



**UNIVERSIDADE FEDERAL DO PARÁ**  
**CENTRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÕES**  
**COORDENADORIA DE SISTEMAS DE INFORMAÇÃO**

**TUTORIAL DE CONFIGURAÇÃO DO JBOSS 5.1**

Paulo Natan Andrade dos Santos

## Tutorial de como preparar o ambiente para o SIPAC 4.22.14

Primeiramente, deve ter o java 6 instalado no computador, para que seja possível fazer a configuração do ambiente.

Utilizando o Eclipse Luna, é necessário alterar o compilador para qu esteja compatível com o ambiente. Para isto:

Acessar a opção **“Window → Preference → Java → Compiler”** e na seção **“JDK Compliance”**, opção **“Compiler compliance level:”** indicar **“1.6”**. Clique no botão **“OK”**. Aparecerá uma tela semelhante a da Figura 1.

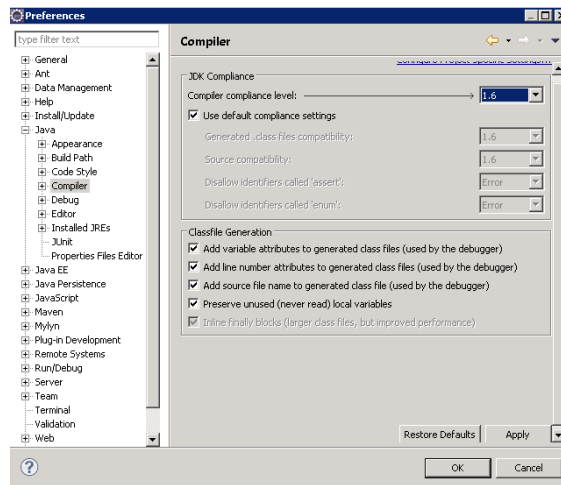


Figura 1 Fonte: wiki UFRN

Em seguida, alterar a configuração do ambiente de execução, através do caminho:

**“Window → Preference → Java → Installed JREs”**

A Figura 2 demonstra a tela de configuração.

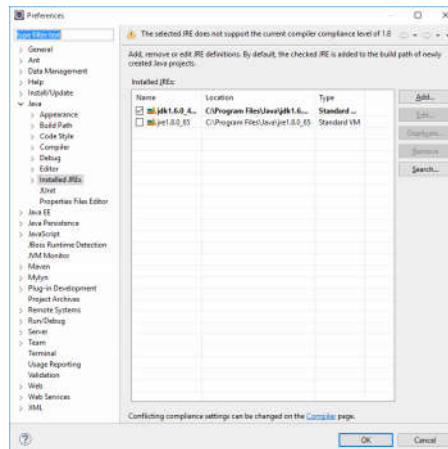


Figura 2 Alterar a versão do java

E indicar o jdk1.6 ou o jre1.6, ficando a critério do usuário. No caso deste tutorial, foi utilizado o jdk1.6.0\_45.

O próximo passo é configurar a codificação de caracteres, que o padrão do ambiente é a ISO-8859-1. Para selecioná-la, deve-se seguir o caminho:

**“Window → Preference → Workspace”**

E em “Text file encoding”, selecionar o ISO-8859-1. A Figura 3 exibe a configuração.

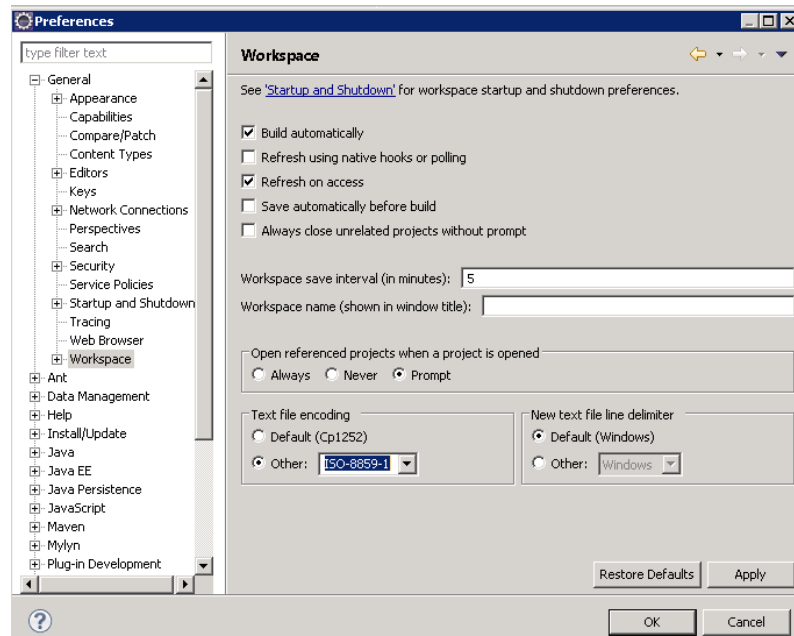


Figura 3 Configuração da codificação

O passo seguinte é a instalação do plugin para SVN, preferencialmente o subclipse 1.10.13. Para isto, é preciso acessar a opção:

**“Help → Eclipse Market Place”**

E pesquisar pelo subclipse, em seguida, instalá-lo.

