

Automatização do Acompanhamento do Andamento de Processos e Projetos de Lei

Ary Andrade¹, Felipe Schreiber¹, Gabriel Oliveira¹,
Henrique Pan¹, Paulo Mattos¹, Victor Raposo¹

¹ Universidade Federal do Rio de Janeiro - UFRJ
Rio de Janeiro, Brazil

[{aryneto, schreiber.felipe, gabriel.oliveira} @poli.ufrj.br

[{henriquepan, paulo.mattos, victor.ravaglia} @poli.ufrj.br

Abstract. *In a law firm, one of the weekly workloads consists of monitoring law-suits, checking for updates. In general, due to the lack of automation involved in the activity, this is done inefficiently, consulting various court sites manually. This document describes a web platform that aims to speed up and increase efficiency in the monitoring process, automating most of the workload.*

Resumo. *Em escritórios de advocacia, uma das cargas de trabalho envolve o acompanhamento de processos e projetos de lei. Em geral, devido a falta de automatização envolvida na atividade, isso é feito de forma ineficiente, consultando diversos sites de tribunais manualmente. Neste documento, é descrita uma plataforma web que visa acelerar e aumentar a eficiência no acompanhamento de processos, automatizando a maior parte da atividade.*

1. Introdução

O acompanhamento de processos e projetos de lei em firmas de advocacia é uma tarefa que em geral é feita manualmente e pode levar horas para ser concluída. O objetivo dessa atividade, é condensar as informações sobre todas as atualizações que ocorreram em um determinado período de tempo, possibilitando a elaboração de relatórios.

A aplicação descrita neste documento se refere ao Menor Produto Viável (MVP¹) para a solução final. Ela consiste de uma aplicação web que visa automatizar parte do processo, reduzindo a carga horária e o pessoal necessários para o acompanhamento do andamento de processos. Segue o link do repositório da aplicação: Github².

Este documento está dividido como se segue. A Seção 2 descreve todo o processo de acompanhamento de atualizações usualmente adotado. Na Seção 3, a plataforma proposta bem como as funções presentes no MVP são apresentadas. A Seção 4 detalha as tecnologias, metodologias e detalhes técnicos associados ao desenvolvimento da solução. Finalmente, na Seção 5, conclusões e propostas para a aplicação finalizada são discutidas.

2. Definição do Problema

No ramo jurídico, parte da carga de trabalho envolve o acompanhamento de processos, em busca de atualizações que possam indicar possíveis resoluções ou empecilhos no andamento do mesmo. Esse acompanhamento é feito visitando sites de tribunais regionais,

¹<https://github.com/netoary/programacao-avancada/wiki>

²<https://github.com/netoary/programacao-avancada>

estaduais e federais. Um advogado pode estar encarregado de mais de um cliente simultaneamente. Cada cliente pode ter um ou mais processos associados a diferentes tribunais.

A carga de trabalho convencional envolve a construção de tabelas de acompanhamento legislativo (TAL), semelhantes a apresentada na Figura 1. Nelas, são contidas as últimas atualizações de cada processo ou projeto de lei que está sendo acompanhado. Para que a tabela seja atualizada, para cada linha da tabela, o encarregado deve visitar o site do tribunal associado, se identificar ou se cadastrar quando necessário, e ler os relatórios fornecidos pelo tribunal na procura de atualizações no andamento. O objetivo final das TAL é resumir o progresso dos processos e projetos de lei para facilitar a elaboração de relatórios. As TAL podem conter dezenas de entradas, e sua atualização manual pode durar horas.

Nem todos os sites de tribunal tem a mesma interface, dificultando a navegação. Alguns tribunais oferecem a possibilidade de atualizações automáticas através de e-mails. No entanto, em geral, as repostas não são padronizadas, podendo conter todo o progresso do processo ou projeto de lei, desde seu início, em vez de só a última atualização. Além disso, há a necessidade de condensar informações para que as TAL sejam preenchidas. Desta forma, os e-mails são utilizados apenas como aviso de que há uma nova atualização, e, conseqüentemente, os sites de tribunais ainda serem visitados independentemente.

