



UNIVERSIDADE FEDERAL DE PERNAMBUCO



CENTRO DE INFORMÁTICA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

---

**Engenharia de Software Educacional e Design  
Instrucional: Otimização de Processos no  
Desenvolvimento de Cursos Digitais para a Plataforma  
AVAMEC**

**Trabalho de Graduação**

**Aluno(a):** Gabriel de Oliveira Sousa ([gos2@cin.ufpe.br](mailto:gos2@cin.ufpe.br))

**Orientador(a):** Prof. Dr. Alex Sandro Gomes ([asg@cin.ufpe.br](mailto:asg@cin.ufpe.br))

**Área:** Engenharia de software e Design Instrucional

Agosto de 2025

Gabriel de Oliveira Sousa

**Engenharia de Software Educacional e Design Instrucional: Otimização de Processos  
no Desenvolvimento de Cursos Digitais para a Plataforma AVAMEC**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Ciência da  
Computação da Universidade Federal de  
Pernambuco, como requisito parcial para  
obtenção do título de bacharel em Ciência da  
Computação.

Orientador (a): Prof. Dr. Alex Sandro Gomes

Recife

2025

Ficha de identificação da obra elaborada pelo autor,  
através do programa de geração automática do SIB/UFPE

Sousa, Gabriel de Oliveira.

Engenharia de Software Educacional e Design Instrucional: Otimização de Processos no Desenvolvimento de Cursos Digitais para a Plataforma AVAMEC / Gabriel de Oliveira Sousa. - Recife, 2025.

92 p. : il., tab.

Orientador(a): Alex Sandro Gomes

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado, 2025.

Inclui referências, apêndices, anexos.

1. Engenharia de Software. 2. Design Instrucional. 3. cursos digitais. 4. otimização de processos. 5. arquitetura de software. 6. desenvolvimento web. I. Gomes, Alex Sandro. (Orientação). II. Título.

000 CDD (22.ed.)

Gabriel de Oliveira Sousa

**Engenharia de Software Educacional e Design Instrucional: Otimização de Processos  
no Desenvolvimento de Cursos Digitais para a Plataforma AVAMEC**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Graduação em Ciência da  
Computação da Universidade Federal de  
Pernambuco, como requisito parcial para  
obtenção do título de bacharel em Ciência da  
Computação.

Aprovado em: 08 / 08 / 2025

**BANCA EXAMINADORA**

---

Prof. Dr. Alex Sandro Gomes (Orientador)

Universidade Federal de Pernambuco

---

Profa. Dra. Carla Taciana Lima Lourenco Silva (Examinadora Interno)

Universidade Federal de Pernambuco

## Resumo

Ao longo dos anos, a crescente busca por cursos online tem sido notável, com a plataforma AVAMEC (Ambiente Virtual de Aprendizagem do Ministério da Educação) destacando-se como um recurso amplamente procurado por professores em formação continuada. Atento a essa demanda, o time AVAMEC do V-Lab, na UFPE, identificou oportunidades para aprimorar o desenvolvimento de cursos digitais de alta qualidade para esta plataforma, que se concentrava principalmente na velocidade de desenvolvimento e refatoração completa da arquitetura do projeto levantada a partir de indicadores de incidência de erros durante o desenvolvimento dos cursos. Através da união entre engenharia de software e boas práticas de design instrucional, foi possível atingir um novo patamar de desenvolvimento de software, conforme detalhado neste trabalho por meio dos resultados obtidos dentro da plataforma AVAMEC. A implementação de modernas tecnologias web otimizou significativamente os processos, enquanto a aplicação de metodologias ágeis, como o Scrum, garantiu a integração eficaz de técnicas de design instrucional para criar um material que possa atender às necessidades pedagógicas dos alunos. Como resultado direto dessa otimização, o desenvolvimento dos cursos tornou-se mais célere e a incidência de erros no ambiente de trabalho diminuiu como mostram os dados coletados do Github, estabelecendo um modelo que possa vir beneficiar outras instituições de ensino no Brasil na criação de cursos de excelência para a plataforma AVAMEC.

**Palavras-chave:** Engenharia de Software, Design Instrucional, AVAMEC, cursos digitais, otimização de processos, desenvolvimento web, arquitetura de software.

## Abstract

Over the years, the growing search for online courses has been notable, with the AVAMEC platform (the Ministry of Education's Virtual Learning Environment) standing out as a widely sought-after resource for teachers in continuing education. Attentive to this demand, the AVAMEC team from V-Lab at UFPE identified opportunities to improve the development of high-quality digital courses for this platform, which focused mainly on development speed and a complete refactoring of the project's architecture, prompted by indicators of error incidence during course development. Through the combination of software engineering and best practices in instructional design, it was possible to reach a new level of software development, as detailed in this work through the results obtained within the AVAMEC platform. The implementation of modern web technologies significantly optimized processes, while the application of agile methodologies, such as Scrum, ensured the effective integration of instructional design techniques to create material that fully meets the students pedagogical needs. As a direct result of this optimization, course development became more swift and the incidence of errors in the work environment decreased, as shown by the data collected from GitHub, establishing a model that can benefit other educational institutions in Brazil in creating excellent courses for the AVAMEC platform.

**Keywords:** Software Engineering, Instructional Design, AVAMEC, digital courses, process optimization, web development, software architecture.

# Sumário

<b>1 Introdução.....</b>	<b>9</b>
1.1. Contexto e Justificativa.....	9
1.2. Objetivos do Estudo.....	11
1.3. Conclusão.....	12
<b>2. Design Instrucional e Engenharia de Software: Estado da Arte, Abordagens e Estudos Precedentes.....</b>	<b>13</b>
2.1 Estado da Arte.....	13
2.2 Quem Já Fez.....	13
2.3 Abordagens Existentes.....	14
<b>3. Metodologia.....</b>	<b>14</b>
3.1 Estudo de Caso.....	14
3.2 Objetivo Geral.....	15
3.3. Descrição do processo atual.....	15
3.4. Como este estudo pode ser reproduzido.....	16
3.4. Coleta e análise de Dados.....	17
<b>4. Contexto e Justificativa.....</b>	<b>19</b>
4.1. Descrição da Arquitetura Anterior.....	19
4.1.1 Contexto Tecnológico.....	19
4.1.2 Estrutura e análise da arquitetura anterior.....	22
4.1.3 Descrição da Imagem da Arquitetura Anterior.....	23
4.1.4 Análise reflexiva.....	24
4.1.5 Análise de Design Instrucional com a nova arquitetura.....	25
4.1.6 Desafios e Limitações da arquitetura anterior.....	26
4.1.7 Estratégias Adotadas e Limitações Correntes.....	28
4.1.8 Conclusão.....	29
2.3. Transição para Novas Tecnologias.....	30
2.3.1 Escolha Tecnológica e Design Instrucional.....	30
2.3.2 React: Uma Biblioteca JavaScript para Construção de Interfaces de Usuário....	30
2.3.3 ModfyJS: Um boilerplate em ReactJS.....	32
2.3.4 TypeScript: Um Superconjunto Tipado de JavaScript.....	40
2.3.5 Tailwind CSS: Framework de Utilitários CSS.....	42
2.3.6 Estrutura da nova arquitetura.....	46
2.3.7 Planejamento e Treinamento.....	49
2.3.8 Desafios Integrados.....	50
2.3.9 Design Instrucional na Transição.....	51
4.2. Otimização de Processos.....	55

4.2.1 Definição de Processo.....	55
4.2.2 Scrum e a Otimização de Processos.....	55
4.2.2.1 Planejamento de Sprints.....	56
4.2.3 Aplicação da otimização de processos.....	57
4.2.4 Integração de Design Instrucional.....	63
4.2.5 Resultados da Otimização de Processos.....	66
4.2.6 Conclusão.....	66
4.2.7 Contribuições do Design Instrucional na otimização de processos.....	66
<b>5. Resultados.....</b>	<b>70</b>
5.1 Análise dos Dados de Desenvolvimento.....	70
- PCR: Plano de Carreira e Remuneração.....	72
5.2 Discussão.....	72
5.2.1 Redução do Tempo de Desenvolvimento.....	72
5.2.2 Aumento da Qualidade do Código.....	72
5.2.3 Experiências de Aprendizagem Mais Engajadoras.....	73
5.2.4 Impacto na Quantidade de Alunos.....	73
5.3 Análise de Commits.....	73
Conclusão.....	73
<b>6. Considerações Finais e Limitações.....</b>	<b>74</b>
6.1 Contribuições do Estudo.....	74
6.2 Limitações do Estudo.....	75
6.3 Sugestões para Trabalhos Futuros.....	75
<b>7. Referências.....</b>	<b>76</b>
<b>8. Apêndices.....</b>	<b>78</b>
8.1 Apêndice A. Tecnologia da Arquitetura Anterior.....	78
8.2 Apêndice B. Tecnologia da Arquitetura Nova.....	78



# Lista de figuras

**Figura 1** – Captura de tela de código exemplo de uso de HTML - 20

**Figura 2** – Captura de tela de código exemplo de uso de CSS - 21

**Figura 3** – Captura de tela de código exemplo de uso de JavaScript - 22

**Figura 4** – Captura de tela da arquitetura do sistema antigo - 23

**Figura 5** – Componente Linha do Tempo - 24

**Figura 6** – Módulo Avaliativo do curso Plano de Carreira e Remuneração (PCR) - 25

**Figura 7** – ReactJS Tutorial - 31

**Figura 8** – Código base de criação de cursos para o AVAMEC - 35

**Figura 9** – Código base de criação de cursos para o AVAMEC - 36

**Figura 10** – Captura de tela de código base de criação de cursos para o AVAMEC - 37

**Figura 11** – Código base de criação de cursos para o AVAMEC - 38

**Figura 12** – Captura de tela de código base de criação de cursos para o AVAMEC - 39

**Figura 13** – Arquitetura do boilerplate ModfyJS para ReactJS - 40

**Figura 14** – What is Typescript? - 41

**Figura 15** – Captura de tela de código HTML exemplo para comparação com TailwindCSS - 44

**Figura 16** – Captura de tela de código Tailwind exemplo para comparação com HTML - 45

**Figura 17** – Nova arquitetura de software para criação de cursos AVAMEC otimizada - 47

**Figura 18** – Componente feito com arquitetura avançada - 54

**Figura 19** – Componente feito com arquitetura avançada - 54

**Figura 20** – Componente feito com arquitetura avançada - 55

**Figura 21** – SCRUM: entenda a metodologia utilizada na Gestão de Projetos - 57

**Figura 22** – Diagrama de funcionamento de processos versão 1.0 - 58

**Figura 23** – Diagrama de funcionamento de processos versão 2.0 - 62

**Figura 24** – Diagrama de funcionamento de processos versão 3.0 - 65

**Figura 25** – ADDIE: 5 Steps To Effective Training - 67

## Lista de tabelas

**Tabela 1** – Tabela de Dados de Desenvolvimento - 71

**Tabela 2** – Tabela de Quantidade de Alunos e seu impacto - 71

# 1 Introdução

## 1.1. Contexto e Justificativa

A procura crescente por cursos digitais de alta qualidade – impulsionada por um público que, em número crescente, prefere alternativas on-line aos modelos presenciais – expõe a necessidade de otimizar os processos de produção desses cursos. No âmbito do AVAMEC (Ambiente Virtual de Aprendizagem do Ministério da Educação), tal demanda por eficiência e inovação mostra-se permanente.

O AVAMEC, acessível em <https://avamec.mec.gov.br/>, consolidou-se como ambiente estratégico para a difusão de conhecimento em grande escala, notadamente no cenário educacional brasileiro. Amplamente adotado por instituições vinculadas ao MEC, ele hospeda predominantemente cursos autoinstrucionais que permitem aos docentes em formação continuada aprender de modo independente e no seu próprio ritmo.

Nesse contexto, o laboratório de inovação da UFPE, o V-Lab [34], atende a órgãos públicos de grande porte – entre eles o MEC [35], a CAPES [36] e o Conselho Nacional de Justiça [37] – por meio de equipes dedicadas. O time AVAMEC, responsável por diagramar, ilustrar, desenvolver, testar e publicar os cursos demandados pelo MEC, historicamente empregava HTML [1], CSS [39] e JavaScript [40]. Embora consolidadas, tais tecnologias impõem limitações evidentes em projetos de alta complexidade: manutenção difícil, baixa reutilização de código e propensão a erros que alongam prazos de entrega.

A adoção de ferramentas contemporâneas – React [41], ModfyJS [42] (boilerplate em React desenvolvido pelo V-Lab), TypeScript [43] e Tailwind CSS [44] – representa a resposta a esse quadro. Tais tecnologias favorecem modularização, reuso de componentes e manutenção simplificada. React estrutura interfaces de forma coesa; TypeScript acrescenta tipagem estática, reduzindo equívocos; Tailwind CSS confere padronização à camada de estilo. Resultados expressivos, discutidos adiante, evidenciam ganhos diretos na qualidade dos cursos.

Ainda assim, a mera introdução de novas ferramentas não basta. É imperativo alinhar essas inovações às boas práticas da engenharia de software e do design instrucional. Este último dedica-se a planejar experiências de aprendizagem eficazes, definindo necessidades,

objetivos e métodos de avaliação. Quando integrado à engenharia de software – que pode empregar, por exemplo, metodologias ágeis como Scrum – obtém-se não só eficiência técnica, mas também solidez pedagógica.

A documentação sistemática do processo, aliada a métricas baseadas nos repositórios de código, possibilita mensurar progresso e qualidade. Ferramentas de análise estática e testes automatizados contribuem para monitorar a evolução dos indicadores. Assim, tecnologia e pedagogia convergem para viabilizar experiências formativas de elevado valor.

## **1.2. Objetivos do Estudo**

Este trabalho busca identificar e aplicar, no desenvolvimento de cursos para o AVAMEC, as melhores práticas de engenharia de software e design instrucional, avaliando-as à migração de HTML/CSS/JavaScript para React, TypeScript e Tailwind CSS. Para tanto, definiram-se três objetivos específicos:

1. **Analisar os processos atuais de desenvolvimento** – Documentar cada etapa, apontar gargalos e levantar métricas (tempo, qualidade, incidentes) por meio de revisões de código, inspeções arquiteturais e dados extraídos do GitHub.
2. **Implantar e avaliar novas tecnologias** – Introduzir React, ModfyJS, TypeScript e Tailwind CSS, mensurando velocidade de implementação, facilidade de manutenção e redução de falhas em comparação ao método anterior.
3. **Mensurar impactos e resultados** – Aplicar métricas quantitativas para aferir ganhos de eficiência e diminuição do tempo de desenvolvimento.
4. **Avaliação** - Investigar a hipótese de que a nova arquitetura de software otimiza a aplicação de princípios de design instrucional, avaliando seu potencial para viabilizar a criação de experiências de aprendizagem mais ricas e eficazes.
5. **Análise de necessidades** - Investigar as origens da demanda por otimização, por meio do mapeamento dos desafios técnicos (no âmbito da Engenharia de Software) e das limitações pedagógicas (no campo do Design Instrucional) que eram presentes no processo anterior.

## **1.3. Contribuições esperadas**

A integração entre engenharia de software e design instrucional almeja instaurar, no AVAMEC, um processo de produção de cursos simultaneamente ágil e pedagógico. Pretende-se não somente acelerar entregas, mas elevar substancialmente a qualidade formativa. Espera-se, ao término, disponibilizar um modelo replicável por outras instituições que busquem aprimorar iniciativas de educação a distância.

#### **1.4. Estrutura do documento**

Este trabalho está organizado em seis seções principais, que buscam apresentar desde a fundamentação teórica até a análise dos resultados práticos.

**A Seção 2, "Design Instrucional e Engenharia de Software"**, estabelece o referencial teórico que fundamenta a pesquisa. Nela, são explorados os conceitos centrais de cada disciplina, analisando-se o estado da arte, as abordagens consolidadas e estudos de caso relevantes que demonstram a convergência entre a pedagogia e a tecnologia na produção de cursos digitais.

**A Seção 3, "Metodologia"**, detalha o percurso investigativo adotado. Nela, apresenta-se o estudo de caso como abordagem de pesquisa, descrevendo o contexto do projeto AVAMEC, os procedimentos para a coleta e análise dos dados quantitativos e as ferramentas empregadas. Adicionalmente, a seção fornece um guia para a replicação do estudo, abarcando tanto os aspectos tecnológicos quanto os processuais.

**A Seção 4, "Desenvolvimento da solução"**, constitui o núcleo prático do trabalho. Nela, é detalhada a transição da arquitetura legada para a nova stack tecnológica, contrastando as limitações do modelo anterior com os benefícios da abordagem moderna. A seção explora a implementação de ferramentas como React, TypeScript e Tailwind CSS, bem como a otimização dos fluxos de trabalho a partir da adoção de metodologias ágeis e da integração com o design instrucional.

**A Seção 5, "Resultados"**, apresenta e analisa as evidências empíricas coletadas. Por meio de métricas quantitativas, como tempo de desenvolvimento, número de commits e quantidade de alunos matriculados, a seção compara o desempenho dos processos antes e depois da intervenção, aferindo os ganhos de eficiência e qualidade alcançados.

**Por fim, a Seção 6, "Considerações Finais e Limitações",** sintetiza as conclusões da pesquisa. Nela, são recapituladas as principais contribuições do trabalho para a otimização de processos no AVAMEC, ao mesmo tempo em que se reconhecem as limitações do estudo. A seção encerra com sugestões para investigações futuras, visando aprofundar e expandir os resultados obtidos.

## 2. Design Instrucional e Engenharia de Software: Estado da Arte, Abordagens, Tecnologias Modernas e Estudos Precedentes

A produção de cursos digitais de excelência, como aqueles voltados para a plataforma AVAMEC, impõe um desafio que não pode ser vencido por uma única disciplina. O sucesso de um projeto dessa natureza depende de um diálogo necessário entre dois campos de conhecimento distintos, mas complementares. De um lado, o Design Instrucional (DI) oferece a visão pedagógica para arquitetar a aprendizagem; do outro, a Engenharia de Software (ES) fornece o rigor técnico para construir a tecnologia que a suporta. Para os fins deste trabalho, é crucial delinear os contornos de cada uma dessas áreas, pois é da sua confluência que emergem os processos otimizados aqui estudados.

### 2.1 Design Instrucional: A Arquitetura da Aprendizagem

O Design Instrucional é fundamentalmente uma abordagem metódica. Ele se ocupa de todo o ciclo de vida de uma solução de aprendizagem, desde o planejamento e desenvolvimento até a sua implementação e avaliação, sempre com o objetivo de tornar a aquisição de competências uma experiência eficaz e motivadora [45, 46]. Seu trabalho vai muito além de uma simples curadoria de conteúdo. Ao se valer de teorias da aprendizagem, o DI busca criar percursos que respondam às necessidades reais dos alunos, com metas de aprendizagem bem definidas e resultados que possam ser observados.

Essa abordagem ganha especial relevância no domínio dos cursos online. Sem um projeto instrucional sólido, uma plataforma educacional corre o risco de se tornar um mero repositório de arquivos. Modelos como o ADDIE (Análise, Desenho, Desenvolvimento, Implementação e Avaliação) surgem como ferramentas para evitar essa armadilha, fornecendo uma estrutura que orienta as equipes a investigar o público-alvo, a projetar soluções baseadas em objetivos pedagógicos e a medir continuamente o sucesso da iniciativa [47]. Para os cursos autoinstrucionais do AVAMEC, essa disciplina é decisiva: é ela que transforma a informação em conhecimento.

### 2.2 Engenharia de Software: A Construção de Soluções Educacionais

De forma análoga, a Engenharia de Software entra como a disciplina que busca trazer método, previsibilidade e qualidade ao complexo processo de desenvolvimento de sistemas [48]. Sua premissa é que a construção de software não deve ser um esforço artesanal e imprevisível. Pelo contrário, ela deve seguir um conjunto de práticas e princípios que permitam gerenciar projetos de forma a cumprir requisitos, prazos e orçamentos, resultando em um produto final robusto e confiável.

Quando aplicada ao desenvolvimento de materiais educativos, sua contribuição ultrapassa a mera escrita de código. A ES oferece um arsenal de estratégias para gerir o trabalho. Metodologias ágeis como o Scrum, por exemplo, não são apenas ferramentas de gestão, mas mecanismos que estruturam a colaboração entre os diferentes especialistas (designers, conteudistas, desenvolvedores) em ciclos curtos que favorecem a adaptação e a melhoria contínua. Práticas como a componentização e os testes automatizados, defendidas pela área, geram um código mais limpo e de fácil manutenção, permitindo que os cursos digitais possam evoluir e ser adaptados com muito mais agilidade e segurança [49].

## **2.3 Transição para Novas Tecnologias**

### **2.3.1 Escolha Tecnológica e Design Instrucional**

A decisão de adotar tecnologias como React, TypeScript e Tailwind CSS no desenvolvimento de cursos na plataforma AVAMEC reflete uma estratégia abrangente que visa otimizar tanto os processos técnicos quanto pedagógicos. Do ponto de vista técnico, essas ferramentas oferecem melhorias significativas em modularidade, reutilização de componentes e manutenção de código [7]. Do ponto de vista pedagógico, facilitam implementar princípios de design instrucional essenciais para a criação de cursos mais eficazes e envolventes.

