



IFPB- Instituto Federal de Educação, Ciência e Tecnologia da Paraíba  
Campus Campina Grande  
Curso de Bacharelado em Engenharia de Computação

**SISPRO: Um aplicativo de atendimento ao cliente para consulta processual**

EDUARDO BRUNO DE ALMEIDA DONATO  
IGOR HENRIQUE MARTINS ARAÚJO  
JULIO CESAR URBANO VASCONCELOS FILHO

Orientador: Victor André Pinho de Oliveira

Campina Grande-PB  
Dezembro de 2022



IFPB- Instituto Federal de Educação, Ciência e Tecnologia da Paraíba  
Campus Campina Grande  
Curso de Bacharelado em Engenharia de Computação

## **SISPRO: Um aplicativo de atendimento ao cliente para consulta processual**

EDUARDO BRUNO DE ALMEIDA DONATO  
IGOR HENRIQUE MARTINS ARAÚJO  
JULIO CESAR URBANO VASCONCELOS FILHO

Monografia apresentada à Coordenação do Curso de Engenharia da Computação do IFPB - Campus Campina Grande, como requisito para conclusão do curso de Engenharia da Computação.

Orientador: Victor André Pinho de Oliveira

Campina Grande-PB  
Dezembro de 2022

D677s Donato, Eduardo Bruno de Almeida.

SISPRO: um aplicativo de atendimento ao cliente para consulta processual. - Campina Grande, 2022.

50 f. : il.

Trabalho de Conclusão de Curso (Curso de Graduação em Engenharia de Computação) - Instituto Federal da Paraíba, 2022.

Orientador: Prof. Me. Victor André Pinho de Oliveira.

1.Aplicação móvel 2. Consulta processual 3.SISPRO I. Araújo, Igor Henrique Martins II. Vasconcelos Filho, Júlio César Urbano. III. Título.

CDU 004.43

# SISPRO: Um aplicativo de atendimento ao cliente para consulta processual

EDUARDO BRUNO DE ALMEIDA DONATO  
IGOR HENRIQUE MARTINS ARAÚJO  
JULIO CESAR URBANO VASCONCELOS FILHO

---

Victor André Pinho de Oliveira  
Orientador

---

Ianna Maria Sodre F de Sousa  
Membro da Banca

---

Petrônio Carlos Bezerra  
Membro da Banca

Campina Grande-PB,  
DEZEMBRO/2022

## **AGRADECIMENTOS**

Eu, Eduardo Bruno de Almeida Donato, queria agradecer a Deus por me proporcionar a vida, por cada desafio que me fez crescer, por cada oportunidade de seguir meus sonhos e por todas as conquistas que ainda estão por vir. A meus pais, Sérgio e Sivanira Donato e minha irmã Laryssa Donato, por toda dedicação e amor. Ao meu orientador, Prof. Victor André Pinho de Oliveira, quero agradecer por toda dedicação e empenho que o professor teve durante esta trajetória. Aos meus colegas de turma, agradeço pela convivência diária e por todos os momentos compartilhados.

Eu, Igor Henrique Martins Araújo, nesta etapa final me sinto muito grato por todo caminho percorrido, todos os conhecimentos e oportunidades que surgiram ao decorrer do curso. Quero agradecer primeiro a Deus, a meu pai Vladimir Araújo e minha mãe Manuela Martins por todo o suporte e atenção, mesmo de longe conseguiram me manter na linha e me ajudaram com absolutamente tudo que precisei, meus maiores exemplos. As minhas irmãs Gabriela e Graziela que serviram de força pra mim pra eu me tornar uma pessoa melhor e ser um exemplo para elas, a minha namorada Ingrid por toda ajuda e suporte emocional para conseguir chegar aqui até hoje de cabeça erguida. Aos meus avós e tios aqui em Campina Grande pelo acolhimento como se eu fosse filho deles e a todos os amigos, que mesmo que indiretamente me deram uma força. Agradeço também ao IFPB e aos seus respectivos professores e gestores por contribuírem com minha formação e em especial ao nosso orientador Victor por ter nos auxiliado e ter dado esse mega suporte para que conseguíssemos concluir o nosso trabalho de conclusão de curso, obrigado!

Eu, Júlio César Urbano Vasconcelos Filho gostaria de agradecer, a Deus pelo dom da vida, aos meus pais, Júlio César e Gizelle Alves, por todo suporte, apoio, amor e carinho durante toda a minha jornada como ser humano, a minha namorada e os amigos que me apoiaram. Também sou grato ao meu orientador Victor André Pinho que tornou esse trabalho possível e também aos professores do curso que guiaram por esse caminho tortuoso, cheio de barreiras e desafios.

## RESUMO

SISPRO trata-se de um sistema idealizado e desenvolvido pensando na comunicação e seus benefícios. Nos dias atuais, os termos jurídicos ainda permanecem complexos para maioria dos cidadãos, pois os processos são atualizados por meio de movimentações, cada uma delas possuindo termos técnicos, o que torna o acompanhamento bastante difícil para o público em geral. O SISPRO veio para elucidar esses termos complexos e agilizar o cotidiano do advogado e do seu cliente, buscando o aprimoramento da agilidade foi implementado uma aplicação *mobile*. Para que a aplicação funcionasse de forma mais eficiente, foi preciso desenvolver um servidor que extraísse e processasse os dados em tempo real do Tribunal da Paraíba. O trabalho em questão explicará como esse projeto foi arquitetado e implementado, para isso foi necessário levantar os requisitos através de pesquisas e questionários, além de estudos relacionados à área de experiência de usuário. Partindo dessas ideias, surge o SISPRO, um aplicativo de atendimento ao cliente para consulta processual.

**Palavras-chave:** consulta processual, PJe, aplicação móvel, experiência de usuário, api rest.

## **ABSTRACT**

SISPRO is a system designed and developed thinking about communication and its benefits. Nowadays, the legal terms remain complex for most citizens, the processes are updated through movements, each of them having technical terms, which makes follow-up quite difficult for the general public. SISPRO came to elucidate these complex terms and streamline the daily life of lawyers and their clients. Seeking to improve agility, a mobile application was implemented. For the application to work more efficiently, it was necessary to develop a server that extracts and processes data in real-time from the Court of Paraíba. The work in question will explain how this project was designed and implemented, for that it was necessary to raise the requirements through surveys and questionnaires, as well as studies related to the area of user experience. Based on these ideas, SISPRO appears as a customer service application for procedural consultation.

**Keywords:** Procedural consultation, PJe, mobile application, User experience, Rest API.

## Lista de siglas e abreviaturas

APIs -Application Programming Interface

Art- Artigo

CNJ -Conselho Nacional de Justiça

Config- configuração

Controller- controladores

Create -Criar

CRUDs -Acrônimo da língua inglesa que aglutina as quatro operações básicas em uma base de dados relacional

Delete -Deletar

HTTP -HyperText Transfer Protocol - Protocolo de Transferência de Hipertexto

ISO- International Organization for Standardization

MVC - Model-View-Controller

PJe, - Processo Judicial eletrônico

Read- Ler

REST - Representational State Transfer - Transferência Representacional de Estado

SISPRO- Sistema Processual

SGBD- Sistema de Gerenciamento de Banco de Dados

SQL Structure Query Language ou Linguagem de Consulta Estruturada

TCC- Trabalho de Conclusão de curso

TDD- Test Driven Development - Desenvolvimento Orientado a Teste

TSE- Tribunal Superior Eleitoral

Update -Atualizar

UX- User Experience



## Lista de figuras

Figura 1 - <i>Smartphones</i> em uso no Brasil.....	15
Figura 2 - Entrevista perguntas 1 a 3.....	27
Figura 3 - Entrevista pergunta 8.....	27
Figura 4 - Diagrama de casos de uso.....	28
Figura 5 - Modelo Entidade Relacionamento.....	29
Figura 6 - Entidades Usuário e Processo.....	30
Figura 7 - Entidades Processo e Movimentação.....	31
Figura 8 - Entidades Parte e Advogado.....	32
Figura 9 - Estudo da Voz da marca.....	32
Figura 10 - Divisão do projeto.....	33
Figura 11 - Estrutura de pastas.....	34
Figura 12 - Estrutura detalhada.....	34
Figura 13 - Estrutura de pastas “src/test”.....	35
Figura 14 - Entidades implementadas.....	36
Figura 15 - Classe Usuário.....	36
Figura 16 - Diretórios “repository” e “service”.....	38
Figura 17- Estrutura de pastas do projeto <i>front-end</i> .....	39
Figura 18 - Diretório “services”.....	40
Figura 19 - Diretório “pages”.....	40
Figura 20 - Telas do aplicativo.....	43
Figura 21 - Tela de Processos.....	44
Figura 22 - Testes de cobertura de código.....	45

## Sumário

<b>1 Introdução</b> .....	10
1.1 Contextualização.....	10
1.2 Trabalhos Relacionados.....	11
1.3 Objetivos.....	12
1.3.1 Objetivo Geral.....	12
1.3.2 Objetivos Específicos.....	13
1.4 Estrutura do trabalho.....	13
<b>2. Fundamentação Teórica</b> .....	14
2.1 Aplicações Móveis.....	14
2.2 Experiência do Usuário (User Experience - UX).....	16
2.3 Do PJe.....	16
2.4 Servidor e comunicação Cliente-Servidor.....	17
<b>3 Metodologia</b> .....	20
3.1 Do processo metodológico.....	20
3.2 Tecnologias Utilizadas.....	22
3.2.1 Java.....	22
3.2.2 Spring Boot.....	23
3.2.3 MySQL.....	23
3.2.4 JavaScript.....	24
3.2.5 React Native.....	24
3.2.6 Expo.....	25
<b>4 Desenvolvimento do SISPRO</b> .....	26
4.1 Da Pesquisa.....	26
4.2 Da análise e elaboração dos documentos.....	28
4.3 Do desenvolvimento.....	33
<b>5 Resultados</b> .....	42
5.1 SISPRO.....	42
5.2 Testes.....	44
<b>6 Considerações Finais</b> .....	46
<b>Referências Bibliográficas</b> .....	47
<b>Apêndice</b> .....	49
A.1 Questionário.....	49
A.2 Requisitos funcionais e não funcionais identificados.....	50

# Capítulo 1

## 1 Introdução

Neste capítulo, será apresentado o contexto em que este trabalho está inserido, apontando a necessidade de um sistema para facilitar consultas a processos jurídicos de clientes. Em seguida, alguns trabalhos relacionados e produtos semelhantes já desenvolvidos serão apresentados e comentados. Na sequência, será apresentado o objetivo geral e os específicos. Por fim, encontra-se a estrutura e organização do restante da monografia.

### 1.1 Contextualização

Nas escrituras sagradas do cristianismo, segundo o livro de Gênesis, cidadãos da cidade da Babilônia construíram uma torre para chegar até o céu, porém tal atitude não agradou a Deus e, como consequência, ele os amaldiçoou confundindo suas línguas. Todos ficaram confusos, visto que não conseguiam comunicar-se de forma efetiva e clara uns com os outros. Com esse fato pode-se observar como a comunicação, desde os primórdios dos tempos, é essencial para a convivência entre os seres humanos. Como disse Aristóteles (384 - 322 a.C.), “o homem é um sujeito social que, por natureza, precisa pertencer a uma coletividade”.

Além da comunicação ser essencial para o homem, o seu desenvolvimento faz parte da evolução tecnológica, pois com o passar do tempo e com os avanços das revoluções industriais, a humanidade foi buscando cada vez mais velocidade na sua troca de informações, que ocasionou na globalização (HOBBSAWM, 2014).

