

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GABRIEL HENRIQUE RIBAS

**DESENVOLVIMENTO DA INTERFACE GRÁFICA PARA CRIAÇÃO DE
OBJETOS DE APRENDIZAGEM NA FERRAMENTA FARMA**

GUARAPUAVA

2024

GABRIEL HENRIQUE RIBAS

**DESENVOLVIMENTO DA INTERFACE GRÁFICA PARA CRIAÇÃO DE
OBJETOS DE APRENDIZAGEM NA FERRAMENTA FARMA**

**DEVELOPMENT OF THE GRAPHICAL INTERFACE FOR CREATING
LEARNING OBJECTS IN THE FARMA TOOL**

Projeto 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.

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

Coorientador: Prof. Dr. Diego Marczał e Prof. Dr. Alex Sandro De Castilho

GUARAPUAVA

2024



[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 FIGURAS

Figura 1 – Criação de Introdução na FARMA	15
Figura 2 – Criação de um novo Exercícios na FARMA	15
Figura 3 – Criação de uma nova Questão na FARMA	16
Figura 4 – Criação de Dicas na FARMA	17
Figura 5 – Página principal da FARMA	19
Figura 6 – Página principal da FARMA Reborn	20
Figura 7 – Tela para visualização de Objetos de Aprendizagem FARMA Reborn . .	20
Figura 8 – Página principal da area para criação de Objetos de Aprendizagem FARMA Reborn	21
Figura 9 – Página para criação de Introdução dentro de um Objetos de Aprendiza- gem FARMA Reborn	21
Figura 10 – Página para criação de Exercicios de um Objetos de Aprendizagem FARMA Reborn	22
Figura 11 – Página para criação de Questões de um Objetos de Aprendizagem FARMA Reborn	22
Figura 12 – Quadro Kanban do GitHub	25
Figura 13 – História de usuário e cenários	25
Figura 14 – Tela inicial para visualização de Objetos de Aprendizagem na FARMA (autoria própria)	31
Figura 15 – Tela inicial para criação de Objetos de Aprendizagem na FARMA (auto- ria própria)	32
Figura 16 – Tela inicial para visualização de Objetos de Aprendizagem já criada na FARMA (autoria própria)	33
Figura 17 – Tela para visualização de componentes de um Objetos de Aprendiza- gem (autoria própria)	33
Figura 18 – Tela para criação de introdução de um Objetos de Aprendizagem (auto- ria própria)	34
Figura 19 – Tela para criação de exercício de um Objetos de Aprendizagem (autoria própria)	34

Figura 20 – Tela para criação de Dicas de um Objetos de Aprendizagem (autoria própria)	35
Figura 21 – Tela para visualização de componentes de um Objetos de Aprendizagem, agora criados (autoria própria)	36
Figura 22 – Tela para pré-visualização do Objeto de Aprendizagem (autoria própria)	36
Figura 23 – Tela de login da nova versão da FARMA (autoria própria)	37
Figura 24 – Tela de criação de conta da nova versão da FARMA (autoria própria) . .	37
Figura 25 – Tela home da nova versão da FARMA (autoria própria)	38
Figura 26 – Tela de acesso as informações de usuário da nova versão da FARMA (autoria própria)	38

LISTA DE ABREVIATURAS E SIGLAS

Siglas

API	Application Programming Interface
CD	Continuous Delivery
CI	Continuous Integration
DDD	Domain-Driven Design
FARMA	Ferramenta de Autoria para Remediação de Erros com Mobilidade na Aprendizagem
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
LAN	Local Area Network
OA	Objetos de Aprendizagem
RTT	Round Trip Time
WAN	Wide Area Network

SUMÁRIO

1	INTRODUÇÃO	6
1.1	Objetivos	7
1.1.1	Objetivo geral	7
1.1.2	Objetivos específicos	7
1.2	Justificativa	7
2	REVISÃO DA LITERATURA	9
2.1	Arquitetura de Software	9
2.1.1	Arquiteturas Monolíticas Versus Distribuídas	9
2.1.2	Benefícios da Arquitetura Monolítica	11
2.1.3	Falácias da Arquitetura Distribuída	12
3	FARMA	14
3.1	Objetos de Aprendizagem na FARMA	14
3.2	Linha do Tempo	17
3.3	FARMA Reborn	19
3.3.1	Criação de Objetos de Aprendizagem FARMA Reborn	20
4	MATERIAIS E MÉTODOS	23
4.1	Materiais	23
4.2	Métodos	24
4.2.1	Levantamento de requisitos	24
4.2.2	Definição de histórias e cenários	25
4.2.3	Prototipação para levantar os requisitos de interface do usuário	25
4.2.4	Organização do fluxo de desenvolvimento	26
4.2.5	Verificação e validação do software	26
4.2.6	Entrega e Implantação	27
5	ANÁLISE E PROJETO	28
5.1	História de usuário	28
5.1.1	Criação e configuração dos objeto de aprendizagem	28
5.1.2	Criação de exercícios, questões e dicas	29
5.2	Prototipação de Telas	30
5.2.1	Interface de criação de Objeto de Aprendizagem	31

5.2.2	Interface de contribuição para trabalhos futuros	35
6	CONSIDERAÇÕES FINAIS	39
	REFERÊNCIAS	40

1 INTRODUÇÃO

A criação de ferramentas tecnológicas educacionais tem o objetivo de facilitar e enriquecer o aprendizado, por meio de dispositivos como computadores, tablets, smartphones com acesso a *Internet*. Essas tecnologias possibilitam acesso a informações e recursos, promovendo a integração de novas abordagens de ensino em diversas áreas do conhecimento. Assim, com essas ferramentas, busca-se alcançar uma melhoria contínua do ensino, garantindo qualidade na aprendizagem e contribuindo para o desenvolvimento pessoal e profissional (SOUSA, 2011).

As tecnologias educacionais têm impactado e transformado a educação de diversas maneiras. Uma dessas formas, é a possibilidade de personalização do ensino, no qual os educadores podem adaptar os conteúdos e dinamizar suas atividades, de modo que os alunos aprendam de forma mais acessível e eficaz (DAMASCENO *et al.*, 2024). Recursos digitais, como plataformas de aprendizado adaptativo, jogos educativos e realidade aumentada, têm o potencial de aumentar o engajamento dos alunos e aprimorar seu desempenho acadêmico.

Neste cenário, um importante recurso utilizado para melhorar as estratégias de ensino são os objetos de aprendizado ou aprendizagem. Os Objetos de Aprendizagem (OA) podem ser desenvolvidos em qualquer tipo de mídia ou formato, desde algo simples como uma animação ou um conjunto de slides até algo mais complexo, como uma simulação. Geralmente, são estruturados em módulos reutilizáveis em diferentes contextos. Em outras palavras, cada OA é um módulo independente, com conteúdo auto explicativo, que é compreensível e autossuficiente, sem exigir materiais adicionais para seu entendimento (TAROUCO, 2014).

A Ferramenta de Autoria para Remediação de Erros com Mobilidade na Aprendizagem (FARMA) é uma plataforma de aprendizado adaptativo, que tem como objetivo principal a criação de OA que possibilitam testar, avaliar e corrigir soluções apresentadas pelos alunos, promovendo assim a remediação de erros (MARCZAL, 2014). A FARMA oferece um espaço flexível e acessível tanto para alunos quanto para educadores para promoção do ensino-aprendizagem da matemática com a criação e visualização de OA. A versão atual da FARMA¹ passou por diversas melhorias desde a versão desenvolvida por Marczal (2014), tais como, as apresentadas no quadro da linha do tempo da FARMA, apresentada nos capítulos posteriores.

Neste contexto, este projeto busca dar continuidade e aprimorar os trabalhos citados. Para isso, com base nas contribuições apresentadas em Vitek (2023), pretende-se redesenhar as telas da criação de OA e implementá-las, além de readequar o sistema para uma arquitetura monolítica, substituindo a atual *Application Programming Interface* (API). Busca-se, com isso, garantir que a nova versão da FARMA seja mais simples, de fácil gerenciamento e com menor complexidade.

¹ Disponível em <https://farma.educacional.mat.br/>, acessada em 17 de outubro de 2024

1.1 Objetivos

Nesta seção, serão apresentados os objetivos para a atualização da FARMA.

1.1.1 Objetivo geral

Desenvolvimento da interface gráfica para criação de Objetos de Aprendizagem (OA) na Ferramenta de Autoria para Remediação de Erros com Mobilidade na Aprendizagem (FARMA), substituindo o uso da *Application Programming Interface* (API), de modo a se beneficiar da arquitetura monolítica, para facilitar a criação e gestão dos objetos de aprendizagem.

1.1.2 Objetivos específicos

- Substituir a arquitetura implementada como API dos trabalhos de Vitek (2023) e Bonin (2024), para um sistema monolítico;
- Desenvolver uma nova interface gráfica da área de criação de introdução ao conteúdo de um OA;
- Desenvolver uma nova interface gráfica da área de criação de exercícios de um OA;
- Desenvolver uma nova interface gráfica da área de criação de questões de um OA.
- Desenvolver uma nova interface gráfica da área de definição de dicas de um OA;

1.2 Justificativa

A migração de um sistema em uma arquitetura distribuída para uma arquitetura monolítica pode trazer vantagens para o desenvolvimento de sistemas, como será visto posteriormente. Enquanto as arquiteturas distribuídas oferecem benefícios como escalabilidade, desempenho e disponibilidade, elas também enfrentam desafios específicos, especialmente em relação à confiabilidade da rede e à complexidade na gestão de múltiplos serviços interconectados.

Em uma arquitetura distribuída, os sistemas dependem de comunicação em rede constante, o que pode resultar em falhas e inconsistências. Mesmo com as melhorias na confiabilidade das redes, a dependência de múltiplos componentes distribuídos pode aumentar a complexidade do desenvolvimento e da manutenção do sistema, impactando diretamente na eficiência do processo.

Por outro lado, a adoção de uma arquitetura monolítica, onde todos os componentes são integrados em uma única unidade, pode simplificar consideravelmente o desenvolvimento

e a manutenção do sistema. A redução de latência e a menor complexidade na gestão de componentes são alguns dos benefícios dessa abordagem. Além disso, a monoliticidade facilita o controle e o gerenciamento do sistema como um todo, permitindo uma visão mais clara e coesa de seu funcionamento.

Portanto, ao migrar para uma arquitetura monolítica, espera-se não apenas reduzir a complexidade operacional e melhorar o desempenho, mas também criar uma base mais sólida e fácil de gerenciar a longo prazo.

2 REVISÃO DA LITERATURA

Nesta seção, serão apresentados conceitos utilizados para justificar a atualização da FARMA.

2.1 Arquitetura de Software

A arquitetura de software é o conjunto de estruturas necessárias para compreender e analisar um sistema. Ela inclui os elementos de software, as relações entre esses elementos e as propriedades de ambos (BASS, 2013). De acordo com Richards (2024), as arquiteturas podem ser classificadas em dois estilos principais: monolítica e distribuída, ambas das arquiteturas apresentam benefícios em suas aplicações, como também complicações.

2.1.1 Arquiteturas Monolíticas Versus Distribuídas

As arquiteturas de software podem ser classificadas em dois estilos principais: monolítica e distribuída. A arquitetura monolítica consiste em uma única unidade de implementação, na qual todo o código da aplicação é desenvolvido e implantado de forma integrada. Já a arquitetura distribuída é composta por várias unidades de implementação que se comunicam por meio de protocolos de acesso remoto.

Embora nenhum esquema de classificação seja perfeito, essa distinção é útil, pois todas as arquiteturas distribuídas compartilham desafios e problemas que não ocorrem em arquiteturas monolíticas. Essa diferenciação permite uma análise mais precisa das vantagens e desvantagens de cada abordagem.

