

## ProNotas

### Robô controlado por Mac

A rapaziada do DevHQ (Macintosh Developer Headquarters) criou o **DevHQ Drone**, um robô meio-carro-meio-tanque para todo tipo de terreno, que pode ser programado e controlado totalmente a partir de um PowerBook. Classificado como "rápido, poderoso, legal e barato", esse "brinquedo" tem recursos de vídeo (é possível ver o mundo pela perspectiva do Drone) e alcance de controle de 15 metros, através de um cabo que se conecta à porta ADB. O custo estimado pela turma do DevHQ é de US\$ 500. O robô está ainda em fase de testes, tanto que tem nome de Drone Alpha. O Drone Beta ainda está em desenvolvimento e o objetivo é oferecer controle sem fio, através da porta infravermelha.

**DevHQ Drone:** [www.devhq.com/drone.shtml](http://www.devhq.com/drone.shtml)

### AutoStart na parada: CD-ROM do LinuxPPC r5 vem com o verme

O site de notícias Macintosh relatou que os CDs do LinuxPPC r5 estão sendo distribuídos com o famoso vírus **AutoStart 9805**.

O AutoStart foi encontrado pela primeira vez em maio de 98, em Hong Kong, e foi batizado com esse nome porque se utiliza do recurso AutoStart do QuickTime (ativado pelo painel de controle QuickTime Settings), permitindo que os CD-ROMs acionem o verme após a sua inserção.

Quando a mídia infectada é acionada, o QuickTime roda automaticamente um programa invisível chamado DB, que se autocopia para a pasta de extensões. O verme restata o computador e se recarrega dentro da memória. Ele começa, então, a destruir os arquivos com mais de 2 MB que estiverem no disco. A Symantec e a NAI (fabricante do Virex) têm vacinas atualizadas que previnem que o seu Mac seja detonado pelo AutoStart.

**Macintosh:** [www.macintouch.com](http://www.macintouch.com)

**Linux PPCr5:** [www.linuxppc.com](http://www.linuxppc.com)

**Symantec:** [www.symantec.com](http://www.symantec.com)

**Network Associates:** [www.nai.com.br](http://www.nai.com.br)

### iTools é alternativa contra bug do Mac OS X Server

Enquanto não sai o patch para o bug do Mac OS X Server, as empresas que oferecem programas alternativos para servir páginas Web no sistema da Apple aproveitam para divulgar suas soluções. Uma delas é o **iTools**, da Tenon, servidor Web que está substituindo o WebTen 4.0.

O conceito por trás do produto é que, em vez de ser completamente separado do Apache incluído no Mac OS X Server (como o WebTen 4.0), ele simplesmente amplia e melhora alguns recursos do Apache. A Tenon afirma que, logo que soube do bug, realizou testes num PowerBook G3, gerando de 32 a 64 processos simultâneos através de CGIs, e não conseguiu fazer com que o sistema travasse. ▶

# É hora de morphar!

Veja, passo-a-passo, como combinar os efeitos de morph do Elastic Reality com o After Effects

por João Velho

Durante muito tempo, o **morph** foi considerado o mais especial dos efeitos especiais. Popularizado há dez anos por um clip de Michael Jackson (*Black or White*), e depois por dezenas de comerciais e filmes de longa-metragem, ainda hoje, mesmo sem o frescor da novidade, essa técnica mantém uma mística própria.

Tanto na plataforma Mac quanto na NT ou SGI, o **Elastic Reality**, da Avid, é tido como o campeão para efeitos de morph. Atualmente na versão 3.0, o Elastic Reality traz ainda outros recursos importantes. Com ele também é possível fazer efeitos de *warp* (distorção de imagem), criação de máscaras em movimento, composição de imagem, animação estilo *in-between frame*, correção de cor e efeitos tipo DVE (*page peels, twirls* etc.).

Muitos imaginam ser complicadíssimo fazer um efeito de morph. No entanto, para o usuário acostumado com programas gráficos e de animação, vê-se logo que essa função do Elastic Reality não é nenhum bicho de sete cabeças. Um pouco de paciência e meticulosidade são suficientes para resolver a maior parte das tarefas.

O tutorial de morph desenvolvido para este artigo comprova isso. Nosso objetivo aqui foi transformar uma série de fotos 3/4 de várias fases da idade de uma pessoa em uma seqüência de animação contínua (eu mesmo virei a primeira cobaia). Vamos ver como se faz essa experiência do começo ao fim.

### Preparando as imagens

O Elastic Reality importa imagens paradas (*stills*) nos formatos PICT e TIFF e imagens em movimento nos formatos QuickTime e OMF. Para o nosso trabalho, o primeiro passo é escanear e salvar as imagens originais em um dos formatos aceitos. Em seguida, de modo a facilitar a correspondência entre as figuras, os vários rostos das 3x4 devem ser redimensionados para tamanhos próximos em imagens com a resolução de tela de 640 x 480 pixels a 72 dpi. A resolução de 640 x 480

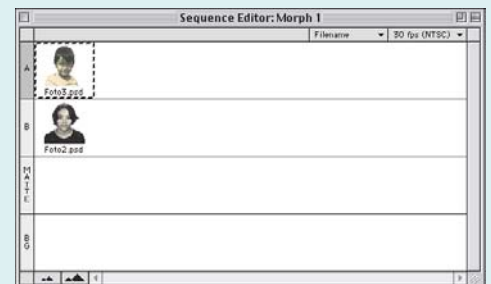
funciona mais como a moldura para a animação no padrão do vídeo em tela cheia.

