



PUC

**On the Interaction between Software Engineers
and Data Scientists building ML-enabled
Systems**

Gabriel de Andrade Busquim

Marcos Kalinowski

Departamento de Informática

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900
RIO DE JANEIRO - BRASIL**

Sumário

1	Introdução	1
2	Descrição do Software	1
3	Especificação de Requisitos	2
4	Arquitetura	3
5	Modelo Funcional	4
6	Manual de Utilização	4
7	Cenários de Sucesso do Software	4
7.1	Cenário 1: Criação de uma Demanda	4
7.2	Cenário 2: Visualizar dados de uma Demanda	4
8	Cenários de Dificuldade do Software	4
8.1	Cenário 1: Criação de uma Demanda	4
8.2	Cenário 2: Upload do Termo de Acordo	4

1 Introdução

Empresas têm cada vez mais se esforçado para adquirir e analisar dados visando tomar decisões mais assertivas. Isso favoreceu o desenvolvimento de componentes de Machine Learning (ML), que são muitas vezes integrados em sistemas já existentes como uma funcionalidade adicional. Neste cenário, diferentes atores como cientistas de dados, engenheiros de software e designers precisam colaborar constantemente.

Nossa pesquisa visa investigar como se dá a interação entre cientistas de dados e engenheiros de software durante o desenvolvimento de um sistema de ML e sua posterior integração com o resto do sistema. Com isso, buscamos entender como responsabilidades são divididas, que dificuldades os atores têm durante sua colaboração e como ela pode ser melhorada.

Para isso, realizamos um estudo de caso com uma equipe construindo um sistema para a indústria. O sistema possui diversos módulos, dentre eles um módulo de ML. Foram feitas entrevistas com cientistas de dados e engenheiros de software da equipe, cujas transcrições foram posteriormente analisadas. Neste documento, descrevemos a aplicação construída pelo time que foi entrevistado. Ela foi desenvolvida dentro do Instituto Tecgraf em parceria com o Tribunal de Justiça do Estado do Rio de Janeiro (TJRJ).

2 Descrição do Software

O +Acordo é uma plataforma pré-processual de Resolução de Disputas Online (em inglês, Online Dispute Resolution, ou ODR) criada para facilitar a resolução de demandas recorrentes de baixa complexidade jurídica. Seu objetivo é resolver os problemas de seus usuários de forma ágil e acessível, minimizando custos. Ela também visa reduzir a burocracia inerente de processos jurídicos, uma vez que permite resolver demandas sem que seja necessário comparecer em um tribunal. Para isso, o software utiliza técnicas de inteligência artificial, que avaliam a possibilidade de oferecimento de uma proposta de acordo para cada caso.

O sistema será disponibilizado para advogados cadastrados no Processo Judicial eletrônico (PJe), sistema criado pelo Conselho Nacional de Justiça (CNJ) visando atender às necessidades do Poder Judiciário brasileiro. Para se cadastrar na plataforma, o advogado deverá informar suas credenciais no PJe e preencher dados adicionais, como seu e-mail e endereço. Após criar sua conta, ele poderá submeter demandas para a plataforma.

A avaliação dos dados da demanda cadastrada na plataforma e sua comparação com processos anteriores já resolvidos determina se há possibilidade de oferecimento de uma proposta de acordo. Caso a resposta seja positiva, a melhor proposta possível é apresentada ao usuário. Caso não seja possível gerar uma proposta, é gerado um relatório com uma justificativa para tal e o processo é encerrado na plataforma.

Ao consultar uma proposta de acordo, o advogado pode tanto aceitá-la quanto recusá-la. Nas situações em que uma proposta é aceita, é preciso enviá-la para ser homologada por um juiz do Centro Judiciário de Métodos Consensuais de Solução de Disputas (CEJUSC), oficializando assim o acordo entre o autor da demanda e a empresa demandada. Essa homologação também é feita através plataforma, onde o advogado deve fazer o upload do termo de acordo assinado digitalmente, que é então transferido para o CEJUSC. A partir do momento da transferência, a demanda recebe um número de processo e o acompanhamento

de seu andamento deve ocorrer dentro do ambiente do PJe.

