

# O COMPUTADOR A PAPEL: UMA ATIVIDADE PEDAGÓGICA PARA INTRODUÇÃO DE CONCEITOS BÁSICOS DE COMPUTADORES

**Valdemar W.Setzer**

Dept. de Ciência da Computação, Universidade de São Paulo, Brasil

[www.ime.usp.br/~vwsetzer](http://www.ime.usp.br/~vwsetzer)

Esta é uma tradução ampliada do original em inglês, disponível no *site* acima –  
versão 2.0 de 18/3/24

## 1. Introdução

Uma das metas básicas da educação é prover uma *compreensão* geral do mundo, em particular das máquinas. Sem essa compreensão, não é possível colocá-las em seu contexto adequado, usando-as somente quando trazem benefícios e não prejuízos para o indivíduo, para a sociedade e para o meio ambiente. Além disso, uma falta de compreensão dos princípios básicos que regulam o funcionamento das máquinas produz o que costumo chamar de “uma paralisia mental”: a falta de curiosidade e de investigação que são tão típicas dos seres humanos (que estão em constante busca de conhecimento consciente e compreensão). Essa falta diminui nossa condição humana. Quantas pessoas conhecem o princípio dos motores a combustão, da sustentação dos aviões ao voarem, etc.? As máquinas tornaram-se tão complexas e seu interior tão escondido dos olhos e do tato – principalmente quando estão dentro de circuitos eletrônicos integrados, os *chip* – que as pessoas desacostumaram-se de perguntar a questão fundamental: “Como funciona esta máquina?”

Para o público em geral, o computador representa o máximo em termos de máquinas. Portanto, considero absolutamente essencial que escolas deem um ensino que leve à compreensão dos princípios de seu funcionamento. Em minha opinião, computadores só devem ser usados por crianças após a puberdade (1). Em (2) estendi aquelas ideias e propus um currículo para o ensino médio que se inicia com atividades de laboratório, introduzindo noções básicas de circuitos digitais usando relés (que são de fácil compreensão), em seguida transistores (cujas propriedades básicas semelhantes aos relés podem ser introduzidas por meio de experiências muito simples); ver também (3). Somente depois disso são ensinadas noções de linguagens de programação e de algoritmos, seguidos de *software* de uso geral – especialmente tópicos avançados – e a internet. Muitas escolas Waldorf (4, 5) no mundo usam um enfoque similar, reconhecendo a importância de se ensinar o funcionamento básico dos computadores e das linguagens de programação antes de entrar no seu uso por meio de editores de texto, planilhas eletrônicas, bancos de dados, computação gráfica e a internet. Mas há uma lacuna em seu enfoque: não há transição entre as aulas ou laboratórios sobre circuitos digitais e aprender o que são as linguagens de programação ou *software* de aplicação geral.

Proponho aqui atividades práticas que preenchem precisamente essa lacuna. O que está faltando é o ensino de uma *Linguagem de Máquina*, o código básico interpretado por qualquer computador (cada tipo de computador tem a sua própria linguagem de máquina), como o computador executa essa interpretação e quais são os elementos básicos da máquina. É por meio de uma linguagem de máquina que o aluno pode aprender conceitos fundamentais como o de programa armazenado, endereço, a diferença entre instruções e dados, desvios incondicionais e condicionais, registradores, unidade central de processamento, unidade aritmética, unidades de entrada e saída, etc.

Tenho uma grande experiência com esse enfoque. Na Universidade de São Paulo, desde o começo dos anos 60 a computação era ensinada aos alunos da Escola Politécnica (inicialmente, no começo de uma disciplina de 2 semestres de Cálculo Numérico, posteriormente como um semestre de introdução à computação) começando com linguagem de máquina, e depois entrando em uma linguagem de programação de "alto nível" (uso aspas pois essa nomenclatura não é correta: essas linguagens ainda são por demais aderentes à estrutura da máquina). Naquela época usava-se FORTRAN, depois ALGOL no fim dos anos 60, e depois Pascal, C e, ultimamente, Python – ver uma folha de consulta que elaborei para essa última linguagem (6). Até 1968, o computador usado para esse ensino era um IBM-1620, que tinha uma linguagem de máquina decimal relativamente simples. Depois, a introdução na USP do minicomputador IBM-1130 e de um computador de médio porte (para a época), um Burroughs 3500, com linguagens de máquinas binária e muito complexa, respectivamente, fez com que o uso de uma máquina real fosse inviável para aquele ensino. Assim, projetei um computador hipotético, que denominei de HIPO, que foi simulado naquelas máquinas e nas subseqüentes. O HIPO era inicialmente um computador virtual com uma palavra fixa de dez algarismos decimais, usado no começo de qualquer disciplina introdutória de programação. As duas primeiras tarefas feitas pelos estudantes eram precisamente um programa em linguagem de máquina do HIPO e depois o mesmo codificado na linguagem de montagem do HIPO. Além disso, eu empregava o HIPO como linguagem-objeto em meu curso de construção de compiladores (7).