### **2.3.2 React: Uma Biblioteca JavaScript para Construção de Interfaces de Usuário**

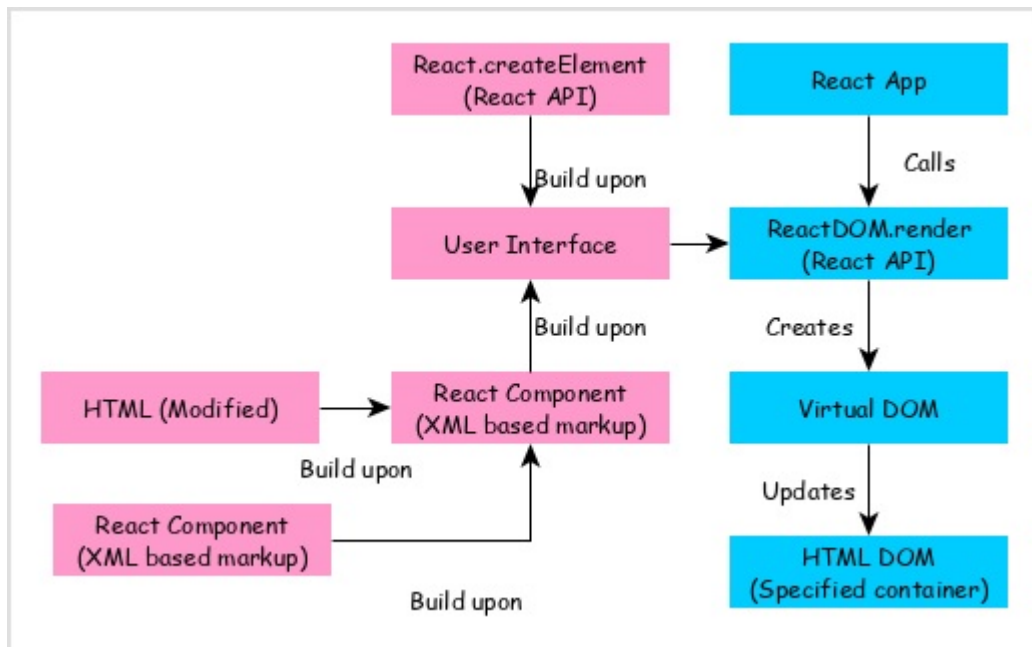
React é uma biblioteca JavaScript amplamente utilizada para a construção de interfaces de usuário (UI). Desenvolvida e mantida pelo Facebook, React adota uma abordagem baseada em componentes, permitindo a criação de partes modulares e reutilizáveis da interface, que podem ser desenvolvidas, testadas e atualizadas de forma independente [8]. Esta modularidade resulta em uma manutenção mais fácil e uma redução significativa de erros, pois cada componente pode ser isoladamente verificado e atualizado.

#### **2.3.2.1 Diagrama de Funcionamento do React**



A imagem fornecida ilustra o fluxo de trabalho do React, destacando a interação entre seus principais componentes e a DOM (Document Object Model) virtual e real. Abaixo, detalhamos cada parte do diagrama para uma melhor compreensão de seu funcionamento.

**Figura 7-** ReactJS Tutorial



Tutorialspoint, 2024.

A arquitetura apresentada na imagem descreve o fluxo de renderização e construção de interfaces no React, um framework JavaScript popular para construção de interfaces de usuário. A imagem detalha como os componentes React, escritos em uma sintaxe baseada em XML (conhecida como JSX), são processados e renderizados no navegador.

Inicialmente, os componentes React são definidos utilizando a sintaxe XML, facilitando a criação de estruturas de interface de maneira declarativa. Esses componentes JSX são então transformados em elementos React usando a API `React.createElement`. Esta API é responsável por traduzir o JSX em um formato que o React possa entender e manipular.

Os componentes React, após sua criação, são combinados para construir a interface do usuário. A construção da interface é iterativa e progressiva, permitindo que os componentes sejam aninhados e reutilizados, promovendo a modularidade e a manutenção do código.

Uma vez que a interface do usuário é montada, ela é passada para a API ReactDOM.render, responsável por renderizar essa interface no DOM do navegador. Nesse processo, o React cria uma representação virtual da interface, conhecida como Virtual DOM. Essa etapa é crucial, pois o Virtual DOM permite ao React gerenciar atualizações eficientemente, comparando a nova interface com a versão anterior e aplicando apenas as mudanças necessárias ao DOM real do navegador, resultando em uma maior desempenho.

Por fim, o Virtual DOM atualiza o DOM real, refletindo as modificações no navegador. Isso garante que o HTML modificado, criado a partir dos componentes React, seja exibido corretamente na tela do usuário.

Em resumo, essa arquitetura mostra como o React utiliza componentes baseados em XML (JSX) para construir interfaces de usuário de maneira declarativa, utilizando o Virtual DOM para garantir uma renderização eficiente no navegador, resultando em uma experiência de usuário mais suave e performática.

### **2.3.2.3 Conclusão**

React é uma biblioteca poderosa e flexível que oferece várias vantagens para o desenvolvimento de interfaces de usuário. Sua abordagem baseada em componentes, juntamente com a utilização da Virtual DOM, fluxos de dados unidirecionais e a sintaxe JSX, facilita a criação de aplicações web escaláveis, manuteníveis e altamente performáticas. A adoção de React na plataforma AVAMEC representa um passo significativo na modernização do desenvolvimento de cursos digitais, promovendo um ambiente de aprendizado mais envolvente e responsivo, alinhado com as melhores práticas de design instrucional e engenharia de software educacional.

### **2.3.3 ModfyJS: Um boilerplate em ReactJS**

**ModfyJS** é um boilerplate desenvolvido pelo V-Lab, o laboratório de informática e inovação da UFPE, visando modernizar e otimizar o desenvolvimento de aplicações educacionais web. A documentação está disponível por meio do link: [Gitbook do ModfyJS](https://vlab-1.gitbook.io/modfyjs-1.0.0-alpha) (<https://vlab-1.gitbook.io/modfyjs-1.0.0-alpha>).

Iniciado em 2022, o ModfyJS surgiu como uma resposta aos desafios enfrentados pela equipe ao trabalhar com uma arquitetura anterior baseada em HTML, CSS e JavaScript puro.

O projeto evoluiu significativamente, resultando em uma ferramenta robusta que integra tecnologias modernas como ReactJS, TypeScript e TailwindCSS.

#### **2.3.3.1 Início e Evolução**

Inicialmente, a equipe do V-Lab desenvolvia componentes e animações diretamente em HTML, CSS e JavaScript, o que resultava em um código complexo e difícil de manter como explicado anteriormente. A necessidade de melhorar a modularidade e a manutenibilidade levou à adoção do ReactJS, escolhido por sua capacidade de criar componentes reutilizáveis e gerenciar eficientemente a interface do usuário.

A transição para React não foi isenta de desafios. A equipe enfrentou problemas, como falhas na navegação e dificuldades na integração com a arquitetura tradicional de MPA (Multi Page Application). Para resolver esses problemas, foi implementado o *react-router-dom* em conjunto com o *Create React App (CRA)*, permitindo a simulação de um ambiente MPA numa aplicação SPA (Single Page Application) baseada em React.

#### **2.3.3.2 Desafios e Soluções Técnicas**

**Adaptação ao TypeScript:** Um dos principais desafios foi a migração do código JavaScript para TypeScript, o que trouxe maior segurança ao desenvolvimento, permitindo a detecção precoce de erros através da tipagem estática. Isso também melhorou a organização do código, facilitando a manutenção e a escalabilidade.

**Navegação e Gerenciamento de Estados:** A equipe utilizou o *react-router-dom* para criar uma navegação fluida entre as "páginas" simuladas, enquanto a API de hooks do React (como *useState*, *useEffect* e *useSearchParams*) facilitou a gestão de estados e parâmetros de URL. Essa solução garantiu que a experiência do usuário fosse contínua e intuitiva, essencial para a navegação dentro dos minicursos e módulos.

**Integração com TailwindCSS e Automação de Testes:** O TailwindCSS foi introduzido para padronizar a estilização dos componentes e acelerar o processo de desenvolvimento, reduzindo a necessidade de CSS personalizado. Além disso, a equipe integrou ferramentas de teste automatizado como Jest. Essas ferramentas foram cruciais para garantir que novas funcionalidades não quebrassem componentes existentes, permitindo a

simulação de interações do usuário e a validação do comportamento da aplicação em um ambiente controlado.

### **2.3.3.3 Refatoração e Manutenção Contínua**

A necessidade de uma arquitetura mais escalável e manutenível levou à decisão de realizar um refactor completo do código. Este processo envolveu a correção de bugs, a resolução de conflitos no Git e a adoção de práticas mais rigorosas de versionamento e gestão de dependências. O refactor resultou na versão alpha do ModfyJS, que agora se posiciona como um framework robusto e versátil para o desenvolvimento de aplicações web educacionais.

**Otimização de Performance:** A reestruturação do código durante o refactor também melhorou a eficiência na renderização dos componentes, reduzindo o tempo de carregamento das páginas e proporcionando uma experiência de usuário mais fluida.

### **2.3.3.4 Aprofundamento técnico**

#### **Arquitetura Modular com React**

Uma das características mais marcantes do ModfyJS é sua arquitetura modular, que se baseia fortemente nos conceitos de componentização do React. Essa abordagem facilita a manutenção, o desenvolvimento iterativo e a reutilização de componentes, o que é essencial para um ambiente de desenvolvimento de cursos dinâmico como o AVAMEC.

No código abaixo, uma das espinha dorsal do ModfyJS, podemos observar o uso do React para renderizar dinamicamente as páginas de um módulo de curso:

**Figura 8** – Código base de criação de cursos para o AVAMEC

```

import React from "react";
import { useSearchParams } from "react-router-dom";

import { PAGES_FROM_MODULE_1 } from "../module-1";
import { PAGES_FROM_MODULE_2 } from "../module-2";
import { PageRenderError } from "../components/PageRenderError";

export const App = () => {
  const [searchParams] = useSearchParams();
  const module = searchParams.get("module");
  const page = searchParams.get("page");

  const index: number = Number(module! + page!);

  const pages = {
    ...PAGES_FROM_MODULE_1,
    ...PAGES_FROM_MODULE_2,
  };

  if (pages[index as keyof typeof pages]) {
    return <div className="App">{pages[index as keyof typeof pages]}</div>;
  }

  return <PageRenderError />;
};

```

Elaboração própria, 2024.

### Explicação do Código

**Uso do React Router:** O código utiliza o `useSearchParams` da biblioteca `react-router-dom` para extrair os parâmetros de URL (`module` e `page`), que determinam a página específica a ser exibida no curso. Essa abordagem permite criar uma navegação eficiente entre diferentes partes do curso, mantendo o estado e o contexto corretos.

**Renderização Dinâmica de Páginas:** As páginas dos módulos (`PAGES_FROM_MODULE_1` e `PAGES_FROM_MODULE_2`) são importadas como objetos. O componente `App` utiliza esses objetos para renderizar dinamicamente a página correspondente aos parâmetros da URL. Se a página solicitada não for encontrada, o componente `PageRenderError` é exibido, garantindo uma experiência de usuário controlada em caso de erros.

## Componentização e Reutilização

O uso de componentes reutilizáveis é uma prática essencial no ModfyJS. Por exemplo, as páginas de um módulo são definidas como componentes e exportadas de maneira organizada:

**Figura 9** – Código base de criação de cursos para o AVAMEC

```
import { Page_1 } from "../page-1";
import { Page_2 } from "../page-2";

export const PAGES_FROM_MODULE_2 = {
  21: <Page_1 />,
  22: <Page_2 />,
};
```

Elaboração própria, 2024.

Este código demonstra como as páginas são definidas e exportadas como um objeto, onde cada chave corresponde a um identificador único da página. Isso permite que o componente principal (App) faça referência a elas de forma eficiente, simplificando a lógica de navegação e o gerenciamento de estado.

## Navegação e Controle de Estado

O ModfyJS também utiliza hooks personalizados para facilitar a navegação e o controle de estado. Veja como o hook useModfy é implementado:

**Figura 10** – Captura de tela de código base de criação de cursos para o AVAMEC

```
import { useSearchParams, useNavigate } from "react-router-dom";

export const useModify = () => {
  const navigate = useNavigate();
  const [searchParams] = useSearchParams();
  const module = searchParams.get("module");
  const page = searchParams.get("page");

  function nextPage() {
    const nextIndex = Number(page!) + 1;
    window.scrollTo(0, 0);
    navigate(`?module=${module!}&page=${nextIndex.toString()}`);
  }

  function previousPage() {
    const previousIndex = Number(page!) - 1;
    window.scrollTo(0, 0);
    navigate(`?module=${module!}&page=${previousIndex.toString()}`);
  }

  function navigateToPage(params: { module: number; page: number }) {
    window.scrollTo(0, 0);
    navigate(`?module=${params.module.toString()}&page=${params.page.toString()}`);
  }

  return { nextPage, previousPage, navigateToPage };
};
```

Elaboração própria, 2024.

#### Explicação do Hook useModify

**Controle de Navegação:** Este hook fornece três funções (nextPage, previousPage e navigateToPage) para controlar a navegação entre as páginas do curso. Utilizando o useNavigate da biblioteca react-router-dom, ele permite que o curso mude dinamicamente de página sem recarregar a aplicação.

**Manutenção do Contexto de Navegação:** As funções de navegação também garantem que a posição de rolagem da página seja resetada para o topo (window.scrollTo(0, 0)), proporcionando uma experiência de usuário mais suave e coerente.

## Contexto para Controle de Fluxo de Perguntas

Além da navegação, o ModfyJS utiliza contextos para gerenciar estados complexos, como o controle de perguntas em um quiz. Veja a implementação do contexto de perguntas:

**Figura 11** – Código base de criação de cursos para o AVAMEC

```
import React, { createContext, useContext, useState } from "react";

const QuestionsContext = createContext();

export const QuestionsProvider = ({ children }) => {
  const [questionArray, setQuestionArray] = useState([]);
  const [questionAnsweredArray, setQuestionAnsweredArray] = useState([]);
  const [showResult, setShowResult] = useState(false);

  const updateArray = (newArray) => setQuestionArray((prev) => typeof newArray ===
    "function" ? newArray(prev) : newArray);
  const updateQuestionAnsweredArray = (newArray) => setQuestionAnsweredArray((prev) =>
    typeof newArray === "function" ? newArray(prev) : newArray);
  const updateShowResult = () => setShowResult((old) => !old);

  return (
    <QuestionsContext.Provider value={{ questionArray, updateArray, showResult,
      updateShowResult, questionAnsweredArray, updateQuestionAnsweredArray }}>
      {children}
    </QuestionsContext.Provider>
  );
};

export const useQuestionsContext = () => {
  const context = useContext(QuestionsContext);
  if (!context) throw new Error("useQuestionsContext must be used within a
    QuestionsProvider");
  return context;
};
```

Elaboração própria, 2024.

## Explicação do Contexto

**Gerenciamento de Estado Centralizado:** O QuestionsContext centraliza o estado relacionado às perguntas do curso, como quais perguntas foram respondidas corretamente e quando exibir o resultado.



**Reutilização e Isolamento de Lógica:** Ao utilizar um contexto, a lógica do quiz fica isolada, facilitando sua reutilização em diferentes partes do curso. Isso também simplifica os testes e a manutenção, pois toda a lógica de controle de estado está em um só lugar.

### Pagination e Interatividade

A interatividade no ModfyJS é ainda reforçada por componentes de paginação personalizados que integram diretamente com a API do AVAMEC para registro de progresso e outras métricas de usuário. Logo abaixo está uma parte do componente pagination que por ser um código muito extenso foi significativamente reduzido para caber na imagem:

**Figura 12** – Captura de tela de código base de criação de cursos para o AVAMEC

```
import { useEffect } from "react";
import { useModfy } from "../../@modfy";
import { useQuestionsContext } from "../../contexts/QuestionsContext";

let errorsCounter = 0;
let updateFlag = 0;

export const Pagination: React.FC<PaginationProps> = ({
  numberOfSlides,
  numberOfMinicourses,
  slideNow,
  minicourseNow,
  numberOfErrors = 3,
  activityPage = false,
}) => {
  const { nextPage, previousPage, navigateToPage } = useModfy();
  const { questionArray, showResult, updateShowResult } = useQuestionsContext();

  useEffect(() => {
    // Registro de progresso na API
    let API = new window.BridgeRestApi();
    let thisPageProgress = (100 / numberOfSlides) * slideNow;
    API.registrarPorcentagemConclusaoUnidade(`minicurso_${minicourseNow}`, `${thisPageProgress}`);
  });

  // Implementação adicional...
};
```

Elaboração própria, 2024.

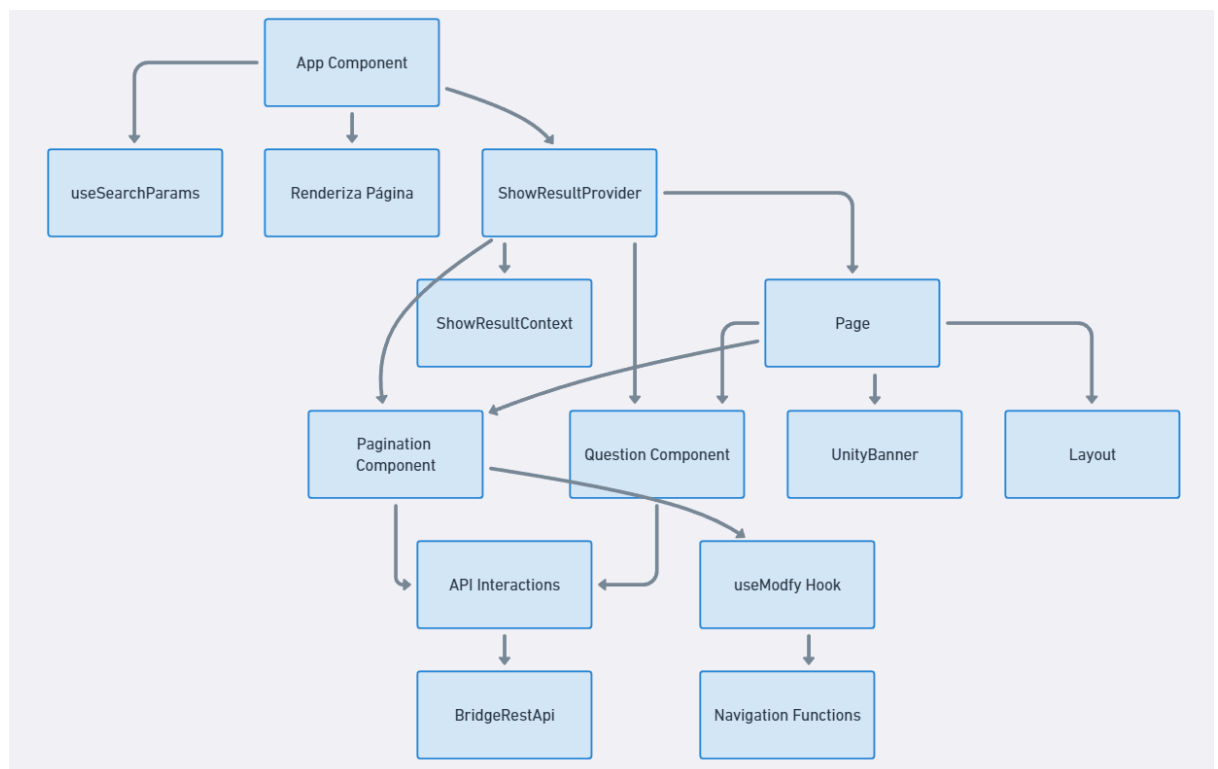
### Explicação da Paginação

**Registro de Progresso na API:** O componente utiliza a API interna (window.BridgeRestApi) para registrar o progresso dos usuários, o que é crucial para manter um ambiente de aprendizado engajador.

**Controle de Acesso com Limitação de Tentativas:** A lógica de controle de acesso é configurada para limitar as tentativas dos usuários, utilizando contadores de erros e estados locais para gerenciar a experiência de aprendizado de forma mais controlada.

Abaixo está a arquitetura do ModfyJS criada pelo time de desenvolvimento do V-Lab UFPE, onde mostra como cada componente e microserviços interagem entre si, onde UnityBanner e Layout são exemplos de componentes, mas não limitado a estes.

**Figura 13** – Arquitetura do boilerplate ModfyJS para ReactJS



Elaboração própria, 2024

## Conclusão

O ModfyJS representa um avanço significativo no desenvolvimento de cursos digitais no V-Lab, proporcionando uma base sólida e moderna que permite maior agilidade, eficiência e qualidade na criação de materiais educacionais. Com a integração de tecnologias modernas

e práticas avançadas de engenharia de software, o ModfyJS posiciona o V-Lab na vanguarda do desenvolvimento de aplicações educacionais, com potencial para servir de modelo para outras instituições.

