

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ISABELA TAQUES VITEK

**DESENVOLVIMENTO DA ÁREA DESTINADA À CRIAÇÃO DE OBJETOS DE
APRENDIZAGEM PARA FERRAMENTA DE AUTORIA FARMA**

GUARAPUAVA

2023

ISABELA TAQUES VITEK

**DESENVOLVIMENTO DA ÁREA DESTINADA À CRIAÇÃO DE OBJETOS DE
APRENDIZAGEM PARA FERRAMENTA DE AUTORIA FARMA**

**Development of the Learning Object Creation Area for the FARMA Authoring
Tool**

Monografia de Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Diego Marczal

Coorientadores: Prof. Me. Alex Sandro De
Castilho e Prof. Me. Denis Lucas
Silva

GUARAPUAVA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ISABELA TAQUES VITEK

**DESENVOLVIMENTO DA ÁREA DESTINADA À CRIAÇÃO DE OBJETOS DE APRENDIZAGEM
PARA FERRAMENTA DE AUTORIA FARMA.**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Tecnólogo em Sistemas para Internet do Curso
de Tecnologia em Sistemas para Internet da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data da aprovação: 6/Dezembro/2023

Prof. Diego Marczal
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

Prof. Eleandro Maschio
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

Prof. Roni Fabio Banaszewski
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

GUARAPUAVA
2023

Dedico este trabalho à minha família, por todo
amor, apoio e incentivo.

AGRADECIMENTOS

Com imensa gratidão gostaria de agradecer a todas as pessoas, que de alguma maneira contribuíram com esta jornada, que me deram apoio e incentivo durante todo o percurso deste trabalho, sendo de total diferença para que fosse possível alcançar meu objetivo.

Em especial, quero agradecer aos meus pais, vocês foram extremamente importantes na minha caminhada desde o início e principalmente nesta reta final, foram sempre meu maior apoio, maior incentivo para continuar e sem dúvidas me deram todo o amor. Vocês estiveram presentes nos dias difíceis trazendo calma e palavras de motivação e não tenho palavras para expressar minha gratidão por ter vocês.

Quero agradecer também aos meus avós, que foram parte da minha base educacional, desde pequena sempre me deram apoio e incentivo para evoluir.

Gostaria também de agradecer minha amiga Amanda, minha dupla e parceira, sua amizade, apoio e ajuda foram fundamentais durante o processo. Agradeço por todos os momentos que compartilhamos nossas alegrias e nossas aflições, com toda certeza sua amizade me trouxe apenas alegrias e sem dúvidas é muito importante para mim.

Quero agradecer todos os meus amigos, os que fiz durante o curso e em outras fases da minha vida, o apoio de vocês ajudou muito a continuar em frente, cada um me ajudou de alguma forma e trouxe alegria na minha caminhada, me incentivaram e torceram pelo meu sucesso.

A cada professor que tive a oportunidade de ser aluna meus sinceros agradecimentos, sem vocês nada seria possível. Vocês foram essenciais no meu aprendizado e evolução, sempre dedicados e pacientes, ensinando com perfeição sempre.

Sou imensamente grata também pelo meu orientador, Dr. Diego Marczal, e meus coorientadores, Me. Alex Sandro de Castilho e Me. Dênis Lucas Silva. Vocês tiveram a paciência e a dedicação de me auxiliarem no desenvolvimento deste trabalho, sem dúvidas foram fundamentais para que conseguisse obter êxito. A cada um de vocês, minha mais profunda gratidão, pois tornaram esta jornada não só possível, mas também repleta de aprendizado e crescimento.

Obrigada a todos!

RESUMO

A Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem (FARMA) foi idealizada como um meio que oferecesse funcionalidades para construir Objetos de Aprendizagem (OAs). Seu objetivo é trazer aos professores leigos em programação a possibilidade de criar e gerenciar OAs de forma dinâmica e descomplicada, permitindo que os alunos e professores possam visualizar os erros que ocorreram até obterem êxito na resolução do exercício, e a cada submissão errada do discente, mostra conteúdos que auxiliam a encontrar a solução. Nesta ferramenta, os docentes podem configurar seu exercício com uma explicação, incorporando recursos como vídeos e *links* para contribuir com a compreensão do tema. Com base nisso, o desenvolvimento deste trabalho teve como propósito a implementação *Frontend* e *Backend* da área de professor da FARMA, unificando as versões já existentes em um único projeto, permitindo a criação de Objetos de Aprendizagem, que pode conter vídeos, gráficos, imagens, dentre outros. Durante a criação, o professor poderá configurar também as dicas que aparecerão em casos de erros, bem como a sequência em que serão exibidas, juntamente com a possibilidade de gerenciar esses OAs (edição, visualização e exclusão).

Palavras-chave: objeto de aprendizagem; atividades; exercício; ferramenta; sistema.

ABSTRACT

The Authoring Tool for Error Remediation with Learning Mobility (FARMA) was designed as a means that offers functionalities to build objects of Learning (LOs). Its objective is to bring lay programming teachers the possibility of creating and managing OAs in a dynamic and uncomplicated way, allowing students and teachers can visualize the mistakes they made until they were successful in solving the exercise, and with each incorrect submission by the student, it shows content that helps to find the solution. In this tool, teachers can configure their exercise with a explanation, incorporating resources such as videos and links to contribute to the understanding of the theme. Based on this, the purpose of developing this work was to implement Frontend and Backend of the FARMA teacher area, unifying existing versions in a single project, allowing the creation of Learning Objects, which can contain videos, graphics, images, among others. During creation, the teacher can also configure the tips that appear in cases of errors, as well as the sequence in which they will be displayed, in addition to the possibility of managing these OAs (editing, viewing and deleting).

Keywords: object learning; activities; exercise; tool; system.

LISTA DE FIGURAS

Figura 1 – Caso de uso do professor que será desenvolvido	12
Figura 2 – <i>Home</i> da FARMA	13
Figura 3 – Criar introdução na FARMA	14
Figura 4 – Criar exercício na FARMA	15
Figura 5 – Criar/editar OA na FARMA	15
Figura 6 – <i>Home FARMA reborn</i>	16
Figura 7 – Criar/editar OA's FARMA <i>reborn</i>	16
Figura 8 – Criar introdução na FARMA <i>reborn</i>	17
Figura 9 – Criar exercício na FARMA <i>reborn</i>	17
Figura 10 – Fluxo de autenticação	20
Figura 11 – Planejamento	21
Figura 12 – Fluxo de desenvolvimento	22
Figura 13 – Git Flow	23
Figura 14 – CI/CD	24
Figura 15 – <i>Projects</i> FARMA API	30
Figura 16 – Rota <i>Index</i> da API	30
Figura 17 – <i>Branch</i> de modificação	31
Figura 18 – Rota <i>Index</i> da API com a modificação	31
Figura 19 – <i>Commit</i> das modificações	31
Figura 20 – Teste <i>linters</i> - exemplo de uma falha	32
Figura 21 – Teste bem sucedido	32
Figura 22 – Criação da PR	33
Figura 23 – Execução do CI	33
Figura 24 – Aprovação da PR	34
Figura 25 – Comentários na PR	34
Figura 26 – PR aprovada	35
Figura 27 – Diagrama Entidade Relacionamento	37
Figura 28 – Correção requisitada	39
Figura 29 – Falha nos testes	40
Figura 30 – Painel <i>coverage</i>	41

Figura 31 – Erro de rotas	41
Figura 32 – Erro de testes	47
Figura 33 – Erro de qualidade de código	47
Figura 34 – Upload de imagem	48
Figura 35 – <i>Home</i>	51
Figura 36 – <i>Login</i>	52
Figura 37 – Registrar novo usuário	52
Figura 38 – Objetos de Aprendizagem	53
Figura 39 – Novo OA	53
Figura 40 – Editar OA	54
Figura 41 – Visualizar OA	54
Figura 42 – Relação dos usuários	61
Figura 43 – Caso de uso visitante	61
Figura 44 – Caso de uso aluno	62
Figura 45 – Caso de uso professor	62

LISTA DE ABREVIATURAS E SIGLAS

Siglas

API	<i>Application Programming Interface</i>
CD	<i>Continuous Delivery</i>
CI	<i>Continuous Integration</i>
CI/CD	<i>Continuous Integration/Continuous Delivery</i>
CORS	<i>Cross-Origin Resource Sharing</i>
CRUD	<i>Create, Read, Update, Delete</i>
CSS	<i>Cascading Style Sheets</i>
FARMA	Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
OA	Objeto de Aprendizagem
PR	<i>Pull Request</i>
SAML	<i>Security Assertion Markup Language</i>
SI	Sistemas para Internet
SSH	<i>Secure Shell Protocol</i>
SWT	<i>Standard Widget Toolkit</i>
UTFPR-GP	Universidade Tecnológica Federal do Paraná, câmpus Guarapuava
VPS	<i>Virtual Private Server</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Objetivos	12
1.1.1	Objetivo geral	12
1.1.2	Objetivos específicos	12
2	FARMA	13
3	MATERIAIS E MÉTODOS	19
3.1	Materiais	19
3.1.1	<i>JSON Web Token (JWT)</i>	19
3.2	Métodos	20
3.2.1	Processo de desenvolvimento	20
4	ANÁLISE E PROJETO	25
4.1	Requisitos	25
4.1.1	Requisitos funcionais	25
4.1.2	Requisitos não funcionais	26
4.2	Escolha do framework CSS	26
4.3	Ambiente de desenvolvimento	27
4.3.1	Configuração da API	27
4.3.2	Exemplo de funcionamento da API	30
5	DESENVOLVIMENTO DO SISTEMA	36
5.1	Procedimentos Iniciais	36
5.2	FARMA API	36
5.2.1	Desenvolvimento do CRUD de OA	37
5.2.2	Desenvolvimento do CRUD de introduções	38
5.2.3	Desenvolvimento do CRUD de exercícios	40
5.2.4	Desenvolvimento do CRUD de passos de solução	41
5.2.5	Desenvolvimento do CRUD de dicas	42
5.2.6	Desenvolvimento da autenticação de usuários com <i>token</i>	42
5.2.7	Desenvolvimento da função para duplicar dicas	43
5.2.8	Desenvolvimento da função para duplicar passos de solução	43
5.2.9	Desenvolvimento da função para duplicar exercícios	44

5.2.10	Desenvolvimento da função para duplicar introduções	44
5.2.11	Desenvolvimento da função para duplicar OA	44
5.2.12	Desenvolvimento da função para atualizar posições das dicas	45
5.2.13	Desenvolvimento da função para atualizar posições dos passos de solução .	45
5.2.14	Desenvolvimento da função para atualizar posições das introduções e exer- cícios dentro do OA	46
5.2.15	Desenvolvimento da função para controlar exibição das dicas	46
5.2.16	Desenvolvimento do <i>upload</i> de imagem no OA	47
5.2.17	Documentação da API	48
5.3	FARMA FRONTEND	48
5.3.1	Desenvolvimento do CRUD de OA com <i>React</i>	49
5.3.2	Desenvolvimento do <i>login</i> e CRUD de OA com <i>Next</i> e <i>Tailwind</i>	49
5.3.3	Métricas	49
6	RESULTADOS	51
7	CONCLUSÃO	56
7.1	Trabalhos Futuros	56
	REFERÊNCIAS	58
	APÊNDICE A CASOS DE USO DOS USUÁRIOS DO SISTEMA FARMA	61
	A.1 Relação dos usuários.	61
	A.2 Caso de Uso Visitante.	61
	A.3 Caso de Uso Aluno.	62
	A.4 Caso de Uso Professor.	62
	APÊNDICE B DOCUMENTAÇÃO DA API	64

1 INTRODUÇÃO

A educação é o ato de educar alguém ou a si próprio em questões culturais, científicas, religiosas ou em bons modos de interação e comunicação com outras pessoas (VICENTE; ROCHA, 2022). Para isso, é preciso adotar métodos e processos de forma a assegurar a formação e o aprendizado.

Aprendizagem é o processo pelo qual as competências, habilidades, conhecimentos, comportamento ou valores são adquiridos, ou modificados, como resultado de estudo, experiência, formação, raciocínio e observação¹.

Para ensinar geografia, além dos textos e discursos, os professores utilizam mapas, fragmentos de rochas e vídeos para ilustrar os temas de estudo em sala de aula. Isto é um exemplo, no qual um professor recorre a outros objetos, além do quadro-negro e giz, para propiciar ao aluno o entendimento e, conseqüente, aprendizado sobre algo. Através da experimentação e vivência rotineira, ambas lúdicas ou não, dúvidas e confusões de entendimento são trabalhados de forma a saná-las. Com o avanço da tecnologia, outros objetos de apoio ao ensino, como, por exemplo, televisões, projetores, computadores, dentre outros, começam a compor o ambiente escolar, permitindo aos alunos visualizarem de uma maneira melhor o que se quer transmitir (KENSKI, 2003).

A importância de garantir a educação a todos começa nos primeiros anos de vida, onde a criança é alfabetizada, com curiosidade, aprende a escrever seu nome e a ler uma placa na rua. São conhecimentos básicos, mas essenciais durante toda vida. Contudo, muitos fatores contribuíram para a precariedade da educação, tendo como resultado os analfabetos funcionais (LORENZO, 2007). Um destes fatores foi o surgimento de outras coisas “mais interessantes” que o aprender, como os computadores, jogos e, atualmente, os *smartphones*.

O acesso cada vez maior a músicas, filmes, jogos e redes sociais propiciado pelos aparelhos já citados, além da “*ilusão do conhecimento*”² (SANTOS, 2017), alimentada pelos diversos vídeos disponibilizados na Internet, teve lugar na vida das pessoas, colocando a educação e aprendizado extraclasse de lado.

Neste novo mundo, a educação por meio de seus gestores, professores e pesquisadores, precisou ser revista. Uma destas revisões trouxe a necessidade de se incluir mais objetos para ensinar. Os softwares educacionais são uma opção viável no processo de ensino-aprendizagem, de forma a complementar a transferência de conhecimentos, auxiliando os professores em atingir o aprendizado de seus alunos (BEHAR, 1993).

Na matemática, por exemplo, os professores dispõem do GeoGebra³, um aplicativo que pode ser usado na Internet, ou instalado em computadores e celulares, o qual combina concei-

¹ <https://pt.wikipedia.org/wiki/Aprendizagem>

² <https://www.bbc.com/portuguese/vert-cap-62985145>

³ <https://www.geogebra.org/calculator>

tos de geometria e álgebra, permitindo aos professores demonstrarem de forma visual, figuras e gráficos, sendo usado para ajudar a resolver vários modelos de equações, principalmente no ensino médio. Também se pode citar o Google Earth⁴, com ele é possível explorar imagens de satélite do mundo todo, terrenos e construções em 3D, podendo ser utilizado tanto nas aulas de geografia sobre o estudo da cartografia, como também ao se estudar pontos de relevância como as sete maravilhas do mundo. Contudo, mesmo sabendo da existência desses, dentre tantos outros, ainda existe a carência por ferramentas computacionais para uso prático em sala de aula ou extra-classe.

Isso significa que trazer as tecnologias para o ambiente educativo pode tornar o processo de ensino e aprendizagem mais prazeroso, mais chamativo e significativo para aquele que aprende e mais dinâmico para aquele que educa (SILVA; CORREA, 2014).

Dentre os mais variados assuntos, um que se destaca, quando se trata de dificuldade de aprendizagem, é a matemática, ou conteúdos de exatas em geral. Segundo Santos, França e SANTOS (2007) “A Matemática não é uma ciência cristalizada e imóvel; ela está afetada por uma contínua expansão e revisão dos seus próprios conceitos”, com isso entende-se que a dificuldade ocorre pela forma com que a matemática foi construída. A matemática é um dos conteúdos primordiais que devem ser aprendidos, assim como o português é essencial para o desenvolvimento humano, desde uma conta básica de troco após uma compra até assuntos mais complexos como finanças.

Com este foco, no período de 2015 a 2018, um grupo de alunos do Curso de Graduação em Sistemas para Internet (SI) da Universidade Tecnológica Federal do Paraná, câmpus Guarapuava (UTFPR-GP), participaram do desenvolvimento de uma ferramenta para auxiliar no processo de ensino-aprendizagem da matemática, idealizado pelo professor Diego Marczal em sua tese de doutorado no ano de 2014 (MARCZAL *et al.*, 2014). O objetivo é dispor de um ambiente (**Ferramenta**) onde cada professor pode criar atividades (**Autoria**) para testar, avaliar e corrigir o processo de resolução adotado pelos alunos (**Remediação de Erros**). Um sistema *Web*, acessível a partir de um computador localizado em qualquer parte do mundo, proporcionando aos alunos (**Mobilidade na Aprendizagem**) e professores mobilidade no seu uso.

Na Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem (FARMA), um Objeto de Aprendizagem (OA) é apresentado em formato de atividades que são formados por uma explicação do conteúdo seguido de exercícios referente ao tema, permitindo o professor montar essa estrutura de uma forma mais fácil e o aluno testar o entendimento do assunto abordado, logo em seguida verificando seus resultados com auditoria de seus erros. Também permite tanto o aluno quanto o professor visualizar as submissões dos exercícios até a resposta final, com isso podendo entender melhor as dificuldades existentes no conteúdo abordado.

⁴ <https://www.google.com.br/intl/pt-BR/earth/>

Foi proposto, neste trabalho, o desenvolvimento *Frontend* e *Backend* da área de professores da FARMA, disponibilizando recursos para a criação e gerenciamento de OA's, conforme a Figura 1. Este módulo disponibiliza aos docentes, principalmente aqueles sem conhecimentos em programação de computadores, a possibilidade de criar OA's interativos, caracterizados como pequenos softwares educacionais de forma mais assertiva e descomplicada.

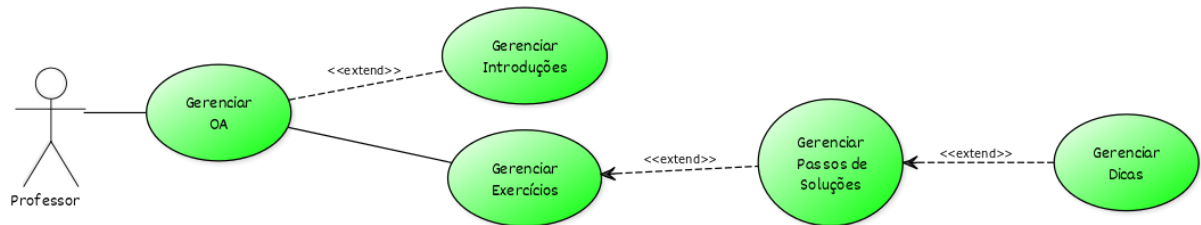


Figura 1 – Caso de uso do professor que será desenvolvido

Fonte: Autoria própria (2023).

O desenvolvimento desta área motivou-se principalmente por existirem várias contribuições anteriores que necessitavam se adequar todas em uma única versão estável da FARMA, corrigindo também as carências que ainda permanecem ao criar um OA. Assim, inicia-se uma nova versão da FARMA que teve sua criação de modo incremental, no qual primeiramente foi desenvolvido a criação de OA's na área do professor.

1.1 Objetivos

Almeja-se neste trabalho o desenvolvimento da área para criação de um OA contendo todos os seus elementos (introduções, materiais de apoio e exercícios), bem como como a possibilidade de configurar as dicas dos exercícios (ordem e quantidade) e realizar as operações de gerenciamento (visualização, edição e listagem) conforme descrito anteriormente.

1.1.1 Objetivo geral

Implementar o módulo da área dos professores, destinado à criação e gerenciamento de OA's da FARMA.

1.1.2 Objetivos específicos

- Entender as necessidades do ponto de vista de professor(es) de matemática na criação de Objetos de Aprendizagem na FARMA;
- Entender o funcionamento da FARMA;
- Desenvolver o módulo de criação de objetos da FARMA.

2 FARMA

A Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem teve sua concepção, projeto e desenvolvimento com a tese de doutorado de Marczal *et al.* (2014). Estimulado pela ausência de meios de averiguar o processo de ensino e aprendizagem, por meio de tentativas de resolução e erros, das ferramentas educativas, como pode-se averiguar no trabalho de Marczal e Direne (2011) que foca na demanda da implementação de um gerenciador de OA's atrativo que utilize os erros do educando ao resolver questões de forma que auxilie seu entendimento. E no trabalho de doutorado de Leite (2013), onde foi desenvolvida uma pesquisa relacionada ao uso e aplicação de remediação de erros.

O objetivo da ferramenta é disponibilizar um ambiente para a aprendizagem de conceitos matemáticos. Para isso, a ferramenta tem embutido em seu escopo Objetos de Aprendizagem, com o intuito de dar flexibilidade na forma de abordar os temas matemáticos. Desta forma, monitores e professores de matemática podem trabalhar conceitos adotando viés e objetivos personalizados a cada aluno. A primeira versão da ferramenta, elaborada por Marczal *et al.* (2014), está disponível na FARMA educacional¹, conforme a Figura 2.



Figura 2 – Home da FARMA
Fonte: <http://farma.educacional.mat.br>.

Objetos de Aprendizagem são qualquer meio digital ou não, para transmitir conhecimento, como animações e simulações (AGUIAR; FLÔRES, 2014; TAROUCO *et al.*, 2014). Exemplos clássicos de OA's muito utilizados nos anos iniciais são jogos, cruzadinhas, caça-

¹ <http://farma.educacional.mat.br/>

palavras, dentre outros. Entretanto vem ganhando destaque formas de misturar a tecnologia ao ensino, trazendo mais interatividade e curiosidade ao aprender.

Na FARMA o Objeto de Aprendizagem é composto por:

- Textos descritivos e explicativos sobre o tema de estudo (introduções), que podem conter figuras, *links* de vídeos e gráficos;
- É possível criar exercícios (Enunciados), que abrange uma ou mais questões;
- As questões devem incluir dicas para auxiliar os alunos quando ocorrer um erro. Cada tentativa de resolução das questões geram um histórico para análise de erros para entender o processo até a resposta final.

