

O Computador como Ambiente para Exploração da Matemática e Física

André Santanchè
UNIFACS - Universidade Salvador

1. Introdução

A máquina pode arranjar e combinar seus valores numéricos exatamente como se fossem letras ou qualquer outro símbolo genérico; e na verdade ela poderia apresentar seus resultados em notação algébrica, desde que se adotem medidas adequadas. (Ada Augusta, 1844, apud [KNU73])

No século XIX, Ada Augusta desenvolveu trabalhos com Charles Babbage na construção do primeiro modelo de máquina que foi precursora dos computadores modernos. Desde então, é possível se observar o início de uma idéia que depois amadureceria com os anos e se transformaria em um dos mais poderosos conceitos da computação: o digital.

A palavra digital vem de dígito, e faz referência a dois dígitos, em especial, que são utilizados na computação: 0 e 1. Porque são apenas dois dígitos e qual a sua importância? Quando estudamos geometria, aprendemos que a menor coisa que se pode representar, geometricamente falando, é um ponto. Este ponto não tem dimensão, indica apenas a sua existência em um local no plano ou no espaço.

Por ser desta forma, o ponto dá origem a todas as demais formas. Podemos imaginar qualquer uma das formas geométricas como uma coleção de pontos dispostos de alguma maneira. Uma reta pode ser entendida como um conjunto infinito de pontos dispostos linearmente. O mesmo ocorre com os dados e as informações.

Os dados podem ser concebidos como representações simbólicas de algo. Esta representação pode-se dar de diversas maneiras, sob a forma de textos, imagens, desenhos, etc. Quando atribuímos um significado aos dados, obtemos então informações.

Por ser o computador uma máquina de processar dados, o desafio consistiu, há muitos anos, em encontrar a menor unidade, o “ponto geométrico”, que pudesse ser utilizada então para se representarem os dados.



Este texto é concedido sob uma [Licença Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/). Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença. Para maiores informações veja licença completa no [site creativecommons.org](https://creativecommons.org/).

O filósofo inglês Francis Bacon foi o inventor do código binário. Utilizando meios simples, que pudessem representar apenas dois estados como clarins (toca, não toca), tiros de mosquete (atira, não atira), etc., ele pretendia transmitir pensamento à distância [BRE91], p.55).

Se experimentarmos reduzir qualquer representação ao seu estado mais elementar, verificaremos que alcançaremos sempre um estado binário (existência ou inexistência de algo). Assim se dá com os números que, em sua forma mais elementar, se resumem a dois possíveis dígitos, 0 e 1.

Claude Shannon, em 1938, desenvolveu uma tese que permitia a representação desses dois dígitos e das operações algébricas entre eles através de circuitos de comutação elétrica. Ele lança então o conceito de BIT (forma contracta de (B)inary digiT) que faz referência aos dígitos binários (0 e 1) e constrói uma poderosa ponte entre a representação abstrata de dados e informações e as máquinas. Está instaurada a época do digital.

Por ser o BIT a menor unidade, gradativamente se mostrou ser viável a representação de qualquer tipo de informação através deles. O BIT se tornou o “ponto geométrico” da computação a partir do qual tudo é moldado.

Nos últimos anos, estas novas tecnologias têm acentuado sua atuação na sociedade. Muitas delas têm adotado ou se convertido ao padrão digital, deste modo, se interligam e multiplicam as possibilidades:

“[...] a convergência da base tecnológica, [...], decorre do fato de se poder representar e processar qualquer tipo de informação de uma única forma, a digital. Pela digitalização, a computação (a informática e suas aplicações), as comunicações (transmissão e recepção de dados, voz, imagens etc.) e os conteúdos (livros, filmes, pinturas, fotografias, música etc.) aproximam-se vertiginosamente – o computador vira um aparelho de TV, a foto favorita sai do álbum para um disquete, e pelo telefone entra-se na Internet”. [TAK00]

Torna-se cada dia mais claro que a educação não pode posicionar-se diante das novas tecnologias, como se elas fossem meras ferramentas digitais, que lhe dão recursos antes não existentes, para incrementar aulas e atividades de ensino/aprendizagem.

Há que considerar-se os novos modos de pensar, compreender e comunicar-se, decorrentes de aprendizes imersos em um mundo tecnológico, e repensar a prática pedagógica que agora deverá lidar com novas ‘linguagens’.

Em paralelo, é verdade também que as novas tecnologias trazem consigo um universo de recursos e possibilidades, que abrem novas perspectivas no ensinar e aprender. Estão aí: o hipertexto e a hiperídia, simulações por computador, ambientes e linguagens para a construção de projetos dos mais variados, etc..

Em última instância, toda a tecnologia de computação que se têm desenvolvido até hoje está fundamentada essencialmente na matemática. O computador é uma máquina matemática.

Por este motivo, uma das mais promissoras áreas para o desenvolvimento de atividades de ensino-aprendizagem é a matemática e por consequência a física, cujos princípios se fundamentam na matemática.

Seymour Papert estabelece uma interessante comparação entre o aprendizado de matemática utilizando sua linguagem LOGO, com o aprendizado do francês na França:

“O que aconteceria se as crianças que não conseguem entender Matemática fossem criadas em Matematicolândia, um lugar que fosse para a Matemática o que a França é para o francês? Muitos professores aceitaram o desafio de construir algo como uma Matematicolândia em suas salas de aula e adotaram o LOGO e sua tartaruga como material de construção.” [PAP80]

Dentro desta perspectiva, buscamos apresentar neste texto algumas linhas de atuação da informática em atividades de ensino-aprendizagem de matemática e física. Para cada linha, é apresentada uma seleção de *softwares* gratuitos ou sistemas abertos, que dão subsídios ao desenvolvimento das práticas.

A ênfase será dada, no entanto, aos *softwares* e seu modo de operação, mas aos fundamentos dos domínios nos quais cada um deles se insere. Acreditamos que a compreensão destes fundamentos é de vital importância para ampliar os horizontes de percepção do papel do computador, no desenvolvimento das atividades de ensino-aprendizagem em matemática e física.

A simulação é uma área riquíssima na qual podemos ensaiar modelos e coletar dados resultantes do processo. Tanto o modelo, quanto à análise dos dados resultantes, pode ser objeto de estudo na matemática e na física. No tópico **Simulação** são analisadas duas técnicas de simulação específicas, baseada autômatos celulares e baseada na dinâmica de sistemas.

Em **Modelos Matemáticos**, são analisados programas que, a partir de modelos matemáticos, representados na forma de expressões, são capazes de apresentar ao usuário o resultado por meio de: gráficos, animações, tabelas, etc. Conjuntamente com estes programas, é apresentada uma abordagem que não se restringe à apresentação de modelos prontos ao aluno, mas que o motiva à participação ativa na construção dos mesmos. O software se transforma um ambiente de expressão, onde são lançados desafios e são propostas soluções.

No tópico **Geometria Plana e Espacial**, o computador é apresentado como um espaço aberto para a composição geométrica em duas e três dimensões. Cada programa atua como uma ferramenta de abstração, onde concepções mentais são transformadas em gráficos.

Finalmente em **Composição**, são apresentados sistemas que utilizam ambientes de construção baseados em componentes de software, ou em atores, para que o aluno possa elaborar e executar modelos, onde as peças se comportam segundo regras e interagem entre si, resultando em um comportamento mais complexo. Nesta abordagem, os componentes atuam como peças de Lego® que são projetados para a composição de um modelo maior. Tais ambientes motivam a atuação ativa do aluno na construção de seu objeto de estudo.

2. Modelos Matemáticos

A construção de modelos matemáticos baseados em expressões é uma das técnicas que se apresentam mais familiares a professores de matemática e física. Por este motivo, foi escolhido como o primeiro tema a ser abordado.

Os *softwares* apresentados não se limitam a replicar as possibilidades já exploradas em sala de aula para apresentação de gráficos. Eles oferecem diversos recursos de exploração com gráficos e animações, além de acrescentarem novos elementos de expressão tipicamente associados a recursos tecnológicos, como apresentaremos adiante.

Este ferramental é bastante apreciado pelos professores de matemática e física, como instrumento para a construção de belas e interessantes apresentações. Por outro lado, seu verdadeiro poder se manifesta quando o aluno assume o papel de autor e explorador.

Estudo de Caso

Para a apresentação de *softwares* baseados em modelos matemáticos e aqueles utilizados para simulação com propósitos educacionais (tópico seguinte), optamos por um estudo de caso, cujo objetivo é apresentar as múltiplas formas de se encarar o mesmo problema, fazendo uso de diversas abordagens que nos permitem estes *softwares*.

Apresentaremos o tema central do estudo de caso na forma do seguinte problema:

Dada uma população de herbívoros que cresce a cada ano 9% e uma outra população de carnívoros que cresce a cada ano 5%. Construir um modelo ou sistema que se comporte conforme os parâmetros apresentados, permitindo uma análise comparativa dos resultados. A população de herbívoros inicia com vinte animais e a de carnívoros com oito.

WinPlot

O WinPlot é um software, produzido por Richard Parris, cujo principal objetivo é a plotagem de curvas e superfícies em duas ou três dimensões. Ele possui recursos de animação, que permitem explorar diversos aspectos da construção de gráficos. Ele possui uma versão em português cuja tradução foi conduzida pelo professor Adelmo Ribeiro de Jesus.

Endereço: <http://math.exeter.edu/rparris/winplot.html>

Na figura 1 apresentamos um gráfico elaborado no WinPlot a partir de duas equações que expressam o crescimento das populações através de Progressão Geométrica.

A curva em azul, traçada a partir da função $y = 20 \times 1,09^x$, representa a população de herbívoros. Já a curva em vermelho, traçada a partir da função $y = 8 \times 1,05^x$, representa a população de carnívoros.

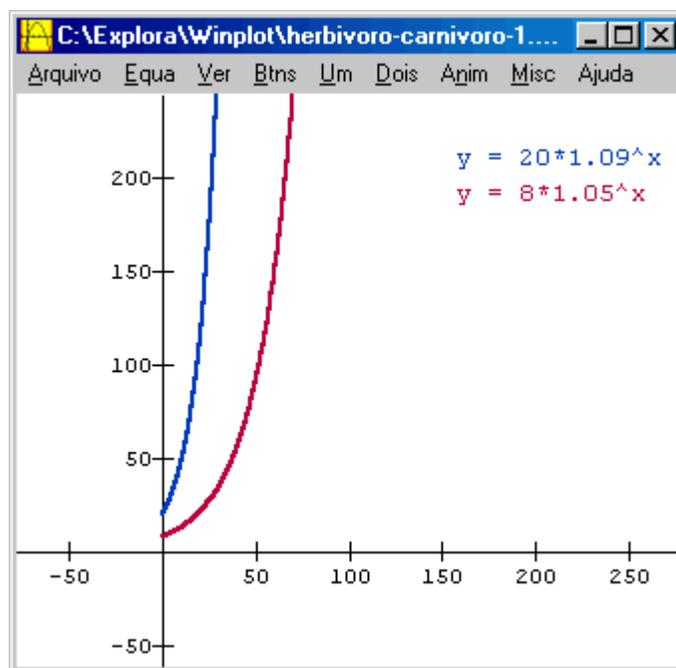


Figura 1: Gráfico de população herbívoros x carnívoros traçado no WinPlot

Além do traçado de curvas, cujos parâmetros podem ser modificados dinamicamente, repercutindo imediatamente no gráfico apresentado, o WinPlot dispõe de interessantes ferramentas que oferecem a possibilidade de apresentar de forma animada, ou explorar de forma interativa, a influência de cada elemento da função no gráfico produzido.

A figura 2 ilustra a aplicação da ferramenta na função dos carnívoros. Para isto foi criada uma equação onde a taxa de crescimento desta população que era fixa em 1.05, foi substituída por uma variável "a".

Em seguida, como mostra a caixa de diálogo na figura, o usuário estabelece uma faixa de variação de valores para família de curvas. Deste modo, o WinPlot traça para cada valor da faixa de variação de "a" um novo gráfico. O efeito resultante permite perceber como a curva varia na medida que aumentamos a taxa de variação.

