

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,
Prof. Dr. Alex Sandro De Castilho

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
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
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
4	CONSIDERAÇÕES FINAIS	14
	REFERÊNCIAS	15

1 INTRODUÇÃO

Atualmente, diversas ferramentas digitais buscam aprimorar o processo de aprendizagem, dentre elas 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 materiais didáticos que tratem o erro como elemento central do processo de aprendizagem, tratando-o como um dado pedagógico ao invés de falhas a serem corrigidas.

Objetos de Aprendizagem (OA) são materiais didáticos assistidos por computador que podem ser aplicados em diversos contextos educacionais. A Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem (FARMA) é uma ferramenta de autoria que permite a criação de OAs com uma característica distinta: a capacidade de utilizar os erros como elementos ativos no processo de aprendizagem através do mecanismo de retroação. Dessa forma, os OAs desenvolvidos com a FARMA possibilitam que professores personalizem suas abordagens didáticas voltadas para o ensino de conceitos matemáticos e que alunos recebam um *feedback* individualizado para corrigir desvios na formação de conceitos, promovendo um ciclo de aprendizado mais iterativo (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 objetos de aprendizagem (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 do Objeto de Aprendizagem (OA) em uma arquitetura monolítica (RIBAS, 2024). A migração para arquitetura monolítica proposta por Ribas (2024) baseou-se na análise das dificuldades enfrentadas em trabalhos anteriores que adotaram sistemas distribuídos, onde a complexidade de manutenção e integração entre componentes resultou em sobrecarga de desenvolvimento sem benefícios proporcionais.

Considerando os avanços já realizados, este trabalho propõe a continuidade do trabalho desenvolvido por Bonin (2024) e o aperfeiçoamento do trabalho de Ramos (2019), ambos com foco na visualização dos OAs na ferramenta FARMA. Para isso, serão elaborados e implementados os protótipos das interfaces necessárias para a visualização das diferentes etapas de um OA, aproveitando a base monolítica estabelecida por Ribas (2024).

1.1 Objetivo Geral

Projetar e desenvolver interfaces gráficas responsáveis pela visualização e interação com Objetos de Aprendizagem (OA) na ferramenta FARMA, utilizando a base de arquitetura monolítica existente e implementando soluções modernas de desenvolvimento web.

1.2 Justificativa

A justificativa para desenvolver este trabalho se dá pela necessidade de implementar a visualização dos OAs na ferramenta FARMA, considerando que essa funcionalidade é necessária para que a plataforma cumpra seu papel pedagógico. Embora a proposta de visualização tenha sido inicialmente abordada por Jefferson Henrique Ramos (RAMOS, 2019), observa-se que o foco do trabalho estava mais concentrado na construção de um teclado virtual, enquanto a proposta atual foca na implementação da visualização dos OAs, utilizando um teclado mais simples e adequado às necessidades do projeto.

Trabalhos subsequentes, como o da Isabela Taques Vitek (VITEK, 2023), adotaram uma arquitetura baseada na separação entre *frontend* e *backend*, fazendo o uso de uma API para comunicação entre esses serviços. Embora essa abordagem esteja alinhada a práticas comuns de desenvolvimento, observou-se na prática que essa separação resultou em uma complexidade adicional e poucas vantagens. A manutenção e integração entre as partes exigiram um esforço significativo, sem um retorno proporcional.

No trabalho de Douglas Vinicius Bonin (BONIN, 2024), foi desenvolvida uma API para a visualização e interação com os OAs. Embora essa implementação tenha avançado em conceitos relacionados a interação com os OAs, este projeto optará por aproveitar essas ideias dentro de uma arquitetura monolítica. Em vez de retornar dados utilizando o formato JavaScript Object Notation (JSON), como em uma API tradicional, a visualização será feita diretamente por meio de páginas que utilizam a linguagem de marcação HyperText Markup Language (HTML) e serão geradas no servidor, mantendo o controle da interface e da lógica de apresentação em um único sistema.