Após instalar o plugin, deve-se clicar em “open perspective” e em seguida em “SVN Repository Exploring”. Na tela que aparecer, deve-se informar a URL que o eclipse irá conectar para baixar os arquivos do projeto.

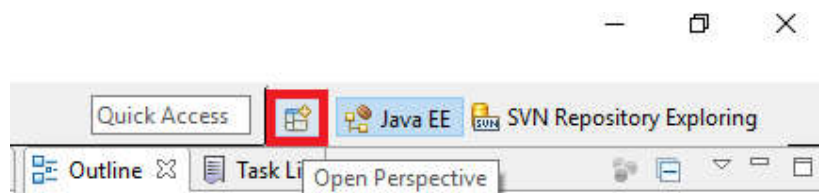


Figura 4: Open Perspective

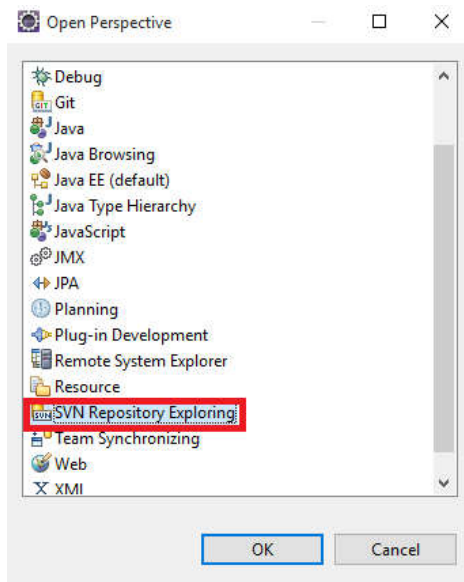


Figura 5 SVN Repository Exploring

Para ter acesso aos arquivos, é necessário configurar o acesso. Para isto, deve-se clicar com o botão direito do mouse na aba “SVN Repositories”, selecione “New → Repository Location...”, conforme exibido pela Figura 6.

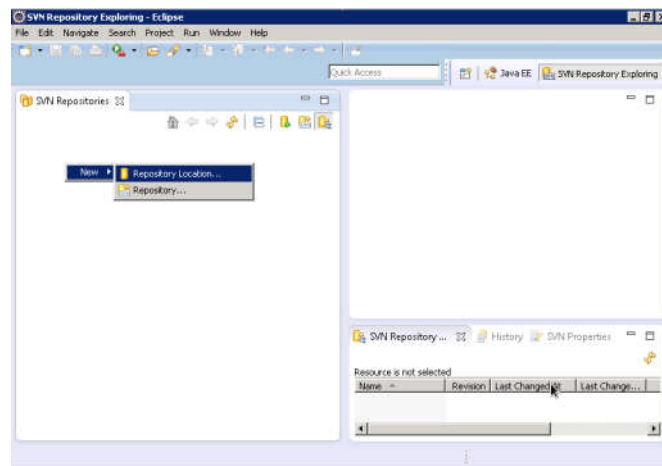


Figura 6 New Repository Location

O passo seguinte é para informar a url, usuário e senha de conexão ao repositório da UFRN. O link é <https://version.info.ufrn.br/cooperacao>, o usuário é “ufpa” e a senha é “@ufpa#ufrn” (sem as aspas).

A Figura 7 demonstra um exemplo do wiki da ufrn.



**Figura 7 Fonte: Wiki UFRN**

Ao conectar com o servidor, deve-se acessar as pastas:

**tags → homologacao**

E a partir disso, fazer o checkout das seguintes pastas:

- **ServicosIntegrados 1.4.12;**
- **Arquitetura 2.7.20;**
- **SharedResources 1.6.9;**
- **EntidadesComuns 1.4.15;**
- **Libs 1.5.5;**
- **SIPAC 4.22.14;**

Após o término do download, é necessário verificar se todas as pastas estão com o java 6 no build path. Caso não esteja, será necessário inseri-lo.

