

Quem tem medo do WebObjects?

Você já deve ter ouvido o famoso termo “cabeça de bacalhau”.

O WebObjects é um bom exemplo dessa expressão:

uma daquelas tecnologias sobre as quais muita gente já ouviu falar, mas poucos efetivamente viram funcionando ou sabem exatamente dizer do que se trata.

O objetivo deste artigo é oferecer uma visão geral da tecnologia WebObjects e como ela pode ser aplicada na vida prática de quem hoje precisa de um ambiente de primeira classe para a criação de soluções de comércio eletrônico.



por Tiago Ribeiro e Fabio Ribeiro

Resumidamente, o WebObjects é três coisas ao mesmo tempo:

- Um ambiente para desenvolvimento visual de aplicativos para a Web e Internet.

- Um *framework*, ou seja, um pacote de funções pré-escritas que facilita o desenvolvimento de aplicações, baseado em padrões como JDBC, XML, RMI e outras tecnologias.

- Um servidor de aplicações Web multi-plataforma.

Para entender a importância de um produto como o WebObjects, é importante, antes de mais nada, entendermos quais tipos de problemas um servidor de aplicações visa resolver.

Em seus primórdios, a World Wide Web (WWW ou, no popular, Web) permitia única e exclusivamente a distribuição de documentos HTML *estáticos*; ou seja, a informação contida em uma página HTML era sempre a mesma, independentemente do usuário que a estivesse visitando.

No momento em que a Web passou a ser vista como uma porta de entrada para toda uma série de serviços – entre eles o de comércio eletrônico, aplicações cliente/servidor e *home banking* –, surgiu a necessidade de integração entre o **servidor de Web** (o software responsável por fazer a distribuição dos documentos HTML) e um **servidor de bancos de dados** (o software responsável por armazenar informações vitais, como o cadastro de clien-

tes, produtos, ordens de serviço etc.). Para isso, passou a ser introduzido dentro do servidor de Web um software específico (conhecido como CGI ou *Common Gateway Interface*), normalmente escrito nas linguagens C ou Perl. Esse software tem como função principal receber as solicitações enviadas pelo usuário que estiver usando o navegador (normalmente, via um comando POST) e processá-las da forma mais adequada (redirecionar a solicitação sob a forma de um email, fazer uma consulta a um banco de dados etc.). Além disso, esse software também pode, a partir das informações obtidas, gerar um documento HTML específico para o usuário que estiver acessando a página em questão.

Páginas dinâmicas: o que acontece por trás daquele clique

1 Um cliente navegando em uma loja de CDs na Internet deseja encontrar todos os CDs da sua banda favorita; para isso, ele digita o nome da banda em um campo de formulário e clica no botão “Enviar”.

2 Um script CGI alojado no próprio servidor Web recebe a solicitação de busca e a informação a ser procurada (no caso, o nome da banda). O script abre uma conexão com o servidor de bancos de dados responsável pelo cadastro das informações do site e executa uma busca nele, usando como parâmetro o nome da banda que foi enviado pelo usuário. ▶

WebObjects continuação

3 O resultado da busca é retornado pelo servidor de bancos de dados para o script CGI, e este usa os dados obtidos para construir, em tempo real, um documento HTML personalizado para o usuário.

4 Finalmente, o servidor Web envia para o usuário a página em HTML que foi gerada pelo script CGI. Simples, não?

A mecânica descrita acima funciona muito bem em casos simples, nos quais as aplicações não sejam complexas, o volume de transações efetuado não seja crítico e não exista um horizonte de crescimento muito grande para o site ou solução em questão.

Os problemas começam a aparecer exatamente no momento em que uma dessas variáveis se altera: no momento em que a sua aplicação ou o seu site começam a crescer.

Outro problema: uma solução deste nível não nos permite pensar em recursos mais avançados, como por exemplo:

- Capacidade de rodar múltiplas cópias de uma mesma aplicação para permitir o atendimento de um número maior de clientes.
- Gerenciamento automático de carga entre

aplicações (no qual o número de conexões de clientes é dividido entre as aplicações existentes, de acordo com o tempo de execução).

- Recuperação automática do servidor em caso de falha.

- *Clustering* (divisão do processamento entre múltiplas CPUs conectadas em rede).

