

```
1
2
3  Algoritmo 'Lógica de' {
4
5      [ Programação 1 ]
6
7
8
9
10
11
12 }
13
14
```

Unidade 02

Prof. Daniel Caixeta



```
1  'Sumário' {
2
3      </01> Lógica & Algoritmos [...]
4          < 1.1. Lógica: Conceitos iniciais. 1.2. A lógica e áreas dos saberes. 1.3. A
5              lógica na informática [...]. >
6
7      </02> Hardware & Software
8          < 2.1. Em busca por soluções [...]. 2.2. O computador [...]. >
9
10     </03> Algoritmizando
11         < 3.1. O desenvolvimento [...]. 3.2. Conceito de Algoritmo. 3.3. Como
12             construir um algoritmo? 3.4. Tipos de algoritmos. 3.5 Alguns exemplos. >
13     Referência
14 }
```

01 {

[Lógica & Algoritmos]

< Conceitos iniciais >

}

1.1. Lógica: Conceitos iniciais

- 1
2 • Aristóteles (384 a.C. – 322 a.C.) é considerado o pai da ló-
3 gica. No entanto, ele a denominava “razão”. O termo “lógi-
4 ca¹” só passou a ser utilizado bem mais tarde.
- 5 • De acordo com o dicionário Michaelis, lógica é a análise das
6 formas e leis do pensamento, mas não se preocupa com a pro-
7 dução do pensamento, [...], sim com a forma, i.e., como o
8 pensamento é organizado e como chegamos a uma conclusão.
- 9
10 • Já um argumento ou ideia pode ser composto por uma ou várias
11 premissas, as quais podem ser Verdadeiras ou Falsas e condu-
12 zem à conclusão, que também poderá ser V ou F.

13
14
1. Do grego *logos* significa linguagem racional.

- 1 • No argumento a seguir, temos as premissas² (1 e 2) e em 3 a
2 conclusão:

3
4 </1> Sandra é mais velha do que Ricardo.

5 </2> Ricardo é mais velho do que Pedro.

6 </3> Logo, Sandra é mais velha do que Pedro.

7
8
9 </1> Pukki, meu cachorro, tem duas orelhas.

10 </2> Eu, seu dono, tenho duas orelhas.

11 </3> Logo, Eu não sou um cachorro.

12
13
14 2. Ideia que serve para a criação de um pensamento, opinião, ponto de vista, raciocínio; axioma, pressuposto. (<https://www.dicio.com.br/premissa/>).

1 • Os argumentos podem ser indutivos ou dedutivos (PUGA &
2 RISSETTI, 2013, grifo meu).

3
4 • Os argumentos indutivos:

5 São aqueles que, com base em dados, chega-se a uma resposta
6 por meio de uma analogia, *i.e.*, pela comparação com algo co-
7 nhecido. Esse tipo de raciocínio, contudo, não oferece certe-
8 za de que a resposta será de fato verdadeira. É necessário
9 conhecer os fatos ou as situações para que se possa fazer a
10 comparação (*ibidem*, grifo meu). Por exemplo:

11 </1> Ontem não havia nuvens no céu e não choveu.

12 </2> Hoje não há nuvens no céu.

13 </3> Portanto, hoje não vai chover.
14

1 • Já os argumentos dedutivos:

2 São aqueles cuja conclusão é obtida como consequência das
3 premissas, *i.e.*, por meio da análise das situações ou fatos
4 pode-se obter a resposta. Trabalha-se com a forma das senten-
5 ças, sem que haja necessidade do conhecimento prévio das si-
6 tuações ou fatos, ou seja, a conclusão é obtida em decorrên-
7 cia das premissas (*ibidem*, grifo meu). Por exemplo:

8 </1> Joana D'Arc era uma mulher.

9 </2> As mulheres são seres dominantes.

10 </3> Logo, Joana D'Arc era um ser dominante.



De modo geral:

</1> A é verdade de B.

</2> B é verdade de C.

</3> Portanto, A é verdade de C.

