Desenvolvimento de Aplicações Java

Plataforma Corporativa

Tutorial

Data Source

JPA

EJB

WildFly

Agosto 2016

Sumário

1.Introdução

Este tutorial mostra como configurar um Data Source Apache Derby no WildFly.

ATENÇÃO:

Este tutorial pressupõe que:

1. O banco de dados Derby está instalado no diretório:

   `C:\Temp\projeto\ServidorDados\db-derby-10.10.2.0-bin`

   e que o banco de dados já exista e tenha sido criado no diretório:

   `C:\Temp\projeto\ServidorDados\db`

2. O Servidor de Aplicações JBoss WIldFly está instalado no diretório:

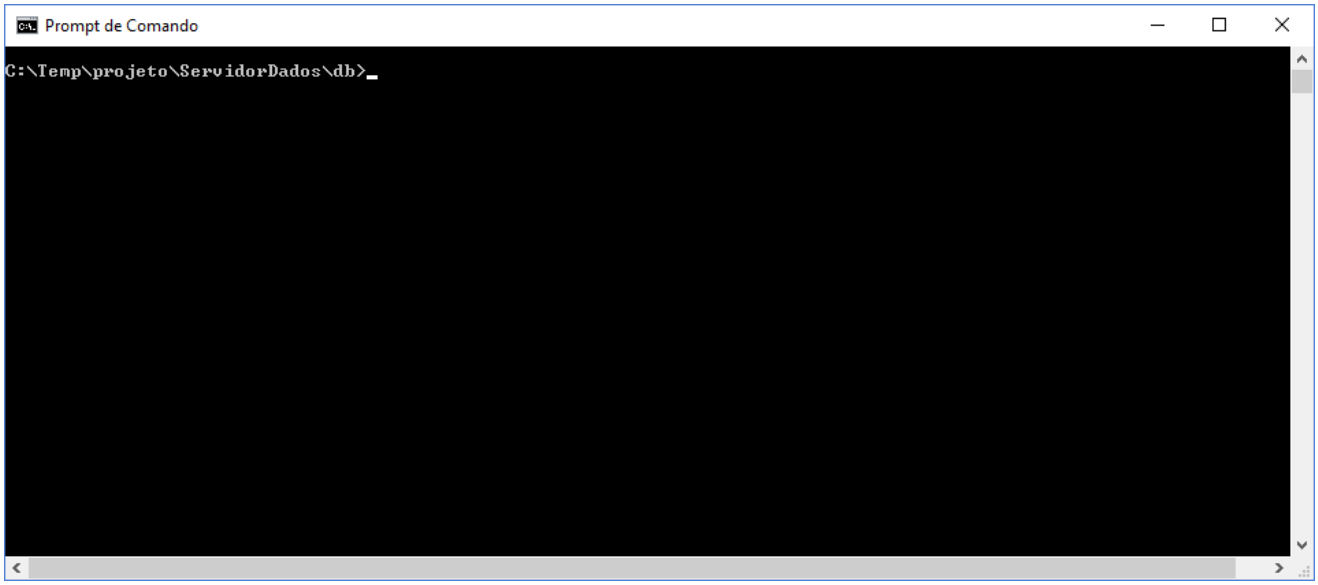
   `C:\Temp\projeto\wildfly-10.1.0.Final`

2. Servidor de Banco de Dados

2.1. Execução do Servidor de Banco de Dados

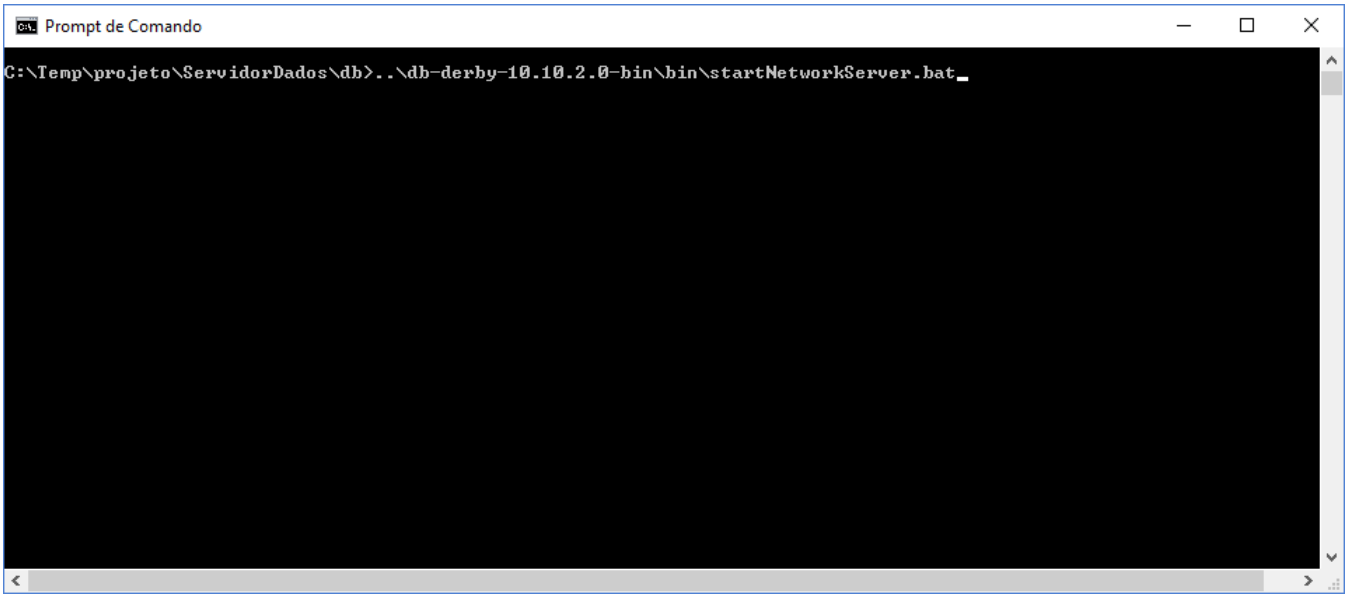
Para executar o servidor de banco de dados, siga os passos descritos a seguir:

1. Abra uma janela de prompt de comando.
2. Navegue até o diretório onde as bases de dados foram criadas (**`C:\Temp\projeto\ServidorDados\db`**)



3. Execute o comando:

   `..\db-derby-10.10.2.0-bin\bin\startNetworkServer.bat`



Prompt de Comando - ..\db-derby-10.10.2.0-bin\bin\startNetworkServer.bat

```
C:\Temp\projeto\ServidorDados\db>..\db-derby-10.10.2.0-bin\bin\startNetworkServer.bat
Fri Sep 30 07:05:41 BRT 2016 : DRDA_SecurityInstalled.I
Fri Sep 30 07:05:42 BRT 2016 : Apache Derby Servidor de Rede - 10.10.2.0 - (1582446) iniciado e pronto para aceitar c
x§es na porta 1527 em (3)
```

3. Servidor de Aplicações

3.1. Usuários do Servidor de Aplicações

- Criar um usuário administrador se ainda não existir.

```
Prompt de Comando - add-user.bat

C:\Temp\projeto\wildfly-10.1.0.Final\bin>add-user.bat

What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : admin
The username 'admin' is easy to guess
Are you sure you want to add user 'admin' yes/no? yes
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.
 - The password should be different from the username
 - The password should not be one of the following restricted values {root, admin, administrator}
 - The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[  ]:
About to add user 'admin' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'admin' to file 'C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\mgmt-users.properties'
Added user 'admin' to file 'C:\Temp\projeto\wildfly-10.1.0.Final\domain\configuration\mgmt-users.properties'
Added user 'admin' with groups  to file 'C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\mgmt-groups.properties'
Added user 'admin' with groups  to file 'C:\Temp\projeto\wildfly-10.1.0.Final\domain\configuration\mgmt-groups.properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <secret value="amF2YWVlMTIzIQ==" />
Pressione qualquer tecla para continuar. . .
```

Esta configuração pode ser observada nos dois arquivos:

```
C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\mgmt-groups.properties
```

```
#

# Properties declaration of users groups for the realm 'ManagementRealm'.

#

# This is used for domain management, users groups membership information is used to assign the user

# specific management roles.

#

# Users can be added to this properties file at any time, updates after the server has started

# will be automatically detected.

#

# The format of this file is as follows: -

# username=role1,role2,role3

#

# A utility script is provided which can be executed from the bin folder to add the users: -

# - Linux

#  bin/add-user.sh

#
```

```
# - Windows
#  bin\add-user.bat
#
# The following illustrates how an admin user could be defined.
#
admin=
```

C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\mgmt-users.properties

```
#
# Properties declaration of users for the realm 'ManagementRealm' which is the default realm
# for new installations. Further authentication mechanism can be configured
# as part of the <management /> in standalone.xml.
#
# Users can be added to this properties file at any time, updates after the server has started
# will be automatically detected.
#
# By default the properties realm expects the entries to be in the format: -
# username=HEX( MD5( username ':' realm ':' password))
#
# A utility script is provided which can be executed from the bin folder to add the users: -
# - Linux
#  bin/add-user.sh
#
# - Windows
#  bin\add-user.bat
# On start-up the server will also automatically add a user $local - this user is specifically
# for local tools running against this AS installation.
#
# The following illustrates how an admin user could be defined, this
# is for illustration only and does not correspond to a usable password.
#
admin=01f8c0797eda875affeac036504cdd02
#
#$REALM_NAME=ManagementRealm$ This line is used by the add-user utility to identify the realm name already used in this file.
```

```
#
```

