**DOCUMENTO DE ESPECIFICAÇÃO**

**DE CONFIGURAÇÃO DO SISTEMA SEIA**

Sistema Estadual de Informações Ambientais e de Recursos Hídricos – SEIA

AGOSTO/2021

**Versão 3.0**

|  |  |  |  |
| --- | --- | --- | --- |
| Controle de Mudança | | | |
| **Versão** | **Data** | **Mudança** | **Responsáveis** |
| 1.0 | 01/02 | Criação do Documento | Matheus Pereira dos Passos Oliveira |
| 1.1 | 05/04 | Revisão do Documento | Matheus Pereira dos Passos Oliveira |
| 2.0 | 24/05 | Revisão do Documento | Matheus Pereira dos Passos Oliveira |
| 2.1 | 25/06 | Revisão do Documento | Matheus Pereira dos Passos Oliveira |
| 3.0 | 27/08 | Revisão do Documento | Matheus Pereira dos Passos Oliveira |
|  |  |  |  |

**Sumário**

[Sumário 3](#_Toc1068715094)

[1. INTRODUÇÃO 4](#_Toc1912063586)

[2. REQUESITOS MÍNIMOS 5](#_Toc415342479)

[3. PREPARAÇÃO INICIAL 6](#_Toc1423797305)

[3.1. LINKS PARA DOWNLOAD 7](#_Toc510509882)

[4. INSTALAÇÃO 8](#_Toc209806804)

[4.1 JAVA 8](#_Toc1097666985)

[4.2 POSTGRESSQL 9.5 9](#_Toc554984514)

[4.2.1 VERIFICANDO INSTALAÇÃO POSTGRESQL 19](#_Toc1888275150)

[4.3 MAVEN 3.6.3 24](#_Toc1388357282)

[4.3.1 CONFIGURAÇÃO DAS VARIÁVEIS DE AMBIENTE 25](#_Toc1332945540)

[4.3.2 ACESSO AO DRIVE DOCUMENTAÇÃO 32](#_Toc1219533858)

[4.5 APACHE2 33](#_Toc332654551)

[4.6 ECLIPSE 35](#_Toc1310537037)

[4.6.1 CONFIGURANDO MAVEN NO ECLIPSE 38](#_Toc1809286360)

[4.6.2 EXTENSÕES PARA O ECLIPSE 40](#_Toc102677071)

[4.7 JBOSS 7.0 41](#_Toc222038841)

[4.8 GIT & GITLAB – CONEXÃO VIA SSH 56](#_Toc1818251514)

[4.8.1 UTILIDADES COMANDOS GIT 68](#_Toc1005249342)

[4.8.2 GITLAB – CONEXÃO VIA SSH 69](#_Toc1595201599)

[5. Adicionais que você precisa saber sobre a SEMA/SEIA 73](#_Toc869072012)

[6. GERAÇÃO DE VERSÃO 77](#_Toc774267611)

[7. RESOLUÇÃO DE ERROS COMUNS DOS TICKETS 86](#_Toc1314723397)

# INTRODUÇÃO

O documento foi criado para facilitar a configuração do sistema para minimizar possíveis complicações para a instalação do sistema e manutenibilidade do sistema SEIA. São abordadas todas as dificuldades que foram enfrentadas até o funcionamento do sistema. Como todos nós sabemos é necessário atender uma especificação mínima de Hardware\* para que o sistema funcione corretamente.

Recomenda-se que siga atentamente todos os passos de forma rígida, sem alterar o fluxo ou mudar os passos descritos sucintamente neste documento.

**Palavras - Chave**

\* Hardware parte física do computador. Ex: PROCESSADOR, MEMÓRIA, CPU, HD, SSD etc.

# REQUESITOS MÍNIMOS

**PROCESSADOR: QUAD CORE OU ACIMA**

**MEMÓRIA RAM: 8GB OU ACIMA**

**HD: 500GB OU ACIMA**

**SSD: 240GB OU ACIMA**

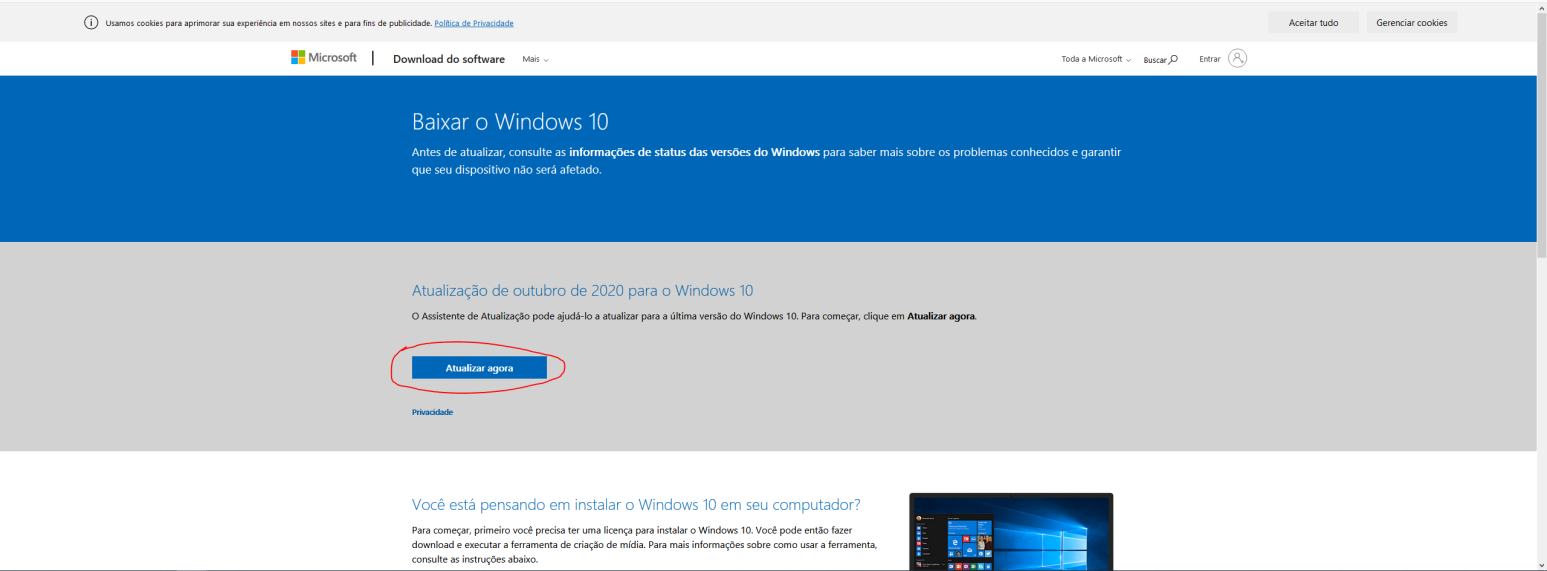
**SISTEMA OPERACIONAL: Windows 10 PRO**

**SISTEMA: 64x bits**

**Obs: Todos os programas citados estão na arquitetura 64x, entretanto, alguns que não forem x64 serão sinalizados na documentação.**

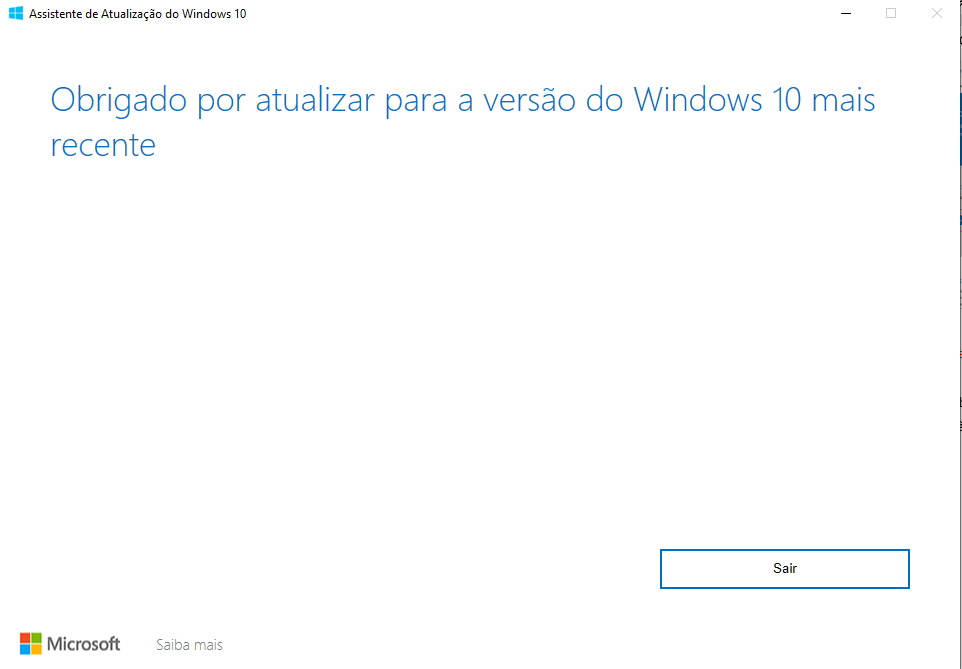
# PREPARAÇÃO INICIAL

Partindo do pressuposto que o usuário já formatou sua máquina com o Windows 10, eu recomendo que entre nesse link: <https://www.microsoft.com/pt-br/software-download/windows10> - e atualize sua versão do Windows para 20H2 ou superior. Favor clicar em atualizar agora conforme a imagem abaixo:



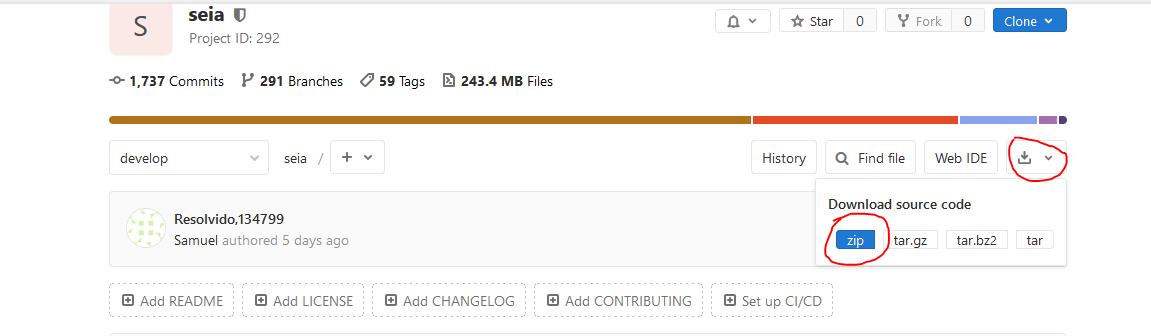
OBS: Caso tenha dificuldade para visualizar a imagem acima favor alterar o zoom do documento word.

Em seguida abra o arquivo baixado e deixe o mesmo atualizar os seus arquivos, após a atualização automática você deverá ver essa imagem abaixo:

 \*Reiniciar máquina ao sair\*

# LINKS PARA DOWNLOAD

Após o mesmo receber o usuário do SEMA, a mesma disponibilizará uma vpn, usuário e senha a ser instalada e após instalar e seguir a documentação da VPN, acesse esse link para baixar: <http://gitlab.prodeb.ba.gov.br/> e vá até o projeto base/sema/seia/documentação lá dentro há uma pasta escrita documentação , você pode baixar ela em zip, conforme a imagem abaixo e salvar numa pasta siga para o próximo tópico.



\*Após baixado o arquivo ele virá zipado, você deverá baixar e instalar um zip de sua preferência:

Indicados: Winrar – <https://www.win-rar.com/postdownload.html?&L=0> (após baixar a página terá um passo a passo de como instalar)

**Instalação Padrão – Programas x64 (64bits) – (Program Files) ou (Arquivos de Programas);**

**Instalação Padrão – Programas x86 (32bits) – Program Files (x86) ou (Arquivos de Programas x86);**

**IMPORTANTE:**

\*OBS: Hoje, dia 01/12/2021, o JAVA 15 é mais atual, favor instalar o mais recente acesse a página abaixo:

<https://www.oracle.com/java/technologies/javase-downloads.html>

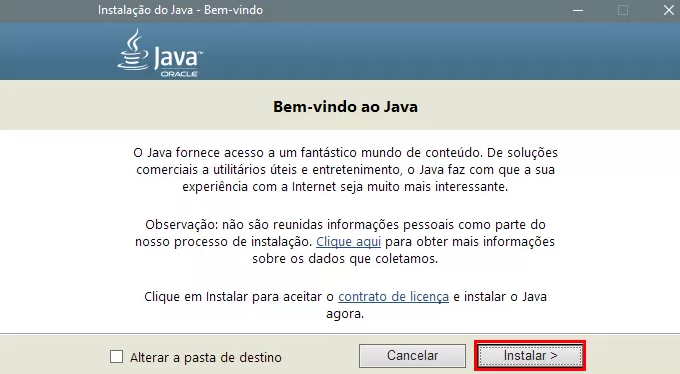
# INSTALAÇÃO

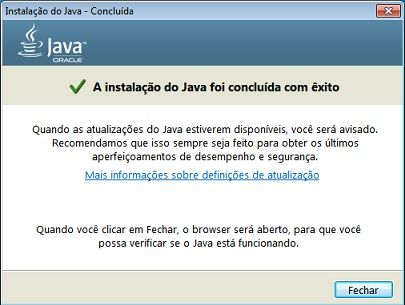
## 4.1 JAVA

Primeiramente vamos realizar a instalação de todos os Java seguindo os passos abaixo:

Abra o programa e siga a área demarcada na figura 1.

Figura 1:





OBS:A instalação do Java é bem simples, basta seguir o next (avançar e concluir no final).

Figura 2

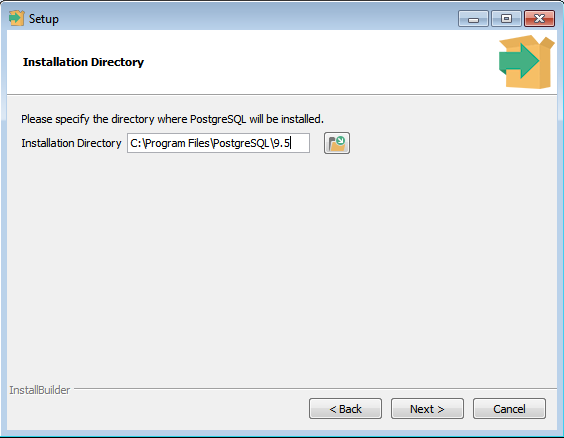
## 4.2 POSTGRESSQL 9.5

1º Passo – Após baixar o instalador, entre na pasta onde o arquivo foi salvo e dê um clique duplo nele para iniciar o processo de instalação. Você verá a janela a seguir:

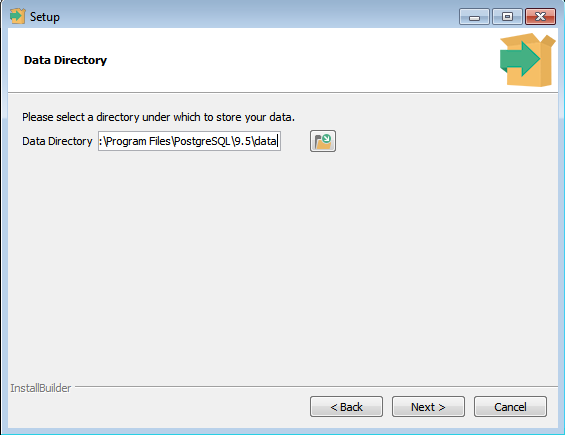


Clique em Next para prosseguir.

2º Passo – Na tela seguinte selecione o local de destino da instalação (vou manter a pasta padrão) e clique em Next para prosseguir:

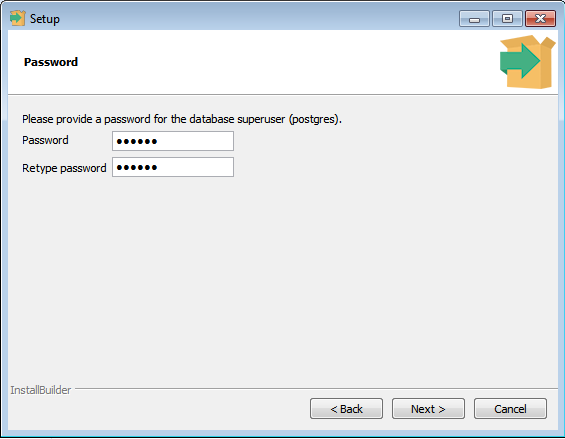


3º Passo – Na tela seguinte escolha o diretório onde os dados serão armazenados (vou manter o padrão também) e clique em Next:

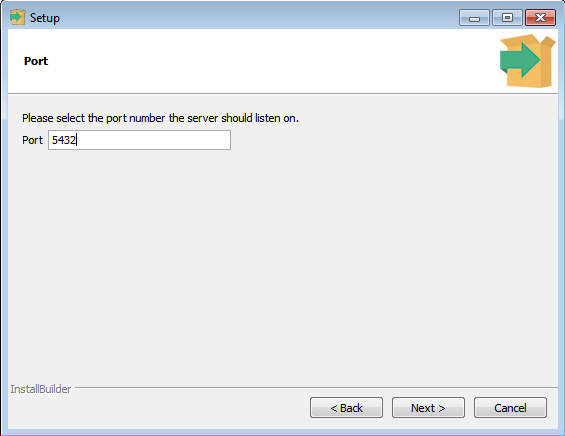


4º Passo – Agora entre com uma senha para o superusuário do banco de dados (postgres), repita-a na caixa de confirmação, e clique em Next para continuar:

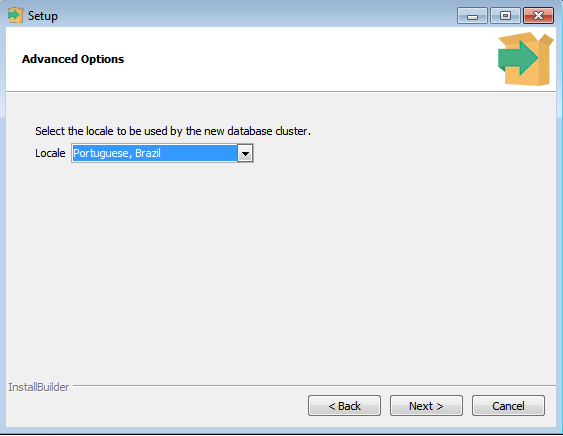
\*Obs: Recomendo colocar a senha inicial: 123456



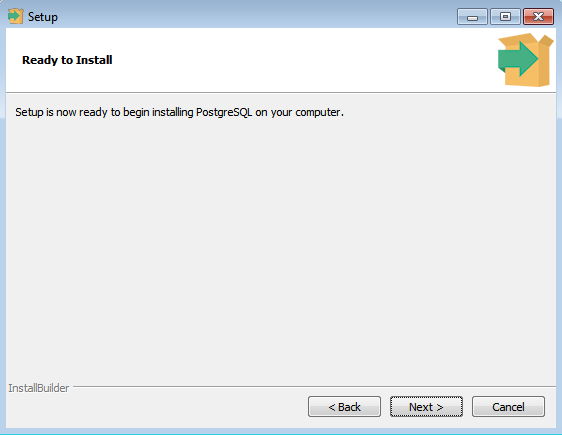
5º Passo – Agora entre com o número da porta que o servidor escutará. Você pode deixar a porta padrão 5432 se desejar. Clique em Next.



