

1

Introdução

ESQUEMA DO CAPÍTULO

1.1 EXERCÍCIO DE PROGRAMAÇÃO
1.2 SISTEMAS DE PROGRAMAÇÃO
1.3 ALGORITMOS E FLUXOGRAMAS

1.4 ESTRUTURAS DE DADOS
1.5 MODULARIZAÇÃO

1.2 Sistemas de Programação

1.2.1 Técnicas de programação

1.2.1.1 Processo de desenvolvimento de programas

PROBLEMA análise	conhecimento do problema
SOLUÇÃO CONCEITUAL programação	criatividade domínio da linguagem
ALGORITMO codificação	conhecimento da linguagem e do equipamento
PROGRAMA FONTE digitação	conhecimento do teclado e do editor
PROGRAMA FONTE DIGITADO compilação	computador
PROGRAMA OBJETO Execução	
RESULTADOS Avaliação	usuário + analista/programador

1.2 Sistemas de Programação (cont.)

1.2.1.2 Desenvolvimento de programas

Etapas:

1. definição do problema
2. projeto do algoritmo
3. codificação
4. depuração
5. testes
6. documentação

1) definição do problema:

- especificação do problema:
 - o que é o problema;
- especificação das entradas:
 - quais são os dados de entrada;
 - formato da entrada;
 - valores possíveis;
 - quantidade de dados de entrada;
- especificação das saídas:
 - quais são os dados de saída;
 - formato da saída;



1.2 Sistemas de Programação (cont.)

1.2.1.2 Desenvolvimento de programas (cont.)

Exemplo:

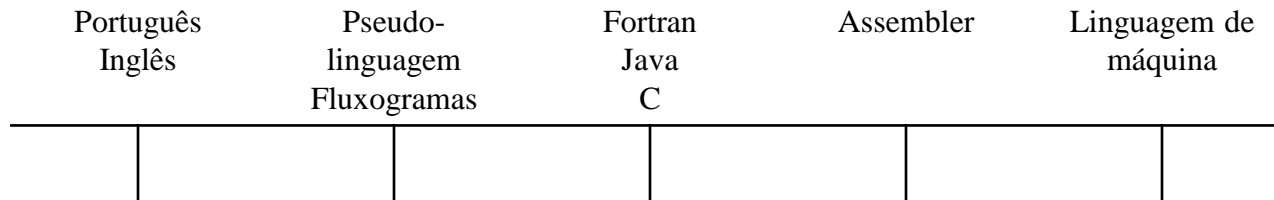
- problema: calcular o salário de um vendedor pela fórmula:
 - **salario** \leftarrow 1.212 + 10% de **vendas**
- entrada:
 - valor de vendas, como um número real
- saídas:
 - vendas = R\$ xx.xxx,xx
 - salário = R\$ x.xxx,xx
- condições excepcionais:
 - se **vendas** negativo, imprimir mensagem de erro
 - Se **salario** > 9.999,99, imprimir mensagem de aviso

1.2 Sistemas de Programação (cont.)

1.2.1.2 Desenvolvimento de programas (cont.)

2) projeto do algoritmo

- algoritmo:
 - método de solução
 - algoritmo: sequência de instruções suficientemente detalhadas para resolver um tipo específico de problema;
- características:
 - instruções não ambíguas
 - instruções “executáveis”
 - termina em tempo finito
- linguagens de descrição de algoritmos
 - diversos níveis



1.2 Sistemas de Programação (cont.)

1.2.1.2 Desenvolvimento de programas (cont.)

Exemplo

– versão 1:

```
algoritmo  
  ler vendas  
  calcular salário  
  imprimir vendas e salário  
fim algoritmo
```

– versão 2:

```
algoritmo  
  declare vendas, comissão: numerico;  
  leia vendas  
  salario ← 1212 + vendas/10  
  escreva “vendas=”, vendas  
  escreva “salario=”, salario  
fim algoritmo
```

1.2 Sistemas de Programação (cont.)

1.2.1.2 Desenvolvimento de programas (cont.)

3) codificação

- MINITAB (disponível em <http://www.est.ufmg.br/ftp/fcruz/pacotes/salario.mac>)

```
macro
# Autor:
# Frederico R. B. Cruz
# Departamento de Estatística - UFMG
# E-mail: fcruz@est.ufmg.br - (c) 2022
# Funcao:
# calcular salario a partir das vendas pela expressao
# salario = 1212 + 10% das vendas
# Utilizacao:
# MTB> %DRIVE:/CAMINHO/salario
# template
salario
# declaracoes
mcolumns vendas salar
# leia vendas
note entre com as vendas
read vendas;
file "terminal";
nobs 1.
# calcule salario
let salar = 1212 + vendas/10
# imprima resultados
print vendas salar
#final
endmacro
```

1.2 Sistemas de Programação (cont.)

1.2.1.2 Desenvolvimento de programas (cont.)

3) codificação (cont.)