Em 1976 tive a ideia de introduzir uma atividade visual para ensinar as noções básicas da estrutura interna de um computador por meio de sua linguagem de máquina. Inventei o "Computador a Papel", que é uma espécie de teatrinho onde os atores são os estudantes, simulando cada unidade de uma máquina HIPO simplificada. Contrariamente a esta última, as instruções não eram codificadas numericamente, mas escritas em forma de texto (por exemplo, "Adicione o conteúdo da posição de memória 45 ao conteúdo do acumulador e deixe o resultado no acumulador"), para uma compreensão fácil e imediata. Essa atividade, que toma cerca de 2 horas, usa 21 estudantes, que atuam

na frente da classe, de modo que os outros estudantes podem seguir o que está acontecendo e dar sugestões.

A impressão que os estudantes obtêm com essa atividade é intensa, ficando guardada em sua lembrança consciente por muito tempo. De fato, quando eu dava essas disciplinas de introdução, eu costumava referir-me com certa frequência às imagens que eles haviam visto durante o teatrinho do Computador a Papel, quando algum conceito podia ser melhor entendido em termos de linguagem de máquina. Por exemplo, a noção de uma "variável" em linguagem de "alto nível" (correspondendo a uma palavra da unidade central de armazenamento do HIPO), ou a sequência de comandos necessária para trocar os valores de duas "variáveis", somente podem ser bem explicados e compreendidos em termos de uma linguagem de máquina.

Depois do teatrinho do Computador a Papel, uma aula era dedicada à introdução do HIPO simplificado. As instruções em forma de texto do primeiro são codificadas nos códigos e formato das instruções decimais do HIPO, no qual cada "palavra" da unidade central de armazenamento contém um sinal (+ ou -) e 4 dígitos. Aí os estudantes iam a um laboratório para testar e fazer pequenas modificações de programas já existentes, usando um simulador do HIPO em um PC.

Descrevo aqui o teatrinho do Computador a Papel. Ver também em meu *site* o artigo sobre o HIPO (8), e um vínculo para o simulador do mesmo que pode ser carregado em qualquer navegador (9, 10).

## **2. Montagem do Computador a Papel**

O Computador a Papel é "montado" com 21 estudantes, que são chamados à frente da classe. Cada um recebe um pedaço de cartolina com mais ou menos 20x10 cm, com um barbante fixado a ele, de modo que pode ser pendurado no pescoço de cada estudante, mais ou menos à altura do peito. 15 estudantes atuam no papel de *posições de memória* (note-se que na seção anterior usei a expressão "unidade central de armazenamento", pois "memória" é uma antropomorfização indevida: não sabemos como a memória humana funciona). Eles recebem cartolinas onde estão escritos os *endereços* 01, 02, etc. até 12, mais 30, 40 e 45. Além disso, um estudante recebe um pedaço de cartolina semelhante onde está escrita a palavra *UCP* ("unidade central de processamento", de *Central Processing Unit*); seria muito útil se o instrutor observasse os estudantes quando eles entram na classe se essa for a primeira aula, ou em aulas anteriores, de modo a atribuir o papel de UCP para o estudante aparentemente mais extrovertido. Esse estudante recebe um pedaço de giz e um apagador (pode ser um pano). Um outro estudante (uma garota, se a CPU for um rapaz, ou vice-versa), recebe um pedaço de cartolina onde está escrito *ACUMULADOR*. Esse estudante deve ter um barbante preso ao redor de sua cintura, e a outra extremidade deve ser presa no estudante fazendo o papel de UCP, de tal modo que haja uma distância de cerca

