



A primeira ferramenta de programação gráfica destinada a desenvolvedores eventuais ou hobbistas nasceu no Macintosh: era o genial **HyperCard**. Só os mais veteranos conhecem esse produto porque, por uma série de erros e infortúnios, a Apple deixou o HyperCard ficar obsoleto e praticamente desaparecer. As grandes inovações do HyperCard foram sua linguagem de programação amigável (o HyperTalk) e seu ambiente de layout intuitivo, que permite que você monte telas arrastando e soltando botões, campos de texto e outros elementos. Em 1991 o mundo Windows ganhou duas ferramentas parecidas: o **Toolbook** (que não deu muito certo) e o **Visual Basic**, que é líder absoluto nessa categoria de produtos na plataforma Intel, mas nunca foi adaptado para o Mac OS. Agora, com o **REALbasic**, os macmaniáticos têm novamente uma ferramenta



barata, fácil e poderosa nessa categoria. O REALbasic é uma versão aperfeiçoada do CrossBasic, um clone de Visual Basic para Macs que surgiu como shareware em 1997. A empresa REALsoftware (nenhum parentesco com o plano Real) adquiriu o programa e contratou seu autor, Andrew Barry, para transformá-lo num produto comercial. A versão 1.0 foi lançada em julho de 98 e a 1.1 em janeiro de 99. Para testar o produto, decidimos implementar um aplicativo simples com a função de armazenar todos aqueles usernames e senhas que somos obrigados a memorizar ou anotar em

algum lugar. O programa, que batizamos de "Chaveiro", grava as senhas do usuário em um arquivo cifrado, protegido por uma senha-mestra. *(Uma descrição mais detalhada do programa você encontra na caixa "Como foi o test drive".)*

Montar a interface de um programa com o REALbasic é como brincar de Lego. Você tem uma caixa cheia de peças prontas, como botões, listas, barras de rolagem etc. Essas peças, chamadas "controles", são coladas sobre uma janela que aparece inicialmente vazia. Janelas adicionais podem ser criadas, e elas

# REALbasic

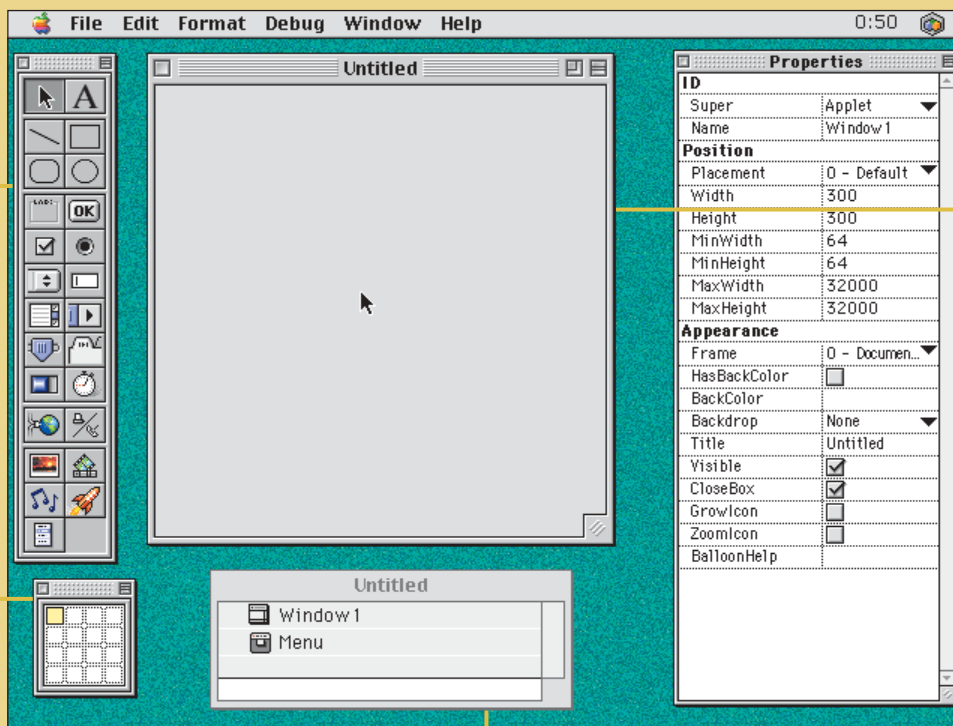
## Construa programas profiças em poucos dias em vez de muitos meses