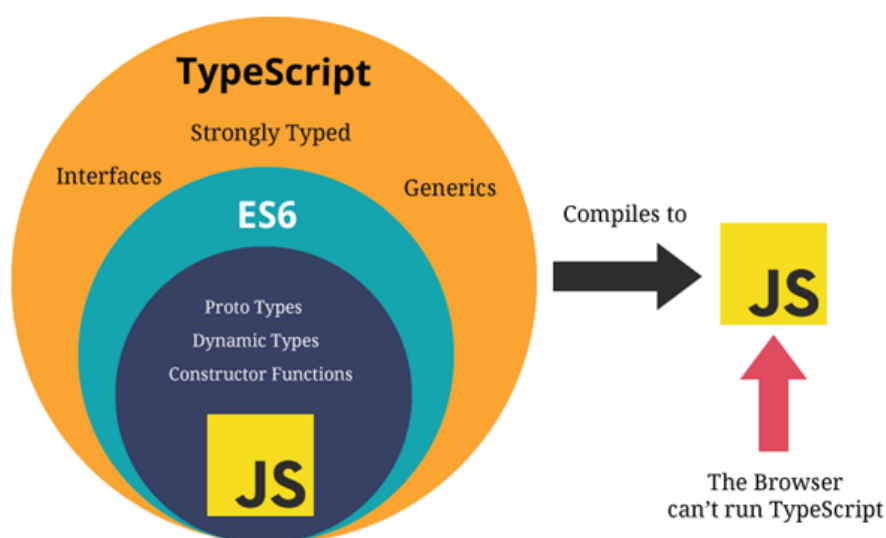
### 2.3.4 TypeScript: Um Superconjunto Tipado de JavaScript

TypeScript é um superconjunto tipado (as variáveis ao serem declaradas recebem um “tipo”, como, por exemplo, string, que deve ser respeitado) de JavaScript que adiciona tipos estáticos à linguagem, melhorando a legibilidade e a manutenibilidade do código [9]. Desenvolvido e mantido pela Microsoft, TypeScript compila para JavaScript puro, garantindo compatibilidade com todos os navegadores e plataformas. Em um ambiente colaborativo como o do AVAMEC, onde vários desenvolvedores trabalham simultaneamente, a tipagem estática ajuda a evitar erros comuns e facilita o entendimento do código por novos membros da equipe.

#### 2.3.4.1 Diagrama de Funcionamento do TypeScript

A imagem fornecida ilustra o relacionamento entre JavaScript, ES6 (ECMAScript 6), e TypeScript, destacando as principais características e benefícios de cada um. Abaixo, detalhamos cada parte do diagrama para uma melhor compreensão de seu funcionamento.

**Figura 14-** What is Typescript?



Medium, 2018.

### 2.3.4.3 Conclusão

TypeScript é uma ferramenta poderosa que traz várias melhorias ao desenvolvimento em JavaScript, especialmente em projetos grandes e colaborativos como os da plataforma AVAMEC. A tipagem estática, as interfaces, e os recursos avançados de ES6 e além, tornam o código mais legível, robusto e fácil de manter. A adoção de TypeScript na AVAMEC não só melhora a eficiência do desenvolvimento, mas também alinha o código com os princípios do design instrucional, garantindo clareza e precisão na transmissão de informações. Com TypeScript, a equipe de desenvolvimento pode criar cursos digitais mais seguros, confiáveis e escaláveis, atendendo às necessidades de uma plataforma de aprendizado moderna e eficaz.

### 2.3.5 Tailwind CSS: Framework de Utilitários CSS

Tailwind CSS é um framework CSS utilitário que facilita a criação de interfaces de usuário consistentes, responsivas e esteticamente agradáveis com menos esforço [10]. Ao contrário de outros frameworks CSS que fornecem componentes predefinidos, Tailwind adota uma abordagem diferente, oferecendo classes utilitárias de baixo nível que podem ser combinadas para criar qualquer design diretamente no HTML. Isso elimina a necessidade de escrever CSS personalizado para cada componente, acelerando significativamente o processo de desenvolvimento e garantindo designs uniformes e facilmente ajustáveis a diferentes dispositivos e resoluções de tela.

#### 2.3.5.1 Vantagens Técnicas do Tailwind CSS

**Classes Utilitárias:** Tailwind oferece um conjunto extensivo de classes utilitárias que permitem aplicar estilos diretamente nos elementos HTML. Isso inclui estilos para margens, espaçamentos, cores, tipografia, layout, entre outros.

**Consistência Visual:** As classes utilitárias garantem que os designs sejam consistentes em toda a aplicação, facilitando a manutenção e a compreensão do código. A consistência visual é fundamental para um design instrucional eficiente, por facilitar a navegação e a absorção de informações pelos alunos.

**Responsividade:** Tailwind facilita a criação de designs responsivos com classes utilitárias específicas para diferentes pontos de interrupção (breakpoints). Isso permite ajustar

os estilos conforme o tamanho da tela, garantindo uma experiência de usuário ótima em dispositivos móveis, tablets e desktops.

**Personalização:** Embora Tailwind ofereça uma ampla gama de classes utilitárias por padrão, ele também permite uma personalização extensa. Através do arquivo de configuração `tailwind.config.js`, os desenvolvedores podem definir temas personalizados, adicionando ou alterando cores, espaçamentos, fontes e outros estilos.

**Desempenho:** Tailwind inclui uma ferramenta de purgação que remove todas as classes CSS não utilizadas do arquivo de produção, reduzindo significativamente o tamanho do CSS final e melhorando o desempenho da aplicação.

**Integração com Ferramentas Modernas:** Tailwind se integra bem com frameworks modernos como React, Vue e Angular. Isso facilita a adoção de Tailwind em projetos novos e existentes.

### **2.3.5.2 Exemplos de Código: Comparação entre CSS Tradicional e Tailwind CSS**

#### **Código com HTML e CSS puro**

**Figura 15** – Captura de tela de código HTML exemplo para comparação com TailwindCSS

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Tradicional</title>
  <style>
    .grid {
      display: grid;
      gap: 1.25rem;
    }
    .card {
      border: 1px solid #e5e7eb;
      border-radius: 0.5rem;
      padding: 1.25rem;
      margin-bottom: 1.25rem;
    }
    .card img {
      width: 100%;
      border-radius: 0.5rem;
      margin-bottom: 0.75rem;
    }
    .card-content {
      font-size: 1rem;
      color: #4b5563;
    }
    @media (min-width: 768px) {
      .grid {
        display: flex;
      }
      .card {
        flex: 1;
        margin-bottom: 0;
      }
    }
  </style>
</head>
<body>
  <div class="grid">
    <div class="card">
      
      <div class="card-content">Content 1</div>
    </div>
    <div class="card">
      
      <div class="card-content">Content 2</div>
    </div>
  </div>
</body>
</html>

```

Elaboração própria, 2024.

## Código com Tailwind CSS

**Figura 16** – Captura de tela de código Tailwind exemplo para comparação com HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tailwind CSS</title>
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
</head>
<body>
  <div class="grid gap-5 md:flex">
    <div class="border border-gray-200 rounded-lg p-5 mb-5 md:mb-0 flex-1">
      
      <div class="text-base text-gray-600">Content 1</div>
    </div>
    <div class="border border-gray-200 rounded-lg p-5 flex-1">
      
      <div class="text-base text-gray-600">Content 2</div>
    </div>
  </div>
</body>
</html>
```

Elaboração própria, 2024.

### 2.3.5.3 Análise da Redução Significativa na Quantidade de Linhas do Código

Comparando os dois exemplos, podemos observar uma redução significativa na quantidade de linhas de código e uma simplificação na aplicação de estilos:

**Código Tradicional:** O código HTML tradicional requer uma quantidade considerável de CSS personalizado para estilizar e tornar a interface responsiva. Isso inclui a definição de classes, propriedades de estilo e media queries.

**Código com Tailwind CSS:** Usando Tailwind CSS, o mesmo layout é alcançado com um HTML muito mais conciso. As classes utilitárias aplicadas diretamente nos elementos HTML eliminam a necessidade de escrever CSS personalizado para cada estilo, resultando em um código HTML mais legível.

### 2.3.5.4 Conclusão

Tailwind CSS oferece uma abordagem poderosa e eficiente para a criação de interfaces de usuário responsivas e consistentes. A utilização de classes utilitárias reduz significativamente o tempo de desenvolvimento e a quantidade de código necessário, enquanto mantém uma consistência visual crucial para um design instrucional eficiente. Com Tailwind, os desenvolvedores podem criar rapidamente interfaces modernas e adaptáveis a diferentes dispositivos, facilitando a navegação e a absorção de informações pelos alunos na plataforma AVAMEC.

A adoção dessas tecnologias não só melhora a eficiência técnica, mas também apoia a implementação de princípios de design instrucional. Com React, por exemplo, é possível criar componentes interativos e reutilizáveis, como quizzes, animações avançadas e atividades práticas, que podem ser facilmente integrados em diferentes cursos, promovendo um aprendizado mais dinâmico e envolvente. TypeScript, por sua vez, facilita a criação de lógicas de feedback imediato e personalizado, essencial para a motivação e o engajamento dos alunos. Tailwind CSS contribui para a consistência visual e a acessibilidade dos cursos, dois pilares fundamentais do design instrucional eficaz.

### **2.3.6 Estrutura da nova arquitetura**

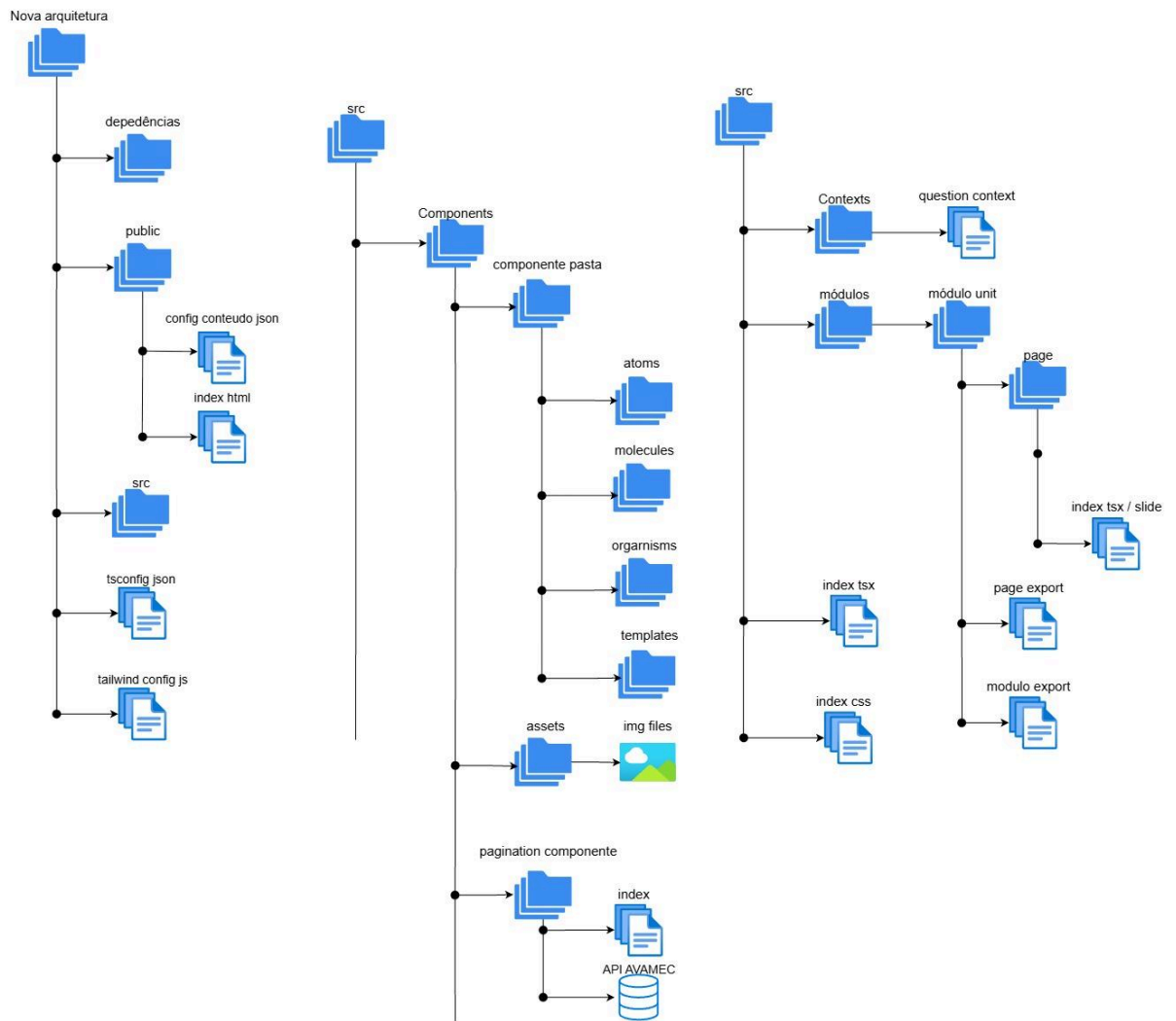
Com a transição para tecnologias mais modernas, a estrutura dos novos cursos digitais para a plataforma AVAMEC foi significativamente otimizada. A nova arquitetura, baseada em React, TypeScript e Tailwind CSS, oferece uma abordagem mais modular e escalável, facilitando a manutenção e a atualização dos cursos. Na figura abaixo ilustra essa nova arquitetura. A seguir, descrevemos detalhadamente essa estrutura, destacando as principais pastas e arquivos.

#### **2.3.6.1 Descrição da Nova Arquitetura**

A imagem é um diagrama que representa a estrutura de diretórios e arquivos de curso desenvolvido com essa stack tecnológica mais moderna. Esta estrutura é organizada em vários níveis, refletindo a modularidade e a reutilização de componentes que essas novas tecnologias permitem.

**Figura 17** – Nova arquitetura de software para criação de cursos AVAMEC otimizada.





Elaboração própria, 2024.

### 2.3.6.2 Estrutura

No desenvolvimento de projetos utilizando React e TypeScript, a estrutura de diretórios desempenha um papel fundamental na organização e no fluxo de trabalho. Abaixo, apresento uma descrição detalhada dessa estrutura e das funções de cada componente no projeto.

O diretório de Dependências é onde todas as bibliotecas e frameworks necessários para o funcionamento do projeto são armazenados. Essas dependências são gerenciadas por ferramentas como o npm ou yarn, que facilitam a instalação e atualização dessas bibliotecas.

No diretório Public, encontra-se o arquivo de configuração Config JSON, que define as propriedades e parâmetros dos conteúdos do projeto. Este arquivo desempenha um papel semelhante ao das configurações em arquiteturas mais antigas, mas agora é integrado de forma mais eficiente com as tecnologias modernas utilizadas na nova stack do projeto. O Index HTML também está localizado neste diretório e serve como o ponto de entrada principal para o aplicativo React, sendo essencial para a renderização inicial da aplicação no navegador.

No diretório Src, encontra-se o arquivo de configuração TSConfig JSON, que define as opções de compilação do TypeScript, assegurando a tipagem estática e a consistência do código ao longo do projeto. Outro arquivo importante é o Tailwind Config JS, que permite customizar as classes utilitárias do Tailwind CSS, facilitando a aplicação de estilos no projeto de maneira modular e eficiente.

Os componentes do projeto são organizados no diretório Components. Aqui, cada componente React é armazenado em subdiretórios específicos, denominados Componente Pasta, facilitando a organização e a modularização do código. Dentro de cada subdiretório, encontram-se os Componente Files, sendo os arquivos .tsx onde os componentes React são definidos. Além disso, imagens específicas para cada componente são armazenadas em arquivos Img Files locais, permitindo uma gestão mais eficiente desses recursos. Um exemplo de organização específica dentro deste diretório é o Pagination Componente, o qual é um diretório dedicado aos componentes de paginação, essenciais para a navegação eficiente nos cursos. O arquivo principal deste diretório, Index, define a lógica de paginação e está integrado com a API AVAMEC, possibilitando à plataforma reconhecer e gerir slides, módulos, atividades avaliativas e o progresso do curso.

O diretório de Contextos e Módulos engloba tanto os arquivos de contexto React, armazenados em Contexts, quanto os módulos do curso, organizados em Módulos. Dentro de Contexts, por exemplo, o Question Context é um contexto específico para a gestão centralizada das perguntas e respostas do curso, facilitando o gerenciamento de estados globais da aplicação. Já no diretório Módulos, cada módulo do curso é organizado em subdiretórios que contêm unidades, chamadas de Módulo Unit, e páginas específicas, denominadas Page. Os recursos estáticos, como imagens, são armazenados no diretório Assets, facilitando sua utilização nas diferentes páginas do curso.

Finalmente, os arquivos Index TSX / Slide são responsáveis por definir o conteúdo das páginas e slides, utilizando a sintaxe JSX/TSX para componentes React. A exportação de páginas e módulos completos é facilitada por arquivos como o Page Export e Módulos Export, que permitem a reutilização dessas partes em diferentes contextos do curso.

Essa estrutura de diretórios e arquivos é crucial para manter a organização, modularização e eficiência do projeto, facilitando tanto o desenvolvimento quanto a manutenção da aplicação.

### 2.3.6.3 Análise Crítica

A nova arquitetura baseada em React, TypeScript e Tailwind CSS oferece diversas vantagens em comparação com a arquitetura anterior:

**Modularidade:** A estrutura de componentes permite uma alta modularização, facilitando a reutilização e manutenção do código. Cada componente pode ser desenvolvido, testado e atualizado de forma independente.

**Manutenção Simplificada:** Com TypeScript, a tipagem estática melhora a legibilidade do código e reduz a incidência de erros, tornando a manutenção mais eficiente.

**Escalabilidade:** A nova arquitetura é altamente escalável, permitindo a adição de novos módulos e funcionalidades sem comprometer o desempenho ou a organização do código.

**Interatividade Avançada:** React facilita a criação de interfaces de usuário dinâmicas e interativas, essenciais para um design instrucional eficaz. A capacidade de gerenciar estados com Contexts e Hooks melhora significativamente a experiência do usuário.

**Consistência Visual:** Tailwind CSS assegura uma estilização consistente e responsiva, crucial para a acessibilidade e a experiência do usuário.

### 2.3.6.4 Conclusão

A nova arquitetura adotada para o desenvolvimento dos cursos digitais na plataforma AVAMEC representa um avanço significativo em termos de eficiência, modularidade e

qualidade. A integração de React, TypeScript e Tailwind CSS permite uma gestão mais eficaz dos conteúdos, facilita a manutenção e atualização dos cursos, e proporciona uma experiência de aprendizagem mais rica e envolvente para os alunos. Esta transição não apenas melhora os processos técnicos, mas também alinha as práticas de desenvolvimento com os princípios do design instrucional, resultando em cursos de alta qualidade que atendem às crescentes demandas por educação à distância no Brasil.

### **2.3.7 Planejamento e Treinamento**

Para que a transição para essas novas tecnologias seja bem-sucedida, é essencial um planejamento detalhado e a realização de treinamentos adequados [10] para os desenvolvedores e designers instrucionais envolvidos. O planejamento deve incluir a criação de uma roadmap que detalhe as etapas da migração, incluindo prazos, recursos necessários e metas a serem atingidas em cada fase.

Os treinamentos devem abranger tanto os aspectos técnicos das novas tecnologias quanto suas aplicações práticas no contexto do design instrucional. Para React, os desenvolvedores precisam se familiarizar com conceitos como estados e props, hooks e gerenciamento de estado com Redux. No caso de TypeScript, é necessário entender a tipagem estática, interfaces, classes e generics. Já para Tailwind CSS, os treinamentos devem focar na utilização das classes utilitárias e na criação de temas customizados para garantir a consistência visual dos cursos.

Além dos treinamentos técnicos, é crucial promover workshops e sessões colaborativas (reuniões) onde os desenvolvedores e designers instrucionais possam trabalhar juntos em projetos piloto. Esses projetos servirão como base para a aplicação prática dos novos conhecimentos e ajudarão a identificar possíveis desafios e áreas de melhoria antes da implementação em larga escala.

**Design Instrucional e Treinamento:** O design instrucional desempenha um papel crucial no treinamento da equipe, garantindo que os desenvolvedores compreendam não apenas os aspectos técnicos, mas também como aplicar esses conhecimentos para melhorar a experiência de aprendizagem. O treinamento deve incluir práticas de design centradas no aluno, como a criação de objetivos de aprendizagem claros, a segmentação de conteúdo em módulos gerenciáveis e a integração de avaliações contínuas.

Os princípios de design instrucional, como a análise de necessidades dos alunos, o planejamento de objetivos de aprendizagem e a avaliação contínua, devem ser incorporados nos treinamentos. Isso garante que os desenvolvedores estejam preparados para criar cursos que não apenas sejam tecnicamente robustos, mas também pedagogicamente eficazes. Por exemplo, a criação de quizzes interativos e atividades práticas utilizando React e TypeScript pode ser praticada em projetos piloto, onde os desenvolvedores podem experimentar e receber feedback sobre a eficácia pedagógica dessas ferramentas.

