

Comece a desenvolver projetos open source

Louise Poubel

louise@osrfoundation.org

Síntese

- Sobre mim
- Open Source Robotics Foundation
- Código aberto
- Como contribuir (passo-a-passo)

Louise Poubel



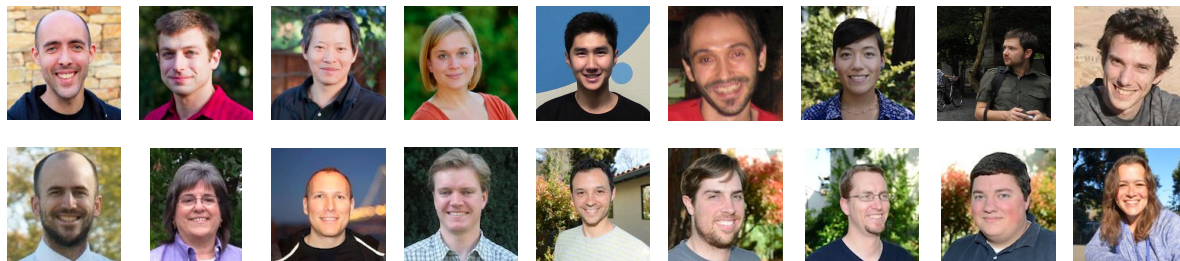
- BS Engenharia mecatrônica
 - Chiba University
- MSc Robótica Avançada
 - Warsaw Institute of Technology
 - Ecole Centrale de Nantes: [Kinect + NAO](#)
- Open Source Robotics Foundation
 - Estágio Outreachy: [GzWeb](#)
 - Engenheira de Software: [Gazebo](#)



Open Source Robotics Foundation

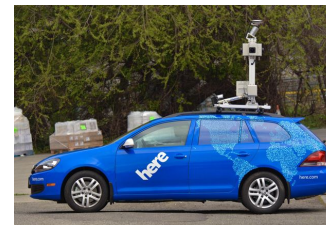
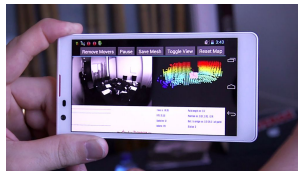
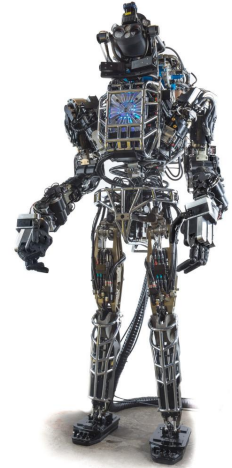
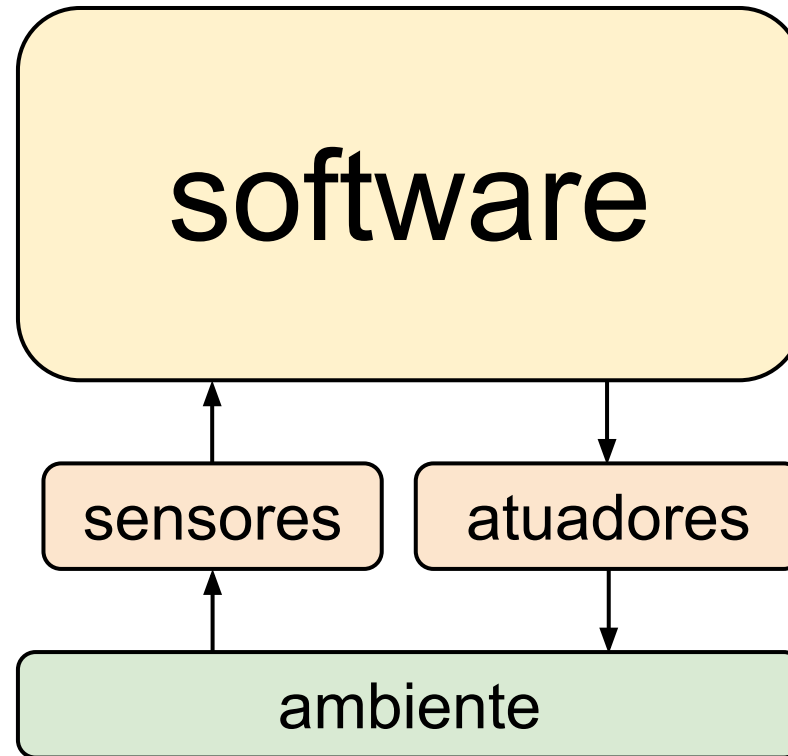
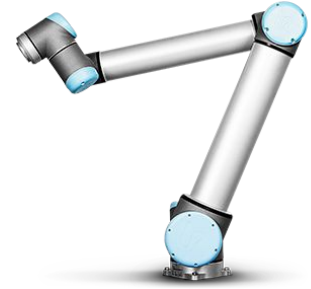
Missão:

“...apoiar o desenvolvimento, distribuição, e adoção de software de código aberto para uso em pesquisa, educação e desenvolvimento de produtos de robótica.”

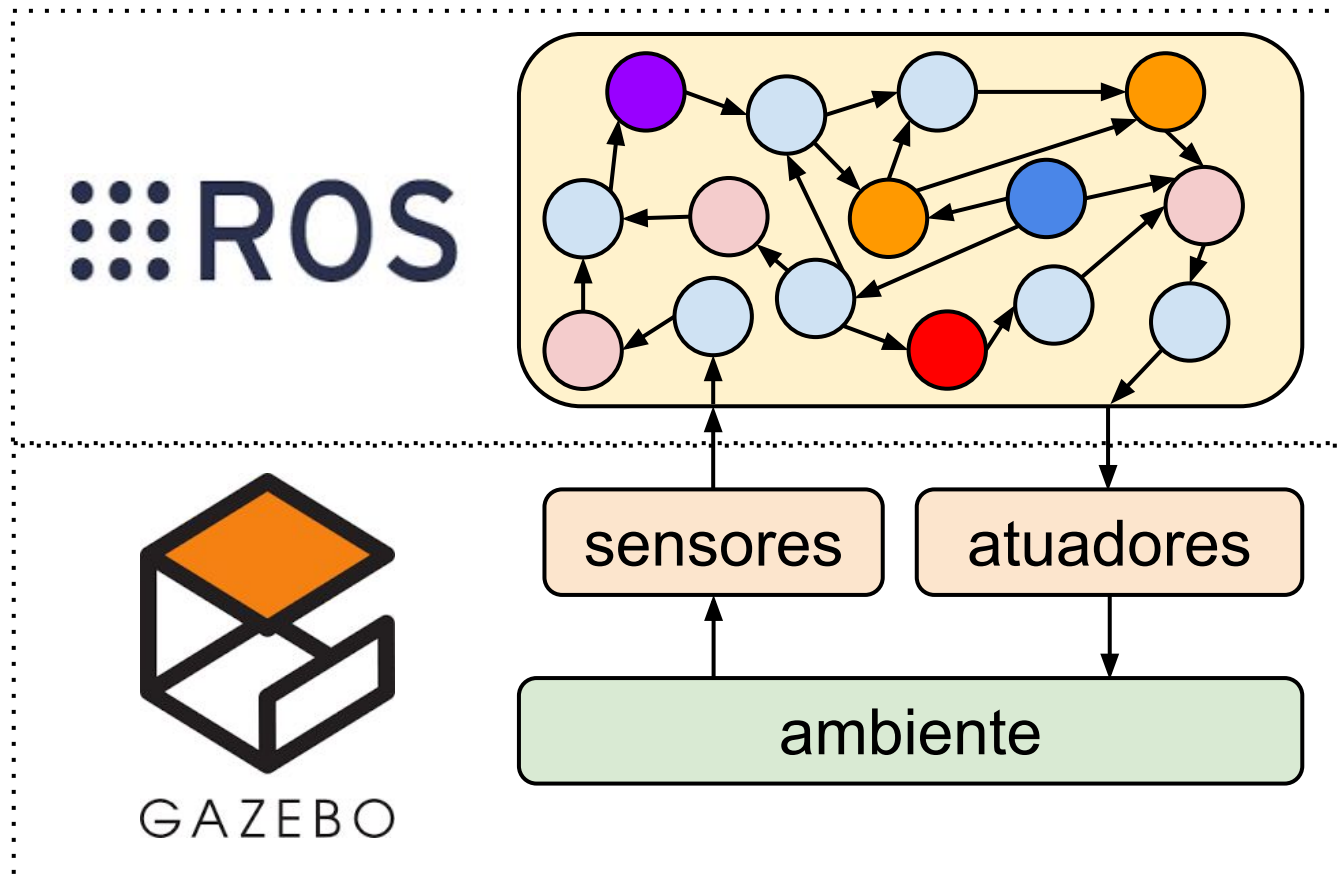


<http://osrfoundation.org>

O problema: robótica é difícil



Nossa proposta: Colaboração, Modularidade e Simulação



Open Source

“Código aberto”

O que é open source?

- Existem apresentações inteiras sobre isso...

<https://www.youtube.com/watch?v=n2GhS-u-5FA>

Para quem é essa apresentação?

Quem quer contribuir com open-source mas não sabe por onde começar.

Formas de colaborar

- Código
 - Adicionar funcionalidades
 - Consertar bugs

Formas de colaborar

- Código
 - Adicionar funcionalidades
 - Consertar bugs
- Conteúdo
 - Imagens
 - Tradução
 - Documentação
- etc...

O que você ganha colaborando?

- Experiência de trabalho em equipes grandes
- Experiência no mundo real
- Desenvolve código de qualidade para mostrar no CV
- Trabalha ao lado dos melhores do ramo
- E até dinheiro!

O que você precisa para começar?

- Muito específico de cada projeto, mas algumas idéias gerais:
 - Inglês*
 - Não ter medo da linha de comando
 - Controle de versão

Pré-requisitos para essa apresentação

- Programação intermediária
- Linux (linha de comando)

Bônus

- Mercurial (é parecido com Git)
- C++

Como contribuir

Em 12 passos

1. Escolha um software

Ótimas listas:

- Outreachy
 - 2x ao ano
 - Inscrições até 22/03/2016
 - <https://wiki.gnome.org/Outreachy/2016/MayAugust>
- Google Summer of Code
 - 1x ao ano
 - Inscrições até 25/03/2016
 - <https://summerofcode.withgoogle.com/organizations/>



