

CHECKLIST – PROJETO FINAL

Para o desenvolvimento do projeto, foi utilizada a linguagem de programação JavaScript. No front-end utilizou-se o framework React e para o back-end, o Node com o framework Express. Para a persistência de dados foi utilizado o MongoDB; e o framework Jest foi utilizado com o objetivo de automatizar os testes necessários. Tendo isso em mente, a Checklist abaixo foi desenvolvida como forma de checar o desenvolvimento do código.

Visão Geral

1. O arquivo README fornece informações gerais sobre como rodar o código localmente?
2. O código é compilado?
3. O código roda sem problemas?
4. O código funciona como esperado?
5. A aplicação está de acordo com o pedido?
6. O padrão de estilização foi seguido?
7. O código possui fácil manutenção?
8. A performance do sistema é a esperada?
9. A documentação foi feita?
10. O código foi verificado?

Testes

1. O framework de testes foi bem configurado?
2. O framework de testes é utilizado?
3. São realizados os testes necessários ao código?
4. Existem casos de teste desnecessários?
5. Existem casos de teste que não são usados?

Nomenclatura

1. Os nomes dos arquivos são consistentes?
2. Os nomes dos arquivos são corretos?
3. Os arquivos possuem a extensão correta?
4. As variáveis possuem nomenclatura consistente?
5. As variáveis possuem nomenclatura descritiva e clara?
6. As funções possuem nomenclatura consistente?
7. As funções possuem nomenclatura descritiva e clara?
8. Os módulos possuem nomenclatura consistente?
9. Os módulos possuem nomenclatura descritiva e clara?
10. A nomenclatura está de acordo com o padrão de estilização?

Código

1. Existe alguma refatoração possível de ser feita?
2. Existe código exageradamente complexo?
3. Existe código desnecessariamente complexo?
4. Existe código não utilizado?
5. O código é comentado?
6. Os comentários feitos são desnecessários?
7. Os comentários feitos são exagerados?
8. Existe código duplicado?
9. Existe uso de valores definidos?
10. O código pode ser reutilizado?

Design

1. A API chama a camada de serviços?
2. A camada de serviços é a única a consumir a camada de persistência?
3. A camada de persistência é dividida em repositórios?
4. O código é construído em torno das entidades?
5. As entidades são classes?