Nesse ínterim, observa-se a substituição da comunicação física (papéis, cartas e documentos) pela comunicação virtual (através da internet). Tal avanço não foi diferente no meio jurídico, pois em meados de 2001, com a instituição dos Juizados Especiais Federais pela Lei n.º 10.259/01, surge, pela primeira vez, a possibilidade da prática dos atos processuais de forma completamente eletrônica. Ademais, com o surgimento do sistema chamado Processo Judicial eletrônico (PJe),

o padrão dos dias de hoje, é possível realizar no meio digital desde a petição inicial até o arquivamento do processo.

Nesse contexto, mesmo diante dos avanços tecnológicos e jurídicos, ainda há clientes que dependem de ligações para o escritório ou que precisam se deslocar até os advogados para o esclarecimento de dúvidas. Pois, embora existam os portais de consulta pública de cada tribunal, a barreira linguística e técnica dos portais os restringe. Conseqüentemente, essa demanda eleva a rotina de trabalho nos escritórios e gera necessidade de mais recursos humanos.

Diante do exposto, propõe-se o Sistema Processual (SISPRO), um sistema com a finalidade de resolver essa problemática. Tal sistema irá atender de maneira completa o cliente a respeito de seu processo de uma forma simples, traduzindo e simplificando as informações. Esse tipo de serviço não acontece em outras soluções semelhantes, pois geralmente as soluções que encontram-se hoje no mercado são voltadas para os advogados e não para os clientes.

## **1.2 Trabalhos Relacionados**

Ao buscar no mercado por soluções similares a que será apresentada neste trabalho, foram encontrados sistemas válidos a serem citados. São eles o Processo Judicial Eletrônico (PJe), Astrea e Jusbrasil.

O Processo Judicial Eletrônico<sup>1</sup> permite que magistrados, servidores e advogados pratiquem atos processuais diretamente no sistema, além de garantir a confiabilidade do processo judicial mediante o uso da certificação digital (TSE, 2018?).

Além de fornecer aos usuários autenticados todas as funcionalidades necessárias para a execução de um processo jurídico, esse sistema também conta com um ambiente para consultas públicas, fornecendo diversas informações de um processo, porém não há um cuidado em explicar as informações de uma forma que o público em geral possa compreender. É nesse ponto que o SISPRO consegue se

---

<sup>1</sup> PJe .Disponível em:< <https://www.pje.jus.br>> Acesso em : 07 ago. 2022.

diferenciar, mesmo que nesse caso ainda exista um espaço público no PJe, ainda existe uma barreira técnica.

Por sua vez, Astrea<sup>2</sup> é um produto desenvolvido pela Aurum uma *lawtech* (empresa do ramo de tecnologia jurídica). Essa empresa possui duas décadas de existência e intitula o Astrea como “O melhor software jurídico para escritórios de advocacia e advogados autônomos”. De fato é possível perceber que eles possuem diversas ferramentas e funcionalidades para auxiliar o escritório e os advogados com o gerenciamento de suas atividades, porém nada é citado sobre o cliente em si. Eis mais uma vez o motivo pelo qual é necessário o SISPRO.

Por fim, o Jusbrasil<sup>3</sup>, um site que é totalmente voltado ao cliente e oferece serviços de forma gratuita, possui uma série de limitações que impedem o usuário de obter todas as informações desejadas. Essa solução é a que mais se aproxima do SISPRO, com a diferença no modelo de monetização, já que no SISPRO a proposta é fornecer aos escritórios que o assinem a total disponibilidade de consulta para os clientes, ou seja, quem pagará pelo serviço é o escritório e não o cliente.

## **1.3 Objetivos**

### **1.3.1 Objetivo Geral**

O objetivo geral deste trabalho é o desenvolvimento de um sistema capaz de realizar consultas processuais - o SISPRO -, mantendo a experiência de usuário, elegibilidade e confortabilidade como principais princípios da aplicação, visando solucionar os problemas e pendências encontrados no cotidiano do consumidor comum de um sistema maçante, além de atender aos requisitos provenientes de uma entrevista realizada com o escritório Alves e Gondim localizado na cidade de João Pessoa, Paraíba.

---

<sup>2</sup> Astrea . Disponível em:< <https://www.aurum.com.br/astrea>> Acesso em : 07 ago. 2022.

<sup>3</sup> Jusbrasil. Disponível em:< <https://www.jusbrasil.com.br>> Acesso em : 07 ago. 2022.

### 1.3.2 Objetivos Específicos

Para alcançar o objetivo geral proposto, os seguintes objetivos específicos foram elencados:

- Identificar as necessidades e expectativas do cliente;
- Realizar uma revisão bibliográfica visando buscar trabalhos relacionados ao projeto proposto;
- Desenvolver *back-end*, o qual conterà toda a lógica de logins, cadastros, edição de usuários, listagens de processos e usuários, além de buscar na *API* do Tribunal Estadual da Paraíba todos os processos vinculados aos clientes cadastrados;
- Desenvolver *front-end*, para permitir que os usuários naveguem em todas as funcionalidades desenvolvidas no *back-end*.

### 1.4 Estrutura do trabalho

O presente trabalho de TCC está dividido em 6 capítulos. O presente capítulo, o Capítulo 1 tratou da introdução do trabalho, com sua contextualização e objetivos. No Capítulo 2 encontra-se a fundamentação teórica, onde será tratado sobre aplicações móveis, experiência do usuário PJe e o servidor.

No Capítulo 3 está presente a metodologia utilizada neste trabalho, com seu processo metodológico e tecnologias utilizadas, onde discorreu-se também sobre o Java, o Spring boot, o MySQL, JavaScript, React Native.

Já no Capítulo 4 encontra-se o desenvolvimento do aplicativo. O Capítulo 5 discorre sobre os resultados obtidos pelo aplicativo. No capítulo 6 está a conclusão do trabalho e, por fim, apêndice e referências.

## Capítulo 2

### 2. Fundamentação Teórica

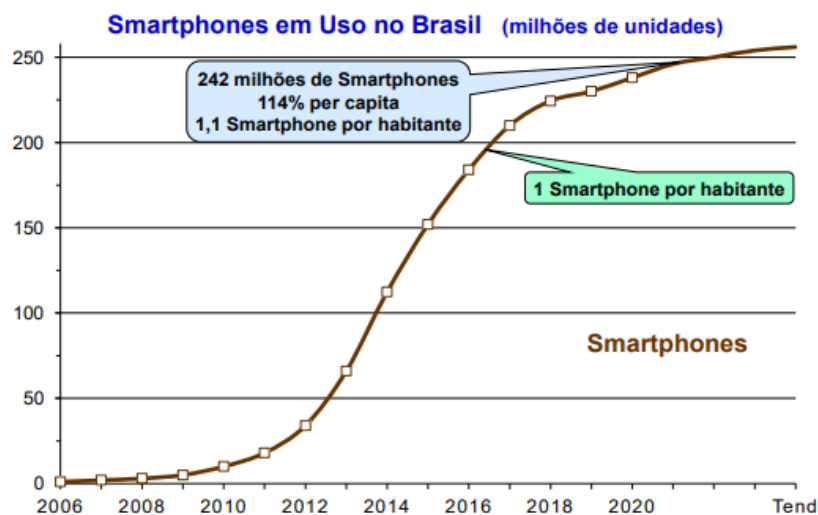
O presente capítulo tem como objetivo pautar os conceitos relacionados a este trabalho, partindo de uma discussão sobre desenvolvimento de aplicações móveis e seus respectivos impactos na sociedade, galgando pela preocupação com a experiência de usuário diante de causas processuais do meio jurídico visando facilitar sua linguagem para com os clientes. Ainda, visa elucidar o processo jurídico que é assistido pelas atividades anteriores, além de definir um servidor, explicar a comunicação cliente-servidor e elucidar o uso de protocolos, arquiteturas e padrões presentes nessa comunicação.

#### 2.1 Aplicações Móveis

O fundador da Apple, Steve Jobs disse, certa vez, que "a tecnologia move o mundo", fazendo uma comparação da importância e crescimento dos avanços tecnológicos no desenvolvimento da sociedade. Na atualidade, a tecnologia e a expertise humana corroboram com níveis de aprendizados e trocas de informações a nível mundial, desfrutando dos celulares e computadores, como os meios mais utilizados da modernidade.

Em junho de 2021, no Brasil, havia 1,2 telefone (fixo ou móvel) por habitante, evidenciando-se, então, o uso massivo desses dispositivos de forma crescente, como mostra a Figura 1. Nota-se, de modo geral, que a população possui preferência de utilização pelo *smartphone* em relação ao computador, e esse fenômeno é mais acelerado a depender da idade do usuário. Pessoas mais jovens são chamados de nativos digitais, visto que utilizam seus celulares para realizar suas tarefas do dia a dia, como marcar consultas, acessar serviços de streaming ou bancos (MEIRELLES, 2021). Posto isso, é possível perceber que com o passar dos anos e das gerações, as soluções móveis serão cada vez mais importantes e presentes no cotidiano.

**Figura 1.** *Smartphones* em uso no Brasil.



Fonte: MEIRELLES, 2021.

Segundo Brito (2019), as aplicações móveis são *softwares* desenvolvidos para dispositivos móveis (*smartphones*, tablets) administradas por um gestor e utilizadas por vários perfis que obedecem às funcionalidades disponíveis, mediante as permissões que lhes são atribuídas.

Tais aplicações surgiram como uma resposta ao mercado de *smartphones*, uma vez que a sua proliferação alterou o modelo convencional de comunicação, visto que transformaram de forma drástica a vida cotidiana da população. As aplicações móveis têm como finalidade simplificar várias tarefas para atividades diárias e para o ambiente de trabalho, pois fornecem funcionalidades idênticas ou superiores às aplicações tradicionais utilizadas nos computadores, podendo ser acessadas em qualquer lugar (BRITO, 2019).

Em concordância, Barra *et al.* (2018) afirmam que os aplicativos móveis visam atender o acesso das pessoas à informação e ao conhecimento, sem restrição de espaço e tempo, permitindo novas formas de comunicação. Tais características agregam valor estratégico para a nova sociedade da era da informação.



## 2.2 Experiência do Usuário (*User Experience - UX*)

Para que uma aplicação móvel possa compor a gama de ferramentas que o usuário pode acessar no seu dia a dia, designar uma solução em um aplicativo não é o suficiente, é preciso que ele possua qualidade, sendo necessário atentar à sua usabilidade e à experiência do usuário que vai utilizá-lo (VALENTIM *et al.*, 2015).

A usabilidade é definida como a “capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas” (*International Organization for Standardization - ISO/IEC 25010*, 2011). Por sua vez, UX é definida como “a percepção e as respostas de uma pessoa resultantes do uso ou da antecipação do uso de um produto” (*International Organization for Standardization DIS 9241-210*, 2010). Portanto, além de se preocupar em solucionar o problema, também deve-se ter cuidado em fornecer uma boa experiência antes, durante e depois do uso do aplicativo pelo cliente.

## 2.3 Do PJe

O PJe é uma plataforma digital desenvolvida pelo CNJ (Conselho Nacional de Justiça) em colaboração com múltiplos tribunais, com a participação do Conselho Nacional do Ministério Público, da Ordem dos Advogados do Brasil, advocacia pública e defensores públicos.

