é um banco de dados não relacional, ou seja, armazena os dados em documentos *JSON* (*JavaScript Object Notation*), diferentemente dos bancos de dados relacionais, que armazenam os mesmos

em tabelas.

- *Unified Modeling Language (UML)*

A UML - linguagem de modelagem unificada, em português, segundo Guedes é:

Uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos. É uma linguagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. (GUEDES, 2009, p. 19)

Esta, por sua vez, foi criada para estabelecer uma linguagem visual, e descomplicar o desenvolvimento do software, para que seja entendido por qualquer pessoa leiga e externa ao mundo da programação.

- HTML

HTML, abreviação para *HyperText Markup Language*, que significa Linguagem de Marcação de Hipertexto, em português, como o próprio nome diz, não se refere a uma linguagem de programação, e sim, uma linguagem de marcação para páginas web.

- CSS

CSS (*Cascading Style Sheets*) são folhas de estilo utilizadas para formatar arquivos *HTML*, *XHTML*(*EXtensible HyperText Markup Language*) e *XML* (*eXtensible Markup Language*) por exemplo e, dar estilo aos mesmos, bem como seu próprio nome diz.

- *Bootstrap*

Para o desenvolvimento do front-end, foi utilizado o kit de ferramentas do *bootstrap*. O *framework*, conta com diversas funcionalidades e plugins que auxiliam a implementação de páginas amigáveis, responsivas e com animações.

- *JavaScript*

JavaScript é uma linguagem de programação desenvolvida para fornecer interatividade a uma página web.

Silva define *JavaScript* como:

JavaScript é uma linguagem desenvolvida para rodar no lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isto é possível porque existe um interpretador JavaScript hospedado no navegador. (SILVA, 2010, p. 23).

- **AJAX**

Para Silva (2009, p. 23) AJAX é a sigla em inglês para *Asynchronous JavaScript and XML* “trata-se de uma técnica de carregamento de conteúdos em uma página web com o uso de *JavaScript* e *XML*.”

- **jQuery**

O *jQuery* é uma biblioteca de funções *JavaScript* que foi desenvolvida para interagir com tags *HTML* e simplificar scripts interpretados pelo navegador.

- **Google Drive API**

API, ou interface de programação de aplicativos, é um conjunto de padrões de programação que permite a construção de softwares que se conectam a outros sistemas para a execução de determinadas funções.

Presentemente, existem as mais variadas opções de *API*'s no mercado, podendo trazer como exemplo as interfaces de pagamentos online como da *PayPal*, Cielo, PagSeguro, entre outras. Pode-se exemplificar uma *API* também, com as ferramentas de localização disponibilizadas pelo *Google Maps* e *Foursquare*.

O *google drive* é um serviço que armazena arquivos na nuvem. Com sua *API*, é possível criar sistemas *web* e aplicativos móveis que sincronizam os dados na plataforma podendo ler os mesmos, e até mesmo inserir novos

arquivos.

- *Node.js*

Tendo em vista sua leveza, praticidade e multiplataforma, no back-end do sistema foi utilizada a linguagem de programação *node.js*.

Ele pode ser definido como uma plataforma *runtime* para a linguagem de programação *JavaScript*. Resumidamente, o *Node.js* surgiu para que não fosse mais necessário depender de um navegador para a execução do *JavaScript*.

- Módulos

Primeiramente, *NPM* é um gerenciador de pacotes do node (*Node Package Manager*), este é utilizado para compartilhar ferramentas, instalar módulos e gerenciar suas dependências. Os módulos utilizados neste projeto foram: *Mongoose*, *Handlebars*, *Body Parser*, *Flash*, *bcryptjs*, *passport*, *passport-local* e *Nodemon*.

- O *Mongoose* é um módulo utilizado para conectar a aplicação ao MongoDB para assim, modelar os dados da aplicação diretamente do Node.js;
- Já o *Handlebars*, foi utilizado para criar templates básicos ao sistema como *navbar*, *footer*, e mensagens de erro e sucesso;
- O *Nodemon* é utilizado em tempo de desenvolvimento, pois o mesmo monitora alterações no código e automaticamente reinicia o servidor;
- O *Body Parser* é um módulo utilizado para converter o *body* de uma requisição para vários formatos;
- O *bcryptjs*, por sua vez, foi utilizado para criptografar as senhas dos usuários, por exemplo, um usuário se cadastrou com a senha “senha1234”, no banco de dados essa senha é salva na forma de uma série de caracteres, como por exemplo “\$2a\$10\$6bAGaUi8LPWAKAIJPabTV.hbKklh1ivUUrt2AWsbpCqhqrTb

N8q6”;

- Os módulos `passport` e `passport-local` foram utilizados para se fazer possível a autenticação dos usuários no sistema;
- Por fim, o *Flash* é um módulo para armazenar mensagens, no formato flash, ou seja, quando a página é atualizada as mesmas desaparecem.
- Express

O *express* é um *framework* para `node.js`, que facilita o desenvolvimento das aplicações, pois conta com um sistema de rotas completo e gerencia requisições *HTTP (Hypertext Transfer Protocol)*.

2 DESENVOLVIMENTO

2.1 Casos de Uso

De acordo com Guedes:

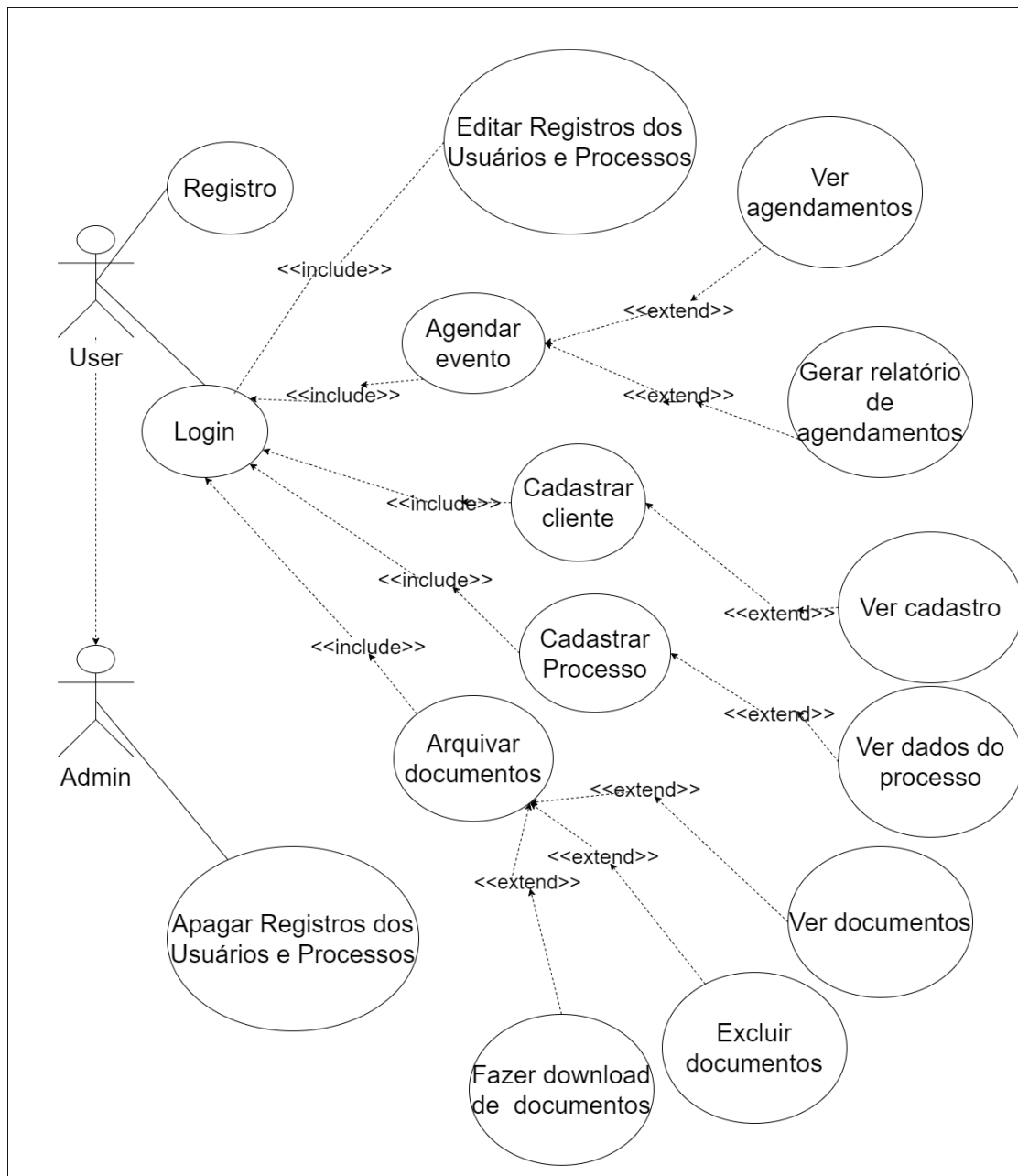
Os casos de uso são utilizados para capturar os requisitos do sistema, ou seja, referem-se aos serviços, tarefas ou funcionalidades identificados como necessários ao software e que podem ser utilizados de alguma maneira pelos atores que interagem com o sistema, sendo usados para expressar e documentar os comportamentos pretendidos para as funções deste. (GUEDES, 2009, p. 53).

Simplificadamente, os casos de usos são utilizados para auxiliar os usuários a entender as funcionalidades do sistema, assim como faz a figura 4.

2.1.1 Diagramas de Casos de Uso

Neste diagrama, são representadas as funcionalidades de cadastro de novos clientes e processos e, manter seus respectivos cadastros atualizados, bem como agendar consultas e se necessário alterar os dados da agenda. Assim como inserir arquivos pessoais e processuais de seus clientes no sistema e, também, fazer login ou se cadastrar no programa.

Figura 4 - Caso de uso



Fonte: Própria (2020)

2.1.2 Especificação de Casos de Uso

Tabela 1: Especificação do caso de uso cadastrar cliente

Identificação de caso de uso: CC01	Nome de caso de uso: Cadastrar cliente
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para cadastrar um novo cliente no sistema.	
Pré-condições: Estar logado no sistema.	Pós-condições: Mensagem de cliente cadastrado com sucesso e redirecionamento para página de clientes.
Fluxo principal	
Ações do ator: 1. Clicar em “Clientes”; 2. Clicar em “Cadastrar Clientes”; 3. Inserir dados; 4. Clicar em “Salvar”.	Ações do sistema: 5. Sistema salva informações no banco de dados; 6. Sistema retorna mensagem de sucesso.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 5. Dados inválidos 6. O sistema retorna ao passo 2.

Fonte: Própria (2020)

Tabela 2: Especificação do caso de uso ver cadastro.

Identificação de caso de uso: VC01	Nome de caso de uso: Ver cadastro de cliente
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para visualizar os dados dos clientes cadastrados no sistema.	
Pré-condições: Estar logado no sistema.	Pós-condições: Redirecionamento para página de clientes.
Fluxo principal	

Ações do ator: 1. Clicar em “Clientes”;	Ações do sistema: 2. Sistema exibe a página de clientes.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 2. Erro ao carregar. 3. Retorna a página de clientes como uma nova tentativa de carregar os dados.

Fonte: Própria (2020)

Tabela 3: Especificação do caso de uso apagar registros

Identificação de caso de uso: AR01	Nome de caso de uso: Apagar registros
Atores principais: Admin	
Resumo: Este caso de uso descreve as sequências de passos que o <i>admin</i> deverá seguir para apagar um registro de cliente, evento ou processo do banco de dados do sistema.	
Pré-condições: Ser <i>admin</i> ; Estar logado no sistema; Ter um cliente, evento ou processo cadastrado no banco de dados do sistema.	Pós-condições: Mensagem de apagado com sucesso e redirecionamento para página principal.
Fluxo principal	
Ações do ator: 1. Clicar na página do registro que deseja apagar; 2. Clicar em “Apagar”;	Ações do sistema: 3. Sistema apaga informações do banco de dados; 4. Sistema retorna mensagem de sucesso.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 3. Erro; 4. O sistema retorna a página principal e exibe mensagem de erro.

Fonte: Própria (2020)

Tabela 4: Especificação do caso de uso editar registros

Identificação de caso de uso: AC01	Nome de caso de uso: Editar registros
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para atualizar os dados de um cliente, evento ou processo no sistema.	
Pré-condições: Estar logado no sistema; Existir um cliente, evento ou processo cadastrado no banco de dados do sistema.	Pós-condições: Mensagem de atualizado com sucesso e redirecionamento para página de clientes.
Fluxo principal	
Ações do ator: 1. Clicar na página do registro que deseja apagar; 2. Clicar em “Editar”; 3. Inserir novos dados; 4. Clicar em “Salvar”.	Ações do sistema: 5. Sistema atualiza informações no banco de dados. 6. Sistema exibe mensagem de sucesso.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 5. Dados inválidos 6. O sistema retorna ao passo 3.

Fonte: Própria (2020)

Tabela 5: Especificação do caso de uso agendar evento

Identificação de caso de uso: AE01	Nome de caso de uso: Agendar Evento
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para agendar eventos no sistema.	
Pré-condições: Estar logado no sistema.	Pós-condições: Mensagem de evento cadastrado com sucesso e redirecionamento para página de agenda.
Fluxo principal	

Ações do ator: 1. Clicar em “Agenda”; 2. Clicar em algum dia do calendário; 3. Inserir dados do evento; 4. Clicar em “Salvar”.	Ações do sistema: 5. Sistema salva informações no banco de dados. 6. Sistema exibe mensagem de sucesso.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 5. Erro 6. Sistema retorna a página da agenda e exibe mensagem de erro.

Fonte: Própria (2020)

Tabela 6: Especificação do caso de uso ver agendamentos

Identificação de caso de uso: VA01	Nome de caso de uso: Ver Agendamentos
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para ver os eventos cadastrados no sistema.	
Pré-condições: Estar logado no sistema.	Pós-condições: Redirecionamento para página de agenda.
Fluxo principal	
Ações do ator: 1. Clicar em “Agenda”;	Ações do sistema: 2. Sistema exibe os eventos cadastrados no banco de dados.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 2. Erro 3. Sistema retorna a página da agenda com uma nova tentativa de carregar os eventos.

Fonte: Própria (2020)

Tabela 7: Especificação do caso de uso gerar relatório de agendamentos

Identificação de caso de uso: GR01	Nome de caso de uso: Gerar relatório de agendamentos
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para gerar um relatório dos agendamentos.	
Pré-condições: Estar logado no sistema.	Pós-condições: Página de impressão do PDF com o relatório de agendamentos.
Fluxo principal	
Ações do ator: 1. Clicar em “Agenda”; 2. Clicar em “Gerar relatório de agendamentos”	Ações do sistema: 3. Sistema gera o relatório e abre a página de impressão do arquivo PDF em uma nova guia do navegador.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 3. Erro 4. Sistema permanece na página de agenda e exibe mensagem de erro.

Fonte: Própria (2020)

Tabela 8: Especificação do caso de uso cadastrar processo

Identificação de caso de uso: CP01	Nome de caso de uso: Cadastrar processo
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para cadastrar um processo no sistema.	
Pré-condições: Estar logado no sistema; O cliente referente ao processo deve estar cadastrado no banco de dados do sistema.	Pós-condições: Mensagem de processo cadastrado com sucesso e redirecionamento para página de processos.
Fluxo principal	

Ações do ator: 1. Clicar em “Processos”; 2. Clicar em “Cadastrar processo”; 3. Inserir dados; 4. Clicar em “Salvar”.	Ações do sistema: 5. Sistema salva informações no banco de dados; 6. Sistema exibe mensagem de sucesso.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 5. Erro 6. Sistema retorna ao item 3 e exibe mensagem de erro.

