

# How-To :: Desenvolvimento da Corisco 3

Este documento se propõe a instruir o público interessado ao desenvolvimento da Corisco 3, uma solução formada principalmente pelo **DSpace 4.x** (entre outros componentes da solução). Originalmente, o objetivo para o desenvolvimento da plataforma estava destinado a atender ao projeto Brasileira - USP.

Ou seja, algumas particularidades frente aos objetivos iniciais deste projeto devem ser consideradas.

Este documento deve auxiliar na instalação dos vários componentes da Corisco 3 e também as ferramentas utilizadas (principalmente a ferramenta de desenvolvimento do NetBeans). A seguir a lista resumida:

- Instalação do **PostgreSQL**, base de dados adotada para as informações;
- Instalação do **pgAdmin**, ferramenta para acessar a base de dados do dspace, caso haja necessidade;
- Instalação do **NetBeans** com o apache tomcat embutido, ferramenta de desenvolvimento **java** mais o **container** java para que a aplicação rode localmente;

E além de instruir como iniciar o trabalho com esses componentes, há uma seção neste documento onde eu, autor e *alterador*, fiz uma customização nos arquivos do dspace 4 para que o procedimento de deploy remoto e parcial pudesse ser feito de forma padronizada. Porque o objetivo de se utilizar o DSpace é instalar a aplicação em um servidor.

**Reforçando:** o objetivo de se trabalhar com plataforma de desenvolvimento é para que todos trabalhem padronizados e organizados, de forma única.

## Índice

[Introdução](#)

[Instalação do PostgreSQL](#)

[Instalação do pgAdmin](#)

[Criar usuário e base de dados para o DSpace](#)

[Instalação do NetBeans](#)

[Download do código-fonte do repositório Git](#)

[Criação do diretório de instalação do DSPACE](#)

[Compilar o projeto principal e Instalar o DSpace](#)

[Criação do usuário administrador do DSpace](#)

[Rodar a aplicação web do DSpace \(Apache Cocoon\)](#)

[Rodar a aplicação web do DSpace \(Java Server Pages\)](#)

[Instalação do Drupal com PostgreSQL \(Mac OS X\)](#)

[Como proceder com o deploy remoto do DSpace 4](#)

## 1. Introdução

Este documento explica como preparar as ferramentas necessárias ao desenvolvimento da plataforma **Corisco 3**, desde as configurações iniciais até a instalação das ferramentas de desenvolvimento e o código-fonte utilizado no projeto.

Na plataforma, o [DSPACE](#) é o software responsável pela administração dos acervos digitalizados e por disponibilizar a biblioteca digital, utilizando o [PostgreSQL](#) como base de dados. A ferramenta para programação e compilação do código-fonte será o [NetBeans](#) (por suportar nativamente projetos maven e oferecer uma opção de download com o Apache Tomcat integrado).

O Apache Tomcat será o container das aplicações do DSPACE, tanto a versão feita em Apache Cocoon (/xmlui) quanto a versão Java Server Pages (/jspui).

## 2. Instalação do PostgreSQL

Acesse o endereço <http://www.postgresql.org/download/> e escolha a opção conforme o sistema operacional que você esteja utilizando. Para computadores com sistemas operacionais **windows** e **osx**, existe uma opção de download do PostgreSQL contendo também o pgAdmin. Se você fizer o download da opção com a ferramenta pgAdmin, não há necessidade de seguir os passos da etapa 2a.

O instalador do PostgreSQL pedirá uma senha padrão para o usuário **postgres**. Guarde-a em um local seguro, pois precisaremos dessa senha para conectar ao servidor local com o pgAdmin e para criar a base de dados e o usuário para a aplicação do dspace.

Para distribuições **linux**, você instala o sistema gerenciador de banco de dados e precisará instalar o pgAdmin para facilitar as etapas de criação do usuário dspace e da base de dados dspace (utilizada por padrão pelo DSPACE). Veja a seção abaixo:

### a. Instalação do pgAdmin

Caso você tenha instalado somente o PostgreSQL, acesse o endereço <http://www.pgadmin.org/download/index.php> e siga as instruções para instalação no seu linux, windows ou mac osx.

Conforme o site, a maioria das distribuições linux oferecem pacotes de instalação em seus respectivos repositórios, ou seja, acessando o *software installer* desses sistemas operacionais linux-based você consegue instalar o pgAdmin tranquilamente.

### b. Criar usuário e base de dados para o DSpace

Abra o **pgAdmin** para conectar ao servidor local. A instancia do servidor devera aparecer sob a seção **Server Groups > Servers (1)**. Clique com o botão direito para conectar e informe a senha determinada para o usuário **postgres**.

Em **Login Roles**, clique com o botão direito para criar um novo login. Utilize os seguintes dados: **dspace** como **Role Name**, **dspace** como **Password** e também na confirmação da senha.

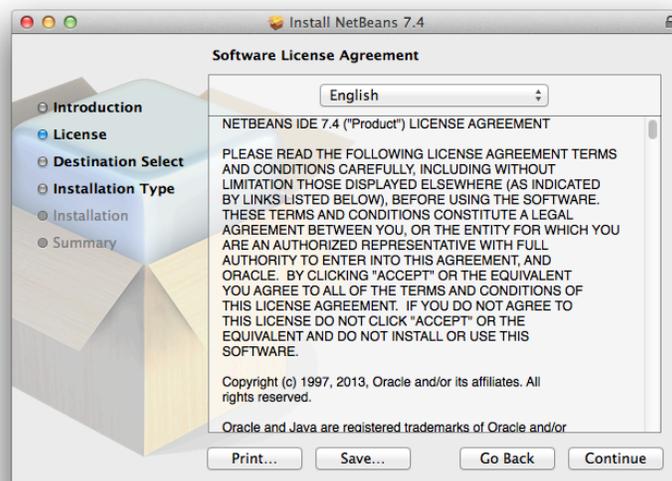
Em **Databases**, clique com o botão direito para criar uma nova base de dados. Utilize os seguintes dados: **dspace** como nome da base de dados e, como **owner**, o usuário **dspace** criado anteriormente.