Para suprir essa demanda surge o **Servidor de Aplicações**, um software responsável única e exclusivamente pelo gerenciamento e execução das aplicações Web.

Hoje, existem diversos padrões no mercado para o desenvolvimento de aplicações corporativas desse nível. Entre eles estão:

- **J2EE** (Java2, Enterprise Edition), tecnologia desenvolvida pela Sun e parceiros.

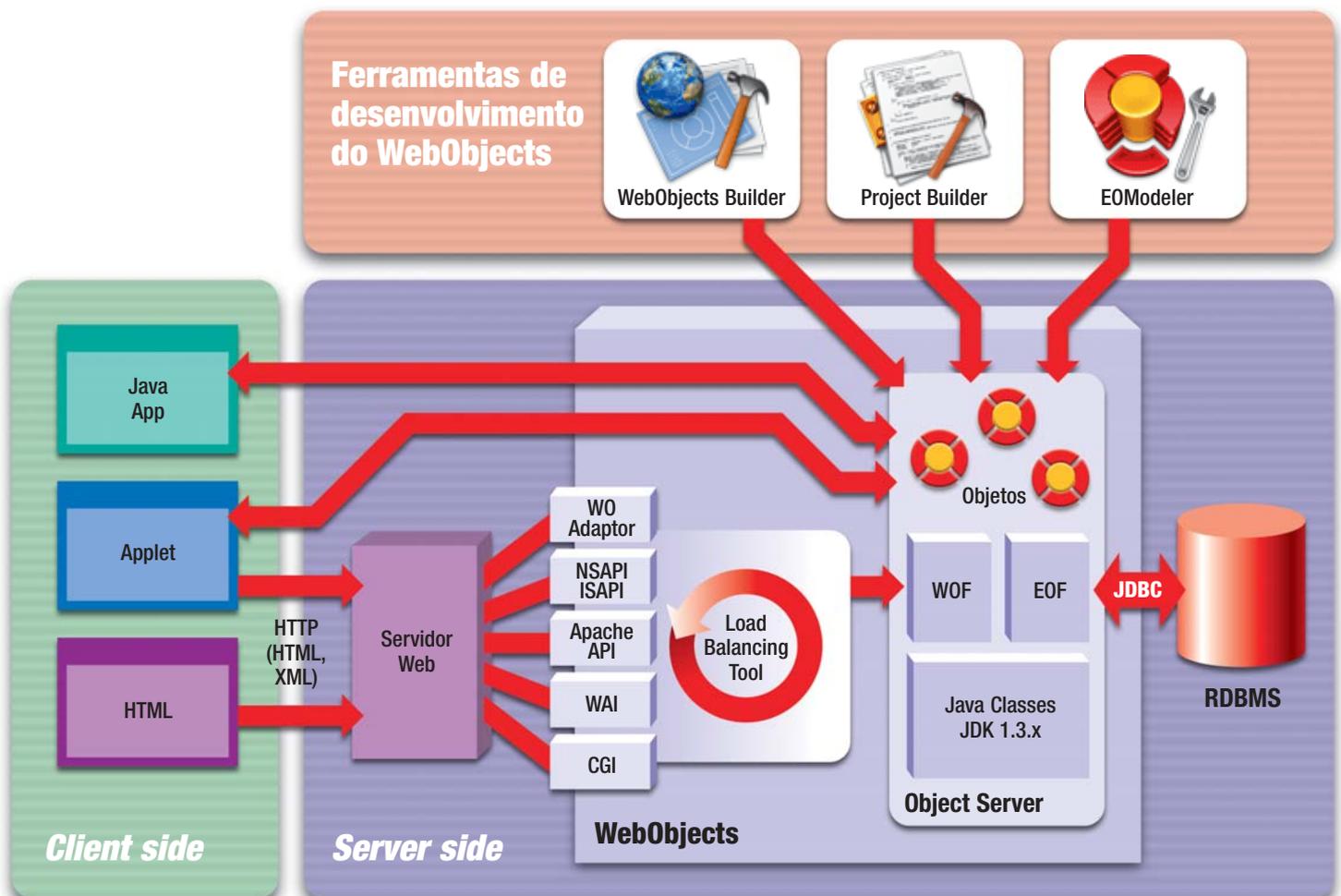
- **.NET** (“dot-NET”), desenvolvida pela Microsoft.