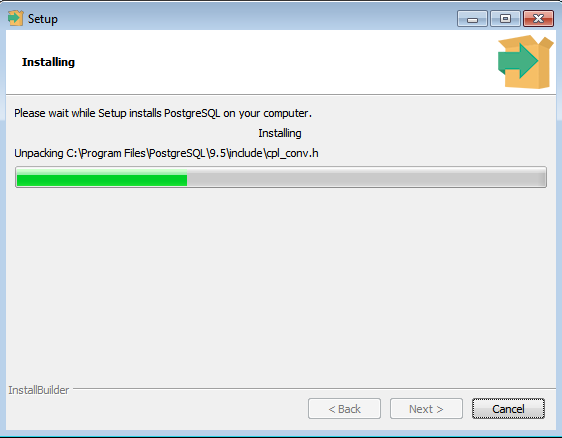
6º Passo – Na tela seguinte selecione o locale a ser usado pelo servidor. Provavelmente será “Portuguese, Brasil”. Clique em Next.



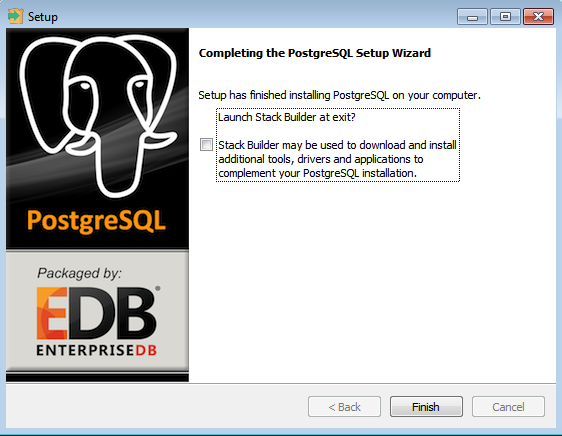
7º Passo – O instalador está pronto agora para realizar as tarefas de instalação. Clique em Next para prosseguir:



8º Passo – Aguarde enquanto a instalação é realizada:



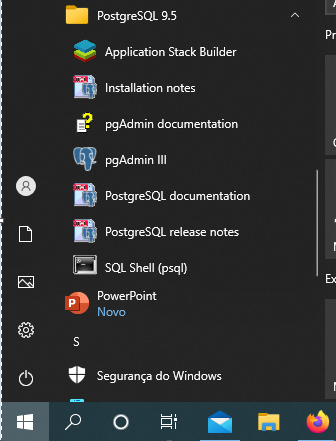
9º Passo – Instalação realizada com sucesso! Desmarque a opção “Launch Stack Builder at exit?”, pois não vamos instalar ferramentas adicionais no momento, e clique em Finish para encerrar o instalador:



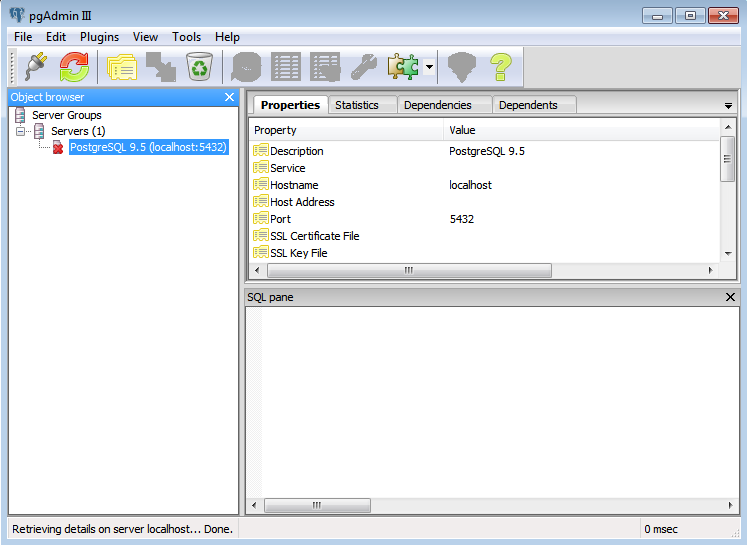
Vamos agora verificar se a instalação foi executada sem problemas e se o PostgreSQL está rodando.

### 4.2.1 VERIFICANDO INSTALAÇÃO POSTGRESQL

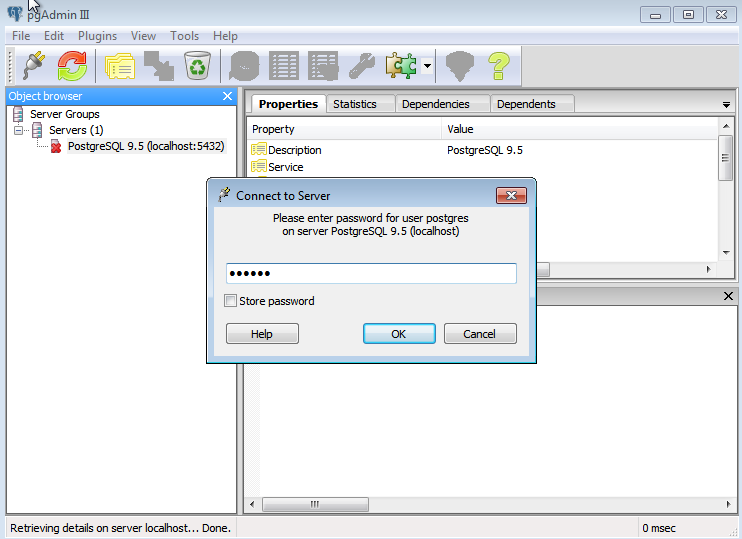
Para verificar a instalação do sistema, vamos efetuar uma conexão a ele. Há várias formas de se realizar essa tarefa, e a mais fácil é usando o utilitário pgAdmin III. Você irá encontrá-lo no menu de atalho criado pelo instalador. Abra-o:



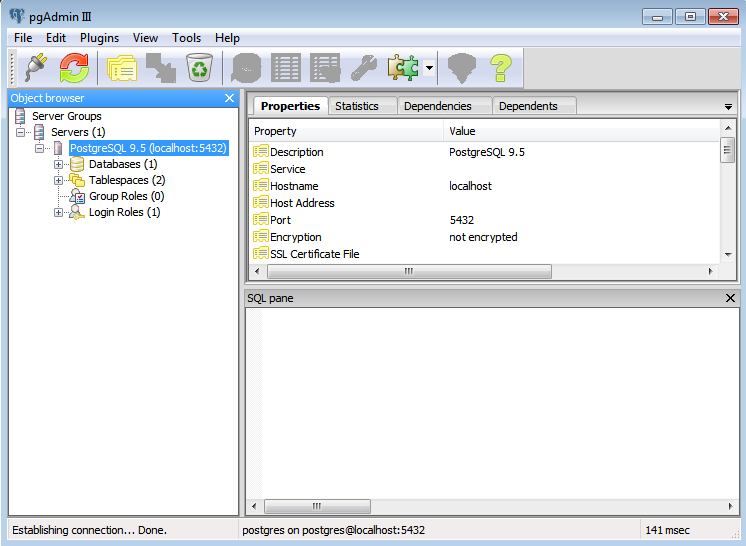
Aí está a tela do pgAdmin III:



Agora dê um clique-duplo sobre o item PostgreSQL 9.5 no Object Browser (lado esquerdo da tela), e entre com a senha do superusuário na caixa de diálogo que irá aparecer. Clique no botão OK.



Se tudo tiver corrido bem, serão exibidos na janela os objetos pertencentes ao servidor:



Obs: Se por acaso aparecer a mensagem de erro “Error running post install step. Installation may not complete correctly. Error reading C:/Program Files/PostgreSQL/9.2/postgresql.conf “, siga os passos abaixo para eliminar o problema:

1. Desinstale o PostgreSQL, e também exclua o diretório de dados na pasta de instalação;
2. Crie um novo usuário no Windows, de nome postgres, e adicione-o ao grupo de administradores;
3. Abra o prompt de comandos digitando o comando a seguir na caixa Executar:

runas /user:postgres cmd.exe

Será aberta uma nova janela – execute o instalador novamente a partir dessa janela (navegue até a pasta onde ele se encontra).

Isso deve resolver o problema.

**Configuração para acessar o banco de dados:**

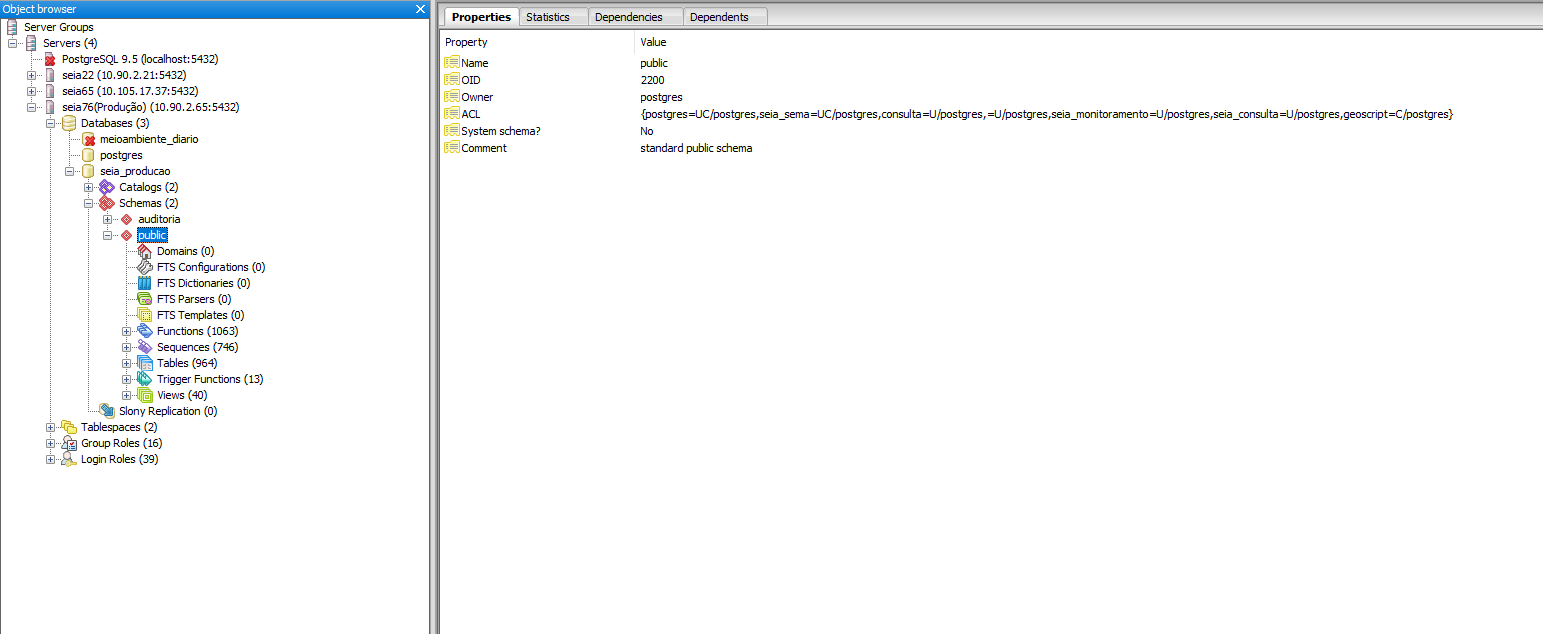
Ambiente 22(Desenvolvimento), Ambiente 65(Homologação)

Senha: @123456

Ambiente 76(Produção) Senha: 123456

EX:

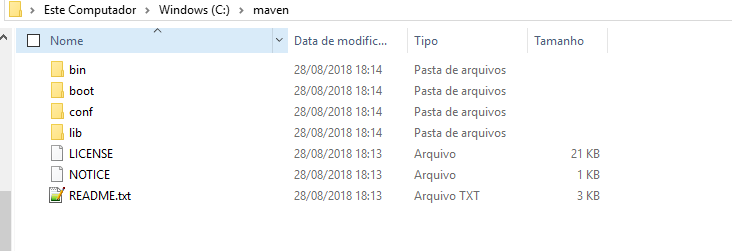
Configurando o ambiente 22, note que deveria ser 10.90.2.22, porém houve uma atualização recentemente no banco de dados e ficou assim 10.90.2.21(chamaremos ele de 22 ainda), o ambiente 65 o endereço é 10.105.17.37(chamaremos ele de 37 por aqui em diante, por causa do final) e 76 o endereço é 10.90.2.65(Chamaremos de 76 ou banco de produção)



\*Caso tenha dificuldade, favor utilizar o zoom.

## 4.3 MAVEN 3.6.3

1º Passo - Após fazer o download do arquivo compactado, você deve descompactá-lo em uma pasta de sua preferência. É importante você saber que esta pasta será a pasta onde o maven estará "instalado". Recomenda-se descompactar o maven em C:\maven. Ao final da descompactação a pasta do maven estará assim:

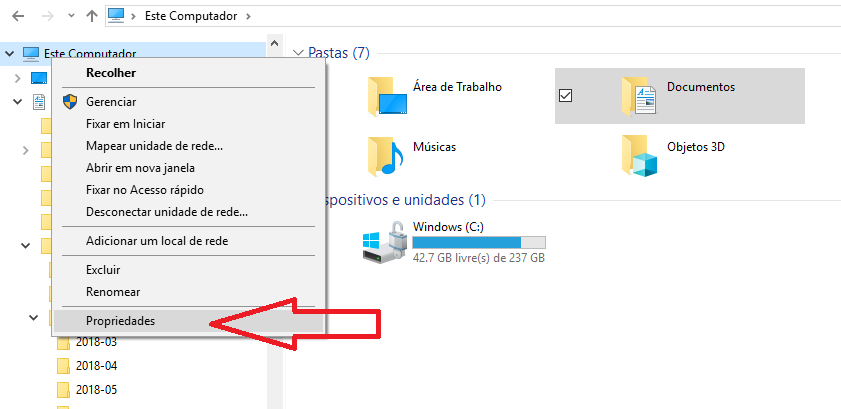


Pode ser que quando você descompactou o arquivo zip a pasta C:\maven tenha ficado só com uma pasta com os arquivos do maven dentro, algo como C:\maven\apache-maven-3.5.4.Não tem problema, basta se lembrar disso quando for configurar a variável de ambiente. Ou você pode mover os arquivos de dentro desta pasta e subir um nível, colocando-os em C:\maven.

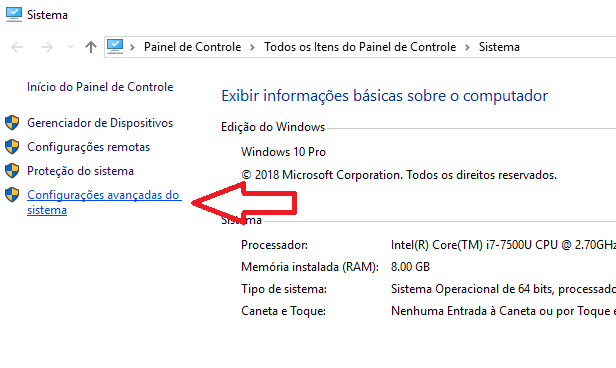
### 4.3.1 CONFIGURAÇÃO DAS VARIÁVEIS DE AMBIENTE

Para fazer o maven funcionar você só precisa adicionar a pasta bin do maven na variável de ambiente PATH. Se você nunca configurou uma variável de ambiente no Windows basta seguir esses passos.

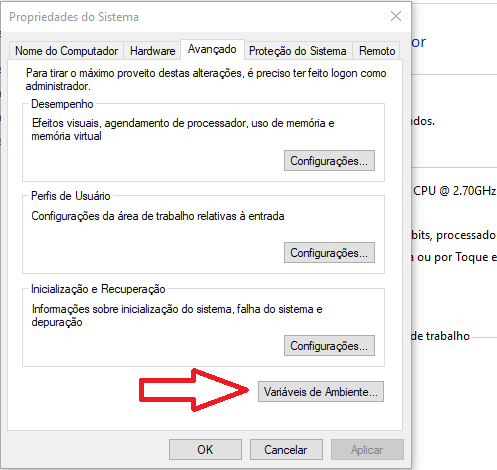
1º Clique com o botão direito do mouse no ícone do seu computador e clique em Propriedades.



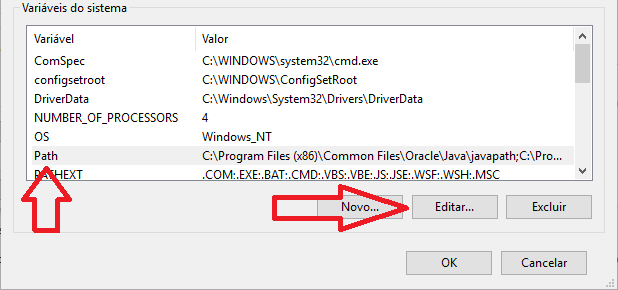
2º Clique em Configurações avançadas do sistema.



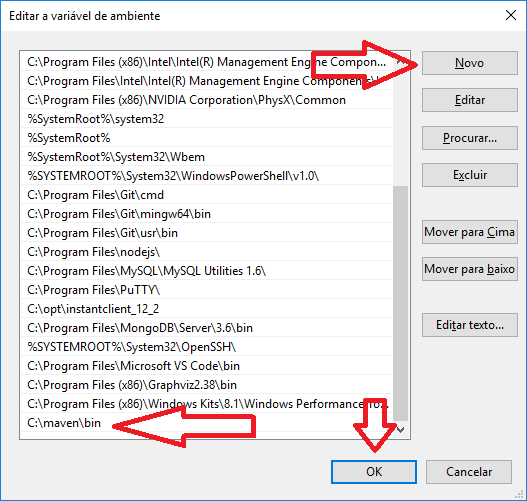
3º Clique no botão Variáveis de ambiente dentro da aba Avançado.

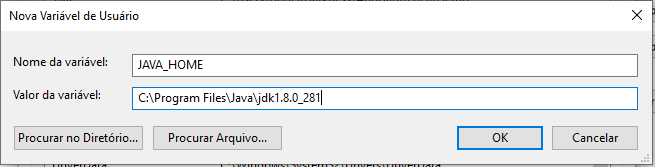


4º Selecione a variável de ambiente em \*Variáveis do sistema\* PATH na lista e clique em Editar.

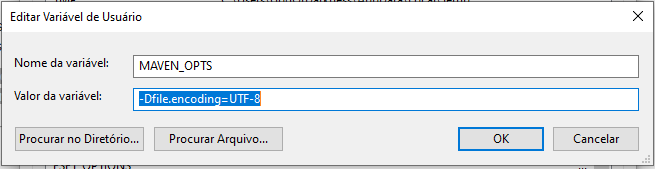


5º Clique no botão Novo e adicione o caminho da pasta bin do maven - C:\maven\bin.

