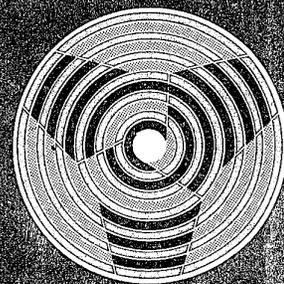


# PALESTRAS DO VII CNPD



**GUANABARA 1974**

004.06  
C749  
1974

**EDITORES DE TEXTO**  
**UMA EXPERIÊNCIA EM APL**

**Daniel Schwabe**  
**Miklos Antal Vasarhelyi**

**Rio Datacentro**  
**PUC/RJ.**

## Editores de Texto Uma Experiência em APL

Daniel Schwabe  
Miklos Antal Vasarhelyi  
Rio Datacentro, PUC/RJ

- **Palavras Chave:** Editores de Texto,  
Processamento da Palavra,  
Formatação,  
Composição Gráfica,  
Processamento de "Strings",  
APL.
- Um editor de texto é um sistema composto de uma série de programas que cria, compõe e edita um texto que contém símbolos especiais para orientar a composição.
- Este sistema pode ser usado para a composição de cartas, artigos, relatórios, quadros, tabelas ou qualquer aplicação que envolva a composição de um dado texto.
- Correções sucessivas, inserções, personalização de relatórios sucessivos são efetuados por um editor de texto sem a necessidade das preparações sucessivas do texto por inteiro.
- O presente trabalho descreve um sistema interativo de edição de textos programados em APL. O APL é um sistema conversacional com um pequeno tempo de resposta, grande flexibilidade para a manipulação de caracteres e, devido ao fato de ser um interpretador (e não um compilador), permite um controle muito grande, ao nível de instrução, facilitando a implantação. Por outro lado, apresenta desvantagens tais como o alto custo, baixa velocidade de impressão e limitações quanto ao formato de saída.
- Descrevemos inicialmente as características principais de editores de texto; a seguir comentaremos o presente editor de texto, apresentando alguns problemas encontrados.

## I. • O Problema

Um editor de texto é um sistema composto de uma série de programas que cria, compõe e edita um texto que contém símbolos especiais para orientar a composição.

Este sistema pode ser usado para a composição de cartas, artigos, relatórios, quadros, tabelas ou qualquer aplicação que envolva a composição de um dado texto.

Correções sucessivas, inserções, personalização de relatórios sucessivos são efetuados por um editor de texto sem a necessidade da preparação sucessiva do texto por inteiro.

O presente trabalho descreve um sistema interativo de edição de textos programados em APL. O APL é um sistema conversacional com um pequeno tempo de resposta, grande flexibilidade para a manipulação de caracteres e, devido ao fato de ser um interpretador (e não um compilador), permite um controle muito grande, ao nível de instrução, facilitando a implantação. Por outro lado, apresenta desvantagens tais como o alto custo, baixa velocidade de impressão e limitações quanto ao formato de saída.

Descrevemos, inicialmente, as características principais de editores de texto; a seguir comentaremos o presente editor de texto, apresentando alguns problemas encontrados.

## II. • Características dos Editores de Texto

Um editor de textos tem, basicamente, 4 módulos semi-independentes:

- a. o de entrada, que permite a criação de um novo texto;
- b. o de composição, que formata o texto "bruto" segundo as especificações do usuário;
- c. o de correção, que permite ao usuário corrigir o texto "bruto";
- d. o de saída, que imprime o texto editado.

O sistema possui uma série de parâmetros globais que definem o estado do mesmo a qualquer instante, e que orientam o seu funcionamento. Eles são do tipo tamanho de página; identificação do caráter de controle, controle de página, etc. ... Estes parâmetros variam grandemente de sistema para sistema.

A composição propriamente dita é orientada por meio de caracteres especiais (símbolos) colocados no texto chamado "bruto", mas que não aparecem no texto editado.

O texto pode ser armazenado na sua forma "bruta", todo condensado, na forma editada ou em

ambas as formas. Com o módulo de correção, podem ser feitas correções no texto bruto; estas correções podem ser tanto relativas a forma final do texto quanto a sua redação (ou ortografia).

Tem-se ainda a possibilidade de inserir texto "variável" num texto fixo básico, por exemplo, dada uma circular, pode-se mudar o destinatário de cada cópia.

