**Nome:** Gabriel Lopes Dos Santos

**R.A.:** 1830918-2

**CPython**

CPython usa um fluxo de trabalho baseado em solicitações pull. Ou seja, o usuário cria um branch no Git, faz suas alterações, envia essas alterações para seu fork no GitHub e, em seguida, cria uma solicitação de pull no repositório oficial do CPython.

**Merge**

Ao mesclar alterações de branches diferentes (ou variantes de uma branches em repositórios diferentes), as duas branches podem conter alterações diferentes em um ou mais arquivos. Eles são chamados de “conflitos de merge” e precisam ser resolvidos manualmente.

Primeiro é necessário verificar quais arquivos têm conflitos de merge, depois tem que editar os arquivos afetados e traze-los ao estado final pretendido.

**controle de solicitações**

**Code Review**

Um dos problemas no processo de desenvolvimento Python é a falta de code reviews. Se você navegar no rastreador de bug, verá que muitos problemas foram corrigidos, mas não podem ser incorporados ao repositório principal de código-fonte, porque ninguém revisou a solução proposta. A revisão de uma solicitação pull pode ser tão informativa quanto fornecer uma solicitação pull e permitirá que você faça comentários construtivos sobre o trabalho de outro desenvolvedor.

Ao revisar uma solicitação pull, o usuário deve fornecer detalhes adicionais e contexto de seu processo de revisão, ou seja não só aprovar mas também deixar comentários.

**controle de solicitações**

Assim que a solicitação pull atingir um estado aceitável, ela será mesclada ou rejeitada. Se for rejeitado, o trablaho ainda é visto, independentemente da solicitação pull ser mesclada. Equilibrar o que entra e o que não entra em Python é complicado e simplesmente não é aceito as contribuições de todos.

Mas a solicitação pull for mesclada, ela irá para o VCS do Python para ser lançada com a próxima versão principal do Python. Também pode ser feito backport para versões mais antigas do Python como uma correção de bug se o desenvolvedor principal que está fazendo a fusão acreditar que isso é garantido.

**Testes**

Escrever testes para o Python é muito parecido com escrever testes para seu próprio código. Os testes precisam ser completos, rápidos, isolados, repetíveis de forma consistente e o mais simples possível.

Uma diferença com o teste comum é que você é incentivado a confiar no módulo [test.support](https://docs.python.org/3/library/test.html#module-test.support). Ele contém vários auxiliares que são feitos sob medida para o conjunto de testes do Python e ajudam a suavizar problemas comuns, como diferenças de plataforma, consumo e limpeza de recursos ou gerenciamento de avisos.

**estratégia de branching**

[O Python Benchmark Suite](https://github.com/python/performance) possui uma coleção de benchmarks para todas as implementações Python.

**Pull requests**

É destinado aos desenvolvedores principais e cobre as etapas necessárias para aceitar, mesclar e, possivelmente, fazer backport de uma solicitação de pull no repositório principal.

Antes de um pull request ser aceito, deve verificar se ele está pronto para entrar na árvore de fontes públicas.

**Integração contínua**

Para afirmar que não há regressões nas [ramificações de desenvolvimento e manutenção](https://devguide.python.org/devcycle/), o python tem um conjunto de maquinas dedicadas usadas para integração contínua. Eles abrangem uma série de combinações de hardware ou sistema operacional. Além disso, cada máquina hospeda vários construtores, um por branch ativo, quando uma nova mudança é enviada para este branch no [repositório](https://github.com/python/cpython) público do [GitHub](https://github.com/python/cpython) , todos os construtores correspondentes agendam um novo build para ser executado o mais rápido possível.

As etapas de versão executadas pelos buildbots são as seguintes:

Verificação da árvore de origem para o changeset que acionou a construção

Compilando Python

Executando o conjunto de testes usando [configurações extenuantes](https://devguide.python.org/runtests/#strenuous-testing)

Limpando a árvore de construção

É responsabilidade do desenvolvedor principal, verificar os resultados de compilação automática depois de enviar uma alteração ao repositório.