Do ponto de vista das funções disponíveis, o PJe caracteriza-se por preconizar a prática de atos jurídicos e acompanhar os processos processuais de forma padronizada, mas levando em consideração as características inerentes a cada Poder Judiciário. Visa direcionar esforços para a adoção de soluções judiciais padronizadas e gratuitas, com foco na racionalização dos ganhos de produtividade na atividade judiciária e na racionalização dos custos de desenvolvimento ou aquisição de software, permitindo o acesso a recursos financeiros e pessoas envolvidas em atividades relacionadas ao meio Judiciário.

Em 2013, foi aprovada a Resolução nº 185/2013 (CONSELHO NACIONAL DE JUSTIÇA (CNJ)<sup>4</sup>, estabelecendo o PJe como sendo o sistema nacional de processamento de informações e implementação de ações processuais. A referida resolução estabelece os preceitos para a implantação e funcionamento do sistema proposto e estabelece um Comitê Gestor dos Estados e do Distrito Federal e Territórios, formado por membros dos tribunais estaduais que fazem uso do PJe para iniciar a coordenação das ações judiciais e atividades dos tribunais de justiça.

Neste sentido, o PJe visa melhorar a infraestrutura, inovar, expandir módulos e dar suporte a todo o sistema. Os projetos são desenvolvidos com métodos ágeis, com portfólios planejados semestralmente, com foco nas necessidades dos usuários, produtividade dos serviços e facilidade de uso dos recursos.

O pressuposto central, está no fato de que a informatização do processo judicial é uma política pública fundamental e dessa forma deve observância aos padrões aplicáveis a todos os segmentos de justiça, independentemente de sua competência ou grau de jurisdição. A motivação para tanto decorre das disposições da Constituição da República (Art. 103-B, §4º, VI e VII) e do Código de Processo Civil (Art. 196), que atribui competência ao Conselho Nacional de Justiça para a gestão e a coordenação dessa política pública.

Dessa forma, é por meio do PJe que os advogados podem realizar petições e acompanhar o andamento de um processo, que o perito publica seus laudos e que um juiz pode proferir sua sentença, estabelecer despachos e decisões, por exemplo. Ou seja, é basicamente uma rede que interliga todos esses profissionais em torno dos autos (que consiste no conjunto de interações e documentos referentes a um processo).

## **2.4 Servidor e comunicação Cliente-Servidor**

Um servidor pode ser definido como um poderoso computador que armazena, centraliza e compartilha dados com quem se conecta com ele, os chamados clientes (TANENBAUM et al, 2010, p 4). Dito isso, é possível entender que o servidor é

---

<sup>4</sup> Resolução nº 185/2013, Disponível em:<<https://atos.cnj.jus.br/atos/detalhar/1933>>. Acesso em: 24 nov. 2022

responsável por cuidar dos dados e o cliente é quem os requisita. Mas como é realizada essa comunicação?

Essa comunicação entre cliente e servidor acontece através de um protocolo chamado de HTTP (*HyperText Transfer Protocol* - Protocolo de Transferência de Hipertexto), baseado em requisição e resposta. Assim, toda requisição ao servidor por um cliente possui uma resposta do servidor para o cliente.

As aplicações que processam requisições HTTP podem ser construídas a partir de APIs (*Application Programming Interface*) que, como define a Red Hat no artigo *O que é API?*:

As APIs funcionam como se fossem contratos, com documentações que representam um acordo entre as partes interessadas. Se uma dessas partes enviar uma solicitação remota estruturada de uma forma específica, isso determinará como a aplicação da outra parte responderá<sup>5</sup>.

Com efeito, é possível inferir que utilizar esse tipo de interface reduz bastante a complexidade de comunicação entre clientes e uma API, visto que os clientes não precisam conhecer profundamente a implementação daquela interface para obter respostas dela.

REST (*Representational State Transfer* - Transferência Representacional de Estado), segundo a Alura

é um modelo de arquitetura e não uma linguagem ou tecnologia de programação, que fornece diretrizes para que os sistemas distribuídos se comuniquem diretamente<sup>6</sup>.

Conforme pontuado, é possível perceber que essa arquitetura busca tornar mais específico e menos genérico a comunicação cliente e servidor via HTTP, delimitando os métodos e quais ações cada método pode realizar, além de padronizar também as respostas.

Surgem então as chamadas API REST que segundo o artigo da Red Hat, *O que é API REST?*

---

<sup>5</sup> O que é API?. Red Hat. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>>. Acesso em: 24 nov. 2022

<sup>6</sup> REST: Conceito e fundamentos. Alura. Disponível em: <<https://www.alura.com.br/artigos/rest-conceito-e-fundamentos>>. Acesso em: 24 nov. 2022

API REST, também chamada de API RESTful, é uma interface de programação de aplicações (API ou API web) que está em conformidade com as restrições do estilo de arquitetura REST.<sup>7</sup>

Essas APIs são orientadas e mapeadas pelos recursos que disponibilizam, os chamados *CRUDs* (acrônimo da língua inglesa que aglutina as quatro operações básicas em uma base de dados relacional, *Create* (Criar), *Read* (Ler), *Update* (Atualizar), *Delete* (Deletar)). Um CRUD é considerado um recurso quando ele é exposto em um *endpoint* (ponto de extremidade ou ponto de acesso, é uma ponte de comunicação com os recursos da API) mapeando cada operação básica com um método HTTP. Seguindo a arquitetura REST, temos *POST* para o *Create*, *Get* para o *Read*, *Put* para o *Update* e, por fim, *DELETE* para o *Delete*. Seguindo todas essas normas é possível desenvolver APIs com contratos simples e fáceis de seguir, além de possuir alta fidelidade com o resultado real.

---

<sup>7</sup> Red Hat. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Acesso em: 24 nov. 2022.

## Capítulo 3

### 3 Metodologia

Este capítulo está dividido em duas seções. A primeira é voltada para dissertar sobre metodologia propriamente dita, os passos envolvidos no desenvolvimento de toda a pesquisa, e que culminou com o desenvolvimento do SISPRO. A segunda seção está focada em apresentar, explicar e relacionar as tecnologias utilizadas com o sistema desenvolvido.

#### 3.1 Do processo metodológico

O processo para executar essa pesquisa ocorreu em três etapas: pesquisa, análise e elaboração de documentos e desenvolvimento.

Na fase de pesquisa, foi realizada uma revisão bibliográfica sobre o tema, juntamente com um estudo de mercado para identificar as soluções apresentadas e os problemas que elas solucionam. Em seguida, também foi realizada uma entrevista com o escritório de advocacia Alves e Gondim localizado na cidade de João Pessoa, Paraíba, objetivando-se alinhar o que foi descoberto na pesquisa e o que foi idealizado com o que realmente é necessário.

Realizada a entrevista, concluiu-se que a melhor solução seria um aplicativo móvel, capaz de realizar consultas a um servidor que processa, agrupa e armazena dados de um Tribunal judiciário, retornando para o usuário final as informações dos processos em seu dispositivo móvel.

Após a coleta de dados dos estudos realizados e da entrevista, foram levantados os requisitos funcionais e não funcionais deste sistema, o que viabilizou uma ideia nítida de como o sistema precisava ser construído, para que fosse possível avançar para a fase de análise e elaboração de documentos.

Nessa fase, o objetivo é analisar a solução proposta e modelá-la de uma forma que facilite o desenvolvimento e reduza erros. Foram criados, portanto, o diagrama de caso de uso se atentando aos requisitos levantados anteriormente e o diagrama de atividade, a partir dos quais foi modelado o diagrama de entidade-relacionamento.

Concluída a modelagem das entidades, foi realizado um estudo de experiência de usuário chamado Voz da Marca (Podmajersky, 2019), com a finalidade de esse estudo contribuir significativamente para o desenvolvimento do desenho das telas do aplicativo e posteriormente da aplicação em si. Após a criação dos documentos citados acima, os autores possuíam uma ideia bastante nítida do que era necessário ser desenvolvido. Começa, então, a etapa de desenvolvimento.

Essa etapa pode ser dividida em duas, o servidor ou *back-end* (o que está por trás das aplicações desenvolvidas na programação, isto é, tudo que dá estrutura e apoio às ações do usuário) e o aplicativo ou *front-end* (o que está relacionado com a interface na qual o usuário vai interagir). Como o *back-end* concentra as regras de negócio, o desenvolvimento foi iniciado por ele.

Primeiramente, foram criados tanto o projeto *back-end* quanto o *front-end* no *GitHub* (site público de repositório de código<sup>8</sup>). Na sequência, deu-se início ao desenvolvimento do servidor, iniciando-se pelo banco de dados (será abordado banco de dados na seção 3.2.3) que foi criado a partir da modelagem de entidades feita anteriormente.

Com o banco de dados criado, deu-se início a criação dos *CRUDs* de cada entidade, além de expô-los em *endpoints* separados. É válido ressaltar que para realizar tal desenvolvimento seguiu-se a metodologia de *TDD* (*Test Driven Development* - Desenvolvimento Orientado a Teste), a fim de assegurar a qualidade e confiabilidade do código escrito.

Com todos os *CRUDs* criados, partiu-se então para a criação das rotinas que iriam colher os dados do Tribunal da Paraíba. Foram mapeadas 2 rotinas necessárias, uma de cadastro de processos, para cadastrar todos os processos que estão em andamento de cada advogado cadastrado no sistema, e uma outra rotina,

---

<sup>8</sup>Disponível em: < <https://github.com>>. Acesso em: 24 nov. 2022.

que é executada com mais frequência, incumbida de atualizar os processos cadastrados.

Com as rotinas criadas, foi desenvolvido o último requisito do *back-end*, o sistema de autenticação, desenvolvido utilizando-se o padrão de autenticação *basic* (padrão de autenticação simples que codifica usuário e senha no cabeçalho de cada requisição utilizando *base64*). Foram criados níveis para os usuários desde o usuário mais simples (que é um cliente, que pode apenas consultar os processos cadastrados no seu nome) até o administrador (que pode, além de consultar os processos no geral, também cadastrar usuários, sejam clientes ou administradores).

Com isso, foi finalizado o desenvolvimento do servidor, abrindo caminho para a criação do aplicativo. Para o *design* do *front-end* foi utilizado o figma (site para modelagem das telas<sup>9</sup>). Ainda na montagem das telas fez-se uso de todo o conhecimento adquirido pelo livro de UX (Podmajersky, 2019) para a redação dos textos presente no SISPRO, visando a melhor qualidade no produto final, além de guiar o tamanho dos botões e os fluxos dentro do aplicativo, por exemplo.

Com todas as telas montadas e mapeadas, deu-se início ao desenvolvimento do *front-end* de fato. Inicialmente as telas desenhadas no figma foram transferidas para a sua versão no código. Na sequência, foi implementado a lógica em cada uma das telas, uma de cada vez, até terminar todo o desenvolvimento das telas. Ademais, o armazenamento em cache para salvar as sessões de um usuário foi utilizado para que não seja necessário inserir as credenciais sempre que um usuário for acessar a aplicação.

## 3.2 Tecnologias Utilizadas

### 3.2.1 Java

Java é uma linguagem de programação orientada a objetos, sendo uma das mais utilizadas pelas empresas na atualidade no desenvolvimento de aplicações WEB e Mobile (Github, 2022).

O Java foi criado em 1995 na empresa Sun Microsystem por uma equipe chefiada por James Gosling, conhecido como o pai do Java. Em 2008, o Java foi

---

<sup>9</sup> <https://www.figma.com/>

adquirido pela Oracle e vem crescendo bastante desde então no mercado de tecnologia.