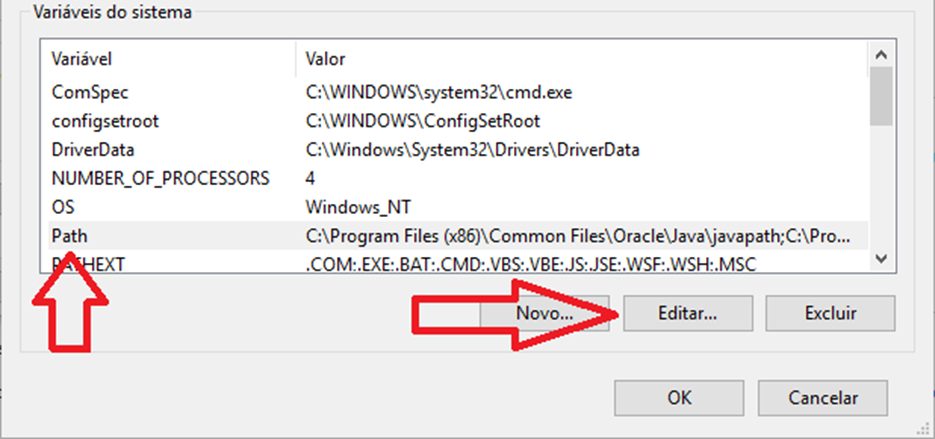


6º Clicando em Novo... em \*Variáveis de Usuário\* e informe o nome JAVA\_HOME e o valor com o caminho para a pasta onde o seu JDK está instalado. Deixe assim: 

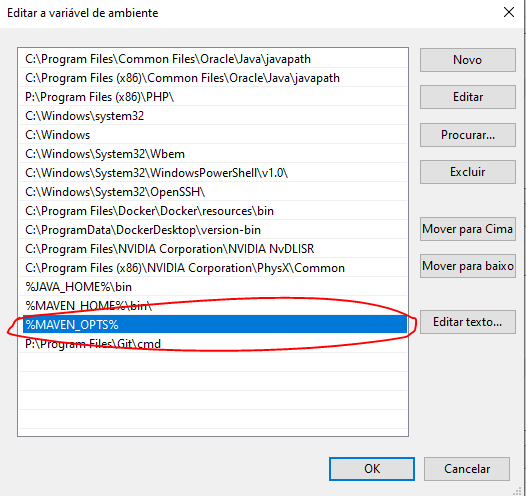
7º Clicando em Novo... em \*Variáveis de Usuário\* e informe o nome MAVEN\_OPTS e o valor com o caminho para esse aqui: -Dfile.encoding=UTF-8



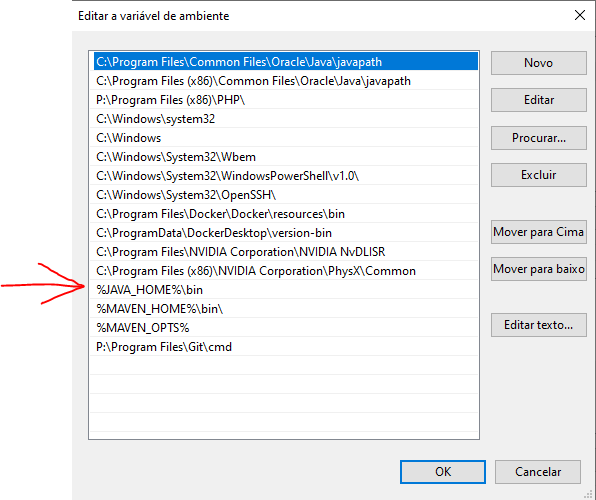
8º Selecione a variável de ambiente em \*Variáveis do sistema\* PATH na lista e clique em Editar.



9º Clique no botão Novo e adicione o caminho da pasta do maven\_opts - %MAVEN\_OPTS%



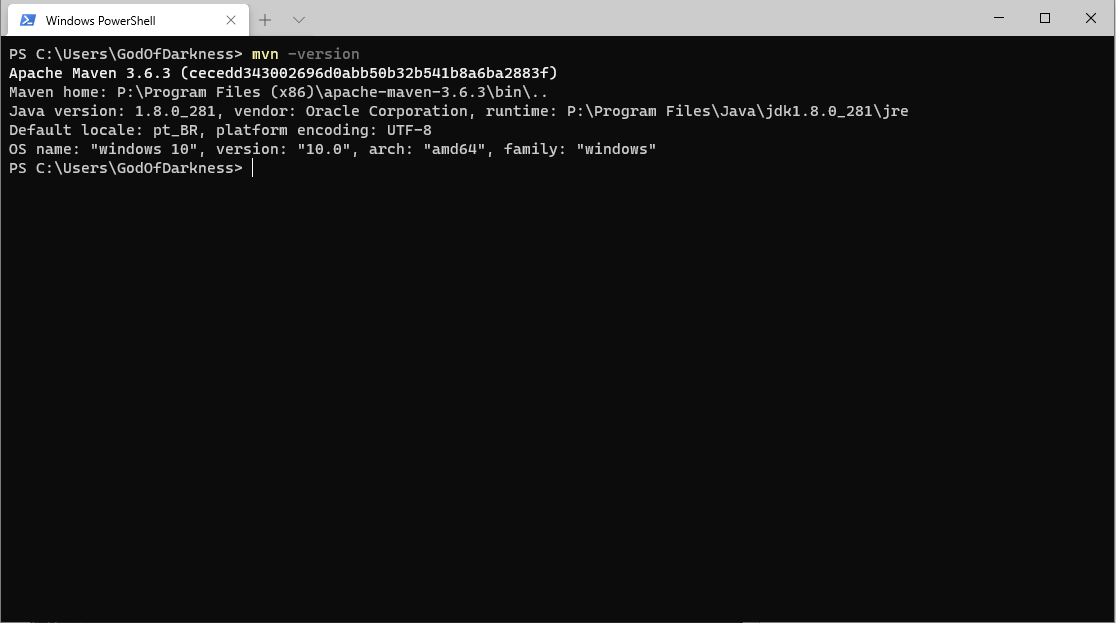
10º Clique no botão Novo e adicione o caminho da pasta do JAVA\_HOME - %JAVA\_HOME%\bin



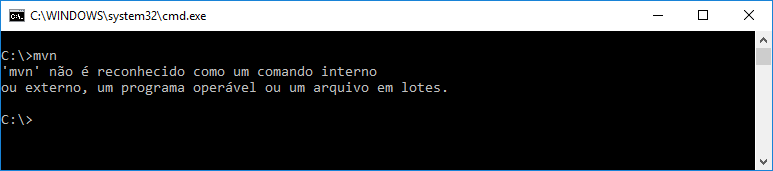
Testando a instalação do maven

Para testar se a configuração do maven funcionou, abra o prompt de comando do seu computador e digite mvn -version.

A saída:



\*Obs: Se notou o P:\ o seu deverá aparecer C:\

Se o resultado do seu comando foi como a imagem abaixo, significa que você não configurou a variável de ambiente PATH com o caminho válido para a pasta bin do maven. 

Repita o procedimento desde o passo 1º e certifique-se que o caminho da pasta bin do maven esteja na variável de ambiente PATH. Após essa correção, feche e abra novamente o prompt de comando e reinicie o seu computador.

### 4.3.2 ACESSO AO DRIVE DOCUMENTAÇÃO

Atualmente temos um gmail com todas as documentações recomendamos fortemente salvar a senha e não alterá-la em hipótese alguma resultando na perda permanente do gmail, não altere nenhuma das configurações de conta.

Email: ***seiasemasustentacao@gmail.com***

Senha: **seia1qaz2wsx3edc**

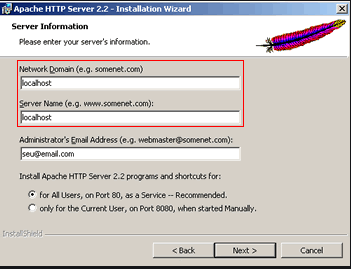
**Acesse o Drive e você vai encontrar uma pasta chamada Documentação e dentro dela há alguns arquivos que devem ser copiados para sua máquina como por ex:**

**A pasta .m2\_.rar copie e substitua sua pasta original m2 criada pelo maven pois senão o projeto SEIA não conseguirá rodar!**

**Assim como o JBoss e a pasta vídeos utilizada para treinamento do sistema.**

## 4.5 APACHE2

1º Passo: Clique no arquivo de instalação e clique em next até a seção Server Information



Preencha os dados, mantenha a porta 80 selecionado.

Clique em NEXT > Na tela Setup type, mantenha a opção Typical(típica)

Defina em Network Domain e Server Name: localhost



2º Passo: Redefina a pasta para C:\apache\ e clique em NEXT >

Clique em INSTALL

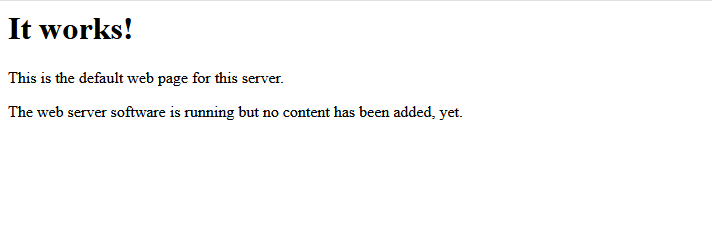
Clique em FINISH, certifique-se de que o ícone do APACHE MONITOR está mostrando que o servidor foi instalado e rodando, próximo ao relógio do windows.

Ex:



Se o ícone estiver como o da figura anterior, significa que o apache foi instalado e já está rodando.

Testando a instalação Apache Para testar abra o navegador e digite o endereço <http://127.0.0.1/> ou digite localhost irá aparecer uma página similar a da figura abaixo:

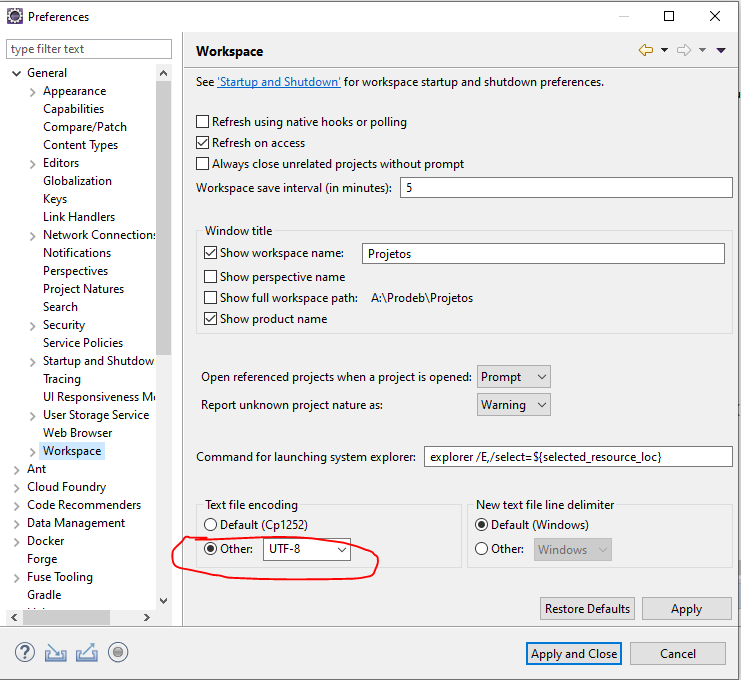


## 4.6 ECLIPSE

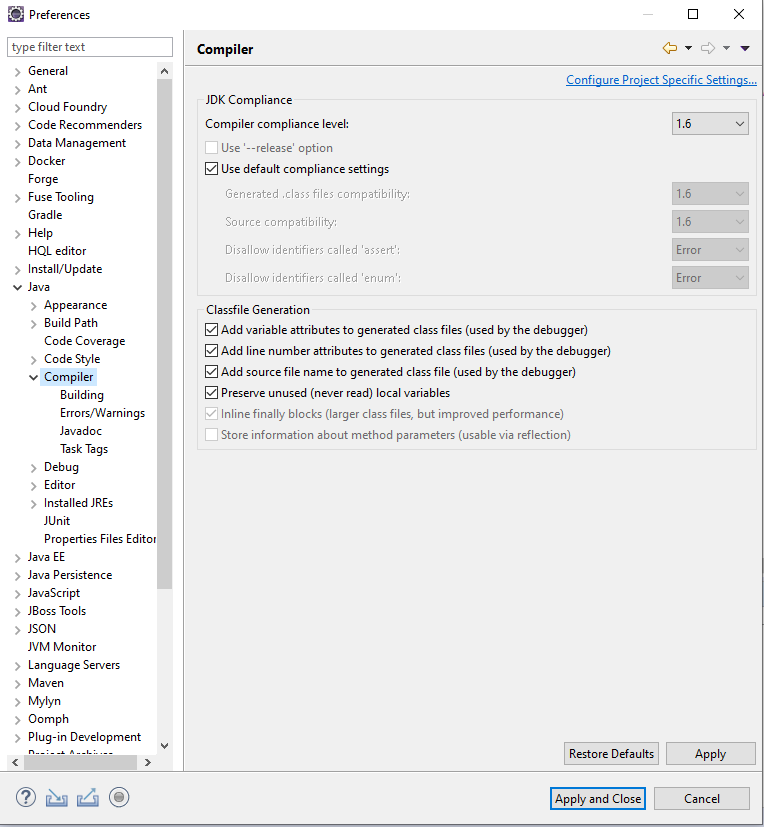
Para baixar o eclipse acesse esse link - <https://www.eclipse.org/downloads/packages/release/2019-03/r/eclipse-ide-enterprise-java-developers>

Após abrir o eclipse vá em Windows – Preferences na aba superior do Software e siga o passo a passo:

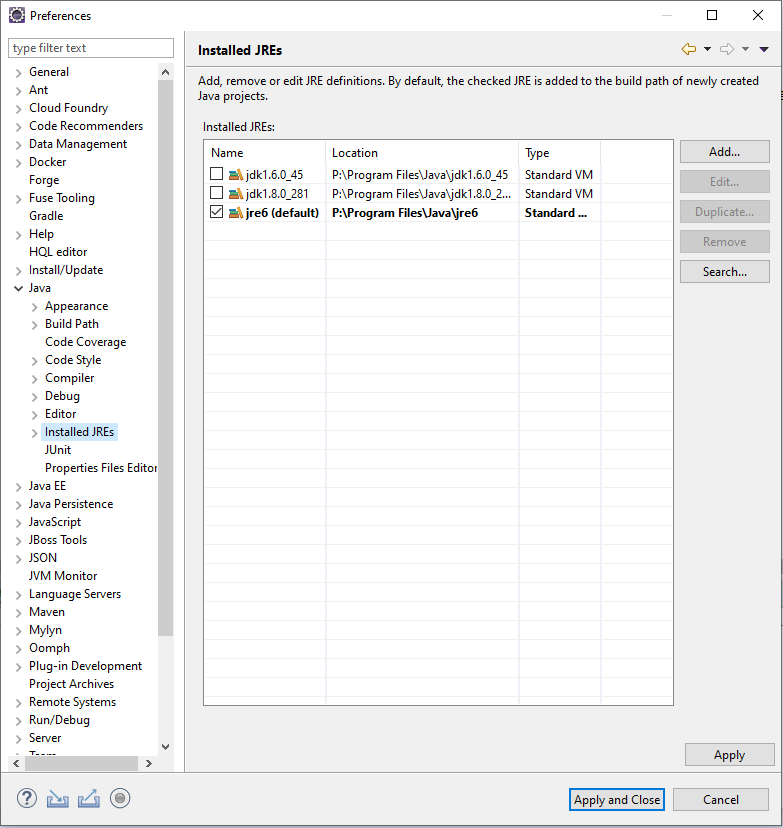
Em General > Workspace:



Java > Compiler:



JAVA > INSTALLED JRES::



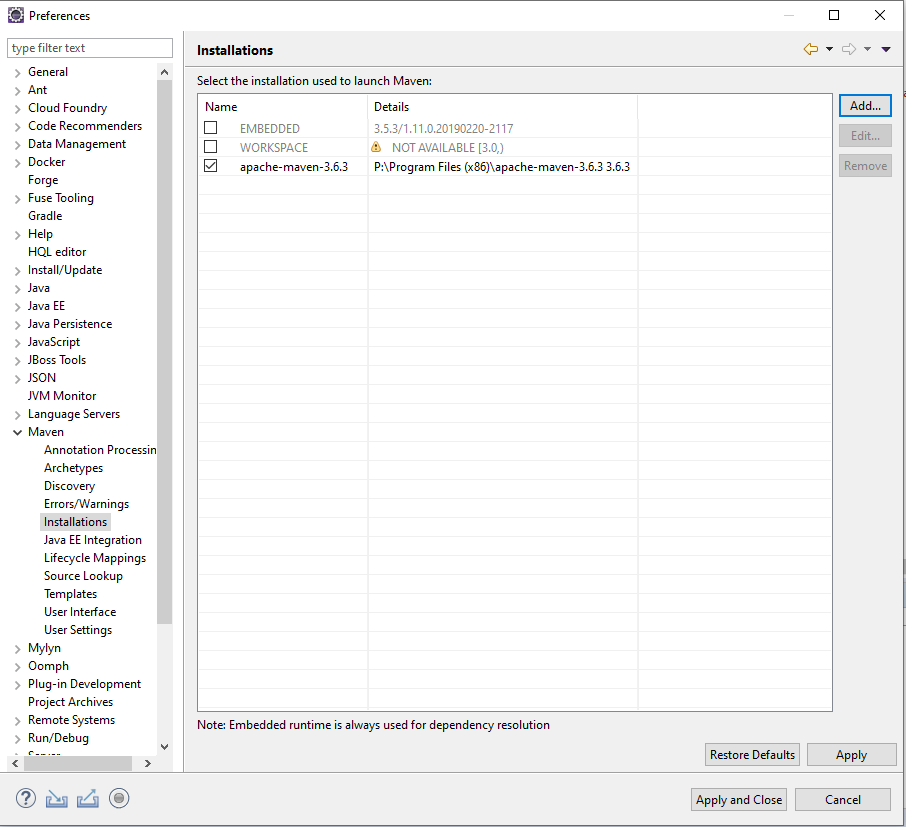
\*OBS: Para adicionar uma nova JREs, clique no botão add, selecione Standard VM > Em JREs Home clique no botão Directory e procure a pasta raiz do seu java e selecione, depois é só dar um FINISH.

### 4.6.1 CONFIGURANDO MAVEN NO ECLIPSE

**(Antes substitua a pasta .m2 - [CLIQUE AQUI](#_4.3.2 ACESSO AO DRIVE DOCUMENTAÇÃO))**

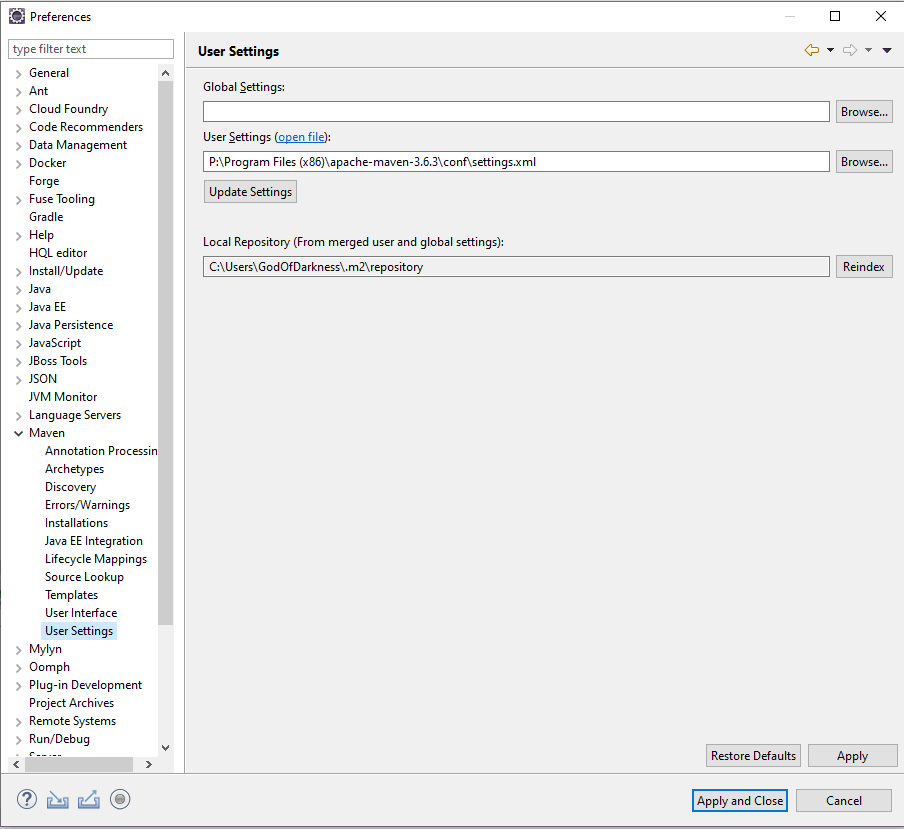
Com o preferences aberto ainda, vá até o MAVEN > Installations e coloque o apache na pasta conforme a imagem:

Clique no add, Installation home e procure o apache em seu computador, o ex abaixo o apache esta instalado no P: , o seu estará no C:



Agora acesse MAVEN > USER SETTINGS:

E aponte sua pasta M2 criada e seu User settings, abstraia o P: , o seu será o C: conforme a imagem abaixo:



Note que na pasta m2 tem outra pasta chamada repository para organizar numa preferência de usuário, todas as bibliotecas utilizadas.

