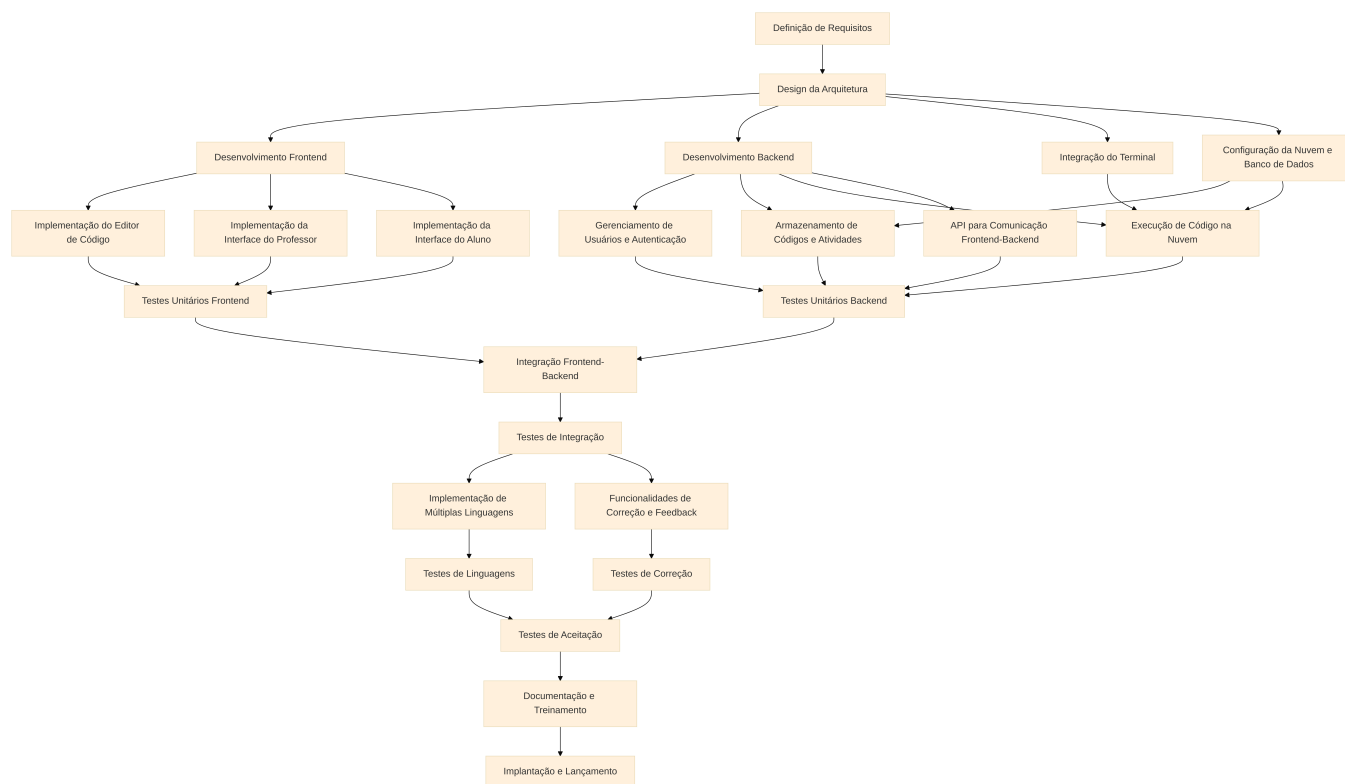


Relatório do Projeto IDE Educacional

Diagrama de Precedência do Projeto

A seguir, é apresentado o diagrama de precedência que ilustra as principais atividades e suas dependências para o desenvolvimento da IDE educacional. Este diagrama foi criado utilizando a sintaxe Mermaid.



Recomendações de Linguagens de Programação e Tecnologias

Conforme a análise dos requisitos do projeto, que visa uma IDE educacional online para o ensino de Lógica de Programação e Introdução à Linguagem de Banco de Dados, com suporte a Portugol, Python, JavaScript e SQL, as seguintes linguagens e tecnologias são recomendadas:

Frontend (Interface do Usuário no Navegador)

Para a interface do usuário, que será executada diretamente no navegador, as seguintes tecnologias são essenciais:

- **HTML5, CSS3 e JavaScript:** São os pilares de qualquer aplicação web moderna. HTML para estruturação do conteúdo, CSS para estilização e JavaScript para interatividade.

