

# Quem tem medo do WebObjects?

Parte 4:  
WebObjects  
Applications

Até agora, só trabalhamos com o WebObjects através de templates que não necessitam nenhum código adicional. A nossa principal intenção com isso foi dar uma idéia de como o WebObjects funciona (três layers), dar uma idéia de como os Frameworks que ele disponibiliza podem tornar a vida do programador mais fácil e mostrar que tipos de aplicações podem ser desenvolvidos com ele. Mas esta é a MacPRO, certo? OK! Feche os olhos; vamos colocar a mão na graxa, digo, código. Caso você não saiba Java, sugerimos livros de referência (alguns gratuitos) no fim do artigo.



por **Tiago Ribeiro e Fabio Ribeiro**

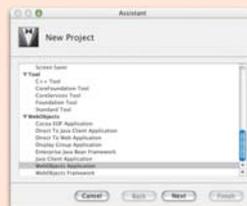
No primeiro exemplo, vamos mostrar uma aplicação onde você escreve um nome e um endereço de email e, ao clicar num botão, esses itens são exibidos numa linha abaixo, como numa saudação.

Isso permitirá entender conceitos básicos, como o relacionamento entre itens da sua página Web e variáveis no seu código, como chamamos uma ação ao clicar em botões etc.

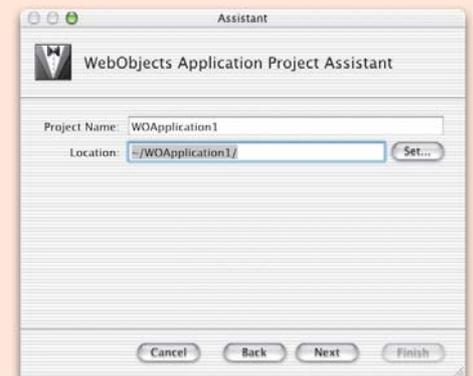
A seguir, corrigiremos alguns erros de design cometidos no primeiro exemplo, permitindo que, a cada vez que o usuário clique no botão, apareça uma nova linha com as informações recém-fornecidas. Pegue uma xícara de café e ligue seu Mac!

## Aplicação 1 Criando a Aplicação

**1** Crie um novo projeto no Project Builder, da mesma maneira que nas edições anteriores, mas desta vez o seu tipo deve ser WebObjects/WebObjects Application.



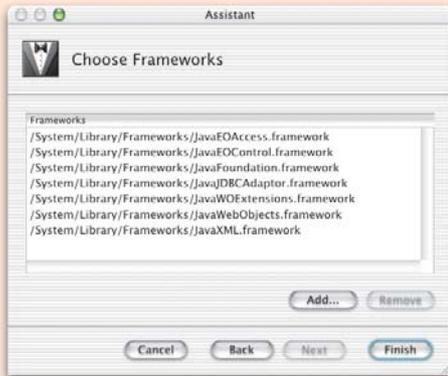
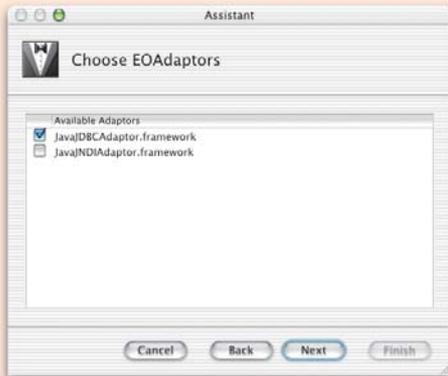
**2** Dê um nome – escolhi WOApplication1 – para o projeto. Salve-o num lugar onde possa achá-lo sem usar o Sherlock ;c)



**3** Na versão 5.1, é possível ou não integrar a sua aplicação WebObjects a um servidor J2EE. Se for questionado sobre “Integração J2EE”, deixe os dois quadrados de opção desmarcados; clique em “Next” ao ser perguntado sobre drivers JDBC, bem como Frameworks. Finalmente, clique em ‘Finish’.