Para criar um OA na FARMA, atualmente disponível, seguiu o fluxo conforme a Figura 3, onde está criando uma nova introdução para o OA, adicionando um título e conteúdo. Na Figura 4 observa-se a tela para criar um novo exercício, adicionando um título e conteúdo. E na Figura 5 visualiza-se a página que permite acessar os formulários de criação de novas introduções e exercícios para um OA, adicionar questões para os exercícios, ou editar os já existentes.

A imagem mostra a interface de usuário da FARMA para criar uma nova introdução. No topo, há uma barra de navegação com links: Home, Meus Objetos de Aprendizagem, Conteúdos do OA OA Teste / nova, e um botão 'Criar Introdução' em destaque. Abaixo, o título da seção é 'Nova introdução para o OA'. Há um campo de texto para 'Título *' e um campo maior para 'Conteúdo *'. O campo de conteúdo possui uma barra de ferramentas rica com opções de formatação de texto (negrito, itálica, sublinhado, cor de texto, tamanho de fonte) e inserção de elementos (imagens, vídeos, tabelas, links). Abaixo do campo de conteúdo, há uma opção 'Publicar' com uma caixa de seleção desativada e um botão 'Salvar' em azul.

Figura 3 – Criar introdução na FARMA

Fonte: <http://farma.educacional.mat.br>.

Figura 4 – Criar exercício na FARMA
Fonte: <http://farma.educacional.mat.br>.

Figura 5 – Criar/editar OA na FARMA
Fonte: <http://farma.educacional.mat.br>.

Neste sistema é possível encontrar três tipos de usuários que interagem de formas diferentes com a ferramenta, sendo eles: o professor, o aluno e o visitante, conforme apresentados nos diagramas de caso de uso no Capítulo A de apêndices.

No período entre 2015 e 2018 ocorreu o “Redesing e Refatoração da Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem – FARMA”, envolvendo acadêmicos do curso de Graduação em SI. Com o objetivo de reestruturar o sistema, organizando suas estruturas internas e telas. Resultando na segunda versão da ferramenta, chamada de FARMA *Reborn*, que está em fase de desenvolvimento, apresentada na Figura 6.

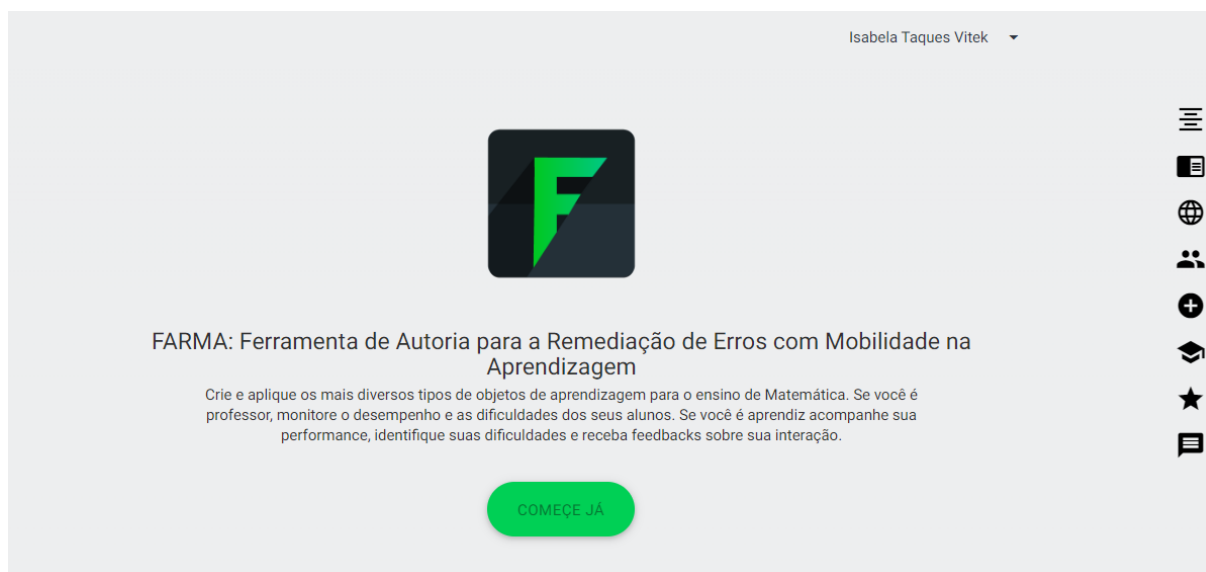


Figura 6 – Home FARMA reborn

Fonte: <http://farma-reborn.educacional.mat.br>.

Nas figuras seguintes é possível visualizar o processo para criar OA's na FARMA reborn, onde, na Figura 7, identifica-se a tela que traz as informações dos conteúdos já criados, e concede acesso para criar novos ou editar, na Figura 8, visualiza-se a tela para criar introdução e na Figura 9, a tela para criar exercício. Todavia nas imagens abaixo é possível notar um visual mais atrativo, com os elementos distribuídos na tela de uma maneira mais interessante e agradável.

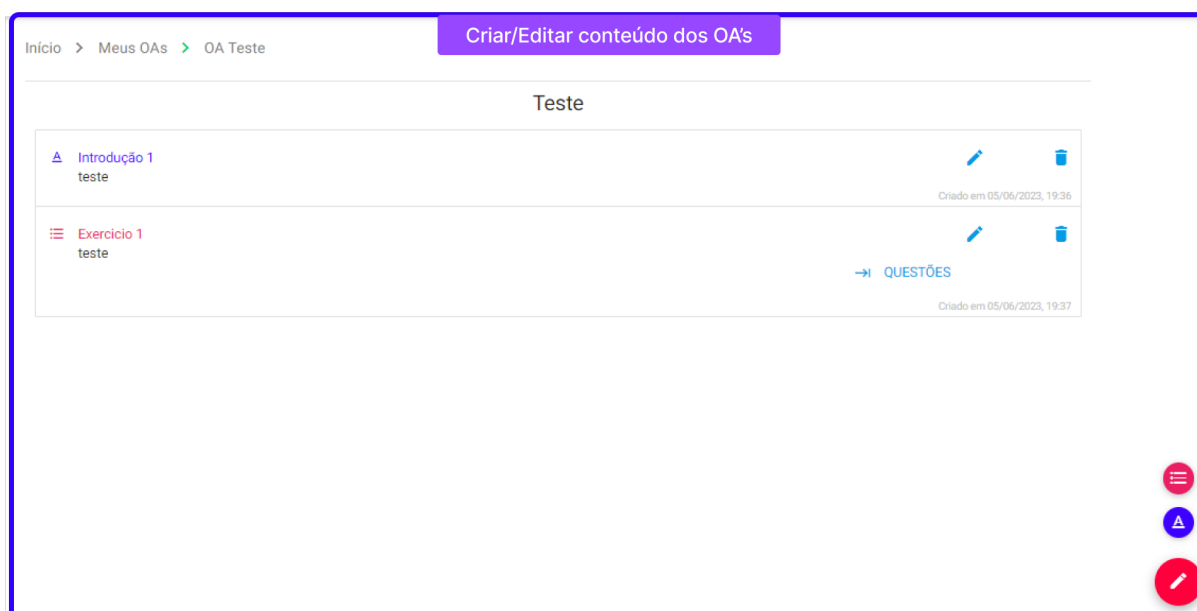


Figura 7 – Criar/editar OA's FARMA reborn

Fonte: <http://farma-reborn.educacional.mat.br>.

Figura 8 – Criar introdução na FARMA reborn
 Fonte: <http://farma-reborn.educacional.mat.br>.

Figura 9 – Criar exercício na FARMA reborn
 Fonte: <http://farma-reborn.educacional.mat.br>.

A FARMA já foi foco de estudo para alguns trabalhos. Em seu doutorado, Silva e Direne (2015) focou na adaptação sequencial de exercícios relacionando a sua dificuldade e o desempenho dos estudantes, ocasionado a implementação de modificações na FARMA nomeado como ADAPTFARMA. No trabalho de Kutzke e Direne (2016), foi foco a possibilidade de

utilização dos erros do aprendiz como forma de aprimorar seu conhecimento. Em trabalhos de mestrado, a FARMA foi foco de Moura e Peres (2017) avaliando o impacto da retroação na aprendizagem com auxílio da FARMA. E Santos (2022), em seu trabalho de conclusão de curso, desenvolveu o módulo de estatísticas para a FARMA.

O professor Alex S. de Castilho está desenvolvendo seu doutorado utilizando a FARMA Calc. Esta versão da FARMA visa suprir as necessidades encontradas nas versões anteriores mantendo o *design* como o da Figura 2 e voltada apenas para uso em relação a cálculos. Devido a sua temática e relevância, a FARMA é citada em diversos artigos que podem ser encontrados na *farma-reborn*².

² <http://farma-reborn.educacional.mat.br/>

3 MATERIAIS E MÉTODOS

Neste capítulo serão descritos os materiais utilizados para o desenvolvimento deste trabalho, bem como os métodos adotados para alcançar os resultados propostos.

3.1 Materiais

Para o desenvolvimento desse trabalho foi utilizado o *React Next* no *Frontend* e *Ruby Rails* no *Backend*, ambos escolhidos pelo fato de já serem as linguagens utilizadas nas versões anteriores da FARMA, e para *Framework* CSS inicialmente foi escolhido o *Bootstrap* como explicado na seção 4.2 e posteriormente alterado para o *Tailwind* para se adaptar melhor ao projeto conforme descrito na subseção 5.3.1. Para gerenciar o banco de dados foi utilizado o *PostgreSQL*.

Para a organização e gestão do projeto foi usado o *Git WorkFlow*, o *GitLab Flow* para o versionamento e a prática CI/CD. A ferramenta *Projects* do *GitHub* foi utilizada para a organização das tarefas. O software de desenvolvimento de código adotado foi o *Visual Studio Code*.

3.1.1 JSON Web Token (JWT)

Segundo o site oficial¹, o *JSON Web Token (JWT)* oferece um método seguro para a transferência de dados entre duas entidades na forma de um objeto JSON. Ele utiliza uma assinatura digital para assegurar que os dados são autênticos e não foram alterados durante o trânsito. Este método é muito utilizado para troca de informações e autenticação de usuários em sistemas, mantendo o estado do usuário entre diferentes chamadas HTTP em uma aplicação, o que é essencial em API's que não mantêm o estado do usuário, ou seja, API's *stateless*. Um JWT é composto por três partes: um cabeçalho (*header*) que contém o tipo do *token* e sua assinatura, como por exemplo SHA256, um conjunto de reivindicações ou declarações (*payload*), que são os dados adicionais que podem ser públicas e privadas ou registradas e uma assinatura (*signature*) que é as duas partes anteriores codificadas conforme passado no cabeçalho e assinadas.

Na autenticação utilizando JWT, após o usuário realizar o *login*, um JWT é gerado e retornado, conforme a Figura 10, onde exemplifica a requisição de *login*, saindo do cliente até o *Auth0*, para ser validada a autenticação. Em seguida no passo dois o retorno ao cliente, que caso autorizado irá para o passo três, que é o acesso as informações da API.

¹ <https://jwt.io/introduction>

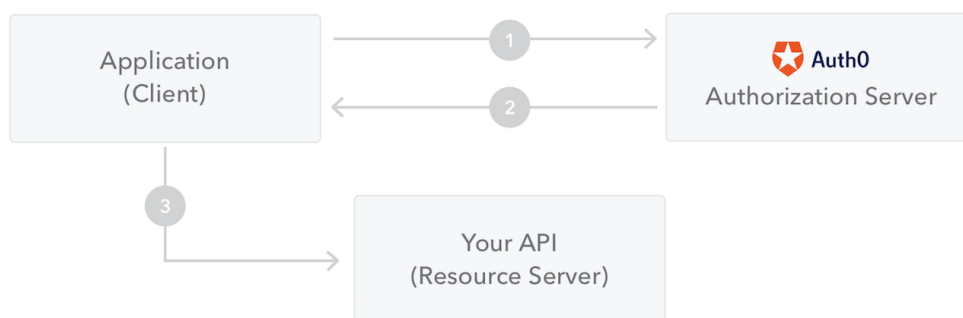


Figura 10 – Fluxo de autenticação

Fonte: <https://jwt.io/introduction>.

Por razões de segurança, especialmente por conter credenciais, é aconselhável que esse *token* tenha um prazo de validade definido, expirando após esse período. Ao utilizar o JWT, é crucial ter cautela para não expor informações sensíveis.

Para acessar rotas protegidas no sistema, o *token* criado no momento do *login* deve ser incluído no cabeçalho das requisições, seguindo o esquema *Bearer*. Isso garante que somente requisições autenticadas possam acessar essas rotas.

Os JWT apresentam vantagens significativas para a transmissão de dados, principalmente devido à sua simplicidade e compactação. Comparados com o SAML, que utiliza XML, os JWTs são mais enxutos porque o JSON é intrinsecamente menos complexo e verboso. Essa compactação os torna ideais para o uso em ambientes onde a eficiência da transmissão de dados é crucial.

Além disso, a segurança dos JWTs é reforçada pela capacidade de serem assinados digitalmente usando um par de chaves pública/privada. Essa característica os coloca à frente dos *Standard Widget Toolkit* (SWT), que se limitam a assinaturas simétricas. A simplicidade na assinatura dos JWTs os torna uma opção segura e prática para a autenticação e transmissão de informações seguras.

Outra vantagem fundamental dos JWTs é a sua compatibilidade e facilidade de integração com diversas linguagens de programação. Isso se deve ao mapeamento direto e eficiente do JSON para objetos, facilitando o uso e aplicação dos JWTs em diferentes plataformas e contextos de desenvolvimento. Essa universalidade contribui para a popularidade e ampla adoção dos JWTs em aplicações modernas, especialmente naquelas que demandam segurança e eficiência na gestão de identidades e na transmissão de dados (INTRODUCTION. . . , 2023).

3.2 Métodos

3.2.1 Processo de desenvolvimento

Inicialmente foi preciso o entendimento do sistema FARMA frente às necessidades dos usuários. Com estas, os requisitos para o sistema podem ser melhor definidos junto aos usuá-

rios, por meio de conversas, reuniões, questionários e *brainstorming*, dentre outras técnicas. A organização dos requisitos foi feita conforme a Figura 11, com o uso do quadro *Kanban*, dividindo as etapas do desenvolvimento. Os requisitos levantados compõem o *Product Backlog*² e, conforme são desenvolvidos, avançam no quadro *Kanban* (*Sprint Backlog*, *Doing*, *Review*, *Done*, *Staging* e *Production*). Na sequência, os requisitos são particionados em tarefas de implementação, ou seja, funcionalidades menores e independentes para serem desenvolvidas.

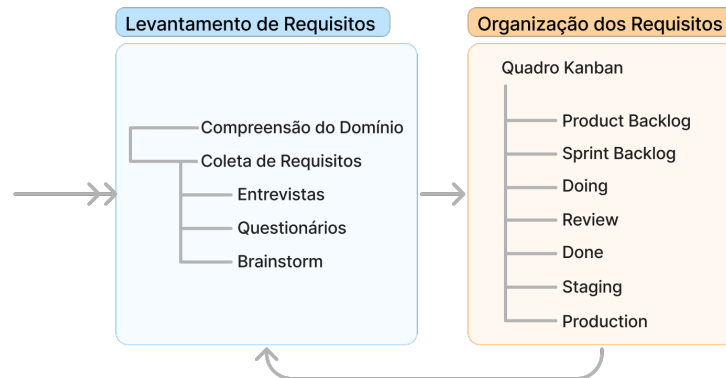


Figura 11 – Planejamento
Fonte: Autoria própria (2023).

Para o fluxo de desenvolvimento das tarefas, o procedimento seguiu conforme a Figura 12. Primeiramente, clona-se o projeto e realiza *checkout* para a *branch* principal (*main*), sendo necessário executar o comando “*git pull*” para manter o projeto local atualizado. Escolhe-se a tarefa para desenvolver com base na sua prioridade e cria-se uma *branch* para iniciar sua implementação. Após finalizada a implementação, é realizado um *commit* e envio das modificações para o repositório. Na sequência, abre-se uma *Pull Request* (PR) para revisão do código, onde em caso de aprovação é enviada para o ambiente de testes. Se não for aprovada, ela retornará para o desenvolvimento a fim de que sejam feitas as devidas correções.

² Lista contendo os requisitos que precisam estar no produto

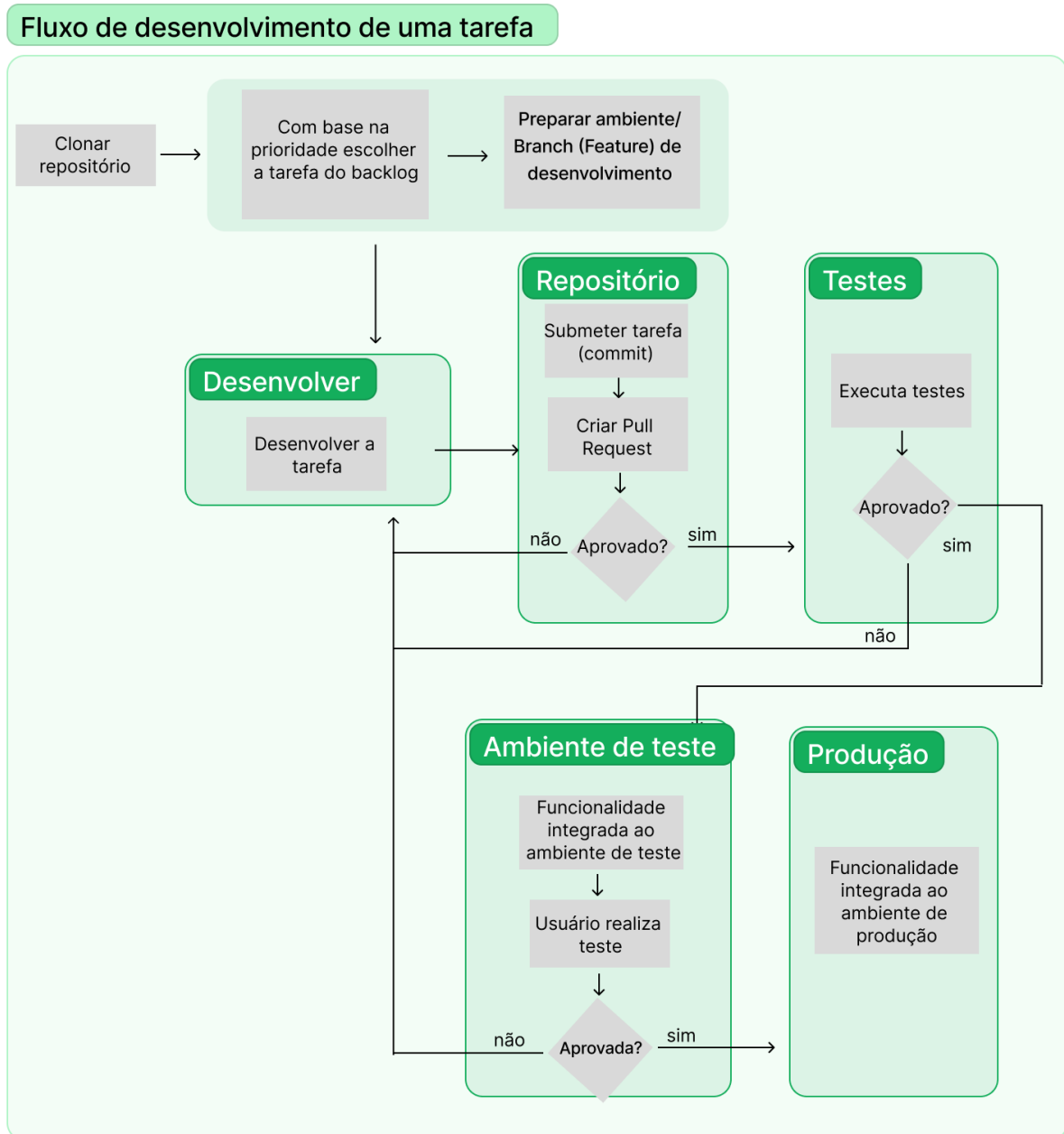


Figura 12 – Fluxo de desenvolvimento

Fonte: Autoria própria (2023).

No ambiente de testes, se o *feedback* do usuário for positivo, a tarefa irá para o ambiente de produção, e em caso de *feedback* negativo, retornará para a fase de desenvolvimento para correção das inconsistências encontradas durante os testes. Após integrada em ambiente de produção em caso de *feedback* de usuários, é criada uma tarefa com demanda urgente para a correção do problema encontrado em produção.

Para o desenvolvimento das tarefas, foi utilizado o *Git Workflow*, “essa técnica” agrega diversos benefícios a codificação, tais como, organização, controle e versionamento de código, proporcionando a rastreabilidade, integração (*Continuous Integration* (CI)) e entrega (*Continuous Delivery* (CD)) de código. Existem alguns tipos de *Workflow*, como por exemplo, *Feature*

Gitlab Flow para o Projeto

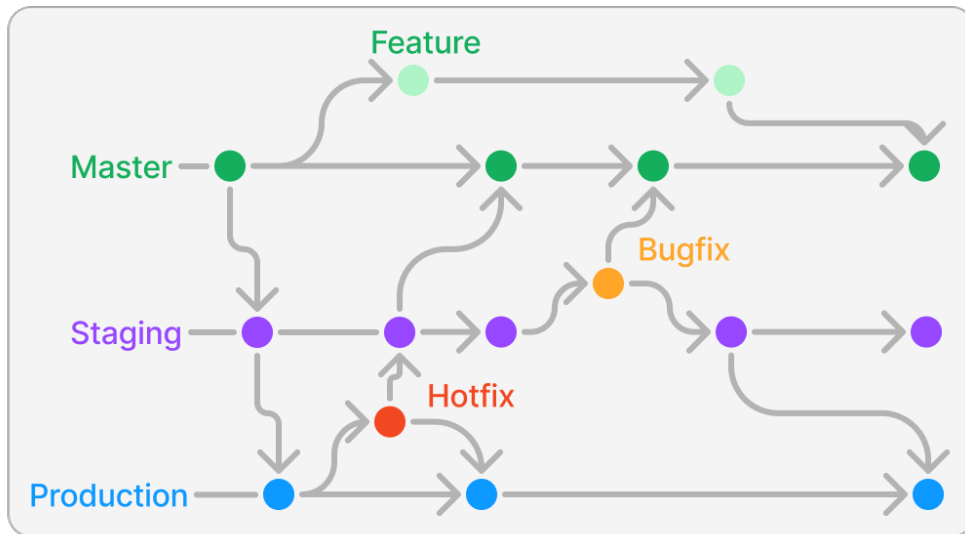


Figura 13 – Git Flow

Fonte: Autoria própria (2023), com base no GitWorkflow.

