**Documentação técnica da Plataforma Educacional Online**

**Problema Identificado**

O ensino público enfrenta desafios significativos relacionados à motivação e à eficácia do aprendizado. Muitos alunos estão restritos a conteúdos exclusivamente baseados em livros, o que pode ser insuficiente para engajá-los e garantir uma compreensão mais profunda. A falta de materiais didáticos interativos, como vídeos, quizzes, enquetes, artigos complementares e outros materiais, contribui para o desinteresse nas aulas e dificulta a assimilação do conteúdo.

**Solução Proposta**

Criamos uma plataforma educacional online voltada para professores e alunos do ensino público, com os seguintes recursos:

* **Publicação de conteúdo multimídia**: Professores podem publicar vídeos explicativos, artigos complementares, gráficos e outros materiais interativos relacionados aos temas das aulas.
* **Aprendizado Imersivo**: Materiais visuais, como vídeos e gráficos, ajudam os alunos a compreenderem conteúdos abstratos ou complexos, complementando o aprendizado tradicional baseado em livros e aulas presenciais.
* **Acessibilidade**: A plataforma é projetada para ser acessível por dispositivos móveis e computadores, visando atender alunos com acesso limitado a recursos tecnológicos.
* **Sistema de Pontuação**: Alunos que assistirem aos vídeos por completo, participarem de debates e deixarem comentários receberão pontuações que poderão ser utilizadas como nota complementar em suas médias escolares.

**Melhorias Futuras**

A publicação de conteúdo será organizada por disciplinas, séries e temas, facilitando o acesso e a navegação.

Alunos poderão deixar comentários, enviar dúvidas e interagir diretamente com os professores sobre os conteúdos publicados. Discussões e debates nos comentários incentivam o aprendizado colaborativo.

A plataforma será projetada para ser acessível por dispositivos móveis visando atender alunos com acesso limitado a recursos tecnológicos.

Alunos que assistirem aos vídeos por completo, participarem de debates e deixarem comentários receberão pontuações que poderão ser utilizadas como nota complementar em suas médias escolares.

Recursos interativos como quizzes, enquetes serão adicionados para reforçar o aprendizado.

**Impacto Esperado**

* **Aumento do Engajamento**: O formato dinâmico e interativo do conteúdo torna o aprendizado mais atrativo.
* **Melhora no Desempenho Escolar**: A complementação das aulas com materiais multimídia reforça a compreensão dos temas.
* **Redução na Taxa de Reprovação**: O sistema de pontuação permite que alunos utilizem as notas adicionais para melhorar seus desempenhos nas avaliações.

**Processo de Desenvolvimento**

O processo de desenvolvimento foi estruturado para garantir eficiência e alinhamento com os objetivos do projeto:

1. **Planejamento Inicial**:
   * Realização de reuniões para identificar problemas do ensino público, como evasão escolar, desinteresse pelo conteúdo ministrado e baixo desempenho acadêmico.
   * Consideração do tempo de desenvolvimento e disponibilidade da equipe.
2. **Estruturação do Projeto**:
   * Reuniões semanais para acompanhamento, validação de entregas e resolução de dúvidas.
   * Divisão de tarefas entre backend, frontend, documentação e pitch.
   * Entregas semanais para garantir o progresso contínuo.

**Detalhes Técnicos**

**Backend**

* **Arquitetura**: Model-View-Controller (MVC)
* **Ferramentas e bibliotecas utilizadas**:
  + **Node.js**: Ambiente de execução JavaScript server-side.
  + **Express**: Framework para APIs RESTful.
  + **Sequelize**: ORM para interagir com o banco de dados PostgreSQL.
  + **Swagger**: Documentação interativa de APIs.
  + **JsonWebToken**: Verificação de tokens JWT.
  + **Docker**: Gerenciamento de contêineres.
  + **Banco de Dados**: PostgreSQL.
* **Estrutura do código**

📦backend

┣ 📂bin

┃ ┗ 📜www # Arquivo de inicialização do servidor

┣ 📂controller

┃ ┣ 📜autenticacaoController.js # Controlador para autenticação

┃ ┣ 📜conteudoController.js # Controlador para conteúdos

┃ ┗ 📜usuarioController.js # Controlador para usuários

┣ 📂database

┃ ┗ 📜database.js # Configuração do banco de dados

┣ 📂middlewares

┃ ┣ 📜autorizarPermissoes.js # Middleware para autorização de permissões

┃ ┗ 📜verificarToken.js # Middleware para verificação de token

┣ 📂model

┃ ┣ 📜Conteudo.js # Modelo de dados para conteúdos

┃ ┗ 📜Usuario.js # Modelo de dados para usuários

┣ 📂routes

┃ ┣ 📜autenticacao.js # Rotas de autenticação

┃ ┣ 📜conteudos.js # Rotas de conteúdos

┃ ┗ 📜usuarios.js # Rotas de usuários

┣ 📂Uploads # Diretório para armazenar arquivos enviados pela aplicação.

