



Tutorial de Instalação e Integração do ROS e V-REP no Ubuntu 16.04

Conteúdo

1	Ubuntu	3
2	ROS	4
3	V-REP	4
4	Integração V-REP + ROS	5
4.1	Definição das variáveis de ambiente do V-REP	5
4.2	Instalação dos pré-requisitos	7
4.3	Instalação do plugin para integrar o ROS ao V-REP	7
4.4	Verificação da instalação correta do plugin	10
5	Instalação dos pacotes para o Robotino	11
6	Dúvidas	14

1 Ubuntu

O Ubuntu é uma distribuição de Linux baseada no Debian que é gratuita e open source. De acordo com o Google Trends, o Ubuntu é a distro mais popular de Linux para desktops, com uma margem significativa em relação ao Debian, a segunda mais popular. Por isso, o Ubuntu conta com o melhor suporte para bibliotecas e programas. Além disso, a Canonical compromete-se a dar suporte por até 5 anos às distribuições LTS do Ubuntu, como a versão 16.04, que deve ser mantida até 2021.

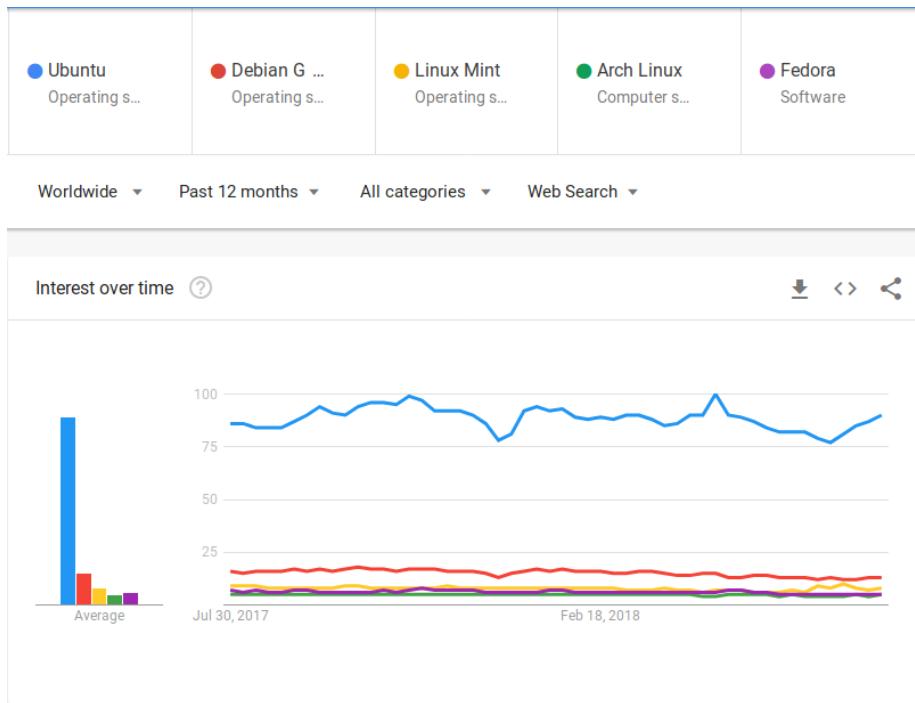


Figura 1: Popularidade das maiores distribuições do Linux.

O tutorial oficial de instalação do Ubuntu é bem detalhado, mas há alguns pré-requisitos para a instalação. Primeiro, é necessário ter um dispositivo botável com a imagem do Ubuntu gravada, como um USB flash ou um DVD. Segundo, deve haver uma partição que possa ser alocada para a instalação do Ubuntu. Caso o seu dispositivo de armazenamento tenha uma única partição com Windows e você não quer comprometer os seus arquivos e programas instalados, a melhor opção é configurar um sistema com dual-boot (i.e. instalar o Ubuntu junto ao Windows). Para isso, será necessário desalocar parte do volume para que na instalação do Ubuntu seja possível selecionar esse espaço e formatá-lo apropriadamente.

