

# Contribuir em projetos open source (ou como ganhar competências sem ter que ir à escola)



Jorge Gustavo Rocha, Joana Simões  
**OSGeo PT**

SASIG 2017  
Biblioteca Municipal Almeida Garrett, Porto  
22 de novembro de 2017

# Inquérito

- Quem usa software open source?
- Quem contribuiu para projetos open source?
  - Porque não?

# Objetivo

- Começar a contribuir
  - Porque é importante para o projeto...
  - ...porque é muito importante para desenvolver competências sociais, comportamentais e técnicas
- Como começar, por onde começar?
  - Passar um pouco da nossa experiência, para que a vontade de participar não se esboroe nas primeiras dificuldades

# Motivação

- Qual a diferença entre “código open source” e “código closed source”?
  - “código **open** source”
  - “código **closed** source”
- A diferença está em “open” versus “closed”.

# Motivação

- A opção “open” não é uma questão tecnológica
- É uma forma de estar (radicalmente diferente).
  - Não se muda a forma de estar de um momento para o outro.
- O código não é tão relevante quanto alguns julgam.
- É código é importante na medida em que é o resultado de um processo.
- No open source damos muita importância ao processo.

# Motivação

*“When I look back over the last 25 years, in some ways what seems most precious is not what we have made but how we have made it and what we have learned as a consequence of that.*

*The power of what you have learned enables you to do the next thing and it enables you to do the next thing better.”*

Jony Ive

# Parte I - O processo

- Não se aprende na escola.
- É um percurso que leva o seu tempo e não pode ser percorrido sozinho, dentro da toca.
- Não é uma passadeira vermelha: é preciso trabalhar e ultrapassar alguns mitos (e algum ego).
- Vamos realçar muitas vezes nesta apresentação que **o software é algo eminentemente fruto de uma atividade colaborativa**.

# Os génios da informática

Lamento, mas não há génios da informática.



# Os génios da informática

- Acreditar que alguém vai para a cave e sai de lá com uma app maravilhosa que vai mudar o mundo é... ficção científica.
- Indiquem um software que usem e que foi escrito por uma pessoa.
- Mas há gente que sabe mais do que eu? Claro! Mas são tão humanos como nós e tiveram o seu tempo de aprendizagem. Provavelmente começaram há mais tempo. Só isso.
- “Não vou conseguir nunca ser um génio da programação...”.
  - Ainda bem. Não há unicórnios, nem sereias, nem génios da informática.

# Ego

- Ok, não sou génio, mas não quero que vejam as minhas fragilidades. As pessoas são inseguras. Quem não é?
- Querem passar por pessoas inteligentes, não querem mostrar as suas fragilidades, muito menos em público.
- Querem só mostrar o que fizeram quando está tudo pronto e polido.  
**Errado!**
- Querem começar sempre com um projeto privado e abrem depois ao mundo. **Errado!**
- Frequentemente pedem para apagar o histórico de todo o projeto.  
**Errado!**
- É preciso errar e aprender com os erros para nos tornarmos mais sábios e produtivos. Não ter medo de errar.

# Que és tu para me criticar?

- Temos “alguma” dificuldade em aceitar críticas, certo? Quem não tem?
- Na verdade, criticar e aceitar a crítica é uma qualidade que se trabalha.
- Contudo, a revisão do código está no âmago do open source.
- As críticas, a imposição de certas regras, de um certo “coding style” são para serem discutidas, obviamente, mas depois devem ser seguidas com bastante rigor. É preciso treinar isto (aceitar regras: não há espaço para os egos pessoais).
- Manter o *low profile*. Não queiram começar a assumir responsabilidades num projeto. À medida que se aumentam as responsabilidades, perdem-se oportunidades de aprender.