- R (disponível em <http://www.est.ufmg.br/ftp/fcruz/pacotes/salario.R>)

```
salario <- function() {  
  # Autor:  
  # Frederico R. B. Cruz  
  # Departamento de Estatística - UFMG  
  # E-mail: fcruz@est.ufmg.br - (c) 2022  
  # Funcao:  
  # calcular salario a partir das vendas pela expressao  
  # salario = 1212 + 10% das vendas  
  # Utilizacao:  
  # > source("DRIVE:/CAMINHO /salario.R")  
  # > salario()  
  
  # ler vendas  
  cat("entre com as vendas ")  
  venda <- as.numeric(readline())  
  
  # calcular salario  
  salar <- 1212 + 0.10 * venda  
  
  # imprimir vendas e salario  
  cat("venda =", venda, " salario =", salar, "\n")  
}
```


1.2 Sistemas de Programação (cont.)

1.2.1.2 Desenvolvimento de programas (cont.)

4) depuração

- descoberta, localização e correção de erros:
 - de compilação:
 - pelo compilador/interpretador;
 - de execução:
 - comparar programa com o algoritmo;
 - simular a execução do programa no papel;
 - imprimir resultados intermediários;
 - de lógica:
 - idem;

5) testes

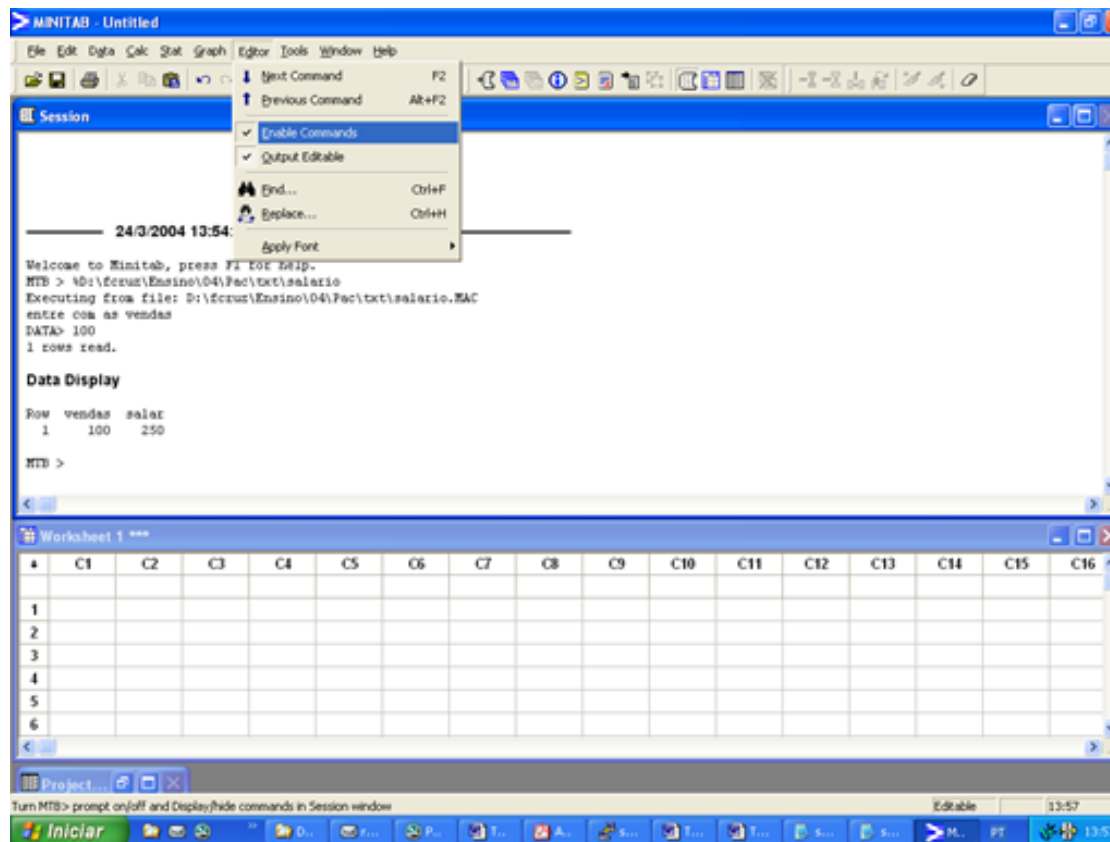
- verificação se o programa funciona para diferentes dados (tentativa de mostrar que o programa produz resultados corretos para qualquer dado);

6) documentação

- documentação do usuário → utilização;
- documentação técnica → metodologia.