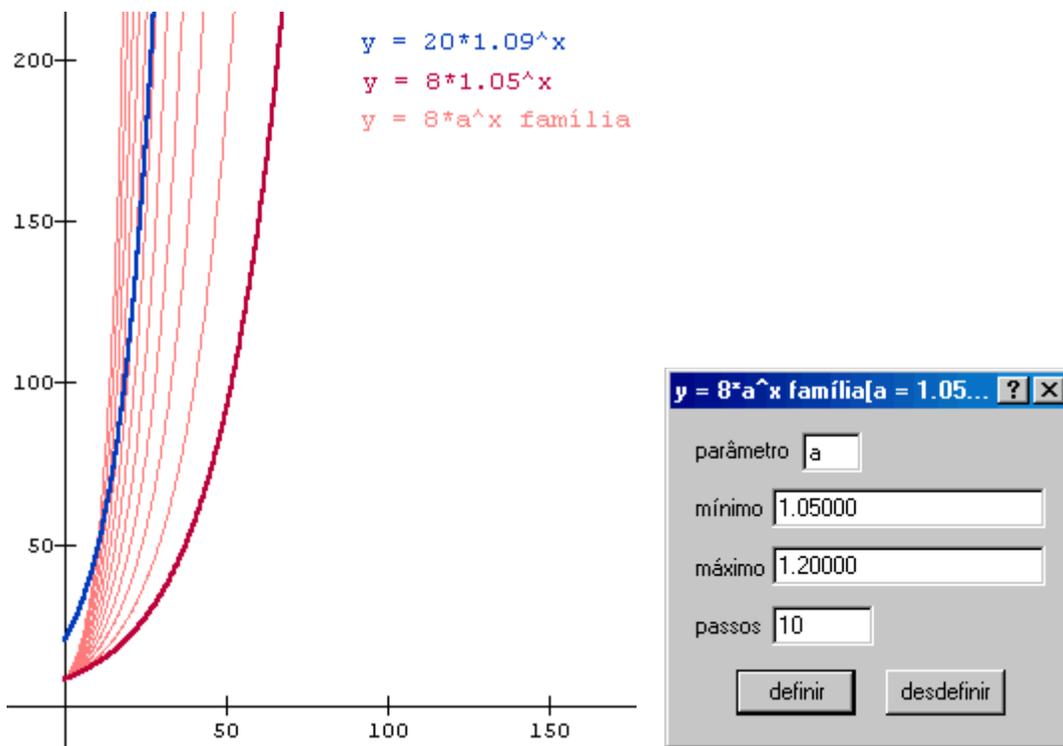


Figura 2: Família de curvas que apresentam alternativas de crescimento dos carnívoros

Modellus

Modellus é um software desenvolvido em português através do qual é possível “criar, simular, e analisar modelos interativamente no computador, seja de dados e imagens experimentais ou do pensamento puramente teórico”.¹ [TEO00]

Endereço: <http://phoenix.sce.fct.unl.pt/modellus/>

Tal como o WinPlot, o Modellus também é capaz de traçar gráficos a partir de funções como as apresentadas. Mas acreditamos que seu verdadeiro diferencial está na possibilidade de interligar os modelos matemáticos, a elementos animados que permitem a intervenção do usuário.

Na figura 3 apresentamos uma animação, que ilustra o processo de crescimento da população de herbívoros em confronto com a de carnívoros.

¹ Tradução nossa do original: “create, simulate, and analyze models interactively on the computer, either from experimental data and images or from pure theoretical thinking”.

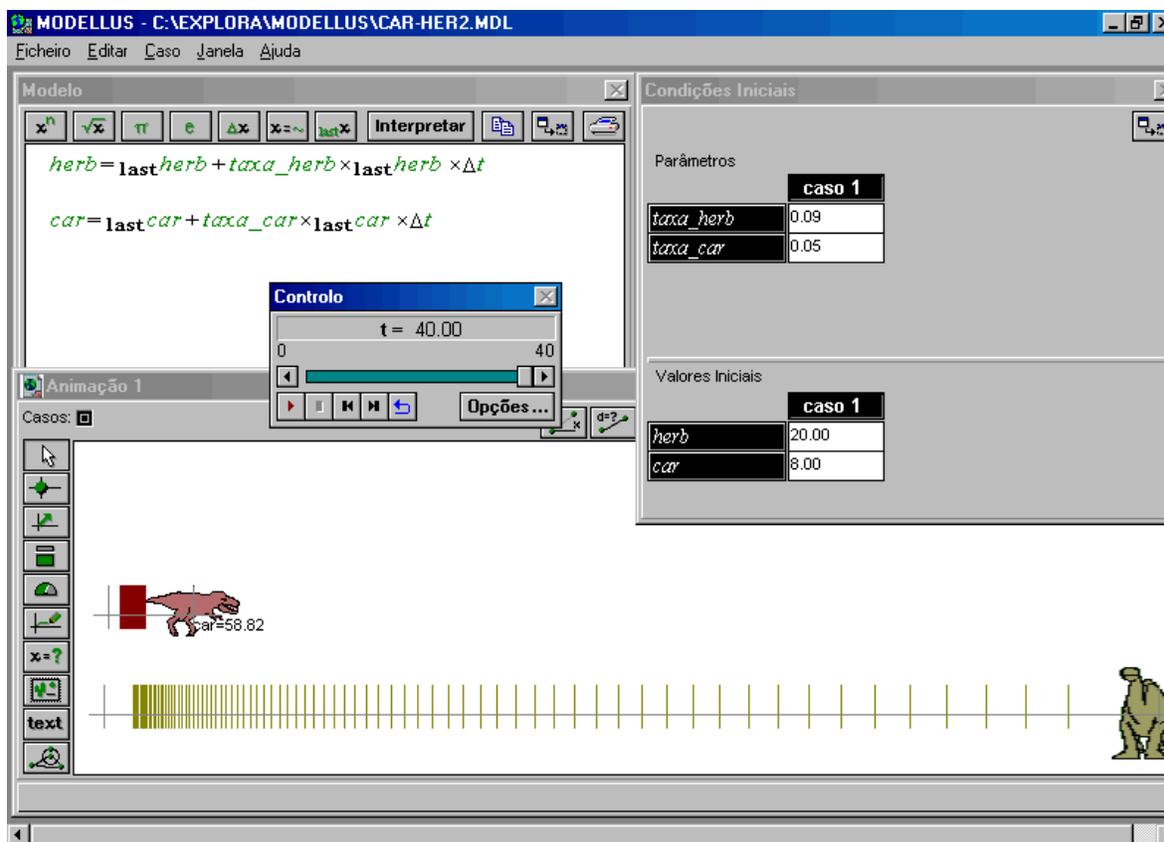


Figura 3: Animação ilustrando o crescimento de populações de herbívoros e carnívoros

Para a construção desta animação, foi utilizada uma poderosa forma alternativa de modelo iterativo. Este tipo de modelo permite o cálculo do novo valor de uma variável, a partir do valor do ciclo imediatamente anterior (técnica bastante familiar para programadores de computadores). O exemplo apresentado se baseou nas equações:

$$\begin{aligned} herb &= \text{last } herb + taxa_herb \times \text{last } herb \times \Delta t \\ car &= \text{last } car + taxa_car \times \text{last } car \times \Delta t \end{aligned}$$

A palavra “*last*” é utilizada para identificar o valor da variável indicada, no ciclo imediatamente anterior.

Além de ser uma outra forma de expressar o fenômeno estudado, o que permite explorar o modelo a partir de uma perspectiva diferente, esta notação permite a criação de animações no Modellus onde é possível a intervenção dinâmica do usuário.

Na figura 4 observamos que foram acrescentados à animação dois elementos circulares ( e ) que cumprem duas funções simultaneamente: apresentam visualmente a taxa de variação dos carnívoros e herbívoros e permitem, através de manipulação direta do usuário, a modificação do valor destas variáveis enquanto a animação está sendo executada.

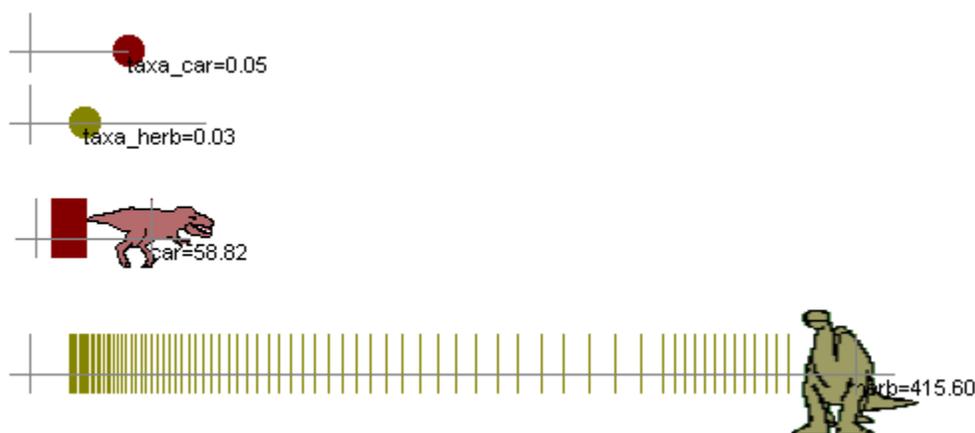


Figura 4: Animação que sofreu intervenção do usuário durante sua execução

No exemplo da figura 4 o aluno, no meio da animação que apresentava o crescimento exponencial dos herbívoros, resolveu reduzir a taxa de variação destes, que era de 9% para 5%.

Como ele pôde constatar na dinâmica, a taxa crescimento dos herbívoros foi diminuindo na medida que ele reduzia a variável.

3. Simulação

Simular é uma especialidade dos computadores. Construídos como máquinas capazes de realizar milhões de cálculos em frações de segundos, seu potencial tem sido usado, desde o primeiro computador, nos mais diversos tipos de simulação.

A educação não poderia deixar de tirar proveito desta capacidade. É justamente sobre simulação que trataremos neste tópico.

3.1. Virtual e Digital

A palavra virtual pode assumir diversos significados a depender do contexto em que é empregada.

Pierre Lévy [LEV96] aponta para a origem da palavra, que vem do latim, *virtualis*, derivada por sua vez de *virtus*, que significa força, potência. Desta maneira, virtual indica o que existe em potência e não em ato. Virtual, sob esta perspectiva, não é o oposto de real, mas um ‘vir a ser’ real.

Gilberto Dimenstein, por outro lado, entende que virtual pode ter uma interpretação genérica de “algo que decorre do atual, como uma espécie de campo de força, algo que se imagina ou deduz a partir do concreto” [DIM98], ou mais específica, ligada aos computadores, como “tudo aquilo que se vê na tela e que pode ser utilizado – como programas e bancos de dados –, mas não existe fisicamente” [DIM98].

Ver e utilizar o que não existe fisicamente tem relação com outra característica marcante do computador, a de ser uma máquina digital.

A representação digital de informações, inicialmente, limitou-se a números e textos, dadas as limitações tecnológicas dos primeiros computadores. No entanto, as máquinas aprimoraram-se e deram origem a uma geração de computadores capazes de representar gráficos e reproduzir sons com grande qualidade, a ponto de criar, aos olhos humanos, uma ilusão de realidade.

A realidade em que vivemos, na verdade, não é dividida em pequenos pedaços, tal como acontece no computador ou, mesmo que o seja, não encontramos, por assim dizer, o menor pedaço. Para poder criar a ilusão de uma coisa real, o computador aproveita-se dos limites dos órgãos de sentido humanos. Nossa visão, audição, etc. possuem um limite de percepção. Se desenharmos dois pontos muito próximos um do outro e afastarmos-nos o suficiente, nosso sentido de visão naturalmente irá fundir ambos em um único ponto. O mesmo ocorre com a noção de movimento, se apresentarmos a um observador várias fotografias de um pássaro voando, de tal forma que o tempo entre uma fotografia e outra seja mínimo, seu cérebro irá fundir a seqüência em um movimento único e contínuo, e ele 'verá' o vôo do pássaro como se não houvesse interrupções. Isto é extensamente explorado pelo cinema, pela televisão e pelos computadores.

