

Manual de Instalação

Fornece as orientações necessárias para efetuar a instalação deste produto em ambiente Linux.

Versão 1.5
27/07/2015



Histórico de Revisão

Data	Versão	Descrição	Autor
13/05/2015	1.3	Alteração do conteúdo do documento.	Otávio Lemos
14/05/2015	1.4	Alteração do layout do documento.	Rosana Silva Godinho; Ciro Anacleto
27/07/2015	1.5	Inserção de configuração da Central de Autenticação.	Ciro Anacleto; Hellen de Sousa Lemes
19/02/2016	1.6	Atualização das informações referente a quantidade de memória do Jboss.	Rosana Silva Godinho

Sumário

1. INTRODUÇÃO.....	5
1.1. Propósito.....	5
1.2. Citsmart GRP	5
2. DEPENDÊNCIAS.....	5
3. DOWNLOAD DAS DEPENDÊNCIAS.....	6
4. FIREWALL E SELINUX	6
5. INSTALAÇÃO.....	7
5.1. JBoss	7
5.2. JDK.....	7
5.3. MSTT Core Fonts.....	7
5.4. Postgresql 9.3	7
5.5. Módulo JDBC do banco Postgres	9
5.6. Módulo Hibernate (atualizado) do Jboss.....	9
5.7. Módulo Javassist (atualizado) do Jboss.....	10
5.8. Configuração do Standalone - Jboss 7.1.1	10
5.8.1. Arquivo standalone.conf.....	11
5.8.2. Arquivo standalone.xml.....	12
5.9. Configurando FQDN Hosts.....	15
6. DEPLOY DO CITSMART-GRP.....	15
6.1. Copiando arquivos de deploy	15
6.2. Configurando central de autenticação	15
7. BANCO DE DADOS POSTGRES.....	16
7.1. Verificação de tabelas criadas	17

7.2. Inserção de dados na tabela.....	18
8. START AUTOMÁTICO DO JBOSS.....	18
8.1. Criar usuário citsmart-grp	18
8.2. Mudar permissões do Jboss.....	19
8.3. Criar daemon para o citsmart-grp	19
8.4. Configurando inicialização da daemon	20
9. INICIALIZANDO A SOLUÇÃO	20
10. TESTE DE ACESSO	21

1. INTRODUÇÃO

1.1. Propósito

Este Manual tem a finalidade de fornecer as orientações necessárias para efetuar a instalação do Citsmart-GRP no ambiente Linux.

1.2. Citsmart GRP

O Citsmart GRP é uma plataforma de gerenciamento integrado inteligente, que visa unificar informações dos departamentos em diversos órgãos em âmbito federal. O nome Citsmart GRP vem da junção dos acrônimos Central IT e Government Resource Planing (do inglês) com propósito semelhante aos sistemas ERP (Enterprise Resource Planing) porém com foco estritamente governamental. Uma tradução livre para o nome em inglês GRP seria Sistema Integrado de Gerenciamento Governamental.

Este sistema consiste no módulo de gerenciamento de bens patrimoniais, possibilitando gerenciar com eficiência aquisições, numerações de patrimônios, transferências, detentores de bens, desfazimento e demais operações inerentes.

2. DEPENDÊNCIAS

- ✓ CentOS x86_64 a partir da versão 6
- ✓ JBOSS AS 7.1.1;
- ✓ JDK Oracle 7u;
- ✓ MTT Core Fonts;
- ✓ PostgreSQL 9.3;

- ✓ Módulo JDBC do banco Postgres;
- ✓ Módulo Hibernate (atualizado) do Jboss;
- ✓ Módulo Javassist (atualizado) do Jboss;
- ✓ Configuração do Standalone do Jboss 7.1.1.

3. DOWNLOAD DAS DEPENDÊNCIAS

CentOS x86_64 (6 ou 7): <http://www.centos.org>

JBoss AS 7.1.1: <http://download.jboss.org/jbossas/7.1/jboss-as-7.1.1.Final/jboss-as-7.1.1.Final.zip>

JDK Oracle (baixar a última da versão 7 64bits):

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

MSTT Core Fonts: <ftp://195.220.108.108/linux/sourceforge/r/re/repo-1c/RPMS/msttcorefonts-2.5-1.noarch.rpm>

Postgresql 9.3 (instalado de repositório oficial):

