

LUÍS DE CAMPOS

# COMPUTADOR ATARI

Manual Completo do Utilizador



TEMPOS  
LIVRES





## CULTURA E TEMPOS LIVRES

1. ABC do Xadrez, *Petar Trifunovitch e Sava Vukovitch*
4. ABC do Bridge, *Pierre Jais e H. Lahana*
5. Guia Prático de Fotografia, *W. D. Emanuel*
6. ABC do Judo, *E. J. Harrison*
7. Como Fazer Cinema, *Paul Petzold*
8. Bridge Moderno, *Pierre Jais e H. Lahana*
9. Fotografia – Técnicas e Truques I, *Edwin Smith*
10. ABC dos Estilos, da Arquitectura ao Mobiliário, *A. Ausel*
11. Fotografia – Técnicas e Truques II, *Edwin Smith*
12. A Pesca Submarina, *António Ribera*
13. Teoria dos Finais de Partida, *Yuri Averbach*
14. Aprenda Rádio, *B. Fighiera*
15. Guia do Cão, *Louise Laliberté-Robert e Jean-Pierre Robert*
16. ABC do Aquário, *Anthony Evans*
17. Iniciação à Electricidade e Electrónica, *Fernand Huré*
18. Os Transístores, *Fernand Huré*
19. Karaté I, *Albrecht Pflüger*
20. Iniciação ao Radiocomando dos Modelos Reduzidos, *C. Péricono*
21. Construa o seu Receptor, *B. Fighiera*
22. Montagens Electrónicas, *B. Fighiera*
23. O Berbequim Eléctrico, *Villy Dreier*
24. Cactos, *J. Nilaus Jensen*
25. Iniciação à Alta Fidelidade, *Peter Turner*
26. O Aquário de Água Doce, *Paulo de Oliveira*
27. ABC do Ténis, *Fonseca Vaz*
28. Karaté II, *Albrecht Pflüger*
29. ABC da Criação de Canários, *Curt Af Enehjelm*
30. Ginástica Feminina, *Sonja Helmer Jensen*
31. Cartomancia, *Rhea Koch*
32. Calculadoras Electrónicas de Bolso, *E. Dam Ravn*
33. O Pastor Alemão, *Gilles Legrand*
34. Xadrez – Teoria do Meio Jogo I, *I. Bondarevsky*
35. Manual do Super 8 – I, *Myron A. Matzkin*
36. ABC da Criação de Periquitos, *Cyril H. Rogers*
37. O Livro dos Gatos, *Barbel Gerber e Horst Bielfeld*
38. Manual do Super 8 – II, *Myron A. Matzkin*
39. ABC do Mergulho Desportivo, *Walter Mattes*
40. Circuitos Integrados/Aplicações Práticas, *F. Bergtold*
41. A Apicultura, *H. R. C. Riches*
42. ABC do Cultivo das Plantas, *H. G. Witham Fogg*
43. ABC da Criação de Pombos, *Kai R. Dahl*
44. Construção de Caixas Acústicas de Alta Fidelidade, *R. Brault*
45. Raças de Canários, *Klaus Speicher*
46. Jogos de Cartas, *Graciano Dolma*
47. Cocker Spaniels, *H. S. Lloyd*
48. ABC da Caça, *Fabian Abril*
49. Aprenda Televisão, *Gordon J. King*
50. Iniciação à Pesca, *Juan Nadal*
51. Basquetebol, *Marius Norregard*
52. Cães de Caça, *Santiago Pons*
53. Aprenda Electrónica, *T. L. Squires e C. M. Deason*
54. A Avicultura, *Jim Worthington*
55. A Produção de Coelho, *P. Surdeau e R. Hénaff*
56. ABC dos Computadores, *T. F. Fry*
57. Natação para Crianças, *John Idorn*
58. O Boxer, *Anni Mortensen*
59. Voleibol, *Ole Hansen e Per-Göran Persson*
60. Iniciação à Vela, *Donald Law*
61. ABC da Filatelia, *Jacqueline Caurat*
62. A Pesca à Beira-Mar, *J.-M. Boëlle e D. Doyen*
63. Enxerto de Árvores de Fruto, *Alejo Rigau*
64. A Cultura do Morangueiro, *Luís Alsina Grau*
65. Emissores-Receptores (Walkies-Talkies), *P. Duranton*
66. Iniciação à Fotoelectrónica, *Heinz Richter*
67. Doces e Conservas de Fruta, *Robin Howe*
68. A Criação de Hamsters, *C. F. Snow*
69. A Criação de Porcos, *Roy Genders*
70. Calendário do Horticultor, *Luís Alsina Grau*
71. Jogos Electrónicos, *F. G. Rayer*
72. Cultivo de Cogumelos e Trufas, *Alejo Rigau*

73. Aprenda Televisão a Cores, *Gordon J. King*
74. Gravação em Fita Magnética, *Ian R. Sinclair*
75. Poda de Árvores e Arbustos, *Roy Genders*
76. Como Treinar o Seu Cão, *E. Fitch Duglish*
77. Instrumentos de Medida e Verificação, *Heinrich Stöckle*
78. A Criação de Caracóis, *Matias Josa*
79. Rádio - Fundamentos e Técnicas, *Gordon J. King*
80. Como Fazer Gelados, *Sylvie Thiébaul*
81. Iniciação à Jardinagem, *Noel Clarasó*
82. A Congelação dos Alimentos, *Suzanne Lapointe*
83. Windsurf - Prancha à Vela, *Ernstfried Prade*
84. Raças de Cães, *O. Hasselfeldt*
85. Rummy e Canasta, *Claus D. Grupp*
86. A Encadernação, *Annie Persuy*
87. Aprenda Electricidade, *Heinz Richter*
88. Taxidermia, Embalsamamento de Aves e Mamíferos, *Harry Hjortaa*
89. Jogging - Correr para Manter a Forma, *Werner Sonntag*
90. ABC da Cozinha Chinesa, *Sonya Richmond*
91. Jogos TV, *C. Tavernier*
92. Amplificadores de Som, *Richard Zierl*
93. O Livro do Poker, *Claus D. Grupp*
94. Aprenda a Desenhar, *Rose-Marie de Prémont e Nicoie Philippi*
95. O Minitrampolim na Escola, *Sonja Helmer Jensen e Klaus Danø*
96. Jogos de Luzes e Efeitos Sonoros para Guitarras, *B. Fighiera*
97. O Cultivo do Tomate, *Louis N. Flawn*
98. Pilhas Solares, *F. Juster*
99. A Criação Doméstica de Coelho, *C. F. Snow*
100. Iniciação ao Futebol, *Wieland Männle e Heinz Arnold*
101. Horóscopos Chineses, *Georg Haddenbach*
102. Guia Prático de Marcenaria, *Charles H. Hayward*
103. Anebol, *Fritz e Peter Hattig*
104. Dispositivos Anti-Roubo, *H. Schreiber*
105. Perus, Pintadas e Codornizes, *Jérôme Sauze*
106. Crepes - Doces e Salgados, *Florence Arzel*
107. Aperitivos e Entradas, *Myrette Tiano*
108. Tênis de Mesa, *Leslie Woollard*
109. Aprenda Surf, *R. Abbott e M. Baker*
110. Futebol - Técnica e Tática, *Kurt Lavall*
111. A Vaca Leiteira, *Colin T. Whittemore*
112. O Cubo Mágico, *Josef Trajber*
113. O Perdigueiro Português, *José M. Correia*
114. Pizzas e Massas à Italiana, *Marieanne Ränk*
115. O Cubo Para Quem Já o Faz, *Josef Trajber*
116. A Pirâmide Mágica, A Torre, O Barril do Diabo, *M. Mrowka-W. J. Weber*
117. Gansos e Patos, *Marie Mourthe*
118. Iniciação ao Kung-Fu, *A. P. Harrington*
119. Electrónica e Fotografia, *Hans-Peter Siebert*
120. O Livro da Fortuna, *Douglas Hill*
121. Construção de um Alimentador de Corrente, *Waldemar Baittinger*
122. Hóquei em Patins, *Francisco Velasco*
123. Técnicas de Tiro, *Anton Kovacic*
124. Aprenda a Tricotar, *Uta Mix*
125. ABC da Patinagem, *Christa-Maria e Richard Kerler*
126. A Pesca e os seus Segredos, *Armand Deschamps*
127. O Osciloscópio, *R. Rateau*
128. Guia Prático da Banda do Cidadão, *T. M. Normand*
129. Sumos e Batidos, *Manfred Donderski*
130. Introdução à Programação de Microcomputadores, *Peter C. Sanderson*
131. Aprenda Croché, *Uta Mix*
132. ABC do Microprocessador, *P. Mélusson*
133. Guia Prático de Basic, *Roger Hunt*
134. Introdução à Electrónica Digital, *Ian R. Sinclair*
135. ABC do Vídeo, *David Matthewson*
136. Fotografia em Movimento, *Don Morley*
137. Guia de Cobol, *Ray Welland*
138. Fotografia a Pequena Distância, *Sidney F. Ray*
139. Guia Moderno da Canaricultura, *Manuel Gonçalves*
140. Minielectrónica para Amadores, *Heinz Richter*
141. ABC da Programação de Computadores, *John Shelley*
142. Tarot - O Futuro Pelas Cartas, *Edwin J. Nigg*
143. ABC da Equitação, *Dorothy Johnson*
144. Como Programar o seu ZX 81, *Patrick Gueulle*
145. 100 Avarias TV e a Maneira Prática de as Detectar, *P. Duranton*
146. ABC da Horticultura, *Louis Giordano*

147. Basic Para Microcomputadores, *A. P. Stephenson*
148. Como Programar o seu ZX Spectrum, *Tim Hartnell e Dilwyn Jones*
149. Iniciação aos Motores Diesel, *David S. Maclean*
150. 60 Jogos Para o ZX Spectrum, *David Harwood*
151. As Linhas da Mão, *Rose Hubert*
152. Cozinha Italiana, *Rotraud Degner*
153. Manual do ZX Spectrum, *R. J. Simpson e T. J. Terrell*
154. Z80 Assembler Para o ZX Spectrum – Iniciação ao Código de Máquina, *João Paulo Fragoso*
155. Aeróbica, *Hans Schulz*
156. ABC do Atletismo, *Denis Watts*
157. 26 Programas Basic Para Microcomputadores, *Derrick Daines*
158. Aprenda Pascal no Seu Microcomputador, *Jeremy Ruston*
159. Guia Moderno da Suinicultura, *Colin Whittemore*
160. O Bar em Sua Casa – 888 Cocktails, *Aladar von Wesendonk*
161. Código de Máquina Para Principiantes, *James Walsh*
162. Código de Máquina Para Programadores Avançados, *Paul Holmes*
163. ABC da Fruticultura, *Henri Gosselin*
164. ABC da Canoagem, *Alan Bye*
165. Guia de Fortran, *Philip Ridler*
166. Manual da Secretária, *Phillippa Ramage*
167. ABC das Antenas, *Gordon J. King*
168. Programar Aventuras no Seu Computador, *Andrew Nelson*
169. Guia do Sinclair QL, *Boris Allan*
170. Novas Aventuras no Seu ZX Spectrum, *Peter Shaw e James Mortleman*
171. O Computador no Escritório, *John Shelley*
172. Sobremesas, *Fred Timber*
173. Rádio – Do Circuito Oscilante ao Receptor de Ondas Curtas, *Richard Zierl*
174. Xadrez: ABC das Aberturas, *V. N. Panov*
175. A Construção de Pequenos Transformadores, *M. Dorian e F. Juster*
176. Guia de Pascal, *David Watt*
177. O ZX Spectrum na Educação, *Tim Hartnell, Christine Johnson e David Valentine*
178. Iniciação ao Snooker, *John Pulman*
179. Circuitos com Diacs, Tiristores e Triacs, *Fritz Bergtold*
180. O Jogo do Mah-Jong, *Ursula Eschenbach*
181. Cultura Hidropónica, Culturas sem Solo, *Laura Fronty*
182. 49 Jogos Explosivos para o ZX Spectrum, *Tim Hartnell, D. Perry, G. Charlton, N. Pellinacci e M. Young*
183. Iniciação ao Golfe, *J. C. Jessop*
184. Como se Constrói um Receptor de FM, *Richard Zierl*
185. Iniciação à Navegação Marítima, *Anselmo Vieira*
186. Processamento de Dados, *T. F. Fry*
187. O ZX Spectrum na Gestão de Pequenas Empresas, *Luis de Campos*
188. Programar Grafismos no seu Computador, *Michel Rousselet*
189. O IBM PC e Sistemas Compatíveis, *Carlos Reis e João Capaz*
190. Antenas para a Banda do Cidadão, *Patrick Gueulle*
191. Paciências, *Irmgard Wolter*
192. Defesa Pessoal, *Syd Hoare*
193. Processamento de Texto, *Lew Hollerbach*
194. Inteligência Artificial no Seu Spectrum e Spectrum +, *Tim Hartnell*
195. Manual do Correspondente de Inglês, *W. G. Tackle e Gomes Pitta*
196. Como Interpretar um Balanço, *Carlos Nabais*
197. Iniciação à Tecnologia da Informação, *Garry Marshall*
198. Manual de Programação Avançada do ZX Spectrum, *John Lettice*
199. O Espargo, *Mário F. Bento Ripado*
200. Computador Atari, Manual Completo do Utilizador, *Luis de Campos*

LUÍS DE CAMPOS

**COMPUTADOR  
ATARI**  
Manual Completo  
do utilizador

EDITORIAL  PRESENÇA



## FICHA TÉCNICA

Título: *COMPUTADOR ATARI – Manual Completo do Utilizador*

Autor: *Luís de Campos*

© Copyright, Lisboa, 1985, Luís de Campos e Editorial Presença, Lda.

Capa: *Fotografia gentilmente cedida pela Triudus*

Arranjos Gráficos: *Sector Gráfico da Editorial Presença*

Fotocomposição: *Nova Força – Lisboa.*

Impressão e Acabamento:

1.<sup>a</sup> edição, Lisboa, 1986

Reservados todos os direitos

para a língua portuguesa à  
EDITORIAL PRESENÇA, LDA.

Rua Augusto Gil, 35-A – 1000 LISBOA

## INTRODUÇÃO

Após o lançamento dos modelos 400 e 800 – por fins dos anos 70 – a Atari teve uma quebra (à qual não terá sido estranho o aparecimento de computadores muito mais evoluídos e baratos).

Na verdade, esses dois modelos eram basicamente máquinas de videogames com características avançadas (aliás, a Atari entrou nos computadores através de um equipamento de videogames, de facto sem concorrência séria, em meados dos anos 70). Muito pobres na definição gráfica, com memórias limitadas e um teclado mais parecido com o de uma máquina de escrever do que com o de um computador, era óbvia a sua desvantagem relativamente aos que surgiram em 1979/1980.

Num esforço para acompanhar essa avançada concorrência – mas não podendo deixar de aproveitar os milhares de componentes ainda em *stock* de fábrica – a Atari lançou o modelo 1200XL o qual, para além dos 64K de memória, apresentava uma caixa inteiramente diferente e um teclado com uma série de funções já nitidamente pertencentes ao reino dos micros. Também este, contudo, não era ainda o computador certo para concorrer com os demais.

Assim, num salto qualitativo importante e decisivo, a Atari lançou os modelos 600XL e 800XL – estes sim, iniciadores de uma série verdadeiramente nova.

O 600XL com 16K e o 800XL com 64K, ambos com um Basic mais poderoso (o qual pode, de qualquer modo, ser utilizado nos modelos antigos, com uma cartridge especial), além de 11 modos gráficos e 5 modos para texto, são computadores que, graças também ao seu baixo preço, colocam finalmente a Atari em igualdade de concorrência com os outros

micros do mercado (exceptuando talvez o Spectrum que, ao que parece, continua a não ter quem o destrone do invejável lugar que ocupa em volume de vendas – cerca de 500 000 unidades no ano passado).

Em Portugal foi este ano lançado, com o apoio de uma forte campanha publicitária, o modelo 800XL. Infelizmente (e uma vez mais) não devidamente acompanhado de software suficiente que permitisse uma segurança na opção de compra. Sem resultados ainda conhecidos, é de crer que a Atari venha a conquistar um lugar importante no nosso mercado, tendo até em conta o lançamento em Outubro, nos Estados Unidos e Canadá, dos modelos 130 XE e 520ST e estando previsto para breve o dos novos modelos 1400XL e 1450XL.

Este livro, sendo embora aplicável a todos os modelos da Atari, é escrito fundamentalmente para a série XL.

## OS COMPUTADORES ATARI

### 1.1. Modelos e características

Como se disse na introdução, a Atari lançou já no mercado cinco modelos de computadores, no que poderão considerar-se duas séries: uma primeira englobando os modelos 400, 800 e 1200XL; uma segunda, actualmente em produção, com os modelos 600XL e 800XL.

Os primeiros três deixaram já de ser fabricados, tendo a venda dos actuais sido iniciada em 1983. E, se bem que fossemos tentados a dividi-los em série simples, por um lado, e série XL por outro, o facto é que o 1200XL foi antes uma antecipação muito limitada aos 600XL e 800XL, estes sim inteiramente renovados.

Quanto aos modelos 1400XL e 1450XL, o seu projecto encontra-se ainda em fase de acabamento, nada se conhecendo, portanto, neste momento, além das características anunciadas, pela Atari.

Para que melhor possam comparar-se os 7 modelos, a página seguinte apresenta um quadro comparativo, no qual podem encontrar-se as características fundamentais.

Os primeiros modelos (400 e 800), possuíam grandes limitações de vária ordem, sobretudo o 400 que nem sequer tinha um teclado móvel (era de membrana como o ZX81 da Sinclair), para além de o seu preço ser demasiado elevado por comparação com micros similares.

Quanto ao 1200XL... não vale a pena perder muito tempo com ele pois nem a própria Atari o produziu para além de alguns meses – tempo apenas para o desenvolvimento do 600XL e 800XL.



QUADRO COMPARATIVO DOS COMPUTADORES «ATARI»

Características	400	800	1200XL	600XL	800XL	1400XL	1450XL
	(cessou fabrico)			(no mercado)		(planeados)	
Memória	16K	48K	64K	16K	64K	64K	64K
Teclado máq. esc.	*	*		*	*	*	*
Nova caixa			*	*	*	*	*
Teclas program.			*			*	*
Basic incluído				*	*	*	*
Saída p/ monitor	*	*			*	*	*
Ficha p/ telefone						*	*
Sint. de fala						*	*
Drive p/ disco							*
Modos gráficos	6-9	6-9	11	11	11	11	11
Modos texto	3	3	5	5	5	5	5

Todavia, todos os modelos apresentados até agora são basicamente variações do mesmo sistema, o que acaba por ser um bem para os possuidores de um qualquer modelo da Atari. De facto, e pondo de parte alguns incrementos (11 modos gráficos, 5 modos de texto, Basic implementado), mesmo que o leitor possua um dos velhos modelos 400 ou 800, com uma cartridge adquirida poderá, no essencial, tornar esses micros compatíveis com os novos.

Olhando, todavia, para o quadro comparativo, estranha-se, pelo menos, a razão que terá levado a Atari a projectar, fabricar e lançar no mercado os modelos 600XL e 800XL, cuja única



diferença, além dos 16K e 64K, é a de o segundo possuir uma saída para monitor, enquanto o primeiro não a tem.

Na verdade, o projecto baseou-se no 800XL, tendo o 600XL sido posteriormente «projectado» como um computador destinado ao grande público – nomeadamente àqueles que limitam o uso do computador à utilização de software comprado – podendo, graças ao baixo preço, competir com os demais. Assim, é fácil a quem tenha adquirido um 600XL transformá-lo num 800XL: basta adquirir um ou dois elementos de expansão (a saída para monitor não põe qualquer problema, podendo ser implementada por qualquer técnico de electrónica qualificado). Um 600XL com uma expansão de memória vem a ser, para todos os efeitos, um 800XL.

Embora basicamente sejam 800XL, o facto é que os futuros 1400XL e 1450XL acabam por possuir algumas novas características importantes e interessantes: *modem* telefónico incorporado, sintetizador de fala e um drive para diskette no modelo 1450XL.

De qualquer modo, é certo que, uma vez mais, adquirindo os periféricos respectivos poderá transformar o seu 600XL ou 800XL num 1450XL.

## 1.2. O teclado

O teclado é, em qualquer computador, o modo de comunicação por excelência, do homem com o micro. Ou seja, é através do teclado que o utilizador pode dizer ao computador o que quer que ele faça, programá-lo ou, mais simplesmente, actuar no seu trabalho por forma a alterar soluções ou movimentos, como é o caso dos jogos. Por isso mesmo, deve ser a parte a que quem adquire um computador deverá dar mais atenção (além do Basic – a linguagem específica dos micros – que será tratado mais adiante).

Conforme pode ver-se na Figura , a parte central do teclado do Atari assemelha-se quase inteiramente à de uma máquina de escrever. Tanto as letras como os símbolos se encontram nos mesmos locais que os de um teclado Internacional (QWERT...). Há, todavia, mais algumas teclas com palavras que pertencem já ao domínio dos micros. Vejamos quais são e que funções desempenham. E, para facilitar a busca, seguiremos a ordem pela qual tais teclas se encontram expostas.