Branch, *Forking*, *GitLab Flow*, entre outros. Neste projeto foi utilizado o *GitLab Flow*, como pode ser verificado na Figura 13, onde o código é dividido em *branches*, permitindo rastrear possíveis problemas. Suas *branches* são divididas em:

Principais

- **Master/Main** é a de desenvolvimento;
- **Production** que é responsável por armazenar os códigos do sistema que está em produção;
- **Staging** armazenará os códigos disponíveis no ambiente de teste, e ao finalizada é incrementada a *branch* de Produção.

Temporárias

- **Feature** criada para desenvolver novas funcionalidades, que serão incrementadas a *Main*;
- **Hotfix** é gerada da *Production* quando existem problemas urgentes para corrigir é incrementada na *Production* e *Staging* após finalizada;
- **Bugfix** é gerada da *Staging* quando existem necessidades de correções e também é incrementada na *Main*.

Foram aplicadas integrações e entregas contínuas com uso da prática *Continuous Integration/Continuous Delivery* (CI/CD), que conforme ilustrado na Figura 14 , consiste em desenvolver, testar, integrar e implantar códigos de maneira contínua. Essa prática permite desenvolver de maneira ágil e de qualidade, pois no ciclo de desenvolvimento seus testes são

automatizados, ou seja, utiliza testes pré definidos. Quando aprovados, passam por *deploy* de forma automática para o ambiente de teste. Haverá dois ambientes funcionando, o ambiente de *Staging* que é restrito apenas para os usuários que forem realizar os testes, e o ambiente de Produção, que fica disponível para acesso ao público em geral.

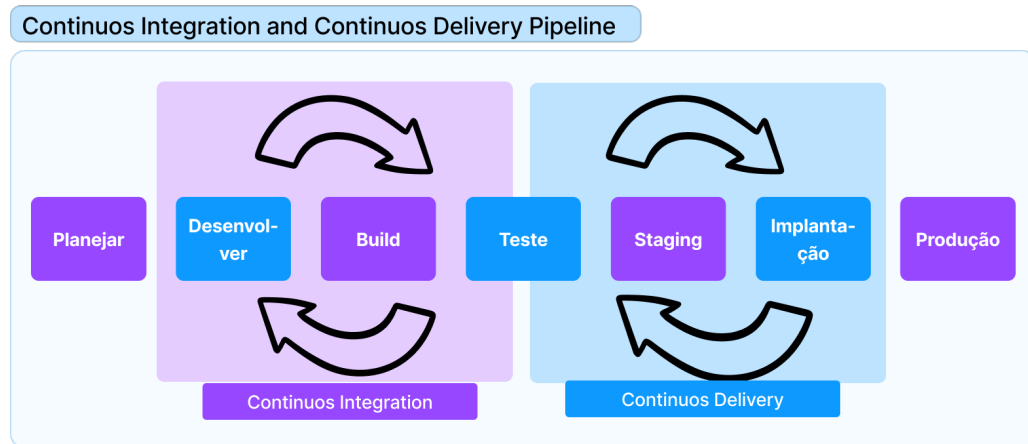


Figura 14 – CI/CD

Fonte: Autoria própria (2023).

Na Figura 14 o CI/CD foram separados em duas cores para diferenciar os seus ciclos, onde o CI ocorre o planejamento, desenvolvimento, *build* e testes, ou seja a parte de desenvolver a tarefa como um todo. Já no CD ocorre o ciclo de entrega, ou seja, são aplicados novamente os testes, em seguida enviado as modificações para o ambiente de testes(*stanging*), restrito para algumas pessoas testarem novas funcionalidades. Por fim é realizado o *deploy* para o ambiente de produção, disponível ao público geral.

4 ANÁLISE E PROJETO

Neste capítulo serão apresentados os resultados que foram obtidos em cada etapa do processo de desenvolvimento deste trabalho.

4.1 Requisitos

A primeira ação realizada no desenvolvimento deste trabalho foi o levantamento dos requisitos necessários para o sistema, conforme descritos abaixo.

4.1.1 Requisitos funcionais

- a. O sistema deve permitir a criação de OA's;
- b. Os OA's devem possuir título (nome), descrição e opcionalmente podem conter uma imagem;
- c. O sistema deve permitir escolher se os OA's serão ou não compartilhados;
- d. O sistema deve permitir criar uma ou mais introduções, em texto formatado, para cada OA, com suporte para adicionar vídeos, imagens e gráficos;
- e. As introduções devem possuir título e conteúdo;
- f. O sistema deve permitir escolher se as introduções serão ou não disponíveis para a visualização;
- g. O sistema deve permitir criar um ou mais exercícios para cada OA. Permite-se adicionar em cada exercício textos, imagens e gráficos;
- h. Os exercícios devem possuir título e conteúdo;
- i. O sistema deve permitir escolher se os exercícios serão ou não disponíveis para a visualização;
- j. O sistema deve permitir criar um ou mais passos de solução (questões) para cada exercício, em formato de textos, imagens ou gráficos;
- k. Os passos de solução devem possuir título e conteúdo a resposta correta e quantas casas decimais são necessárias para avaliar a resposta;
- l. O sistema deve permitir configurar a ordem de exibição dos passos, como por exemplo, todos de uma vez, de dois em dois, um por vez, dentre outras maneiras;

- m. O sistema deve permitir escolher se os passos serão ou não disponíveis para a visualização;
- n. O sistema deve permitir criar uma ou mais dicas para cada questão;
- o. As dicas devem possuir conteúdo;
- p. O sistema deve permitir configurar a ordem de exibição das dicas, como por exemplo, em sequência, de duas em duas, aleatório, todas de uma vez, dentre outras maneiras;
- q. O sistema deve permitir editar e excluir OA's de forma precisa, garantindo que todas as alterações sejam aplicadas corretamente;
- r. O sistema deve permitir duplicar os OA's completamente de forma eficiente e precisa, garantindo que todas as informações sejam duplicadas corretamente;
- s. O sistema deve permitir duplicar as introduções dos OA's, de forma eficiente e precisa, garantindo que todas as informações sejam duplicadas corretamente;
- t. O sistema deve permitir duplicar os exercícios dos OA's, de forma eficiente e precisa, garantindo que todas as informações sejam duplicadas corretamente;
- u. O sistema deve permitir duplicar os passos de solução dos exercícios, de forma eficiente e precisa, garantindo que todas as informações sejam duplicadas corretamente;
- v. O sistema deve permitir duplicar as dicas, de forma eficiente e precisa, garantindo que todas as informações sejam duplicadas corretamente.

4.1.2 Requisitos não funcionais

- a. O sistema deve possuir uma interface atrativa, possuindo mais cores e dispondo também de *layouts* mais harmônicos, pois a versão atual do sistema é monocromática e com *layout* muito quadrado;
- b. O sistema deve ter uma navegação intuitiva, onde não precise necessariamente seguir uma sequência de etapas para chegar ao destino desejado;
- c. As informações do sistema devem ser apresentadas de maneira agradável, evitando repetições e trazendo as mesmas de modo intuitivo.

4.2 Escolha do framework CSS

Para a escolha de *Framework* CSS aplicado foram realizadas reuniões com o Orientador e Coorientador deste trabalho, levando em consideração alguns itens: como popularidade do

Framework entre os desenvolvedores, qual *Framework* se adequaria melhor a FARMA, recursos visuais mais atrativos, dentre outros, conforme a Tabela 1.

Ao todo, 5 (cinco) *Framework* foram escolhidos com intuito de debate e realização da escolha, sendo eles o *Bootstrap*¹, o *Tailwind*², o *Materialize*³, o *Semantic*⁴ e o *Tabler IO*⁵.

O *Bootstrap* é um *Framework* muito conhecido e utilizado em diversos sites e sistemas, como por exemplo no *Twitter*, dentre os escolhidos para comparação é o que mais possui estrelas no *GitHub* e dispõe de diversos recursos personalizáveis. O *Tailwind* por sua vez é um *Framework* de código aberto que ao contrário do *Bootstrap* e demais, que tem classes pré definidas, cria um arquivo com classes base que serão utilizadas mescladas para obter a personalização desejada e também possui muitas estrelas no *GitHub*. O *Materialize* já é utilizado na FARMA e conta com diversos recursos assim como os demais e segue o padrão *Google's Material Design*. O *Semantic* tem como diferencial a possibilidade de criar *layouts* sofisticados e criar animações 3D, é um *Framework* com semântica intuitiva e contém recurso para depuração. E por fim, o *Tabler IO*, que é um *template* com base no *Bootstrap*, utilizado na FARMA, é totalmente responsivo, conta com modo escuro e outros recursos.

Tabela 1 – Tabela de frameworks CSS

	<i>Bootstrap</i>	<i>Tailwind</i>	<i>Materialize</i>	<i>Semantic</i>	<i>Tabler IO</i>
Conhecimento Prévio	5	4	3	1	1
Popularidade	5	4	3	1	1
<i>Stars</i> no <i>GitHub</i>	5	4	1	3	1
Usado na FARMA	1	1	5	1	5
Visual Atrativo	5	4	2	3	1
Média	4,2	3,4	2,8	1,8	1,8

Fonte: Autoria própria (2023).

Analisando todas as ponderações, é possível chegar à conclusão final de utilizar o *Bootstrap*, pois o mesmo contém os recursos esperados, adequa-se melhor as necessidades, e como pode-se observar na Tabela 1, obteve melhores resultados e sua média foi a mais elevada na comparação com os demais.

4.3 Ambiente de desenvolvimento

4.3.1 Configuração da API

Para a aplicação configurou-se uma imagem no *Docker* com o uso do *Dockerfile*, que é utilizada para levantar um container, incluindo as dependências necessárias para executá-la.

¹ <https://getbootstrap.com/>

² <https://tailwindcss.com/>

³ <https://materializecss.com/>

⁴ <https://semantic-ui.com/>

⁵ <https://tabler.io/>

Contém a imagem do *Ruby* em sua última versão (3.2.2) e algumas configurações adicionais responsáveis por criar a aplicação em *Rails* a partir dessa imagem. O *Docker Compose* foi configurado para orquestrar os containers, que inicialmente são: o container da API, container do banco de dados e o container de testes da API.

Além disso, foi configurado o *PostgreSQL* como gerenciador de banco de dados. Com isso é necessário configurar o usuário e senha do *PostgreSQL* e sua localização para que o *Rails* consiga acessar, isso será feito no arquivo *credentials* do *Rails*, editando-o tanto no ambiente de desenvolvimento quanto no ambiente de testes. O *credentials* é criptografado e não fica visível já que vai para o repositório, permitindo visualizar as configurações apenas por meio da chave, visto que contém informações que não podem ficar públicas, como por exemplo senha do banco de dados, chave da *Amazon*, *Rails Secret Key*, dentre outras.

Com o intuito de realizar um teste, foi criada uma rota na API que lista alguns membros da FARMA, que será retornada em formato JSON apenas com finalidade de exemplificar o funcionamento da API.

Posteriormente é feita a configuração dos testes, que serão executados pelo CI. Para isso será realizado o *build* do container de testes que já foi configurado no *Docker Compose* e a imagem que foi configurada no *Dockerfile*, com isso é instalado as dependências dos testes e em seguida executa-os, neste primeiro momento todos serão bem sucedidos, pois o teste é para exemplo então só vai tentar acessar a rota e verificará o *status*.

Em seguida foi configurado no *GitHub Actions* o *build* da aplicação, a execução dos testes e a execução dos verificadores de código, para garantir qualidade e segurança. Para isso foi criada uma organização, visto que existirá mais de um repositório relacionado a FARMA, sendo o primeiro, o da API.

O CI executará quando chegar uma PR para a *main*, quando a PR for atualizada ou quando ela for reaberta. No CI existe os *jobs*, que são uma tarefa específica responsável por executar atividades configuradas previamente, essas configurações são definidas para cada *job* com suas particularidades. Nesta aplicação existirá dois *jobs*.

O primeiro é o *job* de teste, configurado em uma máquina *Ubuntu* na última versão e dentro dele existe um container com a imagem do *Ruby* 3.2.2 (última versão). É necessário também um serviço de banco de dados, que é um container de banco igual o que foi feito no *Docker Compose*, todavia com algumas configurações diferentes por causa do *GitHub Actions*. Em seguida foram definidos os passos (*steps* de execução) que são basicamente baixar a aplicação já atualizada após o *merge* com a *main* e instalar as dependências, fazendo um *cache* para que nas próximas execuções do CI não necessite baixar novamente e seja mais ágil. Só então são executados os testes, sendo necessário acessar as configurações de banco de dados através das *credentials* do *rails*, para isso não pode ser colocado direto no projeto as *credentials*, então é utilizado o *Rails Master Key* que é a chave para acessar essas *credentials*, sendo configurada no *GitHub* como uma variável secreta, colocando nessa configuração a chave que

abre essas credenciais para acessar as configurações. Em seguida prepara o banco de testes e executa-os.

O segundo *job* é utilizado para verificar a qualidade do código, ele é bem semelhante ao anterior, também executará em uma máquina *Ubuntu* na última versão onde irá baixar a aplicação, configurar o *Ruby*, instalar as dependências necessárias. Neste *job* são executados 3 verificadores de código que serão:

- ***RuboCop***, um analisador de código estático *Ruby* e formatador de código. Além de relatar os problemas descobertos em seu código, o *RuboCop* também corrige automaticamente alguns erros;
- ***Brakeman***, uma ferramenta de análise estática que verifica os aplicativos *Ruby on Rails* em busca de vulnerabilidades de segurança;
- ***Rubycritic***, uma *gem* que envolve *gems* de análise estática como *Reek*, *Flay* e *Flog* para fornecer um relatório de qualidade do seu código *Ruby*.

Para o CD é necessário configurar um Servidor Virtual Privado, do inglês *Virtual Private Server* (VPS). Nesse foi instalado o *Docker* para gerenciar os containers e *Docker Compose* para orquestrar os diversos containers necessários, como, por exemplo, o container de aplicação e de banco de dados. Além disso, foi necessário instalar o servidor de aplicações *nginx* na VPS para ser utilizado como *proxy* reverso, o qual recebe as requisições e repassa para o container da aplicação. Para garantir acesso seguro a VPS, foi configurado o *Secure Shell Protocol* (SSH) por meio das chaves públicas e privadas.

Com a VPS configurada, foi definido os passos a serem realizado pelo *GitHub Actions* para a entrega contínua. Sendo que um *deploy* é realizado a cada nova atualização recebida na *branch main*, para não ter inconsistências, as a adição de novas funcionalidades e correções somente são aceitas por meio de PR. Quando uma PR é aceita na *branch main*, o processo de *deploy* se inicia, seguindo o seguinte processo:

1. Baixar versão atual da *branch main*;
2. Realizar o *build* da aplicação;
3. Executar os testes;
4. Executar os *linters*;

Se tudo passar com sucesso realiza o *deploy* por meio dos seguintes passos:

1. Cria uma imagem nova contendo a versão mais atual da aplicação;
2. Registra essa imagem no Container *Registry* do *GitHub*;

3. Loga no servidor via SSH usando a chave privada;
4. Baixa a imagem mais atual do Container *Registry*.
5. Atualiza a imagem em execução para mais atual.

Todo esse processo é automatizado usando *GitHub Actions* o que permite agilidade na entrega das novas funcionalidades para teste em um ambiente de *Staging*.

Com isso, a API foi configurada com sucesso. Na Figura 15 visualizamos o *Projects* do *GitHub*, que foi utilizado para gerenciar e descrever todas as tarefas que foram serem executadas neste projeto.

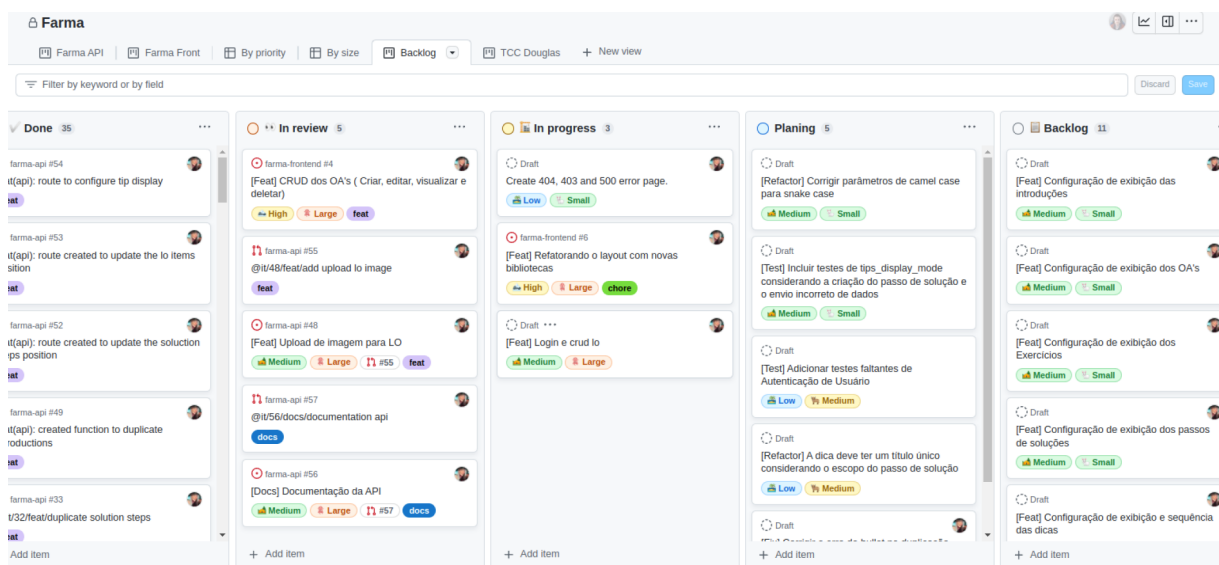


Figura 15 – Projects FARMA API

Fonte: Autoria própria (2023).

4.3.2 Exemplo de funcionamento da API

Segue abaixo um exemplo do fluxo da aplicação que acabamos de configurar. Na Figura 16 é possível visualizar como a API estava funcionando após ter sido configurada.

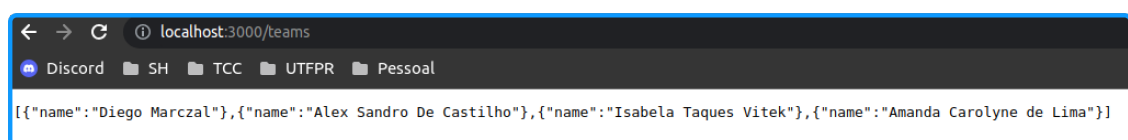


Figura 16 – Rota Index da API

Fonte: Autoria própria (2023).

Na Figura 17, visualiza-se os comandos utilizados para criar a *branch* para realizar as modificações, nomeada como '@it/14/chore/add_new_member_farma'. Nesta mesma figura, observa-se quais foram as modificações. Em seguida na Figura 18 exibe no site as modificações já aplicadas.

```

PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL  GITLENS  COMENTÁRIOS

● belavitek@belavitek:~/Desktop/FARMA/farma-api$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'upstream/main'.
● belavitek@belavitek:~/Desktop/FARMA/farma-api$ git pull
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 14 (delta 4), reused 3 (delta 2), pack-reused 0
Unpacking objects: 100% (14/14), 4.78 KiB | 2.39 MiB/s, done.
From https://github.com/Farma-Authoring-Tool/farma-api
   73d2196..fb3128b  main       -> upstream/main
   * [new tag]         published -> published
Updating 73d2196..fb3128b
Fast-forward
 .github/pull_request_template.md | 51 ++++++
 README.md                       | 25 +-----
 2 files changed, 63 insertions(+), 13 deletions(-)
 create mode 100644 .github/pull_request_template.md
● belavitek@belavitek:~/Desktop/FARMA/farma-api$ git checkout -b @it/14/chore/add_new_member_farma
Switched to a new branch '@it/14/chore/add_new_member_farma'
● belavitek@belavitek:~/Desktop/FARMA/farma-api$

```

Figura 17 – Branch de modificação

Fonte: Autoria própria (2023).

```

localhost:3000/teams
Discord SH TCC UTFPR Pessoal

[{"name":"Diego Marczal"}, {"name":"Alex Sandro De Castilho"}, {"name":"Isabela Taques Vitek"}, {"name":"Amanda Carolyne de Lima"}, {"name":"Denis Lucas Silva"}]

```

Figura 18 – Rota Index da API com a modificação

Fonte: Autoria própria (2023).

