Documento de implantação

Conversão de multas Proponente e Autuado

Histórico de revisão

Data	Versão	Descrição	Autor
14/05/2018	1.0.0	Criação do documento	Paulo Borges
24/05/2018	1.1.0	Melhorias na publicação no jenkins	Paulo Borges
30/05/2018	1.2.0	Ajuste variáveis de produção	Paulo Borges

1. INTRODUÇÃO

Este documento tem como finalidade descrever, o ambiente necessário para manter o sistema IBAMA Conversão de multas(BackEnd e FrontEnd).

Considerando prover os ambientes de *front-end(WebApp) e back-end(WebServices)*. É necessário montar ambiente computacional específico para a aplicação Conversão de multas, a ser utilizado para o atendimento das demandas dessa natureza já abertas, bem como das futuras.

O ambiente descrito neste documento servirá com infraestrutura para outros sistemas que utilizam a mesma arquitetura, desta forma as atividades aqui relacionadas serão feitas apenas quando necessitar da criação de um ambiente novo, para implantação ou adição de um novo sistema deverá ser seguido outra documentação.

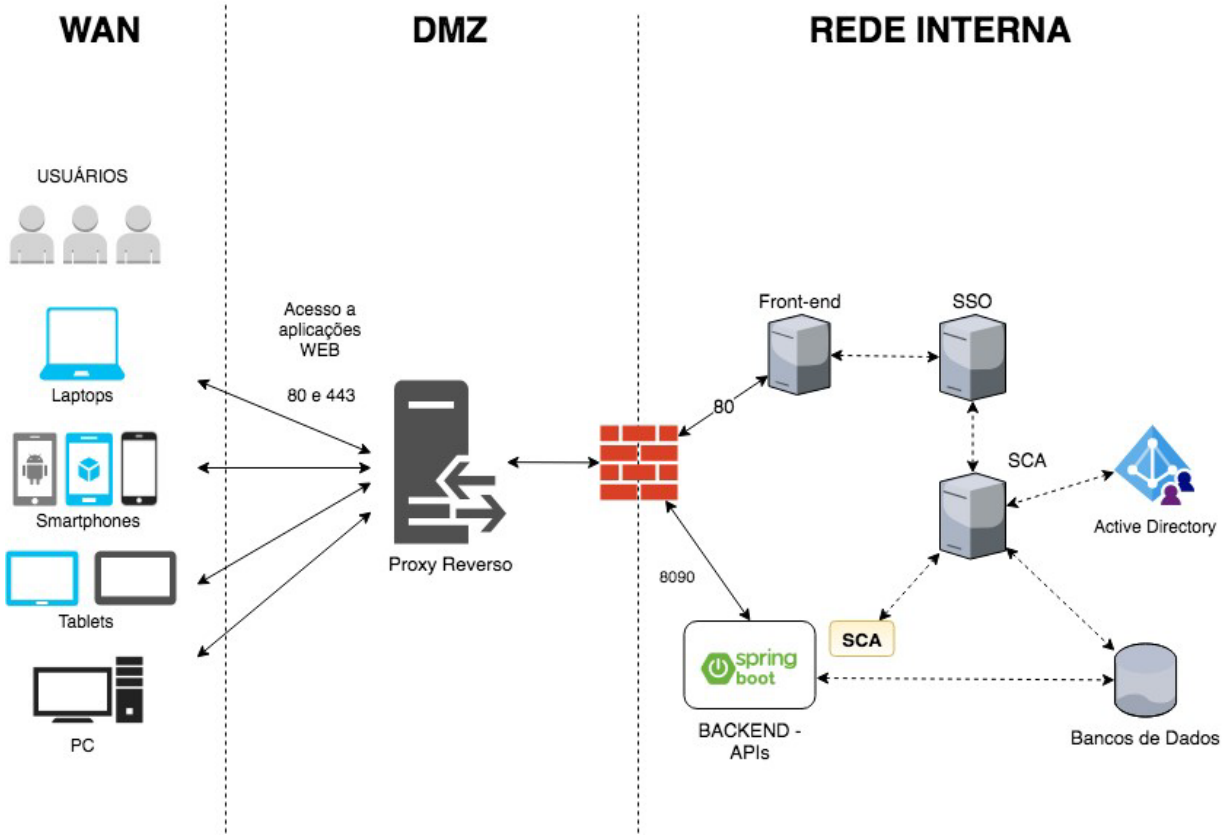

Figura 1 - Representação da Arquitetura de Aplicações WEB

2. OBJETIVOS ESPECÍFICOS

- Preparação das Máquinas Virtuais e Sistemas Operacionais;
- Preparação da Base de Dados;
- Instalação do Servidor SpringBoot (Back-end);
- Instalação do Servidor Apache (Front-end);
- Configuração job no jenkins Servidor Apache;
- Configuração job no jenkins Servidor SpringBoot

3. RESUMO DAS ATIVIDADES

- Preparação das Máquinas Virtuais e Sistemas Operacionais:
  - Instalação da Java 8 JDK;
  - Criação de usuário de serviço;
  - Mapeamento de unidade de Storage;
  - Criação de diretórios necessários;
  - Configuração de permissões;
  - Liberação de Portas Firewall para comunicação com servidores;
- Preparação da Base de Dados
  - Detalhes para criação e configuração da base de dados;
- Configuração job no jenkins Servidor SpringBoot
  - Criar pipeline para publicação
