

# Enterprise Java Beans

## Iniciando desenvolvimento

Prof. Dr. Alexandre L'Erario



# Ferramentas de build podem ajudar

- A criação de projetos podem seguir estruturas já definidas pelas ferramentas de build.
  - Permite que haja uma padronização mínima
  - Permite que outros softwares reconheçam o projeto
    - Teste e otimizações
- Permite o gerenciamento de dependências
  - Inclusive montar pacotes com ou sem dependências
- Permite o gerenciamento do ciclo de vida
  - Build, build+test, deploy, etc....

# Maven – Exemplo



- Estrutura “esqueletos” de projetos pré-definidos
- Exemplo: criar um projeto a partir de um archetype

```
mvn -B archetype:generate \  
-DarchetypeGroupId=io.github.hantsy \  
-DarchetypeArtifactId=maven-archetype-jakartaee9 \  
-DarchetypeVersion=1.0 \  
-DgroupId=com.example \  
-Dpackage=com.example.demo \  
-DartifactId=myapp \  
-Dversion=1.0-SNAPSHOT
```

## Exemplo – executar o projeto

```
# cd myapp
```

```
# mvn clean package cargo:run
```

- Download das dependências
- Download do servidor Glassfish
- Deploy e
- Execução

# Outros exemplos

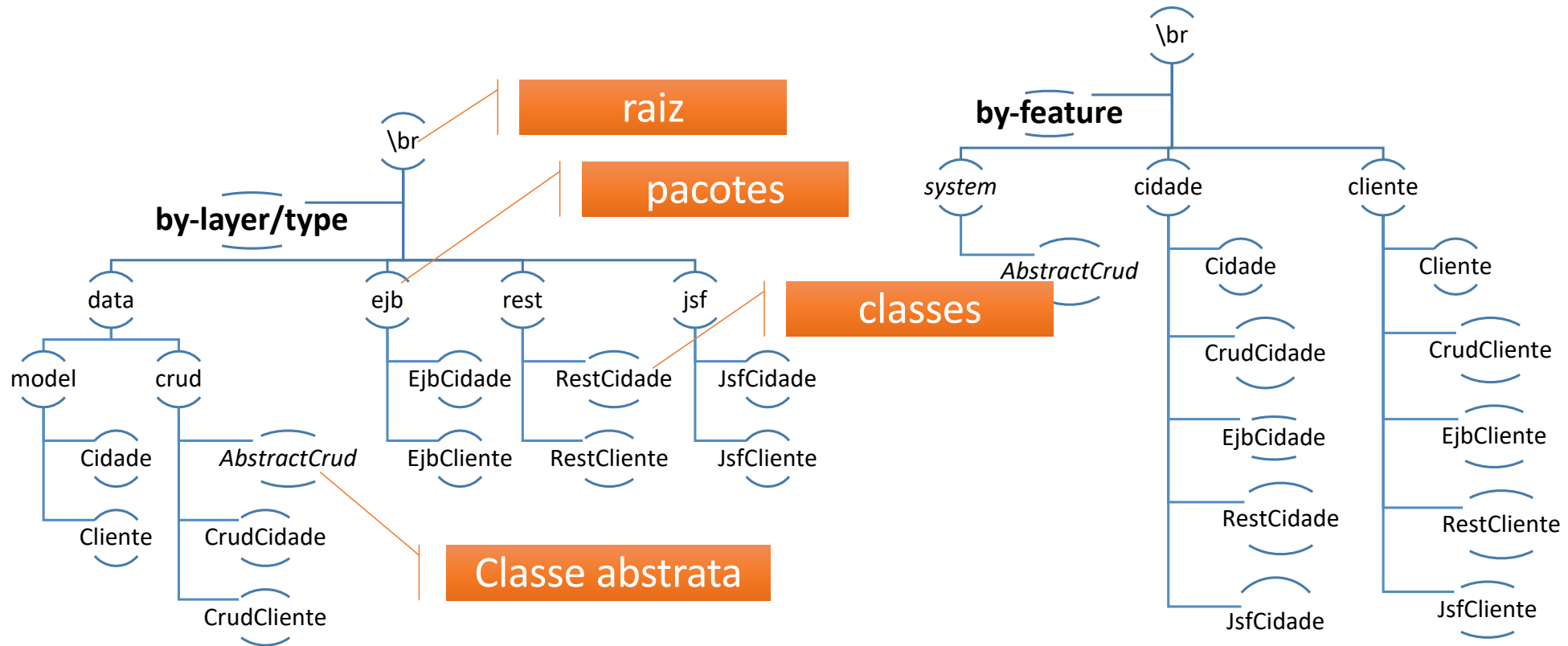
```
./mvn -B archetype:generate \  
-DarchetypeGroupId=io.github.hantsy \  
-DarchetypeArtifactId=jakartaee9-starter-boilerplate \  
-DarchetypeVersion=1.0 \  
-DgroupId=com.example \  
-Dpackage=com.example.demo \  
-DartifactId=myapp \  
-Dversion=1.0-SNAPSHOT
```