### **2.3.8 Desafios Integrados**

A integração de novas tecnologias com práticas de design instrucional apresenta diversos desafios que precisam ser gerenciados cuidadosamente para garantir uma transição suave e bem-sucedida. Entre os principais desafios estão:

**Curva de Aprendizado:** A introdução de novas tecnologias implica em uma curva de aprendizado para a equipe de desenvolvimento. Mesmo com treinamentos intensivos, é natural que a produtividade inicial possa ser impactada até que todos se familiarizem completamente com as novas ferramentas e métodos. Para mitigar esse desafio, é importante que os treinamentos sejam contínuos e que haja suporte disponível para a equipe durante todo o processo de adaptação.

**Compatibilidade e Integração:** Garantir que as novas tecnologias sejam compatíveis com as ferramentas e infraestruturas existentes é um desafio crucial. Isso inclui não apenas a integração técnica, mas também a adequação dos novos processos de desenvolvimento às práticas pedagógicas estabelecidas. A transição deve ser cuidadosamente planejada para evitar interrupções nos cursos existentes e garantir que a qualidade pedagógica seja mantida.

**Resistência à Mudança:** Mudanças significativas nos processos de trabalho podem encontrar resistência por parte de membros da equipe acostumados aos métodos tradicionais. É importante promover uma cultura de aceitação da mudança, destacando os benefícios e proporcionando suporte contínuo durante a transição. A comunicação clara e o envolvimento da equipe no processo de decisão podem ajudar a reduzir a resistência e facilitar a adoção das novas tecnologias [11].

**Manutenção e Atualização de Cursos Existentes:** A migração de cursos desenvolvidos com a arquitetura anterior (HTML, CSS, JavaScript) para a nova stack

tecnológica teve que ser planejada cuidadosamente para evitar interrupções na disponibilidade dos cursos e garantir que a qualidade pedagógica seja mantida. Isso pode envolver a recriação de conteúdos e a adaptação de materiais para se alinharem às novas tecnologias e aos princípios de design instrucional.

**Feedback e Ajustes Contínuos:** A integração de novas tecnologias requer um processo contínuo de feedback e ajustes. Isso significa que a equipe deve estar preparada para identificar e corrigir problemas rapidamente, bem como adaptar as práticas de desenvolvimento conforme necessário para otimizar os resultados. A utilização de metodologias ágeis, como Scrum, pode facilitar essa transição [12], permitindo ajustes rápidos e incrementais conforme a equipe avança no processo de migração.

### **2.3.9 Design Instrucional na Transição**

O design instrucional (DI) é um processo sistemático que envolve a análise, design, desenvolvimento, implementação e avaliação de materiais didáticos e experiências de aprendizagem para garantir que sejam eficazes e eficientes [13]. No contexto da transição tecnológica, o DI desempenha um papel essencial para garantir que as novas ferramentas sejam utilizadas de maneira a maximizar a qualidade pedagógica dos cursos.

**Análise de Necessidades:** Antes de iniciar a transição, é crucial realizar uma análise detalhada das necessidades dos alunos e das limitações dos cursos atuais. Isso inclui identificar quais aspectos dos cursos existentes não atendem às expectativas dos alunos e como as novas tecnologias podem resolver esses problemas. A análise de necessidades também deve considerar o feedback dos educadores para garantir que as mudanças atendam às suas expectativas e melhorem a experiência de aprendizagem.

**Design de Objetivos de Aprendizagem:** Com base na análise de necessidades, devem ser definidos objetivos de aprendizagem claros e mensuráveis para cada curso. Esses objetivos orientam o desenvolvimento de conteúdos e atividades, garantindo que cada componente do curso contribua para o aprendizado dos alunos. A utilização de React permite a criação de componentes interativos [8] que podem ser alinhados diretamente com esses objetivos, proporcionando uma experiência de aprendizagem mais dinâmica e eficaz.

**Desenvolvimento de Conteúdos e Atividades:** O desenvolvimento de conteúdos e atividades deve ser realizado de maneira modular, utilizando as capacidades de

componentização do React. Isso facilita a criação e a manutenção de materiais didáticos, permitindo que componentes individuais, como quizzes e simulações, sejam reutilizados em diferentes cursos. A utilização de TypeScript garante que esses componentes sejam desenvolvidos de maneira robusta e confiável, enquanto o Tailwind CSS assegura a consistência visual e a acessibilidade dos materiais.

**Implementação e Testes:** A implementação dos cursos deve ser acompanhada por testes rigorosos para garantir que todos os componentes funcionem corretamente e atendam aos objetivos de aprendizagem. Isso inclui testes de usabilidade para assegurar que os alunos possam navegar e interagir com os cursos de maneira intuitiva e eficiente.

Em conclusão, a transição para novas tecnologias como React, TypeScript e Tailwind CSS, combinada com práticas avançadas de design instrucional, promete trazer melhorias significativas tanto na eficiência técnica quanto na qualidade pedagógica dos cursos oferecidos pela plataforma AVAMEC. No entanto, essa transição deve ser cuidadosamente planejada e executada, com um foco constante no treinamento da equipe e na gestão dos desafios integrados, para garantir que os benefícios esperados sejam plenamente realizados.

### **Benefícios da nova arquitetura com o alinhamento do design instrucional**

As tecnologias React, TypeScript e Tailwind CSS possibilitam o desenvolvimento de features altamente componentizáveis e complexas na plataforma AVAMEC, mantendo uma facilidade de manutenção essencial para o design instrucional eficaz. Com React, os desenvolvedores podem criar componentes reutilizáveis e modulares que facilitam a construção de interfaces de usuário ricas e interativas, alinhadas aos princípios pedagógicos. TypeScript, ao fornecer tipagem estática, melhora a robustez e a legibilidade do código, permitindo que componentes complexos sejam gerenciados e depurados com mais eficiência, crucial para garantir que os objetivos de aprendizagem sejam consistentemente atendidos. Tailwind CSS, com sua abordagem utilitária, simplifica a aplicação de estilos consistentes, assegurando que o design permaneça uniforme e facilmente ajustável, o que é vital para manter a experiência de aprendizagem visualmente atraente e acessível. Juntas, essas tecnologias não só permitem criar sistemas complexos e dinâmicos no AVAMEC, como também garantem que a manutenção e a atualização desses sistemas sejam simplificadas, reduzindo significativamente o tempo e o esforço necessários para implementar mudanças e corrigir erros, sem comprometer a qualidade pedagógica dos cursos oferecidos.

Com isso, os resultados da nova arquitetura são demonstrados nas figuras abaixo onde foi possível criar componentes complexos e interativos que aumentam o engajamento dos alunos com o conteúdo desenvolvido com tal arquitetura e ferramentas modernas.

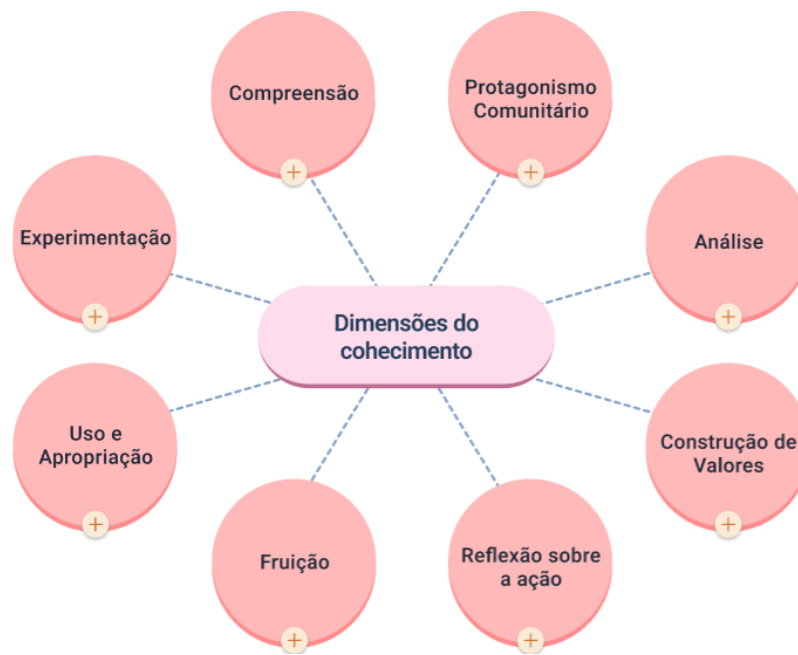
**Figura 18-** Componente feito com arquitetura avançada



V-LAB UFPE, 2024

**Figura 19-** Componente feito com arquitetura avançada





V-LAB UPE, 2024

**Figura 20-** Componente feito com arquitetura avançada



V-LAB UFPE, 2024

### 2.3.10 Atomic Design: integrando padronização e reuso ao ModfyJS

A incorporação da metodologia Atomic Design — proposta por Brad Frost [23] e hoje consolidada em projetos de design system para ambientes educacionais — representa o passo seguinte na modernização da plataforma AVAMEC. Nesse paradigma, a interface é tratada

como um ecossistema hierárquico que parte de átomos, evolui para moléculas e organismos, consolida-se em templates e se materializa em páginas, favorecendo tanto a composição quanto a evolução incremental de cada componente. Quando essa abordagem se combina ao boilerplate ModfyJS, os ganhos introduzidos por React, TypeScript e Tailwind CSS são potencializados, sem perder a aderência às premissas do design instrucional, sobretudo a definição clara de objetivos, a modularização do conteúdo e a escalabilidade dos materiais educacionais [24].

Em termos conceituais, os átomos correspondem a elementos HTML básicos estilizados por classes utilitárias do Tailwind; as moléculas agregam pequenos agrupamentos desses átomos, como um campo de busca completo; os organismos unem diversas moléculas numa seção coesa — por exemplo, um cabeçalho de módulo contendo navegação, avatar do discente e indicador de progresso; os templates distribuem organismos em um grid responsivo que dita a espinha dorsal da interface; e, por fim, as páginas instanciam esses templates com o conteúdo pedagógico efetivo, integrado por vídeos, leituras guiadas e quizzes. Essa gradação explicita as dependências internas da interface e permite que qualquer alteração aplicada em nível atômico se propague por toda a hierarquia sem ruptura de estilo nem de funcionalidade.

Do ponto de vista técnico, o Atomic Design oferece reuso sistemático, pois cada átomo ou molécula passa a integrar um catálogo versionado no ModfyJS importável por diferentes cursos sem duplicação de código; garante consistência visual, já que todos os níveis compartilham tokens de cor, tipografia e espaçamento definidos no arquivo `tailwind.config.js`; simplifica manutenção e escalabilidade, porque ajustes de acessibilidade ou mudanças regulatórias são aplicados uma única vez e refletem-se em toda a interface; e fortalece a testabilidade ao permitir que camadas menores sejam validadas por testes unitários, enquanto organismos e templates recebem testes de integração.

Sob a ótica pedagógica, a granularidade do Atomic Design possibilita vincular moléculas e organismos a verbos cognitivos da Taxonomia de Bloom, assegurando que cada fragmento da interface cumpra uma função instrucional específica — reconhecer, aplicar ou analisar o conhecimento apresentado [25]. Além disso, componentes como cartões de quiz e linhas do tempo interativas encapsulam lógica de feedback imediato, reforçando a motivação do aluno e a assimilação dos conteúdos, enquanto os templates funcionam como protótipos

tangíveis que designers instrucionais podem revisar antes da implementação, reduzindo retrabalho [26].

Na prática, o repositório do ModfyJS passou a incluir o diretório ``src/atomic``, estruturado segundo os cinco níveis do modelo e adotando nomenclatura padronizada que facilita a descoberta e a reutilização dos componentes. Hooks dedicados — ``useAtomicTheme`` e ``useAtomicAnalytics`` — concentram tokens de estilo e eventos de telemetria, alimentando as métricas de eficácia didática definidas pelo modelo ADDIE. Experimentos em unidades piloto já indicam redução expressiva no tempo de implementação de novos módulos, queda nas regressões visuais após atualizações e aumento perceptível do engajamento medido por cliques em elementos interativos, corroborando que o Atomic Design, além de otimizar a engenharia de software, reforça a coesão pedagógica do ambiente AVAMEC.

## **2.4 Contribuições do Design Instrucional na otimização de processos**

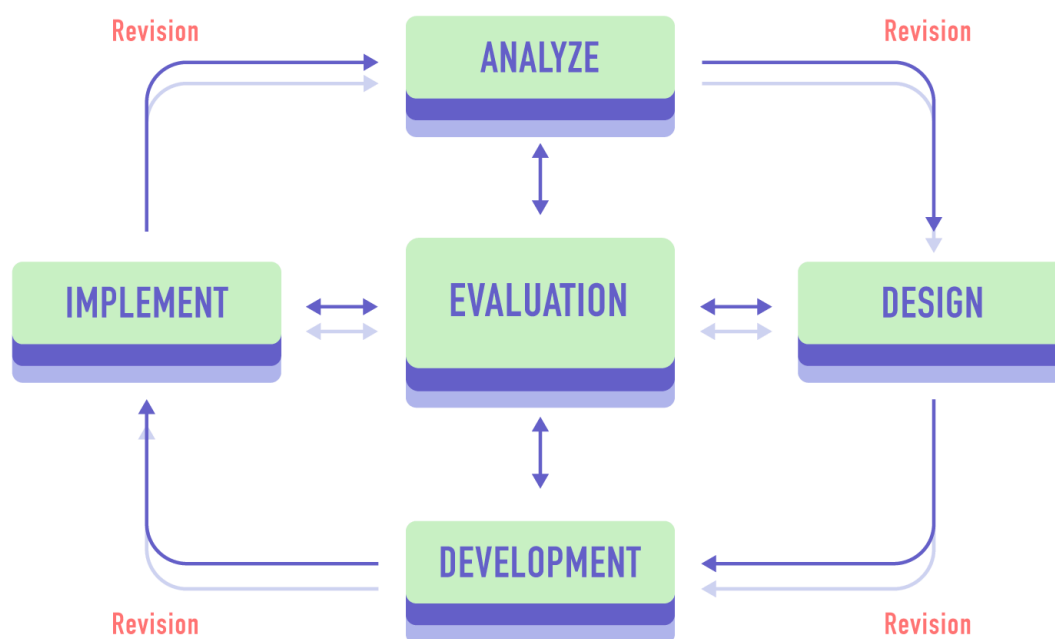
### **2.4.1 Aprofundamento ao Design Instrucional**

O design instrucional é uma abordagem sistemática que visa a criação de experiências de aprendizagem eficazes e engajadoras. Ele utiliza princípios pedagógicos para estruturar conteúdos educativos de maneira que facilite a compreensão e retenção por parte dos alunos. No contexto do desenvolvimento de cursos digitais para a plataforma AVAMEC, o design instrucional desempenha um papel crucial na otimização dos processos, garantindo que as tecnologias de engenharia de software educacional sejam utilizadas de maneira eficiente e alinhadas aos objetivos educacionais.

### **2.4.2 Modelo ADDIE**

Um dos principais frameworks utilizados no design instrucional é o modelo ADDIE [15], que se destaca pela sua estrutura clara e etapas bem definidas. ADDIE é um acrônimo para Análise, Design, Desenvolvimento, Implementação e Avaliação. Cada uma dessas etapas contribui de maneira significativa para a otimização dos processos na criação de cursos digitais.

**Figura 25-** ADDIE: 5 Steps To Effective Training



Learnupon, 2024.

O processo de desenvolvimento dos cursos na AVAMEC segue uma abordagem estruturada, dividida em cinco etapas principais: análise, design, desenvolvimento, implementação e avaliação.

Na fase de análise, são identificadas as necessidades educacionais e do público-alvo, garantindo que os cursos atendam às expectativas dos alunos e desenvolvam as competências desejadas. Além disso, são estabelecidos objetivos claros e mensuráveis, alinhados às diretrizes educacionais e tecnológicas da plataforma.

A etapa de design envolve a estruturação lógica e sequencial do conteúdo, organizando-o em módulos e unidades que facilitam a assimilação do conhecimento. Também são definidas as estratégias pedagógicas a serem utilizadas, como gamificação, atividades interativas e avaliações contínuas, visando aumentar o engajamento e a eficácia do aprendizado.

No desenvolvimento, são criados os materiais didáticos, utilizando tecnologias como React, TypeScript e Tailwind CSS para garantir modularidade e reutilização. Além disso, são desenvolvidos componentes interativos e atividades práticas, explorando as capacidades avançadas das novas ferramentas adotadas.

A implementação consiste na publicação e distribuição dos cursos na plataforma AVAMEC, preparando-os para o acesso dos alunos.

Por fim, na fase de avaliação, são realizadas avaliações formativas ao longo do curso para monitorar o progresso dos alunos e ajustar o conteúdo conforme necessário. Ao término, a avaliação sumativa mede o alcance dos objetivos de aprendizagem e a eficácia do curso, permitindo a coleta de feedbacks para futuras melhorias.

### **2.4.3 Integração com Engenharia de Software Educacional**

A aplicação do design instrucional no desenvolvimento de cursos para a AVAMEC, combinada com as práticas avançadas de engenharia de software educacional, proporciona uma série de benefícios, incluindo:

**Melhoria da Qualidade Educacional:** Por meio de uma estrutura bem definida e alinhada com objetivos pedagógicos claros, os cursos são mais eficazes e engajantes.

**Otimização do Desenvolvimento:** A modularização dos conteúdos e a reutilização de componentes educacionais reduzem significativamente o tempo e esforço necessário para desenvolver novos cursos.

**Feedback e Ajustes Contínuos:** A integração de metodologias ágeis como Scrum permite ajustes rápidos e contínuos com base no feedback dos usuários, garantindo a melhoria constante da qualidade dos cursos.

## **2.5 Estado da Arte**

Os avanços tecnológicos ocorridos na última década vêm intensificando a convergência entre design instrucional (DI) e engenharia de software educacional (ESE) [50]. No panorama atual, destacam-se cinco eixos de investigação e prática:

**Learning Analytics e Experience Analytics** – O tratamento de grandes volumes de dados discentes, por meio de padronizações como xAPI e Caliper, possibilita avaliações formativas em tempo quase real, intervenções pedagógicas adaptativas e refinamento contínuo de objetos de aprendizagem [27].

**Inteligência Artificial** – Sistemas de recomendação, tutores inteligentes e agentes conversacionais sustentam percursos personalizados, ajustando níveis de dificuldade, meios de feedback e sequenciamento de conteúdo [28].

**Realidade Estendida (XR: AR + VR + MR)** – Cenários imersivos de baixo custo, viabilizados por motores gráficos (Unreal Engine, Unity) e dispositivos stand-alone, favorecem aprendizagem experiencial em domínios que exigem prática segura ou observação in loco (medicina, engenharia, direito) [29].

**Arquiteturas baseadas em Componentes e Headless LMS** – A decomposição de funcionalidades instrucionais em micro-serviços e Web Components, consumidos via APIs, permite que múltiplas frentes de desenvolvimento atuem em paralelo, mantendo consistência visual e pedagógica por meio de bibliotecas de design system.

Essas frentes convergem para uma visão na qual o curso digital deixa de ser um produto estático e constitui um serviço informado por dados, evolutivo, escalável e responsivo a perfis variados de aprendizagem.

## **2.6 Quem Já Fez**

Diversas organizações – públicas, privadas e do terceiro setor – já demonstraram resultados tangíveis ao integrar DI e ESE:

O CESAR, por meio da CESAR School, é um centro de inovação em Recife que oferece formação em engenharia de software e design de artefatos digitais, alicerçada na metodologia de aprendizagem baseada em problemas, e promove projetos reais, como a “Fábrica Educacional de Software” para desenvolver soluções tecnológicas para empresas (fonte:

[https://pt.wikipedia.org/wiki/Centro\\_de\\_Estudos\\_e\\_Sistemas\\_Avan%C3%A7ados\\_do\\_Recife](https://pt.wikipedia.org/wiki/Centro_de_Estudos_e_Sistemas_Avan%C3%A7ados_do_Recife) ).

A Veduca é uma plataforma de educação a distância que disponibiliza videoaulas gratuitas e pagas, produzidas em parceria com instituições como USP e B3, e também desenvolve conteúdos personalizados e ambientes de aprendizado para treinamentos corporativos (fonte: <https://pt.wikipedia.org/wiki/Veduca>).

A D2L (Desire2Learn) é a criadora do Brightspace, um LMS com ênfase em ciência da aprendizagem, inteligência artificial e escalabilidade, amplamente adotado por instituições educacionais e corporativas no mundo todo (fonte: <https://www.d2l.com/brightspace>).

A Instructure, responsável pelo Canvas LMS, desenvolve um ambiente de aprendizagem intuitivo e baseado em software como serviço (SaaS), com foco em colaboração, acessibilidade e integração com ferramentas externas (fonte: <https://www.instructure.com/pt-br/canvas>).

O FenixEdu, originado no Instituto Superior Técnico em Lisboa, é um projeto de software open-source voltado à educação superior, que permite a participação de estudantes e profissionais em seu desenvolvimento, promovendo um verdadeiro ambiente de aprendizado por engenharia de software aplicada ao design instrucional e gestão acadêmica (fonte: <https://en.wikipedia.org/wiki/FenixEdu>).