### 3. Instalação do NetBeans

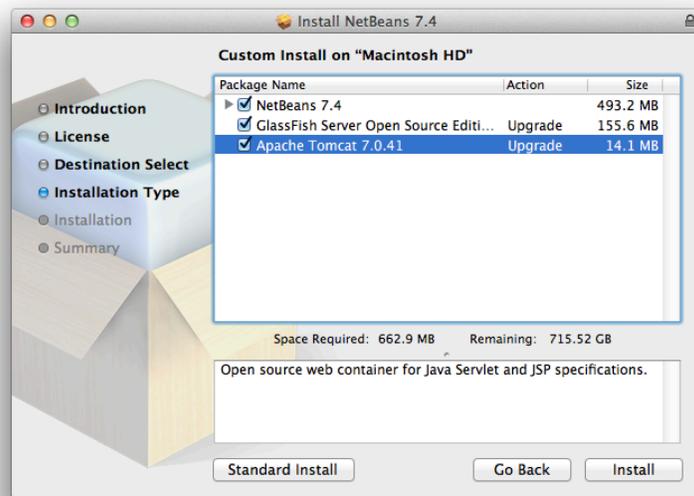
Acesse o endereço <https://netbeans.org/downloads/> e selecione uma opção de download desta ferramenta que já venha com o Apache Tomcat (para facilitar a execução do DSPACE diretamente pela ferramenta).

Execute o arquivo de instalação após o término do download. Para distribuições **Linux**, você precisa alterar os atributos deste arquivo, executando o comando **chmod a+x <arquivo>** para torná-lo executável. É altamente recomendável que você comece a instalação do NetBeans executando o instalador como **root**.

Para usuários **osx**, você **precisa customizar** a instalação e especificar que você realmente quer instalar o Apache Tomcat (por padrão, esse *container* não é instalado mesmo que esteja no arquivo de instalação).



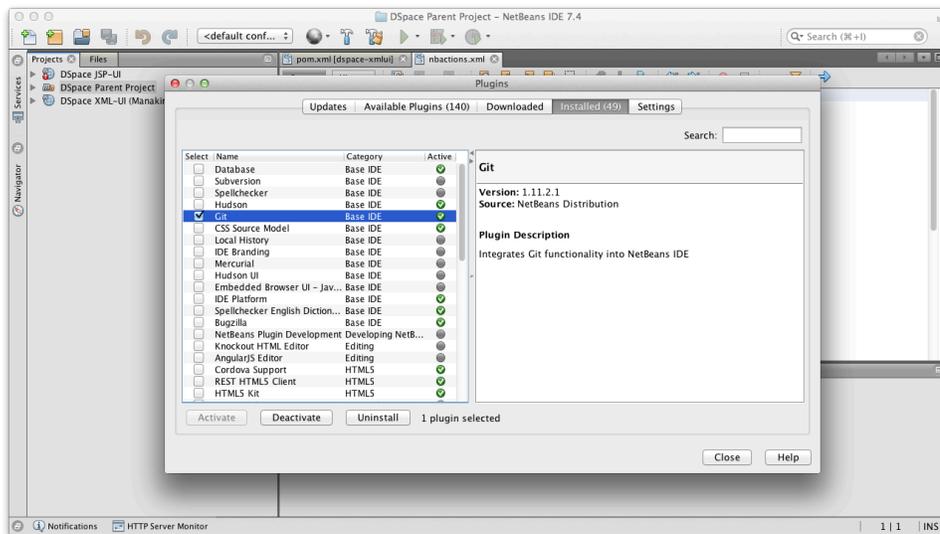
Logo após aceitar a licença de uso, há um botão chamado *customize*. Clicando nele, uma lista de opções será apresentada, conforme ilustra a próxima imagem:



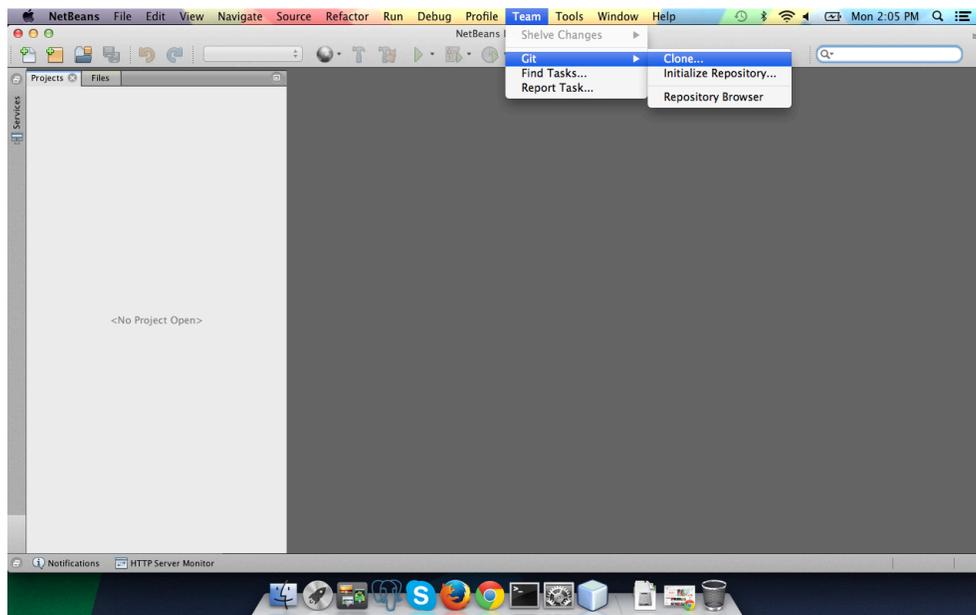
Marque a opção Apache Tomcat para instalar o container integrado ao NetBeans. A partir de agora, o processo de instalação é bem direto, você pode prosseguir sem se preocupar com demais opções e detalhes.

## 4. Download do código-fonte do repositório Git

Antes de baixar o código-fonte do projeto pelo Git, iremos conferir se o plugin do Git está propriamente ativado no NetBeans. Abra o **NetBeans**, menu **Tools > Plugins** e confira, conforme a imagem abaixo.