É possível criar seu  
próprio modelo também

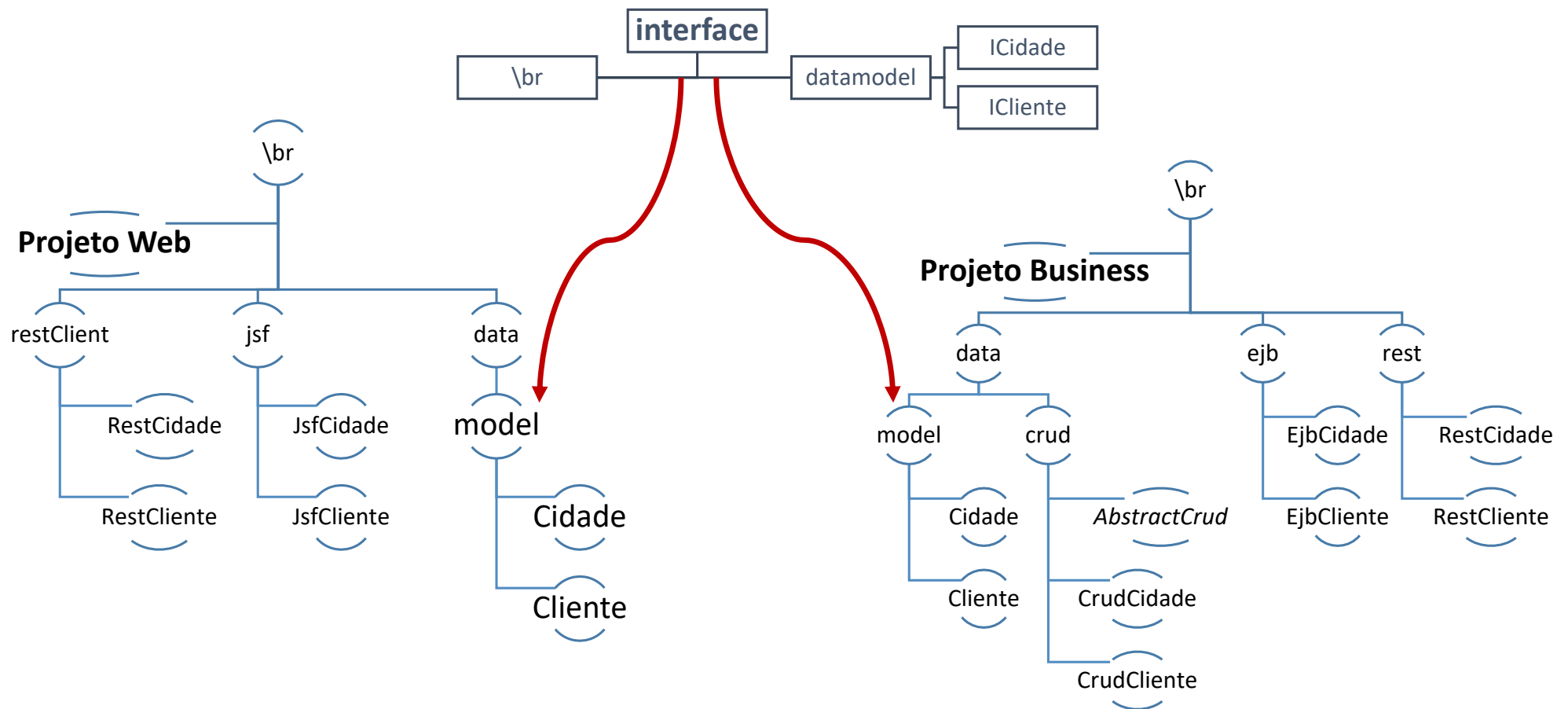
```
./mvn -DarchetypeGroupId=fish.payara.maven.archetypes -DarchetypeArtifactId=payara-micro-  
maven-archetype -DarchetypeRepository=https://repo.maven.apache.org/maven2 -  
DgroupId=com.mycompany -DartifactId=projeto1 -Dversion=1.0-SNAPSHOT -Dpackage=br -  
DpayaraMicroVersion=5.2021.10 -DautoBindHttp=true -DcontextRoot=/ -  
Darchetype.interactive=false -DarchetypeCatalog=local --batch-mode  
org.apache.maven.plugins:maven-archetype-plugin:3.1.2:generate
```

# Definindo estrutura básica

- Código fonte ➔ /src/java/main ou /src/java/test
- Definir convenção (pacotes, classes, serviços, etc)



# Estruturas mais complexas



## Considerações finais

- A ferramenta de build é crítica;
- Estruturar o projeto é uma tarefa crítica;
- As interfaces amparam a autonomia de projetos.