Desenvolvimento de Aplicações Java

Plataforma Corporativa

Tutorial

Implementação do Primeiro Componente Corporativo

Agosto 2015

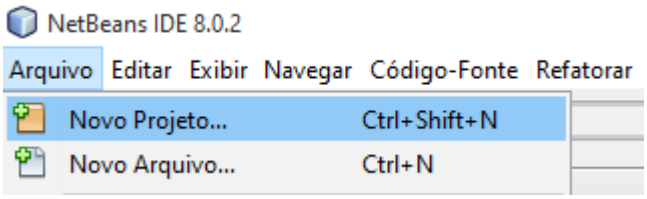
Sumário

1.Introdução

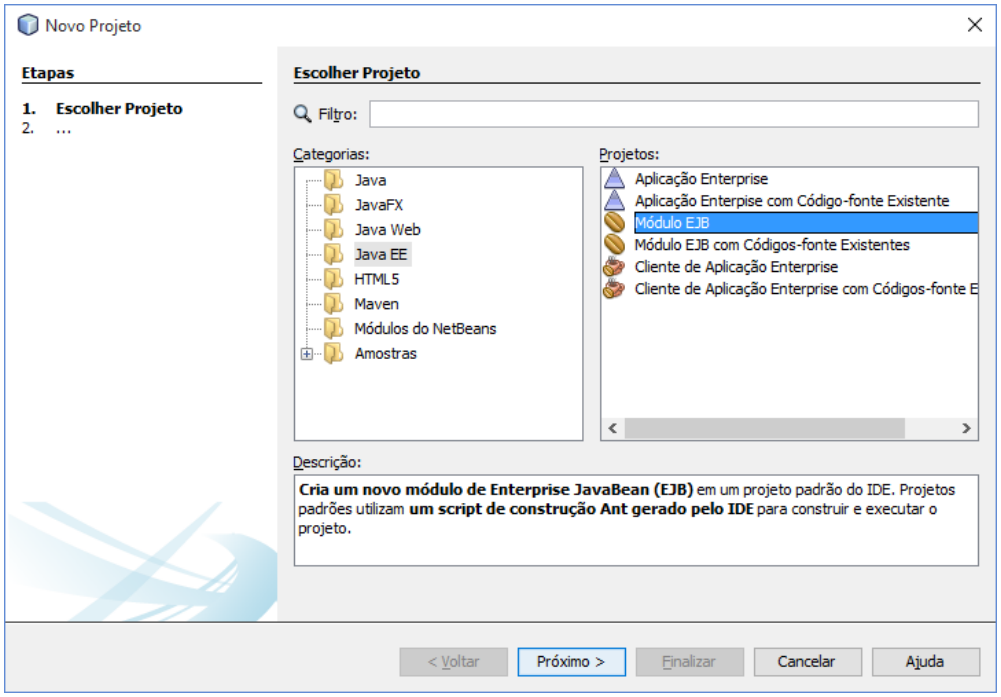
Neste documento os procedimentos para implementação no Netbeans do primeiro componente corporativo e sua implantação no JBoss Wildfly são apresentados.

2. Criação de um projeto para um módulo EJB

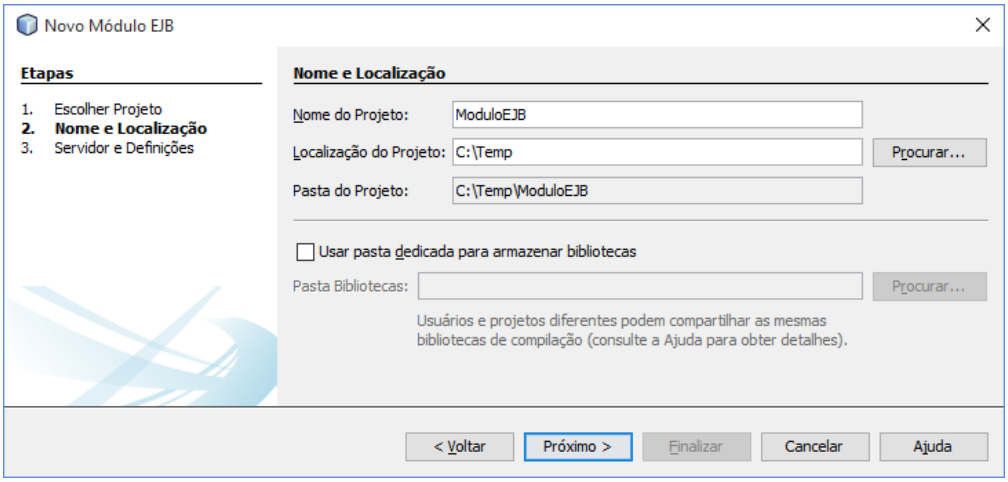
- Execute o netbeans.
- Na barra de comando selecione “Arquivo” -> Novo projeto.



