

## Documentação do Sistema

### SUMÁRIO

<a href="#">Dados do Cliente</a>	<a href="#">2</a>
<a href="#">Equipe de Desenvolvimento</a>	<a href="#">3</a>
<a href="#">1. Introdução</a>	<a href="#">4</a>
<a href="#">2. Objetivo</a>	<a href="#">5</a>
<a href="#">3. Escopo</a>	<a href="#">6</a>
<a href="#">4. Backlogs do Produto</a>	<a href="#">7</a>
<a href="#">5. Cronograma</a>	<a href="#">8</a>
<a href="#">6. Materiais e Métodos</a>	<a href="#">9</a>
<a href="#">7. Resultados</a>	<a href="#">10</a>
<a href="#">8. Conclusão</a>	<a href="#">11</a>
<a href="#">9. Homologação do MVP junto ao cliente</a>	<a href="#">12</a>
<a href="#">10. Divulgação</a>	<a href="#">13</a>
<a href="#">11. Carta de Apresentação</a>	<a href="#">15</a>
<a href="#">12. Carta de Autorização</a>	<a href="#">16</a>
<a href="#">13. Relato individual do processo</a>	<a href="#">18</a>

## Dados do Projeto/Cliente

**Título do Projeto:** CONTROLADOR SAJ - Sistema de Agendamento Jurídico

**Cliente:** Somenzari advocacia

**CNPJ/CPF:** 266.471.468-65

**Contato:** Camila Forte

**Email do contato:** camila.forte@yahoo.com

## Equipe de Desenvolvimento

Nome completo	Curso	Disciplina
Rafael Meira de Oliveira	ADS	Programação Orientada a Objetos em Java
Matheus Correia de Oliveira	ADS	Programação Orientada a Objetos em Java
Vinicius Pereira da Silva	ADS	Programação Orientada a Objetos em Java
Matheus Vinicius Lacerda da Silva	ADS	Programação Orientada a Objetos em Java
Arthur somenzari forte leone	ADS	Programação Orientada a Objetos em Java

Professor Orientador
Kesede Rodrigues Julio

## 1. Introdução

O escritório Somenzari Advocacia enfrenta dificuldades no controle centralizado de agendamentos de reuniões com clientes e prazos relacionados aos processos jurídicos. O uso de anotações manuais e ferramentas não integradas causa conflitos de horário, perda de informações e retrabalho.

Para resolver essa necessidade, será desenvolvido o CONTROLADOR SAJ — Sistema de Agendamento Jurídico, que reúne em um único ambiente o cadastro de clientes, processos e os agendamentos dos dois advogados do escritório. O sistema será implementado em Java (Spring Boot), com persistência em PostgreSQL, autenticação via JWT e entregas guiadas pela metodologia ágil (sprints de 2 semanas).

O sistema será desenvolvido em Java (Spring Boot), com banco de dados PostgreSQL, autenticação simples e acesso restrito aos dois advogados cadastrados. Impacto: o sistema permitirá melhor organização dos agendamentos e dos atendimentos, reduzindo falhas e aumentando a eficiência do escritório.

## 2. Objetivo

Automatizar e centralizar o controle de agendamentos jurídicos do escritório Somenzari Advocacia, permitindo o cadastro de clientes, processos e a marcação de compromissos com os advogados, evitando conflitos de horários e garantindo mais organização no fluxo de atendimentos.

### 3. Escopo

#### **Será implementado:**

- Módulo de agendamentos: criar, editar e cancelar compromissos.
- Cadastro de clientes e processos.
- Visualização da agenda dos dois advogados.
- Autenticação para acesso ao sistema (somente os advogados).

#### **Não será implementado:**

- Sistema de salas.
- Envio de e-mail ou SMS.
- Notificações automáticas.
- Exportação de dados (CSV/PDF).
- Acesso para recepção ou terceiros.
- Integrações externas.

## 4. Backlogs do Produto

### 1. Autenticação e Acesso

- **Descrição:** Login para os dois advogados credenciados.
- **Critério:** Apenas usuários válidos acessam o sistema.

### 2. CRUD de Clientes

- **Descrição:** Cadastrar, editar, listar e remover clientes.
- **Critério:** Validação de CPF/CNPJ e busca por nome.

### 3. CRUD de Processos

- **Descrição:** Cadastrar processos vinculados a clientes.
- **Critério:** Cada processo deve estar associado a um cliente.

### 4. Agendamentos

- **Descrição:** Criar, editar e cancelar agendamentos indicando cliente, processo, advogado, data e horário.
- **Critério:** O sistema deve impedir conflitos de horário entre os dois advogados.

