

# Passo a passo da preparação do Ambiente de Desenvolvimento

## Configurar a resolução do Ubuntu 14.04 no VirtualBox

1. Primeiro, faça o boot do Ubuntu no VirtualBox.
2. Vá ao menu **Dispositivos** do VirtualBox e depois clique em **Inserir imagem de CD dos Adicionais para Convidado**.
3. Confirme o que deseja fazer
4. Entre com a senha
5. Espere pela instalação
6. Reinicie o Ubuntu

## Instalar o JRE/JDK padrão

- Java Runtime Environment (JRE) significa Ambiente de Tempo de Execução Java, e é utilizado para executar as aplicações da [plataforma Java](#). É composto por bibliotecas ([APIs](#)) e pela [Máquina virtual Java](#) (JVM).
- Java Development Kit (JDK) significa Kit de Desenvolvimento Java, e é um conjunto de utilitários que permitem criar sistemas de software para a [plataforma Java](#). É composto por [compilador](#) e [bibliotecas](#).

```
sudo apt-get update
sudo apt-get install default-jre default-jdk
java -version
javac -version
```

## Instalar o Git e utilizar o GitHub

- O Git é um sistema de controle de versões, enquanto o GitHub é um servidor onde enviamos as nossas versões de projeto. Ele serve não só para quem programa com Ruby on Rails, mas como também para qualquer outra linguagem de programação.

Primeiramente vá até o [GitHub](#) e faça seu cadastro. Logo após, abra o terminal e digite os comandos abaixo:

```
sudo apt-get update
sudo apt-get install git-core
```

```
#Configure o nome que aparecerá quando você enviar algo para
ao servidor
git config --global user.name "Your Name Here"
```

```
#Recomendo utilizar o mesmo e-mail do cadastro do GitHub
git config --global user.email "your\_email@youremail.com"
```

#Insira o comando abaixo e pressione Enter até que ele te peça uma senha

```
ssh-keygen -t rsa -C "your\_email@youremail.com"
```

#Aperte a tecla ENTER ate finalizar o comando

#Digite a senha para a sua chave e confirme.

```
sudo apt-get install xclip
```

```
xclip -sel clip < ~/.ssh/id_rsa.pub
```

Abra a página do GitHub e vá nas [configurações de chave ssh](#).

Clique em "Add ssh key"

Dê um nome para a chave

E no campo "Key" dê Colar(Ctrl+v)

Clique no botão verde escrito "Add Key"

Volte para o terminal e digite o comando abaixo:

```
ssh -T git@github.com
```

Yes

#Depois disso deve aparecer uma mensagem contendo o seu usuário do GitHub

## Instalar o Ant 1.9.5

- Apache Ant é uma ferramenta utilizada para [automação de compilação](#) na construção de [software](#). Ela é similar ao [make](#) mas é escrita na linguagem [Java](#) e foi desenvolvida inicialmente para ser utilizada em projetos desta linguagem.

Criar diretório

```
mkdir -p ~/desenv/java/apache/downloads
```

Download

```
cd ~/desenv/java/apache/downloads
```

```
wget http://ftp.unicamp.br/pub/apache//ant/binaries/apache-ant-1.9.5-bin.tar.gz
```

Descompactar arquivo

```
cd ~/desenv/java/apache/downloads
```

```
tar -zxvf apache-ant-1.9.5-bin.tar.gz
```

```
mv ~/desenv/java/apache/downloads/apache-ant-1.9.5
```

```
~/desenv/java/apache
```

Adicionar Variável de Ambiente

```
echo 'export ANT_HOME=~/desenv/java/apache/apache-ant-1.9.5'
```

```
>> ~/.bashrc
```

```
echo 'export PATH=$PATH:$ANT_HOME/bin' >> ~/.bashrc
```

Obs.: Apos **Encerrar a sessão** do usuário execute o comando:

```
ant -version
```

## Instalar o Maven 3.3.3

- Apache Maven, ou simplesmente Maven, é uma ferramenta de [automação de compilação](#) utilizada primariamente em projetos [Java](#). Ela é similar à ferramenta [Ant](#), mas é baseada em conceitos e trabalhos diferentes em um modo diferente. Também é utilizada para construir e gerenciar projetos escritos em [C#](#), [Ruby](#), [Scala](#) e outras linguagens.

