**Tutorial ComoTrabalhar com Merge do Github**

**Breve explicação geral:**

O merge do Github é um mecanismo que permite mesclar conteúdo (alterações feitas em um arquivo) ao próprio arquivo fonte, sendo ele estando sendo modificado ou não. Permitindo trabalho síncrono e paralelo. É recomendado para uso em repositórios compartilhados, onde se têm vários desenvolvedores trabalhando em conjunto ao mesmo tempo. Exemplo: dois desenvolvedores baixam o repositório no seu computador local e logo começam a trabalhar. Sem o merge, quando um deles commitar e subir para o repositório as alterações, o outro desenvolvedor não poderá também subir modificações (commit e push) normalmente. Ao tentar subir, a este desenvolvedor será retornada uma mensagem informando que há atualizações no repositório e que ele precisa baixá-las antes de subir conteúdo. Ao baixar ele terá que inserir novamente suas modificações no repositório – manualmente, geralmente com Ctrl c +Ctrl v – para ter liberdade de subi-las. Agora, imagine os dois ou até mais desenvolvedores alterando o mesmo arquivo, um commita primeiro e os demais precisam logo após commitar. O que fazer? Retrabalho excessivo e desgastando para todos! Perca de tempo e dinheiro para uma empresa. Com o merge do git, cada desenvolvedor passa a ter uma área única de armazenamento para suas modificações, chamada de Branch. Cada qual podem jogar suas modificações para depois uní-las com a Branch master (que é a padrão que todo projeto tem ao se criar um repositório). Com tal união (branch x master) os conteúdos são mesclados sem perda de informações e evitando possíveis conflitos de códigos - informações.

**Breve explicação: branch e merge:**

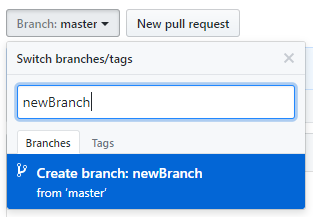
Para usar o merg, deve-se criar duas branchs: uma local (no seu PC) e outra remota que fará referência à local (no Github Web). Na local (nova branch) você irá adiciona medicações e commitar. Após isso, você deve ir na branch master e adicionar a ela a nova branch, com o merg. Após isso você deve dá o push na master para a nova branch. Nunca se deve dá push para a master, sempre dê para a nova branch. E assim segue o mesmo ciclo para cada modificação que queira subir. Segue os passos dos comandos abaixo.

**Criar branch remota (no Github)**

1. Com repositório aberto vá até:



1. Clique no checkbox e no campo com placeholder ‘find or create a branch’ digite o nome da branch que deseja criar e pressione Enter do teclado.



1. Você já tem sua Branch remota. Agora vamos partir para terminal do git e criar a branch local que fará referência a esta.
2. Para criar branch local: no terminal, coloque git checkout -b romulo. O -b é para caso não exista a branch, à criar.
3. Criada a branch você agora está à acessando no terminal, note no terminal que você agora está em newBranch. Você está livre para fazer modificações e adicioná-las a staging area e às commitar dentro da branch. NADA DE DÁ PUSH.
4. Para mesclar - adicionar as suas modificações à máster -, alterne para master no seu terminal com: git checkout master.
5. Feito isso, agora você pode dá o merge, digite: git merge newBranch.
6. Agora, dê o push para a branch. Digite: git push origin newBranch.
7. Parabéns, você subiu a branch. Agora precisa de mesclar com a master remota.
8. No git, vá em compare pull request e selecione na tela que se abrirá, no primeiro checkbox master e no segundo newBranch. E agora vá em create pull request. Uma nova tela abrirá, agora click em Merge pull request (se quiser antes adicionar um comentário – à vontade), depois vá em confirm e ‘vualá’, sua branch foi mesclada com a master.
9. Por segurança, volte a branch newBranch no seu terminal para caso de não modificar a master. Você agora só alterará a branch. Para voltar para master digite: git checkout newBranch.
10. Sempre que fizer um push para a branch, você deve fazer o compare pull request no Github Web para mesclar na remota.