## Como se constrói um programa com o REALbasic

### Aspecto da interface do REALbasic quando se cria um novo projeto

**Paleta de controles:** apresenta o repertório de elementos, como botões, campos de texto, listas etc., usados para construir a interface do aplicativo. Esses elementos são chamados "controles"

A pequena **paleta de cores** facilita a aplicação de um esquema de cores uniforme no aplicativo



A **janela vazia** no centro é para onde se arrastam os controles da paleta para a montagem da interface. Suas dimensões podem ser alteradas livremente aqui, mas se o programador quiser, a janela pode ter um tamanho fixo para os usuários

A janela de **Properties** permite editar os muitos atributos de cada controle

A **janela do projeto** exibe os grandes componentes do aplicativo em construção. Inicialmente, esses componentes são dois: a janela principal do aplicativo e sua barra de menus

podem ser de oito tipos diferentes, inclusive janelóides que flutam sobre qualquer outra janela, ficando sempre visíveis. Essa é a parte mais divertida e fácil de usar do REALbasic. De fato, é tão conveniente montar interfaces de usuário com esse programa que, mesmo se ele fizesse só isso, já valeria seu preço como uma ferramenta de prototipagem.

## Programação por eventos

A próxima etapa do trabalho é quando acontece a programação propriamente dita. Se o programa que você montou tem três botões, cada um deles terá que ser programado para fazer alguma coisa quando clicado. O mesmo acontece com todos os tipos de controles. Na verdade, cada controle pode ter uma série de procedimentos associados a ele, um para cada evento que ele é capaz de detectar.

A maioria dos eventos corresponde a ações do usuário. Por exemplo, alguns dos eventos reconhecidos por um controle de lista em texto são: `MouseMove`, `Change` e `DragRow`. Instruções escritas no evento `MouseMove` de uma lista são executadas sempre que o ponteiro do mouse se deslo-

ca na área da lista, mesmo sem haver um clique. Assim é possível usar uma caixa de texto ao lado da lista para exibir informações adicionais sobre o item da lista que está sob o ponteiro. O evento `Change` é disparado quando o usuário muda o item selecionado com o mouse ou o teclado. E com algumas linhas de código é possível usar o evento `DragRow`, para permitir que o usuário arraste um item para mudá-lo de posição na lista.

As janelas também são capazes de detectar eventos, como o seu iminente fechamento pelo usuário. O evento `CancelClose` é chamado para permitir, por exemplo, que a janela pergunte ao usuário se ele deseja mesmo sair sem salvar as alterações. E há ainda eventos que não dependem de ações do usuário.

## Controles ocultos

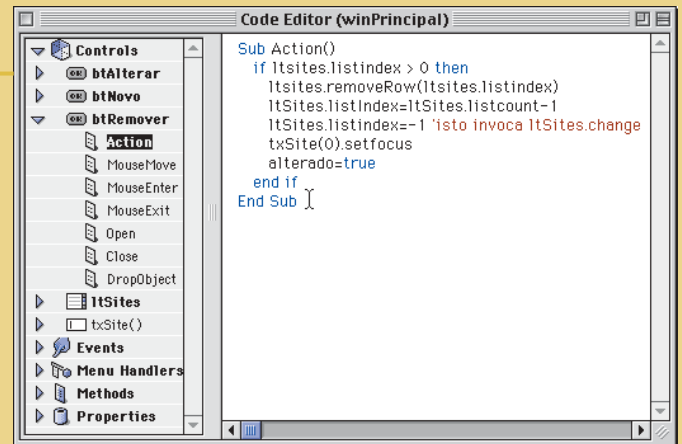
O controle `Timer` pode ser usado para gerar eventos periódicos para controlar uma animação. O controle `MoviePlayer`, que serve para exibir vídeos em QuickTime, dispara o evento `Stop` sempre que o filme pára.