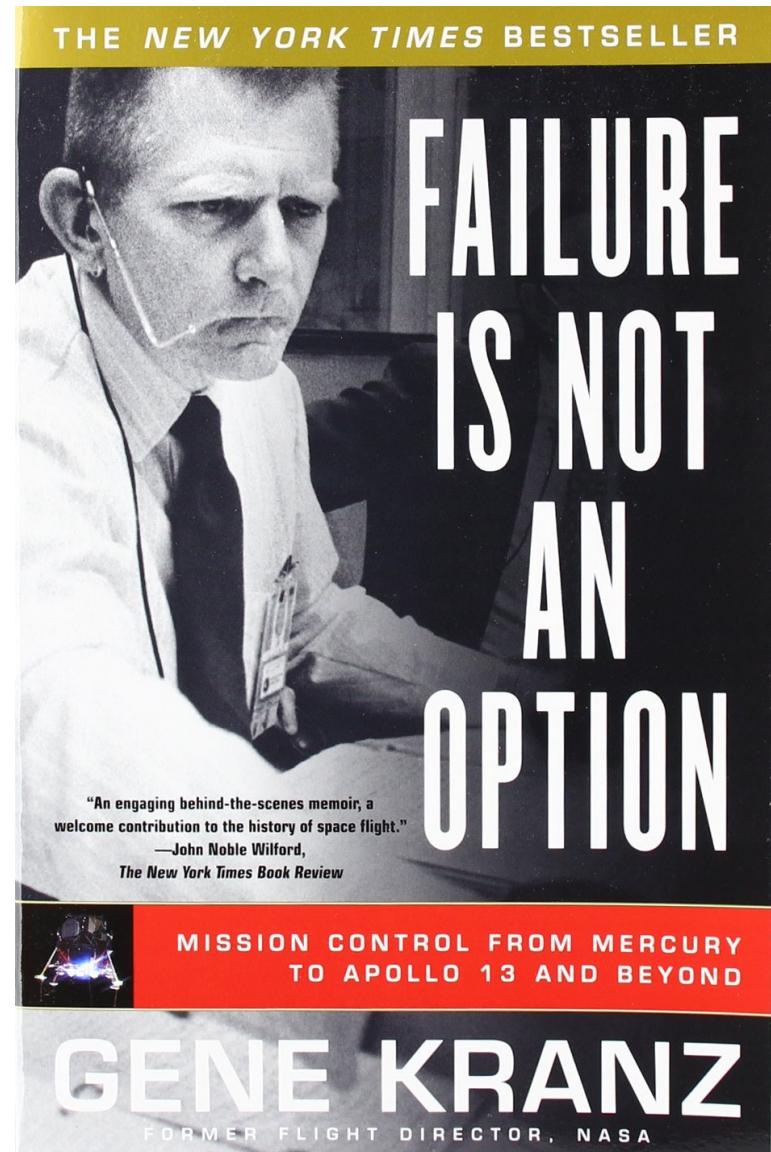
# A minha vida é um conto de fábulas



# A minha vida é um conto de falhas

O slogan “failure is not an option” fica muito bem à NASA.

É um slogan dos anos 60.



# A minha vida é um conto de falhas

- No desenvolvimento de software, só não errou quem nunca tentou. E quem nunca tentou, não aprendeu.
- O caminho faz-se em ciclos de tentativa e erro. Tentar e não sair bem não tem nada de mal. Aprende-se. Faz parte do processo.
- Prever que as coisas possam falhar ajuda também a aceitar melhor o erro. Ter sempre o sucesso e o erro como resultados possíveis, para não criar um situação de desânimo.
- Num projeto open source quando há determinadas falhas, não se faz uma caça às bruxas para ver quem é o culpado. Geralmente faz-se uma análise cuidadosa para ver como se pode evitar que o mesmo erro volte a ocorrer.
- O erro é admissível (cair no mesmo erro repetidamente é que já não é muito pedagógico)

# A minha vida é um conto de falhas

- Ressalva: Não usem este método de tentativa e erro para escolherem cogumelos selvagens.



# A minha vida é um conto de falhas



Miguel Nunes @miguelbnunes · 8 de nov

CEO da Rovio em palco. Antes do sucesso de **Angry Birds** foram criados 51 jogos de pouco sucesso pela empresa. Insistir compensa, guys. #WebSummit



1



3



# A minha vida é um conto de falhas

- Ainda melhor do que aprender com os nossos erros, é aprender com os erros dos outros. Por isso, faz sentido manter a história de um projeto. Faz todo o sentido que lá estejam os erros cometidos.
- Isso faz-se, não só para o código, mas também, por exemplo, na organização de eventos. A OSGeo mantém as notas das **lessons learned** para cada um dos eventos FOSS4G.

