



CtrlDev: Plataforma de Tutoria Peer-to-Peer em Programação

Trabalho de Conclusão de Curso

Camila Vitória Oliveira dos Anjos

Manoel Carvalho Marques Neto
Orientador

Instituto Federal da Bahia - IFBA
Curso de Análise e Desenvolvimento de Sistemas
Campus Salvador

Salvador, Bahia, Brasil
20 de julho de 2025

Sumário

1	Visão Geral	1
1.1	Declaração do Problema	1
1.2	Proposta de Solução de Software	1
1.3	Tecnologias Adotadas	2
1.3.1	Back-end	2
1.3.2	Front-end	2
1.3.3	Banco de dados	3
1.3.4	Versionamento	3
1.3.5	Testes	3
1.4	Trabalhos Relacionados	3
1.4.1	Superprof	4
1.4.2	Profes	4
1.4.3	Alura	4
1.4.4	Tabela Comparativa	5
2	Requisitos	6
2.1	Requisitos Funcionais	6
2.2	Requisitos Não-Funcionais	6
3	Design	7
3.1	Projeto UML	7
3.1.1	Diagrama de Classe	7
3.1.2	Diagrama de Casos de Uso	7
3.2	Visão Arquitetural	8
3.3	Modelo de Banco de Dados	10
3.3.1	Modelo Lógico	10
3.3.2	Modelo Físico	10
4	Testes de Software	12
4.1	Projeto de testes	12
5	Implantação	14
5.1	Projeto de Implantação	14
6	Limitações	15
7	Manual do Usuário	16
	Agradecimentos	22
	Referências	23
	Glossário, Siglas e Abreviações	25

1 Visão Geral

1.1 Declaração do Problema

O ensino de programação, seja em cursos técnicos ou superiores, apresenta desafios significativos para estudantes em fase inicial. Muitos alunos enfrentam dificuldades para compreender conceitos fundamentais, como lógica, estruturas de dados e paradigmas de programação, o que resulta em altos índices de reprovação e evasão em disciplinas da área. Em contrapartida, alunos com maior frequência em atividades de monitoria tendem a alcançar um melhor desempenho acadêmico (Silva, 2021).

Nesse cenário, o aprendizado colaborativo surge como uma alternativa eficiente, permitindo que alunos mais experientes compartilhem conhecimento com colegas que enfrentam dificuldades. Essa abordagem é denominada tutoria entre pares. Conforme Kuo *et al.* (2022, p. 4, tradução própria):

A tutoria entre pares é um dos métodos de aprendizagem entre pares e uma estratégia derivada da tutoria; permite que estudantes com maior capacidade de aprendizagem ou mais experiência atuem como tutores, enquanto aqueles com menor capacidade de aprendizagem ou menos experiência sejam tutorados individualmente. Além disso, os estudantes com menor habilidade ou menos experiência podem reduzir a ansiedade em relação ao aprendizado ao aprimorar seus níveis de conhecimento por meio da aprendizagem mútua e do apoio entre pares.

Apesar dos benefícios, ainda há escassez de ferramentas que facilitem essa interação de maneira organizada e acessível. A maior parte das plataformas existentes é voltada para o ensino formal ou remunerado, ou não oferece suporte adequado às disciplinas ministradas, o que pode restringir o alcance entre os estudantes.

Essa carência limita o potencial de aprendizado ativo e o desenvolvimento de habilidades tanto dos mentores quanto dos mentorados, além de dificultar a organização e o gerenciamento eficiente dessas interações acadêmicas. Portanto, a criação de uma plataforma dedicada que viabilize essa comunicação direta, com funcionalidades adequadas para agendamento, avaliação e acompanhamento, torna-se relevante para fortalecer práticas educacionais colaborativas e promover o engajamento estudantil.

1.2 Proposta de Solução de Software

Diante do exposto, o projeto CtrlDev propõe oferecer um ambiente digital estruturado com o objetivo de conectar estudantes interessados em aprender e ensinar entre si, promovendo o compartilhamento de conhecimento de forma horizontal, gratuita e dinâmica.

A ferramenta busca, dessa forma, fortalecer a aprendizagem colaborativa em disciplinas voltadas à programação de computadores, auxiliando no desenvolvimento técnico e social dos alunos e tornando o aprendizado de programação mais acessível e efetivo.

1.3 Tecnologias Adotadas

A seleção das tecnologias para este projeto levou em conta critérios como compatibilidade com os requisitos, estabilidade e curva de aprendizado. Optou-se por tecnologias com maior potencial de escalabilidade, em detrimento de alternativas com menor capacidade de crescimento. A seguir, são apresentados os *softwares*, plataformas, linguagens e ferramentas utilizadas no desenvolvimento do CtrlDev.

1.3.1 Back-end

Java

O Java é uma linguagem de programação de propósito geral, fortemente tipada, orientada a objetos e multiplataforma, mantida pela Oracle Corporation [3]. Sendo uma plataforma que oferece portabilidade, segurança e alto desempenho, o Java é amplamente adotado em aplicações corporativas.

A escolha dessa linguagem justifica-se principalmente pela sua estabilidade e compatibilidade com diversas bibliotecas e *frameworks*, como o ecossistema Spring. Tais características permitem o desenvolvimento de soluções robustas, escaláveis e manuteníveis.

Spring

O ecossistema Spring oferece um conjunto de ferramentas e frameworks para Java, amplamente utilizados nas mais diversas aplicações [4]. Entre seus principais projetos, destacam-se o Spring Security, responsável por prover mecanismos de autenticação e controle de acesso altamente configuráveis, e o Spring AI, que fornece suporte à integração de soluções de inteligência artificial na aplicação.

Outros componentes suportam a integração com serviços externos, mensageria, ferramentas de automação e arquitetura de microsserviços. Além disso, o Spring fornece suporte a múltiplos bancos de dados por meio de *drivers* e módulos de persistência.

1.3.2 Front-end

Angular

Angular é um *framework* de código aberto desenvolvido pela Google para a criação de aplicações *web*, construído sobre TypeScript [5].

A escolha desta ferramenta deve-se principalmente à sua estrutura modular e abordagem baseada em componentes, que facilitam a organização e a manutenção do código. Adicionalmente, o TypeScript adiciona tipagem estática, detecção de erros em tempo de compilação e outros recursos avançados ao JavaScript [6], o que aumenta a segurança e a produtividade no desenvolvimento.

PrimeNG

O PrimeNG é uma biblioteca *open-source* que oferece uma ampla variedade de compo-

nentes de UI customizáveis, desenvolvida para aplicações baseadas em Angular [7].

Tailwind CSS

O Tailwind CSS é um *framework* utilitário de CSS que permite a construção de interfaces por meio da composição de classes diretamente no HTML [8]. Sua utilização agiliza o desenvolvimento e contribui para a padronização visual da aplicação, além de facilitar a criação de interfaces responsivas que se adaptam a diferentes tamanhos de tela.

1.3.3 Banco de dados

PostgreSQL

PostgreSQL é um Sistema Gerenciador de Banco de Dados (SGBD) de código aberto, reconhecido por sua robustez, confiabilidade e alto desempenho no gerenciamento de dados relacionais [9].

1.3.4 Versionamento

Git

O Git é um sistema de controle de versão distribuído e de código aberto. Utilizado por cerca de 96% dos desenvolvedores ao redor do mundo [10], o Git garante rapidez e eficiência, mesmo no armazenamento de histórico de aplicações de grande porte.

GitHub

GitHub é uma plataforma para hospedagem de código construída sobre o sistema Git. Além do controle de versão, também oferece recursos para gestão de projetos e monitoramento de vulnerabilidades [11].

1.3.5 Testes

JUnit

O JUnit é um *framework* de testes para Java e outras linguagens baseadas na JVM [12], amplamente utilizado para a criação e execução de testes automatizados. Sua utilização facilita a validação contínua do código e contribui para a manutenção da qualidade do software.

Bruno

Bruno é um cliente de API *open-source* com integração nativa ao Git. Desenvolvido com foco em eficiência e segurança, o aplicativo se mantém ágil e não realiza sincronização de dados com a nuvem [13].

1.4 Trabalhos Relacionados

O mercado atual conta com diversas plataformas de aulas particulares na área de tecnologia, todas com modelo comercial. Em contrapartida, o CtrlDev surge como uma alternativa

colaborativa e gratuita voltada para entre estudantes de programação. Esta seção apresenta um panorama dessas soluções, destacando funcionalidades, vantagens e limitações comparadas ao projeto proposto.

1.4.1 Superprof

Superprof é uma plataforma *online* que conecta alunos e professores particulares em diversas áreas do conhecimento [14]. O sistema oferece uma ampla variedade de disciplinas e níveis de ensino, permitindo que os tutores definam seus próprios preços, metodologia e modalidade. Para os alunos, o acesso às aulas requer o pagamento de uma taxa denominada Passe Aluno, além do valor cobrado individualmente por cada professor.

