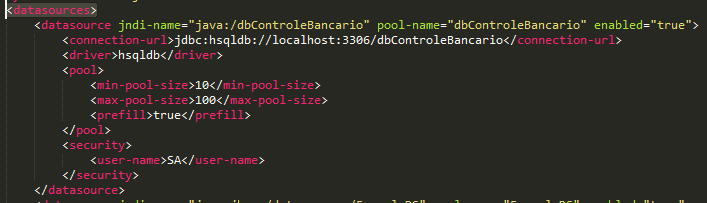
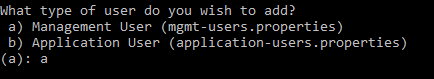
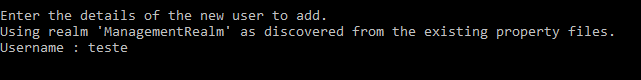
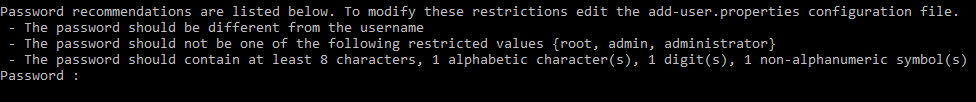
**Manual de instalação, configuração e utilização do WebService Controle Bancário**

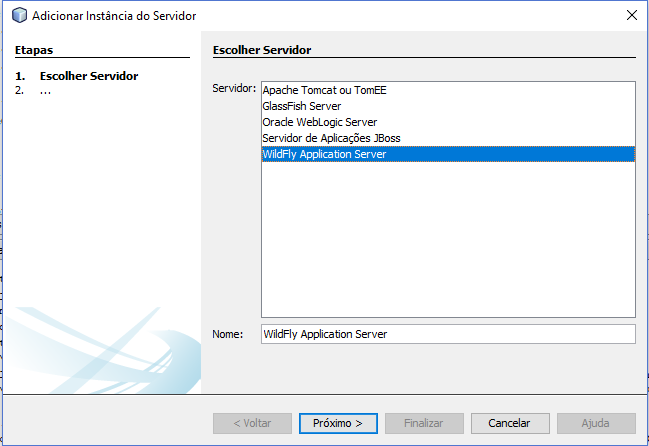
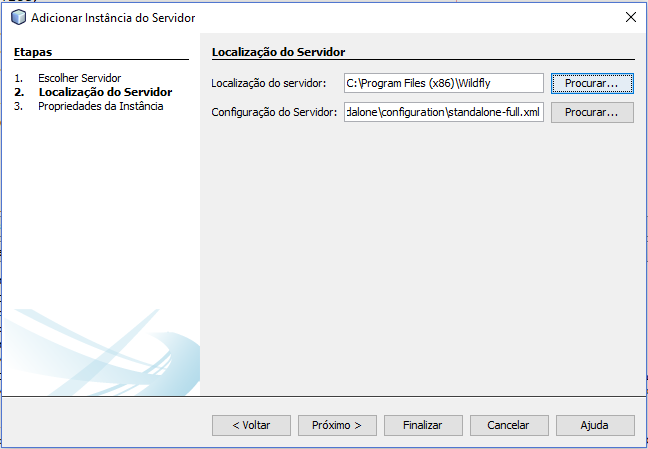
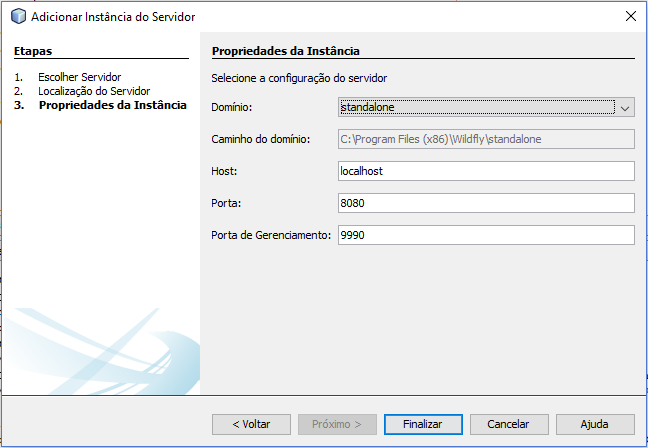
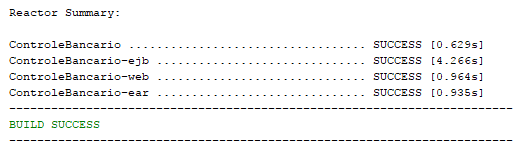
**Pré-Requisitos**

