

## Criação do projeto com JBoss Forge

### Downloads

Caso não queira gerar o projeto com JBoss Tools ou enfrente algum problema com a ferramenta basta baixar o projeto inicial [aqui](https://s3.amazonaws.com/caelum-online-public/auron/auron-stage1.zip) (<https://s3.amazonaws.com/caelum-online-public/auron/auron-stage1.zip>). Esse projeto pode se importado no Eclipse para seguir os primeiros passos nesse vídeo.

### Iniciando o projeto

Nosso objetivo nesse curso será implementar um projeto web para associar amigos e amigos ocultos, o domínio é o jogo amigo secreto. Para tal vamos criar um cadastro de participantes usando a interface web e persistir no banco de dados. Chamaremos o projeto de Auron.

No projeto Auron usaremos as tecnologias: JSF para a interface web, EJB e JPA para gerenciar a transação e persistir os dados e o CDI para juntar essas especificações. Como servidor usaremos o Wildfly que é o servidor JavaEE da JBoss e atende ao padrão JavaEE na versão 7.

O objetivo é criar uma aplicação real com um pouco mais de complexidade e usar as tecnologias estudadas na trilha. Este curso se destina aos alunos que já assistiram aos cursos anteriores e querem aplicar o conhecimento em um novo projeto.

### Servidor Wildfly e Eclipse com JBoss Tools

O primeiro passo é baixar o servidor de aplicações Wildfly da JBoss. Vamos acessar a página <http://wildfly.org/downloads> (<http://wildfly.org/downloads>) e baixar a versão 8.1. Escolhemos o arquivo ZIP para baixar. No nosso ambiente já temos o servidor baixado e podemos extrair o arquivo.

Após disso podemos iniciar o Eclipse. Usaremos a versão Luna, que já temos instalado. Aconselhamos a usar uma versão limpa do Eclipse JavaEE para que a instalação do plugin JBoss Tools seja feita sem problemas.

Uma vez iniciado iremos ao menu *Help* e escolhemos o item *Eclipse Market Place*. No campo de busca procuraremos por *JBoss Tools*, que é um conjunto de plugins que facilita o desenvolvimento JavaEE com Eclipse.

Assim que o Eclipse buscou os dados da pesquisa basta selecionar o item JBoss Tools na versão Eclipse Luna. Nas próximas telas temos que selecionar os plugins do JBoss Tools e confirmar a licença de uso. Todo o processo pode demorar alguns minutos e foi acelerado no vídeo. No fim da instalação devemos reiniciar o Eclipse.

### Usando JBoss Forge para criar o projeto

Após instalação do JBoss Tools vamos abrir o console do **JBoss Forge**. O Forge é uma das ferramentas do JBoss Tools que irá nos ajudar na hora de criar o projeto. Uma vez aberto o console é preciso iniciá-lo como se fosse um servidor. Isso faz que o console fique ativo e assim podemos interagir com ele.

O comando para criar o novo projeto pelo Forge é:

```
project-new --named auron
```

Isso faz que o Forge gere um esqueleto de um projeto dentro do Eclipse já na estrutura padrão do Maven com o arquivo `pom.xml` pré-configurado. Repare também que o prompt da console mudou e estamos agora dentro do projeto `auron`.

## Scaffolding com Forge

O próximo passo é gerar as configurações principais de um projeto JavaEE. O JBoss Forge possui para isso um comando específico:

```
scaffold-setup
```

Na saída podemos ver que foram *instalados* JPA, EJB, CDI entre outros. Infelizmente não foi instalado o JSF mas podemos mudar isso através do comando:

```
faces-setup --facesVersion 2.2
```

No projeto podemos ver que já foram adicionados vários arquivos como o `persistence.xml` e também os arquivos `beans.xml` e `faces-config.xml` do CDI e JSF, respectivamente.

## Primeira página

Agora criaremos uma página inicial simples para podermos checar se nosso projeto está rodando. Essa página deve ficar dentro da pasta `src/main/webapp`. Caso o JBoss Tools já tenha criado a página `index.xhtml` e a `index.html` removidas do projeto que criaremos com outro conteúdo logo adiante. Para criarmos a nossa página, vamos clicar com o botão direito sobre `src/main/webapp` e escolher o menu *New > Other > WEB > HTML Page*. Vamos chamar a página `index.xhtml` e ela será do tipo `xml 1.0 transactional`. Ao confirmar o diálogo uma página com um esqueleto é criada e podemos adicionar o namespace JSF. Como sempre, o namespace fica na tag `<html>`:

```
xmlns:h="http://java.sun.com/jsf/html"
```

Para testar a página vamos adicionar apenas um simples componente e imprimir algo na tela, nesse caso um `<h:outputText value="uma mensagem qualquer" />`.

## Rodando com JBoss Wildfly

Para rodar a aplicação falta instalar o Wildfly dentro do Eclipse. Vamos abrir a aba *Servers* para configurar o Wildfly. Basta criar um novo server e selecionar na lista o Wildfly 8.x. Na tela do *JBoss Runtime* precisamos definir a pasta da instalação do servidor que fizemos o download no começo desse capítulo. Após isso, podemos associar o projeto `auron` com o servidor.

Vamos rodar o Wildfly pelo Eclipse e acessá-lo através navegador. A URI é `http://localhost:8080/auron` para acessar a aplicação e `/faces/index.xhtml` para chamar a página criada anteriormente. Se não erramos algo no XHTML deve aparecer a mensagem no navegador. Ótimo, funcionou!

No próximo capítulo criaremos o primeiro cadastro dessa aplicação.