Com as modificações funcionando da maneira esperada é realizado o *commit* de todos os arquivos que foram modificados, conforme a Figura 19.

```

PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL  GITLENS  COMENTÁRIOS

● belavitek@belavitek:~/Desktop/FARMA/farma-api$ git status
No ramo @it/14/chore/add_new_member_farma
Changes not staged for commit:
  (utilize "git add <arquivo>..." para atualizar o que será submetido)
  (use "git restore <file>..." to discard changes in working directory)
   modified:   app/controllers/teams_controller.rb

nenhuma modificação adicionada à submissão (utilize "git add" e/ou "git commit -a")
● belavitek@belavitek:~/Desktop/FARMA/farma-api$ git add .
● belavitek@belavitek:~/Desktop/FARMA/farma-api$ git commit -m "chore(api): add new member to FARMA"
[@it/14/chore/add_new_member_farma 103150f] chore(api): add new member to FARMA
 1 file changed, 2 insertions(+), 1 deletion(-)
● belavitek@belavitek:~/Desktop/FARMA/farma-api$ git push -u origin @it/14/chore/add_new_member_farma
Enumerating objects: 46, done.
Counting objects: 100% (46/46), done.
Delta compression using up to 12 threads
Compressing objects: 100% (35/35), done.
Writing objects: 100% (37/37), 10.70 KiB | 10.71 MiB/s, done.
Total 37 (delta 14), reused 2 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (14/14), completed with 4 local objects.
remote:
remote: Create a pull request for '@it/14/chore/add_new_member_farma' on GitHub by visiting:
remote:   https://github.com/isabela-tvitek/farma-api/pull/new/@it/14/chore/add_new_member_farma
remote:
To https://github.com/isabela-tvitek/farma-api.git
 * [new branch]      @it/14/chore/add_new_member_farma -> @it/14/chore/add_new_member_farma
branch '@it/14/chore/add_new_member_farma' set up to track 'origin/@it/14/chore/add_new_member_farma'.
● belavitek@belavitek:~/Desktop/FARMA/farma-api$

```

Figura 19 – Commit das modificações

Fonte: Autoria própria (2023).

Após realizar o *commit* foram executados os testes conforme a Figura 20 e Figura 21, onde possui um exemplo, no primeiro teste executado, de uma falha que necessita de correção, exibindo em seguida o teste bem sucedido assim como os demais que foram executados na sequência.

```

1 class TeamsController < ApplicationController
2   def index
3     teams = [
4       { name: 'Diego Marczal' },
5       { name: 'Alex Sandro De Castilho' },
6       { name: 'Isabela Taques Vitek' },
7       { name: 'Amanda Carolyne de Lima' },
8       { name: 'Denis Lucas Silva' },
9     ]
10
11     render json: teams
12   end
13 end
14

```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL GITLENS COMENTÁRIOS

belavitek@belavitek:~/Desktop/FARMA/farma-api\$./run linters
Inspecting 25 files
.....C.....

Offenses:

app/controllers/teams_controller.rb:8:36: C: [Correctable] Style/TrailingCommaInArrayLiteral: Avoid comma after the last item of an array.
{ name: 'Denis Lucas Silva' },
^

25 files inspected, 1 offense detected, 1 offense autocorrectable

Task completed in 0m2,537s
belavitek@belavitek:~/Desktop/FARMA/farma-api\$

Figura 20 – Teste *linters* - exemplo de uma falha

Fonte: Autoria própria (2023).

```

1 class TeamsController < ApplicationController
2   def index
3     teams = [
4       { name: 'Diego Marczal' },
5       { name: 'Alex Sandro De Castilho' },
6       { name: 'Isabela Taques Vitek' },
7       { name: 'Amanda Carolyne de Lima' },
8       { name: 'Denis Lucas Silva' }
9     ]
10
11     render json: teams
12   end
13 end
14

```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL GITLENS COMENTÁRIOS

Controllers: 2
Models: 1
Templates: 1
Errors: 0
Security Warnings: 0

== Warning Types ==

No warnings found

RubyCritic can provide more feedback if you use a Git, Mercurial or Perforce repository. Churn will not be calculated.
running flay smells
running flog smells
.....
running reek smells
.....
running complexity
.....
running attributes
.....
running churn
.....
running simple_cov
.....

New critique at file:///var/www/farma/tmp/rubycritic/overview.html
Score: 99.87

Task completed in 0m4,376s
belavitek@belavitek:~/Desktop/FARMA/farma-api\$

Figura 21 – Teste bem sucedido

Fonte: Autoria própria (2023).

Com os testes bem sucedidos, a *branch* foi enviada para o *GitHub* e vai ser criada uma PR para ela, conforme a Figura 22, que será submetida para revisão e para os testes de CI previamente configurados, ilustrados na Figura 23.

chore(api): add new member to FARMA

Write Preview

Motivation and Context

<!-- Why is this change required? What problem does it solve? -->
<!-- If it fixes an open issue, please link to the issue here. -->

How has this been tested?

has been tested using the already defined linters tests

Screenshots (if appropriate):

Types of changes

- [] fix: A bug fix
- [] feat: A new feature
- [] refactor: A code change that neither fixes a bug nor adds a feature
- [] test: Adding missing tests or correcting existing tests
- [] perf: A code change that improves performance
- [] docs: Documentation only changes
- [] style: Changes that do not affect the meaning of the code (e.g. prettier format)
- [] ci: Changes to our CI configuration files and scripts
- [X] chore: Changes to the build process or auxiliary tools and libraries functionality to not work as expected

Checklist:

<!-- Go over all the following points, and put an 'x' in all the boxes that apply. -->
<!-- If you're unsure about any of these, don't hesitate to ask. We're here to help! -->

- [X] My code follows the code style of this project.
- [] My change requires a change to the documentation.
- [] I have updated the documentation accordingly.

Hours Required:

Attach files by dragging & dropping, selecting or pasting them.

☒ Allow edits and access to secrets by maintainers

Create pull request

Reviewers

MarczaTSIGP

At least 1 approving review is required to merge this pull request.

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Use Closing keywords in the description to automatically close issues

Helpful resources

GitHub Community Guidelines

Figura 22 – Criação da PR
Fonte: Autoria própria (2023).

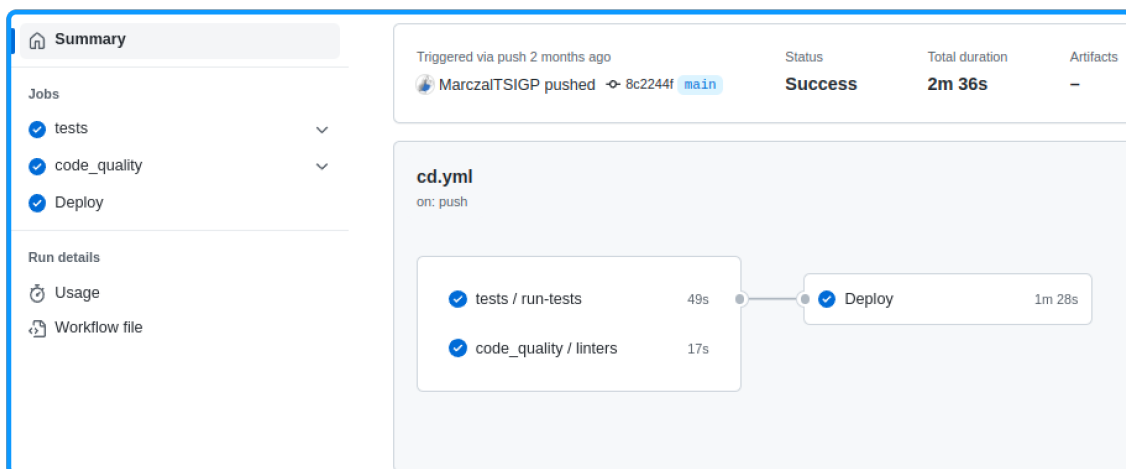


Figura 23 – Execução do CI
Fonte: Autoria própria (2023).

Em seguida, o revisor deverá aprovar esta PR conforme é solicitado na Figura 24, ele pode solicitar mudanças e correções adicionando comentários sobre o que deve ser alterado, conforme a Figura 25. Com todas as modificações corretas nesta PR é realizada a aprovação conforme a Figura 26 e já está pronta para o *merge*.

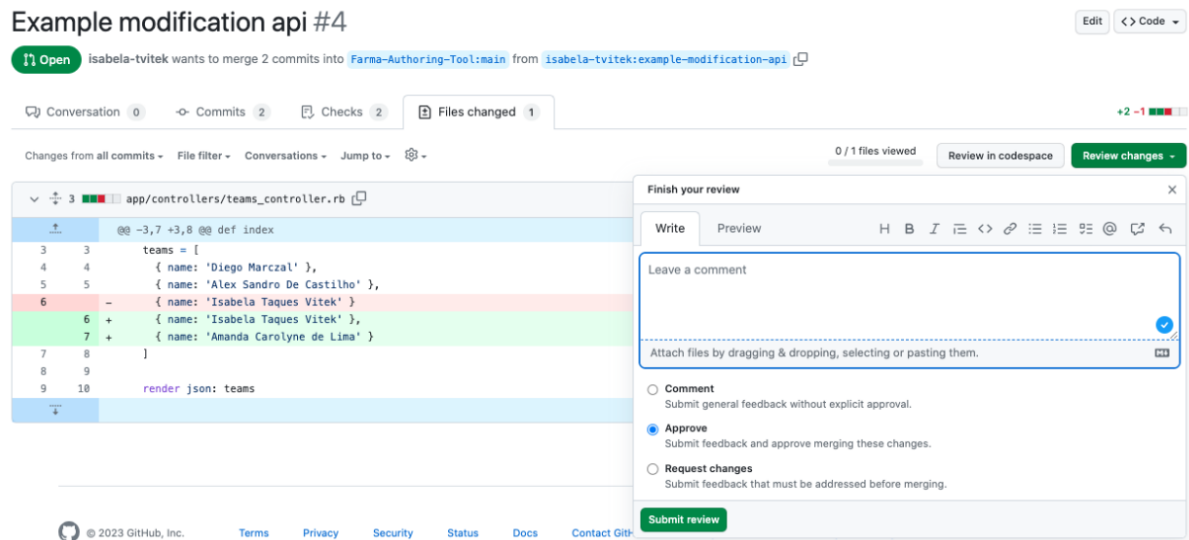


Figura 24 – Aprovação da PR
Fonte: Autoria própria (2023).

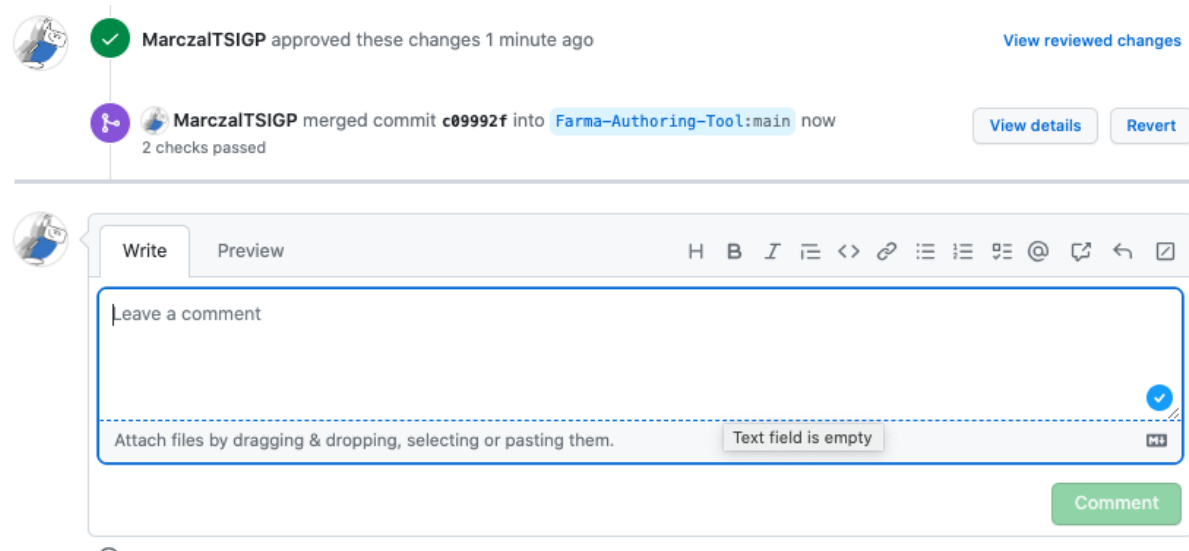




Figura 25 – Comentários na PR
Fonte: Autoria própria (2023).



✓

MarczaITSIGP approved these changes now

[View reviewed changes](#)



✓

Changes approved

1 approving review by reviewers with write access. [Learn more.](#)

✓

1 approval

▼

✓

All checks have passed

2 successful checks

✓

CI / tests (pull_request_target)

Successful in 1m

[Details](#)

✓

CI / code_quality (pull_request_target)

Successful in 30s

[Details](#)

✓

This branch has no conflicts with the base branch

Merging can be performed automatically.

Squash and merge

▼

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Figura 26 – PR aprovada
Fonte: Autoria própria (2023).

5 DESENVOLVIMENTO DO SISTEMA

Neste capítulo será abordado cada tarefa realizada no desenvolvimento, bem como a explicação da codificação e os problemas encontrados durante o processo. Vale ressaltar que todo o desenvolvimento do sistema iniciou pela codificação da API e posteriormente o desenvolvimento de telas para acesso das funcionalidades implementadas.

5.1 Procedimentos Iniciais

Como parte do processo inicial, foram criados os *README's* que fornecem informações sobre o projeto no repositório. Além disso, no *Wiki* do projeto, foi elaborada uma documentação completa fornecendo aos colaboradores informações detalhadas sobre como baixar o projeto, configurações gerais, como contribuir com o mesmo, além de fornecer informações sobre o propósito e as funcionalidades da FARMA no geral.

Para garantir uma implementação eficiente, foi adotada uma abordagem organizada na definição e divisão de tarefas, quebrando-as em pequenos fragmentos, tanto para o desenvolvimento da API, quanto para a interface do *frontend*. Levando em consideração a qualidade do trabalho como um todo, foram definidos padrões de processos, incluindo a criação de modelos para PRs, padrões para mensagens de *commit's* e nomes de *branchs*. Essa padronização garante que cada contribuição e alteração no projeto siga um método pré-definido, facilitando a colaboração e a revisão do código.

5.2 FARMA API

Para o desenvolvimento da API, foi criado um diagrama relacional (Figura 27) com o objetivo de definir as tabelas, seus parâmetros e os relacionamentos entre elas. Nessa fase do projeto, surgiram desafios relacionados ao entendimento dos parâmetros necessários para a criação de cada tabela e à complexidade das conexões entre elas, o que demandou orientação do professor. Como resultado, foram estabelecidas com sucesso seis tabelas com os campos necessários para garantir o funcionamento adequado do projeto.

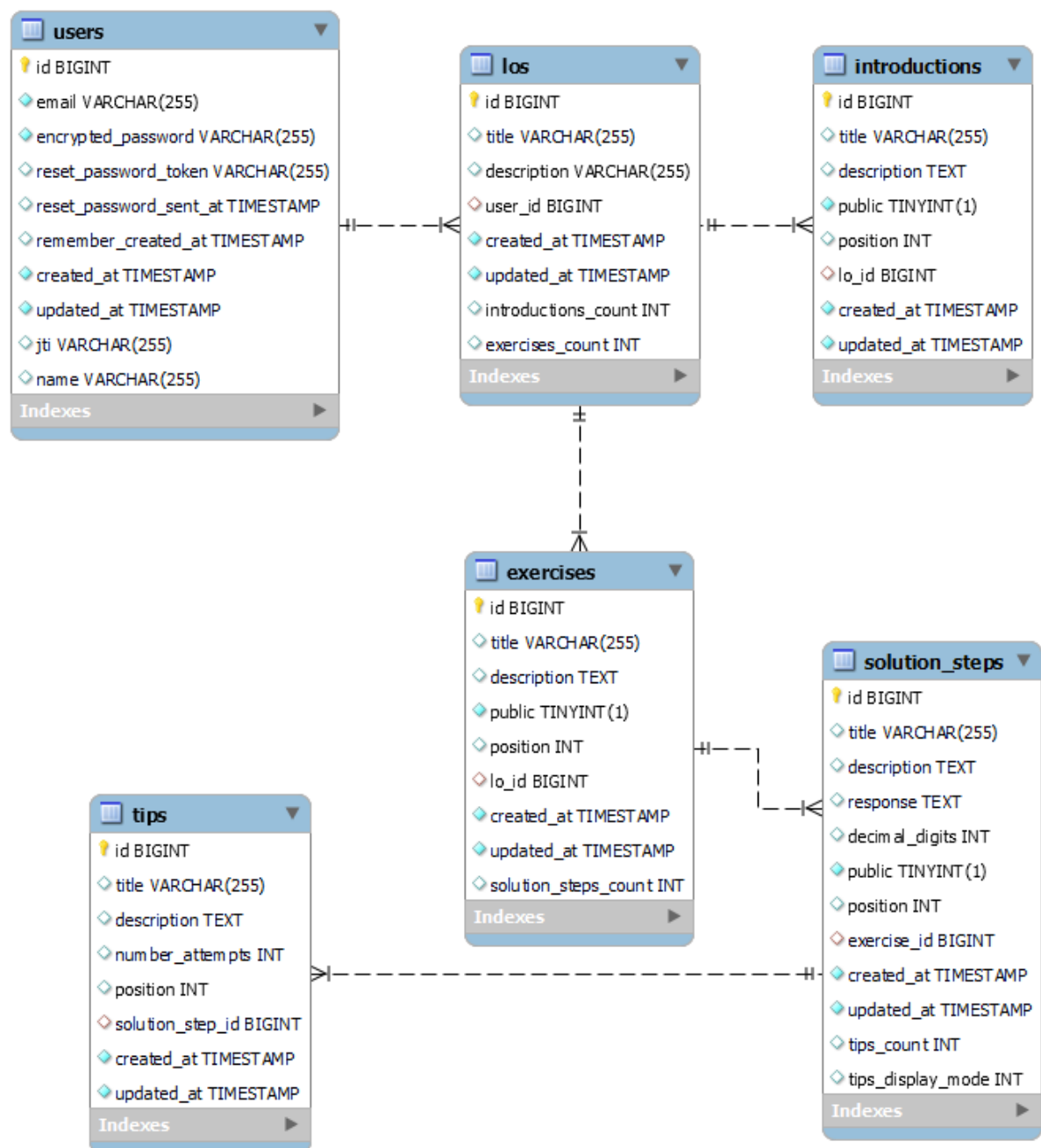


Figura 27 – Diagrama Entidade Relacionamento

Fonte: Autoria própria (2023).

5.2.1 Desenvolvimento do CRUD de OA

O OA é a parte principal do sistema, pois envolve a criação dos itens necessários para seu funcionamento, contendo introduções e exercícios, que são compostos por passos de soluções e dicas para resolução das tarefas. Para o gerenciamento de OA's foi necessário criar *migrations* que populam a estrutura do banco de dados, bem como criar o *controller* com as ações principais, sendo elas visualização, criação, edição e exclusão de OA's.

A atividade principal consistiu no desenvolvimento do CRUD de OA's, iniciando a API. Além disso foi preciso criar um *Model* para representar um OA, contendo as validações de parâmetros, sendo a necessidade de título e descrição não serem nulos e títulos serem únicos. Posterior a isso foram definidas as rotas principais e também configurado o *CORS* para que fosse possível realizar as requisições do *frontend* para a API. O *CORS* é uma segurança utilizada pelos navegadores para proteção de solicitações não autorizadas.

Posterior a isso foi iniciado o desenvolvimento dos testes referente a cada ação do CRUD, além de testes de *models*.

Nesta tarefa, as dificuldades iniciais surgiram ao entender o projeto e compreender o funcionamento de cada componente, bem como a abordagem correta a ser seguida para o funcionamento correto. Após isso, o desenvolvimento dos testes para cada ação do CRUD e os testes dos modelos foram iniciados. Durante essa fase, os primeiros testes criados não estavam abrangendo adequadamente a aplicação, uma vez que um único arquivo foi inicialmente destinado a todo o CRUD, resultando em retornos simplificados que não validavam, por exemplo, se um item havia sido criado, se os parâmetros estavam corretos, entre outros aspectos. Além disso, houve uma reorganização da estrutura de pastas e das rotas para melhor organização e entendimento do projeto. Após a conclusão do desenvolvimento do CRUD, a documentação das rotas foi criada, utilizando o *Postman* como ferramenta. *Postman* é usado para testar requisições, além de servir como documentação da API.

Durante essa atividade, uma refatoração geral foi conduzida pelo professor, estabelecendo um padrão de desenvolvimento a ser seguido. Este ajuste foi necessário, pois, no início, houve dificuldades em determinar a abordagem correta. O CRUD que foi inicialmente implementado estava incompleto, com *controllers* que retornavam informações incorretas ou que não incluíam detalhes, como mensagens de sucesso ou erro, bem como o *status* da requisição. Ao final, todos os testes de funcionalidades e de qualidade do código foram executados para garantir que os ajustes estavam funcionando corretamente como deveriam estar. Foi obtido sucesso em ambos os tipos de testes, o que permitiu a realização da PR no *GitHub*, a qual foi aprovada pelo professor.

5.2.2 Desenvolvimento do CRUD de introduções

Esta atividade consistiu no desenvolvimento do CRUD de Introduções, as quais são criadas para cada OA. Para isso, foi fundamental a criação de *migrations* necessárias para a criação da tabela de Introduções no Banco de Dados, bem como criado o *controller* com as ações principais das Introduções: visualização, criação, edição e exclusão. Além disso, foi criado o *models*, utilizados para representar uma Introdução, contendo as validações de parâmetros, sendo eles: a necessidade de título e descrição não serem nulos, títulos serem únicos, garantir que o parâmetro *public* será um *boolean*, configurar o relacionamento com OA tanto no *model* de OA quanto no de Introdução, e também já definir automaticamente o valor que a posição

receberá inicialmente. Além disso foi criado rotas no sistema para que fosse possível gerenciar este novo CRUD.

Após a finalização do CRUD, foi iniciada a criação da documentação das rotas, para definir o padrão de como devem ser enviados os parâmetros, e como será utilizado cada rota, exemplificando como são os retornos esperados.

Com relação aos testes, foi necessário criar testes referentes a cada ação do CRUD, bem como para a *model*. O desenvolvimento em si dos testes não foi dificultoso, por seguir a mesma ideia da atividade anterior. Porém, no geral, foi necessário entender melhor como seria o relacionamento entre a Introdução e o OA, fazendo com que a Introdução dependa dele para existir.

Nesta atividade em específico, foi necessária algumas correções até ela ser aceita por completo no repositório. O primeiro ajuste se deu em reformular alguns trechos de código, como apresentado em um exemplo na Figura 28.