Número	Assunto	Andamento mais recente	Observação
Processo nº 0059	Questiona nova lei de IPTU/RJ	21.10.2020 - Remessa do escrivão para 3ª VP (Divisão de Comunicação Externa e Gestão) 21.10.2020 - A desembargadora proferiu a decisão de não reatuação 19.10.2020 - Conclusão ao 3º Vice-Presidente para reatuação.	Publicada decisão proferida pelo relator (19.07.2020.2019); o município intimado para exarar par juízo de admissibilidade; 21.11.2019 foi proferida para a 3ª VP. Em 18.02.2020 Remessa do Escrivão/Dit
ADIN nº 5	ISS (LC)	23.10.2020 - Ocorreu a juntada de procuração/substabelecimento 15.10.2020 - A Prefeitura de São Paulo apresentou seus memoriais / 08.10.2020 - Conclusos ao relator após juntada petição da CONFEDERAÇÃO NACIONAL	Publicação no Dje das de Campo (20.02.2019); jur calendário de julgamento; Inclua-se em pauta, Julg;
ADPF 4	ISS (LC)	23.10.2020 - Ocorreu a juntada de procuração/substabelecimento 08.10.2020 - Conclusos ao relator após juntada petição da CONFEDERAÇÃO NACIONAL DE SAÚDE, HOSPITAIS E ESTABELECIMENTOS E SERVIÇOS	Juntada de mandado de

Figura 1. Exemplo de uma Tabela de Acompanhamento Legislativo. As linhas representam processos ou projetos de leis sendo acompanhados. O objetivo é preencher a coluna "Observação" com uma compilação de andamentos.

3. Plataforma Proposta

Seguindo os moldes estabelecidos pelos requisitos da disciplina, a solução proposta foi uma *Single Page Application* (SPA, ou em português, aplicação de página única). O objetivo da aplicação é automatizar a extração da informação de processos e projetos de lei.

Para o MVP, haverá um diretório com exemplos de dados de retorno da API do PJe³, detalhada na Subseção 4.1. A Figura 2 apresenta a página principal da aplicação, contendo um painel de controle. Nele esta contida uma tabela que concentra

³<http://www.pje.jus.br/wiki/>

as informações úteis de processos e projetos de lei cadastrados. Os processos são agrupados pelo seu respectivo número de identificação, de modo que todas as atualizações feitas desde o cadastro ficam associadas a ele.



The screenshot shows a web application interface titled "Overview". It features a table with columns for "Reclamante/Requerente", "Reclamadas", "ADV", "Vara", "Processo", and "Tags". There are two rows of data. The first row has a magnifying glass icon in the first column, followed by "FEDERACAO DOS TRABAL...", "ARMANDO CAPRIO...", "ANTONINO MOURA ...", "2ª VARA CÍVEL - VARA...", "78625720168110050", and two red tags: "Grandes causas" and "Individual". The second row has a magnifying glass icon, "AMANDA SILVA LIMA MAZI...", "ESTADO DE MATO ...", "PROCURADORIA G...", "VARA ESP. DE EXECU...", and "1056001182019811...". A "+" button is located in the top right corner of the table area.

	Reclamante/Requerente	Reclamadas	ADV	Vara	Processo	Tags	
Q	FEDERACAO DOS TRABAL...	ARMANDO CAPRIO...	ANTONINO MOURA ...	2ª VARA CÍVEL - VARA...	78625720168110050	Grandes causas Individual	
Q	AMANDA SILVA LIMA MAZI...	ESTADO DE MATO ...	PROCURADORIA G...	VARA ESP. DE EXECU...	1056001182019811...		

Figura 2. Página principal da aplicação, onde um painel de controle condensa as informações de processos e projetos de lei cadastrados.

Para adicionar um novo processo, o usuário deve clicar no botão "+" no canto superior direito, onde será oferecida a oportunidade de inserir o número do processo desejado. O programa irá buscar na lista de processos disponíveis à procura de uma correspondência. E finalmente, o painel será atualizado. Ao clicar no ícone de lupa, o usuário terá acesso ao histórico do andamento do processo (Figura 3).