### 5. Visualização da Agenda

- **Descrição:** Exibir a agenda diária/semana/mensal de cada advogado.
- **Critério:** Os compromissos devem aparecer corretamente conforme data e hora.

## 5. Cronograma

<b>Sprint</b>	<b>Entregas principais</b>
1	Setup, autenticação, CRUD de advogados
2	CRUD de clientes e processos
3	Implementação dos agendamentos e validação de conflitos
4	Tela de calendário e listagem da agenda
5	Testes finais, ajustes e documentação
6	Homologação e entrega final



## 6. Materiais e Métodos

a. **Modelagem do sistema:** A modelagem do sistema foi desenvolvida de forma simples e direta, considerando apenas os elementos necessários para o funcionamento do agendamento jurídico. As entidades principais são:

- **User (Advogado):** representa os dois advogados que terão acesso ao sistema e poderão visualizar, criar e gerenciar seus próprios agendamentos.
- **Client:** armazena as informações dos clientes atendidos pelo escritório.
- **Process:** representa os processos jurídicos vinculados a cada cliente, permitindo relacionar um agendamento a um processo específico.
- **Appointment:** entidade responsável por armazenar os agendamentos, contendo data, hora, advogado responsável, cliente envolvido e processo relacionado.

b. **Tecnologias utilizadas:**

- **Java 17:** linguagem principal utilizada para o desenvolvimento do backend.
- **Spring Boot 3:** framework responsável por fornecer estrutura para criação da API, gerenciamento de dependências e organização do projeto.
- **Spring Data JPA (Hibernate):** utilizado para o mapeamento objeto-relacional das entidades e acesso ao banco de dados.
- **PostgreSQL:** banco de dados relacional utilizado para armazenar clientes, processos, usuários e agendamentos.
- **Maven:** ferramenta para controle de dependências e build do projeto.
- **Spring Security + JWT:** responsável pela autenticação e controle de acesso dos dois advogados.
- **Git e GitHub:** utilizados para versionamento e gerenciamento colaborativo do código.

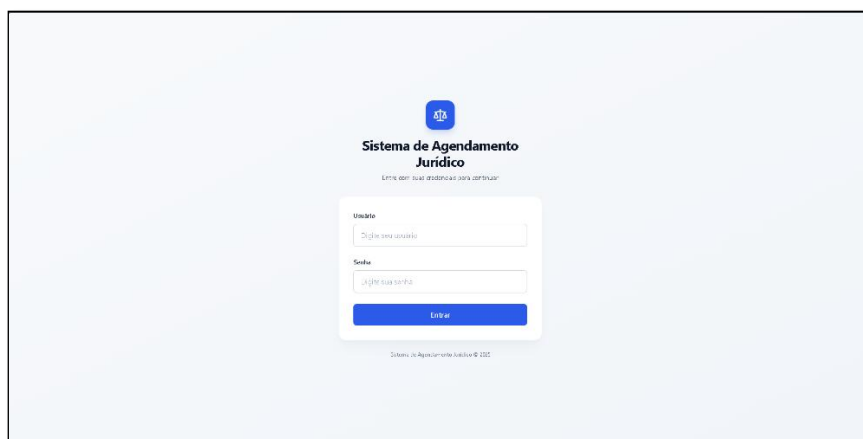
c. **Arquitetura do sistema:**

- **Camada de Apresentação (Frontend):** Responsável pelas telas de login, cadastro de clientes, processos, visualização da agenda e formulário de agendamento. Envia solicitações para a API utilizando HTTP.
- **API REST (Spring Boot):** Recebe as requisições do frontend, valida dados, aplica regras de negócio e coordena a comunicação com as demais camadas.
- **Camada de Serviço (Service Layer):** Contém a lógica de negócio do sistema, como validação de conflitos de horários entre os dois advogados, regras de cadastro e relacionamentos entre entidades.
- **Camada de Persistência (Repository):** Utiliza Spring Data JPA para salvar, buscar, editar e remover dados no banco PostgreSQL.
- **Banco de Dados (PostgreSQL):** Guarda todas as entidades: User, Client, Process e Appointment.