Em seu livro, os autores Richards e Ford¹ apresentam e detalham os seguintes estilos de arquitetura:

- **Arquitetura em Camadas:** A *arquitetura n-tier (multicamadas)* organiza o sistema em camadas lógicas, com apresentação, lógica de negócios, persistência e banco de dados. Essa abordagem é bastante popular devido ao seu baixo custo e simplicidade, tornando-a uma boa opção para aplicações menores. No entanto, à medida que o projeto cresce, sua manutenção pode se tornar mais complexa, o que impacta a agilidade e a escalabilidade do sistema.
- **Arquitetura Pipeline:** A *arquitetura pipes-and-filters*, divide as funcionalidades em componentes menores, chamados filtros, responsáveis por realizar tarefas específicas, interligados por canais unidimensionais. No entanto, à medida que o sistema cresce,

¹ Fundamentos da Arquitetura de Software: uma Abordagem de Engenharia - Mark Richards e Neal Ford(2024)

suas limitações tornam-se evidentes, tornando-a mais indicada para implementações menores e mais simples.

- **Arquitetura Microkernel:** A *arquitetura pipes-and-filters*, é baseada em um sistema central com componentes plugáveis, onde o núcleo contém recursos essenciais e os plug-ins adicionam funcionalidades conforme necessário. É amplamente utilizada em produtos para download e aplicações de diversos formatos. Seus principais pontos positivos são o baixo custo e a simplicidade de implementação. No entanto, enfrenta desafios em escalabilidade, tolerância a falhas e elasticidade.
- **Arquitetura baseada em serviços:** A arquitetura baseada em serviços é uma abordagem flexível, composta por uma camada de interface do usuário e serviços de domínio distribuídos de forma separada e remota, com um banco de dados monolítico. Seus principais pontos positivos são a flexibilidade e a capacidade de separar e escalar componentes de forma independente. No entanto, ela enfrenta desafios em termos de complexidade de implementação, problemas de desempenho e dificuldades na manutenção dos serviços.
- **Arquitetura orientada a eventos:** A arquitetura baseada em eventos é assíncrona e distribuída, amplamente usada em aplicativos de alta escala e desempenho, mas também aplicável em sistemas menores. Ela opera com componentes desacoplados que tratam eventos de forma assíncrona, com requisições encaminhadas a processadores específicos por um orquestrador. Seus pontos positivos incluem escalabilidade, desacoplamento de componentes, flexibilidade e alto desempenho. No entanto, ela enfrenta desafios como complexidade de implementação, risco de inconsistências, sobrecarga de comunicação, dificuldades de monitoramento e custos elevados.
- **Arquitetura baseada em espaços:** Essa arquitetura é projetada para ambientes que exigem alta escalabilidade, elasticidade e simultaneidade, sendo ideal para aplicativos com número variável de usuários simultâneos. Ela lida com falhas em fluxos padrão da web, mas pode gerar gargalos à medida que o tráfego aumenta. Seus pontos fortes incluem elasticidade, escalabilidade e desempenho, enquanto os desafios envolvem complexidade de implementação e dificuldades na testabilidade. A topologia inclui unidades de processamento, data-pumps, mecanismos de gravação e unidades de leitura de dados.
- **Arquitetura orientada a serviços e Baseada em Orquestração:** A arquitetura orientada a serviços baseada em orquestração surgiu nos anos 1990 para integrar negócios e tecnologia de forma ágil, com uma estrutura em camadas que inclui serviços de negócios, corporativos, aplicativos, infraestrutura e um mecanismo de orquestração. Seus pontos positivos são a reutilização de serviços, controle maior e integração

simplificada. No entanto, ela apresenta desafios como alto acoplamento, gargalos operacionais, limitações de escalabilidade e aumento da complexidade com a integração de mais serviços. Embora amplamente adotada, foi superada por arquiteturas mais flexíveis, como os microsserviços, para resolver esses problemas.

- **Arquitetura de microsserviços:** A arquitetura de microsserviços, inspirada no Domain-Driven Design (DDD), divide o sistema em serviços autônomos, cada um com seu próprio banco de dados e comportamento, garantindo um alto grau de desacoplamento. Seus principais pontos positivos são a flexibilidade, escalabilidade e capacidade de evolução, permitindo o desenvolvimento e implantação independentes de cada serviço. No entanto, ela apresenta desafios como alto custo, complexidade operacional e questões de desempenho devido ao gerenciamento de múltiplos serviços.

Os estilos de arquitetura distribuída (como arquitetura baseada em serviços, orientada a eventos, baseada em espaços, orientada a serviços e microsserviços) oferecem vantagens significativas em termos de desempenho, escalabilidade e disponibilidade, se comparados aos estilos de arquitetura monolítica (como as arquiteturas em camadas, pipeline e microkernel). No entanto, esses estilos distribuídos apresentam importantes escolhas em relação a essas vantagens, exigindo um equilíbrio cuidadoso entre o poder de desempenho e as complexidades adicionais envolvidas na sua implementação e manutenção. O primeiro grupo de problemas em todas as arquiteturas distribuídas é descrito nas falácias da computação distribuída, termo criado por L. Peter Deutsch e colegas na Sun. Todas as oito falácias da computação distribuída se aplicam hoje às arquiteturas distribuídas. As seguintes seções descrevem cada uma.

2.1.2 Benefícios da Arquitetura Monolítica

Para justificar a mudança de uma API para uma interface gráfica GUI no contexto de uma arquitetura monolítica, podemos considerar os seguintes benefícios:

- **Simplicidade na Manutenção:** Em uma arquitetura monolítica, todo o código está localizado em um único repositório, facilitando a manutenção e depuração. Segundo *Martin Fowler* em *Patterns of Enterprise Application Architecture*, sistemas monolíticos são mais fáceis de testar e modificar, uma vez que as dependências estão localizadas em um único lugar, permitindo ajustes rápidos sem a necessidade de coordenação entre vários serviços distribuídos (FOWLER, 2006).
- **Menor Complexidade de Infraestrutura:** *Sam Newman*, em *Building Microservices*, destaca que arquiteturas distribuídas aumentam a complexidade da infraestrutura, exigindo sistemas de orquestração, balanceamento de carga e monitoramento para garantir a comunicação entre os serviços. Em contrapartida, uma aplicação monolítica

com uma interface gráfica integrada reduz essa complexidade, pois elimina a necessidade de gerenciar API e servidores separados, facilitando o gerenciamento da aplicação (NEWMAN, 2021).

- **Desempenho Melhorado:** *Martin Kleppmann*, em *Designing Data-Intensive Applications*, aponta que arquiteturas distribuídas, como aquelas que dependem de API, podem sofrer de latências devido à comunicação em rede. Ao centralizar a interface gráfica no monólito, a comunicação entre o *frontend* e o *backend* é local, o que reduz a latência e melhora o desempenho, pois elimina as dependências de chamadas HyperText Transfer Protocol(HTTP) externas (KLEPPMANN, 2017).
- **Facilidade de Integração:** Ao integrar a GUI diretamente no monólito, a troca de dados entre a interface e o *backend* é mais simples e direta. *Newman* discute como a integração entre serviços separados em arquiteturas distribuídas pode exigir mapeamentos e transformações de dados adicionais, o que complica o desenvolvimento. Em um monólito, essas operações são locais, o que simplifica o processo e garante consistência entre as camadas do sistema (NEWMAN, 2021).

A migração de uma API para uma interface gráfica dentro de uma arquitetura monolítica pode trazer benefícios como simplificação do desenvolvimento, redução de latência e de complexidade, além de permitir um gerenciamento mais eficiente do sistema como um todo.

2.1.3 Falácias da Arquitetura Distribuída

O primeiro conjunto de problemas enfrentados pelas arquiteturas distribuídas é conhecido como as falácias da computação distribuída, um termo criado por L. Peter Deutsch (ROTEM, 2014), que significa algo que se acredita ou pressupõe ser verdade, mas não é. Essas falácias representam suposições incorretas. A seguir retrataremos sobre cada uma dessas falácias:

- **A rede é de confiança:** Assume que a rede sempre funcionará sem problemas, ignorando falhas como desconexões ou ataques, e por conta disso, ela se torna uma falácia na arquitetura distribuída, porque não leva em conta a imprevisibilidade e a instabilidade da rede.
- **A latência é zero:** Assume que os dados se movem instantaneamente entre os pontos, o que não é verdade. Na realidade, a latência, o tempo necessário para os dados percorrerem a rede, é um fator crucial e impacta diretamente o desempenho de sistemas distribuídos. Mesmo em Local Area Network (LAN), a latência pode ser maior que o acesso à memória local, o que afeta chamadas remotas.

- **Largura de banda é infinita:** Assume que a capacidade de rede continuará a crescer sem limitações, o que não é verdade. Embora a largura de banda tenha aumentado, ela é limitada pela demanda crescente de dados, como VoIP e vídeos, e por problemas em Wide Area Network (WAN), como perda de pacotes e Round Trip Time (RTT). Além disso, o tamanho dos pacotes e a perda de pacotes impactam diretamente a transmissão de dados.
- **A rede é segura:** Assume que a segurança da rede é garantida, o que é um erro comum, apesar dos avanços em tecnologia de segurança. Muitas redes ainda dependem apenas da segurança no perímetro, o que as torna vulneráveis após invasões externas. Mesmo com a adoção de medidas proativas, os ataques cibernéticos continuam a crescer.
- **A topologia não muda:** Assume que a estrutura da rede será sempre estável, o que não é verdade. A topologia de rede pode mudar devido à adição de novos nós, falhas de links ou desconexões, especialmente em redes dinâmicas, como a internet.
- **Há somente um administrador:** Assume que uma única pessoa ou entidade gerencia toda a infraestrutura, o que é simplista em sistemas distribuídos. Na realidade, esses sistemas envolvem múltiplos administradores, equipes e níveis de controle, como em serviços em nuvem, onde o provedor gerencia a infraestrutura física e os usuários controlam a configuração do software.
- **Custo de transporte é zero:** Assume que transferir dados entre pontos não tem impacto financeiro ou de desempenho, o que não é verdade. Embora os dados possam ser enviados por grandes distâncias, há custos associados à largura de banda, latência e manutenção da rede.
- **A rede é homogênea:** Assume que toda a rede tem as mesmas características e suporta tráfego de forma consistente, o que não é verdade. As redes podem ser heterogêneas, com diferentes tipos de dispositivos, tecnologias e condições de rede, como LAN, WAN e conexões sem fio, que variam ao longo do tempo.

3 FARMA

A Ferramenta de Autoria para a Remediação de Erros com Mobilidade na Aprendizagem (FARMA) teve sua origem na tese de doutorado de Marczał (2014), que identificou a falta de ferramentas educacionais disponíveis para monitorar o processo de ensino e aprendizagem por meio da análise de tentativas e erros. Esse desafio já havia sido discutido no estudo de Marczał (2011), que destacaram a ausência de um gerenciador de OA. A implementação de tal gerenciador permitiria acompanhar a evolução do aluno, utilizando seus erros como uma oportunidade para reforçar o conteúdo e promover a remediação. De maneira semelhante, o trabalho de Leite (2013) também abordou a aplicação da remediação de erros no ensino, enfatizando sua relevância no processo de aprendizagem.

O principal objetivo da FARMA é fornecer um ambiente eficiente para o gerenciamento de erros dos alunos, com foco no ensino de conceitos matemáticos. Para a criação dos exercícios, foram utilizados OA, a fim de proporcionar um ambiente mais flexível e adaptável na abordagem dos conteúdos. Dessa forma, professores e monitores de matemática podem ajustar o ensino conforme as necessidades e objetivos específicos de cada aluno, promovendo uma aprendizagem mais personalizada e eficaz.

3.1 Objetos de Aprendizagem na FARMA

Como supracitado, em um contexto geral, um OA pode ser qualquer tipo de mídia ou formato, desde algo simples até algo mais complexo. Na FARMA uma OA assume o papel de captar o erro de um estudante seguindo a seguinte estrutura:

- **Introduções descritivas e explicativas::** O principal objetivo da *introdução* é apresentar conceitos-chave que serão fundamentais para responder às questões propostas, utilizando imagens, exemplos e explicações para facilitar a compreensão. Dessa forma, a introdução pode ser organizada em partes menores, tornando o conteúdo mais acessível e favorecendo o entendimento do estudante. A tela de criação de introdução na primeira versão pode ser observada na Figura 1.