Ainda com o preferences aberto acesse o Validation e clique em Disable all para não impactar em muito tempo esperando o projeto carregar.

### 4.6.2 EXTENSÕES PARA O ECLIPSE

Acesse o Eclipse > HELP > ECLIPSE MARKETPLACE

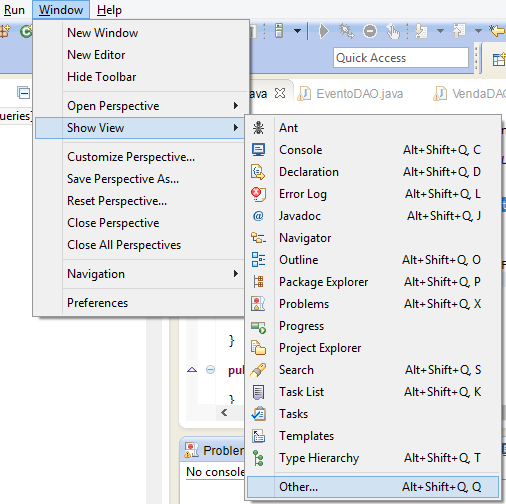
1. DARKEST DARK THEME
2. SONARLINT
3. CODETOGETHER
4. ECLIPSE ENTERPRISE JAVA AND WEB DEVELOPER

## 4.7 JBOSS 7.0

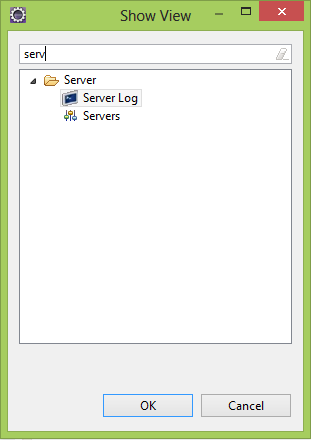
O jboss utilizado é o 7.0 com o java SE 1.6, o jboss para ser configurado no eclipse pode ser conferido abaixo, favor baixar a pasta em Programas no GitHub e o seu jBoss já virá configurado, coloque-o no disco C:

**Configuração**

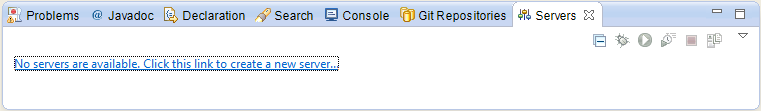
Com o Eclipse aberto, vá no Menu Window > Show View > Other…, conforme Figura 01 abaixo:

 Figura 01 – Acessando a opção Show View > Other

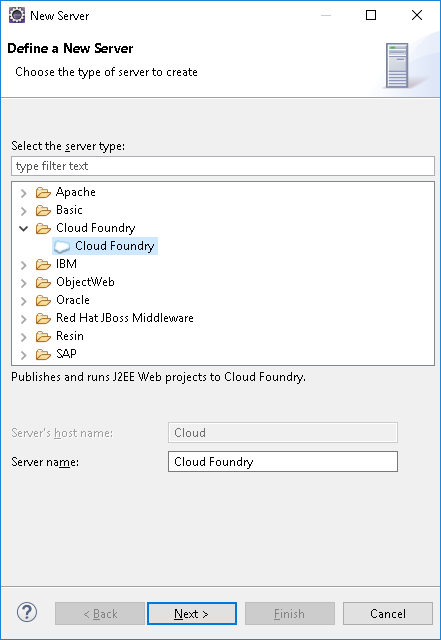
O Eclipse irá exibir uma popup chamada **Show View**. Na textbox que aparece comece a digitar a palavra ***server*** e o Eclipse irá filtrar as opções de view deixando apenas 2, Server Log e Servers, conforme Figura 02.

 Figura 02 – Popup Show View

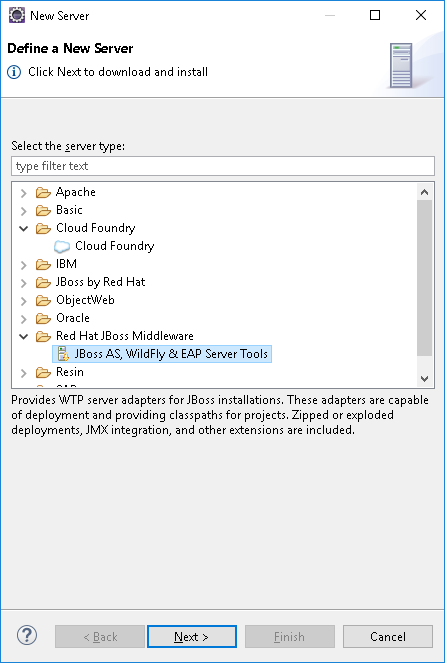
Selecione a opção Servers e clique no botão OK. O Eclipse irá mostrar uma nova aba chamada **Servers**, conforme Figura 03.

Figura 03 – Aba Servers sem nenhum servidor adicionado

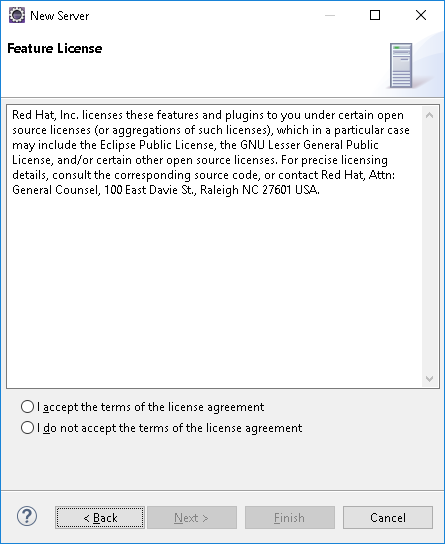
Se você observar nessa aba existe um link “***No servers are available. Click this link to create a new server…***“, indicando que não existe um servidor configurado. Ao clicar neste link será aberta uma dialog **New Server** conforme Figura 04.

 Figura 04 – Dialog New Server

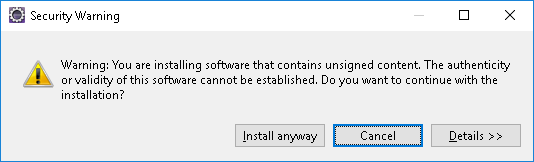
Escolha a pasta **Red Hat JBoss Middleware** e o server **JBoss AS, WildFly & EAP Server Tools**, conforme a Figura 05 e clique no botão **Next >**, para avançar a próxima parte da dialog. Nesse momento o Eclipse irá baixar as informações do repositório onde está a configuração do servidor do JBoss e suas dependências.

 Figura 05 – Escolhendo o server JBoss

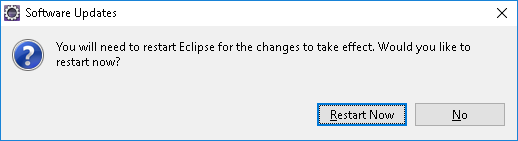
Agora irá aparecer os termos de licença para que o desenvolvedor possa trabalhar com esse Server, conforme Figura 06. Selecione a opção I accept the terms of the license agreement e clique no botão Finish para finalizar a instalação da dependências do JBoss Server.

 Figura 06 – Aceitar os termos de instalação do Server

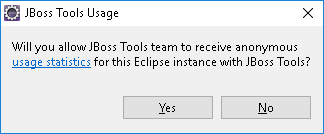
Após o Eclipse baixar as dependências, antes de realizar a instalação, será exibida a mensagem conforme a Figura 07. Ela indica que o software que você está tentando instalar não possui um conteúdo assinado, havendo a possibilidade de causar danos ao computador. Como sabemos que esse é um arquivo da JBoss, então pode clicar no botão **Install anyway**.

 Figura 07 – Aviso de Segurança do Eclipse

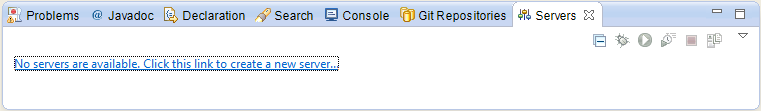
Após alguns minutos de instalação o Eclipse irá mostra a tela da Figura 08, o qual pede para reiniciar o Eclipse. Clique no botão **Restart Now** para que essa reinicialização será feita.

 Figura 08 – Reiniciando o Eclipse

Após reiniciar o Eclipse, aparecerá uma tela indicando se deseja enviar informações de erros e melhorias para a JBoss, Figura 09, aqui é uma escolha sua, não irá influenciar na configuração se escolher Sim ou Não.

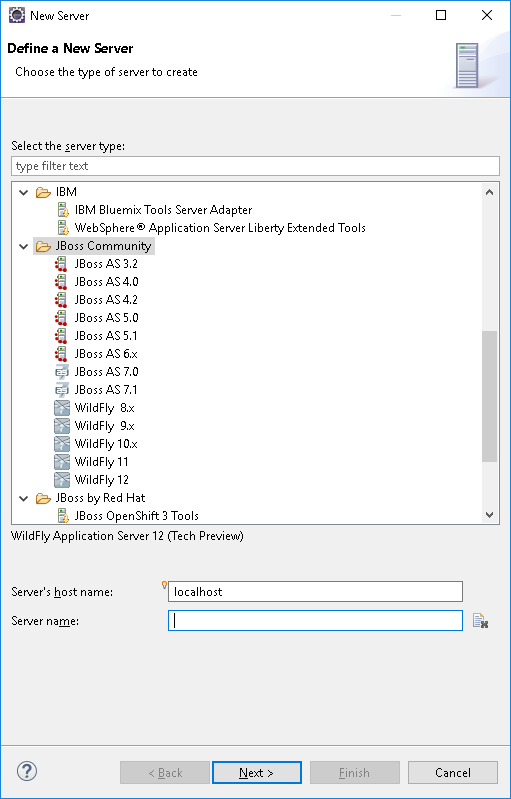
 Figura 09 – Tela de estatísticas do JBoss

 A configuração ainda não terminou, apenas instalamos as dependências. Volte na aba Server e clique link “***No servers are available. Click this link to create a new server…***” conforme Figura 10.

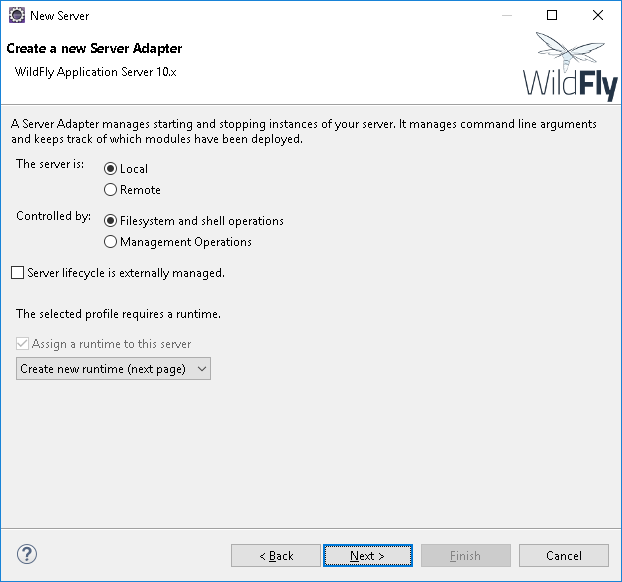
Figura 10 – Aba Servers ainda sem nenhum servidor adicionado

Nesse momento aparecerão várias configurações para os servidores JBoss. Acesse a pasta JBoss Community e escolha o servidor WildFly correspondente a versão que você baixo. Caso não tenha a versão, escolha a maior versão que tiver para o WildFly. Clique no botão Next > Confira a figura 11 abaixo:

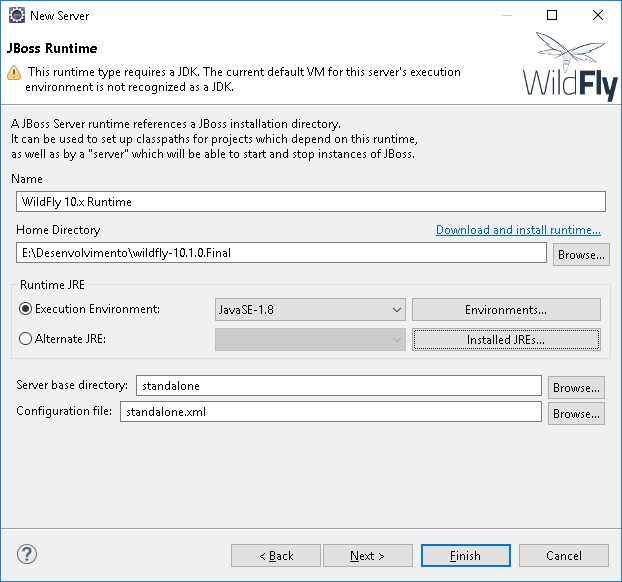
Figura 11 – Várias configurações para os servidores JBoss:



O Eclipse irá apresentar a tela verificando se o servidor é local e como ele é controlado, Figura 12. Deixe as configurações como estão e clique no botão Next >.

 Figura 12 – Outras configurações do Servidor

Agora nesse nova parte da tela carregada, clique no primeiro botão **Browse** e localize a pasta onde foi descompactado o servidor. A sua tela deverá ficar parecida com a da Figura 13.

 Figura 13 – Configurando a pasta local onde está o JBoss descompactado

Onde aparece WildFly o seu devera ser Jboss 7.0 conforme selecionado na figura 11 desse documento e o JAVASE deverá ser 1.6

Caso você não esteja com uma JDK instalada no sistema, aparecerá o aviso “This runtime type requires a JDK”. Para corrigir esse problema é necessário instalar uma JDK na sua máquina e fazer com que o Eclipse aponte para esta nova JDK, o warning irá desaparecer se for realizado isso.

Clique no botão Next> para que você possa adicionar os projetos JSF ao servidor, selecionando o projeto e clicando no botão Add > para que este fique no lado esquerdo da tela, conforme mostra a figura 14.

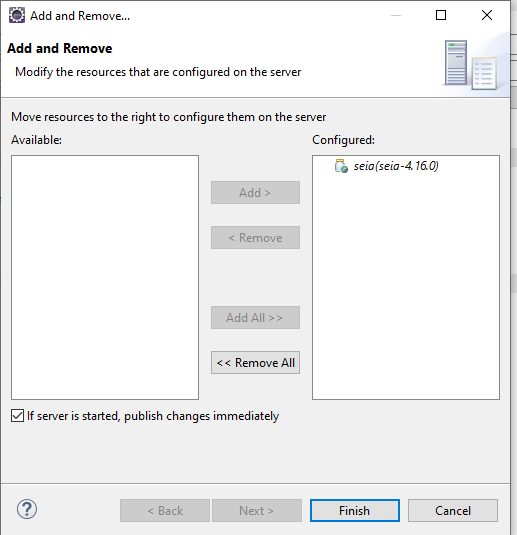


Figura 14 – Adicionando o projeto JSF ao servidor

Clique no botão **Finish** para finalizar a configuração do servidor. O eclipse irá exibir na aba Servers o novo servidor configurado e se expandir as propriedades do Servidor clicando com a Seta, você irá ver que o projeto JSF apareceu dentro do Server, conforme Figura 15.

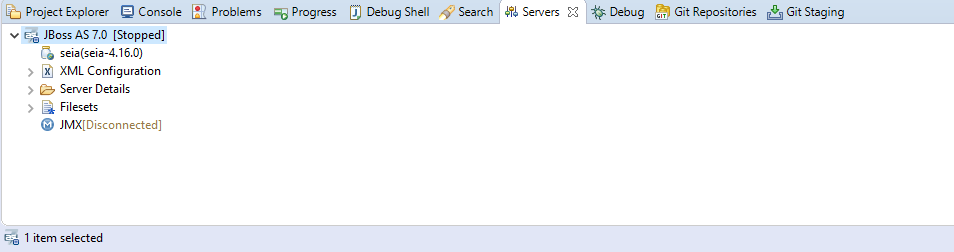


Figura 15 – Servidor configurador e com projeto adicionado

**Iniciando o Servidor**

Para iniciar o servidor basta clicar com o botão direito sobre o Server e ir na opção Start ou Debug (o mais utilizado), conforme Figura 16.

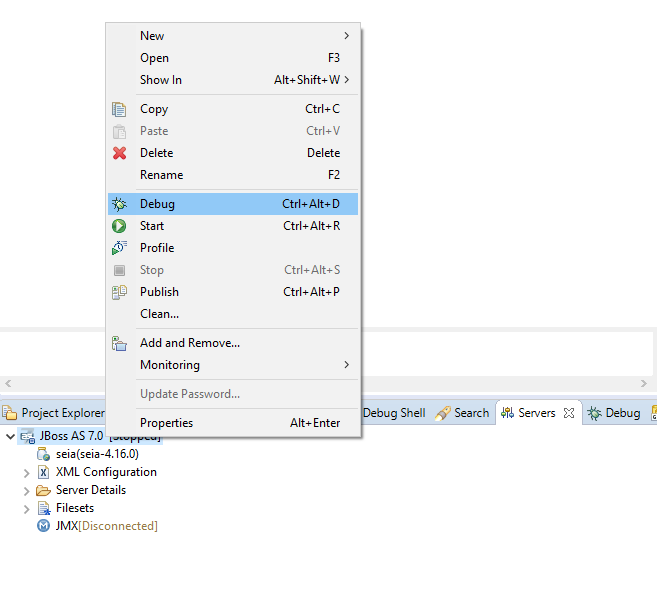


Figura 16 – Opção Debug

O servidor estará inicializado corretamente se aparecer a uma mensagem parecida com essa no console do Eclipse: “JBoss **VersãoJBOSS** iniciado em 33880ms – Iniciado 303 de serviços 340 (os serviços 62 são lazy, passivos ou em demanda)”.

Para descobrir qual é o projeto que o JBoss iniciou basta procurar pela linha:

JBAS018210: Registra o contexto da web: /nomeProjeto

Agora abra um browser e digite a seguinte URL:

**<http://localhost:8080/>**

Se tudo estiver correto irá aparecer a página inicial do sistema conforme Figura 17.