ESC – Uma tecla muito útil é a primeira da primeira fila, a que contém ESC impresso, e tem várias utilizações possíveis:

- 1 – serve para sair de qualquer instrução ou programa.
  - 2 – serve para imprimir os caracteres especiais que se encontram em todas as teclas. Para obter este efeito, basta carregar primeiro em ESC e depois CONTROL e na tecla cujo caracter se pretende.
  - 3 – atrasar o efeito de um comando num programa.
- Se, por exemplo, fizer uma linha

PRINT " (carregar em ESC, em CONTROL e em = duas vezes) com a mensagem depois e as aspas a fechar, verá que o computador move o cursor duas linhas e só imprime na terceira.

**CLEAR** – É a tecla número 12 da primeira fila e serve para limpar o écran. Como para outras teclas com funções especiais, deve premir-se antes a tecla SHIFT ou CONTROL para obter o efeito indicado.

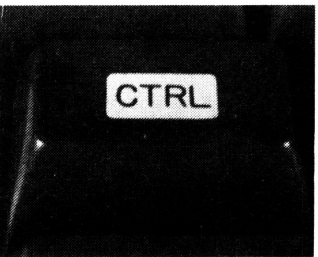
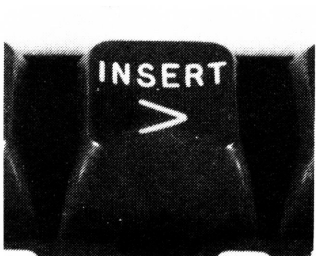
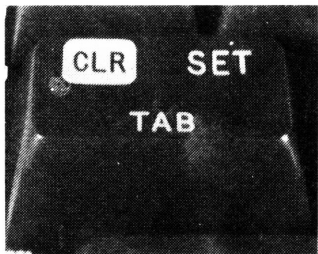
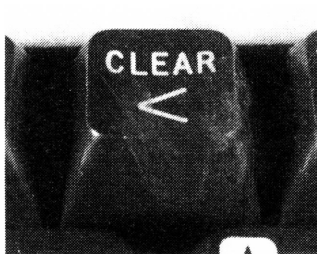
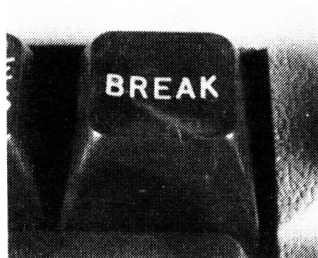
**INSERT** – Antepenúltima da primeira fila, esta tecla serve para inserir uma letra, palavra ou linha numa frase ou instrução já escrita. Para inserir uma letra ou palavra, carrega-se na tecla CONTROL e INSERT; para inserir uma linha, carrega-se na tecla SHIFT e INSERT.

**DELETE** – A penúltima tecla da primeira fila – DELETE BACK SPACE – serve para apagar um caracter ou uma linha que por engano se tenha escrito ou se pretenda simplesmente corrigir. SHIFT com DELETE apaga uma linha; CONTROL com DELETE apaga um caracter de cada vez.

**BREAK** – Última da primeira fila, serve para parar o computador, quando este se encontra a executar um programa. Se estiver a programar, serve também para mudar de linha o cursor, cada vez que tenha completado essa linha.

**CLR/SET/TAB** – Primeira tecla da segunda fila, tem como função principal marcar um ponto no écran para onde o cursor pode ser instantaneamente movido (aliás, tal tecla existe também nas máquinas de escrever vulgares). O computador está programado para um primeiro TAB, a oito espaços do ponto máximo da esquerda; pode todavia mudar-se este ponto para onde se quiser, bastando para isso carregar em CONTROL e TAB para limpar a marcação; em seguida carrega-se em RETURN; por fim, com a barra de espaços faz-se avançar o cursor até ao novo ponto desejado, o qual se marca com pressão simultânea em SHIFT e TAB.





**RETURN** – Este comando serve para introduzir uma linha ou comando na zona do computador que executa trabalho imediato.

**CONTROL** – É a primeira tecla da terceira fila e, como se viu já ao falar das teclas anteriores, possui muitas funções, em conjunto com outras teclas. Não repetiremos, portanto, essas funções. Para além delas, contudo, serve ainda para, uma vez mais em conjunto com as quatro teclas (lado direito do teclado) marcadas com setas, fazer mover o cursor nas quatro direcções indicadas. Uma das suas funções mais importantes é a de dar acesso ao set de caracteres gráficos do Atari, o que se consegue premindo-a em conjunto com uma das outras teclas; nesta função, é possível fixá-la; basta carregar na tecla **CONTROL** e na **SHIFT**.

– A tecla marcada com este símbolo (a última da última fila) é a que faz o **INVERSE VIDEO**. Quer dizer: se está a trabalhar com letras brancas sobre fundo azul, por exemplo, depois de carregar nessa tecla passa a trabalhar com caracteres azuis sobre fundo branco, na zona do écran em que imprimir a partir daí.

**TECLAS ESPECIAIS** – Do lado direito do teclado e numa fila vertical, aparece um jogo de teclas prateadas, distintas das outras: **RESET**, **OPTION**, **SELECT**, **START**, **HELP** e **POWER**. Antes de mais, trata-se de teclas com funções utilizadas em jogos adquiridos no mercado, podendo contudo servir para programação avançada. Algumas delas podem, contudo, ter também utilidade em uso comum:

**HELP** – Pode servir para testar as várias partes do computador: memória, som e video, teclado. Basta ligar o computador com essa tecla premida e ele fará um auto-teste automático e integral.

**OPTION** – Pode servir para escolher em que linguagem pretende trabalhar, ou ainda para retirar o Basic das memórias do computador, caso pretendam introduzir-se outras linguagens a partir de uma diskette.

**RESET** – Como em outros computadores, serve para colocar a máquina em estado zero, isto é, interromper seja qual for o trabalho do computador, limpando todas as memórias e colocá-lo numa situação semelhante à que fica quando é ligado. Deve ter-se pois cuidado

com a sua utilização, já que pode liquidar muito trabalho executado.

### 1.3. Os novíssimos modelos

Em Outubro de 1985, a Atari lançou nos mercados dos Estados Unidos e Canadá os modelos 130XE e 520ST. Embora o tenha feito sem qualquer literatura desenvolvida, tive o privilégio de estar em Toronto nessa data, tendo assistido a uma demonstração, pelo que não quero deixar de fazer uma breve referência ao primeiro desses dois computadores, reservando uma secção para a descrição detalhada do 520ST – verdadeiro membro de uma geração de futuro atendendo às características que possui.

O 130XE é, na verdade, um sistema de processamento de texto. Aliás, a Atari não lançou o computador isoladamente mas sim um conjunto a que chamou «ATARI 130XE WORD PROCESSING SYSTEM»; portanto, o computador foi concebido como a unidade central desse conjunto (embora, evidentemente, possa ser adquirido separadamente e ser utilizado em outros trabalhos).

A composição do sistema é a seguinte:

- O computador 130XE, que possui 130K de RAM, 62 teclas das quais 4 são programáveis e se destinam a funções especiais, e foi concebido com um conjunto de caracteres internacionais (incluindo já, portanto, acentos);

- A unidade Disk Drive 1050 – com uma drive para discos de 5 1/4 polegadas, de dupla densidade, o que permite guardar cerca de 100 páginas de texto, atendendo à sua capacidade de registo que é 127K;

- A impressora 1027, de 80 colunas e vários tipos de letra (os mais usuais em texto comercial), imprimindo a uma velocidade de 20 caracteres por segundo e podendo utilizar qualquer tipo de papel.

- Um programa de processamento de texto – o Atariwriter — com características idênticas a qualquer processador profissional, como sejam a busca e substituição de palavras, titulação, formatação de texto, paginação automática, etc.

## O 520ST

O 520ST não é, decididamente, um computador para mercados com limitações económicas mas, por outro lado, possui características espectaculares: 512K de RAM, definição gráfica máxima de 640×400 pixels (uma das imagens do programa mostrava o voo de uma borboleta e, à primeira vista, parecia mais a imagem real captada por uma câmara de vídeo!), e, imagine-se, 512 cores. Além de conter já implementado um Hard Disk.

Claro que existem computadores com características semelhantes e até melhores. Mas não pelo preço a que a Atari lançou este: cerca de metade do custo de outro qualquer computador com características iguais.

Embora não tenha possibilidades de expansão interna, a RAM poderá partir dos 512K para uma expansão que chegará a 16Mb! Tem um comando especial para acesso directo à memória e uma porta para a inserção de cardtridges com ROM de 128K.

O teclado é composto por 95 teclas, das quais 10 são programáveis para as funções especiais, havendo igualmente um grupo de teclas numéricas. Além disso, vem equipado com uma extensão que permite o controlo directo do cursor.

Embora a utilização das cores seja limitada ao modo em que o computador esteja a funcionar, é possível colocar no écran as 512 cores simultaneamente, graças a um comando especial: o BIOS.

A parte de som está a cargo de um processador especial que produz toda a gama de oitavas captáveis pelo ouvido humano, em três canais independentes. Possui, além disso, um canal de ruídos.

Segundo as primeiras análises críticas, já não se trata de um computador desta geração e, neste momento – atendendo à relação performances/preço – não tem qualquer competidor no mercado mundial.



## A LINGUAGEM DO ATARI

### 2.1. As linguagens dos computadores

Os seres humanos utilizam vários tipos de linguagem para comunicar entre si: língua oral e escrita, sinais, etc.

Também os computadores utilizam vários tipos de linguagem para comunicar entre as suas partes e ainda com periféricos. E tal como há centenas de línguas humanas, há também centenas de linguagens na informática.

A mais importante – porque muito rápida e permitindo realizar tarefas que uma mais lenta e complexa não possibilita, para além da memória reduzida que ocupa – é a LINGUAGEM MÁQUINA. De qualquer modo, esta só será acessível a quem tenha já razoáveis conhecimentos de programação em linguagens mais simples.

Mesmo assim, é bom saber que os computadores Atari podem utilizar vários tipos de linguagem, para tanto bastando adquirir e inserir a cartridge respectiva. Por isso, aqui fica uma lista das existentes no mercado:

**PASCAL** – Linguagem especialmente utilizada em programas educacionais e científicos.

**FORTH** – Linguagem especialmente utilizada em cálculo e desenho, indicada para engenharia e ciências várias.

**PILOT/LOGO** – Linguagem desenvolvida recentemente, com um vocabulário e sintaxe muito simples, destinada sobretudo a fins educacionais, nomeadamente a nível da instrução primária.

**C** – Uma linguagem muito rápida e poderosa, com aplicações muito rentáveis em gestão.

**ADVANCED ATARI BASIC** – É o Basic desenvolvido da Atari, com novos comandos que permitem funções e comandos na programação, alguns deles muito próximos da linguagem máquina.

Mas para que o homem comunique com o computador, é indispensável que o faça numa linguagem comum. Na verdade, o vocabulário humano é tão variado e ilimitado (e isto para já não falar das centenas de línguas e dialectos existentes ) que seria praticamente impossível prever uma máquina capaz de responder a qualquer das infinitas hipóteses possíveis de frases.

Por esta razão se construiu uma linguagem intermédia – entre a dos computadores e a língua inglesa – a que se chamou **BASIC**. É essa a linguagem mais simples para a comunicação homem/computador, sendo também a mais universal.

Apesar disso, cada fabricante introduziu algumas variações que fazem com que, mesmo já conhecedor do Basic, o utilizador tenha que aprender essas alterações quando compra um novo computador de uma marca diferente. Por isso se apresenta aqui a lista completa do Basic da Atari, com alguns comentários sobre as variações específicas que o distinguem de outros Basics mais comuns.

## **2.2. O Basic da Atari**

- ABS** – Dá o valor absoluto de um determinado número.
- AND** – Comando que serve para verificar se duas expressões ou valores resultam numa proposição verdadeira ou falsa.
- ASC** – Dá o código ATASCII de um carácter.
- ATN** – Dá a tangente invertida de um número.
- CHR\$** – Dá um carácter a partir de um número ATASCII.
- CLOAD** – É um comando que ordena ao computador que carregue em memória um programa ou ficheiro anteriormente guardados em cassette. Após dar esta ordem, deverá carregar no botão **PLAY** do gravador e aguardar. Qualquer programa existente em memória será integralmente apagado.
- CLOG** – Dá o logaritmo comum de um número positivo.
- CLOSE** – Fecha um ficheiro anteriormente aberto com o comando **OPEN**.

- CLR – Repõe todas as variáveis com o valor 0 e todas as variáveis alfanuméricas com a dimensão 1.
- COLOR x – Permite a escolha da cor a ser utilizada com os comandos PLOT e DRAWTO. Em modos avançados gráficos, pode desempenhar outras funções.
- CONT – Serve para ordenar ao computador que continue a execução de um programa após ter parado com um comando STOP ou BREAK.
- CSAVE – Ordena ao computador que grave em cassete um programa existente em memória. Após um sinal «beep» do computador, deve carregar simultaneamente nos botões PLAY e RECORD do gravador, o que provocará a transferência referida.
- DATA – Comando utilizado em conjunto com READ, para guardar dados numéricos e alfanuméricos numa determinada ordem. Tais dados serão lidos por essa mesma ordem.
- DEG – Ordena ao computador que converta todas as funções trigonométricas de radianos para graus.
- DIM – Reserva espaços em memória para serem preenchidos por números ou expressões. Pode reservar espaços com uma ou mais dimensões.
- DOS – Ordena ao computador que faça arrancar o sistema de disco, passando o utilizador a ter acesso ao mesmo para as diversas operações. Só pode ser utilizado desde que um disco esteja metido no drive.
- DRAWTO x,y – Comando utilizado nos modos gráficos para desenhar uma recta, partindo do ponto em que se encontra o ponteiro, para um ponto indicado pelas coordenadas x e y. Este ponto deverá estar dentro dos limites definidos pelo modo em que o micro se encontra.
- END – Indica a última instrução num programa. Equivale à instrução STOP de outros computadores.
- ENTER – Utilizado com o gravador de cassettes ou a unidade de disco para carregar um programa, não apagando contudo as linhas existentes eventualmente em memória. Assemelhando-se ao comando MERGE em outros computadores, adiciona as novas linhas de programa às já existentes.
- EXP – Dá o valor de função exponencial a um número.
- FOR/NEXT — O FOR indica o início de uma sequência repetitiva, indicando igualmente quantas vezes deve ser

- repetida. O fim da sequência tem que ser indicado com o comando NEXT.
- GET – Utiliza-se com ficheiros de dados para dar a uma variável o valor numérico de outra.
- GOSUB n – Ordena ao computador que salte para uma subrotina iniciada na linha n, de onde regressará quando encontrar o comando RETURN.
- GOTO n – Semelhante ao comando anterior, mas sem voltar à linha seguinte àquela de onde saltou.
- GRAPHICS n – Limpa o écran e passa a um novo modo gráfico. O número n indica o novo modo, sendo os mais comuns o 0 (texto), 7 (gráficos em quatro cores) e 8 (gráficos de alta-resolução). Adicionando 16 ao número do modo, elimina-se a janela de texto ao fundo do écran.
- IF/THEN – É a instrução mais importante dos computadores, pois permite-lhes tomar decisões. Se a condição expressa a seguir ao comando IF, existir, o computador executa a instrução que segue o comando THEN. Caso contrário, não a executa.
- INPUT – Serve para permitir ao operador a introdução directa de um valor ou frase. Quando este comando aparece, a execução do programa fica suspensa até à resposta do operador.
- INT – Arredonda um número para o integral mais próximo.
- LEN – Dá o número de caracteres existentes numa variável alfanumérica.
- LET – Dá um valor a uma variável.
- LIST – Serve basicamente para ordenar ao computador que imprima no écran a listagem completa de um programa que se encontra em memória. LIST 150, resulta na listagem da linha 150. LIST 150, 500 resulta na listagem de todas as linhas compreendidas entre a 150 e a 500.  
Este comando pode igualmente ser utilizado para gravar um programa em cassette ou diskette. Tal programa só pode ser posteriormente carregado com a utilização do comando ENTER.
- LOAD – Carrega um programa guardado em cassette ou diskete, utilizando o comando SAVE.
- LOCATE – Guarda numa variável o valor de uma localização do écran.
- LOG – Dá o logaritmo natural de um número positivo.
- LPRINT – Imprime dados no papel de uma impressora.

- NEW** – Limpa a memória do computador por forma a permitir a escrita de um novo programa.
- NEXT** – Marca o fim de um ciclo de repetição iniciado com um comando **FOR**. A variável colocada após este comando deve ser a mesma colocada a seguir ao comando **FOR** que iniciou o ciclo.
- ON** – Utilizado em conjunto com os comandos **GOTO** ou **GOSUB** para o computador saltar para uma linha cujo valor é o de uma variável.
- OPEN** – Abre um ficheiro para introduzir ou emitir dados ou outras operações.
- OR** – Comando que compara duas expressões, considerando o resultado verdadeiro se uma delas o for.
- PADDLE** – Utilizado com um «paddle» ligado ao micro, dá um número entre 0 e 228, dependendo da posição do paddle.
- PEEK** – Fornece o valor, entre 0 e 255, de uma determinada posição da memória (ROM ou RAM).
- PLOT x,y** – Em modo de gráficos, coloca um ponto nas coordenadas x e y do écran. Serve igualmente para indicar o início de uma linha desenhada com o comando **DRAW**.
- POKE** – Coloca um número, entre 0 e 225, numa posição da memória (RAM).
- POSITION x,y** – No modo de texto, move o cursor directamente para as coordenadas x e y.
- PRINT** – Imprime números, letras ou frases no écran. Se se quer imprimir uma palavra ou frase, é necessário escrevê-la entre aspas.  
Quando se utilizam os modos de texto 1, 2, 12 e 13, o comando será **PRINT 6;** «mensagem».
- PTRIG** – Dá o valor 0 se o botão do «paddle» for premido.
- PUT** – Utilizado com ficheiros de dados para enviar um valor numérico para uma posição.
- RAD** – Ordena ao computador que converta todas as funções trigonométricas de graus para radianos.
- READ** – Utiliza-se em conjunto com o comando **DATA** e serve para ler (e atribuir a uma variável) os valores indicados nesse comando.
- REM** – Permite introduzir qualquer frase (que não será executada) num qualquer ponto de um programa.
- RESTORE** – Coloca o ponteiro de leitura (**READ**) no primeiro registo de dados (**DATA**).

- RETURN** – Marca o fim de uma subrotina. Ao encontrar este comando, o computador volta à linha a seguir ao **GOSUB** de onde partiu.
- RND** – Dá um número aleatório entre 0 e 1.
- RUN** – Ordena a execução de um programa existente na memória do computador.
- SAVE** – Grava o programa existente em memória, em casete ou diskette.
- SETCOLOR** – Dá uma nova cor ao «pincel» utilizado para pintar o écran nos modos gráficos. Nos modos de texto, controla as cores das margens, papel e letras.
- SIN** – Dá o **SENO** de um ângulo.
- SOUND** – Utilizado para controlar sons produzidos pelo computador.
- SQR** – Dá a raiz quadrada de um número positivo.
- STEP** – Utilizado com um comando **FOR... NEXT** para indicar o intervalo (opcional) entre cada repetição.
- STICK** – Dá o valor correspondente à direcção em que um joystick se encontra.
- STRIG** – Dá um 0 se o botão de um joystick for premido.
- STOP** – Pára a execução de um programa.
- THEN** – Utilizado com um **IF**. Se a condição for verdadeira, o computador executa a instrução; caso contrário, não.
- TO** – Utilizado com o **FOR... NEXT** para separar o valor mais alto e o mais baixo de uma variável.
- TRAP** – Envia o computador para uma linha indicada, se o computador encontrar um erro na execução de um programa.
- XIO** – Utiliza-se para pintar zonas do écran.
- = – O sinal de igual pode ser utilizado como um comando para atribuir um valor a uma variável (o equivalente, em outros computadores, ao comando **LET**).

### **2.3. O uso de alguns comandos**

Não se pretende nesta secção explicar o uso exaustivo de todos os comandos do Atari. Isso daria um livro de centenas de páginas, pouco prático, demasiado caro e fora do âmbito da presente obra.

Destina-se portanto, e sobretudo, a apresentar alguns dos comandos e instruções mais utilizados, ou outros eventualmente

difíceis. Serão deixados para capítulos posteriores, por razões óbvias, instruções que tenham a ver com gráficos, som ou periféricos.

Por outro lado, não vamos iniciar instruções sobre programação, uma vez que pensamos ser melhor, neste ponto, utilizar o modo directo – que permite verificar imediatamente os resultados da experiência que esteja a realizar-se.

## PRINT

O comando PRINT (significa IMPRIMA, em português) é dos mais importantes de um computador: serve para que o micro imprima letras, palavras, frases, números e outros caracteres no écran do televisor ou monitor.