O passo seguinte é recortar cada uma das imagens a serem utilizadas. O recorte dos rostos serve a dois propósitos: para poder colocar outro fundo sob as fotos e para facilitar o processo de animação, tornando desnecessária a criação de *barrier shapes* para defender o fundo das distorções aplicadas sobre a figura principal.

O recorte das imagens é feito no Photoshop, deixando as figuras em uma camada (*layer*) sobre fundo transparente. Para esse trabalho também pode ser o usado o KnockOut, da Ultimatte, que separa figura e fundo automaticamente, preservando os contornos de cabeças com fios de cabelo soltos.

### Importando as imagens e salvando o projeto

Logo que abrimos o Elastic Reality, notamos a janela do editor de seqüência, com duas trilhas de imagem (A e B) imprescindíveis para o efeito de morph. Basta arrastar as imagens iniciais, uma para cada trilha, e pronto: já dá para começar a trabalhar. Na trilha A deve estar sempre a imagem que inicia a seqüência de animação.



A janela de edição propriamente dita aparece quando clicamos duas vezes sobre uma das imagens das trilhas A e B, ou pelo comando Window ▶ Edit. As imagens aparecem sobre fundo branco porque foram previamente recortadas.

Em seguida, é preciso salvar o projeto, através do tradicional File ▶ Save As. Vamos dar o nome de Morph 1. Por padrão, o Elastic Reality cria pro- ▶

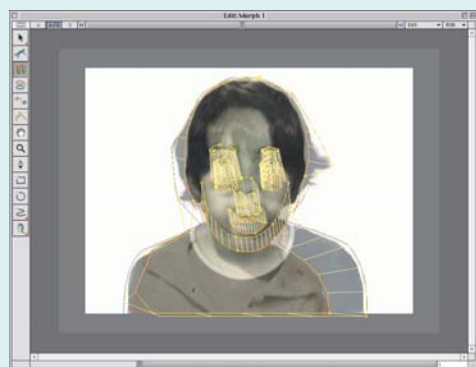
# É hora de morphar!

continuação

jetos de animação com duração de 1 segundo, o que para o nosso objetivo é bem adequado.

## Criando os shapes

Para visualizar bem a animação, estendemos a janela de edição até mostrar as imagens por inteiro, mais uma sobra para a margem de trabalho, a área cinza. Todo o processo do morph se baseia na criação de *shapes* (construções geométricas que seguem os contornos de áreas da imagem). A associação dos shapes de uma figura com os shapes correspondentes da outra figura é que proporciona o morph. O que o morph faz, na verdade, é distorcer (*warp*) as imagens a partir dos shapes correspondentes. Durante a transição, os shapes da primeira imagem tendem para a forma dos shapes correspondentes da segunda imagem e vice-versa. Enquanto isso ocorre, o programa vai fazendo uma fusão simultânea de imagens de uma figura para a outra.



No nosso caso, vamos criar oito shapes para cada figura. Um para o contorno do corpo e da cabeça, incluindo o cabelo, um par para as sobrancelhas, um par para os olhos e outros para nariz, boca e queixo/maxilar. Primeiro criamos os shapes para a figura da trilha A.

A imagens podem ser visualizadas na janela de edição clicando nos botões A, A/B e B, que ficam no canto superior esquerdo. A opção A/B mostra os shapes associados às duas imagens, e com a ajuda do slider ao lado, possibilita checar a correspondência das imagens através da fusão entre as duas.

Com a imagem A selecionada, acionamos a ferramenta Pen na barra de ferramentas, à esquerda da janela de edição. A ferramenta nos permite criar um *path*, tal como no Photoshop, seguindo o contorno que desejarmos. Começamos pelo contorno externo e em seguida passamos para os internos. O ER também possui a ferramenta de traço assistido para essa tarefa.

Para criar shapes fechados, como de hábito, basta criar o último ponto do path sobre o primeiro. Os shapes abertos são obtidos clicando com  $\square$  no último ponto do path ou clicando em uma outra ferramenta. O importante é que os paths nunca podem se cruzar, senão a brincadeira não acontece.

## Editando e juntando os shapes

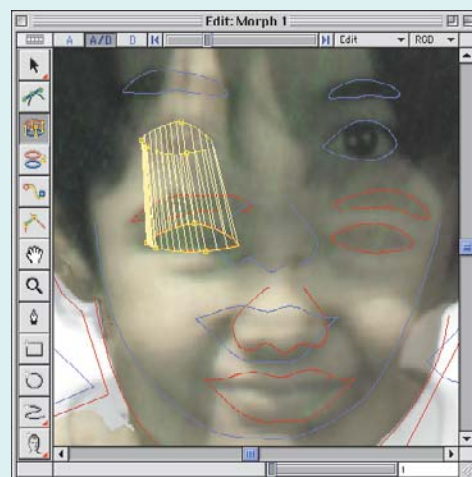
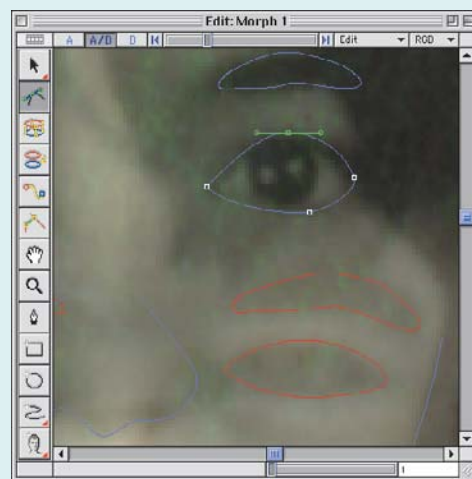
Quando a edição dos shapes é necessária, a ferramenta Reshape é quase sempre a mais indicada, seja para mudar os pontos de posição ou alterar tangentes. Fique atento ao número de pontos para que ele não se exceda.

