



UNIVERSIDADE FEDERAL DO PARANÁ SETOR
DE EDUCAÇÃO PROFISSIONAL E
TECNOLÓGICA CURSO DE TECNOLOGIA EM
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Claudio Henrique Nascimento de Oliveira

Israel de Quadros

Michelly Narita Kuriyama

SQLab: AMBIENTE VIRTUAL DE APRENDIZADO DE LINGUAGEM SQL

CURITIBA

2025

Claudio Henrique Nascimento de Oliveira

Israel de Quadros

Michelly Narita Kuriyama

SQLab: AMBIENTE VIRTUAL DE APRENDIZADO DE LINGUAGEM SQL

Trabalho apresentado como requisito parcial à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas no curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Orientador: Prof. Dr. Dieval Guizelini

CURITIBA

2025

AGRADECIMENTOS

Concluimos o resumo deste curso com muita satisfação e entusiasmo. Esta é uma etapa cheia de desafios e recompensas. Primeiramente, agradecemos a Deus por nos dar força, sabedoria e oportunidade para trilhar esse caminho de aprendizado e crescimento. A vida traz dificuldades e recompensas, e somos gratos por cada momento.

Às nossas famílias, que são os verdadeiros pilares desta jornada, expressamos nossa mais profunda gratidão pelo apoio incondicional, amor e paciência. Suas palavras de incentivo e compreensão foram vitais para me manter firme mesmo nos momentos mais difíceis. Sinceros agradecimentos aos nossos amigos que estiveram ao nosso lado em cada passo do caminho, compartilhando risadas e desafios. Sua amizade tornou esta jornada mais fácil e significativa e cada conquista foi compartilhada.

Agradecemos especialmente aos nossos colegas de curso, cuja cooperação, troca de informações e espírito de equipe foram indispensáveis. Juntos superamos desafios e criamos um ambiente de aprendizagem enriquecedor. Aos professores, nossa eterna gratidão. Suas aulas iam além do conteúdo acadêmico para nos ensinar valores e habilidades para a vida. Em particular, queremos enfatizar a nossa gratidão ao professor orientador Dieval Guizelini. Sua dedicação, paciência e atenção foram fundamentais para a conclusão deste trabalho. Sua orientação útil e atenciosa fez toda a diferença e seremos eternamente gratos.

Por fim, gostaríamos de agradecer a todos que contribuíram de alguma forma para a conclusão deste trabalho. Essa conquista é resultado de muitos esforços e colaborações, e cada aprendizado e lembrança é guardado com amor e gratidão.

“Nada na vida deve ser temido, somente compreendido. Agora é hora de compreender mais para temer menos”.

Marie Curie

RESUMO

O ensino de banco de dados frequentemente enfrenta desafios significativos, com aulas predominantemente teóricas e escassas oportunidades práticas para os alunos. A falta de ferramentas adequadas para práticas efetivas e a dificuldade de acesso a ambientes de desenvolvimento são obstáculos recorrentes. Em resposta a essas limitações, a equipe desenvolveu o SQLab, uma plataforma web abrangente projetada para transformar o ensino e aprendizado de SQL e da administração de bancos de dados. O SQLab foi concebido com uma arquitetura robusta, composta por duas interfaces distintas que atendem às necessidades específicas de professores e alunos. Para os docentes, o sistema oferece um painel de controle intuitivo, que permite o gerenciamento de turma, desde a inscrição até o acompanhamento individualizado. A grande inovação para professores reside na capacidade de cadastrar atividades com uma variedade de tipos de questões, incluindo a inserção de respostas válidas para a correção automática, otimizando o processo avaliativo. Isso permite que o professor foque na didática e na personalização do conteúdo. Já para os estudantes, o SQLab proporciona um ambiente prático e imersivo, que pode ser acessado diretamente pelo navegador, eliminando a necessidade de qualquer configuração complexa ou instalação prévia. Isso democratiza o acesso à prática, permitindo que os alunos se concentrem no aprendizado e na execução de comandos SQL. Construído sobre uma arquitetura moderna e escalável, utilizando Java (Spring Boot) no backend e Angular 17 no frontend. Com o SQLab, espera-se não apenas facilitar o ensino e aprendizado de SQL, mas também proporcionar um ambiente educacional mais engajador e acessível. Ao oferecer ferramentas avançadas para professores e uma plataforma intuitiva para alunos, o SQLab promete melhorar significativamente a experiência de estudantes e educadores no campo de administração de banco de dados.

Palavras-chave: Banco de dados. Aprendizado. SQL. Tecnologia educacional. Ensino interativo.

ABSTRACT

Database education often faces significant challenges, with classes being predominantly theoretical and offering few practical opportunities for students. The lack of appropriate tools for effective practice and the difficulty of accessing development environments are recurring obstacles. In response to these limitations, the team developed SQLab, a comprehensive web platform designed to transform the teaching and learning of SQL and database administration. SQLab was conceived with a robust architecture, composed of two distinct interfaces tailored to the specific needs of both instructors and students. For teachers, the system offers an intuitive control panel that allows for classroom management—from student registration to individualized progress monitoring. A major innovation lies in the ability to register activities with a variety of question types, including the insertion of valid answers for automatic correction, streamlining the assessment process. This enables educators to focus on pedagogy and content customization. For students, SQLab provides a practical and immersive environment that can be accessed directly through the browser, eliminating the need for any complex setup or prior installation. This democratizes access to hands-on learning, allowing students to focus on acquiring knowledge and executing SQL commands. The platform is built on a modern and scalable architecture, using Java (Spring Boot) on the backend and Angular 17 on the frontend. With SQLab, the goal is not only to facilitate SQL teaching and learning, but also to offer a more engaging and accessible educational environment. By providing advanced tools for instructors and an intuitive platform for students, SQLab promises to significantly enhance the experience of both students and educators in the field of database administration.

Keywords: Database. Practical learning. SQL. Educational technology. Interactive teaching.

SUMÁRIO

1 INTRODUÇÃO.....	1
1.1 PROBLEMA.....	2
1.2 OBJETIVOS.....	3
1.2.1 GERAIS.....	4
1.3 JUSTIFICATIVA.....	4
2 FUNDAMENTAÇÃO TEÓRICA.....	6
2.1 AMBIENTE DE APRENDIZAGEM EM SQL PARA ESTUDANTES.....	6
2.2 METODOLOGIAS PEDAGÓGICAS PARA O ENSINO DE SQL.....	7
2.2 SOFTWARES SIMILARES.....	8
2.2.1 FARMAALG.....	8
2.2.2 COURSERA.....	9
2.2.3 PG EXERCISES.....	10
3 MATERIAL E MÉTODOS.....	11
3.1 METODOLOGIA DE DESENVOLVIMENTO.....	11
3.1.1 UML.....	11
3.1.2 KANBAN.....	12
3.2 FERRAMENTAS DE DESENVOLVIMENTO.....	14
3.2.1 Angular 17.....	14
3.2.2 Java.....	15
3.2.3 Spring Boot.....	16
3.2.4 H2.....	17
3.2.5 POSTGRESQL.....	18
3.2.6 DRAW.IO.....	20
3.2.7 EXCALIDRAW.....	20
3.2.8 OBSIDIAN.....	21
4 APRESENTAÇÃO DO SISTEMA.....	23

MODELAGEM DO PROJETO.....	24
4.1.1 DIAGRAMA DE CASO DE USO.....	24
4.1.2 TABELA DE REQUISITOS.....	24
4.1.3 DIAGRAMA DE CLASSES.....	25
4.1.4 DIAGRAMA DE SEQUÊNCIA.....	25
4.2 ARQUITETURA DO SISTEMA.....	25
4.3 FUNCIONALIDADES.....	26
5 CONCLUSÃO.....	43
5.1 CONSIDERAÇÕES FINAIS.....	44
6 REFERÊNCIAS.....	46
7 APÊNDICES.....	48
APÊNDICE A – DIAGRAMA DE CASO DE USO.....	48
APÊNDICE B – TABELA DE REQUISITOS.....	49
APÊNDICE C – ESPECIFICAÇÕES DE CASO DE USO.....	51
APÊNDICE D – DIAGRAMA DE CLASSE DE ANÁLISE.....	87
APÊNDICE E – DIAGRAMA DE SEQUÊNCIA DE ANÁLISE.....	89
APÊNDICE F – MODELO LÓGICO DE DADOS.....	102
APÊNDICE G – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO SELECT.....	103
APÊNDICE H – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO INSERT.....	104
APÊNDICE I – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO UPDATE.....	105
APÊNDICE J – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO CREATE.....	107
APÊNDICE K – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO DELETE.....	108
APÊNDICE L – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO	

TIPO DROP.....	109
----------------	-----

1 INTRODUÇÃO

O ensino de linguagens de consulta e manipulação de dados, como a SQL (Structured Query Language), enfrenta desafios significativos na criação de experiências de aprendizado práticas e interativas para os alunos (Garrison & Anderson, 2003). Tradicionalmente, as aulas teóricas são complementadas por exercícios práticos utilizando sistemas de gerenciamento de bancos de dados como MySQL e PostgreSQL (Clark, 1994). No entanto, a oportunidade para os alunos praticarem autonomamente e aplicarem seus conhecimentos em contextos reais é limitada, devido à falta de ferramentas educacionais que integrem prática e feedback imediato (Mayer, 2009). Esta limitação se deve, em parte, à escassez de recursos educacionais acessíveis que permitam aos alunos desenvolverem habilidades práticas de SQL de maneira eficaz (Clark, 1994; Kirschner et al., 2006).

A abordagem convencional de atribuir atividades para serem realizadas em casa enfrenta diversos obstáculos. Em primeiro lugar, a acessibilidade é uma preocupação central: nem todos os alunos têm acesso aos recursos necessários para configurar e operar ambientes de aprendizado adequados. A preparação de ambientes de desenvolvimento pode ser complexa e exigir conhecimentos técnicos adicionais, o que pode desencorajar muitos estudantes iniciantes. Além disso, as ferramentas disponíveis frequentemente não são projetadas especificamente para o ensino de programação, limitando-se a formatos de entrega de tarefas que não proporcionam uma interação dinâmica e imediata com o conteúdo de SQL (Siemens, 2005).

Em contrapartida, avanços recentes na tecnologia educacional oferecem oportunidades promissoras para superar esses desafios. O presente trabalho de conclusão de curso propõe o desenvolvimento de uma plataforma de aprendizado inovadora, focada em SQL, que permite aos alunos praticarem habilidades de consulta e manipulação de dados em um ambiente virtual interativo. Esta aplicação não só oferecerá acesso a recursos educacionais estruturados e atualizados, mas também permitirá que os educadores monitorem o progresso dos alunos de forma mais eficaz e personalizada.

Ao integrar tecnologia de ponta com metodologias pedagógicas comprovadas, a plataforma proposta tem o potencial de revolucionar o ensino de SQL. Estudantes

de tecnologia e programação poderão beneficiar-se significativamente, adquirindo habilidades práticas essenciais para o mercado de trabalho moderno. Além disso, professores terão à disposição ferramentas poderosas para reforçar o aprendizado da matéria acompanhando o desenvolvimento de seus alunos de maneira mais detalhada.

1.1 PROBLEMA

Uma das principais dificuldades do ensino de SQL reside na lacuna entre a teoria apresentada em sala de aula e a aplicação prática necessária no mercado de trabalho (Garrison & Anderson, 2003). As aulas frequentemente se limitam a exposições teóricas, com poucas oportunidades para os alunos desenvolverem habilidades práticas efetivas em consultas e manipulação de dados reais (Clark, 1994; Kirschner et al., 2006). Além disso, a ausência de plataformas que ofereçam feedback imediato e personalizado compromete a motivação e o engajamento dos alunos (Hattie & Timperley, 2007).

A acessibilidade a recursos para estudos da área é um dos pontos a serem destacados dentro deste assunto. Muitos alunos enfrentam obstáculos para configurar e utilizar ambientes de desenvolvimento apropriados para praticar SQL. A complexidade técnica envolvida na instalação de sistemas de gerenciamento de banco de dados como MySQL ou PostgreSQL pode desencorajar estudantes menos familiarizados com essas tecnologias, impactando negativamente seu aprendizado e participação efetiva nas atividades acadêmicas (Siemens, 2005).

As ferramentas educacionais disponíveis muitas vezes não são projetadas especificamente para o ensino específico de SQL, e não trazem a visibilidade e experiência prática necessária para absorvimento de conteúdo da parte dos alunos. A falta de recursos para acompanhamento e avaliação por parte dos professores dificulta a monitoração do progresso dos alunos e a identificação precisa de áreas que necessitam de reforço (Clark, 1994). Esses desafios têm um impacto significativo na motivação dos alunos e na eficiência do ensino. A falta de experiências práticas e interativas pode diminuir o interesse dos estudantes pela disciplina e pela área de tecnologia como um todo.

Além das limitações já apontadas, é importante destacar que a maioria das plataformas disponíveis atualmente que oferecem exercícios de SQL para prática

estudantil não permite a personalização do conteúdo por parte dos docentes. Essas ferramentas, embora úteis para treinamentos genéricos, não contemplam a diversidade de abordagens pedagógicas ou os objetivos específicos de cada disciplina. A ausência de funcionalidades que possibilitem a criação de exercícios personalizados impede que professores adaptem o conteúdo às necessidades de suas turmas, reduzindo o potencial de contextualização e aprofundamento dos conceitos ensinados. Isso compromete a efetividade do processo de ensino-aprendizagem, dificultando a construção de trilhas de aprendizado coerentes com a evolução dos alunos.

Outro aspecto técnico e estrutural crítico observado em soluções existentes diz respeito à questão do paralelismo e da escalabilidade. Com múltiplos estudantes acessando simultaneamente os exercícios e executando consultas SQL, torna-se essencial que o sistema de backend consiga processar essas operações de forma eficiente. Algumas plataformas, como forma de contornar esse desafio, mantêm centenas de bancos de dados individuais previamente configurados, direcionando cada usuário para um deles. No entanto, essa abordagem apresenta sérias limitações quanto à viabilidade, à manutenção e à escalabilidade a longo prazo. Tais soluções são pouco práticas em contextos educacionais com grande volume de usuários. Em contraste, a utilização do banco de dados H2 — leve, in-memory e de fácil replicação — permite a execução isolada e paralela de atividades SQL, garantindo maior desempenho, economia de recursos e escalabilidade da plataforma, sem comprometer a experiência do usuário.

Para os professores, a dificuldade em gerenciar turmas numerosas e avaliar o desempenho dos alunos de maneira eficaz pode resultar em uma sobrecarga de trabalho e na dificuldade de fornecer suporte individualizado adequado a cada estudante. Diante dessas dificuldades, há uma necessidade urgente de soluções inovadoras que superem as barreiras existentes no ensino de SQL. Plataformas educacionais específicas, que integrem tecnologias avançadas com metodologias pedagógicas eficazes, têm o potencial de transformar a forma como o SQL é ensinado e aprendido.

1.2 OBJETIVOS

1.2.1 GERAL

Desenvolver e implementar uma plataforma educacional focada no ensino de SQL, visando oferecer uma experiência de aprendizado prática, interativa e acessível para alunos de tecnologia. A plataforma será projetada para superar os desafios atuais no ensino de SQL, proporcionando atividades práticas, feedback imediato e ferramentas avançadas para professores acompanharem o progresso destes discentes.

1.2.2 OBJETIVOS ESPECÍFICOS

- A) Definir Requisitos Funcionais e Não Funcionais: Documentar requisitos detalhados, incluindo funcionalidades específicas da plataforma (como execução de consultas SQL em tempo real) e requisitos não funcionais (como desempenho e segurança).
- B) Projetar e Desenvolver Interfaces Intuitivas: Criar interfaces de usuário intuitivas e acessíveis, seguindo princípios de User Experience (UX), para facilitar a navegação e utilização por alunos e professores.
- C) Implementar Funcionalidades de SQL Interativas: Desenvolver funcionalidades que permitam aos alunos escreverem e executar consultas SQL diretamente na plataforma, manipular dados e visualizar resultados de consultas em tempo real.
- D) Integrar lógica que possibilite a personalização do conteúdo a ser praticado, fornecendo aos professores a opção de utilizar questões já cadastradas no sistema ou criar novas atividades com conteúdo personalizado (sendo INSERT, UPDATE, DELETE, CREATE)
- E) Implementar Ferramentas de Acompanhamento Acadêmico: Desenvolver funcionalidades que permitam aos professores acompanharem o desempenho dos alunos ao longo do curso, identificar áreas de dificuldade e oferecer suporte personalizado conforme necessário.

1.3 JUSTIFICATIVA

A justificativa para o desenvolvimento desta plataforma educacional fundamenta-se nas limitações dos métodos tradicionais de ensino de SQL, que priorizam a teoria em detrimento da prática (Mayer, 2009). A ausência de ferramentas que combinem prática interativa com feedback imediato dificulta o desenvolvimento de habilidades aplicáveis no mercado de trabalho (Hattie & Timperley, 2007). Além disso, a personalização do ensino, essencial para atender a diferentes níveis de proficiência, é limitada em muitas plataformas existentes (Fisher & Baird, 2011). Essa abordagem compromete a consolidação do conhecimento e não prepara adequadamente os alunos para os desafios do mercado. Além disso, a escassez de ferramentas acessíveis, integradas e específicas para a prática de SQL dificulta o aprendizado efetivo, especialmente entre estudantes com pouca familiaridade técnica. Muitas das soluções existentes não oferecem exercícios realistas nem permitem a personalização por parte dos professores, limitando a adaptação pedagógica e o acompanhamento individualizado. Diante disso, uma plataforma que una prática real de consultas SQL, personalização docente e escalabilidade técnica representa uma resposta necessária e promissora aos desafios enfrentados no ensino da disciplina.

A implementação desta plataforma visa superar esses desafios ao proporcionar uma experiência de aprendizado prática e interativa. Permitindo aos alunos não apenas aprenderem a teoria por trás das consultas SQL, mas também praticarem diretamente em um ambiente simulado de banco de dados, onde poderão escrever consultas, manipular dados e observar os resultados em tempo real, não somente proporcionando compreensão e confiança aos alunos ao manipular o SQL, mas também os preparando melhor para os desafios futuros na carreira profissional.

Além dos benefícios para os alunos, a plataforma fornecerá ferramentas avançadas para os professores acompanharem o progresso dos estudantes, permitindo que os docentes identifiquem áreas de dificuldade e forneçam suporte personalizado ao aluno. Essa abordagem não apenas melhora a eficácia do ensino, mas também aumenta o envolvimento dos alunos ao proporcionar um feedback imediato e relevante sobre seu desempenho.

Portanto, o desenvolvimento desta plataforma educacional não apenas endereça as deficiências atuais no ensino de SQL, mas também promove um ambiente de aprendizado mais eficaz, engajador e preparatório para os desafios práticos enfrentados pelos futuros profissionais de tecnologia.

2 FUNDAMENTAÇÃO TEÓRICA

A evolução contínua das tecnologias de informação e comunicação tem redefinido as expectativas e necessidades no campo da educação, promovendo a integração de ferramentas digitais para melhorar o aprendizado (Reimers & Schleicher, 2020; Garrison & Anderson, 2003). A personalização do ensino, aliada ao uso de plataformas interativas, é essencial para atender às demandas de um mercado de trabalho cada vez mais técnico (Boulos et al., 2020). Com o aumento da demanda por profissionais qualificados em tecnologia e bancos de dados, é natural que os métodos de ensino acompanhem essas mudanças, oferecendo não apenas conhecimento teórico, mas também habilidades práticas e aplicadas.

Os desafios enfrentados no ensino de SQL também incluem a heterogeneidade de experiências e habilidades dos alunos. Muitos estudantes entram em cursos de tecnologia com diferentes níveis de familiaridade com bancos de dados e linguagens de programação, o que torna essencial a adoção de uma abordagem flexível e adaptável (Fisher & Baird, 2011).

Além da necessidade de personalização e adaptação ao perfil diversificado dos alunos, o SQLab também aborda a importância crescente da avaliação formativa e do feedback construtivo. Segundo Hattie & Timperley (2007), o feedback construtivo é fundamental para melhorar o desempenho dos alunos e promover o aprendizado efetivo. Ao fornecer ferramentas que permitem aos professores monitorarem o progresso dos alunos em tempo real, identificar áreas de melhoria e oferecer orientação personalizada, a plataforma não apenas facilita o aprendizado autônomo, mas também fortalece a relação colaborativa entre educadores e estudantes.

2.1 AMBIENTE DE APRENDIZAGEM EM SQL PARA ESTUDANTES

O aprendizado de SQL é essencial para qualquer estudante que pretende trabalhar com gerenciamento de bancos de dados. A compreensão de como projetar e estruturar bancos de dados, por meio da modelagem de dados, é fundamental para o desenvolvimento de soluções eficientes e organizadas. Os alunos devem entender conceitos como tabelas, chaves primárias, chaves estrangeiras e os relacionamentos entre tabelas. A modelagem de dados é a base sobre a qual são construídas as consultas e manipulações subsequentes de dados.

À medida que os alunos avançam no estudo de SQL, as consultas se tornam cada vez mais complexas. A utilização de junções (INNER JOIN, LEFT JOIN, etc.), subconsultas e agregações (GROUP BY, HAVING) pode ser desafiadora, especialmente para iniciantes. A habilidade de construir consultas complexas é crucial para realizar análises de dados detalhadas e extrair informações significativas a partir de grandes conjuntos de dados. Sem uma prática adequada, os estudantes podem se sentir sobrecarregados e frustrados.

A otimização de consultas é outra área crítica onde os alunos frequentemente encontram dificuldades. Escrever consultas eficientes que utilizem o mínimo de recursos e tenham um desempenho rápido é uma habilidade que se desenvolve com a experiência e a prática constante. A falta de compreensão sobre índices, planos de execução e técnicas de otimização pode levar a consultas que são lentas e consomem muitos recursos, impactando negativamente a performance dos sistemas de banco de dados. Questões técnicas, aspectos como segurança e permissões também são fundamentais no aprendizado de SQL, saber como conceder e revogar permissões de acesso a tabelas e bancos de dados é outra habilidade crucial para manter a integridade e a segurança dos dados. Erros na configuração de permissões podem resultar em vulnerabilidades de segurança significativas, a capacidade de lidar com erros, exceções e problemas de integridade de dados é essencial para garantir que os sistemas de banco de dados sejam robustos e resilientes.

2.2 METODOLOGIAS PEDAGÓGICAS PARA O ENSINO DE SQL

As metodologias pedagógicas desempenham um papel crucial no ensino de SQL, influenciando diretamente a eficácia e o engajamento dos alunos. Uma abordagem construtivista, onde os alunos constroem seu próprio conhecimento através da interação ativa com o conteúdo, é particularmente eficaz no ensino de linguagens de programação. A aplicação de métodos como a aprendizagem baseada em problemas (PBL) e a aprendizagem ativa pode proporcionar um entendimento mais profundo e duradouro dos conceitos de SQL (Jonassen, 1999).

A aprendizagem baseada em problemas (PBL) envolve a apresentação de problemas reais aos alunos, incentivando-os a encontrar soluções através da pesquisa e da prática. No contexto do ensino de SQL, isso pode incluir a análise e solução de problemas de bancos de dados reais, permitindo que os alunos apliquem teorias em situações práticas. Esta abordagem não apenas melhora a compreensão dos conceitos, mas também desenvolve habilidades críticas de resolução de

problemas e pensamento analítico (Hmelo-Silver, 2004).

A aprendizagem ativa, por outro lado, envolve a participação dos alunos em atividades que promovem a análise, a síntese e a avaliação do conteúdo aprendido. Em SQL, isso pode incluir exercícios interativos, discussões em grupo e simulações de consultas e manipulações de dados. A utilização de ferramentas tecnológicas que oferecem feedback instantâneo pode melhorar significativamente a eficácia da aprendizagem ativa, permitindo que os alunos corrijam erros e consolidem seu entendimento em tempo real (Prince, 2004).

2.2 SOFTWARES SIMILARES

2.2.1 FARMAALG

Desenvolver e implementar uma plataforma educacional inovadora focada no ensino de programação, visando oferecer uma experiência de aprendizado prática, interativa e acessível para alunos de tecnologia. A plataforma será projetada para superar os desafios atuais no ensino de programação, proporcionando atividades práticas, feedback imediato e ferramentas avançadas para professores acompanharem o progresso dos alunos.

A ferramenta executa o código submetido pelo aluno utilizando entradas predefinidas e compara os resultados esperados previamente cadastrados. O sistema fornece feedback instantâneo, promovendo uma correção ágil e eficiente para o aluno, o que facilita o aprendizado prático e a retenção de conceitos fundamentais de programação.

Atualmente em sua terceira versão, o FarmaALG é disponibilizado exclusivamente como uma plataforma online acessível via navegador. A primeira versão, também criada por KUTZKE (2015), foi concebida como uma interface web que suportava as linguagens C e Pascal para a gestão online dos códigos-fonte de exercícios entregues pelos alunos. A interface de entrega de atividades do FarmaALG se assemelha a um editor de texto ou IDE moderna, apresentando exemplos da entrega proposta e dicas exibidas ao longo do desenvolvimento do código.

Um dos diferenciais do FarmaALG, conforme análises de KUTZKE (2015) e MUELLER et al. (2021), é a capacidade de comparar respostas de diferentes alunos ou múltiplas submissões do mesmo aluno. As semelhanças e diferenças são exibidas lado a lado, permitindo uma verificação humana de lógica similar ou plágio. Este recurso é particularmente útil para identificar padrões de aprendizado e

possíveis casos de plágio, fornecendo uma ferramenta robusta para professores monitorarem o progresso e a originalidade dos trabalhos dos alunos.

A plataforma continua a evoluir, incorporando feedback dos usuários para melhorar suas funcionalidades e a experiência de aprendizado. A disponibilização exclusiva via navegador facilita o acesso e a utilização, permitindo que alunos e professores utilizem a ferramenta em qualquer lugar e a qualquer momento. Além disso, a interface intuitiva e o feedback instantâneo tornam o processo de aprendizado mais dinâmico e interativo, incentivando os alunos a corrigirem seus erros e melhorarem suas habilidades de programação.

Em resumo, o FarmaALG representa um avanço significativo no ensino de programação, integrando conceitos de aprendizado baseado no erro com tecnologias modernas de desenvolvimento web. Ao proporcionar uma plataforma acessível e eficiente para o aprendizado prático de linguagens de programação, o FarmaALG contribui para a formação de profissionais mais capacitados e preparados para enfrentar os desafios do mercado de tecnologia.

2.2.2 COURSERA

O Coursera é uma das melhores plataformas de ensino online, altamente recomendada para estudantes, funcionários ou gerentes que buscam aprimorar suas habilidades e se manter atualizados nas novidades de suas áreas de trabalho. A

plataforma oferece uma ampla gama de cursos gratuitos, tutoriais e outros recursos educativos sobre praticamente qualquer assunto de interesse. Esses cursos são ministrados por professores universitários das melhores instituições ao redor do mundo, garantindo a alta qualidade do conteúdo disponível.

Uma das principais vantagens do Coursera é a vasta quantidade de cursos voltados para a programação. A plataforma oferece opções que abrangem desde introduções à programação, HTML e CSS, até cursos mais avançados em programação para IoT, linguagem C, Python, Java e muitos outros. Embora os cursos sejam gratuitos, aqueles que desejam obter certificações oficiais pelo Coursera precisam pagar uma taxa. No entanto, a oportunidade de aprender de maneira gratuita e de alta qualidade é uma grande vantagem, especialmente para iniciantes que buscam adquirir novas habilidades ou aprimorar conhecimentos existentes.

Além dos cursos técnicos, o Coursera também disponibiliza uma variedade de conteúdos não técnicos. Estes incluem tópicos como design de produtos,

marketing digital e planejamento de impostos, entre muitos outros. A diversidade de assuntos abordados na plataforma permite que os usuários ampliem seus conhecimentos em diferentes áreas, tornando-se mais versáteis e competentes em suas carreiras. Todos os cursos são desenvolvidos com conteúdos relevantes e interessantes, projetados para atender às necessidades de um público global.