de 1,5 m entre eles. A ideia é que o acumulador faz parte da UCP, de modo que onde esta vai, o acumulador deve ir atrás, o que costuma criar certas situações engraçadas (piadinha para a ocasião: "onde vai a vaca, o boi vai atrás"). Outros três estudantes recebem uma cartolina semelhante, onde estão escritas as seguintes palavras: *APONTADOR DE INSTRUÇÕES*, *UNIDADE DE ENTRADA* e *UNIDADE DE SAÍDA*. Outra possibilidade para os dois últimos é *TECLADO* e *IMPRESSORA*. Os dois últimos estudantes sentam em carteiras em frente à classe; todos os outros ficam em pé, de frente para a classe, com os que atuam como posições de "memória" um ao lado do outro, em ordem ascendente de endereços, da esquerda para a direita quando se os olha de frente. Os estudantes que fazem o papel das posições 40 e 45, bem como o do acumulador, recebem um pedaço de cartolina em branco, de mais ou menos 25x15 cm. Eles devem segurar essa cartolina adicional com suas mãos, logo abaixo da cartolina contendo o endereço (ou a palavra ACUMULADOR) que está pendurada em seu pescoço. O ideal é que essas três cartolinas adicionais sejam de cor escura (por exemplo, azul escuro, cinza escuro ou preto), de modo que seja possível ao estudante que faz o papel de UCP escrever nelas com giz branco (outra possibilidade seria usar cartolinas brancas e escrever nelas com giz colorido escuro). É interessante desenhar previamente nessas três cartolinas, com um marcador, 5 quadrados em linha, onde um sinal (+ ou -) e 4 dígitos decimais serão escritos com giz posteriormente pelo estudante da UCP. Um outro estudante deve fazer o papel de USUÁRIO, atuando como uma pessoa normal, de modo que não deve ter nenhuma cartolina indicando seu papel. Ele iniciará o funcionamento do computador ("inicializar", usada comumente para isso, é um anglicismo) e entrará números na "unidade de entrada" ou "teclado". Para isso, seria interessante ter um teclado de computador à disposição, onde o "usuário" digitará números como descrito na seção 4.

### **3. Carga do programa**

O instrutor diz aos estudantes que o computador está pronto, e um *programa* deve ser "milagrosamente" *carregado* nele. Aí ele dá aos estudantes fazendo os papéis das posições de "memória" 01 a 15 pedaços de cartolinas onde estão escritas as seguintes instruções ("Acc" é a abreviatura de "Acumulador", do inglês *accumulator*), [Acc] de "o conteúdo do Acumulador" e [n] de "o conteúdo da posição de memória n):

<b>Posição de memória</b>	<b>Instrução</b>
01	Carregue o Acc com [30]
02	Armazene [Acc] em 40
03	Entre um número e armazene-o em 45
04	Exiba numericamente [45]
05	Carregue o Acc com [45]
06	Se [Acc]<0 desvie para 11
07	Carregue o Acc com [40]
08	Adicione ao Acc [45]
09	Armazene [Acc] em 40
10	Desvie incondicionalmente para 03
11	Exiba numericamente [40]
12	Pare

Além disso, o estudante fazendo papel da posição 30 recebe uma cartolina (semelhante à dada ao acumulador) onde está escrito o número +0000. Isso é usado como o valor inicial do resultado da somatória (posição 40) executada pelo programa – mas o instrutor não deveria contar aos estudantes esse significado, ou o que o programa faz (somar uma sequência de números inteiros positivos ou nulos da entrada, até que apareça nesta última um número negativo, e aí exibir o resultado). Também recomendo que nenhuma explicação seja dada nesse momento, por exemplo como ler a notação [30].

#### **4. Execução do programa**

O instrutor conta que o computador está pronto para começar a funcionar. Ele diz ao (à) estudante da UCP que deve iniciar a execução quando ele (ela) tem sua orelha delicadamente torcida pelo USUÁRIO “ligando” a “máquina”.

A primeira coisa que a UCP deve fazer é mover o APONTADOR DE INSTRUÇÕES para a frente do estudante da posição de “memória” 01, e erguer seu braço e dedo indicador, de modo que ele aponte para o estudante da posição 01. Idealmente, o apontador de instruções deve

ficar sempre apontando para alguma posição de "memória" (o instrutor deve prestar atenção nisso).