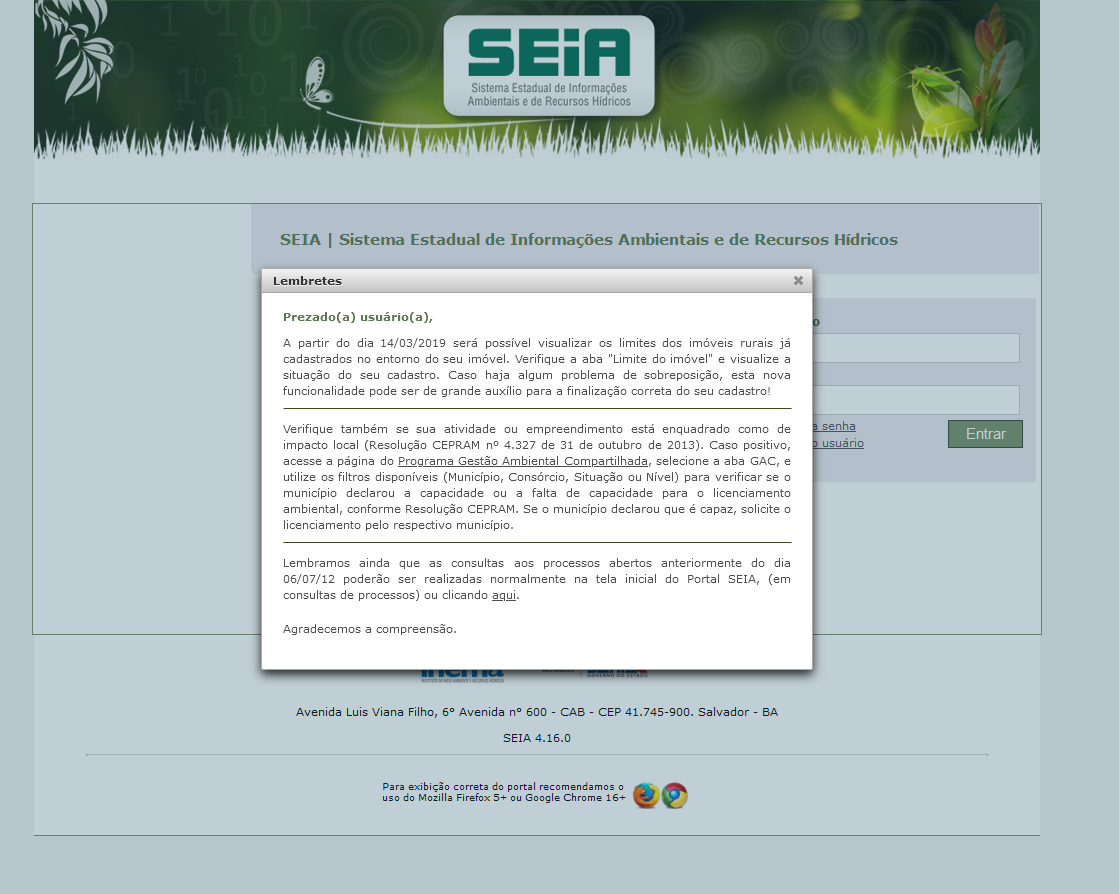
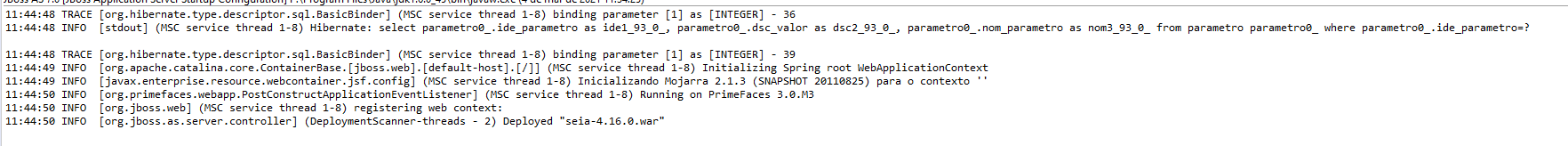


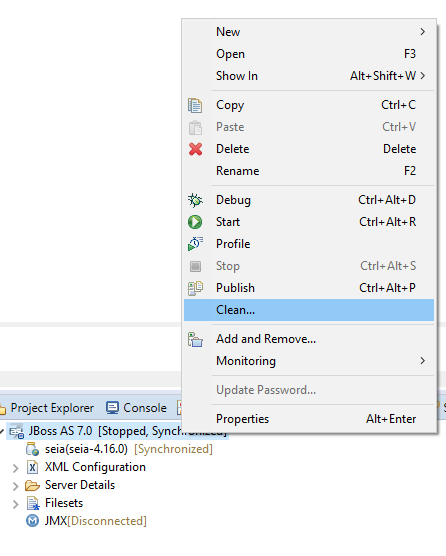
Figura 17 – Servidor inicializado corretamente

Mensagem de Sucesso no console Jboss no Eclipse:

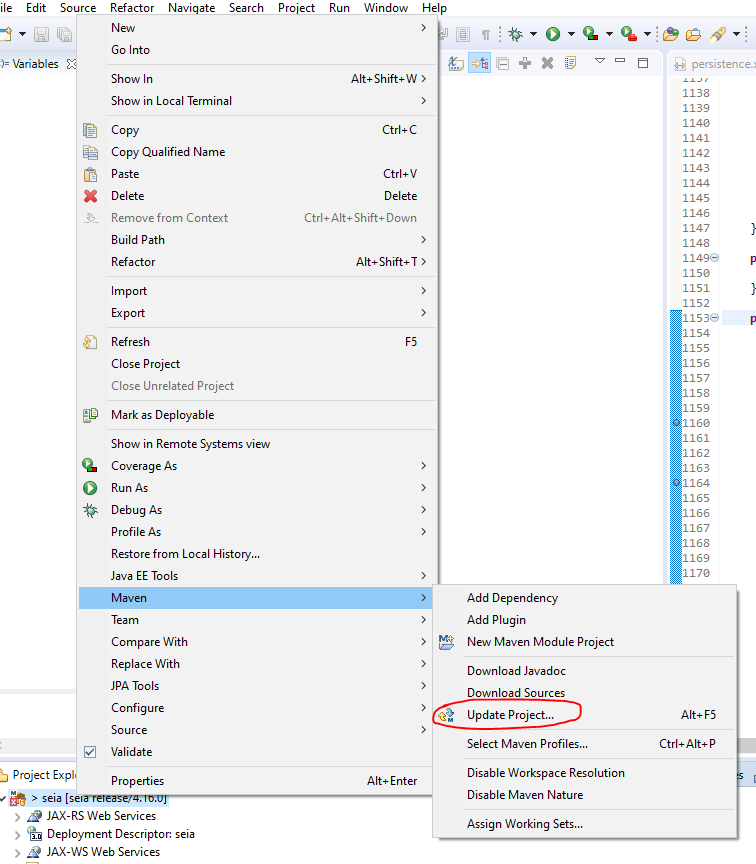


São necessários 10 minutos ou 15 para o sistema subir e aparecer essa mensagem se tudo estiver ok.

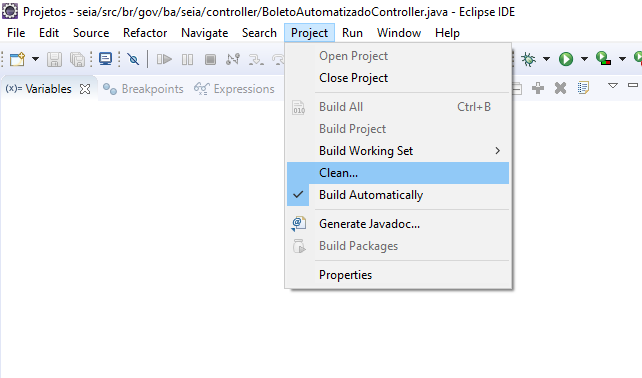
**Recomendações para executar no projeto caso apareçam erros:**



Realize sempre o Clean e espere o status do Jboss para Synchronized.



Selecione sempre o Force Update of Snapshots/Releases



Realize sempre o clean do Projeto quando necessário.

Caso nada tenha dado certo, vá no cmd mvn e realize no projeto o mvn clean install e em seguida mvn clean package -Pprd

E se nada funcionar, com o eclipse fechado faça mvn clean verify e depois mvn eclipse:eclipse

**Programas essenciais para baixar:**

**- Eclipse Jee 2019-03,- Maven 3.6.3,- Git 2.20+,- Postgres 9.5,- Jboss 7 (No Drive [SEIASEMA](#_4.3.2 ACESSO AO DRIVE DOCUMENTAÇÃO))**

**- JDK 1.6.45 e 1.8**

**- php5**

**- Apache2**

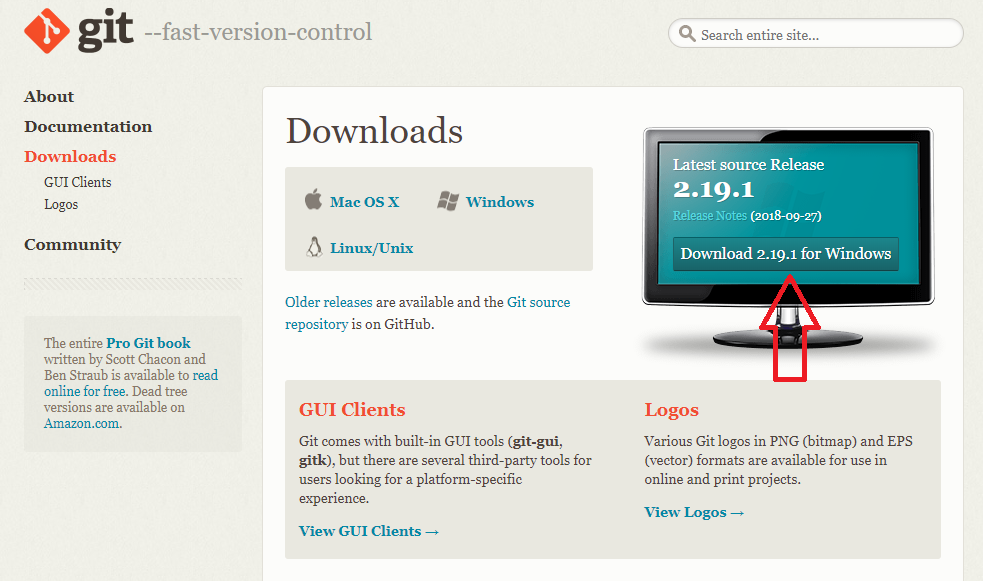
**- iReport 5.0.4**

## 4.8 GIT & GITLAB – CONEXÃO VIA SSH

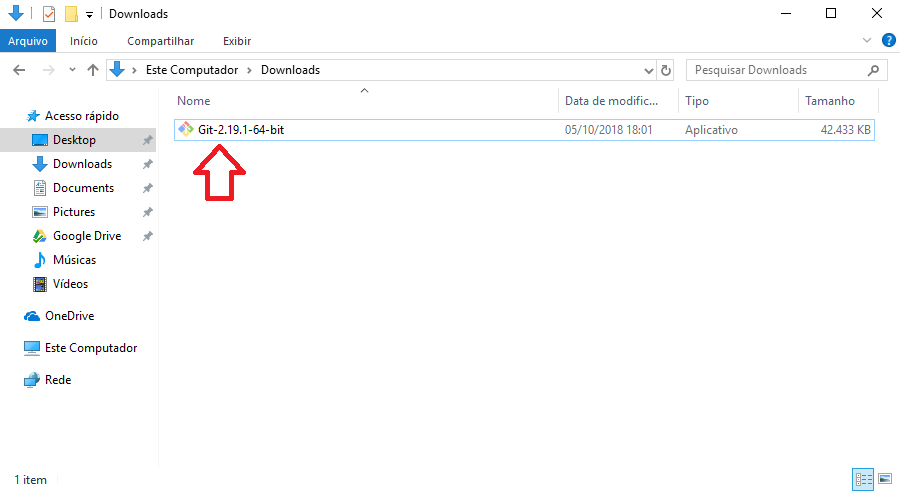
Esse tópico foi criado para auxiliar vocês a configurarem o ambiente corretamente sem dores de cabeça.

**Instalação**

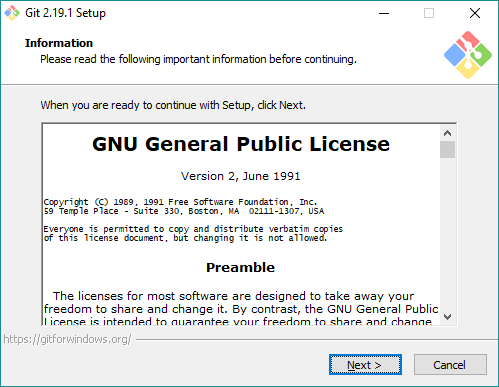
Primeiro, acesse o [site oficial](https://git-scm.com/downloads) para baixar o instalador do git. Escolha o sistema operacional que você pretende instalar (no meu caso é o Windows) e baixe o instalador apropriado.



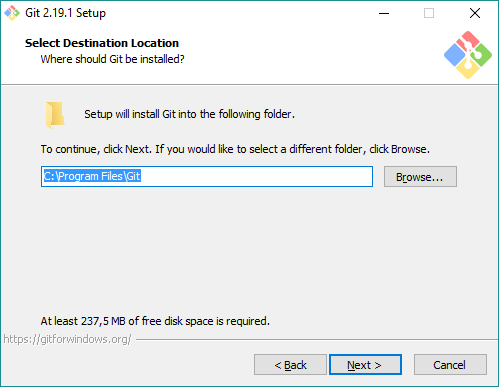
Após o instalador ser baixado, **execute-o**.



A primeira tela do instalador do GIT apresenta os termos da licença de utilização do GIT. Basta clicar em **Next.**



A segunda tela é para a seleção do local da instalação, eu mantenho o padrão C:\Program Files\Git, mas sinta-se livre para alterar. Após definir o local da instalação do GIT, clique em **Next.**

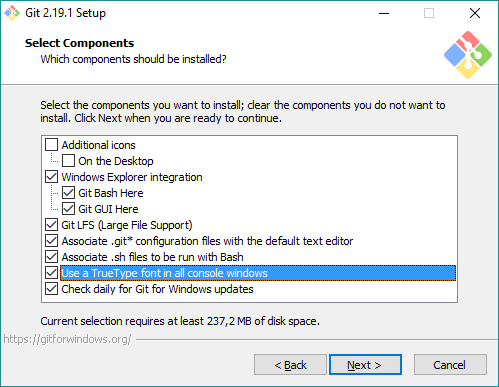


Agora o instalador oferece a possibilidade de selecionar os componentes do GIT que queremos que seja instalado.

**Eu não gosto de adicionar ícones no Desktop, por isso eu desmarco. Os demais componentes eu marquei todos:**

* **Integração com Windows Explorer**
* **Suporte a arquivos grandes**
* **Associação de arquivos, e editor de texto padrão**
* **Permitir que o prompt de comandos seja colorido para as opções do git**
* **Verificação diária de atualização.**

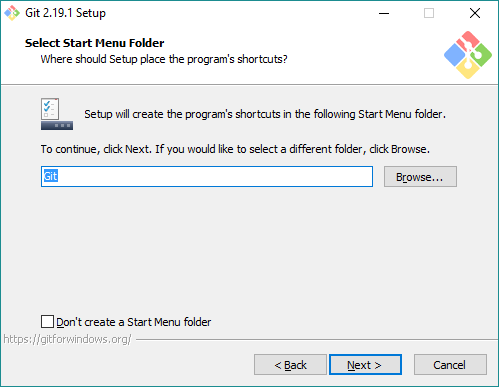
Escolha os componentes de sua preferência e clique em **Next.**



Em seguida o instalador do GIT nos indica que criará uma pasta com atalhos no menu iniciar, se quiser, você pode marcar o checkbox - Don't create a Start Menu folder para o instalador não criar esta pasta.

No meu caso eu deixei o instalador criar os atalhos no menu Iniciar.

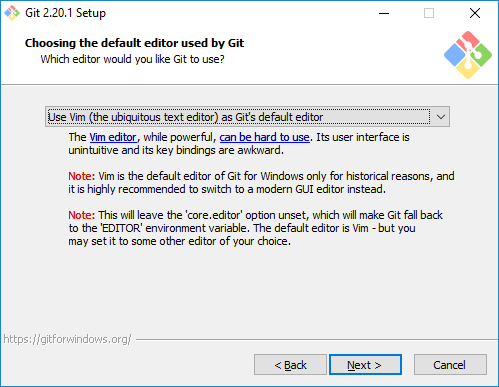
Clique em **Next** para continuar a instalação.



Na sequência o instalador pede para selecionarmos o editor de texto que deve ser utilizado para editar os conflitos que por ventura acontecerem.

A maioria gosta do Vim mesmo, mas você pode não tem muita afinidade com o Vim você pode alterar para [Notepad++,](https://notepad-plus-plus.org/downloads/) [Sublime](https://www.sublimetext.com/3), [Atom](https://atom.io/), [VS Code](https://code.visualstudio.com/), ou outro editor de sua preferência.

Após a escolha, clique em **Next** para continuar com a instalação.

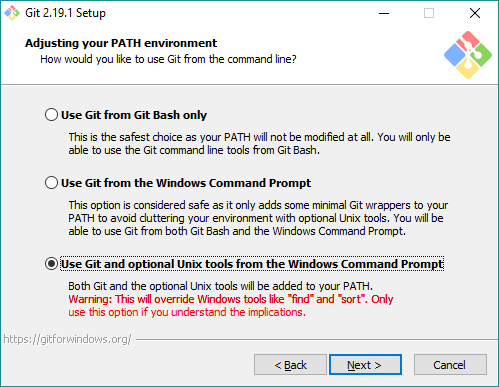


Neste momento começa algumas partes confusas para quem não conhece muito o GIT.

Na próxima tela o instalador do GIT nos pergunta se queremos **usar os comandos do git:**

* somente no prompt de comandos do próprio git (chamado de Git Bash), neste caso ele não vai alterar a variável de ambiente PATH.
* no prompt do Windows (Windows Command Prompt), neste caso a variável de ambiente PATH será alterada para incluir o caminho de onde está o executável git.exe.
* no prompt do Windows + comandos utilitários do línux. Eu sempre marco esta opção, porque o instalador traz para o Windows alguns comandos que eu uso muito no Linux, como cat, ls, find, etc. Neste caso a variável de ambiente PATH será alterada para incluir o caminho do executável git.exe e dos executáveis de cada comando utilitário do linux.

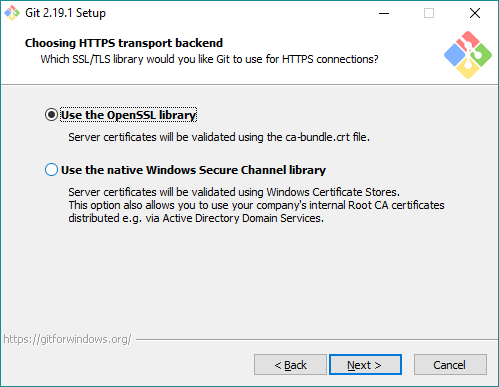
Escolha a opção que acha mais adequada para você e clique em **Next.**



A tela seguinte do instalador oferece a opção de escolhar a biblioteca de validação de chaves de segurança SSL.

É recomendável utilizar a OpenSSL, que é compatível com outras plataformas.

Escolha a que achar apropriada e clique em **Next.**



Neste ponto da instalação, nos é perguntado como o git deve tratar o final dos arquivos de texto.

Este é um assunto complicado de explicar em poucas palavras, mas vamos lá ...

Vamos dar uma pequena pausa na instalação do GIT aqui para entender esse assunto.

**Como é tratado o final de linha dos arquivos de textos**

O Windows e o Unix, tratam o final de linha dos arquivos texto de formas diferentes.

Pra resumir a história ... o Windows segue um padrão antigo de comandos de impressora, chamado de CRLF, que indica para o cabeçote da impressora ir para o início horizontal (Carriage Return) e para iniciar uma próxima linha (Line Feed).

Como se fosse aquela alavanca da máquina de escrever que funciona dessa forma, vai pro início e para a próxima linha.

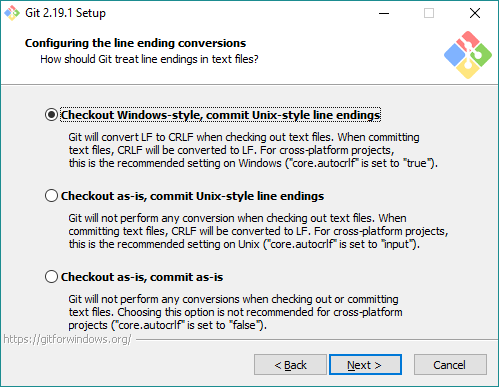
Já no linux, ficou comum somente o comando LF que teria o mesmo funcionamento do CRLF, ou seja, próxima linha...início.

Isso significa que o Linux usa um caracter ASCII para a quebra de linha e o Windows usa dois. Esta é a mesma origem da confusão entre \n e \r, mas isso é assunto pra outro post.

