Formação Desenvolvedor Python   
Ultima.School

## **Módulo 3 -** Controle de versão com GIT **Semana 1-** Iniciando com Sistema de controle de Versionamento

### Detalhes:

**Instruções**: O roteiro foi criado para ajudar a fixar o conteúdo teórico desta semana. As soluções podem ser o mais simples possível, mas é bom tentar explicar o máximo possível. Você pode usar quaisquer ferramentas que julgar necessário para exemplificar o cenário pedido

**Tutores**: Caso você precise de ajuda para completar os exercícios, os tutores estão à disposição no chat do telegram!

**Feedback**: Os tutores vão revisar as atividades enviadas pela plataforma, na página da atividade; e vão dar feedback!

**Soluções/Gabarito**: os alunos que tenham enviado as atividades, e já tenham recebido feedback; podem pedir acesso às soluções/gabarito dos exercícios se tiverem interesse. Este será liberado e estará disponível somente semanas após a conclusão do módulo em questão para os alunos que concluíram as atividades.

### Roteiro

### **Cenário1:** Você trabalha em uma equipe que deseja utilizar o git em seus projetos, vocês já possuem um repositório criado e precisam configurar o repositório (configurações do projeto, bibliotecas, documentação e etc) para começarem a trabalhar.

* Descreva o que deverá ser feito para conseguir realizar o cenário acima. Lembre-se de que é necessário efetuar a configuração e disponibilizá-la para os demais colegas de equipe. Cite comandos que serão usados, qual a ordem que serão usados e etc

RESPOSTA: Com o repositório criado, devemos definir a visibilidade do repositório para equipe, ou público, se for o caso. Adicionaremos a branch main, e faremos um clone, uma cópia local do arquivo.

### **Cenário2:** Você e seus colegas de equipe estão trabalhando em cima do mesmo conjunto de arquivos. Você precisa de uma atualização no seu repositório que algum de seus colegas acabou de finalizar e fez o merge para a branch main.

* Descreva o que deverá ser feito para conseguir realizar o cenário acima. Cite comandos que serão usados, qual a ordem que serão usados e etc

RESPOSTA: Após adicionadas as branches dos colegas, iremos utilizar o comando pull, como sabemos que o merge foi na branch main, não precisamos especificar qual branch queremos dar o pull.

### **Cenário3:** Após um dia de trabalho você finalizou a codificação de uma tarefa e precisa unir o seu código com a main branch.

* Descreva o que deverá ser feito para conseguir realizar o cenário acima. Cite comandos que serão usados, qual a ordem que serão usados e etc

RESPOSTA: Iremos utilizar o comando push.

### **Cenário4:** A sua equipe estava tendo muitos problemas com o versionamento do código e decidiu pensar em uma estratégia para poder organizar melhor o código através de “versões candidatas” para produção. Para isso foi decidido criar uma nova branch com o padrão de nome “rc\_v150” (abreviação para “*Release Candidate Version 1.5.0*”) para a primeira versão a ser usada nesta nova estratégia.

* Descreva o que deverá ser feito para conseguir realizar o cenário acima. Lembre-se de comentar como que a equipe irá unir o código nessa branch, como criá-la e etc. Cite comandos que serão usados, qual a ordem que serão usados e etc

RESPOSTA: Iremos utilizar a ferramenta TAG, iremos criar uma nova branch com essa tag, assim todos deverão fazer o pull dessa nova branch com tag, e trabalhar a partir dela.

### **Cenário5:** Após analisar um problema no código, você fez uma proposta do que precisa ser alterado para que o bug seja resolvido. Para realizar a correção, será necessário criar uma branch, codificar a correção e enviar o código para que outras pessoas do time possam avaliá-lo.

* Descreva o que deverá ser feito para conseguir realizar o cenário acima. Lembre-se. Cite comandos que serão usados, qual a ordem que serão usados e etc

RESPOSTA: Faremos o pull da branch mais atual, modificamos o que se fizer necessário, faremos o push da versão modificada e o merge para juntarmos as modificações, deixando disponível para os demais.