* Será necessário ter instalado na variável de ambiente JAVA\_HOME, a versão do **Java 8.** Durante a implementação e testes foi utilizado a atual versão [jdk1.8.0\_161](https://www.java.com/pt_BR/).
* Será necessário ter instalado o software [SoapUI Open Source](https://www.soapui.org/downloads/soapui.html) para simular as requisições do tipo Soap ao WebService aqui exposto.
* Para a compilação do código fonte será necessário ter instalado a IDE NetBeans. Durante o desenvolvimento e testes foi utilizado a versão mais atual da IDE com JEE: [NetBeans 8.2](https://netbeans.org/downloads/start.html?platform=windows&lang=pt_BR&option=javaee&pagelang=pt_BR)
* Todos os passos de instalação e utilização expostos neste documento foram criados e expostos para o uso em ambiente Windows. No caso foi utilizado a [versão 10](https://www.microsoft.com/pt-br/software-download/windows10) do sistema operacional.

**Instalação Jboss**

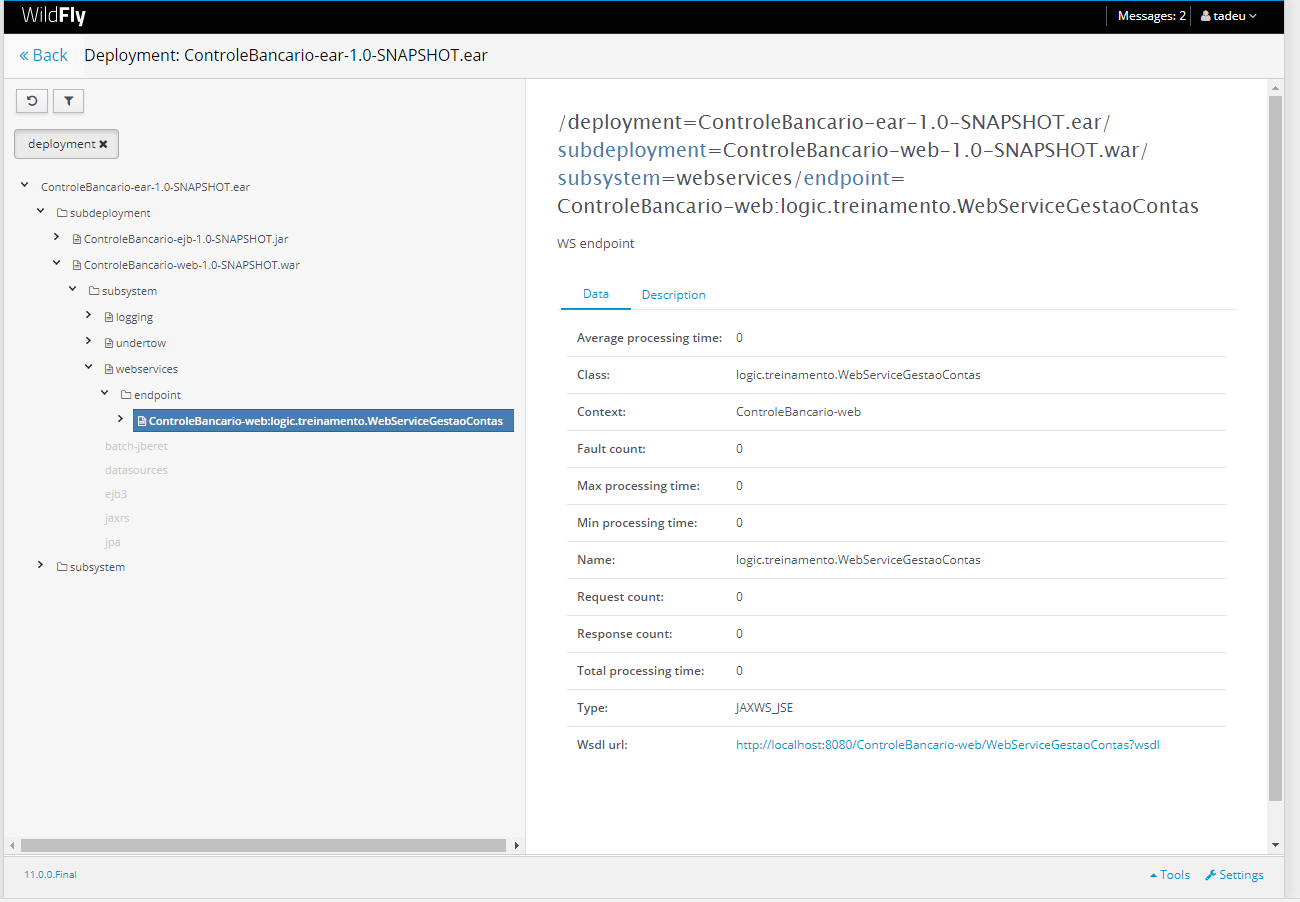
1. Baixe o Jboss **WildFly v.12.0.0** através do link <http://wildfly.org/downloads/>
2. Após o termino do download extraia os arquivos em seu diretório padrão de programas ([*Windows*] Ex: C:\Program Files).
3. *Como no projeto anterior já foi explicado e detalhado como fazer a configuração do servidor para utilizar o banco HSQLDB com o projeto exposto, neste passo estarei disponibilizando os arquivos necessários já configurados para a utilização do sistema.* Dentro do projeto versionado pelo [GitHub](https://github.com/), existe uma pasta chamada ***Configracao\_WildFly****.* Copie o arquivo **standalone.xml** para dentro do diretório *standalone\configuration* presente na pasta do servidor WildFly.
4. Copie também a pasta hsqldb para o diretório \modules\system\layers\base\org presente na pasta do servidor WildFly. Esta pasta contem o Driver de conexão com o banco de dados do qual o servidor irá precisar.
5. Dentro da pasta **org**, crie outra pasta chamada ***hsqldb***.
6. Dentro da pasta **hsqldb**, crie outra pasta chamada ***main***.
7. Dentro da pasta **main**, crie um arquivo do tipo ***xml*** e com o nome module.
8. Abra o arquivo e insira as seguintes configurações:
9. Baixe o driver JDBC do banco **HSQLDB v.2.4.0** através do link <https://sourceforge.net/projects/hsqldb/files/hsqldb/hsqldb_2_4/>
10. Ao termino do download, extraia os arquivos.
11. Copie o arquivo **hsqldb.jar** presente no diretório ***hsqldb-2.4.0/hsqldb/lib*** para dentro da pasta **main** criada no ***passo 5***.
12. Abra o arquivo **standalone.xml** presente no diretório ***/wildfly-11.0.0.Final/standalone/configuration*** da instalação do jboss.
13. Edite-o acrescentando as seguintes configurações:  
    Dentro da tag ***<datasources>*** acrescente mais um **<datasource>**:  
    
