

MAC 113 — Introdução à Ciência da Computação

Aula 5

Nelson Lago

1º/2025



Programando

- 1 algoritmo *vs* implementação
- 2 entrada de dados → processamento → resultado
 - ▶ Mostra o resultado para o usuário → `cat(..., "\n")`
 - ▶ **Utiliza o resultado como dado para fazer outra coisa**
- 3 Existem *tipos de dados* diferentes em R (*integer*, *numeric*, *character*, *logical*...)
- 4 Expressões são coisas que têm um *valor* (de algum *tipo*)
 - ▶ E podem ser combinadas ou utilizadas como partes de outras expressões



`2 + 3 + 7` (*integer*)

`2 > 3 && 5 > 4` (*logical*)

Expressões lógicas e tipos

Só vou à praia se:

- A temperatura no fim-de-semana for maior que 25°C
- Eu tiver pelo menos R\$100
- A minha nota na prova for A ou B

```
temp <- 29L
saldo <- 100.39
nota <- "B"
cat(temp > 25 && saldo >= 100 && (nota == "A" || nota == "B"), "\n")
```

```
TRUE
```

The diagram illustrates the types of variables and the result of an expression. It shows the following assignments and their types:

- `temp <- 29L`: `temp` is assigned the value `29L`, which is an **integer**.
- `saldo <- 100.39`: `saldo` is assigned the value `100.39`, which is a **numeric** value.
- `nota <- "B"`: `nota` is assigned the value `"B"`, which is a **character**.

The expression `cat(temp > 25 && saldo >= 100 && (nota == "A" || nota == "B"), "\n")` is shown, followed by a dashed line and the output `TRUE`. The output `TRUE` is identified as a **logical** type.

Comandos essenciais vistos até agora

- `x <- 5` → atribui o valor 5 ao nome `x`
- `cat(blah, "\n")` → imprime o valor da variável `blah`
- `cat("blah", "\n")` → imprime o texto `blah`
- `x <- readline()` → lê o que o usuário digita e atribui ao nome `x`
 - ▶ `cat("Qual o seu nome?", "\n")`
`nome <- readline()`
 - ▶ Equivale a:
`nome <- readline("Qual o seu nome? ")`
- `celsius <- function(f) { return(5 * (f - 32) / 9) }`
→ cria uma função com o nome `celsius()`
- `temp <- celsius(96)` → expressão usando a função `celsius()`

Execução condicional

```
n <- as.integer(readline("Digite um número natural: "))
if (n %% 2 == 0) {
  cat("O número", n, "é par!", "\n")
}
cat("Ahazei!", "\n")
```

- **`n %% 2 == 0` → TRUE ou FALSE**

- ▶ Embora possamos ler “se condição”, na verdade R faz “se o valor da expressão é verdadeiro (**TRUE**)”
- ▶ É **como se** R executasse `if` (condição == **TRUE**)

- **O bloco a ser executado ou não é definido por { e }**

- ▶ Novamente, a indentação é uma convenção, mas na prática é essencial

Execução condicional – else

```
n <- as.integer(readline("Digite um número natural: "))
if (n %% 2 == 0) {
  cat("O número", n, "é par!", "\n")
} else {
  cat("O número", n, "é ímpar!", "\n")
}
cat("Ahazei!", "\n")
```

- Há dois “lados” do condicional

- ▶ Sem uma variável (“n”), o programa sempre executaria o mesmo “lado”

- O estado da variável só importa no momento do teste

- ▶ Se ela mudar em seguida, não afeta o condicional

- Com else, os “lados” são mutuamente excludentes

- ▶ Um e apenas um deles é executado

Exercício — cálculo da nota 1/3

Faça um programa que recebe as notas de um aluno desta disciplina e calcula sua média final de acordo com os critérios definidos no eDisciplinas

```
p1 <- as.numeric(readline("Digite a nota da prova 1 "))
p2 <- as.numeric(readline("Digite a nota da prova 2 "))
ep1 <- as.numeric(readline("Digite a nota do EP 1 "))
ep2 <- as.numeric(readline("Digite a nota do EP 2 "))
mp <- (p1 + 2*p2)/3
mep <- (ep1 + 2*ep2)/3
if (mp >= 5 && mep >= 5) {
  mf <- (mep + 2*mp)/3
} else {
  if (mp < mep) {
    mf <- mp
  } else {
    mf <- mep
  }
}
cat("A média final é", mf, "\n")
```