1.2. A lógica e áreas dos saberes

- A lógica é aplicada em diversas áreas, entre elas:



Informática



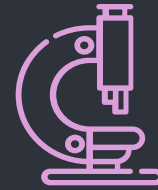
Engenharia



Psicologia



Direito



Física

• Etc., Etc., ...

1.3. A lógica na informática [...]

- 1
2 • Na informática/computação, aplica-se praticamente em todas
3 as áreas, que vai da construção ao pleno funcionamento de
4 *hardwares* e *softwares*. Principalmente, na construção de
5 circuitos integrados usando conceitos de portas lógicas pa-
6 ra a verificar a passagem ou não de pulsos elétricos, a fim
7 de estabelecer comunicação entre os componentes.
- 8 • Já na construção de *software*, por meio do raciocínio lógi-
9 co, construímos algoritmos que podem ser transformados em
10 programas capazes de solucionar problemas dos mais simples
11 aos mais complexos.

12
13
14

02 {

[*Hardware & Software*]

< 0 físico e o lógico >

}

2.1. Em busca por soluções [...]

- 1 • Desde o advento da era tecnológica, a busca por soluções que
2 simplifiquem tarefas e otimizem o tempo tem sido uma cons-
3 tante na história humana. Nesse contexto, o computador des-
4 tacou-se como uma ferramenta excepcionalmente versátil,
5 rápida e segura.
- 6 • Portanto, a função essencial é processar dados, que envolve
7 receber, manipular e armazenar informações. Isso é alcançado
8 por meio de programas projetados especificamente para execu-
9 tar tarefas de processamento de dados.

11 Coleta de dados



12 Processamento de
13 informações



14 Transmissão



Armazenagem
(*data storage*)



2.2. 0 computador [...]

- Um computador é composto por duas partes distintas que operam em conjunto, são elas:

Hardware < /1 > {

| [] < Partes físicas [...] >

}

Software < /2 > {

| [] < Parte lógica [...] >

}

03 {

[Algoritmizando]

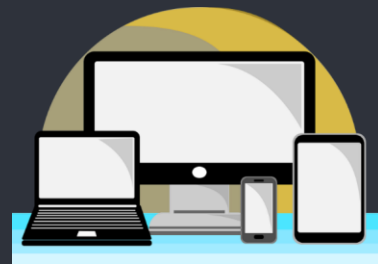
< Análises descritivas e Códigos >

}

3.1. 0 desenvolvimento [...]

- Quando desenvolvemos um *software* para executar um tipo específico de rotina, precisamos escrever códigos que podem ser compostos por várias partes interligadas.
- No entanto, para que o computador compreenda e execute esse programa, devemos escrevê-lo usando uma linguagem que tanto o computador quanto o desenvolvedor/criador entendam. Essa linguagem é chamada linguagem de programação.

```
Welcome C helloworld.c X
C helloworld.c > main()
1  #include <stdio.h>
2
3  int main(){
4      printf("Hello World");
5
6      return 0;
7  }
```



- As etapas para o desenvolvimento de um programa são:

Componentes do 'desenvolvimento' {

- Análise** • Estuda-se o problema para definir os dados de entrada, processamento e dados de saída.
- Algoritmo** • Ferramentas para descrição narrativa ou fluxograma são utilizadas para descrever o problema e possíveis soluções.
- Codificação** • O algoritmo é transformado em códigos da linguagem de programação.

}

- Portanto, um programa é a codificação de um algoritmo em uma linguagem de programação (ASCENCIO, 1999).

3.2. Conceito de algoritmo

```
1  O que é Algoritmo? {
2
3  < Ascencio & Campos (2014), apresentam alguns conceitos
4  de algoritmos >.
5
6  < /1 > * Algoritmo é uma sequência finita de instruções ou
7  operações cuja execução, em tempo finito, resolve
8  um problema computacional, qualquer que seja sua
9  instância" (SALVETTI, 1999).
10
11  < /2 > * É a descrição de uma sequência de passos que deve
12  ser seguida para a realização de uma tarefa"
13  (ASCENCIO, 1999)
14  < /3 > * É uma sequência de passos que visa atingir um
    objetivo bem definido" (FORBELLONE, 1999)
}
```


- Analisando as definições apresentadas, percebemos que diariamente executamos diversos algoritmos. Por exemplo:

Algoritmo 1 'Fazer um sanduíche' {

— Passo 1 – Pegar o pão.