Se observarmos a fotografia na figura 5, a veremos à direita tal como é apresentada por um computador. Ela parece-nos, a princípio, uma imagem única e completa. No entanto, se ampliarmos bastante a imagem, verificaremos que ela consiste em um conjunto de pontos coloridos tão pequenos e tão próximos, que nos transmitiram a ilusão de uma imagem única. O conjunto de dados e programas de computador transmite-nos a sensação de algo que existe, algo concreto, mas que na verdade não passa de uma representação abstrata em sistemas de computação, eis aí o virtual. O virtual, apesar de não possuir existência física, pode ser utilizado através dos computadores e pode inclusive conduzir à idéia de que é um elemento real.

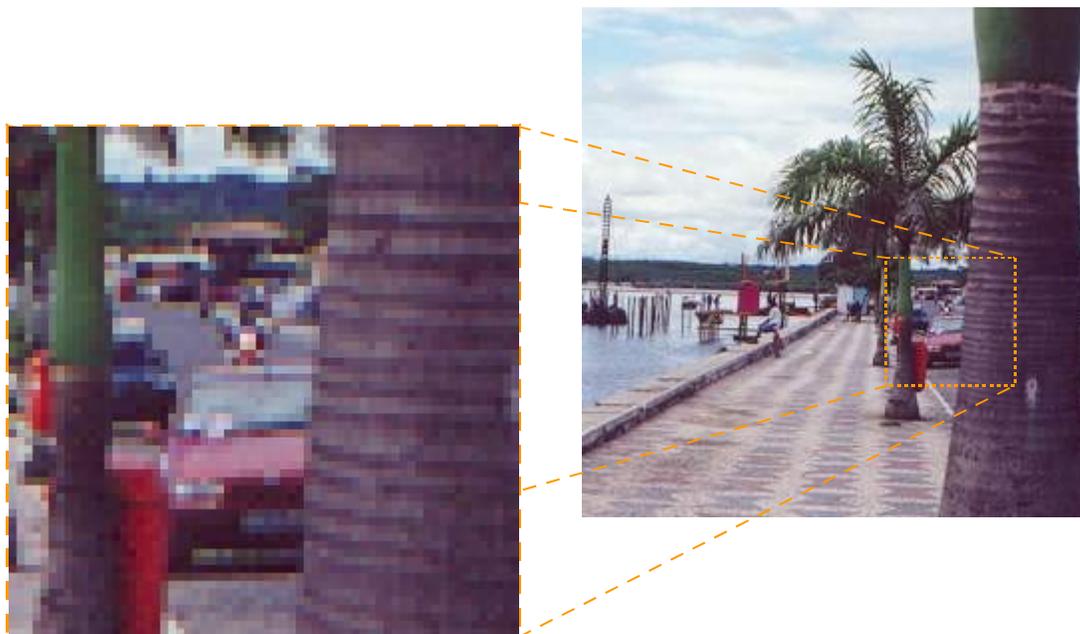


Figura 5: Fotografia digital com detalhe ampliado

3.2. Simulação

Para transmitir-nos a sensação de ‘real’, o computador realiza uma simulação. Segundo o dicionário Aurélio, simular significa “Fingir (o que não é)” [FER89]. Este ‘fingir’ permite-nos interagir com algo que, apesar de não estar fisicamente presente, faz-nos crer na sua existência (ainda que virtual).

Esta capacidade de simulação dos computadores constitui um dos seus mais importantes papéis em termos de educação.

“In a report on educational technology to the President of the USA, a committee of science advisors presented the most promising constructivist applications of technology, with simulations on top of the list”.² [REP99]

Muitos softwares exploraram, de maneiras diferentes, a simulação. Elas são especialmente interessantes em áreas ligadas a ciências exatas, tais como a matemática e a física, cujos fenômenos estudados são dotados de certa previsibilidade.

Alguns sistemas que se baseiam na descrição de modelos matemáticos, podem ser utilizados para a construção de simulações. A figura 6 ilustra uma simulação elaborada no Modellus para estudo de física.

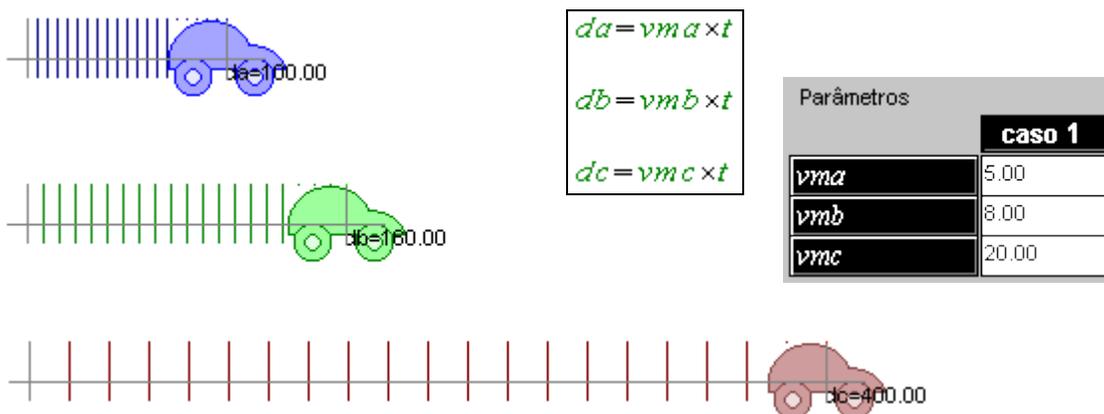


Figura 6: Simulação de Movimento Retilíneo Uniforme, realizada no Modellus

Outra abordagem especialmente interessante para o desenvolvimento de simulações em atividades de ensino-aprendizagem é a dos autômatos celulares.

3.3. Autômatos Celulares

“Autômatos celulares são sistemas dinâmicos discretos cujo comportamento é completamente descrito em termos de relações locais. Um autômato celular pode ser pensado como um universo estilizado. O espaço é representado por um grid uniforme, com cada célula contendo alguns bits de dados; o tempo avança em passos discretos e as leis do universo são expressas em digamos, uma pequena tabela de referência,

² Tradução nossa: “Em um relatório sobre tecnologia educacional para o Presidente dos Estados Unidos, um comitê de consultores científicos apresentaram as mais promissoras aplicações construtivistas da tecnologia, com simulações no topo da lista”.

através da qual a cada passo as células computam seu  o estado a partir do próprio estado e do estado de seus vizinhos próximos” [SMI98]

Um dos mais famosos autômatos celulares foi o “Jogo da Vida” criado por John Horton Conway em 1970 [SMI98]. Baseados em um conjunto de regras simples, e em células que podem assumir apenas dois estados (viva ou morta) ele mostrou que é possível se criar sistemas bastante complexos com comportamento semelhante à vida orgânica.

Sistemas de simulação tem adotado abordagens mistas entre autômatos celulares e objetos ou autômatos celulares e agentes. Os objetos e agentes são elementos que preenchem as células, como veremos adiante esta mixagem alia poderosos recursos de modelagem destas abordagem e tem dado origem a ambientes bastante propícios ao desenvolvimento de atividades de ensino aprendizagem.

Homos
Este software é destinado a simulações com propósitos educacionais e tem como base os autômatos celulares. Em cada célula pode ser colocado um objeto, cuja classe é definida pelo usuário.
O comportamento dos objetos é determinado por um conjunto de regras associadas à classe dos mesmos. O resultado das simulações pode ser traçado em gráficos.

As figuras 7 e 8 apresentam a tela do programa Homos executando a simulação de um cenário onde estão dispostos herbívoros e carnívoros. A cada ciclo, é simulado o comportamento da reprodução destes animais. Os resultados podem ser apresentados em gráfico como o da figura 9.