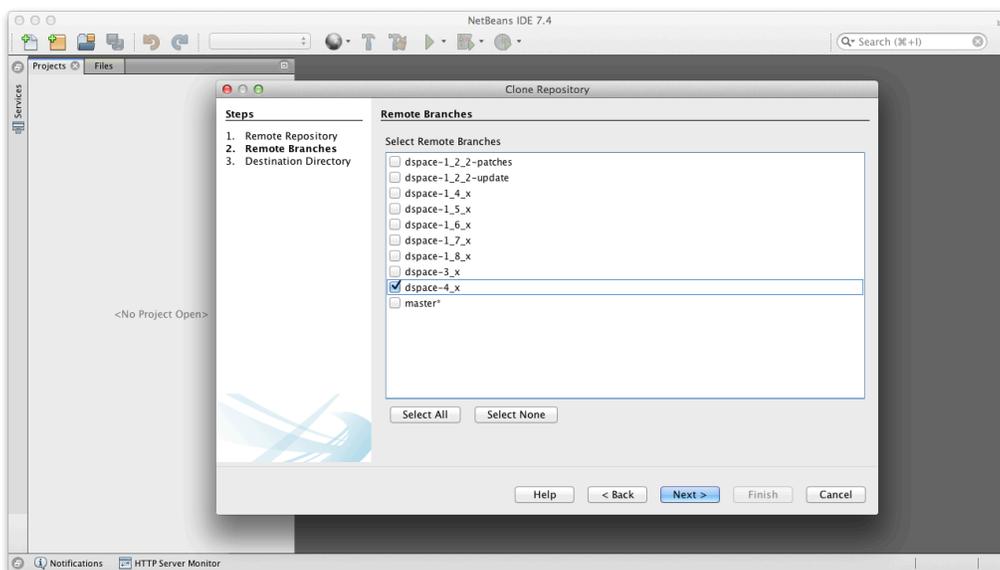


Feche as janelas e vá para o menu **Team > Git > Clone**. A imagem abaixo ilustra essa etapa:

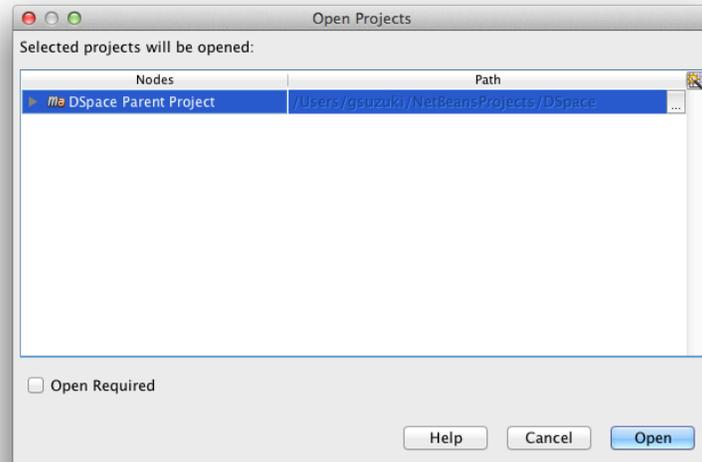


Utilize o endereço <https://github.com/esgomi/DSpace.git> na URL do repositório e informe o seu usuário e senha do GitHub. Caso você não consiga prosseguir nessa etapa, você deve solicitar acesso ao repositório do endereço acima. Atenção, não confunda o repositório Git do projeto com o repositório oficial do DSPACE (um fork entre os repositórios já foi feito)!

Na próxima etapa, deverá aparecer uma lista contendo todas as branches do projeto. Selecione a branch chamada **dspace-4\_x**.



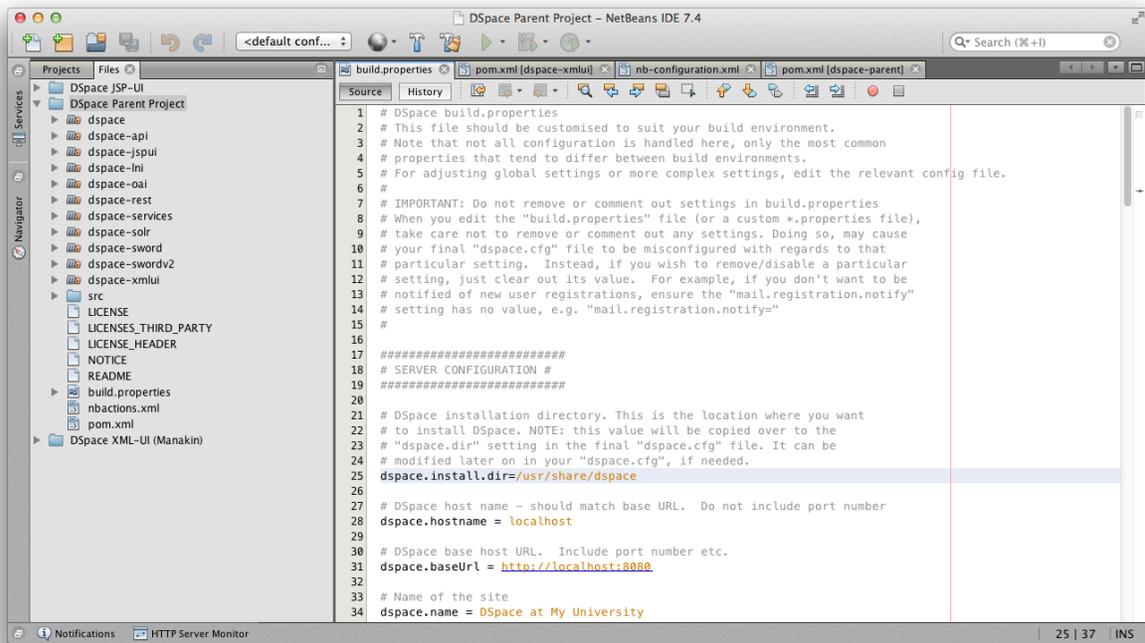
Clicar *Next* para configurar o diretório local deste repositório DSpace. Ao terminar, o NetBeans vai questionar se você deseja abrir o projeto. Ao aceitar, uma janela sera apresentada, conforme a imagem abaixo:



Na primeira vez que você abrir o projeto, o NetBeans avisará sobre um problema com arquivos do projeto. Uma sugestão para resolver esse problema vai ser sugerida e também as bibliotecas necessárias para a aplicação serão baixadas. Essa etapa pode demorar alguns minutos (não se preocupe).

