

[< VOLTAR](#)

# O uso de servlets

Capacitar o aluno na utilização de servlets, acrescentando mais esta tecnologia na integração com JSP e banco de dados para dar ainda maior flexibilidade ao desenvolvimento de aplicações dinâmicas na web.

## NESTE TÓPICO

- > Introdução
- > Servlets
- > Implementando servlet
- > Servlet na prática
- > Entendendo a aplicação



## Introdução

Cada aula depende de algum conceito de aulas anteriores, portanto, caso tenha alguma dificuldade, provavelmente o assunto já tenha sido abordado em aula anterior e lhe passou alguma coisa despercebida. Nesse caso, recomenda-se que volte e revise as aulas que forem necessárias.

## Servlets

Conforme visto na aula 4, entendemos que os servlets são pequenos programas em Java executados em um servidor web, dentro de um contêiner que tem uma infraestrutura de software necessária para entender requisições realizadas por clientes, no qual algum tipo de processamento pode ser realizado e, ainda, devolvida a resposta ao cliente (SANTOS, 2007, p.34).

## Implementando servlet

Aproveitando o exemplo da aula anterior, nesta aula, ele será melhorado com a ajuda do servlet, que nos dá maior flexibilidade de trabalhar com classes e banco de dados no lado servidor.

Embora possa ser feito tudo em JSP, Teruel (2009, p.97) nos esclarece que a tecnologia JSP permite a criação de aplicações dinâmicas, além de suportar diversas outras tecnologias juntas, como HTML, XHTML, CSS, JavaScript, WML, etc.

No entanto, a utilização de servlets e javabeans, por exemplo, permite fazer uma separação mais profissional do código, permitindo, assim, uma maior facilidade no momento de uma alteração. Com isso, também está sendo separada a parte visual (que passa a ficar em JSP) da parte de controle e de dados (que fica com os servlets e javabeans).

## Servlet na prática

Nas aulas anteriores discutimos vários conceitos aqui aplicados. Retomando da última aula, lembremo-nos de que nosso projeto era a construção de um mural de notícia, onde o primeiro registro de uma tabela é a notícia e os demais registros são seus comentários.

Conforme figura a Figura 1, observamos que no projeto da aula anterior havia alguns arquivos JSP para lidar com a interface com o usuário, uma classe DAO para tratar o acesso a dados e uma classe denominada "Mural" (javabean) para servir de armazenamento de dados da aplicação.

Todo controle do fluxo de execução é realizado através da chamada direta dos respectivos códigos escritos em JSP.

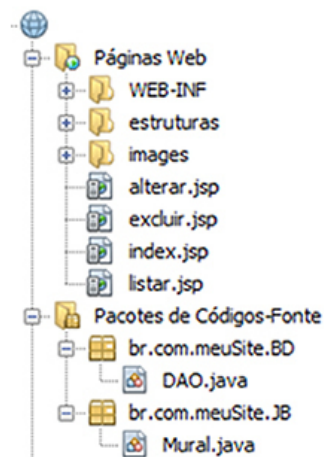


Figura 01

No projeto desta aula (Figura 2), observamos o acréscimo de alguns arquivos JSP, mas a principal mudança se dá na estruturação dos pacotes, definidos dentro de um domínio chamado de **br.com.meuSite**.

Foram criadas as classes Controle.java e Suporte.java, sendo que a primeira é um servlet que servirá para controlar as solicitações e respostas da aplicação, enquanto a segunda é uma classe com métodos genéricos à aplicação.

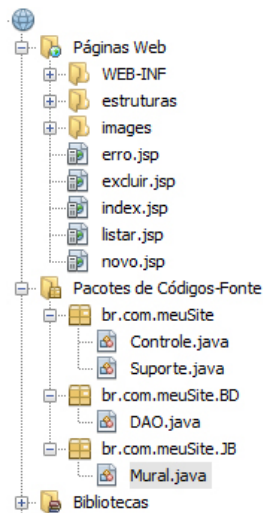


Figura 02

## Entendendo a aplicação

Antes de nos aprofundarmos em códigos de programação, vamos entender o que é e como funciona a aplicação apresentada:

A aplicação já apresentada na aula anterior é um pequeno gerenciador de notícias, que pode ser adaptado para funções semelhantes.