Com o sistema ligado, experimente escrever o seguinte:

```
PRINT "ESTA INSTRUÇÃO SERVE PARA IMPRIMIR"
```

e carregue na tecla RETURN. O computador imprimiu a mensagem que se encontrava entre aspas, no topo do écran. Nos computadores da Atari é possível omitir as aspas no fim da mensagem. Se, por exemplo, escrever:

```
PRINT "SEGUNDA INSTRUÇÃO
```

carregando depois em RETURN, o micro imprime a mensagem sem problemas. Mas se escrever

```
PRINT ENTÃO COMO ESTÁ
```

ou

```
PRINT ENTÃO COMO ESTÁ"
```

o computador imprimirá uma mensagem de erro, em vez da frase indicada.

As aspas só são necessárias para imprimir letras. Se escrever

```
PRINT 1234
```

obterá no écran os números 1234. Portanto, para números não são necessárias aspas.

É possível utilizar a instrução PRINT de muitas outras maneiras. Por exemplo, podem utilizar-se várias destas instruções numa só linha:

```
PRINT "PRIMEIRA": PRINT "SEGUNDA": PRINT  
"TERCEIRA"
```

resultará na impressão das três palavras, uma em cada linha do écran.

É também possível imprimir duas palavras (ou frases) seguidas, utilizando o ponto e vírgula (;):

```
PRINT "PRIMEIRA"; "SEGUNDA"
```

O uso da vírgula (,) em vez do ponto e vírgula (;), faz com que o computador deixe um número fixo de espaços entre o início de cada nova instrução PRINT. Cada zona de vírgula tem 10 espaços.

Experimente

```
PRINT "A", "B", "C", "D"
```

e verá o computador imprimir

A

B

C

D

Para que o computador imprima resultados de contas matemáticas, pode utilizar-se igualmente o comando PRINT. Por exemplo,

```
PRINT 5*2+5
```

fará com que o computador imprima o número 15 no écran.

## CURSOR

O cursor é um pequeno quadrado que certamente já viu e que pisca no écran, precisamente no ponto onde será impresso o próximo carácter teclado pelo utilizador. À medida que se vai escrevendo, este quadrado vai-se movendo à frente da última letra ou espaço.



Embora seja automático no seu movimento, é possível movê-lo em quatro direcções para qualquer ponto do écran. Para isso basta utilizar a tecla CONTROL e qualquer uma das marcadas com setas (estas indicam a direcção de movimento do cursor).

Experimente, tentando escrever várias palavras em vários pontos do écran.

A grande utilidade das teclas com as setas, e a sua capacidade de mover o cursor, manifesta-se, porém, quando se está a fazer programação e se pretendem fazer correcções.

Escreva no écran a frase:

ENGANEI-ME NUMA LITRA NESTA FRASE

Agora carregue em CONTROL e na tecla com a seta virada para a esquerda até sobrepôr o cursor à letra I da palavra LITRA. Solte a tecla CONTROL e escreva a letra E. Pronto, está corrigido.

Suponha agora que escreveu:

FALTA UMA LETA NUMA PALAVRA

Que fazer neste caso? Simples:

Carregue em CONTROL e na tecla da seta virada para a esquerda, por forma a sobrepor o cursor ao T da palavra LETA. Uma vez feito isso, carregue em CONTROL e na tecla INSERT; isto faz abrir um espaço entre o E e o T da palavra em causa. Agora basta escrever o R que faltava.

O mesmo se pode fazer para apagar uma letra a mais, devendo neste caso carregar-se em DELETE em vez de INSERT.

INVERSE

A tecla marcada com um quadrado de duas cores (separadas por uma diagonal), serve para imprimir em video inverso. Quer dizer, se o fundo for azul e as letras brancas, após ter carregado nessa tecla o computador passará a escrever em caracteres azuis sobre fundo branco. Portanto nas cores inversas das utilizadas normalmente.

Experimente carregar na tecla indicada e depois escreva algumas palavras.

Agora outra experiência:

- Escreva PRINT
- Carregue na tecla de INVERSE
- Escreva "ISTO É VIDEO INVERTIDO"
- Carregue em RETURN

A frase entre aspas aparecerá em inverse no écran. O que significa que o comando INVERSE também pode ser utilizado fora do comando directo.

LET

O comando LET (que pode reduzir-se a um sinal de igual no Atari) é muito útil pois permite atribuir qualquer valor a qualquer variável.

Por exemplo, se escrever a seguinte instrução:

```
PRINT MANUEL
```

o computador nada imprime e dá-lhe indicação de uma mensagem de erro, uma vez que tentou escrever uma cadeia alfanumérica - MANUEL - sem aspas.

Experimente agora o seguinte:

```
LET MANUEL=123
```

e em seguida,

```
PRINT MANUEL
```

e o computador não só não recusa cumprir a instrução como imprime o número 123, que foi o que foi atribuído à cadeia alfanumérica MANUEL.

Na realidade, bastaria escrever apenas,

```
MANUEL=123
```

já que no Atari, como atrás se disse, o comando LET é dispensável.

Experimente agora,

MANUEL=667

e,

PRINT MANUEL

e o computador imprimiu o número 667, visto ter sido o último valor atribuído à cadeia MANUEL.

Uma última e importante experiência. Escreva,

PRINT "MANUEL"; MANUEL

e o computador imprime MANUEL 667. Isto é importante pois significa que continua a poder utilizar as cadeias alfanuméricas (desde que dentro de aspas) que entender, mesmo com o mesmo nome de uma variável, porque o computador não as confunde.

Veremos mais tarde a utilidade deste comando.

CLEAR

Neste momento, e após todas estas experiências, o écran deve estar cheio de palavras, frases e outros caracteres; enfim, uma confusão. É necessário limpá-lo, e não precisa desligar o computador para o fazer.

Escreva,

CLEAR e RETURN

Pronto. O écran ficou limpo e pronto para continuar a escrever nele.

A verdadeira utilidade deste comando surge, porém, quando estiver a fazer programação, já que pode apagar o écran sem apagar das memórias as linhas de programa.

PEEK

Peek quer dizer ESPREITAR, em inglês. E é precisamente

para isso que o comando serve; ver qual o número colocado num determinado endereço de memória.

Se escrever, por exemplo,

PEEK 777

obterá um número, entre 0 e 255, que não vamos explicar agora já que exigiria conhecimentos que provavelmente ainda não possui.

De qualquer modo, tal comando virá a ser útil quando tratarmos dos Modos e, mais tarde, quando utilizar a linguagem máquina.

## POKE

O POKE não é um comando de utilização frequente. Na realidade, é mesmo pouco provável que o utilize nos primeiros tempos, já que necessita saber e entender algo sobre localizações de memória e significado dos números guardados nesses endereços.

Todavia, e como é usado por exemplo em alguns modos, sempre fica a saber que tal comando serve precisamente para alterar directamente valores de endereços de memória RAM.

Se, por exemplo, escrever:

PEEK 750

o computador imprime um número, entre 0 e 255, o correspondente ao guardado no endereço 750.

Agora escreva:

POKE 750,0

(que significa: coloque o número 0 no endereço 750), escreva

PEEK 750

e verá que o valor se alterou para 0.

## TRAP

Este é outro comando muito útil, equivalente em outros computadores à instrução ON ERROR GOTO, e que serve para que, ao encontrar um erro, em vez de parar o programa continue na linha indicada. Por isso a sua sintaxe é

### TRAP X

sendo X o número da linha para onde deve saltar.

## 2.4. A programação

Até agora utilizámos, nas explicações e experiências, comandos directos. Quer dizer que cada vez que demos uma ordem ao computador e carregámos em RETURN, imediatamente ele a executou sem a guardar em memória; não a poderia, portanto, repetir outra vez, a menos que voltássemos a escrevê-la.

Ora as grandes potencialidades dos computadores só são verdadeiramente úteis quando é possível dar-lhes umas centenas (ou mesmo milhares) de instruções que os levem a executar um certo número de tarefas complexas. Mas tal não é possível utilizando comandos directos; é necessário organizar tais instruções em programas.

Um programa é, portanto, um conjunto de instruções memorizadas segundo uma determinada ordem. Ordem essa indicada por números, sendo o mais baixo o correspondente à primeira tarefa, e assim por diante.

A cada um desses números corresponde uma linha de programa. E, desde já, uma regra muito importante: **NUNCA DÊ NÚMEROS SEGUIDOS ÀS LINHAS DOS SEUS PROGRAMAS**, pois pode mais tarde querer introduzir instruções intermédias. É costume fazer um intervalo de 10 entre cada linha (10, 20, 30...).

Para ver como funciona, experimente escrever

```
10 PRINT "isto é um programa"
```

(carregue em RETURN, como sempre que uma linha esteja completa)

```
20 PRINT "com várias instruções"  
30 PRINT "e linhas"  
40 END
```

O que aconteceu foi:

- cada vez que carregou em RETURN, o computador não executou a ordem;
- imprimiu a linha completa na parte superior do écran (e a segunda por baixo, etc.)

Se agora escrever RUN e carregar em RETURN... o computador executa todas as instruções contidas nas 4 linhas, uma de cada vez e pela ordem correcta.

Agora escreva LIST e carregue em RETURN... o computador executa todas as instruções contidas nas 4 linhas, uma de cada vez e pela ordem correcta.

Agora escreva LIST e carregue em RETURN. O programa voltou a aparecer no écran.

Quer dizer:

- pode memorizar uma série de instruções;
- poderia até tê-las corrigido com o comando EDIT;
- o computador só as executou quando lhe foi dada ordem para isso;
- manteve-as em memória para repetir a execução ou mesmo para que fossem feitas alterações.

Experimente agora escrever o seguinte programa, tal como aqui é apresentado:

```
50 PRINT "e o computador ordenou-as"  
30 PRINT "as linhas foram escritas"  
60 PRINT „correctamente"  
10 CLEAR  
20 PRINT "ORDENAMENTO"  
40 PRINT "aleatoriamente"  
70 END
```

Escreva RUN e carregue em ENTER

O que aconteceu foi:

- o computador ordenou automaticamente as linhas do programa, começando no número mais baixo - o 10 - e terminando no mais alto - o 70;

– executou o programa pela ordem correcta.

Quer dizer que quando estamos a elaborar um programa não há necessidade de nos preocuparmos com o ordenamento; e, por outro lado, podemos sempre introduzir qualquer linha, em qualquer ponto e em qualquer momento.

Agora experimente escrever,

60

carregando em RETURN.

E a linha 60 foi simplesmente apagada das memórias do computador. Quer dizer que pode corrigir linhas com os já focados comandos DELETE e EDIT, mas para apagar uma linha basta escrever o seu número sem mais nada.

Embora mais adiante se vá falar em pormenor dos modos e gráficos, aqui fica um programa demonstrativo de que TUDO pode ser programado num computador, e não só texto e a instrução PRINT.

```
10 GRAPHICS 7
20 COLOR 1
30 PLOT 10,10
40 DRAWTO 70,10
50 DRAWTO 70,70
60 DRAWTO 10,70
70 DRAWTO 10,10
```

Escrevendo RUN e carregando em ENTER, verá o computador desenhar um quadrado no écran.

E mais uma coisa: neste programa não se escreveu uma última linha com a instrução END. Assim, e embora seja conveniente fazê-lo, é dispensável dar ordem ao computador para parar após a última linha do programa já que ele o faz, de qualquer maneira, quando encontra a linha com o número mais alto.

E, já agora, o número mais alto que pode atribuir a uma linha de programa, nos computadores Atari, é o 32767.

## GRÁFICOS E TEXTO

Uma das mais interessantes características dos computadores é a possibilidade de variação do écran, de cores e dos diferentes tamanhos de caracteres com que se escreve, além dos desenhos e outros efeitos gráficos. Para obter tais efeitos utilizam-se vários MODOS de impressão – chamados DISPLAY MODES – cada um com características específicas, no que respeita às variações possíveis.

Um DISPLAY MODE é, portanto, qualquer modo que permite imprimir no écran texto ou desenhos. Até agora utilizámos apenas o modo automático, ou seja aquele que surge quando se liga o computador, e que imprime letras brancas em fundo azul. É possível, contudo, aumentar ou diminuir o tamanho das letras, imprimi-las em cor diferente sobre o fundo diferente e até utilizar janelas de texto em écrans de desenho.

### 3.1. Écrans

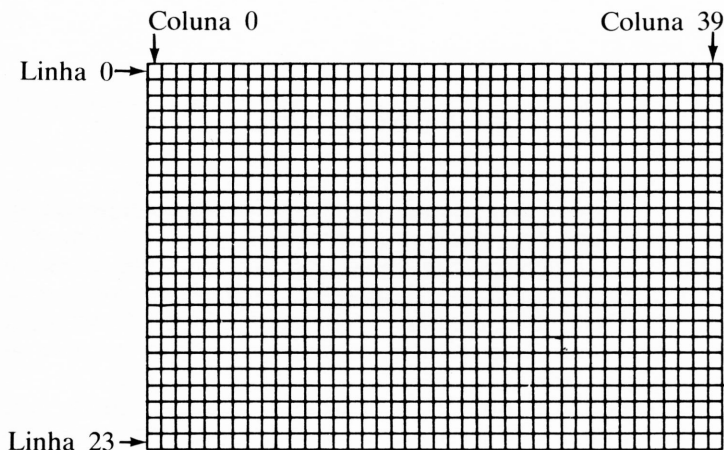
Se o seu Atari estiver ligado a um televisor ou monitor a cores, ao ligá-lo surge-lhe um écran azul circundado por um bordo negro. A área de écran em que pode escrever ou desenhar é apenas azul. O bordo é apenas uma espécie de moldura sobre a qual o computador não pode executar qualquer tarefa.

Podemos dizer que o écran é a área útil marcada no cinescópio da TV ou monitor. Nesta área, cuja cor pode ser alterada, é possível colocar qualquer linha, caracter ou pintura, em qualquer ponto.

O écran encontra-se dividido em quadrados, podendo em



cada um deles ser colocado um caracter. Cada um destes quadrados é composto por 64 pontos ( $8 \times 8$ ) cada um dos quais é chamado PIXEL (elemento mínimo de display). O número de quadrados (horizontal e vertical) de um écran depende do modo em que o computador se encontra a funcionar. Por exemplo no Modo 0 trata-se de uma grelha de  $40 \times 23$  quadrados.



A grelha do modo 0

As colunas e as linhas são numeradas a partir do 0, pelo que a primeira tem o número 0, a segunda o 1, e assim por diante. Para se localizar um quadrado num écran, utilizam-se as variáveis X (para as colunas) e Y (para as linhas). O ponto de encontro das duas coordenadas (X e Y) é o procurado.

Primeiro dá-se sempre o valor do X e em seguida o do Y, separados por uma vírgula. Assim, se quisermos colocar um caracter no ponto de encontro 17,5 (ou seja, na coluna 17 e linha 5) escreve-se o comando respectivo seguido de 17,5.

Quanto mais colunas e linhas tiver um écran maior é a RESOLUÇÃO, quer dizer, a qualidade da imagem. A resolução mede-se sempre pela multiplicação entre o número de colunas pelo de linhas ( $40 \times 24$  no modo 0).

## 3.2 Os modos e as cores

Nos computadores Atari é possível utilizar 16 modos e 15 cores. Destes 16, 5 são chamados modos de texto uma vez que são utilizados sobretudo para escrever texto no écran. Os restantes 11 destinam-se a gráficos.

Antes de falar, com certo pormenor, acerca dos modos, convém referir e explicar os comandos utilizados nos mesmos.

### GRAPHICS

Este é o comando fundamental, já que só através dele pode mudar para qualquer dos modos possíveis. Assim, para passar o computador para o Modo 7,

GRAPHICS 7

e em seguida RETURN.

A sintaxe deste comando é muito simples, bastando indicar o número do modo que se pretende a seguir ao comando GRAPHICS.

### SETCOLOR

Este comando serve para controlar e alterar as cores do fundo do écran. A sua sintaxe é

SETCOLOR A, B, C

em que o A terá que corresponder a um número entre 0 e 4, correspondente aos cinco registos de cor memorizados. O B refere-se à cor e o C à luminosidade.

### TABELA DO SETCOLOR

0 cinzento	8 azul normal
1 dourado	9 azul claro
2 laranja	10 turquesa
3 laranja escuro	11 verde-mar
4 rosa	12 verde
5 café-com-leite	13 amarelo
6 azul escuro	14 verde escuro
7 violeta	15 laranja claro

Os valores do B e do C são dados automaticamente pelo computador, conforme a seguinte tabela:

A	B	C	COR
0	2	8	laranja
1	12	10	verde claro
2	9	4	azul escuro
3	4	6	rosa
4	0	0	preto

## COLOR

O comando COLOR serve para seleccionar uma cor de entre as existentes nos registos.

É um comando sem qualquer utilidade, a não ser utilizado conjuntamente com outros comandos, ao contrário do SETCOLOR que, por si só, altera por exemplo a cor do écran.

A sintaxe de COLOR é simples:

COLOR X

onde X é um número entre 0 e 15.

A utilização prática deste comando será vista mais tarde na aplicação a programas.

## PLOT

É um comando que serve para imprimir um ponto colorido no écran. E aqui está um exemplo do uso do comando COLOR.

Portanto escreve-se, por exemplo,

COLOR 1

e em seguida,

PLOT X, Y

onde X e Y são os valores das coordenadas respectivas do ponto que se pretende imprimir.

## DRAWTO

É um comando utilizado para desenhar uma linha entre dois pontos do écran.

Para o utilizar é necessário dar três indicações ao computador:

- cor da linha (COLOR X);
- ponto de início (PLOT X,Y);
- ponto terminus (DRAWTO X,Y).

Feito isto, o computador traçará uma recta na cor indicada entre os dois pontos referidos.

Experimente,

```
COLOR 1: PLOT 10,10: DRAWTO 20,20
```

e RETURN.

## PRINT

Este comando, de resto já descrito no capítulo anterior, só pode ser utilizado em modos de texto ou em modos gráficos que possuam janela para texto.

## POSITION

Trata-se de um comando importante, já que permite imprimir texto em qualquer ponto do écran. A sua sintaxe é

```
POSITION X, Y
```

onde X é a coordenada da coluna e Y a coordenada da linha. Será a partir do ponto de encontro dessas duas coordenadas que o texto será impresso.

Escreva, por exemplo,

```
POSITION 19,11: PRINT "CENTRO"
```

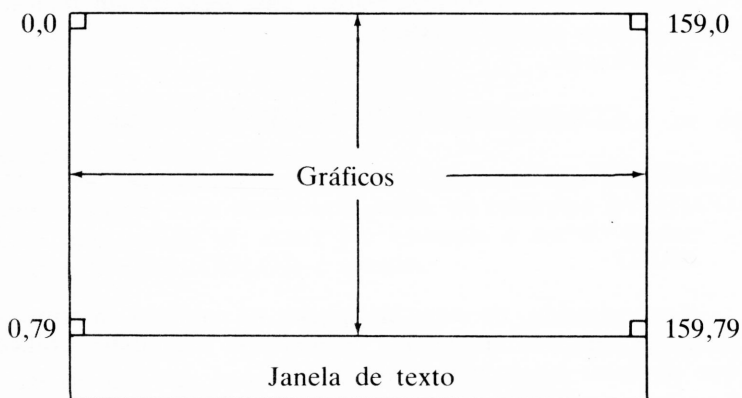
e a palavra CENTRO será escrita com o C no centro do écran.

## OS MODOS GRÁFICOS

### *Modos 3 a 7*

Todos estes modos usam a mesma cor para o bordo e fundo tendo igualmente possibilidade de uso de janela para texto. Podem dividir-se em dois grandes grupos:

- duas cores (modos 4 e 6);
- quatro cores (modos 3, 5 e 7).



O écran no modo de 4 cores

Nos modos 4 e 6, a COR 2 é a do fundo e bordo, enquanto nos modos 3, 5 e 7 é a COR 4 que ocupa aquele lugar.

A única vantagem dos modos 4 e 6 sobre os outros três é a de que os primeiros deixam muito mais memória livre para outras utilizações. Se um programa não for muito grande poderão usar-se, sem problemas, os modos 3, 5 e 7.

Quanto à resolução, é a seguinte:

- modo 3: 40×20
- modo 5: 80×40
- modo 7: 160×80

A área ocupada por cada pixel no modo 3 é 4 vezes mais que a ocupada no modo 5, a qual é, por sua vez, 4 vezes maior que a do modo 7.

### *Modo 8*

Muito diferente dos modos anteriores, tem apenas duas cores mas, em compensação, é o que tem o melhor grau de resolução: 320×160.

Por esta razão é o indicado para desenhos de precisão, conseguindo obter-se resultados surpreendentes.

O bordo e o fundo podem ser de cores diferentes, sendo o segundo controlado pela COR 2 e o primeiro pela COR 1.

### *Modos 9 a 11*

Exclusivos dos Atari da série XL, são todavia os mais utilizados. São eles que permitem criar desenhos multicolores, incluindo o uso de vários tons da mesma cor – característica muito útil para sombreados e relevos.

Nenhuns destes modos permite o uso de janelas para texto tendo todos a mesma resolução: 80×192.

O modo 9 é, entre estes, o de maior gama pois permite a utilização de 16 variações de brilho para uma só cor.