CentOS 7 x86_64: http://yum.postgresql.org/9.3/redhat/rhel-7-x86_64/pgdg-centos93-9.3-1.noarch.rpm

CentOS 6 x86_64: http://yum.postgresql.org/9.3/redhat/rhel-6-x86_64/pgdg-centos93-9.3-1.noarch.rpm

Módulo JDBC PostgreSQL: <https://jdbc.postgresql.org/download/postgresql-9.3-1103.jdbc41.jar>

Módulo Hibernate:

<http://central.maven.org/maven2/org/hibernate/javax/persistence/hibernate-jpa-2.1-api/1.0.0.Final/hibernate-jpa-2.1-api-1.0.0.Final.jar>

Módulo Javassist: <http://central.maven.org/maven2/org/javassist/javassist/3.18.1-GA/javassist-3.18.1-GA.jar>

4. FIREWALL E SELINUX

Esse documento não visa entregar segurança para o ambiente.

O *firewall* e o *selinux* devem ser desabilitados para correto funcionamento.

5. INSTALAÇÃO

Toda a instalação será feita em `/opt`, seguindo o padrão do JBoss 7. Considerando que estou dentro do diretório que possui todas as dependências solicitadas.

5.1. JBoss

```
# unzip-x jboss-as-7.1.1.Final.zip-d /opt/
```

5.2. JDK

```
# rpm-Uvh jdk-7u79-linux-x64.rpm
```

5.3. MTT Core Fonts

```
# rpm-Uvh msttcorefonts-2.5-1.noarch.rpm
```

5.4. Postgresql 9.3

CentOS 7 x86_64

```
# rpm -Uvh http://yum.postgresql.org/9.3/redhat/rhel-7-x86\_64/pgdg-centos93-9.3-1.noarch.rpm
```

CentOS 6 x86_64

```
# rpm -Uvh http://yum.postgresql.org/9.3/redhat/rhel-6-x86\_64/pgdg-centos93-9.3-1.noarch.rpm
```

-----Após instalar repo conforme distro-----

Instalar postgres 9.3 do repo oficial:

```
# yum-y install postgresql93-server postgresql93-contrib
```

Inicializar as Dbs:

```
# sudo-u postgres /usr/pgsql-9.3/bin/initdb-D /var/lib/pgsql/9.3/data/
```

Iniciar o postgres:

```
# sudo-u postgres /usr/pgsql-9.3/bin/postgres-D /var/lib/pgsql/9.3/data/ &
```

Criar usuário citgrp:

```
# sudo-u postgres psql-c "create user citgrp with password 'SENHA'"
```

Criar database citgrp e definir o usuário citgrp como owner:

```
# sudo-u postgres psql-c "create database citgrp with owner citgrp encoding 'UTF8'  
tablespace pg_default"
```

Obs: A mensagem “*Could not change directory*” irá aparecer na execução do *sudo* com o usuário *postgres*.

Apenas uma informação de que o usuário postgres não possui permissão de acesso no diretório que está sendo executado o sudo. Como estamos apenas criando usuário e database não existem problemas.

Habilitando Início Automático Postgres:

CentOS 6:

```
# chkconfig postgresql-9.3 on
```

CentOS 7:

```
# systemctl enable postgresql-9.3
```


5.5. Módulo JDBC do banco Postgres

1. Criar diretório para o módulo postgres:

```
# mkdir -p /opt/jboss-as-7.1.1.Final/modules/org/postgres/main/
```

2. Copiar módulo postgres:

```
# cp postgresql-9.3-1103.jdbc41.jar /opt/jboss-as-7.1.1.Final/modules/org/postgres/main/
```

3. Criar arquivo XML e inserir somente conteúdo:

```
# vi /opt/jboss-as-7.1.1.Final/modules/org/postgres/main/module.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.postgres">
  <resources>
    <resource-root path="postgresql-9.3-1103.jdbc41.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
    <module name="javax.servlet.api" optional="true"/>
  </dependencies>
</module>
```

5.6. Módulo Hibernate (atualizado) do Jboss

1. Apagar o módulo antigo do hibernate:

```
# rm -rf /opt/jboss-as-7.1.1.Final/modules/javax/persistence/api/main/*
```

2. Copiar módulo novo:

```
# cp hibernate-jpa-2.1-api-1.0.0.Final.jar /opt/jboss-as-7.1.1.Final/modules/javax/persistence/api/main/
```