  - Rodar primeiro deploy
- Instalação do Servidor *SpringBoot* (Back-end):
  - Criação de atalho no init.d;
  - Configuração para início do serviço no boot do SO;
- Instalação do Servidor Apache (Front-end)
  - Instalação do Apache 2.4
  - Limpar diretório da aplicação;
  - Configuração de permissões;
- Configuração job no jenkins Servidor Apache
  - Criar pipeline para publicação

4. ATIVIDADES À SEREM EXECUTADAS

4.1. PRÉ-REQUISITOS

ITEM	DESCRIÇÃO	RECURSOS
1	MÁQUINA VIRTUAL - Sistema operacional LINUX baseado em CentOS 7.3 ou superior com JDK 1.8 Hotspot - (Backend);	20 GB disco 16 GB de Mémoria RAM 8 vCPUs
2	MÁQUINA VIRTUAL - Sistema operacional LINUX baseado em CentOS 7.3 ou superior com servidor APACHE 2.4 ou superior (Frontend);	20 GB disco 8 GB de Mémoria RAM 4 vCPUs

4.2. PREPARAÇÃO DAS MÁQUINAS VIRTUAIS E SISTEMAS OPERACIONAIS 4.2.1. CRIAÇÃO DE USUÁRIO DE SERVIÇO

- Atenção as configurações abaixo devem ser executadas o servidor back-end.
- Fazer o instalação da versão 8:
  - Conferir link do JDK em: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads2133151.html

```
#Instalação Java 8 JDK


$ cd ~
$ wget --no-cookies --no-check-certificate --header "Cookie:
gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie"
"http://download.oracle.com/otn-pub/java/jdk/8u172-
b11/a58eab1ec242421181065cdc37240b08/jdk-8u172-linux-x64.rpm"
$ sudo yum localinstall jdk-8u172-linux-x64.rpm
$ ln -s  /usr/java/jdk1.8.0_161/jre/bin/java /usr/bin/java $
rm ~/jdk-8u172-linux-x64.rpm
```

4.2.2. CRIAÇÃO DE USUÁRIO DE SERVIÇO

▪ Criação de usuário

```
## criar usuário
$ useradd -m apijava
## criar grupo
$ groupadd apijava
## adicionar usuário ao grupo
$ usermod -a -G apijava apijava
## alterar usuário do shell
$ chsh -s /bin/bash apijava
```

4.2.3. MAPEAMENTO DE UNIDADE DE STORAGE

▪ Criação de ponto de montagem para storage para armazenamento de arquivos, seguindo os padrões já existentes no ibama.