O modo 10 tem 9 cores independentes (cada uma com um determinado brilho). Para alterar os registos é necessário utilizar a instrução POKE, sendo os seguintes os endereços respectivos:

ENDEREÇO	UTILIZAÇÃO
704	Cor 0, Fundo e bordo
705	Cor 1
706	Cor 2
707	Cor 3
708	Cores 4 e 12
709	Cores 5 e 13
710	Cores 6 e 14
711	Cores 7 e 15
712	Cores 8, 9, 10 e 11

Para encontrar o valor a colocar num destes endereços, com a instrução POKE, multiplica-se o valor B por 16 e adiciona-se o valor C (B e C correspondentes aos segundo e terceiro valores de um comando SETCOLOR). Por exemplo:

SETCOLOR 2,5,4

resulta num POKE 710,84 já que  $5 \cdot 16 + 4 = 84$

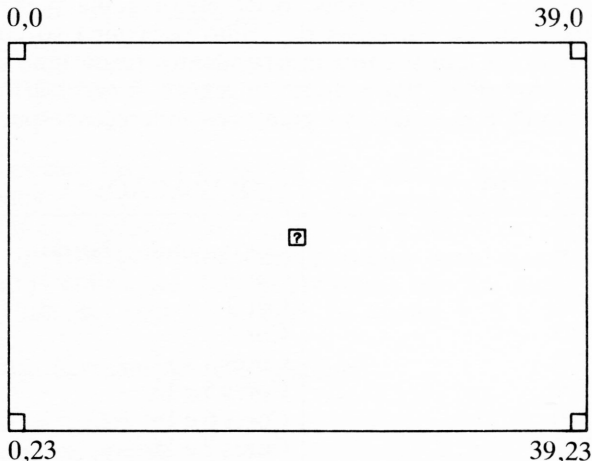
O modo 11 tem 16 tons e 1 brilho, sendo o ideal para desenhos multicolores. A instrução SETCOLOR 4 controla o brilho dos 16 tons.

### *Modos 14 e 15*

Só existem nos Atari da série XL, sendo o modo 14 de 2 cores e o modo 15 de 4.

O modo 15 é o de 4 cores com a melhor resolução:  $160 \times 160$  sendo, por isso, o ideal para desenhos coloridos com muito detalhe.

### OS MODOS DE TEXTO



O MODO 8.

### *Modo 0*

É o modo mais vulgar (e o primeiro a ser conhecido na prática, antes mesmo do conhecimento do que são os modos e de como actuam), já que é aquele em que o computador funciona, automaticamente, quando é ligado.

Embora seja o modo de texto por excelência, pode, em certas circunstâncias, ser utilizado para gráficos.

De resto, já falámos bastante deste modo no capítulo anterior.

### *Modo 1*

As principais diferenças deste modo relativamente ao Modo 0 são:

- O texto pode ser colorido, sendo as cores dos caracteres independentes do fundo;
- Os caracteres são duas vezes mais largos;
- Existe uma janela de texto.

### *Modo 2*

Idêntico ao modo 1, mas com os caracteres duas vezes mais altos.

### *Modo 12*

Trata-se de um modo de 4 cores, com janela de texto. Os caracteres têm o mesmo tamanho que os do modo 0, sendo possível pintar cada carácter com as 4 cores.

### *Modo 13*

Idêntico ao modo 12 e com as mesmas possibilidades, sendo os caracteres duas vezes mais altos.



### 3.3. Desenhar, pintar

Vamos apresentar neste capítulo alguns pequenos programas que permitirão praticar o uso dos comandos e instruções focados no anterior.

Para o comando SETCOLOR:

```
10 FOR B=0 TO 15
20 PRINT "COR"; B
30 SETCOLOR 2,B,6
40 FOR D=1 TO 500
50 NEXT D
60 NEXT B
```

\*

```
10 FOR B=0 TO 15
20 PRINT "COR"; B
25 FOR C=0 TO 14 STEP 2
27 PRINT "BRILHO"; C
30 SETCOLOR 2,B,C
40 FOR D=1 TO 250
50 NEXT D
55 NEXT C
60 NEXT B
```

Para o comando PLOT:

```
10 GRAPHICS
20 COLOR 1
30 PLOT 20,10
```

\*

```
10 GRAPHICS 3
20 SETCOLOR 0,4,8
30 COLOR 1
40 PLOT 20,10
```

```
10 GRAPHICS 3
20 LET B=INT (16*RND(0))
30 LET C=2*INT(8*RND(0))
40 SETCOLOR 0,B,C
50 COLOR 1
60 LET X=INT(40*RND(0))
70 LET Y=INT(20*RND(0))
80 PLOT X,Y
90 FOR D=1 TO 200: NEXT D
100 GOTO 10
```

\*

```
10 GRAPHICS 3
20 LET C=INT(4*RND(0))
30 COLOR C
40 LET X=INT(40*RND(0))
50 LET Y=(20*RND(0))
60 PLOT X,Y
70 FOR D=1 TO 200; NEXT D
80 GOTO 20
```

Para o comando DRAWTO:

```
10 GRAPHICS 3
20 SETCOLOR 0,3,8
30 COLOR 1
40 PLOT 0,0
50 DRAWTO 39,19
```

\*

```
10 GRAPHICS 3
20 SETCOLOR 2,0,0
30 COLOR 1
40 PLOT 0,0
50 DRAWTO 100,0
60 DRAWTO 100,100
70 DRAWTO 0,100
80 DRAWTO 0,0
90 END
```

Para o comando PRINT:

```
10 FOR M=3 TO 8
20 GRAPHICS M
30 IF M=8 THEN SETCOLOR 2,0,0
40 COLOR 1
50 PRINT "MODO"; M
60 FOR H=0 TO 16 STEP 4
70 PLOT 0,H
80 DRAWTO 16,H
90 NEXT H
100 FOR V=0 TO 16 STEP 4
110 PLOT V,0
120 DRAWTO V,16
130 NEXT V
140 FOR D=1 TO 1000
150 NEXT D
160 NEXT M
```

*Para os modos 3 a 7*

```
10 FOR N=0 TO 2
20 GRAPHICS 3+2*N+16
30 COLOR 1
40 PLOT 0,0
50 DRAWTO 40*2 N-1, 20*2 N-1
60 COLOR 3
70 PLOT 0, 20*2 N-1
80 DRAWTO 40*2 N-1, 0
90 FOR D=1 TO 2000: NEXT D
100 NEXT N
```

*Para o modo 8*

```
10 GRAPHICS 8+16
20 SETCOLOR 2,0,0
30 COLOR 1
40 LET A=180: LET B=98
50 PLOT A,B
60 LET D=INT(4*RND(0))+1
70 IF D=1 AND B > 9 THEN LET B=B-10
```

```
80 IF D=2 AND B 182 THEN LET B=B+10
90 IF D=3 AND A 310 THEN LET A=A+10
100 IF D=4 AND A 9 THEN LET A=A-10
110 DRAWTO A,B
120 GOTO 60
130 END
```

\*

```
10 GRAPHICS 8+16: SETCOLOR 2,0,0: COLOR 1
20 FOR A=0 TO 40 STEP 2
30 PLOT A,0: DRAWTO A,191
40 NEXT A
50 FOR A=21 TO 61 STEP 2
60 PLOT A,0: DRAWTO A,191
70 NEXT A
80 FOR A=0 TO 80 STEP 2
90 PLOT 80,A: DRAWTO 160, A+40
95 PLOT 200,A: DRAWTO 280,A+40
100 NEXT A
110 FOR A=20 TO 120 STEP 2
120 PLOT 101, A: DRAWTO 181, A+40
125 PLOT 221, A: DRAWTO 301, A+40
130 NEXT A
140 FOR A=80 TO 182 STEP 2
150 PLOT A,0: DRAWTO A, 191
160 NEXT A
200 GOTO 200
999 END
```

*Para os modos 9 a 11*

```
10 GRAPHICS 9
20 FOR C=0 TO 16
30 FOR L=0 TO 16
40 SETCOLOR 4,C,L
50 NEXT L
60 NEXT C
70 GOTO 20
80 END
```

```

10 GRAPHICS 9
20 FOR C=0 TO 15
30 COLOR C
40 PLOT C+25,0: DRAWTO C+25,191
50 PLOT 56-C,0: DRAWTO 56 - C,191
60 NEXT C
70 FOR A=0 TO 15
80 SETCOLOR 4,A,0
90 FOR D=1 TO 500
100 NEXT D
110 NEXT A
120 END

```

\*

```

5 PRINT "ESCREVA OITO NÚMEROS SIMPLES
SEPARADOS POR RETURN"
10 INPUT Q, W, E, R, T, Y, U, I
20 GRAPHICS 10
30 FOR A=1 TO 8
40 POKE 704+A, 16*A+8
50 NEXT A
60 LET D=79:LET G=79:LET H=191:LET K=191
70 TRAP 130:
80 LET C=C+1:IF C=9 THEN LET C=1
90 COLOR C
100 PLOT A,S: DRAWTO D,F: DRAWTO G,H:
DRAWTO
J,K: DRAWTO A, S
110 LET A=A+Q:LET S=S+W:LET D=D-E:LET
F=F+R:LET G=G-T:LET H=H-Y:LET
J=J+U:LET K=K-I
120 GOTO 80
130 LET A=PEEK(705)
140 FOR B=705 TO 711
150 POKE B,PEEK(B+1)
160 NEXT B
170 POKE 712,A
180 FOR A=1 TO 20:NEXT A
190 GOTO 130
200 END

```

```
10 GRAPHICS 11
20 LET C=0:S=1
30 FOR A=0 TO 191
40 C=C+S
50 COLOR C
60 IF C 16 OR C 1 THEN LET S=-S:GOTO 40
70 PLOT 0,A:DRAWTO 79,A
80 NEXT A
90 FOR A=0 TO 15
100 SETCOLOR 4,0,A
110 FOR D=1 TO 250
120 NEXT D
130 NEXT A
140 END
```

## O SOM

Quando se fala em som, nos computadores, deveria antes utilizar-se o termo «sintetizador» que é, de resto, o preferido por orquestras e conjuntos quando se referem a instrumentos electrónicos. Porque é isso o que, na realidade, um desses instrumentos faz.

A diferença entre «produzir» e «sintetizar» um som é simples (numa definição superficial, claro); qualquer instrumento produz som, quer por vibração, quer por sopro, quer por percussão. Mas nenhum deles produz determinados ruídos, por exemplo, devido a limitações de escala (entre outras).

Para um equipamento electrónico, o som não passa de mais um conjunto de variações de corrente eléctrica. Assim sendo, teoricamente tal equipamento tem capacidade para produzir qualquer som de qualquer tipo, sendo apenas limitado pelo maior ou menor número de elementos para o tratamento dessas variações.

Os computadores Atari possuem um dos mais incrementados equipamentos de sintetização de som que é possível encontrar em computadores pessoais.

### 4.1. Os parâmetros

O Atari possui 4 canais de som. Quer dizer que pode produzir de um a quatro sons simultaneamente.

Por exemplo, uma determinada canção pode ser tocada com sons simples ou a duas vozes: uma primeira voz tocando a canção; uma segunda fazendo acompanhamento.

O comando que se utiliza para a produção de sons é o SOUND, o qual tem a seguinte sintaxe:

SOUND A, B, C, D

com as funções:

A - número do canal ligado (0,1,2 ou 3)

B - nota musical (0 a 255)

C - tonalidade (0 a 14)

D - volume (0 a 14)

Portanto, por exemplo a instrução

SOUND 0,29,10,10

produz, através do canal 0, a nota DÓ, na tonalidade 10 (um pouco agudo) e no volume 10.

A instrução

SOUND 0,0,0,0 desliga o canal 0.

## 4.2. Uma tabela média

No apêndice encontra-se uma tabela completa da escala musical do Atari. Todavia, apresenta-se em seguida uma tabela média que serve para usos normais.

---

Número da NOTA	NOTA musical
29	DÓ
31	ré
33	mi ou réb
35	mi
37	fa ou mi b
40	fa
42	sol ou fá b
45	sol
47	la
50	si ou lá b
53	si
57	dó ou si b
60	DÓ
64	ré



### 4.3. Programas

```
10 FOR N=0 TO 254
20 SOUND 0,N,10,10
30 NEXT N
40 END
```

\*

```
10 FOR V=14 TO 0 STEP -2
20 SOUND 0, 121,10,V
30 FOR D1=1 TO 350
40 NEXT D1
50 SOUND 0,0,0,0
60 FOR D2=1 TO 150
70 NEXT D2
80 NEXT V
90 END
```

\*

```
10 FOR D= 0 TO 14 STEP 2
20 SOUND 0,10,D,10
30 FOR D1=1 TO 350
40 NEXT D1
50 SOUND 0,0,0,0
60 FOR D2=1 TO 150
70 NEXT D2
80 NEXT D
90 END
```

\*

```
10 FOR N1=5 TO 20
20 SOUND 0,N1,0,14
30 FOR D1=1 TO 100
40 NEXT D1
50 NEXT N1
60 FOR N2=20 TO 5 STEP -1
70 SOUND 0,N2,0,10
80 FOR D2=1 TO 75
90 NEXT D2
100 NEXT N2
110 END
```

```
10 T=500
20 SOUND 0,15,2,10
30 FOR D1=1 TO 10
40 NEXT D1
50 SOUND 0,0,0,0
60 FOR D2=1 TO T
70 NEXT D2
80 T=T/1.5
90 IF T 1 THEN END
100 GOTO 20
110 END
```

\*

```
10 FOR P=50 TO 255
20 SOUND 0,P,10,4
30 NEXT P
40 FOR V=14 TO 0 STEP -0.1
50 SOUND 0,255,0,V
60 SOUND 1,10,0,V
70 SOUND 2,150,2,V
80 NEXT V
90 END
```

\*

```
10 SOUND 0,28,10,10
20 SOUND 1,32,10,10
30 GOTO 30
40 END
```

\*

```
10 FOR N1=0 TO 15
20 SOUND 0,N1,2,10
30 FOR D=1 TO 10
40 NEXT D
50 NEXT N1
60 FOR N2=14 TO 0 STEP -1
70 SOUND 0,N2,2,10
80 FOR D=1 TO 10
90 NEXT D
100 NEXT N2
110 END
```

## ARMAZENAMENTO DE DADOS

Teoricamente, a base de um sistema informático é constituída pelo computador e um televisor ou monitor. Na verdade, bastam esses dois elementos para que o sistema funcione e produza os resultados que poderiam esperar-se do mesmo.

Todavia, um dos aspectos em que um sistema informático se tornou notável foi precisamente o de executar determinadas tarefas REPETITIVAS com muito maior velocidade e, sobretudo, libertando o homem dessa mesma necessidade de repetir trabalho para as executar.

Ora, tal como temos trabalhado até este momento – e apesar de a programação evitar já a repetição, até um certo ponto – o que sucede é que, cada vez que o computador se desliga se perde todo o trabalho realizado, sendo necessário voltar a executá-lo caso se pretenda que o computador o repita. Quer dizer: se se fez um programa para desenhar figuras geométricas, por exemplo; se o computador executou esse programa; se depois o desligámos (porque, para além do mais, não se deixa o equipamento ligado permanentemente); quando quisermos que ele volte a desenhar figuras geométricas teremos de perder novamente muito tempo a reescrever o programa.

Para evitar tal absurdo, os computadores têm periféricos destinados a guardar os dados que, num dado momento, se encontram nas suas memórias, podendo mais tarde voltar a ser recuperados e introduzidos nessas mesmas memórias. Além do que alguns desses periféricos têm ainda outras utilidades, como veremos.

Os periféricos de armazenamento de dados são três – dois deles actualmente à venda no mercado, e um terceiro em fase final de desenvolvimento; o gravador de cassetes, a unidade

de disco e o cartão magnético (semelhante aos cartões de crédito utilizados actualmente pelos bancos).

## 5.1. O gravador de cassetes

Este é, sem dúvida, o periférico mais popular e difundido para armazenar dados. Para além de barato, é fácil de manipular, possui como suportes as cassetes mais ou menos vulgares que são adquiridas igualmente a muito baixo preço, e está apto a introduzir no computador centenas de programas comerciais à venda.

Se com outros computadores é possível utilizar praticamente qualquer gravador de cassetes à venda, para o Atari é necessário um gravador especial, com um cabo próprio. Trata-se dos modelos 410 ou 1010 fabricados pela Atari, sendo o último mais moderno e mais aconselhável.

Basicamente, trata-se de um gravador idêntico aos outros, pois possui as mesmas teclas: PLAY, RECORD, REWIND e FORWARD, com as mesmas funções. Portanto o seu funcionamento não põe problemas especiais de aprendizagem, para quem esteja habituado a utilizar gravadores de cassetes.

Como, certamente, não vai utilizar esse periférico apenas para carregar no computador programas adquiridos e já prontos, mas querará guardar programas feitos por si, convém comprar desde logo algumas cassetes virgens, de preferência próprias para o computador. Destas, existe uma grande variedade no mercado. Todavia, aconselho a compra da BASF PC15, pois, embora seja mais cara, garante, mais do que qualquer outra, uma gravação e conservação impecável dos dados que nela forem guardados.

Agora alguns conselhos, no que respeita às cassetes:

- 1 - após ter gravado ou carregado um programa, na ou da cassette, rebobine-a. NUNCA a guarde com a fita ao meio.
- 2 - nunca coloque as cassetes próximo de uma fonte magnética. Se o fizer, pode liquidar uma gravação. Se, por uma questão de comodidade, quiser tê-las junto do sistema, tenha o cuidado de as colocar pelo menos a 30 centímetros do televisor ou monitor.
- 3 - por esta mesma razão, o gravador de cassetes deverá

- ser colocado a uma distância mínima de 30 centímetros, relativamente ao TV.
- 4 – caso passe muito tempo sem utilizar determinada casete, convém fazer passar a fita, em alta velocidade, para um e outro lado da mesma, a fim de evitar a interpenetração magnética que inevitavelmente ocorreria.

Ao contrário de outros computadores, o Atari controla o funcionamento do seu gravador. Por tal motivo, vamos explicar como se utiliza.

### *Para carregar (LOAD) programas*

Nem todas as cassetes para o Atari são fabricadas pela mesma companhia. Assim, aparecem algumas com instruções especiais, mas essas instruções são dadas no folheto que as acompanha. Contudo, eis o procedimento genérico que deve seguir:

- 1 – ligue o gravador à corrente e ao computador, seguindo as instruções que o acompanham;
- 2 – escreva  
CLOAD
- 3 – carregue na tecla RETURN, após o que ouvirá um BIP;
- 4 – coloque a casete no gravador e, caso seja necessário, faça-a voltar ao princípio;
- 5 – carregue na tecla PLAY do gravador (não se admire pelo facto de a fita não começar a andar);
- 6 – carregue novamente na tecla RETURN. A fita começa a mover-se, ao mesmo tempo que um som BIP sairá do seu televisor;
- 7 – aguarde até que a fita pare de correr e apareça no écran a mensagem READY (significa que o programa foi correctamente carregado);
- 8 – carregue no botão STOP do gravador e rebobine a fita. O programa encontra-se em memória e pode ser utilizado normalmente.

Enquanto a fita estiver a correr não deve tocar no teclado. Se durante, ou após, a carga surgir uma mensagem de erro,

é porque houve problemas no processo de carga. Deve rebobinar a fita e tentar outra vez. Caso o erro persista, poderá suceder que haja instruções especiais para a carga de tal programa; deve ler as instruções. Pode ainda acontecer que uma cassete se tenha deteriorado e, nesse caso, não há nada a fazer.

Aliás, é um princípio ter cópias das cassetes com muita utilização. O roçar constante da fita na cabeça do gravador vai fazendo soltar o pó castanho que se encontra na superfície da mesma. E a solução é manter a tal cópia de reserva.

### *Para guardar (SAVE) programas*

Tal como se disse acima, a função mais importante do gravador de cassetes é a de poder guardar dados, directamente a partir das memórias do computador. E para passarmos à prática...

Faça, por exemplo, o seguinte programa:

```
10 PRINT "BOM DIA"  
20 PRINT,,,  
30 PRINT "ESTE É UM SIMPLES PROGRAMA", DES-  
TINADO A UTILIZAR AS POSSIBILIDADES", "QUE O  
COMPUTADOR TEM DE GUARDAR E FORNECER"  
40 PRINT "DADOS, EM E A PARTIR DE CASSETE"  
50 END
```

Após ter feito isto – e, portanto, com este curto programa em memória – execute as seguintes instruções:

- 1 – introduza uma cassete virgem no gravador, tendo o cuidado de verificar se a fita se encontra no princípio;
- 2 – escreva a instrução

CSAVE

- 3 – carregue simultaneamente nas teclas RECORD e PLAY do gravador;
- 4 – carregue na tecla RETURN; ouvirá dois BIP (para LOAD apenas ouviu um);
- 5 – aguarde até que a fita pare e surja no écran a mensagem READY, que significa que a gravação ficou correcta;
- 6 – carregue no STOP do gravador. A operação terminou.

Por uma questão de segurança, e sobretudo se se trata de um programa longo (ou muito importante), que levou horas a teclar, deve fazer uma cópia, noutra cassete. Isto é uma garantia de que, mesmo tendo havido eventualmente algum problema com uma das gravações, possui uma alternativa.

