

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MATHEUS ROCHA CAMARGO

**DESENVOLVIMENTO DA VISUALIZAÇÃO DE OBJETOS DE APRENDIZAGEM
NA FERRAMENTA FARMA**

GUARAPUAVA

2025

MATHEUS ROCHA CAMARGO

**DESENVOLVIMENTO DA VISUALIZAÇÃO DE OBJETOS DE APRENDIZAGEM
NA FERRAMENTA FARMA**

Development of Visualizationalization of Learning Objects at FARMA

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

Orientador: Prof. Dr. Diego Marczal

Coorientador: Prof^a. Dr^a. Renata Luiza Stange Carneiro Gomes

GUARAPUAVA

2025



[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.

LISTA DE ABREVIATURAS E SIGLAS

Siglas

API	Application Programming Interface
FARMA	Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem
OA	Objeto de Aprendizagem
TCC	Trabalho de Conclusão de Curso

SUMÁRIO

1	INTRODUÇÃO	3
1.1	Objetivo Geral	3
1.2	Justificativa	4
2	CONTEXTUALIZAÇÃO	5
3	PROPOSTA	9
3.1	Representação visual dos OAs	9
3.2	Etapas de desenvolvimento	13
4	CONSIDERAÇÕES FINAIS	16
	REFERÊNCIAS	17

1 INTRODUÇÃO

Atualmente, diversas ferramentas buscam aprimorar o processo de aprendizagem, dentre essas ferramentas estão os materiais didáticos disponibilizados na web, como videoaulas, plataformas de ensino e e-books. Muitas dessas ferramentas oferecem recursos para que os alunos aprendam com seus próprios erros, mesmo que de forma indireta, ao proporcionar revisões e repetições de conteúdo. No entanto, não é comum ver esses mesmos materiais destacando o erro como o elemento principal do processo de aprendizagem. A partir do registro, visualização e manipulação dos erros cometidos pelos alunos, a Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem (FARMA) permite ao estudante retroceder ao contexto em que o erro ocorreu, oferecendo ao aluno a possibilidade de refletir diretamente sobre suas falhas, promovendo assim um melhor aproveitamento dos conteúdos (MARCZAL, 2014).

Objeto de Aprendizagem (OA)s, materiais didáticos assistidos por computador, servem ao propósito de utilizar os erros como elementos ativos no processo de aprendizagem, através da retroação. A FARMA, é um projeto que atua no cenário de aprendizagem baseada em erros, utilizando OAs como suporte para o ensino de conceitos matemáticos, assim permitindo que professores personalizem suas abordagens didáticas e que alunos recebam um *feedback* individualizado para corrigir desvios na formação de conceitos, promovendo um ciclo de aprendizado mais eficiente e interativo (MARCZAL, 2014).

Desde sua concepção, a ferramenta FARMA tem sido aprimorada gradualmente por meio de diversos trabalhos acadêmicos desenvolvidos ao longo dos anos. Entre os avanços realizados estão: a implementação da visualização de OAs (RAMOS, 2019), o desenvolvimento do módulo de estatística da FARMA (SANTOS, 2022), o desenvolvimento da área de criação de OAs (VITEK, 2023), a construção de uma Application Programming Interface (API) para comunicação com esses objetos (BONIN, 2024) e a implementação de uma nova interface gráfica para criação de OAs (RIBAS, 2024). Apesar desses avanços, ainda há pontos a serem aprimorados, como a visualização de Objetos de Aprendizagem, que ainda não está completamente finalizada, especialmente no que diz respeito à exibição estruturada dos conteúdos e à usabilidade do usuário durante o processo de aprendizagem.

Este trabalho propõe a continuidade do trabalho desenvolvido por Bonin (2024) e o aperfeiçoamento do trabalho de Ramos (2019), com foco na visualização dos Objetos de Aprendizagem na ferramenta FARMA. Para isso, serão elaborados e implementados os protótipos das interfaces necessárias para exibir as diferentes etapas de um OA.

1.1 Objetivo Geral

Implementar a funcionalidade de visualização dos Objetos de Aprendizagem (OA) na nova versão da ferramenta de autoria FARMA.

1.2 Justificativa

A realização desse trabalho se justifica pela necessidade da implementação da visualização do OA na ferramenta FARMA. Segundo Nielsen (1994), uma interface mal projetada pode comprometer a compreensão e a retenção de informações, afetando diretamente o processo de aprendizagem. Sabendo disso, a visualização do OA será realizada seguindo as diretrizes atuais de uma boa usabilidade.