# Acelerar e diminuir o ciclo de tentativa e erro

- Reagir rápido.
- Depois de falhar, partir imediatamente para uma nova tentativa.
- À medida que treina este ciclo de tentativa erro, ganha-se mais sensibilidade para diminuir a duração do ciclo.
- Depois de ser “criticado”, mostrar logo que se percebeu e aceitou a crítica e que temos capacidade de apresentar ou discutir soluções alternativas.
- Ao reagir rápido, garante-se a atenção de quem reviu o nosso código. E ter esse feedback é muito bom.

# Desenvolver competências específicas

- Ferramentas, plataformas e boas práticas
- Quando estudar e quando sair da toca?

# Ferramentas, plataformas e boas práticas

- Poder expor todas as componentes de um projeto de um forma clara, permitir que ponham questões, identifiquem bugs, etc, é algo muito eficaz. Acelera e melhora imenso o desenvolvimento dos projetos.
- As ferramentas melhoraram imenso nos últimos anos e essas melhorias depois refletem-se na melhoria dos projetos de software.
- O git e as plataformas como o github, gitlab, por exemplo, são sistema de gestão de código que garantem estes aspetos que temos estado a falar.
  - Permitem isolar as contribuições de cada um, suportam a discussão de cada uma das alterações, mantém toda a história do desenvolvimento desde o início, pode-se reverter para alteração, etc.
- Em informática **a única coisa que não muda, é que as coisas estão sempre a mudar.**
  - Além das ferramentas e das plataformas, há cada vez mais partilha de boas práticas, que são importantes porque nos ajudam a manter o desenvolvimento mais adaptado às alterações constantes na informática.

# Quando estudar e quando sair da toca?

- Temos que desenvolver as nossas capacidades técnicas. Há fantásticas plataformas de elearning para todos os temas e mais alguns. Tem que se fazer o trabalho de casa.
- Mas é preciso sair da toca para evoluir. Para evoluir mais rapidamente.
- Mesmo que fossem génios (que não existem) sem comunicarem e sem estarem bem integrados numa equipa não servem para nada.
- Outro mito que convém desmistificar: um geek (no mau sentido do termo, que não comunica, que se refugia no seu mundo e que supostamente sabe tudo sobre computadores) não tem lugar em nenhuma equipa de software. **O software é o resultado de um processo colaborativo.**
- Há que saber onde está o meio termo entre não perguntar nada e perguntar tudo por tudo e por nada.
- Mas quando estou num projeto de software, tenho mesmo de perguntar: faz sentido o que estou a fazer? E como a estou a fazer? Há mais alguém a trabalhar nisto?
- Mostrar o que estamos a fazer e olhar também para o que os outros estão a fazer.
- Ir buscar ideias fora.

# Parte II – Como contribuir

- Começar a contribuir
  - Porque é importante para o projeto...
  - ...porque é muito importante para desenvolver competências sociais, comportamentais e técnicas
- Como começar, por onde começar?
  - Passar um pouco da nossa experiência, para que a vontade de participar não se esboroe nas primeiras dificuldades
- Prometido: **Contribuição para projetos open source: guia de boas práticas para as entidades públicas e privadas**

# Stack Overflow

- Pode-se começar sozinho. E pode-se começar já a praticar. Não esperar por amanhã.
- Ganhar reputação. Fazer coisas pequenas, para podermos ter mais facilmente e rapidamente um retorno positivo.
- Vantagens:
  - Não preciso de me focar num projeto. Posso usar o Stack Overflow para as minhas questões em vários domínios.
  - Não preciso de começar ajudando os outros. Posso começar perguntando (**e aprendendo a perguntar**).

# Que projeto escolher?

- Um software FOSS4G (QGis, proj4, GeoNetwork, Geoserver, MapServer, PostGIS, etc), ou outro software open source que usem: Wordpress, Drupal, Postgresql, etc...
- Escolher algo que seja do meu interesse, preferencialmente que use no dia a dia.

# Conhecer a comunidade (e dar-me a conhecer)

- Começar na próxima semana.
- Inscrever-me na lista de email, ver que wiki/blogs existem, contas no twitter, etc.
- Conhecer as pessoas e envolver-se com a comunidade.
- Dar-se a conhecer: mostrar o interesse e receptividade em começar a colaborar.
- Descobrir algo em que possa ajudar.
- E cuidado com o ego: se nos pedirem para descascar cebolas, começamos a descascar cebolas.