Ao ser executada, a aplicação irá, de início, checar a existência do banco de dados e das tabelas necessárias. Caso não existam, serão criados automaticamente pela própria aplicação.

Para visualizar o diretório do banco de dados Java DB, abre-se a janela de serviços e se clica em "Propriedades de Java DB". Por padrão, o diretório do banco de dados é **.netbeans-derby**.

Exemplo: C:\Users\aaas\.netbeans-derby

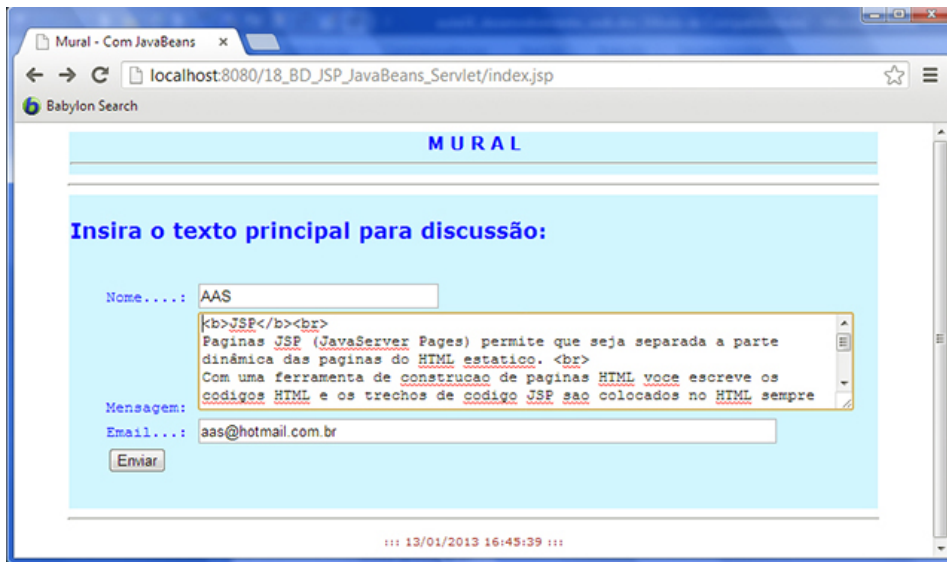
Para que o banco seja criado automaticamente, foi colocado o atributo `create=true` em sua url de abertura. (ORACLE, 2013, p.263).

Exemplo: url = "jdbc:derby://localhost:1527/mural;create=true";

Caso seja necessário criar um link com o banco de dados para manutenção das informações através do próprio NetBeans, veja a seguir o **que o conduzirá passo a passo**.

Depois das devidas checagens, o browser exibe sua primeira tela:





Na primeira execução, será apresentada a tela para inclusão do texto principal. Após isso, as próximas inclusões serão de comentários para o texto principal:



## Referências

ORACLE. *Java DB reference manual*: version 10.3. Disponível em: [docs.oracle.com](http://docs.oracle.com/javadb/10.3.3.0/ref/refderby.pdf) (<http://docs.oracle.com/javadb/10.3.3.0/ref/refderby.pdf>). Acesso em: 12 jan. 2013.

SANTOS, Rui Rossi dos. *Java na web*: programando sites dinâmicos. Rio de Janeiro: Axcel Books, 2007.

TERUEL, Evandro Carlos. **Web Total:** desenvolva sites com tecnologias de uso livre: prático e avançado. São Paulo: Érica, 2009.



Avalie este tópico



ANTERIOR

O uso de JavaBeans e JSP com banco de dados



Índice

(<https://www.uninove.br/conheca-biblioteca>)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Ajuda?

Próximo  
(<https://ava.uninove.br/ava/curso/gerenciamento-de-aplicacoes-com-cpf>)

Gerenciamento de aplicações com CPF

© Todos os direitos reservados