1. Escolha um software



Open Source Robotics Foundation



GAZEBO

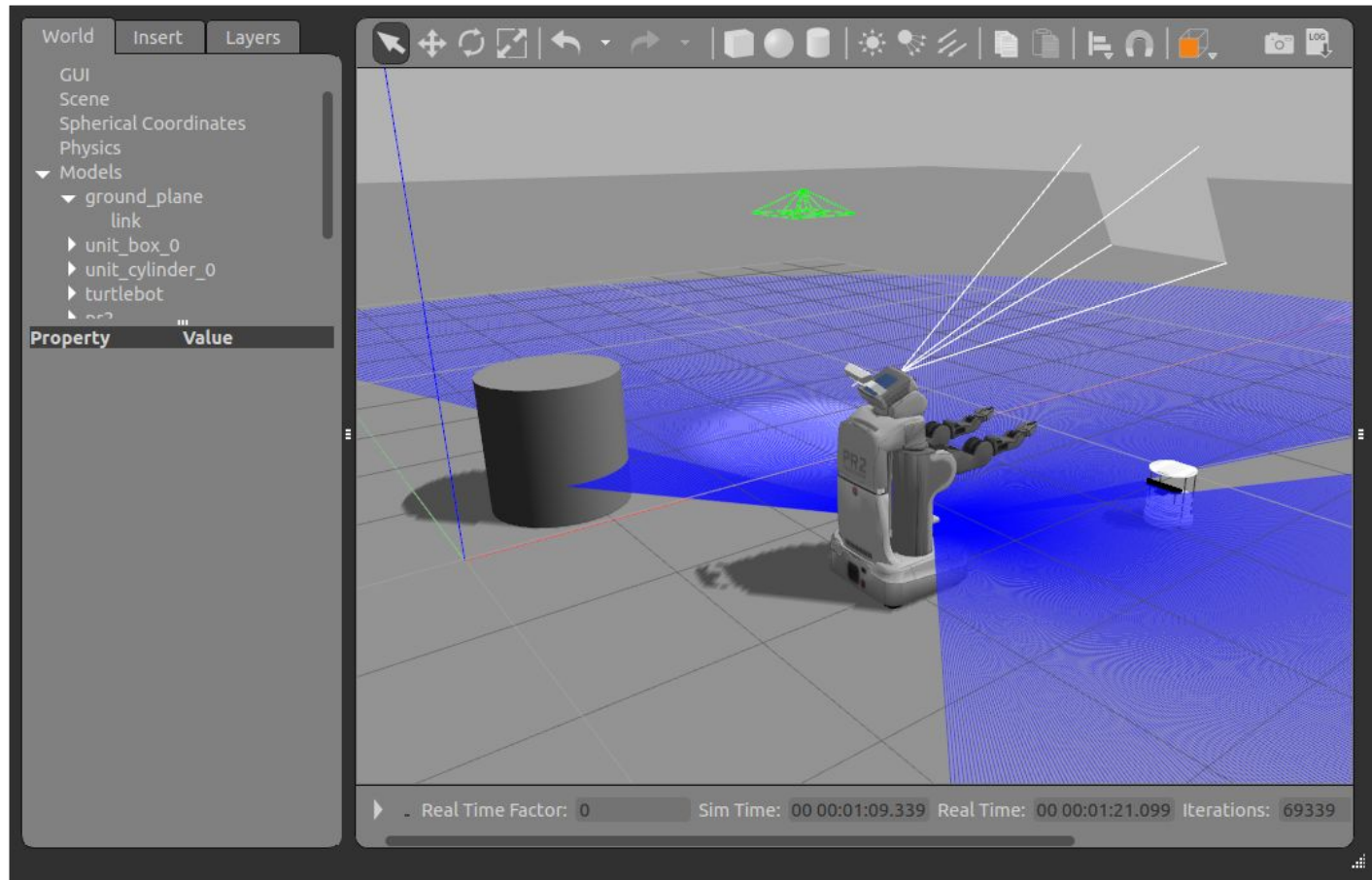


2. Familiarize-se com o projeto

- Se ainda não é usuário, use!
- Faça tutoriais
- Rode exemplos
- Faça perguntas
- Use mais!
- Mais!