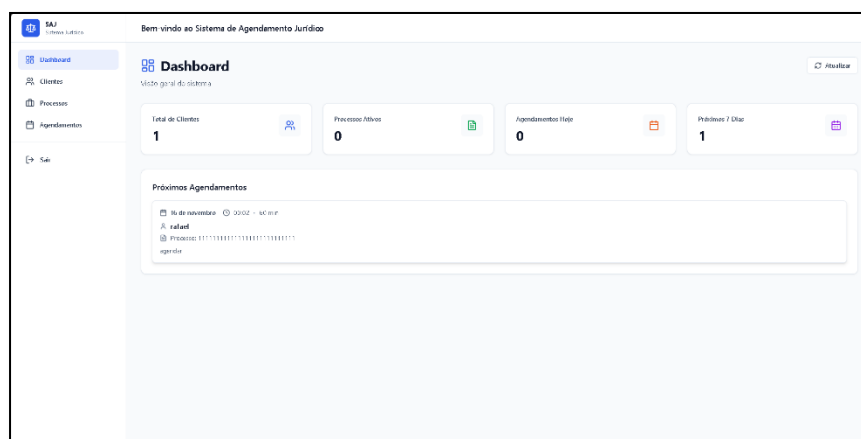
## 7. Resultados

### a. Protótipo:

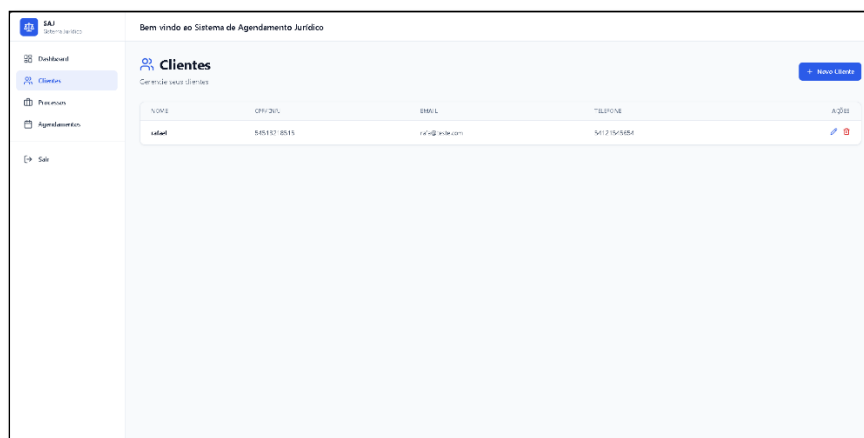
Tela de Login:



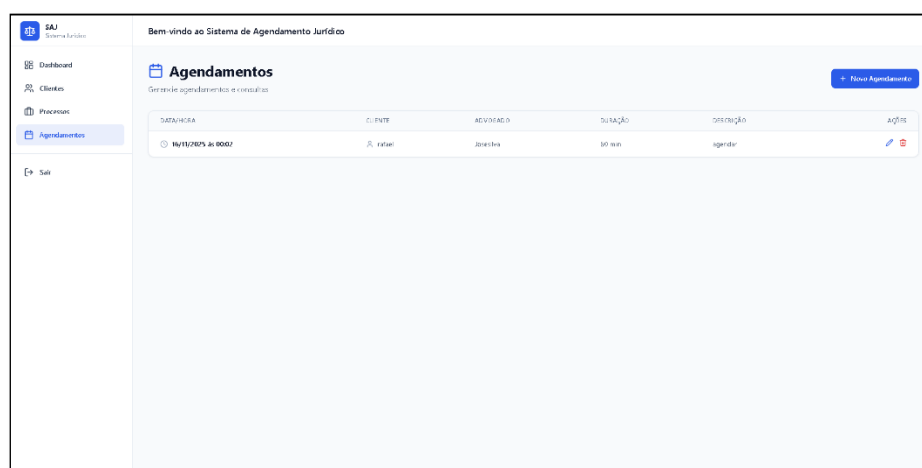
Dashboard:



Tela de cliente/processo:



Formulário de criação de agendamento + “Calendario” utilizado para consulta de agendamentos:



**b. Link do seu GitHub:**

<https://github.com/vinips04/Projeto-Sistema-Controlador-SAJ.git>

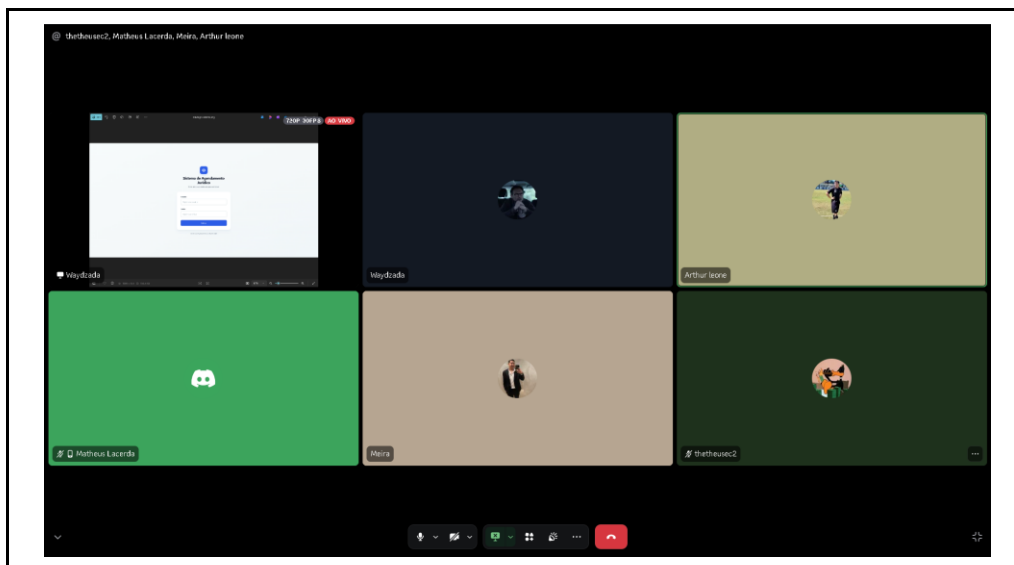
## 8. Conclusão

- a. **Impacto do sistema:** O CONTROLADOR SAJ trouxe mais organização ao fluxo de trabalho do escritório ao centralizar todos os agendamentos em um único sistema. Isso reduziu conflitos de horário entre os dois advogados, facilitou o acompanhamento dos compromissos e tornou o registro de clientes e processos mais claro e acessível. O sistema também contribuiu para diminuir perdas de informações e retrabalho, oferecendo mais controle e segurança no gerenciamento dos atendimentos jurídicos.
- b. **Melhorias Futuras:**
- Implementar relatórios internos para acompanhamento de atendimentos por período.
  - Adicionar lembretes internos dentro do próprio sistema.
  - Criar um painel mais detalhado da agenda, com filtros e estatísticas simples.
  - Desenvolver uma interface mobile para facilitar o acesso rápido dos advogados.

## 9. Homologação do MVP junto ao cliente

### a. Registro da Homologação

Após as entregas parciais, realizadas de acordo com os requisitos do sistema e cronograma, o MVP foi apresentado em uma reunião, realizada entre o time de desenvolvedores e o cliente.



Segue abaixo a lista de presentes na homologação do MVP.

Lista de presentes na Homologação
Rafael Meira de Oliveira
Matheus Correia de Oliveira
Vinicius Pereira da Silva
Matheus Vinicius Lacerda da Silva
Arthur Somenzari Forte Leone

Ao final da apresentação, o sistema foi homologado pelo cliente.

### b. Avaliação de Reação do Cliente:

O formulário abaixo foi enviado ao cliente para que ele pudesse avaliar todo o processo de construção do MVP.

<https://forms.gle/VNnN7G8jgQaPnejV9>

## 10. Divulgação

### Seminário de Projetos de Software

As fotos solicitadas abaixo, serão tiradas no dia da apresentação em sala de aula.

<foto 1: foto do time com o primeiro slide de fundo>	<foto 2: foto de um integrante apresentando o sistema.>
Da esquerda para direita: <legenda 1: descreva quem está na foto>	<legenda 2: coloque o nome de quem está apresentando>
<foto 3: foto plano geral da apresentação de frente para o fundo da sala>	<foto 4: foto plano geral da apresentação do fundo para a frente da sala>
Participantes do evento assistindo a apresentação	Participantes do evento assistindo a apresentação

Segue abaixo a lista de presentes na apresentação.

<b>Lista de presentes na Apresentação</b>
Rafael Meira de Oliveira
Matheus Correia de Oliveira
Vinicius Pereira da Silva
Matheus Vinicius Lacerda da Silva
Arthur Somenzari Forte Leone

## 11. Carta de Apresentação

Vimos por desta apresentar o grupo de acadêmicos do Centro Universitário Unimetrocamp, localizada à Rua Sales de Oliveira, 1661 - Campinas - SP, a fim de convidá-lo a participar de uma atividade extensionista associada ao componente curricular Programação Orientada a Objetos em Java sob responsabilidade do orientador Prof. Kesede Rodrigues Julio (profkesede64@gmail.com).

Em consonância ao Plano Nacional de Educação vigente, o Centro Universitário Unimetrocamp promove o Desenvolvimento de Software que, norteados pela metodologia de Gerenciamento Ágil Scrum, tem por princípios fundantes o diagnóstico dos problemas/demandas/necessidades, a participação ativa dos interessados/públicos participantes, a construção dialógica, coletiva e experiencial de conhecimentos, o planejamento de ações, o desenvolvimento e avaliação das ações, a sistematização dos conhecimentos, a avaliação das ações desenvolvidas.