- **Framework JavaScript (React, Vue.js ou Angular):** Um framework como React (amplamente utilizado e com vasta comunidade), Vue.js (curva de aprendizado mais suave) ou Angular (solução completa para aplicações complexas) facilitará o desenvolvimento de uma interface rica e reativa, além de gerenciar o estado da aplicação de forma eficiente.

Backend (Lógica de Negócio, Armazenamento, Execução de Código)

O backend será responsável por gerenciar usuários, armazenar códigos, processar a execução de código na nuvem e fornecer APIs para o frontend. As opções mais adequadas são:

- **Python (com Flask ou Django):** Python é uma excelente escolha devido à sua simplicidade, vasta biblioteca e popularidade no ensino de programação. Frameworks como Flask (leve e flexível, ideal para APIs) ou Django (completo e robusto) podem ser usados para construir o servidor. Python também é muito adequado para a lógica de execução de código e sandboxing.
- **Node.js (com Express.js):** Se a equipe de desenvolvimento tiver forte familiaridade com JavaScript, Node.js permite usar a mesma linguagem no frontend e backend, o que pode otimizar o processo. Express.js é um framework minimalista e flexível para Node.js.

Banco de Dados

Para o armazenamento de dados, incluindo códigos dos alunos, atividades e informações de usuário, e dado o requisito de suporte a SQL:

- **PostgreSQL ou MySQL:** São sistemas de gerenciamento de banco de dados relacionais robustos, de código aberto e amplamente utilizados. Oferecem boa performance, segurança e são compatíveis com SQL, o que é fundamental para a parte de ensino de banco de dados.
- **SQLite (para desenvolvimento/testes locais):** Pode ser útil para prototipagem e desenvolvimento local devido à sua facilidade de uso (baseado em arquivo), mas para um ambiente de produção na nuvem, PostgreSQL ou MySQL seriam mais apropriados.

Suporte a Múltiplas Linguagens e Terminal

Para a execução de código e o terminal, que são aspectos cruciais da IDE:

- **WebAssembly (Wasm):** Para execução de linguagens como Python e JavaScript diretamente no navegador (client-side), Wasm pode ser uma opção para um ambiente de execução isolado e seguro. No entanto, para execução de código mais complexa ou que exija recursos de sistema, a execução no backend (server-side) é preferível.

- **Docker/Contêineres:** Para a execução segura e isolada de códigos em diferentes linguagens (Portugol, Python, JavaScript, SQL) no backend, o uso de contêineres (Docker) é altamente recomendado. Cada execução de código pode ser feita em um contêiner separado, garantindo isolamento e segurança.
- **Bibliotecas de Parsing e Execução:** Para Portugol, será necessário encontrar ou desenvolver um interpretador. Para Python, JavaScript e SQL, bibliotecas e ambientes de execução padrão podem ser utilizados no backend.
- **WebSockets:** Para a comunicação em tempo real entre o terminal no navegador e o backend (onde o código é executado), WebSockets são ideais para uma experiência fluida e interativa.

Considerações Finais

- **Segurança:** Dada a natureza educacional e a execução de código de alunos, a segurança (sandboxing, validação de entrada) deve ser uma prioridade máxima, especialmente na execução de código no backend.
- **Escalabilidade:** A escolha de tecnologias de nuvem e arquitetura de microsserviços (se aplicável) deve considerar a escalabilidade para suportar um número crescente de alunos e execuções de código simultâneas.
- **Facilidade de Manutenção:** Linguagens e frameworks com boa documentação e comunidades ativas facilitarão a manutenção e o desenvolvimento futuro do projeto.

Recomendação principal: Uma combinação de **React/Vue.js/Angular** para o frontend, **Python (Flask/Django)** para o backend, **PostgreSQL** para o banco de dados e **Docker** para a execução segura de código em múltiplas linguagens na nuvem, com **WebSockets** para a comunicação do terminal, parece ser a abordagem mais equilibrada e robusta para este projeto.