Essa decisão arquitetural é ainda mais justificada considerando que a implementação original de Ramos (2019) utiliza tecnologias que evoluíram significativamente desde sua criação. A modernização completa da base tecnológica existente representaria um esforço desproporcional aos benefícios que seriam obtidos, justificando a decisão de implementar uma nova versão que preserve os conceitos pedagógicos da FARMA, mas utilize padrões tecnológicos atuais.

A decisão por manter uma arquitetura monolítica se baseia no perfil da própria FARMA, que possui um número reduzido de usuários e não requer um nível elevado de escalabilidade. Os trabalhos anteriormente citados indicam que a separação de responsabilidades em múltiplos sistemas causa sobrecarga no desenvolvimento, dificultando a evolução contínua da ferramenta. Dessa forma, a opção por uma aplicação monolítica favorece a manutenção e a continuidade do projeto.

2 CONTEXTUALIZAÇÃO

Pelo 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 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 é exibido como esse mesmo componente poderia ser reescrito atualmente, 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 (atualmente usam-se 'Providers')
- Usa PropTypes (ainda é válido, mas TypeScript é mais comum atualmente)
- O router é utilizado como componente, tendo a store sendo passada como parâmetro e 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 assume-se 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, atualmente 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.

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, 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

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

As novas interfaces serão projetadas de acordo com os elementos que compõem um OA, são eles:

- 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.

As interfaces apresentadas a seguir demonstram a estrutura e organização dos elementos visuais que compõem a visualização de um OA na ferramenta FARMA. O foco das imagens está na disposição da informação, nos padrões de navegação e nos indicadores visuais de progresso.

Cada tela ilustra aspectos funcionais do sistema como: a apresentação do conteúdo teórico e prático, o status das etapas em um OA através de cores específicas, e o teclado virtual para entrada de expressões matemáticas. Estas interfaces servirão como referência para o desenvolvimento da nova implementação. Além disso, é importante ressaltar que o conteúdo matemático exibido nas imagens serve apenas como exemplo, sendo o design e a organização das informações os elementos principais.

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.

The screenshot shows a web interface for 'Matemática Financeira'. At the top, there's a green header with the course name and a logo. On the right, a 'Meu rendimento' (My performance) box shows progress: 70% concluded, 80% explored, and 20% not explored. The main content area is titled 'Exercício 1 Problema 1'. It contains a problem statement about taxi fare and a question asking for a function representing the cost.

Exercício 1
Problema 1

Considere que a tarifa de táxi, em reais, para um percurso de x quilômetros seja calculada da seguinte forma: taxa de embarque de R\$3,20R\$3,20 mais R\$1,7R\$1,7 por quilômetro rodado.

1 Questão
Quilômetros rodados

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 ?

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.

The screenshot shows the same web interface, but now displaying the theoretical introduction to functions. It includes an example about chair production costs and a list of observations defining variables and functions.

Introdução I
Função

Para compreendermos melhor a ideia de função, vamos começar com um exemplo.
Para fabricarmos uma cadeira o custo de produção por unidade será de R\$25,00.

Para produzirmos 1 cadeira o custo será de R\$25,00. Para produzirmos 10 cadeiras o custo será de R\$250,00. Para produzirmos 50 cadeiras o custo será de R\$1250,00. Para produzirmos 100 cadeiras. Se produzirmos " x " cadeiras o custo de produção será $20 \cdot x$.

Observações:

- A quantidade " x " de cadeiras produzidas varia, isto é, podemos chamar " x " de variável independente.
- O custo de produção, que podemos denotar por " y " ou " $f(x)$ ", também varia, mas depende de " x ". Assim, o custo pode ser chamado de variável dependente da quantidade produzida.
- Normalmente a variável independente é representada pelo x e a variável dependente é representada pelo y ou $f(x)$.

Podemos definir uma função da seguinte forma:

Figura 2 – Tela de introdução teórica de um OA

Na Figura 3, do lado esquerdo, é possível visualizar a navegação entre os elementos do OA, incluindo introduções e exercícios organizados por etapas. Cada cor dentro do item de exercício representa o status daquela questão, sendo vermelho quando o aluno erra a questão, amarelo quando visualizado e não respondido e verde quando respondida de forma correta. 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.

