Prova Pleno

Pedro Henrique Mendes Batista

Problema 1

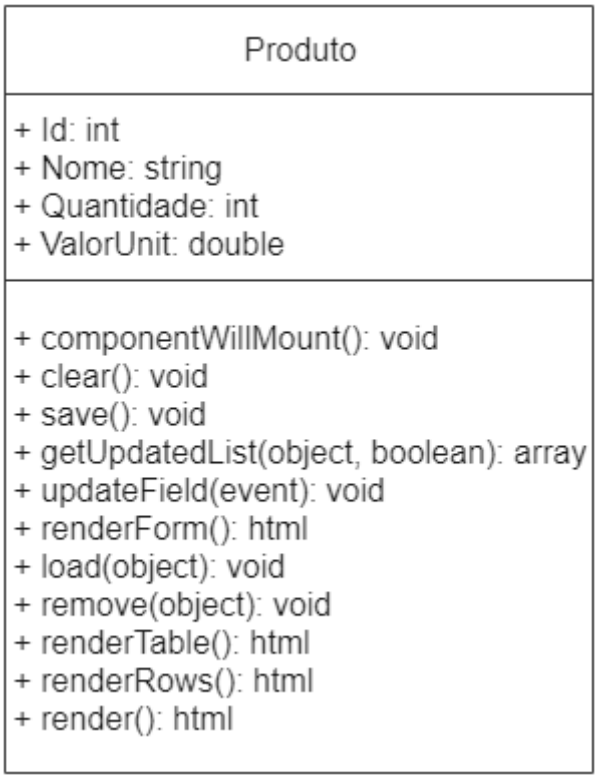
1) O id do aluno, studentId, não foi definido no atributo de valores do componente contexto, StudentContext. Assim, a função getSummaryByStudentId faz uma requisição para um id indefinido gerando o problema. Para solucionar o problema, basta inserir o studentId no atributo de valores do componente contexto, StudentContext.
2) A constante de aprovação, approved, não foi definida dentro do componente Score. Assim, quando o componente é renderizado, o campo de aprovação está indefinido. Para solucionar o problema, basta definir uma função que receba o valor total da nota e estabeleça uma condição que retorne uma string dizendo se o aluno foi aprovado ou não. Exemplo: const approved = total => total >= 60 : ‘Aprovado’ ? ‘Reprovado’;.
3) O componente de notas não está utilizado dentro do componente quadro de notas do aluno. A linha 17 do componente quadro de notas apresenta um chave vazia, { }. Assim, a função não consegue renderizar o componente dentro do quadro de notas. Para solucionar o problema, basta inserir o componente dentro da chave vazia do quadro de notas do aluno através do comando “props.children”. Exemplo: { props.children }.

Prova Pleno

Pedro Henrique Mendes Batista

Problema 2

Diagrama de classe (Front-end)

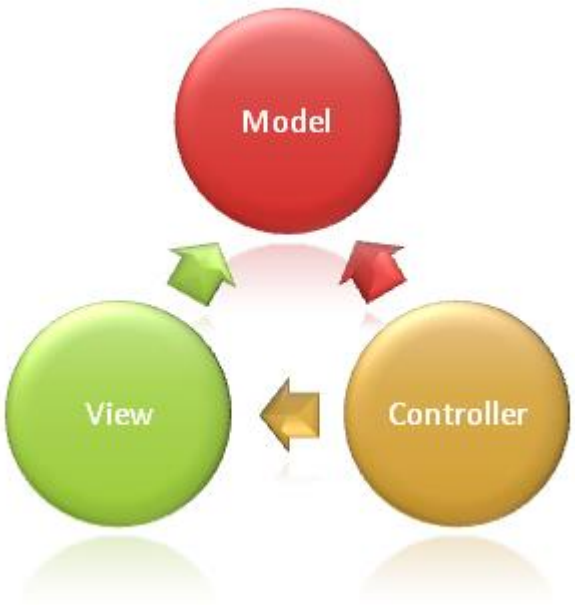


Prova Pleno

Pedro Henrique Mendes Batista

Arquitetura do Backend e Frontend

A arquitetura utilizada no back-end é o padrão MVC que separa a api em três grupos principais de componentes, sendo eles, Models, Views e Controllers. As requisições do usuário são encaminhadas para o componente ProdutoController que é responsável por trabalhar com o Modelo(Produto) para recuperar os resultados das consultas.



No front-end foi utilizado uma arquitetura baseada em componentes com o intuito de reutilização dos mesmos em diferentes cenários, redução no tempo de desenvolvimento e ter o mínimo de dependência com outros componentes. O componente ProductCrud é responsável por todas as funcionalidades do cadastro de produto e pela renderização do formulário, tabela e dados do produto.

Prova Pleno

Pedro Henrique Mendes Batista

Documentação das decisões do projeto

Inicialmente, foi desenvolvido um back-end baseado em objetos json para testar o desenvolvimento das funcionalidades da aplicação front-end.

Com as funcionalidades da aplicação front-end testadas, foi desenvolvido uma API Web com o ASP.net Core que armazena os dados em cache para ser a base do desenvolvimento final da aplicação back-end.

Finalizado a interação entre a API e a aplicação front-end, foi desenvolvido uma API Web com o ASP.net Core e o banco de dados no SQLite para ser o back-end do projeto.

Portanto, na pasta do projeto deve ser considerado apenas as pastas front-end dentro da pasta crud(front-end) e api(back-end).

Obs: leia o arquivo readme antes de inicializar o projeto!