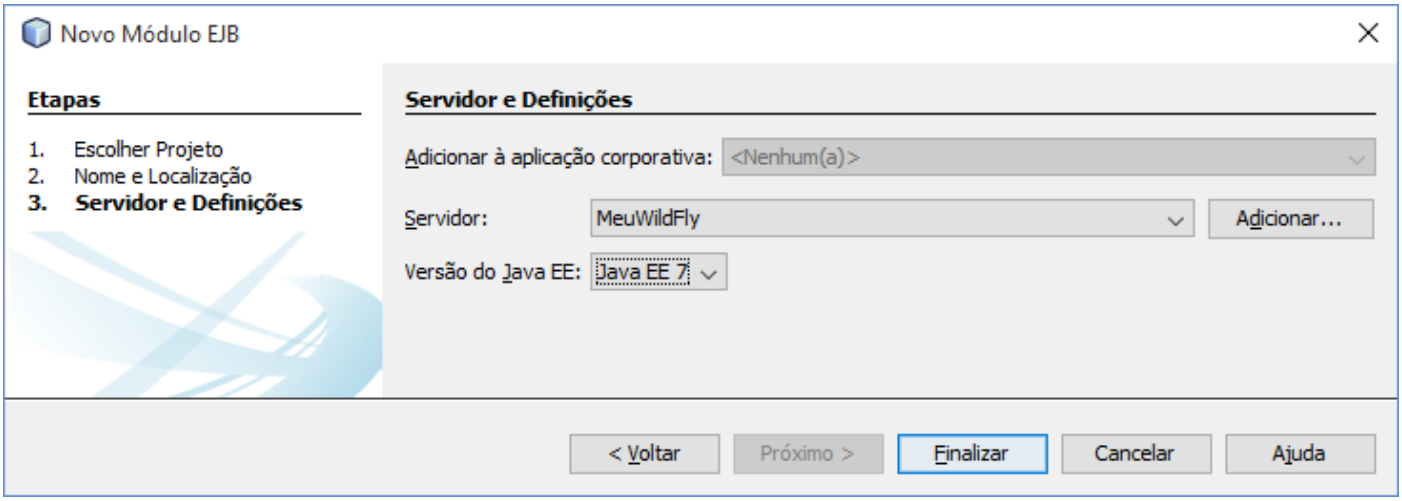
- Em categoria selecione “Java EE” e em Projetos, selecione “Módulo EJB”. Clique em próximo.



- De um nome para seu módulo e indique onde você deseja manter o projeto. Clique em próximo.



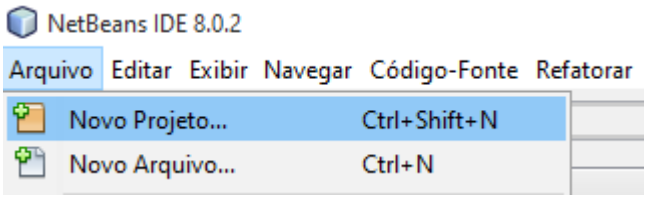
- Indique qual o servidor de aplicação para implantação. Clique em finalizar.



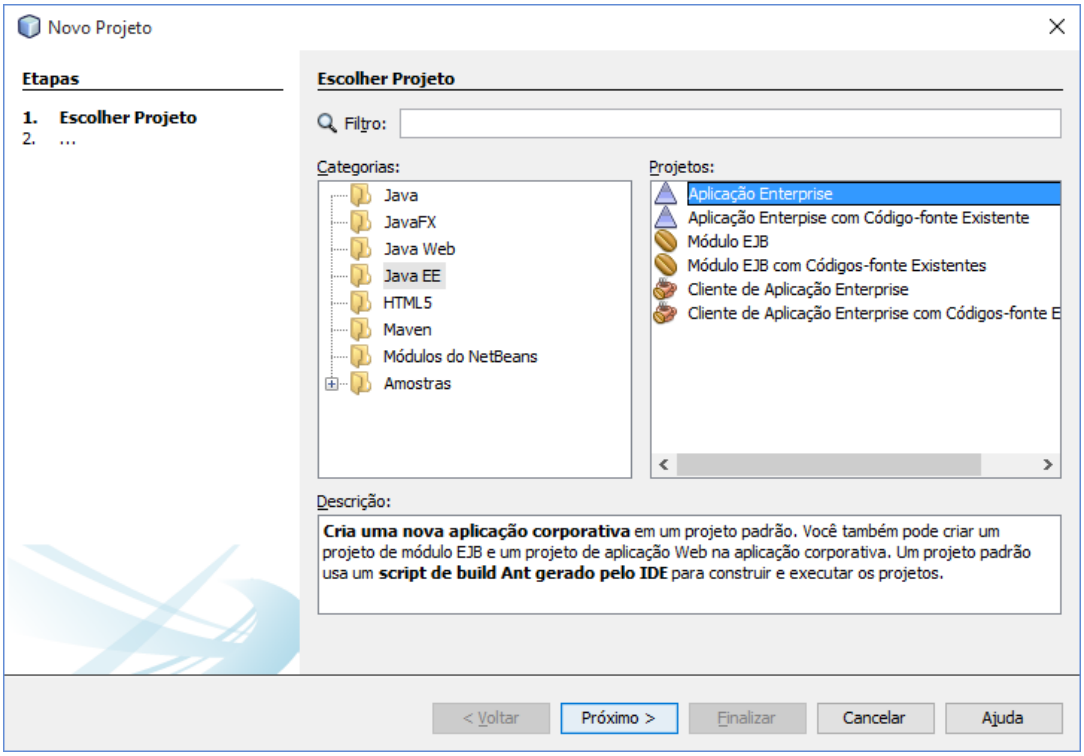
Seu módulo EJB (módulo em que os componentes corporativos serão implementados e implantados).