Ele foi utilizado devido ao seu grande uso no mercado atual de tecnologia, pois possui compatibilidades com todas as outras ferramentas utilizadas no projeto. Também vale salientar a praticidade das bibliotecas e *frameworks* disponíveis. Portanto, todo o *back-end* da aplicação foi realizado com essa linguagem.

### 3.2.2 Spring Boot

Spring boot é um *framework* Java *open source* que nasceu em 2014 com o objetivo de facilitar a criação de projetos Java que necessitavam de um servidor configurado, entre outros problemas (SAMPAIO, 2021). Essa tecnologia traz diversas configurações padrões para problemas já conhecidos e permite que o desenvolvedor altere somente o que for necessário através de simples anotações marcadas com “@”, além de ser possível criar arquivos de configurações programaticamente.

Foi escolhido esse framework por sua enorme praticidade e agilidade no desenvolvimento, além de ser especializado no desenvolvimento de aplicações *API REST*, onde foi utilizado por todo *backend* do produto.

### 3.2.3 MySQL

O MySQL foi desenvolvido por um empresa sueca chamada MySQL AB em 1994, que por definição trata-se de um Sistema de Gerenciamento de Banco de Dados (SGBD) relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados (Devmedia, 2013). O serviço utiliza a linguagem SQL (*Structure Query Language* ou Linguagem de Consulta Estruturada).

O MySQL foi desenvolvido na linguagem C/C ++, o que faz com que ele seja harmonizável com a maioria dos sistemas operacionais oferecidos na atualidade.

Essa tecnologia foi escolhida devido a sua portabilidade, por conseguir integrar-se com diversos sistemas, pelas formas de armazenamento, conveniente



aos diversos tipos de tabelas para armazenamento de dados, a sua velocidade e a sua capacidade.

Através dos pontos citados, o MySQL foi o banco escolhido para o armazenamento das informações.

### 3.2.4 JavaScript

JavaScript é a linguagem mais utilizada no mundo desde 2014 segundo uma pesquisa realizada pelo Github se mantendo até hoje como tal, graças a sua versatilidade (Github, 2022).

Essa tecnologia surgiu para ser executada em navegadores para manipular comportamentos de páginas Web no lado cliente. É uma linguagem de alto nível e não fortemente tipada.

A linguagem JavaScript foi escolhida para o desenvolvimento do aplicativo, pois graças a sua popularidade e versatilidade, foram desenvolvidas diversas ferramentas de qualidade no mercado que a utilizam e que atendem aos requisitos desejados. Assim agiliza o desenvolvimento e traz qualidade também ao produto final. Além disso, ela é a linguagem padrão operada pela ferramenta React Native.

### 3.2.5 React Native

O React Native é um framework pautado no já proclamado React, desenvolvido pela equipe do Facebook, que possibilita o desenvolvimento de aplicações mobile, tanto para Android, como para iOS, utilizando somente Javascript (Meta, 2022).

Graças ao React Native é possível utilizar a versatilidade e agilidade do JavaScript no universo de aplicações móveis, razão pela qual todo o *front-end* da aplicação está baseado nesse framework.

### 3.2.6 Expo

O Expo é um *framework open source* criado a partir do *React Native* (Expo, 2022). Ele provê uma vasta gama de ferramentas e soluções para problemas já conhecidos, agilizando o desenvolvimento do sistema. Com o expo é possível testar o projeto sem ter que construir os artefatos nativos através da plataforma que ele provê.

## Capítulo 4

### 4 Desenvolvimento do SISPRO

Esse presente capítulo tem o objetivo de explicar em detalhes o desenvolvimento do SISPRO seguindo a metodologia apresentada no capítulo anterior, também dividindo entre pesquisa, análise e elaboração de documentos e desenvolvimento. Na Seção 4.1 deste trabalho, encontra-se o estudo de mercado a respeito das soluções relacionadas e também a entrevista realizada com o escritório Alves Gondim. Na Seção 4.2 é detalhado como foram modelados os documentos citados no Capítulo 3, e, por fim, na Seção 4.3 é explicado como se deu o desenvolvimento e implementação do SISPRO em si.

#### 4.1 Da Pesquisa

Como já mencionado no Capítulo 3 foi nessa etapa que foi realizada a revisão bibliográfica, visando o embasamento teórico para que seja possível a construção de um produto como o SISPRO. Feita a revisão, partiu-se então para a entrevista com um escritório jurídico já mencionado na Seção 1.3.1, entrevista a qual encontra-se no Apêndice no final deste trabalho. Ao todo são 9 perguntas abertas, dentre as quais as primeiras são sobre o problema em si, como é possível ver na Figura 2:

**Figura 2.** Entrevista perguntas 1 a 3.

1. O que você precisa que seja informatizado/automatizado?
  - a. O Atendimento de prestar informações sobre os processos aos clientes.
2. Por que você precisa do sistema?
  - a. Para automatizar o atendimento e reduzir custos com funcionários.
3. Como o sistema pode melhorar a sua realidade (suas atividades ou de outras pessoas)?
  - a. Além de melhorar a satisfação dos clientes, auxilia a secretária e reduz a folha de funcionários do escritório.

**Fonte:** Autoria própria.

É possível perceber que existe essa demanda por informações pelo cliente, além da preocupação com a satisfação do cliente, ou seja experiência do cliente, tópico que até então é base para esse projeto.

Na pergunta 8 da entrevista foi questionado se o escritório preferia que a solução fosse disponibilizada como um aplicativo ou como um sistema que funcionasse através da Web, como mostrado na Figura 3 abaixo:

**Figura 3.** Entrevista pergunta 8.

8. Do ponto de vista da experiência do cliente é melhor que as funcionalidades de consulta processual sejam disponibilizadas através de um aplicativo ou uma página web?
  - a. Aplicativo.

**Fonte:** Autoria própria.

Essa resposta, novamente, se encaixa perfeitamente com a Seção 2.1 sobre aplicações móveis, sendo possível perceber que o pensamento do escritório entrevistado está alinhado com a tendência apontada pela pesquisa do MEIRELLES (2021).

A partir da entrevista e da revisão realizada foi possível modelar os requisitos funcionais e não funcionais do sistema, que também se encontram no Apêndice do projeto(A2).

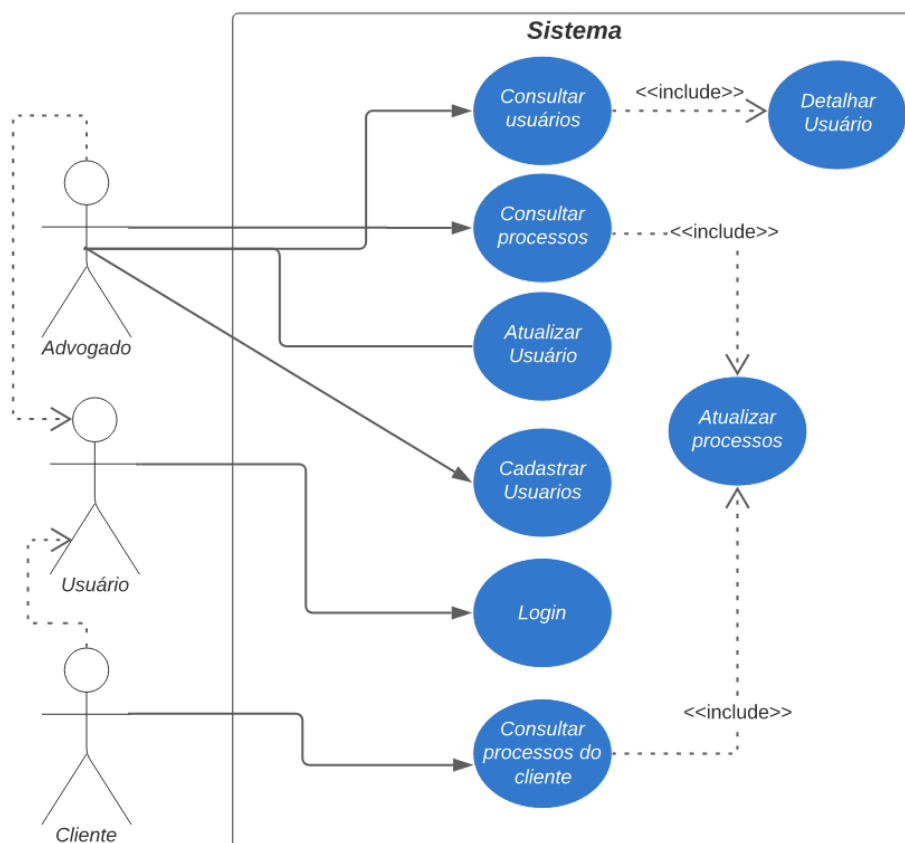
Após isso foi feita uma busca no mercado na tentativa de encontrar soluções que são semelhantes ao SISPRO de alguma forma. Assim foi possível concluir que

dadas as divergências citadas na Seção 1.2 e também com as informações obtidas das pesquisas realizadas, foi possível concluir a importância do projeto.

## 4.2 Da análise e elaboração dos documentos

O primeiro documento que foi elaborado foi o diagrama de caso de uso (Figura 4):

**Figura 4.** Diagrama de casos de uso.



**Fonte:** Autoria própria.

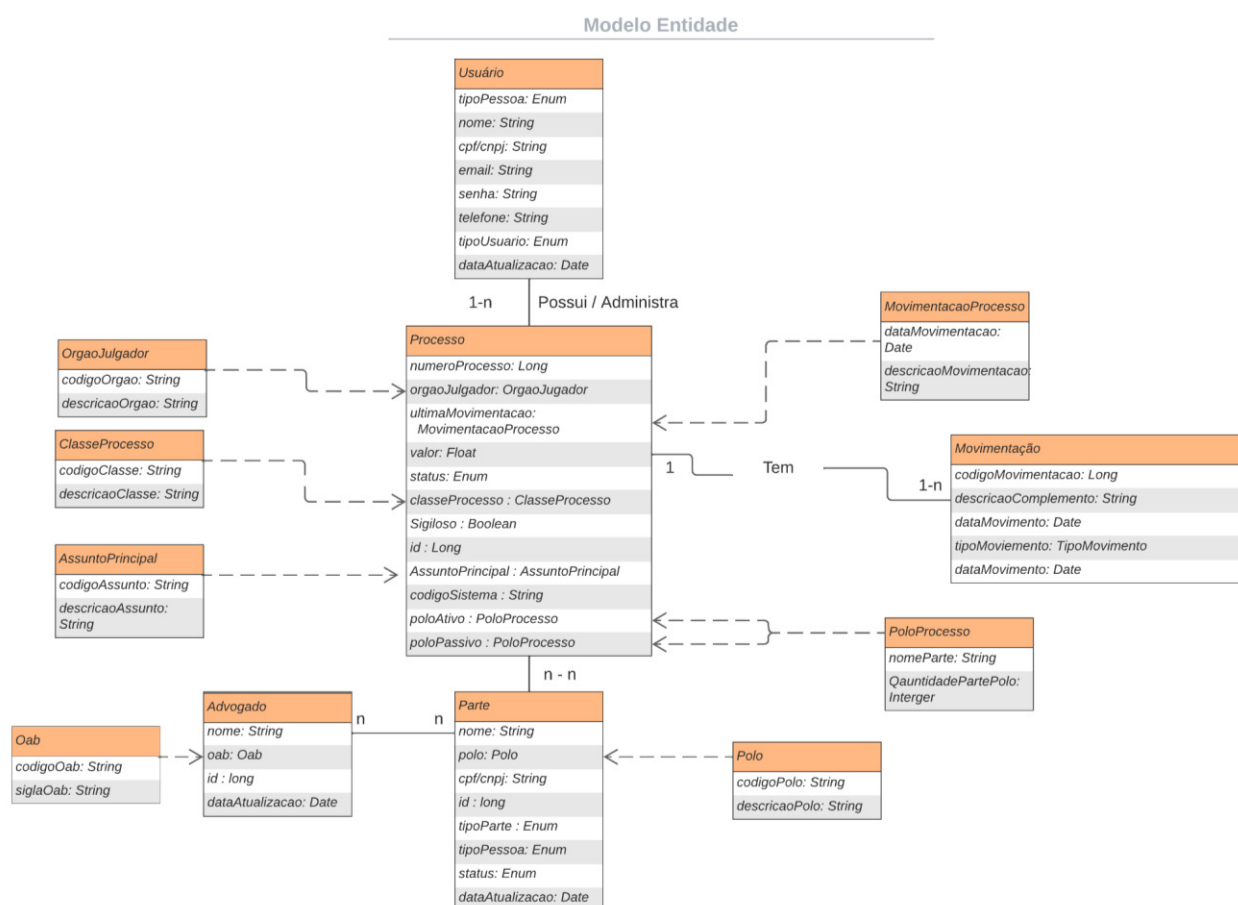
É possível identificar 3 atores: o advogado, o cliente e o usuário. O usuário representa qualquer usuário, seja ele advogado ou cliente. É possível visualizar que qualquer usuário pode realizar o *login* (entrar no sistema), mas todas as outras ações estão restritas às especializações.