Vantagens

- Variedade de disciplinas e instrutores;
- Suporte a diferentes formatos de aula (presencial e remoto).

Limitações

- Custo elevado com taxas adicionais;
- Não há infraestrutura própria para o ensino de diferentes áreas.

1.4.2 Profes

Profes é uma plataforma brasileira de aulas particulares, em modalidade presencial ou *online*, considerada o maior portal de aulas particulares do Brasil [15]. Assim como o Superprof, o Profes oferece liberdade para o professor organizar o seu método de ensino. O sistema também inclui funcionalidades como envio de tarefas e suporte via inteligência artificial.

Vantagens

- Variedade de disciplinas e instrutores;
- Suporte a diferentes formatos de aula (presencial e remoto);
- Assistente virtual.

Limitações

- Custo elevado;
- Não há infraestrutura própria para o ensino de diferentes áreas.

1.4.3 Alura

Alura é uma escola *online*, cujo conteúdo é voltado para diversas áreas do desenvolvimento de *software* [16]. A plataforma disponibiliza fóruns de dúvidas, desafios práticos e atividades interativas. Diferentemente das plataformas de ensino particular, a Alura oferece

cursos e trilhas assíncronas.

Vantagens

- Conteúdo especializado em tecnologia e desenvolvimento de *software*;
- Trilhas organizadas para progressão de conhecimento;
- Fórum de comunidade ativo;
- Assistente virtual.

Limitações

- Custo elevado;
- Não há tutoria síncrona.

1.4.4 Tabela Comparativa

Características	Superprof	Profes	Alura	CtrlDev
Foco em programação	Não	Não	Sim	Sim
Tutoria síncrona	Sim (paga)	Sim (paga)	Não	Sim
Infraestrutura própria para ensino	Não	Parcial	Sim	Sim
Assistente virtual	Não	Não	Sim	Sim
Modalidade das aulas	Presencial e <i>online</i>	Presencial e <i>online</i>	<i>Online</i>	<i>Online</i>
Modelo de funcionamento	Comercial	Comercial	Comercial	Colaborativo
Custo	Pago	Pago	Pago	Gratuito

Tabela 1: Comparação entre plataformas de ensino e o projeto CtrlDev

2 Requisitos

2.1 Requisitos Funcionais

[RF01] Como tutorando, eu gostaria de favoritar os monitores com os quais mais me identifico, de modo que possa encontrá-los com facilidade mais tarde.

[RF02] Como tutorando, desejo agendar uma seção em um horário cômodo para mim e para o tutor.

[RF03] Como usuário, eu gostaria de receber atualizações de agendamento por e-mail para ser notificado sobre confirmações ou cancelamentos das minhas sessões de tutoria.

[RF04] Como tutorando, quero avaliar minhas experiências com cada tutor.

[RF05] Como tutorando, quero um assistente virtual para tirar dúvidas básicas de programação com respostas rápidas.

[RF06] Como usuário, desejo me conectar com o meu par através de chamada de vídeo integrada, para que possamos interagir visualmente durante as sessões.

[RF07] Como usuário, quero escrever e executar código diretamente na plataforma, para testar exemplos e resolver exercícios sem sair do ambiente de aprendizagem.

2.2 Requisitos Não-Funcionais

Código	Descrição	Atributo	Prioridade
RNF01	O sistema deve estar disponível em plataformas <i>web</i>	Compatibilidade	Essencial
RNF02	As senhas dos usuários devem ser armazenadas com criptografia <i>hash</i>	Segurança	Essencial
RNF03	O sistema deve garantir que apenas usuários autenticados possam acessar dados sensíveis	Segurança	Essencial
RNF04	As mensagens de erro e sucesso devem ser claras e objetivas	Usabilidade	Essencial
RNF05	O sistema deve oferecer suporte a leitura em áudio e tradução para Libras	Acessibilidade	Desejável
RNF06	O sistema deve possuir uma interface intuitiva, responsiva e amigável	Usabilidade	Desejável
RNF07	A API deve possuir documentação Swagger/OpenAPI	Documentação	Desejável

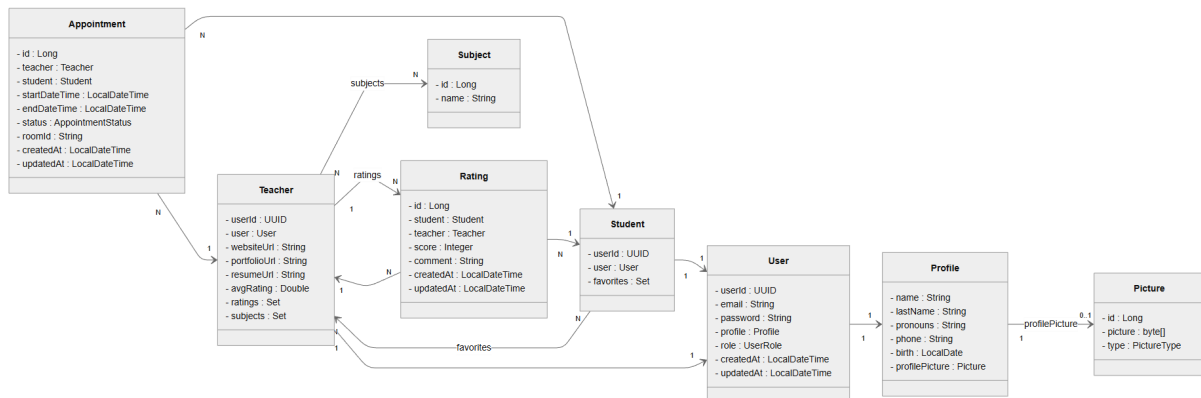
Tabela 2: Requisitos Não-Funcionais

3 Design

3.1 Projeto UML

3.1.1 Diagrama de Classe

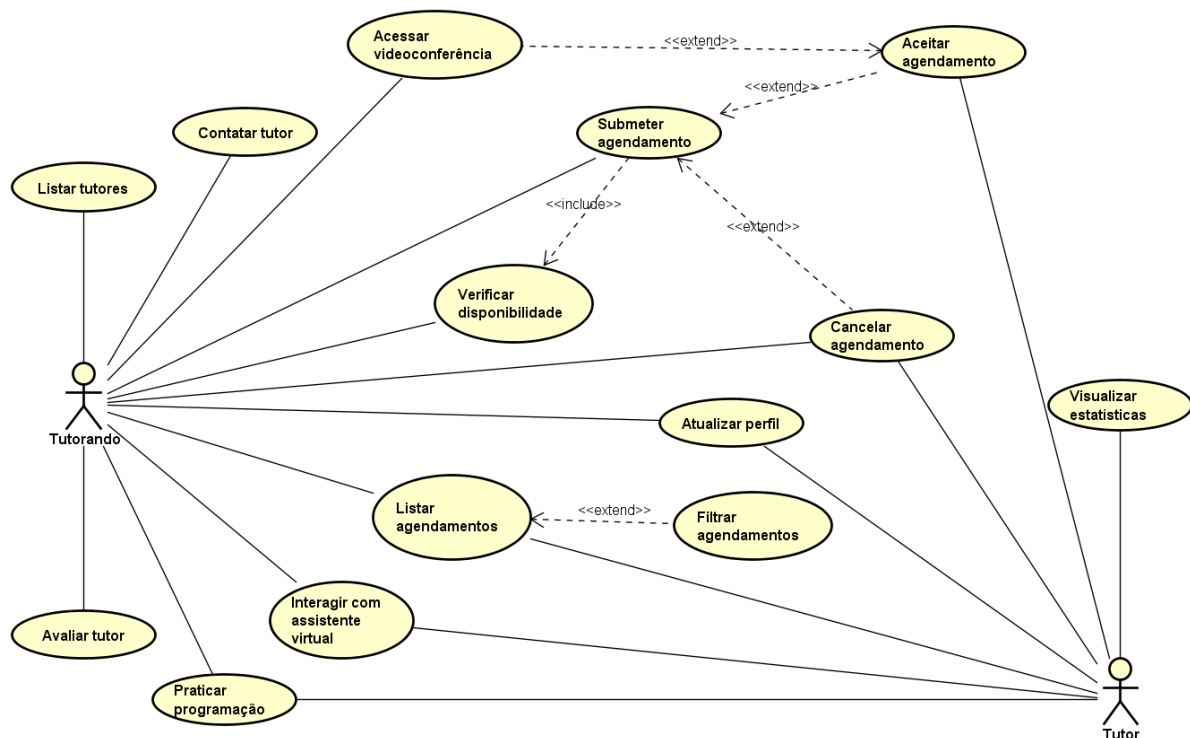
Figura 1: Diagrama de Classes



Fonte: Autoria própria

3.1.2 Diagrama de Casos de Uso

Figura 2: Diagrama de Casos de Uso



Fonte: Autoria própria

3.2 Visão Arquitetural

O CtrlDev adota uma arquitetura monolítica em camadas, que organiza responsabilidades e facilita a manutenção. Essa abordagem tradicional é amplamente utilizada em aplicações *web* devido à sua simplicidade e eficácia na estruturação do código.