### Criar diretório

```
mkdir -p ~/desenv/java/apache/downloads
```

### Download

```
cd ~/desenv/java/apache/downloads
wget http://mirror.nbtelecom.com.br/apache/maven/maven-3/3.3.3/binaries/apache-maven-3.3.3-bin.tar.gz
```

### Descompactar arquivo

```
cd ~/desenv/java/apache/downloads
tar -zxvf apache-maven-3.3.3-bin.tar.gz
mv ~/desenv/java/apache/downloads/apache-maven-3.3.3
~/desenv/java/apache
```

### Adicionar Variável de Ambiente

```
echo 'export MVN_HOME=~/desenv/java/apache/apache-maven-3.3.3' >> ~/.bashrc
echo 'export PATH=$PATH:$MVN_HOME/bin' >> ~/.bashrc
```

Obs.: Após **Encerrar a sessão** do usuário execute o comando:

```
mvn -version
```

## Instalar o Tomcat 8.0.23

- O Tomcat é um [servidor web Java](#), mais especificamente, um [container](#) de [servlets](#). O Tomcat implementa, dentre outras de menor relevância, as tecnologias [Java Servlet](#) e [JavaServer Pages](#) (JSP) e não é um container EJB.

### Criar diretório

```
mkdir -p ~/desenv/java/apache/downloads
```

### Download

```
cd ~/desenv/java/apache/downloads
wget http://mirror.nbtelecom.com.br/apache/tomcat/tomcat-8/v8.0.23/bin/apache-tomcat-8.0.23.tar.gz
```

### Descompactar arquivo

```
cd ~/desenv/java/apache/downloads
tar -zxvf apache-tomcat-8.0.23.tar.gz
```

```
mv ~/desenv/java/apache/downloads/apache-tomcat-8.0.23
~/desenv/java/apache
```

### Adicionar Variável de Ambiente

```
echo 'export CATALINA_HOME=~/desenv/java/apache/apache-
tomcat-8.0.23' >> ~/.bashrc
```

## Instalar o Gradle 2.4

- Gradle é uma ferramenta de [automação de compilação](#) que se baseia em conceitos do [Apache Ant](#) e Apache Maven e introduz o domínio de linguagem específica baseado no Groovy (DSL) em vez de formulário XML.

### Criar diretório

```
mkdir -p ~/desenv/gradle/downloads
```

### Download

```
cd ~/desenv/gradle/downloads
wget https://services.gradle.org/distributions/gradle-2.4-all.zip
```

### Descompactar arquivo

```
cd ~/desenv/gradle/downloads
unzip gradle-2.4-all.zip -d ~/desenv/gradle
```

### Adicionar Variável de Ambiente

```
echo 'export GRADLE_HOME=~/desenv/gradle/gradle-2.4' >>
~/.bashrc
echo 'export PATH=$PATH:$GRADLE_HOME/bin' >> ~/.bashrc
```

Obs.: Após **Encerrar a sessão** do usuário execute o comando:  
`gradle -v`

## Instalar o compilador C/C++

- **C** é uma [linguagem de programação compilada](#) de propósito geral, [estruturada](#), [imperativa](#), [procedural](#), [padronizada pela ISO](#), criada em [1972](#), por [Dennis Ritchie](#), no [AT&T Bell Labs](#), para desenvolver o [sistema operacional Unix](#) (que foi originalmente escrito em [Assembly](#)).
- **C++** (em [português](#) lê-se "cê mais mais", em [inglês](#) lê-se *see plus plus*) é uma [linguagem de programação multi-paradigma](#) e de uso geral. A linguagem é considerada de médio nível, pois combina características de linguagens de [alto](#) e [baixo](#) níveis. Desde os [anos 1990](#) é uma das linguagens comerciais mais populares, sendo bastante usada também na [academia](#) por seu grande desempenho e base de utilizadores. [Bjarne Stroustrup](#) desenvolveu o C++ (originalmente com o nome *C with Classes*, [1](#) que significa *C com classes* em português) em [1983](#) no [Bell Labs](#) como um adicional à linguagem [C](#). Novas