A potência de um editor de textos reside nos seus operadores primitivos, tais como acerto de margens, controle de parágrafo, tabulação, etc. ... Uma combinação eficiente dos operadores primitivos consegue qualquer resultado que se deseje.

Diversos tipos de manipulação de entrada de texto podem ser encontrados entre editores de texto. Os dois modos principais são entrada via *batch* e entrada via *time-sharing*. A entrada via *batch* tem a vantagem de mais baixo custo e dependência exclusiva da manipulação física dos cartões a serem editados. No evento de correção de cartões (ou imagem de cartões, em se usando terminais), é suficiente a substituição manual após a sua correção e/ou duplicação.

No modo *time-sharing*, a maioria das manipulações devem ser feitas através de programas dentro do sistema, que acham o local a ser corrigido e permitem correções no texto iterativamente. O custo destes sistemas deve ser maior devido a maior utilização de armazenagem e iteração (CPU) por parte do computador.

Um editor de textos deve ser medido não só pela sua eficiência em aproveitamento de máquina porém especialmente através de medidas agregadas de custo do sistema homem-máquina. Isto implica em tempo de máquina, tempo de conexão, utilização de recursos físicos como papel, cartão, etc., e também medidas de custo de utilização de recursos humanos, que podem vir a ser uma fração considerável do total.

Aplicações diversas se mostrarão mais adequadas a sistemas de edição de textos diversos.

## III. • Um Editor de Textos em APL

A estrutura do editor de textos em APL (APLET) é composta de quatro módulos principais:

- Módulo de entrada
- Módulo de composição
- Módulo de correção
- Módulo de impressão

A figura 1 descreve as características gerais do sistema em questão. (vide figura 1)

O estado do sistema é definido pelas seguintes variáveis:

- Largura da linha
  - Tamanho da página
  - Tamanho do parágrafo
  - Valor das margens
  - Controle de “canal” (salto automático ou parada para um *return*)
  - Disposição do tabulador
  - Valor do caracter de edição
  - Valor do caracter a ser inserido na tabulação
- O módulo de entrada é constituído pelo programa CRIA que cria um texto novo e pelo programa ADICIONA que adiciona texto a um texto já existente.

O programa cria da oportunidade ao usuário de:

- a) reentrar um pedaço do texto a partir do qual ele não tenha gostado
- b) ter um *display* do texto já editado, a partir de um certo ponto especificado

O módulo de correção é constituído pelos programas MOD e TRECHO. O programa MOD (que usa TRECHO) pede em que trecho do texto será efetuada a correção, e este é exibido; neste ponto, o usuário entre com as correções, e estas são efetuadas.

O módulo de saída é constituído pelo programa SAÍDA. Este programa permite controle de “canal” e numeração das linhas, quando da impressão do texto editado.

O módulo de composição é formado pelos programas EDIT, JU1, JD1, ....., OW. Suas ações são as seguintes:

- TA — Cria margem um nível secundário de margem.
- JE1 — Acerta a margem esquerda, mas deixa a margem direita irregular.
- JD1 — Acerta a margem direita, mas deixa a margem esquerda irregular
- CE1 — Centraliza o texto (na página)
- MO1 — Acerta as 2 margens

Chamados por EDIT1 temos ainda:

- Causa o espaçamento de linhas
- Altera o caracter de inserção na tabulação.
- Permite a inserção de texto variável. Este programa é controlado por uma chave, que pode assumir os valores 0,1 e 2.
  - (a) o computador emite mensagem pedindo o texto a ser inserido
  - (b) o computador simplesmente espera a entrada do texto a ser inserido sem mandar mensagem
  - (c) o texto a ser inserido é tirado de uma lista, que é rodada de 1 posição a cada referência.
- Sublinha a palavra seguinte

- PA — Coloca um parágrafo
- MD — Altera o valor da margem direita
- ME — Altera o valor da margem esquerda
- VA — Altera o valor do tabulador. Isto pode ser feito de 2 maneiras:
  - (a) um vetor numérico é fornecido, indicando as posições do tabulador
  - (b) uma imagem da linha é fornecida, com barras nas posições desejadas.
- TB — Causa um salto do tabulador ou para uma posição determinada ou para a posição mais próxima.
- OW — Permite a criação de um ou mais códigos próprios do usuário, como combinação qualquer dos códigos válidos. A combinação a ser inserida é tirada de uma lista criada pelo usuário (através do programa LISTA), e este especifica qual a combinação desejada

