

Documentação de produto de software

Versão 1.0

Equipe: Trix

Matrícula	Nome	Título do produto	Turma
2023123ISINF008	Eurilane Rodrigues de Melo	Enem Check-List	3°info
2023123ISINF015	Gabriela Araújo Gomes	Enem Check-List	3°info
2023123ISINF013	Yasmin Pereira da Silva	Enem Check-List	3°info

Sumário

1. Introdução
 - a. Tema
 - b. Objetivo do projeto
 - c. Justificativa do tema escolhido
 - d. Método de trabalho
 - e. Organização do trabalho
2. Descrição geral do Sistemas
 - a. Descrição do problema
 - b. Principais envolvidos e suas características
 - c. Regras de negócio
3. Requisitos do sistema
 - a. Requisitos funcionais
 - b. Requisitos não-funcionais
 - c. Protótipo e telas
 - d. Métricas e cronograma
4. Análise e Design
 - a. Arquitetura do sistema
 - b. Modelo do domínio
 - c. Diagramas (desenho do sistema)
 - d. Modelo de dados
 - e. Ambiente de desenvolvimento
 - f. Sistemas e componentes externos utilizados
5. Implementação
6. Testes
7. Implantação
8. Manual do usuário
9. Conclusões e considerações finais
10. Bibliografia

1. INTRODUÇÃO

a. Tema

O tema do projeto é o desenvolvimento de um aplicativo mobile educacional voltado para a organização e acompanhamento dos estudos para o ENEM, utilizando tecnologia React Native.

b. Objetivo do projeto

O objetivo do projeto é auxiliar estudantes do ENEM a organizarem seus estudos, permitindo marcar conteúdos já estudados, acompanhar o progresso geral, criar metas de estudo e acessar informações importantes relacionadas ao exame e ter acesso a cursos disponíveis com foco no Enem e canais no Youtube que disponibiliza vídeos aulas fazendo com que o aluno tenha facilidade em se organizar e estudar para o Enem.

c. Justificativa do tema escolhido

Muitos estudantes enfrentam dificuldades para se organizar e acompanhar o que já estudaram ao longo do ano. O ENEM possui uma grande quantidade de conteúdos, o que pode gerar confusão e desmotivação.

Dessa forma, o aplicativo foi criado para facilitar a organização, visualizar o progresso e estimular a constância nos estudos, tornando o aprendizado mais prático e acessível.

d. Método de trabalho

O projeto foi desenvolvido por meio de pesquisas sobre os conteúdos cobrados no ENEM, canais educacionais e cursos online. As funcionalidades foram planejadas previamente e implementadas utilizando React Native e Expo, permitindo execução em dispositivos Android e iOS. O sistema foi organizado em componentes e telas, com navegação simples e intuitiva. Testes manuais foram realizados para validar o funcionamento das funcionalidades.

e. Organização do trabalho

O trabalho está dividido em capítulos onde o primeira mostra a descrição geral do sistema, explicando o básico do aplicativo. Depois, os requisitos funcionais e não funcionais, ou seja, o que ele faz e como deve funcionar bem. E então vem a análise e design, planejando as telas e conexões, depois já tem implementação prática do sistema. Logo após, testes, implantação e manual para o usuário. Por fim, conclusão com resultados e as referências usadas.

2. DESCRIÇÃO GERAL DO SISTEMAS

a. Descrição do problema

O problema que foi abordado no nosso projeto é a falta de organização nos estudos para o ENEM, que dificulta o acompanhamento dos conteúdos estudados e do progresso ao longo do tempo.

b. Principais envolvidos e suas características

I. Usuários do Sistema

O público-alvo são estudantes se preparando para o ENEM, jovens do ensino médio ou cursinhos pré-vestibular, e usuários que acessam tudo pelo smartphone Android ou iOS.

II. Desenvolvedores do Sistema

Os principais desenvolvedores do sistema são estudantes de informática, que ficam responsáveis pela programação, testes e manutenção do aplicativo

c. Regras de negócio

O usuário pode marcar cada conteúdo como concluído ou pendente, o progresso é calculado sozinho baseado no que já foi feito, e tudo fica salvo mesmo se fechar o aplicativo.