- Criar um usuário de aplicação se ainda não existir (lembrar de inclui-lo no grupo guest para que ele tenha acesso as filas JMS).

```
Prompt de Comando - add-user.bat

C:\Temp\projeto\wildfly-10.1.0.Final\bin>add-user.bat

What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): b

Enter the details of the new user to add.
Using realm 'ApplicationRealm' as discovered from the existing property files.
Username : jmsUser
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.
 - The password should be different from the username
 - The password should not be one of the following restricted values {root, admin, administrator}
 - The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[  ]: guest
About to add user 'jmsUser' for realm 'ApplicationRealm'
Is this correct yes/no? yes
Added user 'jmsUser' to file 'C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\application-users.properties'
Added user 'jmsUser' to file 'C:\Temp\projeto\wildfly-10.1.0.Final\domain\configuration\application-users.properties'
Added user 'jmsUser' with groups guest to file 'C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\application-roles.proper
ies'
Added user 'jmsUser' with groups guest to file 'C:\Temp\projeto\wildfly-10.1.0.Final\domain\configuration\application-roles.properties

Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <secret value="am1zVXNlcjEyMyE=" />
Pressione qualquer tecla para continuar. . .
```

Esta configuração pode ser observada nos dois arquivos:

`C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\application-roles.properties`

```
#

# Properties declaration of users roles for the realm 'ApplicationRealm' which is the default realm

# for application services on a new installation.

#

# This includes the following protocols: remote ejb, remote jndi, web, remote jms

#

# Users can be added to this properties file at any time, updates after the server has started

# will be automatically detected.

#

# The format of this file is as follows: -

# username=role1,role2,role3

#

# A utility script is provided which can be executed from the bin folder to add the users: -

# - Linux

#  bin/add-user.sh

#

# - Windows

#  bin\add-user.bat
```

```
#
# The following illustrates how an admin user could be defined.
#
jmsUser=guest
```

C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\application-users.properties

```
#
# Properties declaration of users for the realm 'ApplicationRealm' which is the default realm
# for application services on a new installation.
#
# This includes the following protocols: remote ejb, remote jndi, web, remote jms
#
# Users can be added to this properties file at any time, updates after the server has started
# will be automatically detected.
#
# The format of this realm is as follows: -
# username=HEX( MD5( username ':' realm ':' password))
#
# A utility script is provided which can be executed from the bin folder to add the users: -
# - Linux
#  bin/add-user.sh
#
# - Windows
#  bin\add-user.bat
# The following illustrates how an admin user could be defined, this
# is for illustration only and does not correspond to a usable password.
#
jmsUser=555b6fa9a0f85b26e87bde31dd2077a7
#
#$REALM_NAME=ApplicationRealm$ This line is used by the add-user utility to identify the realm name already used in this file.
#
```

3.2. Execução do Servidor de Aplicações

Execute o servidor de aplicação a partir de um prompt de comando:

- Executar o comando utilizando as opções de bind com os IPs da máquina (-b 0.0.0.0) utilizando configuração completa, incluindo o servidor de mensageria (-c standalone-full.xml). (**`standalone.bat -b 0.0.0.0 -c standalone-full.xml`**)

Prompt de Comando

```
C:\Temp\projeto\wildfly-10.1.0.Final\bin>_
```

Prompt de Comando

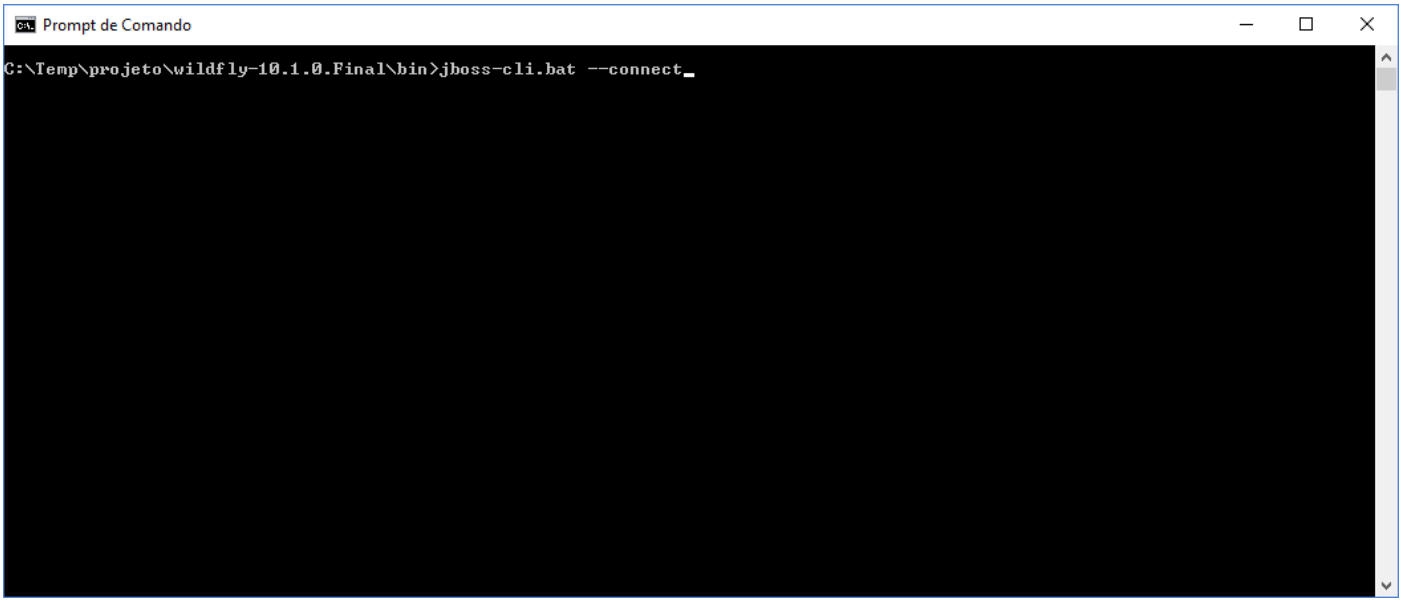
```
C:\Temp\projeto\wildfly-10.1.0.Final\bin>standalone.bat -b 0.0.0.0 -c standalone-full.xml_
```

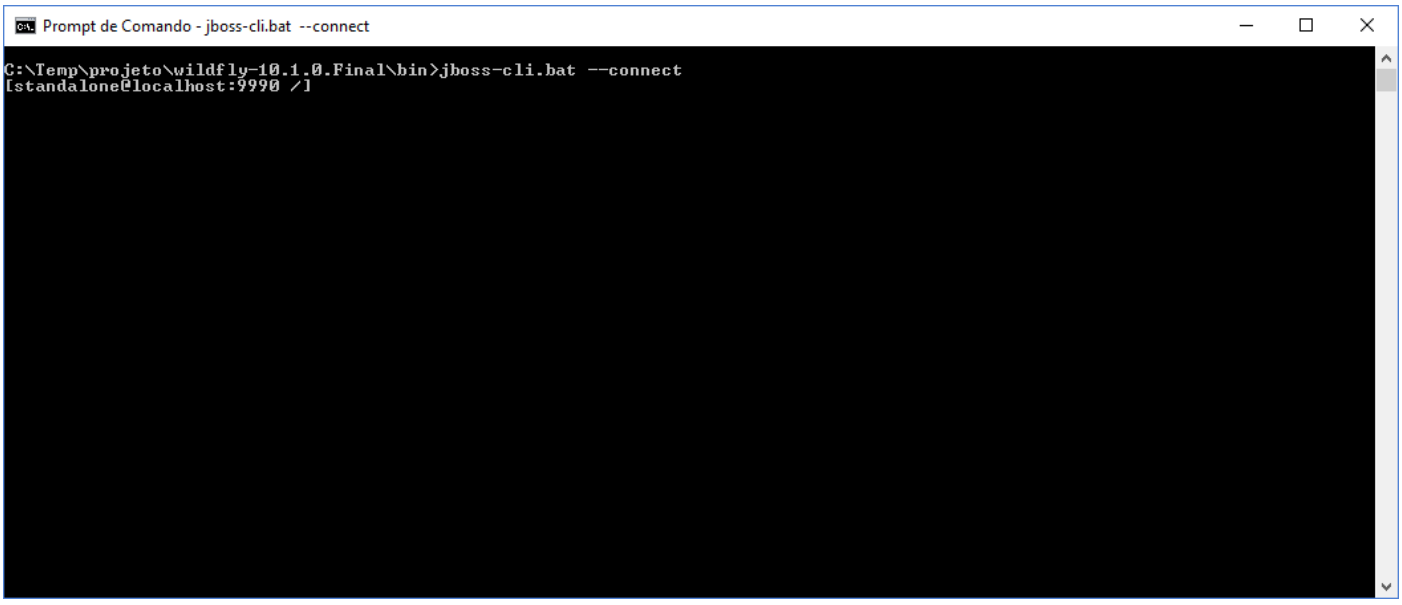
Prompt de Comando - standalone.bat -b 0.0.0.0 -c standalone-full.xml

```
live
07:08:26,963 INFO  [org.apache.activemq.artemis.core.server] (ServerService Thread Pool -- 66) AMQ221001: Apache Acti
Q Artemis Message Broker version 1.1.0.wildfly-017 [nodeID=d3746a0a-86f5-11e6-9dfd-a7b3b820f250]
07:08:26,979 INFO  [org.apache.activemq.artemis.core.server] (ServerService Thread Pool -- 66) AMQ221003: trying to d
oy queue jms.queue.DLQ
07:08:27,026 INFO  [org.jboss.as.connector.deployment] (MSC service thread 1-6) WFLYJCA0007: Registered connection fa
ry java:/JmsXA
07:08:27,088 INFO  [org.apache.activemq.artemis.ra] (MSC service thread 1-6) Resource adaptor started
07:08:27,104 INFO  [org.jboss.as.connector.services.resourceadapters.ResourceAdapterActivatorService$ResourceAdapterA
vator] (MSC service thread 1-6) IJ020002: Deployed: file://RaActivatoractivemq-ra
07:08:27,104 INFO  [org.jboss.as.connector.deployment] (MSC service thread 1-6) WFLYJCA0002: Bound JCA ConnectionFact
 [java:/JmsXA]
07:08:27,104 INFO  [org.wildfly.extension.messaging-activemq] (MSC service thread 1-2) WFLYMSGAMQ0002: Bound messagin
bject to jndi name java:jboss/DefaultJMSConnectionFactory
07:08:27,198 INFO  [org.wildfly.extension.messaging-activemq] (ServerService Thread Pool -- 69) WFLYMSGAMQ0002: Bound
ssaging object to jndi name java:/ConnectionFactory
07:08:27,198 INFO  [org.apache.activemq.artemis.jms.server] (ServerService Thread Pool -- 71) AMQ121005: Invalid "hos
value "0.0.0.0" detected for "http-connector" connector. Switching to "Ambriel". If this new address is incorrect ple
 manually configure the connector to use the proper one.
07:08:27,198 INFO  [org.wildfly.extension.messaging-activemq] (ServerService Thread Pool -- 71) WFLYMSGAMQ0002: Bound
ssaging object to jndi name java:jboss/exported/jms/RemoteConnectionFactory
07:08:27,198 INFO  [org.apache.activemq.artemis.core.server] (ServerService Thread Pool -- 70) AMQ221003: trying to d
oy queue jms.queue.ExpiryQueue
07:08:27,619 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://
7.0.0.1:9990/management
07:08:27,619 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://127.0.0.1:9

07:08:27,635 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full 10.1.0.Final (WildFly Core 2.2.0
nal) started in 6474ms - Started 370 of 615 services (408 services are lazy, passive or on-demand)
```