características foram adicionadas com o tempo, como funções virtuais, sobrecarga de [operadores](#), [herança múltipla](#), gabaritos e [tratamento de exceções](#). Após a padronização [ISO](#) realizada em [1998](#) e a posterior revisão realizada em [2003](#), uma nova versão da especificação da linguagem foi lançada em dezembro de [2014](#), conhecida informalmente como [C++14](#)

```
sudo apt-get update
sudo apt-get install g++
```

## Instalar o AVR

- **AVR** é um [microcontrolador RISC](#) de [chip](#) único com uma [arquitetura Harvard modificada](#) de [8-bit](#) ( $\mu$ C), [1](#) desenvolvido pela [Atmel](#) em [1996](#).<sup>2</sup> Foi um dos primeiros da família de microcontroladores a utilizar uma [memória flash](#) com o intuito de armazenar a programação, diferentemente de seus concorrentes da época, que utilizavam memórias do tipo [PROM](#), [EPROM](#) ou [EEPROM](#).

```
sudo apt-get update
sudo apt-get install avrdude binutils-avr gcc-avr avr-libc
gdb-avr
```

## Instalar o banco de dados PostgreSQL

- **PostgreSQL** é um [sistema gerenciador de banco de dados objeto relacional](#) (SGBDOR), desenvolvido como projeto de [código aberto](#).

```
sudo apt-get update
sudo apt-get install postgresql postgresql-contrib pgadmin3
```

Instalação feita, o próximo passo é **alterar a senha** do usuário **postgres**:

```
sudo -i -u postgres
psql postgres
ALTER USER postgres WITH PASSWORD 'postgres';
\q
exit
```

Feito a instalação e mudança de senha do usuário postgres, você estará apto a desenvolver um trabalho no seu computador conectando normalmente ao PostgreSQL, porém, se a ideia é disponibilizar o acesso ao banco para receber conexões de outras máquinas, vai ter que alterar dois arquivos no diretório `/etc/postgresql/9.3/main:` **postgresql.conf** e **pg\_hba.conf**.

```
sudo su - postgres
```

```
cd /etc/postgresql/9.3/main
vi postgresql.conf
```

1. Digite **/listen** para busca a linha **#listen\_addresses ...**
2. Aperte a tecla **i** para editar o arquivo
3. Remova o caráter **#** do início da linha, e altere **'localhost'** por **'\*'**
4. Após alteração, a linha ficara como segue abaixo:  
**listen\_addresses = '\*' ...**
5. Aperte a tecla **ESC**
6. Digite **:wq** para salvar a alteração do arquivo **postgresql.conf**

```
vi pg_hba.conf
```

1. Digite **/127.0.0.1** para busca a linha **host all all ...**
2. Aperte a tecla **i** para editar o arquivo
3. Altere **127.0.0.1/32** por **0.0.0.0/0**
4. Após alteração, a linha ficara como segue abaixo:  
**host all all 0.0.0.0/0 md5**
5. Aperte a tecla **ESC**
6. Digite **:wq** para salvar a alteração do arquivo **pg\_hba.conf**

```
exit
```

Feito a alteração, basta reiniciar o PostgreSQL

```
sudo /etc/init.d/postgresql restart
```

Obs.: Para testar as alterações no PostgreSQL, execute o comando abaixo:

```
psql -U postgres -h IP da maquina
\q
```

## Instalar o SonarQube Runner

Criar diretório

```
mkdir -p ~/desenv/sonarqube/downloads
```

Download

```
cd ~/desenv/sonarqube/downloads
wget
```

```
http://repo1.maven.org/maven2/org/codehaus/sonar/runner/sonar-runner-dist/2.4/sonar-runner-dist-2.4.zip
```