- Link para alocação de espaço para uma nova partição a partir do Windows:

<http://www.pagestart.com/win7win10tpdb10121403.html>

- Link para download das imagens do Ubuntu:

<http://releases.ubuntu.com/releases/16.04/ubuntu-16.04.4-desktop-amd64.iso>

- Link para gravar a imagem do Ubuntu em um USB flash a partir do Windows:

<https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-windows#0>

- Link para bootar pelo USB (análogo para DVD):

<https://www.lifewire.com/how-to-boot-from-a-usb-device-2626091>

- Link para instalação do Ubuntu:

<https://tutorials.ubuntu.com/tutorial/tutorial-install-ubuntu-desktop-1604#0>

2 ROS

O ROS (Robot Operating System) é uma framework para auxiliar no desenvolvimento de aplicações para robótica. O ROS proviedia abstrações de hardware, drivers, visualizadores, gerenciamento de pacotes e protocolos de mensagem para tornar o desenvolvimento de robôs acessível a um público maior.

Existe mais de uma distribuição do ROS suportada. Análogo ao Ubuntu, ele suporta versões LTS e STS. Em geral, as versões LTS são mais estáveis, já que há menos experimentação por parte dos desenvolvedores. Existe uma versão LTS do ROS para cada versão LTS do Ubuntu, o Kinetic é a distribuição do ROS que corresponde à versão 16.04 do Ubuntu e é a que vamos usar. Para instalar o ROS, basta seguir os passos 1.1 ao 1.7 no tutorial: <http://wiki.ros.org/kinetic/Installation/Ubuntu>. Apesar do “Desktop Full Install” ser a opção recomendada, para o workshop só são necessárias as componentes disponíveis no “Desktop Install”.

3 V-REP

O V-REP é um simulador para robôs baseado em uma arquitetura de controle distribuída onde cada objeto pode ser controlado individualmente através de um script, plugin, nó do ROS ou usando uma API remota.

Para instalar o V-REP basta baixar o arquivo *tar.gz* disponível no site oficial e descompactá-lo:

1. Abra um terminal pressionando **Ctrl+Alt+T**.
2. Crie um diretório para os arquivos do V-REP:

```
mkdir -p ~/path/to/vrep
```

3. Baixe o arquivo comprimido com os binários em: http://coppeliarobotics.com/files/V-REP_PRO_EDU_V3_5_0_Linux.tar.gz.
4. Vá até o diretório de Downloads:

```
cd ~/Downloads
```

5. Extraia o arquivo *tar.gz* do V-REP para o diretório criado:

```
tar xf V-REP_PRO_EDU_V3_5_0_Linux.tar.gz -C ~/path/to/vrep
```

6. Mude para o diretório onde o V-REP foi descompactado:

```
cd ~/path/to/vrep/V-REP_PRO_EDU_V3_5_0_Linux
```

7. Verifique que tudo está funcionando rodando o script:

```
./vrep.sh
```

Caso tenha seguido os passos corretamente, o V-REP deve rodar normalmente. Feche o programa e continue na próxima seção.

4 Integração V-REP + ROS

4.1 Definição das variáveis de ambiente do V-REP

Nesta etapa, devemos adicionar uma variável de ambiente chamada *VREP_ROOT* com o caminho para a pasta do V-REP:

1. Abra o arquivo *.bashrc*:

```
gedit ~/.bashrc
```

2. Adicione a seguinte no final do arquivo e depois salve o arquivo e feche o editor, como indicado na Figura 2, lembrando-se de trocar o caminho no comando abaixo pelo correspondente a sua instalação:

```
export VREP_ROOT=~/path/to/vrep/V-REP_PRO_EDU_V3_5_0_Linux
```

3. Carregue as variáveis de ambiente definidas no passo anterior.

```
source ~/.bashrc
```

4. Verifique que a variável está setada corretamente:

```
echo $VREP_ROOT
```

Se o caminho *~/path/to/vrep/V-REP_PRO_EDU_V3_5_0_Linux* apareceu na saída padrão do terminal, como na Figura 3, siga em frente.

Figura 2: Inserção do comando no arquivo `~/.bashrc` que exporta a variável `VREP_ROOT`.

