



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

ALESSANDRO LIA FOOK SANTOS

**LEXFIRMA:
SISTEMA DE GESTÃO DE CONTRATOS ADVOCATÍCIOS**

CAMPINA GRANDE - PB

2021

ALESSANDRO LIA FOOK SANTOS

LEXFIRMA:

SISTEMA DE GESTÃO DE CONTRATOS ADVOCATÍCIOS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientadora: Professora Dr. Hyggo Oliveira de Almeida.

CAMPINA GRANDE - PB

2021



S2371 Santos, Alessandro Lia Fook.
LexFirma: sistema de gestão de contratos
advocáticos. / Alessandro Lia Fook Santos. - 2021.

10 f.

Orientador: Prof. Dr. Hyggo de Oliveira Almeida.

Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Gestão eletrônica de contratos advocáticos. 2. Sistema de informação em site. 3. Contratos advocáticos - gestão eletrônica. 4. Escritório de advocacia - automação. 5. Sistema de gestão de contratos - criação. 6. Arquitetura Model-View-Controller. 7. Padrão MVC de arquitetura. 8. LexFirma - sistema de gestão de contratos advocáticos. I. Almeida, Hyggo de Oliveira Almeida. II. Título.

CDU:004.415.2(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

ALESSANDRO LIA FOOK SANTOS

LEXFIRMA:

SISTEMA DE GESTÃO DE CONTRATOS ADVOCATÍCIOS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Professor Dr. Hyggo Oliveira de Almeida
Orientador – UASC/CEEI/UFCG

Professora Dr. Nazareno Ferreira de Andrade
Examinador – UASC/CEEI/UFCG

Professor Dr. Tiago Lima Massoni
Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 25 de MAIO de 2021.

CAMPINA GRANDE - PB

RESUMO (ABSTRACT)

Computerization has become a path of no return in today's times, with the power to facilitate access and understanding of information. In this way, the most diverse areas are joining online systems to store and manage information. It is no different with the legal profession. In recent years, process virtualization has made its agents more and more familiar with electronic systems. However, systems to support the management of the various financial aspects of a law firm are still lacking. For this, we propose *LexFirma*. With this tool, from the lawyer who works alone to the newsstands, they can perform the financial management of their contracts.

LEXFIRMA: Sistema de Gestão de Contratos Advocatícios

Alessandro Lia Fook Santos*

A. Fook*

alessandro.liafook@gmail.com

alessandro.santos@ccc.ufcg.edu.br

Universidade Federal de Campina Grande

Campina Grande, Paraíba

Hyggo Oliveira de Almeida

hyggo@computacao.ufcg.edu.br

Universidade Federal de Campina Grande

Campina Grande, Paraíba

RESUMO

A informatização se tornou um caminho sem volta nos tempos atuais, com o poder de facilitar o acesso e compreensão das informações. Desta forma, as mais diversas áreas estão aderindo a sistemas online para armazenar e gerenciar informações. Com o ramo da advocacia não é diferente. Nos últimos anos, a virtualização dos processos tem tornado seus agentes cada vez mais familiarizados com sistemas eletrônicos. Contudo, ainda faltam sistemas para suporte à gestão das diversas facetas financeiras de um escritório de advocacia. Para isto, propomos o *LexFirma*. Com essa ferramenta, desde o advogado que trabalha sozinho até as bancas podem realizar a gestão financeira de seus contratos.

ACM Reference Format:

Alessandro Lia Fook Santos, A. Fook, and Hyggo Oliveira de Almeida. 2021. LEXFIRMA: Sistema de Gestão de Contratos Advocatícios. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

REPOSITÓRIO

<https://github.com/alessandroliafook/lexfirma>

1 INTRODUÇÃO

O sistema processual brasileiro entrou em uma jornada pela informatização instrumentalizada a partir do ano de 2004 com a criação do Creta, que mais tarde passou a ser denominado de PJE [5]. O PJE ganhou mais força e poder com a regulamentação do processo judicial eletrônico através da *lei 11.419/2006* [11]. Este fato aproximou ainda mais os operadores do direito dos sistemas de informação, incluindo os advogados.