## 5. Criação do diretório de instalação do DSPACE

Ainda com o NetBeans aberto, acesse a visão de arquivos do projeto **DSpace Parent Project** e um arquivo chamado **build.properties**. Dentro dele, uma variável **dspace.install.dir** apontando para o diretório **/usr/share/dspace**. Confira na imagem abaixo:



Esse diretório deverá ser criado para que a aplicação compilada do dspace funcione. E mais: as permissões de **acesso**, **escrita** e **leitura** do mesmo deverão ser liberadas para que o NetBeans e o usuário de contexto do Apache Tomcat possam acessar os arquivos dentro dele. Essa recomendação é importante para sistemas operacionais **Linux**.

Geralmente, isso é feito com os seguintes comandos no terminal como usuário **root**:

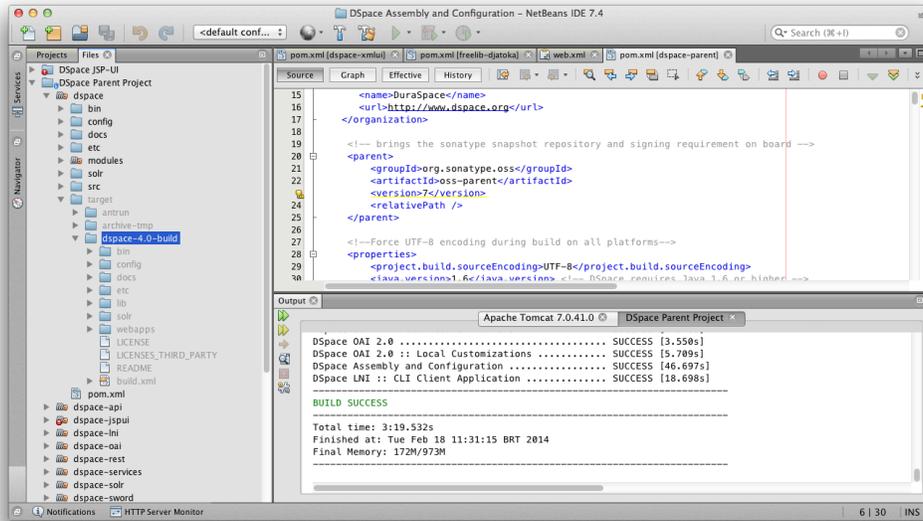
- `mkdir /usr/share/dspace`
- `cd /usr/share/dspace`
- `chown -hR <seu usuário normal no linux, não o root> .` (com ponto no final, representando o diretório atual)

Caso você esteja utilizando um computador com sistema operacional **windows**, altere esse diretório, mudando o valor da variável no arquivo acima para um local conveniente no seu sistema de arquivos.

## 6. Compilar o projeto principal e Instalar o DSpace

Na visão de projetos do NetBeans, clique com o botão direito sobre o projeto **DSpace Parent Project** e depois em **Build**. Isso compilará toda a solução, os módulos e aplicações web da solução.

Acesse a visão de arquivos do projeto principal (**DSpace Parent Project**), sub-projeto **dSPACE**, diretório **target**. Dentro desse diretório, deverá haver um novo sub-diretório chamado **dSPACE-4.0-build**. Veja a imagem abaixo:



Encontre o arquivo **build.xml** e clique nele com o botão direito do mouse. Na opção **Run Target**, você deve selecionar a opção **fresh\_install**. Literalmente, o netbeans criara as tabelas do banco de dados e fara o deploy da aplicação no diretório **/usr/share/dspace** (criado anteriormente neste tutorial). A liberação das permissões nesse diretório e necessária para que o netbeans possa copiar os arquivos da solução apropriadamente.

A compilação deveser acontecer sem nenhum problema. Caso algum erro ocorra, você deve verificar novamente as permissões de escrita e leitura para o diretório já comentado, verificar a conexão com o PostgreSQL com o usuário dspace para a base de dados dspace.

## 7. Criação do usuário administrador do DSpace

Nesta etapa, abra o terminal (ou uma aplicacao semelhante no seu sistema operacional) e mude para o diretório **/usr/share/dspace** (usuários de windows deverão considerar o diretório criado particularmente). Acesse o diretório **bin** para que um comando seja executado.

Execute o seguinte comando: **./dspace create-administrator** (com ponto no começo para sistemas linux). Algumas informações para o seu registro de administrador serão solicitadas, como e-mail, nome completo e senha. Esse será o seu acesso administrador para o seu dspace (independente de você utilizar a aplicação feita em Java Server Pages ou Apache Cocoon).

## 8. Rodar a aplicação web do DSpace (Apache Cocoon)

Após todas essas etapas, o seu ambiente de desenvolvimento está pronto para ser rodado. Na visão de **projetos**, em **DSpace Parent Project** e dentro de módulos, clique duas vezes em 2 subprojetos, um chamado **dspace-solr** e outro **dspace-xmlui**. Isso fará com que a

versão construída em Apache Cocoon seja aberta separadamente. E também o módulo do Apache Solr Webapp será aberto separadamente.

Ainda na visão de projetos, clique com o botão direito do mouse sobre o projeto **Apache Solr Webapp** e depois em **Run**. Isso iniciará uma série de etapas no NetBeans, como a compilação do projeto, o *start* do Apache Tomcat e o deploy da aplicação. Logo depois, o navegador escolhido será aberto com o endereço do solr (localhost:porta/solr/).

