

Reconhecendo padrões

Thassia já tem uma API onde consegue atualizar o nome e o preço de um produto e também tem algumas validações, como no código a seguir:

```
app.put('/api/produtos/:idProduto', async (requisicao, resposta) => {
  try {
    const dados = requisicao.body
    const id = requisicao.params.idProduto

    const encontrado = await TabelaDeProdutos.pegarPorId(id)

    if (!encontrado) {
      throw new Error('Produto não encontrado')
    }

    if (dados.nome.length === 0) {
      resposta.send(JSON.stringify({ mensagem: 'O campo "nome" está vazio' }))
    }

    if (dados.preco === 0) {
      resposta.send(JSON.stringify({ mensagem: 'O preço do produto não pode ser igual a zero' }))
    }

    const produto = new Produto(Object.assign({}, dados, { id: id }))
    await produto.atualizar()
    resposta.end()
  } catch (erro) {
    resposta.send(JSON.stringify({ mensagem: erro.mensagem }))
  }
})
```

Quais alterações Thassia pode criar para usar em sua API e identificar melhor os problemas que ocorrem?

Selecione uma alternativa

A

Trocar o uso de `resposta.send()` nas condicionais sobre o nome e preço do produto para emitir erros no lugar, usando a classe de Erro nativa do JavaScript.

B

Remover o método `resposta.send()` nas condicionais e criar um novo erro `CampoInvalido` para ser usado. Criar um erro chamado `NaoEncontrado` para representar quando uma informação não é encontrada na API.

C

Criar um erro chamado `ErroDaAPI` para usar em todos os casos de erro, enviando a descrição do ocorrido como parâmetro. Também, usar esse novo erro nas condicionais sobre nome e preço do produto.

D

Criar um erro chamado `NaoEncontrado` para usar dentro da condicional com a variável “encontrado”.

