

Quem tem medo do WebObjects?

Parte 3:
Direct to Java

Na segunda parte deste artigo, você viu como criar uma aplicação WebObjects cliente/servidor que roda num browser de Web sem a necessidade de incluir uma única linha de código. Vamos começar esta matéria criando uma aplicação bem similar, mas agora usando uma outra interface.



por Tiago Ribeiro e Fabio Ribeiro

Poucas pessoas sabem, mas o WebObjects também é capaz de criar aplicações desktop – ou seja, programas que não rodam em um browser, mas ainda assim se beneficiam das vantagens de serem aplicações *three tiered* (ver a parte 1, na Macmania 90). Essas aplicações são 100% “Pure Java”, o que quer dizer que podem rodar virtualmente em qualquer plataforma que suporte a especificação Java 2, Standard Edition (J2SE) – incluindo aí os sistemas Linux, Solaris e Windows 2000, entre outros. O maior benefício em relação a aplicações Web está no fato de que aplicações desktop fornecem melhor interatividade e uma interface mais próxima da oferecida por programas tradicionais, usados no dia-a-dia. A maior desvantagem está no fato de que as atualizações da aplicação devem ser enviadas e instaladas para todos os usuários, o que pode tornar seu uso inadequado em algumas situações.

Na parte 2 desta matéria, nós tínhamos criado uma interface Web para um banco de dados de contatos. Nesta parte, iremos um passo além e criaremos uma segunda interface para esse banco de dados – uma aplicação Java carregável a partir do Desktop. Da mesma forma que no exemplo anterior, criaremos essa aplicação sem adicionarmos código algum; para isso, utilizaremos uma tecnologia presente no pacote do WebObjects, deno-

minada **Direct to Java Client**.

Uma aplicação Direct to Java Client é composta por duas partes: uma delas roda dentro do servidor de aplicações e não possui nenhuma interface gráfica – a sua responsabilidade é gerenciar o acesso ao banco de dados e a lógica de negócios da aplicação em si; a outra é o cliente Java propriamente dito, que roda na máquina do cliente e se comunica com o servidor de aplicações através de uma tecnologia Java denominada RMI (Remote Method Invocation). Uma das vantagens de se utilizar esse método é que, estando a lógica da aplicação sempre centralizada em um único local, pode-se criar diferentes tipos de interface para ela sem a necessidade de se repetir código. Ou seja, o mesmo software que pode ser acessado internamente em uma empresa por aplicações-cliente nativas (por exemplo, uma aplicação Cocoa usando EOF) também pode ser disponibilizado para clientes externos, através de uma interface Web ou clientes Java multiplataforma. É possível até mesmo integrar uma aplicação WebObjects a telefones móveis com acesso à Internet – compatíveis com tecnologia WAP – através de produtos de terceiros.

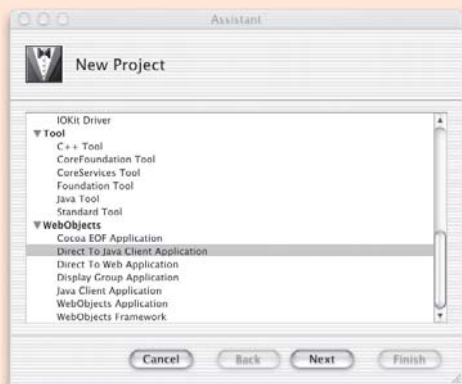
Vamos, agora, deixar a teoria de lado e começar a criar nosso cliente Java. Usaremos alguns recursos e idéias das partes 1 e 2 deste artigo, publicadas nas Macmanias 90 e 91. Por isso, se você não possui uma destas revistas, não perca tempo: compre já a sua!

WebObjects continuação

Iniciando o Direct to Java Client

1 Abra o aplicativo Project Builder, que fica em `/Developer/Applications/Project Builder.app`

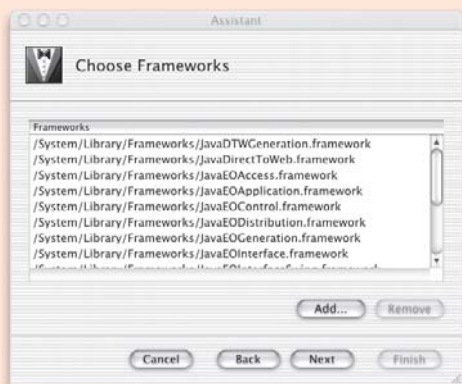
2 Selecione no menu File ► New Project para criar um novo projeto. Na lista que surge, selecione a opção WebObjects ► Direct To Java Client Application e então clique no botão Next.