Para maior segurança ainda, pode partir a pequena patilha de plástico que se encontra na parte posterior da cassete, o que evitará uma futura desgravação accidental. (Se, mais tarde, quiser desgravar a cassete, basta tapar esse orifício com um pouco de fita-cola).

Para verificar se o programa ficou realmente gravado, dê a instrução NEW ao computador, a fim de limpar as memórias; faça CLOAD e, após ter carregado o programa que acabou de gravar, dê a instrução LIST. No écran deverá aparecer listado o programa, tal como foi elaborado.

### *Outros modos de guardar programas*

No Atari, não existem apenas os comandos CLOAD e CSAVE com a função de guardar e carregar programas. Estes são apenas os mais simples. Poderá utilizar outro par de comandos, que irão servir para guardar e carregar programas, também noutros periféricos.

Um dos pares é constituído pelas instruções normais do BASIC: LOAD, para carregar; SAVE, para gravar. Funcionam exactamente como o CLOAD e CSAVE, só que é necessário indicar qual o periférico:

LOAD "C:"

e

SAVE "C:"

Esse C, escrito com a sintaxe indicada, diz ao computador que se pretende gravar ou carregar um programa no, ou a partir do, gravador de cassetes.

Mas atenção: NÃO PODE CARREGAR UM PROGRAMA que tenha sido gravado com CSAVE, utilizando o comando LOAD, nem o contrário. Quer dizer: CSAVE e CLOAD formam um par; LOAD "C:" e SAVE "C:" outro. Todos os programas guardados com SAVE "C:" terão de ser carregados com LOAD "C:", o mesmo se passando com o outro par de comandos.

Mas então, qual a vantagem de utilizar estes comandos que, aparentemente, só vêm causar confusão? É que estes últimos permitem fazer algo de muito importante, que não é possível com os primeiros. Vamos experimentar:

- 1 - se tiver já algum programa em memória (o que fez há pouco, por exemplo) grave-o numa cassete virgem, utilizando a instrução

SAVE "C:"

- 2 - agora, para o carregar de novo, em vez de LOAD "C:" utilize a instrução

RUN "C:"

O que aconteceu foi que, além de carregar o programa, o computador o executou imediatamente, sem necessidade de qualquer instrução extra. Isto pode ser muito útil, em determinadas situações, e só é possível com estes comandos.

Um outro par de comandos utilizáveis para este processo que temos vindo a descrever é o de LIST e ENTER. E também eles, servindo embora para guardar e carregar dados, têm uma possibilidade que nenhum dos anteriores têm: a de guardar apenas partes dos programas (o que virá a ser muito útil quando se pretender, por exemplo, gravar separadamente determinadas rotinas ou sub-rotinas).

Se, por exemplo, escrever a instrução

LIST "C:",30,50

o computador só enviará para o gravador, no acto da gravação, as linhas compreendidas entre a 30 e a 50 do programa.

Mais tarde, e utilizando o comando

ENTER "C:"

poderá carregar as memórias do computador apenas com as linhas que guardou, podendo, por exemplo, utilizá-las noutra programa.



## 5.2. O Disk Drive

O Disk Drive é uma unidade mecânica capaz de gravar e fornecer informação num suporte magnético. Basicamente, tem as mesmas funções que o gravador de cassetes, utilizando um disco em vez de uma cassete com fita. Todavia, executa essa função com uma velocidade muitíssimo maior e, além disso, possui outras capacidades, como veremos.

O suporte utilizado é um disco de plástico com 5 1/4 polegadas, coberto, em ambos os lados, por uma camada de óxido metálico magnetizável. Este disco encontra-se encerrado num envelope também de plástico, provido de ranhuras através das quais as cabeças de gravação e leitura contactam o disco.

Comparando este sistema com o gravador de cassetes, podemos dizer que é muitíssimo mais rápido e que aproveita melhor a superfície de gravação já que não há intervalos entre os diversos programas (ou conjunto de dados) registados. Mas a grande vantagem é a sua flexibilidade.

Enquanto numa cassete terá de gravar e ler a informação a partir do início da fita e, sequencialmente, até ao fim, com o disco o computador vai imediata e directamente ao ponto desejado, gravando ou lendo a informação rapidamente sem necessidade de voltar atrás, como na cassete. Esta característica seria completamente impossível com a cassete.

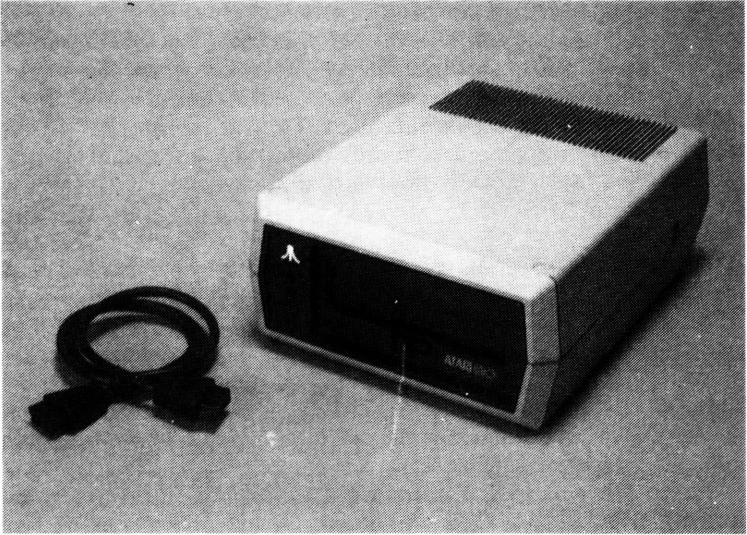
Além de tudo isto, é possível ligar até 4 unidades Drive, simultaneamente, ao computador, o que faculta um acesso directo a alguns milhões de bytes.

Existem dois modelos de Disk Drives da Atari:

- o 810, mais antigo;
- o 10150, actual e muito mais rápido e com maior capacidade.

Qualquer deles, contudo, pode ser ligado a qualquer dos modelos da Atari.

Atendendo a que o manual que o acompanha indica o modo de ligação do Disk Drive ao computador, faremos apenas a descrição do seu funcionamento, sobretudo no que respeita aos diferentes comandos e instruções que utiliza.



### *Trabalhando com o Disk Drive*

Acompanha a unidade Disk Drive, um disco – Atari DOS. Deverá ser esse o primeiro a utilizar, para uma mais fácil adaptação ao sistema.

- 1 – antes de ligar o Drive, certifique-se de que a porta frontal para o disco se encontra aberta e que não há qualquer disco inserido (se ligar ou desligar a unidade com um disco dentro, pode ser o suficiente para estragar definitivamente o disco).
- 2 – ligue o Drive. Acende-se a luz correspondente a PWR ON bem como a do BUSY LIGHT, ao mesmo tempo que a drive começa a funcionar. Após alguns segundos a BUSY LIGHT apaga-se e o drive pára.
- 3 – é esse o momento de introduzir a diskette na ranhura respectiva existente na parte da frente do Drive e, atenção, tem de ser uma diskette não virgem (sugiro a Atari DOS) com o lado A para cima e a abertura oval para trás.

- 4 – quando sentir um som (CLIK), feche a porta do drive com a alavanca própria para o efeito, colocada também na frente e à esquerda. Após uns segundos de funcionamento, o drive pára.
- 5 – ligue agora o computador e TV (ou monitor). O drive põe-se novamente em movimento durante alguns segundos, até parar e aparecer a mensagem READY no écran.

Se, por acaso, após estas operações aparecer no écran a mensagem

### BOOT ERROR

é porque alguma das operações foi executada deficientemente, ou porque foi introduzida uma diskette que o computador não pode, por alguma razão, ler.

Neste caso, o drive continuará a funcionar, imprimindo sucessivos BOOT ERRORS no écran. A luz encarnada não se apaga mas, mesmo assim, deve abrir a porta e retirar o disco. Em seguida desligue o computador e espere até que o drive páre. Nessa altura, e só nessa, desligue o drive; volte a ligá-lo; repita todas as operações acima descritas.

### *O disco DOS*

O disco Atari DOS, de que já falámos, é o principal e sem ele é impossível pôr o sistema a funcionar. DOS significa precisamente DISK OPERATING SYSTEM, o que quer dizer que é nesse disco que estão contidas instruções adicionais que permitem ao computador colocar-se em "modo drive". (Convém ter uma cópia deste disco já que se for destruído, por qualquer razão, não há maneira de pôr o computador a funcionar com o drive).

A Atari possui três versões do DOS: a DOS 1.0, a DOS 2.0 e a DOS 3.0. A primeira está desactualizada pois foi substituída pela 2.0. A versão 3.0 é a desenvolvida para o drive 1050 e possui várias vantagens relativamente à 2.0.

Após ligar o computador com o DOS no drive, uma parte deste é automaticamente carregada na RAM. A partir daí o computador pode ser utilizado de modo normal, mas também

poderá utilizar os novos comandos acrescentados ao BASIC normal.

Se, após ter feito todas as operações de carga, escrever a instrução

## DOS

o computador carrega um programa utilitário que lhe permite recorrer a operações especiais.

Primeiro surge no écran o menu seguinte:

DISK OPERATING SYSTEM II VERSION 2.0  
COPYRIGHT 1983 ATARI

- |                    |                    |
|--------------------|--------------------|
| A. DISK DIRECTORY  | I. FORMAT DISK     |
| B. RUN CARTRIDGE   | J. DUPLICATE DISK  |
| C. COPY FILE       | K. BINARY SAVE     |
| D. DELETE FILE(S)  | L. BINARY LOAD     |
| E. RENAME FILE     | M. RUN AT ADDRESS  |
| F. LOCK FILE       | N. CREATE MEM. SAV |
| G. UNLOCK FILE     | O. DUPLICATE FILE  |
| H. WRITE DOS FILES |                    |

Vejamos o que significa cada uma destas opções.

A. serve para catalogar o disco, ou seja, listar no écran os programas ou ficheiros existentes no disco; todavia, pode utilizar esta opção apenas para verificar se um determinado programa ou ficheiro existe num disco;

B. caso o seu computador seja um XL, este comando serve apenas para introduzir o BASIC;

C. serve para copiar um programa ou ficheiro para outro disco;

D. serve para apagar um ou mais programas;

E. muda o nome de um programa;

F. fecha um ficheiro;

G. abre um ficheiro;

H. copia o DOS para uma diskette nova;

I. formata um disco (primeira operação a fazer quando introduz um disco novo na drive);

J. duplica inteiramente um disco;

K. grava dados em código máquina;

L. carrega dados gravados em código máquina;

M. carrega e põe em funcionamento um programa gravado com o comando LIST "D:" (tal como a cassete);

N. abre espaço de memória no computador, gravando alguma da informação existente na mesma;

O. duplica um programa ou ficheiro.

Para gravar informação (programas ou ficheiros) no disco, terá de utilizar o BASIC. Quer dizer, não poderá fazê-lo através do menu acima descrito. Para tanto, basta escolher a opção B do menu.

Após ter escrito o seu programa, basta o comando

SAVE "D:nome.EXT"

para o gravar na diskette., "Nome" é, evidentemente, aquele que quiser dar ao programa; EXT é a extensão.

Para o carregar de novo utilize os mesmos comandos que foram descritos para a cassete, escrevendo, contudo, um D em vez de um C, no local apropriado. Se tiver mais de um drive ligado, deverá escrever D1, D2, etc., para o identificar.

O nome de um programa nunca poderá ter mais de 8 caracteres e pode gravar até a um máximo de 64 programas ou ficheiros diferentes em cada lado de um disco.

### **5.3. O cartão magnético**

Neste momento este periférico pertence, em termos de mercado, à ficção, uma vez que ainda não se encontra disponível (pelo menos para os microcomputadores). Todavia, e atendendo a que, segundo se sabe, virá a ser posto à venda em 1986, não queremos deixar de lhe fazer referência, mesmo que ao de leve.

Basicamente, trata-se de um periférico do tipo do drive para disco, mas de dimensões muito menores (a ponto de poder vir a integrar a caixa de teclado de um computador). Terá uma porta pela qual o utilizador introduzirá um cartão, pouco maior e mais espesso que um cartão de crédito. Esse cartão contém, contudo, não um suporte magnético mas sim unidades de chips (circuitos integrados) de dimensões microscópicas, capazes de armazenar e manter (mesmo depois de o cartão ser retirado do periférico) dados que poderão ser posteriormente utilizados, como se de um disco se tratasse.

Para além das vantagens evidentes de tal sistema, acrescentam-se de cada cartão poder comportar cerca de 2Mb e de não ser necessário qualquer lapso de tempo para o carregar nas memórias do computador. Até porque, nesse aspecto, funciona como se de uma cartridge se tratasse; ou seja, não ocupa qualquer espaço nas memórias RAM do micro. É como se se tratasse de uma memória adicional.

## O SOFTWARE

Software é o termo universal que designa a parte não física dos sistemas informáticos. Quer dizer, toda a informação que é colocada, tratada e modificada nas memórias do computador e guardada em suportes apropriados, dos quais falámos no último capítulo.

Em suma, o software é aquilo a que vulgarmente se chama programas para um computador.

Para cada computador existe normalmente no mercado um número razoável de programas comerciais, quer dizer, programas elaborados e testados e que se destinam às mais diversas aplicações. Todavia, e se é verdade que esse factor é até decisivo na escolha e aquisição de um sistema informático, tal sistema possui como uma das mais importantes características a possibilidade de adaptação a utilizações específicas individuais. Por isso, de resto, o micro tem uma linguagem com múltiplos comandos que, como já vimos, permitem dar as mais diversas ordens ao sistema e dele obter, conseqüentemente, a execução das mais variadas tarefas.

Portanto, uma das coisas que o novo utente de um sistema informático deve aprender, e bem depressa, é a programar. Ou seja, criar um determinado conjunto de instruções que levem o sistema a responder eficientemente a eventuais necessidades.

E, se é fácil aprender a linguagem do computador bem como os resultados que cada comando ou instrução produzem (pois, para além do mais, não passam de umas dezenas os vocábulos que a compõem), não é menos verdade que existem algumas regras, igualmente fáceis, essenciais a um trabalho efectivamente produtivo. É, pois, basicamente para dar a co-

nhecer essas regras, na prática, que iremos, na primeira parte deste capítulo, acompanhar, a par e passo, a elaboração de um programa.

## 6.1. Escrever um programa

Ligar o computador, escrever o número 10, por exemplo, e começar, só nesse momento, a pensar no que se vai fazer, é um absurdo inútil – a menos que se pretenda somente adquirir prática na manipulação do teclado e observação do funcionamento de comandos ou instruções. Se o que se pretende, de facto, é conseguir que o sistema realize determinado trabalho, então talvez seja mais aconselhável não ligar o computador antes de se saber, com alguma exactidão, quais os objectivos a atingir.

Assim, supondo por exemplo que pretendo fazer um jogo, a primeira coisa a pensar é:

– qual o tipo de jogo?

Trata-se de um jogo de sorte/azar, de aventuras, de lógica, de simulação... enfim, qual o tipo. Isto é fundamental porque da minha opção vão depender as consultas que eventualmente tenha de realizar, os comandos e instruções necessários, o tipo de cálculos e operações que o computador terá que realizar, etc.

Por outro lado, é igualmente fundamental decidir com que objectivos vou fazer um programa de jogo: uso particular ou possível comercialização. A importância deste segundo ponto vem do facto de, caso pretenda comercializá-lo, não escolher, por exemplo, um programa semelhante a um já existente no mercado.

Decididos os dois primeiros pontos e, por uma questão de simplicidade, admitindo que decidi elaborar um «Jogo do Galo», devo passar à planificação. Muitos programadores iniciam frequentemente a elaboração de um programa sem qualquer planificação prévia. O resultado é, quase sempre, um programa demasiado complexo e, não raro, até sem chegar a funcionar convenientemente. Para além do que, uma planificação permite posteriores alterações com muito mais facilidade e eficácia.



## Planificação

- 1 - Definir os elementos do jogo: número de jogadores (2 ou 1 e o computador); escolher o tamanho do tabuleiro (3×3 quadrados);
- 2 - Elaborar uma tabela sequencial do desenvolvimento do jogo:
  - 2.1. Desenhar o tabuleiro no écran;
  - 2.2. Perguntar os nomes dos jogadores:
    - 2.2.1. Perguntar o do primeiro;
    - 2.2.2. Perguntar o do segundo;
  - 2.3. Determinar qual dos jogadores joga primeiro;
  - 2.4. Dizer ao jogador para jogar;
  - 2.5. Verificar a validade da jogada (não pode, por exemplo, jogar para uma casa ocupada);
  - 2.6. Imprimir a marca do jogador no quadrado respectivo;
  - 2.7. Verificar se ganhou:
    - 2.7.1. Se SIM, perguntar se quer outro jogo;
      - 2.7.1.1. Se quer, ir para a primeira linha;
      - 2.7.1.2. Se não, terminar o jogo;
    - 2.7.2. Se NÃO, voltar ao ponto 2.4.

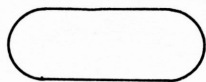
### *O organigrama*

A forma usual de realizar a planificação é através de um organigrama, pois o mesmo permite uma visualização imediata de toda a estrutura do programa, para além de conter indicações sobre os pontos essenciais em que certas operações se realizam.

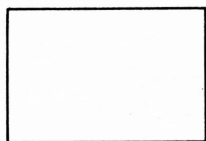
Há uma série de quatro símbolos universais para utilização nos organigramas, conforme pode ver-se na figura. (Na verdade, há uma tabela mais complexa e vasta que, todavia, serve para programas bem mais complexos; decidimo-nos por esta por ser mais que suficiente para usos comuns).

Portanto, e a partir do desenvolvimento realizado, vamos fazer o respectivo organigrama.

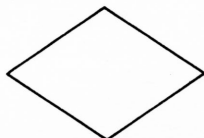
## SÍMBOLOS PARA ORGANIGRAMAS



INÍCIO/FIM



PROCESSAMENTO



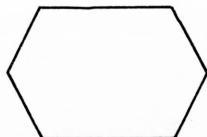
DECISÃO



INPUT/OUTPUT



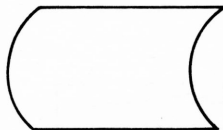
DIRECÇÃO  
DO PROGRAMA



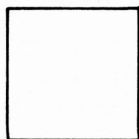
PROCESSO  
PREDEFINIDO



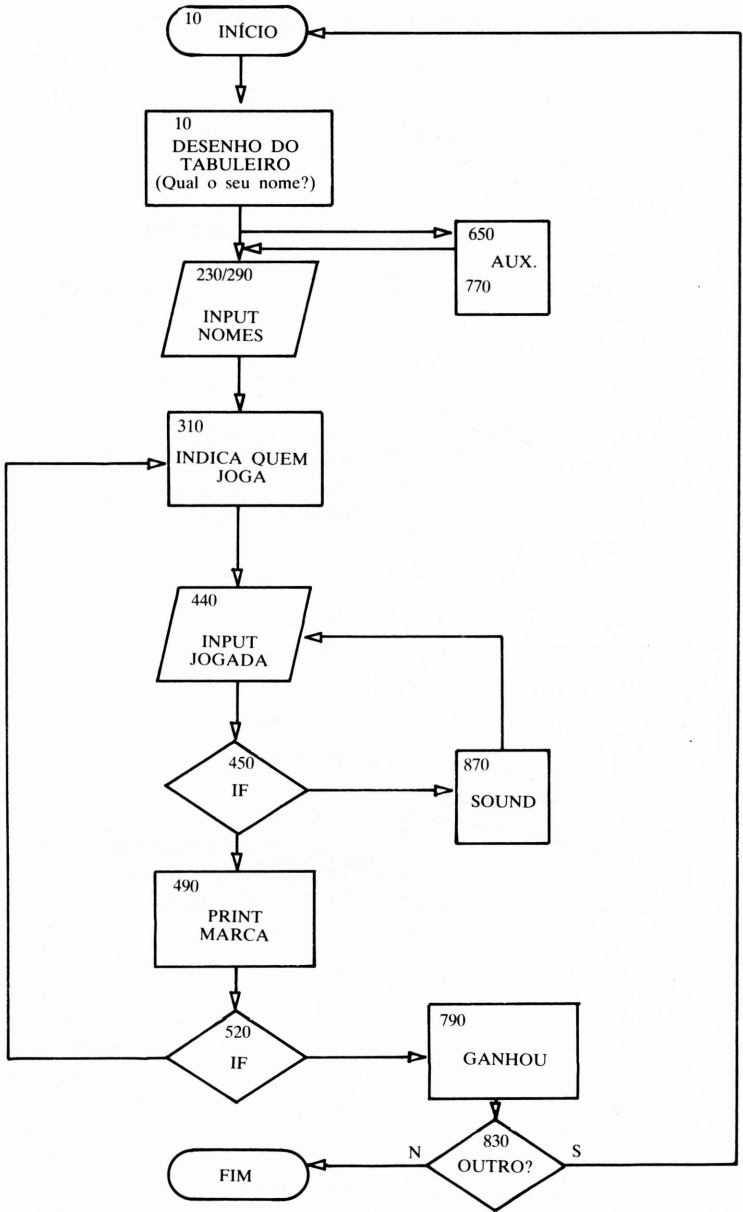
TECLADO



DISCO OU  
FONTE DE DADOS



OPERAÇÃO  
AUXILIAR



### *Lista de variáveis*

É evidente que terão de ser usadas variáveis, no programa. Assim, o melhor é tentar defini-las antecipadamente, pelo menos as que são, desde já e neste momento, previsíveis.