A visualização adequada dos OAs é essencial para que o aluno compreenda o conteúdo apresentado, acompanhe seu próprio progresso de resolução e receba o retorno (retroação) sobre seus erros de maneira clara e contextualizada. Na ferramenta FARMA, o OA envolve a apresentação da introdução teórica, a divisão dos exercícios em passos e a exibição dos feedbacks definidos pelo professor em cada etapa. Sem uma interface clara que apresente esses elementos, o potencial pedagógico da remediação de erros se perde, comprometendo o objetivo central da ferramenta.

Outro ponto é a oportunidade de aplicar os conhecimentos adquiridos ao longo do curso em um projeto relevante, projeto este que pode ser utilizado por professores e alunos em situações reais de ensino. O desafio técnico de implementar interfaces intuitivas e bem estruturadas contribui para formar um profissional mais completo e alinhado com as práticas do mercado.

Por se tratar de um projeto que está evoluindo constantemente, o desenvolvimento da visualização dos OAs pode abrir espaço para novos Trabalho de Conclusão de Curso (TCC)s que podem estar voltados à acessibilidade digital, testes de usabilidade, gamificação ou até aplicação da ferramenta em outras disciplinas que não a matemática.

Quanto à viabilidade técnica do projeto, a existência de documentação, trabalhos anteriores e o acesso aos códigos da versão antiga da FARMA garantem uma base sólida para o desenvolvimento e testes da proposta.

2 CONTEXTUALIZAÇÃO

Pelo do avanço constante das tecnologias utilizadas no desenvolvimento de software, realizar a manutenção em sistemas que não acompanham essas mudanças é cada vez mais desafiador. Projetos que permanecem longos períodos sem atualizações tendem a sofrer com acúmulo de débitos técnicos, bibliotecas sem contribuições há meses ou anos, falhas de segurança e dificuldades de integração com novas ferramentas atualizadas do mercado. Todos esses pontos não apenas dificultam o desenvolvimento, como também impactam negativamente a qualidade, a usabilidade e a manutenibilidade do sistema a longo prazo.

Segundo Li (2014), a defasagem tecnológica de um projeto acontece por conta de aspectos invisíveis relacionados à evolução do ecossistema dos softwares. Como tecnologias que caem em desuso, correções e melhorias que são aplicadas naquelas que já existem, troca por outras tecnologias mais seguras e modernas e assim por diante. Esses aspectos fazem com que os projetos que não recebem manutenção contínua fiquem defasados e percam sua compatibilidade com ferramentas modernas. Além disso, novos desenvolvedores que integram sistemas já em andamento têm um esforço maior para entender o escopo de uma tecnologia que já não é mais tão utilizada e possuem maior risco de introdução de erros durante manutenções corretivas ou evolutivas.

No contexto do desenvolvimento web, esse problema se agrava pela velocidade com que frameworks JavaScript evoluem. O React, por exemplo, recebe atualizações frequentes que mudam a forma como componentes são escritos. A versão utilizada no trabalho de Ramos (2019) era baseada no React 16, onde a estrutura de componentes de classe ainda era o padrão na época, e muitos dos recursos modernos, como hooks, ainda estavam começando.

Abaixo, é extraído o componente App, componente raiz e ponto inicial de uma aplicação React, diretamente do repositório do projeto FARMA Reborn, uma reimplementação da ferramenta de autoria FARMA. A seguir, é feita uma comparação e mostro como esse mesmo componente poderia ser reescrito hoje, com React 19.

```
1 import React, { Component } from 'react';
2 import PropTypes from 'prop-types';
3 import store from './store';
4 import Router from './router';
5 import { loadExercise } from './store/actions/index';
6
7 class App extends Component {
8   constructor (props) {
9     super (props);
10    let { exerciseId } = this.props;
11    store.dispatch (loadExercise (exerciseId));
12  }
13
14  render () {
15    return <Router store={store} />;
16  }
17 }
18
19 App.propTypes = {
20   exerciseId: PropTypes.number.isRequired,
21 }
22
23 export default App;
```

Listing 2.1 – Código retirado do projeto Farma Reborn (React 16 - 2019)

Observações relacionadas a esse código comparado a versões atuais:

- Utiliza classe, enquanto o React moderno utiliza funções.
- Instancia e injeta a store de forma manual (hoje usam-se 'Providers')
- Usa PropTypes (ainda é válido, mas TypeScript é mais comum hoje)
- Router é passado como componente diretamente com a store, essa abordagem não é mais necessária com o uso de useSelector/useDispatch.

```
1 import { useEffect } from 'react';
2 import { useDispatch } from 'react-redux';
3 import { loadExercise } from './store/actions';
4 import Router from './router';
5
6 function App({ exerciseId }) {
7   const dispatch = useDispatch();
8
9   useEffect(() => {
10     dispatch(loadExercise(exerciseId));
11   }, [dispatch, exerciseId]);
12
13   return <Router />;
14 }
15
16 export default App;
```

Listing 2.2 – Sugestão de reescrita com React 19 (2025), Redux e hooks

Analisando a possível reescrita, podemos concluir que:

- A classe foi substituída por uma função com `useEffect`, que executa o carregamento quando o componente é montado.
- A store não é mais importada nem enviada como props, aqui assumimos que o `<Provider store=store>` está definido na raiz da aplicação.
- O Router também não precisa receber manualmente a store, já que os componentes internos acessam o estado com o uso da função `useSelector`.

A estrutura atual simplifica o código, melhora a legibilidade e permite maior reutilização de lógica com os hooks personalizados. Atualizar um projeto que foi escrito inteiramente com base em componentes de classe requer não apenas reescrever partes significativas do código, mas também adaptar bibliotecas, ajustando comportamentos de ciclo de vida, gerenciamento de estado e integração com outras dependências modernas como o React Router e o Redux. Assim, a defasagem tecnológica se torna um dos principais impedimentos para a evolução de projetos como o FARMA Reborn, dificultando sua reutilização como base para novas funcionalidades.

Embora o React moderno simplifique a estrutura dos componentes e promova uma organização mais clara do código, ele ainda depende de ser criado como um projeto à parte, o que pode introduzir uma complexidade desnecessária quando o objetivo não é criar interfaces altamente dinâmicas.

Ainda no aspecto tecnológico, o projeto original utilizava o framework Ruby on Rails na versão 5, que, embora moderno na época, hoje várias versões com novos recursos foram lançadas. Desde então, o framework passou por importantes atualizações, lançando sua versão 8 em dezembro de 2024. Essa evolução trouxe melhorias em termos de segurança, performance, suporte a arquiteturas modernas como APIs RESTful, e um modelo de comunicação assíncrono mais robusto baseado em WebSockets.

Utilizando o Hotwire, um conjunto de ferramentas mantido pela equipe do Rails, a renderização parcial da interface pode ser feita no lado do servidor, utilizando recursos como Turbo Streams e Turbo Frames. Essa solução reduz o acoplamento entre front-end e back-end, permitindo a atualização dinâmica da interface sem a necessidade do uso de JavaScript. A comunicação em tempo real, antes dependia de soluções externas, agora, ela pode ser realizada de forma integrada. Isso significa que manter a estrutura baseada no Rails 5 impede o uso nativo desses novos recursos.

Dessa forma, entende-se que tanto no front-end quanto no back-end, a base tecnológica da primeira reimplantação da FARMA não atende mais às demandas atuais do ecossistema de desenvolvimento. Fazer essa atualização manualmente na aplicação para acompanhar essas mudanças seria tão custoso e arriscado que se justifica a decisão de seguir com um novo projeto, aproveitando as ideias e conceitos pedagógicos, mas reimplantando as tecnologias a partir de padrões modernos.

Ao implementar a funcionalidade de visualização dos OAs com tecnologias atualizadas, espera-se não apenas modernizar a ferramenta, mas também contribuir com o campo educacional ao aplicar conceitos de remediação de erros em um ambiente digital.

A proposta contribui para o conhecimento acadêmico ao integrar aspectos de desenvolvimento web, design centrado no usuário e educação matemática, oferecendo uma abordagem prática sobre como o erro pode ser explorado pedagogicamente em sistemas interativos. Do ponto de vista técnico, o trabalho serve como um estudo sobre atualização tecnológica de sistemas legados, abordando os impactos da defasagem tecnológica de frameworks e a importância da manutenção contínua. Espera-se que assim, este trabalho sirva como base para futuras pesquisas e extensões da ferramenta, fomentando a continuidade do projeto e sua adoção mais ampla na prática docente.

3 PROPOSTA

A ferramenta FARMA foi criada com o objetivo de auxiliar no ensino de matemática por meio da criação de Objetos de Aprendizagem, que utilizam a remediação de erros como estratégia de aprendizado. Desde sua criação, a ferramenta passou por diversas melhorias, como a primeira implementação da visualização de OAs (RAMOS, 2019), a criação de uma API para interagir com OA (BONIN, 2024). No entanto, a parte responsável pela visualização do OA, que é onde o estudante terá contato direto com o conteúdo criado pelo professor, ainda permanece com possibilidades de melhoria.