A arquitetura é composta por uma aplicação *front-end* desenvolvida em Angular como Single Page Application (SPA), que oferece experiência fluida ao usuário e se comunica com uma API REST construída em Spring Boot. O *back-end* segue o padrão CSR (Controller-Service-Repository), em que os *controllers* gerenciam as requisições HTTP, os *services* implementam a lógica de negócio e os *repositories* cuidam da persistência no PostgreSQL. O *front-end* é organizado em componentes reutilizáveis, com serviços responsáveis pela lógica de negócio, enquanto um *proxy* gerencia o encaminhamento das requisições à API, centralizando o acesso e evitando problemas de CORS.

A plataforma utiliza RabbitMQ para processar notificações de e-mail de forma assíncrona em eventos de criação, confirmação ou cancelamento de agendamentos, isolando essas operações e evitando que falhas neste serviço comprometam as funcionalidades principais.

A API também integra Ollama, um modelo de AI *self-hosted* — isto é, executado localmente na própria infraestrutura — que responde a perguntas de programação. Enquanto o Spring AI atua como biblioteca de integração, sendo parte interna da API, o Ollama funciona como *container* independente, processando solicitações de forma autônoma.

O sistema utiliza ainda Quartz Scheduler [17] como componente interno da API para gerenciar tarefas agendadas, como a atualização do *status* dos agendamentos. O Quartz persiste os *jobs* no banco de dados, garantindo que mesmo em caso de reinício do serviço as tarefas programadas sejam retomadas. Além disso, a aplicação implementa Server-Sent Events (SSE) para enviar notificações em tempo real ao *front-end*, permitindo que os usuários sejam notificados sobre o início e o término das sessões de forma imediata (Listagem 1).

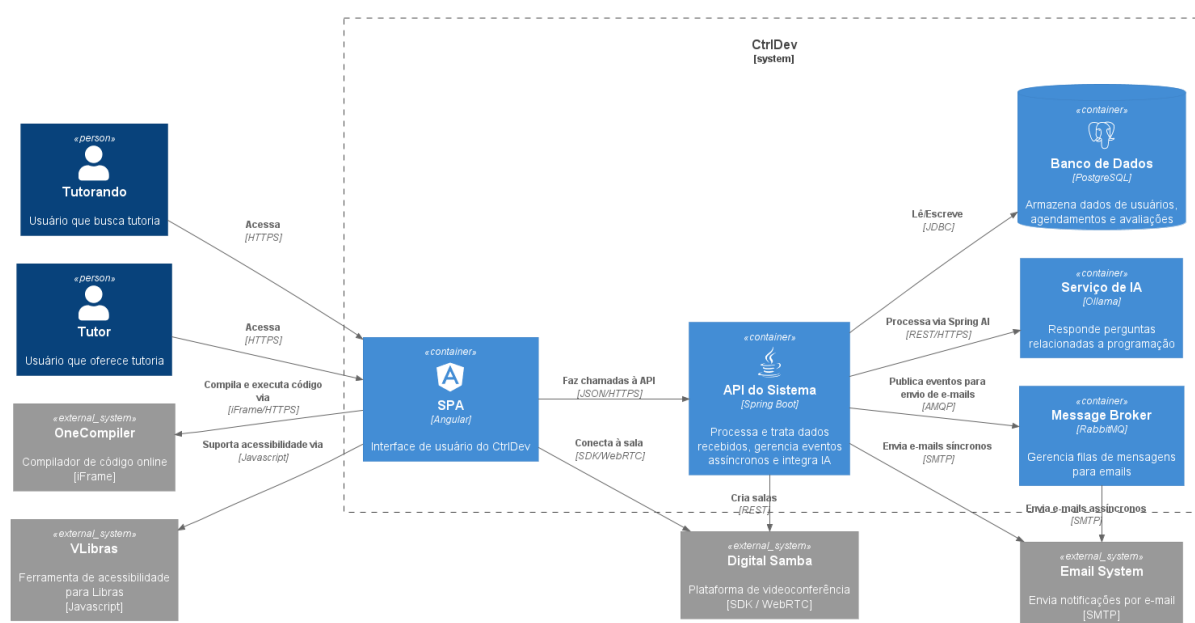
Listagem 1: Método que notifica usuários via SSE

```
1 public void notifyEvent(String eventType, Appointment appointment) {
2     List<UUID> userIds = List.of(appointment.getStudent().getUserId(),
3     appointment.getTeacher().getUserId());
4     AppointmentSSEDTO appointmentDTO = getAppointmentResponseDTO(appointment);
5     userIds.forEach(userId -> {
6         SseEmitter emitter = emitters.get(userId);
7         if (emitter != null) {
8             try {
9                 emitter.send(SseEmitter.event()
10                    .name(eventType)
11                    .data(appointmentDTO));
12             } catch (Exception e) {
13                 emitters.remove(userId);
14             }
15         }
16     });
17 }
```

Por fim, o CtrlDev integra serviços externos que complementam suas funcionalidades: o Digital Samba SDK [18] permite gerenciar aulas virtuais, o OneCompiler [19] possibilita a execução de códigos diretamente na plataforma por meio de *iframe*, garantindo isolamento e fácil integração, e o VLibras [20], disponibilizado como *widget*, oferece acessibilidade traduzindo conteúdos para Libras, assegurando que a aplicação atenda a diferentes necessidades dos usuários.

A seguir, o diagrama de *containers* C4 (Figura 3) ilustra a arquitetura do sistema, mostrando o fluxo de dados desde a interface do usuário até o banco de dados, incluindo a interação com serviços externos e o processamento assíncrono de e-mails mediado pelo *message broker*.

Figura 3: Diagrama de *containers* do CtrlDev



Fonte: Autoria própria

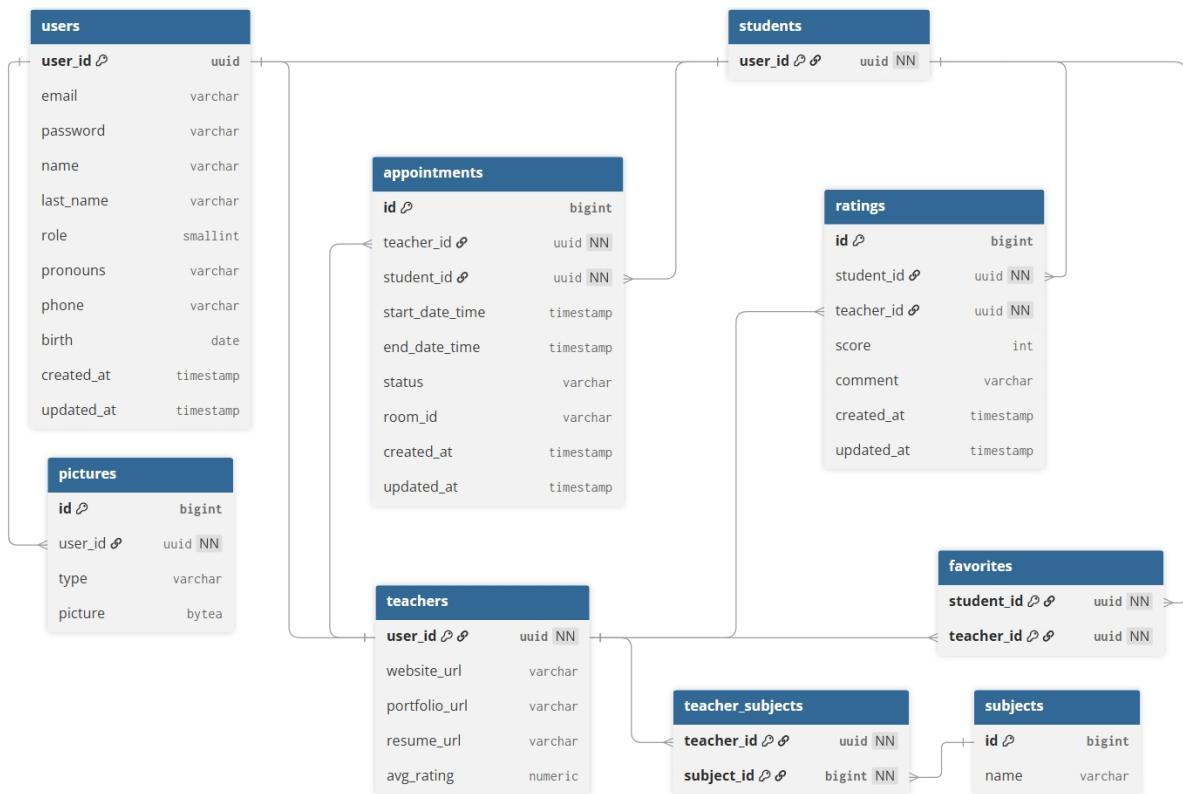
Embora algumas integrações dependam de *softwares* externos e/ou proprietários, o caráter modular da arquitetura permite que essas dependências sejam eventualmente substituídas sem comprometer a integridade ou o funcionamento das demais partes da plataforma.

