- Merge de Branchs na prática:

Comando: git merge <nome da branch>

O merge ira pegar os arquivos da branch informada a junta-los a branch atual, mas para isso ele precisara criar um commit no histórico

Exemplo via git bash:



Como podes ver no inicio os arquivos da branch “master” foram adicionados a branch “test2”, e no “git log” temos o commit “Merge branch ‘test’” que registra esta ação.

Segue abaixo como fica o git log –graph após o merge de branchs:



No ínicio temos os commits “Add foo” e “Add fiz” que são da branch master e conforme subimos o histórico vemos que o commit “Add bar” esta na branch test e foi adicionado a branch master por meio do merge, registrado pelo commit “Merge branch ‘test’”.

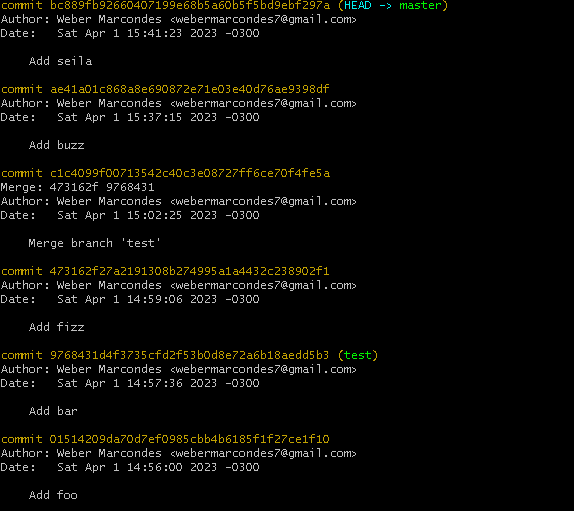
- Rebase de branchs na prática:

Comando: git rebase <nome do branch>

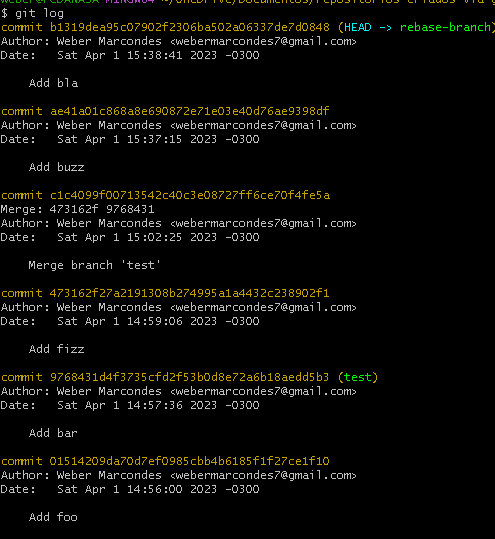
Faz o mesmo processo do merge mas não cria um commit de registro como o merge faz.

Imagine que temos as branchs “master” e “rebase-branch” com os seguintes logs:

- log de “master”:



- log de “rebase-branch”:



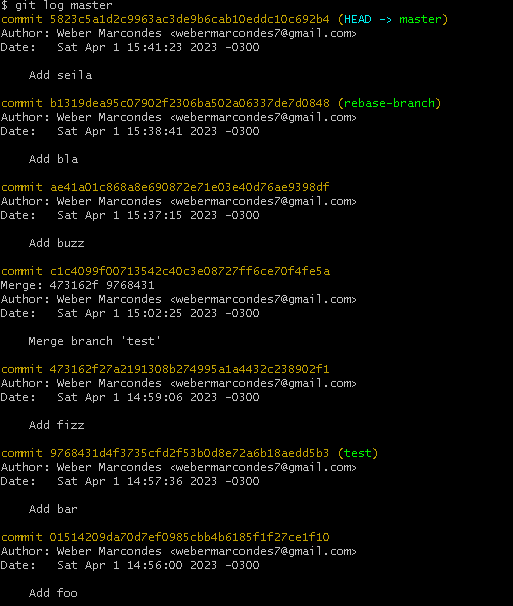
Como podes ver a branch “master” não tem o commit “Add bla” e a branch “rebase-branch” não tem o “Add seila”. Digamos que seja necessário adicionar o “Add bla” a “master”, como fazer isso?

Exemplo via git bash:



Obs.: antes de usar o rebase esteja na branch da qual deseja adicionar os commits da branch mencionada no comando.

Conforme menciona acima o “Rebase” não cria um commit de registro como o merge, veja abaixo como o “git log” da master ficou após o “Rebase”:



O commit “Add bla” da rebase-branch foi adicionado ao histórico conforme visto acima.

Obs.: Uma das desvantagens do rebase é que a ordem cronológica do histórico pode ser afetada.