Além disso, a base de código do projeto anterior utiliza versões antigas de tecnologias como o Ruby on Rails 5, React 16 e o Webpacker, todas descontinuadas nos dias de hoje. O Webpacker em especial, foi removido oficialmente a partir do Rails 7 (RAILS, 2025), tornando a inserção da biblioteca React na sua versão atual ainda mais difícil.

Para resolver essas limitações, este trabalho propõe implementar a funcionalidade de visualização de Objetos de Aprendizagem na nova versão da ferramenta FARMA. Isso será realizado utilizando tecnologias mais atuais e compatíveis com o ecossistema Ruby on Rails, que possibilitem maior integração entre os componentes do sistema e menor dependência de bibliotecas descontinuadas.

3.1 Representação visual dos OAs

A nova implementação não será feita do zero, mas sim como uma continuidade dos trabalhos realizados por Bonin (2024), que estruturou as rotas de API para a visualização dos OAs, e Ramos (2019), que idealizou o primeiro modelo de visualização. O presente trabalho terá como foco principal a reformulação das interfaces responsáveis pela exibição dos conteúdos de um OA, que incluem:

- Introduções teóricas;
- Navegação entre os passos de resolução dos exercícios;
- Indicadores de progresso do aluno.
- Exibição de mensagens de acerto e erro (retroação);
- Exibição de dicas;

Essa visualização será baseada nas interfaces do projeto FARMA Reborn. Essas interfaces representam o funcionamento esperado da visualização de um OA na prática. A seguir, são descritas as principais telas com suas respectivas funcionalidades que compõem essa visualização.

A Figura 1 exibe a estrutura de um exercício de um OA. O conteúdo é organizado em blocos que apresentam um problema, seguido de uma questão que propõe a resolução com base nos dados fornecidos. Esse tipo de apresentação busca promover a compreensão gradativa do conceito trabalhado.

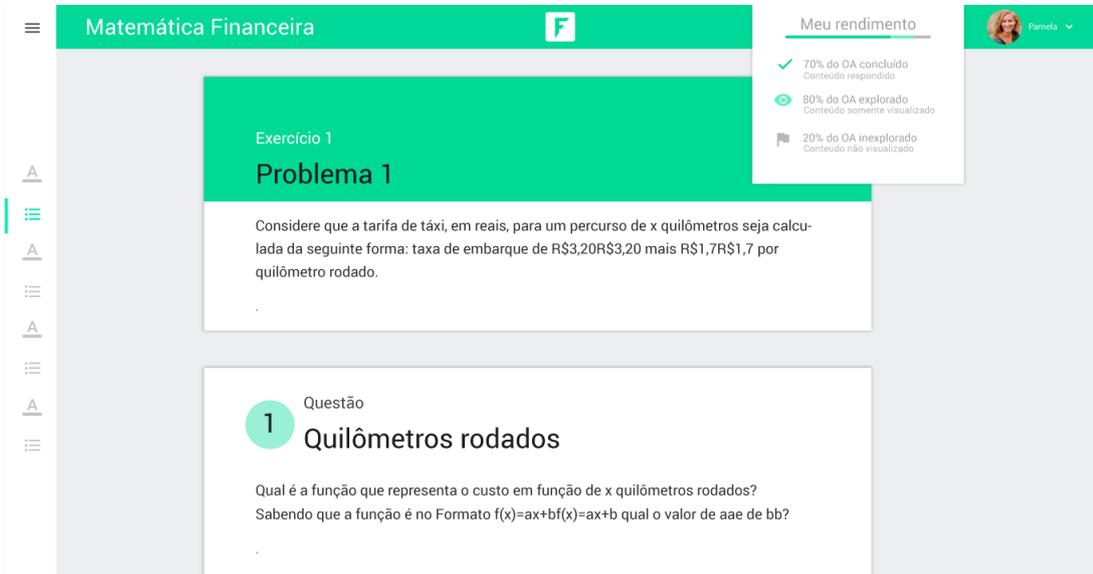


Figura 1 – Tela de exercício com enunciado e questão

A Figura 2 demonstra a introdução teórica de um conteúdo. Nesse exemplo, o conceito de função é apresentado com apoio de um texto explicativo e observações organizadas por tópicos. Essa introdução tem o papel de preparar o estudante para os exercícios, reforçando o entendimento teórico necessário.