Esses casos ilustram que, quando métodos de engenharia são alinhados a princípios instrucionais, instituições de perfis distintos obtêm ganhos em escala, qualidade e engajamento.

## **2.7 Abordagens Existentes**

Dentre as estratégias que despontam como promissoras, destacam-se:

**Microlearning, Nanolearning e Spaced Learning** – Fragmentação de conteúdo em “pílulas” de 2 a 8 minutos, acompanhadas de algoritmos de repetição espaçada (e.g., Leitner, SuperMemo) que otimizam retenção de longo prazo [30].

**Gamificação orientada a competência** – Mecânicas de progressão (badges, leaderboards), combinadas a frameworks como Competency-Based Education (CBE), dão feedback imediato e mapeiam evidências de desempenho a rubricas observáveis [31].

**Universal Design for Learning (UDL)** – Diretrizes que asseguram múltiplas formas de representação, ação e expressão, integradas desde o backlog de requisitos até a etapa de testes de usabilidade, garantindo acessibilidade nativa [32].

**Learning Experience Platforms (LXP)** – Camada sobreposta a LMS tradicionais, baseada em taxonomias de habilidade e machine learning de recomendação, que promove aprendizagem informal e curadoria social de conteúdo [33].

Ao combinar tais abordagens, programas educacionais alcançam maior personalização, escalabilidade e robustez, ao mesmo tempo, em que reduzem custos operacionais e aceleram ciclos de melhoria contínua. Desse modo, a intersecção entre DI e ESE projeta-se como vetor estratégico para atender às demandas emergentes de qualificação profissional e aprendizagem ao longo da vida em um cenário global dinâmico.



## 3. Metodologia

### 3.1 Estudo de Caso

O presente estudo adota uma abordagem do tipo estudo de caso (ref.) baseada em análises de documentos gerados ao longo de um projeto de cooperação para alcançar uma compreensão abrangente e detalhada sobre a otimização dos processos de desenvolvimento de cursos digitais na plataforma AVAMEC. A escolha por uma metodologia mista se justifica pela complexidade do objeto de estudo, que abrange tanto aspectos técnicos da engenharia de software quanto elementos pedagógicos do design instrucional.

A metodologia proposta visa integrar de forma holística a engenharia de software educacional e o design instrucional, promovendo uma sinergia, resultando em cursos digitais mais eficientes e eficazes. Por meio de uma análise criteriosa e da implementação de tecnologias modernas, espera-se alcançar uma otimização significativa dos processos de desenvolvimento na plataforma AVAMEC, contribuindo para a excelência na educação à distância no Brasil.

### 3.2 Objetivo Geral

O objetivo central é investigar e implementar práticas avançadas de engenharia de software e design instrucional no desenvolvimento de cursos digitais, visando otimizar processos para aumentar a eficiência e melhorar a qualidade dos cursos oferecidos pela plataforma AVAMEC.

#### 3.2.1 Objetivos Específicos

**Analisar os Processos Atuais de Desenvolvimento de Cursos:** Documentar e mapear os fluxos de trabalho, identificar gargalos e ineficiências, e levantar dados sobre o tempo de desenvolvimento e a qualidade do desenvolvimento dos cursos.

**Desenvolver e Aplicar Novas Tecnologias:** Implementar e avaliar a nova arquitetura tecnológica baseada em React, TypeScript e Tailwind CSS, comparando-a com a arquitetura anterior em termos de desempenho e qualidade.

**Avaliar os Impactos das Mudanças:** Medir os impactos das mudanças na qualidade e eficiência dos cursos desenvolvidos, utilizando métricas quantitativas e qualitativas.

### 3.3. Contexto: Descrição do processo atual

A pesquisa é realizada no contexto de um projeto de inovação cujo objetivo é a produção de cursos autoinstrucionais para o Ministério da Educação. O processo adotado é o seguinte:

**Fase Inicial: Análise dos Processos Atuais e Levantamento de Dados Iniciais:** A primeira fase do processo envolve uma análise detalhada dos métodos atuais de desenvolvimento de cursos na plataforma AVAMEC. Nessa fase, os fluxos de trabalho são documentados, gargalos e ineficiências são identificados sendo coletados dados sobre o tempo de desenvolvimento e a qualidade dos cursos. Isso inclui a análise da arquitetura de software utilizada e a revisão do código, buscando mapear o estado atual dos processos e estabelecer uma linha de base para futuras melhorias.

**Fase de Implementação: Desenvolvimento de Protótipos e Aplicação de Novas Tecnologias:** Com base nos dados e insights obtidos na fase inicial, a fase de implementação foca no desenvolvimento de protótipos utilizando tecnologias mais modernas, como React, TypeScript e Tailwind CSS. Esses protótipos são desenvolvidos em alinhamento com práticas de design instrucional para garantir que os novos processos atendam tanto às necessidades pedagógicas quanto às técnicas. O objetivo é modularizar o desenvolvimento, melhorar a reusabilidade dos componentes e facilitar a manutenção dos cursos. Durante essa fase, também são realizados testes de desempenho para avaliar a eficácia da nova arquitetura tecnológica.

**Fase de Avaliação: Coleta de Dados de Desempenho e Comparação de Arquiteturas:** Após a implementação dos protótipos, a fase de avaliação envolve a coleta de dados de desempenho e qualidade dos cursos desenvolvidos tanto na arquitetura anterior quanto na nova. Essa análise comparativa visa medir os impactos das mudanças, utilizando métricas quantitativas e qualitativas para avaliar a eficiência das novas práticas e tecnologias. Serão observados aspectos como redução do tempo de desenvolvimento, processos de Design Instrucional bem como prática de engenharia de software.

**Fase de Otimização Contínua: Aplicação de Melhorias e Documentação das Melhores Práticas:** A última fase do processo é caracterizada pela otimização contínua dos cursos e das práticas de desenvolvimento. Com base nos resultados da fase de avaliação, serão aplicadas melhorias nos processos e tecnologias utilizadas. Além disso, as melhores

práticas identificadas ao longo do processo serão documentadas para servir como referência para futuras iterações e projetos na plataforma AVAMEC. Isso garantirá que as lições aprendidas sejam incorporadas de forma sistemática, promovendo uma evolução constante na qualidade e eficiência dos cursos desenvolvidos.

### **3.4. Como este estudo pode ser reproduzido**

#### **Parte 1: tecnologias**

1. Acesse o seguinte repositório do GitHub: <https://github.com/gabzitto/AVAMEC.git>. Esse repositório contém o código-fonte base com as tecnologias mais modernas descritas neste presente trabalho.
2. É importante que tenha o Node instalado na sua máquina na versão 16.7.13 e o gerenciador de pacotes Yarn.
3. Clone o repositório e execute o comando “*yarn install*” na pasta root do projeto.
4. Rode o projeto com “*yarn start*” na pasta root do projeto.
5. Adapte o projeto para suas necessidades conforme o conteúdo do curso previamente planejado.

O projeto está dividido em componentes e módulos. É importante adotar a prática de componentização para otimizar o tempo de desenvolvimento, diminuir a incidência de erros e tornar mais prático a manutenção do curso no decorrer do tempo de desenvolvimento, conforme será explicado com maior profundidade mais a frente.

Cada módulo é composto por slides. Os slides são cada página que estará disponível por vezes ao usuário quando ele estiver acessando.

#### **Parte 2: processos**

Para esse passo, é importante ter pessoas capacitadas em fundamentos e conceitos de design instrucional que irão se integrar junto ao time de desenvolvimento e ao time de design.

Esse especialista coordenará os times de design e tecnologias. Além de ser responsável por analisar o conteúdo que passará do material bruto para curso feito em um

ambiente de prototipação (recomenda-se o uso do Figma). Para, então, ser posteriormente desenvolvido.

O especialista de design instrucional é fundamental para atingir as metas pedagógicas do curso e garantir que o material feito pelo time de desenvolvimento atenda bem aos requisitos do usuário que o conteúdo do curso a ser produzido exige.

### 3.5 Coleta e análise de Dados

A coleta de dados será realizada por meio de várias técnicas, incluindo a análise de código-fonte, a documentação de fluxos de trabalho, e a avaliação de materiais didáticos. A seguir, detalhamos as principais fontes de dados:

Serão analisados repositórios de código dos cursos desenvolvidos para avaliar a eficiência do código, a modularização e a reutilização de componentes. Além disso, a documentação dos processos atuais de desenvolvimento será revisada para identificar pontos críticos e áreas de melhoria. Também serão realizados testes de desempenho comparativos entre a arquitetura anterior, baseada em HTML, CSS e JavaScript, e a nova arquitetura, que utiliza React, TypeScript e Tailwind CSS.

Para a análise quantitativa, serão utilizadas métricas específicas, como:

- **Tempo de Desenvolvimento:** Medido em dias desde o início até a conclusão do curso.
- **Número de Commits:** Representando a quantidade de mudanças no código, logo uma estimativa de quantas vezes um curso precisou passar por ajustes mesmo após sua publicação oficial.
- **Quantidade de Alunos:** Avaliando o impacto dos cursos na quantidade de matriculados.

**A análise qualitativa envolverá:**

- **Revisão da Integração Tecnológica com o Design Instrucional:** Avaliação de como as tecnologias modernas (React, TypeScript, Tailwind CSS) foram integradas com as

práticas de design instrucional. Essa análise focará em identificar como a modularidade e a reutilização de componentes impactaram a qualidade pedagógica dos cursos, assegurando que as soluções técnicas suportem eficazmente os objetivos educacionais.

- **Análise de Fluxos de Trabalho e Eficiência do Processo:** Estudo detalhado dos fluxos de trabalho utilizados no desenvolvimento dos cursos, com ênfase em como as novas tecnologias e práticas de design instrucional contribuíram para a otimização dos processos. Serão analisados aspectos como a redução de tempo de desenvolvimento, a melhoria na colaboração entre as equipes e a eficácia das práticas ágeis implementadas.

Para assegurar a validade e a confiabilidade dos dados coletados, serão adotados os seguintes procedimentos:

- **Análise de Dados:** Coleta de múltiplas fontes de dados do AVAMEC e GitHub para validar as conclusões.

#### **3.4.1 Ferramentas e Tecnologias Utilizadas**

- HTML, CSS, Javascript, React, ModfyJS, TypeScript, Tailwind CSS, Git, Github e FIGMA: Para desenvolvimento tecnológico.
- **Ferramentas de Análise de Código:** Como ESLint e Prettier para avaliar a qualidade do código.
- **Diagrams.net:** para a criação de diagramas e fluxogramas.
- **Plataforma AVAMEC:** Para a implementação e teste dos cursos desenvolvidos.

## 4. Desenvolvimento da solução

### 4.1. Descrição da Arquitetura Anterior

#### 4.1.1 Contexto Tecnológico

A arquitetura anterior utilizada pelo time de desenvolvimento do V-Lab para a plataforma AVAMEC é baseada em tecnologias web tradicionais como HTML, CSS e JavaScript. Essas tecnologias têm sido fundamentais na construção de cursos digitais, permitindo a criação de interfaces de usuário e a incorporação de conteúdos multimídia de forma acessível. A seguir, detalhamos o uso dessas tecnologias:

HTML (HyperText Markup Language) é a espinha dorsal de qualquer página web. No contexto dos cursos AVAMEC, HTML é utilizado para estruturar o conteúdo, definindo a disposição de textos, imagens, vídeos e outros elementos multimídias. A marcação HTML fornece a base sobre a qual CSS e JavaScript operam [1]. Um exemplo prático de uma página HTML para um curso poderia incluir uma estrutura básica com cabeçalho, seções de conteúdo e rodapé.

**Figura 1** – Captura de tela de código exemplo de uso de HTML

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Curso AVAMEC</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Bem-vindo ao Curso AVAMEC</h1>
  </header>
  <main>
    <section id="module1">
      <h2>Módulo 1: Introdução</h2>
      <p>Este módulo cobre os conceitos básicos de...</p>
      <video controls>
        <source src="intro.mp4" type="video/mp4">
        Seu navegador não suporta o elemento de vídeo.
      </video>
    </section>
  </main>
  <script src="scripts.js"></script>
</body>
</html>

```

Elaboração própria, 2024

CSS (Cascading Style Sheets) é responsável pela estilização das páginas. Ele permite a definição de estilos visuais, como cores, fontes, espaçamentos e layout. No desenvolvimento dos cursos AVAMEC, CSS é utilizado para garantir que as páginas sejam visualmente atraentes e consistentes. Ele também permite que a plataforma se adapte a diferentes dispositivos e tamanhos de tela, o que é crucial para a acessibilidade e a experiência do usuário [2]. Por exemplo, o CSS pode ser utilizado para criar um layout responsivo que ajuste o conteúdo adequadamente em dispositivos móveis e desktops.

### **Exemplo Prático e Básico de Implementação em CSS:**

**Figura 2** – Captura de tela de código exemplo de uso de CSS

```
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 0;  
    background-color: #f4f4f4;  
}  
  
h1, h2 {  
    color: #333;  
}  
  
section {  
    margin: 20px;  
    padding: 20px;  
    background-color: #fff;  
    border-radius: 8px;  
}  
  
form {  
    margin-top: 20px;  
}  
  
input[type="submit"] {  
    background-color: #28a745;  
    color: white;  
    padding: 10px 20px;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
}
```

Elaboração própria, 2024

JavaScript adiciona interatividade às páginas web. No contexto dos cursos AVAMEC, JavaScript é utilizado para criar funcionalidades dinâmicas, como quizzes e animações interativas que enriquecem a experiência do aluno [3]. Scripts JavaScript também são usados para validar formulários, controlar a navegação entre módulos do curso e armazenar



temporariamente dados do usuário. Por exemplo, um script JavaScript pode ser utilizado para validar a entrada do usuário em uma atividade avaliativa e dar feedback imediato.

**Figura 3** – Captura de tela de código exemplo de uso de JavaScript.