3.3. Execução do cliente de administração por linha de comando

- Abrir uma janela de prompt de comando.
- Navegar até a pasta **`C:\Temp\projeto\wildfly-10.1.0.Final\bin`**.
- Executar o comando **`jboss-cli.bat --connect`**.





3.4. Instalação do driver JDBC do Apache Derby no WildFly.

3.4.1 Criação de um módulo no WildFly

Todo acesso ao banco de dados ocorre através da API JDBC. No WildFly, para instalar um driver JDBC criamos um módulo para ele. No cliente de administração, execute o comando dado:

```
module add --name=derby.jdbc --resources=C:\Temp\projeto\ServidorDados\db-derby-10.10.2.0-bin\lib\derbyclient.jar --dependencies=javax.api,javax.transaction.api
```

Prompt de Comando - jboss-cli.bat --connect

```
C:\Temp\projeto\wildfly-10.1.0.Final\bin>jboss-cli.bat --connect
[standalone@localhost:9990 /] module add --name=derby.jdbc --resources=C:\Temp\projeto\ServidorDados\db-derby-10.10.2.0-bin\lib\d
erbyclient.jar --dependencies=javax.api,javax.transaction.api
```

Prompt de Comando - jboss-cli.bat --connect

```
C:\Temp\projeto\wildfly-10.1.0.Final\bin>jboss-cli.bat --connect
[standalone@localhost:9990 /] module add --name=derby.jdbc --resources=C:\Temp\projeto\ServidorDados\db-derby-10.10.2.0-bin\lib\d
erbyclient.jar --dependencies=javax.api,javax.transaction.api
[standalone@localhost:9990 /]
```

O módulo é criado na pasta modules na instalação do Wildfly. Note que o nome derby.jdbc é transformado nas subpastas derby\jdbc e o arquivo derbyclient.jar é copiado para esta pasta.

C:\Temp\projeto\wildfly-10.1.0.Final\modules\derby\jdbc\main

Arquivo | Início | Compartilhar | Exibir

Fixar no Acesso rápido | Copiar | Colar | Recortar | Copiar caminho | Colar atalho | Área de Transferência

Mover para | Copiar para | Excluir | Renomear | Organizar

Nova pasta | Novo

Propriedades | Abrir

Selecionar tudo | Selecionar nenhum | Inverter seleção | Selecionar

« OS (C:) › Temp › projeto › wildfly-10.1.0.Final › modules › derby › jdbc › main

Pesquisar ...

Acesso rápido

Área de Trabalho

Nome	Data de modificaç...	Tipo	Tamanho
derbyclient.jar	30/09/2016 07:18	Executable Jar File	573 KB
module.xml	30/09/2016 07:18	Documento XML	1 KB

2 itens

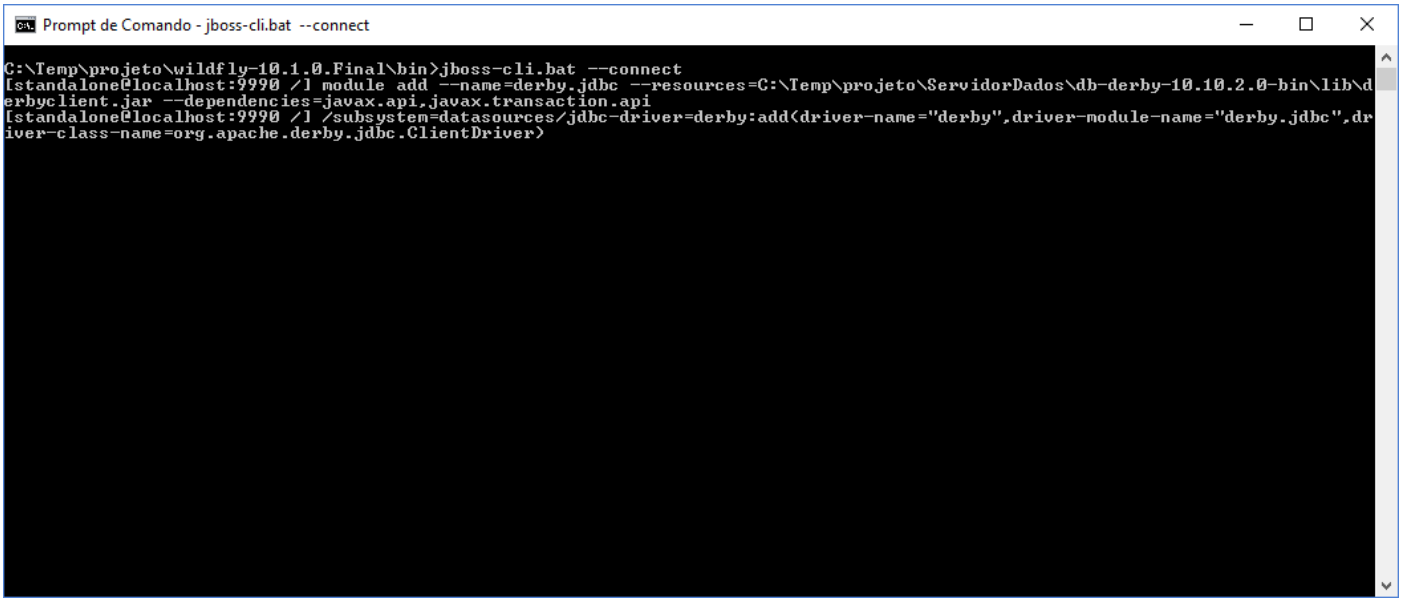
O arquivo module.xml criado descreve o módulo configurado.

```
<?xml version='1.0' encoding='IBM1252'?>


<module xmlns="urn:jboss:module:1.1" name="derby.jdbc">


    <resources>
        <resource-root path="derbyclient.jar"/>
    </resources>


    <dependencies>
        <module name="javax.api"/>
        <module name="javax.transaction.api"/>
    </dependencies>
</module>
```

3.4.2 Registro do driver JDBC do Derby no WildFly

Com o módulo criado, registramos o driver JDBC no servidor de aplicações, executando o comando:

`/subsystem=datasources/jdbc-driver=derby:add(driver-name="derby",driver-module-name="derby.jdbc",driver-class-name=org.apache.derby.jdbc.ClientDriver)`



```
Prompt de Comando - jboss-cli.bat --connect

C:\Temp\projeto\wildfly-10.1.0.Final\bin>jboss-cli.bat --connect
[standalone@localhost:9990 /] module add --name=derby.jdbc --resources=C:\Temp\projeto\ServidorDados\db-derby-10.10.2.0-bin\lib\d
erbyclient.jar --dependencies=javax.api,javax.transaction.api
[standalone@localhost:9990 /] /subsystem=datasources/jdbc-driver=derby:add(driver-name="derby",driver-module-name="derby.jdbc",dr
iver-class-name=org.apache.derby.jdbc.ClientDriver)
{"outcome" => "success"}
[standalone@localhost:9990 /]
```

Este comando irá registrar o driver JDBC do Derby no Wildfly. Podemos verificar esta configuração no arquivo standalone-full.xml (dentro da pasta standalone\configuration da pasta da instalação do wildfly).