3. Criação de um projeto para uma Aplicação Corporativa

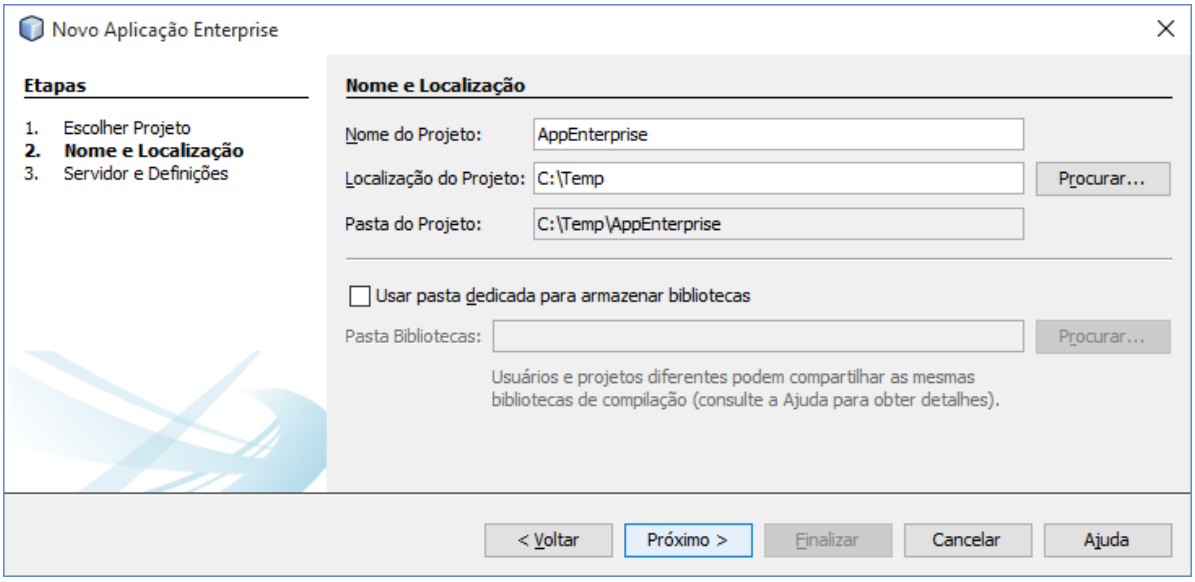
- Execute o netbeans.
- Na barra de comando selecione “Arquivo” -> Novo projeto.



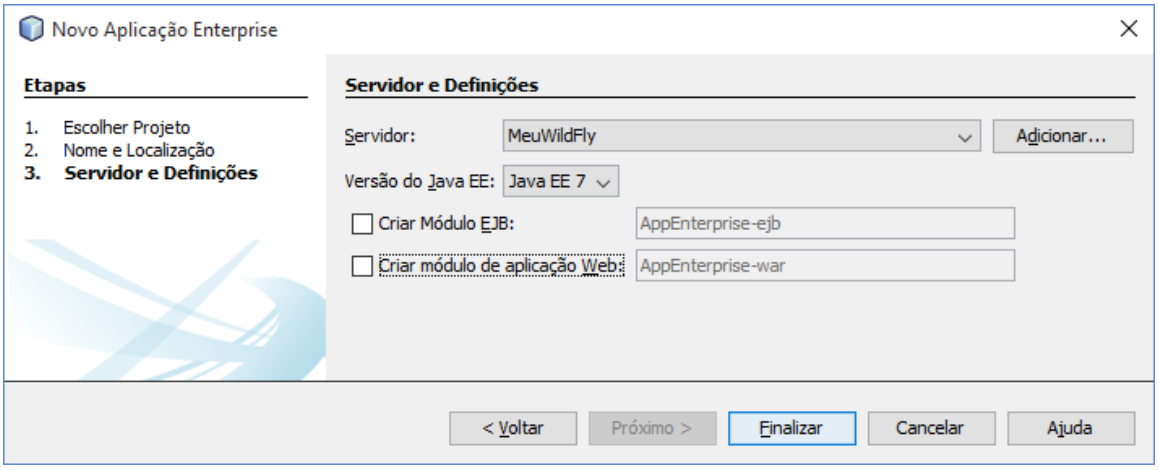
- Em categoria selecione “Java EE” e em Projetos, selecione “Aplicação Enterprise”. Clique em próximo.