Por agora entenda as opções que o instalador do GIT oferece:

* converter LF para CRLF ao baixar arquivos e CRLF para LF ao comitar.
* baixar como é (não converte nada), mas comitar convertendo CRLF para LF.
* não converter nada, baixar como é e comitar como estiver.

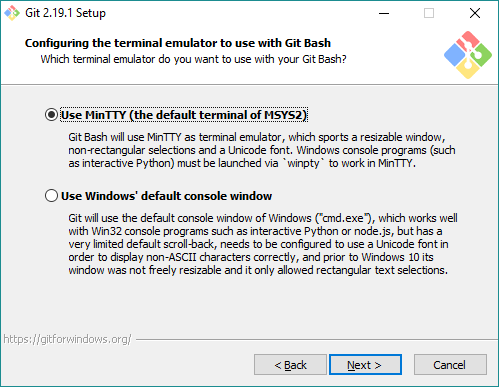
Eu marco a primeira opção, para evitar problemas de compatibilidade entre as plataformas.



**Continuando a instalação ...**

Na sequência, o instalador oferece a opção de escolher o emulador de terminal (prompt) que queremos usar para o GIT. Podemos usar o console padrão do Windows (cmd.exe) ou o MinTTY que é um emulador que oferece algumas características melhores na minha opinião.

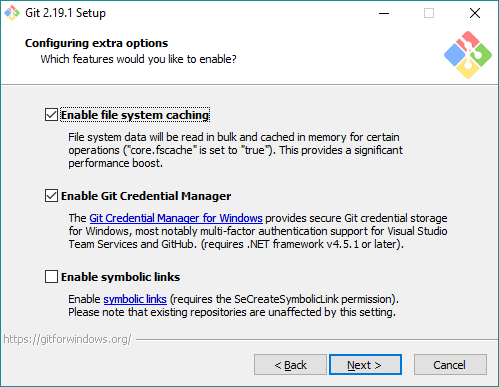
Em suma, o cmd é usado para compatibilidade com plataformas de 32 bits, ou seja, se o seu windows for 32 bits, é melhor escolher o cmd mesmo, mas se o seu windows for de 64 bits, é melhor escolher o MinTTY, pq ele oferece alguns recursos melhores para terminal, como por exemplo maximizar.



A penúltima tela nos oferece algumas opções extras, são elas:

* habilitar cache de arquivos na memória. Isso melhora o desempenho do git em alguns casos.
* habilitar o gerenciador de credenciais do GIT. Isso permite autenticação em duas etapas no VSTS e no GitHub, e precisa do framework .NET 4.5 para funcionar.
* habilitar links simbólicos: Sinceramente, não sei o que é isso no Windows e não me fez falta ainda, se você souber do que se trata, apreciarei muito uma explicação aí nos comentários. ;)

Selecione os itens opcionais que você deseja e clique em **Next**.



Por fim, última tela do instalador do GIT no Windows (UFA!!!), o instalador nos oferece alguns componentes que estão em fase de experimentação.

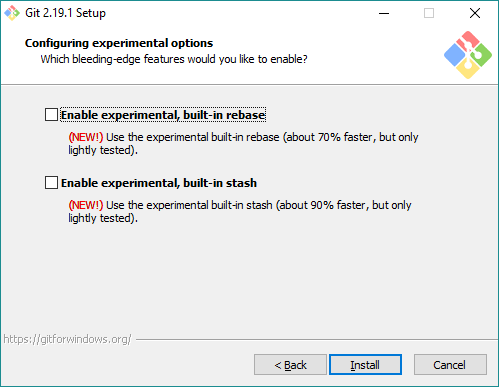
São componentes que não foram muito testados, mas que parecem melhorar bastante a performance.

Você escolhe se que instala ou não ...

Eu não marquei.

Após decidir se instala os componentes experimentais ou não, clique em **Install** para começar a instalação.

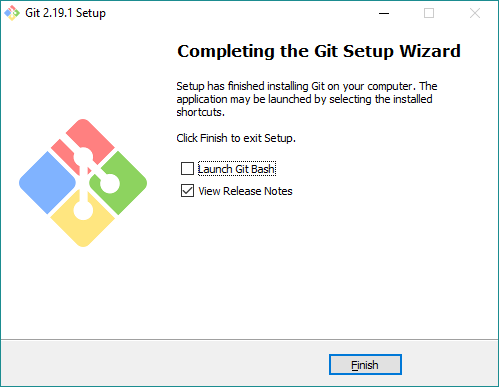
Tela de seleção de componentes experimentais do GIT.



Depois da instalação, o instalador oferece a opção de já rodar o prompt do GIT (Git Bash) e ver o arquivo de texto com as notas da versão lançada.

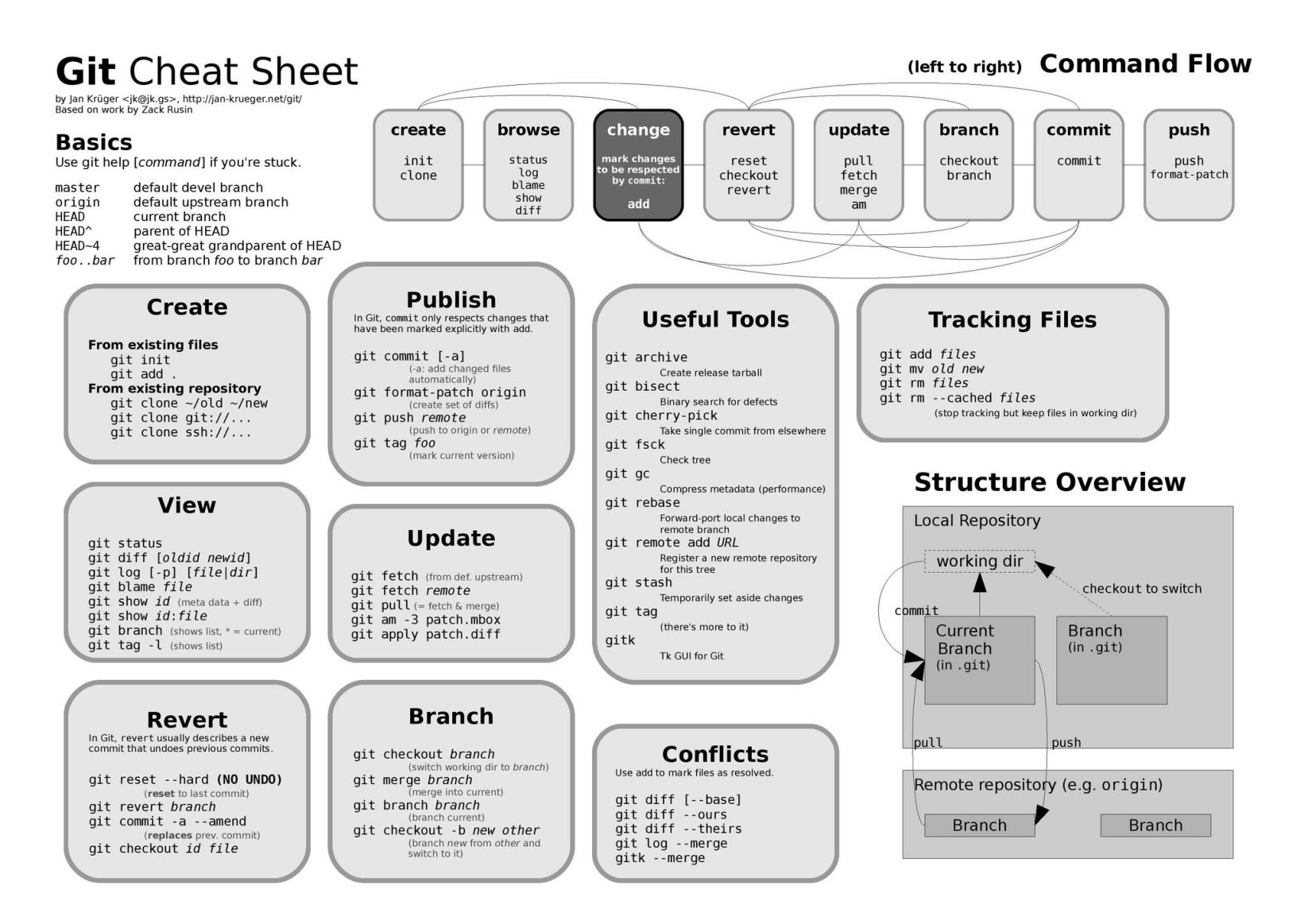
Se quiser uma dessas duas opções, marque-a, eu desmarquei as duas.

Para finalizar, clique em **Finish**.



Pronto, agora você tem o GIT instalado no seu Windows!

### 4.8.1 UTILIDADES COMANDOS GIT



### 4.8.2 GITLAB – CONEXÃO VIA SSH

1 - Checar nome e e-mail na configuração do git no pc

Para assinar a autoria dos commits, precisamos definir o nome e e-mail que utilizaremos para acessar o gitlab.

1.1 - Executar o comando para exibir o nome:

$ git config --global user.name

Se não exibiu o seu nome, então **insira**:

$ git config --global user.name "Exemplo Silva"

1.2 - Mesma coisa para o e-mail.

Checar e-mail:

$ git config --global user.email

Se retornou o e-mail pessoal ou nenhum, **insira-o**:

$ git config --global user.email "exemplo@prodeb.ba.gov.br"

2 - Gerar uma nova chave SSH

Entre na pasta ~/.ssh/ e execute o comando:

$ ssh-keygen -t rsa -C "exemplo@prodeb.ba.gov.br"

**\*Mantenha o nome da chave apenas clique enter após executar o passo anterior e digite uma senha\***

**O comando anterior irá gerar duas chaves: uma privada, outra pública.**

Exiba o conteúdo da chave pública:

$ cat id\_rsa.pub

A saída do comando acima deve retornar algo próximo a isso:

ssh-rsa blablabla...@somospi.com.br

**Copie o conteúdo referente a sua chave pública desde ssh-... até .com.br**

3 - Colar o conteúdo da chave pública no [Gitlab](http://gitlab.prodeb.ba.gov.br/base/sema/seia):

3.1 - Logado na sua conta, vá até Settings/SSH Keys.

3.2 - No campo Key, cole o conteúdo da chave pública.

3.3 - No campo Title, prefira digitar o nome do seu pc para facilitar a identificação.

Pode deixar o campo **Expires at** sem preencher, assim a chave nunca vai expirar.

4 - Testar o funcionamento

**Por garantia, é interessante limpar o cache das chaves SSH:**

$ ssh-add -D

Comando para testar a nova chave:

$ ssh -T gitlab.prodeb.ba.gov.br

O resultado deve ser:

Welcome to GitLab, @exemplo!

5 - Clonar os repositórios via SSH

Agora é possível clonar as URLs de SSH. Não esqueça de sair do diretório ~/.ssh/.

Ex:

$ git clone git@gitlab.prodeb.ba.gov.br:base/sema/seia.git

6 - Atualizar os repositórios que foram clonados via HTTPS

Caso já tenha clonado algum repositório via HTTPS, na pasta do mesmo, execute o comando abaixo para atualizar a url do servidor remoto:

Ex:

git remote set-url origin url-ssh-do-repo

**Configurando duas contas GitLab num mesmo computador**

Você efetua o **clone** em diretórios diferentes e ao fazer o push em cada diretório você estará utilizando uma conta diferente.

Para isso você deve criar uma nova chave **ssh.**

**ssh-keygen -t rsa -C "email@segundaconta.com"**

Ao executar esse comando será exibido algumas mensagens e será solicitado que você selecione o diretório que será gerado sua chave **ssh: Enter file in which to save the key (/Users/SeuUsuario/.ssh/id\_rsa).**

Nesse momento você deve especificar um novo nome para o arquivo (do contrário será substituído o já existente), você pode especificar algo assim: **/Users/SeusUsuario/.ssh/id\_rsa\_segundousuario.**

Faz o processo normal para vincular a chave pública dessa chave **ssh** com o segundo usuário.

Dessa forma agora você tem duas chaves **ssh** mas precisa configurar quando deve ser usada cada uma delas.

Crie um arquivo com o nome **config** dentro do diretório **~/.ssh** com um conteúdo mais ou menos assim.

#Default Gitlab Account

Host gitlab-prodeb HostName gitlab.prodeb.ba.gov.br User git IdentityFile ~/.ssh/id\_rsa

#Second Gitlab Acount

Host gitlab-infojr HostName gitlab.com User git IdentityFile ~/.ssh/id\_rsa\_infojr~/.ssh/id\_rsa\_segundousuario

Agora imagine que temos o seguinte repositório para clonarmos **git@gitlab.prodeb.ba.gov.br:base/sema/seia.git**

Posso usar um dos dois **Host** que tem no arquivo **~/.ssh/config.**

Ao usar git **clone git@github.com:SeuUsuario/teste.git** estarei usando minha conta padrão.

Para usar a segunda conta basta mudar o host do repositório: **git clone git@github-second:SeuUsuario/teste.git.**

No diretório do segundo usuário talvez seja necessário configurar **user.name** e **user.email**.

git config --add user.name "Segundo Usuario"

git config --add user.email email@segundaconta.com

# Adicionais que você precisa saber sobre a SEMA/SEIA

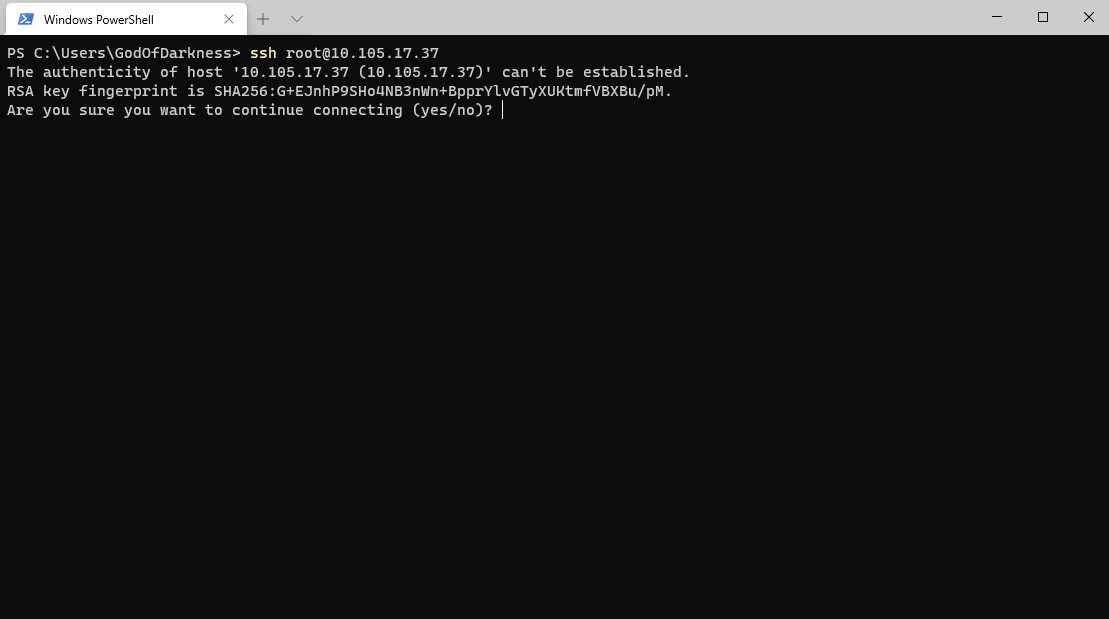
* **Conexão SSH**

Acessar o gitbash e digitar o seguinte comando:

ssh root@10.105.17.37

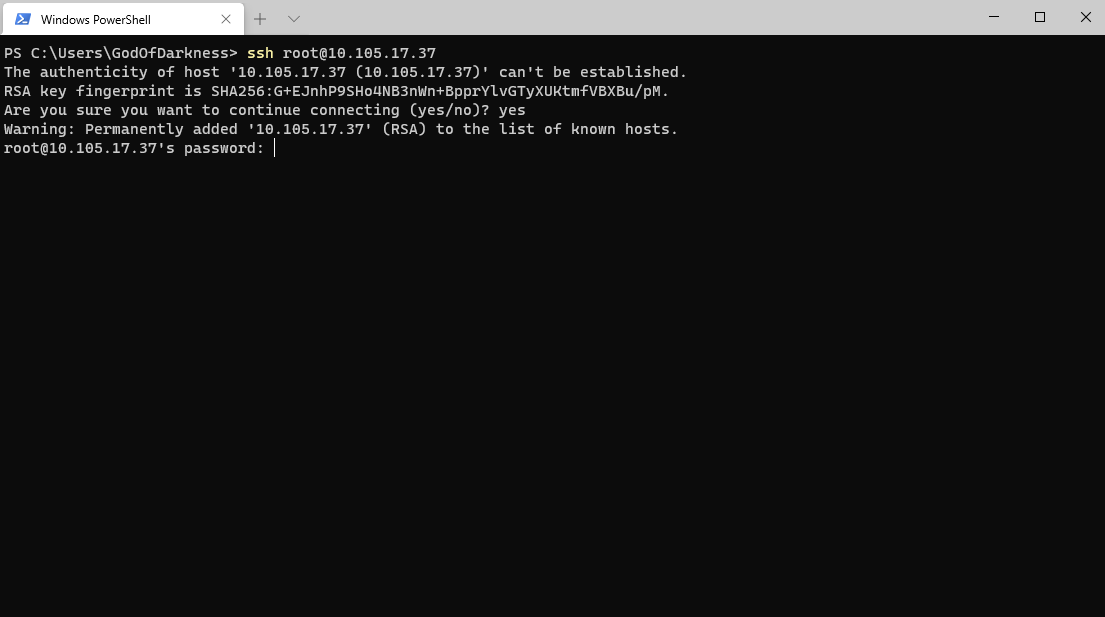
onde o ip 10.105.17.37 deve ser substituído pelo ambiente desejado, caso seja o ambiente 76 é 10.90.2.76 e caso seja o ambiente 22 é 10.90.2.21

assim que você logar aparecerá essa imagem:

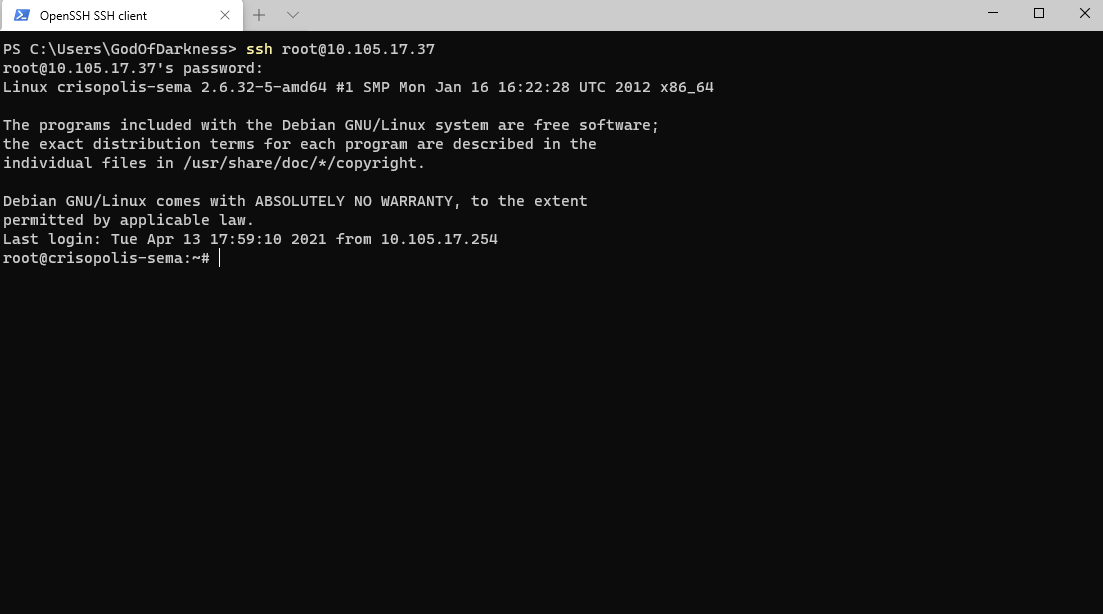


Aqui você deverá escrever yes para ter o acesso ao servidor

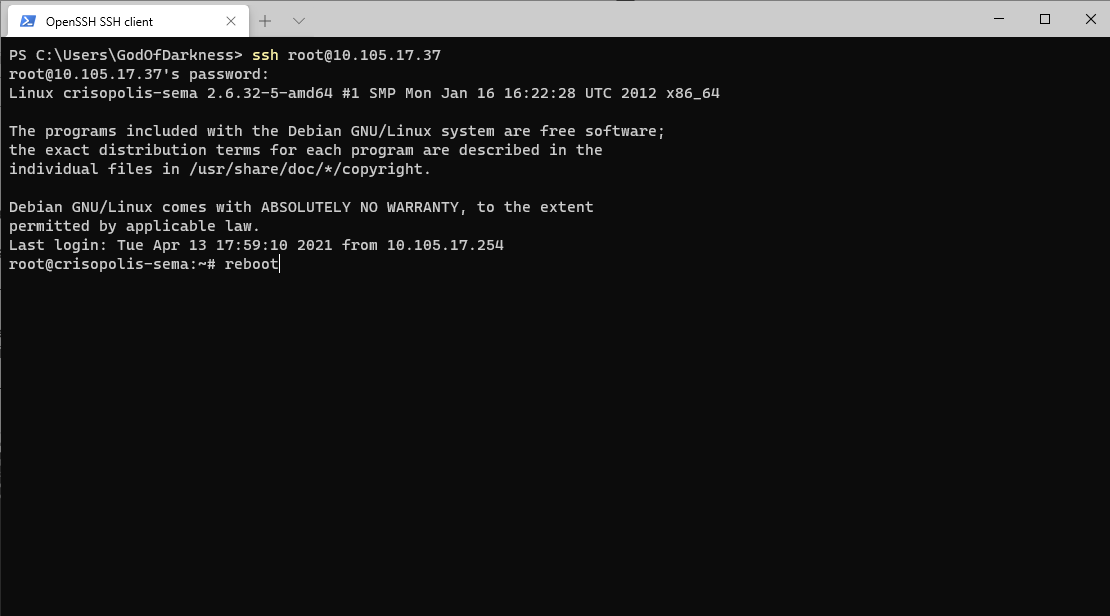
Note que aqui ele dá uma mensagem de warning que desconhece a sua máquina porém nesse primeiro instante vamos ignorar e digitar a senha: sema@seia que é a mesma senha para o ftp.