Fonte: Própria (2020)

Tabela 9: Especificação do caso de uso ver dados do processo

Identificação de caso de uso: VD01	Nome de caso de uso: Ver dados do processo
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para ver os dados dos processos cadastrados no sistema.	
Pré-condições: Estar logado no sistema.	Pós-condições: Redirecionamento para página de processos.
Fluxo principal	
Ações do ator: 1. Clicar em “Processos”;	Ações do sistema: 2. Sistema exibe os processos e seus respectivos dados.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 2. Erro 3. Redirecionamento a página de processos como tentativa de recarregar os dados.

Fonte: Própria (2020)

Tabela 10: Especificação do caso de uso arquivar documentos

Identificação de caso de uso: AD01	Nome de caso de uso: Arquivar documentos
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para arquivar documentos na nuvem.	
Pré-condições: Estar logado no sistema; Ter uma conta no Google Drive e estar logada na mesma.	Pós-condições: Redirecionamento para página de arquivos.
Fluxo principal	
Ações do ator: 1. Clicar em “Documentos”; 2. Clicar em “Upload”; 3. Escolher arquivo; 4. Clicar em “Enviar”	Ações do sistema: 5. Sistema salva arquivo.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 5. Erro ao gravar; 6. Sistema retorna ao passo 2.

Fonte: Própria (2020)

Tabela 11: Especificação do caso de uso ver documentos

Identificação de caso de uso: VD01	Nome de caso de uso: Ver documentos
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para ver os documentos arquivados.	
Pré-condições: Estar logado no sistema; Ter uma conta no Google Drive e estar logada na mesma.	Pós-condições: Sistema exibe documentos
Fluxo principal	

Ações do ator: 1. Clicar em “Documentos”;	Ações do sistema: 2. Redirecionamento para página de documentos.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 2. Erro ao carregar; 3. Sistema retorna ao passo 2.

Fonte: Própria (2020)

Tabela 12: Especificação do caso de uso excluir documentos

Identificação de caso de uso: ED01	Nome de caso de uso: Excluir documentos
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para excluir um documento.	
Pré-condições: Estar logado no sistema/ Ter uma conta no Google Drive e estar logada na mesma; Existir algum documento salvo.	Pós-condições: Mensagem de documento excluído com sucesso e redirecionamento para a página de documentos.
Fluxo principal	
Ações do ator: 1. Clicar em “Documentos”; 2. Clicar no “x” do documento desejado;	Ações do sistema: 3. Sistema exclui arquivo.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 3. Erro ao excluir; 4. Sistema retorna ao passo 2.

Fonte: Própria (2020)

Tabela 13: Especificação do caso de uso fazer download de documentos

Identificação de caso de uso: FD01	Nome de caso de uso: Fazer download de documentos
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para fazer o download de um documento.	
Pré-condições: Estar logado no sistema; Ter uma conta no Google Drive e estar logada na mesma; Existir algum documento salvo.	Pós-condições: Download do arquivo.
Fluxo principal	
Ações do ator: 1. Clicar em “Documentos”; 2. Clicar no “↓” do documento desejado;	Ações do sistema: 3. O sistema inicia o download do arquivo em uma nova guia do navegador.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 3. Erro ; 4. O sistema permanece na página de documentos para nova tentativa de download.

Fonte: Própria (2020)

Tabela 14: Especificação do caso de uso fazer registrar usuário

Identificação de caso de uso: RU01	Nome de caso de uso: Registrar Usuário
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para se registrar no sistema.	
Pré-condições:	Pós-condições: Mensagem de conta criada com sucesso e redirecionamento para página principal.
Fluxo principal	

Ações do ator: 1. Clicar em “Registro”; 2. Inserir dados; 3. Clicar em “Criar Conta”	Ações do sistema: 4. O sistema cria a conta e redireciona a página principal.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 4. Erro ; 5. Retorna ao passo 2.

Fonte: Própria (2020)

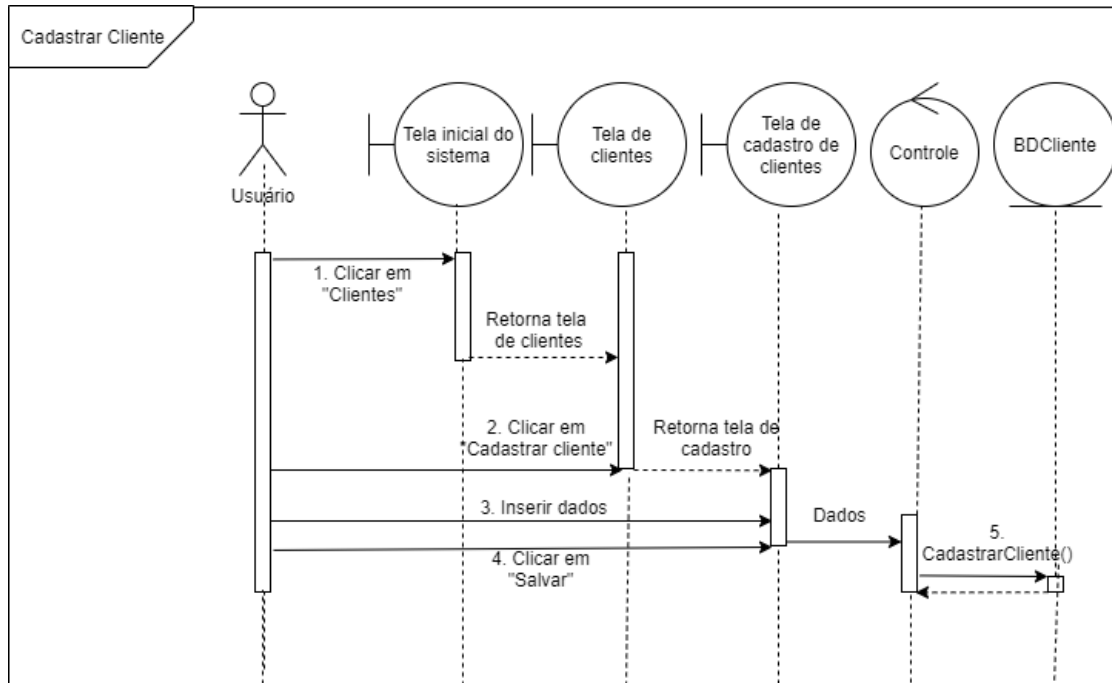
Tabela 15: Especificação do caso de uso fazer login no sistema

Identificação de caso de uso: L01	Nome de caso de uso: Login
Atores principais: User e Admin	
Resumo: Este caso de uso descreve as sequências de passos que o usuário deverá seguir para fazer login no sistema.	
Pré-condições: Ter uma conta cadastrada.	Pós-condições: Redirecionamento para página principal.
Fluxo principal	
Ações do ator: 4. Clicar em “Login”; 5. Inserir dados; 6. Clicar em “Entrar”	Ações do sistema: 5. Redirecionamento a página principal.
Fluxo alternativo	
Ações do ator:	Ações do sistema: 4. Erro ; 5. Retorna ao passo 2.

Fonte: Própria (2020)

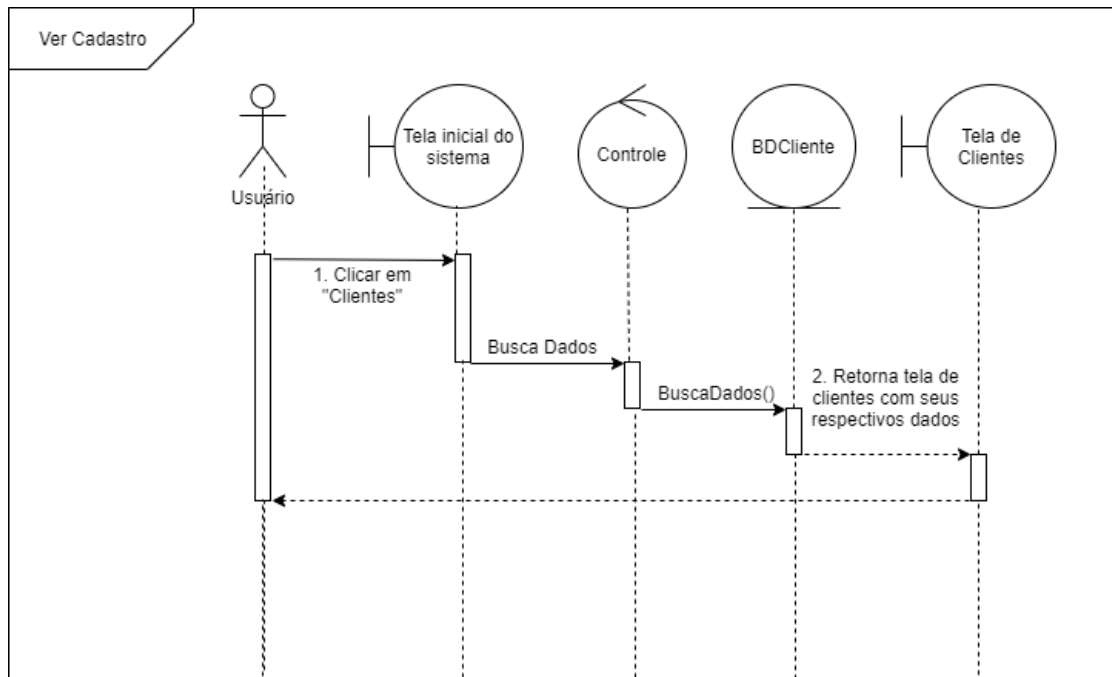
2.2 Diagramas de Sequência

Figura 5: Diagrama de sequência do caso de uso cadastrar cliente

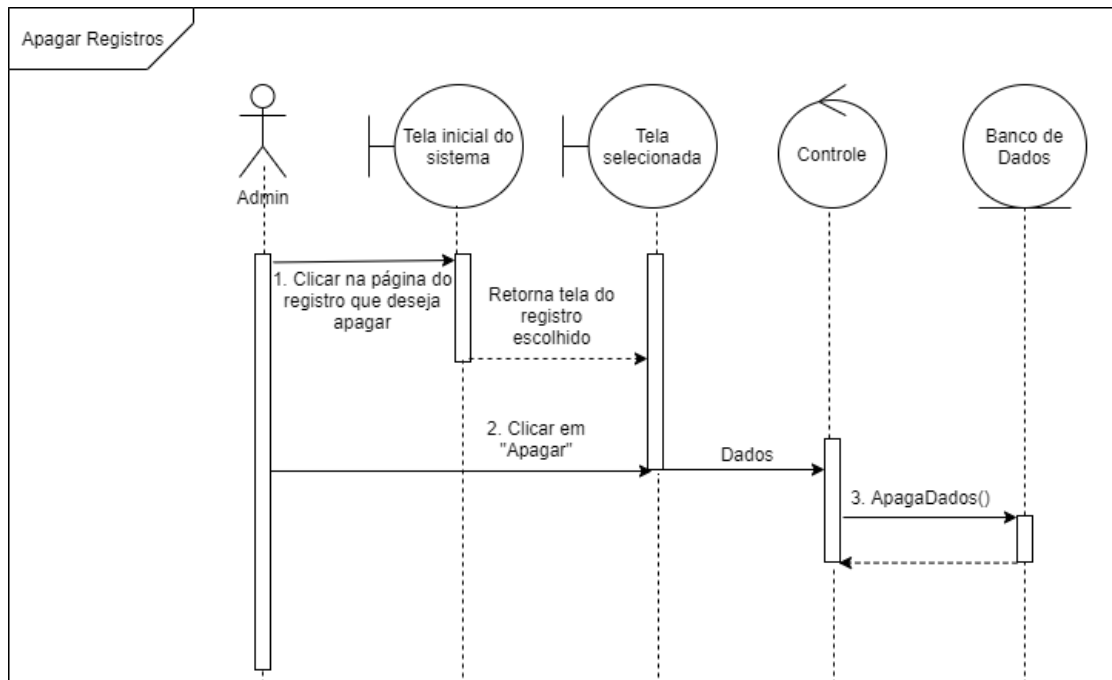


Fonte: Própria (2020)

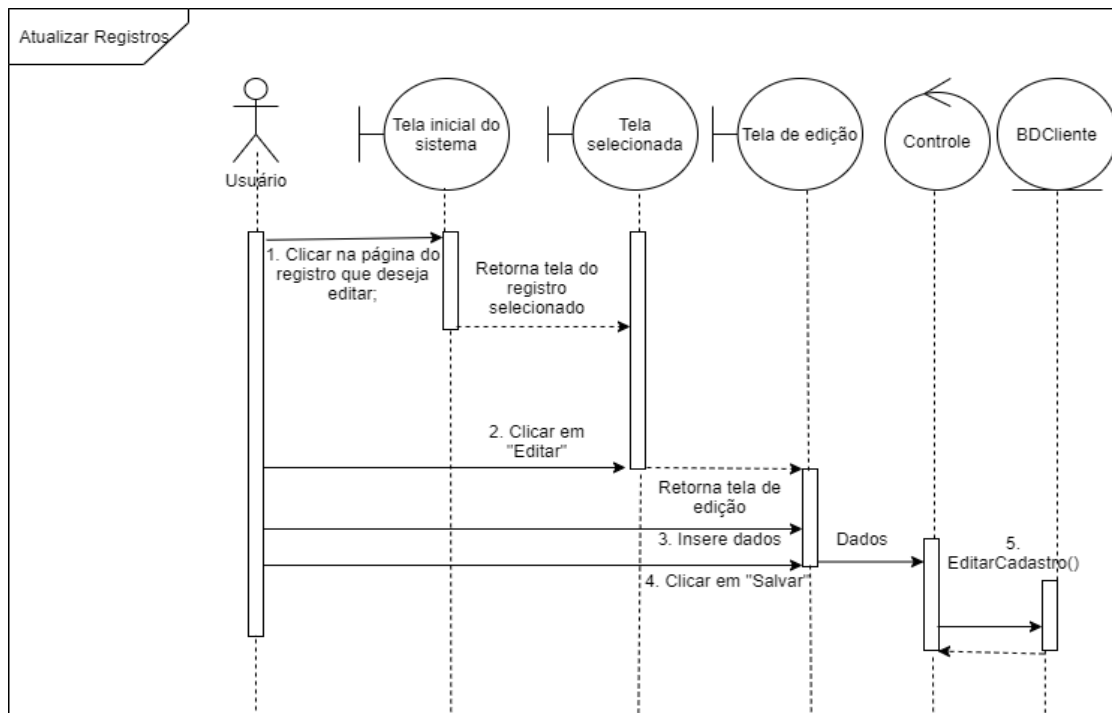
Figura 6: Diagrama de sequência do caso de uso ver cadastro de cliente