- De um nome e escolha o diretório onde você deseja manter os fontes da aplicação e clique em próximo.

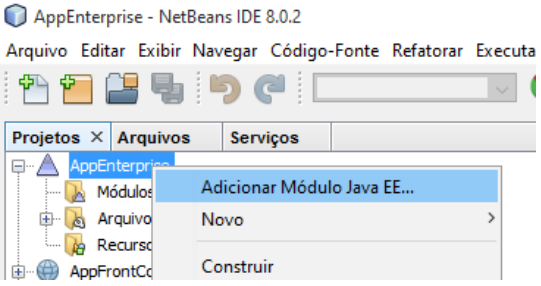


- Como já possuímos um módulo web (aplicação desenvolvida até o momento) e já criamos o módulo EJB, não devemos criar os módulos WEB e EJB. Clicar em Finalizar.

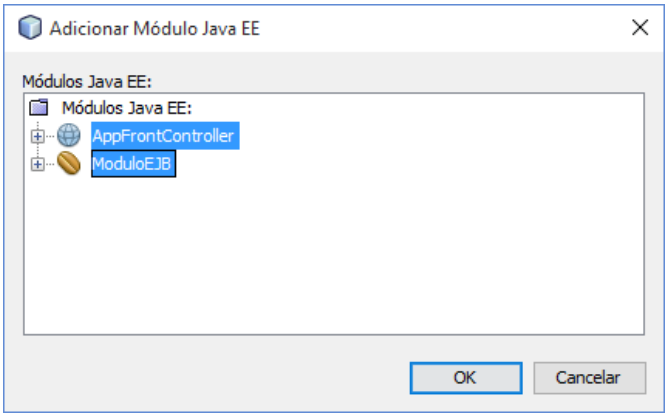


3.1. Configuração dos módulos corporativos da aplicação corporativa.

- Devemos configurar os módulos de uma aplicação corporativa, a saber, o módulo EJB que contém a implementação dos componentes que serão executados no EJB Container e o módulo de aplicação WEB que será executado no servlet container.
- Clicar com o botão da direita sobre o projeto java corporativo na aba Projetos e selecionar “Adicionar Módulo Java EE”.



- Selecione a aplicação WEB e o módulo EJB e clique em OK.

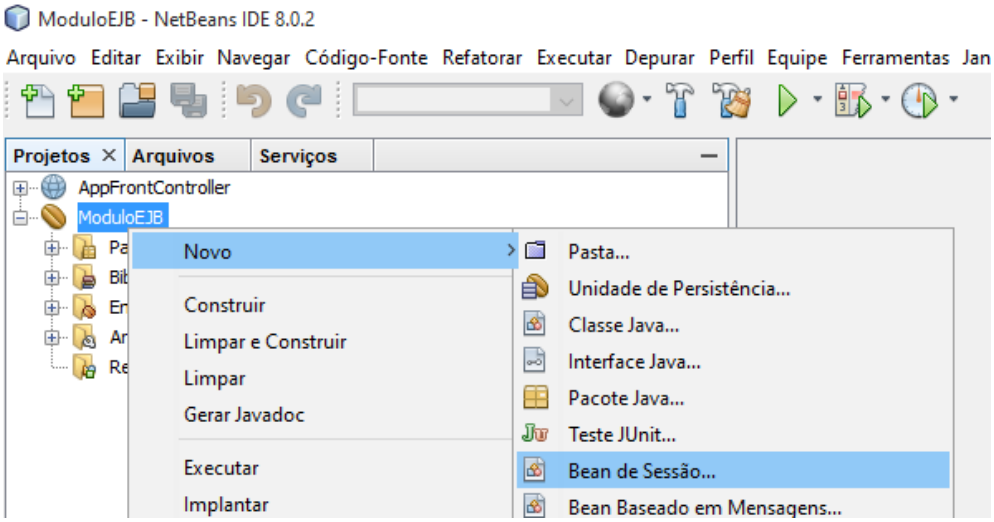


Sua aplicação corporativa foi criada com os módulos de apresentação e de negócios.

