

# FUTXICAÍADA



# PROMESSA



Abstração de um  
valor futuro

Incorpora meios de  
protesto

# PROMESSA

```
garçom('entrada', e => {  
  console.log(e)  
  garçom('bebida', b => {  
    console.log(b)  
    garçom('principal', p => {  
      console.log(p)  
      garçom('sobremesa', s => {  
        console.log(s)  
        garçom('conta', c => {  
          console.log(c)  
        })  
      })  
    })  
  })  
})
```

Broadcast de valores  
“Tunel do tempo” na  
vertical

# PROMESSA

```
/**
** Configuradora da promessa
**
** @callback cbOK - Caso de sucesso
** @param - Valor do sucesso
**
** @callback cbERROR - Caso de fracasso
** @param - Valor do fracasso
*/
function pConf (cbOK, cbERROR) {
  setTimeout(() => {
    const err = new Error('nanãninanão')

    cbERROR(err)
  }, 150)

  cbOK('é pra já')
}

var p = new Promise(pConf)
```

Cria-se uma promessa passando uma função que receberá dois gatilhos. O primeiro disparar o sucesso, o segundo o fracasso

# PROMESSA

```
const entrada = garçom('entrada')  
entrada.then(console.log)
```

```
const bebida = entrada.then(_ => garçom('bebida'))  
bebida.then(console.log)
```

```
const principal = bebida.then(_ => garçom('principal'))  
principal.then(console.log)
```

```
const sobremesa = principal.then(_ => garçom('sobremesa'))  
sobremesa.then(console.log)
```

```
const conta = sobremesa.then(_ => garçom('conta'))  
conta.then(console.log)
```