Exercício — cálculo da nota 2/3

Modifique esse programa para usar uma função auxiliar `menor(a, b)`

```
menor <- function(a, b) { if (a < b) { return(a) } else { return(b) } }  
p1 <- as.numeric(readline("Digite a nota da prova 1 "))  
p2 <- as.numeric(readline("Digite a nota da prova 2 "))  
ep1 <- as.numeric(readline("Digite a nota do EP 1 "))  
ep2 <- as.numeric(readline("Digite a nota do EP 2 "))  
mp <- (p1 + 2*p2)/3  
mep <- (ep1 + 2*ep2)/3  
if (mp >= 5 && mep >= 5) {  
  mf <- (mep + 2*mp)/3  
} else {  
  mf <- menor(mp, mep)  
}  
cat("A média final é", mf, "\n")
```


Exercício — cálculo da nota 3/3

Modifique esse programa para usar uma função auxiliar `ponderada(a, b)`

```
ponderada <- function(a, b) {  
  return ((a+2*b)/3)  
}  
  
menor <- function(a, b) { if (a < b) { return(a) } else { return(b) } }  
  
p1 <- as.numeric(readline("Digite a nota da prova 1 "))  
p2 <- as.numeric(readline("Digite a nota da prova 2 "))  
ep1 <- as.numeric(readline("Digite a nota do EP 1 "))  
ep2 <- as.numeric(readline("Digite a nota do EP 2 "))  
mp <- ponderada(p1, p2)  
mep <- ponderada(ep1, ep2)  
if (mp >= 5 && mep >= 5) {  
  mf <- ponderada(mep, mp)  
} else {  
  mf <- menor(mp, mep)  
}  
  
cat("A média final é", mf, "\n")
```

Exercício — ano bissexto

- **O que é um ano bissexto?**

- ▶ O tempo de translação da terra ao redor do sol não é exatamente 365 dias; assim, a cada 4 anos, temos um ano bissexto para compensar essa diferença
- ▶ No entanto, essa compensação não é perfeita; por conta disso, a cada 100 anos, um ano que normalmente seria bissexto não é
- ▶ Essa segunda compensação também não é perfeita e, por isso, a cada 400 anos, um ano que excepcionalmente deixaria de ser bissexto é bissexto normalmente
 - » *Um ano é bissexto se é múltiplo de 4, exceto quando é múltiplo de 100 mas não de 400*
 - » *Um ano é bissexto se é múltiplo de 400 ou se é múltiplo de 4 mas não de 100*
 - » *Um ano é bissexto se é múltiplo de 4 mas não de 100, exceto se for múltiplo de 400*

Exercício — ano bissexto

Dado um ano, informar se ele é bissexto ou não

```
main <- function() {  
  ano <- as.integer(readline("Digite o ano: "))  
  if ((ano %% 4 == 0 && ano %% 100 != 0) || ano %% 400 == 0) {  
    cat("O ano é bissexto", "\n")  
  } else {  
    cat("O ano não é bissexto", "\n")  
  }  
}  
main()
```

Exercício — estações do ano

Dada uma data (dia e mês), informe se a data acontece no verão (21 de dezembro a 20 de março), outono (21 de março a 20 de junho), inverno (21 de junho a 22 de setembro) ou primavera (23 de setembro a 20 de dezembro). Use números de 1 a 12 para representar os meses do ano.

```
dia <- as.integer(readline("Digite o dia: "))
mês <- as.integer(readline("Digite o mês (1 a 12): "))
if (
    { cat("Verão!", "\n") }
if (
    { cat("Outono!", "\n") }
if (
    { cat("Inverno!", "\n") }
if (
    { cat("Primavera!", "\n") }
```