- **WebObjects/EOF**, desenvolvidos pela Apple. Cada um desses padrões se propõe a solucionar os problemas descritos anteriormente de formas diferentes. Por exemplo, a solução da Microsoft se utiliza de soluções totalmente proprietárias, incluindo aí a linguagem de programação preferencial para o desenvolvi-

mento de aplicações, **C#** (“C-Sharp”). Já as soluções da Sun e da Apple estão construídas no topo de uma tecnologia aberta, a **J2SE** (Java2, Standard Edition). Ambas trabalham com a linguagem Java como linguagem padrão para a criação das aplicações.

O que torna o produto da Apple diferente dos concorrentes é o fato de ele possuir incorporado um conjunto de ferramentas visuais para o desenvolvimento rápido de aplicações e sua integração com bancos de dados. Graças a isso, boa parte do código – que geralmente deve ser escrito pelo desenvolvedor numa solução J2EE para acesso a funcionalidades específicas, como bancos de dados – é totalmente abstraída do desenvolvedor WebObjects, através de um conjunto de *frameworks* já incorporados no pacote, entre eles o EOF (Enterprise Objects Framework), para integração com bancos de dados. Conforme já tínhamos falado, um *framework* é um conjunto de funções pré-escritas e que são de uso frequente do desenvolvedor (por exemplo, funções para controle de sessões, acesso a bancos de dados, transações etc.). No Web-

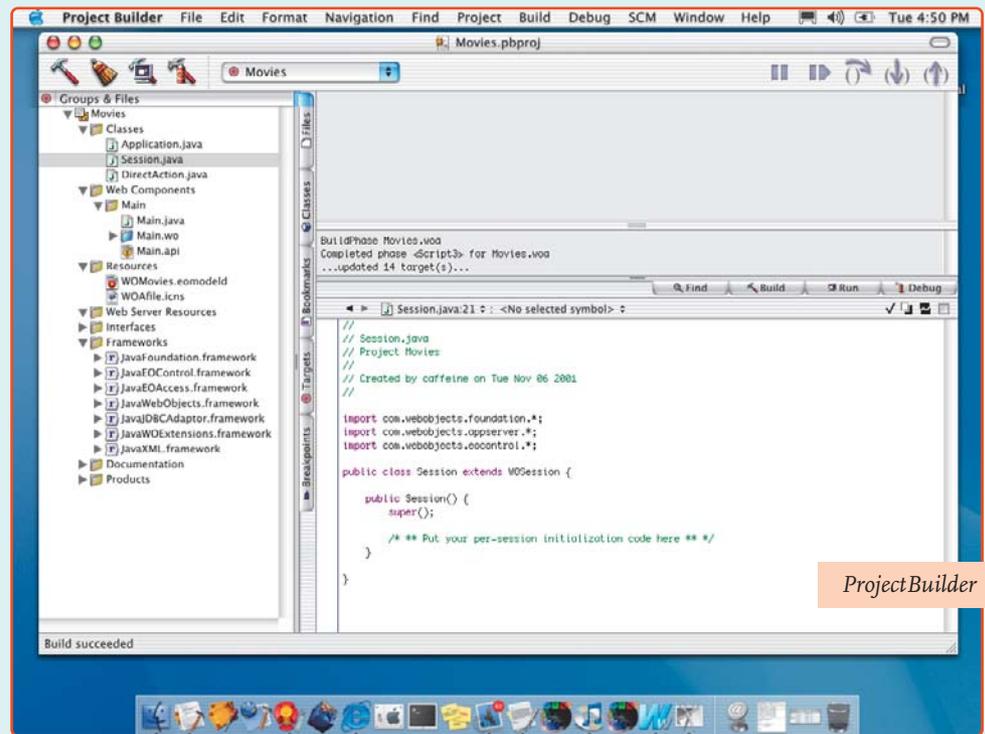
A maior vantagem do WebObjects são suas ferramentas de desenvolvimento visuais



Objects, esses *frameworks* estão integrados às ferramentas de desenvolvimento de tal forma que é possível para um desenvolvedor criar uma aplicação Web completa sem a necessidade de escrever uma única linha de código! Estes recursos reduzem de forma drástica o tempo gasto com a codificação e a manutenção do código já escrito, acelerando desse modo o ciclo de desenvolvimento de projetos e permitindo que mesmo alterações substanciais possam ser feitas sem a necessidade de se reescrever a aplicação por completo.