O Coursera utiliza metodologias de ensino modernas, incluindo aulas em vídeo, leituras, quizzes e fóruns de discussão, que facilitam o aprendizado interativo e colaborativo. A plataforma também oferece funcionalidades avançadas, como a possibilidade de seguir um cronograma flexível, permitindo que os alunos aprendam no seu próprio ritmo. Embora a plataforma seja em inglês, muitos cursos possuem legendas em vários idiomas, o que amplia o acesso ao conteúdo para uma audiência global. No geral, o Coursera é uma excelente ferramenta para quem deseja expandir seus conhecimentos e habilidades de forma acessível e eficiente.

2.2.3 PG EXERCISES

O PGExercises é uma plataforma gratuita de aprendizado online voltada exclusivamente para a prática de SQL, sendo uma ferramenta altamente

recomendada para estudantes, desenvolvedores ou profissionais da área de tecnologia que desejam consolidar seus conhecimentos em bancos de dados relacionais. O site oferece uma variedade de exercícios práticos focados na linguagem SQL, mais especificamente na versão utilizada pelo sistema de gerenciamento PostgreSQL, um dos bancos mais utilizados no mercado.

O principal diferencial do PGExercises é seu enfoque em exercícios aplicados, permitindo ao usuário colocar em prática seus conhecimentos por meio de desafios reais. Os exercícios são organizados em diferentes categorias e níveis de dificuldade, abordando desde operações básicas, como seleções simples e filtros, até consultas mais avançadas com agregações, subconsultas e junções complexas. Essa estrutura progressiva é ideal para quem está começando a aprender SQL, assim como para aqueles que desejam revisar ou aprofundar seus conhecimentos. Todos os exercícios são baseados em um banco de dados fictício de uma academia, o que proporciona um contexto consistente e facilita o entendimento da aplicação dos conceitos. A plataforma também oferece um ambiente interativo onde é possível escrever, executar e validar as consultas diretamente pelo navegador, sem a necessidade de instalar qualquer software adicional. Esse aspecto torna o uso do PGExercises extremamente acessível e prático para os usuários.

Embora o PGExercises não ofereça certificados nem conteúdo teórico detalhado, ele se destaca como uma ferramenta complementar de alto valor educacional, especialmente para quem deseja reforçar a parte prática do aprendizado em SQL. A ausência de tutoriais ou aulas em vídeo é compensada pela clareza dos enunciados e pela estrutura lógica dos exercícios, que incentivam a resolução por tentativa e erro, promovendo um aprendizado ativo e eficaz.

3.1 METODOLOGIA DE DESENVOLVIMENTO

3.1.1 UML

A UML (Unified Modeling Language) é uma linguagem de modelagem padronizada amplamente utilizada na engenharia de software para especificar, visualizar, construir e documentar sistemas complexos. Criada na década de 1990, a UML facilita a compreensão compartilhada entre desenvolvedores e outros stakeholders, promovendo a clareza no design de sistemas. Composta por diagramas estruturais e comportamentais, a UML permite a representação detalhada tanto da organização estática quanto da dinâmica dos sistemas. Os diagramas estruturais, como o diagrama de classes, diagrama de objetos e diagrama de componentes, ajudam a ilustrar a estrutura do sistema, incluindo classes, atributos, operações e os relacionamentos entre as classes. Já os diagramas comportamentais, como o diagrama de casos de uso, diagrama de sequência e diagrama de estados, focam nas interações entre objetos e eventos, destacando como os componentes do sistema interagem e mudam de estado ao longo do tempo (Booch, Rumbaugh & Jacobson, 2005).

Oferece uma linguagem comum que reduz ambiguidades, facilita a documentação clara e organizada do sistema e permite a visualização de problemas e soluções de maneira intuitiva. Além disso, a UML auxilia no planejamento e design, ajudando a identificar e resolver problemas potenciais antes que eles se tornem críticos. A reutilização de componentes e padrões de design é outra vantagem importante, promovendo eficiência e qualidade no desenvolvimento de software. Referências notáveis sobre UML, como "The Unified Modeling Language User Guide" de Booch, Rumbaugh e Jacobson (2005), e "UML Distilled: A Brief Guide to the Standard Object Modeling Language" de Martin Fowler (2004), oferecem uma compreensão aprofundada e prática sobre como utilizar essa ferramenta essencial no desenvolvimento de software. A aplicação eficaz da UML contribui para um desenvolvimento mais estruturado, comunicativo e eficiente, refletindo diretamente na qualidade dos sistemas criados.

3.1.2 KANBAN

O Kanban é uma metodologia ágil que se destaca pela simplicidade e eficiência na organização e gestão de atividades, sendo amplamente utilizada no desenvolvimento de sistemas. Originado no sistema de produção da Toyota na década de 1940, o Kanban visa melhorar a produtividade e a eficiência através da visualização do trabalho e do controle do fluxo de tarefas. Utilizando quadros, geralmente divididos em colunas que representam diferentes estágios de um processo (como "A Fazer", "Em Progresso" e "Concluído"), o Kanban permite que equipes de desenvolvimento visualizem facilmente o progresso das atividades, identifiquem gargalos e priorizem tarefas de forma colaborativa e dinâmica.

Os quadros Kanban são ferramentas poderosas para a gestão de projetos, pois oferecem uma visualização clara e imediata do status das tarefas. Isso facilita a comunicação entre os membros da equipe e permite uma rápida adaptação às mudanças de prioridade, melhorando a flexibilidade do projeto. Além disso, ao limitar o número de tarefas em progresso (WIP - Work In Progress), o Kanban ajuda a evitar a sobrecarga de trabalho e a focar na conclusão de tarefas antes de iniciar novas, promovendo um fluxo de trabalho mais eficiente e sustentável. Essa abordagem não apenas aumenta a produtividade, mas também melhora a qualidade do produto final, pois as equipes podem se concentrar em resolver problemas de forma mais eficaz (Kniberg, 2011).

A aplicação do Kanban no desenvolvimento de sistemas proporciona vários benefícios tangíveis. Primeiramente, a transparência é significativamente aumentada, permitindo que todos os membros da equipe tenham uma visão clara do que está sendo feito e do que precisa ser feito. Isso não apenas melhora a coordenação, mas também fortalece a colaboração e o senso de responsabilidade entre os membros da equipe. Em segundo lugar, a capacidade de visualizar e gerenciar o fluxo de trabalho em tempo real ajuda a identificar e mitigar problemas antes que eles se tornem críticos, reduzindo atrasos e melhorando a entrega contínua de valor ao cliente (Anderson, 2010).

Além disso, o Kanban é flexível e pode ser facilmente integrado com outras metodologias ágeis, como Scrum, para otimizar ainda mais a gestão de projetos. Sua implementação não requer mudanças drásticas nos processos existentes, o que facilita sua adoção gradual e permite melhorias contínuas baseadas em feedback real do fluxo de trabalho. Essa adaptabilidade é crucial no ambiente dinâmico do desenvolvimento de software, onde as necessidades e prioridades podem mudar rapidamente. Em resumo, o Kanban é uma ferramenta valiosa que, ao promover

uma gestão visual e colaborativa das atividades, contribui significativamente para a eficiência, a qualidade e a satisfação das equipes e dos clientes no desenvolvimento de sistemas.

3.2 FERRAMENTAS DE DESENVOLVIMENTO

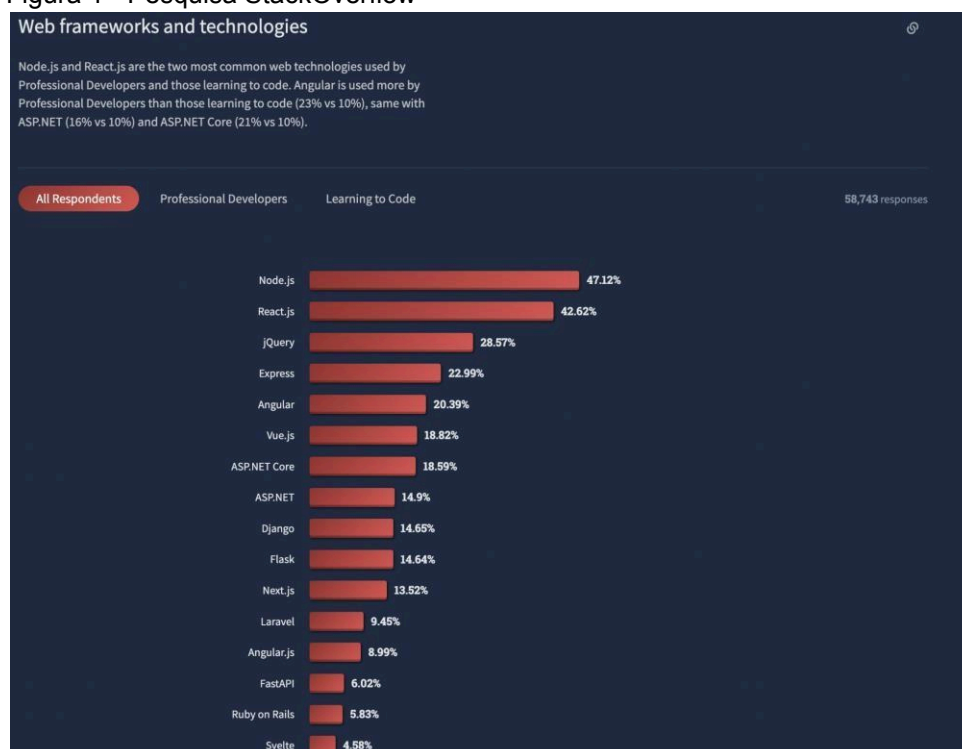
3.2.1 Angular 17

Angular é um framework front-end baseado em TypeScript, desenvolvido pelo Google e lançado inicialmente em 2016 como uma evolução do AngularJS. Seu objetivo é facilitar o desenvolvimento de aplicações web robustas e escaláveis, fornecendo uma estrutura completa para a criação de interfaces dinâmicas. O Angular utiliza a arquitetura baseada em componentes, promovendo modularidade e reutilização de código. Além disso, emprega o conceito de change detection eficiente para atualizar a interface do usuário apenas quando necessário, garantindo melhor desempenho.

Uma das principais vantagens do Angular é seu sistema de vinculação bidirecional de dados (two-way data binding), que mantém o modelo e a visualização sempre sincronizados. Ele também conta com um poderoso sistema de injeção de dependências, roteamento avançado e ferramentas como o Angular CLI para automação de tarefas.

No ano de 2022, o Stack Overflow realizou uma comparação entre frameworks e tecnologias web (Figura 1), onde o Angular se destacou como um dos mais utilizados por desenvolvedores profissionais, ficando atrás apenas de tecnologias como Node.js e React:

Figura 1 - Pesquisa StackOverflow



A robustez do Angular como framework completo para desenvolvimento web elimina a necessidade de bibliotecas externas para muitas funcionalidades essenciais. No entanto, ele também permite a integração com diversas bibliotecas para expandir suas capacidades. No Angular, o roteamento é gerenciado de forma nativa pelo Angular Router, permitindo a navegação eficiente entre páginas e componentes. Para o gerenciamento de estado e cacheamento de dados, pode-se utilizar o RxJS junto com NgRx, que implementa o padrão Redux de forma reativa. Já para o consumo de APIs REST, o HttpClient do Angular fornece uma solução poderosa e otimizada para requisições HTTP.

3.2.2 Java

Java é uma linguagem de programação de propósito geral, orientada a objetos baseada em classes, criada por James Gosling na Sun Microsystems e lançada em 1995. Desde então, consolidou-se como uma das linguagens mais influentes e amplamente utilizadas no cenário global de desenvolvimento de software. A versão Java 17, sendo uma Long-Term Support (LTS), representa um marco importante, incorporando melhorias significativas em desempenho, segurança e introduzindo recursos modernos que otimizam o desenvolvimento de aplicações.

Uma das maiores forças do Java reside em sua portabilidade, encapsulada no princípio "Write Once, Run Anywhere" (WORA). Isso significa que o código-fonte Java compilado pode ser executado em praticamente qualquer plataforma que possua uma Java Virtual Machine (JVM), eliminando a necessidade de modificações específicas para cada ambiente. Essa característica o torna ideal para sistemas distribuídos e heterogêneos.

Além da portabilidade, Java destaca-se por sua robustez, segurança inerente e um vasto ecossistema de bibliotecas, frameworks e ferramentas. Essa riqueza de recursos acelera o desenvolvimento e oferece soluções prontas para desafios complexos. No desenvolvimento de aplicações web e empresariais, frameworks como Spring (incluindo o Spring Boot, que simplifica a criação de microserviços e APIs RESTful) e Jakarta EE (antigo Java EE) são pilares que facilitam a construção de sistemas escaláveis, eficientes e de alta performance. A forte tipagem, o gerenciamento automático de memória (garbage collection) e os recursos de concorrência integrados contribuem para a criação de aplicações mais estáveis e seguras. A comunidade ativa e o suporte contínuo da Oracle garantem que Java permaneça uma escolha confiável e relevante para desenvolvedores de todos os níveis de experiência. Sua aplicabilidade abrange desde aplicações corporativas críticas e big data até sistemas embarcados e o desenvolvimento de aplicativos móveis para Android.

3.2.3 Spring Boot

Spring Boot é um framework open-source, construído sobre o sólido fundamento do Spring Framework, e foi concebido pela Pivotal com o objetivo primordial de simplificar e agilizar o desenvolvimento de aplicações Java robustas e bem estruturadas, especialmente no ecossistema web. Ele se destaca por oferecer um conjunto abrangente de ferramentas e configurações predefinidas que automatizam e simplificam tarefas rotineiras, como o gerenciamento de dependências, a configuração automática de componentes, a implementação de segurança e a integração com diversos bancos de dados.

O framework adota a arquitetura MVC (Model-View-Controller), um padrão que promove a clara separação de responsabilidades, resultando em um código mais modular, organizado e de fácil manutenção. No contexto do SQLab, isso é fundamental para separar a lógica de negócios do backend (gerenciamento de turmas, exercícios e validação de queries) da apresentação no frontend. Entre seus recursos mais valorizados estão o Spring Data JPA, que facilita a manipulação

intuitiva de bancos de dados relacionais , o que é crucial para interagir com o PostgreSQL no SQLab; o Spring Security, que provê mecanismos robustos para autenticação e controle de acesso, essencial para diferenciar usuários Professores e Alunos e proteger as funcionalidades do sistema ; o Spring Boot Actuator, para monitoramento e coleta de métricas da aplicação em tempo real; e um suporte nativo e eficiente para testes automatizados.

Graças à sua forte comunidade de desenvolvedores e a uma documentação vasta e detalhada, o Spring Boot tornou-se uma das escolhas mais populares para o desenvolvimento de aplicações web em Java. Sua compatibilidade e otimização com a versão Java 17, uma LTS (Long-Term Support), garantem não apenas melhorias contínuas de desempenho e segurança, mas também o acesso a novas funcionalidades que tornam o processo de desenvolvimento ainda mais produtivo e alinhado às práticas modernas do mercado. No projeto SQLab, a escolha do Spring Boot reflete a busca por uma base tecnológica sólida, escalável e segura para o backend que gerencia toda a lógica do sistema e a interação com o banco de dados.

3.2.4 H2

O H2 é um sistema de gerenciamento de banco de dados relacional desenvolvido em Java, conhecido por sua leveza, rapidez e flexibilidade. É uma ferramenta de código aberto e gratuita, bastante utilizada em ambientes de desenvolvimento, testes e também em aplicações que exigem execução isolada e dinâmica de consultas SQL. Seu design simples e eficiente permite a criação de bancos de dados temporários e embarcados, o que o torna ideal para cenários onde se busca agilidade, paralelismo e controle.

Uma das principais características do H2 é o suporte ao modo embarcado (in-memory), no qual o banco de dados é executado diretamente na memória da aplicação. Isso permite que diferentes instâncias do banco sejam criadas e finalizadas de forma rápida e independente, sem depender de um servidor externo. Tal funcionalidade é especialmente útil em soluções que demandam a execução de múltiplas consultas SQL simultâneas, como em plataformas de ensino ou avaliação automática de comandos SQL, onde cada usuário pode interagir com seu próprio ambiente isolado.

Além disso, o H2 também pode ser executado no modo cliente-servidor, oferecendo maior flexibilidade conforme a necessidade da aplicação. Sua compatibilidade com a linguagem SQL padrão e a interface web interativa (H2 Console) contribuem para a usabilidade da ferramenta, permitindo que

desenvolvedores ou usuários visualizem e testem suas consultas de forma prática, diretamente no navegador.

O banco de dados H2 é frequentemente integrado a frameworks do ecossistema Java, como Spring Boot e Hibernate, mas também pode ser utilizado de forma independente em projetos de diferentes naturezas. Sua estrutura leve e inicialização rápida o tornam uma opção estratégica para aplicações que exigem a criação dinâmica de ambientes de banco de dados temporários, como é o caso de ferramentas educacionais interativas.

3.2.5 POSTGRESQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional (ORDBMS) de código aberto que tem se destacado no mercado por sua robustez, flexibilidade e conformidade com os padrões SQL. Lançado inicialmente em 1989 como parte do projeto POSTGRES na Universidade da Califórnia em Berkeley, o PostgreSQL evoluiu significativamente ao longo dos anos, tornando-se uma das escolhas mais populares entre desenvolvedores e empresas de todo o mundo. Sua arquitetura avançada, extensibilidade e forte foco em integridade de dados o tornam uma opção atraente para uma ampla variedade de aplicações, desde pequenas startups até grandes corporações.

Uma das principais razões para a ampla adoção do PostgreSQL é sua aderência rigorosa aos padrões SQL e seu suporte a uma vasta gama de funcionalidades avançadas. Ao contrário de alguns outros bancos de dados que podem sacrificar a conformidade com padrões em favor de funcionalidades proprietárias, o PostgreSQL mantém um compromisso firme com a padronização, o que facilita a portabilidade e a manutenção do código. Além disso, ele oferece suporte nativo para tipos de dados avançados, como JSON, XML e arrays, bem como para funções complexas, índices personalizados e operações geométricas, tornando-o extremamente versátil e capaz de atender a diversas necessidades de armazenamento e processamento de dados. Outro fator que contribui para a popularidade do PostgreSQL é sua extensibilidade. Desenvolvedores podem criar suas próprias funções, tipos de dados, operadores e índices personalizados, permitindo que o banco de dados seja adaptado para casos de uso específicos. Essa capacidade de extensão é suportada por uma comunidade ativa e vibrante que continuamente desenvolve novas extensões e melhorias. Ferramentas populares como PostGIS, que adiciona suporte a dados geoespaciais, e PL/pgSQL, uma linguagem procedural similar ao PL/SQL do Oracle, são exemplos de como a comunidade enriquece o ecossistema do PostgreSQL.

A confiabilidade e a integridade dos dados são também pilares fundamentais do PostgreSQL. Ele oferece um modelo de transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade), garantindo que todas as operações sejam completadas com sucesso e que o banco de dados permaneça em um estado consistente mesmo em casos de falha. Além disso, o PostgreSQL implementa controle de concorrência multiversionamento (MVCC), permitindo que múltiplas operações de leitura e escrita ocorram simultaneamente sem conflitos, o que melhora significativamente o desempenho em ambientes de alta concorrência.

Finalmente, o PostgreSQL é uma escolha atraente devido ao seu modelo de licenciamento de código aberto sob a licença PostgreSQL, uma licença permissiva que permite o uso, modificação e distribuição do software sem restrições significativas. Isso não apenas reduz os custos para empresas, mas também proporciona maior controle e flexibilidade sobre suas implementações de banco de dados. Grandes empresas, incluindo Apple, Fujitsu e Cisco, utilizam PostgreSQL em suas operações diárias, validando ainda mais sua confiabilidade e eficácia.

3.2.6 DRAW.IO

O draw.io é uma ferramenta online de diagramação e criação de fluxogramas, lançada em 2012, que permite aos usuários criar diagramas profissionais de forma intuitiva e colaborativa, sem a necessidade de instalar software adicional. Com draw.io, é possível acessar uma ampla variedade de elementos e formas pré-definidas, incluindo símbolos para diagramas de fluxo, organogramas, wireframes e diagramas UML. Além disso, o sistema permite importar e exportar diagramas em diferentes formatos, como XML, PNG e PDF, facilitando o compartilhamento e a colaboração com outras pessoas.

A natureza online do draw.io, aliada à sua integração com serviços de armazenamento em nuvem, torna-o uma opção conveniente para a criação e compartilhamento de diagramas em diferentes cenários e dispositivos. Os usuários podem acessar seus projetos de qualquer lugar, colaborar em tempo real com colegas e compartilhar facilmente os resultados finais. Isso é particularmente útil para equipes distribuídas geograficamente que precisam trabalhar juntas em tempo real. A flexibilidade e robustez do draw.io são complementadas por sua acessibilidade. Sendo uma ferramenta gratuita e de código aberto, o draw.io é uma opção atraente para indivíduos e organizações que procuram uma solução eficaz para a visualização e documentação de processos e ideias. Sua interface amigável e a vasta gama de funcionalidades permitem que usuários de todos os níveis de habilidade criem diagramas detalhados e precisos com facilidade.

3.2.7 EXCALIDRAW

Excalidraw é uma ferramenta online de diagramação e esboço que se destaca por sua simplicidade e versatilidade. Lançada inicialmente como um projeto de código aberto, ela permite que usuários criem diagramas e ilustrações de maneira intuitiva e colaborativa, sem a necessidade de instalação de software adicional. Sua interface limpa e minimalista facilita a criação rápida de esboços, fluxogramas, *wireframes* e outros tipos de diagramas, tornando-a uma escolha popular entre designers, desenvolvedores e equipes de projeto.

Um dos principais atrativos do Excalidraw é a sua abordagem de desenho à mão livre, que dá aos diagramas um visual natural e despretensioso. Isso é especialmente útil em fases iniciais de brainstorming e planejamento, onde a formalidade excessiva pode ser uma barreira à criatividade. Além disso, Excalidraw suporta colaboração em tempo real, permitindo que várias pessoas trabalhem juntas em um mesmo projeto, o que é essencial para equipes distribuídas e projetos colaborativos.

A integração com plataformas de armazenamento em nuvem como Google Drive e o suporte para exportação de diagramas em formatos como PNG e SVG ampliam a utilidade do Excalidraw, facilitando o compartilhamento e a incorporação de diagramas em outras ferramentas e documentos. A combinação de simplicidade, flexibilidade e colaboração faz do Excalidraw uma ferramenta poderosa para qualquer pessoa que precise criar e compartilhar visualizações claras e eficazes.

3.2.8 OBSIDIAN

O Obsidian é uma ferramenta versátil e gratuita, popularmente conhecida como um "segundo cérebro" digital, que atua como um ambiente de notas e gerenciamento de conhecimento baseado em arquivos Markdown locais. Lançado em 2020, ele se destaca pela sua capacidade de criar conexões ricas e interligadas entre diferentes ideias e documentos, utilizando links bidirecionais. Além de ser um poderoso editor de texto, o Obsidian permite a visualização dessas conexões em um "Grafo de Conhecimento" interativo, revelando a teia de relações entre as notas do usuário.

A flexibilidade do Obsidian o torna uma ferramenta valiosa em diversos contextos, incluindo o desenvolvimento de projetos. Para o propósito deste trabalho, o Obsidian foi fundamental na organização de ideias, no planejamento de funcionalidades e na criação e visualização de fluxogramas e diagramas. Utilizando plugins e a capacidade de incorporar outros elementos visuais, foi possível esboçar



e refinar os fluxos de processo e validação do SQLab, aproveitando sua interface intuitiva para mapear complexidades de forma clara e eficiente. Sua natureza offline e o armazenamento local dos arquivos garantem controle total sobre os dados, enquanto a vasta comunidade oferece plugins e temas que expandem ainda mais suas funcionalidades, adaptando-o a diversas necessidades.

4 APRESENTAÇÃO DO SISTEMA

A plataforma SQLab foi desenvolvida como uma resposta prática aos desafios enfrentados no ensino de SQL, especialmente no que se refere à carência de oportunidades para aplicação prática dos conteúdos vistos em sala de aula. Muitas instituições de ensino ainda enfrentam dificuldades ao adotar abordagens que conciliem teoria e prática, especialmente pela ausência de ferramentas acessíveis e flexíveis para execução de atividades com bancos de dados reais (Clark, 1994; Siemens, 2005). Nesse contexto, o SQLab se propõe a ser uma solução funcional e eficiente, voltada para professores e estudantes de cursos relacionados à tecnologia da informação e desenvolvimento de sistemas.

O foco central do SQLab está em oferecer um ambiente educacional acessível onde professores possam configurar e gerenciar atividades práticas de SQL de forma personalizada. A plataforma permite o cadastro de turmas, estudantes e listas de exercícios, além da criação de exercícios com comandos SQL. Onde os alunos escrevem consultas que são executadas e validadas em tempo real, utilizando bancos de dados temporários para simular situações próximas às encontradas no mercado.

Professores também podem definir uma ou mais respostas corretas esperadas para os exercícios, o que possibilita uma verificação automática das submissões dos alunos. Embora o acompanhamento dos resultados seja básico, a plataforma permite aos docentes visualizar os comandos submetidos e identificar se os resultados obtidos pelos alunos correspondem ao esperado, auxiliando na correção manual e no entendimento do progresso individual.

O SQLab foi projetado com ênfase na estabilidade da aplicação, na escalabilidade para suportar múltiplas execuções simultâneas de consultas SQL e na autonomia dos professores para criação de atividades personalizadas. Para garantir uma experiência prática eficiente e sem complexidades técnicas para os usuários, a plataforma utiliza o banco de dados H2 em memória, o que permite a execução isolada e paralela de comandos SQL, mesmo em ambientes com grande volume de acessos. A proposta do sistema visa em possibilitar no futuro a implementação de recursos como gamificação, fóruns de discussão ou sugestões automáticas de dicas, desenvolvendo assim a ideia central de oferecer um ambiente acessível, prático e

configurável para o ensino e aprendizagem da linguagem SQL e também disponibilizar aos professores o acompanhamento aos alunos de uma forma otimizada.

Ao oferecer uma interface simples, porém robusta, para o ensino e a prática de SQL, o SQLab visa não apenas facilitar o trabalho docente, mas também melhorar a experiência dos alunos com atividades práticas, diminuindo barreiras técnicas e promovendo um aprendizado mais significativo. Como destacam Boulos et al. (2020), iniciativas que aproximam o ensino acadêmico das necessidades reais do mercado são fundamentais para a formação de profissionais preparados para os desafios da era digital.

4.1 MODELAGEM DO PROJETO

4.1.1 DIAGRAMA DE CASO DE USO

O diagrama de casos de uso é essencial para a compreensão dos requisitos funcionais de um sistema, representando as interações entre atores e os diferentes casos de uso que o sistema deve suportar. Ele proporciona uma visão clara das funcionalidades esperadas pelo usuário final, facilitando o alinhamento entre desenvolvedores e stakeholders. Para uma visualização detalhada deste diagrama, consulte o apêndice A. As especificações completas de cada caso de uso estão disponíveis no APÊNDICE C.

4.1.2 TABELA DE REQUISITOS

A tabela de requisitos é uma ferramenta fundamental na engenharia de requisitos, onde são listadas e detalhadas todas as funcionalidades e características que o sistema deve ter para atender às necessidades dos usuários e stakeholders. Ela proporciona uma base clara e estruturada para o desenvolvimento do sistema, garantindo que todas as especificações sejam consideradas e implementadas de maneira adequada. Consulte o APÊNDICE B para uma visão completa dos requisitos detalhados do sistema.

4.1.3 DIAGRAMA DE CLASSES

O diagrama de classes descreve a estrutura estática do sistema, mostrando as classes, seus atributos, métodos e relacionamentos. Ele é fundamental para o projeto orientado a objetos, ajudando a definir a arquitetura do software e a visualizar a organização das entidades no sistema. Para uma representação completa deste diagrama, consulte o APÊNDICE D.

4.1.4 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência ilustra a interação entre objetos em uma sequência específica de eventos, mostrando como os objetos colaboram para realizar uma funcionalidade específica do sistema. Ele é crucial para capturar o comportamento dinâmico do sistema, facilitando o entendimento do fluxo de informações e controle entre os objetos. Para uma visualização detalhada deste diagrama, consulte o APÊNDICE E.