Descompactar arquivo

```
cd ~/desenv/sonarqube/downloads
unzip sonar-runner-dist-2.4.zip -d ~/desenv/sonarqube
```

Adicionar Variável de Ambiente

```
echo 'export SONAR_RUNNER_HOME=~/desenv/sonarqube/sonar-runner-2.4' >> ~/.bashrc
```

```
echo 'export PATH=$PATH:$SONAR_RUNNER_HOME/bin' >> ~/.bashrc
```

## Instalar o SonarQube

- SonarQube é uma plataforma de código aberto para inspeção contínua da qualidade do código.

### Criar diretório

```
mkdir -p ~/desenv/sonarqube/downloads
```

### Download

```
cd ~/desenv/sonarqube/downloads
wget http://downloads.sonarsource.com/sonarqube/sonarqube-4.5.4.zip
```

### Descompactar arquivo

```
cd ~/desenv/sonarqube/downloads
unzip sonarqube-4.5.4.zip -d ~/desenv/sonarqube
```

### Adicionar Variável de Ambiente

```
echo 'export SONAR_HOME=~/desenv/sonarqube/sonarqube-4.5.4'
>> ~/.bashrc
echo 'export PATH=$PATH:$SONAR_HOME/bin/linux-x86-64' >>
~/.bashrc
```

### Criar banco **sonar** no PostgreSQL

```
psql -d template1 -U postgres -h IP da máquina
CREATE USER sonar WITH PASSWORD 'sonar';
CREATE DATABASE sonar;
GRANT ALL PRIVILEGES ON DATABASE sonar to sonar;
\q
```

### Alterar o arquivo **sonar.properties**

```
gedit ~/desenv/sonarqube/conf/sonar.properties
```

1. Remova o carácter # do início das linhas abaixo:

```
#sonar.jdbc.username=sonar
#sonar.jdbc.password=sonar
#sonar.jdbc.url=jdbc:postgresql://localhost/sonar
```

Obs.: Após **Encerrar a sessão** do usuário execute o comando:  
`sonar.sh`

## Instalar o Android SDK

- Os aplicativos Android são normalmente desenvolvidos em linguagem de programação Java usando o kit de desenvolvimento de software (SDK).

### Criar diretório

```
mkdir -p ~/desenv/android/downloads
```

## Download

```
cd ~/desenv/android/downloads
wget https://dl.google.com/android/android-sdk\_r24.3.3-linux.tgz
```

## Descompactar arquivo

```
cd ~/desenv/android/downloads
tar -zxvf android-sdk_r24.3.3-linux.tgz
mv ~/desenv/android/downloads/android-sdk-linux
~/desenv/android
```

## Adicionar Variável de Ambiente

```
echo 'export ANDROID_HOME=~/desenv/android/android-sdk-linux'
>> ~/.bashrc
echo 'export PATH=$PATH:$ANDROID_HOME/tools' >> ~/.bashrc
echo 'export PATH=$PATH:$ANDROID_HOME/platform-tools' >>
~/.bashrc
```

Obs.: Após **Encerrar a sessão** do usuário execute o comando:  
`android`

## Instalar o Android Studio

- Android Studio é uma IDE para o desenvolvimento na plataforma Android.

## Criar diretório

```
mkdir -p ~/desenv/android/downloads
mkdir -p ~/desenv/android/AndroidStudioProjects
```

## Download

```
cd ~/desenv/android/downloads
wget https://dl.google.com/dl/android/studio/ide-zips/1.2.2.0/android-studio-ide-141.1980579-linux.zip
sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0
lib32stdc++6
```

## Descompactar arquivo

```
cd ~/desenv/android/downloads
unzip android-studio-ide-141.1980579-linux.zip -d
~/desenv/android
```