O que vem dentro da caixa

Antes de mais nada, é importante destacar que o WebObjects é oferecido para o usuário sob duas formas distintas: uma delas através do Mac OS X Server (sistema operacional da Apple baseado no Mac OS X e destinado para serviços de arquivos, impressão, redes, Internet e gerenciamento de grupos de trabalho); a outra, de forma separada. No primeiro caso, o que o usuário recebe é apenas o servidor de aplicações – o Mac OS X Server não contém as ferramentas para o desenvolvimento de aplicações com o WebObjects. Já quem compra a caixa do WebObjects recebe o pacote completo (a licença/ferramentas para desenvolvimento de aplicações e uma licença ilimitada do servidor de aplicações). Neste caso, o usuário tem a opção de instalar o pacote de desenvolvimento em um dos seguintes sistemas operacionais: Mac OS X, Mac OS X Server e Windows 2000. Da mesma forma, poderá instalar o servidor de aplicações em qualquer sistema operacional compatível com a especificação J2SE (a Apple certificou, até o



ProjectBuilder

momento, o Mac OS X, Mac OS X Server, Windows 2000 e Solaris/SPARC).

Desenvolvimento

As seguintes ferramentas de desenvolvimento acompanham o WebObjects:

Project Builder

Similar ao Project Builder do CD de ferramentas de desenvolvimento que vem na caixa do Mac OS X, porém desenvolvido

para trabalhar também com aplicações WebObjects. Este software é responsável pelo gerenciamento dos componentes de suas aplicações: códigos-fonte, imagens, modelos de bancos de dados, arquivos HTML dinâmicos etc. Além disso, é com ele que você irá compilar (construir) e testar a sua aplicação, com o auxílio de um depurador (software que serve para detectar falhas durante a execução do aplicativo).

EOModeler

Software responsável pela integração entre os componentes de sua aplicação e o servidor de bancos de dados.

WebObjects Builder

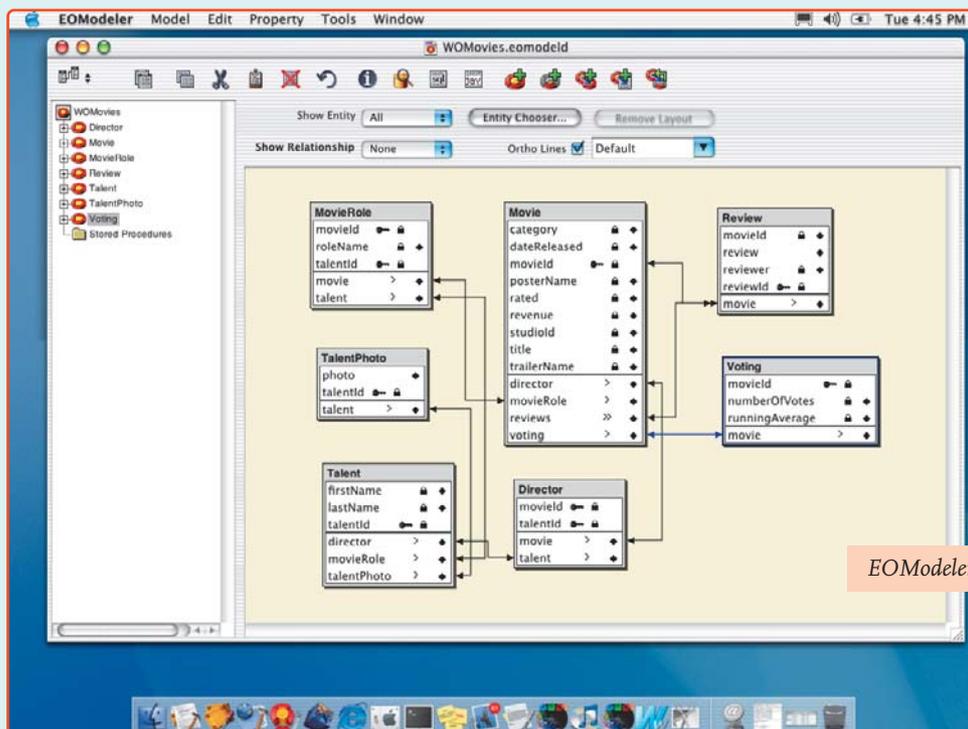
Editor HTML integrado ao WebObjects; permite criar documentos HTML dinâmicos que sirvam como interface para a sua aplicação Web.