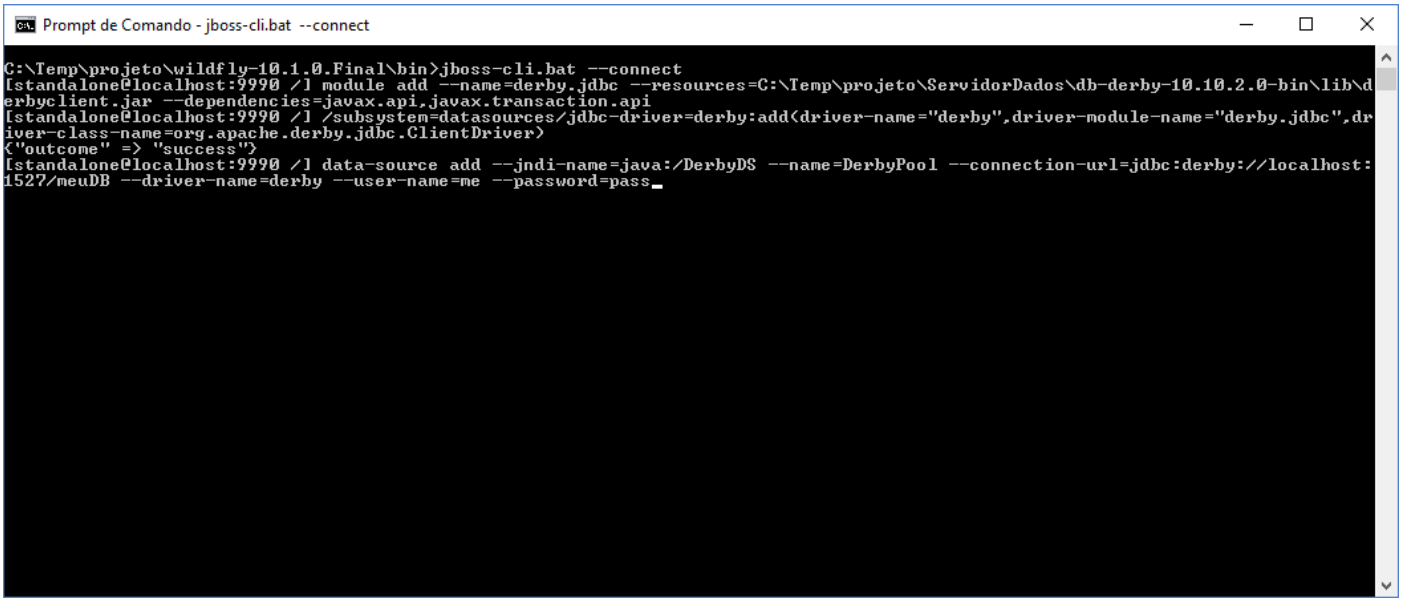
```
...
<subsystem xmlns="urn:jboss:domain:datasources:4.0">
        <datasources>
            <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-
name="ExampleDS" enabled="true" use-java-context="true">
                <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-
1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
                <driver>h2</driver>
                <security>
                    <user-name>sa</user-name>
                    <password>sa</password>
                </security>
            </datasource>
            <drivers>
                <driver name="h2" module="com.h2database.h2">
                    <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-
class>
                </driver>
                <driver name="derby" module="derby.jdbc">
                    <driver-class>org.apache.derby.jdbc.ClientDriver</driver-class>
                </driver>
            </drivers>
        </datasources>
```

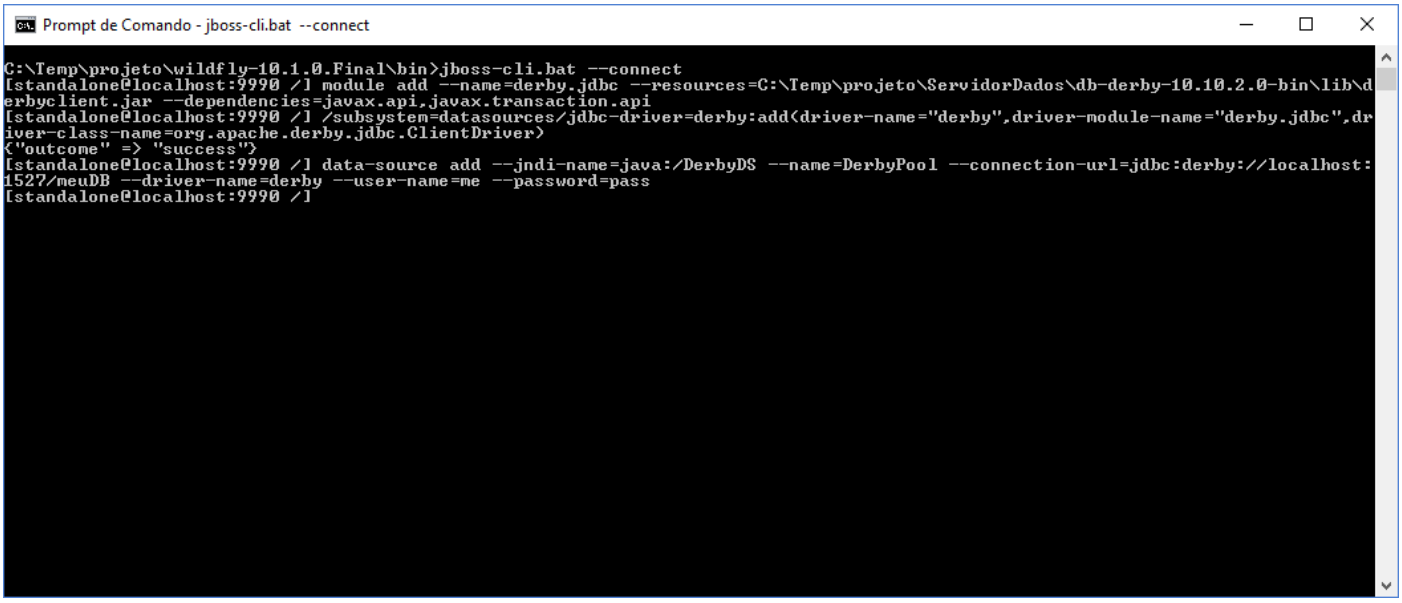
```
        </subsystem>

...
```

3.5. Configuração do Data Source Derby

Para configurar o Data Source Derby (na realidade um pool de conexões ao banco de dados no Derby), execute o seguinte comando:

```
data-source add --jndi-name=java:/DerbyDS --name=DerbyPool --connection-
url=jdbc:derby://localhost:1527/meuDB --driver-name=derby --user-name=me --
password=pass
```





Observe na saída da execução do servidor de aplicações que o data source foi criado:

```
Prompt de Comando - standalone.bat -b 0.0.0.0 -c standalone-full.xml
vemq-remoting protocol handled by http-acceptor acceptor
07:16:45,236 INFO  [org.apache.activemq.artemis.core.server] (ServerService Thread Pool -- 70) AMQ221007: Server is now live
07:16:45,236 INFO  [org.apache.activemq.artemis.core.server] (ServerService Thread Pool -- 70) AMQ221001: Apache ActiveMQ Artemis Mess
ge Broker version 1.1.0.wildfly-017 [nodeID=fc93576f-86f6-11e6-84f9-1713361920ff]
07:16:45,251 INFO  [org.wildfly.extension.messaging-activemq] (ServerService Thread Pool -- 70) WFLYMSGAMQ0002: Bound messaging object
to jndi name java:/ConnectionFactory
07:16:45,298 INFO  [org.jboss.as.connector.deployment] (MSC service thread 1-3) WFLYJCA0007: Registered connection factory java:/JmsXA
07:16:45,314 INFO  [org.apache.activemq.artemis.jms.server] (ServerService Thread Pool -- 68) AMQ121005: Invalid "host" value "0.0.0.0
 detected for "http-connector" connector. Switching to "Ambriel". If this new address is incorrect please manually configure the conne
tor to use the proper one.
07:16:45,314 INFO  [org.wildfly.extension.messaging-activemq] (ServerService Thread Pool -- 68) WFLYMSGAMQ0002: Bound messaging object
to jndi name java:jboss/exported/jms/RemoteConnectionFactory
07:16:45,314 INFO  [org.apache.activemq.artemis.core.server] (ServerService Thread Pool -- 69) AMQ221003: trying to deploy queue jms.q
eue.ExpiryQueue
07:16:45,345 INFO  [org.apache.activemq.artemis.ra] (MSC service thread 1-3) Resource adaptor started
07:16:45,345 INFO  [org.jboss.as.connector.services.resourceadapters.ResourceAdapterActivatorService$ResourceAdapterActivator] (MSC se
vice thread 1-3) IJ020002: Deployed: file://RaActivatoractivemq-ra
07:16:45,345 INFO  [org.wildfly.extension.messaging-activemq] (MSC service thread 1-1) WFLYMSGAMQ0002: Bound messaging object to jndi
ame java:jboss/DefaultJMSConnectionFactory
07:16:45,345 INFO  [org.jboss.as.connector.deployment] (MSC service thread 1-4) WFLYJCA0002: Bound JCA ConnectionFactory [java:/JmsXA]
07:16:45,392 INFO  [org.apache.activemq.artemis.core.server] (ServerService Thread Pool -- 67) AMQ221003: trying to deploy queue jms.q
eue.DLQ
07:16:45,564 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://127.0.0.1:9990/m
nagement
07:16:45,564 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://127.0.0.1:9990
07:16:45,564 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full 10.1.0.Final (WildFly Core 2.2.0.Final) started i
 5901ms - Started 370 of 615 services (408 services are lazy, passive or on-demand)
07:26:43,444 INFO  [org.jboss.as.connector.subsystems.datasources] (management-handler-thread - 5) WFLYJCA0004: Deploying JDBC-complia
t driver class org.apache.derby.jdbc.ClientDriver (version 10.10)
07:26:43,444 INFO  [org.jboss.as.connector.deployers.jdbc] (MSC service thread 1-5) WFLYJCA0018: Started Driver service with driver-na
e = derby
07:32:28,716 INFO  [org.jboss.as.connector.subsystems.datasources] (MSC service thread 1-5) WFLYJCA0001: Bound data source [java:/Derb
DS]
```