4.2 ARQUITETURA DO SISTEMA

Para o sistema SQLab, a arquitetura está sendo meticulosamente planejada para garantir eficiência, escalabilidade e segurança, fundamentais para suportar suas funcionalidades educacionais avançadas. A abordagem cliente-servidor adotada se divide entre o frontend desenvolvido em Angular e o backend em Spring Boot, integrados por meio de uma arquitetura de requisições HTTP RESTful.

No frontend, o Angular 17 será implementado para oferecer uma interface de usuário responsiva e dinâmica. A escolha por Angular não apenas facilita a criação de componentes reutilizáveis e a gestão eficiente do estado da aplicação com RxJS e NgRx, mas também proporciona uma experiência de usuário fluida ao permitir a edição de consultas SQL em tempo real e a visualização imediata dos resultados. A interface intuitiva do SQLab será projetada para guiar os usuários através das atividades de SQL de forma clara e eficiente. Já no lado do servidor, o Spring Boot será utilizado para desenvolver o web service que gerencia a lógica de negócios do sistema. Este framework Java oferece uma estrutura robusta baseada em padrões de projeto modernos, promovendo a separação de responsabilidades e facilitando a manutenção do código. A escolha do Spring Boot também se deve à sua capacidade de criar APIs RESTful de maneira eficiente, garantindo uma comunicação segura e escalável entre o front end e o banco de dados PostgreSQL.

A integração entre frontend e backend será realizada através de requisições HTTP RESTful, onde o frontend enviará requisições para o backend para manipulação de dados, execução de consultas SQL e gestão de usuários. Essa arquitetura não só simplifica a comunicação entre os componentes do sistema, mas também permite que o SQLab ofereça uma experiência de usuário consistente e responsiva, essencial para o aprendizado eficaz de SQL.

Quanto ao banco de dados, o PostgreSQL foi selecionado devido à sua robustez, escalabilidade e suporte avançado para operações SQL complexas. Este SGBD será responsável por armazenar e gerenciar os dados dos usuários, atividades, resultados de consultas e outras informações críticas para o funcionamento do SQLab. A integração entre Spring Boot e PostgreSQL será facilitada pelo uso do Hibernate como ORM (Object-Relational Mapping), o que simplifica o acesso e a manipulação dos dados de forma eficiente e segura.

Embora o SQLab utilize bancos de dados H2 em memória por padrão para garantir isolamento e escalabilidade, uma implementação futura poderá oferecer a opção de bancos persistentes, permitindo que os alunos mantenham o estado do banco entre sessões. Isso possibilitará a construção de conhecimento ao longo de várias tentativas e dias, aumentando a flexibilidade do aprendizado.

Por fim, a arquitetura do SQLab é desenhada para atender aos objetivos educacionais de proporcionar uma plataforma prática e interativa para o ensino e para garantir que o sistema seja escalável, seguro e de alto desempenho. Essa abordagem estruturada oferecerá uma experiência de usuário otimizada e adaptável às necessidades dos alunos e educadores.

4.3 FUNCIONALIDADES

A seção a seguir tem como principal objetivo apresentar as funcionalidades do sistema proposto de forma clara e objetiva. Por meio de um tutorial descritivo, serão apresentados todos os protótipos de telas, junto de uma breve explicação sobre cada componente e fluxos.

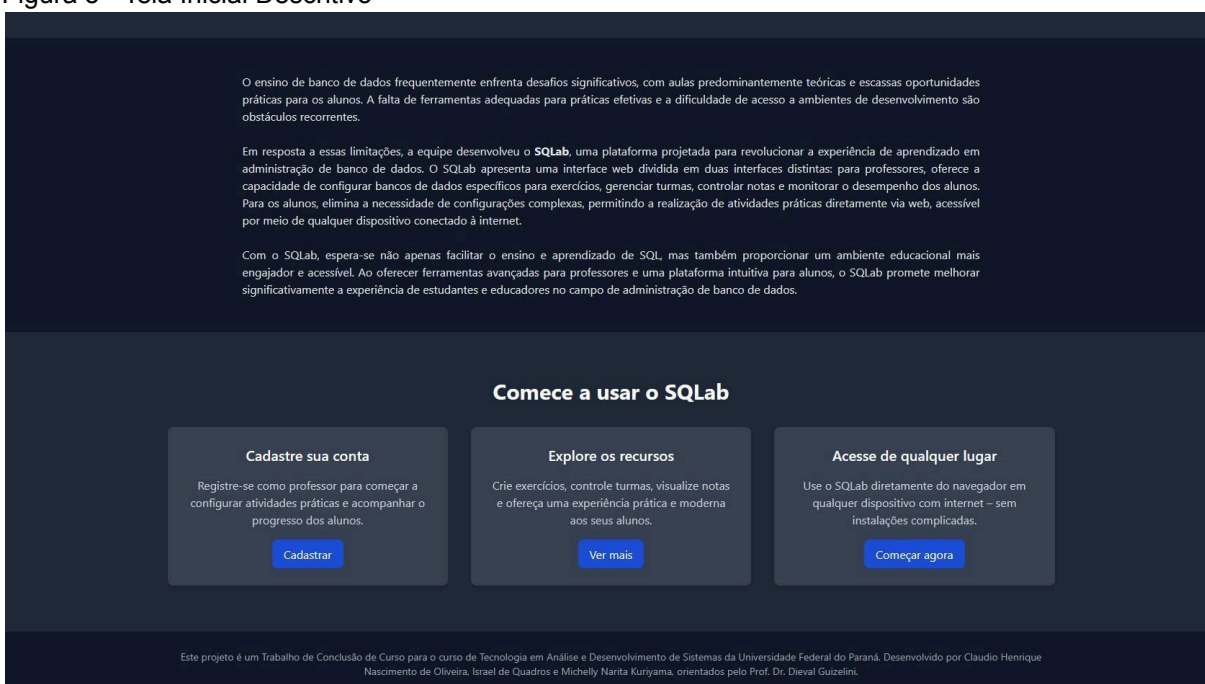
Primeiramente ao acessar o sistema, o usuário é direcionado para uma tela principal (Figura 2), onde possui a visualização de uma breve descrição do sistema (Figura 3) e seu objetivo. Nesta, há o acesso para o auto cadastro e login, sendo redirecionados ao selecionar o botão de “Saiba mais” ou “Login / Cadastro”.

Figura 2 - Tela Inicial



UC01 – Tela inicial

Figura 3 - Tela Inicial Descritivo



UC01 – Tela inicial

Ao selecionar para “Cadastrar”, ou o botão de “Login / Cadastro”, o fluxo segue com a primeira ação que deve ser de um professor, com seu auto cadastro (Figura 4). A tela é composta de um formulário com quatro campos: “Nome Completo”, “E-mail”, “Senha”, “Confirmação de senha”, seguidos de um botão “Confirmar”. Ao preencher todos os campos e clicar no botão, o usuário é registrado no banco de dados e o sistema redireciona para a tela de login.

Figura 4 - AutoCadastro

Escolha o tipo de cadastro

[Sou Aluno](#)
[Sou Professor](#)

[Voltar](#)

Autocadastro de Professor

Nome Completo

Email

Senha

Confirme sua senha

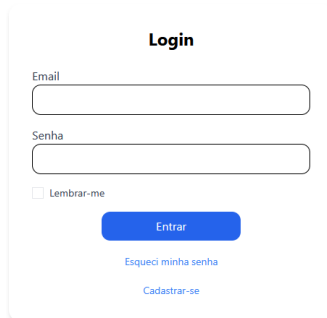
[Confirmar](#)

UC02 – Auto cadastro de professor

Na tela de login (Figura 5) , o usuário deverá se logar com os mesmos dados com os

quais se cadastrou, isso é o “E-mail” e a “Senha”.

Figura 5 - Login



Login

Email

Senha

☐ Lembrar-me

Entrar

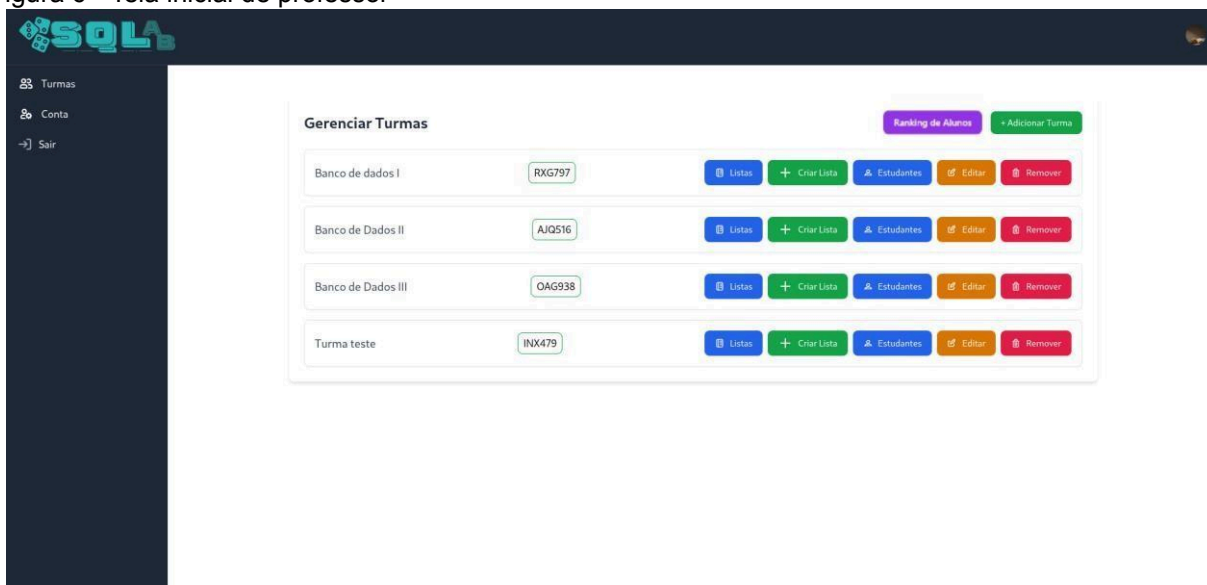
[Esqueci minha senha](#)

[Cadastrar-se](#)

UC03 – Login

Assim que realizar o login, o mesmo identificará o tipo de usuário que foi logado. No caso do professor, a tela será redirecionada para a tela de gerenciamento e visualização das turmas (UC05 – Listar Turmas, Figura 6). Na navegação superior, haverá os botões de “Ranking de Alunos”, responsável por fornecer ao professor a visualização do desempenho dos alunos, um botão “Adicionar Turma” (UC04 – Cadastrar turma) e uma tabela com todas as turmas cadastradas e funções respectivas a essa turma, como visualizar as “Listas”, ou adicionar automaticamente uma lista para a turma com o botão de “Criar Lista”, contempla também a opção de visualizar os alunos no botão de “Estudantes”, alterar o nome da turma através do botão de “Editar” e excluir uma turma pelo botão de “Remover”.

Figura 6 - Tela inicial de professor



Gerenciar Turmas

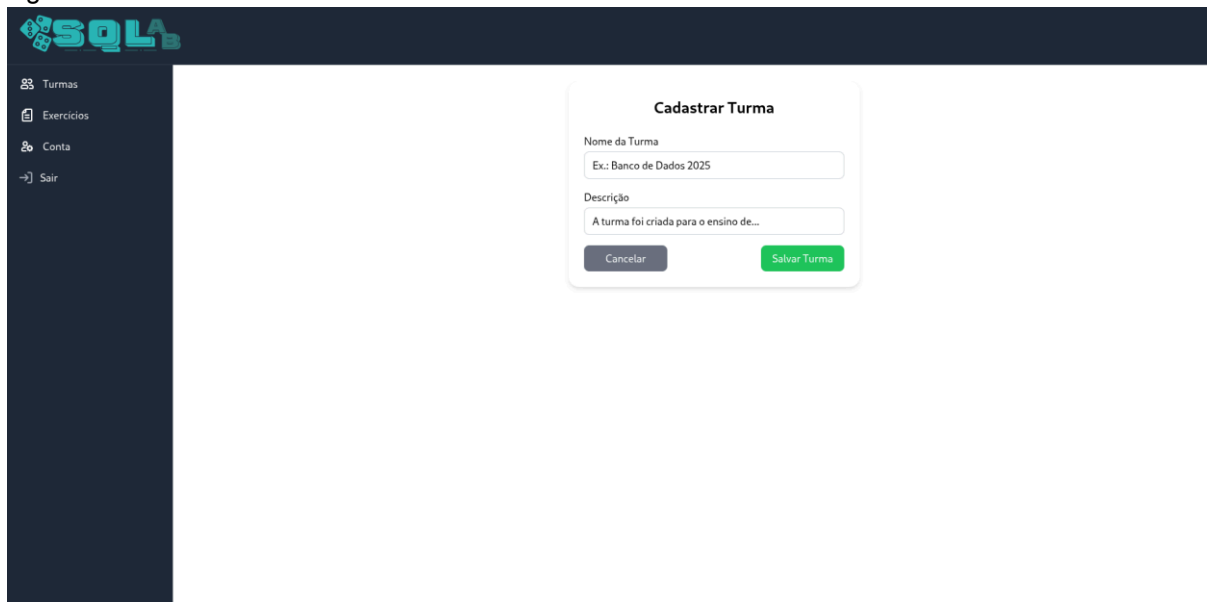
[Ranking de Alunos](#) [Adicionar Turma](#)

Turma	ID	Listas	Criar Lista	Estudantes	Editar	Remover
Banco de dados I	RXG797	Listas	+ Criar Lista	Estudantes	Editar	Remover
Banco de Dados II	AJQ516	Listas	+ Criar Lista	Estudantes	Editar	Remover
Banco de Dados III	OAG938	Listas	+ Criar Lista	Estudantes	Editar	Remover
Turma teste	INX479	Listas	+ Criar Lista	Estudantes	Editar	Remover

UC05 – Listar turmas

O professor poderá realizar o cadastro de turma ao clicar em “Adicionar Turma”, assim será redirecionado para uma tela com o campo de “Nome da Turma” e “Descrição”. A página de edição de turma (UC06 – Editar turmas, Figura 7) é semelhante à de cadastro, mudando apenas o fato de vir preenchida com os dados da turma referente.

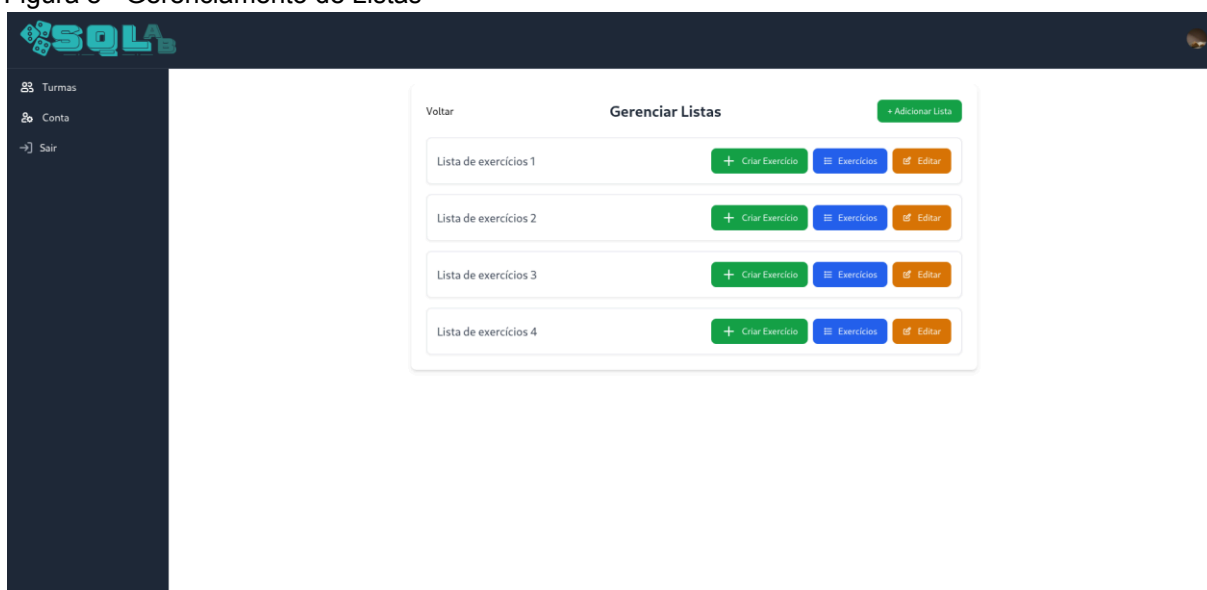
Figura 7 - Cadastro de Turma



UC04 – Cadastrar Turma

Ao retornar para a página de turmas, o professor poderá então visualizar as turmas criadas, e assim, informar aos seus alunos o código para a matrícula, qual se encontra ao lado do nome da turma. Os alunos então, terão de informar esse código para poder usarem o sistema e realizar as listas fornecidas pelo professor nessa turma. Ao clicar em “Listas” na tela de gerenciamento de turmas, o professor terá acesos a tela de “Gerenciar Listas” (Figura 8) daquela turma, onde poderá criar diretamente um exercício, através do botão de “Criar Exercício”, ou visualizar os exercícios já criados, através do botão de “Exercícios” ou editar o título e descrição dessa lista, através do botão de “Editar”.

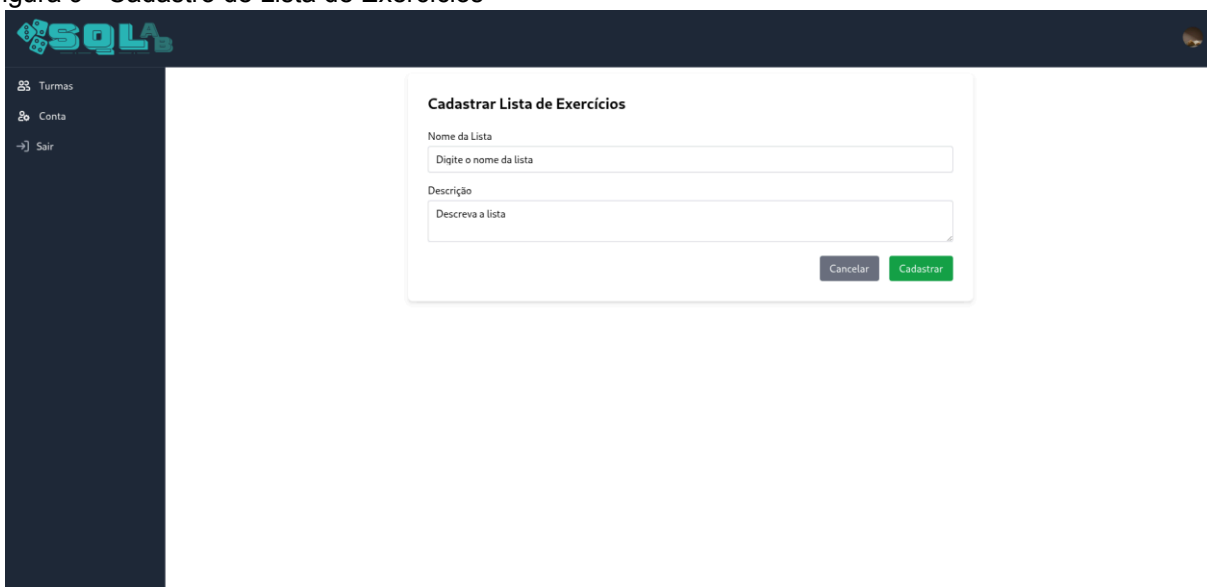
Figura 8 - Gerenciamento de Listas



UC12 - Listar listas de exercícios

Ao clicar em adicionar lista de exercício, o sistema irá redirecionar para a tela de cadastro de lista (Figura 9), onde haverá os campos de “Nome da Lista” e “Descrição”. A tela de editar lista (UC14 – Editar listas de exercícios) é semelhante à de cadastro, alterando apenas o fato de vir preenchida com os dados da lista a ser alterada.

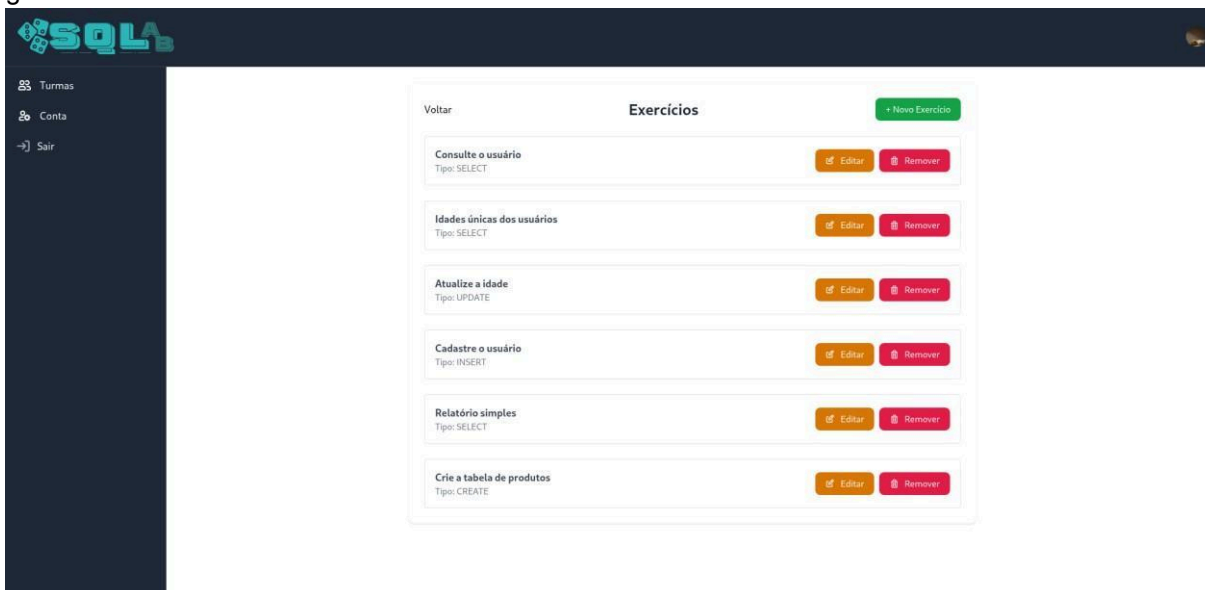
Figura 9 - Cadastro de Lista de Exercícios



UC13 - Cadastrar listas de exercícios

Ao clicar em exercícios, o professor será redirecionado a uma tela com os exercícios cadastrados nessa turma (Figura 10). Ao clicar remover, poderá remover um exercício (UC08 – Remover exercícios).

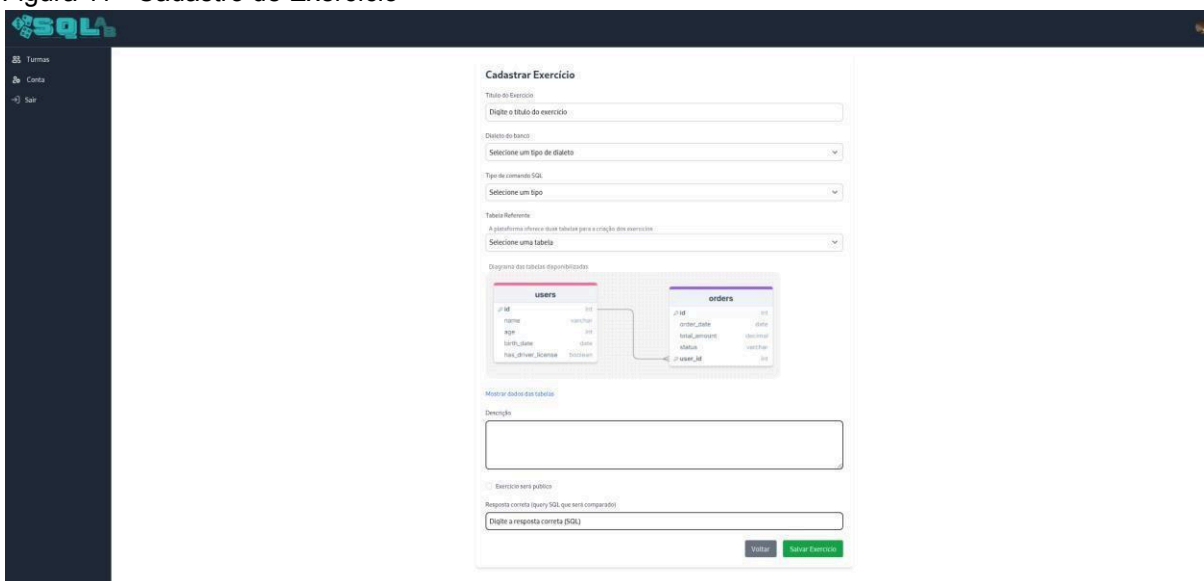
Figura 10 - Exercícios Cadastrados



UC05 – Listar Exercícios

Para As telas de cadastro (Figura 11) e edição (UC07 – Editar exercício) de exercício seguem o mesmo padrão, sendo que quando a de edição é aberta, os dados registrados no banco de dados já aparecem preenchidos. A tela é composta por um formulário contendo “Título do Exercício”, selecionar o “Dialeto do banco”, onde o professor poderá escolher entre “PostgreSQL” ou “MySQL”, campo para selecionar o “Tipo de comando SQL”, onde poderá escolher entre os tipos de select, insert, create, update, delete e drop, a tabela referente, que deverá ser uma das tabelas fornecidas pelo sistema, e que pode ser incrementada ao longo do tempo da aplicação, neste início, possuirá apenas a de users e orders, com um diagrama simples para que o professor saiba que dados estarão disponibilizados até dado momento, um campo para descrição do exercício, com checkbox de “Exercício será público”, implementado visando aplicações futuras em que o professor poderá utilizar de exercícios pré cadastrados para cada lista, e o campo mais fundamental para nossa aplicação, " Resposta correta" onde o professor deverá colocar a query que servirá de validação para a resposta do aluno.

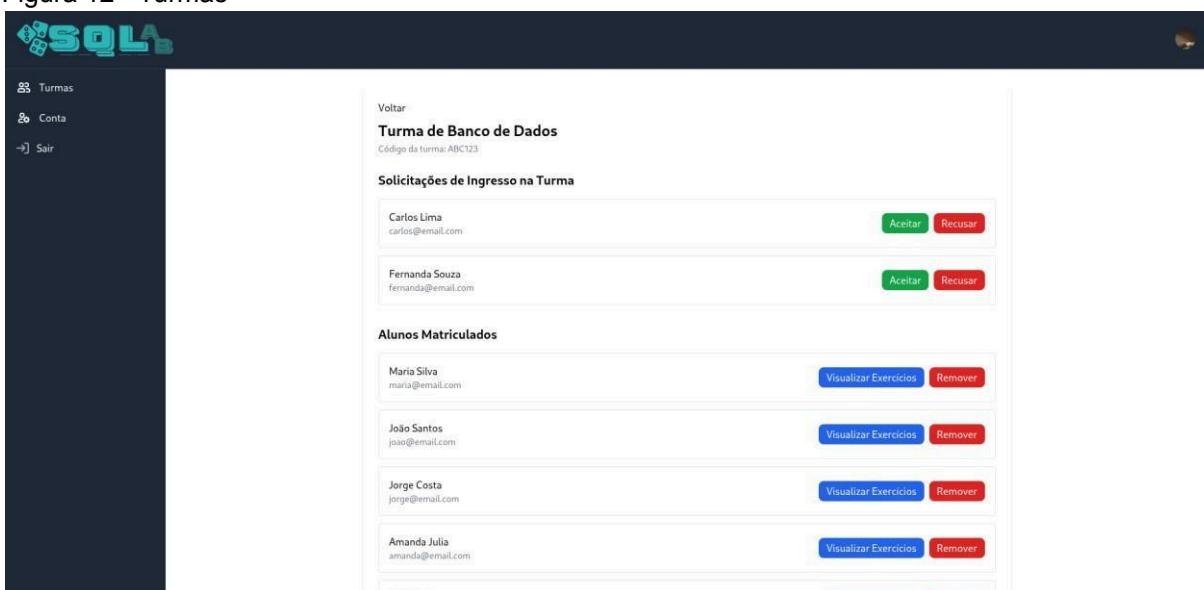
Figura 11 - Cadastro de Exercício



UC06 – Cadastrar exercício

Finalizando assim o fluxo do cadastro de lista de exercícios, e partindo da tela de turmas (Figura 12), o professor também terá a possibilidade de gerenciar os alunos de cada turma, clicando no botão de “Estudantes”, o sistema redireciona para uma tela possuindo os dados da turma, seu nome e descrição, e duas listas, uma com as solicitações de ingresso pendentes, e na outra, os alunos definitivamente matriculados, esses possuindo já o acesso para a resolução das listas de exercícios da turma. Nesta tela o professor pode recusar o ingresso dos alunos, assim como excluir os alunos (UC16 – Remover alunos).