Distribuição

Lógica de desenvolvimento

O desenvolvimento típico de uma aplicação para a Web com o WebObjects passa pelos seguintes passos:

- 1 Modelagem dos componentes de sua aplicação com o EOModeler.
- 2 Criação do projeto de sua aplicação com o Project Builder.
- 3 Criação da interface de suas aplicações com o WebObjects Builder.
- 4 Codificação, construção da aplicação e testes.
- 5 Distribuição.



EOModeler

Ferramentas de desenvolvimento para OS X

Já está na rua o CD Developer Tools 10.1

Está na hora de os desenvolvedores porerem as mãos no CD com as ferramentas de desenvolvimento do novo Mac OS X 10.1, que foi lançado no final de setembro. A Apple colocou no ar o novo **Mac OS version 10.1 Developer Tools**. Para baixar o programa é preciso ir ao site do Apple Developer Connect (ADC) e digitar a senha. Caso você não tenha ainda uma conta no ADC, não se preocupe: o cadastro é gratuito, bastando preencher um breve formulário. Para instalar as ferramentas, baixe do site a imagem de disco (.dmg) do CD Developer Tools 10.1. Esse arquivo (de 187,1 MB) contém o instalador das ferramentas. É preciso ter o Mac OS X 10.1 instalado.

Apple Developer Connect Web:

<http://connect.apple.com>

CodeWarrior para Mac OS X

Compilador da Metrowerks promete converter software para rodar nativo no OS X

A Metrowerks lançou a nova versão do seu programa de desenvolvimento, o **CodeWarrior 7.0**. Segundo a Metrowerks, com o novo CodeWarrior é possível desenvolver aplicativos para o Mac OS clássico, "carbonizados", Cocoa e Java. A versão 7.0 inclui também um conversor que transforma automaticamente projetos "carbonizados" em Mach-O, que rodam nativos no Mac OS X. Outras novidades: desenvolvimento multiplataforma (também para Windows), um *framework* C++ de aplicativos orientado a objeto e uma versão "carbonizada" do Ambiente de Desenvolvimento Integrado (IDE, em inglês). O CodeWarrior 7.0 custa US\$ 599 nos EUA. Usuários registrados da versão anterior fazem o upgrade por US\$ 299.

Metrowerks: www.metrowerks.com

Instale programas de Unix sem mexer no Terminal

Open OS X portou uma porção de programas

Com a chegada do Mac OS X, uma nova fronteira foi aberta aos macmaníacos: a possibilidade de utilizar programas desenvolvidos para Unix. Como o uso da linha de comando ainda assusta muita gente, algumas empresas já começaram a portar seus softwares para rodarem bonitos na interface Aqua.

A **Open OS X** disponibiliza programas conhecidos da plataforma Unix para o Mac OS X, com instaladores que não requerem comandos complicados. Entre os programas oferecidos estão o GIMP (editor de imagem Open Source, similar ao Photoshop), MySQL (banco de dados) e X-Window (interface gráfica). A Open OS X usa um sistema de pacotes similar ao do Installer do Mac OS X, evitando assim contatos indesejados de macmaníacos com a linha de comando e facilitando a instalação.

Os pacotes da Open OS X são vendidos em CD por US\$ 30, apesar de os programas originais serem open source, mas para aqueles que não têm interesse em mexer no Terminal para instalar um aplicativo, o investimento pode ser interessante.

Open OS X: www.openosx.com

WebObjects continuação

Nós nos aprofundaremos mais em cada uma dessas etapas do desenvolvimento de aplicações nos nossos próximos artigos sobre WebObjects. Aguarde! **M**

TIAGO RIBEIRO tiago.r@apple.com.br

FABIO RIBEIRO fabio.g@apple.com.br

Trabalham no Apple Developer Connection (ADC) da Apple Brasil.

Fique ligado

• **CGI** – *Common Gateway Interface*. Software de servidor que provê a serviços de informação (como bancos de dados) a capacidade de gerar páginas Web dinamicamente para o visitante do site.

• **EJB** – *Enterprise Java Beans*. Componentes de software em Java para criação de aplicações cliente-servidor baseadas em servidores de aplicações Java2EE.