O cliente pode consultar os próprios processos, mas todas as outras ações são tomadas pelo advogado ou pela rotina do sistema.

O advogado além de poder consultar todos os processos, também pode realizar o cadastro, consulta, detalhe e atualização de usuários, mas a ação de atualizar processo é tomada pela rotina do sistema.

Em seguida foi necessário analisar melhor como funcionaria o servidor e as entidades que ele teria, foi modelado então o modelo de entidade relacionamento que encontra-se na Figura 5:

**Figura 5.** Modelo Entidade Relacionamento

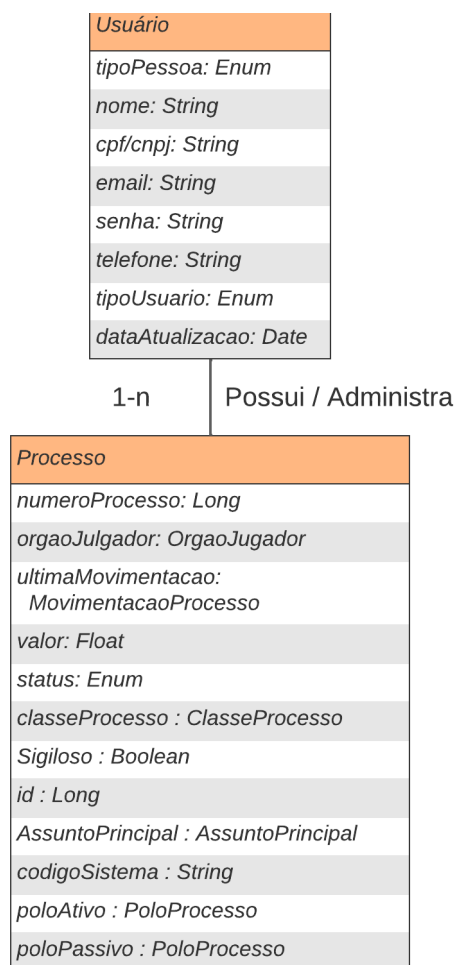


**Fonte:** Autoria própria.

As entidades desse sistema são: Usuário, Processo, Advogado, Parte e Movimento, os outros são apenas campos compostos que estão detalhados no

modelo. Na Figura 6 encontra-se uma imagem mais detalhada das entidades Usuário e Processo.

**Figura 6.** Entidades Usuário e Processo.

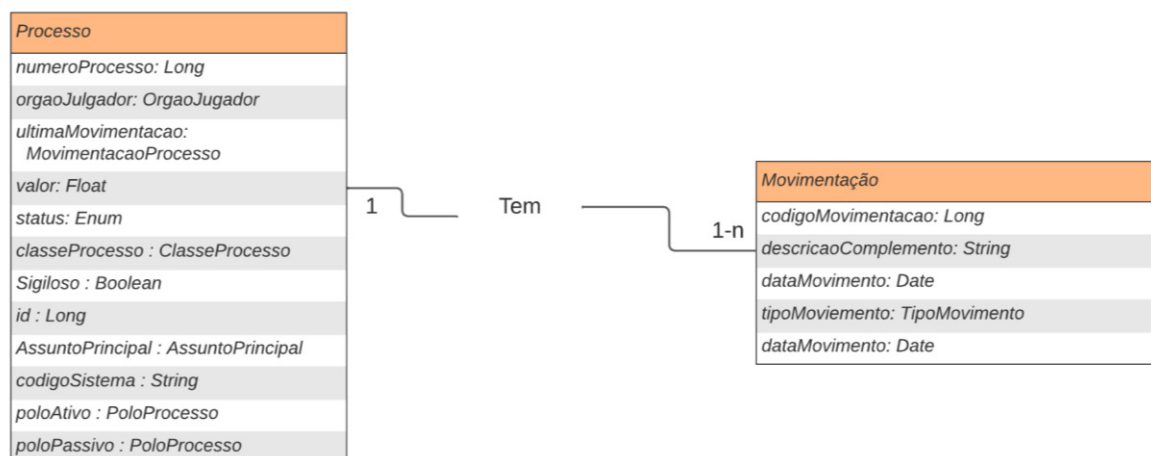


**Fonte:** Autoria própria.

Essas são as entidades principais do sistema, um usuário podendo ser cliente ou advogado, como visto no diagrama de caso de uso, então ele pode possuir processos em seu nome caso seja um cliente ou administrar os processos caso seja um advogado.

A entidade Processo detém diversos campos que fornecem informação para que o sistema consiga classificá-lo e com isso alterar o campo de status para os mais diversos, esses dados que o processo possui foram criados baseados em respostas de requisições à *API* do Tribunal da Paraíba.

Outra entidade essencial para esse sistema é a Movimentação e sua relação com a de Processo, detalhada melhor a seguir na Figura 7:

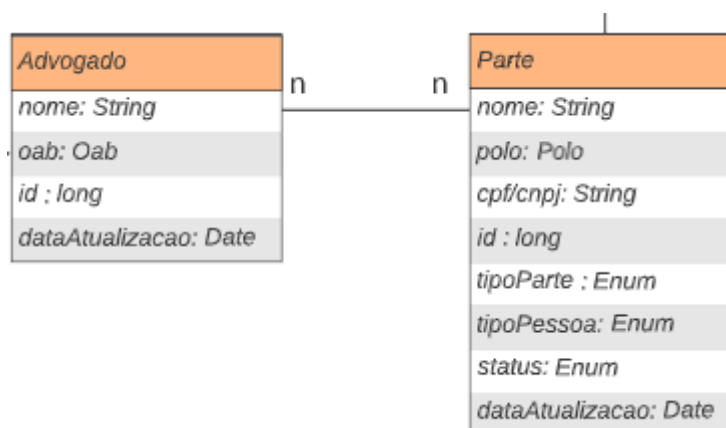
**Figura 7.** Entidades Processo e Movimentação.

**Fonte:** Autoria própria.

Segundo um artigo escrito pelo Tribunal Regional do Trabalho da 4ª região do Rio Grande do Sul (TRT, 2022) "Entende-se por movimentação processual o andamento do processo ao longo da tramitação, que obedece fases e procedimentos definidos por lei e registrados pelo Sistema de Gerenciamento de Informações Administrativas e Judiciárias da Justiça". A entidade de movimentação foi modelada dessa forma, estando contida várias delas em cada processo, já que qualquer atualização processual é gerada uma movimentação no Tribunal, com isso é possível estabelecer essa relação de muitas movimentações para um mesmo processo.

As últimas entidades a serem faladas são a de Parte e Advogado, mostradas na Figura 8.



**Figura 8.** Entidades Parte e Advogado.

**Fonte:** Autoria própria.

Essas entidades representam as partes e os seus advogados em cada processo. Com essas informações é possível saber quais processos estão relacionados com os advogados cadastrados no sistema, além dos clientes também, já que todo cliente que possui um processo em andamento é parte desse processo.

Por causa das pesquisas realizadas anteriormente, foi percebida a necessidade de criar um modelo para nortear a forma como os elementos são dispostos em tela, a escrita, tudo que relaciona a experiência do usuário a utilizar o aplicativo, esse modelo se chama Voz da marca e está demonstrado na Figura 9:

**Figura 9.** Estudo da Voz da marca

		Clareza	Eficiência	Praticidade
	Conceitos	Palavras objetivas e utilizar cores para enfatizar sentido	Comunicação precisa e velozes	Fluxos atividades menores e reaproveitar elementos (ex.: gastar o menor tempo possível pra realizar uma atividade)
	Vocabulário	Palavras objetivas, diminuir a complexidade de termos técnicos	Tentar usar textos mais curtos	Tentar usar textos mais curtos
	Verbosidade	{N/A}	Utilizar apenas a quantidade de palavras necessárias, sem redundância	Utilizar apenas a quantidade de palavras necessárias, sem redundância
	Gramática	Priorizar o tempo presente	Priorizar o tempo presente, caso seja necessário utilizar outro	Utilizar de palavras para significar botões
	Pontuação	Em urgências utilizar de exclamação	Evitar pontuações terminais	Evitar pontuações terminais e Uso de ícones
	Capitalização	Capitalizar STATUS	Somente as primeiras letras de palavras de botões, títulos e sessões serão capitalizados	Somente as palavras principais serão capitalizadas

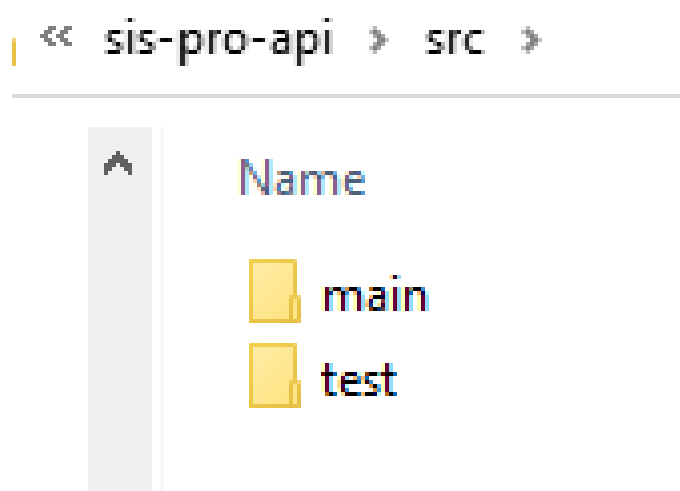
**Fonte:** Autoria própria.

Esse estudo é pautado nos 3 conceitos chave dessa aplicação, Clareza, Eficiência e Praticidade, nas colunas encontram-se esses 3 conceitos, e nas linhas possuem características, onde explica como cada conceito se relaciona com aquela característica.

