

**Curso:** Bacharelado em Ciência da Computação  
**Disciplina:** Linguagens Comerciais de Programação – LCP  
**Professora:** Dra Simone das Graças Domingues Prado

## Aula Prática 07

### Conceito: Multithreading

1. Considere um simulador de corridas de Fórmula 1 que simula uma disputa entre dois pilotos: Hamilton e Fernando Alonso. Cada carro funciona de forma independente; o tempo de cada volta é dado por um valor randômico; programa deve esperar por este tempo sem fazer nada para então iniciar a próxima volta. No final da corrida (quando os dois carros completam 5 voltas): o simulador mostra o tempo acumulado para cada um dos pilotos e mostra quem venceu ou se houve empate. Veja um exemplo de execução abaixo.

```
Tempo do Carro 1: na volta 1 - 19
Tempo do Carro 2: na volta 1 - 97
Tempo do Carro 1: na volta 2 - 225
Tempo do Carro 2: na volta 2 - 180
Tempo do Carro 1: na volta 3 - 492
Tempo do Carro 2: na volta 3 - 456
Tempo do Carro 2: na volta 4 - 50
Tempo do Carro 1: na volta 4 - 38
Tempo do Carro 1: na volta 5 - 496
Tempo do Carro 2: na volta 5 - 198
TempoFinal do Carro 2: - 981
TempoFinal do Carro 1: - 1270
Carro 2 venceu: 981
```

2. Imaginemos um processo de compra on-line pela Internet, onde inúmeras pessoas podem consultar os itens disponíveis em estoque e realizar seus pedidos. Pois bem, como não queremos causar situações indigestas com nossos clientes, precisamos garantir com seus pedidos sejam faturados corretamente. Imagine que temos 5 aparelhos celulares em nosso estoque e que foi lançado uma promoção desse aparelho e 15 pessoas estão dispostas a entrar no tapa por um aparelho, bem temos que garantir que esse processo seja concretizado sem maiores problemas. Cada pessoa pode tentar uma única vez fazer a compra. Veja um exemplo de execução abaixo.

```
Pedido faturado para o cliente Cliente: 0
Pedido faturado para o cliente Cliente: 14
Pedido faturado para o cliente Cliente: 13
Pedido faturado para o cliente Cliente: 12
Pedido faturado para o cliente Cliente: 11
Nao tem estoque para o cliente Cliente: 10
Nao tem estoque para o cliente Cliente: 9
Nao tem estoque para o cliente Cliente: 8
Nao tem estoque para o cliente Cliente: 7
Nao tem estoque para o cliente Cliente: 6
Nao tem estoque para o cliente Cliente: 5
Nao tem estoque para o cliente Cliente: 4
Nao tem estoque para o cliente Cliente: 3
Nao tem estoque para o cliente Cliente: 2
Nao tem estoque para o cliente Cliente: 1
```

**3.** Escreva um programa composto por duas threads: a primeira deve contar e exibir na tela todos os números entre 1 e N (de forma crescente); a segunda deve contar e exibir na tela todos os números entre N e 1 (de forma decrescente). As duas threads devem ser executadas em paralelo. Veja um exemplo de execução abaixo.

```
*** Impressora 1: - 1
$$$$$$ Impressora 2: - 5
*** Impressora 1: - 2
$$$$$$ Impressora 2: - 4
*** Impressora 1: - 3
$$$$$$ Impressora 2: - 3
*** Impressora 1: - 4
*** Impressora 1: - 5
$$$$$$ Impressora 2: - 2
$$$$$$ Impressora 2: - 1
```