

Rafael Vieira Dos Santos

## Tutorial de Instalação e Integração do Jenkins com GitHub

Jenkins é um automatizador de testes unitários que pode ser baixado em <http://jenkins-ci.org> conforme demonstrado na Figura 1:

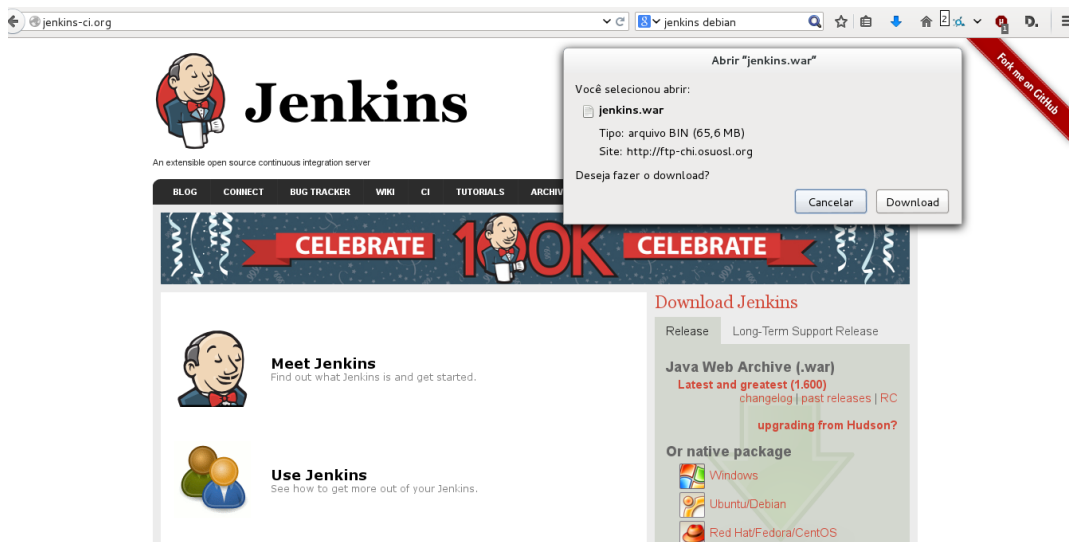


Figura 1: Página de Download do Jenkins

Após devemos colocar o arquivo jenkins.war na pasta webapps do tomcat:

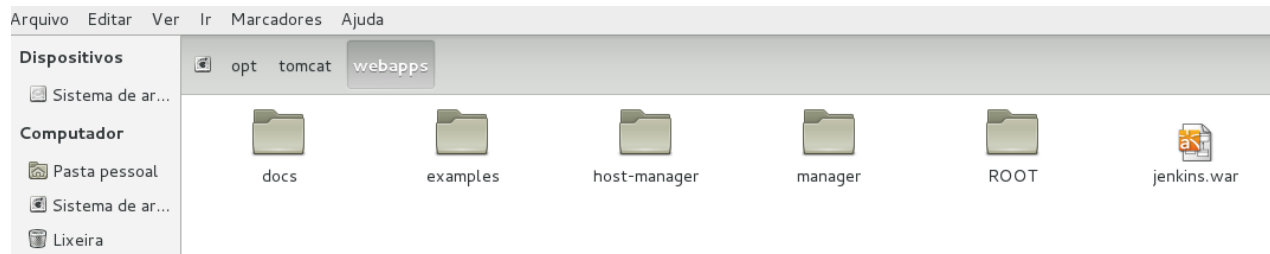


Figura 2: Pasta webapp do Tomcat

Após colocar o arquivo na pasta você deve iniciar o servidor do tomcat, para isto devemos executar o arquivo “startup.sh”:

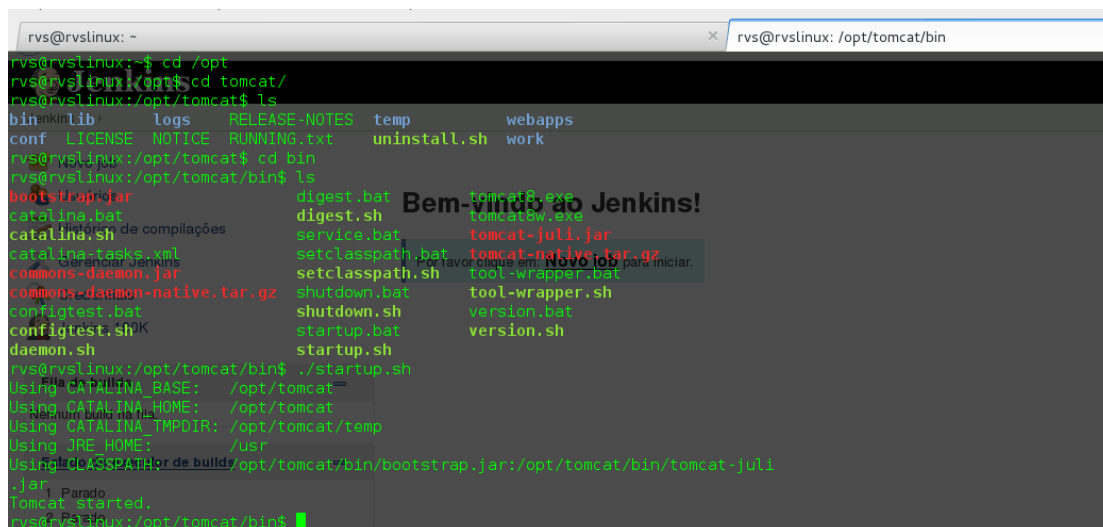


Figura 3: Script de Inicialização do Tomcat

Após o tomcat iniciado podemos acessar o Jenkins como demonstrado na Figura 4:

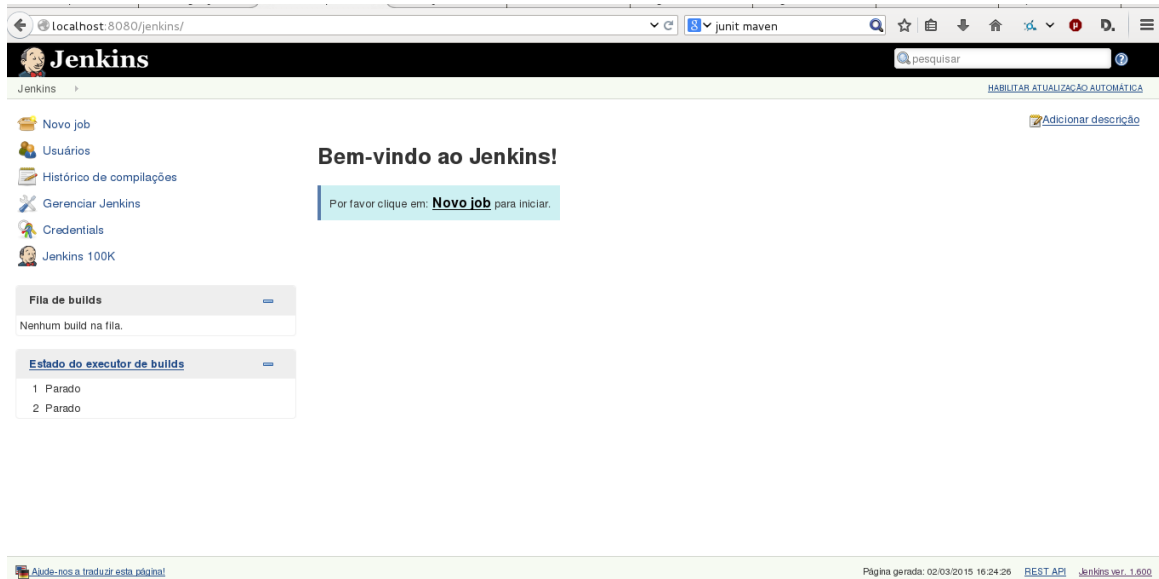


Figura 4: Pagina Inicial do Jenkins

Agora deve-se instalar o plugin GitHub no Menu Gerenciador de Plugins:

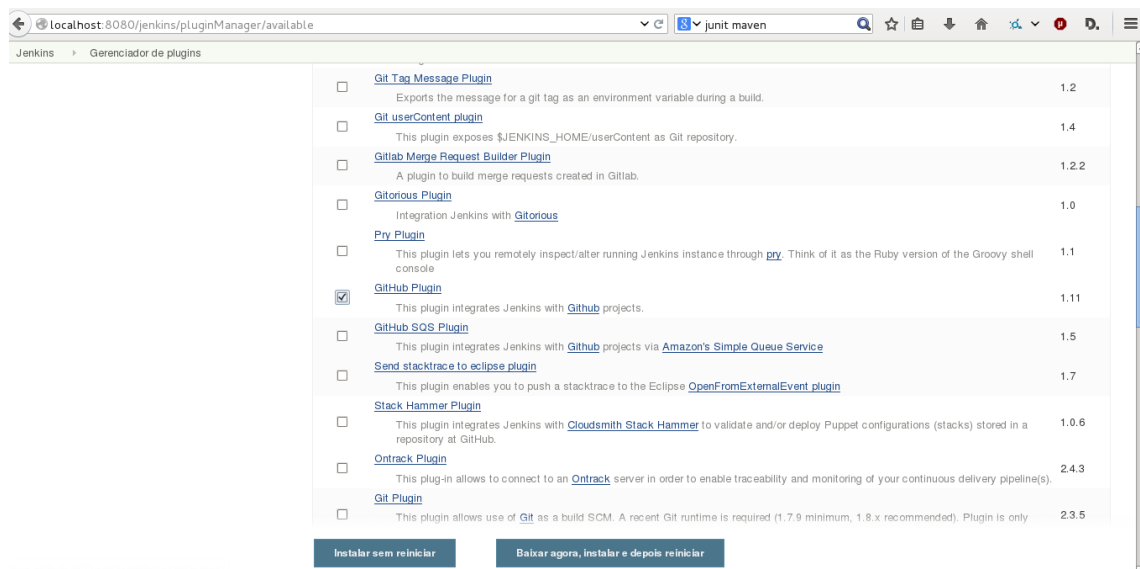


Figura 5: Plugin do GitHub

Em seguida opcionalmente pode ser habilitada a segurança no Jenkins como demonstrado na Figura 6:

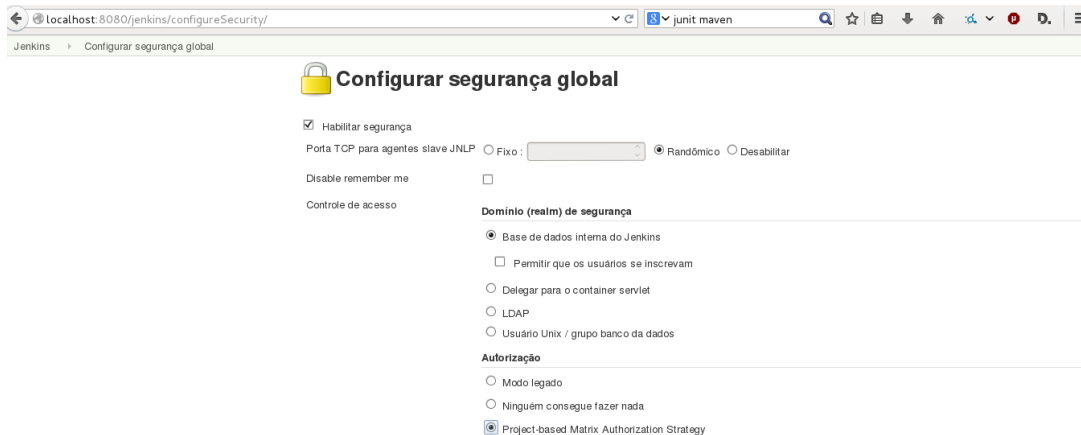


Figura 6: Segurança Jenkins

Outra forma de instalação que particularmente uso, para usuários Arch Linux somente é instalar via terminal com servidor próprio por meio do comando “pacman -S jenkins”:

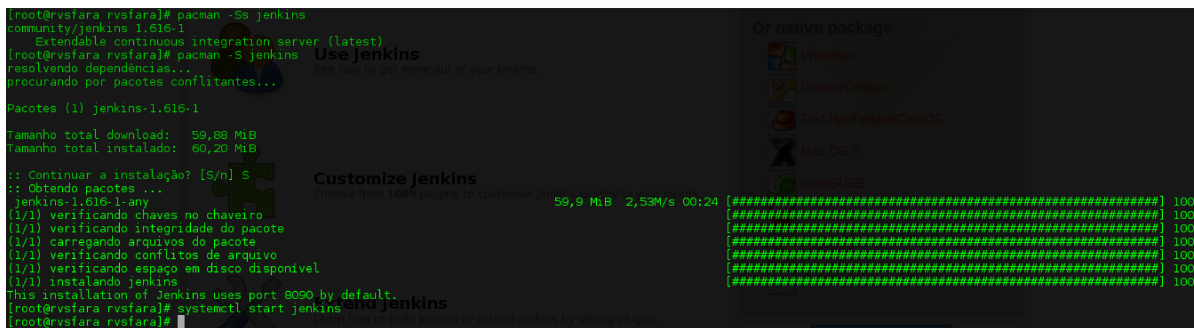


Figura 7: Instalação do Jenkins no Arch Linux

Por padrão ele executa na porta 8090, para iniciar somente o servidor do jenkins podemos utilizar o comando “systemctl start jenkins”, já para iniciar durante o boot da máquina utilizamos o comando “systemctl enable jenkins”, recomendo iniciar quando for utilizar somente.

Para atualizar o jenkins e ver a lista de plugins devemos utilizar o centro de atualizações:

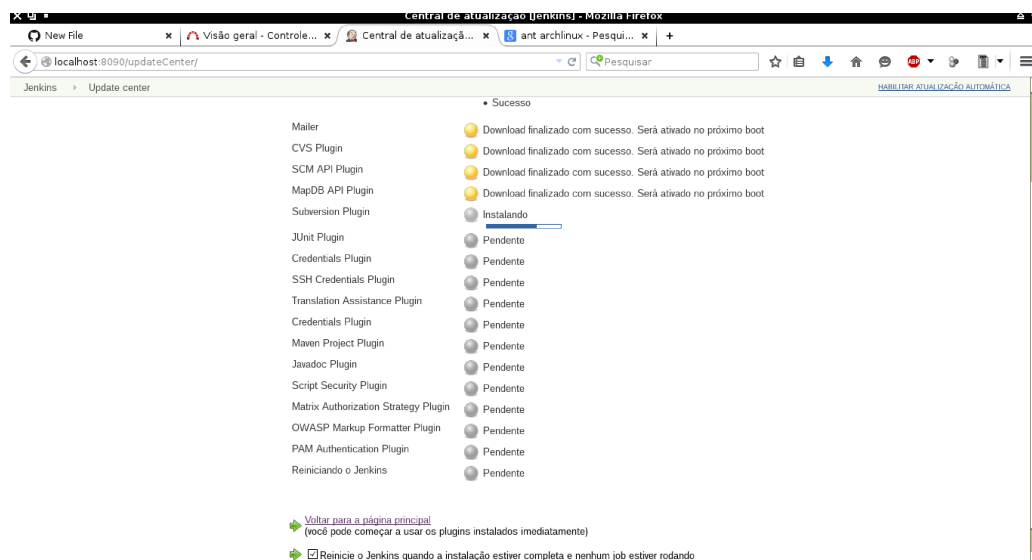


Figura 8: Jenkins Update Center

Em seguida já podemos instalar o plugin do GitHub:

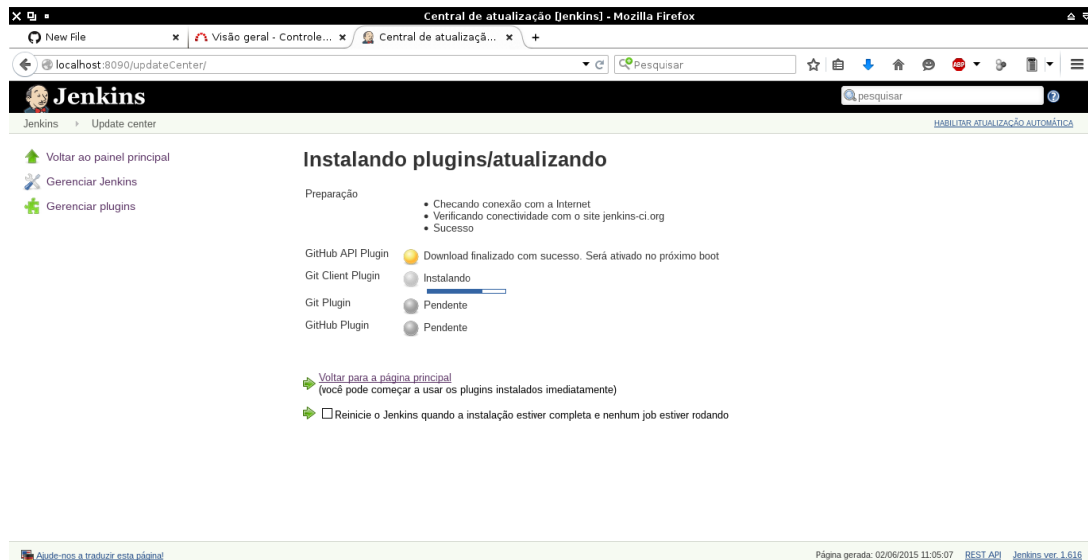


Figura 9: Plugin GitHub

Configuração do JDK no Jenkins:

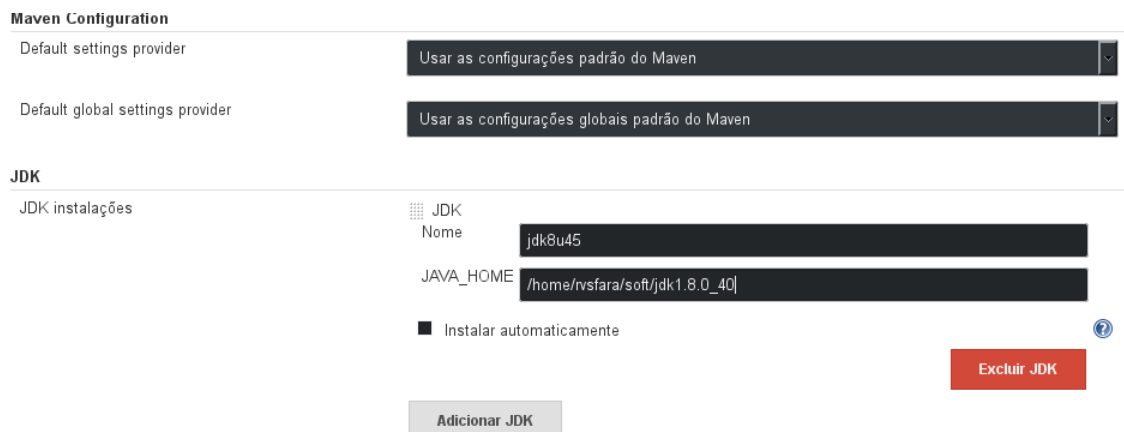


Figura 10: JDK

Configuração do Maven:

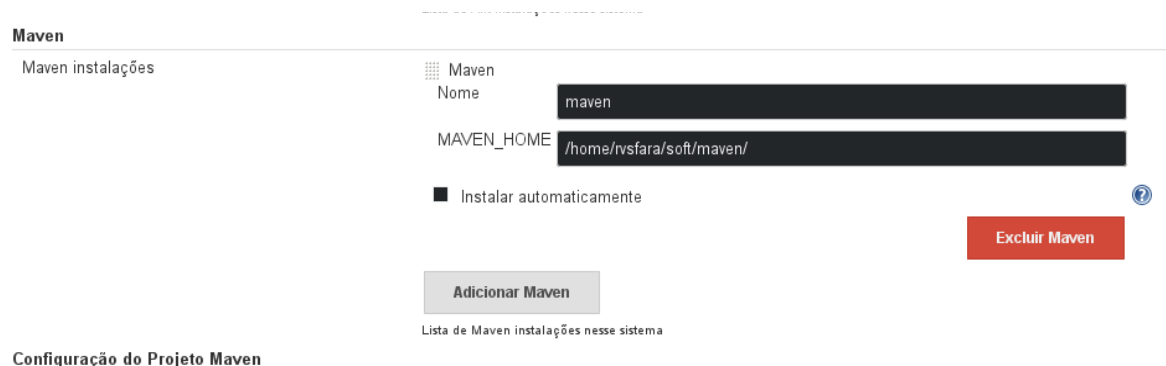


Figura 11: Maven

Para integrar o github com o seu projeto inicie um Projeto Maven:

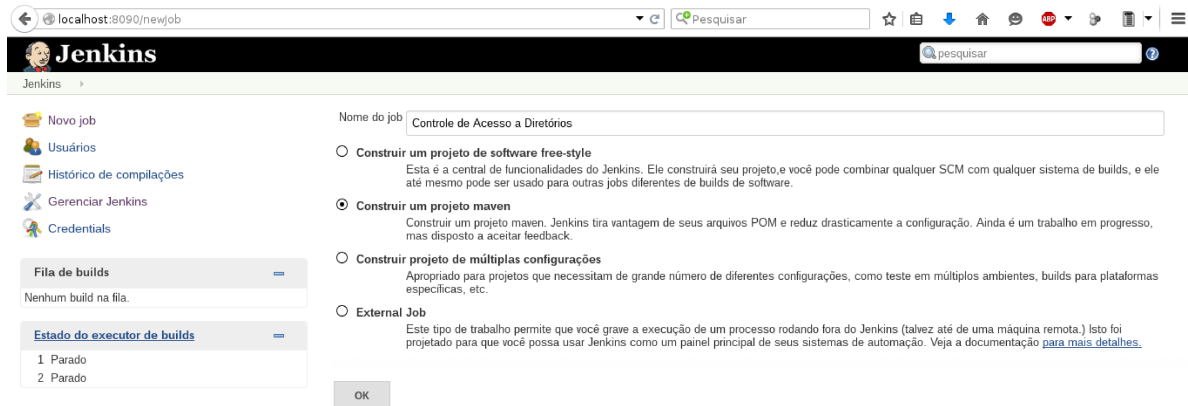


Figura 12: Construir um Projeto Maven

Em seguida colar a URL do GitHub do nosso projeto:

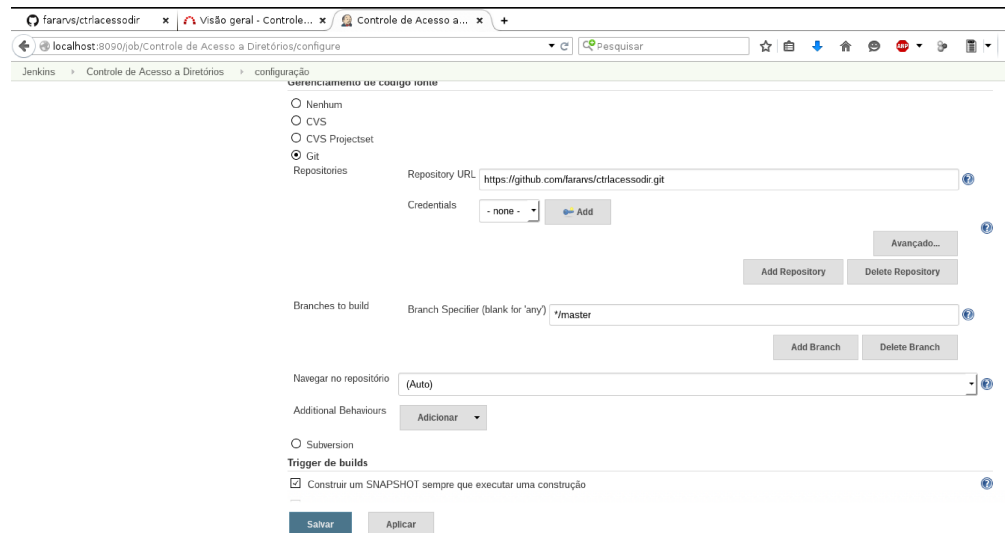


Figura 13: Link do GitHub

Usuario e Senha CVS é opcional caso desejar veja o pdf ssh:

The screenshot shows the Jenkins configuration interface. At the top, there's a field for 'Endereço de email do administrador do sistema' with the value 'rvsfara@gmail.com'. Below this is the 'SSH Server' section with 'SSH Port' set to 'Fixo' and a dropdown menu. The 'Git plugin' section includes 'Global Config user.name Value' (rvsfara), 'Global Config user.email Value' (rvsfara@gmail.com), and a checkbox for 'Create new accounts base on author/commmitter's email'. The 'CVS' section has 'Default Compression Level' set to '3 (Recommended)', 'Private Key Location' (/home/rvsfara/.ssh/id\_rsa), 'Private Key Password' (masked with dots), 'Known Hosts Location' (/home/rvsfara/.ssh/known\_hosts), and an 'Adicionar' button. The 'Subversion' section shows 'Subversion Workspace Version' set to '1.4' and an empty 'Exclusion revprop name' field.