O `Timer` é apenas um dos controles invisíveis

existentes no REALbasic. Esses controles aparecem nas janelas do aplicativo somente quando ele está sendo editado dentro do ambiente de programação. Quando o aplicativo é executado, eles desaparecem, mas continuam funcionando. Um `Timer`, mesmo oculto, continua gerando seus eventos periódicos. Esses controles invisíveis servem para reunir, num pacote conveniente para o programador, um conjunto de funções e eventos relacionados. Outros controles desse tipo são o `Serial` e o `Socket`, que servem para estabelecer e controlar conexões seriais ou IP, respectivamente. O CD player freeware `NetCD`, que foi desenvolvido em REALbasic, usa o controle `Socket` para buscar os nomes das faixas no site [www.eddb.com](http://www.eddb.com). Mas o mais divertido é o controle `NotePlayer`, que toca notas musicais através do QuickTime Musical Instruments.

Em resumo, a rotina do programador de REALbasic consiste em criar controles, criar janelas e codificar o tratamento dos eventos relevantes. Isso tudo é feito dentro de um IDE (Integrated Development Environment – ambiente integrado de desenvolvimento) que

Um duplo-clique sobre um elemento abre a **janela de edição** do programa. No painel esquerdo, sob cada controle, aparecem listados os eventos que esse controle é capaz de gerar em resposta às ações do usuário



No caso de um botão, o evento mais comum é o chamado "action", que é disparado quando o botão é acionado. Neste caso, quando o botão "Remover" é clicado, o aplicativo retira a conta selecionada da lista



As partes principais do nosso aplicativo-exemplo "Chaveiro" são:

- winPrincipal:** a janela principal
- Menu:** a barra de menus
- modCripto:** módulo que contém o método de cifragem
- dlgSenha:** caixa de diálogo que solicita a senha-mestra na entrada
- dlgSenhaNova:** caixa de diálogo que permite alterar a senha-mestra

inclui um editor-navegador de código e um depurador. O editor-navegador, acionado quando se dá um duplo-clique sobre uma janela ou botão do aplicativo, é dividido em duas partes. A da esquerda exibe a lista de controles, eventos e outros itens que pertencem à janela que está sendo editada. A outra parte permite editar o código associado ao item selecionado na esquerda. O editor assinala palavras-chave da linguagem em azul e comentários em vermelho. Quando se digita as letras iniciais de uma palavra conhecida pelo REALbasic, ele se oferece para completá-la escrevendo o resto em cinza. Se o editor adivinhou corretamente, é só teclar **Tab** para completar a palavra. O editor também cuida automaticamente da indentação, o recuo das linhas que serve para deixar mais visível a estrutura do código. Esses recursos tornam o editor ideal para programadores principiantes, mas desenvolvedores experientes notarão que ele está mais para o SimpleText do que para o BBEdit.

### Testando e finalizando

O comando Run do REALbasic esconde as jane-

las de edição e lança o seu aplicativo. Para eliminar bugs, o programador marca "break-points" (pontos de interrupção) nas partes suspeitas de seu código. Sempre que a execução do aplicativo passa por um desses pontos, o programa pára e exibe duas janelas especiais: Variables e Stack. A primeira mostra o valor das variáveis do programador e a segunda, uma listagem que mostra através de que eventos e funções o programa chegou até aquele ponto. O depurador permite que o programa seja executado linha a linha a partir do ponto de interrupção. Dessa forma, o programador pode ir acompanhando em câmera lenta a execução do seu programa, enquanto visualiza as variáveis. É um depurador limitado para padrões profissionais, mas adequado.

Para gerar um aplicativo autônomo, que vai rodar sem o auxílio do REALbasic, você usa o comando Build Application. Entre outras opções, essa caixa de diálogo permite que você defina se quer que o código criado seja PPC, 68K ou fat binary (com ambos os tipos de código). A REALsoftware está prometendo para o início deste ano uma versão capaz de gerar também

## REALBASIC 1.1



REALsoftware: [www.realbasic.com](http://www.realbasic.com)

Preço: US\$ 99

binários Java e, mais para frente, até binários Wintel. A versão demo do REALbasic pode ser baixada de graça do site da REALsoftware. Ela não tem nenhuma limitação, mas expira 30 dias após a instalação original. Os aplicativos autônomos gerados com o demo também expiram 30 dias após a instalação original do ambiente no qual foram produzidos. Apesar de se tratar de uma versão 1.1, a documentação do programa ainda está bastante desorganizada e omissa em muitos pontos. Um ponto positivo, no entanto, é a inclusão de um tutorial para quem nunca programou. Outra limitação relevante é a falta de uma interface que permita conectar o REALbasic a bancos de dados compartilhados, como os servidores SQL, cada vez mais comuns nas empresas modernas. Mas essa também era uma limitação do Visual Basic 1.0, e hoje o que não faltam são maneiras de integrar servidores SQL com o Visual Basic. A REALsoftware, por sua experiência como desenvolvedora de soluções de bancos de dados 4D, deve estar bastante consciente dessa lacuna.