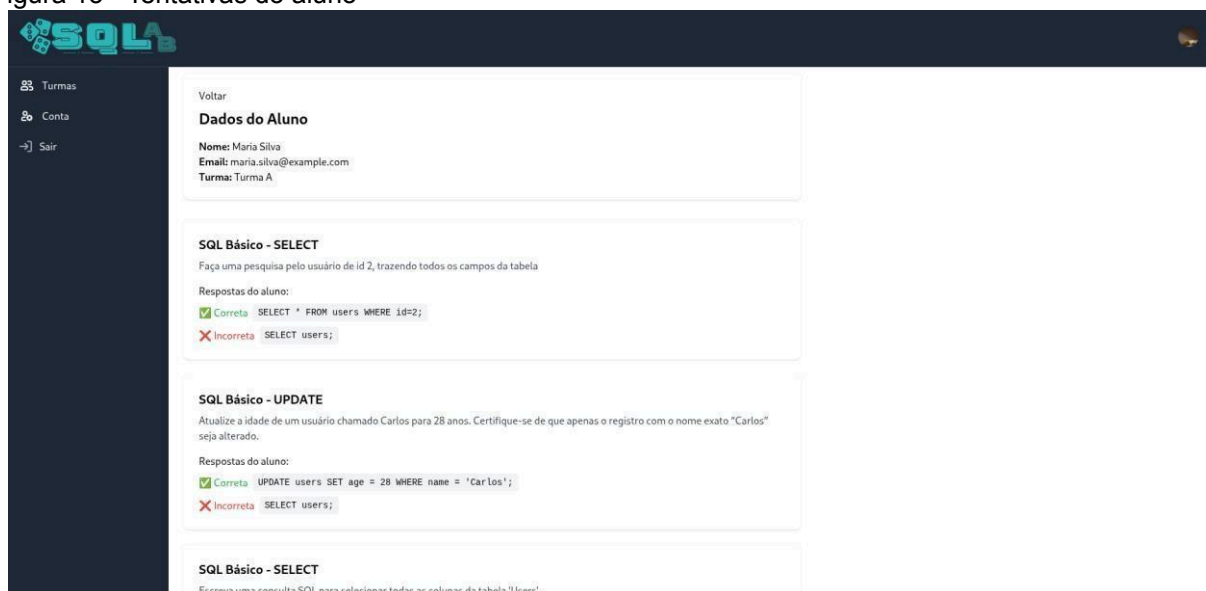
Figura 12 - Turmas



UC15 – Listar alunos

Na tela de listagem dos alunos também é possível visualizar as tentativas de exercícios de cada aluno (Figura 13), apresentando primeiramente os dados do aluno, como e-mail, nome, turma referente ao que está sendo apresentado, os enunciados dos exercícios, seu nome, o tipo, a resposta do aluno, a resposta correta e a resposta incorreta, sendo que se a resposta do aluno estiver correta, não será apresentada a resposta incorreta.

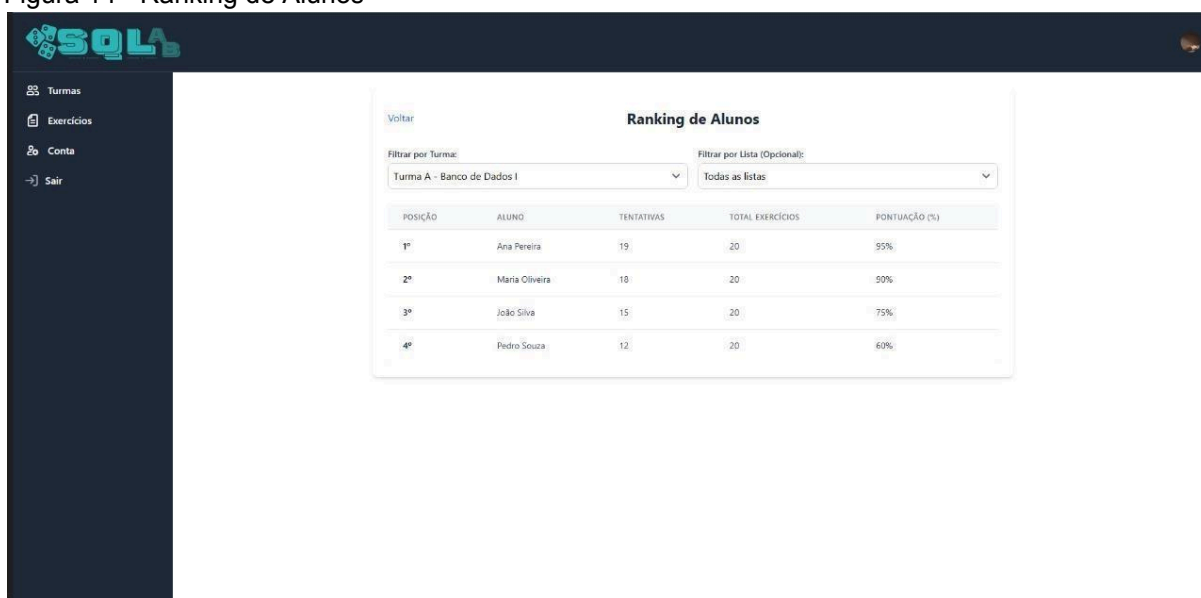
Figura 13 - Tentativas do aluno



UC14 – Visualizar tentativas de exercícios

Também a partir da tela de listagem de turmas, é apresentada a ferramenta de acompanhamento principal do professor. Clicando em “Ranking de Alunos” (Figura 14) o professor terá acesso a visualização de um relatório dos alunos no geral, podendo acompanhar o desempenho destes, através dos exercícios resolvidos, suas tentativas, e conforme os exercícios em que houve acertos, gera sua pontuação em porcentagem. Esse relatório pode ser filtrado por turma e pela lista em que será considerado tais dados.

Figura 14 - Ranking de Alunos

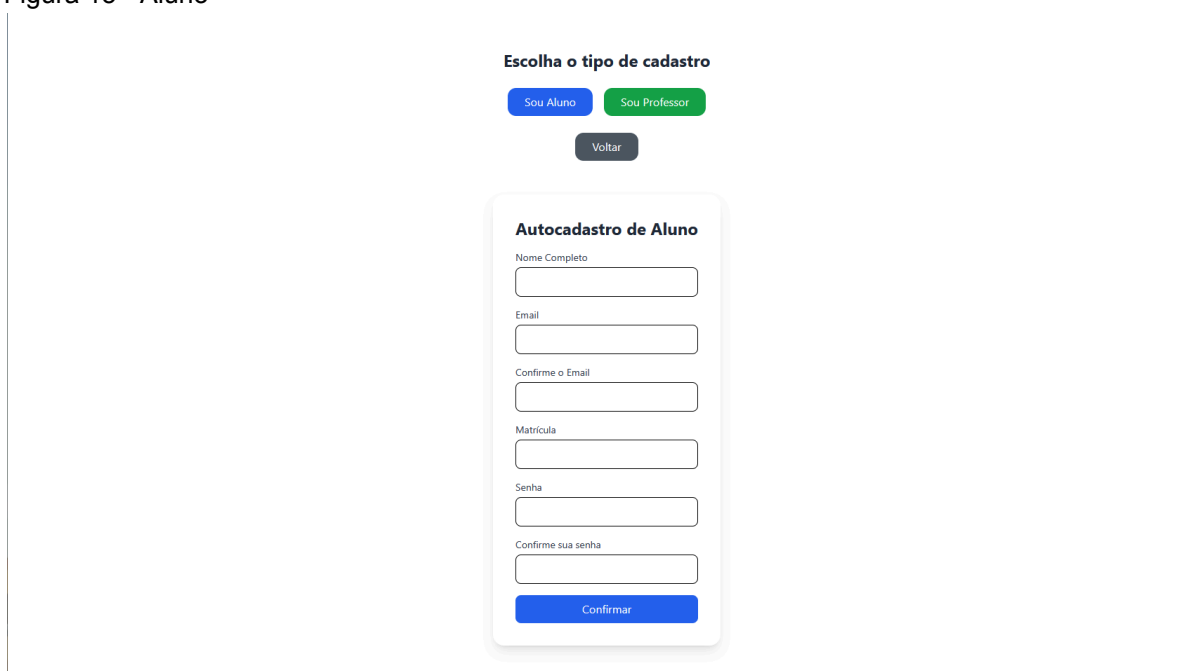


Posição	Aluno	Tentativas	Total Exercícios	Pontuação (%)
1º	Ana Pereira	19	20	95%
2º	Maria Oliveira	18	20	90%
3º	João Silva	15	20	75%
4º	Pedro Souza	12	20	60%

UC18 – Visualizar relatório de desempenho

Referente ao fluxo do professor, essas são as ações possíveis e mapeadas. Retornando ao auto cadastro do aluno (Figura 15), apenas temos alguns campos diferenciais. Neste, encontra-se os campos “Nome Completo”, “Email”, “Confirme o Email”, “Matrícula”, “Senha” e “Confirme sua senha”. Assim que submetido pelo botão de confirmar, o sistema salvará os dados e retornará a tela de login. Assim que o aluno logar, o sistema será redirecionado para a tela de home do aluno.

Figura 15 - Aluno



Escolha o tipo de cadastro

[Sou Aluno](#) [Sou Professor](#)

[Voltar](#)

Autocadastro de Aluno

Nome Completo

Email

Confirme o Email

Matrícula

Senha

Confirme sua senha

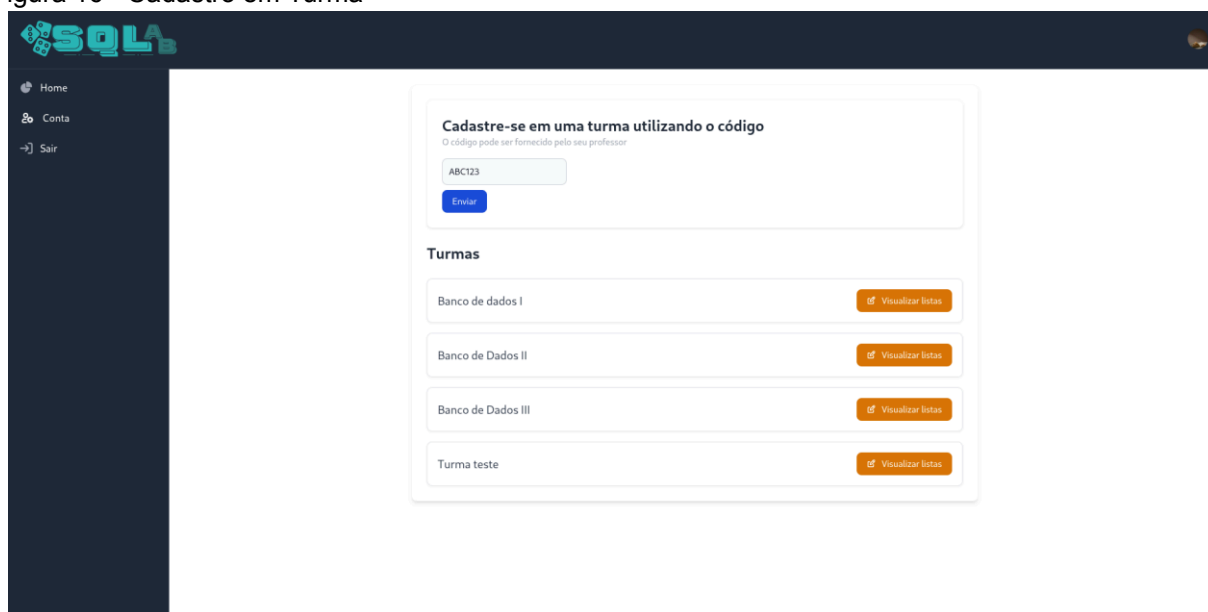
[Confirmar](#)

UC19 – Auto cadastro do aluno

Ao acessar o home do aluno, já é apresentado ao mesmo um input para o aluno se cadastrar em uma turma através do código (Figura 16), anteriormente gerado pelo professor (e que, portanto, deve ser informado pelo mesmo), e assim que clicado em

enviar, a solicitação de ingresso do aluno será enviada ao professor, que pode recusar ou aceitar a mesma.

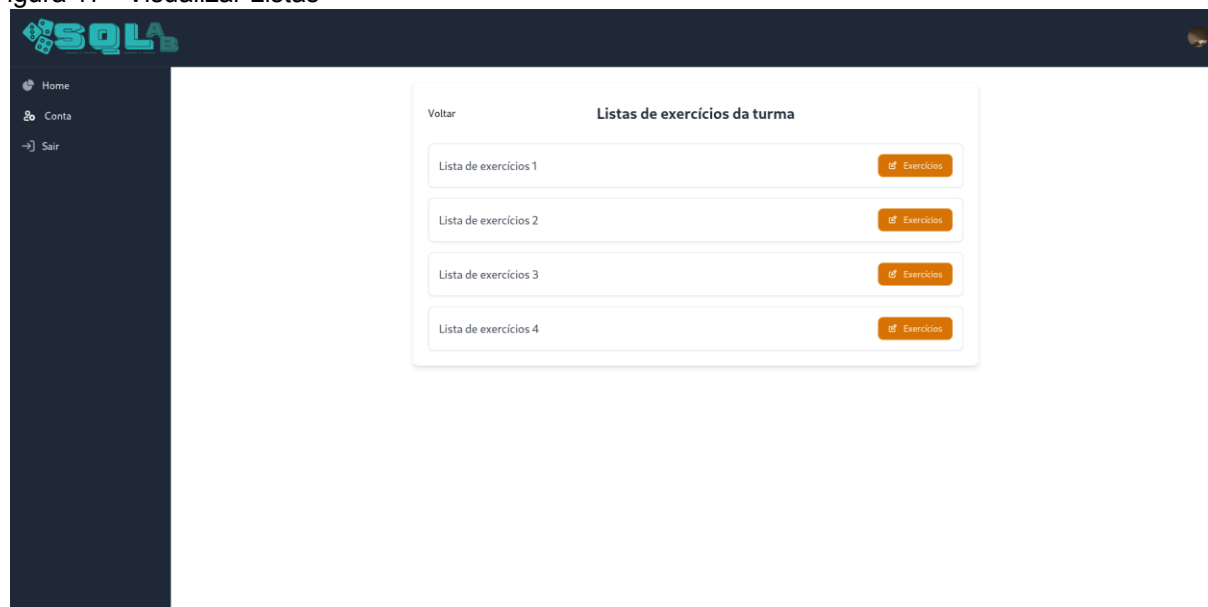
Figura 16 - Cadastro em Turma



UC20 – Inscrever-se em turma e UC21 – Visualiza turmas

Ao clicar em “Visualizar listas” (Figura 17), o aluno será redirecionado para a tela com todas as listas de exercícios da turma correspondente. Nesta, há um botão para cada lista, que redirecionará para os exercícios da lista em questão.

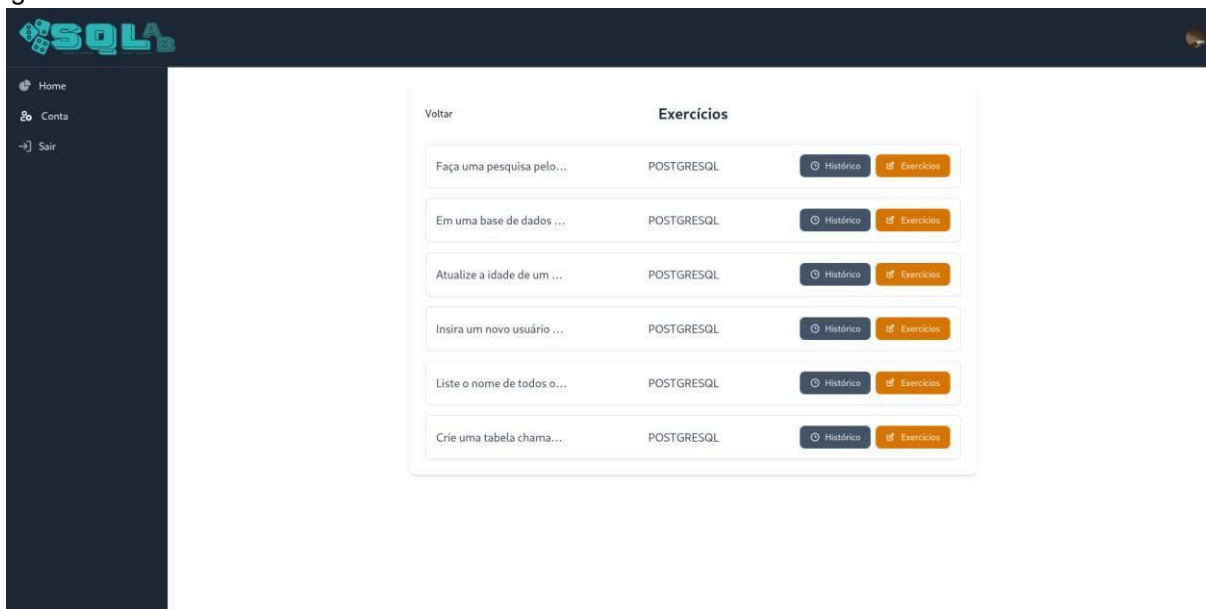
Figura 17 - Visualizar Listas



UC22 – Visualiza listas de exercícios

Ao clicar em “Exercícios”, o aluno será redirecionado para a tela com todas os exercícios da lista correspondente (Figura 18). Nesta, há um botão para cada exercício, que redirecionará para os exercícios da lista em questão.

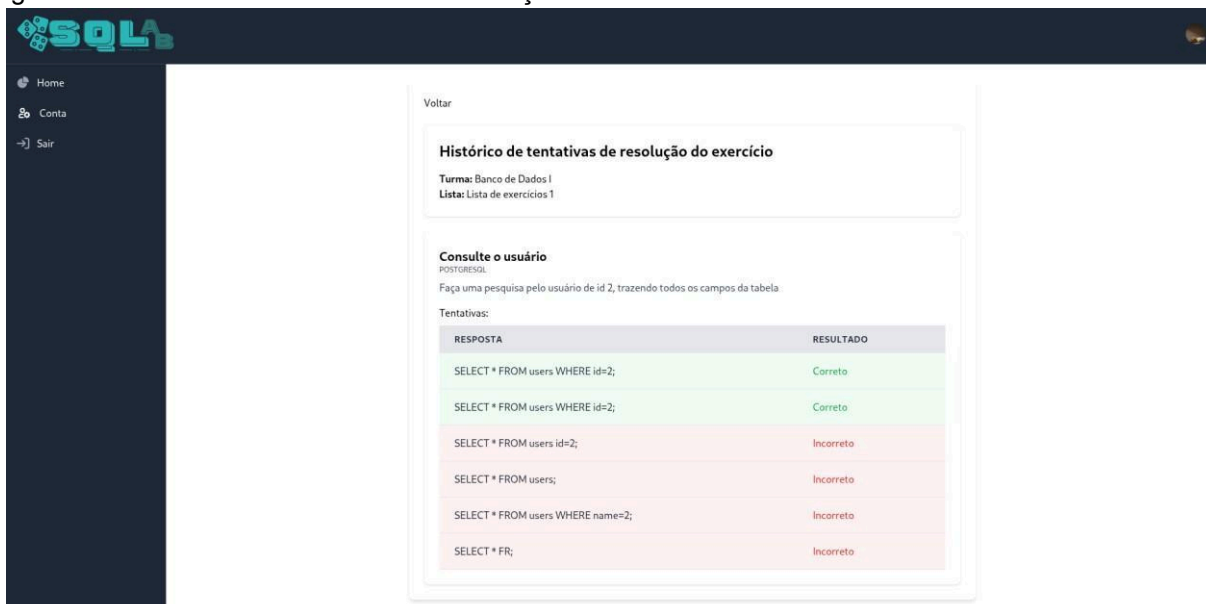
Figura 18 - Exercícios Cadastrados



UC23 – Visualiza exercícios de uma lista

Ao clicar no nome de alguma lista que está pendente de entrega ou que ainda não foi iniciada, o sistema abre a tela de resolução da lista. Com uma lista de todos os exercícios cadastrados naquela lista. Ao selecionar o botão de “Histórico” de um exercício, é apresentada uma tela com as tentativas de resolução do exercício (Figura 19), listando a turma, o nome da lista, o nome do exercício, o dialeto, o enunciado e por fim, uma lista com as respostas fornecidas, e se estão corretas ou não.

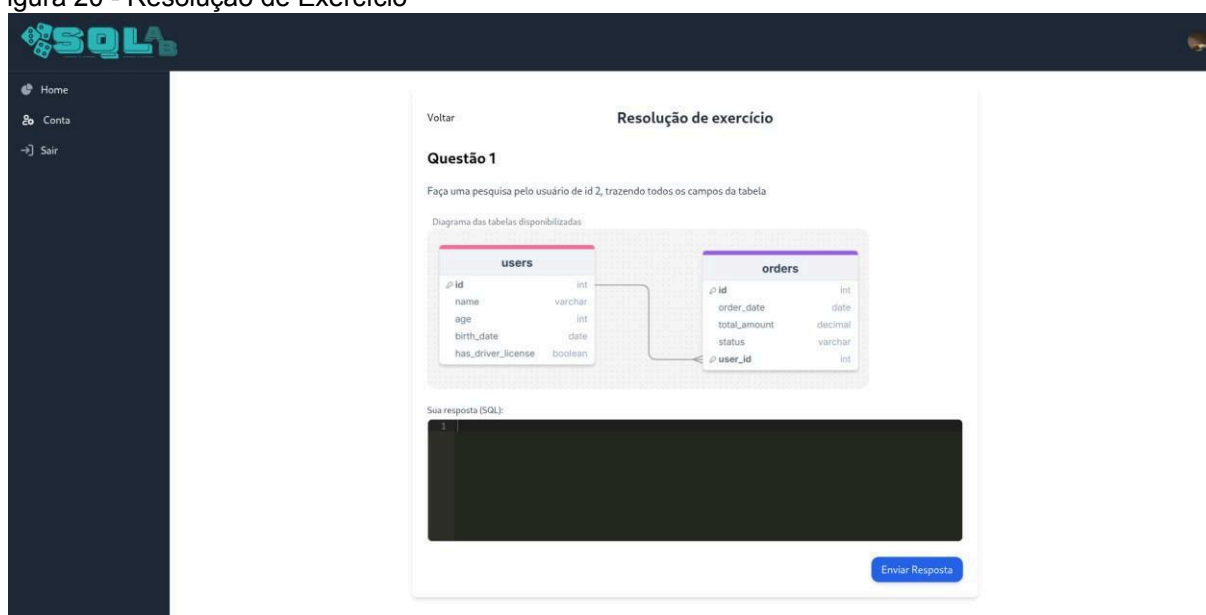
Figura 19 - Histórico de Tentativas de Resolução do Exercício



UC24 - Visualizar histórico do exercício

Ao clicar em “Exercício”, o aluno é redirecionado para a tela de resolução do exercício (Figura 20), projetada com base em princípios de User Experience (UX) para garantir clareza e atratividade. A interface minimalista exibe, na parte superior, o nome do exercício e um botão de navegação para voltar à listagem de exercícios. O enunciado, cadastrado pelo professor, é apresentado de forma clara e destacada. Um diagrama de entidade e relacionamento do banco é exibido de maneira simplificada, com opção de expandi-lo apenas se necessário. O campo “Sua resposta (SQL)” é centralizado, com uma área de texto ampla e intuitiva para escrita da query. Após clicar em “Enviar Resposta”, o sistema valida a query e retorna um feedback imediato, exibido em uma seção limpa com indicação de correção, resultados da query e, se aplicável, o número de linhas afetadas, evitando sobrecarga visual.

Figura 20 - Resolução de Exercício



UC25 – Resolver exercício

Para proporcionar um aprendizado interativo e com feedback imediato, o SQLLab emprega um fluxo de processamento de respostas do aluno, que varia conforme o tipo de comando SQL submetido.

Ao submeter uma questão no SQLLab, o sistema inicia um processo de validação para fornecer feedback preciso ao aluno. Independentemente do tipo de comando SQL (DML ou DDL), o fluxo geral envolve a criação de pelo menos um ambiente de banco de dados isolado para a execução da query. Utilizando instâncias em memória do banco H2 para garantir eficiência e isolamento. A validação, então, compara os resultados dessas execuções, e o sistema retorna um feedback com a correção e informações relevantes sobre a operação.

A seguir uma explicação sobre cada tipo de comando implementado.

- Comando SELECT: A validação se concentra em comparar os conjuntos de resultados das consultas. Após a execução da query do aluno e da query de resposta esperada, ambos os resultados são padronizados para o formato de lista de mapas. A comparação verifica se esses resultados são idênticos em conteúdo e estrutura. O retorno inclui a indicação de correção, o resultado da query do aluno e zero linhas afetadas, pois SELECT não modifica dados. Consulte o APÊNDICE G para o fluxograma completo de questões SELECT.
- Comandos INSERT, UPDATE e DELETE: O processo de validação verifica a modificação dos dados na tabela. Após a execução das queries (do aluno e da resposta) em seus respectivos ambientes, o sistema executa uma query SELECT na tabela em questão (cujo nome é informado no cadastro do exercício) em ambos os ambientes. Os resultados dessas duas queries SELECT são então convertidos para o formato de lista, padronizando a representação dos dados após a operação. A etapa crucial é a comparação dessas listas para verificar se o estado final da tabela, no ambiente do aluno, é idêntico ao estado final da tabela após a execução da query correta do professor. O retorno detalhado ao aluno inclui um indicador de correção, o estado final da tabela via SELECT do ambiente do aluno e o número de linhas afetadas pela operação. Consulte o APÊNDICE H para o fluxograma completo de questões INSERT. Consulte o APÊNDICE I para o fluxograma completo de questões UPDATE. Consulte o APÊNDICE K para o fluxograma completo de questões DELETE.

Para comandos que alteram a estrutura do banco de dados, como CREATE e DROP, o SQLab foca na validação das modificações no esquema. (Figura 21, Figura 22)

- **Comando CREATE:** A validação se concentra na estrutura da tabela criada. Após a execução das queries (do aluno e da resposta) em ambientes separados, o SQLab inspeciona as tabelas a fim de compará-las. O sistema obtém os metadados completos dos dois objetos, e a comparação é feita entre os schemas das colunas obtidas do aluno e da tabela esperada como resposta, verificando atributos como nome da tabela, nomes e tipos das colunas, entre outros detalhes estruturais relevantes. O retorno inclui o indicador de correção, a tabela completa criada pelo aluno (via SELECT) e zero linhas afetadas, já que CREATE não manipula dados. Consulte o APÊNDICE J para o fluxograma completo de questões CREATE.
- **Comando DROP:** A validação principal é a verificação da ausência da estrutura do banco de dados. Após a execução das queries (do aluno e da resposta) em ambientes separados, o sistema verifica se a entidade que deveria ter sido removida realmente deixou de existir em cada ambiente. A validação final compara esses resultados para confirmar se a condição de inexistência da tabela no ambiente do estudante corresponde à condição esperada da resposta do professor. O retorno inclui o indicador de correção, uma confirmação da exclusão (ou não) da tabela e zero linhas afetadas, pois DROP não altera dados. Consulte o APÊNDICE L para o fluxograma completo de questões DROP.