- A – variável reservada para fins genéricos;
- B – variável reservada para fins genéricos;  
(convém sempre reservar duas ou mais variáveis das quais possamos socorrer-nos, em caso de necessidade; se não precisarmos delas, não há problemas);
- D – tempo de espera;
- P – contador para dizer quem joga;
- M – número (coordenada) de uma jogada;
- A\$ – variável alfanumérica para fins genéricos;
- P1\$ – nome do primeiro jogador;
- P2\$ – nome do segundo jogador;
- G(3,3) – Memória reservada para guardar a situação de cada quadrado da grelha.

### *Blocos do programa*

Face à planificação e organigrama, é possível e desejável decidir em que linha se inicia cada bloco, uma vez que fazê-los seguidos resultaria numa confusão. Assim:

- 50 – desenho do tabuleiro;
- 180 – nomes dos jogadores;
- 310 – determinar qual joga primeiro;
- 370 – pede para jogar.

### *O programa*

Finalmente estamos em condições de teclar o programa, progressiva e cuidadosamente, guiados pela planificação, organigrama e tabela de variáveis.

```
10 GRAPHICS 0:POKE 752,1
20 SETCOLOR 2,0,0
30 CLR:DIM P1$(20),P2$(20),A$(1),G(3,3)
```

```

40 FOR A=0 TO 3:FOR B=0 TO 3:G(A,B)=0:NEXT
   B:NEXT A
50 COLOR 124
60 PLOT 5,0:DRAWTO 5,10
70 PLOT 9,0:DRAWTO 9,10
90 COLOR 18
100 PLOT 2,3:DRAWTO 12,3
110 PLOT 2,7:DRAWTO 12,7
120 COLOR 19
130 PLOT 5,3:PLOT 9,3
140 PLOT 5,7:PLOT 9,7
150 POSITION 15,4:PRINT "123"
160 POSITION 15,5:PRINT "456"
170 POSITION 15,6:PRINT "789"
180 PRINT "JOGADOR 1"
200 PRINT "QUAL O SEU NOME?"
210 GOSUB 650
220 POSITION 25,13
230 INPUT P1$
240 GOSUB 770
250 PRINT "JOGADOR 2"
260 PRINT "QUAL O SEU NOME?"
270 GOSUB 650
280 POSITION 25,13
290 INPUT P2$
300 GOSUB 770
310 LET F=INT(2*RND(1))+1
320 IF F=1 THEN PRINT P2$;
330 IF F=2 THEN PRINT P1$;
340 PRINT " JOGA PRIMEIRO."
350 FOR D=1 TO 500:NEXT D
360 REM - LOOP PRINCIPAL -
370 GOSUB 770
380 F=F+1:IF F=3 THEN F=1
390 IF F=1 THEN PRINT P1$
400 IF F=2 THEN PRINT P2$;
410 PRINT "JOGA";
420 IF F=1 THEN PRINT " (X)"
430 IF F=2 THEN PRINT " (O)"
440 PRINT "QUAL A SUA JOGADA";:INPUT M
450 IF M 1 OR M 9 THEN GOSUB 870:GOTO 440

```

```

460 IF M 4 THEN IF G(M,1) 0 THEN GOSUB
      870:GOTO 440
470 IF M 3 AND M 7 THEN IF G(M-3,2) 0 THEN
      GOSUB 870:GOTO 440
480 IF M 6 THEN IF G(M-6,3) 0 THEN GOSUB
      870:GOTO 440
490 IF F=1 THEN A$="X"
500 IF F=2 THEN A$="O"
510 POSITION 4*M-1-12*INT((M-1)/3),
      INT((M-1)/3)*4+1:PRINT A$
520 IF M 4 THEN G(M,1)=F
530 IF M 3 AND M 7 THEN G(M-3,2)=F
540 IF M 6 THEN G(M-6,3)=F
550 IF G(1,1)=F AND G(1,2)=F AND G(1,3)=F
      THEN GOTO 790
560 IF G(1,1)=F AND G(2,2)=F AND G(3,3)=F
      THEN GOTO 790
570 IF G(1,1)=F AND G(2,1)=F AND G(3,1)=F
      THEN GOTO 790
580 IF G(3,1)=F AND G(2,2)=F AND G(1,3)=F
      THEN GOTO 790
590 IF G(3,1)=F AND G(3,2)=F AND G(3,3)=F
      THEN GOTO 790
600 IF G(1,2)=F AND G(2,2)=F AND G(3,2)=F
      THEN GOTO 790
610 IF G(1,3)=F AND G(2,3)=F AND G(3,3)=F
      THEN GOTO 790
620 IF G(1,2)=F AND G(2,2)=F AND G(3,2)=F
      THEN GOTO 790
630 GOTO 370
640 REM - SETA EM FLASH -
650 POKE 752,1
660 FOR B=1 TO 3
670 POSITION 11,12
680 PRINT " --"
690 FOR D=1 TO 50:NEXT D
700 POSITION 11,12
710 PRINT " "
720 FOR D=1 TO 50:NEXT D
730 NEXT B
740 POKE 752,0

```

```

750 RETURN
760 REM - APAGAR TEXTO -
770 POSITION 2,12:PRINT CHR$(156);CHR$(156);
780 RETURN
790 GOSUB 770:IF F=1 THEN PRINT P1$;
800 IF F=2 THEN PRINT P2$;
810 PRINT " GANHOU!!!"
820 PRINT
830 PRINT "QUER JOGAR OUTRA VEZ? (S/N)";
840 INPUT A$
850 IF A$="S" THEN 10
860 IF A$="N" THEN END
870 SOUND 0,15,2,10:FOR D=1 TO 100:NEXT
    D:SOUND 0,0,0,0
880 POSITION 2,13:PRINT CHR$(156);
890 RETURN

```

## 6.2 Testar e alterar

Os ingleses chamam à operação de teste e alteração de um programa, DEBUGGING. Pela simples razão de que aos erros, em informática, chamam BUGs. Portanto, debugging significa «tirar os erros».

Ora, uma vez terminado um programa, há duas coisas indispensáveis a fazer:

- 1 - testá-lo, para verificar se funciona como tínhamos previsto e pretendemos;
- 2 - corrigir-lhe erros de funcionamento, se existirem, e melhorá-lo por forma a que execute as tarefas que pretendemos de modo melhor; ou, até, que execute tarefas adicionais que não prevíamos inicialmente.

Vamos analisar o programa e fazer tais verificações e correcções.

Se já tentou pô-lo a funcionar, verificou certamente que há um erro, de facto (que incluímos propositadamente, para permitir, na prática, explicar esta segunda parte da programação):

- após ter sido dito ao computador que o jogador 1 se

chamava José, por exemplo, e o jogador 2 Mário, ao imprimir resultados e nomes para a jogada seguinte, o computador inverte os nomes.

Que fazer? Simples: listar o programa e começar a verificar, desde a primeira linha (quando tiver prática irá verificar directamente no bloco em que sabe que as variáveis correspondentes aos nomes se encontram), onde é que há uma instrução errada, ou a atribuição de um valor errado a uma determinada variável.

E logo nas linhas 320 e 330 surge o erro:

```
320 IF F=1 THEN PRINT P2$;  
330 IF F=2 THEN PRINT P1$;
```

Na verdade, e para tudo bater certo, estas linhas deveriam ser:

```
320 IF F=1 THEN PRINT P1$;  
330 IF F=2 THEN PRINT P2$;
```

uma vez que o F=1 indica o jogador 1 cujo nome foi guardado na variável P1\$, o mesmo sucedendo em relação ao jogador 2. De resto, isto pode ser verificado nas linhas 390 e 400:

```
390 IF F=1 THEN PRINT P1$;  
400 IF F=2 THEN PRINT P2$;
```

que, como é óbvio, estão correctas.

Não tem mais nada a fazer que corrigir as linhas, utilizando os comandos respectivos já anteriormente explicados.

Uma outra coisa que salta à vista, ao listar o programa, é a repetição de certas linhas ao longo do mesmo. E se isto não tem importância num programa deste tamanho, o mesmo se não pode dizer quando se elaboram programas de grandes dimensões e com grande número de linhas. É que quanto mais linhas um programa tiver mais memória gasta ao computador. E a memória deve ser economizada, sempre que possível, já que está limitada pelas capacidades do micro. Por isso se devem procurar soluções tendentes a diminuir o número de linhas, sempre que possível. Até porque tais soluções não raro aceleram a execução do programa.

Vejamos então o que é possível fazer e, para tanto, vamos apenas mostrar essa possibilidade numa parte do programa.



Escolhemos o bloco compreendido entre as linhas 190 e 300 – 12 linhas, portanto, que podem ser reduzidas a quase metade. Eis como:

- se as olharmos com atenção, veremos que o bloco constituído pelas linhas 190 a 240 é igual ao das linhas 250 a 300, mudando apenas os números 1 e 2.
- se conseguirmos fazer com que o computador passe duas vezes por um desses blocos, de cada uma das vezes lendo um número diferente, temos o problema resolvido.

Ora isto é fácil. Basta utilizar um ciclo FOR...NEXT, ou seja, um ciclo de repetição. Vejamos como.

Tratando-se de um ciclo que provoca repetição, será dentro do mesmo que terá de ficar o bloco a repetir. Portanto teremos de iniciar com a primeira instrução do ciclo:

```
185 FOR A=1 TO 2
```

sendo A a variável que utilizaremos para mudar os números, e sendo o seu valor 1 e 2, sucessivamente, porque só pretendemos dois valores; se fossem 4, escreveríamos FOR A=1 TO 4, claro.

Agora vamos introduzir o bloco, aproveitando o primeiro que lá se encontra mas fazendo duas pequenas alterações, onde há variação de números. Na linha 190,

```
190 PRINT "JOGADOR ";A
```

e na linha 230,

```
230 INPUT PA$
```

A linha 250 passa a ser,

```
250 NEXT A
```

e as linhas 260, 270, 280, 290 e 300 apagam-se simplesmente. Portanto conseguimos poupar cinco linhas, o que significa umas centenas de bytes a menos a ocuparem as memórias do micro.

Mais operações deste género podem ser feitas neste programa. Veja se as descobre, como prática.

### 6.3. 4 programas simples

Atendendo ao desenvolvimento progressivo na prática de utilização dos computadores Atari, vamos listar quatro programas simples que utilizam várias das potencialidades do computador.

À medida que melhor for entendendo a programação, poderá evidentemente melhorá-los e desenvolvê-los muito, a ponto de se tornarem programas de qualidade razoável.

#### *Compositor musical*

Este programa permite-lhe introduzir as notas musicais, bem como o seu tempo, juntando-as posteriormente para tocar uma canção. Para introduzir cada nota ser-lhe-ão pedidos quatro valores:

- 1 - o nome da nota;
- 2 - um espaço para uma nota pura ou um sinal se é sustentada;
- 3 - a oitava (um número entre 1 e 3);
- 4 - o tempo da nota:
  - S - 1/16
  - E - 1/8
  - Q - 1/4
  - H - 1/2
  - W - 1

```
50 DIM INP$(5):DIM A$(3)
60 DIM P(90):DIM L(90)
70 DIM L$(3)
100 GRAPHICS 0
110 POSITION 10,4
120 PRINT "COMPOSITOR MUSICAL"
130 NM=1
135 PRINT "1-TOCAR": PRINT "2-TERMINAR"
140 INPUT INP$
150 IF INP$="1" THEN NM=NM-1:GOTO 1000
160 IF INP$="2" THEN GRAPHICS 0:END
170 READ A$,B,C
```

```

200 IF A$=INP$(1,2) THEN 300
210 GOTO 150
300 D=VAL(INP$(3,3))
310 IF B D THEN GOTO 150
400 P(NM)=C
500 L$=INP$(4,4)
510 IF L$="S" THEN L=25
520 IF L$="E" THEN L=50
530 IF L$="Q" THEN L=100
540 IF L$="H" THEN L=200
550 IF L$="W" THEN L=400
560 L(NM)=L
600 SOUND 1,P(NM),14,15
610 FOR LOOP1=1 TO L(NM):NEXT LOOP1
620 SOUND 1,0,0,0
700 NM=NM+1
710 RESTORE
720 GOTO 140
1000 FOR LOOP2=1 TO NM
1010 SOUND 1,P(LOOP2),14,15
1020 FOR LOOP3=1 TO L(LOOP2):NEXT LOOP3
1030 NEXT LOOP2
2000 SOUND 1,0,0,0
2010 NM=NM+1
2020 GOTO 140
5000 DATA DO,1,251,DO Pt,1,230,RE,1,217
5010 DATA RE Pt,1,204,MI,1,193,FA,1,182
5020 DATA FA Pt,1,173,SOL,1,162,SOL Pt,1,153
5030 DATA LA,1,144,LA Pt,1,136,SI,1,128
5040 DATA DO,2,126,DO Pt,2,114,RE,2,108
5050 DATA RE Pt,2,102,MI,2,96,FA,2,91
5060 DATA FA Pt,2,85,SOL,2,81,SOL Pt,2,76
5070 DATA LA,2,72,LA Pt,2,68,SI,2,64
5080 DATA DO,3,63,DO$,3,57,RE,3,53
5090 DATA RE Pt,3,50,MI,3,47,FA,3,45
5100 DATA FA Pt,3,42,SOL,3,40,SOL Pt,3,37
5110 DATA LA,3,35,LA Pt,3,33,SI,3,31

```

### *Cálculos*

Os computadores são basicamente máquinas de calcular.

Assim, é muito fácil desenvolver programas de matemática para os mesmos. Veja como um programa tão pequeno permite realizar tantas operações.

```
10 GRAPHICS 0
20 DEG
100 POSITION 9,1
110 PRINT "CALCULADORA"
120 PRINT
130 PRINT "FUNÇÕES:"
140 PRINT
150 PRINT "1. SOMA                7. SENO"
160 PRINT "2. SUBTRACÇÃO        8. COSENO"
170 PRINT "3. MULTIPLICAÇÃO     9. TANGENTE"
180 PRINT "4. DIVISÃO           10. LOGARITMO"
190 PRINT "5. POTENCIAS"
195 PRINT "6. RAÍZ QUADRADA"
210 POSITION 5,15
220 PRINT NM;"
225 PRINT:PRINT
230 INPUT NM
240 PRINT "QUAL A FUNÇÃO?"
250 INPUT FUNC:"INDICANDO O SEU NUMERO."
300 ON FUNC GOTO 500,600,700,800,900,1000,
    1100,1200,1300,1400
310 GOTO 210
500 INPUT NM2
510 NM=NM+NM2
520 GOTO 210
600 INPUT NM2
610 NM=NM-NM2
620 GOTO 210
700 INPUT NM2
710 NM=NM*NM2
720 GOTO 210
800 INPUT NM2
810 NM=NM/NM2
820 GOTO 210
900 NM=NM*NM
910 GOTO 210
1000 NM=SQR(NM)
1010 GOTO 210
```

```

1100 NM=SIN(NM)
1110 GOTO 210
1200 NM=COS(NM)
1210 GOTO 210
1300 NM=SIN(NM)/COS(NM)
1310 GOTO 210
1400 NM=CLOG(NM)
1410 GOTO 210
1500 NM=INT(NM)
1510 GOTO 210

```

### *Desenhador*

E aqui está um programa que, apesar de pequeno, constitui um poderoso instrumento de desenho. Com as teclas indicadas a seguir poderá mover o lápis em 8 direcções, utilizar 4 cores e escrever texto.

São as seguintes as teclas e as respectivas funções:

TECLA	FUNÇÃO
Y	Para cima
B	Para baixo
G	Para a esquerda
H	Para a direita
Q	Cor azul claro
W	Cor de laranja
E	Cor azul escuro
R	Apagar

E eis o programa que permite desenhar:

```

10 GRAPHICS 7+16
20 COLOR 2
30 X=0:Y=0
40 PLOT 0,0
100 POKE 764,255
110 K=PEEK(764):IF K=255 THEN GOTO 110
120 IF K=47 THEN COL=2:GOTO 400
130 IF K=46 THEN COL=1:GOTO 400
140 IF K=42 THEN COL=3:GOTO 400
145 IF K=40 THEN COL=0:GOTO 400

```

```

150 IF K=43 THEN Y=Y-1:GOTO 200
160 IF K=61 THEN X=X-1:GOTO 200
170 IF K=57 THEN X=X+1:GOTO 200
180 IF K=21 THEN Y=Y+1:GOTO 200
190 GOTO 100
200 IF Y 0 THEN Y=0
210 IF Y 95 THEN Y=95
220 IF X 0 THEN X=0
230 IF X 159 THEN X=159
235 IF COL=0 THEN GOSUB 500
240 DRAWTO X,Y
250 GOTO 100
400 PLOT X,Y
420 COLOR COL
430 GOTO 100
500 COLOR 2
510 PLOT X,Y
520 FOR D=1 TO 50:NEXT D
530 COLOR 0
540 PLOT X,Y
550 RETURN

```

### *Jogo da caça*

Este é um jogo, igualmente simples mas com alguma qualidade e que tem, de qualquer modo, o mérito de mostrar algumas formas de fazer mover objectos no écran, mesmo em Modo 0.

Os controlos são:

```

CONTROL-T - move o utilizador
INVERSE -S - move o monstro
CONTROL-L - move a flor

```

E o programa é o seguinte:

```

10 GRAPHICS 0:CLR
20 DIM A$(1):POKE 752,1
30 SETCOLOR 2,0,0
50 COLOR 160

```

```

60 PLOT 0,0
70 DRAWTO 38,0
80 DRAWTO 38,22
90 DRAWTO 0,22
100 DRAWTO 0,0
110 PLOT 0,0
130 LET PX=INT(37*RND(1))+1
140 LET PY=INT(21*RND(1))+1
150 LET MX=INT(37*RND(1))+1
160 LET MY=INT(21*RND(1))+1
170 LET CX=INT(37*RND(1))+1
180 LET CY=INT(21*RND(1))+1
190 LET OPX=PX
200 LET OPY=PY
210 LET OMX=MX
220 LET OMY=MY
230 LET OCX=CX
240 LET OCY=CY
250 IF PX=MX AND PY=MY THEN 130
260 IF PX=CX AND PY=CY THEN 130
270 IF MX=CX AND MY=CY THEN 130
290 S=STICK(0)
300 IF S=6 OR S=10 OR S=14 THEN LET PY=PY-1
310 IF S=5 OR S=9 OR S=13 THEN LET PY=PY+1
320 IF S=9 OR S=10 OR S=11 THEN LET PX=PX-1
330 IF S=5 OR S=6 OR S=7 THEN LET PX=PX+1
350 IF PX 1 THEN LET PX=10:GOSUB 690
360 IF PX 37 THEN LET PX=28:GOSUB 690
370 IF PY 1 THEN LET PY=10:GOSUB 690
380 IF PY 21 THEN LET PY=12:GOSUB 690
390 IF OPX PX OR OPY PY THEN POSITION
    OPX,OPY:PRINT " "
400 POSITION PX,PY:PRINT CHR$(20)
410 LET OPX=PX:LET OPY=PY
430 IF PX=CX AND PY=CY THEN 790
450 IF MX PX THEN LET MX=MX+1
460 IF MX PX THEN LET MX=MX-1
470 IF MY PY THEN LET MY=MY+1
480 IF MY PY THEN LET MY=MY-1
490 POSITION OMX, OMY:PRINT " "
500 POSITION MX, MY:PRINT CHR$(147)

```

```

510 LET OMX=MX:LET OMY=MY
530 IF MX=PX AND MY=PY THEN 860
550 LET R=INT(8*RND(1))+1
560 IF R=1 OR R=2 OR R=8 THEN LET CY=CY-1
570 IF R=4 OR R=5 OR R=6 THEN LET CY=CY+1
580 IF R=6 OR R=7 OR R=8 THEN LET CX=CX-1
590 IF R=2 OR R=3 OR R=4 THEN LET CX=CX+1
600 IF CX 1 THEN LET CX=1
610 IF CX 37 THEN LET CX=37
620 IF CY 1 THEN LET CY=1
630 IF CY 21 THEN LET CY=:21
640 IF CX OCX OR CY OCY THEN POSITION
    OCX,OCY:PRINT" "
650 POSITION CX,CY:PRINT CHR$(16)
660 LET OCX=CX:LET OCY=CY
670 GOTO 290
690 FOR A=1 TO 3
700 SOUND 0,1,2,14
710 FORD D=1 TO 10
720 NEXT D
730 SOUND 0,0,0,0
740 FOR D=1 TO 5
750 NEXT D
760 NEXT A
770 RETURN
790 POSITION 14,10:PRINT "BOA BOLA!"
800 FOR A=200 TO 0 STEP -1
810 SOUND 0,A,10,10
820 SOUND 1,A+1,10,10
830 NEXT A
840 GOTO 910
860 POSITION 14,10:PRINT "PERDEU..."
870 FOR A=0 TO 200
880 SOUND 0,A,10,10
890 SOUND 1,A+1,10,10
900 NEXT A
910 SOUND 0,0,0,0:SOUND 1,0,0,0
920 POSITION 7,11:PRINT "OUTRO JOGO?(S/N)";:
    INPUT A$
930 IF A$="S" THEN 10
940 IF A$="N" THEN GRAPHICS 0:END
950 GOTO 920

```



## 6.4. Programas comerciais

Em Portugal, habitualmente, quando se inicia a comercialização de alguma coisa nota-se pouca preocupação com os extras ou acessórios o que, para além de ser um péssimo serviço que se presta ao consumidor, não deixa de mostrar falta de visão comercial. Em países desenvolvidos, qualquer médio ou mesmo pequeno empresário sabe que qualquer investimento inicial no material só será verdadeiramente compensado em termos económicos, com a venda continuada (e sempre amplificada) dos extras ou acessórios.