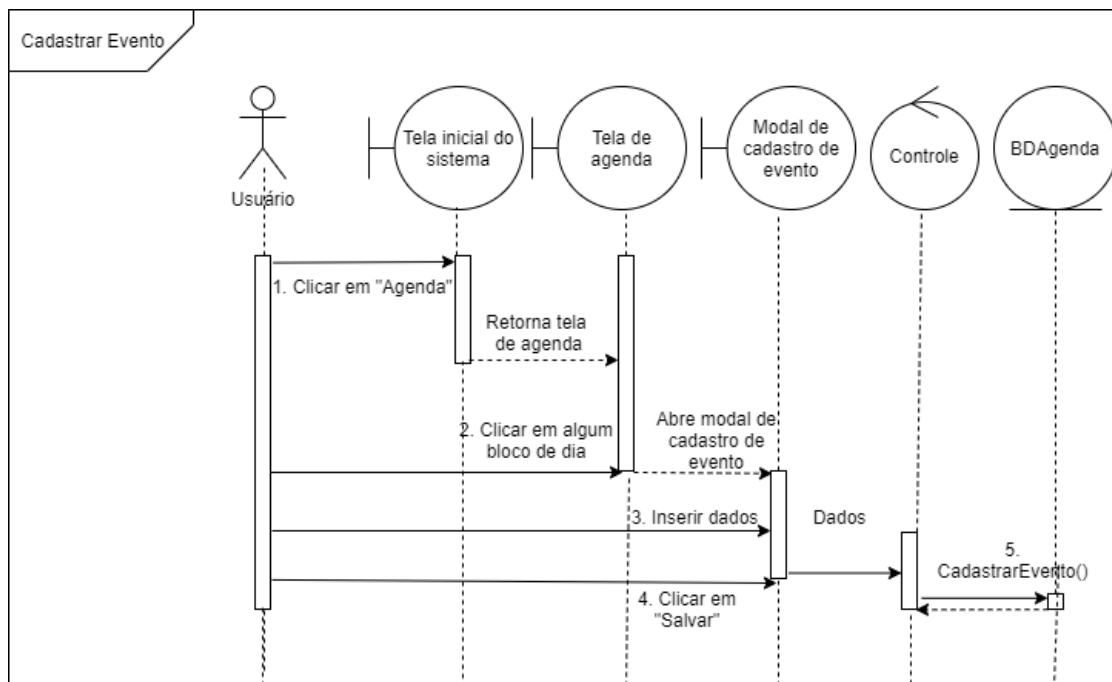
Fonte: Própria (2020)

Figura 7: Diagrama de sequência do caso de uso apagar registros

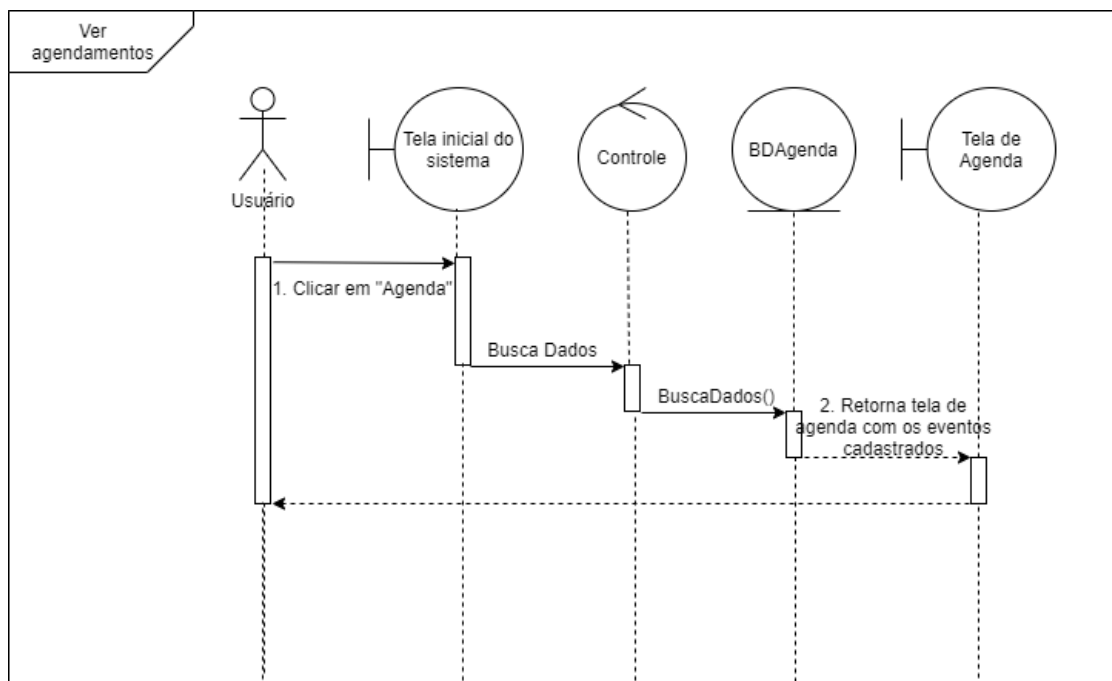
Fonte: Própria (2020)

Figura 8: Diagrama de sequência do caso de uso editar registros

Fonte: Própria (2020)

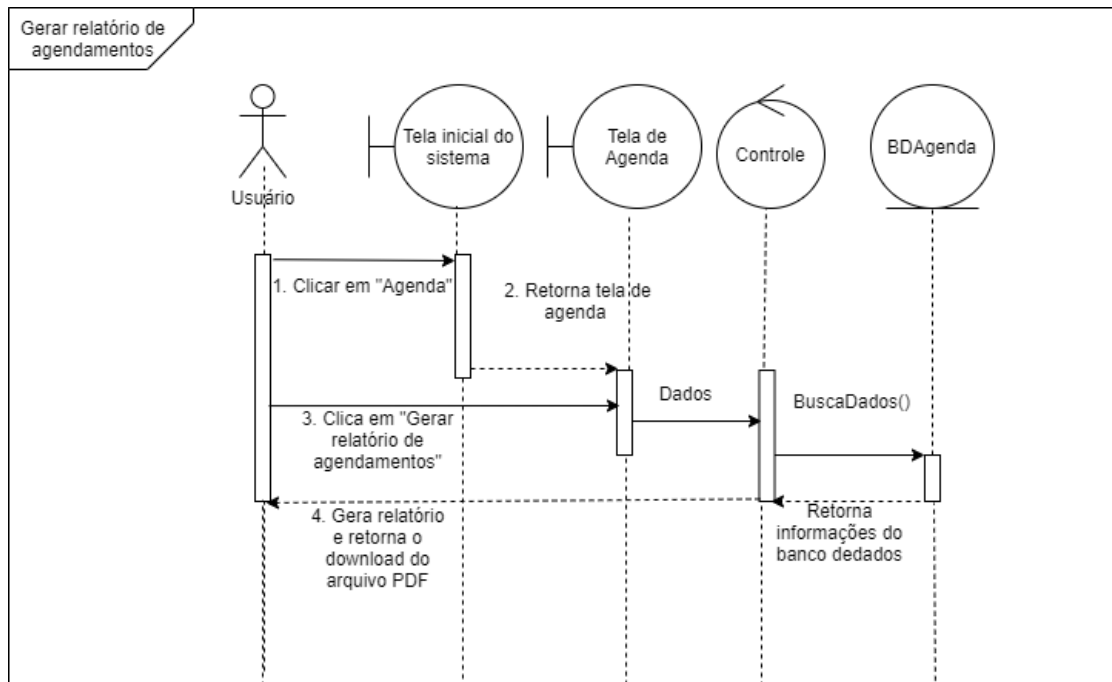
Figura 9: Diagrama de sequência do caso de uso agendar evento

Fonte: Própria (2020)

Figura 10: Diagrama de sequência do caso de uso ver agendamentos

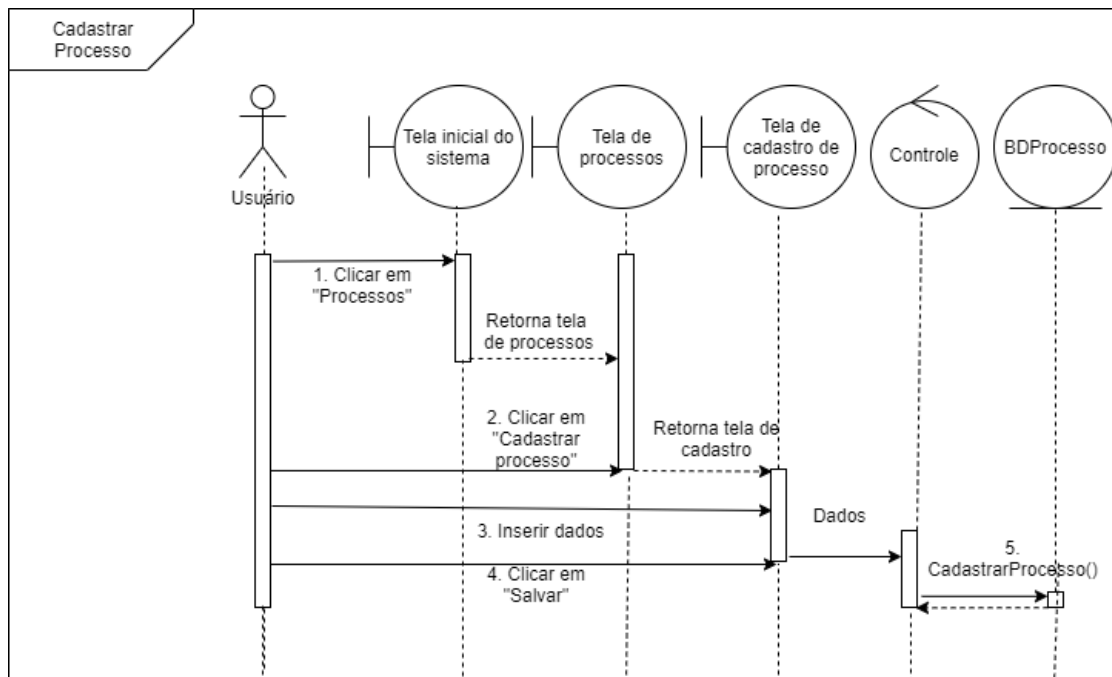
Fonte: Própria (2020)

Figura 11: Diagrama de sequência do caso de uso gerar relatório de agendamentos

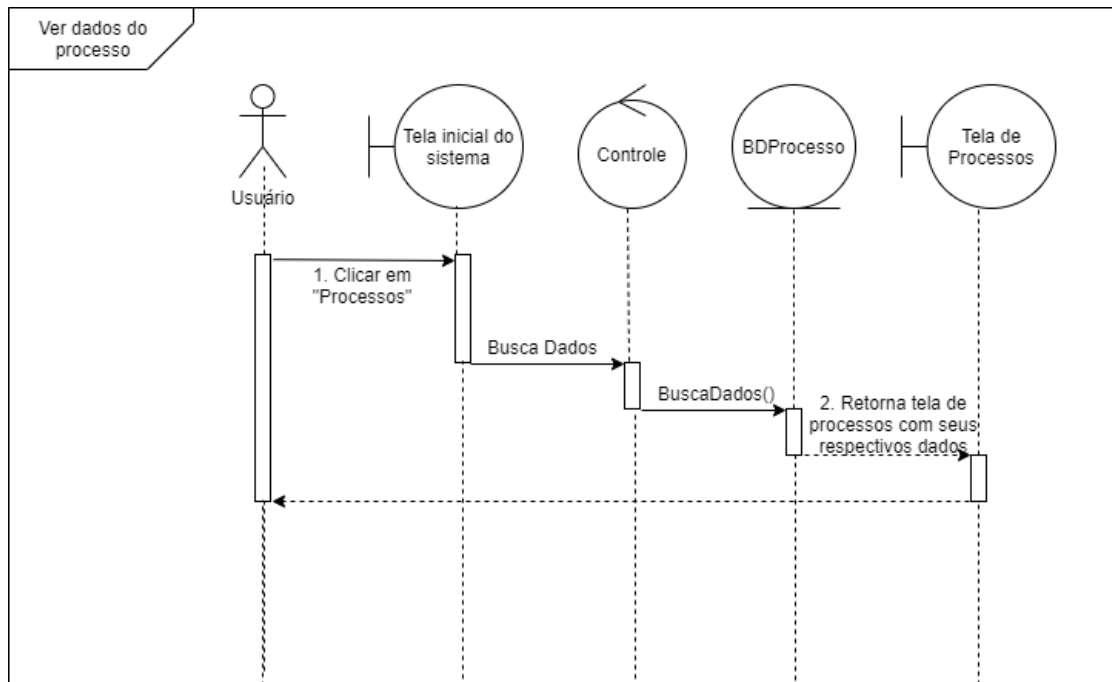


Fonte: Própria (2020)

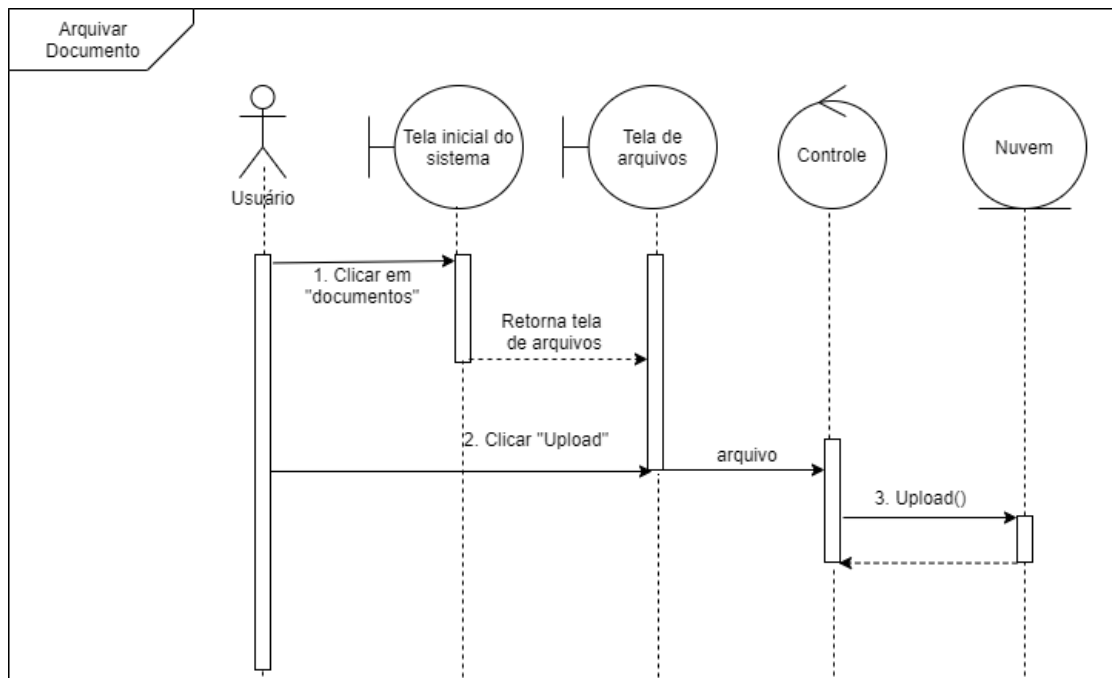
Figura 12: Diagrama de sequência do caso de uso cadastrar processo



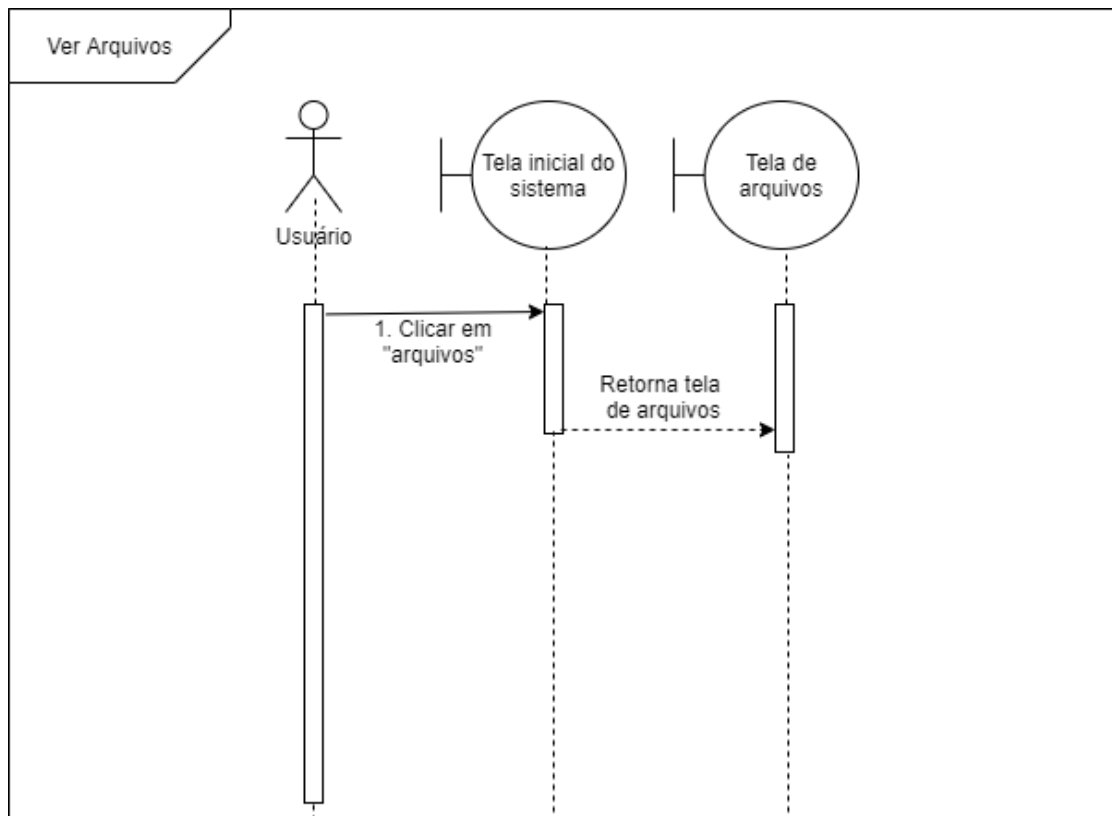
Fonte: Própria (2020)