Aí o instrutor diz ao estudante da UCP para ler em voz alta a instrução sendo segurada pelo estudante que faz o papel da posição de memória apontada pelo apontador de instruções. Nesse ponto o instrutor diz para o estudante da UCP e para o resto da classe como as instruções devem ser lidas, isto é, o significado de [30] e de [Acc], bem como um endereço como 40: "Carregue no acumulador o conteúdo da posição de 'memória' 30", "armazene o conteúdo do acumulador na posição de 'memória' 40", etc. É interessante colocar a interpretação dada na seção (3) para Acc, [Acc] e [n] no quadro-negro.

O instrutor conta à UCP e à classe que, para ela executar um programa, as seguintes regras devem ser aplicadas em sequência:

- 1) Leia a instrução sendo segurada pelo estudante que está apontado pelo apontador de instruções;
- 2) Mova o apontador de instruções para a próxima posição de "memória";
- 3) Execute a instrução;
- 4) Retorne ao passo 1.

Ele escreve essas regras no quadro-negro (ou exhibe e afixa uma cartolina ou folha de *flip chart*) onde ele as escreveu previamente. Note que a ordem das regras está correta, como será explicado na seção 4, mas isso não deve ser mencionado aos estudantes.

A UCP executa a primeira instrução da seguinte maneira: Primeiro, ele/ela vai para a posição 30, lê em voz alta o seu conteúdo ("+, 0, 0, 0, 0"), volta-se para o acumulador, passa o apagador na cartolina que ele está segurando (mesmo se não há nada escrito nela!), e escreve nessa cartolina os símbolos lidos na posição 30, isto é, +0000.

É importante enfatizar para a classe que a UCP somente pode memorizar o conteúdo do acumulador ou de alguma posição de memória *durante* o processo de executar uma instrução. Depois de tê-la executado, esses conteúdos devem ser "esquecidos". É interessante perguntar ao estudante da UCP, no meio da execução, o conteúdo da posição de "memória" usada ou do acumulador empregados em uma instrução anterior. A UCP invariavelmente diz esses conteúdos, dando ao instrutor a chance de enfatizar aquela regra.

O instrutor deixa a UCP executar mais instruções, dizendo como executar instruções novas à medida em que elas aparecem. Por exemplo, a instrução de entrada em 03 significa que um número deve ser dado pelo estudante USUÁRIO. Uma maneira de fazer isso é ele digitar o número em um teclado de computador (sempre na forma *sddd*, onde *s* é ou o sinal + ou o - e *d* é um dígito decimal (0 a 9) - zeros devem sempre ser escritos à esquerda se o valor absoluto do

número desejado é menor do que 1000!). Enquanto o usuário está digitando, o estudante com a cartolina pendurara ENTRADA DE DADOS deve observar exatamente qual tecla está sendo pressionada, e deve dizer isso em voz alta à UCP, de modo que todos os alunos da classe possam ouvi-lo. Isso simula as conexões, barramento (*bus*) entre o teclado e a UCP. A UCP deve memorizar o número inteiro de entrada, por exemplo "+", "0", "2", "4", "3" e depois que o último dígito foi pressionado, o número inteiro deve ser escrito com o giz na cartolina sendo segurada pelo estudante da posição especificada na instrução (no caso, 45). Para fazer isso, a UCP apaga o que estava previamente nessa cartolina.

O instrutor pode contar aos estudantes que, para simular um computador real, quando o usuário pressiona um número inteiro no teclado, ele deve ser armazenado pela unidade de teclado em uma "memória" temporária (*buffer*), cujo conteúdo é entregue à UCP no começo da execução de uma instrução de entrada de dados, pois o movimento de dados dentro da UCP tem sempre o formato de um sinal mais 4 dígitos, sendo essas 5 partes transmitidas em paralelo (noção de "palavra"). Mas, no teatrinho, é mais simples se o usuário digita seu número precisamente quando a UCP começa a execução de uma instrução de entrada e o estudante ENTRADA DE DADOS conta para a UCP em voz alta o sinal e cada dígito pressionado, sem memorizá-los. É também interessante perguntar de repente ao estudante do teclado qual foi o último número que ele deu à UCP. Invariavelmente o primeiro diz o número; o instrutor aproveita para recordar que no nosso computador, o teclado não tem nenhuma memória.