3. Criar XML e inserir somente o conteúdo:

```
# vi /opt/jboss-as-7.1.1.Final/modules/javax/persistence/api/main/module.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="javax.persistence.api">
  <dependencies>
    <module name="javax.api" export="true"/>
  </dependencies>
  <resources>
    <resource-root path="hibernate-jpa-2.1-api-
      1.0.0.Final.jar"/>
  </resources>
</module>
```

5.7. Módulo Javassist (atualizado) do Jboss

1. Apagar o módulo antigo do javassist:

```
# rm -rf /opt/jboss-as-7.1.1.Final/modules/org/javassist/main/*
```

2. Copiar módulo novo:

```
# cp javassist-3.18.1-GA.jar /opt/jboss-as-
7.1.1.Final/modules/org/javassist/main/
```

3. Criar XML e inserir somente conteúdo:

```
# vi /opt/jboss-as-7.1.1.Final/modules/org/javassist/main/module.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.javassist">
  <properties>
    <property name="jboss.api" value="private"/>
  </properties>
  <resources>
    <resource-root path="javassist-3.18.1-GA.jar"/>
  </resources>
</module>
```

5.8. Configuração do Standalone - Jboss 7.1.1

Modificação de 2 arquivos.

5.8.1. Arquivo standalone.conf

vi /opt/jboss-as-7.1.1.Final/bin/standalone.conf

1. Apagar as linhas 50, 51 e 52:

50	JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
51	JAVA_OPTS="\$JAVA_OPTS -Djboss.modules.system.pkgs=\$JBOSSE_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
52	JAVA_OPTS="\$JAVA_OPTS -Djboss.server.default.config=standalone.xml"

2. Inserir no lugar:

```
JAVA_OPTS="-Xms1024m -Xmx3072m -Xss32m -XX:MaxPermSize=1024m -XX:+UseParallelGC -XX:+UseNUMA -XX:+HeapDumpOnOutOfMemoryError -XX:ParallelGCThreads=4"
```

```
JAVA_OPTS="$JAVA_OPTS -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true"
```

```
JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBOSSE_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
```

```
JAVA_OPTS="$JAVA_OPTS -Djboss.server.default.config=standalone.xml"
```

Obs.: A opção **-Xmx3072** deve ser configurada conforme o total de memória de seu host.

3. No final devemos ter:

49	if ["x\$JAVA_OPTS" = "x"]; then
50	JAVA_OPTS="-Xms1024m -Xmx3072m -Xss32m -XX:MaxPermSize=1024m -XX:+UseParallelGC -XX:+UseNUMA -XX:+HeapDumpOnOutOfMemoryError -XX:ParallelGCThreads=4"
51	JAVA_OPTS="\$JAVA_OPTS -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true"

52	<code>JAVA_OPTS="\$JAVA_OPTS -Djboss.modules.system.pkgs=\$JBoss_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"</code>
53	<code>JAVA_OPTS="\$JAVA_OPTS -Djboss.server.default.config=standalone.xml"</code>
54	<code>else</code>
55	<code>echo "JAVA_OPTS already set in environment; overriding default settings with values: \$JAVA_OPTS"</code>
56	<code>if</code>

5.8.2. Arquivo standalone.xml

vi /opt/jboss-as-7.1.1.Final/standalone/configuration/standalone.xml

1. Inserir entre a linha 28 e 29 (após *extensions* e antes de *management*):

```
<system-properties>

  <property name="org.apache.catalina.connector.URI_ENCODING"
    value="UTF-8"/>

  <property name=
    "org.apache.catalina.connector.USE_BODY_ENCODING_FOR_QUERY_STRING"
    value="true"/>

  <property
    name="org.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING"
    value="false"/>

  <property name="org.apache.tomcat.util.http.Parameters.MAX_COUNT"
    value="2000"/>

  <property
    name="org.apache.coyote.http11.Http11Protocol.COMPRESSION"
    value="on"/>

  <property
    name="org.apache.coyote.http11.Http11Protocol.COMPRESSION_MIME_TYPES"
    value="text/javascript, text/css, text/html, text/xml, text/json, image/
    png, image/jpg, image/jpeg, image/gif"/>

</system-properties>
```