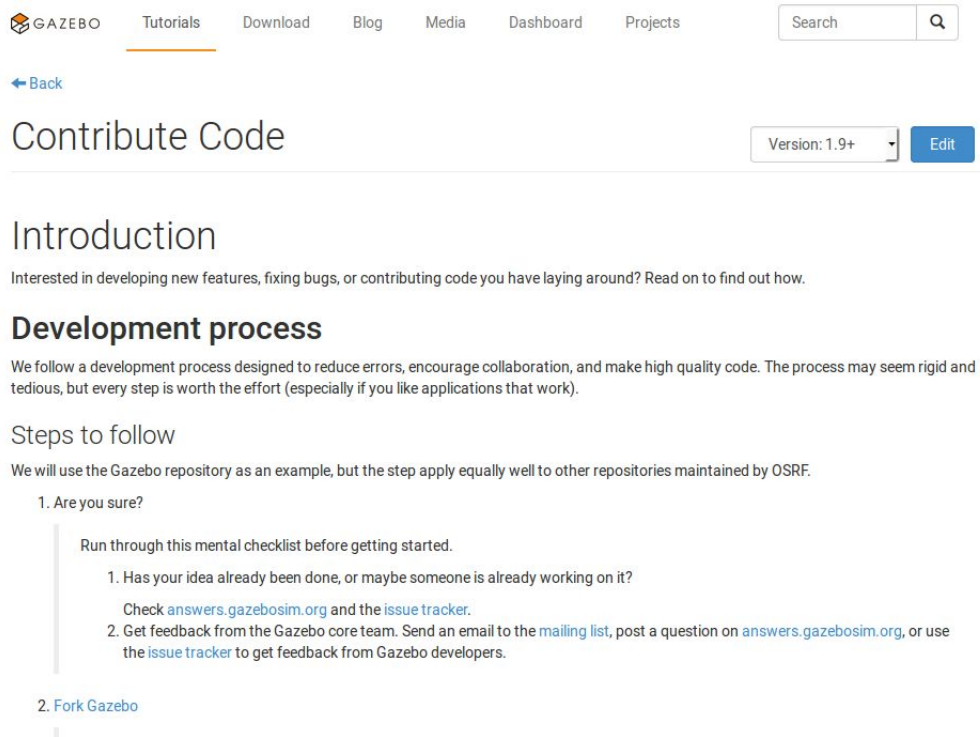
2. Familiarize-se com o projeto



3. Descubra o processo de contribuição

- Procure em sites
- Fóruns
- Canais do IRC
- Mailing lists
- ...

3. Descubra o processo de contribuição



The screenshot shows the 'Contribute Code' page on the Gazebo website. The navigation bar includes links for GAZEBO, Tutorials (highlighted), Download, Blog, Media, Dashboard, and Projects. A search bar is located on the right. Below the navigation bar, there is a 'Back' link and a 'Contribute Code' heading. A dropdown menu shows 'Version: 1.9+' and an 'Edit' button. The main content area starts with an 'Introduction' section, followed by a paragraph about contributing code. Then, there is a 'Development process' section with a paragraph explaining the process. Below that is a 'Steps to follow' section, which includes a list of steps: 1. Are you sure? (with a sub-section 'Run through this mental checklist before getting started.' containing two numbered items) and 2. Fork Gazebo.

GAZEBO Tutorials Download Blog Media Dashboard Projects Search

← Back

Contribute Code

Version: 1.9+ Edit

Introduction

Interested in developing new features, fixing bugs, or contributing code you have laying around? Read on to find out how.

Development process

We follow a development process designed to reduce errors, encourage collaboration, and make high quality code. The process may seem rigid and tedious, but every step is worth the effort (especially if you like applications that work).

Steps to follow

We will use the Gazebo repository as an example, but the step apply equally well to other repositories maintained by OSRF.

1. Are you sure?
Run through this mental checklist before getting started.
 1. Has your idea already been done, or maybe someone is already working on it?
Check answers.gazebosim.org and the [issue tracker](#).
 2. Get feedback from the Gazebo core team. Send an email to the [mailing list](#), post a question on answers.gazebosim.org, or use the [issue tracker](#) to get feedback from Gazebo developers.
2. [Fork Gazebo](#)

- Conta no Bitbucket
- Fork do repositório
- Instalar da fonte
(geralmente diferente da instalação como usuário)

4. Escolha como contribuir

- Lista de problemas (issue tracker)
- Pergunte se alguém já começou a fazer o que você quer fazer, se têm dicas, sugestões...



4. Escolha como contribuir

OpenSourceRoboticsFoundation / Simulation / gazebo

Issues

FILTERS: **All** Open My issues Watching

Issues (1–25 of 1,871)

Title	T	P
#1871: Highlight surfaces when they touch	gui::ModelEdi...	↑
#1870: Document the need to call PhysicsEngine::InitForThread before using RayShape::GetIntersection.	documentation	↓
#1869: INTEGRATION_worlds_installed test is slow in jenkins	testing	↑
#1868: Add example integration test and tutorial for recovering shock response spectrum of a system	physics	↓
#1867: Add function to get compression / tension force across a link element. E.g. Link::GetPrincipalStress, Link::GetShearStress, etc.	physics	↓
#1866: Add ability to get bending moment across a link member: e.g. Link::GetBendingMoment	physics	↓