Figura 14: Usuario e Senha

Selecionar a opção para construir sempre que houver uma alteração de código no GitHub:

The screenshot shows the Jenkins configuration page for a job named 'Controle de Acesso a Diretórios'. The 'configuração' tab is active. Under 'Build Triggers', the checkbox 'Build when a change is pushed to GitHub' is checked. Below this is the 'Pre Steps' section with 'Invocar Ant' set to 'Padrão' and 'Alvos' containing 'teste.jar'. At the bottom, there's a 'Construir' section with 'POM Raiz' set to 'pom.xml'. Buttons for 'Salvar', 'Aplicar', 'Avançado...', and 'Excluir' are visible.

Figura 15: Construir quando houver alteração

E opcionalmente podemos arquivar o Jar gerado pela construção:

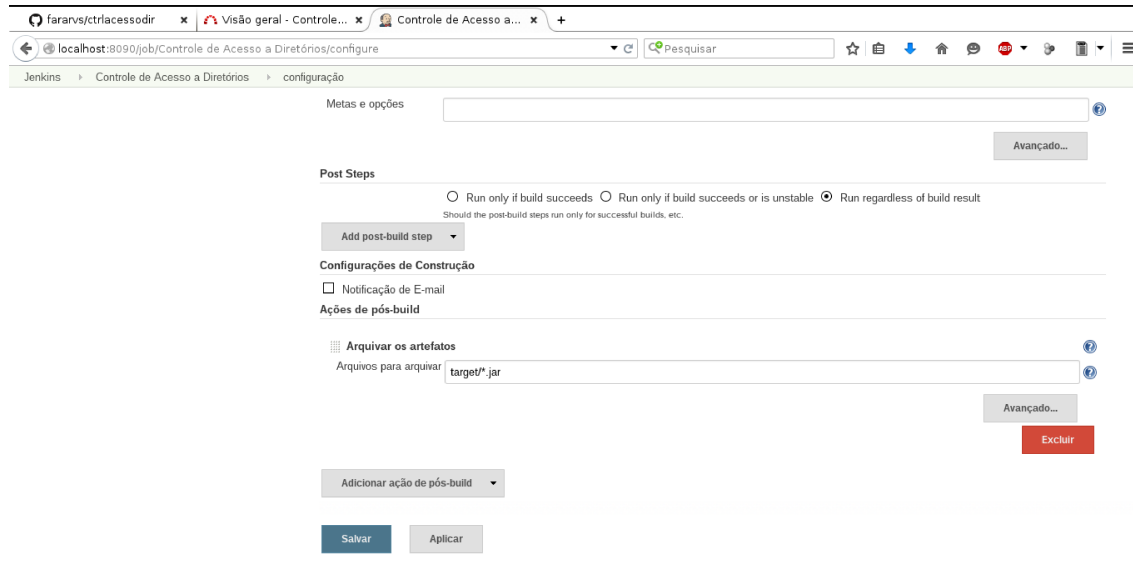


Figura 16: Arquivar jar após construção

Agora já temos nosso projeto configurado:

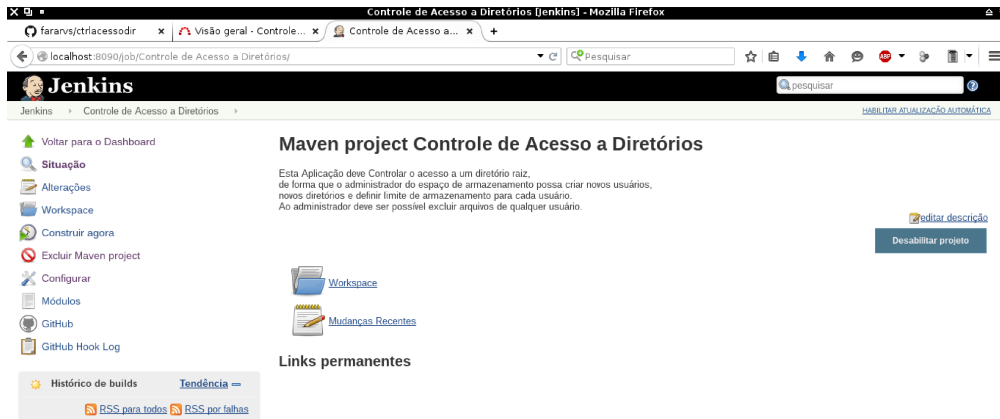


Figura 17: Projeto Configurado

## Construção do Projeto:

The screenshot shows the Jenkins web interface at localhost:8080/jenkins/. The left sidebar contains links for 'Novo job', 'Usuários', 'Histórico de compilações', 'Gerenciar Jenkins', and 'Credenciais'. The main area displays a table of jobs. The 'CtrlAcessoDir' job is highlighted, showing its status as 'N/D' (Not Done) for both 'Último sucesso' and 'Última falha'. Below the table, there is a section for 'Filas de builds' (Build Queues) and 'Estado do executor de builds' (Build Executor Status). The 'Estado do executor de builds' section shows two build executors: 'CtrlAcessoDir' and 'Parado'. The bottom of the page includes a footer with a link to 'Ajude-nos a traduzir esta página!' and a page generation timestamp of 14/05/2015 15:58:26.

S	W	Nome ↓	Último sucesso	Última falha	Última duração
☐	☐	CtrlAcessoDir	N/D	N/D	N/D

Ícone: S M L

Legenda: RSS de tudo RSS das falhas RSS apenas para os últimos builds

Página gerada: 14/05/2015 15:58:26 REST API Jenkins ver 1.612

Figura 18: Construção

## Saída do Console:

The screenshot shows the Jenkins console output for the 'CtrlAcessoDir' job. The console title is 'Saída do console'. The output text is as follows:

```
Iniciado pelo usuário anonymous
Construindo no workspace /home/rvsvfara/.jenkins/jobs/CtrlAcessoDir/workspace
Cloning the remote Git repository
Cloning repository https://github.com/fararvs/ctrlacessodir.git
> git init /home/rvsvfara/.jenkins/jobs/CtrlAcessoDir/workspace # timeout=10
Fetching upstream changes from https://github.com/fararvs/ctrlacessodir.git
> git --version # timeout=10
Setting http proxy: 200.134.81.53:3128
> git -c core.askpass=true fetch --tags --progress https://github.com/fararvs/ctrlacessodir.git +refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url https://github.com/fararvs/ctrlacessodir.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/fararvs/ctrlacessodir.git # timeout=10
Fetching upstream changes from https://github.com/fararvs/ctrlacessodir.git
Setting http proxy: 200.134.81.53:3128
> git -c core.askpass=true fetch --tags --progress https://github.com/fararvs/ctrlacessodir.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f6b863566312a903bfaa656061a6dff07f3a2f78 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f f6b863566312a903bfaa656061a6dff07f3a2f78
First time build. Skipping changelog.
Parsing POMs
Descoberto um novo módulo com.rvsvfara:ControleAcessoDiretorios ControleAcessoDiretorios
Modules changed, recalculating dependency graph
[workspace] $ /home/rvsvfara/soft/jdk1.8.0_40/bin/java -cp /home/rvsvfara/.jenkins/plugins/maven-plugin/WEB-INF/lib/maven3-agent-1.5.jar:/home/rvsvfara/soft/maven/boot/plexus-classworlds-2.5.2.jar:/home/rvsvfara/soft/maven/conf/logging-jenkins-maven3-agent.Maven3Main /home/rvsvfara/soft/maven/opt/tomcat/webapps/jenkins/WEB-INF/lib/remoting-2.51.jar /home/rvsvfara/.jenkins/plugins/maven-plugin/WEB-INF/lib/maven3-interceptor-commons-1.5.jar /home/rvsvfara/.jenkins/plugins/maven-plugin/WEB-INF/lib/maven3-interceptor-1.5.jar 46169
====[JENKINS REMOTING CAPACITY]====channel started
Executing Maven: -B -f /home/rvsvfara/.jenkins/jobs/CtrlAcessoDir/workspace/pom.xml install
[INFO] Scanning for projects...
```

Figura 19: Saída do Console