Coordenado por Roberto Hirata Jr., os estudantes que implementaram o simulador do HIPO, e aplicaram várias vezes o Computador a Papel em um programa de um dia de introdução à computação para alunos de última série do ensino médio (naquela época, poucas pessoas tinham microcomputadores em casa) (11), também implementaram um programa especial para ajudar a execução da instrução de entrada de dados. Um PC foi programado para aceitar a entrada pelo teclado, forçando o formato de dados do HIPO (sinal mais 4 dígitos decimais), e exibindo o que foi entrado no monitor. O estudante da UCP lia o número na tela, e todos os estudantes podiam seguir o processo pois a imagem era também projetada em uma tela grande.

A execução da instrução de desvio condicional na posição 06 deve ser explicada cuidadosamente. A UCP lê o conteúdo do acumulador (algumas vezes o estudante esquece de lê-lo, e conta o seu conteúdo de cor; é uma boa oportunidade de enfatizar que a UCP não tem memória interna para lembrar o que havia lido durante a execução de instruções precedentes), vê se ele tem um sinal "-" e é diferente de zero (0000), e então move o apontador de instruções para o endereço especificado na instrução (endereço 11, no nosso caso).

Depois de explicar a instrução de desvio incondicional na posição 10, talvez fosse interessante chamar a atenção dos estudantes para o fato de que, se as regras 2 e 3 acima tivessem sido trocadas, o desvio seria feito efetivamente para a instrução seguinte à que era referida pelo endereço da instrução de desvio (no exemplo, acabaria sendo feito um desvio para o endereço 04 e não para o 03). Uma maneira de manter essa troca seria sempre desviar para a instrução que antecede o endereço referido pela instrução de desvio. Essa explicação poderia ser deixada para um estágio posterior, depois de os estudantes terem deduzido o que o programa faz, o que mostra uma compreensão do funcionamento básico do Computador a Papel.

A instrução de exibição de dados da posição 11 é executada pela UCP lendo em voz alta o conteúdo do endereço especificado pela instrução, isto é, 03, contando-o ao estudante que faz o papel UNIDADE DE SAÍDA. Este último deve escrever o número num pedaço de papel ou no quadro-negro, sempre seguindo o formato de sinal e 4 dígitos. O próximo número a ser exibido deve ser escrito embaixo do precedente, etc. Toda a classe deve poder ver os números escritos, de modo que se se usa um papel, ele deve ser mostrado para a classe depois de cada novo número escrito. Aqui também o instrutor poderia perguntar de repente ao estudante da UCP "que número você exibiu na saída de dados?", mostrando que a UCP não pode memorizar o que ela tinha feito anteriormente (e nem consegue ler os números que foram escritos!).

Quando executada, a instrução de parada da posição 12 deve colocar a UCP em um estado de espera. Somente quando o usuário torce delicadamente sua orelha a UCP deve começar a executar novamente o programa.

## **5. Obtenção do resultado e sugestão de variações**

Como mencionado, é importantíssimo que o instrutor deixe os estudantes deduzirem por si próprios o que o programa faz. Na minha experiência, depois de 2 ou 3 ciclos os estudantes notam que a posição 40 está armazenando a soma dos números de entrada. Se isso não acontece, o instrutor deve dizer ao usuário para digitar um número negativo, e ver o que ocorre. Se a soma aparece na saída produzida pela unidade de saída, embaixo dos números de entrada que foram escritos, em geral os estudantes notam nesse ponto o que o programa executa.

É importante justificar a existência da instrução de saída no endereço 04: sem ela, o usuário não veria o número que digitou no teclado! Pode-se aqui contar que, sempre que em um computador real se digita um caractere, alguma instrução de entrada de dados está sendo executada; para cada caractere digitado, uma instrução de saída de dados produz a exibição daquele caractere na tela.



O instrutor pode usar a configuração do Computador a Papel para sugerir algumas variações do programa ou da configuração. Por exemplo, o que aconteceria se o usuário parasse de repente de digitar novos números? (A UCP, durante a execução da instrução de entrada de dados na posição 03, iria parar aguardando um dado da unidade de entrada.) Ou, o que aconteceria se não entra nenhum número negativo na entrada? O que aconteceria se a instrução de desvio condicional fosse substituída por uma de desvio incondicional? E assim por diante.