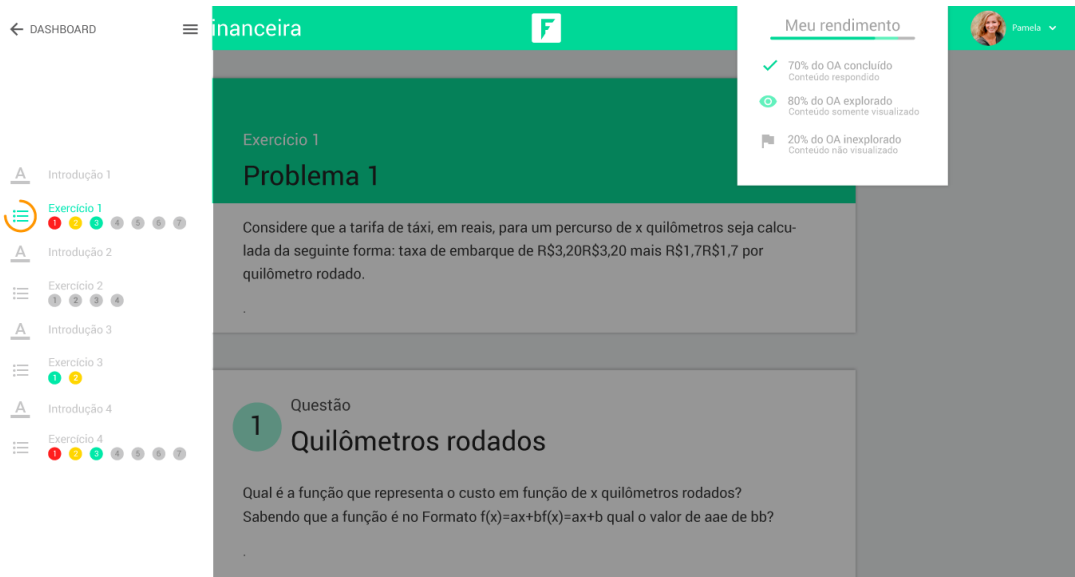


Figura 3 – Navegação entre introduções e exercícios, com painel de desempenho do aluno

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.

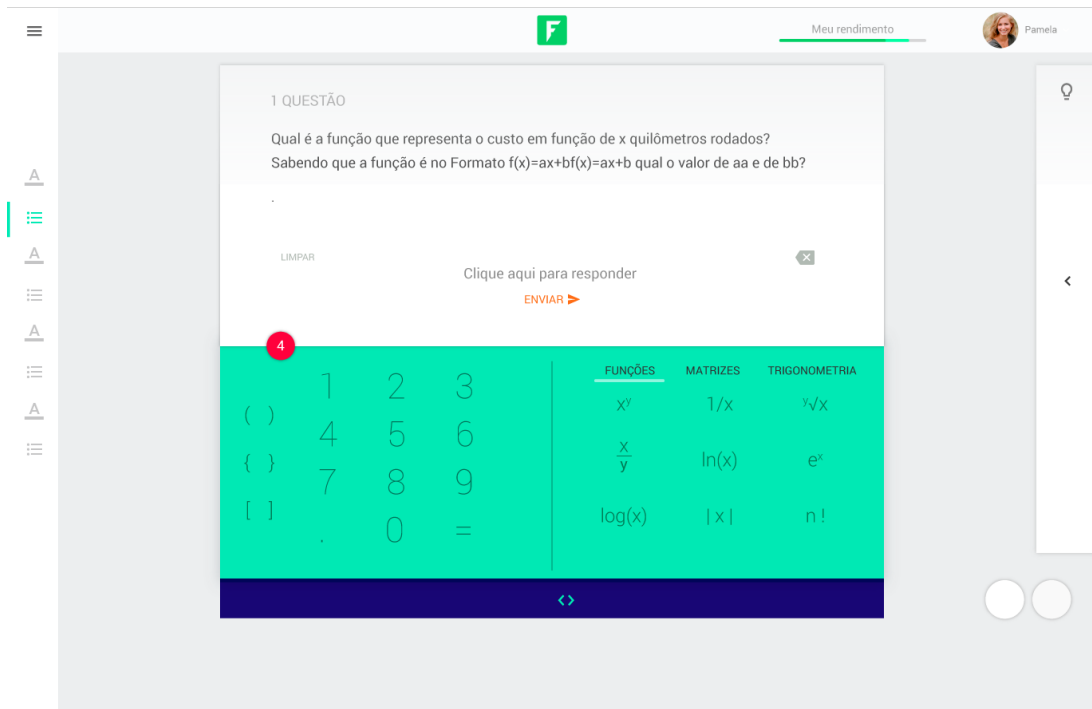


Figura 4 – Tela de resposta com teclado virtual (categoria: Funções)