Ao não aceitar uma proposta de acordo, o advogado deverá indicar pelo menos uma justificativa para a recusa. O sistema então gerará um relatório contendo os dados da demanda que foi cadastrada e o motivo de recusa indicado pelo advogado. Neste caso, a demanda deverá ser tratada por fora da plataforma.

É esperado que o +Acordo tenha uma vida média elevada, visto que se trata de uma solução que será disponibilizada pelo TJRJ. A plataforma tratará inicialmente casos de baixa complexidade jurídica de empresas específicas. Entretanto, espera-se que o número de demandas resolvidas dentro do software cresça à medida que outras empresas demonstrem interesse na plataforma. Além disso, o sistema deverá ser constantemente evoluído de modo a atender novos requisitos definidos pelo Tribunal.

3 Especificação de Requisitos

De modo a ilustrar as funcionalidades da plataforma, esta seção mostra alguns requisitos funcionais e não-funcionais do software, definidos pelo time do +Acordo em conjunto com representantes do TJRJ.

Exemplos de requisitos funcionais:

1. O sistema deve permitir que advogados se cadastrem na plataforma utilizando suas credenciais no PJe.
2. O advogado deve ser capaz de visualizar seus dados cadastrais dentro da plataforma.
3. O advogado deve conseguir criar uma demanda e submetê-la para análise do sistema de IA.
4. O advogado deve ser capaz de visualizar todas as demandas que ele criou.
5. O advogado deve conseguir acessar a situação de uma demanda específica criada por ele.
6. Caso não seja possível gerar uma proposta de acordo, o sistema deve gerar um documento contendo o motivo.
7. Caso uma proposta de acordo seja oferecida, o advogado deve ser capaz de conseguir aceitá-la ou rejeitá-la.
8. Se o advogado aceitar uma proposta de acordo, o sistema deverá gerar um documento com os parâmetros do acordo.
9. O advogado deverá ser capaz de fazer upload do termo de acordo assinado para que ele seja homologado junto a um juiz do CEJUSC.
10. Se o advogado recusar a proposta de acordo, o sistema deverá gerar um documento contendo o motivo da recusa.

Exemplos de requisitos não-funcionais:

1. O advogado só pode visualizar as demandas que ele criou.

2. Uma proposta de acordo não pode mais ser aceita caso tenha se passado mais de um dia desde a sua criação.
3. O advogado não poderá criar duas demandas com os mesmos dados.
4. A plataforma deve ser compatível com todos os navegadores.
5. Após a submissão da demanda, o resultado da proposta de acordo deve ser exibido em até 10 segundos.

4 Arquitetura

O sistema +Acordo possui uma série de componentes. A imagem abaixo mostra uma visão simplificada da sua arquitetura. Nesta seção, descrevemos brevemente cada um dos componentes representados na imagem.