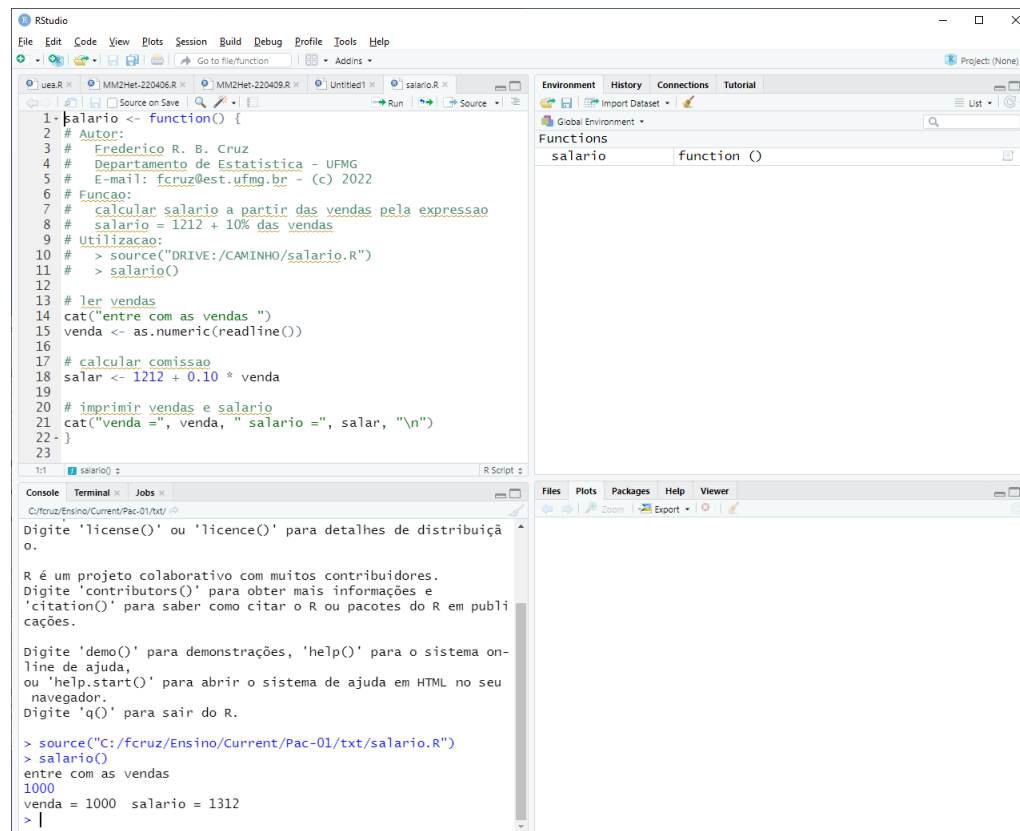
1.2 Sistemas de Programação (cont.)

1.2.1.3 Utilização do MINITAB



1.2 Sistemas de Programação (cont.)

1.2.1.3 Utilização do R



```
1- salario <- function() {
2 # Autor:
3 # Frederico R. B. Cruz
4 # Departamento de Estatística - UFMG
5 # E-mail: fcruz@est.ufmg.br - (c) 2022
6 # Função:
7 # calcular salario a partir das vendas pela expressao
8 # salario = 1212 + 10% das vendas
9 # Utilizacao:
10 # > source("DRIVE:/CAMINHO/salario.R")
11 # > salario()
12
13 # ler vendas
14 cat("entre com as vendas ")
15 venda <- as.numeric(readline())
16
17 # calcular comissao
18 salar <- 1212 + 0.10 * venda
19
20 # imprimir vendas e salario
21 cat("venda =", venda, " salario =", salar, "\n")
22 }
23
```

```
> source("C:/fcruz/Ensino/Current/Pac-01/txt/salario.R")
> salario()
entre com as vendas
1000
venda = 1000  salario = 1312
> |
```