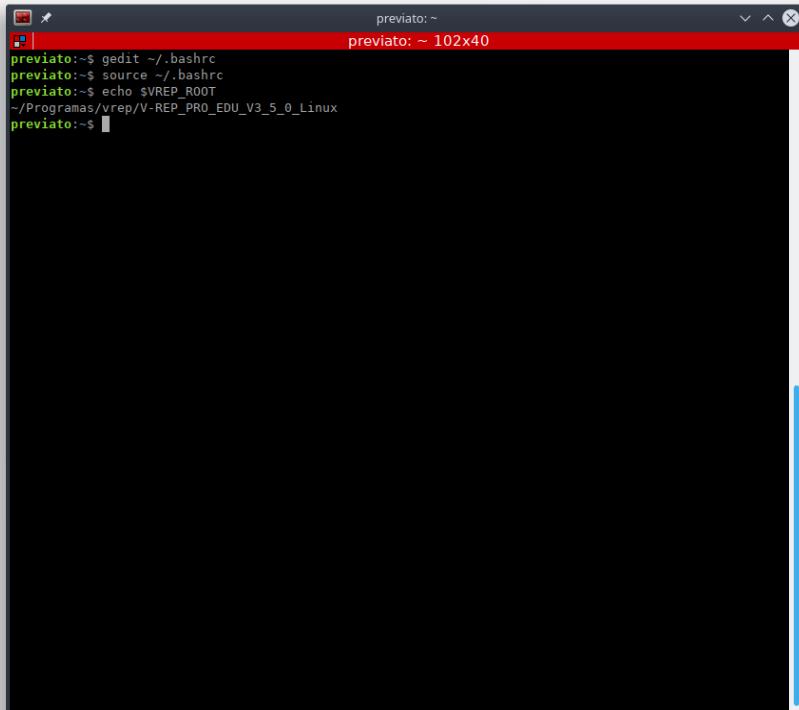


Figura 3: Resultado do comando `echo $VREP-ROOT` após exportar a variável de ambiente.

4.2 Instalação dos pré-requisitos

Antes de instalar as interfaces, precisaremos garantir que temos os alguns pacotes e programas instalados:

1. Instale os programas necessários:

```
sudo apt-get install -y build-essential git cmake xsltproc
```

2. Instale os pacotes para Python:

```
sudo apt-get install -y python-catkin-tools python-lxml \
python-tempita
```

4.3 Instalação do plugin para integrar o ROS ao V-REP

Para instalarmos o plugin do ROS no V-REP, utilizaremos o catkin, sistema de construção de pacotes oficial do ROS. No workshop entraremos em detalhe a respeito do funcionamento do catkin, mas por enquanto vamos apenas utilizá-lo.

1. Crie um workspace do catkin temporário, utilizando o comando mkdir:

```
mkdir -p /tmp/quickstart_ws/src
```

2. Navegue até o diretório criado:

```
cd /tmp/quickstart_ws
```

3. Inicialize o workspace com o comando catkin init. Deve-se observar uma saída semelhante à da Figura 4 no terminal:

```
catkin init
```

4. Clone o RosInterface nesse workspace com o git. Verifique uma saída semelhante à da Figura 5:

```
cd src/
git clone --recursive \
https://github.com/CoppeliaRobotics/v_repExtRosInterface.git
```

5. Construa o RosInterface no workspace que acabamos de criar. A saída no terminal deve ser semelhante à da Figura 6:

```
catkin build
```

6. Verifique que a biblioteca para integração do V-REP com o ROS foi criada:

```
cd ../devel/lib/
ls
```

Você deve ver um arquivo chamado *libv_repExtRosInterface.so*. Os arquivos na pasta devem ser os mesmos indicados na Figura 7.

```

previato:~$ mkdir -p /tmp/quickstart_ws/src
previato:~$ cd /tmp/quickstart_ws/
previato:/tmp/quickstart_ws$ catkin init
Initializing catkin workspace in '/tmp/quickstart_ws'.
Profile:           default
Extending:        [env] /opt/ros/melodic
Workspace:        /tmp/quickstart_ws
Source Space:     [exists] /tmp/quickstart_ws/src
Log Space:        [missing] /tmp/quickstart_ws/logs
Build Space:      [missing] /tmp/quickstart_ws/build
Devel Space:      [missing] /tmp/quickstart_ws/devel
Install Space:    [unused] /tmp/quickstart_ws/install
DESTDIR:          [unused] None
Devel Space Layout: linked
Install Space Layout: None
Additional CMake Args: None
Additional Make Args: None
Additional catkin Make Args: None
Internal Make Job Server: True
Cache Job Environments: False
Whitelisted Packages: None
Blacklisted Packages: None
Workspace configuration appears valid.

previato:/tmp/quickstart_ws$ 

