TAD0203 - GESTÃO DE QUALIDADE DE SOFTWARE
DOCENTE: JOEL SANTOS

AVALIAÇÃO UNIDADE II

Objetivo: Trabalhar com testes de caixa branca e integração.

Parte 1: Testes de Caixa Branca

1. Com base nos preceitos lógicos dos testes de caixa branca, analise o seguinte código que tem o intuito de ajudar entregadores a determinar as rotas de entrega.

```
function verificarMelhorRota(pedidos, rotas) {
  let melhorRota = null;
  let menorDistancia = Infinity;

  rotas.forEach(rota => {
    let distanciaTotal = 0;

    pedidos.forEach(pedido => {
      distanciaTotal += calcularDistancia(pedido.endereco, rota);
    });

    if (distanciaTotal < menorDistancia) {
      menorDistancia = distanciaTotal;
      melhorRota = rota;
    }
  });

  return melhorRota;
}

function calcularDistancia(endereco1, endereco2) {
  //  calcular a distância entre dois endereços
  return Math.sqrt(
    Math.pow(endereco2.latitude - endereco1.latitude, 2) +
    Math.pow(endereco2.longitude - endereco1.longitude, 2)
  );
}
```

Testes:

- Teste a função verificarMelhorRota para diferentes cenários.
- Teste a função calcularDistancia para garantir que calcula corretamente.

Parte 2: Testes de Integração com Node Express

Implemente uma API com Node.js e Express com as seguintes rotas:

- GET /pedidos - Retorna uma lista de pedidos.
- POST /pedidos - Cria um novo pedido.
- GET /rotas - Retorna uma lista de rotas.
- POST /rotas - Cria uma nova rota.
- POST /melhor-rota - Verifica a melhor rota de entrega para os pedidos.

Testes:

- Verifique se a rota GET /pedidos retorna a lista de pedidos corretamente.
- Verifique se a rota POST /pedidos cria um novo pedido corretamente.
- Verifique se a rota GET /rotas retorna a lista de rotas corretamente.
- Verifique se a rota POST /rotas cria uma nova rota corretamente.
- Verifique se a rota POST /melhor-rota retorna a melhor rota de entrega corretamente.

Dicas:

1. POST /pedidos

Descrição: Cria um novo pedido.

- req.body: Deve conter os seguintes campos:
    - endereco: Um objeto com as coordenadas do endereço.
    - latitude: Número representando a latitude do endereço.
    - longitude: Número representando a longitude do endereço.
    - produto: String representando o nome do produto.
    - quantidade: Número inteiro representando a quantidade do produto.

2. POST /rotas

Descrição: Cria uma nova rota.

- req.body: Deve conter os seguintes campos:
    - latitude: Número representando a latitude da rota.
    - longitude: Número representando a longitude da rota.

3. POST /melhor-rota

Descrição: Verifica a melhor rota de entrega para os pedidos.

- req.body: Deve conter os seguintes campos:
    - pedidos: Um array de objetos, onde cada objeto representa um pedido.
        - Cada objeto deve ter:
            - endereco: Um objeto com as coordenadas do endereço do pedido.
            - latitude: Número representando a latitude do endereço do pedido.
            - longitude: Número representando a longitude do endereço do pedido.
    - rotas: Um array de objetos, onde cada objeto representa uma rota.
        - Cada objeto deve ter:
            - latitude: Número representando a latitude da rota.
            - longitude: Número representando a longitude da rota.

Instruções de Envio:

1. Crie um repositório no GitHub.
2. Adicione o código das funções e testes.
3. Adicione um README.md com instruções para executar os testes.
4. Envie o link do repositório como resposta desta avaliação.