# WebObjects continuação



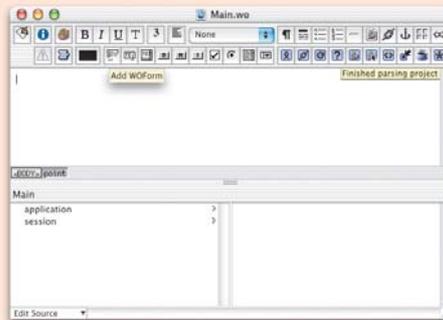
**4** Quando surgir a janela do projeto, abra a aba “Web Component Resources” e preste atenção aos itens que estão dentro dela. Vamos aproveitar e entrar um pouco na teoria. Um *componente* WebObjects é o nome que damos a uma página Web ou parte dela. O componente que é criado por *default* em cada projeto tem o nome de *Main* e corresponde à página inicial exibida para o usuário. Para cada componente são criados, entre outros elementos:

- Um pacote chamado *NomeDoComponente.wo* – Um arquivo que abre com o WebObjects Builder e que usaremos a seguir.
- *NomeDoComponente.java* – Contém o código Java que controla esse componente.
- *NomeDoComponente.api*

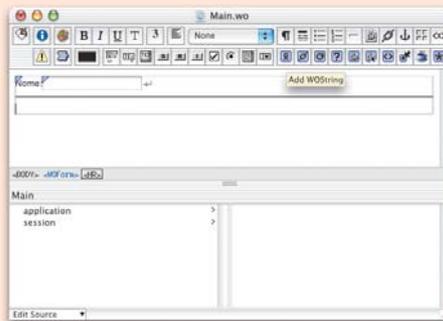
**5** Dê um duplo clique na “pasta” *Main.wo*. O WebObjects Builder será aberto para que você edite seu componente.

## Adicionando um formulário

**6** No WOBuilder, clique no botão para criar um *WOForm*, que corresponde a um formulário HTML para envio de dados.



**7** Escreva nome: dentro do formulário e adicione um *WOTextField*, que corresponde a um campo de texto.



**8** Dê **(Return)** depois do campo de texto e adicione a seguir um *WOSubmitButton*, que é um botão de “submit”.

**9** Adicione uma linha separadora para melhorar um pouco a interface da sua aplicação.

**10** Insira um *WOString* para exibir uma *string* (cadeia de caracteres).

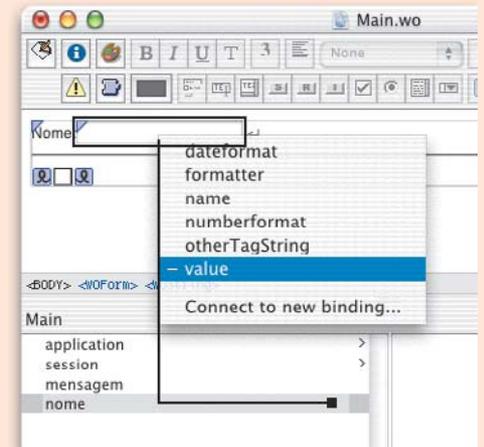
**11** Agora você vai inserir duas *keys* (chaves). Uma *key* corresponde a uma variável dentro do seu código que serve para conectá-lo a um elemento de interface dentro dos componentes de sua aplicação Web. No canto inferior esquerdo da janela do WO-



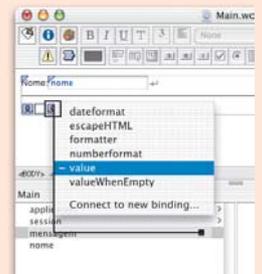
Builder, clique no texto “Edit Source”, o que irá fazer com que um menu apareça. Selecione, neste menu, a opção “Add Key”; na janela que surge, chame

essa *key* de nome e escolha como tipo *String*. Repita o processo, criando agora uma *key* com o nome *mensagem*.