Figura 3. Histórico contendo as atualizações do processo.

Para o MVP, as seguintes funcionalidades principais estão disponíveis:

- Cadastro de processos e projetos de lei. No MVP, isso será feito de forma offline.
- Consulta de estado (movimentação) de processos cadastrados.
- Tabela com resumo dos processos cadastrados e possibilidade de filtragem.
- Busca por processos cadastrados.

4. Implementação

Para construir a aplicação nos baseamos na pilha MERN -MongoDB⁴, Express⁵, React⁶ e Node⁷. Essa pilha é essencialmente baseada na linguagem Javascript. Além disso, o projeto está construído dentro de um container Docker, para facilitar o seu *deploy*. A seguir, dividiremos a implementação em 4 subseções: Dados, dependências, componentes e rotas. Em Dados, descrevemos a comunicação com a fonte dos dados. Já em Dependências, explicitamos as bibliotecas utilizadas no desenvolvimento dos Componentes, que por sua vez descrevem as funcionalidades da aplicação. Por fim, em Rotas, descrevemos a navegação do usuário.

4.1. Dados

Os dados sobre processos consumidos pela aplicação são provenientes da plataforma PJe, que é a solução construída pelo Conselho Nacional de Justiça (CNJ) para unificar os sistemas dos diversos tribunais de justiça espalhados pelo país. O PJe fornece uma API do tipo SOAP, através da qual é possível realizar consultas processuais a partir do número do processo. Entretanto, para fazer pedidos aos endpoints desta API, é necessário possuir credenciais cadastradas na instância do PJe do tribunal desejado. Portanto, para o MVP realizamos consultas pontuais utilizando as credenciais de um advogado conhecido e armazenamos os XMLs no repositório.

4.2. Dependências

Optamos por utilizar o material-ui⁸, que é um biblioteca de componentes web construída em React. Abaixo, listamos os principais componentes utilizados e suas funções:

- DataGrid - Este componente fornece essencialmente uma apresentação de dados no formato de uma tabela interativa e, por isso, foi utilizado na implementação do componente de DashBoard, apresentado na Subseção 4.3. Além da visualização tabular, outras funcionalidades disponíveis foram incorporadas ao projeto, como a possibilidade de filtrar o conteúdo a partir de cada coluna e ocultar colunas.
- Timeline - Fornece um conjunto de ferramentas para exibição de eventos em ordem cronológica. Foi utilizado no componente History, apresentado na Subseção 4.3, e ilustrado na Figura 3.
- MuiAlert - Com esse componente é possível personalizar os alertas mostrados aos usuários.

4.3. Componentes

Basicamente temos 3 componentes implementados:

- RegisterProcess - Classe responsável pelo cadastro de processo. Ela faz uma requisição através de um POST a um endpoint no back-end. O resultado é as informações do processo em JSON.
- History - Classe que explicita o histórico de um determinado processo. Contém os atributos `dateTime`, `message` e `documentId`.
- Dashboard - Classe principal da aplicação, onde a tabela e suas funções são controladas.

⁴<https://www.mongodb.com/>

⁵<https://expressjs.com/>

⁶<https://pt-br.reactjs.org/>

⁷<https://nodejs.org/en/>

⁸<https://material-ui.com>

4.4. Rotas

- Overview - Essa classe/rota faz duas tarefas: Primeiramente pega o arquivo xml que é devolvido da API do tribunal. Em segundo lugar efetua a análise dos dados e os retorna para que o Dashboard possa exibi-los.

5. Conclusão

Devido a falta de automatização na busca por atualizações de processos e projetos de lei, muito do tempo de trabalho de advogados é utilizado de forma ineficiente. Neste trabalho, propomos uma solução em forma de uma página web que tem a expectativa de automatizar a maior parte dessa carga de trabalho. No MVP implementou-se boa parte da interface com o usuário provendo: cadastro de processos, painel de consulta (em um arquivo simulado) e painel de controle dos processos.

Para a próxima entrega pretendemos implementar: busca de processos na API do tribunal, um painel personalizado para cada usuário, cadastro dos usuários, agrupamento dos processos por etiquetas e exportação para csv.