Depois disso, o instrutor pode dispensar os estudantes que tomaram parte no teatrinho, e descrever o que todos aprenderam, relacionando todos os conceitos vistos com computadores reais. Por exemplo, por que cada instrução tinha somente um endereço? Computadores com instruções contendo 2 ou 3 endereços foram usados e ainda são usados, mas computadores com um só endereço em cada instrução são muito mais simples. O primeiro computador tipo *mainframe* de porte médio da USP, um Burroughs 3500 tinha instruções de 2 e de 3 endereços; um exemplo da última era uma instrução de soma dando-se os endereços das parcelas e o endereço onde deveria ser armazenado o resultado. Ele poderia também chamar a atenção para conceitos importantes que foram aprendidos: programa armazenado, endereço, instrução e suas partes, decodificação de uma instrução, desvios, acumulador, UCP, como funcionam unidades de disco magnético e SSD, como programas são inicialmente carregados na "memória", etc.

Foi usada a palavra "acumulador" – o registrador onde fica o resultado de uma operação aritmética, e também contém um dos termos dessa operação, possibilitando assim que as instruções tenham apenas um endereço – pois no começo das máquinas binárias, o acumulador era o único registrador acoplado à unidade aritmética. Posteriormente, apareceram registradores de índices (*index registers*), nos quais eram armazenados endereços da "memória", permitindo endereçamento indireto e acesso eficiente a elementos de matrizes de uma dimensão. Mais tarde, foram introduzidos múltiplos registradores, que podiam funcionar tanto como acumuladores ou termos para instruções aritméticas, como índices – mas nesse caso cada instrução deve especificar quais registradores devem ser usados. No entanto, não há muito sentido em mencionar registradores de índices nesse nível (o ideal é esperar a introdução de uso de matrizes na linguagem de "alto nível", se se tratar de uma disciplina da introdução à computação). É recomendável simplesmente mencionar que hoje em dia os computadores usam múltiplos acumuladores que são chamados de "registradores". Normalmente, nosso Apontador de Instruções é chamado de Contador de Instruções. Isso é baseado no fato de que ele é uma unidade aritmética simples por si próprio, sendo capaz apenas de incrementar de 1 um endereço de "memória", para deslocar-se para a próxima posição se não houver um desvio. Escolhi aquele nome por representar melhor a finalidade daquele registrador.

Depois dessa aula, deve seguir-se uma em que as instruções do Computador a Papel sejam codificadas no formato das instruções do computador HIPO, usando um código de instrução de dois dígitos decimais, escritos na parte mais à esquerda de uma posição de memória, depois do sinal. Assim, cada posição da memória armazenaria um sinal e um número de 4 algarismos, que pode ser interpretado pela UCP como uma instrução fazendo referência a um único endereço de memória (nos dois últimos dígitos à direita), ou a um dado numérico. Aí pode ser introduzida a noção de codificação alfanumérica (2 dígitos para cada caractere, isto é, 2 caracteres por palavra), bem como a notação de ponto flutuante (os dois dígitos mais à esquerda na notação "excesso 50" do expoente de 10, os dois mais à direita ou com mais 4 da próxima palavra representando a mantissa). Esses conceitos podem ser encontrados no artigo sobre o computador HIPO em meu *site* (8).

## **6. Conclusões**

Na minha experiência e de todos que utilizam essa técnica, os estudantes adoram essa aula, pois ela não é puramente direcionada para seus intelectos. Se um estudante extrovertido é escolhido para fazer o papel da UCP, a aula será muito divertida. Ver o funcionamento do Computador a Papel deixa uma impressão muito forte na lembrança dos estudantes. Cheguei a aplicar várias vezes esse teatrinho com adultos, também com muito sucesso.

Quando ministrava disciplinas de introdução à computação, eu mostrava como cada comando de uma linguagem de "alto nível" pode ser traduzida para a máquina HIPO, para que os alunos possam compreender como o computador executa comandos como "if...then...else...", "while...do...", etc. Durante essas aulas eu fazia referência aos estudantes que tinham feito papéis no teatrinho, como "José, que era o acumulador...".