Agora é a vez de executar o subprojeto **DSpace XML-UI (Manakin)**. Clique com o botão direito sobre esse subprojeto e depois em **Run**. Desta vez, o navegador configurado será aberto com a aplicação do dspace, na versão construída em apache-cocoon (localhost:porta/xmlui/).

## 9. Rodar a aplicação web do DSpace (Java Server Pages)

Após todas essas etapas, o seu ambiente de desenvolvimento está pronto para ser rodado. Na visão de **projetos**, em **DSpace Parent Project** e dentro de módulos, clique duas vezes em 2 subprojetos, um chamado **dspace-solr** e outro **dspace-jspui**. Isso fará com que a versão construída em Java Server Pages seja aberta separadamente. E também o módulo Apache Solr Webapp será aberto separadamente.

Ainda na visão de projetos, clique com o botão direito do mouse sobre o projeto **Apache Solr Webapp** e depois em **Run**. Isso iniciará uma série de etapas no NetBeans, como a compilação do projeto, o *start* do Apache Tomcat e o deploy da aplicação. Logo depois, o navegador escolhido será aberto com o endereço do solr (localhost:porta/solr/).

Agora é a vez de executar o subprojeto **DSpace JSP-UI**. Clique com o botão direito sobre esse subprojeto e depois em **Run**. Desta vez, o navegador configurado será aberto com a aplicação do dspace, na versão construída em java server pages (localhost:porta/jspsui/).

## 10. Instalação do Drupal com PostgreSQL (Mac OS X)

Os seguintes passos de instalação do Drupal foram realizados no sistema operacional Mac OS 10.9.1 (Maverick).

### 1. Apache HTTPD:

- a. O Mac OS X já possui o Apache HTTPD instalado. Para iniciá-lo, basta executar o comando abaixo.

```
$ sudo apachectl start
```

### 2. PHP:

- a. O PHP também já está instalado no Mac OS X, para habilitá-lo no apache, basta editar o arquivo `/etc/apache2/httpd.conf` e remover o comentário na linha:

```
#LoadModule php5_module libexec/apache2/libphp5.so
```

- b. Para verificar o funcionamento do PHP, reinicie o apache HTTPD.

### 3. PostgreSQL:

- a. Instalar o PostgreSQL. No caso do Mac OS X, o modo mais simples de fazê-lo é baixar a versão do PostgreSQL do "EnterpriseDB" no site: <http://www.postgresql.org/download/macosx/>. A instalação do PostgreSQL é simples e basta seguir as instruções do instalador.
- b. Crie um novo "role" ou usuário utilizando o pgAdmin ou o shell psql. Crie um usuário com o nome "drupal" e senha "[password]".
- c. Crie um novo banco de dados utilizando uma das ferramentas anteriores. Crie um banco de dados com o nome "drupal" e dono (owner) "drupal".

### 4. PDO\_PGSQL (permite a interação entre PHP e o PostgreSQL):

- a. Obter a versão atualmente instalada no sistema rodando o comando.  

```
$ php -v
```
  - b. Baixar em <http://us.php.net/releases/> a exata versão mostrada pelo comando anterior.
  - c. Descompactar o pacote baixado e entrar na seguinte pasta:  

```
.../php-x.y.z/ext/pdo_pgsql/
```
  - d. Executar o comando:  

```
$ phpize
```
  - e. Executar o comando (no caso do PostgreSQL da EnterpriseDB, o path de instalação padrão é /Library/PostgreSQL/[versão]/):  

```
./configure --with-pdo-pgsql=[path da instalação do
```
- postgresql]
- f. Executar o comando:  

```
$ make
```
  - g. Executar o comando:  

```
$ sudo make install
```
  - h. Editar o arquivo php.ini (no caso do Maverick, este arquivo está localizado em /private/etc/), adicionando a seguinte linha na seção "Dynamic Extensions".  

```
extension=pdo_pgsql.so
```
  - i. Reiniciar o Apache HTTPD.

### 5. Drupal:

- a. Depois de instalar as ferramentas anteriores e reiniciar o Apache HTTPD, baixe o Drupal. Descompacte o pacote baixado e copie todo o conteúdo da pasta descompactada na pasta do Apache HTTPD. (No caso do Mac OS X, esta pasta fica localizada em /Library/WebServer/Documents/).
- b. Acesse no browser: <http://localhost/>.

c. O processo de instalação se iniciará automaticamente. Basta seguir as instruções para completar a instalação do Drupal, colocando as configurações para o banco de dados PostgreSQL criado anteriormente.

6. **Caso ocorra algum problema durante a execução do passo 4.d.** É necessário instalar o software gerenciador de pacotes BREW (<http://brew.sh/>) e rodar os seguintes comandos para instalar o autoconf.

```
$ brew update
```

```
$ brew install autoconf
```

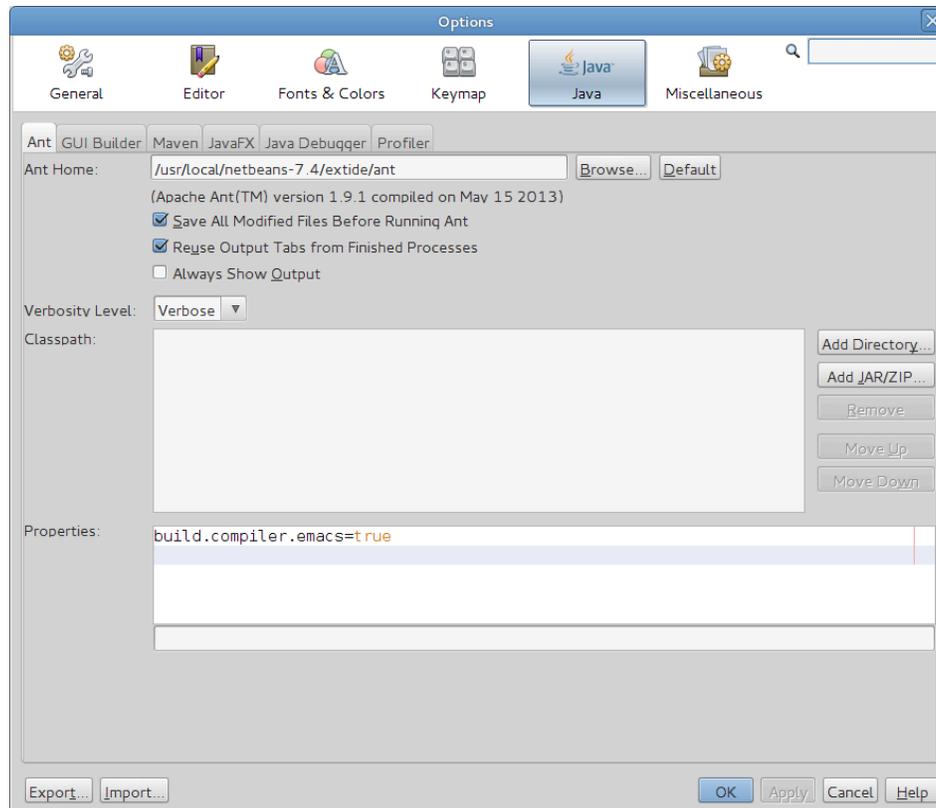
## 11. Como proceder com o deploy remoto do DSpace 4

