January 1986

Competências Mínimas na Programação em BASIC [Minimum Competencies for Programming in BASIC]

Karl M. Lorenz
Sacred Heart University

COMPETÊNCIAS MÍNIMAS NA PROGRAMAÇÃO EM BASIC

Karl Michael Lorenz*

O microcomputador, sem dúvida, está exercendo grande influência no cotidiano. Cada dia mais computadores estão sendo comprados e usados nos lares e em locais de trabalho. Em resultado, maior número de pessoas está aprendendo a programar em BASIC — a linguagem mais comum para os microcomputadores.

Em geral a aprendizagem de BASIC ocorre de duas maneiras. A primeira é mediante o esforço individual. Neste caso, o interessado consulta livros ou manuais e aprende a programar sozinho em sua própria casa. Aprendizagem é lenta e trabalhosa e não garante que a pessoa aprenda tudo aquilo que deve ser aprendido para poder programar de maneira visualizada pelos autores das obras consultadas. A segunda maneira, mais eficiente, porém da mesma forma trabalhosa, é através de um curso formal. Tal curso é dado por um instrutor que, com base em seus conhecimentos sobre o computador, a linguagem BASIC e a didática moderna, monta uma seqüência de atividades que facilita a aprendizagem para o aluno. Em ambas as modalidades de aprendizagem, a utilização de materiais didáticos bem elaborados, tais como livros, manuais e apostilas, são indispensáveis para que o aluno possa ter a orientação correta em suas tentativas de programar.

Fundamental para a preparação destes materiais — e os cursos montados com os mesmos — é a decisão sobre o que deve ser ensinado para que a pessoa interessada adquira, sozinha ou com a ajuda de um instrutor, a capacidade de aprofundar seu conhecimento através de estudo individual. Em outras palavras, antes de tudo é necessário que se determine quais as *competências mínimas* na programação em BASIC que um indivíduo deve adquirir ou demonstrar no final de suas sessões formais ou não-formais de aprendizagem.

A determinação das competências mínimas são de interesse tanto para o curriculista que elabora materiais didáticos, o instrutor que monta cursos de treinamento, o avaliador que julga o nível de rendimento do aluno, quanto para o próprio aluno que tem interesse em saber o progresso de sua aprendizagem.

Até o momento, não existe nenhum material divulgado que reuna as competências mínimas que devem ser utilizadas na elaboração de materiais didáticos ou cursos de programação em BASIC. Desta forma, este artigo pretende apresen-

* Ed.D. Professor visitante do Curso de Pós-Graduação em Educação da Universidade Federal do Paraná.

tar uma relação de competências mínimas que sirva para atividades variadas associadas com a aprendizagem da programação em BASIC. A lista é ainda provisória, porém serve como referência para futuras tentativas de montagem de listas mais completas e especializadas. No entanto, considera-se esta lista suficientemente abrangente para ser utilizada na montagem de materiais para cursos introdutórios de BASIC.

Deve-se salientar que a identificação das competências aqui apresentadas faz parte de um projeto em andamento financiado pela Fundação da Universidade Federal do Paraná para o Desenvolvimento da Tecnologia e da Cultura e registrado no Curso de Pós-Graduação em Educação da mesma Universidade. Tal projeto visa elaborar e testar um manual de programação em BASIC utilizando o método de redescoberta. As competências aqui apresentadas, e outras não incluídas devido o seu alto grau de especificidade e complexidade, serviram de subsídios para desenvolver e seqüenciar as atividades que constam do referido manual.

Do quadro que se segue constam 200 competências consideradas essenciais para desenvolver programas simples na linguagem BASIC. As competências estão divididas em 12 blocos e enumeradas apenas para facilitar sua identificação. A numeração não indica o grau de importância das competências individuais, nem tampouco a ordem de sua apresentação se encontra em uma seqüência de atividades para aprendê-las. Observe-se também, que não foi feita nenhuma discriminação entre competências cognitivas e comportamentais, ou entre competências simples e complexas. Qualquer análise e classificação subseqüente dependerá do leitor.