2. No final devemos ter:

28	<code></extensions></code>
29	<code><property name="org.apache.catalina.connector.URI_ENCODING" value="UTF-8"/></code>
30	<code><property name="org.apache.catalina.connector.USE_BODY_ENCODING_FOR_QUERY_STRING" value="true"/></code>
31	<code><property name="org.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING" value="false"/></code>
32	<code><property name="org.apache.tomcat.util.http.Parameters.MAX_COUNT" value="2000"/></code>
33	<code><property name="org.apache.coyote.http11.Http11Protocol.COMPRESSION" value="on"/></code>
34	<code><property name="org.apache.coyote.http11.Http11Protocol.COMPRESSION_MIME_TYPES" value="text/javascript,text/css,text/html,text/xml,text/json,image/png,image/jpeg,image/gif"/></code>
35	<code></system-properties></code>
36	<code><management></code>

Timeout para deploy

Adicionar timeout na linha 117 aproximadamente:

```
<deployment-scanner path="deployments" relative-to="jboss.server.base.dir"
scan-interval="5000" deployment-timeout="1000"/>
```

Datasources

Apagar configuração de datasource de exemplo e adicionar a conexão com o Postgres:

Apagar da Linha 100 a linha 113 aproximadamente (<datasources> até </datasources>):

Adicionar as linhas abaixo (Alterar conforme ambiente, a linha:

<password>**SENHA**</password>);

```
<datasources>

  <!-- PostgreSQL cit-portal-web -->
  <datasource jndi-name="java:/env/jdbc/cit-portal" pool-
name="/env/jdbc/cit-portal" enabled="true" use-java-
context="true">
    <connection-
url>jdbc:postgresql://127.0.0.1:5432/citgrp</connection-
url>
    <driver>postgres</driver>
    <pool>
      <min-pool-size>10</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <prefill>true</prefill>
    </pool>
    <security>
      <user-name>citgrp</user-name>
      <password>SENHA</password>
    </security>
    <timeout>
      <blocking-timeout-millis>60000</blocking-timeout-
millis>
      <idle-timeout-minutes>1</idle-timeout-minutes>
    </timeout>
  </datasource>

  <!-- PostgreSQL citgrp-patrimonio -web -->
  <datasource jndi-name="java:/env/jdbc/citgrp-patrimonio"
pool-name="/env/jdbc/citgrp-patrimonio" enabled="true" use-
java-context="true">
    <connection-
url>jdbc:postgresql://127.0.0.1:5432/citgrp</connection-
url>
    <driver>postgres</driver>
    <pool>
      <min-pool-size>10</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <prefill>true</prefill>
    </pool>
    <security>
      <user-name>citgrp</user-name>
      <password>SENHA</password>
    </security>
    <timeout>
      <blocking-timeout-millis>60000</blocking-timeout-
millis>
      <idle-timeout-minutes>1</idle-timeout-minutes>
```

```
        </timeout>
    </datasource>
    <drivers>
        <driver name="postgres" module="org.postgres">
            <xa-datasource-class>org.postgresql.Driver</xa-
            datasource-class>
        </driver>
    </drivers>
</datasources>
```

5.9. Configurando FQDN Hosts

```
# HOSTNAME=`hostname`;usr/bin/sed -i '1,$s/localhost/localhost '$HOSTNAME'/'
/etc/hosts
```

6. DEPLOY DO CITSMART-GRP

6.1. Copiando arquivos de deploy

```
# unzip-x citsmart-GRP-VB-1.0.0-BETA.zip
# cp *.war /opt/jboss-as-7.1.1.Final/standalone/deployments/
```

6.2. Configurando central de autenticação

Juntamente com os arquivos de deploy da aplicação existe o projeto CAS (Servidor de Autenticação Central) que deve ser configurado para que seja possível acessar o sistema. Para a devida configuração edite o arquivo `cas.properties`:

Para autenticação via Banco de Dados:

```
cas-db-1.0.0-RELEASE.war\WEB-INF\cas.properties
```

Linhas 22, 33, 34, 35, 36 respectivamente:

```
server.name=https://localhost:8443
cas.database.jdbc.driver=org.postgresql.Driver
cas.database.jdbc.url=jdbc:postgresql://localhost:5432/citgr
cas.database.jdbc.user=postgres
cas.database.jdbc.password=postgres
```

Para este caso de autenticação em banco de dados é importante que essa autenticação seja feita na mesma base de dados configurada anteriormente no Jboss.

