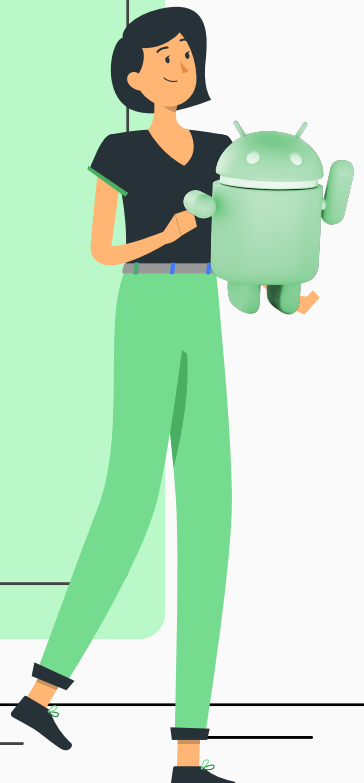
AULA 5

Kotlin



01 KOTLIN

Entendimento básico da linguagem

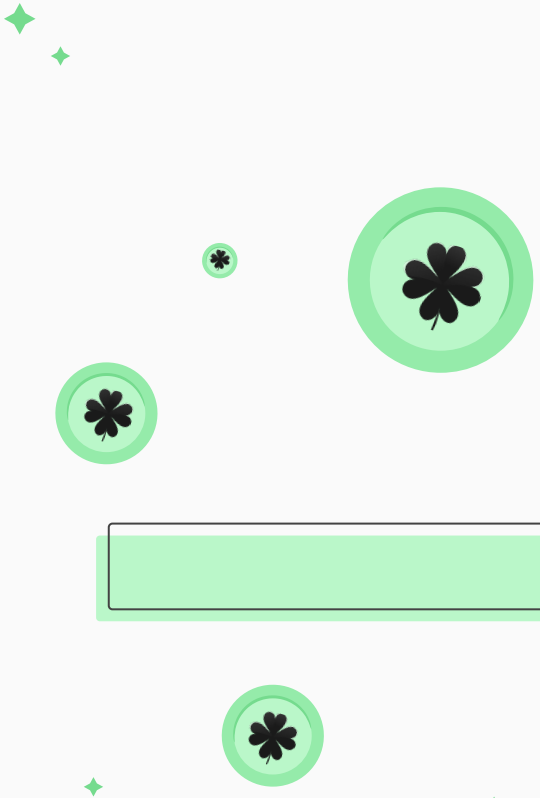
02 ALGORITMOS

Praticar o que aprendemos de kotlin

03 LAMBDAS

Programação funcional
Métodos úteis

AQUECIMENTO



PROGRAMAÇÃO FUNCIONAL

PALAVRAS CHAVE

- IMMUTABILITY
- STATELESS
- RECURSÃO
- HIGH ORDER FUNCTIONS
- LAZY EVALUATIONS

DOJO - FILTER & MAP

```
fun main(args: Array<String>) {
    val timesTwo = { x: Int -> x * 2 }
    val add: (Int, Int) -> Int = { x: Int, y: Int -> x + y }
    val toA: (Char) -> String = {letra: Char -> "$letra OI"}

    val list =(1..100).toList()

    print(list.filter { element ->
        element % 2 == 0
    })

    list.filter {
        it % 2 == 0
    }

    list.filter(::isEven)

}

fun isEven(i: Int) = i % 2 == 0
```

```
fun main(args: Array<String>) {

    //map()
    val list = (1..100).toList()

    val doubled = list.map { element -> element * 2 }
    list.map { it * 2 }

    print(doubled)

    val average = list.average()
    val shifted = list.map { it - average }

    print(shifted)

}
```

DOJO - TAKE, DROP & ZIP

```
fun main(args: Array<String>) {
    //take()
    val list =(1..100).toList()
    val takeFirst10 = list.take(10)
    print(takeFirst10)

    //drop()
    val drop10First = list.drop(10)
    print(drop10First)
}
```

```
fun main(args: Array<String>) {
    val list = listOf("hi", "there", "kotlin", "fans")
    val containsT = listOf(false, true, true, false)

    val zipped: List<Pair<String, Boolean>> = list.zip(containsT)
    val mapping = list.zip(list.map { it.contains("t")})

    print(zipped)
    print(mapping)

}
```

DESAFIO - Análise de dados