Quem tem alguma experiência com o Visual Basic vai se sentir totalmente em casa com o REALbasic. Se você programa HyperCard ou Macromedia Director, terá que se adaptar um pouco, mas os conceitos básicos são os mesmos e o mais importante, que é a programação orientada a eventos, você já domina. Aqueles que já experimentaram o Director notarão que o REALbasic é muito mais fácil de aprender e usar, e deverá ocupar uma parte de seu mercado. Afinal, mesmo sendo um clone do Visual Basic, o REALbasic já nasceu com recursos multimídia, como os Sprites e o Note-Player, que até hoje não fazem parte do produto da Microsoft. Desde que o Mac ganhou telas coloridas e o HyperCard ficou para trás, os macmaníacos esperam uma ferramenta como essa. Graças ao REALbasic, os programadores de fim-de-semana podem agora se divertir produzindo aplicativos de aparência e funcionalidade profissional. E os programadores profissionais poderão deixar o C++ e o Java somente para os projetos mais exóticos. **M**

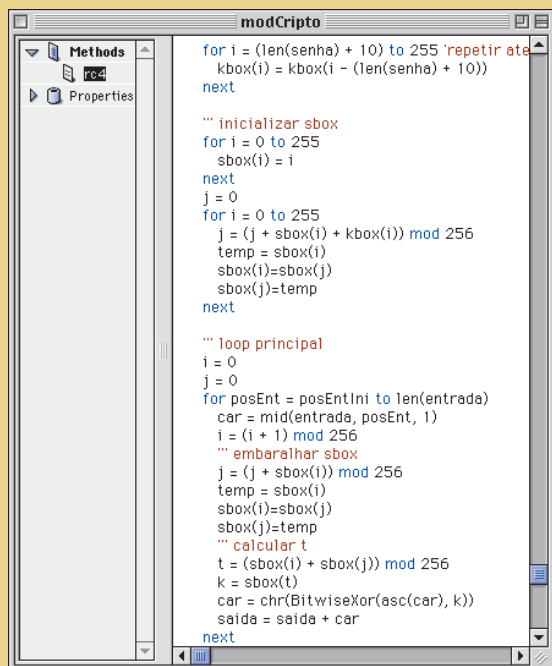
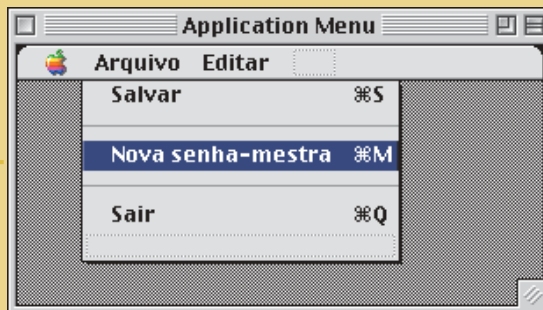
### LUCIANO RAMALHO

Adorava o HyperCard, curtiu o Visual Basic, decepcionou-se com o Java e agora programa mais em Python.

Esta janela é o **editor de menus** do RealBasic. A opção que está sendo editada é "Nova senha-mestra".

Quando acionado, esse comando abre a caixa de diálogo de alteração de senha-mestra.