Após realizada a validação, o sistema apresenta ao aluno se a resposta está correta ou incorreta, e retorna também o retorno do SQL.

Figura 21 - Resolução de Exercício

- Home
- Conta
- Sair

Voltar

Resolução de exercício

Questão 1

Faça uma pesquisa pelo usuário de id 2, trazendo todos os campos da tabela

Diagrama das tabelas disponibilizadas

users

- id: int
- name: varchar
- age: int
- birth_date: date
- has_driver_license: boolean

orders

- id: int
- order_date: date
- total_amount: decimal
- status: varchar
- user_id: int

Sua resposta (SQL):

```
1. SELECT * FROM users WHERE id=2
```

Resposta correta!

ID	NAME	AGE	BIRTH_DATE	HAS_DRIVER_LICENSE
2	Elisabeth	30	1996-07-22	false

Enviar Resposta

UC25 – Resolver exercício

Figura 22 - Resolução de Exercício

- Home
- Conta
- Sair

Voltar

Resolução de exercício

Questão 1

Faça uma pesquisa pelo usuário de id 2, trazendo todos os campos da tabela

Diagrama das tabelas disponibilizadas

users

- id: int
- name: varchar
- age: int
- birth_date: date
- has_driver_license: boolean

orders

- id: int
- order_date: date
- total_amount: decimal
- status: varchar
- user_id: int

Sua resposta (SQL):

```
1. SELECT * FROM users WHERE
```

Resposta incorreta!

Enviar Resposta

UC25 – Resolver exercício

5 CONCLUSÃO

Este Trabalho de Conclusão de Curso propôs o desenvolvimento do SQLab, uma plataforma web concebida para transformar o ensino e aprendizado da linguagem SQL e da administração de bancos de dados. Diante dos desafios significativos identificados na educação tradicional de banco de dados, como aulas predominantemente teóricas e a escassez de ferramentas práticas acessíveis, a justificativa para a criação do SQLab fundamentou-se na necessidade urgente de uma solução que concilie teoria e prática, promova a personalização do conteúdo e ofereça escalabilidade técnica. O projeto buscou superar essas barreiras ao proporcionar um ambiente interativo onde alunos podem praticar e visualizar resultados em tempo real, enquanto professores podem gerenciar atividades e acompanhar o progresso de forma otimizada.

Os objetivos estabelecidos para o SQLab foram plenamente alcançados. Desenvolveu-se uma plataforma educacional com interfaces intuitivas para alunos e professores, implementando funcionalidades interativas que permitem a execução de consultas SQL e a manipulação de dados em tempo real. A arquitetura robusta, baseada em Angular 17 para o frontend e Spring Boot com Java 17 para o backend, integrada com PostgreSQL e H2 em memória, provou ser eficaz para garantir estabilidade, escalabilidade e desempenho. Os resultados demonstram um sistema funcional que oferece um painel de controle intuitivo para docentes gerenciarem turmas e atividades, com a inovadora capacidade de cadastrar questões e definir respostas válidas para correção automática. Para os estudantes, o ambiente prático acessível via navegador elimina complexidades de configuração, permitindo foco total no aprendizado. A validação de queries SELECT foca na comparação de resultados, enquanto para INSERT, UPDATE e DELETE, verifica o estado final da tabela após as modificações. Para CREATE e DROP, a validação ocorre pela inspeção da estrutura do banco de dados, garantindo a correção das operações DDL.

Para trabalhos futuros, diversas frentes de aprimoramento foram identificadas, visando expandir as capacidades pedagógicas e interativas do SQLab. Uma área prioritária é o aprofundamento das ferramentas de acompanhamento do professor, com a proposta de incluir a implementação de outros tipos de questões além das de

SQL que alteram dados, como as de resposta textual, múltipla escolha e verdadeiro ou falso, acompanhadas da capacidade de atribuição de notas para essas novas modalidades. Além disso, planeja-se abordar a implementação de questões do tipo ALTER TABLE, expandindo ainda mais a gama de operações DDL que podem ser praticadas. Adicionalmente, pretende-se integrar funcionalidades de feedback direto do professor através de comentários, bem como permitir comentários compartilhados entre alunos e professores nas questões. A inclusão de dicas contextuais para auxiliar na resolução dos exercícios e a implementação de configurações personalizadas de usuário são aprimoramentos que visam otimizar a experiência individual. No âmbito da gestão de conteúdo, a estrutura do banco de dados já prevê a funcionalidade para que professores possam reutilizar exercícios já cadastrados em outras listas e disponibilizar questões publicamente para a comunidade de educadores. O sistema também poderá contar com um acervo de perguntas padronizadas, fornecidas pela própria plataforma para todos os usuários. Por fim, para promover um maior engajamento, considera-se a implementação de funcionalidades de gamificação, como um sistema de ranking de alunos que leve em conta dicas utilizadas e tentativas realizadas, visando estimular a participação e o aprendizado de forma mais dinâmica. Adicionalmente, o sistema poderá incorporar a capacidade de categorizar exercícios por níveis de dificuldade, permitindo aos professores estruturar o aprendizado de forma progressiva. Pensando na flexibilidade e na continuidade do aprendizado, uma implementação futura poderá oferecer a opção de bancos de dados persistentes, permitindo que os alunos mantenham o estado do banco entre sessões, complementando os ambientes temporários em memória e facilitando a construção de conhecimento ao longo de múltiplos dias. Por fim, no que se refere à estrutura de dados e gerenciamento de banco de dados, planejamos adicionar funcionalidades para “imports”, “exports” e “dumps” de dados, oferecendo maior controle e flexibilidade sobre os ambientes de prática.

5.1 CONSIDERAÇÕES FINAIS

O desenvolvimento do SQLab representou uma jornada de aprendizado e superação para a equipe. Durante este trabalho, a aplicação prática dos conceitos de engenharia de software e metodologias ágeis, como UML e Kanban, foi crucial para a organização e o sucesso do projeto. Enfrentamos desafios inerentes ao desenvolvimento de uma plataforma complexa, desde a integração tecnológica entre frontend e backend até a validação precisa de diferentes tipos de comandos SQL. Essas experiências consolidaram nosso conhecimento e nos prepararam para os desafios do mercado de trabalho.

A relevância do SQLab transcende a mera conclusão de um curso. Acreditamos que esta plataforma possui o potencial de impactar positivamente a forma como SQL é ensinado e aprendido, democratizando o acesso à prática de banco de dados e otimizando o processo de ensino para educadores e estudantes na UFPR e em contextos educacionais mais amplos. O ambiente interativo e o feedback imediato oferecidos pelo sistema são pilares para um aprendizado mais engajador e eficaz.

Embora o escopo deste TCC tenha sido definido, a arquitetura modular do SQLab o posiciona para futuras expansões. Visualizamos a plataforma evoluindo para um ecossistema educacional ainda mais completo, incorporando não apenas as funcionalidades já listadas como trabalhos futuros, mas também explorando a integração com inteligência artificial para feedback mais personalizado, ou expandindo para outras áreas da tecnologia. O SQLab, em sua concepção, é um testemunho da capacidade da tecnologia em transformar a educação, e esperamos que sirva como um ponto de partida para inovações contínuas na área.

6 REFERÊNCIAS

Clark, R. E. (1994). **Media will never influence learning**. Educational Technology Research and Development, 42(2), 21-29.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). **Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching**. Educational Psychologist, 41(2), 75-86.

Siemens, G. (2005). **Connectivism: A learning theory for the digital age**. International Journal of Instructional Technology and Distance Learning, 2(1), 3-10.

Boulos, M. N. K., et al. (2020). **Education and training in health informatics and digital health: A global perspective, challenges, and opportunities reviewed**. Yearbook of Medical Informatics, 29(1), 115-123.

Fisher, K., & Baird, D. E. (2011). **Online learning design that fosters student support, self-regulation, and retention**. Campus-Wide Information Systems, 28(5), 329-340.

Hattie, J., & Timperley, H. (2007). **The power of feedback**. Review of Educational Research, 77(1), 81-112.

Reimers, F., & Schleicher, A. (Eds.). (2020). **Teaching for Tomorrow: Learning from the COVID-19 Pandemic**. OECD Publishing.

Jonassen, D. H. (1999). **Constructivist Learning Environments on the Web: Engaging Students in Meaningful Learning**. Educational Technology, 39(3), 35-38.

Hmelo-Silver, C. E. (2004). **Problem-Based Learning: What and How Do Students Learn?**. Educational Psychology Review, 16(3), 235-266.

Prince, M. (2004). **Does Active Learning Work? A Review of the Research**. Journal of Engineering Education, 93(3), 223-231.



Booch, G., Rumbaugh, J., & Jacobson, I. (2005). **The Unified Modeling Language User Guide**. Addison-Wesley.

Fowler, M. (2004). **UML Distilled: A Brief Guide to the Standard Object Modeling Language**. Addison-Wesley.

```

graph TD
    U((Usuário))
    P((Professor))
    A((Aluno))

    U --> AC(Autocadastro)
    U --> L(Login)
    U --> TI(Tela inicial)

    P --> GT(Gerenciar turmas)
    P --> GA(Gerenciar alunos)
    P --> LA(Listar Alunos)
    P --> GL(Gerenciar listas de exercícios)
    P --> EL(Editar lista)
    P --> GE(Gerenciar exercícios)
    P --> RE(Remover exercício)
    P --> AE(Adicionar exercício)
    P --> EE(Editar exercício)
    P --> RA(Remover aluno)
    P --> GSI(Gerenciar solicitação de ingresso de turma)
    P --> VTE(Visualizar tentativas de exercícios)
    P --> VHT(Visualizar histórico de tentativas)
    P --> REE(R Resolver exercícios)
    P --> VE(Visualizar exercícios)
    P --> VL(Visualizar listas)
    P --> VT(Visualizar turmas)
    P --> IT(Inscrever-se em turma)
    P --> CT(Cadastrar turma)
    P --> VRD(Visualizar relatório de desempenho)

    A --> VT
    A --> IT
    A --> VE
    A --> REE
    A --> VHT

    U -.->|<<extend>>| EE
    U -.->|<<extend>>| RE
    U -.->|<<extend>>| AE
    U -.->|<<extend>>| VTE
    U -.->|<<extend>>| VHT
    U -.->|<<extend>>| REE
    U -.->|<<extend>>| VE
    U -.->|<<extend>>| VL
    U -.->|<<extend>>| VT
    U -.->|<<extend>>| IT
    U -.->|<<extend>>| CT
    U -.->|<<extend>>| VRD

    P -.->|<<extend>>| GE
    P -.->|<<extend>>| EL
    P -.->|<<extend>>| GL
    P -.->|<<extend>>| GA
    P -.->|<<extend>>| LA
    P -.->|<<extend>>| VTE
    P -.->|<<extend>>| VHT
    P -.->|<<extend>>| REE
    P -.->|<<extend>>| VE
    P -.->|<<extend>>| VL
    P -.->|<<extend>>| VT
    P -.->|<<extend>>| IT
    P -.->|<<extend>>| CT
    P -.->|<<extend>>| VRD

    A -.->|<<extend>>| VE
    A -.->|<<extend>>| REE
    A -.->|<<extend>>| VHT
  
```

APÊNDICE B – TABELA DE REQUISITOS

Identificador	Requisito	Classificação
RF001	O sistema deve permitir que o usuário se identifique através de login.	Geral
RF002	O sistema deve possuir uma tela inicial de apresentação para usuários deslogados	Geral
RF003	O sistema deve permitir o auto cadastro de professores.	Professor
RF004	O sistema deve permitir que o professor gerencie alunos.	Professor
RF005	O sistema deve permitir que o professor gerencie turmas.	Professor
RF006	O sistema deve permitir que o professor gerencie listas de exercícios.	Professor
RF007	O sistema deve permitir que o professor visualize relatórios de desempenho dos alunos	Professor
RF008	O sistema deve permitir que o professor visualize as tentativas de exercícios dos alunos.	Professor
RF009	O sistema deve permitir que o professor gerencie exercícios.	Professor
RF010	O sistema deve permitir que o professor atribua listas de exercício a turmas.	Professor
RF011	O sistema deve permitir o auto	Aluno

	cadastro dos alunos	
RF012	O sistema deve permitir que o aluno se inscreva em turmas.	Aluno
RF013	O sistema deve permitir que o aluno visualize o histórico de suas tentativas.	Aluno
RF014	O sistema deve permitir que o aluno resolva listas de exercícios.	Aluno

APÊNDICE C – ESPECIFICAÇÕES DE CASO DE USO

UC01 - Tela inicial

Caso de Uso		Visualizar histórico de tentativas
Ator Principal		Geral
Ator secundário		-
Resumo		Tela inicial para usuários não registrados
Pré-condições		-
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Entra no sistema	-
2	-	Exibe a tela com apresentação geral do sistema
3	Caso de Uso encerra	
Fluxo Alternativo		
A (2.1) Visualizar resposta		
Passo	Ação do Ator	Ação do Sistema
1	Seleciona botão para saber mais	-
2	-	Redireciona o usuário para parte da tela com as opções de auto cadastro e login
3	Caso de uso encerra	

UC02 - Auto cadastro do Professor

Caso de Uso		Auto cadastro
Ator Principal		Professor
Ator secundário		-
Resumo		Permite que professor faça o auto cadastro
Pré-condições		-
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa a página de autocadastro	-
2	-	Exibe a tela com o formulário: Campos: <ul style="list-style-type: none">- E-mail;- Senha;- Nome Completo; Botões: <ul style="list-style-type: none">- Confirmar;
3	Preenche os campos exigidos e clica em “registrar”	-
4	-	Faz a validação das informações fornecidas e armazena no banco de dados
5	-	Salva informações na sessão e redireciona o professor para a página principal.
6	Caso de Uso encerra	



Fluxo de Exceção		
E (2.1) dados obrigatórios não enviados		
Passo	Ação do Ator	Ação do Sistema
1	-	Informa que os dados obrigatórios não foram enviados
2	Caso de uso retorna para o passo 3 do fluxo principal	

UC03 - Login

Caso de Uso		Login
Ator Principal		Usuário
Ator secundário		-
Resumo		Permite que um usuário faça a autenticação no sistema
Pré-condições		O usuário precisa ter um registro de cadastro
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa a página de login	-
2	-	Exibe a tela com o formulário: Campos: <ul style="list-style-type: none">- E-mail- Senha Botões:

		<ul style="list-style-type: none">- Lembrar-me- Entrar- Cadastrar-se
3	Preenche os campos exigidos e clica no botão “login”	
4	-	Faz a validação das informações fornecidas
5	-	Salva informações na sessão e redireciona o usuário para a página principal
6	Caso de Uso encerra	
Fluxo de Exceção		
E (2.1) email ou senha incorretos		
Passo	Ação do Ator	Ação do Sistema
1	-	Informa que os dados estão errados
2	Caso de uso retorna para o passo 3 do fluxo principal	
E (2.2) dados obrigatórios não enviados		
Passo	Ação do Ator	Ação do Sistema
1	-	Informa que os dados obrigatórios não foram enviados
2	Caso de uso retorna para o passo 3 do fluxo principal	

Fluxo Alternativo		
A (2.3) Clicar em “Cadastrar-se”		
Passo	Ação do Ator	Ação do Sistema
1	Clica no botão “Cadastrar-se”	-
2	-	Sistema redireciona o usuário para a tela de cadastro de professor
3	Caso de uso encerra	

UC04 – Cadastrar turma

Caso de Uso		Cadastrar turma
Ator Principal		Professor
Ator secundário		-
Resumo		Permite que os professores realizem o cadastro das turmas
Pré-condições		Usuário precisa estar autenticado com usuário de professor
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Seleciona o botão de Adicionar turma	-
2	-	Redireciona para a tela de



		cadastro de turma
3	-	Exibe a tela com os campos: <ul style="list-style-type: none">- Nome da turma- Descrição
4	Preenche os campos com os dados respectivos	-
5	Clica em “Salvar Turma”	-
	-	Salva os dados enviados, gera um código para a turma
5	Caso de Uso encerra	

UC05 - Listar turmas

Caso de Uso		Listar turmas
Ator Principal		Professor
Ator secundário		-
Resumo		Permite que o professor visualize as turmas registradas
Pré-condições		Professor deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de “turmas”	



2		Exibe uma listagem com as turmas registradas <ul style="list-style-type: none">- Nome da turma- Código da turma Botões: <ul style="list-style-type: none">- Listas- Criar Lista- Estudantes- Editar- Remover
3	Caso de Uso encerra	
Fluxo Alternativo		
A (2.1) Clica no botão “cadastrar turma”		
Passo	Ação do Ator	Ação do Sistema
1	Clica no botão “Adicionar turma”	
2		Abre tela de registro de nova turma
3	Caso de uso encerra	

UC06 - Editar turmas

Caso de Uso		Editar turmas
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que o professor edite turmas	
Pré-condições	Professor deve estar autenticado	
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de detalhe de uma turma	
2		Carrega na tela um formulário com informações do banco de dados <ul style="list-style-type: none">- Nome- Descrição
3	Clica no botão “salvar”	
4		Valida informações e salva informações no banco de dados
5	Caso de Uso encerra	
Fluxo de Exceção		
E (2.1) campos obrigatórios não enviados		
Passo	Ação do Ator	Ação do Sistema



1		Exibe mensagem pedindo as informações obrigatórias
2	Caso de uso retorna para passo 2 do fluxo principal	

UC07 - Remover turmas

Caso de Uso		Remover turmas
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que professor remove turmas cadastradas	
Pré-condições		
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Clica no botão “remover” em alguma turma na listagem de turmas	
2		Exibe modal de confirmação de exclusão, dizendo que todas as informações linkadas a turma serão perdidas Botões: confirmar; cancelar;
3	Clica no botão “confirmar”	
4		Informação é excluída da



		base de dados
5	Caso de Uso encerra	
Fluxo Alternativo		
A (2.1) Clica no botão de fechar o modal		
Passo	Ação do Ator	Ação do Sistema
1	Clica no botão para fechar o modal de confirmação	
2		Modal é fechado
3	Caso de uso encerra	



UC08 - Listar exercícios

Caso de Uso		Listar exercícios
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que o professor visualize uma lista de exercícios	
Pré-condições	Professor deve estar autenticado	
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de “exercícios”	
2		Carrega uma tela com botão para cadastrar novo exercício e uma lista de contendo: <ul style="list-style-type: none">- Nome do exercício- Tipo do exercício- Botão para editar exercício- Botão para remover exercício
3	Caso de Uso encerra	
Fluxo Alternativo		



A (2.1) Clica no botão “adicionar”		
Passo	Ação do Ator	Ação do Sistema
1		Abre tela de novo exercício
2	Caso de uso encerra	



UC09 - Cadastrar exercícios

Caso de Uso		Cadastrar exercícios
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que o professor cadastre novos exercícios	
Pré-condições	Professor deve estar autenticado	
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de cadastro exercício	
2		Carrega na tela um formulário Campos: <ul style="list-style-type: none">- Título do Exercício- Dialeto do banco- Tipo de comando SQL- Descrição- Exercício será público- Resposta correta
3	Clica no botão “cadastrar”	
4		Valida informações e salva registro no banco de dados



5	Caso de Uso encerra	
Fluxo de Exceção		
E (2.1) informações obrigatórias não enviadas		
Passo	Ação do Ator	Ação do Sistema
1		Exibe na tela a informação de que os campos obrigatórios não foram enviados
2	Caso de uso retorna para o passo 2 do fluxo principal	



UC10 - Editar exercícios

Caso de Uso		Editar exercícios
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que o professor edite um exercício	
Pré-condições	Professor deve estar autenticado	
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de edição de exercício	
2		Carrega na tela um formulário com informações do banco de dados Campos: <ul style="list-style-type: none">- Título do Exercício- Dialeto do banco- Tipo de comando SQL- Descrição- Exercício será público- Resposta correta
3	Edita os campos desejados e clica no botão “salvar”	



4		Valida informações e salva registro no banco de dados
5	Caso de Uso encerra	
Fluxo de Exceção		
E (2.1) informações obrigatórias não enviadas		
Passo	Ação do Ator	Ação do Sistema
1		Exibe na tela a informação de que os campos obrigatórios não foram enviados
2	Caso de uso retorna para o passo 2 do fluxo principal	

UC11 - Remover exercícios

Caso de Uso		Remover exercícios
Ator Principal		Professor
Ator secundário		-
Resumo		Permite que o professor remova exercícios
Pré-condições		Professor deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessar listagem de exercícios	
2	Clicar no botão “remover” de algum registro	

3		Valida se exercício não pertence a nenhuma lista e exibe modal de confirmação da exclusão
4	Confirma a exclusão	
5		Registro é excluído da base de dados
6	Caso de Uso encerra	
Fluxo de Exceção		
E (2.1) O exercício excluído pertence a uma lista		
Passo	Ação do Ator	Ação do Sistema
1		Exibe a informação de que o exercício pertence a uma lista e não pode ser excluído enquanto ela existir
2	Caso de uso volta para passo 1 do fluxo principal	
Fluxo Alternativo		
A (2.1) Clica no botão de fechar o modal		
Passo	Ação do Ator	Ação do Sistema
1	Clica no botão para fechar o modal de confirmação	
2		Modal é fechado



3	Caso de uso encerra
---	---------------------

UC12 - Listar listas de exercícios

Caso de Uso		Listar listas de exercícios
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que o professor cadastre listas de exercícios	
Pré-condições	Professor deve estar autenticado	
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de “listas de exercícios”	
2		Exibe uma listagem das listas cadastradas Campos: <ul style="list-style-type: none">- Nome Botões: <ul style="list-style-type: none">- Criar exercício- Exercícios- Editar
4	Caso de Uso encerra	
Fluxo Alternativo		
A (2.1) Clica no botão “cadastrar lista”		
Passo	Ação do Ator	Ação do Sistema
1	Clica no botão “Adicionar lista”	



2		Abre tela de registro de nova lista
3	Caso de uso encerra	

UC13 - Cadastrar listas de exercícios

Caso de Uso		Cadastrar listas de exercícios
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que o professor cadastre uma lista de exercício	
Pré-condições	Professor deve estar autenticado e pelo menos um exercício deve estar registrado	
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de cadastro de nova lista de exercícios	
2		Exibe um formulário com Campos: <ul style="list-style-type: none">- Nome da lista- Descrição Botões: <ul style="list-style-type: none">- Cadastrar;
3	Clica no botão “cadastrar”	
4		Valida informações e



		salva informações no banco de dados
5	Caso de Uso encerra	
Fluxo de Exceção		
E (2.1) campos obrigatórios não enviados		
Passo	Ação do Ator	Ação do Sistema
1		Exibe mensagem pedindo as informações obrigatórias
2	Caso de uso retorna para passo 2 do fluxo principal	



UC14 - Editar listas de exercícios

Caso de Uso		Editar listas de exercícios
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que o professor edite uma lista de exercício	
Pré-condições	Professor deve estar autenticado	
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de detalhe de uma lista	
2		Carrega na tela um formulário com informações do banco de dados Campos: exercícios; Botões: salvar;
3	Clica no botão “salvar”	
4		Valida informações e salva informações no banco de dados
5	Caso de Uso encerra	
Fluxo de Exceção		



E (2.1) campos obrigatórios não enviados

Passo	Ação do Ator	Ação do Sistema
1		Exibe mensagem pedindo as informações obrigatórias
2	Caso de uso retorna para passo 2 do fluxo principal	

UC15 - Listar alunos

Caso de Uso		Listar alunos
Ator Principal		Professor
Ator secundário		-
Resumo		Permite que professor visualize os alunos registrados e aceite ou recuse seu ingresso na turma
Pré-condições		Professor deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa tela de alunos	
2		Exibe uma listagem de alunos que desejam entrar na turma e alunos matriculados <ul style="list-style-type: none">- Nome- Email



		Botões: - Visualizar exercícios - Remover - Aceitar - Recusar
3	Caso de Uso encerra	
Fluxo Alternativo		
A (2.1) Professor clica no botão Visualizar Exercícios		
Passo	Ação do Ator	Ação do Sistema
1	Clica no botão “Visualizar Exercícios”	
2		Abre tela de histórico de resolução de exercícios desse aluno
3	Caso de uso encerra	

UC16 - Remover alunos

Caso de Uso		Remover alunos
Ator Principal		Professor
Ator secundário		-
Resumo		Permite que o professor delete alunos
Pré-condições		Professor deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Clica no botão “remover” em algum aluno na listagem de alunos	
2		Informação é excluída da base de dados
3	Caso de Uso encerra	

UC17 - Visualizar tentativas de exercícios

Caso de Uso		Visualizar tentativas de exercícios
Ator Principal	Professor	
Ator secundário	-	
Resumo	Permite que o professor visualize as tentativas de resolução dos exercícios	
Pré-condições	Professor deve estar autenticado	

Fluxo Principal

Passo	Ação do Ator	Ação do Sistema
1	Acessa página de tentativas de resolução de exercícios	
2		Exibe uma listagem dos exercícios com tentativas Campos: <ul style="list-style-type: none">- Dados do aluno- Nome- Email- Turma- Nome de cada exercício- Tipo de exercício- Enunciado- Resposta do aluno
3	Caso de Uso encerra	

UC18 - Visualizar relatório de desempenho

Caso de Uso		Visualizar relatório de desempenho
Ator Principal		Professor
Ator secundário		-
Resumo		Permite que o professor acesse uma tela com informações mais gerais sobre o desempenho dos alunos
Pré-condições		Professor deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página principal	
2		Exibe informações sobre listas de exercícios Campos: <ul style="list-style-type: none"> - Posição - Aluno - Tentativas - Total Exercícios - Pontuação o Filtros: <ul style="list-style-type: none"> - Turma - Lista
3	Caso de Uso encerra	

UC19 - Auto cadastro do aluno

Caso de Uso		Auto cadastro
Ator Principal		Aluno
Ator secundário		-
Resumo		Permite que o aluno faça o auto cadastro
Pré-condições		-
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa a página de auto cadastro	-
2	-	Exibe a tela com o formulário: Campos: <ul style="list-style-type: none"> - E-mail - Confirmar e-mail - Matrícula - Senha - Confirmar senha - Nome Completo Botões: <ul style="list-style-type: none"> - Confirmar
3	Preenche os campos exigidos e clica em "registrar"	-
4	-	Faz a validação das informações fornecidas e armazena no banco de dados
5	-	Salva informações na sessão e redireciona o aluno para a página principal.
6	Caso de Uso encerra	



Fluxo de Exceção		
E (2.1) dados obrigatórios não enviados		
Passo	Ação do Ator	Ação do Sistema
1	-	Informa que os dados obrigatórios não foram enviados
2	Caso de uso retorna para o passo 3 do fluxo principal	

UC20 – Inscrever-se em turma

Caso de Uso		Inscrever em turma
Ator Principal		Aluno
Ator secundário		-
Resumo		Permite que o aluno se inscreva em uma turma
Pré-condições		O professor deve informar aos alunos o código referente a turma na qual eles devem se cadastrar.
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa ao sistema	
2		Exibe um formulário na página inicial com um input para inserir o código informado pelo professor



3	Informa o código passado pelo professor	
4		Salva informações na sessão e envia requisição de ingresso para o professor
5	Caso de Uso encerra	

UC21 – Visualiza turmas

Caso de Uso		Visualizar histórico de tentativas
Ator Principal		Aluno
Ator secundário		-
Resumo		Permite que o aluno visualize em quais turmas está cadastrado
Pré-condições		Aluno deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa sistema	
2		Carrega uma lista com todas as turmas em que o aluno está cadastrado
3	Caso de Uso encerra	

UC22 – Visualiza listas de exercícios

Caso de Uso		Visualizar histórico de tentativas
Ator Principal		Aluno
Ator secundário		-
Resumo		Permite que o aluno visualize as listas cadastradas para a turma em questão
Pré-condições		Aluno deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa tela de listar listas da turma	
2		Carrega uma lista com todas as listas de exercícios da turma em questão, com um botão para acesso aos exercícios.
3	Caso de Uso encerra	

UC23 – Visualiza exercícios de uma lista

Caso de Uso		Visualizar histórico de tentativas
Ator Principal		Aluno
Ator secundário		-
Resumo		Permite que o aluno visualize os exercícios cadastrados para a lista em questão



