

# **Manual de Instalação de Ambiente Computacional para a Implantação do Sistema Citsmart-GRP**

## **Elaborado por:**

Marcus Sacramento(Spread)

Flávio Oliva(Spread)

Leonardo Borges(Capgemini)

Francisco Neto(+2x)

**Brasília, Abril de 2016**

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
<b>2</b>	<b>VISÃO GERAL .....</b>	<b>1</b>
<b>2.1</b>	<b>SOFTWARES BÁSICOS NECESSÁRIOS PARA A IMPLANTAÇÃO DA SOLUÇÃO.....</b>	<b>1</b>
<b>2.2</b>	<b>LOCALIZAÇÃO DOS ARQUIVOS NECESSÁRIOS .....</b>	<b>1</b>
<b>3</b>	<b>CONFIGURAÇÃO PARA INSTALAÇÃO DA SOLUÇÃO .....</b>	<b>2</b>
<b>3.1</b>	<b>CONFIGURAÇÃO DO LOADBALANCER .....</b>	<b>2</b>
3.1.1	<i>Configuração do SSL para o LoadBalancer(Requisito do CAS). .....</i>	<i>2</i>
<b>3.2</b>	<b>CONFIGURAÇÃO DO WILDFLY.....</b>	<b>3</b>
3.2.1	<i>Adicionando os Módulos para a aplicação.....</i>	<i>3</i>
3.2.2	<i>Adicionando um profile específico para a aplicação.....</i>	<i>4</i>
3.2.3	<i>Importar as chaves de SSL para o cacerts do servidor Wildfly. ....</i>	<i>7</i>
3.2.4	<i>Reiniciar o serviço do Wildfly.....</i>	<i>7</i>
<b>3.3</b>	<b>CONFIGURAÇÃO DO POSTGRES .....</b>	<b>7</b>
3.3.1	<i>Criar os bancos de dados para aplicação. ....</i>	<i>7</i>
<b>3.4</b>	<b>ALTERAÇÃO DO ARQUIVO CAS.WAR .....</b>	<b>7</b>
3.4.1	<i>Descompactar o arquivo cas.war .....</i>	<i>7</i>
3.4.2	<i>Alterar o arquivo cas.properties .....</i>	<i>7</i>
3.4.3	<i>Compactar o arquivo cas.war.....</i>	<i>8</i>
<b>4</b>	<b>ORDEM DE DEPLOYMENT.....</b>	<b>8</b>
<b>5</b>	<b>CONCLUSÃO E TESTE DE ACESSO .....</b>	<b>8</b>

## 1 Introdução

Este manual tem por objetivo orientar os responsáveis pela geração e manutenção do ambiente computacional necessário para a implantação do sistema citsmart-GRP

Para que este programa possa funcionar é necessário criar um ambiente computacional contendo o servidor de aplicações Wildfly. Este manual orienta a instalação e adequação necessárias dos *softwares* bem como do Sistema Citsmart-GRP.

## 2 Visão Geral

### 2.1 Softwares básicos necessários para a implantação da Solução

A seguir é apresentada uma descrição dos softwares básicos necessários para a implantação do citsmart-GRP. As linhas **marcadas em amarelo** nos itens de configurações deverão ser alteradas conforme o ambiente a ser utilizado.

**WildFly:** é o servidor *web*. A versão utilizada para instalação é a 8.2.1 Final, onde a equipe da Central IT realizou adequações em seu software Citsmart-GRP para ser executada nessa versão em modo domain.

**Postgres:** servidor de banco de dados. A versão utilizada é a 9.4.

**Central Authentication System (CAS):** Solução de *Single Sign-On*. A versão utilizada foi disponibilizada pela equipe da Central IT baseada na versão 3.5.2

Suite de aplicações Citsmart GRP: As versões utilizadas foram disponibilizadas pela equipe da Central IT onde se encontram em sua versão 2.0.0. Cada módulo da aplicação é um WAR que precisa ser executado através do Cit Smart Portal Web. Essa versão foi customizada pela Própria Central IT para que executasse no ambiente consolidado do MJ e portanto é única em relação as adequações utilizadas, podendo ser incompatível com outras versões do WildFly e/ou que não se encontrem com algum LoadBalancer:

- Cit Portal Web: Aplicação que suporta os demais módulos do GRP. Necessita do módulo CAS executando
- Cit Tabelas CorpWeb: Módulo para integralização dos demais módulos
- Cit GRP Patrimônio Web: Módulo de Patrimônio
- Cit Contratos Web: Módulo de Contratos
- Cit Almoxarifado Web: Módulo de Almoxarifado
- Cit Adm Materiais Web: Módulo para administração de materiais

### 2.2 Localização dos Arquivos Necessários

Os módulos da Aplicação se encontram no repositório Artfactory do ministério da Justiça:

```
<distributionManagement>
  <repository>
    <id>localhost</id>
    <name>localhost-releases</name>
    <url>http://cgtimaven.mj.gov.br/artifactory/cit-smart-grp</url>
  </repository>
  <snapshotRepository>
    <id>localhost</id>
```

```

    <name>localhost-snapshots</name>
    <url>http://cgtimaven.mj.gov.br/artifactory/cit-smart-grp</url>
  </snapshotRepository>
</distributionManagement>

```

### 3 Configuração para instalação da solução

#### 3.1 Configuração do LoadBalancer

##### 3.1.1 Configuração do SSL para o LoadBalancer(Requisito do CAS).

- Criar arquivo openssl.cnf com as seguintes configurações. Para ambientes de produção é necessário verificar o formato correto com a entidade certificadora para geração das chaves e certificados:

```

[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req

[req_distinguished_name]
countryName = Country Name (2 letter code)
countryName_default = BR
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = DF
localityName = Locality Name (eg, city)
localityName_default = Brasilia
organizationalUnitName = Organizational Unit Name (eg, section)
organizationalUnitName_default = CGTI
commonName = MJ
commonName_max = 64

[ v3_req ]
# Extensions to add to a certificate request
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = dsv1x1b01.mj.gov.br
DNS.2 = dsv1x1b01
DNS.3 = dsv1x1b02.mj.gov.br
DNS.4 = dsv1x1b02
DNS.5 = dsv1xwf01
DNS.6 = dsv1xwf01.mj.gov.br
DNS.7 = dsv.mj.gov.br
IP.1 = 10.10.4.20
IP.2 = 10.10.4.21
IP.3 = 10.10.4.22
IP.4 = 10.10.4.23

```

- Executar os comandos para geração de chaves e certificado auto assinado utilizando como base o padrão do arquivo openssl.cnf criado anteriormente:

```

openssl genrsa -out dsv.key 3650 -config openssl.cnf
openssl req -new -out dsv.csr -key dsv.key -config openssl.cnf
openssl req -text -noout -in dsv.csr
openssl x509 -req -days 3650 -in dsv.csr -signkey dsv.key -out dsv.crt -extensions v3_req -extfile openssl.cnf
openssl pkcs12 -export -in dsv.crt -inkey dsv.key -out dsv.p12

```

- No arquivo de configuração de SSL do Apache (ssl.conf) alterar as seguintes linhas para apontar os arquivos de certificado:

```
SSLCertificateFile /etc/pki/tls/certs/server.crt
SSLCertificateKeyFile /etc/pki/tls/certs/server.key
```

- Reiniciar e testar se o loadbalancer é acessado via protocolo https.

## 3.2 Configuração do Wildfly

### 3.2.1 Adicionando os Módulos para a aplicação

- Módulo Hibernate:
  - Copiar o jar *hibernate-jpa-2.1-api-1.0.0.Final.jar* para o diretório: *javax/persistence/api/main* dentro do diretório *modules* do WildFly
  - Criar o arquivo *module.xml* no diretório *javax/persistence/api/main*:

```
<?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>
```

- Módulo Javassist:
  - Copiar o jar *javassist-3.18.1-GA.jar* para o diretório: *org/javassist/main* dentro do diretório *modules* do WildFly
  - Criar o arquivo *module.xml* no diretório *org/javassist/main*:

```
<?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>
```

- Módulo CITGRP:
  - Criar o arquivo *module.xml* no diretório *br/com/citsmart-grp/main* dentro do diretório *modules* do WildFly:

```
<module xmlns="urn:jboss:module:1.1" name="br.com.citsmart-grp">
  <resources>
    <resource-root path="." />
  </resources>
</module>
```