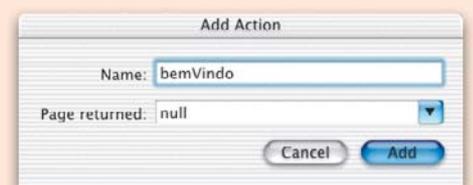
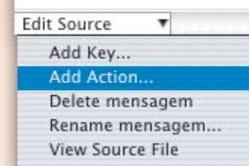
**12** Devem surgir os nomes *mensagem* e nome na coluna à esquerda (abaixo) da janela do componente. Clique no item “nome” e, com o mouse, arraste-o até o campo de texto, o que irá produzir uma linha preta de conexão.



Ao soltar o mouse sobre o campo de texto, irá surgir um menu; selecione a opção “value”, indicando que o valor desse campo de texto será conectado a essa *key*. Siga o mesmo procedimento para *mensagem*, ligando-o desta vez à *String* (*WOString*), de acordo com a figura ao lado.



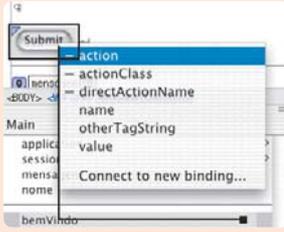
**13** Agora vamos criar uma ação e conectá-la a seu botão. Clique no mesmo menu a partir do qual você criou as *keys*, mas desta vez selecione a opção “Add Action”. Deve surgir uma tela como esta:



Dê o nome *bemVindo*. Escolha *null* como página de retorno. Isso faz com que a mesma página à qual essa ação está inserida seja recarregada. Clique em “Add”. Da mesma forma que você conectou as *keys*,



conecte agora a ação `bemVindo` com o botão – no menu que surge ao soltar o mouse sobre o botão, selecione a opção “action”.



**14** Feche a janela e salve todas as alterações. Volte ao Project Builder e clique em `Main.java`. Sim, várias coisas novas surgiram:

- Duas variáveis do tipo `String` (`nome` e `mensagem`), que correspondem aos itens de nomes correspondentes aos criados no componente.
- Um método novo: `bemVindo()`, criado por você ao editar o componente. Edite esse método, alterando-o da seguinte forma:

```
public WObject bemVindo()
{
    mensagem = "Bem-Vindo(a) " + nome;

    return null;
}
```

**15** Construa e rode a aplicação. Escreva um nome e clique no botão. Tente isso várias vezes.

## Aplicação 2

OK, você aprendeu algumas coisas básicas, mas importantes, sobre o WebObjects. Mas a aplicação ainda está simples demais.

**1** Crie uma nova aplicação, como fizemos há pouco, incluindo um `WForm`, dois campos de texto e os botões de “Submit” e “Reset”.

**2** Selecione na lista de arquivos do projeto a pasta `Classes` e, no menu, `File` ► `New File`. Selecione a opção `WebObjects/Java.class` e salve-o com o nome `Pessoa.java` no nosso projeto. Essa classe representará uma pessoa na sua aplicação.

**3** Edite o código para que fique igual a essa listagem:

```
// Pessoa.java
// WApplication2
//
```

```
// Created by Fabio Ribeiro on Thu Feb 07 2002.
// Copyright (c)2002 Apple. All rights reserved.
//
```

```
import com.apple.cocoa.foundation.*;
import com.apple.cocoa.application.*;
```

```
public class Pessoa {
    protected String    nome;
    protected String    email;

    Pessoa()
    {
        this.setNome("");
        this.setEmail("");
    }

    public String    nome()
    {
        return nome;
    }

    public String    email()
    {
        return email;
    }

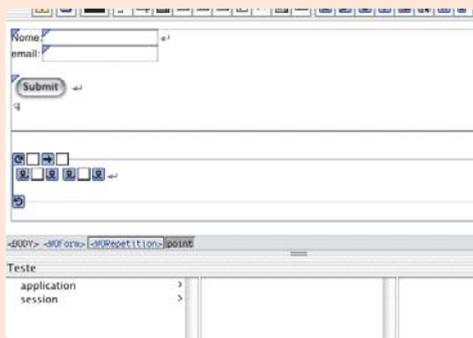
    public void setNome(String novoNome)
    {
```

```
{
    nome = novoNome;
}
public void setEmail(String novoEmail)
{
    email = novoEmail;
}
}
```