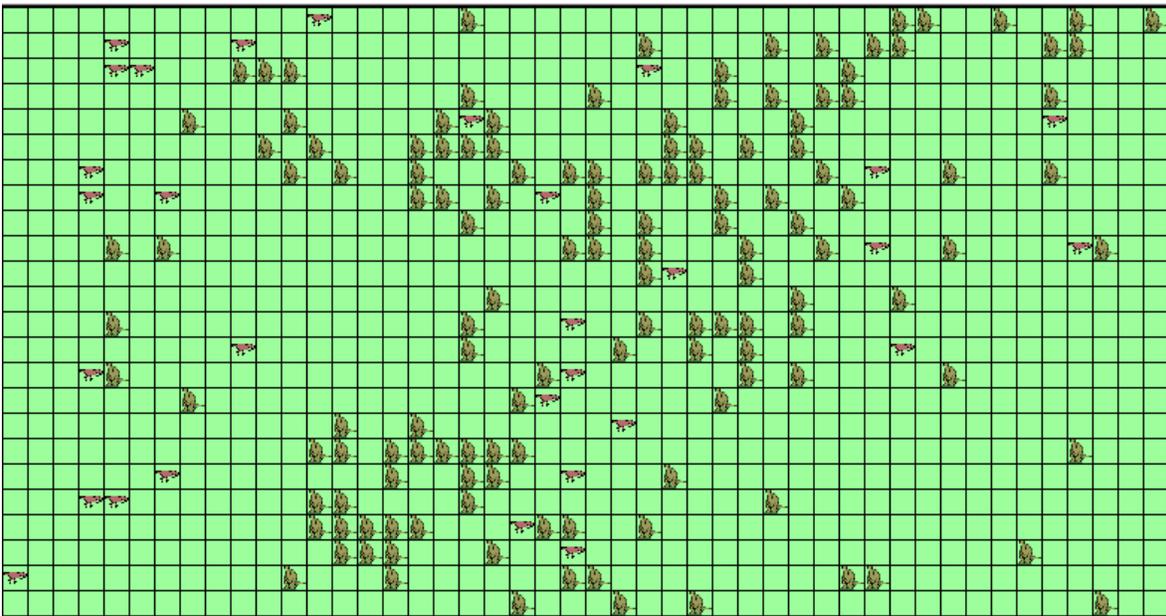


Figura 7: Simulação do crescimento de herbívoros x carnívoros no Homos

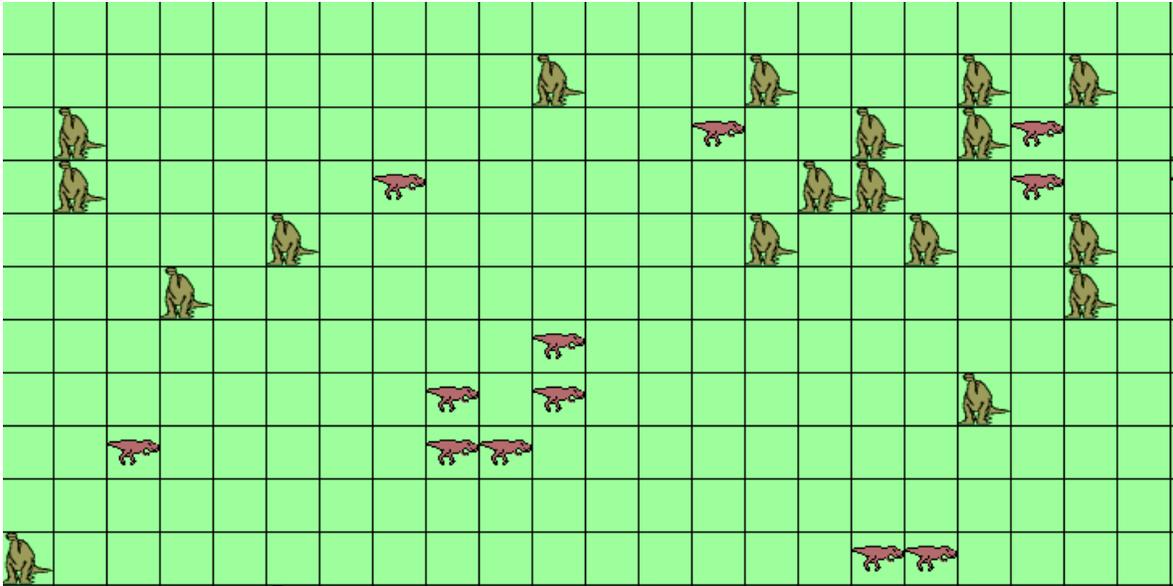


Figura 8: Simulação do crescimento de herbívoros x carnívoros ampliada

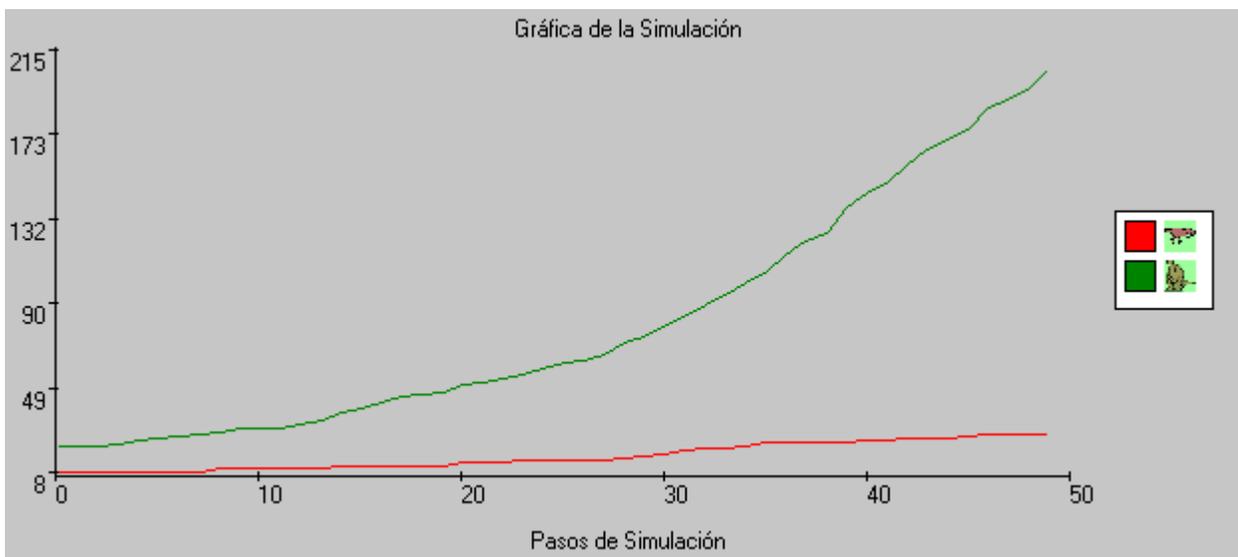


Figura 9: Gráfico gerado a partir da simulação.

Neste tipo de simulação o comportamento de cada objeto, que está disposto em uma célula, é comandado por um conjunto de regras. Estas regras são aplicadas de forma autônoma para cada unidade. Sua interação produz o resultado da simulação.

A figura 10 apresenta uma janela do Homos onde está definida uma regra de reprodução para um herbívoro.

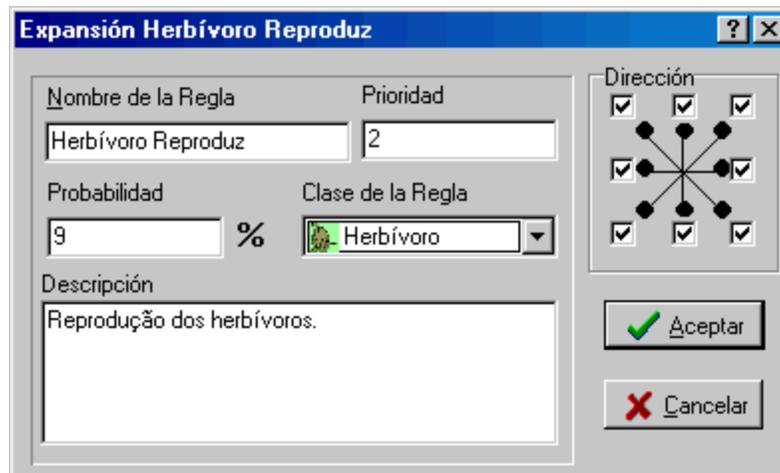


Figura 10: Janela para definição de regra de comportamento do Homos.

Realizar experimentações e analisar resultados são os pontos fortes das simulações quando aplicadas na educação. Como está sendo apresentado neste texto, é possível realizar um trabalho integrado entre vários *softwares* e utilizar os variados resultados como fonte de reflexão e análise sobre o problema estudado.

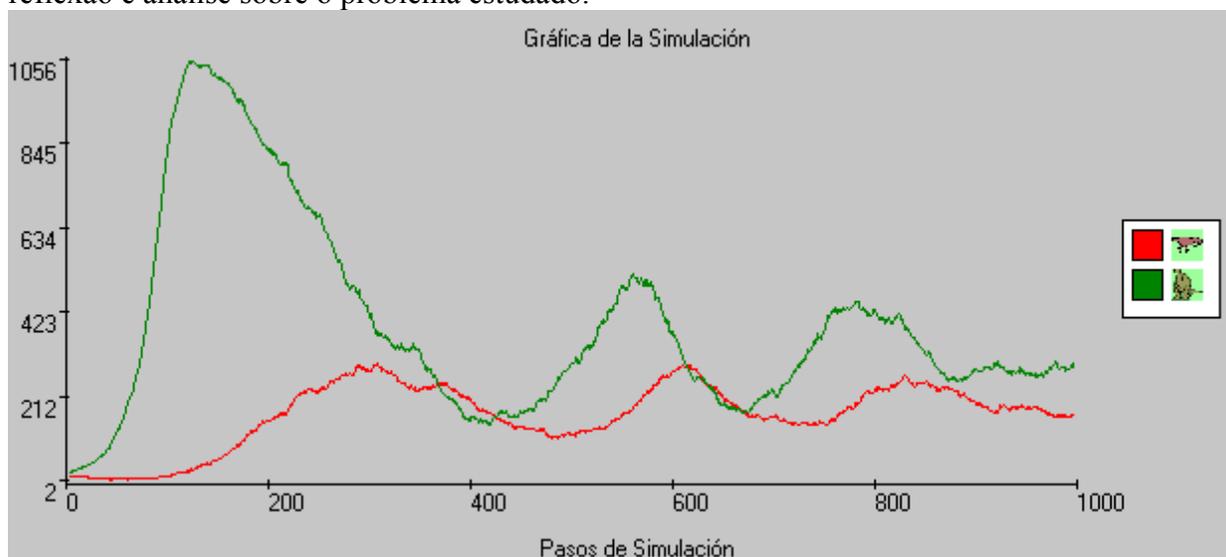


Figura 11: Janela para definição de regra de comportamento do Homos.

A figura 11 ilustra um modelo um pouco mais sofisticado produzido no Homos, onde se estabelece uma interação entre carnívoros e herbívoros (os carnívoros podem devorar os herbívoros). Este modelo tomou como base um interessante modelo predador x presa disponível como exemplo no Homos.

Evolución

Ferramenta de simulação, desenvolvida pelo Grupo SIMON, baseada na dinâmica de sistemas. Fazendo uso da metáfora de um fluxo que deve percorrer um trajeto, construído pelo usuário, o sistema conduz a uma reflexão de fundamentos da matemática e da física sobre uma outra ótica.

Endereço: <http://www.uis.edu.co/investigacion/paginas/grupos/simon/software.html>

Na figura 12, podemos observar o modelo de herbívoros e carnívoros representado sob a ótica da dinâmica de sistemas. A figura  representa um fluxo, transmitido pelo canal (representado pela seta) para um tanque, representado pelo . A figura  representa um registro que controla o fluxo.

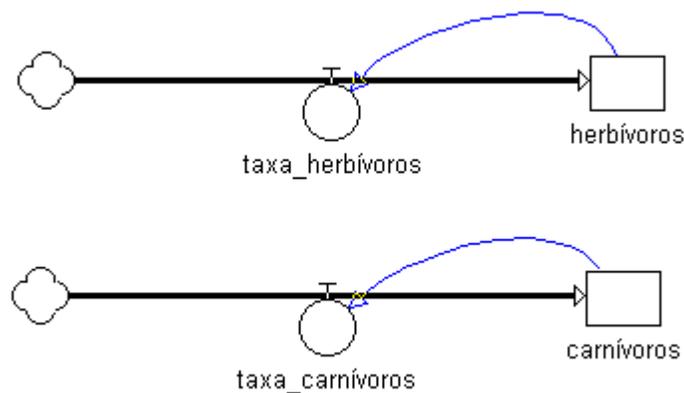


Figura 12: Modelo herbívoro x carnívoro no Evolución.

A seta azul indica uma relação entre o registro e o tanque. Neste caso, o fluxo que o registro deixa passar depende do volume disponível no tanque. A figura 13 ilustra a janela onde é estabelecida a relação.

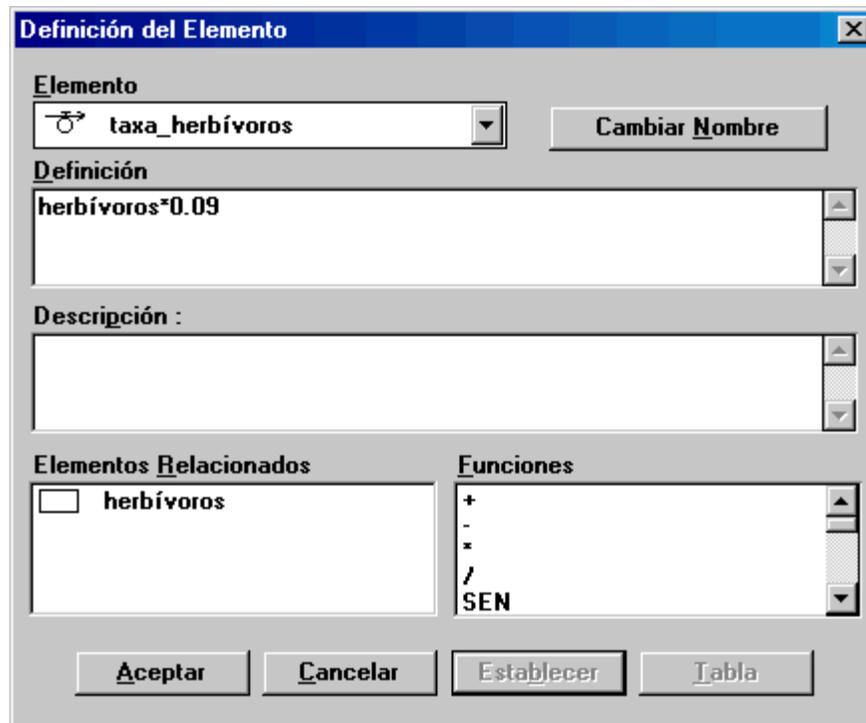


Figura 13: Janela que define regra do controle de fluxo.

O resultado do modelo resulta no gráfico da figura 14.

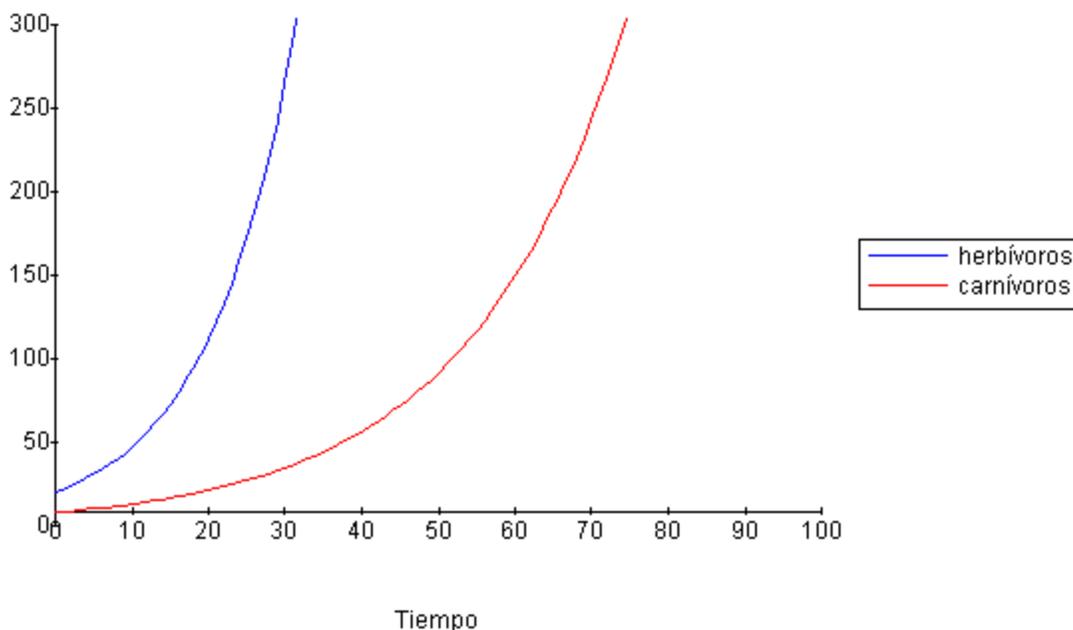


Figura 14: Gráfico resultante do modelo produzido no Evolución.

