

**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=3200
|— package.json          # Scripts para executar o servidor nos modos dev e start
```

**Requisitos funcionais:****Rota 1** - Soma de dígitos pares e ímpares

Descrição: Endpoint que recebe uma string contendo dígitos e retorna a soma dos pares e ímpares separadamente.

**Rota:** /operacoes/somar-digitos

**Exemplo de código base:**

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

**Casos de teste:**

- URL: <http://localhost:3200/operacoes/somar-digitos?texto=>  
Resposta: {"erro": "Parâmetro 'texto' é obrigatório"}
- URL: <http://localhost:3200/operacoes/somar-digitos?texto=abc1234>  
Resposta: {"pares": 6, "impares": 4}

**Rota 2** - Conversão para binário

Descrição: Converte um número decimal para binário.

**Exemplo de código base:**

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

**Casos de teste:**

- URL: <http://localhost:3200/operacoes/binario?numero=>  
Resposta: {"erro": "Parâmetro 'numero' é obrigatório"}
- URL: <http://localhost:3200/operacoes/binario?numero=abc>  
Resposta: {"resultado": " Número deve ser inteiro não negativo"}
- URL: <http://localhost:3200/operacoes/binario?numero=12>  
Resposta: {"resultado": "1100"}