**Nome:** Lucas Matheus Nakasima Soares

**R.A:** 1831943-2

**Gerenciamento de Configuração – Python**

**Python**é uma linguagem de programação de alto nível, de script, imperativa, orientada a objetos, interpretada e amplamente utilizada. Possui o CPython, que é a implementação de referência do Python, é um software livre e de código aberto e tem um modelo de desenvolvimento comunitário, assim como quase todas as suas implementações variantes. CPython é gerido pela organização sem fins lucrativos Python Software Foundation.

O ciclo de vida de desenvolvimento em Python usa a nomenclatura de major, minor, micro para lançamentos prontos para a produção. Há uma ramificação para cada versão de recurso, seja lançada ou não, sendo o master branch a branch para o próximo lançamento de recurso, ativo para todos os tipos de mudanças. Em algum momento durante o ciclo de vida de uma versão, um novo branch de manutenção é criado para hospedar todas as atividades de correção de bugs para outras micros versões em uma versão de recurso.

Branchs de manutenção se tornam uma opção para realizar lançamentos de um recurso anterior sendo mantido para correções de bugs. Branches de segurança são as com menos de 5 anos, mas não mais no modo de manutenção, sendo as únicas alterações feitas em um branch de segurança são aquelas que corrigem problemas exploráveis ​​por invasores. A organização de repositórios possui categorias gerais ao qual esperam-se que os repositórios se adequam, sendo eles:

* A implementação de referência de Python e repositórios relacionados
* Implementações de referência de PEPs
* Ferramentas e suporte em torno do CPython e da linguagem
* Repositórios relacionados ao PSF
* Repositórios de infraestrutura PSF

Na questão de regressões nas ramificações de desenvolvimento e manutenção, o Python tem um conjunto de maquinas dedicadas usadas para integração continua. Abrangendo uma serie de combinações de hardware/ sistemas operacionais, sendo que cada máquina hospeda vários construtores, um por cada branch ativo, e quando uma nova mudança é enviada para este branch no repositório público do GitHub, todos os construtores correspondentes agendam uma execução de build. Sendo a responsabilidade do desenvolvedor verificar os resultados de compilação automática.

Para uma solicitação de pull request ser aceito ela deve atender alguns requisitos antes de ser mergiado:

* O remetente assinou o CLA?
* O conjunto de testes foi aprovado?
* A cobertura do código aumentou ou permaneceu a mesma?
* As mudanças são aceitáveis?
* O configure foi renerado?
* O pyconfig.h.in regenerado?
* O remetente foi adicionado a Misc/ACKS?
* Uma entrada foi adicionada em Misc/NEWS.d/next?
* Foi atualizado?
* Os rótulos apropriados foram adicionados para significar o backport necessário da solicitação pull?
* A solicitação de pull foi feita primeiro contra o máster branch?

Para ser mais cuidadoso com o fluxo de trabalho é recomendável manter um fork do repositório principal por perto, pois permite a reversão simples de todas as mudanças locais caso o seu repositório clone local não esteja em um estado desejável.

O ciclo de vida de uma solicitação pull no CPython, usa um fluxo baseado nas solicitações pull, então o desenvolvedor cria uma branch no Git, faz suas alterações e envia para o seu fork, e em seguida cria ma solicitação de pull no repositório oficial do CPython.

A resolução de conflitos de merge são as ramificações que podem conter alterações incompatíveis em um ou mais arquivos. Que devem ser resolvidos de forma manual. Verificando quais são os arquivos que possuem conflitos, assim que identificado é necessário fazer a edição arquivos afetados e modifica-los ao estado final pretendido e confirmar o merge.

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. A solicitação deve ser testada e deixar comentários na solicitação pull rastreando os problemas, fornecendo detalhes adicionais e contexto de seu processo de revisão. E assim que que uma solicitação pull atinge um estado aceitável, ela será mesclada ou rejeitada.