Para autenticação LDAP:

```
cas-ldap-1.0.0-RELEASE.war\WEB-INF\cas.properties
```

Linhas 20, 21, 22, 23, 27, respectivamente:

```
ldap.host.name=ldap://<IP>:<PORTA>
ldap.manager.password=<PASSWORD>
ldap.userDn=<cn=citsmart,cn=Users,dc=cit,dc=local>
ldap.search.filter=<%u@cit.local>
server.name=<http://localhost:8080>
```

Para este caso de autenticação em servidor LDAP é importante que seja alterado os dados marcados com <>. Acima é dado alguns exemplos.

7. BANCO DE DADOS POSTGRES

Para criação de tabelas devemos subir o Jboss:

```
# cd /opt/jboss-as-7.1.1.Final/bin/
# chmod +x *.sh
# ./standalone.sh
```

Aguardar mensagem abaixo:

```
INFO [org.jboss.as.server] (DeploymentScanner-threads- 2) JBAS018559: Deployed "cit-
```


portal-web.war"

INFO [org.jboss.as.server] (DeploymentScanner-threads- 2) JBAS018559: **Deployed "citgrp-patrimonio-web.war"**

Após a mensagem acima ser apresentada no console, parar o Jboss com um Ctrl+c, voltando ao terminal.

A mensagem abaixo deve ser apresentada confirmando a parada do Jboss.

INFO [org.jboss.as] JBAS015950: JBoss AS 7.1.1.Final "Brontes" stopped

7.1. Verificação de tabelas criadas

1. Entre no PSQL:

```
# sudo-u postgres psql
```

2. Digite o \l para listar as databases:

```
postgres=# \l
```

List of databases

Name	Owner	Encoding	Collate	Ctype	Access privileges
citgrp	citgrp	UTF8	pt_BR.UTF-8	pt_BR.UTF-8	
postgres	postgres	UTF8	pt_BR.UTF-8	pt_BR.UTF-8	
template0	postgres	UTF8	pt_BR.UTF-8	pt_BR.UTF-8	=c/postgres + postgres=CtC/postgres
template1	postgres	UTF8	pt_BR.UTF-8	pt_BR.UTF-8	=c/postgres + postgres=CtC/postgres

(4 rows)

3. Conecte na base do citgrp com \c citgrp:

```
postgres=# \c citgrp;
```

You are now connected to database "citgrp" as user "postgres".

```
citgrp=#
```

4. Exiba as tabelas criadas com o \d:

```
citgrp=# \d
```

List of relations

Schema	Name	Type	Owner
public	accessrole	table	citgrp
public	accessrole_id_seq	sequence	citgrp
public	adicaobemprincipal	table	citgrp
diversas tabelas....			

5. Caso as tabelas existam, você está apto a continuar. Caso contrário reinicie o procedimento.

Para sair do PSQL, basta digitar \q:

```
citgrp=# \q
```

7.2. Inserção de dados na tabela

Após todos os passos estarem OK, adicione dados nas tabelas criadas do Postgres:

```
# sudo psql-U citgrp-W-d citgrp-f atualizacao.sql
```

Senha para usuário citgrp:

Caso não apareça nenhuma mensagem de erro seu sistema está pronto.