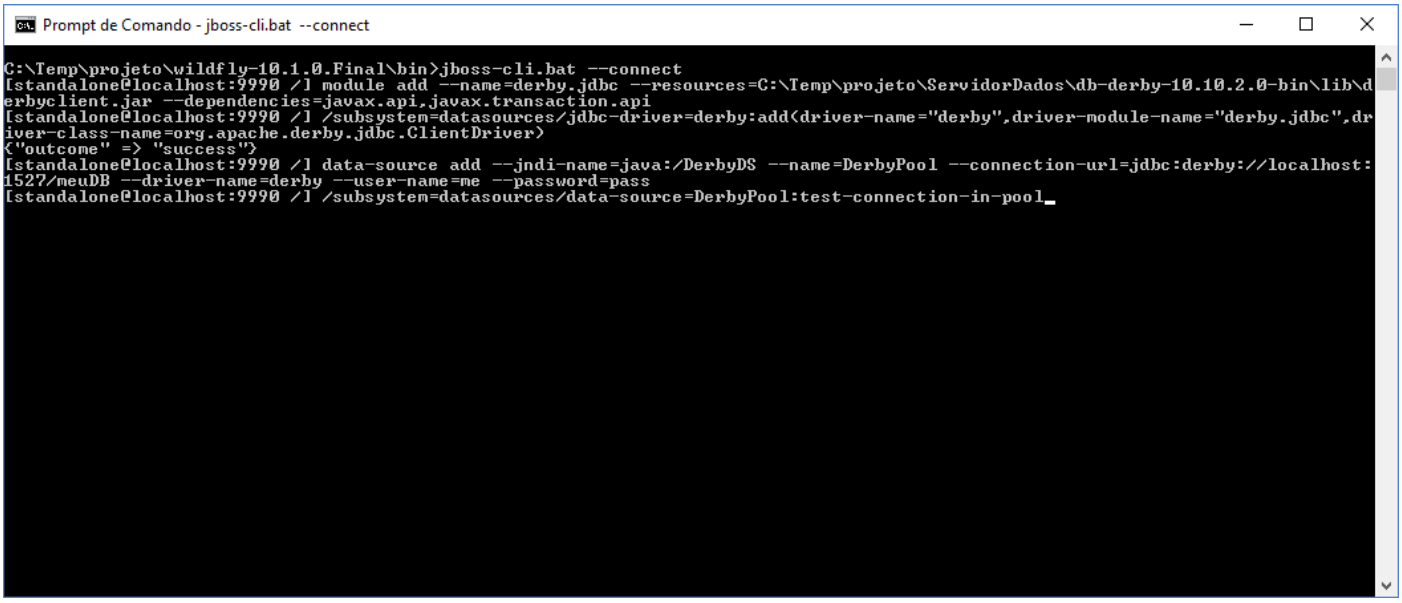
Esta configuração pode ser observada no arquivo: **`C:\Temp\projeto\wildfly-10.1.0.Final\standalone\configuration\standalone-full.xml`**

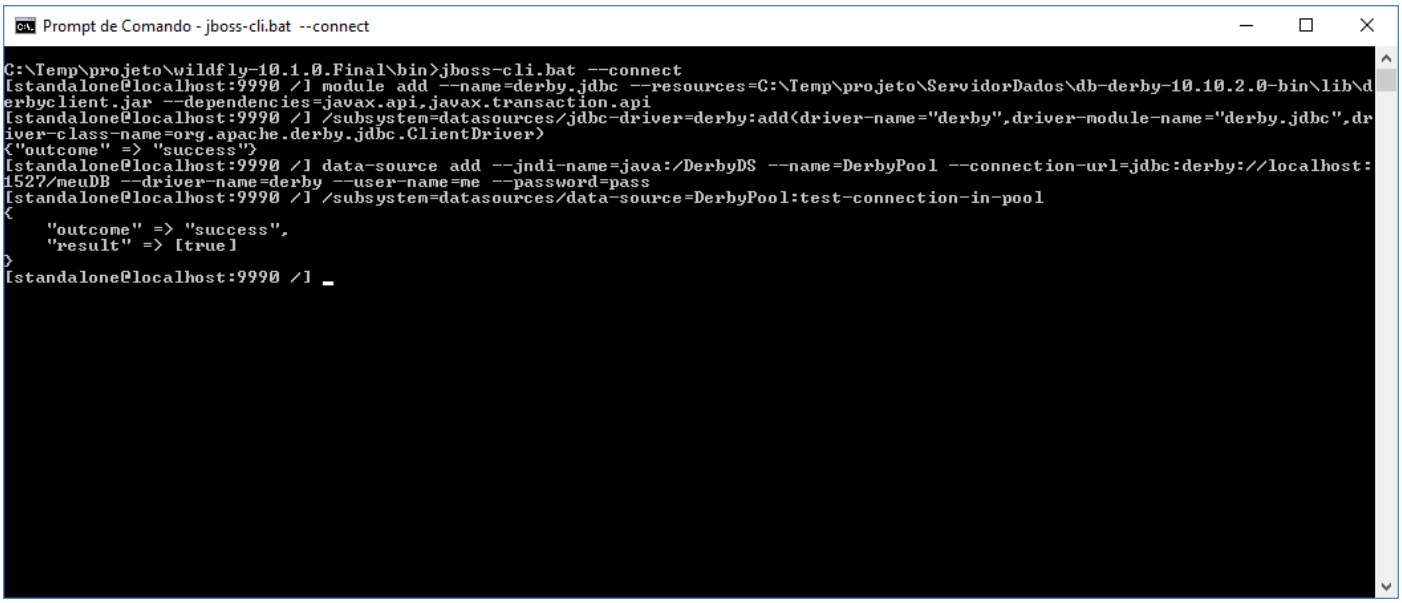
```
...
<subsystem xmlns="urn:jboss:domain:datasources:4.0">
        <datasources>
            <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true">
                <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
                <driver>h2</driver>
                <security>
                    <user-name>sa</user-name>
                    <password>sa</password>
                </security>
            </datasource>
            <datasource jndi-name="java:/DerbyDS" pool-name="DerbyPool">
                <connection-url>jdbc:derby://localhost:1527/meuDB</connection-url>
                <driver>derby</driver>
                <security>
                    <user-name>me</user-name>
                    <password>pass</password>
                </security>
            </datasource>
            <drivers>
```

```
                <driver name="h2" module="com.h2database.h2">
                    <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-
class>
                </driver>
                <driver name="derby" module="derby.jdbc">
                    <driver-class>org.apache.derby.jdbc.ClientDriver</driver-class>
                </driver>
            </drivers>
        </datasources>
    </subsystem>
...
```

Para verificar funcionalmente o pool de conexões criado, execute no cliente de administração o comando:

`/subsystem=datasources/data-source=DerbyPool:test-connection-in-pool`

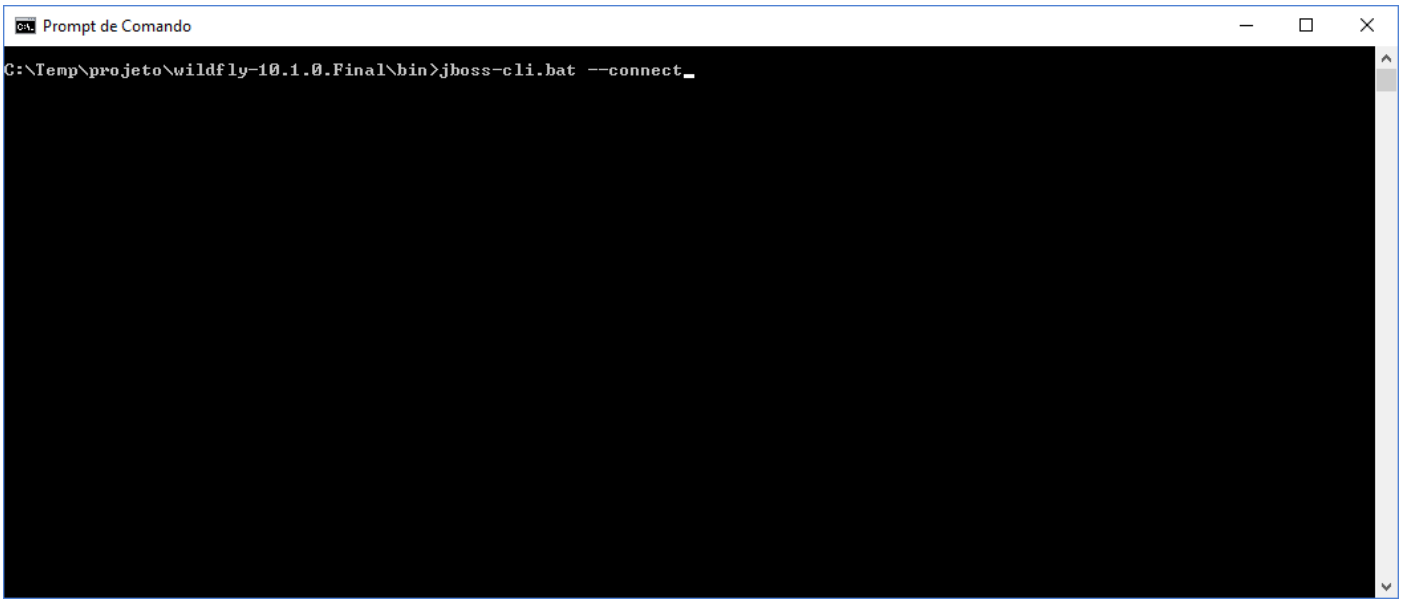


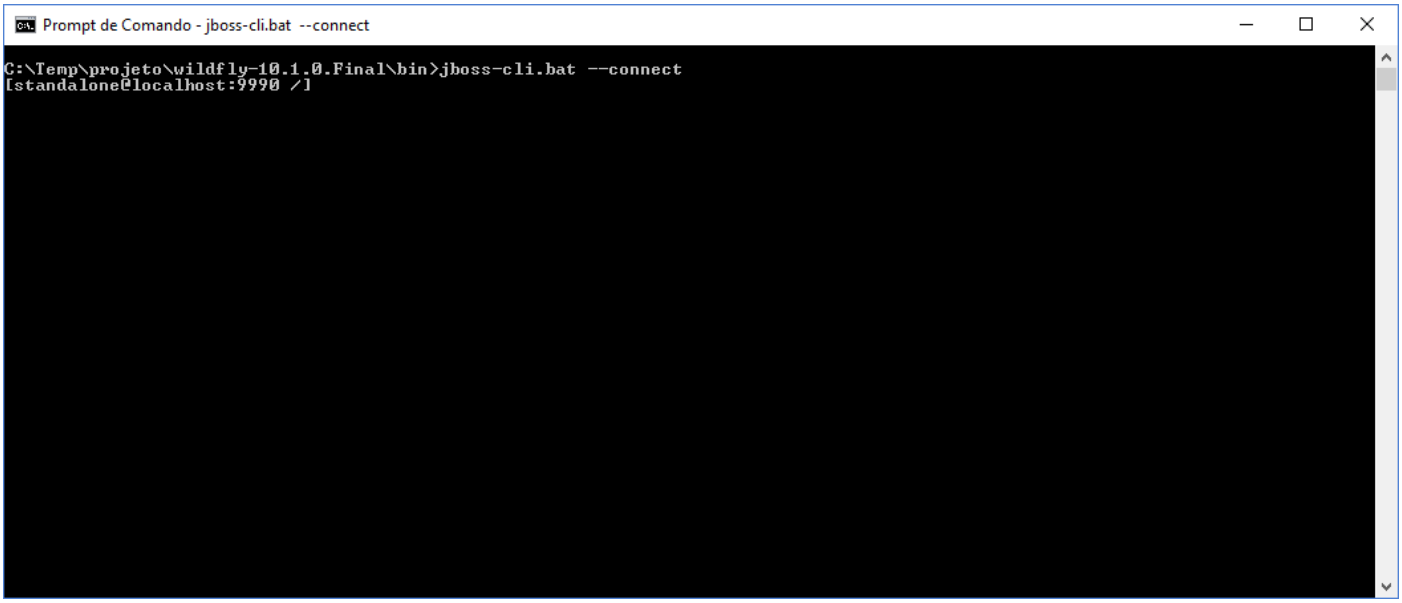


3.6. Configuração das Filas

Para configurar a fila de eventos no JMS, execução do cliente de administração por linha de comando

- Abrir uma janela de prompt de comando.
- Navegar até a pasta **`C:\Temp\projeto\wildfly-10.1.0.Final\bin.`**
- Executar o comando **`jboss-cli.bat --connect`**.





- Criar uma fila chamada eventQueue:
    - Execute o comando abaixo para criar a fila:

```
jms-queue add --queue-address=eventQueue --
entries=java:/jms/queue/eventQueue,java:jboss/exported/jms/queue/eventQueue
```

Prompt de Comando - jboss-cli.bat --connect

```
C:\Temp\projeto\wildfly-10.1.0.Final\bin>jboss-cli.bat --connect
[standalone@localhost:9990 /] jms-queue add --queue-address=eventQueue --entries=java:/jms/queue/eventQueue,java:jboss/e
xported/jms/queue/eventQueue
```

Prompt de Comando - jboss-cli.bat --connect

```
C:\Temp\projeto\wildfly-10.1.0.Final\bin>jboss-cli.bat --connect
[standalone@localhost:9990 /] jms-queue add --queue-address=eventQueue --entries=java:/jms/queue/eventQueue,java:jboss/e
xported/jms/queue/eventQueue
[standalone@localhost:9990 /]
```

- Verifique se a fila foi criada executando o comando abaixo:

  `/subsystem=messaging-activemq/server=default/jms-queue=eventQueue:read-resource`

Prompt de Comando - jboss-cli.bat --connect

```
C:\Temp\projeto\wildfly-10.1.0.Final\bin>jboss-cli.bat --connect
[standalone@localhost:9990 /] jms-queue add --queue-address=eventQueue --entries=java:/jms/queue/eventQueue,java:jboss/e
xported/jms/queue/eventQueue
[standalone@localhost:9990 /] /subsystem=messaging-activemq/server=default/jms-queue=eventQueue:read-resource
```

Prompt de Comando - jboss-cli.bat --connect

```
C:\Temp\projeto\wildfly-10.1.0.Final\bin>jboss-cli.bat --connect
[standalone@localhost:9990 /] jms-queue add --queue-address=eventQueue --entries=java:/jms/queue/eventQueue,java:jboss/e
xported/jms/queue/eventQueue
[standalone@localhost:9990 /] /subsystem=messaging-activemq/server=default/jms-queue=eventQueue:read-resource
{
    "outcome" => "success",
    "result" => {
        "durable" => true,
        "entries" => [
            "java:/jms/queue/eventQueue",
            "java:jboss/exported/jms/queue/eventQueue"
        ],
        "legacy-entries" => undefined,
        "selector" => undefined
    }
}
[standalone@localhost:9990 /]
```

4. Utilização do DataSource

4.1. JPA e Bean de Sessão

- Criar um projeto de biblioteca de classes Java e implementar a classe de entidade Usuario.java. Não se esquecer de incluir na biblioteca do projeto a biblioteca de API Java EE 7.
- Incluir este projeto (biblioteca de classes) nas dependências do módulo EJB.
- Implementar o bean de sessão UsuarioBean e o interceptor LogInterceptor.
- Modifique a tabela do banco de dados:

Será necessário alterar o banco de dados, para que um usuário possua informação de hash de senha de login.

```
ALTER TABLE TB_USUARIO ADD COLUMN HASH VARCHAR(166);

ALTER TABLE TB_USUARIO ADD COLUMN LOGIN VARCHAR(100);


Para uma senha = 'senha' vale o hash abaixo indicado:

update tb_usuario set
HASH='1000:5b42403738353265393232:ab89b0c65354d57bdcf9659a53d410fa0a8b7d0f9dee
270f1e9f51d87d1a9d06bfb5d1e114566b34c6e658f6d4d7d816b90a2841a5e93e7604ee5d5296
3a9135';


update tb_usuario set login = nome;
```

- Mova o arquivo persistence.xml da pasta de configuração do projeto web (AppFrontController) para a pasta de configuração do projeto do módulo EJB (Modulo EJB). Substitua o seu conteúdo com a listagem fornecida para que o Servidor de Aplicação utilize o Data Source configurado anteriormente.
- Apague do projeto web os pacotes mack.dao.exception, mack.dao.usuario e mack.entities.
- Adicione ao projeto da aplicação WEB a dependência ao projeto de biblioteca compartilhada.
- Remova da biblioteca da aplicação WEB o driver JDBC do Derby.
- Modifique o LoginServlet.java de sua aplicação WEB.
- Modifique as implementações dos controllers de sua aplicação.
- Modifique as páginas jsp.
- Altere a página index.html incluindo os novos campos (login e senha).
- Implante a aplicação corporativa e teste.
- Implemente a página de alteração de senha.
- Verifique no Wildfly quantas mensagens foram enfileiradas na eventQueue.