Dessa forma, a escolha desta arquitetura justifica-se pela natureza e escopo do sistema CtrlDev. A arquitetura em camadas organiza responsabilidades, facilita a manutenção e a legibilidade do código, enquanto a comunicação assíncrona para e-mails garante um desempenho consistente e reduz o impacto de falhas isoladas. Além disso, a separação entre *front-end* e *back-end* permite um desenvolvimento contínuo e independente das duas partes de forma estruturada e segura, favorecendo a evolução contínua do sistema.

3.3 Modelo de Banco de Dados

3.3.1 Modelo Lógico

Figura 4: Modelo lógico do banco de dados



Fonte: Autoria própria

3.3.2 Modelo Físico

O modelo físico do banco de dados do sistema CtrlDev foi implementado no PostgreSQL. A seguir, serão descritas as tabelas que compõem o sistema, detalhando seus principais atributos, relacionamentos e a cardinalidade de cada associação.

- **USERS** – Esta tabela registra os dados comuns a todos os tipos de usuários, como nome, email, senha e outros dados pessoais. A separação da tabela **USERS** permite escalabilidade e facilita a manutenção, evitando duplicação de informações para cada tipo de usuário. Cada registro em **USERS** está associado a exatamente um registro em **STUDENTS** ou **TEACHERS**.
- **STUDENTS** – Contém informações específicas dos usuários que atuam como tutorandos, guardando uma referência para seu *user_id* na tabela **USERS**.
- **TEACHERS** – Armazena informações específicas dos usuários que atuam como tutores, também mantendo referência para seu *user_id* na tabela **USERS**. Além disso, mantém

uma média das avaliações recebidas `avg_rating`, e as colunas opcionais `website_url`, `portfolio_url` e `resume_url`.

- **PICTURES** – Armazena imagens de perfil como dados binários. Possui também uma referência para o `user_id` e uma coluna `tipo` para viabilizar a expansão do sistema e o armazenamento de outros tipos de imagem no futuro.
- **APPOINTMENTS** – Representa cada compromisso agendado entre um tutorando e um tutor. Possui `id` como chave primária, além de `student_id`, `teacher_id`, `date`, `time` e `status`. Cada appointment está associado a um *student* e um *teacher* (1,1), enquanto um *student* ou *teacher* pode ter múltiplos appointments distintos.
- **FAVORITES** – Nesta tabela é estabelecida a relação de muitos-para-muitos entre **STUDENTS** e **TEACHERS**, indicando quais tutores cada tutorando marcou como favorito. Possui `student_id` e `teacher_id` como chave primária composta, garantindo que cada par seja único.
- **RATINGS** – Registra avaliações feitas por tutorandos aos tutores. Contém `id` como chave primária, além de `student_id`, `teacher_id`, `score` e `comment`, permitindo que cada tutor receba várias avaliações e que cada tutorando avalie diversos tutores.
- **SUBJECTS** – Armazena categorias de ensino. É utilizada para indicar quais assuntos um tutor está disponível a ensinar.
- **TEACHER_SUBJECTS** – Relaciona *teachers* e *subjects*, em uma configuração de muitos-para-muitos, armazenando as categorias que um tutor ensina. Possui `teacher_id` e `subject_id` como chave primária composta, garantindo que cada par seja único.

Além das tabelas supracitadas, o Quartz Scheduler introduz novas tabelas ao banco de dados para registro e gerenciamento de *jobs* agendados. São elas:

- | | |
|-----------------------|----------------------------|
| • QRTZ_BLOB_TRIGGERS | • QRTZ_PAUSED_TRIGGER_GRPS |
| • QRTZ_CALENDARS | • QRTZ_SCHEDULER_STATE |
| • QRTZ_CRON_TRIGGERS | • QRTZ_SIMPLE_TRIGGERS |
| • QRTZ FIRED_TRIGGERS | • QRTZ_SIMPROP_TRIGGERS |
| • QRTZ_JOB_DETAILS | • QRTZ_TRIGGERS |
| • QRTZ_LOCKS | |

4 Testes de Software

4.1 Projeto de testes

Testes manuais

Foram conduzidos testes exploratórios na camada de *front-end* desenvolvida em Angular, com foco na experiência do usuário e usabilidade. A validação incluiu a navegação entre telas, preenchimento de formulários e *feedback* visual ao usuário. Além disso, foi possível avaliar a responsividade da interface em diferentes dispositivos, verificando a adaptação em telas de diferentes tamanhos. Esses testes ajudaram a identificar falhas e pontos de melhoria nos componentes visuais, direcionando os ajustes necessários para cumprir as expectativas de usabilidade em cenários realistas de uso.

Testes automatizados

Para a camada de lógica de negócio implementada em Spring Boot, foram desenvolvidos testes automatizados utilizando o *framework* JUnit. Os testes cobrem cenários críticos das funcionalidades, abrangendo tanto casos de sucesso quanto cenários de erro. Entre eles, incluem-se o fluxo de autenticação de usuários (Listagem 2), criação, confirmação e cancelamento de agendamentos, e a validação de situações de conflito de agenda (Listagem 3).

Listagem 2: Teste unitário de login de usuário

```
1 @Test
2 void shouldReturn200WhenLoginWithValidCredentials() {
3     Authentication auth = mock(Authentication.class);
4     Cookie cookie = new Cookie("jwt", "token");
5
6     when(authenticationManager.authenticate(any
7         (UsernamePasswordAuthenticationToken.class))) .thenReturn(auth);
8     when(auth.getPrincipal()).thenReturn(user);
9     when(tokenService.generateToken(user)).thenReturn("jwt-token");
10    when(cookieUtil.createJwtCookie("jwt-token")).thenReturn(cookie);
11    when(cookieUtil.createUserInfoCookie(user)).thenReturn(cookie);
12
13    ResponseEntity<UserSummaryDTO> result = controller.login(loginDTO,
14        response);
15
16    assertEquals(200, result.getStatusCodeValue());
17    verify(tokenService).generateToken(user);
18 }
```

Testes de API

A ferramenta Bruno foi utilizada para validação manual dos *endpoints* da API REST expostos pelo *back-end*. Através dela, foi possível testar individualmente cada rota, simulando diferentes tipos de requisições — em operações de criação, consulta, atualização e exclusão de dados — e analisando as respostas correspondentes. A prática permitiu verificar códigos de status HTTP, estruturas de dados retornadas e mensagens de erro em cenários de falha.