Utilizar sistemas para auxiliar na gestão de escritórios de advocacia é um fator relevante para o sucesso[18], o que demonstra um mercado com grande volume de movimentação. Na mesma pesquisa indica-se que ainda há uma considerável fatia de mercado que não foi explorada, que são os jovens advogados e pequenos escritórios, onde a adesão é menor.

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Muitos desses sistemas oferecem gestão de prazos e documentos, agendas de processos e demais funcionalidades que foram implementadas no sistema de gestão processual do poder judiciário PJE [4]. Assim, permite-se ao jovem advogado realizar o controle do fluxo de seus processos utilizando a ferramenta governamental.

Assim, é importante ter a perspectiva de que no ano de 2019 existiam aproximadamente 1 milhão e 100 mil advogados registrados na OAB com uma média de acréscimo diário em 200 novos advogados[9]. No mesmo sentido, e reforçando a estatística mencionada, o MEC informou que em 2017 existiam aproximadamente 879 mil alunos matriculados nos diversos cursos de direito do país, sendo que neste mesmo ano cerca de 113 mil estudantes concluiriam o curso e 25% deles seria aprovado na OAB considerando os resultados dos anos anteriores[8].

Considerando então que as disciplinas de gestão financeira, não fazem parte dos currículos de cursos superiores de Direito no Brasil [2] o jovem advogado termina sua formação sem estar devidamente preparado para a administração de seus contratos.

Neste norte, o acesso à informação através de sites e aplicações se tornou cada vez mais simplificado, com a ampliação dos pacotes de dados oferecidos pelos gestores de acesso à internet, assim como a popularização dos smartphones. A falta de conhecimento formal sobre gestão de negócio faz com que o jovem advogado recorra a planilhas e outros artifícios físicos, como agendas e afins, para possibilitar o controle do fluxo de caixa de suas atividades.

Portanto, o intuito do presente trabalho é realizar a apresentação de um sistema simplificado de cadastro de informações financeiras voltado para o jovem advogado, inteiramente gratuito e capitalizado através de um painel de propagandas. Este sistema ocuparia o espaço inexplorado pelos atuais sistemas, que apresentam um custo muitas vezes inacessível a quem acabou de ingressar em um mercado de trabalho tão concorrido.

O texto é dividido em **quatro partes**: **a seção um** apresenta uma introdução sobre o sistema e as motivações de seu desenvolvimento; **a seção dois** trata sobre a solução que o sistema representa trazendo informações coletadas junto a jovens advogados; **a seção três** apresenta o relato da experiência de desenvolvimento do sistema; e **na seção quatro** apresentamos propostas de como avançar no desenvolvimento do sistema como idéias de trabalhos futuros.

2 SOLUÇÃO

O *LexFirma* é um sistema de informação disponibilizado através de um site que irá auxiliar os jovens advogados a realizar a gestão de seus contratos advocatícios e, por consequência, de suas finanças de forma gratuita e simples.

Já existem no mercado soluções para o gerenciamento de processos online, seja através de aplicativos, sites ou ambos. Todos

têm em comum a cobrança de valores, com gratuidade limitada seja pelo tempo, pelo número de usuários ou de processos.

Com o ingresso de cada vez mais advogados jovens da Geração Z (16 a 24 anos) no mercado de trabalho, a taxa de adesão a sistemas de informação poderá apresentar substancial crescimento, haja vista a familiaridade deste público com tais tecnologias.

Destarte, o sistema será disponibilizado de forma gratuita para aumentar a competitividade dos jovens advogados, focando assim na usabilidade e capacidade de adaptação aos diferentes aparelhos, e capitalizando através de propagandas a serem exibidas nas telas. O uso de propagandas já se tornou uma prática comum em sistemas gratuitos.