Usuario.java

Crie em um projeto de biblioteca de classes Java.

```
package ejb.entities;


import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;


@Entity

@Table(name="tb_usuario")

public class Usuario {


    @Id

    @Column(name="usuario_id")

    @SequenceGenerator(name="usuarioGenerator",
sequenceName="usuario_id_sequence",allocationSize=1)

    @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="usuarioGenerator")

    private int id;

    @Column(name="nome")

    private String nome;

    @Column(name="sobrenome")

    private String sobrenome;


    @Column(name="login")

    private String login;


    @Column(name="hash")

    private String hash;
```

```
public Usuario() {
}

public Usuario(int id, String nome, String sobrenome,String login, String hash) {
    this.id = id;
    this.nome = nome;
    this.sobrenome = sobrenome;
    this.login=login;
    this.hash=hash;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getSobrenome() {
    return sobrenome;
}

public void setSobrenome(String sobrenome) {
    this.sobrenome = sobrenome;
}

public void setId(int id) {
    this.id = id;
}

public int getId() {
    return this.id;
}
```

```
    public String getLogin() {
        return login;
    }

    public void setLogin(String login) {
        this.login = login;
    }

    public String getHash() {
        return hash;
    }

    public void setHash(String hash) {
        this.hash = hash;
    }

    public String toString() {
        StringBuffer sbResult = new StringBuffer();
        sbResult.append("id = ");
        sbResult.append(id);
        sbResult.append(", nome = ");
        sbResult.append(nome);
        sbResult.append(", sobrenome = ");
        sbResult.append(sobrenome);
        return sbResult.toString();
    }

}
```

UsuarioBean.java

Criado no módulo EJB da aplicação corporativa

```
package ejb.beans;


import ejb.interceptors.LogInterceptor;
import ejb.entities.Usuario;
import java.math.BigInteger;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.security.spec.InvalidKeySpecException;
import java.util.Collection;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.PBEKeySpec;
import javax.ejb.Stateless;
import javax.ejb.LocalBean;
import javax.interceptor.Interceptors;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;


@Stateless
@LocalBean
@Interceptors(LogInterceptor.class)
public class UsuarioBean {


    @PersistenceContext(unitName = "DerbyPU")
    private EntityManager em;


    public Usuario criaUsuario(Usuario u) {
        em.persist(u);
        em.flush();
        em.refresh(u);
```

```
        return u;
    }

    public List<Usuario> list() {
        Query query = em.createQuery("FROM Usuario u");

        List<Usuario> list = query.getResultList();
        return list;
    }

    public Usuario buscaUsuarioPorId(final int id) {
        Usuario u = em.find(Usuario.class, id);
        return u;
    }

    public Collection buscaUsuarioPorNome(final String nome) {
        Query q = em.createQuery("select u from Usuario u where u.nome = :par1");
        q.setParameter("par1", nome);
        Collection result = null;
        result = q.getResultList();
        return result;
    }

    public void removeUsuario(final int id) {
        Usuario u = em.find(Usuario.class, id);
        if (u != null) {
            em.remove(u);
        }
    }

    public void updateUsuario(Usuario user) {
        Usuario u = em.find(Usuario.class, user.getId());
        if (u != null) {
            u.setNome(user.getNome());
            u.setSobrenome(user.getSobrenome());
```

```
        u.setLogin(user.getLogin());
        em.merge(u);
    }
}

public boolean autentica(String user, String senha) {
    Query query = em.createQuery("FROM Usuario u where u.login='" + user + "'");
    List<Usuario> list = query.getResultList();
    if (list.size() != 1) {
        return false;
    }
    Usuario u = list.get(0);
    try {
        if (user.equals(u.getLogin()) && validaSenha(senha, u.getHash())) {
            return true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}

public Usuario alteraSenha(String usuario, String senha, String novaSenha) {

    Query query = em.createQuery("FROM Usuario u where u.login='" + usuario + "'");
    List<Usuario> list = query.getResultList();
    if (list.size() != 1) {
        return null;
    }
    Usuario u = list.get(0);
    try {
        if (usuario.equals(u.getLogin()) && validaSenha(senha, u.getHash())) {
            u.setHash(generateStrongPasswordHash(novaSenha));
            em.persist(u);
            return u;
```

```
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    private static boolean validaSenha(String senhaCandidata, String hashSenha) throws
NoSuchAlgorithmException, InvalidKeySpecException {
        String[] parts = hashSenha.split(":");
        int iterations = Integer.parseInt(parts[0]);
        byte[] salt = fromHex(parts[1]);
        byte[] hash = fromHex(parts[2]);

        PBEKeySpec spec = new PBEKeySpec(senhaCandidata.toCharArray(), salt, iterations,
hash.length * 8);
        SecretKeyFactory skf = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
        byte[] hashCandidato = skf.generateSecret(spec).getEncoded();

        int diff = hash.length ^ hashCandidato.length;
        for (int i = 0; i < hash.length && i < hashCandidato.length; i++) {
            diff |= hash[i] ^ hashCandidato[i];
        }
        return diff == 0;
    }

    public String getHash(String senha){
        try {
            return UsuarioBean.generateStrongPasswordHash(senha);
        } catch (NoSuchAlgorithmException ex) {
            Logger.getLogger(UsuarioBean.class.getName()).log(Level.SEVERE, null, ex);
        } catch (InvalidKeySpecException ex) {
            Logger.getLogger(UsuarioBean.class.getName()).log(Level.SEVERE, null, ex);
        }
        return null;
    }
```

```
    private static String generateStrongPasswordHash(String password) throws
NoSuchAlgorithmException, InvalidKeySpecException {
        int iterations = 1000;
        char[] chars = password.toCharArray();
        byte[] salt = getSalt().getBytes();

        System.out.println("Salt:" + salt.length);
        PBEKeySpec spec = new PBEKeySpec(chars, salt, iterations, 64 * 8);
        SecretKeyFactory skf = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
        byte[] hash = skf.generateSecret(spec).getEncoded();
        return iterations + ":" + toHex(salt) + ":" + toHex(hash);
    }

    private static String getSalt() throws NoSuchAlgorithmException {
        SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
        byte[] salt = new byte[16];
        sr.nextBytes(salt);
        return salt.toString();
    }

    private static String toHex(byte[] array) throws NoSuchAlgorithmException {
        BigInteger bi = new BigInteger(1, array);
        String hex = bi.toString(16);
        int paddingLength = (array.length * 2) - hex.length();
        if (paddingLength > 0) {
            return String.format("%0" + paddingLength + "d", 0) + hex;
        } else {
            return hex;
        }
    }

    private static byte[] fromHex(String hex) throws NoSuchAlgorithmException {
        byte[] bytes = new byte[hex.length() / 2];
        for (int i = 0; i < bytes.length; i++) {
```

```
            bytes[i] = (byte) Integer.parseInt(hex.substring(2 * i, 2 * i + 2), 16);
        }
        return bytes;
    }


}
```

LogInterceptor.java

Criado no módulo EJB da aplicação corporativa

```
package ejb.interceptors;


import javax.annotation.Resource;

import javax.interceptor.AroundInvoke;

import javax.interceptor.InvocationContext;

import javax.jms.Connection;

import javax.jms.ConnectionFactory;

import javax.jms.Destination;

import javax.jms.MessageProducer;

import javax.jms.Session;

import javax.jms.TextMessage;


public class LogInterceptor {


    @Resource(mappedName = "java:/ConnectionFactory")

    private ConnectionFactory connectionFactory;

    @Resource(mappedName = "java:/jms/queue/eventQueue")

    private Destination destination;


    @AroundInvoke

    public Object log(InvocationContext context) throws Exception {

        Connection conn = null;

        Session session = null;

        MessageProducer producer = null;

        TextMessage message = null;

        conn = connectionFactory.createConnection(System.getProperty("username", "jmsUser"),
System.getProperty("password", "jmsUser123!"));

        session = conn.createSession(false, Session.AUTO_ACKNOWLEDGE);

        producer = session.createProducer(destination);

        conn.start();

        message = session.createTextMessage(context.getMethod().getName().toString());

        producer.send(message);

// Fecha conexao.
```

```
        if (conn != null) {
            conn.close();
        }
        System.out.println("---" + context.getMethod());
        return context.proceed();
    }
}
```

persistence.xml

Unidade de persistência configuradas no módulo EJB da aplicação corporativa.

```
<?xml version="1.0" encoding="UTF-8"?>

<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">

  <persistence-unit name="DerbyPU" transaction-type="JTA">

    <jta-data-source>java:/DerbyDS</jta-data-source>

    <class>ejb.entities.Usuario</class>

    <exclude-unlisted-classes>true</exclude-unlisted-classes>

    <properties>

        </properties>

  </persistence-unit>

</persistence>
```

LoginServlet.java

```
package mack.servlets;


import ejb.beans.UsuarioBean;
import java.io.IOException;
import java.io.PrintWriter;
import javax.ejb.EJB;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class LoginServlet extends HttpServlet {

    @EJB
    UsuarioBean ub;

    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {

        // get request parameters for userID and password
        String usuarioRequest = request.getParameter("login");
        String senhaRequest = request.getParameter("senha");



        if(ub.autentica(usuarioRequest,senhaRequest)){
            HttpSession session = request.getSession();
            session.setAttribute("usuario", usuarioRequest);
            //setting session to expiry in 30 mins
            session.setMaxInactiveInterval(30*60);
```

```
            Cookie userName = new Cookie("usuario", usuarioRequest);
            userName.setMaxAge(30*60);
            response.addCookie(userName);
            response.sendRedirect("sucessoLogin.jsp");
        }else{
            RequestDispatcher rd = getServletContext().getRequestDispatcher("/login.html");
            PrintWriter out= response.getWriter();
            out.println("<font color=red>Usuario ou senha incorretos.</font>");
            rd.include(request, response);
        }

    }
}
```

