

# Mac OS

## A grande convergência

Confuso com as mudanças na estratégia da Apple para o Mac OS? Nosso enviado à WWDC (World Wide Developers Conference), **Rainer Brockerhoff**, conta aqui quais os planos da empresa para o futuro de seu sistema operacional.

Muita gente ficou boiando quando, na última conferência de desenvolvedores da Apple, Steve Jobs subiu ao palco e apresentou ao público o Mac OS X. Segundo Jobs, a transição para o Rhapsody iria demandar um esforço muito grande por parte dos desenvolvedores, daí a necessidade de uma mudança de rumos. Para provar que o Mac OS X é um caminho mais viável, Jobs levou ao palco representantes da Microsoft, Macromedia e Adobe, que endossaram a nova estratégia.

Mas muita gente saiu gritando que aquilo era o fim do Rhapsody, que a Apple tinha abandonado seus planos de fazer um sistema multiplataforma e resolvido apenas dar uma garibada no Mac OS. Ledo engano. O Mac OS X representa uma das melhores, senão a melhor cartada de Steve Jobs desde que assumiu o controle da empresa que criou. Em menos de um ano, ele conseguiu fazer do Mac OS algo que a Apple tinha tentado durante dez anos com retumbantes fracassos. Finalmente teremos um sistema operacional moderno, estável, veloz e compati-

vel com tudo o que foi desenvolvido para Mac até agora.

Para entender melhor como vai ser a transição para o Mac OS X, veja a figura 1. Ela mostra o caminho das pedras que desemboca no Mac OS X. Para entender o interior das maçãs, considere o usuário observando-as por cima. Ele vê apenas a interface de usuário (verde). Entre essa interface e o hardware do equipamento estão várias camadas de software; cada camada usa recursos fornecidos pela camada inferior. A maçã intitulada “Mac OS 8.1 em 1998” representa o Mac OS atual. Em cima do hardware (azul claro), roda o sistema operacional (azul escuro). Logo acima estão os aplicativos (laranja), que têm um estilo visual comum – também conhecido como “Look & Feel” (verde) – apresentado ao usuário. (Essa estrutura não existe apenas no Mac OS. Ela também se aplica a sistemas menos favorecidos, atuais ou passados, como aqueles que começam com a letra W.) O conjunto de chamadas que os aplicativos usam para se comunicar com o sistema opera-

cional se chama API (Application Programming Interface). Como antigamente havia apenas uma API para cada sistema, não era necessário dar-lhe um nome específico. Outra característica dos sistemas anteriores é que eram monolíticos, isto é, suas funções eram um bloco só. Essa estrutura monolítica funcionava muito bem para sistemas simples da época de 8 ou 16 bits, mas com o avanço da informática tornaram-se monstros de complexidade que ninguém entendia completamente e que eram propensos a catástrofes aleatórias, devido à promiscuidade nas suas operações internas. Algumas versões modernas de UNIX procuraram contornar esses problemas dividindo o sistema em camadas. A camada inferior, chamada de “kernel”, implementava funções muito básicas, como alocação de memória, controle de tarefas e acesso a dispositivos externos. A camada intermediária implementava uma API que mais tarde foi padronizada sob o nome de POSIX. Em cima dessa API rodava ou um emulador de terminal teletipo (a famosa “linha de

