

1) Considere as seguintes Promises:

```
let promise1 = new Promise(function (resolve, reject) {
  resolve(true);
});

let promise2 = new Promise(function (resolve, reject) {
  setTimeout(function () {
    resolve(true)
  }, 3000);
});

let promise3 = new Promise(function (resolve, reject) {
  return resolve('Sucesso');
});
```

setTimeout faz a função ser executada apenas após o intervalo em milissegundos passado por parâmetro

Faça a 'promise3' ser executada somente após a execução com sucesso da 'promise1' e 'promise2';

2) Implemente o método 'catch' na resolução das promises do exercício 1

- 3) Encontre e corrija o erro do código a seguir até que a variável “pedido” receba o objeto criado no método “getDadosPedido” e seja mostrada como ‘objeto’ e seus atributos, e não ‘Promise’.

```
let pedido = null

pedido.then(function(result) {
  pedido = result
})

console.log(pedido)

function getDadosPedido() {
  return new Promise(function (resolve, reject) {
    try {
      resolve(
        {
          id: 600124578,
          data: new Date(),
          desconto: 5.00,
          valor: 100,
          parcelas: 1,
          observacaoPedido: 'Cliente com pressa',
          observacaoNota: '',
          somenteNotaFiscal: true,
          formaPagamento: { descricao: 'Dinheiro' },
          idPessoa: 204578
        }
      )
    } catch (e) {
      reject(e)
    }
  })
}
```

- 3.1) Crie uma nova Promise que receba os dados do pedido e calcule o valor final do pedido, subtraindo o atributo ‘valor’ por ‘desconto’. Insira um novo campo no objeto com o valor final do pedido e informe ao usuário.
- 3.2) Após criar o tratamento no item 3.1, crie uma nova promise que valide se o atributo ‘somenteNotaFiscal’ é verdadeiro. Caso seja, edite o atributo ‘observacaoNota’ com o número de parcelas, o valor final e a descrição da forma de pagamento.