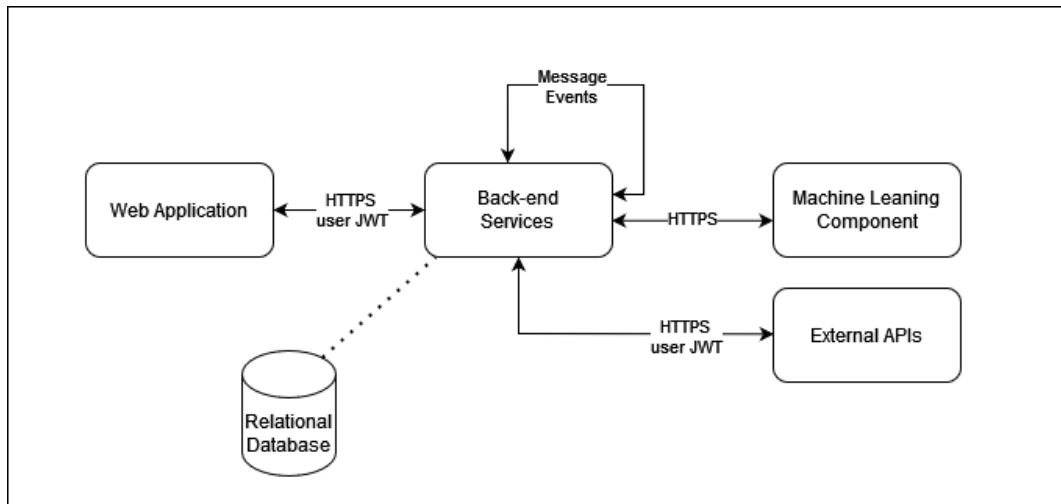


Figure 1: Arquitetura Simplificada do +Acordo

Os advogados têm acesso às funcionalidades do sistema por meio de uma aplicação web. Através dela, os advogados podem se cadastrar na plataforma, cadastrar demandas e visualizar as propostas de acordo sugeridas. A aplicação se comunica com os serviços back-end por meio de uma API REST. Cada chamada feita contém um token JWT, que identifica o usuário logado. Este token é validado no back-end, para que cada usuário tenha acesso somente às suas informações.

A arquitetura back-end do +Acordo é baseada em microsserviços, onde cada serviço é responsável por uma entidade do software (por exemplo, processar disputas, gerenciar acordos, gerar documentos, entre outros). Para prover as funcionalidades do sistema, cada componente se comunicam entre si de modo síncrono e assíncrono. A comunicação síncrona acontece também por meio de APIs REST, enquanto a comunicação assíncrona se dá via mensagens. Dessa forma, se estabeleceu também uma arquitetura orientada a eventos (em inglês, event-driven architecture, ou EDA).

Foi utilizado um banco de dados relacional para armazenar os dados de cada advogado cadastrado e cada demanda criada. Ele é consultado pelas APIs do sistema para exibir informações na aplicação web, assim como para prover as funcionalidades do software.

5 Modelo Funcional

Detalhar fluxo de criação de uma demanda dentro da plataforma?

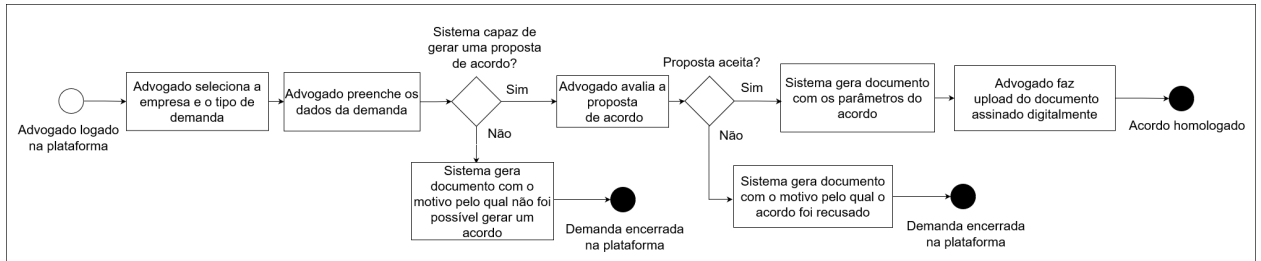


Figure 2: Fluxo de criação de uma demanda

6 Manual de Utilização

Incluir imagens da plataforma?

7 Cenários de Sucesso do Software

7.1 Cenário 1: Criação de uma Demanda

7.2 Cenário 2: Visualizar dados de uma Demanda

8 Cenários de Dificuldade do Software

8.1 Cenário 1: Criação de uma Demanda

8.2 Cenário 2: Upload do Termo de Acordo