De qualquer modo, é em Portugal que este livro é publicado e, portanto, é para essa realidade que é escrito. Apesar disso, e porque nada impede que qualquer um encomende directamente os programas aos Estados Unidos, Canadá ou Inglaterra, decidi listar uma série de software comercial disponível nesses países. Pode ser até que venha a servir de incentivo às firmas que, neste momento, importam e comercializam o Atari em Portugal.

**ATARIWRITER** – é um processador de texto muito completo, fabricado e vendido pela Atari. Serve para todos os modelos pois necessita apenas de 16K. Pode ser adquirido em versão cassette, diskette ou cartridge.

**STOCK MANAG.** – programa de controlo de «stocks», também da Atari mas exigindo 32K pelo que apenas pode ser utilizado pelos modelos a partir do 800. É fornecido em diskette ou cartridge.

**ASTROLOGIA** – programa com o qual é possível fazer horóscopos. Fabricado pela Elcomp (Box 1194, Pomona CA 91769, Estados Unidos), exige 48K e é fornecido em cassette ou disco.

**BIORRITMO** – como o próprio nome indica, destina-se a calcular mapas de biorritmo. Fornecido pela Atari, precisa apenas de 8K e é fornecido em cassette ou cartridge.

**PALAVRAS CRUZADAS** – programa que cria problemas de palavras cruzadas, tendo a grande vantagem de permitir construir as grelhas e escolher os vocábulos, pelo que serve igualmente para fins educacionais. Produzido pela L & S Computerware (1008 Stewart Dr. – Sunnyvale,

CA 94086 – Estados Unidos), exige 40K e é fornecido em disco.

**PAÍSES E CAPITAIS** – programa de Geografia muito completo com países e capitais de todo o mundo. Produzido pela Atari e fornecido em cassete, necessitando apenas de 16K.

**BASQUETE** – jogo de basquetebol produzido pela Atari é fornecido em cartridge, necessitando de 8K e joystick.

**BATALHA DA NORMANDIA** – jogo de simulação da invasão dos aliados em 1944. Produzido pela Strategic Simulations Inc. (883 Stierlin Rd., Building A200 – Mountain View, CA 94043 Estados Unidos), necessita de 32K e é fornecido em cassete ou diskete.

**FUTEBOL** – produzido pela Gamestar, Inc. (1302 State St. – Santa Bárbara, CA 93101 – Estados Unidos), necessitando de 24K e sendo fornecido em cassete ou disco, é um bom jogo de simulação.

**ODISSEIA** – jogo de simulação espacial produzido pela Adventure Internacional (Box 3435, Longwood, FL 32750 – Estados Unidos), fornecido em cassete ou disco e ocupando 24K.

**BLUE MAX** – jogo de simulação de aviação e combate (já conhecido para o Spectrum), produzido pela mesma companhia, ocupando 32K e fornecido em cassete ou disco.

**DONKEY KONG** – é o tradicional jogo do King-Kong, produzido pela Atari e fornecido em cartridge que ocupa 16K.

**FINAL FLIGHT** – é um espectacular programa de simulação de voo, fabricado pela MMG Micro Software (Box 131, Marlboro, NJ 07746, E.U.A.), fornecido em cassete ou disco e ocupando 24K.

**MISSILE COMAND** – fabricado pela Atari e utilizando 8K é, apesar disso, um bom jogo de simulação de defesa e ataque estratégico. É fornecido em cartridge e ocupa 8K.

**POLE POSITION** – também fabricado pela Atari e fornecido em cartridge, é o tradicional jogo de Fórmula 1 (também conhecido no Spectrum) que utiliza 16K.

**SEA DRAGON** – trata-se de um simulador de um submarino

atómico, fabricado pela Adventure International; é fornecido em cassete ou disco, utilizando 16K.

**ABC BASIC COMPILER** – programa compilador de Basic produzido pela Monark Data Systems Inc.

**ATARI LOGO** – programa produzido pela Atari e destinado a utilização da linguagem Logo, especialmente desenvolvida para utilização em fins educacionais.

**GRAPHICS MASTER** – semelhante ao Paint Plus do Spectrum, produzido aliás pela mesma companhia, permite desenhar e pintar de forma espectacular.

**DISASSEMBLER** – desassemblador muito rápido e versátil produzido pela Adventure Internacional.

Muitos outros se poderiam indicar. Todavia, sendo relativamente fácil obter catálogos nas companhias cujos nomes e endereços incluímos em alguns programas, limitamos a nossa lista aconselhando o leitor a escrever para as ditas empresas, uma vez que existem realmente milhares de programas para os computadores Atari.

## EXPANSÃO DO SISTEMA

É evidente que muito e variado trabalho pode ser executado com o sistema básico de um Atari: o computador, um monitor ou televisor, e um gravador de cassetes ou unidade(s) drive. Contudo, a adição de outros periféricos (além dos que indicámos no último capítulo) permite tirar um muito maior rendimento das potencialidades do sistema.

Na verdade, é natural que se pretenda, por exemplo, passar a papel um texto ou um desenho; controlar manualmente cursores ou movimentos de um character no écran (para jogos ou desenho); entrar em comunicação directa via linhas telefónicas, com outros computadores ou, dentro de pouco tempo, com bancos de dados; etc. Os benefícios que advêm destas possibilidades ultrapassam largamente os seus custos, podendo mesmo tornar-se directamente rentáveis, em determinadas situações.

Para que isto seja possível, é necessário adquirir periféricos, ou seja, equipamentos que são ligados ao sistema para execução de tarefas específicas. É, pois, de alguns deles que falaremos em seguida.

### *A impressora*

Supondo até que nunca venha a precisar de escrever um texto em papel, assim mesmo uma impressora é, pelo menos a partir de um certo momento, um acessório indispensável caso decida fazer programas (e quem os não faz?). Quando se teclam programas grandes e, sobretudo, mais complexos (mas também muito mais eficientes), não é prático estudá-los, fazer-lhes correcções imediatas ou desenvolvê-los no écran.



O papel torna-se imperativo para fazer a listagem do programa, nas suas várias fases, tendo ainda a vantagem de nele se poderem marcar, manualmente, sectores do programa; assinalar correcções; indicar alterações; etc. E seria certamente arcaico copiar um programa do écran para o papel, à mão, quando uma impressora poderá fazê-lo em minutos.

Falar de um printer específico – um dos da Atari, por exemplo – seria absurdo. Para além de existirem centenas de impressoras, todas com as mesmas funções básicas e obedecendo a códigos idênticos, qualquer delas servindo para qualquer computador, a escolha depende das necessidades específicas de cada um. Assim, focaremos apenas o funcionamento do computador com a impressora.

Os comandos são os mesmos utilizados para os outros periféricos, embora alguns com funções ligeiramente diversas.

Para listar um programa que se encontra em memória, basta,

LIST "P:"

depois de ter ligado a impressora e de lhe ter colocado papel, claro.

Para escrever directamente na impressora utiliza-se o comando LPRINT, com a sintaxe seguinte:

LPRINT "Escrevendo na impressora"

o que fará, tal como para o écran, com que a impressora imprima de imediato a frase indicada entre aspas.

Para fazer uma cópia de um écran (operação que pode ser necessária após o desenho de um gráfico, por exemplo), basta a seguinte linha:

POKE 838,166:POKE 839,238

Para voltar ao normal, após uma cópia de écran,

POKE 838,244:POKE 839,241

De resto, uma impressora pode executar uma série de impressões diversas. Todavia isso já depende dos códigos de cada marca e modelo. E esses códigos vêm indicados no manual da impressora respectiva.

### *O Modem*

O modem é um periférico que permite a comunicação directa de computadores, através da linha telefónica.

Embora haja igualmente vários modems, de várias marcas, no mercado, a Atari possui o modelo 850 que aconselhamos por ser especialmente adaptado a estes computadores.

Os modems podem ser utilizados para transferir programas, quer em Basic (ou outra linguagem), quer em código-máquina; enviar ou receber mensagens, tabelas ou qualquer outro tipo de informação. Embora provavelmente não seja muito econó-



mico (atendendo ao custo das chamadas em Portugal), um modem pode mesmo servir para manter uma conversação, via écrans, entre dois utentes de computador.

Em países mais desenvolvidos – como é o caso de Inglaterra, França e, sobretudo, Estados Unidos e Canadá – os modems têm uma ainda maior utilidade já que permitem ligação a umas centenas de companhias produtoras de bancos de dados para computadores, bibliotecas, escolas, etc.

### *Touch Tablet*

Trata-se de um lápis electrónico projectado pela Atari que permite fazer desenho sem necessidade de qualquer programação. Com um lápis especial, pode executar-se o desenho normalmente sobre uma mesa e papel, pois o computador memoriza todos os movimentos podendo reproduzi-los posteriormente no écran ou numa impressora.

Para além do desenho, este periférico permite igualmente pintar sectores do mesmo.

### *Light Pen*

Traduzindo literalmente, caneta de luz, este é um periférico muito comum em todos os computadores. Como o nome indica, trata-se de uma caneta com a qual é possível escrever directamente no écran. Não como a anterior, a partir de uma mesa e papel, mas com a ponta encostada ao próprio écran.

Também este periférico permite pintar áreas de écran, além de outras características que o manual explica.

### *Outros periféricos*

A Atari lançou no mercado uma caixa, a que chamou Expansion System, através da qual é possível ligar ao computador uma série de periféricos, mesmo não fabricados especificamente





































para os seus computadores: sintetizadores de voz, discos rígidos, paddles, etc.

É impossível falar de todos eles, tantos e tão variados são os que se encontram já no mercado. Em alguma da literatura que indicaremos em apêndice, encontrará a descrição detalhada desses periféricos, bem como das suas possibilidades de aplicação.





## APÉNDICE A — CÓDIGOS ATASCII

Código decimal	Caract. ATASCII	TECLAS P/cart.	Código decimal	Caract. ATASCII	TECLAS p/cart.
0		CTRL-,	18		CTRL-R
1		CTRL-A	19		CTRL-S
2		CTRL-B	20		CTRL-T
3		CTRL-C	21		CTRL-U
4		CTRL-D	22		CTRL-V
5		CTRL-E	23		CTRL-W
6		CTRL-F	24		CTRL-X
7		CTRL-G	25		CTRL-Y
8		CTRL-H	26		CTRL-Z
9		CTRL-I	27		ESC ESC
10		CTRL-J	28		ESC CTRL-↑
11		CTRL-K	29		ESC CTRL-↓
12		CTRL-L	30		ESC CTRL-←
13		CTRL-M	31		ESC CTRL-→
14		CTRL-N	32		SPACE BAR
15		CTRL-O	33		SHIFT-1
16		CTRL-P	34		SHIFT-2
17		CTRL-Q	35		SHIFT-3

Código decimal	Caract. ATASCII	TECLAS P/cart.	Código decimal	Caract. ATASCII	TECLAS p/cart.
36	\$	SHIFT-4	55	7	7
37	%	SHIFT-5	56	8	8
38	&	SHIFT-6	57	9	9
39	'	SHIFT-7	58	;	SHIFT-;
40	(	SHIFT-9	59	;	:
41	)	SHIFT-0	60	<	<
42	*	*	61	=	=
43	+	+	62	>	>
44	,	,	63	?	SHIFT-
45	-	-	64	@	SHIFT-8
46	.	.	65	A	A
47	/	/	66	B	B
48	0	0	67	C	C
49	1	1	68	D	D
50	2	2	69	E	E
51	3	3	70	F	F
52	4	4	71	G	G
53	5	5	72	H	H
54	6	6	73	I	I

Código decimal	Caract. ATASCII	TECLAS P/cart.	Código decimal	Caract. ATASCII	TECLAS p/cart.
74	J	J	93	⌋	SHIFT- .
75	K	K	94	^	SHIFT- *
76	L	L	95	...	SHIFT--
77	M	M	96	◆	CTRL-.
78	N	N	97	a	(LOWR) A
79	O	O	98	b	(LOWR) B
80	P	P	99	c	(LOWR) C
81	Q	Q	100	d	(LOWR) D
82	R	R	101	e	(LOWR) E
83	S	S	102	f	(LOWR) F
84	T	T	103	g	(LOWR) G
85	U	U	104	h	(LOWR) H
86	V	V	105	i	(LOWR) I
87	W	W	106	j	(LOWR) J
88	X	X	107	k	(LOWR) K
89	Y	Y	108	l	(LOWR) L
90	Z	Z	109	m	(LOWR) M
91	[	SHIFT- ,	110	n	(LOWR) N
92	\	SHIFT- +	111	o	(LOWR) O

Código decimal	Caract. ATASCII	TECLAS P/cart.	Código decimal	Caract. ATASCII	TECLAS p/cart.
112		(LOWR) P	131		(A) CTRL-C
113		(LOWR) Q	132		(A) CTRL-D
114		(LOWR) R	133		(A) CTRL-E
115		(LOWR) S	134		(A) CTRL-F
116		(LOWR) T	135		(A) CTRL-G
117		(LOWR) U	136		(A) CTRL-H
118		(LOWR) V	137		(A) CTRL-I
119		(LOWR) W	138		(A) CTRL-J
120		(LOWR) X	139		(A) CTRL-K
121		(LOWR) Y	140		(A) CTRL-L
122		(LOWR) Z	141		(A) CTRL-M
123		CTRL-;	142		(A) CTRL-N
124		SHIFT- =	143		(A) CTRL-O
125		ESC\CTRL-< or ESC\SHIFT-<	144		(A) CTRL-P
126		ESC\BACK S	145		(A) CTRL-Q
127		ESC\TAB	146		(A) CTRL-R
128		(A) CTRL-,	147		(A) CTRL-S
129		(A) CTRL-A	148		(A) CTRL-T
130		(A) CTRL-B	149		(A) CTRL-U

Código decimal	Caract. ATASCII	TECLAS P/cart.	Código decimal	Caract. ATASCII	TECLAS p/cart.
150		(A) CTRL-V	169		(A) SHIFT-0
151		(A) CTRL-W	170		(A) *
152		(A) CTRL-X	171		(A) +
153		(A) CTRL-Y	172		(A) ,
154		(A) CTRL-Z	173		(A) -
155	EOI <sup>9</sup>	RETURN	174		(A) .
156		<sup>10</sup> ESC\SHIFT-BACK S	175		(A)
157		<sup>11</sup> ESC\SHIFT->	176		(A) 0
158		<sup>12</sup> ESC\CTRL-TAB	177		(A) 1
159		<sup>13</sup> ESC\SHIFT-TAB	178		(A) 2
160		(A) SPACE BAR	179		(A) 3
161		(A) SHIFT-1	180		(A) 4
162		(A) SHIFT-2	181		(A) 5
163		(A) SHIFT-3	182		(A) 6
164		(A) SHIFT-4	183		(A) 7
165		(A) SHIFT-5	184		(A) 8
166		(A) SHIFT-6	185		(A) 9
167		(A) SHIFT-7	186		(A) SHIFT-;
168		(A) SHIFT-9	187		(A) :

Código decimal	Caract. ATASCII	TECLAS P/cart.	Código decimal	Caract. ATASCII	TECLAS p/cart.
188	<	(⌘) <	207	O	(⌘) O
189	=	(⌘) =	208	P	(⌘) P
190	>	(⌘) >	209	Q	(⌘) Q
191	?	(⌘) SHIFT-	210	R	(⌘) R
192	@	(⌘) SHIFT-4	211	S	(⌘) S
193	A	(⌘) A	212	T	(⌘) T
194	B	(⌘) B	213	U	(⌘) U
195	C	(⌘) C	214	V	(⌘) V
196	D	(⌘) D	215	W	(⌘) W
197	E	(⌘) E	216	X	(⌘) X
198	F	(⌘) F	217	Y	(⌘) Y
199	G	(⌘) G	218	Z	(⌘) Z
200	H	(⌘) H	219	[	(⌘) SHIFT-,
201	I	(⌘) I	220	\	(⌘) SHIFT-+
202	J	(⌘) J	221	]	(⌘) SHIFT-.
203	K	(⌘) K	222	^	(⌘) SHIFT-•
204	L	(⌘) L	223	_	(⌘) SHIFT--
205	M	(⌘) M	224	◆	(⌘) CTRL-.
206	N	(⌘) N	225	a	(⌘) (LOWR) A
			226	b	(⌘) (LOWR) B
			227	c	(⌘) (LOWR) C

Código decimal	Caract. ATASCII	TECLAS P/cart.	Código decimal	Caract. ATASCII	TECLAS p/cart.
228		(A) (LOWR) D	242		(A) (LOWR) R
229		(A) (LOWR) E	243		(A) (LOWR) S
230		(A) (LOWR) F	244		(A) (LOWR) T
231		(A) (LOWR) G	245		(A) (LOWR) U
232		(A) (LOWR) H	246		(A) (LOWR) V
233		(A) (LOWR) I	247		(A) (LOWR) W
234		(A) (LOWR) J	248		(A) (LOWR) X
235		(A) (LOWR) K	249		(A) (LOWR) Y
236		(A) (LOWR) L	250		(A) (LOWR) Z
237		(A) (LOWR) M	251		(A) CTRL-;
238		(A) (LOWR) N	252		(A) SHIFT- =
239		(A) (LOWR) O	253		ESC\CTRL-2 * (A) ESC\ CTRL-BACK S
240		(A) (LOWR) P	254		(A) ESC\CTRL->
241		(A) (LOWR) Q	255		(A) ESC\CTRL->



## APÊNDICE B — TABELAS DE MODOS

### B – 1 Modos gráficos

Modo	Resol.	Cores	Cor	Descrição	Valores	
3	40x20 40x24	4	0	Cor 1	Laranja (2, 8)	
			1	Cor 2, texto luminoso	Verde claro (12, 10)	
			2	Cor 3, cor da janela	Azul escuro (9, 4)	
			3	não usada	Rosa (6, 4)	
			4	Cor 4, fundo/bordo	Preto (0, 0)	
4	80x40 80x48	2	0	Cor 1	Laranja (2, 8)	
			1	Texto luminoso	Verde claro (12, 10)	
			2	Cor da janela	Azul escuro (9, 4)	
			3	não usada	Rosa (4, 6)	
			4	Cor 2, fundo/bordo	Preto (0, 0)	
5	80x40 80x48	4	0	Cor 1	Laranja (2, 8)	
			1	Cor 2, texto luminoso	Verde claro (12, 10)	
			2	Cor 3, cor da janela	Azul escuro (9, 4)	
			3	não usada	Rosa (6, 4)	
			4	Cor 4, fundo/bordo	Preto (0, 0)	
6	160x80 160x96	2	0	Cor 1	Laranja (2, 8)	
			1	Texto luminoso	Verde claro (12, 10)	
			2	Cor da janela	Azul escuro (9, 4)	
			3	não usada	Rosa (4, 6)	
			4	Cor 2, fundo/bordo	Preto (0, 0)	
7	160x80 160x96	4	0	Cor 1	Laranja (2, 8)	
			1	Cor 2, texto luminoso	Verde claro (12, 10)	
			2	Cor 3, cor da janela	Azul escuro (9, 4)	
			3	não usada	Rosa (6, 4)	
			4	Cor 4, fundo/bordo	Preto (0, 0)	
8	320x160 320x192	1	1 lum	0	não usada	Laranja (2, 8)
		1		Cor 1, texto luminoso	Verde claro (12, 10)	
		2		Fundo/janela	Azul escuro (9, 4)	
		3		não usada	Rosa (4, 6)	
		4		Bordo	Preto (0, 0)	
9	80x192	4	16 lum	0	não usada	Laranja (2, 8)
		1		não usada	Verde claro (12, 10)	
		2		não usada	Azul escuro (9, 4)	
		3		não usada	Rosa (6, 4)	
		4		Fundo/bordo+especial	Preto (0, 0)	
10	80x192	9	0	Cor 4	Laranja (2, 8)	
			1	Cor 5	Verde claro (12, 10)	
			2	Cor 6	Azul escuro (9, 4)	
			3	Cor 7	Rosa (6, 4)	
			4	Cor 8	Preto (0, 0)	

