

Módulo 1

O Demoiselle é uma composição de projetos, e não apenas um framework Java como muita gente acha. É fácil entender porque pensam assim: o Framework surgiu primeiro e até hoje é o carro-chefe. Então vamos deixar bem claro, o projeto Demoiselle é um agregado de subprojetos, que são:

- **Framework.** Este é o foco do curso. Vamos conhecer e praticar as facilidades que este projeto oferece para o desenvolvimento de aplicações Java baseado nas tecnologias da JEE 6. Definição em duas palavras: simplicidade e flexibilidade.
- **Component.** Conjunto de componentes reusáveis que dão mais poder ao Framework. Nem todos componentes estão atrelados ao Framework, mas os utilizados neste curso estão.
- **Infra.** Se você sente calafrios ao pensar na trabalhadeira que dá para preparar um ambiente de desenvolvimento, este projeto foi feito para você. Sim, vamos utilizá-lo neste curso.
- **Sample.** Está precisando de exemplos para entender melhor o Framework e os Componentes? Disponibilizamos algumas amostras para você. Neste curso não utilizaremos este projeto.
- **Process.** Trata-se de um processo livre de desenvolvimento de software inspirado em metodologias ágeis. Não utilizaremos este projeto neste curso.
- **Community.** Todos os projetos Demoiselle são livres e colaborativos. Este projeto promove a gestão transparente e a participação ativa da comunidade. Citaremos algumas ferramentas no decorrer deste curso.
- **Tools.** São ferramentas focadas em produtividade que podem ser usadas em conjuntos com o Framework. Especificamente a ferramenta Demoiselle Nimble será citada neste curso.

Objetivo

O objetivo deste curso básico do Demoiselle é mostrar, explicar e experimentar as principais funcionalidades do Framework, tais como:

- Testes
- Log
- Resource Bundle
- Tratamento de exceções
- Configuração
- Persistência
- Transação
- Controladores
- Inicializadores
- Controle de acesso
- Web
- Mensagem

Apoio

Ao final deste curso você será capaz de compreender o funcionamento do Demoiselle e utilizá-lo para construir suas aplicações. Para facilitar a sua vida, adicione nos favoritos do seu *browser*: <http://demoiselle.sourceforge.net/docs>. Neste link você encontrará dois guias de referência muito importantes: do Framework e dos Componentes. Recorra a eles sempre que precisar de detalhes.

Aproveite o embalo e cadastre-se na lista de usuários, ferramenta indispensável para quem quer mandar bem e se manter atualizado: <https://lists.sourceforge.net/lists/listinfo/demoiselle-users>. Além de tirar dúvidas básicas e avançadas, você poderá colaborar com outros usuários.

Infra

Chega de lero-lero e vamos para a prática, o primeiro passo é a preparação do ambiente. Se você utiliza uma distribuição Linux baseada no Debian, como o Ubuntu, não tem com o que se preocupar, o Demoiselle Infra vai ajudar muito nesta tarefa:

1. Acesse o menu Instalação no site do Infra: <http://demoiselle.sourceforge.net/infra>
2. Neste curso convencionamos utilizar a distribuição Linux via APT (Advanced Package Tool). Siga o passo-a-passo e instale o pacote **demoiselle-2-infra-1**.
3. No terminal, rode o comando: **demoiselle-workspace** para poupar tempo com configuração do Eclipse. Caso tenha outra instalação do Eclipse e já tenha um workspace anterior, informe um novo para usar durante o curso, com o comando: **demoiselle-workspace workspace_novo**
4. Abra o Eclipse no menu do sistema: Aplicativos / Desenvolvimento / Demoiselle Eclipse
5. Se criou um novo workspace, informe este na abertura do Eclipse.
6. Veja que o Eclipse já está todo configurado e você poupou bastante tempo. Ponto para o Demoiselle!