Pré-condições		Aluno deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa tela de listar listas da turma	
2		Carrega uma lista com todos os exercícios cadastrados para a lista em questão, apresentando o nome do exercício e seu dialeto, com um botão para acesso ao histórico de resolução e um botão para realizar o exercício.
3	Caso de Uso encerra	

UC24 - Visualizar histórico do exercício

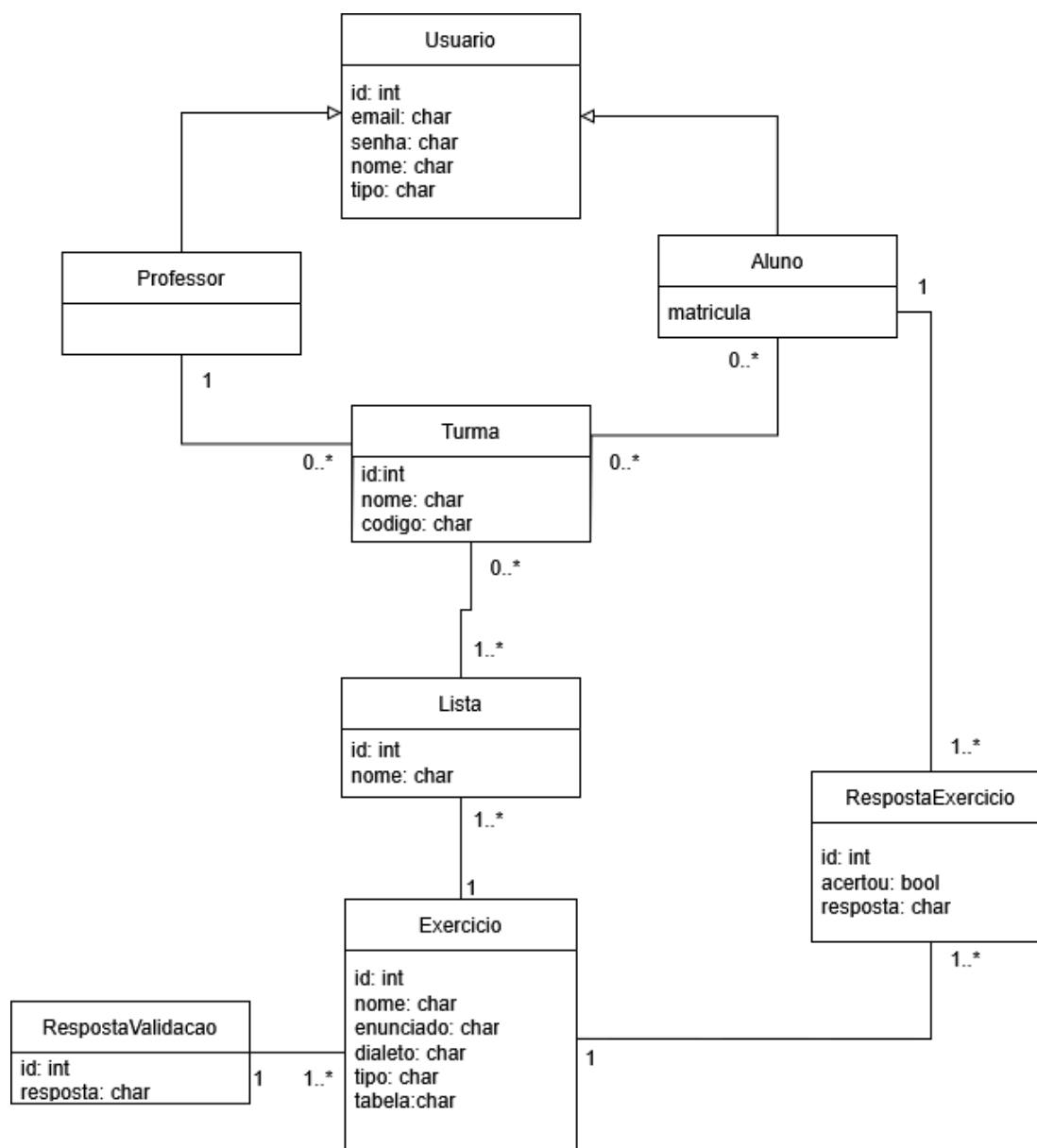
Caso de Uso		Visualizar tentativas de exercícios
Ator Principal		Aluno
Ator secundário		-
Resumo		Permite que o aluno visualize as tentativas de resolução do exercício em questão
Pré-condições		Aluno deve estar autenticado
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa página de histórico de tentativas de resolução de exercício	
2		Exibe a turma, lista, nome do exercício, o dialeto e seu enunciado, abaixo, uma listagem dos exercícios com tentativas de respostas Campos: <ul style="list-style-type: none"> - Resposta - Resultado
3	Caso de Uso encerra	

UC25 - Resolver exercício

Caso de Uso		Resolver exercícios
Ator Principal		Aluno
Ator secundário		-
Resumo		Permite que o aluno resolva exercícios de uma lista
Pré-condições		Aluno deve estar autenticado e com pelo menos uma lista de exercícios pendente de resolução
Fluxo Principal		
Passo	Ação do Ator	Ação do Sistema
1	Acessa a página de um exercício de uma lista	
2		Carrega na página: Campos: <ul style="list-style-type: none"> - Nome - Enunciado - Tabela de relação do banco a ser utilizado - Campo para inserir o SQL Botão para Enviar Resposta
	Preenche o campo "Sua resposta (SQL)" e clica em "Enviar Resposta"	

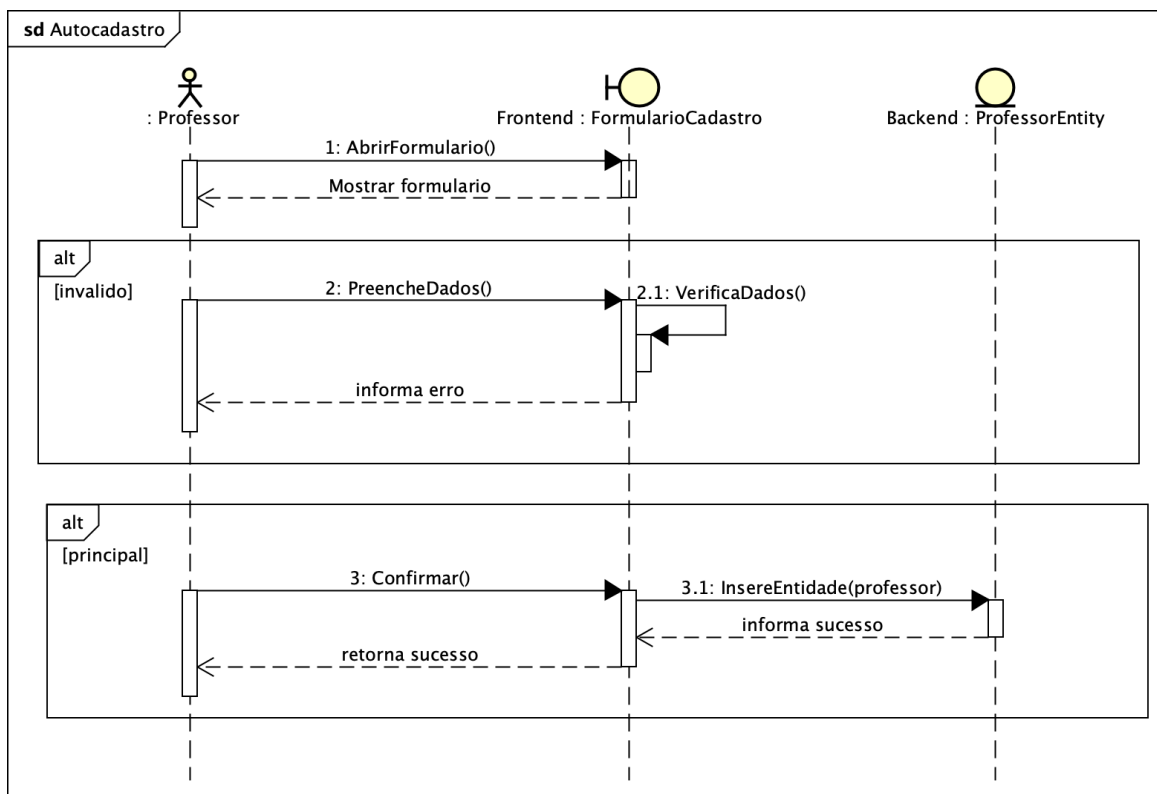
4		Salva informações na base de dados
5		Processa resposta conforme o tipo de exercício
6		Retorna ao estudante o resultado e informações derivadas deste.
7	Caso de Uso encerra	
Fluxo de Exceção		
E (2.1) o campo de resposta não foi preenchido		
Passo	Ação do Ator	Ação do Sistema
1		Exibe mensagem que o campo “resposta” deve ser preenchido
2	Caso de uso volta para o passo 3 do fluxo principal	