Figura 13: Diagrama de sequência do caso de uso ver dados do processo

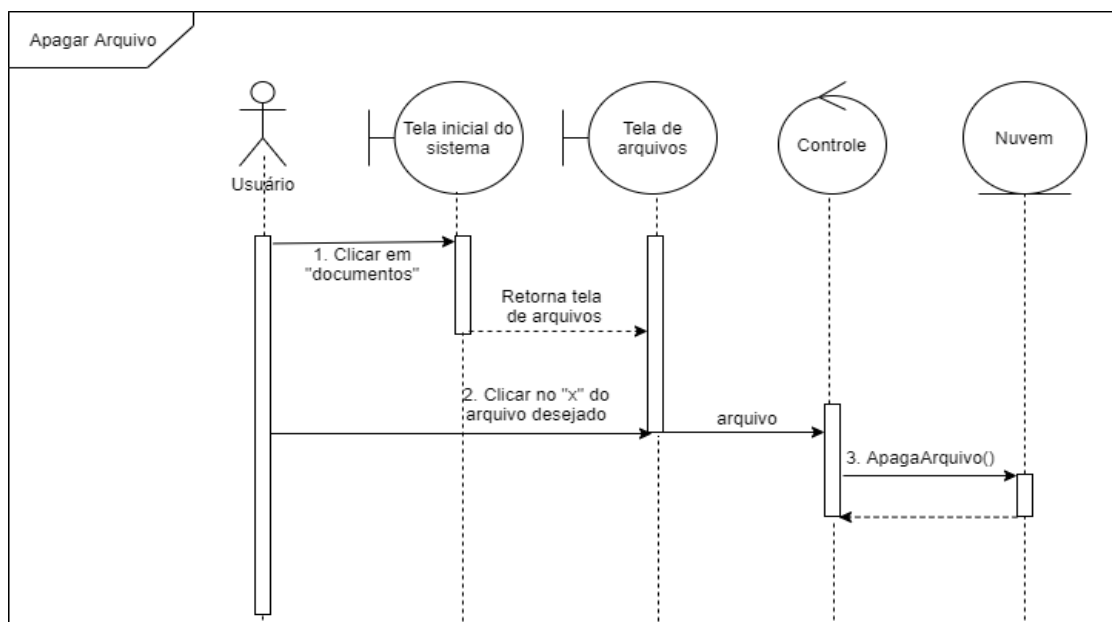
Fonte: Própria (2020)

Figura 14: Diagrama de sequência do caso de uso arquivar documentos

Fonte: Própria (2020)

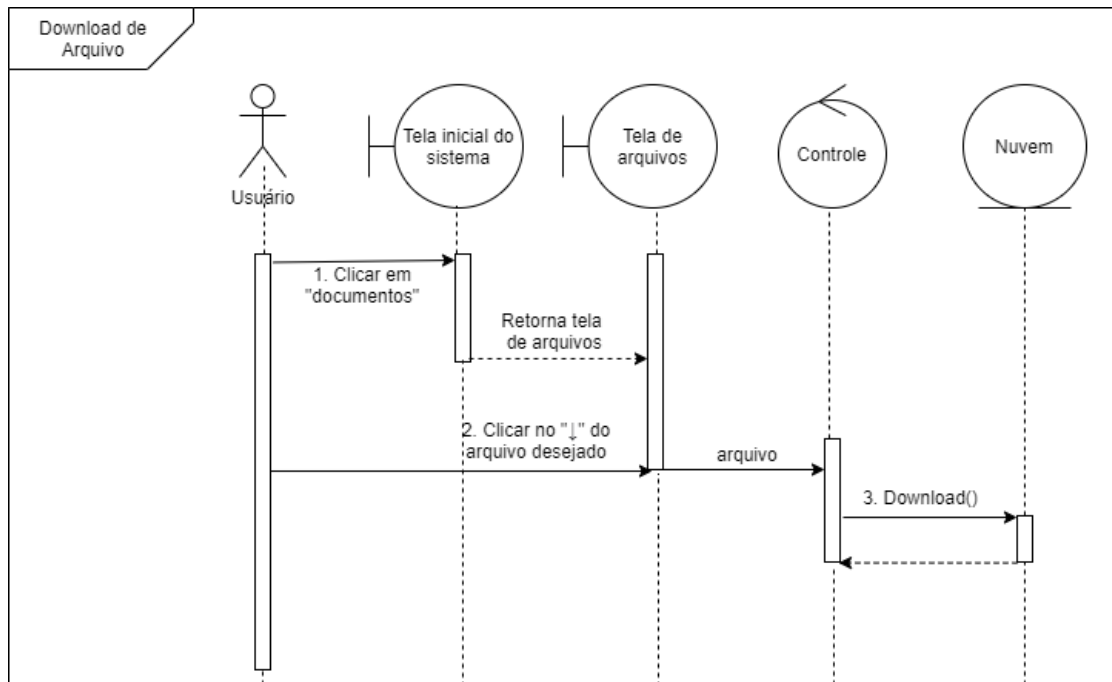
Figura 15: Diagrama de sequência do caso de uso ver documentos

Fonte: Própria (2020)

Figura 16: Diagrama de sequência do caso de uso excluir documentos

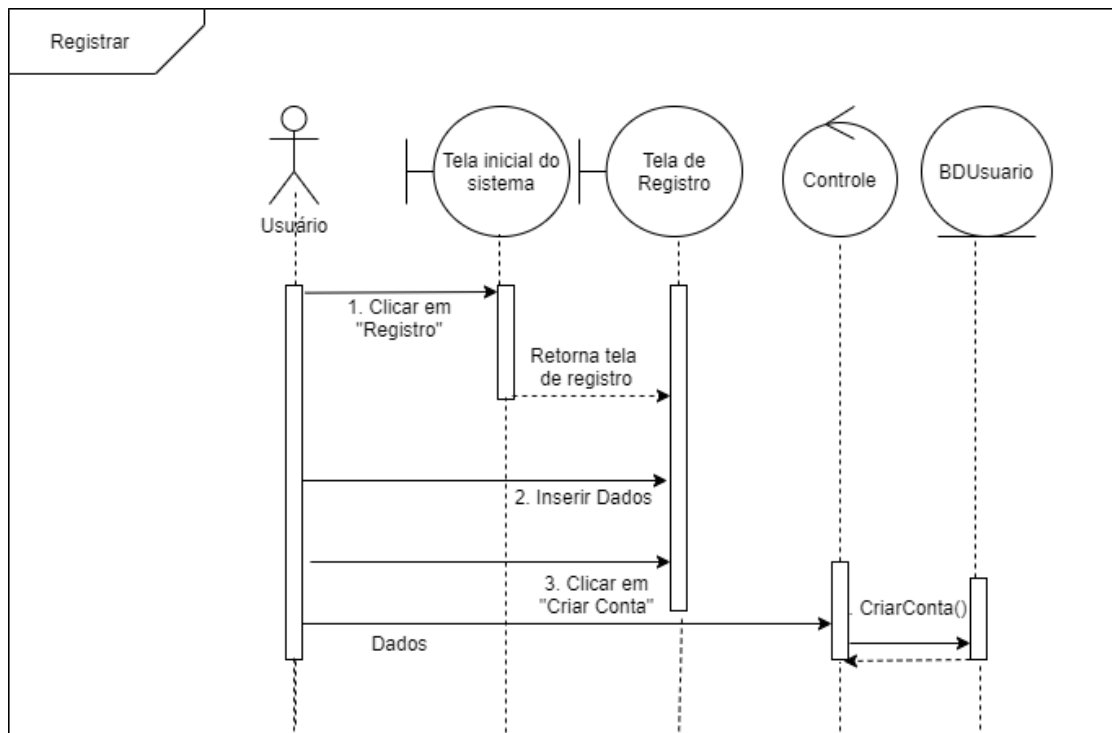
Fonte: Própria (2020)

Figura 17: Diagrama de sequência do caso de uso fazer download de documentos

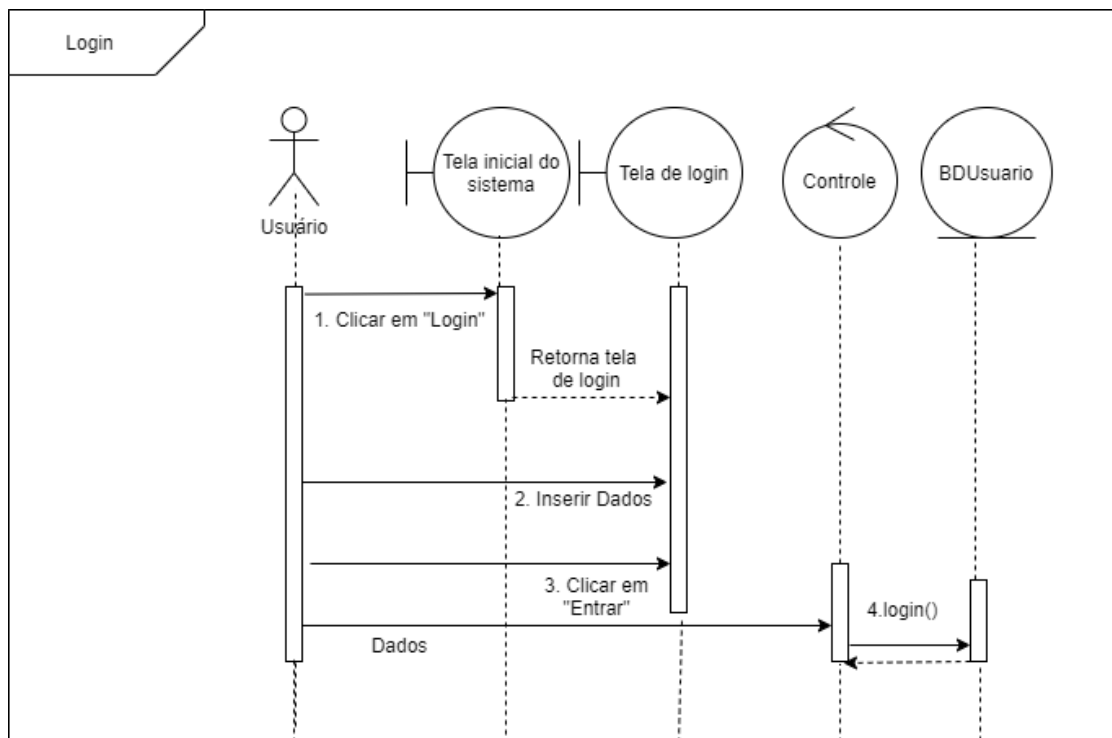


Fonte: Própria (2020)

Figura 18: Diagrama de sequência do caso de uso registrar usuário



Fonte: Própria (2020)

Figura 19: Diagrama de sequência do caso de uso login

Fonte: Própria (2020)

2.3 Banco de Dados

Date (2004), simplifica a definição de banco de dados da seguinte maneira:

O banco de dados, por si, pode ser considerado como o equivalente eletrônico de um armário de arquivamento; ou seja, ele é um repositório ou recipiente para uma coleção de arquivos de dados computadorizados. (DATE, 2004, p. 3)

Neste projeto o banco de dados escolhido para persistir os dados foi o MongoDB, este é um banco de dados não relacional, que, diferentemente dos bancos de dados relacionais, não guarda as informações em tabelas.

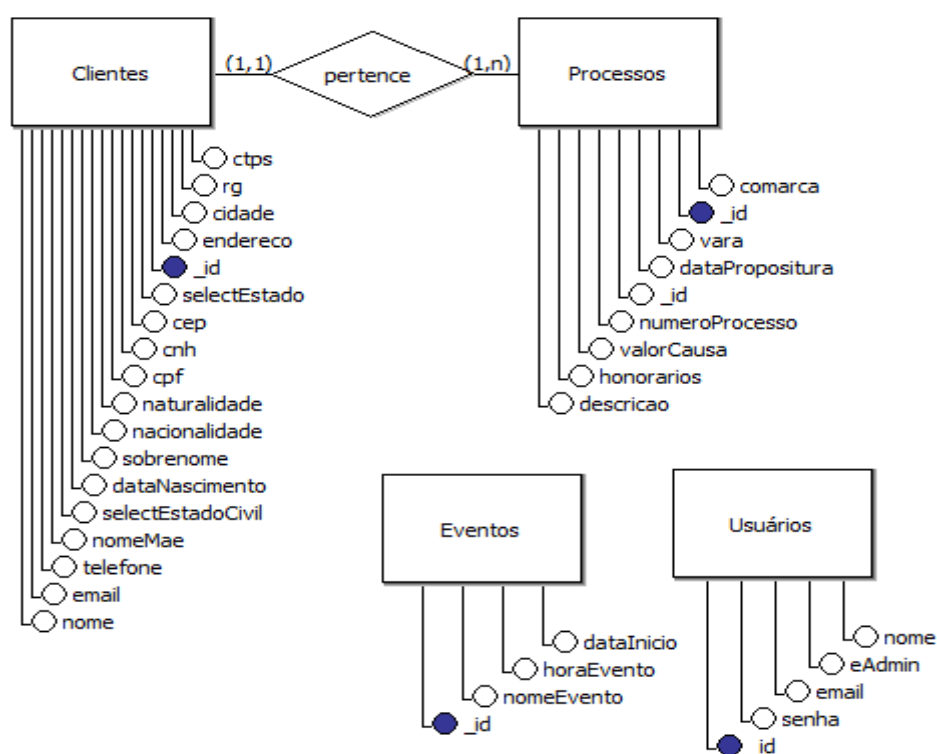
2.3.1 Modelo Conceitual

A figura 20, apresentada logo abaixo, refere-se ao diagrama entidade relacionamento do banco de dados. Este diagrama foi usado para, de uma melhor forma, representar como os dados são armazenados na base de dados.

Na imagem é possível fazer algumas observações. O atributo `_id` possui esta denominação pois é gerado automaticamente pelo sistema, este por sua vez tem uma tipagem diferente dos demais: se trata de um objeto no formato *JSON*. O Atributo `_id` guarda informações como: `_id":{"$oid":"6035c50c5043d82d38df4c74"}`. Este id é criado automaticamente, porém é de grande importância para o desenvolvedor, é a partir dele que é possível conectar as coleções como foi feito entre o documento Clientes e Processos, por este motivo Processos possui dois atributos `_id`, um é o seu id e outro é o id de um cliente. Na figura 21 pode-se ter um exemplo real de como está organizado o banco de dados.