— Passo 2 – Cortar o pão ao meio.

— Passo 3 – Pegar a maionese.

— Passo 4 – Passar a maionese no pão.

— Passo 5 – Pegar e cortar alface e tomate.

— Passo 6 – Colocar alface e tomate no pão.

— Passo 7 – Pegar o hambúrguer.

— Passo 8 – Fritar o hambúrguer.

— Passo 9 – Colocar o hambúrguer no pão.

}

```
1  Algoritmo 2 'Trocar uma lâmpada' {
2
3  — Passo 1 – Pegar uma lâmpada nova.
4  — Passo 2 – Pegar uma escada.
5  — Passo 3 – Posicionar a escada embaixo da lâmpada queimada.
6  — Passo 4 – Subir na escada com a lâmpada nova na mão.
7  — Passo 5 – Retirar a lâmpada queimada.
8  — Passo 6 – Colocar a lâmpada nova.
9  — Passo 7 – Descer da escada.
10 — Passo 8 – Testar o interruptor.
11 — Passo 9 – Guardar a escada.
12 — Passo 10 – Descartar a lâmpada queimada.
13 }
14
```

```
1  Algoritmo 3 'Sacar dinheiro no banco 24 horas' {
2
3  — Passo 1 – Ir até o banco 24 horas.
4  — Passo 2 – Colocar o cartão.
5  — Passo 3 – Digitar a senha.
6  — Passo 4 – Solicitar a quantia desejada.
7  — Passo 5 – Se o saldo for maior ou igual à quantia desejada sacar;
8                caso contrário, mostrar mensagem de indisponibilidade de
9                saque.
10 — Passo 6 – Retirar o cartão.
11 — Passo 7 – Sair do banco 24 horas.
12
13 }
14
```

3.3. Como construir um algoritmo?

- 1 • Passar construirmos um algoritmo é necessário seguir alguns
- 2 passos:
- 3 </1> Compreender o problema, destacando os pontos mais impor-
- 4 tantes.
- 5
- 6 </2> Definir os dados de entrada.
- 7
- 8 </3> Definir o processamento¹.
- 9 </4> Definir os dados de saída.
- 10 </5> Construir o algoritmo utilizando um dos tipos descritos
- 11 (próxima seção).
- 12 </6> Testar o algoritmo realizando simulações.
- 13
- 14 1. O processamento é responsável pela transformação dos dados de entrada em dados de saída.

3.4. Tipos de algoritmos

- Os três tipos mais utilizados de algoritmos são:

Descrição narrativa

< Consiste em analisar o enunciado do problema e escrever, utilizando uma linguagem natural (*e.g.*, Português), os passos a serem seguidos para a sua resolução. >

Vantagem

- Não é necessário aprender nenhum novo conceito.

Desvantagem

- A linguagem natural abre espaços para várias interpretações, o que pode dificultar a transcrição das informações para o algoritmo.

Fluxograma </2> {

< Consiste em analisar o enunciado do problema e escrever, utilizando símbolos gráficos predefinidos os passos a serem seguidos para sua resolução. >

Vantagem

- O entendimento de elementos gráficos é mais simples que o entendimento de textos.


Desvantagem

- Torna-se necessário aprender a simbologia, além disso, o algoritmo resultante não apresenta muitos detalhes, dificultando sua transcrição para um programa.


}

- 1 • Abaixo, os símbolos fundamentais mais utilizados na cons-
2 trução de fluxograma.

3

4  Indica o início e o fim do algoritmo.


5

6  Indica o sentido do fluxo de dados.
7 Conecta exclusivamente os símbolos ou blocos existentes.


8

9  Representa a entrada de dados.


10

11  Representa a saída de dados.

12

13  Indica tomada de decisão.
14 Aponta possibilidade de desvios.

15

16  Indica o sentido do fluxo de dados.
17 Serve exclusivamente para conectar os símbolos ou blocos existentes.

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

Pseudocódigo ou Portugol </3> {

< Consiste em analisar o enunciado do problema e escrever, por meio de regras predefinidas, os passos a serem seguidos para sua resolução. >

Vantagem

- A passagem do algoritmo para qualquer linguagem de programação é quase imediata, bastando conhecer as palavras reservadas da linguagem.

