

# MAC0113 - Introdução à Computação para Ciências Humanas

**Aula 22**

---

**Sejam bem-vindas, sejam bem-vindos!**

**R. Hirata Jr.**

# Objetivos de hoje

- Ao final da aula você deve ter implementado e testado:
  - mais um algoritmo de ordenação clássico
  - a função fatorial recursiva e a função fatorial não recursiva
  - a função média de um vetor
  - a função mediana de um vetor de tamanho ímpar
  - a função variância de um vetor usando a função média implementada anteriormente
  - a função desvio padrão usando a função variância implementada anteriormente
  - ler o help das funções `sort` e `which` do R.
  - Desafio. Implemente a função de Fibonacci recursiva e não-recursiva:
    - $F_0 = 0, F_1 = 1, F_N = F_{N-1} + F_{N-2}$

# Programando em R

```
bubble <- function(V,N) {  
  i = N  
  while (i>1) {  
    j = 1  
    while (j<i) {  
      if (V[j] > V[j+1]) {  
        tmp = V[j]  
        V[j] = V[j+1]  
        V[j+1] = tmp  
      }  
      j = j + 1  
    }  
    i = i - 1  
  }  
  return(V)  
}
```

## Funções recursivas em R

- Uma função recursiva é aquela que chama a si mesma.
- Para se criar uma função recursiva, tem-se que ter dois elementos:
  - Base da recursão. Por exemplo:
    - if ( $K==0$ ) fattmp = 1, que vem do fato que  $\text{fatorial}(0) = 1$
  - Passo da recursão. Por exemplo:
    - fattmp =  $K * \text{meufat}(K-1)$ , que vem do fato que
      - $\text{fatorial}(N) = N * \text{fatorial}(N-1)$
- Identificados esses elementos, escreve-se o código.

## Funções recursivas em R

- Uma função recursiva é aquela que chama a si mesma.

```
meufat <- function(K) {  
  if (K==0) {  
    fattmp = 1  
  } else {  
    fattmp <- K*meufat(K-1)  
  }  
  return(fattmp)  
}
```

# Obrigado!

---