Figura 2 – Tela de introdução teórica de um OA

Na Figura 3, é possível visualizar a navegação entre os elementos do OA, incluindo introduções e exercícios organizados por etapas. Essa estrutura permite ao estudante visualizar o seu progresso e acessar os conteúdos de forma lógica. No canto superior direito, um painel de desempenho mostra estatísticas relacionadas ao completamento do OA pelo estudante.

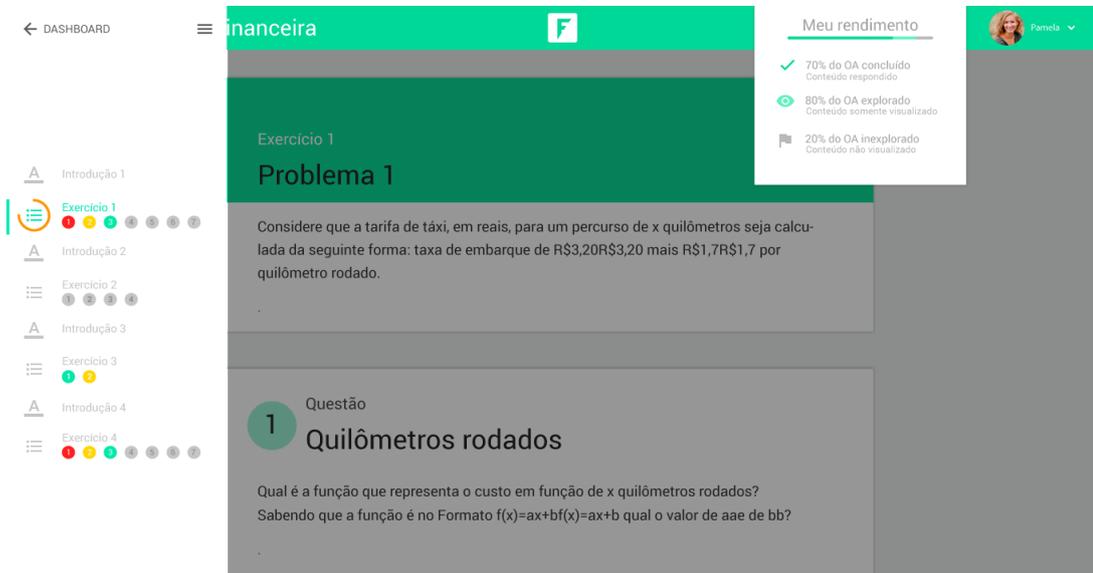


Figura 3 – Navegação entre introduções e exercícios, com painel de desempenho do aluno

As Figuras 4 e 5 ilustram a tela de resposta do exercício, onde o aluno interage com o sistema por meio de um teclado virtual. Esse teclado disponibiliza operadores e símbolos matemáticos organizados por categorias como funções, matrizes e trigonometria. A interface também inclui um campo de resposta e botões para envio e limpeza, promovendo uma interação direta com o exercício proposto.