ACTIONS

- Clone
- Create branch
- Create pull request
- Compare
- Fork

NAVIGATION

- Overview
- Source
- Commits
- Branches
- Pull requests (32)
- Issues (859)

- [Issue #1806: Building Editor: Automatically add visibility layers to floors](#)

5. Familiarize-se com o código fonte

- Organização do código
- Toolchain (ferramentas)
- Tente mudar qualquer coisinha pra ver se funciona



5. Familiarize-se com o código fonte



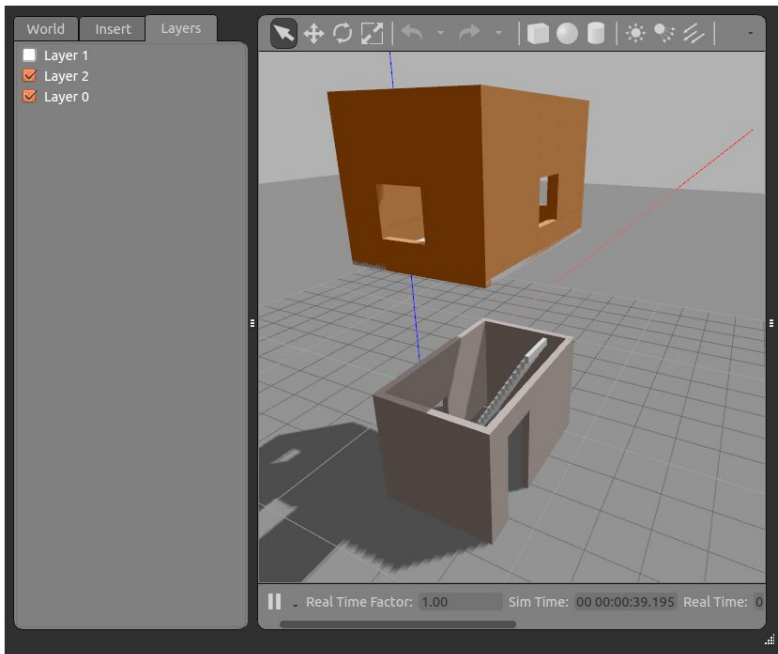
- Fonte: gazebo/ (geralmente `src/`)
- Pastas: `physics`, `rendering`, `gui`, `math`, `transport...`
- Banana!

6. Fun time!

- Crie um “branch” separado
- grep’e muito!
- Não tente entender tudo!
- Faça as modificações necessárias até satisfiteit@
- commit -> push



6. Fun time!



7. Testes automatizados

- Escreva testes para que o problema não volte no futuro
- Verifique que não quebrou nenhum teste existente
- Certifique-se de que funciona em todas as plataformas



7. Testes automatizados

- GTest, QTest
- [Jenkins](#)

8. Estilo do código

- Tente mudar somente o necessário pra facilitar a revisão
- Tente manter o estilo atual
- Siga o estilo oficial, se houver
- Avisos (warnings) do compilador



8. Estilo do código

- `sh /tools/code_check.sh`
- [Jenkins](#)

9. Documentação

- Documentação auto-gerada
- Comentários no código
- Exemplos de como usar
- Tutoriais, vídeos...



9. Documentação

- Doxygen
- Exemplo de “[world](#)”, “[plugin](#)”
- [Tutorial](#)

10. Pull request

- Veja outros pull requests como exemplo da cultura do projeto
- Explique bem o que foi feito
- Facilite ao máximo a vida de quem vai revisar
- Se possível, separe em pedaços menores



10. Pull request

- [Pull requests abertos](#)

11. Revisão

- Responda comentários
- Faça mudanças necessárias



11. Revisão

- Esperar ter 2 aprovações
- Dependendo da complexidade e do tamanho da fila, pode levar entre horas e semanas

12. Merge!

- Seu código passa a fazer parte do código fonte oficial
- Se tem release, seu código aparecerá na próxima versão



12. Merge!

- Uma hora chega lá...

É isso!

Obrigada!

Perguntas?