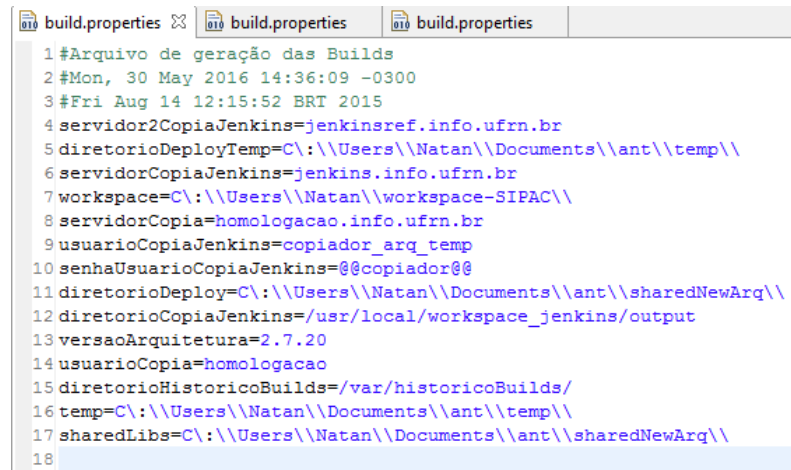
O passo seguinte é a geração dos .jar's que auxiliarão a execução do projeto. Deve-se criar os seguintes arquivos:

- **arq-2.7.20;**
- **dto-1.4.12;**
- **comum-1.4.15**

Para gerar estes três arquivos, é preciso alterar os seguintes arquivos, chamados de build.properties:

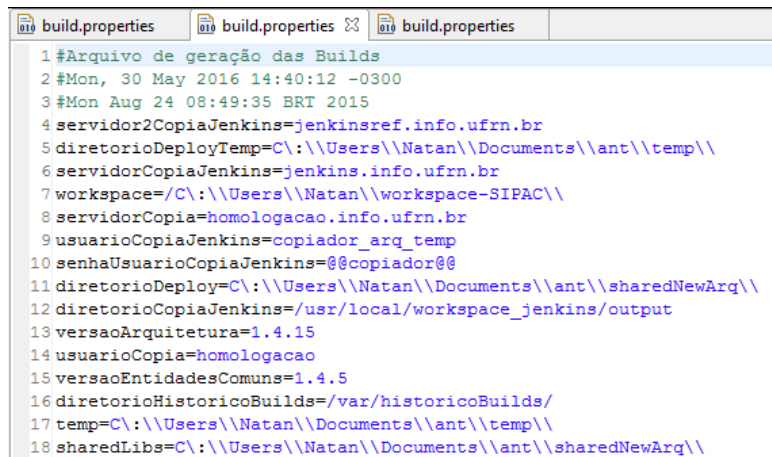
- Da pasta **Arquitetura;**
- Da pasta **EntidadesComuns;**
- Da pasta **ServicosIntegrados;**

As Figuras 8, 9 e 10 demonstram exemplos de configuração.



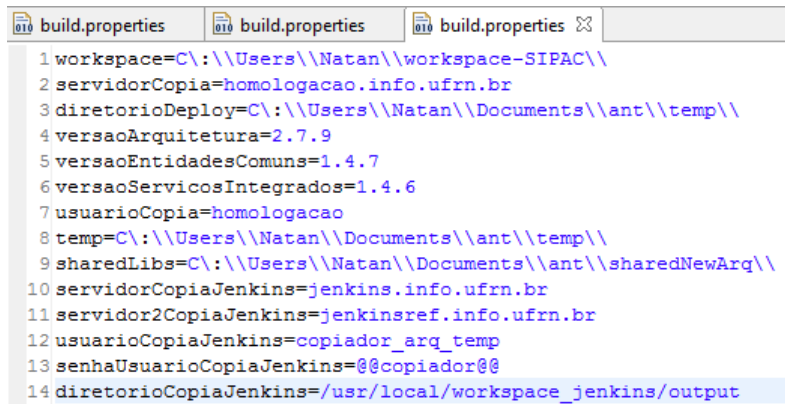
```
1 #Arquivo de geração das Builds
2 #Mon, 30 May 2016 14:36:09 -0300
3 #Fri Aug 14 12:15:52 BRT 2015
4 servidor2CopiaJenkins=jenkinsref.info.ufrn.br
5 diretorioDeployTemp=C:\\Users\\Natan\\Documents\\ant\\temp\\
6 servidorCopiaJenkins=jenkins.info.ufrn.br
7 workspace=C:\\Users\\Natan\\workspace-SIPAC\\
8 servidorCopia=homologacao.info.ufrn.br
9 usuarioCopiaJenkins=copiador_arq_temp
10 senhaUsuarioCopiaJenkins=@@copiador@@
11 diretorioDeploy=C:\\Users\\Natan\\Documents\\ant\\sharedNewArq\\
12 diretorioCopiaJenkins=/usr/local/workspace_jenkins/output
13 versaoArquitetura=2.7.20
14 usuarioCopia=homologacao
15 diretorioHistoricoBuilds=/var/historicoBuilds/
16 temp=C:\\Users\\Natan\\Documents\\ant\\temp\\
17 sharedLibs=C:\\Users\\Natan\\Documents\\ant\\sharedNewArq\\
18
```