Além da matemática e física, esse tipo de simulação pode servir como base para outras áreas de estudo, tal como o meio ambiente.

4. Geometria Plana e Espacial

Para compreendermos como o computador atua na geometria plana e espacial, iremos estudar alguns princípios da Computação Gráfica.

"A computação gráfica é parte da ciência da computação e área de estudo de alguns aspectos da comunicação entre o homem e o computador.

O aspecto principal abordado pela computação gráfica é o da comunicação visual no sentido 'máquina-homem', através da síntese de imagens em dispositivos de saída apropriados". [BAN]

A figura 14 apresenta as operações que podem ser realizadas com imagens. O termo Computação Gráfica é passível de diversas interpretações, estaremos adotando neste texto aquela mais restrita que se aplica apenas à síntese das imagens.

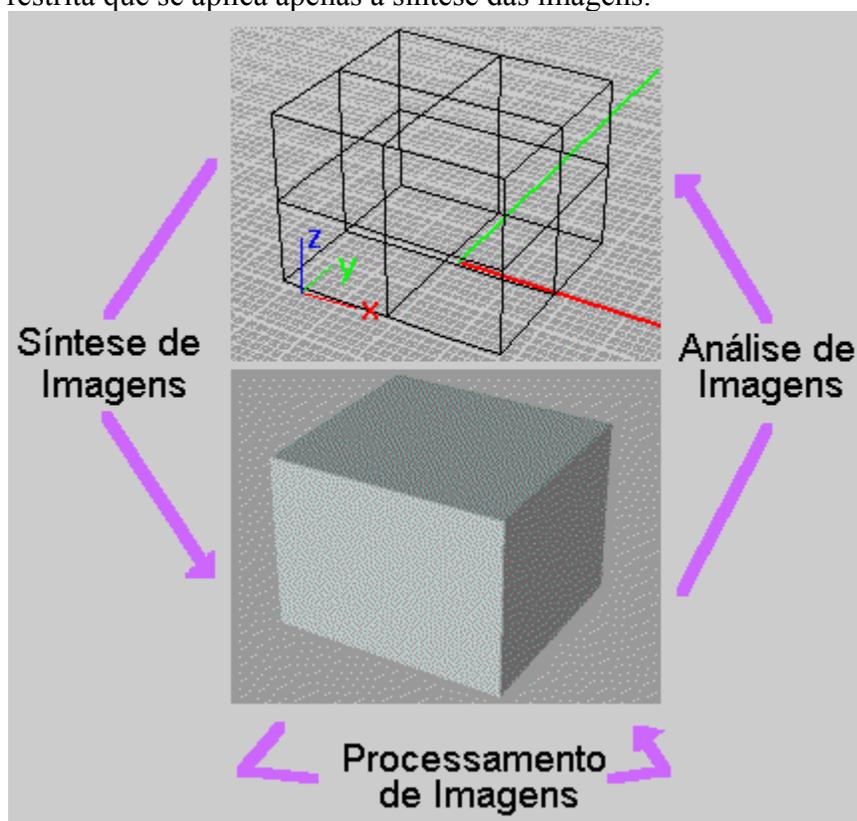


Figura 14: Operações que podem ser realizadas com imagens em um computador.

4.1. Imagens

Transformando Imagens em BITS

Uma imagem é uma projeção de uma realidade que é, por si mesma, infinita (ou quase isto).

A paisagem a partir da qual foi tirado o retrato da figura 5 é real. Se quiséssemos dividi-la em pedacinhos, poderíamos quebrá-la em grãos tão minúsculos até chegar a seus átomos. Os próprios átomos podem ser divididos em partículas menores, e não sabemos até onde isto pode ir. Portanto, a realidade pode ser concebida como algo infinito, ou pelo menos nós ainda não encontramos o seu final.

Quando representamos uma imagem não podemos, contudo, trazer toda esta realidade para dentro do computador pois a memória do computador é algo finito. Para tornar esta operação possível dispomos de duas técnicas básicas:

Representação da Imagem através de um Mapa de Pixels

A imagem da figura 5 precisou ser convertidas em BITS para poder ser incluída neste texto, uma vez que ele foi escrito em um computador.

Como já foi comentado anteriormente, a imagem é composta por pequenos elementos coloridos. Cada elemento deste é chamado de PIXEL e é a menor unidade de representação de uma imagem como esta.

O uso de mapa de pixels é adequado para a representação de fotografias, que são convertidas para este formato através de equipamentos específicos, tal como o Scanner.

Como os pixels são convertidos em BITS? Vamos imaginar que a imagem da torre de um castelo tenha duas cores, como mostra a figura seguinte:

Neste caso poderíamos fazer uma simples associação da seguinte forma:

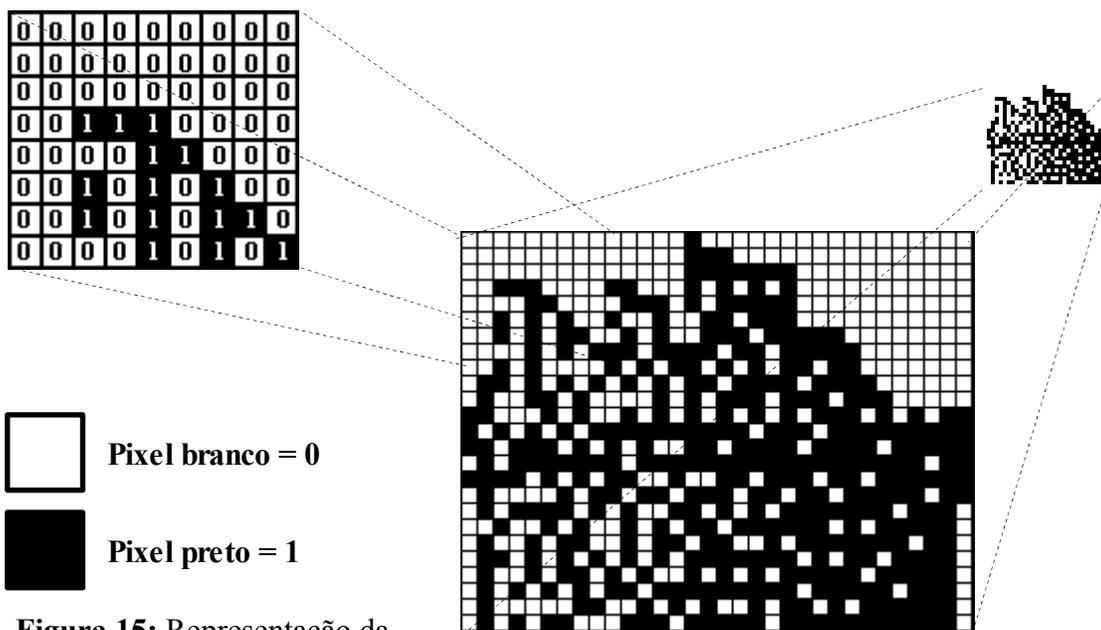
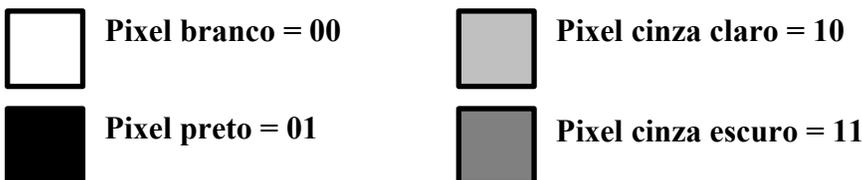


Figura 15: Representação da imagem do castelo em preto e branco

Através deste método os pixels (e consecutivamente a imagem) são transformados em bits e vice-versa.

Se a imagem tiver mais cores - quatro, por exemplo - ela precisará de mais bits combinados para representar cada pixel:



Vetorial x Matricial

Como foi explanado anteriormente existem duas formas de se representar imagens no computador. A primeira, já apresentada, consiste na **Representação da Imagem através de um Mapa de Pixels**; a segunda consiste na representação da imagem sob a forma de modelos geométricos:

Representação Vetorial da Imagem

Quando vemos uma imagem, nosso cérebro não a interpreta como um conjunto de pixels coloridos, como a representação da imagem anterior. Ao invés, vemos cada imagem composta de partes que em conjunto constroem o todo do que observamos.

Se entrarmos em uma sala, poderemos observar cadeiras, uma mesa, o lustre, etc. Se nos detivermos na mesa, constataremos que ela também é composta de partes, como as pernas, a tampa, etc.

Desta forma, quando criamos desenhos, geralmente os produzimos a partir da reconstrução das partes que observamos, sob a forma de contornos e preenchimentos.

O computador também pode tratar a imagem desta forma, ou seja, a partir da representação de cada parte da imagem sob a forma de objetos, este tipo de abordagem recebe o nome de **Representação Vetorial da Imagem**.

Observe a ilustração a seguir, ela foi produzida por um artista a partir de um conjunto de formas e curvas preenchidas com cores e degradê.

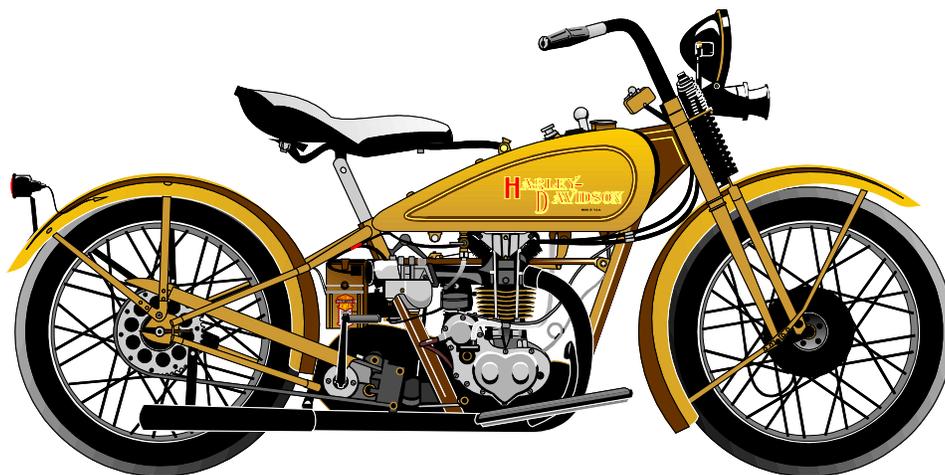


Figura 16: Imagem Vetorial de uma motocicleta

Se a mesma figura for observada sob este outro ângulo, é possível se constatar que a mesma é produzida a partir da combinação de um conjunto de formas geométricas, como círculos, retângulos e segmentos de reta, bem como curvas.

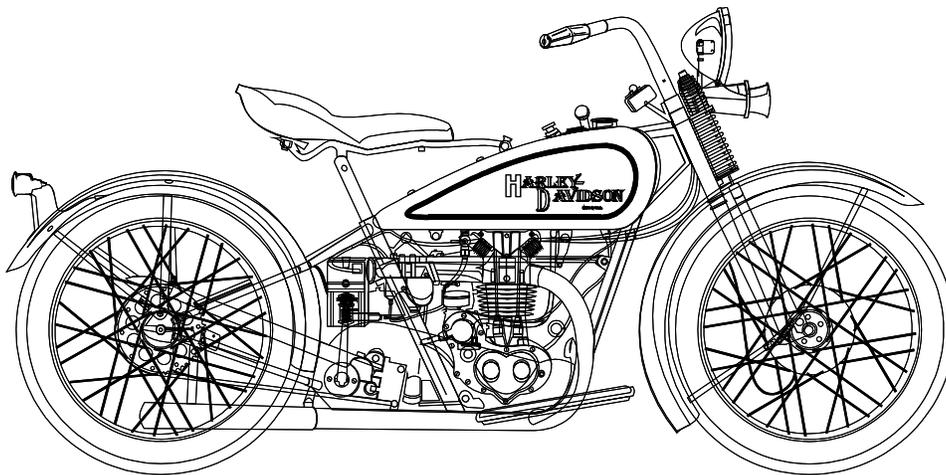


Figura 17: Imagem Vetorial sem os preenchimentos

No diagrama a seguir uma pequena parte da figura foi ampliada afim de poder ser realizada uma análise mais detida.