Antes de começar a pensar no deploy remoto, automatizado e desenvolvido no netbeans para a Corisco 3, você precisa ter certeza qual versão do dspace já está instalada no servidor. Ou seja, este procedimento não confere a versão do dspace instalada e se você insistir em fazer o deploy mesmo assim, sem saber alguns detalhes, vai acabar com problemas na aplicação.

Você deve parar o apache tomcat rodando o dspace ou acessando o manager do tomcat e parando cada uma das 3 webapps (xmlui, jsui e solr).

**ATENÇÃO, CONFIRA A VERSÃO DO DSPACE NO SERVIDOR E TENHA CERTEZA ANTES DE REALIZAR O DEPLOY REMOTO PARA NÃO HAVER CONFUSÕES ENTRE VERSÕES.**

O deploy remoto desenvolvido por mim não utiliza as bibliotecas padrão do **Apache Ant** e portanto, você precisará baixar a versão de uma biblioteca de terceiros chamada [JSCH](#). Baixada a biblioteca complementar, você precisa entrar nas configurações do seu NetBeans, na seção Java particularmente pelo menu *Tools > Options*. Veja o printscreen abaixo:



Acesse o diretório configurado em **Ant Home**. E copie a biblioteca JSCH para o mesmo (você precisará de um usuário **root**, dependendo do local desse diretório).

### **Reinicie o NetBeans!**

Antes de iniciar o procedimento de fato, vale a pena eu ressaltar alguns pontos importantes:

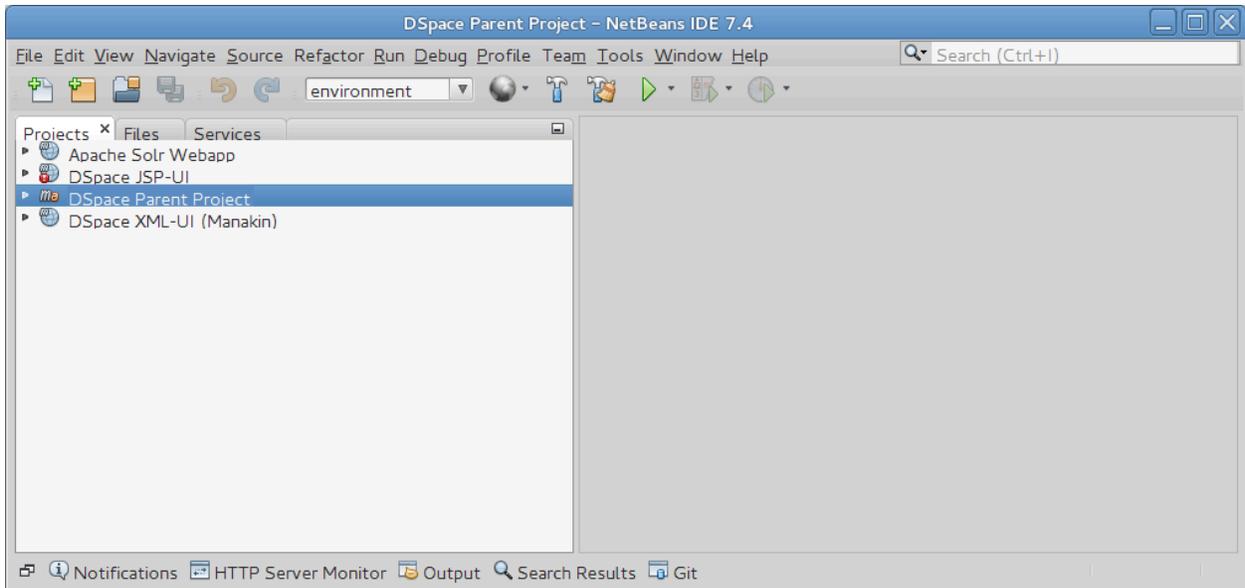
1. Como precisamos seguir um padrão tanto no **desenvolvimento**, quanto na **instalação** da Corisco 3 (pois o projeto almeja atender diversas instituições), foi criada uma *receita* para instalar a solução do zero, inclusive contando com todos os softwares periféricos. [Clique aqui para acessar o documento](#). **Essa receita foi feita toda para a versão 4.1 do DSpace;**
2. Eu mesmo adotei o documento citado acima e por isso alguns detalhes do deploy remoto se *encaixam* com a receita padrão, como o usuário que fará o deploy, que é o mesmo usuário de contexto do tomcat, que é o mesmo nome de usuário na base de dados PostgreSQL. **Ou seja, alterar o nome de usuário pensado no deploy remoto poderá acarretar em problemas de permissão, porque o owner do diretório de instalação do dspace também será esse mesmo usuário;**
3. O apache maven utiliza o arquivo **dspace.cfg** (localizado após a compilação em [dir. do código-fonte dspace] / dspace / target / dspace-4.x-build / config) também para transmitir o valor das variáveis configuradas nos arquivos **build.deploy.properties** e

build.properties. Particularmente para este procedimento, onde configuraremos informações para login ssh, você precisa tomar cuidado para que o arquivo dspace.cfg *compilado* com as configurações reais não vaze para uma pessoa não envolvida com o procedimento. Ou seja, pelo pude entender até agora, o dspace.cfg final conterá as variáveis de login do ssh (tome muito cuidado com essas informações de login);