3 Na janela seguinte, dê um nome para a aplicação (não digite espaços!). Usarei D2JClient. Depois disso, clique em Next.



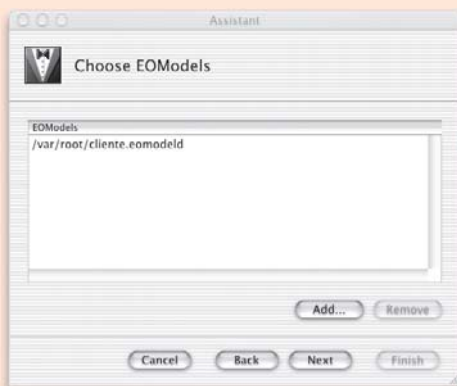
4 Na próxima tela ("Choose Frameworks"), você não precisará adicionar nenhum framework. Clique em Next.



5 Como nós criaremos uma interface Java para a mesma aplicação que desenvolvemos

MacPRO•52

na edição anterior, iremos aproveitar o mesmo modelo que criamos no EOModeler. Na janela "Choose EOModels" será pedido para que você adicione o modelo que descreve o banco de dados com o qual sua aplicação trabalha. Na parte anterior desta matéria, o banco de dados que usamos chamava-se cadastro. Clique em Add e navegue pelo disco rígido até encontrar o modelo correspondente. Encontrando-o, selecione-o e clique no botão Choose. Tendo selecionado o modelo, clique no botão Next.



6 Na última janela do assistente ("Build and Launch Project"), desmarque a opção "Build and launch application now" e então clique em Finish.

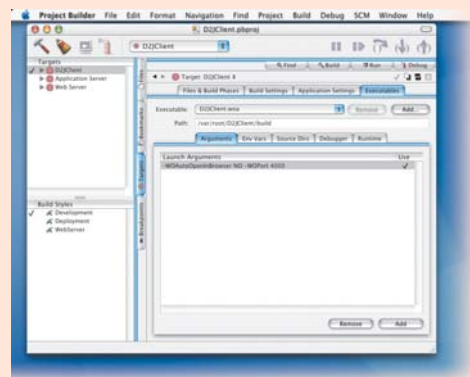


Configurando, compilando e executando

Quando você construir e rodar este projeto que acabamos de configurar pelo Assistente, você estará executando apenas uma das partes de nossa aplicação: aquela que é executada pelo servidor de aplicações e que é responsável pela integração com o banco de dados e pela sua lógica de negócios. Ou seja, estamos executando a parte *não-visível* da aplicação. O Project Builder é configurado para abrir automaticamente um navegador da Web no momento em que roda o projeto. Como nós não iremos utilizar uma interface Web para essa aplicação, convém mudar a configuração para que ela não o faça. Além disso, iremos também configurar a partir de qual porta

TCP/IP a aplicação estará disponível quando for executada.

7 Para configurar a aplicação, tendo o projeto aberto, clique em Targets; selecione D2JClient na lista que aparece na lista de Targets (canto superior esquerdo) e então clique na aba Executables. Mais cinco abas serão exibidas.



Clique na primeira: Launch Arguments. Clique no botão Add (canto inferior direito). Isso adicionará parâmetros que serão usados na abertura da aplicação. Digite:

```
-WOAutoOpenInBrowser NO -WOPort 4000
```

Esses parâmetros indicam que o navegador não será carregado ao se rodar a aplicação (WOAutoOpenInBrowser NO) e que a porta onde a aplicação estará disponível para nosso cliente Java será a 4000 (-WOPort 4000).