Desvantagem

- É necessário aprender as regras do pseudocódigo.

}

3.5. Alguns exemplos

- A seguir, alguns algoritmos desenvolvidos a partir dos três tipos citados anteriormente.

</1> Faça um algoritmo que mostre o resultado da multiplicação de dois números.

Algoritmo em descrição narrativa:

Exemplo 1 'Multiplicar dois números' {

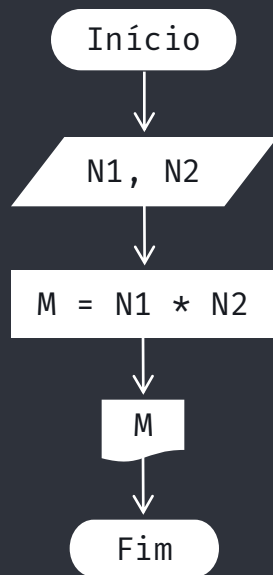
— Passo 1 – Receber dois números que serão multiplicados.

— Passo 2 – Multiplicar os números.

— Passo 3 – Mostrar o resultado obtido na multiplicação.

}

Fluxograma



Pseudocódigo

ALGORITMO

DECLARE N1, N2, M NUMÉRICO

ESCREVA "Digite dois números"

LEIA N1, N2

$M \leftarrow N1 * N2$

ESCREVA "Multiplicação = ", M

FIM_ALGORITMO.

1 </2> Faça um algoritmo que calcule a média aritmética entre
2 duas notas de um aluno e mostre sua situação, que pode ser
3 aprovado ou reprovado.
4

5
6 Algoritmo em descrição narrativa:

7 Exemplo 2 'Calcular média de duas notas [...]' {

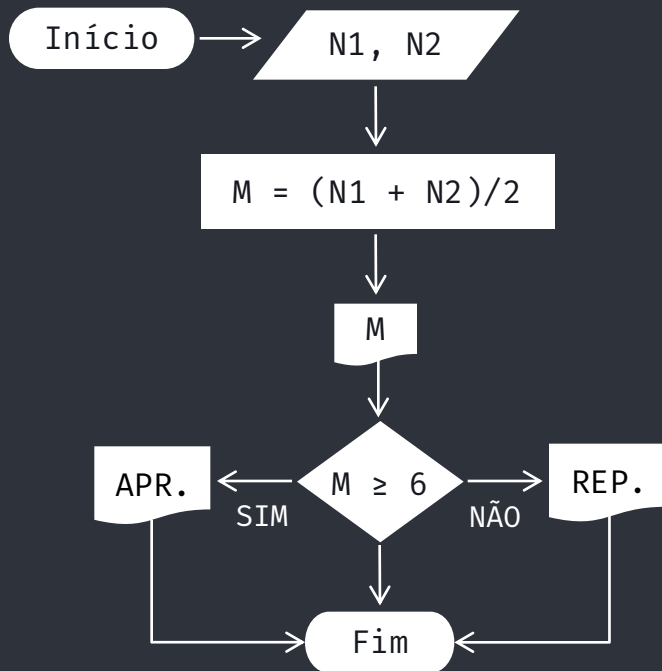
8 — Passo 1 – Receber as duas notas.

9 — Passo 2 – Calcular a média aritmética.

10 — Passo 3 – Mostrar a média aritmética.

11 — Passo 4 – SE a média aritmética for \geq a 6, ENTÃO aluno(a)
12 aprovado; SENÃO, aluno reprovado.
13 }
14

Fluxograma



Pseudocódigo

ALGORITMO

DECLARE N1, N2, M NUMÉRICO

ESCREVA "Digite as duas notas"

LEIA N1, N2

$M \leftarrow (N1 + N2) / 2$

ESCREVA "Média =", M

SE $M \geq 6$

ENTÃO EScreva "Aprovado"

SENÃO EScreva "Reprovado"

FIM_ALGORITMO.