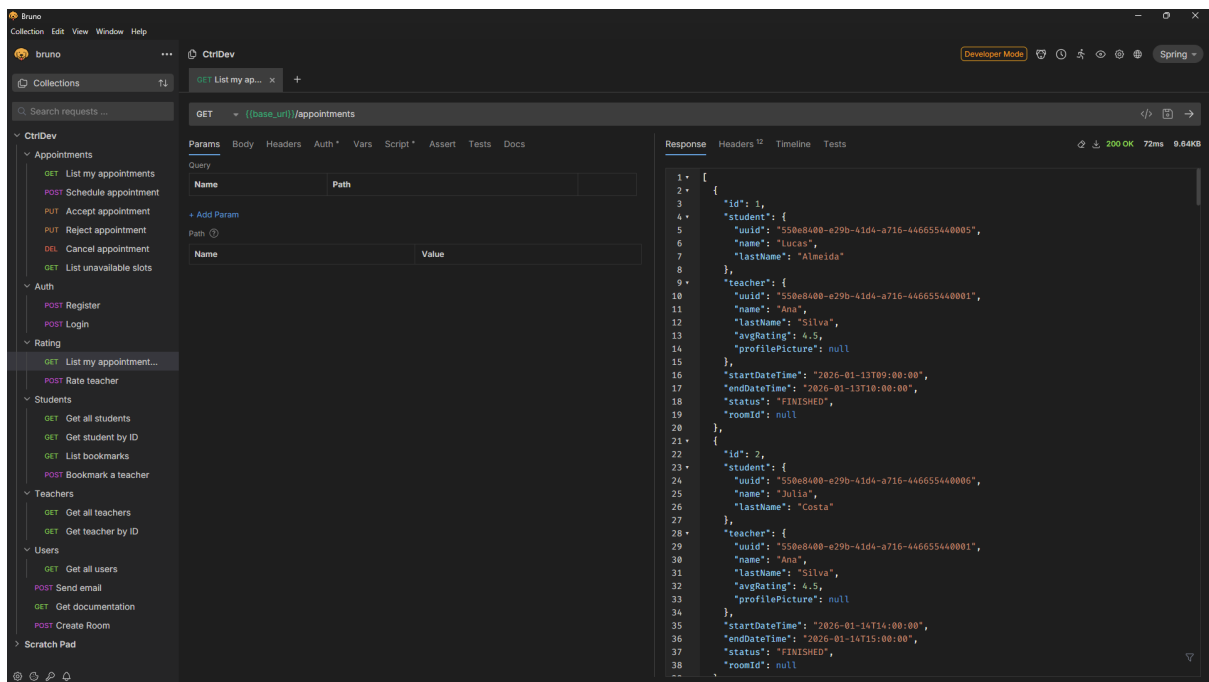
Listagem 3: Teste unitário para validação de conflitos de agenda

```

1 @Test
2 void shouldThrowScheduleConflictExceptionWhenAppointmentConflicts() {
3     when(studentService.findStudentByAuthentication(authentication))
4         .thenReturn(student);
5     when(teacherService.findTeacherByUUID(appointmentDTO.teacher()))
6         .thenReturn(teacher);
7     when(appointmentRepository.hasConflict(any(), any(), any(), any(), any()))
8         .thenReturn(true);
9
10    assertThrows(ScheduleConflictException.class, () ->
11        appointmentService.create(authentication, appointmentDTO));
12 }

```

Figura 5: Teste de listagem de agendamentos de um tutor



Fonte: Autoria própria

5 Implantação

5.1 Projeto de Implantação

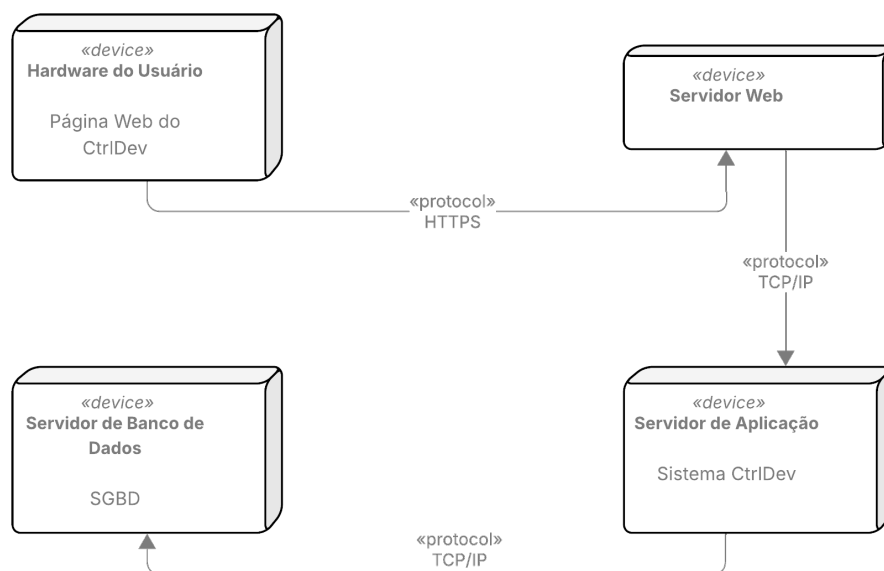
Desenvolvido como uma aplicação *web*, o CtrlDev é acessível por meio de navegadores modernos em múltiplos dispositivos. Sua utilização requer apenas um navegador atualizado com suporte a JavaScript e conexão de rede. Dessa forma, não é necessária a instalação de *software* adicional no cliente, o que simplifica o processo de implantação e acesso ao sistema.

A infraestrutura de servidor necessária inclui recursos de processamento, memória e armazenamento dimensionados de acordo com a carga esperada de usuários, entretanto, o modelo de Inteligência Artificial utilizado exige o mínimo de 4GB a 8GB de espaço livre para seu devido funcionamento.

O ambiente de implantação pode utilizar sistemas operacionais Linux ou Windows Server em versões recentes, exigindo a presença do JDK na versão 25 ou superior para a execução do *back-end* desenvolvido com Spring Boot. Além disso, a comunicação do sistema deve utilizar protocolos padrão da *web*, garantindo acesso seguro aos clientes por meio de HTTPS.

A arquitetura de implantação é apresentada na Figura 6, que ilustra o fluxo de comunicação do sistema:

Figura 6: Diagrama de Implantação



Fonte: Autoria própria

6 Limitações

Durante o desenvolvimento do CtrlDev, foram identificadas algumas limitações relacionadas principalmente à dependência de serviços externos e às restrições de infraestrutura, impactando diretamente o escopo e a escalabilidade da aplicação.

A proposta inicial do CtrlDev, concebida como um sistema democrático e gratuito, foi parcialmente comprometida pela integração da plataforma de videoconferência Digital Samba [18]. A ferramenta foi selecionada entre alternativas similares devido à sua robustez, facilidade de integração, suporte ao idioma português e funcionalidades adequadas ao contexto educacional, como *chat* e quadro branco. Contudo, a plataforma disponibiliza apenas dez mil minutos gratuitos mensais, limite que, com a expansão da aplicação e o aumento no número de usuários, tornaria inviável a continuidade do serviço sem custos adicionais. Considerando a cotação do euro a R\$6,25 em 23 de janeiro de 2026 (Investing.com, 2026), o custo estimado seria de aproximadamente R\$0,98 por aula após o esgotamento do limite gratuito, o que conflita com a proposta de gratuidade do sistema.

Outra limitação está relacionada à execução do modelo de inteligência artificial em ambientes com recursos computacionais restritos. Durante os testes, a utilização do modelo em contextos com memória limitada resultou em elevada latência, com tempos de resposta superiores a 50 segundos mesmo para tarefas simples. Essa limitação compromete a experiência do usuário e a escalabilidade da aplicação, sendo mitigável apenas por meio da ampliação de recursos computacionais ou da utilização de modelos mais leves. Uma alternativa seria o uso de serviços de AI disponibilizados por provedores comerciais, o que reduziria a dependência de infraestrutura local, mas também introduz custos financeiros recorrentes.

No que diz respeito à acessibilidade, o VLibras apresenta limitações na interpretação de conteúdos embutidos em aplicações *web* externas, como é o caso do OneCompiler e Digital Samba. Essas limitações decorrem da política de mesma origem (*Same-Origin Policy*) imposta pelos navegadores, que impede que *scripts* de uma aplicação acessem ou interajam com o conteúdo de componentes carregados a partir de domínios distintos. Como consequência, partes da aplicação que dependem de serviços externos não podem ser plenamente acessadas por meio dessa ferramenta.

Em síntese, essas limitações evidenciam desafios importantes para a escalabilidade, acessibilidade e experiência do usuário, servindo como base para decisões futuras sobre infraestrutura e serviços utilizados pelo CtrlDev.

7 Manual do Usuário

Autenticação

1. Cadastro (Figura 7)

- Selecione o tipo de conta: “Quero aprender (Tutorando)” ou “Quero ensinar (Tutor)”;
- Preencha os campos obrigatórios: nome, sobrenome, e-mail e senha;
- Preencha os campos opcionais: telefone, data de nascimento e pronomes;
- Clique em “Criar conta” para finalizar o cadastro.

2. Login (Figura 8)

- Digite e-mail e senha cadastrados nos campos correspondentes;
- Clique em “Entrar” para acessar a plataforma.

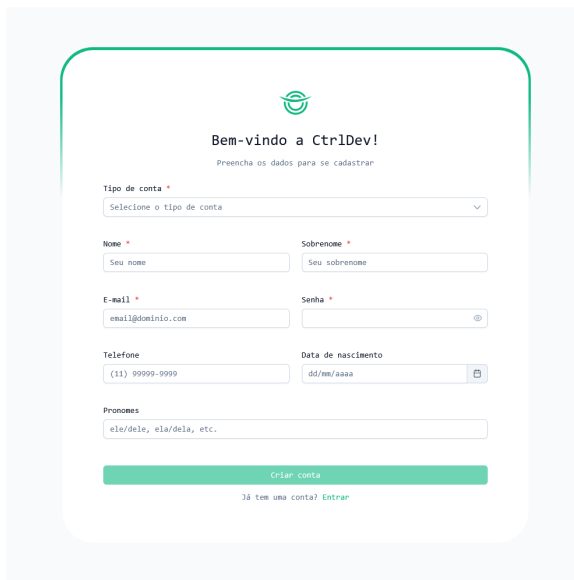
A interface de cadastro do CtrlDev apresenta o título "Bem-vindo a CtrlDev!" e o subtítulo "Preencha os dados para se cadastrar". O formulário contém campos para: "Tipo de conta" (menu suspenso), "Nome" e "Sobrenome" (campos separados), "E-mail" e "Senha" (campos com ícone de olho para alternar visibilidade), "Telefone" (com máscara (11) 99999-9999) e "Data de nascimento" (com máscara dd/mm/aaaa). Há também um campo para "Pronomes" com o exemplo "ele/dela, ela/dela, etc.". No final, há um botão verde "Criar conta" e um link "Já tem uma conta? Entrar".