• **EOF** – *Enterprise Objects Framework*. Desenvolvido pela Apple para facilitar a integração entre aplicações Web e servidores de bancos de dados. Atualmente, a tecnologia é construída no topo de outra tecnologia de acesso a bancos de dados, denominada JDBC.

• **EOModeler** – Ferramenta do pacote do WebObjects destinada à integração entre a aplicação Web e o servidor de bancos de dados.

• **HTML** – *HyperText Markup Language*. Formato de arquivo padrão de documentos de páginas para a World Wide Web.

• **J2EE** – *Java2, Enterprise Edition*. Especificação da Sun para o desenvolvimento de aplicações para o mercado corporativo. Construída sobre a especificação J2SE, inclui tecnologias como JSP (Java Server Pages), EJB (Enterprise Java Beans), JNDI (Java Naming and Directory Interface), JTA (Java Transaction API), JMS (Java Messaging Service) e outras.

• **J2SE** – *Java2, Standard Edition*. Especificação da Sun que inclui as funcionalidades básicas da plataforma Java, entre elas JFC (Java Foundation Classes), AWT (Abstract Windowing Toolkit) e outras. A versão incorporada ao Mac OS X 10.1 é a 1.3.1.

• **Java** – Linguagem de programação orientada a objeto, desenvolvida pela Sun Microsystems e adotada por mais de 400 empresas fabricantes de

hardware e software; hoje, é utilizada por mais de 2,5 milhões de programadores em todo o mundo.

• **JDBC** – *Java Database Connectivity*. Tecnologia Java para conectividade com bases de dados.

• **MVC** – *Model-View-Controller*. Paradigma de programação que envolve a separação entre componentes da aplicação responsáveis pela sua representação lógica (*Model*), a interface de visualização de informações (*View*) e os componentes de controle (*Controller*).

• **OOP** – *Object-Oriented Programming* (Programação Orientada a Objeto).

• **RDBMS** – *Relational DataBase Management Server* (Servidor de Bancos de Dados Relacional).

• **RMI** – *Remote Method Invocation*. Tecnologia Java para comunicação entre aplicações cliente-servidor *three-tiered*.

• **Servidor de aplicações** – Software responsável pela execução de aplicações.

• **SQL** – *Structured Query Language*. Linguagem de programação utilizada para execução de buscas dentro de bancos de dados. Utilizada pela grande maioria dos servidores de bancos de dados disponíveis no mercado.

• **WebObjects Builder** – Ferramenta do pacote do WebObjects destinada à construção de documentos HTML dinâmicos; similar aos editores HTML disponíveis comercialmente, porém otimizado para o WebObjects.

• **XML** – *Extended Markup Language*. Formato de arquivo muito utilizado para armazenamento, conversão (em conjunto com um DTD) e troca de informações. É um dos formatos utilizados para troca de informações entre o que se denomina *Web Services*.

Para saber mais

Onde aprender

DRC Consulting

(011) 3168 2123 www.drc.com.br

A-HAND (Unicamp)

www.ahand.unicamp.br

UVA (Universidade Veiga de Almeida)

www.uva.br

Alguns desenvolvedores WebObjects no Brasil

Web Soluções (11) 3168-3114

www.websolucoes.com.br

Progonos (0800) 770-1915 www.progonos.com

Zenza (11) 3168-9333 www.zenza.com.br

Alguns sites e produtos feitos com WebObjects

Adobe <http://store.adobe.com>

Americanas.com <http://www.americanas.com.br>

ATT Wireless <http://ecare.attws.com/dss/gateway/new-99/1/shep-ocs>

Deutsche Bank <http://group.deutsche-bank.de>

FlightArrivals www.flightarrivals.com

MCI [http://www.mci.com/cgi-bin/](http://www.mci.com/cgi-bin/WebObjects/oam.woa)

[WebObjects/oam.woa](http://www.mci.com/cgi-bin/WebObjects/oam.woa)

PowerSchool <http://www.powerschool.com>

Tok&Stok <http://www.tokstok.com.br>

Toyota Canada www.toyota.ca

Warner Chappell www.warnerchappell.com

Metendo a mão no Unix

Parte 4: configurando o bash

por **Alberto V. Mendonça**

bash (abreviação de *The GNU Bourne Again Shell*) é um ambiente (*shell*) para a linha de comando. É mais antigo e mais refinado que o *tcsh*, o *shell* padrão do Mac OS X, e é o mais usado no Linux e outras variações de Unix. A instalação do **bash** requer um conhecimento básico de utilização da linha de comando.