Pode-se observar também que, no documento Usuários, existe um atributo `eAdmin`, este por sua vez guarda apenas 0 ou 1. Se o valor armazenado for 0, o usuário não é admin e, caso o valor for 1, o mesmo é admin e, tem mais funcionalidades à sua disposição no sistema.

Figura 20: Diagrama Entidade Relacionamento

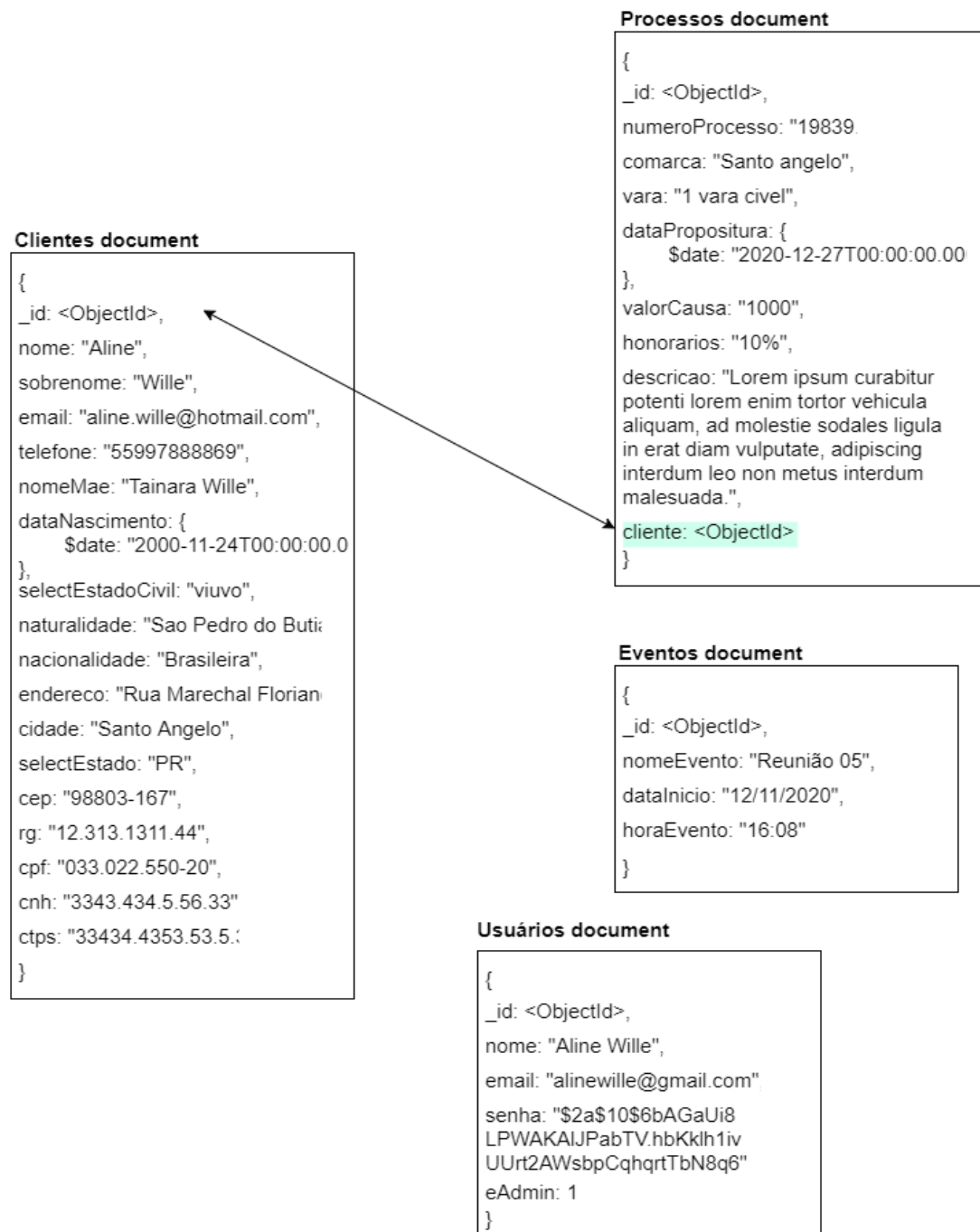


Fonte: Própria (2021)

2.3.2 Modelo de Dados

Um banco de dados orientado a documentos como o MongoDB, pode ser representado através de diagramas de modelos de dados. Este diagrama, pode ser analisado logo abaixo, na figura 21.

Figura 21: Diagrama de modelo de dados



Fonte: Própria (2020)

2.3.3 Models e Schemas

Um *model* faz referência a uma *collection* do banco de dados. Neste projeto foram criados os *models* Cliente, Processo, Evento e Usuario, estes foram utilizados para criar e manipular suas respectivas *collections* com o *mongoose*.

Um *schema* é uma coleção de objetos que é utilizada através do módulo *mongoose* para mapear as *collections* da base de dados. Este mapeamento é feito através de ODM (*Object Document Mapping*), em português Mapeamento Objeto Documento.

Segundo a documentação do *Mongoose*, “tudo em Mongoose começa com um *schema*. Cada *schema* mapeia uma coleção do MongoDB e define a forma dos documentos dentro dessa coleção.”

A) Schema Cliente

```
const mongoose = require("mongoose")
const Schema = mongoose.Schema
```

```
//criando o schema no mongodb
const Cliente = new Schema({
  nome:{
    type: String,
    required: true
  },
  sobrenome:{
    type: String,
    required: true
  },
  email:{
    type: String,
    required: true
  },
  telefone:{
    type: String,
  },
  nomeMae:{
    type: String,
    required: true
  },
  dataNascimento:{
    type: Date,
    required: true
  }
})
```

```

    },
    selectEstadoCivil:{
      type: String,
      required: true
    },
    naturalidade:{
      type: String,
      required: true
    },
    nacionalidade:{
      type: String,
      required: true
    },
    endereco:{
      type: String,
      required: true
    },
    cidade:{
      type: String,
      required: true
    },
    selectEstado:{
      type: String,
      required: true
    },
    cep:{
      type: String,
      required: true
    },
    rg:{
      type: String,
      required: true
    },
    cpf:{
      type: String,
      required: true
    },
    cnh:{
      type: String,
      required: true
    },
    ctps:{
      type: String,
      required: true
    },
    processo:{
      type:String,
    }
  })

```

```
mongoose.model("clientes", Cliente)
```

B) *Schema* Processo

```
const mongoose = require("mongoose")
const Schema = mongoose.Schema

const Processo = new Schema({
  numeroProcesso:{
    type: String,
    required: true
  },
  comarca:{
    type: String,
    required: true
  },
  vara:{
    type: String,
    required: true
  },
  dataPropositura:{
    type: String,
    required: true
  },
  cliente:{
    type: Schema.Types.ObjectId,
    ref:"clientes",
    required: true
  },
  valorCausa:{
    type: Date,
    required: true
  },
  honorarios:{
    type: String,
    required: true
  },
  descricao:{
    type: String,
    required: true
  }
})
mongoose.model("processos", Processo)
```

C) *Schema* Evento

```
const mongoose = require("mongoose")
const Schema = mongoose.Schema

const Evento = new Schema({
```

```

    nomeEvento:{
      type: String,
      required: true
    },
    dataInicio:{
      type: String,
      required: true
    },
    horaEvento:{
      type: String,
      required: true
    }
  })
mongoose.model("eventos", Evento)

```

D) *Schema* Usuario

```

const mongoose = require("mongoose")
const Schema = mongoose.Schema

const Usuario = new Schema({
  nome:{
    type: String,
    required: true
  },
  email:{
    type: String,
    required: true
  },
  senha:{
    type: String,
    required: true
  },
  eAdmin:{
    type:Number,
    default:0
  }
})
mongoose.model("usuarios", Usuario)

```

E) Glossário de Termos

Quadro 1: *Schema* Cliente

Cliente		
Descrição	Atributo	Tipo
Nome do Cliente	nome	String
Sobrenome do Cliente	sobrenome	String
Email do Cliente	email	String
Telefone para Contato	telefone	String
Nome da Mãe do Cliente	nomeMae	String
Data de Nascimento do Cliente	dataNascimento	String
Estado Civil do Cliente	selectEstadoCivil	String
Naturalidade do Cliente	naturalidade	String
Nacionalidade do Cliente	nacionalidade	String
Endereço do Cliente	endereco	String
Cidade do Cliente	cidade	String
Estado do Cliente	selectEstado	String
CEP do Cliente	cep	String
RG do Cliente	rg	String
CPF do Cliente	cpf	String
CNH do Cliente	cnh	String
CTPS do Cliente	ctps	String
Identificador	_id	Object

Quadro 2: *Schema Processo*

Processo		
Descrição	Atributo	Tipo
Número do Processo	numeroProcesso	String
Comarca do Processo	comarca	String
Vara do Processo	vara	String
Data de Propositura do Processo	dataPropositura	String
Referência ao id do cliente	cliente	Object
Valor da Causa	valorCausa	String
Valor dos Honorários do Processo	honorarios	String
Descrição do Processo	descricao	String
Identificador	_id	Object

Quadro 3: *Schema Evento*

Evento		
Descrição	Atributo	Tipo
Nome do Evento	nomeEvento	String
Data em que Ocorrerá o Evento	DataInicio	String
Hora em que Ocorrerá o Evento	horaEvento	String
Identificador	_id	Object

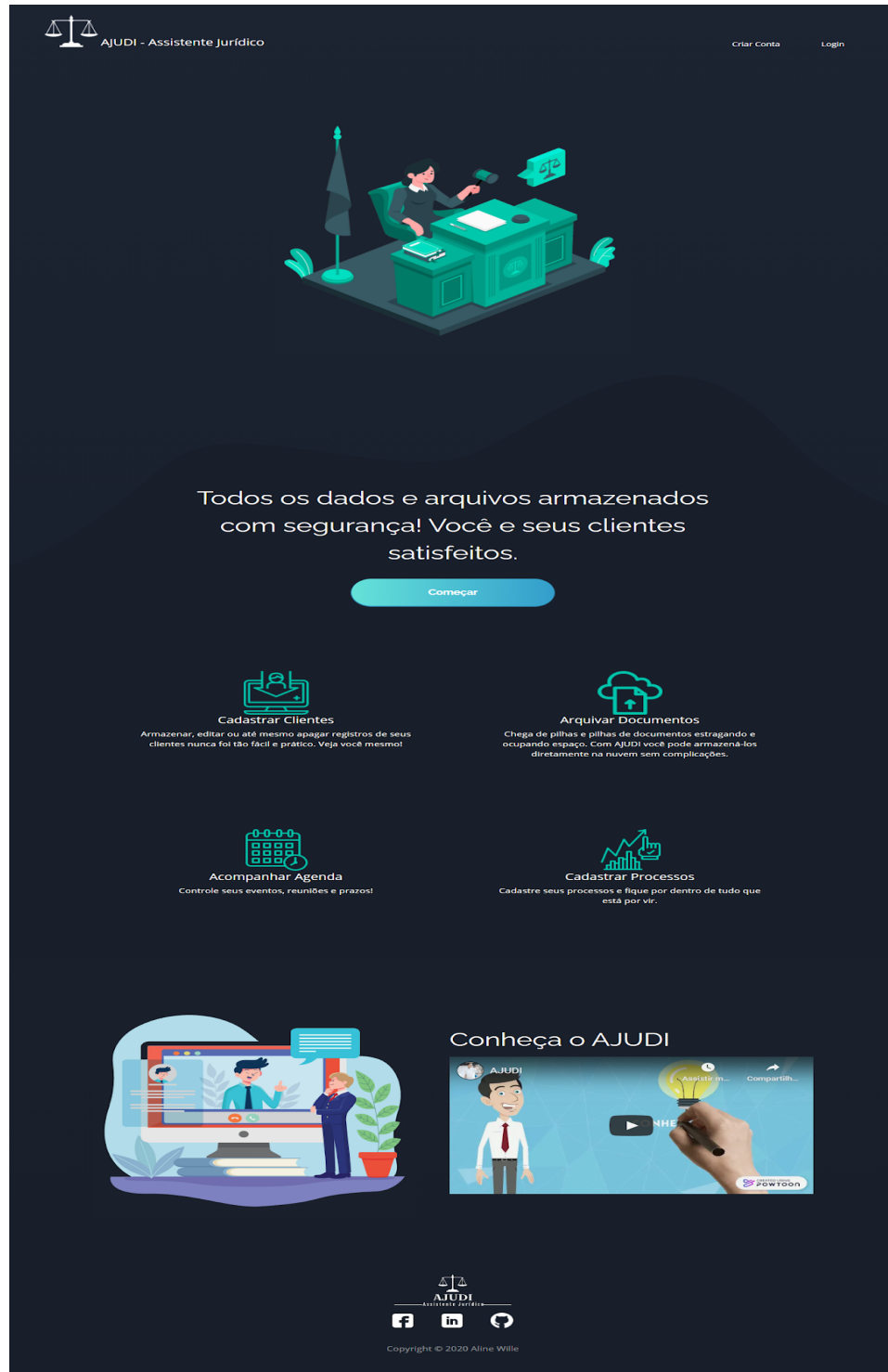
Quadro 4: *Schema Usuario*

Usuario		
Descrição	Atributo	Tipo
Nome do Usuário	nome	String
Email do Usuário	email	String
Senha do Usuário	senha	String
Identificador de Admins	eAdmin	Number
Identificador	_id	Object

2.4 Telas do Sistema

A figura 22 mostra a tela inicial do sistema, com somente os botões de login e criar conta.

Figura 22: Tela inicial do sistema



Fonte: Própria (2021)

Para acessar a página de clientes, como mostra a figura 23, o usuário deve estar logado em sua conta, mantendo assim os dados protegidos. Nota-se que se quando o usuário entra com sua senha e email, os botões de login e criar conta são substituídos pelos botões de clientes, documentos, processos, agenda e sair.