┃ ┣ 📜gitkeep.js # Permite que diretórios vazios sejam incluídos no repositório Git.

┣ 📜.env #Arquivo de variáveis de ambiente locais.

┣ 📜.env.docker # Arquivo de variáveis de ambiente para Docker

┣ 📜.gitignore # Arquivos e diretórios ignorados pelo Git

┣ 📜app.js # Configuração do aplicativo Express

┣ 📜docker-compose.yml # Configuração do Docker Compose

┣ 📜Dockerfile # Dockerfile para criar a imagem do Docker

┣ 📜Dockerfile.db # Dockerfile para o banco de dados

┣ 📜FIAP-Hackathon.postman\_collection.json # Coleção do Postman para testes de API

┣ 📜FIAP\_BIBLIOTECA\_ONLINE.sql # Script SQL para criação do banco de dados

┣ 📜package-lock.json # Arquivo de lock do npm

┣ 📜package.json # Dependências e scripts do npm

┣ 📜README.md # Documentação do projeto

┗ 📜swaggerConfig.js # Configuração do Swagger para documentação da API

**Frontend**

* **Arquitetura**: Modular, com separação de responsabilidades.
* **Ferramentas e bibliotecas utilizadas**:
  + **Next.js**: Framework para construção de aplicações React.
  + **TypeScript**: Superset de JavaScript.
  + **React**: Biblioteca para interfaces de usuário.
  + **Bootstrap**: Biblioteca de estilização responsiva.
  + **Formik e Yup**: Construção e validação de formulários.
  + **Js-cookie:** Biblioteca para manipulação de cookies.
  + **Jwt-decode**: Gerenciamento de cookies e tokens JWT.
* **Estrutura do código**

📦frontend

┣ 📂public #Diretório para arquivos públicos  
 ┃ ┗ 📂assets  
 ┃ ┃ ┗ 📂img #Arquivode imagens   
 ┃ ┃ ┃ ┣ 📜icon-editar.svg  
 ┃ ┃ ┃ ┣ 📜icon-pesquisar.svg  
 ┃ ┃ ┃ ┗ 📜icon-sair.svg  
 ┣ 📂src #Diretório principal do código-fonte.  
 ┃ ┣ 📂app #Contém as páginas e rotas da aplicação.  
 ┃ ┃ ┣ 📂admin #Contém as páginas administrativas.  
 ┃ ┃ ┃ ┣ 📂conteudo-criar   
 ┃ ┃ ┃ ┃ ┗ 📜page.tsx #Página de criação de conteúdo.  
 ┃ ┃ ┃ ┣ 📂conteudo-editar  
 ┃ ┃ ┃ ┃ ┗ 📂[id]  
 ┃ ┃ ┃ ┃ ┃ ┗ 📜page.tsx #Página para editar conteúdo.  
 ┃ ┃ ┃ ┗ 📜page.tsx #Página principal do administrador.

┃ ┃ ┣ 📂 alunos

┃ ┃ ┃ ┣ 📂alunos-criar

┃ ┃ ┃ ┃ ┃ ┗ 📜page.tsx #Página para criação de alunos

┃ ┃ ┃ ┣ 📂alunos-editar

┃ ┃ ┃ ┃ ┃ ┗ 📜page.tsx #Página para edição de alunos

┃ ┃ ┃ ┗ 📜page.tsx #Página para listagem de alunos  
 ┃ ┃ ┣ 📂conteudo-detalhes  
 ┃ ┃ ┃ ┗ 📂[id]  
 ┃ ┃ ┃ ┃ ┗ 📜page.tsx #Página com detalhes do conteúdo.  
 ┃ ┃ ┣ 📂login  
 ┃ ┃ ┃ ┗ 📜page.tsx #Página de login.  
 ┃ ┃ ┣ 📜layout.tsx #Layout principal da aplicação.  
 ┃ ┃ ┗ 📜page.tsx #Página principal da aplicação.  
 ┃ ┣ 📂components #Diretório de componentes reutilizáveis

┃ ┃ ┣ 📜alunosForm.tsx #Componente para criar ou editar informações do aluno.   
 ┃ ┃ ┣ 📜bar.tsx #Componente de barra.  
 ┃ ┃ ┣ 📜conteudoCard.tsx #Componente de cartão de conteúdo.  
 ┃ ┃ ┣ 📜conteudoDetails.tsx #Componente de detalhes de conteúdo.  
 ┃ ┃ ┣ 📜conteudoForm.tsx #Componente de formulário de conteúdo.  
 ┃ ┃ ┣ 📜header.tsx #Componente de cabeçalho.