▪ O ponto de montagem deve apontar para “/opt/storage/”

4.2.4. CRIAÇÃO DE DIRETÓRIOS NECESSÁRIOS

▪ Criação de diretório

```
# criar diretório para aplicação
$  mkdir /opt/microservico
```

4.2.5. CONFIGURAÇÃO DE PERMISSÕES

▪ Criação de usuário

```
# permissões storage
## adicionar dono diretório
$ chown -R apijava:apijava /opt/storage
## adicionar permissões
$ chmod -R 775 /opt/storage

# permissões para binário da aplicação
## adicionar dono diretório
$ chown -R apijava:apijava /opt/microservico
## adicionar permissões
$ chmod -R 775 /opt/microservico

# adicionar usuário do jenkins ao grupo da api
# obs: substituir usuário do jenkins caso seja diferente do especificado
$ usermod -a -G apijava jenkins
```

4.2.6. LIBERAÇÃO DE PORTAS FIREWALL PARA COMUNICAÇÃO COM SERVIDORES E CONFIGURAÇÃO DO PROXY REVERSO

▪ Criação de regras de firewall.

- Substituir “servidor.java” pelo dns-name do servidor back-end.
- Substituir “servidor.banco” pelo dns-name do servidor do banco de dados.

ITEM	DESCRIÇÃO	ORIGEM	DESTINO
1	SERVIDOR DE APLICAÇÃO X BANCO DE DADOS	servidor.java	servidor.banco 1521 (TCP)
2	PROXY REVERSO X SERVIDOR DE APLICAÇÃO	servidor.java 443 (https)	servidor.java 8090 (http)
3	SERVIDOR DE APLICAÇÃO X GATEWAY	servidor.java	servidor.gateway 80 (TCP) 443 (TCP)

4.3. PREPARAÇÃO DA BASE DE DADOS

4.3.1. DETALHES PARA CRIAÇÃO E CONFIGURAÇÃO DA BASE DE DADOS

▪ Realizar execução dos seguintes scripts, em ordem (em anexo, junto a este documento):

- 01-create_schema.sql (alterar senha do usuário da aplicação nesse arquivo)
- Os demais scripts estarão em ordem numerica, exemplo “02-dml-xpto.sql”
- Por último deve-se adicionar os grants para o usuario de banco da aplicação (99-grants usuário aplicação.sql)

4.4. CONFIGURAÇÃO JOB NO JENKINS SERVIDOR SPRINGBOOT 4.4.1. CRIAR PIPELINE PARA PUBLICAÇÃO

▪ Abrir jenkins e criar novo job:

- Acesse: http://jenkins.ibama.gov.br
- Acesse a configuração do job
- Adicionar configurações como a imagem abaixo

Figura 2 - Exemplo “General configurations” para pipeline

- Marque “Este build é parametrizado” - Preencha conforme imagem abaixo.

