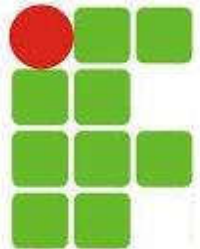


---

# ALGORITMOS

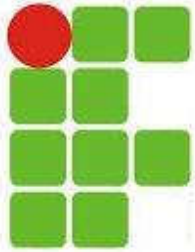
**Professor: Diego Oliveira**



---

**Conteúdo 17:  
Métodos Recursivos**

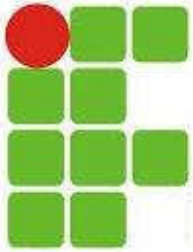




# Métodos Recursivos

- Ao chamar um método, as linhas deste são executadas e o controle volta para o local onde o método foi chamado, geralmente no main()
- Porém é possível chamar outros métodos a partir de um método, ou até mesmo o próprio método novamente, a isso é dado o nome de **RECURSIVIDADE**





# Métodos Recursivos

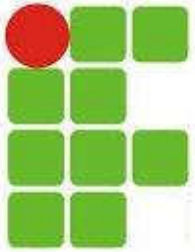
- Vamos desenhar vários quadrados:

```
public static void main(String[] args) {
    Scanner leia = new Scanner(System.in);
    System.out.println("Qual o lado do maior quadrado");
    int lado = leia.nextInt();
    quadrado(lado);
}

public static void quadrado(int lado) {
    for (int i = 1; i <= lado; i++) {
        for (int j = 1; j <= lado; j++) {
            System.out.print("0 ");
        }
        System.out.println("");
    }
    System.out.println("");

    if(lado > 1){
        quadrado(lado-1);
    }
}
```

Qual o lado do maior quadrado  
5  
0 0 0 0 0  
0 0 0 0 0  
0 0 0 0 0  
0 0 0 0 0  
0 0 0 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0  
0 0 0  
0 0 0  
0 0 0  
0 0  
0 0  
0

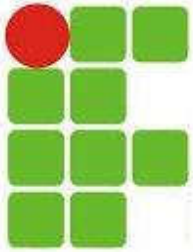


# Métodos Recursivos

- Em uma prova passada fizemos o seguinte algoritmo:

```
public class Fibonacci {  
    public static void main(String[] args) {  
        Scanner x = new Scanner(System.in);  
        System.out.println("Digite a quantidade de termos");  
        int n1 = 0, n2 = 1;  
        for (int i = x.nextInt(); i > 0; i--) {  
            System.out.print(n1 + " ");  
            int temp = n1 + n2;  
            n1 = n2;  
            n2 = temp;  
        }  
    }  
}
```



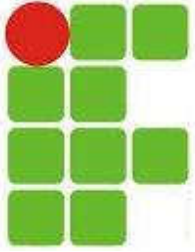


# Métodos Recursivos

- Ele pode ser reescrito recursivamente:

```
public class Recursividade {  
    public static void main(String[] args) {  
        Scanner leia = new Scanner(System.in);  
        System.out.println("Digite quantos números da série você quer: ");  
        int x = leia.nextInt();  
        for (int i = 0; i < x; i++) {  
            System.out.println(fibonacci(i));  
        }  
    }  
  
    public static int fibonacci(int n) {  
        if (n == 0) {  
            return 0;  
        } else if (n == 1) {  
            return 1;  
        }  
        return fibonacci(n - 1) + fibonacci(n - 2);  
    }  
}
```





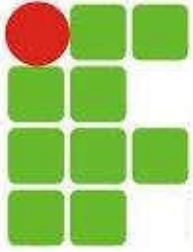
---

# Métodos Recursivos

---

- Métodos recursivos podem ser utilizados para várias finalidades envolvendo sequências, desenhos de figuras e busca de informações dentro de um banco de dados.
- No exercício de hoje vamos refazer uma questão de uma lista passada utilizando recursividade



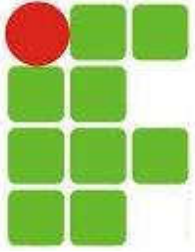


# Exercício

- Leia um número inteiro  $X$
- Verifique CADA número de 1 até  $X$  se é par ou ímpar
- Se for par imprima todos os números pares anteriores a ele separados por VIRGULA usando recursividade
- Se for ímpar imprima todos os números ímpares anteriores a ele separados por PONTO-E-VÍRGULA usando recursividade



- Crie um método para verificar se um número é **PRIMO** e imprima quando o número o for, de 1 a  $X$



---

# Perguntas?

---