- Criar o arquivo ***cit-app.properties*** no diretório *br/com/citsmart-grp/main* dentro do diretório *modules* do WildFly;

```

# 1 - Quando for autenticação via banco de dados direto o arquivo spring-
security
# 2 - Quando for autenticação via jasig-cas o arquivo spring-security-cas
metodo.autenticacao=2
metodo.autenticacao.ldap=false
cas.return.url=https://dsv.mj.gov.br/cit-portal-web/
#environment=prod
environment=dev
server.port=443

authentication.method=spring-security-cas.xml
authentication.url=https://dsv.mj.gov.br

# PROPRIEDADES DE PERSISTENCIA
hibernate.show.sql=false
hibernate.format.sql=true
hibernate.generate.ddl=true
hibernate.hbm2ddl.auto=update
hibernate.use_sql_comments=false
hibernate.generate_statistics=true
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
#hibernate.dialect=br.com.centralit.framework.util.SQLServerCustomDialect

# PROPRIEDADES DO ESI
-- local broker.url=tcp://localhost:61616
-- desenvolvimento broker.url=tcp://10.2.1.61:8161
-- produção broker.url=tcp://172.20.0.70:8161
broker.url=tcp://localhost:61616
esi.input.queue=esi.input.queue
esi.output.queue=esi.output.queue
esi.service.users=esi_service_users
esi.service.groups=esi_service_groups
esi.service.notification=esi_service_notification
esi.show.objects=false

# PROPRIEDADES DE EMAIL
mail.enabled=false
mail.smtp.quitwait=false
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
mail.smtp.socketFactory.fallback=false
mail.debug=false

mail.host=servermail
mail.username=contamail
mail.password=senhaemail
mail.smtp.auth=false
mail.smtp.starttls.enable=true
mail.port=25
mail.from=citsmart@centralit.com.br

```

### 3.2.2 Adicionando um profile específico para a aplicação

- Alterar o arquivo domain.xml em domain/configuration/ acrescentando um novo perfil com base no perfil full-ha:
  - Alterar o nome do profile para citsmart-grp

```
<profile name="citsmart-grp" />
```

- Acrescentar o path de single-sign-on no profile do citsmart-grp no

```
<subsystem xmlns="urn:jboss:domain:undertow:1.2">
```

```

<host name="default-host" alias="localhost">
  <location name="/" handler="welcome-content" />
  <filter-ref name="server-header" />
  <filter-ref name="x-powered-by-header" />
  <single-sign-on path="/" />
</host>

```

- Acrescentar os data-sources para a aplicação

```

<datasource jndi-name="java:/env/jdbc/cit-portal" pool-name="cit-portal">
  <connection-url>jdbc:postgresql://10.10.28.10:5432/citgrp</connection-url>
  <driver>postgresql</driver>
  <pool>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </security>
</datasource>
<datasource jndi-name="java:/env/jdbc/cit-almoxarifado" pool-name="cit-almoxarifado">
  <connection-url>jdbc:postgresql://10.10.28.10:5432/citgrp</connection-url>
  <driver>postgresql</driver>
  <pool>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </security>
</datasource>
<datasource jndi-name="java:/env/jdbc/cit-adm-materiais" pool-name="cit-adm-materiais">
  <connection-url>jdbc:postgresql://10.10.28.10:5432/citgrp</connection-url>
  <driver>postgresql</driver>
  <pool>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </security>
</datasource>
<datasource jndi-name="java:/env/jdbc/cit-contratos" pool-name="cit-contratos">
  <connection-url>jdbc:postgresql://10.10.28.10:5432/citgrp</connection-url>
  <driver>postgresql</driver>
  <pool>

```

```

    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </security>
</datasource>
<datasource jndi-name="java:/env/jdbc/citgrp-patrimonio" pool-name="citgrp-
patrimonio">
  <connection-url>jdbc:postgresql://10.10.28.10:5432/citgrp</connection-url>
  <driver>postgresql</driver>
  <pool>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </security>
</datasource>
<datasource jndi-name="java:/env/jdbc/cit-tabelas-corp" pool-name="cit-
tabelas-corp">
  <connection-url>jdbc:postgresql://10.10.28.10:5432/citgrp</connection-url>
  <driver>postgresql</driver>
  <pool>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </security>
</datasource>