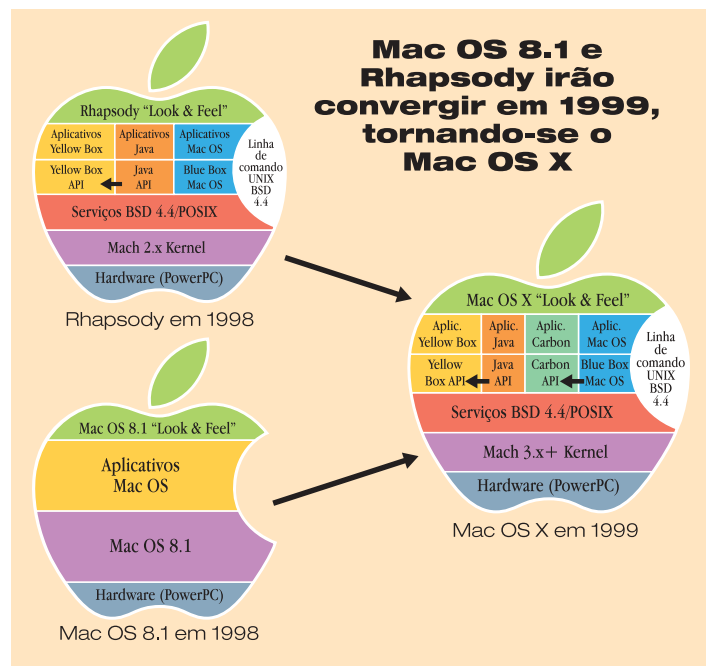


Figura 1

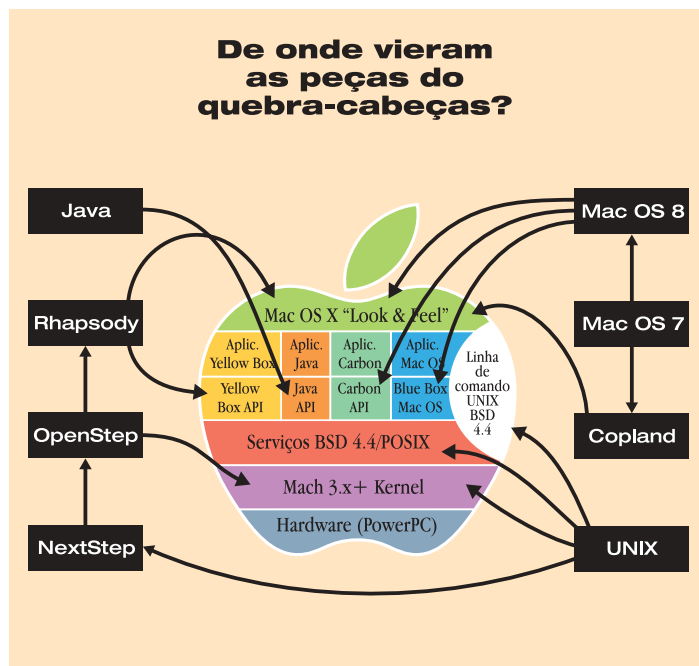


Figura 2

comando”) ou um emulador de terminais gráficos (o “X-Windows”). O pessoal da NeXT adotou essa filosofia básica do UNIX para fazer o NeXTStep e seu descendente, o OpenStep. Quando a Apple adquiriu a NeXT, derivou o Rhapsody do OpenStep, já incorporando idéias do Mac OS. Veja a maçã intitulada “Rhapsody em 1998” na figura 1 e as capturas de tela na página seguinte. A inovação do Rhapsody é que a interface de linha de comando foi praticamente suprimida, formando o vazio da parte “mordida” da maçã. Além disso, foi introduzida mais uma camada com APIs múltiplas para os aplicativos. Essa multiplicidade é uma grande jogada, como já iremos ver.

No Rhapsody, temos então um kernel, que é o “Mach 2.x” (azul escuro); acima deste roda uma API básica que implementa os serviços POSIX (vermelho) derivados do UNIX BSD 4.4. Se algum tarado (ou desenvolvedor) quiser utilizar esses serviços diretamente através de linha de comando, muito bem. Mas, na prática, em cima desta camada há três APIs:

- **“Yellow Box” ou caixa amarela** – Uma API derivada do OpenStep. Ela usa programação orientada a objetos, de maneira especialmente flexível. Um software escrito para essa API tem recursos muito poderosos e é multiplataforma. Por exemplo, ele pode rodar, sem alterações, dentro da “Yellow Box for Windows”, em cima

daqueles famigerados outros sistemas.