2.1 Funcionalidades

Fazendo uso da experiência prévia como advogado de um dos autores deste trabalho, com a ajuda de conversas com colegas de profissão, jovens e veteranos, foi possível realizar o levantamento das funcionalidades que seriam as basilares para o funcionamento do sistema.

2.1.1 Cadastro de Pessoas.

O sistema deve ser capaz de realizar o cadastro de pessoas de interesse, sejam elas advogados parceiros, clientes, prestadores de serviços, dentre outros. Cada pessoa cadastrada deve ter seu conjunto de receitas e despesas, com a data para adimplemento das respectivas, permitindo ao usuário visualizar seus credores e devedores.

2.1.2 Cadastro de Processos.

Deve ser possível realizar o cadastro de processos com suas informações básicas (número, partes, tipo e tribunal), com sua respectiva data de início e final, vinculando ao mesmo as pessoas de interesse (partes, parceiros e prestadores de serviço) com suas respectivas despesas e receitas.

2.1.3 Cadastro de Recibos.

Por fim, o sistema deve permitir realizar a visualização do fluxo de caixa considerando os parâmetros mensal e anual, permitindo ao usuário realizar uma análise de seu crescimento e das perspectivas de rendimento futuro, auxiliando na organização de despesas, investimentos e percepção de crescimento.

2.2 Arquitetura

A arquitetura do sistema segue o padrão MVC (*Model-View-Controller*) com as respectivas camadas de *frontend*, *backend* e banco de dados, implementando o padrão mencionado e realizando comunicação entre si, conforme ilustrado na Figura 1.

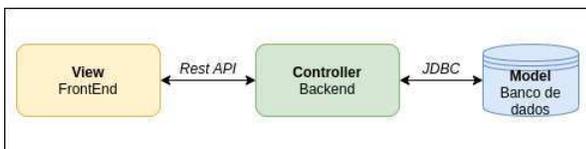


Figura 1: Arquitetura do sistema seguindo o padrão MVC [7].

2.2.1 FrontEnd.

A interface do usuário foi implementada em AngularJS [12] que é um *framework* voltado ao desenvolvimento de sistemas de informação online para a linguagem de programação JavaScript, fazendo uso dos padrões MVC e *Single Page Application*. O AngularJS foi desenvolvido no ano de 2009 com a finalidade de transformar a criação de sites de sistemas em uma atividade mais simples e focada em decisões de design [13].

Desta forma, a comunicação entre o cliente e o servidor passou a ser limitada a troca de informações através de requisições que obtém como resposta arquivos JSON (JavaScript Object Notation) ao invés de trocar todo o HTML da página [15], conforme Figura 2.

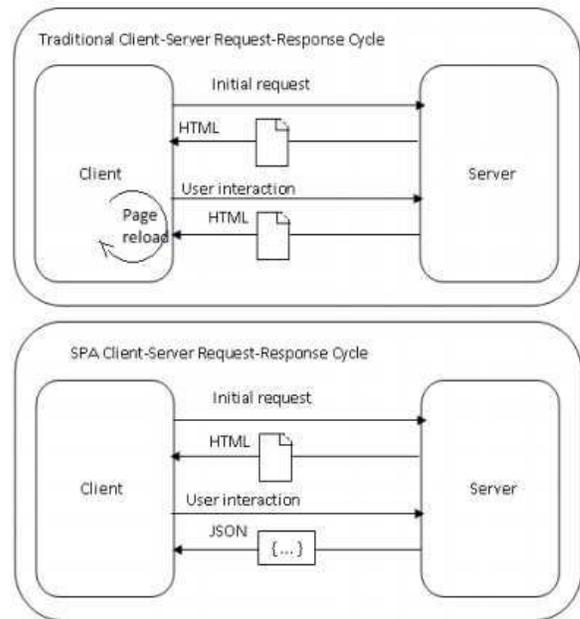


Figura 2: Comunicação entre cliente-servidor.[15]

Noutra ponta, para tornar a interface da aplicação mais amigável, foi utilizado o *framework* de CSS Bootstrap 3, criado em 2010 e lançado em agosto de 2011 como um *framework* de código aberto [3] e que foi amplamente utilizado no desenvolvimento do Twitter.