Finalizando, deve-se notar que foi feita uma tentativa de elaborar as competências de tal forma que nenhum dialeto de BASIC fosse favorecido. Desta maneira, as mesmas podem ser dominadas ou medidas com a utilização de qualquer dialeto. No entanto, não foi possível evitar o uso de termos específicos de determinado dialeto, sendo assim, algumas das competências requerem um conhecimento do dialeto Applesoft BASIC. Porém, estas poderão ser facilmente traduzidas em outro dialeto qualquer.

Espera-se que esta lista seja útil para as pessoas interessadas em aprender ou ensinar a programar em BASIC. Conforme já salientado, a lista engloba aquelas competências consideradas essenciais para o principiante desenvolver programas simples. No entanto, reconhece-se que várias competências não foram incluídas, por isso sugere-se que a mesma seja modificada ou ampliada segundo os critérios de cada leitor.

QUADRO 1 – COMPETÊNCIAS MÍNIMAS EM BASIC

O MICROCOMPUTADOR

1. Explicar o que é um computador.
2. Explicar o que é um microcomputador.
3. Explicar o que é *hardware.*
4. Explicar o que é *software.*
5. Explicar o que é um periférico.
6. Identificar os quatro componentes básicos de um microcomputador.
7. Descrever o teclado de um microcomputador.
8. Explicar como funciona a unidade de disco magnético.
9. Explicar como funciona a unidade de fita magnética.
10. Explicar a função do vídeo.
11. Explicar a função da unidade central de processamento (UCP).
12. Explicar o que é um microprocessador.
13. Explicar o que é RAM.
14. Explicar o que é ROM.
15. Explicar o que é um endereço de memória.
16. Explicar o que é um *bit.*
17. Explicar o que é um *byte.*
18. Explicar o que é um *kbyte.*
19. Explicar a função do cursor.

20. Mudar a posição do cursor para a direita e para a esquerda, para cima e para baixo na tela.
21. Ligar e desligar o microcomputador.
22. *Boot* o microcomputador.*

---

* Para as competências que seguem, o termo "computador" será usado ao invés do termo "microcomputador".

FUNDAMENTOS DA PROGRAMAÇÃO

23. Explicar o que é a linguagem BASIC.
24. Explicar o que é uma mensagem de erro.
25. Explicar o que é um programa de computador.
26. Explicar o que é um algoritmo.
27. Explicar o que faz um programador.
28. Explicar o que faz um usuário.
29. Explicar a diferença entre um comando e uma instrução.
30. Explicar o significado do número que precede uma linha de instrução num programa.
31. Explicar porque se enumera linhas de um programa com números de dez em dez, ou de cinco em cinco.
32. Executar um programa com o comando RUN.
33. Apagar um programa da memória do computador com o comando NEW.
34. Apagar uma linha de instrução de um programa.
35. Alterar uma linha de instrução de um programa.
36. Introduzir uma nova linha de instrução num programa existente.
37. Usar dois pontos para juntar múltiplas instruções numa linha de um programa.
38. Separar por um ponto e vírgula duas instruções numa linha de um programa para tornar mais fácil a leitura das mesmas.
39. Usar um ponto e vírgula no final de uma linha de instrução para juntar na tela do vídeo sua mensagem e a mensagem de uma outra linha de instrução que a precede num programa.
40. Escrever uma instrução REM.
41. Usar a instrução END para indicar o término da execução de um programa.
42. Listar um programa inteiro na tela do vídeo.
43. Listar uma linha de instrução ou um conjunto de linhas de instrução de um programa na tela do vídeo.
44. Armazenar um programa num disquete ou numa fita magnética.
45. Criar um catálogo de programas.
46. Transferir um programa de um disquete ou de uma fita magnética para a memória do computador.