### 4.3 Do desenvolvimento

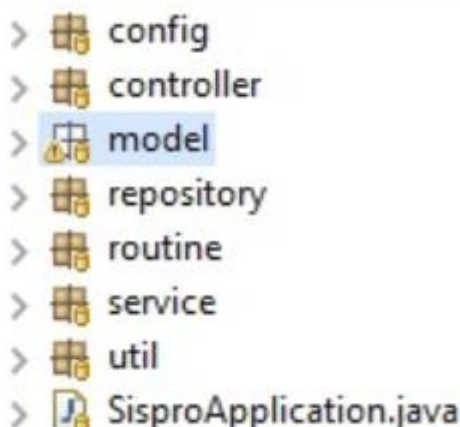
A fase de desenvolvimento foi iniciada com a criação do projeto *back-end*, como mencionado no seção 3.2 para a construção da *API* foi utilizado Spring Boot, um *framework* Java, que é especializado em desenvolver este tipo de aplicação, o projeto foi dividido em 2 diretórios principais, como será demonstrado na Figura 10.

**Figura 10.** Divisão do projeto



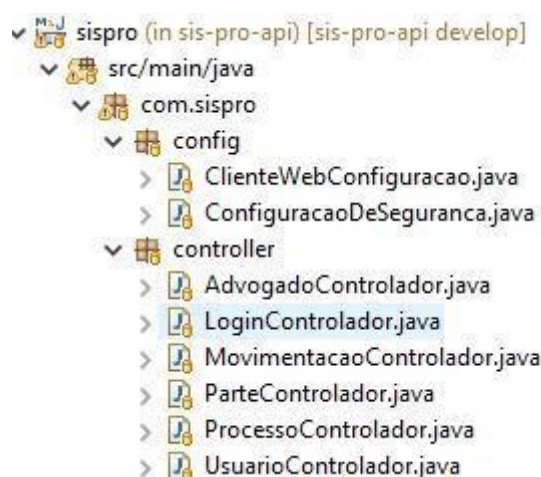
**Fonte:** Autoria própria.

O primeiro diretório “src/main” é onde está o projeto em si, é onde tem todas as funcionalidades. O segundo diretório “src/test”, é totalmente voltado a testes, ele replica parte da estrutura de pastas do primeiro com a diferença de possuir vários arquivos de testes, além de dividi-los por funcionalidade, ou seja, cada funcionalidade tem o seu próprio arquivo para testes. A Figura 11 apresenta a estrutura de pastas do diretório “src/main”.

**Figura 11.** Estrutura de pastas

**Fonte:** Autoria própria.

Detalhando a estrutura, pode-se observar um único arquivo .java na raiz, é o arquivo principal do projeto, é ele que inicializa que o servidor, já a Figura 12 detalha as pastas “config” e “controller”.

**Figura 12.** Estrutura detalhada

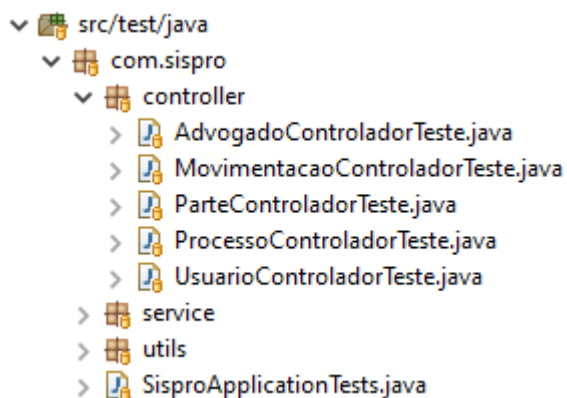
**Fonte:** Autoria própria.

Na pasta “config” é onde fica a parte de configuração. Na pasta “controller”, é onde ficam os controladores que são os arquivos que expõem os recursos da API. Então cada arquivo desses vai ter o seu próprio endpoint, e com isso fica bem separado e específico qual endpoint pertence a qual entidade.

Além disso, foi utilizada como metodologia de desenvolvimento o TDD (*Test-Driven Development* - Desenvolvimento Orientado a Teste), então para cada

funcionalidade da API foram criados testes para ela antes mesmo do seu próprio desenvolvimento, para garantir a qualidade do produto, mais detalhes da estrutura da pasta de “src/test” podem ser encontrados na Figura 13.

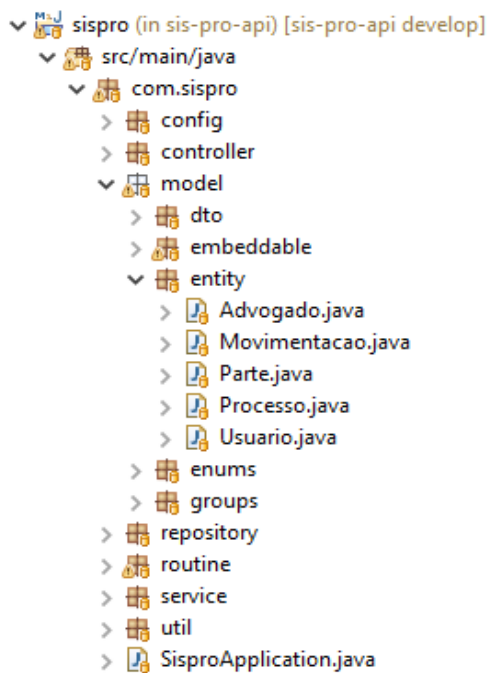
**Figura 13.** Estrutura de pastas “src/test”



**Fonte:** Autoria própria.

Como mencionado anteriormente, esse diretório busca replicar o diretório de funcionalidades, o diretório “src/main”, no entanto os arquivos dentro das pastas são arquivos de teste que testam exatamente os arquivos que estão nesse lugar só que no diretório principal, inclusive até o arquivo na raiz do diretório, o arquivo *main* “*SisproApplicationTests.java*” que nesse caso testa o arquivo *main* do outro diretório.

As entidades, que foram modeladas anteriormente, foram implementadas no diretório “model/entity” como podem ser visualizadas na Figura 14.

**Figura 14.** Entidades implementadas.

**Fonte:** Autoria própria.

Esse diretório “model” agrupa as classes que são puramente dados da aplicação, sejam entidades que são a representação das tabelas no banco, ou *enums*, entre outros. A Figura 15 apresenta a classe “Usuario.java”, e nessa classe está a implementação da entidade.

**Figura 15.** Classe Usuário.

```

package com.sispro.model.entity;

import java.time.LocalDateTime;

@Entity
@AllArgsConstructor
@NoArgsConstructor
@Builder
@GroupSequenceProvider(value = UsuarioGroupSequenceProvider.class)
public class Usuario implements UserDetails, Cloneable {

    private static final long serialVersionUID = -5918664436808499203L;

    @Id
    @Getter
    @Setter
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Getter
    @Setter
    @NotBlank(message = "O campo não pode ser vazio.")
    @Pattern(regexp = "^[\\d]{11}$|^[\\d]{14}$", message = "Este campo em caso de CPF ou CNPJ")
    @CPF(groups = PessoaFisica.class)
    @CNPJ(groups = PessoaJuridica.class)
    private String cpfCnpj;

    @Column
    @Getter
    @Setter
    @NotBlank(message = "O campo não pode ser vazio.")
    private String nome;

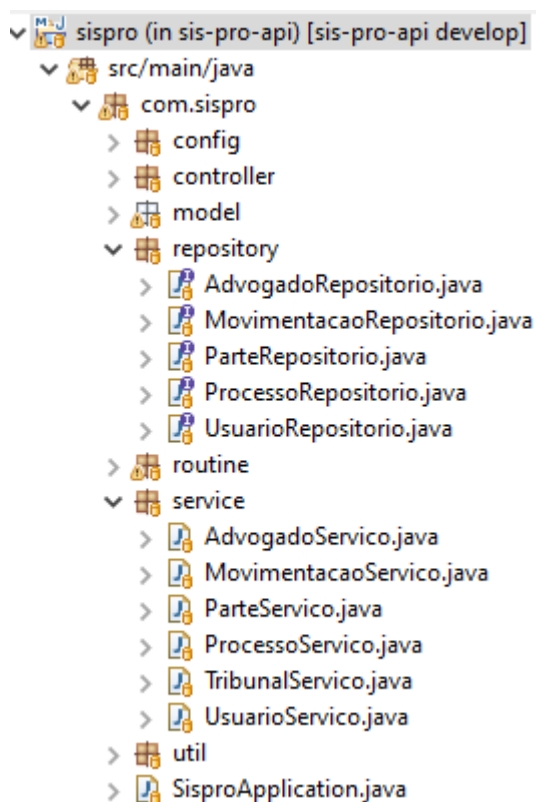
```

**Fonte:** Autoria própria.

É possível perceber os trechos de código marcados com “@” para que o Spring Boot já implemente automaticamente comportamentos predefinidos em cada anotação, no caso a anotação “@Entity”, acima da definição da classe, ela especifica ao Spring que essa classe representa uma tabela no banco que inclusive deve ser criada se ela ainda não existir no momento em que a aplicação é executada, se a classe representa uma tabela, os seus atributos representam as colunas dela, como por exemplo o atributo id que está marcado com a anotação “@Id” que marca tal campo como a chave primária da tabela.

Para ser possível executar operações com o banco de dados é necessário utilizar os chamados repositórios que estão agrupados no diretório “repository”, é possível visualizar o diretório “repository” em conjunto com o “service” na Figura 16.

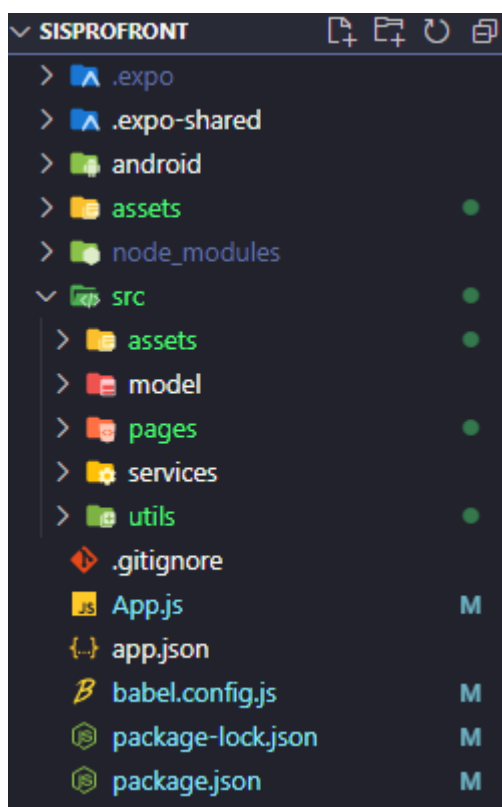
**Figura 16.** Diretórios “repository” e “service”.



**Fonte:** Autoria própria.

A camada de serviço agrupada no diretório “service” utiliza a implementação das interfaces feita pelo próprio *framework* para criar a regra de negócio de cada entidade e utilizando os controladores já citadas anteriormente é possível criar o *CRUD* de cada entidade ou recurso e ainda os expor em endpoints separados, exatamente como a arquitetura *Rest* define.

Após a finalização do *back-end* deu-se início ao desenvolvimento do *front-end*, como mencionado anteriormente na Seção 3.2. Para a construção do aplicativo foi utilizado o *framework React Native* juntamente com o *Expo*, que executam em cima da linguagem *Javascript*. Com essas 3 ferramentas foi possível implementar as telas e a lógica por trás de cada página do aplicativo. A estrutura de pastas do projeto é possível ser visualizada na Figura 17.