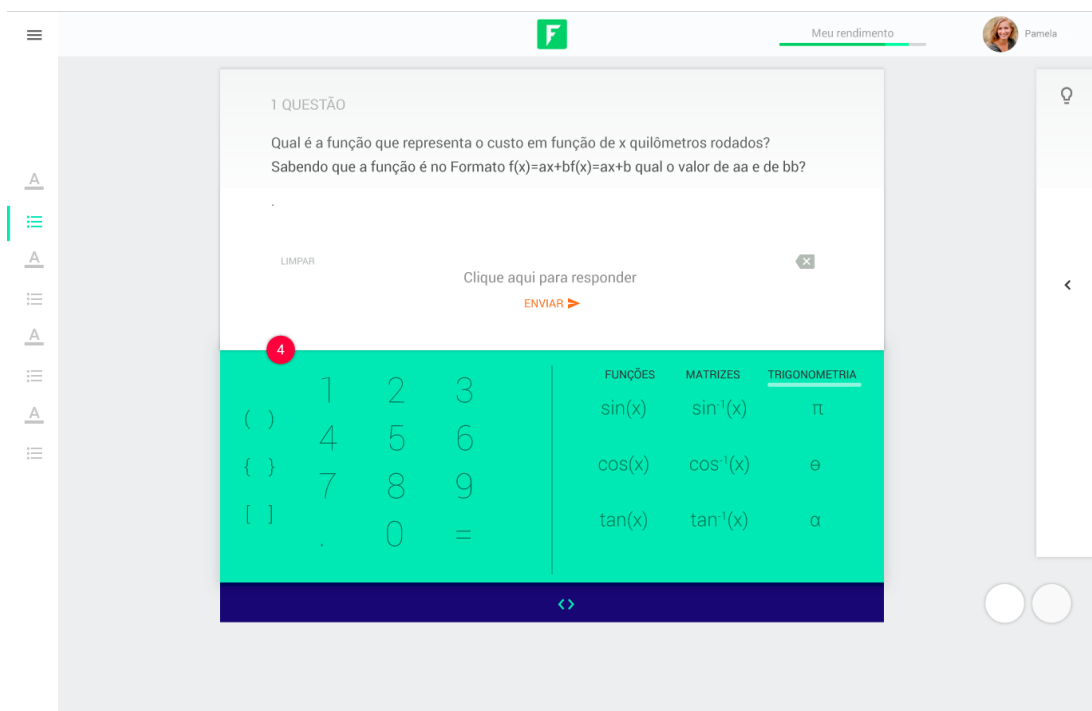


Figura 5 – Tela de resposta com teclado virtual (categoria: Trigonometria)

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.

Qual é a função que representa o custo em função de x quilômetros rodados?
Sabendo que a função é no Formato $f(x) = ax + b$ qual o valor de a e de b ?

LIMPAR

Clique aqui para responder

ENVIAR ➤

	FUNÇÕES	MATRIZES	TRIGONOMETRIA
$\sin(x)$	$\sin^{-1}(x)$	π	
$\cos(x)$	$\cos^{-1}(x)$	e	
$\tan(x)$	$\tan^{-1}(x)$	σ	

LaTeX

DICAS

1ª DICA

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-

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.

4 CONSIDERAÇÕES FINAIS

A proposta se destaca por dar continuidade ao desenvolvimento de uma ferramenta já consolidada no contexto acadêmico, com múltiplos estudos, Trabalho de Conclusão de Curso (TCC)s 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 adaptar as novas estruturas visuais, os dados retornados pela API existente 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>.

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_TQUES_VITEK_MONOGRAFIA.pdf.