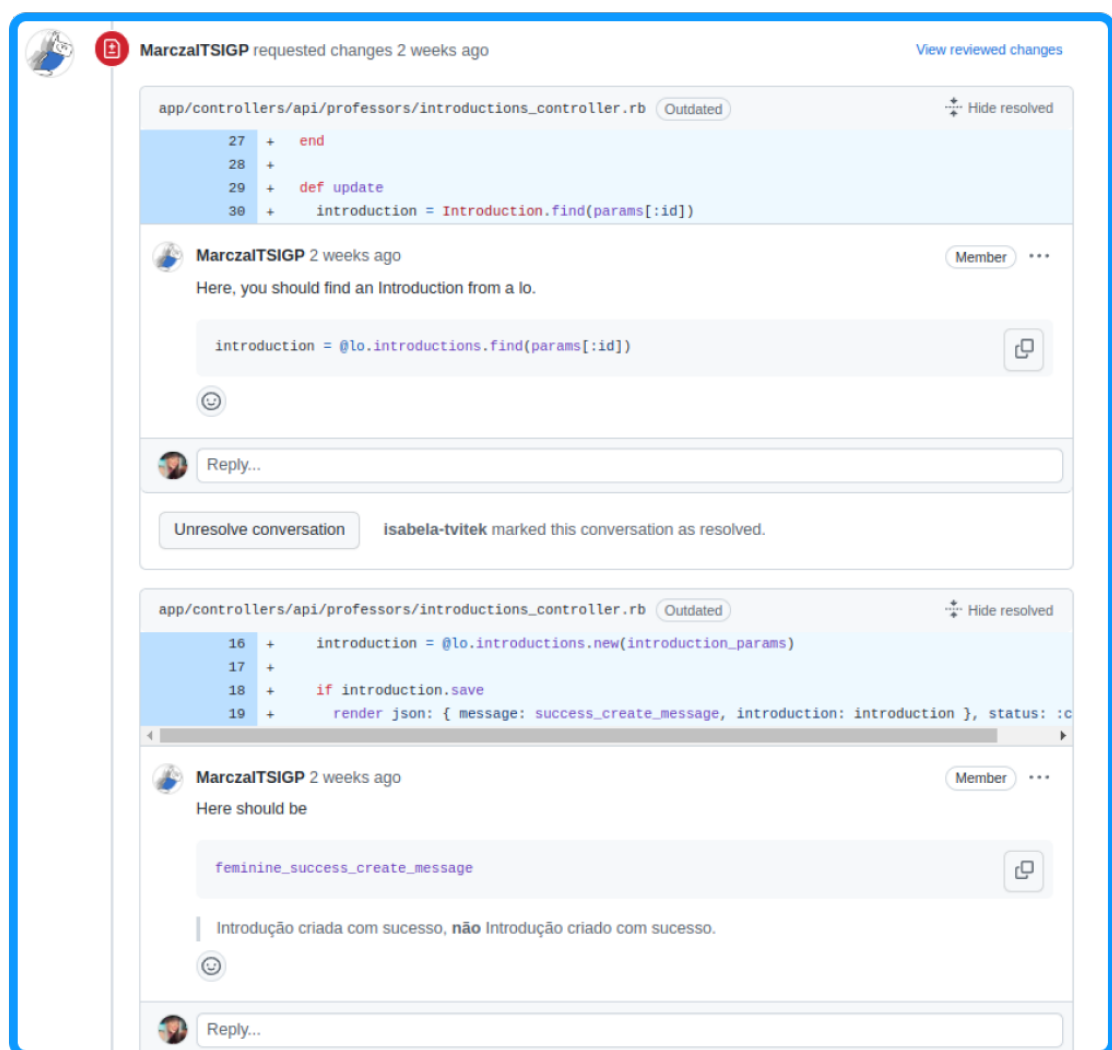


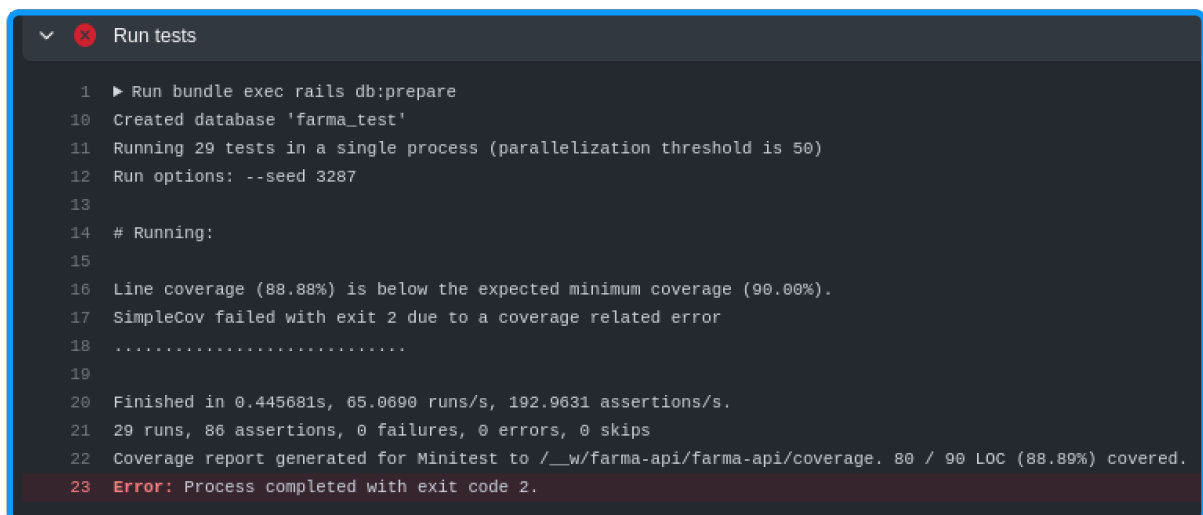
Figura 28 – Correção requisitada

Fonte: Autoria própria (2023).

Outro ajuste necessário foi nos testes, pois estavam passando parâmetros de *public*, *position* e *lo-id* desnecessários, já que tem valor padrão, tanto em *create* quanto em *update*. Ainda com relação aos testes, foi solicitado incluir teste de caso de falha ao tentar deletar introduções de outro OA, bem como realizado o ajuste no *controller* para que isso não fosse permitido.

5.2.3 Desenvolvimento do CRUD de exercícios

Nesta tarefa foi desenvolvido o CRUD de Exercícios, permitindo a criação de exercícios e seu respectivo conteúdo no sistema, oferecendo uma prévia do que será abordado em cada questão. Inicialmente, o CRUD foi criado sem a inclusão de um título, porém, após uma discussão com o professor sobre a necessidade de um título, ele foi adicionado. O *design* do CRUD de Exercícios foi modelado com base no CRUD de Introduções realizado anteriormente, e, à medida que ajustes foram necessários no CRUD de Introduções, também aplicamos alterações no CRUD de Exercícios para seguir um padrão de desenvolvimento. Um problema adicional encontrado no CRUD de Exercícios foi a falta de alguns testes, resultando em uma cobertura de teste inferior a 90%, como é possível visualizar na Figura 29. Isso causou erros ao executar a integração contínua CI no processo de revisão de código. Para verificar onde os testes estavam mais baixos e corrigir, foi utilizado do *coverage*, que é um arquivo que contém um relatório sobre a cobertura do código, conforme a Figura 30.

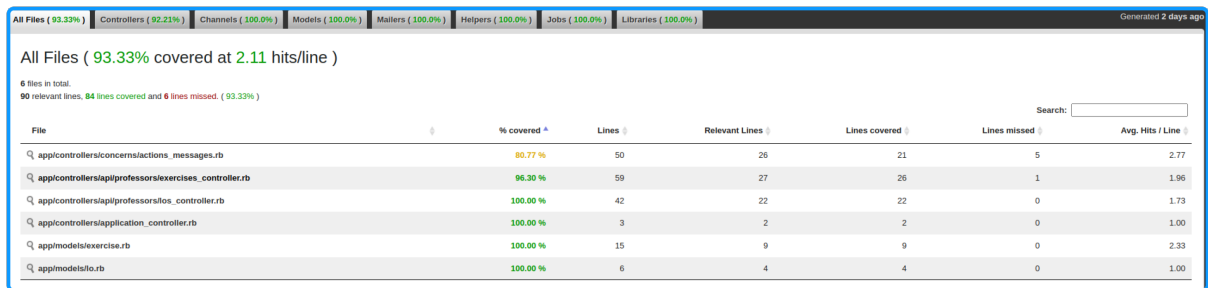


```

1  ▶ Run bundle exec rails db:prepare
10 Created database 'farma_test'
11 Running 29 tests in a single process (parallelization threshold is 50)
12 Run options: --seed 3287
13
14 # Running:
15
16 Line coverage (88.88%) is below the expected minimum coverage (90.00%).
17 SimpleCov failed with exit 2 due to a coverage related error
18 .....
19
20 Finished in 0.445681s, 65.0690 runs/s, 192.9631 assertions/s.
21 29 runs, 86 assertions, 0 failures, 0 errors, 0 skips
22 Coverage report generated for Minitest to /__w/farma-api/farma-api/coverage. 80 / 90 LOC (88.89%) covered.
23 Error: Process completed with exit code 2.
  
```

Figura 29 – Falha nos testes

Fonte: Autoria própria (2023).



All Files (93.33% covered at 2.11 hits/line)

6 files in total.
90 relevant lines, 84 lines covered and 6 lines missed. (93.33%)

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/concerns/actions_messages.rb	80.77 %	50	26	21	5	2.77
app/controllers/api/professors/exercises_controller.rb	96.30 %	59	27	26	1	1.96
app/controllers/api/professors/solutions_controller.rb	100.00 %	42	22	22	0	1.73
app/controllers/application_controller.rb	100.00 %	3	2	2	0	1.00
app/models/exercise.rb	100.00 %	15	9	9	0	2.33
app/models/solution.rb	100.00 %	6	4	4	0	1.00

Figura 30 – Painel coverage
Fonte: Autoria própria (2023).

5.2.4 Desenvolvimento do CRUD de passos de solução

Nesta tarefa foi implementado o CRUD para os Passos de Solução. Esta funcionalidade possibilita aos professores usuários da plataforma criar uma ou mais questões dentro dos exercícios. O objetivo é facilitar a elaboração de listas de exercícios tematicamente relacionados, como, por exemplo, um primeiro exercício composto por questões que envolvem apenas operações de multiplicação e um segundo exercício focado exclusivamente questões sobre divisões.

```
Running 1 tests in a single process (parallelization threshold is 50)
Run options: --seed 33742

# Running:

E

Error:
Api::Professors::SolutionStepsControllerIndexTest#test_should_return_all_solution_steps:
ActionController::UrlGenerationError: No route matches {:action=>"index", :controller=>"api/professors/solution_steps", :io_id=>"Exercise id: 10, title: "Introduction to simple equations" - #<class:Ox000...>.", description: "Learning about equations.", public: false, position: 1698949568, io_id: 10, created_at: "2023-11-02 15:26:08.855114000 -0300", updated_at: "2023-11-02 15:26:08.855114000 -0300">}, missing required keys: [:exercise_id]
test/controllers/api/professors/solution_step/index_test.rb:10:in 'block in <class:SolutionStepsControllerIndexTest>'

rails test test/controllers/api/professors/solution_step/index_test.rb:9

Finished in 0.087102s, 11.4807 runs/s, 0.0000 assertions/s.
1 runs, 0 assertions, 0 failures, 1 errors, 0 skips

Task completed in 0m0.987s
belavitek@belavitek:~/Desktop/FARMA/farma-api$
```

Figura 31 – Erro de rotas
Fonte: Autoria própria (2023).

Um dos principais desafios enfrentados na implementação deste CRUD esteve na fase de criação de testes, particularmente na necessidade de estabelecer referências corretas entre os Passos de Solução, os Exercícios e os OA. Na Figura 31, é possível observar um exemplo de erro ocasionado devido à dificuldade de relacionar a rota correta que devia ser testada uma vez que estava direcionando para “/api/professors/solution_steps” ao invés de “/api/professors/solutions/1/exercises/1/solution_steps”. Uma vez superada a dificuldade de configurar as referências de rotas com todos os parâmetros necessários, o desenvolvimento prosseguiu de maneira eficiente e sem maiores obstáculos.

5.2.5 Desenvolvimento do CRUD de dicas

Esta tarefa consistiu no desenvolvimento do CRUD especificamente para as Dicas. Esta parte do sistema permite aos professores criar uma ou mais dicas para cada questão. As dicas servem como auxílio aos alunos, orientando-os sobre como resolver determinadas questões. É possível gerar várias dicas para uma única questão. Em desenvolvimentos futuros do projeto, serão estabelecidas as configurações relativas ao momento e à maneira de exibição dessas dicas aos alunos.

5.2.6 Desenvolvimento da autenticação de usuários com *token*

Esta tarefa focou no desenvolvimento da funcionalidade responsável pela autenticação de usuários baseada em *tokens*, utilizando a *gem* “*devise-jwt*” do *Rails*. A escolha de desabilitar sessões e adotar a autenticação apenas via *token* foi motivada pelo desenvolvimento de uma API sem estado, além da versatilidade do JWT, explicado melhor na subseção 3.1.1, que suporta tanto aplicações Web quanto *Mobile*, alinhando-se aos planos futuros de expansão para aplicativos móveis.

Os desafios iniciais incluíram a compreensão da necessidade de desativar as sessões ao usar JWT e o entendimento de que não era necessário implementar manualmente toda a lógica de *login* e criação de usuários, graças às facilidades oferecidas pela estratégia de autenticação do “*devise-jwt*”.

Para a implementação de *tokens*, as *gems* “*devise*” e “*devise-jwt*” foram adicionadas ao Gemfile, seguidas pela instalação e configurações específicas. Essas configurações envolveram a definição de uma chave secreta e o ajuste do Devise para gerenciar a emissão e revogação de JWTs. Além disso, foi estabelecido um tempo de vida de trinta minutos para cada *token*.

A criação de um modelo *User* foi essencial para permitir a autenticação dos usuários e o gerenciamento dos *tokens* JWT. A configuração do CORS foi implementada para permitir solicitações seguras entre diferentes domínios, controlando o acesso a partir de origens específicas.

Dois novos controladores foram adicionados para gerenciar o registro, *login* e *logout* no sistema e também definido que seus retornos seriam em formato JSON, e as rotas foram reorganizadas para que as demais funcionalidades exigissem autenticação. As migrações necessárias foram criadas e ajustes adicionais no projeto foram realizados para assegurar o funcionamento conforme o esperado.

Durante o desenvolvimento, outro desafio enfrentado foi a necessidade de alterar a chave criptografada usada nos testes. Foi imprescindível incluir na chave de testes a mesma variável “*secret-key-base*” presente na chave criptografada de *development*. Sem essa inclusão, os testes não podiam ser executados.

Após as correções necessárias, foram criados testes para os novos controladores que gerenciam o *login*, *logout* e cadastro (*signup*). No entanto, surgiu um novo desafio relacionado

à cobertura de testes (*coverage*). Mesmo com os testes implementados para esses controladores, o sistema de cobertura indicava que nenhuma parte dos controladores havia sido testada. Esse cenário exigiu uma análise mais aprofundada para entender e resolver a discrepância entre a execução dos testes e a representação de sua cobertura pelo sistema de *coverage*. A solução desse problema ocorreu após revisão com o professor, onde foi constatado que o erro ocorria por falta de respostas padronizadas e classes que foram definidas e não estavam sendo utilizadas, a realização dos ajustes necessários levou à solução do problema.

5.2.7 Desenvolvimento da função para duplicar dicas

Durante o desenvolvimento dessa funcionalidade para permitir que o professor duplique qualquer dica, houve uma correção na lógica de duplicação das dicas. Inicialmente, a lógica estava no controlador, porém, foi identificado que essa lógica deveria estar no *Model*. Portanto, ajustes foram feitos para mover a lógica de duplicação para o modelo correto, enquanto apenas a função correspondente era chamada no *controller*. Depois desses ajustes, foram criados testes no modelo e no controlador para garantir a funcionalidade correta da duplicação.

Durante a fase de criação dos testes, foram encontradas dificuldades, pois ao tentar duplicar uma dica, ela não estava sendo encontrada com o *ID* criado nos testes. Para solucionar isso, foi necessário incluir “rescue ActiveRecord::RecordNotFound” no método “find_tip” do *controller*, o que gerou erros no teste de remoção, demandando alterações e correções nesse processo também.

Durante a revisão desta PR, foi identificada a necessidade de efetuar algumas correções. Primeiramente, foi criado um novo módulo padrão, denominado FindResources, que será incluído em todos os *controllers*. Isso elimina a necessidade de replicar as mesmas funções em múltiplos locais, contribuindo para um código mais limpo e eficiente. Adicionalmente, a lógica de um *Model* específico foi revisada e otimizada, pois estava executando a mesma operação duas vezes. Por fim, os testes relacionados a essas mudanças também foram ajustados para garantir a consistência e a eficácia das correções implementadas.

5.2.8 Desenvolvimento da função para duplicar passos de solução

Durante o desenvolvimento dessa funcionalidade para permitir que o professor duplique qualquer passo de solução e suas dicas associadas, foram encontradas algumas dificuldades. Inicialmente, o título do passo de solução é único, então foi necessário adicionar um texto como “cópia-X”, sendo X o valor incremental representando o número de cópias criadas. Isso permitiu a duplicação do passo de solução. No entanto, a dificuldade seguinte foi garantir a duplicação correta das dicas. Inicialmente, o erro residia na referência do *ID* das dicas duplicadas, que

originalmente apontavam para o *ID* da questão original em vez de referenciar a cópia. Após o ajuste de referência, o processo funcionou corretamente.

As complicações persistiram nos testes do modelo e do controlador, com problemas semelhantes aos encontrados anteriormente na duplicação das dicas. Especificamente, houve dificuldade em encontrar o *ID* passado nos testes, principalmente no teste de um *ID* inexistente. Além disso, surgiram desafios para garantir o correto funcionamento nos testes da duplicação das dicas “filhas”. Após a implementação das correções e padronizações na tarefa anterior, os resultados foram positivamente bem-sucedidos.

5.2.9 Desenvolvimento da função para duplicar exercícios

Nesta etapa, uma funcionalidade foi desenvolvida para permitir que o professor duplique qualquer exercício, incluindo os passos de solução e suas respectivas dicas. Durante a implementação dessa funcionalidade, surgiram dificuldades ao duplicar os elementos dependentes (questões e dicas). Após resolver esses obstáculos e quando a funcionalidade parecia estar correta, os *linters* apontaram um erro, solicitando a pré-carregamento de informações dos elementos dependentes. Após várias adaptações na função de duplicação, ela passou a funcionar corretamente.

Em seguida, os testes foram criados e encontraram problemas semelhantes aos anteriores, principalmente ao simular a duplicação dos elementos dependentes de um exercício. No entanto, após ajustes tanto nos controladores quanto nos testes, tudo foi corrigido e funcionou como esperado.

5.2.10 Desenvolvimento da função para duplicar introduções

Esta atividade consistiu em desenvolver uma funcionalidade que permite ao professor duplicar qualquer introdução. O desenvolvimento dessa atividade foi similar à duplicação de dicas, pois não havia elementos dependentes que precisassem ser duplicados em conjunto, o que tornou o processo mais ágil. Como se assemelhava à duplicação de dicas, alguns ajustes já haviam sido previamente criados, e os erros que surgiram durante a criação dos testes foram mais simples e fáceis de corrigir.

5.2.11 Desenvolvimento da função para duplicar OA

Nesta tarefa, foi desenvolvida a função para duplicar um OA por completo, incluindo todos os seus elementos (introduções, exercícios, passos de solução e dicas). Durante a implementação deste CRUD, também foi encontrada dificuldades na duplicação dos elementos “filhos”, mas, como nos casos anteriores, foram questões mais relacionadas a ajustes e adap-

tações do que a desafios conceituais. Os testes também exigiram ajustes semelhantes aos anteriores, especialmente para lidar com a identificação do *ID*, principalmente no teste de um *ID* inexistente. Após as correções, os testes funcionaram corretamente.

A criação de serviços específicos para tratar a duplicação dos modelos foi uma estratégia eficaz para organizar o código e simplificar a manutenção. Essa abordagem permite que funções específicas sejam gerenciadas de maneira isolada, facilitando a compreensão e a atualização do código. A revisão subsequente e os ajustes realizados garantiram que esses serviços estivessem alinhados com os padrões estabelecidos, contribuindo para a consistência e eficiência do projeto como um todo.

5.2.12 Desenvolvimento da função para atualizar posições das dicas

Nesta tarefa, foi desenvolvida a funcionalidade que permite a um professor alterar a ordem das dicas configuradas, possibilitando a escolha da posição mais adequada para cada uma delas.

Para realizar essa função, foi necessário criar um método que enviasse um *array* contendo todos os *IDs* das dicas de um passo de solução, organizados na ordem desejada para definir suas posições. Essa funcionalidade foi implementada no modelo de passos de solução e apenas chamada no controlador de dicas, para obter todas as dicas que deveriam estar no *array*. Durante essa tarefa, foram encontradas algumas dificuldades para compreender a lógica necessária para reordenar as dicas, mas após discussões com o professor, foi definida a estratégia a ser adotada. Os testes desta funcionalidade também foram criados e exigiram apenas pequenas correções, como ajustes nos *linters* ou na lógica do próprio teste, para funcionarem conforme o esperado.

5.2.13 Desenvolvimento da função para atualizar posições dos passos de solução

Nesta tarefa, foi desenvolvida a funcionalidade que permite a um professor alterar a ordem dos passos de solução configurados. Além disso, uma função foi criada para retornar um *array* contendo todos os *IDs* dos passos de solução de um exercício na ordem desejada, a fim de definir a posição deles. Para acessar todos os passos de solução, essa função foi implementada no modelo de exercícios e invocada no controlador de passos de solução. Os testes foram criados conforme as funções foram adicionadas e exigiram apenas entendimento para serem desenvolvidos, sem apresentar grandes desafios durante a tarefa.