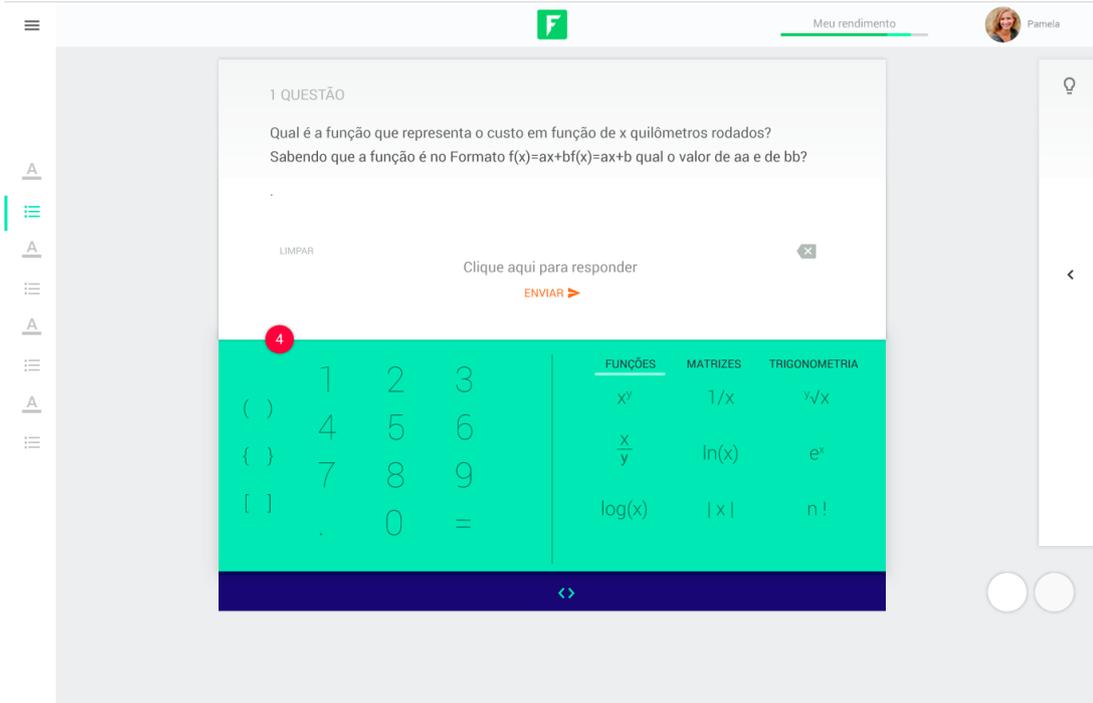


Figura 4 – Tela de resposta com teclado virtual (categoria: Funções)

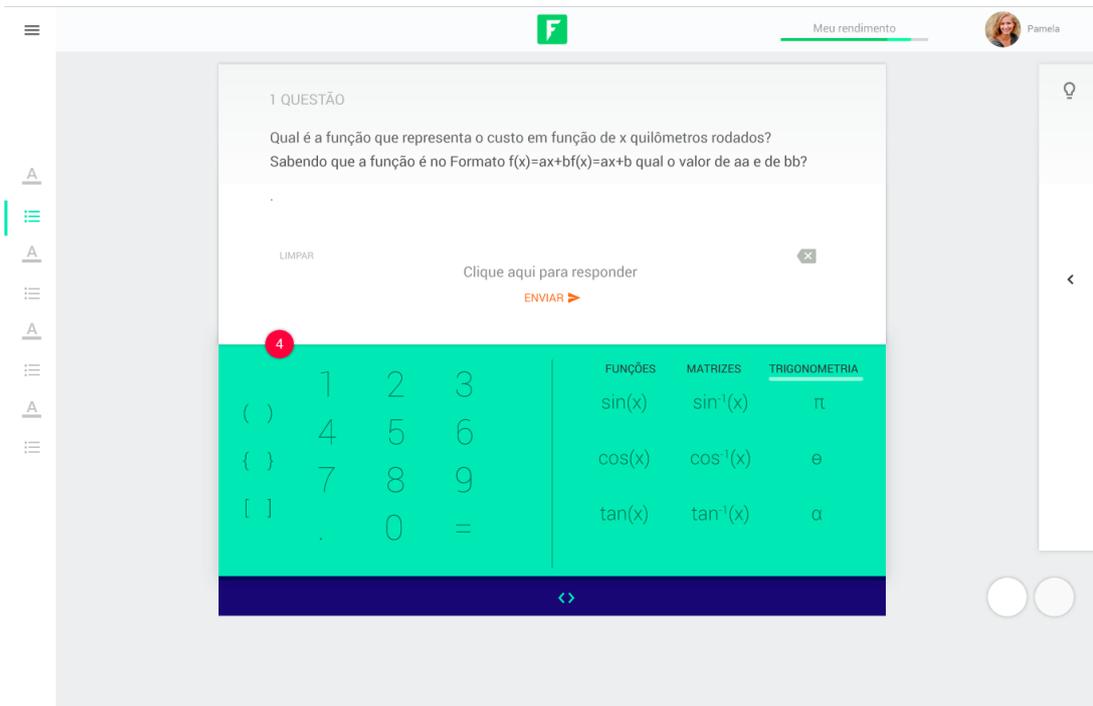


Figura 5 – Teclado virtual com operadores de trigonometria

Por fim, a Figura 6 mostra o painel lateral de dicas. As dicas são exibidas ao lado da questão, possibilitando ao estudante consultar orientações definidas previamente pelo professor, sem sair do ambiente da questão. Essa funcionalidade está diretamente ligada à proposta de retroação, pois permite que o aluno receba orientações baseadas em seus erros ou dificuldades.