- **Java** – O Rhapsody tem uma API padronizada para essa linguagem, podendo executar diretamente aplicativos Java. Há também uma biblioteca “lateral” que permite aos aplicativos Java usar muitos dos recursos da Yellow Box.

- **“Blue Box” ou caixa azul** – Essa é a parte responsável por rodar aplicativos originais do Mac OS 8.1. Esses aplicativos pensam que estão rodando sob o 8.1, mas na verdade estão executando sob o Rhapsody. Ao contrário dos outros programas, os aplicativos que rodam na caixa azul são interdependentes e podem se bombardear mutuamente.

## É DEZ, NÃO XIS

Como falamos em outro lugar, o Mac OS ainda vai sair nas versões 8.5 e 9.0, e o Rhapsody apenas na versão 1.0. Depois disso, em meados de 1999, ambos os sistemas vão convergir no Mac OS X (“X” romano; pronuncia-se “Mac-OS-Dez”). Um gesto que vi dúzias de vezes na WWDC: alguém falando “Mac OS X

(équis)” e alguém da Apple corrigindo: “Mac OS TEN (esticando os dez dedos), not équis” (cruzando os dois indicadores). Eles devem ter ensaiado isso com algum instrutor de aeróbica, de tão automatizados que estavam...

Ainda na figura 1, olhe a maçã “Mac OS X em 1999”. Compare com as maçãs do lado esquerdo. Obviamente, o Mac OS X é o Rhapsody rebatizado e com uma API a mais, o “Carbon” (verde escuro); além disso, a versão do kernel evoluiu para 3.x. Mas não é uma complicação exagerada, toda essa confusão de camadas e APIs? O novo sistema não vai ser muito grande, muito instável, complicado de operar, e ninguém vai querer desenvolver software para rodar nesse monstro?

Nada disso! Vamos explicar parte por parte do Mac OS X e porque ele vai ser simples, rápido e agradar a gregos e troianos entre os desenvolvedores. De baixo para cima:

- **Hardware** – Por enquanto a Apple afirma que o Mac OS X será otimizado para o PowerPC G3. Definitivamente não rodará em Macs antigos (68K ou PowerPC 601). O destino

dos 603 e 604 é inseguro ainda; alguns poderão funcionar, mas sem direito a reclamações.

- **Kernel** – Aqui estão funções que isolam os sórdidos detalhes sobre o hardware. Alocação de memória, memória virtual, se seu disco rígido é SCSI ou IDE, se sua

Ethernet está na placa-mãe ou numa placa PCI, quantas CPUs seu sistema tem, se existem outros softwares competindo por tempo nessas CPUs, tudo isso é administrado pelo kernel. E finalmente teremos memória protegida, o que significa que um aplicativo não vai poder fazer besteiras e levar consigo outros aplicativos ou até crashear o sistema todo.

- **Serviços BSD/POSIX** – Os tarados por UNIX podem fazer as suas gracinhas aqui, mas o usuário comum nem ficará sabendo que existe isso; por outro lado, os implementadores das APIs têm um conjunto poderoso de funções em comum. É onde são administrados os sistemas de arquivos (para que aplicativos Java vejam os mesmos diretórios e arquivos que os aplicativos da Yellow Box, por exemplo) e as telas (para que janelas gerenciadas por diversos aplicativos convivam pacificamente).

Aqui ainda são administradas as conexões de rede local ou Internet, o controle de acesso para sistemas multiusuário ou para servidores e outras funções comuns às APIs.