AtualizaController.java

```
package mack.controllers.impl;


import ejb.beans.UsuarioBean;

import ejb.entities.Usuario;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.naming.Context;

import javax.naming.InitialContext;

import mack.controllers.AbstractController;


public class AtualizaController extends AbstractController {


    public void execute() {

        try {

            Logger.getLogger(AtualizaController.class.getName()).log(Level.INFO, null,
"Atualiza");


            Context ctx = new InitialContext();

            UsuarioBean ub = (UsuarioBean)
ctx.lookup("java:global/AppEnterprise/ModuloEJB/UsuarioBean");


            Usuario usuario;

            String id = this.getRequest().getParameter("id");

            String nome = this.getRequest().getParameter("nome");

            String sobrenome = this.getRequest().getParameter("sobrenome");

            String login = this.getRequest().getParameter("login");

            usuario = new Usuario();

            usuario.setId(Integer.parseInt(id));

            usuario.setNome(nome);

            usuario.setSobrenome(sobrenome);

            usuario.setLogin(login);


            usuario = ub.buscaUsuarioPorId((int)Integer.parseInt(id));
```

```
            this.setReturnPage("/mostraUsuario.jsp");
            this.getRequest().setAttribute("usuario", usuario);
        } catch (Exception ex) {
            Logger.getLogger(AtualizaController.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

BuscaController.java

```
package mack.controllers.impl;


import ejb.beans.UsuarioBean;

import ejb.entities.Usuario;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.naming.Context;

import javax.naming.InitialContext;

import mack.controllers.AbstractController;


public class BuscaController extends AbstractController {


    public void execute() {

        try {

            List usuarios = new ArrayList<Usuario>();


            String nome = this.getRequest().getParameter("nome");

            Context ctx = new InitialContext();

            UsuarioBean ub = (UsuarioBean)
ctx.lookup("java:global/AppEnterprise/ModuloEJB/UsuarioBean");


            Usuario usuario;

            usuarios = (List) ub.buscaUsuarioPorNome(nome);

            this.setReturnPage("/listaUsuarios.jsp");

            this.getRequest().setAttribute("usuarios", usuarios);

        } catch (Exception ex) {

            Logger.getLogger(BuscaController.class.getName()).log(Level.SEVERE, null, ex);

        }

    }

}
```

BuscaIdController.java

```
package mack.controllers.impl;


import ejb.beans.UsuarioBean;

import ejb.entities.Usuario;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.naming.Context;

import javax.naming.InitialContext;

import mack.controllers.AbstractController;


public class BuscaIdController extends AbstractController {


    public void execute() {

        try {

            Logger.getLogger(BuscaIdController.class.getName()).log(Level.INFO, null,
"BuscaId");


            String id = this.getRequest().getParameter("id");

            Context ctx = new InitialContext();

            UsuarioBean ub = (UsuarioBean)
ctx.lookup("java:global/AppEnterprise/ModuloEJB/UsuarioBean");


            Usuario usuario;

            usuario = (Usuario) ub.buscaUsuarioPorId(Integer.parseInt(id));

            this.setReturnPage("/mostraUsuario.jsp");

            this.getRequest().setAttribute("usuario", usuario);

        } catch (Exception ex) {

            Logger.getLogger(BuscaIdController.class.getName()).log(Level.SEVERE, null, ex);

        }

    }


}
```

ListaController.java

```
package mack.controllers.impl;


import ejb.beans.UsuarioBean;

import ejb.entities.Usuario;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.naming.Context;

import javax.naming.InitialContext;

import mack.controllers.AbstractController;



public class ListaController extends AbstractController {


    public void execute() {

        try {

            Logger.getLogger(ListaController.class.getName()).log(Level.INFO, null, "Lista");

             Context ctx = new InitialContext();

            UsuarioBean ub = (UsuarioBean)
ctx.lookup("java:global/AppEnterprise/ModuloEJB/UsuarioBean");


            List usuarios = new ArrayList<Usuario>();

            usuarios = (List) ub.list();

            this.setReturnPage("/listaUsuarios.jsp");

            this.getRequest().setAttribute("usuarios", usuarios);

        } catch (Exception ex) {

            Logger.getLogger(ListaController.class.getName()).log(Level.SEVERE, null, ex);

        }

    }

}
```

NovoController.java

```
package mack.controllers.impl;


import ejb.beans.UsuarioBean;
import ejb.entities.Usuario;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import mack.controllers.AbstractController;


public class NovoController extends AbstractController {


    public void execute() {
        try {
            Logger.getLogger(NovoController.class.getName()).log(Level.INFO, null, "Novo");
            Usuario usuario;
            String nome = this.getRequest().getParameter("nome");
            String sobrenome = this.getRequest().getParameter("sobrenome");
            String login = this.getRequest().getParameter("login");
            String senha = this.getRequest().getParameter("senha");


            Context ctx = new InitialContext();
            UsuarioBean ub = (UsuarioBean)
ctx.lookup("java:global/AppEnterprise/ModuloEJB/UsuarioBean");


            usuario = new Usuario();
            usuario.setNome(nome);
            usuario.setSobrenome(sobrenome);
            usuario.setLogin(login);
            usuario.setHash(ub.getHash(senha));
            usuario = ub.criaUsuario(usuario);
            this.setReturnPage("/mostraUsuario.jsp");
            this.getRequest().setAttribute("usuario", usuario);
```

```
        } catch (Exception ex) {
            Logger.getLogger(NovoController.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

RemoveController.java

```
package mack.controllers.impl;


import ejb.beans.UsuarioBean;

import ejb.entities.Usuario;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.naming.Context;

import javax.naming.InitialContext;

import mack.controllers.AbstractController;


public class RemoveController extends AbstractController {


    public void execute() {

        try {

            Logger.getLogger(RemoveController.class.getName()).log(Level.INFO, null, "Lista");


            String id = this.getRequest().getParameter("id");


            List usuarios = new ArrayList<Usuario>();

            Context ctx = new InitialContext();

            UsuarioBean ub = (UsuarioBean)
ctx.lookup("java:global/AppEnterprise/ModuloEJB/UsuarioBean");

            ub.removeUsuario(Integer.parseInt(id));

            usuarios = (List) ub.list();

            this.setReturnPage("/listaUsuarios.jsp");

            this.getRequest().setAttribute("usuarios", usuarios);

        } catch (Exception ex) {

            Logger.getLogger(RemoveController.class.getName()).log(Level.SEVERE, null, ex);

        }

    }

}
```

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Pagina Principal</title>
    </head>
    <body>
        <h1>Exemplo de uso de FrontController</h1>
        <a href="FrontControllerServlet?control=Lista">Lista</a>

        <form action="FrontControllerServlet?control=Busca" method="post">
            Nome de usuario: <input type="text" name="nome">
            <br>
            <input type="submit" value="Busca">
        </form>
        <form action="FrontControllerServlet?control=BuscaId" method="post">
            Id do Usuario: <input type="text" name="id">
            <br>
            <input type="submit" value="Busca por Id">
        </form>
        <form action="FrontControllerServlet?control=Remove" method="post">
            Id do Usuario: <input type="text" name="id">
            <br>
            <input type="submit" value="Remove">
        </form>
         <form action="FrontControllerServlet?control=Atualiza" method="post">
            Id do Usuario: <input type="text" name="id">
            <br>
            Nome de usuario: <input type="text" name="nome">
            <br>
            Sobrenome de usuario: <input type="text" name="sobrenome">
             <br>
            Login do usuario: <input type="text" name="login">
            <br>
```

```
            <input type="submit" value="Atualiza">
        </form>
        <form action="FrontControllerServlet?control=Novo" method="post">


            Nome de usuario: <input type="text" name="nome">
            <br>
            Sobrenome de usuario: <input type="text" name="sobrenome">
            <br>
             Login de usuario: <input type="text" name="login">
            <br>
             Senha do usuario: <input type="password" name="senha">
            <br>
            <input type="submit" value="Cria">
        </form>
    </body>
</html>
```

listaUsuarios.jsp

```
<%@page import="ejb.entities.Usuario"%>
<%@page import="java.util.List"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type"
              content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Usuarios!</h1>
        <%
            List<Usuario> usuarios
                    = (List<Usuario>) request.getAttribute("usuarios");
        %>
        <%if (usuarios.size() > 0) { %>
        <table>
            <% for (Usuario u : usuarios) {%>
            <tr>
                <td><%=u.getId()%></td>
                <td><%=u.getNome()%></td>
                <td><%=u.getSobrenome()%></td>
                <td><%=u.getLogin()%></td>
                <td><%=u.getHash()%></td>
            </tr>
            <%}%>
        </table>
        <%}%>
    </body>
</html>
```

mostraUsuario.jsp

```
<%@page import="ejb.entities.Usuario"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type"
              content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Usuario!</h1>
        <%
            Usuario u = (Usuario) request.getAttribute("usuario");
        %>

        <%if (u!=null) { %>
        <table>
            <tr>
                <td><%=u.getId()%></td>
                <td><%=u.getNome()%></td>
                <td><%=u.getSobrenome()%></td>
                <td><%=u.getLogin()%></td>
                <td><%=u.getHash()%></td>
            </tr>
        </table>
        <%}%>
    </body>
</html>
```