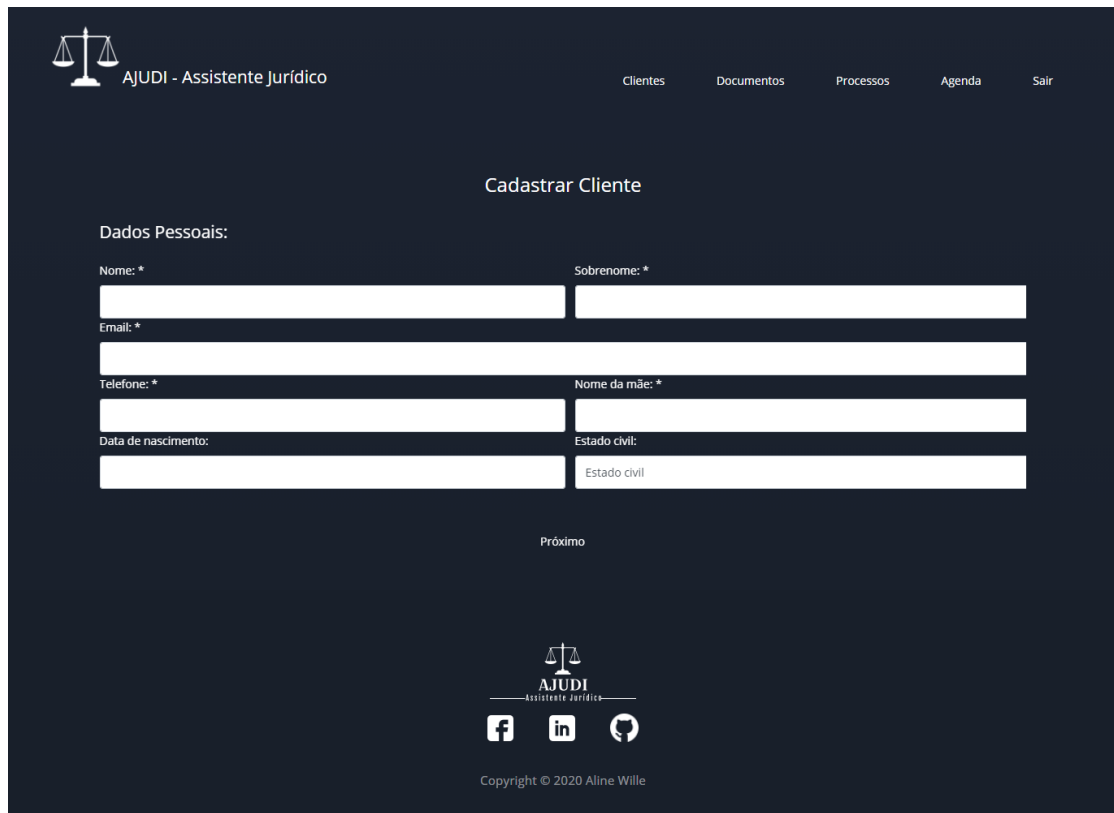
Figura 23 : Tela de clientes cadastrados

The screenshot displays the 'Lista de clientes cadastrados' (List of registered clients) page. At the top, there is a navigation bar with the AJUDI logo and the text 'AJUDI - Assistente Jurídico'. The navigation menu includes links for 'Clientes', 'Documentos', 'Processos', 'Agenda', and 'Sair'. The main heading is 'Lista de clientes cadastrados', with a sub-link 'Cadastrar novo cliente'. Below this, the name 'Tainara Andresa Taborda' is listed. The client's details are organized into three columns: 'Dados Pessoais' (Personal Data), 'Dados de Localização' (Location Data), and 'Documentação' (Documentation). The 'Dados Pessoais' column includes fields for 'Telefone', 'Nome da Mãe', 'Estado Civil', and 'Data de Nascimento'. The 'Dados de Localização' column includes 'Naturalidade', 'Estado', 'Cidade', 'Endereço', and 'CEP'. The 'Documentação' column includes 'RG', 'CPF', 'CNH', and 'CTPS'. At the bottom of the client details, there are buttons for 'Editar' and 'Apagar'. The footer of the page features the AJUDI logo, social media icons for Facebook, LinkedIn, and WhatsApp, and a copyright notice: 'Copyright © 2020 Aline Wille'.

Dados Pessoais	Dados de Localização	Documentação
Telefone : 00 0 0000-0000	Naturalidade : Sao Pedro do Butia	RG : 00000000000
Nome da Mãe : Tainara Andresa Klug Taborda	Estado : Estado	CPF : 000.000.000-00
Estado Civil : selected	Cidade : Santo Angelo	CNH : 000000000-00
Data de Nascimento: 11/31/3232	Endereço: Rua Marechal Floriano	CTPS: 0000000-0000
CEP: 00000-000		

Fonte: Própria (2021)

As figuras 24, 25 e 26 referem-se aos formulários para cadastrar novos clientes, separando as informações nas seções: dados pessoais, dados de localização e documentação, respectivamente.

Figura 24 : Tela de cadastro do cliente


AJUDI - Assistente Jurídico

Clientes Documentos Processos Agenda Sair

Cadastrar Cliente

Dados Pessoais:

Nome: * Sobrenome: *

Email: *

Telefone: * Nome da mãe: *

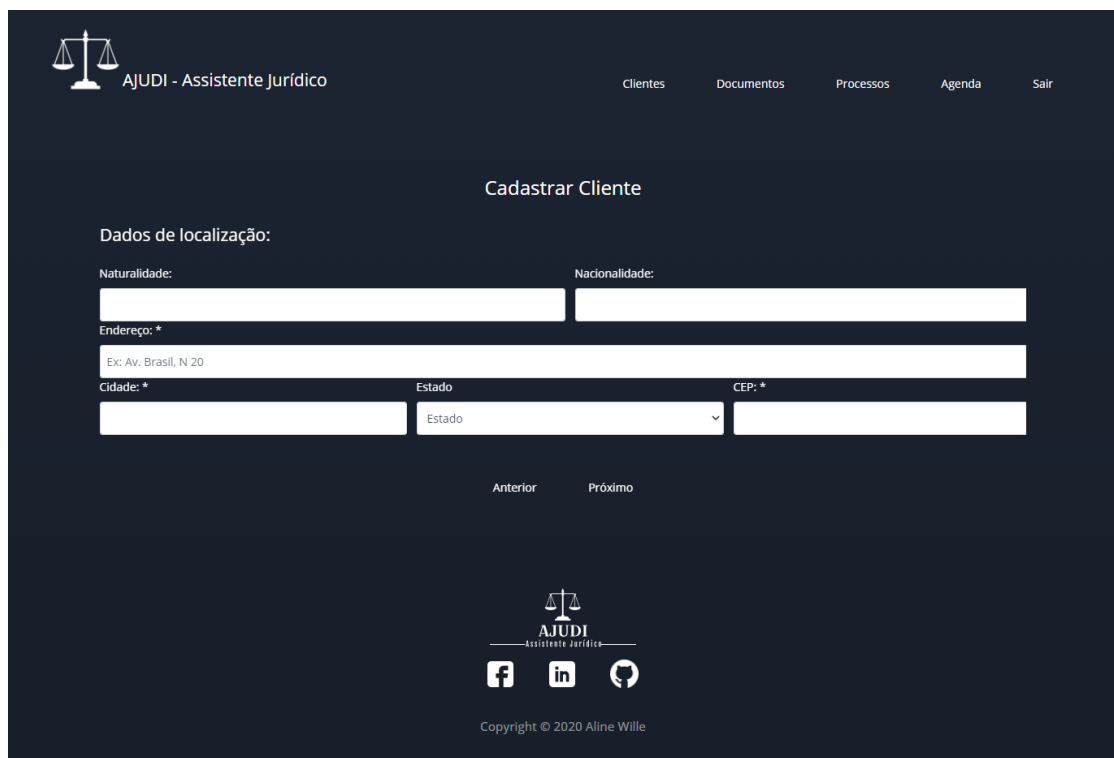
Data de nascimento: Estado civil:

AJUDI
Assistente Jurídico

f in

Copyright © 2020 Aline Wille

Fonte: Própria (2021)

Figura 25 : Tela de cadastro do cliente


AJUDI - Assistente Jurídico

Clientes Documentos Processos Agenda Sair

Cadastrar Cliente

Dados de localização:

Naturalidade: Nacionalidade:

Endereço: *

Ex: Av. Brasil, N 20

Cidade: * Estado CEP: *

AJUDI
Assistente Jurídico

f in

Copyright © 2020 Aline Wille

Fonte: Própria (2021)

Figura 26 : Tela de cadastro do cliente

AJUDI - Assistente Jurídico

Clientes Documentos Processos Agenda Sair

Cadastrar Cliente

Documentação:

RG: *

CPF: *

CNH: *

CTPS: *

Anterior Próximo

Salvar

AJUDI - Assistente Jurídico

Facebook LinkedIn GitHub

Copyright © 2020 Aline Wille

Fonte: Própria (2021)

Para acessar a página de arquivo de documentos, deve-se passar por mais uma etapa de segurança, a página de login com o google, que pode ser vista na figura 27.

Figura 27: Tela de login com o Google para acessar os documentos do Drive

AJUDI - Assistente Jurídico

Clientes Documentos Processos Agenda Sair

Para acessar os arquivos,
Entre com sua conta do Google!

Google

AJUDI - Assistente Jurídico

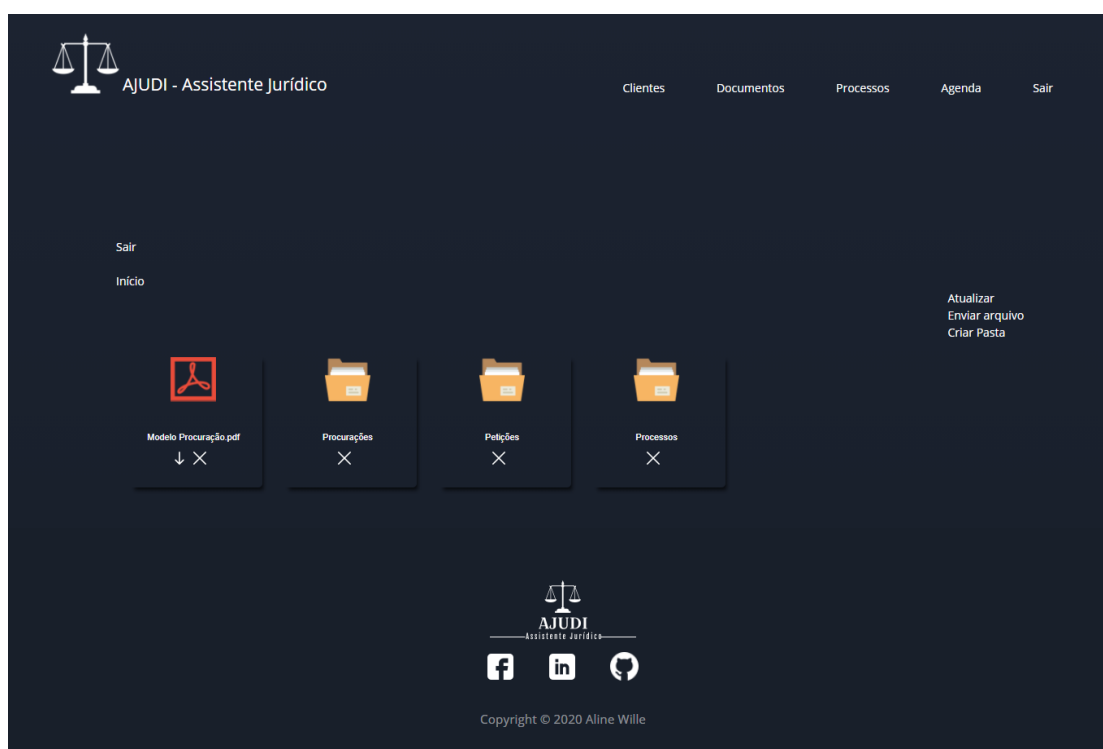
Facebook LinkedIn GitHub

Copyright © 2020 Aline Wille

Fonte: Própria (2021)

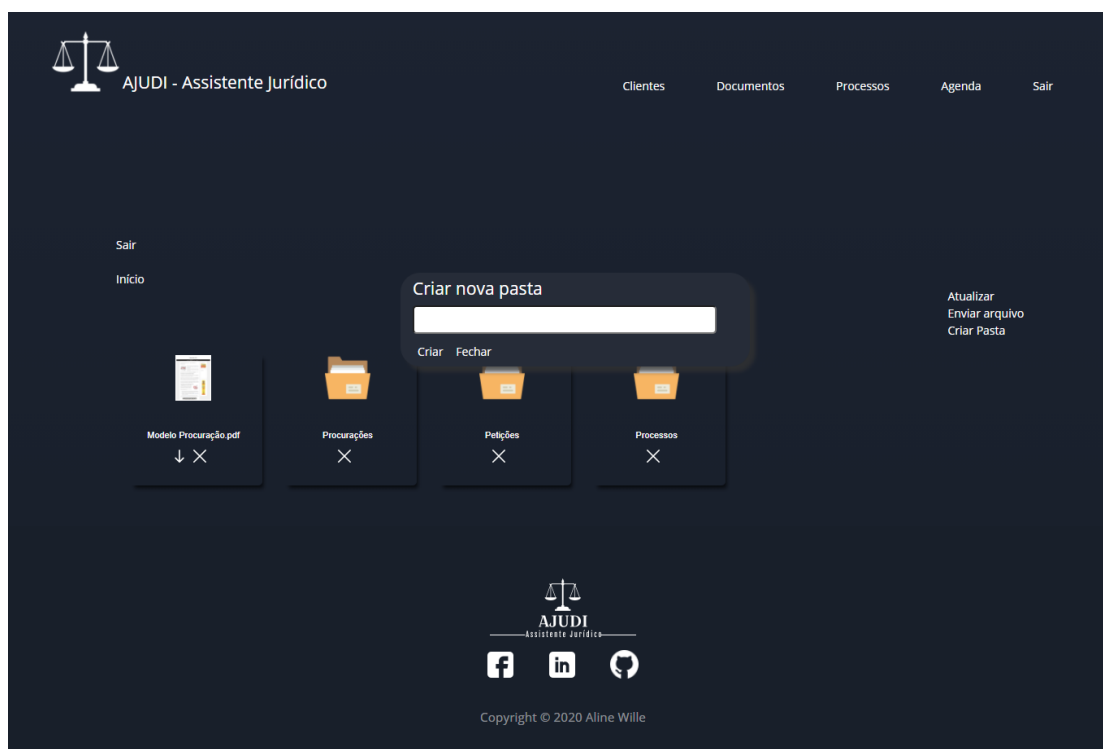
Após fazer login, o usuário é redirecionado para a página onde são armazenados todos os arquivos de sua conta, na figura 28 pode-se observar esta página. Nela constam os botões de sair, início, atualizar, enviar arquivo e criar pasta. Também pode-se analisar que abaixo de cada arquivo existe um botão “X”, para excluir aquele arquivo, e um botão “↓”, para fazer download do mesmo. Já nas pastas, existe apenas o botão “X”, pois é possível somente excluí-las.

Figura 28: Tela de documentos



Fonte: Própria (2021)

Figura 29: Tela de criação de pastas no Drive



Fonte: Própria (2021)

A figura 30 trata-se da página de processos, nela constam os processos que estão cadastrados no sistema juntamente de suas informações e os botões que possibilitam excluir um determinado cadastro ou alterar algum de seus dados.