Lembre-se a conexão ssh tem timeout caso demore para digitar a senha é necessário refazer todos os passos anteriores. Caso tenha dificuldade para enxergar , utilize o zoom do documento.



Após entrar no servidor caso deseje reiniciar, digite o comando reboot e aguarde. Pronto você está conectado ao servidor e caso faça algo de errado sua chave está registrada =) -> utilize com sabedoria.



Problemas que acontecem no ssh

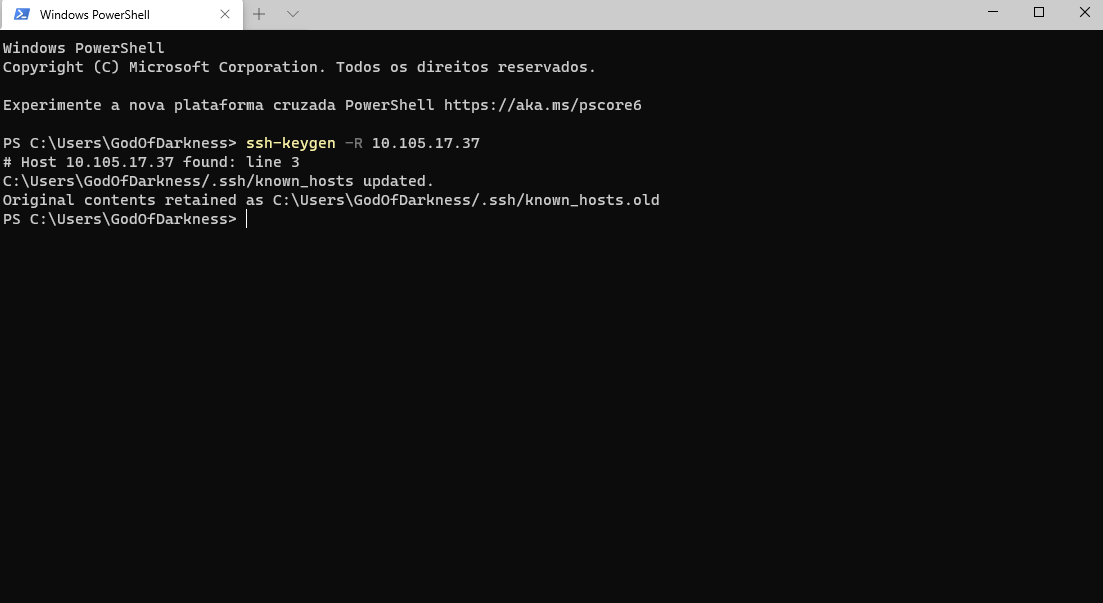
Caso tente logar novamente, reiniciou o modem, o computador é necessário dar o seguinte comando:

ssh-keygen -R 10.105.17.37

-Onde o 10.105.17.37 é o ip desejado no caso é o ambiente 65.

Aparecerá a seguinte mensagem:

Aparecerá a mensagem known\_hosts updated e você conseguirá acessar o servidor novamente retornando ao inicio do tutorial do ssh conforme a mensagem abaixo:



# GERAÇÃO DE VERSÃO

* **Quarta – Feira**

Às quartas devemos gerar a versão de produção e treinamento sempre às 16h

***Obs: Certifique-se de realizar os passos abaixo dentro da VPN, caso não esteja conectado os passos não vão funcionar corretamente.***

1. Entrar no gitbash
2. Na versão que você está atuando digite o git pull
3. Digite: git status
4. Git add . para adicionar todos os arquivos que estão alterados ou selecione arquivo por arquivo e digite: git add nomedoarquivo
5. Git commit -m “comentário geralmente relacionado ao ticket” ex: “resolução do ticket #45186”
6. Commita qualquer alteração que tenha pendente no seu branch.

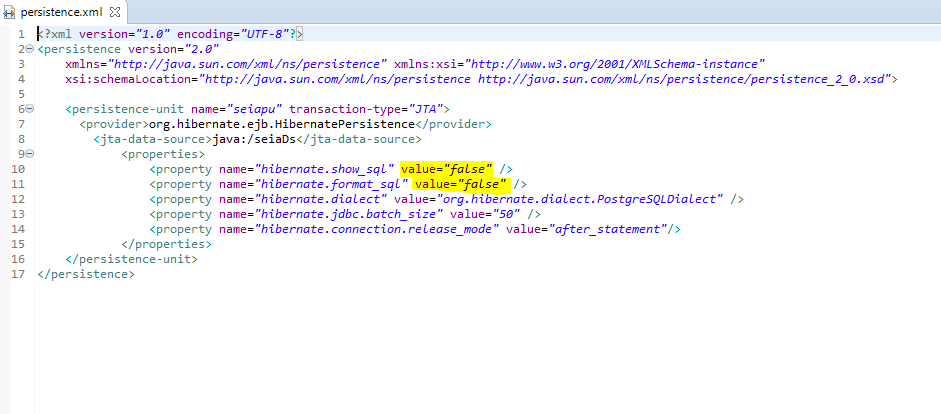
**Boas práticas**: Costumamos criar no nome da branch que estamos trabalhando assim: /feature/onomedoticketquevoceestaatuando ex: feature/15874 e vamos para o próximo passo.

1. Atalho útil para o eclipse para pesquisar arquivos -> Ctrl + Shift + R
2. Verifique o **persistence.xml** no eclipse ele deverá estar conforme a imagem abaixo:

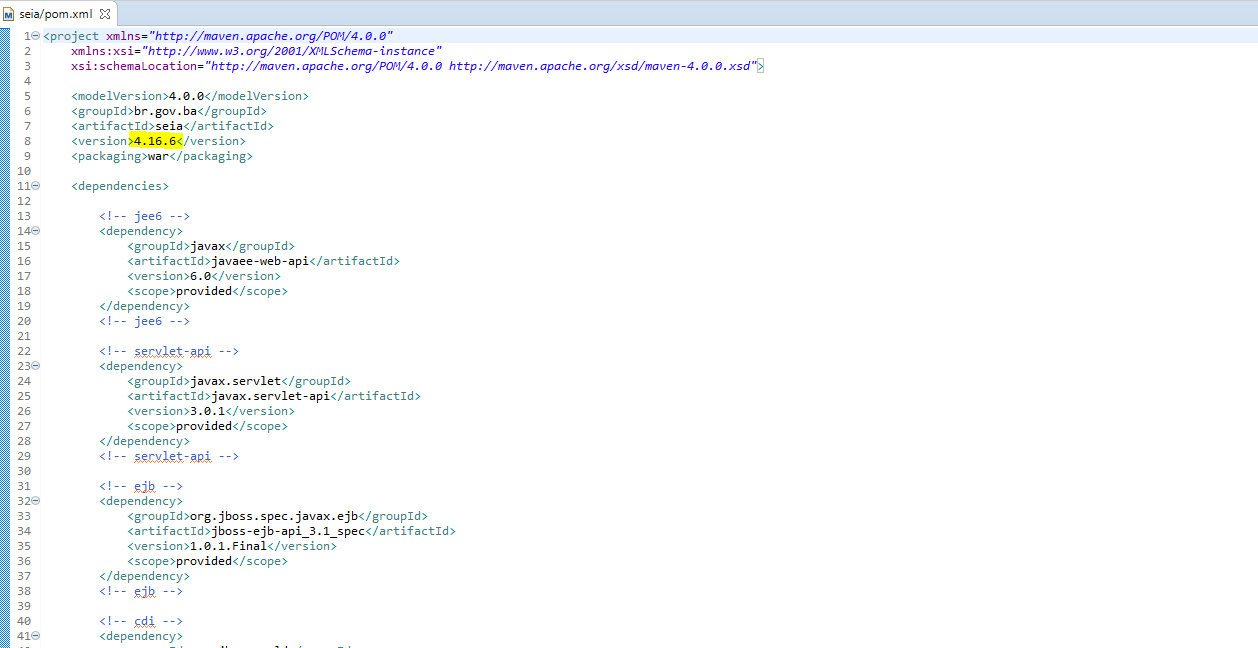
Os valores deverão permanecer assim enquanto geramos a versão:

show.sql = false

format.sql = false



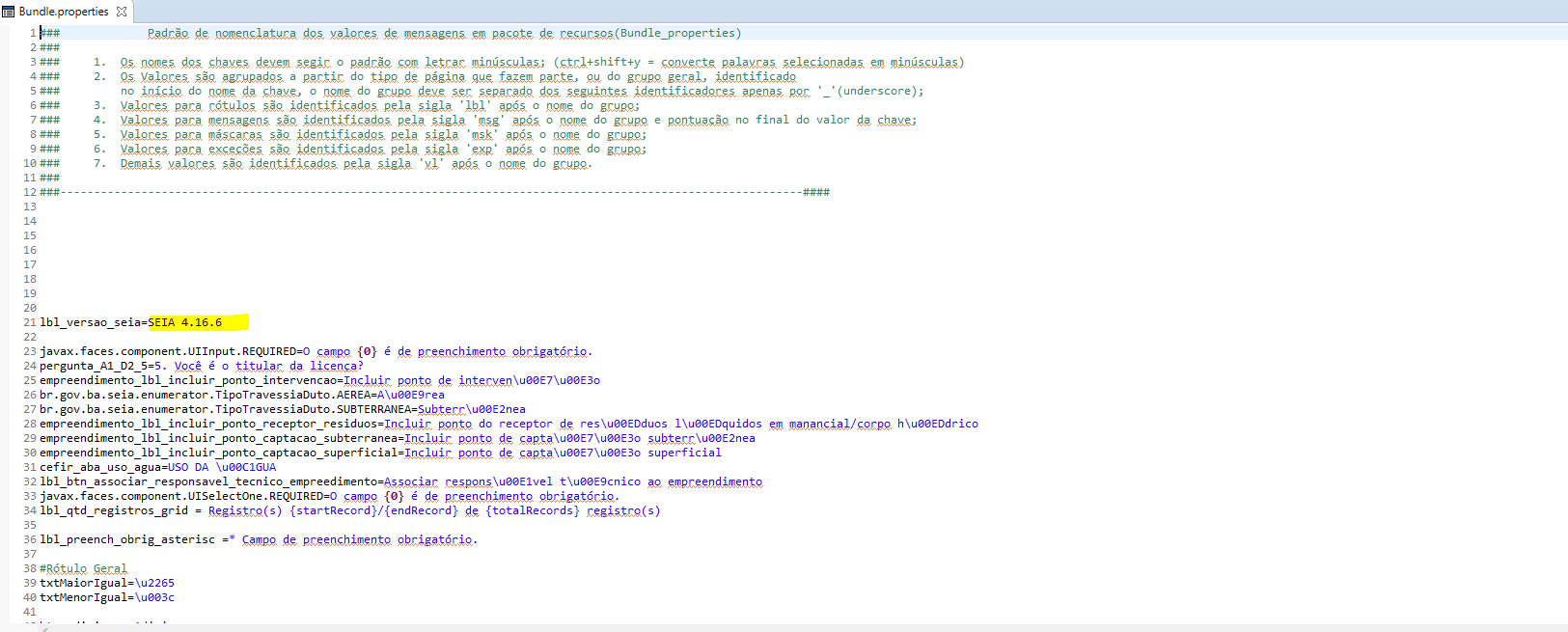
1. Em seguida verifique também o **pom.xml** e deverá estar conforme a imagem:

Devemos modificar aqui nessa tag: <version>4.16.6</version>

Obs: Note que na imagem do passo 8 está destacada a versão que se encontra o pom aqui você deverá modificar a versão conforme o time de sustentação.

1. Em seguida verifique o arquivo **bundle.properties** ele deve está na mesma versão que o pom.xml conforme a imagem abaixo:

Devemos modificar nesse trecho: lbl\_versao\_seia=SEIA 4.16.6



1. No gitbash devemos alterar a Branch com o comando: git checkout develop
2. Devemos realizar o merge da versão que estamos com o branch develop

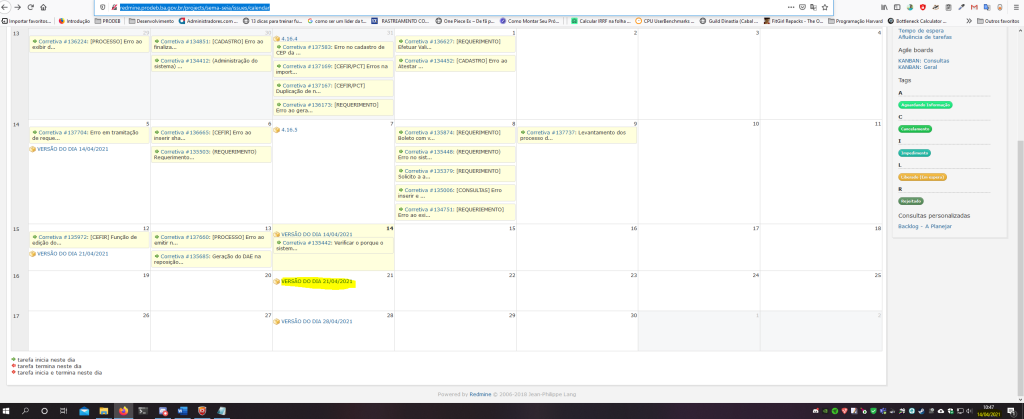
Ex: Estamos na versão 4.16.6 a branch é identificada como release/4.16.6 e devemos realizar um merge na branch develop logo o comando é: **git merge release/4.16.6**