```
document.getElementById('quiz1').addEventListener('submit', function(event) {  
    event.preventDefault();  
    var selected = document.querySelector('input[name="q1"]:checked');  
    if (selected) {  
        alert('Você selecionou: ' + selected.value);  
    } else {  
        alert('Por favor, selecione uma opção.');    }  
});
```

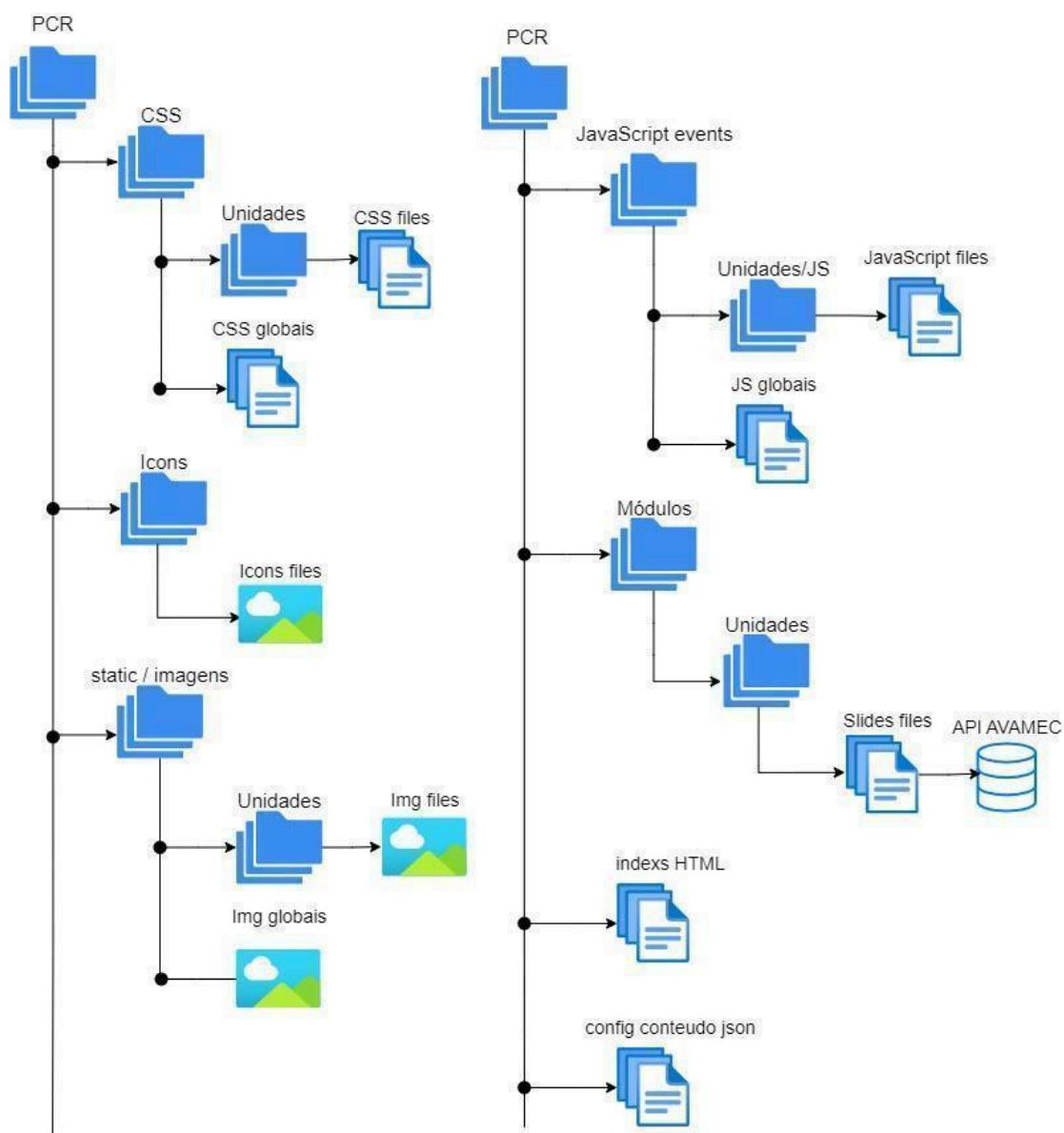
Elaboração própria, 2024

#### **4.1.2 Estrutura e análise da arquitetura anterior**

A arquitetura anterior utilizada pelo time de desenvolvimento do V-Lab para a plataforma AVAMEC, como citado anteriormente, era baseada em tecnologias web tradicionais como HTML, CSS e JavaScript. Essa estrutura tecnológica, apesar de robusta e amplamente adotada, apresenta desafios significativos em termos de modularização, manutenção e escalabilidade quando aplicada em projetos de grande escala e complexidade [4], como os cursos digitais oferecidos pela AVAMEC.

A imagem abaixo ilustra a arquitetura anterior de um curso chamado PCR, desenvolvido exclusivamente com HTML, CSS e JavaScript. A seguir, descrevemos detalhadamente essa arquitetura, destacando as principais pastas e arquivos que compõem a estrutura do curso.

**Figura 4** – Captura de tela da arquitetura do sistema antigo.



Elaboração própria, 2024

#### 4.1.3 Descrição da Imagem da Arquitetura Anterior

A imagem é um diagrama que representa a estrutura de diretórios e arquivos do curso PCR (Plano de Carreira e Remuneração). Essa estrutura é dividida em dois grandes blocos: um para CSS e arquivos estáticos, e outro para JavaScript e HTML, refletindo a separação das responsabilidades de estilização e interatividade no curso.

A estrutura de arquivos para o desenvolvimento de cursos na plataforma AVAMEC é organizada em diferentes diretórios, cada um com uma função específica, visando tanto a modularização quanto a consistência visual e funcional dos cursos.

O diretório de CSS/arquivos estáticos é dividido em duas categorias principais. Primeiramente, os arquivos CSS específicos de cada unidade do curso, responsáveis por aplicar estilos únicos a cada seção. No entanto, essa abordagem pode resultar em uma repetição de código, complicando a manutenção dos estilos ao longo do tempo. Em contraste, os arquivos CSS globais aplicam estilos uniformes a todas as unidades do curso, garantindo uma aparência consistente. No entanto, essa centralização pode dificultar a personalização dos estilos para unidades específicas.

O diretório Icons armazena os arquivos de ícones utilizados no curso. Esses ícones são estilizados com CSS e desempenham um papel fundamental na melhoria da interface do usuário, aplicados em diversas partes do conteúdo para proporcionar uma experiência mais visualmente atraente.

No diretório Static/Imagens, as imagens são organizadas em subdiretórios específicos para cada unidade do curso. Isso inclui tanto imagens específicas de cada unidade, armazenadas em seus respectivos subdiretórios, quanto imagens globais utilizadas em várias unidades e armazenadas em um diretório comum, facilitando o acesso e a reutilização.

Na Estrutura JavaScript/HTML, encontramos o diretório JavaScript Events, que contém arquivos relacionados aos eventos JavaScript, como cliques e movimentos do mouse, que interagem diretamente com o usuário. Além disso, há um diretório para Unidades/JS, onde são armazenados os arquivos JavaScript específicos de cada unidade, adicionando interatividade e funcionalidades dinâmicas. Complementando essa estrutura, os JS globais contêm funções e scripts reutilizáveis por todas as unidades, promovendo a consistência e reduzindo a duplicação de código.

O diretório de Módulos contém scripts que implementam módulos específicos do curso, permitindo a criação de funcionalidades avançadas e a reutilização de componentes. Já o diretório Slides armazena os arquivos relacionados às páginas do curso, sincronizadas com a API AVAMEC para carregar conteúdos dinâmicos e registrar dados na plataforma.

Finalmente, o arquivo Index HTML constitui a base estrutural de cada unidade do curso, sendo vinculado aos arquivos CSS e JavaScript para renderizar a interface e funcionalidades completas. O Config Conteúdo JSON é um arquivo de configuração em formato JSON, que define as propriedades e parâmetros dos conteúdos, permitindo uma gestão centralizada e eficiente dos recursos do curso, utilizado pelo AVAMEC para determinar e reconhecer cada módulo.

#### **4.1.4 Análise reflexiva**

A arquitetura anterior baseada em HTML, CSS e JavaScript apresenta as seguintes características:

A ausência de uma estrutura modular bem definida gera duplicação de código e dificulta a manutenção, especialmente à medida que o curso se torna mais complexo e extenso. Além disso, a necessidade de atualizar estilos ou scripts em múltiplos arquivos aumenta o risco de inconsistências e erros. A abordagem tradicional também impõe desafios à escalabilidade da plataforma, tornando a adição de novos recursos e funcionalidades um processo complexo, com possíveis impactos no desempenho. Por fim, a implementação de interatividade se torna mais trabalhosa e propensa a falhas, uma vez que exige a escrita manual de código JavaScript para cada interação.

A transição para uma arquitetura utilizando tecnologias e frameworks modernos visa mitigar esses desafios, proporcionando uma estrutura mais modular, eficiente e fácil de manter. As novas tecnologias permitem criar componentes reutilizáveis, facilitam a manutenção do código e melhoram a escalabilidade, resultando em um processo de desenvolvimento mais ágil e na entrega de cursos de maior qualidade.

#### **4.1.5 Análise de Design Instrucional com a nova arquitetura**

No contexto dessa arquitetura, o design instrucional é aplicado de maneira a estruturar o conteúdo educacional de forma lógica e pedagógica. O design instrucional é uma abordagem sistemática para o desenvolvimento de materiais educacionais e experiências de aprendizagem, visando garantir que os objetivos educacionais possam a vir de forma eficiente e eficaz [5].

**Suporte Pedagógico:** Atualmente, a estrutura de design instrucional na plataforma AVAMEC inclui a definição de objetivos claros, a divisão do conteúdo em módulos gerenciáveis e a incorporação de avaliações contínuas. A aplicação do modelo ADDIE (Análise, Design, Desenvolvimento, Implementação e Avaliação) é evidente, onde cada etapa do processo instrucional é cuidadosamente planejada e executada para garantir que os materiais educacionais sejam eficazes.

**Objetivos de Aprendizagem:** A plataforma AVAMEC define objetivos de aprendizagem claros e mensuráveis para cada curso. Esses objetivos são fundamentais para guiar o desenvolvimento do conteúdo e garantir que todos os materiais didáticos estejam alinhados com as metas educacionais.

**Divisão em Módulos:** O conteúdo dos cursos é dividido em módulos gerenciáveis, facilitando a assimilação gradual dos conceitos pelos alunos. Cada módulo inclui uma combinação de textos, imagens, vídeos e atividades interativas. A modularização do conteúdo facilita a revisão e atualização dos materiais, mas a arquitetura descrita anteriormente limitava a reutilização de componentes, exigindo personalizações manuais que aumentam o tempo de desenvolvimento.

**Avaliações Contínuas:** A integração de avaliações contínuas é uma prática essencial do design instrucional, permitindo que os alunos recebam feedback regular sobre seu progresso. No entanto, a plataforma enfrenta desafios para dar feedback imediato e personalizado, uma vez que essa arquitetura não suporta plenamente essas funcionalidades. Ferramentas de análise de desempenho e sistemas de feedback automatizados são necessários para melhorar a eficácia das avaliações.

#### **4.1.6 Desafios e Limitações da arquitetura anterior**

Apesar da robustez das tecnologias HTML, CSS e JavaScript, tal arquitetura apresenta várias limitações:

##### **4.1.6.1 Ineficiências Técnicas da arquitetura anterior**

Em arquiteturas que carecem de uma estruturação modular apropriada, a manutenção de códigos extensos torna-se um desafio considerável. Pequenas alterações exigem revisões significativas em diversas partes do código, devido à ausência de módulos bem definidos. Adicionalmente, a falta de otimização pode comprometer o desempenho da plataforma,

resultando em lentidão na renderização de páginas complexas e com grande número de elementos interativos, prejudicando a experiência do usuário [6]. A integração de tais elementos interativos também se mostra problemática, uma vez que a implementação de funcionalidades avançadas frequentemente demanda soluções alternativas com baixa sustentabilidade a longo prazo.

#### **4.1.6.2 Ineficiências Instrucionais da arquitetura anterior**

**Rigidez na Customização de Conteúdo:** A personalização dos cursos exige um esforço manual significativo, tornando o processo de adaptação de conteúdos para diferentes contextos e públicos-alvo moroso e suscetível a erros.

**Componentes Reutilizáveis:** Em uma arquitetura moderna, componentes de curso, como quizzes, vídeos e atividades interativas, podem ser desenvolvidos como módulos independentes que podem ser facilmente reutilizados em diferentes cursos. No entanto, uma arquitetura baseada somente em HTML, CSS e JavaScript dificulta essa modularização, resultando em um trabalho repetitivo e ineficiente para os desenvolvedores. A seguir é apresentada a figura 5 onde mostra o componente “Linha do Tempo”, no qual foi um componente muito complexo a ser desenvolvido utilizando essa arquitetura.

**Figura 5** – Componente Linha do Tempo.



V-LAB UFPE, 2024.

**Engajamento do Aluno e Interatividade:** A interatividade, um dos pilares do design instrucional eficaz, é essencial para o engajamento e a motivação dos alunos. Ferramentas como quizzes, jogos educativos, simulações e atividades práticas são fundamentais para criar experiências imersivas e manter os alunos envolvidos. No entanto, a arquitetura anterior da plataforma AVAMEC dificulta a implementação de elementos altamente interativos e de gamificação avançada, que utilizam elementos de jogos para tornar a aprendizagem mais envolvente. Essa limitação tem o potencial de impactar negativamente a personalização do aprendizado e, consequentemente, a motivação e o engajamento dos alunos.

**Feedback ao errar questões:** Implementação do feedback é crucial para o engajamento dos alunos. Feedback imediato e específico pode auxiliar os alunos a entenderem seus erros e aprenderem com eles. Essa arquitetura anterior limita a capacidade de dar feedback personalizado em tempo real, impactando negativamente a experiência de aprendizagem.

**Figura 6** – Módulo Avaliativo do curso Plano de Carreira e Remuneração (PCR)

**Questão 3**

Sobre o Plano de Cargos e Remuneração da Educação Básica pública, regulamentado pelas Leis 11.738/2008 e 13.005/2014, analise as seguintes afirmações:

- ☐ Segundo as Leis 11.738/2008 e 13.005/2014 somente os municípios com mais de 10 mil habitantes são obrigados a elaborar um Plano de Carreira e Remuneração da Educação Básica.
- ☒ A elaboração do PCR deve prever a evolução do servidor na carreira, segundo os parâmetros de progressão por qualificação e progressão por tempo, segundo tabela de pontos. Esse modelo de progressão incentiva à formação continuada e estimula boas práticas no cotidiano profissional.
- ☐ A elaboração do PCR deve condicionar as despesas da carreira funcional do quadro de servidores, as receitas do Fundeb, parte das transferências constitucionais e arrecadação própria do município.
- ☐ Segundo as Leis do Piso 11.738/2008 e do PNE 13.005/2014, o déficit da folha salarial gerado pelo PCR municipal ao longo do tempo, sempre será coberto por recursos da União.
- ☐ A Lei do Piso 11.738/2008 requer que o vencimento inicial da carreira deve ser pelo menos igual ao valor do piso salarial profissional nacional em vigor.

[Feedback geral](#)

V-LAB UFPE, 2024

#### 4.1.7 Estratégias Adotadas e Limitações Correntes

Para lidar com algumas dessas limitações, a equipe de desenvolvimento tem adotado várias estratégias. No entanto, essas estratégias nem sempre são suficientes para superar todos os desafios.

##### **Estratégias Atuais:**

**Uso de Bibliotecas JavaScript:** Bibliotecas como jQuery são utilizadas para simplificar a manipulação do DOM e adicionar interatividade às páginas. Embora úteis, essas bibliotecas não resolvem todos os problemas de modularização e reutilização de código.

**Frameworks CSS:** Frameworks como Bootstrap são utilizados para acelerar o desenvolvimento de interfaces responsivas. No entanto, isso pode adicionar complexidade e inflar o tamanho do código, além de não oferecerem a flexibilidade necessária para todos os casos de uso.



### **Limitações Correntes:**

**Complexidade de Integração:** A integração de novas funcionalidades frequentemente requer modificações significativas no código existente, o que pode introduzir novos bugs e aumentar a complexidade do projeto.

**Escalabilidade Limitada:** À medida que o número de módulos do curso e a complexidade das interações aumentam, essa arquitetura enfrenta dificuldades para escalar eficientemente. Isso resulta em tempos de carregamento mais longos e uma experiência de usuário menos fluida.

**Manutenção Intensiva:** A falta de modularização e reutilização de componentes significa que a manutenção do código é intensiva em termos de tempo e recursos. Pequenas mudanças podem requerer revisões extensas e testes rigorosos para garantir que não introduzem novos problemas.

### **4.1.8 Conclusão**

O objetivo deste trabalho é realizar a análise comparada de duas arquiteturas apoiadas pelo design instrucional utilizadas na plataforma AVAMEC assim como demonstrar a evolução de processos na intersecção entre engenharia de software educacional e design instrucional.

A análise da arquitetura anterior revela a necessidade urgente de transição para uma stack tecnológica mais moderna e robusta. Tecnologias como React, TypeScript e Tailwind CSS oferecem soluções para muitos dos desafios e limitações identificados. Essas tecnologias permitirão uma maior modularização, reutilização de componentes e integração de estratégias pedagógicas avançadas, resultando em um processo de desenvolvimento mais eficiente e em uma experiência de aprendizagem significativamente melhorada para os usuários da plataforma AVAMEC.

Por fim, a aplicação dessas novas tecnologias em conjunto com práticas avançadas de engenharia de software e design instrucional promete elevar o padrão dos cursos oferecidos, alinhando eficiência técnica e eficácia pedagógica para atender às demandas crescentes por educação à distância de alta qualidade no Brasil.

## **4.2. Otimização de Processos**

Com a implementação da nova arquitetura baseada em React, TypeScript e Tailwind CSS, aliada às práticas avançadas de design instrucional, foi possível iniciar uma otimização significativa dos processos de desenvolvimento de cursos digitais para a plataforma AVAMEC. A integração dessas tecnologias trouxe uma maior modularização e reutilização de componentes, reduzindo significativamente os tempos de desenvolvimento e a propensão a erros. Aqui, a junção entre desenvolvimento e design instrucional é a chave para entregar conteúdos educacionais de alta qualidade, capazes de atingir inúmeros usuários. Mas isso por si só não é o suficiente, é necessário fazer uma análise profunda de processos atuais a fim de aumentar a qualidade dos cursos desenvolvidos.

### **4.2.1 Definição de Processo**

Processo, neste contexto, refere-se à combinação de práticas de desenvolvimento de software educacional com estratégias de design instrucional para criar cursos digitais eficientes e pedagogicamente eficazes. A adoção de metodologias ágeis, especialmente o Scrum, desempenhou um papel crucial na organização e melhoria contínua desses processos.

### **4.2.2 Scrum e a Otimização de Processos**

A metodologia Scrum foi introduzida para estruturar o fluxo de trabalho da equipe de desenvolvimento e design instrucional. No Scrum, o trabalho é dividido em sprints, os quais são ciclos de desenvolvimento curtos e iterativos, geralmente com duração de duas a quatro semanas. Cada sprint envolve etapas de planejamento, desenvolvimento, revisão e retrospectiva [14], garantindo a melhoria contínua e a adaptação rápida às mudanças ou feedbacks, o que permite desenvolver um sistema eficiente do processo de engenharia de software que atende as necessidades da equipe de desenvolvimento para entregar um material de alta qualidade.

#### **4.2.2.1 Planejamento de Sprints**

No início de cada sprint, a equipe realiza uma reunião de planejamento para definir as metas e tarefas a serem realizadas. As tarefas são priorizadas conforme o valor que agregam

ao projeto, e o backlog é constantemente atualizado para refletir as necessidades mais urgentes.

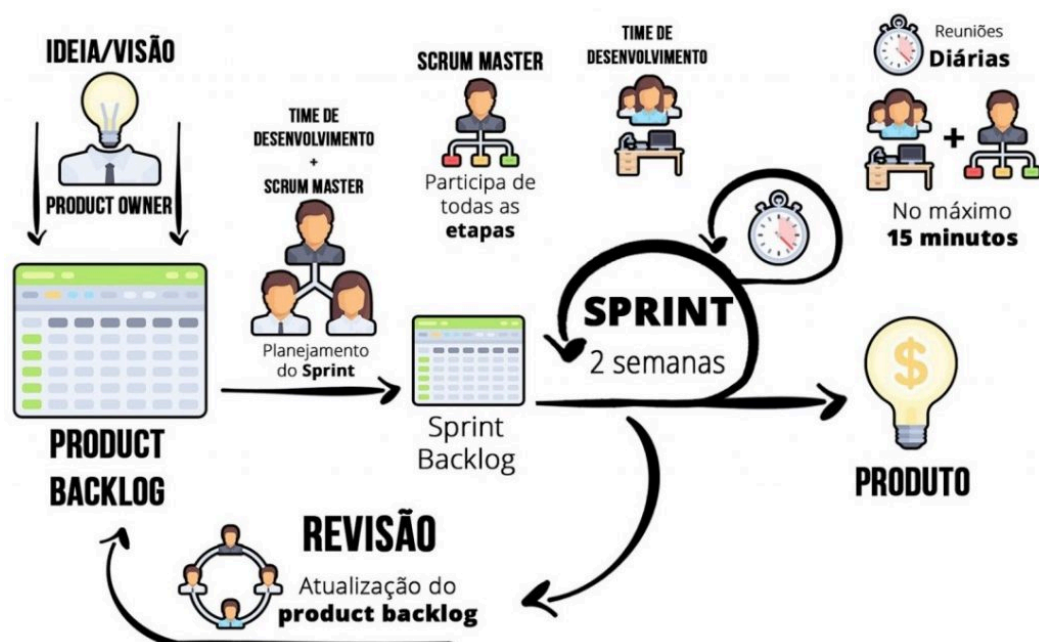
#### 4.2.2.2 Desenvolvimento e Testes

Durante o sprint, os desenvolvedores trabalham em estreita colaboração com os designers instrucionais para criar componentes de curso interativos e modulares. As funcionalidades são desenvolvidas, testadas e ajustadas continuamente, garantindo que cada componente atenda aos requisitos técnicos e pedagógicos.

#### 4.2.2.3 Revisões e Retrospectivas

No final de cada sprint, a equipe realiza uma revisão para demonstrar as funcionalidades desenvolvidas e obter feedback. As retrospectivas permitem que a equipe identifique pontos de melhoria e ajuste suas práticas para o próximo ciclo, promovendo uma cultura de melhoria contínua.