Depois de criar todos os 8 shapes da figura da trilha A, acionamos o comando Edit ▶ Select All para selecionar todos os shapes associados à primeira imagem. A seguir acionamos o comando Copy e vamos para imagem B clicando no botão correspondente no alto da janela de edição.

Sobre a imagem B acionamos o comando Paste. Com isso, garantimos que cada variação do mesmo shape nas imagens A e B tenha o mesmo número de pontos. A partir daí, reposicionamos os shapes nas áreas mais adequadas da imagem B com a ferramenta Transform. Com ela, podemos mover um shape e modificar sua escala e rotação.

Uma pequena edição com a ferramenta Reshape é inevitável. O cuidado nessa fase deve ser no sentido de manter uma certa correspondência entre as posições dos pontos dos shapes da imagem A com os da imagem B. Feito isso, chegou a hora de juntar os shapes das duas imagens.

Clicando no botão A/B e mantendo o slider na



sua região central, visualizamos as duas imagens, uma sobre a outra, e os shapes sobrepostos. Os shapes da figura A estão em vermelho e os da B, em azul.

O próximo passo é juntar os shapes com a ajuda da ferramenta Join. Isso é feito clicando sobre um dos shapes em vermelho e arrastando o cursor sobre o shape correspondente em azul. Quando a correspondência se estabelece, vemos formar-se uma malha de retas finas em amarelo, associando os pontos dos dois shapes numa determinada densidade.

É só repetir o mesmo processo com todos os shapes e o serviço fica completo. Para ter certeza de que não há shapes cruzando em nenhum ponto, usamos o comando Render ▶ Verify.

Para ver como ficou o morph, é só fazer um preview em qualquer ponto da animação. Escolha o comando Render ▶ Preview, com o slider da parte de baixo da janela de edição posicionado sobre o frame 15. Se quiser avaliar o morph em outros estágios, é só mudar o número do frame no slider e usar o atalho de teclado  $\square$ P.

## Fazendo o output e o render final

Chegou a hora de customizar o filme de animação gerado pelo Morph. Vamos acionar o

O morph, mesmo não sendo mais novidade, conserva uma mística própria

No entanto, a empresa diz que podem ter havido vários motivos para que o bug não acontecesse: o PowerBook pode ser uma das máquinas não afetadas pelo problema ou as CGIs utilizadas podem ser menos "perigosas". Enfim, enquanto o patch da Apple não vem, isso pode ser uma saída para os usuários do sistema operacional.

**ITools:** [www.tenon.com/products/itools](http://www.tenon.com/products/itools)

## LinuxPPC tem nova versão

Os macmaniáticos que estiverem a fim de experimentar o sistema operacional Linux já podem usufruir da versão 5 do **LinuxPPC**. O sistema, uma portagem da distribuição RedHat para o chip PowerPC, funciona em quaisquer Power Macs a partir do Performa 6360.

As novidades não são poucas, a começar pelo novo instalador, projetado para uma instalação mais simples (o tradicional instalador RedHat ainda permanece). O sistema vem com a interface gráfica KDE, para os que querem experimentar o novo sistema sem ficar restritos à linha de comando.

**LinuxPPC:** [www.linuxppc.com](http://www.linuxppc.com)

## Tools Plus Pro chega à versão 5

Nem só de CodeWarrior vive o desenvolvedor de Mac. O **Tools Plus Pro**, software de programação para Mac, está chegando à versão 5.0. Desenvolvido pela Water's Edge Software, ele é descrito como "um esqueleto de programa onde você pode colocar suas próprias peças e montar seu aplicativo". Ele suporta várias linguagens, como C/C++ e Pascal, e gera programas tanto para PowerPC quanto para 68K. Está disponível por US\$ 249 (versão completa), mas possui as versões acadêmica e light, mais baratas.

**Tools Plus:** [www.interlog.com](http://www.interlog.com)

## FileMaker pronto para o Grande Bug

Usuários do **FileMaker Pro 4.1** podem fazer o update para a versão 4.1v2, que contém proteção contra o bug do milênio e muito mais.

Agora, o programa de banco de dados tem validação de tipos de dados. Ou seja, todos os campos de data são mostrados por completo, com quatro dígitos no campo do ano. Seguindo esse padrão, ao digitar apenas a data abreviada (com apenas dois dígitos), o programa automaticamente a transforma em uma data completa, com quatro dígitos.

O update também traz outras correções de problemas relacionados ao bug do milênio, como importação de datas depois do ano 2000. O update está disponível gratuitamente apenas para usuários da versão 4.1.

**FileMaker:** [www.filemaker.com](http://www.filemaker.com)

## Sai banco de dados para Mac OS X

A dinamarquesa Frontline Software está lançando o **FrontBase 1.2**, um poderoso banco de dados para o Mac OS X Server que, segundo a empresa, é 100% à prova de panes. O programa oferece compatibilidade com o padrão SQL 92, tem suporte a Unicode e é capaz de gerenciar bancos de dados gigantescos com terabytes de informação. A versão full para servidores de Internet custa US\$ 699, e para qualquer outra aplicação, US\$ 2.999.