Figura 30: Tela de processos



Fonte: Própria (2021)

Figura 31: Tela de cadastro de um novo processo

AJUDI - Assistente Jurídico

Clientes Documentos Processos Agenda Sair

Novo Processo

Número do processo:

Comarca:

Vara:

Data de propositura:

Valor da causa:

Honorários:

Descrição:

Cliente:

Salvar

AJUDI
Assistente Jurídico

f in

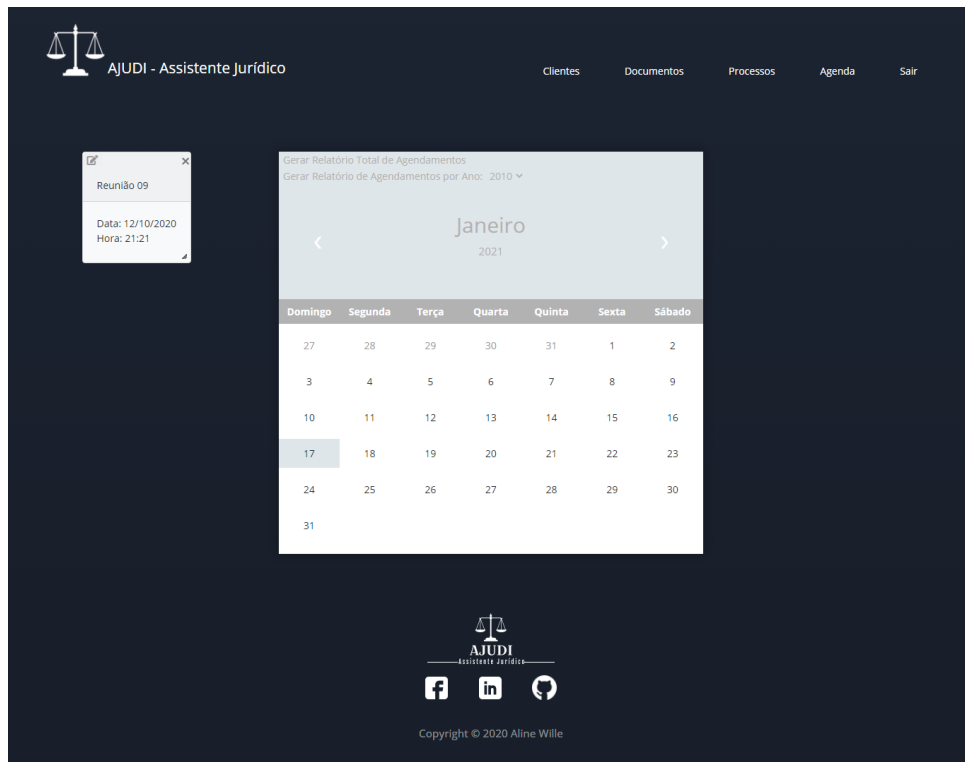
Copyright © 2020 Aline Wille

Fonte: Própria (2021)

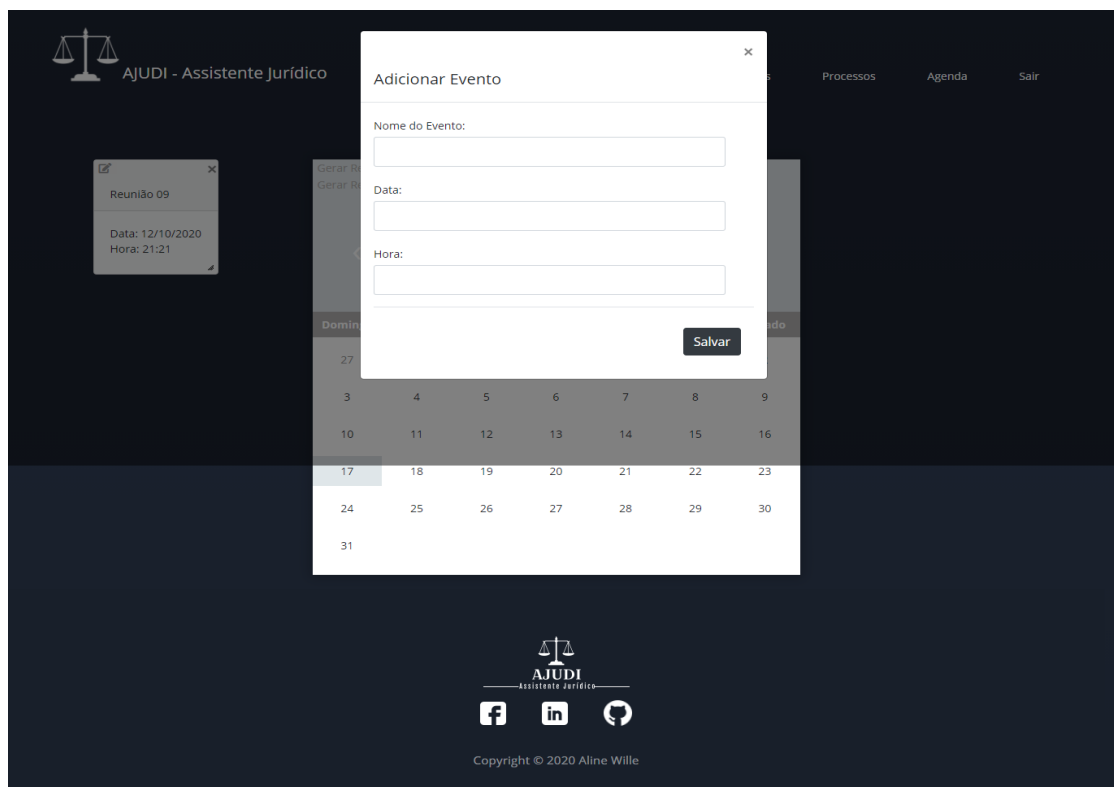
Na figura 32 consta a tela de agendamentos, nela pode ser visto um calendário, onde se clicar em qualquer data, apresenta-se o formulário para cadastrar um novo evento, este, pode ser visto na figura 33.

Na tela de agendamentos, também constam os eventos que estão cadastrados, em forma de *post-its*, estes podem ser editados e excluídos, assim como ter seu tamanho e localização alterados.

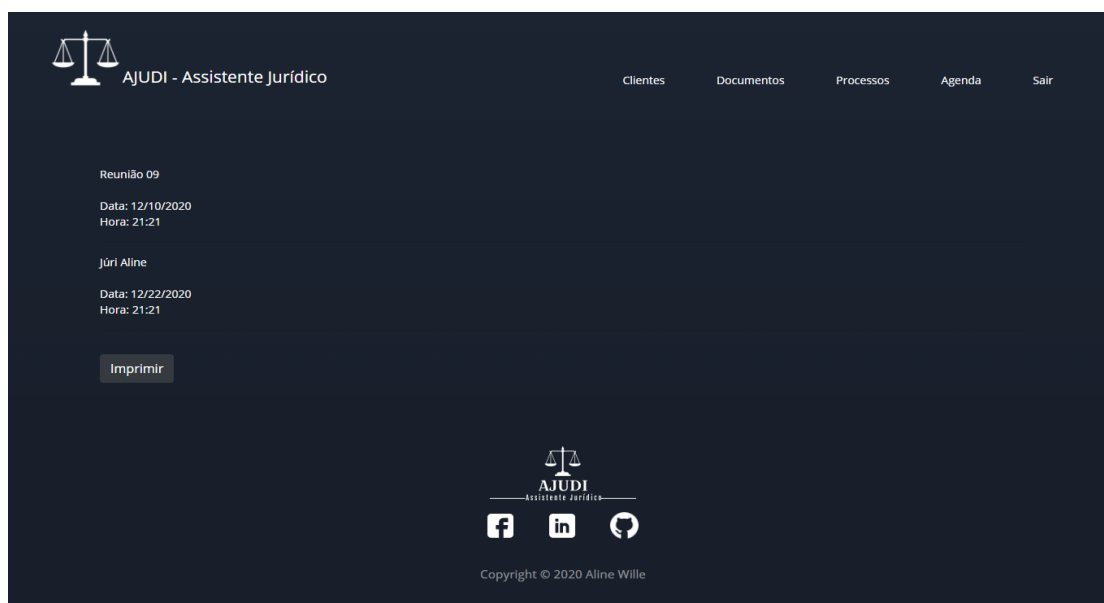
Acima do mês do calendário pode ser visto dois links, o primeiro para gerar um relatório total de agendamentos, e no segundo link o usuário pode escolher o período que ele deseja gerar o relatório. Estes relatórios podem ser vistos nas figuras 34 e 35.

Figura 32: Tela de agenda

Fonte: Própria (2021)

Figura 33 : Tela de cadastro de evento

Fonte: Própria (2021)

Figura 34: Tela de relatórios

Fonte: Própria (2021)

Figura 35:Relatório pronto para ser impresso em PDF

13/01/2021

Relatório de Agendamentos

Relatório de Agendamentos

Reunião 09

Data: 12/10/2020
Hora: 21:21

Júri Aline

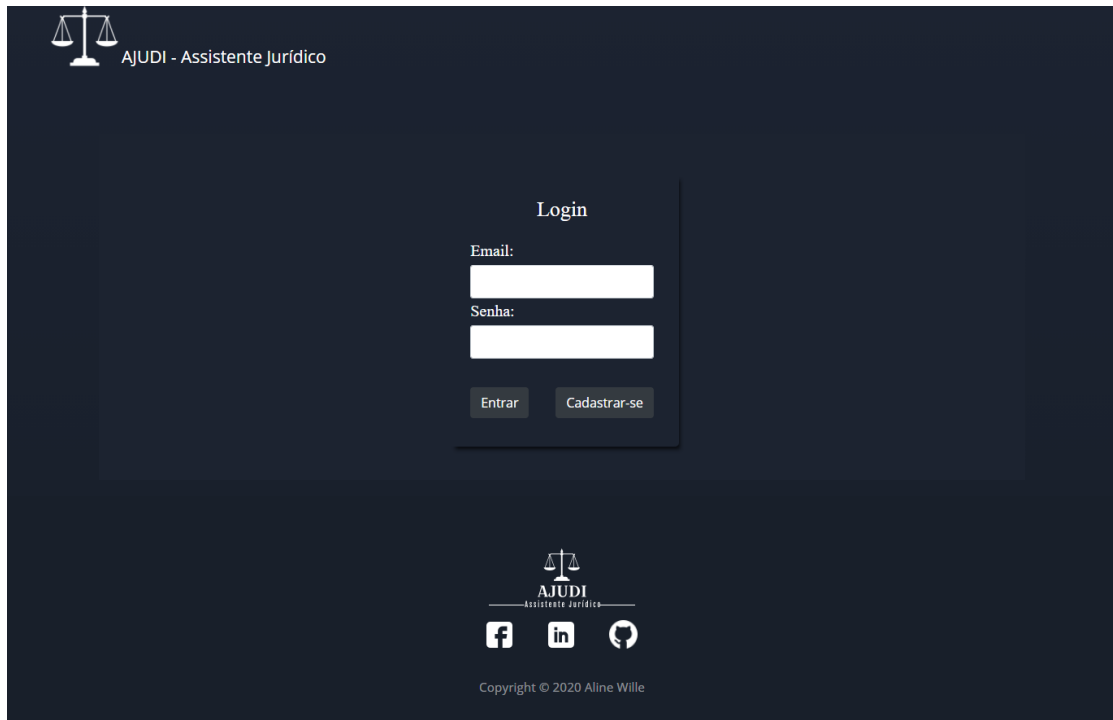
Data: 12/22/2020
Hora: 21:21

1/1

Fonte: Própria (2021)

As imagens a seguir, são as telas de login e cadastro, utilizadas pelos usuários para terem acesso ao sistema. Podem ser analisadas nas figuras 36 e 37, respectivamente.

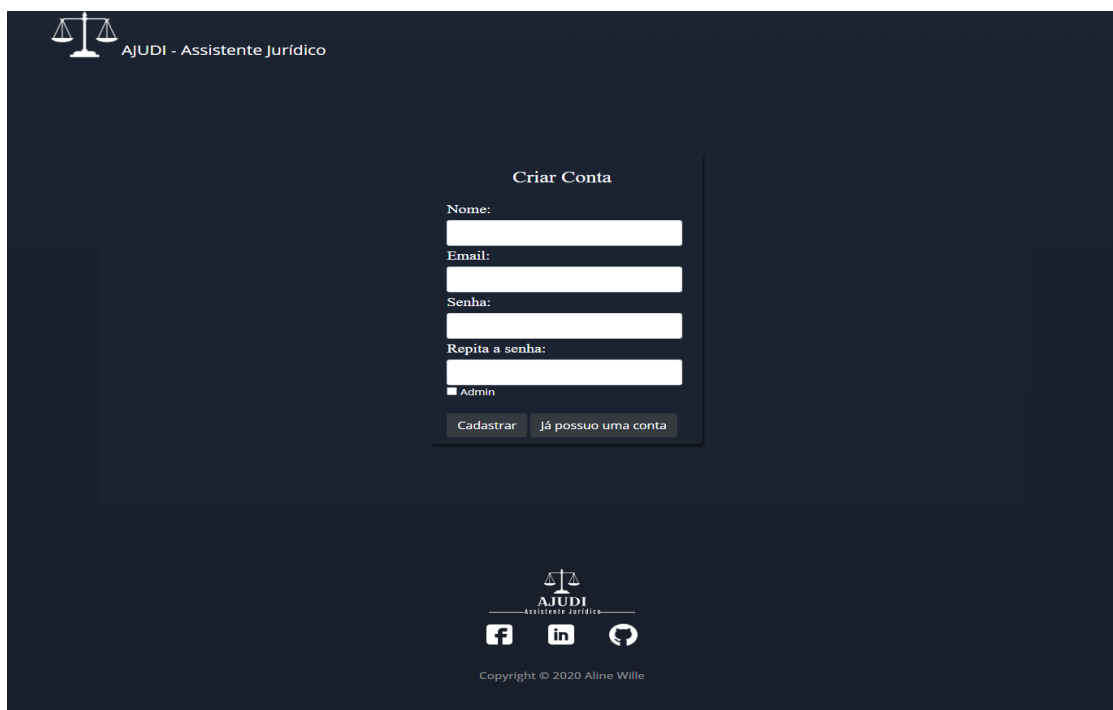
Figura 36:Tela de Login



A screenshot of the login page for the AJUDI - Assistente Jurídico system. The page has a dark blue background. In the top left corner, there is a logo of a scale of justice and the text "AJUDI - Assistente Jurídico". In the center, there is a white box containing the "Login" form. The form has two input fields: "Email:" and "Senha:". Below these fields are two buttons: "Entrar" and "Cadastrar-se". At the bottom of the page, there is a logo of a scale of justice and the text "AJUDI - Assistente Jurídico", followed by social media icons for Facebook, LinkedIn, and YouTube. At the very bottom, there is a copyright notice: "Copyright © 2020 Aline Wille".

Fonte: Própria (2021)

Figura 37:Tela de registro



A screenshot of the registration page for the AJUDI - Assistente Jurídico system. The page has a dark blue background. In the top left corner, there is a logo of a scale of justice and the text "AJUDI - Assistente Jurídico". In the center, there is a white box containing the "Criar Conta" form. The form has five input fields: "Nome:", "Email:", "Senha:", "Repita a senha:", and a checkbox labeled "Admin". Below these fields are two buttons: "Cadastrar" and "Já possuo uma conta". At the bottom of the page, there is a logo of a scale of justice and the text "AJUDI - Assistente Jurídico", followed by social media icons for Facebook, LinkedIn, and YouTube. At the very bottom, there is a copyright notice: "Copyright © 2020 Aline Wille".

Fonte: Própria (2021)

A imagem a seguir, mostra a mensagem exibida quando um usuário comum tenta excluir algum dado de um cliente ou processo. Esses dados podem ser excluídos apenas por administradores, por questão de segurança.

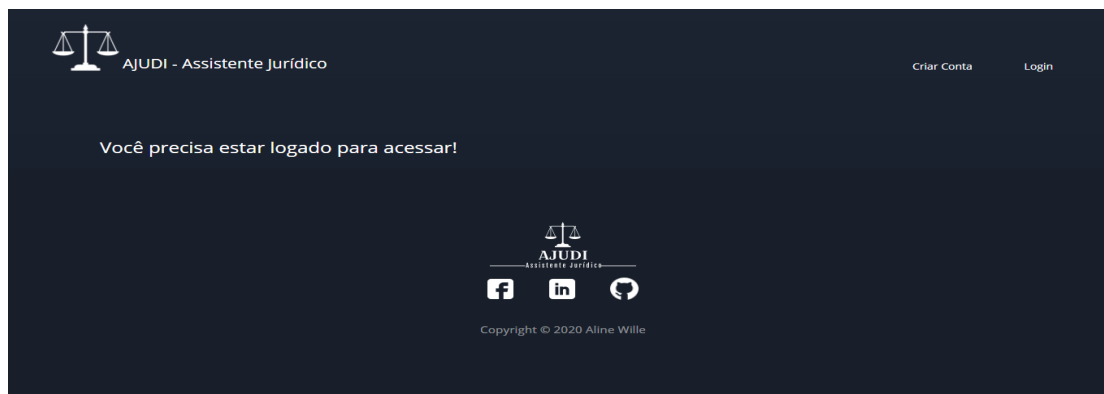
Figura 38: Mensagem exibida ao tentar apagar um registro



Fonte: Própria (2021)

Esta imagem por sua vez, é mostrada quando alguém tenta acessar alguma funcionalidade do sistema pela sua URL, por exemplo <http://localhost:8081/cliente>, sem estar logado no sistema. Solução desenvolvida para fins de segurança.