Figura 8 build.properties Arquitetura



```
1 #Arquivo de geração das Builds
2 #Mon, 30 May 2016 14:40:12 -0300
3 #Mon Aug 24 08:49:35 BRT 2015
4 servidor2CopiaJenkins=jenkinsref.info.ufrn.br
5 diretorioDeployTemp=C:\\Users\\Natan\\Documents\\ant\\temp\\
6 servidorCopiaJenkins=jenkins.info.ufrn.br
7 workspace=/C:\\Users\\Natan\\workspace-SIPAC\\
8 servidorCopia=homologacao.info.ufrn.br
9 usuarioCopiaJenkins=copiador_arq_temp
10 senhaUsuarioCopiaJenkins=@@copiador@@
11 diretorioDeploy=C:\\Users\\Natan\\Documents\\ant\\sharedNewArq\\
12 diretorioCopiaJenkins=/usr/local/workspace_jenkins/output
13 versaoArquitetura=1.4.15
14 usuarioCopia=homologacao
15 versaoEntidadesComuns=1.4.5
16 diretorioHistoricoBuilds=/var/historicoBuilds/
17 temp=C:\\Users\\Natan\\Documents\\ant\\temp\\
18 sharedLibs=C:\\Users\\Natan\\Documents\\ant\\sharedNewArq\\
```

Figura 9 build.properties EntidadesComuns



```
1 workspace=C:\\Users\\Natan\\workspace-SIPAC\\
2 servidorCopia=homologacao.info.ufrn.br
3 diretorioDeploy=C:\\Users\\Natan\\Documents\\ant\\temp\\
4 versaoArquitetura=2.7.9
5 versaoEntidadesComuns=1.4.7
6 versaoServicosIntegrados=1.4.6
7 usuarioCopia=homologacao
8 temp=C:\\Users\\Natan\\Documents\\ant\\temp\\
9 sharedLibs=C:\\Users\\Natan\\Documents\\ant\\sharedNewArq\\
10 servidorCopiaJenkins=jenkins.info.ufrn.br
11 servidor2CopiaJenkins=jenkinsref.info.ufrn.br
12 usuarioCopiaJenkins=copiador_arq_temp
13 senhaUsuarioCopiaJenkins=@@copiador@@
14 diretorioCopiaJenkins=/usr/local/workspace_jenkins/output
```

Figura 10 build.properties ServicosIntegrados

As pastas temp e sharedNewArq, foram criadas no computador onde foi executado a geração dos builds. O endereço foi repassado para os build.properties, conforme as três imagens anteriores exibem.

Para gerar os arquivos .jar, é necessário acessar o eclipse e clicar em “**window → show view → other**” e pesquisar por “**ant**”.

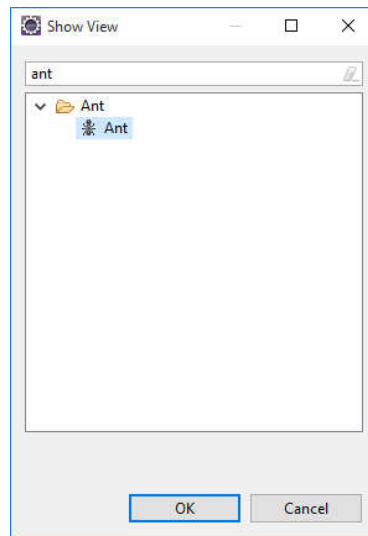


Figura 11 Ant

Aparecerá na parte de baixo do eclipse o ícone do ant, demonstrando que ele foi iniciado. Agora, para gerar os *builds*, é necessário clicar em “*add buildfiles*” e adicionar os build.properties que foram editados, das pastas **Arquitetura**, **EntidadesComuns** e **ServicosIntegrados**.



Figura 12 add Buildfiles

Após adicioná-los, é possível criar os .jar's para a execução do projeto.

Para a geração dos arquivos, primeiramente deve-se clicar em “**DeployARQ → deployAll [default]**” para gerar o jar da arquitetura. Ao clicar, ele vai demonstrar uma janela como a da Figura 13, onde deve ser informado o parâmetro **2.7.20**.