O código do *frontend* foi estruturado seguindo o padrão de estilo específico [10], e devido ao número de arquivos necessários ao desenvolvimento da aplicação, a Figura 3 expõe a árvore de diretórios com omissão dos arquivos de código javascript contidos nos diretórios do controller, do service e nos subdiretórios da lib, por não comprometerem o entendimento da arquitetura.

2.2.2 BackEnd.

O servidor foi desenvolvido utilizando o paradigma de orientação a objetos na linguagem Java 8, fazendo uso do *framework* Spring-Boot [6]. Assim, o *backend* foi utilizado no sistema como o intermediário entre as informações que devem ser armazenadas no banco de dados e aquelas que são repassadas para a interface com o usuário, sendo o desenvolvimento possível graças à simplicidade de implementação de um serviço web utilizando o *framework*.

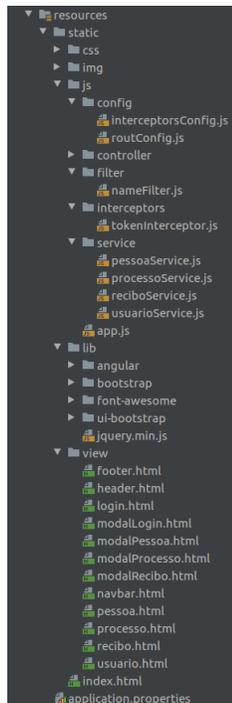


Figura 3: Organização estrutural dos diretórios do frontend

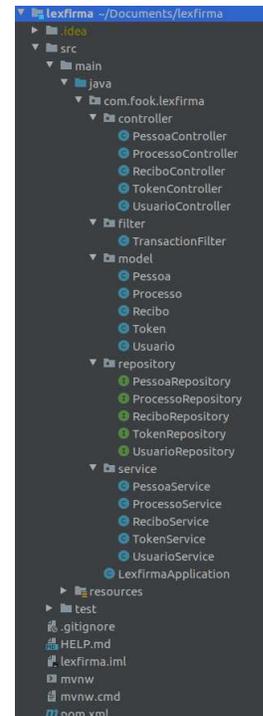


Figura 4: Diretórios do backend seguindo padrão MVC

Para a comunicação com o *frontend* foram criadas funções de troca de informações que seguiram o padrão REST (Representational State Transfer) API (Application Interface) [16], desta forma permitindo um design moderno e simplificado da comunicação entre o cliente e o servidor.

Logo, o desenvolvimento que seguiu o já mencionado padrão MVC, teve sua estrutura de diretórios organizada no mesmo padrão aplicado ao frontend conforme Figura 4.

2.2.3 Banco de Dados.

As informações do sistema que devem ser persistidas na base dados podem ser projetadas para se adequar ao modelo objeto-relacional de banco de dados. Portanto, optou-se por utilizar o PostgreSQL [1] como sistema gerenciador por ser de código aberto e ter uma integração com o SpringBoot através do JDBC (Java Database Connectivity) [17].

Assim, com base nos requisitos levantados foi definido o modelo de dados a ser implementado utilizando as tecnologias mencionadas anteriormente, conforme ilustrado na Figura 5.

2.3 Sistema

Para poder atender os requisitos, o sistema foi desenvolvido tendo em mente a usabilidade, sendo assim seccionado em quatro módulos: Usuário, Pessoas, Processos e Recibos.

2.3.1 Usuário.

O módulo do usuário retém as responsabilidades de realização da entrada no sistema, e retenção de informações básicas e pessoais do jovem advogado, para autenticação e autorização.