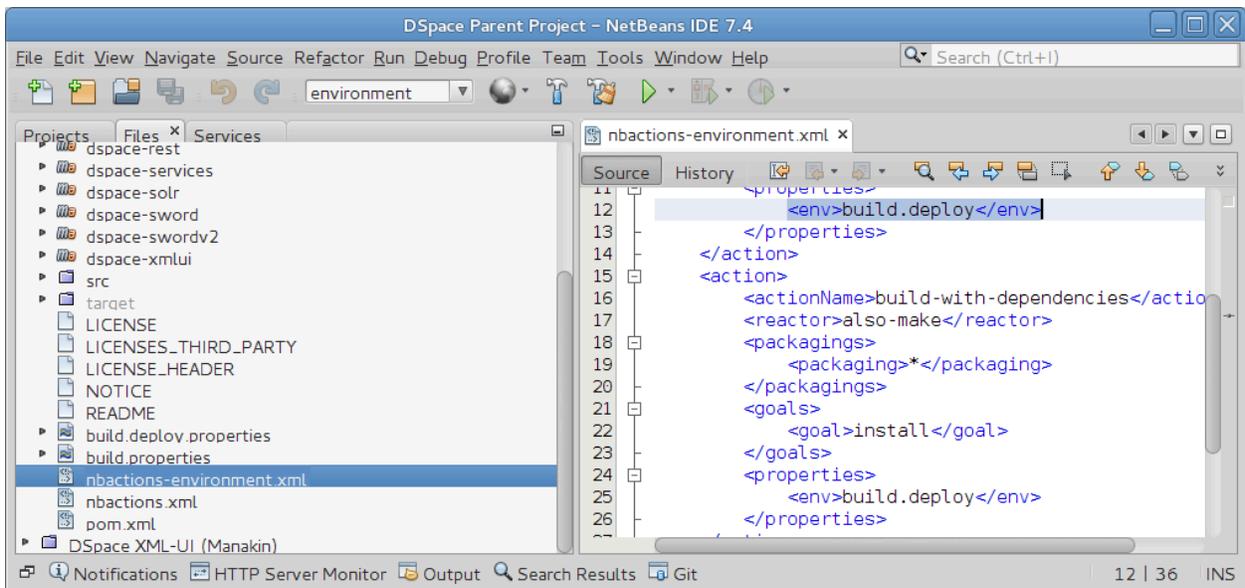
4. Por se tratar apenas do **deploy parcial do dspace**, instalando os módulos **xmlui**, **jspui** e **solr**, não precisaremos nos preocupar com todas as variáveis presentes no arquivo **build.deploy.properties**. Apenas as variáveis **ssh.port**, **ssh.host**, **ssh.username**, **ssh.password** e **dspace.dir**;
5. Você tem duas opções de repositório: a oficial do projeto ([clique aqui](#)) ou a [minha pessoal](#) onde eu transferei todos os meus progressos no projeto. Até aqui, é obrigatório você saber como trabalhar com git e por exemplo, baixar o código-fonte de uma tag específica, mudar de branch no repositório... **o objetivo desta seção não é ensinar como se trabalhar com o git!!!**
6. O deploy é feito utilizando comandos **SSH** e, portanto, o servidor onde será feito o deploy deverá estar com o ssh **funcionando, ativo e disponível na rede** para que este procedimento ocorra sem problemas. **É sempre bom você testar a conexão ssh pelo terminal antes de partir para o deploy, porque a sua máquina pode estar com o ssh desabilitado também!**

A branch no repositório oficial que você pode utilizar é a [dspace-4\\_x](#). Já eu copiei meus progressos numa branch [dspace-4.1](#). Você precisa escolher um código-fonte particular ao projeto Corisco 3, porque foi onde o procedimento do deploy foi gravado!!!! Agora, chega de blá-blá-blá (risos).

Com o NetBeans aberto no projeto, você precisa alterar o perfil (profile) ativo na solução. Independente da visão que você estiver, *projects* ou *files*, você deve alterar o **active profile** de <default config>, ou o que estiver aparecendo, para o perfil **ENVIRONMENT**. Esse perfil foi criado especialmente pelo pessoal do DSpace para termos uma opção de separar os arquivos .properties. Explicação: no código-fonte oficial do **DSPACE**, não há nenhum arquivo **build.deploy.properties**. Esse arquivo foi criado pelo projeto Corisco 3 para o tal deploy. Veja a figura abaixo com o perfil alterado (environment):