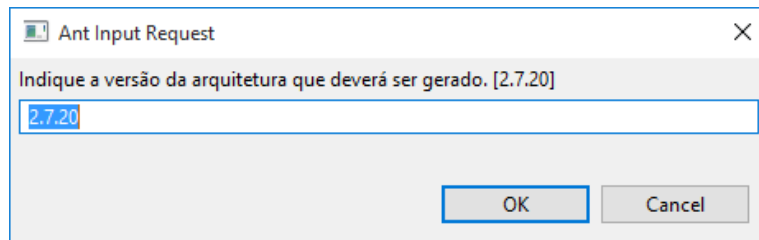


Figura 13 arq-2.7.20

O próximo a ser gerado é o dos serviços. Note que deve ser clicado no ícone “**deployDTO**” e fornecer o valor **1.4.12**, conforme a Figura 14.

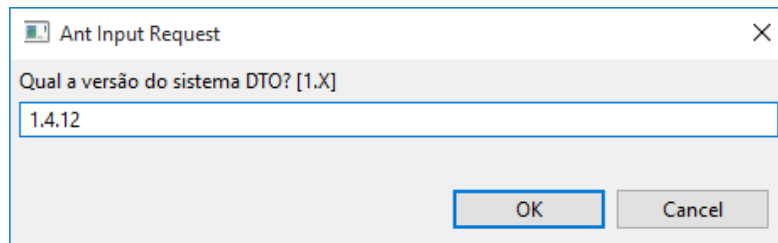


Figura 14 dto-1.4.12

Por fim, deve-se criar o do comum, clicando no ícone “**deployCOMUM** → **deployAll[default]**” e informar o valor **1.4.15**, conforme a Figura 15.

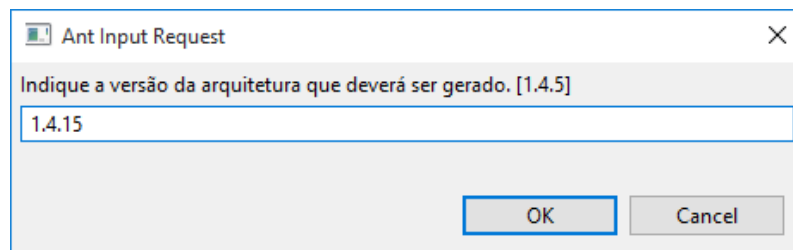


Figura 15 comum-1.4.15

Quando são gerados os .jar's dto e comum, o eclipse demonstra uma mensagem em vermelho, mas geralmente os arquivos serão criados e estarão na pasta sharedNewArq.

Com os três arquivos, é necessário verificar todas as pastas do projeto, exceto as pastas “LIBS”, “Arquitetura” e “ServicosIntegrados”, procurando se existe o diretório “dependencias” dentro delas. Se não existir, será necessário criá-la.

Após as devidas pastas possuírem o subdiretório “dependencias”, deve-se adicionar os .jar's da seguinte maneira:

- EntidadesComuns/dependencias: Adicionar os .jar's **arq-2.7.20** e **dto-1.4.12**;
- SharedResources/dependencias: Adicionar os .jar's **arq-2.7.20**, **dto-1.4.12** e **comum-1.4.15**;



- SIPAC/dependências: Adicionar os .jar's **arq-2.7.20**, **dto-1.4.12** e **comum-1.4.15**;

Estes arquivos que foram gerados não serão adicionados ao build path, pois no teste realizado não foi necessário para o funcionamento do programa, porém os arquivos estavam dentro das referidas pastas. Provavelmente não são necessários, mas somente com mais testes que isto pode ser esclarecido.

Agora é necessário inserir as dependências dos projetos. Para informar as dependências das pastas “**EntidadesComuns**”, “**ServicosIntegrados**”, “**SharedResources**” e “**SIPAC**”. Para isto, é necessário clicar com o botão direito do mouse em uma destas pastas e escolher a opção “*Properties*”, conforme a Figura 16.

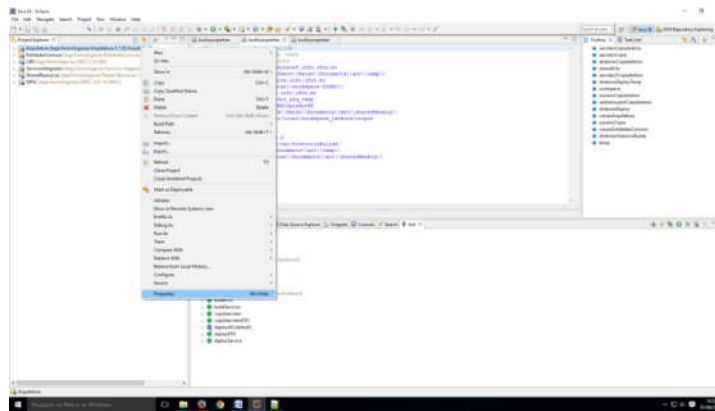


Figura 16 Properties

Depois, escolher a opção “*Java Build Path*”, depois a aba “*Projects*” e no botão “*Add*” para adicionar os projetos que a referida pasta depende. O procedimento é necessário para todas as pastas que foram informadas anteriormente. A Figura 17 demonstra um exemplo da forma de inserir os projetos.