**FrontBase:** [www.frontline-software.dk](http://www.frontline-software.dk)

comando Render ► Output Options. Uma janela se abre. Praticamente não precisa alterar nada no default do programa. Apenas clique sobre o botão Parameters e selecione o formato de compressão Animation com Best setting e Millions of colors+. Desligue a opção Limit data rate to, clique OK nas janelas e deus.

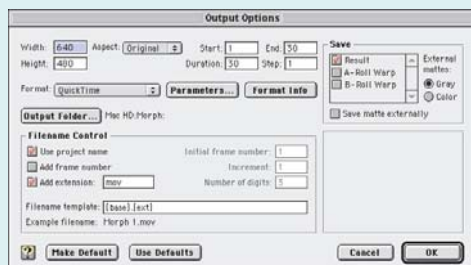
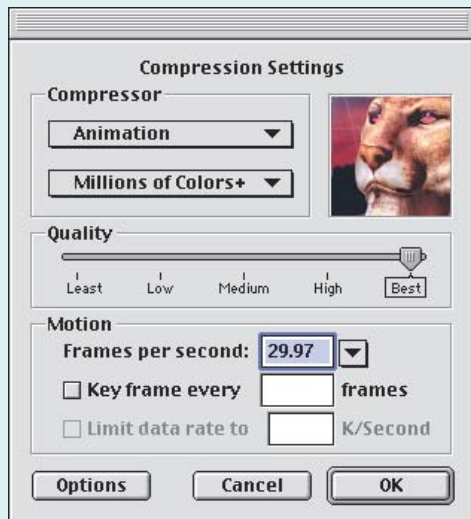
Não há necessidade de mais nenhum ajuste. Agora basta acionar o comando Render ► Render e, em poucos minutos, estará pronto o filme Morph 1.mov com a animação. O filme é salvo no mesmo folder do projeto.

Terminado o render do primeiro movie, podemos passar para o segundo. Cada morph exige um novo projeto e um novo render. Para haver continuidade entre as seqüências de morph, precisamos começar a seqüência seguinte pela imagem com a qual termina a seqüência anterior. Até os shapes podem ser importados e exportados de um filme para o outro.

Este exercício obviamente é superficial e não se utiliza de muitos ajustes disponíveis que podem sofisticar o processo. O objetivo aqui se resume a mostrar os procedimentos básicos e a facilidade de uso do software para obter um efeito simples de morph.

## Composição final no After Effects

No After Effects, vamos completar o filme, aplicando os fundos e juntando os filmes de morph feitos no ER com as várias fases de transformação da pessoa, desde a mais nova até a atual.



O fundo aplicado no nosso caso evoca um "túnel do tempo". A imagem roda para dar a sensação de infinito enquanto a imagem do morph progride. Se for o caso, podemos importar as imagens em PICT ou TIFF para aumentar a duração das figuras originais da animação. No final, fazemos o render definitivo no After Effects com o formato de compressão do sistema de vídeo em que será dada a saída do material: Media 100, Avid, Targa etc.

Nosso filme com a história de alguém contada por fotos 3x4 está pronto, e a conclusão é de que a vida passa rápido, muito rápido. **M**

## JOÃO VELHO

[jvelho@cyberhome.com.br](mailto:jvelho@cyberhome.com.br)

É sócio da Digjworks, empresa de criação de projetos de animação, vinhetas e pós-produção de vídeo digital.





# Corte as sete cabeças do monstro

## ou: Finalmente faça algo útil

Curso de AppleScript, parte 4

por **Maurício L. Sadicoff**

Até agora falamos de AppleScript pra cá, AppleScript pra lá, mas não mostramos ainda as grandes maravilhas do AppleScript. Você já sabe como pode fazer as coisas parecerem um programa, mas para que as coisas fiquem melhores ainda, temos que ter um propósito na criação dos scripts.

Para isso, temos que aprender a escrever scripts como gente grande. Quando somos crianças e começamos a escrever, temos problemas e invariavelmente soletramos as coisas de forma errada, quase sempre impedindo que qualquer outra pessoa compreenda a metade do que escrevemos. Para escrever certo, aprendemos mais tarde a consultar o dicionário, que contém as palavras escritas da maneira certa. AppleScript é igualzinho. Até agora, você já sabe “falar” algumas palavras do idioma, mas pra saber falar, ler e escrever como gente grande, tem que aprender a consultar dicionários. O AppleScript não é brilhante só porque permite que alguém esperto faça scripts. É brilhante porque permite que **qualquer um** escreva scripts, se souber ler os dicionários.

Para citar um exemplo, vou usar a e-cartinha de Ricardo Nóbrega da Silva, que escreveu muito educadamente, pedindo por favor e obrigado pra eu explicar como ele pode usar o “*open applescript dictionary para netscape communicator dictionary*”. Em especial, a dúvida do Rodrigo é com relação ao comando `OpenURL`, que o Netscape aceita.

Epa, pera! Quer dizer que posso usar esse tal de AppleScript pra abrir uma página no Netscape?