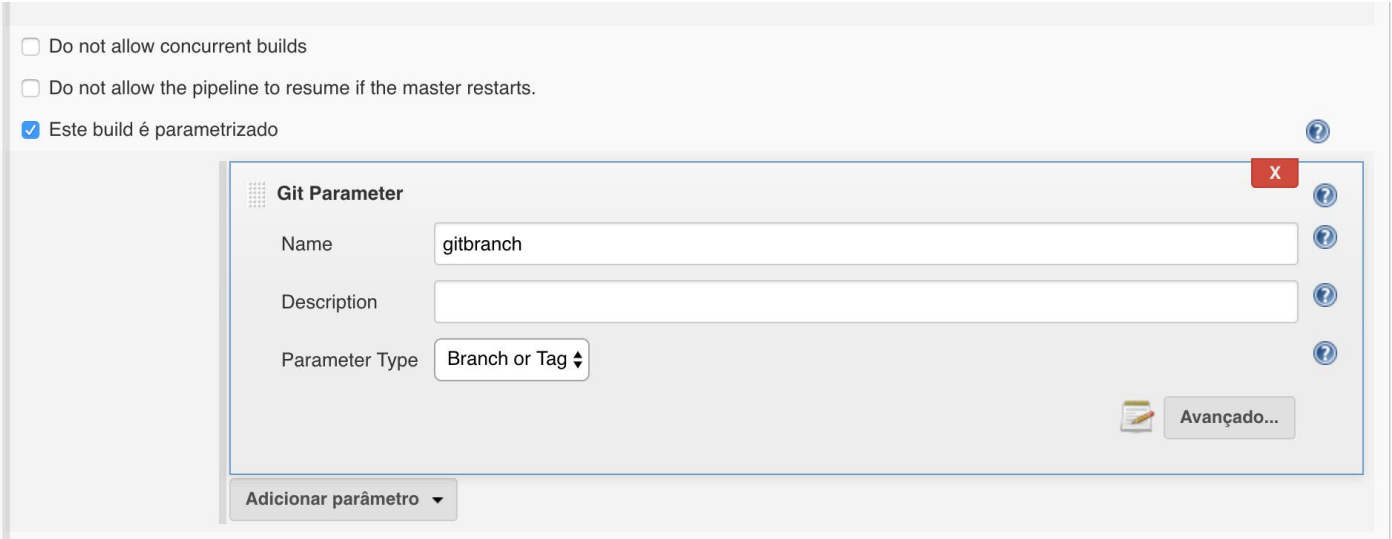

Figura 3 - Exemplo “Git Parameter” para pipeline

Figura 4 - Exemplo pipeline

- Adicione a seguinte pipeline
- substituir “servidor.java” pelo dnsname do servidor back-end.
- subsituir “servidor.banco” por dnsname do servidor de banco de dados.

- subsituir “user.banco” por usuário do servidor de banco de dados.
- subsituir “passwd.banco” por dnsname do servidor de banco de dados.
- subsituir “db.service” por service do servidor de banco de dados. - subsituir “db.host” por service do servidor de banco de dados.

```
#!groovy
gitRepositoryUrl='http://git.ibama.gov.br/java/conversao_multas_api.git'
gitBranch="${params.gitbranch.replaceAll('origin/','')}" node {    def
mvnHome
   stage('Checkout de Código fonte') {
      echo "[INFO] Deploy branch/tag: ${gitBranch}"
git([
               poll: true,
               credentialsId: 'bfcf0e3e-f373-47ae-b892-183812cfe7cc',
url: gitRepositoryUrl,                branch: gitBranch
        ])
      mvnHome = tool 'MAVEN_3.5.3'
   }
   stage('Build') {
        sh "echo 'server.port = 8090' > src/main/resources/application-prod.properties"
        sh "echo 'storage.path = /opt/storage/' >> src/main/resources/application-prod.properties"
sh       "echo       'spring.datasource.url=jdbc:oracle:thin:@db.host:1521:xe/db.service'       >>
src/main/resources/application-prod.properties"
        sh "echo 'spring.datasource.username=user.banco >>
src/main/resources/applicationprod.properties"
  sh               "echo               'spring.datasource.password=passwd.banco                >>
src/main/resources/applicationprod.properties"
        sh "echo 'spring.jpa.database-platform=org.hibernate.dialect.Oracle10gDialect' >>
src/main/resources/application-prod.properties"
```

```
        sh "echo 'logging.file=/var/log/cdmapi.log' >> src/main/resources/applicationprod.properties"
         sh "'${mvnHome}/bin/mvn' -Dmaven.test.failure.ignore clean install"
   }
   stage('Deploy Servidor Remoto') {
      sh "scp -P 3348 target/*.jar jenkins@servidor.java:/opt/microservico/cdm-api.jar"
   }
   stage('Reiniciando o servidor') {
      sh "ssh -p 3348 jenkins@servidor.java 'sudo -u apijava service cdm-api restart -
spring.profiles.active=prod'"
   }
}
```

- Salve o job.