1 </3> Faça um algoritmo para calcular o novo salário de um
2 funcionário. Sabe-se que os funcionários que recebem atual-
3 mente salário de até R\$ 500 terão aumento de 20%; os demais
4 terão aumento de 10%.

5
6 Algoritmo em descrição narrativa:

7
8 Exemplo 3 'Calcular novo salário' {

9 — Passo 1 – Receber o salário atual do funcionário.

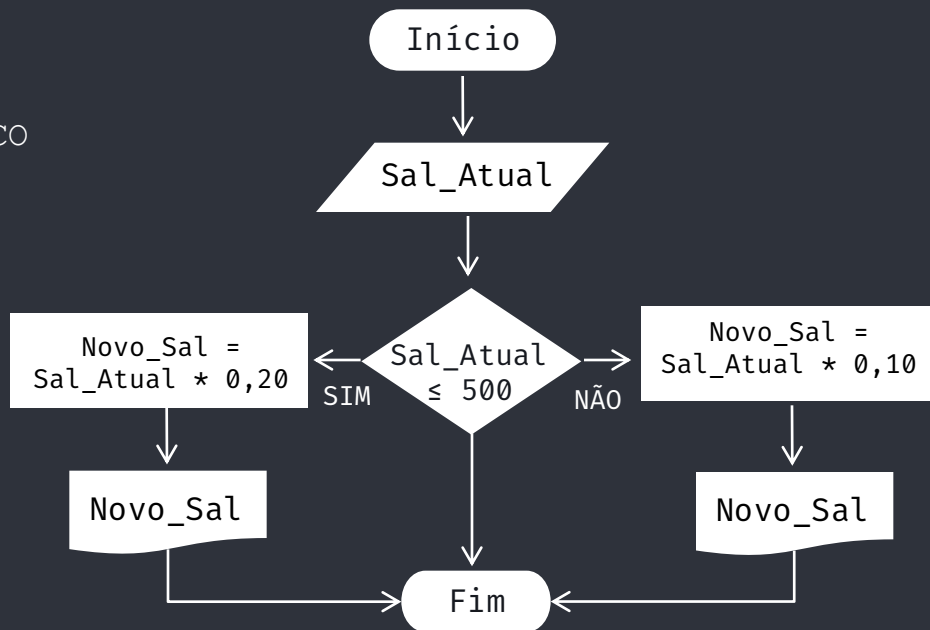
10
11 — Passo 2 – Se o salário atual do funcionário for de até R\$ 500,
12 calcular o novo salário com percentual de aumento de
13 20%; caso contrário, calcular o novo salário com per-
14 centual de aumento de 10%.

}

Pseudocódigo

```
1  
2  
3  
4 ALGORITMO  
5 DECLARE SAL_ATUAL, NOVO_SAL NUMÉRICO  
6 ESCREVA "Digite o salário atual do  
7 funcionário"  
8 LEIA SAL_ATUAL  
9 SE SAL_ATUAL ≤ 500  
10 ENTÃO NOVO_SAL = SAL_ATUAL * 0,20  
11 SENÃO NOVO_SAL = SAL_ATUAL * 0,10  
12 ESCREVA "Novo salário =", NOVO_SAL  
13 FIM_ALGORITMO.  
14
```

Fluxograma



1

2

3 </1> ASCENCIO, Ana Fernanda G. ARAÚJO, Grazielle. S. Estruturas de Dados:
4 algoritmos, análise da complexidade e implementações em Java e C/C++. São Paulo:
5 Pearson Prentice Hall, 2010. (Biblioteca virtual).

5

6 </2> PUGA, Sandra. RISSETTE, Gerson. Lógica de Programação e Estrutura de Dados.
7 2. ed. São Paulo? Pearson Prentice Hall, 2009.

7

8

9

10

11

12

13

14

A 'Obrigado' Is {

Algoritmo & Linguagem de Programação 1

|
}