5.2.14 Desenvolvimento da função para atualizar posições das introduções e exercícios dentro do OA

Nesta tarefa, a funcionalidade desenvolvida permitiu que um professor alterasse a ordem de exibição entre introduções e exercícios. Ambos precisavam estar dentro de um *array*, pois compartilhavam o mesmo valor de posição. Assim, se a introdução ocupasse a posição 1 (um), o exercício não poderia ocupá-la e precisaria avançar para a próxima posição disponível.

O principal desafio nesta tarefa foi realmente sincronizar as atualizações de ambos de forma apropriada, considerando a interdependência entre eles. Nos testes do controlador, surgiram dificuldades para testá-los de forma isolada, mas ainda mantendo a relação entre ambos dentro do mesmo contexto.

Durante a revisão da PR, identificou-se um problema na ordenação: as introduções e exercícios estavam sendo ordenados separadamente, sem considerar a possibilidade de intercalação entre eles. Para resolver isso, a lógica de duplicação foi migrada para um novo modelo, onde a ordenação foi ajustada para considerar a intercalação de introduções e exercícios. Com essa mudança, a ordenação passou a refletir a sequência correta, atendendo à necessidade de misturar esses elementos de forma adequada. Os ajustes necessários foram implementados para garantir que a nova lógica de ordenação funcionasse conforme o esperado.

5.2.15 Desenvolvimento da função para controlar exibição das dicas

Nesta tarefa, foi desenvolvida a funcionalidade que permite configurar a forma como as dicas são exibidas, seja de forma sequencial ou todas de uma vez. A maior dificuldade enfrentada nesta tarefa foi relacionada aos testes e à qualidade do código, conforme ilustrado na Figura 33 e Figura 32. O desafio principal foi alinhar a lógica da aplicação com os testes. Um erro que consumiu tempo até ser identificado foi entre o nome do arquivo e o nome da função, o que resultou em um erro de autenticação. Após corrigir esse problema e aprimorar a lógica da aplicação, a tarefa ficou completamente funcional.

```

belavitek@belavitek:~/Desktop/FARMA/farma-api$ ./run test
Running 121 tests in parallel using 6 processes
Run options: --seed 5383

# Running:

.....F

Failure:
Api::Professors::SolutionStepsControllerConfigDisplayModeTest#test_: config_display_mode with invalid mode should be unsuccessful. [/var/www/farma/test/controllers/api/professors/solution_step/config_display_mode_test.rb:38]:
Expected response to be a <422: unprocessable_entity>, but was a <200: OK>
Response body: {"message":"Erro ao configurar o modo de exibição","errors":{}}.
Expected: 422
Actual: 200

rails test test/controllers/api/professors/solution_step/config_display_mode_test.rb:31

..F

Failure:
Api::Professors::SolutionStepsControllerConfigDisplayModeTest#test_: config_display_mode with valid mode should be successfully. [/var/www/farma/test/controllers/api/professors/solution_step/config_display_mode_test.rb:23]:
Expected response to be a <2XX: success>, but was a <401: Unauthorized>
Response body: {"error":"Translation missing. Options considered were:\n- pt-BR.devise.failure.user.unauthenticated\n- pt-BR.devise.failure.unauthenticated"}

rails test test/controllers/api/professors/solution_step/config_display_mode_test.rb:16

.....

Finished in 0.805928s, 150.1375 runs/s, 425.5962 assertions/s.
121 runs, 343 assertions, 2 failures, 0 errors, 0 skips

```

Figura 32 – Erro de testes

Fonte: Autoria própria (2023).

```

belavitek@belavitek:~/Desktop/FARMA/farma-api$ ./run linters
Inspecting 94 files
.....C.....

Abrir arquivo no editor (ctrl + clique)
test/controllers/api/professors/solution_step/config_display_mode_test.rb:17:23: C: [Correctable] Style/PercentLiteralDelimiters: %w-literals should be delimited by [ and ].
    valid_modes = %w(sequencial todas)
                  ^^^^^^^^^^^^^^^^^^^
test/controllers/api/professors/solution_step/config_display_mode_test.rb:32:25: C: [Correctable] Style/PercentLiteralDelimiters: %w-literals should be delimited by [ and ].
    invalid_modes = %w(invalid mode another invalid mode)
                    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
94 files inspected, 2 offenses detected, 2 offenses autocorrectable

Task completed in 0m2.202s
belavitek@belavitek:~/Desktop/FARMA/farma-api$

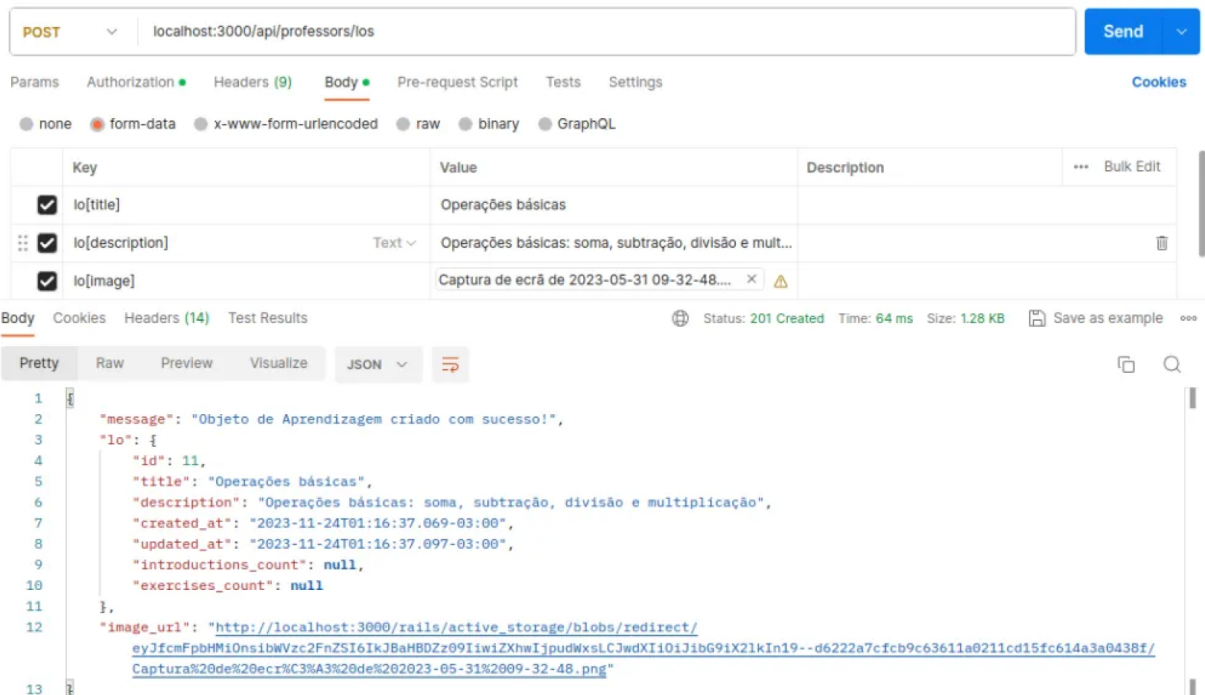
```

Figura 33 – Erro de qualidade de código

Fonte: Autoria própria (2023).

5.2.16 Desenvolvimento do *upload* de imagem no OA

Na tarefa recente, foi implementado a funcionalidade de anexar uma imagem ao criar um OA. Atualmente, essa imagem está sendo armazenada localmente, em vez de em um serviço de armazenamento em nuvem, como a Amazon. A parte do armazenamento local da imagem foi realizada sem grandes desafios. Entretanto, o aspecto mais desafiador desta tarefa foi o desenvolvimento dos testes para o upload da imagem. Isso exigiu uma investigação mais aprofundada e pesquisa para entender a melhor maneira de executá-los. O retorno desse *upload* é um *link* que leva a imagem salva em um *storage* local como mostra a Figura 34.



5.2.17 Documentação da API

No projeto, foi desenvolvida uma documentação abrangente para a API, conforme detalhado no Apêndice B. Esta documentação define de forma clara as rotas do sistema, especificando os parâmetros esperados para cada rota, quais exigem autenticação e os possíveis retornos. Para uma visualização mais eficaz e interativa desta documentação, ela foi disponibilizada em <https://farma-authoring-tool.github.io/Swagger-UI>. Esta ferramenta online proporciona uma visão mais clara e organizada de cada rota, facilitando a compreensão e a interação com a documentação da API.

5.3 FARMA FRONTEND

No início do desenvolvimento do *frontend*, a escolha inicial foi o uso do *React* em conjunto com o *Bootstrap* para estilização em CSS. Porém, durante a etapa inicial de progresso, foi considerado que o *Mantine* seria mais apropriado para a estilização, principalmente pela sua compatibilidade com o *React* e sua capacidade de oferecer um estilo visualmente atraente sem demandar extensas configurações manuais.

Após a conclusão da primeira fase do projeto, houve a decisão de migrar o *frontend* para utilizar o *Next*. Essa escolha foi motivada pelas vantagens que o *Next* apresenta, sendo

um *framework* baseado em *React* que simplifica a integração com APIs. O *Next* oferece recursos adicionais sem requerer configurações manuais complexas, otimizando o processo de desenvolvimento e melhorando a eficiência e funcionalidade geral.

5.3.1 Desenvolvimento do CRUD de OA com *React*

Na primeira atividade realizada no *frontend*, o foco era implementar as funcionalidades de criar, ler, atualizar e deletar (CRUD) OAs para que os professores pudessem gerenciar seus próprios OAs através de uma interface. Desde o início, foi encontrado desafios ao integrar o *Bootstrap* com o *React*, o que demandou várias adaptações.

Após explorar diferentes opções para substituir o *Bootstrap*, o *Mantine UI* se destacou como a escolha mais adequada por ser uma biblioteca de componentes *React* que simplifica a criação de um *design* consistente e atraente. Uma vez resolvida a questão do *design*, o próximo desafio foi estabelecer uma comunicação eficaz entre as rotas do *frontend* e as chamadas necessárias à API. Isso exigiu ajustes adicionais na API e uma compreensão detalhada de como gerenciar essa interação no *frontend*, assim como formatar os dados para atender às expectativas da API. Com essas configurações implementadas e o funcionamento do CRUD garantido, um novo desafio foi encontrado, relacionado ao *linter*, que identificou um problema em um formulário *React*. A solução sugerida pelo *linter*, no entanto, afetava a capacidade de editar corretamente os valores nos formulários, exigindo uma abordagem alternativa para manter a funcionalidade sem comprometer a qualidade do código. Dado que a solução alternativa não era considerada a abordagem ideal, a decisão tomada foi de finalizar a API antes de iniciar o desenvolvimento do *frontend* de uma maneira mais apropriada.

5.3.2 Desenvolvimento do *login* e CRUD de OA com *Next* e *Tailwind*

Após constatar que o *frontend* com o *React* não era a melhor opção, o desenvolvimento prosseguiu com o *Next*. Nesse contexto, a escolha do *framework* CSS *Tailwind* revelou-se bastante vantajosa, uma vez que possibilitou a utilização de diversos componentes prontos, proporcionando um *design* agradável de maneira ágil e descomplicada. Além disso, sua integração ao projeto *Next* facilitou ainda mais o processo.

Na execução dessa tarefa, em consenso com o orientador, optou-se por criar um novo projeto utilizando o *Next*.

5.3.3 Métricas

Conforme descrito na seção 5.2, este trabalho apresentou uma visão abrangente do desenvolvimento de cada tarefa, essenciais para a composição das funcionalidades da área do

professor. Detalhou-se cada etapa do desenvolvimento, destacando as realizações e os desafios encontrados. Este processo não apenas contribuiu para a construção de um sistema mais robusto e funcional, mas também proporcionou aprendizados valiosos e *insights* para futuras melhorias e desenvolvimentos no campo da tecnologia educacional.

Neste trabalho foram desenvolvidas 29 *issues*, totalizando 29 PRs formadas por 7 *models*, 8 *controllers*, 38 rotas, 40 testes, 4089 linhas de códigos. Na Tabela 2 é possível observar as métricas alcançadas em cada tarefa do desenvolvimento, contendo a quantidade de modificações referente em cada coluna.

Tabela 2 – Métricas do desenvolvimento

Tarefas	<i>Models</i>	<i>Controllers</i>	Rotas	Testes	+ linhas	- linhas
Template para PR	0	0	0	0	51	0
CRUD OA	1	2	5	8	473	92
CRUD introduções	1	1	5	7	352	4
CRUD exercícios	1	1	5	7	345	1
CRUD passos de soluções	1	1	5	8	384	6
CRUD dicas	1	1	5	7	315	2
Autenticação de usuário	1	2	3	29	1022	115
Duplicar dicas	1	1	1	3	139	19
Duplicar passos de solução	2	1	1	3	96	25
Duplicar exercícios	3	2	1	4	123	20
Duplicar introduções	1	1	1	3	76	13
Duplicar OA	5	1	1	3	249	45
Reordenar dicas	1	1	1	2	70	1
Reordenar passos de solução	1	1	1	3	69	11
Reordenar itens do OA	1	1	1	2	95	0
Configuração de exibição das dicas	1	1	0	4	84	7
Upload de imagem no OA	1	1	1	1	146	4

Fonte: Autoria própria (2023).

6 RESULTADOS

Com base no Capítulo 4, Análise e Projeto, onde o levantamento de requisitos, a escolha do *Framework* CSS e a configuração do ambiente de desenvolvimento da API foram descritos, mas também, considerando o Capítulo 5, Desenvolvimento do Sistema, onde foi descrito o processo de codificação dos requisitos do sistema, neste capítulo será apresentado o sistema como um todo. Para tanto, as descrições serão guiadas com o uso das telas desenvolvidas na seção 5.3, FARMA *FRONTEND*. Foram desenvolvidas sete telas, são elas: *Home*, *Login*, *Cadastro de usuário*, *Listagem de OA's*, *Criar novo OA*, *Editar OA* e *Visualizar OA*.

A tela *Home* tem como propósito ser a tela inicial do sistema, propiciando ao usuário ter acesso ao *login*. A Figura 35, mostra o *design* desta tela.

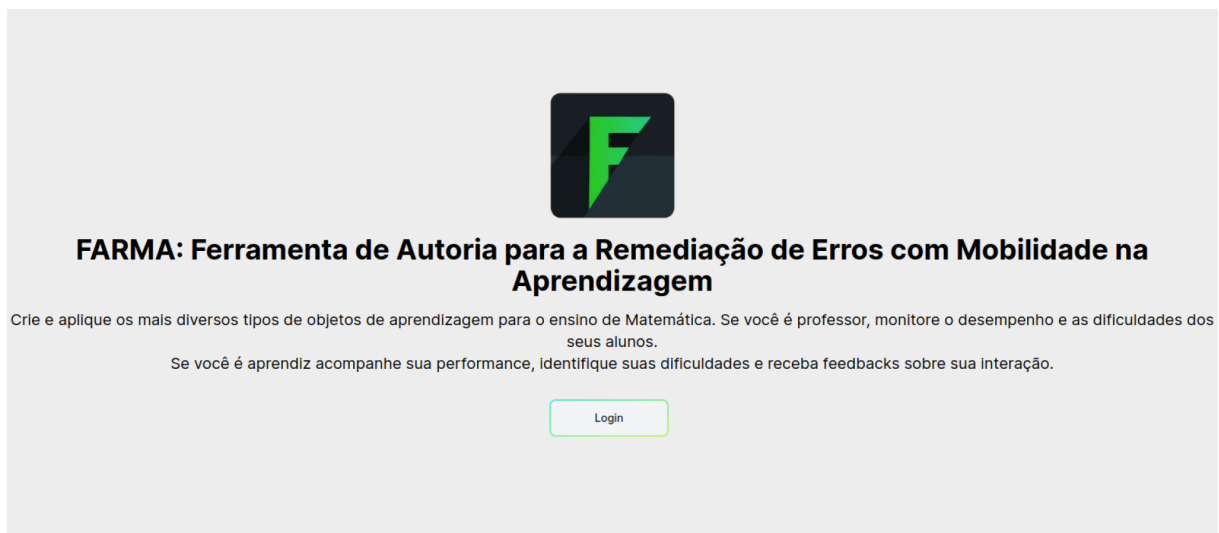
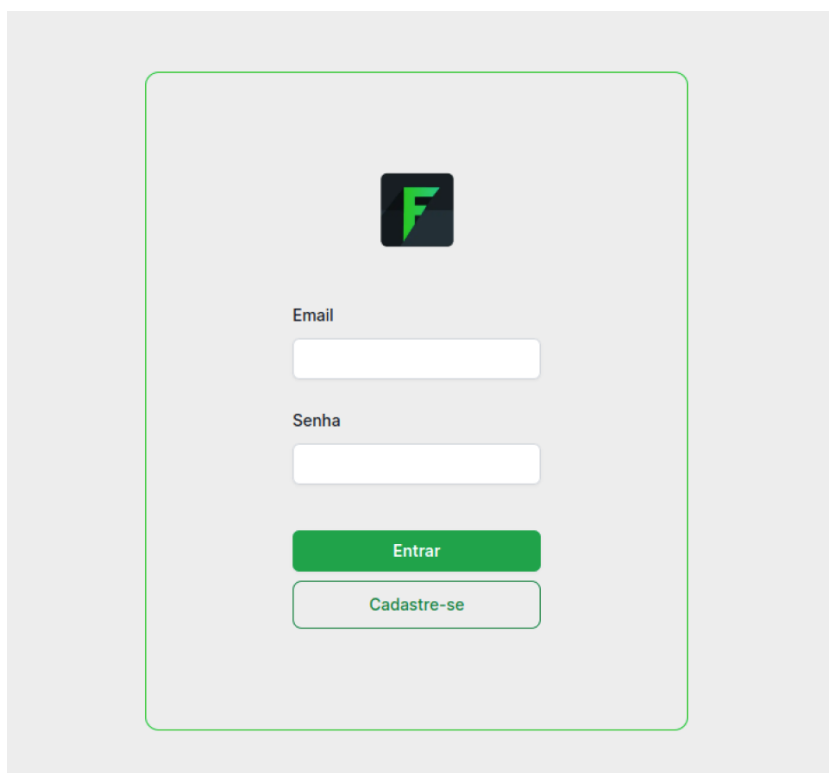


Figura 35 – Home
Fonte: Autoria própria (2023).

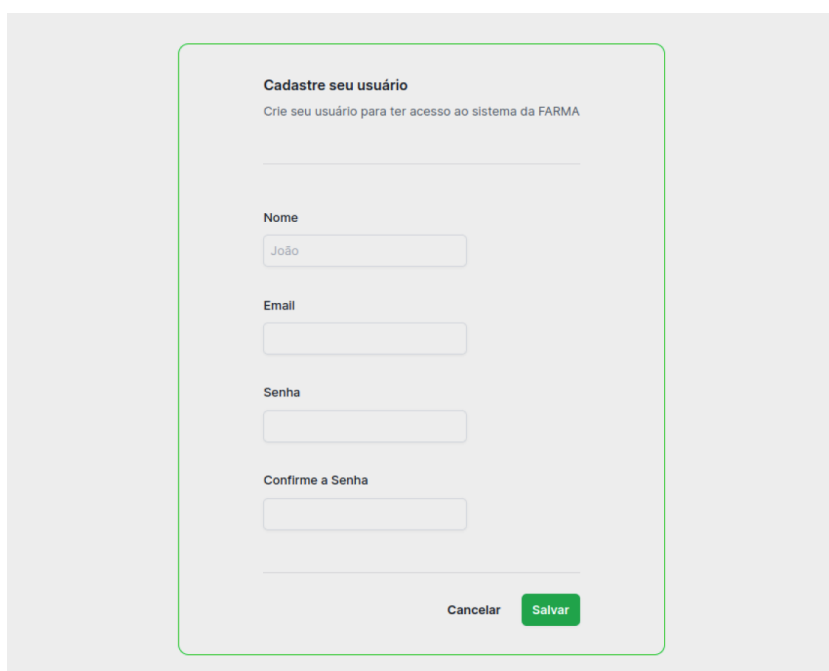
A tela *Login* possibilita ao usuário, por meio de suas credenciais, acessar o sistema ou navegar para a tela de cadastro. A Figura 36 ilustra essa tela.



The login form is centered on a light gray background. It features a dark green square logo with a white 'F' at the top. Below the logo are two white input fields: the first is labeled 'Email' and the second is labeled 'Senha'. Underneath these fields are two buttons: a solid green button labeled 'Entrar' and a white button with a green border labeled 'Cadastre-se'.

Figura 36 – Login
Fonte: Autoria própria (2023).

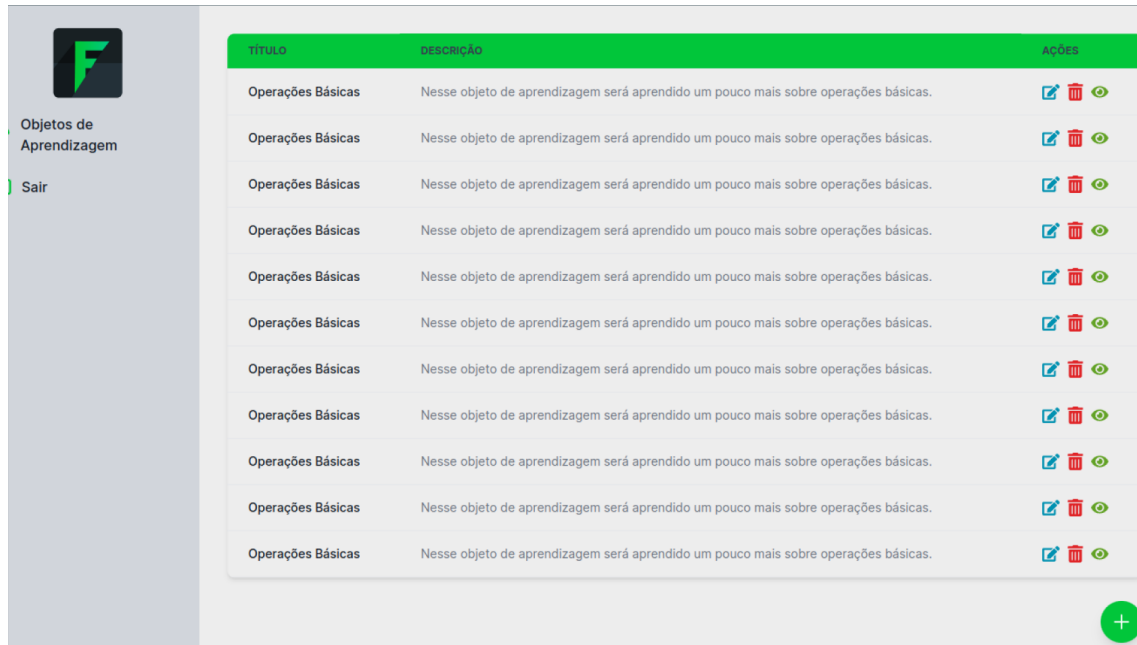
Na Figura 37, observa-se a tela de *Cadastro de usuário*. Nesta tela o usuário tem a possibilidade de criar seu registro no sistema.



