

01 - Ciclo de vida de um bug

Desenvolvimento aberto 2018/2

Igor Montagner

Neste roteiro trabalharemos no workflow padrão para contribuir com projetos hospedados no Github (mas que também serve para projetos git em geral). Antes de começar cada aluno deverá localizar no github da disciplina a issue correspondente a criação de seu avatar.

Nosso fluxo de trabalho será baseado em três grandes partes. Na primeira será criada uma cópia do repositório `igordsm/dev-aberto` onde faremos todas as mudanças necessárias. Na segunda enviaremos nossas modificações para o repositório original usando um *Pull Request*, que é um pedido de aceite das mudanças de um *fork* no repositório original. Por último, atualizaremos nosso *fork* com modificações enviadas pelos colegas.

Alguns pontos a serem destacados no fluxo de trabalho acima:

1. Mesmo que um usuário não tenha acesso ao repositório original ele pode trabalhar em sua cópia e somente quando tudo estiver pronto enviar suas modificações ao repositório original.
2. É necessário que um desenvolvedor do projeto original “se responsabilize” pelas modificações externas aceitas.
3. A aba *Pull requests* permite que os desenvolvedores discutam as propostas de modificações e as melhorem. Todo commit feito após a criação do PR é incluído e pode ser testado por qualquer um.

Parte 1 - Criando uma cópia local

Iremos começar nosso fluxo de trabalho criando um *fork* do repositório `igordsm/dev-aberto`. Todas nossas modificações serão feitas no nosso *fork* em um branch separado (o mais correto é sempre usar um branch diferente para cada issue). Desta maneira nossas modificações ficam completamente isoladas do código original e podemos testá-las lado a lado com o código original.

Primeiro, crie o *fork* via interface do Github. Depois, clone seu fork e crie um novo branch chamado *issue-X*, onde *X* é o número da sua issue no projeto

original.

```
$ git checkout -b issue-X
```

Para garantir que você está no diretório do seu fork, execute o comando

```
$ git remote -v
```

Os endereços mostrados devem ser os do seu *fork*, não os do projeto original.

Com o *fork* criado e estando no branch *issue-X* (você pode checar usando `git branch` e mudar usando `git checkout issue-X`), vamos começar a realizar modificações.

Criando um avatar

Para adicionar um avatar na disciplina basta editar o arquivo *players/player_list.json*. Siga o modelo do avatar já existente criado pelo professor. Imagens interessantes para seu avatar podem ser baixadas [neste site](#). O personagem escolhido deve ser colocado na pasta *players/avatar_data*.

O campo `uuid` deve ser preenchido com um **Universally Unique Identifier**, que pode ser gerado em Python usando o seguinte código.

```
import uuid
print(uuid.uuid1())
```

Ao finalizar a criação do personagem execute `python update_website.py` na raiz do repositório para atualizar o site (localizado na pasta *docs/*). **Confira se o site gerado está funcionando corretamente antes de prosseguir!**. Seu avatar deverá estar presente na lista de participantes de sua cópia local do site da disciplina.

Adicionando uma skill

Com o personagem criado podemos adicionar a skill *Penso logo existo*. Edite o arquivo *players/achievement_list.json* e adicione um novo item. O campo `proof` deve ser preenchido com a futura url do seu avatar (`http://igordsm.github.io/dev-aberto/players.html#nome_github`).

Atualize novamente o site usando o script *update_website.py* e envie suas modificações para seu *fork*. Não se esqueça de mencionar na sua mensagem de commit o número da issue corrigida.

Parte 2 - enviando as modificações para o projeto original

Com nossas modificações já presentes no nosso *fork* é hora de enviá-las para o repositório original. Como somente o professor tem permissão para modificá-lo é necessário criar um *Pull Request* com as modificações feitas na parte 1. Isto é feito na interface do Github. Primeiro, acesse seu *fork* no navegador, localize seu branch *issue-X* e clique no botão “Pull request”.

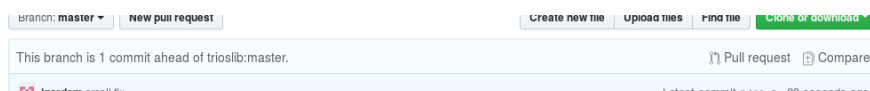


Figure 1: Esta mensagem aparece quando seu *fork* tem commits que não estão presentes no repositório original.

O título de seu Pull Request deverá ser *Fix issue X* e na descrição diga quais modificações foram feitas. O título (e partes da descrição) é parseado automaticamente pelo Github e ao aceitar seu PR a issue é automaticamente fechada.

Ao criar o PR avise o professor. Ele irá conferir se está tudo certo e, em caso positivo, irá aceitar seu PR. O site é automaticamente atualizado.

Parte 3 - atualizando seu *fork*

Um *fork* não é atualizado quando seu repositório original correspondente receber novos commits. Para que isto ocorra é necessário realizar a sincronização *manualmente*. Isto envolve duas etapas. Na primeira, que só precisa ser feita uma vez, é adicionado um novo repositório remoto que aponta para o repositório original. Na segunda baixamos os arquivos deste repositório remoto e os incorporamos aos nossos.

O Github tem uma excelente documentação explicando [como fazer o primeiro passo \(link\)](#) e depois [como sincronizar seu *fork* com o repositório original \(link\)](#).

Após todos alunos criarem seus personagens e fecharem suas issues, execute o segundo passo novamente e confira se o site que você gera localmente está igual ao disponível na internet. Se houver alguma inconsistência chame o professor.