Figura 7: Tela de cadastro

A interface de login do CtrlDev apresenta o título "Bem-vindo a CtrlDev!" e o subtítulo "Entre para continuar". O formulário contém campos para "E-mail" e "Senha", ambos com ícones de olho para alternar visibilidade. Há um link "Esqueci minha senha" e um botão verde "Entrar". No rodapé, há o texto "Não tem uma conta? Criar conta".

Figura 8: Tela de login

Busca de Tutores

1. Pesquisa (Figura 9)

- Digite o nome do tutor no campo “Buscar por nome”;
- Selecione as categorias desejadas no filtro “Categorias”;
- Clique no botão “Buscar” para aplicar os filtros;
- Clique no ícone de filtro com barra para limpar todos os filtros.

2. Favoritos (Figuras 10 e 11, (disponível apenas usuários sob tutoria))

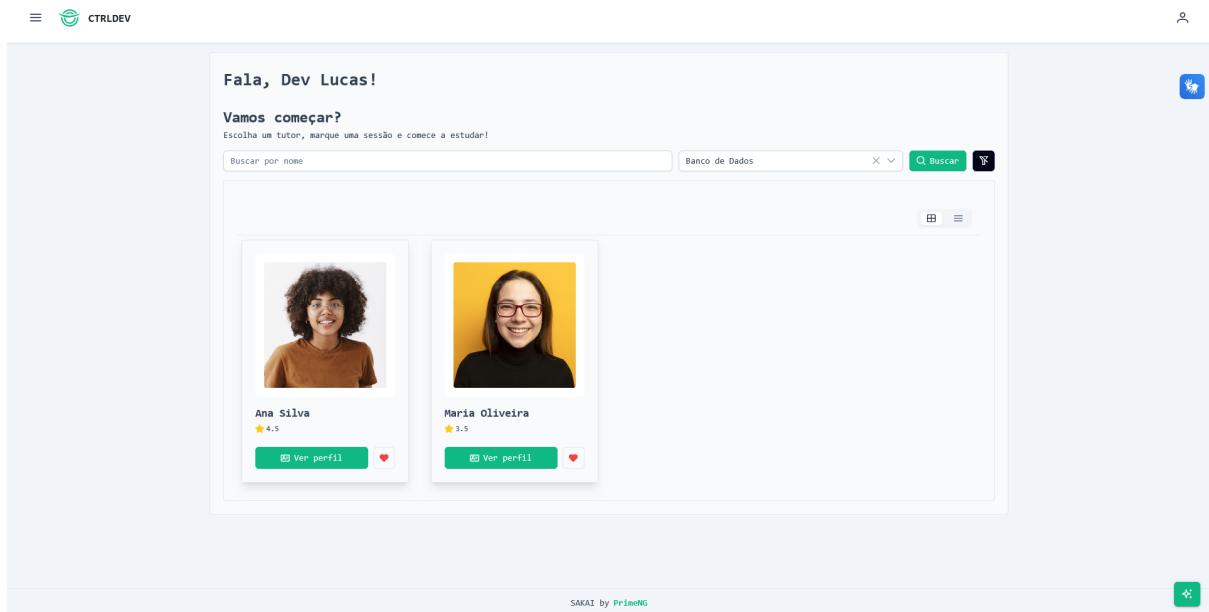


Figura 9: Filtragem por categoria

- Clique no ícone de coração para adicionar tutores aos favoritos;
- Para verificar tutores favoritados, acesse o menu “Favoritos” na barra lateral à esquerda;

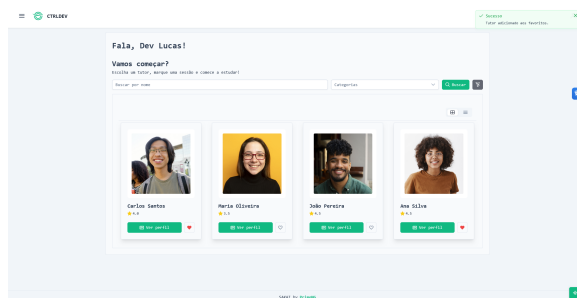


Figura 10: Adição de favorito

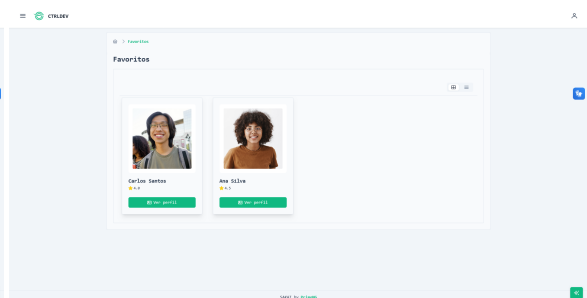


Figura 11: Lista de favoritos

Gestão de Aulas

1. Criar Agendamento (Figura 12, disponível apenas usuários sob tutoria)

- Acesse o perfil do tutor desejado e clique no botão “Agende uma aula”;
- Selecione uma data no calendário;
- Escolha um horário disponível nos blocos apresentados;
- Clique em “Salvar” para concluir a solicitação;
- Aguarde a confirmação do tutor para que a aula seja efetivada;

2. Gerenciar Agendamentos (Figura 13)

- Na barra lateral à esquerda, acesse o menu “Agendamentos”;
- Utilize os filtros de *status* para visualizar agendamentos específicos;
- Clique em “Confirmar” para aceitar um agendamento pendente (disponível apenas para usuários que oferecem tutoria);
- Clique em “Cancelar” para cancelar um agendamento;
- Clique em “Entrar” para acessar a sala de aula virtual;
- Clique em “Avaliar” para avaliar um tutor após uma aula finalizada (disponível apenas para usuários sob tutoria).

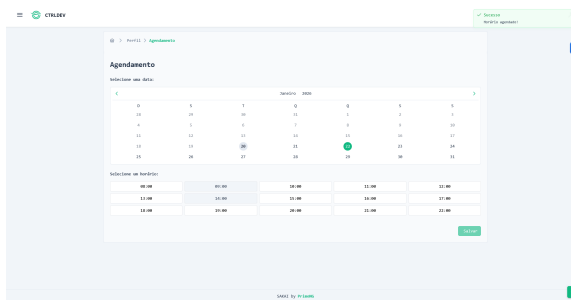


Figura 12: Criação de agendamento

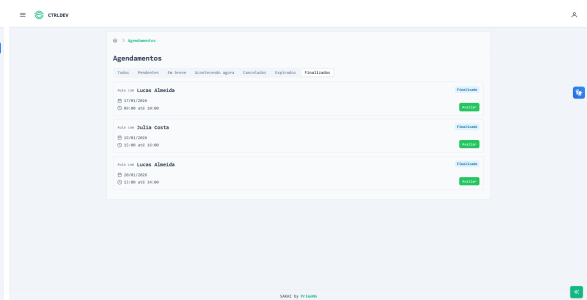


Figura 13: Agendamentos

Ambiente de Aprendizagem

1. Editor de código (Figura 14)

- Acesse a página “Praticar” através do menu lateral;
- Utilize o editor de código integrado para escrever programas;