APÊNDICE D – DIAGRAMA DE CLASSE DE ANÁLISE

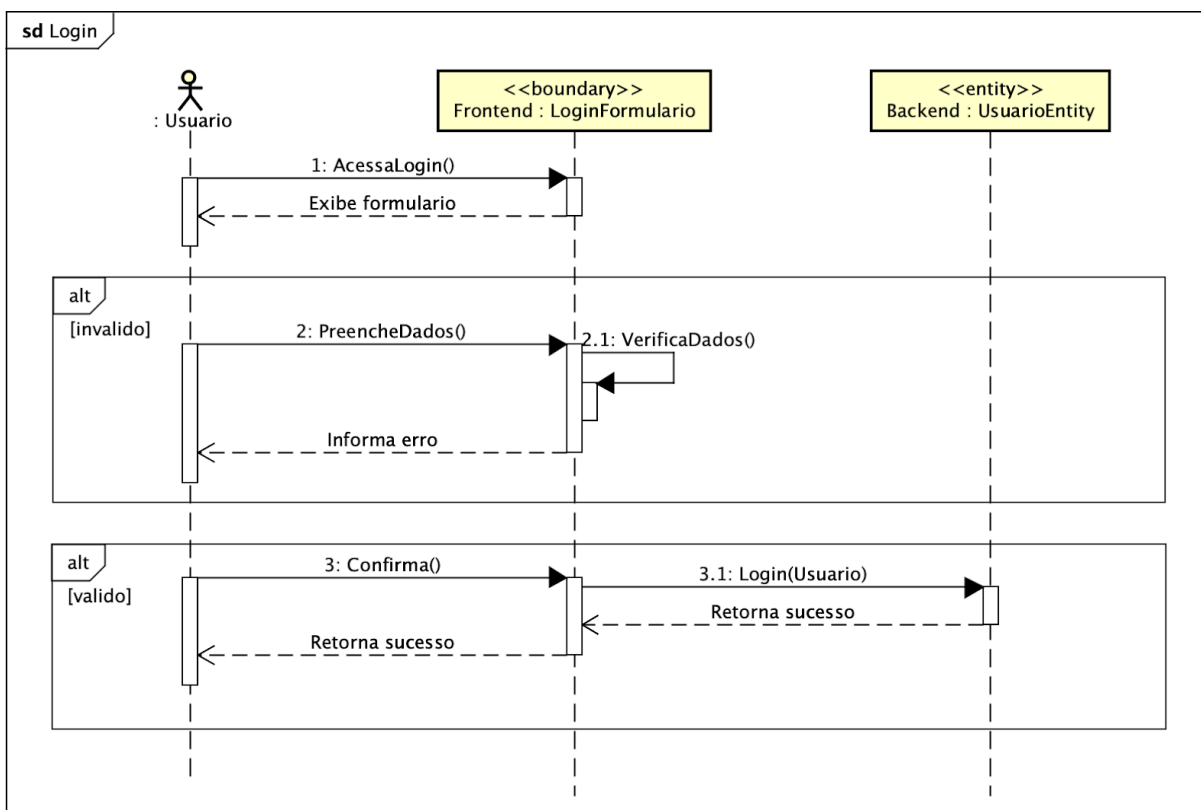


APÊNDICE E – DIAGRAMA DE SEQUÊNCIA DE ANÁLISE

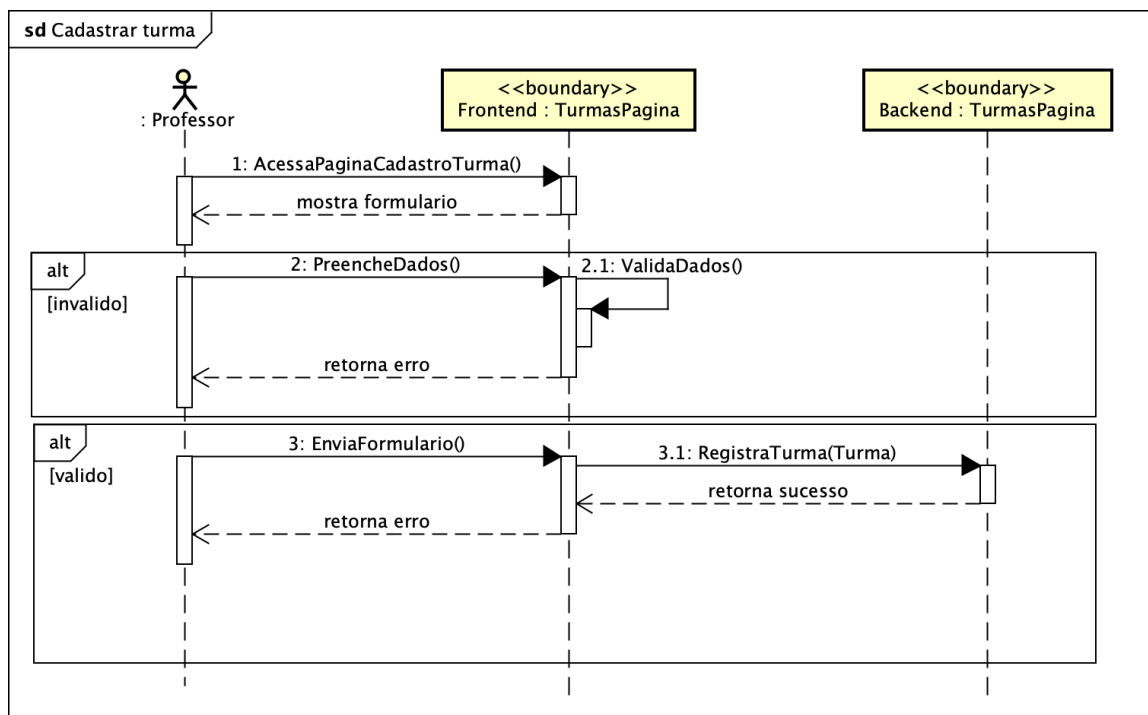
UC02 – Auto cadastro



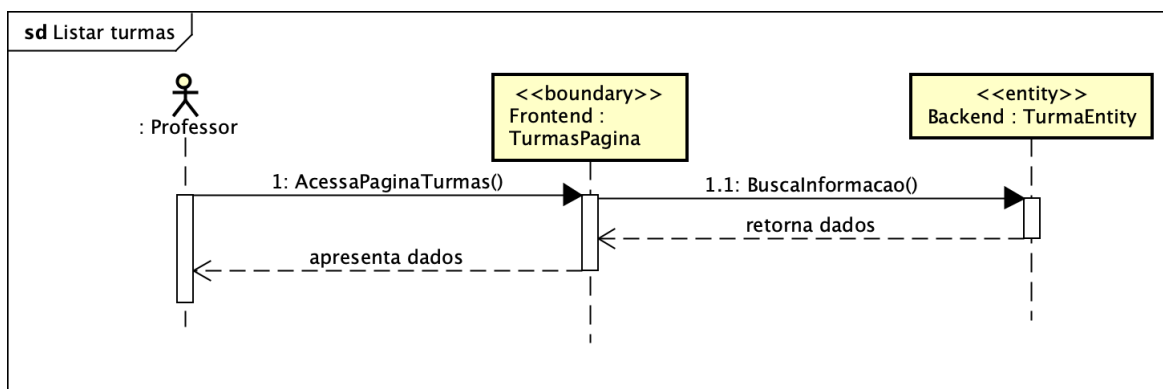
UC03 - Login



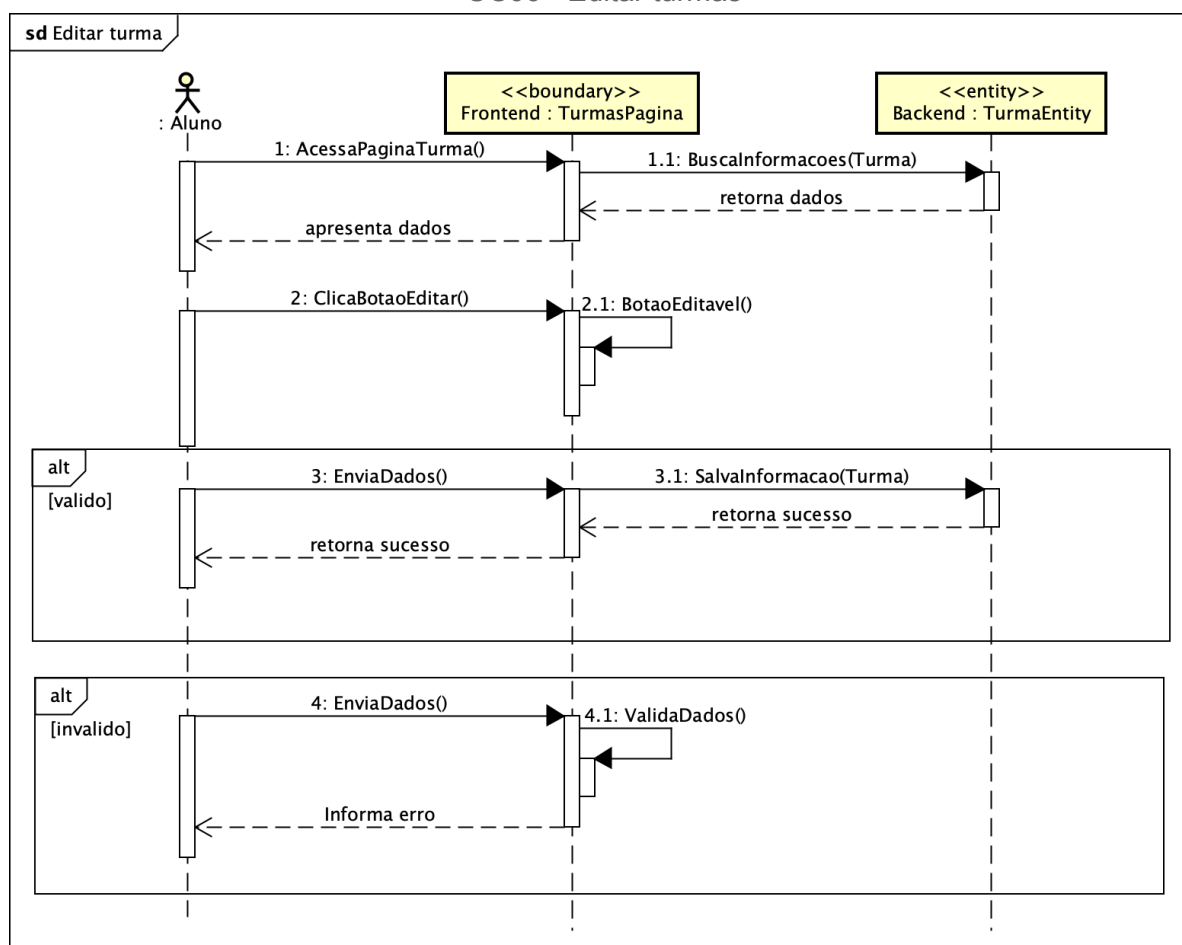
UC04 - Cadastrar turmas



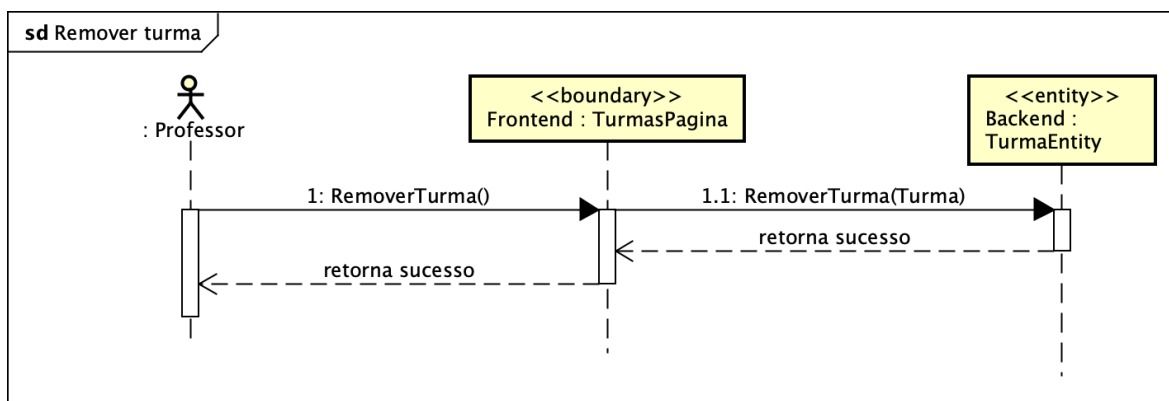
UC05 - Listar turmas



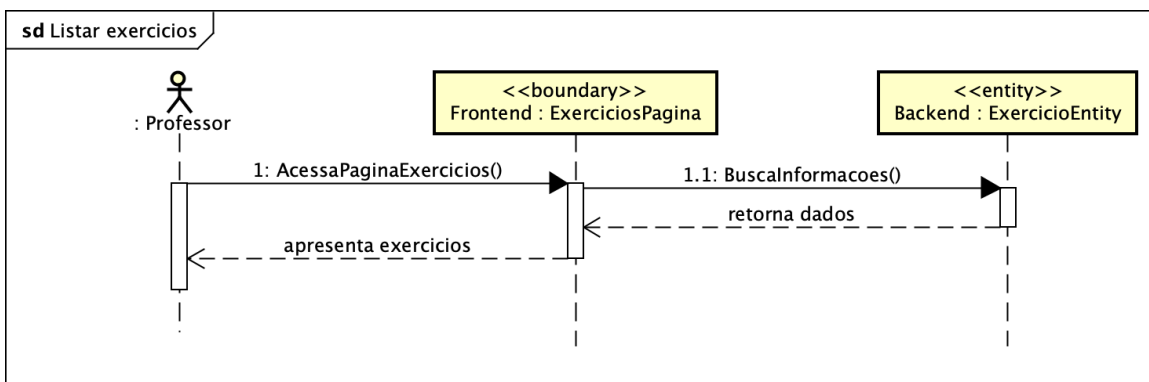
UC06 - Editar turmas



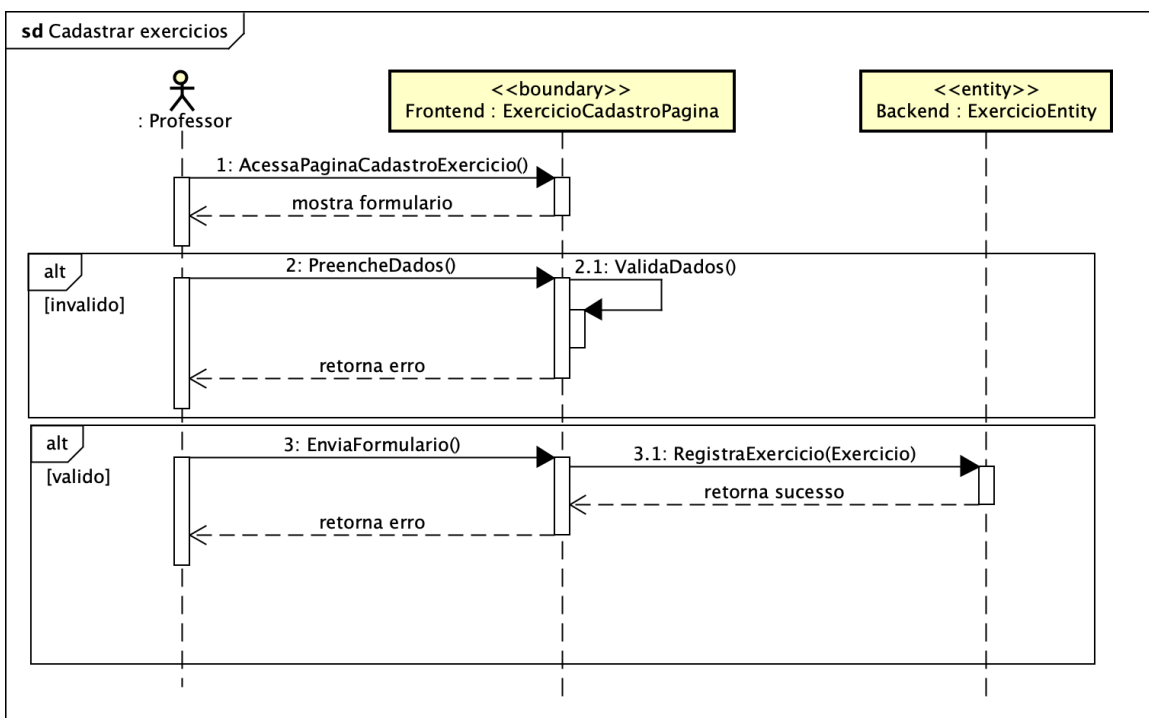
UC07 - Remover turmas



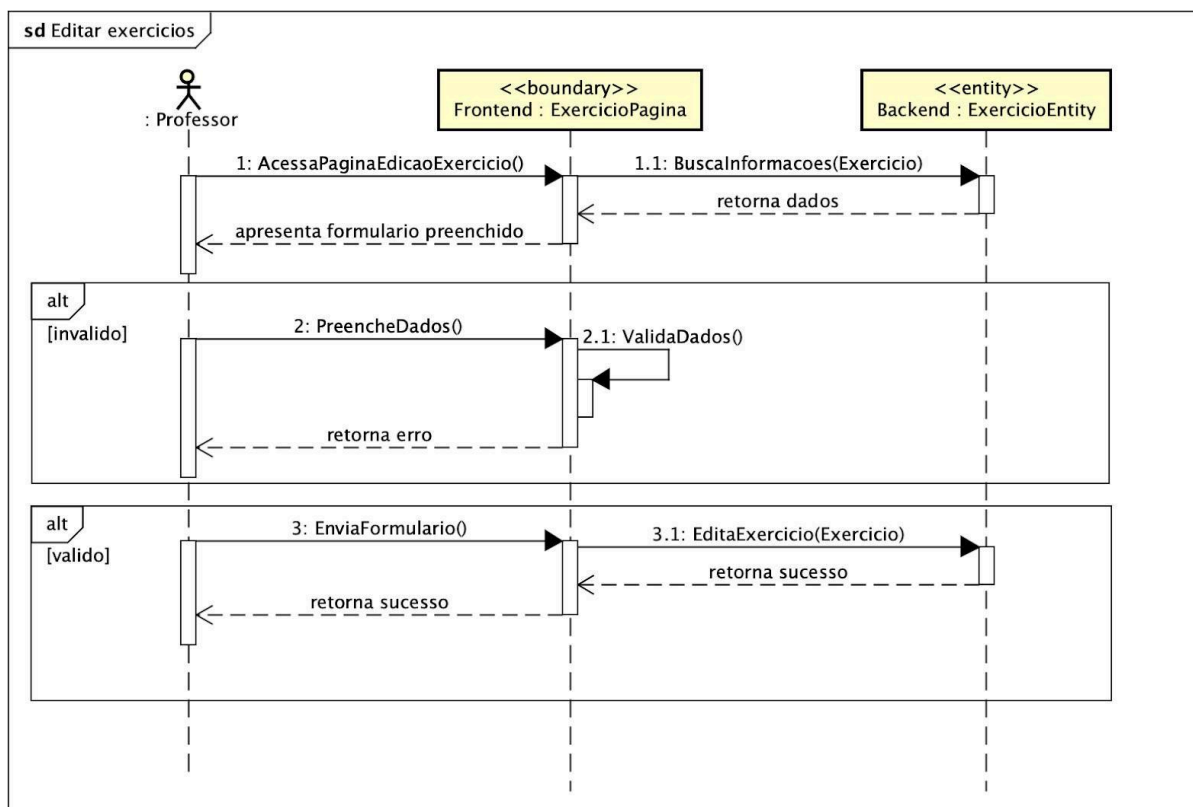
UC08 - Listar exercícios



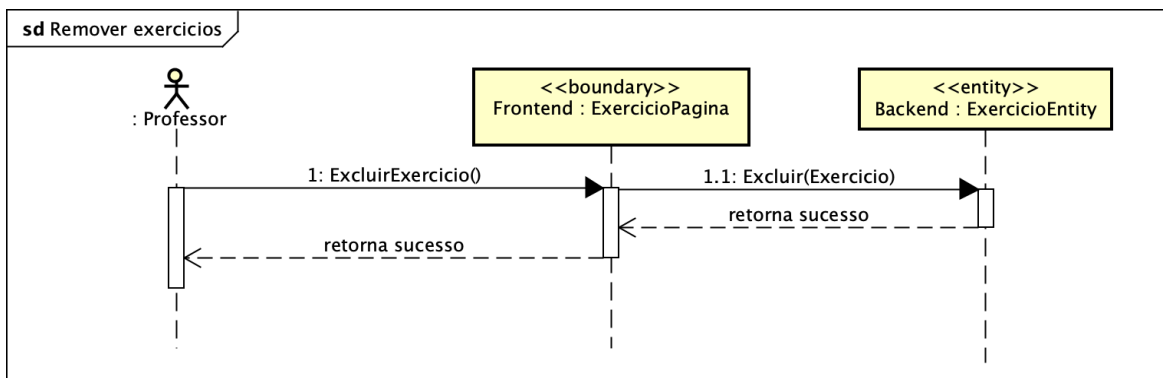
UC09 - Cadastrar exercícios



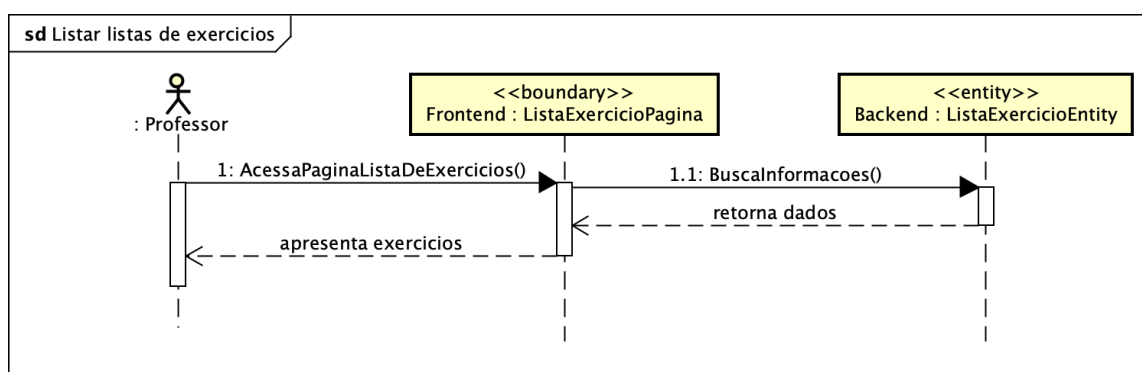
UC10 - Editar exercícios



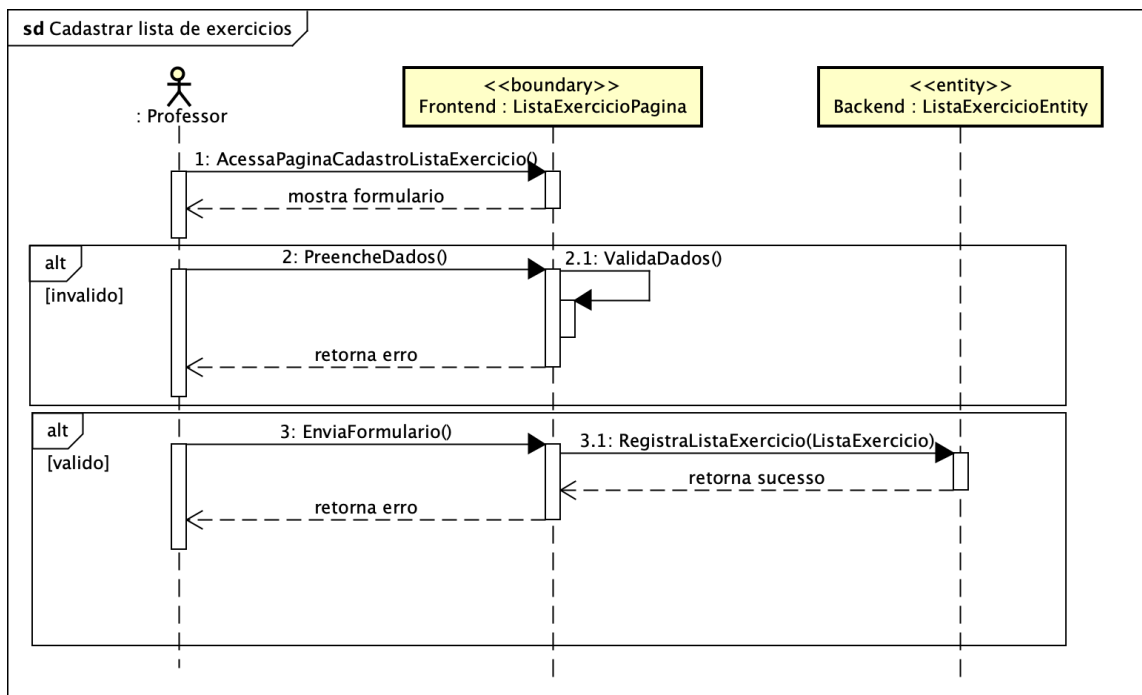
UC11 - Remover exercícios



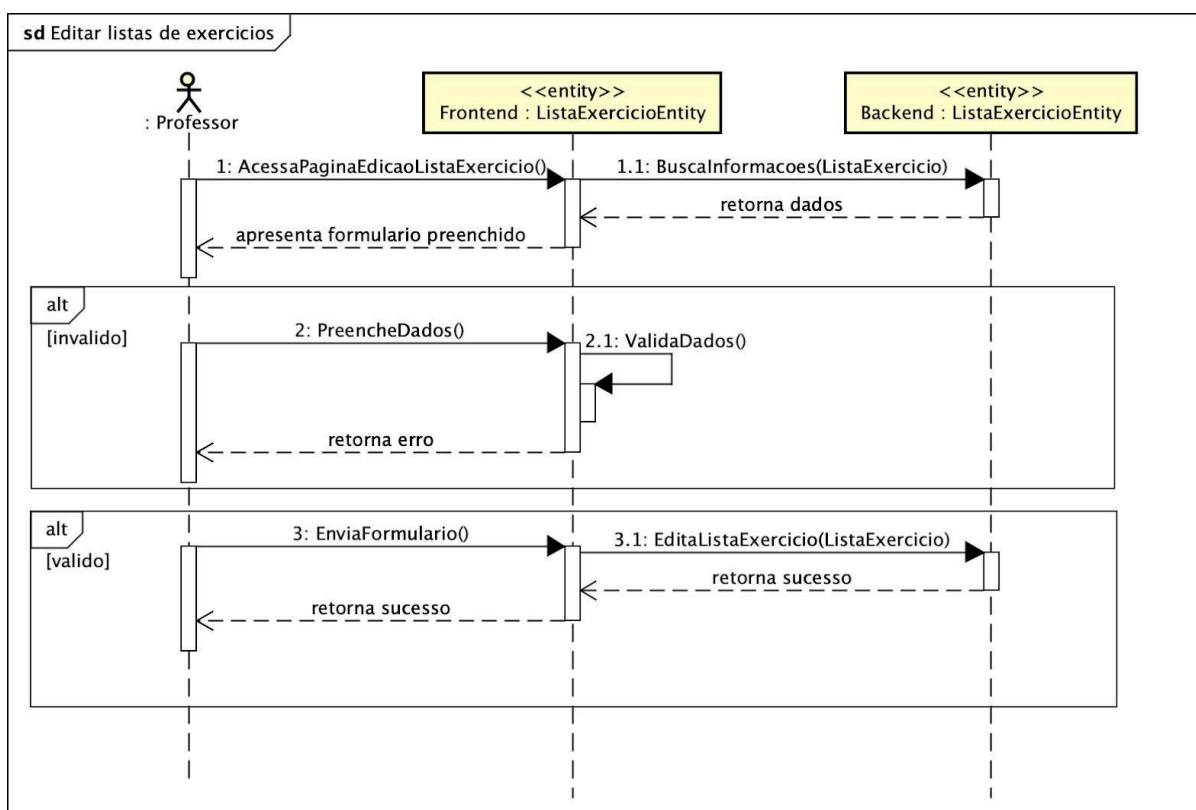
UC12 - Listar listas de exercícios



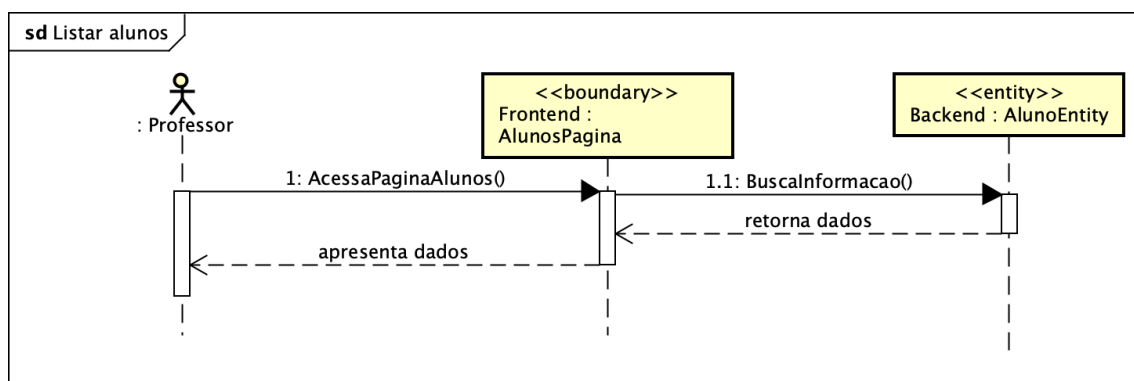
UC13 - Cadastrar listas de exercícios



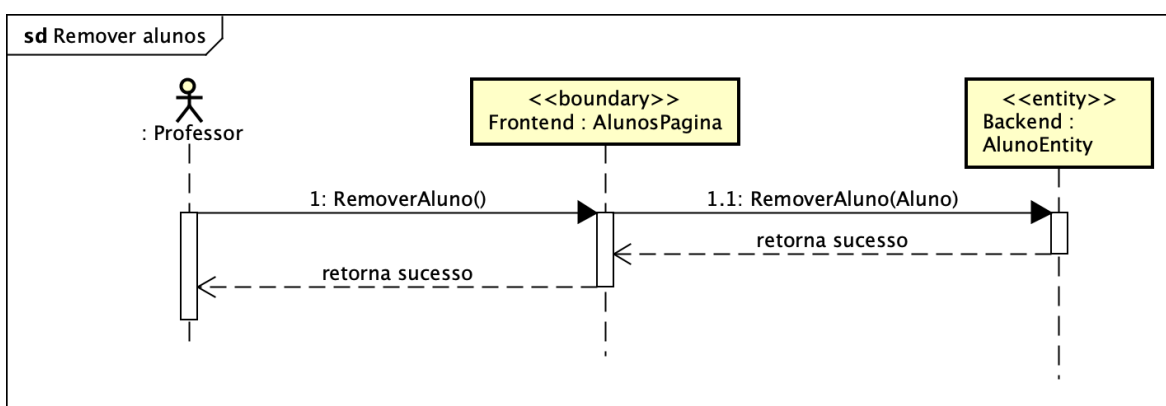
UC14 - Editar listas de exercícios



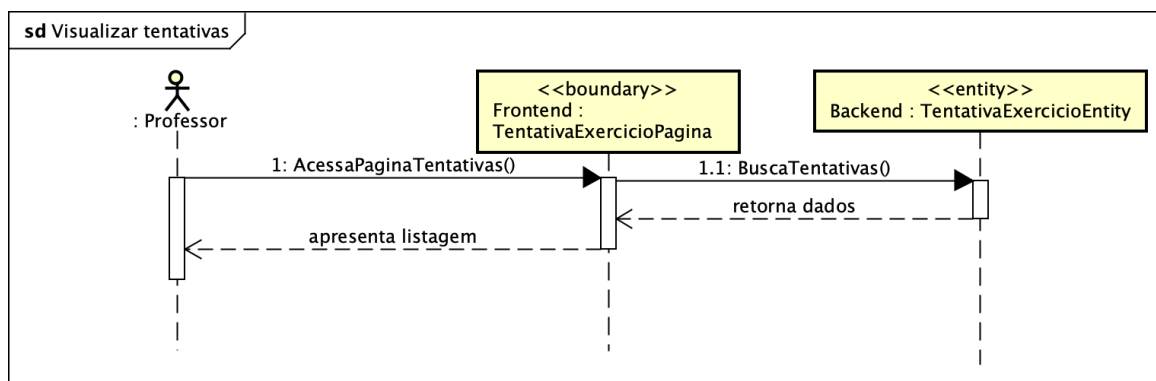
UC15 - Listar alunos



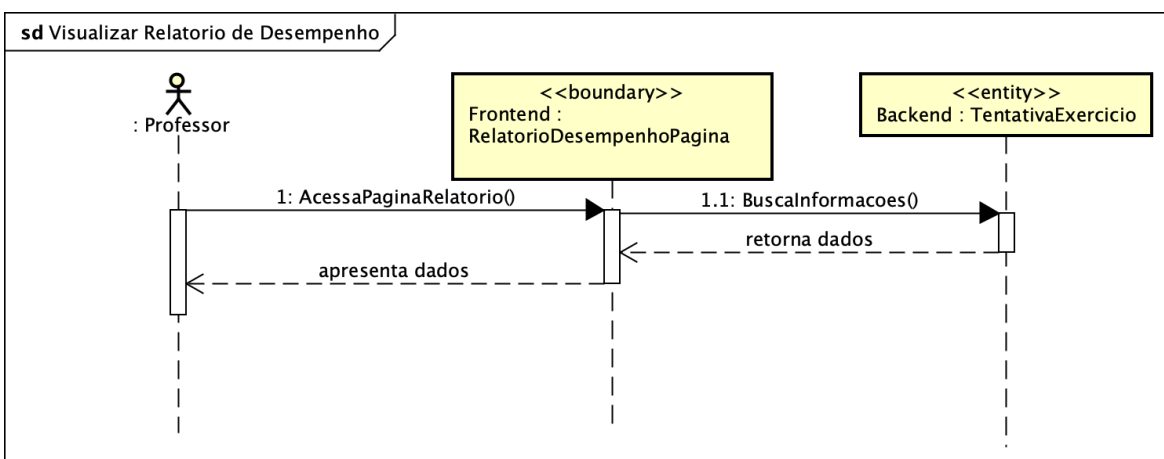
UC16 - Remover alunos



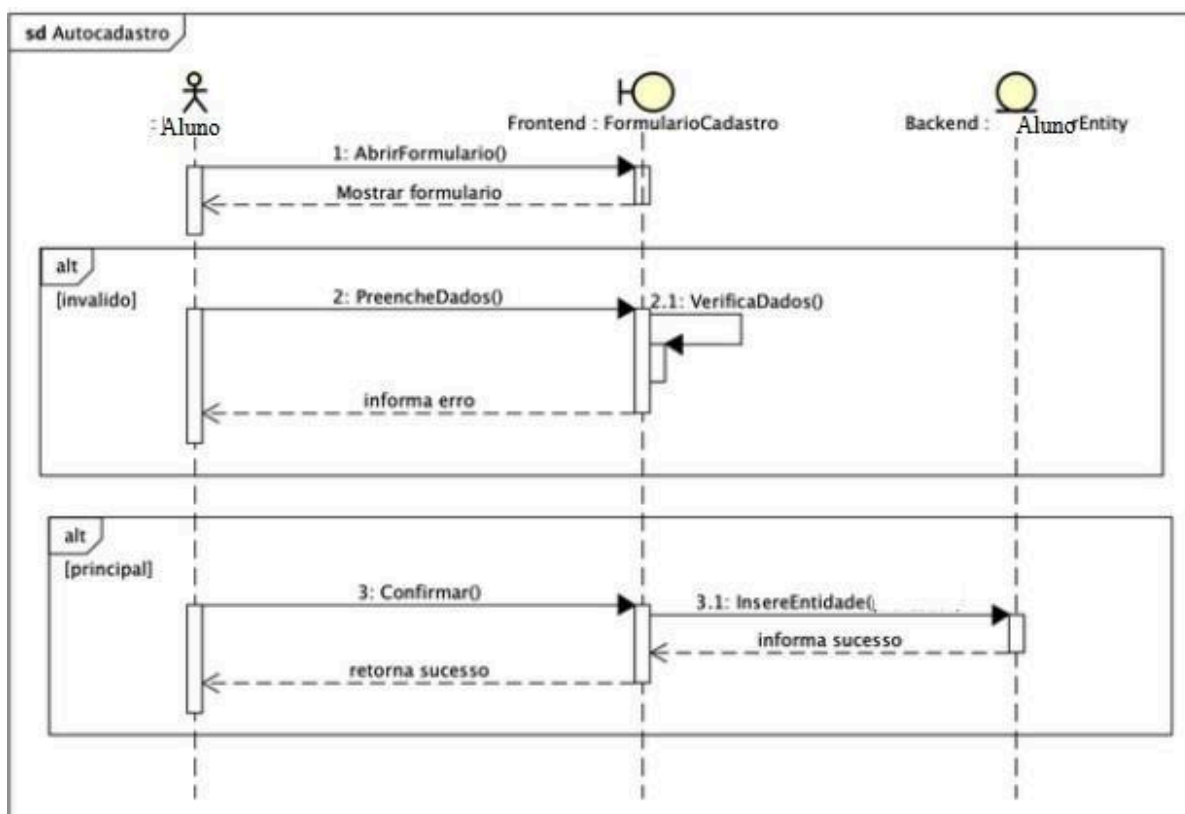
UC17 - Visualizar tentativas de exercícios



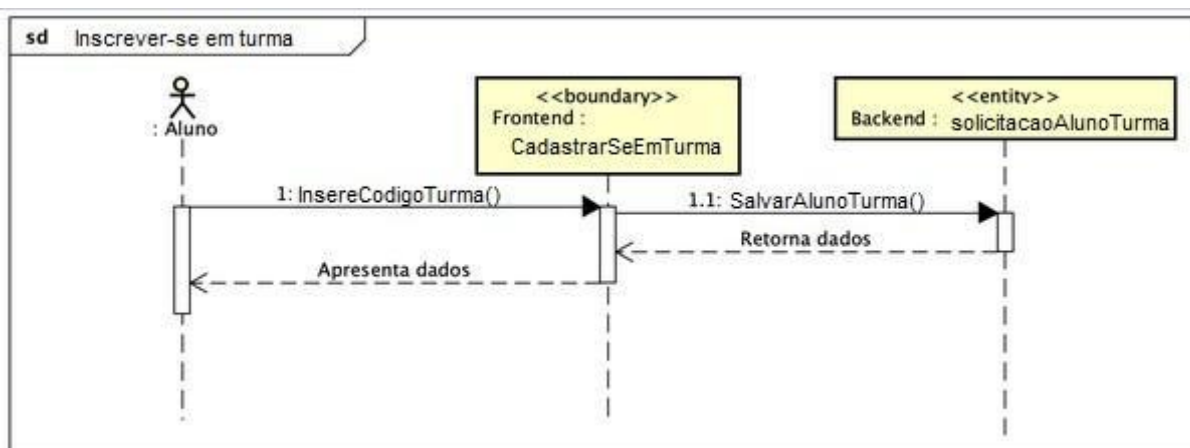
UC18 - Visualizar relatório de desempenho



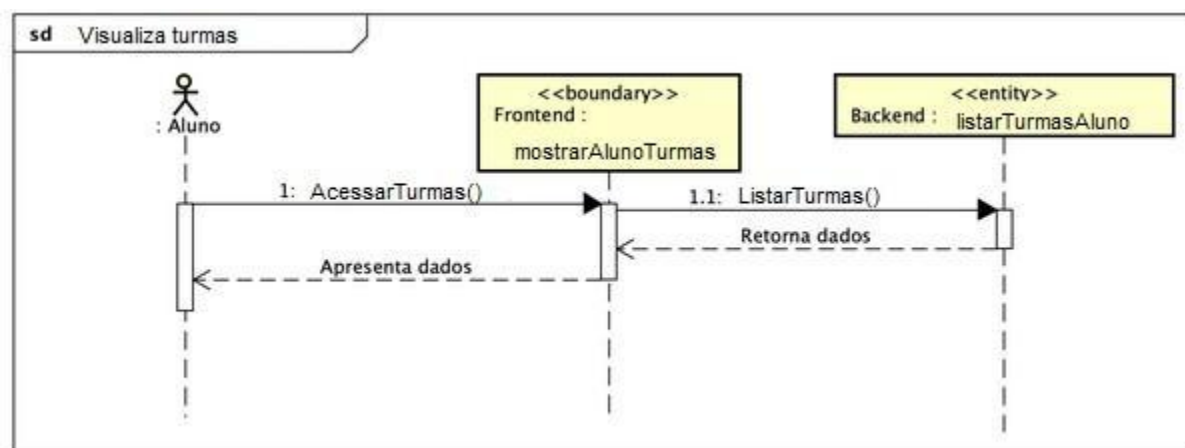
UC19 - Auto cadastro do aluno



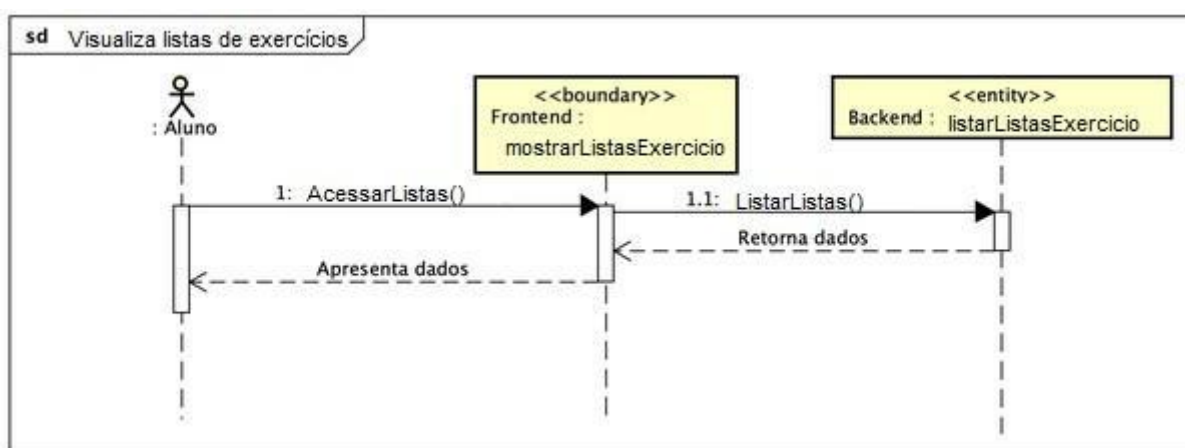
UC20 - Inscrever-se em turma



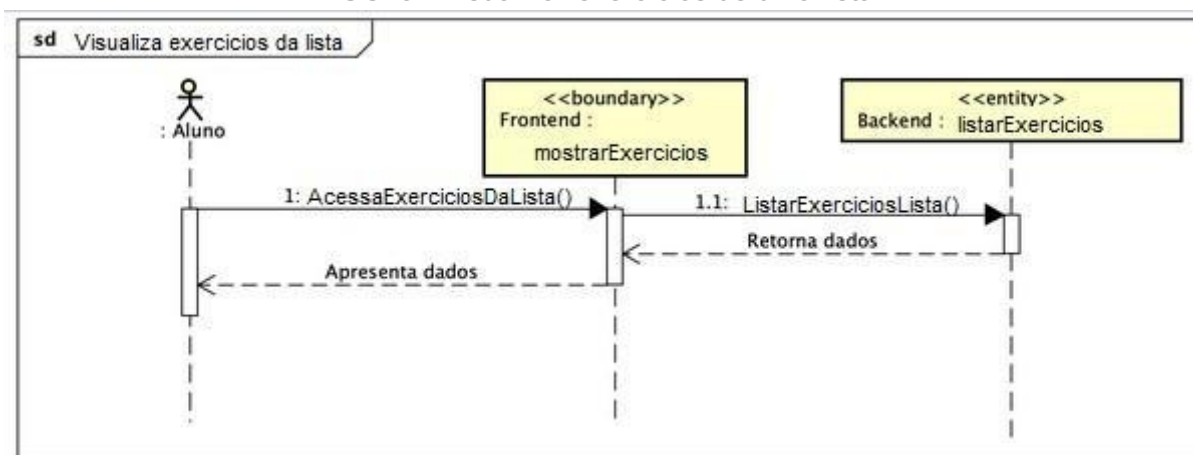
UC21 – Visualiza turmas



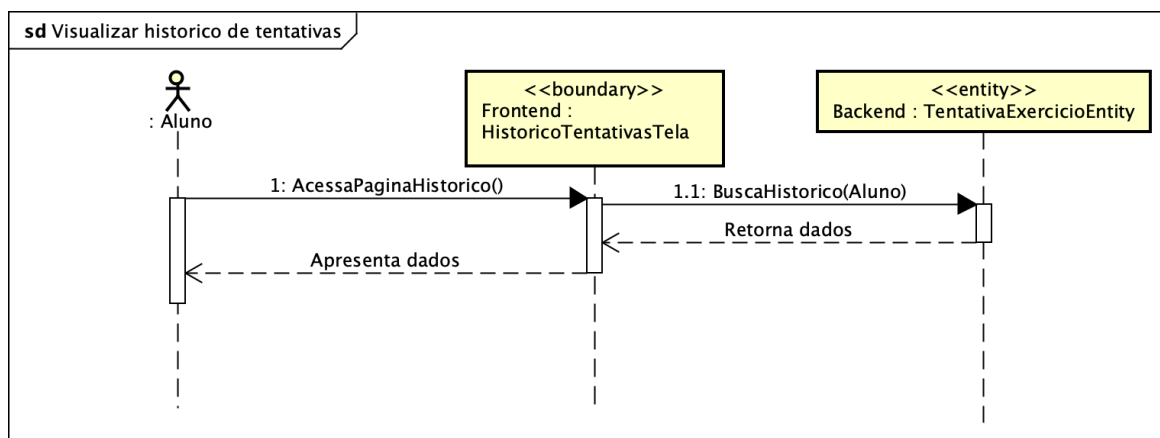
UC22 – Visualiza listas de exercícios



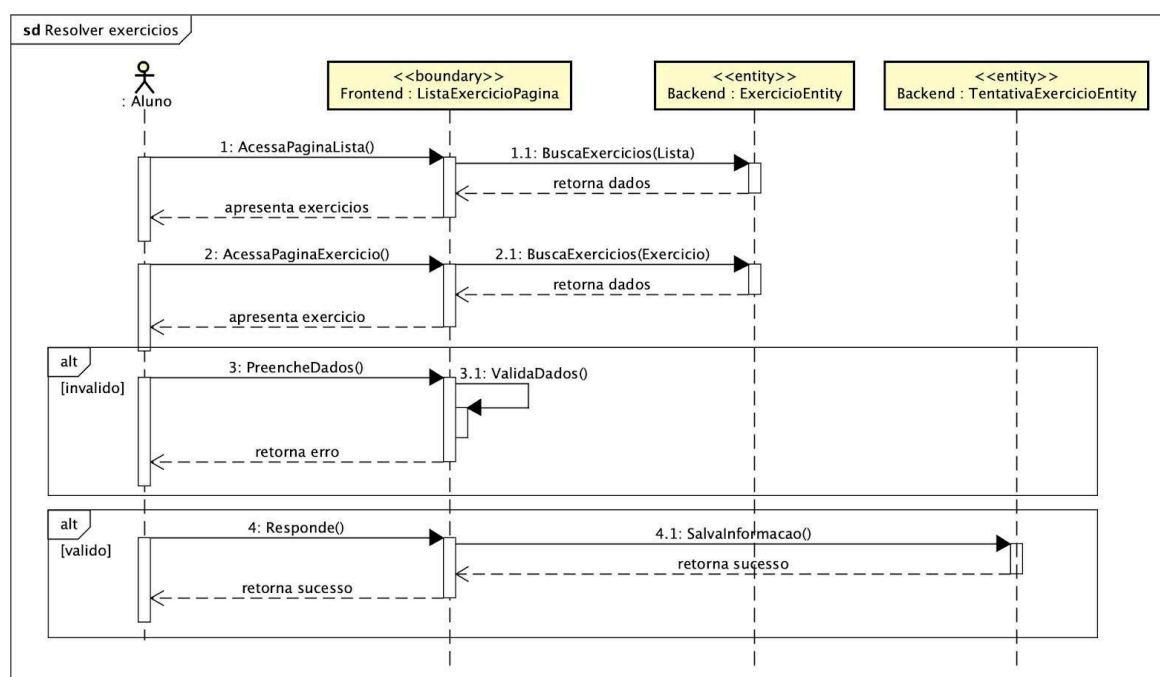
UC23 - Visualizar exercícios de uma lista