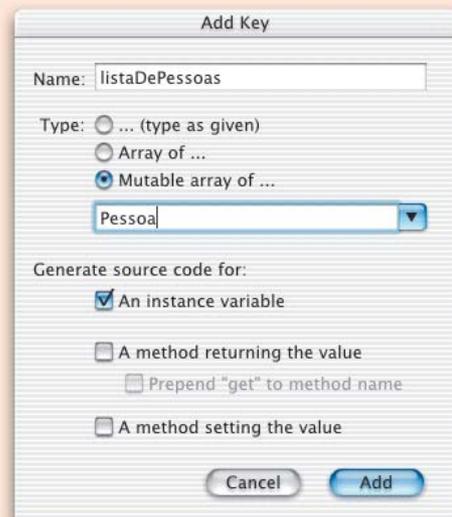
**3** Edite seu componente Main como fizemos há pouco, para que fique semelhante à figura:



**4** Adicione, abaixo do separador, um `WORepetition`. Coloque dentro desta repetição duas `WOStrings` separadas por um espaço. Não se esqueça de dar um **(Shift)Return** no final do último deles; fazendo isso, cada item aparecerá numa linha.



**5** Crie duas `keys` do tipo `Pessoa`, que correspondem a objetos da classe `Pessoa` que você acaba de criar: chame uma de `aPessoa` (o objeto que será editado pelo usuário) e a outra de `umaPessoa` (um dos itens da lista de pessoas). Crie ainda uma `key` do tipo `NSMutableArray` com o nome `listaDePessoas`.

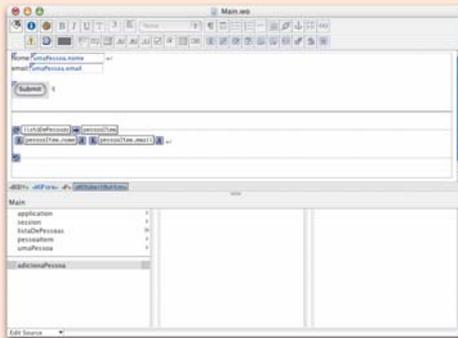


Você ligará o campo de texto `nome` a `aPessoa.nome` e o campo `email` a `aPessoa.email`.

**6** Crie um `action` chamado `adicionaPessoa`, retornando `null`. Ligue-o ao botão "Submit".

**7** Ligue agora a `NSMutableArray` `listaDePessoas` ao primeiro quadradinho da `WORepetition`. Isso indica que as informações da repetição estarão guardadas dentro dessa matriz.

**8** Ligue, agora, a chave `umaPessoa` ao segundo quadradinho da `WORepetition`; isso indica que as informações de cada linha desta repetição estarão guardadas dentro dessa variável. Ligue as propriedades `nome` e `email` de `umaPessoa` às `WOStrings` que estão dentro da `WORepetition`.



**9** Finalmente, edite a sua classe `Main.java` para que ela fique semelhante ao código listado a seguir:

```
//
// Main.java: Class file for WO Component 'Main'
// Project WOApplication2
//
// Created by caffeine on Thu Feb 07 2002
//

import com.webobjects.foundation.*;
import com.webobjects.appserver.*;
import com.webobjects.eocontrol.*;
import com.webobjects.eoaccess.*;

public class Main extends WOComponent {
    protected Pessoa umaPessoa;
    protected Pessoa aPessoa;

    /** @TypeInfo Pessoa */
    protected NSMutableArray listaDePessoas;
    public Main(WOContext context) {
        super(context);

        aPessoa = new Pessoa();
        //Instanciar objeto 'Pessoa'
        listaDePessoas = new NSMutableArray();
        //Instanciar lista de Pessoas
    }

    public WOComponent adicionaPessoa()
```

```
{
    listaDePessoas.addObject(aPessoa);
    //Adiciona uma nova pessoa à lista de pessoas

    aPessoa = new Pessoa();
    //Cria um novo objeto Pessoa em branco, para ser
    editado

    return null;
}
}
```

