

[< VOLTAR](#)

Gerenciamento de aplicações com WAR

Compreender as questões que envolvem gerenciamento, arquivamento e distribuição das aplicações desenvolvidas através de arquivos WAR e, para finalizar, algumas dicas de documentação de software com Javadoc.

NESTE TÓPICO

- > Criando arquivo WAR com NetBeans
- > Javadoc
- > Referências



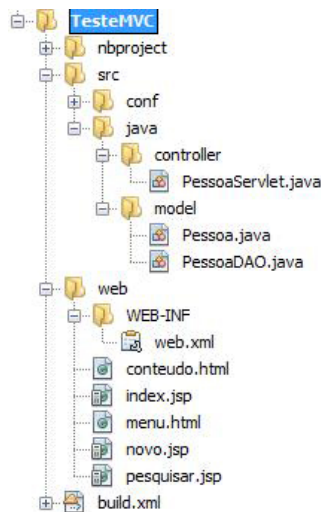
Criando arquivo WAR com NetBeans

Uma aplicação pode ser distribuída ou publicada na web por meio de arquivos WAR (*Web archive*), que nada mais são que arquivos compactados (como os gerados por utilitários como o *Winzip*).

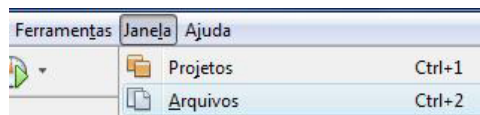
Segundo Kurniawan (2002, p. 389), "em um arquivo WAR, você empacota cada arquivo em um aplicativo. Normalmente, o nome do arquivo WAR é o mesmo nome do aplicativo; porém, você pode usar o nome que quiser".

Para melhor entendermos como isso funciona, vamos criar um arquivo WAR para a aplicação da aula anterior.

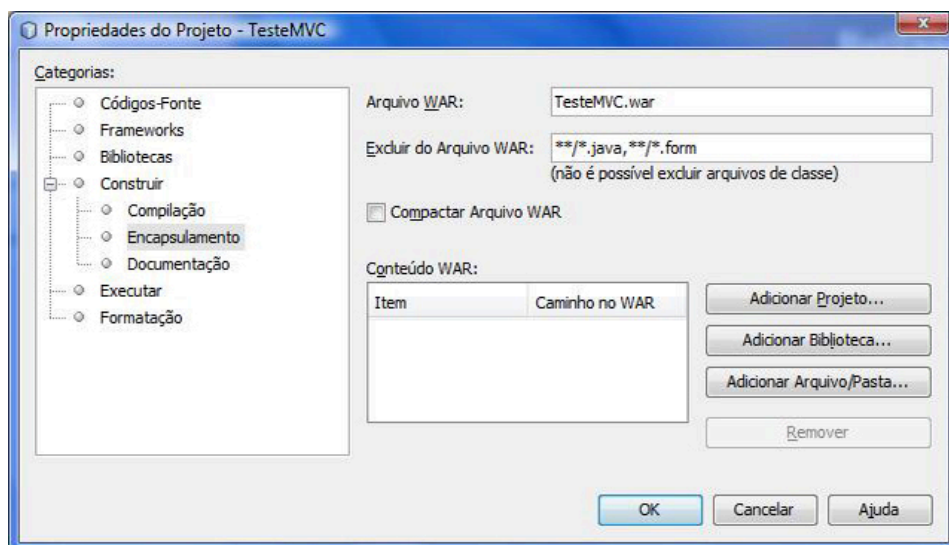
Vejamos, na figura a seguir, a estrutura da aplicação (de nome TesteMVC):



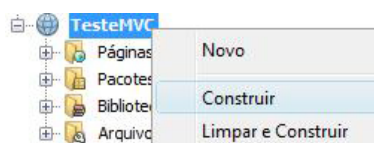
Para visualizar esta estrutura, você pode acessar a opção "Janela" > "Arquivos" no menu do NetBeans:



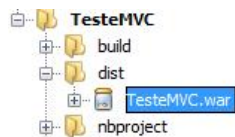
Caso seja necessário alterar alguma configuração para a criação do arquivo WAR a ser criado, pode-se entrar nas propriedades do projeto e acessar a seguinte janela:



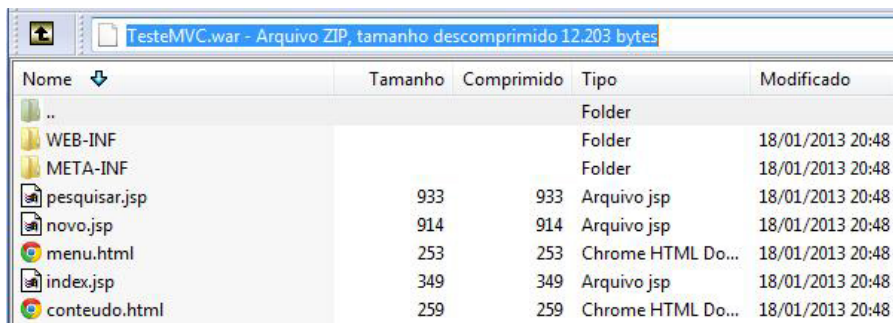
Para que o arquivo WAR seja criado com as configurações definidas com toda aplicação como conteúdo, basta reconstruir a aplicação:



Observamos que o arquivo compactado foi criado:



O arquivo gerado poderá ser aberto, inclusive, por um aplicativo de compactação de arquivos:



O arquivo TesteMVC.war contém toda a aplicação, ou seja, tudo que é necessário para que ela funcione, exceto informações de banco de dados, que geralmente ficam mais simples de ser embutidas, no caso de arquivos de dados como o Access (MDB).

Depois de ter um arquivo WAR, você o distribui sob o diretório webapps. Então, é possível acessar o seu aplicativo exatamente como faria com um aplicativo não empacotado. O nome usado para o seu aplicativo empacotado é o nome de arquivo .war.

((KURNIAWAN, 2002, p. 389))

Veja um exemplo de publicação em um provedor de hospedagem de sites utilizando o EATJ (2013).





Você pode procurar outros provedores para efetuar seus testes, testar com outras aplicações ou, ainda, criar seu próprio ambiente.



Fugindo um pouco do tema da aula, mas um assunto interessante para finalizarmos o conteúdo desta disciplina, a seguir, uma breve explanação sobre os comentários em programas.

Javadoc

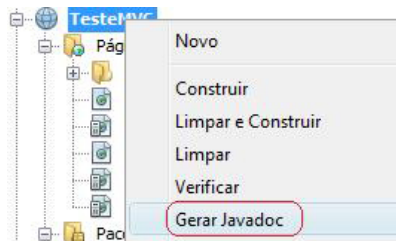
A questão de se comentar códigos de programas acaba gerando, às vezes, algum tipo de polêmica. Uns comentam demais (coisas muito óbvias), outros não comentam nada.

Os dois extremos da polêmica do assunto são perfeitamente compreensivos: comentários óbvios apenas poluem o código desnecessariamente, só são aceitáveis em um ambiente de aprendizagem; caso o programador necessite de informações sobre as regras de negócio, funcionalidades de componentes, estruturas de tabelas em bancos de dados, entre outras informações, sempre deve se recorrer à documentação do sistema.

Conforme a definição do site da Oracle (2013), o javadoc é uma ferramenta utilizada para a criação de documentação em formato HTML através dos comentários de documentação do código-fonte.

Uma das grandes vantagens de se gerar a documentação de uma aplicação é a geração de um documento que pode ser compartilhado entre as equipes envolvidas no projeto de alguma forma.

Para gerar a documentação via NetBeans, é só clicar com botão direito sobre o projeto:



Trechos de comentários colocados entre `/**` e `*/` no código fonte são capturados pelo javadoc:

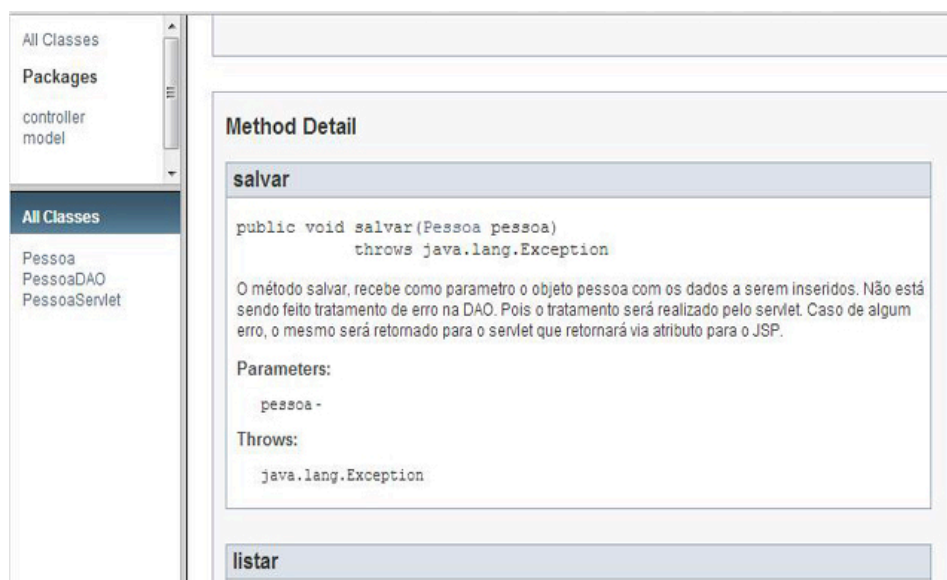
```

79  /**
80   * O método salvar, recebe como parametro o objeto pessoa com os dados a
81   * serem inseridos. Não está sendo feito tratamento de erro na DAO. Pois o
82   * tratamento será realizado pelo servlet. Caso de algum erro, o mesmo será
83   * retornado para o servlet que retornará via atributo para o JSP.
84   *
85   * @param pessoa
86   * @throws Exception
87   */
88  public void salvar(Pessoa pessoa) throws Exception {
89      // Este método recebe o objeto preenchido pelo JavaBeans, via parametro:
90      if (pessoa.getNome().equals("")) {
91          throw new Exception("Pelo menos o NOME precisa ser preenchido!");
92      }

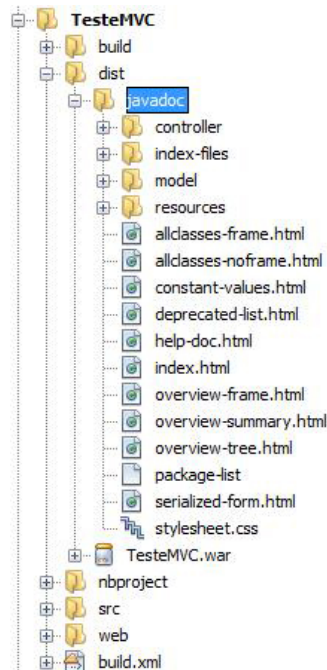
```



Exemplo de como ficou o arquivo gerado pelo javadoc:



Por fim, vejamos que uma nova pasta foi criada na pasta de distribuição da aplicação, especificamente para armazenar toda documentação gerada:



Chegamos ao fim de mais uma aula e da disciplina. No entanto, caso seja necessário, revise as aulas, refaça os exercícios e os testes, invente exercícios, inove, pesquise na internet, veja vídeos sobre o assunto e leia livros. As referências colocadas no final das aulas servem, também, de indicação confiável de fonte de base para o estudo dos assuntos abordados na disciplina.

Referências



EATJ. *Web Hosting Solutions*. Disponível em: <www.eatj.com> (<http://www.eatj.com/>). Acesso em: 1º set. 2012.

KURNIAWAN, Budi. *Java para web com servlets, JSP e EJB*. Rio de Janeiro: Ciência Moderna, 2002.

ORACLE. *How to Write Doc Comments for the Javadoc Tool*. Disponível em: <[www.oracle.com](http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html)> (<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>). Acesso em: 1º set. 2012.



Avalie este tópico



ANTERIOR
O uso de servlets

Biblioteca
(<https://www.uninove.br/conhec-a->
a-



Índice

Ajuda?
(<https://ava.uninove.br/ava/uninove/idCurso=>)

® Todos os direitos reservados



uninove/biblioteca/sobre-
a-
biblioteca/apresentacao/)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