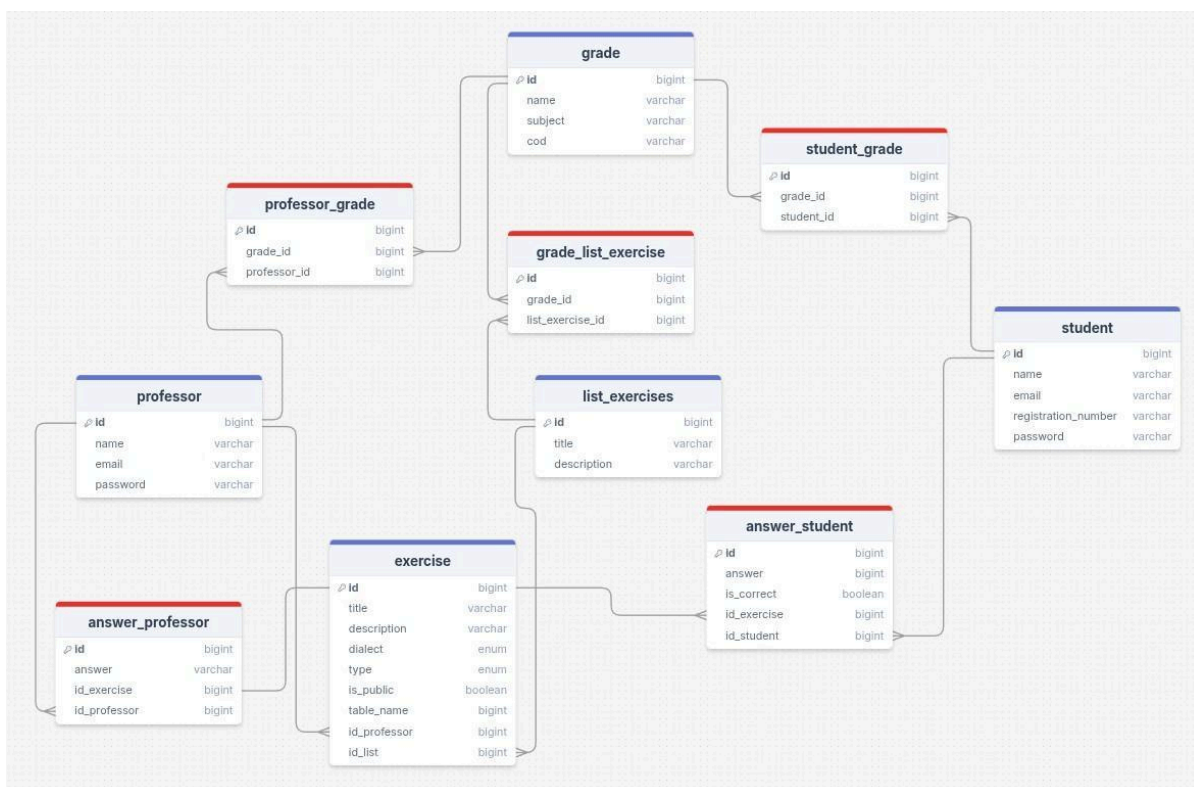
UC24 - Visualizar histórico de tentativas do exercício



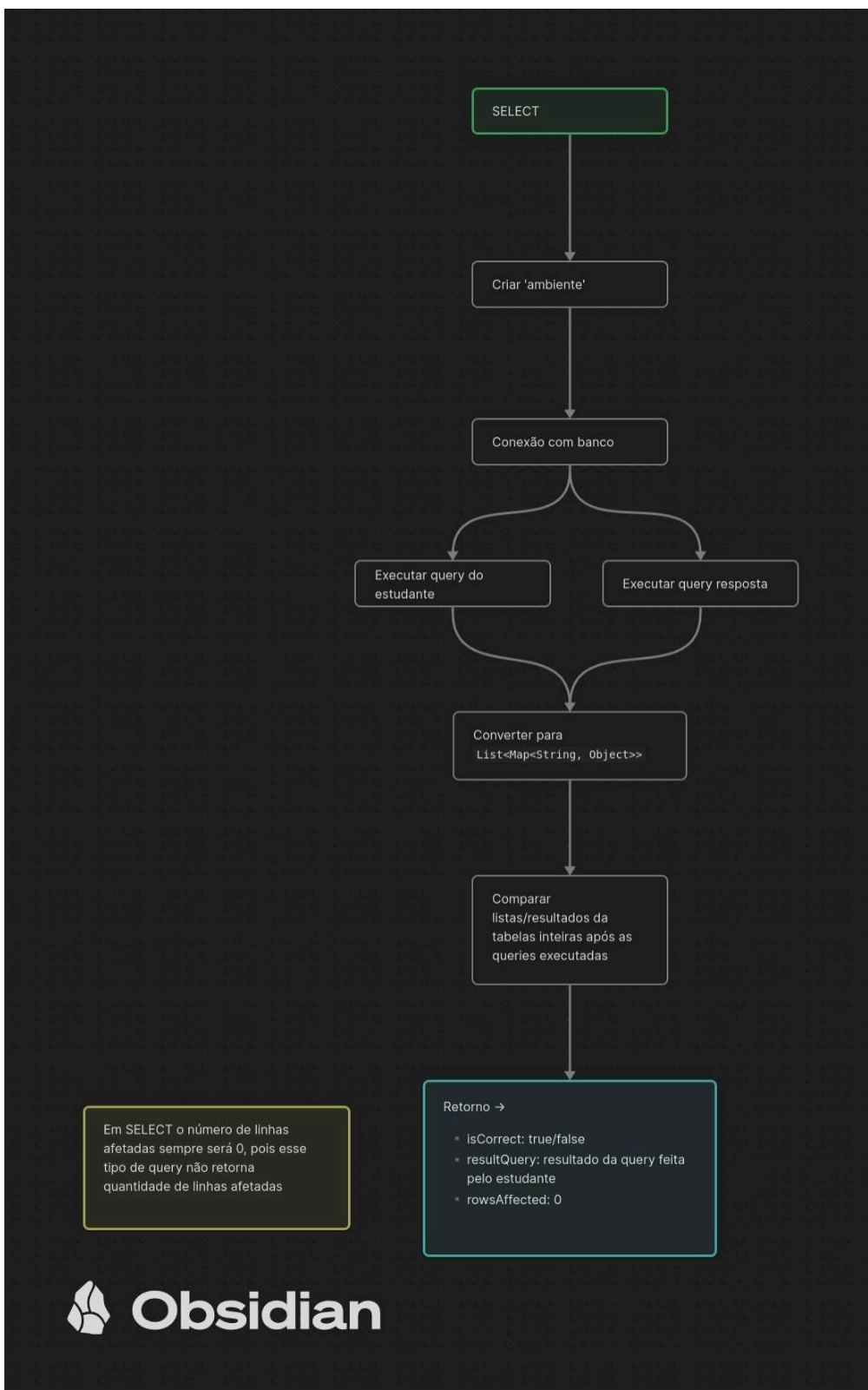
UC25 - Resolver exercícios



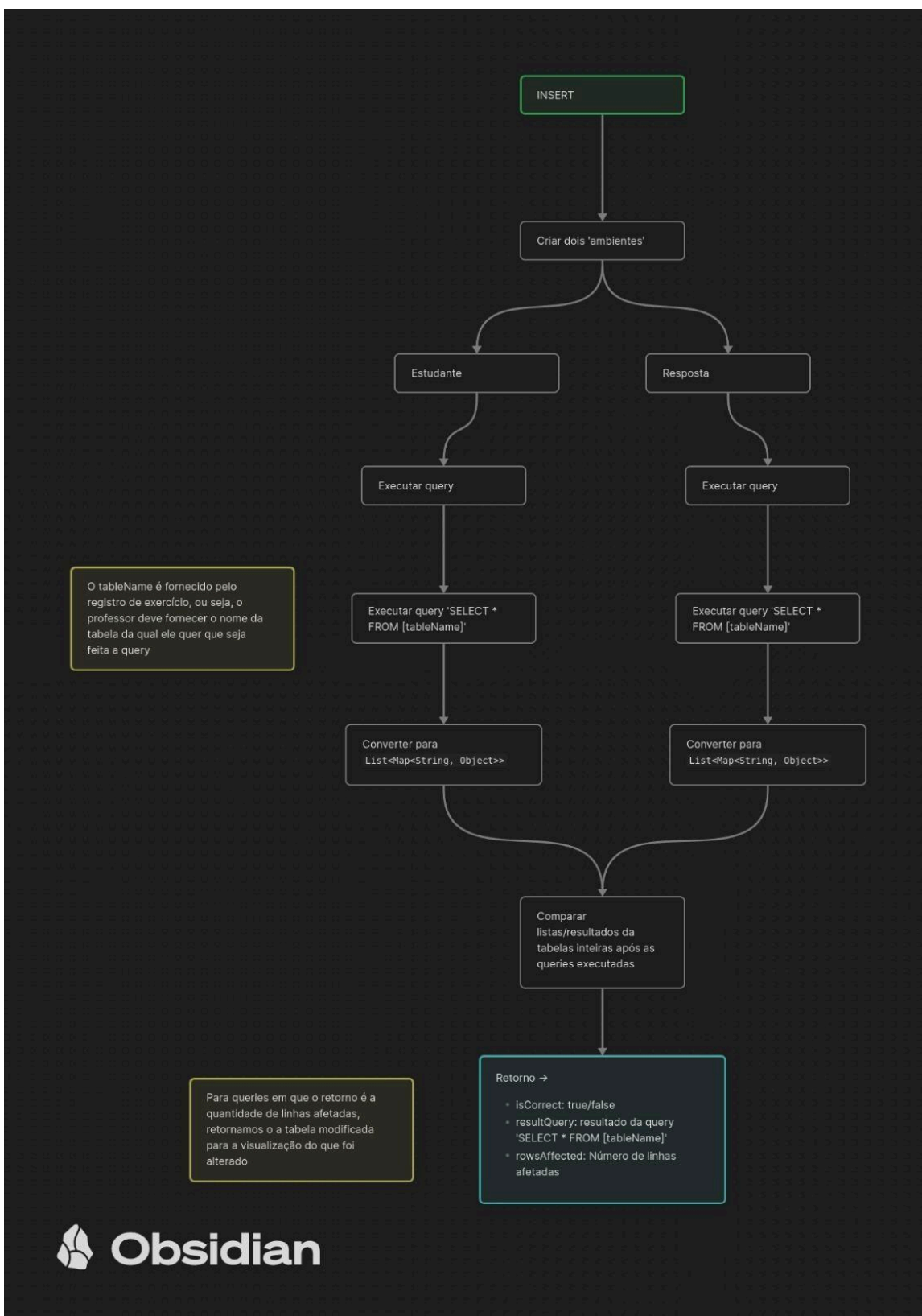
APÊNDICE F – MODELO LÓGICO DE DADOS



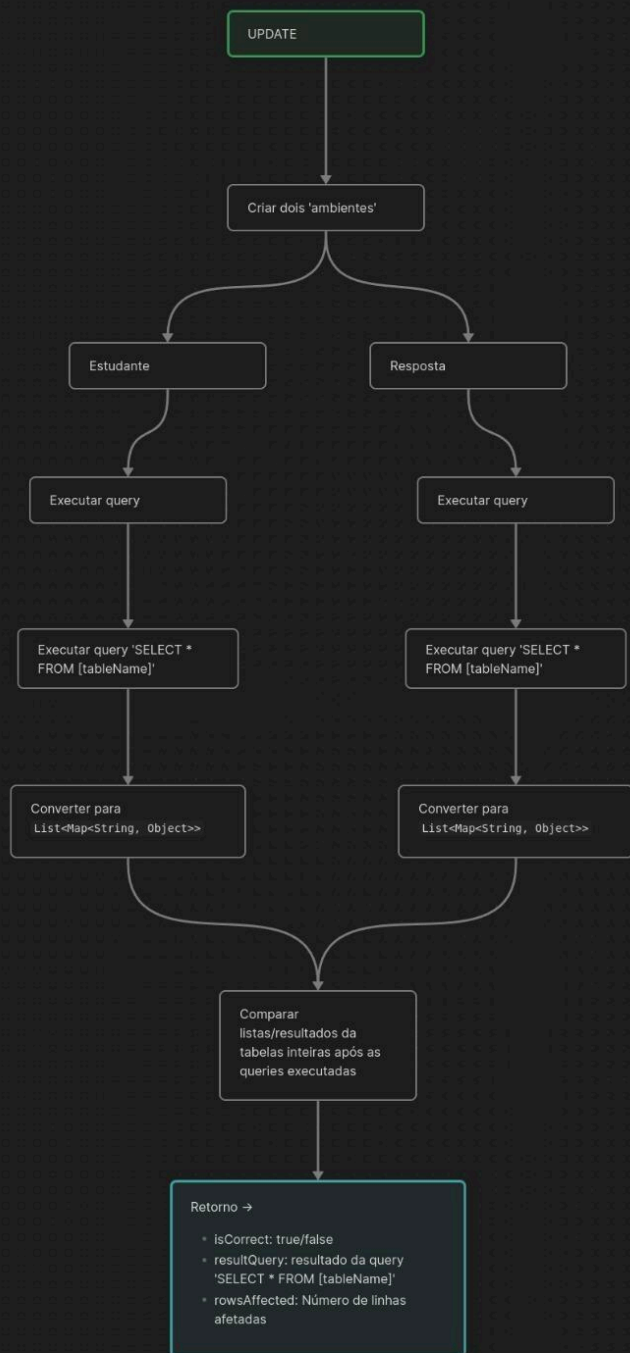
APÊNDICE G – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO SELECT



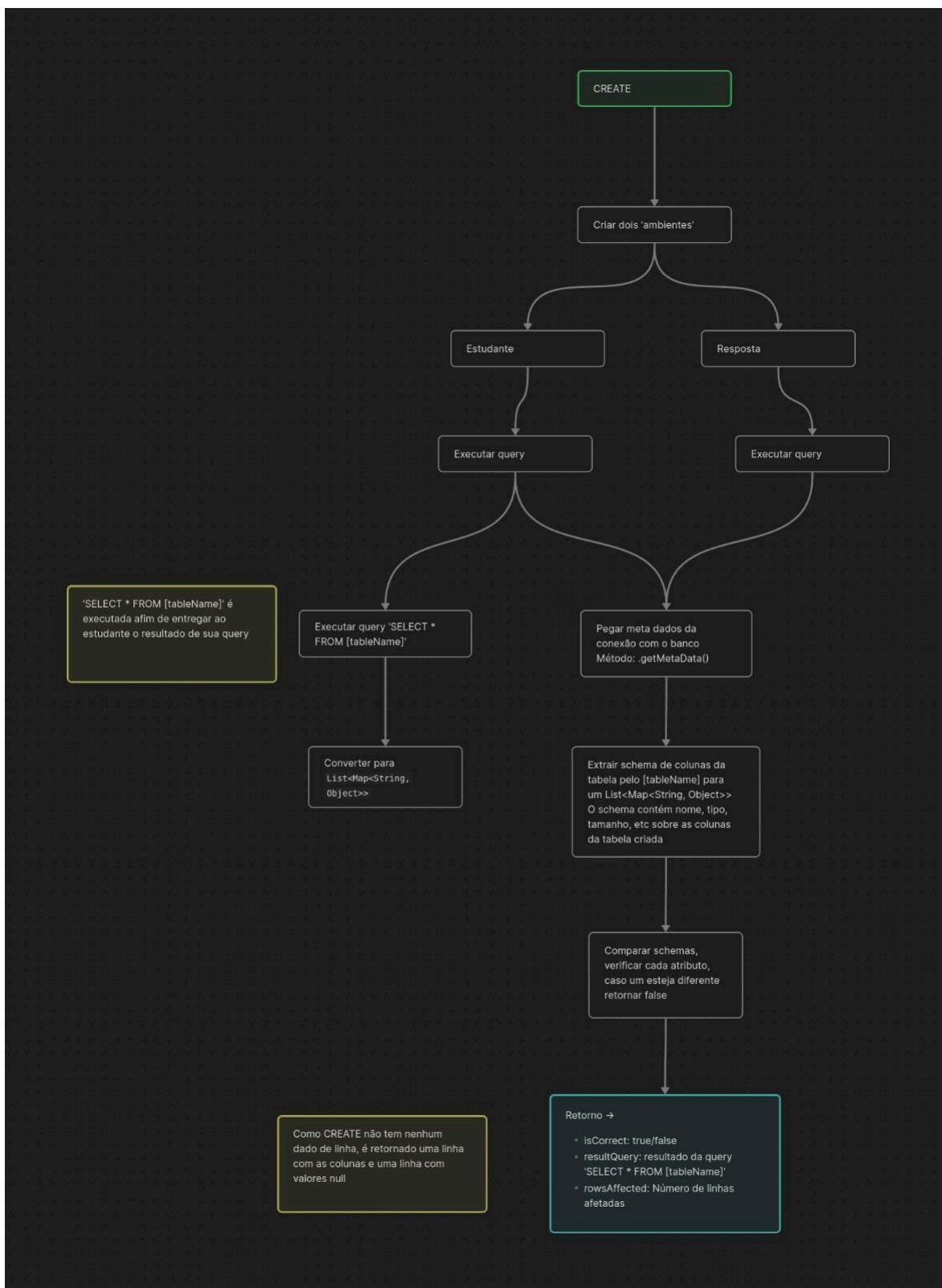
APÊNDICE H – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO INSERT



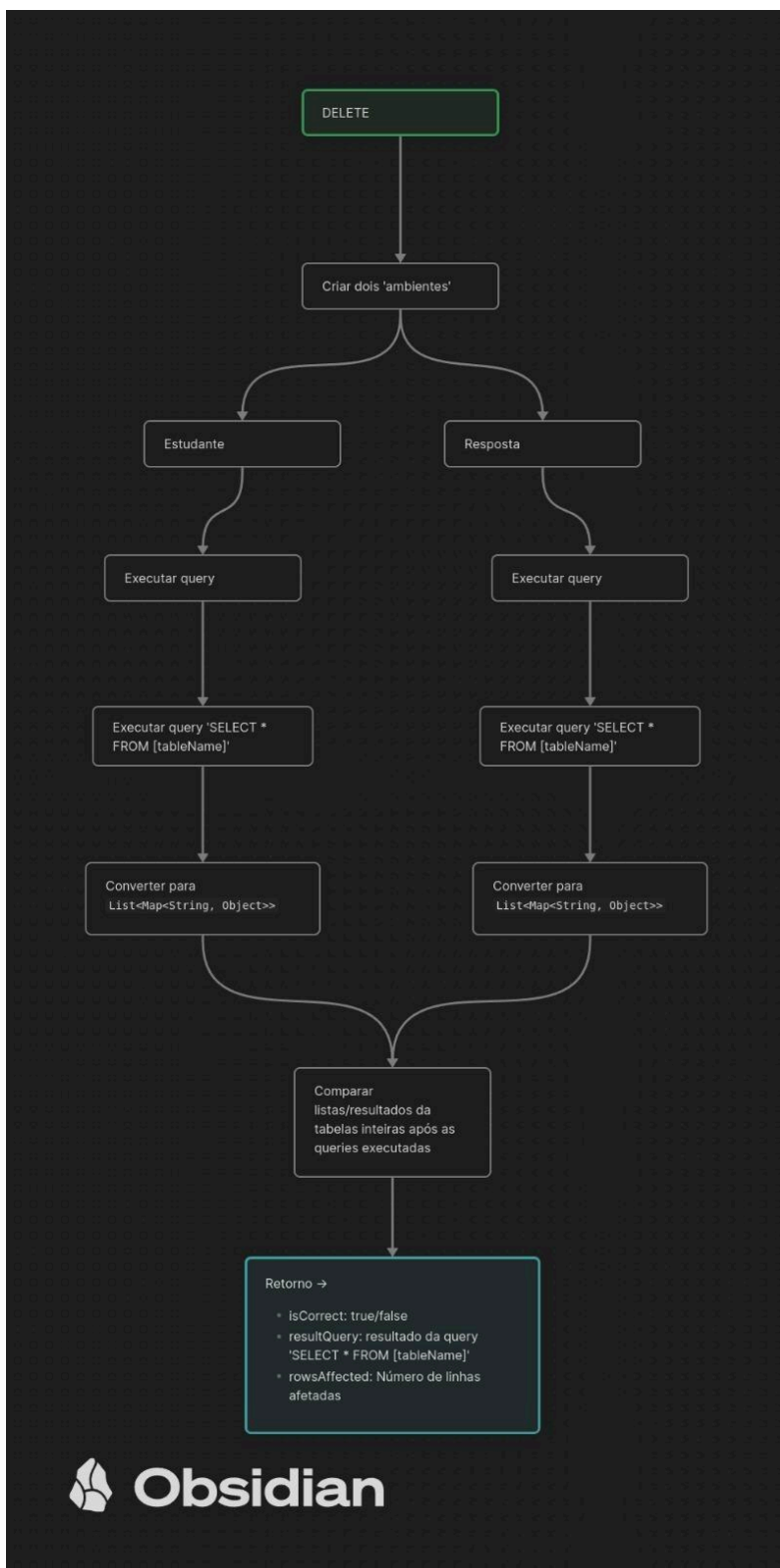
APÊNDICE I – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO UPDATE



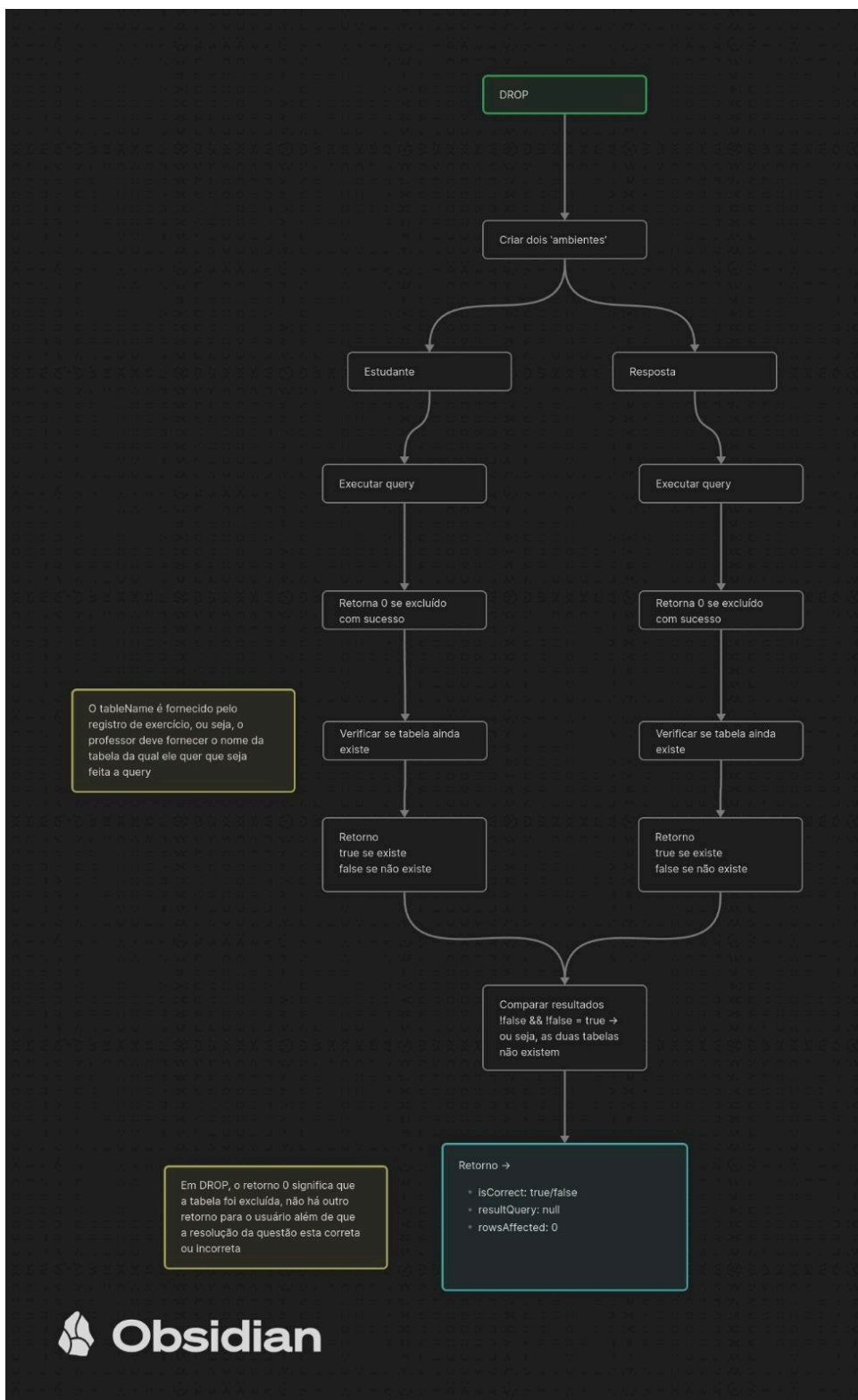
APÊNDICE J – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO CREATE



APÊNDICE K – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO DELETE



APÊNDICE L – FLUXOGRAMA DE PROCESSAMENTO DE QUESTÕES DO TIPO DROP





UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
Rua Alcides Vieira Arcoverde 1225, - - Bairro Jardim das Américas, Curitiba/PR,
CEP 81520-260
Telefone: 3360-5000 - <https://ufpr.br/>

Despacho nº 785/2025/UFPR/R/EP

Processo nº 23075.040118/2025-69

TERMO DE APROVAÇÃO

CLAUDIO HENRIQUE NASCIMENTO DE OLIVEIRA
ISRAEL DE QUADROS
MICHELLY NARITA KURIYAMA

SQLAB: AMBIENTE VIRTUAL DE APRENDIZADO DE LINGUAGEM SQL

Trabalho de Conclusão de Curso aprovado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Curitiba, 04 de julho de 2025

Prof. Dr. Dieval Guizelini
Orientador – SEPT/UFPR

Prof. Dr. Clausius Duque Gonçalves Reis
SEPT/UFPR

Profa. Dra. Jeroniza Nunes Marchaukoski
SEPT/UFPR



Documento assinado eletronicamente por **JERONIZA NUNES MARCHAUKOSKI**,
PROFESSOR ENS BASICO TECN TECNOLOGICO, em 07/07/2025, às 13:23, conforme art.
1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **CLAUSIUS DUQUE GONCALVES REIS**,
PROFESSOR DO MAGISTERIO SUPERIOR, em 07/07/2025, às 17:58, conforme art. 1º, III,
"b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador
7923665 e o código CRC **8EB98756**.