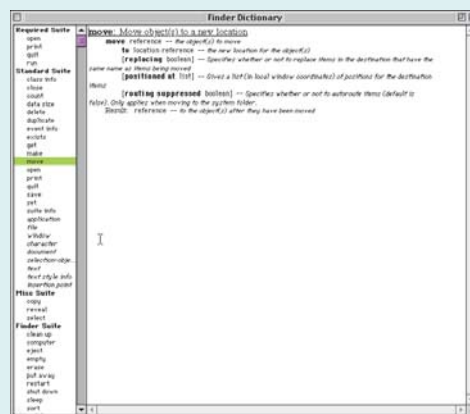
Se você não sabia, sente-se, respire fundo e pause antes do espanto: a resposta é **pode!** Não só para abrir páginas no Netscape, mas, quem sabe, para pegar todos os arquivos de Word de dentro de uma pasta e substituir as datas dos arquivos de 99 pra 1999. Com apenas um duplo clique. E depois de ler esta aula e a continuação no mês que vem, você vai ter uma idéia de como fazer isso.

Para mexer com dicionários, primeiro é necessário entender o que eles são; então, mãos à obra. **Dicionários** são a maneira que a Apple encontrou pra fazer com que todos os programadores explicassem para todo mundo que quisesse usar scripts como fazer para efetivamente usá-los. Para ficar mais fácil de explicar, vou demonstrar. A figura a seguir mostra o que aparece se, utilizando o comando `Open Dictionary`, escolhemos o Finder, dentro do System Folder. Não se preocupe: abrir dicionário não causa mal algum. Você não pode mudar nada no arquivo, então não precisa fazer cópia do Finder (ou qualquer outro programa) para abrir o dicionário. Então, abrindo-o, chegamos a isto:



Repare que há quatro **grupos** (suítes): Required, Standard, Misc e Finder. O grupo Required reúne os quatro **verbos** (ações) que são o mínimo que se pode fazer com AppleScript. Muitos programas, em sua primeira versão, suportam apenas esses quatro primeiros verbos, que servem pra abrir o programa ou um documento, imprimir algo ou fechar o programa. O grupo Standard tem verbos que são o que gostaríamos que todos os programas implementassem, mas infelizmente – porque há outras prioridades quando se faz um programa – nem todos implementam. Se você olhar bem, vai notar que os últimos elementos desse grupo estão em itálico. Isso é pra indicar que estes não são verbos, mas **classes** do AppleScript, algo que faz com que o AppleScript se pareça muito com programação orientada a objeto (se você não entendeu, não esquite... esta última frase foi pro pessoal mais técnico; programadores, mesmo). Vou explicar melhor o que essas classes são mais tarde.

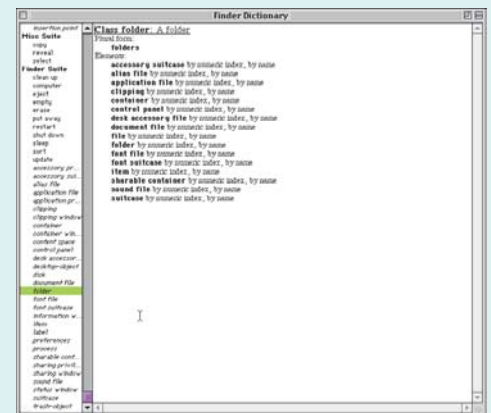
Agora, como eu disse que ia explicar como usar o dicionário, lá vai. Aí em baixo você está vendo como o verbo `move` é descrito. Repare que a primeira linha é apenas uma descrição do que o comando faz. Da segunda linha em diante, temos uma **sintaxe** típica de dicionário de AppleScript. A sintaxe mostra o que é necessário para usar o comando que você escol-



heu, no caso `move`. Primeiro, você terá que dar uma referência ao `move`, pra que ele saiba o que diabos você quer mover. Para ficar bem explicadinho, há um comentário, denotado pelo sinal – e pela grafia em itálico, dizendo que `reference` é o objeto a ser movido. No momento, basta você saber que há uma maneira de referenciar arquivos, pastas, discos rígidos, floppies, CDs, DVDs e o que mais seja possível mover ou fazer qualquer outra ação comum do Finder. Mais tarde, vou explicar; você vai ver que é bem simples.

Depois, você tem que dizer ao `move` para onde vai o arquivo, ou pasta, que você está movendo. Isso é algo que tem que estar em todos os comandos `move` e é referenciado pelo `to` que vem na terceira linha. Mais uma vez, você irá usar o sistema de referência, que vamos explicar mais tarde.

Agora aparece algo novo. Você deve estar perguntando o porquê desses colchetes [ ] na quarta linha, envolvendo `replacing boolean`. Eles estão aí para dizer que esse parâmetro é opcional. **Parâmetro** é tudo aquilo que vem depois de um comando; pode ou não ser obrigatório. No caso, quando você `move` algo, pode no momento da mudança decidir se vai querer apagar qualquer arquivo que esteja na pasta de destino e tenha o mesmo nome do arquivo a ser movido. A palavrinha `boolean` está denotando uma variável lógica, que pode ter dois valores: `true` (verdadeiro) ou `false` (ganha um doce quem adivinhar o que isso significa...). Nesse caso, valor verdadeiro significa que fazemos a ação com (*with*) `replacing`. Se não queremos `replacing`, simplesmente não escrevemos nada. A maneira exata de escrever está no exemplo adiante:



Os outros parâmetros provavelmente serão úteis para alguém algum dia, mas eu, pessoalmente, nunca usei. Repare, no entanto, que há um **resultado** para todas as operações `move`. A operação vai sempre retornar uma referência para o objeto na posição final. Conseqüente-

-mente, a expressão

```
set x to move file "OBELIX:test" to folder  
"OBELIX:teste" with replacing
```

estaria correta. O que o dicionário não diz, e talvez alguns colchetes em torno do resultado ajudassem a lembrar, é que não é necessário usar o resultado; então, a expressão

```
move file "OBELIX:test" to folder  
"OBELIX:teste:" with replacing
```

também está certa. Mais alguns lembretes:

- Para que o `move` funcione, o arquivo `test` tem que estar no disco `OBELIX` e a pasta `teste` já tem que existir, também. Pra testar se eles existem, há a possibilidade de usar os nossos já conhecidos `if... then... else`, mas isso não é papo para a coluna deste mês.