11	80x192	16 tons 1 lum	0	não usada	Laranja (2, 8)
			1	não usada	Verde claro (12, 10)
			2	não usada	Azul escuro (9, 4)
			3	não usada	Rosa (6, 4)
			4	Cor fundo/bordo e luminância outras	Preto (0, 0)
14	160x160 160x192	2	0	Cor 1	Laranja (2, 8)
			1	Texto luminoso	Verde claro (12, 10)
			2	Cor da janela	Azul escuro (9, 4)
			3	não usada	Rosa (6, 4)
			4	Cor 2, fundo/bordo	Preto (0, 0)
15	160x160 160x192	4	0	Cor 1	Laranja (2, 8)
			1	Cor 2, texto luminoso	Verde claro (12, 10)
			2	Cor 3, cor da janela	Azul escuro (9, 4)
			3	não usada	Rosa (6, 4)
			4	Cor 4, fundo/bordo	Preto (0, 0)

## B – 2 Modos de texto

Modo	Resol.	Cores	Cor	Descrição	Valores
0	40x24	1 lum	0	não usada	Laranja (2, 8)
			1	Texto luminoso	Verde claro (12, 10)
			2	Texto colorido/fundo	Azul escuro (9, 4)
			3	não usada	Rosa (4, 6)
			4	Bordo	Preto (0, 0)
1	20x20 20x24	4	0	Cor da cor 32-95	Laranja (2, 8)
			1	Cor da cor 0-31 e 96-127, texto lum.	Verde claro (12, 10)
			2	Cor da cor 160-233	Azul escuro (9, 4)
			3	Cor da cor 128-159 e 224-255	Rosa (4, 6)
			4	Fundo e bordo	Preto (0, 0)
2	20x10 20x12	4	0	Cor da cor 32-95	Laranja (2, 8)
			1	Cor da cor, 0-31 e 96-127, texto lum.	Verde claro (12; 10)
			2	Cor da cor 160-233	Azul escuro (9, 4)
			3	Cor da cor 128-159 e 224-255	Rosa (4, 6)
			4	Fundo e bordo	Preto (0, 0)
12	40x20 40x24	4	0	Especial	Laranja (2, 8)
			1	Texto lum. +especial	Verde claro (12, 10)
			2	Cor texto+especial	Azul escuro (9, 4)
			3	Especial	Rosa (4, 6)
			4	Fundo e bordo	Preto (0, 0)
13	40x10 40x12	4	0	Especial	Laranja (2, 8)
			1	Texto lum. +especial	Verde claro (12, 10)
			2	Cor texto+especial	Azul escuro (9, 4)
			3	Especial	Rosa (4, 6)
			4	Fundo e bordo	Preto (0, 0)

## APÊNDICE C — CÓDIGOS DE ERRO

### ERRO N. SIGNIFICADO

- 2 MEMÓRIA INSUFICIENTE – toda a memória disponível pela RAM do computador (16 ou 64K, depende do modelo) foi utilizada.
- 3 VALOR ERRADO – utilizou um número demasiado grande ou demasiado pequeno para determinado comando. Erro comum na utilização dos comandos PLOT e DRAWTO.
- 4 DEMASIADAS VARIÁVEIS – num mesmo programa, apenas pode utilizar um número máximo de 128 variáveis, limite do ATARI.
- 5 ERRO DE TAMANHO DE STRING – foi atribuído a uma cadeia um número maior do que o reservado por um DIM prévio.
- 6 DADOS INSUFICIENTES – a um comando READ foi dado um valor de leitura de dados superior ao número de dados existente no programa.
- 7 NÚMERO MAIOR QUE 32767 – ou um valor indicado não é um inteiro positivo (quando deveria sê-lo) ou é maior que 32676, número máximo admitido pelo computador para determinados comandos.
- 8 ERRO DE INPUT – a resposta do operador foi, por exemplo, uma variável alfanumérica quando o computador esperava uma numérica.
- 9 ERRO DE DIMENSÃO – ou se tentou DIMensionar uma variável com um valor superior a 32767, ou se tentou utilizar uma variável não dimensionada previamente, ou se tentou dar um comando DIM a uma variável já DIMensionada.

- 10 EXCESSO DE SUBROTINAS – existem no programa demasiados comandos GOSUB sem ter ainda surgido nenhum RETURN, ou foi programada uma expressão com demasiados parêntesis.
- 11 NÚMERO MUITO GRANDE – ou o resultado de uma operação foi superior a 10 elevado a 98 (ou 10 elevado a -99), ou se pediu ao computador que dividisse um número por 0 – coisa que o Atari não está preparado para fazer.
- 12 LINHA NÃO ENCONTRADA – não existe no programa uma determinada linha para onde o computador deveria saltar após um comando GOSUB, GOTO ou THEN.
- 13 NEXT sem FOR – no programa, o computador encontrou um comando NEXT sem antes ter surgido o FOR indispensável.
- 14 LINHA DEMASIADO LONGA – ou uma linha contém mais que 114 caracteres (ou três linhas de écran), ou é demasiado complexa para o Basic.
- 15 FALTA DE GOSUB OU FOR – foi encontrado pelo computador um NEXT ou RETURN sem um FOR ou GOSUB anterior.
- 16 RETURN SEM GOSUB – foi encontrado um RETURN sem que anteriormente surgisse o respectivo GOSUB.
- 17 COMANDO INDECIFRÁVEL – por qualquer razão (frequentemente um POKE ou deficiência de memória) um comando numa linha do programa tornou-se indecifrável.
- 18 CHARACTER NÃO VÁLIDO – surge quando o computador tenta converter um caracter em valor numérico através do comando VAL, e tal caracter não é passível de conversão.

- 19 PROGRAMA DEMASIADO LONGO – o erro é indicado quando se tenta carregar em memória (LOAD) um programa que ultrapassa a disponível.
- 20 NÚMERO DE CANAL ERRADO – foi ordenada ao computador a utilização do canal 0 ou de um maior que 7.
- 21 ERRO NA ENTRADA DE DADOS – se um programa foi guardado com um comando CSAVE ou LIST, terá de ser chamado com um comando CLOAD e RETURN. Portanto, se o comando utilizado for LOAD, surge a indicação deste erro.
- 128 BREAK – foi premida a tecla BREAK quando o computador se encontrava numa operação de Input/Output.
- 129 CANAL ABERTO – foi ordenado ao computador que abrisse um canal que se encontrava já aberto.
- 130 PERIFÉRICO INEXISTENTE – tentou utilizar-se um periférico que se não encontrava ligado ao sistema, ou foi esquecido o D inicial indispensável antes do nome do programa a carregar de uma diskette.
- 131 ENTRADA ILEGAL – foi utilizado um comando INPUT ou GET num canal aberto para saída de dados.
- 132 COMANDO ILEGAL – foi utilizado um comando que não pode ser aplicado a determinado periférico.
- 133 PERIFÉRICO OU FICHEIRO NÃO ABERTOS – surge quando se tenta usar um comando PLOT ou DRAWTO num modo texto. Ou quando o computador tenta ler ou escrever através de um carfãl fechado.
- 134 CANAL COM NÚMERO ERRADO – só podem utilizar-se os canais 1, 2, 3, 4, 5, 6 e 7.

- 135 SAÍDA PARA UM CANAL DE ENTRADA – foi utilizado um comando PUT ou PRINT num canal aberto apenas para entrada de dados.
- 136 FIM DE FICHEIRO – o computador chegou ao fim de um ficheiro, durante uma operação de leitura.
- 137 REGISTO DEMASIADO LONGO – o computador tentou ler um registo com mais de 256 bytes.
- 138 FALTA DE RESPOSTA – o computador não conseguiu obter resposta de um periférico que se tinha pedido.
- 139 DISFUNÇÃO DE PERIFÉRICO – um periférico tem problemas de funcionamento ou não pode responder a determinado comando.
- 140 ERRO DE ENVIO DE INFORMAÇÃO – pode haver deficiência de funcionamento da cassette ou diskette.
- 141 CURSOR FORA DE ÁREA – tentou posicionar o cursor num ponto fora da área de modo de texto que está a utilizar. Ocorre quando se utilizam os comandos PLOT, DRAWTO POSITION, LOCATE ou PRINT 6.
- 142 DEFEITO EM PERIFÉRICO – habitualmente trata-se de uma cassette ou diskette corrompidas.
- 143 DEFEITO EM PERIFÉRICO – gravação ou leitura mal realizadas de uma cassette ou diskette.
- 144 GRAVAÇÃO ILEGAL – tentou gravar um programa ou dados numa diskette protegida.
- 145 COMPARAÇÃO ERRADA – ao comparar dados gravados com os existentes nas memórias do computador, verifica-se uma discrepância; ou pode tratar-se igualmente de um erro no écran.

- 146 FUNÇÃO IMPOSSÍVEL – foi solicitado ao computador que escrevesse no teclado, ou lesse da impressora ou qualquer tarefa absurda e impossível deste tipo.
- 147 RAM INSUFICIENTE – para o screen, atendendo ao modo gráfico que tentou utilizar-se.
- 150 PORTO SÉRIE ABERTO – um porto série só pode ser aberto para um canal de cada vez.
- 160 NÚMERO DE DRIVE ERRADO – foi indicado o número de um drive que o computador desconhece.
- 161 DEMASIADOS FICHEIROS ABERTOS – só podem abrir-se 3 ficheiros simultaneamente para cada drive.
- 162 DISCO CHEIO – o espaço disponível na diskette em uso foi inteiramente utilizado.
- 163 ERRO DESCONHECIDO – foi encontrado um erro que o computador não pode identificar.
- 164 ERRO EM NÚMERO DE FICHEIRO – o ponteiro de ficheiro da diskette não se encontra posicionado num ficheiro aberto.
- 165 NOME DE FICHEIRO ERRADO – foi utilizado um nome errado para um ficheiro gravado em diskette.
- 166 ERRO DE SECTOR EM DISKETTE – foi ordenado ao ponteiro da diskette para se mover para um sector inexistente.
- 167 FICHEIRO FECHADO – foi ordenado ao drive que escrevesse, apagasse ou mudasse o nome de um ficheiro não aberto.



- 168 **COMANDO NÃO VÁLIDO** – o computador tentou usar um comando que não existe ou se não encontra definido para determinado periférico.
- 169 **DIRECTORIA CHEIA** – uma directoria apenas pode conter 64 ficheiros.
- 170 **FICHEIRO NÃO ENCONTRADO** – pediu-se ao computador para localizar um ficheiro que não existe na diskette respectiva.
- 171 **LEITURA INVÁLIDA** – pediu-se a leitura de um sector da diskette que não fazia parte do ficheiro aberto.

## APÊNDICE D — TABELA DE SOM

Código	Nota
29	DO
31	RE
33	MI ou RE <i>b</i>
35	MI
37	FA ou MI <i>b</i>
40	FA
42	SOL ou FA <i>b</i>
45	SOL
47	LA
50	SI ou LA <i>b</i>
53	SI
57	DO ou SI <i>b</i>
60	DO
64	RE
68	MI ou RE <i>b</i>
72	MI
76	FA ou MI <i>b</i>
81	FA
85	SOL ou FA <i>b</i>
91	SOL
96	LA
102	SI ou LA <i>b</i>
108	SI
114	DO ou SI <i>b</i>
121	DO
128	RE
136	MI ou RE <i>b</i>
144	MI
153	FA ou MI <i>b</i>
162	FA
173	SOL ou FA <i>b</i>
182	SOL
193	LA
204	SI ou LA <i>b</i>
217	SI
230	DO ou SI <i>b</i>
243	DO

## APÊNDICE E — OUTRAS PUBLICAÇÕES

Inside Atari Basic – Bill Carris (Reston, 1983).

Atari Sound and Graphics – Herb Moore, Judy Lower and Bob Albrecht (Wiley, 1982).

Your First Atari Program – Rodney Zaks (Sybex, 1984).

Your Atari computer – Lon Poole, Martin McNiff and Steven Cook (Osborne/McGraw-Hill, 1982).

Atari Basic Programs in Minutes – Stanley R. Trost (Sybex, 1984).

Basic Exercises for the Atari – J.P.Lamoitier (Sybex, 1984).

The Easy Guide to Your Atari 600XL e 800XL – Thomas Blackadar (Sybex, 1985).

Atari Users Guide – Mark Ellis, Robert Ellis and Larry Goldstein (Brady, 1985).

Basic Programs for the Atari 600XL & 800XL – Timothy Orr Knight and Darren LaBatt (Tab, 1985).

## APÊNDICE F — CÓDIGOS OPERATIVOS PARA O 6502

Embora este livro seja basicamente para utilizadores com pouca experiência em computadores, tivémos a preocupação de, apesar disso, o tornar extensivo a quem tenha conhecimentos mais profundos, sem conhecer, todavia, os computadores Atari.

Para aqueles que trabalham já em programações mais avançadas aqui fica, pois, a tabela de códigos operativos do microprocessador que equipa os computadores da Atari.

Código	Inst.	Código	Inst.	Código	Inst.
00	BRK	30	BMI	65	ADC
01	ORA	31	AND	66	ROR
05	ORA	35	AND	68	PLA
06	ASL	36	ROL	69	ADC
08	PHP	38	SEC	6A	ROR
09	ORA	39	AND	6C	JMP
0A	ASL	3D	AND	6D	ADC
0D	ORA	3E	ROL	6E	ROR
0E	ASL	40	RTI	70	BVS
10	BPL	41	EOR	71	ADC
11	ORA	45	EOR	75	ADC
15	ORA	46	LSR	76	ROR
16	ASL	48	PHA	78	SEI
18	CLC	49	EOR	79	ADC
19	ORA	4A	LSR	7D	ADC
1D	ORA	4C	JMP	7E	ROR
1E	ASL	4D	EOR	81	STA
20	JSR	4E	LSR	84	STY
21	AND	50	BVC	85	STA
24	BIT	51	EOR	86	STX
25	AND	55	EOR	88	DEY
26	ROL	56	LSR	8A	TXA
28	PLP	58	CLI	8C	STY
29	AND	59	EOR	8D	STA
2A	ROL	5D	EOR	8E	STX
2C	BIT	5E	LSR	90	BCC
2D	AND	60	RTS	91	STA

2E	ROL	61	ADC	94	STY
95	STA	B8	CLV	DD	CMP
96	STX	B9	LDA	DE	DEC
98	TYA	BA	TSX	E0	CPX
99	STA	BC	LDY	E1	SBC
9A	TXS	BD	LDA	E4	CPX
9D	STA	BE	LDX	E5	SBC
A0	LDY	C0	CPY	E6	INC
A1	LDA	C1	CMP	E8	INX
A2	LDX	C4	CPY	E9	SBC
A4	LDY	C5	CMP	EA	NOP
A5	LDA	C6	DEC	EC	CPX
A6	LDX	C8	INY	ED	SBC
A8	TAY	C9	CMP	EE	INC
A9	LDA	CA	DEX	F0	BEQ
AA	TAX	CC	CPY	F1	SBC
AC	LDY	CD	CMP	F5	SBC
AD	LDA	CE	DEC	F6	INC
AE	LDX	D0	BNE	F8	SED
B0	BCS	D1	CMP	F9	SBC
B1	LDA	D5	CMP	FD	SBC
B4	LDY	D6	DEC	FE	INC
B5	LDA	D8	CLD		
B6	LDX	D9	CMP		

## TABELA DO SETCOLOR

A: Registo de cor	B: Cor (tom)	C: Luminância
0 – Pincel n. 1	0 – Cinzento	0 – Fosco
1 – Pincel n. 2	1 – Dourado	2 –
2 – Pincel n. 3	2 – Laranja	4 –
3 – Não utiliz.	3 – Vermelho	6 –
4 – Apagador e fundo	4 – Rosa	Médio
	5 – Violeta	8 –
	6 – Roxo	10 –
	7 – Azul	12 –
	8 – Azul claro	14 – Brilhante
	9 – Azul marinho	
	10 – Turquesa	
	11 – Verde mar	
	12 – Verde	
	13 – Amarelo claro	
	14 – Amarelo torrado	
	15 – Laranja claro	

## TABELA PARA COLOR E SETCOLOR EM MODO 4 CORES

Registo Setcolor	Número Pincel	Cor	Código Setcolor
0	Cor 1	Laranja	0,2,8
1	Cor 2	Verde	1,12,10
2	Cor 3	Azul	2,9,4
3	–	Rosa ou Verm.	3,4,6
4	Cor 0	Preto	4,0,0

**TABELA DE REGISTOS DO SETCOLOR  
EM MODO TEXTO**

Registo Setcolor	Região do écran	Cor	Código Setcolor
0	–	Laranja	0,2,8
1	Luminância de texto	Verde	1,12,10
2	Fundo	Azul	2,9,4
3	–	Rosa ou verm.	3,4,6
4	Bordo	Preto	4,0,0

**TABELA DOS 16 MODOS GRÁFICOS**

Modo	Tipo	Resolução	N. cores	Bytes
0	Texto normal	40 × 24	1	992
1	Texto aumentado	20 × 20	5	674
2	Texto aumentado	20 × 10	5	424
3	Gráf. 4 cores	40 × 20	4	434
4	Gráf. 2 cores	80 × 40	2	694
5	Gráf. 4 cores	80 × 40	4	1174
6	Gráf. 2 cores	160 × 80	2	2174
7	Gráf. 4 cores	160 × 80	4	4190
8	Gráf. alta resol.	320 × 160	1	8112
9	Gráf. GTIA	80 × 192	1	8138
10	Gráf. GTIA	80 × 192	9	8138
11	Gráf. GTIA	80 × 192	16	8138
12	Texto 4 cores	40 × 20	5	1154
13	Texto 4 cores	40 × 10	5	664
14	Gráf. 2 cores	160 × 160	2	4270
15	Gráf. 4 cores	160 × 160	4	8122

## CONTROLO DE CORES NOS MODOS TEXTO 1 E 2

Registo Setcolor	Letras controladas	Cor	Código Setcolor
0	Maiúsculas	Laranja	0,2,8
1	Minúsculas	Verde	1,12,10
2	Maiúsc. invert.	Azul	2,9,4
3	Minúsc. invert.	Rosa ou Verm.	3,4,6
4	Fundo	Preto	4,0,0

## REGISTOS DO SETCOLOR NOS MODOS 12 E 13

Registo Setcolor	Parte control. das Letras	Cor	Código Setcolor
0	Lado esquerdo	Laranja	0,2,8
1	Meio	Verde	1,12,10
2	Lado direito	Azul	2,9,4
3	Lado dir. inver.	Rosa ou Verm.	3,4,6
4	Fundo	Preto	4,0,0





## ÍNDICE

INTRODUÇÃO .....	7
1 – OS COMPUTADORES ATARI .....	9
1.1. Modelos e características .....	9
1.2. O teclado .....	12
1.3. Os novíssimos modelos .....	16
2 – A LINGUAGEM DO ATARI .....	18
2.1. As linguagens dos computadores .....	18
2.2. O Basic da Atari .....	19
2.3. O uso de alguns comandos .....	23
2.4. A programação .....	30
3 – GRÁFICOS E TEXTO .....	33
3.1. Écrans .....	33
3.2. Os modos e as cores .....	35
3.3. Desenhar, pintar .....	42
4 – O SOM .....	48
4.1. Os parâmetros .....	48
4.2. Uma tabela média .....	49
4.3. Programas .....	50
5 – ARMAZENAMENTO DE DADOS .....	52
5.1. O gravador de cassetes .....	53
5.2. O Disk Drive .....	58
5.3. O cartão magnético .....	62
6 – O SOFTWARE .....	64
6.1. Escrever um programa .....	65

6.2. Testar e alterar .....	72
6.3. 4 programas simples .....	75
6.4. Programas comerciais .....	82
7 - EXPANSÃO DO SISTEMA .....	85
APÊNDICES:	
A - Códigos ATASCII .....	91
B - Tabelas de modos .....	98
C - Códigos de erro .....	101
D - Tabela de som .....	107
E - Outras publicações .....	108
F - Códigos operativos para o 6502 .....	109
TABELAS .....	111

Luís de Campos, o autor deste livro, já com várias obras sobre informática publicadas pela Presença, traz agora ao leitor interessado neste campo, o *Computador Atari*. Este, segundo o autor, terá tendência, num curto prazo de tempo, a competir com outras marcas de micros existentes no mercado. Neste manual, são descritos pormenorizadamente os diferentes modelos, suas características, o teclado, a linguagem Basic do Atari, o uso dos comandos, a programação, gráficos, texto, o som, o armazenamento de dados e a expansão do sistema. São dados em Apêndice os códigos ATASCII, Tabelas de Modos Gráfico e de Texto, Código de Erro, Tabela de Som, códigos operativos para o 6502, Tabela para color e setcolor e Bibliografia. Em Portugal, foi este ano lançado o modelo 800XL, prevendo-se que o Atari venha a alcançar um lugar de realce no nosso mercado, com o aparecimento dos novos modelos lançados recentemente no estrangeiro – o 130XE e o 520ST – e dos modelos 1400XL e 1450XL cujo lançamento se prevê para breve. Este livro destina-se a todos os modelos Atari, mas é dedicado, essencialmente, à série XL.



EDITORIAL PRESENÇA