VARIÁVEIS

47. Explicar o que é uma variável na linguagem BASIC.
48. Explicar a função de uma variável num programa de computador.
49. Escrever o símbolo para uma variável à qual é atribuído um valor numérico.
50. Explicar o que é um *string*.
51. Escrever o símbolo para uma variável à qual é atribuída um string.
52. Explicar a diferença entre uma variável numérica e uma variável string.
53. *Inicializar* uma variável simples.
54. Escrever o símbolo de uma variável com um subscrito.
55. Explicar o significado dos componentes do símbolo para uma variável com um subscrito.
56. Explicar a utilidade de uma variável com um subscrito.
57. *Inicializar* uma variável com um subscrito.
58. Explicar como o computador armazena valores numéricos e strings em seus locais de memória quando é utilizada uma variável com um subscrito num programa.
59. Explicar o que é um *Array*.
60. Explicar a função de uma instrução DIM.
61. Explicar o significado dos componentes de uma instrução DIM.
62. Escrever uma instrução DIM para uma variável com um subscrito.
63. Escrever o símbolo de uma variável com dois subscritos.
64. Explicar o significado dos componentes do símbolo para uma variável com dois subscritos.
65. Explicar a utilidade de uma variável com dois subscritos.

66. *Inicializar* uma variável com dois subscritos.
67. Explicar como o computador armazena valores numéricos e strings em seus locais de memória quando é utilizada uma variável com dois subscritos num programa.
68. Escrever uma instrução DIM para uma variável com dois subscritos.

ATRIBUIÇÃO DE VALORES À VARIÁVEIS

69. Explicar a estrutura de uma instrução LET.
70. Atribuir um valor numérico a uma variável com uma instrução LET.
71. Atribuir um string a uma variável com uma instrução LET.
72. Explicar a estrutura de uma instrução INPUT.
73. Atribuir um valor numérico a uma variável com uma instrução INPUT.
74. Atribuir um string a uma variável com uma instrução INPUT.
75. Atribuir valores numéricos ou strings a mais de uma variável com uma única instrução INPUT.
76. Escrever instruções claras e sucintas na tela do vídeo para orientar o usuário quanto ao tipo de dados que ele deve dar entrada através de uma instrução INPUT.
77. Explicar como funciona em conjunto as instruções READ e DATA.
78. Explicar a estrutura da instrução READ e da instrução DATA.
79. Atribuir um valor numérico a uma variável com as instruções READ e DATA.
80. Atribuir um string a uma variável com as instruções READ e DATA.
81. Atribuir mais de um valor a uma variável com as instruções READ e DATA.
82. Usar a instrução RESTORE juntamente com as instruções READ e DATA num programa.
83. Atribuir valores numéricos ou strings a uma variável com um subscrito.
84. Atribuir valores numéricos ou strings a uma variável com dois subscritos.
85. Desenvolver um programa que utilize uma ou mais variáveis com um subscrito.
86. Desenvolver um programa que utilize uma ou mais variáveis com dois subscritos.

LAÇOS

87. Explicar o que é um laço infinito.
88. Usar uma instrução GOTO para executar uma linha de instrução mais adiante num programa.
89. Usar uma instrução GOTO para executar uma linha de instrução anterior num programa.
90. Desenvolver um programa em que um laço FOR-NEXT limite o número de vezes em que o computador repete a execução de determinada instrução indicada numa instrução GOTO.
91. Explicar a estrutura e a função de um laço FOR-NEXT.
92. Usar um laço FOR-NEXT para imprimir um conjunto de mensagens na tela.
93. Usar num laço FOR-NEXT uma instrução FOR na qual o valor mais baixo da variável de controle seja um número maior do que um (1).
94. Usar num laço FOR-NEXT uma instrução FOR no qual um ou ambos dos valores da variável de controle sejam negativos.
95. Atribuir um ou ambos os valores à variável de controle de uma instrução FOR com uma ou duas instruções LET.
96. Atribuir um ou ambos os valores à variável de controle de uma instrução FOR com uma ou duas instruções INPUT.
97. Atribuir um ou ambos os valores à variável de controle de uma instrução FOR com as instruções READ e DATA.
98. Usar num laço FOR-NEXT uma instrução FOR que contenha uma instrução STEP para incrementos positivos.
99. Usar num laço FOR-NEXT uma instrução FOR que contenha uma instrução STEP para incrementos negativos.
100. Usar um laço FOR-NEXT para causar uma pausa na geração de mensagens na tela.
101. Explicar a estrutura de laços embutidos.
102. Explicar a utilidade de laços embutidos.
103. Desenvolver um programa que utilize laços embutidos.