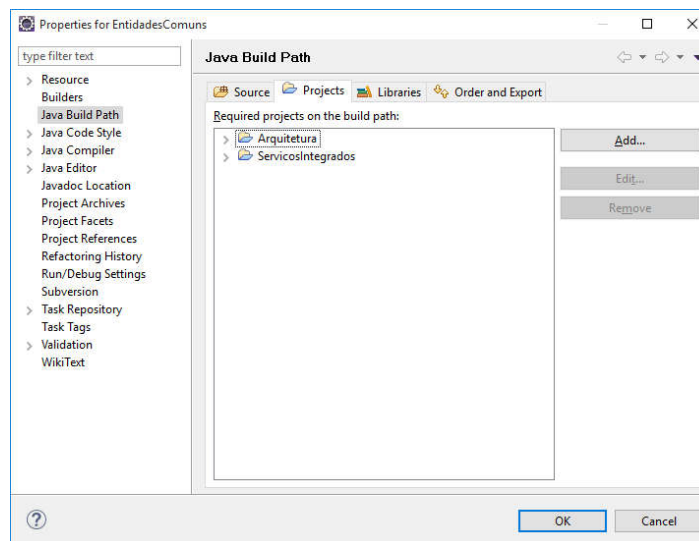


Figura 17 Projects

As dependências são:

- EntidadesComuns **depende** de Arquitetura e ServicosIntegrados;
- ServicosIntegrados **depende** de Arquitetura;
- SharedResources **depende** de Arquitetura, EntidadesComuns e ServicosIntegrados;
- SIPAC **depende** de Arquitetura, EntidadesComuns e ServicosIntegrados;

O próximo passo é configurar o Jboss 5.1, que deve ser baixado do seguinte link da UFRN: <http://migre.me/sLLui>. Ele deve ser descompactado no computador, em uma pasta a critério do usuário.

No eclipse, deve-se clicar na aba “servers” e depois em “No servers are available. Click this link to create a new server...”, conforme a Figura 18.

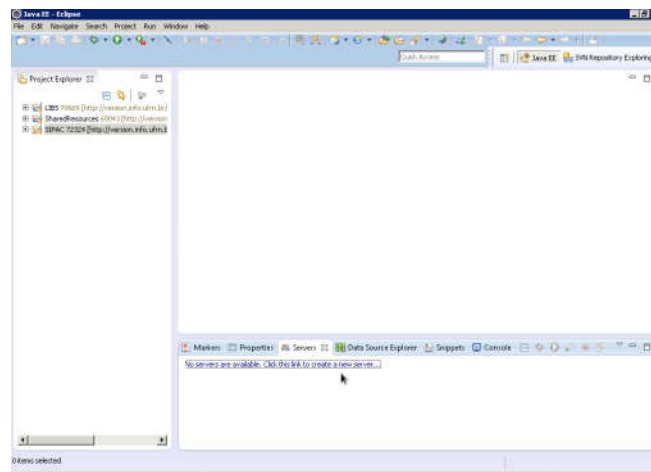


Figura 18 Fonte: Wiki UFRN

Na tela que for exibida, escolher o JBoss 5.1. Caso não apareça nenhum JBoss, é necessário instalá-lo através do “market place” do eclipse, num procedimento similar ao do svn.

A Figura 19 exhibe a escolha do JBoss 5.1.

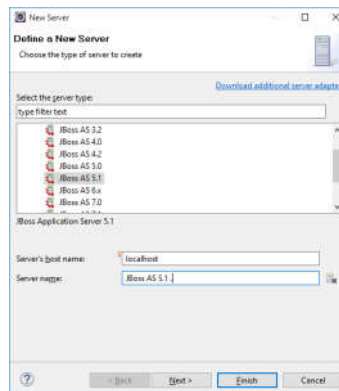


Figura 19 JBoss 5.1

Em seguida, clicar em “*Next*”, a tela a ser exibida é conforme a Figura 20. Em seguida, clicar em “*Next*” novamente.

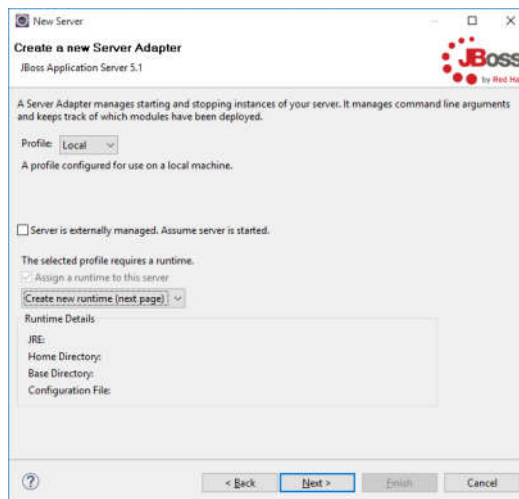


Figura 20 Create a New Server Adapter

Na próxima tela será necessário informar o local onde o JBoss foi descompactado, a versão do java que será utilizado e o nome para o servidor. Conforme o exibido pela Figura 21.