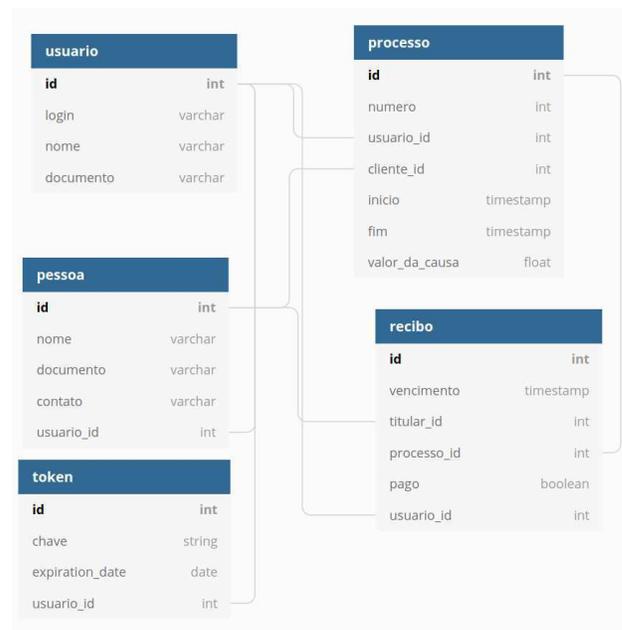


Figura 5: Modelo de dados.

Estas responsabilidades foram separadas no fluxo de cadastro, login e edição do usuário. O cadastro é o momento em que são enviadas informações para o registro na base de dados e criação da referência no sistema, conforme ilustrado na Figura 6.

Figura 6: Cadastro de Usuário.

O login é o momento em que o usuário realiza o acesso ao sistema, através de seu email (Figura 7), permitindo a recuperação das informações necessárias ao uso das funcionalidades disponíveis no site.

Figura 7: Login no sistema.

Por fim, na edição de usuário (Figura 8), este mantém a capacidade de realizar a gestão de suas informações, podendo atualizar seus dados cadastrais conforme lhe aprouver. Desta forma, fica assegurado o fluxo de acesso e edição do sistema a ser realizado pelo usuário.

2.3.2 Pessoas.

O módulo de pessoas é a parte do sistema onde os devedores e credores do usuário são exibidos e poderão ser adicionados e gerenciados. Assim, têm-se botões para adição, exclusão e edição, sendo o primeiro geral, e os demais específicos para cada pessoa já cadastrada no sistema (Figura 9). A tela ainda apresenta a opção de realizar um filtro na lista através da equiparação entre a informação digitada na caixa de texto e a contida nas colunas de cada linha da tabela exibida.

A adição e edição de pessoas (Figura 10) é realizada em um modal que é exibido quando os respectivos botões são pressionados pelo usuário.

2.3.3 Processos.

O módulo de processos é o local onde as informações relativas aos processos podem ser exibidas e gerenciadas através de uma tela de listagem (Figura 11).

Após o clique nos respectivos botões, é possível realizar a criação ou edição de um processo com o devido vínculo entre o processo e o cliente, que é uma pessoa já previamente cadastrada no sistema, sendo a principal referência do advogado (Figura 12).

2.3.4 Recibos.

Por fim, o módulo de recibos é o responsável por armazenar e exibir as informações relativas a pagamentos e recebimentos. Para diferenciar entre ambos, utilizam-se os sinais de mais e menos.

Desta forma, seguindo o padrão de visualização do sistema, foi criada uma tela para a listagem de todos os recibos cadastrados no sistema (Figura 13), e onde também estão os botões que permitem a edição e adição de novos recibos.

Então, durante a edição/adição de recibos é possível realizar o vínculo com o processo e a pessoa a quem se destina através do uso de seletores conforme Figura 14.

2.4 Ambiente de Implantação

O sistema foi disponibilizado na internet através do uso do Heroku[19] que é um serviço de PAAS (*Platform as a Service*) gratuito para a hospedagem de um único sistema e com a funcionalidade de permitir integração com o Github[14], possibilitando a atualização da versão online do sistema sempre que houver uma nova versão da *branch* selecionada como referência do repositório.