FARMA HOME Olá Gabriel Henrique Ribas

Nova Introdução para o OA

Título *

Conteúdo *

☐ Publicar

Salvar

Figura 1 – Criação de Introdução na FARMA

- **Exercícios:** Após definir uma introdução, o educador pode elaborar *exercícios* para avaliar o entendimento do aluno sobre o conteúdo apresentado. A estrutura de uma questão começa pelo título, que destaca o tema abordado. Em seguida, o enunciado explica o que é solicitado, podendo incluir imagens, links, textos ou outros recursos complementares para melhor contextualização. A resposta esperada pode variar entre um número, uma expressão aritmética ou algébrica. A tela de criação de um novo exercícios pode ser vista na Figura 2 e a tela de criação de questão está representada na Figura 3.

FARMA HOME Olá Gabriel Henrique Ribas

Novo exercício para o OA

Título *

Enunciado *

☐ Publicar

Salvar

Figura 2 – Criação de um novo Exercícios na FARMA

Home / Meus Objetos de Aprendizagem / Conteúdos do OA I / Visualizando exercício título teste

Exercício título teste

Criado em 05/02/2025 21:58:18 Atualizado em 05/02/2025 21:59:43

enunciado teste

Questões

[+ Criar nova questão](#)

Nova questão

Título *

Enunciado *

Resposta correta *

A resposta correta pode ser um número, expressão aritmética ou algébrica.

Considerar casas decimais para avaliar a resposta.

☐ Publicar

[Salvar](#) [Cancelar](#)

Figura 3 – Criação de uma nova Questão na FARMA

- **Dicas:** Após salvar a questão, é possível adicionar uma dica, elemento essencial para a remediação de erros dos estudantes. A dica pode ser configurada para aparecer após um determinado número de tentativas erradas, auxiliando o aluno no processo de aprendizagem. A tela de criação de dicas pode ser vista na Figura 4

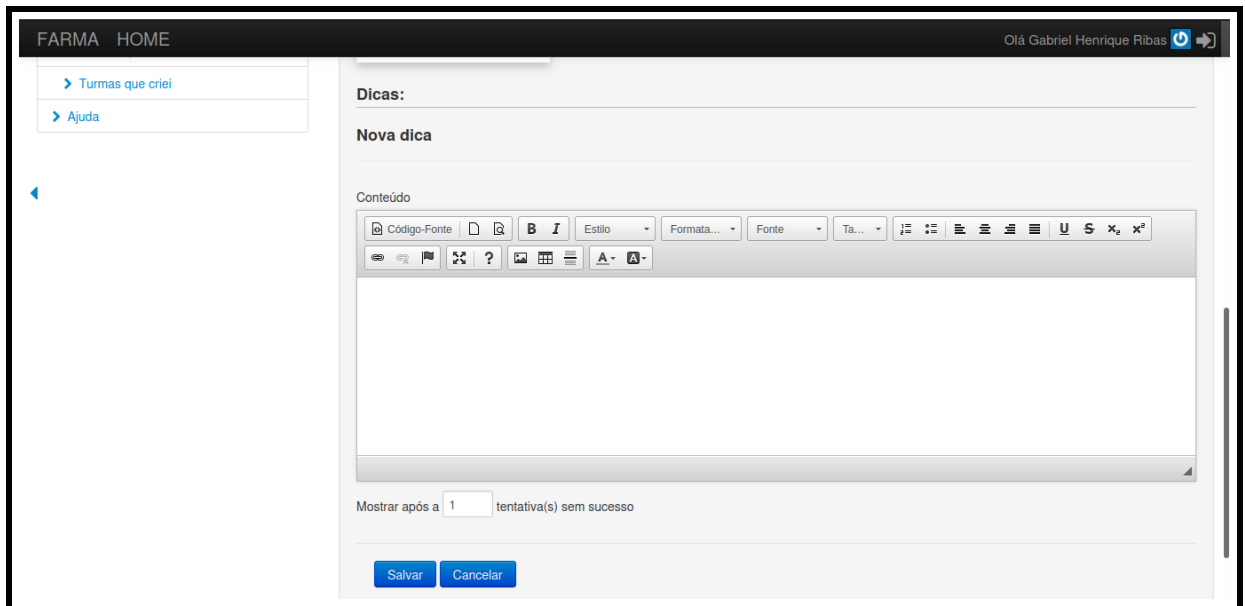


Figura 4 – Criação de Dicas na FARMA

Os OA podem ser segmentados em pequenos blocos, facilitando a assimilação do conteúdo pelo aprendiz e proporcionando uma experiência de aprendizado mais clara e eficaz.

3.2 Linha do Tempo

Ao longo do tempo, desde o seu início, a FARMA tem sido objeto de estudos e pesquisas em diversos trabalhos acadêmicos, como dissertações de mestrado, teses de doutorado, TCC's, artigos científicos e até mesmo premiações. Para ilustrar essa trajetória, foi desenvolvida uma linha do tempo que destaca a evolução da FARMA, a qual pode ser observada na cronologia abaixo.

- 2009 • Melhor Artigo do Workshop de Informática na Escola (WIE) - Sociedade Brasileira de Computação (SBC) (2º Lugar) ¹
- 2010 • Concurso de Teses, Dissertações e Trabalhos de Conclusão de Curso em Informática na Educação (CTD-IE) - Sociedade Brasileira de Computação (SBC) (3º Colocado cat. Dissertação) ²
- 2010 • Um arcabouço que enfatiza a retroação a contextos de erro durante o acesso a conteúdos educacionais (MARCZAL, 2010)
- 2011 • Classificação automática de erros de aprendizes humanos do processo de indução analítica (BAZZO, 2011)
- 2013 • Arquitetura para remediação de erros baseada em múltiplas representações externas (LEITE, 2013)

¹ Disponível em <https://tophotels.com/tecnologia-da-informacao-no-sistema-educacional-do-brasil.pt-br>

² Disponível em https://www.proativa.virtual.ufc.br/sbie2010/Anais_do_XXI_SBIE/CTD-IE-MS.html

- 2014 • Farma: uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos (MARCZAL, 2014)
- 2015 • Sequenciamento adaptativo de exercícios baseado na correspondência entre a dificuldade da solução e o desempenho dinâmico do aprendiz (SILVA, 2015)
- 2015 • Informática educacional e a mediação do erro na educação: um estudo teórico-crítico e uma proposta de instrumento computacional (KUTZKE, 2015)
- 2015 • Avaliação do impacto da retroação na aprendizagem apoiada por uma ferramenta educacional (MOURA, 2015)
- 2015 • Concurso de Teses, Dissertações e Trabalhos de Conclusão de Curso em Informática na Educação (CTD-IE) - Sociedade Brasileira de Computação (SBC) (1º colocado na cat. Tese)³
- 2015 • FARMA: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos (MARCZAL, 2015)
- 2015 • Otimizando o processo de Ensino e Aprendizagem com a Arquitetura para Desenvolvimento de Objetos de Aprendizagem - ADOA (LEITE, 2015)
- 2016 • Metodologia e Software Educacional para a Investigação e Remediação de Erros Conceituais em Matemática (MARCZAL, 2016)
- 2017 • ATAUDIW - Uma ferramenta de autoria para auxiliar o uso da lousa digital interativa (COSTA, 2017)
- 2017 • EduTraceSys como objeto de aprendizagem na remediação de erros (TRIBECK, 2017)
- 2018 • Adaptabilidade de Objetos de Aprendizagem usando Calibragem e Sequenciamento Adaptativo de Exercícios (SILVA, 2018)
- 2019 • Objetos de Aprendizagem com Feedbacks para a Autorregulação da Aprendizagem de Conceitos Matemáticos Necessários para o Cálculo Diferencial e Integral (CASTILHO, 2019)
- 2019 • Visualização de Objetos de Aprendizagem para a Ferramenta de autoria FARMA (RAMOS, 2019)
- 2021 • Mudanças nas ideias prévias sobre ponto, reta e plano por meio da interação com a ferramenta FARMA (ARRUDA, 2021)
- 2022 • Desenvolvimento do Módulo de Estatísticas da Ferramenta de Autoria de Objetos de Aprendizagem FARMA (SANTOS, 2018)
- 2023 • Desenvolvimento do Módulo de Estatísticas da Ferramenta de Autoria de Objetos de Aprendizagem FARMA (VITEK, 2023)
- 2024 • Implementação De Uma Api Para Visualização E Interação Com Objetos De Aprendizagem Para Ferramenta De Autoria Farma (BONIN, 2024)

Além das contribuições históricas, a FARMA passou por várias versões, frutos de mestrados, doutorados e TCCs, como FARMA ALG⁴, FARMA CALC⁵ e FARMA Reborn⁶. A

³ Disponível em <https://tophotels.com/tecnologia-da-informacao-no-sistema-educacional-do-brasil.pt-br>

⁴ Disponível em http://farmaalg.c3sl.ufpr.br/users/sign_in

⁵ Disponível em <https://farma-calc.educacional.mat.br/>

⁶ Disponível em <https://farma-reborn.educacional.mat.br/>

seguir, vamos apresentar a versão FARMA Reborn, um projeto de "Redesign e Refatoração da FARMA". Essa versão trouxe um design moderno, que servirá como inspiração para a criação das novas interfaces gráficas no decorrer deste trabalho.

3.3 FARMA Reborn

A nova versão da FARMA está sendo desenvolvida para facilitar a criação, visualização e edição de OA. Esta nova versão terá como base a versão original ⁷ apresentada na Figura 5.



Figura 5 – Página principal da FARMA

Entre 2015 e 2018, foi desenvolvido o projeto de "Redesign e Refatoração da FARMA", com a colaboração de estudantes do curso de Graduação em Sistemas para Internet. O objetivo desse projeto foi melhorar a ferramenta, otimizando suas estruturas internas e interfaces, que haviam sido criadas no trabalho de Santos (2018).

Como resultado, surgiu a segunda versão da ferramenta, que atualmente foi descontinuada para a criação de uma nova versão, mas que deixou algumas contribuições. A Figura 6 exibe a página principal dessa versão.

⁷ Disponível em <https://farma.educacional.mat.br/>. Acessado em 29 de outubro de 2024



Figura 6 – Página principal da FARMA Reborn

3.3.1 Criação de Objetos de Aprendizagem FARMA Reborn

Como resultado do projeto "Redesign e Refatoração da FARMA", foram desenvolvidas telas para a área de criação de OA, que serão apresentadas a seguir:

Na Figura 7, é possível observar a tela inicial para a visualização de um OA. Nesta imagem, a tela é exibida sem nenhum OA disponível.

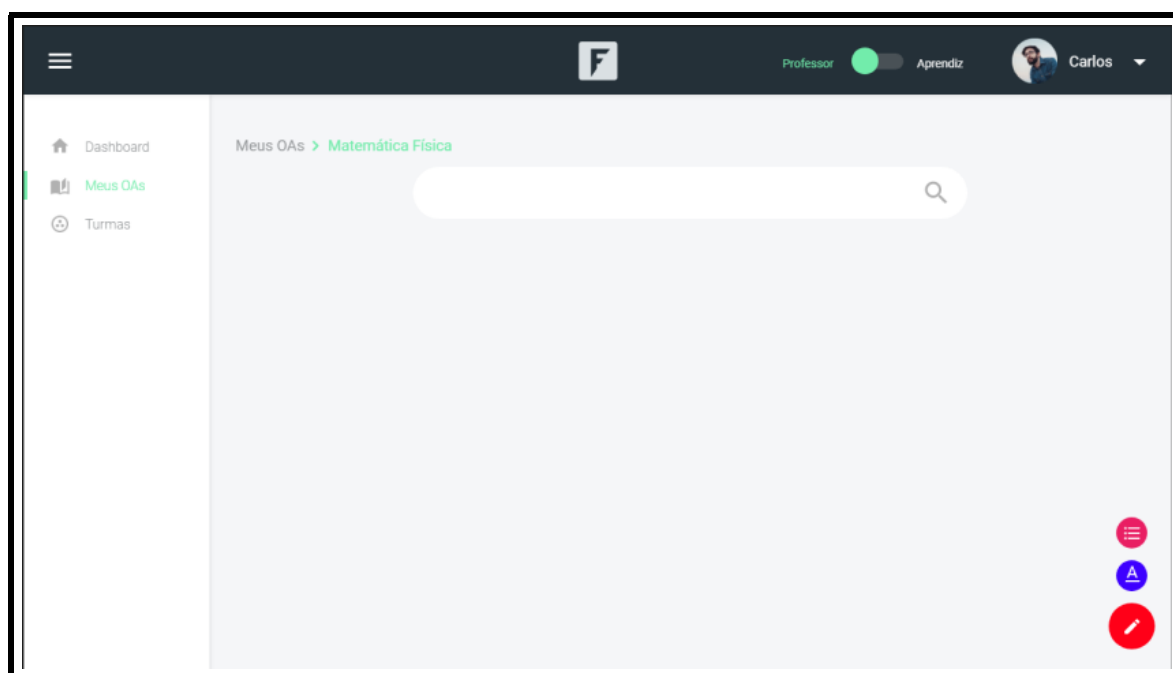


Figura 7 – Tela para visualização de Objetos de Aprendizagem FARMA Reborn

Na Figura 8, é possível observar a tela de criação de um OA. Para fazer esse cadastro será necessário inserir um título da OA, uma descrição e uma imagem (*opcional*).