```

- Criar o server Group para as aplicações do Citsmart GRP

```

<server-group name="citsmart-grp" profile="citsmart-grp">
  <jvm name="citsmart-grp">
    <heap size="1024m" max-size="5200m" />
    <permgen size="256m" max-size="752m" />
    <jvm-options>
      <option value="-XX:MaxPermSize=1024m -XX:+UseParallelGC -XX:+UseNUMA -
      XX:+HeapDumpOnOutOfMemoryError -XX:ParallelGCThreads=4" />
    </jvm-options>
  </jvm>
  <socket-binding-group ref="full-ha-sockets" />
</server-group>

```



### 3.2.3 Importar as chaves de SSL para o cacerts do servidor Wildfly para cada uma das máquinas no server group.

- Copiar os arquivos de chave e keystore para o diretório */domain/configuration* do servidor:
  - Executar o comando keytool para importação da chave:

```
keytool -import -noprompt -trustcacerts -alias GRP -file dsv.crt -keystore $JAVA_HOME/lib/security/cacerts
```

### 3.2.4 Reiniciar o serviço do Wildfly

## 3.3 Configuração do Postgres

### 3.3.1 Criar os bancos de dados para aplicação.

#### Necessário criar o usuário citgrp antes.

- CAS – Solução de Single sign-on.:

```
CREATE DATABASE cas
WITH OWNER = citgrp
ENCODING = 'UTF8'
LC_COLLATE = 'en_US.UTF-8'
LC_CTYPE = 'en_US.UTF-8'
CONNECTION LIMIT = -1;
```

- CITGRP – Banco de dados para a aplicação:

```
CREATE DATABASE citgrp
WITH OWNER = citgrp
ENCODING = 'UTF8'
LC_COLLATE = 'en_US.UTF-8'
LC_CTYPE = 'en_US.UTF-8'
CONNECTION LIMIT = -1;
```

## 3.4 Alteração do arquivo cas.war

É necessário alterar o arquivo cas.war para que sejam realizadas configurações do LDAP e banco da aplicação.

### 3.4.1 Descompactar o arquivo cas.war

### 3.4.2 Alterar o arquivo cas.properties

- Alterar as linhas de acesso ao servidor LDAP:

```
ldap.host.name=ldap://10.0.0.11:389
ldap.userDn=CN=adm.git,OU=Servicos,OU=ADM,DC=mj,DC=gov,DC=br
ldap.manager.password=mj@2015git
ldap.search.filter=%u@mj.gov.br
```

- Alterar as linhas de Gerenciamento do Web UI Security:

```
# Services Management Web UI Security
server.name=https://tst.mj.gov.br
```

- Alterar as linhas de do CIDR:

```
# IP address or CIDR subnet allowed to access the /status URI of
CAS that exposes health check information
```

```
cas.securityContext.status.allowedSubnet=127.0.0.1
```

- Alterar as linhas de acesso ao banco de dados do cas:

```
# == Basic database connection pool configuration ==
database.dialect=org.hibernate.dialect.PostgreSQLDialect
database.driverClass=org.postgresql.Driver
database.url=jdbc:postgresql://10.10.32.10:5432/cas
database.user=citgrp
database.password=citgrp
database.pool.minSize=6
database.pool.maxSize=18
```

### 3.4.3 Compactar o arquivo cas.war

- Compacte novamente para o arquivo cas.war:

## 4 Ordem de Deployment

A aplicação possui uma ordem específica de *deployment*. Isso garante que os serviços da aplicação se integrem entre os diversos módulos instalados.

A Correta ordem do deployment:

```
<deployment name="cas.war" runtime-name="cas.war" />
<deployment name="cit-portal-web.war" runtime-name="cit-portal-web.war" />
<deployment name="cit-tabelas-corp-web.war" runtime-name="cit-tabelas-corp-web.war" />
<deployment name="cit-adm-materiais-web.war" runtime-name="cit-adm-materiais-web.war" />
<deployment name="cit-almoxarifado-web.war" runtime-name="cit-almoxarifado-web.war" />
<deployment name="cit-contratos-web.war" runtime-name="cit-contratos-web.war" />
<deployment name="citgrp-patrimonio-web.war" runtime-name="citgrp-patrimonio-web.war" />
```

## 5 Conclusão e Teste de acesso

Abra o browser e acesse: <http://«url:porta»/cit-portal-web>

A Aplicação se encontra com o LDAP como modo de autenticação. Portanto o primeiro usuário logado se torna o administrador da solução.

Caso não esteja habilitado será necessário o login padrão:

- Usuário: consultor
- Senha: consultor