A distribuição para o sistema Windows é ofertada em dois arquivos, um contendo o ambiente de desenvolvimento e outro contendo outras ferramentas adicionais conforme explicitado em pacotes. Para instalar o ambiente de desenvolvimento, o usuário deverá realizar o download do arquivo de instalação na página de downloads (<http://sourceforge.net/projects/demoiselle/files/infra/>) e executá-lo. Escolha o arquivo que representa o seu tipo de sistema operacional. A instalação é autoexplicativa e o ambiente de desenvolvimento deverá ser instalado na pasta C:\Demoiselle\

Confira o vídeo de demonstração usando o Synaptic clicando no link abaixo:

<http://www.frameworkdemoiselle.gov.br/documentacaodoprojeto/manuais-e-tutoriais/tutorial-da-versao-2-2-3-0/videos/modulo-1-video-1-synaptic>

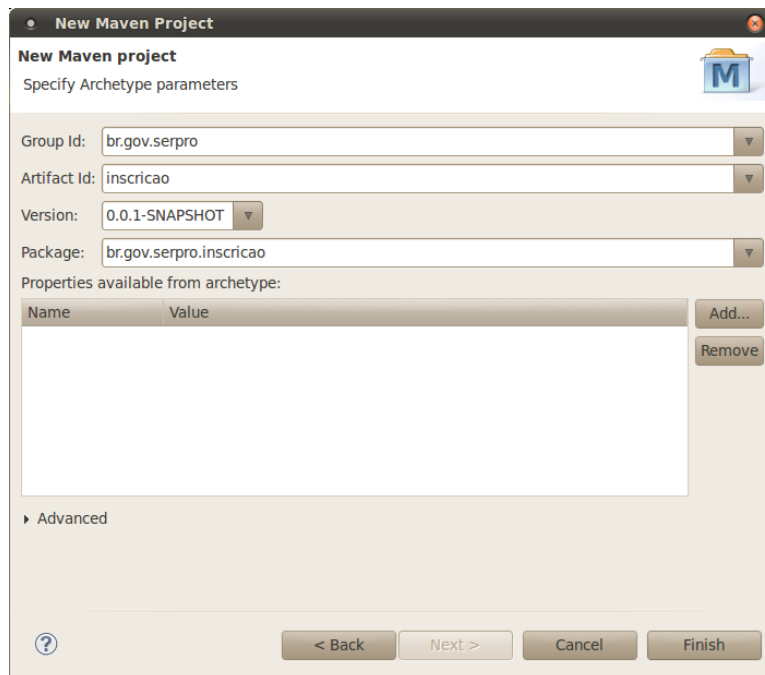
Confira o vídeo de demonstração usando linha de comando clicando no link abaixo:

<http://www.frameworkdemoiselle.gov.br/documentacaodoprojeto/manuais-e-tutoriais/tutorial-da-versao-2-2-3-0/videos/modulo-1-video-1>

Novo projeto

Agora que o ambiente está instalado, vamos criar o nosso primeiro projeto. É nele que experimentaremos as funcionalidades do Framework:

1. No Eclipse, acesse *File / New / Project*.
2. Agora vá em *Maven / Maven project* e clique em *Next* duas vezes.
3. Escolha o Catalog *Demoiselle*.
4. Na listagem, escolha o *demoiselle-minimal* e clique em *Next*.
5. Defina o Group Id: **br.gov.serpro**, o Artifact Id: **inscricao** conforme mostrados abaixo e clique em *Finish*.



6. Aguarde o Maven criar o seu projeto.

Você acabou de utilizar o Maven, uma ferramenta livre que auxilia na criação de projetos. Na realidade ele é muito mais que isto, porém é tudo que você precisa saber no momento. No catálogo *Demoiselle* estão todos os moldes de projeto que fizemos. O Maven denomina-os de *archetypes*, portanto utilizaremos o termo *arquétipos*. O *demoiselle-minimal* é o arquétipo que traz o básico para utilizar o *Demoiselle*.

