

**Tecnologia em Análise e Desenvolvimento de Sistemas**  
**Programação para Dispositivos Móveis – PDMS5**  
Android Studio com suporte ao Kotlin - Exercícios  
Pedro Northon Nobile  
nobile@ifsp.edu.br

## Objetivo

Fixar conceitos iniciais sobre Kotlin

## Exercício 1

Dado o código abaixo, a partir da lista já criada crie uma lista que contenha somente os cursos que tenham a palavra **Sistemas** em qualquer parte do nome. Imprima essa nova lista no formato dado como exemplo

```
val listaCursos: MutableList<String> = mutableListOf(  
    "Sistemas para Dispositivos Móveis",  
    "Análise e Desenvolvimento de Sistemas")  
  
listaCursos.add("Técnico em Informática para Internet")  
listaCursos.add("Manutenção de Aeronaves")  
listaCursos.add("Técnico em Células")  
listaCursos.add("Processos Gerenciais")
```

## Saída

```
0 – Sistemas para Dispositivos Móveis  
1 – Análise e Desenvolvimento de Sistemas
```

## Exercício 2

Dado o código abaixo de um Map, imprima seu conteúdo conforme mostrado na saída.

```
val familiaMap: MutableMap<String, String> = mutableMapOf(Pair("Pai", "Pedro"),  
    Pair("Mãe", "Marcela"),  
    Pair("Filho", "João"),  
    Pair("Caçula", "Cadu"),  
    Pair("Pet", "Paçoca"))
```

## Saída

```
Pai: Pedro  
Mãe: Marcela  
Filho: João  
Caçula: Cadu  
Pet: Paçoca
```

### Exercício 3

Dadas as funções abaixo e as chamadas na função main. Crie duas funções (foo e bar) de alta ordem que recebam essas funções como parâmetro e soltem a seguinte saída.

```
fun soma(i: Int, j: Int) = i + j

fun cat(i: String, j: String): String = "${i}${j}"

fun main() {
    println(foo(10, 20, ::soma))
    println(bar("Ped", "ro", ::cat))
}
```

### Saída

```
30
Pedro
```

#### Exercício 4

Crie uma função de alta ordem genérica (xpto) que receba como parâmetro qualquer uma das duas funções dadas no exercício 1 (soma e cat) e solte a seguinte saída.

```
fun main() {  
    println(xpto(2, 3, ::soma))  
    println(xpto("Jo", "ão", ::cat))  
}
```

#### Saída

```
5  
João
```

## Exercício 5

Dadas as funções abaixo:

1. Na função main, faça chamadas normais das funções de alta ordem passando as funções soma, multiplica e raizQuadrada como parâmetro de acordo com a equivalência dos parâmetros.
2. Na função main, faça chamadas das funções de alta ordem passando expressões lambdas equivalentes as funções soma, multiplica e raizQuadrada.

```
// Funções de alta ordem
fun processaInteiro(i: Int, f: (Int) -> Int): Int{
    return f(i)
}

fun processaInteiros(i: Int, j: Int, f: (Int, Int) -> Int): Int{
    return f(i, j)
}

// Funções
fun soma(i: Int, j: Int): Int{
    return i + j
}

fun multiplica(i: Int, j: Int): Int{
    return i * j
}

fun raizQuadrada(numero: Int): Int {
    for (n in 1..numero) {
        if (n * n == numero)
            return n
    }
    return -1
}

// Main
fun main() {
    // 1. Chamadas do primeiro passo

    // 2. Chamadas do segundo passo
}
```