- Não se preocupe, as referências estão explicadas no box aí embaixo.

Se você tentar usar essas frases no seu Script Editor, vai ver que, se tentar checar a sintaxe e usar nomes corretos de arquivos e pastas, vai dar tudo bem. Só que, se você rodar o programa, vai receber um erro. O AppleScript não vai entender nada. Por quê? Bem, simplesmente porque o AppleScript está lá, sem pai nem mãe, tentando resolver tudo sozinho. Lembre que tudo isso começou quando pegamos o dicionário no Finder; então, temos que arrumar um jeito de dizer ao AppleScript que isso é um comando para o Finder e não é para ser executado sozinho.

Fazemos isso utilizando a expressão `tell application "Finder"`. Então, para que aquela linha de script funcione, ela deve estar dentro de um `tell`, da seguinte maneira:

```
tell application "Finder"  
    move file "OBELIX:test" to folder  
    "OBELIX:teste:" with replacing  
end tell
```

Fácil, não? Tudo que estiver entre o `tell` e o `end`

`tell` vai ser interpretado primeiro pela aplicação (ou script) que está sendo chamada e depois pelos verbos comuns do AppleScript.

Então, respondendo à dúvida do Ricardo, para usar o comando `OpenURL` você pode fazer o seguinte script:

```
tell application "Netscape"  
    activate  
    OpenURL "http://www.macmania.com.br"  
end tell
```

Repare que só coloquei "Netscape" depois do `tell`. Como o AppleScript vai saber que quero usar o Netscape Communicator™ 4.5? Fácil. Quando você usar o Check Syntax, ele vai pedir para você encontrar esse tal de Netscape no seu disco rígido, através de uma das caixas de diálogo padrão do Mac. O comando `activate` está lá pra trazer o programa "para a frente", ou seja, fazer com que o Netscape seja o programa ativo. Se você não colocar o `activate`, o programa será aberto da mesma maneira, porém será lançado em background, com o Script Editor se mantendo como aplicação ativa.

### Até mais...

Se você não pegou tudo de cara, fique grudadado no MacPRO, porque no mês que vem vou mostrar uns exemplos de referências e ensinar uma maneira de fazer tudo mais facilmente. A segunda parte de "Corte as Sete Cabeças do Monstro" vai ser de balançar as estruturas! **M**

MAURÍCIO L. SADICOFF

Escreveu esta coluna no Rio de Janeiro usando OBELIX, seu 7200 gordinho que adora javalis. Enquanto esta coluna estava sendo criada, o Mac consumiu 432 javalis recheados, 394 litros de cerveja e uma cereja. Invariavelmente, a culpa pelo crash indigesto de Obelix na manhã seguinte caiu na cereja.

## Fazendo referências a arquivos

Para fazer referências a arquivos e pastas no Mac, usamos um sistema que existe desde 1984, e é por causa desse sistema que não se pode usar o sinal de dois pontos (:) em nomes de arquivos no Mac. Esse sinal existe para indicar que um arquivo (ou pasta) fica dentro de outro, assim como a barra invertida (\) no DOS.

Por causa disso, muitos usuários de PC tentam diminuir o AppleScript dizendo que é apenas uma "linha de comando" como o DOS. Se um deles disser tamanha monstruosidade, responda com um sorriso de quem sabe do que está falando e mande o cara catar coquinhos. Só não dê uma de superior com o pessoal de UNIX, porque os scripts de UNIX, apesar de muito mais difíceis de aprender, são muito mais poderosos que o AppleScript.

Se o seu amigo que usa Windows insistir em dizer que AppleScript é DOS, mostre o script que você fez que pergunta se o usuário gosta de chocolate (lembra? Saiu na segunda parte do nosso curso...). Nada em Windows faz aquilo tão facilmente.

- Arquivos** – Para referenciar arquivos, existem duas maneiras. Referencia-se diretamente o arquivo

através deste formato:

```
file "Disco Rígido:alguma pasta:  
outra pasta:o arquivo"
```

Ou através de um alias, assim:

```
alias "Disco Rígido:alguma pasta:  
outra pasta:o arquivo"
```

A grande vantagem de usar o segundo formato é que, quando se usa alias, o AppleScript cria um alias e sempre encontra o arquivo com o qual você quer trabalhar, onde quer que ele esteja – ao contrário do `file`, que necessita do caminho correto e não testa nada antes de rodar o script.

A grande desvantagem de usar alias é justamente porque o alias procura o arquivo. Numa rede AppleTalk com vários discos disponíveis, esse processo pode demorar um bom tempo.

- Pastas** – Para referenciar pastas é a mesma coisa, só que no fim da expressão há o sinal de dois pontos (:).

# REALbasic 2.0

por Roberta Rabelo Zouain

A linguagem de programação para o resto de nós

Depois do estrondoso sucesso inicial (o programa foi o ganhador do prêmio revelação do Apple Design Award), o REALbasic chega à segunda versão trazendo muitas novidades. Vários bugs foram corrigidos, com a colaboração de seus fiéis usuários; a empresa criou até um programa específico para o envio de bugs e sugestões (muitas delas atendidas), o REALbugs.