Passo 1

Faça o download do arquivo de instalação do **bash** para Mac OS X, a partir de <http://homepage.mac.com/aquavivae/apps/Bash.dmg.gz>. Esse pacote já contém o programa compilado, isto é, pronto para rodar. O arquivo é baixado, automaticamente descomprimido pelo StuffIt Expander e montado pelo Disk Copy como uma imagem de disco no Desktop.

Passo 2

1 Duplo-clique o arquivo `Bash.pkg` dentro da imagem de disco para abrir o instalador e dê OK nas telas. (O instalador põe os arquivos no diretório oculto `/usr/local`.)

2 Abra o Terminal e obtenha acesso *root* com o comando `su` (entre a senha de *root* quando requisitada – ela deverá ter sido previamente habilitada pelo NetInfo Manager):

3 Como *root*, dê este comando:

```
ln -s /usr/local/bash/bin/bash /bin/bash
```

Isso cria um *link* (*alias* do Unix) do **bash** no diretório `/bin`, onde já residem outros programas de linha de comando.

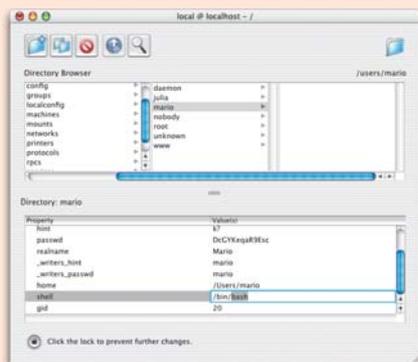
4 Saia da conta *root* digitando `exit` e acesse o *shell* **bash** digitando, simplesmente, `bash`.



O **bash** abre dentro do outro *shell* que você já estava rodando no Terminal (no caso, o *tcsh*). Dando `exit`, você volta ao *shell* inicial.

5 Mude o *shell* padrão para sempre entrar direto no **bash**. Há dois métodos:

Opção 1 - Use o NetInfo Manager (`/Applications/Utilities/NetInfo Manager`). Abra o programa e clique no cadeado no pé da janela para a autenticação (pode usar a sua conta de usuário normal). Na coluna central, clique em `users`; selecione o usuário para o



qual a alteração do *shell* padrão terá efeito. Vá até o campo *Shell*, troque `/bin/tcsh` por `/bin/bash` e dê `Enter`. Saia do programa, confirmando a modificação.

Opção 2 – Mude o *shell* padrão através do Terminal. Abra Preferences, aba *Shell*. Assinale a opção “Use this shell” e preencha com `/bin/bash` em lugar de `/bin/tcsh`.

6 Com o usuário certo, abra uma nova janela do Terminal para verificar se o **bash** é realmente o novo *shell* padrão, ou seja, se uma nova janela abre direto no **bash**.

Como adicionar atalhos ao seu bash

1 Vá para o seu diretório `Home`:

```
cd ~
```

2 Se você não estiver utilizando o **bash** como seu *shell* padrão, faça isso com o `.bashrc`. Ou, se você executou os passos anteriores, faça isso com `.bash_profile`.

```
touch .bashrc
```

3 Abra o `.bashrc` em um editor de texto:

```
vi .bashrc
```

4 Adicione o seguinte:

```
if [ "$PS1" ]; then
# ativar suporte a cores nas listas de diretórios
#e ' val 'dircolors'
#alias ls='ls --color=auto
# aliases interessantes para experimentar
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'
alias dir='ls'
#alias vdir='ls --color=auto --format=long'
# prompt personalizado
PS1='\u@\h [\w]\$ '
fi
```

Para salvar, dê `Esc` seguido de `:w` `Enter`, e para sair do programa, `:q` `Enter`. (Se o `vi` lhe parece muito alienígena, use o `pic`, que mostra menus de comandos no pé da tela.) Para experimentar as várias opções incluídas, é só tirar e colocar o sinal de grade `#` no começo das linhas. As linhas começadas por esse sinal são ignoradas pelo sistema. **M**