8. START AUTOMÁTICO DO JBOSS

8.1. Criar usuário citsmart-grp

```
# useradd citsmart-grp
```

8.2. Mudar permissões do Jboss

```
# chown-R citsmart-grp:citsmart-grp /opt/jboss-as-7.1.1.Final
```

8.3. Criar daemon para o citsmart-grp

```
# vi /etc/init.d/citsmart-grp
```

```
#!/bin/bash
# description: Citsmart-GRP Start Stop Restart
# processname: citsmart-grp
# chkconfig: 345 99 01
JAVA_HOME="/usr/java/jdk1.7.0_79"
export JAVA_HOME
PATH=$JAVA_HOME:$PATH
export PATH
JBOSS_HOME=/opt/jboss-as-7.1.1.Final
export JBOSS_HOME
case $1 in
start)
rm -r /opt/jboss-as-7.1.1.Final/standalone/log
rm -r /opt/jboss-as-7.1.1.Final/standalone/tmp
rm -r /opt/jboss-as-7.1.1.Final/standalone/data
sleep 3
/usr/bin/sudo -u citsmart-grp $JBOSS_HOME/bin/standalone.sh -
Djboss.bind.address=0.0.0.0 -
Djboss.bind.address.management=0.0.0.0 &
;;
stop)
/usr/bin/sudo -u citsmart-grp $JBOSS_HOME/bin/jboss-cli.sh --
connect command=:shutdown
;;
restart)
```

```
/usr/bin/sudo -u citsmart-grp $JBOSS_HOME/bin/jboss-cli.sh --
connect command=:shutdown
echo "aguarde 30 segundos..."
sleep 30
rm -r /opt/jboss-as-7.1.1.Final/standalone/log
rm -r /opt/jboss-as-7.1.1.Final/standalone/tmp
rm -r /opt/jboss-as-7.1.1.Final/standalone/data
sleep 3
/usr/bin/sudo -u citsmart-grp $JBOSS_HOME/bin/standalone.sh -
Djboss.bind.address=0.0.0.0
-Djboss.bind.address.management=0.0.0.0 &
;;
*)
echo "Utilizar /etc/init.d/citsmart-grp <start/restart/stop>"
echo "Para START: /etc/init.d/citsmart-grp start"
echo "Para RESTART: /etc/init.d/citsmart-grp restart"
echo "Para STOP: /etc/init.d/citsmart-grp stop"
;;
esac
exit 0
```

8.4. Configurando inicialização da daemon

```
# chmod 700 /etc/init.d/citsmart-grp
# chkconfig citsmart-grp on
# chkconfig--list citsmart-grp
citsmart-grp 0:não 1:não 2:sim 3:sim 4:sim 5:sim 6:não
```

As opções de runlevel 2, 3, 4 e 5 devem estar habilitadas.

9. INICIALIZANDO A SOLUÇÃO

```
# /etc/init.d/citsmart-grp start
```

10. TESTE DE ACESSO

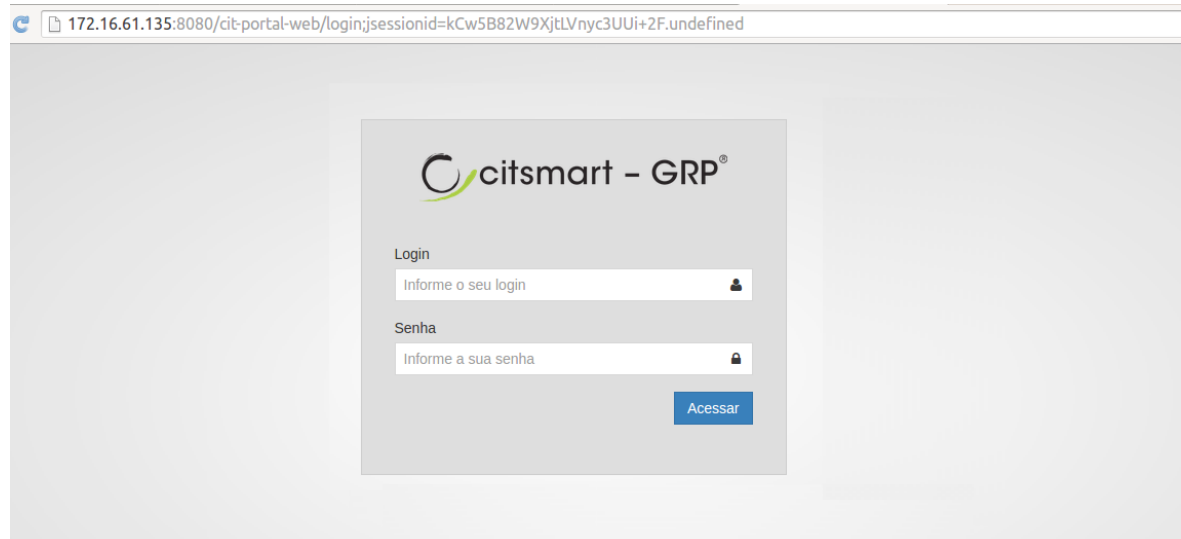
Após efetuar o start da daemon do citsmart-grp, aguardar a informação de deploy:

```
INFO [org.jboss.as.server] (DeploymentScanner-threads- 2) JBAS018559: Deployed "cit-portal-web.war"
```

```
INFO [org.jboss.as.server] (DeploymentScanner-threads- 2) JBAS018559: Deployed "citgrp-patrimonio-web.war"
```

Abra o browser e acesse: <http://HOST:8080/cit-portal-web>

Obs.: Alterar a palavra HOST para IP do servidor onde foi instalado o Jboss.



Dados de acesso:

Usuário: consultor

Senha: consultor