3 EXPERIÊNCIA

3.1 Processo de Desenvolvimento

O processo de desenvolvimento foi iniciado pela coleta dos requisitos realizada de maneira informal e através de conversas com usuários em potencial, nas quais foi avaliado quais funcionalidades seriam consideradas como o fundamentais.

Após a primeira etapa, buscou-se realizar uma avaliação das tecnologias disponíveis para acelerar o processo de desenvolvimento sem sacrificar a capacidade de realizar futura manutenção e incremento do sistema. Neste ponto, as tecnologias do *frontend* foram selecionadas com base no que é mais comum no mercado, mesmo que demandasse aprendizado, que foi o caso. Por outro lado, as tecnologias de *backend* e banco de dados, foram escolhidas por serem do domínio de conhecimento do primeiro autor deste artigo, assim como por proporcionarem um ambiente ágil e estável de desenvolvimento.

Uma vez estabelecidas as tecnologias, o desenvolvimento seguiu um padrão de seleção da funcionalidade, implementação do *frontend* realizando *mock* do *backend*, que seria desenvolvido quando a tela já estivesse pronta.

A evolução do desenvolvimento era armazenada em uma *branch* no github destinada ao desenvolvimento (dev), sendo realizado um *pull request* quando a funcionalidade fosse considerada acabada.

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
MINISTÉRIO DA EDUCAÇÃO

Usuário PESSOAS Processos Recibos Sair

Usuário

Nome: Carlos dos Santos

Documento: 01297812356

Email: carlos@gmail.com

Senha: *****

Alessandro Lia Fook Santos ©2021

Figura 8: Edição de Usuário.

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
MINISTÉRIO DA EDUCAÇÃO

Usuário PESSOAS Processos Recibos Sair

Pessoas

Quem você está buscando?

Nome	Documento	Contato
João da Silva	1910633 SSP/PB	988181597
Maria Do Carmo	259.985.743-81	33564589
Josué Ferreira.	23934 oab/pb	21 988753364

Alessandro Lia Fook Santos ©2021

Figura 9: Listagem de Pessoas.

Mas quando alguma iteração levasse à necessidade de reparo, este era realizado em dev e depois unificado à *branch master*.

Essa divisão de *branch* foi realizada pois o sistema foi disponibilizado em um ambiente online para permitir um teste de uso em um ambiente próximo do que seria o ambiente produtivo.

3.2 Desafios

O primeiro grande desafio do desenvolvimento foi o tempo para finalizar o sistema. Devido ao período de pandemia, foi necessário reduzir o sistema a uma prova de conceito focada no Mínimo Produto Viável (MVP).

Outro desafio foi realizar o aprendizado do AngularJS em tempo de desenvolvimento, o que teve um custo mas foi muito gratificante.

Por estar aprendendo as tecnologias em tempo de desenvolvimento, algumas partes demandaram o uso de bibliotecas que expandissem as capacidades, como por exemplo, o uso de modal e o acesso ao *storage* do usuário, prática aquela é que comum e acelera o processo.

3.3 Avaliação

Foi solicitado ao advogado que auxiliou no levantamento dos requisitos que promovesse uma avaliação do sistema.

Após navegar pelo sistema o jovem advogado relatou que teve uma boa experiência de utilização, porém informou que a exibição da coluna com as informações sobre o estado de pagamento poderia conter uma palavra significativa para indicar o estado do recibo.



Figura 10: Edição e Adição de Pessoas.



Figura 11: Listagem de Processos.

