

07

CallBack Hell

Estamos criando um jogo em que a personagem precisa correr por uma série de labirintos. Para movimentar a personagem pelo primeiro labirinto criamos o seguinte código:

```
novePersonagem('100', 'Esquerda', function() {  
    movePersonagem('800', 'Direita', function() {  
        movePersonagem('200', 'Esquerda', function() {  
            movePersonagem('10', 'Direita', function() {  
                movePersonagem('60', 'Esquerda', function()  
            })  
  
        })  
    })  
})  
})
```

[COPIAR CÓDIGO](#)

No atual formato, código cria o inferno das funções auxiliares(callback hell) ou pirâmide da ruina(pyramid of doom), que acaba dificultando não só o entendimento do código, mas também sua manutenção caso a personagem precise fazer novas movimentações no cenário. Qual solução abaixo entregaria o mesmo resultado, mas resolvendo esses dois problemas citados utilizando promise?

Seleciona uma alternativa

A

```
movePersonagem('100', 'Esquerda')
    .then(() => ('800', 'Direita'))
    .then(() => ('200', 'Esquerda'))
    .then(() => ('10', 'Direita' ))
    .then(() => ('60', 'Esquerda' ))
```

B

```
movePersonagem('100', 'Esquerda')
    .then(() => movePersonagem('800', 'Direita'))
    .then(() => movePersonagem('200', 'Esquerda'))
    .then(() => movePersonagem('10', 'Direita' ))
    .then(() => movePersonagem('60', 'Esquerda' ))
```

C

```
movePersonagem()
    .then(() => movePersonagem('800', 'Direita'))
    .then(() => movePersonagem('200', 'Esquerda'))
    .then(() => movePersonagem('10', 'Direita' ))
    .then(() => movePersonagem('60', 'Esquerda' ))
```