



## Como contribuir com projetos de software livre

---

**Luciana Fujii Pontello**

# Software livre?



Por que contribuir?

---

## Software livre

- Liberdade não preço
- Linux, Ubuntu? Sim, mas não só
- Quatro liberdades básicas



Por que contribuir?

---

## Quatro liberdades

- **0** - A liberdade de executar o programa, para qualquer propósito.
- **1** - A liberdade de estudar como o programa funciona, e adaptá-lo como você quiser. (Acesso ao código-fonte é um pré-requisito para esta liberdade.)
- **2** - A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo.
- **3** - A liberdade de distribuir cópias do seu programa modificado. (Dessa forma toda a comunidade pode se beneficiar).



Por que contribuir?

---

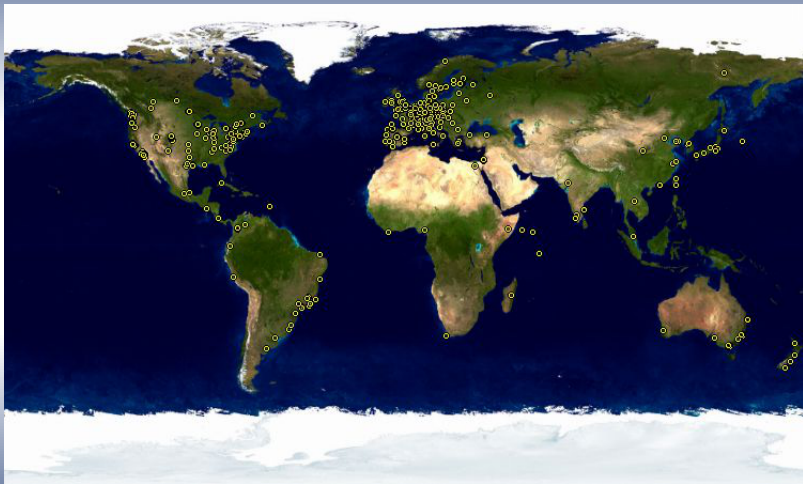
## Por que contribuir?

- Por que você acredita em software livre!
- Devolver o que se recebe
- Melhorar seus conhecimentos
- Porque é divertido
- Coçar aquela coceira
- Ter reconhecimento
- Carreira profissional
- Experiência internacional
- Comunidade, pessoas



Por que contribuir?

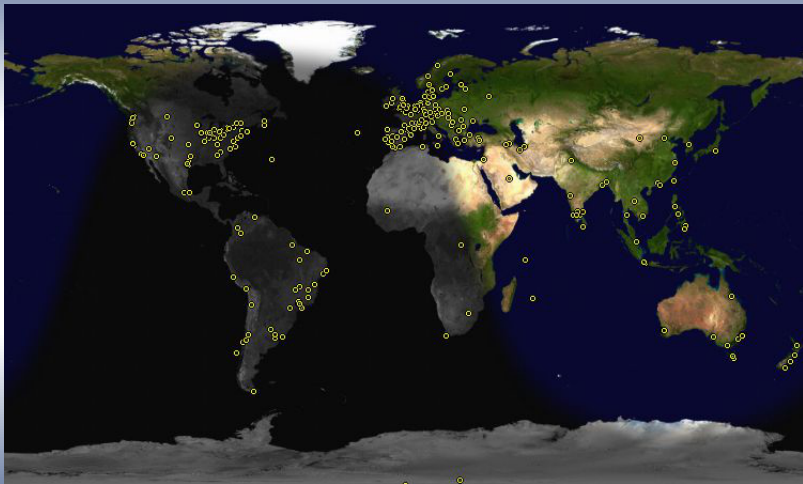
## DDs no mundo





Por que contribuir?

## GNOME no mundo





Por que contribuir?

---

## Contribuição no Brasil

- Muitos usuários, mas poucos contribuidores em geral
- Entre as contribuições, poucas são de código
- Cultura de consumir e não de produzir





## Minha experiência

- 2002/2003 - Usuária e defensora de software livre
- 2004 - Organização do primeiro Encontro Mineiro de Software Livre, trabalhando em projeto de software livre na UFMG
- 2005...
- 2006...
- 2007...
- 2008...
- 2009...
- 2010 - Trabalho com GStreamer, GNOME Outreach Program
- 2011 - Mantenedora do Cheese, mentora do GSOC, comecei a trabalhar com desenvolvimento de software livre



Por que contribuir?

---

## O difícil é começar

- Primeira experiência foi difícil, conhecer o ambiente e o código até conseguir fazer alguma coisa útil
- Depois do primeiro passo as idéias aparecem, é muito mais fácil e rápido consertar pequenas coisas
- Encontrar o que se gosta de fazer



Por que contribuir?