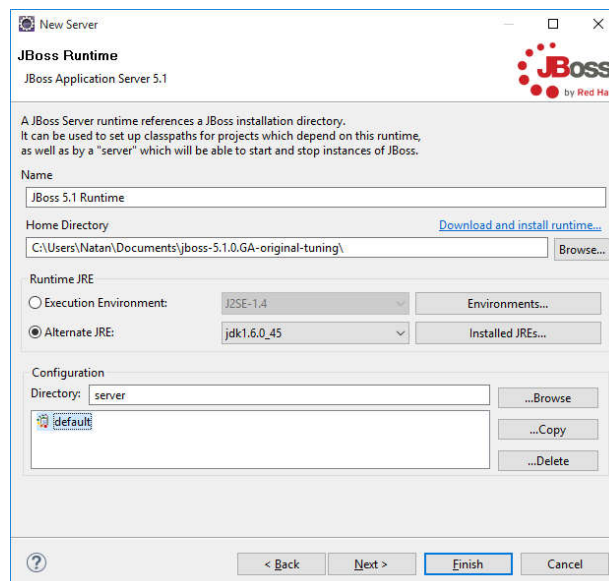


Figura 21 JBoss Runtime

A configuração deve ser similar ao da Figura 21. Terminado isso, clique em “*Next*” e depois em “*Finish*”, para terminar a configuração do JBoss no eclipse.

Antes de iniciar os testes, é necessário editar os arquivos do JBoss: **profile.xml**, **postgres-ds.xml** e **run.conf**, para adaptá-los aos testes.

O arquivo profile.xml fica na pasta que foi descompactada, no subdiretório jboss-5.1.0.GA-original-tuning\server\default\conf\bootstrap. Neste arquivo são inseridos os diretórios que o JBoss irá tentar executar a aplicação a configuração deve ser similar a:

```
<bean name="BootstrapProfileFactory" class="org.jboss.system.server.profileservice.repository.StaticProfileFactory">
  <property name="bindingsURI">${jboss.server.home.url}conf/bindingservice.beans</property>
  <property name="bootstrapURI">${jboss.server.home.url}conf/jboss-service.xml</property>
  <property name="deployersURI">${jboss.server.home.url}deployers</property>
  <property name="applicationURIs">
    <list elementClass="java.net.URI">
      <value>${jboss.server.home.url}deploy</value>
      <value>file:///C:/Users/Natan/workspace-SIPAC/Arquitetura/app</value>
      <value>file:///C:/Users/Natan/workspace-SIPAC/EntidadesComuns/app</value>
      <value>file:///C:/Users/Natan/workspace-SIPAC/ServicosIntegrados/dependencias</value>
      <value>file:///C:/Users/Natan/workspace-SIPAC/ServicosIntegrados/app</value>
      <value>file:///C:/Users/Natan/workspace-SIPAC/SharedResources/app</value>
      <value>file:///C:/Users/Natan/workspace-SIPAC/LTBS/app</value>
      <value>file:///C:/Users/Natan/workspace-SIPAC/SIPAC/app</value>
    </list>
  </property>
</bean>
```