DECISÕES

104. Explicar a função da instrução IF...THEN.
105. Identificar os componentes da instrução IF...THEN.
106. Explicar o processo decisório quando o computador executa uma instrução IF...THEN.
107. Explicar o significado de cada um dos operadores relacionados usados na instrução IF...THEN.
108. Usar uma instrução IF...THEN GOTO para comparar dois valores numéricos ou dois strings.
109. Desenvolver um programa que contenha várias instruções IF...THEN GOTO.
110. Usar uma instrução IF...THEN PRINT para comparar dois valores numéricos ou dois strings.

111. Desenvolver um programa que contenha várias instruções IF...THEN PRINT.
112. Usar uma instrução IF...THEN PRINT TAB para comparar dois valores numéricos ou dois strings.
113. Usar uma instrução IF...THEN para somar valores.
114. Usar uma instrução IF...THEN para contar de 1 até algum valor pré-determinado.

CÁLCULOS

115. Citar a ordem em que as quatro operações aritméticas são executadas pelo computador.
116. Citar a ordem em que diversas operações aritméticas são executadas pelo computador quando algumas delas estão dentro de parânteses.
117. Mandar o computador efetuar um cálculo que envolva várias operações aritméticas, algumas das quais dentro de parênteses.
118. Mandar o computador efetuar o cálculo que envolva várias operações na ordem desejada.
119. Mandar o computador efetuar os cálculos de uma fórmula de maneira correta.
120. Indicar os símbolos para as quatro operações aritméticas realizadas pelo computador.

FUNÇÕES ARITMÉTICAS

121. Usar a função apropriada para calcular a raiz quadrada de um número.
122. Usar a função apropriada para obter o valor absoluto de um número.
123. Usar a função apropriada para calcular o valor exponencial de um número.
124. Usar a função apropriada para obter o logarítmo natural de um número.
125. Usar a função RND para gerar um número randômico que assume um valor entre parênteses pré-determinados, tais como 1 e 10, ou 1 e 100, etc.
126. Usar a função INT para converter um número com decimal em um número inteiro.
127. Usar a função INT junto com a função RND para gerar um número randômico inteiro.

FUNÇÕES STRING

128. Explicar na expressão LEN(X$) o significado da variável dentro de parânteses.
129. Explicar a utilidade da função LEN$.
130. Usar a função LEN$ para contar o número de caracteres de um string.
131. Desenvolver um programa simples em que a função LEN$ limite o número de caracteres de uma expressão ou de uma palavra fornecida pelo usuário.
132. Usar a função LEN$ juntamente com a função apropriada num programa em que uma expressão ou uma palavra é impressa no centro de uma linha.
133. Usar a função LEN$ juntamente com a função apropriada num programa em que uma expressão ou palavra é impressa na parte central da tela.
134. Desenvolver um programa que contenha várias funções LEN$.
135. Explicar na expressão MID$(X$,Y,Z) o significado das variáveis dentro de parênteses.
136. Explicar para que serve a função MID$.
137. Usar a função MID$ num programa para identificar um ou mais caracteres de um string.
138. Desenvolver um programa em que várias funções MID$ identifiquem os caracteres dentro dos strings correspondentes.
139. Explicar quando se usa uma expressão MID$ com duas variáveis dentro de parênteses tal como MID$(X$, Y).
140. Usar num programa uma função MID$ com a forma da expressão MID$(X$, Y).
141. Desenvolver um programa que contenha mais do que uma função MID$.
142. Explicar para que serve a função LEFT$.
143. Explicar para que serve a função RIGHT$.
144. Explicar na expressão LEFT$(X$, Y) o significado das variáveis dentro de parênteses.
145. Explicar na expressão RIGHT$(X$, Y) o significado das variáveis dentro de parênteses.
146. Usar a função LEFT$ para identificar determinado(s) caractere(s) de um string.
147. Usar a função RIGHT$ para identificar determinado(s) caractere(s) de um string.
148. Usar as funções LEFT$ e RIGHT$ juntas num programa.
149. Usar a função LEFT$ num programa em que o computador imprime na tela um string que começa com determinado(s) caractere(s).
150. Usar a função RIGHT$ num programa em que o computador imprime na tela um string que termina com determinado(s) caractere(s).
151. Usar a função LEFT$ num programa para identificar números que começam com determinada(s) cifra(s).
152. Usar a função RIGHT$ num programa para identificar um ou mais números que contenham determinada(s) cifra(s).