The screenshot shows a user interface for an online learning platform. At the top, there is a navigation bar with a menu icon, a logo 'F', and a user profile 'Pamela'. The main content area contains a question: 'Qual é a função que representa o custo em função de x quilômetros rodados? Sabendo que a função é no Formato $f(x)=ax+bf(x)=ax+b$ qual o valor de a e de b ?' Below the question is a text input field with a 'LIMPAR' button and an 'ENVIAR' button. A calculator interface is visible, featuring a numeric keypad (0-9, ., =) and a function keypad with categories: FUNÇÕES (sin(x), cos(x), tan(x)), MATRIZES (sin⁻¹(x), cos⁻¹(x), tan⁻¹(x)), and TRIGONOMETRIA (π, e, o). A red circle with the number '4' is placed over the calculator keypad. To the right of the calculator is a 'DICAS' (Tips) panel with a yellow header and a box containing a '1º DICA' (1st Tip) with placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercita-'. Navigation arrows are visible at the bottom right of the calculator area.

Figura 6 – Exibição de dica contextual ao lado da questão

Esses exemplos representam os principais elementos que compõem a visualização de um OA: introduções, exercícios, estrutura de navegação, interação via teclado e exibição de dicas. A proposta deste trabalho visa implementar essas visualizações utilizando tecnologias atualizadas, conforme discutido na seção anterior, eliminando o impedimento causado pela defasagem tecnológica.

3.2 Etapas de desenvolvimento

A implementação da visualização do OA será realizada em etapas sequenciais, com o objetivo de garantir a qualidade do resultado final. A seguir, são descritas cinco etapas que compõem o plano de execução do trabalho:

1. Análise da Estrutura Atual dos OAs

Será realizada uma análise da estrutura já existente dos OAs na ferramenta, com foco em entender como os dados são organizados e apresentados atualmente. Essa etapa visa mapear quais elementos devem ser exibidos e quais fluxos de interação precisam ser representados visualmente na nova implementação.

2. Definição de requisitos

Com base na análise inicial, serão definidos os requisitos funcionais necessários para a visualização dos OAs. Isso inclui o levantamento de fluxos como: visualizar introduções, acessar exercícios passo a passo, exibir dicas, apresentar mensagens de acerto ou erro, e registrar as ações do usuário. Os requisitos servirão como base para a etapa de prototipação.

3. Prototipação das interfaces

Serão criadas representações visuais (protótipos) das telas que compõem a visualização dos OAs. Esses protótipos servirão como guia para o desenvolvimento das interfaces, permitindo antecipar a organização dos elementos, validar os fluxos de interação e garantir o alinhamento com o objetivo pedagógico da ferramenta.

4. Implementação das interfaces

Com base nos protótipos aprovados, serão desenvolvidas as interfaces responsáveis por exibir cada elemento do OA. O desenvolvimento será realizado utilizando tecnologias compatíveis com as versões mais recentes do Ruby on Rails. A renderização das páginas será feita prioritariamente no servidor, utilizando templates parciais (*partials*) para garantir consistência visual e organização do código. Para formulários e entradas do usuário, serão empregados os *form helpers* nativos do framework, que facilitam a comunicação com o backend sem a necessidade de bibliotecas externas adicionais. Além disso, serão utilizadas técnicas de atualização assíncrona da interface por meio de requisições leves ao servidor, permitindo transições e interações sem recarregamento completo da página. Essa abordagem elimina a necessidade de frameworks JavaScript complexos e favorece uma experiência mais fluida com modelo tradicional do Rails.

5. Validações e entregas

Durante o processo de implementação, serão realizadas validações parciais junto ao orientador, a fim de garantir que as funcionalidades estejam alinhadas com os objetivos do trabalho. A proposta será desenvolvida de forma iterativa, possibilitando ajustes progressivos ao longo da execução.

O público beneficiado por este trabalho inclui os estudantes que utilizarão os OAs como ferramenta de apoio no processo de ensino. Além disso, o projeto também serve como base para os próximos desenvolvedores que desejem contribuir com a FARMA, seja em novos TCCs ou iniciações científicas. Ao adotar tecnologias atualizadas, o trabalho tende a evitar a defa-

sagem tecnológica, promovendo a continuidade do desenvolvimento da FARMA de forma mais sustentável.

4 CONSIDERAÇÕES FINAIS

Este trabalho tem como objetivo implementar a funcionalidade de visualização dos Objetos de Aprendizagem (OAs) na nova versão da ferramenta de autoria FARMA. Para isso, serão desenvolvidas interfaces que permitam a exibição estruturada dos elementos que compõem um OA, como introduções, exercícios, passos e mensagens de retroação.