Figura 22 profile.xml

Após salvar o arquivo, deve-se alterar o postgresql-ds.xml, para que seja possível a conexão com o banco de dados. O arquivo fica no diretório jboss-5.1.0.GA-original-tuning\server\default\deploy.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <datasources>
3
4
5
6
7
8 <xa-datasource>
9 <jndi-name>jdbc/SIPACDB</jndi-name>
10 <xa-datasource-class>org.postgresql.xa.XADataSource</xa-datasource-class>
11 <xa-datasource-property name="ServerName">caire.ufpa.br</xa-datasource-property>
12 <xa-datasource-property name="PortNumber">5435</xa-datasource-property>
13 <xa-datasource-property name="DatabaseName">administrativo</xa-datasource-property>
14 <xa-datasource-property name="User">natan</xa-datasource-property>
15 <xa-datasource-property name="Password"></xa-datasource-property>
16 <track-connection-by-tx/>
17 <min-pool-size>1</min-pool-size>
18 <max-pool-size>10</max-pool-size>
19 <check-valid-connection-sql>select 1</check-valid-connection-sql>
20 </xa-datasource>
21
22
23
24 <xa-datasource>
25 <jndi-name>jdbc/SIGRHDB</jndi-name>
26 <xa-datasource-class>org.postgresql.xa.XADataSource</xa-datasource-class>
27 <xa-datasource-property name="ServerName">caire.ufpa.br</xa-datasource-property>
28 <xa-datasource-property name="PortNumber">5435</xa-datasource-property>
29 <xa-datasource-property name="DatabaseName">administrativo</xa-datasource-property>
30 <xa-datasource-property name="User">natan</xa-datasource-property>
31 <xa-datasource-property name="Password"></xa-datasource-property>
32 <track-connection-by-tx/>
33 <min-pool-size>1</min-pool-size>
34 <max-pool-size>10</max-pool-size>
35 <check-valid-connection-sql>select 1</check-valid-connection-sql>
36 </xa-datasource>
37
38
39
40 <xa-datasource>
41 <jndi-name>jdbc/SIGAADB</jndi-name>
42 <xa-datasource-class>org.postgresql.xa.XADataSource</xa-datasource-class>
43 <xa-datasource-property name="ServerName">caire.ufpa.br</xa-datasource-property>
44 <xa-datasource-property name="PortNumber">5435</xa-datasource-property>
45 <xa-datasource-property name="DatabaseName">sigaa</xa-datasource-property>
46 <xa-datasource-property name="User">natan</xa-datasource-property>
47 <xa-datasource-property name="Password"></xa-datasource-property>
48 <track-connection-by-tx/>
49 <min-pool-size>1</min-pool-size>
50 <max-pool-size>10</max-pool-size>
51 <check-valid-connection-sql>select 1</check-valid-connection-sql>
52 </xa-datasource>
53
54
```

Figura 23 datasource

Todos os campos entre o <xa-datasource> e </xa-daatsource> devem ser preenchidos com os parâmetros do banco, no caso os “property name”, conforme o exemplo da Figura 23.

Após salvar o arquivo, alterar o arquivo run.conf, que está no diretório jboss-5.1.0.GA-original-tuning\bin. Neste arquivo, basta descomentar a linha onde está o JAVA\_HOME e inserir o caminho onde o jdk1.6 está instalado no computador. A Figura 24 exhibe um exemplo da edição.

```

1  ## -*- shell-script -*- #####
2  ##                                     ##
3  ## JBoss Bootstrap Script Configuration ##
4  ##                                     ##
5  #####
6
7  ### $Id: run.conf 88389 2009-05-08 02:37:00Z laprunger $
8
9
10 # This shell script is sourced by run.sh to initialize the environment
11 # variables that run.sh uses. It is recommended to use this file to
12 # configure these variables, rather than modifying run.sh itself.
13 #
14 #
15 #
16 # Specify the maximum file descriptor limit, use "max" or "maximum" to use
17 # the default, as queried by the system.
18 #
19 # Defaults to "maximum"
20 #
21 #MAX_FD="maximum"
22 #
23 #
24 # Specify the JBoss Profiler configuration file to load.
25 #
26 # Default is to not load a JBoss Profiler configuration file.
27 #
28 #PROFILER=""
29 #
30 #
31 # Specify the location of the Java home directory. If set then $JAVA will
32 # be defined to $JAVA_HOME/bin/java, else $JAVA will be "java".
33 #
34 #JAVA_HOME="C:\Program Files\Java\jdk1.6.0_45"
35 #
36 #
37 # Specify the exact Java VM executable to use.
38 #
39 #JAVA=""
40 #
41 #
42 # Specify options to pass to the Java VM.
43 #
44 # if [ "$s2JAVA_OPTS" = "" ]; then
45 #   JAVA_OPTS="-Xmx128m -Xms128m -XX:MaxPermSize=256m -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
46 # fi
47 #
48 # Sample JPSA settings for remote socket debugging
49 #JAVA_OPTS="$JAVA_OPTS -Xrunjdpttransport=ds_socket,address=3707,server=y,suspend="
50 #
51 # Sample JPSA settings for shared memory debugging
52 #JAVA_OPTS="$JAVA_OPTS -Xrunjdpttransport=ds_shmem,address=jboss,server=y,suspend="
53

```

Figura 24 run.conf

Por fim, deve-se salvar o arquivo e voltar ao eclipse para fazer o *deploy* da aplicação. Com o eclipse iniciado, clicar na aba “*servers*” e depois em “*start*”. Alguns erros podem acontecer durante a execução do *deploy*, mas provavelmente irá funcionar de maneira normal.

Quando aparecer a mensagem que o *deploy* foi realizado, deve-se abrir uma nova página do navegador *web* e digitar:

**localhost:8080/sipac**

Se tudo estiver funcionando normalmente, a tela como na Figura 25 será exibida e o *deploy* foi realizado com sucesso.



Figura 25 localhost