┃ ┃ ┣ 📜ListAlunoAdm.tsx #Componente para listagem de alunos.  
 ┃ ┃ ┣ 📜ListPostAdm.tsx #Lista de posts para admin.  
 ┃ ┃ ┣ 📜ListPostUser.tsx #Lista de posts para usuário.  
 ┃ ┃ ┗ 📜longText.tsx #Componente de texto longo.  
 ┃ ┣ 📂interface #Diretório com interfaces TypeScript.  
 ┃ ┃ ┗ 📜IConteudo.tsx #Interface para conteúdo.

┃ ┃ ┗ 📜IUsuario.tsx #Interface para usuário.  
 ┃ ┣ 📂services #Diretório para comunicação com APIs.  
 ┃ ┃ ┣ 📜conteudo.tsx #Serviço para operações de conteúdo.  
 ┃ ┃ ┗ 📜usuario.tsx #Serviço para operações de usuário.  
 ┃ ┣ 📂styles #Diretório com estilos da aplicação  
 ┃ ┃ ┣ 📂modules  
 ┃ ┃ ┃ ┣ 📜barUser.module.css #Estilos para o componente de barra de usuário.  
 ┃ ┃ ┃ ┣ 📜conteudoDetails.module.css #Estilos para o componente de detalhes  
 ┃ ┃ ┃ ┣ 📜conteudoForm.module.css #Estilos para o componente de formulário  
 ┃ ┃ ┃ ┣ 📜header.module.css #Estilos para o componente de cabeçalho  
 ┃ ┃ ┃ ┣ 📜helloWord.module.css #Estilos para o componente de exemplo.

┃ ┃ ┃ ┣ 📜listAlunoAdm.module.css #Estilos para a lista de Alunos do admin.

┃ ┃ ┃ ┣ 📜listPostAdm.module.css #Estilos para a lista de posts do admin.  
 ┃ ┃ ┃ ┣ 📜listPostUser.module.css #Estilos para a lista de posts do usuário.  
 ┃ ┃ ┃ ┣ 📜login.module.css #Estilos para a página de login.  
 ┃ ┃ ┃ ┣ 📜longText.module.css #Estilos para o componente de texto longo.  
 ┃ ┃ ┃ ┗ 📜pageDashAdmin.module.css #Estilos para a página de dashboard do admin.  
 ┃ ┃ ┣ 📜bootstrap.css #Estilos do Bootstrap.  
 ┃ ┃ ┗ 📜globals.css #Estilos globais da aplicação.  
 ┃ ┣ 📂utils #Diretório para funções utilitárias.  
 ┃ ┃ ┗ 📜appUtils.tsx #Funções utilitárias da aplicação.  
 ┃ ┗ 📜middleware.ts #Middleware para a aplicação.  
 ┣ 📜.env #Arquivo de variáveis de ambiente locais.  
 ┣ 📜.eslintrc.json #Configuração do ESLint.  
 ┣ 📜docker-compose.yml #Configuração do Docker Compose.  
 ┣ 📜Dockerfile #Arquivo de configuração do Docker  
 ┣ 📜next.config.mjs #Configuração do Next.js  
 ┣ 📜package-lock.json #Arquivo de bloqueio de dependências do npm.  
 ┣ 📜package.json #Arquivo de configuração do npm.  
 ┣ 📜README.md #Arquivo de documentação do projeto.  
 ┗ 📜tsconfig.json #Configuração do TypeScript.

**Links do projeto**

Repositório do git: [*https://github.com/techchallengegroup28/Hackathon.git*](https://github.com/techchallengegroup28/Hackathon.git)

Drive: [Hackathon - OneDrive - Grupo 28](https://onedrive.live.com/?authkey=%21AIEz%5F9TuE8Lfpj0&id=EA1F7385C284F0DF%21109&cid=EA1F7385C284F0DF)

**Aprendizados e Próximos Passos**

Entre os aprendizados está a importância da colaboração contínua e da flexibilidade para lidar com problemas identificados durante o processo de desenvolvimento. Como próximo passo, planejamos expandir a plataforma com as melhorias citadas acimas e novas ideias que surgirão.

**Equipe: Grupo 28**

Filipe Mulitemo – RM353810

Lucas de Moraes – RM353135

Marcelo Neves – RM355808

Song Shik – RM353956

Thiago Santos – RM355404