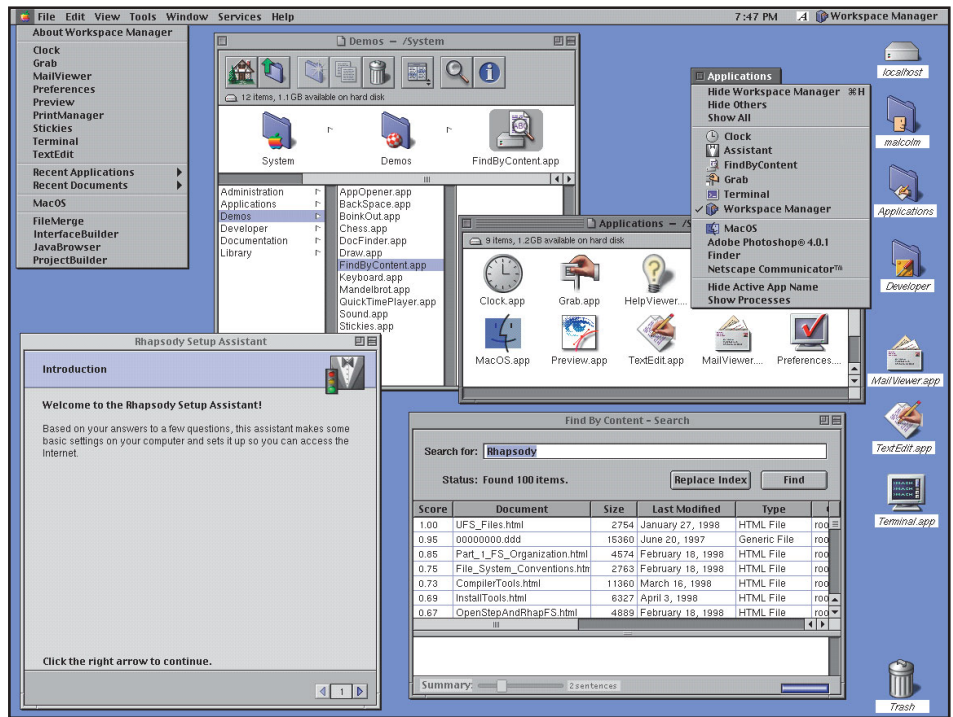
*As boas-vindas provisórias do Rhapsody*

• **API “Yellow Box”** – Essa é a API mais rica e eficiente. A Apple sugere aos desenvolvedores que criem seus novos softwares sempre para essa API, para poder usar uma rica biblioteca de objetos herdada do OpenStep, e para que possa, de quebra, funcionar sob Windows 9X ou Windows NT sem esforço adicional por parte do desenvolvedor.

• **API “Java”** – Aqui também entra a jogada dos aplicativos multiplataforma. Não é tão eficiente, posto que Java é linguagem interpretada, mas o mercado exige. E, de quebra, é um jeito rápido de desenvolver utilitários de menor porte. Muitos dos utilitários e painéis de preferências do Rhapsody atualmente existente já são implementados em Java.

• **API “Blue Box”** – Essa API é efetivamente um emulador para se rodar softwares antigos de Mac, ou seja, aqueles seus aplicativos antigos vão rodar nela, desde que não dependam de extensões e painéis de controle que falem diretamente com o hardware.

• **API “Carbon”** – Essa é a grande jogada que efetivamente assegurou o futuro do Rhapsody. Depois do lançamento original, verificou-se uma relutância de grandes empresas como a Adobe e a Microsoft em converter seus produtos para a Yellow Box; até hoje, 99% dos softwares anunciados para essa API são de empresas que já produziam para OpenStep, e nenhum é dos grandes fabricantes. Com a definição do Carbon, essa relutância terminou. No próprio lançamento do Carbon em maio, a



*A interface do Rhapsody Developer Release 2 mostra algumas características novas (ícones grandes, menus destacáveis, nome do programa na barra de menu) que deverão ser assimiladas pelo Mac OS X. A janelinha característica do Workspace Manager, o Finder do OpenStep, são a única outra dica de que este não é exatamente o Mac OS.*