Dois aspectos importantes podem ser observados:

- a ilustração é composta tanto de formas mais simples como círculos e retângulos, como curvas mais complexas e elaboradas;
- quando a figura é ampliada, ela não perde a sua qualidade como acontecia na outra representação (Mapa de Pixels).

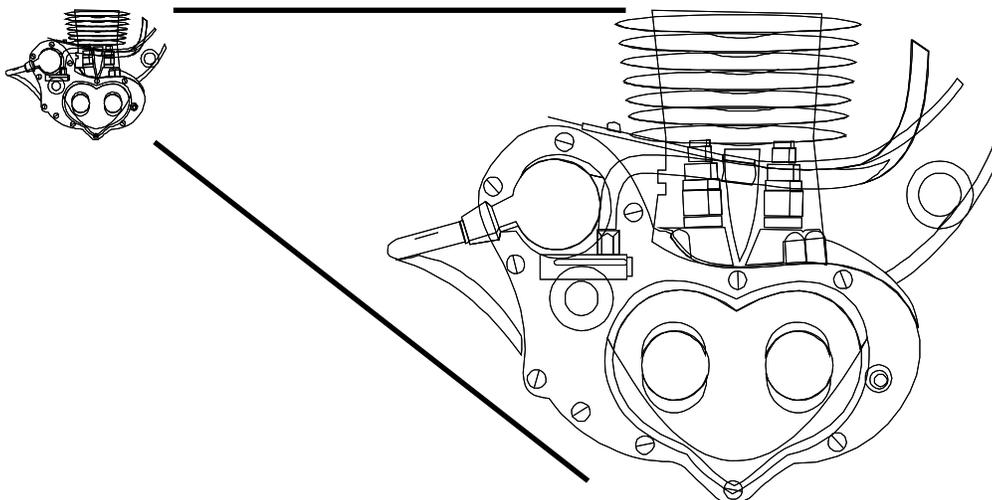
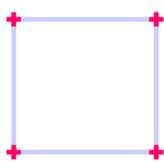
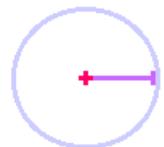


Figura 18: Ampliação de detalhe da figura vetorial.

Este tipo de figura fica armazenado no computador de forma diferente. Ao invés de guardar cada pixel que compõe a imagem, o computador armazena uma descrição geométrica de cada objeto.



um retângulo pode ser representado pelas coordenadas de seus cantos (ou apenas dois cantos opostos);



o círculo pode ser representado pelas coordenadas do seu centro e pela medida do raio;



para o traçado de curvas utiliza-se o modelo de Bézier que faz uso de coordenadas para o traçado de uma curva simples (dois pontos referentes aos extremos e outros dois pontos para aproximação da curva)

Toda a vez que a imagem tiver que ser reproduzida, o computador a reconstrói a partir de sua descrição.

Realidade Virtual

A evolução das técnicas que nos têm levado a interagir com o computador de uma forma cada vez mais natural, através da reprodução cada vez mais fiel do mundo tal como o percebemos, tem conduzido o homem a um universo que existe apenas virtualmente, dando-lhe a sensação de realidade.

É neste contexto que se apresenta a realidade virtual (RV). Apesar de aparentemente contraditória, a expressão traduz um conjunto de esforços no sentido de apresentar o espaço virtual de tal maneira que um indivíduo se sentirá “mergulhado” nele. Não é por acaso que a experiência mais abrangente em termos de RV receba o nome de imersão total.

Existem hoje muitos *softwares* disponíveis para o trabalho com a computação gráfica. Alguns deles não foram projetados especificamente para uso educacional, mas podem ser perfeitamente utilizados neste contexto.

POV-Ray

Ferramenta de síntese de imagens tridimensionais a partir de modelos descritos textualmente. Utiliza a técnica de traçamento de raios (*raytracing*) para produzir imagens foto realísticas.

Endereço: <http://www.povray.org/>

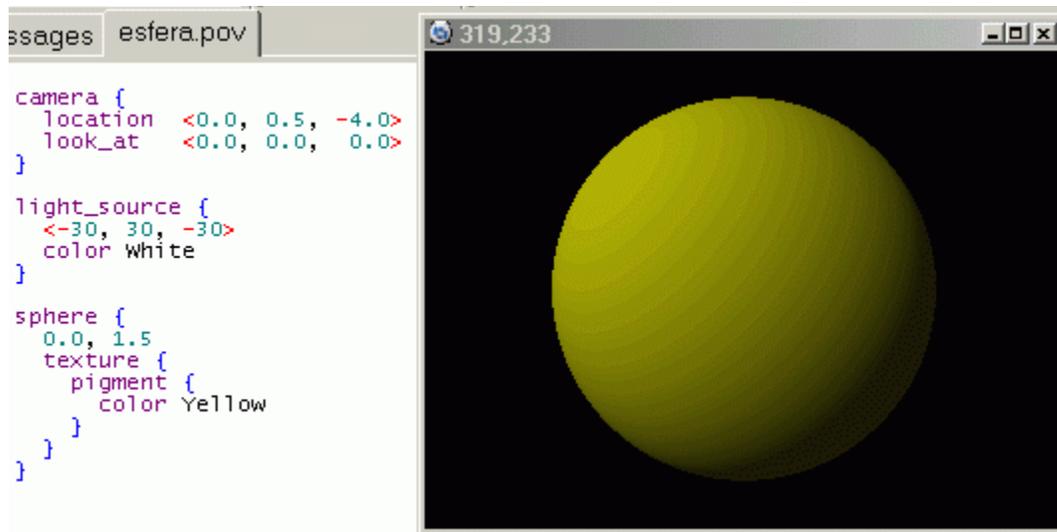


Figura 19: Tela de edição e síntese do POV-Ray.

A figura 19 apresenta uma típica tela de trabalho do POV-Ray, à esquerda é apresentado o modelo e à direita a imagem sintetizada resultante do modelo descrito.

Alice

O Alice, dirigido a usuários leigos em computação gráfica, consiste em um ambiente tridimensional para a construção de animações.

Sua estrutura para organização dos modelos e montagem das animações é toda orientada a objetos. Um modelo tridimensional é considerado um objeto dividido em partes, que também são objetos. O movimento dos objetos é feito através da ativação de módulos de comportamento dos mesmos.

Endereço: <http://www.alice.org/>

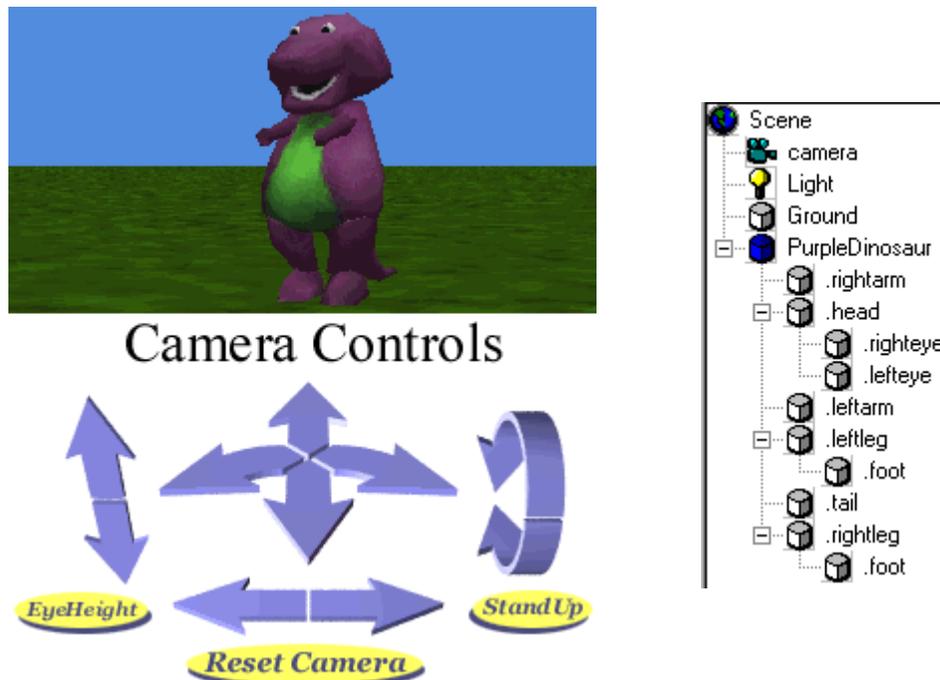


Figura 20: Objeto ‘dinossauro’ apresentado no programa Alice e sua respectiva estrutura.

Na figura 20, observamos um objeto dinossauro construído no programa Alice. A sua direita está apresentada hierarquicamente sua estrutura, composta de objetos: braços (rightarm e leftarm), cabeça (head), pernas (rightleg, leftleg) e cauda (tail). Cada objeto da estrutura pode eventualmente também possuir outros em sua composição, tal como o objeto cabeça (head) que possui: olho direito (righteye) e olho esquerdo (lefteye).

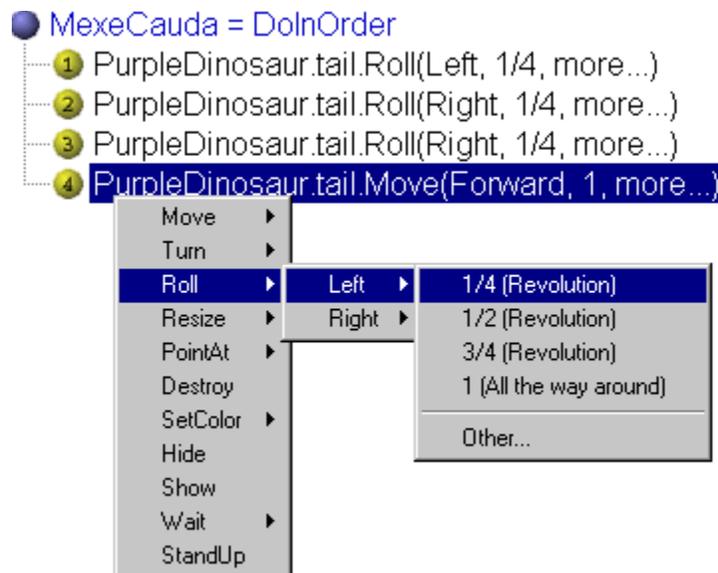


Figura 21: Montagem de um *script* de comportamento no programa Alice.

É possível associarem-se *scripts* de comportamento para os objetos, como pode ser observado na figura 21. Todo o processo é feito visualmente. Os objetos são selecionados e arrastados para a área de código e o sistema solicita, interativamente, as opções de comportamento.

Devido à sua característica tipicamente voltada para usuários leigos em computação gráfica, o Alice constitui uma excelente ferramenta para a construção de animações tridimensionais em atividades de ensino-aprendizagem. Isto é interessante – como veremos adiante em práticas que envolvem simulação ou que se utilizam das múltiplas formas de expressão associadas à tecnologia digital – para enriquecer a linguagem que os alunos utilizam para suas composições.

5. Composição

No final da década de 60, surgiu uma poderosa metodologia de programação de computadores, que se destacou por adotar a metáfora de objetos intercomunicantes para representar os componentes de um programa. A Programação Orientada a Objetos foi o ponto de partida para um paradigma que se expandiu para outras áreas da computação, tal como a análise e o projeto de sistemas, os bancos de dados, etc. Nos últimos anos modelos computacionais baseados em objetos passaram a ser utilizados em outros contextos, inclusive no da educação.

Curiosamente, apesar de seu grande sucesso dar-se originalmente no domínio dos iniciados em computação, o principal propósito de um dos primeiros projetos – que deu verdadeiro impulso ao paradigma da orientação a objetos – era tornar a construção de aplicações em computadores acessível a qualquer usuário leigo.

Em paralelo com a metáfora dos objetos, estrutura-se a dos componentes de *software*, que passaram a dar uma grande ênfase ao planejamento e à construção de modelos computacionais baseados em blocos de código (componentes) facilmente configuráveis e combináveis entre si; de tal modo que os autores do modelo não necessitam de conhecimento específico da forma como os blocos foram construídos. Os componentes representaram um grande avanço dos computadores em direção aos autores de projetos, leigos em computação. Eles se incorporaram ao modelo de objetos, potencializando sua atuação.