A proposta se destaca por dar continuidade ao desenvolvimento de uma ferramenta já consolidada no contexto acadêmico, com múltiplos estudos, TCCs e doutorados realizados ao longo dos anos. Ao aprimorar a visualização dos OAs, o projeto busca proporcionar ao aluno um ambiente mais didático, onde ele possa acompanhar seu próprio desempenho e utilizar seus erros como parte do processo de construção do conhecimento, sem que isso represente uma barreira. Além disso, a adoção de tecnologias atualizadas fortalece a base técnica do sistema e contribui para sua manutenção a longo prazo.

Espera-se que os resultados obtidos com este trabalho contribuam tanto para o avanço técnico da ferramenta quanto para sua aplicação prática em contextos reais de ensino. A nova visualização poderá ser utilizada por alunos e professores no dia a dia, além de servir como base para futuras contribuições acadêmicas, incluindo trabalhos voltados à acessibilidade, testes de usabilidade, ou extensão do sistema para outras áreas do conhecimento além da matemática.

Durante o desenvolvimento, alguns desafios poderão surgir, como a adaptação dos dados retornados pela APIs existente às novas estruturas visuais, e a necessidade de garantir que as interfaces reflitam com fidelidade a lógica pedagógica estabelecida para os OAs. Também se considera como limitação o tempo disponível para implementação e validação completa de todas as funcionalidades previstas. Ainda assim, a documentação existente, os trabalhos anteriores e o acesso à base de código atual oferecem um suporte técnico viável para a execução da proposta dentro dos prazos previstos.

REFERÊNCIAS

- BONIN, D. **IMPLEMENTAÇÃO DE UMA API PARA VISUALIZAÇÃO E INTERAÇÃO COM OBJETOS DE APRENDIZAGEM PARA FERRAMENTA DE AUTORIA FARMA**. 2024. Disponível em: https://tcc.tsi.pro.br/uploads/academic_activity/pdf/273/GP_COINT_2024_1_DOUGLAS_VINICIUS_CALDAS_BONIN_MONOGRAFIA.pdf.
- LI, Z. **A systematic mapping study on technical debt and its management**. 2014. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0164121214002854>. Acesso em: 05 may. 2025.
- MARCZAL, D. **FARMA: UMA FERRAMENTA DE AUTORIA PARA OBJETOS DE APRENDIZAGEM DE CONCEITOS MATEMÁTICOS**. 2014. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2014. Disponível em: <https://acervodigital.ufpr.br/handle/1884/37363>.
- NIELSEN, J. **Usability Engineering**. Morgan Kaufmann Publishers, 1994. ISBN 0125184069. Disponível em: <https://www.nngroup.com/books/usability-engineering/>. Acesso em: 06 may. 2025.
- RAMOS, J. **VISUALIZAÇÃO DE OBJETOS DE APRENDIZAGEM PARA A FERRAMENTA DE AUTORIA FARMA**. 2019. Disponível em: https://tcc.tsi.pro.br/uploads/academic_activity/pdf/19/GP_COINT_2019_2_JEFFERSON_HENRIQUE_RAMOS_MONOGRAFIA.pdf.
- RIBAS, G. H. **DESENVOLVIMENTO DA INTERFACE GRÁFICA PARA CRIAÇÃO DE OBJETOS DE APRENDIZAGEM NA FERRAMENTA FARMA**. 2024. Disponível em: https://tcc.tsi.pro.br/uploads/academic_activity/pdf/329/GP_COINT_2024_2_GABRIEL_HENRIQUE_RIBAS_PROJETO.pdf. Acesso em: 06 may. 2025.
- SANTOS, A. **DESENVOLVIMENTO DO MÓDULO DE ESTATÍSTICAS DA FERRAMENTA DE AUTORIA DE OBJETOS DE APRENDIZAGEM FARMA**. 2022. Disponível em: https://tcc.tsi.pro.br/uploads/academic_activity/pdf/131/GP_COINT_2022_1_AGUINALDO_GOUDINHO_DOS_SANTOS_MONOGRAFIA.pdf.
- VITEK, I. **DESENVOLVIMENTO DA ÁREA DESTINADA À CRIAÇÃO DE OBJETOS DE APRENDIZAGEM PARA FERRAMENTA DE AUTORIA FARMA**. 2023. Disponível em: https://tcc.tsi.pro.br/uploads/academic_activity/pdf/231/GP_COINT_2023_2_ISABELA_TAQUES_VITEK_MONOGRAFIA.pdf.