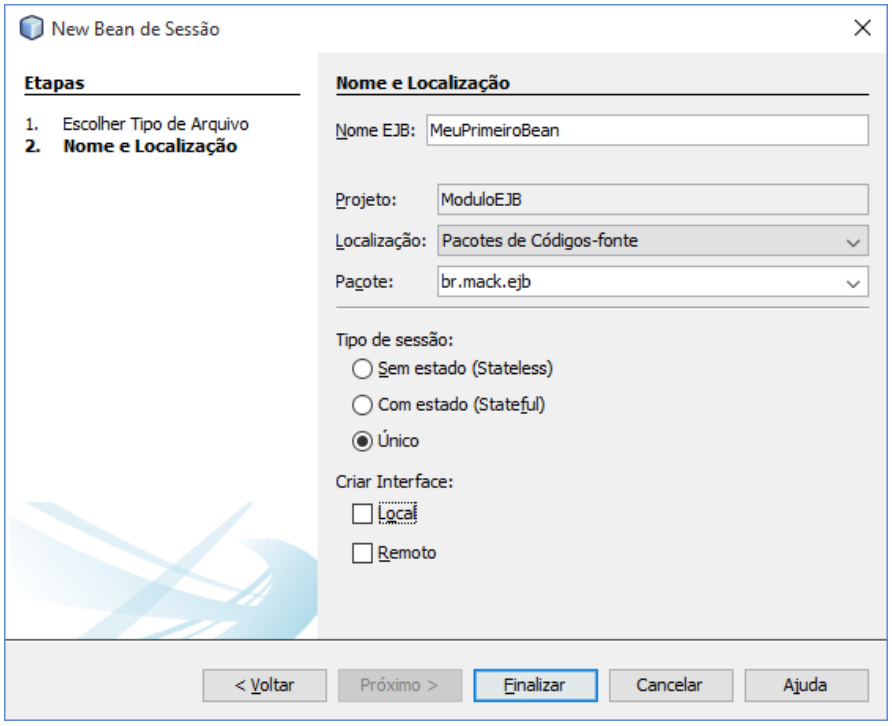
4. Implementação do primeiro componente corporativo

4.1. Criação de um componente corporativo no módulo EJB

- Execute o netbeans.
- Na aba de “Projetos”, clicar com o botão da direita do mouse sobre o projeto do módulo EJB. Selecionar Novo -> bean de sessão.



- Dar um nome para seu bean, indicar o seu pacote e marca-lo como único e clicar em Finalizar.



- Implementar o seu primeiro bean.

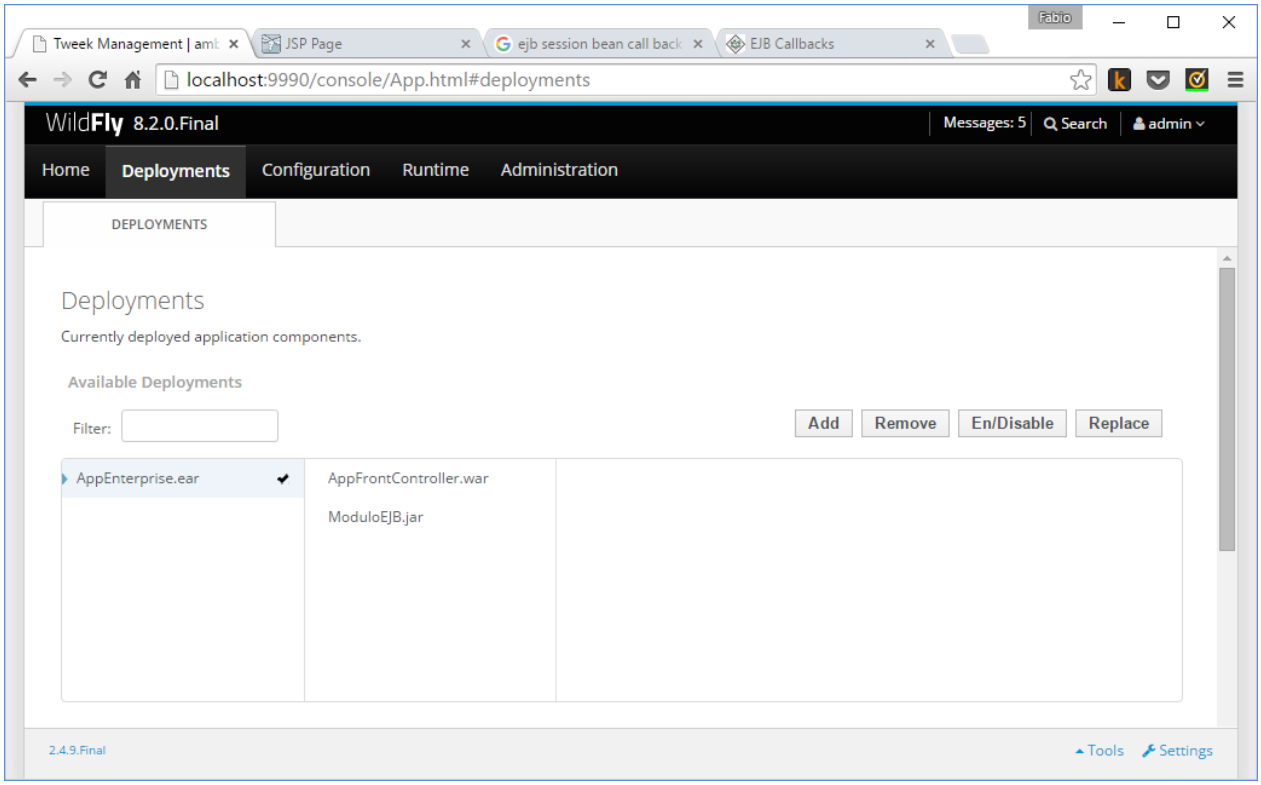
```
package br.mack.ejb;



import javax.annotation.PostConstruct;
import javax.annotation.PreDestroy;
import javax.ejb.Singleton;
import javax.ejb.LocalBean;
import javax.ejb.PostActivate;
import javax.ejb.PrePassivate;

@Singleton
@LocalBean
public class MeuPrimeiroBean {

    private static int count = 0;
    private int id = 0;

    public MeuPrimeiroBean(){
        id = count ++;
```

```
        System.out.println("Bean Criado. ID = " + this.id);
    }

    @PreDestroy
    public void reportaDestruicao(){
        System.out.println("Report de destruicao do Bean #" +this.id );
    }

    @PostConstruct
    public void reportaConstrucao(){
        System.out.println("Report de construcao do Bean #" +this.id);
    }

    @PostActivate
    public void reportaAtivacao(){
        System.out.println("Report de ativacao de Bean #" +this.id);
    }

    @PrePassivate
    public void reportaPassivacao(){
        System.out.println("Report de passivacao do Bean #" +this.id);
    }

    public String metodoDeNegocio(){
        System.out.println("Bean #" + this.id + " executando método de negocio.");
        return "metodo executado pelo bean #" + this.id;
    }

}
```