## Configurar o Ubuntu launcher

```
gksudo gedit /usr/share/applications/AndroidStudio.desktop
```

Copie o conteúdo abaixo no arquivo **AndroidStudio.desktop**

Obs.: Alterar o **Usuário Ubuntu** pelo diretório do usuário logado

```
#-----
[Desktop Entry]
```



```
Type=Application
Name=Android Studio
Comment=Android Studio is an integrated development environment
(IDE) for developing on the Android platform
Exec="/home/Usuário Ubuntu/desenv/android/android-
studio/bin/studio.sh" %f
Icon=/home/Usuário Ubuntu/desenv/android/android-
studio/bin/studio.png
Categories=Development;IDE;
Terminal=false
StartupNotify=true
StartupWMClass=jetbrains-android-studio
Name[en_GB]=Android Studio
#-----
```

## ***Instalar o Arduíno***

- Arduino, palavra por vezes traduzida ao [português](#) como *Arduíno*, [4](#) [2](#) [5](#) é uma plataforma de [prototipagem](#) eletrônica de [hardware livre](#) e de [placa única](#), [6](#) projetada com um [microcontrolador Atmel AVR](#) com suporte de [entrada/saída](#) embutido, uma [linguagem de programação](#) padrão, [7](#) a qual tem origem em [Wiring](#), e é essencialmente [C/C++](#). [8](#) O objetivo do projeto é criar ferramentas que são acessíveis, com baixo custo, flexíveis e fáceis de se usar por artistas e amadores. Principalmente para aqueles que não teriam alcance aos controladores mais sofisticados e de ferramentas mais complicadas.

### **Criar diretório**

```
mkdir -p ~/desenv/arduino/downloads
mkdir -p ~/desenv/arduino/workspace
```

### **Download**

```
cd ~/desenv/arduino/downloads
wget http://downloads.arduino.cc/arduino-1.6.5-linux64.tar.xz
```

### **Descompactar arquivo**

```
cd ~/desenv/arduino/downloads
tar xvf arduino-1.6.5-linux64.tar.xz
mv ~/desenv/arduino/downloads/arduino-1.6.5 ~/desenv/arduino
```

### **Configurar o Ubuntu launcher**

```
gksudo gedit /usr/share/applications/arduino.desktop
```

Copie o conteúdo abaixo no arquivo **arduino.desktop**

Obs.: Alterar o **Usuário Ubuntu** pelo diretório do usuário logado

```
#-----
[Desktop Entry]
Type=Application
Name=Arduino IDE
```

```

GenericName=Integrated Development Environment
Comment=An IDE for Arduino-compatible electronics prototyping
platforms
Exec=/home/Usuário Ubuntu/desenv/arduino/arduino-1.6.5/arduino
Icon=/home/Usuário Ubuntu/desenv/arduino/arduino-
1.6.5/lib/arduino.png
Terminal=false
Categories=Development;IDE;Electronics;
MimeType=text/x-arduino
Keywords=embedded electronics;electronics;avr;microcontroller;
#-----

```

## Instalar o Eclipse JEE Luna SR2

- Eclipse é um [IDE](#) para desenvolvimento [Java](#), porém suporta várias outras linguagens a partir de [plugins](#) como [C/C++](#), [PHP](#), [ColdFusion](#), [Python](#), [Scala](#) e plataforma [Android](#). Ele foi feito em [Java](#) e segue o modelo [open source](#) de [desenvolvimento de software](#).

### Criar diretório

```

mkdir -p ~/desenv/eclipse/downloads
mkdir -p ~/desenv/eclipse/workspace

```

### Download

```

cd ~/desenv/eclipse/downloads
wget
http://eclipse.c3sl.ufpr.br/technology/epp/downloads/release/
luna/SR2/eclipse-jee-luna-SR2-linux-gtk-x86\_64.tar.gz

```

### Descompactar arquivo

```

cd ~/desenv/eclipse/downloads
tar -zxvf eclipse-jee-luna-SR2-linux-gtk-x86_64.tar.gz
mv ~/desenv/eclipse/downloads/eclipse
~/desenv/eclipse/eclipse-jee-luna-SR2