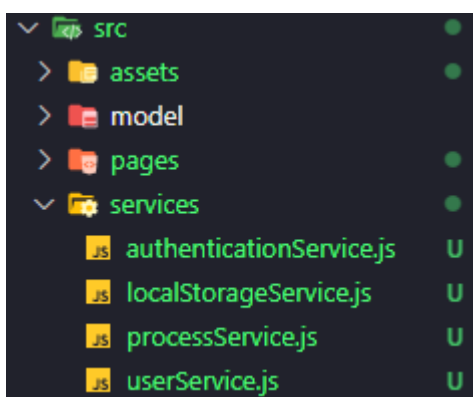
**Figura 17.** Estrutura de pastas do projeto *front-end*.

**Fonte:** Autoria própria.

Nesse tipo de projeto ele segue uma estrutura um pouco diferente da que a API utiliza, pois o diretório “assets” armazena arquivos que não são exatamente códigos do projeto, como, por exemplo, imagens e arquivos de fonte que o aplicativo utiliza.

O diretório “model” possui os arquivos de dados assim como na API, já o diretório “services” agrupa arquivos com funcionalidades que podem ser utilizadas em qualquer lugar do projeto, como por exemplo, a comunicação com os recursos da API. Esse tipo de regra é isolado em arquivos próprios dentro desse diretório, como é possível visualizar na Figura 18.

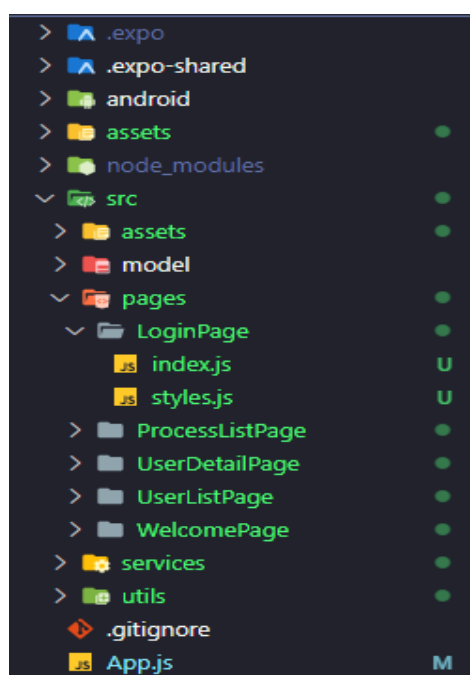


**Figura 18.** Diretório “services”.

**Fonte:** Autoria própria.

Como mencionado anteriormente, toda a comunicação com API é realizada nesses arquivos, porém é possível perceber que cada entidade por possuir seu próprio *endpoint* acaba possuindo seu próprio serviço. Como o aplicativo não chega a utilizar os outros recursos, não é necessário implementar a comunicação com esses endpoints, bastando apenas o de autenticação, o de usuário e o de processo.

No diretório “pages” é onde está localizado o código de cada página ou tela do aplicativo, como é possível visualizar na Figura 19.

**Figura 19.** Diretório “pages”.

**Fonte:** Autoria própria.

As páginas são diretórios que contém dois arquivos o “index.js” que contém o visual da tela e a lógica, e o arquivo “styles.js” contém apenas um objeto com os estilos utilizados no visual.

Foi dessa forma que foi possível desenvolver o SISPRO, seguindo os modelos, arquiteturas e padrões mostrados.

# Capítulo 5

## 5 Resultados

Neste capítulo são apresentados os principais objetivos alcançados pelo SISPRO. Na Seção 5.1 é descrito toda a idealização do produto final, relacionando o *front-end* com tudo que foi estudado mediante a experiência do usuário e como isso trará benefícios no uso da aplicação, e na Seção 5.2 é demonstrado todos os testes realizados pelo servidor que garantem pleno funcionamento do sistema.

### 5.1 SISPRO

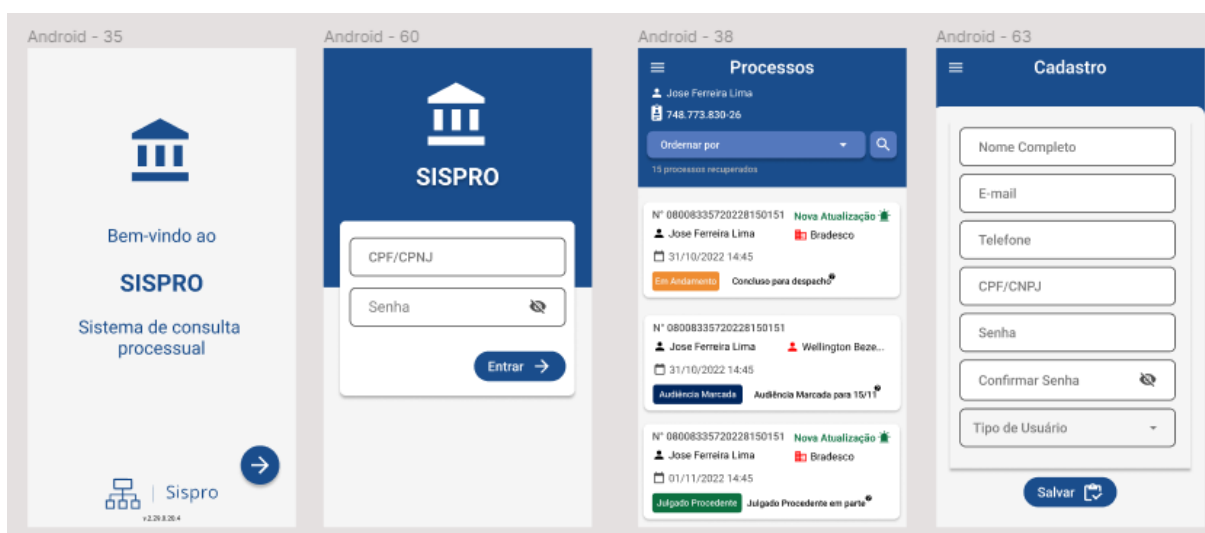
O SISPRO é um sistema desenvolvido para facilitar o dia a dia do advogado e principalmente do cliente desse escritório. A idealização do projeto partiu da conjectura da facilitação da comunicação entre o advogado e cliente, visando agilizar e elucidar toda a situação de um processo jurídico sem toda a complexidade dos termos utilizados pelo meio. Com isso, foram focalizados estudos dentro da área de experiência de usuário, principalmente através das estratégias de redação na ideia de facilitar o entendimento, aumentar engajamento, a conversão e a retenção em cada palavra.

Em termos mais técnicos, o sistema foi desenvolvido no modelo de projeto *Model-View-Controller* (MVC), o qual é utilizado em muitos projetos, sobretudo pela arquitetura que ele possui, pois possibilita a divisão do projeto em camadas, sendo elas o *model*, o *controller* e a *view* (DevMedia, 2013).

A solução usando MVC tem como benefícios o isolamento das regras de negócios da lógica de apresentação (DevMedia, 2013), como foi separada toda a tratativa de banco de dados de toda a parte da interface utilizada pelo usuário.

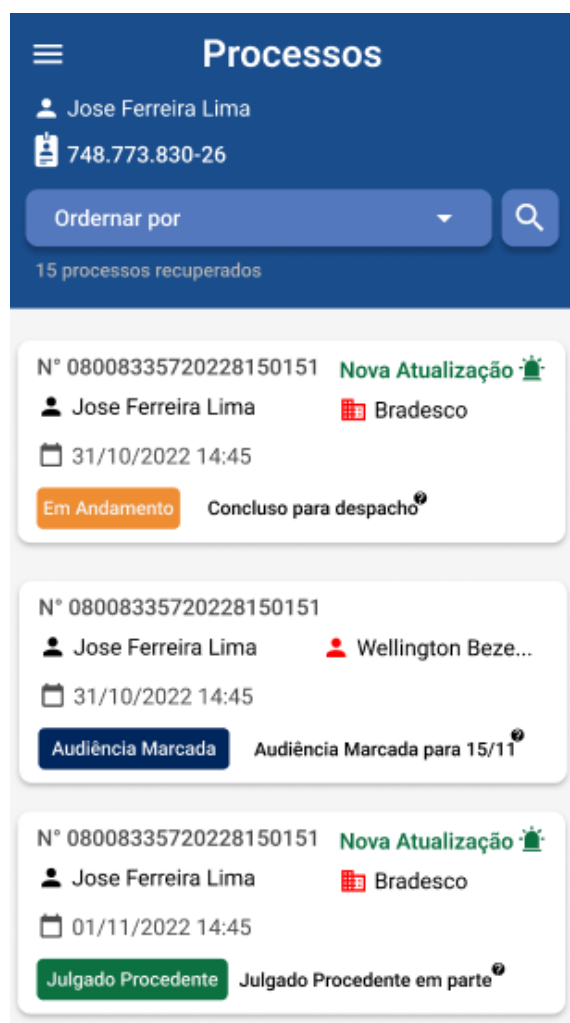
Para a implementação das telas, a ideia foi otimizar o processo de consultas, simplificando o andamento dos processos trazendo a melhoria da comunicação do cliente e do advogado em tempo real, para isso foram desenvolvidas as telas a seguir na Figura 20:

Figura 20. Telas do aplicativo



Fonte: Autoria própria.

Para a visualização dos processos foi construída a tela de processo onde teremos todas as informações do andamento do processo, assim como será mostrada na Figura 21. Nela é possível perceber que contém de forma simplificada as informações necessárias para que o cliente de forma rápida e prática consiga entender o andamento do processo. Se ainda sim o usuário possuir alguma dúvida com a descrição do status, que acaba contendo mais informações técnicas, ele pode com um simples toque no ícone de "?", ter uma rápida explicação daquele termo em específico. O presente aplicativo, além de sanar problemas diretos, servirá de base para a implantação de futuros sistemas.

**Figura 21.** Tela de Processos.

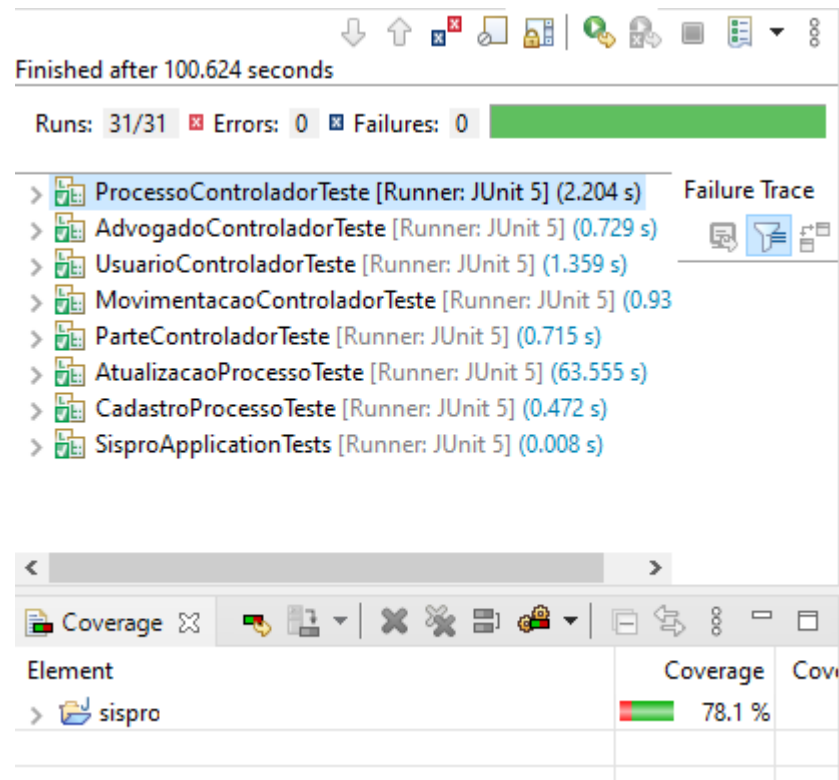
**Fonte:** Autoria própria.