```

Figura 4: Saída padrão após a inicialização do workspace com o comando `catkin init`.

```

previato:/tmp/quickstart_ws/src$ git clone --recursive https://github.com/CoppeliaRobotics/v_repExtRosInterface.git vrep_ros_interface
Cloning into 'vrep_ros_interface'...
remote: Counting objects: 695, done.
remote: Total 695 (delta 0), reused 0 (delta 0), pack-reused 695
Receiving objects: 100% (695/695) 3.49 MiB | 2.72 MiB/s, done.
Resolving deltas: 100% (362/362), done.
Submodule 'external/v_repPlusPlus' (https://github.com/CoppeliaRobotics/v_repPlusPlus) registered for path 'external/v_repPlusPlus'
Submodule 'external/v_repStubsGen' (https://github.com/CoppeliaRobotics/v_repStubsGen) registered for path 'external/v_repStubsGen'
Cloning into '/tmp/quickstart_ws/src/vrep_ros_interface/external/v_repPlusPlus'...
remote: Counting objects: 49, done.
remote: Total 49 (delta 0), reused 0 (delta 0), pack-reused 49
Cloning into '/tmp/quickstart_ws/src/vrep_ros_interface/external/v_repStubsGen'...
remote: Counting objects: 592, done.
remote: Compressing objects: 100% (19/19), done.
remote: Writing objects: 100% (592/592), 117.79 KiB | 773.00 KiB/s, done.
Resolving deltas: 100% (362/362), done.
Submodule path 'external/v_repPlusPlus': checked out 'cb31df76daac19580cbf2b071e431b3c4e8f2d8f2'
Submodule path 'external/v_repStubsGen': checked out '9125e020d5b754dab2bf575a12d5fc55a782d2'
Submodule 'external/pycpp' (https://github.com/ferrari/pycpp.git) registered for path 'external/v_repStubsGen/external/pycpp'
Cloning into '/tmp/quickstart_ws/src/vrep_ros_interface/external/v_repStubsGen/external/pycpp'...
remote: Counting objects: 185, done.
remote: Total 185 (delta 0), reused 0 (delta 0), pack-reused 185
Receiving objects: 100% (185/185) 25.47 KiB | 159.00 KiB/s, done.
Resolving deltas: 100% (101/101), done.
Submodule path 'external/v_repStubsGen/external/pycpp': checked out 'c5ec1cf71c303d1115342cc18f74b8902b0f0c8e'

previato:/tmp/quickstart_ws/src$ 

```

Figura 5: Saída padrão após clonar o repositório com o plugin da interface entre o V-REP e o ROS.

```

previato:/tmp/quickstart_ws/src$ catkin build
[build] Found '1' packages in 0.0 seconds.
[build] Updating package table.
Starting >>> vrep_ros_interface
[build] <<< vrep_ros_interface      [ 44.6 seconds ]
[build] Summary: All 1 packages succeeded!
[build] Ignored:   None
[build] Skipped:  None
[build] Abandoned: None
[build] Failed:   None.
[build] Runtime: 44.9 seconds total.
previato:/tmp/quickstart_ws/src$ 

```

Figura 6: Saída padrão após a construção do pacote com o comando `catkin build`.

```

previato:/tmp/quickstart_ws/src$ cd ..../devel/lib
previato:/tmp/quickstart_ws/devel/lib$ ls
libvrepExtRosInterface.so  pkgconfig python2.7
previato:/tmp/quickstart_ws/devel/lib$ 

