

Relatório Técnico Final — EduStream

Resumo Executivo

Contexto da Equipe: O desenvolvimento foi realizado por uma **equipe de 6 integrantes**, permitindo divisão de responsabilidades entre arquitetura, desenvolvimento de componentes, prototipação visual, testes e documentação técnica. O projeto **EduStream** foi desenvolvido como uma plataforma educacional digital voltada à disponibilização de conteúdos interativos e organização de trilhas de aprendizagem, com foco em experiência do usuário, escalabilidade e modularidade de componentes. O principal objetivo foi construir uma interface moderna, responsiva e de alta performance utilizando tecnologias contemporâneas do ecossistema web.

A solução final consiste em uma aplicação **frontend independente**, implementada com **React + TypeScript + Vite**, estruturada por páginas funcionais e componentes reutilizáveis, possibilitando futura integração com APIs externas. O projeto priorizou arquitetura limpa, separação de responsabilidades e padronização visual para garantir evolução contínua sem comprometer estabilidade.

O resultado obtido demonstra que os objetivos essenciais de arquitetura, modularidade e experiência de uso foram atingidos com sucesso, fornecendo uma base sólida para funcionalidades futuras como aulas ao vivo, integração com backend, gamificação e monitoramento de progresso acadêmico.

Objetivos Alcançados vs Planejados

Objetivo Planejado	Resultado Alcançado	Status
Criar interface responsiva e moderna para plataforma educacional	Interface completa e responsiva implementada com Tailwind CSS e componentes reutilizáveis	Concluído o
Implementar navegação entre páginas principais	Roteamento implementado com React Router DOM e estrutura completa de páginas	Concluído o
Definir arquitetura escalável e modular para evolução futura	Organização em camadas de componentes, hooks e contextos concluída	Concluído o
Integração com API real para funcionalidades dinâmicas	Não implementada nesta versão (dependente de backend)	Planejado para próxima etapa

Objetivo Planejado	Resultado Alcançado	Status
Sistema de notificações e feedback ao usuário	Implementado com Sonner e Toaster	Concluído

Essa comparação evidencia que o escopo central foi cumprido conforme planejado, e a ausência de API não comprometeu os objetivos arquiteturais do protótipo.

—————|—————|—————|—————|—————|—————|—————|—————|

(Descrever o que foi cumprido, parcialmente cumprido ou não cumprido e por quê.)

Arquitetura Final Implementada

A arquitetura implementada no projeto **EduStream** foi concebida seguindo princípios de modularidade, escalabilidade e manutenção simplificada. A aplicação foi desenvolvida como um **frontend independente**, baseado inteiramente em **React com TypeScript**, estruturado em componentes reutilizáveis e organizado em camadas lógicas para facilitar evolução e integração futura com APIs.

Do ponto de vista arquitetural, a solução adotada busca separar apresentação, lógica de interação e gestão de estado, favorecendo clareza e facilidade de manutenção. A aplicação segue a estrutura típica de projetos modernos com Vite, o que proporciona alto desempenho durante o desenvolvimento, tempo de compilação reduzido e modularização eficiente.

Tecnologias Principais Utilizadas

- **React + TypeScript** — base da interface e componentes tipados garantindo maior segurança e qualidade do código.
- **Vite** — ferramenta de build moderna com hot reload otimizado.
- **React Router DOM** — gerenciamento de navegação entre páginas.
- **Context API** — controle de estados globais simples, como elementos de UI compartilhados.
- **React Query (TanStack)** — gerenciamento de dados assíncronos, preparado para futura integração com APIs reais.
- **TailwindCSS** — biblioteca utilitária de estilização, garantindo consistência visual e produtividade.
- **Sonner / Toaster** — sistema de notificações e feedback ao usuário.

Organização Geral de Pastas

```
src/
  components/  -> componentes reutilizáveis
  pages/       -> páginas principais da aplicação
  context/     -> definição e controle de estado global
  hooks/       -> lógica reutilizável isolada
```

assets/	-> imagens e arquivos estáticos
App.tsx	-> definição das rotas da aplicação
main.tsx	-> ponto de entrada e configuração geral

Considerações Arquiteturais

A arquitetura atual não inclui backend integrado e opera de forma totalmente client-side. Entretanto, foi planejada para suportar escalabilidade e integração futura por meio de camada de serviços a ser incorporada com endpoints REST ou GraphQL, utilizando React Query para caching, sincronização e políticas de revalidação.