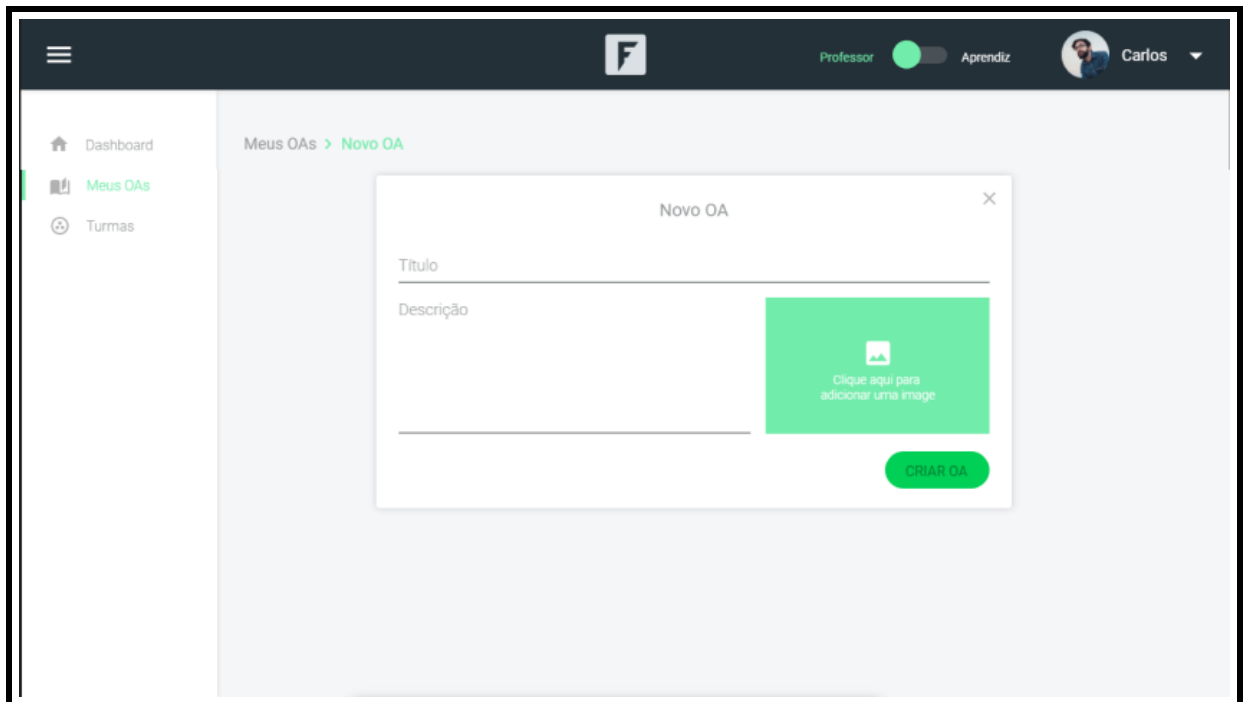


Figura 8 – Página principal da area para criação de Objetos de Aprendizagem FARMA Reborn

Na Figura 9, é possível observar a tela de criação de uma introdução dentro de um OA. Para a criação de uma introdução é necessário inserir um titulo e o conteúdo abordado.

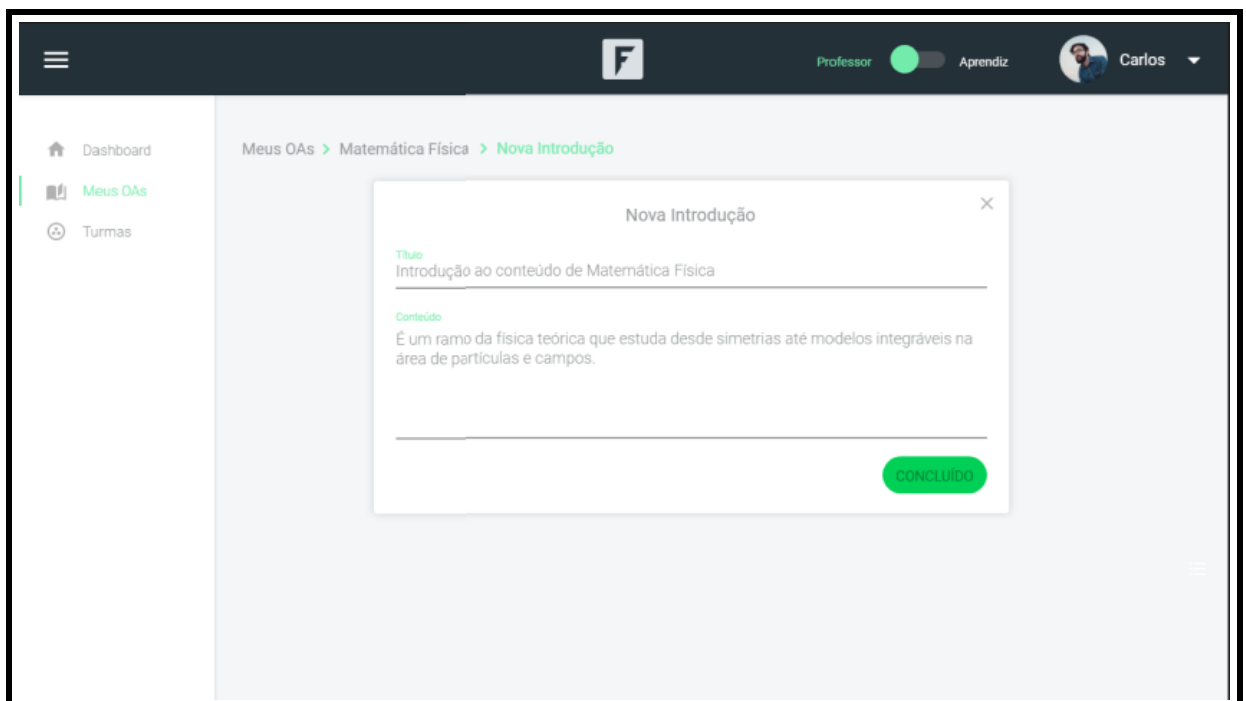


Figura 9 – Página para criação de Introdução dentro de um Objetos de Aprendizagem FARMA Reborn

Na Figura 10, é apresentada a tela de criação de um novo exercício dentro de um OA. Para criar um exercício, é necessário inserir um título e definir o conteúdo a ser abordado.

The screenshot shows the 'Novo Exercício' form within the FARMA Reborn interface. The form is titled 'Novo Exercício' and has a close button (X). It contains two main sections: 'Título' (Title) with the text 'Exercício 1' and 'Conteúdo' (Content) with the text 'É um ramo da física teórica que estuda desde simetrias até modelos integráveis na área de partículas e campos.' There is a green 'CONCLUIDO' (Completed) button at the bottom right.

Figura 10 – Página para criação de Exercícios de um Objetos de Aprendizagem FARMA Reborn

Na Figura 11, é possível observar a tela de criação de novas questões de um OA.

The screenshot shows the 'Nova Questões' form within the FARMA Reborn interface. The form is titled 'Nova Questões' and has a close button (X). It contains three main sections: '1 Título' (1 Title) with a text input field, 'Conteúdo' (Content) with a text input field containing placeholder text, and 'Resposta correta' (Correct Answer) with a text input field. There is a checkbox labeled 'Considerar ordem das resposta' (Consider the order of the answer) and a green 'CRIAR QUESTÃO' (Create Question) button at the bottom right.

Figura 11 – Página para criação de Questões de um Objetos de Aprendizagem FARMA Reborn

4 MATERIAIS E MÉTODOS

Este projeto tem o objetivo desenvolver uma interface gráfica para a criação de OA, seguindo uma arquitetura monolítica, buscando alternativas de tecnologias diferentes das abordadas no trabalho de Vitek (2023), na tentativa de desenvolver a interface gráfica da criação de OA. Para o desenvolvimento do *Frontend* de seu trabalho, foi utilizado *React Next*, com auxílio do *Framework CSS Tailwind*, opções essas, que trouxeram dificuldades em seu trabalho por conta de sua complexidade e uso de uma API para renderização de dados dinâmicos. Após analisar as dificuldades apresentadas em Vitek (2023), foi avaliada alternativas tecnológicas para o desenvolvimento da nova interface gráfica para criação de OA, com o intuito de se beneficiar de uma arquitetura monolítica.

4.1 Materiais

Neste projeto, serão aplicadas as seguintes ferramentas e tecnologias que abrangem desde o design até a implementação e o gerenciamento do código. A seleção desses recursos tem como objetivo otimizar o processo de desenvolvimento e assegurar a qualidade do produto final.

- **ClickUP** ¹: Software de gestão de projetos que oferece recursos como tarefas, bate-papo, quadros brancos, planilhas e colaboração em documentos, tudo integrado em uma plataforma única.
- **Figma** ²: Ferramenta para a criação de protótipos interativos das telas da interface, possibilitando a visualização e testes do fluxo de navegação e da experiência do usuário antes da implementação.
- **VSCode** ³: É um editor de código-fonte aberto, leve e altamente extensível, desenvolvido pela Microsoft. Ele oferece suporte a diversas linguagens de programação, possui recursos avançados como depuração integrada, controle de versão com Git, realce de sintaxe, IntelliSense para autocompletar código e um vasto ecossistema de extensões.
- **Tailwindcss** ⁴: É um framework de CSS utilitário que permite construir interfaces modernas de forma rápida, diretamente no HTML.
- **Postgresql** ⁵: É um sistema de banco de dados objeto-relacional de código aberto, reconhecido por sua confiabilidade, robustez de recursos e alto desempenho.

¹ Disponível em <https://clickup.com/pt-BR>

² Disponível em <https://www.figma.com/pt-br/>

³ Disponível em <https://code.visualstudio.com/>

⁴ Disponível em <https://tailwindcss.com/>

⁵ Disponível em <https://www.postgresql.org/>

- **Docker** ⁶: É um conjunto de soluções de Plataformas que empregam virtualização no nível do sistema operacional para distribuir software em unidades chamadas contêineres.
- **Ruby on Rails** ⁷: Framework web completo e poderoso que acelera o desenvolvimento de aplicações web, utilizando a linguagem Ruby. Facilita a criação de código bem estruturado, escalável e de fácil manutenção.
- **Hotwire** ⁸: Framework padrão do Rails para criação de aplicações web interativas. Ele combina o Turbo(navegação dinâmica) e Stimulus(interação no FrontEnd).
- **GitHub** ⁹: Plataforma de hospedagem de código-fonte e colaboração para projetos de software. Possibilita o controle de versão do código, facilitando o acompanhamento das alterações, a colaboração entre desenvolvedores e a resolução de conflitos.
- **Git** ¹⁰: É um sistema de controle de versão distribuído, gratuito e de código aberto, projetado para gerenciar projetos de qualquer tamanho com rapidez e eficiência.