# Participar em eventos

- **SASIG 2017** - Dia 20 de novembro: workshops para utilizadores e agora também para developers.
- **Date With Data** - Todos os meses. Começaram há uns anos e continuam. O próximo será o #23. Excelente ambiente, onde se mistura o *hands on* com apresentações convidadas.
- **QGIS Hackfest** - Madeira, 21 a 25 de fevereiro, 2018. A nata da nata do QGIS está na Madeira. Oportunidade de começar a entrar no desenvolvimento do QGIS.
- **OSGeo Code Sprint** - Bonn, 18 a 25 de março, 2018. Code sprint anual da OSGeo. O ambiente é fantástico (e o espaço espetacular).
- **Dados Abertos@Águeda** - Também já fizeram duas ou três edições e na última entregaram prémios (1º prémio: 500,00 €)
- OpenStreetMap - O OpenStreetMap é um ótimo projeto de dados abertos, cuja participação não exige programação. Começando numa *party*, conhecendo outras pessoas, o processo de iniciação é muito mais rápido.
- FOSS4G-Europe – 16 a 22 de julho, 2018, Guimarães
  - 16: Workshops
  - 17 a 19: Conferência
  - 20, 21 e 22: Workshops e Code Sprint

# Organizar eventos

- As comunidades de developers estão disponíveis para participar em eventos e acolher novos contribuidores. É um serviço que se presta e é muito reconhecido pelos developers.
- QGIS **call for venues**
- Organizar *Mapping parties* do OpenStreetMap
  - Super simples: arranjar uma sala, wifi (cada vez menos preciso) e convidar duas ou três pessoas mais experientes.

# Desenvolver os hard skills

- Desenvolver competências tecnológicas: ferramentas e programação
- Submeter contribuições
- Exemplos:
  - PostGIS Raster workshop, by Luís Calisto & André Mano

# PostGIS Raster Workshop

Step 1: Fork

Step 2: Clone

```
$ git clone git@github.com:jgrocha/workshop-postgis-raster.git
```

```
$ cd workshop-postgis-raster
```

Step 3: New branch

```
$ git checkout -b geoserver
```

```
* [new branch]           geoserver -> geoserver
```

# PostGIS Raster Workshop

- Fazer as alterações localmente
  - Texto
    - Comandos/operações bem testadas
  - Imagens de suporte
- Testar, testar

# PostGIS Raster Workshop

```
$ git status
```

```
On branch geoserver
```

```
Changes to be committed:
```

```
  (use "git reset HEAD <file>..." to unstage)
```

```
modified: README.md
```

```
new file: doc/geoserver.md
```

```
new file: doc/img/add_raster_data_source.png
```

```
new file: doc/img/geoserver_layer_preview.png
```

```
$ git commit -m "Publish postgis raster on GeoServer"
```

```
[geoserver acc1538] Publish postgis raster on GeoServer
```

```
 4 files changed, 135 insertions(+), 3 deletions(-)
```

```
  create mode 100644 doc/geoserver.md
```

```
  create mode 100644 doc/img/add_raster_data_source.png
```

```
  create mode 100644 doc/img/geoserver_layer_preview.png
```

# PostGIS Raster Workshop

```
$ git remote
origin

$ git remote add upstream git@github.com:lcalisto/workshop-postgis-raster.git

$ git remote -v
origin git@github.com:jgrocha/workshop-postgis-raster.git (fetch)
origin git@github.com:jgrocha/workshop-postgis-raster.git (push)
upstream git@github.com:lcalisto/workshop-postgis-raster.git (fetch)
upstream git@github.com:lcalisto/workshop-postgis-raster.git (push)

$ git push origin geoserver
Counting objects: 8, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 231.09 KiB | 0 bytes/s, done.
Total 8 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To git@github.com:jgrocha/workshop-postgis-raster.git
 * [new branch]      geoserver -> geoserver
```

# PostGIS Raster Workshop

# Compromisso

- Quem alinha estar aqui no próximo ano a apresentar as suas contribuições para um projeto open source?