#### IV. • Razões para as Opções Básicas

1. **Forma do símbolo de edição** — O símbolo de edição pode ter 2 formas básicas, a saber:

- a) Um caracter especial, seguido de um ou mais caracteres representando o código (qual a operação), por exemplo:

caracter especial — E — código — específico do comando

- b) Apenas um caracter, reunindo as duas informações, por exemplo:

O

Foi escolhida a primeira alternativa, pois esta oferece as seguintes vantagens:

- Permite a determinação, por parte do usuário, de qual será o caracter especial, permitindo maior flexibilidade quanto ao tipo de texto impresso.
- Permite também, a especificação de que caracteres que representarão que operações, sendo necessário apenas a preservação da ordem com que estes caracteres são especificados.

2. **Estado geral do sistema** — Certos sistemas assumem, como *default*, que uma dada operação (especificada previamente pelo usuário) será efetuada em qualquer texto, salvo instruções em contrário dentro do próprio texto.

Uma outra opção, que foi a escolhida, obriga ao usuário, no início de cada texto, a especificar qual a operação a ser efetuada. Cabe aqui chamar a atenção quanto à hierarquia de operadores.

Na mais alta ordem estão os operadores que

alteram a forma geral do texto, isto é, separa-o em linhas respeitando uma dada forma, tal como alinhamento à direita, à esquerda, centralização, etc. ....

Na mais baixa ordem estão os que simplesmente alteram dinamicamente o estado do sistema, ou que inserem caracteres (não brancos) no texto, tal como tabulação, margens (1º caso) e texto variável, pular linha (2º caso).

Sendo assim, todo texto bruto deve, necessariamente, começar por um operador de mais alta hierarquia.

**3. Escopo de um operador** – aqui também temos 2 alternativas, a saber:

- a) Cada operador (de mais alta hierarquia) atua sobre o texto imediatamente precedente (até outro de igual hierarquia) ou
- b) Atua sobre o texto imediatamente seguinte (até outro de igual hierarquia).

A diferença principal entre estas alternativas é que, dada uma modificação, no 1º caso, ela é válida apenas para a linha (ou trecho) imediatamente seguinte à sua ocorrência, e no 2º caso, ela é válida para o próprio trecho em que ela ocorre.

Exemplificando, se quisermos uma mudança no valor da margem esquerda, ela, no 1º caso, só será efetivada a partir da próxima linha, ao passo que no 2º caso, a própria linha em que a mudança ocorre já trará esta modificação incorporada.

Em consequência, foi escolhida a segunda alternativa, pois ela engloba, parcialmente, a anterior, e permite uma maior flexibilidade. Desta forma, determinado o trecho sobre qual o operador (de maior hierarquia) vai agir, todas as modificações de menor hierarquia são realizadas antes que este operador atue sobre o texto.

**4. Seleção dos operadores de mais alta hierarquia** – Aqui temos, basicamente, 4 operadores universalmente aceitos, que são:

- Alinhar pela margem esquerda
- Alinhar pela margem direita
- Acertar as 2 margens
- Centralizar o texto na página

Devido à organização foi acrescentado mais um operador, que efetua qualquer uma das operações acima mencionadas, mas respeitando também um nível secundário de margem.

**5. Seleção de operadores de mais baixa hierarquia** – Existem aqui uma infinidade de opções, muitas, comuns a todos os editores de texto.

O APLET possui, no entanto, um operador que lhe é único, que é o que permite a criação de um novo operador (de maior ou menor hierarquia)

como combinação qualquer dos operadores primitivos.

Isto lhe confere uma enorme flexibilidade, uma vez que cada usuário pode construir um (ou mais) operadores que lhe são necessários. Isto será exemplificado mais adiante.

**6. Armazenamento e impressão do texto** – Dado um texto “bruto” (e que está na sua forma mais condensada), o usuário pode desejar guardar, também, o texto editado. Esta opção é oferecida neste editor. Além disso, ele pode desejar uma “prova” do texto editado, para verificação, e, caso este lhe agrade, guardá-lo, ou se não, modificá-lo e editá-lo novamente. Também esta opção é dada ao usuário.