4.4.2. RODAR PRIMEIRO DEPLOY

▪ Acesse o job criado anteriormente:

- Click em “Construir agora” (menu a esquerda)
- O primeiro deploy irá falhar(pois não existe o serviço para iniciar a aplicação), porém é necessário para copiar o binário para o serviço para futura criação do serviço e inicialização.

4.5. INSTALAÇÃO DO SERVIDOR *SPRINGBOOT* (BACK-END) 4.5.1. CRIAÇÃO DE ATALHO NO init.d

▪ Fazer o instalação da versão 8:

- Conferir instalação em: https://docs.spring.io/spring-boot/docs/current/reference/html/deploymentinstall.html

```
# criar um link simbolico no /init.d
$ ln -s /opt/microservico/cdm-api.jar /etc/init.d/cdm-api
```

4.5.2. CONFIGURAÇÃO PARA INÍCIO DO SERVIÇO NO BOOT DO S.O.

▪ Fazer o instalação da versão 8:

- Conferir instalação em: https://docs.spring.io/spring-boot/docs/current/reference/html/deploymentinstall.html

```
# iniciar serviço junto com o OS
$ chkconfig cdm-api on
```

4.6. INSTALAÇÃO DO SERVIDOR APACHE

4.6.1. INSTALAÇÃO DO APACHE 2.4

- Atenção as configurações abaixo devem ser executadas o servidor front-end.

```
# instalar apache
$ yum update
$ yum install httpd
#  iniciar serviço junto com o O.S.
$ systemctl enable httpd.service
$ systemctl restart httpd.service
```

4.6.2. LIMPAR DIRETÓRIO DA APLICAÇÃO

```
# limpar diretório para aplicação
$ rm -rf /var/www/html/*
```

4.6.3. CONFIGURAÇÃO DE PERMISSÕES

- Criação de usuário

```
# adicionar usuário do jenkins ao grupo da aplicação
# obs: substituir usuário do jenkins caso seja diferente do especificado
$ gpasswd -a jenkins apache
#adicionar permissão para pasta da aplicação
$ chmod 775 -R /var/www/html/
```

4.7. CONFIGURAÇÃO JOB NO JENKINS SERVIDOR APACHE

4.7.1. CRIAR PIPELINE PARA PUBLICAÇÃO

- Abrir jenkins e criar novo job:
  - Acesse: http://jenkins.ibama.gov.br
  - Acesse a configuração do job
  - Adicionar configurações como a imagem abaixo