Figura 39: Mensagem exibida ao tentar acessar uma página sem estar logado no sistema



Fonte: Própria (2021)

3 RESULTADOS E DISCUSSÕES

3.1 Questionário de avaliação e testes do sistema

No momento em que o desenvolvimento do sistema deu-se por concluído, foram realizados testes a fim de comprovar sua eficácia e usabilidade. Estes testes foram realizados no mês de janeiro no escritório Advogados Associados S/S, em Santo Ângelo - RS, por três funcionários. A avaliação foi de nota 1 para muito ruim a nota 10 para muito bom. Para demonstrar o resultado foi feita uma média das respostas de todos os avaliadores.

Para realizar os testes, foram utilizados os próprios computadores do escritório em questão e um notebook como servidor local. Todas as máquinas estavam conectadas ao mesmo *wi-fi*, possibilitando assim um fácil acesso a todos. O resultado do questionário pode ser analisado no quadro 5.

Também foram realizadas as seguintes questões para as pessoas que testaram o sistema, estas entraram em um consenso para unificar suas respostas.

Cite 3 pontos fortes do sistema :

- Fácil compreensão e manuseio do sistema.
- Permite reunir dados de clientes, processos e arquivos em um único sistema, o que facilita o armazenamento e principalmente o trabalho de busca por informações, já que o escritório trabalha com grande volume de clientes.
- Permite extinguir o arquivo físico do Escritório, ocupando apenas o arquivo virtual desse sistema, deixando o trabalho nitidamente mais célere.

Por favor, dê sugestões para a melhoria do sistema :

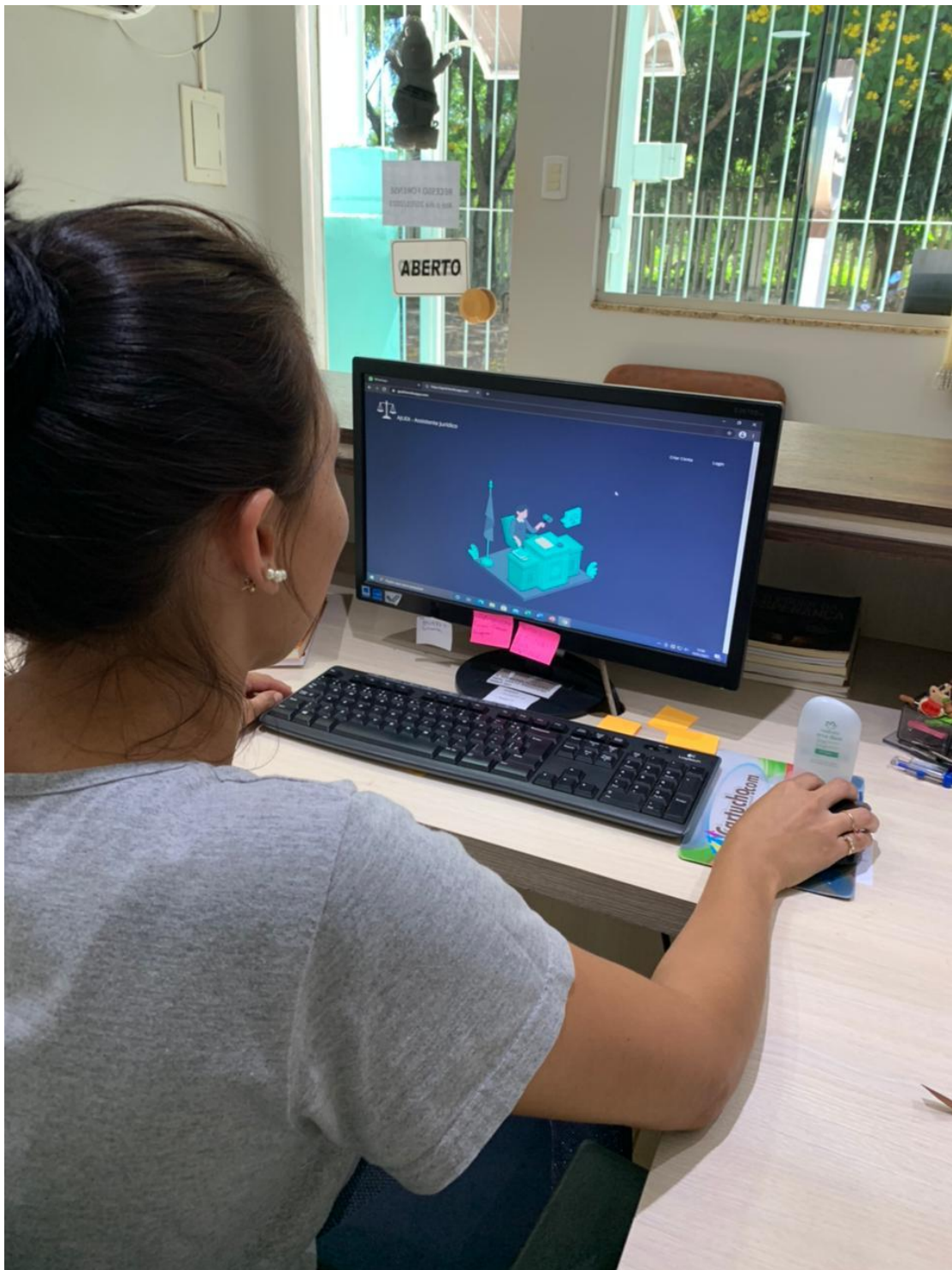
- Acesso/ vinculação aos sites dos Tribunais e Instâncias Superiores, vinculando ao número do processo cadastrado no sistema Ajudi, para busca imediata/automática do andamento dos processos.

Quadro 5: Questionário de avaliação do sistema.

Afirmação	Avaliação
O design do sistema é atraente.	9
O conteúdo do sistema é relevante para os meus interesses.	10
Foi fácil entender o sistema e utilizá-lo.	10
Me senti confortável usando o sistema	9,5
O sistema promove uma melhor organização dentro da empresa.	10
Eu recomendaria este sistema para meus colegas.	10
Gostaria de utilizar este sistema novamente	10
Tive sentimentos positivos de eficiência no desenrolar do teste do sistema	9
O sistema foi eficiente, em comparação com outros softwares	9
O design do sistema é atraente.	9

Nas imagens 39, 40 e 41 constam alguns funcionários do escritório em que os testes foram realizados usando o sistema.

Figura 40: Teste funcional do sistema



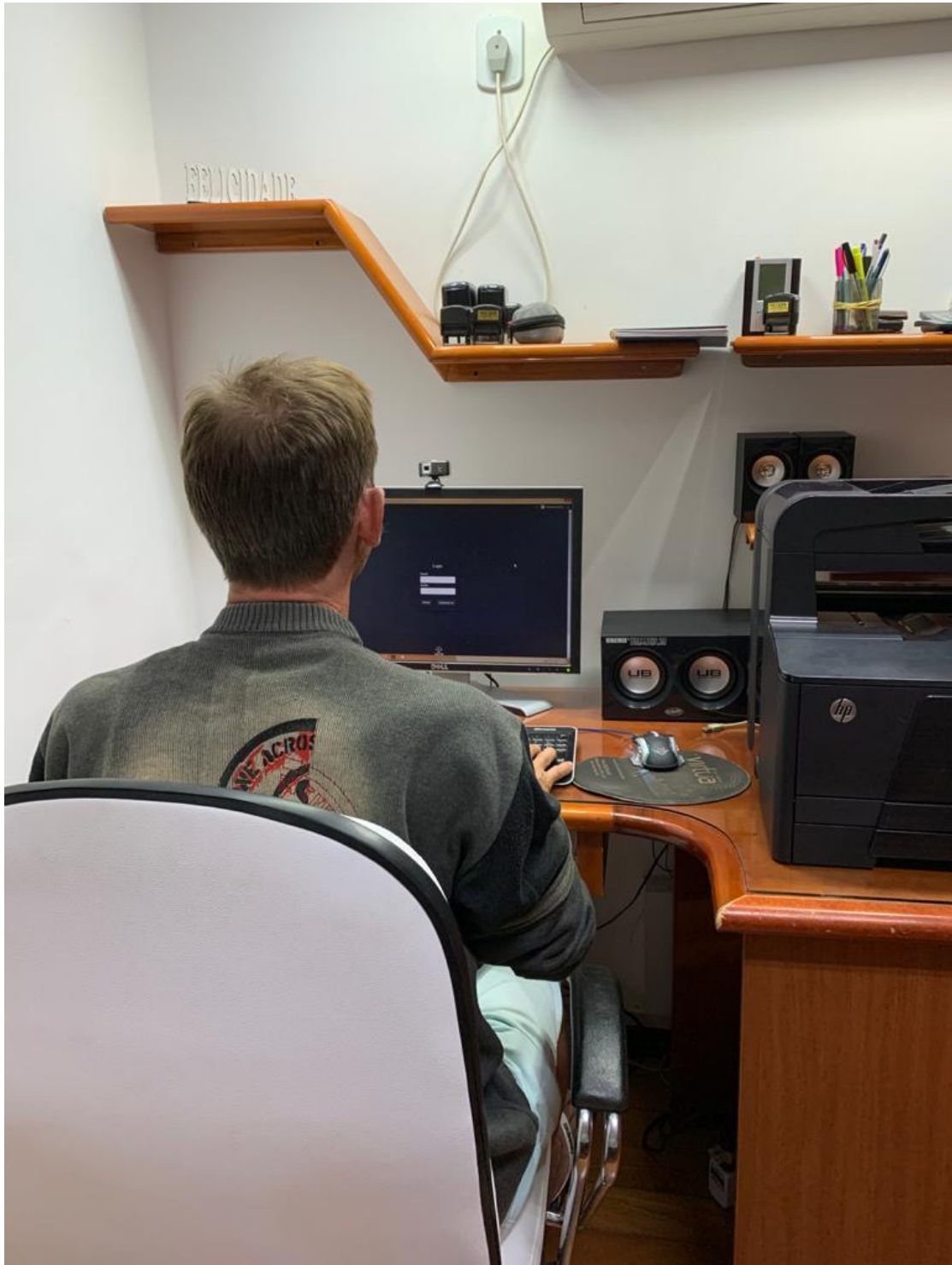
Fonte: Própria (2021)

Figura 41: Teste funcional do sistema



Fonte: Própria (2021)

Figura 42: Teste funcional do sistema



Fonte: Própria (2021)

3.1.1 Resultados do Sistema

Com a execução dos testes pode-se evidenciar que o sistema tem capacidade de ser utilizado no cotidiano da empresa e dar melhores resultados para a mesma, assim como solucionar problemas reais do estabelecimento. Notou-se também, que o programa gera para todos os profissionais do escritório um ganho de produtividade, pois diminui a busca de dados em papéis e extingue o ato de anotar eventos manualmente em um caderno.

Na forma em que o sistema foi desenvolvido, não é possível acompanhar os andamentos dos processos em diários oficiais, ficando assim tal funcionalidade a ser implementada futuramente.

4 CONSIDERAÇÕES FINAIS

Uma empresa que deseja ter resultados positivos no decorrer de seu trabalho necessita de um sistema gerenciador. O avanço exponencial da tecnologia dos últimos anos, e o grande volume de dados está fazendo com o que o gerenciamento da maioria das empresas deixe de ser manual e torne-se virtual.

Este trabalho teve como objetivo desenvolver um sistema, denominado AJUDI - Assistente Jurídico, que visa aprimorar a qualidade de gestão de escritórios de advocacia, com enfoque na otimização de tempo em relação a busca de dados e documentos pessoais de clientes, assim como, informações processuais. Além disso, sistematizar o agendamento de consultas e compromissos importantes. O mesmo, foi inspirado em trabalhos correlatos como o Astrea, Advbox e o Gojur, ambos são sistemas voltados para o âmbito jurídico com os mesmos objetivos citados, porém nenhum se propõe a realizar a tarefa de armazenar arquivos.

Dado o exposto, e averiguado que todos os objetivos definidos inicialmente foram devidamente cumpridos e, levando em consideração esses aspectos, foram realizados testes em um escritório de advocacia e, pode-se analisar que a organização não só obteve um melhor desempenho num todo, como também se mostrou adepta perante as novas tecnologias e como consequência disso teve destaque entre diversas empresas do ramo jurídico.

REFERÊNCIAS

ANGELINI, F; MACIEIRA, M; CUNHA, M; MARANHÃO, M: Sistema Integrado de Gestão para Escritórios de Advocacia, E-Book. ISBN: 9788553060160. Disponível em:

<https://books.google.com.br/books?hl=pt-BR&lr=&id=ucd-DwAAQBAJ&oi=fnd&pg=PT4&dq=gest%C3%A3o+de+escritorio+de+advocacia&ots=GhspID13Zo&sig=B5GwYLiapCYIK7dA1Sd8fbfMdMU#v=onepage&q&f=false> Consultado em 13 de abril de 2020

Ajax. Disponível em: [Ajax | jQuery API Documentation](#). Consultado em 29 de março de 2020.

Bcryptjs. Disponível em: [bcryptjs - npm \(npmjs.com\)](#). Consultado em 01 de dezembro de 2020.

Body Parser. Disponível em: [bcryptjs - npm \(npmjs.com\)](#) Consultado em 20 de novembro de 2020.

Bootstrap. Disponível em: <https://getbootstrap.com/>. Consultado em 29 de março de 2020.

Como surgiu a internet? <https://brasilecola.uol.com.br/curiosidades/como-surgiu-a-internet.htm>. Acesso em 06 de março de 2020.

CSS. Disponível em: [CSS | MDN \(mozilla.org\)](#) Consultado em 29 de março de 2020.

DATE, C. J: Introdução a sistemas de banco de dados (português) – GEN LTC, 2004.

DINIZ, M: Dicionário jurídico universitário - São Paulo: Saraiva, 2010.

Express. Disponível em: [Express - framework de aplicativo da web Node.js \(expressjs.com\)](#). Consultado em 20 de novembro de 2020.

Flash. Disponível em: [flash - npm \(npmjs.com\)](#). Consultado em 20 de novembro de 2020.

Google drive API. Disponível em: [Google Drive API | Google Developers](#). Consultado em 01 de dezembro de 2020.

GUEDES, G: UML 2: Uma abordagem prática - São Paulo: Novatec, 2009.

Handlebars. Disponível em: [Handlebars \(handlebarsjs.com\)](#) Consultado em 01 de dezembro de 2020.

HEUSER, A. Projeto de banco de dados: 6. ed. - Porto Alegre: Bookman, 2009.

HTML. Disponível em: [HTML: Linguagem de Marcação de Hipertexto | MDN](#)

([mozilla.org](https://www.mozilla.org)) Consultado em 29 de março de 2020.

JavaScript. Disponível em: [JavaScript | MDN \(mozilla.org\)](https://developer.mozilla.org) Consultado em 20 de novembro de 2020.

Jquery. Disponível em: [jQuery](https://jquery.com) Consultado em 29 de março de 2020.

MELO, A; NASCIMENTO, M. Php Profissional: Aprenda a desenvolver sistemas profissionais orientados a objetos com padrões de projeto - Novatec, 2007.

MongoDB. Disponível em: [The most popular database for modern apps | MongoDB](https://www.mongodb.com) Consultado em 01 de dezembro de 2020.

Mongoose. Disponível em: [Mongoose ODM v5.12.0 \(mongoosejs.com\)](https://mongoosejs.com) Consultado em 01 de dezembro de 2020.

Nodemon. Disponível em: [nodemon - npm \(npmjs.com\)](https://nodemon.io) Consultado em 01 de dezembro de 2020.

Node.js. Disponível em: [Node.js](https://nodejs.org) Consultado em 20 de novembro de 2020.

Passport. Disponível em: [passport - npm \(npmjs.com\)](https://www.passportjs.org) Consultado em 01 de dezembro de 2020.

SILVA, M. Ajax com jQuery - Requisições AJAX com a simplicidade de Jquery - São Paulo: Novatec, 2009.

SILVA, M. JavaScript - Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript - São Paulo: Novatec, 2010