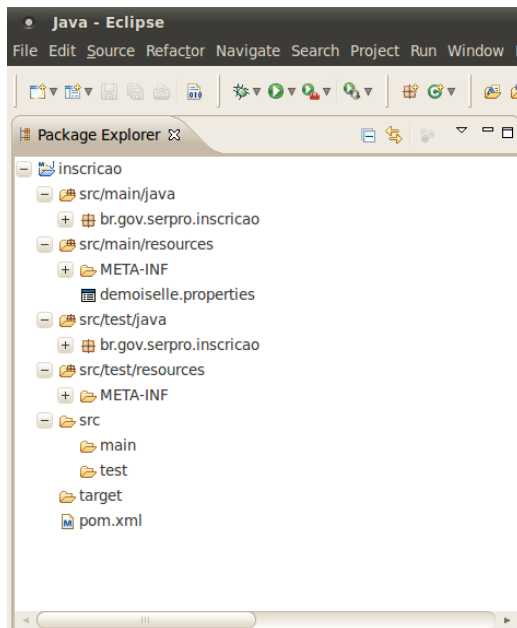
Confira o vídeo de demonstração clicando no link abaixo:

<http://www.frameworkdemoiselle.gov.br/documentacaodoprojeto/manuais-e-tutoriais/tutorial-da-versao-2-2-3-0/videos/modulo-1-video-2>

O que gerou?

No Maven tudo é artefato, inclusive este seu novo projeto. Todos os artefatos são identificados pelo Group Id acrescidos do Artifact Id, e foi isto que você informou para geração do seu projeto. Seguindo o padrão do Maven, seu projeto possui a seguinte estrutura:

- Pasta **src/main/java**: aqui ficarão todas as classes do seu projeto.
- Pasta **src/main/resources**: armazenará os arquivos de configuração de sua aplicação.
- Pasta **src/test/java**: coloque aqui suas classes de teste.
- Pasta **src/test/resources**: os arquivos de configuração dos seus testes ficarão aqui.
- Pasta **target**: esta é uma pasta temporária utilizada exclusivamente pelo Maven para armazenar as classes compiladas e arquivos de configuração processados. Você não precisa mexer nela.
- Arquivo **pom.xml**: configuração do Maven para o seu projeto. Geralmente este arquivo contém muitas informações, porém, com a utilização da herança de configurações, conseguimos torná-lo bem simples para você. Tudo graças ao *demoiselle-minimal-parent*. Abra o **pom.xml** e dê uma olhada. Simples, não?!



Desafio

Vamos incrementar o projeto gerado para criar uma aplicação de verdade. À cada módulo deste curso exploraremos algumas funcionalidades do Framework. O desafio é criar um sistema simples de inscrição com as seguintes regras:

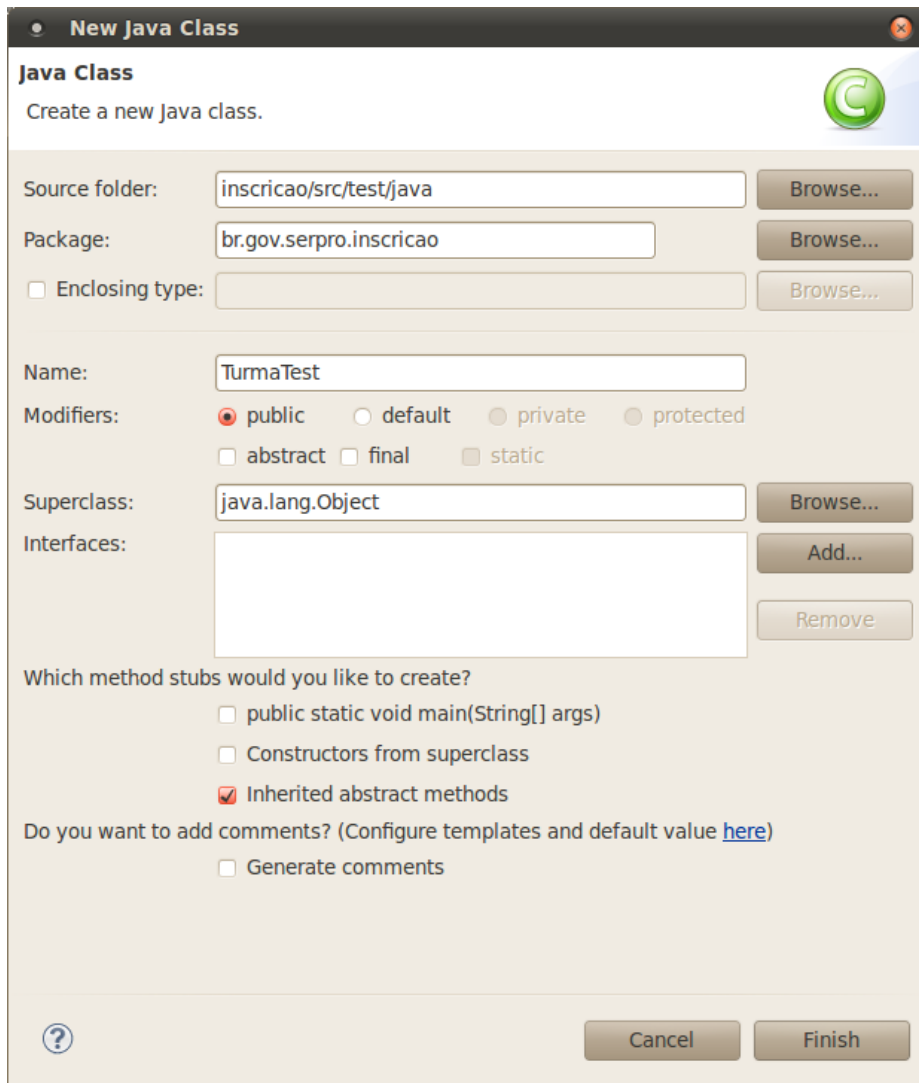
- Possibilitar a matrícula do aluno na turma.
- O aluno só pode matricular-se uma vez.
- A turma tem um limite de alunos matriculados.