4.2 Métodos

Para execução do projeto de uma nova interface gráfica para criação de OA, é proposto os seguintes passos para desenvolvimento:

4.2.1 Levantamento de requisitos

Antes de iniciar o desenvolvimento, é fundamental compreender o sistema para que possamos levantar os requisitos necessários. Esse entendimento é crucial para atender às demandas de desenvolvimento da FARMA. Para organizar esses requisitos e o progresso do desenvolvimento, utilizaremos o quadro *Kanban*, uma ferramenta de gestão de projetos que facilita a visualização do trabalho e otimiza a eficiência. O quadro *Kanban* define as seguintes etapas do processo: No *Backlog*, as tarefas aguardam priorização e inclusão no fluxo de trabalho ativo; *Ready*, são tarefas que estão prontas para serem iniciadas; *In Progress*, tarefas em fase de desenvolvimento; *In Review*, tarefas em fase de revisão dos responsáveis e, *Done* que são tarefas finalizadas.

Dessa forma, cada requisito é subdividido em tarefas menores, o que facilita o gerenciamento e o acompanhamento do progresso. Na Figura 12, podemos observar a representação desse fluxo utilizando a ferramenta GitHub.

⁶ Disponível em <https://www.docker.com/>

⁷ Disponível em <https://rubyonrails.org/>

⁸ Disponível em <https://www.hotrails.dev/>

⁹ Disponível em <https://github.com/>

¹⁰ Disponível em <https://git-scm.com/>

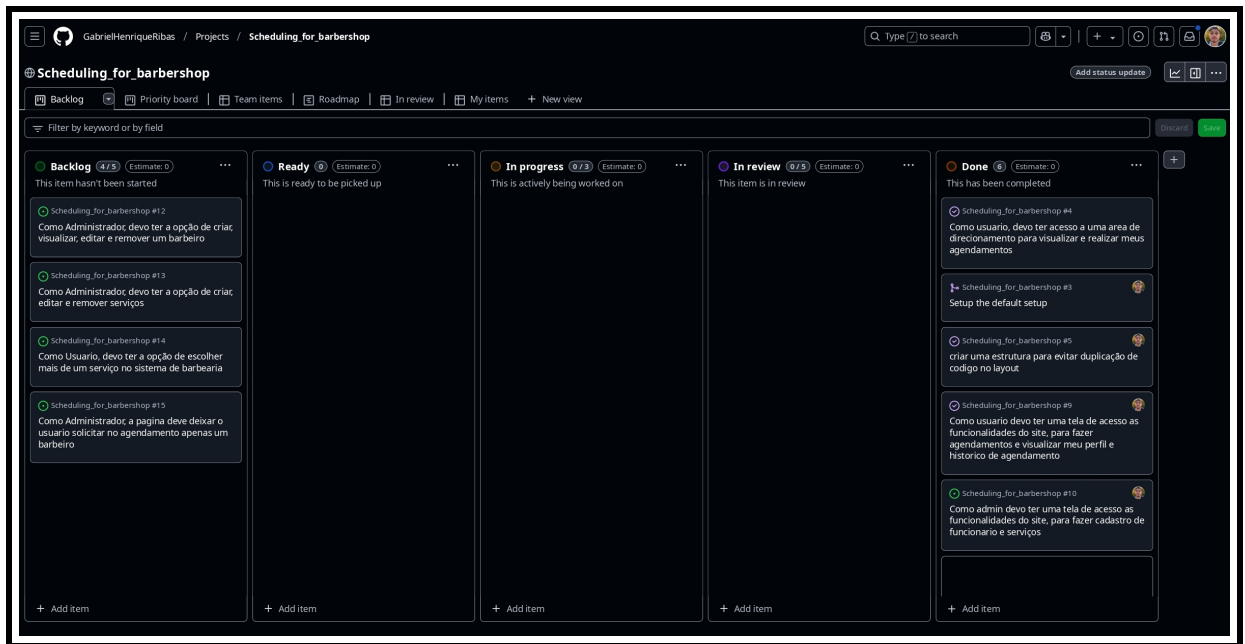


Figura 12 – Quadro Kanban do GitHub

4.2.2 Definição de histórias e cenários

No processo inicial de levantamento de requisitos, o Backlog será utilizado para inserir as histórias de usuário, com o objetivo de definir as tarefas a serem desenvolvidas, baseadas nas necessidades dos usuários. O termo "história de usuário" refere-se a uma funcionalidade específica, que pode incluir um ou mais cenários que demonstram diferentes formas de utilizá-la. Para construir essas histórias, utilizamos a estrutura padrão: **Dado** para identificar o estado atual, **Quando** para indicar a ação a ser realizada e **Então** para descrever a consequência dessa ação. Com essa abordagem, podemos extrair as tarefas necessárias para dar continuidade ao desenvolvimento. Na Figura 13, podemos observar um modelo de história de usuário.



Como educador, **quero** criar um novo OA **para** que eu possa disponibilizar conteúdos interativos para os alunos.

Dado que sou um educador,

Quando eu acessar a plataforma e optar por criar um novo OA,

Então deverei ter a opção de inserir conteúdos interativos.

Figura 13 – História de usuário e cenários

4.2.3 Prototipação para levantar os requisitos de interface do usuário

Será desenvolvido telas interativas que demonstram o funcionamento da criação de OA na FARMA. Seguindo a paleta de cores e a tipografia utilizadas em trabalhos anteriores (LARA,

2017). Para a criação dessas telas será utilizado a ferramenta Figma, para criação de interfaces web, mobile, entre outros. A prototipação interativa no Figma traz múltiplos benefícios para o desenvolvimento da nova plataforma. (AWARI., 2023), tais como:

- **Antecipação visual:** Permite que os *stakeholders* (*partes interessadas*) conheçam a interface da plataforma antes da implementação, promovendo uma comunicação mais clara e alinhamento de expectativas.
- **Avaliação de usabilidade:** Possibilita simular a experiência do usuário na plataforma, identificando dificuldades e validando soluções antes do desenvolvimento.
- **Ajustes e otimização:** Facilita a adaptação e melhoria do design com base no feedback dos stakeholders, assegurando que a plataforma atenda às demandas dos usuários.

4.2.4 Organização do fluxo de desenvolvimento

O *GitLab Flow* é um modelo de fluxo de trabalho que organiza as ramificações (branches) do código e facilita a colaboração entre desenvolvedores. Ele divide as branches em diferentes tipos:

- **Main:** Representa a versão estável do código em produção.
- **Staging:** Usada para testes finais antes da produção.
- **Production:** Espelha o ambiente de produção e é usada exclusivamente para implantações.
- **Feature:** Criadas a partir de Main para desenvolver novas funcionalidades.
- **Hotfix/BugFix:** Criadas a partir de Production (para Hotfix) ou Staging (para correção de bugs).

Esse modelo ajuda a estruturar o desenvolvimento e facilita a gestão de mudanças no código.

4.2.5 Verificação e validação do software

A verificação e validação do software são processos essenciais para garantir que o software atenda às especificações e aos requisitos do cliente. Esses processos podem ser divididos em diferentes tipos de testes:

- **Teste de unidade:** Verifica se um procedimento ou método funciona conforme esperado.
- **Teste de módulo:** Testa a interação entre várias unidades de um módulo, como classes relacionadas.
- **Teste de integração:** Garante que as interfaces entre as unidades estejam consistentes e se comuniquem corretamente, sem testar a funcionalidade interna.
- **Teste de sistema ou aceitação:** Avalia se o software integrado atende às suas especificações.

Cada tipo de teste delega responsabilidades aos níveis inferiores, com foco na verificação de unidades até a validação do sistema como um todo. A monografia aprofundará esses testes e mostrará como as técnicas utilizadas contribuem para melhorar a qualidade e confiabilidade do software.

4.2.6 Entrega e Implantação

No desenvolvimento ágil, a metodologia CI/CD (Integração Contínua e Entrega Contínua) será adotada para otimizar os ciclos de desenvolvimento, proporcionando maior velocidade e eficiência. O CI automatiza a integração de alterações no código-fonte em um repositório central, executando testes automatizados à medida que novas alterações são feitas. Já o CD automatiza a liberação do código validado para ambientes de *staging (teste)* ou produção. Isso permite lançamentos mais frequentes e seguros de novas funcionalidades e melhorias.

5 ANÁLISE E PROJETO

Nesta seção, serão apresentados os resultados preliminares alcançados. Inicialmente, serão descritas as histórias de usuário desenvolvidas com o objetivo de levantar os requisitos necessários para o desenvolvimento da área de criação de OA. A partir dessas informações, as telas foram elaboradas na ferramenta Figma, cada uma representando uma etapa do processo de criação de OA.

5.1 História de usuário

Para o desenvolvimento da nova versão da plataforma, as histórias de usuário coletadas e detalhadas a seguir representam os requisitos funcionais do sistema. Elas servirão como guia para o desenvolvimento, garantindo que a plataforma atenda às necessidades dos usuários. A estrutura das histórias de usuário seguirá o formato apresentado no subitem 4.2.2.

5.1.1 Criação e configuração dos objeto de aprendizagem

As histórias de usuário a seguir ilustram como os educadores irão criar e configurar os OA dentro da plataforma FARMA.

Feature: Criação de um Objeto de Aprendizagem

Como educador, **quero** criar um novo OA **para** que eu possa disponibilizar conteúdos interativos para os alunos.

Dado que sou um educador

Quando eu acessar a plataforma e optar por criar um novo OA

Então deverei ter a opção de inserir conteúdos interativos.

Feature: Inserir título e descrição de um Objeto de Aprendizagem

Como educador, **quero** inserir um título e uma descrição no OA **para** contextualizar o conteúdo apresentado.

Dado que estou criando um OA

Quando eu preencher os campos de título e descrição

Então esses dados devem ser salvos e exibidos corretamente.

Feature: Inserir componentes gráficos em um Objeto de Aprendizagem

Como educador, **quero** adicionar imagens, vídeos e gráficos ao OA **para** tornar a aprendizagem mais visual e interativa.

Dado que estou editando um OA

Quando eu fizer o upload de mídias

Então elas devem ser armazenadas e exibidas corretamente no OA.

Feature: Salvar informações não finalizadas de um Objeto de Aprendizagem

Como educador, **quero** poder salvar o OA como rascunho **para** que eu possa editá-lo antes de publicá-lo.

Dado que estou criando ou editando um OA

Quando eu optar por salvar como rascunho

Então o sistema deve armazená-lo sem disponibilizá-lo para os alunos.

Feature: Editar informações de um Objeto de Aprendizagem

Como educador, **quero** editar um OA já criado **para** atualizar o conteúdo quando necessário.

Dado que tenho um OA salvo

Quando eu acessar a opção de edição e fizer alterações

Então as modificações devem ser aplicadas e salvas corretamente.

Feature: Excluir informações de um Objeto de Aprendizagem

Como educador, **quero** excluir um OA **para** aqueles casos que ele não seja mais relevante ou precise ser removido.

Dado que tenho um OA criado

Quando eu selecionar a opção de exclusão

Então o OA deve ser removido do sistema.

5.1.2 Criação de exercícios, questões e dicas

As histórias de usuário a seguir ilustram como os educadores irão criar os exercícios e questões dentro da plataforma FARMA.

Feature: Criação de Exercício

Como educador, **quero** criar exercícios dentro do OA **para** reforçar o aprendizado dos alunos.

Dado que estou criando um OA

Quando eu acessar a opção de adicionar exercícios

Então devo poder inserir e configurar questões dentro do OA.

Feature: Personalizar perguntas de exercícios

Como educador, **quero** personalizar as perguntas dos exercícios **para** atender às necessidades específicas do conteúdo.

Dado que estou criando um exercício

Quando eu editar o conteúdo da pergunta

Então as modificações devem ser salvas e exibidas corretamente para os alunos.

Feature: Adicionar dicas

Como educador, **quero** adicionar dicas às perguntas **para** ajudar os alunos a encontrar o caminho correto em caso de erro.

Dado que estou configurando um exercício

Quando eu inserir uma dica

Então ela deve ser exibida após o aluno exceder o número de tentativas.

Feature: Configurar números de tentativas para o acionamento de dicas

Como educador, **quero** configurar um número de tentativas por questão **para** que após exceder esse valor, aparecer uma dica.