---

## Várias possibilidades de contribuições

- Desenvolvimento
- Tradução
- Design
- Documentação
- Organização de eventos
- Usabilidade
- Desenvolvimento web
- Teste
- Marketing
- Patrocínio



# Ferramentas de desenvolvimento

Um dos principais desafios é conhecer como os projetos são organizados. Ferramentas de software livre:

- controle de versão (git, bzt, svn, cvs)
- bug tracking (bugzilla...)
- IRC, e-mails



## Bug trackers

- Controle dos bugs do software
- Bugzilla
- Bom lugar para encontrar problemas para trabalhar
- Pode ser usado para controlar os patches
- Usado para discussões de certos problemas
- Reportar bugs e testar são boas formas de começar a se envolver



## Dicas para reportar bugs

- Procurar se o bug já foi reportado
- Tentar verificar se já foi consertado numa próxima versão
- Descrever como e quando acontece o problema
- Colocar versão dos softwares utilizados
- Acompanhar as respostas



## Comunicação

- Cada projeto tem sua dinâmica
- É importante saber onde encontrar as pessoas:
  - lista de e-mails
  - irc
  - bugzilla?
- Lembre-se que a maioria das pessoas também é voluntária
- Mostrar que está interessado em aprender e que procurou a informação (leia a documentação)
- As vezes é necessário insistir



## Controle de versão

- Ajuda você a manter seu programa, mesmo que sozinho
- Cada projeto funciona de um jeito
- Controle de versão centralizado (CVS, SVN)
- Controle de versão distribuído (Git, Bazar)
- Git:
  - Várias facilidades, como branches
  - Formata patches para serem colocados no bug tracker
  - Servidores grátis: GitLab, GitHub





# Desafios

- Começar é difícil
- Cultura do projeto
- Dependência de bibliotecas muito novas
- Compilar o código a primeira vez pode ser custoso
- Tecnologias diferentes
- Pessoas não amigáveis



## Como facilitar pra você

- Acostume-se às ferramentas usadas, elas também vão te ajudar no dia-a-dia.
- Tente instalar o software pelo código fonte.
- **Aprenda inglês.**
- Participe de listas de discussão e IRC, isso ajuda a entender melhor como a comunidade funciona.
- Peça ajuda! Todo mundo já foi iniciante em algum ponto.
- Ajude também: por mais que seus conhecimentos sejam pequenos, você pode ajudar pessoas iniciantes também e aprender com isso.
- Não desanime!



## Incentivos a contribuição

- Bolsas
- Marcar bugs para iniciantes
- Canais de IRC para iniciantes
- Mentores



## Projetos para incentivo a contribuições

- OpenHatch
- Google Summer of Code
- Outreachy
- GNOME Love
- KDE Junior Jobs
- Google Code in
- etc



## OpenHatch

- Objetivo de ajudar desenvolvedores a encontrar comunidades, ferramentas e educação para o que precisam
- Busca por tarefas em projetos para contribuir
- Definido por linguagem, dificuldade, label e projeto
- Educação: missões para aprender a tecnologia necessária para contribuir para projetos de software livre. Exs: usar git, usar shell, usar diff e patch
- Projetos de software livre cadastram no OpenHatch para se tornarem mais amigáveis a novos participantes
- <https://openhatch.org>



## Google Summer of Code

- Voltado para estudantes de curso superior ou pós graduação
- Acontece durante o verão do hemisfério norte
- O Google fornece uma bolsa para o estudante e um valor para o projeto organizador
- Projetos aprovados: GNOME, KDE, GStreamer, Mozilla, Wikimedia Foundation, etc sempre participam
- Os projetos dão idéias do que pode ser feito
- O estudante propõe um projeto para trabalhar durante o período
- Os estudantes são aprovados com um projeto
- O estudante trabalha com um mentor



# Outreachy



- Surgiu com a idéia de incentivar a participação de mulheres no GSOC
- Acontece duas vezes por ano (no período de férias do hemisfério sul e norte)
- Apenas para mulheres
- Não é exclusivo para estudantes
- Não é só "Code"
- Os projetos podem ter mais interação com a comunidade
- Assim como o GSOC, trabalha com bolsa e mentor



## Google Code In

- Voltado para estudantes pré-universitários (13-17 anos).
- Acontece no nosso verão.
- Alguns ganhadores ganham passagens para o Google em Mountain View.
- Projetos aprovados: alguns dos que participam do GSOC (GNOME, KDE...).
- Os projetos definem tarefas de diversas áreas e a dificuldade.
- Inclui tarefas de programação, documentação, marketing, acessibilidade entre outros.





## Primeiro patch no GNOME

- Decida o que você quer fazer, bugs marcados como GNOME Love são boas idéias
- Baixe o software pelo git
- Construa e rode
- Desenvolva, teste... peça ajuda se precisar
- Comemore!



## Primeiro patch no GNOME

- Crie seu patch com git format-patch
- Abra um bug no bugzilla sobre o problema e coloque seu patch
- Responda as perguntas e modifique seu patch de acordo com os comentários
- Se precisar insista!
- Pode ser difícil descobrir o melhor jeito de fazer algumas modificações, não desista!



## Contribuir no github

- Forke o projeto pela interface do github
- Clone o seu fork no seu computador (\$ git clone <endereço>)
- Crie um branch para trabalhar nas suas modificações (\$ git branch -b newbranch)
- Faça commit das suas modificações (\$ git add <file> \$ git commit)
- Faça push para o seu branch (\$ git push newbranch)
- Mande um pull request para o projeto



## Just do it!

- Não deixe para amanhã ou pra daqui a 7 anos.
- Participar de um programa é um bom começo e também um bom objetivo.
- Como fazer uma aplicação ser aceita no GSOC? Contribua antes!
- Peça ajuda.



# Links

- <https://openhatch.org>
- <http://live.gnome.org/JoinGnome>
- <http://br.gnome.org/GNOME BR/Colabore>
- <http://community.kde.org/Getinvolved>
- <http://www.debian.org/devel/join/>
- <https://developers.google.com/open-source/soc/?csw=1>
- e-mail: [luciana@fujii.eti.br](mailto:luciana@fujii.eti.br)