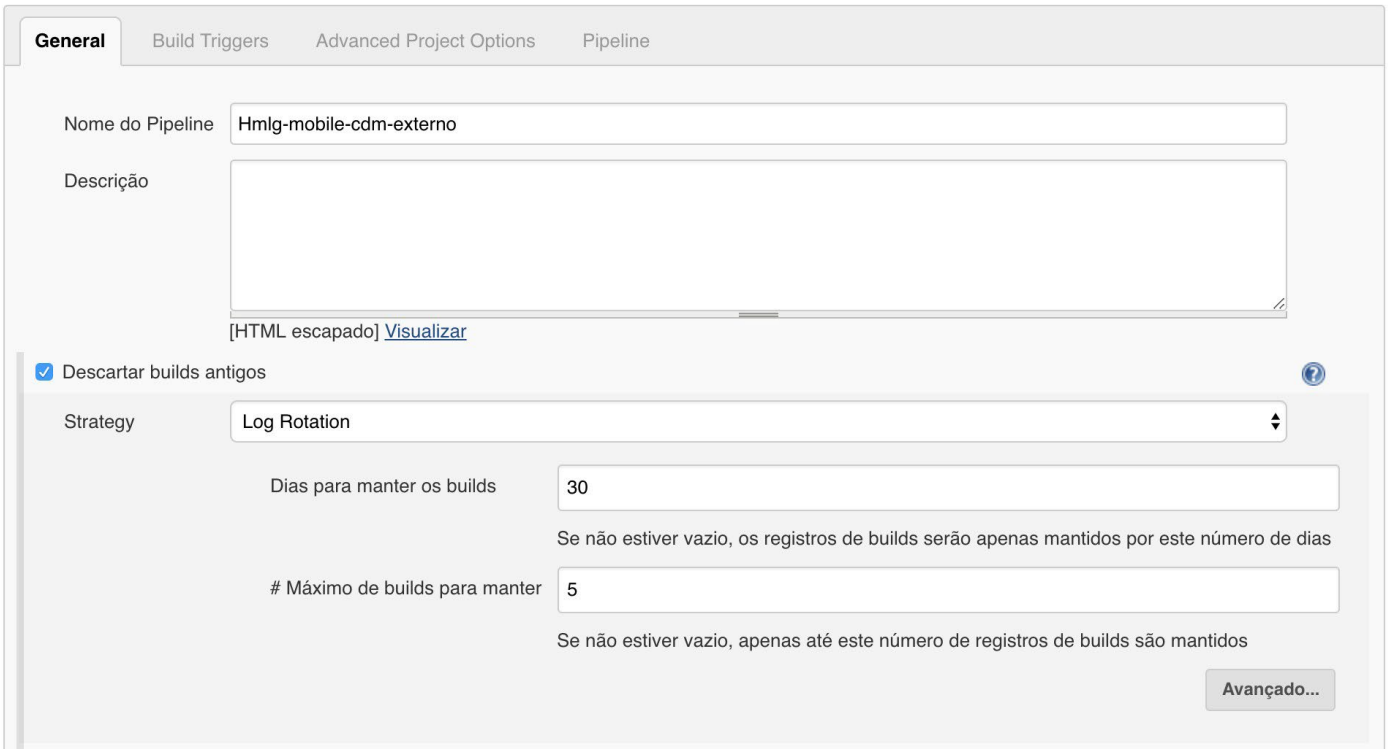

Figura 5 - Exemplo pipeline

- Marque “Este build é parametrizado”
- Preencha conforme imagem abaixo.

