Testes Unitários

Comando utilizado: npx jest --verbose

Retorno:

```
 PASS  tests/Cliente.test.ts (5.539 s)
  Cliente
    constructor
      √ deve criar uma instância de Cliente com nome e número corretos (17 ms)
      √ deve permitir criação com valores vazios (1 ms)
    getters
      √ deve retornar o nome corretamente (1 ms)
      √ deve retornar o número corretamente
    casos especiais
      √ deve aceitar caracteres especiais no nome (1 ms)
      √ deve aceitar números no campo nome (1 ms)

 PASS  tests/Pedido.test.ts (5.615 s)
  Pedido
    constructor
      √ deve criar uma instância de Pedido com o cliente informado e sem pratos (18 ms)
    adicionarPrato
      √ deve adicionar um prato ao pedido (1 ms)
    calcularTotal
      √ deve retornar 0 quando nenhum prato for adicionado
      √ deve calcular corretamente o total com um prato adicionado
      √ deve calcular corretamente o total com múltiplos pratos adicionados (1 ms)
    getters
      √ deve retornar o cliente corretamente através do getter (1 ms)
      √ deve retornar os pratos corretamente através do getter (1 ms)

 PASS  tests/Prato.test.ts (5.756 s)
  Prato
    constructor
      √ deve criar uma instância de Prato com os valores corretos e quantidade padrão (11
ms)
      √ deve criar uma instância de Prato com quantidade informada (1 ms)
    adicionar_quantidade
      √ deve atualizar a quantidade do prato corretamente (1 ms)
    getters
      √ deve retornar os valores corretos através dos getters (1 ms)
    casos especiais
      √ deve atualizar a quantidade para zero quando informado zero (1 ms)

  console.log
```

```
    Garçom Carlos registrou o pedido para a mesa undefined.

      at Garcom.registrarPedido (src/Garcom.ts:14:17)

 PASS  tests/Mesa.test.ts (5.809 s)
  Mesa
    constructor
      √ deve criar uma instância de Mesa com os valores corretos (14 ms)
    métodos
      √ deve atualizar a disponibilidade da mesa (1 ms)
      √ deve realizar um pedido e atualizar a disponibilidade para false (1 ms)
      √ deve calcular a conta corretamente quando houver um pedido (2 ms)
      √ deve retornar zero ao calcular a conta se nenhum pedido foi realizado (1 ms)
    getters
      √ deve retornar o nome, número e disponibilidade corretamente (1 ms)
      √ deve retornar o pedido corretamente após realizar um pedido (2 ms)

  console.log
    Conta total para a mesa undefined: R$ 105.00

      at Garcom.calcularConta (src/Garcom.ts:20:17)

  console.log
    Mesa undefined está indisponível.

      at Garcom.atualizarMesa (src/Garcom.ts:27:17)

 PASS  tests/Garcom.test.ts (5.792 s)
  Garcom
    √ deve registrar um pedido (92 ms)
    √ deve calcular a conta com taxa (5 ms)
    √ deve atualizar a disponibilidade da mesa (3 ms)
    √ deve retornar o nome do garçom (1 ms)
    √ deve retornar a taxa do garçom (1 ms)
    √ deve retornar a disponibilidade do garçom (1 ms)

Test Suites: 5 passed, 5 total
Tests:       31 passed, 31 total
Snapshots:   0 total
Time:        6.617 s, estimated 8 s
Ran all test suites.
```

ao usar npx jest --coverage, obtemos também a tabela de cobertura do código:

```
File        | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
------------|---------|----------|---------|---------|-------------------
All files   |     100 |       75 |     100 |     100 |
```

```
Cliente.ts |     100 |      100 |     100 |     100 |
Garcom.ts  |     100 |       50 |     100 |     100 | 26
Mesa.ts    |     100 |      100 |     100 |     100 |
Pedido.ts  |     100 |      100 |     100 |     100 |
Prato.ts   |     100 |      100 |     100 |     100 |
-----------|---------|----------|---------|---------|-------------------
```

Modelagem

Relação	Tipo	Multiplicidade	Explicação
Cliente — Pedido	**Associação**	**1:1**	Um pedido pertence a um cliente
Pedido — Prato	**Composição**	**1:N**	Um pedido contém vários pratos
Mesa — Pedido	**Agregação**	**0:1**	Uma mesa pode ter um pedido ou estar vazia
Garcom — Mesa	**Dependência Leve**	**N:N**	`Garcom` depende de `Mesa`, mas não mantém uma relação permanente.
Garcom — Pedido	**Dependência Leve**	**1:N**	O garçom depende de `Pedido` para registrar pedidos em uma mesa, mas não mantém referência direta a eles

Diagrama de classes projeto

Bernardo Lopes Braga | February 18, 2025