

# ROBOCODE + DROOLS

**SI - 2018.1**  
**Fernanda Castro**



# DROOLS

- Plataforma de integração de lógica empresarial, fundada em 2001.
- Open Source.
- Plugin para Eclipse.



# QUAL A MOTIVAÇÃO?

- Evitar “código espaguete”
- Evitar recompilar em casos de alteração pontual
- Modularização: separar infraestrutura e regras de negócio
- Reuso e fácil entendimento (o que nos leva a uma manutenção mais fácil)
- Desempenho

# EXEMPLO

```
package bank.model;
```

```
rule "basic rule"
```

```
    when // condition
```

```
        Account( balance < 100 )
```

```
    then // consequence
```

```
        System.out.println("Account balance is less than 100");
```

```
end
```

# ENTENDENDO O DROOLS

- Quando todas as condições de uma regra são satisfeitas, a regra é disparada
  - A execução de uma regra pode ativar novas regras
  - A ativação é feita segundo a estratégia de resolução de conflito
  - O processo só é terminado quando nenhuma nova regra é ativada
- 
- Documentação: <https://goo.gl/S7xOwp>

# ENTENDENDO O DROOLS

- Para criar uma variável: `$account : Account( $type : type )`

- Para criar uma função:

```
function double calculateSquare(double value) {  
    return value * value;  
}
```

# ENTENDENDO O DROOLS

- Condicionais:

And      Customer( name == "John", age < 26 )

Or        Customer( name == "John" || age < 26 )

Not       not Account( type == Account.Type.SAVINGS )

Exists   exists Account( type == Account.Type.SAVINGS )

- Também é possível trabalhar com coleções: member of / contains



COMO USAR?

# TUTORIAL PARA:

- Instalar o Robocode + Drools.
- Criar um projeto de Robô com Drools a partir de um exemplo.
- Importar o seu Robô com Drools para o Robocode.

# INSTALANDO ROBOCODE + DROOLS

- **1:** Baixe o .zip do Robocode+Drools no classroom.
- **2:** Descompacte o arquivo e execute o “droolsrobocode.bat”.
- **3:** Você terá o Robocode com as mesmas funcionalidades vistas anteriormente, porém com suporte ao Drools.

# CRIANDO UM ROBÔ COM DROOLS

- **1:** Baixe o .zip do exemplo no classroom.
- **2:** Descompacte, procure pelo arquivo “pom.xml” e edite o atributo “ArtifactID” para o nome do seu projeto.
- **3:** Abra o Eclipse, clique em: File > Import > Maven > Existing Maven Project.
- **4:** Indique o diretório onde você descompactou o exemplo e importe.
- **5:** Renomeie o pacote do projeto no Eclipse (clique com o botão direito sobre o projeto e escolha: Refactor > Rename).

# CRIANDO UM ROBÔ COM DROOLS

- **6:** Abra as classes “ComprobarReglas” e “RobotDrools”, altere o valor do atributo “Ficheiro\_Reglas” para “Nome\_do\_Projeto/reglas/reglas\_robot.drl”
- **7:** Abra o arquivo “reglas\_robot.drl” e altere o primeiro import para “import Nome\_do\_Projeto”
- **8:** Gere um .jar no Eclipse: botão direito sobre o projeto > Run As > 5 Maven build...
- **9:** Na tela seguinte, escreva o nome do projeto em “Name”, escreva “package” em “Goals”, marque a opção “Skip Tests” e clique em “Run”. Seu arquivo .jar deve surgir na pasta “target” do projeto.

# CRIANDO UM ROBÔ COM DROOLS

- Observações:
  - Nem todas as versões do Eclipse possuem o plugin do Maven. Aparentemente, ele está disponível apenas a partir da versão Luna.
  - Caso seu robô esteja num diretório cujo path possui caracteres especiais (incluindo espaços), o Robocode pode não ser capaz de importá-lo.
  - Este exemplo é apenas...um exemplo! Está escrito em espanhol e é bastante simples. Vocês podem utilizá-lo como base, podem criar outro projeto do zero, podem traduzir... Brinquem bastante para compreender o funcionamento das regras! ;)

# IMPORTANDO UM ROBÔ COM DROOLS PARA O ROBOCODE

- **1:** No Robocode, vá em “Robot > Import Robot” e escolha o arquivo .jar do seu robô.
- **2:** Abra o .jar num 7zip da vida, copie a pasta “Nome\_do\_Projeto” e cole na pasta “robots” dentro do Robocode.

\* Deu erro no passo 2? Tente copiar os arquivos .java e .class diretamente da pasta do seu projeto (workspace) para a pasta “robocode/robots/Nome\_do\_projeto”.

# ROBOCODE - EXERCÍCIOS



# EXERCÍCIO 1 - ROBÔ KAMIKAZE

- O robô permanece parado até que detecte um adversário. Então, move-se (juntamente ao canhão) em direção ao inimigo.
- Só atirar quando achar o inimigo.
- Defina o poder de fogo de acordo com a distância para o inimigo (a seu critério). Quanto mais longe, menor o poder de fogo.
- Se algum adversário morrer, printe algo na tela.

## EXERCÍCIO 2 - ROBÔ COVARDE

- Ao detectar um adversário, verifica o nível de energia do inimigo.
- Se o adversário possui energia maior que o dobro da sua, o robô “desiste”: muda sua cor para vermelho e fica andando para frente e para trás.

## EXERCÍCIO 3 - ROBÔ ARYA STARK

- O robô gira o radar 360 graus e se move pela arena (a seu critério).
- Ao ser atingido por um projétil, o robô deve adicionar o adversário à sua lista de inimigos.
- Caso um robô pertencente à lista de inimigos morra, ele deve ser retirado da lista.
- O robô deve estar sempre repetindo a lista atualizada para si mesmo (printe na tela).

# DÚVIDAS?

- Me manda um email!

**mfcc@cin.ufpe.br**

- Slides disponíveis no **classroom** da disciplina.