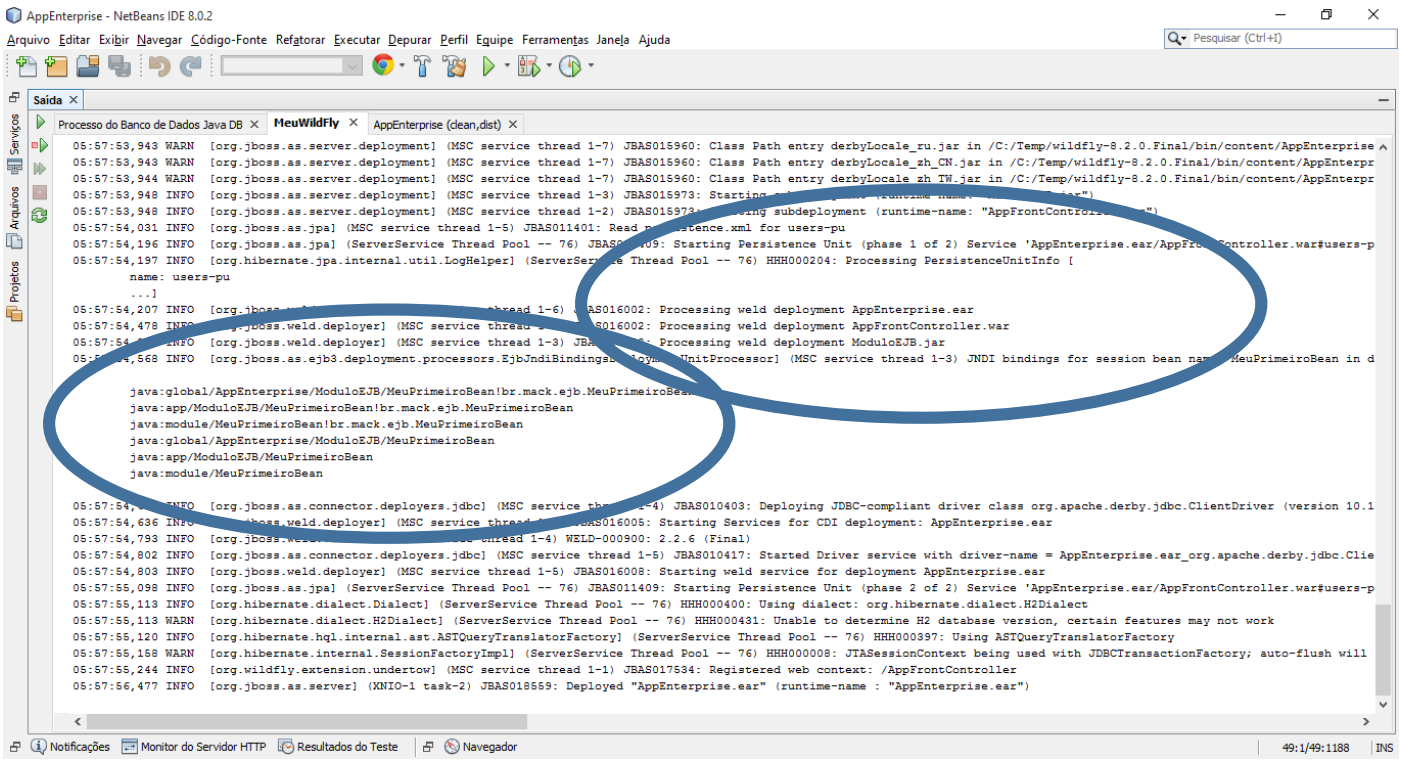
4.2. Realizar a implantação da aplicação corporativa no Wildfly

- "Limpar e construir" a aplicação corporativa.
- Acessar a interface de administração do Wildfly.
- Remover a implantação anterior da aplicação WEB.