Dado que estou criando um exercício

Quando eu definir um limite de tentativas

Então o sistema deve apresentar uma dica para os alunos.

5.2 Prototipação de Telas

A seguir, são apresentadas as interfaces de usuário desenvolvidas com base na coleta dos requisitos de interface, conforme as histórias de usuários descritas no subitem 5.1, além de sugestões de interfaces para trabalhos futuros.

5.2.1 Interface de criação de Objeto de Aprendizagem

A seguir, serão apresentadas as interfaces desenvolvidas com base nas histórias de usuários, ilustrando o processo de criação do OA.

Na Figura 14, é possível visualizar a tela inicial, onde são exibidas as OA's criadas. No caso apresentado, nenhuma OA foi criada, portanto, a mensagem "Nenhum OA foi encontrado" é exibida. Ao lado da mensagem, há um campo de busca, útil quando há um grande número de OA's cadastradas e o usuário deseja encontrá-las rapidamente pelo nome. No canto inferior direito da tela, há um botão para adicionar uma nova OA.

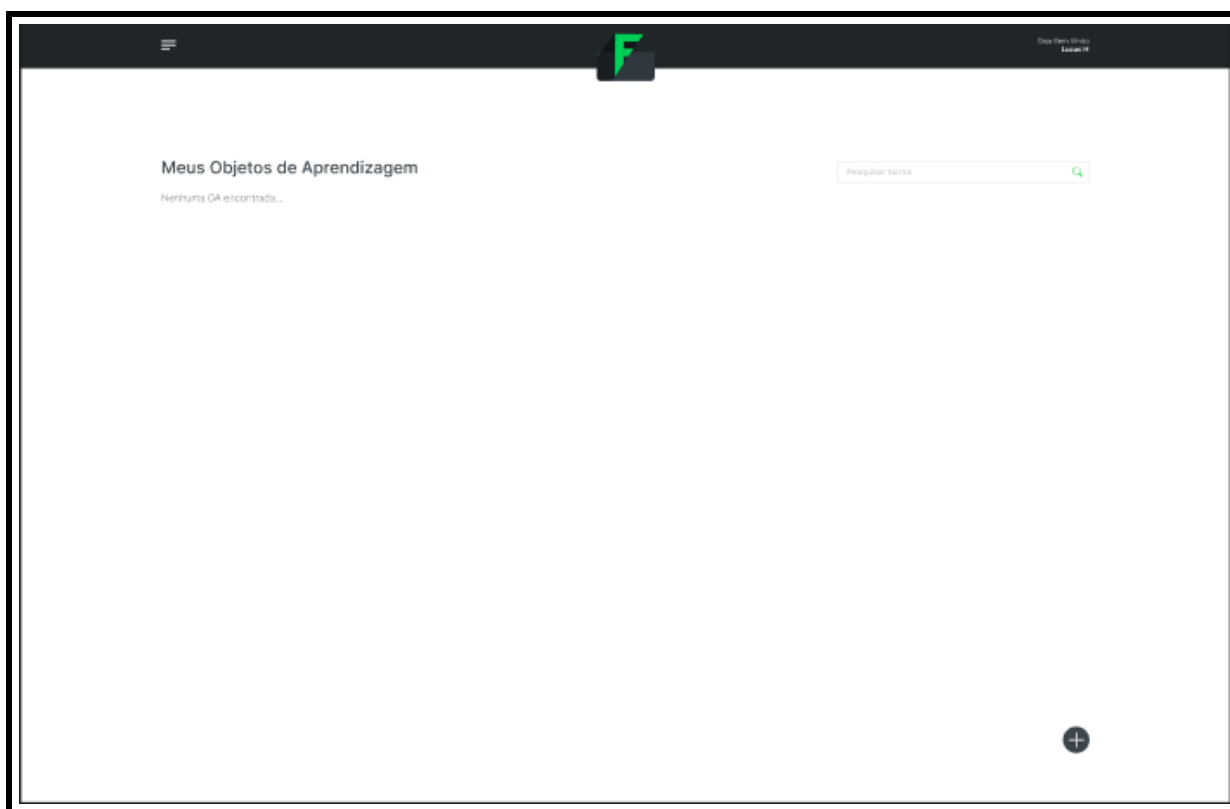


Figura 14 – Tela inicial para visualização de Objetos de Aprendizagem na FARMA (autoria própria)

Na Figura 15, é possível observar a tela de criação de uma OA. Nela, o usuário deve fornecer o título e a descrição da OA, podendo também adicionar uma imagem. Todos os campos, exceto a imagem, são obrigatórios. Na parte superior da tela, há duas opções de navegação: a seta à esquerda permite voltar à página anterior, enquanto o ícone de casa direciona o usuário para a tela inicial da FARMA.

Figura 15 – Tela inicial para criação de Objetos de Aprendizagem na FARMA (autoria própria)

Na Figura 16, vemos a mesma tela apresentada na Figura 14, porém agora com uma OA criada após passar pelo processo descrito na Figura 15. Nesta tela, é possível observar um componente exibindo a imagem inserida, o título e a descrição da OA. Na parte inferior direita, há um botão de acesso que permite visualizar a OA criada.

Na Figura 17, podemos visualizar a tela que exibe os componentes da OA criada. Neste exemplo, não há introduções, exercícios ou dicas cadastrados, portanto, a mensagem "Nenhuma informação encontrada" é exibida. Esses componentes podem ser adicionados utilizando os três botões localizados no canto inferior direito: o primeiro botão adiciona uma nova introdução, o segundo permite criar um novo exercício, e o último botão adiciona uma dica.

Nas Figuras 18, 19 e 20, observamos uma estrutura semelhante para a criação de diferentes componentes da OA. Na criação de uma introdução, é necessário adicionar um título, o conteúdo e uma imagem (*opcional*). Em seguida, na tela de criação de exercícios, é possível inserir um título, uma descrição do exercício, uma imagem opcional e, por fim, a resposta esperada. Por último, na tela de criação de dicas, o usuário pode adicionar o texto da dica e definir o número de tentativas necessárias para que a dica seja acionada.

Na Figura 21, vemos uma tela semelhante à Figura 17, mas com os componentes já criados. Agora, cada componente é exibido com seu título, descrição e opções específicas. No canto superior direito, o ícone à esquerda refere-se à opção de editar, enquanto o ícone à direita permite excluir o componente. Além disso, há um novo botão acima, simbolizado por um "olho",

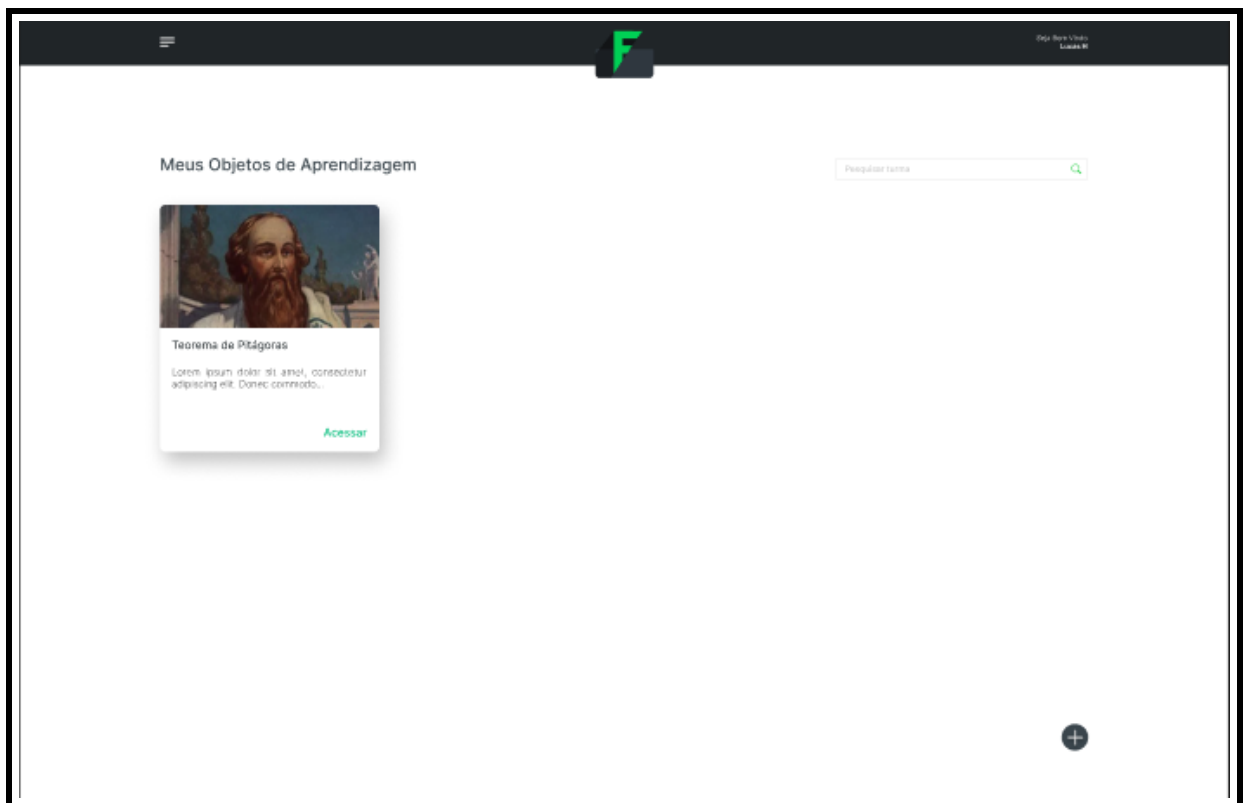


Figura 16 – Tela inicial para visualização de Objetos de Aprendizagem já criada na FARMA (autoria própria)

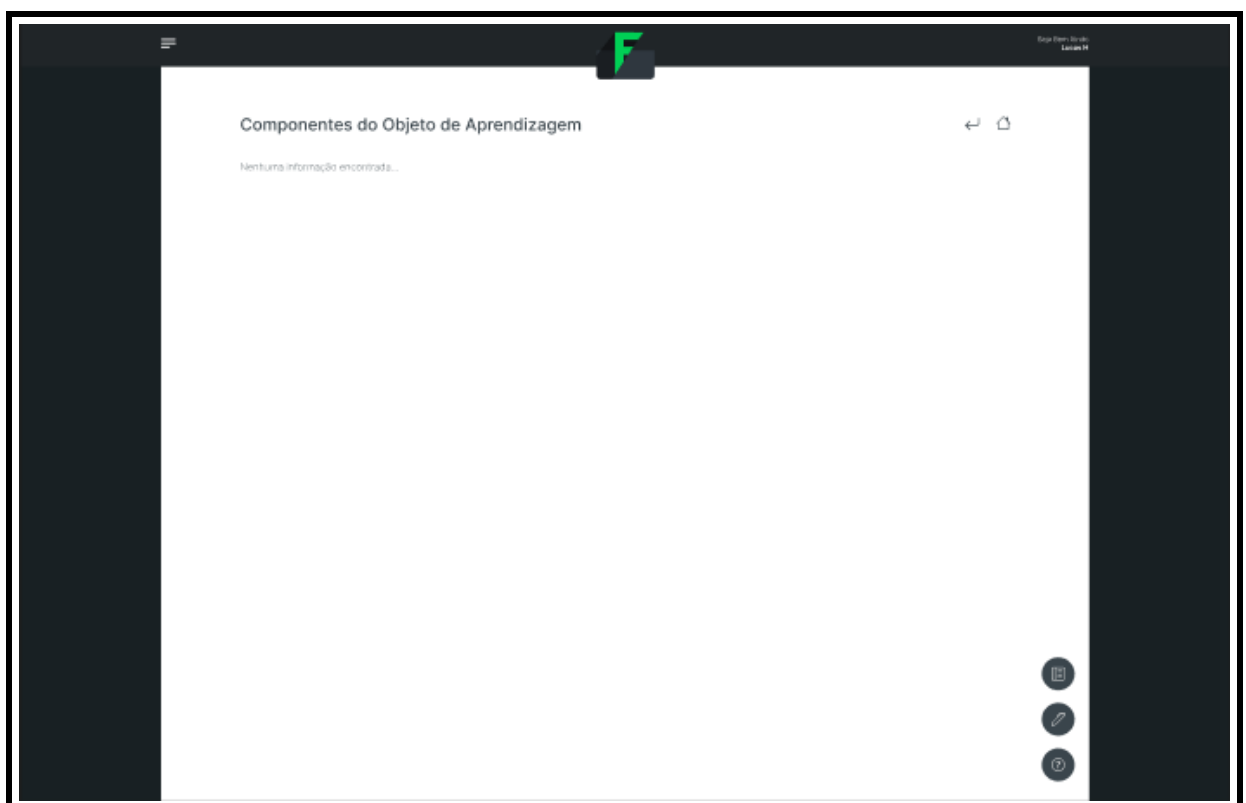


Figura 17 – Tela para visualização de componentes de um Objeto de Aprendizagem (autoria própria)

Introdução

Título
Insira o título do conteúdo abordado

Conteúdo
Descreva o conteúdo abordado

Adicionar Imagem

AVANÇAR

Figura 18 – Tela para criação de introdução de um Objetos de Aprendizagem (autoria própria)

Exercício

Título
Insira o título do exercício

Conteúdo
Descreva o conteúdo abordado

Adicionar Imagem

Resposta correta
Insira a resposta correta do exercício

AVANÇAR

Figura 19 – Tela para criação de exercício de um Objetos de Aprendizagem (autoria própria)

Figura 20 – Tela para criação de Dicas de um Objetos de Aprendizagem (autoria própria)

que serve para pré-visualizar o componente. Esta funcionalidade de pré-visualização pode ser vista nas próximas figuras.