```

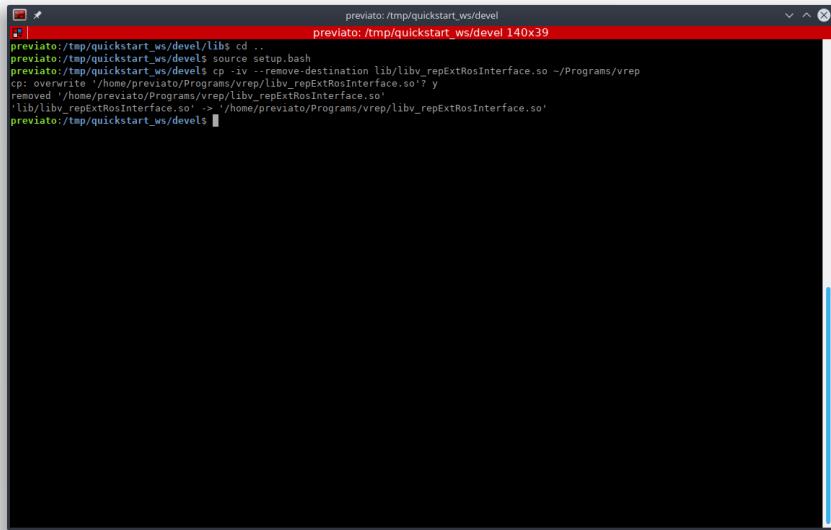
Figura 7: Lista de arquivos no diretório `/tmp/quickstart_ws/devel/lib`.

7. Novamente, rode o comando source para que o console reconheça as mudanças. desta vez, utilize o arquivo *setup.bash* present no workspace como parametro:

```
cd ..  
source setup.bash
```

8. Por fim, copie a biblioteca na pasta do V-REP (caso seja necessário, sobre-escreva o arquivo, veja imagem). A saída no terminal deve ser semelhante à Figura 8:

```
cp -iv --remove-destination lib/libv_repExtRosInterface.so \  
~/path/to/vrep/V-REP_PRO_EDU_V3_5_0_Linux
```



A screenshot of a terminal window titled "previato:/tmp/quickstart_ws/devel". The window shows the command "cp -iv --remove-destination lib/libv_repExtRosInterface.so ~/path/to/vrep/V-REP_PRO_EDU_V3_5_0_Linux" being run. The output indicates that the file was successfully copied, with a confirmation message: "removed '/home/previato/Programs/vrep/libv_repExtRosInterface.so'". The terminal window has a red header bar.

Figura 8: Saída padrão após copiar o plugin para a pasta de instalação do V-REP.

4.4 Verificação da instalação correta do plugin

1. Abra um terminal e execute o comando roscore:

```
roscore
```

Neste momento várias informações aparecerão na tela, como na Figura 9, indicando que o RosCore está rodando e você ficará impossibilitado de digitar comandos nesta instância do terminal.

2. Abra outro terminal e inicie o V-REP:

```

previato:~$ roscore
... logging to /home/previato/.ros/log/972dc504-895d-11e8-a3e2-e03f49e51285/roslaunch-previato-kubuntu-desk-7514.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt.
Done Checking log file disk usage. Usage is <10B.

started roslaunch server http://previato-kubuntu-desk:40061/
ros_comm version 1.14.2

SUMMARY
========
PARAMETERS
  * /rosdistro: melodic
  * /rosversion: 1.14.2
NODES
auto-starting new master
process[master]: started with pid [7560]
ROS_MASTER_URI=http://previato-kubuntu-desk:11311/
setting /run_id to 972dc504-895d-11e8-a3e2-e03f49e51285
process[rosout-1]: started with pid [7574]
started core service [/rosout]

```

Figura 9: Saída padrão após execução do comando `roscore`.

```

cd ~/path/to/vrep/V-REP_PRO_EDU_V3_5_0_Linux
./vrep.sh

```

3. Abra a cena `rosInterfaceTopicPublisherAndSubscriber.ttt`. As Figuras 10-12 ilustram a navegação no V-REP para abrir a cena.
4. Aperte o botão de play para rodar a simulação, como indicado pela Figura 13. Caso as duas janelas flutuantes aparecam na simulação, como na Figura 14, a instalação do plugin funcionou e podemos prosseguir para a próxima etapa.