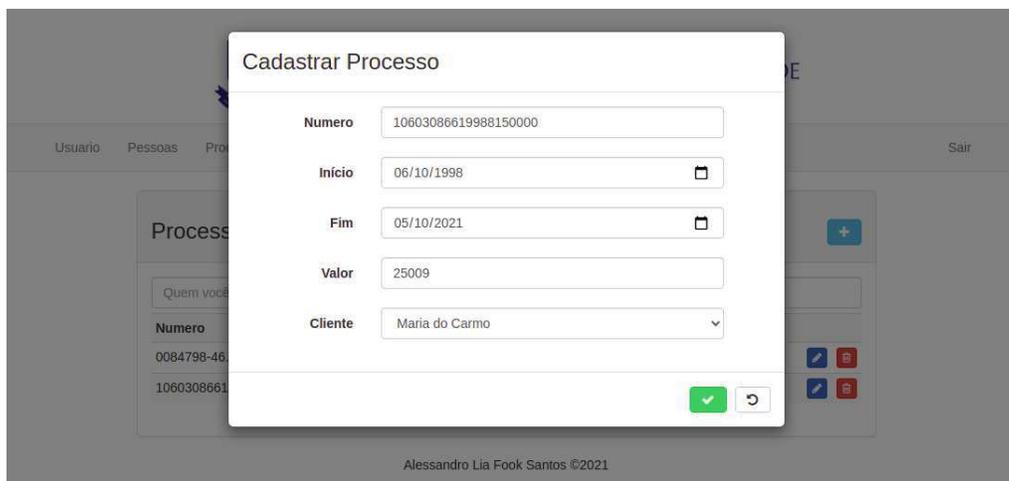


Figura 12: Edição e Adição de Processos.



Figura 13: Listagem de Recibos.

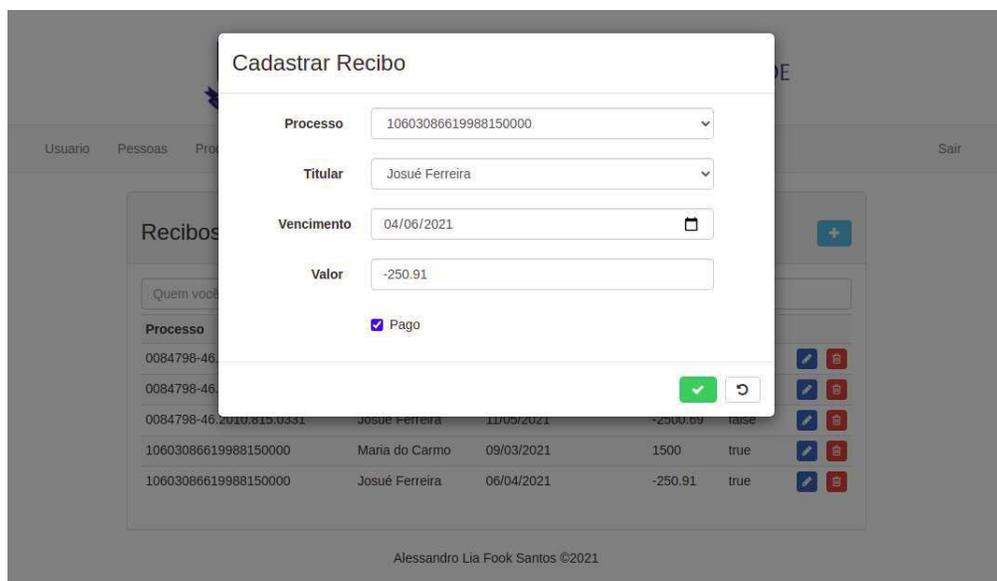


Figura 14: Edição e Adição de Recibos.

Assim foi promovida a mudança solicitada, por se tratar de um ajuste rápido.

Por outro lado, o avaliador considerou o sistema uma solução simples e funcional, que atenderia a sua demanda inicial mas gostaria bastante do acréscimo de algumas funcionalidades, dentre elas:

- Possibilidade de filtrar as colunas de forma individual;
- Uma nova página com relatório financeiros do processo;
- Uma nova página com relatório do fluxo de caixa podendo ajustar a visualização para representar o mês ou o ano a ser selecionado.