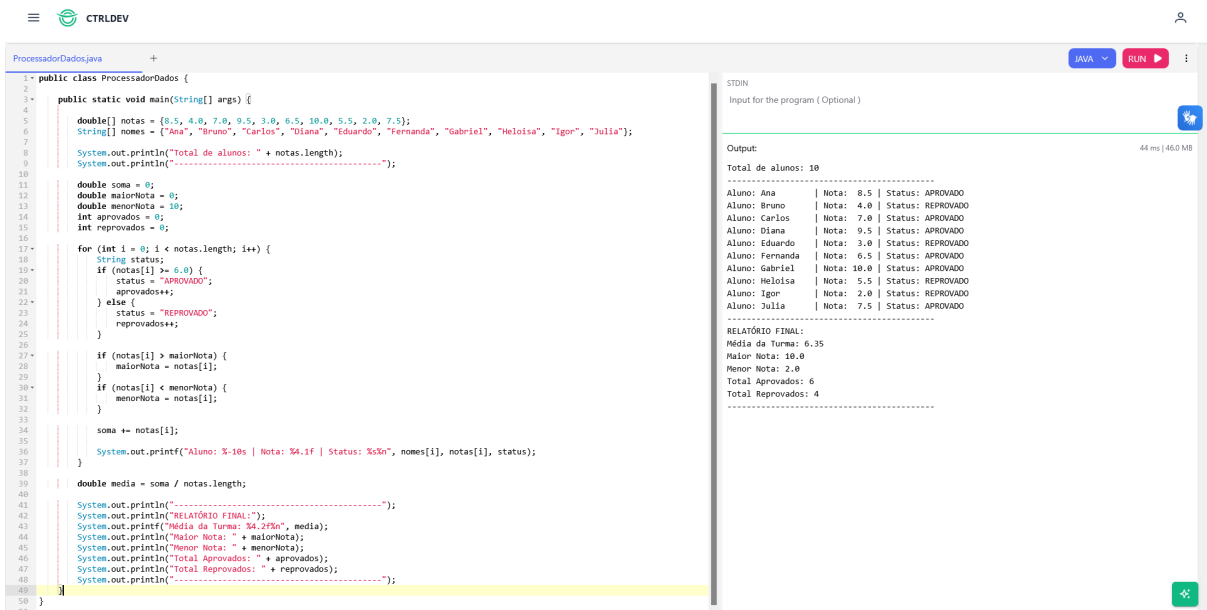


Figura 14: Editor de código

- Execute o código utilizando os controles do editor.

2. Assistente de AI (Figura 15)

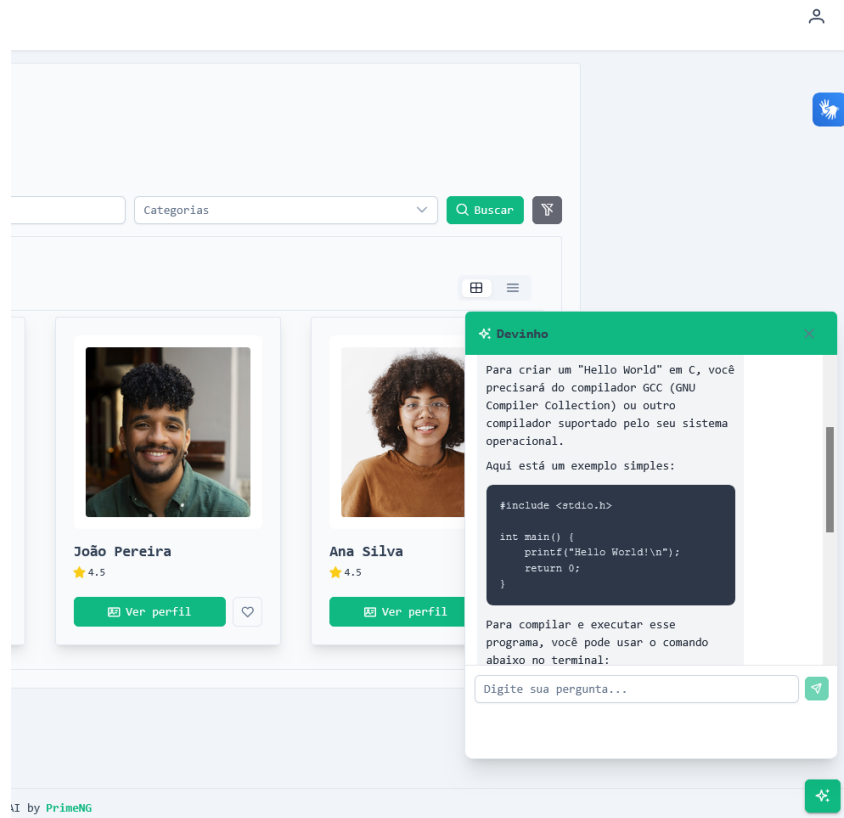


Figura 15: Assistente virtual

- Clique no botão flutuante no canto inferior direito;
- Digite sua pergunta sobre programação no campo de texto;
- Pressione Enter ou clique no botão de enviar e aguarde a resposta do assistente.
- Clique no botão flutuante no canto inferior direito;
- Digite sua pergunta sobre programação no campo de texto;
- Pressione Enter ou clique no botão de enviar e aguarde a resposta do assistente.

3. Sala de aula virtual (Figura 16)

- Clique em “Entrar” no agendamento iniciado ou na notificação para acessar a sala virtual.
- Utilize os controles de áudio e vídeo durante a aula.
- Para sair da sessão, clique em “Sair”. A aula será encerrada automaticamente após 1h de duração.

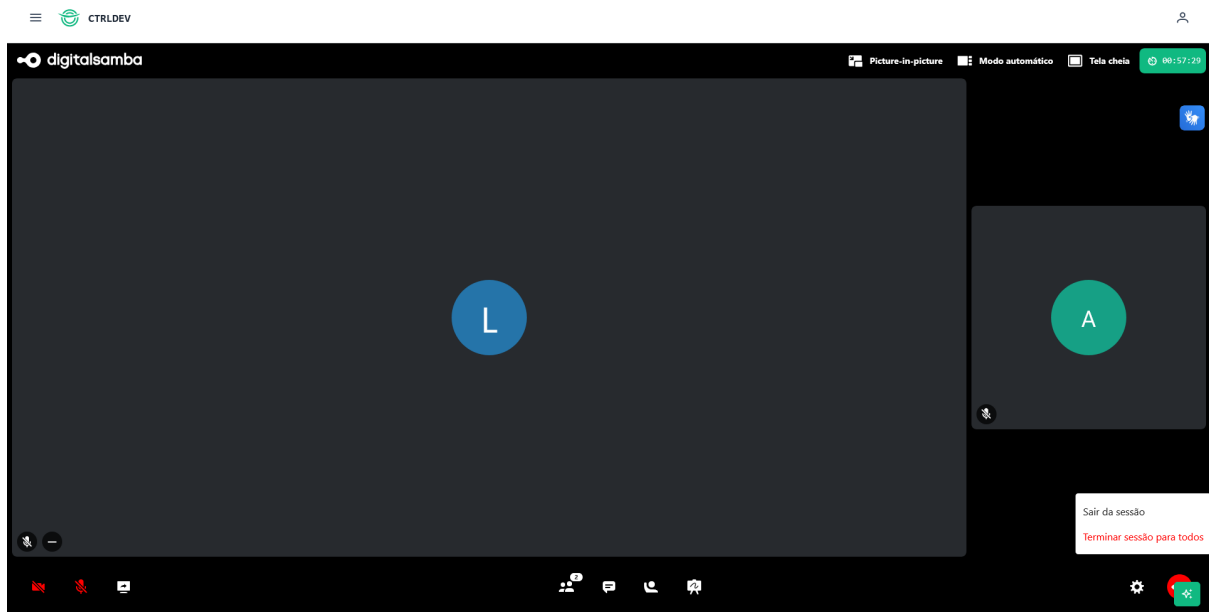


Figura 16: Sala de aula virtual

Usuário

1. Dashboard (Figura 17, disponível apenas para usuários que oferecem tutoria)

- Visualize o resumo das suas próximas aulas, estatísticas de ensino e avaliações;
- Acesse funcionalidades de gerenciamento através dos atalhos.

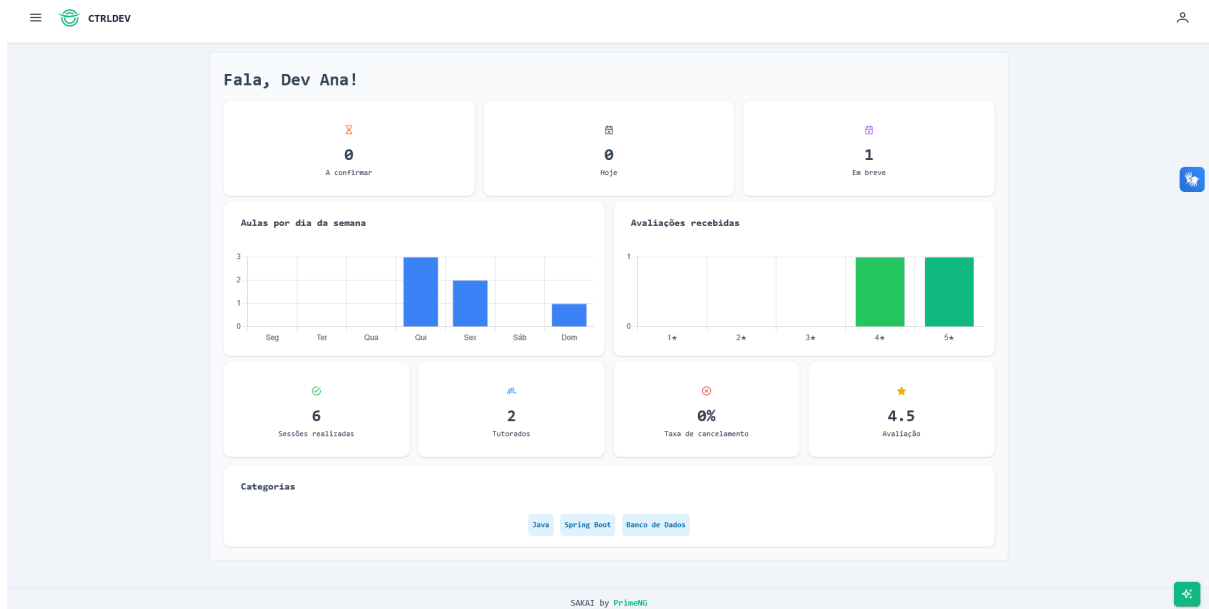


Figura 17: Dashboard

2. Perfil (Figura 18)

- Edite suas informações pessoais nos campos disponíveis;