Na Figura 22, é possível visualizar a tela de pré-visualização da interface da OA.

5.2.2 Interface de contribuição para trabalhos futuros

Além das telas apresentadas, foram desenvolvidos outros protótipos para auxiliar em trabalhos futuros e garantir a consistência do design. Na Figura 23, podemos ver a tela de login. Em seguida, na Figura 24, temos a tela de registro. A Figura 25, apresenta a tela inicial do sistema, e, por fim, a Figura 26, exibe a tela de acesso à conta do usuário.

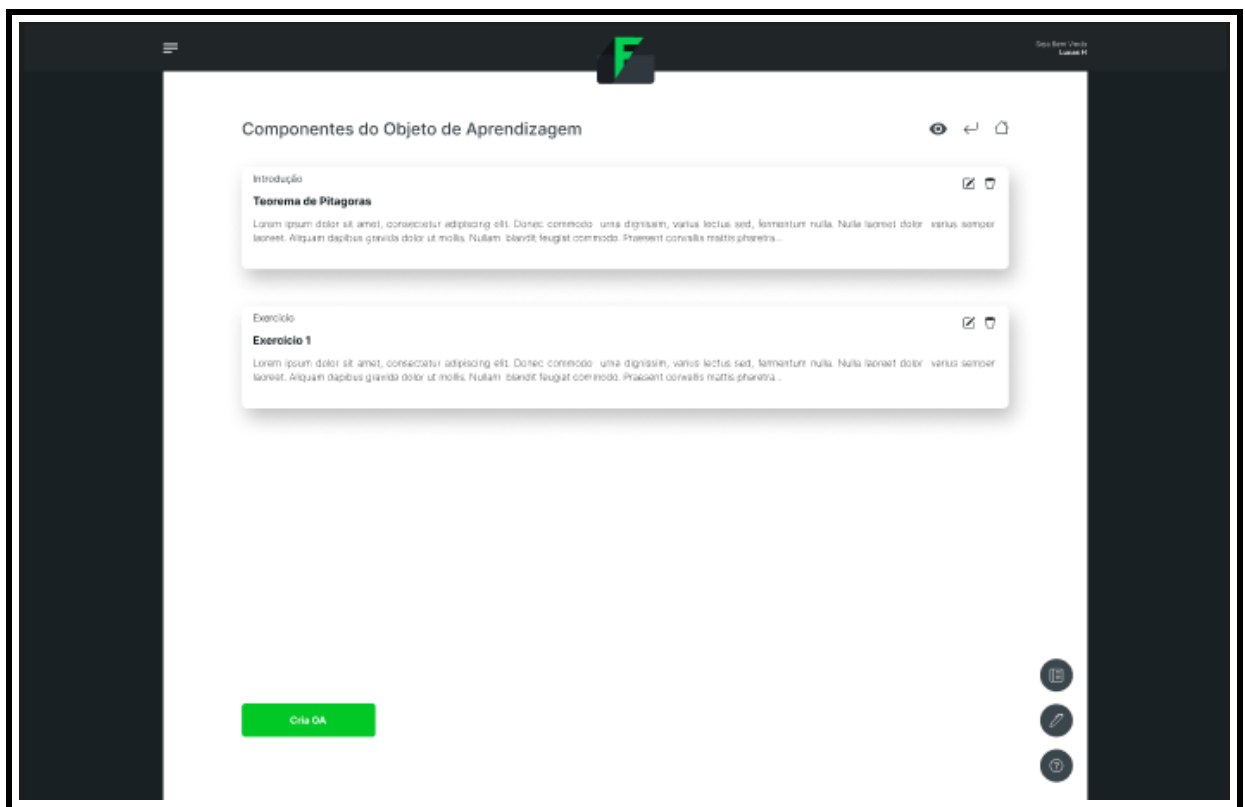


Figura 21 – Tela para visualização de componentes de um Objeto de Aprendizagem, agora criados (autoria própria)

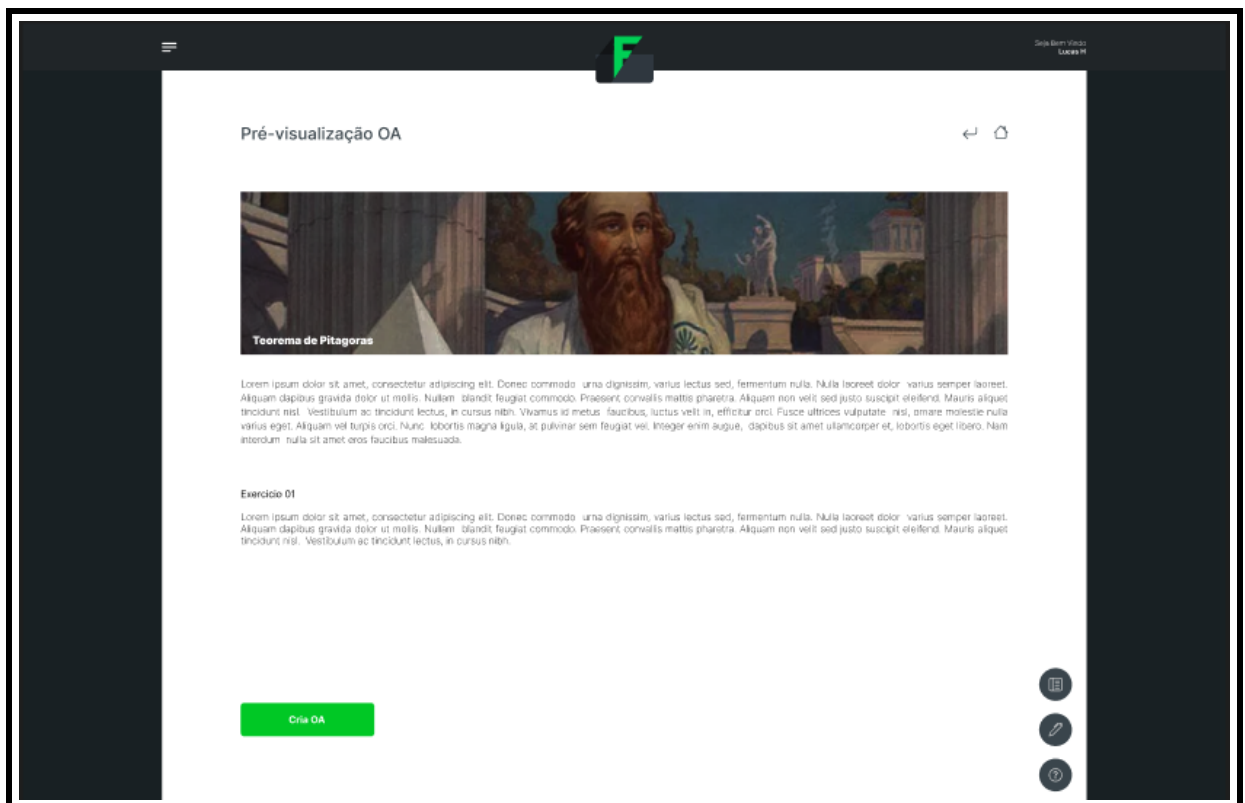
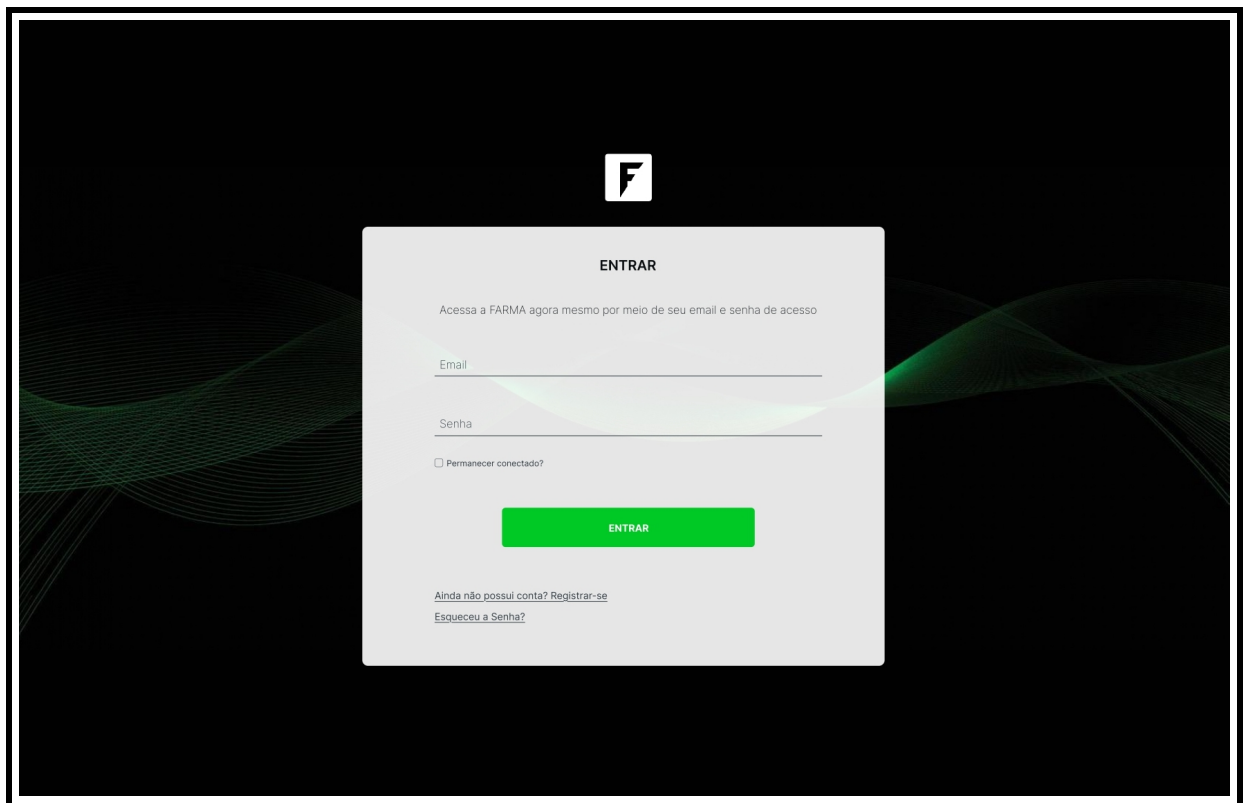
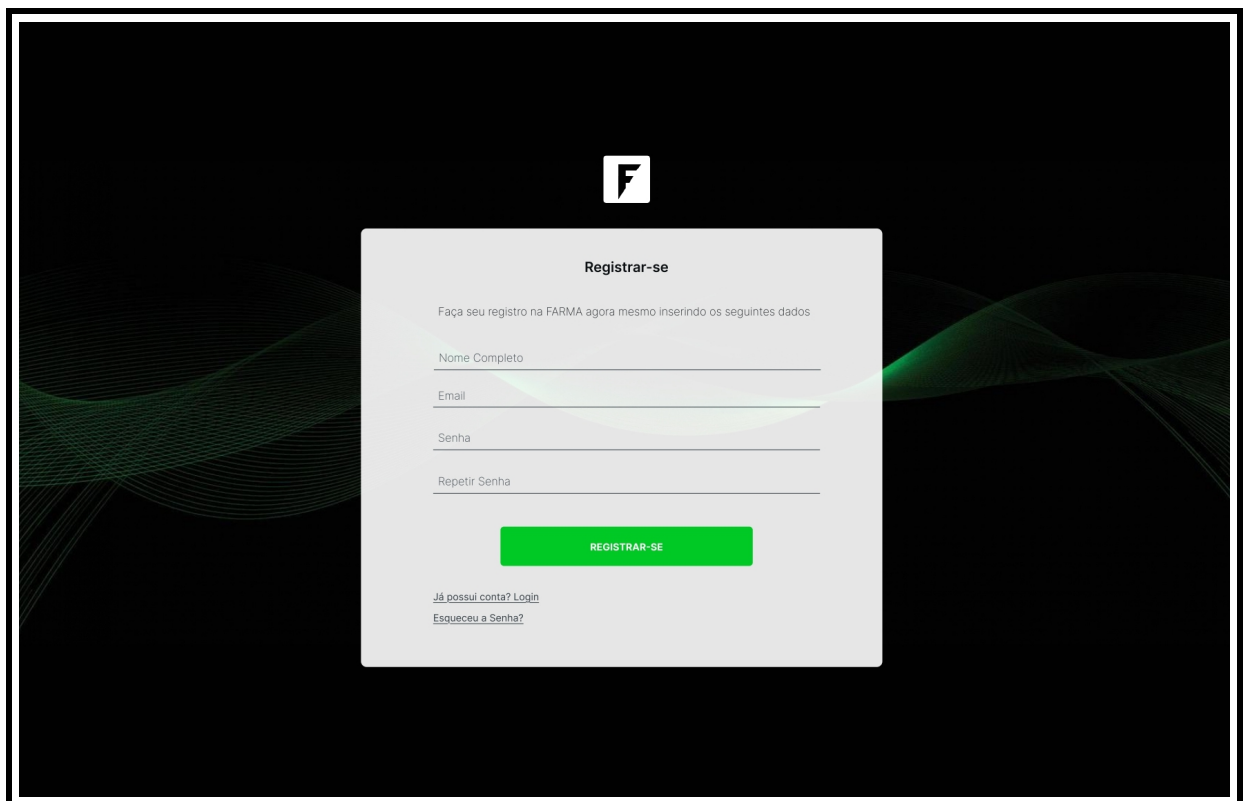