5 Instalação dos pacotes para o Robotino

1. Instale os pacote navigation e grid_map do ROS:

```

sudo apt-get install ros-kinetic-navigation* ros-kinetic-grid-map*

```

2. Execute os comandos abaixo para adicionar o repositório do Robotino:

```

sudo su
echo "deb http://packages.openrobotino.org/xenial xenial main"\ \
> /etc/apt/sources.list.d/openrobotino.list
sudo apt-get update

```

3. Execute os comandos abaixo para instalar a API2 do Robotino:

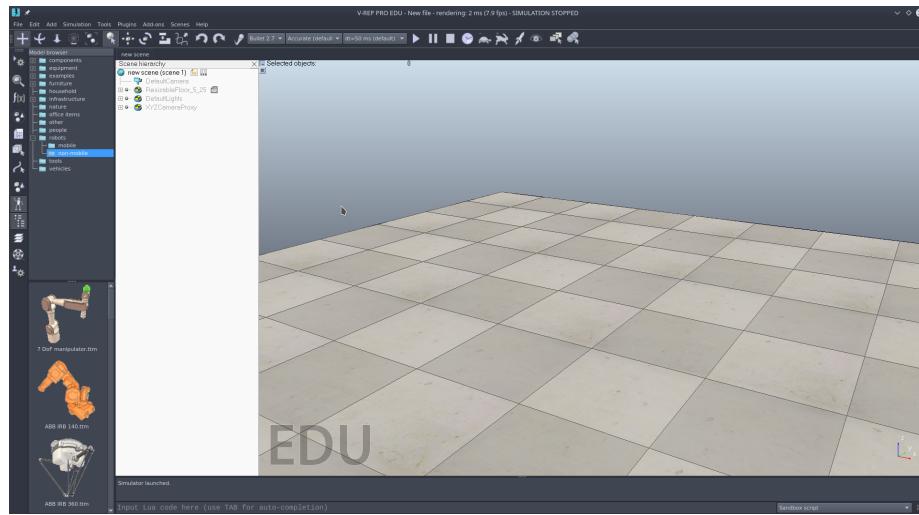


Figura 10: Página inicial do V-REP após executar o script de inicialização.

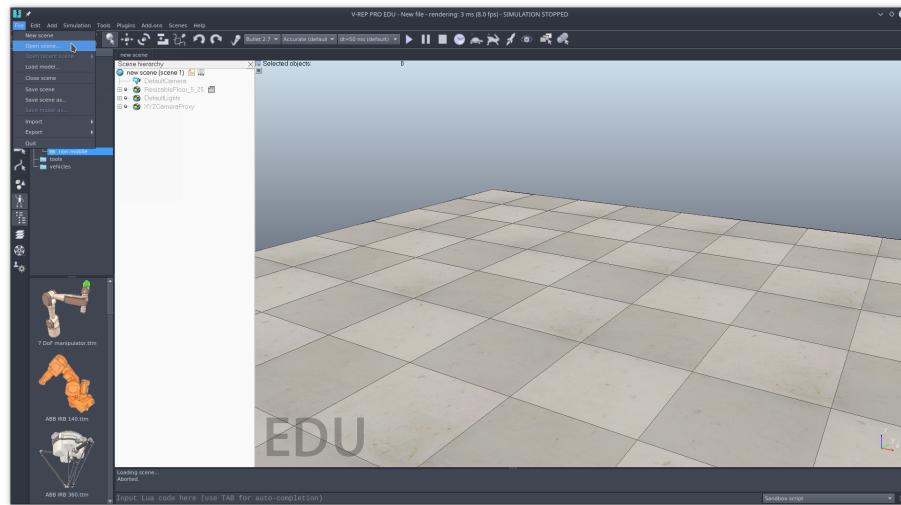


Figura 11: Local no menu onde encontra-se a opção para abrir uma nova cena.