4 TRABALHOS FUTUROS

O objetivo deste trabalho foi desenvolver uma ferramenta que permitisse a um jovem advogado armazenar e consultar suas informações financeiras, que estivessem relacionadas a contratos.

Ademais, por se tratar de um sistema de informação, sempre é possível adicionar novas funcionalidades que teriam impacto na usabilidade do sistema, como mensagens indicando o resultado das requisições, formas de visualização dos dados que se mostrassem mais dinâmicas, como gráficos e relatórios, conforme indicado na avaliação feita.

Assim, como todo sistema computacional, por mais pronto que se encontre, sempre haverá espaço para incremento e melhoria.

REFERÊNCIAS

- [1] 1996. PostgreSQL: The World's Most Advanced Open Source Relational Database. <https://www.postgresql.org/>. Accessed: 2021-04-25.
- [2] 2000. DIRETRIZES CURRICULARES DO CURSO DE DIREITO Ministério da Educação do Brasil. http://portal.mec.gov.br/sesu/arquivos/pdf/dir_dire.pdf. Accessed: 2021-04-24.
- [3] 2021. Bootstrap. <https://getbootstrap.com/docs/4.1/about/overview/>. Accessed: 2021-04-25.
- [4] 2021. Funcionalidades Pje. <https://www.pje.jus.br/wiki/index.php/Funcionalidades>. Accessed: 2021-04-24.
- [5] 2021. PJE Processo Judicial Eletrônico. <https://www.cnj.jus.br/programas-e-acoes/processo-judicial-eletronico-pje/historico/>. Accessed: 2021-04-24.
- [6] 2021. Spring. <https://spring.io/why-spring>. Accessed: 2021-04-25.
- [7] Gaudenz Alder. 2016. Draw.io - Diagrams.net. <https://app.diagrams.net/>. Accessed: 2021-04-25.
- [8] Zínia Baeta. 2019. Advogados enfrentam mercado em queda e alta concorrência. <https://valor.globo.com/legislacao/noticia/2019/10/25/advogados-enfrentam-mercado-em-queda-e-alta-concorrenca.shtml>. Accessed: 2021-04-25.
- [9] Júnior Borneli. 2019. 15 Maracanãs lotados. O Brasil tem advogados demais? <https://www.startse.com/noticia/mercado/15-maracanas-lotados-o-brasil-tem-advogados-demais-lawtech>. Accessed: 2021-05-12.
- [10] Rodrigo Branas. 2014. *AngularJS Essentials*. Packt Publishing Ltd.
- [11] Brasil. 2006. Lei nº 11.419, de 19 de dezembro de 2006. *Diário Oficial [da] República Federativa do Brasil* (2006). http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2006/lei/11419.htm
- [12] Google. 2010. AngularJS. <https://angularjs.org/>. Accessed: 2021-04-25.
- [13] Brad Green and Shyam Seshadri. 2013. *AngularJS*. "O'Reilly Media, Inc."
- [14] Github Inc. 2021. Github. <https://github.com/>. Accessed: 2021-05-13.
- [15] Madhuri A Jadhav, Balkrishna R Sawant, and Anushree Deshmukh. 2015. Single page application using angularjs. *International Journal of Computer Science and Information Technologies* 6, 3 (2015), 2876–2879.
- [16] Mark Masse. 2011. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. "O'Reilly Media, Inc."
- [17] Oracle. 1993. Java JDBC API. <https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/>. Accessed: 2021-04-25.
- [18] Marcos Aurélio Silva. 2019. 97% dos escritórios de sucesso têm software de gestão para advogados. <https://www.migalhas.com.br/depeso/299983/97--dos-escritorios-de-sucesso-tem-software-de-gestao-para-advogados>. Accessed: 2021-04-24.
- [19] Holistics Software. 2021. Heroku. <https://www.heroku.com/about>. Accessed: 2021-04-25.