Nesse contexto, a disciplina acima mencionada tem como principal escopo os temas relacionados à Programação Orientada à Objeto / Padrões de Projetos de Software, no que diz respeito ao desenvolvimento de um software utilizando Programação Orientada à Objeto.

Sendo assim, pedimos o apoio de Camila Forte para a realização das seguintes atividades: levantamento de requisitos, validação das entregas parciais, revalidação dos requisitos, homologação do MVP, ou qualquer outra intervenção que auxilie no desenvolvimento das competências de nossos acadêmicos e ao mesmo tempo possa contribuir para a comunidade em que estamos inseridos.

Aproveitamos a oportunidade para solicitarmos, em caso de aceite, que a parceria seja formalizada, mediante assinatura da Carta de Autorização, as atividades e informações que o(s) aluno(s) poderá(ão) ter acesso.

Em tempo, registramos ainda, o convite para a participação de todos os interessados no fórum semestral de acompanhamento e avaliação das atividades realizadas, que está previsto para o final deste semestre, e será comunicado previamente em convite específico.

Desde já nos colocamos à sua disposição para quaisquer esclarecimentos.

Atenciosamente,



Campinas, 23 de Setembro de 2025.



**Assinatura Direção Acadêmica da IES**



**Assinatura Docente**

## 12. Carta de Autorização

Eu, Camila Forte, Advogada, Somenzari advocacia, situada no endereço Rua Manoel Sylvestre de Freitas filho número 75, autorizo a realização das seguintes atividades acadêmicas do componente extensionista Programação Orientada a Objetos em Java, do Centro Universitário Unimetrocamp, sob orientação do Prof. Kesede Rodrigues Julio.

**Atividades:** Levantamento de requisitos, validação de entregas parciais, homologação do MVP, testes de usabilidade.

Conforme combinado em contato prévio, as atividades acima descritas são autorizadas para os seguintes alunos:

<b>Nome dos/das alunos/as</b>	<b>Curso</b>	<b>Matrícula</b>
Rafael Meira de Oliveira	ADS	Programação Orientada a Objetos em Java
Matheus Correia de Oliveira	ADS	Programação Orientada a Objetos em Java
Vinicius Pereira da Silva	ADS	Programação Orientada a Objetos em Java
Matheus Vinicius Lacerda da Silva	ADS	Programação Orientada a Objetos em Java
Arthur Somenzari Forte Leone	ADS	Programação Orientada a Objetos em Java

Declaro que fui informado por meio da **Carta de Apresentação** sobre as características e objetivos das atividades que serão realizadas na organização/instituição/empresa a qual represento e afirmo estar ciente de tratar-se de uma atividade realizada com intuito **exclusivo de ensino de alunos de graduação**, sem a finalidade de exercício profissional.

Desta forma, autorizo, em caráter de confidencialidade:

- o acesso a informações e dados que forem necessários à execução da atividade;
- o registro de imagem por meio de fotografias;
- outro:



Campinas, 23 de Setembro de 2025.

---

(Assinatura, nome completo do responsável, email de contato e com carimbo da empresa)

### 13. Relato individual do processo

**Vinicius Pereira da Silva**

Participei do levantamento de requisitos com o cliente, da modelagem do banco de dados e da criação das entidades principais do sistema. Durante o processo, aprendi mais sobre modelagem relacional e sobre o funcionamento do Spring Boot.

**Matheus Correia de Oliveira**

Fiquei responsável pela configuração inicial do projeto, integração com o banco de dados e desenvolvimento das APIs. Esse trabalho me ajudou a reforçar meus conhecimentos em Java e na estrutura do Spring Boot.

**Rafael Meira de Oliveira**

Trabalhei na implementação da funcionalidade de agendamentos, incluindo as regras de conflito de horário entre os advogados. Aprendi bastante sobre lógica de negócio e validação no backend.

**Matheus Vinicius Lacerda da Silva**

Atuei na implementação da autenticação do sistema e na estrutura das rotas e controles do backend. Com isso, aprimorei meu entendimento sobre organização de camadas, segurança e boas práticas no desenvolvimento de APIs.

**Arthur Somenzari Forte Leone**

Fui responsável pela documentação do projeto e pela organização das informações para a homologação com o cliente. Esse processo reforçou minha compreensão sobre clareza, comunicação e registro das etapas do desenvolvimento.