1. Realize o push na develop com o comando:

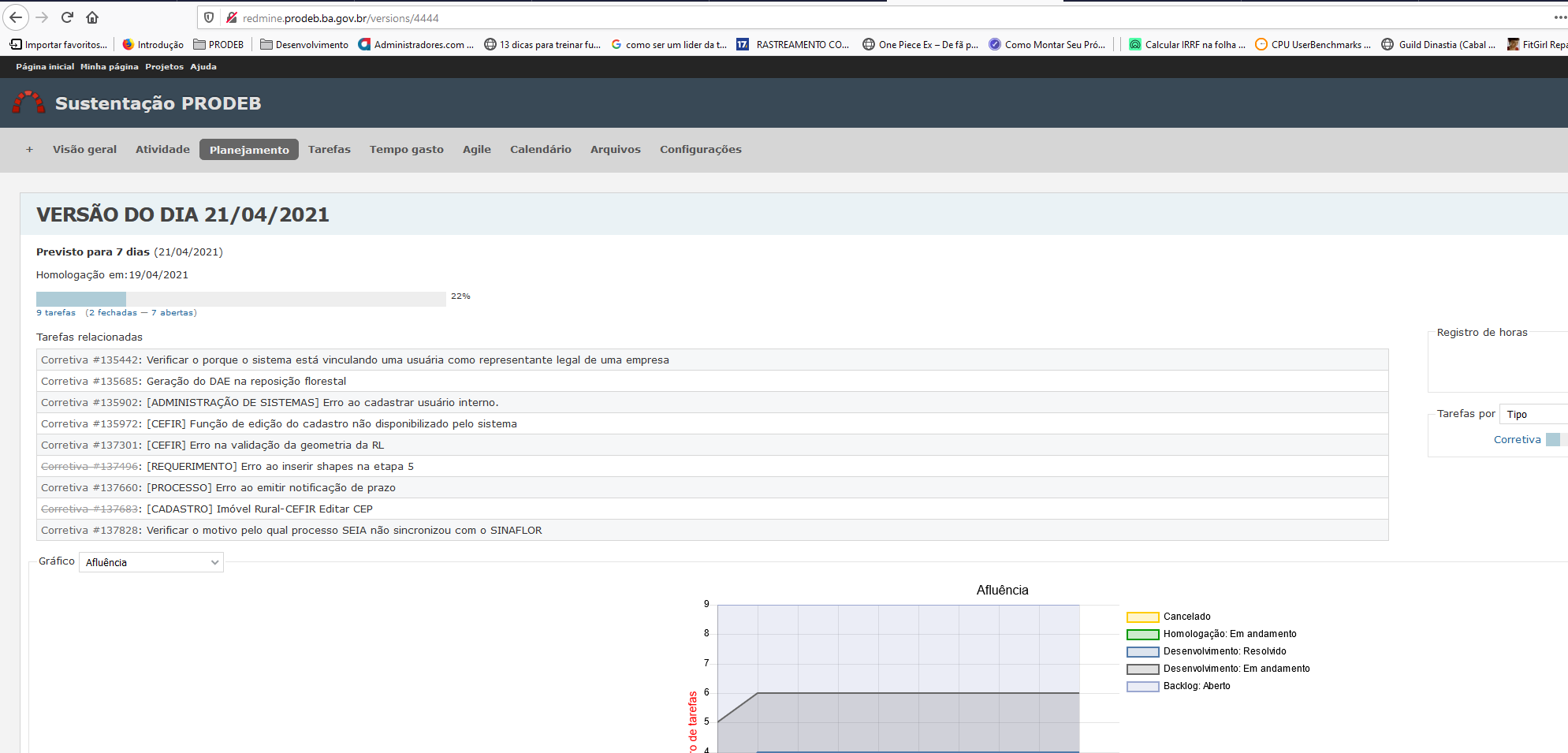
**git push origin develop**

1. Agora vamos criar **a nova branch da próxima versão**, mas antes devemos **verificar os seguintes critérios:**

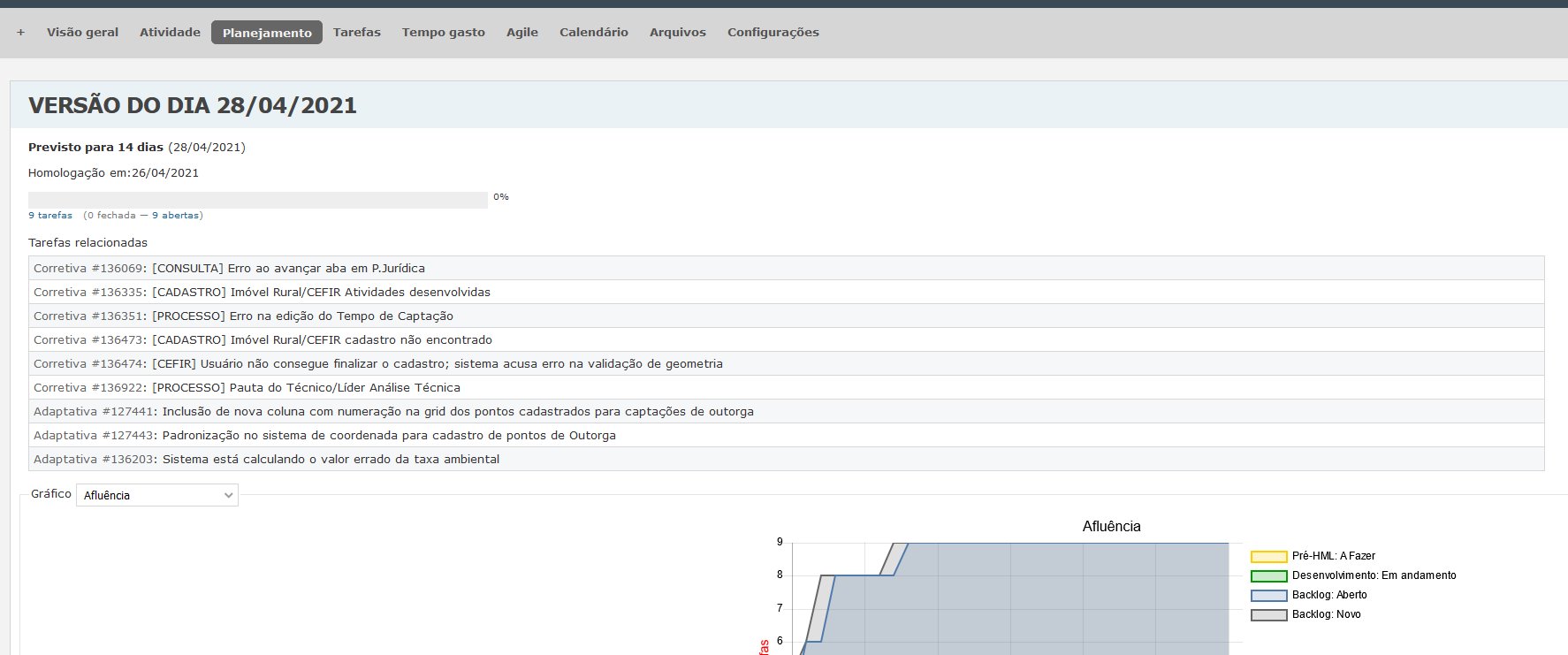
- Verifique o link: <http://redmine.prodeb.ba.gov.br/projects/sema-seia/issues/calendar> e selecione conforma a imagem abaixo a próxima versão:



Verifique que estamos no dia **14/04/2021** conforme a área de trabalho e vamos para a versão do dia **21/04/2021**, clique em cima do link marcado na imagem, você verá os tickets da **próxima versão assim**:



Verifique que há apenas corretivas, logo a próxima versão do branch a ser criada é **4.16.7**, caso houvesse adaptativa não devemos incrementar o último número e sim pular para a próxima sequência: **4.17.0** (como é no caso da versão do dia 24/04/2021 conforme a *imagem abaixo*, **por fins de curiosidade**.



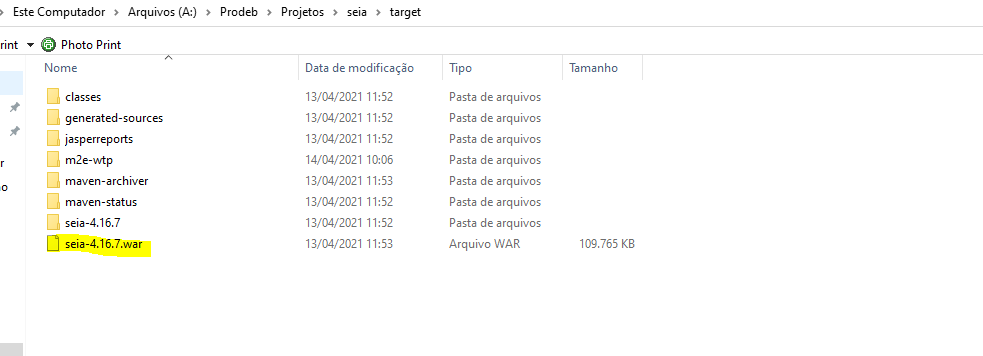
1. Git checkout -b release/4.16.7
2. Certifique que os arquivos **persistence.xml**, **pom.xml** e **bundle.properties** estejam devidamente alterados conforme os passos anteriores **modificando a versão.**
3. Após a verificação de **sua responsabilidade** vamos realizar o push da nova branch com o comando:

**git push --set-upstream origin release/4.16.7**

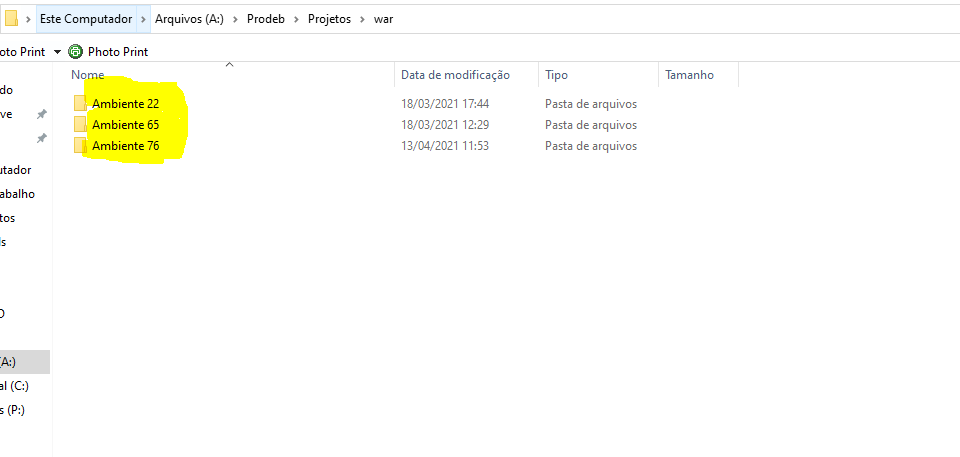
1. Feito isso pode fechar o eclipse e executar o seguinte comando na pasta do seia para gerarmos a versão:

**mvn clean package -Phml**

1. Após executar esse comando vá na pasta target do projeto:

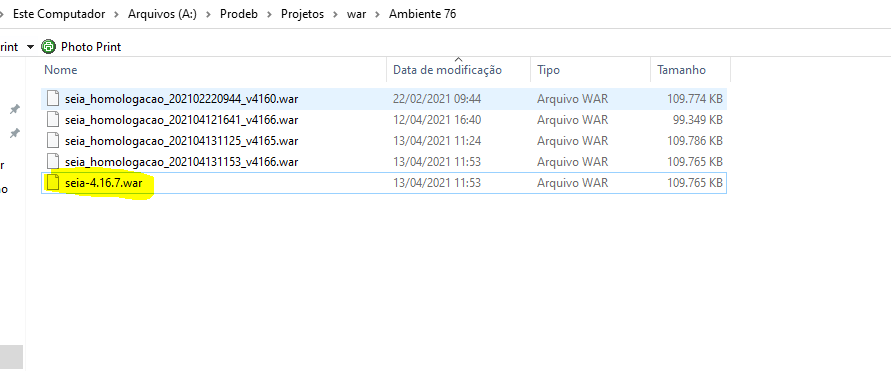


Copie o arquivo selecionado **seia-4.16.7.war**, por recomendação costumamos criar uma pasta fora do projeto seia chamado war e costumamos separar criando mais 3 pastas por ambiente: 22, 65(famoso 37 atualmente) e 76 conforme a imagem abaixo:



1. Note que estamos gerando versão na quarta feira que é produção e treinamento relembrando: **22(treinamento), 65(homologação) e 76(produção)**

Vamos colar o arquivo copiado anteriormente na pasta de produção

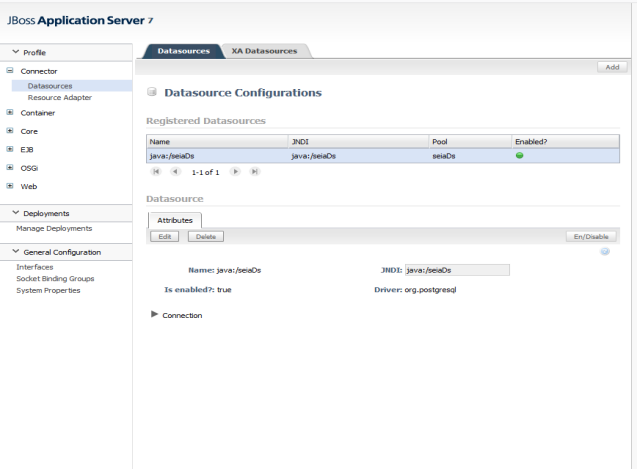


Note que há uma padronização dos arquivos da seguinte maneira:

**seia\_homologacao\_anomesdiahhmm\_v4167.war**

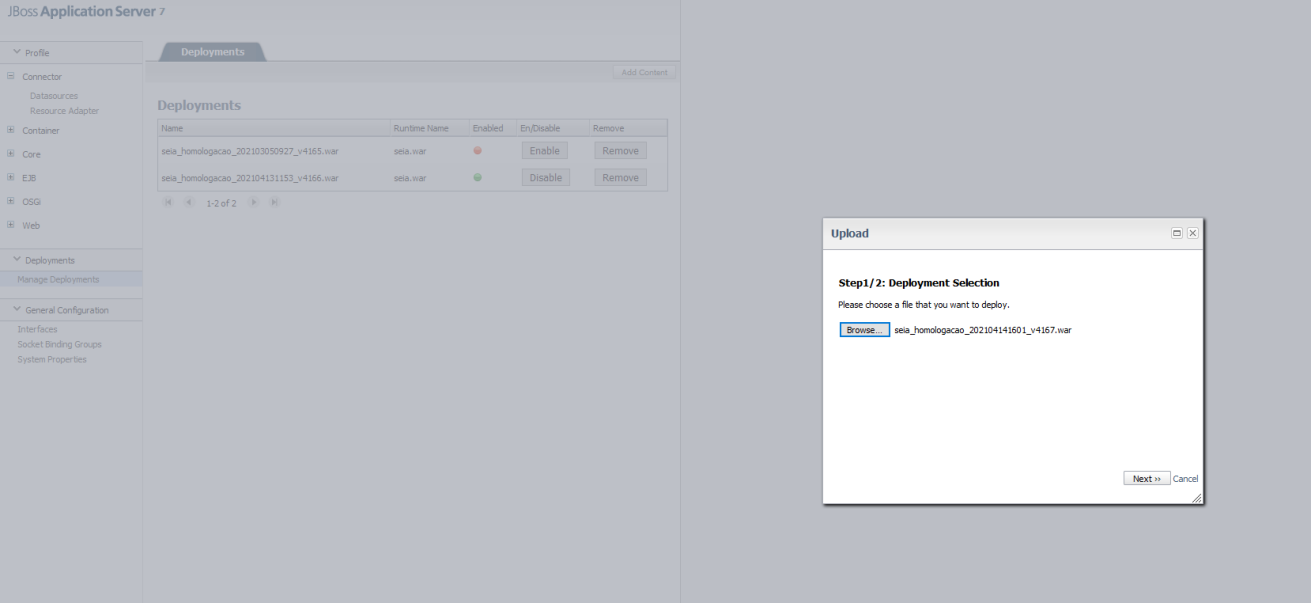
Renomeie o arquivo seia-4.16.7.war para: seia\_homologacao\_202104141601.war lembre-se que é apenas às 16h. (**Desconsidere 1 min de atraso** )

1. Acesse o endereço de produção 10.90.2.76:9990
2. Senha do acesso: admin e senha: admin
3. Você verá essa tela do jboss:

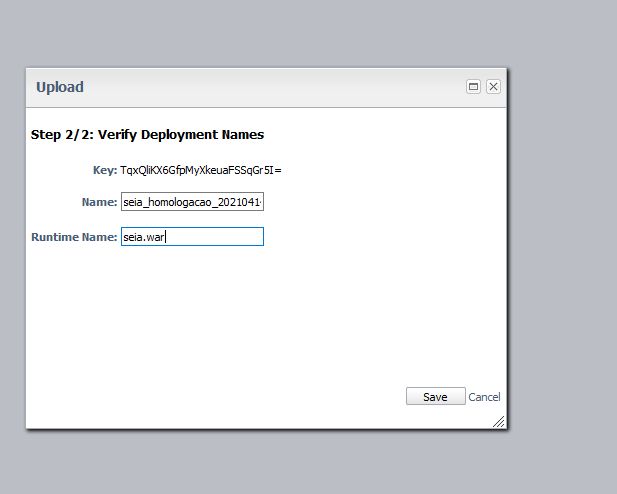


1. Clique em **Manage Deployments** e em seguida: **Add Content**
2. Preste atenção nos passos abaixo, pois qualquer erro é necessário realizar conexão ssh e reiniciar o servidor.
3. Insira o seu arquivo: **seia\_homologacao\_anomesdiahhmm\_v4167.war clicando em browser e em seguida next.**

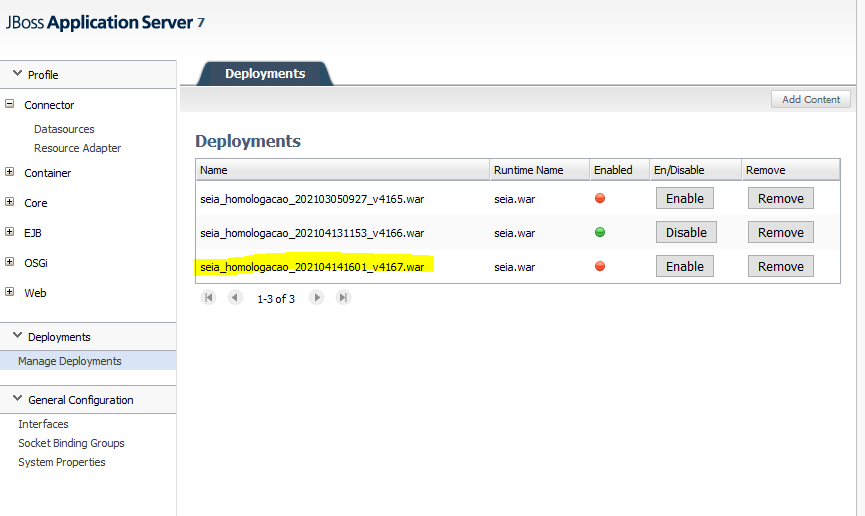
Aperte uma única vez e tenha paciência: Dica: Beber água, olhar e-mail, tomar café. Caso não tenha tido paciência, realize a conexão ssh, reinicie o servidor ~~para deixar de ser apressado~~ e realize todos os passos novamente começando do **passo 18**. #Problemas de conexão? **Passo 18;** ~~Falta de energia?~~ **Passo 18**



1. Renomeie a segunda linha para seia.war, conforme a imagem abaixo:



1. Agora a visualização do seu arquivo estará assim:



1. Antes de clicar em **Disable** vá até o discord no canal **#GERAL** e digite a seguinte mensagem conforme o horário do dia, como são 16h01m é:

**@everyone Boa tarde, vou precisar parar o 76**

1. No **passo 29** observe que é necessário digitar **@everyone** para sinalizar a todos e **Bom dia, Boa tarde ou Boa noite** que é muito difícil e o **numero do ambiente a ser parado,** no caso é ***o servidor de produção é o 76***, caso seja ***o servidor de treinamento é o 22*** e caso seja ***o servidor de homologação é o 37***.
2. **Identifique o arquivo no Jboss e clique no botão Enable e aguarde,** ~~senão terá que voltar ao passo 18, já sabe.~~
3. Quando o status do arquivo ficar verde quer dizer que deu tudo certo.
4. Acesse o endereço ip de produção e confira se no fim da página se deu tudo certo confira a guia se está: Seia 4.16.7 e no rodapé da página a versão.
5. Tudo certo então devemos voltar ao discord no canal **#GERAL** e digitar:

**@everyone 76 de volta**

1. Caso tenha ocorrido algum erro~~, lamento te informar~~, delete a branch criada com o comando: **git branch -d nomedabranch** e realize **todos os passos a partir do passo 15 = (**
2. Lembrando que realizamos os passos acima apenas para o ambiente 76 agora vamos realizar os passos resumidamente para o ambiente 22
3. Realize o git checkout develop
4. Execute o comando de homologação novamente com o comando:

**mvn clean package -Phml**

1. Realize todos os passos após o **passo 18**
2. Caso deu tudo certo e já deu a mensagem no discord para geral que voltou, siga os passos abaixo:
3. **Lembra para qual data estamos criando a branch? No caso hoje são 14/04/2021 estamos criando a branch para dia 24/04/2021 ou seja, a próxima versão logo criaremos a branch da seguinte maneira:**

**git checkout -b release/20210424**

1. Próximo comando: **git push --set-upstream origin release/20210424**
2. Finish a geração de versão de quarta-feira.

* **Quinta – Feira**

Acesse o link que pode ser acessado também pela pasta Documentação/Videos no Drive SEMASEIA

* Link oficial da documentação de geração de versão pode ser acessado na pasta Documentação/VERSÃO -> CONSIDERE APENAS NO MOMENTO DE GERAR AS VERSÕES O DOCUMENTO .TXT CHAMADO GERAR VERSÃO NO DRIVE SEMASEIA

# RESOLUÇÃO DE ERROS COMUNS DOS TICKETS

* Problema: Coordenada informada está fora dos limites da Localidade do empreendimento

**Descrição**: Ao inserir coordenadas geográficas relacionadas a um ponto de localização geográfica de um empreendimento, ao salvar, é retornanda a informação:"Coordenada informada está fora dos limites da Localidade do empreendimento." Este erro ocorre por que o município onde se encontra o empreendimento não possui valor no campo "cod\_ibge\_municipio" na tabela municipio.

**Resolução (Passo a Passo):**

**1 -** Consultar o municipio através do ide\_municipio (obtido no console do jboss ou habilitando show\_sql e método validaVerticeMunicipio na classe "LocalizacaoGeograficaService")ou do nome do municipio:

select \* from municipio where ide\_municipio = <id do municipio>

select \* from municipio where nom\_municipio = '<nome do municipio>'

**2 -** Obter o código do ibge no google ou no site https://cidades.ibge.gov.br/brasil/ba/

**3 -** Atualizar o registro do município com o código do município baseado no ide\_municipio

update municpio set cod\_ibge\_municipio = <código do ibge do município> where ide\_municipio = <id do municipio>

**Ticket referência: #138297**