## O OVO DE JOBS

Como a Apple conseguiu essa reviravolta? Muito simples: em retrospectiva. A API atual do Mac OS 8.1 consiste em cerca de 8.000 funções. Destas, retiraram cerca de 2.000, que

foram consideradas pouco utilizadas ou que comprometeriam as funções desejadas de proteção de memória e funções de multitarefa. Acrescentaram algumas poucas para melhor integração com a camada BSD/POSIX. E pronto! A Apple já publicou um utilitário chamado “Carbon Dater”, que permite verificar a porcentagem de um aplicativo que deve ser modificada para rodar sob o Carbon. A maioria dos aplicativos está na faixa de 5

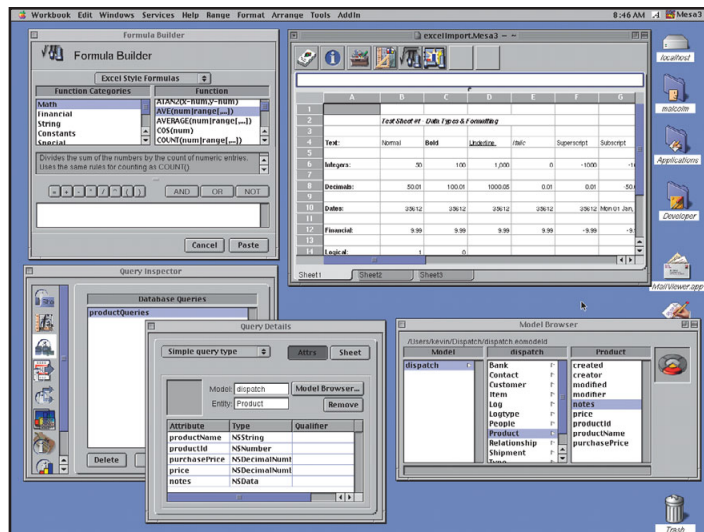
a 10%; estima-se que um aplicativo médio levará uma a duas semanas de programação e um mês de testes para ser convertido. Como o software de compatibilidade para rodar Carbon sob Mac OS deverá sair antes do final de 1998, é garantido que o Mac OS X já sairá do lança-

mento com 99% de todos os softwares importantes rodando sob ele, cifra que, para o Rhapsody, infelizmente será de 1%. Então o Rhapsody já nascerá morto? As opiniões divergem. Certamente a grande maioria dos usuários da plataforma PowerPC só colocará as mãos nele na sua encarnação como Mac OS X. O Rhapsody 1.0, que sairá em setembro de 1998, terá interesse para desenvolvedores e algumas aplicações como servidor em redes locais. A prometida versão do Rhapsody 1.0 para Intel tem seu futuro incerto; muitos acreditam que a Apple não terá recursos humanos para lançar ou manter essa versão, e que seus esforços serão concentrados na Yellow Box para Windows. Esta, sim, deverá ser mantida por tempo indeterminado. Há até rumores que ela poderá ser licenciada pela Microsoft para inclusão no Windows NT 5.0 ou em versões futuras do Windows 98.

O futuro do Rhapsody como produto independente do Mac OS X após o seu lançamento é nebuloso. Nossa opinião é que ele só será mantido se o mercado de servidores aderir maciçamente à Yellow Box; dependendo da paulada que a Microsoft levar do governo americano, não é de todo implausível... mas é pouco provável. **M**

### RAINER BROCKERHOFF

*É desenvolvedor de software para Mac e diretor da Delta Consultoria.*  
rainer@ez-bh.com.br



*O Rhapsody estréia quase sem aplicativos nativos, e os que existem são na maioria herança do OpenStep. Nesta tela vê-se a planilha Mesa e outros programas da atual geração.*

Adobe apresentou uma versão do Photoshop convertida para a nova API. Além disso, aplicativos Carbon rodarão também sob Mac OS 8.5 e 9.0, quando forem lançados. Dessa forma, a Apple recomenda que todos os aplicativos existentes sejam convertidos para o novo padrão.