**Figura 21-** SCRUM: entenda a metodologia utilizada na Gestão de Projetos



### 4.2.3 Aplicação da otimização de processos

Na **Figura 22**, descreve o processo de desenvolvimento de um projeto, detalhando as etapas desde o início até a conclusão, sem incluir a importante participação do design instrucional no processo. Este diagrama mostra o fluxo de trabalho inicial, que representa o estado inicial de processos antes de ser otimizado, onde:

A imagem apresentada mostra o fluxo de desenvolvimento de um projeto no contexto de criação de cursos para a plataforma AVAMEC. Aqui está a descrição detalhada do passo a passo ilustrado na imagem:

**Início do Projeto:** Esta fase marca o começo do projeto, onde são definidos os objetivos, as expectativas, e o planejamento inicial. As equipes envolvidas, como Design, Desenvolvimento e Design Instrucional, alinham suas atividades e recursos para garantir um início organizado e estruturado.

**Prototipação:** Nesta etapa, são criados protótipos dos cursos ou das funcionalidades principais que serão desenvolvidas. A prototipação permite a visualização inicial do produto, possibilitando ajustes antes da fase de desenvolvimento. É uma fase crucial para validar ideias e identificar possíveis melhorias.

**Desenvolvimento:** Com os protótipos aprovados, a equipe avança para o desenvolvimento propriamente dito. Aqui, o código é escrito, os componentes são desenvolvidos e as funcionalidades são integradas. Esta fase é intensiva em termos de codificação e envolve a aplicação de tecnologias como React, TypeScript, e TailwindCSS, conforme descrito no ModfyJS.

**Testes e Ajustes:** Após o desenvolvimento, o projeto passa por uma fase rigorosa de testes. Nessa etapa, são identificados bugs, problemas de usabilidade e quaisquer inconsistências no curso. Se os testes falharem, o projeto retorna à fase de ajustes, onde os erros são corrigidos antes de uma nova rodada de testes.

**Publicação na AVAMEC:** Se o projeto passar nos testes, ele é publicado na plataforma AVAMEC. Esta fase envolve a implementação final dos cursos, onde eles se

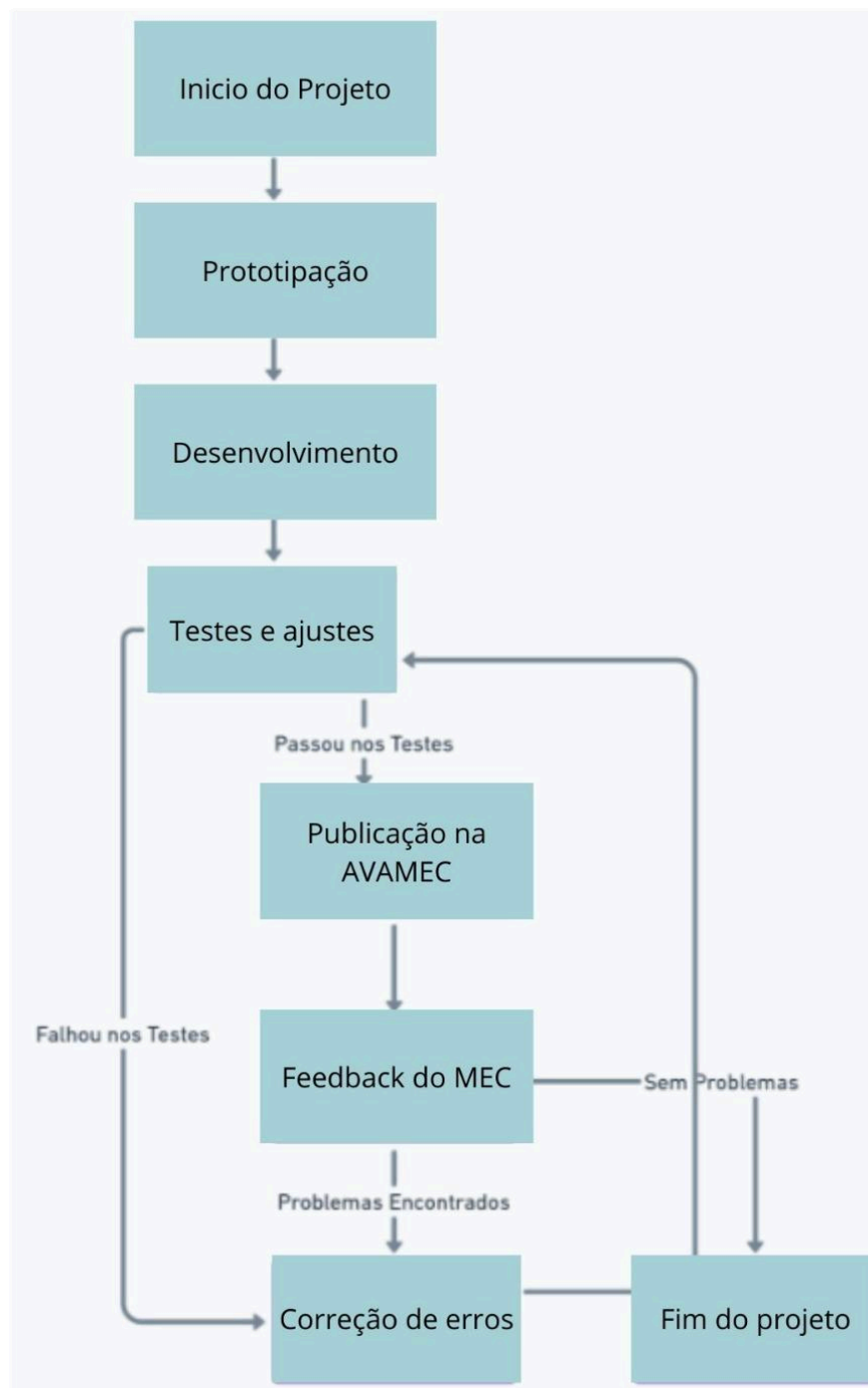
tornam acessíveis aos usuários (professores e alunos). É a transição do ambiente de desenvolvimento para o ambiente de produção.

**Feedback do MEC:** Após a publicação, o MEC (Ministério da Educação) revisa o curso e fornece feedback. Esse retorno é fundamental para garantir que os cursos estejam alinhados com as expectativas e necessidades educacionais.

**Correção de Erros:** Se o feedback do MEC identificar problemas, o projeto entra em uma fase de correção de erros. Os desenvolvedores voltam ao código para ajustar os pontos levantados, garantindo que o produto final atenda aos padrões estabelecidos.

**Fim do Projeto:** Com todas as correções implementadas e os problemas resolvidos, o projeto é concluído. Nesta fase, o curso é finalizado, e o projeto é oficialmente encerrado, estando pronto para o uso contínuo na plataforma AVAMEC.

**Figura 22** – Diagrama de funcionamento de processos, versão 1.0 Elaboração própria, 2024



Elaboração própria, 2024.

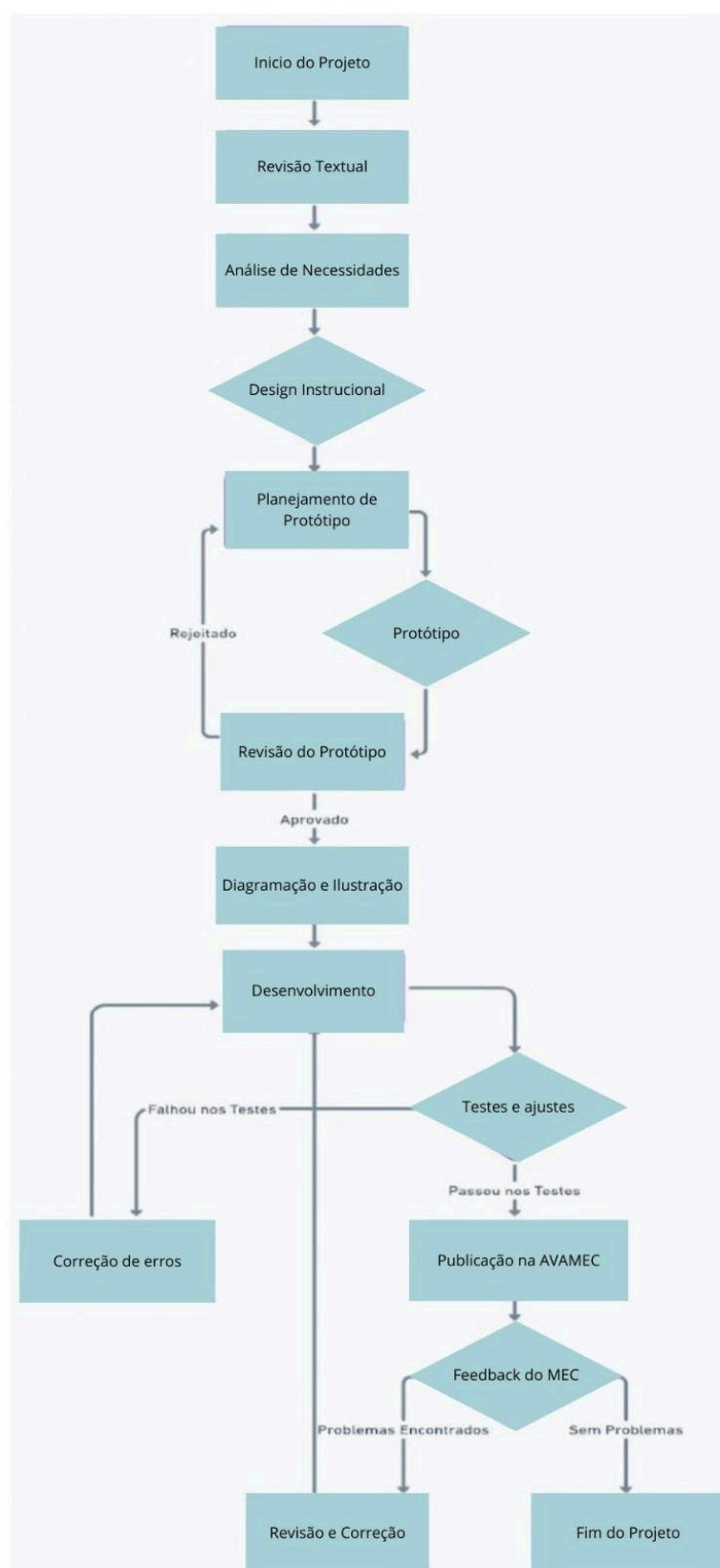
Na **Figura 23**, demonstra a aplicação da otimização de processos após uma análise interna ao longo de vários meses e melhoria contínua do nosso fluxo de trabalho integrado ao design instrucional:

1. **Início do Projeto:** Assim como na versão anterior, esta é a etapa inicial onde o projeto é formalmente iniciado, com a definição das diretrizes e dos objetivos principais.
2. **Revisão Textual:** Após o início, o projeto passa por uma revisão textual, onde o conteúdo é analisado e ajustado para garantir clareza e coerência com os objetivos do projeto.
3. **Análise de Necessidades:** Esta etapa adicional envolve a identificação e análise das necessidades do projeto. É uma fase crítica para garantir que todos os requisitos sejam compreendidos e considerados antes de avançar para o design.
4. **Design Instrucional:** Nesta fase, é desenvolvido um plano de design instrucional que serve como um guia para o desenvolvimento do protótipo. Esta etapa é fundamental para garantir que o projeto educacional atenda às necessidades pedagógicas e técnicas identificadas.
5. **Planejamento de Protótipo:** Com base no design instrucional, é elaborado um plano detalhado para o desenvolvimento do protótipo. Este planejamento inclui a definição de recursos, ferramentas e métodos a serem utilizados.
  - a. **Protótipo:** O protótipo é desenvolvido e avaliado. Nesta etapa, o projeto é testado em uma versão inicial para identificar possíveis falhas ou áreas de melhoria.
  - b. **Rejeitado:** Se o protótipo é rejeitado, ele passa por uma **Revisão do Protótipo**, onde são feitas as correções necessárias antes de ser testado novamente.
6. **Aprovado:** Se o protótipo é aprovado, ele avança para a próxima fase.

7. **Diagramação e Ilustração:** Após a aprovação do protótipo, o projeto passa por uma fase de diagramação e ilustração, onde os elementos visuais e gráficos são desenvolvidos e integrados.
8. **Desenvolvimento:** Com a diagramação e a ilustração concluídas, o projeto entra na fase de desenvolvimento, onde todas as componentes são integradas e o projeto é construído na íntegra.
9. **Testes e Ajustes:** Após o desenvolvimento, o projeto é submetido a testes rigorosos para identificar e corrigir quaisquer problemas. Se o projeto falhar nos testes, ele retorna para a fase de **Correções de Erros**, onde os problemas são corrigidos, e então volta para a fase de desenvolvimento para novas implementações e testes subsequentes.
10. **Publicação na AVAMEC:** Quando o projeto passa nos testes, ele é publicado na AVAMEC, tornando-se disponível para os usuários.
11. **Feedback do MEC:** Após a publicação, o projeto recebe feedback do Ministério da Educação (MEC), essencial para avaliar sua eficácia e conformidade com os requisitos educacionais.
  - a. **Problemas Encontrados:** Se o feedback do MEC identifica problemas, o projeto entra na fase de **Revisão e Correção**, onde os problemas são analisados e corrigidos antes de ser reavaliado.
  - b. **Sem Problemas:** Se não forem encontrados problemas no feedback do MEC, o projeto é considerado concluído.
12. **Fim do Projeto:** Esta é a etapa final do fluxograma, onde o projeto é formalmente encerrado após atender a todos os requisitos e não apresentar mais problemas.



**Figura 23** – Diagrama de funcionamento de processos versão 2.0



Elaboração própria, 2024.

#### 4.2.4 Integração de Design Instrucional

O design instrucional foi integrado ao processo de desenvolvimento desde as fases iniciais de planejamento até a publicação do curso no AVAMEC. Isso assegura que os objetivos de aprendizagem sejam claramente definidos e que os materiais didáticos sejam eficazes e engajantes.

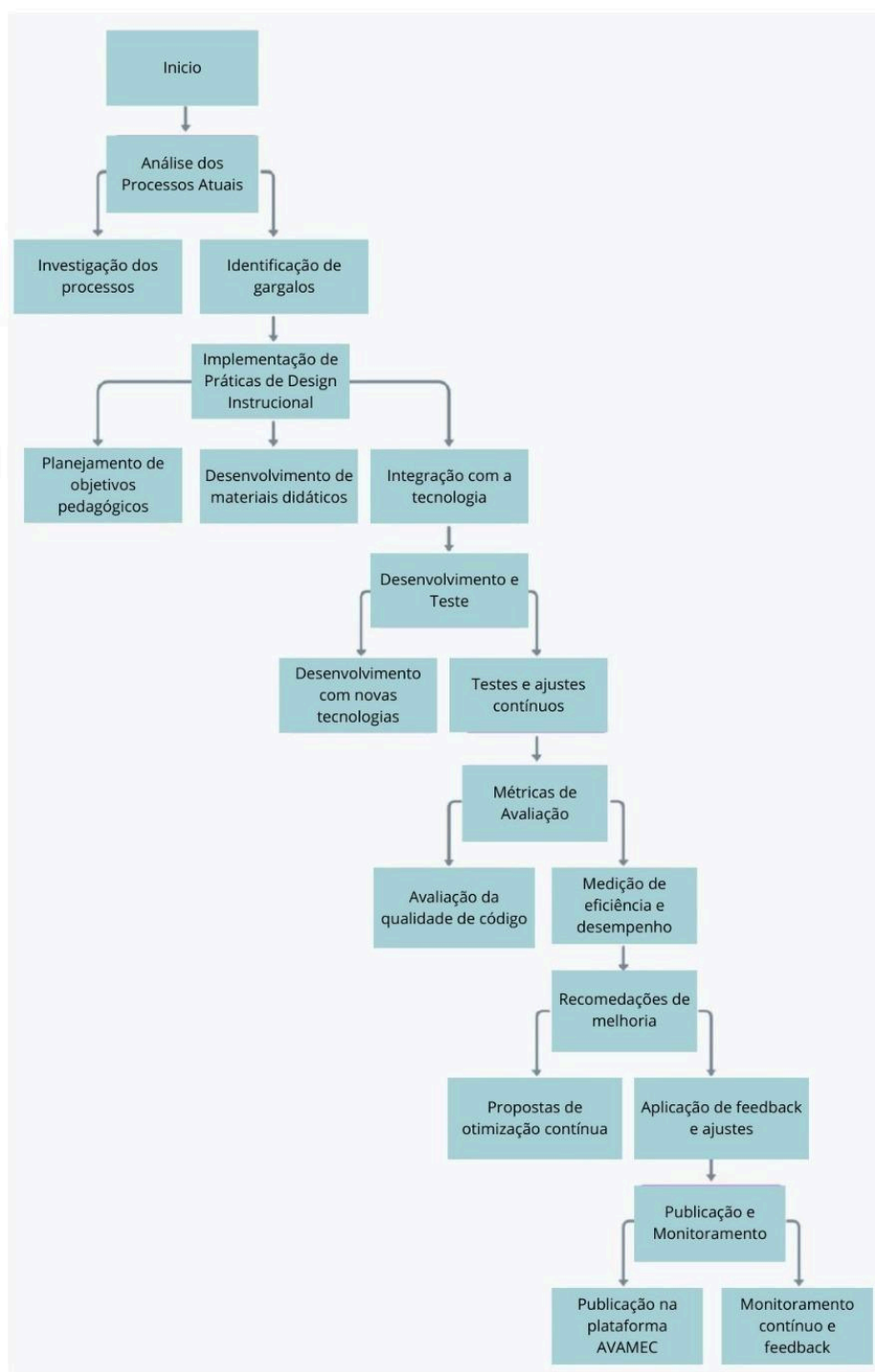
Na **Figura 24**, apresentamos um diagrama de fluxo de trabalho com design instrucional que complementa o que foi mostrado na Figura 23. Este diagrama é fundamentado na ideia de que é essencial reavaliar continuamente os processos e tecnologias atuais para alcançar resultados ainda mais eficazes, especialmente à medida que surgem novos contextos e tecnologias avançadas. No mundo da computação, novas frameworks são desenvolvidas anualmente, que, por vezes, superam as anteriores e oferecem novas oportunidades para otimização e inovação no desenvolvimento de projetos que possibilitam uma maior qualidade, escalabilidade e alcance de usuários.

1. **Início:** A etapa inicial do projeto, onde se dá a largada oficial ao processo de desenvolvimento, estabelecendo os objetivos principais e as diretrizes gerais.
2. **Análise dos Processos Atuais:** Esta fase envolve a análise detalhada dos processos existentes. É dividida em duas subetapas:
  - a. **Investigação dos Processos:** Avaliação e documentação dos processos atuais para entender seu funcionamento e identificar pontos fortes e fracos.
  - b. **Identificação de Gargalos:** Identificação dos principais obstáculos e ineficiências nos processos atuais que precisam ser abordados para melhorar o fluxo de trabalho.
3. **Implementação de Práticas de Design Instrucional:** Com base na análise dos processos, são implementadas práticas de design instrucional para melhorar a eficácia pedagógica e técnica do projeto. Esta etapa se divide em três subetapas:
  - a. **Planejamento de Objetivos Pedagógicos:** Definição clara dos objetivos pedagógicos que o projeto planeja alcançar.

- b. **Desenvolvimento de Materiais Didáticos:** Criação e preparação de materiais didáticos que apoiarão o aprendizado.
  - c. **Integração com a Tecnologia:** Incorporação de tecnologias relevantes para apoiar o design instrucional e melhorar a experiência de aprendizado.
- 4. **Desenvolvimento e Teste:** Nesta fase, o projeto é desenvolvido e testado. Esta etapa é subdividida em:
  - a. **Desenvolvimento com Novas Tecnologias:** Implementação de tecnologias novas e avançadas no desenvolvimento do projeto.
  - b. **Testes e Ajustes Contínuos:** Realização de testes constantes e ajustes necessários para garantir a qualidade e eficácia do projeto.
- 5. **Métricas de Avaliação:** Avaliação do projeto com base em métricas específicas para medir sua qualidade e desempenho. Esta fase se divide em:
  - a. **Avaliação da Qualidade do Código:** Verificação da qualidade do código desenvolvido para garantir sua robustez e eficiência.
  - b. **Medição de Eficiência e Desempenho:** Avaliação da eficiência operacional e desempenho geral do projeto.
- 6. **Recomendações de Melhoria:** Com base nas métricas de avaliação, são feitas recomendações para melhorar o projeto. Esta fase inclui:
  - a. **Propostas de Otimização Contínua:** Desenvolvimento de propostas para otimização contínua dos processos e do projeto.
  - b. **Aplicação de Feedback e Ajustes:** Implementação de ajustes e melhorias com base no feedback recebido.
- 7. **Publicação e Monitoramento:** A fase final envolve a publicação do projeto e seu monitoramento contínuo. Esta fase é subdividida em:
  - a. **Publicação na Plataforma AVAMEC:** Disponibilização do projeto na plataforma AVAMEC para acesso dos usuários.

- b. **Monitoramento Contínuo e Feedback:** Monitoramento constante do projeto para garantir sua eficácia contínua e realização de ajustes baseados no feedback recebido.

**Figura 24** – Diagrama de funcionamento de processos versão 3.0



Elaboração própria, 2024.

#### 4.2.5 Resultados da Otimização de Processos

A transição para a nova arquitetura tecnológica e a adoção do Scrum resultaram em melhorias significativas nos processos de desenvolvimento de cursos na plataforma AVAMEC. As principais vantagens incluem:

**Redução do Tempo de Desenvolvimento:** A modularização permitiu uma reutilização eficiente de componentes, diminuindo o tempo necessário para desenvolver novos cursos.

**Melhoria na Qualidade do Código:** Com TypeScript, a legibilidade e manutenção do código foram aprimoradas, resultando em menos erros e maior robustez.

**Experiências de Aprendizagem Mais Engajadoras:** A integração de princípios de design instrucional garantiu que os cursos fossem não somente tecnicamente eficientes, mas também pedagogicamente eficazes.

**Feedback Contínuo e Ajustes:** O uso de metodologias ágeis facilitou a adaptação rápida às necessidades dos usuários e melhorias contínuas nos cursos.

#### 4.2.6 Conclusão

A otimização dos processos de desenvolvimento de cursos na plataforma AVAMEC através da integração de tecnologias modernas e práticas avançadas de design instrucional representa um avanço significativo na eficiência e qualidade da educação à distância. A abordagem combinada de engenharia de software e pedagogia tem potencial para transformar a plataforma em um ambiente de aprendizagem robusto e eficaz, atendendo às crescentes demandas por educação de qualidade no Brasil.

## 5. Resultados

### **Descrição do Projeto**

#### **Objetivo 1 - Análise dos Processos Atuais de Desenvolvimento de Cursos**

O projeto iniciou com uma análise detalhada dos processos existentes para o desenvolvimento de cursos na plataforma AVAMEC. Essa etapa envolveu a documentação minuciosa dos fluxos de trabalho, a identificação de gargalos e ineficiências, além do levantamento de dados sobre o tempo de desenvolvimento e a qualidade dos cursos. Técnicas como a análise de arquitetura de software, revisão de código e aplicação de métricas de desempenho foram utilizadas para mapear o estado atual dos processos, proporcionando uma visão clara das áreas que necessitavam de melhorias.

#### **Objetivo 2 - Desenvolvimento e Aplicação de Novas Tecnologias**

Após a análise dos processos atuais, foi implementada uma nova arquitetura tecnológica, adotando React, TypeScript e Tailwind CSS. Esta mudança teve como objetivo modularizar o desenvolvimento dos cursos, melhorando a reusabilidade de componentes e facilitando a manutenção. Durante essa fase, protótipos foram criados e testes de desempenho realizados para assegurar que a nova arquitetura atendesse às expectativas de eficiência e escalabilidade. A implementação dessas tecnologias foi acompanhada de uma integração cuidadosa com os princípios de design instrucional, garantindo que as melhorias técnicas fossem acompanhadas de um impacto positivo na qualidade pedagógica dos cursos.

#### **Objetivo 3 - Avaliação dos Impactos das Mudanças**

A avaliação dos impactos das mudanças implementadas será realizada na fase subsequente do projeto, e será abordada detalhadamente na atual seção de resultados. Esta avaliação busca medir a eficácia das novas práticas e tecnologias adotadas, utilizando métricas quantitativas.

### **5.1 Análise dos Dados de Desenvolvimento**

A implementação das novas tecnologias (React, TypeScript e Tailwind CSS) e a aplicação de práticas avançadas de design instrucional resultaram em mudanças significativas nos processos de desenvolvimento dos cursos digitais para a plataforma AVAMEC. A seguir, são apresentados os dados quantitativos coletados para avaliar a eficiência e a qualidade dos cursos desenvolvidos antes e após a otimização dos processos.

**Tabela 1:** Tabela de Dados de Desenvolvimento (dados coletados em julho de 2025)

Curso	Commits	Data de Início	Data de Fim	Duração do tempo de desenvolvimento (Dias)	Tecnologia (nova ou antiga)
PCR	1218	18/07/2022	13/12/2023	514	Antiga
DPE	123	28/08/2023	25/01/2024	150	Nova
ABP	16	16/03/2023	10/04/2023	24	Nova
FPN	174	12/04/2023	06/06/2023	55	Nova
PLAEB	118	26/01/2024	23/05/2024	118	Nova

**Tabela 2:** Tabela de Quantidade de Alunos nos Cursos Desenvolvido que Mostram o Impacto das Abordagem Adotadas (dados coletados em julho de 2025)

Curso	Quantidade de Alunos
PCR	Curso ainda não lançado
DPE	9.814



ABP	84.370
FPN	257.078
PLAEB	69.088

- PCR: Plano de Carreira e Remuneração
- DPE: Desenvolvimento do Protagonismo do Estudante
- ABP: Aprendizagem Baseada em Problemas e Projetos
- FPN: Formação de Professores em Neuroeducação
- PLAEB: Português como Língua de Acolhimento na Educação Básica

## 5.2 Redução do Tempo de Desenvolvimento

A transição para a nova arquitetura tecnológica permitiu uma redução significativa no tempo de desenvolvimento dos cursos. A modularização do código e a reutilização de componentes resultaram em uma diminuição do esforço necessário para criar e manter os cursos.

**PCR:** O curso PCR, que utiliza tecnologias primitivas como HTML, CSS e JavaScript, teve 1218 commits e duração de 513 dias. Este alto número de commits indica que houve uma necessidade frequente de ajustes e correções, refletindo as limitações e desafios das tecnologias utilizadas inicialmente. A falta de modularização resultou em um retrabalho constante e em um maior tempo de desenvolvimento.

**TECDI:** Com 96 commits e duração de 420 dias, o tempo de desenvolvimento foi otimizado pela nova stack tecnológica, resultando em um fluxo de trabalho mais ágil.

**ABP:** Este curso teve somente 16 commits em 385 dias, destacando a necessidade de processos mais dinâmicos e componentes reutilizáveis, implementados posteriormente.

**SAC e PLAEB:** Os cursos mais recentes, com 103 e 118 commits respectivamente, mostraram melhorias contínuas no processo de desenvolvimento, refletindo a maturidade das práticas adotadas.

### **5.3 Aumento da Qualidade do Código**

A utilização de TypeScript pode aumentar a legibilidade e a manutenção do código, resultando em menos erros e maior robustez. A introdução de testes unitários, testes manuais e a aplicação de design instrucional também pode vir a permitir que os cursos venham a ser pedagogicamente eficazes.

### **5.4 Experiências de Aprendizagem Mais Engajadoras**

A integração dos princípios de design instrucional com a nova tecnologia também podem vir a permitir criação de cursos mais interativos e engajantes. Os cursos passaram a incluir quizzes interativos, feedbacks personalizados e atividades práticas que venham a melhorar a experiência de aprendizagem dos alunos.

### **5.5 Impacto na Quantidade de Alunos**

A melhoria na qualidade e na interatividade dos cursos resultou em um aumento no número de alunos matriculados, conforme dados apresentados na tabela acima. Cursos que vão do TECDI ao PLAEB, que tiveram maior foco em interatividade e modularização, mostraram um aumento significativo no número de alunos.

### **5.6 Análise de Commits**

No contexto do desenvolvimento de software, "commits" são registros de mudanças feitas no código-fonte de um projeto. Cada commit representa uma versão do código em um ponto específico do tempo, incluindo adições, modificações e correções. A quantidade de commits pode ser uma estimativa de quantas vezes um curso precisou ser continuamente ajustado e atualizado.