**10** Salve, compile e execute a sua aplicação.

## Além do elementar

Trabalhar com aplicações mais sofisticadas, como aquelas que fazem acesso a bancos de dados, é o próximo passo para quem entendeu os conceitos vistos até aqui. Se você está ansioso e deseja saber mais informações sobre como fazer isso, um bom ponto de partida é o documento em PDF *Discovering WebObjects for HTML*, disponível para download gratuito no site da Apple (<http://developer.apple.com/techpubs/webobjects>). Este é o assunto sobre o qual falaremos na próxima parte deste artigo introdutório. Até lá! **M**

TIAGO RIBEIRO

[tiago.r@apple.com.br](mailto:tiago.r@apple.com.br)

FABIO RIBEIRO

[fabio.g@apple.com.br](mailto:fabio.g@apple.com.br)

Trabalham na ADC da Apple Brasil.

## Referência básica: Java

### Livros

#### Java 2 Exam Cram

Bill Brodgen - Ed. Coriolis

Onde comprar: Amazon ([www.amazon.com](http://www.amazon.com))

#### Java In a Nutshell

David Flanagan - Ed. O'Reilly

Onde comprar: Tempo Real ([www.temporeal.com.br](http://www.temporeal.com.br))

#### Java Examples In a Nutshell

David Flanagan - Ed. O'Reilly

Onde comprar: Tempo Real ([www.temporeal.com.br](http://www.temporeal.com.br))

#### Thinking In Java

Bruce Eckel

Onde comprar: site do autor ([www.bruceekel.com](http://www.bruceekel.com))

### Sites

#### Sun – Java Developer Connection

<http://developer.java.sun.com>

#### TheServerSide.com J2EE Community

[www.theserverside.com](http://www.theserverside.com)

# Metendo a mão no Unix

## Parte 6: editor vi, lição 1 (inserindo texto)

Toda configuração em um sistema Unix (e, por tabela, no Mac OS X) é feita editando-se arquivos de texto. Logo, é natural que o programa utilizado para isso seja um processador de texto. O *shell* do Mac OS X já vem com um processador de texto embutido, o *vi*, essencialmente igual a qualquer outro programa do gênero, se bem que fãs do Word vão se decepcionar por não poder utilizar o mouse nem ter um monte de menus e barras de ferramentas. Antes de começarmos a trabalhar no *vi*, é interessante conhecer um aspecto muito importante. O *vi* é um editor *modal*. Isso significa que uma mesma tecla pode exercer diferentes funções, dependendo do modo no qual estiver trabalhando. Por exemplo, se você estiver no modo de *edição* e teclar **[A]**, adicionará a letra A ao texto; mas se estiver no modo de *comando*, teclar **[A]** fará com que o aplicativo entre em modo de edição.

**Importante:** Se em algum momento você esquecer em que modo está trabalhando, pressione a tecla **[Esc]** do seu teclado. Ela o levará sempre para o modo de comando.

### Passo 1

Utilizando o Terminal, digite o comando *vi*:



A tela ficará limpa; o cursor estará do lado esquerdo superior da tela, e o primeiro caractere de cada linha será um til (~).



### Passo 2

Digite o caractere dois pontos (:). O cursor irá para a parte inferior da tela. Em seguida, aperte a tecla **[Q]** e em seguida **[Return]**.



Com isso, você saiu do *vi* e voltou ao *prompt* do Terminal.

### Passo 3

Agora que sabemos entrar e sair do *vi*, vamos editar nosso primeiro arquivo. Digite:

```
cd ~/Desktop
vi demo
```



Você acaba de criar no seu Desktop um arquivo de texto chamado *demo* e verá no

Terminal a mesma tela do Passo 1, com o pé da janela indicando *demo: new file: line 1*. A expressão *new file* indica que esse é um arquivo recém-criado.