14. Dentro da tag ***<drivers>*** acrescente mais um **<driver>**:
15. Através do prompt de comando do Windows (cmd) acesse a pasta bin presente no seguinte diretório de instalação do jboss WildFly: ***/wildfly-11.0.0.Final/bin***
16. Execute o arquivo **add-user.bat** para adicionar um usuário do tipo administrador.  
    
17. Escolha a opção a e tecle ENTER.  
    
18. Informe o nome do usuário (Ex: teste)  
    
19. Informe uma senha para o usuário.   
    
20. No caso de a senha possuir o mesmo nome do usuário, basta informar a opção **YES** e teclar **ENTER**.
21. A próxima opção refere-se a incluir o novo usuário à um grupo especifico. Neste caso não necessitamos disto, então apenas tecle **ENTER** para continuar.
22. O próximo passo refere-se apenas à uma confirmação quanto ao usuário que esta sendo criado como administrador do jboss. Informe a opção **YES** e tecle **ENTER**.
23. O próximo passo refere-se apenas à permitir ou não o novo usuário acessar o jboss remotamente. Informe a opção **YES** e tecle **ENTER**.
24. Ao termino desse processo, uma semelhante à esta será exibida sinalizando o sucesso da operação.  
    ***To represent the user add the following to the server-identities definition <secret value="dGVzdGU=" />  
    Pressione qualquer tecla para continuar. . .***
25. Ainda neste diretório, execute o arquivo **standalone.bat** para subir o jboss  
    
26. Em caso de sucesso, o jboss irá exibir uma mensagem semelhante à esta:   
    ***20:51:12,378 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full 11.0.0.Final (WildFly Core 3.0.8.Final) started in 6545ms - Started 601 of 829 services (360 services are lazy, passive or on-demand)***
27. Em caso de sucesso, acessando o endereço <http://localhost:8080/> será exibido a seguinte imagem:  
    
28. Caso você tem obtido nenhuma destas respostas que sinalizam o sucesso do carregamento do jboss WildFly, **refaça** todos os passos aqui citados.