Nos últimos anos, podemos observar que os objetos têm gradativamente retomado parte de seu propósito original, pelo menos no que diz respeito à educação. Os objetos, especialmente sob a forma de componentes, inserem-se em atividades de ensino-aprendizagem como blocos fundamentais, sobre os quais alunos e professores constroem modelos que se tornam alvo de estudo.

Neste capítulo, vamos deter-nos no estudo dos objetos e componentes computacionais e seu papel em recentes projetos educacionais

5.1. Objetos – um pouco de história

Objetos são uma expressão cada vez mais natural para os envolvidos com a computação. O conjunto de princípios que deram origem à Orientação a Objetos surgiu no final da década de 60, através da SIMULA [HOL92], uma linguagem inicialmente projetada para realizar simulações por eventos discretos, que posteriormente foi modificada para uma versão genérica. A versão do SIMULA lançada em 1967 (SIMULA 67), apesar de nunca ter sido muito difundida, foi a primeira linguagem a implementar princípios da Programação Orientada a Objetos.

No início da década de 70 Alan Kay, pertencente ao projeto *Dynabook*, liderou os esforços para a concepção de uma linguagem de programação de computadores cujos princípios estivessem em sintonia com o projeto.

*“This ‘Dynabook’ was based on a vision of inexpensive notebook-sized personal computers for both adults and children, with the power to handle all their information-related needs”.*³ [KRE98]

Emerge deste contexto a linguagem *Smalltalk*. Seu projeto tencionava tornar o processo de construção de aplicações acessível a quaisquer usuários, mesmo que eles não tivessem o domínio de linguagens de programação, voltadas para iniciados na computação. Daniel Ingals resume este propósito, enunciando o princípio mestre do projeto que, como ele mesmo afirma, é mais social do que técnico: *“If a system is to serve the creative spirit, it must be entirely comprehensible to a single individual”.*⁴ [ING81]

Apesar de seu propósito claramente voltado a permitir que usuários não especialistas em computação produzissem suas próprias aplicações, o *Smalltalk*, durante muitos anos após o seu lançamento, serviu como base para uma nova configuração das linguagens de programação, que passaram a trabalhar com objetos. A Programação Orientada a Objetos trouxe, deste modo, grandes benefícios à programação de computadores. No entanto, de um modo geral continuou no domínio dos especialistas em computação.

Os princípios sobre os quais os objetos foram concebidos tinham, desde o início, uma proposta bem mais ampla do que simplesmente servir como metodologia de programação aos iniciados em informática. Podemos notar isto no comentário de Daniel Ingals referente ao projeto *Smalltalk*: *“it will save time if we make our computer models compatible with the mind, rather than the other way around”.*⁵ [ING81]

³ Tradução nossa: “Esse ‘*Dynabook*’ foi baseado na visão de computadores pessoais baratos do tamanho de um caderno, tanto para adultos quanto crianças, com a capacidade de lidar com todas as suas respectivas necessidades de informação”.

⁴ Tradução nossa: “Se um sistema serve ao espírito criativo, ele tem que ser inteiramente compreensível para o indivíduo sozinho”.

⁵ Tradução nossa do original: “nós iremos poupar tempo se tornarmos nossos computadores compatíveis com a mente, e não o contrário”.

5.2. Componentes de software

A metáfora dos componentes de *software*, tal como a dos objetos, é proveniente do domínio da programação de computadores. Apesar da grande semelhança existente entre os componentes e os objetos, existem alguns aspectos que os distinguem:

- Assim como os objetos, os componentes de software têm a capacidade de encapsular dentro de um módulo uma coleção de programas interligados entre si, formando um elemento autônomo.
- Por outro lado, os componentes destacam-se por sua facilidade de configuração, quando utilizados na construção de aplicações, e podem ser interligados com outros componentes, sem que o usuário saiba como eles foram implementados. Em geral, os sistemas de construção de aplicações que utilizam componentes dispõem de interfaces visuais para a instanciação, configuração e interligação dos componentes.
- Apesar de não serem obrigatórios, os princípios do paradigma de orientação a objetos podem ser, e é desejável que sejam, associados sem restrições aos componentes. Por este motivo, as linguagens de programação os tem configurado como um subconjunto específico de objetos.

Construir uma aplicação usando componentes é semelhante a um modelo montado com peças de Lego®, que se encaixam para formar um modelo maior. A comunicação entre os componentes dá-se através do envio de mensagens ou chamada de métodos.

Jeremy Roschelle, ao referir-se a componentes de software faz uma analogia:

“Only a very few people can build their own stereo system by wiring together transistors and other components at the circuit level, but many people can assemble a stereo system to meet their needs by attaching cables to their preferred amplifier, receiver, disc player, turntable and speakers”.⁶ [ROS98]

Para exemplificar como acontece esta dinâmica, vamos apresentar um exemplo construído no sistema Casa Mágica, ilustrado na figura 22:

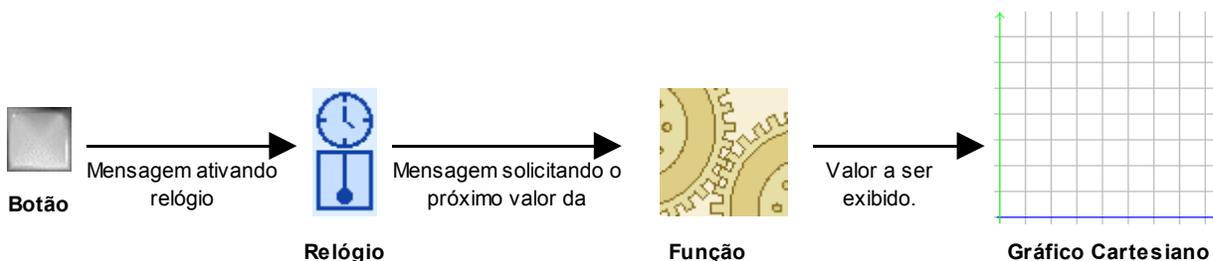


Figura 22: Aplicação que apresenta o gráfico de uma função de segundo grau a partir de uma *framework* de componentes

O modelo da figura 22 é composto de quatro componentes. O botão é responsável por começar o processo enviando uma mensagem de ativação para o componente relógio. O

⁶ Tradução nossa: “Muito poucas pessoas são capazes de construir seu próprio sistema estéreo interligando transistores e outros componentes no nível do circuito, mas muitas pessoas conseguem montar um sistema estéreo adequado a suas necessidades, ligando cabos ao seu amplificador, *receiver*, *disc player*, prato e alto-falantes preferidos”.

relógio inicia então uma seqüência de batimentos, que irá gerar mensagens e propiciar o envio das mesmas para o componente função. A cada mensagem recebida, o componente função calcula um par de valores e envia-os para o componente gráfico cartesiano, que ilustra graficamente os valores recebidos.

Este conjunto de componentes interligados forma uma rede que denominamos *framework*. A partir de bibliotecas de componentes, especializadas em determinado domínio e de ferramentas visuais, que permitem a customização e interligação de componentes, é possível que usuários leigos em computação construam modelos mais complexos, trabalhando em um nível de abstração que desconhece os detalhes de implementação dos componentes.

5.3. Objetos e componentes no ensino

A diferença entre objetos e componentes é bastante sutil, principalmente por que os componentes cada vez mais se apresentam como uma manifestação especial de objetos.

Por considerarmos ambas as abordagens muito ligadas e interdependentes; como a distinção entre as duas, em muitos casos, interessa mais aos profissionais de informática, não nos preocuparemos, na análise apresentada a seguir, em distingui-las.

Objetos e componentes apresentam-se, aqui, sob duas perspectivas. Por um lado, são metáforas que mediam o modo como pensamos algo e como expressamos este pensamento no computador; por outro, constituem uma forma de estruturar as produções computacionais, de tal maneira que, seus elementos de composição de *software* tornam-se fáceis de interligar, trocar e reaproveitar.

Ambas as perspectivas têm sido exploradas em projetos educacionais com o uso de computadores, trazendo-lhes benefícios como:

Do ponto de vista do educador / autor:

- Os componentes por ele produzidos podem ser combinados, ou recombinados, com outros, não limitando sua produção a um programa específico.
- Componentes são facilmente configuráveis e podem ser customizados a necessidades particulares.
- Os dois fatores anteriores, aliados à facilidade de transportar componentes como pacotes autônomos e integráveis, promovem amplamente o seu intercâmbio entre autores diversos, e a Internet facilita e potencializa este intercâmbio.
- Não será necessário partir da estaca zero em cada nova produção, pois os componentes criados podem ser reutilizados, além dos componentes produzidos por terceiros.

Do ponto de vista do aluno:

- Utilizar componentes como blocos de construção promove o desenvolvimento de atividades, onde o aluno atua como agente ativo na construção do conhecimento.

- intercâmbio facilita a colaboração entre alunos dentro de um mesmo grupo ou entre grupos. Aqui a Internet também cumpre um papel importante de mediadora, para uma comunicação que pode estender-se a um âmbito global. Além disto, ela cumpre o papel de repositório de componentes.

Projetos educacionais baseados em objetos e componentes

Nos últimos anos, tem crescido muito o número de projetos que adotam objetos e componentes em atividades de ensino / aprendizagem utilizando computadores e redes.

A seguir, apresentamos alguns desses projetos, suas diferentes abordagens e contribuições.

MathWorlds
O MathWorlds, <i>software</i> integrante do projeto SimCalc, “[...] provê uma coleção de componentes de <i>software</i> , incluindo um conjunto de mundos de animação e uma variedade de gráficos. Os atores nos mundos (como um palhaço, ou um pato) movem-se de acordo com funções matemáticas. Os gráficos mostram essas funções matemáticas e permitem aos alunos editarem diretamente as funções”. [ROS98.2]
Endereço: http://tango.mth.umassd.edu/

A figura 23 apresenta uma típica aplicação do MathWorlds. Dentro de um cenário são dispostos atores, cujo movimento está descrito por curvas de ‘distância x tempo’ e ‘velocidade x tempo’, apresentadas no gráfico logo abaixo desses atores.

É possível, através de uma intervenção direta em qualquer um dos dois gráficos, modificar o movimento dos atores. Modificações feitas com os atores também repercutem no gráfico.

Como resultado, o aluno recebe subsídios que o permitem verificar, de modo bastante interativo, o papel do gráfico na descrição do movimento dos atores. O *software* fornece recursos para a criação de outros cenários, diferentes atores e movimentos a serem estudados.

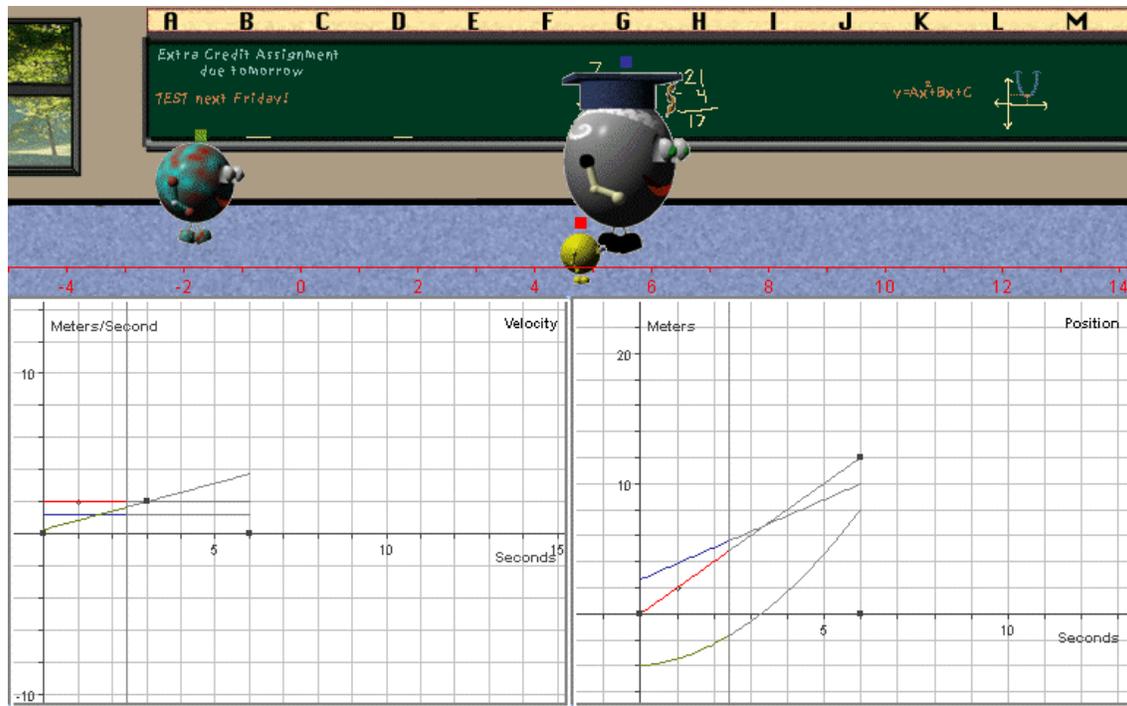


Figura 23: Animação sendo executada no MathWorlds.

