04

# Refatoração

Maria tem um código na sua aplicação que exibe o nome de todas as pessoas cadastradas no sistema da startup que trabalha.

```
const render = () => {
    clienteService.listaClientes()
        .then(data => {
            data.forEach(elemento => {
            tabela.appendChild(criaNovaLinha(elemento.nome
        })
    })
}
```

COPIAR CÓDIGO

Ela agora quer adicionar uma nova funcionalidade em sua aplicação, mas antes de começar a tarefa ela percebe uma chance de refatorar esse código. Qual a opção de refatoração com uso de async/await e de try/catch?

*Selecione uma alternativa*

A

```
const render = () =>  {
    const listaClientes = await clienteService.lista
    listaClientes.forEach(elemento => {
        tabela.appendChild(criaNovaLinha(elemento.no
    })

    catch(erro){
```

```
        console.log(erro)
        window.location.href="../telas/erro.html"
    }
}
```

B

```
const render = () =>  {
    try {
        const listaClientes = await clienteService.l
        listaClientes.forEach(elemento => {
            tabela.appendChild(criaNovaLinha(elemento
        })
    }
    catch(erro){
        console.log(erro)
        window.location.href="../telas/erro.html"
    }
}
```

C

```
const render = async () =>  {
    try {
        const listaClientes = await clienteService.l
        listaClientes.forEach(elemento => {
            tabela.appendChild(criaNovaLinha(elemento
        })
    }
    catch(erro){
        console.log(erro)
        window.location.href="../telas/erro.html"
    }
}
```