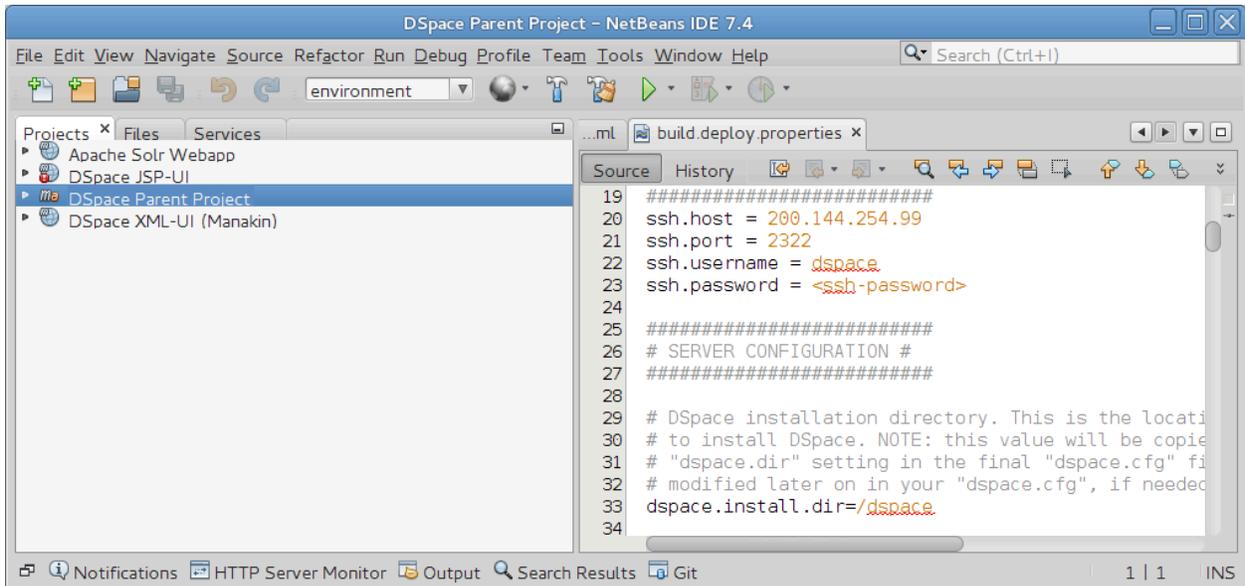
Mas por que o nome do arquivo para o perfil **Environment** deve se chamar **build.deploy.properties**? Uma propriedade foi configurada por mim para que **esse fosse o nome de arquivo adotado**. Na visão de arquivos do NetBeans, abra o arquivo **nbactions-environment.xml**: o valor da propriedade **env** é (adivinhem) **build.deploy**.



Agora vamos alterar algumas propriedades do arquivo **build.deploy.properties**:

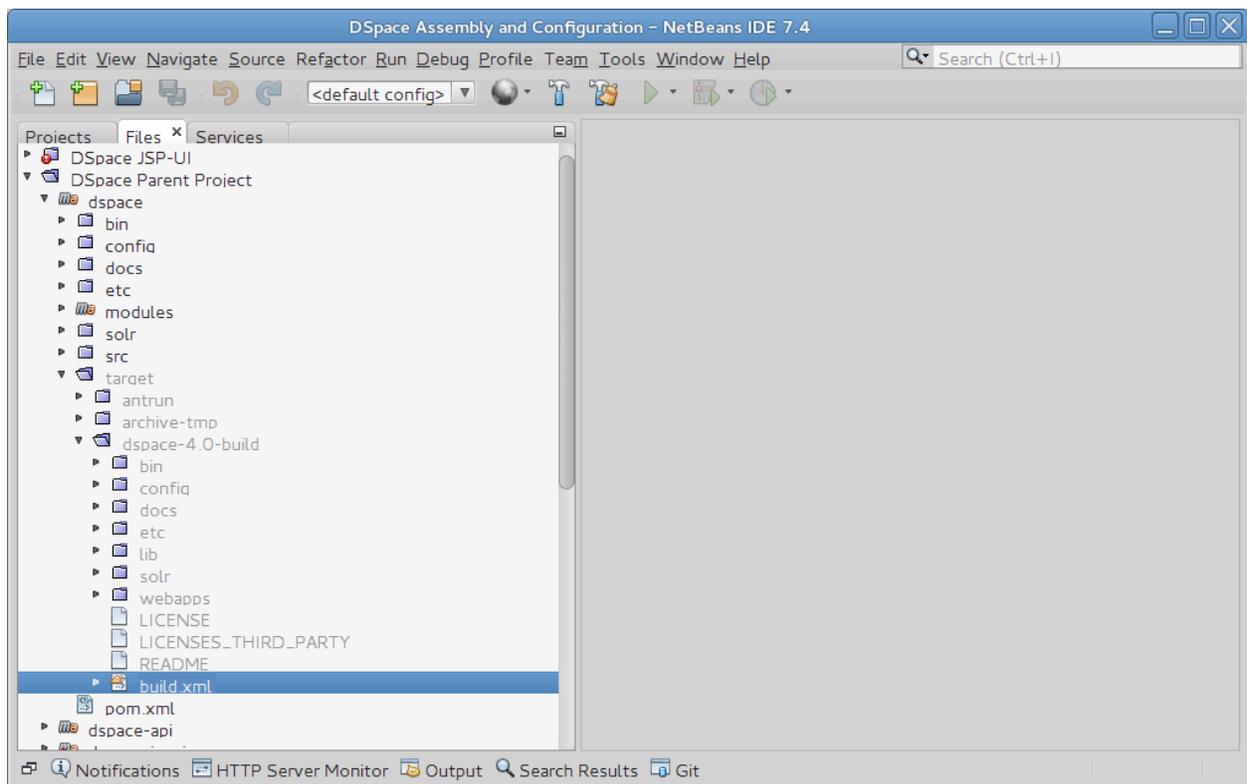
- ssh.host**: insira o ip ou o nome do domínio do servidor da aplicação
- ssh.port**: digite a porta do ssh do servidor

- ❑ **ssh.username**: por padrão, **dspace** (não há necessidade de utilizar outro usuário, caso a receita padrão foi utilizada na instalação)
- ❑ **ssh.password**: a senha do usuário de login
- ❑ **dspace.install.dir**: o diretório onde o dspace foi instalado no **SERVIDOR**



Vá para a visão de Projetos no NetBeans, clique com o botão direito em **DSpace Parent Project** > **clean** e depois **build**. Isso fará com que o dspace seja compilado também com as informações do ssh (por isso, tome muito cuidado para que o seu login não vaze).

Terminada a compilação, agora na visão de arquivos do NetBeans, acesse o seguinte diretório: DSpace Parent Project > dspace > target > **dspace-4.x-build**. Você deve encontrar o arquivo **build.xml**.



Clique com o botão direito sobre build.xml e escolha **Run Target > remote\_deploy\_webapps**. Pronto! Agora você deve aguardar que as etapas do deploy sejam completadas.

Como isso depende da velocidade da sua conexão com o servidor, você precisa esperar que todas as etapas terminem! E execute novamente o tomcat ou reative as webapps paradas no início deste procedimento.