- Adicionar uma nova implantação e selecionar o arquivo do tipo EAR no diretório dist da aplicação corporativa.
- Não se esquecer de habilitar a implantação.
- Note que ao implantar a aplicação corporativa, os módulos WEB e EJB serão implantados.



- No log do servidor de aplicações podemos verificar a implantação dos módulos e a disponibilização do “MeuPrimeiroBean”.

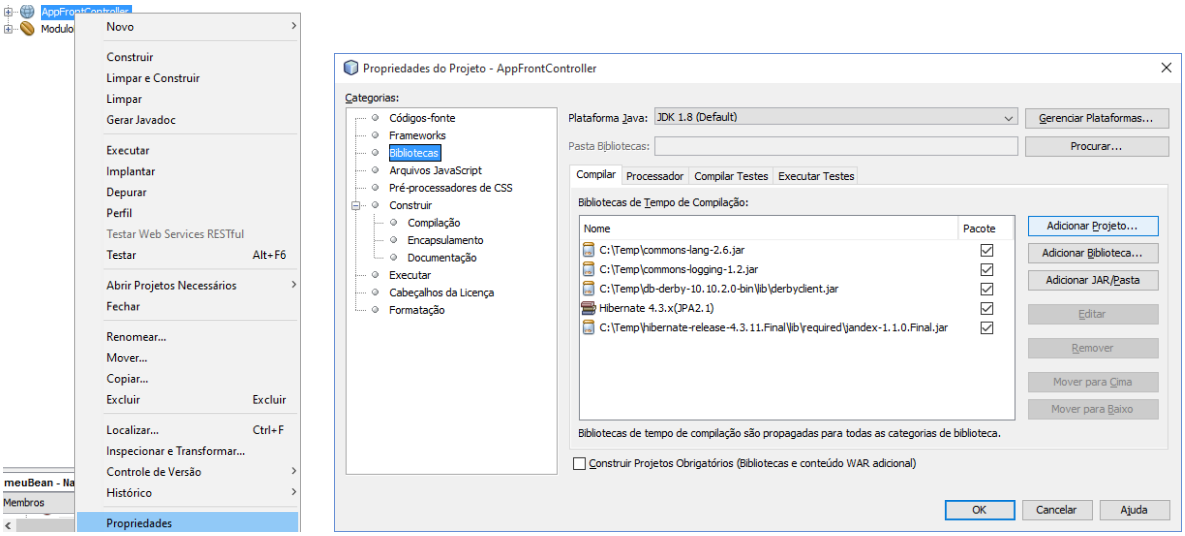


4.3. Utilização do Bean corporativo criado.

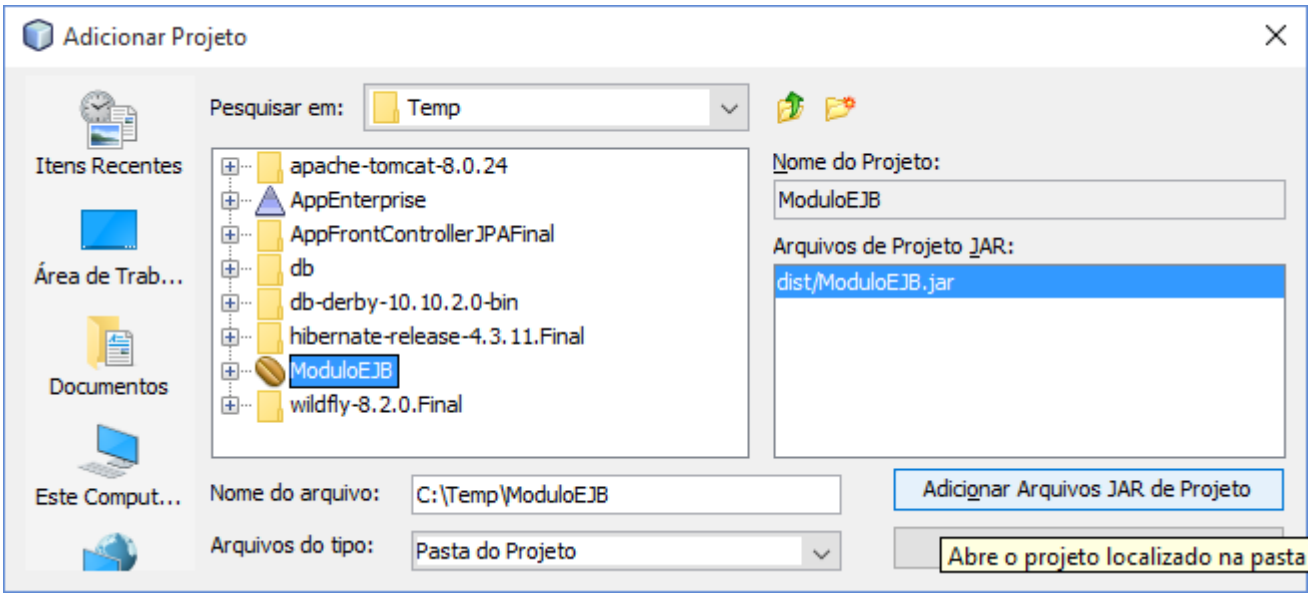
- Toda interação com o sistema implementado ocorrerá através da camada de apresentação, isto é, aplicação WEB.
- Vamos implementar um servlet na aplicação web que fará acesso ao bean corporativo implementado.
- Para que o servlet tenha acesso ao bean corporativo implementado, a aplicação web precisa conhecer a definição da classe.