**7. Descrição dos códigos:**

– Mais alta hierarquia:

- a) *nE* Causa o alinhamento de todo o texto, até o próximo operador, segundo a margem esquerda.
- b) *nD* Causa o alinhamento de todo o texto, até o próximo operador, segundo a margem direita.
- c) *nH* Causa o alinhamento de todo o texto, até o próximo operador, segundo as duas margens.
- d) *nC* Causa a centralização (na linha) de todo o texto, até o próximo operador.
- e) *nXXA* Causa o alinhamento de todo o texto a seguir, até o próximo operador, da seguinte forma:
  - Se *XX* é numérico, a primeira linha é operada (segundo o operador *A*), com o valor atual da margem, e as restantes terão a margem valendo *XX*.
  - Se *XX* não é numérico, o efeito é o mesmo, apenas que a margem esquerda passará a valer a posição na linha em que se encontra o caracter “*V*”, sublinhado. Note-se que esta última opção permite uma modificação dinâmica da margem.

– Operadores de mais baixa hierarquia:

- a) *nLXX* Faz com que *XX* linhas em branco sejam inseridas no texto
- b) *nFC* Faz o caracter de inserção, na tabulação, passar a valer “*C*”.
- c) *nQ* Modifica o tabulador, da seguinte forma:
  - Se o operando (trecho a seguir) contém números, a disposição do

tabulador será estes números  
 – Se o operando contiver alguma “/”, ele será uma “imagem” da linha, com as barras indicando as posições do tabulador.  
 – Se ele for vazio, o tabulador passa para o valor *default*.

- d) *nS* Faz com que a palavra seguinte seja sublinhada.
- e) *nP* Termina a linha e começa um parágrafo
- f) *n←XX* Atribui o valor XX a margem esquerda
- g) *n←XX* Atribui o valor XX a margem direita
- h) *n!* Faz com que seja pulada uma página (ou para a impressão, à espera de um *return*).
- i) *nV* Causa a inserção de um texto variável neste ponto, conforme já explicado.
- j) *nTXX* Faz a tabulação propriamente dita; se XX não é igual a zero, ele vai para a posição XX do tabulador; se XX é igual a zero, então ele vai para a posição do tabulador mais próxima do ponto em que ele se encontra.
- k) *nOXX* Insere uma seqüência qualquer de caracteres (em geral de controle), tiradas da posição XX da lista de especificação.

## V. • Conclusões

O APLET é um sistema poderoso, ainda que muitas de suas limitações o tornem aplicável somente a um pequeno número de problemas.

Muitas das limitações hoje existentes no APL que inevitavelmente se refletiram neste sistema poderão ser sanadas com a utilização do APL-SV ou do APL-Plus File System. Isto permitirá a manipulação de longos textos, sem particionamento e

armazenagem.

Um sistema como o APLET desenvolvido com objetivos meramente acadêmicos não se dispõe a ser mais que um exercício acadêmico. Poderia ser considerado como o primeiro para o desenvolvimento de um editor de textos em *time-sharing*, com características específicas da língua portuguesa. O problema dos acentos ainda não foi resolvido, assim como o de sublinhar-se um texto com mais de 43 posições. Questões como esta seriam de moderadamente fácil solução num esforço comercial.

A experiência aqui descrita também indica a existência de uma problemática tipicamente brasileira em matéria de editores de texto, o que gera a necessidade de desenvolvimento completo de um sistema especializado para as idiossincrasias do léxico brasileiro.

A composição do presente artigo foi toda ela feita através do sistema aqui descrito, utilizando-se o sistema APL/360 da PUC/RJ.

## — BIBLIOGRAFIA

1. GILMAN, L., ROSE, A.J., APL/360 – An Interactive Approach, Wiley, 1970.
2. IBM – “An APL Text Editor and Composer” – Manual SH 20-1089-0.
3. IBM – “APL/360-OS & DOS User’s Manual” – Manual SH 0906-0.
4. KNUTH, D.E., – The Art of Computer Programming – Vol. 1 – Fundamental Algorithms.
5. MOCK, T.J., VASARHELYI, M.A., APL For Management, Wiley, Becker & Hayes, 1972.
6. SILVER, S.S. – “FMS – Format Manipulation System”, Department of Library Science, UCLA, 1970.
7. PAKIN, S. – APL/360 Reference Manual, SRA, 1972. ■

Fig. 1 – A Estrutura do APLET