The screenshot shows a web application interface for a user profile. At the top left, there is a logo with the text 'CTRLDEV'. The page title is 'Meu Perfil'. Below the title, there is a circular profile picture of a woman with curly hair. Below the picture, there are two buttons: '+ escolher foto' and 'X Remover'. The form contains several input fields: 'Nome' (first name) with the value 'Ana', 'Sobrenome' (last name) with the value 'Silva', 'Telefone' (phone number) with the value '(11) 99999-0001', 'Data de nascimento' (date of birth) with the value '14/03/1985', 'Pronomes' (pronouns) with the value 'ela/dela', and 'Categorias' (categories) with a dropdown menu showing 'Spring Boot, Banco de Dados, Java'. A 'Salvar' (Save) button is located at the bottom right of the form. The footer of the page says 'SAKAI by PrimeNG'.

Meu Perfil

+ escolher foto X Remover

Nome * Ana

Sobrenome * Silva

Telefone (11) 99999-0001

Data de nascimento 14/03/1985

Pronomes ela/dela

Categorias * Spring Boot, Banco de Dados, Java

Salvar

SAKAI by PrimeNG

Figura 18: Perfil do usuário

- Faça upload de uma nova foto de perfil;
- Clique em “Salvar” para confirmar as alterações.

Agradecimentos

À minha MÃE, *mamãe*, meu baluarte; por sua inefável habilidade de me acalmar nos momentos de dúvida. Agradeço por isso, portanto, mas principalmente, por tudo.

Ao meu pai, pela compreensão e amparo diante do perigo.

Ao meu orientador, Prof. Dr. Manoel Marques, pelo apoio, paciência e direcionamento no desenvolvimento deste trabalho.

A todos os meus professores, em especial Flávia Maristela, Pablo Florentino e Renato Novais, por toda a assistência, preocupação, oportunidades e ensinamentos.

Ao GSORT, grupo de pesquisa e desenvolvimento em que estive por 3 anos, onde primeiro aprendi a aplicar todo o conhecimento adquirido durante o curso.

A cada amigo que tive o privilégio de conhecer e que ajudou a tornar minhas tardes mais leves e minhas noites mais curtas.

À memória da minha tia Maria Celina, que hoje carrego com carinho
em meu coração.

À memória da minha avó Eremita, que partiu no mesmo dia em que
comecei a escrever este documento. Foi o pior dia da minha vida.

*“Deus não permite que tudo seja revelado ao homem neste mundo.”
— O Livro dos Espíritos*

Referências

- 1 SILVA, Leonardo. Avaliação da Efetividade das Atividades de Monitoria no Componente Curricular de Introdução à Programação. *In*: ANAIS do XXIX Workshop sobre Educação em Computação. Evento Online: SBC, 2021. p. 258–267. DOI: 10.5753/wei.2021.15917. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/15917>.
- 2 KUO, Yu-Chen; YAO, Ching-Bang; WU, Zhe-Yu. Online Peer-Tutoring for Programming Languages Based on Programming Ability and Teaching Skill. **Applied Sciences**, v. 12, n. 17, p. 8513, 2022. DOI: 10.3390/app12178513. Disponível em: <https://doi.org/10.3390/app12178513>.
- 3 CORPORATION, Oracle. **The Java® Language Specification (Java SE 25 Edition)**. [S. l.: s. n.], 2025. Disponível em: <https://docs.oracle.com/javase/specs/jls/se25/jls25.pdf>. Acesso em: 4 out. 2025.
- 4 TANZU, VMware. **Spring**. [S. l.: s. n.]. Disponível em: <https://spring.io/why-spring>. Acesso em: 6 out. 2025.
- 5 LLC, Google. **Angular**. [S. l.: s. n.]. Disponível em: <https://angular.dev/overview>. Acesso em: 4 out. 2025.
- 6 CORPORATION, Microsoft. **TypeScript**. [S. l.: s. n.]. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 4 out. 2025.
- 7 INFORMATICS, PrimeTek. **PrimeNG**. [S. l.: s. n.]. Disponível em: <https://primeng.org/>. Acesso em: 4 out. 2025.
- 8 INC., Tailwind Labs. **Tailwind CSS**. [S. l.: s. n.]. Disponível em: <https://tailwindcss.com/>. Acesso em: 26 dez. 2025.
- 9 GROUP, The PostgreSQL Global Development. **PostgreSQL**. [S. l.: s. n.]. Disponível em: <https://www.postgresql.org/>. Acesso em: 4 out. 2025.
- 10 CONSERVANCY, Software Freedom. **Git**. [S. l.: s. n.]. Disponível em: <https://git-scm.com/about>. Acesso em: 5 out. 2025.
- 11 GITHUB, Inc. **GitHub**. [S. l.: s. n.]. Disponível em: <https://github.com>. Acesso em: 5 out. 2025.
- 12 TEAM, The JUnit. **JUnit**. [S. l.: s. n.]. Disponível em: <https://junit.org/>. Acesso em: 5 out. 2025.
- 13 LABS, Bruno. **Bruno — API Client**. [S. l.: s. n.]. Disponível em: <https://www.usebruno.com/>. Acesso em: 5 out. 2025.
- 14 SUPERPROF. **Superprof Brasil — Aulas particulares**. [S. l.: s. n.]. Disponível em: <https://www.superprof.com.br/>. Acesso em: 11 out. 2025.

- 15 PROFES. **Profes — Aulas particulares, Atividades e Tutoria com IA.** [S. l.: s. n.]. Disponível em: <https://profes.com.br/>. Acesso em: 11 out. 2025.
- 16 ALUN, Grupo. **Sobre a Alura.** [S. l.: s. n.]. Disponível em: <https://www.alura.com.br/sobre>. Acesso em: 12 out. 2025.
- 17 TANZU, VMware. **Quartz Scheduler.** [S. l.: s. n.]. Disponível em: <https://docs.spring.io/spring-boot/reference/io/quartz.html>. Acesso em: 2 jan. 2026.
- 18 DIGITAL SAMBA, S.L. **Digital Samba.** [S. l.: s. n.]. Disponível em: <https://www.digitalsamba.com/>. Acesso em: 26 dez. 2025.
- 19 LTD., One Compiler Pvt. **OneCompiler.** [S. l.: s. n.]. Disponível em: <https://onecompiler.com/>. Acesso em: 5 jan. 2026.
- 20 DIGITAL, BRASIL. Governo. **VLibras.** [S. l.: s. n.]. Disponível em: <https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/vlibras>. Acesso em: 5 jan. 2026.
- 21 INVESTING.COM. **Preços Históricos EUR/BRL.** [S. l.: s. n.], 2026. Disponível em: <https://br.investing.com/currencies/eur-brl-historical-data>. Acesso em: 23 jan. 2026.

Glossário, Siglas e Abreviações

AI	Inteligência Artificial. 2, 8, 15, 19
API	Interface de Programação de Aplicações. 3, 8, 12
CORS	Compartilhamento de Recursos Entre Origens. 8
HTML	Linguagem de Marcação de Hipertexto. 3
HTTP	Protocolo de Transferência de Hipertexto. 8
JDK	Kit de Desenvolvimento Java. 14
JVM	Máquina Virtual Java. 3
REST	Transferência de Estado Representacional. 8
RF	Requisito Funcional. 6
RNF	Requisito Não-Funcional. 6
SDK	Kit de Desenvolvimento de Software. 9
SGBD	Sistema Gerenciador de Banco de Dados. 3
SPA	Aplicativo de Página Única. 8
SSE	Eventos Enviados pelo Servidor. 8
UI	Interface do Usuário. 3