153. Explicar para que serve a função VAL.
154. Explicar na expressão VAL(X$) o significado da variável dentro de parênteses.
155. Usar a função VAL num programa para identificar valores numéricos contidos num string.
156. Usar a função VAL num programa em que é controlada a entrada de dados fornecidos pelo usuário.
157. Explicar para que serve a função STR$.
158. Explicar na expressão STR$(X) o significado da variável dentro de parênteses.
159. Desenvolver um programa simples que contenha uma ou mais funções STR$.
160. Explicar o sistema de codificação ASCII.
161. Explicar para que serve a função STR$.
162. Explicar na expressão CHR$(X) o significado da variável dentro de parênteses.
163. Indicar quais os números códigos ASCII para as letras do alfabeto.
164. Reconhecer os números códigos ASCII para caracteres que não são do alfabeto.
165. Desenvolver um programa que contenha uma ou mais funções CHR$.
166. Desenvolver um programa simples em que o usuário determine o número código ASCII da função CHR$.
167. Explicar para que serve a função ASC.
168. Explicar na expressão ASC(X$) o significado da variável dentro de parânteses.
169. Desenvolver um programa simples que contenha uma ou mais funções ASC.

FUNÇÕES GRÁFICAS

170. Explicar a diferença entre uma saída de resolução alta e uma saída de resolução baixa.
171. Usar a função apropriada para mandar o computador entrar no modo gráfico.
172. Usar a função apropriada para mandar o computador sair do modo gráfico.
173. Usar a função apropriada para imprimir um texto abaixo de um gráfico na tela.
174. Usar a função apropriada para imprimir um ponto em baixa resolução na tela do vídeo.
175. Usar a função apropriada para imprimir em determinada cor uma mensagem ou gráfico na tela.

ORGANIZAÇÃO DA SAÍDA

176. Citar as dimensões da tela do vídeo, isto é, o número de colunas e de linhas de saída possíveis.
177. Enumerar corretamente as colunas da tela.
178. Indicar o número de caracteres que podem ser escritos numa linha de saída.
179. Usar a função apropriada (VTAB, etc.) para iniciar a impressão de uma mensagem em determinada linha da tela.
180. Usar a função apropriada (HTAB, TAB, etc.) para iniciar a impressão de uma mensagem em determinada coluna da tela.
181. Usar as funções apropriadas (VTAB e HTAB, etc.) em conjunto para iniciar a impressão de uma mensagem em determinada coluna e em determinada linha da tela.
182. Indicar o número de zonas em que a tela é dividida.
183. Indicar as dimensões de cada zona da tela.
184. Enumerar corretamente as colunas em cada zona da tela.
185. Usar a vírgula numa instrução para iniciar a impressão de uma mensagem na primeira coluna de determinada zona na tela.
186. Usar mais que uma vírgula numa instrução para iniciar a impressão de mensagens nas primeiras colunas das zonas correspondentes.

SUB-ROTINAS

187. Explicar o que é uma sub-rotina.
188. Explicar qual é a utilidade de uma sub-rotina.
189. Descrever a estrutura geral de um programa que contém uma ou mais sub-rotinas.
190. Explicar a estrutura e a função de uma instrução GOSUB.
191. Explicar a função da instrução RETURN dentro de uma sub-rotina.
192. Descrever a estrutura de uma sub-rotina.
193. Desenvolver um programa simples que contenha uma ou mais sub-rotinas.
194. Desenvolver um programa simples em que o usuário decida se vai ou não usar uma sub-rotina mais de uma vez.
195. Desenvolver um programa simples em que o usuário decida se uma sub-rotina vai ser executada ou não.
196. Desenvolver um programa simples em que uma das instruções de uma sub-rotina mande executar outra sub-rotina.
197. Explicar o que é um menu.
198. Desenvolver um programa simples no qual são executadas várias sub-rotinas com as instruções ON...GOSUB correspondentes.
199. Explicar a diferença entre as instruções ON...GOSUB e ON...GOTO.
200. Desenvolver um programa simples que contenha instruções ON...GOTO.