A escolha por arquitetura leve e orientada a componentes permitiu validação rápida do conceito e criação de uma base sólida para futuras extensões, como autenticação, streaming de aulas ao vivo e armazenamento de progresso.

Desafios Técnicos Enfrentados e Soluções Adotadas

Durante o desenvolvimento do projeto **EduStream**, um dos principais desafios enfrentados esteve relacionado à **gestão de tempo e divisão de responsabilidades dentro da equipe**, especialmente considerando a quantidade de componentes, páginas e decisões arquiteturais envolvidas.

Desafio: Gestão de Tempo e Coordenação entre Integrantes

A necessidade de coordenar atividades simultâneas, alinhar decisões técnicas e manter consistência de código entre os membros exigiu planejamento contínuo e comunicação eficiente. A ausência de um backend disponível na primeira etapa também demandou reorganização do cronograma, priorizando inicialmente a construção do frontend e a definição da arquitetura base.

Solução Adotada

Para superar esse desafio, foram estabelecidas práticas de organização colaborativa, incluindo: - Planejamento e distribuição clara de tarefas para cada membro da equipe - Utilização de padronização visual e estrutural para garantir uniformidade na interface - Reuniões periódicas de alinhamento para monitorar avanços e identificar impedimentos com agilidade - Divisão modular do projeto para permitir trabalho paralelo entre desenvolvedores

Essas estratégias permitiram mitigar riscos de atraso e garantir que o produto final fosse entregue de maneira consistente, organizada e tecnicamente estruturada.

Métricas de Qualidade

Embora o projeto não tenha incorporado uma infraestrutura de monitoramento com métricas automatizadas, foram adotados critérios qualitativos e práticas recomendadas para avaliar desempenho, organização e manutenção do sistema ao longo do desenvolvimento.

Avaliação de Performance e Eficiência

- **Tempo de carregamento reduzido**, proporcionado pelo uso do Vite, que otimiza o processo de compilação e entrega inicial.
- **Renderização eficiente** devido à arquitetura modular de componentes e uso adequado de estados locais.
- **Baixa complexidade de re-renderização**, uma vez que a maioria das telas utiliza gerenciamento de estado leve e bem distribuído.

Qualidade de Código e Manutenibilidade

- **Tipagem estática com TypeScript**, reduzindo erros em tempo de execução e aumentando confiabilidade.
- **Separação de responsabilidades**, com uso de hooks e contextos independentes para evitar duplicação e melhorar reutilização.
- **Padronização visual e de estilos** com TailwindCSS, colaborando para consistência estética e aceleração de ajustes futuros.

Usabilidade e Experiência do Usuário

- **Interface responsiva**, com adaptação a diferentes tamanhos de tela.
- **Sistema de notificações** garantindo retorno rápido ao usuário sobre ações realizadas.

Tais métricas qualitativas demonstram que a solução evoluiu com foco na entrega funcional, na estruturação de base tecnológica sólida e na capacidade de expansão futura.

Trabalhos Futuros e Melhorias Possíveis

Considerando o escopo entregue e o potencial evolutivo da plataforma EduStream, destaca-se um conjunto de melhorias planejadas para versões futuras:

- Desenvolvimento e integração de um **backend completo**, incluindo autenticação, banco de dados e gerenciamento de usuários.
- **Implementação de testes automatizados** (unitários e de integração) para aumentar a confiabilidade do sistema.
- **Integração com CI/CD**, permitindo build automático, verificação de qualidade e implantação contínua.

- **Ampliação de métricas de monitoramento**, com ferramentas como Lighthouse, Sentry ou LogRocket para mensuração de desempenho real.
- **Suporte a transmissões ao vivo e streaming de conteúdo multimídia**, expandindo recursos educacionais.
- **Gamificação e trilhas personalizadas**, com acompanhamento detalhado de progresso.
- **Internacionalização (i18n)** para suporte multilíngue.

Estas ações reforçam a continuidade e a escalabilidade da plataforma, fortalecendo seu fundamento arquitetural e acadêmico.