3. REQUISITOS DO SISTEMA

a. Requisitos funcionais

- RF1 – Exibir as áreas do conhecimento do ENEM
- RF2 – Exibir as subáreas e seus respectivos conteúdos
- RF3 – Permitir marcar conteúdos como concluídos ou pendentes
- RF4 – Calcular e exibir o progresso geral em porcentagem
- RF5 – Permitir a criação e o gerenciamento de metas de estudo
- RF6 – Exibir informações sobre cursos preparatórios
- RF7 – Exibir canais educacionais com videoaulas
- RF8 – Mostrar o calendário do ENEM e a contagem regressiva para a prova
- RF9 – Disponibilizar uma tela de perfil com informações do usuário

b. Requisitos não-funcionais

- RNF1 – O aplicativo deve possuir uma interface simples e intuitiva
- RNF2 – O aplicativo deve ser responsivo, adaptando-se a diferentes tamanhos de tela
- RNF3 – O sistema deve armazenar os dados localmente utilizando AsyncStorage
- RNF4 – O aplicativo deve apresentar boa performance durante o uso
- RNF5 – O sistema deve permitir navegação fácil e intuitiva entre as telas

c. Protótipo

Foi feito um protótipo no Figma para mostrar visualmente como fica a interface do app Enem Check-List antes da implementação do código.

No Figma, foram criadas as principais telas do aplicativo, como:

O app tem tela inicial, telas de áreas e conteúdos, tela de progresso, menu, tela de perfil e splash screen (tela de início que aparece assim que abre o aplicativo). A partir desse protótipo visual, foi possível definir a estrutura das telas, padronizar o layout e a identidade visual, identificar melhorias de usabilidade antes da programação e guiar a implementação do aplicativo em React Native.

Link do protótipo: <https://www.figma.com/design/tTUKJnsQLTKGXGYCMufjqM/Sem-t%C3%ADculo?node-id=0-1&t=Sz9B5HwpICrVPzp0-1>

d. Métricas e cronograma

O desenvolvimento ocorreu dentro do tempo esperado para um projeto acadêmico. As principais funcionalidades foram implementadas conforme o planejamento, e testes manuais foram realizados durante todo o processo de desenvolvimento.

4. ANÁLISE E DESIGN

a. Arquitetura do sistema

Foi utilizado React com componentes modulares e Context API para gerenciar o estado geral, deixando tudo fluir bem entre as partes do app.

b. Modelo do domínio

O aplicativo gerencia usuário, conteúdos, progresso, metas, cursos, canais e calendário de forma integrada e prática.

c. Diagramas (desenho do sistema)

d. Modelo de dados

Os dados ficam salvos localmente no AsyncStorage em formato JSON, com progresso dos conteúdos e metas do usuário.

e. Ambiente de desenvolvimento

No ambiente de desenvolvimento foi utilizado as ferramentas Expo, React Native e Node.js.

f. Sistemas e componentes externos utilizados

Usamos Expo, React Navigation, AsyncStorage e Ionicons para fazer tudo funcionar da forma correta no aplicativo.

5. IMPLEMENTAÇÃO

A implementação foi realizada em JavaScript utilizando React Native, com uso de componentes funcionais, hooks como useState e useEffect, e Context API para o gerenciamento do estado global do aplicativo.

6. TESTES

6.1. Plano de Testes

Os testes foram realizados manualmente, verificando o correto funcionamento da marcação de conteúdos, do cálculo de progresso, do salvamento de dados e da navegação entre as telas.

6.2. Execução do Plano de Testes

Os testes foram realizados manualmente, verificando se todas as funcionalidades funcionam corretamente em diferentes telas.