Observando os dados, o curso PCR, que utiliza tecnologias primitivas como HTML, CSS e JavaScript, foi o que levou mais tempo e teve a maior quantidade de commits, totalizando 1218 commits em 513 dias. Isso indica que houve uma necessidade frequente de ajustes e correções, refletindo as limitações e desafios das tecnologias utilizadas inicialmente. Em contraste, cursos desenvolvidos com a nova stack tecnológica (React, TypeScript e Tailwind CSS) apresentaram menos commits e uma maior eficiência no desenvolvimento, evidenciando os benefícios da modernização tecnológica.

### **Conclusão**

A integração de novas tecnologias com práticas avançadas de design instrucional aponta para um caminho promissor, capaz de gerar processos de desenvolvimento mais eficientes e cursos de maior qualidade na plataforma AVAMEC. Nesse modelo, a modularização e a reutilização de componentes, quando aliadas a uma abordagem pedagógica robusta, têm o potencial de viabilizar a criação de cursos que não apenas atendam às necessidades técnicas, mas que também possam oferecer uma experiência de aprendizagem mais rica e envolvente aos alunos.

## 6. Considerações Finais e Limitações

Neste trabalho, analisamos sobre como melhorar o desenvolvimento de cursos digitais na plataforma AVAMEC. Combinamos técnicas modernas de programação com métodos eficazes de design instrucional para criar cursos melhores e mais rápidos.

Trocamos as tecnologias menos eficazes por outras mais eficazes, como React, TypeScript e Tailwind CSS. Sendo necessário não somente adotar essas tecnologias, como desenvolver um boilerplate completo e documentado para React que se adequa completamente aos nossos desafios e ao AVAMEC. Isso nos ajudou a criar códigos mais organizados e fáceis de manter. Além disso, adotamos uma forma de trabalho mais avançada, usando o Scrum integrada a práticas de design instrucional, o que otimizou completamente como o software feito pela equipe era entregue, tanto em termos de tempo quanto em termos de qualidade e eficiência.

Para garantir que os cursos fossem realmente bons para os alunos, usamos um método chamado ADDIE. Ele nos ajudou a alinhar o que queríamos ensinar com a melhor forma de usar a tecnologia para isso. Como resultado direto, contribuímos para obter cursos mais interessantes e que prendem mais a atenção dos alunos que somado com a eficiência do time de desenvolvimento em construir códigos de alta qualidade, em velocidades muitas mais rápidas que antes e com menor incidência de erros teve, como principal resultado, alcançar milhares de alunos, com maior qualidade e em menor tempo.

### 6.1 Contribuições do Estudo

A principal contribuição deste estudo reside na demonstração prática de como a integração de novas tecnologias e práticas pedagógicas pode transformar o desenvolvimento de cursos digitais. As seguintes melhorias foram observadas:

**Redução do Tempo de Desenvolvimento:** A modularização do código e a reutilização de componentes alinhadas com a otimização de processos reduziram significativamente o tempo necessário para o desenvolvimento de novos cursos.

**Melhoria na Qualidade do Código:** A tipagem estática proporcionada por TypeScript, a clareza e facilidade de aplicação de estilos do Tailwind CSS e a estrutura modular de React melhoram a legibilidade e manutenção do código, resultando em menos erros e maior robustez.

**Experiências de Aprendizagem Mais Engajadoras:** A integração de princípios de design instrucional tem o potencial de deixar os cursos pedagogicamente eficazes e interativos, melhorando a experiência de aprendizagem dos alunos.

**Aumento no Número de Alunos:** A melhoria na qualidade e interatividade dos cursos resultou em um aumento significativo no número de alunos matriculados, refletindo a eficácia das mudanças implementadas.

## **6.2 Limitações do Estudo**

Apesar de todo o progresso, algumas limitações ainda persistem e devem ser consideradas para trabalhos futuros:

**Curva de Aprendizado:** A equipe precisou de um tempo para se adaptar com as novas tecnologias, o que deixou o trabalho um pouco mais lento no começo.

**Complexidade na Integração:** A integração das novas tecnologias com os sistemas existentes apresentou desafios, principalmente quando tentamos usar os cursos antigos no novo sistema.

**Demanda de Recursos:** Implementar todas essas mudanças exigiu tempo e, dependendo do caso, pode exigir dinheiro, especialmente para treinar a equipe. Pode ser que nem todas as instituições possam ter esses recursos disponíveis.

**Feedback Contínuo:** Mesmo com nossa nova forma de trabalhar, ainda é difícil coletar e usar o feedback dos alunos de forma eficiente, sendo necessário para tal um suporte do AVAMEC para que forneça os dados.

## **6.3 Sugestões para Trabalhos Futuros**

Para futuras pesquisas e desenvolvimentos, as seguintes sugestões são propostas:

**Avaliação de Longo Prazo:** Fazer estudos que acompanhem por mais tempo como essas mudanças afetam a qualidade dos cursos e a satisfação dos alunos.

**Expansão da Adoção:** Pesquisar como outras instituições e universidades podem adotar essas melhorias, mesmo possivelmente tendo recursos limitados.

**Otimização Contínua:** Está em consonância com novas tecnologias que possam tornar o desenvolvimento de cursos ainda melhor.

**Análise de Custo-Benefício:** Fazer um estudo detalhado para ver se é possível financeiramente fazer essas mudanças em diferentes tipos de instituições.

**Avaliação qualitativa:** Para além de resultados quantitativos, verificar e analisar se a arquitetura mais moderna implementada foi capaz de produzir resultados qualitativos relevantes como a qualidade do código e qualidade pedagógica dos cursos produzidos.

Para finalizar, vimos que unir boas práticas de programação com métodos eficazes de ensino pode realmente melhorar como criamos cursos online. Conseguimos fazer cursos melhores e mais rápido. Mas é importante lembrar que sempre podemos melhorar. Precisamos continuar experimentando e ajustando nossas práticas para acompanhar as mudanças no mundo da educação à distância.

## 7. Referências

- [1] Clinton, D. (2023). *Understanding HTML Page Structure*. In: *HTML Page Elements – Explained for Beginners*. FreeCodeCamp. Disponível em: <https://www.freecodecamp.org>.
- [2] Youngblood, S. A. (2013). Communicating Web Accessibility to the Novice Developer: From User Experience to Application. *Journal of Business and Technical Communication*, 27(2), 209-232. <https://doi.org/10.1177/1050651912458924>
- [3] Acosta-Medina, J. K., Torres-Barreto, M. L., & Cárdenas-Parga, A. F. (2021). Students' preference for the use of gamification in virtual learning environments. *Australasian Journal of Educational Technology*, 37(4), 145-158. <https://doi.org/10.14742/ajet.6512>
- [4] Anderson, K. M. (1999). Supporting Industrial Hyperwebs: Lessons in Scalability. In *Proceedings of the International Conference on Software Engineering (ICSE '99)*. Los Angeles, CA. Association for Computing Machinery. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/302405.302696>
- [5] Karthik, B. S. S., Chandrasekhar, B., David, R., & Kumar, A. K. (2019). Identification of Instructional Design Strategies for an Effective E-learning Experience. *The Qualitative Report*, 24(7), 1537-1555. <https://doi.org/10.46743/2160-3715/2019.3870>
- [6] Sevenscan, C. (2024). *Optimizing Web Delivery: The Impact Of Rendering Methods On User Experience Across Network Conditions*. MS thesis, California Polytechnic State University, San Luis Obispo. Available at: <https://digitalcommons.calpoly.edu/theses/2831>
- [7] Borello, D. (2024). *Micro Frontends, Server Components and how these technologies can provide a paradigm shift with architectural changes in modern enterprise web app development*. Master's Thesis, Politecnico di Torino, Corso di laurea magistrale in Ingegneria Informatica (Computer Engineering). Available at: <http://webthesis.biblio.polito.it/id/eprint/31061>
- [8] Aggarwal, S. (2018). Modern Web-Development using ReactJS. *International Journal of Recent Research Aspects*, 5(1), 133-137. Available at: <https://ijrra.net/Vol5issue1/IJRRRA-05-01-27.pdf>
- [9] Cocca, G. (2022). *How to Use TypeScript – Beginner-Friendly TS Tutorial*. Retrieved from <https://www.freecodecamp.org/news/an-introduction-to-typescript/#whatsthedealwithtypesandjavascript>
- [10] Abba, I. (2022). *How to Use Tailwind CSS to Rapidly Develop Snazzy Websites*. Retrieved from <https://kinsta.com/blog/tailwind-css/>

[11] Ahmad, N., Amer, N. T., Qutaifan, F., & Alhilali, A. (2013). Technology adoption model and a road map to successful implementation of ITIL. Journal of Enterprise Information Management. <https://doi.org/10.1108/JEIM-07-2013-0041>

[12] Muthucumaru, A. (2022). The future of collaborative technology within Scrum/Agile practices. Faculty of Information, University of Toronto. <https://doi.org/10.33137/ijournal.v7i1.37897>

[13] Isman, A. (2011). Instructional Design in Education: New Model. Turkish Online Journal of Educational Technology - TOJET, 10(1), 136-142. Available at: <https://eric.ed.gov/?id=EJ926562>

[14] Hema, V., Thota, S., Kumar, S. N., Padmaja, C., Rama Krishna, C. B., & Mahender, K. (2020). Scrum: An Effective Software Development Agile Tool. IOP Conference Series: Materials Science and Engineering, 981, Article 022060. <https://doi.org/10.1088/1757-899X/981/2/022060>

[15] Spatioti, A. G., Kazanidis, I., & Pange, J. (2022). A Comparative Study of the ADDIE Instructional Design Model in Distance Education. Information, 13(9), 402. <https://doi.org/10.3390/info13090402>

[16] Componente Linha do Tempo. V-LAB UFPE, Recife, 2024. Disponível em: <https://avamec.mec.gov.br>. Acesso em: 08 jun. 2024.

[17] Módulo Avaliativo do curso Plano de Carreira e Remuneração (PCR). V-LAB UFPE, Recife, 2024. Disponível em: <https://avamec.mec.gov.br>. Acesso em: 17 jun. 2024

[18] ReactJS Tutorial. tutorialspoint, 2024. Disponível em: [https://www.tutorialspoint.com/reactjs/images/workflow\\_jsx.jpg](https://www.tutorialspoint.com/reactjs/images/workflow_jsx.jpg). Acesso em: 28 jun. 2024

[19] What is Typescript?. Medium, 2018. Disponível em: [https://medium.com/@vishnupriya\\_web/what-is-typescript-faa0890b2baf](https://medium.com/@vishnupriya_web/what-is-typescript-faa0890b2baf). Acesso em: 30/09/2024

[20] Componente feito com arquitetura avançada. V-LAB UFPE, Recife, 2024. Disponível em: <https://avamec.mec.gov.br>. Acesso em: 08 jun. 2024

[21] SCRUM: entenda a metodologia utilizada na Gestão de Projetos. Engenharia360, 2020. Disponível em: <https://engenharia360.com/scrum-a-metodologia-utilizada-na-gestao-de-projetos/>. Acesso em: 02/09/2024

[22] ADDIE: 5 Steps To Effective Training. Learnupon, 2024. Disponível em: <https://www.learnupon.com/blog/addie-5-steps/>. Acesso em: 12/09/2024

[23] FROST, B. *Atomic Design*. Nova York: Brad Frost LLC, 2016.



[24] CLARK, R.; MAYER, R. *E-Learning and the Science of Instruction*. Hoboken: Wiley, 2016.

[25] ANDERSON, L. et al. *A Taxonomy for Learning, Teaching, and Assessing*. Nova York: Longman, 2001.

[26] GUSTAFSON, K.; BRANCH, R. *Survey of Instructional Development Models*. Syracuse University, 2002

[27] [1] Heinemann, B., Ehlenz, M., Görzen, S., & Schroeder, U. (2022). *xAPI Made Easy: A Learning Analytics Infrastructure for Interdisciplinary Projects*. International Journal of Online and Biomedical Engineering (iJOE), 18(14), 99–113. <https://doi.org/10.3991/ijoe.v18i14.35079>

[28] Lin, C. C., Huang, A. Y. Q., & Lu, O. H. T. (2023). *Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review*. Smart Learning Environments, 10(41). <https://doi.org/10.1186/s40561-023-00260-y>

[29] Guo, X., Guo, Y., & Liu, Y. (2021). *The Development of Extended Reality in Education: Inspiration from the Research Literature*. Sustainability, 13(24), 13776. <https://doi.org/10.3390/su132413776>

[30] Chamorro-Atalaya, O., Flores-Velásquez, C. H., Olivares-Zegarra, S., Dávila-Ignacio, C., Flores-Cáceres, R., Arévalo-Tuesta, J. A., Cruz-Telada, Y., & Suarez-Bazalar, R. (2024). *Microlearning and Nanolearning in Higher Education: A Bibliometric Review to Identify Thematic Prevalence in the COVID-19 Pandemic and Post-Pandemic Context*. International Journal of Learning, Teaching and Educational Research, 23(4), 15–34. <https://doi.org/10.26803/ijlter.23.4.15>

[31] Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). *Gamification in Education: A Systematic Mapping Study*. Educational Technology & Society, 18(3), 75–88. <https://www.jstor.org/stable/jeductechsoci.18.3.75>

[32] Almeqdad, Q. I., Alodat, A. M., Alquraan, M. F., Mohaidat, M. A., & Al-Makhzoomy, A. K. (2023). *The effectiveness of universal design for learning: A systematic review of the literature and meta-analysis*. Cogent Education, 10(1), Article 2218191. <https://doi.org/10.1080/2331186X.2023.2218191>

[33] Ifenthaler, D., & Widanapathirana, C. (2014). *Development and Validation of a Learning Analytics Framework: Two Case Studies Using Support Vector Machines*. Technology, Knowledge and Learning, 19(1–2), 221–240. <https://doi.org/10.1007/s10758-014-9226-4>

[34] V-Lab UFPE. (s.d.). *V-Lab do CIn anuncia bolsa para desenvolvimento estudantil*. UFPE. Recuperado de

[https://www.ufpe.br/ascom/noticias/-/asset\\_publisher/O3Odar12gQTr/content/v-lab-do-cin-a-nuncia-bolsa-para-desenvolvimento-estudantil/40615](https://www.ufpe.br/ascom/noticias/-/asset_publisher/O3Odar12gQTr/content/v-lab-do-cin-a-nuncia-bolsa-para-desenvolvimento-estudantil/40615)

[35] Ministério da Educação (MEC). (s.d.). *Portal do MEC, agora, é gov.br/mec*. Recuperado de <https://portal.mec.gov.br/>

[36] Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). (s.d.). *CAPES*. Gov.br. Recuperado de <https://www.gov.br/capes/pt-br>

[37] Conselho Nacional de Justiça (CNJ). (s.d.). *Página Inicial*. Portal CNJ. Recuperado de <https://www.cnj.jus.br/>

[39] Mozilla Developer Network (MDN). (s.d.). *CSS*. MDN Web Docs. Recuperado de <https://developer.mozilla.org/pt-BR/docs/Web/CSS>

[40] Mozilla Developer Network (MDN). (s.d.). *JavaScript*. MDN Web Docs. Recuperado de <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

[41] Meta. (s.d.). *React*. Recuperado de <https://react.dev/>

[43] Microsoft. (s.d.). *TypeScript*. Recuperado de <https://www.typescriptlang.org/>

[44] Tailwind Labs. (s.d.). *Tailwind CSS*. Recuperado de <https://tailwindcss.com/>

[45] Kurt, S. (2017). Instructional Design Models and Theories. *Educational Technology*. Available at: <https://educationaltechnology.net/instructional-design-models-and-theories/>

[46] Al-Shalchi, O. N. (2021). A Review of Instructional Design Models. *International Journal of Computer-Assisted Language Learning and Teaching (IJCALLT)*, 11(4), 105-127. Available at: [https://www.researchgate.net/publication/367590207\\_A\\_Review\\_of\\_the\\_Computer-Assisted\\_Language\\_Learning\\_CALL\\_on\\_Teaching\\_and\\_Learning\\_Writing](https://www.researchgate.net/publication/367590207_A_Review_of_the_Computer-Assisted_Language_Learning_CALL_on_Teaching_and_Learning_Writing)

[47] ADEOYE, M. A. et al. Revolutionizing Education: Unleashing the Power of the ADDIE Model for Effective Teaching and Learning. *JPI (Jurnal Pendidikan Indonesia)*, v. 13, n. 1, p. 202-209, 2024. Disponível em: [https://www.researchgate.net/publication/381733325\\_Revolutionizing\\_Education\\_Unleashing\\_the\\_Power\\_of\\_the\\_ADDIE\\_Model\\_for\\_Effective\\_Teaching\\_and\\_Learning](https://www.researchgate.net/publication/381733325_Revolutionizing_Education_Unleashing_the_Power_of_the_ADDIE_Model_for_Effective_Teaching_and_Learning).

[48] Ouhbi, S., Idri, A., Fernández-Alemán, J. L., & Toval, A. (2015). Requirements engineering education: a systematic mapping study. *Requirements Engineering*, 20(2), 119-138. Available at: <https://link.springer.com/article/10.1007/s00766-013-0192-5>

[49] Serrador, P., & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5),

1040-1051. Available at:

<https://www.sciencedirect.com/science/article/pii/S0263786315000071>

[50] Pacheco Sánchez, G., Escudero-Nahón, A., Ibarra Corona, M. A., & Muñoz Mandujano, M. (2025). Bridging pedagogy and technology: A conceptual analysis for software design in educational platforms. *Edelweiss Applied Science and Technology*, 9(6), 1293–1306. <https://doi.org/10.55214/25768484.v9i6.8102>

## 8. Apêndices

## **8.1 Apêndice A. Tecnologia da Arquitetura Anterior**

A arquitetura utilizada anteriormente no desenvolvimento dos cursos para a plataforma AVAMEC baseava-se em tecnologias tradicionais da web, como HTML, CSS e JavaScript. Essas ferramentas, amplamente conhecidas, permitiram construir interfaces funcionais e oferecer interatividade básica nos cursos. No entanto, com o crescimento da complexidade dos projetos, surgiram desafios. A estrutura do código era extensa e pouco modular, o que tornava a manutenção trabalhosa e a implementação de novas funcionalidades um processo lento. Embora fosse uma solução robusta para projetos menores, a arquitetura anterior se mostrou limitada quando aplicada a cursos de maior escala e exigências mais dinâmicas.

## **8.2 Apêndice B. Tecnologia da Arquitetura Nova**

A nova arquitetura trouxe um avanço significativo ao adotar tecnologias modernas, como React, TypeScript e Tailwind CSS. Com essas ferramentas, o desenvolvimento se tornou mais ágil e organizado. O React permitiu uma estrutura modular e reutilizável, facilitando a criação de componentes interativos que melhoraram a experiência dos alunos. TypeScript, por sua vez, adicionou segurança ao código, reduzindo erros e proporcionando uma colaboração mais eficiente entre desenvolvedores. Já o Tailwind CSS simplificou a estilização, garantindo uma interface visual consistente e responsiva. Essa combinação permitiu à equipe desenvolver cursos com mais rapidez, mantendo a flexibilidade e a qualidade necessárias para acompanhar a evolução das demandas pedagógicas.