Os conceitos do Computador a Papel e o HIPO podem ser usados, por exemplo, quando se ensina um editor de textos, mostrando como ele funciona. Depois de uma linha com um número fixo de caracteres, viria uma posição de memória contendo um endereço apontando para o começo da nova linha, que pode estar em outra região da unidade de armazenamento. Com isso, teríamos uma *lista ligada* de linhas, sendo muito fácil inserir ou eliminar linhas. Ao introduzir planilhas eletrônicas, é fácil mostrar que o conteúdo de uma célula pode ser simplesmente um endereço de uma posição de "memória" contendo um valor ou o começo de uma fórmula. Numa fórmula, a notação de coordenadas pode ser traduzida para o endereço do conteúdo da célula. Dessa maneira, podem ser ensinados pelo menos certos princípios básicos do funcionamento de um *software* de aplicação. Estaríamos dessa maneira educando as pessoas a entenderem o que fazem as máquinas que elas usam, mostrando que não há mistério nelas.

Além disso, o Computador a Papel e o HIPO podem ser usados para chamar a atenção para restrições importantes impostas pelos computadores, como o fato de serem deterministas, condenados a executar seus programas, que os computadores não tomam decisões, mas fazem escolhas lógicas (baseadas nos desvios condicionais), que tudo dentro de um computador dados, textos, programas – tem que ser quantificado, etc.

Penso que o ensino de cada novo conhecimento sobre qualquer tecnologia deveria ser sempre seguido de uma crítica sobre suas restrições e o comportamento que ela força sobre o usuário. No caso de computadores, isso é claro: quando um usuário qualquer dá um comando à máquina ou um programador está desenvolvendo programas, eles têm que exercer uma classe muito restrita de pensamentos, de fato, pensamentos que podem ser inseridos na máquina e corretamente interpretados por ela. Esse “pensamento maquinal” deve ser compensado por um pensamento mais intuitivo, não formal, tal como o exercitado quando se faz alguma atividade artística, que considero o antídoto correto para um excesso daquele tipo de pensamento (12).

## Referências

(1) Setzer, V.W. Os meios eletrônicos e a educação, no lar e na escola: uma síntese de problemas e recomendações. 2022. Acesso em 17/3/24:

<http://www.ime.usp.br/~vwsetzer/meios-educacao-sintese.pdf>

(2) Setzer, V.W and Monke, L. “Challenging the Applications: An Alternative View on Why, When and How Computers Should Be Used in Education,” in Muffoletto, R. (Ed.), Education and Technology: Critical and Reflective Practices. Cresskill, New Jersey: Hampton Press, 2001, pp. 141-172.

(3) Setzer, V.W. *Os Meios Eletrônicos e a Educação: uma Visão Alternativa*. São Paulo: Ed. Escrituras, Coleção “Ensaio Transversais” Vol. 10, 2001 (3a. ed. 2005).

(4) Lanz, R. *A Pedagogia Waldorf – Caminho para um Ensino mais Humano*. São Paulo: Antroposófica, 13ª edição 2019 (contém um apêndice meus sobre meios eletrônicos na educação).

(5) Pedagogia Waldorf. Há muito material na internet. Ver, por exemplo (acesso em 17/3/24):

<https://www.sab.org.br/movimento-antropos%C3%B3fico/aplica%C3%A7%C3%B5es-da-antroposofia/pedagogia-waldorf>

(6) Setzer, V.W. Python – folha de consulta. Acesso em 17/3/24: <https://www.ime.usp.br/~vwsetzer/python-opers-funcoes.html>

- (7) Setzer, V.W. e I.S. Homem de Melo. *A Construção de um Compilador*. Rio de Janeiro: Ed. Campus, 1982.
- (8) Setzer, V.W. The HIPO computer – a tool for teaching basic computer principles through machine language. Acesso em 17/3/24: <https://www.ime.usp.br/~vwsetzer/HIPO/hipo-descr.html>
- (9) Hariki, V.S. Simulador do HIPO. Acesso em 17/3/24: <https://seijihariki.github.io/HIPO-simulator/#/>
- (10) Setzer, V.W. How to use the Seiji Hariki hipo simulator. Acesso em 17/3/24: <https://www.ime.usp.br/~vwsetzer/HIPO/using-the-HIPO-simulator.html>
- (11) Setzer, V.W. and R. Hirata Jr. O Dia da Computação – uma breve introdução aos computadores e à computação. *Caderno da Revista do Professor de Matemática*, São Paulo, Vol. 4, No. 1, 1993.
- (12) Setzer, V.W. Um antídoto contra o pensamento computacional. Acesso em 17/3/24: <https://www.ime.usp.br/~vwsetzer/antidoto.html>