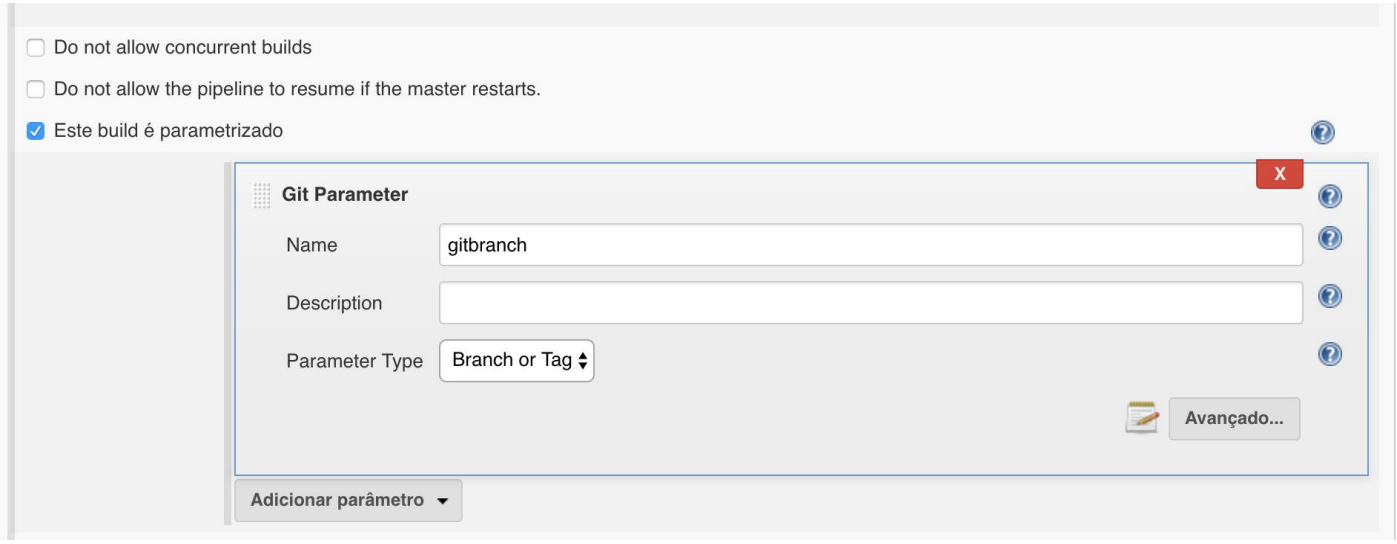

Figura 6 - Exemplo “Git Parameter” para pipeline

- Adicione a seguinte pipeline
- substituir “servidor.web” pelo hostname do servidor front-end.
- substituir “server-java.ibama.gov.br” pelo dnsname do servidor back-end.

```
#!groovy


gitRepositoryUrl='http://git.ibama.gov.br/Mobile/conversao_multas_externo.git'
gitBranch="${params.gitbranch.replaceAll('origin/','')}"


node {    def
nodeHome
  stage('Checkout de Código fonte') {
git([
            poll: true,
            credentialsId: 'bfcf0e3e-f373-47ae-b892-183812cfe7cc',
url: gitRepositoryUrl,
            branch: gitBranch
       ])
       nodeHome = tool 'NODEJS_6.14.2'
       env.PATH = "${nodeHome}/bin:${env.PATH}"
  }
  stage('Instalar/Atualizar  pacotes') {              sh  "npm
install"
       sh "echo 'NG2_APP_HTTP_BASE_URL=http://server-java.ibama.gov.br/' > .env.prod"
       sh "echo 'NG2_APP_ONESIGNAL_APP_ID=9c050a04-5268-43a9-aa31-2cf5818309dc' >> .env.prod"         sh "echo
'CORDOVA_APP_ID=br.gov.ibama.cdmproponente >> .env.prod"         sh "echo 'CORDOVA_IOS_VERSION=~4.5.4' >>
.env.prod"         sh "echo 'CORDOVA_ANDROID_VERSION=~6.4.0' >> .env.prod"         sh "echo
'APK_NAME=IBAMA_CDM' >> .env.prod"         sh "echo 'CORDOVA_APP_VERSION=1.0.0' >> .env.prod"         sh "echo
'APP_NAME=IBAMA CDM' >> .env.prod"
       sh "echo 'APP_DESCRIPTION=Aplicativo IBAMA Conversão de multas' >> .env.prod"          sh "echo
'APP_EMAIL=conversaodemultas@ibama.gov.br     '    >>    .env.prod"                                  sh    "echo
'NG2_APP_ANALYTICS_TRACK_ID=UA-119829322-1' >> .env.prod"
       sh "echo 'NG2_APP_GOOGLE_MAPS_ID=AIzaSyAhyseNI_uE3cg1M012C2eMoh9E-0dua1o' >> .env.prod"
  }
  stage('Build') {
       sh "npm run ionic:build -- --prod --aot --optimizejs --minifyjs --minifycss && touch www/cordova.js"
  }


  stage('Deploy Servidor Remoto') {
    sh "scp -r -P 3348 www/* jenkins@servidor.web:/var/www/html"
  } }
```

- Salve o job.

5. REFERÊNCIAS

Instalação JDR/JDK: https://www.digitalocean.com/community/tutorials/how-to-install-java-on-centos-and-fedora
Download JDK/JRE: http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
Configuração spring: https://docs.spring.io/spring-boot/docs/current/reference/html/deployment-install.html