Figura 22 – Tela para pré-visualização do Objeto de Aprendizagem (autoria própria)



The login screen features a dark background with a subtle green wave pattern. At the top center is the FARMA logo, a white 'F' inside a square. Below the logo is a white rectangular form titled 'ENTRAR'. Inside the form, the text 'Acessa a FARMA agora mesmo por meio de seu email e senha de acesso' is displayed. There are two input fields: 'Email' and 'Senha'. Below these is a checkbox labeled 'Permanecer conectado?'. A large green button labeled 'ENTRAR' is positioned below the checkbox. At the bottom of the form, there are two links: 'Ainda não possui conta? Registrar-se' and 'Esqueceu a Senha?'.

Figura 23 – Tela de login da nova versão da FARMA (autoria própria)



The registration screen has the same dark background and green wave pattern as the login screen. It features the FARMA logo at the top center. Below the logo is a white rectangular form titled 'Registrar-se'. The text 'Faça seu registro na FARMA agora mesmo inserindo os seguintes dados' is shown. There are four input fields: 'Nome Completo', 'Email', 'Senha', and 'Repetir Senha'. A large green button labeled 'REGISTRAR-SE' is located below the input fields. At the bottom of the form, there are two links: 'Já possui conta? Login' and 'Esqueceu a Senha?'.

Figura 24 – Tela de criação de conta da nova versão da FARMA (autoria própria)



Figura 25 – Tela home da nova versão da FARMA (autoria própria)

Perfil de Usuário:

[Precisa de ajuda?](#)
[Alterar senha](#)

Formulário de Dados:

Nome Completo:

E-mail:

Contato:

Escolaridade:

Localização: Estado:

Cidade: Endereço:

[SALVAR](#)

Figura 26 – Tela de acesso às informações de usuário da nova versão da FARMA (autoria própria)

6 CONSIDERAÇÕES FINAIS

A ferramenta FARMA foi desenvolvida para testar, avaliar e corrigir erros dos estudantes por meio de OA. No entanto, a versão atual faz uso de uma API, o que pode gerar maior complexidade na manutenção, dificultar o gerenciamento e impactar o desempenho da aplicação. Além de uma interface gráfica desatualizada na versão original ¹ e com falta do desenvolvimento das áreas de criação de OA na versão FARMA Reborn ².

Com isso, esse trabalho busca contribuir para a nova versão da FARMA, propondo a reformulação de sua estrutura para uma arquitetura monolítica. Essa mudança visa simplificar o ambiente de desenvolvimento, melhorar o desempenho e facilitar a manutenção do sistema. Além disso, será desenvolvida uma nova interface gráfica, aprimorando a experiência do usuário em cada etapa da criação de Objetos de Aprendizagem.

A transição de uma arquitetura distribuída para uma monolítica exige uma remodelagem significativa do código, garantindo que todas as funcionalidades sejam preservadas e otimizadas. Com essa reformulação, espera-se tornar a FARMA mais acessível, eficiente e de fácil gerenciamento. A nova arquitetura monolítica proporcionará maior simplicidade no desenvolvimento e manutenção, enquanto a interface gráfica aprimorada contribuirá para uma melhor experiência do usuário, tornando a criação de Objetos de Aprendizagem mais intuitiva e eficaz.

¹ Disponível em <https://farma.educacional.mat.br/>

² Disponível em <https://farma-reborn.educacional.mat.br/>

REFERÊNCIAS

- ARRUDA, S. Mudanças nas ideias prévias sobre ponto, reta e plano por meio da interação com a ferramenta farma. **Revista Brasileira de Ensino de Ciência e Tecnologia**, 2021.
- AWARI., S. **As Vantagens E Desvantagens Do Figma: Tudo O Que Você Precisa Saber Sobre Essa Ferramenta De Design**. [S.l.]: <https://awari.com.br/as-vantagens-e-desvantagens-do-figma-tudo-o-que-voce-precisa-saber-sobre-essa-ferramenta-de-design>, 2023.
- BASS, L. **Software architecture in practice**. Addison-Wesley, 2013. Disponível em: <https://www.amazon.com.br/Software-Architecture-Practice-Len-Bass/dp/0321815734>.
- BAZZO, G. **CLASSIFICAÇÃO AUTOMÁTICA DE ERROS DE APRENDIZES HUMANOS DO PROCESSO DE INDUÇÃO ANALÍTICA**. 2011. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2011.
- BONIN, D. Implementação de uma api para visualização e interação com objetos de aprendizagem para ferramenta de autoria farma. 2024.
- CASTILHO, A. **Objetos de Aprendizagem com Feedbacks para a Autorregulação da Aprendizagem de Conceitos Matemáticos Necessários para o Cálculo Diferencial e Integral**. 2019. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2019.
- COSTA, D. **ATAUDIW - Uma Ferramenta de Autoria para Auxiliar o Uso da Lousa Digital Interativa**. 2017. Tese (Doutorado) — Universidade Federal de Goiás - Regional Jataí Instituto de Ciências Exatas e Tecnológicas (ICET), Goiás, 2017.
- DAMASCENO, M. d. G. d. A. *et al.* Potencializando a aprendizagem: A contribuição da neurociência e das tecnologias educacionais. p. 2409–2422, jul. 2024. Disponível em: <https://periodicorease.pro.br/rease/article/view/14983>.
- FOWLER, M. **Patterns of Enterprise Application Architecture**. Bookman, 2006. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788577800643/pageid/19>.
- KLEPPMANN, M. **Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems**. O REILLY, 2017. Disponível em: <https://www.amazon.com.br/Designing-Data-Intensive-Applications-Martin-Kleppmann/dp/1449373321>.
- KUTZKE, A. **Informática educacional e a mediação do erro na educação : um estudo teórico-crítico e uma proposta de instrumento computacional**. 2015. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2015.
- LEITE, M. **Arquitetura para remediação de erros baseada em múltiplas representações externas**. 2013. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2013.
- LEITE, M. Otimizando o processo ensino e aprendizagem com a arquitetura para desenvolvimento de objetos de aprendizagem. **Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015)**, 2015.
- MARCZAL, D. **UM ARCABOUÇO QUE ENFATIZA A RETROAÇÃO A CONTEXTOS DE ERRO DURANTE O ACESSO A CONTEÚDOS EDUCACIONAIS**. 2010. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2010.

MARCZAL, D. **Classificação automática de erros de aprendizes humanos do processo de indução analítica**. 2011. Tese (Doutorado) — Universidade Tecnológica Federal do Paraná, Guarapuava - Paraná, 2011.

MARCZAL, D. **FARMA: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos**. 2014. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2014.

MARCZAL, D. Farma: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. **C3SL – Departamento de Informática – Universidade Federal do Paraná (UPFR)**, 2015.

MARCZAL, D. Metodologia e software educacional para a investigação e remediação de erros conceituais em matemática. **Revista Brasileira de Informática na Educação**, 2016.

MOURA, V. **AValiação do impacto da retroação na aprendizagem apoiada por uma ferramenta educacional**. 2015. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2015.

NEWMAN, S. **Building Microservices**. O REILLY, 2021. Disponível em: <https://www.amazon.com/Building-Microservices-Designing-Fine-Grained-Systems/dp/1491950358>.

RAMOS, J. Visualização de objetos de aprendizagem para a ferramenta de autoria farma. 2019.

RICHARDS, M. **Fundamentos da arquitetura de Software: Uma abordagem de engenharia**. Alta Books, 2024. Disponível em: [https://app.minhabiblioteca.com.br/reader/books/9788550819754/epubcfi/6/2\[%3Bvnd.vst.idref%3Dcover\]!/4/2/2%4050:2](https://app.minhabiblioteca.com.br/reader/books/9788550819754/epubcfi/6/2[%3Bvnd.vst.idref%3Dcover]!/4/2/2%4050:2).

ROTEM, A. Explicando as falácias (ou falsas verdades) da computação distribuída. **Icvdata**, 2014.

SANTOS, E. **Projeto de uma nova interface gráfica para a ferramenta de autoria de objetos de aprendizagem matemáticos FARMA**. 2018. Tese (Doutorado) — Universidade Tecnológica Federal do Paraná, Guarapuava - Paraná, 2018.

SILVA, R. **SEQUENCIAMENTO ADAPTATIVO DE EXERCÍCIOS BASEADO NA CORRESPONDÊNCIA ENTRE A DIFICULDADE DA SOLUÇÃO E O DESEMPENHO DINÂMICO DO APRENDIZ**. 2015. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba - Paraná, 2015.

SILVA, R. Adaptabilidade de objetos de aprendizagem usando calibragem e sequenciamento adaptativo de exercícios. **Revista Brasileira de Informática na Educação – RBIE**, 2018.

SOUSA, R. **Tecnologias Digitais na Educação**. SciELO, 2011. Disponível em: <https://static.scielo.org/scielobooks/6pdyn/pdf/sousa-9788578791247.pdf>.

TAROUCO, L. **Objetos de aprendizagem : teoria e prática**. Evangraf, 2014. Disponível em: <http://hdl.handle.net/10183/102993>.

TRIBECK, P. Edutracsys como objeto de aprendizagem na remediação de erros. **Espacios**, 2017.

VITEK, I. Desenvolvimento da Área destinada À criação de objetos de aprendizagem para ferramenta de autoria farma. 2023.