A primeira mudança: a REALsoftware agora disponibiliza o programa em duas versões, Standard e Professional, além da versão demo, que pode ser baixada gratuitamente em [www.realsoftware.com](http://www.realsoftware.com), sendo válida por 30 dias, com algumas limitações.

Quem estava esperando uma revisão estética, interface no estilo do QuickTime 4 e coisas do gênero, se decepcionou (fig. 1). Nesse ponto vale destacar a nova janela Properties, mais bonita e elegante, e a reorganização do sistema de ajuda, que agora pode ser ordenado por Themes (temas) ou Alpha (alfabeticamente). Além disso, algumas funções foram adicionadas para facilitar a vida dos programadores, como a possibilidade de comentar uma seleção de texto e – talvez a melhor delas – poder reatar um alias quebrado, muito útil para quem trabalha em equipe ou está constantemente mudando os documentos de lugar. E, para os mais preguiçosos, dá até para fazer alguns programas (por exemplo, um “MoviePlayer Lite”) sem escrever uma linha sequer de código, utilizando o *object binding*, que associa controles a alguns eventos pré-estabelecidos (fig. 2). Mas é possível criar os seus próprios object bindings sem muito esforço, diretamente do REALbasic.

## As novas peças

Nesta versão foram incluídos oito novos controles do Appearance Manager. São eles:

- **Bevel button:** botões com aparência 3D que podem conter ícones e/ou texto. Podem ser utilizados como *push buttons*, *radio buttons*, *checkboxes* ou *pop-up buttons*.
- **Disclosure triangle:** permitem “abrir” e “fechar” uma parte da informação. São os mesmos triângulos que abrem e fecham as pastas no Finder em uma janela visualizada como lista.



fig. 1

- **Little arrows:** par de setas utilizadas para aumentar e diminuir valores.
- **Chasing arrows:** também conhecidas como *asynchronous arrows*, são uma animação que permite indicar a existência de um processo em background.
- **Separator:** linha divisória com aparência 3D.

- **Image well:** retângulo que pode conter figuras.
- **Pop-up arrow:** utilizada em conjunto com um menu contextual para disponibilizar um menu.

- **Placard:** retângulo com aparência 3D, normalmente situado no “rodapé” da janela e utilizado para mostrar informações. Além disso, alguns controles já existentes ganharam novos métodos e propriedades, como os *sockets* e, principalmente, as *listboxes*, que ganharam três tipos diferentes de células, cabeçalho e “outras coisas más” para atender às necessidades dos programas que utilizam bancos de dados.

Em relação ao código “escrito”, as melhorias foram muitas. Para começar, os usuários ganharam as *classes interfaces*, que permitem criar uma lista comum de métodos suportados por várias classes. A execução do código teve um aumento considerável de velocidade, com novos comandos que permitem interromper alguns processos em background do REALbasic.

A interação com outros programas através de AppleEvents tornou-se mais fácil com o novo AE Recorder, que funciona da mesma maneira que o comando Record do Script Editor (fig. 3).

E, por falar nele, agora seus aplicativos podem ser “graváveis”, com o novo recurso de enviar AppleEvents para ele mesmo. Uma morte anunciada foi a dos XCMDs e XFCNs, com a possibilidade de acesso direto da toolbox (no Mac e no Windows), evitando também o emprego de *shared libraries*. Outra boa notícia é o acesso às funções de *gestalt* (válido apenas para o Mac OS). Na versão 2.0.2 foi

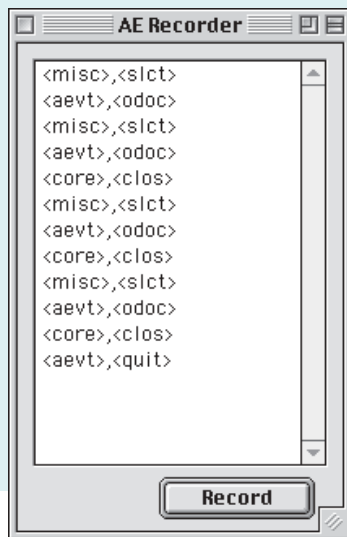


fig. 3

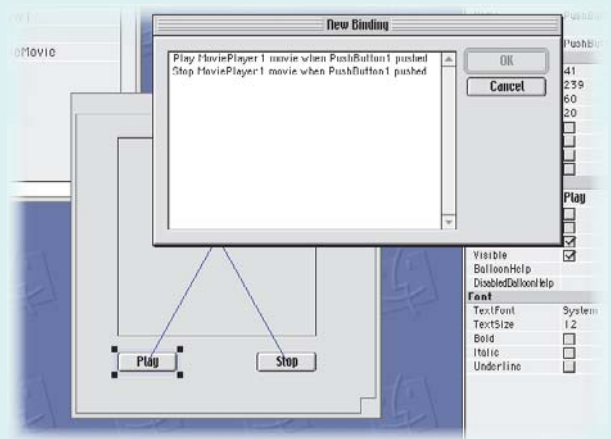


fig. 2

incluída a possibilidade de escolha do tipo de moldura da janela, além dos oito tipos existentes, através da nova propriedade MacProcID.