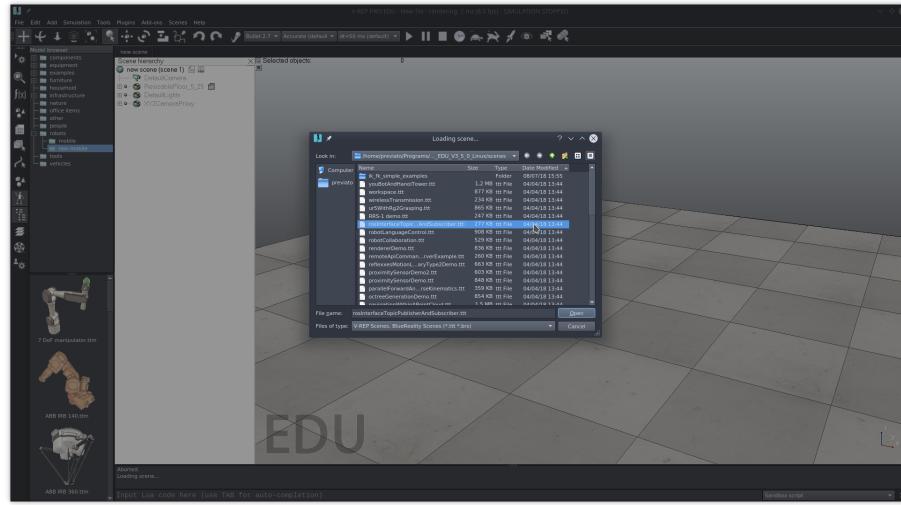


Figura 12: Menu de seleção da cena *rosInterfaceTopicPublisherAndSubscriber.ttt*.

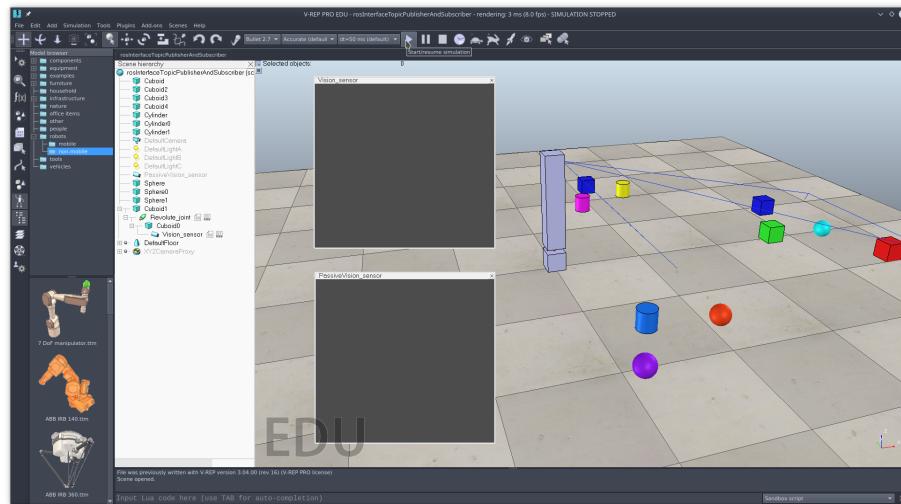


Figura 13: Local do botão play para iniciar a simulação.

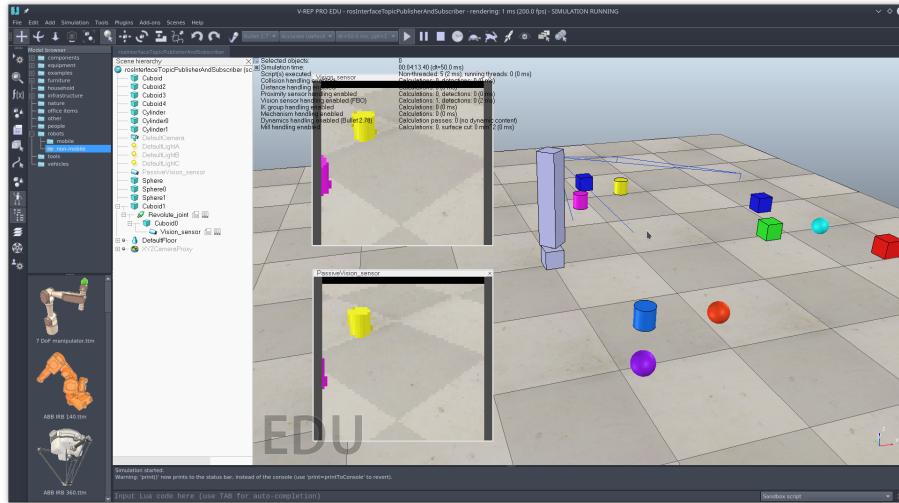


Figura 14: Comportamento esperado caso o plugin tenha sido instalado corretamente.

```
wget -qO - http://packages.openrobotino.org/keyFile | \
sudo apt-key add -
sudo apt-get install -y robotino-common robotino-api2
```

6 Dúvidas

Caso haja alguma dúvida na instalação, entre em contato com a equipe através do canal do Discord: <https://discord.gg/5UF4WTT>.