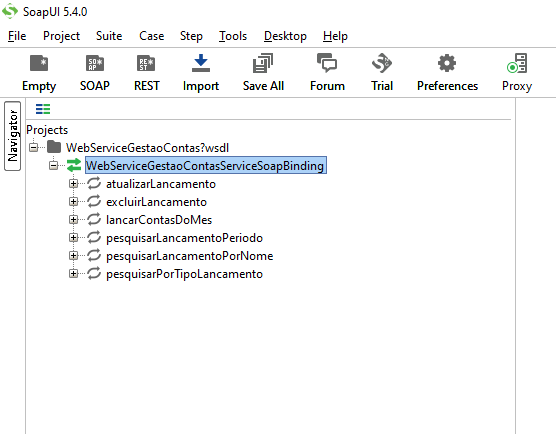
**Projeto Controle Bancário: Download, Compilação e Deploy do projeto no jboss WildFly**

1. Abra a IDE ***NetBeans***.
2. Na guia ***Serviços***, sub-item ***Servidores***, adicione um novo servidor (**WildFly**) conforme detalhado nas imagens abaixo:  
     
     
     
     
   
3. Faça o download do projeto através do link <https://github.com/tadeusantosti/treinamentojavaee>
4. Após o termino do download, abra o projeto na IDE ***NetBeans*** e compile o projeto principal **ControleBancario**.
5. Ao termino da compilação o NetBeans, em caso de sucesso a IDE irá exibir a seguinte mensagem:  
   
6. Com o projeto devidamente compilado acesse a área de gestão do jboss através do link <http://localhost:9990/console/App.html#home>
7. Digite o login e a senha cadastrados anteriormente no **passo 15** do capitulo ***Instalação Jboss***
8. Clique na opção **Deployments**.
9. Clique na opção **ADD**.
10. Na tela exibida, mantenha a opção ***Upload a new deploymentUse this option to upload a new artifact, such as a WAR or EAR archive*** selecionada e clique em **NEXT**.
11. Clique no botão **Escolher Arquivo**.
12. Na janela de navegação aberta, selecione o arquivo **ControleBancario-ear-1.0-SNAPSHOT.ear** presente no diretório ***\treinamentojavaee\ControleBancario-ear\target***
13. Clique em **NEXT** e em seguida **FINISH**.
14. Em caso de sucesso, o jboss irá exibir uma mensagem em destaque verde informando que o arquivo foi carregado com sucesso.

**Configurando o SoupUI para utilização do WebService Controle Bancário**

1. Acesse a pagina **Deployments** presente na sessão de administrador do jboss (**passo 8** do capitulo **Projeto Controle Bancário: Download, Compilação e Deploy do projeto no jboss WildFly**)
2. Clique na opção **VIEW** ao lado do arquivo ***ControleBancario-ear-1.0-SNAPSHOT.ear*** carregado pelo jboss.
3. Acesse o arquivo **ControleBancario-web-logic-treinamento-WebServiceGestaoContas** no endpoint conforme imagem abaixo.



1. Copie a url do **WSDL** presente no campo **Wsdl url.**
2. Abra o programa ***SoupUI.***
3. Tecle o comando ***Ctrl + N*** ou acesse ***File – New SOAP Project****.*
4. No campo **Initial WSDL**, cole a url do WSLD copiada no **passo 4** e clique no botão **OK**.
5. Em caso de sucesso, o sistema irá exibir os serviços do WebService, conforme a imagem abaixo:  
   

**Utilização de cada Serviço Disponível do WebService Controle Bancário**

* **lancarContaDoMes**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Valor** |
| Nome | Texto | Qualquer caractere. |
| Valor | Numérico | Os valores que possuem valores decimais, devem serem separados por *ponto* (Ex: 784.24). |
| idTipoLancamento | Tipo do Lançamento que será salvo | Deve ser informado o código que representa o tipo do lançamento, podendo ser:  (1) TRANSFERENCIA  (2) SAQUE  (3) DEPOSITO |
| Data | Data | A data deve ser informado no seguinte formato: dd/MM/aaaa |

* **atualizarLancamento**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Valor** |
| Id | Inteiro | Código do lançamento que será atualizado. |
| Nome | Texto | Qualquer caractere. |
| Valor | Numérico | Os valores que possuem valores decimais, devem serem separados por *ponto* (Ex: 784.24). |
| idTipoLancamento | Inteiro | Deve ser informado o código que representa o tipo do lançamento, podendo ser:  (1) TRANSFERENCIA  (2) SAQUE  (3) DEPOSITO |
| Data | Data | A data deve ser informado no seguinte formato: dd/MM/aaaa |

* **excluirLancamento**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Valor** |
| Id | Inteiro | Código do lançamento que será excluido. |

* **pesquisarLancamentoPeriodo**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Valor** |
| dataInicial | Data | A data deve ser informado no seguinte formato: dd/MM/aaaa |
| dataFinal | Data | A data deve ser informado no seguinte formato: dd/MM/aaaa |

* **pesquisarLancamentoPorNome**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Valor** |
| nome | Text | Informar o nome ou qualquer parte do mesmo. |

* **pesquisarPorTipoLancamento**

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Valor** |
| TipoLancamento | Inteiro | Deve ser informado o código que representa o tipo do lançamento que será consultado, podendo ser:  (1) TRANSFERENCIA  (2) SAQUE  (3) DEPOSITO |