```

### Configurar o Ubuntu launcher

```

gksudo gedit /usr/share/applications/eclipse.desktop

```

Copie o conteúdo abaixo no arquivo **eclipse.desktop**

Obs.: Alterar o **Usuário Ubuntu** pelo diretório do usuário logado

```

#-----
[Desktop Entry]
Name=Eclipse JEE Luna SR2
Type=Application
Exec=/home/Usuário Ubuntu/desenv/eclipse/eclipse-jee-luna-
SR2/eclipse
Terminal=false
Icon=/home/Usuário Ubuntu/desenv/eclipse/eclipse-jee-luna-
SR2/icon.xpm
Comment=Integrated Development Environment

```

```
NoDisplay=false
Categories=Development;IDE;
Name[en]=Eclipse JEE Luna SR2
#-----
```

### Instalar o Plug-in do **Gradle** no Eclipse (Opcional)

1. Primeiro, vá ao menu (**Help > Eclipse Marketplace...**)
2. No campo "**Find:**", escreva "**gradle ide**" e depois clique no botão "**Go**".
3. Clique no botão "**Install**" da opção "**Gradle IDE Pack...**"
4. Deixe todos os checkboxes marcados, e depois clique no botão "**Confirm**".
5. Aceite os termos do contrato, e depois clique no botão "**Finish**".
6. Após terminar a instalação do Plug-in, reinicie o Eclipse.

### Instalar o Plug-in do **Groovy/Grails** no Eclipse

1. Primeiro, vá ao menu (**Help > Eclipse Marketplace...**)
2. No campo "**Find:**", escreva "**groovy**" e depois clique no botão "**Go**".
3. Clique no botão "**Install**" da opção "**Groovy/Grails Tool Suite (GGTS) for Eclipse Luna...**"
4. Deixe todos os checkboxes marcados, e depois clique no botão "**Confirm**".
5. Aceite os termos do contrato, e depois clique no botão "**Finish**".
6. Após terminar a instalação do Plug-in, reinicie o Eclipse.

### Instalar o Plug-in do **JaCoCo** no Eclipse (Opcional)

- JaCoCo Java Code Coverage
1. Primeiro, vá ao menu (**Help > Eclipse Marketplace...**)
  2. No campo "**Find:**", escreva "**jacoco**" e depois clique no botão "**Go**".
  3. Clique no botão "**Install**" da opção "**EclEmma Java Code Coverage...**"
  4. Deixe todos os checkboxes marcados, e depois clique no botão "**Confirm**".
  5. Aceite os termos do contrato, e depois clique no botão "**Finish**".
  6. Após terminar a instalação do Plug-in, reinicie o Eclipse.

### Instalar o Plug-in do **C/C++** no Eclipse

1. Primeiro, vá ao menu (**Help > Eclipse Marketplace...**)
2. No campo "**Find:**", escreva "**cdt**" e depois clique no botão "**Go**".
3. Clique no botão "**Install**" da opção "**Eclipse CDT(C/C++ Development Tooling)...**"
4. Deixe todos os checkboxes marcados, e depois clique no botão "**Confirm**".
5. Aceite os termos do contrato, e depois clique no botão

**"Finish".**

6. Após terminar a instalação do Plug-in, reinicie o Eclipse.

### Instalar o [Plugin-in do AVR](http://avr-eclipse.sourceforge.net/updatesite) no Eclipse

1. Primeiro, vá ao menu (**Help > Install New Software...**)
2. No campo **"Work with:"**, cole **"http://avr-eclipse.sourceforge.net/updatesite"** e depois clique no botão **"Add"**.
3. No campo **"Name"**, escreva **"AVR-Plugin"** e depois clique no botão **"Ok"**.
4. Deixe todos os checkboxes marcados, e depois clique no botão **"Next>"**.
5. Aceite os termos do contrato, e depois clique no botão **"Finish"**.
6. Após terminar a instalação do Plug-in, reinicie o Eclipse.