Casa Mágica

O sistema Casa Mágica [SAN98] constitui-se em um ambiente para construção de aplicações educacionais que combina os recursos de uma ferramenta de autoria com recursos que permitem a construção e exploração de modelos de estudo.

Casa Mágica oferece suporte para a elaboração de *frameworks* baseadas em objetos e componentes e é capaz de representá-los em XML [SAN00]. Assim, as aplicações podem ser distribuídas pela Internet e executadas diretamente por um módulo *runtime* do sistema ou apresentadas em um navegador *Web*.

Referências Bibliográficas

- [AND00] **Anderson, Thor & Wason, Tom.** IMS Learning Resource Meta-data Information Model – Version 1.1 - Final Specification. IMS Global Learning Consortium, Inc., 5 June 2000, [Online]
<http://www.imsproject.com/metadata/mdinfov1p1.html>
- [BAN] **Banon, Gerald Jean Francis.**
- [BRA98] **Bray, Tim & Hollander, Dave & Layman, Andrew.** Name Spaces in XML - World Wide Web Consortium Note 19-January-1998, W3C - World Wide Web Consortium, [Online] <http://www.w3.org/TR/1998/NOTE-xml-names-0119>
- [BRE91] **Breton, Philippe.** História da Informática. São Paulo: Editora Universidade Estadual Paulista, 1991.
- [BRI99] **Brickley, Dan & Guha, R. V.** Resource Description Framework (RDF) Schema Specification - W3C Proposed Recommendation 03 March 1999, W3C - World Wide Web Consortium, [Online] <http://www.w3.org/TR/PR-rdf-schema/>
- [DC97] **Dublin Core.** The Dublin Core: A Simple Content Description Model for Electronic Resources, February 1997, [Online]
<http://purl.oclc.org/dc/index.htm>.
- [DIM98] **Dimenstein, Gilberto.** Aprendiz do futuro – Cidadania hoje e amanhã. São Paulo: Editora Ática, 1998.
- [FAR99] **Farance, Frank.** LOM Specification - Learning Objects Metadata, Proposed Draft 4. IEEE Learning Technology Standards Committee (LTSC), September, 1999, [Online] <http://edutool.com/lom/lom-04.pdf>
Este é o documento oficial onde está a especificação do LOM, elaborada pelo Learning Technology Standards Committee (LTSC) do IEEE.
- [HOL92] **Holmevik, Jan Rune.** Compiling SIMULA: a historical study of technological genesis. IEEE Annals of the History of Computing, 16 (4), p. 25-37, 1994. [Online] <http://www.cis.um.edu.mt/~jskl/simula.html>
- [IBM] **IBM - International Business Machines Corporation.** IBM Smalltalk Tutorial [Online] <http://www.wi2.uni-erlangen.de/sw/smalltalk/>
- [ING81] **Ingalls, Daniel H. H.** Design Principles Behind SmallTalk. BYTE Magazine, August 1981 [Online]
http://users.ipa.net/%7edwighth/smalltalk/byte_aug81/design_principles_behind_smalltalk.html

Daniel H. H. Ingalls chefiou a implementação das primeiras versões do Smalltalk. Neste artigo ele apresenta os fundamentos desta linguagem que são a primeira expressão concreta da Programação Orientada a Objetos. Aqui são lançadas as bases que nortearão as futuras linguagens e modelos.

- [ISO86] **ISO - International Organization for Standardization.** ISO 8879:1986(E), Information processing - Text and Office Systems - Standard Generalized Markup Language (SGML), First edition - 1986-10-15. Geneva: International Organization for Standardization, 1986.
- [JEN98] **Jennings, Nicholas R., Sycara, Katia and Wooldridge, Michael.** A Roadmap of Agent Research and Development. In: *Autonomous Agents and Multi-Agent Systems Journal*. Boston: Kluwer Academic Publishers, 1998, Volume 1, Issue 1, pages 7-38 [Online]
http://www.ri.cmu.edu/pub_files/pub1/jennings_nicholas_1998_1/jennings_nicholas_1998_1.pdf
- [KON] **Koning-Bastiaan, Martin.** Connected and Scalable: A revolutionary Structure for Online Communities. *EOE – Educational Object Economy* [Online]
<http://www.eoe.org/FMPro?-db=Objects.fp3&-token=libraryPapers&-format=/library/paperdetail.htm&-recid=35433&-lay=all&-Find>
- [KRE98] **Kreutzer, Wolfgang.** Basic Aspects of Squeak and the Smalltalk-80 Programming Language. Junho, 1998 [Online]
<http://kaka.cosc.canterbury.ac.nz/~wolfgang/cosc205/smalltalk1.html>
- [KUH87] **Kuhn, Thomas.** "Posfácio - 1969", in *A estrutura das revoluções científicas*. São Paulo: Perspectiva, 1987.
- [LEV98] **Lévy, Pierre.** Educação e Cybercultura – A nova relação com o saber. 1998 [Online] <http://www.portoweb.com.br/PierreLevy/educaecyber.html>
- [LTS00] **IEEE Learning Technology Standards Committee (LTSC).** Draft Standard for Learning Object Metadata. Institute of Electrical and Electronics Engineers, Inc., 5 February 2000, [Online] http://ltsc.ieee.org/doc/wg12/LOM_WD4.PDF
- [FER89] **Ferreira, Aurélio B. H.** *Minidicionário da Língua Portuguesa*. Rio de Janeiro, Editora Nova Fronteira, 1989.
- [GIR99] **Giraffa, Lucia Maria Martins.** Uma arquitetura de tutor utilizando estados mentais. Porto Alegre: CPGCC/UFRGS, 1999. (Tese de Doutorado) [Online]
<http://www.inf.pucrs.br/~giraffa/ia/tese.zip>
- [KNU73] **Knuth, Donald E.** *Fundamental Algorithms*. USA: Addison-Wesley Publishing Company, 1973.
- [LAK80] **Lakoff, George and Johnson, Mark.** *Metaphors We Live By*. Chicago: University of Chicago Press, 1980.

- [LAS99] **Lassila, Ora & Swick, Ralph R.** Resource Description Framework (RDF) Model and Syntax Specification - W3C Recommendation 22 February 1999, W3C - World Wide Web Consortium, [Online] <http://www.w3.org/TR/REC-rdf-syntax/>
- [LEV96] **Levy, Pierre.** O que é o virtual? São Paulo: Editora 34, 1996.
- [MAN97] **Manola, Frank** (ed.). NICTS Technical Committee H7 Object Model Features Matrix, X3H7-93-007v12b, May 25, 1997 [Online] <http://www.objs.com/x3h7/h7home.htm>.
- [MAN98] **Manola, Frank.** Towards a Web Object Model. Object Services and Consulting, Inc. (OBS), 10 February 1998 [Online] <http://www.objs.com/OSA/wom.htm>
- [MAN99] **Manola, Frank.** Technologies for a Web Object Model. IEEE Internet Computing, January - February 1999.
- [MEY97] **Meyer, Bertrand.** Object-Oriented Software Construction – Second Edition. USA, Prentice-Hall, Inc., 1997.
Realiza uma análise bastante aprofundada da construção de software orientado a objetos. Neste livro são analisados as características de linguagens orientadas a objetos, sua estrutura e conceitos.
- [MIL98] **Miller, Eric.** An Introduction to the Resource Description Framework, May 1998, [OnLine] <http://www.dlib.org/dlib/may98/miller/05miller.html>
- [MIN87] **Minsky, Marvin.** Society of Mind. New York: Simon & Schuster, 1987.
- [PAP80] **Papert, Seymour.** Mindstorms: children, computers and powerful ideas. New York: Basic Books, 1980. No Brasil: LOGO: Computadores e Educação. São Paulo: Brasiliense, 1985.
- [REP96] **Repenning, Alexander & Ambach, James.** Tactile Programming: A Unified Manipulation Paradigm Supporting Program Comprehension, Composition and Sharing. Proceedings of the 1996 IEEE Symposium of Visual Languages, Boulder, CO, Computer Society, 1996, pp. 102-109 [OnLine] <http://www.cs.colorado.edu/~l3d/systems/agentsheets/Documentation/VL-96-Paper.pdf>
- [REP99] **Repenning, Alexander & Ioannidou, Andri & Phillips, Jonathan.** Collaborative Use & Design of Interactive Simulations. Proceedings of Computer Supported Collaborative Learning Conference at Stanford (CSCL'99), 1999 [Online] <http://www.escot.org/escot/External/UseDesign.pdf>

- [ROS98] **Roschelle, Jeremy & Digiano, Chris & Pea, Roy & Kaput, Jim.** Educational Software Components of Tomorrow (ESCOT). [SRI International](http://www.escot.org/escot/External/MSET_ESCOT.html), 1998, [Online] http://www.escot.org/escot/External/MSET_ESCOT.html
- [RSC98] **Roschelle, Jeremy & Kaput, Jim & Stroup, W.** SimCalc: Accelerating students' engagement with the mathematics of change. In M. Jacobson & R. Kozma (Eds.), Educational technology and mathematics and science for the 21st century. Hillsdale, NJ: Erlbaum, 1998 [Online] <http://tango.mth.umassd.edu/website/publications/MathofChange.pdf>.
- [SAN98] **Santanchè, André.** Sistema para Construção de Aplicações Educacionais. IV Congresso Ibero-Americano de Informática na Educação, 1998, [Online] <http://www.brasil.terravista.pt/claridade/1622/publicado/ArtigoRIBIE1998.pdf>
- [SAN00] **Santanchè, André & Teixeira, Cesar.** Construindo e explorando o conhecimento através de Componentes Educacionais embutidos em hiperdocumentos. VI Workshop de Informática na Escola – XX Congresso da SBC, 2000, [Online] <http://www.brasil.terravista.pt/claridade/1622/publicado/WIE2000.pdf>
- [SMI98] **Smith, Sophia.** Artificial Life and Genetic Algorithms. Brunel University, Artificial Intelligence Site, Abril de 1998 [Online] <http://www.brunel.ac.uk:8080/depts/AI/alife/alife-main.html>
- [TAK00] **Takahashi, Tadao** (org.). Sociedade da informação no Brasil: livro verde. Brasília: Ministério da Ciência e Tecnologia, 2000 [online] http://www.socinfo.org.br/livro_verde/
- [TEO00] Teodoro, Vitor Duarte. Modellus – Interactive Modelling with Mathematics – version 2.01. Faculty of Sciences and Technology – New University of Lisbon, 2000.
- [TRA96] **Travers, Michael D.** Programming with Agents: New metaphors for thinking about computation. Massachusetts Institute of Technology, June 1996 [Online] <http://lcs.www.media.mit.edu/people/mt/thesis/mt-thesis.html>
- [W3C99] **W3C XML Working Group.** Mathematical Markup Language (MathML™) 1.01 Specification - W3C Recommendation, revision of 7 July 1999, W3C - World Wide Web Consortium, [Online] <http://www.w3.org/TR/REC-MathML/>
- [W3S98] **W3C XML Working Group.** Synchronized Multimedia Integration Language (SMIL) 1.0 Specification - W3C Recommendation 15-June-1998, W3C - World Wide Web Consortium, [Online] <http://www.w3.org/TR/REC-smil/>
- [W3X98] **W3C XML Working Group.** Extensible Markup Language (XML) 1.0 - W3C Recommendation 10-February-1998, W3C - World Wide Web Consortium, [Online] <http://www.w3.org/TR/1998/REC-xml-19980210>