8 Pressione **⌘R** para compilar e rodar sua aplicação. Nada de excepcional aparecerá.

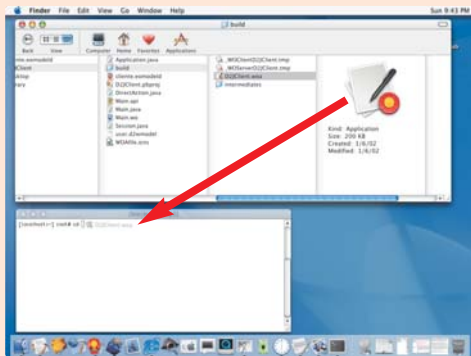
Executando o cliente Java

Vamos agora executar a *aplicação-cliente* – a aplicação Java que o WebObjects criou para nós e iremos distribuir para nossos clientes.

9 Abra o Terminal. Ele está dentro de `/Applications/Utilities/Terminal.app`

10 Através do Finder, encontre o lugar onde você salvou o seu projeto. A sua terminação é `.pbproj`. Na pasta `build` criada ao lado do seu projeto (no meu caso, é `D2JClient.pbproj`), existe um arquivo chamado `D2JClient.woa`, que é o seu aplicativo completo (inclui o software que roda no servidor de aplicações e o cliente Java).

11 No Terminal, digite `cd` seguido de um espaço e arraste o arquivo `D2JClient.woa` para a janela do Terminal. Isso irá fazer com que o nome de caminho completo para o seu aplicativo seja adicionado à linha de comando. No nosso caso, o comando vai ficar assim:
`cd /private/var/root/D2JClient/build/D2JClient.woa`
Pressione Return para ir ao diretório do `D2JClient.woa`.

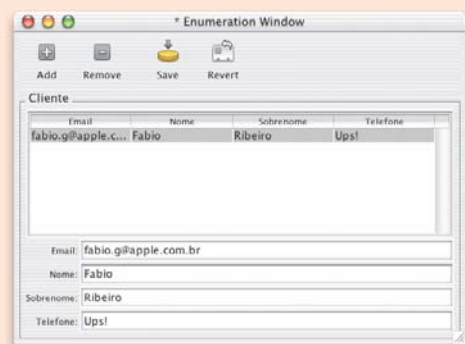


12 Digite `cd Contents/MacOS` para navegar até a pasta onde se encontra o executável do nosso cliente Java. Pressione Return.

13 Finalmente, digite:

`./D2JClient_Client "http://127.0.0.1:4000"`

Em alguns segundos, sua aplicação Direct To Java Client se abrirá (veja a figura abaixo para ter uma idéia do *look and feel*).

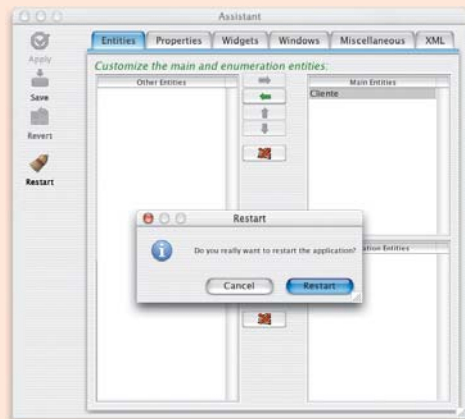


Customizando a aplicação

Essa aplicação está programada para obter todos os dados da tabela Clientes. É possível customizá-la, definindo quais tabelas aparecerão para nossos clientes, quais itens poderão ser modificados, como serão feitas as buscas etc. Vamos fazer isso agora:

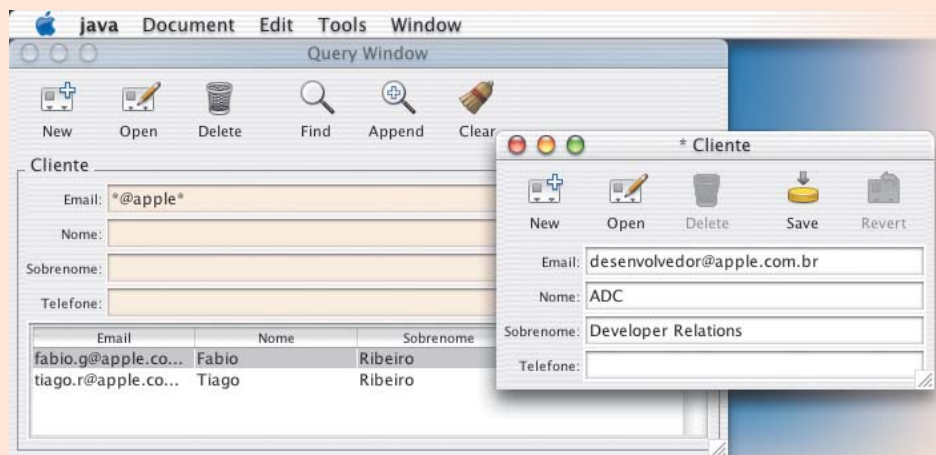
1 Selecione o menu Tools ► Assistant.