7. IMPLANTAÇÃO

7.1. Diagrama de Implantação

7.2. Manual de Implantação

Para rodar o projeto, é necessário que o ambiente de desenvolvimento possua o Node.js instalado. Após a instalação do Node.js, deve-se instalar o Expo CLI. Em seguida, o usuário deve acessar a pasta do projeto por meio do terminal e executar o comando expo start. Após a execução desse comando, o Expo iniciará o projeto e abrirá uma interface no navegador, permitindo que o aplicativo seja executado em um dispositivo móvel através do aplicativo Expo Go ou em um emulador Android ou iOS. O aplicativo funciona localmente, não sendo necessária a utilização de servidores externos, pois os dados de progresso e metas do usuário são armazenados localmente utilizando o AsyncStorage.

8. MANUAL DO USUÁRIO

Ao iniciar o aplicativo Enem Check-List, o usuário visualiza inicialmente uma tela de carregamento (Splash Screen) e, em seguida, tem acesso à interface principal do sistema. O aplicativo é organizado por meio de um menu inferior, que permite a navegação entre as principais seções: Menu, Início e Perfil.

Na aba Início, o usuário pode visualizar as grandes áreas do ENEM, Linguagens, Ciências Humanas, Ciências da Natureza, Matemática e Redação. Ao selecionar uma área, são exibidas as subáreas correspondentes (por exemplo, Português, História, Biologia). Em cada subárea, o usuário tem acesso a uma lista de conteúdos específicos. Esses conteúdos podem ser marcados como concluídos ou pendentes ao serem tocados na tela. Essa marcação é salva automaticamente no sistema.

A aba Menu oferece acesso a funcionalidades adicionais do aplicativo, como:

- Visualização do progresso geral, exibido em porcentagem e em quantidade de conteúdos concluídos;
- Lista de cursos preparatórios para o ENEM;
- Lista de canais educacionais do YouTube;
- Calendário do ENEM, com a contagem regressiva para a próxima prova;
- Metas de estudo, onde o usuário pode adicionar metas personalizadas e marcá-las como concluídas.

Na aba Perfil, o usuário pode visualizar suas informações pessoais, acessar a área de configurações e visualizar opções como idioma. As configurações permitem apagar o progresso salvo ou limpar os dados do aplicativo.

Todas as informações, como progresso e metas, são armazenadas localmente no dispositivo por meio do AsyncStorage, garantindo que os dados permaneçam salvos mesmo após o fechamento do aplicativo.

9. CONCLUSÕES E CONSIDERAÇÕES FINAIS

O desenvolvimento do aplicativo Enem Check-List atingiu os objetivos propostos, oferecendo uma solução prática e funcional para a organização dos estudos voltados ao ENEM. O sistema permite ao estudante acompanhar seu progresso, organizar conteúdos por áreas, criar metas de estudo e acessar informações relevantes sobre o exame. Além disso, o aplicativo apresenta potencial de expansão futura, com a possibilidade de implementação de novas funcionalidades, como integração com banco de dados online, sistema de login, notificações de estudo e relatórios mais detalhados de desempenho, contribuindo para o apoio à preparação dos estudantes para o Enem.

10. BIBLIOGRAFIA

REACT NATIVE. Documentação oficial do React Native.

Disponível em: <https://reactnative.dev>. Acesso em: 2026.

EXPO. Documentação oficial do Expo.

Disponível em: <https://docs.expo.dev>. Acesso em: 2026.

BRASIL. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP).

Guia do Exame Nacional do Ensino Médio (ENEM). Disponível em:

<https://www.gov.br/inep>. Acesso em: 2026.

REACT. Hooks e Context API – Documentação oficial.

Disponível em: <https://react.dev>. Acesso em: 2026.

ASYNC STORAGE. @react-native-async-storage/async-storage – Documentação.

Disponível em: <https://react-native-async-storage.github.io/async-storage/>. Acesso em: 2026.

FIGMA. Figma – ferramenta de prototipação de interfaces.

Disponível em: <https://www.figma.com>. Acesso em: 2026.

IONICONS. Biblioteca de ícones Ionicons.

Disponível em: <https://ionic.io/ionicons>. Acesso em: 2026.