### Passo 4

Estamos prontos para inserir nosso texto, e o *vi* está aguardando nosso comando. Existem diferentes comandos para edição de texto; aqui vão alguns:

<b>i</b>	insere texto no arquivo
<b>a</b>	anexa texto ao arquivo
<b>o</b> (minúsculo)	abre uma linha abaixo da atual
<b>O</b> (maiúsculo)	abre uma linha acima da atual

Comece dando o comando **a** e em seguida digite **macintosh**:



Utilizando a tecla **[Delete]**, mova o cursor até que ele esteja sobre a letra **m** no início da palavra que foi digitada. Pressione **[Esc]** para deixar o modo de edição e retornar ao modo de comando. Observe que todo o texto digitado desapareceu. Isso ocorre porque o *vi* salva apenas o texto inserido à esquerda ou acima do cursor. Como retornamos o cursor até o início da palavra, ao sairmos do modo de edição o texto foi desconsiderado e apagado.

### Passo 5

Volte ao modo de edição e digite o seguinte texto (tirado do livro “O Jeito Macintosh: A Arte da Guerrilha no Gerenciamento”, de Guy Kawasaki):

```
saõ como eunucos num harem
viram fazer todos os dias
mas saõ incapazes de fazer eles mesmos.
```

Propositalmente, estão faltando algumas coisas no texto; precisaremos corrigi-las.

### Passo 6

Entre a primeira e a segunda linha, precisamos adicionar uma frase. Poderíamos entrar no modo de edição, ir até a segunda linha, inserir a frase que falta e teclar **[Enter]**.

Mas vamos utilizar um comando capaz de abrir uma linha abaixo da linha atual.

Vá até a primeira linha do texto e digite o comando **o**:



Agora digite o texto que falta:

```
saõ como eunucos num harem
|
viram fazer todos os dias
mas saõ incapazes de fazer eles mesmos.
```

Aperte **[Esc]** para voltar ao modo de comando.

por **Alberto V. Mendonça**

### Passo 7

Agora precisamos incluir o nome do autor do texto; vamos colocá-lo acima da primeira linha. Para isso, seguiremos os mesmos procedimentos que fizemos anteriormente, mas utilizando o comando **O** para abrir uma nova linha acima do texto. Digite, nessa linha:

```
Bredan Behan:
```

Aperte **[Esc]** para voltar ao modo de comando.

### Passo 8

Temos um último problema no texto. Falta uma palavra no início da segunda linha. Posicione o cursor no início da segunda linha e utilize o comando **i** para inserir a palavra que falta: **criticos**

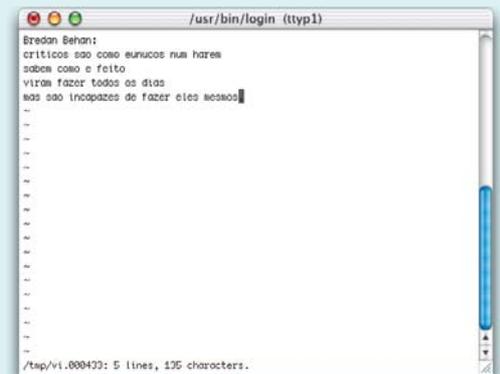


### Passo 9

Volte ao modo de comando e tente sair do *vi* utilizando o comando **:q**. Você receberá uma mensagem lembrando que ainda não salvou seu arquivo de texto:



Precisamos salvar nosso arquivo antes de sair do *vi*. Para isso, digite o comando **:w** e o *vi* passará a indicar algumas informações sobre o arquivo.



Agora você pode sair do *vi*, utilizando o comando **:q**. Caso queira sair sem gravar as alterações, utilize o comando **:q!**.

**Atenção!** Tome bastante cuidado para não abrir arquivos de configuração do seu sistema e alterá-los acidentalmente; isso pode danificar o funcionamento do seu Mac. **M**

ALBERTO V. MENDONÇA