4.3.1. Configuração da aplicação web para conhecer a definição de beans corporativos implementados.

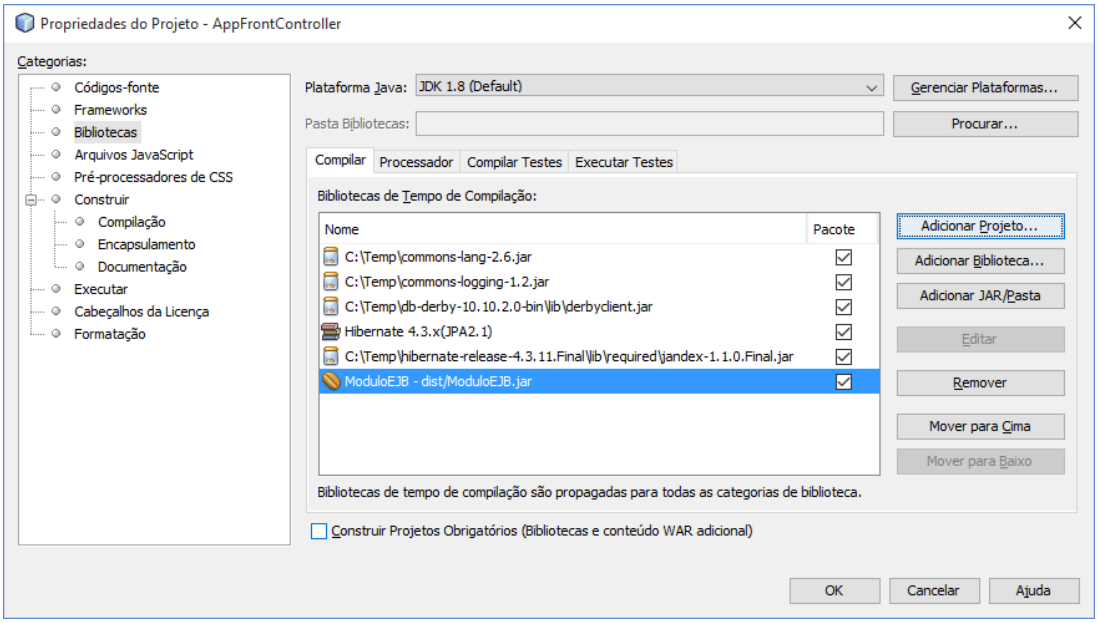
- Clicar com o botão da direita do mouse sobre a aplicação web na aba “Projetos” e selecionar propriedades.
- Selecionar Bibliotecas e clicar em Adicionar Projeto...



- Navegar até a pasta onde se encontra o projeto do módulo EJB, seleciona-lo e clicar em Adicionar Arquivos JAR de Projeto.



- Selecionar “Construir Projetos Obrigatórios ... ” e clicar em OK.



4.3.2. Implementação do servlet.

- Criar um servlet na aplicação web chamado MeuPrimeiroBeanServlet utilizando o assistente. Não se esqueça de incluir o mapeamento do servlet no web.xml.

```
package mack.servlets;


import br.mack.ejb.MeuPrimeiroBean;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;



public class MeuPrimeiroBeanServlet extends HttpServlet {
```

```
    private MeuPrimeiroBean meuBean;


    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            Context context = null;
            try {
                context  = new InitialContext();
                meuBean = (MeuPrimeiroBean)
context.lookup("java:app/ModuloEJB/MeuPrimeiroBean");
            } catch (NamingException ex) {
                Logger.getLogger(MeuPrimeiroBeanServlet.class.getName()).log(Level.SEVERE,
null, ex);
            }

            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet MeuPrimeiroBeanServlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet MeuPrimeiroBeanServlet at " + meuBean.metodoDeNegocio()
+ "</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    @Override
    public String getServletInfo() {
        return "Short description";
    }

}
```