The registration form is titled 'Cadastre seu usuário' and includes the subtitle 'Crie seu usuário para ter acesso ao sistema da FARMA'. It contains five white input fields: a name field (pre-filled with 'João'), an email field, a password field labeled 'Senha', and a confirmation field labeled 'Confirme a Senha'. At the bottom right, there are two buttons: a white 'Cancelar' button and a green 'Salvar' button.

Figura 37 – Registrar novo usuário
Fonte: Autoria própria (2023).

A Figura 38 elucida a tela de *Listagem de OA's*, onde visualiza-se, em formato de tabela, todos os OA's. Além disso, permite o acesso as ações (Edição, Exclusão e Visualização) de cada OA, bem como permite o acesso ao formulário para a criação de um novo OA.



TÍTULO	DESCRIÇÃO	AÇÕES
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	
Operações Básicas	Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.	

Figura 38 – Objetos de Aprendizagem

Fonte: Autoria própria (2023).

A tela *Criar novo OA*, expõe o formulário de criação de OA, conforme apresentado na Figura 39.



Objeto de Aprendizagem
Crie seu novo Objeto de Aprendizagem.

Título
Teste

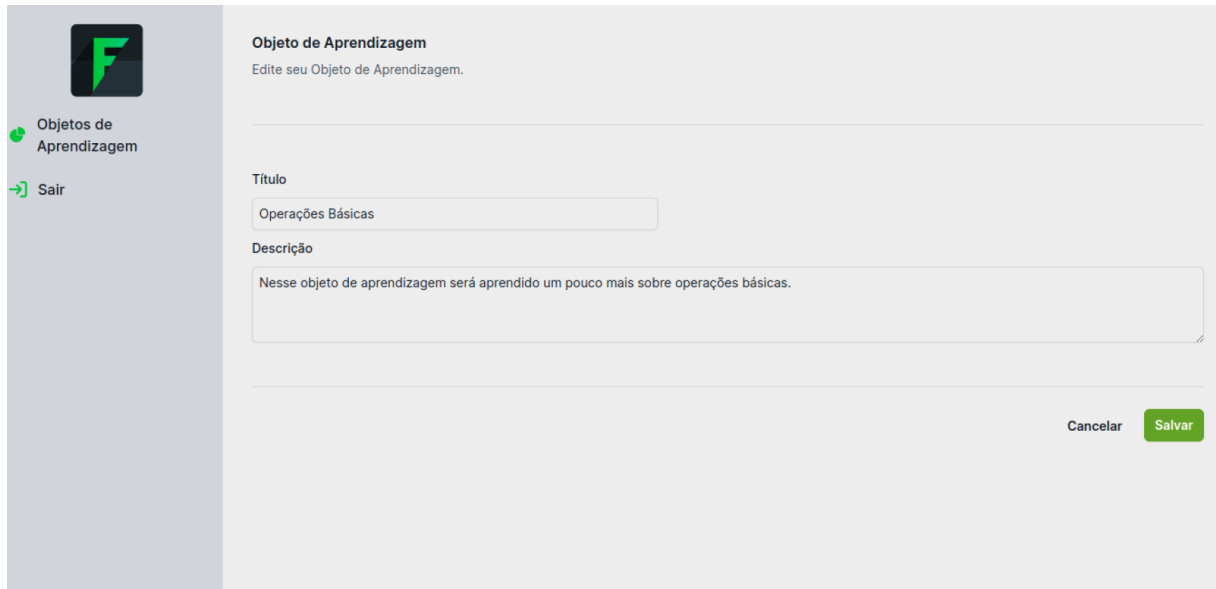
Descrição

Cancelar Salvar

Figura 39 – Novo OA

Fonte: Autoria própria (2023).

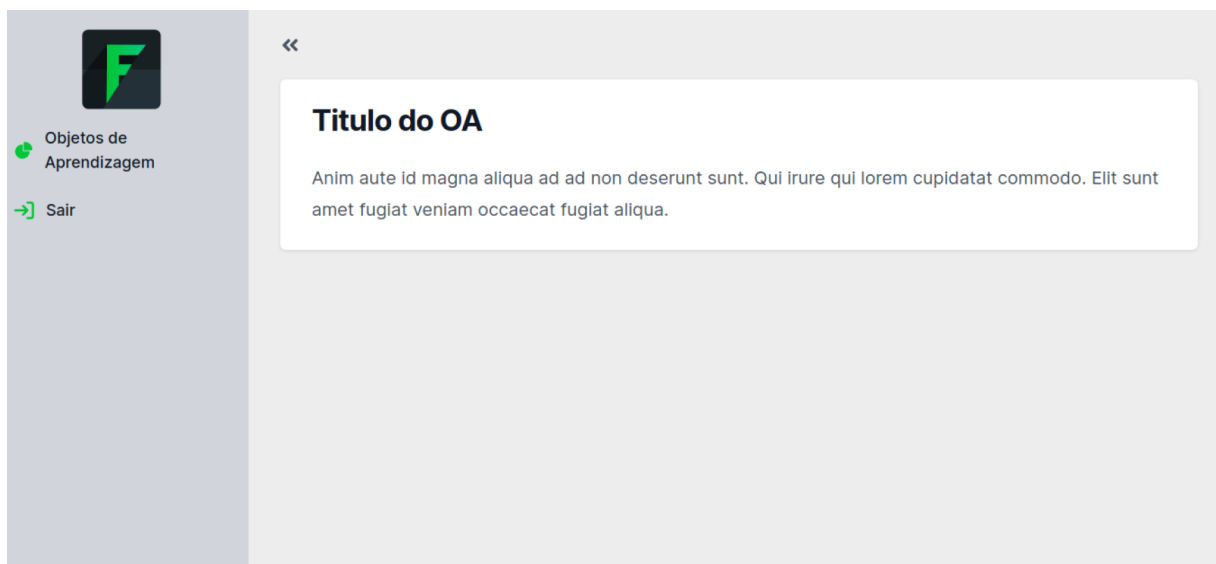
Na Figura 40 visualiza-se a tela de *Editar OA*. Esta tela possui o formulário de edição, onde, os campos já vem preenchidos com as informações do OA e podem ser alterados conforme o desejado.



A interface 'Editar OA' apresenta um menu lateral à esquerda com o logotipo 'F' em verde, o texto 'Objetos de Aprendizagem' e um link 'Sair' com uma seta verde. O conteúdo principal, intitulado 'Objeto de Aprendizagem' com o subtítulo 'Edite seu Objeto de Aprendizagem.', contém um formulário. O campo 'Título' possui o texto 'Operações Básicas'. O campo 'Descrição' possui o texto 'Nesse objeto de aprendizagem será aprendido um pouco mais sobre operações básicas.'. Na base direita do formulário, há dois botões: 'Cancelar' e 'Salvar' (destacado em verde).

Figura 40 – Editar OA
Fonte: Autoria própria (2023).

E por fim, a tela de *Visualizar OA*. Nesta tela o OA é exibido para acesso a sua descrição completa, e futuramente aos demais componentes, conforme ilustrado na Figura 41.



A interface 'Visualizar OA' mantém o mesmo menu lateral. O conteúdo principal, precedido por uma seta de retorno '<<', exibe o 'Título do OA' em negrito. Abaixo dele, há um bloco de texto com o conteúdo: 'Anim aute id magna aliqua ad ad non deserunt sunt. Qui irure qui lorem cupidatat commodo. Elit sunt amet fugiat veniam occaecat fugiat aliqua.'.

Figura 41 – Visualizar OA
Fonte: Autoria própria (2023).

Com isso é possível visualizar o sistema funcionando por completo, onde todas as telas que foram desenvolvidas realizam a conexão com a API, fazendo as requisições necessárias e recebendo as respostas esperadas.

7 CONCLUSÃO

Conforme evidenciado no Capítulo 2, o sistema já possuía versões anteriores que ainda apresentavam carências e seus recursos não estavam unificados em apenas uma versão. Portanto, o trabalho proposto teve como objetivo dar continuidade ao projeto já existente, onde foi desenvolvido uma nova versão a qual engloba todas as funcionalidades, preenchendo as demandas de melhorias de forma incremental. O foco deste trabalho foi o desenvolvimento da área destinada aos professores, que lhes permite criar e gerenciar OA's interativos para auxiliar no ensino de matemática.

Este trabalho concentrou-se em cumprir os requisitos apresentados na seção 4.1, que detalha as funcionalidades necessárias para a área do professor do sistema. Entretanto, não abordou os itens listados na seção 7.1, devido a limitações de tempo no cronograma do projeto. Na seção 4.3 foi possível entender como se deu a realização das configurações do ambiente inicial da API, com todas as tecnologias utilizadas, juntamente com um exemplo de como funciona o fluxo para cada resolução de tarefas.

Com o desenvolvimento deste trabalho foi possível obter a API da área do professor, com suas funcionalidades básicas. Também, obteve-se a documentação de cada rota existente no sistema, como apresentado no Apêndice B, onde exemplifica os parâmetros esperados e os retornos obtidos. Além disso, iniciou-se também o desenvolvimento do *frontend*, deixando apenas um exemplo pré configurado de como deve ser para que seja prosseguido com seu desenvolvimento no futuro.

7.1 Trabalhos Futuros

Devido a restrições de prazo, nem todos os requisitos inicialmente definidos puderam ser atendidos neste trabalho. Abaixo estão descritas as atividades que ainda necessitam ser desenvolvidas em trabalhos futuros, dando continuidade e evoluindo esta ferramenta. São elas:

- a. O sistema deve permitir escolher se os OA's serão ou não compartilhados;
- b. O sistema deve permitir realizar upload de arquivos nas introduções;
- c. O sistema deve permitir escolher se as introduções serão ou não disponíveis para a visualização;
- d. O sistema deve permitir realizar upload de arquivos nos exercícios;
- e. O sistema deve permitir escolher se os exercícios serão ou não disponíveis para a visualização;
- f. O sistema deve permitir realizar upload de arquivos nos passos de solução;

- g. O sistema deve permitir escolher se os passos de solução serão ou não disponíveis para a visualização;
- h. O sistema deve garantir que todas os arquivos enviadas sejam armazenadas em um serviço de storage remoto, em vez de serem armazenados localmente;
- i. O desenvolvimento do *frontend* do sistema deve ser estendido para incorporar as funcionalidades de CRUD relacionadas a introduções, exercícios, passos de solução e dicas.
- j. O sistema deve possuir uma interface atrativa e com *layouts* harmônicos;
- k. O sistema deve ter uma navegação intuitiva, onde não precise necessariamente seguir uma sequência de etapas para chegar ao destino desejado;
- l. As informações do sistema devem ser apresentadas de maneira agradável, evitando repetições e trazendo as mesmas de modo intuitivo.

REFERÊNCIAS

- AGUIAR, E. V. B.; FLÔRES, M. L. P. Objetos de aprendizagem: conceitos básicos. **Objetos de aprendizagem: teoria e prática**. Porto Alegre: Evangraf, p. 12–28, 2014.
- BEHAR, P. A. Avaliação de softwares educacionais no processo de ensino-aprendizagem computadorizado: estudo de caso. 1993.
- INTRODUCTION to JSON Web Tokens. 2023. <https://jwt.io/introduction>. Acessado em 19 de novembro de 2023.
- KENSKI, V. M. **Educação e tecnologias: o novo ritmo da informação**. [S.l.]: Papirus editora, 2003.
- KUTZKE, A. R.; DIRENE, A. Informática educacional e a mediação do erro na educação: um estudo teórico-crítico e uma proposta de instrumento computacional. *In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2016. v. 5, n. 1, p. 12.
- LEITE, M. D. Arquitetura para remediação de erros baseada em múltiplas representações externas. 2013.
- LORENZO, C. O consentimento livre e esclarecido e a realidade do analfabetismo funcional no brasil: uma abordagem para a norma e para além da norma. **Revista Bioética**, Conselho Federal de Medicina, v. 15, n. 2, p. 268–282, 2007.
- MARCZAL, D.; DIRENE, A. I. Um arcabouço que enfatiza a retroação a contextos de erro na solução de problemas. **Revista Brasileira de Informática na Educação**, v. 19, n. 01, p. 63, 2011.
- MARCZAL, D. *et al.* Farma: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. Universidade Federal do Paraná, 2014.
- MOURA, V. A. B. de; PERES, L. M. Avaliação do impacto da retroação na aprendizagem apoiada por uma ferramenta educacional. **Revista Brasileira de Informática na Educação**, v. 25, n. 01, p. 60, 2017.
- SANTOS, A. G. d. **Desenvolvimento do módulo de estatísticas da ferramenta de autoria de objetos de aprendizagem FARMA**. 2022. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2022.
- SANTOS, F. R. L. Vícios intelectuais e as redes sociais: o acesso constante à informação nos torna intelectualmente viciosos? **Veritas (Porto Alegre)**, v. 62, n. 3, p. 657–682, 2017.
- SANTOS, J. A.; FRANÇA, K. V.; SANTOS, L. S. B. d. Dificuldades na aprendizagem de matemática. **Monografia de Graduação em Matemática**. São Paulo: UNASP, 2007.
- SILVA, R. C.; DIRENE, A. I. **Sequenciamento Adaptativo de Exercícios Baseados na Correspondência entre a Dificuldade da Solução e o Desempenho Dinâmico do Aprendiz**. 2015. Tese (Doutorado) — Tese de doutorado, Universidade Federal do Paraná, Curitiba, 2015.
- SILVA, R. F. d.; CORREA, E. S. Novas tecnologias e educação: a evolução do processo de ensino e aprendizagem na sociedade contemporânea. **Educação e Linguagem**, ano, v. 1, n. 1, p. 23–25, 2014.
- TAROUCO, L. M. R. *et al.* Objetos de aprendizagem: teoria e prática. Evangraf, 2014.

VICENTE, J. J. N. B.; ROCHA, M. F. L. da. Sobre o ato de educar: um breve olhar. **Saberes: Revista interdisciplinar de Filosofia e Educação**, v. 22, n. 1, p. 78–93, 2022.

APÊNDICE A – Casos de uso dos usuários do sistema FARMA

A.1 Relação dos usuários.

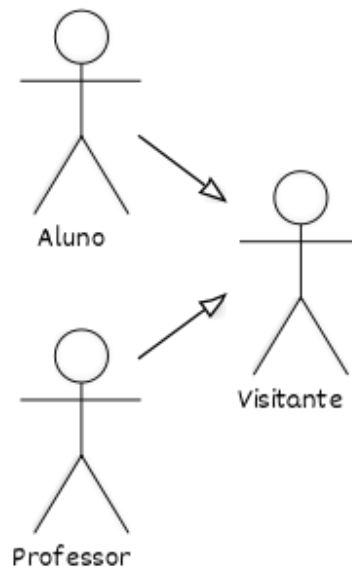


Figura 42 – Relação dos usuários

Fonte: Autoria própria (2023).

A.2 Caso de Uso Visitante.



Figura 43 – Caso de uso visitante

Fonte: Autoria própria (2023).

A.3 Caso de Uso Aluno.

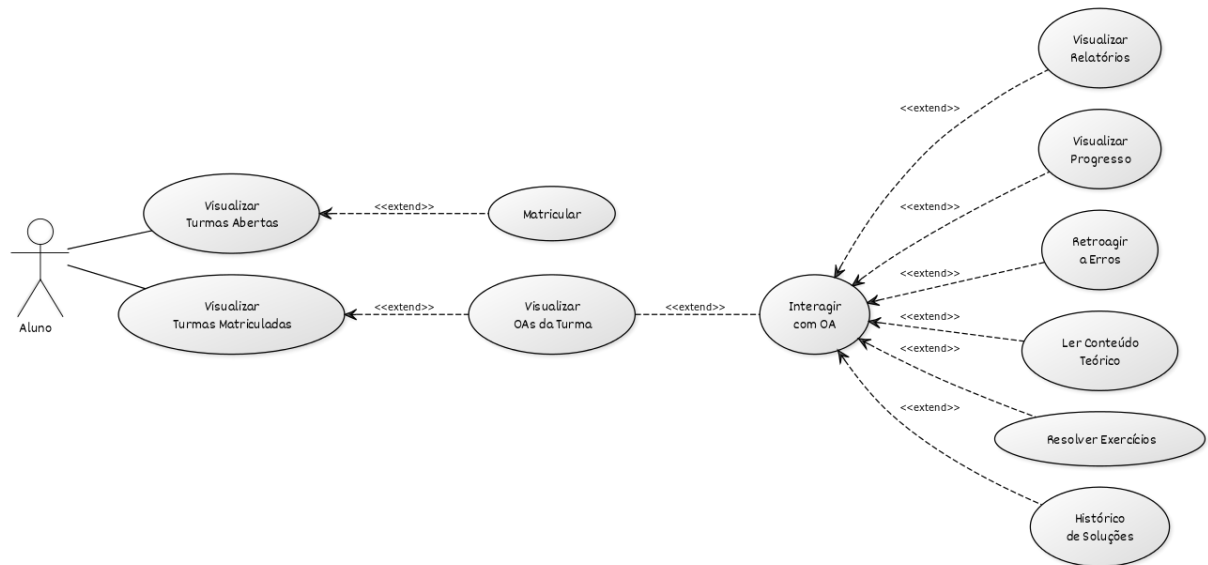


Figura 44 – Caso de uso aluno

Fonte: Autoria própria (2023).

A.4 Caso de Uso Professor.

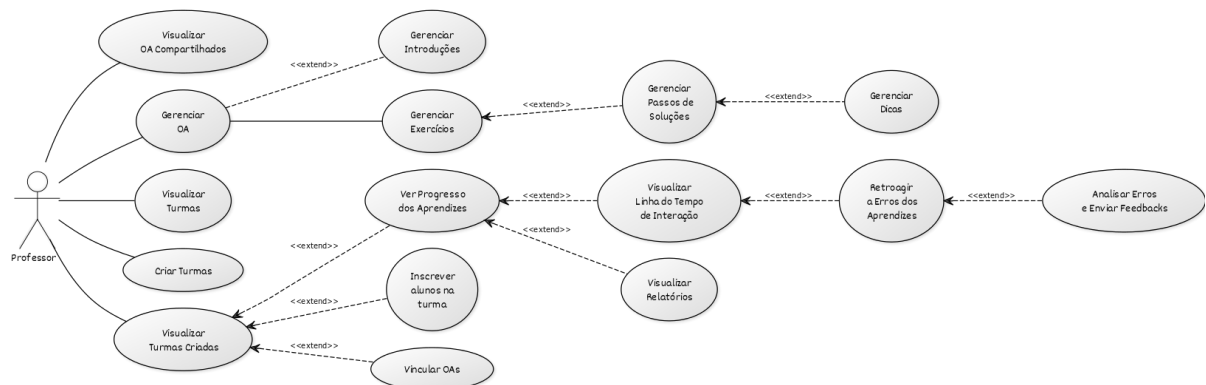


Figura 45 – Caso de uso professor

Fonte: Autoria própria (2023).