## 5.2 Testes

Para fins de testar a lógica utilizada pelo método de *TDD* que foi citado e explicado na Seção 4.3, foram realizados testes de cobertura de código, que tem como finalidade medir a proporção do código exercido por testes automatizados (JetBrains, 2022).

Através dele foi possível obter os resultados transmitidos pela Figura 22, na qual pode-se observar que 78,1% do código foi coberto apenas utilizando os métodos de *TDD* para o desenvolvimento do sistema.

**Figura 22.** Testes de cobertura de código.



**Fonte:** Autoria própria.

## Capítulo 6

### 6 Considerações finais

O presente trabalho, teve como objetivo apresentar o processo de desenvolvimento do SISPRO, um sistema *mobile* que permite melhorar a comunicação entre um escritório e seus possíveis clientes, além de simplificar para usuários comuns as complexas palavras utilizadas pelo meio jurídico, podendo assim dar mais agilidade e qualificação ao escritório.

Durante as fases de levantamento bibliográfico e de requisitos, viu-se a necessidade de uma entrevista com possíveis usuários do sistema e com isso atender as suas reclamações e dificuldades do seu cotidiano. Partindo deste ponto, iniciou-se o desenvolvimento do SISPRO, um sistema de consultas processuais que, por meio da comunicação do sistema e da *API* do Tribunal, torna possível realizar extração, processamento e tratamento dos dados de processos. Tal sistema é decorrente de um minucioso estudo a respeito da experiência do usuário, e visou deixar nítida a informação que está sendo fornecida, além de aumentar engajamento, conversão e retenção em cada palavra.

Ademais é importante ressaltar que esse projeto possui uma ameaça a validade, em razão de a entrevista ter sido realizada com apenas um escritório jurídico. Embora a entrevista tenha contribuído bastante para o entendimento do problema, a realização de mais entrevistas com outros escritórios poderia trazer informações e ideias adicionais e até diferentes das que foram encontradas com o primeiro escritório.

Como trabalhos futuros, pretende-se testar a facilidade de uso da interface; bem como incluir mais um ambiente para acesso ao sistema sendo ele o ambiente *web*, para agregar mais valor ao produto; também percebeu-se a necessidade de adicionar algumas ferramentas que ajudariam os advogados a atender melhor suas demandas, sendo eles um alerta de prazos e atualização do sistema e um canal de comunicação por texto entre o cliente e seu advogado, assim garantindo a qualidade de suas entregas e mantendo cada vez mais ágil a comunicação entre eles.

## Referências Bibliográficas

Alura, **REST: Conceito e fundamentos**. Disponível em: <<https://www.alura.com.br/artigos/rest-conceito-e-fundamentos>>. Acesso em: 24 nov. 2022.

BECK, K. **Programação extrema (XP) explicada: acolha as mudanças**. Tradução Adiana P.S. Machado e Natália N.P.Lopes. Porto Alegre: Bookman, 2004.

BARRA, Daniela C. C.; PAIM, Sibele M. S.; SASSO, Grace T. M. D.; *et al.* **MÉTODOS PARA DESENVOLVIMENTO DE APLICATIVOS MÓVEIS EM SAÚDE: REVISÃO INTEGRATIVA DA LITERATURA**. v. 26, n. 4, 2017. Disponível em: <<https://www.scielo.br/j/tce/a/M3ZvQ3YrvbBb4p7n749JwLv/>>. Acesso em: 23 nov. 2022.

BRITO, Hugo J. B.; **JavaScript para aplicações móveis**. 2019. Disponível em: <<https://comum.rcaap.pt/handle/10400.26/36451>>.

CONSELHO NACIONAL DE JUSTIÇA (CNJ), **Resolução nº 185/2013**, Disponível em: <<https://atos.cnj.jus.br/atos/detalhar/1933>>. Acesso em: 24 nov. 2022.

DevMedia, **Introdução ao MySQL: Principais características desse SGBD**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-mysql/27799>>. Acesso em: 22 nov. 2022.

DevMedia, **Introdução ao Padrão MVC: Primeiros passos na Arquitetura MVC**. Disponível em: <<https://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>>. Acesso em: 24 nov. 2022.

Expo, **Expo**. Disponível em: <<https://expo.dev/>>. Acesso em: 24 nov. 2022.

FILHO, Júlio de M. UNIVERSIDADE ESTADUAL PAULISTA. p. 16, 2016 .Disponível em: <<http://hdl.handle.net/11449/141446>>. Acesso em: 7 ago. 2022.

Github. **The top programming languages**. Disponível em: <<https://octoverse.github.com/2022/top-programming-languages>>. Acesso em: 24 nov. 2022.

HOBBSAWM, Eric J. **A Era das Revoluções 1789-1848**. Rio de Janeiro: Paz e Terra, 2014, p. 79.

JetBrains, **O que é cobertura de código? Uma introdução | Guia de CI/CD do TeamCity**. Disponível em: <[What is Code Coverage? An Introduction | TeamCity CI/CD Guide | JetBrains](#)>. Acesso em: 24 nov. 2022.

Podmajersky, Torrey. **Redação Estratégica para UX**. São Paulo: Novatec editora LTDA. 2019.



MEIRELLES, Fernando S. **Uso da TI nas Empresas - Panorama e Indicadores**. p. 136, 2021. Disponível em: <<http://hdl.handle.net/11449/141446>>. Acesso em: 7 ago. 2022.

Meta, **React Native · Learn once, write anywhere**. Disponível em: <<https://reactnative.dev/>>. Acesso em: 24 nov. 2022.

Red Hat. **O que é API REST ?**. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>. Acesso em: 24 nov. 2022.

Red Hat. **O que é API?** Disponível em:<<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>>. Acesso em: 24 nov. 2022.

SAMPAIO, Juliana. **Spring Boot: como começar. Zup**. Disponível em: <<https://www.zup.com.br/blog/spring-boot>>. Acesso em: 22 nov. 2022.

TANENBAUM, Andrew; WETHERALL, David; **Computer Networks**, 5th ed. London, United Kingdom; Pearson, 2010, p. 4, 683-684.

Tribunal Regional do Trabalho (TRT), **Movimentação Processual**. Disponível em: <<https://www.trt4.jus.br/portais/trt4/movimentacao-processual>>. Acesso em: 24 nov. 2022.

Tribunal Supremo Eleitoral (TSE), **Processo Judicial Eletrônico (PJe)**. Justiça Eleitoral. Disponível em: <<https://www.tse.jus.br/servicos-judiciais/processos/processo-judicial-eletronico/processo-judicial-eletronico-pje-3>>. Acesso em: 24 nov. 2022.

VALENTIM, Natasha; SILVA, Williamson; CONTE, Tayana. Evaluating the user experience and the usability of a mobile web application: An experience report. **CIBSE 2015 - XVIII Ibero-American Conference on Software Engineering**, p. 788–801, 2015. Disponível em: <[https://www.researchgate.net/profile/Natasha-Valentim/publication/281874420\\_Evaluating\\_the\\_user\\_experience\\_and\\_the\\_usability\\_of\\_a\\_mobile\\_web\\_application\\_An\\_experience\\_report/links/569d121008ae78356e563e9e/Evaluating-the-user-experience-and-the-usability-of-a-mobile-web-application-An-experience-report.pdf](https://www.researchgate.net/profile/Natasha-Valentim/publication/281874420_Evaluating_the_user_experience_and_the_usability_of_a_mobile_web_application_An_experience_report/links/569d121008ae78356e563e9e/Evaluating-the-user-experience-and-the-usability-of-a-mobile-web-application-An-experience-report.pdf)>. Acesso em: 07 ago. 2022.

# Apêndice

## A.1 Questionário

1. O que você precisa que seja informatizado/automatizado?

O Atendimento de prestar informações sobre os processos aos clientes.

2. Por que você precisa do sistema?

Para automatizar o atendimento e reduzir custos com funcionários.

3. Como o sistema pode melhorar a sua realidade (suas atividades ou de outras pessoas)?

Além de melhorar a satisfação dos clientes, auxilia a secretária e reduz a folha de funcionários do escritório.

4. Há alguém que você indique para que conversemos sobre mais detalhes de como deverá ser o sistema?

Kehilton Gondim (Advogado Sócio), Janaína (Secretária).

5. Existe algum sistema similar no qual deveríamos nos basear? No que o meu sistema se diferencia dos já existentes?

Atualmente os tribunais possuem um sistema chamado PJe (Processo Judicial Eletrônico) que fornece uma consulta pública.

O diferencial do novo sistema é que ele fornecerá as informações de uma forma que o cliente seja capaz de assimilar, de uma forma menos técnica.

6. Quem utilizará o sistema?

Secretárias, Clientes.

7. Qual o nível de experiência dos usuários do sistema?

Baixa, tanto os clientes quanto as secretárias.

8. Do ponto de vista da experiência do cliente é melhor que as funcionalidades de consulta processual sejam disponibilizadas através de um aplicativo ou uma página web?

Aplicativo.

9. Há alguma funcionalidade adicional que gostaria de acrescentar ?

Atualmente o escritório conta com um sistema que busca do tribunal as informações necessárias para manter os advogados atualizados dos

processos e prazos, seria interessante que o sistema pudesse também possuir essa funcionalidade.

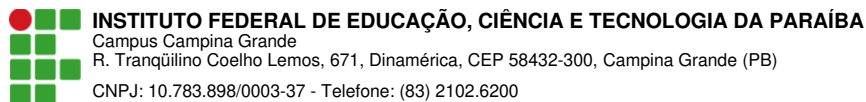
## A.2 Requisitos funcionais e não funcionais identificados

Identificamos os requisitos na tabela abaixo logo após a entrevista com o cliente e o registro das informações para a análise.

Esses são os requisitos oriundos dessa sessão de levantamento de requisitos, no caso, a entrevista com o cliente.

ID	REQUISITO
RF01	Autenticação de clientes
RF02	Cadastro de clientes
RF03	Consulta processual
RF04	Interface Mobile
RF05	Tratamento de dados vindos da API
RF06	Adicionar informações no processo
RF07	Detalhar processos
RF08	
RF09	
RF10	
RF11	

RNF01	Experiência do usuário
RNF02	Armazenar dados em cache para otimização
RNF03	Banco em modelo relacional e cloud
RNF04	
RNF05	



## Documento Digitalizado Ostensivo (Público)

### Versão final do TCC

**Assunto:** Versão final do TCC  
**Assinado por:** Eduardo Donato  
**Tipo do Documento:** Tese  
**Situação:** Finalizado  
**Nível de Acesso:** Ostensivo (Público)  
**Tipo do Conferência:** Cópia Simples

Documento assinado eletronicamente por:

- **Eduardo Bruno de Almeida Donato, ALUNO (201811250056) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE**, em 26/12/2022 23:55:20.

Este documento foi armazenado no SUAP em 26/12/2022. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 705426  
Código de Autenticação: f87f521a2d

