

Instruções gerais:

- **Execução:** Cada aluno deverá codificar apenas uma rota, a ser definida de acordo com sua localização no laboratório;
- **Forma de entrega:** A entrega será feita presencialmente, mediante demonstração do funcionamento ao professor, que poderá realizar questionamentos sobre a implementação.

Descrição da atividade:

Desenvolver um servidor HTTP utilizando Node.js com o framework Express, contendo as rotas especificadas nos requisitos funcionais.

Estrutura esperada do projeto:

Organize os arquivos da aplicação da seguinte maneira:

```
├─ src/
|   └─ index.js           # Arquivo principal com a inicialização do servidor
|   └─ routes/
|       └─ operacoes.js  # contendo as rotas solicitadas
├─ .env                  # Arquivo com a variável de ambiente PORT=3100
└─ package.json          # Scripts para executar o servidor nos modos dev e start
```

Requisitos funcionais:**Rota 1 - Contagem de caracteres**

Descrição: Endpoint para contar a quantidade de vogais, consoantes, dígitos e demais caracteres em uma string recebida via parâmetro texto.

Código base:

```
router.get("/contar", function (req, res) {
  const { texto } = req.query;

  /* Codificar aqui */

  res.json({ mensagem: "Falta codificar" });
});
```

Casos de teste:

- URL: <http://localhost:3100/operacoes/contar>
Resposta: {"erro": "Parâmetro 'texto' é obrigatório"}
- URL: <http://localhost:3100/operacoes/contar?texto=pera: R9,82>
Resposta: {"vogais":2,"consoantes":3,"numeros":3,"outros":3}

Rota 2 - Cálculo de Fibonacci

Descrição: Endpoint que calcula o valor de Fibonacci de um número inteiro positivo recebido via parâmetro numero.

Código base:

```
router.get("/fibonacci", function (req, res) {  
  const { numero } = req.query;  
  
  /* Codificar aqui */  
  
  res.json({ mensagem: "Falta codificar" });  
});
```

Fórmula do Fibonacci:

$$F(n) = F(n-1) + F(n-2), \text{ com } F(0) = 0, F(1) = 1$$

Casos de teste:

- URL: <http://localhost:3100/operacoes/fibonacci>
Resposta: {"erro": "Parâmetro 'numero' é obrigatório"}
- URL: <http://localhost:3100/operacoes/fibonacci?numero=a>
Resposta: {"erro": "Informe um número inteiro"}
- URL: <http://localhost:3100/operacoes/fibonacci?numero=-3>
Resposta: {"erro": "Número deve ser maior ou igual a zero"}
- URL: <http://localhost:3100/operacoes/fibonacci?numero=8>
Resposta: {"resultado": 21}

Rota 3 - Potência

Descrição: Endpoint para calcular a potência com os parâmetros base e expoente.

Código base:

```
router.get("/potencia", function (req, res) {  
  const { base, expoente } = req.query;  
  
  /* Codificar aqui */  
  
  res.json({ mensagem: "Falta codificar" });  
});
```

Dicas:

- Use a função Math.pow(base, expoente) do JavaScript;
- Exemplo: Math.pow(2, -2) retorna 0.25.

Casos de teste:

- URL: <http://localhost:3100/operacoes/potencia>

Resposta: {"erro": "Parâmetro 'base' é obrigatório"}

- URL: <http://localhost:3100/operacoes/potencia?base=a&expoente=b>

Resposta: {"erro": "Parâmetro 'base' precisa ser um número"}

- URL: <http://localhost:3100/operacoes/potencia?base=2&expoente=b>

Resposta: {"erro": "Parâmetro 'expoente' precisa ser um número"}

- URL: <http://localhost:3100/operacoes/potencia?base=2&expoente=0>

Resposta: {"resultado": 1}

- URL: <http://localhost:3100/operacoes/potencia?base=2&expoente=-2>

Resposta: {"resultado": 0.25}

- URL: <http://localhost:3100/operacoes/potencia?base=2&expoente=5>

Resposta: {"resultado": 32}

Rota 4 - Verificação de palíndromo:

Descrição: Endpoint que verifica se uma string recebida via parâmetro texto é um palíndromo.

Código base:

```
router.get("/palindromo", function (req, res) {  
  const { texto } = req.query;  
  
  /* Codificar aqui */  
  
  res.json({ mensagem: "Falta codificar" });  
});
```

Dica:

- Converta para minúsculas para evitar erro com letras maiúsculas.

Casos de teste:

- URL: <http://localhost:3100/operacoes/palindromo>

Resposta: {"erro": "Parâmetro 'texto' é obrigatório"}

- URL: <http://localhost:3100/operacoes/palindromo?texto=arara>

Resposta: {"resultado": true}

- URL: <http://localhost:3100/operacoes/palindromo?texto=123321>

Resposta: {"resultado": true}

- URL: <http://localhost:3100/operacoes/palindromo?texto=banana>

Resposta: {"resultado": false}

Rota 5 - Cálculo do Máximo Divisor Comum (MDC)

Descrição: Endpoint para calcular o MDC entre dois números recebidos pelos parâmetros a e b.

Código base:

```
router.get("/mdc", function (req, res) {  
  const { a, b } = req.query;  
  
  /* Codificar aqui */  
  
  res.json({ mensagem: "Falta codificar" });  
});
```

Fórmula do MDC (algoritmo de Euclides):

$\text{MDC}(a, b) = \text{MDC}(b, a \bmod b)$, até que $b = 0$

Casos de teste:

- URL: <http://localhost:3100/operacoes/mdc>
Resposta: {"erro": "Parâmetro 'a' é obrigatório"}
- URL: <http://localhost:3100/operacoes/mdc?a=x>
Resposta: {"erro": "Parâmetro 'b' é obrigatório"}
- URL: <http://localhost:3100/operacoes/mdc?a=x&b=3>
Resposta: {"erro": "Parâmetro 'a' precisa ser um número"}
- URL: <http://localhost:3100/operacoes/mdc?a=2&b=y>
Resposta: {"erro": "Parâmetro 'b' precisa ser um número"}
- URL: <http://localhost:3100/operacoes/mdc?a=12&b=8>
Resposta: {"resultado": 4}