2 Na janela do assistente que aparece, clique na aba Entities. A tabela Cliente deve estar aparecendo na lista Enumerations Entities



(no canto inferior direito da janela). Selecione-a e clique uma vez na seta que indica o sentido à esquerda (essa seta está posicionada no grupo de botões inferiores dessa janela) para que ela seja movida para a lista Other Entities. Selecione-a nessa nova lista e clique na seta apontando para a direita (nos botões superiores), movendo-a para a lista Main Entities (no canto superior esquerdo).

3 Clique no botão Apply para aplicar as modificações e depois clique em Save.



4 Finalmente, clique no botão Restart e confirme sua intenção de reiniciar a aplicação. Note (acima) que outros botões aparecem e a sua aplicação agora suporta buscas.

O formato usado para a criação dessa interface é o XML. As aplicações mais modernas (e aí incluímos nossa aplicação Direct to Java) estão começando a usar esse formato para descrever os mais diferentes tipos de informação – sejam essas informações páginas Web, documentos a serem impressos, ou mesmo a interface de um programa. A aba XML do Assistant (ao lado) permite que você veja o código XML que o WebObjects usa para descrever a interface dessa aplicação em particular.

Empacotando e distribuindo

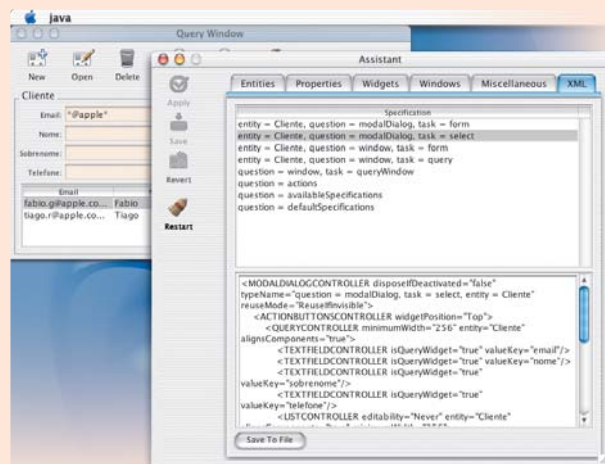
Neste exercício, nós executamos nossa aplicação Java a partir de um terminal Unix. Na prática, é esperado que você crie uma interface mais “humana” (ou seja, um executável duplo-clicável).

Existem várias maneiras de fazer isso; sugerimos aos interessados que leiam o texto “Wrapping Unix Commands” no endereço www.cocoadevcentral.com/tutorials/showpage.php?z=00000017.php

A vida não é feita apenas de templates...

OK. Até agora, estivemos trabalhando com templates que facilitam nossa vida e ajudam bastante na criação de aplicações. Usamos eles até agora porque queríamos mostrar quais tipos de aplicação o WebObjects é capaz de criar e as diferentes ferramentas envolvidas no processo.

Na prática você irá perceber que, embora facilitem sua vida, esses templates apresentam um limite em relação ao que pode ser custo-



mizado. Em um determinado momento, você precisará escrever mais código manualmente para que a sua aplicação seja exatamente do jeito que você quer. Mas, mesmo nesses casos, o WebObjects possui um conjunto grande de frameworks que tornam tarefas complexas bem simples. Veremos como criar essas aplicações na próxima edição. Nos vemos lá! **M**

TIAGO RIBEIRO
tiago.r@apple.com.br
 FABIO RIBEIRO
fabio.g@apple.com.br
 Trabalham na ADC da Apple Brasil.