APÊNDICE B – Documentação da API

```

openapi: 3.0.0
info:
  title: FARMA API
  version: 1.0.0
paths:
  /login:
    post:
      summary: Realiza o login do usuário
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                user:
                  type: object
                  properties:
                    email:
                      type: string
                    password:
                      type: string
      responses:
        '200':
          description: Login bem-sucedido
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/User'
        '401':
          description: e-mail ou senha incorretos
          content:
            application/json:
              schema:
                type: object
                properties:
                  message:
                    type: string
                    example: "E-mail ou senha incorretos."
  /signup:
    post:
      summary: Cria um novo usuário
      requestBody:
        required: true
        content:
          application/json:

```

```

    schema:
      type: object
      properties:
        user:
          type: object
          properties:
            email:
              type: string
            password:
              type: string
            name:
              type: string
  responses:
    '201':
      description: Usuário criado com sucesso
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/User'
    '422':
      description: e-mail já está em uso
      content:
        application/json:
          schema:
            type: object
            properties:
              message:
                type: string
                example: "E-mail já está em uso."
/logout:
  post:
    summary: Realiza o logout do usuário
    security:
      - bearerAuth: []
    responses:
      '200':
        description: Logout realizado com sucesso
        content:
          application/json:
            schema:
              type: object
              properties:
                message:
                  type: string
                  example: "Usuário deslogado com sucesso"
      '401':

```

```
description: Não autorizado - Usuário não está logado
content:
  application/json:
    schema:
      type: object
      properties:
        message:
          type: string
      example: "Usuário não está logado ou a sessão expirou"
```

/api/professors/los:

```
get:
  summary: Lista todos os objetos de aprendizagem (LOs)
  security:
    - bearerAuth: []
  responses:
    '200':
      description: Lista de objetos de aprendizagem
      content:
        application/json:
          schema:
            type: array
            items:
              $ref: '#/components/schemas/Lo'
    '401':
      description: Signature has expired
```

post:

```
summary: Cria um novo objeto de aprendizagem
security:
  - bearerAuth: []
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Lo'
responses:
  '201':
    description: Objeto de aprendizagem criado
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Lo'
  '401':
    description: Signature has expired
  '422':
```

```

description: Erro de validação
content:
  application/json:
    schema:
      type: object
      properties:
        errors:
          type: object
          properties:
            title:
              type: array
              items:
                type: string
                enum: [já está em uso]
                example: já está em uso
            description:
              type: array
              items:
                type: string
                enum: [não pode ficar em branco]
                example: não pode ficar em branco
          required:
            - title
            - description
/api/professors/los/{id}:
  get:
    summary: Obtém um objeto de aprendizagem pelo ID
    security:
      - bearerAuth: []
    parameters:
      - in: path
        name: id
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Detalhes do objeto de aprendizagem
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Lo'
      '401':
        description: Signature has expired
      '422':
        description: Erro de validação

```

```

content:
  application/json:
    schema:
      type: object
      properties:
        errors:
          type: object
          properties:
            title:
              type: array
              items:
                type: string
                enum: [já está em uso]
                example: já está em uso
            description:
              type: array
              items:
                type: string
                enum: [não pode ficar em branco]
                example: não pode ficar em branco
          required:
            - title
            - description

```

```

put:
  summary: Atualiza um objeto de aprendizagem
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: id
      required: true
      schema:
        type: integer
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Lo'
  responses:
    '202':
      description: Objeto de aprendizagem atualizado
    '401':
      description: Signature has expired
    '404':

```

```

    description: Objeto de aprendizagem não encontrado
  '422':
    description: Erro de validação
    content:
      application/json:
        schema:
          type: object
          properties:
            errors:
              type: object
              properties:
                title:
                  type: array
                  items:
                    type: string
                    enum: [já está em uso]
                    example: já está em uso
                description:
                  type: array
                  items:
                    type: string
                    enum: [não pode ficar em branco]
                    example: não pode ficar em branco
              required:
                - title
                - description
delete:
  summary: Remove um objeto de aprendizagem
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: id
      required: true
      schema:
        type: integer
  responses:
    '200':
      description: Objeto de aprendizagem removido
    '404':
      description: Objeto de aprendizagem não encontrado
/api/professors/los/{id}/duplicate:
  post:
    summary: Duplica um objeto de aprendizagem
    security:
      - bearerAuth: []

```



```

parameters:
  - in: path
    name: id
    required: true
    schema:
      type: integer
responses:
  '201':
    description: Objeto de aprendizagem duplicado
  '401':
    description: Signature has expired
  '404':
    description: Objeto de aprendizagem não encontrado

/api/professors/los/sort_pages:
  patch:
    summary: Reordena os objetos de aprendizagem
    security:
      - bearerAuth: []
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              order:
                type: array
              items:
                type: object
                properties:
                  id:
                    type: integer
                  class:
                    type: string
    responses:
      '200':
        description: Objetos de aprendizagem reordenados
      '401':
        description: Signature has expired

/api/professors/los/{lo_id}/introductions:
  get:
    summary: Lista todas as introduções de um objeto de aprendizagem
    security:
      - bearerAuth: []

```

```

parameters:
  - in: path
    name: lo_id
    required: true
    schema:
      type: integer
responses:
  '200':
    description: Lista de introduções
    content:
      application/json:
        schema:
          type: array
          items:
            $ref: '#/components/schemas/Introduction'
  '401':
    description: Signature has expired
post:
  summary: Cria uma nova introdução para um objeto de aprendizagem
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: lo_id
      required: true
      schema:
        type: integer
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Introduction'
  responses:
    '201':
      description: Introdução criada
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Introduction'
    '401':
      description: Signature has expired
    '422':
      description: Erro de validação
      content:
        application/json:

```

```

    schema:
      type: object
      properties:
        message:
          type: string
          example: "Alguns erros foram encontrados, por favor
verifique."
        errors:
          type: object
          properties:
            title:
              type: array
              items:
                type: string
                example: "já está em uso"
            description:
              type: array
              items:
                type: string
                example: "não pode ficar em branco"
/api/professors/los/{lo_id}/introductions/{id}:
  get:
    summary: Obtém uma introdução pelo ID
    security:
      - bearerAuth: []
    parameters:
      - in: path
        name: lo_id
        required: true
        schema:
          type: integer
      - in: path
        name: id
        required: true
        schema:
          type: integer
    responses:
      '200':
        description: Detalhes da introdução
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Introduction'
      '401':
        description: Signature has expired
      '404':

```

```

        description: Introdução não encontrada
    put:
        summary: Atualiza uma introdução
        security:
            - bearerAuth: []
        parameters:
            - in: path
              name: lo_id
              required: true
              schema:
                type: integer
            - in: path
              name: id
              required: true
              schema:
                type: integer
        requestBody:
            required: true
            content:
                application/json:
                    schema:
                        $ref: '#/components/schemas/Introduction'
        responses:
            '202':
                description: Introdução atualizada
            '401':
                description: Signature has expired
            '404':
                description: Introdução não encontrada
            '422':
                description: Erro de validação
                content:
                    application/json:
                        schema:
                            type: object
                            properties:
                                message:
                                    type: string
                                    example: "Alguns erros foram encontrados, por favor
verifique."
                        errors:
                            type: object
                            properties:
                                title:
                                    type: array
                                    items:

```

```

        type: string
        example: "já está em uso"
    description:
        type: array
        items:
            type: string
            example: "não pode ficar em branco"
delete:
    summary: Remove uma introdução
    security:
        - bearerAuth: []
    parameters:
        - in: path
          name: lo_id
          required: true
          schema:
              type: integer
        - in: path
          name: id
          required: true
          schema:
              type: integer
    responses:
        '200':
            description: Introdução removida
        '401':
            description: Signature has expired
        '404':
            description: Introdução não encontrada
/api/professors/los/{lo_id}/introductions/{id}/duplicate:
    post:
        summary: Duplica uma introdução de um objeto de aprendizagem
        security:
            - bearerAuth: []
        parameters:
            - in: path
              name: lo_id
              required: true
              schema:
                  type: integer
            - in: path
              name: id
              required: true
              schema:
                  type: integer
            - in: path
              name: id
              required: true
              schema:
                  type: integer
        description: ID do objeto de aprendizagem ao qual a introdução
        pertence
        - in: path
          name: id
          required: true
          schema:
              type: integer

```

```

        type: integer
        description: ID da introdução a ser duplicada
    responses:
        '201':
            description: Introdução duplicada com sucesso
            content:
                application/json:
                    schema:
                        $ref: '#/components/schemas/Introduction'
        '401':
            description: Signature has expired
        '404':
            description: Introdução ou objeto de aprendizagem não encontrado

/api/professors/los/{lo_id}/exercises:
    get:
        summary: Lista todos os exercícios de um objeto de aprendizagem
        security:
            - bearerAuth: []
        parameters:
            - in: path
              name: lo_id
              required: true
              schema:
                  type: integer
        responses:
            '200':
                description: Lista de exercícios
                content:
                    application/json:
                        schema:
                            type: array
                            items:
                                $ref: '#/components/schemas/Exercise'
            '401':
                description: Signature has expired

    post:
        summary: Cria um novo exercício para um objeto de aprendizagem
        security:
            - bearerAuth: []
        parameters:
            - in: path
              name: lo_id
              required: true
              schema:

```

```

        type: integer
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Exercise'
responses:
  '201':
    description: Exercício criado
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Exercise'
  '401':
    description: Signature has expired
  '422':
    description: Erro de validação
    content:
      application/json:
        schema:
          type: object
          properties:
            message:
              type: string
              example: "Alguns erros foram encontrados, por favor
verifique."
          exercise:
            $ref: '#/components/schemas/Exercise'
          errors:
            type: object
            properties:
              title:
                type: array
                items:
                  type: string
                  example: "não pode ficar em branco"
              description:
                type: array
                items:
                  type: string
                  example: "não pode ficar em branco"
/api/professors/{lo_id}/exercises/{id}:
  get:
    summary: Obtém um exercício pelo ID
    security:

```

```

    - bearerAuth: []
parameters:
  - in: path
    name: lo_id
    required: true
    schema:
      type: integer
  - in: path
    name: id
    required: true
    schema:
      type: integer
responses:
  '200':
    description: Detalhes do exercício
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Exercise'
  '401':
    description: Signature has expired
  '404':
    description: Exercício não encontrado

put:
  summary: Atualiza um exercício
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: lo_id
      required: true
      schema:
        type: integer
    - in: path
      name: id
      required: true
      schema:
        type: integer
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Exercise'
  responses:

```



```

'202':
  description: Exercício atualizado
'401':
  description: Signature has expired
'404':
  description: Exercício não encontrado
'422':
  description: Erro de validação
  content:
    application/json:
      schema:
        type: object
        properties:
          message:
            type: string
            example: "Alguns erros foram encontrados, por favor
verifique."
          exercise:
            $ref: '#/components/schemas/Exercise'
        errors:
          type: object
          properties:
            title:
              type: array
              items:
                type: string
                example: "não pode ficar em branco"
            description:
              type: array
              items:
                type: string
                example: "não pode ficar em branco"

delete:
  summary: Remove um exercício
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: lo_id
      required: true
      schema:
        type: integer
    - in: path
      name: id
      required: true

```

```

        schema:
          type: integer
      responses:
        '200':
          description: Exercício removido
        '401':
          description: Signature has expired
        '404':
          description: Exercício não encontrado
/api/professors/los/{lo_id}/exercises/{id}/duplicate:
  post:
    summary: Duplica um exercício
    security:
      - bearerAuth: []
    parameters:
      - in: path
        name: lo_id
        required: true
        schema:
          type: integer
      - in: path
        name: id
        required: true
        schema:
          type: integer
    responses:
      '201':
        description: Exercício duplicado
      '401':
        description: Signature has expired
      '404':
        description: Exercício não encontrado

/api/professors/los/{lo_id}/exercises/{exercise_id}/solution_steps:
  parameters:
    - in: path
      name: lo_id
      required: true
      schema:
        type: integer
    - in: path
      name: exercise_id
      required: true
      schema:
        type: integer
  get:

```

```

summary: Lista todos os passos de solução de um exercício
security:
  - bearerAuth: []
parameters:
  - in: path
    name: exercise_id
    required: true
    schema:
      type: integer
responses:
  '200':
    description: Lista de passos de solução
    content:
      application/json:
        schema:
          type: array
          items:
            $ref: '#/components/schemas/SolutionStep'
  '401':
    description: Signature has expired

post:
summary: Cria um novo passo de solução para um exercício
security:
  - bearerAuth: []
parameters:
  - in: path
    name: exercise_id
    required: true
    schema:
      type: integer
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/SolutionStep'
responses:
  '201':
    description: Passo de solução criado
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SolutionStep'
  '401':
    description: Signature has expired

```

```

'422':
  description: Erro de validação
  content:
    application/json:
      schema:
        type: object
        properties:
          message:
            type: string
            example: "Alguns erros foram encontrados, por favor
verifique."

          solutionStep:
            $ref: '#/components/schemas/SolutionStep'
          errors:
            type: object
            properties:
              title:
                type: array
                items:
                  type: string
                  example: "não pode ficar em branco"
              description:
                type: array
                items:
                  type: string
                  example: "não pode ficar em branco"
/api/professors/{lo_id}/exercises/{exercise_id}/solution_steps/{id}:
  parameters:
    - in: path
      name: lo_id
      required: true
      schema:
        type: integer
    - in: path
      name: exercise_id
      required: true
      schema:
        type: integer
    - in: path
      name: id
      required: true
      schema:
        type: integer
  get:
    summary: Obtém um passo de solução pelo ID
    security:

```

```

    - bearerAuth: []
parameters:
  - in: path
    name: exercise_id
    required: true
    schema:
      type: integer
  - in: path
    name: id
    required: true
    schema:
      type: integer
responses:
  '200':
    description: Detalhes do passo de solução
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/SolutionStep'
  '401':
    description: Signature has expired
  '404':
    description: Passo de solução não encontrado

put:
  summary: Atualiza um passo de solução
  security:
    - bearerAuth: []
parameters:
  - in: path
    name: exercise_id
    required: true
    schema:
      type: integer
  - in: path
    name: id
    required: true
    schema:
      type: integer
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/SolutionStep'
responses:

```

```

'202':
  description: Passo de solução atualizado
'401':
  description: Signature has expired
'404':
  description: Passo de solução não encontrado
'422':
  description: Erro de validação
  content:
    application/json:
      schema:
        type: object
        properties:
          message:
            type: string
            example: "Alguns erros foram encontrados, por favor
verifique."
          solutionStep:
            $ref: '#/components/schemas/SolutionStep'
          errors:
            type: object
            properties:
              title:
                type: array
                items:
                  type: string
                  example: "não pode ficar em branco"
              description:
                type: array
                items:
                  type: string
                  example: "não pode ficar em branco"

delete:
  summary: Remove um passo de solução
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: exercise_id
      required: true
      schema:
        type: integer
    - in: path
      name: id
      required: true

```

```

        schema:
            type: integer
responses:
    '200':
        description: Passo de solução removido
    '401':
        description: Signature has expired
    '404':
        description: Passo de solução não encontrado

/api/professors/los/{lo_id}/exercises/{exercise_id}/solution_steps/{id}/duplicate:
    parameters:
        - in: path
          name: lo_id
          required: true
          schema:
              type: integer
        - in: path
          name: exercise_id
          required: true
          schema:
              type: integer
        - in: path
          name: id
          required: true
          schema:
              type: integer
    post:
        summary: Duplica um passo de solução
        security:
            - bearerAuth: []
        parameters:
            - in: path
              name: exercise_id
              required: true
              schema:
                  type: integer
            - in: path
              name: id
              required: true
              schema:
                  type: integer
        responses:
            '201':
                description: Passo de solução duplicado

```

```

    '401':
      description: Signature has expired
    '404':
      description: Passo de solução não encontrado

/api/professors/los/{lo_id}/exercises/{exercise_id}/solution_steps/reorder:
  parameters:
    - in: path
      name: lo_id
      required: true
      schema:
        type: integer
    - in: path
      name: exercise_id
      required: true
      schema:
        type: integer
  patch:
    summary: Reordena os passos de solução de um exercício
    security:
      - bearerAuth: []
    parameters:
      - in: path
        name: exercise_id
        required: true
        schema:
          type: integer
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              solution_steps_ids:
                type: array
                items:
                  type: integer
                description: Lista ordenada dos IDs dos passos de solução
                para definir a nova ordem
    responses:
      '200':
        description: Passos de solução reordenados com sucesso
      '401':
        description: Signature has expired
      '404':

```


description: Exercício não encontrado

/api/professors/los/{lo_id}/exercises/{exercise_id}/solution_steps/{solution_step_id}/tips:

parameters:

- in: path
name: lo_id
required: true
schema:
type: integer
- in: path
name: exercise_id
required: true
schema:
type: integer
- in: path
name: solution_step_id
required: true
schema:
type: integer

get:

summary: Lista todas as dicas de um passo de solução

security:

- bearerAuth: []

parameters:

- in: path
name: solution_step_id
required: true
schema:
type: integer

responses:

'200':

description: Lista de dicas

content:

application/json:

schema:

type: array

items:

\$ref: '#/components/schemas/Tip'

'401':

description: Signature has expired

post:

summary: Cria uma nova dica para um passo de solução

security:

```

- bearerAuth: []
parameters:
- in: path
  name: solution_step_id
  required: true
  schema:
    type: integer
requestBody:
  required: true
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/Tip'
responses:
  '201':
    description: Dica criada
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Tip'
  '401':
    description: Signature has expired
  '422':
    description: Erro de validação
    content:
      application/json:
        schema:
          type: object
          properties:
            message:
              type: string
              example: "Alguns erros foram encontrados, por favor
verifique."
            tip:
              $ref: '#/components/schemas/Tip'
          errors:
            type: object
            properties:
              description:
                type: array
                items:
                  type: string
                  example: "não pode ficar em branco"

```

/api/professors/los/{lo_id}/exercises/{exercise_id}/solution_steps/{solution_step_id}/tips/{id}:

parameters:

- in: path
name: lo_id
required: true
schema:
type: integer
- in: path
name: exercise_id
required: true
schema:
type: integer
- in: path
name: solution_step_id
required: true
schema:
type: integer
- in: path
name: id
required: true
schema:
type: integer

get:

summary: Obtém uma dica pelo ID

security:

- bearerAuth: []

parameters:

- in: path
name: solution_step_id
required: true
schema:
type: integer
- in: path
name: id
required: true
schema:
type: integer

responses:

'200':

description: Detalhes da dica

content:

application/json:

schema:

\$ref: '#/components/schemas/Tip'

```

    '401':
      description: Signature has expired
    '404':
      description: Dica não encontrada

put:
  summary: Atualiza uma dica
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: solution_step_id
      required: true
      schema:
        type: integer
    - in: path
      name: id
      required: true
      schema:
        type: integer
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Tip'
  responses:
    '202':
      description: Dica atualizada
    '401':
      description: Signature has expired
    '404':
      description: Dica não encontrada
    '422':
      description: Erro de validação
      content:
        application/json:
          schema:
            type: object
            properties:
              message:
                type: string
                example: "Alguns erros foram encontrados, por favor
verifique."
            tip:
              $ref: '#/components/schemas/Tip'

```

```

        errors:
          type: object
          properties:
            description:
              type: array
            items:
              type: string
            example: "não pode ficar em branco"

delete:
  summary: Remove uma dica
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: solution_step_id
      required: true
      schema:
        type: integer
    - in: path
      name: id
      required: true
      schema:
        type: integer
  responses:
    '200':
      description: Dica removida
    '401':
      description: Signature has expired
    '404':
      description: Dica não encontrada

/api/professors/los/{lo_id}/exercises/{exercise_id}/solution_steps/{solution_step_id}/tips/{id}/duplicate:
  parameters:
    - in: path
      name: lo_id
      required: true
      schema:
        type: integer
    - in: path
      name: exercise_id
      required: true
      schema:
        type: integer
    - in: path

```

```

      name: solution_step_id
      required: true
      schema:
        type: integer
    - in: path
      name: id
      required: true
      schema:
        type: integer
  post:
    summary: Duplica uma dica de um passo de solução
    security:
      - bearerAuth: []
    parameters:
      - in: path
        name: solution_step_id
        required: true
        schema:
          type: integer
      - in: path
        name: id
        required: true
        schema:
          type: integer
    responses:
      '201':
        description: Dica duplicada com sucesso
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Tip'
      '401':
        description: Signature has expired
      '404':
        description: Dica ou passo de solução não encontrado

```

```

/api/professors/{lo_id}/exercises/{exercise_id}/solution_steps/{solution_step_id}/tips/reorder:

```

```

  parameters:
    - in: path
      name: lo_id
      required: true
      schema:
        type: integer
    - in: path
      name: exercise_id

```

```

    required: true
    schema:
      type: integer
  - in: path
    name: solution_step_id
    required: true
    schema:
      type: integer
patch:
  summary: Reordena as dicas de um passo de solução
  security:
    - bearerAuth: []
  parameters:
    - in: path
      name: solution_step_id
      required: true
      schema:
        type: integer
  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          properties:
            tips_ids:
              type: array
              items:
                type: integer
              description: Lista ordenada dos IDs das dicas para
definir a nova ordem
  responses:
    '200':
      description: Dicas reordenadas com sucesso
    '401':
      description: Signature has expired
    '404':
      description: Passo de solução não encontrado

components:
  securitySchemes:
    bearerAuth:
      type: http
      scheme: bearer
      bearerFormat: JWT

```

description: Bearer TOKEN

schemas:

Lo:

type: object

required:

- title
- description

properties:

id:

type: integer

title:

type: string

minLength: 1

description:

type: string

minLength: 1

image:

type: string

format: binary

description: Imagem representativa do objeto de aprendizagem

exercises_count:

type: integer

description: Cache de quantos exerc cios existem nesse objeto de

aprendizagem

introductions_count:

type: integer

description: Cache de quantas introdu  es existem nesse objeto de

aprendizagem

created_at:

type: string

format: date-time

updated_at:

type: string

format: date-time

Introduction:

type: object

required:

- title
- description

properties:

id:

type: integer

title:

type: string


```

    minLength: 1
  description:
    type: string
    minLength: 1
  public:
    type: boolean
  position:
    type: integer
  lo_id:
    type: integer
    description: ID do objeto de aprendizagem ao qual esta introdução
pertence
  created_at:
    type: string
    format: date-time
  updated_at:
    type: string
    format: date-time

```

Exercise:

```

  type: object
  required:
    - title
    - description
  properties:
    id:
      type: integer
    title:
      type: string
      minLength: 1
    description:
      type: string
      minLength: 1
    public:
      type: boolean
    position:
      type: integer
    solution_steps_count:
      type: integer
      description: Cache de quantos passos de solução existem nesse
exercício
    lo_id:
      type: integer
      description: ID do objeto de aprendizagem ao qual esta introdução
pertence
    created_at:

```

```

    type: string
    format: date-time
  updated_at:
    type: string
    format: date-time

SolutionStep:
  type: object
  required:
    - title
    - description
  properties:
    id:
      type: integer
    title:
      type: string
      minLength: 1
    description:
      type: string
      minLength: 1
    response:
      type: string
    decimal_digits:
      type: integer
    public:
      type: boolean
    tips_display_mode:
      type: string
      description: Maneira que as dicas devem ser exibidas, Ex.
        Sequencial
    position:
      type: integer
    tips_count:
      type: integer
      description: Cache de quantas dicas existem nesse passos de
        solução
    exercise_id:
      type: integer
      description: ID do exercício ao qual este passo de solução
        pertence
    created_at:
      type: string
      format: date-time
    updated_at:
      type: string
      format: date-time

```

Tip:

```
type: object
required:
  - description
properties:
  id:
    type: integer
  description:
    type: string
    minLength: 1
  number_attempts:
    type: integer
  position:
    type: integer
  exercise_id:
    type: integer
    description: ID do passo de solução ao qual esta dica pertence
  created_at:
    type: string
    format: date-time
  updated_at:
    type: string
    format: date-time
```

User:

```
type: object
properties:
  id:
    type: integer
  email:
    type: string
  name:
    type: string
  created_at:
    type: string
    format: date-time
  updated_at:
    type: string
    format: date-time
  jti:
    type: string
```