## Globalização

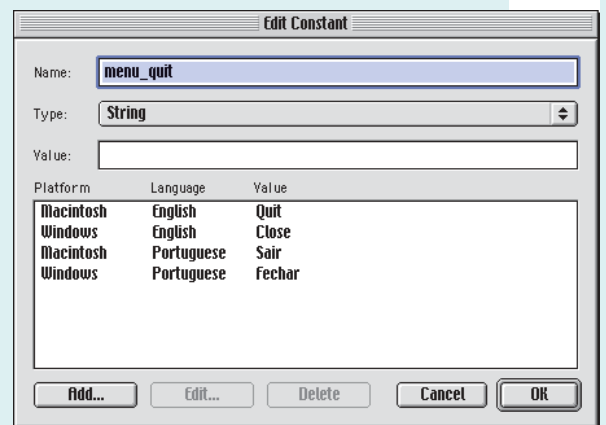
Com a versão 2.0, o REALbasic entra na era da “informática globalizada”, trazendo vários recursos para a localização de seus programas. Além do sistema de Constants (fig. 4), ele suporta diversos idiomas, como os inimagináveis Esperanto, Guarani e Dzongka. Infelizmente não tem Brazilian Portuguese (português “brasileiro”), apenas Portuguese (português de Portugal). Também foi incluído o suporte a caracteres de dois bytes (usados em chinês, japonês etc.).

## O lado de lá

A maior atração da nova versão do REALbasic é sua capacidade de compilar tanto para Mac (PPC e 68K) quanto para Windows (95, 98 e NT), sem nenhum esforço (bem, um clique não é tanto assim). Como isso é possível? Código condicional. Com o novo sistema de Constants nos módulos, que permite a especificação do código quanto à plataforma ou ao idioma, e com os novos comandos `#if`, `*else` e `*endif` (não confundir com os comandos `if`, `else` e `endif`!).

Mas, para os pecevistas mais eufóri-

fig. 4



cos com a novidade, um aviso: a REAL Software já descartou, pelo menos por enquanto, a possibilidade de portar o REALbasic para Windows.

O compilador Java, que esteve presente até os últimos betas, foi retirado, em virtude dos vários bugs e outros problemas encontrados. Quem sabe na próxima versão...

### O lado de cá

Do nosso lado vai tudo bem, obrigado. A nova versão do RB incorporou o “espírito Mac”, melhorando o suporte a várias tecnologias da Apple, como AppleEvents e QuickTime. Aliás, o programa está mais multimídia do que nunca: agora é possível criar, editar e analisar com os novos comandos que dão acesso a trilhas, samples, keyframes, efeitos e compressões do QuickTime. Além disso, o programa suporta todos os tipos de imagens suportadas pelo QT. Nada mau, não é? Mas tem mais: o suporte ao gerenciamento de cores, que na versão anterior só existia em RGB, agora também está disponível em CMYK, e o Color Picker pode ser acessado diretamente, sem a necessidade de plug-in.

### Bancos de dados, enfim

Depois de tantas reclamações, a REAL Software finalmente atendeu às nossas preces: incluiu o suporte a bancos de

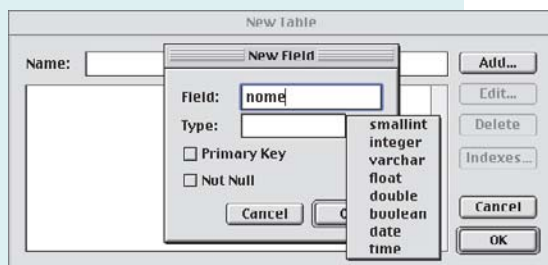
dados no RB (fig. 5). Com ele é possível interagir com bancos de dados 4D Server, Oracle, Microsoft SQL Server, OpenBase, ddf, Postage SQL e qualquer banco de dados compatível com ODBC, além do próprio REAL Database. Os programas construídos podem abrir mais de um tipo de banco de dados, isto é, não é preciso fazer um programa específico para cada tipo de banco de dados, além de eles poderem abrir mais de um banco de dados por vez. As pesquisas (*queries*), por sua vez, são facilitadas pelo novo controle DatabaseQuery.

### Standard ou Professional?

Então, você testou o programa por 30 dias e resolveu comprar. “E agora, qual eu compro: Standard ou Professional?”

Bem, podemos dizer que depende da sua necessidade. Se você vai compilar para Windows ou vai precisar das funções de banco de dados (a versão

fig. 5



## O que faltou

- Compilador Java
- Melhor organização dos projetos
- Documentação
- Melhoria nos controles já existentes

Standard traz isso só como demonstração), então você vai precisar da Professional, que custa US\$ 299,95 (estudantes têm US\$ 120 de desconto). Caso contrário, você pode economizar US\$ 200 e ficar com a versão Standard (se for estudante, paga só US\$ 59,95). E pode escolher entre oito idiomas (mas nada de português, por enquanto). Depois de tantos anos de reclamação, parece que finalmente temos uma linguagem de programação de nível médio (claro que não é nenhum C++) que promete fazer pelo Mac o que o Visual Basic fez pelo Windows. O REALbasic chegou com tudo, e a versão 2.0 veio reafirmar que nem tudo o que é bom precisa, necessariamente, ser difícil. **M**

ROBERTA RABELO ZOUAIN

Co-autora do Overcaster TicTacToe, foi promovida de leitora a colaboradora.

## Onde encontrar

REAL Software: [www.realsoftware.com](http://www.realsoftware.com)

Preço: US\$ 299,95 (versão Professional), US\$ 99 (versão Standard)