Até o final do curso, você criará uma aplicação *web* utilizando JSF, JPA, JTA, CDI e diversas outras siglas. No momento apropriado você saberá o significado de cada uma delas. Um passo de cada vez, é assim que vamos evoluir.

Testes

Você já ouviu falar em TDD? É a sigla para *Test Driven Development*, algo como Desenvolvimento Orientado a (ou “Dirigido por”) Testes. Resumidamente, a ideia é criar os testes antes da solução propriamente dita (por isso, alguns usam o termo “Test First”). Em quase todos os módulos deste curso praticaremos o desenvolvimento orientado a testes. Então vamos converter as regras da nossa aplicação-desafio em casos de teste:

1. No Eclipse, abra a pasta *src/test/java* e selecione o pacote *br.gov.serpro.inscricao*.
2. Acesse o menu vá em *File / New / Class*.
3. Defina o nome da classe *TurmaTest* e clique em *Finish*.



4. Crie os métodos *public void matricularAlunoComSucesso()*, *public void falhaAoTentarMatricularAlunoDuplicado()* e *public void falhaAoTentarMatricularAlunoNaTurmaCheia()*.
5. Anote os métodos com *@Test*.

A sua classe vai ficar assim:

```
public class TurmaTest {  
  
    @Test  
    public void matricularAlunoComSucesso(){  
    }  
  
    @Test  
    public void falhaAoTentarMatricularAlunoDuplicado(){  
    }  
  
    @Test  
    public void falhaAoTentarMatricularAlunoNaTurmaCheia(){  
    }  
}
```

Confira o vídeo de demonstração clicando no link abaixo:

<http://www.frameworkdemoiselle.gov.br/documentacaodoprojeto/manuais-e-tutoriais/tutorial-da-versao-2-2-3-0/videos/modulo-1-video-3>

Retrospectiva

O Demoiselle é composto por diversos outros projetos. Vimos que o Framework provê várias funcionalidades que serão exploradas neste curso. Aprendemos como chegar na documentação de referência e onde pedir ajuda.

Preparamos o ambiente de desenvolvimento e criamos o projeto que será utilizado nos demais módulos deste curso. Descobrimos os artefatos gerados automaticamente pelo arquétipo *demoiselle-minimal* e vimos que o *pom.xml* do projeto ficou bem simples graças às facilidades providas pelo Demoiselle

Conhecemos o problema desafio e preparamos o terreno para consolidar os requisitos em forma de testes automatizados. No próximo módulo, utilizaremos o componente *demoiselle-junit* experimentando as funcionalidades de *log* e internacionalização da aplicação.