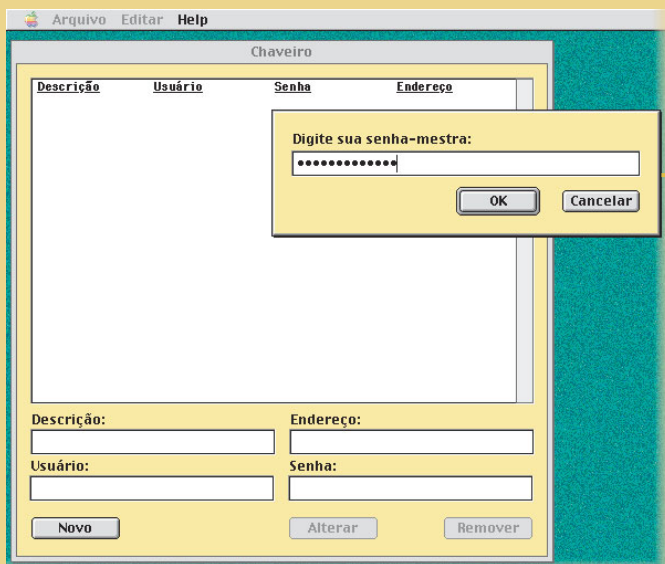
Uma vez captada a nova senha, o arquivo de chaves é cifrado novamente e regravado



O método de cifragem é o **CipherSaber**, uma implementação do algoritmo RC4 da RSA. O método foi escrito em um módulo à parte, chamado "modCripto". Um módulo é uma parte de uma aplicação que não tem interface, somente funções de processamento. A modularização facilita a reutilização do código e a eventual troca do método de cifragem

# Como foi o test drive

Para experimentar o REALbasic, desenvolvemos um aplicativo simples: o “Chaveiro”, um utilitário para armazenar as muitas senhas que hoje somos obrigados a guardar. O Chaveiro grava as senhas do usuário em um arquivo cifrado, protegido por uma senha-mestra.

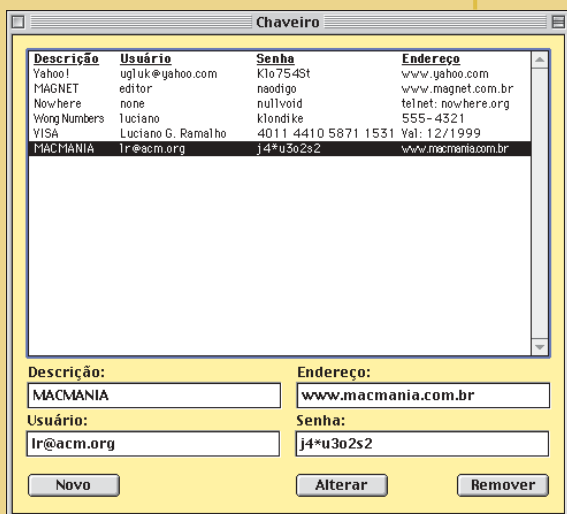


**1** Ao lançar o Chaveiro, o usuário vê a janela principal vazia e uma caixa de diálogo solicitando a senha-mestra.

Enquanto a senha-mestra correta não for digitada, esse diálogo se repete.

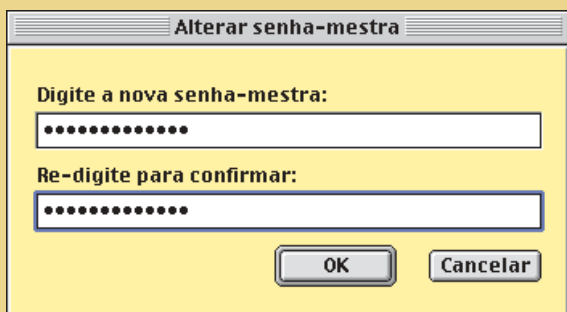
Se o usuário não sabe a senha, sua única opção é clicar “Cancelar”. Isso fecha o aplicativo imediatamente

**2** Uma vez digitada a senha correta, surge a janela principal com as contas do usuário



**3** O principal elemento da janela principal é uma tabela que exibe os dados de acesso das contas. Para cada uma delas aparecem uma descrição, o nome do usuário, a senha e o endereço.

Abaixo da tabela, quatro campos permitem que os dados da conta selecionada sejam alterados. Os três botões servem para inserir novas contas, confirmar alterações e remover contas da lista



**4** Para criar uma nova senha-mestra, o usuário utiliza o comando “Nova senha-mestra” do menu “Arquivo”.

A caixa de diálogo onde a nova senha é cadastrada exige dupla digitação para prevenir erros